



UNIFEI
Universidade Federal de Itajubá

Instituto de Engenharia Elétrica
Departamento de Eletrotécnica

AQUISIÇÃO AUTOMÁTICA DE CONHECIMENTO
APLICADA EM AVALIAÇÃO DE SEGURANÇA DE
TENSÃO

Área de Sistemas Elétricos de Potência

Pedro José Abrão



AQUISIÇÃO AUTOMÁTICA DE CONHECIMENTO APLICADA EM AVALIAÇÃO DE SEGURANÇA DE TENSÃO

**ORIENTADOR: PROF. ALEXANDRE PINTO ALVES DA SILVA
CO-ORIENTADOR: PROF. ANTONIO CARLOS ZAMBRONI DE SOUZA**

**TESE APRESENTADA À
UNIVERSIDADE FEDERAL DE ITAJUBÁ
COMO REQUISITO DO PROGRAMA DE DOUTORADO**

**ITAJUBÁ
ESTADO DE MINAS GERAIS – BRASIL
JULHO/ 2002**

Pedro José Abrão.

Aquisição Automática de Conhecimento Aplicada em Avaliação de Segurança de Tensão / Pedro José Abrão – UNIFEI – 2002, 262p.

Tese de Doutorado apresentada a UNIFEI para obtenção do grau de Doutor em Ciências em engenharia.

1. Estabilidade de Tensão 2. Avaliação de Segurança de Tensão 3. Redes Neurais Artificiais 4. Árvores de Decisão
5. Extração de Regras de Produção de Redes Neurais

I. Título. *Aquisição Automática de Conhecimento Aplicada em Avaliação de Segurança de Tensão*

Banca Examinadora

Prof. Alexandre Pinto Alves da Silva – Orientador - UNIFEI

Prof. Antonio Carlos Zambroni de Souza, Co-orientador - UNIFEI

Prof. Djalma Mosqueira Falcão – COPPE/UFRJ

Prof. Ricardo Bernardo Prada – PUC/RJ

Prof. Germano Lambert Torres, UNIFEI

Prof. Pedro Paulo de Carvalho Mendes, UNIFEI

À minha mãe, Almira.

À minha esposa, Ruth, e aos meus filhos, Lucas e Thiago

À minha irmã, Christiane.

Agradecimentos

A Deus, pela sua infinita sabedoria e amor.

A minha mãe, pelo amor que sempre dedicou a mim, clareando sempre o meu caminho e ajudando-me a vencer as dificuldades com esforço e bondade.

A minha esposa, Ruth, que sempre esteve ao meu lado com sua enorme compreensão e amor.

Aos meus filhos, Lucas e Thiago, que sofreram com a minha ausência infundável, tornando suas infâncias menos tranqüilas.

A minha querida irmã, que mesmo em seus momentos de dificuldade, sua amizade e amor me ajudaram a transformar as tristezas em alegrias.

Ao meu orientador e amigo, Alexandre, pelo reconhecimento do meu esforço e auxílio nas horas certas.

Ao amigo Zambroni, pelas palavras de incentivo e por sempre acreditar que as coisas boas sempre dão certo.

Ao amigo Leonardo, pelo companheirismo, auxílio e pelos bons papos e risadas que tivemos juntos ao longo dessa jornada.

Ao amigo Elder, pela acolhida e amizade dispensada.

Ao amigo Luciano Lima, que além de nossa grande amizade, me proporcionou as melhores condições de trabalho na UFU.

Aos meus colegas de pós-graduação, Luis Manso, Júlia, Agnaldo, Luciano, Agnelo, Paulo César, Ubiratan, Helga, Isaías e muitos outros pelo companheirismo e agradável convívio.

Aos colegas e amigos do CEFETGO - Jataí pela compreensão e ajuda nessa empreitada.

RESUMO

Um dos problemas mais desafiadores em operação de tempo real de sistemas de potência está associado com a Avaliação de Segurança de Tensão (AST), estando diretamente relacionada com o problema da estabilidade de tensão. Esta pode ser definida como a habilidade do sistema de potência em manter um perfil de tensão apropriado em todas as suas barras. A redução da reserva de potência reativa disponível no sistema é uma das principais causas do fenômeno da instabilidade de tensão, que é caracterizado por uma redução progressiva nas magnitudes da tensão.

Com a meta de entender e solucionar o problema de estabilidade de tensão foram propostas várias metodologias durante os últimos anos. Porém, a maioria delas apresenta um alto custo computacional. As técnicas analíticas para a solução do problema de estabilidade não são factíveis para os operadores do sistema implementarem ações de controle preventivas ou corretivas em tempo adequado. Uma possível solução para superar esta desvantagem é a aplicação de técnicas de inteligência artificial relacionadas com aprendizagem automática, as quais devem estar associadas com uma metodologia eficiente para a geração dos padrões de treinamento.

Uma das técnicas de aprendizado automático aplicada em AST *on-line* são as árvores de decisão. Entretanto, as árvores de decisão têm se mostrado imprecisas para esta aplicação. Por outro lado, as redes neurais artificiais têm apresentado excelente precisão para as tarefas de classificação e regressão em VSA. A principal desvantagem do modelo neural é sua opacidade, i.e., apresentam baixo grau de compreensão humana para o processo de inferência, estando o conhecimento codificado (representado pelos valores de pesos de conexões).

Este trabalho lida com esta desvantagem do modelo neural para uma aplicação em VSA. Para isso, é empregado um algoritmo que qualitativamente interpreta a base de conhecimento de uma rede neural tipo *feedforward*. O trabalho também propõe a aplicação de um algoritmo de treinamento muito rápido para este tipo de rede neural, denominado de *Optimal Estimate Training 2* (OET2). Este algoritmo tem a propriedade de lidar com grandes bases de dados satisfatoriamente. Além disso, este é plenamente compatível com o algoritmo para extração de regra *Validity Interval Analysis* (VIA), utilizado neste trabalho. A motivação principal para o modelo proposto é dar ao operador de um sistema elétrico um conjunto de regras simbólicas consistente com o processo de inferência neural.

ABSTRACT

One of the most challenging problems in real time operation of power systems is associated with voltage stability assessment (VSA). The problem of voltage stability is related to the ability of the power system to maintain an appropriate voltage profile on all its buses. Being directly associated with the reduction of the available reactive power reserve, the voltage instability phenomenon is characterized by a progressive reduction on the voltage magnitudes.

With the goal of understanding and solving the voltage stability problem, several methodologies have been proposed during the last few years. However, most of them require a high computational burden. Analytical techniques for solving the VSA problem are infeasible to the operators to implement preventive or corrective control actions in due time. One possible solution to overcome this drawback is the application of automatic learning techniques associated with an efficient methodology for generating training patterns.

Decision trees have been applied for on-line VSA. Nevertheless, decision trees are among the machine learning techniques with the highest variance. On the other hand, artificial neural networks (ANNs) have shown outstanding precision for classification and regression tasks. The major shortcoming of the ANN approach is its opacity, i.e., its low degree of human comprehensibility regarding its inference process and encoded knowledge (represented by the values of synaptic weights).

This work tackles the mentioned above drawback of the ANN approach for VSA. An algorithm for qualitatively interpreting the knowledge base of a nonlinear feedforward ANN is employed. The work also proposes the application of a very fast training algorithm, Optimal Estimate Training 2 (OET2), which is fully compatible with the algorithm for rule extraction, Validity Interval Analysis (VIA), and suitable for dealing with very large databases. The main motivation for the proposed approach is to give the power system operator a set of the symbolic rules, which is consistent with the ANN inference process.

ÍNDICE

Capítulo 1

INTRODUÇÃO	1
1.1 O Sistema Elétrico de Potência atual	1
1.2 Objetivo do Trabalho	3
1.2.1 Ferramentas de Avaliação de Segurança de Tensão	3
1.2.2 Técnicas de IA Aplicadas em AST	6
1.2.3 Aquisição Automática de Conhecimento	7
1.3 O Estado da Arte da Aplicação de Técnicas de IA em AST	9
1.3.1 Redes Neurais Artificiais Aplicadas em Estabilidade de Tensão	9
1.3.2 Aprendizado Automático Aplicado em AST	14
1.4 Etapas do Trabalho	17

Capítulo 2

AVALIAÇÃO DE SEGURANÇA DE TENSÃO	21
2.1 Introdução	21
2.2 Avaliação de Segurança	21
2.2.1 Requisitos gerais para análise de AS	22
2.2.2 Estados Operativos	23
2.2.3 Estabilidade em Sistemas Elétricos de Potência	25
2.2.4 Ferramentas para AST	27
2.3 Ferramentas Convencionais em ET utilizadas para a Formação da Base de Dados	28
2.3.1 Método da Continuação	32
2.3.2 Método do Vetor Tangente	36
2.4 Conclusões	40

Capítulo 3

AQUISIÇÃO AUTOMÁTICA DE CONHECIMENTO	41
3.1 Introdução	41
3.2 Aquisição Automática de Conhecimento	42
3.3 Pré-Processamento dos dados	46
3.4 Técnicas de Inferência na Base de dados	49
3.5 Árvores de Decisão	50
3.6 Conceitos Básicos Sobre Redes Neurais Artificiais	61
3.7 Extração de Regras de Produção de Redes Neurais Artificiais	65
3.8 Validity Interval Analysis (VIA)	76

3.8.1 VIA e a Rede Neural Baseada no Treinamento OET2	82
3.8.2 Resultados do OET2 para o Sistema 3 Barras	83
3.8.3 Fases Forward e Backward do VIA	85
3.8.4 Metodologia para Testar as Hipóteses Obtidas pelo C4.5	89
3.9 Conclusões	93

Capítulo 4

SIMULAÇÕES COMPUTACIONAIS E RESULTADOS	96
4.1 Introdução	96
4.2 Base de dados – Critérios de Formação	96
4.3 Simulações	101
4.3.1 Seleção do Caso	103
4.3.2 Seleção de Variáveis	103
4.3.3 Conjuntos de Treinamento e Teste	104
4.3.4 OET2	105
4.3.5 Estimação da Margem de Potência Ativa	106
4.3.6 C4.5 e Regras C4.5	107
4.3.7 Estatísticas Regras C4.5	109
4.3.8 VIA	110
4.3.9 Estatísticas Regras VIA:	112
4.3.10 Índice de Ganho de Informação:	113
4.3.11 Refinamento das Novas Regras do VIA	117
4.4 Resultados das Simulações	119
4.4.1 Resultados das Simulações Para o Sistema Teste IEEE 14 Barras	120
4.4.1.1 Caso 1 - Contingência Perda do Compensador Síncrono – Barra 6	121
4.4.1.2 Caso 2 - Contingência Perda de Capacitor - Barra 9	123
4.4.1.3 Caso 3 - Contingência Perda de Linha de Transmissão – Barras 02 - 04	125
4.4.1.4 Caso 4 - Contingência Perda de Linha de Transmissão – Barras 04 - 05	126
4.4.1.5 Caso 5 - Contingência Perda de Linha de Transmissão – Barras 06 - 11	128
4.4.1.6 Caso 6 - Contingência Perda de Linha de Transmissão – Barras 06 - 12	130
4.4.1.7 Caso 7 - Contingência Perda de Linha de Transmissão – Barras 06 - 13	132
4.4.1.8 Caso 8 - Contingência Perda de Linha de Transmissão – Barras 09 - 10	133
4.4.1.9 Caso 9 - Contingência Perda de Gerador – Barra 2	135
4.4.2 Resultados das Simulações para o Sistema Teste IEEE 118 Barras	137
4.4.2.1 Caso Base	137
4.4.2.2 Caso 1 – Contingência Perda do Compensador Síncrono – Barra 76	139
4.4.2.3 Caso 2 – Contingência Perda de Linha de Transmissão - Barras 76 - 77	141
4.4.3 – Discussões Sobre as Simulações	142
4.5 Conclusões	144

Capítulo 5

CONCLUSÕES E PROPOSTAS	146
5.1 Introdução	146
5.2 Conclusões do Trabalho	146
5.3 Proposta para Trabalhos Futuros	153

Apêndice A

ESTABILIDADE DE TENSÃO	155
A.1 Introdução	155
A.2 Matrizes de Sensibilidade	154
A.3 Valores Singulares e Autovalores	155
A.4 Determinante Reduzido	159
A.5 Método Direto	161
A.6 Técnica da Função de Energia	162
A.7 Técnicas de Partição de Redes	163
A.8 Principais Considerações sobre Estabilidade de Tensão	164

Apêndice B

TÉCNICAS DE RECONHECIMENTO DE PADRÕES	166
B.1 Introdução	166
B.2 Classificador de Bayes	166
B.3 Estimação de Densidades de Probabilidades	167
B.3.1 Método Baseado em Kernel:	170
B.3.2 Método dos K-Vizinhos-Mais-Próximos:	172
B.4 Funções Discriminantes	174
B.5 Ajuste de Curvas por Polinômios	178
B.6 Observações Importantes	180

Apêndice C

REDES NEURAIIS ARTIFICIAIS	182
C.1 Introdução	182
C.2 Conceitos Básicos - O Neurônio Artificial	182
C.3 Estratégia de Aprendizagem Supervisionada	185
C.3.1 RNA com uma Camada Treinada com a Regra de Hebb	186
C.3.2 RNA <i>Feedforward</i> Multicamadas com Retropropagação do Erro	187
C.3.3 Rede Neural <i>Feedforward</i> com Treinamento OET2	192

Apêndice D

BRAINEE	196
D.1 Introdução	196
D.2 BRAINNE	196
D.2.1 BRAINNE Aplicado a dados Discretos	200
D.2.2 BRAINNE Aplicado a dados Contínuos	215
D.2.3 Exemplo de Aplicação para dados Contínuos	222

Apêndice E

REGRAS DE PRODUÇÃO – C4.5	230
E.1 Introdução	230
E.2 Sistema Elétrico Teste IEEE 14 Barras	230
E.3 Sistema Elétrico Teste IEEE 118 Barras	235

Apêndice F

REGRAS DE PRODUÇÃO – VIA	237
F.1 Introdução	237
F.2 Sistema Elétrico Teste IEEE 14 Barras	237
F.3 Sistema Elétrico Teste IEEE 118 Barras	243

Apêndice G

SISTEMAS ELÉTRICOS – DADOS	246
G.1 Sistema Elétrico Teste IEEE 14 Barras	246
G.1.1 Dados	246
G.1.2 Diagrama Unifilar do Sistema Teste IEEE 14 Barras	249
G.2 Sistema Elétrico Teste IEEE 118 Barras	250
G.2.1 – Dados	250

<i>Referências Bibliográficas</i>	256
--	-----

ÍNDICE DE FIGURAS

Figura 1.1 - Esquema proposto para um aplicativo para AST.	18
Figura 2.1 - Tipos de estabilidade em sistemas de potência.	26
Figura 2.2 - Ilustração gráfica do método da continuação.	34
Figura 2.3 - Diagrama da bifurcação para o sistema teste IEEE 14 barras simulado considerando-se apenas a presença de limites de fornecimento de potência reativa.	36
Figura 2.4 - Diagrama da bifurcação para o sistema teste IEEE 14 barras simulado com um equipamento LTC controlando a barra 9 do sistema e limites de fornecimento de potência reativa.	36
Figura 3.1 - Esquema básico de um Sistema Baseado em Conhecimento.	42
Figura 3.2 - (a) curvas de distribuição levando em conta somente a avaliação da característica C1, (b) avaliação conjunta das características C1 e C2, mostrando a fronteira de decisão.	50
Figura 3.3 - Exemplo de uma árvore de decisão dividindo o conjunto de treinamento.	51
Figura 3.4 - Sistema elétrico com 3 barras.	57
Figura 3.5 - Representação neural da função discriminante linear $g(x)$ para um problema multiclases.	62
Figura 3.6 - Modelo de um neurônio com função de ativação linear.	65
Figura 3.7 - Parte de uma RNA mostrando as saídas das ativações.	80
Figura 4.1 - Estrutura geral do aplicativo para avaliação de estabilidade de tensão e obtenção da base de dados para as simulações do algoritmo VIA.	97
Figura 4.2 - Curvas PV para o caso base, considerando as ações do LTC e do despacho de geração, FP aleatório e limites de fornecimento de potência reativa.	100
Figura 4.3 - Fluxograma do Programa VSA DATA MINING para obtenção das regras de produção que compõe a base de conhecimento em VSA.	102
Figura 4.4 - Erro médio quadrático para o caso base utilizando a RNA OET2 para estimação da margem de potência ativa (pu) com uma perturbação aleatória de 5%.	106
Figura 4.5 - Erro médio quadrático para o caso base utilizando a RNA OET2 para estimação da margem de potência ativa (pu) com uma perturbação aleatória de 15%.	107
Figura B.1 - Regiões de decisão para um problema à duas classes, C1 e C2.	169
Figura B.2 - Região de solução no espaço dos pesos, mostrando a orientação do vetor solução 'a'.	177
Figura C.1 - Neurônio Artificial.	183
Figura C.2 - Modelo de neurônio com uma função de ativação do tipo threshold.	184
Figura C.3 - Representação gráfica da Função Sigmóide.	184
Figura C.4 - Estrutura Neural tipo Multicamadas <i>Feedforward</i> .	188
Figura C.5 - Neurônio Artificial.	189
Figura C.6 - RNA com uma camada oculta.	193
Figura D.1 - Rede Neural Artificial com camadas simples.	197

Figura D.2 - Topologias propostas para RNAs: (a) configuração original; (b) configuração com as saídas desejadas como entradas adicionais.	201
Figura D.3 - Cálculo do SSE entre a entrada original a e a entrada adicional b.	202
Figura D.4 - Obtenção das primeiras regras para o Exemplo Teste.	207
Figura D.5 - Nova configuração da Rede Neural.	211
Figura D.6 - Visualização do processo de extração de regras para o exemplo.	214
Figura G.1 - Diagrama unifilar do sistema teste IEEE 14 barras.	249

ÍNDICE DE TABELAS

Tabela 2.1 - Margem de potência ativa, ranking das 3 barras mais críticas e a posição de cada contingência.	39
Tabela 3.1 - Conjunto de treinamento para o sistema elétrico 3 barras.	58
Tabela 3.2 - Conjunto de teste para o sistema elétrico 3 barras.	57
Tabela 3.3 - Árvore de decisão obtida pelo C4.5 para o Caso Base.	58
Tabela 3.4 - Taxa de classificação da árvore de decisão extraída pelo C4.5. Os valores sublinhados denotam classificação correta. – Caso base sistema teste 3 barras.	58
Tabela 3.5 - Regras C4.5 - Caso base sistema teste 3 barras.	58
Tabela 3.6 - Taxa de classificação das regras do C4.5. – Caso base sistema teste 3 barras.	58
Tabela 3.7 - Taxa de classificação por regra do C4.5 para o sistema teste de 3 barras.	59
Tabela 3.8 - Mapeamento binário das variáveis.	66
Tabela 3.9 - Quadro resumo dos algoritmos para extração de regras de RNAs, mostrando algumas características mais importantes para análise.	76
Tabela 3.10 - Valores mínimos e máximos de cada variável.	81
Tabela 3.11 - Pesos das conexões das 2 camadas.	83
Tabela 3.12 - intervalos do valor de net (neurônios da camada oculta).	84
Tabela 3.13 - intervalos do valor de net (neurônios da camada de saída).	84
Tabela 3.14 - Taxa de classificação obtida pelo OET2 para o conjunto de treinamento do sistema 3 barras.	84
Tabela 3.15 - Resultado da fase Forward para a regra R1, após convergência (2a iteração), onde são mostrados os intervalos refinados para cada neurônio.	86
Tabela 3.16 - Resultado da fase Backward para a regra R1, pós convergência (2a iteração), onde são mostrados os intervalos refinados para cada neurônio.	87
Tabela 3.17 - Taxa de classificação obtida pelo OET2 para o conjunto de teste do sistema 3 barras.	88
Tabela 3.18 - Taxa de classificação por regra extraída do VIA para o sistema 3 barras - conjunto de teste. Hipótese inicial com os intervalos abertos.	89
Tabela 3.19 - Regras originais C4.5 e as novas obtidas pelo VIA, mostrando a alteração percentual do intervalo original em relação ao intervalo novo de uma variável.	92
Tabela 3.20 - Taxa de classificação por regra extraída do VIA para o sistema 3 barras - conjunto de teste. Hipóteses iniciais dadas pelo C4.5.	92
Tabela 4.1 - Variáveis selecionadas os sistemas testes IEEE.	101
Tabela 4.2 - Conjunto de dados para o caso base do sistema teste IEEE 14 barras.	104
Tabela 4.3 - Frequência das barras críticas (5 posições) em função do nível operativo do sistema elétrico teste IEEE 14 barras.	104
Tabela 4.4 - Taxa de classificação do OET2. Os valores sublinhados denotam classificação correta. Os demais, classificação errada. Mostra também o total geral da classificação – Caso base sistema teste IEEE 14 barras.	106
Tabela 4.5 - Árvore de decisão obtida pelo C4.5 para o Caso Base.	108

Tabela 4.6 - Taxa de classificação da árvore de decisão extraída pelo C4.5. Os valores sublinhados denotam classificação correta. Os demais, classificação errada. Mostra também o total geral da classificação – Caso base sistema teste IEEE 14 barras.	108
Tabela 4.7 - Regras C4.5 - Caso base sistema teste IEEE 14 barras.	109
Tabela 4.8 - Taxa de classificação do conjunto de regras extraída pelo C4.5. Os valores sublinhados denotam classificação correta. Os demais, classificação errada. Mostra também o total geral da classificação – Caso base sistema teste IEEE 14 barras.	109
Tabela 4.9 - Taxa de classificação por regra do C4.5 para o caso base do sistema teste IEEE 14 barras.	110
Tabela 4.10 - Regras originais C4.5 e as novas obtidas pelo VIA.	111
Tabela 4.11 - Taxa de classificação por regra do VIA para o caso base do sistema teste IEEE 14 barras.	112
Tabela 4.12 - Índice de ganho de informação para cada regra para o caso base do sistema teste IEEE 14 barras.	116
Tabela 4.13 - Novas regras refinadas para o sistema teste IEEE 14 Barras – caso base	118
Tabela 4.14 - Índice de ganho de informação das regras refinadas - Ganho (VIA_i/VIA_{ii}), para o caso base do sistema teste IEEE 14 barras.	118
Tabela 4.15 - Taxa de classificação por regra do VIA – refinamento - para o caso base do sistema teste IEEE 14 barras.	119
Tabela 4.16 - Estimação da margem de potência ativa pela rede neural OET2 para algumas contingências do sistema teste IEEE 14 barras considerando 15 % de perturbação aleatória.	120
Tabela 4.17a - Taxa de classificação OET2 – Caso 1.	122
Tabela 4.17b - Taxa de classificação C4.5 – Caso 1.	122
Tabela 4.17c - Taxa de classificação por regra C4.5 – Caso 1.	122
Tabela 4.17d - Índice de ganho de informação - C4.5/VIA e VIA_i/VIA_{ii} – Caso 1.	122
Tabela 4.17e - Taxa de classificação por regra VIA_{ii} – Caso 1.	123
Tabela 4.18a - Taxa de classificação OET2 – Caso 2.	123
Tabela 4.18b - Taxa de classificação C4.5 – Caso 2.	124
Tabela 4.18c - Taxa de classificação por regra C4.5 – Caso 2.	124
Tabela 4.18d - Índice de ganho de informação - C4.5/VIA e VIA_i/VIA_{ii} – Caso 2.	124
Tabela 4.18e - Taxa de classificação por regra VIA_{ii} – Caso 2.	124
Tabela 4.19a - Taxa de classificação OET2 – Caso 3.	125
Tabela 4.19b - Taxa de classificação C4.5 – Caso 3.	125
Tabela 4.19c - Taxa de classificação por regra C4.5 – Caso 3.	125
Tabela 4.19d - Índice de ganho de informação - C4.5/VIA - VIA_i/VIA_{ii} – Caso 3.	126
Tabela 4.19e - Taxa de classificação por regra VIA_{ii} – Caso 3.	126
Tabela 4.20a - Taxa de classificação OET2 – Caso 4.	127
Tabela 4.20b - Taxa de classificação C4.5 – Caso 4.	127
Tabela 4.20c - Taxa de classificação por regra C4.5 – Caso 4.	127
Tabela 4.20d - Índice de ganho de informação - C4.5/VIA – VIA_i/VIA_{ii} –Caso 4.	127
Tabela 4.20e - Taxa de classificação por regra VIA_{ii} – Caso 4.	128
Tabela 4.21a - Taxa de classificação OET2 – Caso 5.	128

Tabela 4.21b - Taxa de classificação C4.5 – Caso 5.	129
Tabela 4.21c - Taxa de classificação por regra do C4.5 – Caso 5.	129
Tabela 4.21d - Índice de ganho de informação – C4.5/VIA – VIAi/VIAii – Caso 5.	129
Tabela 4.21e - Taxa de classificação por regra VIAii – Caso 5.	129
Tabela 4.22a - Taxa de classificação OET2 – Caso 6.	130
Tabela 4.22b - Taxa de classificação C4.5 – Caso 6.	130
Tabela 4.22c - Taxa de classificação por regra C4.5 – Caso 6.	131
Tabela 4.22d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso 6.	131
Tabela 4.22e - Taxa de classificação por regra pelo VIAii – Caso 6 .	131
Tabela 4.23a - Taxa de classificação OET2 – Caso 7.	132
Tabela 4.23b - Taxa de classificação C4.5 – Caso 7.	132
Tabela 4.23c - Taxa de classificação por regra C4.5 – Caso 7.	132
Tabela 4.23d - Índice de ganho de informação C4.5/VIA – VIAi/VIAii – Caso 7.	133
Tabela 4.23e - Taxa de classificação por regra VIAii – Caso 7.	133
Tabela 4.24a - Taxa de classificação OET2 – Caso 8.	133
Tabela 4.24b - Taxa de classificação C4.5 – Caso 8.	134
Tabela 4.24c - Taxa de classificação por regra C4.5 – Caso 8.	134
Tabela 4.24d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso 8.	134
Tabela 4.24e - Taxa de classificação por regra VIAi – Caso 8.	134
Tabela 4.25a - Taxa de classificação OET2 – Caso 9.	135
Tabela 4.25b - Taxa de classificação C4.5 – Caso 9.	135
Tabela 4.25c - Taxa de classificação por regra C4.5 – Caso 9.	136
Tabela 4.25d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso 9.	136
Tabela 4.25e - Taxa de classificação por regra VIAi – Caso 9.	136
Tabela 4.26 - Estimção da margem de potência ativa pela rede neural OET2 para algumas contingências do sistema teste IEEE 118 barras considerando 15 % de perturbação aleatória.	137
Tabela 4.27a - Taxa de classificação OET2 – Caso base.	137
Tabela 4.27b - Taxa de classificação C4.5 – Caso base	138
Tabela 4.27c - Taxa de classificação por regra C4.5 – Caso base.	138
Tabela 4.27d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso base	138
Tabela 4.27e - Taxa de classificação por regra VIAi – Caso base.	139
Tabela 4.28a - Taxa de classificação OET2 – Caso 1.	139
Tabela 4.28b - Taxa de classificação C4.5 – Caso 1.	140
Tabela 4.28c - Taxa de classificação por regra C4.5 – Caso 1.	140
Tabela 4.28d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso 1.	140
Tabela 4.28e - Taxa de classificação por regra VIAi – Caso 1.	140
Tabela 4.29a - Taxa de classificação OET2 – Caso 2.	141
Tabela 4.29b - Taxa de classificação C4.5 – Caso 2.	141
Tabela 4.29c - Taxa de classificação por regra C4.5 – Caso 2.	141
Tabela 4.29d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso 2.	142
Tabela 4.29c - Taxa de classificação por regra C4.5 – Caso 2.	142

Tabela D.1 - Listas de produtos.	206
Tabela D.2 - Medidas elétricas de um fictício sistema elétrico de potência.	222
Tabela D.3 - Resultados do primeiro teste.	226
Tabela D.4 - Resultados para o segundo teste.	229
Tabela G.1 - Dados do sistema.	246
Tabela G.2 - Dados de barras.	247
Tabela G.3 - Dados das linhas de transmissão e transformadores.	247
Tabela G.4 - Dados do LTC.	248
Tabela G.5 - Dados da barra do gerador.	249
Tabela G.6 - Dados dos geradores (Barra 2).	249
Tabela G.7 - Dados do sistema.	250
Tabela G.8 - Dados de barras.	250
Tabela G.9 - Dados de Linhas de transmissão.	252

ABREVIATURAS

AAC	Aquisição Automática de Conhecimento
AS	Avaliação de Segurança
AST	Avaliação de Segurança de Tensão
BRAINNE	<i>Building Representations for AI using Neural Networks</i>
CART	<i>Classification and Regression Trees</i>
C-M-S	Separação pela média da classe (<i>Class-Mean-Separation</i>)
COS	Centro de Operação do Sistema
DSA	Avaliação de Segurança Dinâmica (<i>Dynamic Security Assessment</i>)
DT	Árvore de Decisão
ES	Sistema Especialista (<i>Expert System</i>)
ET	Estabilidade de Tensão
GA	Algoritmos Genéticos (<i>Genetic algorithms</i>)
IA	Inteligência Artificial
ID3	<i>Interactive Dichotomizer Tree</i>
K-L	Expansão de Karhunen-Loève
K-NN	K-Vizinhos mais próximos (<i>K-Nearest Neighbors</i>)
LTCs	<i>Load Tap Changers</i>
MACIE	<i>Matrix Controlled Inference Engine</i>
MC	Método da Continuação
OET2	<i>Optimal Estimate Training 2 .</i>
RNA	Rede Neural Artificial
RNA-KM	RNA tipo Mapa de Kohonen
RNA-ML	RNA com múltiplas camadas e treinamento baseado na retropropagação de erros
RNA-SL	RNA sem camadas intermediárias
RT	Árvore de Regressão (<i>Regression Tree</i>)
VIA	<i>Validity Interval Analysis</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
PL	Programação Linear

Capítulo 1

INTRODUÇÃO

1.1 O SISTEMA ELÉTRICO DE POTÊNCIA ATUAL

Muitos países do mundo atualmente têm atravessado problemas no que diz respeito à questão energética. A situação crítica no fornecimento de energia é, em muitos casos, reflexo de políticas implementadas para o setor elétrico sem que as características básicas da confiabilidade e fornecimento ininterrupto de energia sejam satisfatoriamente atendidas. Muito destas políticas foram implementadas para atender a idéia dos mercados livres para a energia elétrica. O objetivo seria o aumento da competitividade e da oferta de energia, uma vez que os governos têm-se mostrado incapazes de atender o crescimento da demanda de energia.

Além das idéias mercadológicas para o setor, ocorre uma mudança de comportamento da carga elétrica, cada vez mais sensível à qualidade de fornecimento da energia elétrica. Isto determina altos investimentos para se manter a tensão e a frequência dentro das faixas operativas de tolerância (cada vez mais estreitas) em todos os pontos de um sistema elétrico. Logo, a configuração dos sistemas elétricos apresenta-se mais interligada, seja para aumentar a confiabilidade/qualidade ou para atender a demanda ou ambas. Além disso, tais sistemas devem permanecer em fornecimento contínuo de energia mesmo diante de distúrbios no sistema, como curtos-circuitos e perda de equipamentos de transmissão e geração.

Obviamente, frente ao cenário descrito, a situação operativa atual de muitos sistemas elétricos tem inspirado cuidados. Pode-se então observar que a deterioração das condições de operação imposta pelos fatores descritos é função do aumento e da mudança da característica da

carga elétrica, da falta de investimentos em geração, transmissão e equipamentos de compensação de potência reativa e por ações de controle equivocadas propostas pelo Planejamento da Operação e/ou da Operação em Tempo Real em função do desconhecimento de alguns fenômenos elétricos que podem ocorrer.

Neste ambiente, visualizar as condições de segurança elétrica torna-se um jogo de tentativa e erro. A operação do sistema perde em confiabilidade na implementação de ações de controle preventivas e/ou corretivas. Isto tem levado a situações de extremo risco operacional, ocasionando desligamentos em cascata e configurando os blecautes no sistema. Vale salientar que os custos financeiros e de credibilidade destes blecautes geralmente são altos.

Tradicionalmente, um Centro de Operação do Sistema (COS) é alimentado com instruções de operação do sistema advindas de estudos feitos com os programas computacionais convencionais como o de fluxo de carga e análise transitória. Na maioria das vezes, os estudos são feitos para as situações mais críticas e abrangem apenas situações de perda de equipamentos, avaliadas através da análise de estabilidade transitória. Os estudos de estabilidade de tensão ainda não estão devidamente contemplados nestas instruções, o que acarreta uma situação operativa indesejável. Além disso, há uma necessidade de uma complexa modelagem matemática dos equipamentos e da carga para aproximá-los do modelo real. Logo, as ações de controle indicadas são baseadas na sensibilidade do especialista em interpretar os resultados dos estudos utilizando modelos nem sempre realistas dos equipamentos do sistema elétrico, além do comportamento aleatório da carga elétrica. Isso pode levar a indicação de ações de controle que não espelham a melhor relação custo/benefício.

Entretanto, a solução dos problemas operativos em tempo real se faz cada vez mais necessária, uma vez que a dinâmica e complexidade destes problemas requerem do operador uma atuação rápida e precisa para evitar desligamentos desnecessários. Além disso, muitos problemas não são contemplados nas instruções de operação, sendo preciso uma avaliação imediata pelo operador para uma solução adequada. Desta forma, surge a necessidade de implementar nos COSs instrumentos que auxiliem o operador do sistema elétrico no processo de tomada de decisão em ambiente de tempo real. Estes instrumentos, tendo em vista a característica das ações desenvolvidas pelo por um COS, além de apresentarem respostas rápidas, devem ter alto grau de clareza na informação disponibilizada, abrangência e acesso direto. Tudo isso para que sejam atendidos os requisitos de confiabilidade no fornecimento de energia elétrica, atendimento da carga e competitividade do sistema elétrico.

Considerando especificamente o problema da Avaliação de Segurança de Tensão (AST),

abordado neste trabalho, uma ferramenta de apoio à tomada de decisões em AST deve processar primeiramente as informações que governam o problema da Estabilidade de Tensão (ET), como por exemplo, as barras críticas do sistema. A manipulação adequada destas informações deve então proporcionar ao operador do sistema elétrico a visualização das melhores ações de controle para manter ou retornar o sistema elétrico a uma posição segura de operação. Um exemplo de tais ações seria a indicação do melhor custo/benefício para proceder cortes de carga no sistema. Vale salientar que disponibilizar estas informações independe da metodologia escolhida para o processamento de AST.

De acordo com o exposto nos parágrafos anteriores, é desejável e necessário contribuir com trabalhos que proporcionem avaliações de técnicas que auxiliem o desenvolvimento de uma ferramenta de apoio à tomada de decisão em AST. Este trabalho de tese caminha nesta direção e o detalhamento do objetivo deste é mostrado no item a seguir.

1.2 OBJETIVO DO TRABALHO

Este trabalho tem o objetivo de mostrar que as Redes Neurais Artificiais (RNAs) são adequadas à tarefa de formação de um conjunto de regras de produção (*Se – Então*) [MIT97], o qual pode ser aplicável no desenvolvimento de uma ferramenta baseada em técnicas de Inteligência Artificial (IA) para o processamento de AST. A técnica utilizada neste trabalho para a extração deste conjunto de regras (denominado de Aquisição Automática de Conhecimento - AAC) de uma RNA é comparada com os resultados obtidos por uma Árvore de Decisão (DT), as principais agentes de AAC utilizadas até o momento em AST. Os sub-itens a seguir apresentam o detalhamento da proposta deste trabalho, onde são abordados os requisitos e os resultados do processamento de uma ferramenta para AST, a motivação para utilização de técnicas de IA frente às ferramentas convencionais e o porque de se fazer comparações entre uma técnica de extração de conhecimento simbólico baseada em RNAs e uma baseada em DT.

1.2.1 Ferramentas de Avaliação de Segurança de Tensão

O desenvolvimento de ferramentas computacionais destinadas a processar AST implica primeiramente em compreender o problema da ET. Conforme [KUN97], esta é definida como a

capacidade do sistema elétrico em manter as tensões de regime permanente aceitáveis em todas as barras do sistema, tanto sob condições normais de operação como após um distúrbio. O processo que leva o sistema à instabilidade de tensão é caracterizado por uma queda de tensão constante sem que os mecanismos de controle de tensão sejam eficazes para atenuar o problema. Algumas vezes, estes mecanismos atuam de modo a agravar o estado geral do sistema sob este cenário. A principal causa para a degeneração do perfil de tensão reside na insuficiente compensação de potência reativa e/ou capacidade de transmissão de potência frente ao aumento da demanda de carga. As baixas tensões implicam em perda de sincronismo dos geradores síncronos e conseqüente desligamento do sistema. Denomina-se o ponto de operação onde isto ocorre como ponto de colapso de tensão. Este acontece, na maioria das vezes, primeiro em algumas barras do sistema (problema local) e num processo em cascata pode estender-se ao restante do sistema.

Os problemas de ET podem ser classificados de duas maneiras [KUN97]: Estabilidade de Tensão de grandes distúrbios e Estabilidade de Tensão de pequenos distúrbios. No primeiro, seguido a um grande distúrbio no sistema, a manutenção das tensões de regime está relacionada com a característica da carga do sistema e as interações entre os controles contínuos/discretos e proteções. A análise de Estabilidade de Tensão frente a grandes distúrbios abrange simulações dinâmicas no domínio do tempo que podem se estender por poucos segundos ou minutos ou mesmo horas, com a finalidade de visualizar a atuação de dispositivos tais como os LTCs (*Load Tap Changers*). No segundo, a resposta do sistema a pequenos distúrbios, como por exemplo mudanças incrementais na carga do sistema, é a questão a ser avaliada num dado instante de tempo. Esta avaliação é útil para determinar áreas com problemas em potencial. Neste caso, modelos dinâmicos linearizados são utilizados, tais como a análise modal da matriz Jacobiana.

Em Avaliação de Segurança (AS), a questão chave é o diagnóstico do estado operativo atual do sistema. Ou seja, se o ponto de operação avaliado em um dado instante é um ponto de operação seguro ou inseguro, estando em operação normal ou após a ocorrência de um distúrbio na rede. Como pode ser observado, o critério agora não é se o sistema é estável ou instável, mas sim seguro ou inseguro do ponto de vista elétrico. Muitas vezes, em relação à estabilidade, o sistema pode ser estável. Entretanto, encontra em uma região de alto risco operacional, sendo então considerado inseguro. Em [PAV85] são identificados cinco níveis operativos: normal, alerta, emergência, extremo e restaurativo. Cada região de operação determina um conjunto de informações próprias que devem ser processadas de modo a atender os requisitos de segurança. Pode ser observado que a definição dos níveis de segurança é uma condição particular de cada

sistema elétrico. Logo, objetiva-se encontrar as faixas correspondentes a cada um destes níveis. Isso implica em proceder análises de ET de pequenos sinais.

Atendo-se especificamente ao processamento de AST, isso implica em obter três informações básicas: diagnóstico do ponto de operação do sistema, identificação das áreas/barras críticas e obtenção da margem de potência em relação ao ponto de colapso, onde possibilita a definição dos níveis operativos inerentes a cada sistema elétrico. Estas informações por si só não são de muita utilidade ao operador do sistema se o mesmo não tiver as indicações de como atuar no sistema elétrico em função delas. Logo, a idéia então é que estas informações sejam manipuladas adequadamente para fornecer as melhores ações de controle para prevenir situações inseguras de operação ou retornar o sistema para a uma condição minimamente segura quando da ocorrência de alguma contingência. Exemplificando, algumas destas ações de controle seriam o corte de carga, o chaveamento de banco de capacitores, o redespacho de geradores, etc.

Nota-se então que o desenvolvimento de uma ferramenta em AST envolve a necessidade de uma análise abrangente, decorrente das constantes mudanças que ocorrem no sistema, tanto na configuração quanto no carregamento. Isso acarreta um incontável número de situações a serem simuladas, exigindo um enorme tempo computacional para a compilação das informações obtidas sobre o sistema. O problema torna-se mais crítico em um ambiente de tempo real e algumas questões devem ser observadas. A primeira preocupação é com a velocidade de resposta. Avaliação de segurança de um sistema elétrico deve ser processada em intervalos de tempo relativamente curtos. As metodologias analíticas convencionais têm evoluído bastante neste quesito [SOU97], entretanto o esforço computacional envolvido para sistemas elétricos de grande porte ainda é significativo, exigindo um considerável tempo de processamento além do permitido para este tipo de aplicação.

Outra questão diz respeito à apresentação dos resultados. Muitos índices de segurança propostos mostram-se de difícil interpretação e necessitam de decodificação apropriada. Além disso, um dado índice isolado é ineficaz, pois o operador necessita de instruções de como interpretá-lo. Ou seja, além do esforço computacional que as metodologias convencionais necessitam para processar este primeiro nível de informação, outros processamentos devem ser implementados para mostrar ao operador a direção de tomada de decisões.

Em relação aos dados que alimentam programas para AST, estes podem conter ruídos, apresentarem-se incompletos ou conterem erros de medição. Além disso, programas baseados nestes algoritmos convencionais geralmente são inflexíveis e requerem grande esforço para atualizações. Estas dificuldades têm motivado os engenheiros a buscarem novas técnicas para a

solução destes problemas. Uma alternativa que vem sendo utilizada é o desenvolvimento de ferramentas que utilizam técnicas de Inteligência Artificial. A seguir é apresentado as motivações para o uso de IA em AST.

1.2.2 Técnicas de IA aplicadas em AST

A utilização de técnicas baseadas em IA em problemas relacionados aos sistemas elétricos de potência tem aumentado muito nos últimos anos. Estas técnicas tendem, pela eficiência computacional que proporcionam, a se agregar à evolução natural dos COS. Isso é devido à forma de como é conduzido o processo de tomada de decisões nestes locais, onde são fundamentais o conhecimento e a experiência sobre o comportamento do sistema. Embora pelo fato dos métodos baseados em IA serem computacionalmente eficientes, isto por si só não justifica a sua utilização como alternativa mais eficiente aos métodos convencionais em ambiente *on-line*. Devem sim fornecer uma significativa melhora ao processo de tomada de decisão pelo operador do sistema. Ou seja, as ações de controle desenvolvidas a partir da informação disponibilizada por uma técnica de IA deve ser qualitativamente melhor que as convencionais. Este fato é a grande motivação para a aplicação de técnicas de IA em AST.

O fato de se buscar em técnicas de IA as informações qualitativamente melhores que as metodologias convencionais é devido à forma como estas técnicas manipulam a informação. Algumas técnicas proporcionam extrair relações entre as variáveis que governam o problema, uma vez que estas relações não são observadas explicitamente. Um exemplo disso em ET seria a correlação entre a tensão de uma barra que sofre a maior variação desta em função do aumento na carga (barra crítica) em relação ao aumento das perdas em algumas linhas de transmissão. Isso só é obtido com as metodologias convencionais após sucessivos estudos e uma análise criteriosa e experiente do especialista, tornando inviável para uma aplicação *on-line*, principalmente para sistemas elétricos de médio e grande porte. Entretanto, com a aplicação correta de uma técnica de IA pode-se descobrir estas relações com rapidez e facilidade. Uma das técnicas empregadas com esta finalidade é denominada de aprendizado automático, também denominado de Aquisição Automática de Conhecimento (AAC) ou *Knowledge Discovery* ou *Data Mining*.

Técnicas baseadas em aprendizado automático podem ser definidas como técnicas computacionais com capacidade de obter conhecimento, de forma automática, a respeito de soluções para certos problemas e colocá-lo sob forma interpretável aos seres humanos. Estas

técnicas são ditas indutivas, uma vez que este aprendizado é feito através de exemplares que contém as informações concisas sobre o sistema em estudo. Segundo Dy-Liacco [LIA97], aprendizado automático é aplicável a duas categorias de problemas: àqueles onde as relações entradas/saídas não podem ser modeladas analiticamente e só podem ser tratadas com técnicas de regressão e, problemas que podem ser modelados analiticamente mas requerem computação intensiva e soluções repetitivas.

A formação de um conjunto de regras de produção extraídas de uma RNA é feita através da utilização de um algoritmo baseado em aprendizado automático. As principais questões envolvendo AAC e a motivação para este trabalho são apresentadas no item a seguir.

1.2.3 Aquisição Automática de Conhecimento

Atendo-se ao problema da Aquisição Automática de Conhecimento, algumas técnicas indutivas utilizadas com esta finalidade são: aprendizado de máquina (*machine learning*), métodos estatísticos convencionais, algoritmos de extração de regras simbólicas de RNAs (modelos conexionistas), dentre outras.

No aprendizado por máquina destacam-se as técnicas baseadas em DT ([SES94], [WEI91]). O método do K- vizinhos mais próximos (K-NN - *K-Nearest Neighbors*) ([DUD73], [WEI91]) tem a preferência entre os métodos estatísticos. Utilizando-se redes neurais em processos de aprendizagem supervisionados destaca-se a RNA-ML, com treinamento baseado na retropropagação de erros [BIS95]. Quando o aprendizado é não supervisionado, as estruturas neurais do tipo Mapa de Kohonen [HAS95] são as preferidas.

Existe também a possibilidade de utilização destas técnicas de forma híbrida, por exemplo, DT/RNA e K-NN/RNA [WEI91], aproveitando-se das características positivas de cada técnica. O que se observa nestes modelos híbridos é a utilização das DTs como seletores de características e RNAs e K-NNs como classificadores.

Nas DTs, o processo de inferência no conjunto de treinamento determina uma seleção de atributos que diretamente pode gerar regras de produção, sem que nenhum processo exclusivo para isso seja implementado. Desta forma, pode-se dizer que as DTs são modelos simbólicos. Em AST, as DTs tem sido utilizadas para a estimação da margem de potência ativa ([WEH96],[LOP98]), onde a DT recebe é denominada de árvore de regressão. Em problemas de classificação, as DTs são utilizadas para avaliar se um ponto de operação do sistema elétrico é

estável ou instável [WEH93].

As RNAs apresentam-se como uma ferramenta com potencial para serem utilizadas em AST. Este potencial pode ser observado através de algumas propriedades relevantes das RNAs. Estas apresentam grande capacidade em lidar com relações altamente não-lineares, em generalizar, proceder aproximações funcionais e trabalham bem com dados que apresentem ruídos ou estejam incompletos. Além disso, após a fase de treinamento e teste, quando um caso é apresentado a estas, a resposta é imediata. Entretanto, são observados problemas tais como: RNAs são modelos ‘caixa-preta’, não simbólicos, apresentando, em geral, lentidão de treinamento, entre outras dificuldades.

As principais aplicações de RNAs em AST até o momento estão relacionadas com problemas de classificação e estimação. Atuando como classificadores em AST, a tarefa é distinguir pontos de operação seguros dos inseguros, ou classificar contingências em severas/inseguras ou não-severas/seguras. Como estimadores, as respostas aos padrões de entradas são diversas. Pode-se citar algumas como, parâmetros de estado, margem de carga, índices de estabilidade, índices de severidade, etc. Geralmente, em problemas de classificação, as RNAs com treinamento supervisionado (resposta inicialmente conhecida e comparada) são mais utilizadas por proverem resultados mais precisos que redes que fazem uso de treinamento não supervisionado (resposta desconhecida).

Uma vantagem da utilização das RNAs é o fato de serem mais precisas que as demais técnicas [WEH94]. Isto é devido à forma como o treinamento das RNAs é feito. A abordagem do treinamento neural é realizada estabelecendo as devidas correlações entre as variáveis de entrada de forma paralela. Nos métodos estatísticos convencionais e nas DTs, esse processamento é feito avaliando-se variável por variável, de forma seqüencial. Essa característica das RNAs tem estimulado seu uso como agentes de AAC. O problema a ser resolvido torna-se então como extrair conhecimento simbólico do conhecimento adquirido pelas RNAs. Alguns exemplos de algoritmos para esta finalidade podem ser vistos em [CRA96], [SES94] e [THR94]. Assim, extrair conhecimento simbólico das RNAs significa colocá-las no mesmo nível de processamento que as DTs.

Para a obtenção do conjunto de regras de produção que sintetizam o conhecimento (processamento da AAC) sobre o sistema elétrico no que diz respeito ao problema de AST, três etapas distintas são observadas. A primeira constitui a formação de uma base de dados consistente onde estão colocadas as informações pertinentes ao problema. A segunda é proceder o processamento do agente de inferência (tais como as RNA, DT, etc) nesta base de dados, de tal

forma a discriminar as principais questões chaves do problema. A terceira etapa realiza o processo de compilar as informações obtidas pelo agente e colocá-las em formato de regras de produção.

Concluindo, de acordo com o objetivo deste trabalho, a idéia então é contribuir com o desenvolvimento de uma ferramenta para AST baseada em IA, onde um conjunto de regras de produção é obtido, o qual sintetiza o conhecimento sobre o sistema elétrico em estudo. As comparações entre as regras obtidas por uma DT e as obtidas por uma RNA são feitas no sentido de buscar uma maior qualidade na informação a ser processada pela ferramenta, de tal forma que a confiabilidade dos resultados apresentados proporcionem um ganho nas ações de controle executadas pelos operadores do sistema elétrico.

1.3 O ESTADO DA ARTE DA APLICAÇÃO DE TÉCNICAS DE IA EM AS TENSÃO

O final da última década mostra um aumento na utilização de algumas técnicas de inteligência artificial aplicadas ao problema de AST. Conforme já comentado, a motivação em utilizar-se destas técnicas é devida não só a eficiência computacional em aplicações *on-line*, a capacidade para lidar bem com incertezas ou apresentar interpretabilidade, mas também por oferecer contribuições adicionais ao processo de tomada de decisões. Em AST são distinguidas basicamente duas linhas de pesquisa: a aplicação de RNAs e utilização de técnicas de aprendizado automático, embora alguns trabalhos contemplem a utilização de outras técnicas como algoritmos genéticos, lógica difusa, etc.

1.3.1 Redes Neurais Artificiais Aplicadas em Estabilidade de Tensão

Neste contexto, as RNAs têm sido muito utilizadas de forma direta, como classificadores e estimadores. Destaca-se também, concomitante com aplicações em avaliação de segurança, a utilização de metodologias que visam atacar os problemas que prejudicam o desempenho das RNAs. Em problemas de estimação, a referência [CHO91] faz uso de uma RNA para estimar as tensões e ângulos. Neste caso, é utilizada uma RNA sem camadas intermediárias (RNA-SL), a qual faz uso de treinamento baseado em mínimos quadrados. O vetor de entrada utiliza-se das potências ativa e reativa das barras de carga e tem como saída os valores de saída de tensões e

ângulos. Através dos valores obtidos destas tensões e ângulos são calculados os fluxos nas linhas. A idéia é verificar se ocorrem violações de limites de tensões e sobrecarga nas linhas. O autor apresenta também uma proposta para simular perda de linhas sem que seja preciso mudar a topologia da rede, obtendo os novos dados da configuração. Entretanto, a técnica é testada utilizando um sistema elétrico muito pequeno, o que pode levar a superestimar o potencial de utilização deste tipo de RNA-SL.

Em [HUI91], uma RNA-ML é utilizada como aproximador funcional, obtendo o valor singular mínimo como índice de Estabilidade de Tensão. Neste artigo, a rede neural monitora apenas as condições de operação da barra crítica, previamente determinada pela metodologia analítica. Tal aplicação pode falhar se ocorrer uma mudança na direção de crescimento de carga, alterando a barra crítica, sendo ainda que este índice não é muito útil ao operador. A referência [MOR91] também utiliza uma RNA-ML para estimar o módulo do autovalor mais crítico da matriz do sistema dinâmico linearizado ($\dot{\mathbf{x}} = \mathbf{A}_s \mathbf{x}$), onde \mathbf{x} é o vetor de estado e \mathbf{A}_s é a matriz do sistema, o qual é utilizado como índice de estabilidade dinâmica. Vale ressaltar que o módulo do autovalor é calculado após a matriz \mathbf{A}_s sofrer uma transformação do plano convencional imaginário (s -plano) em um plano imaginário delimitado (z -plano). O mesmo autor, em [MOR93], faz uso de uma RNA-ML para estimar dois índices de Estabilidade de Tensão, denominados de θ_x e θ_y . Em paralelo a estas estimações, uma rede neural não supervisionada do tipo Mapa de Kohonen é utilizada também para estimar os mesmos índices (de forma menos precisa). Neste caso, o enfoque é a identificação de regiões seguras e inseguras.

A referência [KEI95] faz uso de uma RNA-ML para estimar a margem de energia, cujos padrões de treinamento são obtidos através do Método da Função de Energia. Uma seleção prévia de atributos de entrada é processada baseada nas barras que mais contribuem ao colapso, obtidas através da sensibilidade da matriz Jacobiana. Tal método apresenta o inconveniente de processar duas informações distintas, ou seja, obter a função de energia e processar uma análise de sensibilidade.

A referência [MOM96] propõe, utilizando uma RNA-ML, a estimação da demanda de potência ativa e a compensação de potência reativa. A aplicação do Método do Jacobiano Reduzido proporciona aqui uma seleção prévia das barras e linhas mais relacionadas com o problema de Estabilidade de Tensão. É processada também uma análise do erro de estimação desta margem de energia quando o número de neurônios na camada interna varia, tanto para o caso base como em algumas contingências selecionadas. Uma proposta mais complexa é apresentada em [CHE93], onde as equações de fluxo de carga são transformadas num problema

de mínima energia. Esta transformação leva a obtenção de uma função de energia associada. Para a solução do problema, uma rede neural de Hopfield [HAS95] é utilizada como ferramenta de otimização. Seu modelo de treinamento convencional, cujo critério de convergência é baseado na Análise de Estabilidade de Lyapunov, é transformado em um modelo caótico para minimizar os problemas de mínimos locais que o treinamento convencional apresenta. Esta transformação possibilita que, a cada padrão de carga apresentado a esta rede neural alterada, as soluções de alta e baixa tensões sejam encontradas, construindo ao final do processo a curva PV. Embora o autor afirme que, para pontos próximos ao ‘nariz’ da curva a resposta é mais rápida, o tempo de processamento para cada padrão de carga apresentado à rede neural é alto.

Dois outros trabalhos apresentados recentemente, [NET98] e [GAR98], mostram a utilização de RNA-MLs na estimação de índices de colapso de tensão. O primeiro utiliza a margem de potência ativa obtida pelo Método da Continuação e o segundo estima um índice de Estabilidade de Tensão proposto na literatura denominado de *L-Index*.

Redes neurais que utilizam treinamento não supervisionado estão sendo também utilizadas, principalmente em problemas de classificação. A referência [NIE91] utiliza uma RNA do tipo Mapa de Kohonen (RNA-KM) para classificar os padrões de carregamento das linhas de transmissão após contingências simples e duplas. Este trabalho é também apresentado em [I3E96] e [DIL95]. Conforme os próprios autores observam, o maior problema da utilização deste tipo de rede neural é o tamanho da grade de neurônios. Um pequeno número de neurônios pode determinar perda de informação pelo grande número de padrões relacionados a pequenos grupos de neurônios da grade. Por outro lado, muitos neurônios na grade podem levar ao não estabelecimento de qualquer relação de alguns neurônios com os padrões apresentados.

Algumas pesquisas apontam na direção de estabelecer uma metodologia híbrida, onde um Sistema Especialista (ES - *Expert System*) é utilizado concomitante com as RNAs. Isto é proposto em [KUM91] e reapresentado em [I3E96], onde são analisados os requisitos para a aplicação de ESs e RNAs em avaliação de segurança dinâmica *on-line*. Neste caso, é proposto um ambiente de programação, onde cada módulo é uma das etapas de processamento da avaliação de segurança. Neste contexto, são mostradas as possíveis aplicações dos ES (seleção de contingências) e RNAs (estimação de índices e classificação de estados operativos) no problema, observando que a base de dados é obtida *off-line*, através de metodologias analíticas convencionais. Apesar de estabelecer bases em avaliação de segurança dinâmica, os autores mostram, como exemplo, as variáveis que podem ser utilizadas em AST (uma vez que a metodologia proposta também mostra a possibilidade deste tipo de análise). O mesmo autor, em

[KUM97], desenvolve a proposta anterior, mostrando as evoluções para o ambiente computacional, principalmente no que diz respeito à orientação a objeto, interface gráfica com o usuário e integração de sistemas especialistas baseados em regras. Nesta mesma linha de proposta de estrutura encontra-se [VAD91], onde são mostradas duas arquiteturas propostas para modelos híbridos do tipo ES/RNAs. Entretanto, isto é feito de forma mais superficial que em [KUM91].

Os autores em [YAN91] utilizam-se de um pré-processamento analítico para primeiramente estabelecer uma lista de contingências. Um ES utilizando-se de uma argumentação dedutiva é aplicado de modo a fornecer se uma dada contingência é segura (não viola limites) ou insegura (viola limites). Caso seja considerada insegura, o ES deve informar a razão da insegurança. No segundo estágio, uma rede neural de Hopfield [HAS95] é utilizada para classificar o estado operativo que o sistema se encontra quando da ocorrência de uma contingência. Entretanto, a arquitetura proposta neste trabalho mostra-se muito complexa, exigindo vários estágios de processamento.

Conforme já mencionado, a utilização de RNAs não se apresenta livre de problemas. Muito se tem feito para que a aplicação destas seja realizada de forma segura e confiável. Assim, muitas linhas de pesquisa procuram, além da aplicação ao problema de avaliação de segurança, utilizar técnicas que proporcionem um melhor desempenho. Isto é observado em [WEE91] que utiliza a expansão de Karhunen-Loève (K-L) [DIL95] para fazer uma redução na dimensionalidade do vetor das variáveis de entrada, melhorando a separação entre classes. Uma avaliação é feita em relação ao desempenho do redutor de dimensionalidade quando os padrões reduzidos são aplicados a uma RNA-ML para verificar estados seguros e inseguros. Baseado nos valores que foram obtidos para os índices de falso alarme e falso negativo, e após o sistema ser treinado e testado, o algoritmo permite adicionar novos atributos ao vetor de entrada com a finalidade de melhorar o desempenho deste classificador. Esta metodologia é comparada com a técnica de separação pela média da classe (C-M-S, *Class-Mean-Separation*), mostrando como resultado, o melhor desempenho da RNA-ML quando treinada com a K-L. Em [DIL95] uma metodologia é proposta combinando as duas técnicas para melhorar a seleção de características.

Em [ZAY96], a K-L é comparada com o método da mínima entropia para a seleção de características. O desempenho dos métodos é testado utilizando-se uma RNA-ML para classificação de padrões de operação seguros e inseguros. Mostra-se aí que o método da mínima entropia é mais eficiente. Um grande problema apresentado pelos métodos acima citados é que a redução da dimensionalidade do vetor de entrada se faz aplicando-se transformações lineares,

onde o sentido físico das variáveis é perdido. Logo, tais métodos são eficazes onde apenas a resposta final ao problema é a questão central, sem a necessidade de conhecer as características envolvidas no processo de inferência.

Outro problema abordado diz respeito à redução da cardinalidade do conjunto de treinamento. [SHA93] propõe utilizar uma RNA-KM para classificar os estados seguros e inseguros de um sistema elétrico. O objetivo é reduzir o esforço computacional *off-line*, mostrando que, com um reduzido número de padrões de treinamento é possível obter a classificação adequada. Os autores mostram também que o modelo é hábil em detectar diferenças dentro de uma mesma classe. Entretanto, não são apresentadas as taxas de erro de classificação obtidas. Um dos autores do trabalho anterior apresenta em [SHA93a] uma técnica baseada na inversão da RNA-ML e a utilização do treinamento *query-based* para diminuir a necessidade de muitos padrões de treinamento. É mostrado que, com o modo tradicional de utilização da RNA-ML, com 1500 padrões de treinamento, o erro na definição da superfície de separação entre as regiões segura e insegura é maior do que com 500 padrões quando a metodologia proposta é aplicada. Ambos os trabalhos, [SHA93] e [SHA93a], são novamente apresentados em [DIL95].

Uma metodologia baseada em Algoritmos Genéticos (GA - *Genetic algorithms* [HAS95]) é proposta em [SHA94] para otimizar o treinamento de uma RNA-ML. Neste caso, os pesos das conexões são inicializados utilizando-se uma metodologia de GA, promovendo um substancial ganho em velocidade de treinamento, além de evitar mínimos locais. A aplicação em AST destina-se a distinguir padrões seguros dos inseguros e é utilizada também a redução da dimensionalidade do vetor de entrada pela técnica combinada de C-M-S e K-L. Estes autores utilizam a mesma técnica por GA em [SHA96a], aplicada agora em uma RNA-KM. A técnica é direcionada a problemas de segurança cujo índice é o tempo de remoção de curto-circuito no sistema.

As referências [COR94] e [SCA95] aplicam uma mesma técnica baseada na minimização da função erro, através da pseudoinversa da matriz de saída dos neurônios da camada interna. Tal técnica é utilizada em substituição a retropropagação do erro para melhorar a velocidade de treinamento. Este procedimento visa estimar a margem de potência ativa e margem de tensão (calculadas em relação à potência ativa e tensão no ponto de colapso, respectivamente) em ambos os trabalhos. Para isto, fazem uso do sistema teste Nova Inglaterra de 39 barras e o sistema teste do IEEE de 118 barras, respectivamente. São utilizadas todas as potências ativa e reativa e tensões das barras de carga, não havendo preocupação com redução da dimensionalidade dos respectivos vetores. Pode ser observado que a combinação da metodologia de treinamento com a dimensão

elevada do vetor de entrada leva a obtenção de um número elevado de neurônios na camada interna para obtenção de baixos erros de estimação. Os autores mostram que o tempo de treinamento utilizando esta técnica é muito baixo quando comparado ao da técnica de retropropagação do erro. Além disso, os autores mostram que, utilizando-se a retropropagação de erros, obtém-se maiores erros médios.

Uma proposta para avaliar o desempenho de RNAs como classificadores é apresentada em [YAN94]. A metodologia propõe a utilização de índices baseados nas probabilidades de ocorrência de falsos alarmes e não detecção. Um projeto desenvolvido pela empresa B. C. HYDRO (Canadá) [SHA96] visa explorar a aplicação de RNAs em avaliação de segurança dinâmica. Apesar de não apresentar resultados para segurança de tensão, algumas idéias são também aplicáveis a este problema, como a que mostra que a partição do sistema elétrico em subsistemas apresenta melhores resultados que a observação do sistema completo, ocasionando uma redução natural nos vetores de entrada. A escolha correta das variáveis para formular o problema também é um fator que melhora a precisão dos resultados. Finalizando esta parte de aplicação direta de RNAs ao problema de AST, dois trabalhos, [DIL95] e [I3E96], apesar de não serem recentes, ainda se destacam como representantes do estágio atual das pesquisas na área, apresentando estudos envolvendo algumas das principais técnicas aplicadas ao problema de AST.

1.3.2 Aprendizado Automático Aplicado em AST

A linha de pesquisa utilizando aprendizado automático no problema de AST mostra-se menos exploradas que a baseada em RNAs, com poucos trabalhos direcionados ao problema. Um grupo de pesquisa estabelecido na Universidade de Liège, Bélgica, tem apresentado alguns trabalhos mostrando a utilização de DTs em AST.

Em um destes trabalhos proposto por Wehenkel [WEH93], uma revisão teórica mais detalha das DTs é realizada, onde são abordados os aspectos referentes à construção destas como: divisão ótima de um nó, critério de parada, avaliação da qualidade da DT, dentre outros. Utilizando DTs em AST, a questão chave é classificar os estados de operação pré e pós-contingências em seguros e inseguros. Neste trabalho, são apresentadas duas DTs mostrando os atributos que classificam os estados em seguros e inseguros. A primeira DT mostra um processo de classificação onde nenhuma variável é descartada. A segunda árvore é obtida após ser descartada a variável com maior *score*. O resultado mostra que o mascaramento desta variável

produz queda no desempenho da DT. Entretanto, como esta queda não é muito significativa, os autores assinalam com a possibilidade de explorar este aspecto, mostrando as relações entre os vários parâmetros, ou seja, construindo-se várias DTs, cada uma relacionada com alguma particularidade. Outra questão mostra que o aumento do número de classes a serem consideradas, geralmente produz um aumento do erro de classificação. Embora o artigo apresente as vantagens da utilização das DTs, realçando os aspectos da interpretação dos resultados, não é mostrado como as variáveis selecionadas, classificando um dado estado operativo, estão relacionadas ao problema da Estabilidade de Tensão.

Em [WEH94], é apresentado um estudo comparativo entre metodologias de aprendizado de máquina, redes neurais e reconhecimento estatístico de padrões. Embora tenham sido feitas comparações entre várias metodologias, foram efetivamente comparados os resultados da aplicação das DTs, das RNA-ML e K-NN, e modelos híbridos DT/RNA e DT/K-NN. Retorna-se aqui ao problema de classificação de estados seguros e inseguros presentes numa base de dados gerada aleatoriamente. A interpretação dada pelos autores aponta que, apesar de mais precisas, as RNA-ML enfrentam o problema do modelo caixa-preta e da lentidão do treinamento. Pelos resultados obtidos para as RNA-MLs, a redução da dimensionalidade do vetor de entrada não é favorável à melhora do desempenho desta, aumentando o erro de teste. O melhor resultado obtido entre os métodos comparados é encontrado quando uma RNA-ML é utilizada com o vetor de entrada completo. O K-NN apresenta um comportamento irregular. A alta dimensionalidade do vetor de entrada geralmente impõe degradação no desempenho deste método, mas ocasionalmente pode apresentar bons resultados. As DTs mostram-se menos precisas, porém seus resultados são interpretáveis e sua velocidade de treinamento e teste muito alta. O método híbrido DT/RNA apresenta desempenho ruim pois, neste caso, a DT promove a seleção de variáveis, reduzindo a dimensionalidade do vetor de entrada. Como já observado pelos autores, isto proporcionou a piora do desempenho do classificador baseado em RNA. Para o modelo DT/K-NN, os resultados mostram uma melhora no desempenho quando a DT fornece as variáveis selecionadas. Um modelo híbrido mais promissor foi detectado quando uma DT faz a seleção de variáveis e uma RNA-ML estima a margem de potência. Os autores concluem que as DTs são efetivas na análise de grandes bases de dados para informação de segurança, as RNAs lidam bem com problemas de regressão, como estimação de margem de segurança e índices, e o método K-NN necessita de aprimoramento para ser usado sistematicamente.

A idéia de utilizar uma DT em problemas de estimação, denominada de Árvore de Regressão (RT - *Regression Tree*) é mostrada por [WEH96]. Neste trabalho, uma RT não

paramétrica é utilizada como uma DT, só que a margem de potência para cada contingência em relação à margem do caso base é estimada baseada na média da classe. Uma vez obtida as variáveis que estimam as margens, uma rede neural fornece o refinamento para melhorar a estimação. O autor afirma que as variáveis selecionadas pela RT são as mais fortemente relacionadas com a severidade de uma contingência. Muitas das conclusões observadas e interpretadas pelos trabalhos citados são abordadas num *Tutorial Course* [WEH97] a respeito dos aspectos da utilização de aprendizado automático em problemas de avaliação de segurança. Além das metodologias, são observadas também as questões referentes à geração da base de dados, da seleção dos atributos primários envolvidos em cada problema e o aspecto promissor da tradução dos resultados fornecidos pelas DTs em ações de controle. A tendência de aplicação de técnicas de aprendizado automático em AST é notada em outros trabalhos. Mais recentemente, [LOP98] apresenta uma metodologia denominada de *Kernel RTs*, onde uma função de regressão *Kernel* é utilizada nas ‘folhas’ da RT para estimar a margem de potência, mostrando erros muito baixos de estimação.

O que pode ser observado nesta breve retrospectiva bibliográfica é que as RNAs são uma ferramenta precisa em problemas de classificação/regressão, a despeito de seus problemas inerentes, como a dimensionalidade do vetor de entrada e a característica de caixa-preta. Muitos trabalhos têm procurado atacar estes aspectos, no sentido de melhorar o desempenho das RNAs, seja em problemas de classificação ou de regressão. Alguns grupos de pesquisa vêm trabalhando em outra direção, principalmente com DTs e RTs como ferramenta principal, uma vez que a interpretabilidade destas metodologias é um estímulo ao processo de aquisição de conhecimento. Esta questão do modelo caixa-preta (interpretação) de uma rede neural tem sido o principal obstáculo à utilização plena destas em aquisição automática de conhecimento. Entretanto, para atacar este problema existem algoritmos suficientemente robustos para extrair conhecimento simbólico de uma RNA. Um destes algoritmos, BRAINNE ([SES94]), foi recentemente utilizado ([ABR98]) em problemas de AST, apresentando não só um desempenho muito bom em termos de classificação, mas também proporcionando uma boa interpretabilidade física das variáveis selecionadas no processo de aquisição de conhecimento. Entretanto, este algoritmo mostra-se muito lento, uma vez que utiliza repetidos treinamentos de uma associação entre RNA-ML e RNA-SL, sendo inviável sua utilização para sistemas de grande porte. A tendência na continuidade das pesquisas é utilizar as variáveis que caracterizam uma dada classe como parâmetros de ações de controle a serem implementadas.

1.4 ETAPAS DO TRABALHO

O processo de construção de uma ferramenta de AST *on-line* baseada em técnicas de IA é um trabalho multitarefa, onde cada etapa apresenta suas próprias características. Neste processo, três fases podem ser propostas. A etapa de construção da base de dados corresponde na definição da metodologia de análise de Estabilidade de Tensão, em questões como formação desta através de simulações *on-line* ou *off-line*, no volume de dados necessário, tempo de simulação, dentre outras. Vale salientar que a base de dados deve ser consistente para conter as informações relevantes do processo.

Na segunda etapa, as RNAs são utilizadas no problema de estimação da margem de potência ativa do sistema, considerando os problemas inerentes ao uso desta ferramenta. A estimação da margem de potência ativa possibilita indicar quais as contingências mais críticas a partir do ponto de operação corrente. Esta estimação da margem de potência ativa é obtida através da utilização de uma RNA denominada de *Optimal Estimate Training 2* (OET2).

Finalizando, na terceira etapa, que é a utilização de técnicas de aprendizado automático para a classificação dos níveis de segurança do sistema, as informações relevantes do comportamento do sistema são extraídas. Toda a informação, ao final do processamento, deve ser colocada em formato de regras de produção. Isto abre a ‘caixa-preta’ que muitas pesquisas apontam como a principal restrição à aplicação das RNAs em AAC. Além do processo de classificação

A Figura 1.1 apresenta o esquema utilizado neste trabalho para gerar o conjunto de regras de produção associado com o problema de AST. Neste esquema, três módulos básicos são identificados: Formação da Base de Dados, Aquisição Automática de Conhecimento e a utilização de uma RNA para um problema de regressão. A idéia neste trabalho é desenvolver estes módulos básicos de forma independente, de modo a se constituírem em ferramentas individuais que possam ser utilizadas para outras finalidades. Assim, o desenvolvimento de cada módulo apresenta suas próprias metas a serem alcançadas. A seguir são mostrados os passos implementados em cada módulo.

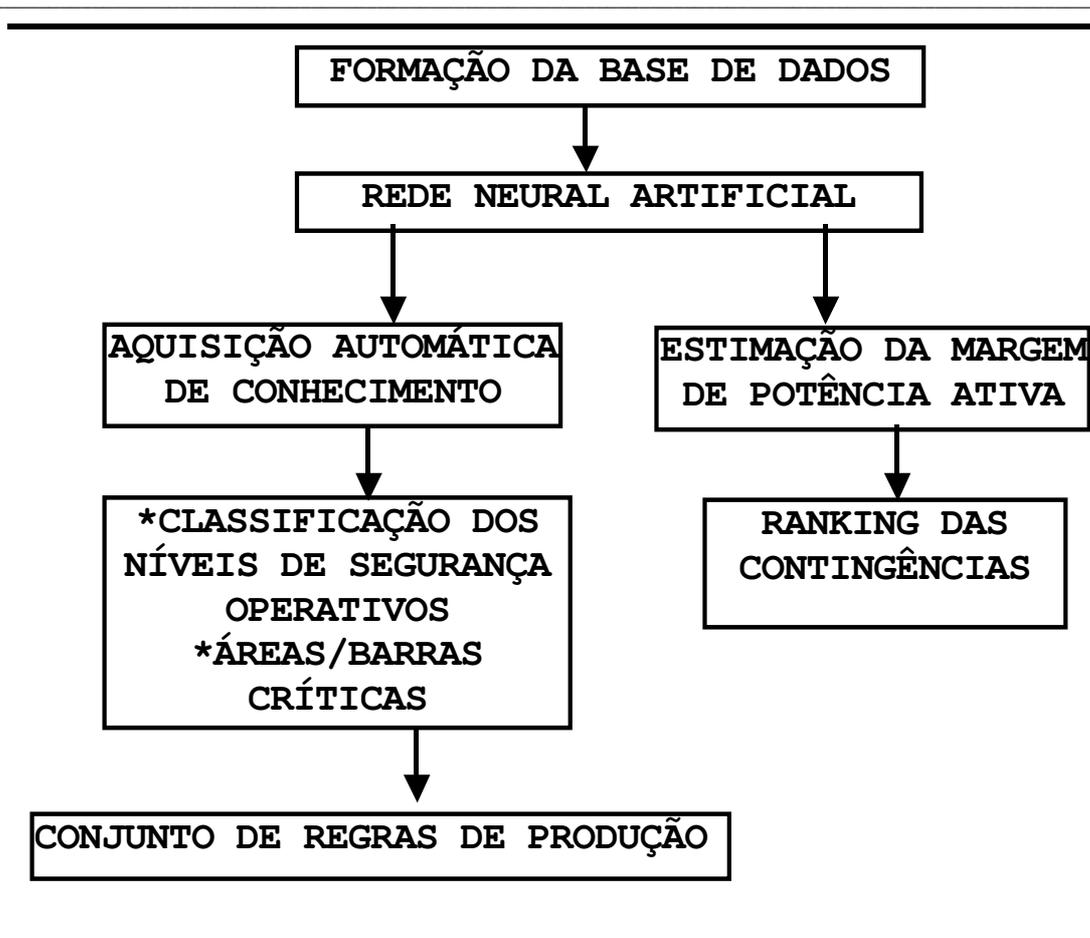


Figura 1.1 – Esquema das etapas do trabalho

- A. FORMAÇÃO DA BASE DE DADOS:** Obtenção da base de dados utilizando o Método da Continuação como a ferramenta de análise de estabilidade de tensão aplicada aos casos bases e as contingências prováveis. Dois módulos de desenvolvimento são processados:
- A. 1** Dotar o programa de fluxo de carga de modelagens de equipamentos, carga e controles que se fizerem necessários para desenvolver simulações mais realistas.
- A. 2** Desenvolver uma ferramenta computacional para selecionar as contingências e acionar o Método da Continuação para cada caso. Esta ferramenta deve também proporcionar a formatação da base de dados de acordo com as variáveis escolhidas.
- B. REDE NEURAL ARTIFICIAL:** Estimar a margem de potência ativa utilizando uma RNA OET2. Dentro do objetivo principal do trabalho, esta etapa destina-se apenas a mostrar como seria a aplicação da RNA OET2 dentro do aplicativo, além de mostrar sua precisão para problemas de regressão. Duas etapas são previstas utilizando o sistema teste IEEE 14 barras:
- B. 1** Treinamento e teste do OET2.
- B. 2** Verificação do erro de estimação para os casos simulados.
-

C. AQUISIÇÃO AUTOMÁTICA DE CONHECIMENTO: Obter a devida classificação dos níveis de segurança em função da margem de potência ativa. Desenvolvimento em dois módulos:

C.1 Implementação e aplicação de um algoritmo de extração de regras de produção baseado em RNAs, denominado *Validity Interval Analysis* (VIA) como módulo de aquisição automática de conhecimento para os sistemas testes IEEE 14 barras e 118 barras.

C.2 Verificação dos índices estatísticos das regras obtidas para cada caso simulado para os referidos sistemas elétricos, comparando com os resultados obtidos com uma Árvore de Decisão do tipo C4.5.

A apresentação deste trabalho de tese é feita em cinco capítulos e dois apêndices, como apresentada a seguir:

Capítulo 1 - ‘Introdução e Objetivos’: Apresenta a introdução geral ao trabalho bem como o objetivo deste.

Capítulo 2 - ‘Avaliação de Segurança de Tensão’: São apresentados os conceitos na área de análise de estabilidade de tensão e de avaliação de segurança de tensão. Mostra também o desenvolvimento do método da continuação e sua justificativa como formador da base de dados. Apresenta também o método do vetor tangente como a metodologia para indicar as barras críticas do sistema.

Capítulo 3 - ‘Aquisição Automática de Conhecimento’: Os conceitos básicos relativos a alguns problemas em reconhecimento de padrões, como a ‘maldição da dimensionalidade’ e técnicas de inferência em uma base de dados são explorados. É também apresentada a teoria do Aprendizado por Máquina, especificamente das Árvores de Decisão e suas aplicações em AST. O conceito básico de como extrair conhecimento simbólico é mostrado, bem como a teoria de alguns algoritmos de extração de regras de RNAs e o algoritmo VIA, implementado neste trabalho.

Capítulo 4 - ‘Simulações computacionais e resultados’: Neste, são apresentados os aspectos relativos às implementações feitas para cada aplicativo desenvolvido e os resultados das simulações para os sistemas elétricos em estudo.

Capítulo 5 - ‘Conclusões e propostas’: São apresentadas as conclusões do trabalho e as propostas para trabalhos futuros.

-
- Apêndice A** - ‘Estabilidade de Tensão’: Apresenta a teoria das principais metodologias aplicadas para a análise de estabilidade de tensão.
- Apêndice B** - ‘Técnicas de Reconhecimento de Padrões’: Mostra algumas técnicas básicas relativas ao processo de aprendizado indutivo.
- Apêndice C** - ‘Redes Neurais Artificiais’: Apresenta os algoritmos das RNAs Hebb, RNA-ML e OET2.
- Apêndice D** - ‘BRAINNE’: Apresenta o desenvolvimento teórico do algoritmo BRAINNE para extração de regras de produção de redes neurais artificiais.
- Apêndice E** - ‘Regras de produção C4.5’: Mostra as regras de produção obtidas pelo C4.5 para os sistemas testes IEEE 14 barras e 118 barras.
- Apêndice F** - ‘Regras de Produção VIA’: Mostra as regras de produção obtidas pelo VIA para os sistemas testes IEEE 14 barras e 118 barras.
- Apêndice G** - ‘Sistemas Elétricos – Dados de entrada’: Apresenta os dados relativos aos sistemas testes IEEE utilizados no âmbito deste trabalho (14 barras e 118 barras).
-

Capítulo 2

AVALIAÇÃO DE SEGURANÇA DE TENSÃO

2.1 INTRODUÇÃO

Neste capítulo são apresentadas as principais definições relativas ao problema de Avaliação de Segurança de Tensão. O desenvolvimento de uma ferramenta utilizando-se de técnicas de inteligência artificial envolve trabalhar com uma ferramenta que forneça uma base de dados fidedigna do fenômeno de estabilidade de tensão. O Método da Continuação preenche estes requisitos e é apresentado aqui como o dispositivo gerador da base de dados construída *off-line*. É também mostrado o Método do Vetor Tangente como a metodologia utilizada para a obtenção das barras críticas do sistema.

2.2 AVALIAÇÃO DE SEGURANÇA

A AS de um sistema elétrico de potência torna-se prioritária a cada dia devido ao aumento da complexidade dos problemas operativos. Isto se deve a combinação de fatores como o maior número de áreas interconectadas nos sistemas elétricos, as múltiplas escalas de tempo dos dispositivos de controle, o comportamento da carga elétrica, a escassez de investimentos, etc.

Para garantir a operação de um sistema elétrico de forma otimizada, deve-se antever os problemas os quais o sistema pode ser submetido. Logo, os centros de controle da operação podem ser dotados de ferramentas que apresentem ao operador quais os prováveis cenários no qual o sistema pode evoluir. Além disso, faz-se necessário indicar as melhores opções de ações

de prevenção e correção. Tais ferramentas são conhecidas na literatura como *ferramentas de apoio à tomada de decisões*. Pode-se concluir que a contribuição relevante destes dispositivos de análise é proporcionar ganhos de confiabilidade nas ações implementadas pela operação do sistema.

Para o desenvolvimento de ferramentas de AS em sistemas elétricos de potência é necessário entender os fenômenos que estão associados a estas. A definição clássica de AS é dita ser a verificação da capacidade do sistema em manter-se em operação estável após a ocorrência de distúrbios. Se houver necessidade de implementar ações de controle para retornar o sistema à condição estável, estas devem ser também avaliadas. As referências [PAV85], [MOR99] e [KUN97] utilizam o termo ‘Avaliação de Segurança Dinâmica (DSA, *Dynamic Security Assessment*)’ para se referirem ao conjunto de estudos dos fenômenos de estabilidade que ocorrem em um sistema elétrico. Estes estudos podem ser feitos por programas computacionais estanques ou inseridos num aplicativo maior, com objetivo de não só avaliar como também indicar as ações necessárias para garantir uma operação segura. Neste trabalho, o enfoque específico é a AST.

Vale salientar que o objetivo deste trabalho, como já observado no Capítulo 1, não é o desenvolvimento de uma ferramenta para AST e sim realizar estudos comparativos de técnicas que porventura possam se agregar a estes aplicativos. Os itens a seguir visam mostrar quais são bases necessárias para o desenvolvimento de um aplicativo com esta finalidade. Além disso, a uma vez que o objetivo do trabalho é comparar técnicas de aprendizado automático baseadas em técnicas indutivas, faz-se então necessário constituir uma base de dados. A idéia é utilizar uma metodologia convencional em ET para atuar como o mecanismo de geração de dados, seguindo uma estratégia apropriada para estudos em AS.

2.2.1 Requisitos gerais para análise de AS

De acordo com [MOR99], um aplicativo direcionado à DSA deve contemplar a análise de três funções críticas como requisito. Estas são:

- *Análise do planejamento da operação*: Visa avaliar as estratégias de operação processadas para um dia, uma semana ou meses a seguir. Isso é feito para assegurar que as mudanças na configuração (manutenções programadas) possam ser avaliadas e devidamente instruídas.

- *Análise da operação próxima*: São análises processadas para o mesmo dia ou para o dia seguinte. São necessárias para assegurar que certas condições de operação diária não contempladas sejam realizadas.
- *Análise de segurança on-line*: Realiza a avaliação de segurança para o ponto atual de operação. Deve mostrar também quais são as ações da operação e as prováveis mudanças no sistema ainda não contempladas que podem ser executadas. Geralmente é feito em tempo real (informações diretamente do sistema via Estimador de Estados) ou em tempo real dito estendido (caso base montado para a finalidade de avaliação das mudanças a serem implementadas).

Esses são os requisitos gerais desejáveis para uma ferramenta em AS. Entretanto, cada componente desta tem suas necessidades específicas de análise. No que diz respeito a AST, enfocada neste trabalho, três estudos devem ser executados como condição primária para a análise. Estes seriam a seleção de contingências, a análise de segurança propriamente dita e as ações preventivas e corretivas a serem implementadas. Cada um destes requisitos apresenta também características próprias de aplicação.

O processo de seleção de contingências é impraticável para sistemas elétricos de grande e médio porte, caso se queira avaliar todas as concebíveis. Geralmente, a análise recai sobre as mais prováveis, o que reduz bastante o espaço de busca destas. Entretanto, continua um número elevado a ser avaliado e, portanto, deve ser definida uma ferramenta de análise de contingências que indique o mais rapidamente possível o grau de criticidade de cada uma processada.

A análise de segurança visa definir os limites das tensões nas faixas operativas, os limites que os equipamentos de potência reativa devem permanecer, a margem de carga de cada ponto de operação e a margem de segurança para um aumento brusco de carga, etc.

As ações preventivas e corretivas devem ser definidas sempre que houver contingências, para o caso base em estudo, que não apresentem margem suficiente para manter o sistema seguro após a ocorrência destas.

2.2.2 Estados operativos

De acordo com o apresentado no item 2.2.1 para a função de *análise de segurança on-line*, o ponto atual de operação do sistema deve ser avaliado. Isto implica em indicar o tipo de estado operativo (faixa de operação) e a margem de carga deste ponto. Conforme proposto em [PAV85], podem ser identificados cinco estados operativos em sistemas elétricos. Considerando

(I) as restrições de igualdade, representadas pelo balanço entre geração/consumo do sistema, e (D) as restrições de desigualdade, representadas pelas limitações físicas como limites de transmissão de potência, limites de tensão, etc., os estados operativos propostos neste artigo são:

- *Normal*: Todas as restrições (I) e (D) são satisfeitas e não há perigo de serem violadas.
- *Alerta*: Todas as restrições (I) são satisfeitas. Entretanto, a probabilidade de alguma restrição (D) ser violada é alta. A questão é se o sistema resiste a qualquer distúrbio plausível. Se este não for o caso, o controle preventivo deve ser capaz de retornar o sistema para a condição normal. O objetivo final deste estado é o balanço entre a redução de probabilidade de perda de integridade com o custo econômico da operação [WEH97].
- *Emergência*: As restrições (I) ainda estão todas satisfeitas mas alguma restrição (D) é violada. O controle de emergência deve (rapidamente) ser capaz de restaurar o sistema no mínimo no estado de alerta. Esta é uma evolução mais determinística, onde a resposta no tempo é crítica e as considerações econômicas tornam-se temporariamente secundárias [WEH97].
- *Extremo*: Ambas as restrições (I) e (D) são violadas. Quando os controles preventivo e de emergência falham, os dispositivos automáticos de proteção atuam de modo a evitar que os componentes do sistema de potência operem sob condições inaceitáveis, ocasionando danos irreparáveis. Estas ações podem levar a novos distúrbios que resultam em blecautes parciais ou completos.
- *Restaurativo*: É o estado de transição que se segue às ações de emergência. Ações são implementadas para restaurar o sistema para o estado normal ou no mínimo alerta. Neste estado, a tarefa do operador é minimizar o montante de energia não suprida, sincronizando os geradores perdidos tão logo quanto possível.

Os estágios operativos normal, alerta e emergência enfocam o sistema elétrico quando o mesmo se encontra intacto. Logo, a previsibilidade dos eventos é a tônica nestes estágios. Para isso, é necessário avaliar as contingências que o sistema pode ser submetido. Além disso, determinar quais as mais críticas e avaliar se, no corrente ponto de operação, a ocorrência de alguma contingência pode levar o sistema a blecautes. Ou seja, primeiramente deve ocorrer um processo de seleção de contingências (*contingency screening*) e a devida classificação por criticidade destas (*ranking*). Se alguma contingência apresentar margem insuficiente, podem ser tomadas ações preventivas do tipo redespacho de geradores, chaveamento de capacitores ou reatores, possibilitando trazer o sistema para a margem de segurança adequada. Todo esse

processo garante o aumento da confiabilidade da operação elétrica. Para isso ser verdade, é desejável que as ferramentas empregadas com esta finalidade indiquem as ações que apresentem a melhor relação custo/benefício. Desta forma, é importante a definição das técnicas envolvidas neste processo para garantir que qualidade da informação proporcione ganhos na operação do sistema elétrico.

Nos estados extremo e restaurativo, as ações deferidas são realizadas após a ocorrência de alguma contingência no sistema. Geralmente, as ações de controle aqui empreendidas têm o caráter corretivo, tais como corte de carga, bloqueio de tape de transformadores, etc. Para determinar quais são os controles mais efetivos são utilizadas técnicas convencionais de análise de sensibilidade combinadas com métodos de otimização.

2.2.3 Estabilidade em Sistemas de Potência

De acordo com as definições propostas para os estados operativos, nota-se que, além do monitoramento constante do valor do carregamento do sistema para atender as restrições de igualdade, é necessário encontrar os limites para definir as restrições de desigualdade. Estes limites são definidos nos estudos dos fenômenos de estabilidade que ocorrem no sistema. A evolução dos sistemas elétricos tem resultado em modos mais complexos de estabilidade, requerendo considerações mais claras de vários aspectos da estabilidade do sistema. A referência [KUN97] apresenta uma proposta de definições e classificação dos problemas atuais de estabilidade em sistemas de potência, sintetizadas na Figura 2.1.

A seguir são apresentadas algumas definições que auxiliam no processo de construção do conhecimento em estabilidade de sistemas elétricos. Os distúrbios que podem levar a instabilidade de um sistema elétrico podem ser definidos como externos (faltas subsequentes a descargas atmosféricas) ou internos (falhas em equipamentos) e, lentos (comportamento aleatório do padrão de demanda) ou rápidos (saída de geradores ou linhas de transmissão).

De uma maneira geral, a estabilidade de sistemas de potência é a propriedade que os habilita a permanecer em um estado de equilíbrio aceitável de operação sob condições normais ou após um distúrbio. Como a geração de potência é feita majoritariamente por máquinas síncronas, a condição necessária é que estas permaneçam em sincronismo. Isto está relacionado com a dinâmica do ângulo do rotor do gerador e a relação potência-ângulo.

Em algumas situações pode acontecer instabilidade sem perda de sincronismo; por exemplo, um sistema composto de um gerador suprindo um motor de indução pode tornar-se

instável por causa do colapso da tensão na carga. Outro evento é a ocorrência de um desbalanço significativo entre carga e geração. Neste caso, gerador e controle de velocidade tornam-se importantes, bem como os controles do sistema e proteções especiais. Caso não ocorra uma coordenação apropriada destes dispositivos, a frequência do sistema leva-o à instabilidade e geradores e/ou cargas podem ser retirados intempestivamente, levando o sistema a blecautes.

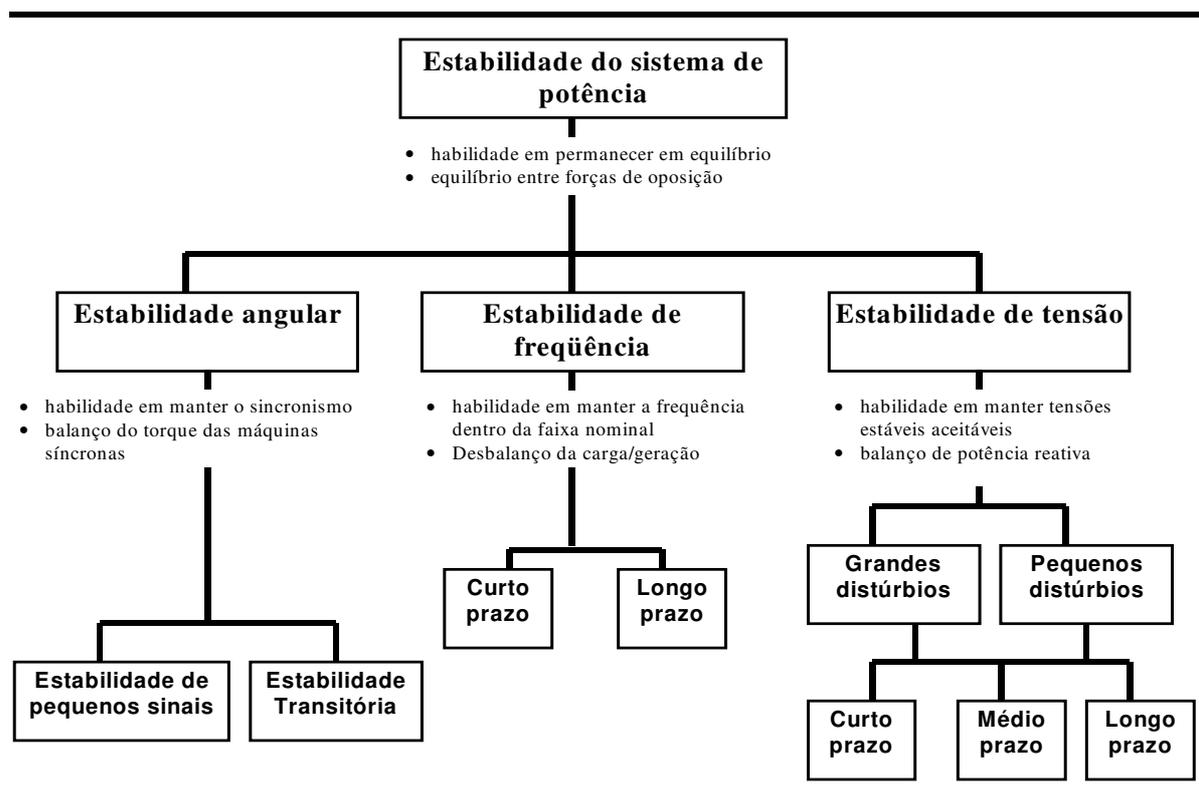


Figura 2.1 - Tipos de estabilidade em sistemas de potência.

A definição para estabilidade de tensão proposta por [KUN97] é definida como a habilidade do sistema de potência em manter as tensões de regime permanente aceitáveis em todas as barras do sistema tanto como sob condições normais de operação como após a um distúrbio. O principal fator que causa a instabilidade de tensão é a incapacidade de um sistema de potência em manter um apropriado balanço de potência reativa por todo o sistema. Pode se dividida em duas classes: grandes e pequenos distúrbios (capacidade em manter a estabilidade de tensão após um grande distúrbio ou a um pequeno distúrbio, respectivamente).

Esta mesma referência propõe que existe diferença entre os fenômenos de instabilidade de tensão e colapso de tensão. Assim, um sistema é instável por tensão se, no mínimo em uma barra do sistema, a magnitude da tensão desta barra diminui enquanto a injeção de potência reativa nesta mesma barra aumenta. O colapso de tensão é definido então como uma seqüência

de eventos que acompanham a instabilidade de tensão, levando a um baixo e inaceitável perfil de tensão em uma parte significativa do sistema.

2.2.4 Ferramentas para AST

Decorrentes do fenômeno da ET, uma ferramenta para AST deve então ser capaz de fornecer um conjunto mínimo de informações qualitativas. Conforme já observado, estas informações constituem a base para definir as melhores ações de controle a serem implementadas pela operação do sistema. Estas são:

- *Diagnóstico do ponto de operação do sistema:* Isto implica em determinar, inicialmente, se o ponto de operação é estável sob o ponto de vista de tensão. Esta análise pode ser determinada por algumas ferramentas conhecidas da literatura, como determinação dos autovalores da matriz Jacobiana, por exemplo.
- *Determinação da área/barra crítica do sistema:* Visto que o problema é reconhecido como um problema local, a determinação da área/barra crítica permite que sejam tomadas ações corretivas efetivas para a prevenção do fenômeno.
- *Margem de carga:* Diferença de carregamento entre o ponto de operação conhecido e o ponto de colapso de tensão: Isto implica no conhecimento de um método que identifique o ponto de colapso de tensão.

Para auxiliar na justificativa da necessidade do desenvolvimento de uma ferramenta para AST, é apresentado o exemplo a seguir, mostrado na referência [GCOI97]. Em 25/04/97 observou-se que o Sistema Elétrico Interligado das regiões Sul/Sudeste/Centro-Oeste do Brasil operava com um nível de carga típico para o período. Devido ao fato de algumas fontes importantes de potência reativa estarem fora de operação, como compensadores síncronos e bancos de capacitores nas áreas São Paulo e Minas Gerais, as tensões de alguns barramentos destas áreas encontravam-se abaixo da faixa operativa, próximo as 18:00 horas. Com o aumento de carregamento do sistema, os compensadores síncronos da CESP (Companhia Energética de São Paulo) atingiram seus limites de geração, reduzindo ainda mais os recursos de controle de tensão. Observou-se que a carga continuava a crescer e novas medidas, como corte de carga e remanejamento de geração, são tomadas e não surtiram o efeito esperado. Esta cascata de eventos levou então o sistema a operar em condições de subfrequência e subtensão, o que fez com que novos cortes de carga fossem implementados. Somente as 19:27 horas as cargas foram novamente liberadas, e o sistema voltou a operar normalmente. Destes fatos, duas importantes

características podem ser notadas:

- Desconhecimento dos operadores a respeito do fenômeno. Isto se refletiu nas decisões de controle adotadas, que não contemplavam problemas de colapso de tensão especificamente.
- O prejuízo financeiro com as ocorrências mostra que são necessárias medidas efetivas de controle em relação ao problema de colapso de tensão, já que, cerca de 15% da carga total de 40000MW deixou de ser suprida durante a ocorrência.

Deste exemplo de ocorrência, justifica-se a implementação de uma ferramenta de auxílio à tomada de decisões na área de estabilidade de tensão. A indicação prévia da direção em que o sistema caminha, em função dos equipamentos não operantes, pode proporcionar que ações de controle sejam adequadamente tomadas, evitando maiores prejuízos.

Uma vez definidos os requisitos mínimos a serem atendidos por uma ferramenta para AST, pode-se desenvolver, a partir de metodologias empregadas para análise de ET, uma ferramenta que proporcione que as informações sobre o comportamento do sistema elétrico em estudo sejam sintetizadas em uma base de dados. A idéia neste trabalho é que esta ferramenta funcione como um módulo individual que, não só gere a base de dados mas que também seja uma ferramenta convencional de análise de ET. Desta forma, esta ferramenta deve atender alguns requisitos de análise, os quais são apresentados a seguir.

Um fato observado em técnicas indutivas de aprendizado é que, quanto maior a qualidade das informações contidas na base de dados, melhor é o resultado proporcionado por estas técnicas. Logo, deve-se então utilizar para a geração de dados métodos de avaliação de ET que sejam bastante informativos sobre o comportamento do sistema. Nesta direção, a escolha, neste trabalho, recai no MC [I3E93a]. Além disso, o Método do Vetor Tangente, proposto em [SOU96a, SOU96b, SOU97] auxilie na obtenção das informações necessárias para atender os requisitos de uma ferramenta em AST. Ambos os métodos são também descritos no item a seguir.

2.3 FERRAMENTAS CONVENCIONAIS EM ET UTILIZADAS PARA A FORMAÇÃO DE UMA BASE DE DADOS

As características da metodologia proposta para AAC, a serem mostradas nos Capítulos 03 e 04, são baseadas em aprendizado por exemplos. Isto determina a formação de uma base de dados em que devem estar contidas as informações relevantes para a solução do problema.

Assim, a base de dados com as informações do processo de avaliação de estabilidade de tensão deve conter o comportamento do sistema em um ambiente de crescimento de carga. Isto pode ser representado pelos inúmeros pontos de operação que constitui a trajetória do ponto inicial até o ponto de colapso.

A formação desta base de dados constitui um ponto de controvérsia entre processamentos *off-line* e *on-line*. A tarefa da construção de uma base de dados (ou conjunto de treinamento) *off-line* é tediosa, uma vez que há um enorme número de combinações de variáveis e topologias além de um grande esforço computacional requerido para cada simulação.

A tarefa de construção do conjunto de treinamento pode ser muito facilitada se uma sistemática *on-line* for desenvolvida. O principal argumento da construção *on-line* é o fato de exigir esforço muito baixo na aquisição de dados. O próprio sistema de potência se encarrega de fornecer as diversas situações de operação e usá-las como base das simulações de Estabilidade de Tensão. Exemplificando, para a carga prevista com horizonte de quinze minutos, obtém-se uma amostra do estado do sistema. No passo seguinte, o processo de seleção e computação das contingências prováveis é feito através do programa de análise de Estabilidade de Tensão. Isto forma a base de dados para esta situação. Duas vantagens são nitidamente observadas em relação à metodologia *off-line*. A primeira mostra que, para a entrada de novos equipamentos no sistema, a base de dados é complementada apenas com o caso base representativo da nova topologia do sistema, reduzindo o esforço computacional. A segunda é que, a base de conhecimento formada espelha inúmeras situações de operação mais prováveis e realistas. Entretanto, o desenvolvimento de uma ferramenta *on-line* para formação de uma base de dados é uma tarefa bastante complexa, uma vez que estas são baseadas em programas convencionais e os dados de entrada devem ser tratados para eliminação de problemas como medição incompleta, erros grosseiros, etc.

Com o aumento da velocidade de processamento e de armazenamento dos computadores atuais, uma estratégia híbrida *off-line/on-line* pode ser desenvolvida. Neste caso, todos os prováveis cenários seriam inicialmente simulados *off-line*. As novas situações topológicas e a avaliação do realismo das simulações *off-line* poderiam ser feitas em ambiente *on-line*.

Uma metodologia bastante empregada para a geração da base de dados *off-line* é a geração de pontos de operação por processos aleatórios, como Monte Carlo. Este tipo de simulação proporciona um maior número de pontos de operação, abrangendo um universo maior de situações operativas que o sistema pode assumir. Entretanto, o MC, apesar de gerar menos pontos, pressupõe que o ambiente de crescimento de carga estabelecido seja realista em termos de conhecimento do comportamento do sistema. Isto mostra que a aplicação do MC fornece um

filtro, onde os pontos de operação obtidos estabelecem um cenário qualitativo ao problema de estabilidade de tensão. Podem ser também introduzidos na aplicação do MC critérios como restrição e aceleração do crescimento das cargas em algumas barras e/ou áreas, ação de controladores, etc., o que determina simulações mais realistas.

A cada ponto obtido pelo MC, as informações necessárias dos parâmetros do sistema, como tensão, carga, etc. devem ser armazenadas. Lembrando que o incremento na carga promove o movimento do sistema em direção ao ponto de colapso, o operador deve ser informado, a cada ponto de operação, o quão distante está o sistema do ponto de colapso, traduzido em termos de margem de potência ativa. Para uma dada topologia, um nível de segurança de tensão pode ser estabelecido a partir da informação desta margem de potência. Desta forma, a informação da margem pode ser diretamente armazenada como saída do processo de avaliação de segurança de tensão. Concluindo, o MC é a ferramenta analítica utilizada neste trabalho com o propósito da formação da base de dados.

A proposta neste estágio é proporcionar que o aplicativo desenvolvido para a formação da base de dados constitua um módulo de operação independente das etapas posteriores deste trabalho. Uma vez que tal aplicativo tem como base uma metodologia analítica convencional destinada à avaliação de estabilidade de tensão, este pode ser utilizado para outras finalidades como treinamento, teste de modelos, didática, etc. Pensando unicamente neste módulo utilizado como uma ferramenta de AS, a proposta é que o processamento siga uma sistemática própria direcionada a esta finalidade. A referência [GAO95] apresenta uma proposta mais detalhada para as bases para o desenvolvimento de uma sistemática em AST. Esta referência apresenta os requisitos mínimos necessários a esta aplicação e são descritos como segue:

- 1) *Estabelecimento do caso base*: Envolve determinar quais os modelos dos equipamentos relacionados ao problema de estabilidade de tensão devem ser utilizados e o detalhamento da representação do sistema elétrico.
- 2) *Seleção de contingências*: Escolha das contingências críticas a serem analisadas. Antes de se obter a lista destas, deve-se estabelecer um modelo para se estudar todas as contingências prováveis, abrangendo contingências de linhas de transmissão, de geradores, elementos de compensação *shunt*, dentre outros.
- 3) *Definição do parâmetro chave*: A partir do cálculo da margem de estabilidade de tensão do sistema, este parâmetro é a medida do quanto o sistema está próximo à instabilidade. Geralmente a margem é definida como a diferença de valores entre o ponto atual de operação e o ponto crítico.
- 4) *Especificação dos critérios de estabilidade de tensão*: São definidas quais margens

são suficientes para cada região de operação do sistema, atendendo ao caso base e as prováveis contingências.

- 5) *Determinação da margem de estabilidade de tensão do sistema:* A margem é determinada para o caso base e todas as prováveis contingências. Pode ser escolhido tanto o modelo dinâmico como o estático para o cálculo da margem. Entretanto, deve ser observado o tempo computacional envolvido em cada um dos modelos. Uma proposta apresentada é a utilização do modelo estático para o cálculo do caso base e todas as contingências e a utilização do modelo dinâmico aplicado a algumas contingências severas, para obtenção da cronologia do processo de instabilidade de tensão.
- 6) *Projeto e aplicação de medidas corretivas/preventivas:* Nos casos em que os critérios de estabilidade de tensão não são satisfeitos, ações de controle devem ser implementadas com a finalidade de satisfazer estes critérios. Para isto, um estudo de como, onde e quando aplicar estas medidas deve ser elaborado.

Neste trabalho, o roteiro acima é aplicado à formação da base de dados *off-line* e apresenta algumas variações, como poderá ser observado no Capítulo 5. Um caso base é estabelecido como proposto em (1), sendo incorporados o modelo discreto de LTC, limites de fornecimento de potência reativa e diferentes combinações de característica da carga (através do modelo ZIP). As contingências aqui estudadas, atendendo ao item (2), são as perdas de linhas de transmissão, perda de transformadores, perda de equipamentos de compensação de potência reativa e perda de geradores. Não são simuladas neste trabalho contingências duplas ou acima. Como todas as contingências devido à saída dos equipamentos acima descritos são consideradas equiprováveis, é proposto um modelo em que as simulações são feitas *off-line*.

O parâmetro chave para cálculo da margem de estabilidade de tensão é o aumento da potência ativa total no sistema. Neste ponto, o método da continuação (método estático) é aplicado a cada caso (base e todas as contingências). Isto fornece a evolução do sistema quando uma direção de crescimento de carga é proposta. Ao final do processo, a margem de potência ativa para cada caso, atendendo aos os requisitos (3) e (5) simultaneamente. Vale salientar que a evolução do sistema elétrico ao colapso é representada pelos inúmeros pontos de operação encontrados entre o ponto inicial e o ponto de colapso. Devido à característica deste trabalho, não há interesse na cronologia de casos mais críticos, não envolvendo simulações dinâmicas. Entretanto, como são armazenados os pontos de operação de cada caso e o *ranking* das contingências críticas é facilmente obtido, pode-se processar análise dinâmica, se necessário.

Para o requisito (4), uma variação mais adequada é utilizada. O modelo proposto em

[PAV85] e [WEH97] para os estados operativos de um sistema (Seção 2.2.1) é a base para estabelecer os níveis de segurança. Lembrando que, estes níveis são funções da margem de potência ativa e a faixa de valores que abrange cada nível é inerente da observação e experiência da engenharia de planejamento e operação de cada sistema elétrico. Deste fato, como o acesso a estas questões é muito difícil, o requisito para se estabelecer os níveis operativos neste trabalho é apenas um percentual da margem de potência ativa. Assim, por exemplo, o nível operativo Normal pode estar compreendido entre a margem total e 50% desta. Este critério é extremamente simples e não compromete os objetivos do trabalho.

Finalizando, o requisito (6) constitui um item que envolve metodologias apropriadas para esta finalidade, sendo as técnicas de otimização uma das principais. Neste caso, a aplicação de uma medida preventiva e/ou corretiva está associada com a obtenção de uma maior relação de custo/benefício da aplicação desta. A referência [HON02] faz uso de uma ferramenta desta natureza aplicada ao problema de estabilidade de tensão. Entretanto, este requisito (6) não é objeto de estudo neste trabalho. A seguir são apresentados o MC e o Método do Vetor Tangente.

2.3.1 Método da Continuação

Este é um dos principais métodos propostos na literatura para determinação do ponto de colapso de tensão. O comportamento de um sistema de equações genérico no espaço pode ser verificado através de um parâmetro que o movimentando ponto a ponto. Isto possibilita construir sua trajetória no espaço (ou diagrama da bifurcação) e identificar o ponto de bifurcação.

O estudo do comportamento da solução de um sistema de equações não lineares pode ser obtido através da teoria da bifurcação [SEY88]. As bifurcações mais comuns encontradas nos estudos de estabilidade de tensão são as de sela-nó e de Hopf. Se um modelo dinâmico de sistema é empregado, ambos os tipos podem ser detectados. Entretanto, se o modelo de fluxo de carga (estático) é utilizado, somente a bifurcação de sela-nó é possível de ser encontrada.

A bifurcação de Hopf é caracterizada pela existência de um par de autovalores puramente imaginários, enquanto que a bifurcação de sela-nó é caracterizada pela existência de um autovalor real nulo (matriz Jacobiana singular). Esta importante característica tem sido largamente explorada na literatura.

O MC, inserido no contexto da análise de estabilidade de sistemas e da teoria da bifurcação, destaca-se como a metodologia mais utilizada com objetivo de encontrar o ponto de bifurcação [I3E93a]. O método é dividido em duas etapas. A primeira visa encontrar uma

direção para mover o sistema e a segunda obter uma correção na trajetória de modo a encontrar a nova solução. Obtida a nova solução, o processo se repete até que a trajetória seja definida. Uma característica destes sistemas é apresentarem a matriz Jacobiana mal condicionada para pontos próximos do ponto da bifurcação. Desta forma, para se obter a solução no ponto de colapso deve-se empregar uma parametrização às equações para obtenção destes pontos. Isso remove a singularidade desta matriz. Os pontos de operação na parte de baixo da curva PV são obtidos da solução normal do fluxo de carga, apesar de serem considerados instáveis sob o ponto de vista elétrico. A seguir é mostrada a teoria básica do método da continuação.

De uma maneira genérica, pode-se descrever o método a partir do sistema de equações abaixo:

$$f(x, \lambda) = 0 \quad (2.1)$$

onde x é o vetor das variáveis de estado do sistema de equações e λ é o parâmetro que movimentam este sistema um ponto de equilíbrio a outro. No sentido de encontrar o ponto em que ocorre a bifurcação, ou seja, o limite em que o sistema se torna instável, é necessário encontrar a direção em que se deve mover o sistema. A tangente a um ponto de operação específico do sistema de equações, (x^k, λ^k) , onde k indica a solução atual, fornece a tendência de movimento. Na literatura, esta direção para o movimento é chamada de "passo previsor". A obtenção deste passo, a partir do ponto (x^k, λ^k) , é mostrado a seguir:

$$\frac{df}{dx} \Delta x + \frac{df}{d\lambda} \Delta \lambda = 0 \quad (2.2)$$

e obtém-se que

$$\frac{\Delta x}{\Delta \lambda} = - \frac{dx}{d\lambda} = t \quad (2.3)$$

onde t é denominado de vetor tangente. O passo previsor é obtido pela seguinte equação:

$$\Delta \lambda = \frac{k_r}{\|t\|} \quad (2.4)$$

com k_1 sendo uma constante que regula a velocidade com que $\Delta\lambda$ cresce ($k_1 < 1$ - retardado; $k_1 = 0$ - normal; $k_1 > 1$ - acelerado) e $\|t\|$ a norma euclidiana do vetor tangente. Deve-se observar que, $\lambda^{k+1} = \lambda^k + \Delta\lambda$, onde k é o estado corrente do sistema e $k+1$ o próximo ponto de equilíbrio. A Figura 2.2 ilustra a metodologia.

Um exemplo de como o vetor tangente e, conseqüentemente, o passo predictor são obtidos é mostrado no item 2.4.2 (Método do vetor tangente aplicado a um sistema elétrico de potência). Para a solução adequada, deve-se então corrigir a direção para a obtenção do novo ponto de solução estável (x^{k+1}, λ^{k+1}) a partir de $(x^k + \Delta x, \lambda^k + \Delta\lambda)$. Neste caso, este novo ponto pode ser obtido através de uma equação extra. Esta é dada pela intersecção perpendicular entre os vetores predictor e corretor (Δx , mostrado na Figura 2.2) como mostrada a seguir:

$$\rho(x, \lambda) = \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix}^T \begin{bmatrix} x - x^k - \Delta x \\ \lambda - \lambda^k - \Delta \lambda \end{bmatrix} = 0$$

$$\rho(x, \lambda) = \Delta x (x - x^k - \Delta x) + \Delta \lambda (\lambda - \lambda^k - \Delta \lambda) = 0 \tag{2.5}$$

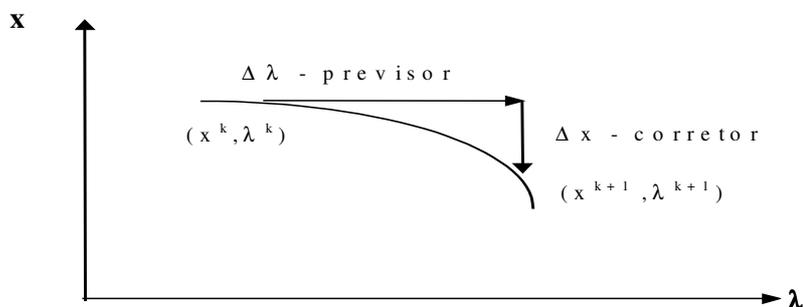


Figura 2.2 - Ilustração gráfica do método da continuação

Assim, considerando $(x^k + \Delta x, \lambda^k + \Delta\lambda)$ o valor inicial, a solução converge à (x^{k+1}, λ^{k+1}) . A desvantagem do uso da segunda equação é o esforço computacional para sistemas de grande porte. O vetor corretor Δx pode ser encontrado pela seguinte relação:

$$\Delta x = \Delta \lambda t \tag{2.6}$$

Algumas observações podem ser feitas a respeito do método. Muito próximo do ponto de bifurcação, o passo predictor é muito pequeno pois a inclinação da curva aumenta

significativamente. Se houver a necessidade de traçar o diagrama completo, o mal condicionamento da matriz Jacobiana muito perto do ponto de bifurcação, é resolvido parametrizando-se as equações. Isto é feito substituindo-se a coluna da matriz associada ao maior componente do vetor tangente t , pelo próprio vetor tangente. Isto produz uma rotação na trajetória, e a nova solução das equações fica em função do $\Delta\lambda$ relacionado com a variável substituída. Esta parametrização deve ser feita alguns passos antes do ponto de bifurcação, podendo-se utilizar um valor de $\Delta\lambda$ pequeno como referência, por exemplo.

No âmbito de sistemas elétricos de potência, o parâmetro λ pode ser descrito como um incremento de carga elétrica. Utilizando-se o MC e tendo o incremento de carga como parâmetro, o diagrama da bifurcação para o sistema elétrico em análise pode ser construído. Logo, o ponto de bifurcação (bifurcação de sela-nó) indica o ponto de colapso de tensão do sistema. Vale salientar também que existe uma alternativa computacionalmente mais eficiente do que aplicar as Equações 2.5 e 2.6 (passo corretor). A solução é considerar o valor inicial dado por V^k e θ^k da solução atual do fluxo de potência, uma vez que o método numérico de Newton-Raphson proporciona a obtenção de V^{k+1} e θ^{k+1} .

As Figuras 2.3 e 2.4 mostram o diagrama da bifurcação (ou a curva PV) para o sistema teste IEEE 14 barras. Observe que os diagramas são construídos até o ponto de colapso de tensão, não sendo de interesse do trabalho realizado a obtenção de pontos localizados na parte de baixo da curva, dito pontos de solução instáveis. Além disso, a variável de estado monitorada é a tensão em função da margem de potência ativa do sistema. Neste caso, a margem de potência ativa é função do parâmetro λ , implícito na obtenção desta margem.

Nos diagramas obtidos por ambas figuras, pode ser notado que ocorre uma descontinuidade na curva. Isso é devido ao limite de fornecimento de potência reativa ter sido alcançado. Isso determina, na realidade, uma mudança na trajetória do sistema, ou seja, ocorre uma mudança na trajetória do sistema, configurando uma nova curva. Na Figura 2.4 é mostrada a simulação onde é introduzido um equipamento LTC no transformador presente entre as barras 4 e 9 do sistema (Apêndice G), controlando a tensão na barra 9 (secundária). A principal diferença entre estas curvas é a margem de potência ativa. Observe que a introdução do LTC contribuiu para a redução da mesma, fruto da maior solicitação de potência reativa exigida para se manter o nível de tensão na barra controlada. Isso, como tem sido colocado na literatura atinente à área, leva mais rapidamente o sistema ao colapso de tensão. A comparação destas curvas mostra claramente este fenômeno.

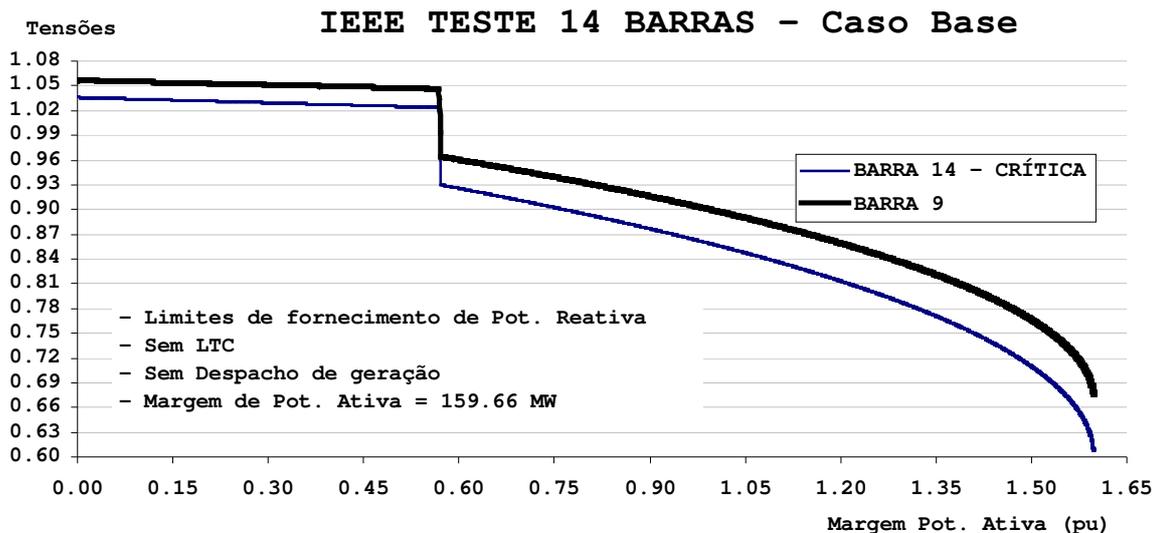


Figura 2.3 – Diagrama da bifurcação para o sistema teste IEEE 14 barras simulado considerando-se apenas a presença de limites de fornecimento de potência reativa

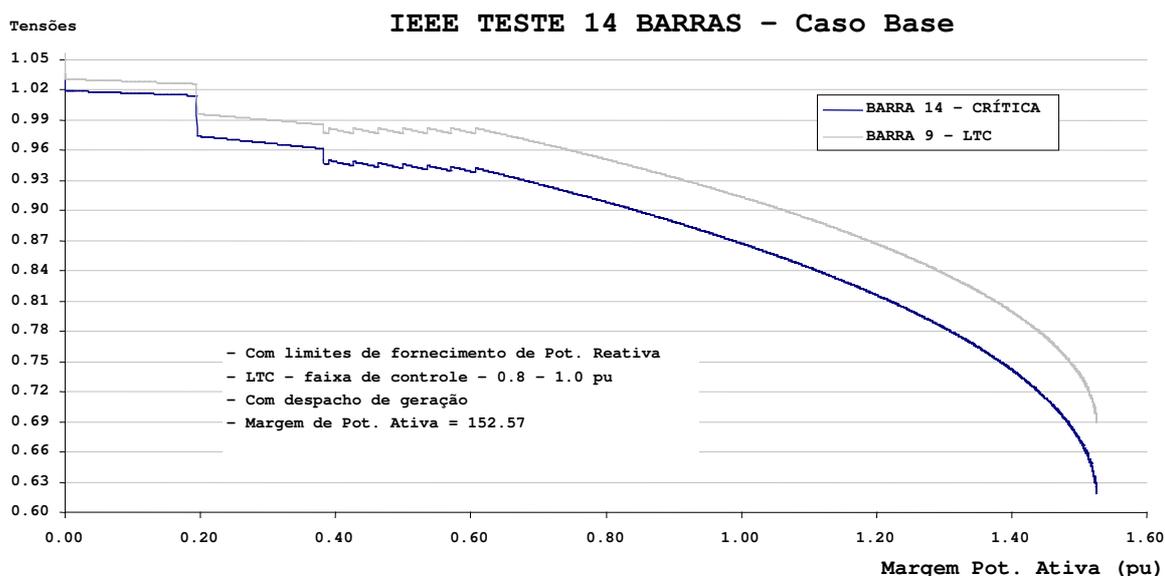


Figura 2.4 – Diagrama da bifurcação para o sistema teste IEEE 14 barras simulado com um equipamento LTC controlando a barra 9 do sistema e limites de fornecimento de potência reativa

2.3.2 Método do Vetor Tangente

Proposto em [SOU96a, SOU96b, SOU97] para aplicações em análise de estabilidade de tensão, baseia-se no comportamento do maior componente do vetor tangente em função do crescimento da carga elétrica. O cálculo deste vetor é facilmente incorporado em qualquer programa de fluxo de carga convencional. Logo, sua facilidade de obtenção constitui uma grande

vantagem em relação aos métodos previamente descritos.

O vetor tangente, genericamente formulado pela Equação 2.3, é obtido a partir da manipulação das equações linearizadas do fluxo de carga, dadas por:

$$\begin{bmatrix} \Delta\theta_g \\ \Delta\theta_l \\ \Delta V_l \end{bmatrix} = [J^{-1}] \begin{bmatrix} \Delta P_g \\ \Delta P_l \\ \Delta Q_l \end{bmatrix} \quad (2.7)$$

onde θ é o ângulo de fase da tensão (V) para as barras de geração (índice g) ou de carga (índice l), P e Q são as potências ativa e reativa respectivamente e J é a matriz Jacobiana do sistema (Método de Newton-Raphson). Para efeito de simplificação, aqui é apresentado o desenvolvimento matemático apenas em a relação ao componente tensão ($\Delta V_l/\Delta\lambda$) do vetor tangente. Quanto às componentes ângulos ($\Delta\theta_g/\Delta\lambda$ e $\Delta\theta_l/\Delta\lambda$), a obtenção é similar. A seguinte equação pode ser obtida:

$$\Delta V_l = \sum_{k=l}^{npv} J_{lk}^{-1} \Delta P_k + \sum_{k=npv+1}^{npv+npq} J_{lk}^{-1} \Delta P_k + \sum_{k=npv+1+npq}^{npv+2*npq} J_{lk}^{-1} \Delta Q_k \quad (2.8)$$

onde os índices npv e npq são os números de barras de tensão controlada e de carga, respectivamente. Considerando λ o incremento na carga de uma barra k , as novas potências são:

$$\begin{aligned} P_k &= P_{k0}(1 + \Delta\lambda) \\ Q_k &= Q_{k0}(1 + \Delta\lambda) \end{aligned} \quad (2.9)$$

onde P_k e Q_k são as cargas ativa e reativa considerando a variação do parâmetro λ , e P_{k0} e Q_{k0} são as cargas ativa e reativa inicialmente ligadas à barra k . Lembrando que, os incrementos de potência são dados pela Equação 2.10 abaixo,

$$\begin{aligned} \Delta P_k &= P_k - P_{k0} \\ \Delta Q_k &= Q_k - Q_{k0} \end{aligned} \quad (2.10)$$

obtem-se de (2.10) em (2.9) as seguintes relações abaixo,

$$\begin{aligned}\frac{\Delta P_k}{\Delta \lambda} &= P_{k0} \\ \frac{\Delta Q_k}{\Delta \lambda} &= Q_{k0}\end{aligned}\quad (2.11)$$

Logo, pode-se obter

$$\begin{aligned}\Delta P_k &= P_{k0} \Delta \lambda \\ \Delta Q_k &= Q_{k0} \Delta \lambda\end{aligned}\quad (2.12)$$

Substituindo (2.12) em (2.8), obtém-se a Equação 2.13 abaixo

$$\Delta V_l = \sum_{k=1}^{npv} J_{lk}^{-1} P_{k0} \Delta \lambda + \sum_{k=npv+1}^{npv+npq} J_{lk}^{-1} P_{k0} \Delta \lambda + \sum_{k=npv+1+npq}^{npv+2*npq} J_{lk}^{-1} Q_{k0} \Delta \lambda \quad (2.13)$$

O vetor tangente é então obtido através a Equação 2.14 abaixo, lembrando que o procedimento para obtenção de $\Delta \theta_g / \Delta \lambda$ e $\Delta \theta_l / \Delta \lambda$ seguem o mesmo caminho.

$$\frac{\Delta V_l}{\Delta \lambda} = \sum_{k=1}^{npv} J_{lk}^{-1} P_{k0} + \sum_{k=npv+1}^{npv+npq} J_{lk}^{-1} P_{k0} + \sum_{k=npv+1+npq}^{npv+2*npq} J_{lk}^{-1} Q_{k0} \quad (2.14)$$

O vetor tangente identifica a barra crítica do sistema elétrico em análise. Esta é obtida observando-se a maior componente em valores absolutos deste vetor. Conforme citado anteriormente, a vantagem do vetor tangente é que esta informação é fornecida para pontos anteriores ao ponto de colapso, característica não observada com o autovetor à direita associado ao menor autovalor da matriz Jacobiana (Apêndice A). Entretanto, ambos vetores convergem à mesma solução no ponto de colapso. Isto é mostrado na referência [SOU00].

A Tabela 2.1 sintetiza a aplicação do MC e da análise proporcionada pelo método do vetor tangente para o rol de contingências do sistema teste IEEE 14 barras (Apêndice G) para a simulação mostrada na Figura 2.3. Nesta tabela é apresentada a margem de potência ativa de cada caso, as respectivas barras críticas no ponto de colapso de tensão e o *ranking* de cada caso no rol de contingências.

A referência [CAN95] analisa o comportamento da componente do inverso da maior entrada do vetor tangente e a da componente associada à barra crítica para o sistema teste 300 barras do IEEE, computado com limites de geração de potência reativa. Os resultados mostram

que ambas as componentes resultam em uma única curva, estando o sistema bem longe do ponto de colapso. Isso mostra que o vetor tangente proporciona a indicação da barra crítica do sistema de forma bastante antecipada, caracterizando sua maior vantagem. Vale salientar que a informação antecipada da barra crítica não é fornecida por nenhuma das outras técnicas, como autovalores, valores singulares e Jacobiano reduzido (Apêndice G).

Entretanto, o comportamento do vetor tangente em função do aumento de carga produz uma descontinuidade na curva quando os limites de fornecimento de potência reativa são considerados (observada também no comportamento do determinante reduzido). Isso desqualifica o método do vetor tangente como indicador de proximidade do colapso. Este problema é superado pelo método da extrapolação quadrática [SOU97].

Tabela 2.1 – Margem de potência ativa, *ranking* das 3 barras mais críticas e a posição de cada contingência.

CONTINGÊNCIA	CASO B14N	Margem ($P_c - P_0$) (MW)	BARRAS CRÍTICAS			POS
			+	→	-	
BASE	B14N	159.70	14	13	12	22
PERDA BC9	01	148.73	14	13	12	16
PERDA LT1	02	54.10	14	3	10	3
PERDA LT2	03	62.75	14	13	12	4
PERDA LT3	04	22.99	3	14	10	2
PERDA LT4	05	108.77	14	13	12	6
PERDA LT5	06	127.29	14	13	12	10
PERDA LT6	07	130.12	3	14	13	11
PERDA LT7	08	119.91	14	9	10	8
TRAFO 4-7	09	113.40	14	9	7	7
TRAFO 4-9	10	137.40	14	10	13	14
TRAFO 5-6	11	18.13	12	13	6	1
PERDA LT11	12	156.26	14	11	10	19
PERDA LT12	13	153.78	12	14	13	17
PERDA LT13	14	134.77	13	14	12	12
PERDA LT14	15	135.64	10	14	11	13
PERDA LT15	16	85.84	14	9	10	5
PERDA LT16	17	146.68	10	11	12	15
PERDA LT17	18	124.30	14	13	12	9
PERDA LT18	19	159.31	14	13	12	20
PERDA LT19	20	159.44	14	13	12	21
PERDA LT20	21	155.57	14	10	9	18

2.4 CONCLUSÕES

Este capítulo apresentou as principais definições e requisitos mínimos para o problema de avaliação de segurança de tensão. Mais especificamente, para uma aplicação on-line, o enfoque é analisar o ponto de operação atual do sistema e os prováveis cenários que o sistema pode evoluir a partir daí. Desta forma, mostra-se necessário saber a região de operação em que se encontra o sistema e o quão seguro este se encontra. Os critérios para a determinação destas regiões são estabelecidos de acordo com a avaliação das restrições de igualdade e desigualdade.

As definições proposta na literatura sobre os tipos de fenômenos de estabilidade encontrados atualmente auxiliam a delimitar o âmbito do processo de avaliação de segurança. Isto norteia a utilização das ferramentas necessárias para o processamento de cada estudo. No contexto deste trabalho, foram apresentadas as considerações principais sobre estabilidade de tensão, avaliando as vantagens e desvantagens de cada metodologia empregada, sendo descritos os métodos da continuação e do vetor tangente.

Para a formação do banco de dados, o método da continuação é escolhido devido à obtenção da curva PV, dispondo informação do sistema até ao colapso através de pontos de operação. Além disso, o vetor tangente como parte integrante do método, facilita a análise das barras críticas.

De acordo com um roteiro proposto na literatura, uma sistemática de avaliação de estabilidade é proposta. Isto proporciona que a etapa de formação da base de dados constitua um módulo independente e pode ser utilizada com outras finalidades.

Capítulo 3

AQUISIÇÃO AUTOMÁTICA DE CONHECIMENTO

3.1 INTRODUÇÃO

O presente capítulo mostra a motivação para a utilização de técnicas de IA utilizando AAC e apresenta conceitos imprescindíveis que afetam estas técnicas. Estes conceitos são a representação do conhecimento, o problema da ‘maldição da dimensionalidade’ e a maneira como um mecanismo de AAC infere em uma base de dados. Uma das principais técnicas utilizadas para AAC é o aprendizado por máquina, sendo as DTs uma das principais representantes desta abordagem. Este capítulo mostra a conceituação teórica de DTs e os aspectos relativos a alguns algoritmos nesta área.

Na direção do objetivo deste trabalho de tese, este capítulo apresenta os conceitos da extração de regras de produção de RNAs. Esta tarefa é realizada pelo algoritmo denominado de VIA [THR94]. As informações provenientes de um sistema elétrico de potência geralmente geram uma grande base de dados, tanto em relação à cardinalidade do conjunto de padrões, quanto em relação à dimensionalidade do vetor de entrada. Para atender estes requisitos, a rede neural utilizada no processo é a OET2 ([SIL91], [SIL94]). Além de atender as características citadas, o algoritmo de aprendizado desta é extremamente rápido, sendo as informações resultantes do treinamento repassadas ao algoritmo VIA.

3.2 AQUISIÇÃO AUTOMÁTICA DE CONHECIMENTO

Conforme já observado, a complexidade dos problemas em sistemas elétricos de potência tem aumentado significativamente. A utilização de técnicas de IA como metodologia alternativa para o tratamento destas complexidades vem se tornando cada vez mais popular. Dentro desta perspectiva, e como já observado no Capítulo 01, o objetivo deste trabalho é comparar técnicas de IA enfocadas na extração de conhecimento de um sistema ou processo. Estas comparações auxiliam que tais técnicas possam ser utilizadas em ferramentas de apoio à tomada de decisão para AST baseadas em IA. A Figura 3.1 mostra a tarefa de produzir um conjunto de regras de produção.

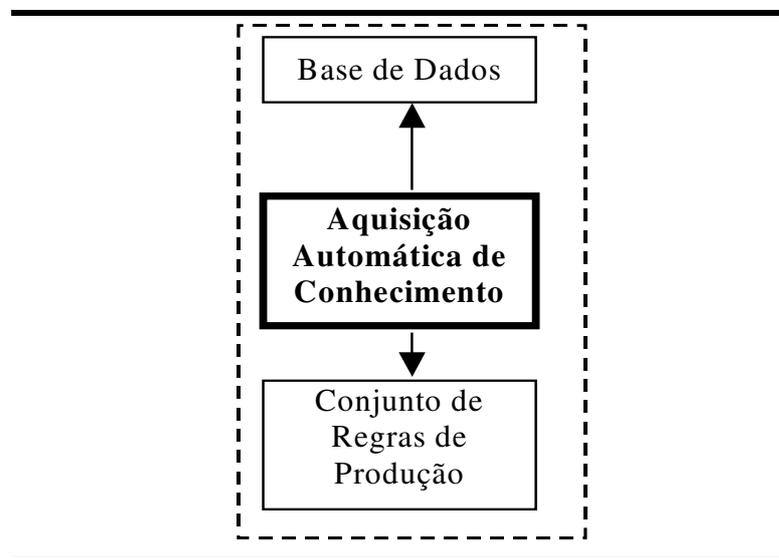


Figura 3.1 - Esquema básico para gerar um conjunto de regras de produção

No Capítulo 01 foram apresentadas as vantagens de uma aplicação de IA comparada às metodologias analíticas convencionais para AST. Entretanto, a primeira questão a ser respondida é: por quê obter conhecimento através de um processo automático? No processo de aquisição manual, as informações são obtidas diretamente do especialista e/ou materiais de referência. A principal barreira deste tipo de abordagem é que, a informação como está estruturada no especialista ou no material de referência, nem sempre é adequadamente transferida a um programa computacional. Isto é inerente aos seres humanos, que não são hábeis em explicitar o conhecimento sobre um dado assunto, podendo ocasionar inconsistência no conhecimento codificado. Logo, a idéia é tornar o processo automático, devendo o mesmo refletir de forma precisa o problema em estudo.

Como o processo de AAC começa na base de dados, esta deve se espelhar no comportamento do sistema, como dito no Capítulo 2. É importante que a técnica de AAC utilizada seja precisa o bastante para extrair o conhecimento que se encontra implícito nesta base de dados. Entretanto, surge uma nova questão envolvida neste processo. Esta diz respeito à forma de representação do conhecimento. Esta questão pode ser bem entendida através da referência [RUS95], onde as principais formas de representação são discutidas, tais como Lógica Proposicional e Lógica de Primeira-Ordem. Neste trabalho é utilizada uma representação do tipo *Se-Então*, ou seja, uma regra tem o formato *premissa-conclusão*. A premissa (ou antecedente) da regra é composta por atributos como variáveis lógicas, variáveis numéricas, etc. Estes atributos são conectados por conexões lógicas E (\wedge) e OU (\vee). A conclusão (ou consequência) contém a implicação da combinação dos atributos da regra. A esta forma de representação é dado o nome de *regra de produção*. Um exemplo deste tipo de representação pode ser dado através da seguinte regra:

Se((TensãoBarra1 \geq 0.97 pu) **E** (PotênciaAtivaBarra2 \leq 125,5 MW))
Então sistema elétrico estável.

Observa-se neste exemplo que os dois atributos que compõe o antecedente desta regra estão determinando um fato. Conclui-se então que o conhecimento do fato 'sistema elétrico estável' está sendo explicitado através dos atributos que compõe o antecedente da regra.

Uma característica própria da IA é tentar reproduzir a forma do raciocínio humano. Dentro desta perspectiva, o conhecimento adquirido sobre um sistema só estará completo se situações novas apresentadas ao conjunto de regras fossem associadas corretamente com algum fato do problema. Esta característica é chamada de 'capacidade de generalizar do modelo' e é o grande diferencial em relação às metodologias computacionais convencionais. Como os atributos selecionados pela técnica de AAC para compor o antecedente da regra respondem pelo conhecimento adquirido, se estes não forem suficientemente correlacionados com o fato, a regra obtida pode classificar um número reduzido de situações novas que envolvem o fato em análise. Logo, o conhecimento traduzido por esta regra é pequeno. Nota-se então que a qualidade do aprendizado está diretamente relacionada com a explicitação dos atributos mais correlacionados com um determinado fato.

Há casos onde regras classificam poucas situações novas mas são importantes, refletindo situações de difícil correlação com o fato. Estas situações atípicas são denominadas de *outliers*. Manter ou não este tipo de regra no conjunto de regras extraídas é uma análise de custo/benefício

em função do aumento da complexidade desta base de informação. A referência [DIL01] apresenta uma técnica de AAC para a extração de regras de produção de RNAs onde os autores se preocupam em extrair regras que também propiciam visualizar os *outliers* do sistema.

Especificamente neste trabalho, as regras extraídas pela técnica de AAC têm a função de classificar os níveis operativos (Capítulo 2) para AST. Entretanto, inserido neste processo está a determinação dos atributos que compõe os antecedentes das regras mais correlacionáveis com estes níveis operativos. A idéia é que estes atributos relevantes sejam suficientemente informativos. Isso possibilita que a manipulação adequada destes indiquem à operação do sistema elétrico as ações de controle mais eficientes. Logo, justifica-se a comparações entre técnicas de extração de regras de produção para determinar qual dessas pode proporcionar um conjunto de regras que atendam aos requisitos de uma ferramenta para AST, como descrito no Capítulo 2.

Um processo de aprendizado automático para a obtenção de um conjunto de regras de produção como mostrado na Figura 3.1, é dito um *processo indutivo*. Isto é devido ao fato do aprendizado ser baseado em exemplos. Um aplicativo que utiliza técnicas indutivas de aprendizado automático pode ser formalmente dividido em seis passos, como proposto em [CHE98]. A tarefa de AAC pode ser então melhor entendida analisando-se cada um destes passos propostos e definindo claramente as opções adotadas. Tais passos são:

1. *Declaração do problema*: O conhecimento específico a respeito do problema é usualmente utilizado para modelá-lo apropriadamente. No âmbito deste trabalho, o conhecimento da teoria de estabilidade de tensão auxilia na escolha de uma metodologia apropriada para fornecer os dados.
 2. *Formalização das hipóteses*: As hipóteses são formuladas para se detectar os fatos desconhecidos presentes nos dados. Geralmente, nesta etapa são definidas as variáveis de entrada e de saída. Em avaliação de segurança de tensão, o que se espera é visualizar barras e áreas críticas do sistema elétrico, bem como os fatos que influenciam o comportamento atual do sistema elétrico, tais como ações de controle e comportamento da carga. Estes fatos não são explicitamente observados nos dados, e a escolha das variáveis de entrada é um processo advindo da experiência do projetista e/ou especialista em sistema elétrico de potência.
 3. *Geração dos dados*: Este passo consiste de como os dados são gerados. Duas possibilidades se apresentam: na primeira, o projetista detém o controle do processo de
-

geração dos dados, denominado de *experiência projetada*. A segunda, o projetista não interfere no processo. Neste caso, o processo é denominado *observacional*. O Capítulo 02 deste trabalho mostra a estratégia (neste caso, observacional) do aplicativo utilizado para gerar a base de dados, tendo como base o método da continuação.

4. *Coleta de dados e Pré-processamento*: Uma vez coletados, os dados devem sofrer um pré-processamento antes que possam ser utilizados. Numa primeira fase, devem ser observadas as questões que levam a inconsistências nos dados, como erros de medições, erros de codificação e casos anormais. Numa segunda fase, os dados podem sofrer um processo de mudança de escala e redução de dimensionalidade do vetor de variáveis de entrada.
5. *Estimação do modelo*: As hipóteses formuladas no passo 2 correspondem a relações desconhecidas entre as características de entrada e as saídas. Estas relações podem ser quantificadas utilizando dados e o conhecimento primário sobre o problema. O objetivo torna-se então construir modelos para a estimação precisa das saídas futuras associadas às entradas conhecidas. Isto é denominado de capacidade de generalização do modelo.
6. *Interpretação do modelo*: Os modelos desenvolvidos no passo 5 precisam, muitas vezes, ser utilizados em processos de tomada de decisão. Ou seja, eles necessitam ser interpretáveis para serem úteis na prática. Nota-se que as metas de interpretação e estimação precisas são contraditórias entre si. Enquanto modelos interpretáveis devem ser simples, modelos para estimação podem ser complexos. Os modelos estatísticos convencionais lidam com este dilema usando modelos paramétricos, enquanto modelos mais modernos favorecem a utilização de métodos para estimação de alta precisão, considerando o problema da interpretação como uma tarefa realizada a parte.

Pela descrição dos passos acima, pode-se notar que os passos 5 e 6 representam o enfoque principal deste trabalho, que é a utilização de técnicas de AAC em avaliação de segurança de tensão. Entretanto, vale salientar que uma técnica adequada de AAC é também dependente dos resultados obtidos nos passos anteriores. Em muitas situações, bons resultados obtidos nestes passos determinam um melhor desempenho da AAC.

Observando os passos descritos, o Capítulo 2 apresentou a formulação do problema de AST, o levantamento das prováveis questões implícitas (hipóteses) e como é formada a base de dados. Existem 3 grupos básicos de abordagens para o problema de AAC dentro da área de reconhecimento de padrões: estatístico, neural e aprendizado por máquina.

As técnicas estatísticas são as mais clássicas e amplamente difundidas na área de reconhecimento de padrões. Aprendizado por máquina, representado principalmente pelas DTs, tem a estratégia de dividir o espaço amostral, por intermédio de hiperplanos, em regiões cada vez menores à medida que o mapeamento do problema cresce. Embora estes três grupos de técnicas podem ser ditos conceitualmente diferentes, alguns autores ([SCH97],[CHE98]) têm procurado mostrar que algumas técnicas neurais estão fundamentadas em técnicas estatísticas.

Entretanto, um paradigma, seja estatístico, neural ou baseado em DTs, pode fazer uso de outro para implementar melhoramentos. Neste caso são chamados de *híbridos*. Por exemplo, em [SES94] são mostradas algumas árvores de decisão que utilizam conceitos estatísticos para promover aperfeiçoamentos na construção da árvore. Esta mesma referência apresenta uma metodologia (BRAINNE) baseada em uma estrutura de árvore de decisão para transformar o conhecimento adquirido por uma RNA para o nível simbólico. Assim, muitas metodologias podem ser melhoradas utilizando-se de aspectos positivos de outras.

O Apêndice B mostra algumas técnicas estatísticas mais comumente utilizadas, representadas pelo Classificador de Bayes, a estimação de densidades de probabilidade, funções discriminantes e ajuste de curvas por polinômios. Estas técnicas podem ser consideradas as metodologias mais naturais para aprendizado baseados em exemplares (dados). Entretanto, estas enfrentam problemas como a necessidade de um grande requerimento de dados e dificuldades em lidar com relações mais complexas e não-lineares. Além disso, os resultados são apresentados em um formalismo estatístico, necessitando decodificação para tornarem-se úteis em processos de tomada de decisões.

Neste capítulo é mostrado o mecanismo de AAC baseado em DTs. Estas são modelos onde a classificação e a interpretação caminham juntas. Entretanto, na direção da extração de regras de produção, primeiramente algumas questões devem ser abordadas. Estas seriam relacionadas com pré-processamento dos dados (passo 4) e a forma como os mecanismos de inferência para AAC atuam na base de dados.

3.3 PRÉ-PROCESSAMENTO DOS DADOS

Atualmente, com o desenvolvimento de sistemas automáticos de coleta de dados em tempo real e a evolução dos sistemas computacionais (*hardware* e *software*), é possível a

construção de grandes bases de dados. Entretanto, a manipulação das informações contidas nestas não é um processo simples. Um exemplo disto pode ser visto a seguir. Obtendo-se um conjunto de medidas de tensão das barras de um sistema elétrico, supõe-se que é observado que estas tensões estão diminuindo ao longo do tempo. Necessitando-se corrigir este problema, apenas esta informação não é suficiente para indicar a melhor maneira de fazê-lo. Então, muitas questões que não são facilmente observadas podem ser formuladas. Por exemplo, em quais barras do sistema esta queda de tensão é mais acentuada? Há alguma ação de controle de tensão já sendo implementada? Para sistemas de pequeno porte é possível que o engenheiro especialista consiga esclarecer estas questões apenas pela observação e experiência. Todavia, isto não é plausível para sistemas elétricos de grande porte, onde a base de dados formada contém uma massa de informações muito grande.

O problema torna-se então como extrair o conhecimento implícito e relevante presente na base de dados. Isto pode ser feito através das técnicas de AAC. Todavia a aplicação da grande maioria destas técnicas envolve adequar, reparar ou ceifar os dados a serem utilizados. Algumas referências ([CHE98],[BIS95]) indicam que este pré-processamento pode melhorar significativamente o desempenho da técnica escolhida.

Basicamente são identificadas duas tarefas. A primeira destina-se à verificação da existência de inconsistências nestas informações que podem afetar significativamente a técnica escolhida. Como já mencionado, tais inconsistências são causadas devido a erros grosseiros de medição, erros de codificação e armazenamento e casos considerados anormais, podendo ser corrigidas ou removidas. Outra possibilidade é a utilização de métodos estatísticos robustos.

No segundo estágio do pré-processamento define-se o fator de escala de cada variável de entrada. Muitas variáveis apresentam unidades de medida diferentes, como tensão (volts) e potência ativa (watts). Algumas técnicas são sensíveis a esta diferença, como por exemplo os métodos baseados em medidas de distância, outras não, como DTs. Então, dependendo do método escolhido, cada variável de entrada precisa sofrer uma mudança de escala, de tal modo que isto proporcione a estas variáveis ter valores similares. Uma maneira é utilizar um processo ao qual aplica uma transformação linear nas variáveis, como por exemplo, uma normalização dos valores das variáveis no intervalo [0,1].

Outro estágio do pré-processamento é reduzir a dimensionalidade do vetor das variáveis de entrada, também denominado de seleção de características (*feature selection*), cuja discussão é feita a seguir.

Seleção e extração de características

A questão da alta dimensionalidade do vetor de entrada pode ser vista pelo seguinte exemplo: considerando um número fixo de exemplares de um conjunto de dados. Cada exemplar \mathbf{x} , indicando uma dada variável de saída y , contém uma quantidade d de variáveis, onde $\mathbf{x} = \{x_1, \dots, x_d\}$. Levando-se em conta que cada variável de \mathbf{x} apresenta uma faixa de valores, esta pode ser dividida em I intervalos. Repetindo este procedimento para todas as variáveis, isto determina que o espaço de entrada seja dividido em células d -dimensionais. Cada exemplar da amostra é mapeado dentro de uma destas células, levando consigo o valor da variável de saída y .

Testando-se um novo exemplar, deve ser verificado em que célula este se encaixa. Pode-se daí retornar com o valor médio de y , obtido pelos pontos de treinamento pertencentes a esta célula. Observa-se que, se o número de células aumentar, a precisão da resposta aumenta. Entretanto, como cada variável é dividida em I intervalos, o número de células é igual a I^d . Logo, se o número de variáveis (d) aumenta, o número de células cresce exponencialmente. Como cada célula contém no mínimo um dado, isto significa que o número de exemplares deve também crescer exponencialmente. Este fenômeno é conhecido como *maldição da dimensionalidade*.

De uma maneira geral, pode-se dizer que o aumento da densidade amostral é benéfico à precisão da técnica. Se um dado trabalho não proporciona condições de aumentar o número de exemplares, então o aumento da densidade amostral só é processado reduzindo-se a dimensionalidade do vetor de variáveis de entrada. Isso pode ser feito de duas maneiras: retirando-se simplesmente as variáveis que não contribuem com informação significativa (seleção de características), ou sintetizando a informação de todas as variáveis em um número menor de novas variáveis através de uma transformação (extração de características). Vale salientar que, geralmente, ocorre perda de informação no processo de redução da dimensionalidade, o que não é necessariamente prejudicial ao processo de classificação em particular.

As referências [BIS95], [DIL95] e [CHE98] apresentam uma técnica estatística muito utilizada para promover redução de dimensionalidade em problemas de regressão e de compactação de dados, denominada de *Análise de Componentes Principais*. Outras técnicas são propostas para problemas de classificação, tais como análise de discriminantes, *clustering* ([CHE98],[DUD73]) e RNAs tipo Auto-organizáveis como a RNA-MK ([HAS95],[CHE98]).

3.4 TÉCNICAS DE INFERÊNCIA NA BASE DE DADOS

Uma outra questão a ser analisada é a maneira como uma dada técnica de AAC manipula o vetor de variáveis de entrada ou, caso este sofra um processo de seleção de características, as novas características encontradas. Considere a existência de duas classes, **C1** e **C2**, a serem classificadas a partir de um conjunto de dados com **N** exemplares, onde o vetor de variáveis de entrada é dado por $\mathbf{x}=\{x_1, \dots, x_d\}$ e \mathbf{X}_1 e \mathbf{X}_2 são as características obtidas através de um método qualquer de redução de dimensionalidade.

A Figura 3.2a mostra as distribuições dos **N** exemplares em função das classes e apenas do valor de \mathbf{X}_1 . Observa-se que existe uma região de sobreposição de valores. Se um novo exemplar \mathbf{X}'_1 , pertencente à classe **C1**, é testado e constata-se que este pertence à região de sobreposição, este novo exemplar pode ser classificado incorretamente, uma vez que esta região gera uma indefinição na classificação.

Entretanto, para minimizar este problema, um ponto (valor) limite de \mathbf{X}_1 pode ser estabelecido, ou seja, abaixo deste valor, classifica-se como **C1**, senão **C2** (quando as probabilidades de ocorrência de elementos pertencentes às classes **C1** e **C2** são iguais). As referências [DUD73] e [BIS95] mostram que o ponto onde é obtido o erro mínimo de classificação é o ponto de intersecção entre as funções de distribuição. Embora este procedimento maximize o grau de discriminação entre as classes, exemplares ainda podem ser classificados erradamente.

Um caminho para melhorar a taxa de classificação é avaliar as características \mathbf{X}_1 e \mathbf{X}_2 conjuntamente. A Figura 3.2b mostra um exemplo de distribuição das classes **C1** e **C2** no plano definido por \mathbf{X}_1 e \mathbf{X}_2 . Pode ser observada a definição de uma linha de separação entre as classes, isto é, um classificador linear. Supondo que este seja obtido de tal forma a minimizar o erro de classificação, nota-se que a discriminação entre as classes é sensivelmente melhorada em relação à *avaliação individual*, feita somente através de \mathbf{X}_1 . Entretanto, ainda há alguma superposição entre as distribuições de classes. A conclusão que se pode observar é que, o aspecto da *avaliação conjunta* das variáveis de entrada é favorável à precisão do classificador. Exemplos de abordagem com avaliação conjunta são as RNAs e exemplos com *avaliação individual* (uma variável analisada em cada etapa) são as árvores de decisão e regressão.

Aprendizado por máquina (do inglês, *machine learning*) são algoritmos de aprendizado indutivo cujo resultado do processamento destes é expresso em um formato interpretável aos seres humanos (modelo simbólico). As árvores de decisão proporcionam isso diretamente, fruto

do tipo da metodologia empregada. Para as RNAs isso não acontece, sendo necessário que métodos sejam implementados para extrair conhecimento simbólico destas. Entretanto, como dito no parágrafo anterior, as árvores de decisão utilizam-se de uma abordagem em que as variáveis são avaliadas individualmente. Logo, o conhecimento extraído de um problema por uma árvore de decisão pode levar a resultados menos precisos que as RNAs, pois estas fazem uso da avaliação conjunta de variáveis.

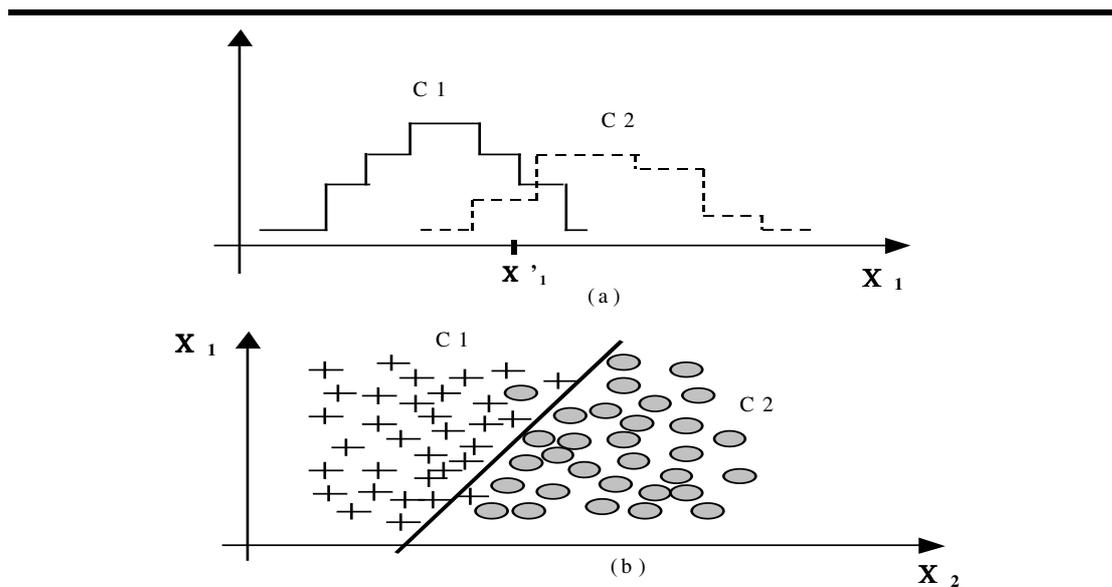


Figura 3.2 - (a) curvas de distribuição levando em conta somente a avaliação da característica X_1 , (b) avaliação conjunta das características X_1 e X_2 , mostrando a fronteira de decisão.

Baseado na premissa que as RNAs são mais precisas que as árvores de decisão, este trabalho tem o objetivo de aproximar o modelo neural de um modelo simbólico, cujos conceitos e métodos para esta finalidade são apresentados no item 3.6. As comparações entre as regras simbólicas obtidas por uma árvore de decisão e as obtidas através de um processo de extração de regras de uma rede neural são então apresentadas no Capítulo 4. A seguir, finalizando este capítulo, são apresentados os conceitos a respeito das árvores de decisão.

3.5 ÁRVORES DE DECISÃO

Topologicamente, uma árvore de decisão é um gráfico direto constituído de nós e arcos direcionados. Os nós correspondem a um ponto da árvore onde um teste é executado. Este teste tem a finalidade de verificar a possibilidade de expansão da árvore. Podem ser divididos em três

classificações: *nó raiz*, *nós intermediários* e *folhas*. O nó raiz é o nó inicial o qual a árvore é expandida. Nós intermediários são os pontos nos ramos da árvore onde há expansão. As folhas são nós que não sofrem mais expansão e contém uma decisão final da árvore. Em problemas de classificação, cada folha designa uma classe. Ao final do processo, a classificação é feita traçando-se um caminho do nó raiz para uma folha, através dos ramos que levam a esta.

O teste realizado em um nó significa encontrar a variável com maior capacidade de discriminação, de maneira seqüencial. Uma das principais características das árvores de decisão é que, a cada variável selecionada, o teste realizado particiona o conjunto de treinamento. O exemplo hipotético mostrado na Figura 3.3 ilustra este mecanismo. Considere um conjunto de treinamento S composto de 30 exemplares designando a classe X e 30 exemplares designando a classe Y. Cada exemplar é composto de três variáveis a , b , c . Para efeito de simplificação, é necessário considerar a existência implícita de uma técnica de seleção de variáveis a ser utilizada em cada ponto de teste T. Cada variável selecionada por esta técnica apresenta um valor de limiar, V , o qual divide a árvore em dois ramos (ou formando dois subconjuntos). Isto é feito através do teste de decisão: Sim (S) se a sentença no ponto de teste T for satisfeita ou Não (N) em caso contrário.

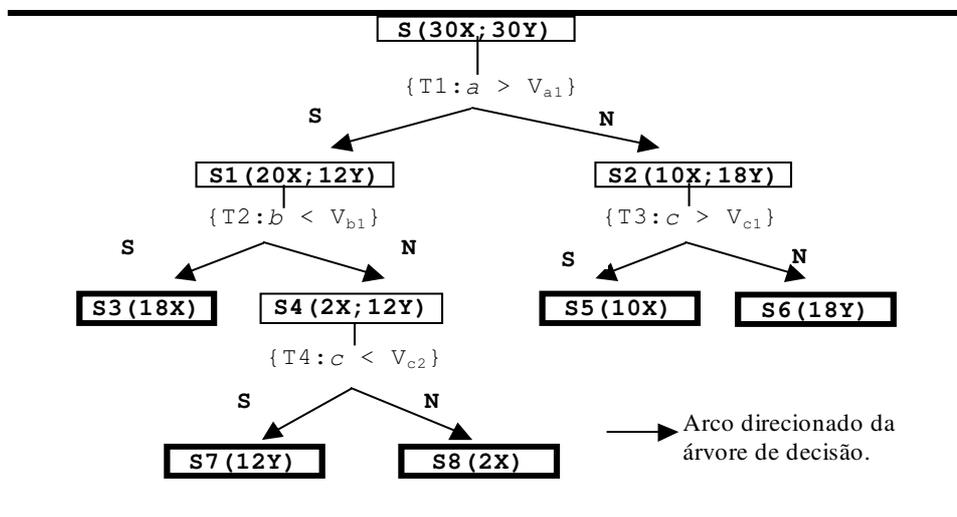


Figura 3.3 - Exemplo de uma árvore de decisão dividindo o conjunto de treinamento

De acordo com a Figura 3.3, o primeiro teste T1 realizado particiona o conjunto inicial em dois ramos. Considerando que a variável a é mais discriminatória e que V_{a1} é um valor desta que melhor particiona o conjunto S , então neste ponto são obtidas duas ramificações, formando os conjuntos S1 (se $a > V_{a1}$) e S2 (se $a \leq V_{a1}$). Observa-se que os dois conjuntos formados apresentam ainda exemplares pertencentes às duas classes X e Y. Então uma nova seleção de

variável deve realizada nos dois pontos de teste T2 e T3.

O teste T2 proporciona que uma folha (caixa em negrito) seja encontrada, ou seja, o conjunto S3 é formado somente por exemplares pertencentes à classe X. O outro ramo, formando o conjunto S4, ainda contém exemplares das duas classes e pode ser expandido. O teste T3, por sua vez, produz uma partição que gera duas folhas (subconjuntos S5 e S6), contendo somente exemplares de uma única classe cada um.

O último teste executado, T4, proporciona a obtenção das folhas referentes aos conjuntos S7 e S8. Desta forma, todas as folhas formadas neste exemplo contêm apenas exemplares de uma respectiva classe. Então, de uma forma geral, observa-se que conjunto inicial S foi dividido em 5 subconjuntos (S3, S5, S6, S7 e S8) contendo exemplares somente pertencentes a uma classe cada. Este fato mostra claramente o processo de partição seqüencial do conjunto de treinamento em uma árvore de decisão. Neste processo, nota-se também a informação discriminatória e individual que cada variável proporciona.

A classificação de um exemplar, como já dito, se faz estabelecendo um caminho do nó raiz até uma folha. No exemplo, pode-se então classificar os exemplares como sendo da classe X ou da classe Y da seguinte forma:

Caminho 1 (folha S3): $\{a > V_{a1} \text{ e } b < V_{b1}\} \rightarrow \text{classe X}$
 Caminho 2 (folha S5): $\{a \leq V_{a1} \text{ e } c > V_{c1}\} \rightarrow \text{classe X}$
 Caminho 3 (folha S8): $\{a > V_{a1} \text{ e } b < V_{b1} \text{ e } c \geq V_{c2}\} \rightarrow \text{classe X}$
 Caminho 4 (folha S6): $\{a \leq V_{a1} \text{ e } c \leq V_{c1}\} \rightarrow \text{classe Y}$
 Caminho 5 (folha S7): $\{a > V_{a1} \text{ e } b \geq V_{b1} \text{ e } c < V_{c2}\} \rightarrow \text{classe Y}$

Observa-se então que a classificação de um exemplar por um ‘caminho’ é simbólico, pois a representação destes é interpretável por um especialista humano. Esta representação pode ser melhorada e facilmente transformada em regras de produção, as quais podem ser escritas da seguinte forma:

Regra 1: **Se** $((a > V_{a1}) \text{ e } (b < V_{b1}))$ **Então** classe X
 Regra 2: **Se** $((a \leq V_{a1}) \text{ e } (c > V_{c1}))$ **Então** classe X
 Regra 3: **Se** $((a > V_{a1}) \text{ e } (b < V_{b1}) \text{ e } (c \geq V_{c2}))$ **Então** classe X
 Regra 4: **Se** $((a \leq V_{a1}) \text{ e } (c \leq V_{c1}))$ **Então** classe Y
 Regra 5: **Se** $((a > V_{a1}) \text{ e } (b \geq V_{b1}) \text{ e } (c < V_{c2}))$ **Então** classe Y

As regras extraídas da árvore mostram como o processo de aquisição do conhecimento é inerente à metodologia proposta para as árvores de decisão. Este é um das principais vantagens da utilização destas em AAC. Outra vantagem diz respeito à velocidade de processamento das

árvores. Esta é muito rápida, pois não envolve recursividade e a avaliação das variáveis é baseada em índices de fácil obtenção.

Entretanto, a forma individual e seqüencial como são obtidas as variáveis com maior capacidade de discriminação pode levar a resultados menos precisos, como observado no item 3.4. Além disso, se os caminhos até as folhas forem muito longos (i.e., maior número de variáveis) a interpretação do processo pode ficar comprometida. Este fato estimula que algumas técnicas utilizem *critérios de parada* para avaliar se um nó intermediário deve ser expandido ou não. Assim, este procedimento estabelece uma poda (*prunning*) da árvore. O objetivo da poda seria estabelecer uma relação ótima entre o tamanho da árvore gerada e a interpretação do processo. Por exemplo, um critério de parada poderia ser estabelecido de forma a encontrar que S4 não fosse expandido, pois há poucos exemplares pertencentes à classe X. Logo, S4 é seria considerado como uma folha.

Este procedimento auxilia para que seja evitado o sobre-ajuste (*overfitting*) para o resultado do treinamento pela diminuição do número de variáveis, podendo melhorar a capacidade de avaliar exemplares não observados no treinamento (generalização). Outra maneira de diminuir o sobre-ajuste do treinamento é realizar um processo de poda nas regras extraídas da árvore. A seguir são apresentadas as questões relacionadas a alguns algoritmos de árvore de decisão mais comumente utilizados em processos de *Data Mining*.

Algoritmos de Árvores de Decisão

Um algoritmo muito comumente utilizado é o ID3 (*Interactive Dichotomizer Tree*) ([MIT97],[SES94]). Em um dado nó da árvore, considere que S seja o subconjunto de treinamento (no nó raiz, S é o conjunto completo de treinamento). Uma maneira de obter uma avaliação de variáveis no ponto de teste é verificar o ganho de informação obtido se uma ramificação ocorrer devido a escolha de um atributo A . Este ganho é dado por:

$$I_A(S) = H(S) - H_A(S) \quad (3.1)$$

onde $H(S)$ é a *entropia a priori* e $H_A(S)$ é a *entropia a posteriori média*. A medida $H(S)$ fornece o grau de impureza de S . Isto é, dado que S contém n exemplares, quanto maior o número de classes contidas em S , mais impuro é este conjunto. Esta medida é dada por:

$$H(S) = \sum_{k=1}^c -P(C_k) \log_2 P(C_k) \quad (3.2)$$

com $P(C_k)$ é a probabilidade *a priori* da classe k ($k=1, \dots, c$ classes). Exemplificando, considerando que o conjunto de treinamento S contém 17 exemplares que designam duas classes, X e Y. Hipoteticamente, a distribuição deste em relação às classes é $S(11X,6Y)$. Logo, $P(C_X) = 11/17$ e $P(C_Y) = 6/17$. Aplicando-se (3.2), obtém-se então que $H(S)$ é igual 0.9367. A medida $H_A(S)$ é obtida através da seguinte expressão:

$$H_A(S) = \sum_{i=1}^v \sum_{k=1}^c P(A_i) H_k(A_i) \quad (3.3)$$

onde A_i denota os v valores que a variável A pode assumir. Então $P(A_i)$ é a probabilidade *a priori* que um valor A_i pode assumir no conjunto S . Para o conjunto S anterior e considerando que A pode assumir os valores $A1$ e $A2$ ($v = 2$), a seguinte distribuição ocorre em S para os valores de A : $S_{A1}(6X,4Y)$ e $S_{A2}(5X,2Y)$. Logo, $P(A1) = 10/17$ e $P(A2) = 7/17$. A grandeza $H_k(A_i)$ denota a entropia, calculada por (3.2), para cada valor que A pode assumir para uma dada classe k . Assim, para o exemplo tem-se que:

$$\begin{aligned} H_X(A1) &= -6/10 \log_2 6/10 = 0.4422 \\ H_Y(A1) &= -4/10 \log_2 4/10 = 0.5288 \\ H_X(A2) &= -5/7 \log_2 5/7 = 0.3467 \\ H_Y(A2) &= -2/7 \log_2 2/7 = 0.5164 \end{aligned}$$

Desta forma, o resultado obtido por (3.3) é:

$$\begin{aligned} H_{A1}(S) &= P(A1) (H_X(A1) + H_Y(A1)) = 0.5711 \\ H_{A2}(S) &= P(A2) (H_X(A2) + H_Y(A2)) = 0.3554 \\ H_A(S) &= H_{A1}(S) + H_{A2}(S) = 0.9265 \end{aligned}$$

E calculando-se agora o ganho da informação proporcionado pela variável A , através de (3.1), tem-se que este resultado é:

$$I_A(S) = H(S) - H_A(S) = 0.9367 - 0.9265 = \underline{\underline{0.0102}}$$

Este procedimento acima descrito deve ser realizado para todas as variáveis que fazem parte do

conjunto de treinamento. Finalizando, é escolhida a variável com maior valor de ganho, proporcionando a partição do conjunto S neste ponto de teste.

O exemplo acima mostra a maneira como o ID3 lida com dados binários e discretos. Para lidar com dados contínuos em ID3, é necessário discretizar os valores em faixas. Entretanto, o maior problema disto é exatamente definir as faixas corretas para enquadrar os valores contínuos. Por exemplo, supondo que um valor limite V de uma variável *temperatura* seja obtido em 75°C, o qual divide o conjunto de treinamento em duas faixas. Logo, valores como 70°C e 10°C, pertencentes à faixa inferior (mesma classe), são valores bem distantes um do outro, podendo não refletir a verdade sobre o problema.

Além do problema em lidar com atributos contínuos, ID3 não está livre de problemas relacionados com ruídos presentes nos dados. Este não responde bem a esta classe de problemas. Outra questão diz respeito ao fato de ID3 possibilitar a obtenção de nós *nulos* e *folhas deadend*. No primeiro, a este nó não está associada qualquer classe e o segundo, há exemplares de classe diferentes presentes no subconjunto sem que seja possível continuar a expansão, refletindo uma classificação não discriminatória dos exemplares.

O algoritmo ID3 pode produzir árvores grandes e com ramos muitos pequenos, dos quais podem estar associados pouco exemplares. Isso acontece porque ID3 trabalha fazendo a varredura global no espaço das hipóteses. Conseqüentemente, estas árvores podem ser incompreensíveis aos seres humanos. Este problema pode ser atenuado propondo-se um adequado critério de parada ou de poda da árvore, como já dito.

Existem na literatura algumas propostas de refinamento para o ID3. A principal delas é o algoritmo C4.5. A primeira modificação introduzida diz respeito à tendência natural em (3.1) de se privilegiar variáveis com muitos valores. Isto acontece porque a variável neste caso divide o conjunto de treinamento em subconjuntos muito pequenos. A conseqüência é que esta variável tem um ganho de informação muito alto. Neste caso, estas variáveis seriam primariamente escolhidas (no nó raiz) e definiriam árvores que perfeitamente classificam os exemplares de treinamento [MIT97]. Mas, quando isso acontece, a árvore perde em capacidade de enxergar exemplares não observados. Para se evitar isso, a idéia é penalizar variáveis que ocasionem este tipo de inconveniente através de uma *relação de ganho*. Uma forma é medir o quão ampla e uniformemente uma variável divide os dados. Isto é dado por:

$$SG_A(S) = - \sum_{i=1}^v P(A_i) \log_2 P(A_i) \quad (3.4)$$

Observa-se que (3.4) é a entropia de S em relação aos valores de A . A *relação de ganho* pode então ser definida como:

$$GR_A(S) = \frac{I_A(S)}{SG_A(S)} \quad (3.5)$$

Para o exemplo do conjunto S anterior e aplicando-se (3.4) e (3.5), tem-se:

$$SG_A(S) = (10/17) \log_2(10/17) + (7/17) \log_2(7/17) = 0.9774$$

Logo, a *relação de ganho* para a variável A é:

$$GR_A(S) = 0.0102/0.9774 = 0.01044$$

Comparando-se agora para a condição em que A tenha agora três valores, $A1$, $A2$ e $A3$ e a seguinte distribuição em S : $\{S_{A1}(4X, 1Y); S_{A2}(2X, 3Y); S_{A1}(5X, 2Y)\}$. O ganho $I_A(S)$ calculado para esta nova condição é de 0.0834. Observa-se então que o ganho cresceu muito em relação à condição com 2 valores de A . Calculando-se $SG_A(S)$ para a nova condição tem-se o resultado igual a 1.5657. A *relação de ganho* $GR_A(S)$ obtida é então de 0.0533. Por este resultado, observa-se como a *relação de ganho* procura compensar a tendência de se privilegiar variáveis com muitos valores. Essa implementação permite ao C4.5 lidar com dados contendo ruídos.

De uma forma geral, (3.4) mostra que se um conjunto tem n exemplares e existem $v = n$ valores de A , então (3.4) é máximo ($\log_2 n$). Se houver uma segunda variável B , binária, sendo avaliada e que divide o mesmo conjunto de n exemplares em dois subconjuntos iguais, então (3.4) é obtida igual a 1 para esta variável. Hipoteticamente, se A (exemplo anterior) e B tiverem o mesmo valor de ganho ($I_A(S)$), então B , que apresenta a menor medida dada por (3.4), terá maior *relação de ganho*, i.e. apresenta divisões menos uniformemente distribuídas.

A segunda modificação introduzida no ID3, que resultou no C4.5, é a introdução de uma metodologia que permite lidar com dados contínuos. A técnica envolve escolher o ponto médio entre dois valores consecutivos como o valor limite (*threshold*), o qual divide o conjunto de treinamento. Se R for um atributo contínuo com R_1, \dots, R_n valores, então o valor limite V para cada intervalo é $(R_1+R_2)/2, \dots, (R_{n-1}+R_n)/2$. Observa-se que podem ser testados $n-1$ valores limites. O valor V que produz o melhor valor do critério de divisão, dado pela Equação 3.1, é então selecionado.

Entretanto, a técnica acima descrita para se encontrar o valor limite V pode produzir sub-árvores que classificam pouco ou nenhum exemplar. Se for aplicações onde há presença de ruído nos dados, então o nível de complexidade da árvore resultante aumenta bastante. Obviamente, nestes casos, um atributo deve possuir mais valores V distintos e tende a ser escolhido prioritariamente por causa desta influência. A referência [QUI96] propõe resolver este problema verificando o ganho aparente de cada valor V de uma dada variável. O resultado pode ser a diminuição da quantidade do número de valores V encontrados e conseqüentemente redução no número de testes potenciais a serem efetuados. Conseqüentemente, há uma redução na importância desta variável em relação às variáveis discretas e ou com poucos valores V .

O algoritmo C4.5 proposto (www.rulequest.com) utiliza também um critério para poda da árvore. Este é implementado cortando-se sub-árvores que pouco adicionam na precisão da árvore global, com o objetivo de produzir menores árvores e, portanto, mais interpretáveis. Este processo é executado diretamente nas regras extraídas. Inicialmente, são construídas regras para cada caminho, do nó raiz a um nó folha. Isto é feito para que a retirada de antecedentes que não ocasione uma piora na precisão da classificação.

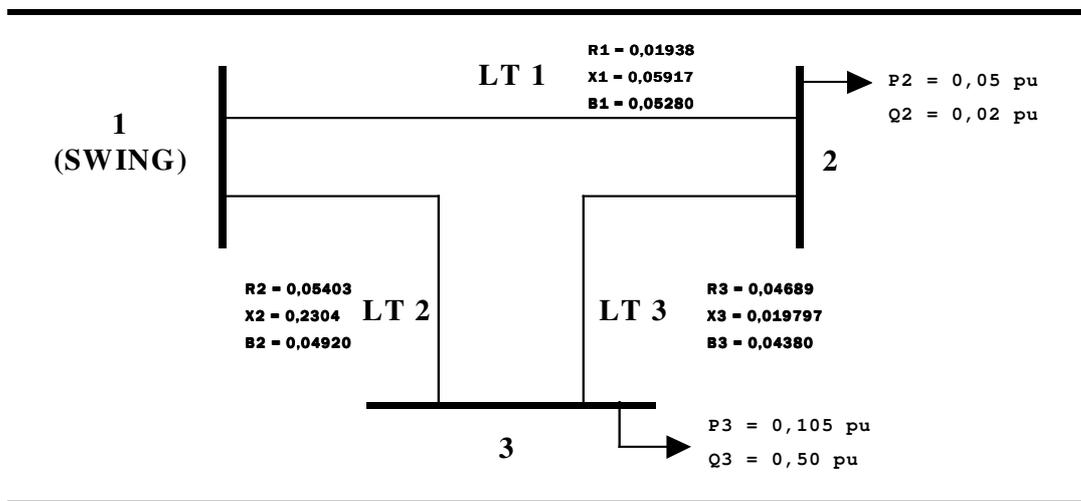


Figura 3.4 – Sistema elétrico com 3 barras.

A Figura 3.4 mostra o diagrama unifilar para o sistema elétrico que exemplifica a aplicação do C4.5 para um sistema elétrico teste contendo 3 barras. Este sistema elétrico contém 2 barras de carga, barra *swing* e apresenta 3 linhas de transmissão. Para o sistema, é considerada a tensão na barra 1 (*swing*) igual a 1,0 pu e ângulo desta igual a zero. A base de dados, formada através da metodologia proposta no Capítulo 2, é mostrada através das Tabelas 3.1 e 3.2, onde esta é dividida em dois conjuntos, treinamento e teste.

Tabela 3.1 – Conjunto de treinamento para o sistema elétrico 3 barras.

SISTEMA ELÉTRICO 3 BARRAS					
Exemplar	V2	V3	ZI2	ZI3	Classe
01	0.936	0.973	1.229	2.564	Normal
02	0.916	0.966	0.901	2.090	Normal
03	0.899	0.962	0.806	2.121	Normal
04	0.908	0.964	0.858	2.115	Normal
05	0.894	0.956	0.748	1.542	Normal
06	0.885	0.953	0.698	1.472	Normal
07	0.828	0.932	0.440	1.187	Alerta
08	0.830	0.932	0.467	1.119	Alerta
09	0.850	0.938	0.527	1.251	Alerta
10	0.797	0.921	0.366	1.045	Alerta
11	0.761	0.905	0.313	0.880	Alerta
12	0.761	0.909	0.308	0.971	Alerta
13	0.690	0.887	0.216	0.902	Inseguro
14	0.674	0.882	0.202	0.880	Inseguro
15	0.662	0.878	0.193	0.865	Inseguro
16	0.648	0.874	0.183	0.849	Inseguro
17	0.680	0.883	0.206	0.878	Inseguro
18	0.671	0.881	0.199	0.867	Inseguro
19	0.597	0.859	0.150	0.795	Inseguro

Tabela 3.2 – Conjunto de teste para o sistema elétrico 3 barras.

SISTEMA ELÉTRICO 3 BARRAS					
Exemplar	V2	V3	ZI2	ZI3	Classe
01	0.928	0.972	1.079	2.557	Normal
02	0.880	0.951	0.629	1.643	Normal
03	0.877	0.950	0.615	1.616	Normal
04	0.874	0.949	0.601	1.590	Normal
05	0.887	0.953	0.708	1.485	Normal
06	0.858	0.941	0.555	1.287	Normal
07	0.835	0.935	0.456	1.212	Alerta
08	0.849	0.938	0.525	1.189	Alerta
09	0.773	0.913	0.327	0.998	alerta
10	0.752	0.906	0.296	0.953	alerta
11	0.814	0.926	0.398	1.153	Alerta
12	0.790	0.921	0.347	1.203	Alerta
13	0.711	0.894	0.248	0.895	inseguro
14	0.701	0.890	0.225	0.917	inseguro
15	0.685	0.885	0.211	0.895	inseguro
16	0.680	0.884	0.207	0.887	inseguro
17	0.640	0.872	0.178	0.841	inseguro
18	0.632	0.869	0.172	0.833	inseguro
19	0.629	0.868	0.169	0.822	inseguro

Tabela 3.3 – Árvore de decisão obtida pelo C4.5 para o Caso Base.

V2 <= 0.69: inseguro
V2 > 0.69:
 V3 <= 0.938: alerta
 V3 > 0.938: normal

Tabela 3.4 - Taxa de classificação da árvore de decisão extraída pelo C4.5. Os valores sublinhados denotam classificação correta. – Caso base sistema teste 3 barras.

Normal	Alerta	Inseguro	Total
<u>06</u>	0	0	06
0	<u>06</u>	0	06
0	02	<u>05</u>	07
Total	Class. Correta		Class. Errada
Geral	<u>89.47% (17/19)</u>		<u>10.50% (02/19)</u>

As regras extraídas da árvore de decisão obtida pelo C4.5 são mostradas na Tabela 3.5 e a taxa de classificação para o conjunto de teste destas regras é dada pela Tabela 3.6.

Tabela 3.5 - Regras C4.5 - Caso base sistema teste 3 barras.

```

Regra 1:
V3 > 0.938
-> classe normal
Regra 2:
V2 > 0.690
V3 <= 0.938
-> classe alerta
Regra 3:
V2 <= 0.690
-> classe alerta
    
```

Tabela 3.6 - Taxa de classificação das regras do C4.5. – Caso base sistema teste 3 barras.

Normal	Alerta	Inseguro	Total
<u>06</u>	0	0	06
0	<u>06</u>	0	06
0	02	<u>05</u>	07
Total	Class. Correta		Class. Errada
Geral	<u>89.47% (17/19)</u>		<u>10.50% (02/19)</u>

Neste caso, observa-se que a taxa de classificação das regras é igual a taxa de classificação da árvore. Entretanto, este fato não é uma sistemática do C4.5 devido ao processo de extrair regras e de poda implementado. Entretanto, a taxa de classificação explicitada pela Tabela 3.6, é obtida, segundo os autores da versão de demonstração utilizada, através de uma ‘competição’ estatística para verificar qual a regra classifica melhor cada exemplar. Assim, nota-se que um exemplar só é classificado uma vez, seja pela regra correta ou não. Para efeito de comparações feitas neste trabalho, é interessante observar as regras individualmente. O resultado disso é que um exemplar pode ser classificado por mais de uma regra (*superposição* de

classificação). O resultado encontrado para a taxa de classificação individual pode ser visualizado na Tabela 3.7, mostrando que o número de classificações erradas aumenta.

Tabela 3.7 – Taxa de classificação *por regra* do C4.5 para o sistema teste de 3 barras

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>06</u>	01	0	100.00% (06/06)	7.69% (01/13)
02	0	<u>06</u>	02	100.00% (06/06)	15.38% (02/13)
05	0	0	<u>05</u>	71.43% (05/07)	0.00% (0/12)
Total Geral	Class. Correta			Class. Errada	
	<u>89.47% (17/19)</u>			<u>15.79% (03/19)</u>	

Além dos algoritmos de árvore de decisão ID3 e C4.5, outros são observados a seguir. O algoritmo denominado de CART (*Classification and Regression Trees*) ([CHE98][SES94]) também proporciona lidar com dados contínuos da mesma forma que C4.5. Difere do C4.5 na aplicação do teste para expandir o nó. Neste caso, o critério de divisão (*splitting criterion*), também é baseado em uma medida de impureza. A referência [CHE98] mostra três critérios para esta medida de impureza: Função Gini, Função Entropia e uma função custo de erro de classificação. Em CART é aplicado um algoritmo de poda que é baseado numa função de custo-complexidade mínima, onde a idéia é obter uma relação precisão/simplicidade ótima para a árvore.

A referência [WEH93] utiliza um modelo refinado para o ID3, denominado de TDIDT, onde são incorporadas algumas soluções encontradas em outros algoritmos. Este algoritmo lida com dados contínuos como no C4.5, ordenando os valores limites em ordem crescente como o C4.5. Entretanto, o cálculo de critério de divisão (*ganho*) é feito de forma alternativa (e mais eficiente, segundo os autores) à proposta pela Equação 3.1.

O modelo TDIDT leva em conta também que folhas *deadend* são produzidas. Isto proporciona estabelecer um critério de parada, onde é utilizado um teste de hipótese estatística denominado de Chi-Quadrado (χ^2). Os autores propõem estabelecer também uma relação entre a simplicidade versus confiabilidade da árvore, podendo então ser quantificada esta relação de complexidade. Baseados nesta quantidade, dois métodos de poda são então propostos. Conforme já observada no Capítulo 01, esta técnica TDIDT é utilizada para proceder avaliação de segurança dinâmica e de tensão.

3.6 CONCEITOS BÁSICOS SOBRE REDES NEURAS ARTIFICIAIS

Esta seção tem o objetivo de auxiliar o entendimento do paradigma neural e conseqüentemente o processo de extração do conhecimento simbólico. Partindo-se dos conceitos básicos sobre *funções discriminantes* (Apêndice B), o entendimento a respeito do mapeamento feito por uma rede neural pode ser melhor entendido. Isto é, uma rede neural é uma maneira alternativa de se manipular funções discriminantes. A referência [BIS95] apresenta uma introdução à abordagem neural, partindo-se das funções discriminantes lineares multiclases.

A Figura 3.5 mostra uma rede neural artificial sem camada intermediária, onde cada saída $g(\mathbf{x})$ está relacionada a um vetor peso w_{ik} e um *bias* w_{0k} . Ao *bias* w_{0k} está associado uma entrada adicional x_0 , que é sempre igual a +1.

Sob este enfoque neural, tem-se a seguinte função discriminante:

$$g_k(\mathbf{x}) = \sum_{i=1}^d w_{ik} x_i + w_{0k} \quad (3.6)$$

onde d é o número de atributos pertencentes ao vetor de entrada e c o número de classes (Figura 3.5).

Após a rede neural ser treinada de acordo com (3.6), um novo exemplar \mathbf{x} é classificado através da aplicação deste à entrada da rede, devendo então ser observado qual saída g apresenta o maior valor (neste caso, denominado de maior ativação). Nota-se que isto corresponde à regra de decisão dada pela equação B.17 (Apêndice B). A conseqüência disso é que duas regiões de decisão só conectadas por uma única fronteira linear. Este fato restringe a atuação das funções discriminantes apenas a problemas linearmente separáveis.

Em [BIS95] é argumentado que, as RNAs sem camadas intermediárias, como na Figura 3.5, possibilitam lidar com problemas não-linearmente separáveis (por exemplo, a função OU-Exclusivo). Embora haja uma sobreposição significativa entre as classes, fruto da natureza destes problemas, estes tipos de RNAs possibilitam classificar, com a menor taxa de erro, padrões ainda não observados. Isto acontece pois, embora os padrões de entrada do conjunto de treinamento não sejam adequadamente separados, a fronteira de decisão ótima resultante (linear) proporciona boa capacidade de generalização (respostas aos novos padrões). Algumas vezes, uma análise do desempenho do classificador versus o treinamento mais simples e rápido das RNAs-SL possibilita decidir por estas em vez de modelos mais complexos de RNAs.

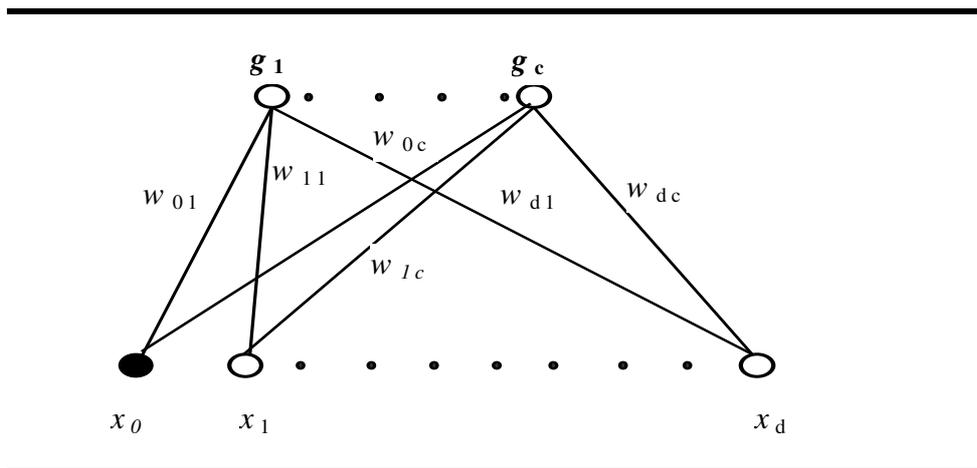


Figura 3.5- Representação neural da função discriminante linear $g(x)$ para um problema multiclass

A Equação B.32, apresentada no Apêndice B, pode ser rescrita da seguinte forma:

$$g_k(x) = f\left(\sum_{i=1}^d w_{ik}x_i + w_{0k}\right) \tag{3.7}$$

onde $f(.)$ é denominada de função de ativação e geralmente é monotônica. Observa-se que, se $f(.)$ tiver uma característica linear ($f(xw) = xw$), a Equação 3.7 recai novamente na equação B.32. Logo, a equação B.32 pode ser considerada um caso particular de (3.7). O Apêndice C mostra dois tipos de função de ativação: a função *threshold* e a sigmóide. A Equação 3.7, utilizada com a função *threshold*, é denominada de função *Perceptron*.

Observa-se que a estrutura da rede não muda em função da modificação da função de ativação. Um algoritmo bastante utilizado para treinamento de uma RNA com uma camada, como mostrada na Figura 3.5, é a Regra de Hebb. Este algoritmo (Apêndice C) tem característica linear e permite treinar a rede de forma a encontrar uma adequada relação entre os atributos de entrada e uma dada saída.

Se a função $f(.)$ é não-linear e monotônica (por exemplo, a sigmoidal), a função discriminante $g_k(x)$ dada por (3.7) ainda continua linear, uma vez que a fronteira de decisão ainda se mantém linear, próprio da natureza monotônica da função $f(.)$. Logo, para que uma RNA do tipo *feedforward* possa lidar com problemas não linearmente separáveis, camadas intermediárias de neurônios devem ser introduzidas, onde cada saída de um neurônio destas camadas é dada por (3.7). Entretanto, para que o mapeamento produzido pela rede seja eficaz, a função de ativação desta camada intermediária deve também ser não-linear. Pode ser notado que, se a função de

ativação para estes neurônios for linear, a combinação de transformações lineares produz também uma transformação linear. Logo, o modelo com camadas intermediárias se torna equivalente ao modelo sem camadas intermediárias.

Se o mapeamento neural é feito com função de ativação sigmoidal nos neurônios das camadas intermediárias, então isto produz uma aproximação funcional não-linear para um amplo espectro de funções, sendo então considerado um *aproximador funcional universal* ([SCH97],[CHE98]). Em problemas de classificação, a rede neural fornece a aproximação funcional para qualquer fronteira de decisão. Em [BIS95] é argumentado que, tendo as redes neurais esta característica, então a aproximação funcional proporcionada por esta pode ser utilizada para a estimar a probabilidade (*a posteriori*) condicional de classe.

A introdução das camadas intermediárias pode gerar um mapeamento tão complexo quanto o do modelo polinomial multivariável (Apêndice B), produzindo regiões de decisões não-convexas e disjuntas. No modelo neural, o número de graus de liberdade é dado pela quantidade de neurônios presentes em cada camada intermediária. Entretanto, uma melhora significativa é introduzida pelo mapeamento neural. Enquanto o número de graus de liberdade cresce exponencialmente com o aumento da dimensionalidade para o modelo polinomial, no modelo neural *feedforward* multicamadas este cresce apenas linearmente, refletindo no requerimento de menos exemplares.

O preço pago pela redução do problema da dimensionalidade é que o ajuste dos pesos (treinamento da rede) torna-se necessariamente um problema de otimização não-linear e, portanto, sujeito à obtenção de vários mínimos locais. Na literatura ([HAS95], [BIS95]) várias técnicas são propostas para minimização da função erro. Uma técnica muito utilizada, uma vez que a função de ativação é diferenciável, é a descida em gradiente com uma estratégia de aprendizado baseada na retropropagação de erros (RNA-ML - Apêndice C). As referências [HAS95] e [BIS95] também discutem outras técnicas de otimização como *gradiente conjugado*, *Método Quasi-Newton* e *algoritmo de Levenberg-Marquardt*.

Outro tipo de modelo de RNAs *feedforward* são as baseadas em treinamento por batelada. Neste tipo enquadram-se as redes OET e OET2 [SIL91]. Os aspectos teóricos destas são apresentados no Apêndice C. Uma das principais vantagens da utilização deste tipo de paradigma neural é a capacidade de lidar com conjuntos de treinamento com uma cardinalidade muito grande, pois apresenta uma velocidade de treinamento muitíssimo maior que as redes neurais treinadas a base de métodos iterativos (e.g., descida em gradiente). Além dessas vantagens, apesar dos algoritmos baseados em métodos iterativos oferecerem melhor controle da

capacidade de generalização do modelo neural, a relação custo/benefício não se justifica para grandes bases de dados.

Como observado, as RNAs tipo *feedforward* multicamadas são aproximadores universais. Além dessa característica, estas são dispositivos muito robustos em lidar com dados ruidosos e proporcionam boa capacidade de generalização. Entretanto, para muitos problemas, apenas um erro baixo de classificação ou estimação não é suficiente. Geralmente, associado ao processo de aprendizado para classificar ou estimar bem, existem questões que não podem ser explicitamente observadas, e estas podem proporcionar um maior grau de interpretação do processo. Na literatura ([WEH97]), tem-se argumentado que as redes neurais apresentam uma característica tipo *caixa-preta* e que, devido a isso, não são adequadas à tarefa de expressar o conhecimento em forma interpretável. Este argumento está relacionado ao fato de que este conhecimento se distribui pelos pesos da rede ou nas funções de ativação. A observação por si só destes valores não fornece qualquer interpretação. Entretanto, os pesos das conexões podem ser manipulados de tal forma a extrair este conhecimento, devendo ser representado em algum formato interpretável.

A questão torna-se então como extrair o conhecimento sintetizado pelos pesos ou saída das funções de ativação da rede e representa-lo em formato útil. Um dos formatos bastante utilizados é o de *regras de produção* (IF-THEN). Em aprendizado automático, alguns modelos (denominados de *conexionistas*) são propostos com esta finalidade.

A seguir serão mostrados os conceitos básicos de como proceder a extração de regras simbólicas das redes neurais bem como as idéias principais de alguns algoritmos utilizados para esta finalidade. Ao final, será mostrado o desenvolvimento teórico do algoritmo VIA, utilizado neste trabalho como o agente para se obter regras de produção para a formação da base conhecimento em avaliação da segurança de tensão.

3.7 EXTRAÇÃO DE REGRAS DE PRODUÇÃO DE REDES NEURAS ARTIFICIAIS

Uma maneira fácil de visualizar um processo de extração de regras é mostrar como uma rede neural com uma camada e uma única saída se comporta em relação à codificação do conhecimento. A idéia é que um conjunto finito de regras de produção SE-ENTÃO (*IF-THEN*) descreva, qualitativamente, o conhecimento sintetizado pela rede neural. As regras especificam que, dada uma condição de entrada, a implicação indica o estado esperado para esta condição. A

Figura 3.6 mostra esta rede, onde $X=\{x_1, x_2, x_3\}$ representa os vetor contendo as entradas do modelo, $W=\{w_1, w_2, w_3\}$ é o vetor com os pesos das conexões, $f(.)$ é a função de ativação aplicada ao somatório, resultando na saída y . O parâmetro θ é um valor de limiar com valor zero neste caso.

Considerando $f(.)$ uma função linear, tal que

$$y_k = \sum w_{ik} x_i + \theta \tag{3.8}$$

uma regra de decisão (que pode ser caracterizada por uma função do tipo “degrau” – *threshold*) pode ser arbitrada para mapear a saída da rede em classes diferentes. Desconsiderando o índice k , uma vez que só há uma saída, tem-se que:

$$y' = \begin{cases} A & \text{se } y > 0 \\ B & \text{se } y < 0 \\ desconhecido & \text{se } y = 0 \end{cases} \tag{3.9}$$

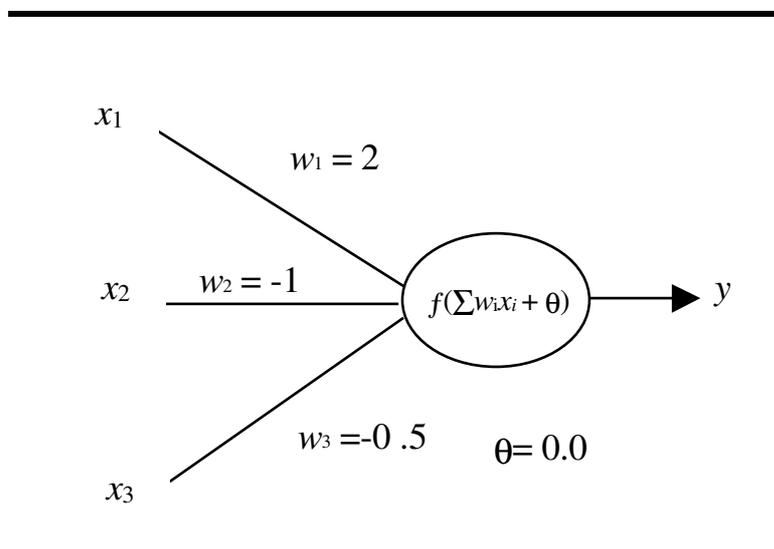


Figura 3.6 – Modelo de um neurônio com função de ativação linear

A Tabela 3.8, mostra o mapeamento das variáveis de entrada com valores binários 0 ou 1. Aplicando-se $f(.)$ a esta tabela, obtém-se os valores da ativação y e conseqüentemente, através da regra de decisão mostrada acima, a classificação y' .

Tabela 3.8 – Mapeamento binário das variáveis

x_1	1	1	1	1	0	0	0	0
x_2	1	1	0	0	1	1	0	0
x_3	1	0	1	0	1	0	1	0
y	0.5	1.0	1.5	2.0	-1.5	-1.0	-0.5	0
y'	A	A	A	A	B	B	B	---

Observando a classificação obtida, pode-se então obter regras SE-ENTÃO que expressam o mapeamento acima. Três regras gerais são extraídas:

R1: Se $(x_1 = 1)$ então classe **A**

R2: Se $\{(x_1 = 0 \wedge x_2 = 1 \wedge x_3 = 0) \vee (x_1 = 0 \wedge x_2 = 0 \wedge x_3 = 1) \vee (x_1 = 0 \wedge x_2 = 1 \wedge x_3 = 1)\}$ então classe **B**

R3: Se $(x_1 = 0 \wedge x_2 = 0 \wedge x_3 = 0)$ então Desconhecido

onde os operadores \wedge e \vee representam conjunção e disjunção, respectivamente.

As três regras obtidas são regras gerais e refletem o conhecimento a respeito do mapeamento dado. Observa-se que a regra R1 contém apenas um antecedente, a variável x_1 . Ou seja, a observação do valor ‘1’ para esta variável por si só discrimina a classe A. Entretanto, a classe B não pode ser completamente descrita pelo valor zero da variável x_1 . Isto reflete que o conhecimento a respeito desta classe não está tão bem definido nos dados, necessitando de maior número de antecedentes e suas respectivas condições para mapear corretamente esta classe. Vale salientar que estas regras são mutuamente exclusivas e plenamente discriminatórias, i.e., a regra R1 não classifica qualquer exemplar pertencente à classe B e vice-versa.

Para uma rede neural com uma camada, como a apresentada aqui, o processo de extração de regras se mostra simples mas este processo torna-se problemático quando se necessita lidar com RNAs multicamadas . De uma forma geral, as metodologias para extração de regras de RNAs podem ser agrupadas segundo cinco critérios básicos [CRA96, TIC98] . Estes são:

1. **Formato da regra:** Este critério indica qual é o formato que o algoritmo de extração de regras expõe o conhecimento adquirido. Existem basicamente três tipos: convencional, que são as regras booleanas; regras baseadas em lógica e conjuntos difusos e, regras na forma de lógica de primeira ordem, cujo formato apresenta e variáveis e intervalos (contínuos) de valores das mesmas.
2. **Qualidade das regras:** Apresentam quatro quesitos a serem avaliados. Estes são:
 - a. **Precisão:** avaliação quanto à capacidade de generalizar.

- b. Fidelidade:* expressa a capacidade do algoritmo de reproduzir de forma precisa o modelo da RNA, expressada através das regras extraídas.
 - c. Consistência:* sob diferentes seções de treinamento da RNA, o algoritmo gera um conjunto de regras que produz sempre a mesma taxa de classificação para exemplares não observados.
 - d. Compreensibilidade:* é função do tamanho do conjunto de regras e do número de antecedentes por regra.
3. **Translucidez da técnica:** Define o modo de extração de conhecimento da RNA. Para definir o grau de translucidez de uma técnica, são utilizadas três subdivisões: decomposicional, eclética ou pedagógica. Decomposicional significa que as regras são extraídas a partir dos neurônios da camada oculta e de saída e é o de maior grau de translucidez. No lado oposto, uma técnica é dita pedagógica quando a RNA é vista como uma *caixa-preta* e as regras são extraídas das relações diretas entre entradas e saídas. Finalizando, uma técnica eclética analisa as unidades da rede individualmente, mas as regras são agrupadas para que as relações finais sejam extraídas destas.
4. **Complexidade da técnica:** Esta análise é mais difícil devido ao fato de poucos autores tratarem deste problema. Entretanto, se uma regra é extraída por neurônio, por exemplo, pode-se dizer que sua complexidade é linear em relação ao número de neurônios. Se o procedimento é testar combinações de entradas, a complexidade pode ser exponencial, fruto das inúmeras possibilidades de combinações.
5. **Portabilidade da técnica:** Esta análise possibilita averiguar a utilização da técnica com múltiplas arquiteturas, metodologias de treinamento e diversos tipos de funções de ativação. Isto significa olhar se uma dada técnica foi desenvolvida para um tipo de RNA em particular.

Os critérios acima destacados propõem um esquema *objetivo* para avaliar as regras extraídas e o respectivo algoritmo implementado com esta finalidade. Entretanto, quando se trabalha com uma ferramenta para mineração de dados (*Data Mining*), a relação entre o conhecimento simbólico explicitado e o problema em análise é importante. Isso significa que, além de atender o critério da *fidelidade*, as correlações entre as variáveis de entrada, e as correlações destas com as classes de interesse (variáveis de saída) devem ser tal que o fenômeno em estudo seja interpretado através das próprias regras. Logo, o critério de *compreensibilidade* pode ser entendido como a interpretação *subjetiva* das regras. Entretanto, esta análise requer a presença de um especialista para a devida avaliação.

Segundo [MIT02], essa análise subjetiva é importante de acordo com dois pontos de vista: surpresas nas regras encontradas e a utilidade das regras extraídas. O autor relata também a possibilidade de ambos pontos de vista estarem correlacionados em alguns problemas. Sob este ponto de vista, pode-se introduzir o conceito de *novidade* para o critério da qualidade das regras. Ou seja, deve-se observar também a presença de regras que não reflitam meramente o conhecimento já estabelecido a respeito de um problema e sim fatos novos, de importância para o problema em análise.

Reportando-se ao conjunto de regras extraídas para o exemplo anterior, observa-se que as regras refletem exatamente o comportamento da RNA treinada sob a base de dados, dada pela Tabela 4.1. Devido a natureza das entradas, neste caso binárias, as regras são extraídas a partir da cobertura de todas as possíveis combinações de entradas. Neste caso a metodologia empregada é *pedagógica*, uma vez que as regras são extraídas das relações diretas entre as entradas e a saída da RNA. Entretanto, caso a análise fosse feita para uma rede neural com um maior número de neurônios, considerando cada neurônio individualmente como no exemplo, então a metodologia seria *decomposicional*.

Algoritmos para extração de regras de redes neurais artificiais

Tem-se observado um número crescente de algoritmos desenvolvidos para se trabalhar com conhecimento simbólico e redes neurais artificiais. Estes algoritmos podem ser divididos em três categorias: 1) Técnicas de extração de regras propriamente dita; 2) Técnicas de refinamento de regras e, 3) Técnicas que utilizam o conhecimento *a priori* para inicializar uma rede neural. A análise empreendida a seguir mostra os principais algoritmos encontrados na literatura para a tarefa proposta. Entretanto, será dada maior ênfase nos algoritmos que extraem e refinam regras simbólicas de redes neurais supervisionadas tipo *feedforward*.

Um dos primeiros algoritmos desenvolvidos foi o *MACIE (Matrix Controlled Inference Engine)* [GAL88]. Este algoritmo justifica cada inferência por uma RNA, não constituindo na verdade um algoritmo para extração de regras que representem o conhecimento sintetizado pela RNA. Pode ser notada a semelhança deste com o exemplo didático apresentado acima, pois ambos são bastante básicos em alguns conceitos para aquisição de conhecimento utilizando uma rede neural. Este algoritmo primariamente lida com dados discretos. Neste senso discreto, um neurônio pode ter os valores verdadeiro (+1), falso (-1) e desconhecido (0). No caso da rede

neural utilizada pelo *MACIE*, a função de ativação u' aplicada ao discriminante linear $S_k = \sum w_{ik}x_i$ é a *função perceptron*, tal que, se $S_k > 0$, então $u' =$ verdadeiro ou $+1$, se $S_k < 0$, então $u' =$ falso ou -1 e, se $S_k = 0$, então $u' =$ desconhecido ou 0 .

O ponto de partida do algoritmo é a formulação da seguinte pergunta: por que um neurônio da camada oculta ou de saída é deduzido verdadeiro (ativação = $+1$)? A primeira providência é ordenar, em valores absolutos decrescentes, os pesos das conexões das entradas relativas ao neurônio k em análise. Os pesos aqui ordenados só se referem aquelas entradas cujo valor é conhecido (verdadeiro ou falso). Com esta informação é iniciado o processo de geração de uma regra SE-ENTÃO, tal que (3.10) seja satisfeita para o neurônio k :

$$\left\{ \sum_{i1=\text{sub-conjunto de entradas conhecidas}} |w_{i1}| \right\} > \left\{ \sum_{i2=\text{sub-conjunto da entradas remanescentes}} |w_{i2}| \right\} \quad (3.10)$$

Seja o seguinte exemplo. Considere 6 entradas (x_1, \dots, x_6) relativas ao neurônio k com ativação $u_j' =$ verdadeira, das quais 4 são conhecidas (verdadeiras ou falsas), tal que:

$$\begin{aligned} x_1 &= 1 ; x_2 = -1 ; x_3 = ? ; x_4 = ? ; x_5 = 1 ; x_6 = -1 \\ w_1 &= 1 ; w_2 = -8 ; w_3 = -6 ; w_4 = -1 ; w_5 = 4 ; w_6 = 5 \end{aligned}$$

Assim, os pesos são ordenados para as entradas conhecidas e as entradas cujos valores $x_i w_i$ são menores que zero são descartadas (neste caso, $x_6 w_6$). Isso é feito porque estas entradas não contribuem para tornar a saída da ativação verdadeira (+). Utilizando (3.10), os antecedentes das regras são escolhidos um por um até que a equação acima seja satisfeita. Logo, a regra extraída é:

$$\mathbf{SE} \{ (\text{entrada } x_2 \text{ é falsa}) \wedge (\text{entrada } x_5 \text{ é verdadeira}) \} \mathbf{ENTÃO} \{ \text{ativação } u_j' \text{ é verdadeira} \}.$$

Uma vez formulada a regra, pode-se avaliar se as variáveis deduzidas são suficientes para explicar a inferência neural. Caso contrário, pode-se retropropagar a regra de tal forma a deduzir alguma variável com valor desconhecido que auxilie neste processo. Considerando diferentes inferências, o conjunto de regras correspondentes pode constituir a base de conhecimento de um sistema especialista tradicional. Entretanto, o autor argumenta que *MACIE* só pode ser utilizado se a rede neural for pequena, fruto do crescimento exponencial do número de regras com o

aumento do número de entradas. Outra questão é que, como proposto, o algoritmo lida somente com dados discretos, o que restringe o campo de atuação do mesmo. Observa-se então que o *MACIE* é um algoritmo do tipo *eclético*, onde são analisados os elementos internos para se obter uma relação precisa entre as entradas e saídas.

O algoritmo *MofN*, apresentado em [TOW92] e [TOW92a], propõe formular regras do tipo: **SE** (*M* dos seguintes *N* antecedentes são verdadeiros) **ENTÃO** classe X. Por exemplo, se 2 antecedentes do conjunto {a,b,c} satisfazem a consequência da regra, então a regra é válida. Isso é o mesmo que regras disjuntivas do tipo **SE** {(a e b) ou (a e c) ou (b e c)} **ENTÃO** classe X. O processo é feito em 6 etapas:

1. Os pesos das conexões de cada unidade da camada oculta e de saída são agrupados de acordo com suas similaridades (*clusters*).
2. Cada peso é redefinido com o valor da média do grupo ao qual pertence.
3. Eliminação dos grupos de pesos que não afetam a ativação de cada neurônio.
4. Sob essa nova arquitetura, a RNA é treinada novamente para que para otimizar o *bias*.
5. Para cada unidade oculta e de saída é formada uma regra simples, cujo processo é mostrado através do exemplo a seguir. Considere o valor do *bias* igual a 10.0 após a aplicação do passo 4. Pelos passos 1, 2 e 3 foi encontrado um único grupo com 3 pesos. Seus respectivos valores foram redefinidos no passo 2 e obtidos iguais a 6.0. Suponha as variáveis relacionadas a estes pesos sejam A, D e F. Então é montada a seguinte regra simples: **Se** $6.0 \times \text{Var}(A,D,F) > 10.0$ **então** classe X. *Var* retorna o número de antecedentes verdadeiros.
6. A regra final é extraída na forma M-of-N. Para o exemplo do passo 5, a regra é: R1:**Se** 2 de {A D F} **então** classe X.

Os autores de *MofN* argumentam que, o conjunto de regras formado pode ter desempenho superior à própria rede neural a qual foram extraídas as regras. Isso acontece porque estas regras podem reduzir o sobre-ajuste do conjunto de treinamento. Entretanto, esse algoritmo pode gerar regras cuja explanação do mapeamento neural pelos antecedentes presentes nas regras é de difícil compreensão, pois a necessidade de novos treinamentos da RNA pode levar a inclusão de novos antecedentes às regras. Além disso, se os dados forem incompletos, o algoritmo pode não extrair com precisão um dado conceito. Outra questão é que, os autores não mostram se é possível trabalhar com dados contínuos.

A referência [MAI97] mostra uma maneira de sintetizar as regras extraídas pelo método *MofN* em regras menores. Isso é feito através conversão destas em vetores binários equivalentes, o qual a manipulação destes possibilita reduzir o tamanho das regras. A referência [SET00] apresenta um algoritmo de extração de regras do tipo M-of-N, o qual o autor denomina de *M-of-N3*. Este método é a evolução de um método do mesmo autor e apresentado em [TIC98], denominado de *RX*. Este método apresenta um esquema especial de treinamento de uma rede neural RNA-ML, onde são removidas conexões (poda) que não afetam a classificação. Além disso, o método prevê a transformação dos dados de entrada e dos pesos das conexões entrada-camada oculta em valores binários. Isso é feito através da obtenção de agrupamentos. Os resultados apresentados pelo autor mostram que este procedimento reduz o número de regras extraídas e a fidelidade das regras não é afetada pela metodologia. Exemplos de aplicação deste método são mostrados pelas referências [SAN99] e [HRU99].

A referência [FU094] apresenta um algoritmo denominado de *KT*. Este é constituído de duas fases: treinamento de uma RNA-ML e, obtenção a partir desta rede neural treinada de um conjunto de regras de produção. Inicialmente, o algoritmo reconhece dois tipos de atributos: *pos-atts*, que são atributos cujos pesos das conexões com um dado neurônio é positivo, e *neg-atts*, que são atributos com pesos das conexões negativos. Uma vez obtidos estes atributos para um dado neurônio, *KT* testa combinações de *pos-atts* e *neg-atts*, estabelecendo uma metodologia de poda de combinações. Em redes neurais do tipo multicamadas, este procedimento é feito camada a camada, de tal forma que, os neurônios das camadas intermediárias são tratados como atributos para os neurônios da camada de saída. Este algoritmo é do tipo *decomposicional*.

Um outro método, transcrito em [SES94], denominado de método de Niida *et al.*, procura diminuir a complexidade de uma RNA-ML pela remoção de redundâncias através da retirada de atributos de entrada, formando uma rede esqueleto. Esta rede esqueleto é formada por um processo de otimização em dois níveis: verificação matemática da independência linear entre os atributos de entrada e uma seleção entre as remanescentes por um especialista. A rede é processada no ciclo de treinamento e teste e uma análise de sensibilidade é feita para os atributos de entrada. Se o desempenho não for satisfatório, o processo reinicia. Quando a uma rede esqueleto aceitável é encontrada, são extraídas as questões dominantes da seguinte forma: se o peso da conexão entre uma dada entrada e um dado neurônio da camada oculta tem o mesmo sinal que o peso da conexão entre este neurônio e uma dada saída, então o efeito desta entrada em relação a esta saída é positivo. Caso apresentem sinais diferentes, o efeito é negativo. Entretanto, estes resultados devem ser observados por um especialista, o qual deve colocá-lo em

um formato útil. Logo, a obtenção das regras é totalmente automática, sendo estas absolutamente dependentes da rede esqueleto final e da análise do especialista. Além disso, nota-se que gerar a melhor rede pode levar a uma complexidade computacional intratável. Este método é considerado *decomposicional*.

O método *RN* ou *RF5* ([SAI90],[SAI96]), utiliza um mecanismo de avaliação de fatos positivos e negativos através das faixas de valores dos atributos de entrada para os quais uma dada saída responde positiva ou negativamente. A partir daí, regras disjuntivas SE-ENTÃO são obtidas. Este método é do tipo *decomposicional*, pois os fatos internos obtidos são arranjados para compor as regras finais. Este método não portável, pois é somente aplicável em redes que utilizam ‘unidades produtos’ na camada oculta [TIC98].

Alguns métodos utilizam o conhecimento *a priori* para inicializar uma rede neural. Nos métodos de Sethi ([SET90],[SET95]) e Ivanova e Kubat [TIC98] este conhecimento *a priori* é fornecido através de uma árvore de decisão. *TREPAN* [CRA96] é um método aplicável tanto a uma rede neural quanto a outros métodos simbólicos de aprendizado indutivo. A idéia deste algoritmo é obter uma árvore de decisão que aproxima da metodologia utilizada (e.g., neural). Este método é *pedagógico*, pois apresenta a árvore resultante é obtida das relações globais entre entradas e saídas. Em *KBANN* [TOW92] o conhecimento, exposto através de regras, é transformado em uma rede neural que as refina. O algoritmo *TopGen* [OPI93] é uma evolução do método *KBANN*, pois permite que novas regras sejam incorporadas. Isso é feito buscando-se heurísticamente, através do espaço de hipóteses, nós na rede neural que melhor refina as hipóteses iniciais formuladas. Pode ser notado que tanto o *KBANN* quanto o *TopGen* são do tipo *decomposicional*.

A referência [BEN97] mostra que as redes neurais podem ser transformadas num conjunto de regras difusas. O objetivo é aproximar ambos os modelos, fuzzy e neural, para ‘abrir a caixa-preta’ da rede neural. Para proporcionar uma melhora na interpretação de uma RNA por regras difusas, os autores introduzem um conceito, denominado de *f-dualidade*. Este conceito define um operador lógico difuso, denominado de *iterativo-ou* (*i-or*), que, aplicado às regras difusas, produzem um conjunto de regras mais interpretáveis. Uma das vantagens da metodologia apresentada é que as regras extraídas independem do algoritmo de aprendizagem da rede neural, importando somente o mapeamento dado pelos pesos das conexões ao final do treinamento. Entretanto, as regras produzidas, apesar de refletirem a rede neural, não mostram o quanto cada variável é importante para o mapeamento. Neste caso, os antecedentes das regras são exatamente todas as variáveis de entrada.

tanto supervisionada quanto não supervisionada. A descrição completa deste algoritmo para RNAs supervisionadas é mostrada no Apêndice D. BRAINNE é um algoritmo do tipo *pedagógico* onde, para se obter as entradas mais correlacionadas com uma classe do problema, as saídas são colocadas como entradas adicionais da rede neural. Treinando-se uma RNA-ML com esta nova topologia, a idéia é obter o grau de correlação entre uma variável de entrada com uma dada saída. Isto é feito através de uma medida de distância entre os pesos das entradas originais e os pesos das saídas como novas entradas.

Além dessa medida de distância, BRAINNE utiliza-se de uma RNA-SL com treinamento baseado na regra de Hebb (Apêndice C) para enfatizar as variáveis que são menos relevantes através dos pesos das conexões entradas-saídas. Para realizar esse processo, as entradas são negadas, ou seja, se uma entrada tem valor '1', seu novo valor será '0'. Caso a entrada seja contínua, então o valor negado será $|1 - \text{valor da entrada}|$. A composição do índice final de correlação de uma entrada com uma saída é o produto entre a medida de distância obtida para esta entrada e o seu grau de irrelevância para o processo. Isso é feito para auxiliar situações onde há pouco exemplares no conjunto de dados e, onde há várias saídas, isto é, auxiliar na determinação da contribuição de uma entrada para cada uma das saídas.

Uma vez obtido o índice para cada variável, este é colocado em ordem decrescente. A variável de maior índice é selecionada, funcionando como o *ganho* de informação em uma árvore de decisão. A diferença deste processo com o de uma árvore é que as regras vão sendo formadas e testadas em cada ponto de decisão. O processo para quando a taxa de classificação for a melhor possível com as regras extraídas.

O algoritmo BRAINNE tem a vantagem de classificar bem com as regras extraídas, ou seja, o critério da *precisão* é satisfeito. Outra vantagem é que as variáveis presentes nos antecedentes das regras são muito correlacionadas com as saídas, o que mostra um ótimo seletor de características, satisfazendo o critério da *fidelidade*. Entretanto, se não houver a ocorrência de um ponto de corte (Apêndice D), o número de antecedentes das regras pode crescer muito, gerando regras com baixa *compreensibilidade*. Como na maioria dos problemas do mundo real é muito difícil encontrar um ponto de corte que, num primeiro processamento, possibilita obter um conjunto de variáveis já fortemente correlacionadas, então o retreinamento das redes neurais se faz necessário para toda nova variável a ser testada. Isso é inviável para grandes bases de dados ou quando a velocidade de extração de regras é importante, como em ambiente *on-line*.

Na tentativa de amenizar esse problema, neste trabalho de tese foi testado a combinação do BRAINNE com a rede neural OET2 em substituição à RNA-ML por suas já citadas

características. Entretanto, o BRAINNE não se mostrou *portátil*, pois o esquema de mapear as saídas nas entradas não é possível com o OET2. Neste caso, o OET2 apresentou uma sensibilidade muito grande para com as entradas adicionais (saídas), fazendo com que os pesos das conexões das variáveis de entrada originais fossem praticamente anuladas em todos os testes realizados. A referência [DIL01] apresenta um modelo alternativo ao BRAINNE para retirar a recursividade da técnica. Se não ocorrer pontos de corte no BRAINNE convencional, então é aplicada uma técnica baseada na obtenção de agrupamentos através de uma RNA não supervisionada tipo Mapa de Kohonen, o qual são transformados em regras. Como pode ocorrer mais de um agrupamento por classe, então pode haver regras disjuntivas classificando uma mesma classe. A metodologia proposta gera regras mutuamente exclusivas. Estas são então otimizadas para, não só se obter o mínimo de taxa de erro de classificação, como também observar fatos com baixa probabilidade de ocorrência. Os resultados mostram que o conjunto de regras extraídas é mais preciso que o BRAINNE convencional. Não é informado, entretanto, o tempo de processamento para gerar esse conjunto de regras.

Os trabalhos [ABR98] e [ABR98a] mostram a aplicação de algoritmo BRAINNE ao problema de avaliação de segurança de tensão. A idéia destes artigos é obter o nível de segurança operativa do sistema através de regras extraídas pelo BRAINNE. Além disso, mostrar que as variáveis que compõem estas regras são efetivas para a análise do problema. A metodologia empregada para a geração da base de dados é a mesma descrita no Capítulo 2. Através do valor da margem de potência ativa encontrada para o caso base e algumas contingências no sistema teste IEEE 14 barras, são aferidas regiões de operação (classes do problema). As variáveis escolhidas para compor o vetor de entrada são o módulo da tensão e a potência ativa consumida em cada barra de carga do sistema. Os resultados mostram que as regras extraídas apresentam erros baixos de classificação para o conjunto de teste, e as variáveis presentes nas regras mostram-se fortemente correlacionadas com o problema de estabilidade de tensão, ou seja, as barras críticas são obtidas como as relevantes do processo. Entretanto, nota-se a recursividade das chamadas do procedimento de gerar e testar atributos do BRAINNE, empreendendo um custo computacional muito alto.

A Tabela 3.9 apresenta um quadro resumo das principais metodologias para extração de regras de RNAs e suas respectivas características.

Tabela 3.9 – Quadro resumo dos algoritmos para extração de regras de RNAs, mostrando algumas características mais importantes para análise.

ALGORITMO	FIDELIDADE	COMPREENSIBILIDADE	PORTABILIDADE	PRECISÃO	TIPO
MACIE	Alta	Média	Sim	Média	Eclético
MoFN	Alta	Média	Sim	Alta	Decomp.
RX	Alta	Alta	Não	Alta	Decomp.
KT	Alta	Media	Sim	Alta	Decomp.
Niida at al	NA	NA	Não	Média	Decomp.
RN	Alta	NA	Não	Alta	Decomp.
TREPAN	Alta	Alta	Sim	Alta	Pedagógico
Ben97 (ref.)	Alta	Baixa	Sim	Alta	Eclético
Liu99 (ref.)	Alta	Baixa	Não	Alta	Decomp.
BRAINNE	Alta	Média/baixa	Não	Alta	Pedagógico

A seguir são apresentados os conceitos teóricos do algoritmo VIA, utilizado neste trabalho para a tarefa de extração de regras de produção de redes neurais artificiais.

3.8 VALIDITY INTERVAL ANALYSIS

O algoritmo VIA [THR94] é uma técnica de extração e refino de regras de produção aplicável às redes neurais artificiais do tipo *feedforward*. Antes da descrição do algoritmo, é interessante observar suas principais características. Uma vez que o VIA é utilizado neste trabalho para a tarefa de extração de regras de produção de RNAs, a análise de suas características possibilitam justificar sua utilização, como se segue:

1. O algoritmo apresenta *portabilidade*, sendo aplicável a redes neurais tipo *feedforward* de forma direta. Neste caso, basta o armazenamento dos pesos das conexões e valores de saída dos neurônios por classe.
2. Não há necessidade de implementação de qualquer esquema especial de treinamento das redes neurais utilizadas.
3. A *fidelidade* das regras extraídas é alta, uma vez que a convergência do algoritmo só

acontece se os intervalos refletirem o mapeamento neural.

4. A implementação computacional do algoritmo é simples e os conceitos teóricos envolvidos para a solução do processo são robustos. Além disso, são métodos bastante difundidos e discutidos, como é o caso do método Simplex para Programação Linear.
5. A forma de processamento do VIA é bastante similar ao realizado pela rede neural com treinamento OET2. Aliado a excelente portabilidade do VIA, a utilização conjunta de ambos algoritmos pode proporcionar convergência mais rápida do algoritmo. Este fato mostra-se bastante interessante para aplicações que exigem eficiência computacional.
6. A *compreensibilidade* das regras extraídas está relacionada com a técnica empregada para explorar o espaço das hipóteses. Técnicas baseadas em processos aleatórios de busca tendem a ter um custo computacional muito elevado, gerando um número muito grande de regras. Entretanto, esta tarefa pode ser bastante minimizada se for utilizada alguma técnica primária de geração de hipóteses, como por exemplo as DTs. Neste trabalho, esta última abordagem descrita é utilizada para gerar as hipóteses iniciais.
7. A *precisão* das regras extraídas pelo VIA depende da hipótese inicial formulada. Dependendo das variáveis selecionadas nas regras extraídas pelo VIA, além de imputarem maior precisão às regras, estas podem ser capazes de reduzir o sobre-ajuste do treinamento da rede neural.
8. O tipo do algoritmo VIA é *eclético*. Os elementos da rede, através de suas respectivas saídas, são observados individualmente, mas as regras são extraídas das combinações dos resultados obtidos de cada processamento individual realizado.

Das características acima citadas, a portabilidade do VIA, a facilidade de implementação computacional e similaridade com a técnica OET2 são as características de maior influência na escolha deste algoritmo. Além disso, VIA lida com dados contínuos sem que qualquer esquema especial implementado para esta finalidade. Dos algoritmos descritos no item 3.7 e sumarizados na Tabela 3.9, embora muitos deles atendam de forma ótima os critérios de precisão e fidelidade, alguns não lidam com dados contínuos (e.g. MACIE e MofN) e outros dependem de esquemas especiais de treinamento da rede neural ou topologias específicas (e.g. Liu99, RX, RF5 e KT). Outros, como os algoritmos BRAINNE e Niida *at al* são computacionalmente ineficientes, e métodos como TopGen e Sethi necessitam de conhecimento *a priori* para treinar uma rede

neural. Já a proposta do TREPAN é transformar o conhecimento implícito na RNA em uma DT, necessitando de um novo processamento para extração de regras. Logo, de acordo com as características citadas para o VIA, nota-se que este preenche as lacunas deixadas por estes algoritmos avaliados. A seguir, o algoritmo VIA é descrito passo a passo.

Inicialmente, vale lembrar que cada variável de um conjunto de treinamento com n exemplares apresenta um valor máximo e outro mínimo, sem distinção de classes. Estes valores definem o intervalo de cada variável e este é dito ser máximo (ou *aberto*). Um intervalo aberto mostra que há desconhecimento total sobre a informação que cada variável pode proporcionar. Neste caso, a variável fornece uma informação completamente não-discriminatória. Se uma variável for encontrada como tendo um alto grau de correlação com uma dada classe do problema, o intervalo desta deve ser restringido para a faixa de valores da classe em questão. Ou seja, o conhecimento sobre esta classe é expresso através da variável e seu respectivo intervalo.

Uma técnica de extração de regras que observa estes intervalos deve ser capaz de redefini-los para proporcionar a melhor taxa de classificação para exemplares não observados. Formalizando, existe então um subespaço d -dimensional que proporciona uma taxa de classificação ótima, onde d é o número de variáveis selecionadas entre as m variáveis presentes. A idéia básica é encontrar este subespaço através da relação das variáveis mais correlacionadas a uma dada classe e seus respectivos intervalos. Observa-se que cada subespaço pode ser expresso na forma de uma regra de produção da seguinte maneira:

Regra: Se $[(Vl_{min} \leq Vl \leq Vl_{max}) \wedge \dots \wedge (Vd_{min} \leq Vd \leq Vd_{max})]$ **então**
classe **A**

onde $Vl\dots Vd$ denotam as variáveis selecionadas. Esta regra é dita conjuntiva pois ela só é válida se seus antecedentes forem validados em conjunto. Entretanto, podem ser obtidas múltiplas regras para uma mesma classe. Uma característica do VIA é a impossibilidade de encontrar um único subespaço ótimo e, conseqüentemente, gerar uma única regra que expresse todo o conhecimento por si só sobre uma dada classe. Assim, o resultado é, normalmente, um conjunto grande de regras para uma dada classe. Isso mostra a priori que para o VIA, o critério da *compreensibilidade* deixa a desejar em relação ao número de regras.

Na realidade, o VIA testa hipóteses. Estas hipóteses nada mais são que regras, onde os intervalos das variáveis apresentam valores inicialmente arbitrados ou indicados por alguma metodologia externa. Se uma hipótese é testada pelo VIA, o resultado são novos intervalos que

procuram refletir o treinamento da rede neural. Nota-se, entretanto, que existirão uma quantidade não mensurável de hipóteses a serem testadas, determinando a necessidade de estabelecer um mecanismo de gerar e testar hipóteses. Várias metodologias são propostas com esta finalidade. A mais intuitiva seria partir da hipótese mais geral possível, i.e., o desconhecimento total sobre os intervalos das variáveis. A partir daí, através de um processo de busca, seriam testadas combinações de intervalos das variáveis. Entretanto, o custo computacional desta proposta é muito elevado. A referência [THR93] utiliza esta metodologia para gerar um conjunto de regras para caracterizar os movimentos de um robô, sendo o número de regras extraídas muito grande.

Uma outra proposta é utilizar uma informação de um ponto no espaço de hipóteses que classifica com certeza absoluta uma dada classe. Este ponto funciona como uma semente para que sejam testadas novas combinações de intervalos das variáveis em torno deste, até que se obtenha regras mais gerais.

A terceira proposta é utilizar a informação obtida de um processo primário de seleção de variáveis e/ou regras, como os das DTs. Neste caso, VIA refina estas hipóteses de tal forma a explicitar variáveis mais correlacionadas com uma dada classe, bem como seu respectivo intervalo. Neste trabalho, são utilizadas as hipóteses geradas pela DT C4.5 como a informação inicial a ser refinada pelo VIA.

Outra questão envolvida diz respeito à diferença entre *refinar* uma regra ou *extrair* uma regra. No VIA, uma regra a ser testada deve conter todas as variáveis do processo. Por exemplo, considerando um problema onde existam 4 variáveis de entrada (A, B, C, D) e a seguinte regra é apresentada ao VIA como hipótese:

Regra: Se $[(A1 \leq A \leq A2) \wedge (C1 \leq C \leq C2)]$ **então** classe **X**

onde A1, A2, C1 e C2 representam a faixa de valores iniciais destas variáveis. Entretanto, como B e D não são partes da regra original, seus intervalos são apresentados totalmente *abertos* (ou seja, irrelevância destas variáveis). A regra apresentada ao VIA fica:

Regra: Se $[(A1 \leq A \leq A2) \wedge (B_{\min} \leq B \leq B_{\max}) \wedge (C1 \leq C \leq C2) \wedge (D_{\min} \leq D \leq D_{\max})]$ **então** classe **X**

Onde B_{\min} , B_{\max} , D_{\min} e D_{\max} são os valores mínimos e máximos que as variáveis B e D assumem no conjunto de treinamento. Se, após o processamento do VIA, somente os intervalos das variáveis A e C forem modificados, então pode-se dizer que esta regra foi *refinada*. Entretanto,

se pelo menos um dos intervalos das outras variáveis, B e D, forem também modificados, uma nova regra foi *extraída*.

Uma vez treinada uma rede neural tipo *feedforward* (e.g. RNA-ML ou OET2), pode-se mostrar o processo pelo qual o VIA obtém os intervalos válidos de cada variável em relação a uma dada classe. O VIA atua por partes, estabelecendo intervalos válidos para os neurônios da próxima camada. Estes intervalos podem ser vistos na Figura 3.7, onde duas camadas adjacentes de uma RNA são mostradas. Esta apresenta m neurônios na camada P e n na camada S .

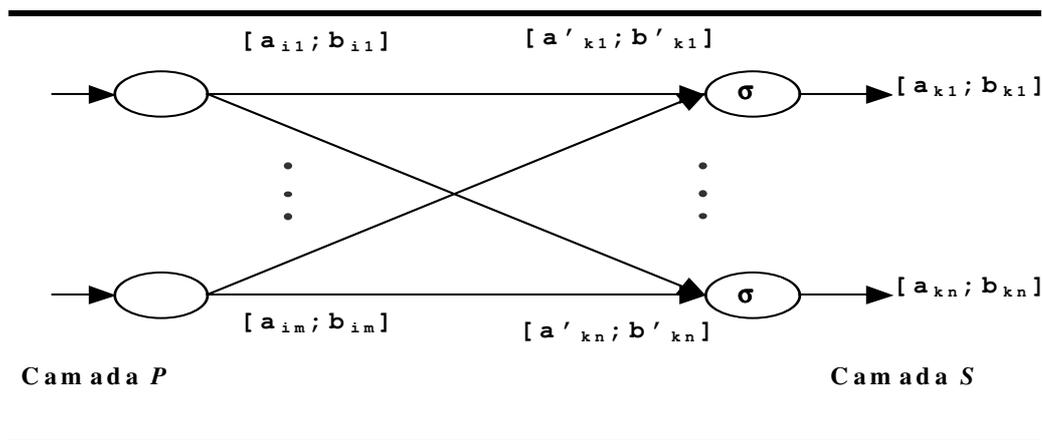


Figura 3.7 – Parte de uma RNA mostrando as saídas das ativações.

Observa-se que, caso a camada P seja a camada de entrada, então os intervalos dos sinais $[a_{i1...im}, b_{i1...im}]$ para os m canais de entrada são os próprios intervalos das variáveis de entrada, obtidos do conjunto de treinamento. Se a RNA for tipo multicamadas, então a camada S , neste caso, seria uma camada oculta com seus n neurônios. Como o VIA trabalha por partes, processando camada por camada, então a saída de uma camada constitui a entrada para a próxima, se for o caso.

Considerando o exemplo do sistema de potência teste 3 barras apresentado no item 3.5.1 deste capítulo, contendo uma barra *swing* e duas barras de carga, conforme mostra a Figura 3.5. Os respectivos conjuntos de treinamento e teste são mostrados nas Tabelas 3.1 e 3.2, gerados conforme a metodologia proposta no Capítulo 2. Este exemplo é mostrado com o intuito de auxiliar o entendimento do algoritmo VIA. Uma vez obtido o conjunto de treinamento, é necessário estabelecer os intervalos inicialmente propostos para cada variável de entrada, constituindo-se estes nos intervalos da camada P . Assim, obtém-se os valores mínimos e máximos das variáveis, mostrados na Tabela 3.10.

Tabela 3.10 – Valores mínimos e máximos de cada variável.

SISTEMA ELÉTRICO 3 BARRAS		
Mínimo	Variável	Máximo
0.597	V2	0.936
0.859	V3	0.973
0.150	ZI2	1.229
0.795	ZI3	2.564

A seguir, define-se as hipóteses a serem testadas pelo algoritmo VIA. Para exemplificar o algoritmo inicialmente serão consideradas as hipóteses mais gerais, ou seja, para uma dada classe, existe o desconhecimento total dos intervalos das variáveis. Logo, sob o formato de regras de produção, estas hipóteses são:

- R1: Se** $[(0.597 \leq V2 \leq 0.936) \wedge (0.859 \leq V3 \leq 0.973) \wedge (0.150 \leq ZI2 \leq 1.229) \wedge (0.795 \leq ZI3 \leq 2.564)]$ **então** Normal
- R2: Se** $[(0.597 \leq V2 \leq 0.936) \wedge (0.859 \leq V3 \leq 0.973) \wedge (0.150 \leq ZI2 \leq 1.229) \wedge (0.795 \leq ZI3 \leq 2.564)]$ **então** Alerta
- R3: Se** $[(0.597 \leq V2 \leq 0.936) \wedge (0.859 \leq V3 \leq 0.973) \wedge (0.150 \leq ZI2 \leq 1.229) \wedge (0.795 \leq ZI3 \leq 2.564)]$ **então** Inseguro

Observa-se que é proposta uma regra para cada classe e estas são iguais pela consideração de desconhecimento total sobre os intervalos (dito *abertos*). Então, a camada **P** apresenta 5 entradas, sendo 4 para as variáveis e uma para o *bias*.

Propagando-se estes intervalos abertos, como mostrado na Figura 3.7, sabe-se que a saída da função de ativação, x_k , de uma unidade na camada oculta ou de saída é calculada em duas etapas, onde a função de ativação σ aqui utilizada é uma função logística (e.g. sigmoidal), não linear e monotônica, tal que:

$$x_k = \sigma(net_k) \quad \text{com} \quad net_k = \sum_{i \in P} w_{ik} x_i + \theta_k \quad (3.11)$$

onde w_{ik} e θ_k são os pesos das conexões e *bias* respectivamente, e net_k a entrada do neurônio k . Os intervalos válidos para os valores da ativação, x_k , são representados por $[a_k; b_k]$. Neste trabalho, a função de ativação utilizada é a *tangente hiperbólica*.

3.8.1 VIA e a rede neural baseada no treinamento OET2

Neste ponto do desenvolvimento do VIA, é importante salientar a motivação de utilizar a

RNA baseada no treinamento OET2 [SIL91]. O algoritmo matemático é mostrado no Apêndice C. Muitas das aplicações de redes neurais utilizam algoritmos baseados na regra delta generalizada com retropropagação de erros. Esse tipo de técnica apresenta dificuldades em lidar com grandes problemas no mundo real. Embora tenham sido implementados melhoramentos neste algoritmo visando contornar estas deficiências, ainda assim um método de treinamento mais confiável e eficiente se faz necessário. O maior problema é a forte não linearidade da função erro no espaço dos pesos das conexões. O OET2, proposto por [SIL91], faz com que a função erro seja aproximada por sucessivas aproximações quadráticas, ao invés de realizar a busca utilizando a função erro (multimodal) original, como na regra delta generalizada.

OET2 é um método de treinamento supervisionado, onde ocorre um mapeamento ótimo, não linear, entre os padrões de entradas e os de saída do conjunto de treinamento. Isso é feito através de sucessivas soluções de sistemas lineares via métodos diretos de mínimos quadrados. Na realidade, é um procedimento análogo aos métodos baseados em mínimos quadrados não lineares. Ou seja, são aproximações locais da função erro por uma função quadrática e sua respectiva minimização (exata).

Um importante atributo desta técnica é que o problema de escala é muito bem caracterizado. Baseado na complexidade das operações envolvidas ($O[r.m^2]$, onde 'r' é o número de entradas dos padrões de treinamento e 'm' sua dimensão), é fácil estimar como o OET se comporta com o tamanho da rede. A referência [SIL91] mostra também novos atributos incorporados ao OET, que são: estabilidade numérica, melhoramento das entradas e possibilidade de limitar a magnitude dos pesos das conexões.

Do exposto acima, uma das principais motivações para a utilização do aprendizado via OET2 para a extração do conhecimento simbólico é a velocidade de treinamento. Uma vez que a contribuição deste trabalho é proporcionar que as técnicas aqui observadas sejam destinadas a um ambiente *on-line*, a velocidade do processo torna-se importante. Logo, OET2 não adiciona custo computacional significativo ao algoritmo VIA, diferentemente da RNA-ML. Outra vantagem, a ser observada a seguir, é que a forma como o VIA refina os intervalos das variáveis (utilizando Programação Linear (PL) [IGN94]) é bastante similar ao algoritmo de aprendizagem OET2. Ou seja, ambos processamentos são feitos de forma linear e por partes. A consequência disso é que o mapeamento dos pesos das conexões possibilita a convergência mais rápida do VIA.

3.8.2 Resultados do OET2 para o sistema 3 barras

Reportando-se ao exemplo do sistema de 3 barras, após o treinamento com o OET2, obtém-se os pesos das conexões de cada camada utilizada. Além disso, são armazenadas as 19 saídas dos neurônios, fruto da aplicação do conjunto de treinamento na rede treinada. Entretanto, vale notar que, como se está testando as hipóteses por classe do problema, existe para cada classe do problema um respectivo intervalo na saída de cada neurônio. Se há 3 regiões de operação definidas no sistema 3 barras, então existem 3 intervalos correspondentes em cada saída de um neurônio. A topologia do OET2 aqui utilizada é composta de duas camadas, sendo que o número de neurônios na camada oculta é igual ao número de entradas. Assim, têm-se 5 entradas, 5 neurônios na camada oculta e 2 saídas.

O fato de existirem apenas 2 saídas diz respeito à forma de codificação destas. Neste caso, a codificação é feita da seguinte forma: Normal \rightarrow -1 e -1, Alerta \rightarrow 1 e -1 e Inseguro \rightarrow 1 e 1. Este tipo de codificação lida melhor com classes que possuem relação de ordem. Para este problema, não foram encontradas dependências lineares entre as variáveis e o valor do *bound1* definido foi de 43,225. Neste trabalho, o valor do *bound1* corresponde à média dos valores dos pesos entrada/camada oculta multiplicado por um fator 5. A Tabela 3.11 mostra os valores dos pesos das conexões e as Tabelas 3.12 e 3.13, os intervalos de saída dos neurônios por classe.

Tabela 3.11 – Pesos das conexões das 2 camadas.

Pesos conexões (w_{ik})					
Entradas	Neurônios camada oculta				
	H1	H2	H3	H4	H5
Bias	13.2288	-43.2250	-4.6471	-0.2958	0.7208
V2	-2.4567	4.9018	-3.6569	0.2420	-0.8781
V3	-12.1702	41.9435	8.0983	0.1139	-0.0336
ZI2	-0.4778	2.0163	0.1135	0.0129	-0.0350
ZI3	0.9487	-2.5269	-0.0125	-0.0345	0.1045
Neurônios Camada saída	H1	H2	H3	H4	H5
S1	116.5160	10.2730	834.2992	-62233.1285	-21162.2929
S2	81.7074	8.3276	433.2111	-26142.5214	-9120.1547

Tabela 3.12 – Intervalos do valor de *net* (neurônios da camada oculta).

CLASSES	Neurônios camada oculta	Valor mínimo	Valor máximo
NORMAL	H1	-0.9273	-0.6921
	H2	2.1844	3.1097
	H3	-0.1216	-0.0887
	H4	0.0292	0.0386
	H5	-0.1214	-0.0923
ALERTA	H1	-0.4767	0.1156
	H2	-0.5812	1.2929
	H3	-0.1185	-0.0310
	H4	-0.0025	0.0225
	H5	-0.0731	0.0063
INSEGURO	H1	0.5730	1.1534
	H2	-3.6067	-1.9098
	H3	0.0551	0.1799
	H4	-0.0500	-0.0234
	H5	0.0734	0.1592

Tabela 3.13 – Intervalos do valor de *net* (neurônios da camada de saída).

CLASSES	Neurônios camada de saída	Valor mínimo	Valor máximo
NORMAL	S1	-3.3281	-1.4830
	S2	-3.3860	-2.0939
ALERTA	S1	1.1200	3.5148
	S2	-3.0696	-2.0752
INSEGURO	S1	2.6604	3.3046
	S2	1.9183	2.7990

A Tabela 3.14 apresenta a taxa de classificação para o conjunto de treinamento obtido pelo OET2.

Tabela 3.14 - Taxa de classificação obtida pelo OET2 para o conjunto de treinamento do sistema 3 barras.

Normal	Alerta	Inseguro	Total Class. Correta	Total Class. Errada
<u>06</u>	0	0	100.00% (06/06)	0.0% (0/13)
0	<u>06</u>	0	100.00% (06/06)	0.0% (0/13)
0	0	<u>07</u>	100.00% (07/07)	0.0% (0/12)
Class. Correta			Class. Errada	
<u>100.00% (19/19)</u>			<u>0.0% (0/19)</u>	

3.8.3 Fases *Forward* e *Backward* do VIA

Considerando Φ o conjunto dos intervalos válidos para todos os neurônios da camada S tal que $\Phi = \{[a_{k1}, b_{k1}]; \dots; [a_{kn}, b_{kn}]\}$, um vetor de ativações $x_k = \langle x_1, \dots, x_n \rangle$ é dito *admissível* com respeito ao conjunto Φ se todas as ativações estão contidas em Φ . Um conjunto de intervalos é dito *consistente* se todos os vetores de ativação são *admissíveis*, senão Φ é dito ser *inconsistente*. Logo, o algoritmo *converge* quando o conjunto Φ_R dos intervalos resultantes do refinamento é *consistente* com o conjunto Φ (intervalos iniciais), ou seja, $\Phi_R \subset \Phi$. Uma *contradição* ocorre se, ao final do processo, pelo menos um intervalo não satisfaz as restrições impostas pelos intervalos iniciais válidos. Conseqüentemente, os intervalos iniciais são inconsistentes com os pesos e *biases* da RNA. Este fato mostra a dependência do VIA em relação à qualidade do treinamento da rede neural e dos intervalos hipóteses que inicializam a análise. Assim, se deve haver *consistência* na convergência do VIA, então o critério de *fidelidade* das regras extraídas é alto. A maneira como o VIA refina Φ iterativamente é através de PL.

Sendo que os neurônios da camada P estão todos completamente conectados com os neurônios da camada S , a não linearidade das funções de ativação deve ser eliminada para possibilitar a aplicação de uma técnica de PL. Isto é feito projetando-se o intervalo $[a_k; b_k]$ tal que $[a'_k; b'_k] = \sigma^{-1}([a_k; b_k])$ para todo $k \in S$, conforme mostra a Figura 3.7. Logo, os intervalos válidos resultantes em P e S formam um conjunto de restrições lineares em relação aos valores de ativação em P :

$$\begin{aligned} \forall i \in P : a_i &\leq x_i \leq b_i \\ \forall k \in S : a'_k &\leq net_k = \sum_{i \in P} w_{ik} x_i + \theta_k \leq b'_k \end{aligned} \quad (3.12)$$

A programação linear permite maximizar ou minimizar combinações lineares das variáveis arbitrárias x deste que não violem o conjunto das restrições lineares. Isto possibilita aplicar PL para refinar os valores altos e baixos dos intervalos válidos um por um. Neste trabalho, o método utilizado de PL é o Simplex Revisado. O algoritmo VIA é processado em duas fases:

Fase *Forward*: Os valores limites a_k e b_k são refinados para os neurônios $k \in S$, obtendo-se

$$\begin{aligned} \hat{a}_k &= \sigma(\hat{a}'_k) \text{ with } \hat{a}'_k = \min net_k \\ \hat{b}_k &= \sigma(\hat{b}'_k) \text{ with } \hat{b}'_k = \max net_k \end{aligned} \tag{3.13}$$

Se $\hat{a}_k > a_k$, um valor limite inferior mais elevado é encontrado e a_k é atualizado para \hat{a}_k . De forma análoga, se $\hat{b}_k < b_k$, b_k é atualizado para \hat{b}_k .

Neste ponto do desenvolvimento, é interessante observar a questão da convergência do algoritmo VIA. Segundo [MAI00], a convergência para uma rede neural com uma camada ocorre em uma iteração. Entretanto, quando a rede neural utilizada é do tipo multicamadas, a convergência ocorre segundo uma taxa média exponencial. Isso é devido a não linearidade do mapeamento neural produzido pela arquitetura multicamadas. Desta forma, o VIA termina quando a diferença entre os valores dos intervalos obtidos da varredura anterior e dos valores destes na iteração corrente é menor que uma tolerância. Embora isso possa levar à dedução que o algoritmo se torna lento, na prática, esta convergência é obtida com poucas iterações.

Para o exemplo do sistema 3 barras, a hipótese dada pela regra R1 *converge com 2 iterações*. Vale salientar que uma iteração corresponde ao ciclo de varredura das fases *forward* e *backward*. Para simplificação, são apresentados apenas os resultados correspondentes à segunda iteração, ou seja pós-convergência. Os resultados para a fase *forward*, tanto para os intervalos de saída dos neurônios da camada oculta quanto para os da camada de saída, são mostrados na Tabela 3.15.

Tabela 3.15 – Resultado da fase *Foward* para a regra R1, após convergência (2ª iteração), onde são mostrados os intervalos refinados para cada neurônio.

Neurônios camada oculta		
H1	-0.7293	-0.5993
H2	0.9750	0.9960
H3	-0.1210	-0.0884
H4	0.0292	0.0385
H5	-0.1208	-0.0921
Neurônios camada de saída		
S1	-0.9974	-0.9021
S2	-0.9977	-0.9701

Para simplificar a apresentação do VIA, são apresentados apenas os resultados para a regra R1. Entretanto, apenas a título informativo, as regras R2 e R3 também convergem em 2

iterações. Após a obtenção dos intervalos de ativação de cada neurônio na fase *forward*, é necessário retropropagar estes intervalos até a entrada para completar o ciclo de refino das variáveis. Isto é feito através da fase *Backward*, mostrada a seguir.

Fase Backward: Nesta fase, os valores limites a_i e b_i de todos os neurônios $i \in P$ são refinados, tal que:

$$\begin{aligned} \hat{a}_i &= \min x_i \\ \hat{b}_i &= \max x_i \end{aligned} \tag{3.14}$$

Assim como na fase *forward*, se $\hat{a}_i > a_i$, um valor limite inferior mais elevado é encontrado e a_i é atualizado para \hat{a}_i . De forma similar, se $\hat{b}_i < b_i$, b_i é atualizado para \hat{b}_i . Completada uma iteração, é verificado se existe algum intervalo que viola o valor de tolerância. Caso positivo, o VIA é reprocessado até que seja atingida a convergência. Os valores encontrados na fase *backward* após a convergência são mostrados na Tabela 3.16.

Tabela 3.16 – Resultado da fase Backward para a regra R1, pós convergência (2ª iteração), onde são mostrados os intervalos refinados para cada neurônio

Neurônios camada oculta		
H1	-0.7293	-0.5993
H2	0.9750	0.9960
H3	-0.1210	-0.0884
H4	0.0292	0.0385
H5	-0.1208	-0.0921
Entradas		
V2	<u>0.8300</u>	0.9360
V3	<u>0.9232</u>	0.9730
ZI2	0.1500	1.2290
ZI3	0.7950	2.5640

Na Tabela 3.16 pode ser observado que os valores dos intervalos dos neurônios da camada oculta são os mesmos que o da Tabela 3.15. Isso mostra que, a região de solução factível para a saída dos neurônios da camada oculta foi alcançada na fase *forward*. Retropropagando os intervalos (fase *backward*), estes permanecem inalterados, fruto da solução ótima alcançada.

O resultado para os intervalos das variáveis mostra que apenas os valores mínimos do intervalo das variáveis V1 e V2 foram refinados. Como as variáveis ZI2 e ZI3 estão com seus intervalos abertos, quem determina classificação da regra R1 são as variáveis V2 e V3. A nova regra obtida pelo VIA fica:

R11 **Se** $[(0.830 \leq V2 \leq 0.936) \wedge (0.9232 \leq V3 \leq 0.973)]$ **então** Normal

e as regras resultantes para as outras classes são:

R21: **Se** $(0.7264 \leq V2 \leq 0.9149) \wedge (0.8784 \leq V3 \leq 0.973)$ **então** Alerta

R31: **Se** $(0.597 \leq V2 \leq 0.7556) \wedge (0.859 \leq V3 \leq 0.9226)$ **então** Inseguro

A Tabela 3.17 mostra as taxas de classificação do OET2 para o conjunto de testes do sistema 3 barras, dado pela Tabela 3.2, Capítulo 3.

Tabela 3.17 - Taxa de classificação obtida pelo OET2 para o conjunto de teste do sistema 3 barras.

Normal	Alerta	Inseguro	Total Class. Correta	Total Class. Errada
<u>02</u>	04	0	33.33% (02/06)	66.67% (4/13)
0	<u>06</u>	0	100.00% (06/06)	0.0% (0/13)
0	0	<u>07</u>	100.00% (07/07)	0.0% (0/12)
Class. Correta			Class. Errada	
<u>78.95% (15/19)</u>			<u>21.05% (4/19)</u>	

A primeira análise a ser feita diz respeito ao resultado encontrado pela rede neural OET2. Observa-se que a taxa de classificação para o conjunto de treinamento (Tabela 3.14) apresentou erro de 0.0%. Entretanto, quando o conjunto de teste é observado, nota-se que o erro de classificação é alto (Tabela 3.17). Este resultado mostra que, devido ao fato de haver poucos exemplares de treinamento, pode estar ocorrendo um sobre-ajuste da superfície de separação entre as classes, refletindo-se na capacidade de generalizar do classificador. Este fato está ocorrendo mesmo com a diminuição do valor do *Bound1*, cuja função é promover uma regularização dos pesos (Apêndice C) na tentativa de reduzir a complexidade do modelo.

Observa-se também que neste exemplo em particular, a taxa de classificação do OET2 para o conjunto de teste é maior que a obtida pela regras extraídas pelo C4.5. Este fato é reflexo direto do baixo número que exemplares de treinamento, beneficiando o algoritmo do C4.5.

Entretanto, o C4.5 apresenta os maiores erros na região mais não linear (Inseguro) enquanto para o OET2 isso acontece na região menos não linear da curva PV (Normal), fruto da propriedade das RNAs de lidarem melhor com dados não lineares.

As regras extraídas pelo VIA são então testadas no conjunto de teste. Vale salientar que estas regras são obtidas da hipótese inicial em que considera os intervalos das variáveis de entrada abertos. A Tabela 3.18 apresenta as taxas de classificação de cada regra. São observados os exemplares corretamente classificados pela regra e os classificados incorretamente por uma ou mais regras. Apesar da regra R11 classificar corretamente todos os exemplares pertencentes a sua respectiva classe, esta não é discriminatória, classificando incorretamente exemplares pertencentes a outras classes.

A taxa de classificação errada global das regras é alta e igual a 42.11% (8 exemplares em 19). Isso acontece porque os intervalos das variáveis selecionadas não foram suficientemente refinados, pois as hipóteses iniciais (intervalos totalmente abertos) são muito pouco informativas. Uma vez que a proposta do exemplo foi inicializar o VIA com desconhecimento total sobre os intervalos das variáveis, os resultados mostram a necessidade de explorar o espaço de hipóteses para uma melhor inicialização. Todavia, é interessante observar que a regra R11 diminui o sobreajuste obtido pelo OET2 para a classe Normal.

Tabela 3.18 - Taxa de classificação por regra extraída do VIA para o sistema 3 barras - conjunto de teste. Hipótese inicial com os intervalos abertos

Regra	Normal	Alerta	Inseguro	Total Class. Correta	Total Class. Errada
R11	<u>06</u>	02	0	100.00% (06/06)	15/38% (02/13)
R21	05	<u>06</u>	0	100.00% (06/06)	38.46% (05/13)
R31	0	01	<u>07</u>	100.00% (07/07)	8.33% (01/12)
Total Geral	Class. Correta			Class. Errada	
	<u>100.00% (19/19)</u>			<u>42.11% (08/19)</u>	

3.8.4 Metodologia para testar as hipóteses obtidas pelo C4.5

Conforme mostrado no item anterior, o exemplo foi feito utilizando as regras a serem testadas pelo VIA com seus intervalos completamente abertos. A proposta neste item é utilizar as

regras geradas pelo C4.5 como as hipóteses que inicializarão o VIA. Desta forma, além de obter um ponto no espaço de hipóteses obtido por um processo primário de geração de regras, é possível comparar os regras finais extraídas pelo VIA com as próprias hipóteses iniciais formuladas pelo C4.5. A metodologia adotada segue os seguintes passos:

1. Processar o C4.5.
2. Reformatar as regras geradas de acordo com o seguinte exemplo:
 Seja a seguinte regra gerada pelo C4.5: R1: **A => 0.5 → classe X.**
 Supondo que o problema tem duas variáveis, $A=[0.2;0.8]$ e $B=[0.3;0.87]$, cujos valores explicitados são os valores mínimo e máximo de A e B no conjunto de treinamento, respectivamente. Então a regra R1 do C4.5 levado ao VIA terá o seguinte formato:
 R1: {A(0.5;0.8); B(0.3;0.87)} → classe X
3. Processar o VIA.
4. Verificar se a taxa de classificação das novas regras obtidas pelo VIA é melhor que a do C4.5.
5. Caso afirmativo, considere estas regras como as finais.
6. Caso negativo, execute os seguintes passos:
 - a. Selecione, em cada regra, a variável cujo intervalo teve maior alteração percentual em relação ao intervalo original, independente de ser uma nova variável destacada pelo VIA ou não. Se não foi destacada nenhuma nova variável, escolha uma das variáveis originais da respectiva regra.
 - b. Obtenha, no conjunto de treinamento, a faixa de valores de cada uma das variáveis selecionadas no item a considerando a respectiva classe da regra a qual pertence.
 - c. Substitua os valores encontrados nas novas regras extraídas pelo VIA, formando uma nova hipótese a ser testada.
 - d. Processar novamente o VIA com as novas hipóteses
 - e. Verifique as novas taxas de classificação. Se não houve melhora, considere a regra extraída do primeiro processamento como a regra válida, senão reter a nova regra.

Aplicando-se o procedimento acima no exemplo do sistema teste de 3 barras, as regras iniciais a serem testadas pelo VIA são as dada pela Tabela 3.5, Capítulo 3. As regras levadas ao VIA são:

- R1: Se** [(0.597 <= V2 <= 0.936) ∧ (0.938 <= V3 <= 0.973) ∧ (0.150 <= ZI2 <= 1.229) ∧ (0.795 <= ZI3 <= 2.564) **então** Normal
- R2: Se** [(0.690 <= V2 <= 0.936) ∧ (0.859 <= V3 <= 0.938) ∧ (0.150 <= ZI2 <= 1.229) ∧ (0.795 <= ZI3 <= 2.564) **então** Alerta
- R3: Se** [(0.597 <= V2 <= 0.690) ∧ (0.859 <= V3 <= 0.973) ∧ (0.150 <= ZI2 <= 1.229) ∧ (0.795 <= ZI3 <= 2.564) **então** Inseguro

Observa-se que as regras do C4.5 são formatadas conforme descrito no item 2 da

metodologia, sendo as variáveis destacadas em itálico e seus respectivos valores destacados em negrito. Vale salientar que, como a regra do C4.5 define apenas um valor limite para a variável, então nota-se que o outro valor definindo o intervalo corresponde ao valor máximo ou mínimo, dependendo do sinal da desigualdade da regra.

O resultado do processamento do VIA é mostrado na Tabela 3.19, onde são apresentadas as regras iniciais (hipóteses C4.5) e as regras finais obtidas pelo VIA. Nesta tabela, para efeito de simplificação dos resultados, são mostradas apenas as variáveis que fazem parte da regra do C4.5 (sublinhados) e as novas variáveis que tiveram seus intervalos alterados pelo VIA. Para exemplificar, observa-se apenas a regra 1. Nota-se que, a partir desta, uma nova regra foi extraída, dada por:

R11: Se [*(0.8484* <= *V2* <= *0.936)* \wedge (*0.938* <= *V3* <= *0.973)* então Normal

Este resultado mostra que a variável V3 não teve seu intervalo refinado e o VIA destacou a variável V2 para compor a nova regra. Observa-se então que, em relação ao intervalo original de V2, houve uma *alteração* neste intervalo de 74.44%. Os resultados encontrados então mostram que as hipóteses iniciais geradas pelo C4.5 para as regras 1 e 3 foram transformadas em novas regras pelo VIA. A regra 2, neste caso, teve apenas seus intervalos refinados.

A taxa de classificação destas novas regras para o conjunto de teste do sistema teste de 3 barras é apresentada na Tabela 3.20. Nesta tabela observa-se que a taxa de classificação errada das regras 1 e 2 foi significativamente reduzida em relação ao apresentado pelas regras do C4.5 originais (Tabela 3.7). Para a regra 3, as taxas de classificação correta e errada permaneceram inalteradas. Outro resultado mostra que, para a regra 1, houve uma melhora na taxa de classificação em relação ao OET2, reduzindo o sobre-ajuste do treinamento. Entretanto, para a regra 3 a hipótese inicial não foi suficientemente robusta para igualar ao resultado do OET2.

Processando o item 6 da metodologia apenas para a regra 3, a variável V3 é selecionada pelo critério adotado. A faixa de valores desta em relação à classe Inseguro, a qual a regra pertence, é [0.8590;0.8870]. Processando novamente com esta alteração na regra 3, o resultado mostra que as taxas de classificação permanecem inalteradas (correta - 5/7 e errada – 0/0). Este resultado mostra que a seleção da variável pelo C4.5, em particular para esta regra, não foi uma escolha adequada. Entretanto, o VIA não tem a robustez suficiente para buscar outras variáveis que podem melhorar a precisão da regra, fruto da técnica utilizada pelo algoritmo.

Tabela 3.19 - – Regras originais C4.5 e as novas obtidas pelo VIA, mostrando a alteração percentual do

intervalo original em relação ao intervalo novo de uma variável.

REGRAS INICIAIS E FINAIS
Caso base - Sistema 3 barras

INTERVALOS ORIGINAIS				INTERVALOS NOVOS			Alteração(%)
Min.	Var.	Max.		Min.	Var.	Max.	
REGRA NUM. == 1 ---->				CLASSE Normal			
0.5970	V2	0.9360		0.8494	V2	0.9360	74.44
0.9380	V3	0.9730		0.9380	V3	0.9730	0.00
REGRA NUM. == 2 ---->				CLASSE Alerta			
0.6900	V2	0.9360		0.7264	V2	0.8650	43.64
0.8590	V3	0.9380		0.8784	V3	0.9380	24.54
REGRA NUM. == 3 ---->				CLASSE Inseguro			
0.8590	V3	0.9730		0.8590	V3	0.9037	60.79
0.5970	V2	0.6900		0.5970	V2	0.6900	0.00

Tabela 3.20 - Taxa de classificação por regra extraída do VIA para o sistema 3 barras - conjunto de teste. Hipóteses iniciais dadas pelo C4.5

Regra	Normal	Alerta	Inseguro	Total Class. Correta	Total Class. Errada
R11	<u>06</u>	0	0	100.00% (06/06)	0.00% (0/13)
R21	0	<u>06</u>	0	100.00% (06/06)	0.00% (0/13)
R31	0	02	<u>05</u>	71.43% (05/07)	0.00% (01/12)
Total Geral	Class. Correta			Class. Errada	
	<u>89.47% (17/19)</u>			<u>0.00% (0/19)</u>	

Outro fato observado é que as únicas variáveis explicitadas pelo VIA são as tensões, tanto para o caso das hipóteses com intervalos abertos quanto para as hipóteses geradas pelo C4.5. Isso é reflexo do treinamento proporcionado pelo OET2 e são bastante intuitivas ao problema de avaliação de segurança de tensão. Entretanto, não se pode afirmar se realmente estas variáveis são as mais relevantes ao problema. O exemplo para essa afirmação é o resultado encontrado para a regra 3 descrito no parágrafo anterior. Para o VIA, a definição das variáveis relevantes só pode ser obtida se existir um processo de gerar e testar hipóteses, de tal forma a obter um conjunto maior de regras as quais possam ser avaliadas quanto ao ganho de informação que cada uma pode fornecer. Sob esse ponto de vista, para a regra 3 seria necessário testar novas combinações de variáveis e intervalos para melhorar as taxas de classificação dessa regra.

Vale salientar que, no exemplo para o sistema de 3 barras, o número de variáveis é

pequeno e não houve necessidade de seleção prévia de característica por alguma técnica escolhida para esta finalidade. Para sistemas de médio e grande porte, a seleção de um grupo primário de variáveis de interesse para o problema deve ser feita. A execução desta tarefa constitui-se numa das etapas que compõe uma metodologia para *Data Mining*. Especificamente neste trabalho, a técnica utilizada é a informação provida pelo método do vetor tangente (Capítulo 2) em relação às barras críticas de um sistema elétrico.

O objetivo final deste trabalho é mostrar que as regras extraídas de uma rede neural são qualitativamente melhores que as extraídas por uma árvore de decisão. Embora não seja possível afirmar que as variáveis selecionadas pelo VIA sejam as mais correlacionadas com o problema, o mesmo é verdade para as árvores de decisão. Apesar destas apresentarem um critério de seleção de variáveis embutido na metodologia de treinamento (item 3.5), a forma como esta seleção se processa não é suficiente para se estabelecer um conjunto seguro de variáveis fortemente correlacionadas com as classes de interesse. Logo, partindo-se das hipóteses geradas por uma árvore de decisão, pode-se utilizar o VIA para se obter relações mais precisas entre as variáveis de entrada e uma dada saída. Se este fato ocorrer, então as regras extraídas pelo VIA serão mais precisas que as da árvore de decisão, como mostrado no exemplo para o sistema elétrico de 3 barras.

3.9 CONCLUSÕES

Neste capítulo foi mostrado os aspectos relativos à adoção de técnicas de IA utilizando AAC, os quais motivaram a utilização destas neste trabalho. Assim, os aspectos como a capacidade de generalização e a eficiência computacional dos modelos empregados para AAC justificaram a abordagem em IA para problemas em sistemas elétricos de potência.

A utilização destas técnicas deve ser feita segundo algumas diretrizes. Estas diretrizes auxiliam que o desempenho de um classificador seja constantemente melhorado através da utilização correta de cada passo do processo.

Verificou-se também a importância da etapa de pré-processamento dos dados, uma vez que isto favorece significativamente a técnica de extração de conhecimento. Uma dos principais problemas enfrentados por técnicas de AAC é a *madição da dimensionalidade* onde, para se ter um vetor de variáveis de entrada grande, necessita-se de um requerimento de dados muito

grande. Como isso geralmente não é possível, então ocorre a necessidade da redução da dimensionalidade deste vetor.

Outra questão abordada diz respeito de como o mecanismo de AAC infere na base de dados. Duas opções são avaliadas: a seqüencial (individual) e global (conjunta). Na seqüencial, a idéia é que cada variável seja observada individualmente, dividindo o conjunto de dados. Na global, as correlações entre as variáveis são analisadas. O resultado é que a avaliação conjunta favorece a precisão da técnica.

Em aprendizado por máquina foi apresentado o desenvolvimento teórico do algoritmo ID3 de árvore de decisão. A idéia central do algoritmo é testar variáveis que proporcionam a melhor partição do conjunto de treinamento. Entretanto, tal algoritmo tem a impossibilidade de lidar com dados contínuos e pode gerar árvores grandes e não interpretáveis.

Uma proposta é feita no sentido de possibilitar este a lidar com dados contínuos. Isso é feito através do algoritmo C4.5, o qual é utilizado neste trabalho. Este apresenta a possibilidade lidar também com dados que apresentam ruídos e tem um processo de poda da árvore baseado na extração de regras de produção.

Neste capítulo foram mostrados alguns conceitos sobre o comportamento de uma rede neural e as diferenças entre topologias com uma única camada e com camadas intermediárias. Com visto, as RNAs-SL só conseguem lidar com dados que são linearmente separáveis, enquanto que as RNAs-ML são capazes de lidar com dados não lineares e proporcionar uma aproximação funcional bastante precisa.

A apresentação de um exemplo de extração de regras de produção feita para um neurônio apenas mostrou como um processo neural pode ser transformado em conhecimento simbólico. Vários algoritmos foram observados na literatura, constituindo um estado-da-arte para o tema de extração de conhecimento simbólico de redes neurais.

Para utilizar um algoritmo que permitisse extrair fatos relevantes de uma base de dados em avaliação de segurança de tensão, foi proposta a utilização do algoritmo VIA, descrito em detalhes na seção anterior. Aliado a este algoritmo, a técnica OET2, algoritmo de treinamento baseado em mínimos quadrados, foi utilizada como agente de síntese de conhecimento sobre um problema. A técnica OET2 pode tratar grandes bases de dados, com alta velocidade de treinamento.

A combinação do OET2 e VIA mostrou-se bastante promissora, apesar do VIA necessitar de exploração do espaço de hipóteses para completar o trabalho de *Data Mining*. Os resultados têm mostrado que as variáveis selecionadas inicialmente pelo VIA para compor as regras

extraídas são relevantes ao processo, embora essa afirmação só possa ser verificada em sua plenitude caso um processo global de gerar e testar hipóteses seja implementado. Entretanto, como observado no exemplo para o sistema elétrico de 3 barras e pelos resultados encontrados no Capítulo 4, as regras extraídas de uma rede neural para o problema de avaliação de segurança de tensão mostraram-se mais precisas que as obtidas de uma Árvore de Decisão.

Capítulo 4

SIMULAÇÕES COMPUTACIONAIS E RESULTADOS

4.1 INTRODUÇÃO

Neste capítulo são apresentados os resultados das simulações computacionais visando extrair regras de produção de redes neurais artificiais. Estas simulações são aplicadas no problema de avaliação de segurança de tensão. Para a obtenção dos resultados, é necessário realizar os seguintes procedimentos: formação da base de dados, seleção de variáveis, treinamentos da rede neural OET2 e da árvore de decisão C4.5 e aplicação do algoritmo VIA. Os resultados finais apresentam as comparações entre as regras extraídas pelo C4.5 e as extraídas pelo VIA.

4.2 BASE DE DADOS – CRITÉRIOS DE FORMAÇÃO

A Figura 4.1 apresenta o diagrama do aplicativo **VSA GERDATA** utilizado para gerar a base de dados dos sistemas elétricos em estudo. As palavras sublinhadas denotam os nomes dos programas computacionais desenvolvidos no âmbito deste trabalho e as caixas pontilhadas são programas obtidos de fontes externas. Para cada sistema elétrico, o objetivo do aplicativo é gerar um conjunto de informações para um caso base escolhido e suas contingências mais prováveis.

Vale salientar que o aplicativo **VSA GERDATA** é desenvolvido conforme as orientações fornecidas no Capítulo 2 para o desenvolvimento de ferramentas para avaliação de segurança de tensão.

O processo inicia-se estabelecendo um caso base e executando o fluxo de carga para este ponto de operação. Este resultado constitui o ponto inicial de operação estável. O (MC) é aplicado (subprograma **MetCon**) a partir deste ponto, gerando informações sobre o sistema elétrico até o ponto de colapso de tensão. O resultado é um conjunto de pontos de operação do sistema, configurando o diagrama de bifurcação, o qual é armazenado sob a forma de um arquivo de dados.

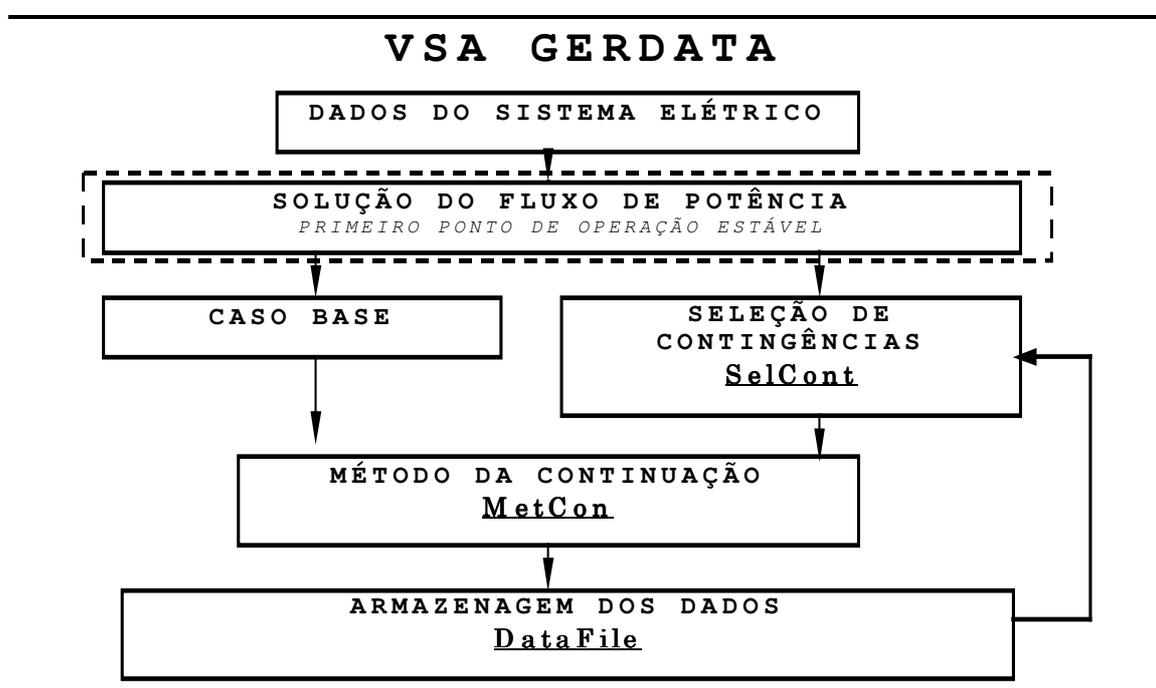


Figura 4.1 - Estrutura geral do aplicativo para avaliação de estabilidade de tensão e obtenção da base de dados para as simulações do algoritmo VIA.

Uma vez que a forma de aquisição automática de conhecimento implementada neste trabalho é feita através de um processo indutivo (baseado em exemplos), cada um dos pontos de operação constitui um exemplar do caso simulado. Logo, para armazenar o conjunto dos exemplares obtidos para o caso base simulado, é necessário escolher previamente as grandezas que devem compor o vetor das variáveis de entrada, tais como tensões, potências, etc. Uma boa escolha destas pode melhorar a precisão dos resultados, uma vez que algumas são mais relacionadas ao problema.

Na seqüência, após o processamento do caso base, o programa oferece a possibilidade de selecionar qualquer equipamento a ser retirado (subprograma **SelCont**), configurando uma nova

topologia para o sistema. O MC é então aplicado a este caso. Novamente, os resultados são armazenados como descrito anteriormente pelo subprograma **DataFile**. O processo se repete para qualquer contingência selecionada. Isto produz um conjunto de arquivos, constituindo a base de dados.

O aplicativo **VSA GERDATA** é desenvolvido em linguagem FORTRAN 90, utilizando-se o pacote computacional Fortran Power Station 6.0 da Compaq [FOR95]. As características de cada subprograma do aplicativo são mostradas a seguir. Para tornar as simulações mais realistas, foram implementados em um programa de fluxo de carga convencional, recursos como LTCs, despacho de geradores, crescimento de carga diferenciado por área e/ou barras e modelo ZIP de cargas.

Características do fluxo de carga:

- O método numérico de solução das equações algébricas é o Newton-Raphson.
- Podem ou não ser considerados os limites de fornecimento de potência reativa.
- Apresenta equipamento LTC controlando uma barra adjacente ou remota.
- Apresenta a possibilidade de modelagem ZIP da carga (PV^α , QV^α) por barra, admitindo a composição de modelos, como por exemplo 40% impedância constante e 60% potência constante.

Características do Método da Continuação:

- Neste desenvolvimento é utilizada a variação para obtenção do ponto de solução estável (V^{k+1} e θ^{k+1}) do MC. Isto é, utiliza-se o ponto atual V^k e θ^k como inicialização para o passo corretor, dado pela solução das equações algébricas do fluxo de carga.
- Possibilita diferenciar o crescimento da carga por área do sistema.
- Os valores das potências ativa e reativa de cada barra são perturbados aleatoriamente em percentuais definidos para cada caso. Isto representa um crescimento da carga diferenciado por barra e, obviamente, proporciona fatores de potência diferenciados em cada barra de carga, a cada novo ponto de operação.
- Para cada barra do sistema que contém geração de potência ativa, é estipulado um gerador ou um grupo de geradores atuando nesta, com exceção da barra *swing*. Isto é implementado para estabelecer um critério de despacho de geração. O critério aqui adotado é simples, levando em conta o fator de participação desta barra na composição da potência ativa total do sistema.
- O ponto de colapso considerado é o último ponto de operação encontrado antes da

divergência do fluxo de carga. Entretanto, à medida que o incremento na carga $\Delta\lambda$ torna-se menor que a tolerância do erro, esta é novamente ajustada abaixo do valor deste incremento.

Características gerais do programa VSA GERDATA:

- No presente programa, apenas contingências simples são consideradas, envolvendo perda de banco de capacitores, perdas de linhas de transmissão e transformadores e perdas de geradores.
- Pode-se selecionar, a partir do caso base escolhido, cada contingência possível do sistema. Estas são automaticamente verificadas e numeradas pelo programa quando o fluxo de carga para o caso base é executado. Isso possibilita gerar o diagrama de bifurcação para cada caso escolhido. Os resultados são apresentados em arquivos individuais para cada caso.
- As informações contidas nos arquivos de saída podem ser escolhidas de acordo com o estudo, escolhendo-se a composição do vetor de entrada (Ex.: tensões, potências ativa e reativa das barras PQs, o parâmetro de crescimento de carga).

De acordo com o apresentado acima, pode ser observado que o **VSA GERDATA** apresenta a possibilidade de ser utilizado como ferramenta de estudos em estabilidade de tensão, não sendo um aplicativo destinado exclusivamente a geração de dados. Na direção de torna-lo um programa mais realista, as equações de atualização das potências ativa e reativa de cada barra podem ser adaptadas para levar em conta algumas das características citadas e são dadas por:

$$\begin{aligned} P_{L_i} &= P_{L0_i} (1 + \lambda a1_j) + a21_i (P_{L0_i} (1 + \lambda a1_j)) \\ Q_{L_i} &= Q_{L0_i} (1 + \lambda a1_j) + a22_i (Q_{L0_i} (1 + \lambda a1_j)) \end{aligned} \quad (4.1)$$

onde:

- P_{L0} e Q_{L0} - Potências ativa e reativa iniciais em cada barra i
- $a1$ - Fator de crescimento diferenciado da carga por área j
- $a21$ e $a22$ - Fator de crescimento aleatório da carga por barra, referente às potências ativa e reativa, respectivamente.
- λ - Valor atualizado do fator de crescimento da carga, dado pelo MC, lembrando que $\lambda_{\text{atual}} = \lambda_{\text{anterior}} + \Delta\lambda$.

Um exemplo da aplicação da Equação 4.1, feito para o caso base do sistema teste IEEE

14 barras (Apêndice G) é mostrado na Figura 4.2. Nesta simulação, o fator $a1$ é feito igual a 1 em todas as barras do sistema. Observando-se estas curvas e comparando-as com as mostradas na Figura 2.4, Capítulo 2, nota-se o comportamento extremamente não linear, diferentemente do comportamento quase monotônico anteriormente obtido. Este comportamento deve-se aos fatores $a21$ e $a22$ introduzidos.

A introdução destes fatores resulta em um fator de potência diferente em cada ponto de operação encontrado. O efeito é a constituição de uma família de curvas PV em que o sistema excursiona até o colapso de tensão. Uma consequência observa-se que a mesma significativamente reduzida, dependendo do valor da perturbação aleatória introduzida. Nesta simulação, a faixa de variação aplicada aos fatores $a21$ e $a22$ é de $\pm 15\%$.

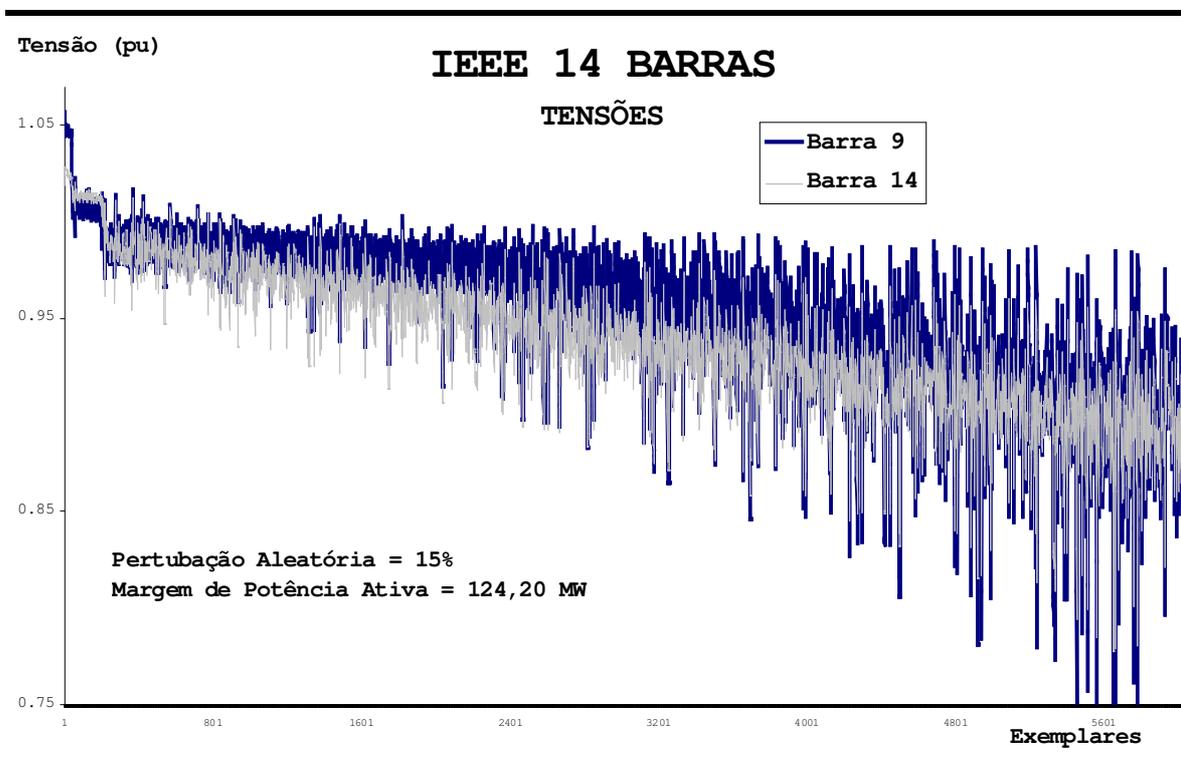


Figura 4.2- Curvas PV para o caso base, considerando as ações do LTC e do despacho de geração, FP aleatório e limites de fornecimento de potência reativa.

Na Figura 4.2, duas curvas de tensões são mostradas, uma referente à Barra 9 (regulada pelo LTC) e a outra referente à Barra 14 (obtida como crítica, através do vetor tangente). Pode-se notar que a variação no fator de potência afeta mais a região próxima do ponto de colapso (final da curva). Neste caso, para uma pequena variação na direção de crescimento de carga produz maior variação no resultado. Isto demonstra que a característica da carga, representada por esta variação no fator de potência, influencia no fenômeno da estabilidade de tensão.

Seleção das variáveis - base de dados

A base de dados a ser formada, como já informado, é composta de exemplares que contém variáveis que são mapeadas na solução do problema. Exemplos destas em sistemas de potência são as tensões, as potências ativas e reativas, os fluxos de potência, perdas elétricas em linhas de transmissão, etc. Obviamente, cada problema pode ser melhor representado por um conjunto de variáveis que proporcionam um maior ganho de informação qualitativa. De acordo com [MEN99], uma boa escolha para estabilidade de tensão recai sobre as impedâncias características das barras de carga do sistema. Além destas variáveis, neste trabalho são selecionadas as tensões destas mesmas barras para compor o vetor de entrada.

Uma vez escolhidas as variáveis para compor a informação sobre o sistema elétrico, é necessário escolher o rol das barras do sistema que devem fazer parte da base de dados. Neste trabalho, as barras escolhidas são as inicialmente PQs. Se, por exemplo, a barra 56 originalmente PQ de um sistema elétrico qualquer é selecionada, então a tensão e a impedância característica desta barra irão compor o vetor de entrada. Para efeito de padronização dos resultados, a tensão na barra 56, por exemplo, é representada por **V56** e a impedância característica por **ZI56**. Logo, o vetor de entrada tem a dimensão de 2 vezes o número de barras selecionadas.

Para o sistema teste IEEE 14 barras e 118 barras, a Tabela 4.1 mostra a distribuição dos atributos e variáveis que compõe o vetor de entrada. Para o sistema 14 barras existem 9 barras originalmente PQs. Entretanto, a barra 7 é apenas uma barra de transferência e, obviamente apresenta impedância igual a zero, reduzindo o vetor de entrada em uma variável. Para o sistema 118 barras, são 64 barras originalmente PQs, com as barras 5, 9, 30, 37, 38 63, 64, 68, 71 e 81 com impedância característica igual a zero.

Tabela 4.1 - Variáveis selecionadas os sistemas testes IEEE.

CASO BASE	TENSÕES (V)	IMPEDÂNCIAS (ZI)	TOTAL DE VARIÁVEIS
14 barras	9 var	8 var	17
118 barras	64 var	54 var	118

4.3 SIMULAÇÕES

Nesta seção são apresentados os estudos comparativos entre a informação qualitativa

provida pelas regras simbólicas do C4.5 e as do VIA. Isto é feito através da composição de um índice o qual é aplicado às regras obtidas por cada um destes algoritmos. O objetivo, como já informado, é mostrar que a utilização das RNAs promove ganhos de informação em problemas de avaliação de segurança de tensão quando comparadas com as árvores de decisão.

O programa **VSA DATA MINING** mostrado na Figura 4.3 apresenta as etapas do processo para a obtenção dos resultados. Os nomes sublinhados denotam subprogramas do aplicativo desenvolvidos no âmbito deste trabalho. Os subprogramas dentro das caixas pontilhadas são aplicativos obtidos de fontes externas. A seguir, será apresentada a descrição de cada etapa deste processo.

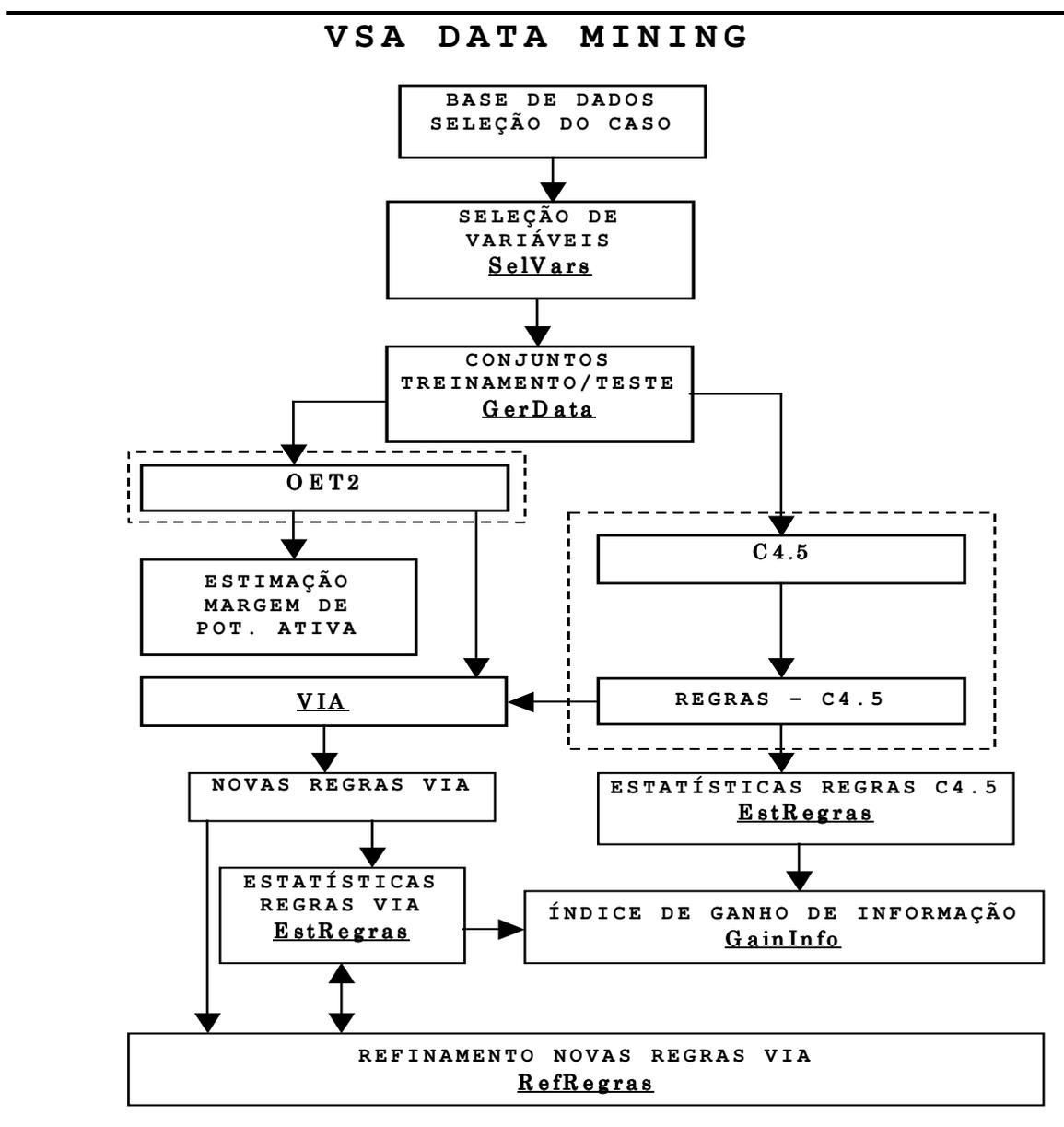


Figura 4.3 – Fluxograma do Programa VSA DATA MINING para obtenção das regras de produção em AST.

4.3.1 Seleção do caso

O programa acessa a base de dados, selecionando o caso em estudo. Outro dado importante acessado nesta etapa é a relação das barras críticas por ponto de operação. Vale lembrar que os casos e o rol de barras críticas estão armazenados numa base de dados construída pelo programa **VSA GERDATA**, anteriormente descrito.

4.3.2 Seleção de variáveis

Esta etapa é executada pelo subprograma `SelVars` e visa reduzir o número de variáveis inicialmente selecionadas para compor o vetor de entrada, montado na etapa de formação da base de dados. A redução no número de variáveis do vetor de entrada é devido a dois fatores: a *maldição da dimensionalidade* (Capítulo 3) e versão do programa de DT C4.5 utilizada, que pode processar, no máximo, conjuntos de treinamento/teste de com 400 exemplares. Como o número de exemplares está restrito, tem-se então maior necessidade de redução das variáveis que fazem parte do vetor de entrada, conforme mencionado no Capítulo 3.

Todavia, vale salientar que uma das maiores qualidades da rede OET2 fica prejudicada, que é lidar com conjuntos de dados com grande cardinalidade de forma rápida e precisa. Esta propriedade é mostrada nas simulações para obtenção da margem de potência ativa (problema de regressão).

Nas simulações que se seguem para os sistemas teste envolvidos é adotado um critério para a redução destas variáveis. Lembrando que o Método do Vetor Tangente (Capítulo 2) proporciona a indicação do *ranking* das barras críticas do sistema elétrico em cada ponto de operação, pode-se processar uma avaliação deste *ranking* por nível operativo. Assim, o critério adotado é a combinação entre o *ranking* das barras críticas por nível operativo e a frequência com que as mesmas aparecem entre as primeiras ' x ' posições. Neste trabalho, ' x ' é escolhido igual a cinco (cinco primeiras posições) por nível operativo. Primeiramente, a Tabela 4.2 apresenta os dados gerais obtidos da simulação do método da continuação para o caso base, mostrando a distribuição de exemplares.

A Tabela 4.3 mostra então a distribuição das frequências das cinco barras críticas que mais aparecem entre as cinco primeiras posições do *ranking*. Observa-se que as barras 14 e 13 aparecem entre as 5 primeiras posições das barras críticas para todo o conjunto de dados, independente do nível operativo. Isso mostra a sensibilidade do vetor tangente em relação à barra

crítica do sistema. Eventualmente, em alguns casos simulados, pode ser levada em conta a informação adicional de uma barra nitidamente predominante em apenas um dado nível operativo. Desta forma, o número de variáveis (' x ') que farão parte do vetor de entrada é aumentado. Para a padronização dos resultados, a variável tensão em uma barra é referenciada como V .

Tabela 4.2 – Conjunto de dados para o caso base do sistema teste IEEE 14 barras.

Caso base:

- Exemplos do conjunto de dados (trein. + teste): 6020
- Variáveis originais: 17 (9 tensões - 8 impedâncias)
- Exemplos no nível operativo **normal**: 2428
- Exemplos no nível operativo **alerta**: 1191
- Exemplos no nível operativo **inseguro**: 1207

Tabela 4.3 – Frequência das barras críticas (5 posições) em função do nível operativo do sistema elétrico teste IEEE 14 barra.

BARRA	FREQUÊNCIA			
	<i>Geral</i>	<i>Normal</i>	<i>Alerta</i>	<i>Inseguro</i>
14	6020	2826	1589	1605
13	6020	2826	1589	1605
12	6009	2815	1589	1605
11	6009	2815	1589	1605
10	6006	2812	1589	1605

4.3.3 Conjuntos de treinamento e teste

A formação dos conjuntos de treinamento e teste para cada caso simulado é feita sorteando-se aleatoriamente os exemplares de tal forma que os dois conjuntos tenham no máximo 400 exemplares pela restrição do programa C4.5. Além disso, a frequência de exemplares entre as classes é mantida praticamente a mesma. Ou seja, se um dos conjuntos tem 399 exemplares, cada classe para esse conjunto contém 133 exemplares. Essa distribuição é feita através do subprograma, GerData, que promove o sorteio aleatório, distribuindo os dados conforme este critério. Esse procedimento é aplicado a todos os casos simulados.

Outra opção introduzida é a possibilidade de treinar o OET2 utilizando duas codificações diferentes para representar as classes do problema. Foram utilizadas as codificações '1 de N' e 'Temperatura'. Na primeira, a classe do respectivo exemplar é representada pela codificação '1', enquanto as demais são representadas por '-1'. Na 'Temperatura', a codificação é feita

representando uma combinação de '1' e '-1' para cada classe. Assim, para a classe Normal, a codificação adotada aqui é '-1' e '-1', Alerta '1' e '-1' e Inseguro '-1' e '-1'. Observa-se que, com esta codificação, é possível representar até quatro classes. Além disso, tal representação está de acordo com a forma sequencial (isto é, com relação de ordem) como são mapeados os estados operativos do problema de avaliação de segurança de tensão. Conseqüentemente, essa codificação 'Temperatura' é adotada. As taxas de classificação do OET2 utilizando a mesma mostraram-se melhores que a codificação '1 de N'.

4.3.4 OET2

Nesta etapa, Os conjuntos de treinamento e teste são processados pela RNA OET2. Os pesos das conexões e valores das saídas dos neurônios por classe são armazenados e é obtida a taxa de erro de classificação para o conjunto de teste. O objetivo é verificar a qualidade do treinamento da rede neural, uma vez que esta é a condição necessária para um melhor desempenho do VIA. Isto serve também para observar o critério da *fidelidade* das regras refinadas pelo VIA.

Outra verificação feita nesta etapa é quanto às variáveis presentes que tenham dependência linear. Isto é necessário porque métodos baseados em mínimos quadrados, como é o algoritmo de treinamento OET2 (Apêndice C), apresentam sensível melhora em seus desempenhos quando as dependências lineares são eliminadas. Logo, as variáveis encontradas como tal são excluídas dos conjuntos de treinamento e teste. A Tabela 4.4 a seguir mostra a estatística de classificação obtida pelo OET2 para o conjunto de teste gerado a partir do caso base do sistema teste IEEE 14 barras. Nestes resultados, as variáveis eliminadas por dependência linear foram as V13 e V11.

Observa-se que os maiores erros de classificação ocorrem na classe Inseguro pois o aumento da não linearidade nos dados é aumentada significativamente nesta região. De uma forma geral, regras da classe Alerta classificando exemplares da Inseguro (falso negativo) e regras da classe Inseguro classificando exemplares da Alerta (falso positivo) são situações indesejáveis em AST. Então, a utilização de uma ferramenta para classificação de níveis operativos deve apresentar menores erros nessas regiões.

Uma vez executado o OET2, para o processamento do VIA é necessário armazenar as informações dos pesos das conexões e os valores máximos e mínimos das variáveis de saída associadas às classes de interesse. São armazenados também os valores máximos e mínimos de

cada variável de entrada.

Tabela 4.4 Taxa de classificação do OET2. Os valores sublinhados denotam classificação correta. Os demais, classificação errada. Mostra também o total geral da classificação – Caso base sistema teste IEEE 14 barras.

Normal	Alerta	Inseguro	Total
<u>123</u>	9	0	132
5	<u>117</u>	9	132
0	19	<u>115</u>	133
Total Geral	Class. Correta		Class. Errada
	89.42% (355/397)		10.57% (42/397)

4.3.5 Estimação da Margem de Potência Ativa

Nesta etapa, a RNA OET2 é utilizada para o problema de estimar a margem de potência ativa para o sistema em estudo. Os gráficos mostrados nas Figuras 4.4 e 4.5 apresentam as curvas referentes ao valor alvo e ao valor estimado para o caso base do sistema 14 barras para os valores de perturbação aleatória de 5% e 15%, respectivamente.

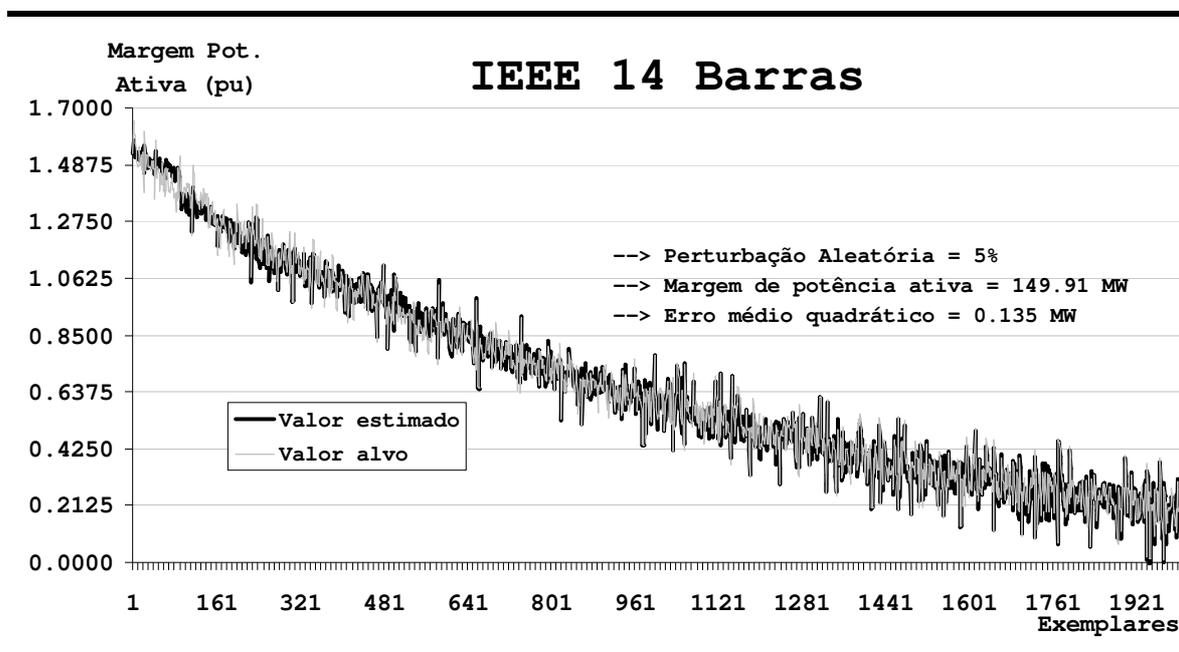


Figura 4.4 - Erro médio quadrático para o caso base utilizando a RNA OET2 para estimação da margem de potência ativa (pu) com uma perturbação aleatória de 5%.

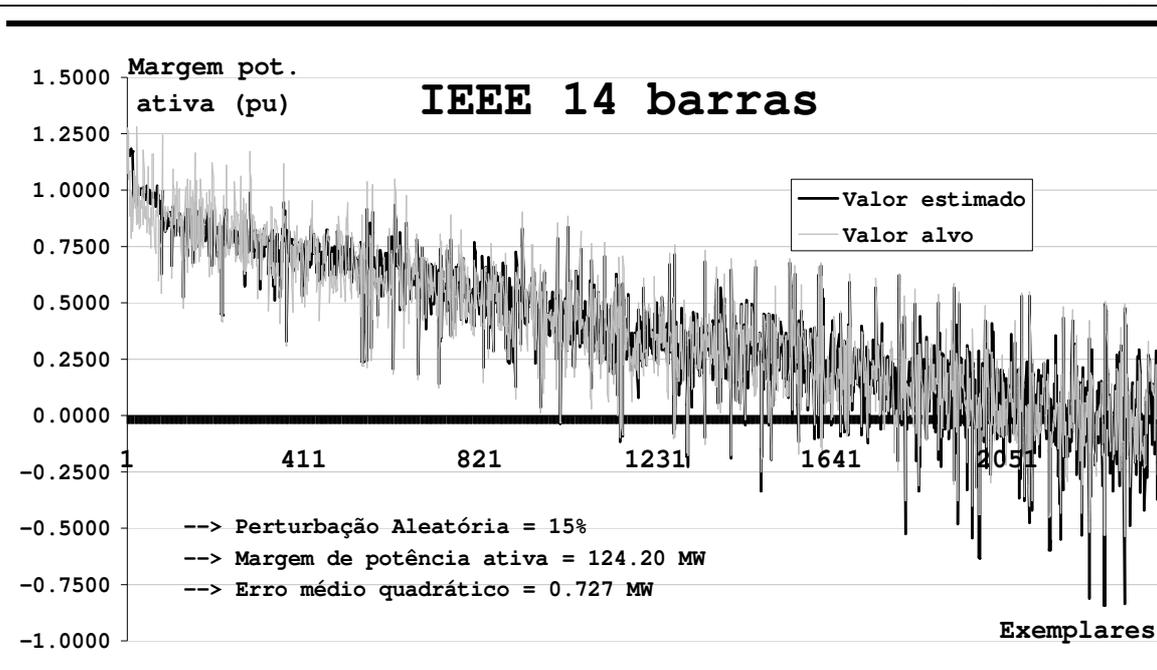


Figura 4.5 - Erro médio quadrático para o caso base utilizando a RNA OET2 para estimação da margem de potência ativa (pu) com uma perturbação aleatória de 15%.

Observa-se que o erro médio quadrático aumenta muito com o aumento da perturbação aleatória. Outra constatação do aumento da perturbação aleatória é a redução da margem de potência ativa. Nota-se que a margem encontrada para uma perturbação de 5% é próxima da encontrada para a simulação sem a mesma, mostrada na Figura 2.4, Capítulo 2.

4.3.6 C4.5 e Regras C4.5

Nesta etapa, os conjuntos de treinamento e teste de cada caso são processados pelo algoritmo C4.5. Vale salientar que estes conjuntos não contemplam mais as variáveis com dependência linear encontradas pelo OET2. O processamento do conjunto de treinamento possibilita a obtenção da DT para o caso estudado. A Tabela 4.5 mostra a árvore de decisão obtida do processamento do C4.5.

A Tabela 4.6 apresenta a estatística para o conjunto de teste do caso base do sistema teste IEEE 14 barras. Nesta tabela, os valores na diagonal mostram os exemplares classificados corretamente. Os outros elementos mostram os exemplares classificados de forma errada, ou seja, uma regra pertencente a uma dada classe classificando exemplares pertencentes a uma outra (regras não discriminatórias). Comparando-se este resultado obtido pela árvore do C4.5 com o resultado obtido pelo OET2 (Tabela 4.4), observa-se que além da precisão global ser maior para

o OET2, os erros correspondentes ao falso positivo (exemplar Alerta classificado como Inseguro) e ao falso negativo (exemplar Inseguro classificado como Alerta) são menores no OET2. Isso mostra que as RNAs são mais precisas que as DTs para a região da curva PV próxima ao ponto de colapso de tensão, como já apresentado no Capítulo 3 deste trabalho.

Tabela 4.5 – Árvore de decisão obtida pelo C4.5 para o Caso Base.

```

V12 > 0.952: normal
V12 <= 0.952:
: ... V10 > 0.933: alerta
      V10 <= 0.933:
: ... V10 <= 0.912: inseguro
      V10 > 0.912:
: ... ZI12 > 10.077: inseguro
      ZI12 <= 10.077:
: ... ZI14 <= 3.585: alerta
      ZI14 > 3.585:
: ... V12 <= 0.901: inseguro
      V12 > 0.901:
: ... ZI11 <= 14.805: alerta
      ZI11 > 14.805: inseguro
    
```

Tabela 4.6 Taxa de classificação da árvore de decisão extraída pelo C4.5. Os valores sublinhados denotam classificação correta. Os demais, classificação errada. Mostra também o total geral da classificação – Caso base sistema teste IEEE 14 barras.

Normal	Alerta	Inseguro	Total
<u>122</u>	10	0	132
12	<u>115</u>	4	131
0	29	<u>105</u>	134
Total Geral	Class. Correta		Class. Errada
	<u>86.15% (342/397)</u>		<u>13.85% (55/397)</u>

Em seguida, o programa utiliza-se da metodologia descrita no Capítulo 3 para extrair regras de produção da DT. Estas regras constituem as hipóteses iniciais a serem testadas pelo algoritmo VIA. As regras extraídas para o caso base do sistema teste IEEE 14 barras são mostradas na Tabela 4.7.

Os antecedentes destas regras de produção são as variáveis previamente descritas, sendo que a designação destas está de acordo com a Tabela 4.1. Observa-se que cada variável presente na regra apresenta apenas um valor de corte devido a discretização das variáveis contínuas feita pelo C4.5, reforçando o que foi mostrado no Capítulo 3.

Tabela 4.7 - Regras C4.5 - Caso base sistema teste IEEE 14 barras.

<p>Regra 1: V12 > 0.952 -> classe normal</p> <p>Regra 2: V12 <= 0.952 V10 > 0.933 -> classe alerta</p> <p>Regra 3: V12 > 0.901 ZI14 > 3.585 ZI12 <= 10.077 ZI11 <= 14.805 -> classe alerta</p>	<p>Regra 4: V10 > 0.912 V10 <= 0.933 ZI14 <= 3.585 ZI12 <= 10.077 -> classe alerta</p> <p>Regra 5: V12 <= 0.952 -> classe inseguro</p>
--	--

4.3.7 Estatísticas regras C4.5

As regras obtidas pelo C4.5 são avaliadas quanto a sua capacidade de generalização utilizando o conjunto de teste, composto de casos não avaliados na etapa de treinamento. Esta estatística é importante pois estas taxas de classificação das regras obtidas pelo C4.5 são comparadas com as regras extraídas pelo VIA.

A Tabela 4.8 mostra as taxas de classificação (e respectivas taxas de erros) de cada conjunto de regras extraídas para uma dada classe. São avaliados os erros de não classificação de cada regra (exemplares que deveriam ser cobertos por uma dada regra) e os erros de classificação indevida (exemplares de uma classe classificados erroneamente por uma outra regra).

Tabela 4.8 Taxa de classificação do conjunto de regras extraída pelo C4.5. Os valores sublinhados denotam classificação correta. Os demais, classificação errada. Mostra também o total geral da classificação – Caso base sistema teste IEEE 14 barras.

Normal	Alerta	Inseguro	Total
<u>122</u>	10	0	132
12	<u>115</u>	4	131
0	29	<u>105</u>	134
Total Geral	Class. Correta		Class. Errada
	<u>86.15% (342/397)</u>		<u>13.85% (55/397)</u>

Entretanto, as estatísticas mostradas nesta Tabela 4.8 são obtidas através de uma heurística proposta pelos autores do programa C4.5 utilizado neste trabalho. A heurística é

baseada em testes de hipótese estatística entre as regras, verificando qual delas é a mais provável em classificar o respectivo exemplar corretamente. Este procedimento evita que um exemplar seja classificado mais de uma vez, mesmo que incorretamente.

A Tabela 4.9 apresenta o resultado quando a avaliação é feita olhando-se cada regra individualmente, isto é, sem o teste de hipótese mencionado. Os valores indicados entre parênteses nesta tabela indicam os exemplares correta ou incorretamente classificados pelo número total de exemplares avaliados.

Tabela 4.9 - Taxa de classificação por regra do C4.5 para o caso base do sistema teste IEEE 14 barras.

Regra	Normal	Alerta	Inseguro	Total Class. Correta	Total Class. Errada
01	<u>126</u>	15	0	94.45% (126/132)	4.66% (15/265)
02	12	<u>110</u>	17	89.31% (117/131)	16.17% (43/266)
03	5	<u>9</u>	3		
04	2	<u>3</u>	5		
05	33	96	<u>134</u>	100.00% (134/134)	49.05% (129/263)
Total Geral	Class. Correta			Class. Errada	
	<u>94.96% (377/397)</u>			<u>47.10% (187/397)</u>	

Comparando-se as estatísticas mostradas nas Tabelas 4.8 e 4.9, observam-se casos onde há melhora na taxa de classificação e em outros não. Entretanto, olhando-se a taxa global, apesar do número de exemplares corretamente classificados ter aumentado, o número de classificações erradas também aumentou. Nota-se também que as taxas de classificação totais refletem a possibilidade de um exemplar ser classificado por mais de uma regra.

No âmbito deste trabalho, a estatística proporcionada pela Tabela 4.8 é comparada com a estatística obtida pelo OET2. Já a estatística para as regras individuais, dada pela Tabela 4.9, tem o propósito de auxiliar as comparações entre as regras obtidas pelo C4.5 e as obtidas pelo VIA.

4.3.8 VIA

O processamento do VIA visa obter novas regras a partir das hipóteses inicialmente formuladas pelo C4.5 (regras – Tabela 4.7). A metodologia empregada para este procedimento

está detalhada no Capítulo 3. Vale lembrar que este procedimento visa minimizar o processo de gerar e testar regras processadas pelo VIA, uma vez que o espaço de busca muito grande. Um exemplo de regras extraídas pelo VIA a partir das hipóteses geradas pelo C4.5 para o caso base do sistema teste IEEE 14 barras é mostrado na Tabela 4.10. Esta tabela segue o formato apresentado no Capítulo 3, na exemplificação para o sistema teste de 3 barras.

Tabela 4.10 – Regras originais C4.5 e as novas obtidas pelo VIA.

INTERVALOS ORIGINAIS			INTERVALOS NOVOS			Alteração(%)
Min.	Var.	Max.	Min.	Var.	Max.	
REGRA NUM. == 1 ----> CLASSE NORMAL						
0.7320	V14	0.9890	0.8934	V14	0.9890	62.80
0.7740	V10	1.0070	0.9155	V10	1.0070	60.75
0.9520	V12	1.0140	0.9520	V12	1.0140	0.00
REGRA NUM. == 2 ----> CLASSE ALERTA						
0.7320	V14	0.9890	0.8292	V14	0.9709	44.86
0.7560	V12	0.9520	0.8175	V12	0.9520	31.40
0.9330	V10	1.0070	0.9330	V10	1.0067	0.40
REGRA NUM. == 3 ----> CLASSE ALERTA						
0.7740	V10	1.0070	0.8985	V10	0.9876	61.74
0.7320	V14	0.9890	0.8783	V14	0.9890	56.91
0.9010	V12	1.0140	0.9010	V12	1.0140	0.00
3.5850	ZI13	6.8430	3.5850	ZI13	6.8430	0.00
4.8640	ZI12	10.0770	4.8640	ZI12	10.0770	0.00
9.8270	ZI11	14.8050	9.8270	ZI11	14.8050	0.00
REGRA NUM. == 4 ----> CLASSE ALERTA						
0.7320	V14	0.9890	0.7983	V14	0.9577	37.97
0.7560	V12	1.0140	0.7948	V12	0.9890	24.71
0.9120	V10	0.9330	0.9120	V10	0.9330	0.00
2.2130	ZI13	3.5850	2.2130	ZI13	3.5850	0.00
4.8640	ZI12	10.0770	4.8640	ZI12	10.0770	0.00
REGRA NUM. == 5 ----> CLASSE INSEGURO						
0.7320	V14	0.9890	0.7320	V14	0.9685	7.98
0.7740	V10	1.0070	0.7740	V10	0.9937	4.72
0.7560	V12	0.9520	0.7560	V12	0.9520	0.00

A informação da *alteração* percentual de cada intervalo das variáveis presentes nas regras tem o objetivo de avaliar a sensibilidade do treinamento do OET2 em relação a estas variáveis. Além disso, a informação da variável que sofre a maior alteração percentual em cada regra é utilizada para proceder um novo refinamento destas regras, quando necessário. Isso é feito de acordo com a metodologia proposta no Capítulo 3.

Uma questão a ser observada é o valor da tolerância para a convergência do VIA. Este valor deve ser testado para cada caso, observando-se a relação tempo de processamento versus

qualidade do resultado. Um valor de tolerância muito baixo pode determinar uma convergência lenta, fora dos padrões desejados.

4.3.9 Estatísticas regras VIA:

As novas regras extraídas do VIA são avaliadas para conjunto de teste. Nesta etapa são então verificadas as taxas de erro de classificação do novo conjunto de regras. Esta estatística é importante pois mostra, juntamente com o índice de ganho proposto a seguir, o desempenho de cada nova regra frente à hipótese inicial formulada pelo C4.5.

A Tabela 4.11 apresenta os resultados estatísticos para as regras obtidas pelo VIA para o caso base do sistema teste IEEE 14 barras. Observa-se que os resultados para as taxas de classificação foram os mesmos obtidos pelo C4.5. Entretanto, conforme mostra a Tabela 4.10, novas variáveis foram incorporadas às regras originais. Este fato é devido à forma como as redes neurais manipulam as variáveis de entrada, como descrito no Capítulo 3. Assim, o OET2 pode correlacionar um maior número de variáveis a uma classe de interesse. Esse fato é captado pelo VIA, refletindo nas novas regras extraídas. Na realidade, o VIA procura as variáveis (e respectivos intervalos mais prováveis) para obter a região factível da solução em torno do ponto inicial, dado pela hipótese do C4.5.

Tabela 4.11 – Taxa de classificação por regra do VIA para o caso base do sistema teste IEEE 14 barras.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>126</u>	15	0	94.45% (126/132)	4.66% (15/265)
02	12	<u>110</u>	17	89.31% (117/131)	16.17% (43/266)
03	5	<u>9</u>	3		
04	2	<u>3</u>	5		
05	33	96	<u>134</u>	100.00% (134/134)	49.05% (129/263)
Total Geral	Class. Correta			Class. Errada	
	<u>94.96% (377/397)</u>			<u>47.10% (187/397)</u>	

Se o processo de treinamento de uma rede neural proporciona que o nível de informação sobre o problema em análise aumente, a idéia é obter um índice que mostra o ganho de

informação que por ventura ocorra, das regras obtidas pelo VIA em relação às do C4.5. Isto é mostrado no item 4.3.10.

4.3.10 Índice de ganho de informação:

Uma maneira de verificar se as regras obtidas do VIA são mais efetivas que as do C4.5, é analisando o conteúdo específico de cada regra. Isto pode ser feito se a informação obtida por uma regra for suficientemente abrangente para a análise do problema em estudo. Logo, pode-se propor um índice que forneça o *ganho de informação* de uma nova regra em relação à regra original. O exemplo a seguir auxilia na descrição de como é feita a obtenção deste índice. Para esse propósito, são utilizadas as regras 4 e 5 da Tabela 4.10 obtidas para o caso base do sistema IEEE 14 barras. Inicialmente, observa-se as hipóteses iniciais obtidas pelo C4.5, mostradas na Tabela 4.7 e repetidas aqui:

- **Regra 4:** Se $(V_{10} > 0.912 \text{ e } V_{10} \leq 0.933 \text{ e } Z_{I14} \leq 3.585 \text{ e } Z_{I12} \leq 10.077)$ então alerta
- **Regra 5:** Se $(V_{12} \leq 0.952)$ então inseguro

Atendo-se ao problema da avaliação de segurança de tensão, cada variável proporciona um *grau* de informação diferenciado, observado em função do *ranking* do vetor tangente. Assim, pode-se atribuir a estas variáveis um peso acordo com o *ranking* encontrado. Para este caso base, é considerada seguinte atribuição de pesos:

- **Pond** = $(V_{14} = 10.0; \underline{V_{13}} = 7.5; V_{12} = 5.0; \underline{V_{11}} = 2.5; V_{10} = 1.0; Z_{I14} = 10.0; \underline{Z_{I13}} = 7.5; Z_{I12} = 5.0; \underline{Z_{I11}} = 2.5; Z_{I10} = 1.0)$

Algumas considerações podem ser feitas a respeito da obtenção do ganho de informação de uma regra. Este ganho é máximo quando uma regra extraída contém todas as variáveis do vetor de entrada. Além disso, os respectivos intervalos destas variáveis devem estar definidos exatamente para a classe a qual a regra pertence. Isto significa que quanto mais uma regra se aproxima desta condição, mais especializada esta é. Em problemas de AST, a informação de poucas barras críticas pode não ser suficiente para auxiliar o processo tomada de decisão e controle do sistema elétrico. Assim, quanto mais uma regra for especializada, maior sua utilidade para o problema de avaliação de segurança.

Como dito, cada variável é também definida em função de seu intervalo na regra e tal

informação deve ser explorada. Logo, na comparação entre as regras extraídas do C4.5 e as do VIA, é necessário levar em conta o intervalo destas variáveis. O índice final é dado pela composição do grau de informação de uma variável com seu respectivo intervalo e define o volume de informação de cada regra. Prosseguindo na exemplificação, as novas regras 4 e 5 obtidas pelo VIA de acordo com a Tabela 4.10 são mostradas novamente abaixo. Como já observado, as variáveis V13 e V11 (sublinhadas) foram eliminadas por dependência linear e não fazem mais parte do rol de variáveis analisadas.

INTERVALOS ORIGINAIS				INTERVALOS NOVOS			
Min.	Var.	Max.		Min.	Var.	Max.	Alteração (%)
REGRA NUM. == 4 ---->				CLASSE	ALERTA		
0.7320	V14	0.9890		0.7983	V14	0.9577	37.97
0.7560	V12	1.0140		0.7948	V12	0.9890	24.71
0.9120	V10	0.9330		0.9120	V10	0.9330	0.00
2.2130	ZI13	3.5850		2.2130	ZI13	3.5850	0.00
4.8640	ZI12	10.0770		4.8640	ZI12	10.0770	0.00
REGRA NUM. == 5 ---->				CLASSE	INSEGURO		
0.7320	V14	0.9890		0.7320	V14	0.9685	7.98
0.7740	V10	1.0070		0.7740	V10	0.9937	4.72
0.7560	V12	0.9520		0.7560	V12	0.9520	0.00

Do exposto nos parágrafos anteriores, para aplicar o índice de ganho de informação para estes conjuntos de regras, são necessários os seguintes passos:

1º - Cálculo do *índice da variável i* ($i = 1, \dots, n$ variáveis) pertencente à regra j ($j = 1, \dots, m$ regras) no C4.5 e no VIA, de acordo com o respectivo intervalo obtido. Isto é dado pelas Equações 4.2 e 4.3:

$$IDT_{i,j} = \frac{VIA_{max_{i,j}} - VIA_{min_{i,j}}}{DT_{max_{i,j}} - DT_{min_{i,j}}} \tag{4.2}$$

$$IVIA_{i,j} = \frac{DT_{max_{i,j}} - DT_{min_{i,j}}}{VIA_{max_{i,j}} - VIA_{min_{i,j}}} \tag{4.3}$$

onde os valores VIA_{max} e VIA_{min} correspondem aos valores máximos e mínimos da variável i destacada pelo VIA, constituindo o intervalo desta. Os valores DT_{max} e DT_{min} indicam então o intervalo para a mesma variável i no C4.5. Por exemplo, o IDT da variável V14 da regra 4 é: $IDT_{V14,1} = (0.9577 - 0.7983) / (0.9890 - 0.7320) = 0.6202$. Nota-se que V14 foi destacada pelo

VIA e não faz parte da regra original do C4.5. Desta forma, o intervalo de V14 no C4.5 é totalmente aberto. Observa-se também que uma variável presente na regra do C4.5, cujo intervalo não foi refinado pelo VIA, tem ambos valores *IDT* e *IVIA* igual a 1.0, pois seus intervalos são iguais. Assim, esta variável não proporciona ganho de informação. Aplicando-se as Equações 4.2 e 4.3 às variáveis das regras 4 e 5, tem-se:

Regra R4:

$IDT_{V14,4}$	=	0.6202	$IVIA_{V14,4}$	=	1.6124
$IDT_{V12,4}$	=	0.7527	$IVIA_{V12,4}$	=	1.3286
$IDT_{V10,4}$	=	1.0000	$IVIA_{V10,4}$	=	1.0000
$IDT_{ZI13,4}$	=	1.0000	$IVIA_{ZI13,4}$	=	1.0000
$IDT_{ZI12,4}$	=	1.0000	$IVIA_{ZI12,4}$	=	1.0000

Regra R5:

$IDT_{V14,5}$	=	0.9202	$IVIA_{V14,5}$	=	1.0867
$IDT_{V10,5}$	=	0.9429	$IVIA_{V12,5}$	=	1.0606
$IDT_{V12,5}$	=	1.0000	$IVIA_{V12,5}$	=	1.0000

2º - Cálculo do *índice de ganho individual* para o conjunto de regras do C4.5 e do VIA calculados através das seguintes equações:

$$i_{RDT_j} = \frac{\sum_{i=1}^n POND_i IDT_{i,j}}{\sum_{i=1}^n POND_i} \tag{4.4}$$

$$i_{RVIA_j} = \frac{\sum_{i=1}^n POND_i IVIA_{i,j}}{\sum_{i=1}^n POND_i} \tag{4.5}$$

Substituindo-se em (4.4) e (4.5) os respectivos valores de **POND** e os valores *IVIA* e *IDT* de cada variável, os índices de ganho individual de cada regra podem então obtidos. Os valores encontrados são:

Regra 4:

$$i_{RDT4} = [(10.0 \times 0.6202) + (5.0 \times 0.7527) + (1.0 \times 1.0000) + (7.5 \times 1.0000) + (5.0 \times 1.000)] / 28.5 = \mathbf{0.8234}$$

$$i_{RVIA4} = [(10.0 \times 1.6124) + (5.0 \times 1.3286) + (1.0 \times 1.0000) + (7.5 \times 1.0000) + (5.0 \times 1.000)] / 28.5 = \mathbf{1.2725}$$

Regra 5:

$$i_{RDT5} = \mathbf{0.811}$$

$$i_{RVIA5} = \mathbf{1.295}$$

3º - Cálculo do *índice de ganho de informação relativo* para cada regra. Uma vez obtido os índices i_{RDT} e i_{VIA} para cada regra, pode-se calcular o valor percentual do ganho de informação das novas regras extraídas pelo VIA em relação às regras originais obtidas pelo C4.5. Este valor é calculado pela seguinte expressão:

$$Gain_{VIA-DT} \% = \frac{|i_{RDT_j} - i_{VIA_j}|}{i_{RDT_j}} \times 100 \quad (4.6)$$

A Tabela 4.12 mostra o resultado do índice $Gain_{VIA-DT}$ para todas as regras do caso base do sistema teste IEEE 14 barras. Pode ser observado que algumas regras obtiveram um índice elevado de ganho. Conforme mostram as regras 4 e 5 extraídas pelo VIA, a variável com maior alteração nos intervalos foi a tensão na barra 14 (V14). Além desta variável apresentar maior peso (melhor posicionada no *ranking* das barras críticas), seu intervalo também foi refinado. Como não é uma variável inicialmente selecionada pelo C4.5, o ganho relativo para estas regras tendem a aumentar significativamente. Alguns índices, com valores relativos menores que menores que 5%, podem ser desconsiderados para efeito da análise, pois podem estar indicando apenas ajustes numéricos do algoritmo.

Tabela 4.12 - Índice de ganho de informação para cada regra para o caso base do sistema teste IEEE 14 barras.

ÍNDICE QUALITATIVO DAS REGRAS					
Caso	Classe	Regra	i_{RDT}	i_{RVIA}	$Gain_{VIA-DT}$
Base	Normal	1	0.5700	2.152	277.84%
	Alarme	2	0.6180	1.6610	168.84%
	Alarme	3	0.8120	1.4420	77.70%
	Alarme	4	0.8350	1.2560	50.33%
	inseguro	5	0.9460	1.0580	11.79%

Embora todas as regras iniciais do C4.5 tenham sido especializadas, o algoritmo VIA não foi suficientemente robusto para que as regras extraídas melhorem a taxa de classificação (Tabelas 4.9 e 4.11). Todavia não ocorre uma degradação da taxa de classificação. Uma maneira de proporcionar a melhora nestas taxas é testar novos pontos do espaço de hipóteses. Conforme mostrado no Capítulo 3, item 3.8.4, um procedimento nesta direção é proposto.

4.3.11 Refinamento das novas regras do VIA

Embora as hipóteses iniciais fornecidas pelo C4.5 proporcionem que o procedimento de gerar e testar regras seja minimizado, os resultados mostram que isso não é o bastante para melhorar as taxas de classificação das regras. Para que o modelo neural seja devidamente representado pelo conjunto de regras extraídas pelo VIA (critérios da *precisão* e *fidelidade*), é necessário testar novas hipóteses para que estas taxas se aproximem das taxas de classificação encontradas pelo OET2 (Tabela 4.4). Desta forma, a idéia é novamente processar o VIA, redefinindo alguns intervalos das novas regras geradas. O critério aqui adotado é descrito no item 3.8.4, Capítulo 3.

Do conjunto das regras novas mostradas na Tabela 4.10, observa-se que a variável V14 é que sofre maior alteração percentual no seu intervalo para as regras 1, 2, 4 e 5 e a variável V10 para a regra 3. Logo, as variáveis V14 e V10 terão seus respectivos intervalos redefinidos conforme descrito na metodologia. Na realidade, esta redefinição de intervalos fazem que as regras a serem testadas pelo VIA constituam um novo conjunto de hipóteses. A Tabela 4.13 mostra as novas regras encontradas executando este novo procedimento.

Assim como foi obtido o índice de ganho VIA/C4.5 (Tabela 4.12), pode-se avaliar o ganho destas regras novas refinadas em relação a este novo conjunto de hipóteses proposto. O resultado é mostrado na Tabela 4.14. Observa-se nesta tabela que os ganhos de informação diminuem. Isso é devido ao fato dos intervalos das variáveis mais destacadas anteriormente estarem melhor definidos. Este fato mostra que a região de solução factível em torno do ponto dado pelas hipóteses iniciais do C4.5 pode estar atingindo seu limite. Caso este fato seja verdadeiro, quaisquer novas alterações nos intervalos deste mesmo conjunto de variáveis, não devem produzir melhora na taxa de classificação.

O maior benefício da introdução do critério de testar novas regras da maneira descrita é a melhora na taxa de classificação das regras. Esta não é significativa, mas os resultados apontam que um procedimento de gerar e testar adequado pode empreender maior ganho, melhorando a *fidelidade* e a *precisão* das regras. Estes resultados podem ser vistos na Tabela 4.15. Nesta tabela, a designação VIA_i indica as regras extraídas pelo VIA no primeiro processamento e VIA_{ii} as novas regras obtidas.

Tabela 4.13 – Novas regras refinadas para o sistema teste IEEE 14 Barras – caso base.

REGRAS INICIAIS E FINAIS REFINADAS							
Caso BASE							
INTERVALOS ORIGINAIS				INTERVALOS NOVOS			
Min.	Var.	Max.		Min.	Var.	Max.	Alteração (%)
REGRA NUM. == 1 ----> CLASSE NORMAL							
0.9155	V10	1.0070		0.9193	V10	1.0070	4.14
0.9210	V14	0.9890		0.9210	V14	0.9890	0.00
0.9520	V12	1.0140		0.9520	V12	1.0140	0.00
REGRA NUM. == 2 ----> CLASSE ALERTA							
0.8188	V12	0.9520		0.8500	V12	0.9520	23.45
0.9330	V10	1.0067		0.9330	V10	1.0055	1.64
0.8790	V14	0.9570		0.8790	V14	0.9570	0.00
REGRA NUM. == 3 ----> CLASSE ALERTA							
0.8783	V14	0.9890		0.8783	V14	0.9890	0.00
0.9010	V12	1.0140		0.9010	V12	1.0140	0.00
0.9130	V10	0.9810		0.9130	V10	0.9810	0.00
3.5850	ZI13	6.8430		3.5850	ZI13	6.8430	0.00
4.8640	ZI12	10.0770		4.8640	ZI12	10.0770	0.00
9.8270	ZI11	14.8050		9.8270	ZI11	14.8050	0.00
REGRA NUM. == 4 ----> CLASSE ALERTA							
0.7956	V12	0.9883		0.8750	V12	0.9883	41.19
0.8790	V14	0.9570		0.8790	V14	0.9570	0.00
0.9120	V10	0.9330		0.9120	V10	0.9330	0.00
2.2130	ZI13	3.5850		2.2130	ZI13	3.5850	0.00
4.8640	ZI12	10.0770		4.8640	ZI12	10.0770	0.00
REGRA NUM. == 5 ----> CLASSE INSEGURO							
0.7560	V10	0.9520		0.7804	V10	0.9520	12.45
0.7740	V12	0.9937		0.7740	V12	0.9797	6.36
0.7320	V14	0.9270		0.7436	V14	0.9270	5.96

Tabela 4.14 - Índice de ganho de informação das regras refinadas - Ganho (VIAi/VIAii), para o caso base do sistema teste IEEE 14 barras.

ÍNDICE QUALITATIVO DAS REGRAS					
Caso	Classe	Regra	I_{VIAi}	i_{VIAii}	Ganho (VIAi/VIAii)
Base	normal	1	0.9970	1.003	0.53%
	alarme	2	0.9260	1.0970	18.48%
	alarme	3	1.0000	1.0000	0.00%
	alarme	4	0.9340	1.1130	19.22%
	inseguro	5	0.9350	1.0700	14.40%

Tabela 4.15 – Taxa de classificação por regra do VIA – *refinamento* - para o caso base do sistema teste IEEE 14 barras.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	126	15	0	94.45% (126/132)	4.66% (15/265)
02	10	118	14	90.08% (118/131)	14.66% (39/266)
03	5	9	3		
04	2	3	5		
05	13	45	134	100.00% (134/134)	22.05% (58/263)
Total Geral	Class. Correta			Class. Errada	
	94.21% (378/397)			28.21% (112/397)	

Na Tabela 4.15, os resultados em negrito são as respectivas taxas de classificação percentuais que foram alteradas. Comparando com a Tabela 4.11, nota-se que a taxa de classificação correta total melhora pouco. Entretanto, a taxa de classificação errada total é bastante reduzida. Para a classe normal, como o ganho de informação foi praticamente nulo, as taxas permaneceram inalteradas.

O maior ganho ocorreu na regra 5, classe Inseguro. A nova definição dos intervalos foi extremamente benéfica à melhoria da classificação. Isso ocorre por dois motivos. No primeiro processamento do VIA (hipóteses C4.5), embora este tenha inserido novas variáveis bem relacionadas ao problema, o algoritmo não foi suficientemente robusto para refinar os intervalos que propiciasse melhorar as taxas de classificação. O segundo motivo foi a redefinição do intervalo da variável V14 que é fortemente relacionada com este nível operativo. Esta redefinição possibilitou ao VIA utilizar melhor a informação proveniente do OET2, ajustando o intervalo de todas as variáveis presentes na regra.

4.4 RESULTADOS DAS SIMULAÇÕES

Os resultados apresentados a seguir dizem respeito a casos simulados para os sistemas testes IEEE 14 barras e 118 barras. Serão mostrados os resultados de contingências relacionadas com a perda de equipamentos de geração de potência reativa, perdas de linhas de transmissão e perda de gerador. Os aspectos específicos da formação da base de dados, das simulações do OET2, C4.5 e do VIA são os informados no item 4.3. Os dados referentes aos sistemas testes

estão apresentados no Apêndice G.

4.4.1 Resultados das simulações para o sistema teste IEEE 14 barras

O sistema IEEE 14 barras é um sistema de pequeno porte e as simulações feitas para gerar a base de dados são computacionalmente rápidas. Vale lembrar que o número de exemplares presentes na base de dados pode ser controlado pelo passo k do MC. A Tabela 4.13 mostra o resultado do erro médio quadrático para a estimação da margem de potência ativa pelo OET2. Esta taxa é avaliada para um conjunto de teste formado por 40% dos exemplares do conjunto de dados obtido. É mostrada também nesta tabela a relação das barras críticas selecionadas de acordo com o critério adotado no item 4.3.2 – Tabelas 4.2 e 4.3. Para os casos simulados, foi utilizada uma perturbação aleatória de 15% (fatores $a21$ e $a22$, Equação 4.1).

Tabela 4.16 – Estimação da margem de potência ativa pela rede neural OET2 para algumas contingências do sistema teste IEEE 14 barras considerando 15 % de perturbação aleatória

MARGEM DE POTÊNCIA ATIVA – OET2 –			
CONTINGÊNCIA	Nº EXEMPLARES CONJUNTO DE TESTE	5 BARRAS MAIS CRÍTICAS + → -	ERRO MÉDIO QUADRÁTICO (pu)
BASE	1257	14 - 13 - 12 - 11 - 10	0.0076
CS barra 06	1222	14 - 13 - 12 - 11 - 10	0.0063
Capacitor barra 09	1871	14 - 13 - 12 - 10 - 11	0.0053
LT barras 02 - 04	1196	13 - 14 - 10 - 11 - 12	0.0048
LT barras 04 - 05	1325	14 - 10 - 13 - 11 - 12	0.0046
LT barras 06 - 11	1894	13 - 14 - 11 - 10 - 12	0.0077
LT barras 06 - 12	1966	14 - 13 - 12 - 10 - 11	0.0085
LT barras 06 - 13	1699	13 - 12 - 14 - 10 - 11	0.0095
LT barras 09 - 10	1240	11 - 10 - 13 - 12 - 14	0.0073
Gerador barra 2	1177	14 - 13 - 11 - 11 - 10	0.0079

Para verificar a potencialidade do algoritmo de treinamento OET2, uma comparação com a RNA-ML pode ser feita. Considerando o problema da estimação da margem de potência ativa para o caso base do 14 Barras, verifica-se que o erro médio quadrático obtido pela RNA-ML para o conjunto de teste foi de 0.00726 p.u. Este erro é praticamente igual ao OET2. Isto mostra que métodos baseados mínimos quadrados não lineares podem ser tão precisos quanto aos baseados em descida em gradientes, como é o algoritmo de treinamento da RNA-ML. Entretanto, o tempo computacional exigido para o treinamento da RNA-ML é muito grande. A rede neural OET2 executa o ciclo de treinamento com um tempo em torno de 250 milissegundos

(código fonte Fortran). A mesma tarefa é executada pela RNA-ML (Toolbox Matlab) em torno de 960 segundos, com uma simulação com 5000 épocas. Estes resultados mostram que o OET2 lida com grandes bases de dados com precisão e eficiência computacional, reforçando as justificativas descritas no Capítulo 3 para a utilização deste neste trabalho. Logo, justifica-se também a utilização do OET2 em aplicações *on-line* para avaliação de segurança de tensão.

Nestas simulações, os valores dos pesos para compor o cálculo do índice de ganho de informação seguem o critério abaixo relacionado. Vale lembrar que o *ranking* das barras críticas encontrado para cada caso é utilizado para atribuir estes pesos. O critério geral adotado tem o seguinte formato:

```
Variáveis V e ZI da barra crítica 1 (+) → peso 10.0
Variáveis V e ZI da barra crítica 2      → peso 7.5
Variáveis V e ZI da barra crítica 3      → peso 4.0
Variáveis V e ZI da barra crítica 4      → peso 2.5
Variáveis V e ZI da barra crítica 5 (-) → peso 1.0
```

Os itens a seguir mostram os resultados obtidos para as simulações das contingências propostas. São apresentados os resultados estatísticos para a taxa de classificação do OET2, as taxas de classificação das regras do C4.5, o índice qualitativo de ganho de informação, as regras finais refinadas e respectivas estatísticas. Vale lembrar que todos os resultados apresentados a seguir são realizados utilizando um conjunto de teste contendo 397 exemplares. As regras obtidas de cada simulação, tanto para o C4.5 quanto para o VIA, são apresentadas nos Apêndices E e F, respectivamente.

4.4.1.1 Caso 1: Contingência perda do compensador síncrono – Barra 6

Dados:

- a. Variáveis excluídas por dependência linear: V13 e V11.
- b. Valor *Bound1* - OET2: 18.5
- c. Número de regras C4.5: Normal: 1; Alerta: 4; Inseguro: 1

Tabela 4.17a Taxa de classificação OET2 – Caso 1.

Normal	Alerta	Inseguro	Total
<u>115</u>	17	0	132
12	<u>102</u>	17	131
0	10	<u>124</u>	134
Total Geral	Class. Correta		Class. Errada
	84.89% (341/397)		14.11% (56/397)

Tabela 4.17b Taxa de classificação C4.5 – Caso 1.

Normal	Alerta	Inseguro	Total
<u>124</u>	8	0	132
22	<u>96</u>	13	131
0	19	<u>115</u>	134
Total Geral	Class. Correta		Class. Errada
	84.38% (335/397)		14.60% (62/397)

Tabela 4.17c - Taxa de classificação por regra C4.5 – Caso 1.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>132</u>	97	0	100.00% (132/132)	36.60% (97/265)
02	3	<u>41</u>	0	81.68% (107/131)	14.29% (38/266)
03		<u>7</u>	2		
04	11	<u>88</u>	21		
05	2	<u>12</u>	3		
06	9	42	<u>134</u>	100.00% (134/134)	19.39% (51/263)
Total Geral	Class. Correta			Class. Errada	
	93.70% (372/397)			46.85% (186/397)	

Tabela 4.17d - Índice de ganho de informação - C4.5/VIA e VIAi/VIAii – Caso 1.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
CS barra 6	normal	1	264.13%	10.42%
	alarme	2	923.56%	0.45%
	alarme	3	386.53%	3.17%
	alarme	4	129.19%	0.65%
	Alarme	5	248.07%	3.80%
	inseguro	6	0.00%	0.00%

Tabela 4.17e - Taxa de classificação *por regra* VIAii – Caso 1.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>132</u>	53	0	100.00% (132/132)	20.00% (53/265)
02	32	<u>81</u>	25	97.71% (128/131)	32.33% (86/266)
03	0	<u>7</u>	2		
04	52	<u>128</u>	34		
05	2	<u>12</u>	3		
06	9	42	<u>134</u>	100.00% (134/134)	19.39% (51/263)
Total Geral	Class. Correta			Class. Errada	
	<u>99.24% (394/397)</u>			<u>47.86% (190/397)</u>	

Os resultados mostram que o OET2 é mais preciso em classificar que as regras extraídas pelo C4.5 (85.89% / 84.38 %). As variáveis destacadas pelo VIA foram exclusivamente as tensões (Apêndice F – caso 1, 14 barras). Pode ser observado também que ocorrem ganhos significativos no primeiro processamento do VIA em relação ao C4.5. Comparando-se os resultados encontrados para as taxas de classificação das regras do C4.5 e do VIAii, nota-se que as regras 2 e 4 tiveram suas taxas deterioradas.

4.4.1.2 Caso 2 - Contingência perda de capacitor - barra 9

Dados:

- Variáveis excluídas por dependência linear: V13 e V11.
- Valor *Bound1* - OET2: 23.84
- Número de regras C4.5: Normal: 1; Alerta: 1; Inseguro: 2

Tabela 4.18a Taxa de classificação OET2 – Caso 2.

Normal	Alerta	Inseguro	Total
<u>117</u>	15	0	132
5	<u>116</u>	10	131
0	18	<u>116</u>	134
Total Geral	Class. Correta		Class. Errada
	<u>87.90% (349/397)</u>		<u>12.09% (48/397)</u>

Tabela 4.18b Taxa de classificação C4.5 – Caso 2

Normal	Alerta	Inseguro	Total
<u>117</u>	15	0	132
5	<u>118</u>	8	131
0	26	<u>108</u>	134
Total Geral	Class. Correta		Class. Errada
	<u>86.40% (343/397)</u>		<u>13.60% (54/397)</u>

Tabela 4.18c - Taxa de classificação por regra C4.5 – Caso 2.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>118</u>	6	0	89.39% (118/132)	6.60% (6/265)
02	25	<u>124</u>	32	94.66% (124/131)	21.43% (57/266)
03	0	02	<u>93</u>	80.60% (108/134)	3.04% (8/263)
04	0	06	<u>27</u>		
Total Geral	Class. Correta			Class. Errada	
	<u>88.16% (350/397)</u>			<u>17.88% (71/397)</u>	

Tabela 4.18d Índice de ganho de informação - C4.5/VIA e VIAi/VIAii – Caso 2.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
Capacitor barra 9	normal	1	137.70%	19.64%
	alarme	2	83.35%	42.91%
	inseguro	3	3.45%	0.00%
	inseguro	4	10.57%	2.20%

Tabela 4.18e - Taxa de classificação por regra VIAii – Caso 2.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>118</u>	6	0	89.39% (118/132)	6.60% (6/265)
02	2	<u>124</u>	6	94.66% (124/131)	2.26% (8/266)
03	0	02	<u>93</u>	80.60% (108/134)	3.04% (8/263)
04	0	06	<u>27</u>		
Total Geral	Class. Correta			Class. Errada	
	<u>88.16% (350/397)</u>			<u>8.37% (22/397)</u>	

Os resultados mostram que o OET2 é mais preciso em classificar que as regras extraídas pelo C4.5 (87.90% / 86.40 %). As variáveis destacadas pelo VIA foram exclusivamente as

tensões (Apêndice F – caso 2, 14 barras). Comparando-se os resultados encontrados para as taxas de classificação das regras do C4.5 e do VIAii, nota-se que a regra 2, classe Alerta, apresentou uma redução significativa na taxa de classificação errada.

4.4.1.3 Caso 3 - Contingência perda de linha de transmissão – Barras 02 - 04

Dados:

- Variáveis excluídas por dependência linear: V13 e V11.
- Valor *Bound1* - OET2: 14.56
- Número de regras C4.5: Normal: 2; Alerta: 4; Inseguro: 1

Tabela 4.19a Taxa de classificação OET2 – Caso 3.

Normal	Alerta	Inseguro	Total
<u>108</u>	24	0	132
9	<u>99</u>	23	131
0	22	<u>112</u>	134
Total Geral	Class. Correta		Class. Errada
	80.35% (319/397)		19.65% (78/397)

Tabela 4.19b Taxa de classificação C4.5 – Caso 3.

Normal	Alerta	Inseguro	Total
<u>116</u>	16	0	132
16	<u>93</u>	22	131
0	32	<u>102</u>	134
Total Geral	Class. Correta		Class. Errada
	78.34% (311/397)		21.17% (86/397)

Tabela 4.19c - Taxa de classificação por regra C4.5 – Caso 3.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>116</u>	12	0	91.67%	7.55%
02	<u>5</u>	8	0	(121/132)	(20/265)
03	5	<u>26</u>	14		
04	2	<u>12</u>	7	78.63%	21.80%
05	12	<u>59</u>	8	(103/131)	(58/266)
06	0	<u>16</u>	19		
07	5	65	<u>131</u>	97.76%	26.62%
			(131/134)	(70/263)	
Total Geral	Class. Correta			Class. Errada	
	89.42% (372/397)			37.28% (148/397)	

Tabela 4.19d - Índice de ganho de informação - C4.5/VIA - VIAi/VIAii – Caso 3.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
LT barras 02 -04	normal	1	31.19%	54.11%
	normal		70.37%	0.00%
	alarme	2	273.56%	0.09%
	alarme	3	36.58%	24.04%
	alarme	4	166.78%	83.84%
	alarme	5	38.45%	24.10%
	inseguro	5	17.07%	0.00%

Tabela 4.19e - Taxa de classificação por regra VIAii – Caso 3.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>116</u>	12	0	91.67%	7.17%
02	<u>5</u>	7	0	(121/132)	(19/265)
03	5	<u>26</u>	14	78.63%	17.29%
04	2	<u>12</u>	7		
05	2	<u>59</u>	6		
06		<u>16</u>	19		
07	5	65	<u>131</u>	97.76%	26.62%
				(131/134)	(70/263)
Total Geral	Class. Correta			Class. Errada	
	<u>89.42% (372/397)</u>			<u>34.07% (135/397)</u>	

Os resultados mostram que o OET2 é mais preciso em classificar que as regras extraídas pelo C4.5 (80.35% / 78.34 %). As variáveis destacadas pelo VIA foram exclusivamente as tensões (Apêndice F – caso 3, 14 barras). Comparando-se os resultados encontrados para as taxas de classificação das regras do C4.5 e do VIAii, nota-se que houve uma redução na taxa de classificação errada para as classes Normal e Alerta.

4.4.1.4 Caso 4 - Contingência perda de linha de transmissão – Barras 04 - 05

Dados:

- Variáveis excluídas por dependência linear: V13 e V11.
- Valor *Bound1* - OET2: 4.22
- Número de regras C4.5: Normal: 1; Alerta: 3; Inseguro: 1

Tabela 4.20a Taxa de classificação OET2 – Caso 4.

Normal	Alerta	Inseguro	Total
<u>106</u>	26	0	132
5	<u>114</u>	12	131
0	16	<u>118</u>	134
Total Geral	Class. Correta		Class. Errada
	84.14% (338/397)		14.86% (59/397)

Tabela 4.20b Taxa de classificação C4.5 – Caso 4.

Normal	Alerta	Inseguro	Total
<u>111</u>	21	0	132
10	<u>105</u>	16	131
0	17	<u>117</u>	134
Total Geral	Class. Correta		Class. Errada
	83.87% (333/397)		16.10% (64/397)

Tabela 4.20c - Taxa de classificação por regra C4.5 – Caso 4.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>126</u>	18	0	94.45% (126/132)	6.79% (18/265)
02	10	<u>88</u>	12	87.79% (115/131)	17.29% (46/266)
03	5	<u>11</u>	0		
04	5	<u>32</u>	19		
05		16	<u>117</u>	87.31% (117/134)	6.08% (80/263)
Total Geral	Class. Correta			Class. Errada	
	90.18% (358/397)			20.15% (80/397)	

Tabela 4.20d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii –Caso 4.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
LT barras 04 -05	normal	1	91.72%	0.00%
	alarme	2	639.09%	4.42%
	alarme	3	568.89%	6.78%
	alarme	4	971.31%	62.75%
	inseguro	5	43.69%	0.00%

Tabela 4.20e - Taxa de classificação *por regra* VIAii – Caso 4.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>126</u>	18	0	94.45% (126/132)	6.79% (18/265)
02	10	<u>115</u>	12	87.79% (115/131)	15.41% (41/266)
03	5	<u>10</u>	0		
04	0	<u>32</u>	18		
05		16	<u>117</u>	87.31% (117/134)	6.08% (16/263)
Total Geral	Class. Correta			Class. Errada	
	<u>90.18% (358/397)</u>			<u>18.89% (75/397)</u>	

Os resultados mostram que o OET2 é mais preciso em classificar que as regras extraídas pelo C4.5 (85.35% / 83.87 %). As variáveis destacadas pelo VIA foram exclusivamente as tensões (Apêndice F – caso 4, 14 barras). Comparando-se os resultados encontrados para as taxas de classificação das regras do C4.5 e do VIAii, nota-se que a taxa de classificação errada para a classe Alerta foi reduzida. Observa-se que esta redução ocorre na regra 4, sendo a que apresentou maior ganho de informação tanto em relação ao C4.5 quanto ao VIAi.

4.4.1.5 Caso 5 -Contingência perda de linha de transmissão – Barras 06 - 11

Dados:

- Variáveis excluídas por dependência linear: V13 e V10.
- Valor *Bound1* - OET2: 15.43
- Número de regras C4.5: Normal: 1; Alerta: 2; Inseguro: 1

Tabela 4.21a Taxa de classificação OET2 – Caso 5.

Normal	Alerta	Inseguro	Total
<u>123</u>	9	0	132
9	<u>100</u>	22	131
0	10	<u>122</u>	134
Total Geral	Class. Correta		Class. Errada
	<u>86.69% (345/397)</u>		<u>12.40% (50/397)</u>

Tabela 4.21b Taxa de classificação C4.5 – Caso 5.

Normal	Alerta	Inseguro	Total
<u>122</u>	10	0	132
25	<u>88</u>	18	131
0	12	<u>122</u>	134
Total Geral	Class. Correta		Class. Errada
	<u>83.63% (332/397)</u>		<u>16.40% (65/397)</u>

Tabela 4.21c - Taxa de classificação *por regra* do C4.5 – Caso 5.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>123</u>	29	0	93.18% (123/132)	10.94% (29/265)
02	12	<u>44</u>	0	67.94% (89/131)	9.77% (26/266)
03	9	<u>50</u>	7		
04		18	<u>122</u>	91.06% (122/134)	6.84% (18/263)
Total Geral	Class. Correta			Class. Errada	
	<u>84.13% (334/397)</u>			<u>18.39% (73/397)</u>	

Tabela 4.21d - Índice de ganho de informação – C4.5/VIA – VIAi/VIAii – Caso 5.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
LT barras 06 -11	normal	1	147.38%	0.00%
	alarme	2	1184.45%	0.43%
	alarme	3	721.69%	54.38%
	inseguro	4	44.26%	0.78%

Tabela 4.21e - Taxa de classificação *por regra* VIAii –Caso 5.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>123</u>	29	0	93.18% (123/132)	10.94% (29/265)
02	12	<u>44</u>	0	78.62% (103/131)	9.02% (24/266)
03	8	<u>71</u>	6		
04		18	<u>122</u>	91.06% (122/134)	6.84% (18/263)
Total Geral	Class. Correta			Class. Errada	
	<u>87.66% (348/397)</u>			<u>17.88% (71/397)</u>	

Os resultados mostram que o OET2 é mais preciso em classificar que as regras extraídas pelo C4.5 (86.69% / 83.63 %). As variáveis destacadas pelo VIA foram exclusivamente as tensões (Apêndice F – caso 5, 14 barras). Comparando-se os resultados encontrados para as taxas de classificação das regras do C4.5 e do VIAii, nota-se ocorreu redução na taxa de classificação errada para a classe Alerta. Observa-se que esta redução ocorre na regra 3, sendo a que apresentou maior ganho de informação tanto em relação ao C4.5 quanto ao VIAi.

4.4.1.6 Caso 6 - Contingência perda de linha de transmissão – Barras 06 - 12

Dados:

- Variáveis excluídas por dependência linear: V12 e V11.
- Valor *Bound1* - OET2: 12.87
- Número de regras C4.5: Normal: 1; Alerta: 2; Inseguro: 1

Tabela 4.22a Taxa de classificação OET2 – Caso 6.

Normal	Alerta	Inseguro	Total
<u>117</u>	15	0	132
8	<u>107</u>	5	131
0	32	<u>102</u>	134
Total Geral	Class. Correta		Class. Errada
	84.49% (337/397)		14.11% (60/397)

Tabela 4.22b Taxa de classificação C4.5 – Caso 6.

Normal	Alerta	Inseguro	Total
<u>108</u>	24	0	132
5	<u>125</u>	1	131
0	40	<u>94</u>	134
Total Geral	Class. Correta		Class. Errada
	82.37% (327/397)		17.60% (70/397)

Tabela 4.22c - Taxa de classificação *por regra* C4.5 – Caso 6.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>125</u>	32	0	94.70% (125/132)	12.08% (32/265)
02	9	<u>9</u>	0	56.49% (74/131)	7.89% (21/266)
03	4	<u>65</u>	8		
04	0	42	<u>130</u>	97.01% (130/134)	14.97% (42/263)
Total Geral	Class. Correta			Class. Errada	
	<u>82.87% (329/397)</u>			<u>23.93% (95/397)</u>	

Tabela 4.22d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso 6.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
LT barras 06 -12	normal	1	60.36%	2.49%
	alarme	2	984.09%	1.27%
	alarme	3	766.30%	63.18%
	inseguro	4	64.00%	0.00%

Tabela 4.22e - Taxa de classificação *por regra* pelo VIAii – Caso 6.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>125</u>	32	0	94.70% (125/132)	12.08% (32/265)
02	9	<u>24</u>	0	71.75% (94/131)	8.65% (23/266)
03	6	<u>78</u>	10		
04	0	42	<u>130</u>	97.01% (130/134)	14.97% (42/263)
Total Geral	Class. Correta			Class. Errada	
	<u>87.91% (349/397)</u>			<u>24.43% (97/397)</u>	

Os resultados mostram que o OET2 é mais preciso em classificar que as regras extraídas pelo C4.5 (84.49% / 82.37%). As variáveis destacadas pelo VIA foram exclusivamente as tensões (Apêndice F – caso 6, 14 barras). Comparando-se os resultados encontrados para as taxas de classificação das regras do C4.5 e do VIAii, nota-se um sensível aumento da classificação correta para a classe Alerta. Observa-se que isto ocorre na regra 3, sendo a regra que apresentou maior ganho de informação tanto em relação ao C4.5 quanto ao VIAi. Entretanto, a regra 3 teve sua taxa de classificação errada um pouco aumentada.

4.4.1.7 Caso 7- Contingência perda de linha de transmissão – Barras 06 - 13

Dados:

- Variáveis excluídas por dependência linear: V12 e V14.
- Valor *Bound1* - OET2: 23.78
- Número de regras C4.5: Normal: 1; Alerta: 2; Inseguro: 1

Tabela 4.23a Taxa de classificação OET2 – Caso 7.

Normal	Alerta	Inseguro	Total
<u>108</u>	24	0	132
12	<u>106</u>	13	131
0	22	<u>112</u>	134
Total Geral	Class. Correta		Class. Errada
	82.12% (326/397)		17.88% (71/397)

Tabela 4.23b Taxa de classificação C4.5 – Caso 7.

Normal	Alerta	Inseguro	Total
<u>108</u>	24	0	132
5	<u>120</u>	6	131
0	40	<u>94</u>	134
Total Geral	Class. Correta		Class. Errada
	81.11% (322/397)		18.89% (75/397)

Tabela 4.23e - Taxa de classificação por regra C4.5 – Caso 7.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>108</u>	6	0	81.82% (108/132)	2.26% (6/265)
02	12	<u>90</u>	35	97.71% (128/131)	25.19% (67/266)
03	6	<u>39</u>	14		
04		1	<u>94</u>	70.15% (94/134)	0.38% (1/263)
Total Geral	Class. Correta			Class. Errada	
	83.12% (330/397)			29.47% (117/397)	

Tabela 4.23d - Índice de ganho de informação C4.5/VIA – VIAi/VIAii – Caso 7.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
LT barras 06 -13	normal	1	187.05%	0.00%
	alarme	2	551.02%	28.49%
	alarme	4	870.77%	10.27%
	inseguro	5	62.06%	1.26%

Tabela 4.23e - Taxa de classificação por regra VIAii – Caso 7.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>108</u>	6	0	81.82% (108/132)	2.26% (6/265)
02	12	<u>90</u>	35	97.71% (128/131)	24.44% (65/266)
03	6	<u>42</u>	12		
04		1	<u>94</u>	70.15% (94/134)	0.38% (1/263)
Total Geral	Class. Correta			Class. Errada	
	<u>83.12% (330/397)</u>			<u>18.14% (72/397)</u>	

Os resultados mostram que o OET2 é mais preciso em classificar que as regras extraídas pelo C4.5 (82.12% / 81.11%). As variáveis destacadas pelo VIA foram exclusivamente as tensões (Apêndice F – caso 7, 14 barras). Comparando-se os resultados encontrados para as taxas de classificação das regras do C4.5 e do VIAii, verifica-se que apenas a taxa de classificação errada foi diminuída para a regra 3, classe Alerta.

4.4.1.8 Caso 8 - Contingência perda de linha de transmissão – Barras 09 - 10

Dados:

- Variáveis excluídas por dependência linear: V13 e V11.
- Valor *Bound1* - OET2: 21.70
- Número de regras C4.5: Normal: 1; Alerta: 2; Inseguro: 2

Tabela 4.24a Taxa de classificação OET2 – Caso 8.

Normal	Alerta	Inseguro	Total
<u>116</u>	16	0	132
10	<u>101</u>	20	131
0	20	<u>113</u>	134
Total Geral	Class. Correta		Class. Errada
	<u>83.12% (330/397)</u>		<u>16.88% (67/397)</u>

Tabela 4.24b Taxa de classificação C4.5 – Caso 8.

Normal	Alerta	Inseguro	Total
<u>129</u>	3	0	132
26	<u>76</u>	29	131
0	30	<u>104</u>	134
Total Geral	Class. Correta		Class. Errada
	<u>77.83% (309/397)</u>		<u>22.20% (88/397)</u>

Tabela 4.24c - Taxa de classificação por regra C4.5 – Caso 8.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>130</u>	28	0	98.48% (130/132)	10.57% (28/265)
02	3	<u>58</u>	5	57.25% (75/131)	12.78% (34/266)
03	12	<u>26</u>	14		
04		2	<u>68</u>	77.61% (104/134)	11.03% (29/263)
05		27	<u>50</u>		
Total Geral	Class. Correta			Class. Errada	
	<u>77.83% (309/397)</u>			<u>23.93% (95/397)</u>	

Tabela 4.24d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso 8.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
LT barras 09 -10	normal	1	16.57%	0.00%
	alarme	2	53.85%	8.54%
	alarme	3	44.81%	0.85%
	Inseguro	4	32.29%	0.00%
	inseguro	5	21.53%	4.15%

Tabela 4.24e - Taxa de classificação por regra VIAi – Caso 8.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>130</u>	28	0	98.48% (130/132)	10.57% (28/265)
02	3	<u>83</u>	5	77.86%	12.78%
03	11	<u>25</u>	15	(102/131)	(34/266)
04		2	<u>68</u>	77.61% (104/134)	11.03% (29/263)
05		27	<u>50</u>		
Total Geral	Class. Correta			Class. Errada	
	<u>84.63% (336/397)</u>			<u>23.93% (95/397)</u>	

Os resultados mostram que o OET2 é mais preciso em classificar que as regras extraídas pelo C4.5 (83.12% / 77.83%). As variáveis destacadas pelo VIA foram exclusivamente as tensões (Apêndice F – caso 8, 14 barras). Comparando-se os resultados encontrados para as taxas de classificação das regras do C4.5 e do VIAii, apenas a taxa de classificação correta aumenta para a regra 2, classe Alerta.

4.4.1.9 Caso 9 -Contingência perda de gerador – Barra 2

Dados:

- Variáveis excluídas por dependência linear: V13 e V10.
- Valor *Bound1* - OET2: 15.43
- Número de regras C4.5: Normal: 1; Alerta: 4; Inseguro: 1

Tabela 4.25a Taxa de classificação OET2 – Caso 9.

Normal	Alerta	Inseguro	Total
<u>116</u>	16	0	132
10	<u>113</u>	8	131
0	15	<u>119</u>	134
Total Geral	Class. Correta		Class. Errada
	87.66% (348/397)		12.34% (49/397)

Tabela 4.25b Taxa de classificação C4.5 – Caso 9.

Normal	Alerta	Inseguro	Total
<u>122</u>	10	0	132
17	<u>91</u>	23	131
0	18	<u>116</u>	134
Total Geral	Class. Correta		Class. Errada
	<u>82.87% (329/397)</u>		<u>17.13% (68/397)</u>

Tabela 4.25c – Taxa de classificação *por regra* C4.5 – Caso 9.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>125</u>	19	0	94.70% (125/132)	7.17% (19/265)
02	5	<u>18</u>	0	71.76% (94/131)	10.15% (27/266)
03	3	<u>77</u>	7		
04	3	<u>42</u>	8		
05	4	<u>2</u>	6		
06		41	<u>134</u>		
Total Geral	Class. Correta			Class. Errada	
	<u>88.91% (353/397)</u>			<u>18.39% (73/397)</u>	

Tabela 4.25d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso 9.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
LT barras 09 -10	normal	1	59.86%	10.41%
	alarme	2	46.69%	25.54%
	alarme	3	204.08%	1.75%
	Alarme	4	81.07%	13.68%
	Alarme	5	272.83%	68.39%
	inseguro	6	23.37%	10.98%

Tabela 4.25e – Taxa de classificação *por regra* VIAi – Caso 9.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>125</u>	19	0	94.70% (125/132)	7.17% (19/265)
02	5	<u>18</u>	0	71.76% (94/131)	10.15% (27/266)
03	3	<u>77</u>	7		
04	3	<u>42</u>	8		
05	4	<u>2</u>	6		
06		38	<u>134</u>		
Total Geral	Class. Correta			Class. Errada	
	<u>88.91% (353/397)</u>			<u>18.39% (73/397)</u>	

Os resultados mostram que OET2 é mais preciso em classificar que as regras extraídas pelo C4.5 (87.66% / 82.87%). As variáveis destacadas pelo VIA foram exclusivamente as tensões (Apêndice F – caso 9, 14 barras). Comparando-se os resultados encontrados para as taxas de classificação das regras do C4.5 e do VIAii, ocorreu uma pequena redução na taxa de classificação errada para a regra 6, classe Inseguro.

4.4.2 Resultados das simulações para o sistema teste IEEE 118 barras

O sistema IEEE 118 barras é um sistema de médio porte. Os dados utilizados nas simulações são apresentados no Apêndice G. A Tabela 4.26 mostra o resultado do erro médio quadrático para a estimação da margem de potência ativa pelo OET2 para o caso base e 2 contingências. Esta taxa é avaliada para um conjunto de teste formado por 40% dos exemplares do conjunto de dados obtido. É mostrada também nesta tabela, a relação das barras críticas selecionadas de acordo com o critério adotado no item 4.3.2 – Tabelas 4.2 e 4.3. Nestas simulações foram utilizadas as informações das 6 barras mais críticas. A perturbação aleatória introduzida foi de 15%.

Tabela 4.26 – Estimação da margem de potência ativa pela rede neural OET2 para algumas contingências do sistema teste IEEE 118 barras considerando 15 % de perturbação aleatória.

MARGEM DE POTÊNCIA ATIVA – OET2 –			
CONTINGÊNCIA	Nº EXEMPLARES CONJUNTO DE TESTE	6 BARRAS MAIS CRÍTICAS + → -	ERRO MÉDIO QUADRÁTICO (pu)
BASE	1568	41 – 53 – 39 – 52 – 109 – 58	0.2880
CS barra 76	1528	41 – 53 – 39 – 52 – 109 – 58	0.2460
LT barras 76 – 77	1234	9 – 5 – 30 – 7 – 11 – 23	0.3460

4.4.2.1 Caso base

Dados:

- a. Variáveis excluídas por dependência linear: V41, V53 e V58.
- b. Valor *Bound1* – OET2: 14.29
- c. Número de regras C4.5: Normal: 1; Alerta: 4; Inseguro: 2

Tabela 4.27a Taxa de classificação OET2 – Caso base.

Normal	Alerta	Inseguro	Total
<u>126</u>	6	0	132
6	<u>106</u>	20	132
0	91	<u>124</u>	133
Total Geral	Class. Correta		Class. Errada
	89.67% (356/397)		10.32% (41/397)

Tabela 4.27b Taxa de classificação C4.5 – Caso base.

Normal	Alerta	Inseguro	Total
<u>126</u>	6	0	132
12	<u>94</u>	26	132
0	21	<u>112</u>	133
Total Geral	Class. Correta		Class. Errada
	<u>83.63% (332/397)</u>		<u>16.40% (65/397)</u>

Tabela 4.27c – Taxa de classificação *por regra* C4.5 – Caso base.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>132</u>	56	0	100.00% (132/132)	21.13% (56/265)
02	2	<u>17</u>	0	88.64% (115/132)	26.79% (71/265)
03	5	<u>26</u>	7		
04	0	<u>29</u>	3		
05	23	<u>89</u>	41		
06		6	<u>32</u>	100.00% (133/133)	37.12% (98/263)
07	21	77	<u>133</u>		
Total Geral	Class. Correta			Class. Errada	
	<u>88.91% (353/397)</u>			<u>56.68% (225/397)</u>	

Tabela 4.27d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso base.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
Caso Base	Normal	1	0.00%	0.00%
	Alarme	2	0.00%	8.81%
	Alarme	3	12.00%	0.00%
	Alarme	4	0.00%	7.39%
	Alarme	5	9.40%	0.00%
	Inseguro	6	1.80%	1.87%
	Inseguro	7	0.00%	0.00%

Tabela 4.27e – Taxa de classificação *por regra* VIAi – Caso base.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>132</u>	56	0	100.00% (132/132)	21.13% (56/265)
02	2	<u>17</u>	0	89.39% (118/132)	26.79% (71/265)
03	5	<u>26</u>	7		
04	0	<u>32</u>	3		
05	23	<u>89</u>	41		
06		4	<u>32</u>	100.00% (133/133)	37.12% (76/263)
07	0	76	<u>133</u>		
Total Geral	Class. Correta			Class. Errada	
	96.65% (383/397)			51.13% (203/397)	

Os resultados mostram que o OET2 é mais preciso em classificar que as regras extraídas pelo C4.5 (87.66% / 82.87%). As variáveis destacadas pelo VIA foram exclusivamente as tensões (Apêndice F – caso base, 118 barras). Comparando-se os resultados encontrados para as taxas de classificação das regras do C4.5 e do VIAii, a taxa de classificação correta da regra 3 (classe Alerta) aumentou, e para a regra 6 (classe Inseguro) houve uma pequena redução na taxa de classificação errada. São notados também que os ganhos de informação ocorridos nessa simulação são pequenos.

4.4.2.2 Caso 1 – Contingência perda do compensador síncrono – barra 76

Dados:

- Variáveis excluídas por dependência linear: V41 e V58.
- Valor *Bound1* – OET2: 23.48
- Número de regras C4.5: Normal: 1; Alerta: 2; Inseguro: 1

Tabela 4.28a Taxa de classificação OET2 – Caso 1.

Normal	Alerta	Inseguro	Total
<u>119</u>	13	0	132
7	<u>109</u>	16	132
0	23	<u>110</u>	133
Total Geral	Class. Correta		Class. Errada
	85.13% (338/397)		14.86% (59/397)

Tabela 4.28b Taxa de classificação C4.5 – Caso 1.

Normal	Alerta	Inseguro	Total
<u>121</u>	11	0	132
8	<u>106</u>	18	132
0	24	<u>109</u>	133
Total Geral	Class. Correta		Class. Errada
	<u>84.63% (336/397)</u>		<u>15.36% (61/397)</u>

Tabela 4.28c – Taxa de classificação por regra C4.5 – Caso 1.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>127</u>	31	0	96.21% (127/132)	11.70% (31/265)
02	21	<u>108</u>	44	82.56% (109/132)	44.28% (119/265)
03	23	<u>7</u>	54		
04		10	<u>84</u>	63.16% (84/133)	3.79% (10/263)
Total Geral	Class. Correta			Class. Errada	
	<u>80.60% (320/397)</u>			<u>40.30% (160/397)</u>	

Tabela 4.28d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso 1.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
Caso Base	Normal	1	0.00%	0.00%
	Alarme	2	406.54%	0.00%
	Alarme	3	0.13%	0.13%
	Inseguro	4	0.00%	4.15%

Tabela 4.28e - Taxa de classificação por regra VIAi – Caso 1.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>127</u>	31	0	96.21% (127/132)	11.70% (31/265)
02	27	<u>112</u>	61	84.85% (112/132)	43.37% (116/265)
03	15	<u>7</u>	42		
04		10	<u>84</u>	63.16% (84/133)	3.79% (10/263)
Total Geral	Class. Correta			Class. Errada	
	<u>81.36% (323/397)</u>			<u>39.55% (157/397)</u>	

Os resultados mostram que o OET2 é mais preciso em classificar que as regras extraídas pelo C4.5 (85.13% / 84.63%). As variáveis destacadas pelo VIA foram exclusivamente as tensões (Apêndice F – caso 1, 118 barras). Comparando-se os resultados encontrados para as

taxas de classificação das regras do C4.5 e do VIAi, nota-se que a regra 2 (classe Alerta) teve aumento na taxa de classificação errada e aumento na taxa de classificação correta. A regra 3 teve uma redução na taxa de classificação errada. Entretanto, na taxa global de classificação para ambas taxas ocorreu melhora em relação C4.5.

4.4.2.3 Caso 2 – Contingência perda de linha de transmissão - barras 76 - 77

Dados:

- Variáveis excluídas por dependência linear: V9 e V5.
- Variáveis com impedância característica nula: ZI9, ZI5 e ZI30
- Valor *Bound1* - OET2:48.31
- Número de regras C4.5: Normal: 1; Alerta: 2; Inseguro: 1

Tabela 4.29a Taxa de classificação OET2 – Caso 2.

Normal	Alerta	Inseguro	Total
<u>115</u>	16	1	132
3	<u>106</u>	23	132
0	19	<u>114</u>	133
Total Geral	Class. Correta		Class. Errada
	84.38% (335/397)		14.86% (62/397)

Tabela 4.29b Taxa de classificação C4.5 – Caso 2.

Normal	Alerta	Inseguro	Total
<u>129</u>	2	1	132
17	<u>98</u>	17	132
1	25	<u>107</u>	133
Total Geral	Class. Correta		Class. Errada
	84.13% (334/397)		15.90% (63/397)

Tabela 4.29c – Taxa de classificação por regra C4.5 – Caso 2.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>130</u>	43	0	98.48% (130/132)	16.27% (43/265)
02	08	<u>108</u>	16	86.36% (114/132)	18.87% (50/265)
03	26	<u>103</u>	26	80.45% (107/133)	6.82% (18/263)
04		18	<u>107</u>		
Total Geral	Class. Correta			Class. Errada	
	88.41% (351/397)			27.96% (111/397)	

Tabela 4.29d - Índice de ganho de informação - C4.5/VIA – VIAi/VIAii – Caso 2.

Caso	Classe	Regra	Ganho (C4.5/VIA)	Ganho (VIAi/VIAii)
Caso Base	Normal	1	0.00%	0.00%
	Alarme	2	406.54%	0.08%
	Alarme	3	0.13%	0.05%
	Inseguro	4	0.00%	0.00%

Tabela 4.29c – Taxa de classificação por regra C4.5 – Caso 2.

Regra	Normal	Alerta	Inseguro	Total Class. Correto	Total Class. Errado
01	<u>130</u>	42	0	98.48% (130/132)	15.85% (42/265)
02	08	<u>102</u>	12	92.42% (122/132)	18.87% (50/265)
03	26	<u>116</u>	22	80.45% (107/133)	6.82% (18/263)
04		18	<u>107</u>		
Total Geral	Class. Correta			Class. Errada	
	<u>90.43% (359/397)</u>			<u>27.71% (110/397)</u>	

Os resultados mostram que o OET2 é um pouco mais preciso em classificar que as regras extraídas pelo C4.5 (84.38% / 84.13%). As variáveis destacadas pelo VIA foram exclusivamente as tensões (Apêndice F – caso 2, 118 barras). Comparando-se os resultados encontrados para as taxas de classificação das regras do C4.5 e do VIAii, a taxa de classificação errada da regra 1 (classe Normal) foi ligeiramente diminuída. A regra 2 teve sua taxa de classificação correta piorada e esta, para a regra 3, foi melhorada. Entretanto, a taxa de classificação correta total para classe Alerta aumentou.

4.4.3 – discussões sobre as simulações

Os resultados das simulações para os sistemas teste IEEE 14 e 118 barras para a tarefa de estimar a margem de potência ativa mostram que o OET2 é bastante preciso. A média dos erros médios quadráticos em relação à média das margens de potência ativa foi de 0.65% (0.7 MW em 107,0 MW – 14 barras) e 2.66% (25.4 MW em 954 MW – 118 barras). O procedimento para reduzir a dimensionalidade do vetor de entrada escolhido está de acordo com o *ranking* do vetor tangente. Em ambos os sistemas foi selecionado um número baixo de variáveis. Entretanto, esta redução de dimensionalidade afetou bem mais o sistema 118 barras. Isso mostra que a

informação contida nas variáveis selecionadas para o 118 barras foi menos efetiva para estimar a margem de potência, fato observado através do crescimento do erro médio quadrático.

As taxas de classificação encontradas para todo o conjunto de simulações mostram que o OET2 é mais preciso que o C4.5. Além disso, os erros nas faixas mais não lineares (Alerta e Inseguro) são mais baixos também no OET2, fato este que reforça a característica de lidar melhor com dados não lineares de uma rede neural.

O ganho de informação das regras do VIA em relação às do C4.5 (primeiro processamento), na maioria dos casos para o sistema 14 barras foi alto. Assim, grande parte das regras foi especializada. Este fato realça o critério de *fidelidade* e expõe a qualidade de uma rede neural em avaliar de forma conjunta as variáveis de entrada em relação a uma classe de interesse. Observou-se que as tensões são as variáveis mais relevantes para o sistema 14 barras, reveladas tanto pelas regras do C4.5 quanto pelas do VIA. Isto também é observado para o sistema 118 barras. Vale lembrar que as variáveis tensões encontram-se implicitamente agregadas ao cálculo das impedâncias características. Entretanto, conforme mostra a referência [MEN99], estas impedâncias mostram-se efetivas para a tarefa de estimar a margem de potência ativa do sistema.

No conjunto de simulações notou-se que a maioria das melhoras nas taxas de classificação ocorreu para as regras extraídas para a classe Alerta. Nesta classe ocorrem os maiores erros de classificação pois apresenta duas fronteiras de decisão. O resultado disso foi uma maior especialização destas regras, fato este observado tanto para o C4.5 quanto para o VIA. De uma forma geral, notou-se que a precisão deste conjunto de regras aumenta com a especialização.

A classe Normal apresenta as melhores taxas de classificação. Em apenas duas simulações a taxa de classificação errada foi melhorada. Isto é devido ao fato das regras para esta classe serem as mais precisas (região menos não linear), resultando numa menor especialização. Para a classe Inseguro, apenas 3 casos apresentaram melhora na taxa de classificação errada. Mas, neste caso, a causa deveu-se ao fato das variáveis selecionadas serem menos efetivas para esta região de operação. Isso determinou a busca no espaço de hipóteses de novas combinações de variáveis e intervalos. Logo, o procedimento adotado para o segundo processamento não é adequado para esta classe em particular, requerendo aprimoramento futuros que solucionam esta deficiência.

Os resultados encontrados para o sistema 118 barras apresentam um ganho de informação bem baixo em relação ao C4.5. Apesar da informação ainda estar mais concentrada nas variáveis tensões, esta esta bem mais distribuída pelos pesos das conexões. Logo, as hipóteses iniciais fornecidas pelo C4.5 tornam-se mais robustas perante o VIA. Isto acarretou em explicitar ou

mesmo refinar poucas variáveis nas regras extraídas. O resultado encontrado indica que a redução de dimensionalidade como proposta aqui afeta os resultados do VIA. Isto também auxilia em visualizar a característica importante do VIA que é focar o refinamento nos intervalos relacionados com as variáveis de maior relevância (maior volume de informação).

4.5 CONCLUSÕES

A utilização do MC para gerar uma base de dados consistente mostrou-se efetiva, uma vez que podem ser introduzidos dispositivos que simulam um sistema elétrico real. Este foi o propósito de introduzir uma perturbação aleatória na geração das curvas PVs de cada caso, simulando o caráter aleatório do comportamento da carga elétrica. Isso possibilitou também aumentar a não linearidade da base de dados.

Os resultados encontrados indicam que a rede neural OET2 é bastante precisa quando é utilizada em problemas de regressão. Comparando-se esta com a RNA-ML, observou-se que a precisão de ambas é equivalente. Entretanto, notou-se que a velocidade de treinamento do OET2 é bastante elevada quando a cardinalidade do conjunto de treinamento é grande. O mesmo não ocorre com a RNA-ML, cujo treinamento mostrou-se muito lento comparado com o OET2 para o mesmo conjunto.

Para o problema de classificação, objetivo deste trabalho, foram feitas comparações entre as taxas de classificação das regras extraídas pelo C4.5 em relação ao OET2. Estes indicam também a maior precisão do OET2 em relação à C4.5. Entretanto, como as redes neurais não apresentam resultados simbólicos, foi utilizado um algoritmo que extrai regras de produção (IF THEN) destas. Isto possibilitaria comparar efetivamente as DTs e as RNAs, uma vez que ambas agora produzem resultados simbólicos. Para este propósito foi implementado o algoritmo VIA.

As hipóteses testadas pelo VIA corresponderam exatamente as regras obtidas pelo C4.5. Ou seja, um ponto inicial no espaço das regras prováveis é fornecido ao VIA. A idéia foi que as regras refinadas pelo VIA obedecem ao critério de *fidelidade*, esperado para um algoritmo de extração de regras. Os resultados do primeiro processamento do VIA para estas hipóteses mostram que o algoritmo não é suficientemente robusto para refinar os intervalos, buscando a melhora nas taxas de classificação. Entretanto, foi implementado um critério simples no sentido de mostrar que a *precisão* das regras pode ser perseguida.

Uma vez realizados o primeiro processamento do VIA, este critério é utilizado. Os resultados mostram que as taxas de classificação podem ser melhoradas a partir de um

procedimento de gerar e testar regras. Outra questão observada indica que a redução de dimensionalidade afeta a qualidade das respostas do VIA. Se a informação estiver muito distribuída, pode não obter especialização das regras e mesmo uma melhora nas taxas de classificação.

A conclusão final obtida mostra que as regras simbólicas geradas a partir das redes neurais artificiais podem empreender uma melhora significativa na qualidade destas. Além da maior precisão, as variáveis explicitadas apresentam maior relação com o domínio do problema. Ou seja, o processo de *data mining* para avaliação de segurança de tensão pode ser feito pelas redes neurais.

Capítulo 5

CONCLUSÕES E PROPOSTAS

5.1 INTRODUÇÃO

Neste capítulo são mostradas as conclusões obtidas neste trabalho de tese. As propostas para que trabalhos futuros sejam executados a partir dos prováveis desdobramentos desta tese são também apresentadas.

5.2 CONCLUSÕES DO TRABALHO

A aplicação de técnicas de inteligência artificial em problemas de sistemas elétricos de potência tem procurado solucionar questões em que as técnicas convencionais não se mostram eficazes. Este é o caso de aplicações onde é exigido um alto esforço computacional e a resposta a um problema deve ser colocada de forma rápida e precisa. Ambientes com estas características são encontrados em centros de controle da operação de sistemas elétricos, onde o operador necessita de informações rápidas, seguras e com a máxima clareza. Em avaliação de segurança de tensão, as principais questões apresentadas ao operador são: quais são as condições de operação no presente instante e para onde o sistema pode evoluir. As respostas a estas perguntas requerem processamento de um volume muito grande informações. Além disso, o resultado deste processamento deve ser colocado ao operador para auxiliá-lo no processo de tomadas de decisão. Isto possibilita estabelecer ações de controle convenientes. Logo, o desenvolvimento de ferramentas que possuam estas características se faz necessário, fornecendo informações

qualitativas ao operador do sistema. Isto pode ser feito utilizando-se de técnicas de inteligência artificial que produzam um conjunto de informações baseado no conhecimento adquirido sobre o sistema elétrico e o respectivo problema operativo. Para esta tarefa, uma metodologia deve ser implementada para extrair este conhecimento. Entretanto, no sentido de diminuir a interferência humana neste processo, a proposta é implementar técnicas que realizem esta tarefa automaticamente.

De acordo com estas considerações, o objetivo deste trabalho foi contribuir com estudos comparativos entre técnicas de AAC que possam nortear o desenvolvimento de ferramentas de apoio à tomada de decisão baseadas em técnicas de IA para o processamento de AST.

Conforme mostrado no Capítulo 01, as pesquisas apontam uma grande diversidade em aplicações de técnicas de IA para problemas de avaliação de segurança de tensão. Neste contexto, a utilização de redes neurais está amplamente difundida, tanto para problemas de classificação como de estimação. Muitas destas pesquisas trabalham com as questões relacionadas ao desempenho da rede neural, como a redução da dimensionalidade do vetor de entrada, técnicas de otimização do treinamento, redução da cardinalidade do conjunto de treinamento, métodos de aprendizado, etc. Outras técnicas são também utilizadas, como aprendizado por máquina e de reconhecimento estatístico de padrões.

O mapeamento produzido pelas RNAs possibilita lidar com problemas do mundo real, cuja principal característica é uma não linearidade muito alta. Isto proporciona a utilização destas para proceder aproximações funcionais não-lineares para um amplo espectro de funções. Entretanto, a utilização das RNAs como agentes para AAC sofre algumas restrições por apresentarem um modelo tipo ‘caixa-preta’ e tempo computacional elevado de treinamento. Comparando-se como outras técnicas, alguns resultados apontam que as RNAs são mais precisas. Uma das principais causas da maior precisão é devido às variáveis que governam um problema serem testadas simultaneamente no espaço de representações (avaliação conjunta). Em consequência, isto colabora para a detecção mais precisa de variáveis mais correlacionadas entre si e com uma dada classe. Árvores de decisão, por exemplo, trabalham estas variáveis no espaço unidimensional, ou seja, são testadas uma a uma (avaliação seqüencial). Este fato acarreta menos precisão na classificação e estas variáveis podem apresentar fracas correlações, ocasionando perda de informações relevantes do processo.

Os estudos a respeito da estabilidade de tensão apresentam um estágio de desenvolvimento considerado maduro, onde o fenômeno do colapso de tensão é adequadamente entendido, como mostrado no Capítulo 02 e Apêndice A. Novas metodologias analíticas, tais

como o método do vetor tangente, identificam o ponto de colapso e destacam a característica local do fenômeno, através da indicação antecipada das barras críticas. Dois métodos baseados no vetor tangente são propostos como identificador do ponto de colapso. O método da extrapolação quadrática e o método da continuação. O da extrapolação quadrática é caracterizado pela rapidez no processamento, mesmo para sistemas de grande porte. Entretanto, não traça o diagrama de bifurcação. O método da continuação mostra a evolução do sistema ao colapso (diagrama da bifurcação), mas apresenta o inconveniente de exigir alto esforço computacional.

O Capítulo 2 apresentou a metodologia para a formação de uma base de dados contendo as informações sobre um sistema elétrico. Como o problema de avaliação de segurança está relacionado com a indicação de operação segura, mesmo após a ocorrência de distúrbios no sistema, é necessário observar os prováveis cenários topológicos que o sistema elétrico pode evoluir. Logo, programa de seleção e processamento de contingências deve ser implantado. O método da continuação possibilita gerar a massa de dados necessária e é a metodologia analítica aplicada ao propósito. Para melhorar a representatividade da base de dados, o aplicativo de formação desta é dotado de implementações adicionais que tornam as simulações de um sistema elétrico mais realistas.

O Capítulo 3 mostrou a importância da etapa de pré-processamento dos dados. Um dos principais problemas é o da *maldição da dimensionalidade*, que afeta significativamente a qualidade das respostas das técnicas utilizadas para aprendizado automático. Como no mundo real, na maioria das vezes, não se tem a possibilidade de dispor de um número elevado de exemplares para um conjunto de dados, necessita-se reduzir a dimensão do vetor de entrada para que a técnica utilizada seja eficaz. Neste capítulo foi abordado também o tipo de inferência na base de dados que uma técnica executa. Foi mostrado que as técnicas onde as variáveis são testadas de forma conjunta no espaço de representações são mais precisas que as que testam variáveis de forma individual. Um exemplo que confirma estas questões é o mapeamento executado por uma rede neural em contraste com o executado por árvore de decisão. As redes neurais mostram-se mais precisas em classificar que as árvores de decisão.

Neste capítulo, os aspectos seqüencial e individual do mapeamento feito por uma árvore de decisão bem como algoritmo C4.5 são apresentados. Este algoritmo apresenta uma etapa de desenvolvimento superior ao seu algoritmo de origem, ID3. Contrastando com o ID3, o C4.5 possibilita lidar com problemas que contenham dados contínuos e com presença de ruído. Além disso, para tentar minimizar o problema do sobre-ajuste para o conjunto de treinamento, um processo de poda das regras extraídas pelo C4.5 é implementado.

Na seqüência, o capítulo mostra como é feito o processamento de uma rede neural tipo *feedforward*. São abordadas as questões relativas à utilização de redes neurais com uma camada e redes multicamadas. Foi mostrado que redes multicamadas proporcionam lidar com problemas não lineares, além de lidarem melhor com a presença de ruídos nestes. Apesar destas qualidades, uma rede neural se apresenta como uma caixa-preta. Entretanto, o conhecimento adquirido no processo de aprendizagem de uma rede neural pode ser extraído. Além disso, este conhecimento pode ser expresso de forma simbólica, formando uma base de conhecimento sobre o problema. Este capítulo mostra como pode ser extraído este conhecimento de uma rede neural através de um exemplo didático.

Atualmente, muitos algoritmos são implementados com a finalidade de extrair conhecimento simbólico de uma rede neural. Isto proporcionou que um conjunto de critérios fosse apresentado com o intuito de avaliar estes algoritmos. Cinco critérios básicos foram definidos: *Formato* e *Qualidade* da regra, *Translucidez*, *Complexidade* e *Portabilidade* da técnica. A observação destes critérios para um conjunto de algoritmos descritos, auxiliou a escolha de um algoritmo utilizado neste trabalho. Um dos algoritmos descritos, BRAINNE - Apêndice D, foi implementado e aplicado ao problema de avaliação de segurança de tensão. Entretanto, devido à técnica utilizada, este algoritmo é computacionalmente ineficiente pois, na maioria das vezes, é necessário proceder chamadas recursivas das redes neurais envolvidas. Além disso, a cada novo processamento, a topologia da rede neural *Backpropagation* deve ser mudada, aumentando a complexidade de implementação. Outro detalhe observado diz respeito a *portabilidade* do BRAINNE. Na tentativa de melhorar a eficiência computacional, foi testada a substituição da *Backpropagation* pelo OET2. Os resultados mostraram que não é possível utilizar o OET2 de acordo à técnica utilizada pelo BRAINNE. Desta forma, a utilização do BRAINNE neste trabalho foi abandonada.

A escolha do algoritmo de extração de regras para ser utilizada como agente de aquisição automática de conhecimento recaiu sobre o VIA. Este algoritmo é portátil pois não necessita que a rede neural utilizada sofra qualquer mudança em sua topologia ou no algoritmo de treinamento. Além disso, atende ao requisito de *fidelidade* das regras extraídas devido ao fato da convergência do algoritmo só ser possível se os intervalos iniciais das saídas dos neurônios refletirem o mapeamento produzido pelos pesos das conexões. Outra fato determinante na escolha do VIA diz respeito a similaridade da técnica utilizada com o algoritmo de treinamento OET2. No VIA, o refinamento dos intervalos é feito individualmente para cada unidade da rede neural utilizando um método de programação linear. No OET2, o mapeamento é feito através de métodos baseados

em mínimos quadrados não linear. Além disso, o OET2 é extremamente rápido, mesmo para grandes bases de dados. Isso proporciona não acrescentar esforço computacional significativo ao conjunto OET2/VIA. Outra grande vantagem do OET2 é a precisão, mesmo comparando-se com as RNAs-ML.

A principal desvantagem do VIA refere-se a formulação das hipóteses iniciais. Estas hipóteses devem ser obtidas através de um processo de busca. Logo, o esforço computacional empreendido neste processo de gerar e testar hipóteses é muito alto. Entretanto, pode-se minimizar este esforço através de técnicas auxiliares que indiquem as hipóteses iniciais. Neste trabalho, as próprias regras extraídas pelo C4.5 são utilizadas como as hipóteses iniciais para o VIA.

Utilizando um sistema teste de 3 barras, o Capítulo 3 mostra também a aplicação do VIA quando as hipóteses iniciais consideram o desconhecimento total sobre os intervalos das variáveis. Os resultados mostram que o VIA não é suficientemente robusto para extrair regras precisas com um único teste, mostrando a necessidade de explorar o espaço das hipóteses. Outro teste executado, utiliza-se das hipóteses geradas pelo C4.5 para este sistema teste. Os resultados indicam que, num primeiro processamento do VIA, as regras extraídas foram especializadas, mas as taxas de classificação permaneceram iguais às do C4.5. Utilizando-se de um critério definido neste trabalho, as regras extraídas pelo VIA foram novamente testadas com a restrição do intervalo da variável com maior alteração percentual em cada regra. Estas novas regras são então testadas. Os resultados mostram que as taxas de classificação das novas regras foram melhoradas em relação à hipótese inicial do C4.5. Além disso, houve uma redução do super-ajuste destas regras em relação ao OET2.

No Capítulo 04, é apresentado o formato das implementações computacionais desenvolvidas no âmbito desse trabalho. Destacam-se inicialmente as implementações feitas para se obter a base de dados referentes aos sistemas elétricos em estudo. Foram feitos aprimoramentos no programa de fluxo de carga e no método da continuação para que as simulações se tornassem mais próximas de um sistema real. Foi introduzido também um aplicativo para automatizar o processo de seleção de contingências, disparar o MC e formatar a base de dados de acordo com as variáveis escolhidas para compor o vetor de entrada dos padrões de treinamento/teste.

Utilizando o caso base para o sistema teste IEEE 14 barras, foi mostrada a sensibilidade dos resultados quando uma perturbação aleatória é introduzida na direção de crescimento de carga. Observou-se que esta perturbação afeta negativamente a margem de potência ativa, sendo

que os pontos de operação próximos ao ponto de colapso são os mais afetados.

A seguir, o Capítulo 4 apresentou os critérios adotados em cada etapa das implementações para a extração de regras de produção. Após ser selecionado o caso a ser analisado, conforme já informado, é necessário reduzir a dimensionalidade do vetor de entrada. Isto se deve ao fato do programa C4.5 permitir apenas que conjuntos de treinamento e teste tenham no máximo 400 exemplares. A definição das variáveis foi feita de acordo com o *ranking* das barras críticas obtidas pelo Método do Vetor Tangente. A metodologia aqui implementada utiliza a informação das barras críticas de acordo o *ranking*, averiguado por classe do problema. O objetivo neste caso foi verificar as barras críticas mais correlacionadas com um dado nível operativo.

O programa implementado para a montagem das bases de dados para o C4.5 mantém a proporcionalidade de exemplares entre as classes. Esta uniformidade na distribuição de dados entre as classes é benéfica também ao algoritmo OET2. Entretanto, dois outros conjuntos de treinamento/teste são montados com todos os dados disponíveis na base de dados para cada caso. Cada um tem 60% e 40% dos exemplares, distribuídos aleatoriamente. Estes conjuntos são montados para a tarefa de estimação da margem de potência ativa. A tarefa de estimar essa margem é realizada pelo OET2. Os resultados encontrados para o caso base do sistema teste 14 barras mostram que o erro médio quadrático é bastante baixo. Vale lembrar que o ciclo de treinamento do OET2 é bastante rápido, mesmo para grande conjunto de dados. Este fato não ocorre em RNAs-ML, cuja velocidade de treinamento é muito baixa. Apesar da ineficiência computacional, a RNA-ML pode se mostrar um pouco mais precisa que o OET2, não sendo este fato justificativa para sua utilização em detrimento do OET2.

Para a tarefa de classificar os níveis operativos, as comparações entre as taxas de classificação proporcionadas pelo OET2 são melhores que as proporcionadas pelas regras extraídas pelo C4.5. Além disso, verificou-se que o OET2 erra menos nas regiões mais não lineares, isto é, próximas da região de instabilidade. Este fato mostra que, apesar do C4.5 lidar com dados não lineares, o crescimento da não linearidade favorece o OET2. Estas comparações só são possíveis porque é implementado no algoritmo do C4.5 uma ‘competição estatística’ que possibilita classificar um exemplar por apenas uma regra, mesmo que indevidamente.

Com o intuito de proporcionar uma comparação qualitativa das regras extraídas pelo C4.5 e pelo VIA, foi proposto um índice que indica o ganho que a regra extraída pelo VIA obteve em relação à regra inicial gerada pelo C4.5. A idéia é que se os antecedentes de uma regra extraída for composto por todas as variáveis do vetor de entrada, esta regra é considerada mais específica

para o problema de avaliação de segurança de tensão. Além disso, os intervalos de cada variável devem ser exatamente a faixa de valores destas para a classe de interesse. Esta abordagem é proposta porque a informações de poucas barras críticas, geralmente, não é suficiente para a análise do problema de segurança. Apesar do fenômeno do colapso de tensão ser local, a informação de uma região ou regiões críticas apresenta-se mais adequada sob o enfoque de ações de controle. Logo, se o VIA especializa uma regra, então houve um ganho de informação nessas regras em relação às regras iniciais oriundas do C4.5.

Utilizando a metodologia proposta no Capítulo 3, foram feitas simulações para os sistemas teste IEEE 14 e 118 barras. A primeira avaliação diz respeito ao valor do erro médio quadrático obtido pelo OET2 para a estimação da margem de potência ativa. Para o sistema 14 barras, verificou-se para o conjunto de simulações uma média de 0,7% para os erros médios quadráticos encontrados. Para o sistema 118 barras esta média foi de 2,7%. Estes resultados indicaram que a redução de dimensionalidade do vetor de entrada para o caso 118 barras foi mais prejudicial, uma vez que a perda de informação para o problema de estimação foi mais acentuada.

Na seqüência, pode-se verificar os resultados para o problema de classificar os níveis operativos. Nota-se que em todos os casos simulados para ambos os sistemas teste, as taxas de classificação pelo OET2 foram melhores que as das regras do C4.5. Este fato explicita a forma mais precisa da inferência neural em uma base de dados, como exposto no Capítulo 3.

As regras extraídas pelo VIA mostram que para o sistema IEEE 14 barras o ganho de informação foi grande em relação ao C4.5. Logo, as regras foram especializadas. Este resultado reflete o treinamento do OET2 que correlaciona um maior número de variáveis com as classes de interesse. Um fato observado diz respeito às variáveis selecionadas. Nas hipóteses iniciais geradas pelo C4.5, a maioria das regras destaca as tensões como as variáveis mais relevantes. Isso também foi notado nas regras extraídas pelo VIA. Logo, as informações a respeito do sistema foram concentradas nestas variáveis. Fato este também observado para o sistema IEEE 118 barras. Além disso, vale lembrar que as variáveis tensões encontram-se implicitamente agregadas ao cálculo das impedâncias características. Entretanto, segundo a referência [MEN99], as impedâncias mostram-se efetivas para a estimação da margem de potência ativa, fato este não observado neste problema de classificação.

No segundo processamento do VIA, em que são redefinidos os intervalos das variáveis mais alteradas no primeiro processamento, foi observado que dos 10 casos simulados para o sistema IEEE 14 barra, em apenas para um caso não foi notada melhora nas taxas de

classificação. Vale salientar que, na grande maioria, houve redução das taxas de classificação errada na classe Alerta, observando-se que as regras pertencentes a esta classe são as mais imprecisas, pois esta classe apresenta duas fronteiras de decisão. A classe Normal apresenta melhores taxas de classificação, tanto para o C4.5 tanto quanto para o VIA. Entretanto, para a classe Inseguro, em apenas 3 casos foram alteradas as taxas correspondentes. Lembrando que esta região é a mais não linear, estes resultados mostram que as variáveis selecionadas para esta classe não são efetivas. Isso determina a busca no espaço de hipóteses de novas combinações de variáveis e intervalos. Logo, o procedimento adotado para o segundo processamento não é adequado para esta classe em particular, requerendo aprimoramento futuros que solucionam esta deficiência.

No sistema teste 118 barras, o ganho de informação foi baixo em relação ao C4.5. Isto mostra que a concentração de informação é mais distribuída nos pesos das conexões das variáveis selecionadas. Neste caso, as hipóteses iniciais tornam-se robustas perante o VIA, levando este a explicitar, e mesmo refinar, poucas variáveis nas regras extraídas. Assim, o VIA converge para o ponto dado pela hipótese inicial. Logo, como observado também no problema de estimação, a redução de dimensionalidade como proposta não é benéfica para este sistema em particular. Estes resultados mostram uma característica importante do VIA, que é refinar os intervalos relacionados com variáveis com maior relevância. Isso reforça o critério de *fidelidade* das regras extraídas pelo VIA.

De acordo com os resultados encontrados neste trabalho de tese, nota-se que a extração de regras de produção através das redes neurais é uma alternativa viável aos algoritmos de árvore de decisão. Foi mostrado que o VIA reproduz com fidelidade o treinamento do OET2, embora a precisão do conjunto de regras seja dependente da hipótese inicial formulada. Observou-se também que o problema da dimensionalidade afeta o VIA, quando as informações estão mais distribuídas nos pesos das conexões, exigindo para o mesmo conjunto de variáveis um maior volume de testes de hipóteses.

5.3 PROPOSTA PARA TRABALHOS FUTUROS

No desenvolvimento deste trabalho foram observadas algumas dificuldades e tendências das metodologias implementadas. Assim, pode-se propor alguns trabalhos que proporcionem a continuidade do desenvolvimento de técnicas de aquisição automática de conhecimento aplicadas

ao problema de avaliação de segurança de tensão. São elas:

- Análise de novas técnicas de redução de dimensionalidade do vetor de entrada, cujas variáveis selecionadas sejam mais efetivas, principalmente para sistemas elétricos de grande porte.
 - Desenvolvimento de técnicas que possibilitem ao algoritmo BRAINNE trabalhar sem a recursividade das chamadas das redes neurais envolvidas.
 - Desenvolvimento de metodologias que tornem eficiente a busca de hipóteses a serem testadas pelo VIA, para que um conjunto maior de regras possa ser extraído, melhorando de *precisão* e visualização do problema de interesse.
 - Testar novas técnicas de programação linear e não linear em substituição ao método Simplex Revisado no algoritmo VIA, obtendo intervalos refinados de variáveis efetivamente mais relevantes.
 - Melhoramentos nos programas de fluxo de carga e do método da continuação para tornarem mais realista as bases de dados.
 - Desenvolvimento de técnicas de inferência na base de conhecimento montada pelo VIA para auxiliar sistemas de controle baseados em conhecimento, utilizados para avaliação *on-line* de segurança de tensão.
-

Apêndice A

ESTABILIDADE DE TENSÃO

A.1 INTRODUÇÃO

Neste apêndice são apresentadas as principais técnicas utilizadas para avaliar a estabilidade de tensão de um sistema elétrico, bem como alguns respectivos índices propostos como indicadores de proximidade de colapso de tensão.

A.2. MATRIZES DE SENSIBILIDADE

Neste procedimento, a sensibilidade das variáveis de estado do sistema é obtida através da matriz Jacobiana do fluxo de carga para um ponto de operação conhecido. Esta análise permite determinar as barras cujas variáveis de estado variam com maior intensidade em relação a uma variação de carga. Se o sistema elétrico não sofre nenhuma variação de potência ativa [SCH90], a matriz Jacobiana (que neste caso contém as equações de potência ativa das barras de tensão controlada) é reduzida às derivadas parciais das equações de potência reativa em relação ao nível de tensão. Observa-se neste caso que a barra swing absorve as variações de perdas elétricas do sistema.

Devido a não linearidade das equações de sistemas de potência, as informações obtidas para um ponto de operação conhecido podem não ser válidas para outros pontos de operação. Portanto, além de determinar a barra/área crítica do sistema para um ponto de operação conhecido, deve-se saber se esta barra/área será a mesma na medida em que um aumento no carregamento do sistema o dirigir ao ponto de colapso. Estas informações, entretanto, não são fornecidas pelas matrizes de sensibilidade.

A.3 VALORES SINGULARES E AUTOVALORES

Consiste em calcular os menores valor singular ([LOF92], [BAR95]) e autovalor [[KUN90], [MAR94]) da matriz Jacobiana, para cada ponto de operação. Isto permite identificar o ponto de bifurcação. A medida em que o carregamento do sistema aumenta, o menor valor singular e o menor autovalor diminuem até se tornarem nulos, identificando o ponto de sela-nó. Entretanto, estes índices sofrem uma variação brusca no ponto de bifurcação, de acordo com os testes apresentados em [CAN95]. Portanto, tais métodos tendem a falhar se um programa de fluxo de carga convencional for empregado. Isto se deve ao fato que, tal variação brusca só é observada se a este programa for aplicado à parametrização das equações de fluxo de potência, possibilitando obter as soluções das equações do fluxo de carga para pontos situados além do ponto de colapso. Da teoria da álgebra linear, uma matriz A pode ser decomposta por valores singulares da seguinte forma:

$$A = U \Sigma V^T$$

onde U e V são os vetores singulares à direita e à esquerda, respectivamente, e Σ é a matriz de valores singulares. Para sistemas de potências, sendo a matriz A representada pela matriz Jacobiana do sistema, tem-se que:

$$\begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} = \Sigma^{-1} V U^T \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}$$

Para a decomposição por autovalores, o procedimento é o mesmo. Obtém-se então

$$A = X \lambda Y^T$$

e, aplicando aos sistemas de potência, onde A é a matriz Jacobiana do sistema, tem-se

$$\begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} = \lambda^{-1} X Y^T \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}$$

Interpretação dos autovalores

Em [LOF92] é apresentada uma interpretação dos autovalores aplicado às equações de fluxo de potência. Assuma que os autovalores da matriz A estejam dispostos na seguinte forma: $\lambda_1 > \lambda_2 > \dots > \lambda_n$ e que o autovalor λ_n seja zero, i.e., a matriz A é singular. Para um sistema de potência, a matriz A é representada pelo Jacobiano J , de forma que

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \mathbf{x} & \lambda & \mathbf{y}^T \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} \quad (\text{A.1})$$

ou

$$\begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} = \begin{bmatrix} \mathbf{x} & \lambda^{-1} & \mathbf{y}^T \end{bmatrix} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}$$

porque

$$\begin{bmatrix} \mathbf{x} & \lambda & \mathbf{y}^T \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{y}^T & -1 & \lambda^{-1} & \mathbf{x}^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{x} & \lambda^{-1} & \mathbf{y}^T \end{bmatrix}$$

Note portanto, que a inversa é facilmente obtida. Assumindo que a última coluna de \mathbf{y} (associada ao autovalor nulo) seja dada por $\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}$, tem-se:

$$\mathbf{y}_n = \begin{bmatrix} \mathbf{x}_n & \lambda_n & \mathbf{y}_n^T \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix}$$

ou

$$\begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} = (\mathbf{x}_n \quad \lambda_n \quad \mathbf{y}_n^T)^{-1} \mathbf{y}_n$$

$$\begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} = \mathbf{x}_n \quad \lambda_n^{-1} (\mathbf{y}_n^T \mathbf{y}_n)$$

Como os autovetores são ortonormais, i.e., $\mathbf{y} \mathbf{y}^T = 1$, tem-se:

$$\begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} = x_n \lambda_n^{-1} \quad (\text{A.2})$$

O produto $x_n \lambda_n^{-1} y_n^T$ tem dimensão $n \times n$, o que fornece, após a multiplicação pelo vetor de ângulos e tensões, um vetor com dimensões $n \times 1$. Note que seguindo a definição dada ao autovetor à esquerda acima, obtém-se que o autovetor à direita está associado a sensibilidade de tensão e ângulo, enquanto que o autovetor à esquerda está associado a sensibilidade de injeção de potências ativa e reativa. Portanto, o maior componente no autovetor à direita indica a variável mais sensível, i.e., a componente que mais fortemente direciona o sistema à singularidade. Por outro lado, um autovetor nulo indica que esta variável não desempenha nenhum papel na singularidade.

A referência [LOF92] aplica uma decomposição em valores singulares quando o fluxo de carga é reduzido às derivadas parciais das equações de potência reativa em relação à tensão, conforme mostrado abaixo. Assuma o Jacobiano do fluxo de carga da seguinte forma:

$$\begin{bmatrix} \Delta P_g \\ \Delta Q_g \\ \Delta Q_l \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Delta\theta_g \\ \Delta\theta_l \\ \Delta V_l \end{bmatrix}$$

Assumindo que o sistema não tenha variação de potência ativa:

$$\begin{bmatrix} 0 \\ 0 \\ \Delta Q_l \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Delta\theta_g \\ \Delta\theta_l \\ \Delta V_l \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = [A] \begin{bmatrix} \Delta\theta_g \\ \Delta\theta_l \end{bmatrix} + [B] [\Delta V_l] \quad (\text{A.3})$$

$$[\Delta Q_l] = [C] \begin{bmatrix} \Delta\theta_g \\ \Delta\theta_l \end{bmatrix} + [D] [\Delta V_l] \quad (\text{A.4})$$

Explicitando $\begin{bmatrix} \Delta\theta_g \\ \Delta\theta_l \end{bmatrix}$ na equação (A.3),

$$\begin{bmatrix} \Delta\theta_g \\ \Delta\theta_l \end{bmatrix} = - [A]^{-1} [B] [\Delta V_l]$$

e substituindo na equação (D.4), obtém-se finalmente a equação (A.5) abaixo

$$\begin{aligned} [\Delta Q_I] &= [D - CA^{-1}B][\Delta V_I] \\ [\Delta Q_I] &= [J_{QV}][\Delta V_I] \end{aligned} \quad (\text{A.5})$$

A referência [CAN95] mostra que o Jacobiano $[J_{QV}]$ é quasi-simétrico. Portanto, a decomposição por autovalores ou valores singulares é facilmente obtida. Como este tipo de matriz tem autovalores e valores singulares similares, o resultado obtido é o mesmo.

A.4 DETERMINANTE REDUZIDO

Este método baseia-se na redução da matriz Jacobiana à dimensão das equações de potência ativa e reativa de cada barra de carga em relação ao seu ângulo de fase e módulo da tensão [SOU91]. Esta matriz de dimensão 2×2 é calculada para todas as barras de carga do sistema. Para cada barra de carga, a matriz reduzida tem seu determinante calculado. A barra de carga associada ao menor determinante é a barra de crítica do sistema naquele ponto de operação. Entretanto, a barra crítica avaliada em um ponto de operação conhecido pode não ser a mesma no ponto de operação seguinte. Se a barra crítica no ponto de bifurcação é conhecida e os limites de geração de potência reativa são desprezados, o determinante reduzido calculado em relação a esta barra fornece um comportamento quadrático em relação ao fator de crescimento de carga. Quando os limites forem considerados (situação mais realista), uma descontinuidade é observada [CAN95], porém menos abrupta que aquela observada no comportamento do menor autovalor e valor singular, ocorrendo antes do ponto de bifurcação. Para as outras barras de carga, o comportamento do determinante reduzido aproxima-se do comportamento obtido para os menores autovalor e valor singular. Logo, este método não é adequado para identificar a proximidade do ponto de bifurcação a partir de um ponto de operação conhecido. A seguir é apresentado a formulação matemática do método aplicado à sistemas elétricos de potência.

Assuma o Jacobiano do fluxo de carga abaixo:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix}$$

A matriz Jacobiana acima pode ser reordenada de tal forma que as linhas e as colunas associadas as equações da barra l sejam trocadas com as duas últimas linhas e colunas, obtendo:

$$\begin{bmatrix} \Delta P \\ \Delta Q \\ \Delta P_1 \\ \Delta Q_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \\ \Delta \theta_1 \\ \Delta V_1 \end{bmatrix}$$

Assuma que para um dado ponto de operação, o sistema não tenha nenhuma variação de carga ou geração, exceto na barra de carga l , que tem uma variação de carga infinitesimal. Como nenhum dos geradores experimenta variação de geração, a barra swing deve absorver o acréscimo de potência na barra l e as diferenças de perdas nas linhas. Essas considerações permitem que a matriz Jacobiana possa ser decomposta como:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = [A] \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} + [B] \begin{bmatrix} \Delta \theta_1 \\ \Delta V_1 \end{bmatrix} \tag{A.6}$$

$$\begin{bmatrix} \Delta P_1 \\ \Delta Q_1 \end{bmatrix} = [C] \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} + [D] \begin{bmatrix} \Delta \theta_1 \\ \Delta V_1 \end{bmatrix} \tag{A.7}$$

De (A.6) obtém-se

$$\begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} = - [A]^{-1} [B] \begin{bmatrix} \Delta \theta_1 \\ \Delta V_1 \end{bmatrix}$$

Substituindo em (A.7):

$$\begin{bmatrix} \Delta P_1 \\ \Delta Q_1 \end{bmatrix} = [D] - [C][A]^{-1}[B] \begin{bmatrix} \Delta \theta_1 \\ \Delta V_1 \end{bmatrix} \tag{A.8}$$

Considerando a matriz Jacobiana do fluxo de carga com dimensão $n \times n$, cada submatriz tem as dimensões:

$$\begin{aligned} A &= (n - 2) \times (n - 2) & C &= 2 \times (n - 2) \\ B &= (n - 2) \times 2 & D &= 2 \times 2 \end{aligned}$$

Logo, a nova matriz $D_{ll}' = D - C A^{-1} B$ tem dimensão 2×2 .

No ponto de colapso, pequenas variações de carga produzem grandes variações de tensão e ângulo. Isto é refletido em

$$\begin{bmatrix} \Delta \theta_I \\ \Delta v_I \end{bmatrix} = [D_{II}]^{-1} \begin{bmatrix} \Delta P_I \\ \Delta Q_I \end{bmatrix} \quad (\text{A.9})$$

Portanto, o menor determinante de D_{II}' indica a barra mais sensível a variações de carga, i.e., a barra crítica do sistema. Em [CAN95] são apresentados os resultados para o sistema teste IEEE 300 barras em relação ao determinante reduzido para a barra 9042, que é a barra crítica encontrada no ponto de operação inicial, e para a barra 192, a barra crítica encontrada no ponto de colapso.

A.5 MÉTODO DIRETO

Esta metodologia propõe encontrar o ponto de sela-nó através do método de Newton-Raphson modificado. Ao conjunto de equações de fluxo de carga é incorporado um conjunto de equações para impor a singularidade da matriz Jacobiana e uma outra equação para garantir a não trivialidade da solução (autovetor não nulo) [CAN91]. O método pode falhar se os limites de geração de potência reativa forem considerados. Problemas de convergência podem também ocorrer se o chute inicial estiver distante da solução. Vale salientar que este método não traça o diagrama da bifurcação.

Para achar iterativamente o ponto de bifurcação (x_0, λ_0) , o método pode ser equacionado da seguinte forma:

$$f(x, \lambda) = 0 \quad (\text{A.10a})$$

$$D_x f(x, \lambda) v \quad \text{ou} \quad D_x^T f(x, \lambda) w = 0 \quad (\text{A.10b})$$

$$\|v\| \quad \text{ou} \quad \|w\| \neq 0 \quad (\text{A.10c})$$

onde $D_x f(x, \lambda)$ é a matriz Jacobiana e, v e w são as matrizes de autovetores à direita e à esquerda, respectivamente. A equação (A.10a) representa o diagrama de bifurcação, a (A.10b) representa a imposição da condição de singularidade e (A.10c) assegura que a solução obtida é não trivial. Portanto, o sistema de equações tem dimensão $2n + 1$.

As desvantagens do método são

- Requer um bom chute inicial para obter convergência, i.e., necessita do conhecimento de um ponto de operação próximo a $(\mathbf{x}_0, \lambda_0)$. A estimativa inicial para v ou w pode ser obtida pelo vetor tangente associado ao primeiro ponto de operação.
- Problemas quando os limites de geração de potência reativa são considerados, uma vez que a estrutura da matriz Jacobiana muda de acordo com o carregamento do sistema.

Os dois métodos anteriormente descritos podem não ser eficientes quando sistemas de potência de grande porte são considerados. Entretanto, o método da continuação pode ser significativamente melhorado se partição de redes for empregada.

A.6 TÉCNICA DA FUNÇÃO DE ENERGIA

Funções de energia foram inicialmente empregadas em sistemas de potência para estudos de estabilidade transitória [CHI89]. O método consiste em avaliar a energia total (cinética e potencial) de um sistema de potência durante um distúrbio (curto-circuito, por exemplo) no sistema. Para o sistema operar estável, a energia máxima que este pode vir a ter durante o tempo de distúrbio deve ser igual a um nível de energia associado a uma condição pós-falta. Esta metodologia determina o tempo crítico de abertura para o qual o sistema se mantém estável.

O método pode também produzir resultados interessantes para a análise de estabilidade de tensão. Nesta aplicação, somente a energia potencial do sistema é necessária, dispensando a inclusão de amortecimento e modelos de geradores. As referências [OVE91a, OVE91b] mostram que a variação de função de energia pode ter um comportamento linear em função de um aumento de carga, possibilitando uma medida relativa no índice de colapso de tensão para qualquer ponto de operação. Como a distância até o ponto de colapso pode ser estimada, informações a respeito da margem de carga podem ser obtidas para qualquer ponto de operação, o que é um aspecto importante a ser considerado.

A aplicação da função de energia à análise de estabilidade de tensão baseia-se na medida da distância entre os pontos de equilíbrio estável e instável de um sistema. Como um sistema elétrico de n barras tem $2^n - 1$ possíveis soluções de fluxo de carga, a determinação da solução instável de interesse não é trivial, como relatado em [OVE91a, OVE91b]. Na medida em que aumenta-se o carregamento do sistema, o número possível de soluções diminui, até que somente uma solução exista, com nível de energia zero (ponto de bifurcação). Note que uma vez que o objetivo é determinar a margem de carga do sistema, deve-se considerar a inclusão de perdas, de forma que a função de energia não é estritamente relacionada com a função de Lyapunov.

A.7 TÉCNICAS DE PARTIÇÃO DE REDES

Os métodos expostos anteriormente são capazes de produzir resultados úteis para a análise de estabilidade de tensão. Entretanto, alguns destes índices podem falhar, se um programa de fluxo de carga convencional for empregado. Ainda que um processo de parametrização fosse incorporado, o comportamento destes índices mostra que seria necessário “caminhar” com o sistema até que o ponto de bifurcação seja encontrado. O método da função de energia, por outro lado, é capaz de estimar o ponto de colapso a partir de dois pontos de operação conhecidos, mas a determinação das soluções instáveis de interesse se constitui uma barreira. Baseado nestas dificuldades, técnicas de partição de redes pode ser de especial interesse, uma vez que a dimensão do sistema analisado pode ser substancialmente reduzida.

Diversos artigos na literatura mostram que colapso de tensão é um fenômeno que começa localmente e se espalha pelas barras vizinhas. Baseado nesta característica, procura-se analisar o índice de segurança de um sistema de potência através do estudo de uma pequena parte do sistema. Uma vez obtido o subsistema de interesse, aplica-se qualquer uma das técnicas previamente discutidas. A Referência [OVE91a] mostra que é possível particionar um sistema usando o método de Ward, que consiste em reter as barras de interesse através de manipulações na matriz Y . Entretanto, esta referência não propõe nenhum método para a identificação da área crítica, e diversas partições devem ser feitas para que o sistema inteiro seja avaliado.

A referência [SOU94] propõe uma técnica de partição de redes baseada no determinante reduzido da matriz Jacobiana, mostrado na Seção 1.2.7. Os resultados obtidos para o sistema de 24 barras do IEEE mostraram-se razoáveis, uma vez que para este sistema a barra crítica não muda durante o processo de carregamento. Além dessa característica, este sistema não é grande o suficiente para ser considerado representativo ao nível de desacoplamento entre as barras. A referência [SOU97] apresenta algumas técnicas de partição de redes baseadas no determinante do Jacobiano reduzido, autovetor à direita, vetor tangente e técnicas mixadas. Estas técnicas aplicadas ao sistema de 300 barras do IEEE fornecem resultados pobres, uma vez que a maior parte do sistema deve ser considerada até que se obtenha um subsistema representativo. A referência [OVE97] propõe uma técnica de partição onde as barras vizinhas são tratadas como barras *swing*. Tal consideração tende a produzir resultados melhores, muito embora a obtenção da solução correta não seja trivial.

A.8 PRINCIPAIS CONSIDERAÇÕES SOBRE ESTABILIDADE DE TENSÃO

Em sistemas elétricos de potência, as bifurcações mais comuns encontradas são as de Hopf e de Sela-nó. Quando utilizado o modelo de equações algébricas do fluxo de carga, apenas a bifurcação de sela-nó é identificada e esta é caracterizada por um autovalor nulo (matriz Jacobiana singular) no ponto de bifurcação. Esta característica é bastante explorada na avaliação de estabilidade de tensão.

Os índices menor autovalor e menor valor singular não são adequados como previsores do ponto de colapso, pois apresentam uma variação brusca neste ponto. Devido a matriz Jacobiana ser quasi-simétrica, estes índices são muito semelhantes entre si e, portanto, o resultado obtido é o mesmo. Para o determinante do Jacobiano reduzido, o comportamento é quadrático para a barra crítica, quando não são considerados os limites de fornecimento de potência reativa e apresenta uma descontinuidade considerando-se estes limites. Outra desvantagem destes três índices é a indicação da barra crítica somente muito próximo ao ponto de colapso.

O método direto apresenta as desvantagens de necessitar de uma boa estimativa inicial para convergência e falha quando limites de geração de potência reativa são incluídos. As técnicas de partição de redes são adequadas quando se deseja reduzir o esforço computacional requerido para o estudo de estabilidade para grandes sistemas, dividindo-os em subsistemas menores. Entretanto, apresentam a desvantagem de necessitar de diversas partições para obter o estudo completo do sistema.

Inicialmente utilizada em estudo de estabilidade transitória, a técnica da função de energia dispensa modelagem complexa de geradores e inclusão de amortecimento para estudos de estabilidade de tensão. Esta função pode ter um comportamento linear em função de um aumento de carga, possibilitando uma medida relativa do índice de colapso para qualquer ponto de operação. Isto possibilita informar a margem de carga para qualquer um destes pontos. Tem a desvantagem da não trivialidade da obtenção da solução instável de interesse, necessária à técnica.

O vetor tangente apresenta as vantagens de ser de fácil inclusão em um programa de fluxo de carga e é efetivo em indicar a barra crítica bem antes do ponto de colapso. Este apresenta também a característica de convergir ao autovetor à direita associado ao menor autovalor no ponto de colapso, identificando o ponto de bifurcação. Entretanto, falha como indicador por apresentar a mesma descontinuidade obtida pelo determinante do Jacobiano reduzido. Dois métodos são implementados utilizando o vetor tangente. O método da continuação, que visa encontrar o ponto de bifurcação e traça o diagrama da bifurcação e, o

método da extrapolação quadrática, que visa encontrar o ponto da bifurcação supondo um comportamento quadrático do vetor tangente. Este método apresenta esforço computacional reduzido comparado com o método da continuação.

Apêndice B

TÉCNICAS DE RECONHECIMENTO DE PADRÕES

B.1 INTRODUÇÃO

Este apêndice mostra algumas técnicas básicas utilizadas em reconhecimento de padrões. Estas são, em muitos casos, bases teóricas para aplicativos na área de apoio a tomada de decisões. São apresentadas as seguintes técnicas: Classificador de Bayes, estimação de densidades de probabilidades, funções discriminantes e ajuste de curvas por polinômios.

B.2 CLASSIFICADOR DE BAYES

O mais intuitivo conceito de classificação pode ser dado através da probabilidade de ocorrência de uma dada classe C_k ($k = 1, \dots, c$ classes) sem que se conheça qualquer outra informação. Ou seja, se existem 10 exemplares em um conjunto de n exemplares pertencentes à classe C_k , então a probabilidade desta classe k é de $10/n$. Formalmente, esta probabilidade é denominada de *probabilidade a priori*, representada neste caso por $P(C_k)$. Logo, se uma dada classe C_1 tem probabilidade *a priori* maior que uma classe C_2 ($P(C_1) > P(C_2)$), pode-se classificar um novo exemplar baseado nesta regra de decisão (ou critério de decisão), o que significa reduzir a probabilidade de erro.

Um dos métodos clássicos mais conhecidos aplicados a problemas de classificação é o Classificador de Bayes ([DUD73],[BIS95],[SCH97],[CHE98], [RUS95]). Supondo que sejam obtidas duas características, X_1 e X_2 , de um processo qualquer de redução de dimensionalidade aplicado a um vetor de entrada \mathbf{x} . Cada uma destas características (X_1 e X_2) apresenta um conjunto de valores discretos. Cada exemplar então tem o seu correspondente valor X^1 presente em um dado intervalo e pertencente à classe C_k . Pelo Teorema de Bayes tem-se

$$P(C_k | X^1) = \frac{P(X^1 | C_k)P(C_k)}{P(X^1)} \quad (\text{B.1})$$

onde $P(X^1 | C_k)$ indica a probabilidade de se ter o valor X^1 dado que se conhece a classe C_k ($k=1, \dots, c$ classes) do exemplar (designada de *probabilidade condicional de classe*) e $P(C_k | X^1)$ indica a probabilidade do exemplar pertencer a classe C_k , dado que se tem o valor X^1 . Esta quantidade é denominada de *probabilidade a posteriori*. $P(X^1)$ indica a probabilidade de se observar o valor X^1 para todo o conjunto, independente de qualquer classe. Na hipótese de mundo fechado, tem-se que:

$$\sum_{k=1}^c P(C_k | X^1) = 1 \quad (\text{B.2})$$

e, substituindo-se (B.1) em (B.2), obtém-se

$$P(X^1) = \sum_{k=1}^c P(X^1 | C_k)P(C_k). \quad (\text{B.3})$$

Observe que (B.3) funciona como um fator de normalização quando aplicada no âmbito de (B.1), para as c classes. Então, a classificação é feita se $P(C_k | X^1)$ for maior que qualquer $P(C_j | X^1)$, para todo j diferente de k .

Em muitos problemas, pode ser necessário representar X_1 em valores contínuos em vez da discretização proposta anteriormente. Para efeito de simplificação, deve ser suposto que um pré-processamento (normalizações e seleção de características) tenha sido efetuado, representando X_1 apenas por x_l . Logo, a representação do vetor de características é $\mathbf{x} = \{x_l, \dots, x_{dl}\}$, sendo dl a nova dimensão do vetor. Se as variáveis de \mathbf{x} são contínuas, o que se tem

agora são densidades de probabilidade. A definição de como os valores de \mathbf{x} são distribuídos no espaço $d1$ -dimensional é dada por uma *função densidade de probabilidade* $p(\mathbf{x})$, onde a probabilidade de \mathbf{x} pertencer a uma região \mathfrak{R} do espaço de \mathbf{x} é dado por:

$$\mathbf{P}(\mathbf{x} \in \mathfrak{R}) = \int_{\mathfrak{R}} p(\mathbf{x}) d\mathbf{x}. \quad (\text{B.4})$$

Observe que a função $p(\mathbf{x})$ define a forma geométrica da distribuição dos valores de \mathbf{x} no espaço, ou seja, se os valores de \mathbf{x} se distribuem segundo uma função *normal*, então tem-se um hiperparaboloide. Segundo esta premissa, o teorema de Bayes pode ser rescrito e a probabilidade *a posteriori* $\mathbf{P}(\mathbf{C}_k|\mathbf{x})$ ($k=1, \dots, c$ classes) é fornecida através da seguinte equação:

$$\mathbf{P}(\mathbf{C}_k | \mathbf{x}) = \frac{p(\mathbf{x} | \mathbf{C}_k) \mathbf{P}(\mathbf{C}_k)}{p(\mathbf{x})}. \quad (\text{B.5})$$

Neste caso, $p(\mathbf{x})$ é denominada de probabilidade incondicional e $p(\mathbf{x}|\mathbf{C}_k)$ é a função densidade condicional de classe. Assim, as Equações B.2 e B.3 tornam-se

$$\sum_{k=1}^c \mathbf{P}(\mathbf{C}_k | \mathbf{x}) = 1 \quad (\text{B.6})$$

e

$$p(\mathbf{x}) = \sum_{k=1}^c p(\mathbf{x} | \mathbf{C}_k) \mathbf{P}(\mathbf{C}_k) \quad (\text{B.7})$$

respectivamente. Vale ressaltar que, para calcular as probabilidades *a posteriori*, $\mathbf{P}(\mathbf{C}_k|\mathbf{x})$ e $\mathbf{P}(\mathbf{C}_k|\mathbf{x})$, tanto (B.1) quanto (B.5), admitem que as probabilidades *a priori* e (densidade) condicional de classes sejam conhecidas. Entretanto, no mundo real, isto pode não ser possível diretamente, o que determina que estas probabilidades devam ser estimadas por alguma técnica própria para esta finalidade.

Retornando ao problema de classificação, conhecendo-se agora $\mathbf{P}(\mathbf{C}_k|\mathbf{x})$, um exemplar \mathbf{x} designa a classe \mathbf{C}_k , com o mínimo erro de classificação se

$$P(C_k | x) > P(C_j | x) \quad \text{para todo } j \neq k. \quad (\text{B.8})$$

Sabendo-se que $p(x)$ é independente da classe, logo de (B.5) em (B.8) obtém-se que:

$$p(x|C_k)P(C_k) > p(x|C_j)P(C_j) \quad \text{para todo } j \neq k. \quad (\text{B.9})$$

Esta Equação B.9 mostra que existe um critério de decisão associado a cada classe do problema. Logo, isto estabelece regiões de decisão $\mathfrak{R}_1, \dots, \mathfrak{R}_c$, tal que um ponto que caia dentro de uma região \mathfrak{R}_k designa a classe C_k . Os limites entre estas regiões são conhecidos como *fronteiras de decisão* ou *superfícies de decisão*. A Figura B.1 mostra as *densidades de probabilidades conjuntas* ($p(x, C_k) = p(x|C_k)P(C_k)$) como função do valor da característica x , para um problema a duas classes C_1 e C_2 . Se a fronteira de decisão entre as duas curvas é estabelecida no ponto de cruzamento destas (indicado pela seta maior e o ponto), a área de erro de classificação é a menor possível.

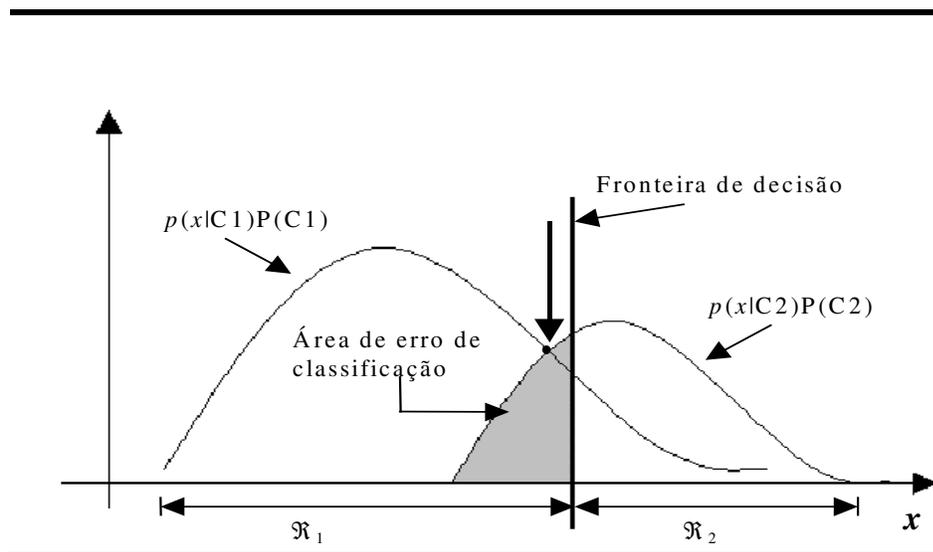


Figura B.1 - Regiões de decisão para um problema à duas classes, C_1 e C_2 .

Em muitos processos de tomada de decisão podem ocorrer situações em que errar numa dada classificação acarreta maiores prejuízos que outras. Por exemplo, num sistema elétrico o risco de classificar um dado estado de operação como normal, estando este em estado inseguro, é maior que classificar estado inseguro, estando este em estado normal. No primeiro, medidas corretivas não são implementadas, o que pode levar o sistema a blecautes. No segundo, algumas medidas emergenciais são tomadas, mas acarreta bem menos prejuízo que o caso anterior. Então,

a idéia é obter o valor do risco mínimo associado (de uma classe em relação a outra) ao classificar cada classe do problema. Este conceito é definido como *minimização do risco*.

Outro conceito que pode ser introduzido é o de rejeição. Espera-se que os maiores erros de classificação sejam encontrados onde as probabilidades *a posteriori* são baixas, podendo ou não apresentar superposição entre classes. Desta forma, pode-se estabelecer um limite mínimo (θ) para a máxima probabilidade, tal que, se a $\max_x P(C_k|\mathbf{x}) \geq \theta$ classifica-se o exemplar \mathbf{x} , senão ele é rejeitado. No exemplo para pontos de operação, algumas indicações de estados operativos podem ser desconsideradas (no sentido de classificação automática) se houver margem de dúvida quanto à correta classificação.

B.3 ESTIMAÇÃO DE DENSIDADES DE PROBABILIDADES

Conforme observado anteriormente, a utilização do Classificador de Bayes implica conhecer as probabilidades *a priori* e condicional de classes. Na maioria das vezes isto não é possível, devendo-se então encontrar algum caminho para estimar estas probabilidades.

Uma primeira alternativa, denominada de técnica *paramétrica*, é assumir conhecida a forma da função densidade de probabilidade. Se a $p(\mathbf{x}|C_k)$ é assumida ser uma função densidade *normal* (ou *gaussiana*), com média μ_k e matriz de covariância Σ_k , o problema pode ser simplificado, bastando para isto estimar estes parâmetros, os quais podem ser representados através de um vetor parâmetro θ . Para proceder estas estimações, são propostos duas metodologias, a Verossimilhança Máxima e a Inferência Bayesiana.

A técnica Verossimilhança Máxima busca encontrar os valores ótimos dos parâmetros por meio de um processo de maximização de uma função de verossimilhança [DUD73]. Esta mesma referência mostra um exemplo clássico onde assume-se que uma função de densidade normal multivariável seja conhecida e os parâmetros μ e Σ desconhecidos.

Na Inferência Bayesiana, a forma da função de densidade de probabilidade de \mathbf{x} dado que se conhece θ ($p(\mathbf{x}|\theta)$) é conhecida. Entretanto, o parâmetro θ é inicialmente conhecido por sua probabilidade *a priori* $p(\theta)$. Este conhecimento é próprio da noção inicial sobre a faixa de valores de um dado parâmetro. Esta é geralmente ampla, fruto da incerteza da não observação dos dados até o momento. Então, o que se deseja é, observando-se os dados, estimar os parâmetros, ou seja, a probabilidade *a posteriori* $p(\theta|\mathbf{X})$, onde \mathbf{X} é o conjunto dos N exemplares observados. Note que, à medida que o número de exemplares cresce, os valores encontrados para θ aproximam-se

do valor máximo estimado para a função de verossimilhança.

A segunda alternativa proposta na literatura é a utilização de técnicas *não-paramétricas* para processar estimação de densidades de probabilidades. Diferentemente da paramétrica, aqui as formas das funções de densidade não são conhecidas, mas admite-se que estas podem ser obtidas dos dados.

Considere a função de densidade $p(\mathbf{x})$ desconhecida, onde \mathbf{x} é um novo exemplar a ser testado. Esta função pode ser aproximadamente estimada pela equação

$$p(\mathbf{x}) \cong \frac{K}{NV} \quad (\text{B.10})$$

onde N é o número total de exemplares e K é uma fração destes N exemplares. Se há uma probabilidade diferente de zero destes K exemplares estarem dentro de uma região \mathfrak{R} , então o volume de \mathfrak{R} é dado por V . A aplicação da Equação B.10 para problemas de estimação de densidade envolve os dois métodos. No Método Baseado em *Kernel* (também conhecido como Janela de Parzen), o volume V é fixado e K pode então ser determinado. Para o Método dos *K-Vizinhos-Mais-Próximos* (*K-NN – K-Nearest Neighbours*), escolhe-se um valor fixo de K e determina-se o volume V correspondente aos dados. Em [DUD73] é visto que à medida que N tende ao infinito, ambos os métodos convergem à densidade de probabilidade verdadeira. Nesta situação, V diminui e K cresce com o aumento de N .

B.3.1 Método Baseado em Kernel:

Considere uma região \mathfrak{R} do espaço d -dimensional, sendo esta um hipercubo de lado h . Então o volume deste hipercubo é dado por:

$$V = h^d. \quad (\text{B.11})$$

Pode-se então encontrar uma expressão para K através de uma função *Kernel* $H(\mathbf{u})$, sendo esta um hipercubo unitário centrado na origem. Esta expressão é dada por:

$$\mathbf{K} = \sum_{n=1}^N \mathbf{H}\left(\frac{\mathbf{x} - \mathbf{x}^n}{h}\right) \quad (\text{B.12})$$

tal que a quantidade $\mathbf{H}((\mathbf{x} - \mathbf{x}^n)/h)$ é igual a 1 se o ponto \mathbf{x}^n cai dentro de um hipercubo de lado h e centrado em \mathbf{x} , do contrário é zero. Logo, a Equação B.12 fornece o número de exemplares que cai dentro deste hipercubo. Substituindo (B.11) e (B.12) em (B.10), obtém-se finalmente a estimativa da função de densidade, mostrada abaixo

$$p(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \frac{1}{h^d} \mathbf{H}\left(\frac{\mathbf{x} - \mathbf{x}^n}{h}\right). \quad (\text{B.13})$$

As referências, [DUD73] e [BIS95], mostram que h atua como um parâmetro de suavização e deve ser bem escolhido para se obter uma boa aproximação do valor estimado de $p(\mathbf{x})$ em relação à densidade verdadeira.

B.3.2 Método dos K-vizinhos-mais-próximos:

Retornando à Equação B.10, a técnica dos K-vizinhos mais próximos (K-NN) estipula um valor fixo para K e permite que o volume V possa variar. Tem-se agora uma hiperesfera com centro em \mathbf{x} , tal que o raio desta possa crescer até conter todos os K pontos de dados. Neste caso, o parâmetro K é o parâmetro de suavização e deve ser encontrado um valor ótimo para este. Os mesmos problemas detectados para a utilização da função *Kernel* são também observados aqui. Ambas as técnicas apresentam a desvantagem de terem os pontos de dados de treinamento retidos, ocasionando problemas de armazenamento e podem requerer alto esforço computacional para se testar novos valores.

Como uma das aplicações da estimação de densidade é a construção de classificadores baseados no teorema de Bayes, um modelo de classificador pode ser obtido diretamente da técnica K-NN, introduzindo uma pequena modificação. Suponha que o conjunto de dados contenha N_k exemplares designando a classe C_k , lembrando que N é o total de exemplares. Uma hiperesfera com centro em \mathbf{x} pode ser feita de modo a conter os K pontos (ou exemplares) independente da classe. Se esta esfera de volume V contém K_k pontos da classe C_k , isto fornece uma aproximação para a densidade condicional de classe, tal que

$$p(\mathbf{x} | C_k) = \frac{K_k}{N_k V}. \quad (\text{B.14})$$

Se a Equação B.10 fornece a probabilidade incondicional e a probabilidade *a priori* pode ser aproximada por

$$P(C_k) = \frac{N_k}{N} \quad (\text{B.15})$$

então, usando o teorema de Bayes, tem-se que

$$P(C_k | \mathbf{x}) = \frac{p(\mathbf{x} | C_k)P(C_k)}{p(\mathbf{x})} = \frac{K_k}{K}. \quad (\text{B.16})$$

Logo, de (B.16) nota-se que, se a relação K_k/K for a maior possível, a probabilidade de erro de classificação de um novo vetor \mathbf{x} , designando uma classe C_k , é minimizada. Resumindo, a técnica K-NN pode ser desenvolvida através da escolha de uma hiperesfera em torno de \mathbf{x} , a qual contém os K pontos, independente das classes. Deve-se então classificar \mathbf{x} de acordo com a classe que tem maior número de pontos contidos nesta hiperesfera. Esta regra de classificação dos K-NN produz uma fronteira de decisão linear por partes, ou seja vários segmentos lineares. Cada um destes segmentos é perpendicular a um bissetor entre dois pontos de classes diferentes. Uma desvantagem de ambas as técnicas, função *Kernel* e K-NN, é que o custo computacional cresce com o aumento do conjunto de treinamento, uma vez que este deve ser sempre retido.

Além das técnicas paramétricas e não-paramétricas, existem métodos para estimação que utilizam conjuntamente as qualidades destas duas técnicas. Estes métodos não ficam restritos a uma forma funcional específica e nem ao tamanho do conjunto de treinamento. Tais métodos são conhecidos como *mixture densities* ou *mixture models*. Apesar destes serem eficientes para estimação, o esforço computacional é mais intenso que nas técnicas anteriores.

B.4 FUNÇÕES DISCRIMINANTES

Uma outra maneira de se fazer uma classificação é através de um conjunto de *funções discriminantes* $g_k(\mathbf{x})$ ($k=1, \dots, c$ classes), ou seja, um exemplar \mathbf{x} indica a classe C_k se

$$g_k(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{para todo } j \neq k. \quad (\text{B.17})$$

Então, este classificador é visto como uma máquina que computa c funções discriminantes e seleciona a categoria correspondente ao maior discriminante [DUD73]. O classificador de Bayes é facilmente representado desta forma, onde $g_k(\mathbf{x}) = P(C_k|\mathbf{x})$, tal que a maior função discriminante corresponde a maior probabilidade *a posteriori*. Lembrando que a probabilidade incondicional não depende das classes do problema e, portanto, não afeta a decisão de classificação, a função discriminante $g_k(\mathbf{x})$ pode ser dada por:

$$g_k(\mathbf{x}) = p(\mathbf{x}|C_k)P(C_k). \quad (\text{B.18})$$

Desde que, somente a magnitude relativa das funções discriminantes é o importante para classificar uma dada classe, a aplicação de um fator de multiplicação, *bias* ou função monotônica qualquer ($f(g_k(\mathbf{x}))$) não afeta o classificador. Supondo aplicado à Equação B.18 uma função logarítmica, tem-se

$$g_k(\mathbf{x}) = \ln(p(\mathbf{x}|C_k)) + \ln(P(C_k)) \quad (\text{B.19})$$

Se duas regiões \mathfrak{R}_k e \mathfrak{R}_j são contíguas, então a fronteira de decisão é dada por

$$g_k(\mathbf{x}) = g_j(\mathbf{x}) \quad (\text{B.20})$$

e num problema a duas classes, uma função discriminante simples é dada por

$$g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x}) \quad (\text{B.21})$$

onde, se $g(\mathbf{x}) > 0$, classifica-se C_1 , senão, se $g(\mathbf{x}) < 0$, classifica-se C_2 . A Equação B.21 também pode ser rescrita da seguinte forma:

$$g(\mathbf{x}) = \ln \frac{p(\mathbf{x} | C1)}{p(\mathbf{x} | C2)} + \ln \frac{P(C1)}{P(C2)}. \quad (\text{B.22})$$

De uma forma genérica, as funções discriminantes podem ser escritas como uma combinação linear dos componentes de \mathbf{x} . Estas podem ser expressas da seguinte forma:

$$g_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0} \quad (k=1, \dots, c \text{ classes}) \quad (\text{B.23})$$

e o problema se torna estimar os parâmetros \mathbf{w}_k e w_{k0} , respectivamente o vetor peso e *bias*. Em [DUD73] é mostrado um exemplo da obtenção destes parâmetros quando a função de densidade de probabilidade é uma função *normal*. Isto é feito para três situações: na primeira, as variáveis são estatisticamente independentes e apresentam a mesma variância σ^2 ; na segunda, as matrizes de covariância Σ são idênticas para todas as classes e na terceira, as matrizes de covariância para cada classe são diferentes (caso genérico). Dois classificadores são obtidos da primeira e segunda situações, sob certas considerações. Estes são os classificadores *Euclidiano* e de *Mahalanobis* [SCH97].

Se uma regra de decisão é arbitrada, como a dada pela Equação B.17, o espaço de \mathbf{x} é dividido em c regiões de decisão. Este classificador pode ser chamado de *máquina linear* e a Equação B.17 pode ser rescrita da seguinte forma:

$$(\mathbf{w}_k - \mathbf{w}_j)^T \mathbf{x} + (w_{k0} - w_{j0}) = 0 \quad (\text{B.24})$$

sendo que, a interpretação geométrica de (B.24) mostra que, se $g_k(\mathbf{x})=0$, então a fronteira de decisão é um hiperplano de dimensão $(d-1)$. No caso de $d=2$, a fronteira de decisão é uma reta. A diferença $(\mathbf{w}_k - \mathbf{w}_j)$ fornece um vetor saindo da origem e normal ao hiperplano. A diferença $(w_{k0} - w_{j0})$ fornece a posição deste hiperplano no espaço, em relação à origem. Assim, a distância da origem ao hiperplano pode ser dada por:

$$\mathbf{r} = - \frac{(w_{k0} - w_{j0})}{\|\mathbf{w}_k - \mathbf{w}_j\|}. \quad (\text{B.25})$$

Logo, (B.25) mostra que, se o valor da distância \mathbf{r} for negativa (lado positivo do hiperplano, $g_k > g_j$), classifica-se o novo exemplar como pertencente à classe C_k , caso contrário,

C_j . Observe que, se $(w_{k0} - w_{j0})$ for igual a zero, então o hiperplano passa pela origem.

A Equação B.23 pode ser reescrita da seguinte forma homogênea:

$$\mathbf{g}(\mathbf{x}) = \mathbf{a}^T \mathbf{y} \quad (\text{B.26})$$

onde

$$\mathbf{y} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \quad \text{e} \quad \mathbf{a} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix}.$$

A Equação B.26 representa o mapeamento de \mathbf{x} (d -dimensional) num novo espaço $(d+1)$ -dimensional. Lembrando que o objetivo é estimar os parâmetros \mathbf{w} e w_0 para se obter a melhor separação entre classes, então a utilização de (B.26) reduz o problema em encontrar somente o vetor peso \mathbf{a} . Logo, um exemplar y_k é corretamente classificado se $\mathbf{a}^T \mathbf{y}_k > 0$ para a classe C_k ($k = 1, \dots, c$ classes).

Num problema a duas classes, como mostrado na Figura B.2, se o vetor solução \mathbf{a} existe, este deve estar no lado positivo de qualquer hiperplano. Observe que \mathbf{y}_k é normal ao seu respectivo plano e o vetor solução encontra-se na intersecção dos dois semi-planos e qualquer vetor nesta região de solução é um vetor solução. Isto mostra que, se o vetor solução existe, ele não é único. Algumas técnicas são propostas com o objetivo de encontrar o vetor solução \mathbf{a} que satisfaça o conjunto de desigualdades lineares $\mathbf{a}^T \mathbf{y}_k > 0$. Isto pode ser feito encontrando-se uma função critério $\mathbf{J}(\mathbf{a})$ que deve ser minimizada se \mathbf{a} é o vetor solução.

As mais comuns destas técnicas minimizando $\mathbf{J}(\mathbf{a})$ são a *Função Critério Perceptron* (FCP) e a *Função Critério da Soma dos Erros Quadráticos* (FCSSE). Estas duas técnicas utilizam procedimentos de *Gradiente Descendente* para encontrar o vetor \mathbf{a} .

A FCP baseia-se numa função critério que minimiza os exemplares classificados erradamente pelo vetor \mathbf{a} . Em [DUD73] é mostrado o Teorema de Convergência do Perceptron, onde é obtido um número k_0 de iterações, tal que o vetor peso resultante possibilita classificar adequadamente todos os exemplares. Se os dados não forem linearmente separáveis, k_0 tende ao infinito e o algoritmo nunca termina. Se for arbitrariamente parado, não há garantias que \mathbf{a} proporcione a melhor generalização para novos dados.

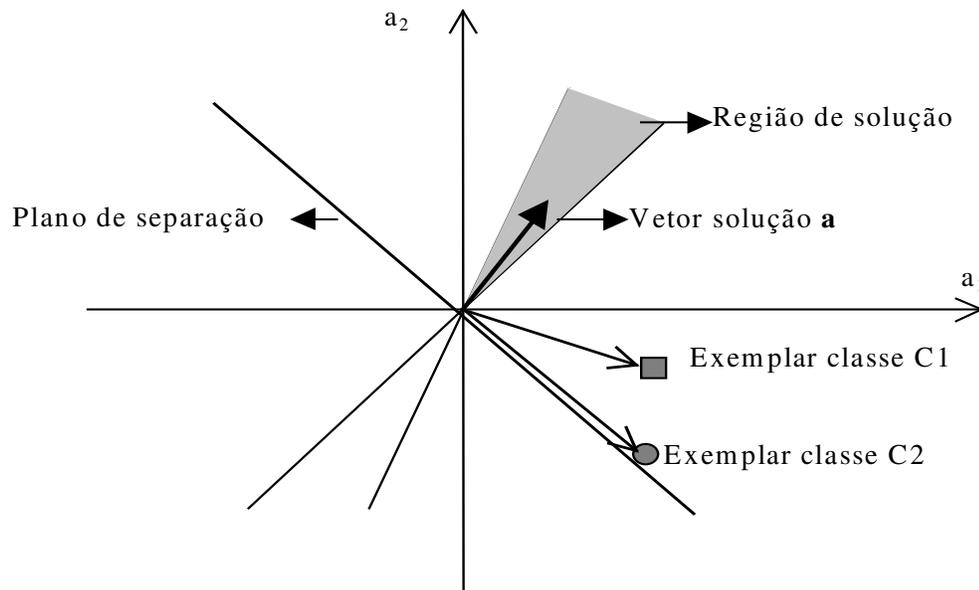


Figura B.2 - Região de solução no espaço dos pesos, mostrando a orientação do vetor solução 'a'.

Considere agora a FCSSE, dada pela Equação B.27, rescrita em função da formulação $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$, para a soma de todos os exemplares do conjunto de treinamento, tal que:

$$J(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^c \{g_k(x^n, \mathbf{w}) - t_k^n\}^2 \quad (3.27)$$

onde t_k^n representa o valor desejado (alvo) para a classe C_k (ou saída k) quando o vetor de entrada é \mathbf{x}^n . Basicamente, existem duas técnicas mais comumente utilizadas para minimizar (B.27). A primeira é denominada de *Solução da Pseudoinversa* ([HAS95],[BIS05]) e a segunda de *Procedimento de Widrow-Hoff ou LMS* [HAS95]. Esta segunda técnica tem a vantagem de evitar o problema de singularidade da matriz pseudoinversa. Em [DUD73] é mostrado que a solução para um erro médio quadrático mínimo aproxima-se assintoticamente da função discriminante de Bayes (Equação B.19).

Uma técnica conhecida envolvendo uma análise de funções discriminantes é o *Discriminante Linear de Fisher*. Esta técnica é utilizada para a redução da dimensionalidade do vetor de entrada. Se este é d -dimensional e há c classes do problema, então a projeção deste espaço é feita no espaço $(c-1)$ -dimensional, assumindo que $d \geq c$. Pode-se colocar os componentes de \mathbf{x} na forma de uma combinação linear, obtendo-se então $c-1$ funções

discriminantes, tal que

$$y_k = w_k^T \mathbf{x} \quad k = 1, \dots, c-1 \quad (\text{B.28})$$

ou sob a forma matricial

$$\mathbf{y} = \mathbf{W}^T \mathbf{x} \quad (\text{B.29})$$

onde \mathbf{y} é um vetor $(c-1)$ -dimensional e \mathbf{W} é a matriz peso com dimensão $(d \times (c-1))$. O maior problema reduzindo a dimensionalidade é uma grande perda de informação e classes que são bem separadas no espaço original podem apresentar uma considerável superposição na dimensão reduzida. Então, o objetivo da técnica do Discriminante Linear de Fisher é encontrar um ajuste dos componentes de \mathbf{W} que maximize a separação entre classes. Isto é feito através da obtenção de uma *função critério* [DUD73]. Esta mesma referência mostra que, para um problema a duas classes, o resultado obtido pela técnica FCSSE é igual ao obtido pelo Discriminante Linear de Fisher.

B.5 AJUSTE DE CURVAS POR POLINÔMIOS

Uma técnica bastante conhecida quando for necessário lidar com problemas de regressão é o ajuste de curvas por polinômios. A idéia principal é obter uma função matemática característica que mais se aproxima da função intrínseca presente em um conjunto de dados. Seja a seguinte função polinomial abaixo:

$$y(x) = w_0 + w_1 x + \dots + w_M x^M = \sum_{j=0}^M w_j x^j \quad (\text{B.30})$$

onde w são os parâmetros, y é a função e x é uma variável. Os parâmetros podem ser representados por $\mathbf{w} = \{w_0, \dots, w_M\}$. Observe que estes parâmetros são semelhantes aos pesos das funções discriminantes e das redes neurais (Item 3.3.5). A Equação B.30 pode ser vista como um mapeamento não-linear que tem x como entrada e y como saída. Logo, o polinômio pode ser escrito como uma um mapeamento funcional $y = y(x, \mathbf{w})$. Se existem N pontos (exemplares) em

um conjunto de dados, então x^n é um destes pontos ($n = 1, \dots, N$) e t^n é o valor desejado para a saída y . O objetivo então é encontrar os valores adequados para os parâmetros, tal que a função polinomial represente mais fielmente a função implícita nos dados. Se o erro entre a saída do polinômio e o valor desejado for zero ou o mínimo possível, então espera-se que o valor de \mathbf{w} seja ótimo. Para isto, pode ser utilizada a expressão para a soma dos erros quadráticos

$$E = \frac{1}{2} \sum_{n=1}^N \{y(x^n, \mathbf{w}) - t^n\}^2 \quad (\text{B.31})$$

em que pode-se aplicar uma técnica para minimizar E em função dos parâmetros \mathbf{w} . Observe que E é uma função quadrática e, a obtenção do mínimo desta produz um conjunto de equações algébricas lineares. Se os valores desejados t^n são conhecidos, o aprendizado é dito supervisionado, como já mencionado anteriormente e não-supervisionado se este não é fornecido. Neste caso, a meta pode ser modelar a distribuição de probabilidade dos dados de entrada ou agrupamentos ou descobrir agrupamentos ou outras estruturas.

Um problema que ocorre em problemas de regressão é o *overfitting*. No exemplo da utilização da função polinomial $y(x)$ (Equação 3.30) para ajustar uma curva $h(x)$ se M (número de graus de liberdade ou parâmetros livres) é igual a 1, pode o resultado do ajustamento fornecer uma representação pobre de $h(x)$ em função da falta de flexibilidade do modelo polinomial para $M=1$. Isto pode ser resolvido pelo aumento do grau de liberdade da função. Entretanto, se este crescer muito, pode haver um ajuste ótimo para os dados do conjunto de treinamento (pelo aumento significativo do grau do polinômio) com uma pobre representação da função $h(x)$.

Neste caso, este ajuste ótimo dos dados é feito pela geração de fortes oscilações. A consequência disto é uma resposta muito ruim quando novos dados são testados, ou seja, baixa capacidade de generalização. Então o melhor desempenho quanto a capacidade de generalização é encontrar um modelo cuja complexidade (no caso do polinômio, dado pelo grau deste) que não seja nem pequena nem grande. Em problemas de classificação, *overfitting* também é observado. A medida que o modelo cresce em complexidade, as fronteiras de decisão tornam-se também muito complexas, respondendo bem aos dados de treinamento, mas perdendo em capacidade em classificar novos dados.

A Equação B.30 mostra um mapeamento de uma saída $y(x)$ em relação à uma variável de entrada x . Entretanto, este modelo pode ser estendido para um problema multivariável, onde agora a uma dada saída está relacionada com o vetor de entrada \mathbf{x} d -dimensional. Em [BIS95] é

mostrado que a medida que d aumenta, o número de graus de liberdade do polinômio aumenta exponencialmente (d^M). Conseqüentemente cresce também o requerimento de exemplares pela maldição da dimensionalidade.

B.6 OBSERVAÇÕES IMPORTANTES

Conforme apresentado no Item 3.3, os principais métodos aplicados a problemas de classificação são estatísticos, neurais e aprendizado de máquina. Dois aspectos devem ser abordados: a comparação das técnicas enquanto classificadores (estimação precisa) apenas e a capacidade da técnica em expressar o conhecimento de forma interpretável aos seres humanos, ou seja, de forma simbólica (formalismo da interpretação). Em sistemas baseados em conhecimento, além da definição de um classificador ótimo para ser aplicado ao processo de aquisição em si, a técnica tem que ser completa o suficiente para fornecer este conhecimento simbólico, formando a base de conhecimento mostrada na Figura 3.1. Alguns pontos importantes podem ser destacados a seguir em relação às técnicas utilizadas em processos de classificação.

Em reconhecimento de padrões destaca-se o Classificador de Bayes como um dos mais intuitivos e tradicionais métodos de aprendizado. Em [SCH97] é argumentado que, sob a consideração da probabilidade a priori igual a $1/c$, onde c é o número de classes do problema, o resultado encontrado para a probabilidade a posteriori é considerado o melhor resultado possível para uma técnica de classificação. A questão central da aplicação deste classificador é que a função densidade de probabilidade condicional de classes, $P(\mathbf{x}|C_k)$, é muito difícil de ser estimada. Para esta tarefa, basicamente, são utilizadas técnicas paramétricas e não-paramétricas. A primeira supõe que as formas funcionais da distribuição de probabilidade são conhecidas e a segunda não, mas supõe que estas possam ser estimadas dos dados. Uma das técnicas não-paramétricas mais utilizadas é a K-NN. Entretanto, esta apresenta o inconveniente de ter que estimar o valor ótimo do número de exemplares (K) que define a região do espaço amostral a ser analisado para estimar $P(\mathbf{x}|C_k)$, pois assim como outros métodos, há a possibilidade da ocorrência *overfitting* dependendo do valor de K estipulado. Outra questão mostra que este método é sensível ao aumento da dimensionalidade do vetor de entrada, ocasionando degradação nos resultados [WEH94].

Métodos baseados em ajuste de curvas por polinômios apresentam a desvantagem de serem dependentes de se encontrar um valor adequado para o grau do polinômio para evitar o

overfitting. Além disso, o requerimento de exemplares cresce exponencialmente de acordo com o aumento do grau do polinômio (complexidade do modelo). Este fato não ocorre com as redes neurais, onde o requerimento de exemplares cresce linearmente com o aumento do número de neurônios na camada interna. Entretanto, a dimensionalidade do vetor de variáveis de entrada ainda constitui um problema que afeta o desempenho dos classificadores, pois independente da complexidade do modelo, o requerimento de exemplares cresce também exponencialmente de acordo com a dimensão deste vetor. Ambos os modelos, neural e polinomial, podem ser vistos como aplicações específicas das funções discriminantes.

Apêndice C

REDES NEURAIS ARTIFICIAIS

C.1 INTRODUÇÃO

Este apêndice apresenta uma revisão teórica sobre as Redes Neurais Artificiais sob o enfoque do treinamento supervisionado. São mostrados os algoritmos para as redes neurais camada simples com treinamento por Regra de Hebb, redes neurais *feedforward* multicamadas com Regra Retropropagação de Erros e o algoritmo de treinamento OET2.

C.2 CONCEITOS BÁSICOS - O NEURÔNIO ARTIFICIAL

Baseado na forma de processamento do pensamento humano, as rede neurais artificiais são algoritmos computacionais que procuram assimilar o padrão de conhecimento obtido de um conjunto de exemplares que contém a informação sobre um determinado problema. A partir da assimilação deste padrão, estas deverão ser capazes de generalizar, dentro de suas limitações específicas, informações pertencentes à mesma classe do problema.

A forma como o pensamento humano é processado deve-se as interconexões bastante distribuídas entre suas unidades básicas de processamento, os neurônios. Estes, recebem a informação através dos dendritos (entrada), processa-a no corpo da célula nervosa e passa a informação resultante através do axônio (saída). A conexão do neurônio com outras unidades é feita através de conexões denominadas de sinapses, cuja ‘força’ determina a transmissão do sinal e estabelece um padrão para cada informação. Guardando-se a similaridade, a unidade básica das

RNAs é o neurônio artificial. As diversas maneiras como são dispostos e interligados estes neurônios determinam a diferença entre as diversas formas de estruturas de redes neurais artificiais. Os algoritmos de aprendizagem também determinam as formas como os padrões são estabelecidos para cada tipo de rede, obtendo como resultado final uma RNA capaz de classificar novos padrões de uma mesma classe com erro mínimo admissível.

A figura C.1 abaixo, mostra o modelo esquemático do neurônio artificial. Comparando-o com o neurônio biológico, tem-se as entradas x_i ($i = 1,2,\dots,n$ entradas) representando os dendritos no modelo biológico, os respectivos pesos w_i , as sinapses, o somatório $\sum w_i x_i$ e a função de ativação ou de transferência $f(\sum w_i x_i)$, o corpo da célula e, as saídas y_i , o axônio.

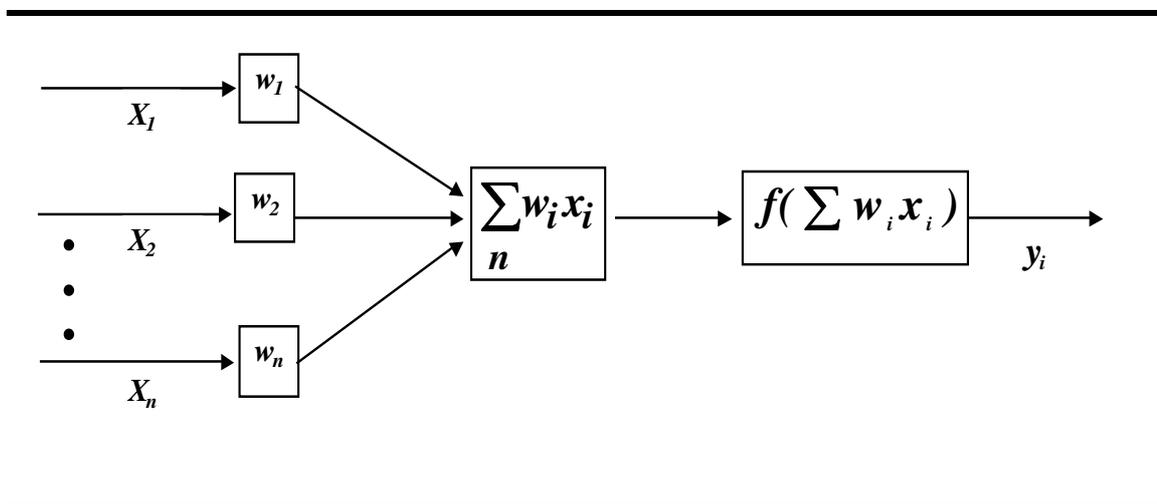
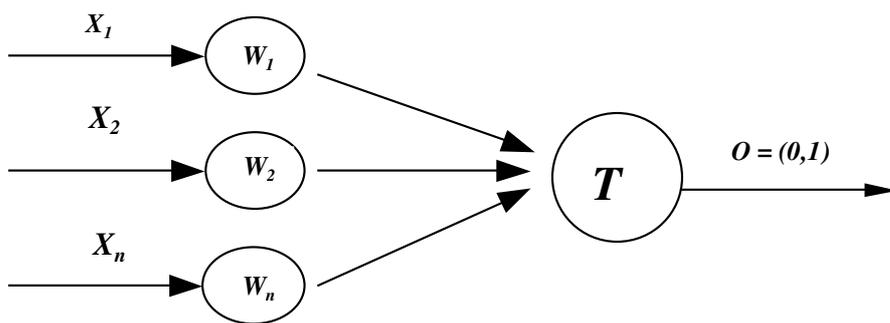


Figura C.1 - Neurônio Artificial

A saída y_i do modelo matemático dado pela figura C.1 é representado equação C.1 abaixo:

$$y_i = f\left(\sum_n w_i x_i\right) \tag{C.1}$$

A função de ativação ou de transferência tem o objetivo de ‘ativar’ o neurônio a partir de um determinado valor limite. Um exemplo de como isso é feito pode ser visto Figura C.2



Entrada	Saída
$\sum_{k=1}^n x_i w_i \leq T$	0
$\sum_{k=1}^n x_i w_i > T$	1

Figura C.2 - Modelo de neurônio com uma função de ativação do tipo *threshold*

Normalmente, a função de ativação é representada por funções não-lineares, diferenciáveis, contínuas e limitadas. Um exemplo típico deste tipo de função é a Função Sigmóide, mostrada por (C.2)

$$y(t) = (1 + e^{-\lambda x})^{-1} \quad 0 < y(t) < 1 \quad (C.2)$$

onde λ fornece o grau de achatamento da curva, como pode ser observado na figura C.3.

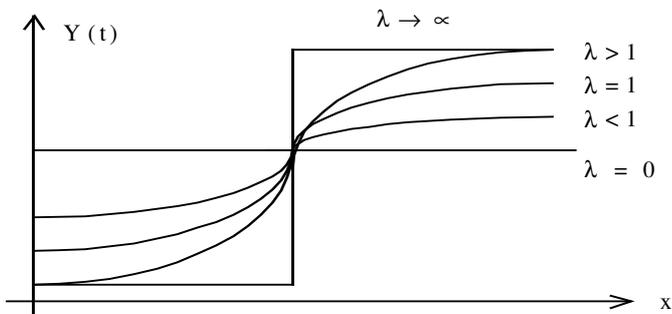


Figura C.3- - Representação gráfica da Função Sigmóide

A obtenção do padrão da informação processada pelo neurônio é feita pelo ajuste dos pesos w_i durante o processo de aprendizagem e que estabelece uma relação entre a entrada e a saída. A estrutura de uma RNA é determinada pela conectividade com que seus neurônios são dispostos, podendo estes ser interconectados localmente em forma de aglomerados ou organizados em grupos ou camadas.

C.3 ESTRATÉGIA DE APRENDIZAGEM SUPERVISIONADA

O processo de aprendizagem de uma RNA pode ser feito de duas maneiras: supervisionado ou não supervisionado. O primeiro, para cada sinal de entrada, é apresentado ao sistema uma saída desejada e a rede configura, ela mesma, a melhor relação entrada/saída. Para o treinamento não supervisionado, somente existe o sinal de entrada e a rede organiza-se internamente tal que, cada neurônio interno responde diferentemente (forte ou fraco) para um dado conjunto de sinais de entrada. Reportando-se ao objetivo do trabalho, são apresentados apenas os aspectos inerentes ao aprendizado supervisionado. Existem três características de treinamento supervisionado [HAS95]:

- REGRA DE HEBB: Os pesos só serão incrementados se existir uma entrada alta para uma saída alta.
- COMPETITIVO: Os neurônios competem entre eles. O vencedor é o que responde mais fortemente a um sinal de entrada, modificando-se para tornar-se o mais semelhante com a entrada.
- REGRA DELTA: Baseado na redução do erro entre a atual saída e a saída desejada pela modificação dos pesos das conexões

Tornando esta exposição mais direcionada aos tipos de RNAs utilizadas no trabalho, serão apresentados os aspectos teóricos para a Regra de Hebb aplicada a uma RNA com uma camada (RNA-SL) e a regra de treinamento Retropropagação de Erros aplicada a uma RNA do tipo *feedforward* multicamadas (RNA-ML).

C.3.1 RNA com uma camada treinada com a Regra de Hebb

A premissa de um treinamento utilizando a Regra de Hebb é que, quanto mais relacionada estiver uma entrada de uma saída desejada, mais forte é a conexão entre elas. A Equação C.3, mostrando a variação no peso w_{ij} da conexão entre uma entrada e uma saída, reflete este modelo:

$$\Delta w_{ij} = \varepsilon x_i d_j \quad (\text{C.3})$$

onde ε é a taxa de aprendizagem, x_i é a entrada i ($i=1,2,\dots,n$ entradas) e d_j é a saída j desejada ($j=1,2,\dots,k$ saídas). A Equação C.3 reflete a força da conexão entre elas à medida que os valores de uma entrada aproximam-se dos de uma saída,. Considere o peso inicial da conexão igual a zero e m o número de padrões de treinamento, então o peso da conexão é calculado pela expressão

$$w_{ij} = \varepsilon \sum_{p=1}^m x_{pi} d_{pj} \quad (\text{C.4})$$

Como este tipo de rede consiste somente de neurônios lineares, então a ativação de uma saída é apenas a soma ponderada de todas as entradas em relação a uma dada saída j mais o termo *bias*. Então tem-se que:

$$y_j = \sum_{i=1}^n x_i w_{ij} + \textit{bias} \quad (\text{C.5})$$

Este aprendizado linear pode produzir aprendizado associativo interessante por causa desta forte correlação obtida. Entretanto, isto pode não ser suficiente para aprender padrões mais complexos. Outra característica é que esta não consegue lidar com dados aleatórios ou ruídos. Isto determina que o desempenho deste tipo de RNA utilizando a Regra de Hebb é extremamente dependente da similaridade dos padrões de entrada. Outra desvantagem é que esta regra de aprendizado só consegue lidar bem com dados linearmente independentes, quando é necessário exigir capacidade de generalização desta RNA.

C.3.2 RNA *feedforward* multicamadas com retropropagação do erro

Para entender o processo de treinamento de uma RNA-ML é importante entender o mesmo aplicado a apenas um neurônio. Conhecida como Regra de Aprendizado μ -LMS, esta regra é considerada de treinamento simples e é muito utilizada. Esta é importante, pois tem uma aplicação no aprendizado em redes de múltiplas unidades neurais. Seja a seguinte função objetivo abaixo:

$$J(w) = \frac{1}{2} \sum_{i=1}^m (d^i - y^i)^2 \quad (\text{C.6})$$

onde d é o valor desejado, y é a saída do neurônio, w é o peso do respectivo neurônio e m o número de padrões do conjunto de treinamento. Vale salientar que esta função é a soma dos erros quadráticos (SSE - *Sum of Squared Error*). Neste caso a saída y^i do neurônio para o padrão i é uma função linear dada por:

$$y^i = (x^i)^T w \quad (\text{C.7})$$

Usando gradiente descendente para minimizar $J(w)$ na equação C.6 obtém-se

$$\begin{aligned} w^{k+1} &= w^k - \mu \Delta J(w) \\ &= w^k + \mu \sum_{i=1}^m (d^i - y^i) x^i \end{aligned} \quad (\text{C.8})$$

Assim, $J(w)$ define uma superfície hiperparaboloidal convexa [HAS95] com mínimo global no peso w^* . Portanto, se μ (taxa de aprendizagem) for escolhido muito pequeno, a pesquisa em gradiente descendente irá assintoticamente convergir à solução w^* indiferente do ajuste do ponto inicial w^l .

Uma extensão a esta regra, também chamada de Regra Delta, é mostrada a seguir. A função de ativação utilizada agora é não-linear e diferenciável. A aplicação desta regra de aprendizagem é utilizada para treinamentos de RNAs tipo *feedforward*, sendo muito utilizada

nas mais variadas aplicações teóricas e práticas.

Considere a seguinte estrutura neural:

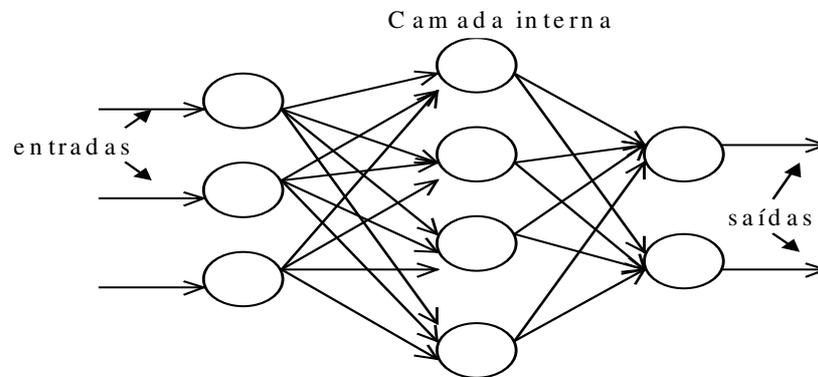


Figura C.4 - Estrutura Neural tipo Multicamadas *Feedforward*

Pode-se observar que os neurônios estão dispostos em camadas. Os neurônios pertencentes à mesma camada não se comunicam. O fluxo de informação flui da entrada para a saída em um único sentido (*feedforward*). Basicamente, estruturas deste tipo possuem três características de camada: a camada de entrada, a(s) camada(s) interna(s) e a camada de saída.

Como já mencionado, uma estratégia de treinamento supervisionado baseado em gradiente descendente, Regra Delta, é aplicada a esta estrutura para minimização do erro de saída. A cada passo, este erro é retropropagado pela RNA, atualizando dos pesos das conexões. A Figura C.5 abaixo mostra o modelo generalizado de um dos neurônios artificiais presentes na estrutura mostrada na figura C.4, onde para cada vetor entrada x_k contendo n elementos, tem-se uma saída desejada d_k , onde $k = 1, 2, \dots, m$ padrões do conjunto de treinamento. O índice j representa o j -enésimo neurônio da estrutura. Assim, para o padrão k de entrada, reescrevendo (C.6) tem-se:

$$E_k = \frac{1}{2} \sum_j (d_{jk} - y_{jk})^2 \tag{C.9}$$

e o incremental de erro dado por:

$$\Delta_k w_{ji} = \frac{\partial E_k}{\partial w_{ji}} = \mu (d_{jk} - y_{jk}) x_{ki} = \mu \delta_{jk} x_{ki} \tag{C.10}$$

A camada de entrada atua apenas como repetidora dos sinais de entrada, distribuindo-os nas entradas do neurônio de saída.

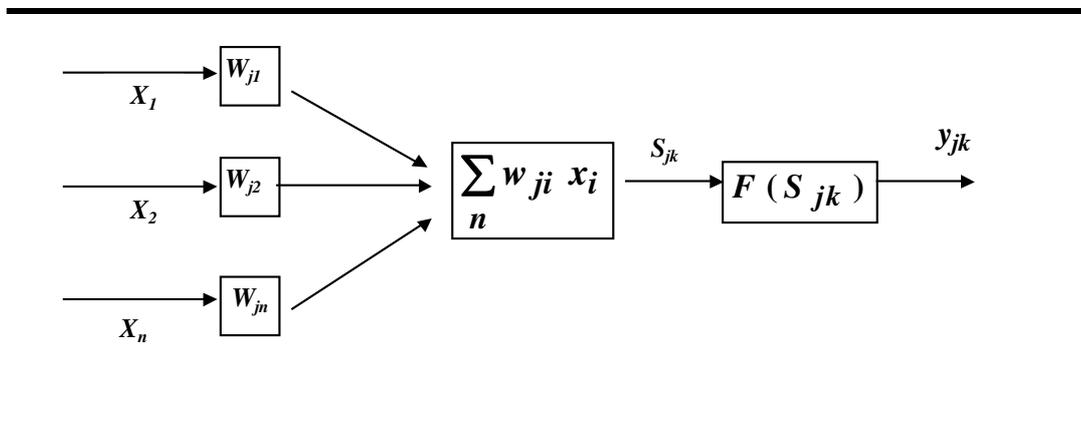


Figura C.5 - Neurônio Artificial

A Regra Delta Generalizada a ser mostrada abaixo atende a estruturas *feedforward* com camadas internas e faz uso de funções de ativação não-lineares. Sendo (C.9) o erro médio quadrático parcial, o erro médio quadrático global então pode ser dado por $E = \sum E_k$, calculado após a apresentação de todos os padrões utilizados no treinamento e é a função a ser minimizada pelo algoritmo de treinamento.

A relação genérica entre a saída da camada *i* e a próxima camada *j* pode ser dada por:

$$S_{jk} = \sum_i w_{ji} y_{ki} \tag{C.11}$$

Utilizando-se a função sigmóide, dada por (C.2), calcula-se então a saída, obtida pela expressão abaixo:

$$y_{kj} = f_j(S_{jk}) \tag{C.12}$$

onde f_j é a função sigmóide. Aplicando-se a regra da cadeia à Equação C.10 obtém-se a relação:

$$\frac{\partial E_k}{\partial w_{ji}} = \frac{\partial E_k}{\partial S_{kj}} \frac{\partial S_{kj}}{\partial w_{ji}} \tag{C.13}$$

Aplicando-se a Equação C.11 ao primeiro termo da C.13, definindo $\delta_{kj} = - \partial E_k / \partial S_{kj}$ e substituindo-se em C.13 tem-se

$$- \frac{\partial E_k}{\partial w_{ji}} = \delta_{kj} y_{ki} \quad (\text{C.14})$$

e, para a implementação do gradiente em descida em E , a variação do peso é dada por

$$\Delta_k w_{ji} = \mu \delta_{kj} y_{ki} \quad (\text{C.15})$$

O objetivo é fazer a retropropagação dos δ s através da rede. Assim, redefinindo δ_{kj} , tem-se:

$$\delta_{kj} = \frac{\partial E_k}{\partial S_{kj}} = - \frac{\partial E_k}{\partial y_{kj}} \frac{\partial y_{kj}}{\partial S_{kj}} \quad (\text{C.16})$$

Da Equação C.12 tem-se que:

$$\frac{\partial y_{kj}}{\partial S_{kj}} = f_j'(S_{kj}) \quad (\text{C.17})$$

Para o cálculo do segundo fator, dois casos são obtidos. No primeiro assume-se que a unidade j é uma unidade de saída da rede, então da definição de E_k tem-se que

$$\frac{\partial E_k}{\partial y_{kj}} = -(t_{kj} - y_{kj}) \quad (\text{C.18})$$

Substituindo as Equações C.17 e C.18 em C.16, obtém-se

$$\delta_{kj} = (t_{kj} - y_{kj}) f_j'(S_{kj}) \quad (\text{C.19})$$

para qualquer unidade de saída u_j . Se u_j não é unidade de saída, tem então

$$\sum_h \frac{\partial E_k}{\partial S_{kh}} \frac{\partial E_{kh}}{\partial y_{kj}} = \sum_h \frac{\partial E_k}{\partial S_{kh}} \frac{\partial}{\partial y_{kj}} \sum_i w_{hi} y_{ki} =$$

$$\sum_h \frac{\partial E_k}{\partial S_{kh}} w_{hj} = - \sum_h d_{kh} w_{hj} \quad (\text{C.20})$$

onde h representa o somatório sob os neurônios da camada posterior à camada do neurônio j . Substituindo as Equações C.20 e C.17 em C.16, obtém-se

$$\delta_{kj} = f_j'(S_{kj}) \sum_h \delta_{kh} w_{hj} \quad (\text{C.21})$$

Através das Equações C.15, C.19 e C.21, a regra delta generalizada fornece o procedimento computacional necessário para o ajuste de todos os pesos da rede *feedforward* multicamadas. Pelo fato da rede possuir camadas internas, pode-se ter vários mínimos locais, comprometendo o ajuste ótimo dos pesos, pois pode-se encaixar em um mínimo local, longe de um mínimo global.

Algoritmo de implementação

1. Inicializar pesos e *offsets* com valores aleatórios pequenos, próximos a zero
2. Propagar um padrão de entrada pela rede
3. Comparar o sinal de saída obtido com o valor desejado
4. Calcular o retropropagar a medida de erro (começando pela camada de saída) através da rede
5. Minimizar o erro, ajustando os pesos das conexões através da Equação C.22:

$$w_{ji}(K+1) = w_{ji}(K) + \Delta_k w_{ji}(K) \quad (\text{C.22})$$

onde $\Delta_k w_{ji} = \mu \delta_{kj} y_{ki}$, e o índice K refere-se a iteração corrente. Nesta equação deve ser observado que, se a unidade j é uma unidade de saída, então a Equação C.19 deve ser usada no cálculo de δ_{kj} . De outro modo, se a unidade j é uma unidade interna, então, a equação C.21 deve ser usada. Os *offsets* podem ser encarados como pesos e

ajustados de maneira similar. Um termo de momento pode ser adicionado na equação C.22 com o objetivo de aumentar a velocidade de convergência da aprendizagem. O termo é colocado da seguinte forma:

$$w_{ji}(K+1) = w_{ji}(K) + \Delta_k w_{ji}(k) + \alpha \Delta_k w_{ji}(K-1) \quad (\text{C.23})$$

onde α é basicamente uma constante, que determina o efeito da variação anterior na variação atual

6. Apresentar o próximo padrão de entrada/saída e retornar ao passo 2. Os padrões de entrada e saída são apresentados de forma cíclica e contínua, até que a saída acompanhe os resultados esperados. A rede será capaz de generalizar a aprendizagem adquirida se o conjunto de padrões de entrada e saída for adequadamente escolhido. Deve-se salientar que o sucesso total da aprendizagem nem sempre é garantido. Isto devido a convergência poder tomar tempo e pode-se cair em um mínimo local.

C.3.3 Rede Neural Feedforward com treinamento OET2

Embora muitos melhoramentos tenham sido implementados na regra delta generalizada com retropropagação de erros, nos dias de hoje ainda existe a necessidade métodos de treinamento de RNAs que sejam mais confiáveis e eficientes. O principal problema é que a função erro é altamente não linear. Visando caminhar na solução deste problema, foi proposto em [referência Tese Alex] um novo método para treinamento de RNAs, que utiliza sucessivas aproximações quadráticas para a função erro até um ponto muito próximo do mínimo global.

Um mapeamento não linear que associa padrões de treinamento de entrada com os de saída é encontrado através de sucessivas soluções de sistemas lineares usando métodos diretos. Esta metodologia, na realidade, constitui um procedimento baseado em métodos de mínimos quadrados não lineares (aproximação local da função erro por uma função quadrática e a minimização exata desta por uma função aproximada).

Algoritmo de treinamento OET2

O procedimento para o treinamento de uma rede com uma camada oculta apenas (Figura C.6) é descrito a seguir. A seguinte notação é usada durante a descrição do OET2:

- r - número de entradas/saídas no conjunto de treinamento;
- n - número de neurônios de saída;
- $m+1$ - número de neurônios de entrada (o qual é o mesmo número de neurônios na camada oculta)
- H_{d-out} - r por n matriz de saída desejada;
- H_{in} - r por m+1 matriz de entrada;
- Z_{d-out} - H_{d-out} retropropagada através da camada de saída (matriz r por n); a i-ésima coluna de Z_{d-out} é denotada por \underline{z}_{d-out}^i e a j-ésima coluna de Z_{d-out}^i é denotada por \underline{z}_{d-out}^j ;
- W_{out} - matriz m+1 por n dos pesos das conexões entre a camada oculta e camada de saída; a i-ésimo coluna de W_{out} é denotada por \underline{w}_{out}^i ;
- H_{d-hid} - matriz r por m+1; a j-ésima coluna de H_{d-hid} é denotada por \underline{h}_{d-hid}^j ;
- Z_{d-hid} - H_{d-hid} retropropagada através da camada oculta (matriz r por m+1); a k-ésima coluna de Z_{d-hid} é denotada por \underline{z}_{d-hid}^k
- W_{hid} - matriz m+1 por m+1 do pesos das conexões entre a camada de entrada e a camada oculta; a k-ésima coluna de W_{hid} é denotado por \underline{w}_{hid}^k
- Z_{hid} - produzida pela multiplicação de H_{in} e W_{hid} (matriz r por m+1);
- H_{hid} - Z_{hid} propagatada através da camada oculta (matriz r by m+1);
- Z_{out} - produzida pela multiplicação de H_{hid} e W_{out} (matriz r por n);
- H_{out} - Z_{out} propagada através da camada de saída (matriz r por n);

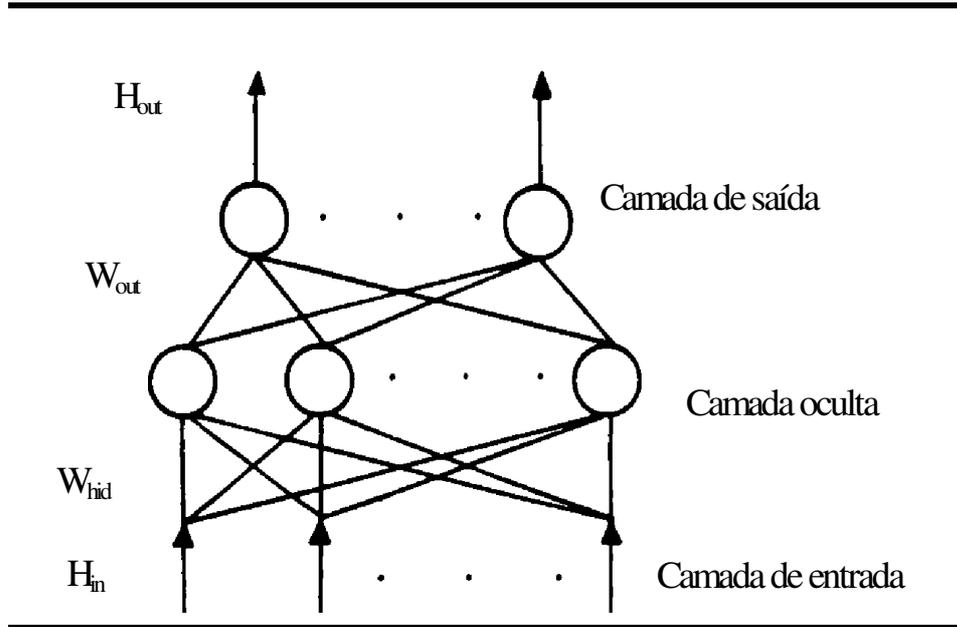


Figura C.6 – RNA com uma camada oculta.

Passos do Optimal Estimate Training 2:

1. Determine uma matriz de entrada H_{in} ($r \times m$), formada pelas m variáveis de entrada linearmente independente da matriz original H_{in} .
2. Calcular W_{out} inicial

$$H_{in} W_{out} = Z_{d-out} \quad (C.24)$$

$$\min_{\underline{w}_{out}^i} \left\| H_{in} \underline{w}_{out}^i - \underline{z}_{d-out}^i \right\|_2, \quad \text{para } i=1, 2, \dots, n \quad (C.25)$$

assumindo que r é maior que m .

3. Dado W_{out} , obter H_{d-hid} que produz Z_{d-out} ,

$$W_{out}^t H_{d-hid}^t = Z_{d-out}^t \quad (C.26)$$

Se $n \geq m$ então:

$$\min_{\underline{h}_{d-hid}^j} \left\| W_{out}^t \underline{h}_{d-hid}^j - \underline{z}_{d-out}^j \right\|_2, \quad \text{para } j=1, \dots, r \quad (C.27)$$

Se $n < m$ então:

$$\min \left\| \underline{h}_{d-hid}^j \right\|_2, \quad \text{s.t., } W_{out}^t \underline{h}_{d-hid}^j = \underline{z}_{d-out}^j \quad (C.28)$$

para $j=1, \dots, r$

4. Normalizar os elementos de H_{d-hid} para assegurar que eles estarão localizados dentro da faixa de saída da função de ativação dos neurônios da camada oculta. (tangente hiperbólica),

$$H_{d-hid} = H_{d-hid} \cdot 0.99 / |\lambda| \quad (C.29)$$

onde λ é o elemento de H_{d-hid} com maior magnitude.

5. Calcular Z_{d-hid} usando a tangente hiperbólica,
-

$$\mathbf{Z}_{d-hid} = \tanh^{-1}(\mathbf{H}_{d-hid}) \quad (\text{C.30})$$

6. Dados \mathbf{H}_{in} e \mathbf{Z}_{d-hid} , calcular \mathbf{W}_{hid} ,

$$\mathbf{H}_{in} \mathbf{W}_{hid} = \mathbf{Z}_{d-hid} \quad (\text{C.31})$$

$$\min_{\mathbf{w}_{hid}^k} \left\| \mathbf{H}_{in} \mathbf{w}_{hid}^k - \mathbf{z}_{d-hid}^k \right\|_2, \quad \text{s.t.} \quad \left| w_{hid,l}^k \right| \leq \text{Bound } 1, \quad (\text{C.32})$$

para $k=1, \dots, m$ e $l=1, \dots, m$

onde $w_{hid,l}^k$ é a componente de \mathbf{w}_{hid}^k .

7. Obter a saída intermediária atual para a camada oculta,

$$\mathbf{Z}_{hid} = \mathbf{H}_{in} \mathbf{W}_{hid} \quad (\text{C.33})$$

e

$$\mathbf{H}_{hid} = \tanh(\mathbf{Z}_{hid}) \quad (\text{C.34})$$

8. Recalcular \mathbf{W}_{out} ,

$$\mathbf{H}_{hid} \mathbf{W}_{out} = \mathbf{Z}_{d-out} \quad (\text{C.35})$$

$$\min_{\mathbf{w}_{out}^i} \left\| \mathbf{H}_{hid} \mathbf{w}_{out}^i - \mathbf{z}_{d-out}^i \right\|_2, \quad \text{s.t.}, \quad \left| w_{out,l}^i \right| \leq \text{Bound } 2, \quad (\text{C.36})$$

para $i=1, \dots, n$ and $l=1, \dots, m$

onde $w_{out,l}^i$ é a componente de \mathbf{w}_{out}^i .

Métodos numéricos para a solução de problemas de mínimos quadrados lineares para (C.25), (C.27), (C.28), (C.32) e (C.36) são descritos em [GOL91].

Apêndice D

BRAINNE

D.1 INTRODUÇÃO

Neste apêndice são apresentadas as etapas envolvidas no algoritmo de extração de regras de produção denominado de BRAINNE. O mecanismo utilizado é similar ao de uma árvore de decisão onde, em cada nó da árvore, um atributo é selecionado para cada classe em função de sua maior relevância no problema. As regras são extraídas a cada nó e testadas quanto a sua capacidade de classificação. O processo termina quando a taxa de classificação é maior possível, estabelecendo um conjunto de regras para cada classe do problema.

D.2 BRAINNE

O algoritmo BRAINNE [SES94] destaca-se como um agente inteligente que proporciona uma metodologia para aquisição automática de conhecimento. Este algoritmo é basicamente dividido em duas etapas:

- 1ª. Extração de conhecimento que se encontra sintetizado em uma base de dados
- 2ª. Colocação do conhecimento em formato compreensível aos seres humanos

A primeira etapa utiliza-se das Redes Neurais Artificiais (RNAs) como o dispositivo que infere na base de dados (aprendizado indutivo) para extrair as informações relevantes do problema. A segunda etapa proporciona que estas informações sejam colocadas em um formato simbólico, ou seja, os resultados obtidos são regras de produção. Estas regras são produzidas

utilizando-se de uma técnica similar a uma estrutura em árvore.

Para cada nó desta árvore, um atributo é selecionado através da obtenção de um grau de importância deste. Comparando-se com as árvores de decisão, isto equivale a determinar o *ganho* que um dado atributo proporciona ao processo de classificação.

Antes de apresentar o algoritmo, pode-se introduzir o conceito de extração de regras de produção de RNAs. Para isto é utilizada uma RNA sem camada intermediárias. A Figura D.1 mostra a estrutura topológica deste tipo de RNA, onde há m entradas (a_1, a_2, \dots, a_m) e n saídas (b_1, b_2, \dots, b_n). O primeiro passo é treinar esta RNA utilizando-se a Regra de Hebb como estratégia de treinamento (ver Apêndice C).

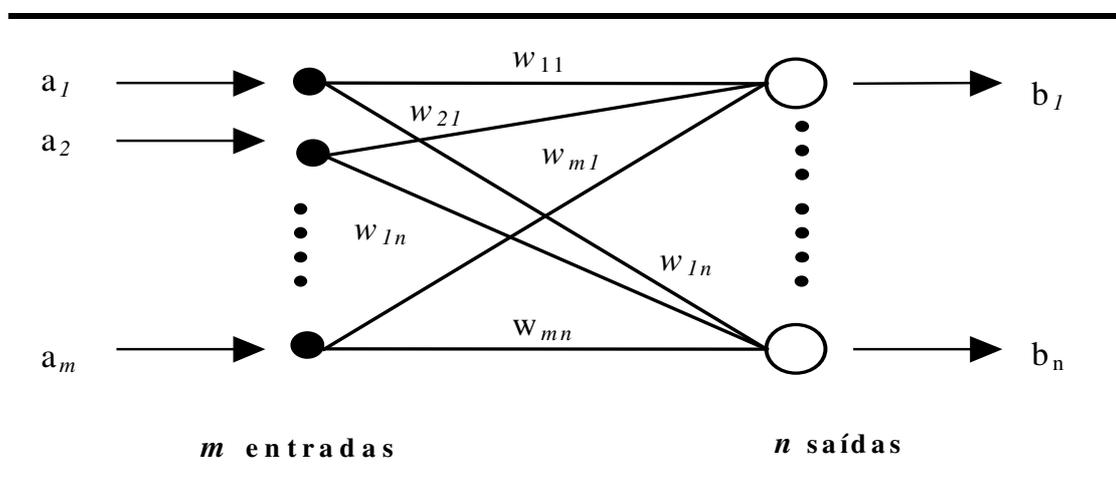


Figura D.1 - Rede Neural Artificial com camadas simples

Após a rede treinada, os pesos de todas as conexões entrada-saída são obtidos, formando a matriz de pesos. Cada peso pondera a contribuição que uma entrada fornece a uma dada saída. Considere uma dada saída b_k ($k = 1, 2, \dots, n$). Os pesos das conexões entre todas as m entradas e esta referida saída b_k são dados por $w_{1k}, w_{2k}, \dots, w_{mk}$. Pode-se então verificar o valor máximo entre estes pesos, formalmente $W_{\max} = \max(w_{1k}, w_{2k}, \dots, w_{mk})$.

Seja a seguinte regra de decisão: Se $w_{ik} \geq \beta W_{\max}$ ($i = 1, \dots, m$ entradas), então a entrada i é considerada fortemente relacionada com a saída b_k . Estas entradas são denominadas de *entradas contribuintes*. O valor limite (*threshold*) β tem valor no intervalo $(0;1]$. O tamanho da faixa determinada por β é influenciada por ruídos que podem estar presentes nos dados e é crucial para a correta geração das regras iniciais. Por exemplo, se $\beta = 0,95$, então todos os pesos maiores ou iguais a 95% do peso máximo (W_{\max}) são válidos e suas respectivas entradas são selecionadas como entradas contribuintes em relação a uma dada saída b_k .

As regras formadas podem ser visualizadas através do seguinte exemplo: uma base de

dados tem seu vetor de variáveis de entrada composto por A, B, C, D e saídas O1, O2 e O3. Após o treinamento e utilizando a regra de decisão mostrada no parágrafo anterior, as entradas contribuintes obtidas são A e B, para uma saída O1. Pode-se agora formar a *regra conjuntiva* em relação a esta saída da seguinte forma:

$$\text{SE (A E B) ENTÃO (O1)}$$

ou, representada por uma notação alternativa

$$(A \text{ E } B) \Rightarrow (O1) .$$

Na literatura, o lado esquerdo da regra é denominado de *antecedente* da regra e o lado direito é denominado de *conseqüência* da regra.

Neste ponto, é interessante definir o que é um *conceito*. De uma forma geral, conceito é uma coleção de atributos que descrevem uma idéia. Um conceito é dito de alto-nível quando representa generalizações e de baixo-nível quando representa especializações. Se poucos atributos definem uma saída ou idéia, então o conceito formado é de alto-nível. Por exemplo, no exemplo anterior, A e B podem definir um conceito mais geral (ou alto-nível) O1. Se β é alterado de modo a permitir que mais atributos sejam obtidos, à medida que estes vão sendo incorporados à regra conjuntiva, mais especializada esta se torna. Logo, uma regra especializada é composta dos conceitos de alto-nível, adicionada de novos atributos. Este conceito de alto-nível pode se repetir em qualquer outra regra. Isto proporciona que conceitos mais especializados se sobreponham a conceitos mais gerais. O exemplo a seguir mostra como são obtidos os conceitos.

Considere a regra do exemplo anterior, além de mais duas regras para as outras saídas, como mostrado abaixo:

$$\begin{aligned} R1: & (A \text{ AND } B) \Rightarrow O1 \\ R2: & (A \text{ AND } B \text{ AND } C) \Rightarrow O2 \\ R3: & (A \text{ AND } B \text{ AND } C \text{ AND } D) \Rightarrow O3. \end{aligned}$$

Destas regras, um conceito inicial pode ser obtido. Nota-se que a regra R1 apresenta menor número de atributos, portanto pode ser escolhida como o conceito inicial, tal que: Conceito 0: $(A \text{ AND } B) \Rightarrow O1$. Logo, o novo conjunto de regras fica:

$$\begin{aligned} R1: & (O1) \Rightarrow O1 \\ R2: & (O1 \text{ AND } C) \Rightarrow O2 \end{aligned}$$

$$R3: (O1 \text{ AND } C \text{ AND } D) \Rightarrow O3.$$

Observe que a regra R1 está reduzida ao próprio conceito. Um novo conceito é então definido, observando que R2 tem o menor número de atributos. Este é dado por: Conceito 1: $(O1 \text{ AND } C) \Rightarrow O2$, o qual forma o novo conjunto de regras abaixo:

$$\begin{aligned} R1: (O1) &\Rightarrow O1 \\ R2: (O2) &\Rightarrow O2 \\ R3: (O2 \text{ AND } D) &\Rightarrow O3. \end{aligned}$$

Pode ser visto que a cada novo passo, um novo conceito é formado. Isto determina uma hierarquia de conceitos, onde os conceitos mais baixos na hierarquia consistem de especializações de conceitos das camadas mais altas. No exemplo, o conceito 0 é hierarquicamente mais alto que o conceito 1. Se o último conjunto de regras formado fornece um grau máximo de especialização para o problema, então o processo pode ser parado.

Em muitos problemas do mundo real, uma saída pode ser definida por mais de uma entrada. Por exemplo, se A e B assinalam a saída O1, poderia existir uma nova combinação de variáveis, do tipo A e C que também assinalem a saída O1. Logo, existe mais de uma região no espaço de entrada que fornece a saída 1. Isto leva a obter uma regra que é uma composição de regras conjuntivas, tal que:

$$(A \text{ e } B) \text{ ou } (A \text{ e } C) \Rightarrow O1.$$

Este tipo de regra é denominada de *regra disjuntiva*. Observe que este tipo de regra possibilita que o conhecimento a respeito do problema seja melhorado, podendo levar à redução das taxas de erros de classificação. Então é desejável que uma técnica de extração de regras possibilite que tais regras sejam formadas.

Os problemas importantes enfrentados quando utiliza-se RNAs sem camadas intermediárias estão relacionados ao fato delas não trabalharem bem com dados não-lineares e em situações que exigem capacidade de generalização. Entretanto, podem extrair conhecimento suficiente para gerar regras simples. Outro problema encontrado é que elas geram apenas regras conjuntivas (uma única regra é obtida por saída), o que pode não ser o bastante em situações no mundo real.

O processo mostrado anteriormente para de extração de regras de produção de uma rede neural com uma camada mostra que é possível traduzir o conhecimento obtido por esta em

formato simbólico. Entretanto, quando se deseja extrair regras de uma RNA com múltiplas camadas, a tarefa não é tão simples como mostrado para RNA sem camadas intermediárias. A informação contida nos pesos das conexões das RNAs multicamadas é mais distribuída. Logo, obter a correlação adequada de uma entrada em relação a uma saída se torna uma tarefa mais difícil. A proposta de BRAINNE é justamente possibilitar que regras de produção sejam obtidas de uma RNA *feedforward* multicamadas, pois possibilitam lidar com problemas com maior grau de complexidade. O algoritmo BRAINNE pode ser dividido em quatro etapas, que são:

1. *PRODUTOS*: Obtém uma medida de correlação entre os atributos de entrada e cada saída do problema.
2. *SELEÇÃO DE ATRIBUTOS*: Estipula o critério de seleção dos atributos para compor os antecedentes da regras a serem formadas.
3. *PROCEDIMENTO GERAR-E-TESTAR*: Forma as regras que devem ser testadas. Os testes verificam a quantidade de exemplares classificados pelas regras e a unicidade destas. Isto estabelece um *critério de parada* para as regras encontradas como válidas.
4. *BASE DE CONHECIMENTO*: Coloca as regras de produção finais em formato adequado ao acesso de um mecanismo de inferência ou do operador.

A seguir são apresentadas as etapas que compõe o algoritmo, tanto para o caso em que dados discretos são utilizados, como para o caso da utilização de dados contínuos.

D.2.1 BRAINNE aplicado a dados discretos

O entendimento do algoritmo BRAINNE pode ser melhor visualizado através do processamento aplicado primeiramente a dados discretos, uma vez que, a estrutura do algoritmo não muda com a utilização de dados contínuos. As Etapas 1 a 4 acima descritas são então mostradas a seguir.

Etapa 1 - Produtos

Passo 1.1 - Treinamento da RNA-ML:

Considere uma RNA-ML com a seguinte configuração original: m entradas, n saídas e h neurônios na camada interna (Figura D.2a). Para se obter as entradas mais correlacionadas com uma dada saída, o algoritmo propõe estender o conjunto de entrada para incluir as saídas

desejadas (alvos), sendo estas consideradas agora entradas adicionais.

Como as entradas e saídas são colocadas no mesmo nível (i.e., no nível das entradas), elas podem então ser comparadas diretamente sem passar pela camada interna. Isto promove uma associação direta entre as entradas e as respectivas saídas. Este técnica de aprendizado corresponde em mapear o espaço das saídas no espaço das entradas.

Assim, a RNA-ML deve ser treinada com a nova configuração. Neste caso, a estratégia de aprendizado utilizada no treinamento é a retropropagação de erros (ver Apêndice C). Atendo-se apenas ao problema do treinamento, os parâmetros de ajuste deste tipo de rede neural, como taxa de aprendizado e momento, devem ser adequadamente escolhidos para garantir a qualidade deste treinamento. Vale ressaltar também que a base de dados deve sofrer um pré-processamento, fornecendo uma normalização entre 0 e 1 para cada variável do vetor de variáveis de entrada. Após treinada, a matriz de pesos obtida entre as entradas (inclusive as adicionais) e todos os neurônios da camada interna deve ser armazenada.

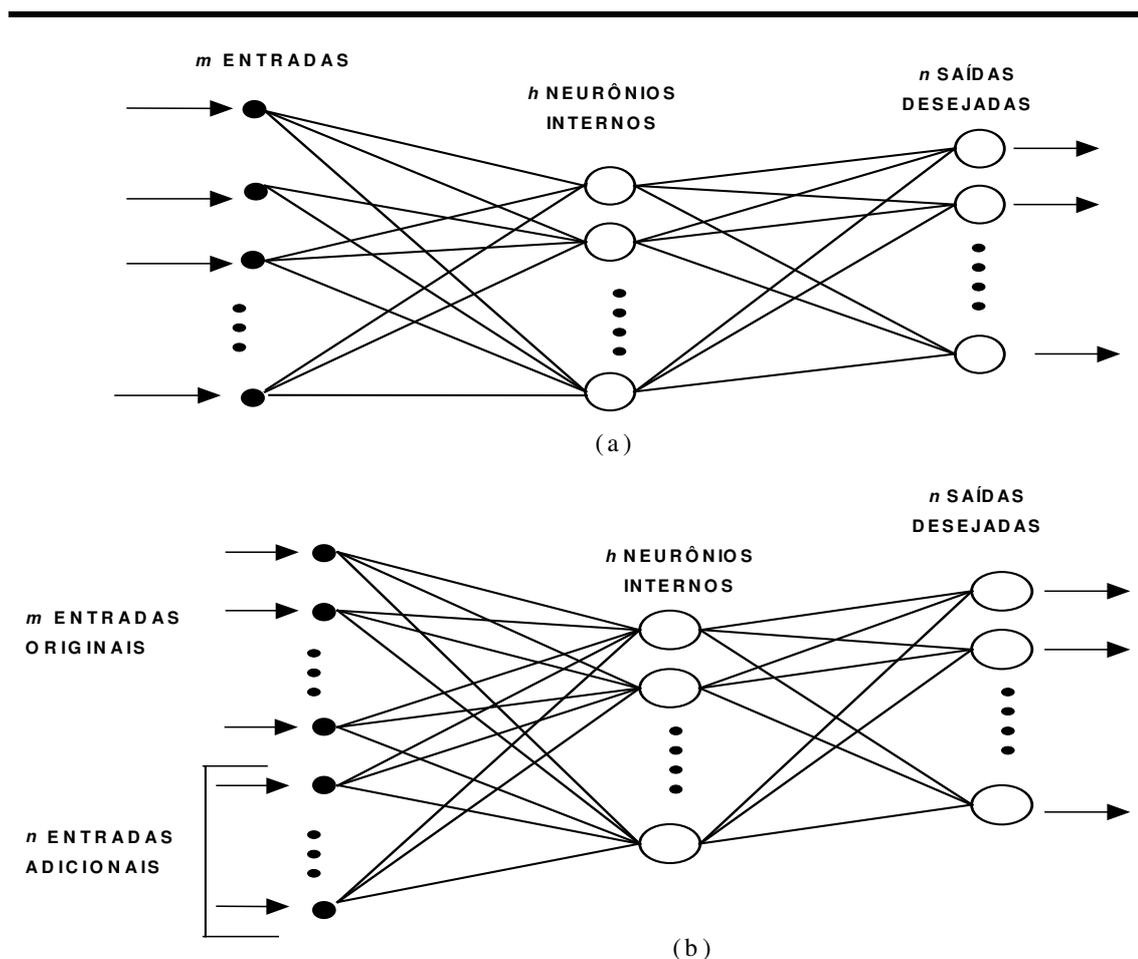


Figura D.2 - Topologias propostas para RNAs: (a) configuração original; (b) configuração com as saídas desejadas como entradas adicionais

Passo 1.2 - Soma dos erros quadráticos (SSE):

Se os pesos de uma entrada original e de uma entrada adicional com a camada interna são similares ou idênticos, é provável que haja uma forte associação entre eles. Se uma entrada é diretamente associada com a saída, estes pesos são idênticos. Geralmente não o são, pois pode haver ruído nos dados ou correlações fortes entre as próprias variáveis de entradas. O critério SSE é adotado para determinar a proximidade entre os pesos das entradas originais e os pesos de uma saída (entrada adicional) para com a camada interna. A expressão do SSE é dada por:

$$SSE_{ab} = \sum_j (W_{bj} - W_{aj})^2 \tag{D.1}$$

onde j é o número de unidades da camada interna, W_{bj} é o peso entre a saída desejada b (entrada adicional) e a unidade j da camada interna. O peso W_{aj} é o peso entre a entrada a e a unidade j da camada interna. A Figura D.3 mostra a aplicação da Equação D.1. Vale lembrar que os valores dos pesos são obtidos diretamente da matriz de pesos de dimensão $(m+n) \times s$.

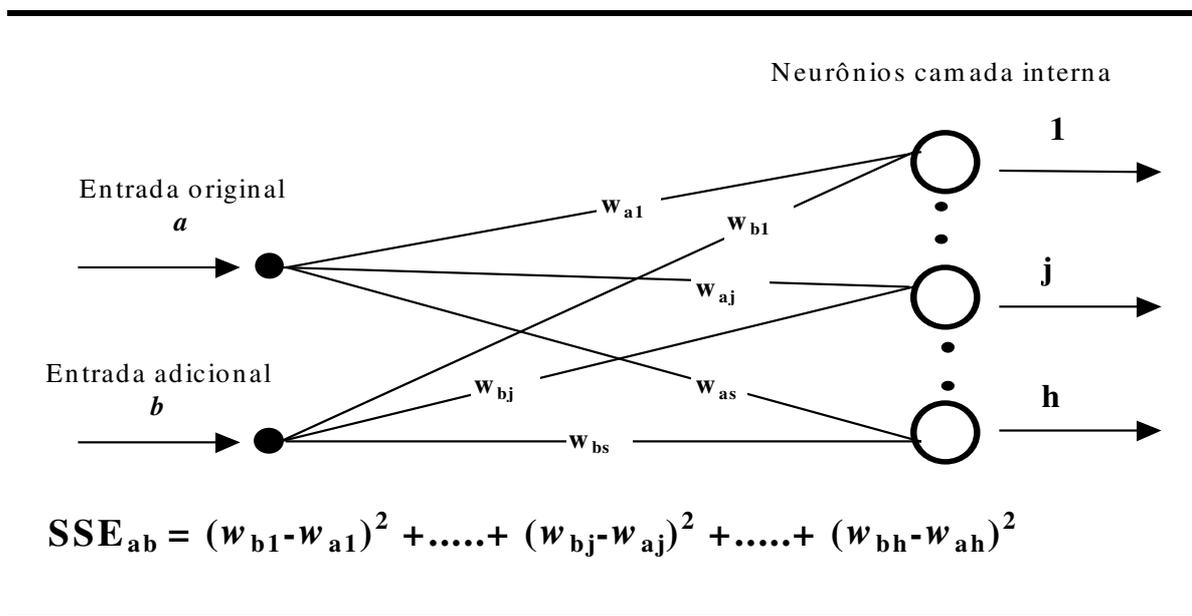


Figura D.3 - Cálculo do SSE entre a entrada original a e a entrada adicional b

Um pequeno valor de SSE geralmente indica que os atributos são fortemente relacionados. Se uma entrada é diretamente associada com uma dada saída, seu índice SSE é igual ou próximo de zero. Elevando-se ao quadrado as diferenças, diminui-se a medida para pequenas diferenças e acentua-se para grandes.

Quando o número de exemplares do conjunto de treinamento é pequeno, pode-se obter

pequenos SSEs para entradas não correlacionadas com uma dada saída. Neste caso, pode ser útil conhecer o número de vezes que uma entrada em particular é ativada e a saída não. O fato do conhecimento estar distribuído através da rede é outro obstáculo para se usar somente o índice SSE. Uma entrada em particular pode estar não apenas correlacionada a uma única saída. Por exemplo, a variável Mamífero está correlacionada com duas saídas: Marsupial e Morcego. Entretanto, é a combinação com outras variáveis de entrada (Mamífero AND Bolsa => Marsupial) que reforça a correta correlação entre esta entrada e a saída Marsupial. Então, uma medida adicional pode ser obtida para reforçar a correlação entre entradas e saída, a qual é mostrada sua obtenção no próximo passo.

Passo 1.3 - Obtenção da conexão inibitória:

Algumas vezes, o uso de características negativas (ou aspectos *inibitórios*) em aprendizagem indutiva, auxiliam a realçar as características positivas (ou aspectos *contribuintes*). Isto possibilita estabelecer melhor a correlação entre uma dada característica (atributo) e uma dada classe (saída). Desta forma, há um ganho em qualidade no resultado, utilizando o mesmo conjunto de treinamento. Uma maneira de obter uma *conexão inibitória* (ou aspecto inibitório) é negar a entrada. Seja o seguinte exemplar:

$$(1, 0, 1) \Rightarrow (1, 0).$$

Obtém-se, para a conexão inibitória, a seguinte negação do exemplar

$$(0, 1, 0) \Rightarrow (1, 0).$$

Para dados contínuos e normalizados, a negação se faz subtraindo de 1 o valor do atributo (Ex.: $\text{atrA} = 0,43$, $\text{atrNãoA} = 1,0 - 0,43 = 0,57$). Lembrando que uma RNA com uma camada treinada com regra de Hebb é uma associação linear entre entrada e saída, os pesos das conexões refletem a correlação entre uma entrada e uma saída. Observe que, se as entradas são negadas, quanto mais forte a correlação obtida entre uma entrada e uma saída, mais forte é a medida de irrelevância desta conexão. Então a matriz de pesos das conexões inibitórias, \mathbf{W} , de dimensão m entradas \times n saídas deve ser armazenada.

Passo 1.4 - Obtenção do produto:

Para se obter a medida da correlação entre uma entrada e uma saída é feita o produto da

conexão inibitória com a medida SSE entre todas as entradas e a respectiva saída. Este produto é zero ou próximo de zero para as entradas que estão diretamente associadas com a saída. O produto é dado por:

$$\text{Prod}_{ab} = W_{ab} \times \text{SSE}_{ab} \quad (\text{D.2})$$

onde W_{ab} é o peso da conexão inibitória entre a entrada a e a saída b e, SSE_{ab} é a medida SSE entre a entrada a e a entrada adicional b (correspondente à mesma saída b da conexão inibitória). Como o produto é de duas medidas, os efeitos são combinados dentro de uma estatística simples, ou seja, a medida da relevância entre uma entrada e uma saída. Observe que, se existem n saídas, então existem n listas de produtos. Se existem m atributos no vetor de entrada, então cada lista tem m valores de produtos, cada um destes associado a um atributo.

Etapa 2 - Seleção de Atributos

Passo 2.1 - Seleção dos atributos de definição para uma regra conjuntiva

Cada lista de produtos deve ter seus valores de produto ordenados em ordem decrescente. Deve existir um valor entre estes produtos que indica um ponto de corte (*cutoff*) distinto, determinando qual entrada deve ser utilizada como atributo da regra que define uma dada saída. Um ponto de corte é determinado quando o valor dos produtos de dois atributos consecutivos (da lista ordenada) é da ordem de 2 a 3 vezes maior em magnitude. Como um menor produto determina uma alta correlação entrada-saída, o atributo com valor de produto abaixo do ponto de corte é o atributo válido. Note que pode haver mais de um atributo abaixo do ponto de corte, como mostrado no exemplo a seguir. Neste exemplo é mostrado uma lista de produtos na qual cada atributo está associado com seu respectivo valor do produto:

$$\text{ListaProd} = (\text{B } 5,1) (\text{C } 3,15) (\text{E } 2,23) (\text{D } 0,72) (\text{A } 0,65) .$$

Pode ser observado que o valor do atributo E é aproximadamente 3,1 vezes maior que o produto do atributo D. Neste caso, o ponto de corte é estabelecido em E. Logo, os atributos válidos como antecedente da regra conjuntiva IF-THEN são os atributos A e D. A regra formada é dada por:

$$R1: (\text{A AND D}) \Rightarrow \text{saída } 1.$$

Valores de produtos como estes mostrados neste exemplo refletem dados com ruídos,

uma vez que não foram encontrados valores de produtos igual a zero. Quando um atributo apresenta valor de produto igual a zero, não há dúvidas quanto à correta correlação deste com a respectiva saída. Isto mostra que não há necessidade de verificar sua capacidade de especializar mais uma regra. Este atributo é então automaticamente incorporado à regra. Quando ocorrer casos onde não há produtos zero como o exemplo anterior, deve-se então testar cada atributo. Recomenda-se que isto seja feito para um atributo por vez, evitando compor regras como a R1 acima.

Para efeito de melhor apresentação do algoritmo, um exemplo é introduzido neste ponto, ilustrando as próximas etapas e seus respectivos passos. Observe que durante a execução é mostrado como são obtidas as regras disjuntivas.

Exemplo Teste

Seja o seguinte conjunto de treinamento, contendo 4 exemplares. Cada exemplar fornece uma saída X ou Y.

1. (a, b, c) \Rightarrow (X)
2. (a, e, f) \Rightarrow (X)
3. (g) \Rightarrow (X)
4. (a, d, e) \Rightarrow (Y)

O conjunto de treinamento, explicitando a presença (1) ou ausência (0) do atributo no exemplar, pode ser montado já considerando as saídas também como entradas adicionais. Este é feito da seguinte forma:

<u>ATRIBUTO</u>							<u>ENTRADAS ADICIONAIS</u>		<u>SAÍDA</u>	
a	b	c	d	e	f	g	X	Y	X	Y
1	1	1	0	0	0	0	1	0	1	0
1	0	0	0	1	1	0	1	0	1	0
0	0	0	0	0	0	1	1	0	1	0
1	0	0	1	1	0	0	0	1	0	1

Para a formação das regras que classificam corretamente este problema, é necessário primeiramente obter os valores dos produtos. Estes são obtidos através execução dos passos 1.1 a 1.4. A tabela D.1 mostra um provável resultado, onde os produtos já estão ordenados

Tabela D.1- Listas de produtos

LISTA PRODUTOS 1 SAÍDA X		LISTA DE PRODUTOS 2 SAÍDA Y	
ATRIBUTO	VALOR DO PRODUTO	ATRIBUTO	VALOR DO PRODUTO
D	5,50	G	5,40
B	1,15	F	5,40
C	1,15	C	5,40
F	1,15	B	5,40
G	1,15	E	0,0
F	1,15	D	0,0
A	0,55	A	0,0

1º ponto:

Uma vez que os passos 1.1 a 1.4 são executados e as listas de produtos adequadamente ordenadas, pode-se então executar iniciar a execução da Etapa 2. O Passo 2.1 é então aplicado. Pela Tabela D.1, nota-se que o atributo (a) apresenta menor valor de produto não zero para a saída X. Logo, este é o atributo inicial (ponto inicial) selecionado para começar o processo. Observe que existe um ponto de corte situado no atributo (d). Entretanto, se todos os atributos abaixo deste ponto são selecionados de uma vez, compondo uma única regra, todos os exemplares não são classificados. Então é necessário testar um atributo por vez. Testar um atributo, lidando-se com variáveis discretas, significa testar o efeito que a presença ou ausência deste produz na capacidade de classificação da regra. Para que este teste seja possível, duas regras devem ser formadas, refletindo a presença ou ausência do atributo. Assim, quando um atributo é selecionado, duas possíveis regras são geradas e, portanto, devem ser testadas. Este procedimento é denominado de *Gerar-e-Testar* e constitui a próxima etapa do BRAINNE. Para a saída Y, os atributos (a), (d) e (e) apresentam valores de produto igual a zero neste ponto, o que é esperado, uma vez que existe apenas um exemplar.

Etapa 3 – Procedimento Gerar-e-Testar

Uma metodologia gerar-e-testar constitui uma das mais básicas técnicas de busca heurística [RIC94]. Esta técnica envolve três passos: no primeiro, uma possível solução deve ser obtida, no segundo, esta possível solução deve ser testada e no terceiro, verificar se a solução é encontrada. Caso a solução não seja encontrada, deve-se retornar ao primeiro passo. No âmbito do BRAINNE, estes passos são mostrados como se segue.

Passo 3.1 – Formação das regras a serem testadas

Para o BRAINNE, gerar uma possível solução significa formar as regras, que podem ser vistas como possíveis caminhos de solução em uma estrutura em árvore. Através da continuidade da solução do Exemplo Teste, isto pode ser claramente observado.

Exemplo Teste – continuação

Conforme obtido, para a saída X, o atributo escolhido é o (a), formando duas regras que possibilitem verificar a capacidade de classificação, quanto à presença e ausência deste. As regras são:

Regra_1X: (a) \Rightarrow (X1) *Presença do atributo (a)
 Regra_2X: (Nãoa) \Rightarrow (X2) *Ausência do atributo (a)

onde X1 corresponde a saída X e X2 também.

Assim, a regra obtida para a saída Y é dada por:

Regra_1Y: (a AND d AND e) \Rightarrow (Y1)

onde Y1 denota a saída Y. A Figura D.4 ilustra graficamente o processo. Observe que a estrutura é uma árvore, onde cada variável a ser testada (neste caso, somente (a)) abre dois ramos.

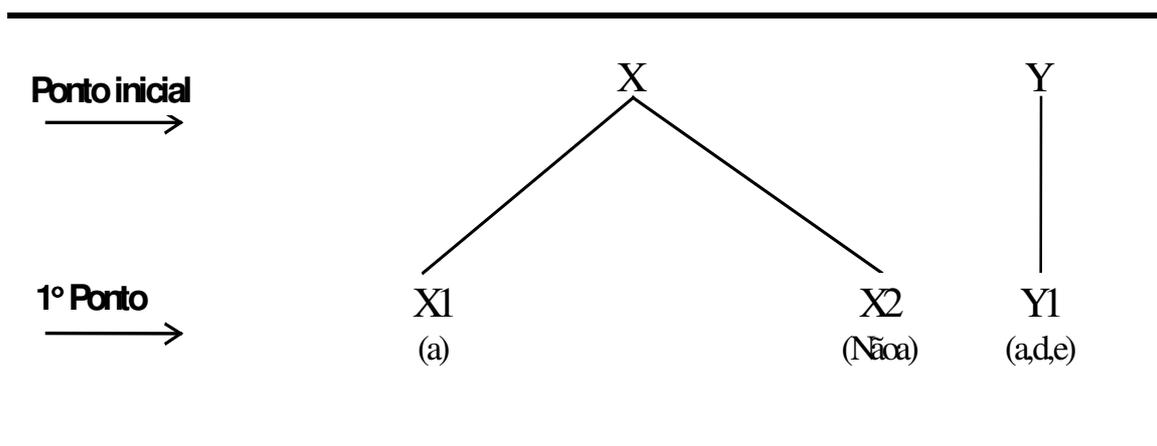


Figura D.4 – Obtenção das primeiras regras para o Exemplo Teste

Para a saída Y, como os atributos são automaticamente selecionados e tem poder máximo de classificação, apenas um ramo é formado, indicando somente a presença dos atributos (a), (d) e (e). Uma vez formadas as regras, estas devem ser testadas quanto a sua capacidade de classificação. Este processo é mostrado pelo próximo passo.

Passo 3.2 – Teste das regras formadas:

Neste passo devem ser executados dois diferentes testes para que uma determinada regra seja validada.

1º Teste: Número de exemplares cobertos por uma regra

O objetivo é verificar a capacidade das regras obtidas pelo passo anterior em classificar o maior número possível de exemplares. Neste caso a regra fornece ou não uma cobertura ao exemplar testado. Isto é feito verificando-se se os atributos que definem uma regra estão presentes ou ausentes em todos os exemplares do conjunto de treinamento, sem distinção de classes. Se a regra testada não cobre nenhum exemplar, então esta é descartada. Isto indica que a regra não é relevante para quaisquer exemplares no conjunto de treinamento. Se cobre uma minoria, isto pode estar refletindo presença de ruídos nos dados. Neste caso, esta pode ser armazenada justamente para esta finalidade. A continuidade do Exemplo Teste possibilita verificar este processo de teste.

Exemplo Teste - continuação

De acordo com o Passo 3.1, são obtidas duas regras para a saída X. Verificando a cobertura dos exemplares pelas estas duas regras, obtém-se que a Regra_1X cobre 3 exemplares. Entretanto, note que, dos 3 exemplares por ela cobertos, dois são corretamente classificados pois indicam a saída X e um é incorretamente classificado pois indica a saída Y (a regra é para a saída X). Este fato gera um procedimento que é discutido nos próximos passos. A Regra_2X cobre 1 exemplar, no caso o exemplar 3, classificando-o corretamente. Observe que a Regra_1Y classifica corretamente o exemplar 4 (único). Neste caso esta regra pode ser considerada uma regra válida e como classifica corretamente todos os exemplares pertencentes a sua respectiva classe (saída Y), então pode ser considerada uma regra final, onde $Y1 \Rightarrow Y$.

2º Teste : - O critério da unicidade:

O teste da unicidade permite verificar se entre os exemplares classificados por uma dada regra designando uma classe, não existem exemplares cuja saídas referem-se a(s) outra(s) classe(s). Isto possibilita verificar o poder do atributo em separar uma dada classe. Dois casos devem então serem considerados para cada regra:

- Não ocorre classificação errada - a regra pode ser parada se o 1º teste indicar todos os exemplares desta classe cobertos por esta regra ou uma grande e admissível maioria.
- Ocorre classificação errada - a regra necessita ser expandida para especialização, pois

o atributo não é suficientemente específico para promover a devida separação entre classes.

Vale salientar que este teste só ocorre após a execução do 1º teste, uma vez que os exemplares cobertos por uma regra devem ser ‘marcados’ e conseqüentemente contados. Através desta ‘marca’ é que o 2º teste pode verificar a unicidade da regra. Então, se todos os exemplares ou grande parte destes forem cobertos pela regra e esta é única, um *critério de parada* da regra é estabelecido. Para expandir uma regra, BRAINNE propõe uma sistemática própria, a qual se utiliza das regras formadas neste passo atual para constituírem as novas saídas do processo e então retornar ao passo 1.1 (processo recursivo). Este mecanismo é mostrado no próximo passo.

Nesta proposta de tese, um mecanismo de tomada de decisão é proposto para que o usuário possa também indicar se deseja ou não continuar a expansão de uma ou mais regras. Isto é colocado como um dispositivo adicional para visualizar outros atributos relevantes, porém não necessários ao processo de classificação. A cada final de execução deste passo atual, é mostrado ao usuário o *status* de cada regra em relação à necessidade de expansão. Neste caso, este pode decidir se expande ou não as regras que não necessitam de expansão. Se for indicado que todas as regras podem ser paradas e se constituírem em regras finais, pode também o usuário indicar que deseja continuar a expansão e quais regras devem ser expandidas.

Em [SES95] é levantado a intenção de proceder um terceiro teste para verificar se o atributo que é adicionado à regra fornece algum critério contraditório. Por exemplo, tendo-se uma regra com antecedente da seguinte forma: (A AND NãoB), um próximo atributo com menor produto diferente de zero selecionado poderia ser B. Isto proporciona gerar duas novas regras a serem testadas:

```
RN1: (A AND NãoB AND B) => 01  
RN2: (A AND NãoB AND NãoB) => 01
```

Entretanto, observe que a introdução de B produz uma contradição em RN1 e RN2 é igual a regra anterior que a gerou. Introduzindo-se um critério de não repetição de um atributo já selecionado e incorporado à uma regra considerada válida, o teste do atributo contraditório pode ser eliminado. A continuidade da execução do Exemplo Teste possibilita visualizar o 2º teste.

Exemplo Teste - Continuação

Uma vez realizado o 1º teste, pode-se verificar que a Regra_1X classifica errado o exemplar 4, pois sua saída é Y. Neste caso, para esta regra o critério da unicidade não é satisfeito.

Assim, a Regra_1X deve sofrer expansão. A Regra_2X cobre todos os exemplares (exemplar 3) e é única. Esta regra então pode ser considerada uma regra final. O mesmo ocorre com a Regra_1Y, onde esta cobre o exemplar 4 e é única. Assim, esta regra é parada e é considerada uma regra final. Logo, após os procedimentos de testes, as regras válidas obtidas são:

Regra_1X: (a) \Rightarrow (X1)	*Sofre expansão
Regra_2X: (Nãoa) \Rightarrow (X2)	*Regra final
Regra_1Y: (a AND d AND e) \Rightarrow (Y1)	*Regra final

onde $X1$ e $X2 = X$ e $Y1 = Y$. Uma vez que, pelo menos uma regra necessita de expansão para obter mais informação discriminatória, o processo deve ser recursivamente repetido. Entretanto, como observado, BRAINNE utiliza-se destes regras formadas como as novas saídas do processo.

Passo 3.3 - Utilização de todas as regras como as novas saídas:

O primeiro processo determinando os produtos relacionados com cada atributo de uma dada saída é feito considerando os conjuntos de treinamento e teste originalmente formados. Então, as duas primeiras regras são formadas para cada saída, a partir do primeiro atributo selecionado. Quando são executados os Passos 3.1 e 3.2, é verificada a validade destas primeiras regras obtidas. As regras consideradas válidas são então armazenadas. Para a obtenção dos novos atributos especializando as regras (caso necessário), BRAINNE propõe utilizar todas as regras válidas, que no momento identificam as saídas, como as novas saídas do sistema e portanto podem ser usadas como entradas adicionais, o que determina a remontagem do conjunto de treinamento. Isto determina uma nova configuração da RNA-ML (conforme mostra a Figura D.5) para a obtenção dos novos produtos. Note também que o conjunto de treinamento para a RNA com uma camada é também remontado. Num processo recursivo, a Etapa 1 é executada e obtém-se uma nova lista de produtos. Uma vez obtido os produtos, a Etapa 2 seleciona os novos atributos. Logo, o procedimento de gerar-e-testar pode ser novamente acionado (Etapa 3), formando as novas regras conjuntivas, especificando a presença e ausência do atributo, determinado pelo passo 3.1. Definidas as regras, segue-se então a verificação dos critérios de parada, executando-se o passo 3.2. Este passo 3.3, atual, só é novamente acionado se não houver a indicação, seja por necessidade ou controle do usuário, de expandir mais nenhuma regra. Caso a decisão é terminar o processo, deve-se então executar a próxima etapa (Etapa 4).

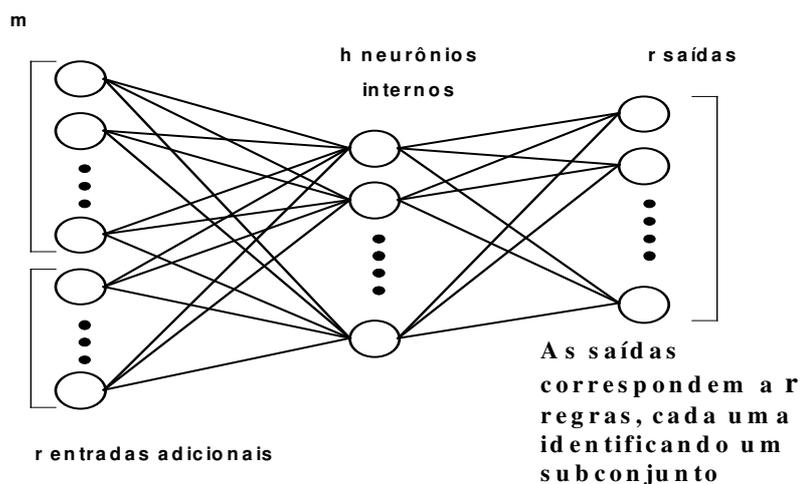


Figura D.5 - Nova configuração da Rede Neural.

Uma regra é utilizada como saída da seguinte forma: os exemplares por ela cobertos são indicados por 1 e os não cobertos por zero. Observe que, se uma regra cobre erradamente um exemplar, o valor da regra para este exemplar é também zero. O Exemplo Teste mostra com clareza este processo. A continuidade deste exemplo, a partir deste ponto é feita até que todas as regras finais sejam obtidas, uma vez que este ponto de execução do algoritmo introduz a recursividade do processo.

Exemplo Teste - Continuação

Conforme o executado até o momento, três regras de produção são geradas e testadas, indicando a necessidade expansão da Regra_1X.

2º ponto:

Uma vez que Regra_1X deva ser expandida, recursivamente o processo é reiniciado. Antes de se retornar a Etapa 1, as regras obtidas são utilizadas agora como novas saídas, remontando novamente um conjunto de treinamento. Quando um exemplo é apresentado à rede, o valor de cada regra é indicado 0 ou 1. O valor 1 indica que o exemplar satisfaz a regra. Se 0, a regra não cobre o exemplar. Pode-se observar que, apesar da Regra_1X também classificar o exemplar 4, esta o faz de forma errada. Portanto, este exemplar não satisfaz a regra, sendo indicado em 0. Logo, o novo conjunto de treinamento é dado por:

<u>ATRIBUTO</u>							<u>ENT. ADICIONAIS</u>			<u>SAÍDA</u>		
							<u>(REGRAS NOVAS)</u>			<u>(REGRAS NOVAS)</u>		
<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>1X</i>	<i>2X</i>	<i>1Y</i>	<i>1X</i>	<i>2X</i>	<i>1Y</i>
1	1	1	0	0	0	0	1	0	0	1	0	0
1	0	0	0	1	1	0	1	0	0	1	0	0
0	0	0	0	0	0	1	0	1	0	0	1	0
1	0	0	1	1	0	0	0	0	1	0	0	1

O processo de obtenção dos novos produtos segue os passos 1.1 ao 1.4 anteriormente definidos. Como são definidas as novas regras como saídas, então tem-se uma lista de produtos para cada regra (que são as novas saídas). Após a obtenção destas novas listas de produtos, são selecionados os novos atributos com menor valor de produto (excetuando-se os já selecionados), pelo processamento da Etapa 2. Para efeito de simplicidade, suponha o atributo (f) o próximo atributo com menor valor de produto para a Regra_1X. Para a lista correspondente a Regra_2X, fica evidente que o atributo (g) tem produto zero neste ponto e é pode ser automaticamente selecionado. Apesar da Regra_2X ser considerada uma regra final, pois classifica corretamente e unicamente o exemplo 3 no ponto anterior, a formação da Regra_21X se deve ao fato de obter-se produto zero para o atributo (g). Logo, a inclusão do atributo (g) como antecedente da Regra_2X, formando a Regra_21X, proporciona informação adicional especializada sobre o domínio e não reduz a capacidade de classificar. Logo, as regras obtidas são:

Para a saída X, tem-se

Regra_11X: (a AND f) \Rightarrow (X3)
 Regra_12X: (a AND Nãoof) \Rightarrow (X4)
 Regra_21X: (Nãooa AND g) \Rightarrow (X5)

onde X3 \Rightarrow X, X4 \Rightarrow X e X5 \Rightarrow X.

Para a saída Y, a Regra_1Y é regra final e pode ser parada. Portanto, esta é repetida:

Regra_1Y: (a AND d AND e) \Rightarrow (Y2)

com Y2 \Rightarrow Y.

Avaliando a cobertura dos exemplares pelas regras, observa-se que a Regra_11X não produz classificação errada, portanto é uma regra válida. Entretanto, a Regra_12X classifica errado o exemplo 4. Logo, esta regra deve ser especializada.

3° ponto:

Uma vez que a Regra_12X deve ser especializada, o processo é novamente desencadeado. Desta forma, um novo conjunto de treinamento é montado, utilizando-se destas novas regras formadas. Este pode ser visualizado abaixo:

<u>ATRIBUTO</u>							<u>ENT. ADICIONAIS</u>				<u>SAÍDA</u>			
							<u>(REGRAS NOVAS)</u>				<u>(REGRAS NOVAS)</u>			
<u>a</u>	<u>b</u>	<u>c</u>	<u>d</u>	<u>e</u>	<u>f</u>	<u>g</u>	<u>11X</u>	<u>12X</u>	<u>21X</u>	<u>1Y</u>	<u>11X</u>	<u>12X</u>	<u>21X</u>	<u>1Y</u>
1	1	1	0	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	1	1	0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	1	0	0	0	1	0
1	0	0	1	1	0	0	0	0	0	1	0	0	0	1

Calculando-se novamente os produtos, obtém-se a lista de produtos para cada regra. Para a regra Regra_11X, o atributo (e) apresenta produto zero neste e é automaticamente selecionado. Na Regra_12X, o mesmo acontece com os atributos (b) e (c) e são também automaticamente selecionados. Desta forma, o atributo (g) apresenta o menor valor de produto não zero, para a Regra_12X, sendo escolhido. Isto produz as regras mostradas abaixo:

Para a saída X, tem-se

Regra_111X: (a AND f AND e) \Rightarrow (X6)
 Regra_121X: (a AND Nãoof AND b AND c AND g) \Rightarrow (X7)
 Regra_122X: (a AND Nãoof AND b AND c AND Nãoog) \Rightarrow (X8)
 Regra_21X: (Nãooa AND g) \Rightarrow (X5) * Regra final

onde X6 \Rightarrow X, X7 \Rightarrow X, X8 \Rightarrow X. Para a saída Y:

Regra_1Y: (a AND d AND e) \Rightarrow (Y2) * Regra final

onde Y2 \Rightarrow Y.

Verificando-se a cobertura dos exemplares pelas novas regras formadas, nota-se que a Regra_111X já classificava corretamente o exemplar 1 e é única, obtendo agora a informação adicional promovida pelo atributo (e) como antecedente. Entretanto, a Regra_121X não classifica qualquer exemplar, sendo abandonada. A Regra_122X classifica corretamente o exemplar 2 e também é única. As Regras_21X e 1Y não sofreram qualquer modificação. Observe que, neste ponto, todos os atributos são selecionados e as regras obtidas são consideradas as regras finais, pois classificam corretamente todo o conjunto de treinamento.

Etapa 4 – Base de Conhecimento

Passo 4.1 - Terminar o processo:

Enquanto houver necessidade de especialização de uma regra, o processo é recursivamente reiniciado. Isto é feito até que somente regras finais (regras paradas em algum ponto da expansão) sejam obtidas e/ou somente um pequeno número de exemplares (preferencialmente zero) tenha sido classificado erradamente. Se o número total de exemplares cobertos pela regra é menor que algum limite w , então a regra deve ser descartada. Entretanto, uma regra pode ser retida para refletir alguma exceção e/ou ruído nos dados.

Exemplo Teste – Continuação:

O conjunto final das regras que classificam corretamente este conjunto de exemplares é dado por:

```

Regra_111X: (a AND f AND e) ⇒ (X) == final
Regra_122X: (a AND Nãoof AND b AND c AND Nãoog) ⇒ (X) == final
Regra_21X: (Nãooa AND g) ⇒ (X) == final
Regra_1Y: (a AND d AND e) ⇒ (Y) == final
    
```

A Figura D.6 ilustra todo o processo. Desta forma, a Base de Conhecimento para este domínio é composta da regra disjuntiva final obtida para a saída X e da regra final conjuntiva (única) para a saída Y, como mostrado abaixo:

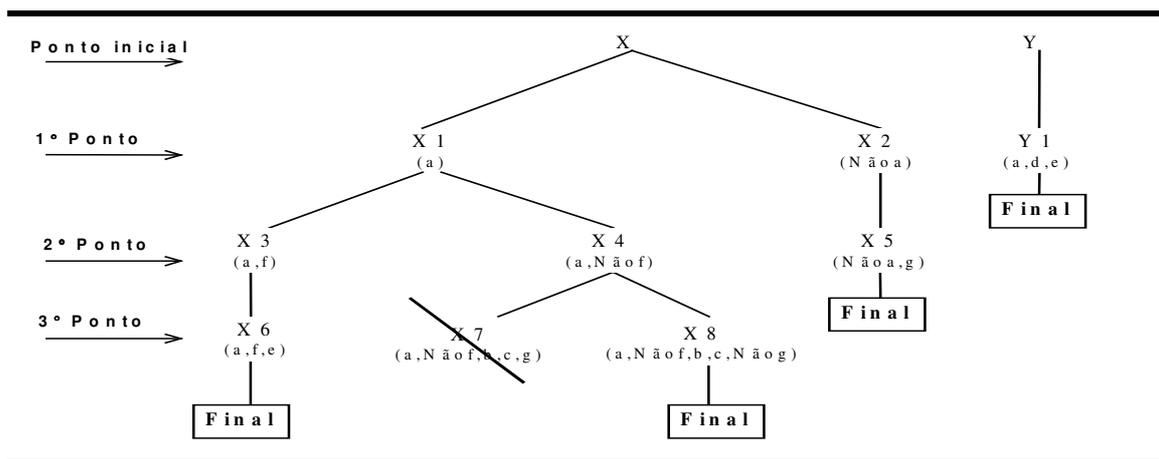


Figura D.6 - Visualização do processo de extração de regras para o exemplo

Passo 4.2 – Formação da Base de Conhecimento

O objetivo neste passo é colocar as regras de forma adequada para uma possível asserção por parte de um mecanismo de inferência.

Base de Conhecimento:

Saída X:

$$\{ (a \text{ AND } f \text{ AND } e) \quad \text{OR} \\ (a \text{ AND } \text{Nãof} \text{ AND } b \text{ AND } c \text{ AND } \text{Nãog}) \quad \text{OR} \\ (\text{Nãoa} \text{ AND } g) \}.$$

Saída Y:

$$\{ (a \text{ AND } d \text{ AND } e) \}.$$

Observe que o *Exemplo Teste* descrito é simples, sendo que as regras finais obtidas são reflexos diretos dos exemplares originais. Quatro regras são produzidas, cada uma descrevendo um exemplar positivo da saída. Em exemplos reais, cada regra deve cobrir um grupo de exemplares com mesmas características. Entretanto, regras cobrindo exceções (i.e., somente poucos exemplares) podem também ser encontradas, como já observado.

Relembrando, BRAINNE obtém uma medida de correlação entre as entradas do processo e uma dada saída. Esta medida indica o atributo a ser selecionado, formando um grupo de regras as quais devem ser testadas. Os testes incluem a contagem dos exemplares cobertos por uma dada regra e se esta é única. Isto estabelece um critério de parada. Caso este não é satisfeito, então deve haver especialização das regras que não satisfazem este critério. Caso necessário especializar ao menos uma regra, então, recursivamente, o processo é repetido até que regras finais sejam encontradas. Estas regras finais definem a Base de Conhecimento do domínio do problema.

D.2.2 BRAINNE aplicado a dados contínuos

Conforme observado para a aplicação do BRAINNE à dados discretos, o valor do atributo representado no conjunto de dados é 1 se este está presente ou 0 se ausente. Quando se lida com dados contínuos, o atributo está sempre presente, ou seja, existe uma faixa de valores a qual um dado valor do atributo pertence. Formalmente, uma variável é contínua se seus valores estiverem dentro de um intervalo A e B, tal que $A < B$. Uma maneira de se lidar com dados contínuos é discretizando-os. Isto pode ser feito por duas técnicas:

1. Um valor limite T é estabelecido tal que $A < T$ e $T < B$, onde os intervalos são dados

por $(A, T]$ e (T, B) .

2. São estabelecidos vários valores limites de forma que, $A < T_1, T_1 < T_2, \dots, T_n < B$. Neste caso, existem $n+1$ intervalos e geralmente são iguais.

O problema com a primeira é que, a escolha de T não é trivial e pode descaracterizar o problema. Por exemplo, se um valor de 60°C é obtido como valor limite para temperatura, o valor de 80°C pertence ao segundo intervalo. Entretanto, 245°C também pertence a este segundo intervalo, mas é significativamente maior, podendo estar indicando outro tipo de saída, diferente da indicada pela temperatura de 80°C . Para a segunda técnica, dois problemas são observados: o aumento da dimensionalidade do vetor de entrada e a determinação de faixas irrelevantes. Observe que, se existem três variáveis de entrada e são designados 10 intervalos de discretização, então têm-se 3×10 variáveis de entrada, o que não é bom devido ao curso da dimensionalidade, como já observado. No processo de definição dos intervalos, existe, por exemplo, um de temperatura dado por 10°C e 20°C . Entretanto, a faixa mais significativa pode ser 15°C e 25°C , o que torna a faixa anterior irrelevante ao problema.

Se a um problema for adequado impor uma das discretizações acima mostradas, então a metodologia do BRAINNE aplicada a dados discretos pode ser utilizada. Mas, uma característica positiva das RNAs é que elas lidam bem tanto com dados discretos como contínuos, o que possibilita não ter que pré-processar os dados significativamente. No caso do BRAINNE aplicado a dados contínuos, deve-se proceder apenas uma normalização dos dados entre 0 e 1.

Vale ressaltar que, mesmo lidando com dados contínuos, as Etapas 1 e 2 do BRAINNE aplicado a dados discretos são as mesmas, uma vez que, estas etapas são independentes do valor que o atributo pode assumir, sendo dependentes apenas do treinamento das RNAs e do critério de seleção dos atributos. Assim, a seleção de um atributo segue os mesmos passos anteriormente indicados. Entretanto, para disparar o procedimento gerar-e-testar, existem distinções a serem observadas lidando-se com dados contínuos. Seja o seguinte atributo AtrA em relação à uma saída $O1$ selecionado pelas Etapas 1 e 2 tal que, a regra formada é dada por:

$R1: (\text{AtrA}) \Rightarrow O1.$

Como já observado, os valores de uma variável contínua estão contidos em um dado intervalo. Então, existe um valor mínimo e um valor máximo indicando o intervalo. Isto é também refletido no antecedente da regra da seguinte forma:

$R1: (\text{MinA} \leq \text{AtrA} \leq \text{MaxA}) \Rightarrow O1.$

A questão agora torna-se encontrar os valores MinA e MaxA . Se o processo de aquisição de

conhecimento é automático, então o próprio método deve ser capaz de encontrar estes valores automaticamente, sem qualquer critério arbitrário. Isto é feito através de três passos.

Passo 1 - Proximidade de um exemplar com a regra:

Um regra é definida pela presença ou ausência de um atributo. Se o vetor de entrada é constituído de quatro atributos, (A, B, C, D), um possível antecedente de regra designando uma classe pode ser dado por (A AND C AND NãoD). Observe que, como mostrado neste antecedente, A e C estão presentes e D ausente. Como dito anteriormente, um dado contínuo está sempre presente. Entretanto, um atributo contínuo pode ser dito ausente (ou negado) se os valores da faixa que o define são próximos de zero, ou seja, são valores baixos (lembrando que estes estão normalizados entre 0 e 1).

Observe que no caso discreto, os atributos (A, B, C, D) podem assumir os valores (1/0,1/0,1/0,1/0), onde 1/0 significa um **ou** zero. Isto determina a situação ideal para uma regra, permitindo polarizar os valores e tornando a separação entre classes mais distinta. Assim, um atributo contínuo com a faixa de seus valores próximos de 1 ou de 0, possibilita que a regra mais efetiva em classificar. O exemplo abaixo ilustra a metodologia. Seja a Regra_1 definida por:

$$(A \text{ AND } B \text{ AND } \text{Não}D) \Rightarrow 01$$

então os valores ideais para o antecedente da Regra_1 são:

$$(1, 0 \text{ } 1, 0 \text{ } 0, 0) .$$

Estes valores são os *valores dos atributos na regra*. Entretanto, estes atributos são contínuos e, portanto, uma medida estatística é adotada para calcular a proximidade de um exemplar (a_1, b_1, c_1, d_1) com esta Regra_1. Primeiro, determina-se a diferença entre os valores de um atributo na regra e o *valor real do atributo correspondente no exemplar*. Isto fornece uma medida do erro entre a regra e um exemplar para um particular atributo. Elevando ao quadrado e somando esta medida para todos os atributos que definem a regra resulta numa medida global de proximidade entre um exemplar e a regra, formalmente definida por:

$$\text{SumDif} = \sum_i (R : \text{ideal} : \text{atr}_i - E : \text{real} : \text{atr}_i)^2 \quad (\text{D.3})$$

onde i é o número de atributos definindo uma regra, atr_i é o atributo que neste instante é considerado, $R:\text{ideal}:\text{atr}_i$ é o valor do atributo na regra e $E:\text{real}:\text{atr}_i$ é o valor do atributo no

exemplar. Retornando ao exemplo anterior, dado pela Regra_1, sendo os valores dos atributos dado por (1,0, 1,0, 0,0), o valor do índice SumDif (Equação D.38) para o exemplar é dado por:

$$SD1 = (1,0 - a_1)^2 + (1,0 - b_1)^2 + (0,0 - d_1)^2.$$

Observe que, se o atributo está presente, sua medida de proximidade é feita com 1. Se o atributo está negado (ausente), sua medida de proximidade é feita com zero, como é o caso do atributo D. Este índice é obtido para todos os exemplares designando a saída O1. Se existem n_1 exemplares, há n_1 índices SumDif. A questão então se torna: Quais, dentre os n_1 índices SumDif encontrados, efetivamente indicam que a Regra_1 tem boa capacidade de classificar? O próximo passo mostra como isto pode ser obtido.

Passo 2 – Determinação do Valor Limite:

Um valor limite t (*threshold*) é escolhido para delimitar se o exemplar é coberto ou não pela regra. Se o valor de SumDif está abaixo ou é igual ao valor de t , então o correspondente exemplar é coberto pela Regra_1: (A AND B AND NãoD), caso contrário, não é. Formalmente, para o caso do exemplar (a_1, b_1, c_1, d_1) tem-se:

- $SD1 \leq t$ – o exemplar é coberto pela Regra_1
- $SD1 > t$ – o exemplar não é coberto pela Regra_1

Uma amostra de cálculo pode ser vista através do seguinte exemplo. Considere a existência de dois exemplares do conjunto de dados, para os atributos (A,B,C,D):

Ex1: (0,75 0,85 0,45 0,20)
 Ex2: (0,75 0,85 0,45 0,80)

Logo, o cálculo da SumDif para os dois exemplares são:

$$\begin{aligned} SD_{Ex1} &= (1,0 - 0,75)^2 + (1,0 - 0,85)^2 + (0,0 - 0,20)^2 = 0,1250 \\ SD_{Ex2} &= (1,0 - 0,75)^2 + (1,0 - 0,85)^2 + (0,0 - 0,80)^2 = 0,7250 \end{aligned}$$

Como o valor da SumDif para o exemplar 1 ($SD_{Ex1} = 0,1250$) foi menor que o valor limite $t=0,2$, então a Regra_X cobre o exemplar 1. O mesmo não acontece com o exemplar 2, cuja SumDif ($SD_{Ex2} = 0,7250$) é maior que $t=0,2$, o que corresponde a não cobertura pela Regra_X.

Pelo observado, a escolha do valor limite é uma consideração importante na metodologia. Para dados contínuos, a medida que as regras crescem (se tornam mais específicas), mais atributos são incorporados à regra e necessariamente precisam ser considerados no cálculo do

índice SumDif. Se o valor limite t é pequeno, alguns dos exemplares não são cobertos por causa do aumento do número de atributos contribuindo com a SumDif. Então, a medida que as regras crescem, é importante que o valor limite seja incrementado para classificar corretamente os exemplares.

No BRAINNE, o valor limite é dinamicamente checado e incrementado (se necessário) depois da divisão do conjunto. Após a divisão, os exemplares são primeiro classificados usando o mesmo valor limite t aplicado ao subconjunto anterior não dividido. Dois casos podem acontecer:

- Se um suficiente grande número de exemplares é coberto pelo novo conjunto de regras então não há mudança no valor de t
- Por outro lado, se há um não suficiente números de exemplares não cobertos, t deve ser incrementado e a classificação dos exemplares é refeita.

O processo é continuado, se requerido, até que todos (ou um suficiente grande número) dos exemplares originais sejam cobertos. Idealmente, o valor limite t máximo que é aquele o qual o número de exemplares cobertos por uma regra é igual ao número total de exemplares do conjunto de treinamento para uma dada saída. Entretanto, é prudente escolher um valor limite t máximo previamente, pois isto proporciona que os atributos selecionados para compor a regra sejam efetivos em mostrar a interpretação do processo. Caso contrário, pela ação de t , atributos fracamente correlacionados podem ser incorporados às regras, perdendo-se interpretação do processo.

A metodologia para a escolha dinâmica do valor limite pode ser melhor visualizada através do seguinte exemplo: Considere o exemplo anterior, com os seguintes atributos de entrada:

(A, B, C, D)

e uma regra formada dada por

Regra_X: (A AND B AND NãoD) \Rightarrow O₁

e valor limite $t = 0,2$. O exemplar 1 a ser examinado tem os seguintes valores:

EX1: (0,75 0,85 0,45 0,20)

Faça o próximo atributo escolhido para a Regra_X ser C. Logo:

Regra_X1: (A AND B AND NãoD AND C) \Rightarrow O₁
 Regra_X2: (A AND B AND NãoD AND NãoC) \Rightarrow O₁

e o cálculo de suas respectivas SumDifs

$$\begin{aligned} \text{SDX1} &= (1, 0-0, 75)^2 + (1, 0-0, 85)^2 + (0, 0-0, 20)^2 + (1, 0-0, 45)^2 = 0,4275 \\ \text{SDX2} &= (1, 0-0, 75)^2 + (1, 0-0, 85)^2 + (0, 0-0, 20)^2 + (0, 0-0, 45)^2 = 0,3275 \end{aligned}$$

Checando estes valores com o valor de t , verifica-se o exemplar EX1 não é coberto nem pela Regra_X1 nem pela Regra_X2. Isto não é correto, pois este exemplar é coberto pela regra original X. Para solucionar este problema o valor limite deve ser dinamicamente incrementado, até que todos (ou a maioria) dos exemplares sejam cobertos pelas regras determinadas.

Passo .3 – Determinação da faixa de cada atributo presente na regra:

O próximo passo é determinar a magnitude dos atributos contínuos. Isto requer o seguinte processo: executando-se os Passos 1 e 2, é possível definir quais os exemplares cobertos por uma regra. Como estes exemplares são ‘marcados’, os valores mínimo e máximo dos atributos definindo a regra são obtidos neste conjunto de exemplares coberto pela regra (‘marcados’). A seguir, são mostrados os 2 passos para estabelecer a faixa contínua das entradas contribuintes de uma regra:

- Passe por todos os exemplares e determine quais são cobertos pela regra através do cálculo da SumDif. Estabeleça uma ‘marca’ em todos os exemplares cobertos pela regra.
- Para todos os exemplares cobertos pela regra, determine os valores mínimo e máximo para cada atributo de definição da regra, examinando os valores de cada atributo no conjunto de exemplares cobertos pela regra.

Os valores mínimos e máximos representam um intervalo para o correspondente atributo contínuo. Estes são determinados porque este refinamento restringe os valores dos atributos a um intervalo fechado. Destas bandas, a forma de cada cláusula no antecedente da regra conjuntiva é: (MinA \leq atrA \leq Max), conforme já demonstrado anteriormente. Veja o seguinte exemplo: seja a Regra_X: (a AND Nãoc) e considere os seguintes exemplares definindo os atributos a, b e c cobertos pela Regra_X:

EX1: (0,90 0,51 0,20)
 EX2: (0,80 0,68 0,10)

EX3: (0,85 0,75 0,25)

Então, as seguintes cláusulas podem ser obtidas:

$$((0,80 \leq a \leq 0,90) \text{ AND } (0,10 \leq c \leq 0,25))$$

que são determinadas dinamicamente. Fica evidente a vantagem destes intervalos obtidos diretamente dos dados e não do usuário, que pode não saber qual é o intervalo real ou distribuição dos atributos de entrada. Outra vantagem é que estes intervalos são determinados separadamente para cada regra, o que possibilita um atributo ter intervalos diferentes em diferentes regras.

Os exemplares são finalmente classificados de acordo com o determinado intervalo e não de acordo com o arbitrário valor limite que foi dinamicamente escolhido. Isto se deve fato da distribuição dos exemplares usando o valor limite pode não ser a mesma que a distribuição obtida através dos intervalos. Entretanto, a classificação do valor limite é necessária para calcular os intervalos de um atributo em particular definindo várias regras. Estas regras habilitam BRAINNE a criar grupos de exemplares que podem ser classificados juntos.

O que pode ser observado é que, após a execução das Etapas 1 e 2 para a seleção dos atributos candidatos, as regras formadas são testadas quanto a cobertura de exemplares diferentemente da Etapa 3 – Passo 3.1 para dados discretos. Alternativamente, para dados contínuos, devem ser executados os Passos 1, 2 e 3. Entretanto, o teste da unicidade (Passo 3.2) deve ser também executado, agora sob o enfoque do novo tipo de cláusula (contendo as faixas dos atributos) obtida para dados contínuos.

Passo 4 – Testando novos exemplares

Uma vez formadas as regras consideradas finais, estas podem ser testadas através do conjunto de teste inicialmente obtido pela etapa da formação da base de dados. Testando-se estes exemplares do conjunto de teste, pode-se obter taxas mais elevadas de erros de classificação além do esperado. Isto pode estar refletindo a presença de ruídos. BRAINNE proporciona que os intervalos de cada atributo sejam expandidos. Isto melhora a capacidade de generalização do conjunto de regras obtidas. Considere a seguinte regra final:

$$(a_1 \leq \text{atrA} \leq a_2) \Rightarrow O_1.$$

Os limites a_1 e a_2 podem ser modificados de acordo com as equações:

$$\begin{aligned} a_1 &= a_1 - (a_2 - a_1) * \theta \\ a_2 &= a_2 - (a_2 - a_1) * \theta \end{aligned} \quad (D.4)$$

Aqui, θ é escolhido para refletir as questões próprias de cada problema e é normalmente uma pequena fração. Se esta modificação é requerida, cada cláusula é ajustada separadamente.

D.2.3 Exemplo de aplicação para dados contínuos

O exemplo a seguir mostra a aplicação do algoritmo quando os dados são contínuos. Considere algumas medidas elétricas obtidas em um sistema elétrico de potência, onde tem-se tensões e potências ativa das barras 1 e 2. Tais medidas foram obtidas sob condições seguras e inseguras de operação. A Tabela D.2 mostra os dados, normalizados entre 0 e 1, para as medidas observadas.

Tabela D.2 - Medidas elétricas de um fictício sistema elétrico de potência.

exemplar	ATRIBUTOS DE ENTRADA				SAÍDA	
	POT. ATIVA BARRA 1 (PA1)	TENSÃO BARRA 01 (V1)	TENSÃO BARRA 02 (V2)	POT. ATIVA BARRA 2 (PA2)	SEGURO (SE)	INSEGURO (IN)
1	0,00	1,00	1,00	0,00	1	0
2	0,13	0,95	0,93	0,21	1	0
3	0,22	0,91	0,83	0,25	1	0
4	0,30	0,85	0,79	0,35	1	0
5	0,48	0,64	0,69	0,46	1	0
6	0,53	0,43	0,48	0,54	0	1
7	0,78	0,35	0,40	0,82	0	1
8	0,86	0,27	0,37	0,87	0	1
9	0,92	0,20	0,25	0,94	0	1
10	1,00	0,00	0,00	1,00	0	1

Seguindo a execução de BRAINNE, o primeiro passo é proceder o treinamento da RNA-ML com a configuração das saídas como entradas adicionais. Os dados foram divididos em dois conjuntos: 60% das medidas em um conjunto de treinamento e os restantes 40% em um conjunto de teste. Desta forma, os dados apresentados para a rede neural devem ser colocados na seguinte forma:

CONJUNTO DE TREINAMENTO							
ENTRADAS ORIGINAIS				ENT. ADICIONAIS		SAÍDA	
0,00	1,00	1,00	0,00	1	0	1	0
0,22	0,91	0,83	0,25	1	0	1	0
0,48	0,64	0,69	0,46	1	0	1	0
0,53	0,43	0,48	0,54	0	1	0	1
0,86	0,27	0,37	0,87	0	1	0	1
1,00	0,00	0,00	1,00	0	1	0	1
CONJUNTO DE TESTE							
ENTRADAS ORIGINAIS				ENT. ADICIONAIS		SAÍDA	
0,13	0,95	0,93	0,21	1	0	1	0
0,30	0,85	0,79	0,35	1	0	1	0
0,78	0,35	0,40	0,82	0	1	0	1
0,92	0,20	0,25	0,94	0	1	0	1

Para este exemplo, a configuração proposta para a RNA-ML é considerada com 7 camadas internas e parâmetros (ver Apêndice A): momentum = 0,40 e coeficiente de aprendizagem = 0,50. Uma característica do treinamento desta rede é que o critério de parada do treinamento é feito quando uma taxa de classificação ótima é obtida. Para isto, a cada 2000 iterações é acessado o conjunto de teste e verificado a taxa de classificação (validação cruzada). Assim, quando os pesos da rede convergirem a valores ótimos em relação a esta classificação, o treinamento é parado e os pesos armazenados.

Obtida a matriz de pesos entradas-camada interna, é necessário encontrar o SSE para cada entrada original em relação à saída, como já explicado anteriormente. Um exemplo deste cálculo é mostrado para entrada Potência Ativa barra 1 e a saída Seguro. Considere o vetor pesos da conexão entrada PA1 e a camada interna (7 neurônios), W_{PA1} e, o vetor pesos da conexão entrada adicional SE e a camada interna, W_{SE} .

$$W_{PA1} = \begin{bmatrix} -0,2548 \\ -0,0238 \\ -0,0198 \\ 0,3740 \\ -0,2031 \\ 0,2571 \\ 0,4980 \end{bmatrix} \qquad W_{SE} = \begin{bmatrix} -0,6799 \\ 0,2149 \\ 1,9392 \\ -2,1674 \\ 0,3420 \\ 0,2667 \\ 1,0076 \end{bmatrix}$$

Desta forma, o SSE pode ser calculado da seguinte forma, através da Equação D.36:

$$SSE_{PA1-SE} = (-0,6799 + 0,2548)^2 + (0,2149 + 0,0238)^2 + \dots + (1,0076 - 0,4980)^2 = 11,091$$

Este procedimento é executado para todas as outras entradas em relação à SE e todas as entradas em relação à saída IN.

O passo seguinte é obter as relações inibitórias, através da rede neural auto-associativa Regra de Hebb (Apêndice C). Conforme já observado, as relações inibitórias são obtidas negando-se as entradas (1,0 - ValorEntrada). Note que, as saídas permanecem inalteradas. Logo, o conjunto de dados apresentado a esta rede neural é obtido como mostrado abaixo:

CONJUNTO DE TREINAMENTO					
ENTRADAS				SAÍDAS	
1,00	0,00	0,00	1,00	1	0
0,78	0,09	0,17	0,75	1	0
0,52	0,36	0,31	0,54	1	0
0,47	0,57	0,52	0,46	0	1
0,14	0,73	0,63	0,13	0	1
0,00	1,00	1,00	0,00	0	1

Esta rede neural apresenta os seguintes parâmetros: coeficiente de aprendizagem = 0,05 e taxa de decaimento = 0,02 (Apêndice A) e nº de iterações = 1000. Ao final do treinamento, a matriz de pesos (WH) das relações inibitórias é armazenada.

Um vez obtido os SSEs e os pesos da matriz da RNA com uma camada, pode-se proceder o cálculo do produto para cada entrada em relação a cada saída. Isto é feito utilizando-se a Equação D.37. Abaixo, é apresentado o exemplo para o cálculo do produto da entrada PA1 em relação à saída SE.

$$Prod_{PA1-SE} = WH_{PA1-SE} * SSE_{PA1-SE} = 0,6500316 * 11,091 = 7,2095$$

Observando-se o valor do peso da conexão inibitória, nota-se que o valor mais próximo de 1 que 0. Isto reforça a correlação muito fraca desta entrada com esta saída. O resultado final de todos os produtos em relação a cada saída, ordenados em ordem crescente de valores, é mostrado abaixo.

ENTRADA	PRODUTO SAÍDA 1 (SE)		ENTRADA	PRODUTO SAÍDA 2 (IN)
V1	0,7781		PA2	2,1373
V2	1,3876		PA1	2,2690
PA1	7,2097		V2	8,3652
PA2	7,6781		V1	13,1107

Observando-se os valores encontrados para os produtos, nota-se que nenhum apresenta produto zero. Isto determina a escolha de somente um atributo inicialmente para compor a regra de definição para cada saída. Desta forma, V1 é o atributo inicial escolhido para SE e PA2 é o atributo escolhido para IN. Duas regras então são formadas para cada saída:

```

Regra_1SE: (V1)  $\Rightarrow$  SE
Regra_2SE: (NãoV1)  $\Rightarrow$  SE
Regra_1IN: (PA2)  $\Rightarrow$  IN
Regra_2IN: (NãoPA2)  $\Rightarrow$  IN

```

Neste ponto, é necessário a verificação da cobertura dos exemplares por uma dada regra (Passo C.1). Deve-se então calcular a SumDif (Equação D.38) para cada exemplar do conjunto de treinamento em relação a regra. Os resultados são apresentados pela Tabela D.3.

Assumindo um valor limite $t = 0,20$, classifica-se então os exemplares cobertos por cada regra (em destaque sublinhado simples no quadro). Desta forma, a Regra_1SE cobre os três exemplares e é uma regra final. Seu valor de SumDif é menor que 0,20 para os três primeiros exemplares, cuja saída SE é igual a 1. Verificando-se as outras regras, Regra_2SE e Regra_2IN não cobrem qualquer exemplar, não sendo consideradas regras válidas. A Regra_1IN cobre apenas os exemplares 5 e 6. Observe que, com um pequeno aumento no valor de t (Passo C.4) para 0,25, pode-se cobrir o exemplar 4 por esta Regra_1IN. Entretanto, como é o primeiro atributo sendo selecionado, a opção de não se alterar t , permitindo que a regra possa ser mais especializada. Esta opção é introduzida na implementação do algoritmo, mostrada no Capítulo 04 desta proposta. Assim, mantendo t constante neste ponto, esta regra é válida mas necessita ser

expandida, pois não classifica totalmente os exemplares. Note que, o exemplar 4 em destaque, não é coberto por esta regra. Assim, as duas regras válidas são:

Regra_1SE: (V1) \Rightarrow SE
 Regra_1IN: (PA2) \Rightarrow IN

Tabela D.3 - Resultados do primeiro teste

EX.	ATRI- BUTO	Regra_ 1SE	Regra_ 2SE	SAÍDA SE	ATRI- BUTO	Regra_ 1IN	Regra_ 2IN	SAÍDA IN
	(V1)	SumDif (1,0- Vatr) ²	SumDif (0,0- Vatr) ²		(PA2)	SumDif (1,0- Vatr) ²	SumDif (0,0- Vatr) ²	
1	1,00	<u>0,0</u>	1,00	<u>1</u>	0,00	1,00	0,00	0
2	0,91	<u>0,0081</u>	0,8281	<u>1</u>	0,25	0,5625	0,0625	0
3	0,64	<u>0,1296</u>	0,4096	<u>1</u>	0,46	0,2916	0,2116	0
4	0,43	0,3249	0,1849	0	0,54	<u>0,2916</u>	0,2916	<u>1</u>
5	0,22	0,6084	0,0484	0	0,87	<u>0,0169</u>	0,7569	<u>1</u>
6	0,00	1,00	0,00	0	1,00	<u>0,00</u>	1,00	<u>1</u>

Uma vez que manipulando dados contínuos, deve-se então determinar os intervalos que compõe cada cláusula das regras. Isto é feito diretamente no conjunto de treinamento. Então as regras válidas ficam da seguinte forma:

Regra_1SE: (0,64 \leq V1 \leq 1,00) \Rightarrow SE
 Regra_1IN: (0,87 \leq PA1 \leq 1,00) \Rightarrow IN

O passo seguinte é determinar se a critério da unicidade da regra. Para a faixa de valores do atributo V1, verifica-se que a Regra_1SE não classifica qualquer exemplar indicado pela saída IN, ou seja, todos exemplares cujo valor de IN é igual a 1 apresentam faixas de valores de V1 diferentes da definida na regra. O mesmo se verifica para a Regra_1IN em relação à PA1. Logo, o critério da unicidade é satisfeito.

Uma vez que, a Regra_1IN deve ser expandida, um novo conjunto de treinamento deve ser montado, agora, com as regras válidas como saídas e, conseqüentemente, como entradas adicionais. O novo conjunto é mostrado abaixo.

NOVO CONJUNTO DE TREINAMENTO							
ENTRADAS ORIGINAIS				Regra_1S	Regra_1	Regra_1	Regra_1
0,00	1,00	1,00	0,00	1	0	1	0
0,22	0,91	0,83	0,25	1	0	1	0
0,48	0,64	0,69	0,46	1	0	1	0
0,53	0,43	0,48	0,54	0	0	0	0
0,86	0,27	0,37	0,87	0	1	0	1
1,00	0,00	0,00	1,00	0	1	0	1

Observe que, o valor da Regra_1IN para o exemplar 4 é zero, decorrente da não cobertura deste exemplar pela regra. O processo de gerar novos produtos é então desencadeado. O treinamento da RNA-ML é feito, utilizando-se apenas o aprendizado normal, com 50.000 iterações. Da nova matriz de pesos entrada-camada interna armazenada, os novos SSEs de cada entrada em relação a cada regra gerada (novas saídas) são calculados. Para obtenção do nova relação inibitória, um novo conjunto de treinamento para a RNA com uma camada é também montado, conforme mostrado abaixo.

CONJUNTO DE TREINAMENTO					
ENTRADAS				SAÍDAS	
1,00	0,00	0,00	1,00	1	0
0,78	0,09	0,17	0,75	1	0
0,52	0,36	0,31	0,54	1	0
0,47	0,57	0,52	0,46	0	0
0,14	0,73	0,63	0,13	0	1
0,00	1,00	1,00	0,00	0	1

Observe que, aqui também o exemplar 4, para a Regra_1IN é zero. Uma vez armazenado sua respectiva matriz de pesos entrada-saída, os novos produtos são então processados. Os valores encontrados, colocados em ordem crescente são:

ENTRADA	PRODUTO SAÍDA 1 (SE)		ENTRADA	PRODUTO SAÍDA 2 (IN)
V1	1,0096		PA2	1,1144
V2	1,8004		PA1	1,2041
PA1	9,3550		V2	8,8807
PA2	9,9627		V1	13,8400

De acordo com os produtos obtidos, os novos atributos selecionados não inclusos nas regras são: para saída SE, o atributo é V2 e, para saída IN, o atributo é PA1. As novas regras

formadas a serem testadas são:

```

Regra_11SE: (V1 AND V2) ⇒ SE
Regra_12SE: (V1 AND NãoV2) ⇒ SE
Regra_11IN: (PA2 AND PA1) ⇒ IN
Regra_12IN: (PA2 AND NãoPA1) ⇒ IN

```

Uma vez obtidas as novas regras, deve-se então testar a cobertura dos exemplares. Observe que, anteriormente, a Regra_1SE cobre todos os exemplares e é considerada uma regra final. Aqui, ela está sendo expandida por opção de especializá-la mais, uma vez que, regras especializadas se sobrepõe a conceitos mais gerais. O novo resultado é mostrado pela Tabela D.3.

Mais uma vez, o valor limite inicialmente assumido é $t=0,2$. Observando-se os valores obtidos, nota-se que, as Regra_12SE e Regra_12IN não classificam qualquer exemplar. A inclusão do atributo V2 à Regra_1SE anterior, formando a Regra_11SE, classificou menos exemplares, conforme destacado no quadro de resultados. Isto não é plausível, pois a inclusão de um atributo deve acrescentar especialização a uma regra e conseqüentemente classificar, no mínimo, o mesmo número de exemplares. Assim, o valor limite t pode ser redefinido para $t=0,3$ e desta forma classificar todos os exemplares, em relação à saída SE. Em relação à Regra_11IN formada para a saída IN, nota-se que a inclusão do atributo PA1 não classifica maior número de exemplares. Neste caso, duas alternativas são apresentadas: a primeira é mover o valor limite até que todos os exemplares sejam classificados ou, especializar novamente a regra, obtendo um novo atributo mais discriminatório. A segunda alternativa gera maior esforço computacional e a inclusão de um novo atributo discriminatório aumenta o valor de SumDif para este exemplar não classificado. Admitindo-se que é possível mover o valor limite até 2,5 vezes o seu valor inicial (neste caso, a natureza do problema é que determina esta possibilidade), então $t=0,5$ pode ser assumido e assim todos exemplares classificados. As regras válidas serão:

```

Regra_11SE: (V1 AND V2) ⇒ SE
Regra_11SO: (PA2 AND PA1) ⇒ IN

```

Acessando o conjunto de treinamento, os valores máximos e mínimos de cada atributo são então encontrados para a formação das cláusulas. Logo, as regras finais que compõe a **Base de Conhecimento** em relação a este problema são:

```

((0,64 ≤ V1 ≤ 1,00) AND (0,69 ≤ V2 ≤ 1,00)) ⇒ SE
((0,54 ≤ PA2 ≤ 1,00) AND (0,53 ≤ PA1 ≤ 1,00)) ⇒ IN

```

Vale salientar que, neste exemplo, não foram obtidas regras disjuntivas, uma vez que,

apenas uma regra é formada para cada saída. Verificando a capacidade de generalização das regras formadas, o conjunto de teste montado anteriormente foi utilizado para testá-las. Assumindo $\theta = 0$ inicialmente, o resultado mostra que os exemplares presentes neste conjunto de teste foram cobertos pelas regras formadas com este mesmo valor de θ . Assim, não há necessidade de alterar θ para estas regras aumentem suas capacidades de generalização.

Tabela D.4 - Resultados para o segundo teste

EX.	ATRI- BUTOS	Regra_ 11SE	Regra_ 12SE	SAÍDA SE	ATRI- BUTOS (PA2) e (PA1)	Regra_ 11IN	Regra_ 12IN	SAÍDA IN
	(V1) e (V2)	SumDif	SumDif		SumDif	SumDif		
1	1,00/ 1,00	<u>0,0</u>	1,00	<u>1</u>	0,00/ 0,00	2,00	1,00	0
2	0,91/ 0,83	<u>0,037</u>	0,697	<u>1</u>	0,25/ 0,22	1,1709	0,6109	0
3	0,64/ 0,69	<u>0,2257</u>	0,6057	<u>1</u>	0,46/ 0,48	0,5620	0,5220	0
4	0,43/ 0,48	0,5953	0,5553	0	0,54/ 0,53	<u>0,4325</u>	0,4925	<u>1</u>
5	0,22/ 0,37	1,0053	0,7453	0	0,87/ 0,86	<u>0,0365</u>	0,7565	<u>1</u>
6	0,00/ 0,00	1,00	1,00	0	1,00/ 1,00	<u>0,00</u>	1,00	<u>1</u>

Explorando fisicamente o problema, nota-se que os atributos salientes em cada regra (que compõe o antecedente da regra) mapeiam o problema da seguinte forma: as tensões nas barras 2 e 3 são as características salientes da região de operação indicada como segura e o carregamento (potências ativas das barras 2 e 3) são as características salientes da região de operação indicada como insegura.

Apêndice E

REGRAS DE PRODUÇÃO – C4.5

E.1 INTRODUÇÃO

Este apêndice mostra as regras de produção extraídas pelo C4.5 para os sistemas testes do IEEE 14 barras e 118 barras

E.2 SISTEMA ELÉTRICO TESTE IEEE 14 BARRAS

As regras de produção extraídas do C4.5, mostradas neste apêndice, são referentes às seguintes contingências do sistema elétrico teste IEEE 14 barras:

1. Perda do compensador síncrono na barra 6
 2. Perda do capacitor na barra 9
 3. Perda da linha de transmissão - barras 02 e 04
 4. Perda da linha de transmissão - barras 04 e 05
 5. Perda da linha de transmissão - barras 06 e 11
 6. Perda da linha de transmissão - barras 06 e 12
 7. Perda da linha de transmissão - barras 06 e 13
 8. Perda da linha de transmissão - barras 09 e 10
 9. Perda do gerador na barra 2
-

Perda do compensador síncrono barra 6 – Caso 1

Regra 1:
V12 > 0.947
-> class normal

Regra 2:
V12 <= 0.947
V10 > 0.954
-> class alerta

Regra 3:
V12 > 0.947
V12 <= 0.958
ZI10 <= 6.739
-> class alerta

Regra 4:
V14 > 0.896
V12 <= 0.947
ZI12 <= 11.02
-> class alerta

Regra 5:
V12 > 0.947
V12 <= 0.958
ZI12 > 11.81
-> class alerta

Regra 6:
V10 <= 0.954
-> class inseguro

Perda do capacitor – barra 9 – Caso 2

Regra 1:
V12 > 0.922
-> classe normal

Regra 2:
V12 > 0.872
V12 <= 0.922
-> classe alerta

Regra 3:
V12 <= 0.872
-> classe inseguro

Regra 4:
V10 <= 0.895
ZI12 > 9.358
ZI11 <= 16.635
-> classe inseguro

Perda da LT – barras 02 e 04 – Caso 3

Regra 1:
V12 > 0.925
-> classe normal

Regra 2:
V10 > 0.915
ZI11 <= 18.361
ZI12 <= 11.547
-> classe normal

Regra 3:
V10 > 0.905
V10 <= 0.915
-> classe alerta

Regra 4
V10 > 0.885
ZI13 <= 4.261
-> classe alerta

Regra 5:
V10 > 0.915
V12 <= 0.925
-> classe alerta

Regra 6:
V10 > 0.885
ZI14 <= 3.846
-> classe alerta

Regra 7:
V10 <= 0.915
-> classe inseguro

Perda da LT – barras 04 e 05 – Caso 4

Regra 1:
V14 > 0.923
-> classe normal

Regra 2:
V14 > 0.895
V12 <= 0.95
-> classe alerta

Regra 3:
V10 <= 0.955
V12 > 0.95
ZI10 > 7.243
-> classe alerta

Regra 4:
V10 <= 0.955
V12 > 0.95
-> classe alerta

Regra 5:
V14 <= 0.895
-> classe inseguro

Perda da LT – barras 06 e 11 – Caso 5

Regra 1:
V14 > 0.943
-> classe normal

Regra 2:
V14 <= 0.943
V12 > 0.955
-> classe alerta

Regra 3:
V12 > 0.943
V12 <= 0.955
-> classe alerta

Regra 4:
V12 <= 0.943
-> classe inseguro

Perda da LT – barras 06 e 12 – Caso 6

Regra 1:
V14 > 0.926
-> classe normal

Regra 2:
V13 <= 0.913
V11 > 0.931
-> classe alerta

Regra 3:
V14 <= 0.926
V13 > 0.917
-> classe alerta

Regra 4:
V13 <= 0.917
-> classe inseguro

Perda da LT – barras 06 e 13 – Caso 7

Regra 1:
V10 > 0.963
-> classe normal

Regra 2:
V11 > 0.924
V11 <= 0.954
-> classe alerta

Regra 3:
V10 <= 0.963
V11 > 0.954
-> classe alerta

Regra 4:
V11 <= 0.924
-> classe inseguro

Perda da LT – barras 09 e 10 – Caso 8

Regra 1:
V12 > 0.914
-> classe normal

Regra 2:
V12 <= 0.914
V14 > 0.897
-> classe alerta

Regra 3:
V14 > 0.87
ZI14 <= 3.73
-> classe alerta

Regra 4:
V14 <= 0.87
-> classe inseguro

Regra 5:
V14 <= 0.897
ZI14 > 3.73
-> classe inseguro [0.759]

Perda Gerador – barra 2 – Caso 9

Regra 1:
V10 > 0.961
-> classe normal

Regra 2:
V12 > 0.911
ZI12 <= 9.985
-> classe alerta

Regra 3:
V10 > 0.939
V10 <= 0.961
-> classe alerta

Regra 4:
V12 > 0.911
V10 <= 0.961
ZI13 <= 4.313
-> classe alerta

Regra 5:
V14 <= 0.893
V12 > 0.902
-> classe alerta

Regra 6:
V10 <= 0.939
-> classe inseguro

E.3 SISTEMA ELÉTRICO TESTE IEEE 118 BARRAS

As regras de produção extraídas do C4.5, mostradas neste apêndice, são referentes às seguintes contingências do sistema elétrico teste IEEE 118 barras:

1. Caso Base
2. Perda do compensador síncrono barra
3. Perda da linha de transmissão - barras 76 e 77

Caso Base

Regra 1:
V52 > 0.907
-> classe normal [0.807]

Regra 2: V39 <= 0.963
V52 > 0.907
V52 <= 0.91
ZI109 <= 8.704
-> classe alerta [0.941]

Regra 3: V39 > 0.958
V39 <= 0.96
V52 > 0.902
V109 > 0.93
-> classe alerta [0.875]

Regra 4:
V39 <= 0.963
V52 > 0.907
V52 <= 0.91
ZI39 > 2.677
-> classe alerta [0.826]

Regra 5:
V39 > 0.959
V52 > 0.902
V52 <= 0.907
-> classe alerta [0.779]

Regra 6:
V39 <= 0.96
V52 <= 0.907
V109 <= 0.93
ZI52 > 3.813
-> classe inseguro [0.822]

Regra 7:
V52 <= 0.907
-> classe inseguro [0.558]

Perda Gerador – barra 2 – Caso 9

Regra 1:
V10 > 0.961
-> classe normal

Regra 2:
V12 > 0.911
ZI12 <= 9.985
-> classe alerta

Regra 3:
V10 > 0.939
V10 <= 0.961
-> classe alerta

Regra 4:
V12 > 0.911
V10 <= 0.961
ZI13 <= 4.313
-> classe alerta

Regra 5:
V14 <= 0.893
V12 > 0.902
-> classe alerta

Regra 6:
V10 <= 0.939
-> classe inseguro

Perda Gerador – barra 2 – Caso 9

Regra 1:
V10 > 0.961
-> classe normal

Regra 2:
V12 > 0.911
ZI12 <= 9.985
-> classe alerta

Regra 3:
V10 > 0.939
V10 <= 0.961
-> classe alerta

Regra 4:
V12 > 0.911
V10 <= 0.961
ZI13 <= 4.313
-> classe alerta

Regra 5:
V14 <= 0.893
V12 > 0.902
-> classe alerta

Regra 6:
V10 <= 0.939
-> classe inseguro

Apêndice F

REGRAS DE PRODUÇÃO - VIA

F.1 INTRODUÇÃO

Este apêndice mostra as regras de produção extraídas pelo VIA para os sistemas testes do IEEE 14 barras e 118 barras. São apresentadas apenas as regras finais, decorrentes do segundo processamento do VIA para o caso em estudo.

F.2 SISTEMA ELÉTRICO TESTE IEEE 14 BARRAS

As regras de produção extraídas do VIA, mostradas neste apêndice, são referentes às seguintes contingências do sistema elétrico teste IEEE 14 barras:

1. Perda do compensador síncrono na barra 6
 2. Perda do capacitor na barra 9
 3. Perda da linha de transmissão - barras 02 e 04
 4. Perda da linha de transmissão - barras 04 e 05
 5. Perda da linha de transmissão - barras 06 e 11
 6. Perda da linha de transmissão - barras 06 e 12
 7. Perda da linha de transmissão - barras 06 e 13
 8. Perda da linha de transmissão - barras 09 e 10
 9. Perda do gerador na barra 2
-

Perda do compensador síncrono barra 6 – Caso 1

REGRAS INICIAIS E FINAIS

REGRA NUM. == 1 ----> CLASSE 1							

0.8710	V12	1.0250		0.8942	V12	1.0250	15.06
0.9340	V14	1.0030		0.9340	V14	1.0030	0.00
0.9470	V10	1.0170		0.9470	V10	1.0170	0.00

REGRA NUM. == 2 ----> CLASSE 2							

0.8788	V14	0.9820		0.8792	V14	0.9820	0.37
0.9030	V12	0.9710		0.9030	V12	0.9710	0.00
0.9540	V10	0.9942		0.9540	V10	0.9942	0.00

REGRA NUM. == 3 ----> CLASSE 2							

0.9210	V10	0.9890		0.9324	V10	0.9890	16.83
0.8936	V14	0.9831		0.8936	V14	0.9831	0.03
0.9470	V12	0.9580		0.9470	V12	0.9580	0.00
3.3070	ZI10	6.7390		3.3070	ZI10	6.7390	0.00

REGRA NUM. == 4 ----> CLASSE 2							

0.8949	V10	0.9942		0.8994	V10	0.9942	4.51
0.8960	V14	0.9820		0.8960	V14	0.9820	0.00
0.9030	V12	0.9710		0.9030	V12	0.9710	0.00
5.3380	ZI12	11.0200		5.3380	ZI12	11.0200	0.00

REGRA NUM. == 5 ----> CLASSE 2							

0.9210	V10	0.9890		0.9332	V10	0.9890	17.89
0.8936	V14	0.9872		0.8936	V14	0.9863	1.03
0.9470	V12	0.9580		0.9470	V12	0.9580	0.00
11.8100	ZI12	18.7340		11.8100	ZI12	18.7340	0.00

REGRA NUM. == 6 ----> CLASSE 3							

0.7430	V10	0.9600		0.7430	V10	0.9600	0.00

Perda do capacitor - barra 9 – Caso 2

REGRAS INICIAIS E FINAIS

REGRA NUM. == 1 ----> CLASSE 1							

0.7953	V14	0.9810		0.8223	V14	0.9810	14.52
0.9220	V12	1.0010		0.9220	V12	1.0010	0.00
0.9080	V10	0.9950		0.9080	V10	0.9950	0.00

REGRA NUM. == 2 ----> CLASSE 2							

0.7556	V14	0.9810		0.8014	V14	0.9662	26.86
0.8720	V12	0.9220		0.8720	V12	0.9220	0.00
0.8790	V10	0.9380		0.8790	V10	0.9380	0.00

REGRA NUM. == 3 ----> CLASSE 3							

0.6790	V12	0.8720		0.6790	V12	0.8720	0.00
0.6890	V10	0.9160		0.6890	V10	0.9160	0.00

REGRA NUM. == 4 ----> CLASSE 3							

0.6790	V12	0.9100		0.6907	V12	0.9100	5.07
0.6440	V14	0.9644		0.6440	V14	0.9644	0.00
0.6890	V10	0.8950		0.6890	V10	0.8950	0.00
9.3580	ZI12	18.3190		9.3580	ZI12	18.3190	0.00
7.9390	ZI10	16.6350		7.9390	ZI10	16.6350	0.00

Perda da LT – barras 02 e 04 – Caso 3

##### REGRAS INICIAIS E FINAIS #####							
REGRA NUM. == 1 ---->				CLASSE 1			
0.7050	V14	1.0360		0.8094	V14	1.0360	31.54
0.9160	V10	1.0490		0.9160	V10	1.0490	0.00
0.9250	V12	1.0550		0.9250	V12	1.0550	0.00
REGRA NUM. == 2 ---->				CLASSE 1			
0.8920	V14	1.0360		0.8920	V14	1.0360	0.00
0.9150	V10	1.0389		0.9150	V10	1.0389	0.00
0.8049	V12	1.0550		0.8049	V12	1.0550	0.00
10.2970	ZI11	18.3610		10.2970	ZI11	18.3610	0.00
6.2770	ZI12	11.5470		6.2770	ZI12	11.5470	0.00
REGRA NUM. == 3 ---->				CLASSE 2			
0.7978	V12	1.0272		0.7978	V12	1.0265	0.32
0.8560	V14	0.9290		0.8560	V14	0.9290	0.00
0.9050	V10	0.9150		0.9050	V10	0.9150	0.00
REGRA NUM. == 4 ---->				CLASSE 2			
0.8850	V10	1.0298		0.8850	V10	0.9664	43.79
0.7829	V12	1.0547		0.7829	V12	1.0547	0.02
0.8560	V14	0.9290		0.8560	V14	0.9290	0.00
2.6290	ZI13	4.2610		2.6290	ZI13	4.2610	0.00
REGRA NUM. == 5 ---->				CLASSE 2			
0.9150	V10	1.0217		0.9150	V10	0.9577	59.96
0.8560	V14	0.9290		0.8602	V14	0.9290	5.77
0.8052	V12	0.9250		0.8052	V12	0.9250	0.03
REGRA NUM. == 6 ---->				CLASSE 2			
0.8850	V10	1.0357		0.8850	V10	0.9722	42.13
0.7829	V12	1.0550		0.7829	V12	1.0550	0.00
0.8560	V14	0.9290		0.8560	V14	0.9290	0.00
2.3030	ZI14	3.8460		2.3030	ZI14	3.8460	0.00
REGRA NUM. == 7 ---->				CLASSE 3			
0.7050	V14	0.8960		0.7050	V14	0.8960	0.00
0.7400	V10	0.9150		0.7400	V10	0.9150	0.00

Perda da LT – barras 04 e 05 – Caso 4

REGRAS INICIAIS E FINAIS

REGRA NUM. == 1 ----> CLASSE 1								

0.9230	V14	1.0330		0.9230	V14	1.0330		0.00
0.9310	V10	1.0480		0.9310	V10	1.0480		0.00
0.9128	V12	1.0570		0.9128	V12	1.0570		0.00

REGRA NUM. == 2 ----> CLASSE 2								

0.9210	V12	1.0530		0.9210	V12	1.0168		27.45
0.8740	V10	0.9748		0.8742	V10	0.9748		0.18
0.8950	V14	0.9503		0.8950	V14	0.9503		0.00

REGRA NUM. == 3 ----> CLASSE 2								

0.9030	V10	1.0370		0.9038	V10	1.0232		10.91
0.8905	V14	0.9680		0.8905	V14	0.9680		0.02
0.9500	V12	1.0034		0.9500	V12	1.0034		0.00
7.2430	ZI10	11.7990		7.2430	ZI10	11.7990		0.00

REGRA NUM. == 4 ----> CLASSE 2								

0.8810	V14	1.0220		0.8872	V14	0.9759		37.04
0.8948	V10	0.9550		0.8948	V10	0.9550		0.04
0.9500	V12	1.0118		0.9500	V12	1.0118		0.00

REGRA NUM. == 5 ----> CLASSE 3								

0.7810	V12	0.9783		0.7810	V12	0.9783		0.01
0.7230	V14	0.8950		0.7230	V14	0.8950		0.00
0.7480	V10	0.9330		0.7480	V10	0.9330		0.00

Perda da LT – barras 06 e 11 – Caso 5

REGRAS INICIAIS E FINAIS

REGRA NUM. == 1 ----> CLASSE 1								

0.9430	V14	1.0370		0.9430	V14	1.0370		0.00
0.9420	V11	1.0430		0.9420	V11	1.0430		0.00
0.9024	V12	1.0560		0.9024	V12	1.0560		0.00

REGRA NUM. == 2 ----> CLASSE 2								

0.8838	V11	0.9983		0.8845	V11	0.9983		0.60
0.8910	V14	1.0230		0.8910	V14	1.0230		0.00
0.9550	V12	1.0256		0.9550	V12	1.0256		0.00

REGRA NUM. == 3 ----> CLASSE 2								

0.8910	V14	1.0230		0.8910	V14	0.9800		32.58
0.8769	V11	0.9986		0.8789	V11	0.9986		1.68
0.9430	V12	0.9550		0.9430	V12	0.9550		0.00

REGRA NUM. == 4 ----> CLASSE 3								

0.6740	V11	0.9777		0.6740	V11	0.9744		1.08
0.6640	V14	0.9330		0.6640	V14	0.9330		0.00
0.7160	V12	0.9430		0.7160	V12	0.9430		0.00

Perda da LT – barras 06 e 12 – Caso 6

REGRAS INICIAIS E FINAIS

REGRA NUM. == 1 ----> CLASSE 1							

0.8745	V13	1.0450		0.8800	V13	1.0450	3.23
0.9260	V14	1.0360		0.9260	V14	1.0360	0.00
0.9370	V10	1.0530		0.9370	V10	1.0530	0.00

REGRA NUM. == 2 ----> CLASSE 2							

0.9120	V10	0.9690		0.9120	V10	0.9662	4.91
0.8612	V14	0.9445		0.8612	V14	0.9445	0.00
0.8706	V13	0.9130		0.8706	V13	0.9130	0.00

REGRA NUM. == 3 ----> CLASSE 2							

0.8010	V10	0.9500		0.9004	V10	0.9500	66.70
0.9071	V13	0.9907		0.9071	V13	0.9720	22.40
0.8820	V14	0.9450		0.8820	V14	0.9450	0.00

REGRA NUM. == 4 ----> CLASSE 3							

0.7630	V14	0.9475		0.7630	V14	0.9475	0.00
0.7750	V13	0.9170		0.7750	V13	0.9170	0.00
0.8010	V10	0.9630		0.8010	V10	0.9630	0.00

Perda da LT – barras 06 e 13 – Caso 7

REGRAS INICIAIS E FINAIS

REGRA NUM. == 1 ----> CLASSE 1							

0.7820	V13	1.0080		0.7820	V13	1.0080	0.02
0.9630	V10	1.0500		0.9630	V10	1.0500	0.00
0.9510	V11	1.0570		0.9510	V11	1.0570	0.00

REGRA NUM. == 2 ----> CLASSE 2							

0.9130	V10	1.0380		0.9130	V10	0.9766	49.09
0.7849	V13	0.9437		0.7849	V13	0.9437	0.00
0.9240	V11	0.9540		0.9240	V11	0.9540	0.00

REGRA NUM. == 3 ----> CLASSE 2							

0.9130	V10	1.0380		0.9221	V10	1.0174	23.81
0.8171	V13	0.9683		0.8171	V13	0.9683	0.00
0.9540	V11	1.0026		0.9540	V11	1.0026	0.00

REGRA NUM. == 4 ----> CLASSE 3							

0.7170	V10	0.9460		0.7170	V10	0.9382	3.39
0.5890	V13	0.9200		0.5890	V13	0.9200	0.01
0.7310	V11	0.9240		0.7310	V11	0.9240	0.00

Perda da LT – barras 09 e 10 – Caso 8

REGRAS INICIAIS E FINAIS

REGRA NUM. == 1 ----> CLASSE 1							

0.9140	V12	1.0560		0.9140	V12	1.0560	0.00
0.9020	V14	1.0410		0.9020	V14	1.0410	0.00

REGRA NUM. == 2 ----> CLASSE 2							

0.7068	V10	1.0300		0.7068	V10	1.0300	0.00
0.8610	V12	0.9390		0.8610	V12	0.9390	0.00
0.8970	V14	0.9829		0.8970	V14	0.9829	0.00

REGRA NUM. == 3 ----> CLASSE 2							

0.8142	V12	1.0464		0.8144	V12	1.0445	0.93
0.8710	V14	0.9430		0.8710	V14	0.9430	0.00
2.5540	ZI14	3.7300		2.5540	ZI14	3.7300	0.00

REGRA NUM. == 4 ----> CLASSE 3							

0.7680	V14	0.8700		0.7680	V14	0.8700	0.00
0.7640	V12	0.9090		0.7640	V12	0.9090	0.00

REGRA NUM. == 5 ----> CLASSE 3							

0.7640	V12	0.9090		0.7718	V12	0.9090	5.37
0.7811	V14	0.8970		0.7811	V14	0.8970	0.00
3.7300	ZI14	7.5090		3.7300	ZI14	7.5090	0.00

Perda do gerador barra 2 – Caso 9

REGRAS INICIAIS E FINAIS

REGRA NUM. == 1 ----> CLASSE 1							

0.7719	V12	1.0540		0.8144	V12	1.0540	15.06
0.9200	V14	1.0180		0.9200	V14	1.0180	0.00
0.9610	V10	1.0280		0.9610	V10	1.0280	0.00

REGRA NUM. == 2 ----> CLASSE 2							

0.8159	V14	1.0180		0.8460	V14	1.0120	17.87
0.9110	V12	1.0540		0.9110	V12	1.0410	9.11
0.9130	V10	0.9770		0.9130	V10	0.9770	0.00
5.8270	ZI12	9.9850		5.8270	ZI12	9.9850	0.00

REGRA NUM. == 3 ----> CLASSE 2							

0.8312	V12	1.0278		0.8332	V12	1.0242	2.82
0.8810	V14	0.9520		0.8810	V14	0.9520	0.00
0.9390	V10	0.9610		0.9390	V10	0.9610	0.00

REGRA NUM. == 4 ----> CLASSE 2							

0.8177	V14	0.9904		0.8423	V14	0.9904	14.25
0.9110	V12	1.0255		0.9110	V12	1.0255	0.00
0.9130	V10	0.9770		0.9130	V10	0.9770	0.00
2.4480	ZI13	4.3130		2.4480	ZI13	4.3130	0.00

REGRA NUM. == 5 ----> CLASSE 2							

0.8083	V14	0.8930		0.8375	V14	0.8930	34.52
0.9130	V10	0.9770		0.9130	V10	0.9739	4.91
0.9020	V12	0.9818		0.9020	V12	0.9818	0.00

REGRA NUM. == 6 ----> CLASSE 3							

0.7300	V14	0.9995		0.7300	V14	0.9773	8.24
0.7540	V12	0.9240		0.7540	V12	0.9240	0.00
0.7730	V10	0.9390		0.7730	V10	0.9390	0.00

F.3 SISTEMA ELÉTRICO TESTE IEEE 118 BARRAS

As regras de produção extraídas do VIA, mostradas neste apêndice, são referentes às seguintes contingências do sistema elétrico teste IEEE 118 barras:

1. Caso Base
2. Perda do compensador síncrono barra
3. Perda da linha de transmissão - barras 76 e 77

Caso Base

REGRAS INICIAIS E FINAIS

REGRA NUM. == 1 ---->				CLASSE 1			

0.9060	V52	0.9220		0.9060	V52	0.9220	0.00

REGRA NUM. == 2 ---->				CLASSE 2			

0.9110	V109	0.9670		0.9110	V109	0.9556	20.31
0.9070	V52	0.9100		0.9070	V52	0.9100	0.00
0.9580	V39	0.9640		0.9580	V39	0.9640	0.00
7.0390	ZI109	8.7040		7.0390	ZI109	8.7040	0.00

REGRA NUM. == 3 ---->				CLASSE 2			

0.9580	V39	0.9600		0.9580	V39	0.9600	0.00
0.9020	V52	0.9220		0.9020	V52	0.9220	0.00
0.9200	V109	0.9380		0.9200	V109	0.9380	0.00

REGRA NUM. == 4 ---->				CLASSE 2			

0.9110	V109	0.9670		0.9110	V109	0.9556	20.31
0.9580	V39	0.9640		0.9580	V39	0.9640	0.00
0.9070	V52	0.9100		0.9070	V52	0.9100	0.00
2.6770	ZI39	3.6970		2.6770	ZI39	3.6970	0.00

REGRA NUM. == 5 ---->				CLASSE 2			

0.9590	V39	0.9710		0.9590	V39	0.9710	0.00
0.9020	V52	0.9070		0.9020	V52	0.9070	0.00
0.9200	V109	0.9380		0.9200	V109	0.9380	0.00

REGRA NUM. == 6 ---->				CLASSE 3			

0.8840	V52	0.9060		0.8848	V52	0.9060	3.80
0.9490	V39	0.9600		0.9490	V39	0.9600	0.00
0.9110	V109	0.9300		0.9110	V109	0.9300	0.00
3.8130	ZI52	5.2260		3.8130	ZI52	5.2260	0.00

REGRA NUM. == 7 ---->				CLASSE 3			

0.8840	V52	0.9060		0.8840	V52	0.9060	0.00

Perda do compensador síncrono – barra 76

REGRAS INICIAIS E FINAIS

REGRA NUM. == 1 ---->				CLASSE 1			

0.9300	V109	0.9650		0.9300	V109	0.9650	0.00

REGRA NUM. == 2 ---->				CLASSE 2			

0.8936	V53	0.9310		0.8936	V53	0.9310	0.00
0.9590	V39	0.9720		0.9590	V39	0.9720	0.00
0.9000	V52	0.9110		0.9000	V52	0.9110	0.00
0.9140	V109	0.9340		0.9140	V109	0.9340	0.00

REGRA NUM. == 3 ---->				CLASSE 2			

0.6800	V53	0.9310		0.8818	V53	0.9310	80.40
0.9520	V39	0.9590		0.9520	V39	0.9590	0.00
0.9000	V52	0.9110		0.9000	V52	0.9110	0.00
0.9140	V109	0.9340		0.9140	V109	0.9340	0.00

REGRA NUM. == 4 ---->				CLASSE 3			

0.6910	V52	0.9030		0.6910	V52	0.9030	0.00

Perda da LT 138 –barras 76 - 77

REGRAS INICIAIS E FINAIS

REGRA NUM. == 1 ---->				CLASSE 1			

0.9740	V30	0.9870		0.9740	V30	0.9870	0.00
0.9950	V239	1.0010		0.9950	V239	1.0010	0.00

REGRA NUM. == 2 ---->				CLASSE 2			

0.9898	V239	0.9998		0.9898	V239	0.9998	0.27
0.9710	V30	0.9740		0.9710	V30	0.9740	0.00
0.9500	V7	0.9610		0.9500	V7	0.9610	0.00

REGRA NUM. == 3 ---->				CLASSE 2			

0.9907	V239	0.9998		0.9907	V239	0.9998	0.22
0.9700	V30	0.9760		0.9700	V30	0.9760	0.00
0.9370	V7	0.9731		0.9370	V7	0.9731	0.00
0.9520	V11	0.9860		0.9520	V11	0.9860	0.00

REGRA NUM. == 4 ---->				CLASSE 3			

0.9560	V30	0.9710		0.9560	V30	0.9710	0.00
0.9370	V7	0.9570		0.9370	V7	0.9570	0.00
0.9380	V11	0.9520		0.9380	V11	0.9520	0.00
0.9880	V239	0.9969		0.9880	V239	0.9969	0.00

Apêndice G

SISTEMAS ELÉTRICOS – DADOS

G.1 SISTEMA ELÉTRICO TESTE *IEEE* 14 BARRAS

O Apêndice G mostra o diagrama unifilar simplificado e os dados de entrada para o caso base selecionado do sistema teste IEEE 14 barras. Este é composto por 9 barras de carga (PQ), 4 barras com tensão constante, 17 linhas de transmissão e 3 transformadores. Apresenta 9 barras de carga. Um LTC está presente no transformador entre as barras 4 e 9, controlando a tensão na Barra 9 na faixa de 0.98 a 1,00 pu, com tolerância $\pm 0,001$ pu nos limites. Um capacitor de 19MVar está instalado na barra 9. Os dados completos referentes ao LTC encontram-se também neste mesmo apêndice. Nas simulações é previsto um gerador síncrono com potência ativa máxima 220,00 MW na barra 2. A potência base do sistema é 100MVA.

G.1.1 Dados

Tabela G.1 - DADOS DO SISTEMA

NBUS	NPV	NPQ	NLI	TOLIN	NITER
14	4	9	20	0.001	25

NBUS: Número de barras do sistema
NPV: Número de barras PV
NPQ: Número de barras PQ
NLI: Número de Linhas de Transmissão/Transformadores
TOLIN: Tolerância inicial
NITER: Número máximo de iterações

Tabela G.2 - DADOS DE BARRAS

NB	TP	V	ANG	PG	PL	QL	CAP	LMIN	LMAX
1	1	1.060	0.000	2.324	0.000	0.000	0.000	-999.0	999.0
2	2	1.045	-4.980	1.660	0.217	0.127	0.000	-0.400	0.500
3	2	1.010	-12.720	0.000	0.942	0.190	0.000	-999.0	0.400
4	3	1.019	-10.330	0.000	0.478	-0.039	0.000	-999.0	999.0
5	3	1.020	-8.780	0.000	0.076	0.016	0.000	-999.0	999.0
6	2	1.070	-14.220	0.000	0.112	0.075	0.000	-0.060	0.240
7	3	1.062	-13.370	0.000	0.000	0.000	0.000	-999.0	999.0
8	2	1.090	-13.360	0.000	0.000	0.000	0.000	-0.060	0.240
9	3	1.056	-14.940	0.000	0.295	0.166	0.190	-999.0	999.0
10	3	1.051	-15.100	0.000	0.090	0.058	0.000	-999.0	999.0
11	3	1.057	-14.790	0.000	0.035	0.018	0.000	-999.0	999.0
12	3	1.055	-15.070	0.000	0.061	0.016	0.000	-999.0	999.0
13	3	1.050	-15.160	0.000	0.135	0.058	0.000	-999.0	999.0
14	3	1.036	-16.040	0.000	0.149	0.050	0.000	-999.0	999.0

NB: Número da barra
 TP: Tipo da barra (1 - swing, 2 - PV, 3 - PQ)
 V: Tensão
 ANG: Ângulo
 PG: Potência Gerada
 PL: Potência Ativa da Carga
 QL: Potência Reativa da Carga
 CAP: Capacitor
 LMIN: Limite mínimo de fornecimento de potência reativa
 LMAX: Limite máximo de fornecimento de potência reativa

Tabela G.3 - DADOS DAS LINHAS DE TRANSMISSÃO E TRANSFORMADORES

NL	NE	NR	R	X	B	TAP1	TAP2	LTC
1	1	2	0.01938	0.05917	0.0528			
2	1	5	0.05403	0.22304	0.0492			
3	2	3	0.04699	0.19797	0.0438			
4	2	4	0.05811	0.17632	0.0374			
5	2	5	0.05695	0.17388	0.034			
6	3	4	0.06701	0.17103	0.0346			
7	4	5	0.01335	0.04211	0.0128			
8	4	7	0	0.20912	0	0.978		
9	4	9	0	0.55618	0	0.969		1
10	5	6	0	0.25202	0	0.932		
11	6	11	0.09498	0.1989	0			
12	6	12	0.12291	0.25581	0			
13	6	13	0.06615	0.13027	0			
14	7	8	0	0.17615	0			
15	7	9	0	0.11001	0			
16	9	10	0.03181	0.0845	0			
17	9	14	0.12711	0.27038	0			
18	10	11	0.08205	0.19207	0			
19	12	13	0.22092	0.19988	0			
20	13	14	0.17093	0.34802	0			

NL: Número da linha ou transformador
 NE: Número da barra emissora
 NR: Número da barra receptora
 R: Resistência
 X: Reatância
 B: Susceptância
 TAP1: Valor do tap no primário (não especificado = 1.0 pu)
 TAP2: Valor do tap no secundário (não especificado = 1.0 pu)
 LTC: Presença (1) ou não (0) de LTC

Tabela G.4 - DADOS DO LTC

NBR	NE	NR	VMIN	VMAX	TMIN	TMAX	STEP
9	9	2	1.01	1.03	.876	1.124	1.55

NBR: Número da barra a ser regulada pelo LTC
 VMIN: Tensão mínima da faixa de regulagem do LTC
 VMAX: Tensão máxima da faixa de regulagem do LTC
 TMIN: Tap mínimo
 TMAX: Tap máximo
 STEP: Valor do passo entre os taps

Tabela G. 5 - DADOS DA BARRA DO GERADOR

NB	NGER	FPART	PRIOR
2	4	100.00	1

NB: Número da barra do gerador
 NGER: Número de geradores na barra
 FPART: Fator de participação na potência ativa do sistema
 PRIOR: Nível de prioridade de atendimento do crescimento de carga
 (1 = maior nível)

Tabela G.6 - DADOS DOS GERADOR (Barra 2)

NG	PMIN	PMAX	FPARG
1	40,0	220,00	100,0

NG: Número do gerador
 PMIN: Potência ativa mínima (MW)
 PMAX: Potência ativa máxima (MW)
 FPARG: Fator de participação de cada gerador na potência ativa gerada pela barra (%)

G.1.2 Diagrama unifilar do sistema teste IEEE 14 barras

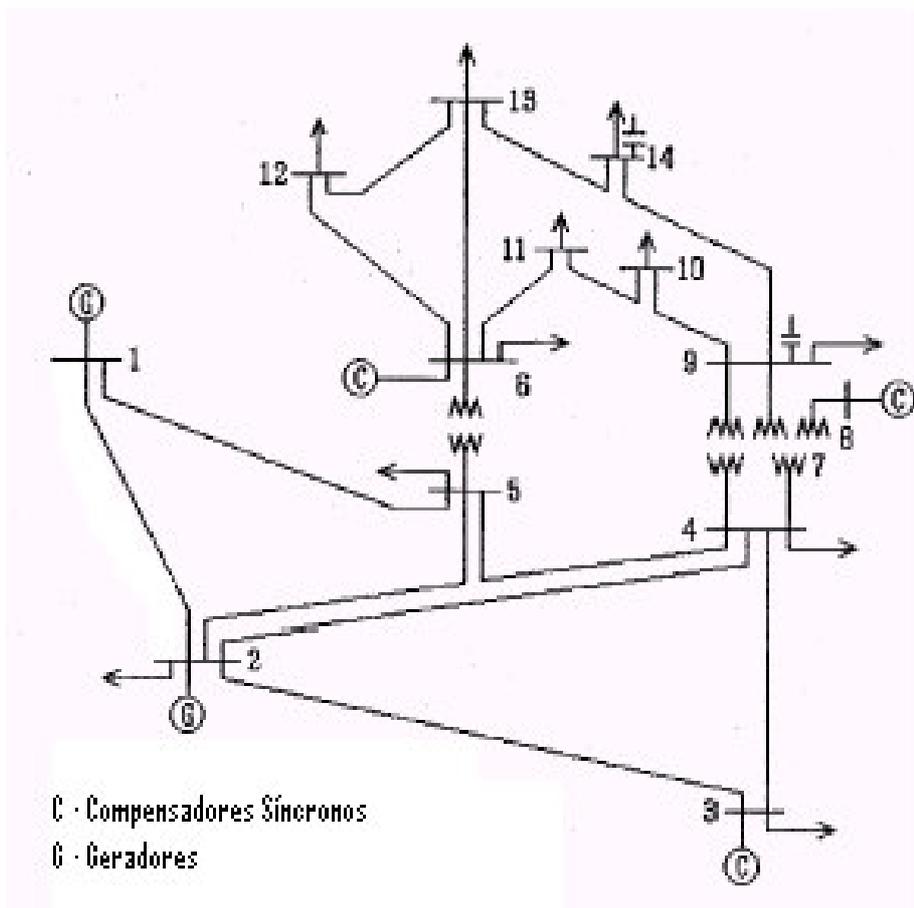


Figura G.1 – Diagrama unifilar do sistema teste IEEE 14 barras

G.2 SISTEMA ELÉTRICO TESTE *IEEE* 118 BARRAS

O Apêndice G mostra o diagrama unifilar simplificado e os dados de entrada para o caso base selecionado do sistema teste IEEE 118 barras. Este é composto por 118 barras de carga (PQ), 4 barras com tensão constante, 17 linhas de transmissão e 3 transformadores. Apresenta 9 barras de carga Um LTC está presente no transformador entre as barras 4 e 9, controlando a tensão na Barra 9 na faixa de 0.98 a 1,00 pu, com tolerância $\pm 0,001$ pu nos limites. Um capacitor de 19MVar está instalado na barra 9. Os dados completos referentes ao LTC encontram-se também neste mesmo apêndice. Nas simulações é previsto um gerador síncrono com potência ativa máxima 220,00 MW na barra 2. A potência base do sistema é 100MVA. Os formatos dos dados apresentados abaixo são os mesmos dos indicados para o sistema 14 barras.

G.2.1 - Dados

Tabela G.7 Dados do sistema

118	53	64	186	0.001	25
-----	----	----	-----	-------	----

Tabela G.8 – Dados de barras

69	1	1.0350	30.0000	5.1640	0.0000	0.0000	0.0000	0.0000	-3.00	3.00	1
1	2	0.9550	10.6700	0.0000	0.5100	0.2700	0.0000	0.0000	-0.05	0.15	1
2	3	0.9710	11.2200	0.0000	0.2000	0.0900	0.0000	0.0000	0.00	0.00	1
3	3	0.9680	11.5600	0.0000	0.3900	0.1000	0.0000	0.0000	0.00	0.00	1
4	2	0.9980	15.2800	-0.0900	0.3000	0.1200	0.0000	0.0000	-3.00	3.00	1
5	3	1.0020	15.7300	0.0000	0.0000	0.0000	0.0000	0.0000	0.00	0.00	1
6	2	0.9900	13.0000	0.0000	0.5200	0.2200	0.0000	0.0000	-0.13	0.50	1
7	3	0.9890	12.5600	0.0000	0.1900	0.0200	0.0000	0.0000	0.00	0.00	1
8	2	1.0150	20.7700	-0.2800	0.0000	0.0000	0.0000	0.0000	-3.00	3.00	1
9	3	1.0430	28.0200	0.0000	0.0000	0.0000	0.0000	0.0000	0.00	0.00	1
10	2	1.0500	35.6100	4.5000	0.0000	0.0000	0.0000	0.0000	-1.47	2.00	1
11	3	0.9850	12.7200	0.0000	0.7000	0.2300	0.0000	0.0000	0.00	0.00	1
12	2	0.9900	12.2000	0.8500	0.4700	0.1000	0.0000	0.0000	-0.35	1.20	1
13	3	0.9680	11.3500	0.0000	0.3400	0.1600	0.0000	0.0000	0.00	0.00	1
14	3	0.9840	11.5000	0.0000	0.1400	0.0100	0.0000	0.0000	0.00	0.00	1
15	2	0.9700	11.2300	0.0000	0.9000	0.3000	0.0000	0.0000	-0.10	0.30	1
16	3	0.9840	11.9100	0.0000	0.2500	0.1000	0.0000	0.0000	0.00	0.00	1
17	3	0.9950	13.7400	0.0000	0.1100	0.0300	0.0000	0.0000	0.00	0.00	1
18	2	0.9730	11.5300	0.0000	0.6000	0.3400	0.0000	0.0000	-0.16	0.50	1
19	2	0.9630	11.0500	0.0000	0.4500	0.2500	0.0000	0.0000	-0.08	0.24	1
20	3	0.9580	11.9300	0.0000	0.1800	0.0300	0.0000	0.0000	0.00	0.00	1
21	3	0.9590	13.5200	0.0000	0.1400	0.0800	0.0000	0.0000	0.00	0.00	1
22	3	0.9700	16.0800	0.0000	0.1000	0.0500	0.0000	0.0000	0.00	0.00	1
23	3	1.0000	21.0000	0.0000	0.0700	0.0300	0.0000	0.0000	0.00	0.00	1
24	2	0.9920	20.8900	-0.1300	0.0000	0.0000	0.0000	0.0000	-3.00	3.00	1
25	2	1.0500	27.9300	2.2000	0.0000	0.0000	0.0000	0.0000	-0.47	1.40	1
26	2	1.0150	29.7100	3.1400	0.0000	0.0000	0.0000	0.0000	-10.00	10.00	1
27	2	0.9680	15.3500	-0.0900	0.6200	0.1300	0.0000	0.0000	-3.00	3.00	1
28	3	0.9620	13.6200	0.0000	0.1700	0.0700	0.0000	0.0000	0.00	0.00	1
29	3	0.9630	12.6300	0.0000	0.2400	0.0400	0.0000	0.0000	0.00	0.00	1
30	3	0.9680	18.7900	0.0000	0.0000	0.0000	0.0000	0.0000	0.00	0.00	1
31	2	0.9670	12.7500	0.0700	0.4300	0.2700	0.0000	0.0000	-3.00	3.00	1
32	2	0.9640	14.8000	0.0000	0.5900	0.2300	0.0000	0.0000	-0.14	0.42	1
33	3	0.9720	10.6300	0.0000	0.2300	0.0900	0.0000	0.0000	0.00	0.00	1
34	2	0.9860	11.3000	0.0000	0.5900	0.2600	0.0000	0.0000	-0.08	0.24	1
35	3	0.9810	10.8700	0.0000	0.3300	0.0900	0.0000	0.0000	0.00	0.00	1
36	2	0.9800	10.8700	0.0000	0.3100	0.1700	0.0000	0.0000	-0.08	0.24	1
37	3	0.9920	11.7700	0.0000	0.0000	0.0000	0.0000	0.0000	0.00	0.00	1
38	3	0.9620	16.9100	0.0000	0.0000	0.0000	0.0000	0.0000	0.00	0.00	1
39	3	0.9700	8.4100	0.0000	0.2700	0.1100	0.0000	0.0000	0.00	0.00	1

40	2	0.9700	7.3500	-0.4600	0.2000	0.2300	0.0000	-3.00	3.00	1
41	3	0.9670	6.9200	0.0000	0.3700	0.1000	0.0000	0.00	0.00	1
42	2	0.9850	8.5300	-0.5900	0.3700	0.2300	0.0000	-3.00	3.00	1
43	3	0.9780	11.2800	0.0000	0.1800	0.0700	0.0000	0.00	0.00	1
44	3	0.9850	13.8200	0.0000	0.1600	0.0800	0.0000	0.00	0.00	1
45	3	0.9870	15.6700	0.0000	0.5300	0.2200	0.0000	0.00	0.00	1
46	2	1.0050	18.4900	0.1900	0.2800	0.1000	0.0000	-1.00	1.00	1
47	3	1.0170	20.7300	0.0000	0.3400	0.0000	0.0000	0.00	0.00	1
48	3	1.0210	19.9300	0.0000	0.2000	0.1100	0.0000	0.00	0.00	1
49	2	1.0250	20.9400	2.0400	0.8700	0.3000	0.0000	-0.85	2.10	1
50	3	1.0010	18.9000	0.0000	0.1700	0.0400	0.0000	0.00	0.00	1
51	3	0.9670	16.2800	0.0000	0.1700	0.0800	0.0000	0.00	0.00	1
52	3	0.9570	15.3200	0.0000	0.1800	0.0500	0.0000	0.00	0.00	1
53	3	0.9460	14.3500	0.0000	0.2300	0.1100	0.0000	0.00	0.00	1
54	2	0.9550	15.2600	0.4800	1.1300	0.3200	0.0000	-3.00	3.00	1
55	2	0.9520	14.9700	0.0000	0.6300	0.2200	0.0000	-0.08	0.23	1
56	2	0.9540	15.1600	0.0000	0.8400	0.1800	0.0000	-0.08	0.15	1
57	3	0.9710	16.3600	0.0000	0.1200	0.0300	0.0000	0.00	0.00	1
58	3	0.9590	15.5100	0.0000	0.1200	0.0300	0.0000	0.00	0.00	1
59	2	0.9850	19.3700	1.5500	2.7700	1.1300	0.0000	-0.60	1.80	1
60	3	0.9930	23.1500	0.0000	0.7800	0.0300	0.0000	0.00	0.00	1
61	2	0.9950	24.0400	1.6000	0.0000	0.0000	0.0000	-1.00	3.00	1
62	2	0.9980	23.4300	0.0000	0.7700	0.1400	0.0000	-0.20	0.20	1
63	3	0.9690	22.7500	0.0000	0.0000	0.0000	0.0000	0.00	0.00	1
64	3	0.9840	24.5200	0.0000	0.0000	0.0000	0.0000	0.00	0.00	1
65	2	1.0050	27.6500	3.9100	0.0000	0.0000	0.0000	-0.67	2.00	1
66	2	1.0500	27.4800	3.9200	0.3900	0.1800	0.0000	-0.67	2.00	1
67	3	1.0200	24.8400	0.0000	0.2800	0.0700	0.0000	0.00	0.00	1
68	3	1.0030	27.5500	0.0000	0.0000	0.0000	0.0000	0.00	0.00	1
70	2	0.9840	22.5800	0.0000	0.6600	0.2000	0.0000	-0.10	0.32	1
71	3	0.9870	22.1500	0.0000	0.0000	0.0000	0.0000	0.00	0.00	1
72	2	0.9800	20.9800	-0.1200	0.0000	0.0000	0.0000	-1.00	1.00	1
73	2	0.9910	21.9400	-0.0600	0.0000	0.0000	0.0000	-1.00	1.00	1
74	2	0.9580	21.6400	0.0000	0.6800	0.2700	0.0000	-0.06	0.09	1
75	3	0.9670	22.9100	0.0000	0.4700	0.1100	0.0000	0.00	0.00	1
76	2	0.9430	21.7700	0.0000	0.6800	0.3600	0.0000	-0.08	0.23	1
77	2	1.0060	26.7200	0.0000	0.6100	0.2800	0.0000	-0.20	0.70	1
78	3	1.0030	26.4200	0.0000	0.7100	0.2600	0.0000	0.00	0.00	1
79	3	1.0090	26.7200	0.0000	0.3900	0.3200	0.0000	0.00	0.00	1
80	2	1.0400	28.9600	4.7700	1.3000	0.2600	0.0000	-1.65	2.80	1
81	3	0.9970	28.1000	0.0000	0.0000	0.0000	0.0000	0.00	0.00	1
82	3	0.9890	27.2400	0.0000	0.5400	0.2700	0.0000	0.00	0.00	1
83	3	0.9850	28.4200	0.0000	0.2000	0.1000	0.0000	0.00	0.00	1
84	3	0.9800	30.9500	0.0000	0.1100	0.0700	0.0000	0.00	0.00	1
85	2	0.9850	32.5100	0.0000	0.2400	0.1500	0.0000	-0.08	0.23	1
86	3	0.9870	31.1400	0.0000	0.2100	0.1000	0.0000	0.00	0.00	1
87	2	1.0150	31.4000	0.0400	0.0000	0.0000	0.0000	-1.00	10.00	1
88	3	0.9870	35.6400	0.0000	0.4800	0.1000	0.0000	0.00	0.00	1
89	2	1.0050	39.6900	6.0700	0.0000	0.0000	0.0000	-2.10	3.00	1
90	2	0.9850	33.2900	-0.8500	0.7800	0.4200	0.0000	-3.00	3.00	1
91	2	0.9800	33.3100	-0.1000	0.0000	0.0000	0.0000	-1.00	1.00	1
92	2	0.9930	33.8000	0.0000	0.6500	0.1000	0.0000	-0.03	0.09	1
93	3	0.9870	30.7900	0.0000	0.1200	0.0700	0.0000	0.00	0.00	1
94	3	0.9910	28.6400	0.0000	0.3000	0.1600	0.0000	0.00	0.00	1
95	3	0.9810	27.6700	0.0000	0.4200	0.3100	0.0000	0.00	0.00	1
96	3	0.9930	27.5100	0.0000	0.3800	0.1500	0.0000	0.00	0.00	1
97	3	1.0110	27.8800	0.0000	0.1500	0.0900	0.0000	0.00	0.00	1
98	3	1.0240	27.4000	0.0000	0.3400	0.0800	0.0000	0.00	0.00	1
99	2	1.0100	27.0400	-0.4200	0.0000	0.0000	0.0000	-1.00	1.00	1
100	2	1.0170	28.0300	2.5200	0.3700	0.1800	0.0000	-0.50	1.55	1
101	3	0.9930	29.6100	0.0000	0.2200	0.1500	0.0000	0.00	0.00	1
102	3	0.9910	32.3000	0.0000	0.0500	0.0300	0.0000	0.00	0.00	1
103	2	1.0010	24.4400	0.4000	0.2300	0.1600	0.0000	-0.15	0.40	1
104	2	0.9710	21.6900	0.0000	0.3800	0.2500	0.0000	-0.08	0.23	1
105	2	0.9650	20.5700	0.0000	0.3100	0.2600	0.0000	-0.08	0.23	1
106	3	0.9620	20.3200	0.0000	0.4300	0.1600	0.0000	0.00	0.00	1
107	2	0.9520	17.5300	-0.2200	0.2800	0.1200	0.0000	-2.00	2.00	1
108	3	0.9670	19.3800	0.0000	0.0200	0.0100	0.0000	0.00	0.00	1
109	3	0.9670	18.9300	0.0000	0.0800	0.0300	0.0000	0.00	0.00	1
110	2	0.9730	18.0900	0.0000	0.3900	0.3000	0.0000	-0.08	0.23	1
111	2	0.9800	19.7400	0.3600	0.0000	0.0000	0.0000	-1.00	10.00	1
112	2	0.9750	14.9900	-0.4300	0.2500	0.1300	0.0000	-1.00	10.00	1
113	2	0.9930	13.7400	-0.0600	0.0000	0.0000	0.0000	-1.00	2.00	1
114	3	0.9600	14.4600	0.0000	0.0800	0.0300	0.0000	0.00	0.00	1
115	3	0.9600	14.4600	0.0000	0.2200	0.0700	0.0000	0.00	0.00	1
116	2	1.0050	27.1200	-1.8400	0.0000	0.0000	0.0000	-10.00	10.00	1
117	3	0.9740	10.6700	0.0000	0.2000	0.0800	0.0000	0.00	0.00	1
118	3	0.9490	21.9200	0.0000	0.3300	0.1500	0.0000	0.00	0.00	1

Tabela G 9 – Dados de Linhas de transmissão

1	1	2	0.0303	0.0999	0.0254	0	0	1	81	50	57	0.0474	0.134	0.0332	0	0	1
2	1	3	0.0129	0.0424	0.01082	0	0	1	82	56	58	0.0343	0.0966	0.0242	0	0	1
3	4	5	0.00176	0.00798	0.0021	0	0	1	83	51	58	0.0255	0.0719	0.01788	0	0	1
4	3	5	0.0241	0.108	0.0284	0	0	1	84	54	59	0.0503	0.2293	0.0598	0	0	1
5	5	6	0.01119	0.054	0.01426	0	0	1	85	56	59	0.0825	0.251	0.0569	0	0	2
6	6	7	0.00459	0.0208	0.0055	0	0	1	86	56	59	0.0803	0.239	0.0536	0	0	1
7	8	9	0.00244	0.0305	1.162	0	0	1	87	55	59	0.04739	0.2158	0.05646	0	0	1
8	8	5	0	0.0267	0	0	0.985	0	88	59	60	0.0317	0.145	0.0376	0	0	1
9	9	10	0.00258	0.0322	1.23	0	0	1	89	59	61	0.0328	0.15	0.0388	0	0	1
10	4	11	0.0209	0.0688	0.01748	0	0	1	90	60	61	0.00264	0.0135	0.01456	0	0	1
11	5	11	0.0203	0.0682	0.01738	0	0	1	91	60	62	0.0123	0.0561	0.01468	0	0	1
12	11	12	0.00595	0.0196	0.00502	0	0	1	92	61	62	0.00824	0.0376	0.0098	0	0	1
13	2	12	0.0187	0.0616	0.01572	0	0	1	93	63	59	0	0.0386	0	0.96	0	1
14	3	12	0.0484	0.16	0.0406	0	0	1	94	63	64	0.00172	0.02	0.216	0	0	1
15	7	12	0.00862	0.034	0.00874	0	0	1	95	64	61	0	0.0268	0	0.985	0	1
16	11	13	0.02225	0.0731	0.01876	0	0	1	96	38	65	0.00901	0.0986	1.046	0	0	1
17	12	14	0.0215	0.0707	0.01816	0	0	1	97	64	65	0.00269	0.0302	0.38	0	0	1
18	13	15	0.0744	0.2444	0.06268	0	0	1	98	49	66	0.018	0.0919	0.0248	0	0	2
19	14	15	0.0595	0.195	0.0502	0	0	1	99	49	66	0.018	0.0919	0.0248	0	0	1
20	12	16	0.0212	0.0834	0.0214	0	0	1	100	62	66	0.0482	0.218	0.0578	0	0	1
21	15	17	0.0132	0.0437	0.0444	0	0	1	101	62	67	0.0258	0.117	0.031	0	0	1
22	16	17	0.0454	0.1801	0.0466	0	0	1	102	65	66	0	0.037	0	0.935	0	1
23	17	18	0.0123	0.0505	0.01298	0	0	1	103	66	67	0.0224	0.1015	0.02682	0	0	1
24	18	19	0.01119	0.0493	0.01142	0	0	1	104	65	68	0.00138	0.016	0.638	0	0	1
25	19	20	0.0252	0.117	0.0298	0	0	1	105	47	69	0.0844	0.2778	0.07092	0	0	1
26	15	19	0.012	0.0394	0.0101	0	0	1	106	49	69	0.0985	0.324	0.0828	0	0	1
27	20	21	0.0183	0.0849	0.0216	0	0	1	107	68	69	0	0.037	0	0.935	0	1
28	21	22	0.0209	0.097	0.0246	0	0	1	108	69	70	0.03	0.127	0.122	0	0	1
29	22	23	0.0342	0.159	0.0404	0	0	1	109	24	70	0.00221	0.4115	0.10198	0	0	1
30	23	24	0.0135	0.0492	0.0498	0	0	1	110	70	71	0.00882	0.0355	0.00878	0	0	1
31	23	25	0.0156	0.08	0.0864	0	0	1	111	24	72	0.0488	0.196	0.0488	0	0	1
32	26	25	0	0.0382	0	0	0.96	0	112	71	72	0.0446	0.18	0.0444	0	0	1
33	25	27	0.0318	0.163	0.1764	0	0	1	113	71	73	0.00866	0.0454	0.01178	0	0	1
34	27	28	0.01913	0.0855	0.0216	0	0	1	114	70	74	0.0401	0.1323	0.03368	0	0	1
35	28	29	0.0237	0.0943	0.0238	0	0	1	115	70	75	0.0428	0.141	0.036	0	0	1
36	30	17	0	0.0388	0	0	0.96	0	116	69	75	0.0405	0.122	0.124	0	0	1
37	8	30	0.00431	0.0504	0.514	0	0	1	117	74	75	0.0123	0.0406	0.01034	0	0	1
38	26	30	0.00799	0.086	0.908	0	0	1	118	76	77	0.0444	0.148	0.0368	0	0	1
39	17	31	0.0474	0.1563	0.0399	0	0	1	119	69	77	0.0309	0.101	0.1038	0	0	1
40	29	31	0.0108	0.0331	0.0083	0	0	1	120	75	77	0.0601	0.1999	0.04978	0	0	1
41	23	32	0.0317	0.1153	0.1173	0	0	1	121	77	78	0.00376	0.0124	0.01264	0	0	1
42	31	32	0.0298	0.0985	0.0251	0	0	1	122	78	79	0.00546	0.0244	0.00648	0	0	1
43	27	32	0.0229	0.0755	0.01926	0	0	1	123	77	80	0.017	0.0485	0.0472	0	0	2
44	15	33	0.038	0.1244	0.03194	0	0	1	124	77	80	0.0294	0.105	0.0228	0	0	1
45	19	34	0.0752	0.247	0.0632	0	0	1	125	79	80	0.0156	0.0704	0.0187	0	0	1
46	35	36	0.00224	0.0102	0.00268	0	0	1	126	68	81	0.00175	0.0202	0.808	0	0	1
47	35	37	0.011	0.0497	0.01318	0	0	1	127	81	80	0	0.037	0	0.935	0	1
48	33	37	0.0415	0.142	0.0366	0	0	1	128	77	82	0.0298	0.0853	0.08174	0	0	1
49	34	36	0.00871	0.0268	0.00568	0	0	1	129	82	83	0.0112	0.03665	0.03796	0	0	1
50	34	37	0.00256	0.0094	0.00984	0	0	1	130	83	84	0.0625	0.132	0.0258	0	0	1
51	38	37	0	0.0375	0	0	0.935	0	131	83	85	0.043	0.148	0.0348	0	0	1
52	37	39	0.0321	0.106	0.027	0	0	1	132	84	85	0.0302	0.0641	0.01234	0	0	1
53	37	40	0.0593	0.168	0.042	0	0	1	133	85	86	0.035	0.123	0.0276	0	0	1
54	30	38	0.00464	0.054	0.422	0	0	1	134	86	87	0.02828	0.2074	0.0445	0	0	1
55	39	40	0.0184	0.0605	0.01552	0	0	1	135	85	88	0.02	0.102	0.0276	0	0	1
56	40	41	0.0145	0.0487	0.01222	0	0	1	136	85	89	0.0239	0.173	0.047	0	0	1
57	40	42	0.0555	0.183	0.0466	0	0	1	137	88	89	0.0139	0.0712	0.01934	0	0	1
58	41	42	0.041	0.135	0.0344	0	0	1	138	89	90	0.0518	0.188	0.0528	0	0	2
59	43	44	0.0608	0.2454	0.06068	0	0	1	139	89	90	0.0238	0.0997	0.106	0	0	1
60	34	43	0.0413	0.1681	0.04226	0	0	1	140	90	91	0.0254	0.0836	0.0214	0	0	1
61	44	45	0.0224	0.0901	0.0224	0	0	1	141	89	92	0.0099	0.0505	0.0548	0	0	2
62	45	46	0.04	0.1356	0.0332	0	0	1	142	89	92	0.0393	0.1581	0.0414	0	0	1
63	46	47	0.038	0.127	0.0316	0	0	1	143	91	92	0.0387	0.1272	0.03268	0	0	1
64	46	48	0.0601	0.189	0.0472	0	0	1	144	92	93	0.0258	0.0848	0.0218	0	0	1
65	47	49	0.0191	0.0625	0.01604	0	0	1	145	92	94	0.0481	0.158	0.0406	0	0	1
66	42	49	0.0715	0.323	0.086	0	0	2	146	93	94	0.0223	0.0732	0.01876	0	0	1
67	42	49	0.0715	0.323	0.086	0	0	1	147	94	95	0.0132	0.0434	0.0111	0	0	1
68	45	49	0.0684	0.186	0.0444	0	0	1	148	80	96	0.0356	0.182	0.0494	0	0	1
69	48	49	0.0179	0.0505	0.01258	0	0	1	149	82	96	0.0162	0.053	0.0544	0	0	1
70	49	50	0.0267	0.0752	0.01874	0	0	1	150	94	96	0.0269	0.0869	0.023	0	0	1
71	49	51	0.0486	0.137	0.0342	0	0	1	151	80	97	0.0183	0.0934	0.0254	0	0	1
72	51	52	0.0203	0.0588	0.01396	0	0	1	152	80	98	0.0238	0.108	0.0286	0	0	1
73	52	53	0.0405	0.1635	0.04058	0	0	1	153	80	99	0.0454	0.206	0.0546	0	0	1
74	53	54	0.0263	0.122	0.031	0	0	1	154	92	100	0.0648	0.295	0.0472	0	0	1
75	49	54	0.073	0.289	0.0738	0	0	2	155	94	100	0.0178	0.058	0.0604	0	0	1
76	49	54	0.0869	0.291	0.073	0	0	1	156	95	96	0.0171	0.0547	0.01474	0	0	1
77	54	55	0.0169	0.0707	0.0202	0	0	1	157	96	97	0.0173	0.0885	0.024	0	0	1
78	54	56	0.00275	0.00955	0.00732	0	0	1	158	98	100	0.0397	0.179	0.0476	0	0	1
79	55	56	0.00488	0.0151	0.00374	0	0	1	159	99	100	0.018	0.0813	0.0216	0	0	1
80	56	57	0.0343	0.0966	0.0242	0	0	1	160	100	101	0.0277	0.1262	0.0328	0	0	1

REFERÊNCIAS BIBLIOGRÁFICAS

- [ABR98] P. J. Abrão, A. P. A. da Silva and A. C. Z. de Souza, **Voltage Security Assessment using Artificial Intelligence Techniques**, VI Symposium of Specialists in Electric Operational and Expansion Planning, May, 24th – 29th – 1998, Salvador – BA, Brazil.
- [ABR98A] P. J. Abrão, A. P. Alves da Silva e A. C. Zambroni de Souza, **Utilização de Aquisição Automática de Conhecimento no Processo de Avaliação de Segurança de Tensão**, XII Congresso Brasileiro de Automática, Vol. VI, pp. 2156 - 2160 Uberlândia – MG, 14 – 18/Setembro/1998.
- [AJJ89] V. Ajjarapu, **Identification of Steady State Voltage Stability in Power Systems**, Proc. of International Conference on High Technology in the Power Industry, March 1988, pp. 244-247.
- [ALV89] F. L. Alvarado and T. H. Jung, **Direct Detection of Voltage Collapse Conditions**, Proc. Bulk Power System Voltage Phenomena - Voltage Stability and Security, EL-6183, EPRI, January 1989, pp. 5.23-5.38.
- [BAR95] J. Barquin, T. Gómez and F. L. Pagola, **Estimating the Loading Limit Margin Taking Into Account Voltage Collapse Areas**, IEEE/PES Winter Meeting - Paper 95 WM 183-4 PWRs, 1990, jan 29 - feb 2 1995.
- [BEN97] J. M. Benítez, J. L. Castro and I. Requena, **Are Artificial Neural Networks Black Box?**, IEEE Transactions on Neural Networks, Vol. 8, No. 5, September 1997.
- [BIS95] C. M. Bishop, **Neural Networks for Pattern Recognition**, Oxford University Press Inc., New York, 1995.
- [CAN91] C. A. Canizares, **Voltage Collapse and Transient Energy Function Analyses of AC/DC Systems**, Ph. D. thesis, University of Wisconsin-Madison, 1991.
- [CAN95] C. A. Canizares, A. C. Z. de Souza and V. H. Quintana, **Comparison of Performance Indices for Detection of Proximity to Voltage Collapse**, IEEE Transactions on Power Systems, vol.11, No. 3, August 1996, pp. 1441-1447.
- [CHE93] L. Chen, **Application of Chaotic Simulation and Self-Organizing Neural Net to Power System Voltage Stability Monitoring**, Proc. 2nd International Forum on Applications of Neural Networks to Power Systems (ANNPS), pp. 367-372, Yokohama - Japan, April 1993.
- [CHE98] Vladimir Cherkassky and Filip Mulier, **Learning from Data**, John Wiley & Sons, New York, 1998.
-

- [CHI89] H. D. Chiang , **Study of the Existence of Energy Functions for Power Systems with Losses**, IEEE Transactions on Power Systems, vol. 36, no. 11, November 1989, pp. 1423-1429.
- [CHO91] B. H. Chowdhury and B. M. Wilamowski, **Security Assessment using Neural Computing**, Proc. 1st International Forum on Applications of Neural Networks to Power Systems (ANNPS), pp. 54-58, Seattle - WA, July 1991.
- [COR94] B. J. Cory, U. G. Knight, R. Sbrizzai and M. Trovato, **Towards a Neural Networks Based Voltage Stability Assessment**.
- [CRA96] Mark W. Craven, **Extracting Comprehensible Models from Trained Neural Networks**, Ph. D. Thesis, University of Wisconsin – Madison, 1996.
- [DIL01] T. S. Dillon, D. Wang, E. Chang, **Data Mining Using Neural Networks**, Engineering Intelligent Systems, Vol. 9, No. 4, December 2001, pp. 187 -194.
- [DIL95] T. S. Dillon and D. Niebur, **Neural Networks Applications in Power Systems**, CRL International Series in Intelligent Systems and their Applications - CRL Publishing Ltd. – London.
- [DUD73] Richard O. Duda and Peter E. Hart, **Pattern Classification and Scene Analysis**, John Wiley & Sons, New York, 1973.
- [FER94] N. F. Ferrara e C. P. C. do Prado, **Caos uma introdução**, Editora Edgard Blücler LTDA, Brasil, 1994.
- [FOR95] **Visual Fortran v6.0**, Compag Co.,1998.
- [FU094] L. Fu, **Rule Generation from Neural Networks**, IEEE Transaction on Systems, Man, and Cybernetics, vol. 24, no. 8, august 1994.
- [GAL88] S. I. Galant, **Connectionist Expert Systems**, Communications of the ACM, Vol. 31, number 2, February 1988.
- [GAO95] B. Gao, G. K. Morison and P. Kundur, **Towards the Development of a Systematic Approach for Voltage Stability Assessment of Large-Scale Power Systems**, IEEE/PES Summerr Meeting - Paper 95 SM 526-4 PWRS, July 23-27, 1995.
- [GAR98] J. N. Garcez, E. S. Cruz and R. K. da Silva, **A Connectionist Approach to Voltage Security Assessment in Power Systems**, VI Symposium of Specialists in Electric Operational and Expansion Planning, May, 24th - 29th - 1998, Salvador - BA, Brazil
- [GCOI97] ..., Relatório Técnico GCOI, Eletrobrás, Maio 97
- [GOL91] G. H. Golub and C. F. Van Loan, **Matrix Computations**, Second Edition, The Johns Hopkins University Press, 1991.
-

- [HAS95] M. H. Hassoun, **Fundamentals of Artificial Neural Networks**, A Bradford Book, The Massachusetts Institute of Technology Press, 1995.
- [HON02] L. M. Honório, **Consideração de Técnicas de Otimização Aplicadas ao Problema de Colapso de Tensão**, Tese de Doutorado, Instituto de Engenharia Elétrica, Escola Federal de Engenharia de Itajubá – EFEI, 2002.
- [HRU99] E. R. Hruschka e N. F. F. Ebecken, **Extração de Regras de Redes Neurais por meio do Algoritmo RX Modificado: Um exemplo de Aplicação em Modelagem de Dados Meteriológicos**, IV Congresso Brasileiro de Redes Neurais, págs 047-051, Julho 20-22, 1999, ITA, São José dos Campos – SP – Brasil.
- [HUI91] K. C. Hui and M. J. Short, **A Neural Networks Approach to Voltage Security Monitoring and Control**, Proc. 1st International Forum on Applications of Neural Networks to Power Systems (ANNPS), pp. 89-93, Seattle - WA, July 1991.
- [I3E93] **Suggest Techniques for Voltage Stability Analysis**, IEEE Power Engineering Society, 1993
- [I3E96] **A Tutorial Course on Artificial Neural Networks with Applications To Power Systems**, IEEE Power Engineering Society, 1996 TP 112-0, pp. 104-127.
- [IGN94] J. P. Ignizio and T. M. Cavalier, **Linear Programming**, Prentice Hall, 1994.
- [KEI95] AAEI-Keib and X. Ma, **Application of Artificial Neural Networks in Voltage Stability Assessment**, IEEE Trans. On Power Systems, Vol. 10, nº 4, nov. 1995, pp. 1890-1896.
- [KLU97] R. P. Klump and T. Overbye, **Assessment of Transmission System Loadability**, IEEE Transactions on Power Systems, vol. 12, no. 1, February 1997, pp. 416-423.
- [KUM91] A.B. R. Kumar, A. Ipakchi, V. Brandwajn, M. El-Sharkawi and G. Cauley, **Neural Networks for Dynamic Security Assessment of Large-Scale Power Systems: Requirements Overview**, Proc. 1st International Forum on Applications of Neural Networks to Power Systems (ANNPS), pp. 65-71, Seattle - WA, July 1991.
- [KUM97] A. B. R. Kumar, V. Brandwajn, A. Ipakchi and R. Adapa, **Integrated Framework for Dynamic Security Analysis**, pp. 260-265.
- [KUN90] B. Gao, G. K. Morrison and P. Kundur, **Voltage Stability Evaluation using Modal Analysis**, IEEE/PES Winter Meeting – Paper 90 WM 096-8 PWRS, 1990.
- [KUN97] P. Kundur and G. K. Morison, **A Review of Definitions and Classification of Stability Problems in Today's Power Systems**, IEEE PES Meeting, Feb. 2-6, 1997, New York, NY.
-

- [LIA97] T. E. Dy-Liaco, **Enhancing Power System Security Control**, IEEE Computer Applications in Power, pp. 38-41, July 1997.
- [LIU99] M. Su, C. Liu and C. Chang, **Rule Extraction for Voltage Security Margin Estimation**, IEEE Transaction on Industrial Electronics, Vol. 46, No. 6, December 1999.
- [LOF92] P. A. Löff, T. Smed, G. Anderson and D. J. Hill, **Fast Calculation of a Voltage Stability Index**, IEEE Trans. Power Systems, vol. 7, n°. 1, Feb. 1992, pp. 54-64.
- [LOP98] J. A. P. Lopes and F. D. Fernandes, **Fast Evaluation of Voltage Collapse Risk using Machine Learning Techniques**, VI Symposium of Specialists in Electric Operational and Expansion Planning, May, 24th – 29th – 1998, Salvador – BA, Brazil
- [MAI00] Frédéric Maire, **On the Convergence of Validity Interval Analysis**, IEEE Transaction on Neural Networks, Vol. 11, No. 3, May 2000.
- [MAI97] Frédéric Maire, **A Partial Order for the M-of-N Rule-Extraction Algorithm**, IEEE Transaction on Neural Networks, Vol. 8, No. 6, November 1997.
- [MAR94] Marannino, P. Bresesti, M. Delfanti, G.P. Granelli and M. Montagna, **Voltage Collapse Proximity Indicators for Very Short Term Security Assessment**, Proc. Bulk Power System Voltage Phenomena III-Voltage Stability and Security, ECC Inc. , Switzerland, August 1994.
- [MEN99] P. P. C. Mendes, **Aplicação de Redes Neurais Artificiais na Análise em Tempo Real da Estabilidade de Tensão de Regime Permanente de Sistemas Elétricos de Potência**, Tese de Doutorado, COPPY/UFRJ, 1999.
- [MIT97] T.M. Mitchell: **Machine Learning**, McGraw-Hill, 1997.
- [MOM96] J. A. Momoh, L. G. Dias and R. Adapa, **Voltage Stability Assessment and Enhancement using Artificial Neural Networks and Reactive Compensation**, Proc. Of the 2nd International Conference on Intelligence Systems Applications to Power Systems, 1996
- [MOO89] R. Mooney, J. Shavlik, G. Towell and AGove, **An Experimental Comparison of Symbolic and Connectionist Learning Algorithms**, Proc. Of the 11th International Joint Conference on Artificial (IJCAI-89), Detroit, pp. 775-780
- [MOR91] H. Mori, **Na Artificial Neural-Net Based Method for Estimating Power System Dynamic Stability Index**, Proc. 1st International Forum on Applications of Neural Networks to Power Systems (ANNPS), pp. 129-133, Seattle – WA, July 1991.
- [MOR93] H. Mori and Y. Tamaru, **A Decentralized Scheme for Voltage Instability Monitoring with Hybrid Artificial Neural Networks**, Proc. 2nd International Forum on Applications of Neural Networks to Power Systems (ANNPS), pp. 378-381, Yokohama – Japan, April 1993.
-

- [MOR99] K. Morison, H. Hamadanizadeh and L. Wang, **Dynamic Security Assessment Tools**, IEEE Summer Meeting 1999.
- [NET98] J. F. Neto, E. Sodr e e M. Ade Carvalho Jr., **Utiliza o de Redes Neurais para Determina o da Seguran a de Sistemas de Pot ncia em Rela o   Instabilidade de Tens o de Longo Termo**, VI Simp sio de Especialistas em Planejamento da Opera o e Expans o El trica, 24-29 maio 1998, Salvador – BA, Brasil
- [NEU95] **NeuralWorks v5.21**, NeuralWare Co., 1995
- [NIE91] D. Niebur and A. J. Germond, **Power Flow Classification for Static Security Assessment**, Proc. 1st International Forum on Applications of Neural Networks to Power Systems (ANNPS), pp. 83-88, Seattle – WA, July 1991.
- [OPI93] D. W. Opitz and J. W. Shavlik, **Heuristically Expanding Knowledge-Based Neural Networks**, 13^a International Joint Conference on Artificial Intelligence (IJCAI-93), 1993.
- [OVE91a] T. Overbye, **Application of an energy based security method to voltage instability in electrical power systems**, PhD thesis, University of Wisconsin – Madison, 1991.
- [OVE91b] T. J. Overbye and C. L. de Marco, **Voltage Security Enhancement using Energy Based Sensitivities**, IEEE Trans. Power Systems, vol. 6, no. 3, August 1991, pp. 1196-1202.
- [PAV85] M. Ribbens-Pavella, T. Van Cutsem and P. Rousseaux, **On-Line Stability and Dynamic Security Assessment of Electric Power Systems**, IFAC Electric Energy Systems, Rio de Janeiro, Brazil, 1985.
- [PRA91] R. B. Prada, A. C. Zambroni, X. V. Filho, A. G. Massaud and J. C. Oliveira, **Voltage Stability: Phenomena Characterization Based on Reactive Control Effects and System Critical Areas Identification**, Proceedings of the third Sepope meeting , Belo Horizonte , 1991, SP-14.
- [QUI96] J. R. Quinlan, **Improved Use of Continuous Attributes in C4.5**, Journal of Artificial Intelligence Research 4, pp. 77 – 90, March 1996.
- [RIC94] E. Rich and K. Knight, **Intelig ncia Artificial**, MAKRON Books do Brasil Editora Ltda., 2^a edi o, 1994.
- [RUS95] S. Russel and P. Norvig, **Artificial Intelligence – A modern Approach**, Prentice Hall, Inc., 1995
- [SAI90] K. Saito and R. Nakano, **Medical Diagnostic Expert System Based on PDP Model**, pp. i-255 – i-262
-

- [SAI96] K. Saito and Nakano, **A Constructive Learning Algorithm for an HME**, pp. 1268 – 1273.
- [SAN99] R. T. Santos, J. C. Nievola, A. A. Freitas, H. S. Lopes, **Extração de Regras de Redes Neurais via Algoritmos Genéticos**, IV Congresso Brasileiro de Redes Neurais, págs 158-163, Julho 20-22, 1999, ITA, S. J. dos Campos – SP – Brasil.
- [SCA95] M. La Scala, M. Trovato and F. Torelli, **A Neural Network-Based Method for Voltage Security Monitoring**, IEEE PES Summer Meeting, paper 95SM560-6 PWRs, 1995.
- [SCH90] R. A. Schlueter, I. Hu and M. W. Chang, **Methods for Determining Proximity to Voltage Collapse**, *IEEE/PES Winter Meeting - Paper 90 WM 096-8 PWRs*, 1990.
- [SCH97] Jürgen Schürmann, **Pattern Classification – A Unified View of Statistical and Neural Approaches**, John Wiley & Sons, New York, 1996.
- [SES94] S. Sestito and T. S. Dillon, **Automated Knowledge Acquisition**, Prentice Hall of Australia Pty Ltd, 1994.
- [SET00] R. Setiono, **Extracting M-of-N Rules from Trained Neural Networks**, IEEE Transactions on Neural Networks, Vol. 11, No. 2, March 2000.
- [SET90] I. K. Sethi, **Entropy Nets: From Decision Trees to Neural Networks**, Proceeding of the IEEE, Vol. 78, No. 10, October 1990.
- [SET95] I. K. Sethi, **Neural Implementation of Tree Classifiers**, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 25, No. 8, August 1995.
- [SEY88] R. Seydel, **From Equilibrium to Chaos - Practical Bifurcation and Stability Analysis**. Elsevier Science, North-Holland, 1988.
- [SHA93] M. A. El-Sharkawi and Rajasekhar Atteri, **Static Security Assessment of Power System using Kohonen Neural Network**, Proc. 2nd International Forum on Applications of Neural Networks to Power Systems (ANNPS), pp. 373-377, Yokohama - Japan, April 1993.
- [SHA93a] M. A. El-Sharkawi and S. S. Huang, **Application of Query-Based Learning to Power System Static Security Assessment**, Proc. 2nd International Forum on Applications of Neural Networks to Power Systems (ANNPS), pp. 111-117, Yokohama - Japan, April 1993.
- [SHA94] M. A. El-Sharkawi and S. J. Huang, **Application of Genetic-Based Neural Networks to Power System Static Security Assessment**, Proc. of the 1st International Conference on Intelligence Systems Applications to Power Systems, Montpellier - France, Sep 5-9, 1994.
-

- [SHA96] M. A. El-Sharkawi, Y. Mansour and E. Vaahedi, **Large Scale Dynamic Security Screening and Ranking using Neural Networks**, V Symposium of Specialists in Electric Operational and Expansion Planning, May 19th-24th, 1996, Recife - PE, Brazil.
- [SHA96a] M. A. El-Sharkawi and S. J. Huang, **Development of Genetic Algorithm Embedded Kohonen Neural Network for Dynamic Security Assessment**, Proc. of the 2nd International Conference on Intelligence Systems Applications to Power Systems, 1994.
- [SIL91] A. P. Alves da Silva: **Pattern Analysis and Parallel Distributed Processing in Power System State Estimation**, Ph.D. Thesis, Dept. Electrical Engineering, University of Waterloo, Canada, 1991.
- [SIL94] A. P. Alves. da Silva, V. H. Quintana and G. L. Torres, **Optimal Estimate Training 2 (OET-2)**, 1° Congresso Brasileiro de Redes Neurais, pag.'s 72 - 77, 24 a 27 de outubro de 1994, Escola Federal de Engenharia de Itajubá, Itajubá – MG – Brasil.
- [SOU00] A. C. Zambroni de Souza, **Discussion on some Voltage Collapse Índices**, Electric Power Systems Research 53, pp. 53 – 58, 2000.
- [SOU94] A. C. Z. de Souza and V. H. Quintana, **A New Technique of Network Partitioning for Voltage Collapse Margin Calculations**, IEE Proc.- Generation, Transmission and. Distribution, vol. 141, November 1994, pp. 630-636.
- [SOU96a] A. C. Z. de Souza, **Determinação da Barra Crítica e do Ponto de Colapso Pelo Vetor Tangente**, XI CBA, São Paulo, Setembro de 1996, vol. 3, pags 1393-1998.
- [SOU96b] A. C. Z. de Souza, C. A. Cañizares and V. H. Quintana, **Critical Bus and Point of Collapse Determination Using Tangent Vectors**, 28th North American Power Symposium, Cambridge, USA, November 10-12 1996, pp.329-333.
- [SOU96c] A. C. Z. de Souza, C. A. Cañizares and V. H. Quintana, **Critical Bus and Point of Collapse Determination Using Tangent Vectors**, 28th North American Power Symposium, , Cambridge, USA, November 10-12 1996, pp.329-333.
- [SOU97] A. C. Zambroni de Souza ,C. A . Cañizares and V. H. Quintana, **New Techniques to Sped up Voltage Collapse Computations Using Tangent Vectors**, IEEE/PES Summer Meeting - Paper PE-219- PWRS-0-11-1996, Berlin, July 23 - 27 1997.
- [SRI97] R. D. Sriram, **Intelligent Systems for Engineering: A Knowledge-Based Approach** , Springer-Verlag Limited 1997.
- [THR94] S.B. Thrun, **Extracting Provably Correct Rules from Artificial Neural Networks**, Inst. Informatik III, Univ. Bonn, Germany, Tech. Rep. IAI-TR-93-5, 1994.
-

- [TIC98] A. B. Tickle, R. Andrews, M. Golea and J. Diederich, **The Truth Will Come to Light: Directions and Challenges in Extracting the Knowledge Embedded Within Trained Artificial Neural Networks**, IEEE Transactions on Neural Networks, Vol. 9, No. 6, November 1998.
- [TOW92] G. Towell and J. W. Shavlik, **Extracting Refined Rules from Knowledge-Based Neural Networks**, Machine Learning, November, 1992.
- [TOW92a] G. Towell and J. W. Shavlik, **Interpretation of Artificial Neural Networks: Mapping Knowledge-Based Neural Networks into Rules**, Advances in Neural Information Processing Systems 4, Morgan Kaufmann, 1992.
- [VAD91] S. V. Vadari and S. S. Venkata, **A Hybrid Artificial Neural Network/Artificial Intelligence Approach for Voltage Stability Enhancement**,
- [WEE91] S. Weerasooriya and M. A. El-Sharkawi, **Use of Karhunen-Loève Expansion in Training Neural Networks for Static Security Assessment**, Proc. 1st International Forum on Applications of Neural Networks to Power Systems (ANNPS), pp. 59-64, Seattle - WA, July 1991.
- [WEH93] L. Wehenkel and M. Pavella, **Decision Tree Approach to Power Systems Security Assessment**, Int. Journal of Electric Power and Energy Systems, 15, nº 1, pp. 13-36.
- [WEH94] L. Wehenkel, T. Van Cutsem, M. Pavella, B. Heilbronn and P. Pruvot, **Machine Learning, Neural Networks and Statistical Pattern Recognition for Voltage Security: A Comparative Study**, Proc. of the 1st International Conference on Intelligence Systems Applications to Power Systems, pp. 521-528, Montpellier - France, Sep 5-9, 1994.
- [WEH96] L. Wehenkel, **Contingency Severity Assessment for Voltage Security using Non-Parametric Regression Techniques**, IEEE Trans. on Power Systems, Vol. 11, Nº 1, Feb. 1996, pp. 101-111.
- [WEH97] L. Wehenkel and Y. Jacquemart, **Tutorial Course on Automatic Learning Methods Application to Dynamic Security Assessment**, Presented at the IEEE PICA 97, May 12, 1997, Columbus, Ohio, USA.
- [WEI91] S. M. Weiss and C. A. Kulikowski, **Computer Systems that Learn**, Morgan Kaufmann Publishers, Inc, 1991.
- [YAN91] H. H. Yan, J-C Chow, M. Kam, R. Fischl and C. R. Sepich, **Hybrid Expert System/Neural Network Hierarchical Architecture for Classifying Power System Contingencies**, Proc. 1st International Forum on Applications of Neural Networks to Power Systems (ANNPS), pp. 76-82, Seattle - WA, July 1991.
-

-
- [YAN94] H. H. Yan, R. Fischl and J. C. Chow, **Indices for Evaluation of Neural Network Performance in Power System Security Assessment**, Proc. of the 1st International Conference on Intelligence Systems Applications to Power Systems, pp. 187-191, Montpellier - France, Sep 5-9, 1994.
- [ZAY96] M. B. Zayan, M. A. El-Sharkawi and N. R. Prasad, **Comparative Study of Feature Extraction Techniques for Neural Network Classifier**, Proc. of the 2nd International Conference on Intelligence Systems Applications to Power Systems, 1996.
-