

**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

**José Paulo Freiman**

**UTILIZAÇÃO DE REDES NEURAIS ARTIFICIAIS NA  
PREVISÃO DE INDICADORES FINANCEIROS PARA  
AVALIAÇÃO ECONÔMICA DE NEGÓCIOS EM  
SITUAÇÃO DE RISCO**

Dissertação submetida ao Programa de Pós-  
Graduação em Engenharia de Produção como  
requisito parcial à obtenção do título de *Mestre em  
Ciências em Engenharia de Produção*

**Orientador: Prof. Edson de Oliveira Pamplona, Dr.**

**Co-orientador: Prof. Pedro Paulo Balestrassi, Dr.**

**Itajubá**

**2004**

**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

**José Paulo Freiman**

**UTILIZAÇÃO DE REDES NEURAS ARTIFICIAIS NA  
PREVISÃO DE INDICADORES FINANCEIROS PARA  
AVALIAÇÃO ECONÔMICA DE NEGÓCIOS EM  
SITUAÇÃO DE RISCO**

Dissertação aprovada por banca examinadora em     de dezembro de 2004, conferindo ao autor o título de *Mestre em Ciências em Engenharia de Produção*

**Banca examinadora:**

Prof. Dr. Marcelo Lacerda Resende - FACESM

Prof. Dr. José Arnaldo Barra Montevechi - UNIFEI

Prof. Dr. Pedro Paulo Balestrassi - UNIFEI (Co-orientador)

Prof. Dr. Edson de Oliveira Pamplona - UNIFEI (Orientador)

**Itajubá**

**2004**

## **Dedicatória**

À minha esposa Marisa, pelo apoio, carinho e participação, compartilhando sempre do meu esforço e do sacrifício dos momentos de convívio, em prol do enfrentamento deste desafio.

Aos meus filhos Paulo Felipe e Bruno pelo incentivo e inspiração, fazendo-me retomar com saudosismo e prazer a prática discente.

## Agradecimentos

A Deus, pela proteção e condução segura em todos os momentos;

À minha família pelo apoio e estímulo constantes;

Aos meus companheiros de mestrado por dividirmos os momentos de incertezas e alegrias;

Aos professores Edson de Oliveira Pamplona e Pedro Paulo Balestrassi, pela orientação amiga, segura e objetiva;

E a todos aqueles que direta ou indiretamente participaram deste trabalho.

# SUMÁRIO

Dedicatória .....	ii
Agradecimentos .....	iii
Sumário .....	iv
Resumo .....	vii
Abstract .....	viii
Lista de Figuras .....	ix
Lista de Tabelas .....	x
<b>1. INTRODUÇÃO .....</b>	<b>01</b>
1.1 Considerações iniciais .....	01
1.2 Justificativa e relevância do tema .....	02
1.3 Caracterização da atividade de pesquisa .....	02
1.4 Objetivos .....	02
1.4.1 Objetivo geral .....	02
1.4.2 Objetivos específicos .....	02
1.5 Metodologia de pesquisa .....	03
1.6 Estrutura da dissertação .....	03
<b>2. REDES NEURAIS ARTIFICIAIS E SUA APLICAÇÃO EM</b>	
<b>SÉRIES TEMPORAIS .....</b>	<b>04</b>
2.1 Considerações iniciais .....	04
2.2 O risco na avaliação de investimentos .....	04
2.3 Fundamentos das redes neurais artificiais .....	07
2.3.1 Principais propriedades das redes neurais artificiais .....	09
2.3.2 Histórico .....	09
2.3.3 Aplicações das redes neurais artificiais .....	12
2.3.4 O neurônio artificial .....	15
2.3.5 Funções de ativação .....	17
2.3.6 Topologia das redes neurais artificiais .....	20
2.3.7 A rede neural artificial multicamadas .....	23
2.3.8 Processos de aprendizado de uma rede neural artificial .....	23
2.3.9 Algoritmo de retropropagação do erro .....	26
2.4 Redes neurais artificiais na previsão de séries temporais .....	28
2.4.1 Séries temporais .....	28

2.4.2	Procedimentos estatísticos de previsão .....	30
2.4.3	Previsão de séries temporais .....	31
2.5	Considerações finais .....	33
<b>3</b>	<b>TÓPICOS SOBRE O PROJETO DE UM SISTEMA NEURAL .....</b>	<b>34</b>
3.1	Considerações iniciais .....	34
3.2	Modelagem da rede neural artificial .....	34
3.2.1	Etapas no desenvolvimento de um modelo de RNA .....	35
3.2.2	Configuração da rede neural artificial .....	37
3.2.3	Treinamento e aprendizado .....	38
3.2.4	Fatores de desempenho .....	40
3.2.5	Controle dos fatores determinantes no desempenho das RNA's .....	41
3.2.5.1	Padronização dos dados .....	41
3.2.5.2	Funções de ativação .....	43
3.2.5.3	Funções erro .....	44
3.2.5.4	Parâmetros de controle do gradiente descendente .....	47
3.2.5.5	Arquitetura da RNA .....	47
3.3	Teste .....	48
3.4	Avaliação dos resultados .....	48
3.5	Integração .....	49
3.6	Considerações finais .....	49
<b>4</b>	<b>APLICAÇÃO DA RNA : PREVISÃO DA COTAÇÃO DA COMMODITY “ARROBA DE CARNE BOVINA” .....</b>	<b>50</b>
4.1	Considerações iniciais .....	50
4.2	Exemplo de aplicação .....	50
4.3	Configuração da rede .....	51
4.3.1	Camada de entrada .....	51
4.3.1.1	Mês da previsão .....	52
4.3.1.2	Cotações de meses anteriores .....	52
4.3.1.3	Taxa de inflação .....	52
4.3.1.4	Taxa de juros .....	53
4.3.2	Camada intermediária .....	53
4.3.3	Camada de saída .....	55
4.4	Treinamento e teste .....	55

4.4.1	Seleção dos dados .....	55
4.4.2	Implementação .....	56
4.4.3	Treinamento .....	59
4.4.4	Teste da rede .....	61
4.5	Estatística de comparação .....	61
4.6	Avaliação do modelo proposto .....	64
4.7	Considerações finais .....	67
<b>5</b>	<b>CONCLUSÕES E RECOMENDAÇÕES PARA TRABALHOS</b>	
	<b>FUTUROS</b> .....	68
5.1	Considerações iniciais .....	68
5.2	Conclusões .....	68
5.3	Limitações .....	69
5.4	Recomendações para trabalhos futuros .....	70
	<b>ANEXO A - Dados do conjunto de treinamento e teste</b> .....	73
	<b>ANEXO B - Resultados da análise de regressão múltipla</b> .....	76
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	81

## Resumo

A previsão ou predição está vinculada à utilização do conhecimento de eventos presentes e passados para se realizar estimativas para eventos futuros. Isto é particularmente importante na Economia, campo em que temos um sistema dinâmico com características não lineares adquiridas através das tendências de longo prazo, padrões sazonais, movimentos cíclicos e fatores irregulares.

Atualmente, no campo dos negócios, as redes neurais artificiais (RNA) estão se tornando cada vez mais populares, atuando como uma ferramenta de grande importância como auxiliar na interpretação do mercado. Muitos sistemas que utilizam redes neurais têm funcionado bem na identificação de padrões complexos, aprendendo pela experiência, concluindo sobre algo e realizando previsão.

O principal objetivo deste trabalho é mostrar as características comuns das redes neurais e discutir a possibilidade de sua utilização no campo dos negócios, particularmente no agronegócio, realizando previsões de preços de *commodities* e comparando-as com aquelas obtidas com o uso de um método estatístico tradicional.



## **Abstract**

Prediction is concerned with using the knowledge of present and past events to make calculated estimates of future events. This is particularly important in Economy, field where we have a dynamic system that inherits nonlinearity through long term trends, seasonal patterns, cyclical movements and irregular factors.

In today's business fields, Artificial Neural Networks (ANN) are becoming increasingly popular acting as a tool of prime importance in helping interpret the market. Many neural-net systems have been shown to work well in identifying intricate patterns, learning from experience, reaching some conclusion and making prediction.

The main goal of this work is to show the characteristics of neural networks and discuss the feasibility of neural-net applications in business fields, particularly agrobusiness, making predictions of commodities prices and comparing them with results obtained using a traditional statistical method.

## Lista de Figuras

Figura 1.1	Roteiro da pesquisa .....	03
Figura 2.1	Risco Baixo .....	07
Figura 2.2	Risco Alto .....	07
Figura 2.3	Computação neural .....	08
Figura 2.4	O neurônio biológico .....	15
Figura 2.5	Representação de um neurônio artificial .....	17
Figura 2.6	Funções de ativação .....	19
Figura 2.7	Exemplo de uma RNA não recorrente .....	21
Figura 2.8	Exemplo de uma RNA recorrente .....	22
Figura 3.1	Desenvolvimento de um projeto baseado em RNA .....	36
Figura 3.2	Estrutura de uma RNA e seus parâmetros de projeto .....	37
Figura 3.3	Aprendizado: erro em relação ao conjunto de aprendizado e de teste .....	39
Figura 4.1	Modelo de rede neural MLP com três camadas .....	51
Figura 4.2	Diagrama em blocos da rede proposta .....	54
Figura 4.3	Tela inicial da interface gráfica da <i>toolbox</i> de Redes Neurais do MATLAB .....	57
Figura 4.4	Janela de criação da rede neural .....	58
Figura 4.5	Estrutura da rede neural definida na janela de criação da rede .....	58
Figura 4.6	Janela de definição dos parâmetros de treinamento da rede .....	59
Figura 4.7	Gráfico do resultado do treinamento .....	60
Figura 4.8	Melhor resultado da simulação .....	61
Figura 4.9	Equação de regressão .....	62
Figura 4.10	Equação de regressão após a retirada da variável @ - 2 e <i>outliers</i> .....	63
Figura 4.11	Histograma dos resíduos .....	64
Figura 4.12	Análise da correlação entre as variáveis .....	65
Figura 5.1	Resultados da Análise de Regressão Múltipla .....	76
Figura 5.2	Análise gradual dos elementos da equação de regressão .....	78
Figura 5.3	Melhores subconjuntos para a ARM .....	78
Figura 5.4	Gráfico de resíduos x ordem dos dados .....	79
Figura 5.5	Gráfico de resíduos x valores ajustados .....	79
Figura 5.6	Gráfico da probabilidade normal dos resíduos .....	80

## Lista de Tabelas

Tabela 4.1	Resultados da previsão da RNA e da ARM .....	65
Tabela 4.2	Resultados da previsão com a ARM reduzida da variável @-2 e outliers ..	66
Tabela 4.3	Terminologia correspondente entre RNA e Estatística .....	66
Tabela 5.1	Matriz do conjunto de treinamento .....	73
Tabela 5.2	Matriz do conjunto de teste .....	75
Tabela 5.3	Valores críticos de Durbin-Watson .....	80

# CAPÍTULO 1

## 1. INTRODUÇÃO

### 1.1 Considerações iniciais

É próprio do ser humano especular sobre o futuro. Que o digam os profetas, tarólogos, ciganas, videntes, adivinhos e futurólogos de toda sorte. É o mistério e a magia que envolvem o novo, o desconhecido e o inesperado. É quase impossível viver sem pensar no futuro, pelo menos conjecturar sobre alguma ocorrência num prazo qualquer adiante. A preocupação com o amanhã estimula a criatividade. “A história se repete”, dizem alguns e a análise de certos eventos por vezes reforçam esta idéia, tornando-a quase uma verdade absoluta, um dogma.

Em qualquer área do conhecimento ou da atividade humana, planejar, pensar sobre o futuro, sempre consumiu muito de nosso tempo e de nosso raciocínio.

Como estabelecer relações entre os eventos passados e concluir sobre hipóteses futuras?

Buscando responder a perguntas como essas, os pesquisadores têm estudado diferentes métodos que possam ajudar a concluir, com boa probabilidade de acerto, previsões sobre os mais variados assuntos, em particular aqueles ligados à tomada de decisões no campo da economia e das finanças.

Nesse contexto, destacamos o agronegócio, pela sua importância como fonte geradora de riquezas para o país. Tem sido crescente a participação dos produtos do *agrobusiness* na pauta de exportações brasileiras, cuja participação global no comércio internacional ainda é bastante modesta, incompatível com a grandiosidade e potencial deste país. Alguns produtos, entretanto, têm mostrado um ritmo vigoroso de crescimento, tanto na produção quanto na produtividade e destacam-se como importantes geradores de divisas. Neste grupo estão inseridos as *commodities* agrícolas e assemelhados, entre elas a carne bovina, na qual o Brasil constitui-se no maior exportador mundial, posição adquirida mercê de intensos estudos e pesquisas por parte de nossos especialistas, particularmente aqueles dedicados ao melhoramento genético de matrizes.

Procurou-se neste trabalho focar dois tópicos vinculados aos assuntos citados: previsão e agronegócio. A eles foi acrescentada uma ferramenta de amplo emprego: as redes neurais artificiais e sua capacidade de previsão.

## 1.2 Justificativa e relevância do tema

Várias áreas do conhecimento com necessidade de realizar previsões têm utilizado as redes neurais artificiais (RNA) como uma ferramenta de apoio. Em análise de investimentos, a previsão de indicadores econômico-financeiros é fundamental para a tomada de decisões, e a utilização das RNA poderá trazer significativa contribuição nessa atividade.

A principal motivação para o uso de redes neurais na previsão de séries temporais seria dispor de um método para solução de problemas de previsão como alternativa aos procedimentos estatísticos convencionais, os quais possuem dificuldades no tratamento de complexas relações não lineares.

## 1.3 Caracterização da atividade de pesquisa

A presente pesquisa foi desenvolvida junto ao Programa de Pós-graduação em Engenharia de Produção da Universidade Federal de Itajubá (UNIFEI).

Curso de Mestrado em Engenharia de Produção.

**Área de concentração:** Economia e Finanças.

**Linha de pesquisa:** Economia e Finanças.

**Projeto de Pesquisa:** Análise Econômica de Decisões.

## 1.4 Objetivos

### 1.4.1 Objetivo geral

Apresentar a técnica de RNA como alternativa para a solução de problemas de previsão de indicadores financeiros para análise de investimentos.

### 1.4.2 Objetivos específicos

- ◆ Executar o projeto de uma rede neural para realizar a previsão de uma série temporal vinculada à área de Economia e Finanças.
- ◆ Estabelecer comparações entre a RNA e as técnicas estatísticas convencionais na avaliação de investimento.

## 1.5 Metodologia de pesquisa

A pesquisa a ser conduzida é do tipo simulação, para a qual se utilizará o roteiro a seguir:

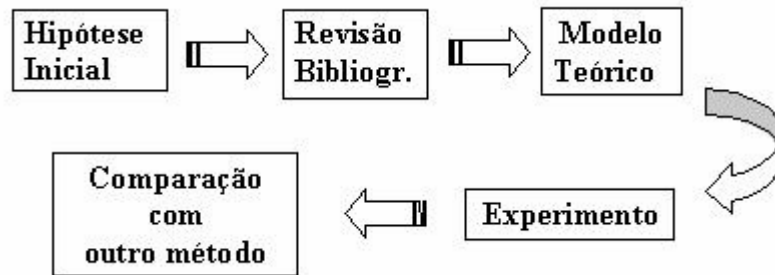


Figura 1.1 - Roteiro da pesquisa

## 1.6 Estrutura da dissertação

Esta dissertação está estruturada em cinco capítulos e dois anexos.

O Capítulo 1 constitui a parte introdutória.

O Capítulo 2 é destinado a apresentar a revisão da literatura que enfoca o conceito de risco na avaliação de investimentos e a fundamentação de redes neurais artificiais.

O Capítulo 3 descreve as etapas a serem seguidas no desenvolvimento de um modelo de RNA.

O Capítulo 4 trata da aplicação da RNA em um caso prático, mostrando através de uma simulação como criar, treinar e testar uma rede com o objetivo de executar a previsão da cotação da *commodity* arroba de carne bovina para três meses à frente. A mesma previsão será realizada através de um método estatístico como referência para avaliação dos resultados.

O Capítulo 5 fará as considerações finais a respeito do desenvolvimento dessa pesquisa e apresentará propostas para trabalhos futuros dentro do mesmo tema.

O Anexo A mostra os dados utilizados para montagem do conjunto de treinamento e teste da rede neural.

O Anexo B mostra a documentação correspondente à Análise de Regressão Múltipla utilizada como modelo estatístico de previsão.

## CAPÍTULO 2

### 2 REDES NEURAIS ARTIFICIAIS E SUA APLICAÇÃO EM SÉRIES TEMPORAIS

#### 2.1 Considerações iniciais

Toda sociedade econômica, qualquer que seja sua forma de trabalho, enfrenta três problemas fundamentais e determinados pela lei da escassez: **o que** e em que quantidade produzir; **como** produzir, com que recursos tecnológicos, financeiros, etc.; **para quem** produzir, ou seja, para quem deverão ser distribuídos os diversos bens produzidos (ASSAF NETO, 1999).

Essa tríade de problemas é comum a todas as sociedades econômicas, porém respondida de maneira diferente segundo sejam suas condições e verdades. As sociedades econômicas mais liberais costumam delegar ao mercado a solução de seus problemas econômicos básicos, por meio de um sistema de preços e de livre comércio.

Dentro desse contexto, sabemos que os recursos da economia são movimentados no mercado, em sua maior parte, por intermediários financeiros, que trabalham de forma especializada e voltados para ajustar expectativas e interesses de agentes econômicos com capacidade de poupança e os tomadores de recursos. Essa intermediação se processa pela colocação de títulos e valores econômicos no mercado por meio de instituições, como bancos, corretoras de valores, fundos de pensão, entre outras.

O mercado financeiro cumpre sua finalidade quando permite eficiente interação entre poupadores e tomadores de recursos, promovendo investimentos e crescimento da economia. A intermediação financeira nesse mercado deve permitir a aproximação entre os vários agentes econômicos, promovendo transferências de poupanças a um custo mínimo e a um nível reduzido de risco, cuja avaliação é fator primordial no contexto maior da análise de investimentos. Como parte do processo de avaliação da magnitude do risco é que são efetuadas previsões de indicadores financeiros, baseadas, por exemplo, em séries históricas de resultados, fornecendo subsídios para os tomadores de decisão. Dentre os diferentes métodos de previsão, destaca-se neste trabalho aquele que utiliza as redes neurais artificiais, motivado

pelo uso crescente desta ferramenta nos últimos anos nas mais variadas áreas, particularizando a sua aplicação na análise de investimentos com risco.

## 2.2 O risco na avaliação de investimentos

Na prática do mercado, as decisões financeiras não são tomadas em ambiente de total certeza com relação a seus resultados. Em verdade, por estarem essas decisões fundamentalmente voltadas para o futuro, é imprescindível que se introduza a variável **incerteza** como um dos mais significativos aspectos do estudo das operações do mercado financeiro.

Conforme Assaf Neto (1999), toda vez que a incerteza associada à verificação de determinado evento possa ser quantificada por meio de uma distribuição de probabilidades dos diversos resultados previstos, diz-se que a decisão está sendo tomada sob uma **situação de risco**. Dessa maneira, o risco pode ser entendido pela capacidade de se mensurar o estado de incerteza de uma decisão mediante o conhecimento das probabilidades associadas à ocorrência de determinados resultados ou valores.

A idéia de risco, de forma mais específica, está diretamente associada às probabilidades de ocorrência de determinados resultados em relação a um valor médio esperado. É um conceito voltado para o futuro, revelando uma possibilidade de perda. Daí a necessidade de se fazer **previsões e estimativas**, baseadas em fatos passados e tendências sinalizadas pelo mercado.

No estudo de avaliação de investimentos e risco, é comum que seus vários modelos sejam construídos e discutidos conceitualmente a partir das hipóteses de um mercado eficiente. Esse procedimento é adotado com o intuito de facilitar a realização dos testes empíricos dos modelos, avaliando seus resultados quando aplicados a uma situação prática. Em síntese, o **risco** é interpretado pelo nível de incerteza associado a um acontecimento (evento).

Na abrangência do entendimento do risco, a avaliação de uma empresa limita-se aos componentes de seu risco total: *econômico* e *financeiro*. As principais causas determinantes do **risco econômico** são de natureza conjuntural (alterações na economia, tecnologia, etc.), de mercado (crescimento da concorrência, por exemplo) e do próprio planejamento e gestão da empresa (vendas, custos, preços, investimentos, etc.). O **risco financeiro**, de outro modo, está mais diretamente relacionado com o endividamento (passivos) da empresa, sua



capacidade de pagamento, e não com as decisões de ativos, conforme definidas para o risco econômico.

Dessa maneira, pode-se inferir que o risco total de qualquer ativo é definido pela sua parte *sistemática* (risco sistemático ou conjuntural, também conhecido como risco de mercado) e *não sistemática* (risco específico ou próprio do ativo).

O risco *sistemático* é inerente a todos os ativos negociados no mercado, sendo determinado por eventos de natureza política, econômica e social, tais como PIB, taxas de juros ou inflação, entre outros. Cada ativo comporta-se de forma diferente diante da situação conjuntural estabelecida. Não há como se evitar totalmente o risco sistemático, sendo indicada a diversificação da carteira de ativos como medida preventiva para redução desse risco.

O risco definido por *não sistemático* é identificado nas características do próprio ativo, não se alastrando aos demais ativos da carteira. É um risco intrínseco, próprio de cada investimento realizado, e sua eliminação de uma carteira é possível pela inclusão de ativos que não tenham correlação positiva entre si.

Dessa maneira, o risco total de qualquer ativo é mensurado da forma seguinte:

$$\text{Risco Total} = \text{Risco Sistemático} + \text{Risco Não Sistemático}$$

A mensuração do risco de um investimento processa-se, geralmente, por meio do critério probabilístico, o qual consiste em atribuir probabilidades - subjetivas ou objetivas - aos diferentes estados de natureza esperados e, em consequência, aos possíveis resultados do investimento. Dessa maneira, é delineada uma distribuição de probabilidades dos resultados esperados, e mensuradas suas principais medidas de dispersão e avaliação do risco.

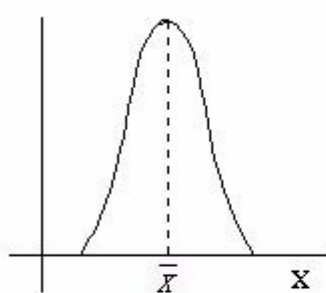
A probabilidade *objetiva* pode ser definida a partir de séries históricas de dados e informações, frequências relativas observadas e experiência acumulada no passado. A probabilidade *subjetiva*, por seu lado, tem como base a intuição, o conhecimento, a experiência do investimento e, até mesmo, um certo grau de crença na unidade tomadora de decisão.

Nesse ambiente, o risco pode ser interpretado pelos desvios previsíveis dos fluxos futuros de caixa resultantes de uma decisão de investimento, encontrando-se associado a fatos considerados como de natureza incerta. Sua principal medida estatística é a variabilidade dos resultados esperados de caixa em relação à média. O retorno esperado está vinculado aos

fluxos incertos de caixa do investimento, sendo aquele determinado pela ponderação entre os valores financeiros esperados e suas respectivas probabilidades de ocorrência.

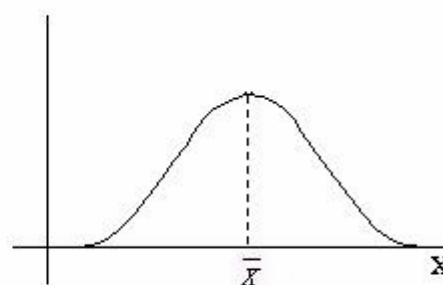
As figuras 2.1 e 2.2 retratam diferentes graus de dispersão dos valores em relação à média da distribuição, indicando, em conseqüência, diferentes níveis de risco aos eventos.

A mais elevada medida de dispersão (variância e desvio-padrão) revela seu maior grau de risco. O desvio-padrão como medida de risco é determinada pela decisão de investimento ser tomada a partir da média dos possíveis retornos de um ativo. Indica, em outras palavras, se o valor médio esperado dos retornos é representativo da distribuição apresentada pelo investimento em avaliação.



**Menor dispersão**

Figura 2.1 - Risco Baixo



**Maior dispersão**

Figura 2.2 - Risco Alto

Estas considerações baseadas em procedimentos estatísticos convencionais caracterizam a **abordagem clássica** na avaliação de investimentos em situação de risco. Elas trazem embutidas **relações eminentemente lineares**, possuindo dificuldade no tratamento de dados reais compostos de complexas **relações não lineares**. Para estas, têm-se atualmente as **redes neurais artificiais**, ferramenta que vem se mostrando de grande utilidade como instrumento de análise de dados históricos para estabelecimento de previsão e estimativa de resultados futuros. Devemos entender a utilização das redes neurais artificiais como uma interessante complementação aos procedimentos estatísticos tradicionais já tão amplamente utilizados, e não como concorrente àqueles.

### 2.3 Fundamentos das redes neurais artificiais

As *redes neurais artificiais* (RNA) são ferramentas de Inteligência Artificial que possuem a capacidade de se adaptar e de aprender a realizar uma certa tarefa, ou comportamento, a partir de um conjunto de exemplos dados (OSÓRIO & BITTENCOURT, 2000).

Na sua forma mais geral, uma *rede neural* é uma máquina que é projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse; a rede é normalmente implementada utilizando-se componentes eletrônicos ou é simulada por programação em um computador digital (HAYKIN, 1999). Na fig. 2.3 caracterizamos a computação neural como uma forma alternativa de computação que procura imitar o funcionamento do cérebro humano na solução de problemas de reconhecimento de padrões.



Figura 2.3 - Computação neural  
Fonte: Osório & Bittencourt (2000)

As redes neurais nada mais são do que ferramentas de aproximação de funções que aprendem a relação entre variáveis independentes e dependentes, à semelhança da *regressão* ou outras abordagens mais tradicionais.

A principal diferença entre redes neurais e as abordagens estatísticas está no fato de que as redes neurais não estabelecem hipóteses ou suposições sobre a distribuição ou propriedades dos dados e, desta forma, tendem a ser mais úteis em situações práticas. As RNA's constituem também uma abordagem inerentemente *não linear*, fornecendo mais precisão quando modelando dados de padrões complexos. Existem vários tipos de redes, cada uma com diferentes objetivos, arquitetura e algoritmo de aprendizagem (SMITH & GUPTA, 2000).

Uma rede neural é formada pela interconexão de um grande número de unidades de processamento não linear denominadas *neurônios*, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:

1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.

2. Forças de conexão entre neurônios, conhecidas como **pesos sinápticos**, são utilizadas para armazenar o conhecimento adquirido.

A estrutura interna de uma rede neural pode envolver somente caminhos diretos, somente caminhos indiretos ou ambos.

### 2.3.1 Principais propriedades das redes neurais artificiais

- **Não-linearidade** - essa propriedade é atribuída às RNA's devido à característica não linear dos neurônios da rede, característica esta de grande utilidade, pois os mecanismos físicos de geração de grande parte dos sinais são não lineares (por exemplo: sinal de voz).
- **Poucas considerações estatísticas** - uma rede neural é capaz de “aprender” sobre as características do ambiente no qual ela opera, desde que os dados utilizados para o seu “aprendizado” sejam suficientemente representativos desse ambiente.
- **Aprendizado** - uma rede neural tem a capacidade de aprender sobre os dados por ela manipulados durante um período de treinamento dedicado a esse fim.

### 2.3.2 Histórico

As primeiras informações mencionadas sobre a neurocomputação (técnica baseada no comportamento das células nervosas biológicas) surgiram no início da década de 40. Em 1943, Warren McCulloch, psiquiatra e neuroanatomista, e Walter Pitts, um matemático recém-graduado, sugeriram a construção de uma máquina inspirada e baseada no cérebro humano, denominada **Psychon**, a partir de um modelo matemático (artificial) do neurônio biológico. O trabalho de Mc Culloch e Pitts concentrou-se mais em descrever um modelo artificial de um neurônio e apresentar suas capacidades computacionais, do que em apresentar técnicas de aprendizado (McCULLOCH & PITTS, 1943). Muitos outros artigos e livros surgiram desde então, porém, por um longo período pouco resultado foi obtido.

Em 1949, Donald O. Hebb escreveu um livro intitulado “*The Organization of Behaviour*” (A Organização do Comportamento) que perseguia a idéia de que o condicionamento psicológico clássico está presente em qualquer parte dos animais pelo fato de que esta é uma propriedade de neurônios individuais. Suas idéias não eram completamente novas, mas Hebb foi o primeiro a propor uma lei de aprendizagem específica para as sinapses dos neurônios. Este primeiro e corajoso passo serviu de inspiração para que muitos outros pesquisadores perseguissem a mesma idéia e influenciou vários modelos de RNA's de destaque na atualidade.

Também proveniente deste período foi a construção do primeiro neurocomputador, denominado **Snark**, por Marvin Minsky, em 1951. O Snark operava com sucesso a partir de um ponto de partida técnico, ajustando seus pesos automaticamente. Entretanto, ele nunca executou qualquer função de processamento de informação interessante, mas serviu de inspiração para as idéias de estruturas que o sucederam.

Em 1956, no Dartmouth College, nasceram os dois paradigmas da Inteligência Artificial (IA): a *simbólica* e a *conexionista*. A Inteligência Artificial Simbólica tenta simular o comportamento inteligente humano, desconsiderando os mecanismos responsáveis por tal. Já a Inteligência Artificial Conexionista acredita que, construindo-se um sistema que simule a estrutura do cérebro, este sistema apresentará inteligência, ou seja, será capaz de aprender com os seus erros e assimilar.

O primeiro neurocomputador a obter sucesso (**Mark I Perceptron**) foi criado por Frank Rosenblatt, Charles Wightman e outros, o qual posteriormente deu origem aos modelos denominados MLP - Multi-Layer Perceptron (Perceptron de Múltiplas Camadas). Devido à profundidade de seus estudos, suas contribuições técnicas e sua maneira de pensar, estes cientistas são vistos por muitos como os fundadores da neurocomputação na forma em que a temos hoje. Seu interesse inicial para a criação do **Perceptron** era o reconhecimento de padrões.

Alguns outros modelos similares ao Perceptron foram também desenvolvidos nesta época, como é o caso do **Adaline** (Adaptive Linear Element), criado por Bernard Widrow em 1962. Este tipo de elemento de processamento de redes neurais era equipado com uma poderosa lei de aprendizado, que diferente do Perceptron, ainda permanece em uso. Widrow também fundou a primeira companhia de hardware de neurocomputadores e componentes. Os modelos do tipo Perceptron, incluindo o Adaline, são baseados no aprendizado supervisionado por correção de erros, uma classe muito importante de redes neurais artificiais que possui uma larga aplicação na atualidade.

Em 1969 os modelos baseados no Perceptron receberam uma dura crítica feita por Minsky e Papert através de sua obra "*Perceptrons: An Introduction to Computational Geometry*", em que estes provaram matematicamente que os modelos de redes neurais baseados no Perceptron (redes de um só nível, o que na época era o tipo de rede de Perceptrons utilizado), não eram capazes de aprender uma simples função lógica do tipo "ou-exclusivo" ( $XOR = Exclusive Or$ ). A função XOR possui um padrão de valores de entrada e de saída cuja associação não podia ser aprendida pelos modelos de redes baseados em

Perceptron disponíveis naquela época. O impacto da publicação desta obra abalou profundamente as pesquisas realizadas nesta área de estudos.

Um período de pesquisas silenciosas seguiu-se até a década de 80, quando surgiram novos modelos que deram um novo impulso às redes neurais, estabelecendo um campo concreto para o renascimento da área. Em 1982, surgia um modelo importante de rede criado por John Hopfield, renomado físico de reputação mundial, que se interessou pela neurocomputação e escreveu artigos que percorreram o mundo todo persuadindo centenas de cientistas, matemáticos e tecnólogos altamente qualificados a se unirem nesta nova área emergente.

O modelo que Hopfield criou era baseado em um tipo de rede diferente dos modelos que tinham como base o Perceptron. Tratava-se de uma rede com conexões recorrentes e com um comportamento baseado na competição entre os neurônios, em que o aprendizado era *não supervisionado*. Outros modelos similares ao de Hopfield surgiram pouco depois, e entre esses podemos citar alguns como, por exemplo, a máquina de Boltzmann e o BAM (Binary Associative Memory).

A década de 80 ficou também marcada profundamente pelo reaparecimento das redes baseadas em Perceptrons. Isto deveu-se ao desenvolvimento dos computadores, que eram mais velozes e permitiam realizar melhores simulações das redes neurais, bem como o desenvolvimento de modelos matemáticos que permitiram a solução do problema apontado por Minsky e Papert. Também podemos associar em parte este renascimento das redes neurais ao suposto desencanto com a Inteligência Artificial clássica. O modelo que permitiu o ressurgimento das redes baseadas em Perceptrons foi o das redes multinível, onde o novo algoritmo de aprendizado chamado **Back-Propagation** (Retropropagação) desenvolvido por Rumelhart, Hinton e Williams, em 1986, resolveu em grande parte os problemas de aprendizado existentes até então. Naquele mesmo ano, foi publicado o célebre livro “*Parallel Distributed Processing: Explorations in the Microstructures of Cognition*” por Rumelhart e McClelland, que exerceu uma grande influência na utilização da aprendizagem por retropropagação, que emergiu como o algoritmo de aprendizagem mais popular para o treinamento de perceptrons de múltiplas camadas. Na verdade, a aprendizagem por retropropagação foi descoberta de modo independente em outros dois lugares, na mesma época (PARKER, 1985 ; LECUN, 1985). Este algoritmo, o Back-Propagation, permitia realizar o aprendizado por correção de erros em uma rede com múltiplas camadas (níveis) e, conseqüentemente, resolveria o problema do XOR.

Além dos modelos de Hopfield e do modelo de redes multinível com Back-Propagation (chamado de Multi-Layer Perceptron - MLP), outro modelo importante que surgiu nesta década foi o modelo de Kohonen, muito interessante, pois permite o aprendizado competitivo com uma auto-organização da rede neural, criando os chamados “*mapas de atributos auto-organizáveis*” (self-organizing maps).

Por fim, o último modelo de destaque neste período foi o modelo ART (Adaptive Resonance Theory) criado por Gail Carpenter e Stephen Grossberg. Este modelo possui um aprendizado do tipo *não supervisionado*, criando protótipos (*clusters*) dos padrões aprendidos. O modelo ART teve diversas versões posteriores, entre elas versões do tipo *semi-supervisionado* e com uso de conceitos de lógica nebulosa (Fuzzy-ART).

Os estudos sobre as redes neurais sofreram uma grande revolução a partir dos anos 80, conforme já citado. A partir daí, cada vez mais, esta área de estudos tem-se destacado, seja pelas promissoras características apresentadas pelos modelos de redes neurais propostos, ou seja pelas condições tecnológicas atuais de implementação que permitem desenvolver arrojadas arquiteturas neurais paralelas em hardwares dedicados, obtendo assim ótimas performances destes sistemas, bastante superiores aos sistemas convencionais.

Em 1987, ocorreu em São Francisco a primeira conferência de redes neurais em tempos modernos, a *IEEE International Conference on Neural Networks*, e também foi formada a *International Neural Networks Society (INNS)*. A partir destes acontecimentos, decorreram a fundação do *INNS Journal* em 1989, seguido do *Neural Computation* e do *IEEE Transactions on Neural Networks* em 1990.

Desde 1987, muitas universidades anunciaram a formação de institutos de pesquisa e programas de educação em neurocomputação.

### 2.3.3 Aplicações das redes neurais artificiais

As aplicações das redes neurais são inúmeras, atendendo a diferentes tipos de tarefas, tais como:

- reconhecimento de padrões (exemplo: reconhecimento de faces humanas);
- classificação de dados (exemplo: reconhecimento ótico de caracteres);
- predição (exemplo: previsão de séries temporais, como cotações em bolsa de valores ou também para diagnóstico médico);
- controle de processos e aproximação de funções (exemplo: robótica);
- análise e processamento de sinais;

- filtros contra ruídos eletrônicos;
- análise de imagens e de voz;
- avaliação de crédito.

Além dessas, há diversas outras possibilidades, já que as pesquisas na área estão em contínuo desenvolvimento, surgindo novas aplicações a cada dia.

Neste trabalho, em particular, desejamos destacar as aplicações das redes neurais artificiais na área de economia e finanças. Aplicações já bem consolidadas são citadas por Smith & Gupta (2000), como se segue:

#### a) **Marketing**

O objetivo das modernas técnicas de *marketing* consiste em identificar clientes que respondam positivamente a um produto e direcionar a propaganda para estes clientes. Isto envolve uma segmentação do mercado, em que o mesmo é dividido em grupos distintos de clientes com diferentes hábitos de consumo. Esta segmentação de mercado pode ser obtida utilizando-se redes neurais, separando-se a clientela a partir de características básicas, tais como: localização geográfica, condição sócio-econômica, poder aquisitivo e atitude em relação ao produto.

Uma vez definida a segmentação do mercado, o *marketing direto* pode ser utilizado para vender o produto sem a necessidade de ação intermediária como propaganda ou promoção.

As RNA's podem ainda ser usadas para monitorar o padrão de comportamento do cliente ao longo do tempo e detectar quando ele estiver propenso a optar pelo produto do concorrente. A análise das pesquisas de mercado é outra possibilidade de aplicação das redes neurais artificiais.

#### b) **Vendas no varejo**

Freqüentemente, faz-se necessária a previsão de vendas para que se possam tomar decisões sobre estoque, contratação de funcionários e preço do produto. A utilização de RNA tem tido muito sucesso na previsão de vendas, devido à sua habilidade para considerar simultaneamente múltiplas variáveis tais como: demanda do mercado pelo produto, capacidade de compra dos consumidores, tamanho da população e preço do produto. A previsão de vendas nos supermercados e centrais de atacadistas tem sido bastante estudada e



os resultados têm mostrado bom desempenho, quando comparados tanto com as técnicas estatísticas tradicionais, como a regressão, quanto com a opinião de especialistas.

### **c) Finanças**

Um dos principais setores da área de finanças influenciado pelas redes neurais é o de comércio e previsões financeiras. As RNA's têm sido aplicadas com sucesso em problemas de preço e *hedge* de derivativos de seguro, previsão de preço futuro para taxa de câmbio e seleção e previsão de desempenho de ações.

Os bancos têm usado por muito tempo uma técnica baseada na estatística para classificar os tomadores de empréstimo. Atualmente, as redes neurais constituem a técnica básica para auxiliar na tomada de decisão com respeito ao risco de crédito, bem como na previsão de falência das corporações. Uma área promissora é a que utiliza RNA's para avaliar as relações entre a estratégia, desempenho e saúde financeira das empresas. Da mesma forma, a sua utilização já é significativa para a detecção de fraudes na utilização de cartões de crédito e falsificação de assinaturas em cheques.

### **d) Seguros**

Na indústria de seguros, a questão de definição do valor dos prêmios também pode se beneficiar do uso das RNA's através da previsão da frequência dos pedidos de indenização. Da mesma forma que na área bancária e em outros setores financeiros, também aqui se faz necessária a detecção de fraudes a partir da análise de circunstâncias incomuns. Por ser uma indústria muito competitiva, o oferecimento de certos incentivos como redução do prêmio e carência são fortes atrativos para se manterem os bons clientes. Infelizmente, esta natureza competitiva da indústria de seguros permite que apenas poucos detalhes de aplicações bem sucedidas das RNA's nesta área tenham sido publicados.

### **e) Telecomunicações e Informática**

Como nas demais empresas de serviços altamente competitivas, a indústria de telecomunicações preocupa-se em manter a fidelidade de seus clientes. Para tal utiliza produtos (programas) baseados em RNA para analisar o perfil do consumidor, propondo, a partir daí, estratégias promocionais para conquista de novos clientes e identificando ainda os mais lucrativos para a empresa.

Além das aplicações tradicionais de RNA nos setores de engenharia da indústria de telecomunicações, existe um expressivo interesse em utilizá-las nas questões de otimização

envolvendo projeto de redes, controle de rotina de tráfego, etc. O uso das RNA's nos programas para identificação de vírus em computadores também já se tornou bastante comum.

#### f) Administração da Produção

A gestão das operações executadas nas indústrias, tais como planejamento de instalação das máquinas, linhas de montagem e células de manufatura utilizando RNA constituem tópicos de pesquisa que se têm tornado bastante comuns ao longo da última década. Outro setor desta área atendido pelo uso das RNA's é o de controle da qualidade, no qual elas podem ser integradas com as técnicas estatísticas tradicionais para melhorar o desempenho como ferramenta de diagnóstico para a detecção de falhas nos diversos níveis da produção.

O fato principal a ser destacado nessa discussão é a extensa diversidade das aplicações das redes neurais artificiais, trazendo benefícios para diferentes setores e estimulando as pesquisas no sentido de ampliar cada vez mais o seu emprego.

#### 2.3.4 O neurônio artificial

O cérebro humano é considerado o mais fascinante processador baseado em carbono existente, sendo composto por mais ou menos  $10^{11}$  neurônios de diversos tipos diferentes. Todas as funções e movimentos do organismo estão relacionados ao funcionamento destas pequenas células. A Figura 2.4 mostra o esquema de um **neurônio biológico** comum.

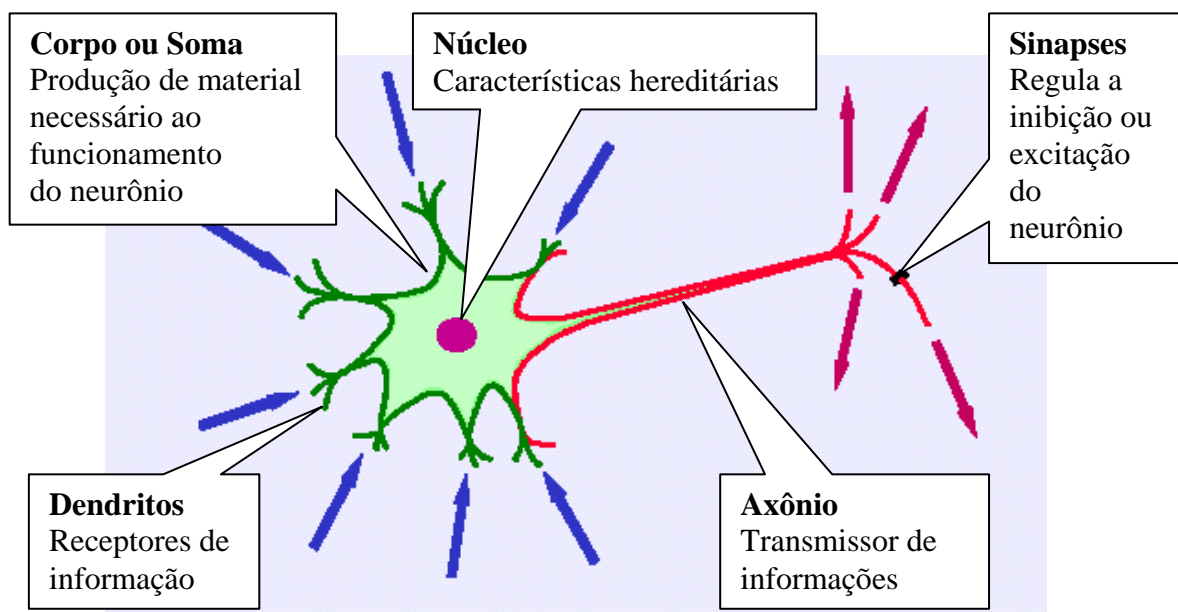


Figura 2.4 - O neurônio biológico.

Fonte: Falcão (2003)

O núcleo da célula está localizado no corpo da mesma, sendo este último também chamado de *soma*. Conectados ao corpo da célula estão fibras nervosas com estruturas similares a raízes, chamadas *dendritos*. Estendendo-se do corpo da célula existe uma única fibra nervosa mais grossa chamada *axônio*, da qual surgem ramificações e sub-ramificações. No fim destas ramificações, estão os pontos de transmissão para os outros neurônios, chamados de *junções sinápticas* ou *sinapses*. Os pontos de recepção onde as *sinapses* vão se conectar aos outros neurônios podem ser os *dendritos* ou mesmo o corpo da célula. O *axônio* de um neurônio típico possui algumas centenas de *sinapses* que irão se conectar com outros neurônios, formando uma grande rede, chamada **rede neural**. Esta grande rede proporciona uma fabulosa capacidade de processamento e armazenamento de informação.

A transmissão do sinal de uma célula para outra é um complexo processo químico, no qual substâncias específicas são liberadas pelo neurônio transmissor. As *sinapses* transmitem estímulos através de diferentes concentrações de **Na<sup>+</sup>** (sódio) e **K<sup>+</sup>** (potássio), e o efeito é um aumento ou uma queda no potencial elétrico no corpo da célula receptora. Se este potencial alcançar o limite de ativação da célula, um pulso de duração e potência fixas é enviado através do *axônio*. Diz-se então que o neurônio está ativo.

Nos neurônios, a comunicação é realizada através de impulsos. Quando um impulso é recebido, o neurônio o processa e, passado um limite de ação, dispara um segundo impulso que produz uma substância neurotransmissora, a qual flui do corpo celular para o axônio (que por sua vez pode ou não estar conectado a um dendrito de outra célula). O neurônio que transmite o pulso pode controlar a frequência de pulsos aumentando ou diminuindo a polaridade na membrana pós-sináptica. Os neurônios têm um papel essencial na determinação do funcionamento, comportamento e do raciocínio do ser humano. Ao contrário das redes neurais artificiais, redes neurais naturais não transmitem sinais negativos, sua ativação é medida pela frequência com que emite pulsos, frequência esta de pulsos contínuos e positivos. As redes naturais não são uniformes como as redes artificiais e apresentam uniformidade apenas em alguns pontos do organismo. Seus pulsos não são síncronos ou assíncronos, devido ao fato de não serem contínuos, o que as difere das redes artificiais.

Assim como o sistema nervoso é composto por bilhões de células nervosas, a rede neural artificial também seria formada por unidades que nada mais são do que pequenos módulos que simulam o funcionamento de um neurônio. Estes módulos devem funcionar de acordo com os elementos em que foram inspirados, recebendo e retransmitindo informações.

O **neurônio artificial** foi projetado para imitar as características de primeira ordem de um neurônio biológico. Essencialmente, um conjunto de entradas é aplicado ao neurônio artificial, cada uma delas representando a saída de outros neurônios (Figura 2.5). Cada entrada é multiplicada por um *peso* correspondente ( $W_{ij}$ ), gerando entradas ponderadas, de forma análoga à força das *sinapses*. Em seguida, todas estas entradas ponderadas são somadas, obtendo-se um valor **NET** que será comparado com o valor limite para ativação do neurônio. Caso esse valor alcance o valor limite de ativação do neurônio, ele se ativará; caso contrário, ele ficará inativo.

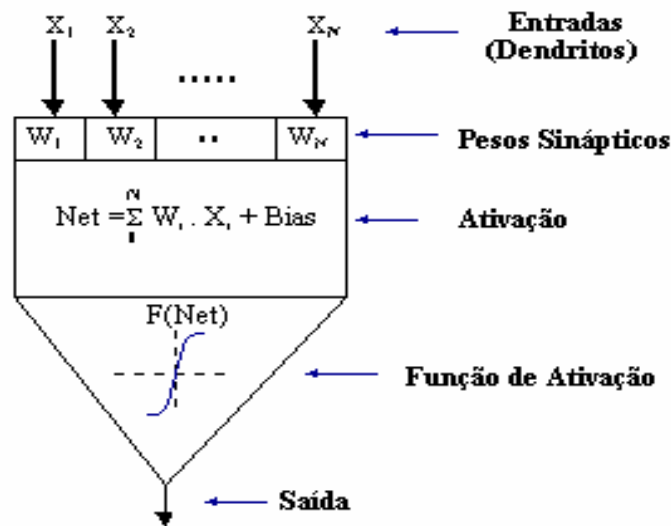


Figura 2.5 - Representação de um neurônio artificial

Fonte: Osório & Bittencourt (2000)

### 2.3.5 Funções de ativação

A função de ativação  $F$  é a que processa o sinal **NET** para produzir a saída final do neurônio (**SAÍDA**), causando um estado de ativação do neurônio (correspondente à frequência de descarga do neurônio biológico). Esta função pode ter várias formas: uma simples função linear, uma função limiar (função degrau) ou ainda uma função que simule mais precisamente as características não lineares do neurônio biológico. No caso de  $F$  ser uma função linear, a saída do neurônio é dada por

$$SAÍDA = K * (NET) \quad (2.1)$$

onde  $\mathbf{K}$  é uma constante. Caso  $\mathbf{F}$  seja uma função degrau (limiar), a saída seria:

$$\begin{aligned} \mathbf{SAÍDA} &= 1 && \text{Se } \mathbf{NET} > \mathbf{T} \\ \mathbf{SAÍDA} &= 0 && \text{Caso contrário} \end{aligned} \quad (2.2)$$

onde  $\mathbf{T}$  é o valor de limiar (*threshold*) do neurônio artificial (constante). A função básica de um neurônio é somar as entradas e retornar uma saída, caso esse valor seja maior que o valor de soma (*threshold*). No caso de se querer imitar com maior precisão os neurônios biológicos, usam-se funções não lineares, chamadas funções “*squashing*”. Elas caracterizam-se por não deixarem a SAÍDA exceder certos valores limites, os quais, em geral, são menores que os valores de NET. A função “*squashing*” mais usada é a *função logística sigmoidal* (Figura 2.6), a qual é representada matematicamente por  $F(x) = 1/(1+e^{-x})$ . Adequando-a ao neurônio artificial tem-se:

$$\mathbf{SAÍDA} = 1/(1+e^{-\mathbf{NET}}) \quad (2.3)$$

Pode-se pensar na função de ativação como definindo um ganho para o neurônio artificial, fazendo uma analogia com os sistemas eletrônicos analógicos. Este ganho é calculado encontrando-se a razão entre a variação da saída (SAÍDA) e a variação da entrada (NET). O ganho é a inclinação da curva para um nível de excitação específico, podendo variar desde um valor baixo para grandes excitações negativas, passando por valores altos para excitação zero, até voltar a cair quando a excitação se torna grande e positiva. Em 1973, Grossberg descobriu que esta característica de ganho não linear resolvia o problema do dilema da saturação com ruído. Neste dilema, Grossberg questionava como a mesma rede poderia manipular tanto sinais pequenos como grandes. Pequenos sinais necessitariam de um alto ganho através da rede para produzir saídas usáveis, exigindo um grande número de estágios de alto ganho. Contudo, estes estágios poderiam saturar a saída, amplificando o ruído (variáveis randômicas) que está presente em qualquer rede criada. Para sinais de entrada com valores altos, os estágios de alto ganho também iriam saturar a saída, eliminando qualquer saída usável. A região central de alto ganho da função sigmóide resolve este problema do processamento de pequenos sinais, enquanto que suas regiões de ganho decrescente nos extremos negativo e positivo são adequadas a grandes excitações.

Outra função de ativação “*squashing*” bastante usada é a tangente sigmoidal (Figura 2.6). Ela possui uma forma similar à da logística sigmoidal e é freqüentemente usada por

biólogos para o modelo matemático de ativação das células nervosas. Como função de ativação dos neurônios artificiais a função tem a forma:

$$\text{SAÍDA} = \text{tansig}(\text{NET}) \tag{2.4}$$

Apesar da forma semelhante à da logística sigmoidal, a tangente sigmoidal possui valores de SAÍDA bipolares, característica que é benéfica para certos tipos de RNA's.

Função	Equação (com polarização)	Gráfico (sem polarização)	Gráfico (com polarização)
<b>Degrau</b>	$y = \begin{cases} 1, & x > -b \\ 0, & x < -b \end{cases}$		
<b>Degrau Simétrico</b>	$y = \begin{cases} 1, & x > -b \\ -1, & x < -b \end{cases}$		
<b>Linear</b>	$y = x + b$		
<b>Logística Sigmoidal</b>	$y = \frac{1}{1 + e^{-(x+b)}}$		
<b>Tangente Sigmoidal</b>	$y = \frac{e^{(x+b)} - e^{-(x+b)}}{e^{(x+b)} + e^{-(x+b)}}$		

Figura 2.6 - Funções de ativação

De acordo com Abelém (1994), é interessante ressaltar que este modelo simples de neurônio artificial ignora diversas características do neurônio natural, tais como a não consideração dos atrasos de tempo que afetam a dinâmica do sistema - as entradas produzem saídas imediatas - e a não inclusão dos efeitos de sincronismo ou de modulação de frequência, característica que alguns pesquisadores acham de fundamental importância. Apesar destas limitações, as RNA's formadas por simples neurônios artificiais possuem atributos semelhantes aos do sistema biológico, como a capacidade de aprendizado e generalização, podendo-se dizer que a essência do funcionamento do neurônio natural foi absorvida.

### 2.3.6 Topologia das redes neurais artificiais

Um dos objetivos da pesquisa sobre redes neurais na computação é desenvolver morfologias neurais matemáticas, não necessariamente baseadas na biologia, que podem realizar funções diversas.

Quando um processo é criado visando utilizar aspectos de redes neurais, começam com o desenvolvimento de um neurônio artificial ou computacional baseado no entendimento de estruturas biológicas neurais, seguidas do aprendizado de mecanismos voltados para um determinado conjunto de aplicações. Ou em outras palavras, seguindo as três etapas:

- O desenvolvimento de modelos neurais motivado por neurônios biológicos;
- Modelos de estruturas e conexões sinápticas;
- O aprendizado das regras (um método de ajuste de pesos ou forças de conexões internodais).

Por causa de diferenças entre algumas, ou às vezes todas as entidades envolvidas, diferentes estruturas de redes neurais têm sido desenvolvidas por pesquisadores. Do ponto de vista estrutural, a arquitetura de redes neurais pode ser classificada como estática, dinâmica ou fuzzy, e de única camada ou múltiplas camadas. Além disso, diferenças computacionais surgem também quando se trata da maneira com que são feitas as conexões existentes entre os neurônios.

As várias topologias de redes neurais estão divididas, basicamente, em duas classes:

- ◆ Não recorrentes
- ◆ Recorrentes

As RNAs não recorrentes são aquelas que não possuem realimentação de suas saídas para suas entradas e por isso são também ditas “sem memória”. Sua estrutura é em camadas, podendo ser formada por uma ou mais camadas. Redes multicamadas contêm um conjunto de neurônios de entrada, uma camada de saída e uma ou mais camadas escondidas. A entrada não é considerada uma camada da rede, pelo fato de apenas distribuir os padrões, ou seja, onde os padrões são apresentados à rede. A camada com os neurônios que fornecem a saída da rede é chamada camada de saída, onde o resultado final é concluído e apresentado. As camadas que não possuem ligações diretas nem com a entrada, nem com a saída são chamadas de camadas ocultas (escondidas ou intermediárias), em que é feita a maior parte do processamento, através das conexões ponderadas, e podem ser consideradas como extratoras de características. Não existem conexões ligando um neurônio de uma camada a outro de uma camada prévia, nem a um neurônio da mesma camada (Figura 2.7). As RNA's de uma camada, também chamadas de *perceptrons*, por possuírem um espectro de representações limitado, não serão aqui tratadas. As RNA's multicamadas, por suprirem as deficiências das redes de uma única camada, serão as utilizadas neste trabalho.

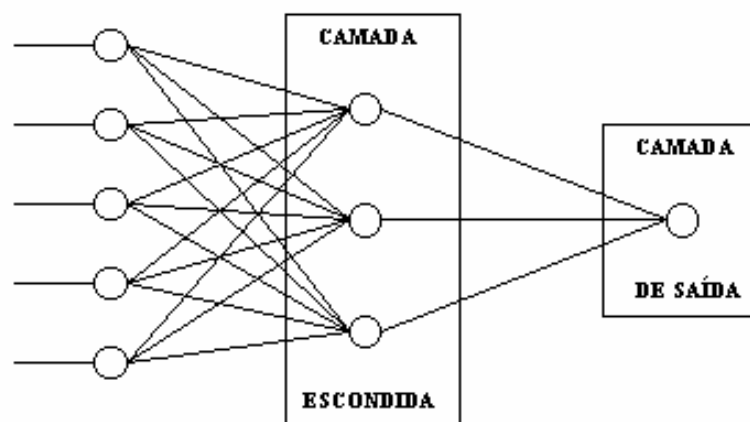


Figura 2.7 - Exemplo de uma RNA não recorrente.

As RNA's recorrentes são redes mais gerais que contêm realimentação das saídas para as entradas, sendo suas saídas determinadas pelas entradas atuais e pelas saídas anteriores. Além disso, suas estruturas não são obrigatoriamente organizadas em camadas e, quando são, estas redes podem possuir interligações entre neurônios da mesma camada e entre camadas não consecutivas, gerando interconexões bem mais complexas que as RNA's não recorrentes (Figura 2.8).



As RNA's recorrentes, também chamadas de RNA's com memória, por possuírem realimentação, respondem a estímulos dinamicamente, isto é, após aplicar uma nova entrada, a saída é calculada e então realimentada para modificar a entrada. Para as redes estáveis, este processo é repetido várias vezes, produzindo pequenas mudanças nas saídas, até estas ficarem constantes. Todavia, as RNA's recorrentes nem sempre são estáveis, mesmo com entradas constantes. O fato de não se conseguir prever quais redes seriam estáveis foi um problema que preocupou os pesquisadores até o início da década de 80, quando Cohen e Grossberg provaram um poderoso teorema para definir quando as RNAs eram estáveis. Este teorema determina que para as RNA's recorrentes alcançarem um estado estável é necessário que possuam conexões simétricas, melhor dizendo,  $W_{ij} = W_{ji}$ , sendo  $W_{ii} = 0$ . Contribuições importantes também foram dadas por John Hopfield, tanto para a teoria como para a prática, sendo que algumas configurações passaram a ser chamadas de redes de Hopfield em sua homenagem.

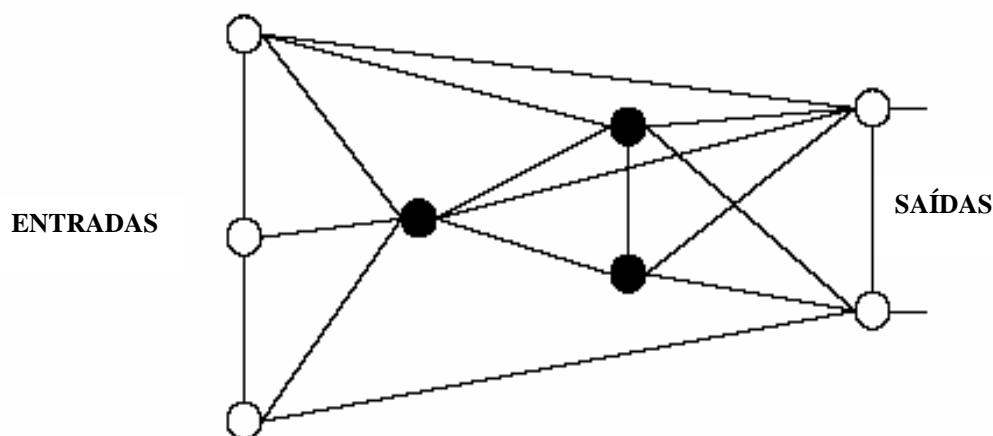


Figura 2.8 - Exemplo de uma RNA recorrente.

De acordo com Rumelhart (1986), a rede neural deve possuir no mínimo duas camadas: a de entrada de dados e a de saída dos resultados. Como a rede apresenta desempenho muito limitado com somente duas camadas, a adição de uma camada intermediária faz-se necessária. Neste tipo de configuração, cada neurônio está ligado com todos os outros das camadas vizinhas. Neurônios da mesma camada não se comunicam. A comunicação por ser unidirecional apresenta, assim, um comportamento estático.

Já a rede neural de Hopfield apresenta comportamento dinâmico e fluxo de dados multidirecional devido à integração total dos neurônios, desaparecendo a idéia de camadas

bem distintas. Com isso, seu funcionamento é mais complexo, havendo certas complicações, seja na fase de aprendizado ou na fase de testes. Seu uso é direcionado a problemas de minimização e otimização.

Hinton e Sejnowski (1986) também tiveram seus méritos, introduzindo regras gerais de treinamento para redes, denominadas por eles de máquina de Boltzmann, devido ao cálculo das saídas dos neurônios artificiais ser efetuado através de probabilidade com distribuição de Boltzmann.

### 2.3.7 A rede neural artificial multicamadas (Multilayer Perceptron - MLP)

A rede neural artificial é um sistema de neurônios ligados por conexões sinápticas e dividido em neurônios de entrada, que recebem estímulos do meio externo, neurônios internos ou ocultos (*hidden*) e neurônios de saída, que se comunicam com o exterior. A forma de se arranjar *perceptrons* em camadas é denominada **Multilayer Perceptron**, que foi concebida para resolver problemas mais complexos, os quais não poderiam ser resolvidos pelo modelo de neurônio básico. Um único *perceptron* ou uma combinação das saídas de alguns *perceptrons* poderia realizar uma operação **XOR** (Exclusive OR), porém seria incapaz de aprendê-la. Para isto são necessárias mais conexões, as quais só existem em uma rede de *perceptrons* dispostos em camada. Os neurônios internos são de suma importância na rede neural, pois provou-se que sem estes torna-se impossível a resolução de problemas linearmente não separáveis. Em outras palavras, pode-se dizer que uma rede é composta por várias unidades de processamento, cujo funcionamento é considerado simples. Essas unidades geralmente são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma RNA vem das interações entre as unidades de processamento da rede.

A maioria dos modelos de redes neurais possui alguma regra de treinamento, em que os pesos de suas conexões são ajustados de acordo com os padrões apresentados, ou seja, elas aprendem através de exemplos. A rede passa por um processo de treinamento a partir dos casos reais conhecidos, adquirindo, a partir daí, a sistemática necessária para executar adequadamente o processo desejado dos dados fornecidos. Sendo assim, a rede neural é capaz de extrair regras básicas a partir de dados reais, diferindo da computação programada, em que é necessário um conjunto de regras rígidas prefixadas e algoritmos.

### 2.3.8 Processos de aprendizado de uma rede neural artificial

De todas as propriedades interessantes das redes neurais artificiais, nenhuma captura tão bem a característica humana como a habilidade de aprender. Ao invés de especificar todos os detalhes de uma computação, tem-se a possibilidade de treinar uma rede para fazer esta computação. Isto significa que se podem tratar problemas cujas regras apropriadas são muito difíceis de se conhecer *a priori* (ABELÉM, 1994).

A habilidade de aprender a partir de seu ambiente, e com isso melhorar o seu desempenho, realiza-se através de um processo iterativo de ajustes aplicado a seus pesos, sendo chamado de **treinamento**. O **aprendizado** ocorre quando a rede atinge uma solução generalizada para uma classe de problemas.

Denomina-se **algoritmo de aprendizado** a um conjunto de regras bem definidas para a solução de um problema de aprendizado. Existem muitos tipos de algoritmos de aprendizado específicos para determinados modelos de redes neurais, diferindo entre si, principalmente pelo modo como os pesos são modificados.

A rede neural se baseia nos dados para extrair um modelo geral. Portanto, a fase de aprendizado deve ser rigorosa e verdadeira, a fim de se evitarem modelos espúrios. Todo o conhecimento de uma rede neural está armazenado nas sinapses, ou seja, nos pesos atribuídos às conexões entre os neurônios. De 50 a 90% do total de dados devem ser separados para o treinamento da rede neural, dados estes escolhidos aleatoriamente, a fim de que a rede “aprenda” as regras e não “decore” exemplos. O restante dos dados só é apresentado à rede na fase de testes, a fim de que ela possa “deduzir” corretamente o inter-relacionamento entre os dados.

O objetivo do treinamento de uma RNA é fazer com que a aplicação de um conjunto de entradas produza um conjunto de saídas desejado ou, no mínimo, um conjunto de saídas consistentes. Cada conjunto de entrada ou saída é chamado de **vetor**. O treinamento é realizado pela aplicação seqüencial dos vetores de entradas (e em alguns casos também os de saída), enquanto os pesos da rede são ajustados de acordo com um procedimento de treinamento predeterminado. Durante o treinamento, os pesos da rede gradualmente convergem para determinados valores, tal que a aplicação dos vetores de entrada produza as saídas necessárias.

Os procedimentos de treinamento (ou aprendizado) que levam as RNAs a aprender determinadas tarefas podem ser classificados em duas classes:

- ◆ Treinamento supervisionado
- ◆ Treinamento não supervisionado

O **treinamento supervisionado** necessita de um par de vetores composto do *vetor de entrada* e do *vetor-alvo* que se deseja como saída, ou seja, este tipo de aprendizado utiliza um agente externo que indica à rede um comportamento bom ou ruim de acordo com o padrão de entrada. Juntos, estes vetores são chamados de *par de treinamento* ou **vetor de treinamento**, sendo interessante ressaltar que geralmente a rede é treinada com vários vetores de treinamento.

O procedimento de treinamento funciona da seguinte forma: o *vetor de entrada* é aplicado. A saída da rede é calculada e comparada com o correspondente *vetor-alvo*. O erro encontrado é então realimentado através da rede e os pesos são atualizados de acordo com um algoritmo determinado a fim de minimizar este erro. Este processo de treinamento é repetido até que o erro para os vetores de treinamento tenha alcançado níveis bem baixos.

O **treinamento não supervisionado** (auto-organização), por sua vez, não requer vetor-alvo para as saídas e, obviamente, não faz comparações para determinar a resposta ideal. Ele não utiliza um agente externo indicando a resposta desejada para os padrões de entrada, mas, entretanto, utiliza exemplos de coisas semelhantes para que a rede responda de maneira semelhante.

O conjunto de treinamento modifica os pesos da rede de forma a produzir saídas que sejam consistentes, isto é, tanto a apresentação de um dos vetores de treinamento, como a apresentação de um vetor que é suficientemente similar, irão produzir o mesmo padrão nas saídas. O processo de treinamento extrai as propriedades estatísticas do conjunto de treinamento e agrupa os vetores similares em classes. A aplicação de um vetor de uma determinada classe à entrada da rede irá produzir um vetor de saída específico, mas não existe maneira de se determinar, antes do treinamento, qual o padrão que será produzido na saída para um vetor de entrada de uma determinada classe. Desta forma, a saída de algumas RNA's deve ser transformada em uma forma compreensível após o processo de treinamento, o que é um simples problema de identificação das relações entrada-saída estabelecida pela rede.

No que diz respeito aos **algoritmos de treinamento** usados, existe uma grande variedade, tanto para o treinamento supervisionado, como para o não supervisionado. Muitos deles baseiam-se no algoritmo proposto por Donald O. Hebb em 1961. Entretanto, os algoritmos mais eficientes foram desenvolvidos nos últimos trinta anos, baseados nos

trabalhos de Rosenblatt (1962) e Widrow e Hoff (1960). Entre estes, o mais difundido, com certeza, foi o algoritmo de retropropagação (Backpropagation).

### 2.3.9 Algoritmo de retropropagação do erro (Backpropagation)

A capacidade limitada de representação das RNA's com uma única camada foi superada na década de 60, a partir do perfeito entendimento do problema da separabilidade linear (MINSKY, 1988). Contudo, apesar de se ter descoberto como aumentar a capacidade de representação da rede com o acréscimo de novas camadas, faltava ainda uma forma de treinar a rede. Esta forma só foi descoberta em meados da década de 70, com a criação de vários algoritmos, entre eles o de **retropropagação** (backpropagation), que veio a se tornar um dos algoritmos mais populares para o treinamento, sendo em parte responsável pelo ressurgimento do interesse na área de Redes Neurais Artificiais.

Para cada modelo de rede neural existe um ou mais algoritmos de aprendizagem (treinamento). Neste capítulo apresenta-se apenas o modelo de retropropagação do erro para redes perceptron multicamadas, pois este modelo será a base do modelo predictor proposto neste trabalho.

Basicamente, o algoritmo de retropropagação consiste em fornecer variações para alterar os pesos da RNA, objetivando minimizar o erro obtido na saída através do aprendizado do vetor de treinamento (entrada-saída). Para isto, o algoritmo baseia-se no método do gradiente descendente, no qual dada uma medida de erro  $E(W)$ , procura-se melhorar o conjunto de pesos  $W_{ij}$ , deslizando para baixo na superfície definida no espaço  $W$  (WASSERMAN, 1989). Em outras palavras, o algoritmo do gradiente descendente sugere mudanças nos pesos  $W_{ij}$  por uma quantidade  $\Delta W_{ij}$  proporcional ao gradiente de  $E$ :

$$\Delta W_{ij} = -\eta \left( \frac{\partial E}{\partial W_{ij}} \right) \quad (2.5)$$

onde  $\eta$  é conhecido como *taxa de aprendizado*,  $i$  refere-se ao neurônio  $i$  da camada posterior e  $j$  ao neurônio  $j$  da camada anterior. Executando-se a derivada e fazendo-se a mudança para cada padrão de entrada individualmente, chega-se a:

$$\Delta W_{ij} = \eta * \delta_i * X_j \quad (2.6)$$

onde:

$$\delta_i = (T - O) * F'(NET) \quad (2.7)$$

A equação (2.7) é conhecida como *regra delta* (WASSERMAN, 1989).  $F'(NET)$  refere-se à *derivada da função de ativação*,  $T$  é o *vetor-alvo* e  $O$  é a *saída* fornecida pela RNA.

Neste ponto é importante ressaltar que, como existe um valor-alvo, o ajuste dos pesos da camada de saída é executado utilizando-se as equações (2.6) e (2.7). Todavia, a atualização dos pesos da camada escondida já é um pouco mais complexa, pois não existe valores-alvo para a comparação. Desta forma, o cálculo do delta ( $\delta$ ) deve ser feito substituindo-se a diferença entre o vetor-alvo e a saída fornecida pela RNA na equação (2.7), pelo somatório dos produtos entre os pesos que ligam o neurônio em questão com os neurônios da camada de saída e os deltas dos respectivos neurônios da camada de saída:

$$\delta_i = \left( \sum_j (\delta_j * W_{ij}) \right) * F'(NET) \quad (2.8)$$

Após calculado o  $\delta$  para o neurônio  $i$ , os pesos ligados a este neurônio podem ser ajustados através da equação (2.6) e assim, sucessivamente, para todos os neurônios da camada escondida. Caso haja mais de uma camada escondida, o procedimento deve ser o mesmo. Os deltas ( $\delta$ ) devem ser calculados e todos os pesos associados àquela camada devem ser ajustados, pois agora a camada escondida posterior é tida como referência para o cálculo do delta.

Como já citado, o objetivo do treinamento da rede é ajustar os pesos, de modo que a aplicação de um conjunto de entradas produza um conjunto de saídas desejadas. Antes de se iniciar o processo de treinamento, todos os pesos devem ser inicializados randomicamente com valores pequenos, garantindo dessa forma que a rede não ficará saturada com grandes valores de pesos e prevenindo-se contra certas patologias de treinamento.

O treinamento pode ser dividido em passos da seguinte forma (ABELÉM, 1994):

- 1- Selecionar o próximo par do conjunto de treinamento e aplicar o vetor de entrada à rede.
- 2- Calcular a saída da rede.
- 3- Calcular o erro entre a saída da rede e a saída-alvo.
- 4- Ajustar os pesos da rede de maneira a minimizar o erro.
- 5- Repetir o passo 1 até o passo 4 para cada vetor do conjunto de treinamento, até o erro se tornar aceitavelmente baixo para o conjunto inteiro.

Pode-se observar que os passos 1 e 2 constituem a etapa de propagação para frente, em que o sinal de entrada é transmitido através da rede da entrada até a saída. Os passos 3 e 4, por sua vez, constituem-se na etapa de retropropagação, na qual o erro calculado é propagado de volta através da rede para ajustar os pesos.

Apesar do grande sucesso das aplicações do algoritmo de retropropagação e da sua enorme popularidade, alguns problemas ainda o acompanham, entre os quais citamos:

- ◆ o longo período de treinamento, principalmente para problemas complexos, em que não há garantias de que, depois deste tempo, o treinamento tenha sido feito com sucesso;
- ◆ mínimos locais, já que a superfície do erro geralmente é cheia de vales e desníveis e o algoritmo emprega um tipo de gradiente descendente, existindo sempre a possibilidade de ficar preso em um mínimo local;
- ◆ paralisia da rede, pois durante o treinamento os pesos podem ser ajustados para valores muito grandes, os quais vão levar a derivada da função de ativação a zero (no caso de funções “*squashing*”), impedindo a rede de aprender o conjunto de treinamento.

Devido às deficiências do algoritmo de retropropagação, inúmeras variações têm sido propostas nos últimos anos, embora, até agora, nenhuma tenha solucionado de forma definitiva e confiável os problemas do “*Backpropagation*”.

## 2.4 Redes neurais artificiais na previsão de séries temporais

Dentre as inúmeras aplicações das RNA's, neste trabalho busca-se destacar a predição de resultados típicos relacionados às séries temporais.

### 2.4.1 Séries temporais

Pode-se definir uma *série temporal* como qualquer conjunto de observações ordenadas no tempo (MENDENHALL, 1993). Uma enorme quantidade de fenômenos de natureza física, biológica, econômica e outros estão enquadrados nesta categoria e têm suas características estudadas através da análise de séries temporais (MORETTIN, 1985).

As séries temporais podem ser representadas como a combinação de quatro componentes (MENDENHALL, 1993):

- ◆ **Componentes de tendência:** produzem mudanças graduais a longo prazo. São normalmente provocadas, por exemplo, pelo crescimento constante da população, pelo produto interno bruto, pelo efeito da competição ou por outros fatores que falham na tentativa de produzir mudanças repentinas, mas produzem variações graduais e regulares ao longo do tempo.
- ◆ **Componentes cíclicas:** provocam oscilações de subida e de queda nas séries, de forma suave e repetitiva ao longo da componente de tendência. Geralmente os efeitos cíclicos em uma série são causados por mudanças na demanda do produto, por ciclos de negócios e, em particular, pela inabilidade de se suprirem as necessidades do consumidor.
- ◆ **Componentes sazonais:** são aquelas oscilações de subida e de queda que sempre ocorrem em um determinado período do ano, do mês, da semana ou do dia. A diferença essencial entre as componentes sazonais e cíclicas é que as sazonais possuem movimentos facilmente previsíveis, ocorrendo em intervalos regulares de tempo, por exemplo, ano a ano, mês a mês, semana a semana ou mesmo dia a dia. Já os movimentos cíclicos tendem a ser irregulares, ocorrendo sobre um período de muitos anos.
- ◆ **Componentes de erro:** representam movimentos ascendentes e descendentes da série após a ocorrência de um efeito de tendência, um efeito cíclico ou de um efeito sazonal. Aparecem como flutuações de período curto, com deslocamento inexplicável e geralmente são causadas, entre outros motivos, por eventos políticos e oscilações climáticas imprevisíveis.

Na análise de séries temporais existem, basicamente, dois tipos de abordagem: a *análise no domínio do tempo* e/ou a *análise no domínio da frequência* (MORETTIN, 1985). Nos dois casos existem variados modelos que atendem às mais diversas aplicações no âmbito das ciências físicas, biológicas, das engenharias e da economia.

O interesse na análise de uma série temporal pode ser para:

- investigar o mecanismo gerador da série;
- descrever o comportamento da série;
- procurar periodicidades relevantes nos dados;
- fazer previsões de valores futuros da série.



### 2.4.2 Procedimentos estatísticos de previsão

Os procedimentos de previsão utilizados na prática variam muito, podendo ser simples e intuitivos com pouca ou nenhuma análise dos dados, ou mais quantitativos e complexos, com uma análise de dados considerável. É importante ressaltar que a previsão não constitui um fim em si, mas deve ser vista como parte integrante de um processo de tomada de decisão, visando determinados objetivos (ABELÉM, 1994).

Na área econômica, por exemplo, existem três procedimentos de previsão predominantes: o *econométrico*, o de *séries temporais* e o de *previsões qualitativas*. Os dois primeiros são projeções técnicas que envolvem ajustes dos modelos teóricos para os dados a serem analisados. A suposição utilizada nestes dois procedimentos é de que as observações de amostras de um ou mais processos, aparentemente randômicos, forneçam evidências confiáveis sobre o comportamento futuro do processo em estudo. A diferença entre eles é que o procedimento econométrico se baseia fortemente na teoria econômica para construir o modelo, a partir de uma ou mais relações causais entre variáveis. Já no procedimento de séries temporais, em princípio, apenas os dados da própria série são utilizados para construir o modelo (MENDENHALL, 1993).

O procedimento de previsão qualitativa foi projetado para prever o comportamento futuro de processos cujos dados históricos não existem ou não estão disponíveis, como por exemplo, a previsão de vendas de um novo produto lançado no mercado. O modelo qualitativo confia na análise de entradas subjetivas obtidas de várias fontes, como opiniões de especialistas, de executivos, carências de consumo, entre outros.

Devido à diversidade de procedimentos de previsão, os especialistas optaram por fazer agrupamentos em macrocategorias. Um dos tipos de classificação é aquele que se baseia no número de séries temporais envolvidas na modelagem, independente do tipo de estatística na qual elas se baseiam (SOUZA, 1989). Esta classificação divide os procedimentos em três grupos:

- **Modelos univariados:** inclui os métodos que se baseiam em uma única série histórica. Entre os mais conhecidos, merecem destaque: *decomposição por componentes não observáveis*, modelo mais utilizado até a década de 60; *modelos automáticos ou da caixa preta*, que surgiram na década de 60 e incluem os modelos de regressão, médias móveis, ajustamento sazonal e alisamento exponencial; *modelos Box-Jenkins*, propostos no início da década de 70, consistindo de uma classe geral de modelos lineares conhecidos como ARIMA; *modelos bayesianos*, introduzidos também na década de 70 e que se

caracterizam pela possibilidade de incorporação de informações subjetivas no processo de modelagem da série.

- **Modelos de função de transferência:** métodos nos quais a série de interesse é explicada não só pelo seu passado histórico, como também por outras séries temporais não correlacionadas entre si. Devem ser ressaltados os *modelos Box-Jenkins de função de transferência*, *modelos estruturais* e os *modelos de função de transferência bayesiana* (SOUZA, 1989).
- **Modelos multivariados:** grupo que modela simultaneamente duas ou mais séries temporais sem qualquer exigência com relação à direção da causalidade entre elas. Podem ser destacados (SOUZA, 1989) *modelo MARMA*, de Box e Tiao, *formulação em espaço de estado* e *formulação bayesiana multivariada*.

### 2.4.3 Previsão de séries temporais

Como citado, para se realizar uma análise, ou mesmo previsão de uma série temporal utilizando procedimentos estatísticos, é necessário, além de todas as considerações e transformações que possam vir a ser feitas para tornar a série adequada, um modelamento matemático *a priori*. Aliado a isto, tem-se o fato de que as técnicas estatísticas para predição e tomada de decisão não têm bons resultados para muitas aplicações que possuem tamanho de amostras restrito e com alguma não linearidade no conjunto de dados (KINGDON, 1993).

Em função deste desempenho, considerado apenas moderado, dos métodos estatísticos convencionais para previsão de séries temporais, principalmente na área financeira, os analistas passaram a procurar formas alternativas de previsão. As RNA's, dentre as muitas técnicas que surgiram, ganharam especial atenção pela sua habilidade de aprendizado e sua capacidade de generalização, associação e busca paralela. Estas qualidades as tornam capazes de identificar e assimilar as características mais marcantes das séries, tais como sazonalidade, periodicidade, tendências, entre outras, na maioria das vezes camufladas por ruídos, sem necessitar do laborioso passo da formulação teórica, imprescindível para os procedimentos estatísticos.

Além disso, as RNAs também têm-se destacado pelos seguintes aspectos: sua capacidade em lidar com dados não lineares; sua robustez, pois as RNA's são capazes de se autocorrigirem mesmo depois de previsões erradas e, sua facilidade de utilização, já que existem várias opções de software para simulação de RNA's dedicado a pessoas com pouca

experiência, tanto em redes neurais como em séries temporais. É interessante ressaltar, no entanto, que um bom conhecimento sobre RNA's e o domínio da área de aplicação, no caso séries temporais, são fatores que têm muita importância no desenvolvimento de aplicações bem sucedidas, principalmente pelo fato de que os projetos de redes neurais artificiais são fortemente baseados na experimentação e na experiência do projetista, o qual precisa ter bom senso e sensibilidade para fazer os ajustes adequados nos parâmetros de aprendizado a fim de se obterem os melhores resultados (ABELÉM, 1994).

Apesar de oferecer vantagens, as RNA's também possuem problemas. Um dos principais é a falta de procedimentos para definir com precisão o número de camadas ocultas ou o número de neurônios em cada uma destas camadas, ou seja, a configuração mais apropriada para a aplicação.

Outro problema é a incapacidade de explicação dos resultados obtidos. Estando as informações das RNAs intrínsecas aos pesos, não existem meios explícitos (ou imediatos) de se justificar os resultados fornecidos pelas redes neurais, o que em certas aplicações, como na concessão de empréstimos ou concessão de cartões de crédito, é fundamental.

Um terceiro problema, desta feita mais específico das RNA's que utilizam o algoritmo de retropropagação para o treinamento, é a possibilidade de não convergência das redes neurais devido a mínimos locais. Todavia, a introdução de fatores no algoritmo como o termo de *momentum* tem minimizado este problema para certas aplicações.

Como citado anteriormente, vários são os procedimentos de previsão, indo desde os mais simples e intuitivos até os mais quantitativos e complexos. As RNA's, quando usadas para previsão de séries temporais, normalmente utilizam a filosofia de procedimento na análise dessas séries, a qual baseia-se na observação dos dados históricos do sistema a fim de descobrir regularidades em fenômenos aparentemente randômicos, frequentemente encobertos por ruídos. Dentro deste contexto, as previsões ainda podem ser classificadas em dois tipos:

- **Previsões “*multi-step*”:** são aquelas que se caracterizam por possuir realimentação das saídas das RNAs para as entradas das mesmas. Neste tipo de previsão, o sistema neural usa um conjunto de valores correntes da série para prever os valores futuros desta série por um período fixo. Em seguida, esta previsão é realimentada na entrada do sistema para prever o próximo período. Estas previsões são muito usadas para identificar tendências e pontos de mudanças preponderantes nas séries. Devido ao erro que é inserido a cada nova previsão, as previsões “*multi-step*” são usadas para prever apenas alguns passos à frente, em média de 15 a 20 passos, quando o erro ainda é aceitável. Por

utilizarem realimentação das saídas, ficam restritas a previsões univariantes, já que no caso das multivariantes, nem todas as séries fornecidas nas entradas são previstas pelas RNA's para poderem ser realimentadas.

- **Previsões “single-step”:** neste tipo de previsão não existe realimentação. As RNAs utilizam apenas os valores anteriores da série para prever um passo à frente. Todavia, este passo tanto pode ser para previsões de curto prazo como para previsões de médio e longo prazos, bastando que se tenham dados suficientes para treinar a rede. A previsão “single-step” também serve para avaliar a adaptabilidade e a robustez do sistema, mostrando que, mesmo quando as RNA's fazem previsões erradas, elas são capazes de se autocorrigirem e fazer as próximas previsões corretamente. Como as previsões “single-step” não utilizam realimentação, elas estão habilitadas a fazer tanto previsões univariantes como multivariantes, tornando-se neste aspecto muito mais poderosas que as previsões “multi-step”.

## 2.5 Considerações finais

Neste capítulo, procuramos situar o conceito de risco e a sua importância na análise de investimentos. Além disso, introduzimos também os fundamentos das redes neurais artificiais, citando histórico, principais propriedades e aplicações, particularmente na previsão de séries temporais. O capítulo seguinte mostrará a RNA como parte de um sistema, estabelecendo as etapas a serem seguidas na sua implementação e definindo os fatores a serem utilizados para a análise de seu desempenho.

# CAPÍTULO 3

## 3. TÓPICOS SOBRE O PROJETO DE UM SISTEMA NEURAL

### 3.1 Considerações iniciais

A concepção básica dos modelos neurais está centrada na figura do neurônio. A potencialidade do modelo para solução de problemas complexos se baseia no paralelismo, na capacidade de processamento advinda da integração, na não-linearidade proporcionada pela atividade operacional de cada elemento da rede e pela capacidade da mesma de buscar a solução através de um método próprio de treinamento e auto-aprendizado. Como citado no capítulo 2, as redes neurais artificiais podem ser aplicadas a diferentes tipos de tarefas. Todas essas tarefas podem ser reagrupadas em dois grupos principais, segundo o tipo de saída fornecido pela rede neural e o comportamento que é buscado. Estes dois grupos são:

- **Redes para a classificação de padrões:** neste tipo, deve-se atribuir para cada exemplo apresentado à rede uma classe à qual o exemplo pertence. Portanto, a saída da rede é a classe associada ao exemplo e, por consequência, as classes são valores discretos e não contínuos. A classificação é um caso particular de aproximação de funções em que o valor de saída da rede é discretizado e pertence a um conjunto finito de classes. No caso do aprendizado supervisionado, o conjunto de classes é bem definido e conhecido antes de ser iniciado o processo de aprendizado.

- **Redes para a aproximação de funções:** neste tipo, as redes devem ter uma saída com valores contínuos e usualmente são empregadas para realizar aproximações de funções (interpolação), as quais são representadas por um conjunto de pontos-exemplo. Este tipo de rede é capaz de aprender uma função de transformação (ou de associação) de valores de entrada em valores de saída, usualmente estimando por interpolação as respostas para os casos que não aparecem na base de exemplos. É um tipo de problema de aprendizado neural de funções conhecido por ser uma aplicação de um *problema de regressão*. Em geral, as funções a serem aprendidas pelas redes possuem tanto as entradas como as saídas indicadas através de valores contínuos (variáveis não discretas). Este será o tipo de rede utilizado neste trabalho.

## 3.2 Modelagem da rede neural artificial

Diversos são os modelos de rede propostos na literatura, cada qual advindo de uma linha de pesquisa diferente e visando a um melhor desempenho na solução de um tipo específico de problema. Basicamente, os modelos neurais podem ser classificados segundo os aspectos a seguir:

- a) **estratégia de treinamento:** em *supervisionado* (quando a rede dispõe de um “instrutor” apontando erros e acertos) ou *não supervisionado* (caso contrário).
- b) **forma de treinamento:** em *incremental* (quando o conhecimento da rede se ajusta após a apresentação de cada padrão de entrada (estímulo)) ou *lote* (*batch* - onde o ajuste do conhecimento só é realizado após a “visão” de todos os estímulos), e
- c) **forma de operação:** em *unidirecional* (*feedforward* - os sinais internos se propagam apenas na direção entrada/saída) e *recorrente* (*recurrent* - quando há realimentação).

Infelizmente, não existe qualquer método ou procedimento sistemático e determinado para a configuração e especificação dos parâmetros envolvidos em uma rede neural. O número de parâmetros e detalhes de projeto é grande e o maior ou menor sucesso na especificação dos mesmos depende da experiência e do sentimento do projetista.

### 3.2.1 Etapas no desenvolvimento de um modelo de RNA

Para o desenvolvimento de um modelo ou projeto baseado em RNA são necessárias diversas etapas (Figura 3.1). São elas: definição do problema; coleta dos dados de treinamento e de teste; pré e pós-processamento dos dados; projeto da estrutura da rede; treinamento; teste e validação.

Em termos práticos, as redes neurais são criadas através de softwares exclusivamente voltados para este fim ou o projetista poderá desenvolver seu próprio código, específico para o problema. Podem-se ainda utilizar os pacotes comerciais de sistemas simuladores de redes neurais ou de domínio público. Caso o interessado possua bom conhecimento, irá indicar a estrutura da sua rede para uma determinada aplicação. Também poderá projetar e construir um hardware utilizando *chips* de redes neurais disponíveis no mercado.

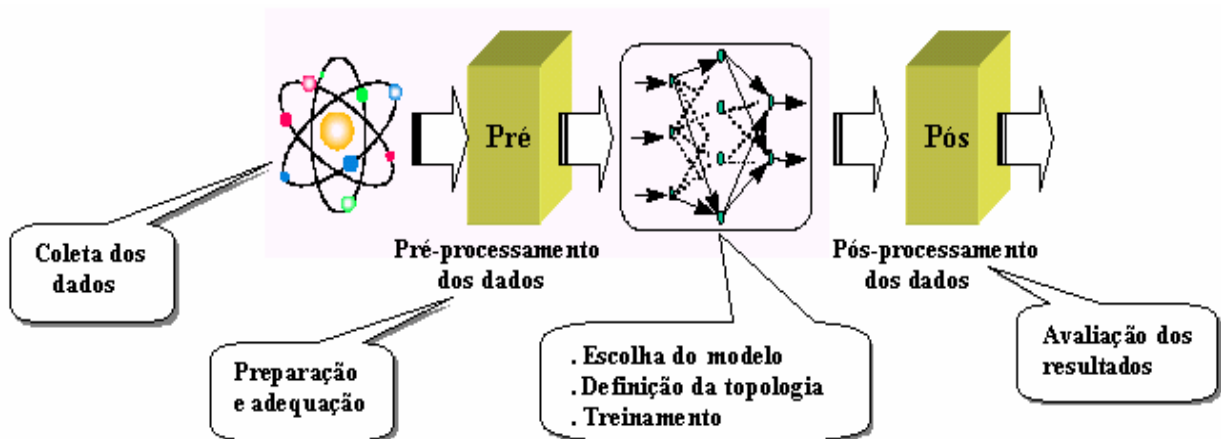


Figura 3.1 - Desenvolvimento de um projeto baseado em RNA.

Fonte: Thomé (2003)

Conforme mostrado na Figura 3.1, tudo começa pela escolha adequada da representação dos *dados* a serem usados na *caracterização do problema* e no *treinamento da rede*. A capacidade de resolução do problema, o desempenho do treinamento e a fidelidade dos resultados estão intimamente ligados à qualidade da representação adotada. Esta tarefa requer uma análise cuidadosa sobre o problema para minimizar ambigüidades e erros nos dados. Além disso, os dados coletados devem ser significativos e cobrir amplamente o domínio do problema; não devem cobrir apenas as operações normais ou rotineiras, mas também as exceções e as condições nos limites do domínio do problema.

Normalmente, os dados coletados são separados em duas categorias: **dados de treinamento**, que serão utilizados para o treinamento da rede, e **dados de teste**, que serão utilizados para verificar sua performance sob condições reais de utilização. Além dessa divisão, pode-se usar também uma subdivisão do conjunto de treinamento, criando um **conjunto de validação**, utilizado para verificar a eficiência da rede quanto à sua capacidade de generalização durante o treinamento.

Depois de determinados estes conjuntos, eles são geralmente colocados em ordem aleatória para prevenção de tendências associadas à ordem de apresentação dos dados. Além disso, pode ser necessário pré-processar estes dados, através de normalizações, escalonamentos e conversões de formato para torná-los mais apropriados à sua utilização na rede.

### 3.2.2 Configuração da rede neural artificial

A etapa seguinte no projeto da RNA diz respeito à configuração da rede. Ela pode ser conduzida segundo três atividades:

- I) Seleção do paradigma neural (tipo de rede) apropriado à aplicação.
- II) Determinação da topologia da rede a ser utilizada - número de camadas, número de neurônios em cada camada, etc.
- III) Determinação de parâmetros do algoritmo de treinamento e funções de ativação. Este passo tem um grande impacto na performance do sistema resultante.

Existem metodologias, “dicas” e “truques” na condução destas tarefas. Normalmente estas escolhas são feitas de forma empírica. A definição da configuração de redes neurais é ainda considerada uma arte, que requer grande experiência dos projetistas.

A rede é organizada geralmente em camadas e o padrão de conexão mais freqüentemente utilizado é o de conexão completa intercamadas (*fully connected*), apenas na direção entrada-saída (*feedforward*), e nenhuma conexão intracamada. Isto é, um neurônio situado em uma determinada camada tem sua saída conectada a todos os neurônios da camada seguinte (à sua direita) e a nenhum outro neurônio de qualquer outra camada, seja ela anterior, posterior ou a sua própria. A Figura 3.2 exemplifica esta topologia.

Cabe ressaltar que o aumento do número de camadas acarreta o aumento da complexidade e do tempo de processamento da rede. Aumentando-se o número de neurônios por camada, acarreta o aumento do grau de liberdade da função de transferência, e quanto maior a quantidade de variáveis livres, menor será a capacidade de generalização da rede. Logo, o objetivo de projeto deve ser o de resolver o problema com a menor topologia possível.

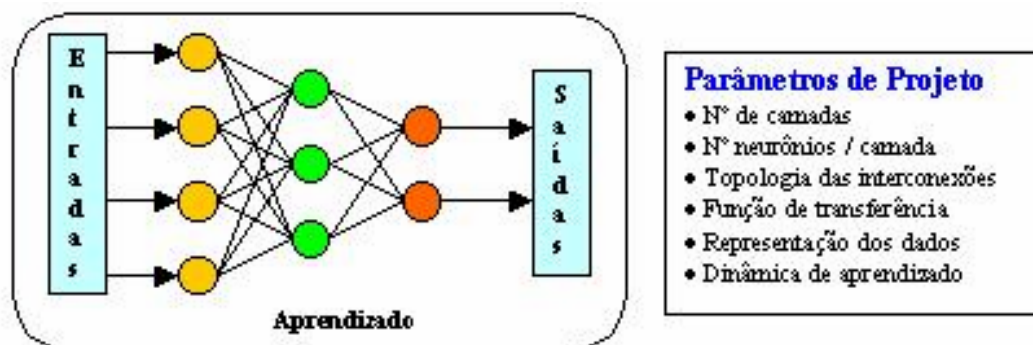


Figura 3.2 - Estrutura de uma RNA e seus parâmetros de projeto.

Adaptado de Thomé (2003).



### 3.2.3 Treinamento e aprendizado

Com a configuração da rede e a base de dados definidas, a próxima etapa a ser executada é o **treinamento da rede**, a fim de que ela possa “aprender”. O aprendizado é, em geral, um processo gradual e iterado, onde os pesos das conexões são modificados várias vezes, pouco a pouco, seguindo-se uma regra de aprendizado (algoritmo) que estabelece como estes pesos são alterados. O aprendizado é realizado utilizando-se um conjunto de dados de aprendizado disponível (base de exemplos). Cada iteração deste processo gradativo de adaptação dos pesos de uma rede neural, em que se faz uma apresentação completa do conjunto de dados, é chamada de **época de aprendizado** (*epoch*).

É importante considerar, nesta fase, alguns aspectos tais como a inicialização da rede, o modo de treinamento e o tempo de treinamento.

Uma boa escolha dos valores iniciais dos pesos da rede pode diminuir o tempo necessário para o treinamento. Normalmente, os valores iniciais dos pesos da rede são números aleatórios uniformemente distribuídos, em um intervalo definido. A escolha errada destes pesos pode levar a uma saturação prematura. Nguyen e Widrow encontraram uma função que pode ser utilizada para determinar valores iniciais melhores que valores puramente aleatórios.

Quanto ao modo de treinamento, na prática é mais utilizado o modo padrão devido ao menor armazenamento de dados, além de ser menos suscetível ao problema de mínimos locais, devido à pesquisa de natureza estocástica que realiza. Por outro lado, no modo *batch* se tem uma melhor estimativa do vetor gradiente, o que torna o treinamento mais estável. A eficiência relativa dos dois modos de treinamento depende do problema que está sendo tratado.

Uma rede pode se especializar demasiadamente em relação aos exemplos contidos na base de aprendizado. Este tipo de comportamento vai nos levar a um problema de aprendizado conhecido como **superaprendizado** (*over-training* ou *over-fitting*). Normalmente, o *over-fitting* pode ser detectado/evitado através do uso de um teste de generalização por validação cruzada (*cross validation*).

Quanto ao tempo de treinamento, vários fatores podem influenciar a sua duração, porém sempre será necessário utilizar algum critério de parada. O critério de parada do algoritmo backpropagation não é bem definido, e geralmente é utilizado um número máximo de ciclos. Mas, devem ser consideradas a taxa de erro médio por ciclo e a capacidade de generalização da rede. Pode ocorrer que em determinado instante do treinamento a generalização comece a degenerar, causando o *over-training*.

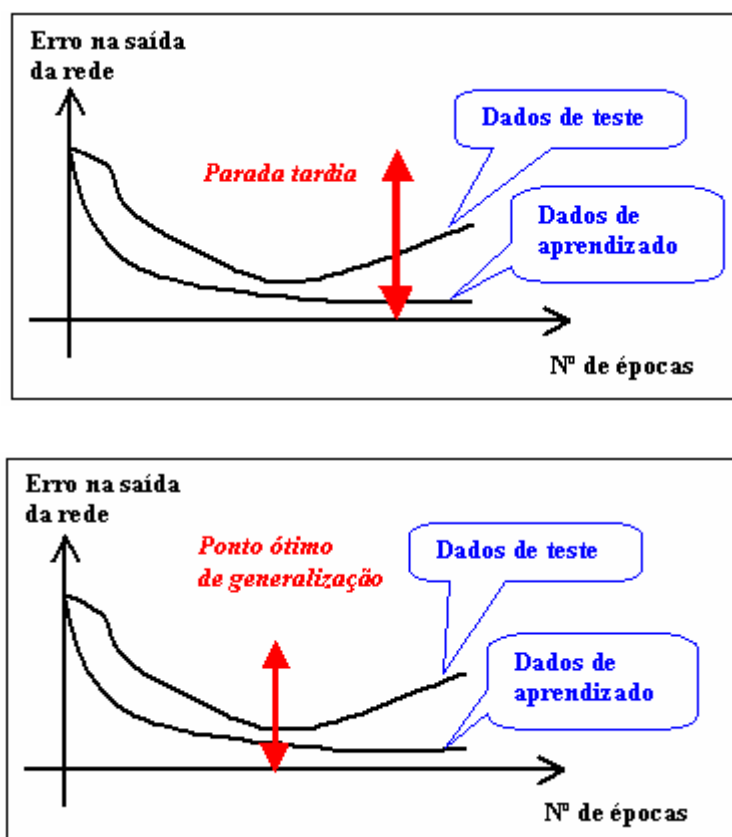


Figura 3.3 - Aprendizado: erro em relação ao conjunto de aprendizado e de teste.

Fonte: Osório & Bittencourt (2003)

O treinamento deve ser interrompido quando a rede apresentar uma boa capacidade de generalização e quando a taxa de erro for suficientemente pequena, ou seja, menor que um erro admissível. Assim, deve-se encontrar um ponto ótimo de parada com erro mínimo e capacidade de generalização máxima.

A Figura 3.3 apresenta um gráfico típico da evolução do **erro** durante o aprendizado de uma RNA, comparando a curva do erro (aprendizado supervisionado) referente à base de aprendizado com a curva do erro da base de teste de generalização. O conceito de **generalização** será apresentado no item a seguir, sendo que um modelo que tem uma boa generalização é aquele que responde corretamente aos exemplos contidos na base de aprendizado, mas também a outros exemplos diferentes daqueles da base de aprendizado, e que estão contidos em uma base de teste. A capacidade de generalizar é a principal capacidade buscada nas tarefas que envolvem aprendizado.

### 3.2.4 Fatores de desempenho

O desempenho das RNAs, em geral, é avaliado em função de três fatores (ABELÉM, 1994):

- Convergência
- Generalização
- Escalabilidade

A **convergência** é um fator importantíssimo para o bom desempenho das RNA's, pois diz respeito à capacidade das redes em aprender um determinado conjunto de dados definido no procedimento de treinamento. A convergência é que indica quando o procedimento de treinamento é capaz de aprender a classificação definida nos padrões apresentados, sob que condições das redes e sob quais exigências computacionais. Redes com topologias fixas que utilizam o algoritmo da retropropagação provam a convergência mostrando que em um tempo limite, quando o tempo de treinamento tende a infinito, o erro minimizado pelo método do gradiente descendente tende a zero (HERTZ *et ali*, 1991). Outros métodos provam a convergência mostrando que quando o tempo de treinamento tende a infinito, o método pode classificar o número máximo de possíveis mapeamentos com uma probabilidade arbitrariamente grande. A utilização da convergência como fator determinante para o desempenho da rede é feita por motivos óbvios, já que se uma determinada RNA não consegue aprender o conjunto de treinamento que lhe foi apresentado (não converge), dificilmente conseguirá oferecer bons resultados no momento dos testes.

A **generalização** diz respeito à habilidade das RNAs de reconhecer padrões fora do conjunto de treinamento e é a principal propriedade que deve ser exigida de um sistema neural. Ela determina a quantidade de dados necessários para treinar o sistema a fim de que este responda corretamente para padrões fora do conjunto de treinamento. Apesar das várias aplicações bem-sucedidas de RNA's descritas na literatura, as condições para que boas generalizações sejam encontradas ainda não são totalmente conhecidas. Abelém (1994) cita que estudos teóricos têm mostrado que a probabilidade de generalizações corretas depende do tamanho do espaço de hipóteses (número total de redes consideradas), do tamanho do espaço de soluções (conjunto de RNA's que fornece boas soluções) e do número de exemplos do treinamento. Se o espaço de hipóteses é muito grande e/ou o número de exemplos do treinamento é muito pequeno, então existirá um vasto número de RNA's, as quais são consistentes para o conjunto de treinamento. Porém, somente uma pequena porção deste

conjunto se encontra no espaço de soluções corretas, fazendo com que seja provável uma generalização pobre.

De uma forma geral, a diminuição dos parâmetros livres (número de conexões) da rede aumenta a probabilidade de uma boa generalização. Contudo, isto deve ser feito sem reduzir demais o tamanho da rede, já que esta redução pode implicar na incapacidade da rede em computar a função desejada. Outro ponto importante é a escolha de um conjunto de treinamento bastante representativo, apresentando as relações mais importantes que devem ser aprendidas pela rede. A maioria dos pesquisadores experientes concordam que uma boa representação dos dados de entrada é um ponto de fundamental importância para o sucesso de uma aplicação (WASSERMAN, 1993).

Finalmente, a **escalabilidade**, que diz respeito a ambos, convergência e generalização, é relativa ao procedimento de ajustar os parâmetros das RNA's para que elas consigam convergir e generalizar bem. Sabe-se que para uma rede neural artificial ter um bom desempenho, ela deve ser grande o suficiente para aprender o problema (convergir), mas também pequena o bastante para generalizar bem.

### 3.2.5 Controle dos fatores determinantes no desempenho das RNAs

O controle dos fatores determinantes de desempenho se dá pela escolha adequada dos parâmetros da rede, tais como: função de ativação, função erro, parâmetros de controle do gradiente descendente e do tempo de treinamento e arquitetura da rede, os quais, como já citado, variam de aplicação para aplicação e não possuem regras de escolha muito bem definidas.

#### 3.2.5.1 Padronização dos dados

Apesar de não controlar diretamente os fatores determinantes de desempenho das RNAs, a padronização pode, freqüentemente, melhorar a performance das redes através da remoção de características insignificantes do conjunto de dados de entrada. As informações relevantes das séries são, geralmente, independentes de valores de compensação (“*offset*”) e de desvios-padrão. Estas características podem obscurecer a tarefa das redes. A padronização, entretanto, admite várias formas e exige cuidados, já que é muito fácil remover, inadvertidamente, informações preciosas dos dados. Por outro lado, a padronização correta dos dados pode melhorar consideravelmente o desempenho de uma rede neural. Assim, temos a seguir os principais tipos de padronização para os dados:

➤ **Remoção da média**

É um dos tipos mais usados de padronização. Em muitos casos, a informação significativa de uma série está contida em pequenas variações que se encontram adicionadas a um grande valor de compensação, que pode levar os neurônios não lineares à saturação e, assim, inibir o treinamento. Este valor de compensação pode ser retirado, se é sabido que ele é irrelevante para o sistema e está inibindo o treinamento. Para isto, basta que se retire de todos os pontos da série a média da mesma. Assim, para cada componente dos vetores da série, tem-se:

$$\mathbf{vetor\_sm} = \mathbf{vetor} - \text{média} \quad (3.1)$$

➤ **Diferenciação**

Este tipo de padronização é utilizado principalmente quando se tem valores com pequenas variações absolutas, ficando a informação significativa na variação percentual dos mesmos. As pequenas variações em termos absolutos comumente camuflam a informação importante, fazendo com que a rede tenha um desempenho ruim. Desta forma, para cada componente dos vetores da série, tem-se:

$$\mathbf{vetor\_d} = \frac{\mathbf{vetor}_t - \mathbf{vetor}_{t-1}}{\Delta t} \quad (3.2)$$

➤ **Padronização da magnitude**

Grandes variações na magnitude dos componentes do vetor é outro fator que pode confundir a rede. Nestes casos, faz-se a padronização nas magnitude do vetor, dividindo cada componente do vetor pelo desvio-padrão calculado sobre o total de componentes. Com isto, para cada componente dos vetores da série, tem-se:

$$\mathbf{vetor\_sdp} = \mathbf{vetor} / \text{desvio padrão} \quad (3.3)$$

### ➤ Padronização não linear

A padronização *não linear* dos dados, como o logaritmo, é usada principalmente em casos em que a magnitude dos dados varia de muitas ordens de grandeza, mas a informação significativa está contida na variação percentual. Por exemplo, quando os maiores valores dos dados excedem a região da rede, as variações percentuais nos pequenos valores provavelmente serão perdidas. Neste caso, o logaritmo dos dados irá ajustar as faixas, de modo que os maiores valores serão comprimidos mais que os pequenos, alocando uma faixa constante para os desvios percentuais, independentes do valor dos dados. Seguem alguns exemplos de padronização não linear para cada componente dos vetores da série:

$$\begin{aligned} \mathbf{vetor\_nl} &= \log(\mathbf{vetor}) \\ \mathbf{vetor\_nl} &= \mathbf{vetor}^{0,25} \\ \mathbf{vetor\_nl} &= \exp(\mathbf{vetor}) \end{aligned} \tag{3.4}$$

Para todos estes tipos de padronização, quando a característica está sobre todos os componentes dos vetores, faz-se uma *padronização total*. Caso os componentes dos vetores representem diferentes tipos de dados, faz-se uma *padronização vertical* para cada tipo. Já no caso em que componentes adjacentes do vetor representam amostras consecutivas de um processo contínuo, utiliza-se a *padronização horizontal*. Por fim, em casos em que os dados possuem uma distribuição multimodal, com os vários modos tendo diferentes médias e desvios padrões, utiliza-se a *padronização local*.

### 3.2.5.2 Funções de ativação

A escolha da função de ativação dos neurônios é a primeira importante consideração no projeto de RNA's. Como visto no capítulo 2 (Seção 2.3.5), existem vários tipos de funções utilizadas, como: funções lineares, funções de limiar ou funções não lineares, que simulam mais precisamente as características dos neurônios biológicos. A escolha da função de ativação varia muito em função da aplicação e do algoritmo de treinamento utilizado. Entre as funções não lineares, as funções “*squashing*” são as mais usadas para todos os tipos de treinamento. Elas são mais complexas, diferenciáveis e comportam-se bem para a maioria das aplicações.

Em aplicações que possuem os valores dos padrões de treinamento muito altos, utiliza-se a função “*squashing*” somente nos neurônios das camadas escondidas, ficando os neurônios da camada de saída com funções lineares. Este artifício tem o objetivo de evitar a

paralisa da rede, pois quando os valores de entrada são altos, mesmo pequenos erros podem ser grandes em valores absolutos, forçando os neurônios a trabalharem em regiões onde a derivada das funções “squashing” é praticamente nula. Como o erro que é retropropagado para o treinamento é diretamente proporcional a esta derivada, o processo de treinamento fica paralisado e a rede não converge.

### 3.2.5.3 Funções erro

A função erro informa o grau de convergência das RNA's. São muitas vezes referidas como **métricas (ou função) de desempenho** (*performance function*). Através delas, é possível saber o quanto as RNA's estão aprendendo sobre os padrões apresentados durante o treinamento.

Alguns exemplos de funções-erro muito utilizadas como métricas de desempenho são:

- **Mean Square Error (MSE)**

O **MSE** é uma medida da exatidão da previsão em termos de valores absolutos. Possui a virtude de penalizar muito mais os erros quanto maiores eles forem. Porém, algumas vezes, pode apresentar resultados enganadores devido ao fato de trabalhar com médias, fornecendo resultados piores do que são realmente. O cálculo dessa métrica é feito da seguinte forma:

$$\mathbf{MSE} = \frac{1}{n} \sum_{t=1}^n (\mathbf{alvo}_t - \mathbf{prev}_t)^2 \quad (3.5)$$

onde:

**n** é o número de padrões apresentados.

**alvo<sub>t</sub>** é o valor-alvo no instante **t**.

**prev<sub>t</sub>** é o valor previsto no instante **t**.

Apesar de nem sempre ser a função mais indicada, devido aos problemas causados por trabalhar com médias, a **MSE** é a mais utilizada pela sua facilidade de cálculo e por ser uma métrica encontrada em praticamente todos os pacotes computacionais que trabalham com estatísticas e RNA's, facilitando assim a comparação dos resultados.

- **Mean Absolute Deviation (MAD)**

O **MAD** é dado pela seguinte expressão:

$$\mathbf{MAD} = \frac{1}{n} \sum_{t=1}^n |\text{alvo}_t - \text{prev}_t| \quad (3.6)$$

A diferença básica entre o MAD e o MSE é que o segundo eleva o erro ao quadrado, penalizando erros maiores com muito mais rigor.

- **Mean Absolute Percentage Error (MAPE)**

O **MAPE** é definido a partir do erro de previsão de acordo com a seguinte expressão:

$$\mathbf{MAPE} = \frac{\sum_{t=1}^n \left| \frac{\text{alvo}_t - \text{prev}_t}{\text{alvo}_t} \right|}{n} \times 100\% \quad (3.7)$$

O MAPE indica, como o próprio nome diz, o valor médio do erro percentual das previsões sobre todo o conjunto de teste.

- **Root Mean Square Error (RMSE)**

O **RMSE** é uma famosa medida absoluta de erro com características semelhantes ao do MSE, muito usada em diferentes campos da engenharia. É uma medida de erro que não permite que se determine com clareza o quanto o erro é realmente grande ou pequeno, pois isto depende da ordem de grandeza dos valores utilizados na série. Esta métrica também penaliza muito mais os erros maiores. Assim, uma técnica que apresente ótimos resultados na maioria das previsões, mas que tenha erros elevados em uma previsão específica, irá fornecer um alto RMSE. A sua expressão característica é dada por:

$$\mathbf{RMSE} = \sqrt{\frac{\sum_{t=1}^n \left| \frac{\text{alvo}_t - \text{prev}_t}{\text{alvo}_t} \right|^2}{n}} \quad (3.8)$$



- **Normalized Root Mean Square Error (NRMSE)**

A fim de resolver os problemas devido à falta de clareza do RMSE, criou-se o **NRMSE** (Raiz Quadrada Normalizada do Erro Médio Quadrático), tornando assim a medida de erro livre de qualquer unidade. Ele foi criado, dividindo-se o RMSE por um mesmo RMSE entre o valor-alvo e a média deste valor-alvo. A sua expressão característica é:

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n} \sum_{t=1}^n (\text{alvo}_t - \text{prev}_t)^2}}{\sqrt{\frac{1}{n} \sum_{t=1}^n (\text{alvo}_t - \overline{\text{alvo}_t})^2}} \quad (3.9)$$

onde:

**n** é o número de padrões.

**alvo<sub>t</sub>** é o valor-alvo no instante **t**.

**prev<sub>t</sub>** é o valor previsto no instante **t**.

$\overline{\text{alvo}_t}$  é a média da série-alvo.

- **Coefficiente U de Theil**

É uma métrica que mede o quanto os resultados estão melhores que uma previsão ingênua ou trivial que diz que “a melhor estimativa de um valor para amanhã é o valor de hoje”. Através deste coeficiente pode-se analisar a qualidade de uma previsão da seguinte maneira:

- quando  $U \geq 1$ , o erro do modelo é maior que o erro ingênuo.
- quando  $U < 1$ , o erro do modelo é menor que o erro ingênuo.

Assim, o coeficiente **U** de Theil menor que 1 já indica uma previsão melhor que a previsão trivial. Portanto, quanto mais próximo de zero for este coeficiente, melhor é a previsão. A sua expressão característica é :

$$\mathbf{U} = \frac{\sqrt{\sum_{t=1}^n (\text{alvo}_t - \text{prev}_t)^2}}{\sqrt{\sum_{t=1}^n (\text{alvo}_t - \text{alvo}_{t-1})^2}} \quad (3.10)$$

onde:

$n$  é o número de padrões.

$alvo_t$  é o valor-alvo no instante  $t$ .

$prev_t$  é o valor previsto no instante  $t$ .

$alvo_{t-1}$  é o valor previsto no instante anterior a  $t$ .

#### 3.2.5.4 Parâmetros de controle do gradiente descendente e do tempo de treinamento

O terceiro importante mecanismo para controlar o desempenho do aprendizado das RNA's é a escolha dos parâmetros de controle do gradiente descendente e do tempo de treinamento. Pesquisadores têm experimentado termos extras na fórmula do gradiente descendente como o *termo de momentum*, *segunda derivada*, entre outros (AZOFF, 1993). O *termo de momentum* é utilizado para diminuir o tempo de treinamento e o risco de mínimos locais, sem que o risco de oscilações das RNA's aumente.

A utilização da segunda derivada é um artifício que tem sido usado por vários métodos de treinamento (WASSERMAN, 1993) e que chega a reduzir por um fator de 100, o tempo de treinamento. Nesta técnica, ao invés de utilizar a inclinação da superfície do erro (primeira derivada) como faz o algoritmo de retropropagação, utiliza-se a taxa de variação da inclinação (segunda derivada) desta superfície.

Outro termo de fundamental importância no controle do gradiente descendente e no tempo de treinamento é a **taxa de aprendizado**. Esta essencialmente influencia a magnitude das mudanças dos pesos, desempenhando papel fundamental na evolução do aprendizado. Encontrar uma taxa de aprendizado apropriada para cada aplicação é uma tarefa difícil que depende basicamente de experimentações. Uma taxa de aprendizado pequena implica pequenas variações, mesmo quando grandes mudanças são necessárias, tornando o treinamento lento e aumentando as chances de paradas em mínimos locais. Por outro lado, altas taxas de aprendizado provocam grandes mudanças nos pesos, diminuindo o tempo de treinamento e evitando mínimos locais, mas podendo levar a RNA à saturação ou mesmo à oscilação, comprometendo todo o processo de aprendizado.

#### 3.2.5.5 Arquitetura da RNA

A arquitetura da rede é muito importante para as capacidades de convergência e generalização das RNA's. A escolha de uma arquitetura adequada, nem tão grande, nem tão pequena, é a principal responsável pelo sucesso de uma aplicação. Para se obter uma boa habilidade de generalização, deve-se projetar a rede colocando dentro dela o máximo

conhecimento possível sobre o problema (e.g. a topologia do espaço de entradas) e limitar o número de conexões adequadamente.

Existem várias técnicas que tentam encontrar a arquitetura ideal através da diminuição de neurônios redundantes e do compartilhamento de pesos, porém nenhuma conseguiu resultados convincentes para todas as aplicações. Existe também a possibilidade de fazer uma busca no espaço de todas as arquiteturas possíveis. Para isso, deve-se treinar cada uma das arquiteturas separadamente e então avaliá-las com uma função-erro apropriada que incorpore o desempenho e o número de unidades. A busca também pode ser feita através de um **algoritmo genético**, de forma que as boas arquiteturas encontradas em uma triagem sejam mantidas para serem combinadas com boas arquiteturas de outras triagens. Contudo, estes tipos de busca se mostram pouco práticos para aplicações que utilizam grandes RNA's, em que cada treinamento exige um alto esforço computacional. A verdade é que a arquitetura ideal das RNA's ainda continua sendo determinada principalmente pela experimentação manual e pela experiência e sensibilidade do projetista.

### 3.3 Teste

Após a fase de treinamento e aprendizagem, ocorre a etapa de testes, na qual o conjunto de dados de teste é utilizado para determinar o desempenho da rede com dados que não foram previamente utilizados. Este desempenho da rede, medido nesta fase, é uma boa indicação de seu comportamento real.

Devem ser considerados ainda outros testes como análise do comportamento da rede utilizando entradas especiais e análise dos pesos atuais da rede, pois se existirem valores muito pequenos, as conexões associadas podem ser consideradas insignificantes e, assim, serem eliminadas (*prunning*). De modo inverso, valores substancialmente maiores que os outros poderiam indicar que houve *over-training* da rede.

### 3.4 Avaliação dos resultados

A consistência dos resultados fornecidos pela rede neural pode ser avaliada a partir de uma comparação com os resultados obtidos através de métodos estatísticos já amplamente utilizados para a mesma aplicação. No caso de previsão de séries temporais, a *análise de regressão* constitui-se numa boa ferramenta para a comparação dos resultados.

### 3.5 Integração

Finalmente, com a rede treinada e avaliada, ela pode ser integrada em um sistema do ambiente operacional da aplicação. Para maior eficiência da solução, este sistema deverá conter facilidades de utilização como interface conveniente e facilidades de aquisição de dados através de planilhas eletrônicas, interfaces com unidades de processamento de sinais ou arquivos padronizados. Uma boa documentação do sistema e o treinamento de usuários são necessários para o sucesso do mesmo.

Além disso, o sistema deve periodicamente monitorar seu desempenho e fazer a manutenção da rede quando for necessário ou indicar aos projetistas a necessidade de retreinamento. Outras melhorias poderão ainda ser sugeridas à medida que os usuários forem se tornando mais familiares com o sistema, tornando-se muito úteis em novas versões ou em novos produtos.

### 3.6 Considerações finais

Neste capítulo, foram apresentados os tópicos essenciais do projeto de um sistema baseado em rede neural artificial, como configuração, treinamento e aprendizado, além dos fatores utilizados na determinação do desempenho da rede. O capítulo seguinte mostrará a aplicação de uma RNA num caso prático: a previsão do valor da cotação de um tipo de *commodity* do agronegócio.

# CAPÍTULO 4

## 4. APLICAÇÃO E RESULTADOS OBTIDOS

### 4.1 Considerações iniciais

A grande capacidade de modelar funções extremamente complexas e a relativa facilidade de utilização, apoiada em programas computacionais cada vez mais versáteis e poderosos, possibilitaram uma verdadeira explosão de interesse pelas RNA's nos últimos anos. Inumeráveis também são as áreas de aplicação em que podemos utilizar as suas funções mais aplicadas: predição e controle. Deve-se atentar para o fato de que, em praticamente qualquer situação onde exista uma relação entre variáveis de predição (variáveis de entrada, variáveis independentes) e variáveis preditas (variáveis de saída, variáveis dependentes), as RNA's são aplicáveis, particularmente se essas relações forem tão complexas que não possam ser eficientemente articuladas em termos de “correlação” ou “diferença entre grupos” (“*clusters*”). Como estabelecido na proposta de trabalho citada no item 1.4 (Objetivo), a RNA será utilizada como uma ferramenta alternativa na previsão de indicadores financeiros, os quais são de fundamental importância para a elaboração de fluxos de caixa na análise de investimentos, dentro de um contexto mais amplo de análise econômica de decisões.

### 4.2 Exemplo de aplicação

A proposta deste capítulo é utilizar a rede neural como instrumento para efetuar a previsão da cotação média mensal no mercado interno de um tipo de *commodity*: a arroba de boi gordo. Iremos definir como referência para esta previsão os valores da cotação histórica da praça da região noroeste do estado de São Paulo, tradicional pólo de negociação desse produto em nosso país.

Uma vez definido o problema a ser solucionado, será necessário coletar os dados a serem utilizados no treinamento e teste da rede. Estes dados irão constituir os conjuntos de treinamento e de teste/validação, devendo incluir certa quantidade de valores característicos das variáveis de entrada e de saída da rede.

A primeira decisão deve ser: que variáveis utilizar e quantos conjuntos de observações ou “*cases*” (relação de valores entrada-saída) serão necessários para o problema?

A escolha das variáveis, ao menos inicialmente, é levada pela intuição. A experiência do pesquisador no assunto irá definir esta escolha, buscando selecionar um conjunto significativo e ao mesmo tempo com o menor número possível de variáveis, de forma que o desempenho da rede não seja comprometido pela dimensionalidade.

Quanto à ordem de grandeza dos valores das variáveis, sabe-se que a rede neural processa bem os dados numéricos em uma faixa de valores relativamente limitada, daí a necessidade de se fazer um pré-processamento desses dados, objetivando esta padronização (“*scaling*”). Além disso, existe a dificuldade natural de se trabalhar com dados não numéricos (ou nominais), os quais deverão ser convertidos para numéricos através de uma regra preestabelecida, já que as RNA’s recebem entradas numéricas e produzem saída numérica. Se isso não for possível, a variável deverá ser descartada.

As redes neurais são tolerantes a ruídos (sinais espúrios), mas existem limites para esta tolerância. Se, no conjunto de dados existirem ocasionalmente valores fora da faixa de normalidade (“*outliers*”), estes, se possível, deverão ser removidos, de forma a não afetar, pela tendenciosidade (“*bias*”), o treinamento.

### 4.3 Configuração da rede

De forma a modelar o problema proposto, será utilizada uma rede com três camadas: entrada, intermediária e saída, conforme ilustrado na figura 4.1.

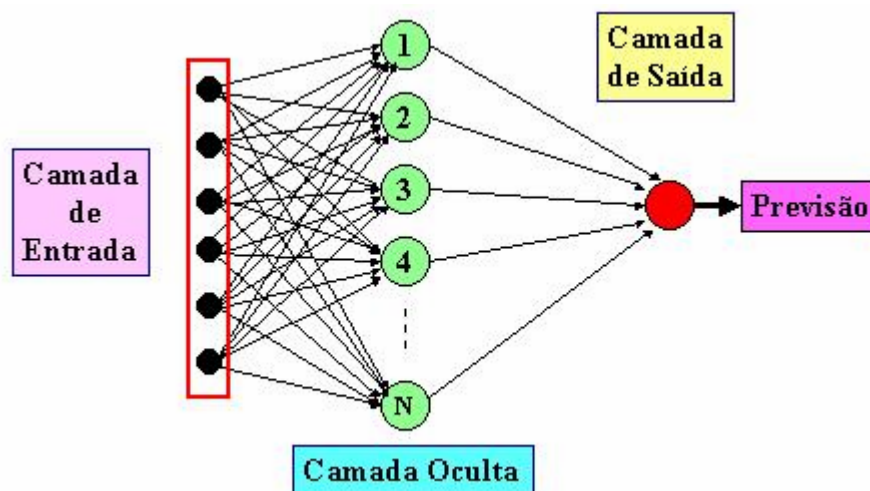


Figura 4.1 Modelo de rede neural MLP (Multilayer Perceptron) com três camadas.

### 4.3.1 Camada de entrada

Esta camada é constituída pelo conjunto de variáveis, econômicas ou não, que o pesquisador julga serem representativas para o caso estudado. Diante de um amplo conjunto de variáveis que podem influir no preço da arroba bovina, optou-se por utilizar as seguintes:

- Mês da previsão;
- Cotação do mês anterior ( @ - 1 ) em US\$;
- Cotação de dois meses atrás ( @ - 2 ) em US\$;
- Taxa de inflação do mês anterior ( **IGPM - 1** );
- Taxa de inflação de dois meses atrás ( **IGPM - 2** );
- Taxa de juros do mês anterior ( **Selic - 1** ).

Evidentemente que muitas outras variáveis não citadas acima podem influir de forma mais ou menos intensa no preço final da *commodity* em questão, porém, procurou-se apoiar esta escolha no fato de que os fatores econômicos e ambientais com suas características de sazonalidade sejam os mais relevantes e próprios para este caso.

#### 4.3.1.1 Mês da previsão

Esta variável incorpora a característica da sazonalidade tão expressiva na definição do custo da arroba bovina, tendo em vista a predominância da criação extensiva no Brasil e, naturalmente, tão dependente das condições das pastagens naturais.

#### 4.3.1.2 Cotações de meses anteriores

Esta informação contém o melhor conteúdo para fornecer à rede um indicativo de tendência para a cotação do mês seguinte, por já embutir em seu valor todos os elementos que influem no valor final da cotação, só que defasada no tempo em relação ao mês da previsão. O próprio Coeficiente U de Theil, uma das métricas de avaliação do erro, parte do pressuposto que “a melhor estimativa para o preço de amanhã é o preço de hoje”.

#### 4.3.1.3 Taxa de inflação (IGP-M)

A taxa de inflação representa, em essência, o acréscimo percentual médio verificado nos preços de diversos bens e produtos produzidos pela economia e o controle desta taxa constitui um dos mais importantes objetivos da política econômica do governo, pela sua

influência no nível de produção e consumo do mercado. Existem diversos indicadores de variação de preços adotados na economia nacional, cujos usos são praticados em diferentes segmentos de atividade. Utilizaremos o IGP-M (Índice Geral de Preços de Mercado), por ser de uso amplo e consagrado no mercado financeiro.

#### 4.3.1.4 Taxa de juros (Selic)

As taxas de juros e os preços dos bens físicos do mercado, expressos em porcentagens e unidades monetárias, respectivamente, têm por finalidade comum homogeneizar os valores de um conjunto de bens (ASSAF NETO, 2001). Por meio dos juros, é possível aos agentes econômicos tomar as mais diferentes decisões (alternativas de pagamento, políticas de estocagem, etc.). A taxa de juros escolhida para aplicação neste trabalho é a **Selic** (Sistema Especial de Liquidação e Custódia), que é a utilizada para o financiamento dos títulos públicos, sendo definida mensalmente pelo **Copom** (Comitê de Política Econômica), órgão que atua no âmbito do Banco Central, e geralmente divulgada com acompanhamento de um indicativo de viés, podendo este ser de baixa, de alta ou neutro, caracterizando uma tendência de comportamento da taxa. É evidente a influência da taxa de juros no crescimento ou diminuição da produção dos insumos agregados à produção da carne bovina e, conseqüentemente, estendendo sua influência ao preço final da *commodity*.

A definição do número de neurônios da camada de entrada é diretamente dependente do número de variáveis, e na indicação do mês da previsão optou-se por utilizar o código binário, sendo necessário para isto quatro dígitos, que corresponderão a quatro neurônios, além de um neurônio para cada uma das demais variáveis, totalizando nove neurônios nesta camada.

#### 4.3.2 Camada intermediária (ou oculta)

Existem algumas regras clássicas utilizadas na definição da quantidade de elementos da camada intermediária. Nesta determinação é necessário evitar-se o ***under-learning*** (a rede não alcança um nível de desempenho aceitável) ou o ***over-learning*** (o erro começa a crescer). Neste último caso, deve-se buscar diminuir o número de neurônios ou mesmo o número de camadas ocultas.

Uma rede do tipo MLP com duas camadas ocultas é teoricamente suficiente para modelar qualquer tipo de problema. Isto não implica que uma rede com mais camadas não possa modelar mais facilmente um determinado problema. Entretanto, na prática, redes com uma única camada oculta são as mais usuais, sendo que ocasionalmente utilizam-se duas camadas; redes com três camadas ocultas são praticamente desconhecidas.



A quantidade de neurônios a serem utilizados na camada oculta é uma questão ainda nebulosa. Como ponto de partida, uma boa referência prática é utilizar-se uma única camada com uma quantidade de elementos igual à metade da soma do número de unidades de entrada com a(s) de saída. Um parâmetro também muito citado em diversos trabalhos é o baseado no **pressuposto de Hecht-Nielsen/Kolmogorov**:

*“Uma rede neural com três camadas (entrada, oculta e saída) pode modelar qualquer função matemática contínua, desde que a camada oculta contenha  $2 \times I + 1$  neurônios, onde  $I$  é o número de variáveis de entrada.”*

Assim, de acordo com este pressuposto, teremos  $(2 \times 9) + 1 = 19$  neurônios na camada intermediária.

Baseado, então, na referência prática e no pressuposto acima, teremos uma quantidade de neurônios entre 5 e 19 a serem considerados para esta camada, buscando-se através do treinamento e simulação, qual será a quantidade que trará o melhor resultado.

Outra questão a ser considerada é a função de transferência da camada intermediária, a qual deve ser não linear, de forma a interpolar uma solução para qualquer tipo de problema. A função escolhida foi a **sigmoidal** (logística), por sua natureza não linear, assim como também pelo fato de sua saída ser formada apenas por valores positivos, tendo em vista que os valores de cotação da *commodity* também o serão, sem exceção.

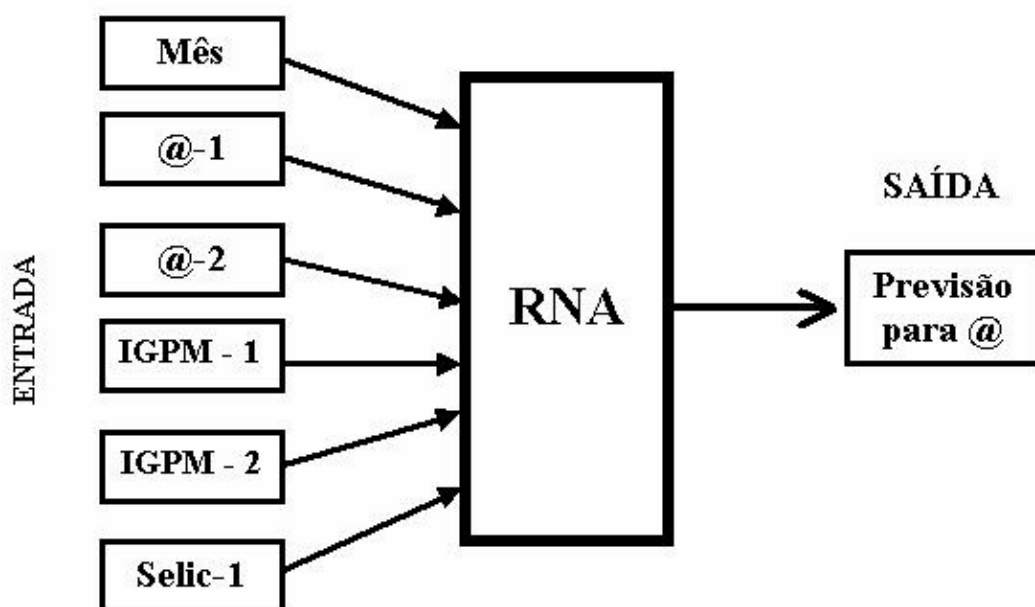


Figura 4.2 - Diagrama em blocos da rede proposta.



Regression Neural Network) e **Linear**. Os resultados mais consistentes neste trabalho foram obtidos com a MLP e a RBF.

Uma questão importante na regressão é a padronização dos dados (*scaling*), como citado no item 4.2, objetivando restringir a faixa de variação dos dados para facilitar o trabalho da rede e melhorar o seu desempenho. Nesta padronização utilizou-se o **escore z**, que é definido pela expressão 4.1 e que representa o nº de desvios-padrão que um valor dista da média.

$$z = \frac{x - \mu}{\sigma} \quad (4.1)$$

onde:

$x$  : valor da variável;

$\mu$ : média aritmética dos valores da variável  $x$ ;

$\sigma$ : desvio padrão dos valores da variável  $x$ .

Foram apresentados à rede, para treinamento, dados sobre as variáveis de entrada e saída correspondentes ao período de abril de 1998 até fevereiro de 2004. Como teste foram realizadas as previsões para os meses de março até maio de 2004, ou seja, três meses à frente. A seguir, a previsão obtida foi então comparada com os valores reais e calculados os erros. Os resultados estão indicados no item .

Para a padronização dos valores das variáveis foi utilizado o software MINITAB<sup>®</sup> versão 13 e para o trabalho com as redes neurais o software MATLAB<sup>®</sup> versão 6.0 R12, num computador com processador Pentium<sup>®</sup> III 1100 MHz.

#### 4.4.2 Implementação

São inúmeros os programas computacionais disponíveis no mercado para uso com redes neurais artificiais. Alguns são específicos para essa finalidade e outros, mais abrangentes, disponibilizam esta aplicação como uma de suas ferramentas. Dentre estes últimos, destacamos o MATLAB<sup>®</sup> e o STATISTICA<sup>®</sup>. Neste trabalho utilizamos a *toolbox* Neural Networks da versão 6.0 do MATLAB<sup>®</sup>, que disponibiliza uma interface gráfica para seleção dos diferentes parâmetros de treinamento e teste sem a necessidade de utilização de comandos específicos e suas sintaxes para desenvolvimento de código fonte em linguagem de programação. Esta característica facilita e incentiva a sua utilização, por incorporar características dos softwares mais modernos que permitem a seleção de opções a partir de

caixas de diálogo e botões. Na figura 4.3 temos a tela inicial da interface gráfica do MATLAB®, onde VE4 e S3 são as matrizes que contêm os valores das variáveis de entrada utilizadas no treinamento da rede e na simulação para a previsão. A matriz T2 corresponde aos valores alvo do treinamento (valores históricos da cotação).

A figura 4.4 apresenta a janela de criação da rede, onde estão indicados o nome (identificação) da rede, o tipo de rede, as funções de treinamento e de aprendizagem, a métrica de cálculo do erro, a quantidade de camadas, o nº de neurônios por camada e a função de transferência utilizada em cada camada.

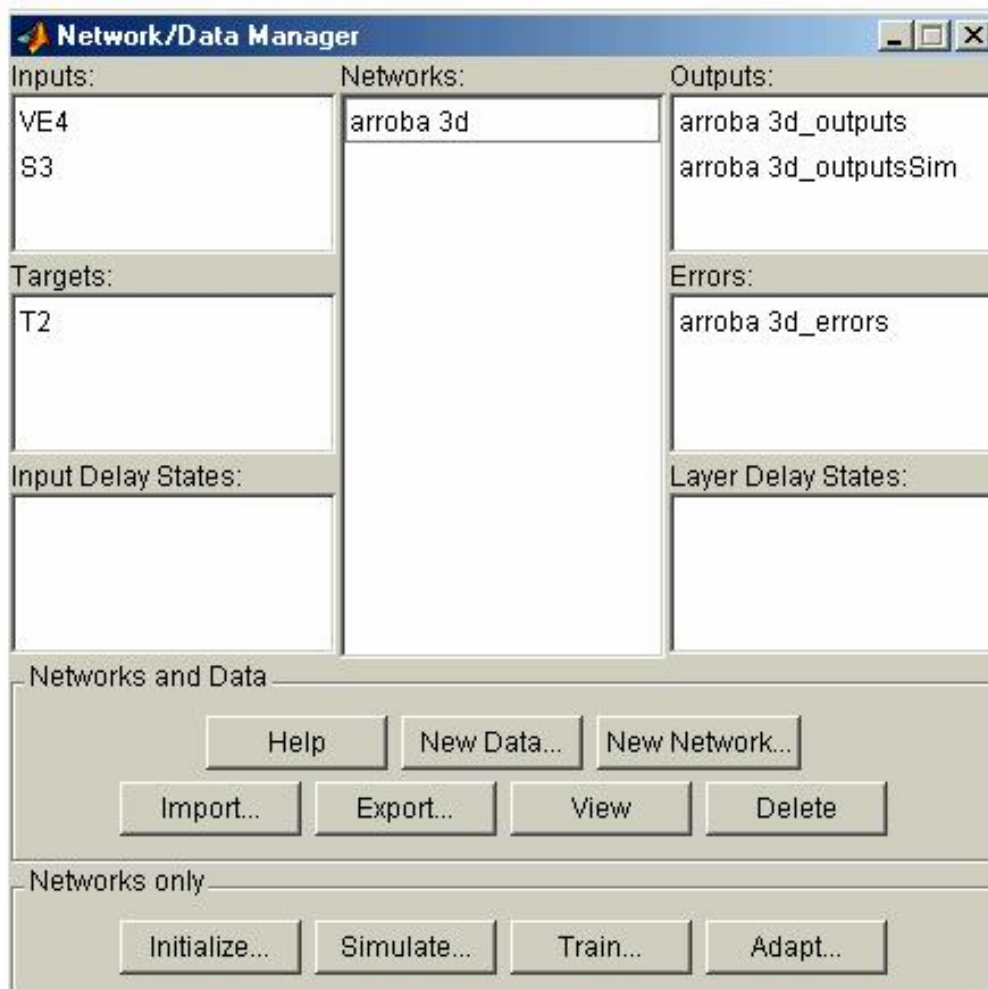


Figura 4.3 - Tela inicial da interface gráfica da *toolbox* de Redes Neurais do MATLAB®.

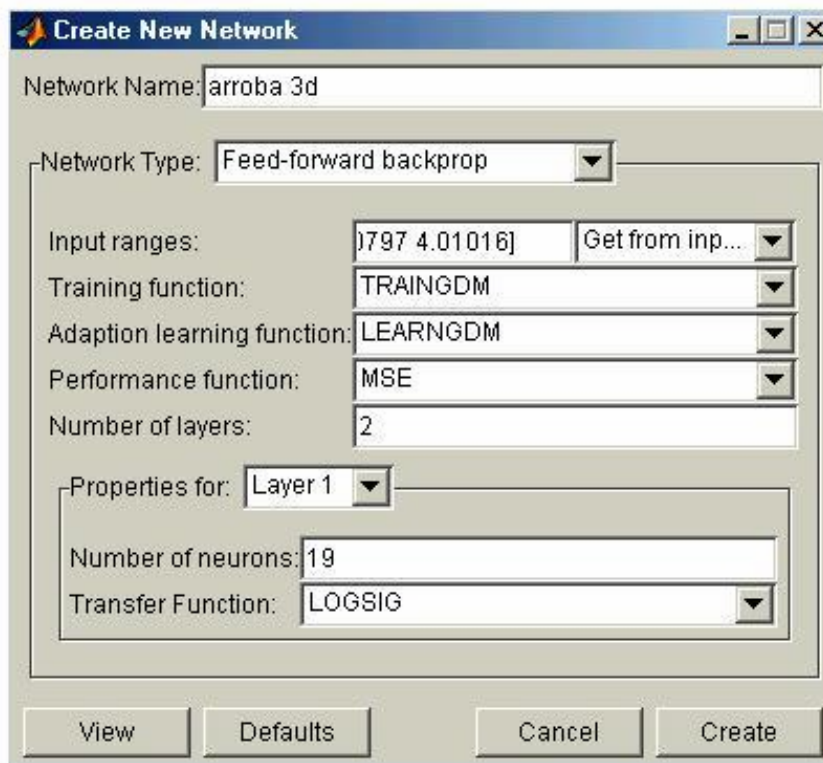


Figura 4.4 - Janela de criação da rede neural.

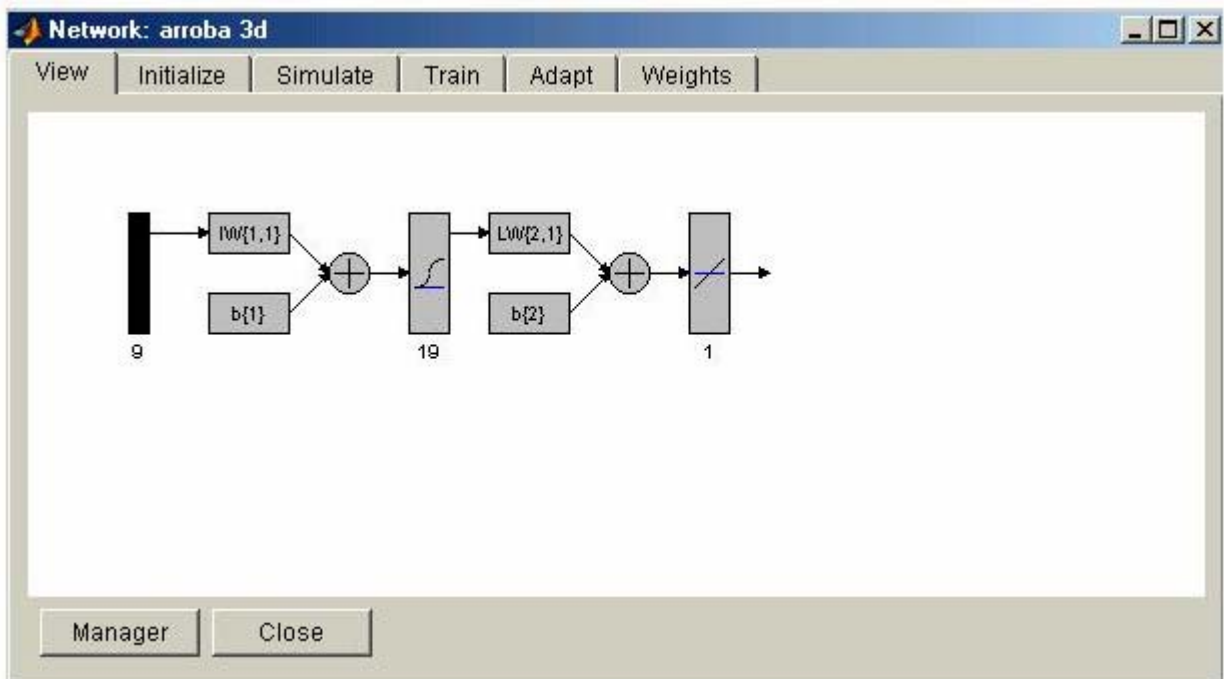


Figura 4.5 - Estrutura da rede neural definida na janela de criação da rede.

A figura 4.5 mostra a estrutura da rede criada, onde se destacam o número de elementos de cada camada (9 neurônios na camada de entrada, 19 na camada intermediária e 1 na camada de saída) e a função de transferência utilizada na camada intermediária (logsig) e na de saída (purelin).

#### 4.4.3 Treinamento

Para a configuração de rede escolhida foram definidos os parâmetros de treinamento indicados na figura 4.6, onde se destacam:

- n° de épocas: 3000
- limite definido para o erro em relação aos valores-alvo: 0,001
- taxa de aprendizagem: 0,01.

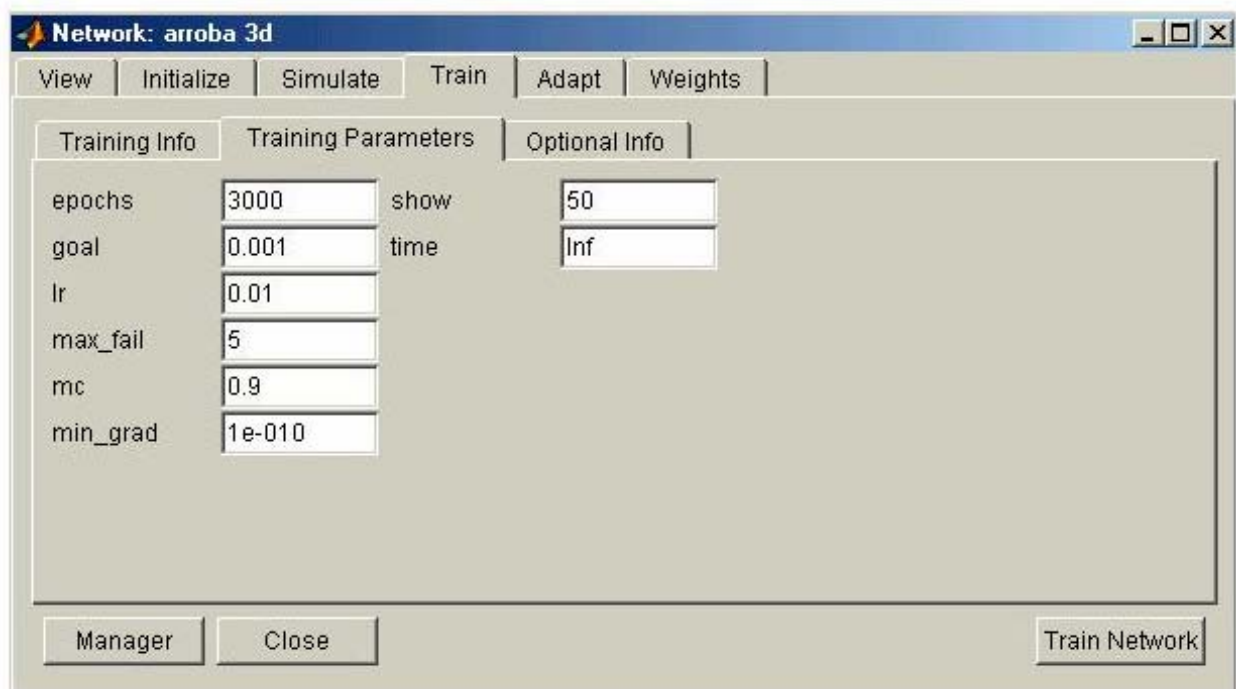


Figura 4.6 - Janela de definição dos parâmetros de treinamento da rede.

A escolha dos parâmetros acima, combinados ainda com aqueles mostrados na figura 4.4, está intimamente ligada às características dos dados que se quer prever e a sua influência na capacidade de generalização da rede neural proposta. É sabido que muitos neurônios na camada intermediária, assim como um erro total muito pequeno, têm influência decisiva na capacidade de generalização da rede. Esta atinge um ponto máximo durante o treinamento, a partir do qual começa a decorar os padrões de entrada degradando a sua performance. Um

número muito grande de neurônios na camada intermediária diminui o tempo de treinamento, mas também diminui a capacidade de generalização da rede. O erro total de treinamento determina o grau de precisão com que a função de aproximação se ajusta ao conjunto de treinamento. Um erro muito pequeno tende a fazer com que a rede decore os padrões de treinamento e não generalize de forma adequada.

Definida a rede com todos os seus elementos, passa-se ao treinamento, cujos parâmetros foram indicados na figura 4.6. Durante este treinamento, o gráfico da figura 4.7 vai sendo gerado, encerrando com o número de épocas definido anteriormente na caixa de diálogo da figura 4.6.

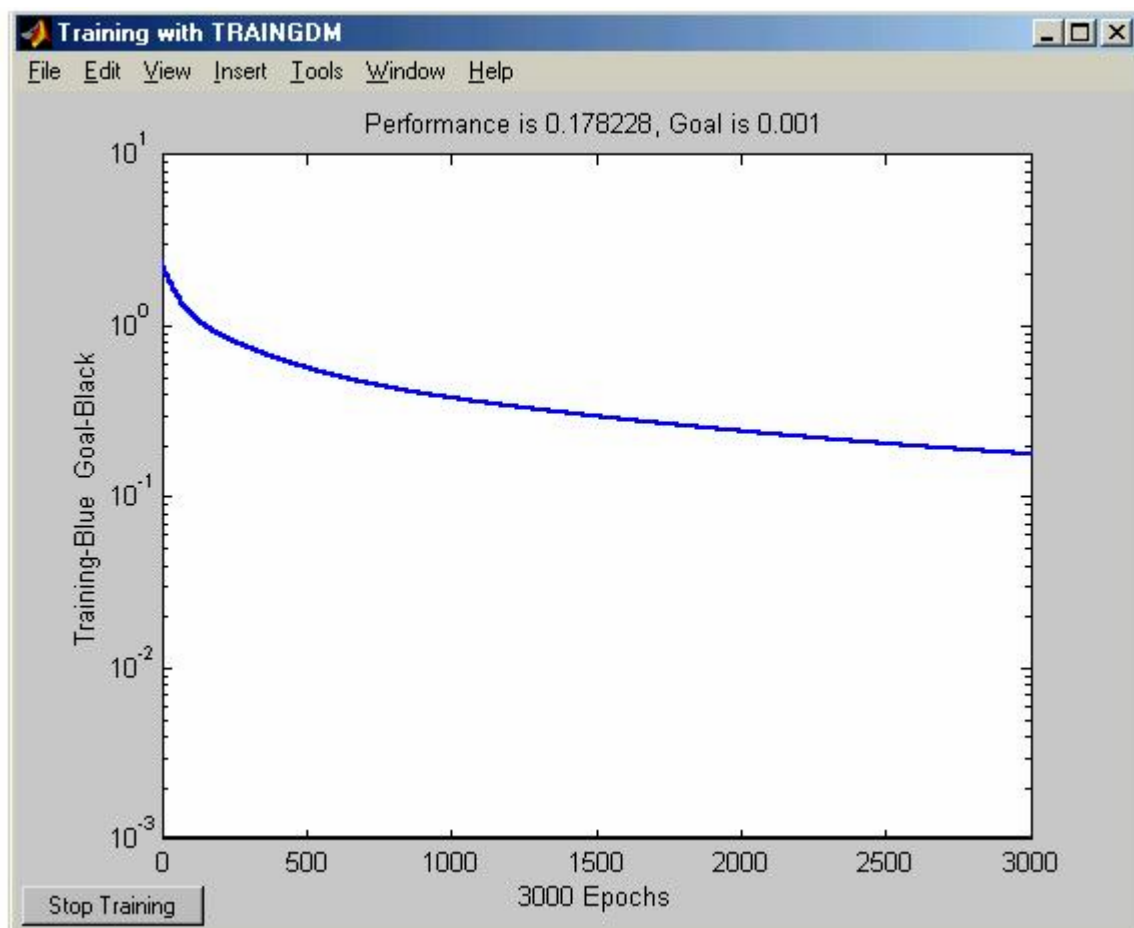


Figura 4.7 - Gráfico do resultado do treinamento.

Após a execução do treinamento, pode-se verificar, através do gráfico, a tendência de diminuição do erro à medida que o número de épocas vai aumentando. O valor do erro final (performance) para esta rede foi de 0,178228, atingido em 3000 epochs.

#### 4.4.4 Teste da rede

Para a avaliação da qualidade da previsão neural foi apresentada à rede a massa de dados de entrada correspondente aos meses de março, abril e maio de 2004, através da matriz S3, composta daqueles valores.

Efetuar-se-iam testes com diferentes configurações de redes MLP e RBF, sendo que o melhor resultado obtido foi o indicado na figura 4.8.

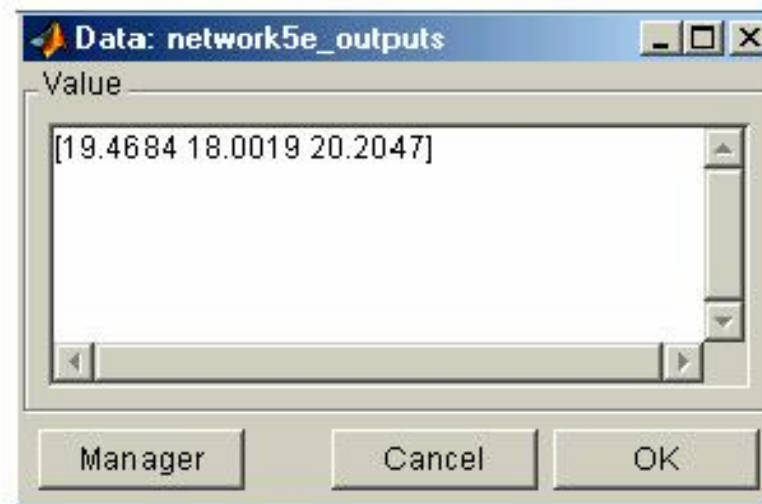


Figura 4.8 - Melhor resultado da simulação.

O resultado apresentado na figura acima foi obtido com uma rede MLP utilizando o algoritmo *Backpropagation* e com escolha dos seguintes parâmetros:

- Função de treinamento: TRAINGDM
- Função de aprendizagem: LEARNINGDM
- N° de neurônios na camada intermediária (*Layer 1*): 15
- N° de épocas de treinamento: 3000

#### 4.5 Estatística de comparação

Uma boa referência para avaliação dos resultados apresentados pela rede neural seria a comparação com um método estatístico consagrado e bastante utilizado na previsão, que é a Análise de Regressão Múltipla (**ARM**). A partir do mesmo conjunto de variáveis e da mesma massa de dados de entrada utilizada pela rede neural e utilizando o software MINITAB<sup>®</sup>, obteve-se a equação de regressão indicada na figura 4.9 e demais dados para análise daquele resultado.



**Regression Analysis: @ versus Mês; @-1; @-2; IGPM-1; IGPM-2; Selic-1**

The regression equation is  

$$\hat{@} = 0,93 + 0,0680 \text{ Mês} + 1,08 \text{ @-1} - 0,172 \text{ @-2} - 0,034 \text{ IGPM-1} + 0,215 \text{ IGPM-2} + 0,080 \text{ Selic-1}$$

Predictor	Coef	SE Coef	T	P	VIF
Constant	0,932	1,301	0,72	0,476	
Mês	0,06802	0,03763	1,81	0,075	1,1
@-1	1,0845	0,1249	8,68	0,000	6,8
@-2	-0,1725	0,1315	-1,31	0,194	7,7
IGPM-1	-0,0344	0,1878	-0,18	0,855	2,9
IGPM-2	0,2154	0,1877	1,15	0,255	2,9
Selic-1	0,0796	0,2999	0,27	0,792	1,1

S = 1,032                      R-Sq = 84,8%                      R-Sq(adj) = 83,4%  
PRESS = 95,2912              R-Sq(pred) = 78,72%

Figura 4.9 - Equação de regressão

A equação de regressão acima mostra a relação existente entre a variável de saída @ e as variáveis de entrada escolhidas para o modelo. Verifica-se na coluna VIF (*Variance Inflation Factors*) dois valores superiores a 5,0 que indicam a existência de multicolinearidade e, conseqüentemente, uma das variáveis (no caso, @-2) poderá ser retirada do modelo estatístico. A figura 4.10 mostra a nova equação de regressão múltipla, com os valores VIF das variáveis de entrada todos abaixo de 5,0 e traduzindo, com isso, não mais existir multicolinearidade. Os valores percentuais obtidos para os parâmetros **R-Sq** e **R-Sq (adj)** indicam que cerca de 90% da variação da cotação da arroba de boi podem ser explicados pelos elementos que compõem a equação de regressão, ou seja, as variáveis independentes explicam exatamente 90,4% das variações na variável dependente (fig. 4.10), o que traduz uma boa definição de modelo para a previsão desejada. Quanto ao significado dos parâmetros anteriormente citados, temos que:

- **R-Sq** denota o coeficiente de determinação múltipla, o qual mede o grau de ajustamento da equação de regressão múltipla aos dados amostrais. É uma medida de aderência da equação de regressão aos dados amostrais;
- **R-Sq (adj)** é o coeficiente **R-Sq** modificado de modo a levar em conta o número de variáveis e o tamanho da amostra.

**Regression Analysis: @ versus Mês; @-1; IGPM-1; IGPM-2; Selic-1**

The regression equation is  

$$\text{@} = -0,062 + 0,0587 \text{ Mês} + 0,963 \text{ @-1} + 0,088 \text{ IGPM-1} - 0,033 \text{ IGPM-2} + 0,209 \text{ Selic-1}$$

Predictor	Coef	SE Coef	T	P	VIF
Constant	-0,0615	0,9944	-0,06	0,951	
Mês	0,05873	0,03057	1,92	0,059	1,1
@-1	0,96308	0,04463	21,58	0,000	1,3
IGPM-1	0,0883	0,1513	0,58	0,562	2,7
IGPM-2	-0,0330	0,1519	-0,22	0,828	2,3
Selic-1	0,2092	0,2435	0,86	0,393	1,0

S = 0,8220                      R-Sq = 90,4%                      R-Sq(adj) = 89,6%  
PRESS = 50,3589                R-Sq(pred) = 88,46%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	5	394,527	78,905	116,78	0,000
Residual Error	62	41,893	0,676		
Total	67	436,420			

Figura 4.10 - Equação de regressão após a retirada da variável @-2 e outliers.

As relações entre as variáveis estarão bem definidas e os resultados reconhecidos e aceitos como confiáveis após efetuada a análise dos resíduos. Estes deverão ser idêntica e independentemente distribuídos (**iid**). A verificação de que os resíduos são independentemente distribuídos é feita através do fator VIF, já citado. Quanto ao fato de serem identicamente distribuídos, podemos constatar tal característica através do histograma de resíduos da figura 4.11, que mostra que os mesmos possuem uma distribuição bem próxima da gaussiana ou normal. A obtenção desse resultado foi possível após a análise dos gráficos de resíduos (Anexo B - figuras 5.1, 5.2 e 5.3), que sugeriam a retirada de três valores da amostra inicial (*outliers*), de forma que o modelo estatístico passou a contar com 68 observações, ao invés das 71 originais.

Quanto à correlação entre as variáveis, a figura 4.12 mostra este relacionamento através do **Coefficiente de Pearson (r)** e do **Valor P (P-Value)**. Através do *P-Value*, constata-se uma maior dependência da variável de saída @ em relação às variáveis que exprimem a cotação do mês anterior e a taxa de inflação, pois para elas o *P-Value* mostra valores menores que 0,05.

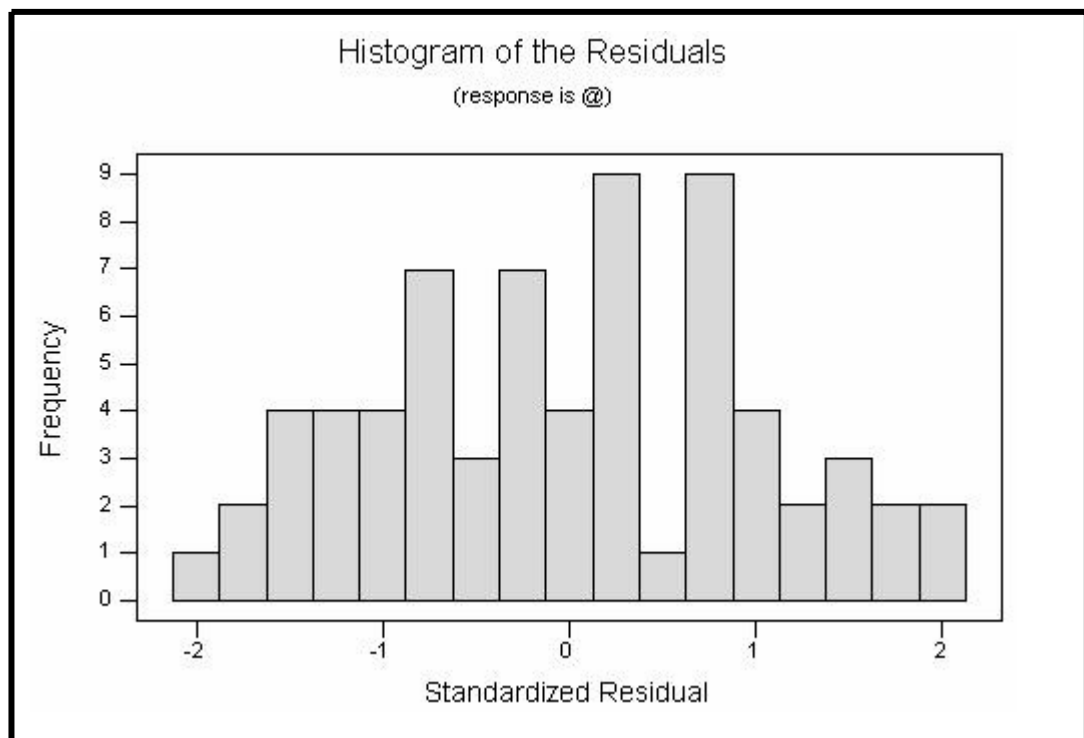


Figura 4.11 - Histograma dos resíduos.

Quando  $P\text{-Value} < 0,05$  temos a rejeição da hipótese nula ( $H_0$ ), a qual afirma que o modelo não é bom. Nessas condições, podemos afirmar que o modelo escolhido é bom, baseado no valor do  $P\text{-Value} < 0,05$  e ciente do fato de que a regressão se constitui num teste de hipóteses. Conclui-se então que o modelo permite significativamente prever a resposta, ou seja, estimar o valor da variável de saída @.

A documentação completa com a análise dos resultados da equação de regressão múltipla para o modelo estão mostradas no Anexo B. Lá pode-se verificar que o valor obtido na estatística de Durbin-Watson foi de 1,85 e mostra a ausência de correlação serial, já que na tabela dos valores críticos de DW para 60, ou mesmo 80 observações, e 5 variáveis, como é o caso do modelo em estudo (68 observações), o valor limite é 1,77 (Prob. = 0,05) Logo, acima deste valor como foi o caso, está garantida a inexistência de correlação serial.

#### 4.6 Avaliação do modelo proposto

A tabela 4.1 a seguir resume os resultados da previsão obtida pela rede neural e também pela análise de regressão múltipla, considerando-se as mesmas variáveis para os dois modelos. Pode-se verificar que os dois métodos apresentaram uma bom desempenho, indicado pelo erro percentual, com vantagem, neste caso, para a ARM nas três previsões. Com a

retirada da variável @-2 e eliminação de *outliers* para aperfeiçoamento do modelo estatístico, houve melhoria em duas estimativas e piora na terceira, conforme mostra a tabela 4.2.

<b>Correlations: Mês; @-1; IGPM-1; IGPM-2; Selic-1; @</b>					
	Mês	@-1	IGPM-1	IGPM-2	Selic-1
@-1	-0,068 0,579				
IGPM-1	0,139 0,260	-0,464 0,000			
IGPM-2	-0,000 0,997	-0,336 0,005	0,744 0,000		
Selic-1	0,091 0,461	0,045 0,715	0,069 0,575	0,001 0,992	
@	0,020 0,872	0,946 0,000	-0,406 0,001	-0,309 0,010	0,087 0,479
Cell Contents: Pearson correlation P-Value					

Figura 4.12 - Análise da correlação entre as variáveis

Cabe ressaltar que os valores obtidos na previsão da rede neural mostram-se diferentes a cada sessão de treinamento, em função da inicialização dos pesos e, caso se executem várias tentativas para a mesma previsão, o mais apropriado é utilizar-se o valor médio dos resultados obtidos.

Em alguns trabalhos pesquisados, os seus autores destacam um melhor desempenho na previsão para a rede RBF em relação à rede MLP. Neste trabalho não obtivemos este mesmo resultado, possivelmente em razão das características do modelo (quantidade de dados disponíveis, número de variáveis de entrada, etc.). Para cada caso uma determinada topologia se mostra mais eficiente.

Mês/Ano	Cotação Real	Previsão RNA	Previsão ARM	Erro RNA	Erro ARM	% Erro RNA	% Erro ARM
Mar/04	20,27	19,4684	19,57	- 0,8016	- 0,70	- 3,95	- 3,45
Abr/04	20,43	18,0019	19,83	- 2,4281	- 0,60	- 11,88	- 2,93
Mai/04	19,63	20,2047	20,14	0,5747	0,51	2,92	2,60

Tabela 4.1 - Resultados da previsão da RNA e da ARM (idênticas variáveis de entrada).

Mês/Ano	Cotação Real	Previsão RNA	Previsão ARM	Erro RNA	Erro ARM	% Erro RNA	% Erro ARM
Mar/04	20,27	19,4684	19,84	- 0,8016	- 0,43	- 3,95	- 2,10
Abr/04	20,43	18,0019	20,06	- 2,4281	- 0,37	- 11,88	- 1,82
Mai/04	19,63	20,2047	20,22	0,5747	0,59	2,92	3,01

Tabela 4.2 - Resultados da previsão com a ARM reduzida da variável @-2 e outliers.

É importante buscar referência através de processos alternativos e já consagrados como forma de comparação e validação para os resultados obtidos. Em particular os métodos estatísticos são bastante convenientes para este fim e a Tabela 4.3 mostra a correspondência existente entre elementos característicos das duas abordagens (BALESTRASSI, 2000).

Redes Neurais	Estatística
Rede	Modelo
Aprendizagem	Estimação
Aprendizagem supervisionada	Regressão
Generalização	Interpolação
Conjunto de treinamento	Observações
Pesos	Parâmetros
Entradas	Variáveis independentes
Saídas	Variáveis dependentes
Decaimento dos pesos	Regressão de Ridge

Tabela 4.3 - Terminologia correspondente entre RNA e Estatística.

Existe uma diferença fundamental, entretanto, na aproximação que essas duas técnicas usam para representar um ambiente real (SANTOS, 2001). Nos métodos estatísticos, os modelos matemáticos do ambiente são formulados primeiramente. Os parâmetros são então estimados e finalmente o modelo é validado. Com RNA, a arquitetura de rede é primeiramente construída (pelas características do problema) e depois disso a rede é treinada para predizer respostas a um conjunto de dados de treinamento da melhor forma possível (HAYKIN, 1994). A solução é então obtida através da experimentação e simulação ao invés de ser através de uma aproximação formal. Também, em estatística, algumas restrições são muitas vezes assumidas com respeito às distribuições que os dados devem ter. Nas RNA's raramente são feitas

suposições sobre as distribuições dos dados. Elas usam os dados para, implicitamente, desenvolver o relacionamento funcional que representa o ambiente estudado.

É importante deixar claro que os dois modelos (RNA e estatístico) não são de forma alguma concorrentes e sim complementares. Se a não-linearidade entre as variáveis aumentar, com certeza o resultado da rede neural tende a ser melhor, pois é nesta característica que as redes concentram a sua grande potencialidade.

#### **4.7 Considerações finais**

O presente capítulo mostrou um exemplo de aplicação da rede neural artificial na previsão de um particular indicador financeiro. Optou-se por utilizar a RNA para estimar o valor futuro de um tipo de *commodity* do agronegócio bastante negociada no mercado internacional: a arroba de carne bovina. Como elemento comparativo para avaliação dos resultados fornecidos pela rede, foi efetuada a mesma previsão, só que através de um processo estatístico convencional e de uso consagrado, que é a análise de regressão múltipla. No capítulo seguinte serão apresentadas as conclusões do trabalho e algumas recomendações para trabalhos futuros.

# CAPÍTULO 5

## 5. Conclusões e recomendações para trabalhos futuros

### 5.1 Considerações iniciais

O presente capítulo apresenta as conclusões do trabalho de pesquisa e das simulações executadas, confrontando-os com a proposta inicial estabelecida, além de sugerir algumas recomendações para trabalhos futuros.

### 5.2 Conclusões

O tema central desta pesquisa consistiu em apresentar as redes neurais artificiais como uma ferramenta alternativa para a solução de problemas de previsão de indicadores financeiros para análise de investimentos. Esta “nova” metodologia constitui-se em mais uma fonte de apoio à tomada de decisão em nível gerencial, já que num mercado eficiente, a previsão obtida indica tão somente uma tendência, sendo praticamente impossível prever-se o resultado correto, ou seja, com erro nulo. Quanto mais eficiente for o mercado, mais difícil será a previsão, pela quantidade de variáveis envolvidas e pelo fato de não se ter controle sobre várias delas.

A diferença entre os valores reais e os resultados da simulação deve-se, muito provavelmente, aos fatores não representados no conjunto de dados de entrada. A utilização da RNA prende-se ao fato de que não se conhece a real natureza da relação entre as entradas e a saída. Se esta relação fosse conhecida, poder-se-ia modelá-la diretamente.

Existem, reconhecidamente, fatores de difícil quantificação e previsão, como por exemplo, o aparecimento de um surto de febre aftosa (surto epidêmico), que são episódicos, esporádicos e influem, pelo menos temporariamente, na cotação da *commodity* em estudo, por criarem dificuldades e restrições já esperadas na sua negociação. A aplicação de cotas e barreiras tarifárias impostas por alguns países, bem como subsídios governamentais e outras atitudes protecionistas trazem também a sua quota de influência.

A previsão da cotação da arroba de carne bovina foi apenas um exemplo das inúmeras possibilidades de aplicação das redes neurais, constituindo-se numa justificativa ou pretexto para ampliar a divulgação desta ferramenta com suas características peculiares e distintas dos métodos estatísticos tradicionais.

A inevitável comparação de resultados com a análise de regressão múltipla serviu para concluir sobre o grau de eficiência da rede neural, já que não é possível deduzir sobre a superioridade de qualquer dos métodos, em razão das diferenças estruturais. Ambos apresentam vantagens e desvantagens, e o resultado, dependerá, evidentemente, do tipo de aplicação e do modelo utilizado. Quanto maior for a não-linearidade existente entre as variáveis envolvidas, melhor tende a ser o resultado da rede neural em relação ao modelo estatístico. Neste trabalho, buscou-se apenas mostrar a viabilidade da utilização da técnica de redes neurais, sabendo-se ainda que os resultados podem ser melhorados à medida que o conjunto de treinamento aumente, incorporando-se novos parâmetros que possam traduzir padrões de comportamento do mercado em resposta às constantes alterações das políticas de governo ou de outros elementos que de alguma forma, com maior ou menor significância, possam influir nos resultados da previsão.

A partir dos resultados obtidos, pode-se então, concluir sobre a adequação da metodologia como uma boa alternativa, ou mesmo complementação, às atuais técnicas estatísticas baseadas em séries temporais. Vale ressaltar a relativa facilidade de se trabalhar com as redes neurais a partir dos modernos programas computacionais que vêm permitindo popularizar cada vez mais o seu uso, já que elas não precisam de um especialista para a criação de sua base de conhecimentos. Por um lado elas não trabalham com regras e por outro sua aquisição de conhecimento é feita automaticamente, a partir de exemplos coletados em bancos de dados.

A criação das redes neurais é uma tarefa bem mais simples do que a criação de um sistema especialista e bem mais flexível do que uma técnica estatística. Esta flexibilidade permite às redes fazerem análises mais elaboradas, pouco perceptíveis por técnicas tradicionais ou dificilmente formalizáveis a partir do conhecimento subjetivo de um especialista.

### **5.3 Limitações**

O estabelecimento de um sistema previsor é definido a partir da escolha do conjunto de variáveis que o pesquisador julga serem as mais representativas para a tarefa. Como citado no item 4.2, esta escolha é intuitiva, baseada na experiência, e pode-se daí concluir sobre a existência de um certo grau de subjetividade na composição da matriz de entrada para o treinamento da rede.



Para o desenvolvimento de sistemas baseados em redes neurais existe um elenco variado de programas computacionais, cada um com suas características específicas, e naturalmente o pesquisador irá trabalhar com aquele que preferir, ou que esteja disponível.

Existem, ainda, vários métodos de aprendizagem das RNA's. Pode-se começar pelo método da retropropagação, pois é um método eficiente. Outra questão fundamental é a característica da massa de dados. Eles podem ser quantitativos ou qualitativos.

O passo seguinte é construir a rede. A dificuldade nesta tarefa está no grande número de parâmetros a serem considerados, como número de neurônios, número de camadas intermediárias, tipos de funções de ativação, etc., que irão determinar a qualidade da rede construída.

Todas essas questões, que são intrínsecas ao trabalho com RNA's, constituem limitações ao desempenho do modelo. Existem, entretanto, ações que podem reduzir em parte estas limitações. Podemos citar, entre outras:

- aumentar o conjunto de treinamento, com ampliação da base de dados;
- incluir dados relativos às políticas de governo, de forma a ensinar a rede as respostas do mercado à variações nestas políticas.
- trabalhar, se possível, com versões mais avançadas dos softwares dedicados, que possuem rotinas poderosas de pesquisa da configuração mais adequada para cada caso.
- utilizar a técnica de projetos de experimentos (DOE) na determinação da configuração ótima de treinamento da rede, caso não se disponha de software de RNA que já inclua esta facilidade.

Para finalizar, é importante ressaltar que por melhor que seja o modelo ou o método utilizado, haverá sempre algum erro na estimação por simulação.

## **5.4 Recomendações para trabalhos futuros**

Como citado ao longo deste trabalho, a utilização das RNA's atualmente está bastante diversificada, ensejando variadas aplicações, além das clássicas predição e classificação de padrões. Na área de economia e finanças tem-se encontrado um crescente interesse, uma vez que as redes precisam de dados para serem criadas, e este setor tem podido fornecê-los em abundância.

Os tipos de aplicação mais comuns em finanças são a previsão e a avaliação de riscos. Dentre estas podemos citar avaliação de créditos, riscos de inadimplência de empresas, riscos

de seguros, riscos de papéis financeiros, avaliação de cotações de ações, de imóveis, gestão de portfólios e muitos outros. Uma instituição financeira, por exemplo, leva em conta inúmeros tipos de papéis (ações, opções, etc.) e um sistema baseado em redes neurais poderia ser usado para analisar as características destes papéis, considerando-se riscos, valores conjunturais, decisões do governo, flutuações do mercado, etc. As redes neurais são bastante adaptadas a problemas do tipo não estruturado, onde as relações entre as variáveis ou mesmo sua importância não é conhecida. As redes permitem que se trabalhe com dados incompletos se estes não estão todos disponíveis ou mesmo com a existência de certas informações incorretas, pois elas aprenderão a ignorá-las.

Uma evolução natural no trabalho com as redes neurais é aproveitá-las no tratamento de dados qualitativos. Tem-se por hábito acumular e tratar somente dados quantitativos, mas o interessante das RNA's é também a sua capacidade de tratar dados qualitativos. Como exemplo de dados qualitativos de interesse podemos citar o setor de atividade de uma empresa ou tipo de produto vendido. Pode-se até mesmo usar variáveis menos acessíveis como características de gestão dos administradores, propensão da direção a assumir riscos, capacidade de inovação, flexibilidade a mudanças, etc.

A área de marketing e vendas mostra-se também um mercado bastante promissor para o uso de redes neurais. Uma proposta seria a simulação do comportamento do consumidor diante de novos produtos a partir do treinamento da rede com produtos de características conhecidas e já lançados no mercado, onde se pôde medir o grau de sucesso e volume de vendas obtido.

Uma outra aplicação poderia ser o uso de uma rede para sugerir produtos mais adaptados ao perfil de cada cliente. A rede seria treinada a associar perfis de clientes a produtos da empresa. Poder-se-ia assim esperar, por um lado, a obtenção de um maior grau de satisfação do cliente, por outro lado, apresentar produtos mais direcionados ao cliente em questão.

Seja para simulações, previsões ou sugestões, as RNA's constituem uma tecnologia interessante na área de marketing. Há que se levar em conta também o fato de que sua atualização, isto é, a aquisição de novos conhecimentos é bastante facilitada e rápida uma vez que o aprendizado é automático. Basta entrar com as características do novo produto ou da nova situação de mercado e refazer o processo de aprendizado.

De forma geral, no caso de aplicações em áreas onde o conhecimento evolui, é necessário retreinar a rede. Por exemplo, no caso de avaliações financeiras, o mercado é extremamente ágil e dinâmico e o que era válido em determinado momento, pode não

significar nada em tempos futuros. Deve-se mais uma vez retomar o processo de aprendizado, como já citado.

Em resumo, as redes neurais têm mostrado resultados positivos em um período mais curto que outras técnicas de criação de bases de conhecimentos como os sistemas especialistas. Sua facilidade de utilização, sua capacidade em tratar problemas pouco ou não estruturados, com dados incompletos ou incorretos são características originais e interessantes. É uma técnica que irá ainda evoluir bastante e irá sem dúvida permear nosso dia-a-dia, quer seja nos sistemas aqui focados para apoio à decisão, quer em aplicações completamente diversas como robótica, diagnósticos médicos, reconhecimento de caracteres ou simulação da visão por computador.

## ANEXO A - Dados do conjunto de treinamento e teste

### Conjunto de Treinamento

Mês/Ano	Mês	@-1	@-2	IGPM-1	IGPM-2	Selic-1	@
abr 98	4	23,62	23,44	0,19	0,18	1,71	23,67
mai 98	5	23,67	23,62	0,13	0,19	1,63	22,88
jun 98	6	22,88	23,67	0,14	0,13	1,60	22,65
jul 98	7	22,65	22,88	0,38	0,14	1,70	23,08
ago 98	8	23,08	22,65	-0,17	0,38	1,48	23,52
set 98	9	23,52	23,08	-0,16	-0,17	2,49	24,21
out 98	10	24,21	23,52	-0,08	-0,16	2,94	23,40
nov 98	11	23,40	24,21	0,08	-0,08	2,63	23,53
dez 98	12	23,53	23,40	-0,32	0,08	2,40	23,30
jan 99	1	23,30	23,53	0,45	-0,32	2,18	19,95
fev 99	2	19,95	23,30	0,84	0,45	2,38	16,27
mar 99	3	16,27	19,95	3,61	0,84	3,33	16,69
abr 99	4	16,69	16,27	2,83	3,61	2,35	18,20
mai 99	5	18,20	16,69	0,71	2,83	2,02	17,59
jun 99	6	17,59	18,20	-0,29	0,71	1,67	16,99
jul 99	7	16,99	17,59	0,36	-0,29	1,66	17,91
ago 99	8	17,91	16,99	1,55	0,36	1,57	17,36
set 99	9	17,36	17,91	1,56	1,55	1,49	18,47
out 99	10	18,47	17,36	1,45	1,56	1,38	20,16
nov 99	11	20,16	18,47	1,70	1,45	1,39	21,66
dez 99	12	21,66	20,16	2,39	1,70	1,60	22,10
jan 00	1	22,10	21,66	1,81	2,39	1,46	22,81
fev 00	2	22,81	22,10	1,24	1,81	1,45	22,27
mar 00	3	22,27	22,81	0,35	1,24	1,45	21,67
abr 00	4	21,67	22,27	0,15	0,35	1,30	21,37
mai 00	5	21,37	21,67	0,23	0,15	1,49	19,91
jun 00	6	19,91	21,37	0,31	0,23	1,39	21,40
jul 00	7	21,40	19,91	0,85	0,31	1,31	22,37
ago 00	8	22,37	21,40	1,57	0,85	1,41	23,00
set 00	9	23,00	22,37	2,39	1,57	1,22	22,31

Tabela 5.1(a) - Matriz do conjunto de treinamento.

**Conjunto de Treinamento (cont.)**

Mês/Ano	Mês	@-1	@-2	IGPM-1	IGPM-2	Selic-1	@
out 00	10	22,31	23,00	1,16	2,39	1,29	22,59
nov 00	11	22,59	22,31	0,38	1,16	1,22	21,33
dez 00	12	21,33	22,59	0,29	0,38	1,20	20,71
jan 01	1	20,71	21,33	0,63	0,29	1,27	20,52
fev 01	2	20,52	20,71	0,62	0,63	1,02	19,77
mar 01	3	19,77	20,52	0,23	0,62	1,26	19,05
abr 01	4	19,05	19,77	0,56	0,23	1,19	19,02
mai 01	5	19,02	19,05	1,00	0,56	1,34	17,90
jun 01	6	17,90	19,02	0,86	1,00	1,27	17,28
jul 01	7	17,28	17,90	0,98	0,86	1,50	16,81
ago 01	8	16,81	17,28	1,48	0,98	1,60	16,81
set 01	9	16,81	16,81	1,38	1,48	1,32	16,08
out 01	10	16,08	16,81	0,31	1,38	1,53	16,96
nov 01	11	16,96	16,08	1,18	0,31	1,39	18,01
dez 01	12	18,01	16,96	1,10	1,18	1,39	19,48
jan 02	1	19,48	18,01	0,22	1,10	1,53	19,22
fev 02	2	19,22	19,48	0,36	0,22	1,25	18,83
mar 02	3	18,83	19,22	0,06	0,36	1,37	19,07
abr 02	4	19,07	18,83	0,09	0,06	1,48	18,30
mai 02	5	18,30	19,07	0,56	0,09	1,41	17,00
jun 02	6	17,00	18,30	0,83	0,56	1,33	15,69
jul 02	7	15,69	17,00	1,54	0,83	1,54	15,03
ago 02	8	15,03	15,69	1,95	1,54	1,44	15,87
set 02	9	15,87	15,03	2,32	1,95	1,38	15,09
out 02	10	15,09	15,87	2,40	2,32	1,65	14,06
nov 02	11	14,06	15,09	3,87	2,40	1,54	16,13
dez 02	12	16,13	14,06	5,19	3,87	1,74	15,93
jan 03	1	15,93	16,13	3,75	5,19	1,97	19,59
fev 03	2	19,59	15,93	2,33	3,75	1,83	19,59
mar 03	3	19,59	19,59	2,28	2,33	1,78	19,59

Tabela 5.1(b) - Continuação da matriz do conjunto de treinamento.

### **Conjunto de Treinamento (cont.)**

<b>Mês/Ano</b>	<b>Mês</b>	<b>@-1</b>	<b>@-2</b>	<b>IGPM-1</b>	<b>IGPM-2</b>	<b>Selic-1</b>	<b>@</b>
abr 03	4	19,59	19,59	1,53	2,28	1,87	18,13
mai 03	5	18,13	19,59	0,92	1,53	1,97	18,16
jun 03	6	18,16	18,13	-0,26	0,92	1,86	18,76
jul 03	7	18,76	18,16	-1,00	-0,26	2,08	19,34
ago 03	8	19,34	18,76	-0,42	-1,00	1,77	19,51
set 03	9	19,51	19,34	0,38	-0,42	1,68	20,68
out 03	10	20,68	19,51	1,18	0,38	1,64	21,09
nov 03	11	21,09	20,68	0,38	1,18	1,34	21,02
dez 03	12	21,02	21,09	0,49	0,38	1,37	20,85
jan 04	1	20,85	21,02	0,61	0,49	1,27	21,24
fev 04	2	21,24	20,85	0,88	0,61	1,08	20,22

Tabela 5.1(c) - Continuação da matriz do conjunto de treinamento.

### **Conjunto de Teste/Validação**

<b>Mês/Ano</b>	<b>Mês</b>	<b>@-1</b>	<b>@-2</b>	<b>IGPM-1</b>	<b>IGPM-2</b>	<b>Selic-1</b>	<b>@</b>
mar 04	3	20,22	21,24	0,69	0,88	1,38	20,27
abr 04	4	20,27	20,22	1,13	0,69	1,18	20,43
mai 04	5	20,43	20,27	1,21	1,13	1,23	19,63

Tabela 5.2 - Matriz do conjunto de teste.

## ANEXO B - Resultados da Análise de Regressão Múltipla

**Regression Analysis: @ versus Mês; @-1; IGPM-1; IGPM-2; Selic-1**

The regression equation is  

$$\text{@} = -0,062 + 0,0587 \text{ Mês} + 0,963 \text{ @-1} + 0,088 \text{ IGPM-1} - 0,033 \text{ IGPM-2} + 0,209 \text{ Selic-1}$$

Predictor	Coef	SE Coef	T	P	VIF
Constant	-0,0615	0,9944	-0,06	0,951	
Mês	0,05873	0,03057	1,92	0,059	1,1
@-1	0,96308	0,04463	21,58	0,000	1,3
IGPM-1	0,0883	0,1513	0,58	0,562	2,7
IGPM-2	-0,0330	0,1519	-0,22	0,828	2,3
Selic-1	0,2092	0,2435	0,86	0,393	1,0

S = 0,8220                      R-Sq = 90,4%                      R-Sq(adj) = 89,6%  
PRESS = 50,3589                  R-Sq(pred) = 88,46%

**Analysis of Variance**

Source	DF	SS	MS	F	P
Regression	5	394,527	78,905	116,78	0,000
Residual Error	62	41,893	0,676		
Total	67	436,420			

Source	DF	Seq SS
Mês	1	0,174
@-1	1	393,477
IGPM-1	1	0,327
IGPM-2	1	0,051
Selic-1	1	0,499

Obs	Mês	@	Fit	SE Fit	Residual	St Resid
1	4,0	23,6700	23,2900	0,2071	0,3800	0,48
2	5,0	22,8800	23,3745	0,1958	-0,4945	-0,62
3	6,0	22,6500	22,6690	0,1690	-0,0190	-0,02
4	7,0	23,0800	22,5480	0,1679	0,5320	0,66
5	8,0	23,5200	22,9183	0,1892	0,6017	0,75
6	9,0	24,2100	23,6312	0,2920	0,5788	0,75
7	10,0	23,4000	24,4553	0,3900	-1,0553	-1,46
8	11,0	23,5300	23,6806	0,3276	-0,1506	-0,20
9	12,0	23,3000	23,7758	0,3172	-0,4758	-0,63
10	3,0	16,6900	16,7719	0,5928	-0,0819	-0,14 X
11	4,0	18,2000	16,8696	0,3583	1,3304	1,80
12	5,0	17,5900	18,1520	0,3607	-0,5620	-0,76
13	6,0	16,9900	17,5318	0,2409	-0,5418	-0,69
14	7,0	17,9100	17,1010	0,2235	0,8090	1,02
15	8,0	17,3600	18,1106	0,1936	-0,7506	-0,94
16	9,0	18,4700	17,5845	0,1544	0,8855	1,10
17	10,0	20,1600	18,6792	0,1655	1,4808	1,84
18	11,0	21,6600	20,3933	0,1921	1,2667	1,58
19	12,0	22,1000	21,9933	0,2711	0,1067	0,14

Figura 5.1 (a) - Resultados da Análise de Regressão Múltipla.

20	1,0	22,8100	21,6677	0,2883	1,1423	1,48
21	2,0	22,2700	22,3769	0,2505	-0,1069	-0,14
22	3,0	21,6700	21,8558	0,2014	-0,1858	-0,23
23	4,0	21,3700	21,3170	0,1682	0,0530	0,07
24	5,0	19,9100	21,1403	0,1468	-1,2303	-1,52
25	6,0	21,4000	19,7764	0,1349	1,6236	2,00R
26	7,0	22,3700	21,2984	0,1711	1,0716	1,33
27	8,0	23,0000	22,3580	0,2177	0,6420	0,81
28	9,0	22,3100	23,0324	0,3104	-0,7224	-0,95
29	10,0	22,5900	22,3055	0,2944	0,2845	0,37
30	11,0	21,3300	22,5910	0,2540	-1,2610	-1,61
31	12,0	20,7100	21,4499	0,2351	-0,7399	-0,94
32	1,0	20,5200	20,2544	0,2332	0,2656	0,34
33	2,0	19,7700	20,0657	0,2217	-0,2957	-0,37
34	3,0	19,0500	19,4182	0,1753	-0,3682	-0,46
35	4,0	19,0200	18,8109	0,1846	0,2091	0,26
36	5,0	17,9000	18,9001	0,1491	-1,0001	-1,24
37	6,0	17,2800	17,8386	0,1492	-0,5586	-0,69
38	7,0	16,8100	17,3636	0,1457	-0,5536	-0,68
39	8,0	16,8100	17,0308	0,1583	-0,2208	-0,27
40	9,0	16,0800	17,0056	0,1802	-0,9256	-1,15
41	10,0	16,9600	16,3140	0,2901	0,6460	0,84
42	11,0	18,0100	17,3032	0,2248	0,7068	0,89
43	12,0	19,4800	18,3373	0,2090	1,1427	1,44
44	1,0	19,2200	19,0612	0,2223	0,1588	0,20
47	4,0	18,3000	18,8550	0,1718	-0,5550	-0,69
48	5,0	17,0000	18,1980	0,1743	-1,1980	-1,49
49	6,0	15,6900	16,9963	0,1762	-1,3063	-1,63
50	7,0	15,0300	15,8912	0,2011	-0,8612	-1,08
51	8,0	15,8700	15,3061	0,2110	0,5639	0,71
52	9,0	15,0900	16,1804	0,2044	-1,0904	-1,37
53	10,0	14,0600	15,5393	0,2377	-1,4793	-1,88
54	11,0	16,1300	14,7102	0,3277	1,4198	1,88
55	12,0	15,9300	16,8724	0,4197	-0,9424	-1,33
56	2,0	19,5900	19,3875	0,3552	0,2025	0,27
57	3,0	19,5900	19,4783	0,2237	0,1117	0,14
58	4,0	18,1300	19,4912	0,2090	-1,3612	-1,71
59	5,0	18,1600	18,1357	0,1908	0,0243	0,03
60	6,0	18,7600	18,1162	0,2500	0,6438	0,82
61	7,0	19,3400	18,7724	0,2884	0,5676	0,74
62	8,0	19,5100	19,4006	0,2404	0,1094	0,14
63	9,0	20,6800	19,6557	0,1944	1,0243	1,28
64	10,0	21,0900	20,8771	0,1778	0,2129	0,27
65	11,0	21,0200	21,1708	0,2242	-0,1508	-0,19
66	12,0	20,8500	21,2046	0,2124	-0,3546	-0,45
67	1,0	21,2400	20,3808	0,2226	0,8592	1,09
68	2,0	20,2200	20,7953	0,2333	-0,5753	-0,73

R denotes an observation with a large standardized residual  
X denotes an observation whose X value gives it large influence.

Durbin-Watson statistic = 1,85

Figura 5.1 (b) - Resultados da Análise de Regressão Múltipla (cont.)





**Gráficos de Resíduos**

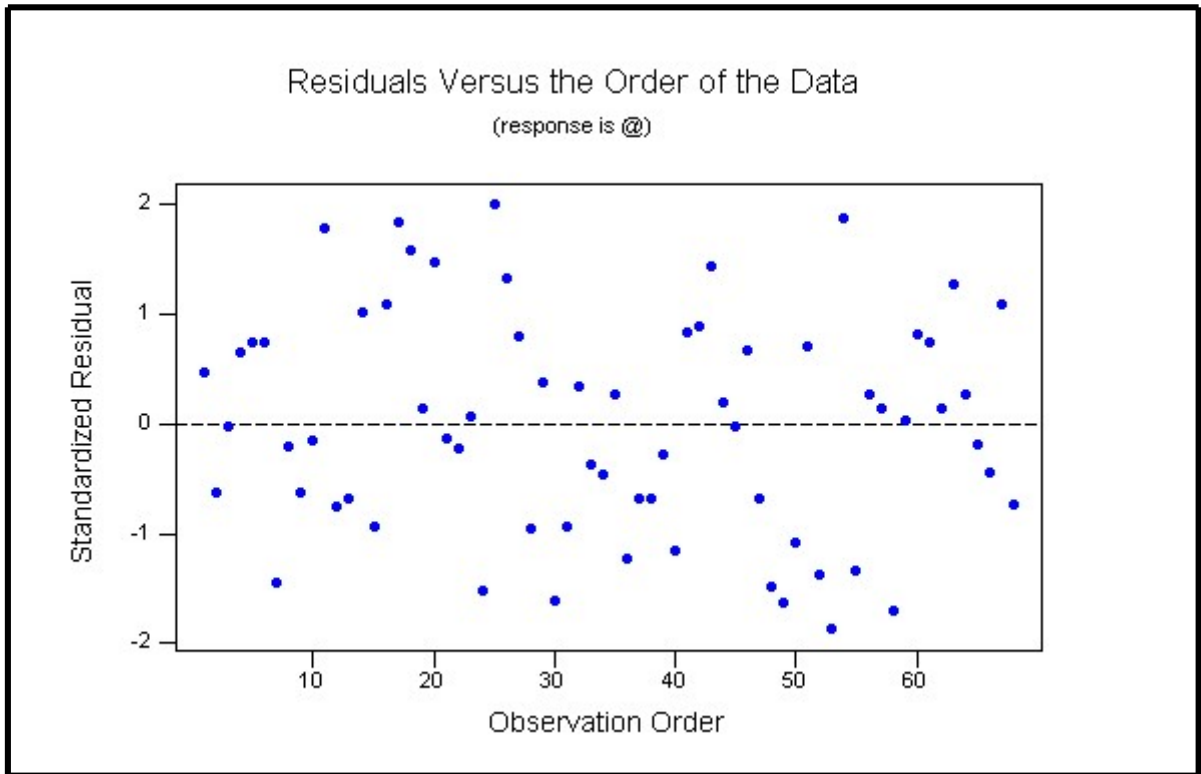


Figura 5.4 - Gráfico de resíduos x ordem dos dados.

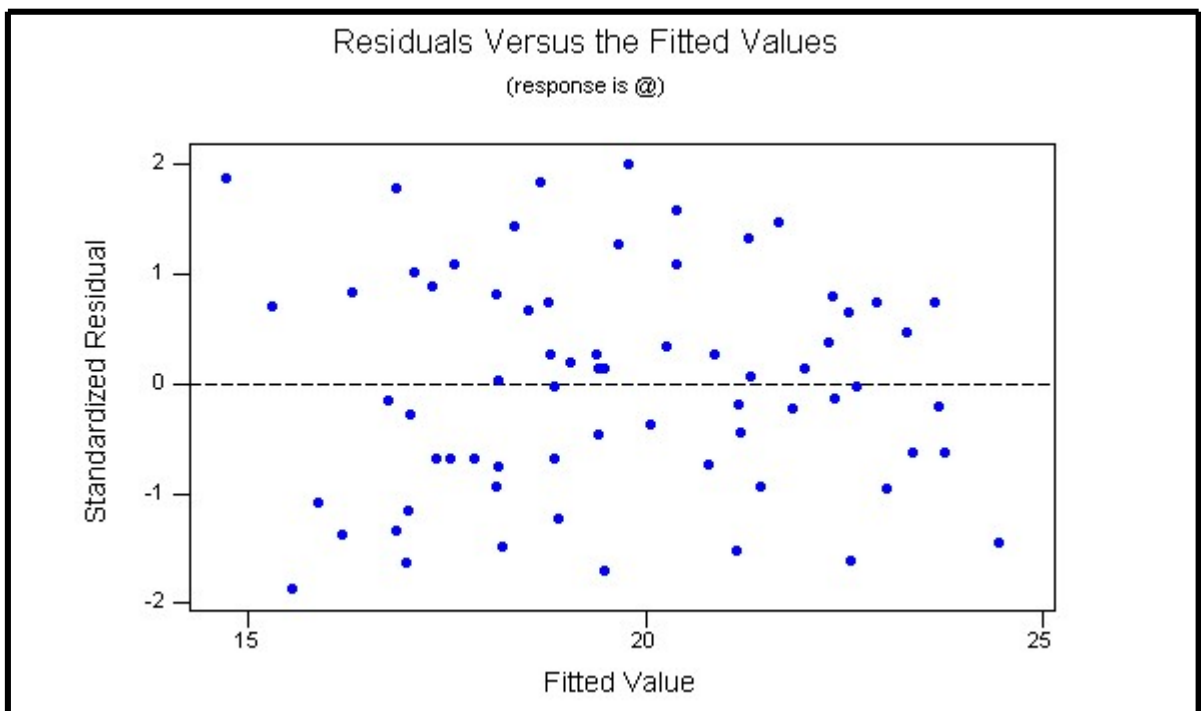


Figura 5.5 - Gráfico de resíduos x valores ajustados.

**Gráficos de Resíduos (cont.)**

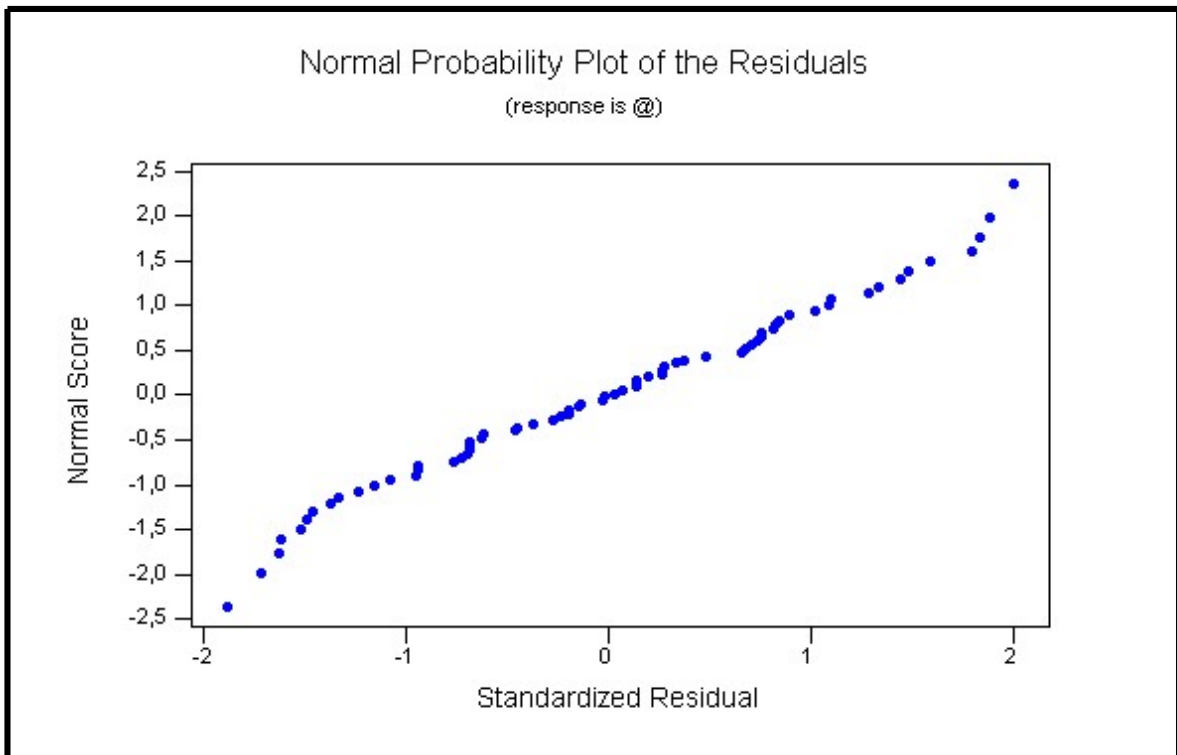


Figura 5.6 - Gráfico da probabilidade normal dos resíduos.

Observations	N	Prob.	1		2		3		4		5	
			D-L	D-U	D-L	D-U	D-L	D-U	D-L	D-U	D-L	D-U
15	0.05	1.08	1.36	0.95	1.54	0.82	1.75	0.69	1.97	0.56	2.21	
	0.01	0.81	1.07	0.7	1.25	0.59	1.46	0.49	1.70	0.39	1.96	
20	0.05	1.20	1.71	1.10	1.54	1.00	1.68	0.90	1.83	0.79	1.99	
	0.01	0.95	1.15	0.86	1.27	0.77	1.41	0.68	1.57	0.60	1.74	
25	0.05	1.29	1.45	1.21	1.55	1.12	1.66	1.04	1.77	0.95	1.89	
	0.01	1.05	1.21	0.98	1.30	0.90	1.41	0.83	1.52	0.75	1.65	
30	0.05	1.35	1.49	1.28	1.57	1.21	1.65	1.14	1.74	1.07	1.83	
	0.01	1.13	1.26	1.07	1.34	1.01	1.42	0.94	1.51	0.88	1.61	
40	0.05	1.44	1.54	1.39	1.60	1.34	1.66	1.39	1.72	1.23	1.79	
	0.01	1.25	1.34	1.20	1.40	1.15	1.46	1.10	1.52	1.05	1.58	
50	0.05	1.50	1.59	1.46	1.63	1.42	1.67	1.38	1.72	1.34	1.77	
	0.01	1.32	1.40	1.28	1.45	1.24	1.49	1.20	1.54	1.16	1.59	
60	0.05	1.55	1.62	1.51	1.65	1.48	1.69	1.44	1.73	1.41	1.77	
	0.01	1.38	1.45	1.35	1.48	1.32	1.52	1.28	1.56	1.25	1.60	
80	0.05	1.61	1.66	1.59	1.69	1.56	1.72	1.53	1.74	1.51	1.77	
	0.01	1.47	1.52	1.44	1.54	1.42	1.57	1.39	1.60	1.36	1.62	
100	0.05	1.65	1.69	1.63	1.72	1.61	1.74	1.59	1.76	1.57	1.78	
	0.01	1.52	1.56	1.50	1.58	1.48	1.60	1.46	1.63	1.44	1.65	

Tabela 5.3 - Valores críticos de Durbin-Watson.