



UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA

UMA ABORDAGEM PARA PARAMETRIZAÇÃO DE REDES NEURAIS
DE FUNÇÃO DE BASE RADIAL BASEADA NA COMBINAÇÃO DE
PROCEDIMENTOS NÃO SUPERVISIONADOS E DE UMA NOVA
PROPOSIÇÃO DE ESCALONAMENTO DE PARÂMETROS

Flávia Aparecida Oliveira Santos

Itajubá, Agosto de 2015.



UNIVERSIDADE FEDERAL DE ITAJUBÁ

**PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA**

Flávia Aparecida Oliveira Santos

**UMA ABORDAGEM PARA PARAMETRIZAÇÃO DE REDES NEURAIS
DE FUNÇÃO DE BASE RADIAL BASEADA NA COMBINAÇÃO DE
PROCEDIMENTOS NÃO SUPERVISIONADOS E DE UMA NOVA
PROPOSIÇÃO DE ESCALONAMENTO DE PARÂMETROS**

Tese submetida ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Itajubá como parte dos requisitos para a obtenção do Título Doutor em Ciências em Engenharia Elétrica.

Área de concentração: Sistemas Elétricos de Potência

Orientador: Dr. Benedito Isaías de Lima Lopes
Dr. Carlos Alberto Murari Pinheiro

Agosto de 2015

Itajubá - MG

Dedicatória

Dedico esta tese aos meus amigos e familiares, em especial ao meu filho Davi.

Agradecimentos

A DEUS, por providenciar esta oportunidade, sempre me protegendo e iluminando.

Ao professor Isaías, meus agradecimentos pela paciência nos momentos de indecisões e pela confiança em mim depositada.

Ao professor Pinheiro, pelos conhecimentos transmitidos, pela enorme dedicação e entusiasmo.

A minha família por compreender e partilhar as angústias e aflições por mim demonstradas, em especial ao meu esposo pelo companheirismo e cumplicidade acreditando nos meus sonhos, respeitando-os, pelos vários incentivos e admirando-me constantemente, em especial por tudo o que temos compartilhado.

Aos amigos, Nenen e Gil, pela hospitalidade e carinho, a minha eterna gratidão.

Muitos foram os que contribuíram de forma decisiva para a conclusão deste trabalho, agradeço sinceramente a todos.

Entretanto, não poderia deixar de mencionar o amor e a dedicação da minha Mãe que, incansavelmente me deu apoio nas incontáveis vezes que necessitei deixar os meus afazeres para dar prosseguimento ao curso, ela esteve sempre pronta para cuidar do meu filho.

A cada um, que direta ou indiretamente colaborou para que este projeto fosse concluído, da mesma maneira todos os amigos que me acompanham e aqueles que conquistei, cada um do seu jeito, me estimulando incessantemente.

Finalmente, agradeço ao IESTI – Instituto de Engenharia de Sistemas e Tecnologia da Informação e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES pela bolsa concedida.

Resumo

Neste trabalho será apresentada uma abordagem para parametrização de redes RBF (*Radial Basis Function*) baseada na combinação de procedimentos não supervisionados e uma nova proposição de escalonamento de parâmetros. A metodologia consiste em combinar procedimentos referenciados na literatura com o objetivo de obter modelos de redes RBF com melhores exatidões e algoritmos computacionais mais compactos. Alguns exemplos serão utilizados para ilustrar o emprego da abordagem proposta e também servirão para realizar comparações de resultados com os principais procedimentos referenciados em textos da área.

As redes neurais com funções de base radial (RBF) são modelos não lineares que podem realizar um mapeamento (interpolação) eficiente de dados de entrada e saída de diversos tipos de sistemas, resultando em boa capacidade de generalização aliada a processamentos de informações de forma compacta, possibilitando na representação eficiente de sistemas dinâmicos complexos e de séries temporais, por exemplo. Os bons resultados na capacidade de interpolação de uma RBF dependem de alguns parâmetros que devem ser adequadamente ajustados. Algumas abordagens foram desenvolvidas nesse contexto.

O procedimento proposto neste trabalho mostrou-se ser uma alternativa promissora, com aplicação direta e que apresenta uma exatidão adequada para várias aplicações práticas. Exemplos como aproximações de funções, modelagem de sistemas dinâmicos não lineares, previsão de série temporal e classificação de padrões serão discutidos com a finalidade de exemplificar os procedimentos propostos, além de servir de comparações com os resultados obtidos por outras técnicas utilizadas em redes RBF.

Palavras-Chaves: Redes de função de base radial; ajustes de parâmetros; redes neurais artificiais, aproximações de funções; modelagem de sistemas dinâmicos; previsões de séries temporais; classificação de padrões.

Abstract

In this work an approach to parameterization of Radial-Basis Function (RBF) Neural Networks based on a combination of unsupervised procedures and a new way for scaling parameters will be presented.

The methodology consists of combining procedures as referenced in the literature in order to obtain accurate and more compact computational in the algorithms of RBF networks.

Some examples will be used to illustrate the applications of the approach, and to enable comparisons of results with main procedures referenced in papers.

Neural networks with radial basis functions are nonlinear models that can perform an efficient interpolation between data input and output of various types of systems, resulting in good ability to generalization of the models, enabling an efficient representation of complex systems and time series, for example.

The interpolation capability in RBF models depends on some parameters that should be adjusted. Some approaches have been developed in this context.

The procedure proposed in this work has proved to be a promising alternative, with direct application and an adequate accuracy for many practical applications. Examples such as approximations of functions, modeling of nonlinear dynamic systems, time series prediction and patterns classification will be discussed with the purpose of exemplify the approach, besides serving as comparisons with results obtained through other techniques.

KeyWords: *Radial basis function neural network; parameter adjustments; artificial neural networks, approximations of functions; modeling of dynamic systems; time series prediction; pattern classification.*

“A dúvida é o princípio da sabedoria”

Aristóteles

Sumário

Dedicatória	3
Agradecimentos	4
Resumo	5
Abstract	6
Sumário	8
Lista de Figuras	10
Lista de Tabelas	11
Lista de Siglas e Abreviações.....	12
1. Introdução.....	13
1.1 Motivação e Objetivos.....	14
1.2 Estrutura do Trabalho	15
2. Revisão Bibliográfica	16
3. Revisão de Conceitos.....	25
3.1 Modelagem de Sistemas Dinâmicos.....	25
3.1.1 Representação Matemática de Modelos Lineares e Não-Lineares.....	27
3.1.2 Estimação de Parâmetros – Método dos Mínimos Quadrados	33
3.2 Previsão de Séries Temporais.....	34
4. Metodologia	36
5. Simulações.....	40
5.1 Exemplos em Aproximações de Funções.....	40
5.2 Exemplo de Modelagem de Sistema Dinâmico Não Linear.....	52
5.3 Exemplo de Previsão de Série Temporal.....	61
5.4 Exemplo de Classificação de Padrões	66

6. Conclusão	70
6.1 Sugestões para Trabalhos Futuros	71
Referências Bibliográficas	72

Lista de Figuras

Figura 2.1 - Estrutura típica de uma rede neural RBF com múltiplas entradas e uma saída.	16
Figura 3.1 - Exemplo de modelo TS com três regras (modelos locais afins).	30
Figura 4.1 - Estrutura típica de uma rede RBF.	36
Figura 4.2 - Fluxograma da abordagem CPNSEP proposta.	39
Figura 5.1 – Dados de treino para o Exemplo 1 e as aproximações obtidas pela rede RBF associada. ..	42
Figura 5.2 - Dados de treino do Exemplo 1 e as aproximações obtidas via rede RBF associada.	44
Figura 5.3 - Dados da função (5.1) e os respectivos agrupamentos.	45
Figura 5.4 - Dados da função (5.1) e as aproximações dadas pela rede RBF resultante.	48
Figura 5.5 - Dados da função (5.2) e os respectivos agrupamentos.	49
Figura 5.6 - Dados da função (5.2) e as aproximações realizadas pela rede RBF associada.	52
Figura 5.7 - Dados originais de entrada e saída de um sistema de nível.	53
Figura 5.8 - Dados originais de entrada e saída de um processo de nível.	54
Figura 5.9 - Dados do processo de nível e do modelo ARX associado.	55
Figura 5.10 - Dados do processo de nível e do modelo NARX correspondente.	56
Figura 5.11 - Exemplo de estrutura de ANN utilizada para modelagem de sistemas.	56
Figura 5.12 - Dados do sistema de nível e os respectivos agrupamentos.	58
Figura 5.13 - Dados reais do sistema de nível e os valores estimados pela rede RBF.	60
Figura 5.14 - Dados da série de preço de arroba de boi.	61
Figura 5.15 - Dados da série temporal e os respectivos agrupamentos.	63
Figura 5.16 - Dados da série temporal e os valores aproximados pelas redes RBF.	65
Figura 5.17 - Dados da série temporal e os valores estimados pelas redes consideradas.	66
Figura A.0.1 - Arquitetura ANFIS de 5 camadas. Fonte: (Jang, 1993), (Faustino, 2011).	80

Lista de Tabelas

Tabela 5.1 - Valores dos centros das funções da rede referente ao Exemplo 1.	41
Tabela 5.2 - Valores dos pesos da camada de saída da rede referente ao Exemplo 1.	41
Tabela 5.3 - Valores das larguras das funções da rede referente ao Exemplo 1.	42
Tabela 5.4 - Valores dos pesos da camada de saída da rede referente ao Exemplo 1.	42
Tabela 5.5 - Valores das larguras das funções da rede referente ao Exemplo 1.	43
Tabela 5.6 - Valores dos pesos da camada de saída da rede referente ao Exemplo 1.	43
Tabela 5.7 - Dados de centros, larguras e pesos da rede RBF do Exemplo 1 com os erros associados. .	43
Tabela 5.8 - Valores dos centros das funções da rede referente ao Exemplo 1.	45
Tabela 5.9 - Valores das larguras das funções da rede referente ao Exemplo 1.	45
Tabela 5.10 - Valores dos pesos da camada de saída da rede referente ao Exemplo 1.	45
Tabela 5.11 - Dados de centros, larguras e pesos da rede e o valor do erro associado.	46
Tabela 5.12 - Valores dos desvios padrões das funções da rede associada ao Exemplo 1.	46
Tabela 5.13 - Valores das larguras das funções radiais da rede referente ao Exemplo 1.	47
Tabela 5.14 - Valores dos pesos da camada de saída da rede referente ao Exemplo 1.	47
Tabela 5.15 - Dados dos centros, larguras e pesos da RBF com a abordagem proposta.	47
Tabela 5.16 - SEQ associado ao método utilizado em relação ao Exemplo 1.	48
Tabela 5.17 - Valores dos centros das funções da rede referente ao Exemplo 2.	50
Tabela 5.18 - Valores dos pesos da camada de saída da rede referente ao Exemplo 2.	50
Tabela 5.19 - Valores das larguras das funções da rede referente ao Exemplo 2.	51
Tabela 5.20 - Valores dos pesos da camada de saída da rede referente ao Exemplo 2.	51
Tabela 5.21 - Valores dos centros das funções da rede referente ao Exemplo 3.	58
Tabela 5.22 - Valores das larguras das funções da rede referente ao Exemplo 3.	59
Tabela 5.23 - Valores dos pesos da camada de saída da rede referente ao Exemplo 3.	59
Tabela 5.24 - Comparação de resultados dos modelos do processo de nível.	61
Tabela 5.25 - Valores dos centros das funções da rede referente ao Exemplo 4.	63
Tabela 5.26 - Valores das larguras das funções da rede referente ao Exemplo 4.	64
Tabela 5.27 - Valores dos pesos da camada de saída da rede referente ao Exemplo 4.	64
Tabela 5.28 - Valores dos centros das funções da rede referente ao Exemplo 5.	67
Tabela 5.29 - Valores das larguras das funções da rede referente ao Exemplo 5.	68
Tabela 5.30 - Valores dos pesos da camada de saída da rede referente ao Exemplo 5.	68
Tabela 5.31 - Comparação de resultados (em %).	69

Lista de Siglas e Abreviações

ANFIS - Adaptive Network Based Fuzzy Inference System

ANN – Artificial Neural Network

AR - AutoRegressive

ARIMA - AutoRegressive Integrated Moving Average

ARMA – AutoRegressive Moving Average

ARMAX – AutoRegressive Moving Average with eXogenous input

ARX – AutoRegressive with eXogenous input

DENFIS - Dynamic Evolving Neural-Fuzzy Inference System

FCM - Fuzzy C-Means Model

FIS – Fuzzy Inference System

FSOM - Fuzzy Self-Organization Map

MAPE - Mean Absolute Percentage Error

MLP - Multi Layer Perceptron

NARX – Nonlinear AutoRegressive with exogenous input

NARMAX – Nonlinear AutoRegressive Moving Average with exogenous input

NEFClass - Neuro Fuzzy Classification

SARIMA - AutoRegressive Integrated Moving Average Seasonal

SSE - Sum of the squared error

RBF - Radial Basis Function

CAPÍTULO 1

1. Introdução

Questões relacionadas a aproximações de funções, representações de sistemas dinâmicos complexos, previsão de séries temporais, classificações de padrões e outras, são temas de grande interesse em áreas diversas como modelagem de sistemas, aplicações financeiras, sistemas de controle, climatologia, medicina, entre outros. Neste contexto geral, a utilização de técnicas de inteligência artificial se mostrou ser uma alternativa eficiente em relação a aplicações de métodos convencionais, para as classes de problemas citados, sendo que alguns exemplos são encontrados em Noman et al., (2009), Jin-Yue; Bao-Ling, (2013), Li S., (2013) Jun e Zuhua (2014).

A determinação de relações pertinentes entre as variáveis envolvidas em aplicações práticas em áreas como climatologia, economia, e outras no sentido de se obter modelos associados com um determinado grau de exatidão, juntamente com a dificuldade de se escolher adequadamente as amostras de maiores influências, torna difícil o emprego de modelos analíticos para tratar problemas desta natureza. Assim, o uso de ferramentas baseadas em *inteligência computacional* vem ganhando força e a literatura mostra diversas aplicações realizadas com sucesso nestes contextos. Alguns estudos neste sentido podem ser encontrados em Aslanargun et al. (2007), Wu e Liu (2012), Santos et al. (2012), Li et al. (2013).

As redes neurais artificiais destacam-se como uma alternativa eficiente no tratamento dos problemas mencionados devido à capacidade de modelagem de funções não lineares, na representação de sistemas complexos, e na robustez quanto à incidência de ruídos nos dados disponíveis. As redes neurais apresentam características específicas, intrínsecas à sua formulação, que possibilitam a aproximação de qualquer função contínua, inclusive funções não lineares complexas, e também podem generalizar os resultados obtidos para dados previamente desconhecidos, ou seja, produzir respostas coerentes e apropriadas para padrões ou exemplos que não foram utilizados em seu treinamento (HAYKIN, 2007; BISHOP, 1997). Entre as estruturas de redes neurais artificiais, as de função de base radial apresentam uma boa característica de

aproximação de funções com uma estrutura computacional compacta e exatidões apropriadas a vários contextos de aplicações práticas.

As redes neurais artificiais com função de base radial são efetivamente utilizadas na classificação de padrões (Babu, 2013), em aproximações de funções (Haykin, 2007); (Shi e Shan, 2010), em previsões de séries temporais (Sun et. al., 2005; Kebriaei, et, al., 2011), sistemas de controle (Jafarnejadsani et. al, 2013) e em outras aplicações.

Uma rede neural RBF em sua forma mais básica possui apenas uma única camada oculta, enquanto que redes do tipo MLP (*Multi Layer Perceptron*) apresentam uma ou mais camadas ocultas segundo suas especificações (Haykin, 2007). A camada oculta de uma rede RBF é constituída por funções de ativação de base radial, onde as mais usuais são as gaussianas que são definidas por seus valores de centros e larguras. A camada de saída da rede possui pesos que ponderam as informações de saídas de cada função de base radial, agregando-as linearmente na saída da rede (pode haver mais de uma saída conforme a aplicação).

Existem algumas estratégias utilizadas para ajustar os parâmetros (valores de centros, desvios padrões e pesos) das redes RBF, sendo classificadas como empíricas, auto-organizadas (ou não supervisionadas) e supervisionadas.

1.1 Motivação e Objetivos

Neste trabalho será apresentada uma abordagem para parametrização de redes RBF baseada na combinação de procedimentos não supervisionados e de uma nova proposição de escalonamento de parâmetros (particularmente o ajuste de valores das larguras das funções de base radial). A metodologia consiste em combinar os procedimentos considerados mais efetivos com o objetivo de obter modelos de redes RBF com melhores exatidões e utilizando-se de métodos não supervisionados. Alguns exemplos serão utilizados para ilustrar o emprego da abordagem proposta, e também servirão para realizar comparações de resultados com os principais procedimentos não supervisionados referenciados em textos da área.

Como justificativa para o tema pesquisado nesta tese, notou-se que não existem nas referências bibliográficas consultadas, muitos textos que abordam a combinação

efetiva de técnicas não supervisionadas para os ajustes de parâmetros de redes RBF, no qual se vislumbrou uma possibilidade de desenvolvimento neste contexto.

1.2 Estrutura do Trabalho

O texto deste trabalho está estruturado conforme descrito a seguir. O Capítulo 2 contém uma revisão bibliográfica dos textos consultados no desenvolvimento da tese. O Capítulo 3 traz uma revisão dos conceitos básicos que foram utilizados neste trabalho. No Capítulo 4 será apresentada a metodologia proposta nesta tese. O Capítulo 5 mostra os exemplos utilizados para ilustrar a aplicação da metodologia, os resultados obtidos e comparações com outros procedimentos citados na bibliografia. No Capítulo 6 encontram-se as considerações finais, as contribuições resultantes da pesquisa realizada, e sugestões para trabalhos futuros.

CAPÍTULO 2

2. Revisão Bibliográfica

Redes neurais artificiais são modelos matemáticos que, inspirados na estrutura de sistemas nervosos biológicos, são compostos de unidades de processamento básicas (neurônios artificiais) interligadas por um determinado número de conexões (Braga, Carvalho e Ludemir, 2000).

O interesse específico por redes RBF é proveniente do fato delas possuírem fundamentação estatística, serem consideradas aproximadores universais de funções (Chen *et al.*, 1991), realizarem um mapeamento não linear efetivo entre as informações de entrada e saída de um determinado sistema, e também apresentarem boa capacidade de generalização para dados distintos. Estas características são úteis para aplicações em áreas como: processamento digital de sinais; previsão de séries temporais (Hutchinson, 1994; Kuan, 1995); modelagem de processos complexos; sistemas de controle; e outras.

A arquitetura básica de uma rede RBF é ilustrada na Figura 2.1, onde sua modelagem é descrita por uma soma ponderada de M funções de base radial $h_j(x)$ do vetor de entradas $x = [x_1 \dots x_n]^T$. A ponderação determinada pelo vetor de pesos sinápticos $w = [w_1 \dots w_n]^T$ define o valor de saída da rede e é dada pelo produto interno $\hat{y} = w^T h(x)$, onde $h(x) = [h_1(x) \dots h_M(x)]^T$.

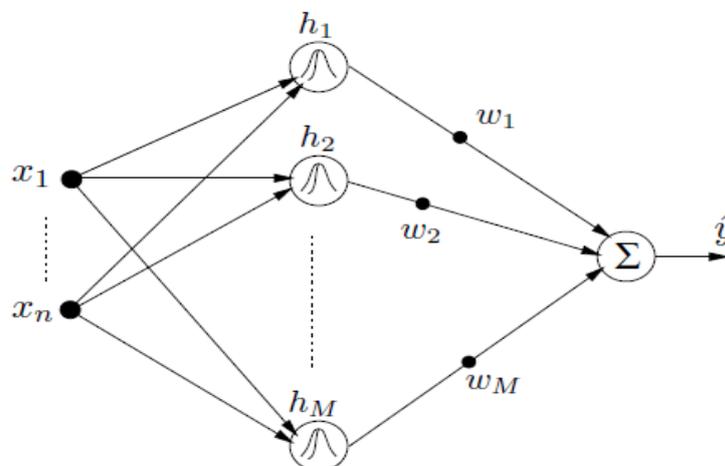


Figura 2.1 - Estrutura típica de uma rede neural RBF com múltiplas entradas e uma saída.

A forma mais usual de função de base radial é a Gaussiana, ou seja:

$$h_j(x) = e^{\left(-\frac{(x-c_{kj})^T \Sigma_j^{-1} (x-c_{kj})}{\sigma_{kj}}\right)}$$

onde $C = [C_{1j} \dots C_{nj}]^T$ é o vetor com as coordenadas do centro da i -ésima função e Σ_j é uma matriz $n \times n$ cujos autovalores representam as variâncias da i -ésima função ao longo das suas direções características (direções dos respectivos autovetores). Se Σ_j for diagonal, então os elementos da diagonal representam as variâncias (σ_{kj}) da função ao longo de cada uma das n direções axiais.

As redes RBF podem ter seus parâmetros ajustados de forma auto-organizada (Broomhead, 1988; Mulgrew, 1996; Bishop, 1997), com métodos dos mínimos quadrados, por exemplo, ou via procedimentos supervisionados (Billings, 1998), por algoritmos de computação evolucionária (Yao, 1999), via técnicas de agrupamentos de dados (Guerra, 2006), entre outras.

Os procedimentos utilizados para ajustar parâmetros de redes RBF são usualmente classificados como empíricos, auto-organizados (ou não supervisionados) e supervisionados. Em Lowe (1989) foi apresentada uma abordagem onde o valor central de cada função de base radial de uma determinada rede era escolhido aleatoriamente a partir do conjunto de dados de treinamento. Este procedimento era justificado no contexto em que as informações utilizadas para treinamento tivessem uma distribuição representativa do problema considerado. Um valor comum para os desvios padrões de todas as funções de base radial era adotado segundo um critério baseado na distância máxima entre os valores dos centros escolhidos. A finalidade era assegurar que as larguras das funções não fossem nem pontiagudas nem planas demais. Os valores dos pesos da camada de saída da rede considerada eram obtidos pelo método da pseudo-inversa. Outros trabalhos Moody e Darken (1989); Saha e Keeler (1989) consideraram a informação da distância Euclidiana entre o valor do centro de cada RBF e sua vizinhança próxima para o objetivo de ajustar os valores das larguras das funções, uma vez que os dados de entrada de uma rede RBF nem sempre possuem uma distribuição uniforme na maioria das aplicações práticas. Uma abordagem para ajustar os valores dos centros das funções de entrada de redes RBF foi apresentada em Chen et al. (1991), onde empregou-se um valor comum para os valores dos desvios padrões das funções, e

um algoritmo de mínimos quadrados ortogonais para ajustar os valores dos centros das RBF.

Como citado anteriormente, alguns procedimentos utilizam técnicas de agrupamento de dados para estabelecer os valores dos centros de cada função de base radial de uma rede RBF, onde o algoritmo K-means é usualmente empregado neste contexto, ou então algoritmos mais elaborados como o de Gustafson-Kessel e outros métodos como mapas auto-organizáveis (Nelles, 2001). No artigo de Benoudjit e Verleysen (2003) foi sugerida a utilização de um fator de escala para ajustar conjuntamente os valores das larguras das funções de uma rede de base radial, possibilitando uma melhor capacidade de aproximação da rede resultante. O trabalho de Sun et. al. (2005) utilizou um método otimizado de partição de dados baseado na distância entre os valores dos centros dos agrupamentos dos dados de treinamento da rede, onde uma função de custo adicional foi associada ao método de otimização empregado com a finalidade de ajustar os valores dos centros e das larguras das funções da rede considerada.

Em Chen et al. (2008) os valores dos centros das redes de base radial foram determinados através de um método de agrupamento subtrativo e otimizados por um algoritmo de enxame de partículas. Noman et al. (2009) utilizaram um procedimento híbrido para ajustar parâmetros de redes RBF. A abordagem proposta inicialmente utilizava um procedimento não supervisionado que empregava um algoritmo K-means de agrupamento para processar os dados de treinamento e obter os valores dos centros e larguras das funções de base radial da rede. Os valores dos pesos da camada de saída eram obtidos pelo método dos mínimos quadrados. Posteriormente empregou-se um algoritmo de enxame de partículas como procedimento supervisionado para otimizar os parâmetros obtidos anteriormente. A abordagem proposta foi aplicada na classificação de padrões de cinco bases de dados distintas e os resultados obtidos foram comparados com aqueles resultantes de técnicas de ajuste de parâmetros para redes RBF via método do gradiente descendente.

Em Qing-wei et al. (2010) e Pan et al. (2011) foram utilizados algoritmos genéticos para determinar todos os parâmetros de uma rede RBF. Estratégias supervisionadas foram detalhadas em Mahdi e Rouchka (2011) e Su et. al. (2012). Os procedimentos utilizam métodos como o do gradiente descendente para ajustar os parâmetros (centros, desvios padrões e pesos) de uma determinada rede RBF, onde a

informação do erro relativo à informação de saída da rede em questão e o padrão desejado para a mesma é utilizada no processo de treinamento.

Kalyvas (2001) utilizou um algoritmo genético para definir a quantidade de neurônios da camada de entrada de redes RBF. A quantidade de neurônios nesta camada estava relacionada com o tamanho da janela associada com o conjunto de dados de entrada. Na proposta de Fritzke (1995), o procedimento de treinamento se inicia com uma rede com dois neurônios de entrada e acrescenta novos neurônios a cada ciclo de avaliação. Cada neurônio possui um parâmetro de avaliação que é calculado com base nos padrões dos dados de entrada. Um novo neurônio é inserido na vizinhança do neurônio com menor avaliação, fazendo com que a estrutura resultante contemple todas as áreas do espaço dos dados avaliados. Eventualmente, neurônios também podem ser excluídos da rede. Estas características permitem que a rede cresça até que um critério de desempenho seja atingido, como a quantidade máxima de neurônios ou um percentual de padrões classificados corretamente.

Os procedimentos para ajuste de parâmetros de redes RBF são classificados usualmente como não supervisionados ou supervisionados. Geralmente os procedimentos não supervisionados são mais simples de se aplicar, pois utilizam algoritmos não recursivos (Haykin, 2007). Já os modelos de redes RBF resultantes de procedimentos supervisionados tendem a apresentar melhores exatidões, mas exigem algoritmos recursivos no ajuste dos parâmetros associados (Haykin, 2007).

A seguir serão descritas as abordagens mais frequentemente utilizadas nos procedimentos não supervisionados.

I- Valores dos Centros Seleccionados Aleatoriamente:

- a) O valor central de cada função de base radial de uma determinada rede é escolhido aleatoriamente a partir do conjunto de dados de treinamento. A justificativa deste procedimento é explicada em Lowe (1989).
- b) O valor da largura (desvio padrão) de cada função de base radial da rede é dado pela expressão (2.1), onde N_c é o número de funções utilizadas na camada escondida e D_{max} é a distância máxima entre os pares de valores escolhidos dos centros das funções.

$$\sigma = \frac{D_{max}}{\sqrt{2N_c}} \quad (2.1)$$

Um valor de largura comum ($\sigma_{kj} = \sigma$) a todas as funções da rede pode ser empregado quando os dados de entrada possuem uma distribuição uniforme, condição não usual na maioria das aplicações práticas. Um procedimento mais factível é expresso por (2.2), onde c_{ij} constitui os “ p ” valores da vizinhança do correspondente centro c_{kj} associado à função radial em questão, e um valor sugerido seria $p = 2$ (Moody e Darken, 1989).

$$\sigma_{kj} = \frac{1}{p} \sqrt{\sum_{m=1}^p \|c_{kj} - c_{ij}\|^2} \quad (2.2)$$

Outra abordagem (Saha e Keeler, 1989) utiliza o dado relacionado com a distância Euclidiana entre o valor do centro de cada RBF e sua vizinhança próxima, cuja informação é multiplicada por uma constante “ K ” (2.3) com o objetivo de estabelecer uma sobreposição adequada entre funções adjacentes da rede em questão, cuja finalidade é melhorar a capacidade de interpolação do modelo resultante.

$$\sigma_{kj} = K \cdot \min(\|c_{kj} - c_{ij}\|) \quad (2.3)$$

No artigo de Benoudjit e Verleysen (2003) sugeriu-se a utilização de um fator de escala “ K_e ”, como o expresso em (2.4), para ajustar conjuntamente os valores dos desvios das RBF, cujos valores podem ser estimados por uma das três expressões citadas anteriores, possibilitando uma melhor capacidade de aproximação da rede resultante.

$$\sigma_{kj} = K_e \cdot \sigma_{kj} \quad (2.4)$$

- c) Os valores dos pesos da camada de saída da rede podem ser calculados pelo método da pseudo-inversa (2.5), onde G^+ constitui os valores correspondentes da matriz associada aos dados $G_j(x)$ das funções RBF, e Y_d corresponde aos valores do vetor de resposta desejado (relacionado com o conjunto de dados de treinamento da rede). Quando não houver problemas de inversão de matrizes, o método convencional dos mínimos quadrados é empregado.

$$w_j = G^+ \cdot Y_d \quad (2.5)$$

II- Valores de Centros Obtidos via Mínimos Quadrados Ortogonais:

Uma abordagem para ajustar os valores dos centros das funções de entrada de redes RBF foi apresentada em Chen et al. (1991), onde empregou-se um valor comum para os valores dos desvios padrões das funções e um algoritmo de mínimos quadrados ortogonais para ajustar os valores dos centros das RBF. Devido à complexidade computacional dos algoritmos ortogonais em relação à dimensão dos regressores empregados, a estratégia de ajuste é factível para um determinado limite de dados de treinamento. Os valores dos pesos da camada de saída da rede associada também podem ser estimados por métodos dos mínimos quadrados.

III- Valores dos Centros Determinados por Métodos de Agrupamento de Dados:

- a) O valor central de cada função de base radial é determinado por métodos de agrupamento de dados, que processam as informações de treinamento dos dados de entrada da rede e fornecem os valores dos centros dos *clusters* associados. O algoritmo K-means é usualmente empregado no contexto de redes RBF (Nelles, 2001), e alternativamente outros métodos mais elaborados podem ser aplicados, por exemplo, via mapa auto-organizável (HAYKIN, 2007).
- b) Os valores das larguras das RBF podem ser ajustados por intermédio das informações das distâncias entre os valores relacionados com os centros das funções e os dados das vizinhanças associadas, similarmente as formas citadas no item (b) do procedimento I. Outras abordagens fornecem os valores dos desvios padrões das RBF associados com as informações de matrizes de covariância resultantes de algoritmos elípticos de agrupamento de dados, como o algoritmo de Gustafson-Kessel (Nelles, 2001), por exemplo, onde este tipo de abordagem tende a ser computacionalmente preterida devido à complexidade dos algoritmos envolvidos.
- c) Os valores dos pesos da camada escondida da rede são calculados através de métodos dos mínimos quadrados.

IV- Valores dos Centros Baseados em Grids:

Uma das maiores dificuldades das abordagens que utilizam métodos de agrupamento de dados para a determinação dos valores dos centros das RBF diz respeito à obtenção dos valores das larguras das funções em questão. Os procedimentos que utilizam informações das vizinhanças dos dados de treinamento geralmente não evitam lacunas de informações em determinadas regiões do espaço dos dados, necessitando de uma “sintonia fina” das funções da camada escondida que podem ocasionar uma deficiência na capacidade de interpolação da RBF resultante. Para aplicações onde o mapeamento dos dados de interesse não constitui um problema de dimensão elevada, uma abordagem alternativa seria os procedimentos baseados em grids (Nelles, 2001), que possibilitam determinar os valores das larguras das funções de uma rede em relação à resolução dos grids associados aos dados de treinamento. Essa abordagem possibilita uma sobreposição regular entre as funções da rede resultante, melhorando a capacidade de aproximação da mesma. O problema deste procedimento está relacionado com a dimensionalidade dos dados da aplicação considerada.

A seguir serão descritas as abordagens mais frequentemente utilizadas nos procedimentos supervisionados.

I- Valores dos Parâmetros da Rede Ajustados por Métodos do Gradiente:

Vários procedimentos supervisionados para ajuste de parâmetros de redes RBF são baseados em métodos do gradiente descendente. O processo de treinamento típico utiliza a informação do erro (e_i) ou da sua somatória quadrática (2.6), entre os valores fornecidos para o padrão de treinamento/teste (Y_d) e os dados de saída (Y_e) estimados pela rede em questão.

$$e_i = Y_d - Y_e ; \quad \varepsilon = \frac{1}{2} \sum_{i=1}^N (e_i)^2 \quad (2.6)$$

As expressões a seguir (Haykin, 2007) são empregadas para ajustar os parâmetros de redes RBF, onde η_1 , η_2 e η_3 são os fatores de aprendizado das equações em questão.

$$c_{kj}(n+1) = c_{kj}(n) - \eta_1 \frac{\partial \varepsilon(n)}{\partial c_{kj}(n)}$$

$$\sigma_{kj}(n+1) = \sigma_{kj}(n) - \eta_2 \frac{\partial \varepsilon(n)}{\partial \sigma_{kj}(n)}$$

$$w_j(n+1) = w_j(n) - \eta_3 \frac{\partial \varepsilon(n)}{\partial w_j(n)}$$

II- Valores dos Parâmetros Ajustados por Outros Métodos de Otimização:

Além dos métodos clássicos de otimização não lineares aplicados no ajuste dos parâmetros de redes RBF, trabalhos mais recentes utilizam técnicas como algoritmos genéticos (Ding et al., 2012), enxame de partículas (Fathi; Montazer, 2013) e outras neste contexto.

Estes procedimentos geralmente resultam em ajustes bem otimizados dos parâmetros considerados, mas estas abordagens devem ser empregadas de forma adequada para evitar problemas de mínimos locais, além dos tempos de processamento inerentes aos algoritmos associados.

Com o objetivo de indicar as diferenças típicas entre uma rede RBF e uma rede MLP, seja a descrição básica desta última que por simplicidade será representada com uma única camada intermediária e uma única saída. A saída do i -ésimo neurônio na camada intermediária da MLP considerada é dada por

$$y_i = g_i \left(\sum_{j=1}^n w_{ij} x_j + b_i \right),$$

onde w_{ij} é o peso sináptico da conexão entre o neurônio e a j -ésima entrada x_j , b_i é um termo de polarização (*bias*) e g_i é a função de ativação do neurônio. A função g_i deve realizar um limiar de ativação, comportamento que pode ser modelado, por exemplo, por uma função sigmoide (sendo a_i a inclinação da função sigmoídal correspondente):

$$g_i(v) = \frac{1}{1 + e^{(-a_i v)}}$$

O cálculo da saída da rede é análogo àquele dos demais neurônios, porém as informações das entradas para o neurônio de saída são dadas pelas saídas dos neurônios da camada intermediária. Logo, a saída da rede pode ser escrita como

$$\hat{y} = g_s \left(\sum_{i=1}^M w_i g_i \left(\sum_{j=1}^n w_{ij} x_j + b_i \right) + b_s \right),$$

onde w_i é o peso sináptico entre o i -ésimo neurônio da camada intermediária e o neurônio de saída, e b_s é o valor *bias* da função de ativação g_s correspondente. Em problemas de regressão, como é o caso da modelagem de sistemas dinâmicos, o neurônio de saída da rede é usualmente linear, mais especificamente, $g_s(v) = v$.

Nota-se que as redes MLP diferem das RBF principalmente no que diz respeito ao tipo de unidade elementar (neurônio) utilizada. Enquanto os neurônios das redes RBF são equipados com funções de ativação de base radial, as MLP possuem neurônios com funções de ativação diversas, por exemplo, tipo sigmoideal. Além disso, as redes RBF possuem apenas uma camada escondida e não há ponderações (pesos) multiplicando as informações de entrada da rede.

Diferentes estratégias para treinamento de ANN são conhecidas (Braga et al., 2000; Haykin, 2007), incluindo abordagens como otimização multi-objetivo ou via máquinas de vetores de suporte (SVMs - Support Vector Machines), entre outras Braga et al., (2000). Um estudo comparativo de diferentes algoritmos de treinamento supervisionado de ANN foi realizado em De Castro, (1998). Um algoritmo de treinamento supervisionado não usual, baseado em um operador de extensão, foi proposto em Pedroza e Pedreira, (2000). Uma referência clássica sobre a utilização de ANN na modelagem de sistemas dinâmicos é a de Narendra e Parthasarathy, (1990).

CAPÍTULO 3

3. Revisão de Conceitos

A modelagem de sistemas dinâmicos é de grande importância em vários campos das ciências e engenharias (Aguirre, 2007). Os métodos clássicos de modelagem de sistemas são baseados em modelos matemáticos acurados, entretanto, para sistemas complexos com características não lineares e/ou parâmetros variáveis no tempo, a obtenção dos modelos correspondentes não é trivial. Para sistemas com características lineares existem métodos clássicos que apresentam estruturas e procedimentos de identificação bem conhecidos. Para sistemas com características não lineares existem diferentes abordagens segundo as aplicações e os métodos utilizados. Entre os modelos não lineares conhecidos têm-se as estruturas NARX, NARMAX, redes neurais artificiais, sistemas *fuzzy*, entre outros.

Previsões de séries temporais também são questões importantes em várias áreas das ciências e engenharias. Uma série temporal compreende conjuntos de valores passíveis de ordenação cronológica (Milone e Angelini, 1995). O estudo de séries temporais trabalha com o comportamento de dados compreendidos em intervalos de tempo passados e atuais, de modo a se extrair possíveis variações dos dados em um determinado horizonte de tempo futuro, o principal objetivo de previsões de séries temporais (Corrêa et al., 2000; Serrão, 2003; Nogueira, 2004; Sáfadi e Nogueira, 2004; Topalli e Erkmen, 2013).

3.1 Modelagem de Sistemas Dinâmicos

Nas análises e estudos de sistemas complexos são necessárias representações adequadas que permitam modelar características relevantes dos mesmos (Zambroni e Pinheiro, 2008). A modelagem matemática de sistemas dinâmicos é realizada por meio de equações que representam suas dinâmicas com um determinado grau de exatidão estabelecido. A complexidade que os sistemas reais podem apresentar na prática, ou o

desconhecimento de todos os detalhes constituintes dos mesmos, leva à necessidade de se trabalhar com aproximações. Dependendo da exatidão requerida existem vários modelos possíveis de serem utilizados. Geralmente, busca-se um compromisso entre a simplicidade do modelo matemático adotado e a exatidão resultante.

Os modelos de sistemas dinâmicos podem ser classificados em duas classes:

Sistemas Contínuos – São aqueles cujas variáveis apresentam alterações contínuas no tempo. Os sistemas dinâmicos geralmente possuem grandezas que variam continuamente no tempo, e quando apresentam comportamentos lineares são usualmente modelados por meio de equações diferenciais, funções de transferências ou variáveis de estado contínuas;

Sistemas Discretos – Como as técnicas modernas de medição e registro de informações são realizadas atualmente com equipamentos digitais, as variáveis de entrada e saída mensuradas em sistemas práticos estão em formato discreto no tempo. Para sistemas com comportamentos lineares os modelos resultantes são frequentemente representados por equações a diferenças, com funções de transferências amostradas, ou por variáveis de estado discretas. A equação (3.1) ilustra uma equação a diferenças genérica, onde T representa o tempo de amostragem das variáveis de entrada e saída do modelo, ou seja, $t = kT$ com $k = [0, 1, 2, 3, 4, \dots]$. Por questões de simplificações, o tempo de amostragem pode ser omitido, como em (3.2).

$$y(kT) = -a_1y((k-1)T) - \dots - a_ny((k-n)T) + b_0u(kT) + b_1u((k-1)T) + \dots + b_mu((k-m)T) \quad (3.1)$$

$$y(k) = -a_1y(k-1) - \dots - a_ny(k-n) + b_0u(k) + b_1u(k-1) + \dots + b_mu(k-m) \quad (3.2)$$

Vários modelos podem ser expressos matematicamente na forma conhecida como modelo de regressão, muito utilizado em várias áreas da ciência e engenharia. O conjunto de dados (φ) usado para obter a equação de um determinado modelo, definido pelos seus coeficientes (θ), é chamado de regressor. Esse conjunto é constituído por pares de medidas de entrada e saída do sistema em questão. Em relação à equação (3.2) esse conjunto é constituído de pares de valores como $\varphi = [y(k-1)u(k-1)y(k-2)u(k-2)y(k-3)u(k-3) \dots]$. O vetor (ou matriz) de coeficientes são os

parâmetros do modelo $\theta = [a_1 \ a_2 \dots a_n \ b_0 \ b_1 \dots b_m]^T$, onde na notação utilizada o símbolo “ \cdot ” denota uma matriz ou um vetor transposto e “ $^{-1}$ ” simboliza matriz inversa. Os valores dos coeficientes de um determinado modelo podem ser obtidos pelo método dos mínimos quadrados (3.3) a partir de valores de informações de entrada e saída do sistema considerado.

$$\theta = [\phi^T \phi]^{-1} \phi^T Y \quad (3.3)$$

3.1.1 Representação Matemática de Modelos Lineares e Não-Lineares

Dependendo da estrutura do modelo utilizado, os mesmos podem ser classificados como: Modelo auto-regressivo com entradas externas (*ARX-Autoregressive with Exogenous Inputs*); Modelo auto-regressivo com média móvel e entradas externas (*ARMAX-Autoregressive Moving Average with Exogenous Inputs*); Modelo *Box-Jenkins*; e outros. Existem outros tipos de modelos adequados a sistemas não lineares como NARX, NARMAX, entre outros. Sistemas dinâmicos também podem ser representados por modelos de redes neurais artificiais ou por conjuntos de regras difusas que utilizem conjuntos nebulosos (*Fuzzy Sets*).

3.1.1.1 Modelos ARX/ARMAX

Considere o seguinte modelo geral abaixo.

$$A(q)y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}v(k) \quad (3.4)$$

As entradas e as saídas dos sistemas modelados são representadas por $u(k)$ e $y(k)$, respectivamente. A variável $v(k)$ representa erros de medidas, imprecisões e ruídos que podem estar agregados a medições práticas, e o operador q representa as amostragens das informações do sistema decorrentes do intervalo de tempo entre amostras consecutivas. Os coeficientes dos modelos são expressos pelos polinômios a seguir, que conforme os valores dos coeficientes associados definem os modelos conhecidos como ARX, ARMAX, *Box-Jenkins*, e outros definidos pela estrutura citada.

$$\begin{aligned}
A(q) &= 1 + a_1 q^{-1} + \dots + a_{ny} q^{-ny} ; \\
B(q) &= b_1 q^{-1} + \dots + b_m q^{-nu} ; \\
C(q) &= 1 + c_1 q^{-1} + \dots + C_{nc} q^{-nc} ; \\
D(q) &= 1 + d_1 q^{-1} + \dots + d_{nd} q^{-nd} ; \\
F(q) &= 1 + f_1 q^{-1} + \dots + f_{nf} q^{-nf} .
\end{aligned}$$

3.1.1.2 Modelos NARX

Os modelos NARX (*Nonlinear AutoRegressive with eXogenous inputs*) são descritos pela equação (3.5), onde C_i são os coeficientes da expressão. Este tipo de modelo é adequado para a modelagem de sistemas dinâmicos com comportamentos não lineares (Aguirre, 2007).

$$y(k) = \sum_i C_i \prod_{j=1}^{Ny} y(k-j) \prod_{r=1}^{Nu} u(k-r) \prod_{v=0}^{Nu} e(k-v) \quad (3.5)$$

3.1.1.3 Modelos NARMAX

Os modelos NARMAX (*Nonlinear AutoRegressive Moving Average with eXogenous inputs*) são descritos pela equação (3.6), onde $C_{p,m-p}$ são os coeficientes da expressão. Este tipo de modelo também é adequado para a modelagem de sistemas dinâmicos com comportamentos não lineares.

$$y(k) = \sum_{m=0}^l \sum_{p=0}^m \sum_{n_1, n_m}^{n_y, n_u} C_{p,m-p}(n_1, \dots, n_m) \prod_{i=1}^p y(k-n_i) \prod_{i=p+1}^m u(k-n_i) \quad (3.6)$$

3.1.1.4 Modelos Fuzzy

Modelos *fuzzy* (Pedrycz e Gomide, 1998) são representações desenvolvidas com a utilização de conjuntos difusos (*fuzzy sets*) propostos por Lotfi Zadeh (1965). Esses modelos podem ser subdivididos em três classes principais: modelos relacionais; modelos linguísticos ou de Mamdani e modelos funcionais ou de Takagi-Sugeno (TS). Dentre essas três classes, os modelos TS são aqueles que apresentam as propriedades mais favoráveis para a modelagem de sistemas dinâmicos complexos (Babuska, 1998).

Os modelos *fuzzy* do tipo TS são constituídos por um conjunto de regras expressas por (3.7).

$$R^i: IF x_1 \text{ é } X_1^i E \dots E x_n \text{ é } X_n^i THEN y = f_i(x_1, \dots, x_n) \quad (3.7)$$

Onde R^i denota a i -ésima regra, $x_j \in X_j \subset \mathfrak{R}$ ($j = 1, \dots, n$) são as variáveis de entrada, $y \in Y \subset \mathfrak{R}$ é a variável de saída, f_i ($i = 1, \dots, M$) são funções que relacionam as entradas do modelo com sua saída, e X_j^i são conjuntos *fuzzy* definidos sobre os universos de discurso X_j das respectivas variáveis, ou seja, $X_j^i : X_j \rightarrow [0,1]$. O procedimento que converte os dados das variáveis de entrada no respectivo grau de pertinência correspondente é conhecido como fuzificação de dados.

A inferência de um valor de saída y a partir de um conjunto específico de valores de entrada é calculada como a média ponderada das saídas individuais (f_i) de cada regra e o grau de pertinência (w_i) correspondente a cada conjunto difuso associado, e este procedimento é conhecido como defuzificação de dados (3.8).

$$y = \frac{\sum_{i=1}^M w_i f_i(x_1, \dots, x_n)}{\sum_{i=1}^M w_i} \quad (3.8)$$

Em sua configuração mais usual, a saída individual de cada regra é dada por uma relação linear ou afim de todas as variáveis do modelo, sendo expressa por (3.9).

$$f_i(x_1, \dots, x_n) = \theta_0^i + \sum_{j=1}^n \theta_j^i x_j \quad (3.9)$$

Essa estrutura permite uma interpretação matemática simples do modelo como uma interpolação de diferentes modelos locais afins e implica que a saída em (3.9) é linear nos parâmetros $\theta_0^i, \dots, \theta_n^i$. Logo, esses parâmetros podem ser obtidos numericamente utilizando algoritmos de estimação linear, desde que todos os conjuntos *fuzzy* tenham sido determinados a priori a partir de alguma heurística ou algoritmo não supervisionado (Babuska, 1998). Caso contrário, algoritmos de programação não linear

ou abordagens evolutivas de otimização devem ser utilizados para a determinação global do modelo (Delgado, 2002).

Um exemplo de um modelo TS com uma única variável de entrada e três regras está ilustrado na Figura 3.1. As regras (r_i) do modelo são expressas por (3.10).

$$\begin{aligned} r_1: & \text{IF } x \text{ é Pequena THEN } y_1 = b_1 + a_1x \\ r_2: & \text{IF } x \text{ é Média THEN } y_2 = b_2 + a_2x \\ r_3: & \text{IF } x \text{ é Grande THEN } y_3 = b_3 + a_3x \end{aligned} \quad (3.10)$$

Sendo as etiquetas “Pequena”, “Média” e “Grande” rótulos linguísticos associados a conjuntos *fuzzy* definidos sobre o universo de discurso da variável de entrada, conforme ilustrado na Figura 3.1. Esses conjuntos determinam, através de (3.10), o nível (e consequentemente a região) de ativação de cada regra, isto é, o peso de cada modelo local no cálculo ponderado da saída.

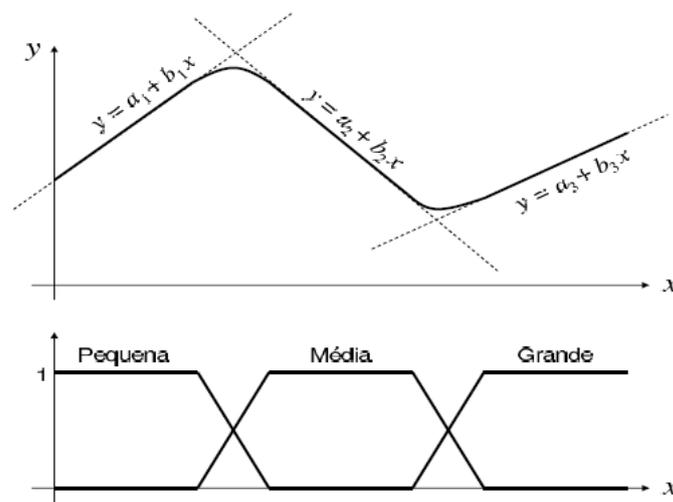


Figura 3.1 - Exemplo de modelo TS com três regras (modelos locais afins).

Modelos TS dinâmicos possuem exatamente a mesma formulação descrita anteriormente, exceto que os modelos locais são dinâmicos ao invés de estáticos. Em princípio, os modelos locais podem possuir qualquer estrutura dinâmica. Por exemplo, para modelos locais lineares, basta redefinir as variáveis das premissas como sendo $x_1 = y(k - 1), \dots, x_n = y(k - n)$.

3.1.1.5 Algoritmo de Agrupamento Fuzzy C-Means

Entre os algoritmos de agrupamento utilizados, o algoritmo *Fuzzy C-Means* (Bezdek, 1980) apresenta uma boa relação de custo/benefício entre a exatidão resultante na obtenção dos clusters associados e a demanda computacional, em relação a outros algoritmos usualmente utilizados na prática. O conceito básico deste algoritmo em relação a um conjunto de dados e um determinado agrupamento correspondente pode ser descrito como: quanto mais próximo do centro de um agrupamento um determinado dado estiver, maior será seu grau de pertinência a esse agrupamento. Considerando um grupo de dados correspondente as informações de treinamento $X = x_1, \dots, x_n$, onde cada ponto x_k ($k = 1, \dots, p$) é um vetor em R^p e considerando U_{cn} como um grupo de matrizes reais $c \times n$, sendo c é um inteiro que obedece a relação $2 \leq c < n$, define-se a partição *fuzzy* para o conjunto X segundo a equação (3.11).

$$M_{fnc} = U \in U_{cn}: u_{ik} \in [0, 1], \quad \sum_{i=1}^c u_{ik} = 1 < \sum_{k=1}^n u_{ik} < n \quad (3.11)$$

Sendo U_{ik} o grau de pertinência de x_k em um agrupamento i ($i = 1, \dots, c$). A equação (3.12) pode ser utilizada para determinar se o algoritmo *Fuzzy C-Means* encontrou uma partição ótima.

$$J_m(U, V; X) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m \|x_k - v_i\|^2 \quad (3.12)$$

Da equação (3.12), tem-se que $V = (v_1, \dots, v_c)$ é a matriz que contém os centros dos agrupamentos, onde todo $v_i \in R^p$ é um peso constante entre $[1, \infty)$. O centro $v_i^{(t)}$ de cada agrupamento i ($i = 1, \dots, c$) para uma interação t pode ser calculada segundo a equação (3.13).

$$v_i^{(t)} = \frac{\sum_{k=1}^n (u_{ik}^t)^m x_k}{\sum_{k=1}^n (u_{ik}^t)^m} \quad (3.13)$$

De posse dos novos centros $v_i^{(t)}$, os valores de pertinência podem ser obtidos pela equação (3.14).

$$u_{ik}^{(t+1)} = \frac{1}{\sum_{j=1}^c \left(\frac{\|x_k - v_i^{(t)}\|^2}{\|x_k - v_j^{(t)}\|^2} \right)^{\frac{2}{m-1}}} \quad (3.14)$$

De forma simplificada um algoritmo *Fuzzy C-Means* pode ser descrito conforme segue:

1. Escolher valores para c , m e ε ;
2. Gerar a partição fuzzy u^o de forma aleatória obedecendo às restrições impostas;
3. Atribuir um valor nulo ao contador de interações;
4. Fazer $j_m^{(t)} = 0$;
5. Calcular os centros $v_i^{(t)}$ segundo a equação correspondente;
6. Avaliar a função objetivo $j_m^{(t+1)}$;
7. Determinar os graus de pertinência $u_{ik}^{(t+1)}$;
8. Calcular $\delta = j_m^{(t+1)} - j_m^{(t)}$;
9. Incrementar o contador de interações;
10. Se a condição de saída for falsa, então se retorna ao quinto passo do algoritmo. Senão, finaliza-se o procedimento. A condição de parada do algoritmo é atingida quando $\delta \leq \varepsilon$, ou quando um determinado número de interações é alcançado.

3.1.1.6 Modelos Neurais

A ideia básica para a utilização de ANN para aproximações de funções complexas, modelagem de sistemas dinâmicos não lineares, e previsões de séries temporais, por exemplo, é fundamentalmente a mesma daquela apresentada para os modelos *fuzzy*. Tanto os modelos neurais como os *fuzzy* podem ser classificados como aproximadores universais de funções, isto é, modelos capazes de aproximar com exatidão arbitrária qualquer mapeamento contínuo definido sobre um domínio compacto (Kosko, 1992). Nesse tipo de abordagem, a parte dinâmica fica então externa ao modelo que é expressa pelo vetor de regressores φ , cujos elementos serão as entradas para a rede em questão. É possível mostrar que as redes RBF são equivalentes, exceto por um fator de normalização, a modelos *fuzzy* TS que possuam conjuntos *fuzzy* modelados por

funções Gaussianas e modelos locais constantes, i.e., $f_i(x_1, \dots, x_n) = \theta_0^i$ ao invés de relações a afins.

3.1.2 Estimação de Parâmetros – Método dos Mínimos Quadrados

Na determinação dos coeficientes/parâmetros de modelos, frequentemente empregam-se procedimentos baseado no método dos mínimos quadrados. O conceito está associado em se minimizar a soma dos quadrados das diferenças (resíduos) entre os valores reais e os estimados de um determinado modelo.

Para um sistema representado na forma de tabela com N medidas, cujos dados são valores dos conjuntos de entradas/saída anteriores φ (os chamados regressores), deseja-se determinar os coeficientes θ de uma equação que represente o sistema de maneira mais adequada possível. O erro ξ da aproximação obtida é minimizado conforme um critério quadrático J . As equações (3.15) expressam a conceituação.

$$Y(l) = \varphi(m)^T \theta + \xi(m), \quad (3.15)$$

$$J = \sum_{i=1}^N \xi(i)^2$$

Denotando F o vetor transposto dos regressores, E o vetor de erro, e y os valores reais do sistema considerado, tem-se a expressão de erro $E = y - F\theta$. Dessa forma, a função de custo pode ser escrita como $J = [y - F\theta]^T [y - F\theta]$. Desenvolvendo, obtém-se $J = y^T y - 2\theta^T F^T y + \theta^T F^T F \theta$. O vetor de coeficientes que minimiza a função de custo é obtido derivando J em relação a θ e igualando a expressão obtida a zero, resultando em $-2F^T y + 2F^T F \theta = 0$. Assim, os valores estimados dos coeficientes da equação de modelagem são dados por (3.16).

$$\theta = [F^T F]^{-1} F^T y. \quad (3.16)$$

3.2 Previsão de Séries Temporais

Os métodos de previsão de séries temporais são importantes em vários contextos como: estimação de carga em sistemas elétricos; previsão de precipitação pluviométrica; de valores financeiros, entre outros.

Uma série temporal é definida como um conjunto de observações de uma dada variável geralmente distribuídas de maneira equidistante no tempo, e que possuem como característica central a presença de uma dependência serial entre elas (Ribeiro e Paula, 2000). Uma série temporal é denotada por Z_t , onde $t = \{1, 2, 3, 4, \dots, n\}$ são os intervalos de amostragem com função densidade de probabilidade $p(Z_t)$ para cada t .

A análise de séries temporais parte do pressuposto de que uma determinada série pode ser entendida como a composição de comportamentos de tendência, fatores cíclicos, variações sazonais, além de fatores aleatórios representados de maneira genérica como um ruído branco. A tendência em uma série temporal é a mudança gradual observada por meio da variação dos valores da série ao longo do tempo, e que se mantém ao se remover os componentes de ciclos, sazonalidades e fatores aleatórios. Ciclos e sazonalidades são comportamentos estocásticos que acontecem de maneira recorrente ao longo de um período definido. Morettin e Tolo (2006) conceituam que os comportamentos sazonais são flutuações ocasionadas na série temporal devido à influência de algum fator externo de sazonalidade. Os componentes de ciclo apresentam um comportamento similar, no entanto, normalmente apresentam comprimento maior que os componentes sazonais e não apresentam duração uniforme. Com a remoção das componentes de tendência, ciclos e sazonalidades, têm-se as componentes *residuais* que podem, eventualmente, representar fatores aleatórios para o entendimento do comportamento de uma determinada série.

Previsões podem ser realizadas em diferentes intervalos de tempo, tendo finalidades distintas para cada horizonte (Mohamed, 1998). Os horizontes utilizados são classificados como longo, médio, curto e curtíssimo prazo. A previsão de longo prazo utiliza intervalos anuais. A previsão de médio prazo utiliza intervalos mensais. A previsão de curto prazo é realizada em intervalos semanais. A previsão de curtíssimo prazo ocorre em intervalos inferiores a uma hora.

As técnicas de modelagem de séries temporais buscam representar matematicamente a demanda futura relacionando os dados passados do item em questão

no tempo, ou seja, partem do princípio de que a demanda futura será uma projeção do passado (Tubino, 2000). As abordagens procuram identificar uma tendência nas observações ao longo do tempo (Moreira, 1996), e frequentemente as observações consideradas tem um espaçamento constante (em dias, semanas, meses, anos, entre outros). Um dos pressupostos desses métodos é que os valores projetados podem ser estimados com base nos valores passados.

Na descrição de séries temporais são utilizados modelos estatísticos que podem ser classificados em duas classes conforme o número de parâmetros envolvidos (Morettin e Tolo, 2006):

- Modelos Paramétricos: Os modelos mais comumente usados são AR, ARMA, ARIMA, SARIMA e modelos não lineares.
- Modelos Não Paramétricos: Podem ser modelados por funções de autocovariância ou autocorrelação, modelos de redes neurais, via regras *fuzzy*, e outros, por exemplo.

Em Santos et al., (2012) abordou-se a previsão de séries temporais via paradigma de composição *Ensemble*, uma forma de combinar soluções obtidas por intermédio de múltiplos modelos individuais, como redes neurais artificiais com topologias distintas ou modelos *fuzzy*, por exemplo.

CAPÍTULO 4

4. Metodologia

Os procedimentos não supervisionados para ajustes de parâmetros de redes RBF podem suprir os valores iniciais para procedimentos supervisionados, os quais podem refinar estes valores resultando em melhores exatidões e menores tempos de processamento nas tarefas associadas (Reis, 2014).

Não existem muitos trabalhos recentes (SANTOS et al., 2015) sobre métodos não supervisionados para ajuste de parâmetros de redes RBF. E neste contexto uma abordagem que seja prática e fácil de aplicar e que possa melhorar a exatidão dos modelos das redes resultantes é bem-vinda.

Neste trabalho será apresentada uma abordagem para parametrização de redes RBF baseada na combinação de procedimentos não supervisionados e uma nova proposição de escalonamento de parâmetros. A metodologia consiste em combinar os procedimentos considerados mais efetivos com o objetivo de obter modelos de redes RBF com melhores exatidões e menores demandas computacionais.

A estrutura típica de rede de base radial considerada neste trabalho está ilustrada na Figura 4.1.

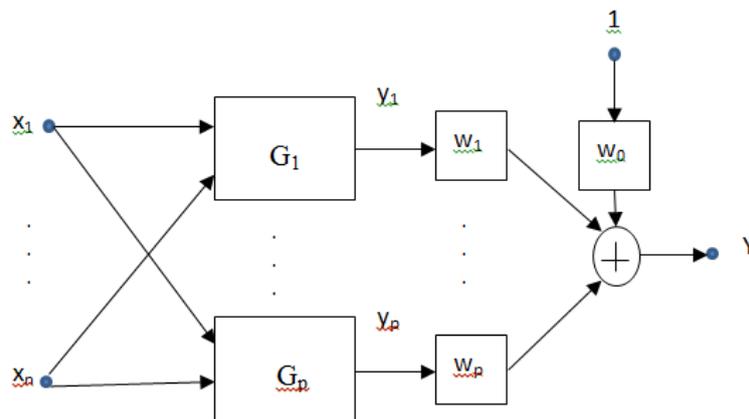


Figura 4.1 - Estrutura típica de uma rede RBF.

A equação (4.1) modela as funções gaussianas adotadas na estrutura da rede considerada, onde c_{kj} representa os valores dos centros das RBF e σ_{kj} os valores das larguras das mesmas.

$$y_j = G_j(\mathbf{x}) = e^{-\left(\frac{(x_1 - c_{1j})^2}{2\sigma_{1j}^2} + \dots + \frac{(x_n - c_{nj})^2}{2\sigma_{nj}^2}\right)} \quad (4.1)$$

Os pesos (w_j) da camada de saída da rede multiplicam os valores (y_j) fornecidos pelas funções de base radial do modelo, que totalizados (somados) fornecem o valor (Y) de saída da rede, que é adicionado a um valor de bias unitário multiplicado pelo peso “ w_0 ” associado. Uma rede RBF pode ter outras saídas associadas a outros pesos correspondentes. A informação de cada saída de uma determinada rede é dada por (4.2).

$$Y = \sum_{i=1}^m w_i G_i(x) \quad (4.2)$$

A abordagem proposta para parametrização de redes RBF está detalhada nos procedimentos descritos a seguir, sendo abreviada por CPNSEP (Combinação de Procedimentos Não Supervisionados e com Escalonamento de Parâmetros).

Abordagem CPNSEP:

- 1- Escolher o número de nós da camada escondida da rede RBF em questão.
- 2- Utilizar um algoritmo de agrupamento de dados como o Fuzzy C-Means, por exemplo, que tende a apresentar melhores resultados em relação aos algoritmos convencionais (como o K-means e o C-means). O algoritmo processa os dados de treinamento da rede com a finalidade de otimizar os agrupamentos correspondentes, cujo o número de clusters é aquele definido no item anterior. As informações dos clusters serão utilizadas para selecionar os parâmetros das funções de ativação e o número de agrupamentos definirá o número de nós da camada escondida da rede.
- 3- Os valores dos centros (c_{kj}) das funções gaussianas da rede serão os valores centrais dos clusters determinados pelo algoritmo de agrupamento.
- 4- Os valores das larguras das funções gaussianas são inicialmente estimados através da fórmula (4.3) clássica que servirá para calcular os valores dos desvios padrões (σ_{kj}^*). Neste caso, a partir dos dados (\mathbf{x}_j) de cada variável relacionada com um determinado cluster identificado pelo algoritmo de agrupamento.

$$\sigma_{kj}^* = \sqrt{\frac{\sum_{j=1}^m x_j^2 - \frac{1}{m} (\sum_{j=1}^m x_j)^2}{m - 1}} \quad (4.3)$$

- 5- Os valores obtidos no item anterior são escalonados através da expressão (4.4), objetivando-se obter larguras adequadas para uma boa capacidade de interpolação da rede resultante. Este procedimento de escalonamento é originalmente proposto neste trabalho e não se encontrou nenhuma citação neste sentido em trabalhos anteriores, e conforme será mostrado nos exemplos de aplicação se mostrou efetivo no ajuste dos valores das larguras das funções RBF. O estabelecimento deste procedimento foi originado ao se verificar, nos testes realizados neste trabalho, que os valores de desvios que resultavam em melhores exatidões estavam relacionados com a média da distância entre os valores extremos dos dados dos clusters associados a cada agrupamento obtido no segundo passo da abordagem proposta. No caso de se obter um valor nulo decorrente da aplicação da equação (4.3) ou (4.4), repete-se o valor do desvio do cluster mais próximo.

$$\sigma'_{kj} = \frac{|\max(x_j) - \min(x_j)|}{2} \sigma_{kj}^{*-1} \quad (4.4)$$

- 6- Um fator multiplicativo (K_m) com valor entre [1, 10] (resultante de testes empíricos), comum a todos os dados de desvios, servirá como ajuste adicional das larguras resultantes (4.5). Esta faixa de valores foi estabelecida após os testes realizados neste trabalho, onde se verificou que valores maiores que dez não resultavam em melhoria de exatidão nos modelos RBF correspondentes, e valor menor que a unidade reduzia a exatidão resultante.

$$\sigma_{kj} = K_m \cdot \sigma'_{kj} \quad (4.5)$$

- 7- Os pesos w_j da camada de saída são calculados pelo método dos mínimos quadrados convencional (4.6), onde “ T ” denota matriz transposta e “ -1 ” inversa. Para os regressores F_r são utilizados as informações de saída (y_1 a y_p) das funções gaussianas $G_j(x)$ relacionadas com os dados de treinamento da entrada (x) da rede, e o correspondente padrão (Y_d) esperado para as informações de saída da rede em questão.

$$\begin{aligned} F_r &= [y_1 \ y_2 \ \dots \ y_p \ 1]; \\ w_j &= [F_r^T F_r]^{-1} F_r^T Y_d. \end{aligned} \quad (4.6)$$

- 8- Verificar a exatidão obtida pela rede. Se necessário variar o valor do fator multiplicativo do passo (6) aumentando o dado correspondente, ou então aumentar o número de nós (ou clusters) da rede de uma unidade na etapa (1), e repetir as etapas consecutivas do procedimento.

A Figura 4.2 ilustra o fluxograma que apresenta os passos envolvidos na abordagem proposta.

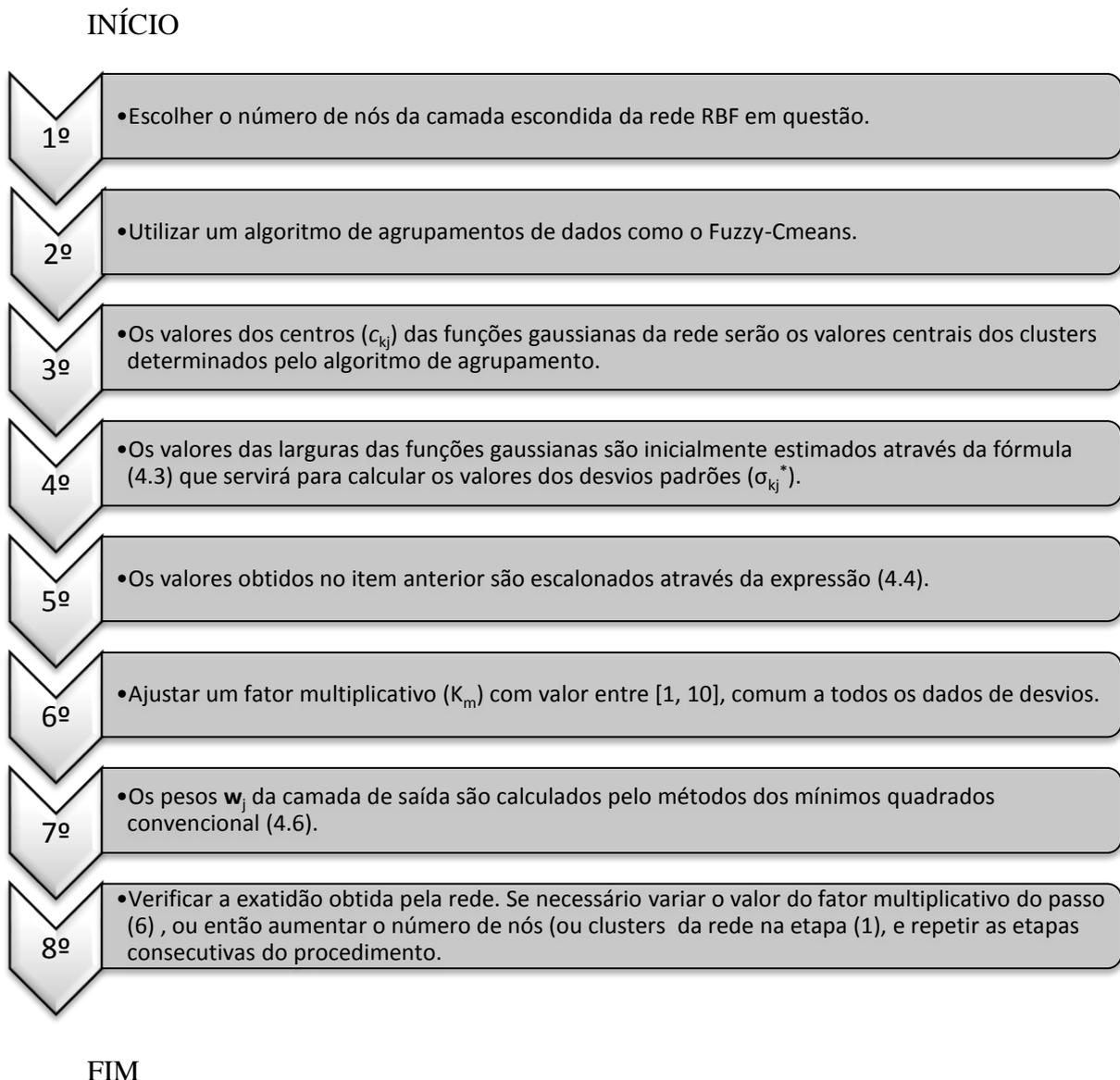


Figura 4.2 - Fluxograma da abordagem CPNSEP proposta.

O próximo capítulo contém resultados de algumas simulações realizados com a abordagem apresentada neste trabalho. Exemplos como aproximações de funções, modelagem de sistema dinâmico não linear, previsão de série temporal e classificação de padrões serão mostrados com a finalidade de exemplificar os procedimentos propostos, assim como comparar os resultados obtidos com outras técnicas de modelagem e parametrização de dados.

CAPÍTULO 5

5. Simulações

Neste capítulo serão realizadas algumas simulações com o objetivo de aplicar e testar a proposta apresentada no capítulo anterior.

5.1 Exemplos em Aproximações de Funções

Neste item serão exemplificadas algumas funções não lineares básicas com a finalidade de ilustrar a aplicação da abordagem proposta.

5.1.1 Exemplo 1

Este primeiro exemplo é meramente didático e tem como objetivo mostrar a aplicação da abordagem proposta e ilustrar os cálculos associados, onde será considerada uma rede neural RBF para modelar a função não linear expressa por (5.1).

$$y = x_1^2; \quad x_1 \in [0, 1]. \quad (5.1)$$

Neste exemplo foram utilizados dois conjuntos de dados com 21 amostras cada, sendo um deles utilizado para treinamento da rede e outro para efeito de teste da RBF resultante. Os vinte e um dados que representam a função em questão (e que serão utilizados para treinamento) são $x_I = [0; 0,05; 0,1; 0,15; 0,2; 0,25; 0,3; 0,35; 0,4; 0,45; 0,5; 0,55; 0,6; 0,65; 0,7; 0,75; 0,8; 0,85; 0,9; 0,95; 1]$ e $y = [0; 0,0025; 0,01; 0,0225; 0,04; 0,0625; 0,09; 0,1225; 0,16; 0,2025; 0,25; 0,3025; 0,36; 0,4225; 0,49; 0,5625; 0,64; 0,7225; 0,81; 0,9025; 1]$. Serão consideradas as seguintes simulações descritas a seguir.

I.1 - Inicialmente considerou-se o procedimento não supervisionado I (página 19 do Capítulo 2), com os valores dos centros das funções RBF selecionados aleatoriamente. Foi assumida uma rede com três nós na camada de entrada, sendo os dados dos valores dos centros escolhidos aleatoriamente a partir do vetor de dados x_I considerado no

exemplo (a Tabela 5.1 ilustra os valores dos centros das funções de base radial da rede em questão). Utilizando a expressão (2.1), obteve-se o valor comum das larguras das funções: $\sigma = \sigma_{11} = \sigma_{12} = \sigma_{13} = 0,26$. Com a expressão (2.5) têm-se na Tabela 5.2 os valores dos pesos da camada de saída da rede.

Tabela 5.1 - Valores dos centros das funções da rede referente ao Exemplo 1.

c_{11}	0,2
c_{12}	0,6
c_{13}	0,85

Tabela 5.2 - Valores dos pesos da camada de saída da rede referente ao Exemplo 1.

w_0	0,3299
w_1	-0,2448
w_2	-0,2815
w_3	0,6421

A Figura 5.1 ilustra a simulação da rede resultante para o conjunto de dados (de treinamento) da função exemplificada. A somatória do erro quadrático (SEQ) obtida foi de $7,0 \cdot 10^{-2}$. Um conjunto adicional de dados foi utilizado para testar a capacidade de interpolação da rede, onde os valores de entrada (x_1) relacionados com o exemplo foram incrementados em 0,0125 e o valor do erro quadrático obtido foi $SEQ = 7,8 \cdot 10^{-2}$. Nota-se da figura que a aproximação da função original não foi muito efetiva.

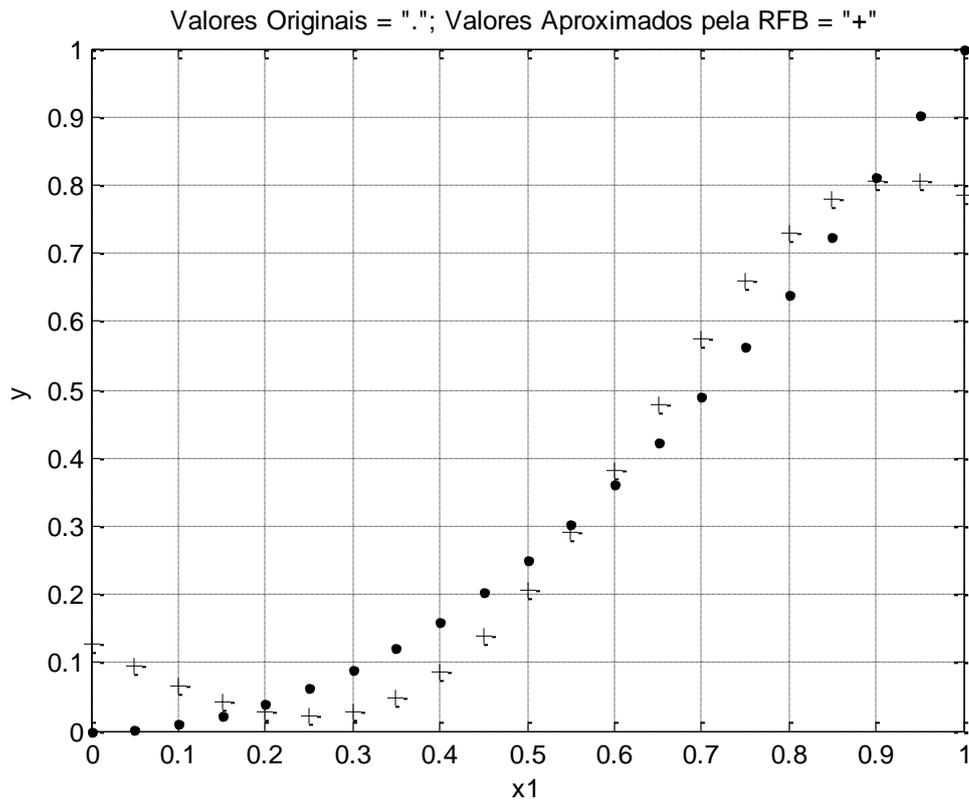


Figura 5.1 – Dados de treino para o Exemplo 1 e as aproximações obtidas pela rede RBF associada.

Ajustando agora os valores das larguras por meio da expressão (2.3), obtêm-se para $K = 1$ os valores ilustrados na Tabela 5.3, e na Tabela 5.4 os valores dos pesos da camada de saída.

Tabela 5.3 - Valores das larguras das funções da rede referente ao Exemplo 1.

σ_{11}	0,40
σ_{12}	0,25
σ_{13}	0,20

Tabela 5.4 - Valores dos pesos da camada de saída da rede referente ao Exemplo 1.

w_0	1,3901
w_1	-1,4084
w_2	0,0585
w_3	-0,3285

Considerando os mesmos conjuntos de dados de treinamento e teste, foram obtidos os valores da somatória do erro quadrático em $2,8 \cdot 10^{-2}$ e $2,5 \cdot 10^{-2}$, respectivamente, indicando uma melhoria na exatidão da modelagem resultante em relação aos valores anteriores de SEQ.

Adotando agora um fator de escala como em (2.3) ou (2.4) de valor $K = K_e = 3$, cujo dado foi testado objetivando alcançar uma melhor exatidão na modelagem obtida, têm-se os valores das larguras das funções da rede ilustrados na Tabela 5.5 e os valores dos pesos da camada de saída indicados na Tabela 5.6. Os valores dos erros de aproximação foram SEQ de $7,7 \cdot 10^{-5}$ e $7,0 \cdot 10^{-5}$ para os dados de treinamento e teste, respectivamente, indicando uma sensível melhora na exatidão do modelo resultante (Figura 5.2). A Tabela 5.7 apresenta um resumo dos dados obtidos até agora.

Tabela 5.5 - Valores das larguras das funções da rede referente ao Exemplo 1.

σ_{11}	1,2
σ_{12}	0,85
σ_{13}	0,60

Tabela 5.6 - Valores dos pesos da camada de saída da rede referente ao Exemplo 1.

w_0	7,1102
w_1	-8,8043
w_2	4,1214
w_3	-2,6859

Tabela 5.7 - Dados de centros, larguras e pesos da rede RBF do Exemplo 1 com os erros associados.

c_{11}	0,2	/	
c_{12}	0,6		
c_{13}	0,85		
σ_{11}	0,26	0,4	1,2
σ_{12}	0,26	0,25	0,85
σ_{13}	0,26	0,20	0,60
w_0	0,3299	1,3901	7,1102
w_1	-0,2448	-1,4084	-8,8043
w_2	-0,2815	0,0585	4,1214
w_3	0,6421	-0,3285	-2,6859
SEQ	$7,0 \cdot 10^{-2}$	$2,8 \cdot 10^{-2}$	$7,7 \cdot 10^{-5}$

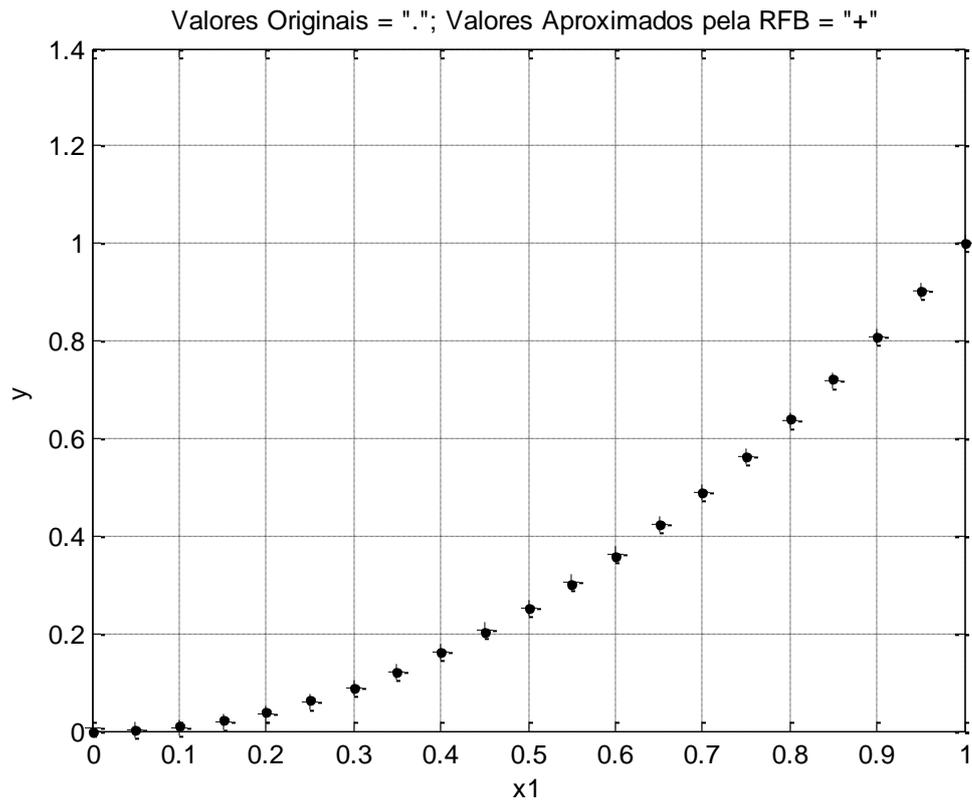


Figura 5.2 - Dados de treino do Exemplo 1 e as aproximações obtidas via rede RBF associada.

I.2 - Agora será assumido o procedimento não supervisionado III (descrito na página 21 do Capítulo 2) via utilização de método de agrupamento de dados. Serão empregados os mesmos dados que definem o exemplo e a rede com três funções de base radial. Aplicando o algoritmo Fuzzy C-Means nos dados de treinamento do exemplo, segue na Figura 5.3 a ilustração dos três agrupamentos de dados resultantes representados por três cores, e os respectivos centros dos clusters indicados por círculos na cor preta. Os valores dos centros dos agrupamentos são $(0,1728; 0,0435)$, $(0,5707; 0,3354)$ e $(0,8945; 0,8063)$, respectivamente, para as variáveis (x_1 e y) da função em questão. A Tabela 5.8 mostra os correspondentes valores dos centros dos *clusters* associados a variável x_1 .

Utilizando o procedimento definido pela equação (2.3) com $K = 3$, chega-se aos valores das larguras das funções (Tabela 5.9). Os valores dos pesos da camada de saída da rede estão indicados na Tabela 5.10. A rede RBF resultante apresentou os valores $SEQ = 4,4 \cdot 10^{-5}$ e $4,5 \cdot 10^{-5}$ para os dados de treinamento e os de teste, respectivamente, indicando uma determinada melhoria na exatidão da rede em relação à RBF anterior. A Tabela 5.11 apresenta um resumo dos dados obtidos neste contexto.

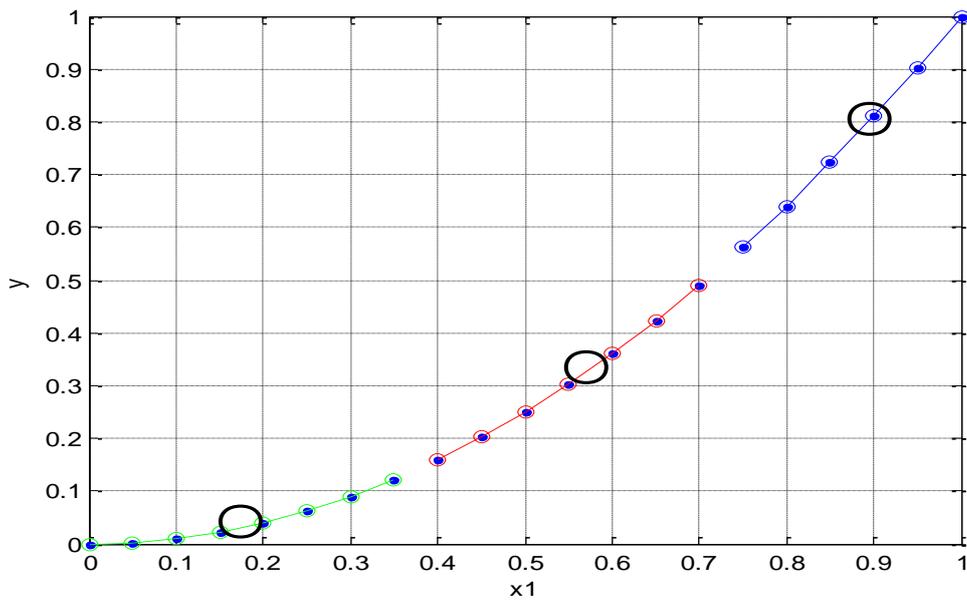


Figura 5.3 - Dados da função (5.1) e os respectivos agrupamentos.

Tabela 5.8 - Valores dos centros das funções da rede referente ao Exemplo 1.

c_{11}	0,1728
c_{12}	0,5707
c_{13}	0,8945

Tabela 5.9 - Valores das larguras das funções da rede referente ao Exemplo 1.

σ_{11}	1,19
σ_{12}	0,97
σ_{13}	0,97

Tabela 5.10 - Valores dos pesos da camada de saída da rede referente ao Exemplo 1.

w_0	8,7252
w_1	-14,7917
w_2	13,6765
w_3	-8,5432

Tabela 5.11 - Dados de centros, larguras e pesos da rede e o valor do erro associado.

c_{11}	0,1728
c_{12}	0,5707
c_{13}	0,8945
σ_{11}	1,19
σ_{12}	0,97
σ_{13}	0,97
w_0	8,7252
w_1	-14,7917
w_2	-13,6765
w_3	- 8,5432
SSE	$4,4 \cdot 10^{-5}$

I.3 - Agora será considerada a abordagem CPNSEP proposta. Assumindo as mesmas informações da função exemplificada, e o mesmo agrupamento de dados obtido com a aplicação do algoritmo Fuzzy C-Means, seguem os resultados dos passos propostos no procedimento. Para as informações de x_1 , os três agrupamentos de dados resultantes são [0; 0,05; 0,1; 0,15; 0,2; 0,25; 0,3; 0,35], [0,4; 0,45; 0,5; 0,55; 0,6; 0,65; 0,7] e [0,75; 0,8; 0,85; 0,9; 0,95; 1]. Calculando os desvios padrões associados a cada agrupamento com a equação (4.3), têm-se na Tabela 5.12 os valores correspondentes.

Tabela 5. 12 - Valores dos desvios padrões das funções da rede associada ao Exemplo 1.

σ_{11}^*	0,1225
σ_{12}^*	0,1080
σ_{13}^*	0,0935

Os respectivos valores escalonados pela expressão (4.4) são então obtidos: $\sigma_{11}' = 0,5 \cdot (0,35 - 0) / 0,1225 = 1,4286$; $\sigma_{12}' = 0,5 \cdot (0,7 - 0,4) / 0,108 = 1,3889$; $\sigma_{13}' = 0,5 \cdot (1 - 0,75) / 0,0935 = 1,3369$. E utilizando o mesmo valor para os fatores de escala considerados anteriormente ($K = K_e = K_m = 3$) em (4.5), ajustado com a finalidade de obter-se uma melhor exatidão nas modelagens consideradas, segue na Tabela 5.13 os valores das larguras resultantes.

Tabela 5.13 - Valores das larguras das funções radiais da rede referente ao Exemplo 1.

σ_{11}	4,2857
σ_{12}	4,1667
σ_{13}	4,0107

Empregando os dados de entrada (\mathbf{x}_l) de treinamento nas funções gaussianas resultantes e calculando os valores das três saídas de ativação correspondentes (\mathbf{y}_1 , \mathbf{y}_2 e \mathbf{y}_p), juntamente com os dados de saída do exemplo utilizados para treinamento ($\mathbf{Y}_d = \mathbf{y}$), chega-se ao regressor (\mathbf{F}_r) expresso por (4.6) do qual são calculados os pesos da camada de saída da rede (Tabela 5.14).

Tabela 5.14 - Valores dos pesos da camada de saída da rede referente ao Exemplo 1.

w_0	33,8119
w_1	-369,0504
w_2	672,7098
w_3	-339,8387

Simulando (Figura 5.4) o modelo da rede RBF resultante, os valores da SEQ obtidos foram $8,6 \cdot 10^{-8}$ e $9,2 \cdot 10^{-8}$ para os dados de treinamento e teste, respectivamente, indicando um sensível incremento na exatidão da modelagem obtida em relação aos resultados anteriores. Neste exemplo um aumento no fator K_m (em valores maiores que 3), não resultou em uma melhora da exatidão da rede RBF em questão. A Tabela 5.15 apresenta um resumo dos dados obtidos com abordagem CPNSEP proposta.

Tabela 5.15 - Dados dos centros, larguras e pesos da RBF com a abordagem proposta.

c_{11}	0,1728
c_{12}	0,5707
c_{13}	0,8945
σ_{11}	4,2857
σ_{12}	4,1667
σ_{13}	4,0107
w_0	33,8119
w_1	-369,0504
w_2	672,7098
w_3	-339,8387

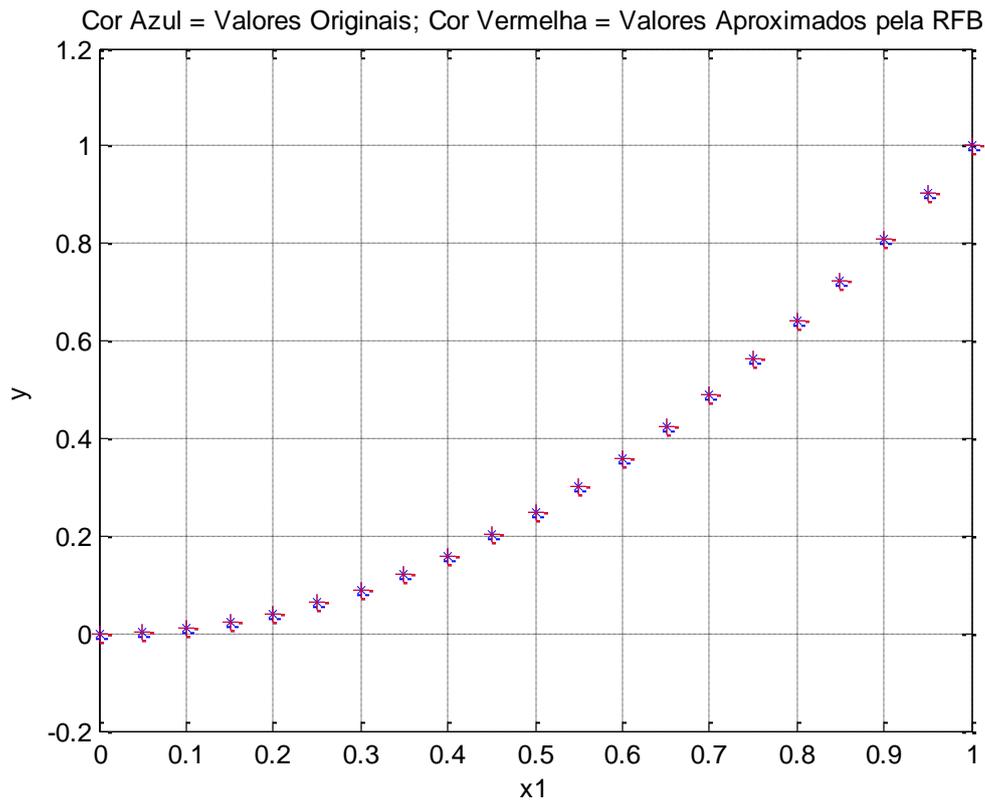


Figura 5.4 - Dados da função (5.1) e as aproximações dadas pela rede RBF resultante.

O resultado obtido com a abordagem proposta se mostrou melhor em relação aos obtidos com as estratégias não supervisionadas referenciadas, com os valores da somatória do erro quadrado indicados na Tabela 5.16. O procedimento proposto para ajustes de parâmetros de redes RBF se mostrou de aplicação direta, e os algoritmos computacionais envolvidos são todos não supervisionados.

Tabela 5.16 - SEQ associado ao método utilizado em relação ao Exemplo 1.

Método	SEQ
I	$7,0 \cdot 10^{-2}$
I e (2.3) com $K=1$	$2,8 \cdot 10^{-2}$
I e (2.4) com $K=3$	$7,7 \cdot 10^{-5}$
III com $K=3$	$4,4 \cdot 10^{-5}$
CPNSEP com $K_m=3$	$8,6 \cdot 10^{-8}$

5.1.2 Exemplo 2

Este exemplo utiliza uma função não linear (5.2) com duas variáveis de entrada que exemplificará um modelo estático com duas não linearidades (o produto das duas variáveis e o quadrado de uma delas). Este modelo pode representar também uma equação de estados de segunda ordem resultante da modelagem de um sistema dinâmico não linear específico.

$$y = x_1x_2 + x_2^2; \quad x_1, x_2 \in [-1, +1]. \quad (5.2)$$

A função deste exemplo está representada por cem dados para cada variável e será aproximada por uma rede RBF, onde inicialmente se atribuiu três nós para a camada oculta da mesma. Foram gerados aleatoriamente dois conjuntos de dados, onde o primeiro foi utilizado para treinamento e o segundo para teste. Como a aproximação obtida não atingiu um SSE estipulado da ordem de 10^{-2} com a utilização dos procedimentos citados anteriormente, por exemplo, o número de nós da rede foi aumentado para seis. Serão consideradas as simulações descritas a seguir.

II.1 - Inicialmente será considerada a parte do procedimento não supervisionado III descrito no Capítulo 2 relativo à aplicação de um método de agrupamento de dados. Aplicando o algoritmo *Fuzzy C-Means* nos dados de treinamento do exemplo, têm-se na Figura 5.5 os seis agrupamentos resultantes, cujos valores dos centros relativos às variáveis x_1 e x_2 estão indicados na Tabela 5.17.

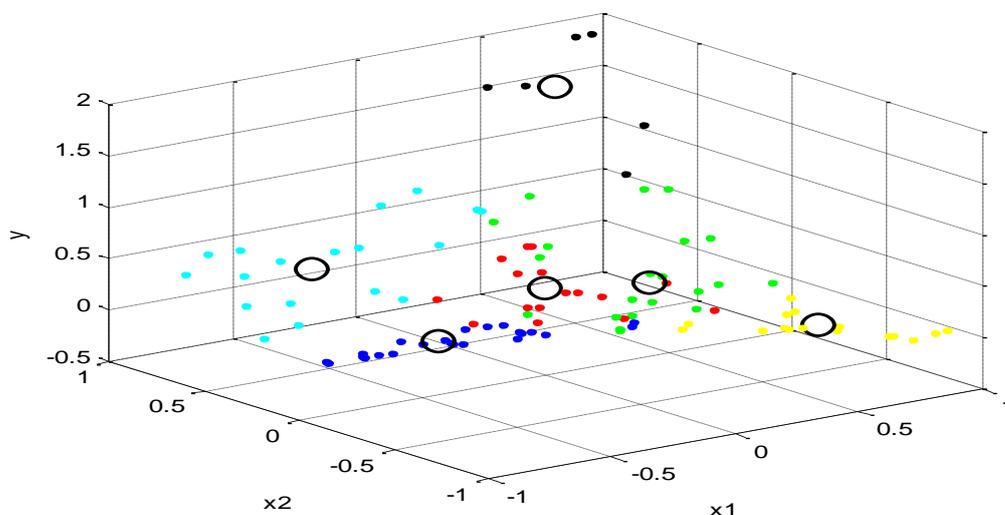


Figura 5.5 - Dados da função (5.2) e os respectivos agrupamentos.

Tabela 5.17 - Valores dos centros das funções da rede referente ao Exemplo 2.

c_{11}	-0,5761
c_{12}	-0,5041
c_{13}	0,4933
c_{14}	0,7217
c_{15}	0,7127
c_{16}	-0,3878
c_{21}	-0,7507
c_{22}	-0,0900
c_{23}	0,0975
c_{24}	0,8879
c_{25}	-0,5115
c_{26}	0,7286

Será utilizada a expressão (2.3) para estabelecer um valor de desvio padrão comum para as funções gaussianas da rede, resultando em $\sigma = 0,473$. Escolhendo um fator de escala comum com o valor $K_e = 4$, no qual este dado foi testado até se alcançar uma melhor exatidão nos modelos obtidos, têm-se na Tabela 5.18 os valores dos pesos da camada de saída calculados através do método dos mínimos quadrados a partir dos dados de treinamento. Simulando a rede RBF com os dados de treinamento obteve-se um valor SEQ de $4,4 \cdot 10^{-2}$, e $4,1 \cdot 10^{-2}$ para os dados de teste.

Tabela 5.18 - Valores dos pesos da camada de saída da rede referente ao Exemplo 2.

w_0	-1,5935
w_1	31,6080
w_2	-34,7809
w_3	-28,6639
w_4	21,1745
w_5	6,8275
w_6	12,1974

II.2 - Utilizando agora a abordagem proposta e empregando o mesmo valor do fator de escala ($K_m = 4$) de ajuste das larguras das RBF, têm-se na Tabela 5.19 os valores das larguras correspondentes. Com as informações de treinamento (dados x_1 , x_2 e y da

Figura 5.5) se calcula via método dos mínimos quadrados os valores dos pesos da camada de saída (Tabela 5.20). A Figura 5.6 ilustra a simulação da RBF resultante com os dados de teste, onde se obteve um valor da somatória do erro quadrático de $1,2 \cdot 10^{-2}$, indicando uma melhora na exatidão do modelo em relação ao teste anterior. Um aumento no valor do fator de escalonamento ($K_m = 4$), não resultou em uma melhoria significativa na exatidão da rede em questão para o procedimento proposto.

Tabela 5.19 - Valores das larguras das funções da rede referente ao Exemplo 2.

σ_{11}	7,3743
σ_{12}	6,4388
σ_{13}	6,6256
σ_{14}	4,8571
σ_{15}	7,1114
σ_{16}	5,9998
σ_{21}	6,4676
σ_{22}	6,9011
σ_{23}	6,8731
σ_{24}	4,6284
σ_{25}	6,8415
σ_{26}	6,7363

Tabela 5.20 - Valores dos pesos da camada de saída da rede referente ao Exemplo 2.

w_0	57,1692
w_1	-165,0271
w_2	$2,0472 \cdot 10^3$
w_3	$3,025 \cdot 10^3$
w_4	67,9285
w_5	$-2,7074 \cdot 10^3$
w_6	$-2,3491 \cdot 10^3$

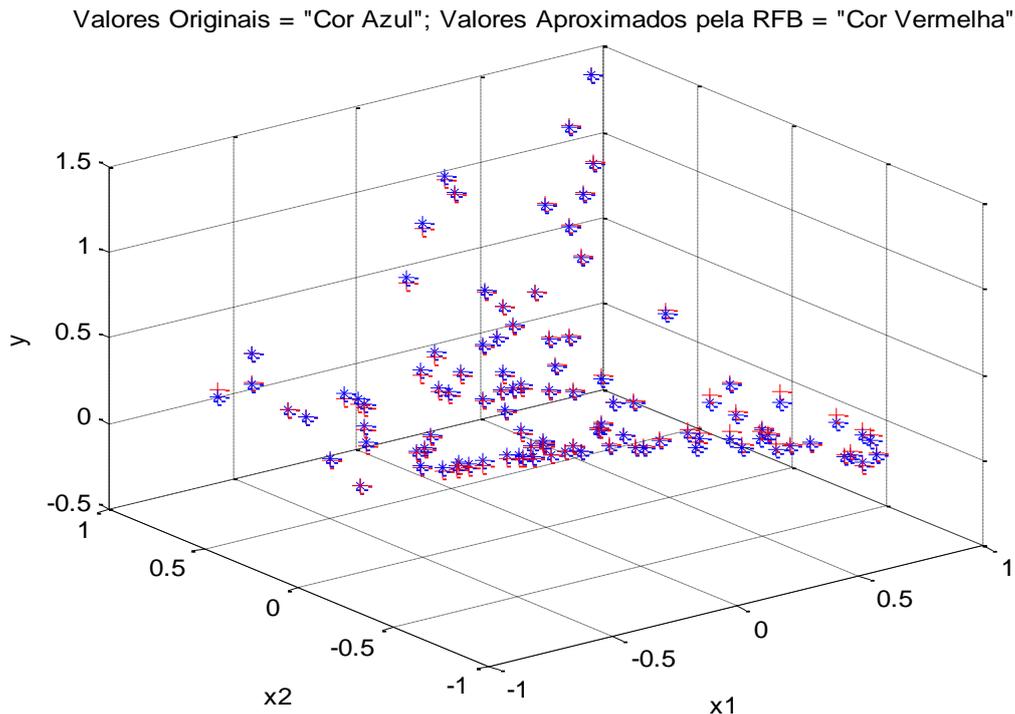


Figura 5.6 - Dados da função (5.2) e as aproximações realizadas pela rede RBF associada.

Nos próximos itens serão mostradas aplicações em modelagem de um sistema dinâmico não linear, em previsão de série temporal e em classificação de padrões.

5.2 Exemplo de Modelagem de Sistema Dinâmico Não Linear

Neste item será exemplificada a modelagem de um sistema não linear usualmente empregado em processos industriais.

5.2.1 Exemplo 3

Sistemas de nível são processos usuais em plantas industriais. Quando se conhece a priori as características estruturais de um processo de nível típico é possível utilizar a equação (5.3) para a modelagem destes sistemas (Ogata, 2010). Na equação o símbolo “ u ” representa a variável de comando da válvula de vazão de entrada do sistema, e “ y ” o nível do processo correspondente. Onde “ A ” denota o valor da seção transversal do reservatório do sistema, “ a ” o valor da seção transversal da tubulação de saída do processo, “ c ” é uma constante de proporcionalidade e “ g ” a aceleração gravitacional.

$$A \frac{dy(t)}{dt} = c \cdot u(t) - a\sqrt{2g} \cdot \sqrt{y(t)} \quad (5.3)$$

Com a utilização frequente de instrumentos de medidas digitais em processos industriais, uma representação discreta (auto-regressiva) é mais adequada para efeito de modelagens práticas, onde usualmente as informações disponíveis estão relacionadas com as informações de entrada e saída do sistema a ser modelado.

As figuras seguintes ilustram amostras de entrada e saída de um sistema físico que será considerado neste exemplo. Os dados nas figuras representam as informações de entrada “ u ” e de saída “ y ” de um sistema de nível prático, constituindo os valores de tensão de comando da válvula do processo e do nível do sistema, que foi devidamente medido por um transdutor. Cada amostra (n), atribuída as informações de entrada e saída do sistema, possui um intervalo de tempo de um segundo em relação à próxima. Os dados da Figura 5.7 serão utilizados para a obtenção dos modelos deste exemplo, e os da Figura 5.8 servirão para testar as modelagens obtidas.

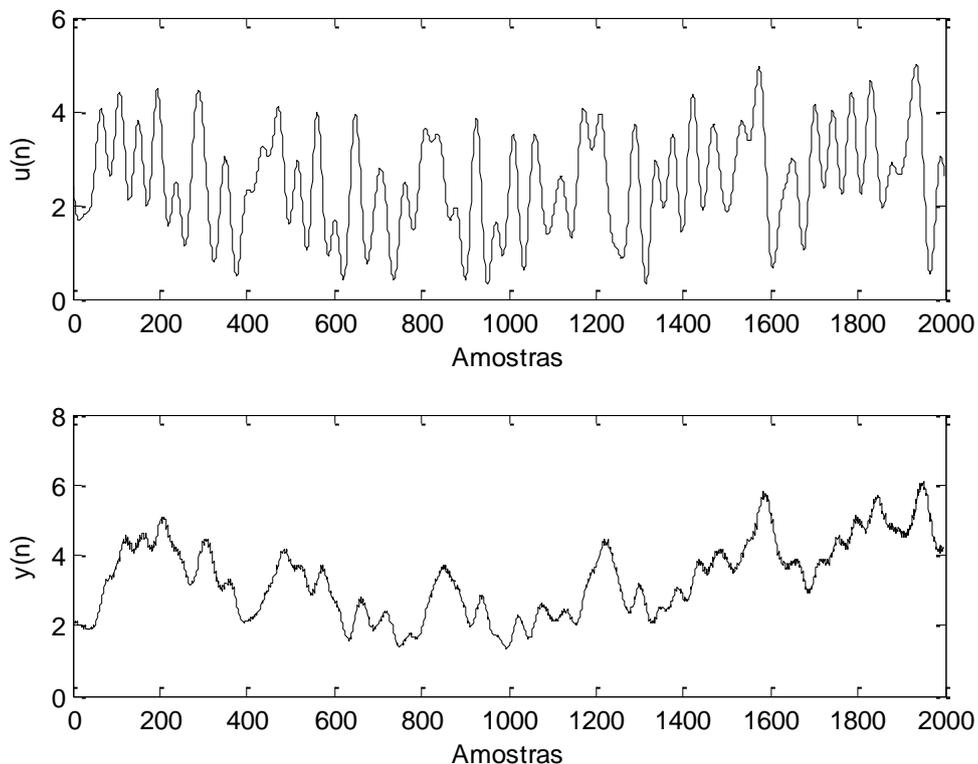


Figura 5.7 - Dados originais de entrada e saída de um sistema de nível.

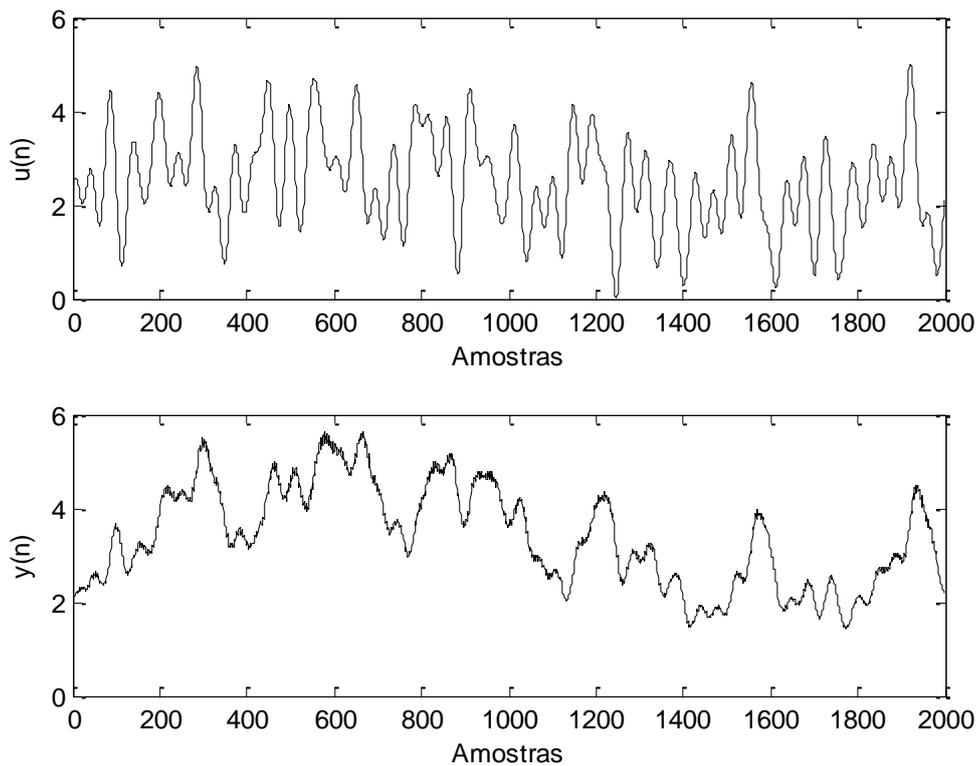


Figura 5.8 - Dados originais de entrada e saída de um processo de nível.

Serão consideradas as simulações descritas a seguir.

III.1 - Inicialmente será utilizado um modelo ARX com a estrutura definida pela equação $y(n) = a_1y(n-1) + b_1u(n-1)$, cujos coeficientes $a_1 = 0,9787$ e $b_1 = 0,0298$ foram obtidos pelo método dos mínimos quadrados a partir dos dados da Figura 5.7. A Figura 5.9 ilustra os valores do modelo ARX resultante e os dados originais do sistema de nível. A correlação entre os dados medidos do sistema e os estimados pelo modelo resultou em um grau de correlação cruzada (*best fits*) de 58,47%. Nota-se do gráfico que a modelagem obtida não reproduziu adequadamente as informações originais do sistema, o que é explicado pelo processo real possuir característica não linear e o modelo ARX ser linear. Foram testados outros modelos ARX com mais dados de amostras anteriores das informações de entrada e saída do sistema, mas que não resultaram em melhorias significativas nas exatidões das modelagens consideradas.

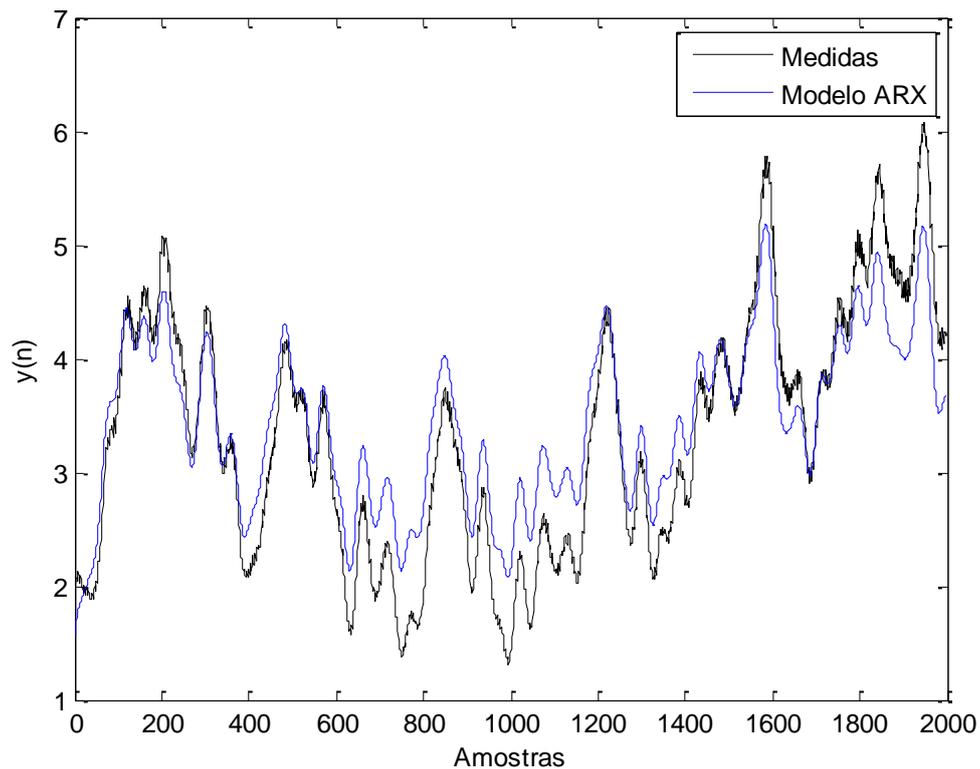


Figura 5.9 - Dados do processo de nível e do modelo ARX associado.

III.2 - A Figura 5.10 mostra valores de um modelo NARX obtido pelo *toolbox* “*ident*” do MATLAB (<http://www.mathworks.com/>), onde foram utilizadas as seguintes informações de entrada e saída do processo $u(n-1)$, $u(n-2)$, $y(n-1)$ e $y(n-2)$ para estimar-se a informação atual $y(n)$. O grau de correlação cruzada (*best fits*) foi agora de 84,05% (um valor correspondente foi obtido com os dados de teste), indicando uma melhor exatidão deste modelo em relação ao linear. Na prática um dado de correlação cruzada com valor maior que 85% é considerado adequado em contextos de identificação de sistemas dinâmicos, principalmente em processos que possuam não linearidades (Aguirre, 2007).

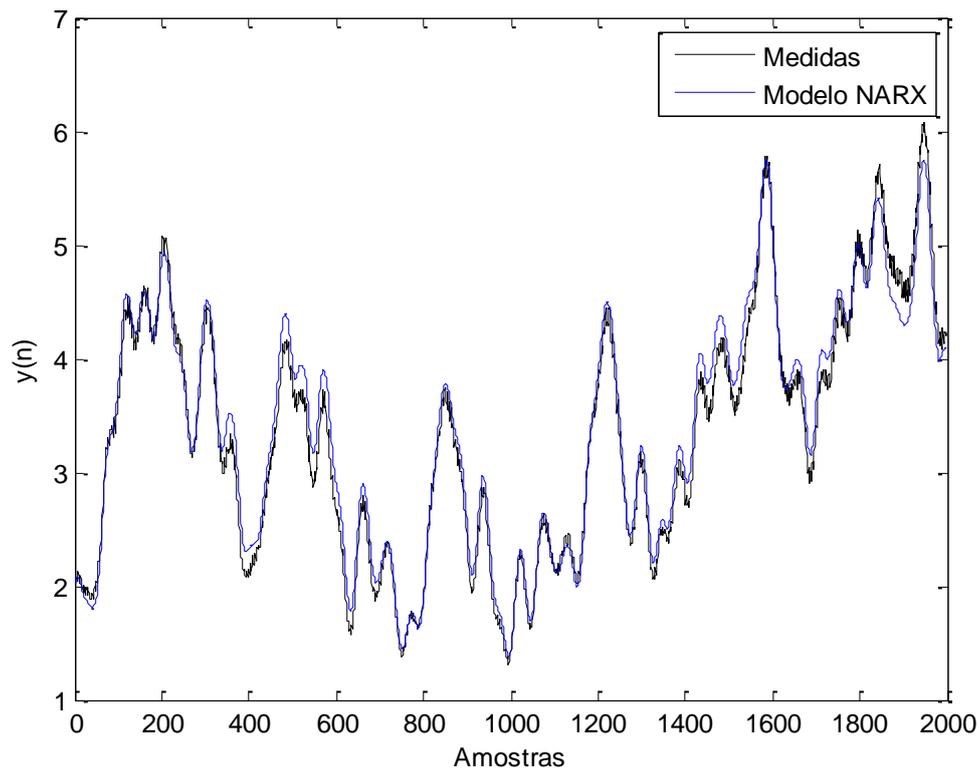


Figura 5.10 - Dados do processo de nível e do modelo NARX correspondente.

Redes neurais artificiais do tipo MLP são muitas vezes utilizadas na modelagem de sistemas não lineares como alternativas aos modelos NARX e outros (Narendra e Parthasarathy, 1990). A Figura 5.11 ilustra uma estrutura de modelagem via rede neural artificial, cujas amostras “ $k - 1$ ” (ou “ $n - 1$ ”) das informações de comando e de saída do modelo são utilizadas como entrada da rede correspondente.

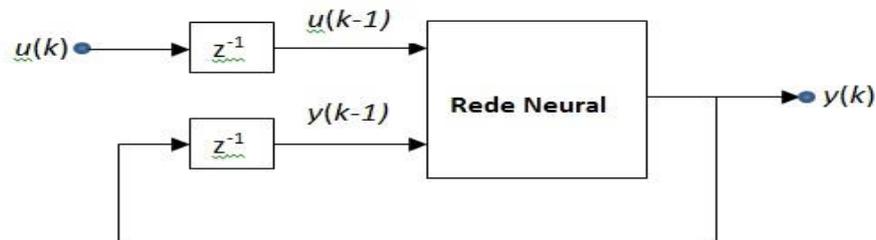


Figura 5.11 - Exemplo de estrutura de ANN utilizada para modelagem de sistemas.

III.3 - Utilizando os dados de treinamento em uma rede MLP com seis neurônios na camada escondida, funções de ativação tipo sigmoide, saída com função linear, algoritmo de aprendizado via retro-propagação com termo de momento (valor inicial em

0,07, incremento de 1,2 e decremento de 0,5), obteve-se após 1000 épocas de treinamento os seguintes pesos e bias para a MLP: $W_{input} = [-0,19515 \ -6,2093; \ -2,5127 \ -0,93316; \ -0,11678 \ -4,1868; \ -0,18255 \ -5,6846; \ 0,32383 \ 7.6596; \ -5,4033 \ -5,5194]$; $B_{input} = [4,6742; \ 7,4321; \ -1,0862; \ 1,2081; \ 5,6553; \ -12,0461]$; $W_{output} = [-0,63542; \ 0,48483; \ -0,73324; \ -0,38714; \ 0,39476; \ 0,12763]$; $B_{output} = 0,21881$. Simulando a MLP obteve-se um dado de SQE de 0,09 para os dados de treinamento, e um valor de 31,35 com os dados de teste. Em relação ao modelo NARX convencional (com dois pares de amostras anteriores para cada variável de entrada do modelo), a resposta da rede neural obtida (com apenas uma amostra anterior para as informações de entrada e saída de modelagem), apresentou uma melhor capacidade de aproximação.

III.4 - Foi testado um modelo *neuro-fuzzy* obtido através do *toolbox* do MATLAB denominado ANFIS (maiores detalhes deste tipo de modelo *neuro-fuzzy* estão no Anexo 1). Empregou duas funções de pertinência gaussianas para cada variável de entrada e após cinquenta épocas de treinamento se obteve as seguintes regras de modelagem:

$$\begin{aligned} \text{IF } x_1 = A_1 \text{ AND } x_2 = B_1 \text{ THEN } y_1 &= c_{11}x_1 + c_{12}x_2 + c_{13}; \\ \text{IF } x_1 = A_1 \text{ AND } x_2 = B_2 \text{ THEN } y_2 &= c_{21}x_1 + c_{22}x_2 + c_{23}; \\ \text{IF } x_1 = A_2 \text{ AND } x_2 = B_2 \text{ THEN } y_3 &= c_{31}x_1 + c_{32}x_2 + c_{33}; \\ \text{IF } x_1 = A_2 \text{ AND } x_2 = B_1 \text{ THEN } y_4 &= c_{41}x_1 + c_{42}x_2 + c_{43}. \end{aligned}$$

Em relação às grandezas do sistema de nível, as variáveis do modelo *fuzzy* são $x_1 = u(n-1)$, $x_2 = y(n-1)$ e $Y = y(n)$. Os parâmetros foram ajustados pelo algoritmo de treinamento para as funções de pertinência gaussianas A_1 , A_2 , B_1 e B_2 , cujos valores dos larguras/centros foram: 2,221/0,6816; 1,563/5,532; 1,946/1,228; 2,105/6,011. Os valores obtidos para os coeficientes do modelo foram: $c_{1i} = [0,03206; \ 0,9814; \ -0,02772]$; $c_{2i} = [0,03859; \ 0,9805; \ -0,01021]$; $c_{3i} = [0,03013; \ 0,953; \ 0,02059]$; $c_{4i} = [0,05979; \ 0,9428; \ 0,08217]$. Simulando o modelo *fuzzy* resultante com os dados de treinamento e de teste, foram obtidos os valores da somatória do erro quadrático de modelagem em 0,001 e 30,58, respectivamente.

III.5 - Agora uma rede RBF será empregada na modelagem do processo de nível em questão, onde a correspondência entre as grandezas do processo e do modelo neural são $x_1 = u(n-1)$, $x_2 = y(n-1)$ e $Y = y(n)$. Escolhendo seis nós para a camada escondida da rede, no mesmo contexto do exemplo anterior, e aplicando o algoritmo de agrupamento *Fuzzy C-Means* nos dados de treinamento, a Figura 5.12 ilustra os seis

clusters associados (cujos valores dos centros relativos às variáveis x_1 e x_2 estão indicados na Tabela 5.21).

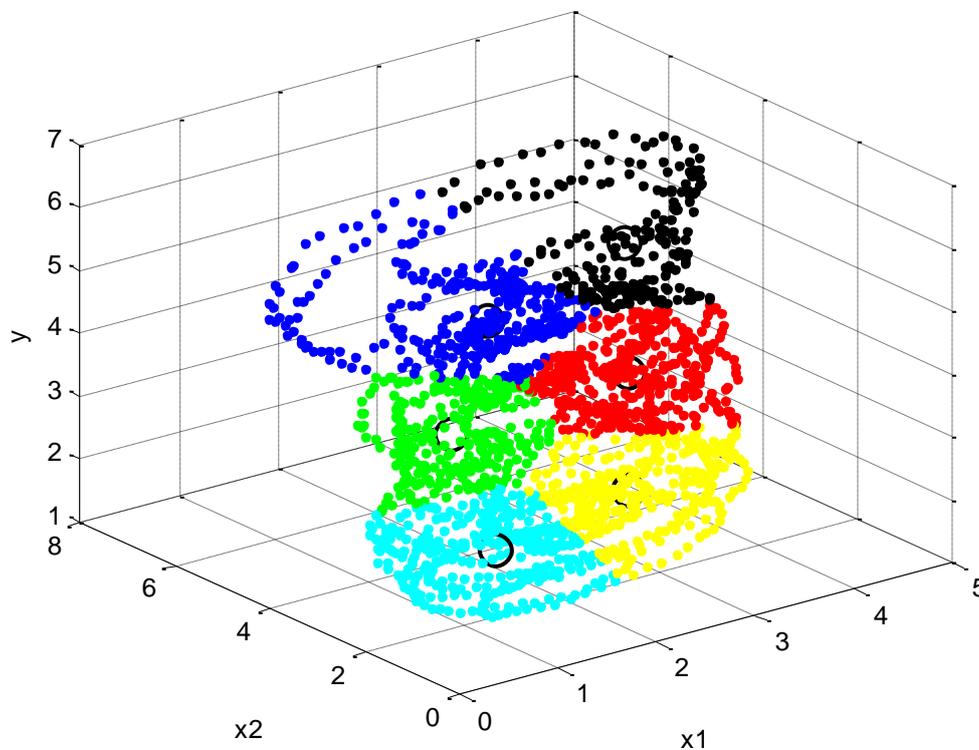


Figura 5.12 - Dados do sistema de nível e os respectivos agrupamentos.

Tabela 5.21 - Valores dos centros das funções da rede referente ao Exemplo 3.

c_{11}	3,386
c_{12}	2,4084
c_{13}	1,5415
c_{14}	3,9878
c_{15}	2,7912
c_{16}	1,3438
c_{21}	3,4805
c_{22}	4,4395
c_{23}	3,3628
c_{24}	4,8107
c_{25}	2,2687
c_{26}	2,0532

Utilizando a abordagem proposta e empregando um fator de escala com valor $K_m = 10$, onde este dado foi testado até se alcançar uma melhor exatidão na modelagem obtida, seguem os valores das larguras na Tabela 5.22 e os valores dos pesos da camada de saída da rede na Tabela 5.23.

Tabela 5.22 - Valores das larguras das funções da rede referente ao Exemplo 3.

σ_{11}	20,4875
σ_{12}	22,0715
σ_{13}	20,5906
σ_{14}	22,5496
σ_{15}	19,5475
σ_{16}	18,8138
σ_{21}	22,1254
σ_{22}	23,9608
σ_{23}	21,3108
σ_{24}	19,3668
σ_{25}	24,5088
σ_{26}	20,4538

Tabela 5.23 - Valores dos pesos da camada de saída da rede referente ao Exemplo 3.

w_0	-63,9392
w_1	-291,7164
w_2	441,7945
w_3	-201,6988
w_4	78,1658
w_5	45,6544
w_6	-4,5233

Simulando a rede RBF resultante com os dados de treinamento obteve-se um valor para a somatória quadrática do erro de 4,81 e um valor de 10,25 com os dados de teste. Segue na Figura 5.13 o resultado da simulação do modelo do sistema de nível via rede RBF (nos Anexos 2 e 3 encontram-se os programas associados). O resultado obtido indica que o procedimento proposto é útil para ajustar parâmetros de redes RBF em aplicações de modelagem de sistemas dinâmicos com características não lineares.

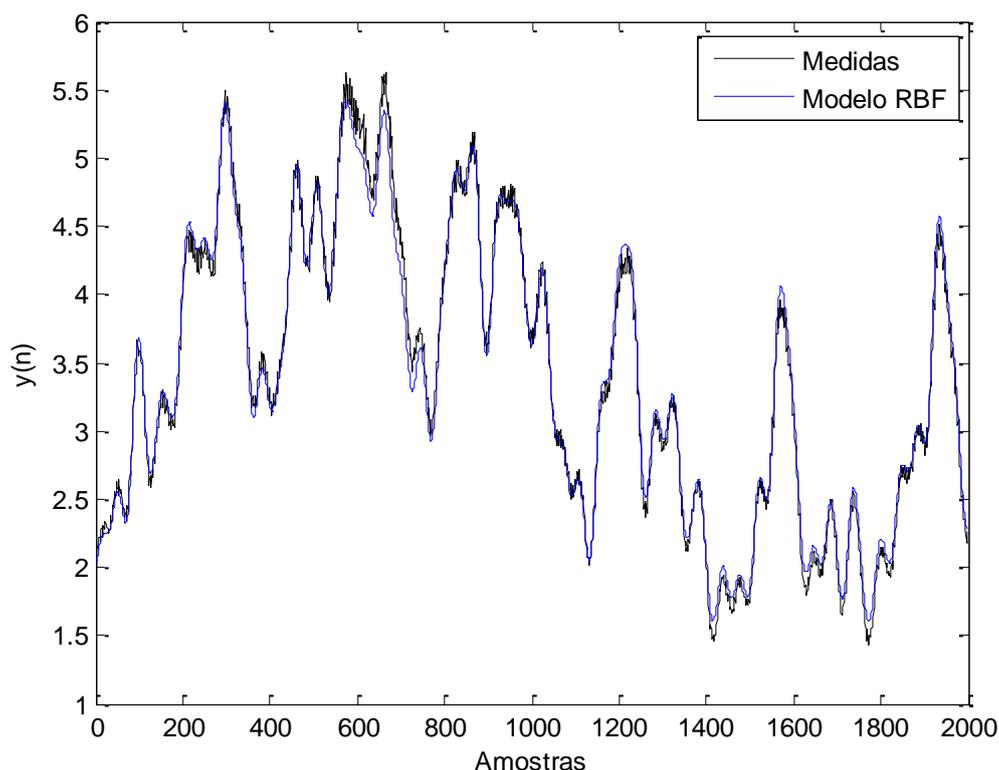


Figura 5.13 - Dados reais do sistema de nível e os valores estimados pela rede RBF.

Os valores dos erros obtidos com a rede RBF se mostraram melhores em relação à rede MLP utilizada que empregou seis neurônios na camada escondida.

Os dados práticos utilizados nos testes deste exemplo foram provenientes de medidas da instrumentação do processo. Os dados coletados exibidos na Figura 5.7 e na Figura 5.8 apresentaram um determinado nível de ruído, o que pode ser notado nos gráficos correspondentes das medidas pelas ondulações nas informações do sistema, destacando que não foi utilizado nenhum tipo de filtragem nos dados coletados do processo de nível em questão.

Em simulações realizadas preliminarmente com dados sem a presença de ruídos, se verificou melhores exatidões nos modelos considerados (MLP, *Fuzzy* e RBF) em relação à utilização de informações provenientes de medidas reais. Estes resultados eram esperados, pois a presença de ruídos ou incertezas em dados práticos pode decrementar a exatidão dos modelos resultantes. O exemplo prático empregado indica que o procedimento apresentado nesta pesquisa para o ajuste de parâmetros de redes RBF, se mostrou adequado para aplicações de modelagem de sistemas reais. Os valores das somatórias dos erros quadráticos dos modelos considerados estão na Tabela 5.24.

Tabela 5.24 - Comparação de resultados dos modelos do processo de nível.

SEQ		
<i>Modelo</i>	Treino	Teste
MLP	0,090	31,35
Fuzzy	0,001	30,58
RBF	4,810	10,25

5.3 Exemplo de Previsão de Série Temporal

Nesta simulação será empregada uma rede RBF obtida com a metodologia proposta neste trabalho, objetivando uma aplicação de previsão de série temporal.

5.3.1 Exemplo 4

A Figura 5.14 contém dados de uma série temporal relativa a valores de arroba de boi, utilizada por Faustino (2011), onde se têm registros de valores (em reais) relativos a setenta e três meses. Serão utilizados valores dos primeiros sessenta e dois meses com a finalidade de treinar três modelos, para depois verificar a capacidade destes de estimarem os valores dos onze meses seguintes. Nos modelos foram consideradas as informações da amostra atual (n) de valor de preço, e outra do valor anterior ($n-1$) com a finalidade de estimar o preço do mês seguinte ($n+1$).

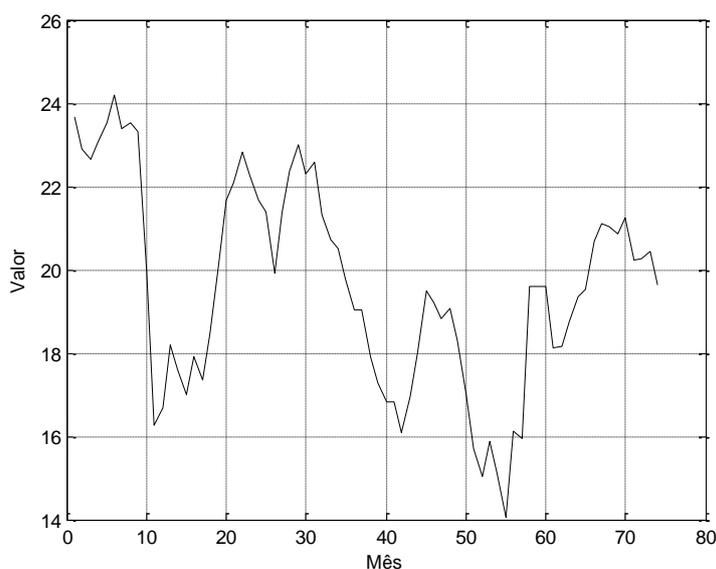


Figura 5.14 - Dados da série de preço de arroba de boi.

Serão consideradas as simulações descritas a seguir.

4.1 – Inicialmente testou-se um modelo *fuzzy*, onde o ajuste de parâmetros com os dados de treinamento apresentou um erro de modelagem aceitável, mas o resultado obtido com os dados de teste não se mostrou adequado (apesar das configurações testadas com variados números e tipos das funções de pertinência utilizadas, assim como o emprego de técnicas de agrupamento de dados no pré-processamento de informações de treinamento). A utilização de mais entradas com informações de outras amostras anteriores dos dados da série pode conduzir a exatidões melhores na modelagem correspondente. Entretanto, a finalidade das simulações deste exemplo é comparar modelos com o mesmo número de entradas (neste caso, duas).

4.2 - Neste exemplo considerou-se o emprego da função *newrb* do *toolbox* de rede neural do MATLAB, onde um procedimento supervisionado utiliza os dados de treinamento da aplicação para ajustar os parâmetros da rede RFB correspondente. Foram simulados alguns valores de larguras (*spread*) para as funções de ativação incluídas na rede, e notou-se que quanto mais exato o modelo resultante do treinamento maior o erro nas previsões subsequentes. Para efeito de comparações foi utilizado o erro porcentual absoluto médio (EPAM) entre os valores fornecidos para o padrão de treinamento/teste (Y_d) e os dados de saída (Y_e) estimados, ou seja, $\sum |(Y_d - Y_e)/ Y_d| / N$, sendo N o número de valores considerados. Tanto para os dados de treinamento como para as informações de teste de previsão, os melhores valores de EPAM obtidos foram de 3,8%.

4.3 - Agora será utilizada a abordagem proposta, onde as variáveis da rede RBF considerada são as mesmas das simulações anteriores deste exemplo, ou seja, $x_1 =$ “Valor do Preço Atual”, $x_2 =$ “Valor do Preço Anterior” e $Y =$ “Valor do Mês Seguinte”. Para a camada escondida da rede adotou-se seis nós. Os seis agrupamentos de dados processados pelo algoritmo *Fuzzy C-Means* estão ilustrados na Figura 5.15.

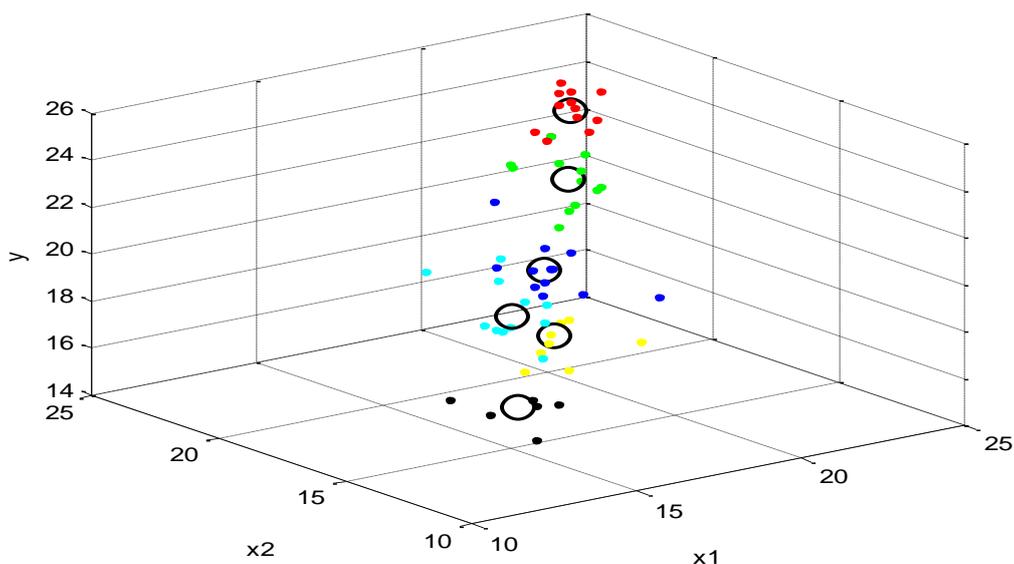


Figura 5.15 - Dados da série temporal e os respectivos agrupamentos.

Os seis clusters associados têm seus valores dos centros relativos às variáveis x_1 e x_2 indicados na Tabela 5.25.

Tabela 5.25 - Valores dos centros das funções da rede referente ao Exemplo 4.

c_{11}	23,1373
c_{12}	19,2555
c_{13}	21,7906
c_{14}	15,5351
c_{15}	18,2458
c_{16}	16,823
c_{21}	23,1892
c_{22}	19,2246
c_{23}	21,5552
c_{24}	15,3847
c_{25}	17,4568
c_{26}	17,3302

Utilizando a abordagem proposta e empregando um fator de escala com valor $K_m = 10$, cujo este dado foi testado até se alcançar uma melhor exatidão na modelagem obtida, e tem-se na Tabela 5.26 os valores referentes às larguras das funções da rede e os valores dos pesos da camada de saída na Tabela 5.27.

Tabela 5.26 - Valores das larguras das funções da rede referente ao Exemplo 4.

σ_{11}	18,6332
σ_{12}	19,5991
σ_{13}	16,7311
σ_{14}	14,8883
σ_{15}	15,4140
σ_{16}	15,1891
σ_{21}	17,0218
σ_{22}	18,3433
σ_{23}	18,1947
σ_{24}	13,7824
σ_{25}	15,3732
σ_{26}	16,7565

Tabela 5.27 - Valores dos pesos da camada de saída da rede referente ao Exemplo 4.

w_0	-5,5924
w_1	-3,2244
w_2	-55,865
w_3	122,6187
w_4	1,5879
w_5	-282,6299
w_6	246,7825

A Figura 5.16 mostra os dados originais da série temporal utilizados para treinamento, mais os dados das simulações dos modelos das duas redes RBF dos experimentos relacionados, onde uma delas foi obtida com a abordagem proposta e a outra com o procedimento supervisionado citado.

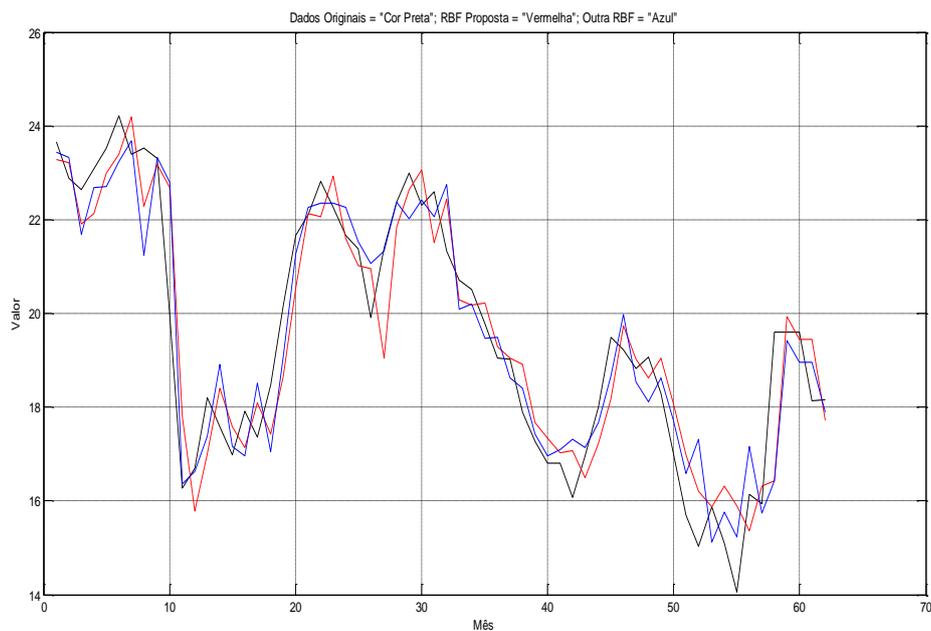


Figura 5.16 - Dados da série temporal e os valores aproximados pelas redes RBF.

Na Figura 5.17 têm-se os dados originais subsequentes (nos onze meses seguintes) da série temporal, que foram utilizados para testar a capacidade de estimação dos modelos considerados, mais os dados estimados pelas duas redes RBF das simulações relacionadas. Os valores de EPAM obtidos foram de 3,8% para a rede resultante do procedimento supervisionado citado, e 2,7% para a RBF obtida com a abordagem proposta (indicando neste exemplo, uma melhor modelagem com a rede resultante via metodologia sugerida neste trabalho). Nas previsões citadas se considerou previsões de um passo à frente. Para previsões de múltiplos passos seria importante empregar mais amostras de valores anteriores da série temporal considerada.

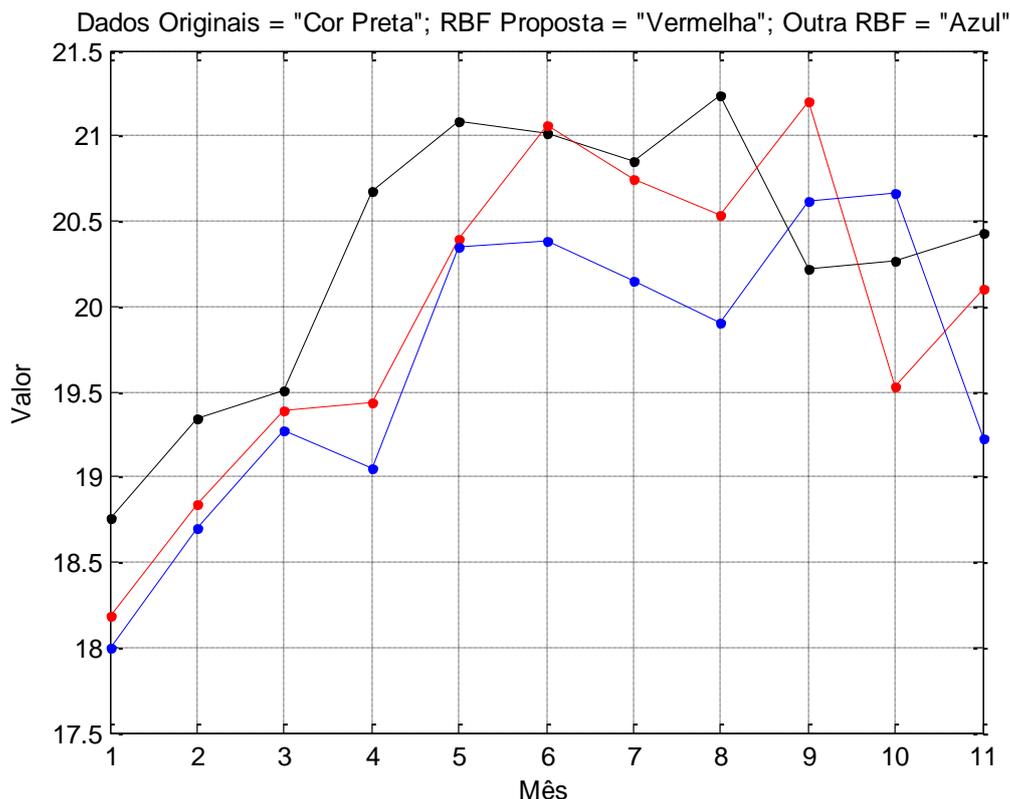


Figura 5.17 - Dados da série temporal e os valores estimados pelas redes consideradas.

5.4 Exemplo de Classificação de Padrões

Neste exemplo será utilizada uma rede RBF parametrizada com a abordagem proposta neste trabalho, objetivando uma aplicação de classificação de padrões.

5.4.1 Exemplo 5

Será considerada uma aplicação de classificação de dados relacionados ao diagnóstico de câncer de mama, cujas informações fazem parte do repositório UCI *Machine Learning* (Bache e Lichman, 2013), com o banco de dados correspondente compilado pela University of Wisconsin Hospitals com a coordenação do Dr. William H. Wolberg. As informações desta base de dados de acesso público (utilizado pela comunidade científica para análise de algoritmos de aprendizagem de máquina) são compostas por nove atributos: Clump Thickness; Uniformity of Cell Size; Uniformity of Cell Shape; Marginal Adhesion; Epithelial Cell Size; Bare Nuclei; Bland Chromatin; Normal Nucleoli; Mitoses. A classificação do diagnóstico é “0” para tumor benigno e “1” para maligno. A base de dados consta de 699 exemplos, onde 65,4% dos dados são

classificações referentes a tumores benignos e 34,6% de malignos. Algumas informações que apresentaram inconsistências foram removidas da base de dados, resultando em um conjunto com 683 amostras, sendo 341 amostras utilizadas para treinamento e outras 341 amostras para teste. As informações foram embaralhadas (selecionados aleatoriamente) a partir do banco de dados original. Os resultados foram comparados com os obtidos pelo método proposto por Noman et al. (2009).

A estrutura da rede RBF utilizada nesta simulação consiste de nove entradas referentes às informações dos atributos do modelo, tendo dois nós na camada escondida e a camada de saída informando a classificação correspondente. Os valores dos centros dos agrupamentos resultantes da aplicação do algoritmo Fuzzy C-Means estão indicados na Tabela 5.28 e constituem os dados dos centros das funções de base radial da rede.

Tabela 5.28 - Valores dos centros das funções da rede referente ao Exemplo 5.

c_{11}	0,678187
c_{12}	0,693594
c_{13}	0,689836
c_{14}	0,584668
c_{15}	0,536870
c_{16}	0,778231
c_{17}	0,615053
c_{18}	0,595280
c_{19}	0,268146
c_{21}	0,298546
c_{22}	0,131776
c_{23}	0,141380
c_{24}	0,136420
c_{25}	0,211951
c_{26}	0,135695
c_{27}	0,209271
c_{28}	0,125844
c_{29}	0,111176

A Tabela 5.29 indica os valores das larguras ajustados para a rede RBF considerada, onde se utilizou os dados de treinamento citados e a abordagem proposta. A Tabela 5.30 apresenta os valores dos pesos da camada de saída.

Tabela 5.29 - Valores das larguras das funções da rede referente ao Exemplo 5.

Parâmetros	Valores
σ_{11}	10,714678
σ_{12}	10,055280
σ_{13}	10,528070
σ_{14}	8,134845
σ_{15}	10,243011
σ_{16}	8,446178
σ_{17}	10,174372
σ_{18}	8,027342
σ_{19}	9,844451
σ_{21}	12,937149
σ_{22}	13,991633
σ_{23}	15,930176
σ_{24}	28,721439
σ_{25}	23,413578
σ_{26}	23,907565
σ_{27}	17,850148
σ_{28}	25,343032
σ_{29}	35,015549

Tabela 5.30 - Valores dos pesos da camada de saída da rede referente ao Exemplo 5.

w_0	25,392314
w_1	71,508524
w_2	-96,028134

A Tabela 5.31 mostra os valores referentes às etapas de treinamento/teste com o percentual de acerto na classificação dos dados, confrontando os resultados obtidos pela abordagem proposta neste trabalho e do artigo Noman et al. (2009). Pode-se verificar que o resultado obtido com a abordagem proposta resultou em uma classificação melhor em relação ao do artigo citado (que utilizou procedimentos supervisionados para ajustar os parâmetros de uma rede neural correspondente).

Tabela 5.31 - Comparação de resultados (em %).

Contexto	Noman et al. (2009)		Abordagem Proposta	
	Treino	Teste	Treino	Teste
Passos	10000	-	1	-
Classificação	97,65	71,77	97,36	96,77

Neste exemplo não foi obtida a matriz de confusão correspondente, e também não foi testado à porcentagem de falsos negativos e positivos nos testes de classificação. Estas informações serão tratadas em trabalho futuro como extensão do exemplo utilizado neste item.

Agradecemos a Felipe Andery Reis (REIS, 2014), mestre em Ciência e Tecnologia da Computação pela UNIFEI, pela disponibilização do programa utilizado no processamento dos parâmetros da rede RBF deste exemplo. Neste trabalho os parâmetros iniciais da rede em questão foram inicializados com a abordagem proposta nesta tese, e posteriormente os dados das larguras das funções da rede foram refinados via algoritmo de enxame de partículas.

CAPÍTULO 6

6. Conclusão

O procedimento proposto neste trabalho para o ajuste de parâmetros de redes RBF, se mostrou eficiente e com aplicação direta em contextos diversos de aplicações. Os resultados obtidos em exemplos de aproximações de funções, na modelagem de sistema dinâmico com características não linear, em previsão de série temporal e em classificação de padrões, sugerem um potencial promissor para aplicações diversas em áreas como reconhecimento de padrões, sistemas de controle, processamento de sinais e outras. As exatidões obtidas nas aproximações e estimativas realizadas nos exemplos deste trabalho indicam que a abordagem proposta consiste em uma alternativa interessante às estratégias não supervisionadas conhecidas para ajuste de parâmetros de redes RBF. Que também pode constituir em uma abordagem útil para a inicialização de valores de parâmetros de procedimentos supervisionados, onde geralmente os valores associados são inicializados de forma aleatória, constituindo uma alternativa que pode levar a melhores exatidões nos modelos resultantes e convergência mais rápida dos algoritmos utilizados.

Como contribuição original desta tese, podemos destacar o desenvolvimento de uma abordagem sistemática de ajuste de parâmetros de redes RBF que utiliza uma combinação de procedimentos não supervisionados, e uma proposta nova de parametrização dos dados relacionados com os valores das larguras das funções de base radial utilizadas na constituição de uma rede RBF para uma determinada aplicação prática.

A complexidade computacional envolvida na abordagem proposta nesta tese é reduzida em relação aos procedimentos supervisionados citados em textos da área. Como são utilizados procedimentos não supervisionados, a computação dos dados (relacionados com os ajustes de parâmetros de redes RBF) é realizada sem maiores dificuldades computacionais ou numéricas. O que pode ser constatado nas listagens que estão nos Anexos 2 e 3 no final da tese, referentes aos programas utilizados na obtenção

dos parâmetros e simulação da rede RBF utilizada no exemplo da modelagem do sistema de nível (com listagens similares para os outros exemplos apresentados).

No decorrer do trabalho desta tese foram obtidas três publicações citadas a seguir:

Sistema Inteligente para previsão de séries temporais: uma aplicação em previsão de carga em sistemas elétricos de potência. SANTOS, F.A.O; PINHEIRO, C.A.M; LIMA, B.I.L; ZAMBRONI, A.C.S; BARRETO, M.N; NOGUEIRA, D.A; SALGADO, R.M. XIX Congresso Brasileiro de Automática (CBA), Campina Grande, 2012.

Electric and plug-in hybrid vehicles recharge in distribution power systems. SANTOS, F.A.O; OLIVEIRA, D. Q; DELBONI, L.F.N.; DE SOUZA, A.C. Z.; LOPES, B.I.L. IEEE 11th International Conference on Environment and Electrical Engineering (EEEIC), p. 920-925, 2012.

Parameterization of RBF Neural Networks via Combination of Unsupervised Procedures and a New Way of Scaling Parameters. PINHEIRO, C. A. M. ; SANTOS, F. A. O. ; LIMA, I.; BALESTRASSI, P. P. Proceedings of the 19th International Conference on Circuits, Systems, Communications and Computers, Zakynthos Island. p. 469-475, 2015.

6.1 Sugestões para Trabalhos Futuros

Para trabalhos futuros são propostas as seguintes sugestões:

- Utilizar na primeira etapa do procedimento proposto para ajustes de parâmetros de redes RBF, uma abordagem para determinar o número de agrupamentos (clusters) mínimos para uma determinada aplicação, por exemplo, com a utilização de algoritmos de *silhueta fuzzy* (Campello e Hruschka, 2006) ou outros citados na literatura específica da área. Desta forma espera-se obter estruturas mais reduzidas para as redes projetadas.
- Utilizar o procedimento proposto nesta tese como ajuste inicial de parâmetros de uma rede RBF, que posteriormente (após a última etapa do procedimento), são refinados por técnicas de treinamento baseadas no método do gradiente descendente ou outras como algoritmos genéticos, por exemplo. Deste modo espera-se obter melhores exatidões nas redes resultantes.

Referências Bibliográficas

AGUIRRE, L. A. *Introdução à Identificação de Sistemas: Técnicas Lineares e Não Lineares Aplicadas a Sistemas Reais*. 3ª edição, Editora UFMG, 2007.

ASLANARGUN, A., MAMMADOV, M., YAZICI, B., E YOLACAN, S. *Comparison of ARIMA, neural networks and hybrid models in time series: tourist arrival forecasting*. Journal of Statistical Computation and Simulation, 77(1):29–53, Janeiro, 2007.

BABU GS, SURESH S. *Sequential Projection-Based Metacognitive Learning in a Radial Basis Function Network for Classification Problems*. IEEE Transactions on Neural Networks and Learning Systems 24(2):194-206, 2013.

BABUSKA, R. *Fuzzy Modeling for Control*. Kluwer Academic Publishers, 1998.

BACHE, K., LICHMAN, M. *UCI Machine Learning Repository* [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science, 2013.

BENOUDJIT, N., VERLEYSSEN, M. *On the kernel widths in Radial-Basis Function Networks*. Neural Processing Letters, n. 18, p. 139-154, 2003.

BISHOP, C. M. *Neural networks for pattern recognition*. Oxford University Press, Oxford, England, 1997.

BRAGA, A. P., CARVALHO, A. C. P. L. F.; LUDEMIR, T. B. *Redes Neurais Artificiais: Teoria e Aplicações*, LTC, 2000.

BEZDEK, J. C. *A convergence theorem for the fuzzy Isodata clustering algorithms*. IEEE Trans. on Pattern Analysis and Machine Intelligence, n. 1, pp. 1-8, 1980.

BILLINGS, S. A.; XONG, X. *Dual-orthogonal radial basis function networks for nonlinear time series prediction*. Neural Networks, Vol. 11, No. 3, p. 479-493, 1998.

BROOMHEAD, D. S.; LOWE, D. *Multivariable functional interpolation and adaptive networks*. Complex Systems, Vol. 2, No. 1, p. 321-355, 1988.

CAMPELLO R. J. G. B., HRUSCHKA E. R. *A fuzzy extension of the silhouette with criterion for cluster analysis*. Fuzzy Sets and Systems 157(21):2858-2875, 2006.

CHEN, S.; COWAN, C.; GRANT, P. *Orthogonal least squares learning algorithm for radial Basis function networks*, IEEE Transactions on Neural Networks, Vol. 2, No. 2, p. 302-309, 1991.

CHEN, J. Y., QIN, Z., JIA, J. *A PSO-based subtractive clustering technique for designing RBF neural networks*. IEEE World Congress on Computational Intelligence, pp. 2047-2052, 2008.

DE CASTRO, L. N. *Análise e síntese de estratégias de aprendizado para redes neurais artificiais*, Master's thesis, DCA/FEEC/UNICAMP - Campinas/SP/Brasil, 1998.

DING, S. *An optimizing method of rbf neural network based on genetic algorithm*. Neural Computing and Applications, Springer, v. 21, n. 2, p. 333-336, 2012.

FAHIMIFARD, S. M.; SALARPOUR, M.; SABOUHI, M.; SHIRZADY, S. *Application of anfis to agricultural economic variables forecasting. Case study: Poultry retail price*. Journal of Artificial Intelligence, v. 2, n. 2, p. 65-72, 2009.

FAUSTINO, C. P. *Previsão de Séries Temporais via Modelos Baseados em Regras*. Dissertação de Mestrado, Universidade Federal de Itajubá, 2011.

FAUSTINO, C. P.; PINHEIRO, C. A. M.; CARPINTEIRO, O. A.; LIMA, I. *Time series forecasting through rule-based models obtained via rough sets*. Artificial Intelligence Review, 2011.

FAUSTINO, C. P.; NOVAES, C. P.; PINHEIRO, C. A. M.; CARPINTEIRO, O. A. *Improving the performance of fuzzy rules-based forecasters through application of FCM algorithm*. Artificial Intelligence Review. v. 1, p. 10-24, 2012.

FATHI, V.; MONTAZER, G. A. *An improvement in rbf learning algorithm based on pso for real time applications*. Neurocomputing, Elsevier, v. 111, p. 169-176, 2013.

FRITZKE, B. *Fast learning with incremental RBF Networks*. Neural Processing Letters, v. 1, n.1, p.2-5, 1994.

GUERRA, F. A. *Análise de métodos de agrupamento para o treinamento de redes neurais de base radial aplicadas à identificação de sistemas*. Dissertação (Mestrado em

Engenharia de Produção e Sistemas) - Pontifícia Universidade Católica do Paraná, Curitiba, PR, 2006.

HAYKIN, S. *Redes Neurais: Princípios e Prática*. 2^a edição, Artmed Editora S.A., Porto Alegre – RS, 2007.

HUTCHINSON, J. M. *A radial basis function approach to financial time series analysis*. PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA, 1994.

JAFARNEJADSANI H, H, PIEPER J, EHLERS J. *Adaptive Control of a Variable-Speed Variable-Pitch Wind Turbine Using Radial-Basis Function Neural Network*. IEEE Transactions on Control Systems Technology 21(6):2264-2272, 2013.

JANG, J.-S. R. *Anfis: Adaptive-network-based fuzzy inference system*. IEEE Transaction on Systems, Man and Cybernetics, v. 23, n. 3, p. 665–685, 1993.

JANG, J.-S. R.; SUN, C.-T.; MIZUTANI, E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. 1st. ed., Prentice Hall, 1997.

JIN-YUE, L.; BAO-LING, Z. *Research on the non-linear function fitting of rbf neural network*. In: IEEE. Computational and Information Sciences (ICCIS), Fifth International Conference on. [S.l.], p. 842-845, 2013.

JUN, L.; ZUHUA, G. *Network traffic prediction using radial basis function neural network optimized by ant colony algorithm*. 2014.

KALYVAS, E. *Using Neural Networks and Genetic Algorithms to Predict Stock Market Returns*. Department of Computer Science, The University of Manchester, UK, 2001.

KEBRIAEI H, ARAABI BN, KIAN AR. *Short-Term Load Forecasting with a New Nonsymmetric Penalty Function*. IEEE Transactions on Power Systems 26(4):1817-1825, 2011.

KOSKO, B. *Neural Networks and Fuzzy Systems: A Dynamical Systems approach to Machine Intelligence*, Prentice Hall, 1992.

KUAN, C. M.; LIU, T. *Forecasting exchange rates using feedforward and recurrent neural networks*. Journal of Applied Econometrics, Vol. 10, No. 4, p. 347-364, 1995.

LI, B.; CONG, L.; ZHANG, W. *Research on optimized rbf neural network based on ga for sewage treatment*. In: IEEE. Intelligent Human-Machine Systems and Cybernetics (IHMSC), 5th International Conference on. [S.l.], v. 2, p. 520-523, 2013.

LI, S. *Study of personal credit evaluation method based on pso-rbf neural network model*. American Journal of Industrial and Business Management, Scientific Research Publishing, v. 3, p. 429, 2013.

LOWE, D. *Adaptive radial basis function nonlinearities, and the problem of generalization*. First IEEE International Conference on Artificial Neural Networks, p. 171-175, London, 1989.

MAHDI R. N.; ROUCHKA E.C. *Reduced Hyper BF Networks: Regularization by Explicit Complexity Reduction and Scaled Rprop-Based Training*. IEEE Transactions on Neural Networks 22(5):673-686, 2011.

MILONE, G.; ANGELINI, F. *Estatística Aplicada*. São Paulo: Atlas, 1995.

MING-BAO, P.; XIN-PING, Z. *Traffic flow prediction of chaos time series by using subtractive clustering for fuzzy neural network modeling*. Second International Symposium on Intelligent Information Technology Application, 2008.

MOHAMED, E. A.; MANSOUR, M. M.; EL-DEBEIKY, S.; MOHAMED, K. G. *Egyptian Unified Grid Hourly Load Forecasting Using Artificial Neural Network*, Electrical Power & Energy Systems, Vol. 20, No 7, pp. 495 – 500, 1998.

MOODY, J., DARKEN, C. J. *Fast learning in networks of locally-tuned processing units*. Neural Computation, n. 1, p. 281-295, 1989.

MOREIRA, D. A. *Administração da produção e operações*. 2.ed. São Paulo: Pioneira, 1996.

MORETTIN, P. A.; TOLOI, C. M. *Análise de Séries Temporais*. São Paulo: Edgard Blucher, 2006.

MULGREW, B. *Applying radial basis functions*. IEEE Signal Processing Magazine, Vol.13, No. 2, p. 50-65, 1996.

NARENDRA, K. S., PARTHASARATHY, K. *Identification and control of dynamical systems using neural networks*. IEEE Trans. on Neural Networks, n.1, pp. 4-27, 1990.

NELLES, O. *Nonlinear System Identification*. Springer-Verlag, Berlin, 2001.

NOMAN, S., SHAMSUDDIN, S. M., HASSANIEN, A. E. *Hybrid learning enhancement of RBF network with particle swarm optimization*. In *Foundations of Computational, Intelligence*, v. 1, pp. 381-397, Springer – Berlin, 2009.

OGATA, K. *Engenharia de Controle Moderno*. 5ª edição, Pearson Prentice Hall, São Paulo, 2010.

OLIVEIRA, D. Q.; SANTOS, F. A. O.; DELBONI, L. F. N.; DE SOUZA, A. C. Z.; LOPES, B. I. L. *Electric and plug-in hybrid vehicles recharge in distribution power systems*. IEEE 11th International Conference on Environment and Electrical Engineering (EEEIC), p. 920-925, 2012.

OLIVEIRA, J. V.; PEDRYCZ, W. et al. *Advances in fuzzy clustering and its applications*. [S.l.]: Wiley Online Library, 2007.

PAN, Y., XUE, W., ZHANG, Q., ZHAO, L. *A forecasting model of RBF neural network based on genetic algorithms optimization*. IEEE Seventh International Conference on Natural Computation, v. 1, pp. 48-51, 2011.

PEDRYCZ, W.; GOMIDE, F. A. C. *An Introduction to Fuzzy sets*. Analysis and Design, MIT Press, 1998.

PEDROZA, L. C. C. PEDREIRA, C. E. *Uma nova metodologia para treinamento em redes neurais multi-camadas*, *Controle & Automação* 11: 49–54, 2000.

QING-WEI, Z., ZHI-HAI, X., JIAN, W. *Prediction of electricity consumption based on genetic algorithm-RBF neural network*. IEEE 2nd International Conference on Advanced Computer Control v.5, pp. 339-342, 2010.

REIS, A. F. *Procedimento de Ajuste de Parâmetros de Redes RBF via PSO*. Dissertação de Mestrado, Universidade Federal de Itajubá, 2014.

SÁFADI, T.; NOGUEIRA, D. A. *Previsão de preços na bovinocultura de corte*. 2004. Disponível em

<http://revista.dae.ufla.br/index.php/ora/article/viewArticle/288>. Acesso em nov. 2012.

SAHA, A., KEELER, J. D. *Algorithms for better representation and faster learning in Radial Basis Function Networks*. Advances in Neural Information Processing Systems 2, edited by Davis S. Touretzky, p. 482-489, 1989.

SANTOS, F. A. O; BARRETO, M. N; NOGUEIRA, D. A; SALGADO, R. M; PINHEIRO, C. A. M; ZAMBRONI, A. C. S; LIMA, I. B. L. *Sistema Inteligente para Previsão de Séries Temporais: Uma Aplicação em Previsão de Carga em Sistemas Elétricos de Potência*. XIX Congresso Brasileiro de Automática, Campina Grande, PB, 2012.

SANTOS, F. A. O; OLIVEIRA, D. Q; DELBONI, L. F. N.; DE SOUZA, A. C. Z.; LOPES, B. I. L. *Electric and plug-in hybrid vehicles recharge in distribution power systems*. IEEE 11th International Conference on Environment and Electrical Engineering (EEEIC), p. 920-925, 2012.

SANTOS, F. A. O.; PINHEIRO, C. A. M.; LIMA, I.; BALESTRASSI, P. P. *Parameterization of RBF Neural Networks via Combination of Unsupervised Procedures and a New Way of Scaling Parameters*. Proceedings of the 19th International Conference on Circuits, Systems, Communications and Computers, Zakynthos Island. p. 469-475, 2015.

SERRÃO, F. C. C. *Modelo de previsão de carga de curto prazo utilizando redes neurais e lógica fuzzy*. Dissertação de Mestrado ao Programa de Pós-graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica da PUC-Rio. Universidade Católica, Brasil, Rio de Janeiro, 2003.

SHI Y, SHAN CH. *Comparison of Interpolating Functions and Interpolating Points in Full-Wave Multilevel Green's Function Interpolation Method*. IEEE Transactions on Antennas and Propagation 58(8):2691-2699, 2010.

SU SF, CHUANG CC, TAO CW, JENG JT, HSIAO CC. *Radial Basis Function Networks with Linear Interval Regression Weights for Symbolic Interval Data*. IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics 2(1):69-80, 2012.

SUN, Y. F. *Optimal partition algorithm of the RBF neural network and its application to financial time series forecasting*. Neural Comput. & Applic., 14: 36-44, 2005.

SYED-AHMAD, M. N; BENSENOUCI, A. *Short-term load forecasting using adaptive neuro-fuzzy inference system (anfis)*. Application to aleppo load demand, 2007.

TOPALLI, A. K.; ERKMEN, I. *A hybrid learning for neural networks applied to short term load forecasting*. Disponível em: www.ComputerScienceWeb.com. Acesso em: 12 abr, 2013.

TUBINO, D. F. *Manual de Planejamento e Controle da Produção*. 2.ed. São Paulo: Atlas S.A., 2000.

WU, J. D.; LIU, J. C. *A forecasting system for car fuel consumption using a radial basis function neural network*. *Expert Systems with Applications*, Elsevier, v. 39, n. 2, p. 1883-1888, 2012.

YAO, X. *Evolving artificial neural networks*. *Proceedings of IEEE*, Vol. 87, No. 9, p. 1423-1439, 1999.

YING, L. C.; PAN, M. C. *Using adaptive network based fuzzy inference system to forecast regional electricity loads*. *Energy Conversion and Management*, v. 49, p. 205–211, 2008.

ZADEH, L. A. *Fuzzy Sets*. *Information and Control*, n. 8, p. 338-353, 1965.

ZAMBRONI, A. C. S; PINHEIRO, C. A. M. *Introdução à Modelagem, Análise e Simulação de Sistemas Dinâmicos*. Editora Interciência Ltda, Rio de Janeiro, 2008.

ANEXO 1 - Modelos Neuro-Fuzzy

Jang et al. (1997) apresentaram uma estrutura neuro-fuzzy com a finalidade de gerar modelos *fuzzy* via procedimentos de treinamento através de redes neurais. A ideia básica consiste em realizar sistemas de inferência *fuzzy* (denominados FIS) sobre arquiteturas de redes neurais artificiais, caracterizando um sistema híbrido. O conceito consiste em utilizar paradigmas de aprendizado comuns às redes neurais para a geração de modelos difusos. Dessa forma, aproveitam-se os benefícios das redes neurais, que são apropriadas para a criação de modelos a partir de conhecimento implícito embutido em um conjunto de dados, e dos sistemas fuzzy, que são adequados para a criação de modelos a partir de um conhecimento explícito.

Dentre os vários sistemas neuro-fuzzy propostos, pode-se destacar o ANFIS (*Adaptive Network Based Fuzzy Inference System*), o FSOM (*Fuzzy Self-Organization Map*), o NEFClass (*Neuro Fuzzy Classification*), o NEFCON (*Neuro Fuzzy Control*), o NEFPROX (*Neuro Fuzzy Function Approximation*) e o DENFIS (*Dynamic Evolving Neural-Fuzzy Inference System*).

As referências citadas a seguir mostram algumas das aplicações realizadas com sucesso via sistemas neuro-fuzzy. Fahimifard et al. (2009) realizaram um estudo de desempenho entre métodos lineares e não lineares quando aplicados em processos de previsão. A análise dos resultados permitiu concluir que o ANFIS obteve um desempenho consideravelmente maior em todas as simulações realizadas em relação a modelos convencionais. Syed-Ahmad et al. (2007) e Ying e Pan (2008) utilizaram-se de modelos *fuzzy* para prever demanda de carga em um sistema elétrico. Foram consideradas observações horárias de consumo de energia elétrica para compor o conjunto de treinamento usado para treinamento do ANFIS. O modelo *fuzzy* obtido foi aplicado na previsão de demanda vinte e quatro horas à frente. Ying e Pan (2008) utilizaram observações anuais de 1981 até 1996 para previsão de carga para 1997 até 2000 em diferentes regiões de Taiwan. Em ambos os casos os modelos obtidos via ANFIS obtiveram resultados compatíveis com os modelos neurais usados para fins de comparação. Ming-Bao e Xin-Ping (2008) aplicaram um sistema híbrido para prever o fluxo de tráfego de automóveis. Os dados de treinamento foram agrupados por meio do algoritmo “Subtractive Clustering” e, com base nesse agrupamento, o processo de treinamento foi inicializado. A definição de um raio de agrupamento ótimo foi realizada por meio da aplicação de um algoritmo genético. Os resultados foram comparados aos

resultados obtidos via aplicação de modelos gerados por rede neural ou lógica difusa distintamente, onde o sistema híbrido apresentou melhores resultados. Faustino (2011) e Faustino et. al (2011, 2012), realizaram previsões de diferentes tipos de séries temporais utilizando modelos obtidos via ANFIS e por regras aproximadas (*rough rules*), fazendo comparações com modelos neurais e estatísticos. Em boa parte dos casos testados os melhores resultados foram obtidos com modelos difusos.

Adaptive Network-Based Fuzzy Inference System (ANFIS)

A Figura A.1 ilustra a arquitetura de um ANFIS composto por duas regras e que recebe duas variáveis de entrada mapeando cada uma delas para dois conjuntos nebulosos distintos.

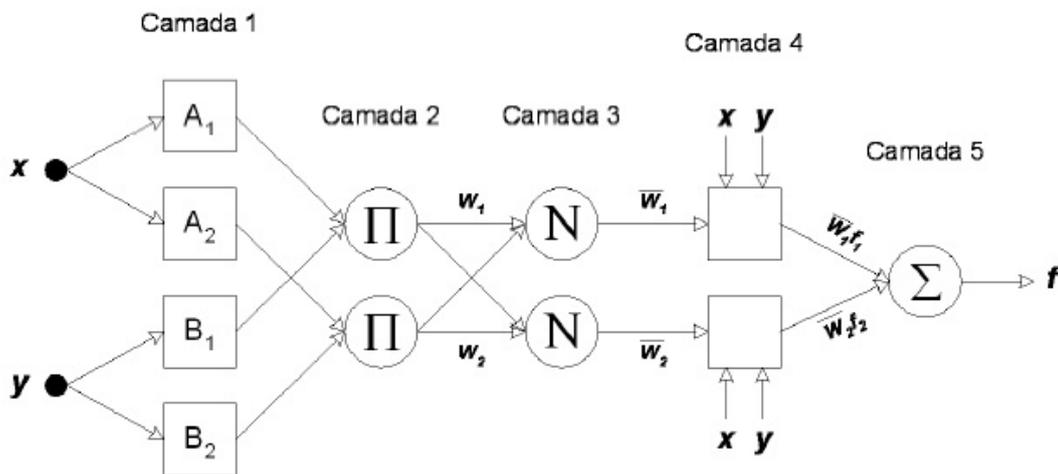


Figura A.0.1 - Arquitetura ANFIS de 5 camadas. Fonte: (Jang, 1993), (Faustino, 2011).

O modelo ANFIS exemplificado recebe as variáveis de entrada e, através de interações em suas camadas, realiza as etapas de fuzificação e defuzificação, resultando ao final no valor de saída correspondente ao padrão de entrada. Na primeira camada ocorre o mapeamento das variáveis de entrada do universo de discurso para o universo *fuzzy*, ou seja, é realizado o processo de fuzificação por meio da aplicação de uma função de pertinência. Os neurônios da primeira camada são todos do tipo adaptativo, sendo assim, os parâmetros, que nesse caso dizem respeito às constantes das funções de pertinência que são ajustadas via processo de treinamento. A segunda camada representa os antecedentes das regras fuzzy que compõem o FIS. O valor resultante (w_i , onde i representa o índice do neurônio) dos neurônios dessa camada é obtido por meio da aplicação de um operador t-norma como o mínimo ou produto, por

exemplo. A terceira camada é responsável pela normalização dos valores gerados na segunda camada. A saída dos neurônios que compõem essa camada indica o grau de ativação de uma regra em relação às outras.

Na quarta camada encontram-se as composições dos consequentes das regras, onde os neurônios dessa camada são todos do tipo adaptativo e, sendo assim, seus parâmetros serão determinados por meio de processo de treinamento. Por fim, a quinta camada finaliza o processo de defuzificação que consiste da totalização dos valores obtidos na quarta camada.

As seguintes abordagens podem ser utilizadas para ajustar as funções de pertinência das regras *fuzzy* e os coeficientes polinomiais associados a modelos TS:

- *Backpropagation*: propaga-se o erro por meio do cálculo do gradiente descendente para as camadas anteriores ajustando-se os parâmetros dos nós adaptativos da primeira e quarta camadas;
- *Backpropagation* + Método dos Mínimos Quadrados: aplicação do método *backpropagation* para ajustar os parâmetros dos neurônios da primeira camada e o método dos mínimos quadrados para ajustar os da quarta camada;
- *Backpropagation* + Método dos Mínimos Quadrados uma única vez: aplicação da técnica *backpropagation* para ajustar os parâmetros dos neurônios da primeira camada e o método dos mínimos quadrados para ajustar os da quarta camada somente na primeira interação.

ANEXO 2 – Listagem do Programa de Agrupamento de Dados

```

% Agrupamento de Dados via Fuzzy-Cmeans das Informações da Função ou
Sistema

clear all;

% Dados do Sistema/Função
load DadosNivel_Treino;
DadosFunc = DadosNivel_Treino;
x1=DadosFunc(:,1); x2=DadosFunc(:,2); y = DadosFunc(:,3);

% Aplicação do Algoritmo Fuzzy C-Means
dados = [x1 x2 y];
[centros,U,obj_fcn] = fcm(dados,6); % Definido 6 Agrupamentos
maxU = max(U);
index1 = find(U(1,:) == maxU);
index2 = find(U(2,:) == maxU);
index3 = find(U(3,:) == maxU);
index4 = find(U(4,:) == maxU);
index5 = find(U(5,:) == maxU);
index6 = find(U(6,:) == maxU);

% Indicações Gráficas dos Agrupamentos Resultantes
plot3(dados(index1,1),dados(index1,2),dados(index1,3),'r.','MarkerSize
',12); hold on;
plot3(dados(index2,1),dados(index2,2),dados(index2,3),'b.','MarkerSize
',12);
plot3(dados(index3,1),dados(index3,2),dados(index3,3),'g.','MarkerSize
',12);
plot3(dados(index4,1),dados(index4,2),dados(index4,3),'k.','MarkerSize
',12);
plot3(dados(index5,1),dados(index5,2),dados(index5,3),'y.','MarkerSize
',12);
plot3(dados(index6,1),dados(index6,2),dados(index6,3),'c.','MarkerSize
',12);
xlabel('x1'); ylabel('x2'); zlabel('y');
plot3(centros(:,1),centros(:,2),centros(:,3),'ko',...
'MarkerSize',12,'LineWidth',2);
grid; hold off;

% Valores dos Centros dos Clusters
c1a=centros(1,1); c2a=centros(1,2);
c1b=centros(2,1); c2b=centros(2,2);
c1c=centros(3,1); c2c=centros(3,2);
c1d=centros(4,1); c2d=centros(4,2);
c1e=centros(5,1); c2e=centros(5,2);
c1f=centros(6,1); c2f=centros(6,2);

% Cálculo dos Valores dos Desvios Padrões dos Dados dos Agrupamentos
Resultantes
d1an=std(dados(index1,1)); d2an=std(dados(index1,2));
d1bn=std(dados(index2,1)); d2bn=std(dados(index2,2));
d1cn=std(dados(index3,1)); d2cn=std(dados(index3,2));
d1dn=std(dados(index4,1)); d2dn=std(dados(index4,2));
d1en=std(dados(index5,1)); d2en=std(dados(index5,2));
d1fn=std(dados(index6,1)); d2fn=std(dados(index6,2));

```

```
% Escalonamento dos Valores dos Desvios Padrões Calculados
d1a=0.5*abs(max(dados(index1,1))-min(dados(index1,1)))/d1an;
d2a=0.5*abs(max(dados(index1,2))-min(dados(index1,2)))/d2an;
d1b=0.5*abs(max(dados(index2,1))-min(dados(index2,1)))/d1bn;
d2b=0.5*abs(max(dados(index2,2))-min(dados(index2,2)))/d2bn;
d1c=0.5*abs(max(dados(index3,1))-min(dados(index3,1)))/d1cn;
d2c=0.5*abs(max(dados(index3,2))-min(dados(index3,2)))/d2cn;
d1d=0.5*abs(max(dados(index4,1))-min(dados(index4,1)))/d1dn;
d2d=0.5*abs(max(dados(index4,2))-min(dados(index4,2)))/d2dn;
d1e=0.5*abs(max(dados(index5,1))-min(dados(index5,1)))/d1en;
d2e=0.5*abs(max(dados(index5,2))-min(dados(index5,2)))/d2en;
d1f=0.5*abs(max(dados(index6,1))-min(dados(index6,1)))/d1fn;
d2f=0.5*abs(max(dados(index6,2))-min(dados(index6,2)))/d2fn;

% Salvar Informações para o Programa de Cálculo dos Pesos de Saída da
Rede RBF e Simulação da Modelagem Resultante
DadosPar =
[c1a;c2a;c1b;c2b;c1c;c2c;c1d;c2d;c1e;c2e;c1f;c2f;d1a;d2a;d1b;d2b;d1c;d
2c;d1d;d2d;d1e;d2e;d1f;d2f];
save DadosPar;
```

ANEXO 3 – Listagem do Programa para Cálculo de Parâmetros de Rede RBF e Simulação Desta

```

% Obtenção dos Parâmetros da Rede RBF de Modelagem do Sistema/Função

clear all;

% Dados da Função ou Sistema
load DadosNivel_Treino;
DadosFunc = DadosNivel_Treino;
x1=DadosFunc(:,1); x2=DadosFunc(:,2); y=DadosFunc(:,3);

% Carregar valores dos Centros e Larguras das Funções Radiais obtidas
via Técnica de Agrupamento
load DadosPar;
c1a=DadosPar(1); c2a=DadosPar(2); c1b=DadosPar(3); c2b=DadosPar(4);
c1c=DadosPar(5); c2c=DadosPar(6);
c1d=DadosPar(7); c2d=DadosPar(8); c1e=DadosPar(9);
c2e=DadosPar(10); c1f=DadosPar(11); c2f=DadosPar(12);
d1a=DadosPar(13); d2a=DadosPar(14); d1b=DadosPar(15);
d2b=DadosPar(16); d1c=DadosPar(17); d2c=DadosPar(18);
d1d=DadosPar(19); d2d=DadosPar(20); d1e=DadosPar(21);
d2e=DadosPar(22); d1f=DadosPar(23); d2f=DadosPar(24);

fm = 10; % Fator de Escala Adicional dos Valores dos Desvios Padrões
d1a=fm*d1a; d2a=fm*d2a; d1b=fm*d1b; d2b=fm*d2b; d1c=fm*d1c;
d2c=fm*d2c;
d1d=fm*d1d; d2d=fm*d2d; d1e=fm*d1e; d2e=fm*d2e; d1f=fm*d1f;
d2f=fm*d2f;

% Funções de Base Radial da Rede Considerada
f1 = exp(-((x1-c1a).*(x1-c1a)/(2*d1a^2)+(x2-c2a).*(x2-
c2a)/(2*d2a^2)));
f2 = exp(-((x1-c1b).*(x1-c1b)/(2*d1b^2)+(x2-c2b).*(x2-
c2b)/(2*d2b^2)));
f3 = exp(-((x1-c1c).*(x1-c1c)/(2*d1c^2)+(x2-c2c).*(x2-
c2c)/(2*d2c^2)));
f4 = exp(-((x1-c1d).*(x1-c1d)/(2*d1d^2)+(x2-c2d).*(x2-
c2d)/(2*d2d^2)));
f5 = exp(-((x1-c1e).*(x1-c1e)/(2*d1e^2)+(x2-c2e).*(x2-
c2e)/(2*d2e^2)));
f6 = exp(-((x1-c1f).*(x1-c1f)/(2*d1f^2)+(x2-c2f).*(x2-
c2f)/(2*d2f^2)));

[zL,zC]=size(y);

% Regressores
Fi=[f1 f2 f3 f4 f5 f6 ones(1,zL)']; Y=y;

% Cálculo dos Pesos da camada de Saída da Rede
W = inv(Fi'*Fi)*Fi'*Y;
w1=W(1); w2=W(2); w3=W(3); w4=W(4); w5=W(5); w6=W(6); w0=W(7);

% Inicialização das Variáveis do Modelo
u1=0;y1=0;

for n=1:np-1 % Loop de Simulação

```

```

u = x1(n); % Entrada do Sistema

% Funções de Base Radial da Rede
f1n = exp(-((u-c1a)*(u-c1a)/(2*d1a^2)+(y1-c2a)*(y1-
c2a)/(2*d2a^2)));
f2n = exp(-((u-c1b)*(u-c1b)/(2*d1b^2)+(y1-c2b)*(y1-
c2b)/(2*d2b^2)));
f3n = exp(-((u-c1c)*(u-c1c)/(2*d1c^2)+(y1-c2c)*(y1-
c2c)/(2*d2c^2)));
f4n = exp(-((u-c1d)*(u-c1d)/(2*d1d^2)+(y1-c2d)*(y1-
c2d)/(2*d2d^2)));
f5n = exp(-((u-c1e)*(u-c1e)/(2*d1e^2)+(y1-c2e)*(y1-
c2e)/(2*d2e^2)));
f6n = exp(-((u-c1f)*(u-c1f)/(2*d1f^2)+(y1-c2f)*(y1-
c2f)/(2*d2f^2)));

% Simulação do Modelo da Rede RBF com os Dados de Treino
ye = w0 + w1*f1n + w2*f2n + w3*f3n +w4*f4n + w5*f5n + w6*f6n;

vy(n)=y1; % Armazenamento de Informações

u1=u; y1=ye; % Atualização de Variáveis

end

vn=1:1999;
% Gráficos
subplot(211);
plot(vn,x1,'k'); grid;
xlabel('Amostras'); ylabel('u');
title('Cor Preta = Dados Originais; Cor Azul = Modelo RBF (Dados
Treino)');
subplot(212);
plot(vn,y,'k',vn,vy,'b');grid
xlabel('Amostras'); ylabel('y');
figure

% Somatória do Erro Quadrático
er = y' - vy; eq = er.*er;
seq = 0.5*sum(eq)

% Dados da Função ou Sistema
load DadosNivel_Testes;
DadosFunc = DadosNivel_Testes;
x1=DadosFunc(:,1); x2=DadosFunc(:,2); y=DadosFunc(:,3);
[nL,nC]=size(y); Np=nL;

% Inicialização das Variáveis do Modelo
u1=0;y1=0;

for n=1:Np % Loop de Simulação

    u = x1(n); % Entrada do Sistema

    % Funções de Base Radial da Rede
    f1n = exp(-((u-c1a)*(u-c1a)/(2*d1a^2)+(y1-c2a)*(y1-
c2a)/(2*d2a^2)));

```

```

    f2n = exp(-((u-c1b)*(u-c1b)/(2*d1b^2)+(y1-c2b)*(y1-
c2b)/(2*d2b^2)));
    f3n = exp(-((u-c1c)*(u-c1c)/(2*d1c^2)+(y1-c2c)*(y1-
c2c)/(2*d2c^2)));
    f4n = exp(-((u-c1d)*(u-c1d)/(2*d1d^2)+(y1-c2d)*(y1-
c2d)/(2*d2d^2)));
    f5n = exp(-((u-c1e)*(u-c1e)/(2*d1e^2)+(y1-c2e)*(y1-
c2e)/(2*d2e^2)));
    f6n = exp(-((u-c1f)*(u-c1f)/(2*d1f^2)+(y1-c2f)*(y1-
c2f)/(2*d2f^2)));

    % Simulação do Modelo da Rede RBF com os Dados de Teste
    ye = w0 + w1*f1n + w2*f2n + w3*f3n +w4*f4n + w5*f5n + w6*f6n;

    vyt(n)=y1;    % Armazenamento de Informações

    u1=u; y1=ye; % Atualização de Variáveis

end

vn=1:999;
% Gráficos
subplot(211);
plot(vn,x1,'k'); grid;
xlabel('Amostras'); ylabel('u');
title('Cor Preta = Dados Originais; Cor Azul = Modelo RBF (Dados
Teste)');
subplot(212);
plot(vn,y', 'k',vn,vyt, 'b');grid
xlabel('Amostras'); ylabel('y');

% Somatória do Erro Quadrático
er = y' - vyt; eq = er.*er;
seq = 0.5*sum(eq)

```