DISSERTAÇÃO DE MESTRADO Nº 383

TREINAMENTO MULTI-OBJETIVO DE REDES NEURAIS RBF

Daniel Henrique Dominguete Carvalho

DATA DA DEFESA: 03/12/2004

Universidade Federal de Minas Gerais

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

TREINAMENTO MULTI-OBJETIVO DE REFES NEURAIS RBF

Daniel Henrique Dominguete Carvalho

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Mestre em Engenharia Elétrica

Orientador: Antônio de Pádua Braga

Co-orientador: Marcelo Azevedo Costa

Belo Horizonte - MG

Dezembro de 2004

"TREINAMENTO MULTI-OBJETIVO DE REDES NEURAIS RBF"

DANIEL HENRIQUE DOMINGUETE CARVALHO

Dissertação de Mestrado submetida à banca examinadora designada pelo Colegiado do Programa de Pós–Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do grau de *Mestre em Engenharia Elétrica.*

Aprovada em 03 de dezembro de 2004.

Por:

Antônio de Pádua Braga – Dr.
Prof. DELT/EEUFMG – orientador

Marcelo Azevedo Costa – Dr.
Prof. ICEX/UFMG – co –orientador

Luis Antônio Aguirre– Dr.
Prof. DELT/EEUFMG

Ricardo Hiroshi Caldeira Takahashi– Dr.
Prof. DMAT/UFMG

Petr Ekel– Dr.
Prof. PUC/MG

Agradecimentos

Primeiramente meu sincero agradecimento a Deus, que se faz presente em minha vida pela Sua infinita misericórdia.

Aos meus pais e familiares, pela presença, pela confiança, pelo incentivo, pelas orações e até mesmo, pelo apoio financeiro nos momentos difíceis.

À minha namorada Tatiane pela presença marcante em minha vida, pelo seu amor, carinho e dedicação.

Aos meus amigos e colegas do LITC Talles, Willian, Lane, Thiago, Cristiano, Eduardo, Mansur e Marcelo que nunca pouparam esforços em me ajudar e especialmente ao grande amigo Bernardo pela companhia durante toda a caminhada da graduação e pós-graduação.

Aos Professores Antônio de Pádua Braga e Marcelo Azevedo Costa pelo apoio técnico e moral, pela paciência, compreensão e tranqüilidade com que conduziram a orientação deste trabalho.

Aos meus amigos Sebastião, Antonicelli e Daniel pelo incentivo e harmoniosa convivência.

Aos colegas de trabalho da Arte & Byte que me apoiaram e incentivaram, pelo suporte financeiro, em especial Sanderson e Ricardo que acreditaram no meu trabalho.

Às inúmeras pessoas que participaram de alguma forma deste trabalho.

Aos meus pais

Pedro e Teresa

aos meus irmãos

Giselle e Pedro Paulo

familiares

e

Tatiane

com amor...

Senhor, vós me perscrutais e me conheceis, sabeis tudo de mim, quando me assento ou me levanto. De longe penetrais meus pensamentos, quando ando e quando repouso, vós me vêdes, observais todos os meus passos.

A palavra ainda me não chegou à língua, e já, Senhor, a conheceis toda. Vós me cercais por trás e pela frente, e estendeis sobre mim a vossa mão.

Conhecimento assim maravilhoso me ultrapassa, ele é tão sublime que não posso atingi-lo.

Para onde irei, longe de vosso Espírito? Para onde fugir, apartado de vosso olhar? Se subir até aos céus, ali estais; se descer à região dos mortos, lá vos encontrais também.

Se tomar as asas da aurora, se me fixar nos confins do mar, é ainda vossa mão que lá me levaria, e vossa destra que me sustentaria, se eu dissesse: "Pelo menos as trevas me ocultarão, e a noite, como se fôra luz, me há de envolver."

As próprias trevas não são escuras para vós, a noite vos é transparente como o dia, e a escuridão, clara como a luz. Fôstes vós que plasmastes as entranhas de meu corpo, vós me tecestes no seio de minha mãe.

Sêde bendito por me haverdes feito de modo tão maravilhoso, pelas vossas obras tão extraordinárias, conheceis até o fundo a minha alma. Nada de minha substância vos é oculto, quando fui formado ocultamente, quando fui tecido nas entranhas subterrâneas.

Cada uma de minhas ações vossos olhos viram, e todas elas foram escritas em vosso livro; cada dia de minha vida foi prefixado, desde antes que um só deles existisse.

Ó Deus, como são insondáveis para mim vossos desígnios, quão imenso é o número deles! Como os contar? São mais numerosos que a areia do mar; se pudesse chegar ao fim, seria ainda com vossa ajuda.

Perscrutai-me, Senhor, para conhecer meu coração; provai-me e conhecei meus pensamentos. Vêde se ando na senda do mal, e conduzi-me pelo caminho da eternidade.

Salmos 138:1-18,23-24

Resumo

Este trabalho apresenta uma nova proposta para treinamento de Redes de Função de Base Radial (RBF) que utiliza otimização multi-objetivo para encontrar soluções de alta capacidade de generalização.

A influência dos parâmetros livres das funções de base gaussianas na geração do conjunto de soluções eficientes foi analisada. Foi possível propor uma medida de complexidade para neurônios da camada escondida de redes RBF.

Variações para o algoritmo proposto são apresentadas, como a geração eficiente de soluções com otimização de raio e a geração de soluções com *Rigde Regression* e *Subset Selection*.

Alguns resultados utilizando o treinamento multi-objetivo para redes RBF e suas variações são apresentadas demonstrando a possibilidade de encontrar soluções de alta capacidade de generalização por meio do controle de complexidade.

Abstract

This work presents a new proposal for Radial Basis Function Networks (RBF) training that uses multi-objetive optimization to improve generalization performance.

The influence of free parameters of gaussian basis functions in the generation of a set of efficient solutions was analised. It was possible to propose a complexity measure for RBF hidden layer neurons.

Variations for the proposed algorithm are presented, as the efficient generation of solutions with radius optimization and generation of solutions with ridge regression and subset selection.

Some results using multi-objective RBF training and its variations were presented showing the possibility to find solutions with good generalization performance through model complexity control.

Sumário

Abreviações

As principais abreviações utilizadas nesta dissertação são listadas a seguir.

RNA	Redes Neurais Artificiais
MLP	Redes Neurais Multi-Camadas (*Multi-Layer Perceptrons*)
RBF	Redes de Base Radial (*Radial Basis Functions*)
SVM	Máquina de Vetores de Suporte (*Support Vector Machine*)
MOBJ	Algoritmo Multi-Objetivo
MOBJ-RBF	Algoritmo Multi-Objetivo para RBF
MOBJ-RBFr	Algoritmo Multi-Objetivo para RBF com otimização de raio
RR-MOBJ-RBF	Algoritmo Multi-Objetivo para RBF com *Ridge Regression*
SS-MOBJ-RBF	Algoritmo Multi-Objetivo para RBF com *Subset Selection*
GCV	Generalised Cross-Validation
BIC	Bayesian Information Criterion
MSE	Média do Somatório dos Erros Quadráticos

Lista de Símbolos

Neste trabalho, vetores são indicados por letras latinas minúsculas em negrito. Escalares são representados por letras minúsculas gregas ou latinas em itálico. Matrizes são representadas por letras maiúsculas em negrito. Símbolos específicos serão definidos *in loco*.

$\mathbf{H}$	Matriz de interpolação de redes RBF
$\mathbf{W}$	Matriz de pesos
$\mathbf{R}$	Matriz de raios das funções de base radial
$\mathbf{C}$	Matriz de centros das funções de base radial
$\mathbf{P}$	Matriz de projeção
$\mathbf{x}$	Vetor de entrada de uma RNA
$\mathbf{y}$	Vetor de saída real de uma RNA
$\mathbf{d}$	Vetor de saída desejada para uma RNA
$\mathbf{c}$	Vetor representativo de um centro de uma função radial
ϕ	Função de base radial
dist	Distância entre vetores
λ	Parâmetro de regularização
dim	Dimensão dos padrões de entrada
p	Número de padrões de entrada
h	Número de neurônios da camada escondida
k	Número de neurônios da camada de saída
e_T	Função de erro de treinamento
e_V	Função de erro de validação
e_G	Função de erro de generalização
J	Função de custo
$f_{rna}(.)$	Função representada por uma RNA
$f_g(.)$	Função geradora dos dados de treinamento
ξ	Função para critério de seleção de modelo
Γ	Conjunto de dados

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Nos dias atuais, as redes neurais artificiais (RNA) são aplicadas para a resolução de vários problemas de alta complexidade e de grande número de variáveis nas mais diversas áreas do conhecimento. Inserida na grande área de inteligência artificial, as redes neurais artificiais se destacam em aplicações de classificação e reconhecimento de padrões, aproximação de funções, controle de processos, processamento de sinais, entre outras.

As redes neurais artificiais podem ser interpretadas como um modelo computacional de processamento de informação inspirado no comportamento de neurônios biológicos. São constituídos de unidades de processamento intercomunicantes organizadas em camadas. O modo como são localizadas as conexões entre estas unidades (neurônios) e o processamento executado por cada uma definem as diversas topologias de RNA.

O sucesso de uma aplicação baseada em RNA, depende principalmente da sua fase de aprendizado, período em que a rede é treinada para executar uma determinada tarefa.

Um treinamento de redes neurais artificiais pode ser tratado como um problema de ajuste de superfície em um espaço de alta dimensionalidade. Esta dimensionalidade é dada pelo número de parâmetros livres da rede em questão. Deve-se então encontrar parâmetros caracterizadores desta superfície a fim de garantir a máxima capacidade de generalização de uma rede neural artificial. Entende-se por generalização a capacidade de uma rede neural produzir de forma satisfatória saídas para entradas que não foram apresentadas durante a fase de treinamento.

A capacidade de generalização se torna maior à medida em que a função aproximada por uma rede neural ($f_{rna}(\mathbf{x})$) representa melhor a função geradora

($f_g(\mathbf{x})$) dos dados de treinamento (Haykin 1994).

A topologia de uma RNA, definida pelo número de neurônios em cada camada e funções de ativação, o conjunto de padrões utilizados no treinamento e o método de ajuste de seus parâmetros livres são fatores que influenciam a capacidade de generalização.

Considerando um conjunto de treinamento adequado, ou seja, estatisticamente representativo, o ajuste dos valores dos parâmetros e a sua topologia definirão a capacidade de uma RNA fazer um mapeamento satisfatório do conjunto de treinamento e prover uma alta generalização.

Existem na literatura vários processos de aprendizagem destinados aos diversos tipos de RNA. Para cada processo de aprendizagem existem algoritmos de treinamento que visam a ajustar os parâmetros da rede de forma a realizar com sucesso a atividade destinada.

O texto visa a apresentar um estudo sobre fatores que influênciam as possíveis soluções de redes neurais de função de base radiais (Broomhead and Lowe 1988) (RBF), de maneira a caracterizar uma medida de complexidade para os neurônios da camada escondida.

É feita então uma proposta de treinamento de redes RBF para ajuste de pesos da camada escondida e parâmetros das funções de base, a partir de técnicas de otimização multi-objetivo para melhorar a generalização desta rede baseada em conceitos apresentados em trabalhos para treinamento multi-objetivo para redes MLP (Teixeira, Braga, Takahashi, and Saldanha 2000) (Costa, Braga, de Menezes, Parma, and Teixeira 2002).

Ao final do trabalho são realizados testes para validação da metodologia de treinamento proposta. Os resultados destes testes se mostraram satisfatórios demonstrando sua possível utilização para treinamento de redes RBF.

1.1 Motivação

Neste trabalho, foram desenvolvidas técnicas de treinamento de redes RBF como extensão do método multi-objetivo para redes MLP (McClelland and Rumelhart 1988).

As redes RBF são modelos constituídos normalmente por duas camadas, semelhantes aos modelos MLP. A principal diferença consiste na transformação dos padrões de entrada pela camada escondida. As redes MLP utilizam o produto escalar do vetor de entrada e do vetor de pesos como argumento da função de ativação sigmoidal dos neurônios de camadas escondidas. O processamento de informação para o caso de redes RBF é realizado por meio de funções radiais.

As redes MLP e RBF são teoricamente equivalentes por se tratarem de apro-

ximadores universais de funções. Entretanto, a distinção entre as funções de base utilizadas em cada modelo possibilitam algumas diferenças.

Cada função de base radial de uma rede RBF define um hiperelipsóide enquanto que os neurônios das redes MLP definem hiperplanos no espaço dos padrões de entrada. Desta forma, as redes RBF definem aproximadores locais, isto é, apenas certas regiões do espaço serão mapeadas. Já os aproximadores globais, gerados pelas redes MLP possuem maior capacidade de generalização para regiões onde não há dados de treinamento.

A aplicação dos conceitos de treinamento multi-objetivo para redes RBFs permitirá encontrar soluções de alta capacidade de generalização para aproximadores locais, sendo uma alternativa interessante para problemas que possuem dados com agrupamentos bem definidos.

1.2 Metodologia

No projeto de modelos neurais é desejável obter soluções de alta capacidade de generalização. Dado um conjunto de amostras de padrões de entrada estatisticamente representativo, a definição do número de parâmetros livres, bem como os seus valores afetam a qualidade de resposta destes modelos.

Quando se estabelecem topologias sub-dimensionadas para um dado problema, as soluções geradas por processos de treinamento possuem muita similaridade, caracterizando uma alta polarização (Geman, Bienenstock, and Doursat 1992). Soluções polarizadas possuem valores altos de erro para conjuntos de dados de treinamento e validação com baixa generalização. Desta forma, torna-se necessário aumentar a complexidade do modelo adicionando mais neurônios em camadas intermediárias. Entretanto, o incremento excessivo de complexidade pode provocar outro efeito indesejado, a alta variância. As soluções obtidas desta forma são capazes de se ajustar ao conjunto de dados de treinamento, alcançando o erro mínimo, modelando o ruído freqüentemente presente nos dados. Redes superdimensionadas possuem uma queda da capacidade de generalização.

O melhor modelo para um determinado problema deve ser capaz de fazer uso de um nível ideal de complexidade afim de equilibrar os efeitos de polarização e variância de sua solução.

A maioria dos métodos de ajuste dos parâmetros livres de redes RBFs são realizados em duas etapas: inicialmente procede-se com o ajuste dos centros e raios das funções radiais, através de aprendizados não-supervisionados e, em seguida, com o ajuste dos pesos da camada de saída, ao otimizar uma função de custo, como a soma do erro quadrático.

Nota-se que o equilíbrio entre polarização e variância não é abordado neste

tipo de procedimento, pois não se leva em consideração o controle de complexidade do modelo.

Existem diversos algoritmos para aprimorar a capacidade de generalização de redes MLP como: Weight-Decay (Hinton and Nowlan 1987), Early-Stopping (Weigend, Rumelhart, and Huberman 1990), Cross-Validation (Stone 1978) e Multi-Objetivo (Teixeira, Braga, Takahashi, and Saldanha 2000) (Costa, Braga, de Menezes, Parma, and Teixeira 2002) que não alteram a estrutura da rede e os métodos de poda (*pruning*) que utilizam modificações estruturais em redes previamente treinadas (Hassibi and Stork 1993). Algumas destas técnicas podem ser extendidas para o ajuste dos pesos da camada de saída de redes RBF devido à similaridade entre os neurônios de ambas as redes.

Dentre as principais técnicas para treinamento de redes RBF estão o *Ridge Regression* e o *Subset Selection* (Orr 1996). A primeira controla a magnitude dos pesos da camada de saída enquanto que a segunda faz uma busca do melhor modelo inserindo ou retirando neurônios do modelo inicial.

O algoritmo proposto neste trabalho e suas variações (Carvalho, Costa, and Braga 2004) propõe o controle da complexidade de redes RBF de forma a encontrar soluções de alta capacidade de generalização. Utilizam-se os conceitos de treinamentos multi-objetivo para a determinação de um conjunto de soluções eficientes segundo dois critérios: o erro e a complexidade.

A medida de complexidade utilizada para as redes MLP é representado pela norma do vetor de pesos de conexões. Devido à diferença entre os tipos de neurônios, as redes RBF terão duas medidas distintas de complexidade. A camada escondida será caracterizada pela norma da matriz de interpolação, uma vez que esta pode representar o comportamento das funções de base radiais. Para a camada de saída utiliza-se a mesma medida das redes MLP, a norma da matriz de pesos.

Pelo fato de não se obter uma relação bem definida entre as duas medidas de complexidade, cada quantidade será tratada de maneira independente.

O problema multi-objetivo representativo do treinamento de redes RBF fica então descrito pela minimização de três funções de custo: o erro de treinamento (e_T), a norma da matriz de interpolação ($\|\mathbf{W}\|$) e a norma da matriz de pesos ($\|\mathbf{H}\|$) segundo a Equação 1.1.

$$\psi^* = \arg_\psi \min \begin{cases} f_1(\psi) = e_T \\ f_2(\psi) = \|\mathbf{W}\| \\ f_3(\psi) = \|\mathbf{H}\| \end{cases} \tag{1.1}$$

Utilizando a otimização multi-objetivo pode-se gerar soluções de complexidades diferentes de forma que uma destas soluções tenha a complexidade adequada maximizando a capacidade de generalização.

As grandezas complexidade e erro são conflitantes, isto é, não é possível estabelecer uma mesma solução que represente simultaneamente o mínino para ambas. Logo a primeira etapa do treinamento consiste em encontrar um conjunto de soluções eficientes, denominado conjunto Pareto-ótimo (Pareto 1896), onde não é possível melhorar um dos objetivos sem que um outro seja degradado. Como por exemplo, não é possível reduzir o erro de treinamento sem que haja um incremento de complexidade, ou analogamente, não é possível diminuir a complexidade sem aumentar o valor de erro.

O método MOBJ-RBF é capaz de gerar soluções de vários níveis distintos de complexidade com valores mínimos de erro de treinamento, constituintes do conjunto de soluções eficientes. Uma vez obtidas as soluções do conjunto Pareto, o erro de ajuste para um conjunto de dados de validação é utilizado como critério de seleção da solução final. A solução que possuir o valor mais baixo de erro para o conjunto de validação representará a rede RBF de maior capacidade de generalização.

São apresentadas algumas variações do método MOBJ-RBF para a aceleração da busca de soluções eficientes e uma técnica de seleção de modelos para construção automática do modelo de complexidade mínima que garanta alta capacidade de generalização.

O método MOBJ-RBF abre uma linha de pesquisa para futuros trabalhos na área de generalização para redes RBF. Em comparação com outras técnicas de treinamento de redes RBF, as soluções encontradas pelo método proposto são superiores ou, no pior dos casos, equivalentes.

1.3 Organização do texto

No Capítulo 2 é realizada uma revisão dos principais conceitos e algoritmos de treinamento de redes neurais de base radial. São apresentados alguns algoritmos de treinamento em etapas separadas e alguns de etapa única. Ao final, são abordadas algumas técnicas avançadas para busca de soluções de alta generalização, tais como *Ridge Regression* e *Subset Selection*.

No Capítulo 3 são apresentados os conceitos gerais de otimização multi-objetivo bem como os métodos de treinamento multi-objetivo aplicados a redes MLP, por se tratar do ponto inicial da construção de um algoritmo multi-objetivo para redes RBF.

No Capítulo 4, se caracteriza o comportamento multi-objetivo para as redes RBF, bem como a sugestão de uma medida de complexidade para neurônios de base radial. Por fim, é apresentado o algoritmo multi-objetivo para treinamento de redes RBFs.

No Capítulo 5 são apresentadas algumas variações do método multi-objetivo

proposto. As metodologias apresentadas são capazes de gerar o conjunto de soluções eficientes através da otimização de raio e sua aproximação por regularização, bem como encontrar a topologia ideal para a solução de problemas adotando a técnica de *Subset Selection*.

No Capítulo 6, a eficiência das metodologias apresentadas são avaliadas em relação a outros algoritmos de treinamento de redes RBF. São utilizados dados provenientes de simulações de aproximação de funções bem como dados de problemas reais.

No Capítulo 7 são apresentadas as conclusões gerais bem como propostas para a continuidade do trabalho.

Em anexo, ao final do trabalho se encontra o apêndice, onde são apresentados alguns conceitos gerais sobre álgebra linear que foram utilizados no trabalho.

1.4 Conclusões do Capítulo

Neste capítulo foi apresentada uma visão geral da utilização e projeto de redes neurais enfatizando o problema inerente ao processo de treinamento que é o equilíbrio entre a polarização e a variância. Aborda-se a motivação de se estender a teoria de otimização multi-objetivo desenvolvida para redes MLP para o treinamento de redes RBF de forma a obter soluções de alta capacidade de generalização.

CAPÍTULO
2

Redes Neurais Artificiais de Função de Base Radial

Neste capítulo serão apresentados os conceitos básicos de redes neurais de função de base radial. É realizada uma revisão bibliográfica das principais metodologias de treinamento destas redes, bem como métodos de controle de generalização.

2.1 Introdução

As redes de função de base radial (RBF) (Broomhead and Lowe 1988) fazem parte da classe de aproximadores universais de funções multivariaveis (Girosi, Poggio, and Caprile 1991) (Hartman, Keeler, and Kowalski 1990). São amplamente utilizadas para a estimação não-paramétrica de funções multi-dimensionais a partir de um conjunto finito de realizações de um sistema físico.

Assim como em outras topologias de redes neurais artificiais multi-camadas (Haykin 1994), os neurônios presentes nas camadas ocultas representam um conjunto de funções não-lineares constituintes de uma base para um mapeamento não-linear dos vetores de entrada. A camada de saída realiza uma combinação linear das amostras de entrada submetidas a este mapeamento não-linear.

Desta forma, as redes RBFs são capazes de aproximar funções multivariadas a partir de uma combinação de funções de base radiais. Em sua versão mais usual, as aproximações são geradas a partir de uma combinação linear de funções gaussianas.

Segundo o teorema de Cover (Cover 1965), quanto maior a dimensão do

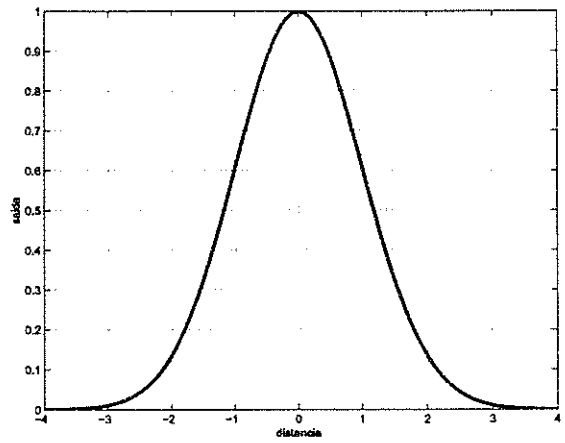

Figura 2.1: Função gaussiana.

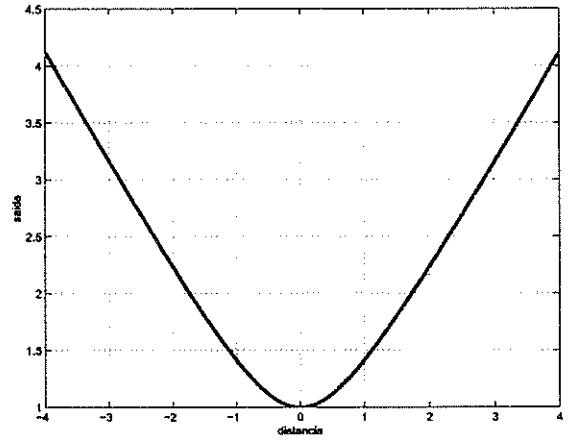

Figura 2.2: Função multiquadrática.

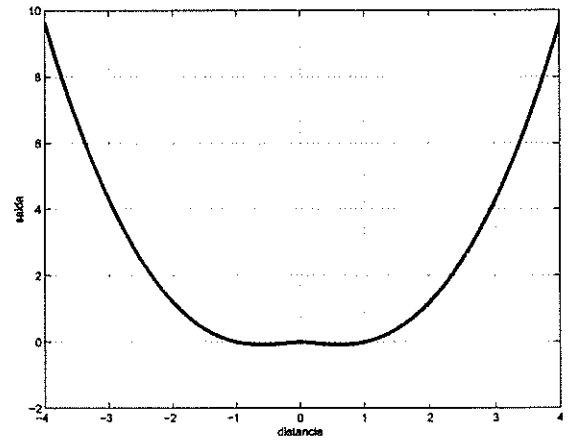

Figura 2.3: Função *thin-plate-spline*.

Em algumas aplicações, utiliza-se a distância de Mahalonobis (Fessant, Aknin, Oukhellou, and Midenet 2001) em lugar da distância euclidiana (Equação

2.4). Esta distância é função da matriz definida positiva $\mathrm{R_m}$ que representa um escalonamento de distância entre as dimensões de x e c.

$$\mathrm{dist} = (\mathbf{x} - \mathbf{c})^T \mathrm{R_m} (\mathbf{x} - \mathbf{c}) \tag{2.4}$$

Para redes RBF, cada função de base possui sua própria $\mathrm{R_j}$, normalmente definida como a inversa da matriz de covariância dos padrões de entrada relacionados a um dado centro $\mathrm{c_j}$.

Desta forma, existirá uma quantização diferente para cada dimensão do espaço, suprindo por exemplo, a necessidade de se estabelecer uma dispersão diferente para cada dimensão de uma função de base. A distância de Mahalonobis se torna idêntica à distância euclidiana se $\mathbf{R}_m$ for uma matriz identidade.

2.2 Arquitetura

As RBFs (Broomhead and Lowe 1988) possuem uma topologia similar as redes MLP (Rumelhart, Hinton, and Williams 1986). Ambas são redes multicamadas alimentadas adiante ("*feed-foward*"). A diferença principal é o modo como é feito o mapeamento entre a camada de entrada e a camada intermediária.

A Figura 2.4 representa um diagrama da topologia de uma rede RBF.

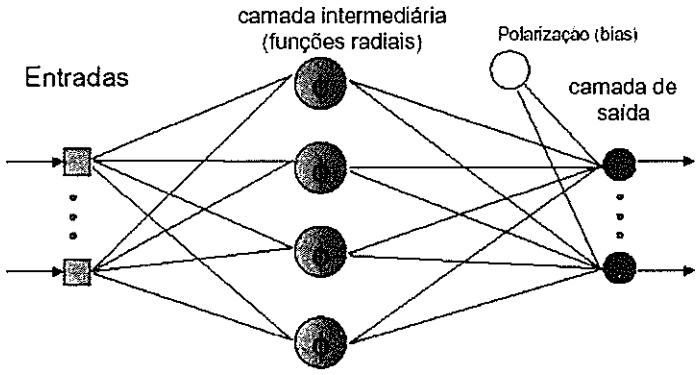

Figura 2.4: Exemplo de topologia para redes RBF.

Dado um vetor de dados de entrada x , a saída y_k de um neurônio de saída em uma rede RBF é dada pela Equação 2.5.

$$y_k = \sum_{i=1}^{h} w_{ik} \phi_i(\mathbf{x}, \mathbf{c}, \mathbf{r}) \tag{2.5}$$

O índice h é o número de neurônios na camada escondida e ϕ é uma função de base radial, definida em função do seu valor de raio (r) e centro (c). A função

de base mais utilizada em redes RBF é a função gaussiana, representada na Equação 2.6.

$$\phi_m(\mathbf{x_n}) = \mathbf{e}^{\frac{-(\|\mathbf{x_n}-\mathbf{c_m}\|)^2}{2\mathbf{r_m^2}}} \tag{2.6}$$

A matriz H, denominada de **matrix de interpolação**, é definida em (2.7) como representação da saída dos neurônios da camada escondida. Seus elementos são dados pelos valores das funções de base avaliadas de acordo com o valor do raio (r) e a distância euclidiana entre seu centro (c) e o vetor de entrada (x).

$$\mathbf{H} = \begin{pmatrix} \phi_1(\mathbf{x_1}) & \cdots & \phi_h(\mathbf{x_1}) \\ \vdots & & \vdots \\ \phi_1(\mathbf{x_p}) & \cdots & \phi_h(\mathbf{x_p}) \end{pmatrix} \tag{2.7}$$

A matriz W representa os valores de peso das conexões entre o(s) neurônio(s) da camada de saída e o(s) neurônio(s) da camada escondida.

$$\mathbf{W} = \begin{pmatrix} w_{11} & \cdots & w_{1k} \\ \vdots & & \vdots \\ w_{h1} & \cdots & w_{hk} \end{pmatrix} \tag{2.8}$$

A saída de uma rede RBF para um conjunto de dados de avaliação pode ser representada por um sistema linear segundo a Equação 2.9.

$$\mathbf{Y} = \mathbf{HW} \tag{2.9}$$

Os elementos da matriz H são definidos pelas funções ϕ e seus valores pertencem ao intervalo real [0,1], aproximando-se da unidade para casos onde existe um alto valor de raio ou a distância entre o padrão de entrada e o centro de uma função de base seja próximo de zero. Não existem, portanto, elementos negativos ou superiores à unidade na matriz H de saída de neurônios da camada escondida.

Ao realizar um treinamento para uma rede RBF, devem-se ajustar os parâmetros da camada escondida definindo o número de funções de base, seus centros, seus valores de raio e os pesos das conexões entre a camada escondida e a camada de saída.

2.3 Treinamento supervisionado de RNAs

As redes RBFs são utilizadas em problemas de classificação, aproximação de funções ou identificação de sistemas, onde se deseja estimar uma função a partir de amostras de entrada e saída de um dado sistema. A estimação é realizada pela determinação de parâmetros que não possuem significado físico diretamente relacionado ao problema.

No contexto de redes neurais artificiais, a estimação é denominada de treinamento supervisionado, onde a solução é representada por uma rede neural obtida a partir de ajustes de seus parâmetros livres.

Apesar de seus parâmetros, inicialmente, não representarem nenhuma informação física para o problema, existem formas de se extrair conhecimento do modelo gerado pela RBF (Teodorescu and Bonciu 1997).

Faz-se uso então de um conjunto de p amostras do sistema em questão denominado conjunto de treinamento (Γ), onde a cada padrão $\mathbf{x_i}$ de entrada do sistema é relacionado um valor de saída desejado $\mathbf{d_i}$.

$$\Gamma = \{\mathbf{x_i}, \mathbf{d_i}\}_{\mathbf{i=1}}^{\mathbf{P}} \tag{2.10}$$

Existem na literatura vários algoritmos de treinamento para processos de aprendizado supervisionado que visam à minimização do erro de treinamento. A Equação 2.11 apresenta uma função de custo representativa do erro de treinamento para um dado conjunto de amostras Γ.

$$e_T = 0,5. \sum_{i=1}^{p} [d_i - f_{rna}(\mathbf{x_i})]^2 \tag{2.11}$$

O algoritmo de retro-propagação (*"Backpropagation"*) (Rumelhart, Hinton, and Williams 1986) e suas variações utilizam a função de custo da Equação 2.11 a ser minimizada. Apesar de o erro de treinamento (e_T) ser uma aproximação do erro de generalização (Bengio 1996), sua minimização não garante soluções de alta capacidade de generalização.

Existem algoritmos de treinamento supervisionados que utilizam outras informações além do erro de treinamento para o ajuste de parâmetros livres de redes neurais visando minimizar o erro de generalização. Este trabalho apresenta uma nova metodologia para treinamento supervisionado de redes RBF que utiliza informações de complexidade para alcançar o equilíbrio entre a polarização e a variância (Geman, Bienenstock, and Doursat 1992). A seção 2.7.1 apresenta uma discussão mais detalhada sobre o equilíbrio entre a polarização e a variância e sua influência na capacidade de generalização de redes neurais artificiais.

2.4 Aprendizado de redes RBFs

O treinamento de redes RBFs constitui-se em um problema de otimização, onde se deseja encontrar um conjunto de parâmetros da rede que minimize um ou mais funcionais.

Os parâmetros a serem determinados são :

- O número de funções de base radiais, ou seja, o número de neurônios na camada intermediária;
- As posições dos centros das funções de base;
- Os valores de dispersão ou raios para cada função de base;
- As magnitudes dos pesos das conexões entre os neurônios da camada intermediária e a camada de saída.

O número de funções de base determina a precisão que se deseja alcançar com a aproximação. Quanto maior o número de neurônios maior será a dimensão do espaço de soluções dos parâmetros, exigindo um maior esforço computacional para o ajuste.

O ajuste de parâmetros pode ser estático, no sentido que mantém constante o número de neurônios, ou dinâmico quando se permite adicionar ou retirar neurônios durante seu ajuste.

O treinamento deste tipo de RNA pode ser realizado aplicando algoritmos distintos para cada conjunto de parâmetros ou aplicando qualquer técnica de programação não-linear que otimize todos os parâmetros do modelo.

2.5 Treinamento em três etapas

Como os centros, os raios e os pesos são de naturezas distintas, pode-se realizar o treinamento ajustando cada parâmetro separadamente através de um treinamento em três etapas.

2.5.1 Algoritmos para seleção de centros

A determinação do número de unidades radiais e suas posições é realizado com base na distribuição espacial do conjunto de vetores de treinamento. Cada função de base pode representar o centróide de um agrupamento de dados denominado de *cluster*.

O ajuste das posições dos centros pode ser realizado através de técnicas de agrupamento, onde se deseja agrupar padrões de características semelhantes

em sub-conjuntos de dados. Cada sub-conjunto possui um ponto que representa o centro do agrupamento, este ponto pode ser atribuído a um centro de uma função radial e a dispersão dos dados pode indicar o valor de raio para a mesma função de base.

Esses algoritmos poderão ser aplicados também a outros problemas como, por exemplo, para a redução de número de amostras, onde cada conjunto de pontos próximos será substituído pelo centro de um agrupamento.

Em sua formulação original, a RBF utiliza as posições dos padrões de entrada como sendo os centros das funções de base. No entanto, neste caso, se o número de padrões for grande, podem haver problemas de sobreajuste (*"overfitting"*) apresentando uma baixa generalização além de exigir um esforço computacional relativamente alto.

Uma alternativa para a seleção dos centros das funções da camada escondida pode ser realizada atribuindo-se aleatoriamente, para um dado número de funções de base, padrões do conjunto de entradas como centros das funções de base da rede RBF (Hassoun 1995). Algoritmos de agrupamento mais elaborados utilizam critérios para a seleção de centros que garantem uma distribuição mais representativa dos agrupamentos de dados. Dentre os algoritmos mais conhecidos se encontra o k-médias (*K-means*) (MacQueen 1967) e os mapas auto-organizativos de Kohonen (Kohonen 1982).

Algoritmo de Agrupamento - K-médias

O algoritmo de agrupamento mais comum é o k-médias (MacQueen 1967) onde a partir de um número h constante de agrupamentos, definem-se as posições dos seus centros iterativamente. O número de agrupamentos é definido anteriormente e não é alterado durante a execução do algoritmo.

Inicialmente são selecionados h pontos para as posições dos centros dos agrupamentos. A seguir, cada ponto é atribuído ao conjunto de dados mais próximo de modo que todos os pontos possuam um agrupamento correspondente.

Cada centro então é recalculado a partir das distâncias médias dos pontos pertencentes a um mesmo conjunto de dados. Novos centros são então atribuídos a cada agrupamento de acordo com a Equação 2.12.

$$\mathbf{c_i} = (1/Q_i) \sum_{Q_i} \mathbf{x}_q \tag{2.12}$$

O vetor $\mathbf{c_i}$ corresponde ao centro do agrupamento H_i , Q_i é o número de pontos pertencentes ao agrupamento H_i e $\mathbf{x}_q$ é o ponto q pertencente ao agrupamento H_i.

O ajuste do posicionamento de cada agrupamento se encerra caso a movi-

mentação seja inferior a um limite inicialmente proposto. Este algoritmo garante a minimização da soma quadrática das distâncias entre os h centros e os vetores de entrada.

$$J = \sum_{i=1}^{h} \sum_{Q_i} \|\mathbf{x_q} - \mathbf{c_i}\|^2 \tag{2.13}$$

Em (Bradley and Fayyad 1998) apresenta-se uma técnica para utilização em grandes massas de dados, aplicando-se k-médias em subconjuntos do conjunto total de treinamento.

Outras variações do algoritmo de k-médias são apresentados na literatura, como por exemplo, o *k-médias adaptativo* (Moody and Darken 1989a) e o *g-médias* (Hamerly and Elkan 2003).

Algoritmo de Agrupamento - Mapas auto-organizativos de Kohonen

Um outro método de agrupamento de dados é o algoritmo de mapas auto-organizativos de Kohonen (Kohonen 1982). Assim como o algoritmo de k-médias descrito anteriormente, sua função é de escolher a melhor posição para os centros de um número pré-definido de agrupamentos.

O passo inicial para o uso do algoritmo de Kohonen é definir o número h de agrupamentos (ou neurônios) que serão utilizados para o treinamento. A seguir são atribuídos posições aleatórias para os centros. Apresenta-se então todos os vetores de entrada do conjunto treinamento seqüencialmente à rede. Cada vetor de entrada será atribuído ao agrupamento de centro mais próximo. O centro escolhido será atualizado na direção do vetor de entrada atribuído a ele de acordo com a Equação 2.14, onde η é o parâmetro de taxa de aprendizado.

$$\mathbf{c_j} = \mathbf{c_j} + \eta(\mathbf{x_i} - \mathbf{c_j}) \tag{2.14}$$

Após algumas iterações, os h agrupamentos terão se deslocado afim de cobrir todo o espaço de distribuição de pontos agrupando os pontos mais próximos aos centros comuns.

Assim como o k-médias, este algoritmo possui a desvantagem de não escolher o número ideal de neurônios da camada escondida (ou centros) automaticamente. Um número inadequado de neurônios pode causar o efeito de *overfitting* ou *underfitting* para uma rede RBF. Cabe ao agente externo fazer um estudo sobre a distribuição dos vetores no espaço de entrada afim de estimar o melhor número de centros a serem distribuídos.

2.5.2 Algoritmos para seleção de raios

Para a escolha da dispersão das funções radiais, representada pelos parâmetros r, pode-se definir um valor único para todas as funções, atribuir um valor específico para cada nodo ou até um valor próprio para cada dimensão do vetor de entrada.

O ajuste dos raios possui uma característica importante que é a suavização da função estimada por uma RBF. Procura-se encontrar o compromisso entre localidade e suavidade da função gerada. A localidade ocorre quando se possui raios muito pequenos de forma que a camada escondida somente realize o mapeamento em regiões muito próximas dos centros das funções radiais. O aumento em excesso do valor de raio pode levar à construção de uma matriz de interpolação mal-condicionada, dificultando a determinação dos parâmetros livres de uma rede RBF.

A seguir serão apresentados algumas formulações presentes na literatura que fazem uma estimação de valores de raios para funções de base radiais. Outras abordagens podem ser encontradas em (Borş and Pitas 1994).

Formulação de Moody e Darken

Em (Moody and Darken 1989b), sugere-se a minimização da função de erro representada pela Equação 2.15 onde Q é um fator pré-definido de sobreposição de campos receptivos. Entende-se por campo receptivo a região ao redor do centro de uma função radial caracterizada pelo valor da dispersão (r) desta função.

$$E(r_1 \ldots r_c) = \frac{1}{2} \sum_{t=1}^{h} \left[\sum_{s=1}^{h} e^{-\left(\frac{\|c_s - c_t\|}{r_t}\right)^2} \left(\frac{\|c_s - c_t\|}{r_t} \right)^2 - Q \right]^2 \tag{2.15}$$

De acordo com Moody e Darken (Moody and Darken 1989a) o valor de raio para todos os neurônios é dado pela Equação 2.16.

$$r = (1/m) \sum_{j=1}^{m} \|c_j - c_{near}^{(j)}\| \tag{2.16}$$

O vetor $c_{near}^{(j)}$ é o centro mais próximo e m é o número de centros existentes ou um número de vizinhos mais próximos utilizados para o cálculo.

Formulações de Hausson

Haussoun, em (Hassoun 1995), propõe um valor distinto para cada neurônio da camada escondida através da Equação 2.17 onde Ψ_j é o conjunto de N vetores de entrada mais próximos do centro c_j.

$$r_j^2 = (1/N) \sum_{\mathbf{x}_i \in \Psi_j} \|c_j - \mathbf{x}_i\|^2 \tag{2.17}$$

Uma outra formulação também é proposta no mesmo documento:

$$r_j = \alpha \|c_j - c_{near}^{(j)}\| \tag{2.18}$$

onde $c_{near}^{(j)}$ é o centro mais próximo e o valor de α varia entre 1,0 e 1,5 .

Formulação de Haykin

Haykin (Haykin 1994) apresenta uma outra formulação para determinação do valor de raio para funções radiais.

$$r = \frac{d_{max}}{\sqrt{2M}} \tag{2.19}$$

onde M é o número de centros e d_{max} é a distância máxima entre estes centros.

Método Iterativo de Verleysen

No trabalho de Verleysen (Verleysen and Hlavácková 1994), sugere-se estabelecer o valor do raio segundo um ajuste iterativo a ser realizado paralelamente ao algoritmo de mapas auto-organizativos.

Primeiramente, inicializa-se os centros a partir de uma escolha de p padrões de treinamento. A cada iteração atualizam-se os valores de r segundo a Equação 2.20,

$$r(t+1) = (1 - \beta(t))r(t) + \beta(t)2\|x_i - c(t)\| \tag{2.20}$$

onde $\beta(t)$ é um fator de adaptação ($0 < \beta < 1$) que pode ser o mesmo que a taxa de atualização (η) dos mapas auto-organizativos.

Método de Fator de Escala

Para o método de fator de escala (Benoudijit, Archambeau, Lendasse, Lee, and Verleysen 2002), inicialmente calcula-se o desvio padrão para cada agrupamento de dados, a seguir determina-se o fator de escala q comum a todos os centros que garanta uma boa suavização da função aproximativa

$$r_j' = qr_j. \tag{2.21}$$

O fator de escala q é dependente da função geradora, da dimensão do espaço de entrada e da distribuição dos dados. Escolhe-se q heuristicamente,

avaliando o erro de conjunto de validação para vários valores até que se encontre o mínimo.

Ajuste Dinâmico de Deteriorização

Em (Berthold and Diamond 1995) é apresentado o algoritmo de Ajuste Dinâmico de Deteriorização (*Dynamic Decay Adjustment*) para definição de dispersão de funções de base radiais, através de um treinamento supervisionado de classificação de padrões. Trabalha-se com valores de *thresholds*, que podem ser convertidos em valores de raios, para a determinação de sobreposição dos campos receptivos.

2.5.3 Algoritmos para determinação de pesos

Como a camada de saída possui neurônios de função de ativação linear, os métodos aplicados a redes *Adaline* (Widrow and Hoff 1960), bem como métodos de solução de sistemas lineares podem ser aplicados para esta camada.

Por se tratar de um problema linear, a determinação dos pesos da camada de saída garante encontrar o mínimo global. A grande dificuldade encontrada para a solução de problemas lineares é o mal-condicionamento do sistema de equações representado pela Equação 2.9.

A seguir serão apresentadas as princípais metodologias para determinação de pesos para a camada de saída de redes RBFs.

Método dos Mínimos Quadrados - Método da pseudo-inversa

O método dos mínimos quadrados tem como objetivo ajustar os pesos W de camada de saída em função de um conjunto de treinamento $\Gamma = (\mathbf{x}_i, d_i)_{i=1}^{p}$ através de um treinamento supervisionado.

As saídas dos neurônios da camada escondida pode ser caracterizada pela *matriz de interpolação* (Equação 2.7) e a *matrix de transformação linear* (Equação 2.8) representa os pesos dos neurônios de saída. A *matriz de saídas desejadas* (**D**) possui todos valores desejados de acordo com o conjunto de dados de treinamento Γ

$$\mathrm{D} = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1k} \\ d_{21} & d_{22} & \dots & d_{2k} \\ \vdots & \vdots & \dots & \vdots \\ d_{p1} & d_{p2} & \dots & d_{pk} \end{pmatrix}. \tag{2.22}$$

Pela equação 2.23 pode-se encontrar o valor ótimo de W pelo cálculo da pseudo-inversa da *matriz de interpolação* conforme a Equação 2.24.

$$\mathbf{HW} = \mathbf{D} \tag{2.23}$$

$$\mathbf{W} = \mathbf{H}^{+}\mathbf{D} \tag{2.24}$$

onde

$$\mathbf{H}^{+} = \begin{cases} (\mathbf{H}^{\mathbf{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathbf{T}} & h > p \\ (\mathbf{H})^{-1} & h = p \\ \mathbf{H}^{\mathbf{T}}(\mathbf{H}\mathbf{H}^{\mathbf{T}})^{-1} & h < p \end{cases} \tag{2.25}$$

Vale lembrar que H é a matriz de interpolação representativa das saídas dos neurônios da camada escondida e W é a matriz de pesos dos neurônios da camada de saída.

Existem outros modos de solucionar a Equação 2.23 através da decomposição em valores singulares (Leon 1994) (Golub and Reinsch 1970) evitando possíveis problemas com o mal-condicionamento da matriz H.

Método dos Mínimos Quadrados Adaptativo

O método dos mínimos quadrados adaptativo é também conhecido como *regra Delta* ou *regra de Adaline* (Widrow and Hoff 1960). Este algoritmo ajusta os parâmetros livres de forma a minimizar a função de erro representada pela Equação 2.26.

$$E(x) = (1/2)(y(x) - d(x))^2 \tag{2.26}$$

A busca da solução é realizada na direção que minimiza o valor da função de erro, de forma que os parâmetros da solução (**W**) são ajustados no sentido oposto ao vetor de gradiente da superfície de erro segundo a Equação 2.27.

$$\mathbf{W}_t = \mathbf{W}_{t-1} - \eta \nabla E_{t-1} \tag{2.27}$$

onde η é o valor de taxa de aprendizado, onde se define o tamanho do *passo* de ajuste nos parâmetros livres.

O método é interativo e os ajustes dos pesos são feitos até que se alcance alguma condição de parada como, por exemplo, o número máximo de iterações ou gradiente nulo.

O cálculo do gradiente do erro é calculado em relação ao parâmetro peso, e $\phi_s(\mathrm{x})$ representa a função de transferência da camada escondida, neste caso, as funções de base radiais.

$$\frac{\partial E}{\partial w_s} = e.\phi_s(\mathrm{x}) \tag{2.28}$$

$$e_k = y_k - d_k \tag{2.29}$$

A equação de atualização dos pesos fica então como:

$$w_{ij}^t = w_{ij}^{t-1} - \eta e_k \phi_h(x) \tag{2.30}$$

Os pesos são atualizados a cada iteração do algoritmo até que se atinja algum critério de parada.

Método dos Mínimos Quadrados Ortogonal

Com o objetivo de se evitar o mal-condicionamento da matriz **H**, dificultando o cálculo de sua inversa, o método de mínimos quadráticos ortogonal oferece um metodologia de cálculo da solução da Equação 2.23 através da decomposição da matriz de interpolação em componentes ortogonais.

O mal-condicionamento ocorre pela presença de dependência linear entre os vetores de **H** ocasionados, por exemplo, pela proximidade dos centros das funções de base ou pelo excesso de valor de dispersão (raio) que fazem com que diversos padrões sejam mapeados para um mesmo ponto na saída da camada intermediária.

A determinação dos pesos da camada de saída das redes RBFs é um problema de regressão linear, conforme a Equação 2.31. O sinal de erro $e(t)$ é assumido como não correlacionado com as variáveis regressoras $h_i(t)$

$$y(t) = \sum_{i=1}^{C} w_i h_i(t) + e(t). \tag{2.31}$$

A Equação 2.31 pode ser reescrita como

$$\mathbf{Y} = \mathbf{HW} + \mathbf{E} \tag{2.32}$$

O método de mínimos quadrados ortogonais transforma os vetores constituintes da matriz **H** em vetores ortogonais. A Equação 2.33 representa esta transformação, onde **A** é uma matriz triangular de valores unitários na diagonal principal e valores nulos abaixo dela

$$\mathbf{H} = \mathbf{QA} \tag{2.33}$$

$$\mathbf{A} = \begin{pmatrix} 1 & \alpha_{1,2} & \dots & \alpha_{1,c-1} & \alpha_{1,c} \\ 0 & 1 & \dots & \alpha_{2,c-1} & \alpha_{2,c} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & \alpha_{c-1,c} \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \tag{2.34}$$

e $\mathbf{Q}$ é uma matriz de colunas ortogonais que

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{R} \tag{2.35}$$

onde $\mathbf{R}$ é uma matriz diagonal de elementos

$$r_i = q_i^T q_i \tag{2.36}$$

Os vetores de $\mathbf{Q}$ representam o mesmo espaço que $\mathbf{H}$, sendo possível encontrar $\mathbf{G}^*$ (Equação 2.38) que é solução da Equação 2.37.

$$\mathbf{Y} = \mathbf{Q}\mathbf{G} + \mathbf{E} \tag{2.37}$$

$$\mathbf{G}^* = \mathbf{H}^{-1}\mathbf{Q}^T\mathbf{Y} \tag{2.38}$$

A solução $\mathbf{G}^*$ pode ser convertida para o problema original segundo a Equação 2.39.

$$\mathbf{A}\mathbf{W}^* = \mathbf{G}^* \tag{2.39}$$

O método de Gram-Schmidt (Bjork 1967) pode ser aplicado ao problema representado pela Equação 2.39 para determinação de $\mathbf{W}^*$.

2.6 Treinamentos em etapa única

Como foi visto anteriormente, pode-se utilizar um treinamento em etapas distintas para se definir os parâmetros livres de uma rede RBF. No entanto, existem metodologias para um ajuste simultâneo de todos os parametros. A seguir serão apresentados, de maneira resumida, algumas metodologias de aprendizado em etapa única para redes RBFs.

2.6.1 Aprendizado por Retro-Propagação

A técnica clássica de gradiente descedente pode ser aplicada para todos os parâmetros, desde que todos os parâmetros sejam continuamente deriváveis em todo a sua extensão.

Seja μ um parâmetro qualquer de uma rede RBF, o seu valor será reajustado de acordo com o valor de atualização $\Delta\mu$ segundo a regra de aprendizado representada pela Equação 2.40 onde η é a taxa de aprendizado.

$$\Delta\mu = -\eta\frac{\partial e_T}{\partial \mu} \tag{2.40}$$

As Equações 2.41, 2.42 e 2.43 apresentam as derivadas relativas para uma rede RBF com funções de base gaussianas.

$$\frac{\partial e_T}{\partial w_{kj}} = (d_k^n - y_k(\mathbf{x}^n))\phi_j(\mathbf{x}^n) \tag{2.41}$$

$$\frac{\partial e_T}{\partial c_j} = \phi_j(\mathbf{x}^n)\frac{\|x^n - c_j\|}{r_j^2}\sum_k (y_k(t_k^n) - \mathbf{x}^n)w_{kj} \tag{2.42}$$

$$\frac{\partial e_T}{\partial r_j} = \phi_j(\mathbf{x}^n)\frac{\|x^n - c_j\|^2}{r_j^3}\sum_k (y_k(t_k^n) - \mathbf{x}^n)w_{kj} \tag{2.43}$$

Outras técnicas de otimização, como *gradiente conjugado* (Fletcher and Reeves 1964) , são baseadas em gradiente descendente e podem ser aplicadas para o treinamento de redes RBFs. Estes métodos são sensíveis às condições iniciais de treinamento, porém pode-se partir de pontos definidos pelos treinamentos não-supervisionados, favorecendo o encontro do mínimo global da função de erro.

2.6.2 *Aprendizado por Vetores de Suporte*

As máquinas de vetores de suporte (Vapnik 1995) foram inicialmente criadas para a solução de problemas de classificação. Cada classe se distingüe de outra através de um plano de separação determinado pela maximização de uma margem de separação. A margem é formada pela distância entre o plano de separação e os pontos mais proximos a ele. Quanto maior a margem de separação entre as classes, maior é a generalização. Os pontos localizados na margem são os denominados *vetores de suporte.*

A abordagem SVM pode ser aplicada para problemas não-lineares. Realiza-se um mapeamento não-linear, transformando os vetores de entrada em elementos de um novo espaço de características através de funções de mapeamento não-lineares. As funções de kernel realizam este mapeamento não-linear de forma que o conjunto de dados de entrada passa a ser linearmente separável e os planos de separação possam ser aplicados.

Uma importante função de kernel é a função gaussiana. A superfície de separação, então, passa a ser uma combinação linear de funções gaussianas representadas pelos vetores de suporte. Desta forma, a SVM se reduz a uma

rede RBF de centros automaticamente selecionados (Scholkopf, Sung, Burges, Girosi, Niyogi, Poggio, and Vapnik 1997). As metodologias utilizadas para a estimação de parâmetros de SVMs com funções de kernel gaussianas podem ser analogamente estendidas para redes RBFs.

2.6.3 *Aprendizado por Algoritmos Genéticos*

A computação evolucionária (Goldberg 1989) pode ser aplicada ao treinamento de redes RBFs (Whitehead and Chaote 1994). Realiza-se uma estratégia de busca para problemas de otimização, baseando-se nos princípios da teoria da evolução das espécies. Para uma dada função objetivo, denominada de função de *fitness*, busca-se uma solução que a minimiza através de operações genéticas entre os indivíduos de uma população.

Cada época de ajuste de parâmetros representa uma população de possíveis soluções, estas representadas por cada indivíduo. Este método pode ser aplicado para a definição de todos os parâmetros de uma rede RBF, desde que se caracterizem funções de *fitness* para cada um.

Em (Chen, Wu, and Luk 1999) realiza-se um treinamento de redes RBFs, onde se definem os parâmetros de regularização e raios através de algoritmos genéticos e Mínimos Quadráticos Ortogonal Regularizado para treinamento com regularização dos pesos da camada de saída.

2.7 *Seleção de modelos*

O objetivo principal de um treinamento de RNAs não é encontrar o mapeamento exato dos dados de treinamento mas modelar o processo responsável pela geração dos dados, ou em outras palavras, encontrar uma aproximação para a função geradora dos dados. Desta forma, o modelo será capaz de avaliar com sucesso padrões que não foram utilizados durante a fase de treinamento. Uma rede neural que possui respostas satisfatórias para padrões de entrada *inéditos* possui uma alta capacidade de generalização (Haykin 1994).

O problema de encontrar um modelo neural de alta capacidade de generalização é resolvido encontrando a complexidade ideal de um modelo que se adeque ao problema em questão. Desta forma, o treinamento de RNA não se concentra somente em minimizar o erro de treinamento mas otimizar a complexidade do modelo garantindo que a rede se comporte satisfatoriamente com diversos padrões de entrada.

A seleção de modelos é a tarefa de determinar o modelo de complexidade ótima para um dado problema de forma que se obtenham soluções de alta capacidade de generalização.

Partindo do princípio de que entradas semelhantes devem possuir saídas semelhantes, a função gerada por uma rede neural deve ser a mais suave possível. Onde a suavidade da função gerada por uma rede neural é representada pela sua complexidade dada pelo número e magnitude dos seus parâmetros livres.

São utilizados vários critérios para seleção de modelos, bem como, diversas metodologias para estimação de parâmetros e medidas de complexidade. Dentre os principais métodos, destacam-se a regularização ("*Ridge Regression*") e a seleção de sub-conjuntos ("*Subset Selection*") (Orr 1996). As abordagens multi-objetivo também se enquadram neste contexto e serão estudados mais detalhadamente no capítulo 3.

2.7.1 Polarização e Variância

Um treinamento de redes neurais artificiais pode ser tratado como um problema de ajuste de superfície em um espaço de alta dimensão. Esta dimensão é dada pelo número de parâmetros livres da rede em questão.

Busca-se um equilíbrio entre a complexidade do modelo e a complexidade exigida pelo problema, de forma que a construção da superfície seja capaz de criar um mapeamento suave que minimize o erro de generalização. Redes de complexidade superior à exigida são capazes de um ajuste fiel aos dados, porém, quase sempre o conjunto de dados possui ruídos que não são desejáveis ao mapeamento. Em um outro extremo, redes de complexidade inferior à exigida não são capazes de fazer um mapeamento satisfatório, gerando um mapeamento polarizado.

Existem na literatura algumas medidas de complexidade para modelos neurais, dentre eles o *VC-dimension* (Vapnik and Chervonenkis 1971), o número de derivadas contínuas (Girosi, Jones, and Poggio 1978) e a magnitude da transformada de Fourier (Barron 1993). No trabalho atual, bem como nos algoritmos de treinamento multi-objetivo, utiliza-se a norma euclidiana de uma matriz representativa dos parâmetros livres, como por exemplo, a norma da matriz de pesos.

Redes com complexidade superior ao problema em questão resultam em um sobreajuste (*overfitting*) do modelo. As soluções obtidas por estas redes sobreajustadas possuem alta variância, ou seja, possuem grande variabilidade de soluções, pois o elevado número de parâmetros livres resulta em uma maior flexibilidade no ajuste da função. Neste caso, as funções geradas são menos suaves e com um baixo erro de ajuste, como ilustra a Figura 2.5.

No caso análogo, redes de complexidade inferior ao problema proporcionam um sub-ajuste (*underfitting*) do modelo. As soluções obtidas por estas redes serão semelhantes para várias realizações do conjunto treinamento, pos-

suindo alta polarização. São soluções com um alto erro de ajuste e excessivamente suaves como ilustra a Figura 2.6.

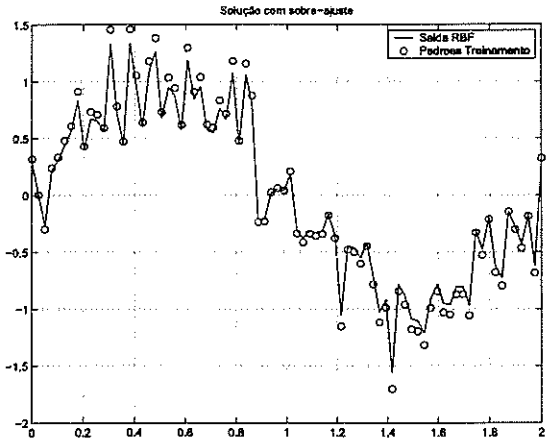

Figura 2.5: Solução de uma RBF para aproximação de função seno com sobre-ajuste.

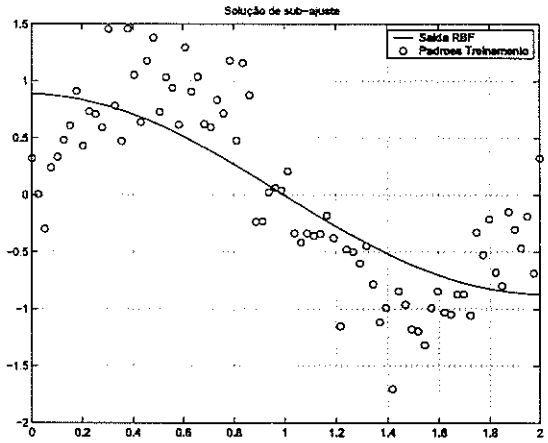

Figura 2.6: Solução de uma RBF para aproximação de função seno com sub-ajuste.

Uma solução com alta capacidade de generalização (Geman, Bienenstock, and Doursat 1992) pode ser considerada como uma solução que equilibra os efeitos de polarização e variância obtendo um ajuste ideal para o conjunto de dados de treinamento como ilustra a Figura 2.7.

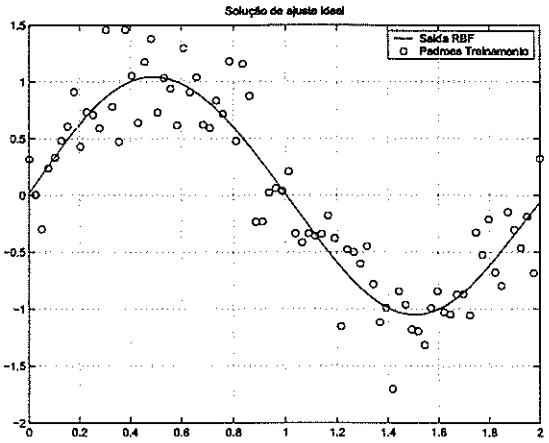

Figura 2.7: Solução ideal de uma RBF para aproximação de função seno.

A Equação 2.44 caracteriza o erro de treinamento segundo a média dos erros quadráticos para uma função ($f_{rna}(x;\Gamma)$) gerada por uma RNA como estimador da regressão $E[d|\mathbf{x}]$ dependente do conjunto de treinamento $\Gamma = \{\mathbf{x}_i, d_i\}_{i=1}^{p}$ (Geman, Bienenstock, and Doursat 1992).

$$E_\Gamma[(f(x;\Gamma) - E[d|x])^2] = (E[d|x] - E_\Gamma[f_{rna}(x,\Gamma)])^2 + E_\Gamma[(f_{rna}(x;\Gamma) - E_\Gamma[f_{rna}(x;\Gamma)])^2] \quad (2.44)$$

A função $E_\Gamma[.]$ representa a esperança em relação ao conjunto treinamento.

A Equação 2.44 caracteriza o erro de treinamento em termos da contribuição da polarização e da variância. O termo $((E[d|x] - E_T[f_{rna}(x,\Gamma)])^2)$ caracteriza o erro devido à polarização onde em média $f_{rna}(x,\Gamma)$ é diferente de $E[d|x]$. Mesmo que a solução não seja polarizada, $f_{rna}(x,\Gamma)$ pode ser altamente sensível aos dados, onde a variabilidade de soluções distintas representada por $(E_T[(f_{rna}(x;\Gamma) - E_\Gamma[f_{rna}(x;\Gamma)])^2])$ caracteriza a variância.

O algoritmo *Backpropagation* (Rumelhart, Hinton, and Williams 1986) e suas variações realizam a minimização da função de erro (Equação 2.11). A utilização deste tipo de procedimento pode ocasionar sobreajustes, pois nem sempre é interessante obter um erro de treinamento mínimo.

Alguns métodos possibilitam controlar a complexidade de uma RNA, através de alterações na estrutura física, como por exemplo o *Optimal Brain Damage* (Cun, Denker, and Solla 1990). A divisão dos dados em subconjuntos (treinamento e validação) é também utilizada para evitar o sobreajuste como o *Early Stopping* (Weigend, Rumelhart, and Huberman 1990) e o *Cross-Validation* (Stone 1978), além de métodos estatísticos como as SVMs (Vapnik 1995).

Segundo (Bartlett 1997), soluções com alta capacidade de generalização podem ser obtidas através do controle da magnitude dos parâmetros livres, sem que seja necessário eliminá-los da rede.

Baseando-se neste princípio, as técnicas de regularização (Wahba 2000) (D. Plaut and Hinton 1986) são utilizadas de forma a aumentar a generalização de RNAs. Durante o processo de treinamento, penalizações sobre a magnitude dos parâmetros são aplicadas de forma a reduzir o erro de generalização.

Um conjunto de métodos de treinamento de RNAs utilizam técnicas de otimização multi-objetivo para treinamento de redes MLP que minimizam o erro de treinamento e a norma dos parâmetros livres encontrando soluções com alta capacidade de generalização (Teixeira, Braga, Takahashi, and Saldanha 2000) (Costa, Braga, de Menezes, Parma, and Teixeira 2002). Uma visão geral destes métodos será apresentada no capítulo 3. O objetivo principal deste trabalho é apresentar uma metodologia de treinamento de redes RBFs utilizando os conceitos de treinamento multi-objetivo.

2.7.2 Regularização

Existem problemas matemáticos, conhecidos como problemas mal-definidos, nos quais não existe uma quantidade de informação suficiente para a deter-

minação de uma única solução pois as informações disponíveis não são suficientes. Problemas desta natureza podem ser resolvidos através de técnicas de regularização (Tikhonov 1963).

A métodologia de regularização conhecida como *ridge regression* é utilizada para solução de problemas de regressão linear mal-condicionados (Hoerl and R. W 1970). O mal-condicionamento é determinado pela dificuldade em calcular a matriz inversa na solução de um sistema linear ocasionado pela dependência linear entre os padrões de treinamento.

Ridge Regression

Partindo de um modelo de alta complexidade, é possível reduzir a sua variância inserindo uma quantidade de polarização no modelo. A introdução de polarização proporciona uma restrição no domínio das possíveis soluções que uma RNA pode apresentar. Tipicamente, isto pode ser alcançado removendo graus de liberdade, ou seja, parâmetros livres da rede. O *rigde regression* não remove parâmetros mas diminui o número de parâmetros efetivos através do controle da magnitude dos mesmos.

Esta técnica é a mesma que a de *weight decay* (Hertz, Krough, and Palmer 1991), onde é acrescentado um termo de penalização à função de custo utilizada para otimização do modelo penalizando soluções que possuam um alto valor de magnitude de seus parâmetros. Algumas funções de penalidades são estudadas em (Friedman 1994).

Para as redes RBFs, as técnicas de *ridge regression* são aplicadas para os parâmetros da camada de saída por serem de natureza linear. O efeito da regularização é provocar uma suavização na resposta da RNA reduzindo o número de parâmetros efetivos ao diminuir a flexibilidade do modelo.

A função de custo utilizada para determinação do modelo neural é dada pela soma dos erros quadráticos de erro e soma quadrática dos parâmetros livres, ponderado por um único termo de regularização (*Global Ridge Regression*) (Equação 2.45) ou com termos individuais para cada parâmetro livre (*Local Ridge Regression*) (Equação 2.46).

$$J = \sum_{i=1}^{k}(y_i - f_{rna}(x_i))^2 + \lambda \sum_{j=1}^{h}(w_j)^2 \tag{2.45}$$

$$J = \sum_{i=1}^{k}(y_i - f_{rna}(x_i))^2 + \sum_{j=1}^{h}\lambda_j w_j^2 \tag{2.46}$$

O termo λ é o parâmetro de regularização que pondera a relação entre a minimização do erro de treinamento e o controle da complexidade do modelo. Um valor pequeno de λ possibilita encontrar soluções de alta complexidade,

ou seja, pesos de alta magnitude. Para valores altos de λ encontra-se soluções altamente polarizadas ou de baixa complexidade.

A matriz de pesos ótima, dada como solução que minimiza a Equação 2.45 é representada na Equação 2.47 e a solução da Equação 2.46 é dada pela Equação 2.48 (Orr 1996).

$$\mathbf{W}^* = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}_h)^{-1}\mathbf{H}^T\mathbf{D} \tag{2.47}$$

$$\mathbf{W}^* = (\mathbf{H}^T\mathbf{H} + \Lambda)^{-1}\mathbf{H}^T\mathbf{D} \tag{2.48}$$

Onde Λ é uma matriz diagonal formada pelos distintos termos de polarização.

A técnica de *local ridge regression* facilita o controle de complexidade em funções que possuem diferenças significativas de suavidade em diferentes partes do espaço de entrada (Orr 1996).

Regularização de raio

A otimização por gradiente descendente para todos os parâmetros parece forçar o valor de raio para zero. Por causa de tal característica, é necessário manter o raio fixo e otimizar as outras variáveis (ou Wang and Zhu 2000). Uma solução encontrada foi aplicar a otimização sobre uma função de custo que penaliza raios muito pequenos (Cohen and Intrator 2000) segundo a Equação 2.49.

$$E = 0,5\sum_{n=1}^{N}\sum_{k=1}^{M}(y_k^n - t_k^n)^2 + \alpha\sum_{k=1}^{M}\frac{1}{r_k} \tag{2.49}$$

O termo de regularização α deve ter um valor pequeno que possa ser estimado a partir de um conjunto validação.

2.7.3 Critérios de seleção de modelos

Afim de se obter uma medida que garanta a seleção de um modelo de alta capacidade de generalização, foram desenvolvidos estimadores para representar o comportamento de um modelo para padrões de entrada desconhecidos. Os critérios de seleção de modelos levam em consideração a matriz de projeção ($\mathbf{P}$) e o número de parâmetros efetivos (γ).

$$\mathbf{P} = \mathbf{I}_p - \mathbf{H}\mathbf{A}^{-1}\mathbf{H}^T \tag{2.50}$$

$$\gamma = p - \text{trace}(\mathbf{P}) \tag{2.51}$$

A matriz de projeção projeta vetores no espaço p-dimensional perpendicular, onde p é o número de padrões de treinamento, para um subespaço h-dimensional, onde h é o número de funções radiais.

O número de parâmetros efetivos (Moody 1992) (Mackay 1992) é uma representação da complexidade de um modelo que leva em consideração o parâmetro de regularização utilizado na solução de um sistema linear. O número de parâmetros efetivos equivale ao número de parâmetros livres quando o termo de regularização é nulo.

No contexto de critérios de seleção de modelo, a validação cruzada (*Cross-Validation*) (Golub, Heath, and Wahba 1979) e suas variações são as ferramentas mais utilizadas para estimação do erro de generalização. A forma básica da validação cruzada é separar os dados disponíveis para criação de modelos em dois subconjuntos: o conjunto de treinamento e o conjunto de validação. Utiliza-se o conjunto de treinamento para o ajuste dos parâmetros e o conjunto de validação para a avaliação da capacidade de generalização do modelo.

O **erro de validação** (e_V) passa a ser então um critério para a seleção de modelos. O modelo selecionado como o de melhor generalização será o que minimiza o erro para um conjunto de dados de validação $\Gamma_V = \{\mathbf{x}_i, d_i\}$.

$$e_V = 0,5. \sum_{i=1}^{p} [d_i - f(\mathrm{x_i})]^2 \tag{2.52}$$

Para se evitar a polarização da solução ocasionada por uma subdivisão particular dos dados, realizam-se várias subdivisões de dados e o erro atribuído ao modelo será a média de todas as realizações. O caso extremo seria utilizar um padrão para validação e o restante para treinamento para todas as combinações de subconjuntos. Esta técnica é denominada de *leave-one-out* (LOO) e sua variância pode ser calculada analiticamente pela Equação 2.53.

$$\xi^2_{LOO} = \frac{\mathbf{Y}^T\mathbf{P}(\mathrm{diag}(\mathbf{P}))^{-2}\mathbf{PY}}{p} \tag{2.53}$$

Uma outra variação da validação cruzada, *Generalised Cross-Validation*, é um critério que envolve um ajuste para erro médio quadrático muito semelhante ao *leave-one-out*. A Equação 2.54 apresenta o cálculo da variância deste erro.

$$\xi^2_{GCV} = \frac{p\mathbf{Y}^T\mathbf{P}^2\mathbf{Y}}{\mathrm{trace}(\mathbf{P})^2} \tag{2.54}$$

Outros critérios de seleção de modelos apresentados na literatura (Efron and Tibshirani 1993) são o *Unbiased Estimate of Variance* (Equação 2.55) , *Final Prediction Error* (Equação 2.56) e o *Bayesian Information Criterion* (Equação 2.57).

$$\xi^2_{UEV} = \frac{\mathbf{Y}^T\mathbf{P}^2\mathbf{Y}}{p-\gamma} \tag{2.55}$$

$$\xi^2_{FPE} = \frac{p+\gamma}{p-\gamma}\frac{\mathbf{Y}^T\mathbf{P}^2\mathbf{Y}}{p} \tag{2.56}$$

$$\xi^2_{BIC} = \frac{p+(\ln(p)-1)\gamma}{p-\gamma}\frac{\mathbf{Y}^T\mathbf{P}^2\mathbf{Y}}{p} \tag{2.57}$$

A Figura 2.8 apresenta um exemplo de avaliação de critérios de seleção de modelos para o treinamento de redes RBF com 80 neurônios de raio 0,05 para o problema de regressão da função seno.

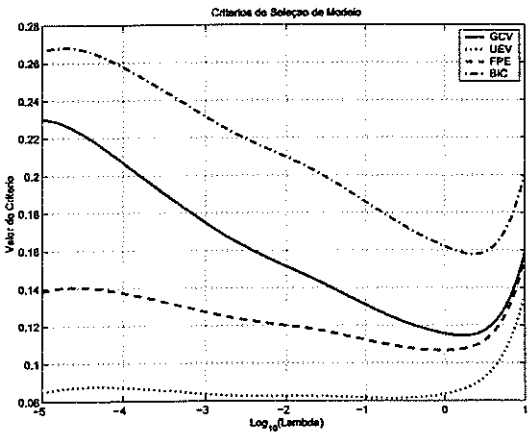

Figura 2.8: Exemplo de critérios de seleção de modelos baseados em termos de regularização.

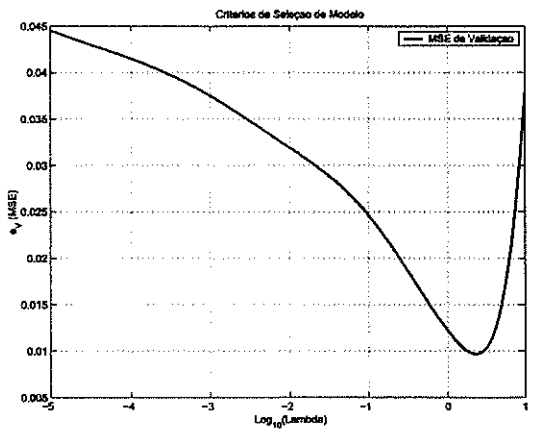

Figura 2.9: Exemplo de critério de seleção de modelos baseado em conjunto validação.

Nota-se que todas as soluções de mínino critério de avaliação convergem para valores próximos de um mesmo valor de termo de regularização.

2.7.4 *Subset selection*

Outra maneira de controlar o equilíbrio de polarização e variância é trabalhar com modelos distintos gerados a partir de um único conjunto de dados. Ao contrário das técnicas de regularização que atuam na magnitude dos parâmetros livres, o *subset selecion* controla a complexidade do modelo alterando o número de parâmetros livres, ou seja, incluindo ou retirando neurônios de redes RBFs.

Forward Selection

A partir de um conjunto de padrões de treinamento, a metodologia de *fowards selection*, acrescenta um neurônio na camada escondida por iteração.

A inclusão de neurônios extras se encerra quando se atinge um valor mínimo de algum critério de seleção de modelos.

Nesta técnica se avalia a quantidade de neurônios utilizados na camada escondida e a cada nova estrutura de rede realiza-se alguma estratégia de aprendizagem para estimar os parâmetros das funções de base bem como os pesos das conexões da camada de saída.

Adotando a técnica de mínimos quadrados ortogonais (Chen, Cowan, and Grant 1991), a eficiência do técnica de *foward selection* pode ser aumentada garantindo que cada novo vetor inserido na matriz de interpolação correspondente ao novo neurônio seja ortogonal aos outros vetores.

O algoritmo RCE, apresentado em (Reily, Cooper, and Elbaum 1982) e sua extensão, P-RCE são exemplos de *foward selection* onde se define a estrutura de um rede RBF através de introdução de novos neurônios quando necessário.

Backward Elimination

Analogamente ao algoritmo de *foward selection*, a técnica de *backward selection* parte de uma rede com um número excessivo de funções de base. A cada iteração, o neurônio que causa o menor incremento de erro é eliminado da rede. O algoritmo é finalizado quando se atinge um valor limite de erro de treinamento ou então quando se atinge um valor de mínimo de erro para um conjunto de validação ou qualquer outro critério de seleção de modelos. Neste momento, a complexidade do modelo é considerada a mínima necessária para resolver o problema em questão.

2.8 Conclusões do capítulo

Neste capítulo foram apresentados os conceitos gerais sobre redes RBFs e seus principais algoritmos de treinamento. Percebe-se que as RBFs possuem uma formalização simples, assim como seus métodos de ajuste de parâmetros, constituindo uma alternativa interessante para as redes MLP. O capítulo seguinte abordará a metodologia multi-objetivo para treinamento de RNAs, em especial redes MLP, na qual foi apresentada a primeira proposta deste tipo de treinamento.

CAPÍTULO

3

Otimização Multi-objetivo para Treinamento de Redes Neurais

Neste capítulo serão apresentados os conceitos básicos sobre Otimização Multi-objetivo (MOBJ) e sua aplicação para treinamento de redes neurais do tipo MLP.

3.1 Introdução

A maioria do algoritmos para treinamento de redes neurais artificiais utiliza apenas uma função objetivo para a determinação de seus parâmetros livres, representada pelo somatório dos erros quadráticos (e_T) do conjunto de padrões de treinamento $\Gamma = \{\mathbf{x}_i, d_i\}_{i=1}^{p}$

$$e_T = 0,5. \sum_{i=1}^{p} [d_i - f(\mathbf{x}_i)]^2. \tag{3.1}$$

As soluções encontradas por esses métodos não garantem uma alta capacidade de generalização, pois a minimização do erro de treinamento (e_T) não garante a minimização do erro de generalização, principalmente quando o conjunto de treinamento Γ é constituído de dados ruidosos.

Conforme discutido no Capítulo 2, o projeto de redes neurais tem como objetivo encontrar soluções de alta capacidade de generalização, onde se encontra a complexidade adequada de um modelo neural necessária para a solução de problemas de regressão e classificação, estabelecendo um equilíbrio entre a polarização e a variância.

Através da abordagem multi-objetivo, é possível limitar a complexidade

efetiva de modelos neurais ao se representar uma medida de complexidade através de uma função objetivo a ser otimizada simultaneamente com a função de erro.

A complexidade pode ser restringida a partir da limitação do espaço de soluções. Esta limitação pode ser representada pela restrição do número de dimensões, representado pelo número de parâmetros livres (Lawrence, Giles, and Tsoi 1996) ou pela restrição de valores atribuídos aos parâmetros, como por exemplo, limitando a magnitude dos mesmos (Bartlett 1997).

Ao se controlar a magnitude dos parâmetros livres dos modelos neurais, redes super-dimensionadas podem se comportar como sistemas menos complexos. Desta forma, o problema da definição do número de neurônios a ser utilizado em uma rede neural passa a ser contornado pelo controle da magnitude dos seus parâmetros.

No processo de treinamento de redes neurais, as funções de complexidade e erro sobre o conjunto de treinamento são conflitantes, uma vez que atingir níveis baixos de erro de aproximação exige modelos mais complexos, enquanto que modelos muito simples não são capazes de mapear soluções de baixo erro. Desta forma, a otimização multi-objetivo se apresenta como uma importante ferramenta para a solução deste problema por buscar a melhor relação entre complexidade e erro para soluções de redes neurais.

A seguir serão apresentados os conceitos básicos de soluções de problemas multi-objetivos e a utilização de técnicas de busca de solução para treinamento de redes do tipo MLP.

3.2 Fundamentos da Otimização Multi-Objetivo

A solução de problemas multi-objetivo consiste em buscar soluções que otimizem simultaneamente mais de um objetivo, satisfazendo todas as restrições impostas. Dificilmente são encontrados problemas cuja solução torne mínimo todos os objetivos desejados. Tais soluções são denominadas *soluções utópicas*, uma vez que não há nenhum conflito entre as funções objetivo. Para a maioria dos problemas, as relações de custo são contraditórias desejando encontrar a solução que represente o melhor equilíbrio entre todos os objetivos.

Para um conjunto de soluções de um problema multi-objetivo, existirão soluções que, ao comparadas com outras, serão consideradas melhores quando um dos objetivos for considerado, mas piores ao se considerar os outros objetivos. A busca por este conjunto de soluções, denominado de soluções *eficientes* ou *Pareto-ótimo*, constitui um dos principais passos na resolução de problemas de mais de um objetivo.

Formalmente pode-se descrever um problema de otimização multi-objetivo como encontrar um vetor $\Omega = [\omega_1, \omega_2, ...\omega_n]^T$, correspondente ao vetor de *variáveis de decisão* ou de otimização, que minimize n critérios de avaliação, expressos como funções matemáticas f_i, denominadas *funções objetivo*, que atendam a um conjunto de p restrições de igualdade e m restrições de desigualdade, representadas pelas funções h_i e g_i respectivamente.

$$\Omega^* = \underset{\Omega}{arg\ \min} \begin{Bmatrix} f_1(\Omega) \\ \vdots \\ f_n(\Omega) \end{Bmatrix}$$

$$sujeito\ a: \begin{cases} h_1(\Omega) = 0 \\ \vdots \\ h_p(\Omega) = 0 \\ g_1(\Omega) \leq 0 \\ \vdots \\ g_m(\Omega) \leq 0 \end{cases} \tag{3.2}$$

Em problemas reais, as funções h_i e g_i podem representar os limites tecnológicos a partir dos quais não é possível a realização concreta de uma solução. A região de soluções representada por estas funções constitui a região de soluções factíveis. Para o caso de treinamento de redes neurais, não existem restrições para os valores possíveis de suas variáveis de otimização.

O conjunto Pareto-Ótimo (Pareto 1896) caracteriza soluções *não-dominadas* ou *eficientes* de problemas multi-objetivos onde não é possível determinar *a priori*, qual solução pertencente a este conjunto é a melhor.

Quase sempre o conjunto Pareto-ótimo (ϑ^*) é formado por mais de uma solução, de forma que se necessite estabelecer critérios de decisão para determinação da solução final de um problema multi-objetivo. Esta etapa é conhecida como *Etapa de Decisão*. Neste momento, todas as soluções constituintes de um conjunto Pareto são avaliadas segundo algum critério de seleção, de forma que apenas uma solução seja selecionada como resposta para o problema.

A Figura 3.1 apresenta um exemplo de soluções que aproximam um conjunto Pareto. Este conjunto de soluções determina o limite entre o espaço de soluções possíveis e o espaço de soluções não existentes.

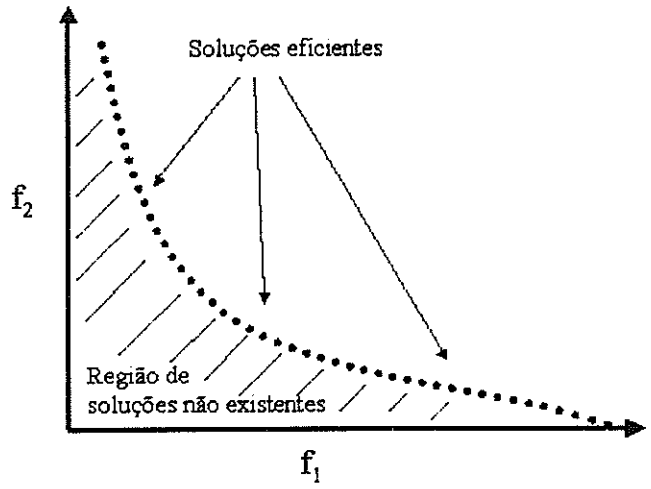

Figura 3.1: Exemplo de um Conjunto Pareto-ótimo para Otimização bi-objetivo.

Atualmente, diversos pesquisadores buscam desenvolver metodologias eficientes para geração de conjunto Pareto-ótimo (Takahashi, Peres, and Ferreira 1997) (Costa, Braga, de Menezes, Parma, and Teixeira 2002), bem como critérios para seleção de soluções na etapa de decisão (Medeiros 2004).

Neste trabalho será utilizado o método ε-restrito (Chankong and Haimes 1983) para solução do problema multi-objetivo, pois por enquanto não se deseja avaliar a eficiência computacional, mas a possibilidade de se encontrar soluções para redes neurais do tipo RBF de alta capacidade de generalização.

3.3 Treinamento Multi-Objetivo para redes MLP

A utilização de um método multi-objetivo para treinamento de redes MLPs foi proposta em (Teixeira, Braga, Takahashi, and Saldanha 2000). Uma outra técnica de geração de soluções eficientes foi apresentada em (Costa, Braga, de Menezes, Parma, and Teixeira 2002) que adota os mesmos conceitos de controle de generalização para redes MLPs porém tratando o problema da convergência na busca de soluções do conjunto Pareto-ótimo.

A formulação do problema multi-objetivo para treinamento de redes MLPs é construída a partir de dois funcionais. O funcional erro de treinamento (e_T) representa a qualidade de aproximação do modelo segundo um conjunto de padrões de treinamento e a medida de complexidade é representada pela norma euclidiana de seus parâmetros livres ($\|W\|$), constituído pelo vetor de pesos das conexões entre os neurônios das camadas de redes MLPs.

Uma vez que a flexibilidade de um modelo é controlada pela magnitude de seus parâmetros livres, não se realiza nenhuma modificação estrutural na

arquitetura da rede em treinamento. Dado um conjunto de padrões de treinamento ($\Gamma = \{\mathbf{x}_i, d_i\}_{i=1}^{p}$), o algoritmo parte de uma rede super-dimensionada e busca encontrar a melhor relação entre e_T e $\|W\|$ que leve a aumentar as chances de encontrar soluções de alta capacidade de generalização através da limitação de complexidade e minimização do erro de treinamento. O treinamento de redes MLP pode então ser representado pelo problema multi-objetivo descrito por

$$\mathbf{W}^* = \underset{\mathbf{W}}{arg\ \min} \begin{cases} f_1(\mathbf{W}) = e_T(\mathbf{W}) \\ f_2(\mathbf{W}) = \|\mathbf{W}\| \end{cases} \tag{3.3}$$

em que não há nenhuma restrição no espaço de soluções.

As soluções encontradas podem ser representadas em um espaço solução $\Re^2$ definido pelos funcionais norma e erro. Existirão neste espaço, z soluções que, comparadas a todas as outras, serão melhores em algum dos objetivos, mas piores ao se considerar o outro objetivo. Tais soluções são ditas *eficientes* e constituem o conjunto de soluções Pareto-ótimo ($\vartheta^* = \{\mathbf{W}_i^*\}_{i=1}^{z}$).

A primeira etapa de treinamento de redes MLPs é encontrar o conjunto de soluções eficientes, onde cada solução representa uma RNA de complexidade distinta que possua um erro de treinamento mínimo. Desta forma, o universo de modelos disponíveis para seleção se torna mais restrito com a garantia de que estas soluções são as mais eficientes possíveis para uma dada arquitetura de RNA.

Existem algumas metodologias para a geração do conjunto de soluções eficientes para redes MLP, onde pode-se destacar o método ε-restrito, o algoritmo de Relaxação (Teixeira 2001) e o treinamento por Modos Deslizantes (Costa 2001).

A Figura 3.2 ilustra um conjunto de soluções eficientes para o problema de treinamento de redes MLPs.

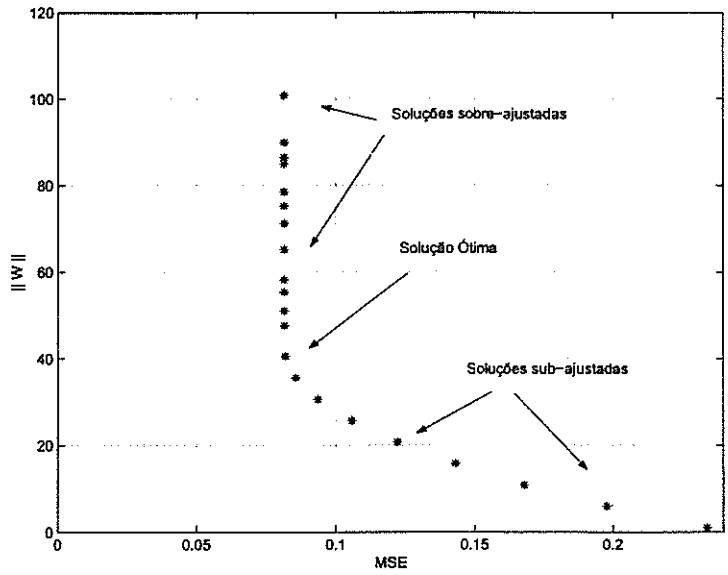

Figura 3.2: Exemplo de um Conjunto Pareto-ótimo para treinamento de redes MLP.

É importante notar que o conjunto Pareto possui soluções sub-ajustadas, representadas por modelos de baixo valor de norma e alto valor de erro e soluções super-ajustadas, localizadas em pontos de alto valor de norma e baixo valor de erro. Entre os dois extremos de ajuste, está localizada a solução que equilibra os efeitos de polarização e variância. Esta solução possui um ajuste ideal para um dado conjunto de dados.

O último passo é selecionar a solução pertencente ao conjunto de soluções eficientes que possua a maior capacidade de generalização. Esta etapa é conhecida como *Etapa de Decisão*, onde se aplica algum critério ou algoritmo para a seleção de modelo, dito como *decisor*.

Por se tratar de um critério simples e aplicável a modelos não lineares, o critério de seleção de modelo utilizado para o treinamento de redes MLPs é o erro para o conjunto de validação. Desta forma, o conjunto de soluções eficientes é avaliado para um conjunto de dados de validação ($\Gamma_V = \{\mathbf{x}_{vi}, d_{vi}\}_{i=1}^{pv}$) e a solução final é escolhida como a que possui o valor mínimo de erro de aproximação (e_V) para este conjunto.

A regra de decisão utilizada para treinamento de redes MLP é dada por

$$\mathbf{W}^* = \underset{\mathbf{W}\in\vartheta^*}{arg\ \min} e_V(\mathbf{W}) \tag{3.4}$$

$$e_V = 0,5. \sum_{i=1}^{pv} [d_{vi} - f(\mathbf{x}_{vi})]^2, \tag{3.5}$$

onde e_V representa o erro de predição para um conjunto Γ_V de padrões de

validação.

A seguir serão apresentadas algumas metodologias de geração de conjunto Pareto para treinamento de redes MLP.

3.3.1 Método ε-restrito

A solução de problemas de múltiplos objetivos pode ser obtida por meio do método ε-restrito (Duckstein 1984), que transforma um problema multi-objetivo em vários sub-problemas mono-objetivos. A geração de soluções eficientes ocorre ao realizar-se a otimização de um objetivo enquanto que os demais serão considerados restrições. Logo, um problema multi-objetivo é resolvido através de diversas soluções encontradas para um problema mono-objetivo ao se considerar diferentes valores de restrição (ε_i). A variação de ε_i permite gerar o conjunto Pareto-ótimo mesmo que o problema seja não-convexo.

De acordo com o método ε-restrito (Duckstein 1984), um problema de m objetivos, sem restrições, pode então ser formulado segundo

$$\begin{aligned} &\arg\min_{\Omega} f_1(\Omega) \\ \text{sujeito a}: &\begin{cases} f_2(\Omega) \leq \varepsilon_{2i}, & \text{para } i = 1, 2, ..., z_2 \\ \vdots & \\ f_m(\Omega) \leq \varepsilon_{mi}, & \text{para } i = 1, 2, ..., z_m \end{cases} \end{aligned} \tag{3.6}$$

e resolvido para z valores de restrição.

Sua aplicação para o treinamento de redes MLP foi apresentada em (Teixeira 2001). Neste problema a função objetivo escolhida para se tornar uma restrição foi a função norma. A busca pelo conjunto Pareto-ótimo, parte de redes sub-dimensionadas de norma restrita a baixos valores de ϵ_w em direção a soluções super-dimensionadas de normas restritas a altos valores de ϵ_w.

$$\begin{aligned} &\arg\min_{\mathbf{w}} e_T(\mathbf{w}) \\ \text{sujeito a}: &\ \|\mathbf{W}\| \leq \epsilon_{wi}, \quad \text{para } i = 1, 2, ..., z \end{aligned} \tag{3.7}$$

O método ε-restrito possui uma baixa complexidade computacional para problemas lineares, porém problemas não lineares exigem um grande esforço computacional. A grande dificuldade está em determinar os valores de ϵ_{wi}, pois dependendo dos valores escolhidos, o problema pode se tornar infactível ou até gerar pontos não-eficientes.

3.3.2 Método de Relaxação

No trabalho (Takahashi, Peres, and Ferreira 1997) é apresentada uma variação do método ε-restrito. Esta nova abordagem contorna o problema de geração de problemas infactíveis pelo fato de se estabelecer valores de restrições aceitáveis.

Seja f^{**} o vetor de objetivos referentes à solução utópica do problema e f^{*} o vetor de objetivos correspondentes a cada mínimo individual do problema, determina-se o vetor v de valores objetivos

$$v = f^{**} + \sum_{i=1}^{z} \sigma_i (f_i - f^{**}). \tag{3.8}$$

Assim como no método ε-restrito, o problema multi-objetivo é reescrito como um problema mono-objetivo descrito por

$$\begin{aligned} &\arg\min_{\psi} \ \min_{\psi,\eta} \eta \\ &\text{sujeito a}: \ f(\psi) \leq f^{**} + \eta v, \end{aligned} \tag{3.9}$$

onde as funções de restrição incorporam as funções objetivo do problema original.

Desta forma, o método de relaxação estrutura a busca de soluções para a formação do conjunto Pareto-ótimo. Com o uso da função de custo auxiliar, todas as funções objetivo iniciais são tratadas como restrições as quais são linearmente dependentes da variável auxiliar η.

Para o treinamento de redes MLP, primeiramente realiza-se um treinamento supervisionado de uma rede super-dimensionada segundo um método de treinamento mono-objetivo. A solução encontrada é utilizada para o treinamento multi-objetivo com a proposta de obter um número z de novas redes de complexidade inferior.

O vetor v é construído por meio de z combinações convexas de vetores formados pelo ótimo de cada função objetivo segundo

$$v_i = f^{**} + \sigma_i (f_1^{*} - f^{**}) + (1 - \sigma_i)(f_2^{*} - f^{**}), \text{ para } 0 < \sigma_i < 1. \tag{3.10}$$

O valor de f_1^{*} equivale ao valor da função erro avaliado para a solução (W_1) encontrada no treinamento realizado anteriormente e o valor de f_2^{*} é trivial, uma vez que a solução que minimiza a norma de pesos (W_2) equivale a valores nulos para todos os parâmetros. A solução utópica f^{**} é formada pelo vetor correspondente ao valor de erro para a solução W_1 e a norma de pesos da solução W_2, ou seja,

$$W_1^* = \arg\min_W f_1 = e_T(W) \tag{3.11}$$

$$W_2^* = \arg\min_W f_2 = \|W\| \tag{3.12}$$

$$\varphi_1 = f_1(W_1^*) \tag{3.13}$$

$$\varphi_2 = f_2(W_2^*) \tag{3.14}$$

$$f^{**} = \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix}. \tag{3.15}$$

Cada vetor v_i será responsável pela geração de uma solução do conjunto Pareto-ótimo que corresponde a uma solução factível para o problema multi-objetivo. O problema de treinamento de redes MLPs fica então representado pelo problema mono-objetivo descrito por

$$\begin{aligned} W^* &= \arg\min_W \min_{W,\eta} \eta \\ \text{sujeito a}: &\begin{cases} g_1(W,\eta) = e_T - \varphi_1 - \eta v_{k_1} \leq 0 \\ g_2(W,\eta) = \|W\| - \varphi_2 - \eta v_{k_2} \leq 0 \end{cases} \end{aligned} \tag{3.16}$$

O método apresentado propõe a utilização dos dois extremos para a determinação das soluções intermediárias do conjunto Pareto-ótimo. Após a determinação do conjunto de soluções eficientes, adota-se um decisor para aplicação de algum critério de seleção de modelo.

3.3.3 *Controle por Modos Deslizantes*

O algoritmo SMC-MOBJ proposto em (Costa, Braga, de Menezes, Parma, and Teixeira 2002) faz uso de uma técnica de controle conhecida como *Teoria de Modos Deslizantes* (Itkis 1976) para o problema de treinamento multi-objetivo de redes MLP (Teixeira, Braga, Takahashi, and Saldanha 2000). A versão para treinamento mono-objetivo destas redes é apresentado em (Parma, Menezes, and Braga 1998).

A direção de busca utilizada para ajuste de parâmetros livres de redes MLP é determinada pelas *Superfícies de Deslizamento*, definida em função do erro e norma de pesos e suas respectivas derivadas.

Durante o treinamento da RNA, estabelece-se um ponto desejado no espaço $\Re^2$ definido pelos funcionais norma e erro. O algoritmo SMC-MOBJ conduz a

solução ao ponto desejado através de duas superfícies de deslizamento que se interceptam neste ponto desejado.

A primeira superfície é definida pela diferença entre o erro atual da solução e_{Tk} e o erro desejado e_{Td}, segundo a Equação 3.17. A segunda superfície é definida pela diferença entre a norma atual $\|W_k\|$ e a norma desejada $\|W_d\|$ (Equação 3.18).

$$S_e = e_{Tk} - e_{Td} \tag{3.17}$$

$$S_w = \|W_k\|^2 - \|W_d\|^2 \tag{3.18}$$

A função de custo referente à busca de um ponto desejado no plano de soluções caracterizado por erro e norma é representada por

$$J = 0,25(e_{Tk} - e_{Td})^2 + 0,25(\|W_k\|^2 - \|W_d\|^2)^2. \tag{3.19}$$

A direção utilizada para ajuste dos parâmetros livres (Δw) apresenta um termo relativo à superfície de erro e um termo relativo à superfície de complexidade (Equação 3.20). Ambos os termos são ponderados por respectivos ganhos (α e β) e a direção é dada pelos sinais (sgn) das superfícies de deslizamento, conduzindo a solução ao valor de erro e norma desejados.

$$\Delta w_{ij} = -\alpha.sgn(S_e).\frac{\partial e_T}{\partial w_{ij}} - \beta.sgn(S_w).w_{ij} \tag{3.20}$$

$$sgn(S) = \begin{cases} +1, & para\ S > 0 \\ 0, & para\ S = 0 \\ -1, & para\ S < 0 \end{cases} \tag{3.21}$$

O conjunto de soluções eficientes pode ser obtido, através deste método, ao tentar alcançar soluções localizadas na região de soluções não-existentes. Desta forma, a busca de uma solução ao movimentar em direção a um valor de erro e norma não-existente, ficará interrompida no limite da região de soluções factíveis, onde se localiza do conjunto Pareto-ótimo. O processo de busca de soluções não-existentes é realizado para vários valores de $\|W\|$ desejados e erro muito baixos, de forma a gerar um conjunto de soluções eficientes como aproximação do conjunto Pareto-ótimo. Outras variações sobre o algoritmo apresentado, bem como um estudo sobre os critérios de convergência são detalhados em (Costa 2001).

3.4 Conclusões

Neste capítulo, abordaram-se os fundamentos de otimização multi-objetivo utiizados no treinamento de redes MLP. Todos os métodos necessitam de uma busca de soluções com vários níveis de complexidade distintas afim de encontrar a solução que melhor equilibra os efeitos de polarização e variância. A seleção da solução final é realizada pela etapa de decisão onde se aplica algum critério de seleção de modelo que garanta um baixo erro de generalização, em geral representado pelo erro para um conjunto de validação. O capítulo seguinte será apresentada uma nova proposta de algoritmo de treinamento multi-objetivo, desta vez para redes RBF.

CAPÍTULO 4

Otimização Multi-objetivo para treinamento de RBF

Este capítulo apresenta uma nova metodologia de treinamento de redes neurais artificiais de função de base radial. O algoritmo apresentado foi desenvolvido com o objetivo de encontrar um equilíbrio entre polarização e variância através do controle da complexidade de redes RBF por meio da otimização multi-objetivo.

Inicialmente será apresentado o comportamento das soluções obtidas com redes RBFs em função de seus parâmetros livres, assim como uma medida de complexidade para a camada escondida destas redes.

4.1 Introdução

Na abordagem multi-objetivo para redes multi-layer perceptron (MLP) (Teixeira, Braga, Takahashi, and Saldanha 2000)(Costa, Braga, de Menezes, Parma, and Teixeira 2002), o equilíbrio entre a polarização e variância é representado pela otimização da soma quadrática do erro de treinamento e da complexidade da rede neural, representada pela norma dos pesos das conexões entre seus neurônios.

Em função da diferença de funções internas dos neurônios das camadas ocultas, a medida da complexidade utilizada para redes MLP não pode ser aplicada aos parâmetros livres presentes nas camadas escondidas de redes RBFs. Enquanto que as redes MLP utilizam valores de pesos de conexões para ponderação das entradas e funções sigmoidais, as redes RBF fazem uso de distâncias relativas e funções de transferência radiais. A diferença entre

o tratamento da informação entre estas redes exige a definição de uma nova medida de complexidade a ser aplicada para os neurônios da camada intermediária de redes RBF.

Os neurônios presentes na camada de saída de uma rede RBF são semelhantes aos neurônios da camada de saída de redes MLP, podendo fazer uso da norma da matriz de pesos das conexões destes neurônios. Para a caracterização da camada escondida, que possuem uma natureza diferente das redes MLP, utilizou-se a norma da matriz de interpolação.

Nota-se que o estudo envolve três grandezas distintas: a *complexidade da camada de saída*, a *complexidade da camada escondida* e o *erro de treinamento*. Deseja-se encontrar a complexidade mínima necessária de modo que a resposta representada por uma rede neural seja a mais suave possível, não mapeando os ruídos dos dados de treinamento. Desta forma, será possível encontrar soluções de maior capacidade de generalização.

Este problema possui a natureza de um problema de otimização multi-objetivo onde se deseja minimizar simultaneamente a complexidade do modelo neural e o erro de ajuste.

4.2 Caracterização de complexidade em redes RBF

De acordo com a arquitetura de redes RBF, o tratamento da informação é distinto entre as suas duas camadas. A camada escondida realiza uma transformação não linear dos dados de entrada, de acordo com os valores de distância e dispersão das funções de base radial e a disposição espacial dos dados de entrada. O estudo realizado neste trabalho é abordado em redes de função de base gaussianas. A extensão dos resultados aqui representados para outros tipos de função de base constitui uma proposta de continuidade.

O número de funções de base radial e seus parâmetros caracterizam a complexidade da camada escondida de uma rede RBF. É desejável encontrar uma medida que caracterize um nível de complexidade para esta camada de forma que se possa controlar a flexibilidade do modelo RBF.

Pelo fato de as redes RBF utilizarem uma combinação linear de funções de base radiais de parâmetros de naturezas distintas, existem comportamentos específicos que diferem suas soluções quanto à questão do dilema de polarização e variância.

Para modelos de poucos neurônios na camada escondida e raios pequenos, o espaço de entrada coberto pelas funções de base é restrito a regiões muito próximas aos centros das funções de base caracterizando soluções de respostas locais conforme ilustra a Figura 4.1.

Soluções de sub-ajuste ocorrem quando se possui poucos neurônios de

raios relativamente altos (Figura 4.2). Enquanto que, soluções de sobre-ajustes são mais comuns em redes de muitos neurônios na camada escondida e baixos valores de raio (Figura 4.4).

Outra situação específica de redes RBFs ocorre em modelos de muitos neurônios na camada escondida e raios grandes. Neste caso, o espaço de entrada é amplamente coberto, ocorrendo grande superposição dos espaços de mapeamento das funções de base.

Desta forma, os padrões de entrada são mapeados para pontos muito próximos em um espaço de alta dimensionalidade. Este conjunto de dados mapeados acaba perdendo sua representatividade, tornando a matriz de interpolação uma matriz mal condicionada, formada por um grande número de vetores linearmente dependentes. O problema linear de determinação dos pesos da camada de saída se torna mal-condicionado, impossibilitando a geração de soluções satisfatórias. A Figura 4.5 apresenta uma solução encontrada para uma situação de mal condicionamento da matriz de interpolação.

A Tabela 4.2 apresenta um resumo dos comportamentos das soluções de redes RBF segundo as condições da camada escondida e as suas respectivas respostas ilustrativas para um problema de regressão da função *seno* à qual foi adicionado um ruído gaussiano de média nula e desvio padrão de valor 0,4.

A determinação do número de neurônios, bem como as posições dos centros e magnitude do valor de raio das funções de base radial é totalmente dependente do conjunto de dados utilizado para treinamento. Os valores de pesos são dependentes do mapeamento realizado pela camada escondida, tendo forte correlação com o condicionamento do sistema linear formado pela matriz de interpolação e a matriz de pesos de conexões.

Tabela 4.1: Comportamento de soluções em relação aos parâmetros da camada escondida de redes RBFs.

Arquitetura	Ajuste	Solução	Figura
Poucos neurônios e raios pequenos	Local	Local	4.1
Poucos neurônios e raios grandes	Sub-ajuste	Polarizada	4.2
Neurônios e raios médios	Ajuste	Equilibrada	4.3
Muitos neurônios e raios pequenos	Sobre-ajuste	Variância	4.4
Muitos neurônios e raios grandes	Global	Mal-Condicionada	4.5

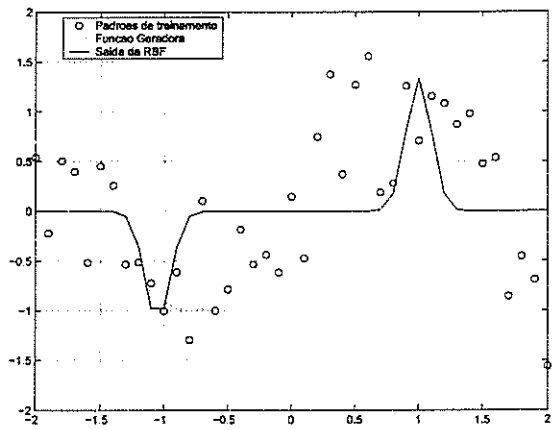

Figura 4.1: Solução de ajuste local - 2 neurônios e raio 0,1.

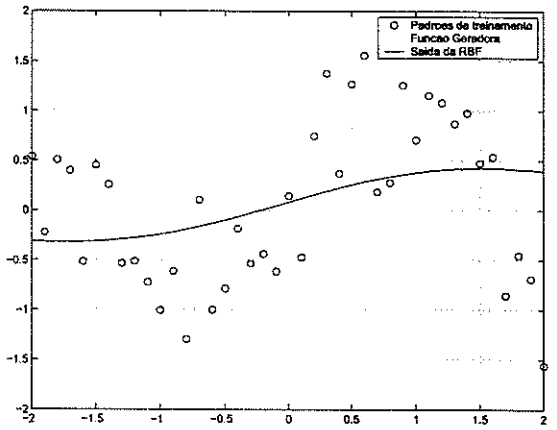

Figura 4.2: Solução de sub-ajuste - 2 neurônios e raio 1,5.

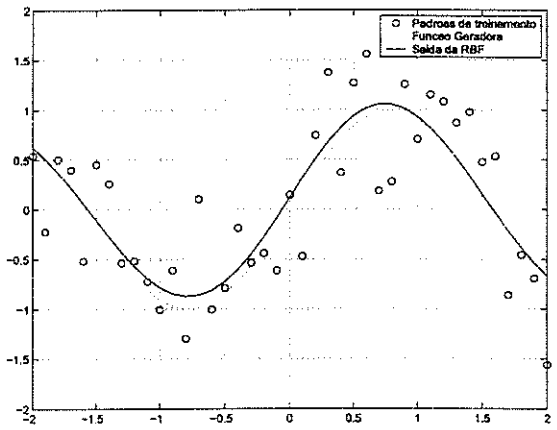

Figura 4.3: Solução de ajuste - 4 neurônios e raio 1.

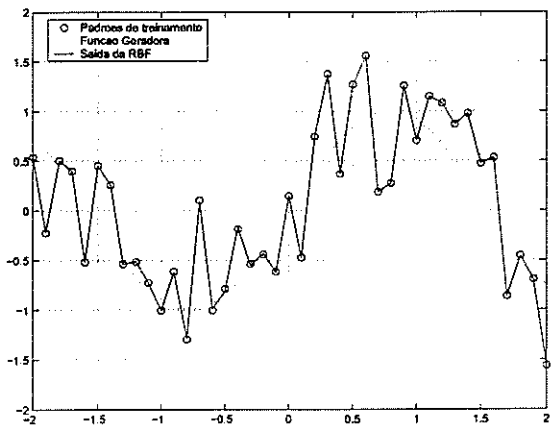

Figura 4.4: Solução de sobre-ajuste - 40 neurônios e raio 0,1.

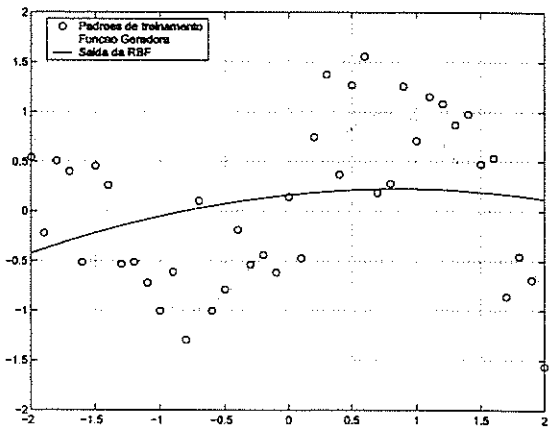

Figura 4.5: Solução de mal-condicionamento - 40 neurônios e raio 180.

Conforme discutido anteriormente existem cinco comportamentos distintos na caracterização de soluções para redes RBF. Tais características correspondem a certos valores de erro de ajuste para os conjuntos de dados de treinamento e validação.

As superfícies 4.6 e 4.7 apresentam um exemplo de comportamento do erro de treinamento e validação para diversas combinações de número de centros e valores de raio.

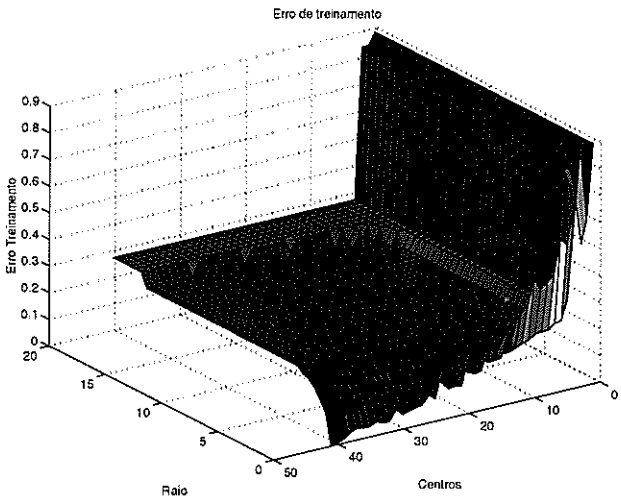

Figura 4.6: Superfície de erro de treinamento segundo valores de raio e centros.

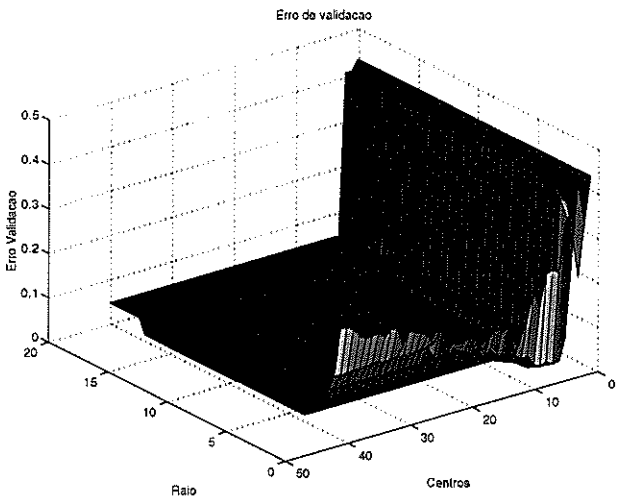

Figura 4.7: Superfície de erro de validação segundo valores de raio e centros.

As superfícies apresentadas foram obtidas avaliando várias combinações de valores de raios e número de centros para o problema de regressão da função *seno*. Os centros foram determinados pelo algoritmo *K-médias* (MacQueen 1967) e os pesos pelo método dos *Mínimos Quadráticos* (Orr 1996).

As superfícies ilustram o comportamento de ajuste para redes superdimensionadas, nas quais o erro de treinamento atinge seu valor mínimo em uma configuração diferente da solução de menor erro de validação.

A Tabela 4.2 apresenta a relação geral das possíveis combinações de topologia de redes RBF que foram discutidas na Tabela 4.2 e as Figuras 4.8 e 4.9 apresentam as curvas de nível de erro e a localização de cada categoria de solução para as superfícies de erro de treinamento e validação para o caso da aproximação da função *seno*. O ponto assinalado representa a solução de melhor generalização, caracterizado pela configuração de valor mínimo para o erro de validação.

Tabela 4.2: Categorias de soluções segundo os parâmetros da camada escondida de redes RBFs.

Solução	Arquitetura
1	Poucos neurônios e raios pequenos
2	Poucos neurônios e raios grandes
3	Número de neurônios e raios médios
4	Muitos neurônios e raios pequenos
5	Muitos neurônios e raios grandes

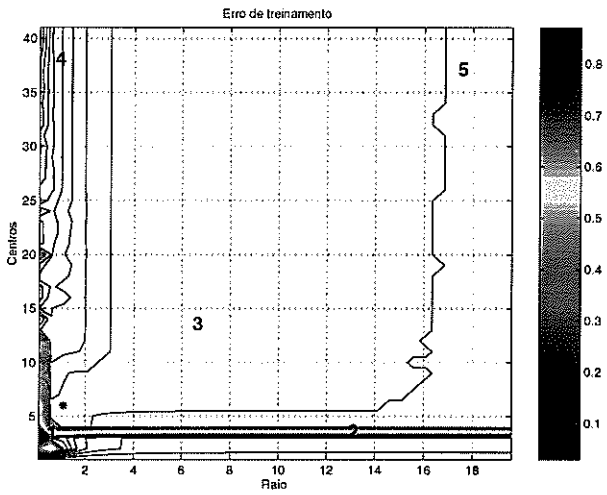

Figura 4.8: Projeção de erro de treinamento para configurações de centros e raios.

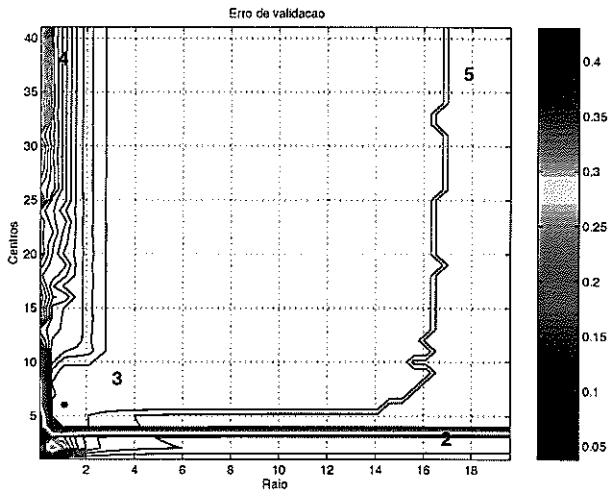

Figura 4.9: Projeção de erro de validação para configurações de centros e raios.

Após apresentado o comportamento das soluções segundo a configuração do número de centros e raios da camada escondida de redes RBF, serão sugeridas quantidades para medição de complexidade para estas redes.

4.2.1 *Caracterização de complexidade para a camada escondida*

Conforme apresentado, o número de centros e valor dos raios são grandezas determinantes para a caracterização das soluções de redes RBFs. Neste trabalho é apresentada uma forma de representar a integração entre estas duas grandezas para caracterização de complexidade de neurônios da camada escondida. A medida proposta a ser utilizada é a norma euclidiana da matriz de interpolação ($\|\mathbf{H}\|$). Existem outras medidas para norma de matrizes mas estas não foram exploradas neste trabalho.

As Figuras 4.10 e 4.11 ilustram o comportamento desta medida de complexidade segundo variação do número de centros e valores de raio para um problema de regressão da função *seno*. O comportamento dos parâmetros

ilustrados neste trabalho se estendeu para todos os outros conjuntos de dados nos quais foram realizados os mesmos testes.

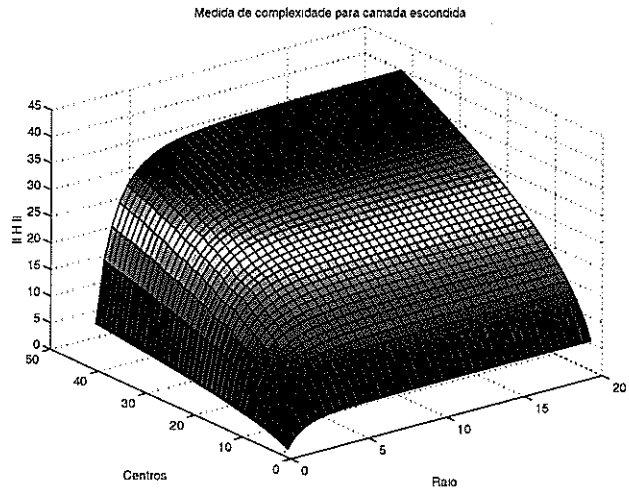

Figura 4.10: Superfície de caracterização de complexidade para camada escondida.

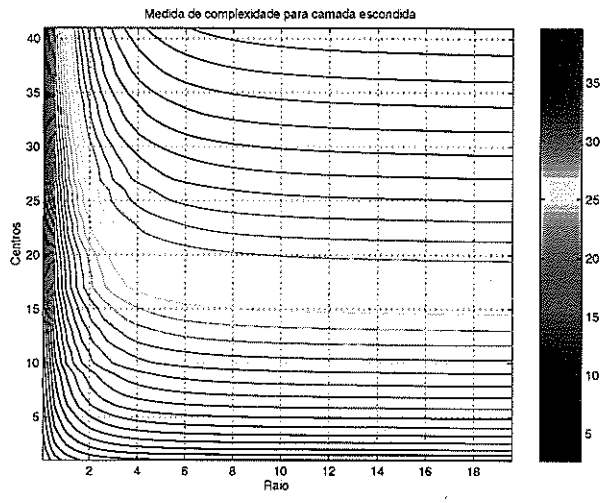

Figura 4.11: Curvas de nível da superfície de caracterização de complexidade para camada escondida.

A matriz **H** é construída a partir da avaliação de cada função de base para todo o conjunto de treinamento. Nas Figuras 4.12 e 4.13 são apresentados os comportamentos das funções de base gaussianas segundo o valor de distância e raios.

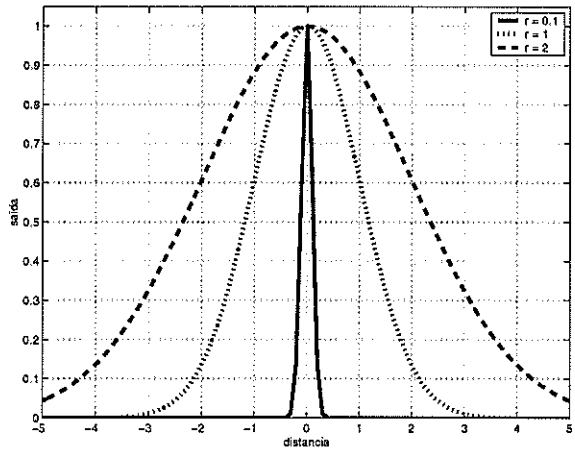

Figura 4.12: Resposta de função gaussiana segundo variação do parâmetro distância para três valores de raio.

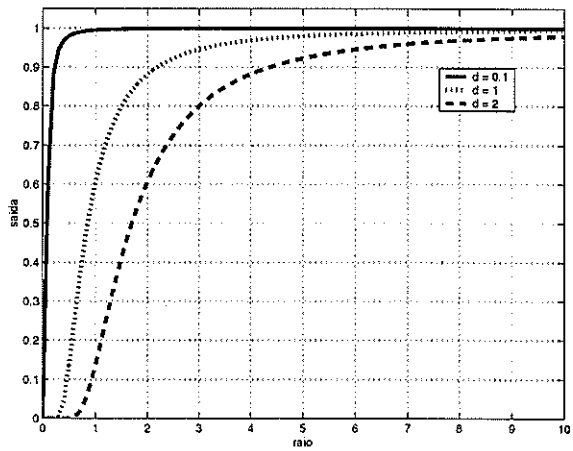

Figura 4.13: Resposta de função gaussiana segundo variação do parâmetro raio para três valores de distâncias.

Analisando as Figuras 4.12 e 4.13 pode-se observar que quanto maior a distância euclidiana entre um padrão de entrada e o centro de uma função de base ou menor for o raio, mais próximo de zero será o valor do respectivo elemento da matriz H.

Analogamente, a saída se aproxima da unidade para casos onde existe um alto valor de raio ou a distância entre o padrão de entrada e o centro de uma função de base seja próximo de zero. Não existem, portanto, elementos negativos ou superiores à unidade na matriz H de saída de neurônios da camada escondida.

Logo, se uma quantidade provoca o incremento do valor de alguma função de base, a norma de **H** também será afetada da mesma forma. Assim, a medida de complexidade para a camada escondida é função do comportamento conjunto de todas as funções radiais, dada por

$$\|\mathrm{H}\| = f(\mathrm{X}, \mathrm{C}, \mathrm{R}) = \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{c}\phi_{ij}^2}. \tag{4.1}$$

Sendo os elementos de **H** limitados entre [0,1], o intervalo de valores possíveis para $\|\mathbf{H}\|$ pertence a $[0, \sqrt{p.c}]$, onde p é o número de padrões de entrada e c o número de funções de base radial.

Dado uma disposição espacial **C** de centros escolhida aleatoriamente, o método k-médias encontra a disposição de posições de centros que minimiza a função de custo da Equacao 4.2 que representa a distância média entre os padrões de treinamento e centros de funções de base.

$$J = \sum_{h=1}^{k}\sum_{Q_h} \|\mathbf{x_q} - \mathbf{c_h}\|^2 \tag{4.2}$$

Em relação à construção da matriz **H**, o método k-médias pode ser visto como um problema de otimização que minimiza a distância média e maximiza a norma de **H** para um dado conjunto **X** de dados de treinamento e valores **R** de raios.

$$\mathrm{C}^* = \arg\max_{\mathrm{C}}(\|\mathrm{H}\|) \tag{4.3}$$

As Figuras 4.14 e 4.15 ilustram o comportamento da norma de **H**, segundo a aplicação do método k-médias para vários números de neurônios na camada escondida.

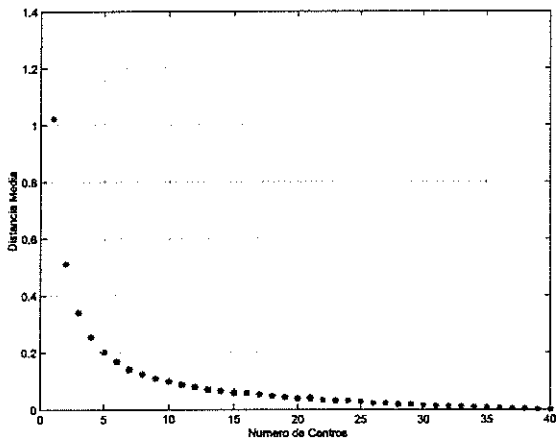

Figura 4.14: Distância média em função do número de centros.

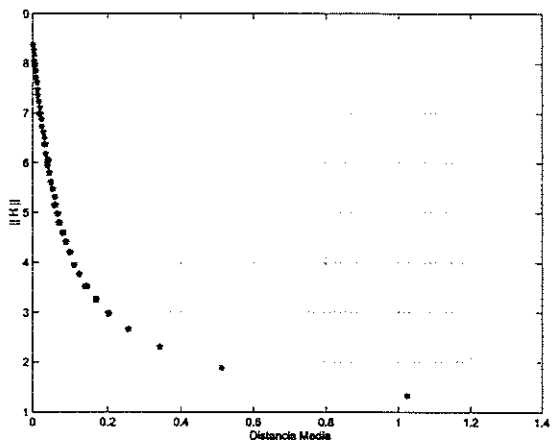

Figura 4.15: Norma de H em função das distâncias médias.

Uma vez que o algoritmo k-médias determina automaticamente as posições de centros que maximizam a norma de **H**, o parâmetro **C** fica pré-determinado em função do número de funções de base utilizados.

A influência no valor de raio para o valor da norma de **H** é uma relação análoga à questão da distância, conforme foi apresentado para as funções de base. Uma vez incrementado o valor de um elemento de **H**, representado pela saída de uma função de base, a norma de **H** também será incrementada. A Figura 4.16 apresenta um exemplo do comportamento da norma de **H** em função do valor de raio.

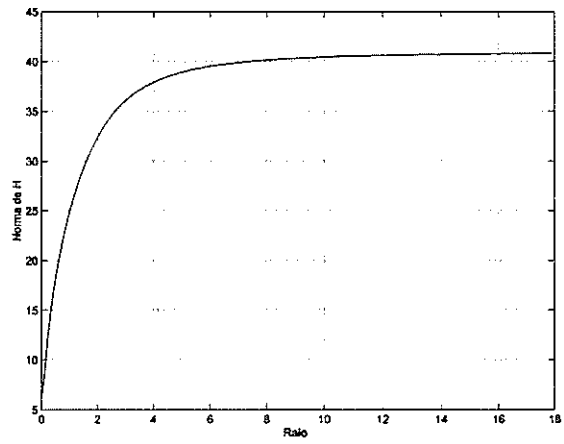

Figura 4.16: Exemplo de relação entre norma de H e valor de raio.

Pode-se concluir então que o valor de norma de **H**, representado por $\|\mathbf{H}\|$, é uma quantidade que leva em consideração a influência do raio e centros para caracterização de complexidade para a camada escondidada de RBFs.

A qualidade da solução encontrada por redes RBF está intimamente ligada ao condicionamento da matriz de **H**. Quando ocorre um mapeamento onde

existe perda de representatividade dos dados de entrada, a matriz **H** formada pela transformação não-linear dos dados de entrada possui um alto condicionamento (Equação A.8) e os vetores começam a se tornar linearmente dependentes. Isto pode ser representado pela diminuição do posto (*rank*) da matriz **H**.

A Figura 4.17 apresenta o comportamento dos erros de treinamento e validação para uma rede super-dimensionada onde os centros são coincidentes com os elementos do conjunto de validação. É possível visualizar o erro gerado pelo mal condicionamento de **H**, uma vez que os padrões se tornam linearmente dependentes como ilustra o comportamento do posto de **H** na Figura 4.18.

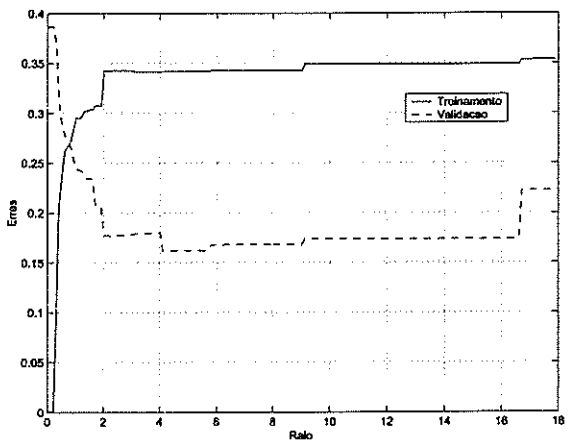

Figura 4.17: Comportamento de erro segundo valores de raio para rede super-dimensionada.

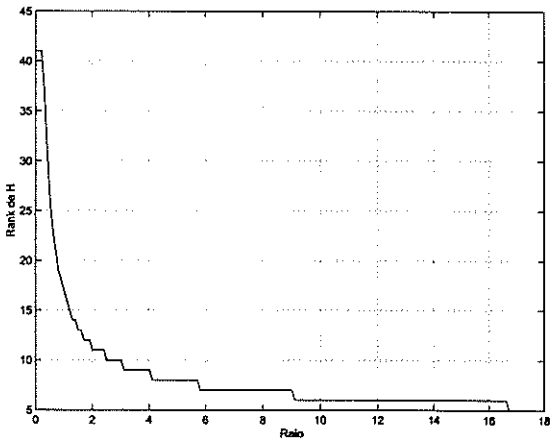

Figura 4.18: Comportamento do posto da matriz H segundo valores de raio para rede super-dimensionada.

O objetivo de se definir a combinação entre raios e centros está relacionada ao intervalo entre o valor mínimo de condicionamento (Equação A.7) e o valor determinado como mal-condicionado para um dado problema. A norma da matriz de interpolação ($\|\mathbf{H}\|$) é uma medida contínua e pode representar indiretamente o condicionamento de uma rede RBF.

As Figuras 4.19 e 4.20 apresentam os valores de condicionamento (em escala logarítmica) para a combinação de número de centros e valores de raio para redes RBFs. Pode-se observar nitidamente uma fronteira de alto incremento de condicionamento. Esta fronteira identifica o subespaço de combinações de centros e raios que resultam em soluções mal condicionadas.

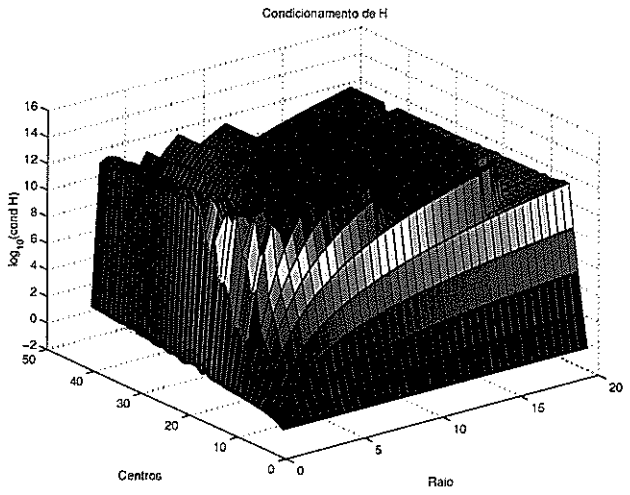

Figura 4.19: Superfície de caracterização de condicionamento da matriz de interpolação.

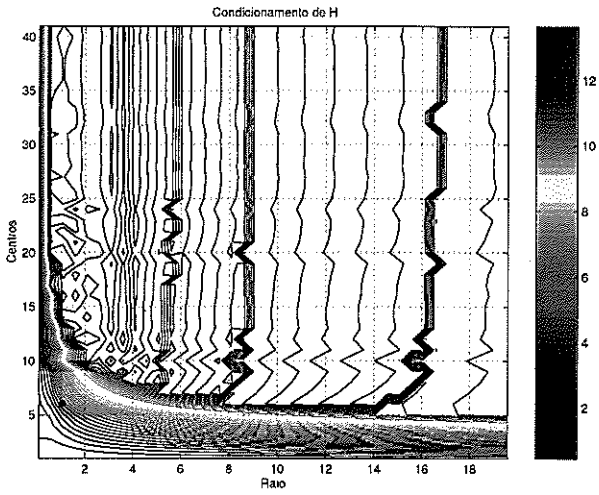

Figura 4.20: Curvas de nível de caracterização do condicionamento da matriz de interpolação.

Após explorado o comportamento das soluções segundo a variação dos parâmetros livres da camada escondida, pode-se assumir que a matriz de interpolação possui informações relevantes para treinamento de redes RBFs. A norma de $\mathbf{H}$ é uma medida contínua que leva em consideração os valores de raio e centros das funções de base radiais tendo relação direta com a qualidade do mapeamento não-linear, representado pelo condicionamento da mesma.

4.2.2 Caracterização de complexidade para a camada de saída

Pelo fato de a camada de saída das redes RBFs ser linear, como as redes MLP, a caracterização de complexidade desta camada fica estabelecida pela norma da matriz de pesos, representada por $\|\mathbf{W}\|$. As Figuras 4.21 e 4.22 apresentam um comportamento da norma da matriz de pesos para um caso simples de dois parâmetros livres.

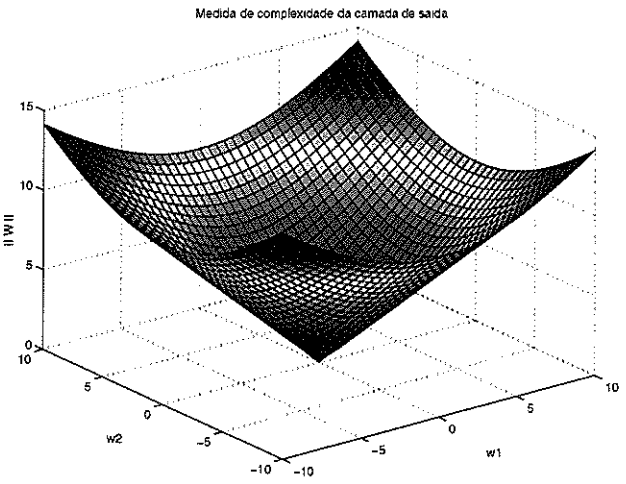

Figura 4.21: Superfície de caracterização de complexidade para camada de saída.

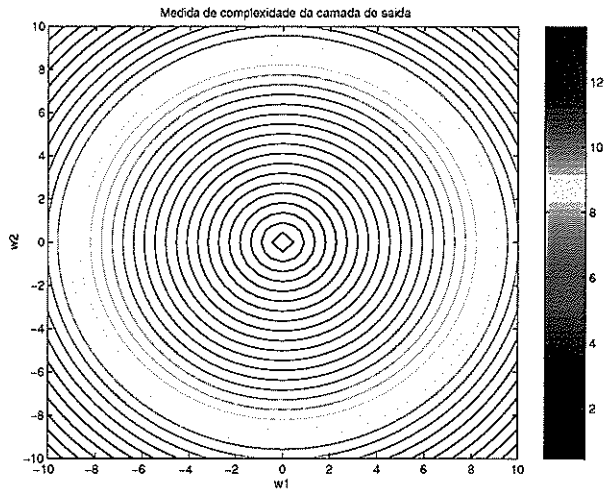

Figura 4.22: Curvas de nível da superfície de caracterização de complexidade para camada de saída.

A matriz de pesos é determinada a partir da saída dos neurônios da camada escondida, representada pela matriz **H**. Desta forma, o comportamento da norma de **W** é definido pelas características da matriz **H**, principalmente o seu condicionamento.

A influência do mal-condicionamento pode ser observada pelo aumento excessivo da norma da matriz de pesos conforme ilustram as Figuras 4.23 e 4.24.

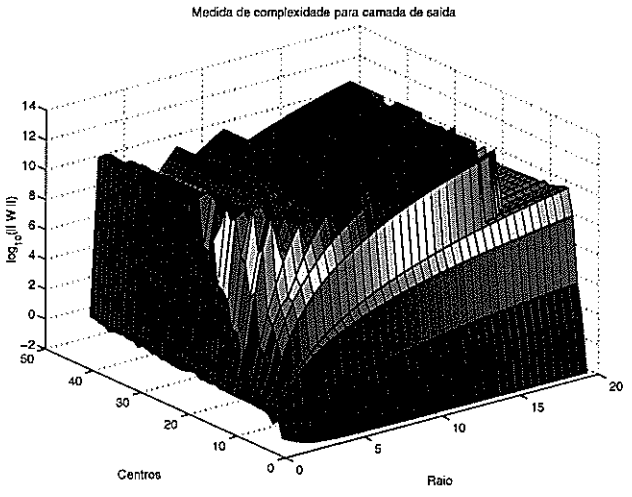

Figura 4.23: Superfície de caracterização de complexidade para camada de saída segundo valores de raio e centros.

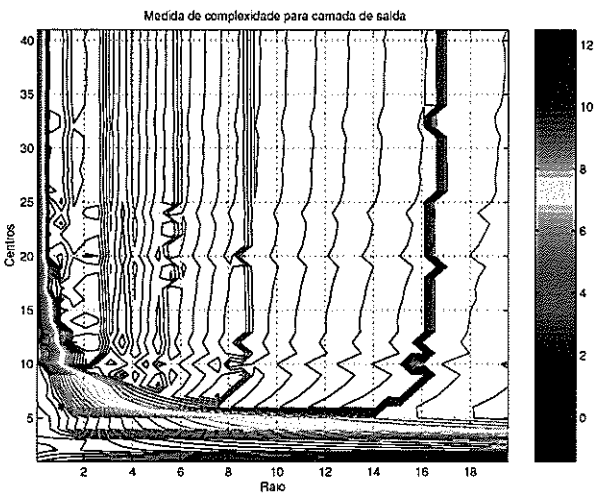

Figura 4.24: Superfície de caracterização de complexidade para camada de saída.

O comportamento da norma de pesos é muito relacionada com o condicionamento de **H**, uma vez que mapeamentos pouco representativos, caracteriza-

dos por alto condicionamento exigem uma complexidade maior para a camada de saida atingir baixos erros de treinamento. A Figura 4.25 ilustra a relação entre o condicionamento de **H** e a norma de **W** para várias combinações de centros e raios avaliadas.

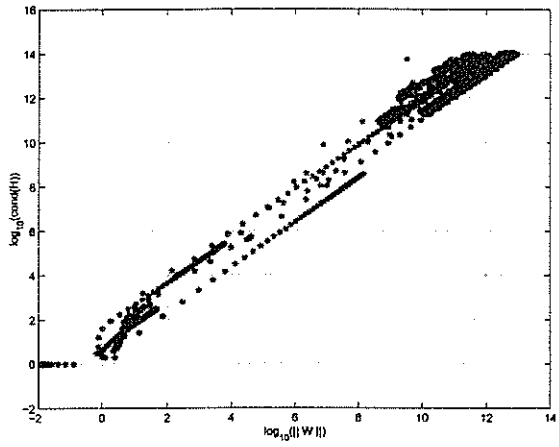

Figura 4.25: Exemplo de soluções para a relação entre norma de W e condicionamento de H.

Apesar de o aumento da norma da matriz de interpolação estar relacionado com o aumento da norma da matriz de pesos, não foi encontrada uma relação analítica direta entre estas duas grandezas. A seção seguinte apresentará um relação de desigualdade entre as relações de complexidade. Como não se pode inferir efetivamente na relação entre as duas camadas, as duas medidas de complexidade serão tratadas independentemente.

4.3 Relação de complexidades entre camadas

Utilizando uma análise de sensibilidade de sistemas lineares e sua solução por mínimos quadráticos é possível estabelecer uma relação entre as grandezas presentes nas duas camadas de uma rede RBF.

Uma matriz **H** inicial é estabelecida a partir de uma distribuição de centros e valores de raio muito baixos. Partindo de um valor de raio mínimo e o aumentando gradativamente, os elementos de H começam a aumentar sua magnitude. A matriz de interpolação modificada (H_2) pode então ser uma soma algébrica da matriz original H e uma matriz de perturbação representada por δH (Equação 4.4).

$$H_2 = H + \delta H \tag{4.4}$$

Sendo a matriz de saídas desejadas, Yd, constante durante o treinamento, a matriz de pesos também deverá ser modificada de modo a compensar a

perturbação δH (Equação 4.5).

$$\mathbf{W_2} = \mathbf{W} + \delta\mathbf{W} \tag{4.5}$$

Pode-se então analisar a influência da perturbação δH, causada pelo aumento do raio de suas funções radiais na matriz W de pesos da solução de uma rede RBF. Considerando-se um mesmo conjunto de treinamento, tem-se:

$$\begin{aligned}
\mathbf{Yd} &= \mathbf{H_2W_2} \\
\mathbf{Yd} &= (\mathbf{H}+\delta\mathbf{H})(\mathbf{W}+\delta\mathbf{W}) \\
\mathbf{W}+\delta\mathbf{W} &= (\mathbf{H}+\delta\mathbf{H})^{-1}\mathbf{Yd} \\
\delta\mathbf{W} &= -\mathbf{W} + (\mathbf{H}+\delta\mathbf{H})^{-1}\mathbf{Yd} \\
\delta\mathbf{W} &= [(\mathbf{H}+\delta\mathbf{H})^{-1} - \mathbf{H}^{-1}]\mathbf{Yd} \\
\delta\mathbf{W} &= [(\mathbf{M})^{-1} - \mathbf{H}^{-1}]\mathbf{Yd}
\end{aligned} \tag{4.6}$$

onde :

$$\begin{aligned}
\mathbf{M}^{-1} - \mathbf{H}^{-1} &= (\mathbf{H}^{-1}\mathbf{H})\mathbf{M}^{-1} - \mathbf{H}^{-1}(\mathbf{M}\mathbf{M}^{-1}) \\
\mathbf{M}^{-1} - \mathbf{H}^{-1} &= \mathbf{H}^{-1}(\mathbf{H}-\mathbf{M})\mathbf{M}^{-1}
\end{aligned} \tag{4.7}$$

substituindo 4.7 em 4.6,

$$\begin{aligned}
\delta\mathbf{W} &= [\mathbf{H}^{-1}(\mathbf{H}-\mathbf{M})\mathbf{M}^{-1}]\mathbf{Y} \\
\delta\mathbf{W} &= [\mathbf{H}^{-1}(\mathbf{H}-(\mathbf{H}+\delta\mathbf{H}))(\mathbf{H}+\delta\mathbf{H})^{-1}]\mathbf{Y} \\
\delta\mathbf{W} &= \mathbf{H}^{-1}(-\delta\mathbf{H})(\mathbf{H}+\delta\mathbf{H})^{-1}\mathbf{Y} \\
\delta\mathbf{W} &= \mathbf{H}^{-1}(-\delta\mathbf{H})(\mathbf{W}+\delta\mathbf{W})
\end{aligned}$$

e aplicando normas consistentes, tem-se a seguinte inequação:

$$\| (\delta\mathbf{W}) \| \leq \| \mathbf{H}^{-1} \| \| -\delta H \| \| (\mathbf{W}+\delta\mathbf{W}) \|$$

Substituindo $cond(H) = \| \mathbf{H}^{-1} \| \| \mathbf{H} \|$ tem-se:

$$\frac{\| (\delta\mathbf{W}) \|}{\| (\mathbf{W}+\delta\mathbf{W}) \|} \leq cond(\mathbf{H}) \frac{\| \delta\mathbf{H} \|}{\| \mathbf{H} \|} \tag{4.8}$$

Analisando a Equação 4.8 pode-se concluir que perturbações em H, oca-

sionadas, por exemplo, pelo aumento dos raios possibilitam perturbações relativas na matriz de pesos. Se as perturbações na matriz de pesos são altas então a norma também será. Conclui-se que o aumento da norma de H permite o aumento no valor da norma de pesos. Desta forma, a norma de H e a norma de W são correlacionadas e representam medidas de complexidade para a rede RBF.

4.4 Análise multi-objetivo de soluções para neurônios lineares

A relação entre complexidade de um neurônio e sua capacidade de ajuste é estendida também a camadas lineares de redes neurais artificiais. Apesar de constituírem modelos simples de tratamento de informação, a sua capacidade de representar o conhecimento exige um nível de complexidade ideal.

É possível representar o comportamento de neurônios lineares no espaço representado por erro e norma. As Figuras 4.26 e 4.27 apresentam o comportamento de complexidade e erro para neurônios lineares, constituintes da camada de saída de redes RBF.

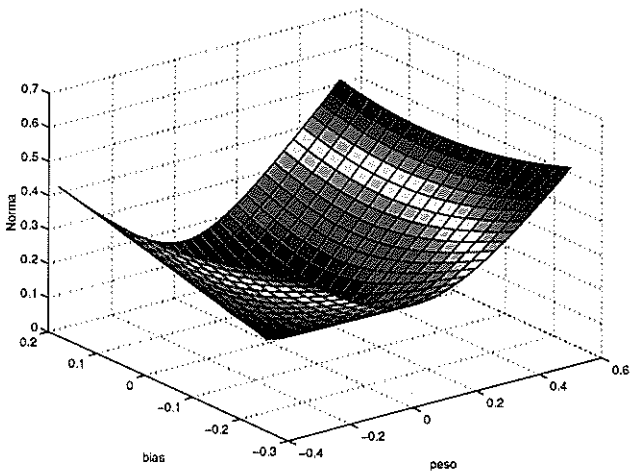

Figura 4.26: Exemplo de avaliação de complexidade para neurônios lineares.

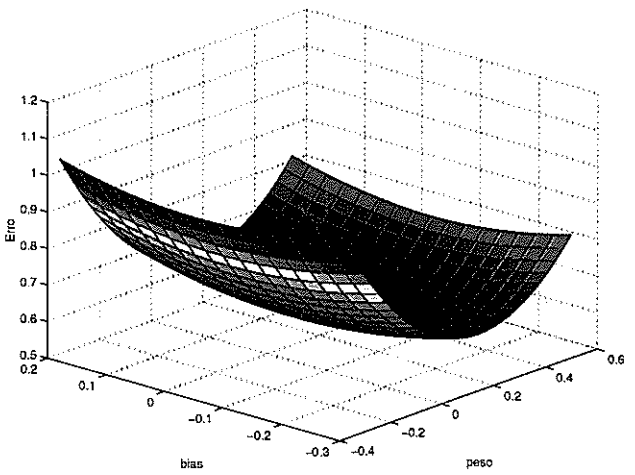

Figura 4.27: Exemplo de avaliação de erro para neurônios lineares.

As superfícies de erro de neurônios lineares são quadráticas em função aos parâmetros de otimização. Dentro do contexto multi-objetivo, pode se afirmar que na grande parte dos problemas, as funções de complexidade e erro possuem pontos de mínimos distintos, conforme ilustra a Figura 4.28.

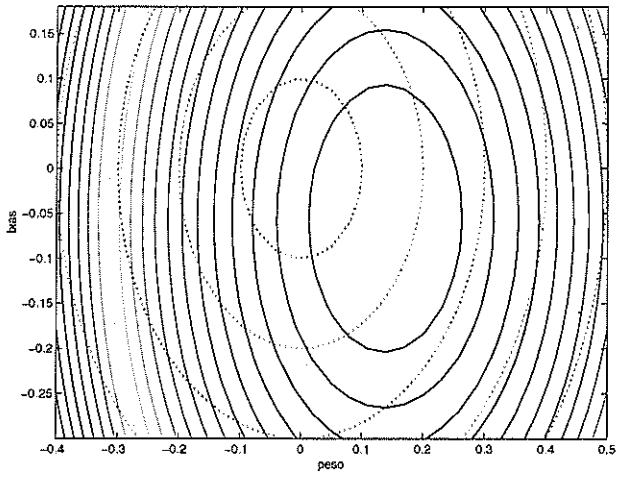

Figura 4.28: Exemplo de projeção das superfícies de erro e norma para neurônios lineares.

Ao se mapear todas as soluções no espaço de erro e norma é possível identificar um conjunto de soluções eficientes. As Figuras 4.29 e 4.30 apresentam um exemplo de caracterização de soluções no espaço definido pelo erro e norma, onde é possível identificar o espaço de soluções não-existentes e o conjunto de soluções eficientes.

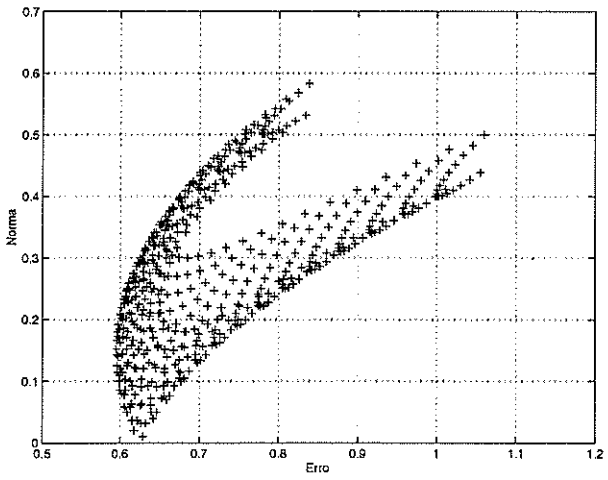

Figura 4.29: Exemplo de soluções para neurônios lineares no espaço erro e norma.

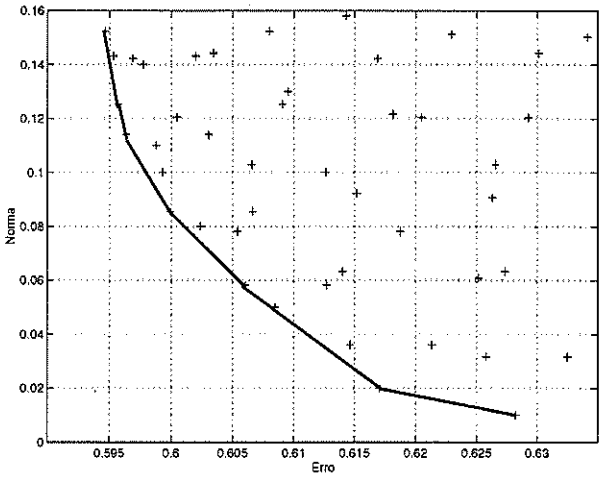

Figura 4.30: Exemplo de soluções eficientes para neurônios lineares no espaço erro e norma.

A teoria multi-objetivo utilizada para o ajuste de redes MLP pode ser aplicada para os pesos dos neurônios da camada de saída de redes RBF. O comportamento das soluções dadas por estes neurônios no espaço erro e norma apresenta um compromisso conflitante entre o erro e complexidade, justificando a utilização de treinamento multi-objetivo para determinação de parâmetros livres.

4.5 Análise multi-objetivo de soluções para neurônios de base radial

Como forma de ilustrar o comportamento multi-objetivo de redes RBFs segundo parâmetros de neurônios de base radial, foram caracterizadas várias soluções destas redes segundo variações de número de centros e valores de raio.

A Figura 4.31 apresenta a relação entre erro de treinamento e norma da matriz de interpolação para diversas soluções. Cada solução foi obtida pela minimização do erro de treinamento através da aplicação de mínimos quadrados sem nenhuma restrição. As mesmas soluções são também representadas em função da norma da matriz de pesos na Figura 4.32.

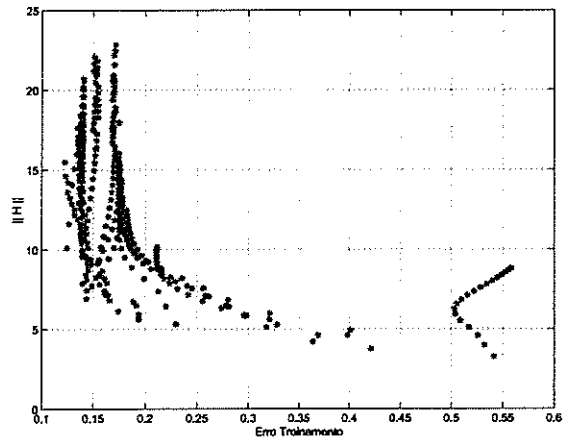

Figura 4.31: Exemplo de soluções no espaço erro e norma de H para diversas topologias RBF.

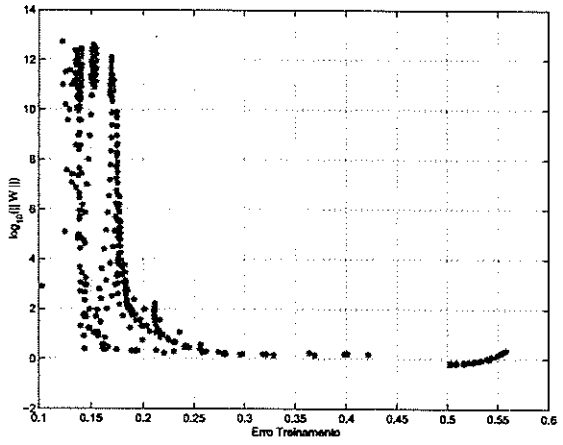

Figura 4.32: Exemplo de soluções no espaço erro e norma de W para diversas topologias RBF.

Analisando as Figuras 4.31 e 4.32 pode-se notar o compromisso entre o erro de ajuste e as complexidades, norma de **H** e norma de **W**. Conforme discutido, é possível verificar que soluções de erros inferiores necessitam de valores maiores de complexidade.

Apesar das medidas de complexidade não possuírem uma relação direta, para soluções de mínimo erro, um alto valor de norma de **H** implica também um alto valor de norma de **W**. A Figura 4.33 apresenta um exemplo da relação entre as complexidade de redes RBF.

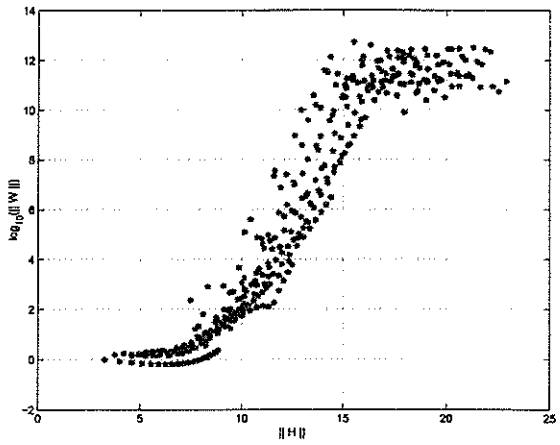

Figura 4.33: Exemplo de soluções no espaço norma de **H** e norma de **W** para diversas topologias RBF.

O comportamento das grandezas de erro e complexidade para camadas escondidas de redes RBFs também apresenta um compromisso conflitante entre as mesmas. As duas medidas de complexidade, mesmo não possuindo uma relação bem definida, apresentam uma mesma tendência de crescimento ao atingir valores menores de erro.

4.6 Treinamento Multi-Objetivo para redes RBFs

Em um processo de aprendizado de redes neurais, deseja-se encontrar uma solução que possua um erro reduzido para padrões de treinamento sem comprometer a capacidade de generalização do modelo neural. Através do método multi-objetivo, é possível encontrar o modelo que possui a complexidade mínima necessária para garantir um erro de treinamento reduzido sem perder a capacidade de responder satisfatoriamente a padrões desconhecidos.

A relação entre a complexidade de um modelo e o erro de treinamento é conflitante, pois modelos mais complexos permitem obter funções de alta flexibilidade permitindo um erro muito baixo. Soluções desta natureza são super-ajustadas possuindo uma alta variância em suas soluções. Quando o modelo possui baixa complexidade, o seu erro de treinamento aumenta e suas soluções são polarizadas, comprometendo novamente a capacidade de generalização de uma rede neural.

O controle da complexidade de uma rede RBF pode ser realizado através da magnitude da norma da matriz de interpolação e da norma da matriz de pesos independentemente da dimensionalidade de cada uma. Deseja-se encontrar o ponto de equilíbrio entre o erro para os padrões de treinamento e a complexidade do modelo neural para que os efeitos de polarização e variância sejam minimizados.

Pelo fato de as redes RBFs possuírem duas medidas distintas de complexidade, foram sugeridas algumas adaptações para a aplicação da teoria de treinamento multi-objetivo, como será descrito a seguir.

4.6.1 Descrição do Método

O método de treinamento multi-objetivo para redes RBFs consiste em contornar o problema da complexidade do modelo através da minimização simultânea da norma da matriz de interpolação, da norma da matriz de pesos e do erro para o conjunto de treinamento, conforme descrito na Equação 4.9.

$$\psi^* = arg_{\psi}\ min \begin{cases} f_1(\psi) = e_T(\psi) \\ f_2(\psi) = \|\mathbf{W}\| \\ f_3(\psi) = \|\mathbf{H}\| \end{cases} \tag{4.9}$$

As funções objetivo são representadas por

$$e_T = \frac{1}{p}\sum_{j=1}^{p}(\mathbf{d}_j - f(\mathbf{x}_j, \psi))^2 \tag{4.10}$$

$$\|\mathbf{H}\| = \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{h}\phi_{ij}^2} \tag{4.11}$$

$$\|\mathbf{W}\| = \sqrt{\sum_{i=1}^{h}\sum_{j=1}^{k}w_{ij}^2}. \tag{4.12}$$

A variável ψ representa o conjunto de parâmetros livres que fornecem uma solução para o treinamento de uma rede RBF. Os parâmetros livres podem ser representados pelas matrizes de posições dos centros (**C**), valores de raio (**R**) e pesos de conexões (**W**).

$$\psi = \{\mathbf{C}, \mathbf{R}, \mathbf{W}\} \tag{4.13}$$

$$\mathrm{C} = \begin{pmatrix} c_{11} & \cdots & c_{1h} \\ \vdots & & \vdots \\ c_{d1} & \cdots & c_{dh} \end{pmatrix} \tag{4.14}$$

$$\mathrm{R} = \begin{pmatrix} r_1 \\ \vdots \\ r_h \end{pmatrix} \tag{4.15}$$

$$\mathbf{W} = \begin{pmatrix} w_{11} & \cdots & w_{1k} \\ \vdots & & \vdots \\ w_{h1} & \cdots & w_{hk} \end{pmatrix} \tag{4.16}$$

Utilizando o método ε-restrito (Chankong and Haimes 1983), o problema de otimização multi-objetivo pode ser reescrito na forma mono-objetivo com restrições. Para o caso presente, deseja-se encontrar a solução de mínimo erro de treinamento para cada nível de complexidade. Desta forma o problema mono-objetivo é resolvido otimizando o somatório dos erros quadráticos para várias restrições de complexidade, representadas pelas limitações de norma de pesos (ε_W) e matriz de interpolação (ε_H) conforme apresenta o problema descrito por

$$\begin{aligned} \Psi^* &= \arg_{\Psi} \min\ e_T(\Psi) \\ \text{sujeito a}: & \begin{cases} g_1(\Psi) = \|\mathbf{H}\| \leq \varepsilon_H \\ g_2(\Psi) = \|\mathbf{W}\| \leq \varepsilon_W \end{cases} \end{aligned} \tag{4.17}$$

A variação paramétrica dos limites ε_W e ε_H permitirá a geração de uma aproximação do conjunto Pareto-ótimo segundo as três funções objetivo. Após definidos os δ valores de restrições para a norma de **H** e os ν valores de restrições para a norma de **W**, inicia-se a busca de soluções que atendem às restrições e possuem o valor mínimo de erro de treinamento.

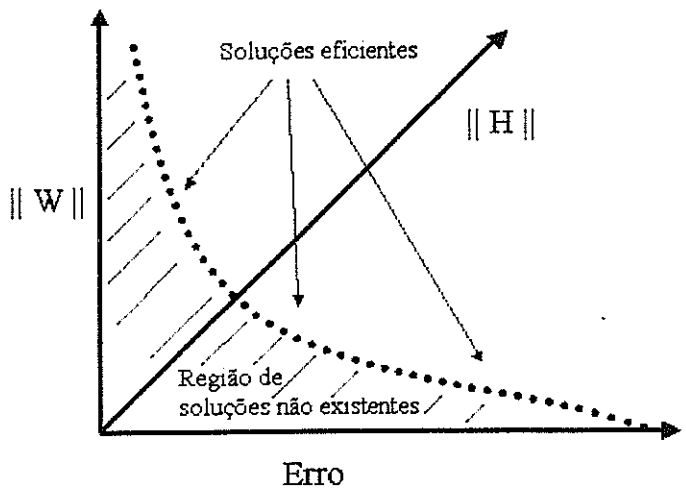

Figura 4.34: Exemplo de um conjunto de restrições de complexidade para redes RBFs.

4.6.2 Descrição do algoritmo de geração de soluções eficientes

A primeira etapa em um processo de treinamento multi-objetivo é a geração de soluções eficientes. Este conjunto de soluções, que constitui uma aproximação do conjunto Pareto-ótimo, pode ser obtido fazendo uso de qualquer método de otimização multi-objetivo, em especial, os algoritmos apresentados no Capítulo 3. Neste trabalho, utilizou-se o método ε-restrito para geração de soluções eficientes.

As variáveis de otimização para o problema de treinamento de RBFs possuem naturezas distintas. Desta forma, os métodos adotados para ajuste destes parâmetros podem ser aplicados independentemente.

O algoritmo parte de uma rede pré-dimensionada, onde o número de funções radiais é determinado pelo projetista. Conforme discutido anteriormente, as posições dos centros da funções radiais podem ser determinados pelo algoritmo k-médias (MacQueen 1967).

Uma vez determinadas as posições dos centros para uma topologia pré-definida, o conjunto de parâmetros livres a ser otimizado constitui dos raios e pesos das conexões, conforme o problema descrito pela Equação 4.18, onde $\psi = \{R, W\}$ corresponde ao conjunto de variáveis de otimização.

$$\begin{aligned} \psi^* &= \arg\min_{\psi} \; e_T(\psi) \\ \text{sujeito a} : & \begin{cases} g_1(\mathbf{R}) = \|\mathbf{H}\| \leq \varepsilon_H \\ g_2(\mathbf{W}) = \|\mathbf{W}\| \leq \varepsilon_W \end{cases} \end{aligned} \tag{4.18}$$

A geração de soluções eficientes é realizada através da variação dos limites para as funções de restrição. Primeiramente, estabelece-se um valor de ε_H limite que determinará o valor de raio para as funções de base de redes RBF segundo o problema descrito em 4.19. A solução deste problema pode ser encontrada através de uma busca sucessiva e incremental do valor de raio, caso se trabalhe com um valor comum para todas as funções de base.

$$\begin{aligned} \mathbf{R}^* &= \arg\max_{\mathbf{R}} \; \|\mathbf{H}\| \\ \text{sujeito a} : & \; g_1(\mathbf{R}) = \|\mathbf{H}\| \leq \varepsilon_H \end{aligned} \tag{4.19}$$

Dentro do contexto de treinamento multi-objetivo para redes neurais artificiais, espera-se que determinados valores de raios sejam capazes de gerar aproximações mais eficientes do conjunto pareto, ou seja, mais próximas do eixo das abcissas, onde se localizam as soluções de maior capacidade de generalização.

O valor de raio determinado na solução do problema da Equação 4.19 é en-

tão utilizado para a solução do segundo sub-problema, descrito pela Equação 4.20. Neste caso, pode-se fazer uso do algoritmo elipsoidal (Shor 1977) para a determinação da matriz de pesos ótima.

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} e_T$$
$$\text{sujeito a: } g_1(\mathbf{W}) = \|\mathbf{W}\| \leq \varepsilon_W \tag{4.20}$$

A Figura 4.35 apresenta exemplos de soluções obtidas para valores distintos de raios, ou seja, valores distintos de ||H|| para uma mesma arquitetura de rede RBF.

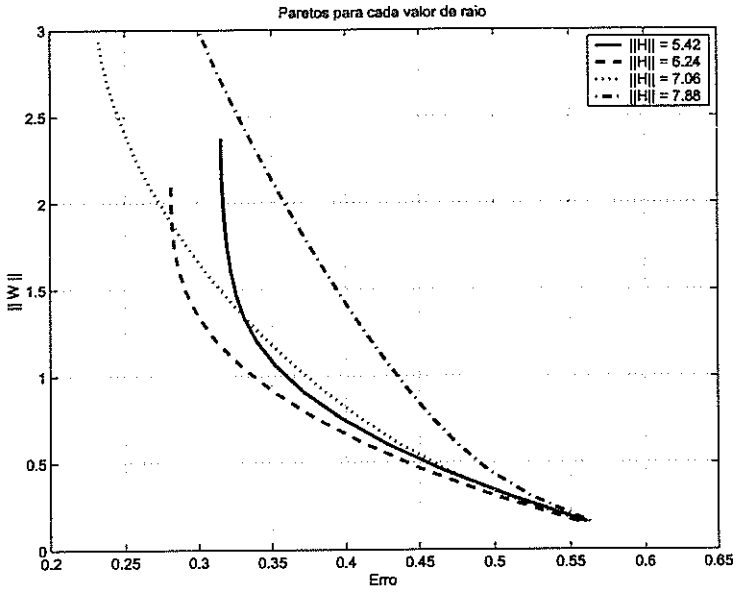

Figura 4.35: Exemplo de um conjunto de soluções eficientes para diversas restrições de H.

O conjunto de soluções obtidas com as restrições de norma de **H** e norma de **W** formam o conjunto de soluções que possui o menor valor de erro de treinamento para cada nível de complexidade. O conjunto de soluções Pareto-ótimo, presente neste conjunto de soluções, é representado por soluções onde não é possível minimizar uma função de custo sem causar o incremento de outra. Algumas soluções geradas neste algoritmo podem ser soluções dominadas ou não-eficientes necessitando desta forma utilizar alguma metodologia para seleção das soluções eficientes mas no trabalho atual trabalhou-se com todas as soluções geradas.

A próxima etapa consiste do processo de *Decisão*, onde uma solução deste conjunto é selecionada segundo algum critério estabelecido pelo *Decisor* para representar a solução de maior capacidade de generalização.

4.6.3 Decisor de solução final

De posse do conjunto de soluções eficientes, localizadas na fronteira da região de soluções factíveis e não-existentes, conforme ilustra a superfície das Figuras 4.36 e 4.37, utiliza-se um critério de seleção de solução final.

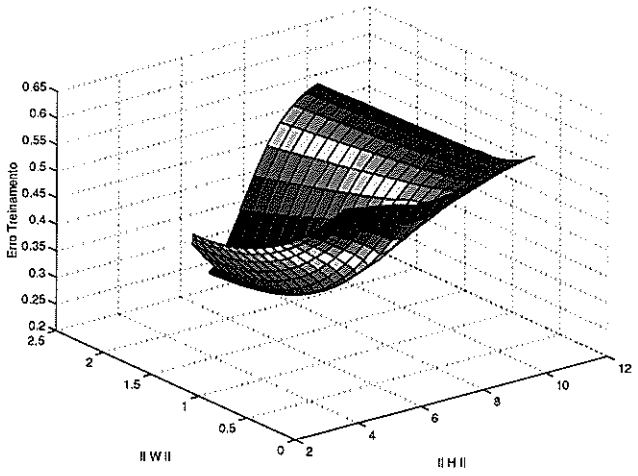

Figura 4.36: Superfície de decisão para os três objetivos.

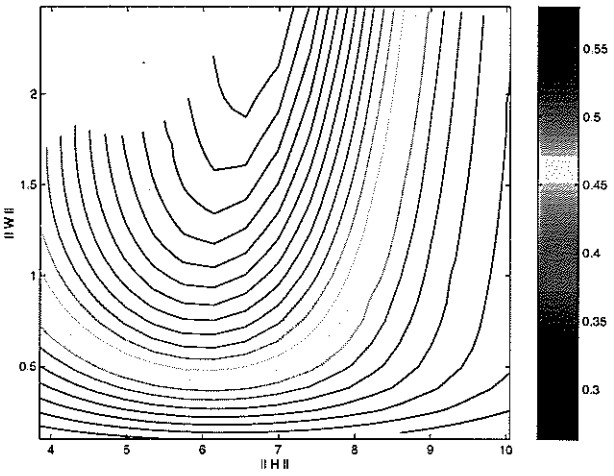

Figura 4.37: Projeção da superfície de decisão para os três objetivos.

Assim como no caso de redes MLP, o decisor utilizado para o algoritmo proposto se baseia no erro de ajuste para um conjunto de dados de validação. A solução de menor erro para o conjunto validação será selecionada como a solução de maior capacidade de generalização, ou seja, a solução que supostamente atenda

$$\psi^* = \arg\min_{\psi} e_V(\psi). \tag{4.21}$$

4.7 Descrição do algoritmo MOBJ-RBF

O Algoritmo 1 apresenta como o método multi-objetivo para treinamento de redes RBFs foi implementado fazendo uso do método ε-restrito. Além de ser simples, o método ε-restrito possibilita a geração das soluções candidatas ao modelo de maior capacidade de generalização.

4.8 Conclusões do capítulo

Uma nova proposta para treinamento de redes RBF foi apresentada neste capítulo. Esta nova metodologia faz uso da norma da matriz de interpolação como uma grandeza de complexidade para camada escondida de redes RBF. O capítulo seguinte apresentará algumas variações para o método original

Algorithm 1 Algoritmo MOBJ-RBF

Carregar conjunto treinamento; {*Vetores de entrada e saída*}
Carregar conjunto validação; {*Vetores de entrada e saída*}
$\delta \leftarrow$ Número de restrições de norma de H
$\nu \leftarrow$ Número de restrições de norma de W
$\Delta\delta \leftarrow$ Diferença de norma de H entre soluções
$\Delta\nu \leftarrow$ Diferença de norma de W entre soluções
$\varepsilon_H \leftarrow \Delta\delta$
$\varepsilon_W \leftarrow \Delta\nu$
$C_H \leftarrow 1$ {*Contador de modelos restritos de H*}
{*Gerador de soluções eficientes*}
while $C_H \leq \delta$ **do**
  Obter R^* que maximize $\|\mathbf{H}\|$ sujeito a: {*Utiliza Avaliação Sucessiva*}
  Restrição 1: $g_1(\zeta) = \|\mathbf{H}\| \leq \varepsilon_H$
  $C_W \leftarrow 1$ {*Contador de modelos restritos de W*}
  Inicializar W_0 {*Pesos randômicos, média zero e variância pequena*}
  while $C_W \leq \nu$ **do**
    Obter W^* que minimize $e_T(\mathbf{R}^*, W)$ sujeito a: {*Utiliza Algoritmo Elipsoidal*}
    Restrição 1: $g_2(\mathbf{W}) = \|\mathbf{W}\| \leq \varepsilon_W$
    $W_0 \leftarrow W^*$ {*Próxima solução se inicia no ponto ótimo da busca anterior*}
    Armazenar solução SOLUCAO(C_H, C_W) = $\{\mathbf{R}^*, W^*\}$
    $C_W = C_W + 1$
    $\varepsilon_W = \varepsilon_W + \Delta\nu$
  end while
  $C_H = C_H + 1$
  $\varepsilon_H = \varepsilon_H + \Delta\delta$
end while
{*Decisor de solução de alta capacidade de generalização*}
Obter $\{\mathbf{R}^*, W^*\}$ armazenado que minimize $e_V(\mathbf{R}^*, W)$

proposto apresentando alternativas para a geração otimizada de soluções eficientes como aproximação do conjunto Pareto-ótimo.

CAPÍTULO 5

Variações do método MOBJ-RBF

Após apresentados, no Capítulo 4, os fundamentos para treinamento multi-objetivo de redes RBFs, serão apresentadas algumas variações para o algoritmo proposto buscando uma maior eficiência computacional na busca de soluções de alta capacidade de generalização.

Serão abordados métodos para aceleração de busca de soluções eficientes através da otimização de raio (MOBJ-RBFr) e a aproximação de conjunto Pareto utilizando *Ridge Regression* (RR-MOBJ-RBF) e a proposição de *Subset Selection* para determinação automática de arquitetura através de otimização multi-objetivo (SS-MOBJ-RBF).

5.1 Introdução

Uma dificuldade para a aplicação de métodos multi-objetivos para treinamento de redes neurais, de acordo com sua proposição inicial (Teixeira 2001), está ligada à geração de várias soluções como aproximação do conjunto Pareto-ótimo. Não se sabe, a priori, qual o nível de complexidade ideal para um dado problema. Isto requer que sejam geradas soluções para diversos níveis de complexidade, aumentando o custo computacional dos algoritmos multi-objetivo.

O treinamento multi-objetivo para redes RBFs, apresentado no capítulo 4, requer o controle de duas medidas independentes de complexidade. Isto representa um custo computacional ainda maior, por se tratar um problema de otimização com três funções objetivo.

Como alternativas para a geração de soluções eficientes, serão apresentadas duas variações para o método MOBJ-RBF original. A primeira abordagem reduz o número de soluções através da otimização do valor de raio

(Carvalho, Costa, and Braga 2004), enquanto que a segunda variação apresenta uma alternativa de geração de conjuntos Pareto a partir de técnicas de regularização para os parâmetros lineares presentes na camada de saída.

A metodologia de treinamento MOBJ-RBF é aplicada em uma rede neural de topologia pré-definida. Neste capítulo apresenta-se uma variação do método MOBJ-RBF para busca automática de arquitetura utilizando *Subset Selection*.

5.2 Acelerando a geração de soluções eficientes

Algoritmos multi-objetivos para treinamento de redes neurais são capazes de encontrar soluções de alta capacidade de generalização, porém o tempo de processamento é muito afetado pelo número de soluções a serem geradas para representar o conjunto Pareto-ótimo.

A seguir serão apresentadas duas variações do método MOBJ-RBF para tornar a busca de soluções eficientes mais rápida.

5.2.1 Estimação automática de raios

O treinamento MOBJ-RBF faz uso de duas medidas independentes de complexidade aumentando a dimensão de busca bem como o número de soluções a serem geradas ao se aproximar o conjunto de soluções eficientes.

A variação MOBJ-RBFr dispensa o controle da norma da matriz de interpolação, deixando com que o próprio algoritmo de otimização determine o nível de complexidade, representado pelo valor de raio das funções de base, que minimize o erro de treinamento.

Sob a hipótese de que o nível de complexidade representado pela norma de $\mathbf{H}$ está relacionado ao nível de complexidade da norma de $\mathbf{W}$ (ver Figura 4.33), realiza-se a otimização conjunta dos valores de raio e pesos ($\psi = \{\mathbf{R}, \mathbf{W}\}$) segundo a minimização do erro de treinamento e complexidade da camada de saída.

$$\psi^* = \arg\min_{\psi} \begin{cases} f_1(\psi) = e_T(\psi) \\ f_2(\psi) = \|\mathbf{W}\| \end{cases} \tag{5.1}$$

O treinamento passa a utilizar somente dois objetivos podendo ser resolvido através da técnica de ε-restrito, utilizando o algoritmo elipsoidal para várias restrições de norma de W (ε_W), segundo o problema mono-objetivo descrito em 5.2.

$$\psi^* = \arg\min_{\psi} f(\mathbf{W}, \mathbf{R}) = e(\mathbf{W}, \mathbf{R})$$
$$\text{sujeito a} : g(W) = \|\mathbf{W}\| < \varepsilon_W \tag{5.2}$$

A solução do problema de otimização mono-objetivo (Equação 5.2) fornece um valor para o raio que minimiza o erro de treinamento para um determinado valor de restrição de $\|\mathbf{W}\|$. Isso significa que a partir de um valor de raio pré-definido, pode-se obter uma solução arbitrária no espaço de soluções e caminhar em direção a uma solução de erro reduzido obtendo uma solução que seja mais eficiente, como ilustra a Figura 5.1.

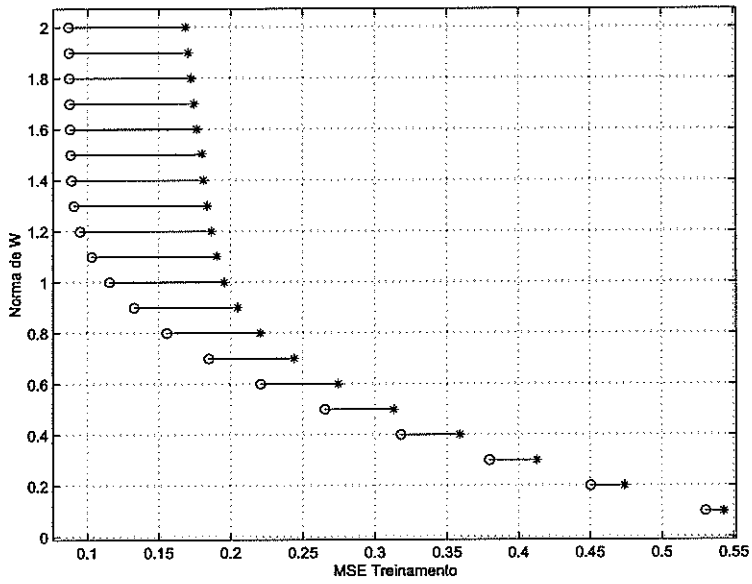

Figura 5.1: Deslocamento de soluções ocasionado pela otimização do valor de raio através da minimização do erro de treinamento. Os pontos assinalados com asteriscos são as soluções iniciais para cada restrição de norma de W e os círculos são as soluções após a otimização do valor de raio.

A otimização do valor de raio determina automaticamente o valor ótimo da norma da matriz de interpolação, uma vez que a mesma é função somente dos valores de raio. A técnica MOBJ-RBFr pode ser estendida para o caso de valores de raio independentes para cada função de base.

A Figura 5.2 apresenta um exemplo de aplicação da variação proposta. Observa-se que existe um valor ótimo de norma de **H** ou valor de raio que minimiza o erro de treinamento para cada restrição da norma de pesos.

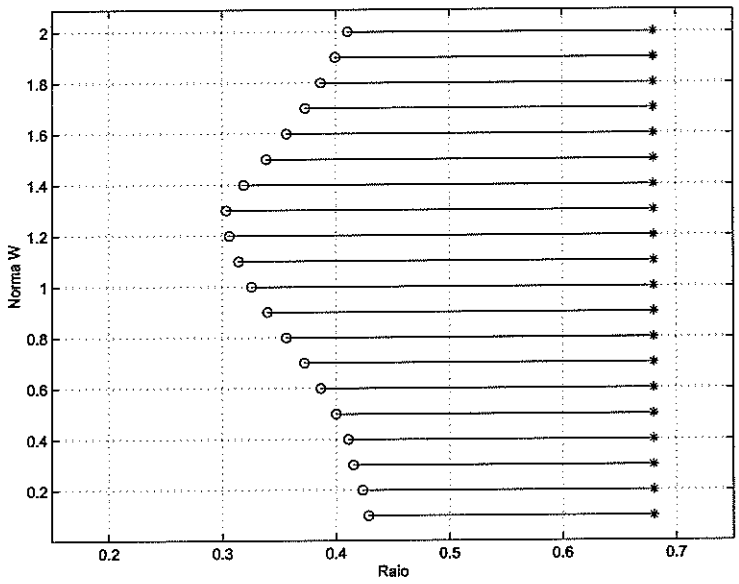

Figura 5.2: Deslocamento de soluções determinando valores distintos de raio para cada restrição de norma de W. Os pontos assinalados com asteriscos são as soluções iniciais para cada restrição de norma de W e os círculos são as soluções após a otimização do valor de raio.

Cada solução obtida possui um valor de raio que minimiza o erro de treinamento mantendo constante o valor de norma. Desta forma realiza-se uma geração automática de soluções candidatas à solução de maior capacidade de generalização, definindo também os valores dos pesos das conexões da camada de saída e os raios das funções de base da camada intermediária de redes RBF.

De posse de um número reduzido de soluções eficientes, basta aplicar um *Decisor* baseado em conjunto validação para a escolha do modelo de maior capacidade de generalização.

O Algoritmo 2 ilustra um pseudo-código para implementação do algoritmo MOBJ-RBFr.

O algoritmo MOBJ-RBFr (Carvalho, Costa, and Braga 2004) possui a grande vantagem de detectar automaticamente o valor de norma de **H** que garante a minimização do erro de treinamento, diminuindo assim o número de soluções a serem geradas para a aproximação do conjunto Pareto-ótimo.

5.2.2 *Estimação de soluções eficientes via Regularização*

Uma outra alternativa para a geração de soluções para a camada de saída, é utilizar técnicas de regularização para a minimização de erro através da penalização da norma dos pesos. Conforme apresentado no Capítulo 2, a

Algorithm 2 Algoritmo MOBJ-RBFr

Carregar conjunto treinamento; {*Vetores de entrada e saída*}
Carregar conjunto validação; {*Vetores de entrada e saída*}
$\nu \leftarrow$ Número de restrições de norma de W
$\Delta\nu \leftarrow$ Diferença de norma de W entre soluções
$\varepsilon_W \leftarrow \Delta\nu$
{*Gerador de soluções eficientes*}
$C_W \leftarrow 1$ {*Contador de modelos restritos de W*}
Inicializar $\mathbf{W}_0$ {*Pesos aleatórios, média zero e variância pequena*}
Inicializar $\mathbf{R}_0$ {*Definido segundo algum critério de dispersão de dados*}
while $C_W \leq \nu$ **do**
  Obter W^* que minimize $e_T(\mathbf{R}, \mathbf{W})$ sujeito a: {*Utiliza Algoritmo Elipsoidal*}
  Restrição 1: $g_2(\mathrm{W}) = \|\mathrm{W}\| \leq \varepsilon_W$
  $W_0 \leftarrow W^*$ {*Próxima solução se inicia no ponto ótimo da busca anterior*}
  Armazenar solução SOLUCAO(C_W) = $\{\mathrm{R}^*, \mathbf{W}^*\}$
  $C_W = C_W + 1$
  $\varepsilon_W = \varepsilon_W + \Delta\nu$
end while
{*Decisor de solução de alta capacidade de generalização*}
Obter $\{\mathrm{R}^*, W^*\}$ armazenado que minimize $e_V(\mathrm{R}^*, W)$

técnica de *Ridge Regression* (Orr 1996) permite obter soluções para os pesos da camada de saída através da penalização da magnitude destes parâmetros.

Assume-se que o conjunto de soluções eficientes pode ser aproximado pela solução de vários treinamentos com regularização. Neste caso, a parametrização da busca de soluções eficientes será realizada através da variação do termo de regularização λ, gerando soluções para diversas normas de $\mathbf{W}$.

A grande vantagem de se utilizar esta abordagem, é que a solução para a matriz de pesos que minimiza o erro de treinamento é dada analiticamente (Equação 5.3) (Orr 1996). Isto aumenta a eficiência computacional para geração de soluções para diversos valores de norma de $\mathbf{W}$.

$$\mathbf{W}^* = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}_h)^{-1}\mathbf{H}^T\mathbf{D} \tag{5.3}$$

Não são abordados neste trabalho, estudos para comparação da qualidade de aproximação do conjunto Pareto-ótimo a partir do método de ε-restrito e regularização. Pelo fato da camada de saída das redes RBF ser de natureza linear e os dois métodos possuírem funcionais convexos, as soluções obtidas são as mesmas para ambos os métodos, conforme foi observado nos testes realizados.

A diferença entre as soluções obtidas pelas duas técnicas, MOBJ-RBF e RR-MOBJ-RBF, está em como parametrizar λ e ε_W para o controle de complexidade. A Figura 5.3 apresenta um exemplo de aproximações de conjunto Pareto-ótimo geradas pelo método ε-restrito e o método de regularização.

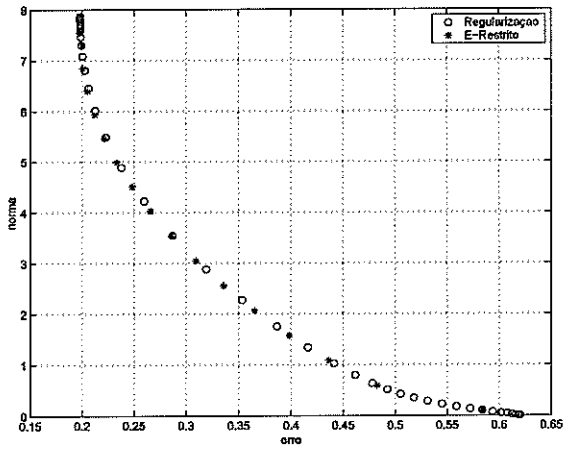

Figura 5.3: Estimativas de conjunto Pareto-ótimo

O exemplo da Figura 5.3 demonstra que a aproximação do conjunto Pareto-ótimo dado pela regularização e pelo ε-restrito podem ser muito semelhantes para a determinação de pesos da camada de saída. Os testes realizados tendem a confirmar esta semelhança porém não foi realizado nenhum estudo mais detalhado sobre este comportamento.

O Algoritmo 3 apresenta um pseudo-código para a aplicação do método de treinamento multi-objetivo com geração de soluções através de regularização.

Conforme pode ser observado, a regularização é uma alternativa viável para a geração de soluções eficientes de restrições de norma de $\mathbf{W}$. A mesma técnica pode ser estendida para o caso de valores distintos de penalização para cada conexão, basta que se utilize *Local Ridge Regression* (Orr 1996) para a estimação dos pesos da camada de saida.

5.3 Treinamento Multi-Objetivo com Subset Selection

O método de treinamento multi-objetivo para redes RBFs consiste em limitar a complexidade do modelo através da minimização simultânea da norma da matriz de interpolação, da norma da matriz de pesos e do erro para o conjunto de treinamento para uma determinada topologia pré-determinada.

A proposição do método SS-MOBJ-RBF consiste em realizar uma seleção adiante ("*foward selection*") de forma a encontrar a topologia mínima que garanta o nível de complexidade exigida para um dado problema.

Partindo de um modelo de complexidade mínima, o método SS-MOBJ-RBF deve determinar o número de neurônios da camada escondida e o valor de raio de modo a atingir o valor de norma da matriz de interpolação limitada por ε_H, resolvendo o problema descrito na Equação 5.4, onde $\zeta = \{\mathbf{C}, \mathbf{R}\}$ representa as posições de centro e valores de raio para as funções de base radiais.

Algorithm 3 Algoritmo RR-MOBJ-RBF

Carregar conjunto treinamento; {*Vetores de entrada e saída*}
Carregar conjunto validação; {*Vetores de entrada e saída*}
$\delta \leftarrow$ Número de restrições de norma de H
$\nu \leftarrow$ Número de parâmetros de regularização
$\Delta\delta \leftarrow$ Diferença de norma de H entre soluções
$\Delta\nu \leftarrow$ Diferença de magnitude dos parâmetros de regularização
$\varepsilon_H \leftarrow \Delta\delta$
$\lambda \leftarrow \Delta\nu$
$C_H \leftarrow 1$ {*Contador de modelos restritos de H*}
{*Gerador de soluções eficientes*}
while $C_H \leq \delta$ **do**
  Obter R^* que maximize $\|\mathbf{H}\|$ sujeito a: {*Utiliza Avaliação Sucessiva*}
  Restrição 1: $g_1(\zeta) = \|\mathbf{H}\| \leq \varepsilon_H$
  $C_W \leftarrow 1$ {*Contador de modelos restritos de W*}
  Inicializar W_0 {*Pesos randômicos, média zero e variância pequena*}
  while $C_W \leq \nu$ **do**
    Obter W^* que minimize $e_T(\mathbf{R}^*, W) + \lambda\|\mathbf{W}\|$ {*Utiliza Ridge Regression*}
    Armazenar solução SOLUCAO(C_H, C_W) = $\{\mathbf{R}^*, W^*\}$
    $C_W = C_W + 1$
    $\lambda = \lambda + \Delta\nu$
  end while
  $C_H = C_H + 1$
  $\varepsilon_H = \varepsilon_H + \Delta\delta$
end while
{*Decisor de solução de alta capacidade de generalização*}
Obter $\{\mathbf{R}^*, W^*\}$ armazenado que minimize $e_V(\mathbf{R}^*, W)$

$$\zeta^* = \arg_\zeta \max \|\mathbf{H}\|$$
$$\text{sujeito a} : g_1(\zeta) = \|\mathbf{H}\| \leq \varepsilon_H \tag{5.4}$$

A busca desta combinação pode ser realizada através do incremento sucessivo do número de centros e valores de raio até que se atinja a norma desejada da matriz de interpolação.

Como foi apresentado no Capítulo 3, existem várias combinações de número de centros e valores de raio para um mesmo valor de norma da matriz de interpolação. Desta forma, para cada solução de um subconjunto de soluções restritas de mesma norma de **H**, realiza-se um ajuste dos pesos que garanta o mínimo de erro de treinamento e seleciona-se apenas uma solução como representante de cada subconjunto. Aplica-se neste caso um *Decisor* baseado em erro de validação pois é interessante selecionar modelos de alta capacidade de generalização.

A Figura 5.4 apresenta um exemplo de 5 subconjuntos de modelos restritos, onde os pontos assinalados com um asterisco (*) indicam as soluções selecionadas pelo *Decisor* baseado em erro de validação.

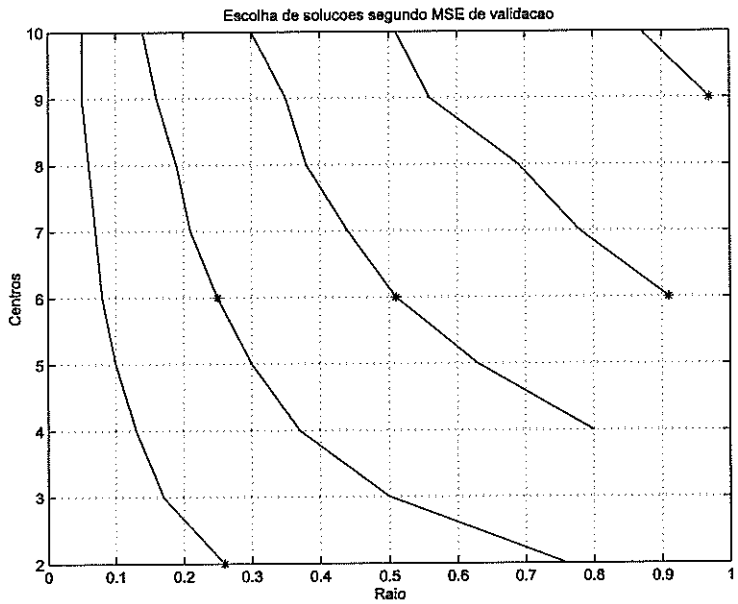

Figura 5.4: Conjuntos de soluções para 5 restrições de norma da matriz de interpolação. As soluções representadas por * correspondem as soluções selecionadas para representar cada restrição de norma da matriz de interpolação.

Para cada restrição de norma de **H**, uma configuração de centros e raios que minimiza o erro de validação será selecionada para prosseguimento da realização da otimização restrita dos pesos de conexão da camada de saída.

O problema, na qual serão submetidas as δ topologias, onde δ representa o número de restrições da norma de **H**, está descrito na Equação 5.5.

A geração de cada aproximação do conjunto Pareto para este problema, pode ser realizado utilizando o algoritmo elipsoidal (Shor 1977) para a minimização do erro de treinamento para ν valores distintos de restrições (ε_W) de norma da matriz de pesos.

$$\begin{aligned} \mathbf{W}^* &= \arg_{\mathbf{W}} \min\ e_T(\mathbf{W}) \\ \text{sujeito a} &: g_1(\mathbf{W}) = \|\mathbf{W}\| \leq \varepsilon_W \end{aligned} \tag{5.5}$$

A Figura 5.5 apresenta δ = 6 estimativas de conjuntos Paretos para ν = 10 restrições de norma de W.

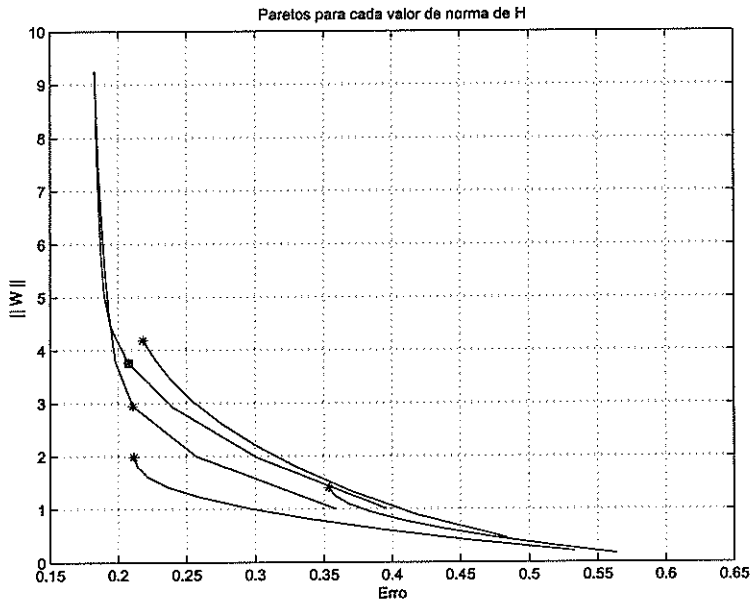

Figura 5.5: Superfície de caracterização de complexidade para camada de saída, onde * indicam as melhores soluções para cada estimativa de conjunto Pareto e o quadrado indica a solução final.

De posse das ν aproximações de conjunto Pareto, uma para cada restrição de norma da matriz de interpolação, um total de δ*ν modelos serão os candidatos à solução de melhor generalização. Novamente algum critério de decisão deve ser aplicado. Como sugestão, utiliza-se o erro de ajuste para o conjunto de validação. O modelo que possuir o valor mínimo de erro de validação será considerado o melhor modelo.

O Algoritmo 4 demonstra como o método multi-objetivo SS-MOBJ-RBF foi implementado fazendo uso do método ε-restrito.

O método SS-MOBJ-RBF possui a grande vantagem de selecionar automaticamente a topologia ideal para uma rede RBF, bem como o nível de com-

Algorithm 4 Algoritmo MOBJ-RBFss

Carregar conjunto treinamento; {*Vetores de entrada e saída*}
Carregar conjunto validação; {*Vetores de entrada e saída*}
$\delta \leftarrow$ Número de restrições de norma de H
$\nu \leftarrow$ Número de restrições de norma de W
$\Delta\delta \leftarrow$ Diferença de norma de H entre soluções
$\Delta\nu \leftarrow$ Diferença de norma de W entre soluções
$\varepsilon_H \leftarrow \Delta\delta$
$\varepsilon_W \leftarrow \Delta\nu$
$C_H \leftarrow 1$ {*Contador de modelos restritos de H*}
{*Gerador de soluções eficientes*}
while $C_H \leq \delta$ **do**
  {*Utiliza busca sucessiva incremental*}
  Obter conjunto de soluções $\Omega = \{\mathbf{R}_i, \mathbf{C}_i\}$ que maximize $\|\mathbf{H}\|$ sujeito a:
  Restrição 1: $g_1(\mathbf{R}, \mathbf{C}) = \|\mathbf{H}\| \leq \varepsilon_H$
  {*Utiliza Mínimos Quadráticos*}
  Obter $\mathbf{W}$ que minimize $e_T(\zeta^*, W)$ para cada solução de Ω
  {*Decisor de topologia*}
  Obter $\{R^*, C^*\} \in \Omega$ que minimize e_V
  $C_W \leftarrow 1$ {*Contador de modelos restritos de W*}
  Inicializar W_0 {*Pesos randômicos, média zero e variância pequena*}
  while $C_W \leq \nu$ **do**
    Obter W^* que minimize $e_T(W)$ sujeito a: {*Utiliza Algoritmo Elipsoidal*}
    Restrição 1: $g_2(\mathbf{W}) = \|\mathbf{W}\| \leq \varepsilon_W$
    $W_0 \leftarrow W^*$ {*Próxima solução se inicia no ponto ótimo da busca anterior*}
    Armazenar solução: SOLUCOES(C_H, C_W) = $\{R^*, C^*, W^*\}$
    $C_W = C_W + 1$
    $\varepsilon_W = \varepsilon_W + \Delta\nu$
  end while
  $C_H = C_H + 1$
  $\varepsilon_H = \varepsilon_H + \Delta\delta$
end while
{*Decisor de solução de alta capacidade de generalização*}
Obter $\{R^{**}, C^{**}, W^{**}\}$ armazenado em SOLUCOES que minimize $e_V(R^*, C^*, W^*)$

plexidade mínimo necessário para cada camada da rede.

5.4 Conclusões do capítulo

Neste capítulo foram apresentadas, variações para treinamento multi-objetivo de redes RBF. Os métodos apresentados neste capítulo não foram profundamente explorados para se garantir que suas soluções possuam uma alta capacidade de generalização. A eficiência de cada um foi somente avaliada e comparada com outros métodos para alguns testes que estão apresentados no capítulo seguinte.

CAPÍTULO 6

Aplicações do Método Proposto: Simulações e Resultados

Neste capítulo são apresentados alguns testes para avaliação da capacidade de aprendizagem e generalização do método proposto para alguns problemas de regressão. As variações do método proposto também serão avaliadas juntamente com outros principais métodos de treinamento de redes RBF.

6.1 Introdução

As redes neurais artificiais constituem-se em uma ferramenta muito eficiente para problemas de regressão, sendo muito utilizadas em tarefas de identificação e controle de sistemas (Carvalho, Medeiros, and Fortuna 2004) e predição de séries temporais (Costa, Braga, and Aguirre 2000).

São utilizados quatro problemas para avaliar a capacidade de realizar aproximações de funções e um problema para avaliação da capacidade de predição de séries temporais. As bases de dados utilizadas são as seguintes:

- Regressão
  - Função seno (com ruído gaussiano).

$$f(x) = \text{seno}(\pi x) + \varsigma \tag{6.1}$$

  - Função sinc (com ruído gaussiano).

$$f(x) = \frac{\text{seno}(\pi x)}{(\pi x)} + \varsigma \tag{6.2}$$

– Função d(x) (com ruído gaussiano).

$$d(x) = \frac{(x-2)(2x+1)}{(1+x^2)} + \varsigma \tag{6.3}$$

- Predição

  – Série caótica de Mackey-Glass

O termo ς representa a presença de um componente de ruído normalmente distribuído com média zero e variância $\sigma^2 = 0,2^2$.

Os resultados divulgados neste trabalho equivalem aos melhores resultados obtidos com cada metodologia para os mesmos conjuntos de dados. Diversas metodologias de treinamento de RBF foram comparadas com o método MOBJ-RBF e suas variações, dentre elas as técnicas de regularização (*Ridge Regression*) e seleção de subconjuntos (*Subset Selection*) originais (Orr 1997).

Para todos os algoritmos abordados procurou-se manter a mesma topologia da rede RBF a ser treinada. Mesmo não sendo a topologia ideal para cada problema, as topologias utilizadas são super-dimensionadas de forma a comparar as diversas metodologias para controle de complexidade. As funções de base são representadas por funções gaussianas e os neurônios de saída possuem ativação linear.

As soluções encontradas pela metodologia de regularização estão representadas pelo método *Ridge Regression* (RR) avaliado para alguns critérios de seleção de modelos (*GCV, BIC*). A implementação deste algoritmo se encontra disponível em (Orr 1999). O algoritmo disponível possui estimação para os valores de raio das funções de base através de um fator de escala independente para cada dimensão de cada função de base.

Para o treinamento de redes RBF segundo o algoritmo MOBJ-RBF proposto neste trabalho, quatro parâmetros devem ser fornecidos pelo projetista. Os valores são o número de soluções (δ) e a diferença ($\Delta\delta$) entre duas restrições consecutivas de norma de $\mathbf{H}$ e o número de soluções (ν) e a diferença ($\Delta\nu$) entre duas restrições consecutivas de norma de $\mathbf{W}$. Pode-se elaborar técnicas mais elaboradas para a parametrização da busca de soluções eficientes, constituindo uma proposta de trabalhos futuros.

Para a variação MOBJ-RBFr, estabelece-se inicialmente a quantidade de soluções (ν) e a diferença ($\Delta\nu$) entre duas restrições consecutivas de norma de $\mathbf{W}$, enquanto que o valor de norma de $\mathbf{H}$ é automaticamente determinado pela otimização do valor de raio.

A aproximação do conjunto Pareto pela técnica de regularização é utilizada no algoritmo RR-MOBJ-RBF. Para esta metodologia não se determina nada sobre a norma de $\mathbf{W}$, sua parametrização fica sujeita à escolha dos termos de

regularização λ utilizados para a geração de soluções eficientes.

Para os algoritmos com metodologias de *subset selection*, partiu-se da rede com um único nodo submetida a um limite de número máximo de neurônios a serem inseridos. Tomou-se a preocupação de que a resposta retornada pelo algoritmo esteja bem inferior ao limite máximo de complexidade imposto na busca de soluções.

O treinamento *NewRB*, disponível no *"Toolbox Neural Networks"* (Demuth and Beale 2001) para o *Matlab 6.1*, realiza uma seleção adiante (*"foward selection"*) incrementando um neurônio de função de base gaussiana a cada iteração, com o centro da gaussiana igual ao padrão de entrada mais distante de todos os centros. O algoritmo é executado até que se atinja o número máximo de neurônios permitidos ou ao atingir o valor de erro de treinamento desejado. Neste algoritmo, o valor de raio é pré-determinado pelo projetista e é único para todas as funções de base. Não existe um tratamento de aumento de generalização neste algoritmo, de forma que suas soluções dependem exclusivamente do número de neurônios máximo pré-determinado. Desta forma foi adicionada ao método original a utilização da técnica de *Cross-Validation*, onde se seleciona como solução final o modelo de menor erro para um conjunto de validação.

O método de *foward selection* (Orr 1999) utilizado faz uso de mínimos quadrados ortogonais para determinação de pesos da camada de saída. A implementação disponível, possibilita a utilização de *global ridge regression* e a determinação da melhor escala para o valor de raio. O modelo final é selecionado segundo algum critério de seleção de modelo, dentre os quais foram utilizados o Generalised Cross-Validation (GCV) e o Bayesian Information Criteira (BIC) (Orr 1996).

A variação SS-MOBJ-RBF, que utiliza *subset selection* para geração da topologia ideal em conjunto com a técnica de treinamento multi-objetivo, busca a melhor combinação de centros e raio para cada restrição de norma de **H**. O ajuste dos pesos de camada de saída é aplicado a cada configuração determinada para a camada escondida.

As soluções encontradas pelo método MOBJ-RBF e suas variações são comparadas com as respostas dos demais algoritmos abordados.

Faz-se ao final de cada exemplo uma análise da performance computacional entre o algoritmo MOBJ original e suas variações para aceleração de busca de soluções. Os testes foram realizados na mesma máquina nas mesmas condições de operação (Processador Duron XP2000 com 128 MB de RAM).

6.2 Exemplo 1: Regressão da Função Seno

No problema de regressão da função seno, foram geradas 80 amostras da função geradora $f_1(x) = \text{seno}(x)$ adicionadas a um ruído gaussiano de média zero e variância $\sigma^2 = 0,2^2$ para o conjunto de dados de treinamento. O conjunto de validação consiste em 30 amostras e o conjunto de dados para teste é formado por 400 amostras da função geradora sem a presença de ruído. A Figura 6.1 apresenta o conjunto de treinamento e validação para o problema de regressão Seno.

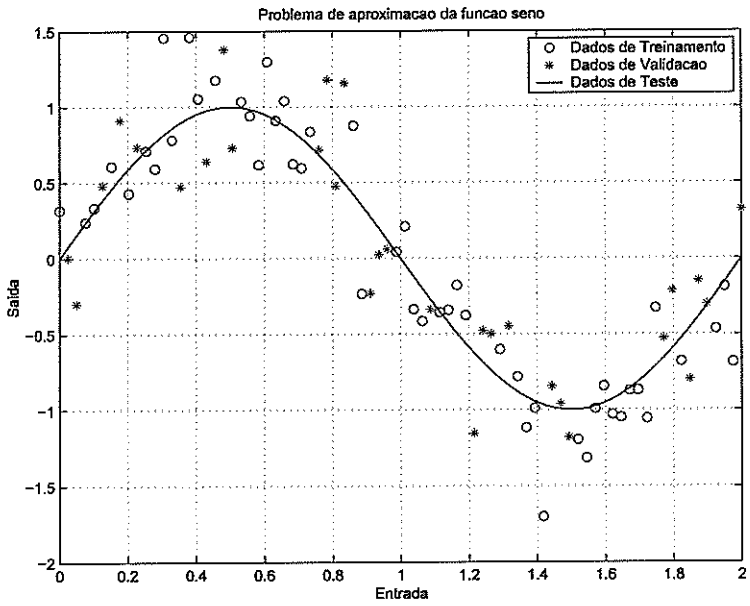

Figura 6.1: Conjunto de dados para problema de regressão Seno.

Foi adotada a topologia de 15 neurônios na camada escondida para todos os métodos de treinamento de redes RBF que não utilizam *subset selection*. Tal topologia é super-dimensionada para o problema em questão. Deseja-se desta forma, avaliar a capacidade de restringir a complexidade do modelo e a capacidade de busca de soluções de alta capacidade de generalização.

As Figuras 6.2 a 6.6 apresentam os resultados obtidos para o problema de regressão da função seno.

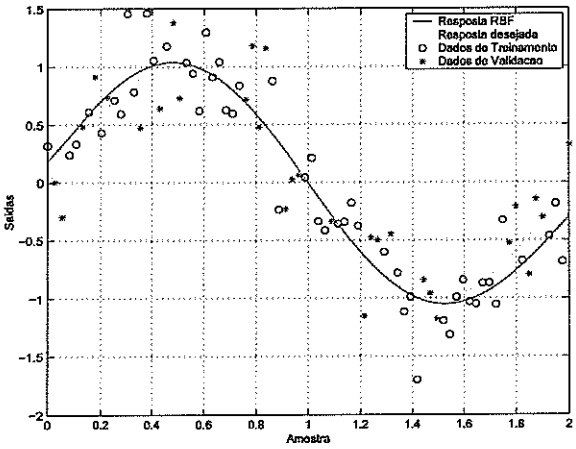

Figura 6.2: Solução de RR-GCV.

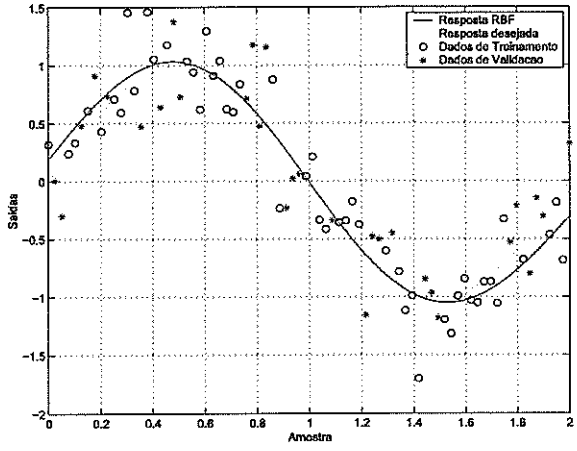

Figura 6.3: Solução de RR-BIC.

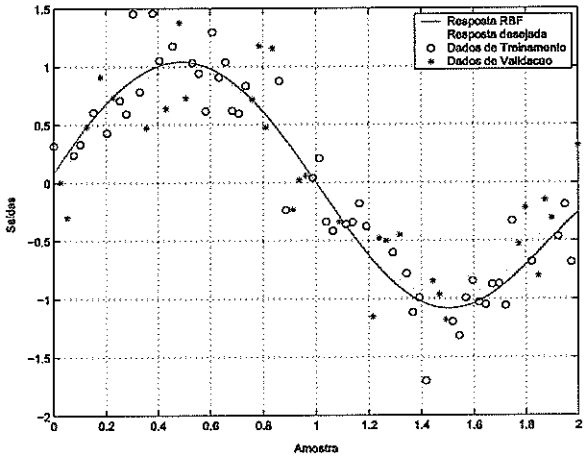

Figura 6.4: Solução de MOBJ-RBF.

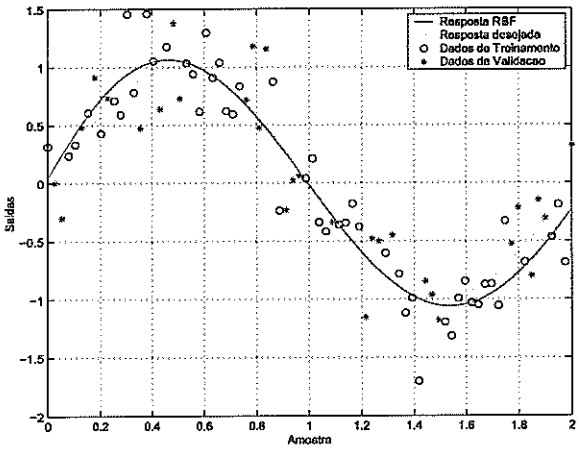

Figura 6.5: Solução de RR-MOBJ-RBF.

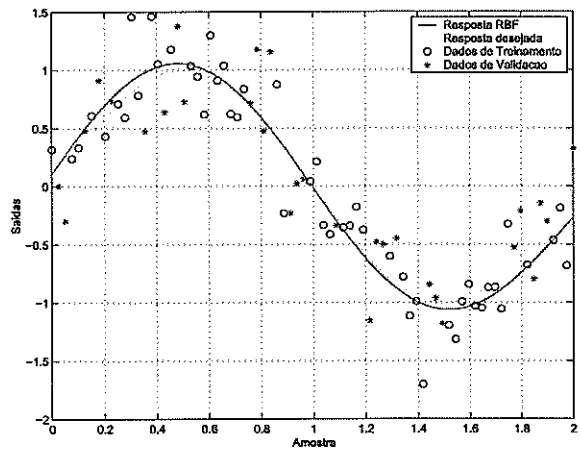

Figura 6.6: Solução de MOBJ-RBFr.

Nota-se que todas as soluções obtidas foram capazes de aproximar, com relativa fidelidade, a função geradora. Para quantificar a qualidade de cada resposta, são apresentados na Tabela 6.1 algumas quantidades caracterizadoras das redes RBF obtidas.

Tabela 6.1: Qualidade de soluções para problema de regressão Seno.

Quantidade	RR-GCV	RR-BIC	MOBJ-RBF	RR-MOBJ-RBF	MOBJ-RBFr
MSE Trein.	0,0644	0,0696	0,0639	0,0650	0,0641
MSE Valid.	0,1272	0,1284	0,1178	0,1223	0,1227
MSE Teste	0,0106	0,0111	0,0067	0,00962	0,0089
Neurônios	15	15	15	15	15
λ	0,0045	0.0037	-	$5{,}29\text{x}10^{-7}$	-
Raio	1,04	1,06	0,76	1,01	0,83
$\|H\|$	19,33	19,74	19,98	21,79	21,77
$\|W\|$	7,56	9,16	50,74	102,26	49,71

Analisando os erros para o conjunto de teste podes-se notar uma pequena melhoria nas soluções obtidas pelos métodos MOBJ. Pode-se destacar a resposta para o método MOBJ-RBF original que obteve o menor erro para o conjunto de teste.

Outras técnicas de treinamento de RBFs fazem uso de modificações em sua topologia a fim de encontrar a complexidade ideal para um dado problema. Tais modificações são feitas inserindo ou retirando neurônios da camada escondida com o uso de metodologias de *subset selection*.

As Figuras 6.7 a 6.6 apresentam as soluções obtidas pelas estruturas definidas utilizando técnicas de *subset selection* para cada método listado na Tabela 6.2.

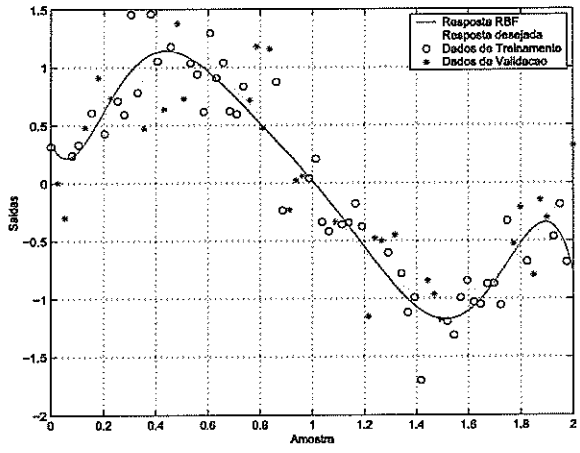

Figura 6.7: Solução de NewRB-GCV.

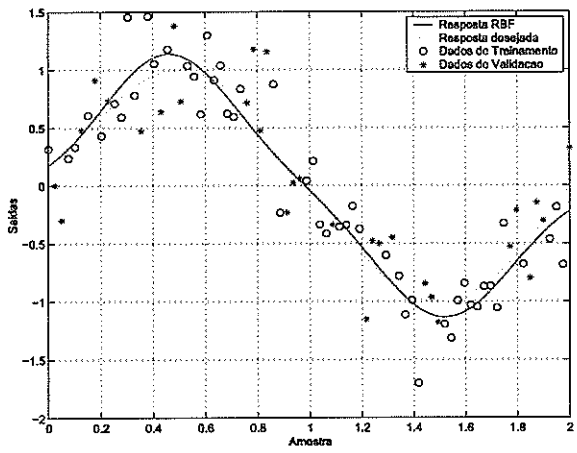

Figura 6.8: Solução de FS-GCV.

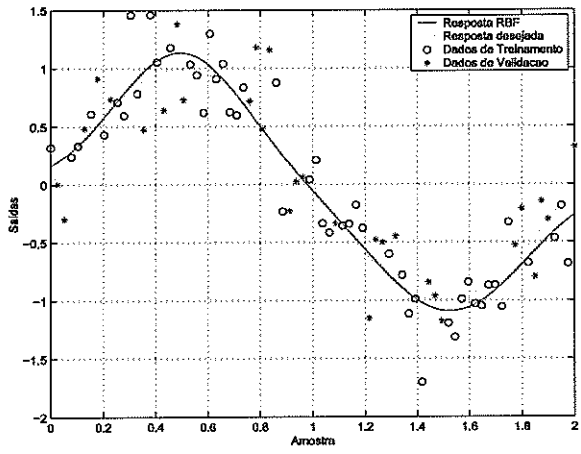

Figura 6.9: Solução de FS-BIC.

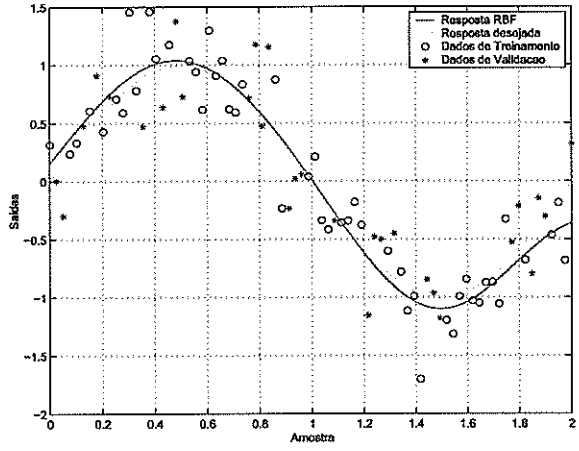

Figura 6.10: Solução de SS-MOBJ-RBF.

Tabela 6.2: Qualidade de soluções para problema de regressão Seno utilizando *subset selection*.

Quantidade	NewRB-CV	FS-GCV	FS-BIC	SS-MOBJ-RBF
MSE Trein.	0,0550	0,0590	0,0610	0,0615
MSE Valid.	0,1579	0,1318	0,1267	0,1257
MSE Teste	0,0174	0,0101	0,0079	0,0089
Neurônios	15	3	3	4
λ	-	0,1277	0,1549	-
Raio	2	0,37	0,35	0,34
$\|H\|$	-	6,03	5,95	9,04
$\|W\|$	-	2,04	1,43	2,07

Para a metodologia de *Subset Selection* a solução encontrada pelo método MOBJ proposto foi próxima da melhor solução obtida para o problema. Nota-se que todas os algoritmos, exceto o NewRB, convergiram para uma mesma topologia próxima de 3 neurônios na camada escondida.

As variações MOBJ-RBF propostas no trabalho visam acelerar a busca de soluções. Adotando a mesma topologia e mesmo conjunto de pontos foi possível comparar a eficiência computacional adquirida pelas variações MOBJ-RBFr e RR-MOBJ-RBF, conforme pode se observar na Tabela 6.3.

Tabela 6.3: Tempo gasto para treinamento MOBJ para problema Seno.

Algoritmo	Tempo
MOBJ-RBF	3310 segundos
MOBJ-RBFr	377 segundos
RR-MOBJ-RBF	17 segundos

Como conclusão geral para o problema de regressão seno, pode-se admitir que as soluções obtidas pelos métodos MOBJ alcançaram resultados próximos e algumas vezes ligeiramente superiores aos obtidos com outras metodologias. As variações propostas também alcançaram resultados satisfatórios e diminuiram o tempo gasto para o treinamento de redes RBF.

6.3 Exemplo 2: Regressão da Função d(x)

A função *d(x)* apresenta regiões de resposta suave em seus extremos seguida de uma região central de grande variação, conforme ilustra a Figura 6.11. O conjunto de dados de treinamento é formado por 50 amostras e 30 amostras para conjunto de validação. O conjunto de teste consiste de 400 amostras da função geradora sem a presença de ruídos.

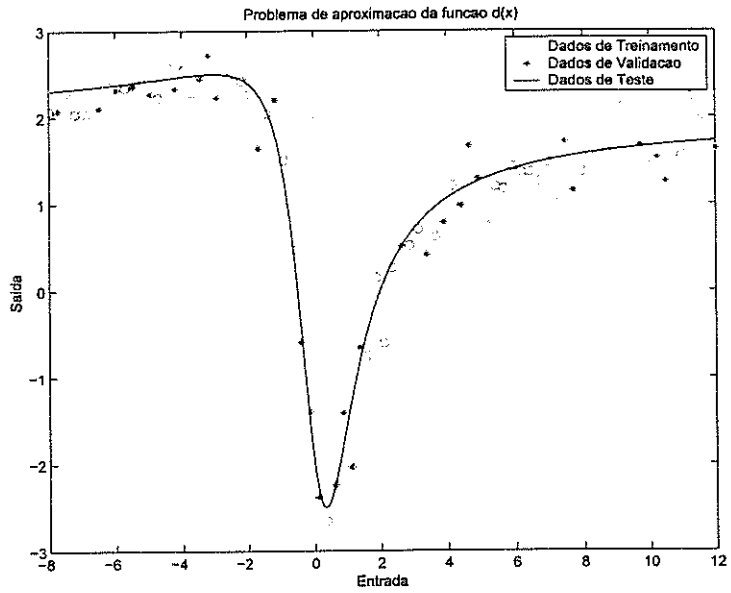

Figura 6.11: Conjunto de dados para problema de regressão da função d(x).

Foram utilizadas arquiteturas de 20 neurônios na camada escondida para todos os métodos. Nota-se nas Figuras 6.12 a 6.16 que todas as soluções encontradas foram capazes de aproximar bem a região central, enquanto que os extremos ficaram distorcidos pelo ruído presente. Isto ocorre porque as redes estão todas sobre-ajustadas, ou melhor, as redes possuem, intencionalmente, um número excessivo de neurônios na camada escondida. Isto foi necessário para avaliar a capacidade de se ajustar a complexidade do modelo entre todos os algoritmos em estudo.

As soluções obtidas estão caracterizadas na Tabela 6.4. Pode-se analisar que as respostas dadas pelos algoritmos MOBJ-RBF são de qualidade comparáveis com as outros algoritmos de treinamento.

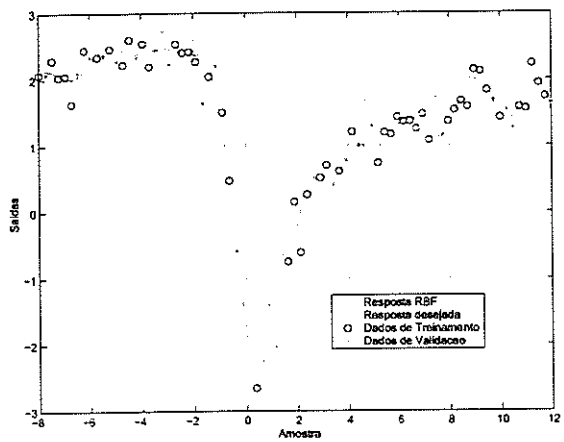

Figura 6.12: Solução de RR-GCV.

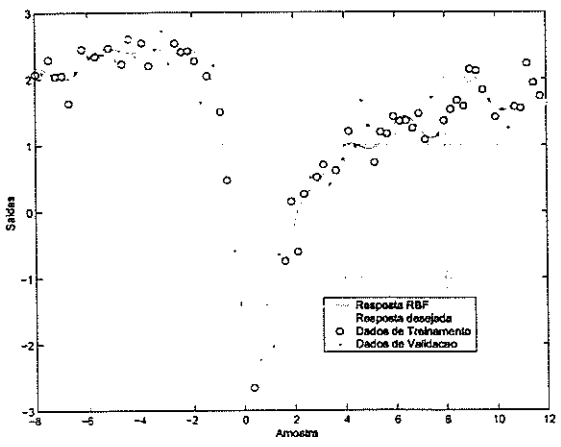

Figura 6.13: Solução de RR-BIC.

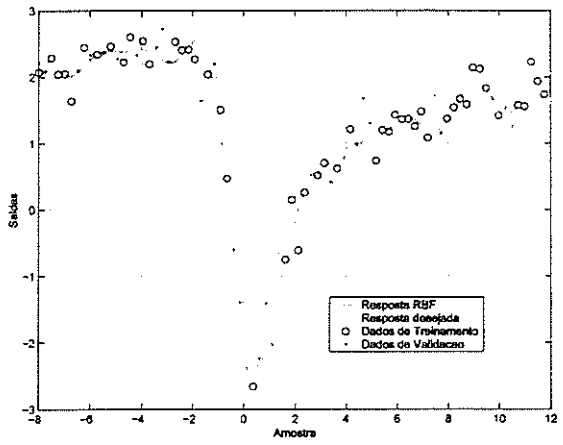

Figura 6.14: Solução de MOBJ-RBF.

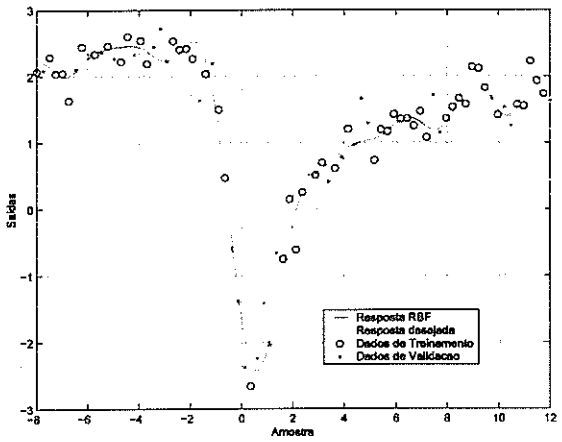

Figura 6.15: Solução de RR-MOBJ-RBF.

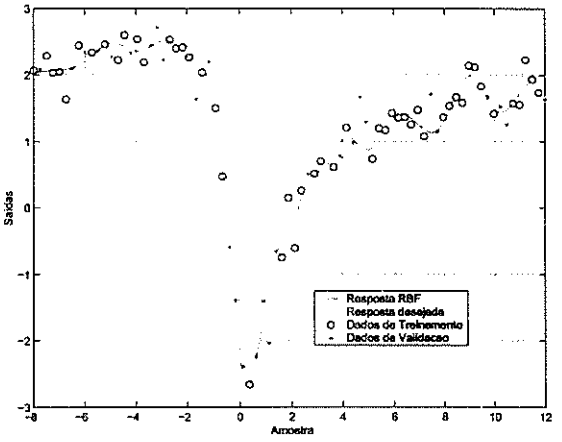

Figura 6.16: Solução de MOBJ-RBFr.

Tabela 6.4: Qualidade de soluções para problema de regressão d(x).

Quantidade	RR-GCV	RR-BIC	MOBJ-RBF	RR-MOBJ-RBF	MOBJ-RBFr
MSE Trein.	0,0344	0,0369	0,0447	0,0415	0,0341
MSE Valid.	0,1285	0,1316	0,1087	0,1263	0,1041
MSE Teste	0,0512	0,0489	0,0382	0,0511	0,0376
Neurônios	20	20	20	20	20
λ	2.10^{-6}	$1{,}19.10^{-4}$	-	$1{,}0.10^{-8}$	-
Raio	1,97	1,77	1,15	1,3	0,71
$\|\|H\|\|$	10,88	10,37	10,03	10,50	12,80
$\|\|W\|\|$	278,10	38,13	18,22	51,10	7,67

Analisando o erro para o conjunto de teste, pode-se notar que as soluções dos algoritmos MOBJ possuem um ajuste ligeiramente superior, ilustrando a

capacidade dos algoritmos propostos neste trabalho de encontrar soluções de alta capacidade de generalização.

Adotando a técnica de *subset selection*, cada algoritmo foi capaz de determinar uma topologia ideal para o problema de regressão deste exemplo. As Figuras 6.7 a 6.10 apresentam as soluções obtidas e a Tabela 6.5 caracteriza cada solução encontrada por estes algoritmos.

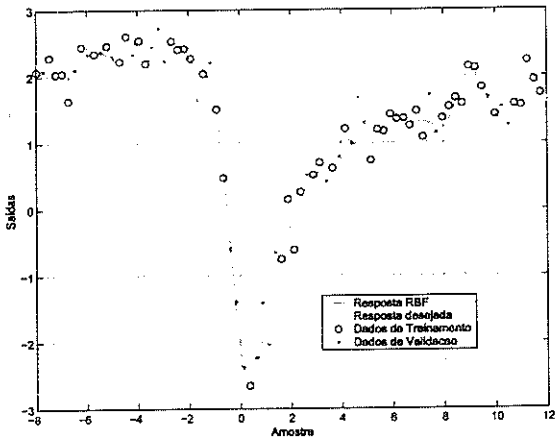

Figura 6.17: Solução de NewRB-GCV.

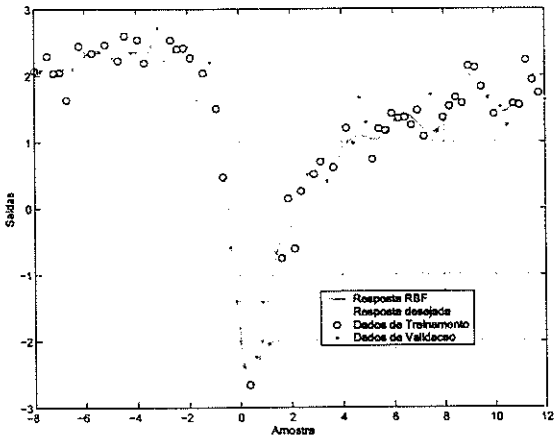

Figura 6.18: Solução de FS-GCV.

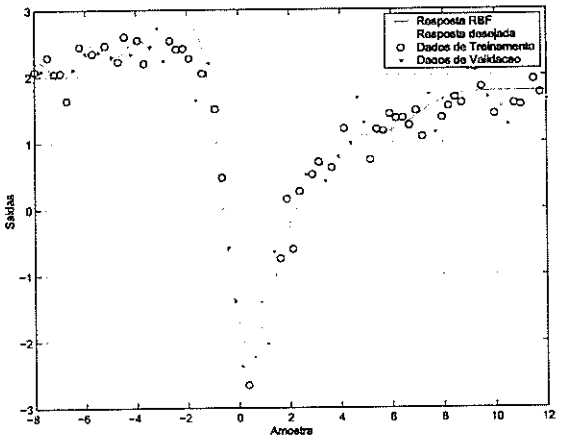

Figura 6.19: Solução de FS-BIC.

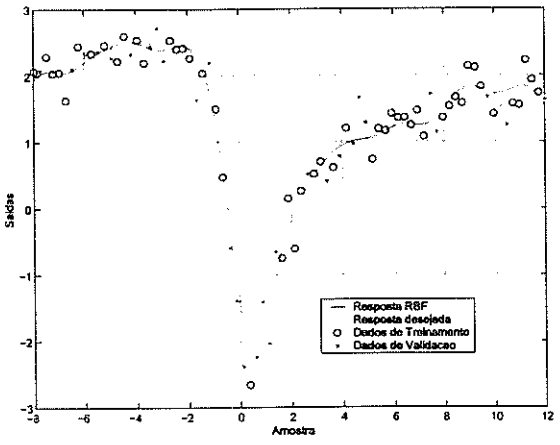

Figura 6.20: Solução de SS-MOBJ-RBF.

Tabela 6.5: Qualidade de soluções para problema de regressão d(x) utilizando *subset selection*.

Quantidade	NewRB-CV	FS-GCV	FS-BIC	SS-MOBJ-RBF
MSE Trein.	0,0294	0,0387	0,0679	0,0515
MSE Valid.	0,0909	0,0993	0,1181	0,1123
MSE Teste	0,0397	0,0350	0,0381	0,0319
Neurônios	20	11	12	19
λ	-	0,0205	$9{,}7.10^5$	-
Raio	0,7	0,987	1,77	0,81
$\|H\|$	-	9,28	10,61	8,36
$\|W\|$	-	6,31	255,97	3,97

Novamente pode-se observar que a solução encontrada pelo método MOBJ-RBF é ligeiramente superior aos outros algoritmos para treinamento de redes RBF.

Tabela 6.6: Tempo gasto para treinamento MOBJ para problema Função d(x).

Algoritmo	Tempo
MOBJ-RBF	4020 segundos
MOBJ-RBFr	1933 segundos
RR-MOBJ-RBF	18 segundos

Analisando o custo computacional entre as técnicas para aceleração de busca de soluções, pode-se comprovar que as variações propostas exigiram um tempo muito inferior para realização do treinamento e obtendo soluções de erros próximos.

6.4 Exemplo 3: Regressão da função Sinc

O problema de regressão da função *sinc* é representado por 50 amostras para conjunto treinamento e 30 para conjunto validação, conforme ilustra a Figura 6.21. O conjunto validação é constituido pela função geradora enquanto o conjunto de treinamento e validação são representados por amostras da função geradora $f(x) = \text{sinc}(x)$ adicionada de um ruído gaussiano de média nula e variância $\sigma^2 = 0,2^2$.

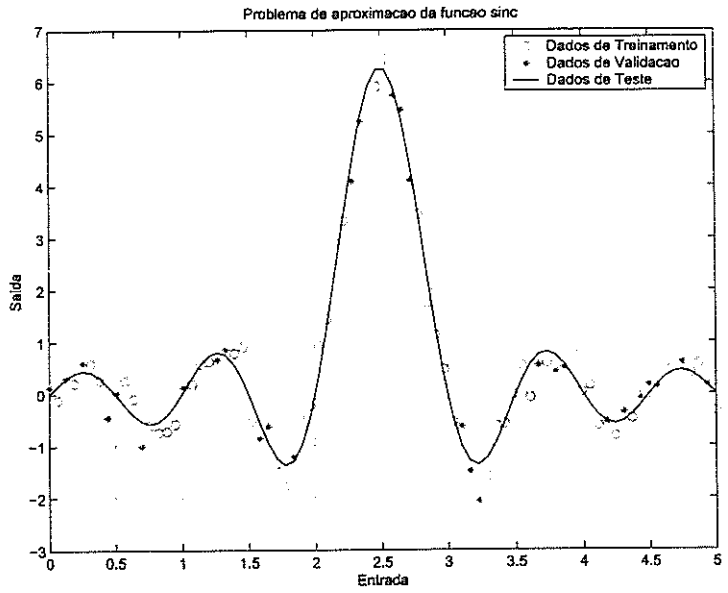

Figura 6.21: Conjunto de dados para problema de regressão Sinc.

A topologia escolhida para a comparação de métodos foi de 20 neurônios na camada escondida. As soluções obtidas para cada algoritmo de treinamento estão ilustradas nas Figuras 6.22 a 6.26.

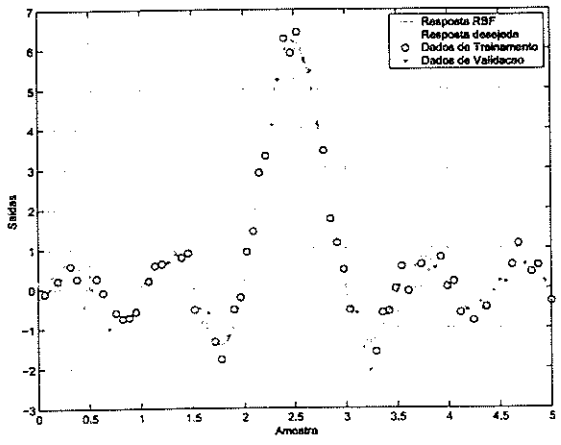

Figura 6.22: Solução de RR-GCV.

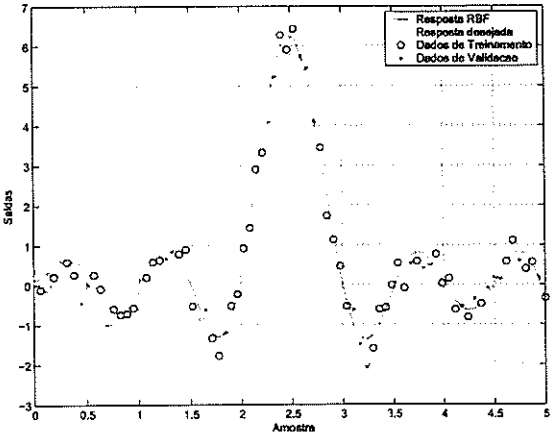

Figura 6.23: Solução de RR-BIC.

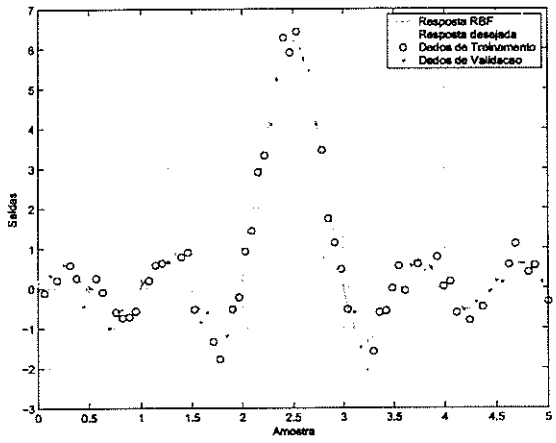

Figura 6.24: Solução de MOBJ-RBF.

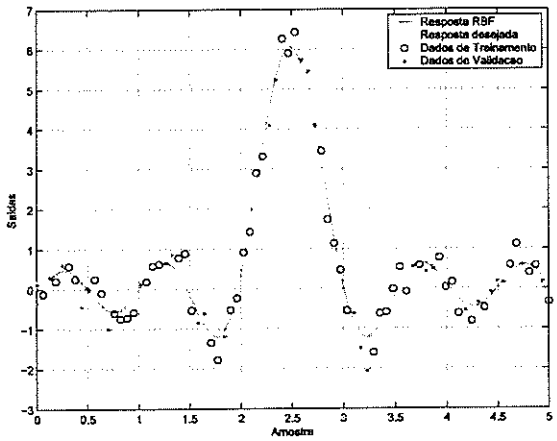

Figura 6.25: Solução de RR-MOBJ-RBF.

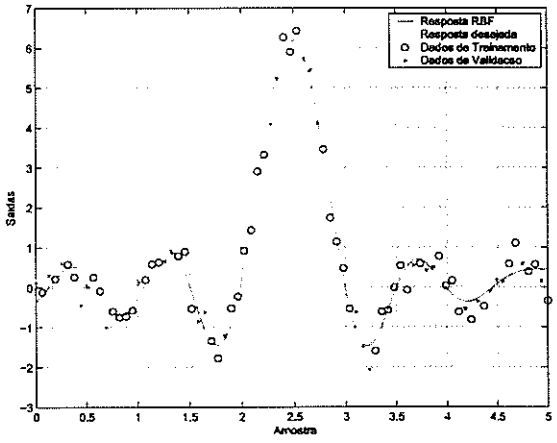

Figura 6.26: Solução de MOBJ-RBFr.

Pela análise da qualidade de cada solução apresentada na Tabela 6.7, percebe-se que a qualidade das soluções pelos métodos MOBJ são ligeiramente superiores aos métodos de regularização, em especial, a solução MOBJ-RBF alcançou um valor de erro muito notável em comparação com os demais.

Tabela 6.7: Qualidade de soluções para problema de regressão Sinc.

Quantidade	RR-GCV	RR-BIC	MOBJ-RBF	RR-MOBJ-RBF	MOBJ-RBFr
MSE Trein.	0,0614	0,0690	0,0985	0,0949	0,0918
MSE Valid.	0,1221	0,1193	0,0787	0,0892	0,1153
MSE Teste	0,0340	0,0348	0,0166	0,0250	0,0339
Neurônios	20	20	20	20	20
λ	0,00014	0	-	$1{,}0.10^{-8}$	-
Raio	0,54	1,28	0,38	1,8	0,4
$\|H\|$	11,24	17,01	11,42	12,54	24,10
$\|W\|$	41,26	12482	13,61	25,9	73,37

Para a determinação automática de topologia, comparou-se as soluções obtidas utilizando cada método de *subset selection*. As respostas estão apresentadas na Figura 6.27 a 6.30.

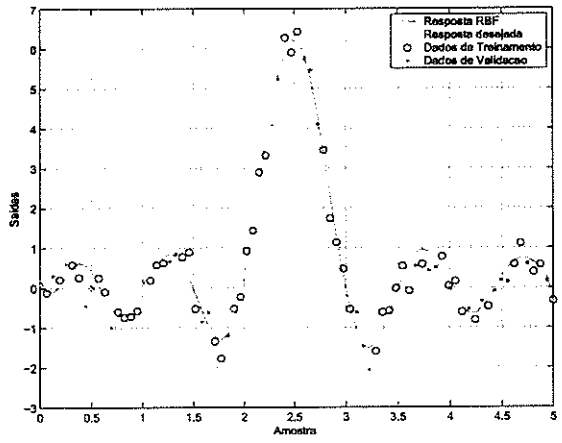

Figura 6.27: Solução de NewRB-GCV.

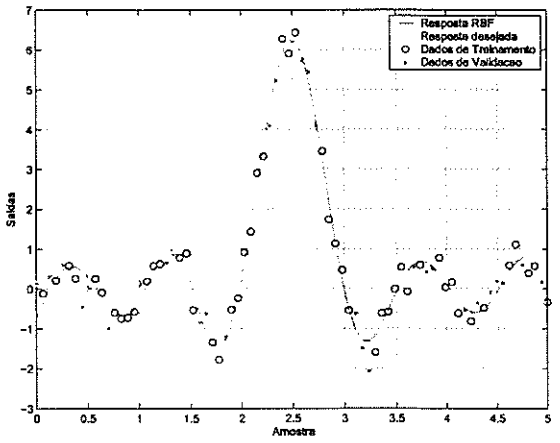

Figura 6.28: Solução de FS-GCV.

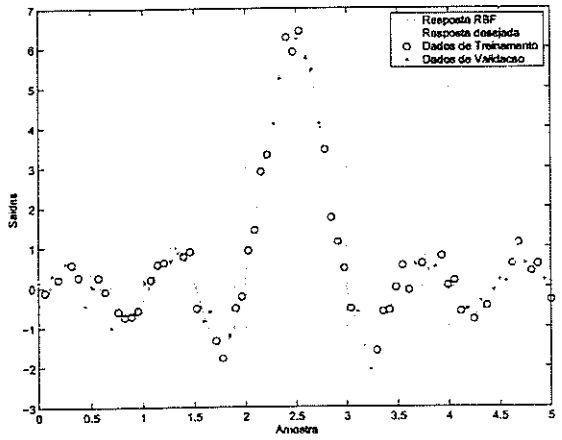

Figura 6.29: Solução de FS-BIC.

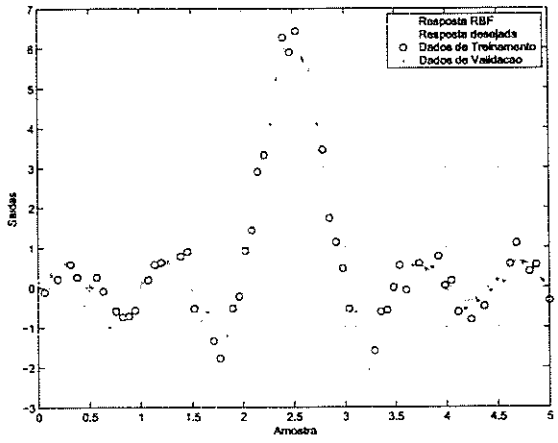

Figura 6.30: Solução de SS-MOBJ-RBF.

Tabela 6.8: Qualidade de soluções para problema de regressão Sinc utilizando *subset selection*.

Quantidade	NewRB-CV	FS-GCV	FS-BIC	SS-MOBJ-RBF
MSE Trein.	0,0741	0,0600	0,0573	0,1054
MSE Valid.	0,1173	0,1405	0,1305	0,0785
MSE Teste	0,0347	0,0444	0.0371	0,0303
Neurônios	20	13	20	11
λ	-	0,00578	0,00169	-
Raio	1,8	0,44	0,49	0,48
$\|\|H\|\|$	-	11,01	13,18	9,50
$\|\|W\|\|$	-	19,86	36,11	42,02

Dentre as alternativas de *subset selection* abordadas neste exemplo, o resultado obtido pelo método multi-objetivo atingiu o valor mais baixo de erro para o conjunto de dados de teste.

Pela Tabela 6.9 é possível verificar mais uma vez a diminuição no tempo de treinamento para as variações MOBJ para a aceleração de busca de soluções.

Tabela 6.9: Tempo gasto para treinamento MOBJ para problema Sinc.

Algoritmo	Tempo
MOBJ-RBF	4800 segundos
MOBJ-RBFr	993 segundos
RR-MOBJ-RBF	19 segundos

6.5 Exemplo 4: Predição da série caótica de Mackey-Glass

Problemas de predição são caracterizados por estimar o valor de alguma variável em um instante futuro, a partir de amostras de variáveis em instantes anteriores.

O problema da predição da série caótica de Mackey-Glass, tratado em (Jang, Sun, and Mizutani 1997), é determinado pela predição do valor da série em instantes de tempo a frente ($x(t+6)$) em função de um vetor de dados formado por dados gerados a partir de uma solução obtida pelo método de Runge-Kutta de quarta ordem nos instantes descritos pela Equação 6.4. Os dados foram obtidos por meio do toolbox de fuzzy disponível para Matlab já formatados da maneira apresentada na Equação 6.4.

$$\mathrm{X} = \begin{pmatrix} x(t-18) & x(t-12) & x(t-6) & x(t) \end{pmatrix} \tag{6.4}$$

O conjunto de dados utilizado foi dividido em 350 amostras para treinamento, 150 amostras para validação e 500 amostras para teste, conforme ilustra a Figura 6.31.

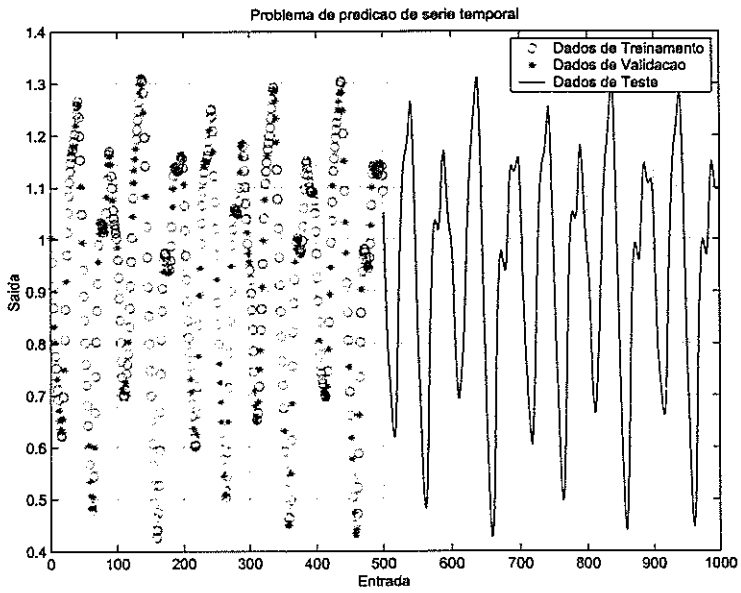

Figura 6.31: Conjunto de dados para problema de predição da série caótica de Mackey-Glass.

A topologia escolhida para a comparação de métodos foi de 15 neurônios na camada escondida. Os resultados obtidos estão ilustrados nas Figuras 6.32 a 6.33.

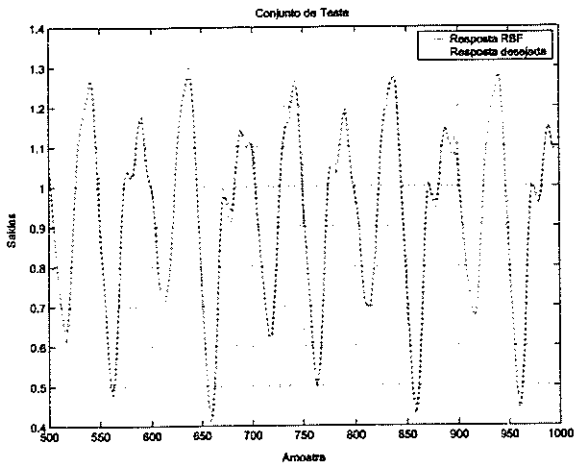

Figura 6.32: Solução de RR-GCV.

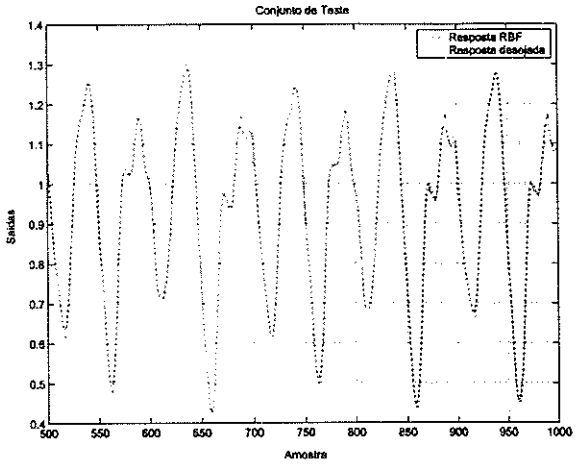

Figura 6.33: Solução de RR-BIC.

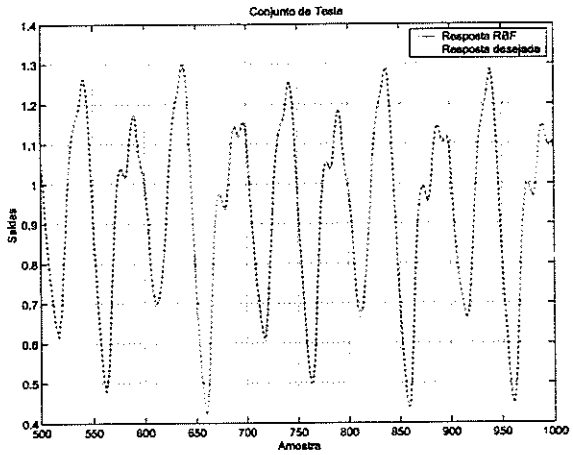

Figura 6.34: Solução de MOBJ-RBF.

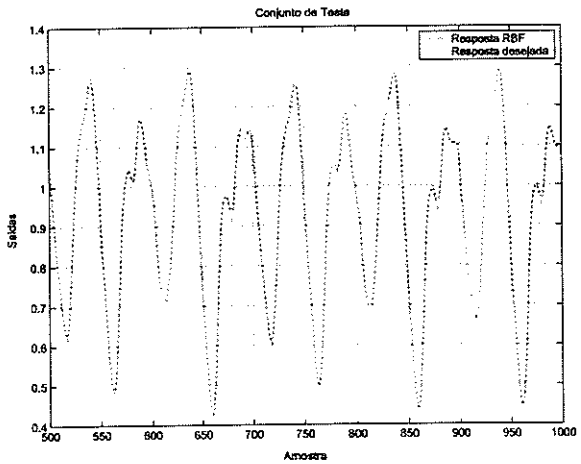

Figura 6.35: Solução de RR-MOBJ-RBF.

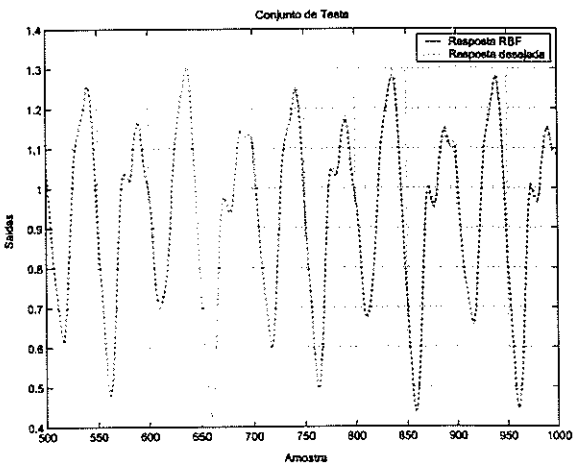

Figura 6.36: Solução de MOBJ-RBFr.

Pela análise do erro para cada solução apresentada na Tabela 6.10, percebe-se que a qualidade das soluções MOBJ-RBF é ligeiramente superior aos métodos de regularização.

Tabela 6.10: Qualidade de soluções para problema de predição de série caótica.

Quantidade	RR-GCV	RR-BIC	MOBJ-RBF	RR-MOBJ-RBF	MOBJ-RBFr
MSE Trein.	0,000436	0,000582	0,000117	0,000212	0,000240
MSE Valid.	0,000448	0,000644	0,000125	0,000268	0,000224
MSE Teste	0,000411	0,000565	0,000113	0,000218	0,000230
Neurônios	15	15	15	15	15
λ	0,0	0,0	-	$1{,}0.10^{-8}$	-
Raio	0,93	1,26	0,46	0,55	0,49
$\|H\|$	50,53	57,80	46,39	50,82	60,78
$\|W\|$	45,60	104,3	10,97	45,06	11,29

Para a determinação automática de topologia, comparou-se as soluções obtidas utilizando cada método de *subset selection*. As respostas estão apresentadas nas Figuras 6.37 a 6.40.

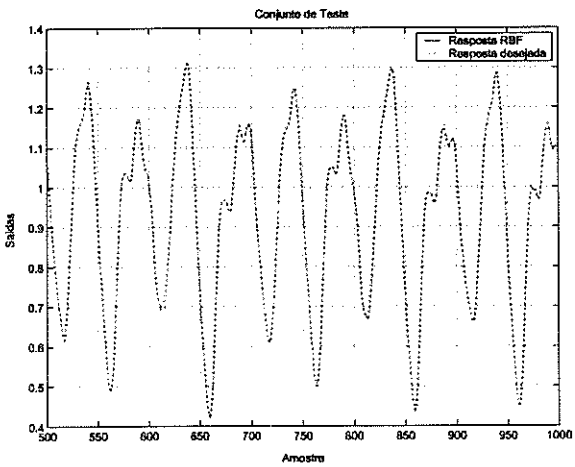

Figura 6.37: Solução de NewRB-GCV.

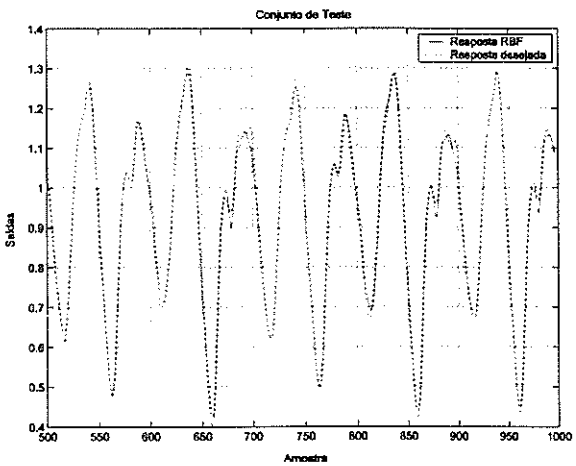

Figura 6.38: Solução de FS-GCV.

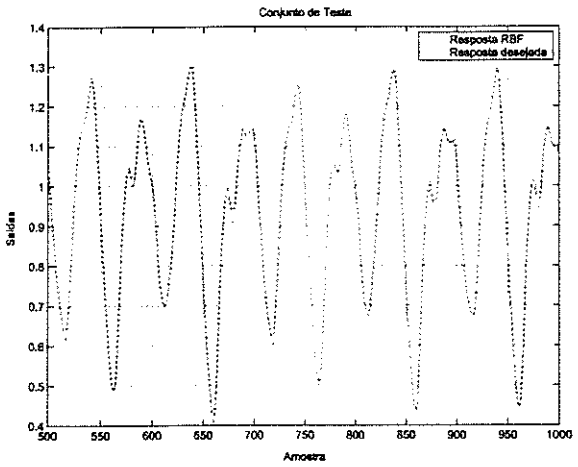

Figura 6.39: Solução de FS-BIC.

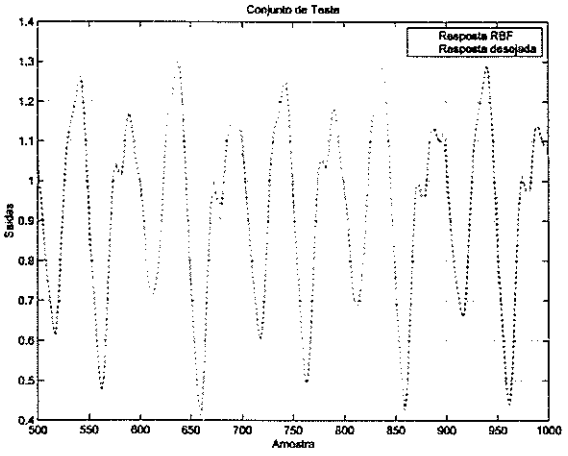

Figura 6.40: Solução de SS-MOBJ-RBF.

Tabela 6.11: Qualidade de soluções para problema de predição de série caótica utilizando *subset selection*.

Quantidade	NewRB-CV	FS-GCV	FS-BIC	SS-MOBJ-RBF
MSE Trein.	0,000120	0,000331	0,000148	0,000232
MSE Valid.	0,000136	0,000414	0,000202	0,000261
MSE Teste	0,000121	0,000340	0,000160	0,000231
Neurônios	15	15	15	15
λ	-	0.000331	0.000635	-
Raio	0,5	0,75	0,80	0,66
$\|H\|$	-	46,99	49,93	50,76
$\|W\|$	-	10,57	33,95	36,84

A solução obtida pelo método MOBJ é relativamente próxima das soluções encontradas pelos outros métodos. As topologias foram semelhantes pois todos os algoritmos tiveram seu espaço de busca limitado em 15 neurônios na camada escondida. A eficiência computacional das variações propostas no trabalho está ilustrada na Tabela 6.12, onde a variação RR-MOBJ-RBF se apresenta como o algoritmo que gastou menos tempo para a geração de soluções eficientes.

Tabela 6.12: Tempo gasto para treinamento MOBJ para problema de predição de série caótica.

Algoritmo	Tempo
MOBJ-RBF	5720 segundos
MOBJ-RBFr	2281 segundos
RR-MOBJ-RBF	85 segundos

6.6 Conclusões do capítulo

Conforme pode-se observar nos resultados obtidos, o método multi-objetivo proposto e suas variações foram capazes de encontrar soluções de alta capacidade de generalização, representada pelo erro para o conjunto de dados de teste.

Para o exemplo de regressão da função seno, o método MOBJ-RBF e suas variações para os métodos de topologia fixa obtiveram os melhores resultados. Já para a técnica de *subset selection*, a solução obtida pelo método SS-MOBJ-RBF se apresenta muito próxima da solução do método FS-BIC.

As soluções obtidas para o exemplo da função d(x) apresentaram um comportamento sobre-ajustado, isto porque a topologia possui, intencionalmente, um número de neurônios superior ao necessário, permitindo analisar a capacidade de se amenizar os efeitos do sobre-ajuste a partir de técnicas multi-objetivo em comparação com algoritmos de regularização. As soluções obtidas para os métodos MOBJ-RBF de topologia fixa e por meio da técnica de *subset selection* atingiram resultados ligeiramente superiores aos outros métodos.

No exemplo de regressão da função sinc, o método MOBJ-RBF alcançou um valor de erro para o conjunto de teste bem inferior aos demais seguido do método RR-MOBJ-RBF. Para a metodologia de *subset selection* a solução do método SS-MOBJ-RBF é ligeiramente superior ao alcançado pelo NewRB incrementado com validação cruzada.

A metodologia para avaliação de modelos de predição para a série caótica de Mackey-Glass foi seguida segundo (Jang, Sun, and Mizutani 1997). Todas as soluções apresentaram um comportamento com grande capacidade de generalização vizualizado pelo comportamento das predições para o conjunto de teste. As diferenças foram mínimas, onde as soluções MOBJ-RBF e variações obtiveram um erro de aproximação relativamente próximos.

Para todos os exemplos foram avaliados o tempo gasto para a geração de soluções eficientes e o processo de decisão. Conforme era esperado, as variações do método original se mostraram muito eficientes. O método MOBJ-RBFr apresentou um ganho considerável de tempo ao se utilizar somente uma aproximação de conjunto Pareto-ótimo. Os ganhos foram bem mais significativos para o método RR-MOBJ-RBF, onde a técnica de *ridge regression* permitiu encontrar soluções para a camada de saída analiticamente diminuindo muito o tempo necessário para treinamento de redes RBF.

De modo geral, a qualidade do ajuste para os métodos propostos no trabalho se mostraram compatíveis com as soluções geradas pelos outros métodos e até na maioria deles ligeiramente superiores.

CAPÍTULO 7

Conclusões e Propostas de Continuidade

São apresentadas neste capítulo, discussões a respeito do método de treinamento multi-objetivo para redes RBF proposto e suas variações. São abordadas as principais conclusões relacionadas com a busca de soluções de alta capacidade de generalização em redes RBF e propostas para trabalhos futuros.

7.1 Conclusões

O trabalho presente teve como objetivo principal, estender os conceitos de treinamento multi-objetivo desenvolvido para redes MLP (Teixeira, Braga, Takahashi, and Saldanha 2000) (Costa, Braga, de Menezes, Parma, and Teixeira 2002) para o ajuste de parâmetros livres de redes RBF. As soluções multi-objetivo para redes MLP são obtidas por meio da minimização do erro sujeita à restrições de nível de complexidade, representadas pela norma da matriz de pesos entre conexões de neurônios. A partir de um conjunto restrito de soluções eficientes, seleciona-se o modelo de menor erro para um conjunto de validação.

Devido à semelhança, entre redes MLP e RBF, de suas funções de ativação da camada de saída, a norma da matriz de pesos foi adotada como medida de complexidade para esta camada. A camada intermediária das redes RBF são formadas por funções radiais, diferentemente das funções sigmoidais das redes MLP. Diante desta situação, foi sugerida uma medida de complexidade para a camada escondida representada pela norma da matriz de interpolação ($\|\mathbf{H}\|$).

A norma da matriz de interpolação foi selecionada como forma de medir a complexidade desta camada, por ser uma grandeza que é calculada em função das distâncias médias entre os padrões de entrada e centros de funções radiais, pelo valor de raio e número de neurônios. Esta medida se mostrou muito interessante para o caso de redes RBF por ser calculada de modo relativamente simples, apresentando um comportamento contínuo em relação aos parâmetros caracterizadores da mesma.

Foram apresentados os comportamentos gerais de soluções de redes RBF segundo o número de centros e valores de raio. Existe um nível de complexidade ótimo determinado por um valor de norma da matriz de interpolação que garante um erro mínimo. Soluções de nível de complexidade inferior representam soluções de respostas locais e sub-ajustadas com erros altos, enquanto que soluções de norma muito superiores também possuem valor de erro alto devido ao mal-condicionamento da matriz de interpolação.

Valores de norma de **H** muito altos representam um condicionamento ruim desta matriz, ocasionado pelo excesso de cobertura das funções radiais do espaço dos padrões de entrada. Situações como estas caracterizam um mapeamento não-linear ineficiente da camada escondida.

Não foi possível estabelecer uma relação direta entre as duas medidas de complexidade. Desta forma, os algoritmos de treinamento elaborados tratam cada grandeza de maneira independente. O problema de treinamento de redes RBF fica então representado por um problema multi-objetivo constituído de três funcionais: o erro de treinamento (e_T), a norma da matriz de pesos ($\|\mathbf{W}\|$) e a norma da matriz de interpolação ($\|\mathbf{H}\|$) conforme a Equação 7.1.

$$\psi^* = arg_\psi \ min \begin{cases} f_1(\psi) = e_T(\psi) \\ f_2(\psi) = \|\mathbf{W}\| \\ f_3(\psi) = \|\mathbf{H}\| \end{cases} \tag{7.1}$$

O treinamento multi-objetivo pode então ser resolvido através do método ε-restrito (Chankong and Haimes 1983), onde um problema multi-objetivo é decomposto em problemas mono-objetivo restritos.

O limite do nível de complexidade é imposto pelas restrições nas normas das matrizes **W** e **H**, onde para cada nível de complexidade busca-se encontrar a solução de mínimo erro. A variação paramétrica das medidas de complexidade permite obter um conjunto reduzido de soluções candidatas à solução final que se aproximam do conjunto Pareto-ótimo. A solução de menor erro para um conjunto validação é selecionada como a resposta que supostamente seja a de maior capacidade de generalização.

A grande vantagem deste tipo de abordagem é diminuir a influência de arquiteturas mal-projetadas, contornando o problema do *overfitting* por meio

da limitação de complexidade através da magnitude de seus parâmetros livres.

É necessária a geração de várias soluções restritas para a aplicação do método MOBJ-RBF, o que resulta em um tempo maior de processamento e alocação de memória. Para isto, foram desenvolvidas variações de métodos que geram de maneira mais eficiente computacionalmente as soluções candidatas à solução final.

Uma primeira variação do método proposto, denominada de MOBJ-RBFr (Carvalho, Costa, and Braga 2004), descarta o controle da norma de **H**, considerando o valor de raio como variável de otimização para o funcional erro. O algoritmo MOBJ-RBFr é uma alternativa para o treinamento multi-objetivo com geração reduzida de soluções. Os resultados obtidos com este algoritmo demonstraram que é possível encontrar soluções de alta generalização ao se utilizar esta variação. O tempo gasto nesta variação é bem inferior quando comparado ao método original e possui a vantagem de determinar automaticamente o valor de raio das funções de base radial.

No método original MOBJ-RBF faz-se uso do método elipsoidal (Shor 1977) para a determinação de soluções de norma de **W** restritas. A segunda variação, RR-MOBJ-RBF utiliza regularização (*Ridge Regression*) para a geração de soluções eficientes para a camada de saída. A utilização da regularização acelera a geração de soluções eficientes uma vez que existe uma solução analítica para os pesos da camada de saída que minimiza o erro de treinamento em função do termo de regularização. Os resultados obtidos com esta metodologia exigiram um tempo de processamento muito inferior aos demais métodos MOBJ-RBF com qualidade semelhante. O algoritmo RR-MOBJ-RBF é capaz de encontrar soluções de mesma qualidade que o método original. As diferenças encontradas entre as soluções foi devido à questão da parametrização dos níveis de complexidade durante o treinamento. Esta variação se apresenta, do ponto de vista do autor, como uma alternativa muito viável para o treinamento de redes RBF devido às características citadas anteriormente.

As abordagens apresentadas são aplicadas em uma topologia RBF pré-definida. Pode-se, utilizando um técnica de *Subset Selection*, realizar uma busca do nível de complexidade ótimo levando em consideração o número de funções radiais e os seus centros. Esta metodologia é apresentada na variação SS-MOBJ-RBF, onde a norma da matriz de interpolação é utilizada como restrição de complexidade para a combinação de número de centros e raios.

Parte-se de uma rede mínima e aumentando a complexidade do modelo estabelece-se uma combinação de número de centros e valor de raio a ser submetida ao ajuste dos pesos de saída. Este tipo de abordagem permite explorar a complexidade de redes RBF de dois modos distintos: alterando o número de parâmetros livres e também limitando a magnitude de seus parâmetros.

Isto aumenta a capacidade de encontrar soluções de alta capacidade de generalização. Apesar de exigir um tempo de processamento maior, o algoritmo diminui a influência do projetista no resultado final.

Para todos os algoritmos propostos neste trabalho, utilizou-se um decisor baseado em conjunto validação. Desta forma, a solução pertencente ao conjunto de soluções eficientes que possuir menor valor de erro para um conjunto validação é selecionada como a resposta que equilibra os efeitos de polarização e variância.

Conforme os testes realizados, o método multi-objetivo para treinamento de redes RBF proposto e suas variações se mostraram uma alternativa eficiente para a busca de soluções de alta capacidade de generalização. Além de dar prosseguimento aos trabalhos de treinamento multi-objetivo para redes neurais artificiais, o trabalho presente apresenta discussões sobre influências de parâmetros livres em redes RBF no resultado final, propondo utilizar informações de complexidade de ambas as camadas para o seu treinamento enquanto que na maioria dos outros algoritmos utiliza-se somente a informação de complexidade da camada de saída. Apesar da formulação inicial de treinamento MOBJ para redes RBF exigir um tempo de processamento maior, foi possível desenvolver técnicas para diminuir esta necessidade.

O método RR-MOBJ-RBF, em destaque, possui uma exigência de tempo de processamento comparável aos demais algoritmos e ainda faz uso de informações de complexidade ainda não utilizadas para o treinamento de redes RBF encontrando soluções de alta capacidade de generalização justificando sua utilização como alternativa para os algoritmos mais avançados de treinamento para redes RBF.

7.2 Propostas de Continuidade

Foram propostos neste trabalho alguns métodos de treinamento de redes RBF que utilizam o conceito de otimização multi-objetivo para a determinação de soluções de alta capacidade de generalização. Novas discussões foram geradas com respeito ao comportamento e determinação de soluções para redes RBF, dentre as quais podem ser identificados diversos tópicos de pesquisa. Alguns tópicos estão relacionados a seguir:

- Assumiu-se no trabalho que o método de K-médias determina a melhor distribuição de centros das funções de base, não inserindo as posições dos centros no processo de otimização multi-objetivo. Trabalhos futuros podem explorar a relação entre as posições de centros e a qualidade das soluções para as redes RBF.

- Sabe-se que o conjunto de soluções geradas para todo o conjunto de restrições de norma da matriz de interpolação e matriz de pesos para RBF possui soluções dominadas por outras, de modo que o conjunto Pareto-ótimo é apenas um subconjunto das soluções geradas pelos métodos apresentados neste trabalho. Torna-se necessário determinar metodologias para redução do número de soluções a ser geradas de forma a determinar somente pontos no espaço tridimensional determinado por $\|H\|$, $\|W\|$ e e_T pertencentes ao conjunto Pareto-ótimo.

- Toda análise de comportamento de soluções e medida de complexidade foram abordadas para funções de base radial gaussianas. Os mesmos procedimentos podem ser explorados para outros tipos de função de base radial, como por exemplo as multiquádricas.

- Não foi possível estabelecer neste trabalho uma relação analítica entre as duas medidas de complexidade. Caso isto seja possível de ser determinado, o problema de natureza tri-objetivo poderá ser tratado de maneira bi-objetivo a partir de outra proposta de medida de complexidade que represente todos os parâmetros livres de redes RBF.

- Pode-se explorar novos critérios para representar o decisor de solução final. Pelo fato de a rede RBF apresentar uma parte de natureza linear, os critérios de seleção de modelos GCV e BIC podem ser uma boa aproximação para a solução de maior capacidade de generalização, dispensando a necessidade de um conjunto de dados de validação, que é um grande ganho para problemas com um número baixo de amostras para projeto de redes neurais.

APÊNDICE

A

Propriedades gerais de álgebra linear

Neste apêndice serão apresentadas algumas características de sistemas lineares relevantes ao trabalho apresentado.

A.1 Matrizes

A seguir estão apresentadas alguns tipos de matrizes mais utilizadas nas formulações de treinamento de redes neurais.

- Matrizes diagonais: são matrizes cujos os elementos fora da diagonal principal são nulos $m_{ij} = 0$, $i \neq j$.

$$\mathbf{M} = \begin{pmatrix} m_{11} & 0 & \dots & 0 \\ 0 & m_{22} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & m_{pq} \end{pmatrix} \tag{A.1}$$

- Matrizes identidade: são matrizes cujos os elementos fora da diagonal principal são nulos $m_{ij} = 0$, $i \neq j$ e os elementos da diagonal principal são iguais a unidade $m_{ij} = 1$, $i = j$.

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} \tag{A.2}$$

- Matrizes simétricas: são matrizes que são idênticas a sua transposta . Qualquer matriz diagonal é uma matriz simétrica.

$$\mathrm{M} = \mathrm{M}^{\mathrm{T}} \tag{A.3}$$

- Matrizes ortogonais: são matrizes cuja sua inversa é idêntica a sua transposta.

$$\mathrm{M}^{-1} = \mathrm{M}^{\mathrm{T}} \tag{A.4}$$

Se uma matriz é ortogonal então:

$$\mathrm{M}^{\mathrm{T}}\mathrm{M} = \mathrm{M}^{\mathrm{T}}\mathrm{M} = \mathrm{I} \tag{A.5}$$

- Traço: O traço (*trace*) de uma matriz quadrada é a soma dos elementos de sua diagonal.

$$trace(\mathrm{M}) = \sum m_{ij} \text{ , } para\ i = j \tag{A.6}$$

- Posto: O posto (*rank*) é dado pelo número de linhas ou colunas linearmente independentes de uma matriz.

- Norma: A norma euclidiana de uma matriz é dado pela raiz quadrada do somatório quadráticos de seus elementos.

$$\|M\| = \sqrt{\sum m_{ij}^2} \tag{A.7}$$

Existem outras formulações para a norma de uma matriz que não foram utlizadas no desenvolvimento deste trabalho.

- Condicionamento: O condicionamento (cond) de uma matriz é dado pelo produto entre a sua norma e a norma de sua inversa.

$$cond(\mathrm{M}) = \|M\|\|M^{-1}\| \tag{A.8}$$

A.2 Sistemas de equações lineares

Um sistema de equações lineares ou simplesmente um sistema linear é caracterizado por um conjunto de relações de igualdade onde cada variável independente é ponderado por um coeficiente.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1q} = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2q} = b_2 \\ \vdots \\ a_{p1}x_1 + a_{p2}x_2 + \ldots + a_{pq} = b_p \end{cases} \tag{A.9}$$

Um sistema linear pode ser descrito matricialmente conforme a Equação A.10.

$$\mathbf{Ax} = \mathbf{b} \tag{A.10}$$

Onde a solução para o vetor de variáveis independentes x pode ser calculado através da inversa da matriz de coeficientes **A** se esta for não-singular.

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{A.11}$$

Um matriz **A** é não singular se, e somente se:

- **A** for uma matriz quadrada ($p = q$);
- Existe $\mathbf{A}^{-1}$;
- Determinante de **A** for não-nulo;
- Possuir posto completo ($rank(\mathbf{A}) = p$)

A.3 Condição de um sistema linear

Um problema é dito mal-condicionado quando pequenas mudanças nos elementos de uma matriz produzem grandes modificações no resultado de um sistema linear.

Existem dois casos onde se pode analisar a influência do condicionamento da matriz de coeficientes para a solução de um sistema linear (Ciarlet 1995).

No primeiro caso, uma comparação a solução exata obtida para um sistema linear inicial e a solução para um sistema linear onde houve uma pertubação na matriz de valores desejados **b**.

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ \mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) &= \mathbf{b} + \Delta\mathbf{b} \end{aligned} \tag{A.12}$$

Neste caso, a diferença relativa de soluções representada por $\|\Delta\mathbf{x}\|/\|\mathbf{x}\|$ é limitado pelo termo de erro relativo $\|\Delta\mathbf{b}\|/\|\mathbf{b}\|$ ponderado pelo condicionamento da matriz de coeficientes (Equação A.13).

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq cond(\mathbf{A})\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} \tag{A.13}$$

No segundo caso, a perturbação ocorre na matriz de coeficientes, onde se compara as soluções entre dois sistemas lineares de erros nulos.

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{b} \\ (\mathbf{A}+\Delta\mathbf{A})(\mathbf{x}+\Delta\mathbf{x}) &= \mathbf{b} \end{aligned} \tag{A.14}$$

Neste caso, a diferença relativa de soluções representada por $\|\Delta\mathbf{x}\|/\|\mathbf{x}\|$ é limitado pelo termo perturbação relativa $\|\Delta\mathbf{A}\|/\|\mathbf{A}\|$ ponderado pelo condicionamento da matriz de coeficientes (Equação A.15).

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}+\Delta\mathbf{x}\|} \leq cond(\mathbf{A})\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|} \tag{A.15}$$

Em ambos os casos, pode-se verificar que o erro relativo no resultado de um sistema linear é limitado pela perturbação relativa nos dados do sistema representados pela matriz de coeficientes e saídas desejadas, multiplicado pelo condicionamento da matriz de coeficientes. Em outras palavras, o condicionamento de **A** pode medir a sensibilidade da solução de um sistema linear.

Um sistema linear é dito bem-condicionado se o valor de cond(**A**) for próximo da unidade. Neste casos as soluções possuem baixa variabilidade sendo pouco sensíveis à perturbações nos dados do sistema. Analogamente, sistemas com alto valor de cond(**A**) são sistemas mal-condicionados, cujas soluções são muito sensíveis a variações dos coeficientes (ruídos) e saídas desejadas.

Referências

Barron, A. R. (1993). Universal approximations bounds for superpositions of a sigmoid function. *IEEE Transactions on Information Theory*, 39:930–945.

Bartlett, P. L. (1997). For valid generalization the size of the weights is more important than the size of the network. *Advances in Neural Information Processing Systems 9*, 134.

Bengio, Y. (1996). Neural networks for speech and sequence recognition.

Benoudijit, N., C. Archambeau, A. Lendasse, J. Lee, and M. Verleysen (2002). Width optimization of the gaussian kernels in radial basis functions networks. In *Proceedings of European Symposium on Artificial Networks*, pp. 425–432.

Berthold, M. R. and J. Diamond (1995). Boosting the performance of rbf networks with dynamic decay adjustment. In *Advances in Neural Information Processing System 7*, pp. 521–528. MIT Press.

Bjork, A. (1967). Solving linear least squares problems by Gram-Schmidt orthogonalization. *Nordisk Tidskr 7*, 1–21.

Borş, A. G. and I. Pitas (1994, September). Robust estimation for radial basis functions. In *Proc. NNSP'94, IEEE Workshop on Neural Networks for Signal Processing*, Piscataway, NJ, pp. 105–114. IEEE: IEEE Service Center.

Bradley, P. S. and U. M. Fayyad (1998). Refining initial points for k-means clustering. In *International Conference on Machine Learning*.

Broomhead, D. S. and D. Lowe (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems 2*, 321–355.

Carvalho, D. H. D., M. A. Costa, and A. P. Braga (2004, Setembro). Ajuste da generalização em redes neurais de base radial: uma abordagem multi-objetivo para a estimação de parâmetros. *VII Simpósio Brasileiro de Redes*

Neurais.

Carvalho, D. H. D., T. H. Medeiros, and R. Fortuna (2004). Sistema de detecção de falhas na formação da casca em processos de lingotamento contínuo utilizando redes neurais artificiais. *VII Seminário de Automação de Processos (ABM).*

Chankong, V. and Y. Y. Haimes (1983). *Multiobjetive Decision Making : Theory and Methodology.* North-Holland.

Chen, S., C. F. N. Cowan, and P. M. Grant (1991). Orthogonal least squares learning algorithm for radial basis functions networks. *IEEE Trans. Neural Networks.*

Chen, S., Y. Wu, and L. Luk (1999). Combined genetic algorithm optimization and regularizes orthogonal least squares learning for radial basis function networks. *IEEE Transactions on Neural Networks.*

Ciarlet, P. G. (1995). *Introduction to numerical linear algebra and optimisation*, Volume 2. Cambridge Texts in Applied Mathematics.

Cohen, S. and N. Intrator (2000). Global optimization of RBF networks.

Costa, M. A. (2001). *Controle por Modos Deslizantes da Generalização em Aprendizado de Redes Neurais Artificiais.* Tese de Doutorado, Universidade Federal de Minas Gerais, MG, Brasil.

Costa, M. A., A. P. Braga, and A. Aguirre (2000, Novembro). Um estudo do comportamento temporal do Índice de preços do boi gordo nas Últimas décadas e sua modelagem através de redes neurais artificiais. *V Simpósio Brasileiro de Redes Neurais.*

Costa, M. A., A. P. Braga, B. R. de Menezes, G. G. Parma, and R. A. Teixeira (2002, november). Control of generalization with a bi-objective sliding mode control algorithm. In *XVI Brazilian Symposium on Neural Networks.*

Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers EC-14*, 326–334.

Cun, Y. L., J. S. Denker, and S. A. Solla (1990). Optimal brain damage. *Advances in Neural Information Processing Systems 2*, 598–605.

D. Plaut, S. and G. Hinton (1986). Experiments on learning by backpropagation. Technical report, Carnegie Mellon University.

Demuth, H. and M. Beale (2001). *Neural Networks Toolbox Users's Guide.* Mathworks.

Duckstein, L. (1984). *Multiobjetive optimization in structural design: The model choice problem.* John Wiley & Sons.

Efron, B. and R. J. Tibshirani (1993). An introduction to the bootstrap. *Monographs on Statistics and Applied Probability 57*.

Fessant, F., P. Aknin, L. Oukhellou, and M. Midenet (2001). Comparison of supervised self-organizing maps using Euclidian or Mahalanobis distance in classification context. In *Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence. 6th International Work-Conference on Artificial and Natural Neural Networks, IWANN 2001. Proceedings, Part I (Lecture Notes in Computer Science Vol. 2084). Springer-Verlag, Berlin, Germany*, pp. 637–44.

Fletcher, R. and C. M. Reeves (1964). Function minimization by conjugate gradients. *Computer Journal 7*, 149–154.

Friedman, J. H. (1994). An overview of predictive learning and function approximation. *Proc. NATO/ASI Worshop*, 1–61.

Geman, S., E. Bienenstock, and R. Doursat (1992). Neural networks and the bias/variance dilemma. *Neural Computation* (4(1)), 1–58.

Girosi, F., M. Jones, and T. Poggio (1978). Regularization theory and neural networks architetures. *Neural Computation*, 219–269.

Girosi, F., T. Poggio, and B. Caprile (1991). Extensions of a theory of networks for approximation and learning. In R. P. Lippmann, J. E. Moody, and D. S. Touretzky (Eds.), *Advances in Neural Information Processing Systems*, Volume 3, pp. 750–756. Morgan Kaufmann Publishers, Inc.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.

Golub, G. H., M. Heath, and G. Wahba (1979). Generalised cross-validation as a method for choosing a good ridge parameter. *Technometrics 21*, 215–223.

Golub, G. H. and C. Reinsch (1970). Singular value decomposition and least squares solutions. *Numerishe Mathematik 14*, 403–420.

Hamerly, G. and C. Elkan (2003). Learning the k in k-means. *Proceedings of the seventeenth annual conference on neural information processing systems*.

Hartman, E., J. D. Keeler, and J. M. Kowalski (1990). Layered neural networks with Gaussian hidden units as universal approximations. *Neural Computation 2*(2), 210–215.

Hassibi, B. and D. G. Stork (1993). Second order derivatives for network pruning: Optimal brain surgeon. *Advances in Neural Information Processing Systems*, 164–171.

Hassoun, M. H. (1995). *Fundamentals of artificial neural networks*. MIT Press Cambridge/Boston/London.

Haykin, S. (1994). *Neural Networks : A Comprehensive Foundation*. NY: Macmillan College Publishing Company.

Hertz, J., A. Krough, and R. G. Palmer (1991). *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley.

Hinton, G. E. and S. J. Nowlan (1987). How learning can guide evolution. *Complex Systems*, 495–502.

Hoerl, A. E. and K. R. W (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics 12*, 55–67.

Itkis, U. (1976). *Control Sustems of Variable Structure*. Keter Publishing House Jerusalem LTD.

Jang, J. S. R., C. T. Sun, and E. Mizutani (1997). *Neuro-Fuzzy and Soft Computing*. Upper Saddle River, NJ, USA: Prentice Hall.

Kohonen, T. (1982). Self-organized formation of topological feature maps. *Biological Cybernetics 43*, 59–69.

Lawrence, S., C. L. Giles, and A. Tsoi (1996). What size neural network gives optimal generalization? Technical report, Institute for Advanced Computer Studies, University of Maryland.

Leon, S. J. (1994). *Linear algebra wiht aplications*. Prentice Hall.

Mackay, D. J. C. (1992). Bayesian interpolation. *Neural Computation 4*, 415–447.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman (Eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Volume 1, Berkeley, CA, pp. 281–297. University of California Press.

McClelland, J. L. and D. E. Rumelhart (1988). *Explorations in Parallel Distributed Processing*. Cambridge: MIT Press.

Medeiros, T. H. (2004). Decisor de mínima correlação para treinamento multiobjetivo e redes neurais mlp. Trabalho de Pós-Graduação, Centro de Pesquisa e Desenvolvimento em Engenharia Elétrica, UFMG, Brasil.

Moody, J. and C. Darken (1988). Learning with localised receptive fields. Research report, Yale University Department of Computer Science.

Moody, J. and C. Darken (1989a). Fast learning in networks of locally-tuned processing units. Technical Report YALEU/DCS/RR-654, Dept. of Computer Science, Yale University, New Haven, CT.

Moody, J. and C. Darken (1989b). Learning with localized receptive fields. In *Proceedings of the 1988 Connectionist Models Summer School*. Morgan Kaufmann Publishers, Inc.

Moody, J. E. (1992). The effective number of parameters: An analysis of generalisation and regularisation in nonlinear learning systems. *Neural Information Processing Systems 4*, 847–854.

Niyogi, P. and F. Girosi (1996). On the relationship between generalization error, hypothesis complexity and sample complexity for radial basis functions. *Neural Computation 8*, 819–842.

Orr, M. J. L. (1996). Introdution to radial basis function networks. Technical report, University of Edinburgh.

Orr, M. J. L. (1997). Matlab routines of subset selection and ridge regression in linear neural networks. Technical report, University of Edinburgh.

Orr, M. J. L. (1999). *Matlab Functions for Radial Function Networks*. Institute for Adaptative and Neural Computation, Edinburgh University, Scotland, UK.

ou Wang, Z. and T. Zhu (2000). An efficient learning algorithm for improving generalization performance of radial basis function neural networks. *Neural Networks 13*(4-5), 545–553.

Pareto, V. (1896). *Cours D'Economic Politique*.

Parma, G. G., B. R. Menezes, and A. P. Braga (1998). Sliding mode algorithm for training multi-layer neural networks. *IEE Electronics Letters*, pp. 97–98.

Reily, D. L., L. N. Cooper, and C. Elbaum (1982). A neural model for category learning. *Biol. Cybernet* (45), 35–41.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning internal representation by error propagation. In D. E. Rumelhart and J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 1, pp. 318–362. Cambridge, MA: MIT Press.

Scholkopf, B., K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik (1997). Comparing support vector machines with Gaussian kernels to radial basis functions classifiers. *IEEE Transactions on Signal Processing 45*.

Shor, N. Z. (1977). Cut-off method with space extension in convex programming problems. *Cybernetics*, 94–96.

Stone, M. (1978). Cross-validation: A review. *Mathematische Operationsforschung Statistischen*, 127–140.

Takahashi, R. H. C., P. L. D. Peres, and P. A. V. Ferreira (1997). H2/H-infinity multiobjetive PID design. *IEEE Control Systems Magazine*, 37–47.

Teixeira, R. A. (2001). *Treinamento de Redes Neurais Artificiais através de Otimização Multi-Objetivo: Uma nova abordagem para o equilíbrio entre a polarização e variância*. Tese de Doutorado, Universidade Federal de Minas Gerais, MG, Brasil.

Teixeira, R. A., A. P. Braga, R. H. C. Takahashi, and R. R. Saldanha (2000). Improving generalization of MLPs with multi-objective optimization. *Neurocomputing 35*(1–4), 189–194.

Teodorescu, H. and C. Bonciu (1997). Learning algorithm for RBF networks as feature extractors. *First International Conference on Knowledge-Based Intelligent Electronic Systems*.

Tikhonov, A. N. (1963). Solution of incorrectly formulated problems and the regularization method. *Soviet Math 4*, 1035–1038.

Vapnik, V. (1995). The nature of statistical learning theory.

Vapnik, V. N. and A. Y. Chervonenkis (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theoritical Probability and Its Applications*.

Verleysen, M. and K. Hlavácková (1994). An optimized rbf network for approximation of functions. In *Proceedings of the 1994 European Symposium on Artificial Neural Networks*, pp. 175–180.

Wahba, G. (2000, February 28). Generalization and regularization in nonlinear learning system.

Weigend, A. S., D. E. Rumelhart, and B. A. Huberman (1990). Predicting the future, a connectionist approach. *International Journal of Neural Systems*, 193–209.

Whitehead, B. A. and T. D. Chaote (1994, January). Evolving space-filling curves to distribute Radial Basis Functions over an input space. *IEEE Transactions on Neural Networks* 5(1), 15–23.

Widrow, B. and M. E. Hoff (1960). Adaptive switching circuits. *IRE WESCON Convention Record 4*, 96–104.