

TESE DE DOUTORADO Nº 025

**CONTROLE POR MODOS DESLIZANTES DA GENERALIZAÇÃO EM  
APRENDIZADO DE REDES NEURAIIS ARTIFICIAIS**

*Marcelo Azevedo Costa*

DATA DA DEFESA: 18.12.2002

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA  
CENTRO DE PESQUISA E DESENVOLVIMENTO EM ENGENHARIA ELÉTRICA

CONTROLE POR MODOS DESLIZANTES DA  
GENERALIZAÇÃO EM APRENDIZADO DE REDES  
NEURAS ARTIFICIAIS

POR  
MARCELO AZEVEDO COSTA

Tese de Doutorado submetido à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como requisito parcial a obtenção de título de Doutor em Engenharia Elétrica

Orientador : Antônio de Pádua Braga  
Co-Orientador : Benjamim Rodrigues de Menezes

18 de Dezembro de 2002

**"CONTROLE POR MODOS DESLIZANTES DA  
GENERALIZAÇÃO EM APRENDIZADO DE REDES  
NEURAS ARTIFICIAIS"**

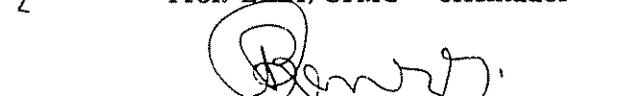
MARCELO AZEVEDO COSTA

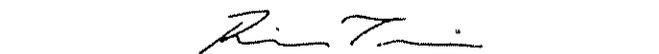
Tese de Doutorado submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do grau de Doutor em Engenharia Elétrica.

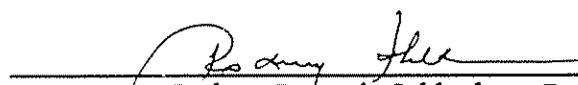
Aprovada em 18 de dezembro de 2002.

Por:

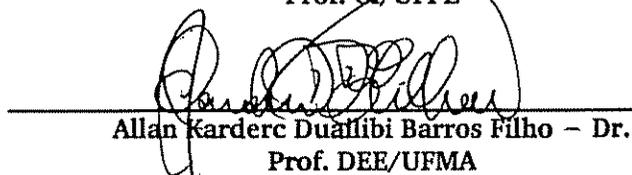
  
\_\_\_\_\_  
Antônio de Pádua Braga – Ph.D.  
Prof. DELT/UFMG – orientador

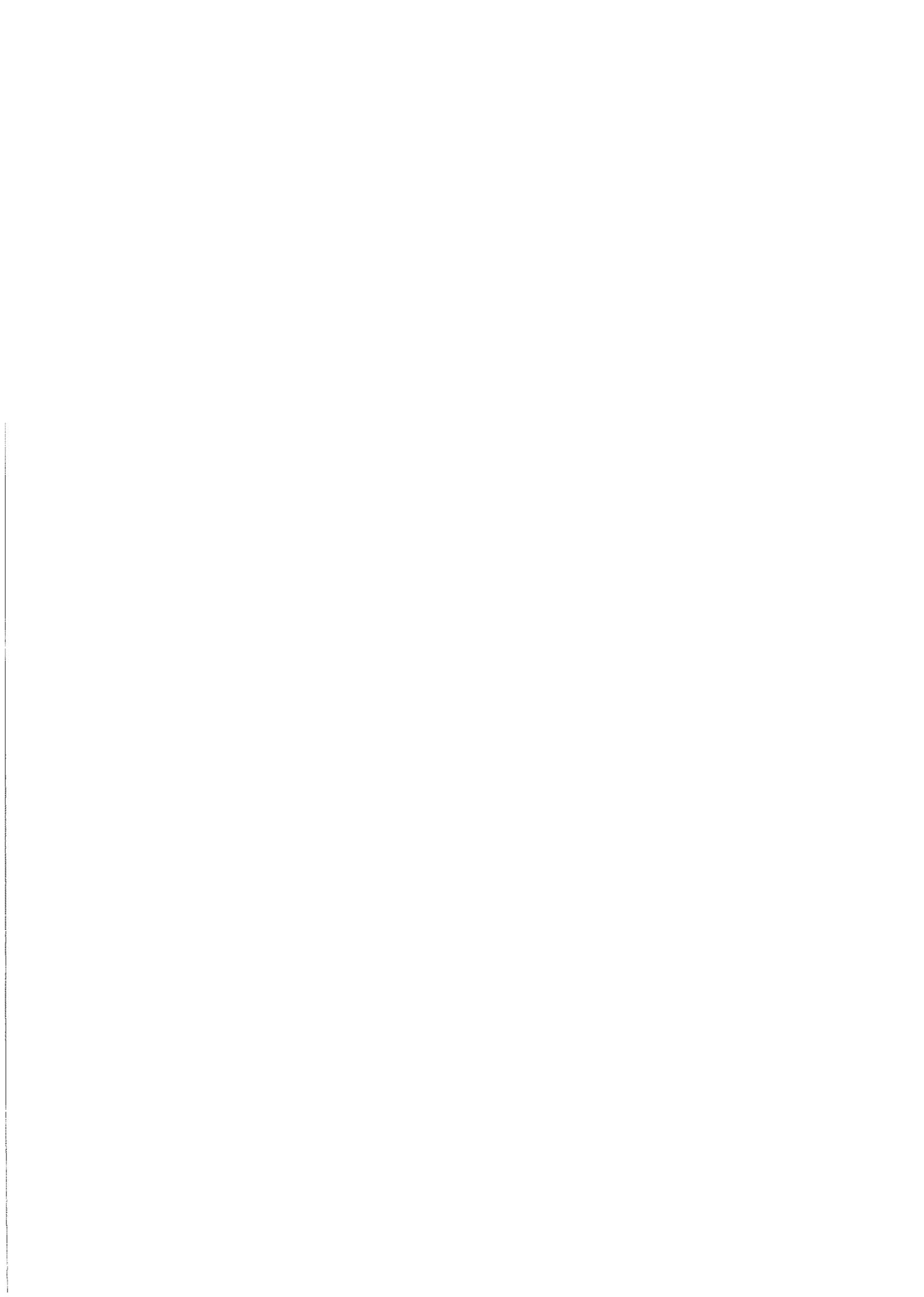
  
\_\_\_\_\_  
Benjamim Rodrigues de Menezes – Dr.  
Prof. DELT/UFMG – co-orientador

  
\_\_\_\_\_  
Ricardo Hiroshi Caldeira Takahashi – Dr.  
Prof. Depto. de Matemática/ICEx/UFMG

  
\_\_\_\_\_  
Rodney Rezende Saldanha – Dr.  
Prof. DEE/EEUFMG

  
\_\_\_\_\_  
Aluzio Fausto Ribeiro Araújo – Dr.  
Prof. CI/UFPE

  
\_\_\_\_\_  
Allan Karderc Duailibi Barros Filho – Dr.  
Prof. DEE/UFMA



# Agradecimentos

Ao Deus Altíssimo, o meu sincero louvor, pela Sua fidelidade, graça e infinita misericórdia.

À grande amiga e mãe Maria Helena pelo seu caráter, amor, paciência. Pelas palavras de confiança, conforto, pela presença marcante em minha vida. Tivemos momentos difíceis, mas a sua fé em Deus sempre foi um exemplo para mim.

Aos meus amados irmãos Érica e André, fiéis companheiros.

Aos amigos e eternos colegas do LITC Marcelo Barros, Éber, Roselito, Parma, Wilian, Lane, Bernardo, Mansur, Daniel e Eduardo que nunca pouparam esforços em me ajudar.

Aos Professores Antônio de Pádua Braga, que tem me orientado desde a graduação, durante 7 anos, e Benjamin Rodrigues de Menezes pelo apoio técnico e moral. pela paciência, compreensão e tranquilidade com que conduziram a orientação deste trabalho.

Aos meus amigos Walmir M. Caminhas, Luís Aguirre, Robinho, Rosane e Mércia que sempre estiveram dispostos a dar uma palavra amiga.

Aos colegas e amigos do CPDEE, em especial, Rojane, Glória, Elza, Henrique e Rogério.

Às inúmeras pessoas que participaram de alguma forma deste trabalho.

À CAPES pelo apoio e suporte financeiro.



*À minha mãe Maria Helena  
Ao meu pai João (in memoriam)  
Ao meu amigo Marcos (in memoriam)  
Aos meus irmãos Érica e André*



*Ainda que eu fale as línguas dos homens e dos anjos,  
se não tiver amor, serei como o bronze que soa, ou o címbalo que retine.  
Ainda que eu tenha o dom de profetizar e conheça todos os mistérios e toda a ciência;  
ainda que eu tenha tamanha fé ao ponto de transportar montes,  
se não tiver amor, nada serei.*

*E ainda que eu distribua todos os meus bens entre os pobres.  
e ainda que entregue o meu próprio corpo para ser queimado.  
se não tiver amor, nada disso me aproveitará.*

*O amor é paciente, é benigno, o amor não arde em ciúmes,  
não se ufana. não se ensoberbece.*

*Não se conduz inconvenientemente, não procura os seus interesses, não se exaspera,  
não se ressentido do mal; não se alegra com a injustiça, mas regozija-se com a verdade;  
tudo sofre, tudo crê, tudo espera, tudo suporta.*

*O amor jamais acaba; mas, havendo profecias, desaparecerão;  
havendo línguas, cessarão; havendo ciência, passará;*

*Agora, pois, permanecem a fé, a esperança e o amor, estes três:  
porém o maior destes é o amor.*

*I Coríntios 13:1-8,13*



# Resumo

Neste trabalho é desenvolvido um novo algoritmo multi-objetivo que utiliza controle por modos deslizantes para o treinamento de redes Multi-Layer Percetron (MLP). O algoritmo é capaz de controlar a trajetória da rede em um plano de estados definido por duas funções objetivo: o erro de treinamento e a norma do vetor de pesos. Os resultados obtidos demonstram a capacidade do método de encontrar um conjunto de soluções eficientes denominado conjunto Pareto-ótimo, sendo que a solução final, obtida a partir deste conjunto, corresponde a uma rede MLP com boa capacidade de generalização.

Variações para o algoritmo proposto são apresentados como o método multi-objetivo utilizando o algoritmo de Levenberg-Marquardt para treinamento de MLPs. Este algoritmo realiza o treinamento minimizando o erro de treinamento mas, restringindo a norma dos pesos a um valor pré-estabelecido. Soluções para diferentes valores de norma são geradas selecionando-se a solução que apresenta a melhor resposta de generalização. Os resultados demonstram a eficiência do algoritmo em relação ao aumento da velocidade de convergência e a capacidade de generalização da rede obtida.

A partir do treinamento multi-objetivo de redes MLPs, que permite obter redes com boa capacidade de generalização, métodos de poda ou simplificação foram desenvolvidos a fim de obter topologias reduzidas sem perda de generalização. Informações referentes às características da topologia e o erro de validação são utilizadas por esses métodos.

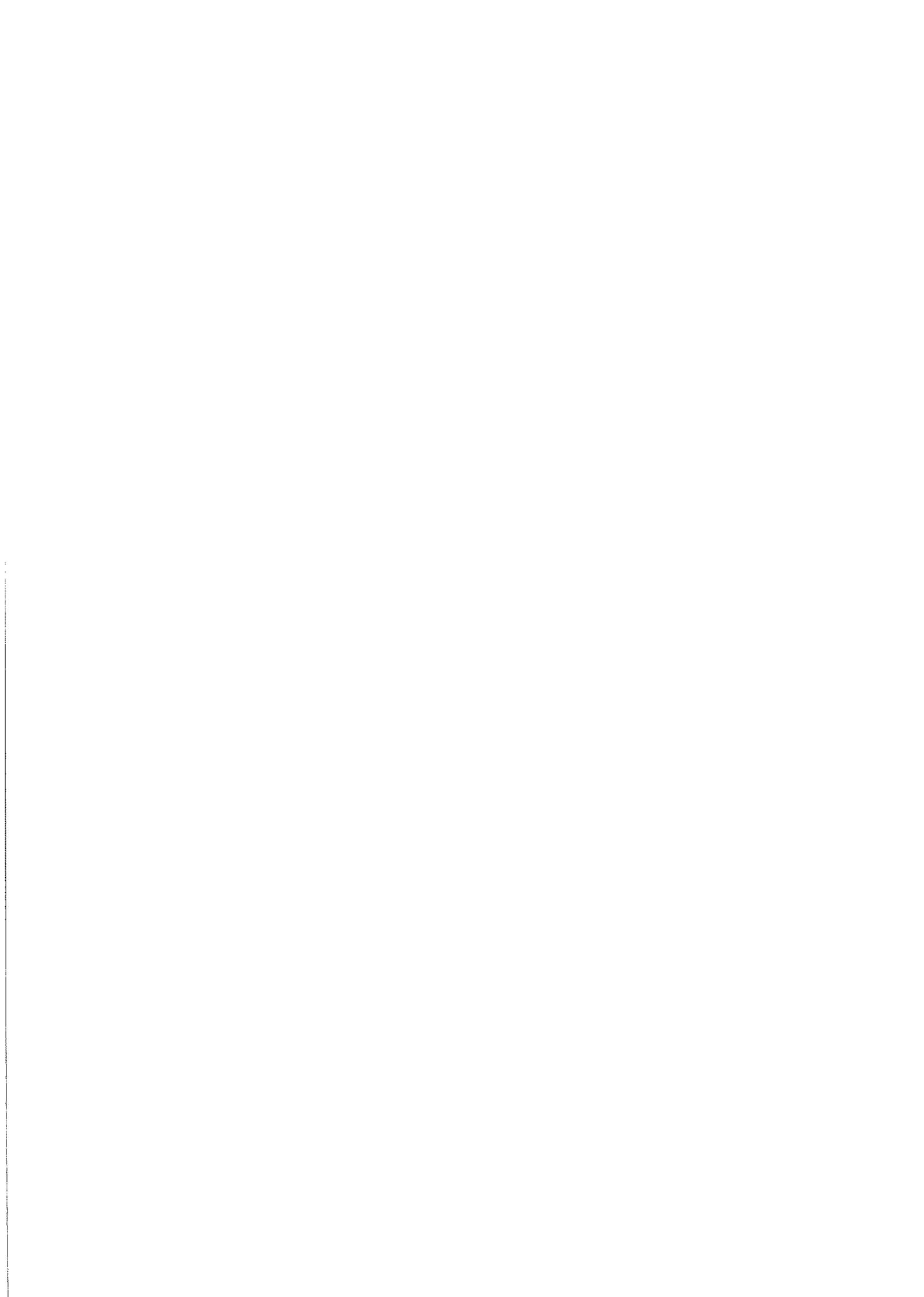


# Abstract

This work presents a new sliding mode control algorithm that is able to guide the trajectory of a Multi-Layer Perceptron (MLP) within the plane formed by two objective functions: training set error and norm of the weight vectors. Results show that the neural networks obtained are able to generate an approximation to the Pareto set, from which an improved generalization performance model is selected.

Variations for the proposed algorithm are also presented, as the multi-objective approach of the well known Levenberg-Marquardt algorithm for training neural networks. This algorithm restricts the norm of the weights vector to a preestablished norm value and finds the minimum error solution for that norm value. A range of different norm solutions is generated and the best generalization solution is selected. The results show the efficiency of the algorithm in terms of convergence speed and generalization performance.

Using the multi-objective algorithms to find solutions with good generalization performance, methods for pruning the original topology without loss in performance are presented. Topology information and validation sets are used. The results show that these techniques are able to choose a reduced network topology and to simplify trained networks.

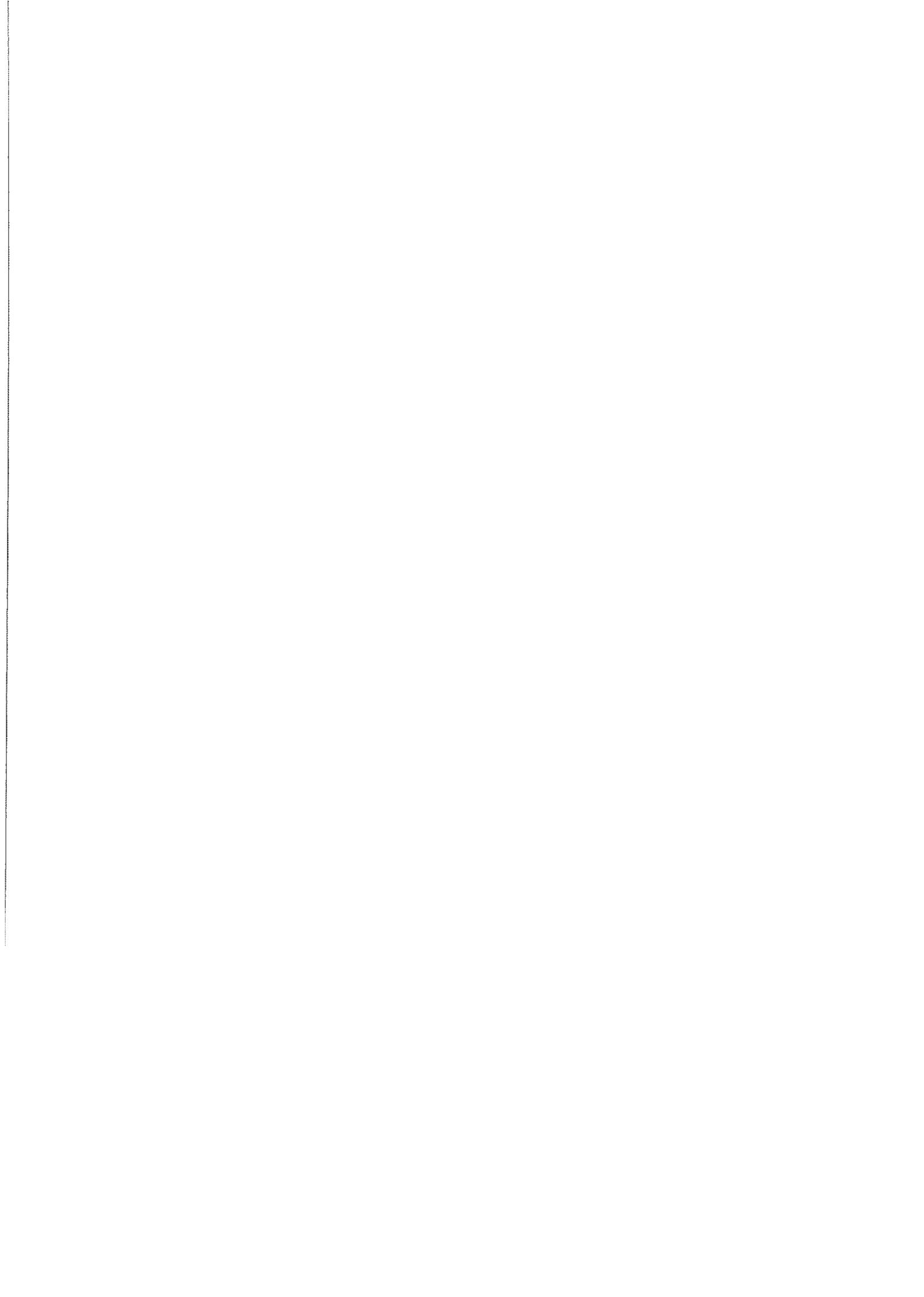


# Sumário

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introdução</b>   | <b>1</b>  |
| 1.1      | Motivações  | 3         |
| 1.2      | Metodologia   | 3         |
| 1.3      | Aplicações do Algoritmo MOBJ-SMC                                  | 6         |
| 1.4      | Organização do Texto  | 7         |
| <b>2</b> | <b>Redes Neurais Artificiais</b>                                  | <b>9</b>  |
| 2.1      | Introdução  | 9         |
| 2.2      | O Treinamento de RNAs   | 11        |
| 2.3      | Método do Gradiente Descendente                                   | 12        |
| 2.3.1    | ADALINE - ADaptive LInear NEuron                                  | 12        |
| 2.3.2    | Regra Delta   | 13        |
| 2.4      | Redes <i>Multi-Layer Perceptron</i> (MLP)                         | 14        |
| 2.4.1    | Projeto de RNAs   | 15        |
| 2.4.2    | O Algoritmo BackPropagation                                       | 16        |
| 2.4.3    | O Algoritmo de Levenberg-Marquardt                                | 17        |
| 2.4.4    | O Algoritmo Weight Decay  | 18        |
| 2.4.5    | O Algoritmo de Modos Deslizantes                                  | 19        |
| 2.4.6    | O Algoritmo Multi-Objetivo  | 21        |
| 2.5      | Conclusões do Capítulo  | 24        |
| <b>3</b> | <b>O Método Multi-Objetivo</b>                                    | <b>25</b> |
| 3.1      | A abordagem Mono-Objetivo no treinamento de MLPs                  | 25        |
| 3.2      | A abordagem Multi-Objetivo no treinamento de MLPs                 | 27        |
| 3.3      | O Comportamento Multi-Objetivo do Perceptron                      | 30        |
| 3.3.1    | Análise do Comportamento dos Gradientes                           | 32        |
| 3.4      | Representação dos Pesos em Coordenadas Polares                    | 35        |
| 3.5      | Análise dos Gradientes de uma Rede MLP de Duas Camadas            | 38        |
| 3.6      | Análise do Algoritmo BackPropagation no Plano Erro $\times$ Norma | 43        |
| 3.7      | Conclusões do Capítulo  | 45        |

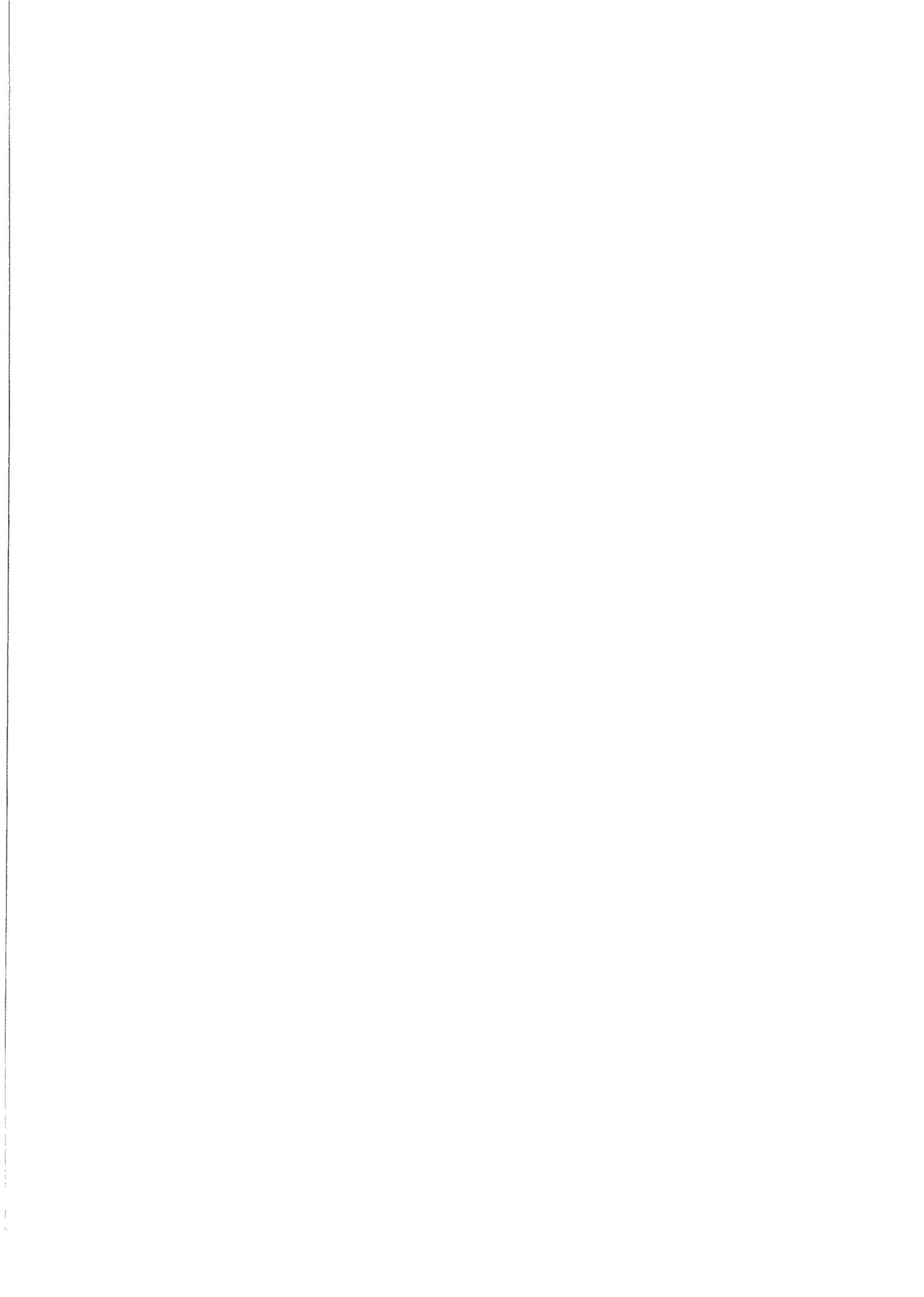
|          |   |            |
|----------|---|------------|
| <b>4</b> | <b>Algoritmo Proposto</b>   | <b>47</b>  |
| 4.1      | Algoritmo Proposto  | 47         |
| 4.1.1    | Algoritmo Proposto utilizando Teoria de Modos Deslizantes                               | 48         |
| 4.2      | Estudo da Convergência das Superfícies de Deslizamento                                  | 49         |
| 4.2.1    | Estudo da Convergência da Função Norma dos Pesos  | 60         |
| 4.2.1.1  | A natureza das equações de convergência da superfície da Norma                          | 62         |
| 4.2.2    | Estudo da Convergência da Função do Erro  | 65         |
| 4.2.2.1  | O Perceptron Linear   | 68         |
| 4.2.2.2  | O Perceptron Não-Linear   | 70         |
| 4.2.2.3  | Rede MLP de duas camadas  | 75         |
| 4.2.3    | Considerações sobre a Convergência do Método  | 81         |
| 4.3      | Conclusões do Capítulo  | 82         |
| <b>5</b> | <b>Determinação das Soluções Eficientes e Variações para o Algoritmo Proposto</b>       | <b>83</b>  |
| 5.1      | Obtenção do Conjunto Pareto através de Trajetórias no Plano de Soluções                 | 83         |
| 5.2      | Obtenção do Conjunto Pareto através da Restrição da Norma                               | 85         |
| 5.3      | Variações do algoritmo Multi-Objetivo proposto  | 86         |
| 5.3.1    | O Algoritmo Multi-Objetivo Utilizando o Método de Levenberg-Marquardt                   | 87         |
| 5.4      | Treinamento Multi-Objetivo através do Controle da Taxa de Aprendizado                   | 88         |
| 5.4.1    | Controle da norma de MLPs através da limitação da taxa de aprendizado: $\alpha$         | 89         |
| 5.4.1.1  | Casos onde ocorre o decréscimo da norma dos pesos                                       | 92         |
| 5.4.1.2  | Casos onde ocorre o crescimento da norma dos pesos                                      | 94         |
| 5.4.1.3  | Descrição do Algoritmo  | 97         |
| 5.5      | Conclusões do Capítulo  | 99         |
| <b>6</b> | <b>Aplicação do Método Proposto: Simulações e Resultados</b>                            | <b>101</b> |
| 6.1      | Descrição dos Problemas   | 101        |
| 6.2      | Problemas de Classificação  | 102        |
| 6.2.1    | Separação de duas Classes no Espaço Bi-dimensional                                      | 103        |
| 6.2.2    | Classificação: Adult Database   | 103        |
| 6.3      | Regressão   | 106        |
| 6.4      | Controle de trajetórias no Espaço Solução   | 109        |
| 6.4.1    | Controle de Trajetórias na Região Factível  | 109        |
| 6.4.2    | Análise da Variação do Erro e da Norma a Partir de um Ponto Definido no Conjunto Pareto | 111        |

|          |   |            |
|----------|---|------------|
| 6.4.3    | Variações do algoritmo multi-objetivo: o algoritmo MOBJ-SMC-LM                                | 113        |
| 6.4.4    | O Controle da norma através da taxa de aprendizado  | 115        |
| 6.5      | Predição de Séries Temporais  | 116        |
| 6.6      | Conclusões do Capítulo  | 119        |
| <b>7</b> | <b>Métodos de Pruning Aplicados a Soluções Multi-Objetivo</b>                                 | <b>121</b> |
| 7.1      | Introdução  | 121        |
| 7.2      | Identificação Multi-Objetivo de Topologias  | 122        |
| 7.2.1    | Descrição Matemática do Método  | 126        |
| 7.3      | Comportamento Linear/não-Linear do nodo Perceptron  | 128        |
| 7.4      | Identificação de Nodos com Semelhança de Respostas  | 135        |
| 7.5      | Simplificação de Topologias pela Simples Extração de Conexões                                 | 139        |
| 7.6      | Simplificação de Topologias Utilizando um Método de Saliência                                 | 140        |
| 7.7      | Associação de Técnicas para Simplificação de Topologias                                       | 142        |
| 7.8      | Identificação de Topologias Utilizando re-Treinamento   | 144        |
| 7.9      | Conclusões do Capítulo  | 146        |
| <b>8</b> | <b>Conclusões e Propostas de Continuidade</b>   | <b>149</b> |
| 8.1      | Propostas de continuidade   | 151        |
| <b>A</b> | <b>Apêndice: Equações Matemáticas e Propriedades</b>  | <b>159</b> |
| A.1      | Transformação de Coordenadas em $\mathbb{R}^2$  | 159        |
| A.2      | A equação geral do segundo grau: o Indicador $I = b^2 - 4ac$                                  | 160        |
| A.2.1    | Caso particular   | 160        |
| A.3      | Determinação das raízes de um polinômio de 2º grau  | 161        |
| <b>B</b> | <b>Apêndice: Resultados para o Banco de Dados Proben - Classificação (Cancer, Card, Gene)</b> | <b>163</b> |



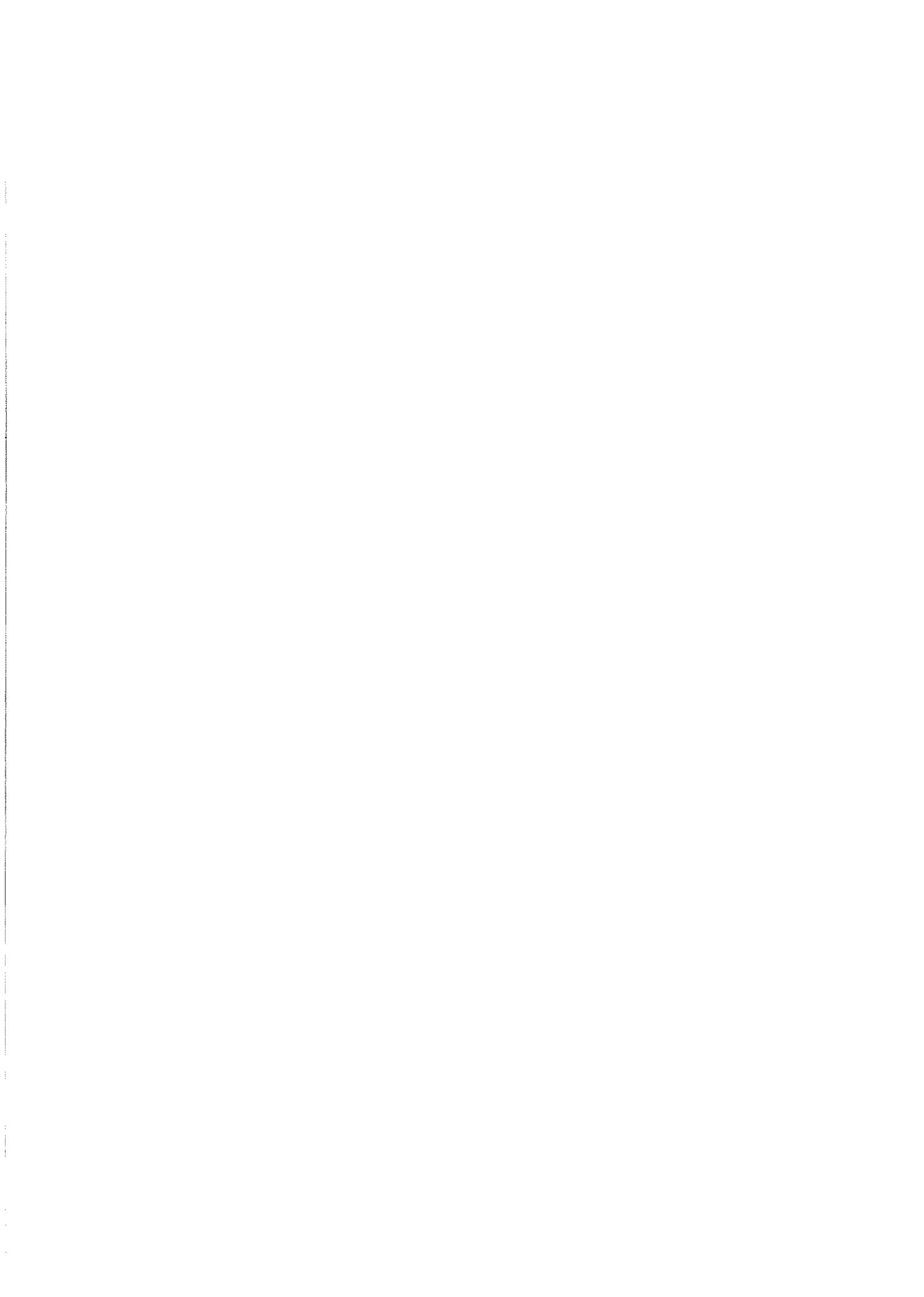
# Abreviaturas

|                  |   |
|------------------|---|
| RNA              | Redes Neurais Artificiais                                       |
| MCP              | McCulloch e Pitts (modelo de RNA proposto, 1943)                |
| MLP              | Redes Neurais Multi-Camadas ( <i>Multi-Layer Perceptron</i> )   |
| MOBJ             | Algoritmo Multi-Objetivo ( <i>Multi-Objective</i> )             |
| SMC              | Algoritmo de Modos Deslizantes ( <i>Sliding Mode Control</i> )  |
| SMC-MOBJ         | Algoritmo de Modos Deslizantes Multi-Objetivo                   |
| MOBJ-SMC-LM      | Algoritmo Multi-Objetivo de Levenberg-Marquardt                 |
| RPROP            | Resilient PROPagation   |
| MOBJ-BP          | Algoritmo Multi-Objetivo com Controle da Taxa de Aprendizado    |
| SVM              | Máquina de vetores de suporte ( <i>Support Vector Machine</i> ) |
| SVM <sup>l</sup> | <i>Support Vector Machine light</i>                             |
| SOR              | <i>Successive Overrelaxation</i>                                |
| OBS              | Algoritmo de <i>pruning</i> ( <i>Optimal Brain Surgeon</i> )    |
| OBD              | Algoritmo de <i>pruning</i> ( <i>Optimal Brain Damage</i> )     |
| RBF              | Redes de Base Radial ( <i>Radial Basis Function</i> )           |



# Lista de Símbolos

|                         |  |
|-------------------------|--|
| $\alpha$                | Taxa de aprendizado do gradiente do erro;                            |
| $\beta$                 | Taxa de aprendizado da função da norma;                              |
| $S_{E(k)}$              | Superfície de deslizamento do erro;                                  |
| $S_{\ \mathbf{w}(k)\ }$ | Superfície de deslizamento da norma;                                 |
| $E(k)$                  | Função do erro: Somatório do erro médio quadrático no instante $k$ ; |
| $\mathbf{w}$            | Vetor de pesos da RNA;   |
| $d$                     | Saída desejada para a RNA;   |
| $y$                     | Saída real da RNA;   |
| $J$                     | Função custo da RNA;   |
| $\eta$                  | Constante de proporcionalidade;                                      |
| $\nabla J$              | Gradiente da função custo;   |
| $f(\cdot)$              | Função de ativação;  |
| $f'(\cdot)$             | Derivada da função de ativação;                                      |
| $e(k)$                  | Diferença entre a saída desejada e a saída da RNA;                   |
| $w_{ji}$                | Peso $ji$ de uma rede MLP;   |
| $V(x)$                  | Soma das funções quadráticas do erro;                                |
| $\nabla V(x)$           | Gradiente de $V(x)$ ;  |
| $\nabla^2 V(x)$         | Hessiana de $V(x)$ ;   |
| $\mu$                   | Parâmetro de ajuste do método Levenberg-Marquardt;                   |
| $\lambda$               | Parâmetro de regularização (Weight-Decay);                           |
| $\ \mathbf{w}\ $        | Norma euclidiana (comprimento) do vetor de pesos de uma RNA;         |
| $x_{j(k)}$              | Entrada $j$ da RNA no instante $k$ ;                                 |
| $\theta$                | Vetor de ângulos de uma rede MLP;                                    |
| $w_2$                   | Vetor de pesos da camada de saída;                                   |
| $b_2$                   | Vetor de polarização da camada de saída;                             |
| $w_1$                   | Vetor de pesos da camada de entrada;                                 |
| $b_1$                   | Vetor de polarização da camada de entrada;                           |
| $c$                     | Termo de multiplicação para ajuste da norma;                         |
| $\mathbf{g}$            | Vetor gradiente dos pesos da MLP;                                    |



# Lista de Figuras

|      |   |    |
|------|---|----|
| 1.1  | Comportamento do Conjunto Pareto no Plano de Soluções . . . . .                                   | 4  |
| 1.2  | Comportamento do Algoritmo SMC . . . . .  | 5  |
| 1.3  | Deslocamento da MLP no Plano de Soluções (MOBJ-SMC) . . . . .                                     | 6  |
| 1.4  | Problema de Regressão - solução MOBJ-SMC . . . . .  | 7  |
| 1.5  | Problema de Classificação - solução MOBJ-SMC . . . . .  | 7  |
| 2.1  | Modelo de um Nodo ADALINE . . . . .   | 12 |
| 2.2  | Exemplo de uma rede MLP de três camadas . . . . .   | 14 |
| 2.3  | Curva Pareto e vetores $v_k$ definidos no plano $\ w\  \times e^2$ . . . . .                      | 23 |
| 3.1  | Desempenho de algoritmos no espaço solução (Erro $\times$ Épocas) . . . . .                       | 26 |
| 3.2  | Curva Pareto-ótimo no Plano de Soluções . . . . .   | 28 |
| 3.3  | Solução utópica (não-existente) . . . . .   | 29 |
| 3.4  | Superfície do Erro . . . . .  | 30 |
| 3.5  | Superfície da Norma . . . . .   | 30 |
| 3.6  | Curvas de nível das funções Erro e Norma . . . . .  | 31 |
| 3.7  | Plano de Soluções: Erro $\times$ Norma . . . . .  | 31 |
| 3.8  | Conjunto Pareto-Ótimo . . . . .   | 31 |
| 3.9  | Superfície do Erro: Conjunto Pareto-Ótimo . . . . .   | 32 |
| 3.10 | Conjunto Pareto-Ótimo: Curvas Equipotenciais de Erro e Norma . . . . .                            | 32 |
| 3.11 | Derivada - Função Hiperbólica . . . . .   | 33 |
| 3.12 | Derivada - Função Sigmoidal . . . . .   | 33 |
| 3.13 | Representação dos pesos na forma polar - bidimensional . . . . .                                  | 36 |
| 3.14 | Representação polar - tridimensional . . . . .  | 36 |
| 3.15 | Soluções de raio $r = 0.8396$ . . . . .   | 37 |
| 3.16 | Soluções de raio $r = 3.5$ . . . . .  | 37 |
| 3.17 | $f_i$ - $\tanh(x)$ e $f_p$ - $\text{linear}(x)$ . . . . .   | 40 |
| 3.18 | $f_i$ - $\tanh(x)$ e $f_p$ - $\text{sigmoidal}(x)$ . . . . .                                      | 41 |
| 3.19 | $f_i$ - $\tanh(x)$ e $f_p$ - $\tanh(x)$ . . . . .   | 41 |
| 3.20 | Comportamento do Algoritmo Backpropagation no Plano de Soluções: Erro<br>$\times$ Norma . . . . . | 44 |

|      |   |     |
|------|---|-----|
| 3.21 | Backpropagation com valores reduzidos de norma e taxa de aprendizado . . .              | 45  |
| 4.1  | Regiões definidas pelas superfícies de deslizamento . . . . .                           | 52  |
| 4.2  | Intervalo de convergência para o ganho $\alpha$ sob a condição: $S_{(k)} > 0$ . . . . . | 57  |
| 4.3  | Intervalo de convergência para o ganho $\alpha$ sob a condição: $S_{(k)} < 0$ . . . . . | 58  |
| 4.4  | Região de convergência para $S_{\ \mathbf{w}_{(k)}\ } > 0$ . . . . .                    | 65  |
| 4.5  | Região de convergência para $S_{\ \mathbf{w}_{(k)}\ } < 0$ . . . . .                    | 65  |
| 4.6  | Nodo perceptron com Função de Ativação $f(u)$ . . . . .                                 | 71  |
| 4.7  | Camada de saída de uma rede MLP . . . . .   | 76  |
| 4.8  | Região de convergência para $S_{E_{(k)}} > 0$ . . . . .                                 | 80  |
| 4.9  | Região de convergência para $S_{E_{(k)}} < 0$ . . . . .                                 | 80  |
| 4.10 | Região de convergência para uma rede MLP de duas camadas . . . . .                      | 80  |
| 4.11 | Região de convergência para uma rede MLP de duas camadas . . . . .                      | 81  |
| 5.1  | Método para a obtenção da Curva Pareto . . . . .  | 84  |
| 5.2  | Aproximação de soluções eficientes de norma elevada . . . . .                           | 85  |
| 5.3  | Aproximação de soluções eficientes de norma reduzida . . . . .                          | 85  |
| 5.4  | Método de Geração do Conjunto Pareto através da Restrição da Norma . . . . .            | 86  |
| 5.5  | Formato da parábola para $a > 0$ . . . . .  | 96  |
| 6.1  | Classificação de duas classes com distribuição bivariada . . . . .                      | 103 |
| 6.2  | Resultados obtidos para a regressão da função $d(x)$ . . . . .                          | 107 |
| 6.3  | Resultados obtidos para a regressão da função seno . . . . .                            | 108 |
| 6.4  | Resultados obtidos para a regressão da função seno amortecido . . . . .                 | 108 |
| 6.5  | Resultados obtidos para a regressão da função <i>sinc</i> . . . . .                     | 108 |
| 6.6  | Exemplos de trajetórias na região factível . . . . .                                    | 110 |
| 6.7  | Exemplos de trajetórias no Plano de Soluções . . . . .                                  | 110 |
| 6.8  | Resposta da MLP após treinamento utilizando o algoritmo SMC-MOBJ . . . . .              | 111 |
| 6.9  | Soluções da MLP para $\ \mathbf{w}_{(k)}\ $ constante . . . . .                         | 112 |
| 6.10 | Soluções da MLP para $E_{(k)}$ constante . . . . .                                      | 112 |
| 6.11 | Soluções obtidas com o algoritmo MOBJ-SMC-LM . . . . .                                  | 113 |
| 6.12 | Análise da velocidade de convergência do algoritmo MOBJ-SMC-LM . . . . .                | 114 |
| 6.13 | O conjunto Pareto e a solução de Levenberg-Marquardt . . . . .                          | 115 |
| 6.14 | Soluções obtidas com o algoritmo MOBJ-BP (1) . . . . .                                  | 115 |
| 6.15 | Soluções obtidas com o algoritmo MOBJ-BP (2) . . . . .                                  | 116 |
| 6.16 | Série temporal utilizada . . . . .  | 117 |
| 6.17 | Soluções multi-objetivo para a predição da Série do Boi Gordo (1) . . . . .             | 117 |
| 6.18 | Soluções multi-objetivo para predição da Série do Boi Gordo (2) . . . . .               | 118 |
| 6.19 | Soluções multi-objetivo para predição da Série do Boi Gordo (3) . . . . .               | 118 |

|      |  |     |
|------|--|-----|
| 7.1  | Comportamento do conjunto Pareto-ótimo com o aumento dos nodos da camada escondida . . . . . | 123 |
| 7.2  | Curvas Pareto obtidas . . . . .  | 124 |
| 7.3  | Soluções obtidas . . . . .   | 124 |
| 7.4  | Comportamento da Curva Pareto em função da Topologia . . . . .                               | 125 |
| 7.5  | Soluções obtidas . . . . .   | 125 |
| 7.6  | Conjunto tridimensional de soluções <i>eficientes</i> . . . . .                              | 127 |
| 7.7  | Convergência do conjunto Pareto com o aumento dos nodos <i>escondidos</i> . . . . .          | 127 |
| 7.8  | Funções de ativação: tangente hiperbólica e linear . . . . .                                 | 128 |
| 7.9  | Rede MLP antes (a) e após (b) simplificação . . . . .  | 130 |
| 7.10 | Resposta da MLP antes e após simplificação . . . . .   | 132 |
| 7.11 | Topologia simplificada . . . . .   | 132 |
| 7.12 | Erro de Treinamento em Função dos Nodos Restantes . . . . .                                  | 145 |
| 7.13 | Erro de Validação em Função dos Nodos Restantes . . . . .                                    | 145 |
| A.1  | Comportamento da concavidade da parábola em relação ao coeficiente $a$ . . . . .             | 162 |



# Lista de Tabelas

|      |   |     |
|------|---|-----|
| 3.1  | Pontos singulares e intervalos de confiança . . . . .   | 43  |
| 4.1  | Funções de ativação e suas inversas . . . . .   | 72  |
| 4.2  | Erros médios e intervalos de confiança para a aproximação da função Tangente Hiperbólica utilizando a expansão de primeira ordem em Série de Taylor . . . . . | 73  |
| 5.1  | Características do Algoritmo Proposto (SMC-MOBJ) e variações . . . . .  | 99  |
| 6.1  | Número de pares entrada-saída (padrões) dos conjuntos de treinamento e validação . . . . .  | 104 |
| 6.2  | Resultados do problema de classificação com os dados provenientes do <i>Adult Database</i> . . . . .  | 104 |
| 6.3  | Tabela de porcentagem de acertos: médias e desvios . . . . .  | 105 |
| 6.4  | Erro obtido para o conjunto de treinamento ( $E = \sum e^2$ ): médias e desvios   | 106 |
| 6.5  | Norma da MLP obtida ( $\ \mathbf{w}\ $ ): médias e desvios . . . . .  | 106 |
| 7.1  | Desempenho das soluções MOBJ em função da variação dos nodos da camada escondida . . . . .  | 123 |
| 7.2  | Identificação de topologias de RNAs utilizando um treinamento multi-objetivo . . . . .  | 124 |
| 7.3  | Performance do Método de Simplificação - Regressão Função Seno . . . . .  | 133 |
| 7.4  | Características iniciais após treinamento MOBJ . . . . .  | 133 |
| 7.5  | Resultados provenientes do processo de simplificação proposto . . . . .   | 134 |
| 7.6  | Resultados provenientes da simplificação por semelhança de respostas . . . . .  | 139 |
| 7.7  | Resultados provenientes da simplificação pela simples extração de conexões  | 140 |
| 7.8  | Resultados provenientes da simplificação utilizando o método OBD . . . . .  | 142 |
| 7.9  | Resultados provenientes da associação dos métodos de simplificação . . . . .  | 143 |
| 7.10 | Resultados provenientes dos processos de simplificação combinados e com ajuste do conjunto de validação . . . . .   | 144 |
| 7.11 | Resultados do processo de identificação de topologias utilizando re-treinamento . . . . .   | 145 |

|     |   |     |
|-----|---|-----|
| B.1 | Resultados obtidos através dos dados: Cancer/Proben . . . . . | 164 |
| B.2 | Resultados obtidos através dos dados: Card/Proben . . . . .   | 164 |
| B.3 | Resultados obtidos através dos dados: Gene/Proben . . . . .   | 164 |

# Capítulo 1

## Introdução

No desenvolvimento de modelos neurais é desejável que os mesmos apresentem respostas coerentes em relação a padrões desconhecidos ou que não foram apresentados na etapa do treinamento. Essa característica é definida como *generalização*. Diversos fatores exercem influência na capacidade da generalização de uma Rede Neural Artificial (RNA), dentre esses, a sua topologia, a representabilidade do conjunto de treinamento, a existência de ruído incorporado aos dados e o tipo de algoritmo utilizado para o treinamento.

A topologia de uma RNA, definida pelo número de entradas da rede, saídas, número de camadas, número de nodos por camada e função de ativação, influi na sua capacidade de aprendizado. Uma rede composta por nodos lineares ou conexões lineares entre as entradas e saídas, apresentará uma resposta puramente linear, mesmo que a função geradora do conjunto de treinamento apresente um comportamento mais complexo, limitando o aprendizado da rede. O mesmo comportamento se observa em relação ao treinamento utilizando redes sub-dimensionadas no qual o modelo ou a topologia utilizada é insuficiente para garantir uma boa aproximação da função geradora expressa no conjunto de treinamento. A principal característica observada nesse tipo de treinamento consiste na similaridade das soluções encontradas, definida como *polarização* [Geman et al., 1992]. As soluções *polarizadas* de um modelo neural possuem características de erro em relação ao conjunto de treinamento e validação semelhantes, sem variabilidade e com pouca generalização. A fim de solucionar este problema, o projetista deve optar por topologias mais complexas, mais flexíveis, adicionando mais nodos ou múltiplas camadas à topologia original. Entretanto, um super-dimensionamento da rede pode provocar outro efeito indesejado na resposta do modelo neural. Redes *superdimensionadas* são capazes de se ajustar ao conjunto de treinamento, alcançando soluções de erro mínimo e modelando o ruído presente nos dados. Tal efeito provoca a perda da capacidade de generalização do modelo. As respostas obtidas a partir de modelos superdimensionados apresentam alta *variância*.

A melhor topologia ou modelo para um determinado problema deve ser suficientemente

flexível para evitar soluções polarizadas ou *underfitting*. Mas o grau de flexibilidade ou complexidade do modelo também deve ser limitado, evitando a variabilidade de soluções ou *overfitting*. Na literatura, esse equilíbrio entre a polarização e a variância é conhecido como “o dilema entre polarização e a variância” [Geman et al., 1992]. Entretanto, nem sempre é possível medir a complexidade de um problema, o que pode tornar a escolha da topologia um processo empírico.

A representabilidade dos dados visa garantir que a amostra ou o conjunto de dados utilizado no treinamento apresente as características desejáveis e que deverão ser assimiladas pela RNA. Um conjunto pode ser definido como representativo se for possível, a partir desse conjunto, obter um modelo cuja resposta seja próxima à função geradora. É comum dizer que o conjunto de treinamento deve ser estatisticamente representativo.

Uma vez conhecida a capacidade de aproximação de funções das redes Multi-Layer-Perceptrons (MLPs), que podem ser definidas como aproximadores universais de funções [Cybenko, 1989], e definido um conjunto de treinamento representativo, a principal dificuldade é estimar a topologia que apresente o melhor resultado, ou a melhor aproximação. Normalmente, podem-se especificar várias topologias de MLPs efetuando-se diversos treinamentos e escolhendo-se aquela que apresentar o melhor desempenho.

Diversos algoritmos que buscam aprimorar a capacidade de generalização de MLPs são propostos na literatura [Reed, 1993] como: Weight-Decay [Hinton and Nowlan, 1987], Early-Stopping [Weigend et al., 1990], Cross-Validation [Stone, 1974], Optimal-Brain-Surgeon (*OBS*) [Hassibi and Stork, 1993], Multi-Objective (*MOBJ*) [Teixeira et al., 2000]. Alguns algoritmos utilizam técnicas de construção, alterando a topologia da MLP. Os métodos de *poda*, também conhecidos como métodos de *pruning*, necessitam de uma rede super-dimensionada já treinada. Outros métodos utilizam técnicas de restrição dos valores dos pesos (parâmetros de ajuste) da MLP sem alterar a topologia original (Weight-Decay, *MOBJ*).

Dentre os diversos métodos utilizados para aprimorar a capacidade de generalização das redes MLPs, o método multi-objetivo realiza o treinamento através da minimização de duas funções: o erro médio quadrático e a norma do vetor de pesos da rede. Durante o treinamento, o método busca por um conjunto de soluções denominado conjunto Pareto-ótimo [Pareto, 1896] e representado por soluções de erro mínimo para valores específicos de norma. Uma vez obtido este conjunto, a solução final é escolhida utilizando-se um conjunto de validação. A rede que apresenta o menor erro em relação a esse conjunto corresponde à solução final. O método é capaz de alcançar em um único treinamento, uma solução ou rede com boa generalização. Para isso, é necessário que o projetista defina uma topologia super-dimensionada para o treinamento, normalmente, de uma maneira empírica.

## 1.1 Motivações

Neste trabalho, foram desenvolvidas variações para o método MOBJ proposto [Teixeira et al., 2000]. O método original utiliza técnicas de otimização multi-objetivo, como o problema  $\epsilon$ -restrito [Chankong and Haimes, 1983] e uma variante desse problema denominado de técnica de relaxação [Takahashi et al., 1997]. No presente trabalho, o método multi-objetivo foi implementado através do uso de técnicas de controle por Modos Deslizantes [Itkis, 1976], que tornou mais simples a descrição do problema e a implementação da equação de correção dos parâmetros de ajuste das RNAs.

Apesar de o método proposto apresentar uma velocidade de convergência inferior ao método MOBJ, as soluções obtidas são coerentes. Posteriormente, em virtude de o método proposto utilizar uma técnica baseada no controle da direção dos gradientes das funções de custo, erro e norma, foi possível implementar técnicas de otimização mais eficientes, acelerando a convergência do método sem perda de eficiência.

Uma vez que os métodos multi-objetivo são capazes de gerar soluções com alta capacidade de generalização para redes super-dimensionadas, foram utilizadas técnicas de *pruning* para reduzir a dimensão dos parâmetros dos modelos neurais mantendo a capacidade de generalização original. Utilizando os conceitos do conjunto Pareto-ótimo, também foi possível implementar um algoritmo *construtivo* para identificar uma topologia reduzida capaz de obter uma resposta de generalização coerente com aquelas obtidas para redes superdimensionadas.

## 1.2 Metodologia

O algoritmo proposto [Costa et al., 2003, Costa et al., 2002c] surge como consequência da aplicação de uma técnica de controle conhecida como Teoria de Modos Deslizantes (*Sliding Mode Control: SMC*), já utilizada no treinamento mono-objetivo de MLPs [Parma et al., 1999, Parma et al., 1998], e dos princípios do algoritmo Multi-Objetivo [Teixeira et al., 2000].

O algoritmo proposto, definido como algoritmo Multi-Objetivo com Controle por Modos Deslizantes (*Sliding Mode Control Multi-Objective : SMC-MOBJ*) apresenta as seguintes características em sua formulação:

1. O algoritmo Multi-Objetivo (MOBJ) [Teixeira et al., 2000] permite obter um conjunto de soluções através de um processo de minimização em um espaço bi-dimensional definido pelas funções do Erro e da Norma dos pesos da MLP. Através da definição do Conjunto Pareto-ótimo, que representa o limite no plano de soluções entre as soluções factíveis e as soluções não-existentis, o algoritmo Multi-Objetivo encontra pontos pertencentes a esse conjunto e analisa a sua capacidade de gene-

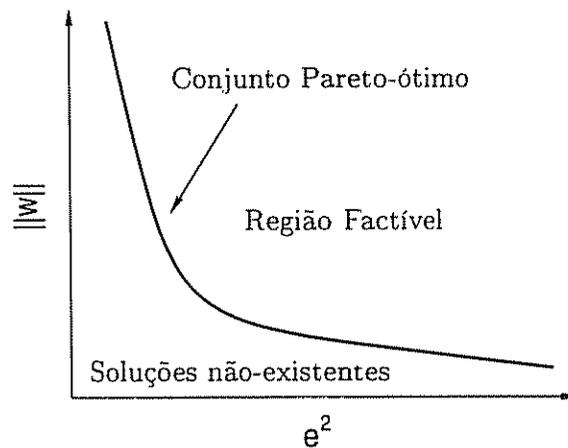


Figura 1.1: Comportamento do Conjunto Pareto no Plano de Soluções

realização. O ponto ou solução que apresenta a melhor capacidade de generalização representa a MLP desejada.

Em um problema multi-objetivo existe um conjunto de restrições de igualdade e desigualdade em relação ao vetor de parâmetros de interesse que restringe a espaço de busca de soluções a um subconjunto denominado conjunto *factível* de soluções. Em relação ao treinamento multi-objetivo de MLPs, o vetor de parâmetros de interesse é representado pelo vetor de pesos da rede,  $w$ . Não existem, no método proposto por Teixeira [Teixeira, 2001], restrições de igualdade ou desigualdade em relação ao vetor de pesos. Dessa forma, o conjunto factível de soluções é representado pelos vetores definidos no espaço  $\mathcal{R}^n$ , onde  $n$  é a dimensão do vetor de pesos da rede MLP. As soluções factíveis podem ser mapeadas no plano  $\mathcal{R}^2$  utilizando as funções ou *funcionais* Erro ( $E(w)$ ) e Norma ( $\|w\|$ ) definindo uma *região factível* nesse plano. Cada ponto nesse plano caracteriza um estado de erro e norma. Os pontos pertencentes a esse plano cujas características de Erro e Norma não podem ser definidas por um único vetor de pesos em  $\mathcal{R}^n$  caracterizam uma segunda região denominada *região de soluções não-existentes*.

A Figura 1.1 permite caracterizar o espaço de estados ou o plano de soluções definido pelas funções do erro e da norma, a região de soluções factíveis, a região de soluções não-existentes e a curva Pareto-ótimo da qual são extraídas as soluções. Uma vez obtidas as soluções da curva Pareto, um conjunto de validação é utilizado para definir a solução final que apresenta o menor erro de validação. Esse ponto representa uma rede MLP (Multi-Layer-Perceptron) [Rumelhart et al., 1986] cujos valores de norma e erro de treinamento caracterizam a sua capacidade de generalização.

2. O algoritmo SMC utiliza uma superfície denominada *Superfície de Deslizamento* para *orientar* o treinamento da rede MLP. A superfície é definida em função do erro

e da derivada do erro. O treinamento consiste na convergência do estado inicial da rede à superfície e posterior convergência à sua origem, que corresponde ao ponto onde o erro e a sua derivada são nulos. Na Figura 1.2 é possível identificar a superfície de deslizamento, que é definida por uma reta que passa pela origem. O método de Modos Deslizantes possibilita a convergência de um ponto qualquer a essa superfície e posterior convergência à sua origem. Dessa forma, se um sistema é controlado por este método o seu estado converge para a superfície e, uma vez sobre a superfície, o sistema é controlado para que o estado se desloque em direção à origem.

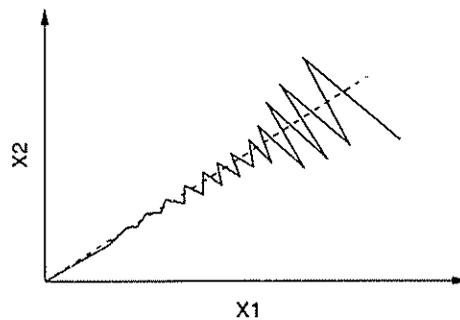


Figura 1.2: Comportamento do Algoritmo SMC

Uma vez demonstradas as principais características dos algoritmos MOBJ e SMC, o algoritmo SMC-MOBJ propõe o uso de coordenadas do plano de soluções (Erro  $\times$  Norma) para definir superfícies de deslizamento a fim de conduzir a função do Erro e da Norma da MLP até um determinado ponto no plano. O algoritmo conduz a rede a um estado de erro e norma pré-estabelecidos ( $E_t, \|\mathbf{w}_t\|$ ) através da definição de duas superfícies de deslizamento: a primeira definida pela diferença entre o erro atual da MLP e o erro desejado ( $S_{E(k)} = E(k) - E_t$ ) e a segunda, em função da diferença entre a norma<sup>1</sup> atual da MLP e a norma desejada ( $S_{\|\mathbf{w}(k)\|} = \|\mathbf{w}(k)\|^2 - \|\mathbf{w}_t\|^2$ ). A equação de ajuste dos pesos apresenta duas parcelas: uma referente ao gradiente da função do erro em relação aos pesos e a segunda referente ao gradiente da função da norma em relação aos pesos. Os gradientes são *controlados* pelos sinais das superfícies de deslizamento conduzindo a MLP ao erro e norma desejados. A equação final do ajuste dos pesos é descrita na Equação 1.1.

$$\Delta w_{ji(k)} = -\alpha \cdot \text{sgn}(S_{E(k)}) \cdot \frac{\partial E(k)}{\partial w_{ji(k)}} - \beta \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot w_{ji(k)} \quad (1.1)$$

onde  $S_{E(k)}$  e  $S_{\|\mathbf{w}(k)\|}$  correspondem às superfícies de deslizamento,  $\alpha, \beta$  aos respectivos ganhos e  $\text{sgn}(\cdot)$ , à função sinal:

<sup>1</sup>raiz quadrada da soma quadrática dos pesos da rede

$$\text{sgn}(S) = \begin{cases} +1 & \text{se } S > 0 \\ -1 & \text{se } S < 0 \\ 0 & \text{se } S = 0 \end{cases}$$

Podem ser obtidas múltiplas soluções através da especificação de pontos ou trajetórias nesse espaço. Como consequência, o conjunto Pareto pode ser obtido traçando-se trajetórias que convergem para a região de soluções não-existentes. Uma vez que não é possível obter tais soluções, o treinamento estará limitado às soluções da curva Pareto. Outra maneira de obter pontos próximos ao conjunto Pareto-ótimo consiste em definir um objetivo nulo para a função do Erro,  $E_t = 0$ . Uma vez que este ponto seja definido na região de soluções não existentes, o algoritmo irá convergir para uma solução próxima localizada na região factível mas, restringindo a norma ao valor desejado.

A Figura 1.3 exemplifica o comportamento do método. O algoritmo é capaz de alcançar soluções na região factível através da especificação de pontos de trajetórias. A curva Pareto-ótimo é a fronteira entre as soluções factíveis e as soluções não-existentes.

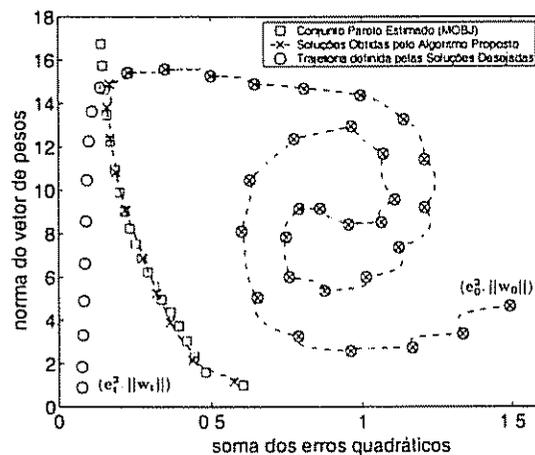


Figura 1.3: Deslocamento da MLP no Plano de Soluções (MOBJ-SMC)

### 1.3 Aplicações do Algoritmo MOBJ-SMC

O método proposto, além de ser capaz de gerar as soluções ou aproximações do conjunto Pareto-ótimo, é capaz de convergir para qualquer solução na região factível partindo de qualquer ponto inicial definido nessa região, o que permite analisar as soluções obtidas. A partir do conjunto Pareto é possível determinar os intervalos de norma que caracterizam soluções com *overfitting* e *underfitting*.

A Figura 1.4 demonstra a capacidade do método de encontrar modelos neurais com alta capacidade de generalização. Mesmo em presença de um conjunto de treinamento com ruído gaussiano na saída, o algoritmo foi capaz de minimizar o erro da rede em

relação à função geradora, obtendo, ao final, uma boa aproximação da função desejada.

Em problemas de classificação, uma vez definido um conjunto de treinamento e validação estatisticamente representativos, o algoritmo proposto é capaz de obter superfícies de separação com alta taxa de generalização, conforme pode ser visualizado na Figura 1.5. A figura também demonstra a solução obtida com o algoritmo *backpropagation* padrão no qual a minimização da função do erro provocou uma perda na capacidade de classificação da rede.

Conforme será apresentado no capítulo 7, os métodos multi-objetivo para treinamento de MLPs (MOBJ, SMC-MOBI) podem ser utilizados para definir topologias reduzidas para um determinado problema sem perda da capacidade de generalização uma vez que o treinamento utilizando esses algoritmos pode ser realizado de forma iterativa, variando-se a topologia da RNA.

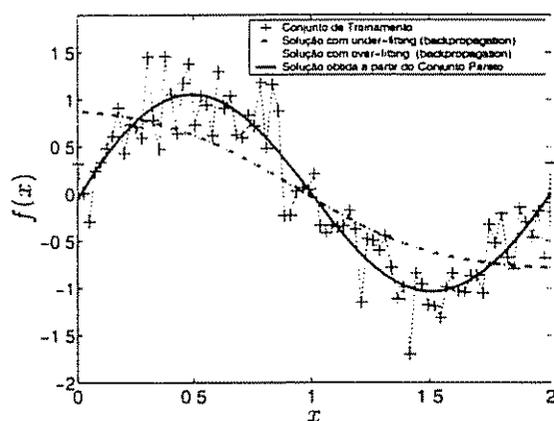


Figura 1.4: Problema de Regressão - solução MOBJ-SMC

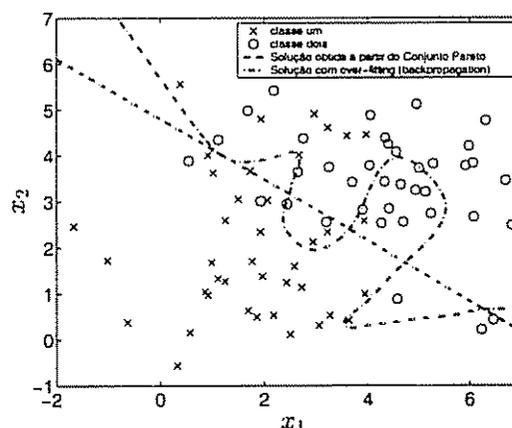


Figura 1.5: Problema de Classificação - solução MOBJ-SMC

## 1.4 Organização do Texto

No Capítulo 2 é realizada uma revisão dos principais conceitos e algoritmos de treinamento de RNAs utilizados na elaboração deste trabalho. São enumerados alguns algoritmos que utilizam variações do método *backpropagation* padrão, como o algoritmo de *Levenberg-Marquardt*, cujas soluções são obtidas através da minimização da função do erro. O algoritmo *Weight-Decay*, que penaliza a função Norma, é apresentado bem como o algoritmo Multi-Objetivo (MOBJ) e o de Modos Deslizantes (SMC - Sliding Mode Control).

No Capítulo 3, o espaço solução Erro  $\times$  Norma é caracterizado. Uma análise do comportamento multi-objetivo de um nodo Peceptron é realizada, a partir do qual são obtidos critérios de inicialização dos pesos do nodo. É apresentada a decomposição dos pesos de uma MLP em coordenadas polares, permitindo a análise e obtenção de soluções de norma fixa. Os resultados são estendidos para uma topologia MLP de duas camadas.

Por fim, é apresentado o comportamento do algoritmo *Backpropagation* padrão no plano Erro  $\times$  Norma.

No Capítulo 4, o algoritmo multi-objetivo com controle por modos deslizantes (SMC-MOBJ) é descrito. A análise segundo a Teoria de Modos Deslizantes para um nodo linear, não-linear e uma rede MLP de duas camadas é realizada a fim de definir os intervalos para os ganhos  $\alpha$  e  $\beta$  que garantem a convergência do método às superfícies de deslizamento.

No capítulo 5, variações do método SMC-MOBJ são propostas para gerar o conjunto de soluções eficientes do qual é extraída a solução com a maior capacidade de generalização. Duas variações para o método são apresentadas: a primeira utilizando o ajuste dos pesos segundo o método de Levenberg-Marquardt mas, restringindo a norma a um determinado valor. A segunda, utilizando a decomposição dos gradientes e o controle da taxa de aprendizado para garantir a restrição da norma.

No Capítulo 6 são apresentados os resultados obtidos através do uso do algoritmo SMC-MOBJ (Sliding Mode Control Multi-Objective). São utilizados dados provenientes de simulações de funções bem como dados obtidos a partir de problemas reais. Os resultados foram agrupados segundo o tipo de problema: classificação, regressão e predição. Resultados utilizando variações do método SMC-MOBJ também são demonstrados.

No Capítulo 7, vários métodos para a escolha de topologias são apresentados. O uso dos algoritmos multi-objetivo para a escolha dessas estruturas é abordado. Técnicas de pruning também são descritas na tentativa de simplificar uma topologia inicialmente obtida por um método multi-objetivo. Os resultados para os métodos propostos são demonstrados.

No Capítulo 8 são descritas as conclusões gerais bem como as propostas para a continuidade do trabalho.

# Capítulo 2

## Redes Neurais Artificiais

Neste capítulo serão descritos os algoritmos regularmente utilizados para o treinamento de RNAs do tipo Multi-Layer Perceptron (MLP). Partindo da análise de um simples nodo perceptron, o algoritmo *backpropagation* será descrito para uma topologia MLP de duas camadas. Posteriormente, serão descritas algumas variações desse método como o algoritmo de *Levenberg-Marquardt*. Os algoritmos Multi-Objetivo (MOBJ) e Modos Deslizantes (SMC) também fazem parte do escopo deste capítulo, no qual serão demonstradas suas principais características.

### 2.1 Introdução

O passo fundamental para o surgimento das RNAs foi o modelo matemático proposto por McCulloch e Pitts, em 1943 [McCulloch and Pitts, 1943]. O modelo, denominado neurônio MCP (McCulloch e Pitts), é descrito por um conjunto de  $n$  entradas ao qual cada entrada é multiplicada por um determinado peso e, em seguida, os resultados são somados e comparados a um limiar. Esse modelo é amplamente citado na literatura da área de RNAs [Braga et al., 2000].

Uma vez modelada a estrutura do neurônio, é necessário definir uma metodologia para o treinamento ou a aprendizagem do nodo. Tal conceito foi introduzido em 1958 através do trabalho de Frank Rosenblatt [Rosenblatt, 1958]. O *perceptron*, estrutura proposta por Rosenblatt e constituída por nodos MCP, era arranjado em forma de uma rede composta de duas camadas. A primeira camada era constituída por pesos definidos aleatoriamente e constantes em todo o processo de treinamento. O aprendizado era restrito ao ajuste dos pesos na segunda camada. A rede proposta por Rosenblatt possibilitou um aumento de trabalhos e de estudos nessa área até que em 1969, um trabalho publicado por Minsky e Papert [Minsky and Papert, 1969], levou a um desinteresse na área em função das críticas sobre a capacidade computacional do perceptron.

Foi a partir de 1982 com a publicação do trabalho de Hopfield [Hopfield, 1982]

e posteriormente em 1986 com a formulação do algoritmo *backpropagation* [Rumelhart et al., 1986], que as pesquisas em RNAs vieram, novamente, a intensificar-se.

A partir da formulação do algoritmo *backpropagation* padrão e em virtude do método gradiente descendente apresentar um convergência lenta, foram desenvolvidas variações desse método com o objetivo de acelerar o processo de minimização da função de erro da rede e impedir a parada do processo de treinamento em regiões de gradientes nulos conhecidas como *platôs*. Como consequência, surgiram os métodos: Momentum [Rumelhart et al., 1986], QuickProp [Fahlman, 1988], RProp [Riedmiller and Braun, 1993] e variações. Alguns algoritmos implementam técnicas para o ajuste da taxa de aprendizado [Silva and Almeida, 1990, Tollenaere, 1990]. Posteriormente, foram utilizadas técnicas avançadas de otimização e aproximações quadráticas da superfície do erro, ou seja, uso de derivadas de segunda ordem, para implementar o ajuste dos pesos. Dentre esses algoritmos, o mais utilizado é conhecido como o algoritmo de *Levenberg-Marquardt* [Hagan and Menhaj, 1994a].

Apesar de os métodos desenvolvidos serem capazes de acelerar a convergência da rede, não podiam evitar regiões de mínimos locais. Recentemente, foi desenvolvida uma técnica de treinamento de RNAs utilizando métodos de controle por Modos Deslizantes [Parma et al., 1998] que *controla* a convergência da rede para o ponto de mínimo erro e derivada. Por outro lado, alcançar uma região de mínimo global, ou seja, de erro mínimo, implica no super-ajuste da rede ao conjunto de treinamento provocando um efeito denominado *overfitting* que impede a rede de obter respostas coerentes a um conjunto de dados não conhecidos a priori. Essa característica é denominada capacidade de generalização. Várias técnicas foram desenvolvidas para auxiliar na escolha de modelos neurais como o *VC-dimension* [Vapnik, 1995], a medida do número de derivadas contínuas ou número de parâmetros livres [Girosi et al., 1995], a magnitude da transformada de Fourier da resposta da rede [Barron, 1993]. Mas, algumas medidas são inviáveis como o *VC-dimension* ou de difícil interpretação (Transformada de Fourier). Normalmente, pode-se medir a capacidade de generalização do modelo através da medida da função do erro de um conjunto de validação.

Uma vez definido o problema do dimensionamento da rede, que é conhecido na literatura como o dilema entre a polarização e a variância [Geman et al., 1992], diversos algoritmos foram desenvolvidos na tentativa de modificar a estrutura original da rede alterando a sua capacidade de generalização. Entretanto, uma rede superdimensionada pode apresentar uma boa capacidade de generalização a partir do correto dimensionamento dos seus parâmetros de ajuste ou pesos [Bartlett, 1997].

O método Multi-Objetivo [Teixeira et al., 2000] realiza o equilíbrio entre a norma dos pesos e o erro do treinamento de redes MLPs garantindo assim a capacidade de generalização do modelo. Uma vez definida uma topologia superdimensionada, o algoritmo gera um

conjunto de soluções com normas variadas e erro minimizado para cada valor de norma, selecionando o modelo com a melhor resposta em relação ao conjunto de validação.

## 2.2 O Treinamento de RNAs

O treinamento é definido como o ajuste dos parâmetros de uma RNA com o objetivo de aproximar uma função estabelecida pelo conjunto de dados apresentado. Durante o treinamento, deve-se estabelecer um critério para a interrupção do mesmo. Normalmente, o processo de treinamento é interrompido quando:

1. Número máximo de iterações (passos ou epochs) é alcançado;
2. Limite de erro: o erro de treinamento ou validação torna-se menor do que um limiar especificado.

Em relação ao conjunto de dados, normalmente este é subdividido em dados para treinamento e dados para validação. O critério de escolha das dimensões dos conjuntos é empírico. Como padrão, pode-se utilizar 60% do conjunto para treinamento e o restante para a validação da RNA, sendo que o conjunto de validação pode ser incorporado ao treinamento através do uso de técnicas como *cross-validation* [Stone, 1974], *early-stopping* [Haykin, 1999].

Quanto ao treinamento, este pode ser realizado de duas maneiras:

1. Treinamento em lote (*batch*): a cada passo, os parâmetros de treinamento são ajustados uma única vez. Tal ajuste é realizado após a apresentação de todo o conjunto de treinamento;
2. Treinamento por padrão: o ajuste dos parâmetros da rede é realizado para cada par entrada-saída (padrão) do conjunto de treinamento de tal modo que, em um passo (epoch), são realizados tantos ajustes quanto o número de padrões de treinamento.

O processo de aprendizado é realizado através de sucessivos ajustes no vetor de pesos da RNA. Como ponto de partida para o treinamento, é necessário que a cada peso seja atribuído um determinado valor aleatório. Esse processo é denominado *inicialização*.

Após a inicialização do vetor de pesos,  $\mathbf{w}$ , o algoritmo de treinamento irá calcular o vetor de ajuste,  $\Delta\mathbf{w}$ , que será somado ao valor atual dos pesos. A equação de ajuste é descrita como:

$$\mathbf{w}_{(k+1)} = \mathbf{w}_{(k)} + \Delta\mathbf{w} \quad (2.1)$$

## 2.3 Método do Gradiente Descendente

O Método do Gradiente Descendente realiza os ajustes dos pesos no sentido contrário do vetor gradiente da função de custo, de forma a minimizá-la. Definindo um conjunto de treinamento formado pelos pares  $\{(x^i, d^i)\}_{i=1}^P$  onde  $x$  é o vetor de entrada e  $d$  é a saída desejada, a função de custo a ser minimizada é normalmente definida como a soma dos erros quadráticos:

$$J = \frac{1}{2} \sum_{i=1}^P (d^i - y^i)^2 \quad (2.2)$$

onde  $y^i$  é a saída da RNA.

A partir de uma condição inicial para os valores dos pesos, o ajuste dos mesmos é definido por:  $\Delta w \propto -\nabla J$ , onde  $\nabla J$  é o gradiente da função de custo.

Como o ajuste dos pesos depende do cálculo do gradiente, a função objetivo deve apresentar as seguintes características:

1. Deve ser diferenciável;
2. Deve ser contínua,

### 2.3.1 ADALINE - ADAPtive LInear NEuron

O modelo do neurônio proposto por Bernard Widrow e Marcian Hoff em 1960 [Widrow and Hoff, 1960], foi apresentado quase simultaneamente ao perceptron. Apesar de topologicamente idênticos, o ajuste dos pesos do nodo ADALINE utiliza o gradiente descendente da função de custo da saída linear do nodo, ou seja, antes da aplicação da função de ativação não-linear.

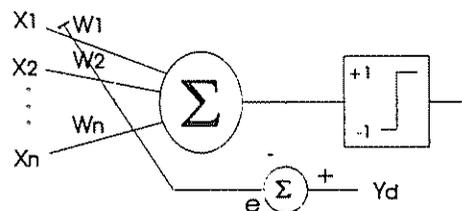


Figura 2.1: Modelo de um Nodo ADALINE

Definindo-se a função de custo através da Equação 2.2, e a constante de proporcionalidade  $\eta$ , a equação de ajuste dos pesos do nodo adaline é dada por:

$$\Delta w_i = -\eta \cdot \nabla J \quad (2.3)$$

onde  $\nabla J = \frac{\partial J}{\partial w_i}$ . Utilizando a regra da cadeia tem-se:  $\frac{\partial J}{\partial w_i} = \frac{\partial J}{\partial y} \cdot \frac{\partial y}{\partial w_i}$ , desenvolvendo-se as derivadas,  $\frac{\partial J}{\partial w_i} = -x_i(d - (w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n))$  e definido o erro,  $e_{(k)}$ , como a diferença entre a saída desejada e a saída do nodo, a expressão final para o ajuste dos pesos é definida por:

$$\mathbf{w}_{(k+1)} = \mathbf{w}_{(k)} + \eta e_{(k)} \mathbf{x}_{(k)} \quad (2.4)$$

### 2.3.2 Regra Delta

Após o surgimento da regra de treinamento do nodo ADALINE, também conhecida como *Regra de de Widrow-Hoff*, a aplicação do método do gradiente descendente para ajuste dos pesos, baseado na saída da função de ativação, permitiu o surgimento de um novo conjunto de modelos e, conseqüentemente, uma nova regra de treinamento conhecida como *Regra Delta*.

Para que a regra delta possa ser aplicada à saída de uma função de ativação, esta deve ser diferenciável. Como a função de ativação é aplicada na saída linear, a sua representação é da forma:  $y = f(h)$ ,  $h = \mathbf{w} \cdot \mathbf{x}^T$ .

Seja um nodo com entrada  $\mathbf{x}$  e o vetor de pesos  $\mathbf{w}$ , ambos de dimensão  $m$ . A saída do nodo é definida por:

$$y_{(k)} = f(h_{(k)}), \text{ onde } h_{(k)} = \sum_{l=1}^m (w_{l(k)} \cdot x_{l(k)})$$

Segundo a regra da cadeia, o gradiente da saída em relação aos pesos é da forma:

$$\frac{\partial y_{(k)}}{\partial w} = \frac{\partial f(h_{(k)})}{\partial h_{(k)}} \frac{\partial h_{(k)}}{\partial w}$$

portanto,

$$\frac{y_{(k)}}{\partial w} = f'(h_{(k)}) \cdot x_{(k)}$$

Aplicando o método para o cálculo do gradiente da função de custo, o gradiente do erro para  $J_{(k)} = \frac{1}{2} (d_{(k)} - y_{(k)})^2$ , pode ser definido por:

$$e = d_{(k)} - y_{(k)}, \text{ o que leva à seguinte função custo: } J_{(k)} = \frac{1}{2} e_{(k)}^2$$

$$\nabla J = \frac{\partial J}{\partial e} \frac{\partial e}{\partial y_{(k)}} \frac{\partial y_{(k)}}{\partial w_{(k)}}$$

$$\nabla J = -1 \cdot e \cdot f'(h_{(k)}) \cdot \mathbf{x}_{(k)}$$

Como o ajuste é realizado no sentido contrário à direção do gradiente e acrescentando-se, novamente, a taxa de aprendizado,  $\eta$ , a regra de atualização dos pesos para nodos com

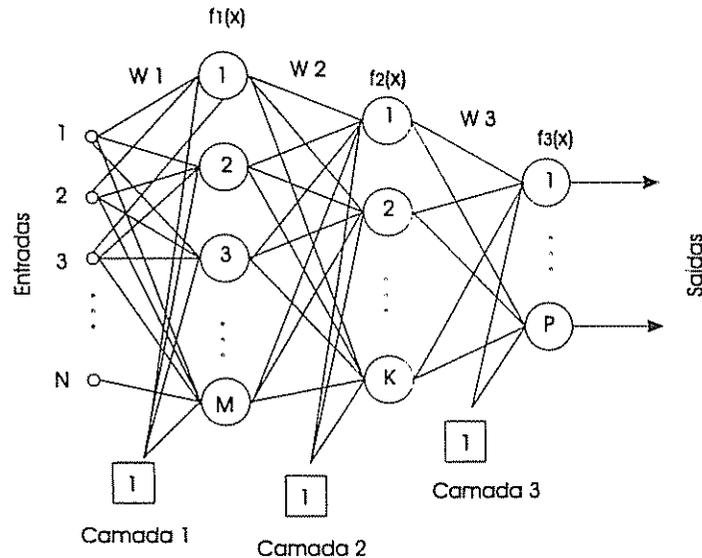


Figura 2.2: Exemplo de uma rede MLP de três camadas

funções de ativação diferenciáveis pode ser assim escrita:

$$\mathbf{w}_{(k+1)} = \mathbf{w}_{(k)} + \eta \cdot e_{(k)} \cdot f'(h_{(k)}) \cdot \mathbf{x}_{(k)} \quad (2.5)$$

Para a função de ativação linear, a sua derivada será unitária ( $f'(h_{(k)}) = 1$ ), reduzindo a Regra Delta à Regra de *Widrow-Hoff*.

## 2.4 Redes *Multi-Layer Perceptron* (MLP)

Uma vez definida a estrutura básica de uma RNA, o perceptron e a sua função de ativação, uma rede neural do tipo MLP corresponde ao arranjo desses elementos segundo uma ordem ou topologia. Basicamente, uma rede é subdividida em camadas: camada de entrada, camadas intermediárias ou escondidas e camada de saída. Um exemplo desse arranjo é expresso na Figura 2.2. A topologia mais simples de uma RNA apresenta uma única camada. As redes MLPs são uma topologia específica na área de RNAs. Neste trabalho, as demais topologias não serão abordadas, entretanto, os resultados obtidos não se restringem às MLPs.

O funcionamento de uma rede MLP é síncrono, ou seja, dada uma entrada (um vetor) esta é propagada para a saída multiplicado-se pelos pesos de cada camada, aplicando-se a função de ativação e propagando-se esse valor para a camada seguinte até que a camada de saída seja atingida. Por ser constituída por um conjunto de nodos MCPs, pode-se definir funções de ativação diferenciadas para cada nodo da MLP. Geralmente definem-se funções de ativação iguais em uma mesma camada. É necessário lembrar que uma rede que apresenta somente funções de ativação lineares nas camadas intermediárias é equivalente

a uma rede de uma única camada. A topologia da MLP mais utilizada apresenta uma camada escondida e é capaz de implementar qualquer função contínua [Cybenko, 1989].

Os algoritmos utilizados para o treinamento de MLPs derivam-se do método do gradiente descendente. Uma vez que as funções de ativação definidas para cada camada são diferenciáveis e contínuas, a saída da rede pode ser descrita através de uma equação, conseqüentemente, é possível calcular o gradiente da função custo em relação aos parâmetros de ajuste da RNA (os pesos).

### 2.4.1 Projeto de RNAs

Muitos são os fatores que influenciam na escolha da topologia, número de camadas e quantidade de nodos em cada camada:

- número de pares (entrada-saída) do conjunto de treinamento;
- existência de ruído no conjunto de treinamento;
- complexidade da função a ser modelada pela MLP;
- distribuição estatística dos dados de treinamento.

Não existe, a priori, uma metodologia específica para a escolha dos parâmetros da rede. Geralmente, o dimensionamento é definido empiricamente baseado na quantidade de entradas e saídas, evitando-se redes com topologias muito grandes que têm um grande custo computacional. O dimensionamento inadequado da rede pode levar às seguintes situações:

- *overfitting*: Para um conjunto de treinamento experimental, o ruído apresentado pode ser *aprendido* pela rede, o que provoca uma resposta imprecisa. Outro fato característico é a perfeita adaptação da rede ao conjunto de treinamento, ou seja, a rede não é capaz de extrapolar para o conjunto de validação.
- *underfitting*: Nesse caso a topologia escolhida é insuficiente para a modelagem do problema, a rede não converge durante o treinamento.

Existem técnicas na literatura [Reed, 1993] que procuram eliminar pesos e nodos que possuem pouca ou nenhuma influência na resposta da rede evitando, dessa forma, o *overfitting* e o superdimensionamento da rede. Dentre as técnicas utilizadas, algumas envolvem mudanças na função custo com o objetivo de restringir, durante o treinamento, a norma dos pesos da rede resultando em soluções com baixos valores de normas.

## 2.4.2 O Algoritmo BackPropagation

Existem vários algoritmos para o treinamento de MLPs. Dentre esses, o mais conhecido é o *BackPropagation* formulado em 1986 [Rumelhart et al., 1986] e que utiliza o método do gradiente descendente na formulação do ajuste dos pesos:

$$\mathbf{w}_{(k+1)} = \mathbf{w}_{(k)} - \eta \cdot \nabla J_{(k)} \quad (2.6)$$

A função de custo,  $J_{(k)}$  é definida pela Equação 2.2, sendo a saída da rede,  $y$ , uma função dos pesos e das funções de ativação. O ajuste dos pesos é realizado em duas etapas: propagando-se a entrada da rede para a sua saída (*forward*), calculando-se o erro e propagando-o novamente para a entrada (*backward*). Para uma rede de duas camadas, a atualização dos pesos é feita da seguinte maneira:

Para os nodos de saída:

$$\begin{aligned} w_{ji(k+1)} &= w_{ji(k)} + \eta (d_j(k) - y_j(k)) f'(net_j(k)) x_{Hi(k)} \\ w_{ji(k+1)} &= w_{ji(k)} + \eta \cdot \delta_j(k) \cdot x_{Hi(k)} \end{aligned} \quad (2.7)$$

onde  $net_j$  é a saída linear do nodo  $j$  da camada de saída e  $x_H$  é a saída dos nodos da camada escondida após a aplicação da função de ativação.

Para os nodos da camada escondida:

$$w_{ji(k+1)}^h = w_{ji(k)}^h + \eta \cdot x_{(k)}^h \cdot f'_H(r_i(k)) \cdot \sum_{j=1}^p [\delta_j(k) \cdot w_{ji(k)}] \quad (2.8)$$

onde  $w^h$  são os pesos da camada escondida,  $x^h$  é a entrada da RNA,  $f'_H$  é a derivada da função de ativação da camada escondida e  $r_i$  é a saída linear do neurônio  $i$  da camada escondida. Para descrever a relação entre a entrada da rede  $x^h$  e a sua saída, a equação na forma matricial do funcionamento da rede MLP de duas camadas é da forma:

$$y = f(w \cdot (f_H(u^h \cdot x^h))) \quad (2.9)$$

Uma descrição detalhada das equações de ajuste dos pesos para cada camada é amplamente encontrada na literatura [Braga et al., 2000].

As desvantagens do algoritmo backpropagation consistem na sua baixa velocidade de convergência e na sua limitação de busca das soluções, pois se trata de um algoritmo de busca local. Tal fato é observado na própria formulação do algoritmo. Por se tratar de um algoritmo que depende somente do gradiente local, caso haja um mínimo local próximo<sup>1</sup>, o algoritmo ficará *preso*. Uma das maneiras de melhorar a resposta de uma rede utilizando esse algoritmo, consiste no treinamento de múltiplas redes com pesos inicializados aleatoriamente. Nesse caso, a probabilidade de uma rede convergir para uma

<sup>1</sup> região de gradiente nulo

região próxima ao mínimo global é maior. Entretanto, tal procedimento apresenta um elevado custo computacional e não é capaz de garantir uma boa solução.

Vários métodos, desenvolvidos a partir do *backpropagation* para evitar a convergência da rede para mínimos locais, podem ser encontrados na literatura. O *Momentum* [Rumelhart et al., 1986] procura aumentar a velocidade de convergência através de um termo de ajuste resultante da multiplicação dos pesos anteriores por um ganho ( $\mu$ ).

$$w_{ji(k+1)} = w_{ji(k)} + \eta \sigma_{j(t)} x_{i(t)} + \mu \cdot w_{ji(k-1)} \quad (2.10)$$

Entretanto, o aumento da velocidade de convergência não garante a convergência para o mínimo global e a especificação do valor do parâmetro  $\mu$  é empírica.

O *RPROP* [Riedmiller and Braun, 1993] utiliza o sinal do gradiente e não o seu valor para realizar a correção dos pesos ao passo que o método *QUICKPROP* [Fahlman, 1988] aproxima a superfície do erro por uma parábola em função dos pesos. O ajuste dos pesos é realizado de forma que, em uma única iteração, o erro mínimo (da parábola) seja alcançado. Entretanto, nem sempre a superfície do erro pode ser modelada por uma parábola. Os métodos de *Taxa Adaptativa* [Silva and Almeida, 1990, Tollenaere, 1990] utilizam técnicas para o ajuste da taxa de aprendizado,  $\eta$ , utilizada no *backpropagation*. Contudo, nenhum desses métodos é capaz de garantir uma convergência ao mínimo global.

### 2.4.3 O Algoritmo de Levenberg-Marquardt

O algoritmo desenvolvido por Levenberg-Marquardt [Marquardt, 1963] e aplicado ao treinamento de redes neurais [Hagan and Menhaj, 1994a] é um exemplo de algoritmo que visa minimizar a função de custo, buscando o mínimo global, utilizando técnicas de otimização mais complexas. Esses algoritmos apresentam um custo computacional mais elevado, entretanto, apresentam um ganho significativo em termos de número de iterações. Em virtude da complexidade dos cálculos envolvidos, um aumento na dimensão da MLP ou do número de padrões de treinamento provoca um aumento no tempo de convergência podendo causar a perda da eficiência do algoritmo.

Utilizando uma aproximação do Método de Newton, a função de custo é definida em função de um vetor de parâmetros  $x$ :

$$V_{(x)} = \sum_{i=1}^N e_i^2(x) \quad (2.11)$$

Para o treinamento de MLPs o vetor  $x$  é composto pelos pesos da rede arranjados na forma vetorial.

O ajuste do vetor de parâmetros utilizando o Método de Newton é descrito por:

$$x^{k+1} = x^k - \lambda H_{(x^k)}^{-1} \nabla f(x^k) \quad (2.12)$$

onde  $H_{(x^k)}$  é a matriz Hessiana,  $\nabla f(x^k)$  é o vetor gradiente no ponto  $x^k$  e  $\lambda$  define o passo do algoritmo.

Semelhantemente, o ajuste do vetor de pesos é definido da seguinte maneira:

$$\Delta x = - [\nabla^2 V_{(x)}]^{-1} \cdot \nabla V_{(x)} \quad (2.13)$$

onde  $\nabla^2 V_{(x)}$  é a matriz Hessiana de  $V_{(x)}$  e  $\nabla V_{(x)}$  é o gradiente da função  $V_{(x)}$ .

Utilizando o Método de Gauss-Newton  $\sum_{i=1}^N [e_{i(x)} \cdot \nabla^2 e_{i(x)}] \approx 0$ , obtém-se as seguintes aproximações:

$$\begin{aligned} \nabla V_{(x)} &= J_{(x)}^T \cdot e_{(x)} \\ \nabla^2 V_{(x)} &= J_{(x)}^T \cdot J_{(x)} \end{aligned} \quad (2.14)$$

onde  $J_{(x)}$  é a matriz jacobiana. O ajuste de  $x$  é obtido através de uma modificação no método de Gauss-Newton:

$$\Delta x = - [J_{(x)}^T \cdot J_{(x)} + \mu \cdot I]^{-1} \cdot J_{(x)}^T \cdot e_{(x)} \quad (2.15)$$

De acordo com a Equação 2.15, para valores de  $\mu$  elevados o método descrito é o gradiente descendente, caso contrário, o método é o Gauss Newton. O algoritmo segue um padrão para o ajuste do parâmetro  $\mu$ : que é multiplicado por um fator  $\beta$  quando o ajuste dos pesos provoca um aumento na função de erro e dividido caso contrário.

Outros tipos de algoritmos utilizam métodos de otimização para a minimização da função custo [Kramer and S.-Vincentelli, 1989, Battiti and Massuli, 1990, Hagan and Menhaj, 1994b]. Dentre eles, o algoritmo de Levenberg Marquardt é o mais conhecido.

#### 2.4.4 O Algoritmo Weight Decay

O algoritmo Weight Decay [Hinton and Nowlan, 1987] utiliza um termo de regularização demonstrado na Equação 2.2, com o objetivo de minimizar a norma dos pesos em conjunto com a minimização da soma dos erros quadráticos.

$$J = \frac{1}{2} \sum_{i=1}^k (d_i - y_i)^2 + \frac{1}{2} \lambda \|\mathbf{w}\|^2 \quad (2.16)$$

Tal formulação necessita do dimensionamento, por parte do projetista, de um novo parâmetro escalar,  $\lambda$ , denominado parâmetro de regularização. Caso o valor de  $\lambda$  seja muito grande, a solução encontrada tende a  $\|\mathbf{w}\| = 0$  e, para valores muito pequenos, a Equação 2.16 minimiza somente a parcela do erro.

A equação do ajuste dos pesos para a Equação 2.16 é descrita a seguir:

$$J = f_1 + f_2, \text{ onde } f_1 = \sum_{i=1}^k (d_i - y_i)^2 \text{ e } f_2 = \frac{1}{2} \lambda \|\mathbf{w}\|^2$$

$$\Delta w_{ji} \propto -\frac{\partial J}{\partial w_{ji}}$$

$$\frac{\partial J}{\partial w_{ji}} = \frac{\partial f_1}{\partial w_{ji}} + \frac{\partial f_2}{\partial w_{ji}} \quad (2.17)$$

A solução do primeiro termo da Equação 2.17,  $\left(\frac{\partial f_1}{\partial w_{ji}}\right)$ , é idêntica à solução do algoritmo backpropagation expressa nas Equações 2.7 e 2.8. Já a derivada do segundo termo é obtida a seguir:

$$\frac{\partial f_2}{\partial w_{ji}} = \frac{\lambda}{2} \cdot \frac{\partial \|\mathbf{w}\|^2}{\partial w_{ji}}, \text{ onde } \|\mathbf{w}\| = \sqrt{\sum_{ji} (w_{ji})^2} = \left(\sum_{ji} (w_{ji})^2\right)^{1/2} \quad (2.18)$$

$$\frac{\partial \|\mathbf{w}\|^2}{\partial w_{ji}} = 2 \cdot \|\mathbf{w}\| \cdot \frac{\partial \|\mathbf{w}\|}{\partial w_{ji}} \quad (2.19)$$

$$\frac{\partial \|\mathbf{w}\|}{\partial w_{ji}} = \frac{1}{2} \left(\sum_{ji} (w_{ji})^2\right)^{-1/2} \cdot 2 \cdot w_{ji} = \frac{w_{ji}}{\|\mathbf{w}\|} \quad (2.20)$$

Expressando  $\frac{\partial f_2}{\partial w_{ji}}$  em função dos resultados obtidos em 2.19 e 2.20:

$$\frac{\partial f_2}{\partial w_{ji}} = \frac{\lambda}{2} \cdot 2 \cdot \|\mathbf{w}\| \cdot \frac{w_{ji}}{\|\mathbf{w}\|}$$

$$\frac{\partial f_2}{\partial w_{ji}} = \lambda \cdot w_{ji} \quad (2.21)$$

A equação geral para o ajuste dos pesos para o algoritmo weight decay é definida por:

$$\Delta w_{ji} = -\eta \frac{\partial f_1}{\partial w_{ji}} - \lambda \cdot w_{ji}$$

$$w_{ji(k+1)} = w_{ji(k)} - \eta \frac{\partial f_1}{\partial w_{ji}} - \lambda \cdot w_{ji} \quad (2.22)$$

### 2.4.5 O Algoritmo de Modos Deslizantes

Grande parte dos algoritmos apresentados limitam-se ao uso em treinamento *off-line*. O algoritmo proposto por Parma, Menezes e Braga [Parma et al., 1999], permite o seu uso em treinamento *off-line* e *on-line* porque a teoria envolvida em sua formulação permite a sua configuração para ambos os casos. O algoritmo é formulado segundo a teoria de controle por modos deslizantes (SMC - Sliding Mode Control) [Utkin, 1978], que agrega velocidade de treinamento e robustez. Mesmo sendo um algoritmo que utiliza o gradiente da função de custo no ponto (derivado do backpropagation), ele apresenta um melhor

desempenho. Outra vantagem desse algoritmo é a teoria envolvida na obtenção analítica dos parâmetros de treinamento. Em virtude da teoria de controle utilizada, é possível configurar e definir os limites para os parâmetros de aprendizado da rede de forma a garantir a convergência da rede.

Existem exemplos do uso de técnicas de controle por modos deslizantes (SMC) para treinamento de RNAs em casos específicos, como nodos perceptron [Sira-Ramírez and Colina-Morles, 1995, Sira-Ramírez et al., 2000]. A vantagem do algoritmo *Modos Deslizantes (SMC)* consiste na adaptabilidade da teoria envolvida para qualquer configuração ou topologia de RNA do tipo MLP. Para exemplificar o uso do algoritmo foi escolhida a topologia MLP com uma camada escondida. Para descrever melhor as equações, a seguinte nomenclatura foi definida para a rede:

- $T$  : vetor de entrada da rede acrescido do termo de bias;
- $Y_H$  : vetor de saída da camada escondida;
- $Y$  : vetor de saída da rede (MLP);
- $Y_{dj}$  : saída desejada para o nodo  $j$  da camada de saída;
- $W^H$  : matriz de pesos das conexões entre a entrada e a camada escondida;
- $W_{ih}^H$  : peso entre o nodo de entrada  $i$  e o nodo escondido  $j$ ;
- $W$  : matriz de pesos das conexões entre a camada escondida e a de saída;
- $W_{ji}$  : peso entre o nodo escondido  $i$  e o nodo de saída  $j$ ;
- $f_{H(\cdot)}$  : função de ativação da camada escondida;
- $f(\cdot)$  : função de ativação da camada de saída;
- $R$  : vetor da saída linear da camada escondida ( $Y_H = f_{H(R)}$ ,  $R = W^H \cdot T$ );
- $V$  : vetor da saída linear da camada de saída ( $Y = f(V)$ ,  $V = W \cdot Y_H$ ).

As equações para o treinamento *on-line* da rede MLP são subdivididas em duas etapas:

#### 1. Definição das equações das superfícies de deslizamento.

Para a camada de saída:

$$\begin{aligned} S_j &= X_{2j} + C \cdot X_{1j}, \quad C > 0 \\ X_{1j} &= Y_{dj} - Y_j \\ X_{2j} &= \frac{\partial X_{1j}}{\partial t} \end{aligned} \tag{2.23}$$

Para a camada escondida:

$$\begin{aligned} S_H &= X_{2H} + C_H \cdot X_{1H}, \quad C_H > 0 \\ X_{1H} &= 0.5 \sum_{k=1}^p (Y_{dk} - Y_k)^2 \\ X_{2H} &= \frac{\partial X_{1H}}{\partial t} \end{aligned} \quad (2.24)$$

2. Definição dos ajustes dos pesos e dos parâmetros de modo a satisfazer as condições de convergência.

$$\dot{W}_{ji} = \frac{\alpha \cdot \text{sign}(S_j) \cdot |X_{1j}|}{\frac{\partial f(V_j)}{\partial V_j}} \cdot Y_{Hi} \quad (2.25)$$

$$\dot{W}_{ih}^H = \frac{\beta \cdot \text{sign}(S_H) \cdot |X_{1H}|}{\left[ \sum_{j=1}^p \left( (Y_{dj} - Y_j) \cdot \frac{\partial f(V_j)}{\partial V} \cdot W_{ji} \right) \right] \cdot \frac{\partial J_H(R_i)}{\partial R}} \cdot T_h \quad (2.26)$$

Para as condições de convergência em regime de modos deslizantes [Utkin, 1978], temos os seguintes limites para os ganhos  $C$  e  $C_H$ :

$$C \geq \max \left\{ -\frac{2}{Y_{Hi}} \cdot \frac{\partial Y_{Hi}}{\partial t} - \frac{|X_{2j}|}{X_{1j}} \right\} \quad (2.27)$$

$$C_H \geq \max \left\{ -\frac{2}{T_H} \cdot \frac{\partial T_H}{\partial t} - \frac{|X_{2H}|}{X_{1H}} \right\} \quad (2.28)$$

O algoritmo de Modos Deslizantes, partindo de um estado inicial de erro e derivadas, controla o estado da rede de forma que as superfícies de deslizamento, identificadas pelas Equações 2.23 e 2.24, sejam alcançadas ou inicialmente atravessadas. A partir dessa situação ocorre o que é denominado regime de Modos Deslizantes no qual, pelas condições de convergência (Equações 2.27 e 2.28), é garantida a convergência do estado da rede à origem do espaço de estados, representado pelo erro e derivadas nulas.

Os chamados algoritmos híbridos, como o SMC, permitem incorporar ao treinamento de RNAs características de convergência e comportamento de outros métodos. O uso da técnica de controle por modos deslizantes possibilita, além da otimização do tempo de convergência do treinamento, uma formulação matemática e precisa capaz de garantir a minimização da função de custo.

### 2.4.6 O Algoritmo Multi-Objetivo

A maioria dos algoritmos para treinamento de MLPs procura minimizar a soma quadrática dos erros para o conjunto de treinamento (Equação 2.2). Entretanto, em dados reais, o erro *nulo* alcançado no treinamento representa um modelamento do ruído por parte da rede. Outro fator limitante é a dimensão da rede que pode impedir a convergência a

um resultado satisfatório. Dessa forma, o treinamento pode resultar em *overfitting* ou *underfitting*. Na tentativa de solucionar tais problemas, os algoritmos de *pruning* surgem como *cirurgias* com o objetivo de simplificar a rede e melhorar a sua resposta. Sob esse aspecto, é evidente que existe um equilíbrio entre a dimensão da rede (flexibilidade) e a resposta desejada (erro).

O algoritmo Multi-Objetivo [Teixeira et al., 2000] apresenta o treinamento de MLPs sob uma nova perspectiva: o plano bi-dimensional definido pelo erro e norma e o conjunto Pareto-ótimo. Outro algoritmo desenvolvido recentemente, o SVM (*Support Vector Machine*) [Cortes and Vapnik, 1995] também procura obter redes através da minimização da norma e do erro<sup>2</sup>.

O primeiro passo do algoritmo multi-objetivo (MOBJ) é o levantamento do conjunto de soluções ótimas que se localizam no limiar entre as soluções factíveis e as soluções não-existentes definida no espaço de estados representando pelas funções do erro e da norma. A partir do levantamento desse conjunto, um decisor escolhe a solução de melhor resposta, ou menor erro de validação. Para gerar o conjunto Pareto-ótimo, uma MLP superdimensionada é gerada. o algoritmo define, inicialmente, a seguinte formulação:

- $f_1^* \in \mathbb{R}^2$ : vetor que representa a solução de *underfitting*: norma baixa e erro elevado;
- $f_2^* \in \mathbb{R}^2$ : vetor que representa a solução de *overfitting*: norma elevada e erro baixo;
- $f^* \in \mathbb{R}^2$ : vetor de solução *utópica* (não-existente): contém a norma de  $f_1^*$  e o erro de  $f_2^*$ ;
- $v_k$ : vetor construído pela combinação de  $f_1^*$  e  $f_2^*$

$$v_k = f^* + \gamma_k(f_1^* - f^*) + (1 - \gamma_k)(f_2^* - f^*), \text{ para } 0 < \gamma_k < 1 \quad (2.29)$$

onde  $k$  é o número de pontos a serem levantados para formar o Conjunto Pareto-ótimo.

O problema multi-objetivo é então redefinido como um problema mono-objetivo da seguinte maneira:

$$\mathbf{w}^* = \underset{\mathbf{w}, \eta}{\operatorname{arg\,min}} \eta \quad (2.30)$$

sujeito a:

$$f_i(\mathbf{w}) \leq f^* + \eta v_k \quad (2.31)$$

<sup>2</sup>entende-se *erro* por função de custo - soma dos erros quadráticos

com  $f_1(\mathbf{w})$  igual a soma dos erros quadráticos  $\left(\frac{1}{2} \sum_{j=1}^n (d_j - y_j)^2\right)$ ,  $f_2(\mathbf{w})$  definido como a norma dos pesos ( $f_2(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2$ ) e  $\eta$  é uma variável auxiliar.

Reescrevendo o problema mono-objetivo em função das restrições, tem-se:

$$\mathbf{w}^* = \underset{\mathbf{w}, \eta}{\operatorname{arg\,min}} \eta \quad (2.32)$$

sujeito a:

$$\begin{cases} g_1(\mathbf{w}, \eta) = \frac{1}{2} \sum_{j=1}^n (d_j - y_j)^2 - f_1^* - \eta v_k \leq 0 \\ g_2(\mathbf{w}, \eta) = \frac{1}{2} \|\mathbf{w}\|^2 - f_2^* - \eta v_k \leq 0 \end{cases} \quad (2.33)$$

O algoritmo Multi-Objetivo [Teixeira et al., 2000] utiliza o método de otimização Elipsoidal [Shor, 1977] para encontrar a solução do problema mono-objetivo (Equações 2.32 e 2.33).

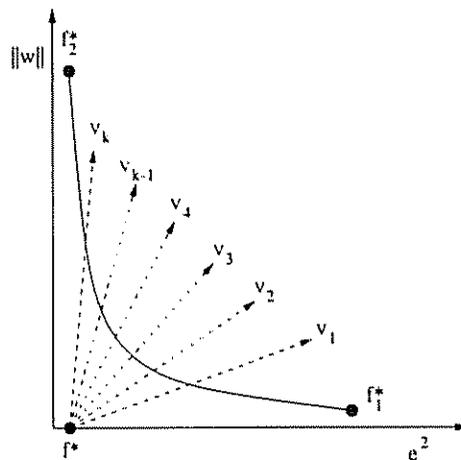


Figura 2.3: Curva Pareto e vetores  $v_k$  definidos no plano  $\|\mathbf{w}\| \times e^2$

O algoritmo descrito acima utiliza um método de otimização multi-objetivo denominado *Algoritmo de Relaxação* [Takahashi et al., 1997]. Esse método apresenta um custo computacional elevado pois são realizadas  $k$ -operações de otimização, uma para cada ponto a ser levantado do Pareto-ótimo. Um segundo método de otimização multi-objetivo para o treinamento de MLPs também foi proposto por Teixeira [Teixeira, 2001]. Denominado *Método  $\varepsilon$ -restrito*, este método também permite expressar o problema multi-objetivo na forma mono-objetiva sendo as soluções encontradas utilizando o algoritmo elipsoidal. Nesse caso o objetivo da minimização é aplicada à função do erro e a função da norma é definida como uma restrição.

$$\min_{\mathbf{w} \in W} \frac{1}{N_T} \sum_{j=1}^{N_T} (d_j - f(\mathbf{x}_j; \mathbf{w}))^2 \quad (2.34)$$

sujeito a:  $\|\mathbf{w}\| \leq \varepsilon$

Além de tornar desnecessário o cálculo de uma solução utilizando o algoritmo *back-propagation* padrão, em algumas situações esse método apresenta uma redução no tempo de cálculo dos pontos pertencentes ao conjunto Pareto.

## 2.5 Conclusões do Capítulo

Neste capítulo, realizou-se uma revisão de alguns dos principais algoritmos de treinamento de RNAs do tipo MLP. O algoritmo *backpropagation* e suas variações (Momentum, RPROP, QPROP, Taxa Adaptativa) têm como principal objetivo o aumento da velocidade de convergência durante o treinamento e a minimização da função de custo definida pela soma dos erros quadráticos. O Algoritmo de *Levenberg-Marquardt* utiliza uma aproximação quadrática da função de custo, reduzindo o número de iterações durante o processo de treinamento.

O algoritmo de Modos Deslizantes (SMC) garante a convergência do erro e do gradiente para a origem do sistema uma vez que as superfícies de deslizamento são alcançadas. Os ganhos dos ajustes dos pesos bem como as inclinações das superfícies de deslizamentos são obtidas a partir da análise da convergência segundo a teoria de modos deslizantes.

O algoritmo *Multi-Objetivo* apresenta o treinamento de MLPs sob uma nova perspectiva: o espaço solução bi-dimensional definido pelo erro (função custo) e a norma dos pesos (soma dos valores absolutos). Um conjunto de soluções, que apresenta erro mínimo para valores específicos de norma, é gerado e a solução que apresenta a melhor resposta de validação é selecionada. Para a obtenção de  $k$ -elementos do conjunto Pareto, o algoritmo implementa a otimização de  $k$ -problemas de minimização onde cada problema é definido a partir de duas restrições: do erro e da norma. O uso do método  $\varepsilon$ -restrito possibilita uma redução do custo computacional.

O método SMC-MOBJ (Sliding Mode Control - Multi Objective) proposto utiliza os gradientes descendentes das funções do erro e da norma controlado por duas superfícies de deslizamento para controlar o deslocamento do estado da rede no plano multi-objetivo. Apesar do uso do método do gradiente descendente, o algoritmo é capaz de convergir para qualquer solução do conjunto Pareto, partindo de qualquer ponto no plano de soluções. Uma vez que a teoria dos Modos Deslizantes é capaz de garantir a convergência do método ao ponto desejado, variações desse método possibilitam o aumento da sua velocidade de convergência.

# Capítulo 3

## O Método Multi-Objetivo

Neste capítulo será caracterizado o treinamento multi-objetivo de RNAs, especificamente as do tipo MLP. O plano de soluções, Erro  $\times$  Norma, utilizado no treinamento multi-objetivo, bem como o conjunto de soluções *eficientes* também denominado de conjunto Pareto-ótimo constituído pelas soluções de erro mínimo para valores específicos de norma serão descritos. Serão analisadas as equações e os comportamentos expressos pelos nodos, perceptrons, e redes MLPs. O objetivo de tal análise visa a identificação de características desses elementos bem como extrair informações úteis aos processos de inicialização e treinamento de topologias. Exemplos de treinamento utilizando o algoritmo backpropagation padrão serão demonstrados no plano de soluções, justificando a necessidade do uso de métodos multi-objetivo.

### 3.1 A abordagem Mono-Objetivo no treinamento de MLPs

Seja a definição de um funcional da seguinte forma:

**Funcional** : uma função cuja imagem é escalar

Sendo  $\mathbf{w}$ , onde  $\mathbf{w} \in \mathbb{R}^n$ , o vetor de parâmetros de ajuste ou de pesos de uma rede MLP. O funcional-objetivo que quantifica cada solução  $\mathbf{w}$  pode ser caracterizado como :  $f(\mathbf{w}) : \mathbb{R}^n \rightarrow \mathbb{R}$ . Através desta definição é possível caracterizar a função do erro médio quadrático de uma rede MLP,  $E_{(\mathbf{w})}$ , como um *funcional-objetivo*.

O problema de otimização mono-objetivo, nesse caso, pode ser descrito como:

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} E_{(\mathbf{w})} \quad (3.1)$$

Dessa forma, o processo de otimização deve ser capaz de determinar o vetor  $\mathbf{w}^*$  que minimiza o funcional  $E_{(\mathbf{w})}$ . O algoritmo *backpropagation* [Rumelhart et al., 1986] e suas

variações utilizam a abordagem mono-objetivo descrita na Equação 3.1.

Os algoritmos de treinamento mono-objetivo de MLPs procuram minimizar uma única função de custo, normalmente definida pela soma dos erros quadráticos, utilizando métodos para o ajuste do vetor de pesos da rede. Basicamente, para a realização do treinamento, o projetista deve especificar o limiar do erro a ser alcançado. Uma vez que esse limiar é atingido o projetista pode redefiní-lo utilizando um novo valor e realizando um novo treinamento ou finalizar a etapa de treinamento da rede. Dessa forma, o espaço solução utilizado é definido pelo Erro<sup>1</sup> em função dos passos (epochs). O objetivo dos algoritmos mono-objetivo consiste então, em alcançar ou ultrapassar o limiar especificado para o erro no menor número de iterações (passos) possível. A Figura 3.1 exemplifica o comportamento do processo de treinamento segundo a representação Erro  $\times$  Epoch que evidencia a velocidade de convergência. Nesse exemplo, foram utilizados o algoritmo *back-propagation* padrão e o algoritmo de *Levenberg-Marquardt*, ambos procuram minimizar o funcional Erro.

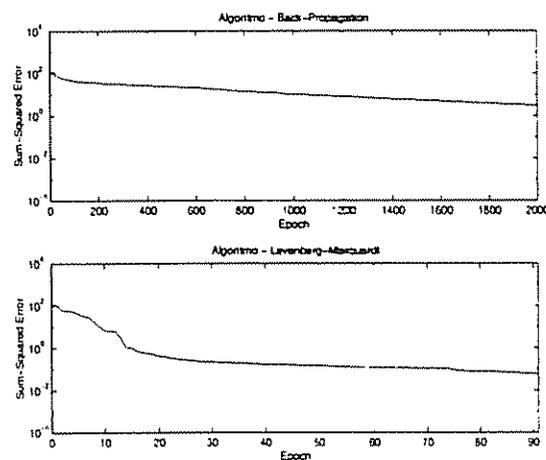


Figura 3 1: Desempenho de algoritmos no espaço solução (Erro  $\times$  Épocas)

É desejado que, utilizando uma representação gráfica do estado de um rede em um espaço solução, o seu comportamento ou algumas características do treinamento possam ser identificadas. Se, em um espaço de soluções, duas regiões pudessem ser associadas aos efeitos de *underfitting* e *overfitting* então, a solução de equilíbrio estaria localizada entre essas regiões. Na representação Erro  $\times$  Epoch, esses estados estão associados ao limiar do erro estabelecido bem como ao dimensionamento da rede. Um limiar de erro elevado pode associar à rede um comportamento com *underfitting* e um limiar reduzido pode causar o efeito de *overfitting*. Mas, independente do limiar do erro, uma rede mal dimensionada pode causar *underfitting*.

<sup>1</sup>subentende-se Erro como a soma dos erros quadráticos

A proposta do método multi-objetivo para treinamento de MLPs [Teixeira et al., 2000] implica na definição de um espaço solução em  $\mathcal{R}^2$  definido pelos funcionais Norma e Erro. Uma vez que os efeitos de *overfitting* e *underfitting* estão diretamente associados aos valores dos pesos e ao erro durante o treinamento da RNA, tal representação possibilita a visualização do comportamento da rede durante o treinamento e a definição de regiões *ideais* de busca para a solução final. Entretanto, essa nova representação não permite que seja analisada a característica temporal da convergência, ou seja, não é possível extrair informações sobre a velocidade de convergência no espaço Norma  $\times$  Erro. Como solução, pode-se utilizar o espaço Erro  $\times$  Epoch para complementar a análise temporal. Na implementação do algoritmo proposto, além do uso da representação no espaço solução: Erro  $\times$  Norma, os espaços definidos pelo erro e pela norma em função dos passos (epochs) também foram utilizados, possibilitando uma análise da velocidade de convergência do método.

## 3.2 A abordagem Multi-Objetivo no treinamento de MLPs

Sejam os funcionais erro e norma ( $E_{(w)}$ ,  $\|w\|$ ) agora definidos como funções objetivo uma vez que esses funcionais permitem avaliar o vetor de parâmetros de ajuste,  $w$ . Segundo a abordagem da *otimização multiobjetivo*, havendo mais de um objetivo em questão existirão dois tipos de soluções:

1. Soluções capazes de minimizar ou melhorar simultaneamente ambos os objetivos implicando na existência de soluções melhores.
2. Soluções que, comparadas com outras soluções, serão melhores em algum dos objetivos, seja erro ou norma, mas piores quando for considerado o outro objetivo.

As soluções pertencentes a esse segundo grupo são denominadas *soluções eficientes* ou *soluções Pareto-ótimo*. A determinação de tais soluções é um dos problemas da otimização multi-objetivo. A princípio, todas as soluções pertencentes ao conjunto *Pareto-ótimo* são candidatas a se tornarem a solução final. Não há como definir a partir das funções objetivo que uma solução é melhor que outra.

O problema multi-objetivo para treinamento de RNAs [Teixeira et al., 2000] pode ser descrito pela Equação 3.2

$$w^* = \arg \min_w \begin{cases} E_{(w)} \\ \|w\| \end{cases} \quad (3.2)$$

Em problemas genéricos de otimização multi-objetivo são definidas restrições que limitam a região de *busca* das soluções *eficientes*. Essas restrições podem ser definidas através

de  $m$  relações de desigualdade:

$$g_i(\mathbf{w}) \geq 0 \quad i = 1, 2, \dots, m \quad (3.3)$$

e/ou  $p$  restrições de igualdade:

$$h_i(\mathbf{w}) = 0 \quad i = 1, 2, \dots, p \quad (3.4)$$

As restrições definidas pela Equações 3.3 e 3.4 definem a *Região Factível*  $\mathcal{W}$  sendo a *Solução Factível* definida como sendo qualquer ponto  $\mathbf{w} \in \mathcal{W}$ . O treinamento Multi-objetivo de RNAs proposto [Teixeira et al., 2000] não possui restrições sendo que o conjunto factível  $\mathcal{W}$  abrange todas as soluções pertencentes ao espaço  $\mathbb{R}^n$  ( $\mathbf{w} \in \mathbb{R}^n$ ).

Uma vez definida a topologia de uma RNA, a qualquer ponto  $\mathbf{w} \in \mathcal{W}$  é associado um ponto de coordenadas  $(E_{(\mathbf{w})}, \|\mathbf{w}\|)$  no plano de soluções definido pelas funções objetivo (erro e norma). Existe então um mapeamento do conjunto  $\mathcal{W}$  no espaço  $\mathbb{R}^2$  definindo então um conjunto  $\Omega$  nesse espaço. A Figura 3.2 permite visualizar o conjunto  $\Omega$  e a curva Pareto ou o conjunto Pareto-ótimo. A curva Pareto representa o limite entre o conjunto  $\Omega$  (região factível) e a região de soluções não-existent.

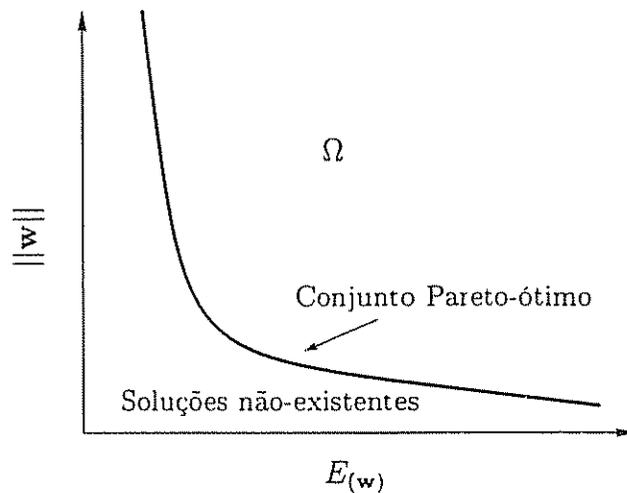


Figura 3.2: Curva Pareto-ótimo no Plano de Soluções

A definição para o conjunto Pareto-ótimo pode ser dada pela Equação 3.5.

$$\mathcal{W}^* \triangleq \{\mathbf{w}^* \in \mathcal{W} \mid \nexists \mathbf{w} \text{ tal que } \mathbf{f}(\mathbf{w}) \leq \mathbf{f}(\mathbf{w}^*) \text{ e } \mathbf{f}(\mathbf{w}) \neq \mathbf{f}(\mathbf{w}^*)\} \quad (3.5)$$

onde  $\mathbf{f}$  é o vetor de funções definido da seguinte forma:

$$\mathbf{f}(\mathbf{w}) = [E_{(\mathbf{w})}, \|\mathbf{w}\|]^T, \quad \mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^2 \quad (3.6)$$

e os operadores  $\leq$  e  $\neq$  são definidos na forma:

$$f(\mathbf{w}) \leq f(\mathbf{w}^*) \Rightarrow \{E_{(\mathbf{w})} \leq E_{(\mathbf{w}^*)}, \|\mathbf{w}\| \leq \|\mathbf{w}^*\|\}$$

$$f(\mathbf{w}) \neq f(\mathbf{w}^*) \Rightarrow \{E_{(\mathbf{w})} \neq E_{(\mathbf{w}^*)} \text{ ou } \|\mathbf{w}\| \neq \|\mathbf{w}^*\|\}$$

Segundo a Equação 3.5, uma solução  $\mathbf{w}^*$  é Pareto ótimo se não há mais como melhorar todos os objetivos simultaneamente, ou seja, não existe outro vetor  $\mathbf{w}$  capaz de reduzir um dos objetivos sem causar o incremento do outro e vice-versa.

Considere  $\mathbf{w}_1^*$  e  $\mathbf{w}_2^*$  os vetores referentes aos pontos de mínimo do funcional erro,  $E_{(\mathbf{w}_1^*)}$ , e norma,  $\|\mathbf{w}_2^*\|$ , respectivamente. Se os vetores são distintos,  $\mathbf{w}_1^* \neq \mathbf{w}_2^*$ , a coordenada  $(E_{(\mathbf{w}_1^*)}, \|\mathbf{w}_2^*\|)$  encontra-se mapeada na região de soluções não-existent e representa a solução ideal ou *Solução Utópica*. Idealmente, a solução *utópica* representa uma rede com erro nulo e norma nula. Entretanto, as funções objetivo são conflitantes: uma rede com norma nula apresenta um erro não nulo, normalmente elevado, já uma rede superdimensionada é capaz de modelar o conjunto de treinamento, mapeando inclusive distorções ou ruídos presentes nos dados resultando em uma rede com erro reduzido e norma elevada. A Figura 3.3 exemplifica esse comportamento entre as funções objetivo do erro e da norma. É possível distinguir os vetores de pesos que minimizam cada uma das funções e que caracterizam a solução *utópica*.

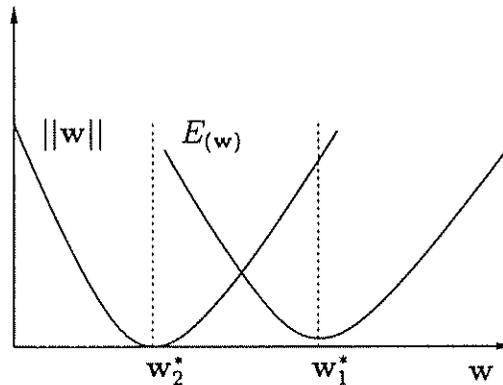


Figura 3.3: Solução utópica (não-existente)

A partir da definição da solução *utópica* pode-se definir a região de soluções não-existent da seguinte forma: Um par de coordenadas em  $\mathbb{R}^2$   $(E_{(\mathbf{w}^i)}, \|\mathbf{w}^i\|)$  caracteriza uma solução não-existente se não existe um único vetor real  $\mathbf{w}^i$  em  $\mathbb{R}^n$  associado a esse ponto (Equação 3.7).

$$\{(E_{(\mathbf{w}^i)}, \|\mathbf{w}^i\|) \notin \Omega \rightarrow \mathbf{w}^i \notin \mathcal{W}\} \quad (3.7)$$

Uma vez determinado o conjunto das soluções eficientes, a próxima etapa do método Multi-Objetivo (MOBJ) consiste na escolha de uma única solução, a solução final. Considerando que a solução final, ou seja, a rede MLP deverá apresentar um ajuste adequado

em relação à função geradora dos dados de treinamento, pode-se concluir que a solução final deverá possuir uma alta capacidade de generalização. A esse processo de escolha da solução final denomina-se de *Etapa de Decisão* sendo o *decisor* o método ou o critério que irá escolher a alternativa final dentre aquelas contidas no conjunto das soluções eficientes. O decisor implementado no método MOBJ [Teixeira, 2001] e que será utilizado neste trabalho é baseado no erro de validação para a tomada de decisão. Um conjunto de validação é apresentado a todas as redes obtidas na primeira fase, que são as soluções eficientes. aquela que apresentar o menor erro para os padrões de validação é escolhida como solução final. A regra de decisão é dada pela Equação 3.8

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathcal{W}^*} E_v(\mathbf{w}) \quad (3.8)$$

onde  $E_v$  é o funcional erro em relação ao conjunto de validação.

*o vetor  $\mathbf{w}^*$  que minimizar o funcional  $E_v$  é a melhor solução pertencente ao conjunto Pareto-ótimo e sua alta capacidade de generalização é assegurada [Teixeira, 2001]*

### 3.3 O Comportamento Multi-Objetivo do Perceptron

Uma análise do comportamento multi-objetivo será realizada para um nodo Perceptron simples. Um exemplo constituído por uma entrada, um termo de polarização (bias) e uma saída foi proposto. Neste caso, é possível representar a superfície do Erro bem como a superfície da Norma em um espaço 3-dimensional. O nodo possui função de ativação tangente hiperbólica e o conjunto de treinamento definido pelos seguintes pares de entrada-saída:  $\{x,y\}:\{-3.0,-0.4\}$  e  $\{2.0,0.8\}$ .

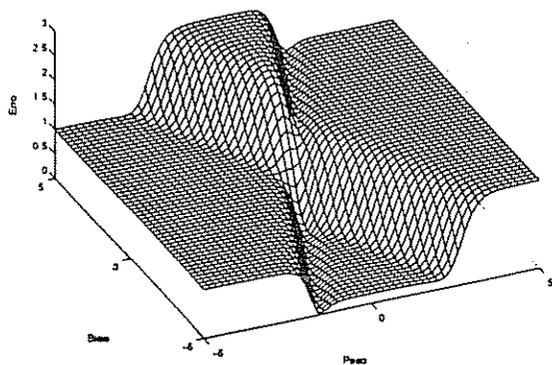


Figura 3.4: Superfície do Erro

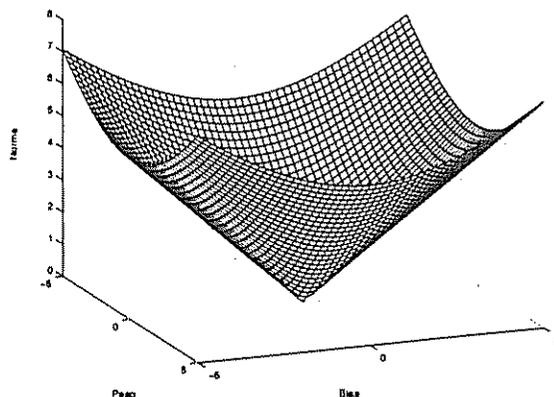


Figura 3.5: Superfície da Norma

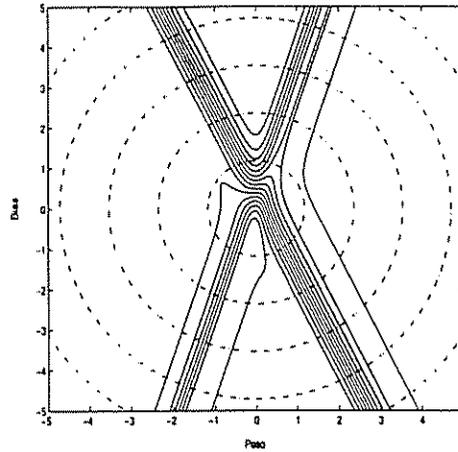


Figura 3.6: Curvas de nível das funções Erro e Norma

As Figuras 3.4, 3.5 e 3.6 representam o comportamento das funções do Erro e da Norma no espaço solução definido pelo peso e bias do nodo. As soluções foram obtidas a partir de uma distribuição homogênea de seus valores. A partir desses resultados, pode-se mapear essas soluções de forma a obter uma distribuição no espaço bi-dimensional Erro  $\times$  Norma bem como o conjunto Pareto-ótimo.

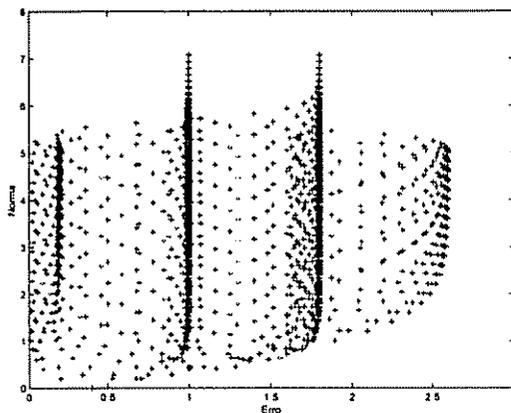
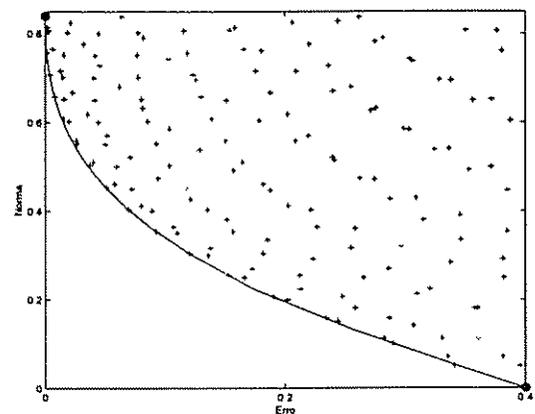
Figura 3.7: Plano de Soluções: Erro  $\times$  Norma

Figura 3.8: Conjunto Pareto-Ótimo

A Figura 3.7 apresenta a distribuição das soluções obtidas no plano Erro  $\times$  Norma. Apesar de os valores dos pesos utilizados apresentarem uma distribuição homogênea, a figura demonstra uma concentração de soluções em regiões de mesmo erro. De acordo com a Figura 3.4, existem quatro regiões bem distintas onde o gradiente é aproximadamente nulo. Em tais regiões o erro é praticamente constante, independente do valor da norma. De acordo com a Figura 3.6, existem regiões com valores constantes de norma que correspondem a círculos de raio constante. Esses círculos, dependendo dos valores do raio (norma), são capazes de *atravessar* todas as regiões de erro nulo. Neste exemplo, quanto maior o raio do círculo, maiores são as regiões de gradiente nulo abrangidas. Tal

comportamento justifica o fato de que, no espaço solução Erro  $\times$  Norma existem regiões de mesma norma que apresentam o mesmo erro. Tais regiões são denominadas *mínimos locais* pois estão localizadas em uma região plana da superfície do erro (gradiente nulo) e, nesse caso, correspondem às faixas verticais visualizadas na Figura 3.7.

Na Figura 3.8 é possível observar o conjunto Pareto-ótimo que corresponde às soluções localizadas à esquerda do gráfico, ou seja, para valores constantes de norma, correspondem às soluções de erro mínimo. Essas soluções estão localizadas entre os mínimos globais das funções de erro e norma respectivamente. As Figuras 3.9 e 3.10 permitem visualizar o conjunto Pareto-ótimo na superfície do Erro e nas curvas de nível do Erro e da Norma, respectivamente.

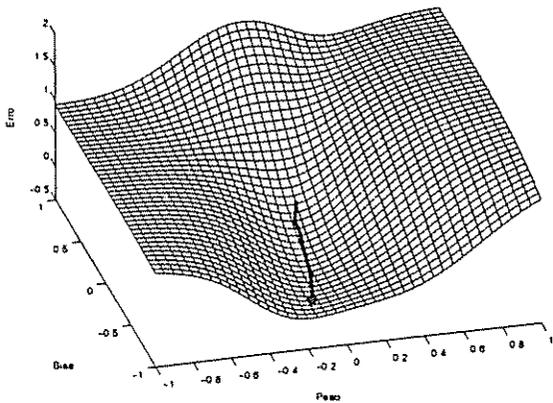


Figura 3.9: Superfície do Erro: Conjunto Pareto-Ótimo

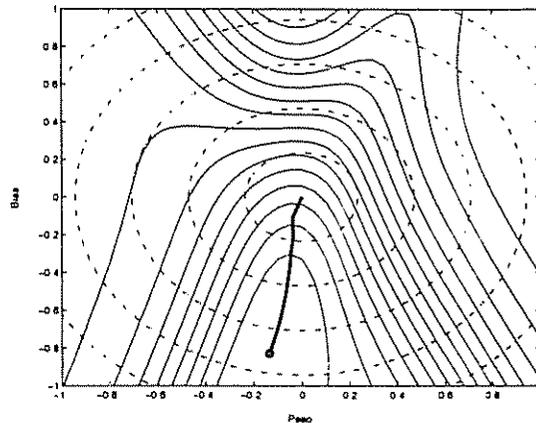


Figura 3.10: Conjunto Pareto-Ótimo: Curvas Equipotenciais de Erro e Norma

### 3.3.1 Análise do Comportamento dos Gradientes

Em relação aos gradientes ou às inclinações da superfície do Erro é possível concluir que, em regiões de baixa norma, os gradientes são não-nulos. Seja a equação do gradiente da função do erro para o nodo:

$$\frac{\partial E_{(k)}}{\partial w_{j(k)}} = -e_{(k)} \cdot f'_{(k)} \cdot x_{j(k)} \quad (3.9)$$

onde:  $e_{(k)} = d - y$ ,  $f_{(k)} = f\left(\sum_{j=1}^n x_{j(k)} \cdot w_{j(k)}\right)$  (função de ativação) e  $x_j$  é a entrada do nodo.  $x_{n(k)} = 1$ , refere-se ao termo de polarização.

A seguinte situação define um gradiente não nulo:

$$\left(\frac{\partial E_{(k)}}{\partial w_{j(k)}} \neq 0\right) \Rightarrow (e_{(k)} \neq 0, f'_{(k)} \neq 0, x_{j(k)} \neq 0) \quad (3.10)$$

Para pesos com valores próximos de zero, tem-se a seguinte situação:

$$e_{i(k)} = \lim_{w_{j(k)} \rightarrow 0} \left[ d_i - f_i \left( \sum_{j=1}^n x_{j(k)} \cdot w_{j(k)} \right) \right] = d_i \quad (3.11)$$

uma vez que as funções de ativação linear e tangente hiperbólica apresentam a seguinte característica:

$$\lim_{u \rightarrow 0} f(u) = 0$$

onde  $u = \sum_{j=1}^n x_{j(k)} \cdot w_{j(k)}$ .

Para entradas não-nulas,  $x_{j(k)} \neq 0$ , o gradiente em relação a função do erro pode ser escrito como:

$$\frac{\partial E_{(k)}}{\partial w_{j(k)}} \propto f' \left( \sum_{j=1}^n x_{j(k)} \cdot w_{j(k)} \right)$$

Normalmente, as seguintes funções de ativação são utilizadas: tangente hiperbólica, sigmoidal ou linear. Os gradientes das funções: tangente hiperbólica e sigmoidal podem ser visualizados nas Figuras 3.11 e 3.12. O gradiente da função linear é a constante unitária ( $f'_i = 1$ ).

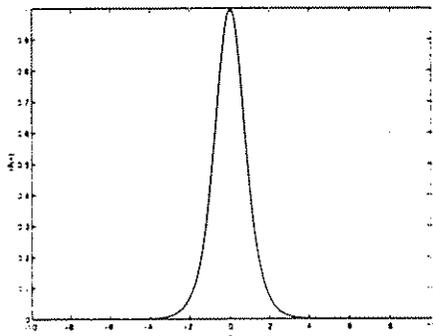


Figura 3.11: Derivada - Função Hiperbólica

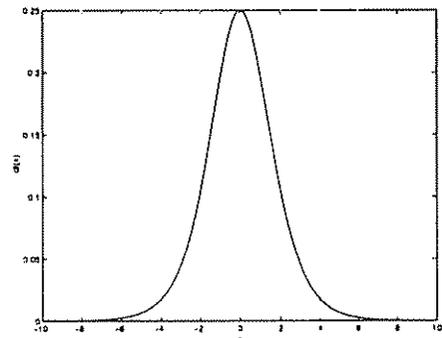


Figura 3.12: Derivada - Função Sigmoidal

Segundo as Figuras 3.11 e 3.12, o gradiente das funções de ativação tangente hiperbólica e sigmoidal apresentam valores máximos quando a entrada é nula. Para garantir um valor do gradiente não nulo e próximo do valor máximo, os pesos iniciais do Perceptron devem ser dimensionados da seguinte forma:

$$\max \left\{ f' \left( \sum_{j=1}^n x_{j(k)} \cdot w_{j(k)} \right) \right\} \Rightarrow \sum_{j=1}^n x_{j(k)} \cdot w_{j(k)} = 0 \quad (3.12)$$

Segundo a Equação 3.12, a situação onde  $w_{j(k)} = 0$  é uma das soluções. Para o Perceptron, as entradas do nodo,  $x_j$ , influenciam no valor do gradiente da função de ativação em relação a cada padrão de entrada. Dessa forma, para garantir que o gradiente do Perceptron seja não-nulo, basta inicializar os pesos com norma baixa ou nula.

Por outro lado, a partir da Equação 3.9 é possível determinar algumas características que permitem definir o comportamento de mínimo global e mínimos locais em um nodo perceptron. Em ambos os casos os gradientes são nulos.

$$\frac{\partial E_{(k)}}{\partial w_{j(k)}} = 0 \Rightarrow -e_{(k)} \cdot f'_{(k)} \cdot x_{j(k)} = 0 \quad (3.13)$$

As situações que garantem a condição 3.13 podem ser representadas da seguinte forma:

1.  $e_{(k)} = 0$  ou  $e_{(k)} \approx 0$
2.  $f'_{(k)} = 0$  ou  $f'_{(k)} \approx 0$
3.  $x_{j(k)} = 0$

Restringindo o problema a situações nas quais:  $\{x_{j(k)} \neq 0, \forall (j, k)\}$ , é possível determinar as condições em relação ao gradiente que caracterizam e diferenciam mínimo global e mínimos locais.

- Mínimo Global:

$$e_{(k)} = 0. \quad \forall f'_{(k)}$$

Significa que o erro entre a resposta da rede ( $y$ ) e a resposta desejada ( $d$ ) é praticamente nulo, independente do valor da derivada da função de ativação no ponto, tornando o gradiente nulo.

- Mínimos Locais:

$$e_{(k)} \neq 0$$

$$f'_{(k)} = 0$$

O gradiente da função de custo em um ponto de mínimo global está fortemente associado ao valor reduzido da diferença entre o padrão desejado na saída do nodo e o real, ou seja, o erro. Essa característica independe do valor da derivada da função de ativação no ponto. Por outro lado, o gradiente da função de custo em relação a um ponto ou a uma região de mínimo local está associado à redução do valor da derivada da função de ativação, apresentando um valor nulo ou próximo de zero mesmo que a diferença entre a saída desejada e a saída real do nodo seja diferente de zero.

Analisando novamente as Figuras 3.11 e 3.12, percebe-se que as derivadas das funções de ativação não-lineares não apresentam um ponto de mínimo, mas duas regiões nas quais ocorrem um decaimento do seu valor absoluto para valores próximos de zero, o que indica que nas situações onde  $f'_{(k)} = 0$  são representadas, na realidade, por intervalos onde a

derivada da função de ativação apresenta valores abaixo de um determinado limiar  $\zeta$ . Para as funções tangente hiperbólica e sigmoidal, pode-se definir a seguinte situação:

$$f'(u) < \zeta, \text{ se } |u| > f'_{(\zeta)}^{-1} \quad (3.14)$$

onde  $f'_{(u)}^{-1}$  representa a função inversa de  $f'(u)$ :  $f'(u) = b \Rightarrow f'_{(b)}^{-1} = u$ . Ou seja, para que a derivada da função de ativação seja menor do que um determinado limiar  $\zeta$ , a entrada  $u$ , ou o valor absoluto do produto interno entre o vetor de entrada do nodo e o vetor dos pesos, definido pelo escalar  $u$ , deve ser superior à função inversa desse limiar.

A Equação 3.14 também pode ser escrita como:

$$\begin{aligned} f' \left( \sum_{j=1}^n x_{j(k)} \cdot w_{j(k)} \right) < \zeta \\ \left| \sum_{j=1}^n x_{j(k)} \cdot w_{j(k)} \right| > f'_{(\zeta)}^{-1} \end{aligned} \quad (3.15)$$

Seja então, definido um valor para o parâmetro  $\zeta = 1e - 3$ , a função tangente hiperbólica apresenta valores de derivada abaixo desse limiar quando:  $f'_{(1e-3)}^{-1} \approx 4.15$ ,  $\left| \sum_{j=1}^n x_{j(k)} \cdot w_{j(k)} \right| > 4.15$ . Dessa forma, a Equação 3.15 pode ser utilizada para verificar se um nodo apresenta ou não características associadas aos mínimos locais.

### 3.4 Representação dos Pesos em Coordenadas Polares

Os pesos de um perceptron ou mesmo de uma rede MLP podem ser descritos por um raio e ângulos. Uma vez que a rede pode ser representada pela função da Norma, é possível decompor essa função em equações envolvendo um raio e ângulos.

Seja a função da Norma de um MLP:

$$\|\mathbf{w}_{(k)}\| = r = \sqrt{\sum_{ji} w_{ji(k)}^2} \quad (3.16)$$

$$\sum_{ji} w_{ji(k)}^2 = r^2 \quad (3.17)$$

A Equação 3.17 representa a equação de uma circunferência  $n$ -dimensional onde  $n$  refere-se ao número de pesos da rede. Para o caso de um neurônio constituído de apenas dois pesos,  $w_1$  e  $w_2$ , pela Equação 3.17 tem-se:

$$w_1^2 + w_2^2 = r^2$$

o que corresponde à equação de um círculo de raio  $r$ , onde:

$$w_1 = r \cdot \text{sen}\theta, w_2 = r \cdot \text{cos}\theta$$

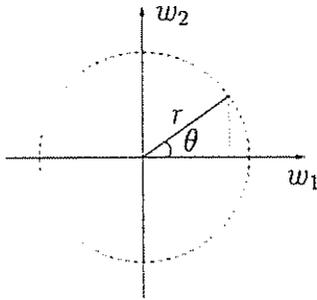


Figura 3.13: Representação dos pesos na forma polar - bidimensional

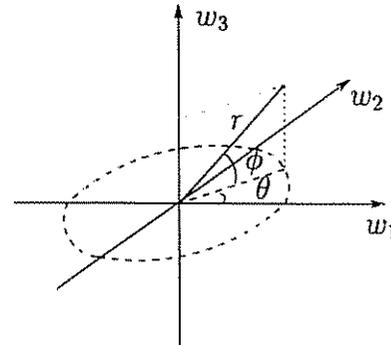


Figura 3.14: Representação polar - tridimensional

O conjunto solução para um nodo de duas entradas com norma igual a  $r$ , pode ser descrito como:

$$[w_1, w_2] \in \{r \cdot \text{sen}\theta, r \cdot \text{cos}\theta\}, \text{ onde: } 0 \leq \theta < 2\pi \text{ e } r \in \mathbb{R}^+ \quad (3.18)$$

Se o nodo apresentar três pesos na entrada a equação da Norma será da forma:

$$w_1^2 + w_2^2 + w_3^2 = r^2$$

As soluções podem ser representadas na forma polar da seguinte maneira:

$$\begin{aligned} w_1 &= r \cdot \text{sen}\phi \\ w_2 &= r \cdot \text{cos}\phi \cdot \text{sen}\theta \\ w_3 &= r \cdot \text{cos}\phi \cdot \text{cos}\theta \end{aligned} \quad (3.19)$$

$$0 \leq \theta < 2\pi, \frac{-\pi}{2} \leq \phi \leq \frac{+\pi}{2}$$

Novamente, o conjunto solução para um nodo com 4 entradas é da forma:

$$\begin{aligned} w_1 &= r \cdot \text{sen}\phi \\ w_2 &= r \cdot \text{cos}\phi \cdot \text{sen}\theta \\ w_3 &= r \cdot \text{cos}\phi \cdot \text{cos}\theta \cdot \text{sen}\alpha \\ w_4 &= r \cdot \text{cos}\phi \cdot \text{cos}\theta \cdot \text{cos}\alpha \end{aligned} \quad (3.20)$$

De posse dos resultados para as representações polares bidimensional e tridimensional, é possível generalizar os resultados para uma rede MLP n-dimensional descrevendo os seus pesos através da norma e dos respectivos ângulos. No caso n-dimensional a representação polar dos pesos é obtida através da decomposição em um raio,  $r$ , e  $n - 1$  ângulos.

Dado um vetor de pesos:  $\mathbf{w} = \{w_1, w_2, w_3, \dots, w_n\}$  e um vetor de ângulos:  $\theta = \{\theta_1, \theta_2, \theta_3, \dots, \theta_{n-1}\}$  os pesos podem ser descritos pela combinação da norma  $\|\mathbf{w}\|$  e dos

ângulos da seguinte maneira:

$$w_i = \begin{cases} r \cdot \delta_i \cdot \text{sen}(\theta_i) & \text{se } i \neq n \\ r \cdot \delta_i & \text{se } i = n \end{cases} \quad (3.21)$$

onde  $r = \|\mathbf{w}\|$  e  $\delta_i$  é descrito como:

$$\delta_i = \begin{cases} 1, & \text{se } i = 1 \\ \prod_{j=1}^{i-1} \cos(\theta_j) & \text{se } i \neq 1 \end{cases} \quad (3.22)$$

A equação de transformação para a forma polar é descrita a seguir:

$$\theta_i = \begin{cases} \arcsen\left(\frac{w_i}{r \delta_i}\right) & \text{se } i \neq n - 1 \\ \text{arctg}\left(\frac{w_{i+1}}{w_i}\right) & \text{se } i = n - 1 \end{cases} \quad (3.23)$$

Soluções de mesma norma e erro nem sempre apresentam pesos idênticos. Tal comportamento indica que essas soluções se localizam em regiões diferentes no espaço  $n$ -dimensional definido pelos seus pesos. Mas, uma vez que a coordenada é conhecida podem-se desenvolver técnicas para se deslocar no espaço solução independentemente da região onde a solução se encontra.

Um exemplo de análise utilizando a decomposição em coordenadas polares é apresentada nas Figuras 3.15 e 3.16. Para o perceptron utilizado anteriormente, círculos de raios iguais a  $r = 0.8396$ , onde encontra-se o mínimo global, e  $r = 3.5$  foram utilizados. Variando-se o ângulo  $\theta$  é possível obter os respectivos pesos e conseqüentemente, a função custo do erro. Os resultados também demonstram que a soma das derivadas da função de ativação apresenta valor nulo ou próximo de zero quando a função de custo é constante, validando a afirmativa de que, em regiões onde o erro é constante, a derivada da função de ativação é praticamente nula.

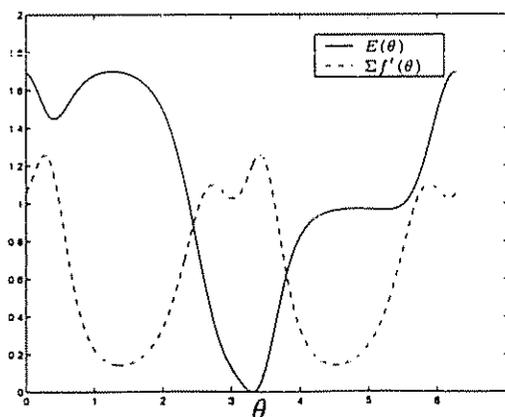


Figura 3.15: Soluções de raio  $r = 0.8396$

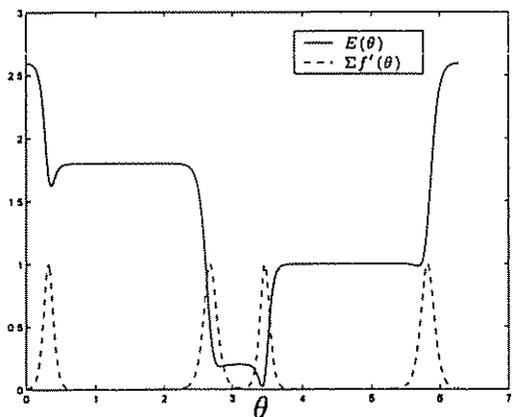


Figura 3.16: Soluções de raio  $r = 3.5$

A principal característica da decomposição dos pesos em coordenadas polares consiste na capacidade de análise de soluções de mesma norma. É possível encontrar o ponto de mínimo erro variando-se os ângulos do vetor de pesos sem que a norma dos pesos seja alterada. O método permite obter qualquer solução através do ajuste dos ângulos e da definição da norma dos pesos. Entretanto, o custo computacional do método cresce com o aumento da dimensão do espaço de busca, ou seja, o tamanho do vetor de pesos. Em relação ao exemplo proposto (Perceptron), o método permite obter os pontos pertencentes ao Pareto através da variação do ângulo do vetor bidimensional dos pesos.

### 3.5 Análise dos Gradientes de uma Rede MLP de Duas Camadas

A análise das equações do nodo perceptron permitiu analisar a influência da inicialização dos pesos com norma baixa. Foi observado que, no perceptron, pesos com baixos valores de norma ou praticamente nulos asseguram gradientes não nulos. A seguir, uma análise semelhante é realizada. Desta vez, as equações para uma rede MLP de duas camadas são analisadas. O objetivo é analisar ou identificar equações que caracterizam um conjunto de pesos com gradientes não-nulos.

Definindo:

**Q** : Número de saídas da rede

**H** : Número de nodos na camada escondida

**N** : Número de entradas da rede

**w** : Vetor de pesos da rede

**b** : Vetor de pesos referentes aos termos de polarização

a saída  $p$  da rede MLP pode ser descrita pela seguinte equação:

$$y_p = f_p \left\{ \sum_{i=1}^H w_{2ip} \cdot f_i \left[ \sum_{j=1}^N (x_j \cdot w_{1ji}) + b_{1i} \right] + b_{2p} \right\} \quad (3.24)$$

Sendo a função de custo do erro definido por:

$$E_{(k)} = \frac{1}{2} \sum_{p=1}^Q (d_p - y_{p(k)})^2$$

A sua derivada em função dos parâmetros de ajuste pode ser obtida da seguinte forma:

$$\frac{\partial E_{(k)}}{\partial w} = \frac{1}{2} \cdot 2 \cdot \sum_{p=1}^Q (d_p - y_{p(k)}) \cdot (-1) \cdot \frac{\partial y_{p(k)}}{\partial w}$$

$$\frac{\partial E_{(k)}}{\partial w} = - \sum_{p=1}^Q e_{p(k)} \frac{\partial y_{p(k)}}{\partial w} \quad (3.25)$$

A partir da Equação 3.25 é possível obter os gradientes referentes a cada peso contido na MLP

- Pesos da camada de saída

$$\frac{\partial y_{p(k)}}{\partial w_{2_{ip}}} = f'_{p(k)} \cdot f_{i(k)}$$

$$\frac{\partial E_{(k)}}{\partial w_{2_{ip}}} = -e_{p(k)} \cdot f'_{p(k)} \cdot f_{i(k)} \quad (3.26)$$

- Polarização (bias) da camada de saída

$$\frac{\partial E_{(k)}}{\partial b_{2_{ip}}} = -e_{p(k)} \cdot f'_{p(k)} \quad (3.27)$$

- Pesos da camada escondida ou de entrada

$$\frac{\partial y_{p(k)}}{\partial w_{1_{ji}}} = f'_{p(k)} \cdot w_{2_{ip}} \cdot f'_{i(k)} \cdot x_j$$

$$\frac{\partial E_{(k)}}{\partial w_{1_{ji}}} = - \sum_{p=1}^Q e_{p(k)} \cdot f'_{p(k)} \cdot w_{2_{ip}} \cdot f'_{i(k)} \cdot x_j \quad (3.28)$$

- Polarização (bias) da camada de entrada

$$\frac{\partial E_{(k)}}{\partial b_{1_i}} = - \sum_{p=1}^Q e_{p(k)} \cdot f'_{p(k)} \cdot w_{2_{ip}} \cdot f'_{i(k)} \quad (3.29)$$

onde:

$$f_{p(k)} = f_p \left\{ \sum_{i=1}^H w_{2_{ip}} \cdot f_i \left[ \sum_{j=1}^N (x_j \cdot w_{1_{ji}}) + b_{1_i} \right] + b_{2_p} \right\} \quad (3.30)$$

$$f_{i(k)} = f_i \left[ \sum_{j=1}^N (x_j \cdot w_{1_{ji}}) + b_{1_i} \right] \quad (3.31)$$

Através das Equações 3.26, 3.27, 3.28 e 3.29, é possível analisar o comportamento dos gradientes para pesos praticamente nulos ou mesmo nulos. Se todos os pesos forem

inicialmente nulos, devido à presença dos pesos da camada de saída ( $w_{2,ip}$ ) nas equações do gradiente da camada de entrada ou escondida, o gradiente dessa camada será nulo, o que também ocorre em relação ao gradiente dos pesos da camada de saída. Nessas equações, não existem pesos que anulam o gradiente, mas as entradas desses nodos correspondem às saídas dos nodos da camada anterior. Sendo a tangente hiperbólica a função de ativação normalmente utilizada na camada escondida, para pesos nulos a saída da camada de entrada será nula anulando os termos da equação do gradiente em relação aos pesos da camada de saída. Por fim, a equação do gradiente para o bias da camada de saída, para pesos nulos, não será nula. Entretanto, os ajustes desses pesos não influenciam nas equações dos demais gradientes.

Uma vez definidas as equações dos gradientes dos parâmetros de ajuste de uma rede MLP é possível extrair informações e definir intervalos de confiança de forma a garantir a não-saturação dos nodos. A Equação 3.26 bem como as Equações 3.28 e 3.29, apresentam respectivamente os seguintes termos:

$$1. f'_{p(k)} \cdot f_{i(k)}$$

$$2. f'_{p(k)} \cdot f'_{i(k)}$$

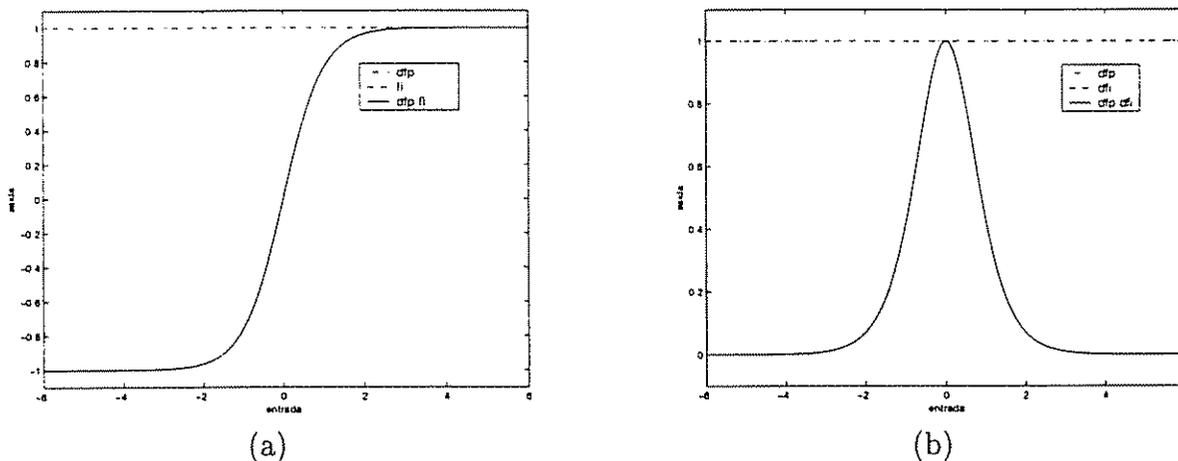


Figura 3.17:  $f_i$  -  $\tanh(x)$  e  $f_p$  -  $\text{linear}(x)$

O comportamento desses termos nas equações dos gradientes dependem do tipo de função de ativação definida nos nodos da camada escondida e da camada de saída da rede. As Figuras 3.17, 3.18 e 3.19 apresentam o comportamento desses termos em relação às funções de ativação normalmente utilizadas para uma rede MLP de duas camadas.

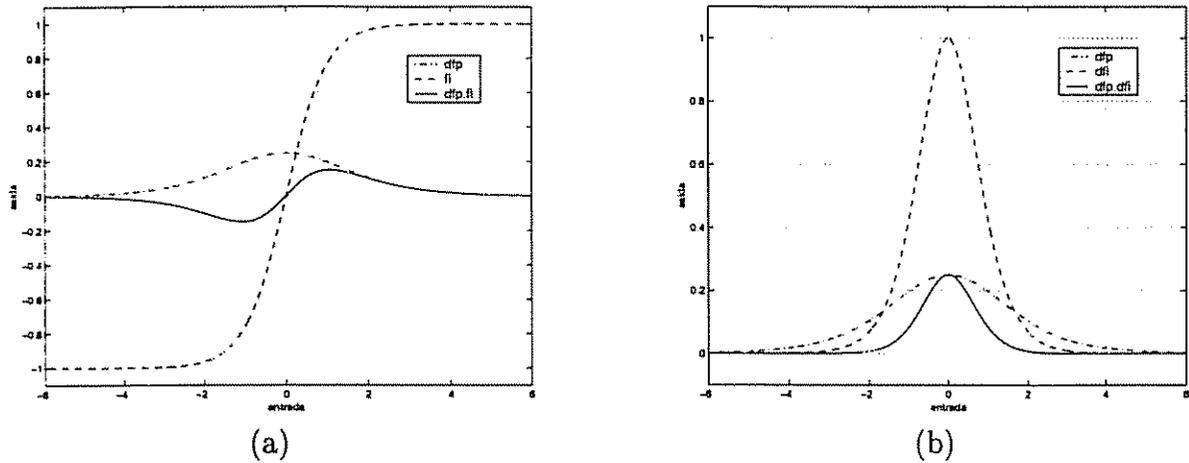


Figura 3.18:  $f_i$  -  $\tanh(x)$  e  $f_p$  -  $\text{sigmoidal}(x)$

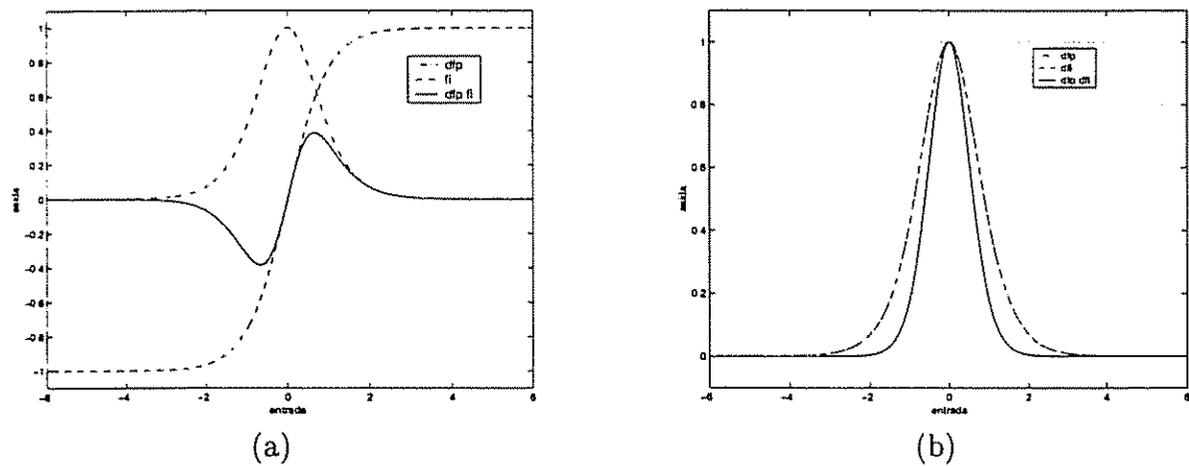


Figura 3.19:  $f_i$  -  $\tanh(x)$  e  $f_p$  -  $\tanh(x)$

A partir desses resultados podem-se especificar, para cada caso, os seguintes parâmetros:

- *Pontos de máximo e mínimo dos termos:*  $f'_{p(k)} \cdot f_{i(k)}$

As Figuras 3.18 (a) e 3.19 (a) apresentam pontos de máximo e mínimo simétricos que representam entradas onde o valor absoluto do produto  $f'_{p(k)} \cdot f_{i(k)}$  é máximo. A inicialização dos parâmetros da rede próximo a esses pontos assegura valores de gradientes elevados para os pesos da camada de saída. Para assegurar tal característica, de acordo com as Equações 3.30 e 3.31, as entradas das funções de ativação devem apresentar a seguinte forma:

$$f'_{p(k)} : \sum_{i=1}^H (f_{i(k)} \cdot w_{2ip}) + b_{2p} = \pm \varphi \quad (3.32)$$

$$f_{i(k)} : \sum_{j=1}^N (x_j \cdot w_{1ji}) + b_{1i} = \pm \varphi \quad (3.33)$$

onde  $\varphi$  é o valor absoluto do ponto de máximo do termo  $f'_{p(k)} \cdot f_{i(k)}$ , que varia de acordo com as funções de ativação

- Intervalo de confiança para os gradientes

Semelhante ao caso do Perceptron e de acordo com as Figuras 3.17 (b), 3.18 (b) e 3.19 (b), existe um intervalo para a entrada dos nodos da rede MLP no qual o termo  $f'_{p(k)} \cdot f'_{i(k)}$  apresenta valores abaixo de um limiar,  $\zeta$ , devido ao decaimento da função representada por esse produto com o aumento do valor absoluto de suas entradas. Esses limites estão representado pelas Equações 3.34 e 3.35.

$$f'_{p(k)} : \left| \sum_{i=1}^H (f_{i(k)} \cdot w_{2ip}) + b_{2p} \right| < \psi \quad (3.34)$$

$$f_{i(k)} : \left| \sum_{j=1}^N (x_j \cdot w_{1ji}) + b_{1i} \right| < \psi \quad (3.35)$$

Tal efeito também se observa em relação ao termo  $f'_{p(k)} \cdot f_{i(k)}$ . O valor de  $\psi$  é então definido como sendo o maior valor absoluto para a entrada dos nodos  $f_{p(k)}$  e  $f_{i(k)}$  que garante que ambos os termos  $f'_{p(k)} \cdot f'_{i(k)}$  e  $f'_{p(k)} \cdot f_{i(k)}$  apresentem valores abaixo de um limiar  $\zeta$ . No exemplo proposto para o nodo perceptron o valor de  $\zeta$  utilizado foi de  $1e - 3$ .

De acordo com as Figuras 3.18 (a) e 3.19 (a), em relação ao termo  $f'_{p(k)} \cdot f_{i(k)}$ , existe um intervalo adicional que inclui a origem e apresenta um comportamento semelhante. Ou seja, para garantir que os termos  $f'_{p(k)} \cdot f'_{i(k)}$  e  $f'_{p(k)} \cdot f_{i(k)}$  apresentem valores acima de um limiar  $\xi$  é preciso especificar dois limites, o primeiro definido pelo valor de  $\psi$  e o segundo pelo limiar  $\xi$ , tal que  $\psi > \xi$ , ( $\psi > 0$ ,  $\xi > 0$ ). As Equações 3.36 e 3.37 definem então, o intervalo para a entrada dos nodos de forma a garantir que os termos  $f'_{p(k)} \cdot f'_{i(k)}$  e  $f'_{p(k)} \cdot f_{i(k)}$  apresentem valores acima de um limiar  $\zeta$ .

$$f'_{p(k)} : \xi < \left| \sum_{i=1}^H (f_{i(k)} \cdot w_{2ip}) + b_{2p} \right| < \psi \quad (3.36)$$

$$f_{i(k)} : \xi < \left| \sum_{j=1}^N (x_j \cdot w_{1ji}) + b_{1i} \right| < \psi \quad (3.37)$$

A Tabela 3.1 mostra os pontos simétricos de máximo e mínimo bem como os parâmetros  $\psi$  e  $\xi$  referentes ao intervalo de confiança para as variações comumente utilizadas das funções de ativação das camadas de entrada e de saída de uma rede MLP de duas camadas. O limiar de saída foi definido como  $\zeta = 1e - 3$ .

Tabela 3.1: Pontos singulares e intervalos de confiança

| índice | $f_p$     | $f_i$ | $\varphi$ | $\psi$ | $\xi$ |
|--------|-----------|-------|-----------|--------|-------|
| 1      | linear    | tanh  | -         | 4.15   | 1e-3  |
| 2      | sigmoidal | tanh  | 1.06      | 2.72   | 4e-3  |
| 3      | tanh      | tanh  | 0.66      | 2.41   | 1e-3  |

A identificação desses intervalos em relação às entradas dos nodos permitem caracterizar e identificar os nodos que apresentam um comportamento de saturação ou cuja amplitude do gradiente esteja próxima de zero.

Na primeira situação, Figura 3.17, o termo  $f'_{p(k)} \cdot f_{i(k)}$  não apresenta pontos singulares nem intervalo de confiança o que indica que não existe, exceto na origem, valores de gradientes não nulos para os pesos ou os termos de polarização da camada de saída. Tal situação é obtida através da função de ativação linear da saída. Já o termo  $f'_{p(k)} \cdot f'_{i(k)}$  apresenta um intervalo de confiança. Os valores demonstrados na Tabela 3.1 referem-se ao intervalo de confiança dos pesos e bias da camada de entrada ( $f'_{p(k)} \cdot f'_{i(k)}$ ) mas, uma vez garantido esse intervalo, assegura-se a não saturação de todos os nodos da rede.

A segunda situação é semelhante à terceira (Figuras 3.18 e 3.19). Apesar da semelhança entre a função tangente hiperbólica e sigmoidal, seus intervalos de confiança bem como os respectivos pontos singulares são distintos. De um modo geral, os intervalos referentes à função tangente hiperbólica são mais estreitos.

A Tabela 3.1 fornece os intervalos para os valores dos pesos (Equações 3.36 e 3.37) que garantem a não-saturação da rede. Esses valores podem ser utilizados na geração do vetor inicial dos pesos para evitar regiões de saturação.

## 3.6 Análise do Algoritmo BackPropagation no Plano Erro $\times$ Norma

Os algoritmos multi-objetivos caracterizam-se pelo uso de mais de uma função de custo no treinamento de MLPs. Tais funções de custo representam características da rede, como a função norma dos pesos. Portanto, mesmo realizando um treinamento mono-objetivo, as características de convergência da rede durante o treinamento podem ser analisadas através dos valores do erro e da norma dos pesos. O objetivo desta seção consiste em analisar o comportamento do treinamento convencional, que utiliza o algoritmo *backpropagation* padrão, na perspectiva do plano multi-objetivo de soluções: Erro  $\times$  Norma.

A Figura 3.20 ilustra o comportamento de uma rede MLP treinada a partir de pesos inicializados com diferentes valores de norma. O treinamento foi realizado utilizando o algoritmo *backpropagation* padrão mantendo o mesmo número de iterações para cada valor inicial de norma. Os valores da taxa de aprendizado foram mantidos constantes.

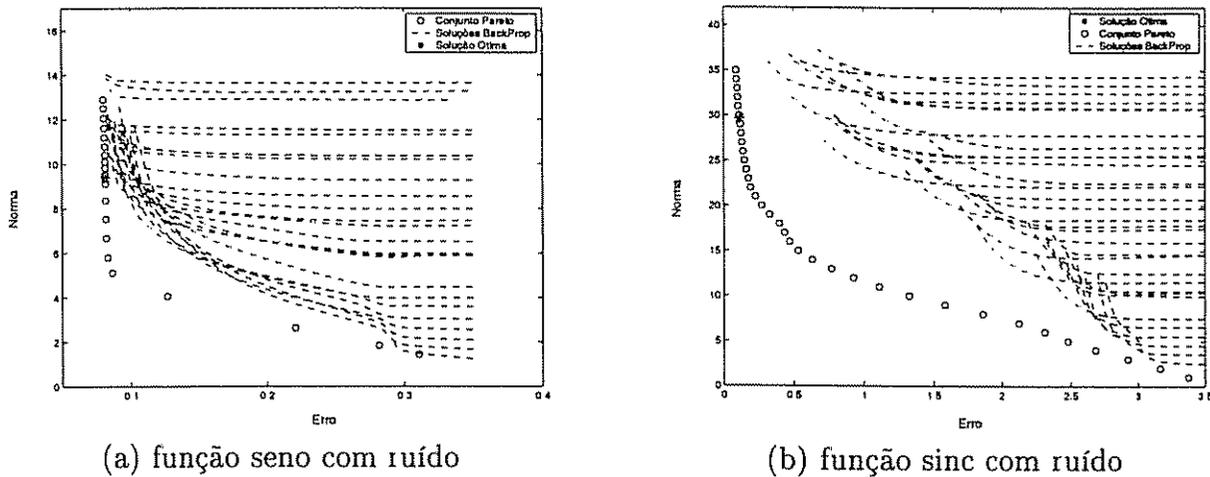
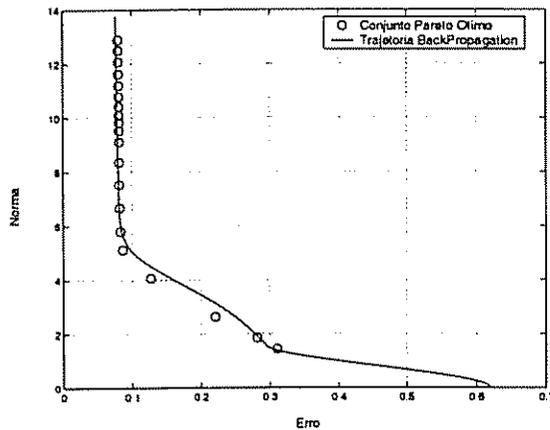


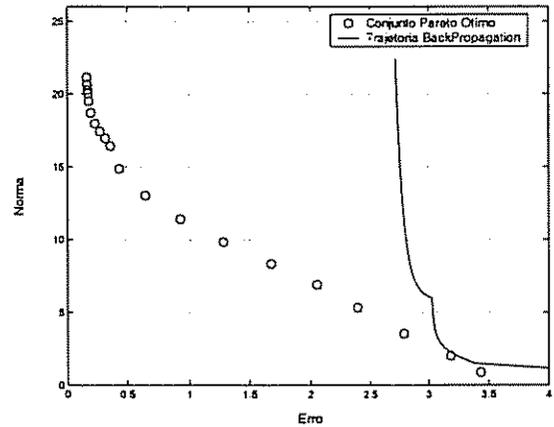
Figura 3.20: Comportamento do Algoritmo Backpropagation no Plano de Soluções: Erro  $\times$  Norma

Os resultados demonstram a convergência da rede para valores de erros próximos ao conjunto Pareto-ótimo. Entretanto, há uma tendência ao aumento da norma dos pesos com o aumento do número de iterações seguindo, aproximadamente o modelo expresso pela curva Pareto. Tal fenômeno prejudica a capacidade da rede de convergir ao ponto com o melhor equilíbrio ou ao ponto com alta capacidade de generalização, visto que o objetivo do treinamento convencional visa a obtenção do menor valor possível da função de custo definida pelo erro médio quadrático e que se encontra na extremidade superior do conjunto Pareto-ótimo. A partir dos resultados também é possível inferir que a inicialização dos pesos influi na convergência da rede. Pesos inicializados com valores elevados de norma podem apresentar uma maior velocidade de convergência do erro mas, tais soluções não apresentam trajetórias próximas do ponto ótimo de equilíbrio entre erro e norma. O que significa que um método de parada por validação, como o algoritmo *Early Stopping*, não é capaz de apresentar uma solução com boa capacidade de generalização em tais situações.

Com base nesses resultados, pode-se supor que uma rede treinada a partir do algoritmo *backpropagation* poderia alcançar as soluções contidas no conjunto Pareto-ótimo a partir de uma inicialização dos pesos com norma baixa e especificando um baixo valor para a taxa de aprendizado. Como consequência, o número de iterações para que o algoritmo possa abranger essa região do espaço terá um aumento significativo, aumentando o tempo de treinamento dessas redes. A Figura 3.21 foi obtida a partir de simulações utilizando as condições descritas: pesos inicializados com baixa norma e taxa de aprendizado reduzida.



(a) função seno com ruído



(b) função sinc com ruído

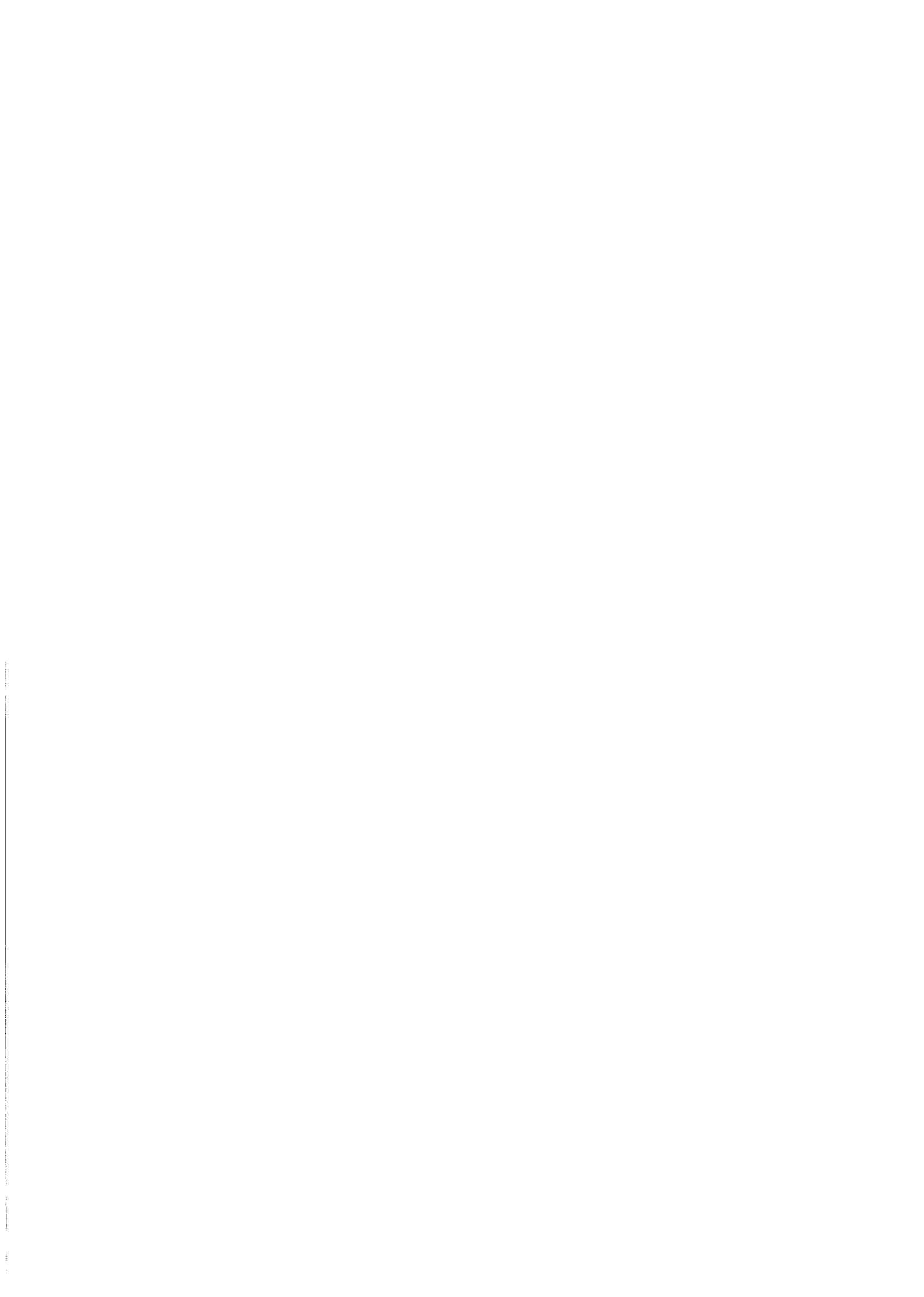
Figura 3.21: Backpropagation com valores reduzidos de norma e taxa de aprendizado

Os resultados demonstram que uma rede inicializada com valores reduzidos de norma e com uma taxa de aprendizado reduzida, pode abranger o conjunto Pareto-ótimo (Figura 3.21 (a)), mas não significa que tal afirmativa seja válida para qualquer problema conforme expresso na Figura 3.21 (b). Tal comportamento justifica o uso de algoritmos Multi-Objetivos, uma vez que promovem a busca específica das soluções contidas no conjunto Pareto-ótimo.

## 3.7 Conclusões do Capítulo

Neste capítulo foi caracterizado o plano de soluções: Erro  $\times$  Norma, utilizado nos métodos multi-objetivos para treinamento de RNAs: MObj/SMC-MObj. Um exemplo utilizando um nodo Perceptron foi demonstrado para avaliar as superfícies do erro e da norma dos pesos bem como o mapeamento das soluções no plano de soluções. A partir dessa análise foi apresentado um método de decomposição dos pesos de uma MLP em coordenadas polares que permite alcançar soluções de mesma norma.

Uma decomposição dos gradientes para um nodo perceptron permitiu a análise do processo de inicialização de seus pesos. Para o nodo perceptron, a inicialização utilizando pesos nulos garante gradientes não nulos. A mesma análise foi realizada para uma rede MLP de duas camadas, sendo obtidas as seguintes conclusões: uma rede MLP não deve ser inicializada com pesos nulos, uma vez que os pesos nulos geram gradientes nulos. Existe um intervalo de confiança que pode ser utilizado para definir valores para os pesos de uma MLP de forma a garantir valores de gradiente superiores a um determinado limiar. A partir desse intervalo é possível analisar a saturação dos pesos e nodos da rede.



# Capítulo 4

## Algoritmo Proposto

Neste capítulo é descrito um novo algoritmo para o treinamento de MLPs. A abordagem do problema é semelhante ao algoritmo Multi-Objetivo, onde o espaço solução é definido em função do Erro (soma dos erros quadráticos) e da Norma (raiz quadrada da soma quadrática dos pesos). A equação de correção dos pesos, apesar de utilizar os gradientes das funções do erro e da norma, implementa a técnica de controle por Modos Deslizantes sendo capaz de realizar um deslocamento no plano de soluções em busca de pontos ou soluções específicas. Dessa forma, verificada a existência do conjunto Pareto-ótimo e definida a região de sua localização (entre os mínimos globais das funções de custo), é possível traçar trajetórias ou pontos-objetivo a serem alcançados durante o treinamento, que permitem obter soluções pertencentes ao conjunto Pareto-ótimo. Uma vez obtidas as soluções eficientes é possível selecionar aquela com a melhor resposta de generalização.

### 4.1 Algoritmo Proposto

Sendo o objetivo do treinamento a obtenção de pontos no plano de soluções definido pelo Erro e pela Norma e, caracterizado o conjunto Pareto-ótimo, é preciso definir a região de busca. O algoritmo de treinamento pode ser escrito em função de um ponto do plano que se deseja alcançar.

Antes de prosseguir com a descrição do algoritmo é necessário especificar as funções que caracterizam o plano de soluções:

$$\text{Erro: } E_{(k)} = \frac{1}{2} \sum_{i=1}^p (d_i - y_{i(k)})^2 \quad (4.1)$$

$$\text{Norma: } \|\mathbf{w}_{(k)}\| = \sqrt{\sum_{ji} (w_{ji(k)})^2} \quad (4.2)$$

onde  $p$  é o número de padrões para treinamento.

Seja um ponto,  $m$ , definido no espaço solução:  $\{m \in \mathfrak{R}^2 : m(E_t, \|\mathbf{w}_t\|)\}$ . Seja o ob-

jetivo do algoritmo proposto alcançar esse ponto  $m$ . A direção de busca para o Erro já é conhecida ( $\frac{\partial E_{(k)}}{\partial w_{ji(k)}}$ ) bem como a direção de busca da norma ( $\frac{\partial \|\mathbf{w}_{(k)}\|}{\partial w_{ji(k)}}$ ). Definindo-se um termo de restrição para o ajuste do Erro da seguinte maneira: o termo deve ser negativo se o erro atual da rede ( $E_{(k)}$ ) for maior do que o do ponto objetivo ( $E_t$ ) (para caminhar na direção de minimização do Erro), positivo se  $E_{(k)}$  for menor que  $E_t$  (para caminhar na direção de maximização) e nulo se  $E_{(k)}$  for igual a  $E_t$ . O termo de ajuste para o Erro pode ser definido, então, pela diferença entre o erro desejado ( $E_t$ ) e o erro atual ( $E_{(k)}$ ):

$$(E_t - E_{(k)}), \text{ ou na forma: } -(E_{(k)} - E_t) \quad (4.3)$$

Para o termo de restrição do ajuste da Norma, a mesma teoria demonstrada anteriormente é válida. Dessa forma, o termo de ajuste para a Norma é definido por:

$$(\|\mathbf{w}_t\|^2 - \|\mathbf{w}_{(k)}\|^2), \text{ ou na forma: } -(\|\mathbf{w}_{(k)}\|^2 - \|\mathbf{w}_t\|^2) \quad (4.4)$$

Definindo o ajuste dos pesos na direção expressa pelos gradientes da função Erro e Norma e acrescentando os termos de restrição representados nas Equações 4.3 e 4.4, a equação final para o ajuste dos pesos de uma MLP é da forma:

$$\Delta w_{ji(k)} = -\alpha \cdot (E_{(k)} - E_t) \cdot \frac{\partial E_{(k)}}{\partial w_{ji(k)}} - \beta (\|\mathbf{w}\|^2 - \|\mathbf{w}_t\|^2) \cdot w_{ji(k-1)} \quad (4.5)$$

onde  $\alpha$  e  $\beta$  são as taxas de aprendizado, ou ganhos do Erro e da Norma, respectivamente.

Uma vez que o termo de ajuste dos pesos foi definido, a função de custo referente à Equação 4.5 pode ser escrito como:

$$J_{(k)} = \frac{1}{4} \left[ \left( \sum_{i=1}^p (d_i - y_{i(k)})^2 \right) - E_t \right]^2 + \frac{1}{4} \left[ \sum_{ji} (w_{ji(k)})^2 - \|\mathbf{w}_t\|^2 \right]^2 \quad (4.6)$$

$$J_{(k)} = \frac{1}{4} (E_{(k)} - E_t)^2 + \frac{1}{4} (\|\mathbf{w}_{(k)}\|^2 - \|\mathbf{w}_t\|^2)^2$$

#### 4.1.1 Algoritmo Proposto utilizando Teoria de Modos Deslizantes

Comparando a equação proposta, o *backpropagation* e a equação do método *weight decay*, a Equação 4.6 é uma equação simples, que utiliza os gradientes das funções do erro e da norma, sem o uso de técnicas para o aumento da velocidade de convergência ou uso de algoritmos mais complexos. Como será discutido adiante, a aplicação de técnicas de otimização conhecidas para o algoritmo *backpropagation* é capaz de aprimorar o desempenho do algoritmo proposto durante o treinamento.

Sira-Ramírez, em seu artigo [Sira-Ramírez and Colina-Morles, 1995], propôs um algoritmo onde o ajuste dos pesos de um nodo adaline é controlado por uma superfície para o erro instantâneo. No método proposto [Costa et al., 2002c], tal teoria é aplicada

às funções do Erro e da Norma apresentadas na Equação 4.6 através do uso de duas superfícies de deslizamento, uma definida para o Erro e outra para a Norma:

$$S_{E(k)} = E(k) - E_t$$

$$S_{\|\mathbf{w}(k)\|} = \|\mathbf{w}(k)\|^2 - \|\mathbf{w}_t\|^2$$

$$sgn(S) = \begin{cases} +1 & \text{se } S > 0 \\ 0 & \text{se } S = 0 \\ -1 & \text{se } S < 0 \end{cases}$$

A equação de ajuste dos pesos não é mais definida pelos termos de restrição do Erro e da Norma e sim pelo seu sinal.

$$\Delta w_{ji(k)} = -\alpha \cdot sgn(S_{E(k)}) \cdot \frac{\partial E(k)}{\partial w_{ji(k)}} - \beta \cdot sgn(S_{\|\mathbf{w}(k)\|}) \cdot w_{ji(k)} \quad (4.7)$$

Apesar da diferença entre as Equações 4.5 e 4.7 ser definida pelo uso dos valores e dos sinais dos termos de restrição, a Equação 4.7 permite a análise segundo a teoria de controle por Modos Deslizantes [Itkis, 1976]. Dessa forma, através dessa técnica, é possível estabelecer intervalos para os ganhos  $\alpha$  e  $\beta$  no qual a convergência para as superfícies, ou para um região próxima, é garantida.

Em seu trabalho, Parma, Menezes e Braga [Parma et al., 1999] descrevem o treinamento de MLPs através do uso da Teoria de Modos Deslizantes aplicados a uma superfície de deslizamento de primeira ordem. Apesar do aumento no número de parâmetros a serem ajustados, a própria teoria utilizada possibilitou a definição de restrições para os ganhos de modo a garantir a convergência da rede. Os resultados obtidos foram comparáveis ao algoritmo de *Levenberg Marquardt*.

## 4.2 Estudo da Convergência das Superfícies de Deslizamento

Uma vez determinada a equação de correção dos pesos de uma rede MLP (Equação 4.7), é possível aplicar a Teoria de Modos Deslizantes [Itkis, 1976] para estabelecer as condições para os ganhos  $\alpha$  e  $\beta$ , e garantir a convergência a partir de um estado inicial da rede a um estado de erro e norma pré-estabelecidos ou próximo deste.

A convergência do estado da rede à superfície de deslizamento pode ser analisada considerando-se a convergência da série:

$$Z = \sum_{k=1}^{\infty} S(k) \quad (4.8)$$

Uma vez garantida a convergência desta série, garante-se que o estado da rede irá convergir, ao menos assintoticamente, para a superfície  $S_{(k)} = 0$ .

A partir do princípio de convergência de Cauchy [Kreyszig, 1993]: a série  $\{S_1 + S_2 + \dots +\}$  converge se e somente se, para um dado valor  $\epsilon > 0$ , puder ser encontrado um valor de  $N$  tal que  $|S_{n+1} + S_{n+2} + \dots + S_{n+p}| < \epsilon$  para todo  $n > N$  e  $p = 1, 2, \dots$ . A série será absolutamente convergente se a série:

$$\hat{Z} = \sum_{k=1}^{\infty} |S_{(k)}| \quad (4.9)$$

for convergente. Para o estudo da convergência da série expressa na Equação 4.9, será realizado o teste da razão [Butkov, 1978], a série  $Z$  é absolutamente convergente se:

$$\left| \frac{S_{(k)}}{S_{(k-1)}} \right| \leq Q < 1 \quad (4.10)$$

A Equação 4.11, obtida a partir da Equação 4.10, representa a condição necessária e suficiente tanto para o regime de deslizamento quanto para a convergência do estado do sistema para a superfície. Entretanto, esta equação não garante que o sistema irá atravessar a superfície, mas garante uma convergência assintótica à superfície de deslizamento.

$$|S_{(k+1)}| < |S_{(k)}| \quad (4.11)$$

A partir da Equação 4.11, é possível especificar as condições de convergência para o problema proposto.

**Teorema 1** Seja  $S_{(k)} : \mathbb{R} \rightarrow \mathbb{R}$ , a superfície de deslizamento definida por  $S_{(k)} = X_{(k)} - X_t$ , onde  $\{X_{(k)}, X_t\} \in \mathbb{R}^+$  ( $X_{(k)} \geq 0$  e  $X_t \geq 0$ ). Se  $X_{(k)}$  é definido como um funcional objetivo e  $X_t$  como o valor desejado para este funcional, então uma condição suficiente de convergência do estado  $X_{(k)}$  para a superfície de deslizamento,  $S_{(k)} = 0$ , é que o sistema deve satisfazer às Equações 4.12 e 4.13, abaixo.

$$\text{sgn}(S_{(k)}) \cdot [X_{(k+1)} - X_{(k)}] < 0 \quad (4.12)$$

$$\text{sgn}(S_{(k)}) \cdot [X_{(k+1)} + X_{(k)} - 2X_t] > 0 \quad (4.13)$$

**DEMONSTRAÇÃO:** Inicialmente, será analisada a condição de Sarpturk (Equação 4.11) para o problema em questão, onde duas situações devem ser estabelecidas:

1. Quando a trajetória converge assintoticamente para a superfície  $S_{(k)} = 0$ , sem atravessá-la.
2. Quando a trajetória converge para  $S_{(k)} = 0$ , atravessando-a.

Na demonstração, as seguintes propriedades serão utilizadas para definir a norma, ou o valor absoluto da superfície de deslizamento:

$$|S| = \begin{cases} +S, & \text{se } S \geq 0 \\ -S, & \text{se } S < 0 \end{cases} \quad (4.14)$$

$$|S| = \text{sgn}(S) \cdot S \quad (4.15)$$

**Situação 1:** A superfície de deslizamento não é atravessada durante a convergência.

Neste caso, a seguinte condição é observada:

$$\text{sgn}(S_{(k+1)}) = \text{sgn}(S_{(k)}) \quad (4.16)$$

Substituindo em 4.11, as relações expressa pelas Equações 4.15 e 4.16, tem-se:

$$\text{sgn}(S_{(k+1)}) \cdot S_{(k+1)} < \text{sgn}(S_{(k)}) \cdot S_{(k)}$$

$$\text{sgn}(S_{(k)}) \cdot [S_{(k+1)} - S_{(k)}] < 0 \quad (4.17)$$

É possível expandir a Equação 4.17, utilizando as relações,  $S_{(k)} = X_{(k)} - X_t$  e  $S_{(k+1)} = X_{(k+1)} - X_t$ , resultando na seguinte condição:

$$\text{sgn}(S_{(k)}) \cdot [X_{(k+1)} - X_{(k)}] < 0$$

que corresponde à condição expressa pela Equação 4.12.

**Situação 2:** A superfície de deslizamento é atravessada durante a convergência.

A condição observada é da forma:

$$\text{sgn}(S_{(k+1)}) = -\text{sgn}(S_{(k)}) \quad (4.18)$$

Partindo da Equação 4.11, a seguinte expressão é obtida:

$$-\text{sgn}(S_{(k)}) \cdot S_{(k+1)} < \text{sgn}(S_{(k)}) \cdot S_{(k)}$$

Logo:

$$-\text{sgn}(S_{(k)}) \cdot [S_{(k+1)} + S_{(k)}] < 0 \quad (4.19)$$

Novamente, expandindo a Equação 4.19, tem-se o seguinte resultado:

$$-sgn(S_{(k)}) \cdot [X_{(k+1)} + X_{(k)} - 2 \cdot X_t] < 0$$

ou na forma:

$$sgn(S_{(k)}) \cdot [X_{(k+1)} + X_{(k)} - 2 \cdot X_t] > 0$$

que corrobora com a segunda condição expressa pela Equação 4.13

■

• **Comentário do Teorema 1:**

É importante observar que, segundo o Teorema 1, ambas as condições expressas pelas Equações 4.12 e 4.13 devem ser satisfeitas para garantir a convergência do método proposto. A utilização do Teorema 1 permitirá, então, a identificação dos intervalos de convergência para os ganhos associados às superfícies  $S_E$  e  $S_{\|w\|}$ .

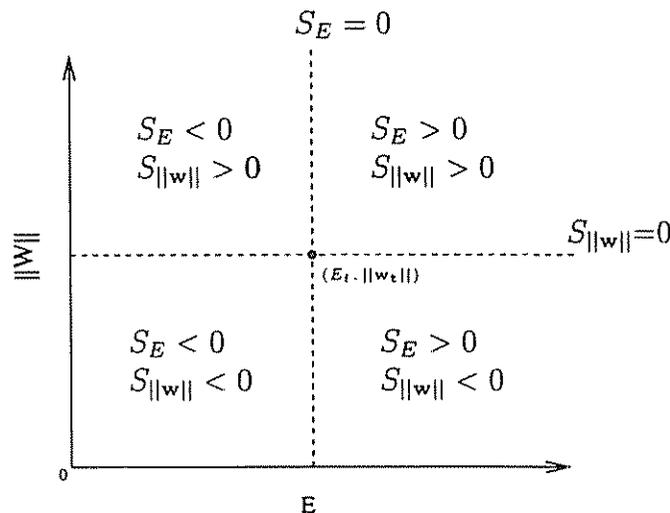


Figura 4.1: Regiões definidas pelas superfícies de deslizamento

Conforme pode ser observado na Figura 4.1, uma vez que o ponto objetivo seja definido, é possível distinguir quatro regiões no plano de soluções. A cada região está associado um determinado comportamento para as superfícies de deslizamento. Se o Teorema 1 for satisfeito para cada superfície em cada região, o método proposto é capaz de convergir para o ponto objetivo.

Uma vez que o método proposto utiliza duas superfícies de deslizamento, a primeira associada ao funcional erro ( $S_E$ ) e a segunda ao funcional norma dos pesos da rede ( $S_{\|w\|}$ ), aplicando-se o Teorema 1 a cada superfície podem ser identificadas as seguintes condições de convergência do método:

– Condições de convergência para a superfície  $S_{\|\mathbf{w}(k)\|} = 0$  :

$$\text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot [ \|\mathbf{w}(k+1)\|^2 - \|\mathbf{w}(k)\|^2 ] < 0 \quad (4.20)$$

$$\text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot [ \|\mathbf{w}(k+1)\|^2 + \|\mathbf{w}(k)\|^2 - 2 \cdot \|\mathbf{w}_t\|^2 ] > 0 \quad (4.21)$$

– Condições de convergência para a superfície  $S_{E(k)} = 0$

$$\text{sgn}(S_{E(k)}) \cdot [ E_{(k+1)} - E_{(k)} ] < 0 \quad (4.22)$$

$$\text{sgn}(S_{E(k)}) \cdot [ E_{(k+1)} + E_{(k)} - 2 \cdot E_t ] > 0 \quad (4.23)$$

Dessa forma, a partir de um estado inicial de erro e norma de uma rede MLP, as seguintes condições de convergência devem ser satisfeitas:

1. Superfície do Erro,  $S_E$  :

(a)  $S_E > 0$  :

$$[ E_{(k+1)} - E_{(k)} ] < 0$$

$$[ E_{(k+1)} + E_{(k)} - 2 \cdot E_t ] > 0$$

(b)  $S_E < 0$  :

$$[ E_{(k+1)} - E_{(k)} ] > 0$$

$$[ E_{(k+1)} + E_{(k)} - 2 \cdot E_t ] < 0$$

2. Superfície da Norma,  $S_{\|\mathbf{w}\|}$  :

(a)  $S_{\|\mathbf{w}\|} > 0$  :

$$[ \|\mathbf{w}(k+1)\|^2 - \|\mathbf{w}(k)\|^2 ] < 0$$

$$[ \|\mathbf{w}(k+1)\|^2 + \|\mathbf{w}(k)\|^2 - 2 \cdot \|\mathbf{w}_t\|^2 ] > 0$$

(b)  $S_{\|\mathbf{w}\|} < 0$  :

$$[ \|\mathbf{w}(k+1)\|^2 - \|\mathbf{w}(k)\|^2 ] > 0$$

$$[ \|\mathbf{w}(k+1)\|^2 + \|\mathbf{w}(k)\|^2 - 2 \cdot \|\mathbf{w}_t\|^2 ] < 0$$

Pode-se observar que os termos referentes aos sinais das superfícies de deslizamento,  $\text{sgn}(S_{\|\mathbf{w}(k)\|})$  e  $\text{sgn}(S_{E(k)})$ , determinam as características associadas às inequações, ou seja, determinam os limites *externos* ou *internos* das regiões de convergência em relação às seguintes composições das funções objetivo:

$$\|\mathbf{w}(k+1)\|^2 - \|\mathbf{w}(k)\|^2 \quad (4.24)$$

$$\|\mathbf{w}(k+1)\|^2 + \|\mathbf{w}(k)\|^2 - 2 \cdot \|\mathbf{w}_t\|^2 \quad (4.25)$$

$$E_{(k+1)} - E_{(k)} \quad (4.26)$$

$$E_{(k+1)} + E_{(k)} - 2.E_t \quad (4.27)$$

Uma vez definidas as condições de convergência, o estudo da convergência das superfícies de deslizamento implica na decomposição das Equações 4.24, 4.25, 4.26 e 4.27 em relação aos ganhos  $\alpha$  e  $\beta$  de forma a obter um conjunto de equações para essas variáveis e, a partir das condições de convergência, determinar um intervalo ou uma região no espaço  $\mathfrak{R}^2$ , definido em função dos ganhos, capaz de garantir a convergência do método proposto (SMC-MOBJ).

**Teorema 2** *Seja definido o funcional  $X_{(k+1)}$ ,  $\{X_{(k)} \geq 0, \forall k\}$ , escrito na forma:*

$$X_{(k+1)} = a_1 \cdot \alpha^2 + a_2 \cdot \alpha\beta + a_3 \cdot \beta^2 + a_4 \cdot \alpha + a_5 \cdot \beta + X_{(k)} \quad (4.28)$$

onde  $\{a_1, a_2, a_3, a_4, a_5\} \in \mathfrak{R}$ . As novas expressões para as condições de convergência, análogas às Equações 4.12 e 4.13, são reescritas como:

$$\text{sgn}(S_{(k)}) \cdot [a_1 \cdot \alpha^2 + a_2 \cdot \alpha\beta + a_3 \cdot \beta^2 + a_4 \cdot \alpha + a_5 \cdot \beta] < 0 \quad (4.29)$$

$$\text{sgn}(S_{(k)}) \cdot [a_1 \cdot \alpha^2 + a_2 \cdot \alpha\beta + a_3 \cdot \beta^2 + a_4 \cdot \alpha + a_5 \cdot \beta + 2 \cdot S_{(k)}] > 0 \quad (4.30)$$

Se as seguintes restrições:

$$a_1 > 0 \quad (4.31)$$

$$a_3 > 0 \quad (4.32)$$

$$a_2^2 - 4 \cdot a_1 a_3 \leq 0 \quad (4.33)$$

são respeitadas, então a existência de uma região para os ganhos  $\alpha$  e  $\beta$  que satisfazem ambas as condições de convergência é garantida.

**DEMONSTRAÇÃO:** Sendo o funcional  $X_{(k+1)}$  uma função quadrática escrita na forma 4.28, os termos referentes às condições de convergência descritos no Teorema 1, Equações 4.12 e 4.13, apresentam a forma:

$$[X_{(k+1)} - X_{(k)}] = a_1 \cdot \alpha^2 + a_2 \cdot \alpha\beta + a_3 \cdot \beta^2 + a_4 \cdot \alpha + a_5 \cdot \beta + X_{(k)} - X_{(k)}$$

$$[X_{(k+1)} + X_{(k)} - 2 \cdot X_t] = a_1 \cdot \alpha^2 + a_2 \cdot \alpha\beta + a_3 \cdot \beta^2 + a_4 \cdot \alpha + a_5 \cdot \beta + X_{(k)} + X_{(k)} - 2 \cdot X_t$$

onde:  $X_{(k)} - X_{(k)} = 0$  e  $X_{(k)} + X_{(k)} - 2.X_t = 2.(X_{(k)} - X_t) = 2.S_{(k)}$ , justificando, assim, as Equações 4.29 e 4.30.

Para analisar as regiões de convergência limitadas pelas condições 4.29 e 4.30 é necessário determinar as bordas dessas regiões. Segundo a restrição 4.33, essas regiões podem ser do tipo elíptico ( $a_2^2 - 4.a_1.a_3 < 0$ ) ou parabólico ( $a_2^2 - 4.a_1.a_3 = 0$ ) (ver Seção A.2).

A forma geral para a equação de 2ª ordem relacionada às condições de convergência pode ser escrita da seguinte maneira:

$$a_1.\alpha^2 + a_2.\alpha\beta + a_3.\beta^2 + a_4.\alpha + a_5.\beta + a_6 \quad (4.34)$$

onde:

$$a_6 = \begin{cases} 0, & \text{Equação 4.29} \\ 2.S_{(k)}, & \text{Equação 4.30} \end{cases}$$

é o termo independente.

Estabelecendo uma relação entre as variáveis de interesse,  $\alpha$  e  $\beta$ :

$$\beta = p.\alpha$$

onde  $p \in \mathbb{R}^+$  ( $\{\alpha > 0, \beta > 0\} \Rightarrow p > 0$ ), é a inclinação da reta associada às variáveis  $\alpha$  e  $\beta$ , obtém-se a seguinte expressão:

$$\alpha^2.(a_1 + a_2.p + a_3.p^2) + \alpha.(a_4 + a_5.p) + a_6 \quad (4.35)$$

O objetivo de tal operação consiste em subdividir a área de interesse ou de estudo da superfície definida pela Equação 4.34 analisando a interseção da superfície com o hiperplano  $\beta = p.\alpha$ .

A forma simplificada da Equação 4.35 pode ser obtida como:

$$a.\alpha^2 + b.\alpha + a_6 \quad (4.36)$$

onde:

$$a = a_1 + a_2.p + a_3.p^2 \quad (4.37)$$

$$b = a_4 + a_5.p \quad (4.38)$$

As restrições 4.31, 4.32 e 4.33 garantem que o coeficiente  $a$  da Equação 4.36 é sempre positivo ou nulo ( $a \geq 0$ ), uma vez que, em relação à variável  $p$ , a Equação 4.37 apresenta o seguinte valor para o  $\Delta$  (ver Seção A.3) das raízes:

$$\Delta = a_2^2 - 4.a_1.a_3 \leq 0$$

onde  $a_3 > 0$  (restrição 4.32) define a concavidade positiva da parábola, ou seja,  $a \geq 0$ , para qualquer valor real da variável  $p$ .

Para determinar as raízes da Equação 4.36, o valor de  $\Delta$  é definido por:

$$\Delta = b^2 - 4.a.a_6$$

fazendo  $a_6 = \text{sgn}(S_{(k)}) \cdot |a_6|$ , pois  $a_6 = 2.S_{(k)}$  e uma vez que  $a \geq 0$ , tem-se:

$$\Delta = b^2 - 4.\text{sgn}(S_{(k)}) \cdot |a| \cdot |a_6| \quad (4.39)$$

Segundo a Equação 4.39, o valor de  $\Delta$  está relacionado ao sinal da superfície de deslizamento,  $S_{(k)}$ . A partir dessa relação, pode-se subdividir a análise utilizando as variações possíveis para  $S_{(k)}$ :

1.  $S_{(k)} > 0$  :

As condições de convergência serão definidas como:

$$(a) \ a_1.\alpha^2 + a_2.\alpha\beta + a_3.\beta^2 + a_4.\alpha + a_5.\beta < 0$$

$$a.\alpha^2 + b.\alpha < 0$$

Raízes:

$$\Delta = b^2$$

$$\alpha = \frac{-b \pm b}{2a}$$

ou na forma:

$$\alpha = \frac{-b}{2a} \pm \left| \frac{b}{2a} \right| \quad (4.40)$$

$$(b) \ a_1.\alpha^2 + a_2.\alpha\beta + a_3.\beta^2 + a_4.\alpha + a_5.\beta + 2.S_{(k)} > 0$$

$$a.\alpha^2 + b.\alpha + 2.S_{(k)} > 0$$

Raízes:

$$\Delta = b^2 - 4.|a| \cdot |a_6| \Rightarrow \Delta < b^2$$

Existem, então duas variações possíveis para  $\Delta$ :

i.  $0 < \Delta < b^2$

Considerando um escalar real positivo:  $\zeta$ , tal que  $\zeta > 1$ , as raízes podem ser escritas na forma:

$$\Delta = \frac{b^2}{\zeta^2}$$

$$\alpha = \frac{-b}{2a} \pm \frac{b}{2a\zeta}$$

ou na forma:

$$\alpha = \frac{-b}{2a} \pm \left| \frac{b}{2a\zeta} \right| \quad (4.41)$$

ii.  $\Delta \leq 0$

Nesse caso, por definição ( $a \geq 0$ ), a parábola  $a.\alpha^2 + b.\alpha + 2.S_{(k)}$  é positiva para qualquer valor real de  $\alpha$ , ou seja, a respectiva condição de convergência é verdadeira para qualquer valor de  $\alpha$ . Não existem raízes reais positivas nessas condições.

A partir das Equações 4.40 e 4.41, pode-se estabelecer a seguinte relação:

$$\left| \frac{b}{2a\zeta} \right| < \left| \frac{b}{2a} \right| \quad (4.42)$$

Observada a relação 4.42 e uma vez que o ponto central dos intervalos de convergência pode ser definido pelo termo:  $\frac{-b}{2a}$  (ver Equações 4.40 e 4.41), sendo os critérios de convergência para  $S_{(k)} > 0$ , é possível traçar um diagrama identificando o intervalo de convergência nessas condições.

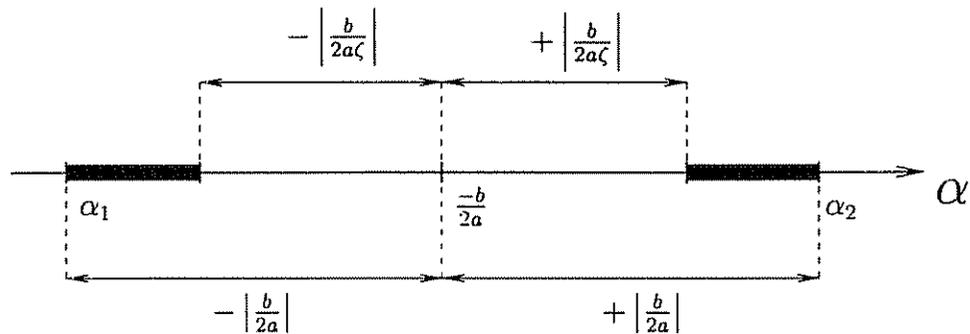


Figura 4.2: Intervalo de convergência para o ganho  $\alpha$  sob a condição:  $S_{(k)} > 0$

A partir da Figura 4.2 é possível identificar o comportamento dos intervalos de convergência para  $S_{(k)} > 0$ . Os intervalos de convergência que garantem a convergência do método SMC-MOBJ são identificados pela região em negrito. A análise realizada garante a existência desse intervalo.

2.  $S_{(k)} < 0$  :

$$(a) a_1.\alpha^2 + a_2.\alpha\beta + a_3.\beta^2 + a_4.\alpha + a_5.\beta > 0$$

$$a.\alpha^2 + b.\alpha > 0$$

Raízes:

$$\Delta = b^2$$

$$\alpha = \frac{-b}{2a} \pm \left| \frac{b}{2a} \right| \quad (4.43)$$

$$(b) \ a_1 \cdot \alpha^2 + a_2 \cdot \alpha \beta + a_3 \cdot \beta^2 + a_4 \cdot \alpha + a_5 \cdot \beta + 2 \cdot S_{(k)} < 0$$

$$a \cdot \alpha^2 + b \cdot \alpha + 2 \cdot S_{(k)} < 0$$

Raízes:  $\text{sgn}(S_{(k)}) = -1$

$$\Delta = b^2 + 4 \cdot |a| \cdot |a_6| \Rightarrow \Delta > b^2$$

$$\Delta = b^2 \cdot \zeta^2 \quad (4.44)$$

A partir da Equação 4.44, é garantida a existência de pelo menos uma raiz real positiva.

$$\alpha = \frac{-b}{2a} \pm \frac{b \cdot \zeta}{2a}$$

ou na forma:

$$\alpha = \frac{-b}{2a} \pm \left| \frac{b \cdot \zeta}{2a} \right| \quad (4.45)$$

A seguinte relação pode ser obtida a partir das Equações 4.43 e 4.45:

$$\left| \frac{b \cdot \zeta}{2a} \right| > \left| \frac{b}{2a} \right| \quad (4.46)$$

Utilizando o mesmo procedimento em relação ao ponto central,  $\frac{-b}{2a}$ , e a partir das Equações 4.43 e 4.45, obtêm-se um diagrama no qual são dispostas as raízes e identificados, segundo os critérios de convergência, os intervalos cujos valores de  $\alpha$  garantem a convergência do método.

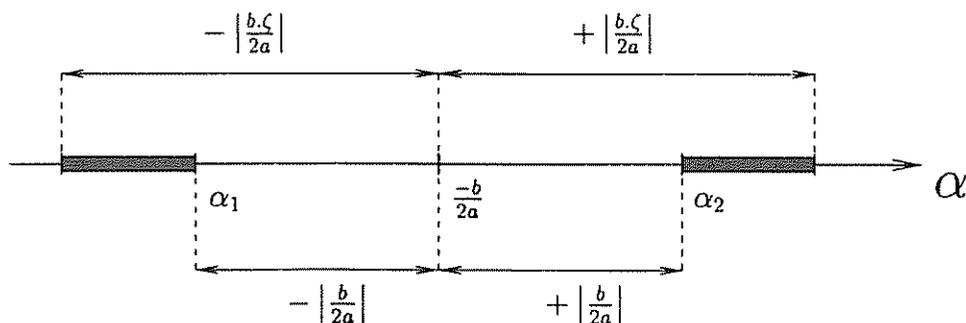


Figura 4.3: Intervalo de convergência para o ganho  $\alpha$  sob a condição:  $S_{(k)} < 0$

Uma vez identificados os intervalos de convergência para o ganho  $\alpha$ , os intervalos em

relação ao ganho  $\beta$  são definidos segundo a relação:  $\beta = p \cdot \alpha$ .

O pontos  $\alpha_1$  e  $\alpha_2$  identificados respectivamente nos limites externos e internos dos intervalos definidos nas Figuras 4.2 e 4.3 indicam as possíveis regiões de interseção com a origem do plano definido pelos ganhos  $\alpha$  e  $\beta$ . Ou seja, de acordo com o sinal do coeficiente  $b$ , um desses pontos estará localizado na origem do plano. ■

• **Comentário do Teorema 2:**

A característica quadrática do funcional  $X_{(k)}$ , definido no Teorema 2, está relacionada à definição dos funcionais Erro ( $E_{(k)}$ ) e Norma ( $\|\mathbf{w}_{(k)}\|$ ). Por definição, esses funcionais são quadráticos em relação ao vetor de parâmetros, ou o vetor de pesos  $\mathbf{w}$ .

$$\|\mathbf{w}_{(k+1)}\|^2 = \sum_{j^i} (w_{j^i(k+1)})^2$$

$$E_{(k+1)} = \frac{1}{2} \sum_{i=1}^p (d_i - y_{i(\mathbf{w}_{(k+1)})})^2$$

Dessa forma, considerando-se a equação de ajuste segundo o método SMC-MOBJ proposto (Equação 4.7), verifica-se que o vetor de pesos no instante  $k+1$  pode ser escrito em função linear dos ganhos,  $\alpha$  e  $\beta$ , e do vetor de pesos no instante  $k$ .

$$w_{j^i(k+1)} = w_{j^i(k)} - \alpha \cdot \text{sgn}(S_{E(k)}) \cdot \frac{\partial E_{(k)}}{\partial w_{j^i(k)}} - \beta \cdot \text{sgn}(S_{\|\mathbf{w}_{(k)}\|}) \cdot w_{j^i(k)}$$

A característica quadrática do funcional  $X_{(k)}$  e a definição do vetor de parâmetros em relação aos ganhos indica que a equação final para o funcional no instante  $k+1$  poderá ser representada por uma relação quadrática, de segundo grau, em relação às variáveis  $\alpha$  e  $\beta$ .

$$X_{(k+1)} = a_1 \cdot \alpha^2 + a_2 \cdot \alpha\beta + a_3 \cdot \beta^2 + a_4 \cdot \alpha + a_5 \cdot \beta + a_6$$

O Teorema 2 garante a existência de intervalos reais para os ganhos  $\alpha$  e  $\beta$  que satisfazem os critérios de convergência (Teorema 1). Entretanto, o teorema não garante, diretamente, a existência de um intervalo positivo para os ganhos. De acordo com o método proposto:  $\alpha > 0$ ,  $\beta > 0$ , como consequência,  $p > 0$ . Para  $S_{(k)} < 0$ , é garantida a existência de pelo menos uma raiz real positiva, o que corrobora a existência de um intervalo positivo para os parâmetros  $\alpha$ ,  $\beta$ . Por outro lado, a existência de soluções positivas quando  $S_{(k)} > 0$  está condicionada à definição dos coeficientes  $a_4$  e  $a_5$  da equação de segundo grau referentes ao funcional  $X_{(k+1)}$ . Para garantir, neste caso, a

existência de um intervalo positivo para os ganhos é necessário garantir que  $b < 0$  (ver Equação 4.38) para  $S_{(k)} > 0$ , o que não é assegurado pelo Teorema 2.

Seja a condição  $S_{(k)} > 0$ ,  $a.\alpha^2 + b.\alpha + 2.S_{(k)} > 0$ , onde  $\Delta = b^2 - 4|a|.|2.S_{(k)}|$ . As raízes podem ser obtidas segundo a Equação 4.41, para  $0 < \Delta < b^2$ :

$$\alpha = \frac{-b}{2a} \pm \left| \frac{b}{2a\zeta} \right|$$

segundo essa equação, a existência de raízes reais positivas está condicionada à existência da condição:

$$\frac{-b}{2a} > 0 \quad (4.47)$$

como  $a \geq 0$ , a condição 4.47 se restringe a:

$$-b > 0 \Rightarrow b < 0, \text{ para } S_{(k)} > 0 \quad (4.48)$$

Uma vez que a condição 4.48 seja satisfeita, em conjunto com o Teorema 2, é garantida a existência de um intervalo positivo para os ganhos  $\alpha$  e  $\beta$ . Entretanto, decompondo-se o termo  $b$ , observa-se que a condição 4.48 está vinculada aos parâmetros  $a_4$ ,  $a_5$  e  $p$ :  $b = a_4 + a_5.p$  (Equação 4.38). Desse modo, mesmo que  $p > 0$ , os parâmetros  $a_4$  e  $a_5$  devem ser analisados em relação às superfícies da norma ( $S_{\|\mathbf{w}\|}$ ) e do erro ( $S_E$ ). Dessa forma, se a condição 4.48 puder ser definida para cada caso, é garantida a existência de regiões positivas para os ganhos.

### 4.2.1 Estudo da Convergência da Função Norma dos Pesos

Seja a equação da norma dos pesos:

$$\|\mathbf{w}_{(k+1)}\|^2 = \sum_{ji} w_{ji(k+1)}^2 \quad (4.49)$$

É possível descrever a seguinte relação para os pesos  $w_{ji(k+1)}$ :

$$w_{ji(k+1)} = w_{ji(k)} + \Delta w_{ji} \quad (4.50)$$

onde  $\Delta w_{ji}$  corresponde ao ajuste dos pesos segundo o método SMC-MOBJ:

$$\Delta w_{ji} = -\alpha.\text{sgn}(S_{E(k)}) \cdot g_{ji(k)} - \beta.\text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot w_{ji(k)} \quad (4.51)$$

$g_{ji(k)}$  é a derivada da função do erro em relação ao peso  $w_{ji(k)}$ , no instante  $k$ .

A norma dos pesos,  $\|\mathbf{w}_{(k+1)}\|^2$ , pode ser descrita em função do valor atual da norma,  $\|\mathbf{w}_{(k)}\|^2$ , e dos ganhos  $\alpha$  e  $\beta$  através da substituição da equação de ajuste dos pesos (Equação 4.51) na Equação 4.50:

$$\sum_{ji} w_{ji(k+1)}^2 = \sum_{ji} [w_{ji(k)} - \alpha \cdot \text{sgn}(S_{E(k)}) \cdot g_{ji(k)} - \beta \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot w_{ji(k)}]^2$$

$$\|\mathbf{w}_{(k+1)}\|^2 = \sum_{ji} [w_{ji(k)}^2 + \alpha^2 \cdot g_{ji(k)}^2 + \beta^2 \cdot w_{ji(k)}^2 - 2 \cdot \alpha \cdot \text{sgn}(S_{E(k)}) \cdot w_{ji(k)} \cdot g_{ji(k)} - 2 \cdot \beta \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot w_{ji(k)}^2 + 2 \cdot \alpha \cdot \beta \cdot \text{sgn}(S_{E(k)}) \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot w_{ji(k)} \cdot g_{ji(k)}]$$
 (4.52)

A partir da Equação 4.52, obtém-se a seguinte expressão:

$$\|\mathbf{w}_{(k+1)}\|^2 = A \cdot \alpha^2 + 2 \cdot B \cdot \text{sgn}(S_{E(k)}) \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot \alpha \cdot \beta + C \cdot \beta^2 - 2 \cdot B \cdot \text{sgn}(S_{E(k)}) \cdot \alpha - 2 \cdot C \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot \beta + C$$
 (4.53)

onde:

$$A = \sum_{ji} g_{ji(k)}^2$$

$$B = \sum_{ji} w_{ji(k)} \cdot g_{ji(k)}$$

$$C = \sum_{ji} w_{ji(k)}^2 = \|\mathbf{w}_{(k)}\|^2$$

são os coeficiente da equação de 2º grau, de duas variáveis ( $\alpha$ ,  $\beta$ ), obtida (Equação 4.53).

Utilizando a Equação 4.53, as condições de convergência da norma (Equações 4.20 e 4.21) podem ser descritas em função dos ganhos  $\alpha$  e  $\beta$ :

$$\|\mathbf{w}_{(k+1)}\|^2 - \|\mathbf{w}_{(k)}\|^2 = A \cdot \alpha^2 + 2 \cdot B \cdot \text{sgn}(S_{E(k)}) \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot \alpha \cdot \beta + C \cdot \beta^2 - 2 \cdot B \cdot \text{sgn}(S_{E(k)}) \cdot \alpha - 2 \cdot C \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot \beta$$
 (4.54)

$$\text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot [A \cdot \alpha^2 + 2 \cdot B \cdot \text{sgn}(S_{E(k)}) \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot \alpha \cdot \beta + C \cdot \beta^2 - 2 \cdot B \cdot \text{sgn}(S_{E(k)}) \cdot \alpha - 2 \cdot C \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot \beta] < 0$$
 (4.55)

$$\begin{aligned} \|\mathbf{w}_{(k+1)}\|^2 + \|\mathbf{w}_{(k)}\|^2 - 2.\|\mathbf{w}_t\|^2 &= A.\alpha^2 + 2.B.\text{sgn}(S_{E(k)}).\text{sgn}(S_{\|\mathbf{w}(k)\|}).\alpha.\beta + C.\beta^2 \\ &\quad - 2.B.\text{sgn}(S_{E(k)}).\alpha - 2.C.\text{sgn}(S_{\|\mathbf{w}(k)\|}).\beta + 2.C - 2.\|\mathbf{w}_t\|^2 \end{aligned} \quad (4.56)$$

onde:  $2.C - 2.\|\mathbf{w}_t\|^2 = 2.S_{\|\mathbf{w}(k)\|}$ .

$$\begin{aligned} \text{sgn}(S_{\|\mathbf{w}(k)\|}). [A.\alpha^2 + 2.B.\text{sgn}(S_{E(k)}).\text{sgn}(S_{\|\mathbf{w}(k)\|}).\alpha.\beta + C.\beta^2 \\ - 2.B.\text{sgn}(S_{E(k)}).\alpha - 2.C.\text{sgn}(S_{\|\mathbf{w}(k)\|}).\beta + 2.S_{\|\mathbf{w}(k)\|}] > 0 \end{aligned} \quad (4.57)$$

As Equações 4.55 e 4.57 descrevem as condições de convergência da superfície de deslizamento da norma e as Equações 4.54 e 4.57, as equações de 2º grau associadas.

#### 4.2.1.1 A natureza das equações de convergência da superfície da Norma

Seja a forma simplificada segundo a equação de 2ª ordem para as condições de convergência (Equação 4.34) da superfície da norma:

$$a_1.\alpha^2 + a_2.\alpha\beta + a_3.\beta^2 + a_4.\alpha + a_5.\beta + a_6$$

onde:

$$\begin{aligned} a_1 &= \sum_{ji} g_{ji(k)}^2 \\ a_2 &= 2.\text{sgn}(S_{E(k)}).\text{sgn}(S_{\|\mathbf{w}(k)\|}). \sum_{ji} w_{ji(k)}.g_{ji(k)} \\ a_3 &= \|\mathbf{w}_{(k)}\|^2 \\ a_4 &= -2.\text{sgn}(S_{E(k)}). \sum_{ji} w_{ji(k)}.g_{ji(k)} \\ a_5 &= -2.\text{sgn}(S_{\|\mathbf{w}(k)\|}).\|\mathbf{w}_{(k)}\|^2 \\ a_6 &= \begin{cases} 0, & \text{Condição 4.29} \\ 2.S_{\|\mathbf{w}(k)\|}, & \text{Condição 4.30} \end{cases} \end{aligned}$$

A partir da análise dos coeficientes pode-se concluir que:  $a_1 > 0$  e  $a_2 > 0$ .

Em relação ao indicador  $I = a_2^2 - 4.a_1.a_3$  (Seção A.2), este apresenta a seguinte característica:

$$\begin{aligned}
 a_2^2 &= (2 \cdot \text{sgn}(S_{E(k)}) \cdot \text{sgn}(S_{\|w(k)\|}) \cdot \sum_{ji} w_{ji(k)} \cdot g_{ji(k)})^2 \\
 a_2^2 &= 4 \cdot \left( \sum_{ji} w_{ji(k)} \cdot g_{ji(k)} \right)^2 \\
 I &= 4 \cdot \left( \sum_{ji} w_{ji(k)} g_{ji(k)} \right)^2 - 4 \cdot \sum_{ji} g_{ji(k)}^2 \cdot \sum_{ji} w_{ji(k)}^2 \quad (4.58)
 \end{aligned}$$

Para a análise da Equação 4.58, será utilizada uma notação vetorial. Sejam os vetores  $w_{\vec{k}}$  e  $g_{\vec{k}}$ , e o produto interno entre dois vetores definido da seguinte forma:

$$w_{\vec{k}} \cdot g_{\vec{k}} = |w_{\vec{k}}| \cdot |g_{\vec{k}}| \cdot \cos\theta \quad (4.59)$$

onde  $\theta$  corresponde ao menor ângulo compreendido entre os vetores  $w_{\vec{k}}$  e  $g_{\vec{k}}$ , e  $|g_{\vec{k}}|$ , a norma do vetor  $g_{\vec{k}}$ . O produto interno entre os dois vetores pode ser ainda calculado segundo a soma do produto de seus elementos:

$$w_{\vec{k}} \cdot g_{\vec{k}} = \sum_n^N w_{n(k)} \cdot g_{n(k)} \quad (4.60)$$

onde  $N$  é a dimensão dos vetores.

A partir das equações 4.59 e 4.60 é possível estabelecer as seguintes relações:

$$\begin{aligned}
 \sum_{ji} g_{ji(k)}^2 &= g_{\vec{k}} \cdot g_{\vec{k}} = |g_{\vec{k}}| \cdot |g_{\vec{k}}| \cdot \cos 0 \\
 \sum_{ji} w_{ji(k)}^2 &= w_{\vec{k}} \cdot w_{\vec{k}} = |w_{\vec{k}}| \cdot |w_{\vec{k}}| \cdot \cos 0 \\
 \sum_{ji} w_{ji(k)} g_{ji(k)} &= w_{\vec{k}} \cdot g_{\vec{k}} = |w_{\vec{k}}| \cdot |g_{\vec{k}}| \cdot \cos\theta
 \end{aligned}$$

Substituindo essa relação na Equação 4.58, sendo  $\cos 0 = 1$ , pode-se determinar o seguinte indicador:

$$\begin{aligned}
 I &= 4 \cdot \left( \sum_{ji} w_{ji(k)} g_{ji(k)} \right)^2 - 4 \cdot \sum_{ji} g_{ji(k)}^2 \cdot \sum_{ji} w_{ji(k)}^2 \\
 I &= 4 \cdot \left[ (|w_{\vec{k}}| \cdot |g_{\vec{k}}| \cdot \cos\theta)^2 - |w_{\vec{k}}|^2 \cdot |g_{\vec{k}}|^2 \right] \\
 I &= 4 \cdot \left[ |w_{\vec{k}}|^2 \cdot |g_{\vec{k}}|^2 \cdot \cos^2\theta - |w_{\vec{k}}|^2 \cdot |g_{\vec{k}}|^2 \right] \quad (4.61)
 \end{aligned}$$

Sendo os valores possíveis para o termo  $\cos^2\theta$  compreendidos no intervalo  $[0, 1]$ , o indicador apresenta o seguinte comportamento:

$$I \leq 0 \quad (4.62)$$

que corresponde à equação de um parabolóide quando  $\cos^2\theta = 1$  ( $I = 0$ ,  $w_{(k)} = g_{(k)}$  ou  $w_{(k)} = -g_{(k)}$ ) ou de um elipsóide quando  $\cos^2\theta \neq 1$  ( $I < 0$ ).

Uma vez que, para a superfície da norma,  $S_{\|\mathbf{w}(k)\|}$ , os coeficientes  $a_1$  e  $a_3$  são positivos e o indicador  $I \leq 0$ , a partir do Teorema 2 é garantida a existência de uma região de convergência para os ganhos  $\alpha$  e  $\beta$ . Seja ainda a seguinte situação:

- $S_{\|\mathbf{w}(k)\|} > 0$

$$b = a_4 + a_5 p$$

$$p > 0$$

em relação à equação de convergência da função norma, tem-se:

$$a_4 = -2 \cdot \text{sgn}(S_{E(k)}) \cdot \sum_{j^i} w_{ji(k)} \cdot g_{ji(k)}$$

$$a_5 = -2 \cdot \|\mathbf{w}_{(k)}\|^2, \quad (a_5 < 0)$$

Para garantir, nesse caso, a existência de uma região positiva para os ganhos é necessário que  $b < 0$ . Sejam então analisadas as seguintes situações:

1.  $a_4 < 0$  :

Nesse caso,  $b < 0$  uma vez que  $a_5 \cdot p < 0$ .

2.  $a_4 > 0$  :

Uma vez que  $a_5 < 0$ , pode-se estimar um intervalo para o parâmetro  $p$  de forma a garantir a condição  $b < 0$ .

$$\begin{aligned} |a_5 \cdot p| &> |a_4| \\ -a_5 \cdot p &> a_4 \\ p &> \frac{a_4}{-a_5} \end{aligned} \quad (4.63)$$

Observado o intervalo definido pela Equação 4.63 sob as condições:  $S_{\|\mathbf{w}(k)\|} > 0$ ,  $a_4 > 0$ , é garantida a existência de uma região positiva no espaço  $\mathcal{R}^2$  dos ganhos ( $\alpha \times \beta$ ) que garante a convergência da função norma dos pesos para a respectiva superfície de deslizamento.  $S_{\|\mathbf{w}(k)\|}$ .

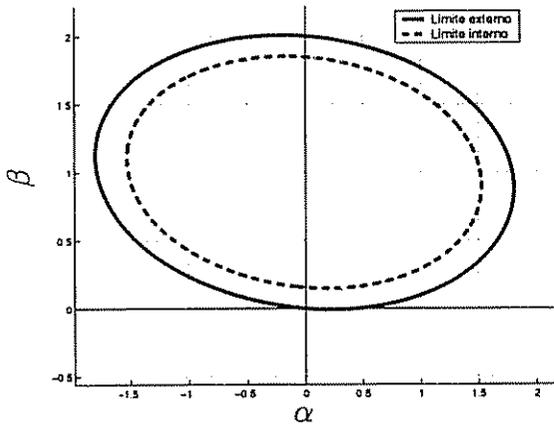


Figura 4.4: Região de convergência para  $S_{||w(k)||} > 0$

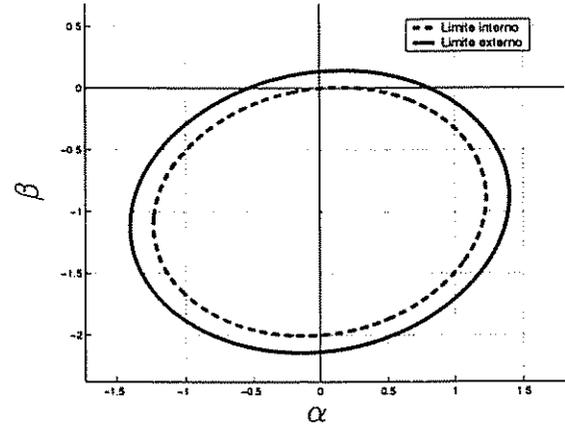


Figura 4.5: Região de convergência para  $S_{||w(k)||} < 0$

As Figuras 4.4 e 4.5 exemplificam os limites que definem a região de convergência para os ganhos  $\alpha$  e  $\beta$ , para  $S_{||w(k)||} > 0$  e  $S_{||w(k)||} < 0$ , respectivamente. Os resultados foram obtidos para uma rede MLP, sendo os limites calculados em relação a um determinado padrão de entrada. Como pode ser observado, os limites externos e internos são representados por elipses.

Os coeficientes das condições de convergência para a superfície da norma são definidos em função do vetor de pesos  $w$  e do gradiente,  $g$  no ponto. Ou seja, os coeficientes variam de acordo com as condições atuais dos pesos. Essa característica indica que as regiões de convergência também variam de acordo com o vetor de pesos, ou à medida que este se desloca no espaço de soluções. Dessa forma, redes MLPs podem apresentar um mesmo valor para o funcional erro e norma mas, regiões distintas de convergência para os ganhos  $\alpha$  e  $\beta$ .

### 4.2.2 Estudo da Convergência da Função do Erro

Seja a função do Erro definido pela soma quadrática da diferença entre as saídas desejadas,  $d_m$ , e as saídas de uma rede MLP,  $y_m$ , em relação a um padrão ou vetor de entrada:

$$E_{(k)} = \frac{1}{2} \sum_{m=1}^Q (d_m - y_{m(k)})^2 \quad (4.64)$$

onde  $Q$  é o número de saídas da rede.

Expandindo a Equação 4.64, tem-se:

$$E_{(k)} = \frac{1}{2} \sum_{m=1}^Q (d_m^2 - 2.d_m.y_{m(k)} + y_{m(k)}^2) \quad (4.65)$$

A mesma propriedade pode ser aplicada em relação à função do erro no próximo passo,  $E_{(k+1)}$ :

$$E_{(k+1)} = \frac{1}{2} \sum_{m=1}^Q (d_m^2 - 2 \cdot d_m \cdot y_{m(k+1)} + y_{m(k+1)}^2) \quad (4.66)$$

A partir das Equações 4.65 e 4.66 pode-se expandir os termos definidos pelas Equações 4.22 e 4.23:

•  $[E_{(k+1)} - E_{(k)}]$  :

$$\begin{aligned} [E_{(k+1)} - E_{(k)}] &= \frac{1}{2} \sum_{m=1}^Q (d_m^2 - 2 \cdot d_m \cdot y_{m(k+1)} + y_{m(k+1)}^2) \\ &\quad - \frac{1}{2} \sum_{m=1}^Q (d_m^2 - 2 \cdot d_m \cdot y_{m(k)} + y_{m(k)}^2) \end{aligned}$$

$$[E_{(k+1)} - E_{(k)}] = \frac{1}{2} \sum_{m=1}^Q (-2 \cdot d_m \cdot y_{m(k+1)} + y_{m(k+1)}^2 + 2 \cdot d_m \cdot y_{m(k)} - y_{m(k)}^2)$$

$$[E_{(k+1)} - E_{(k)}] = \frac{1}{2} \sum_{m=1}^Q \{2 \cdot d_m \cdot (y_{m(k)} - y_{m(k+1)}) + y_{m(k+1)}^2 - y_{m(k)}^2\} \quad (4.67)$$

•  $[E_{(k+1)} + E_{(k)} - 2 \cdot E_t]$  :

$$\begin{aligned} [E_{(k+1)} + E_{(k)} - 2 \cdot E_t] &= \frac{1}{2} \sum_{m=1}^Q (d_m^2 - 2 \cdot d_m \cdot y_{m(k+1)} + y_{m(k+1)}^2) \\ &\quad + \frac{1}{2} \sum_{m=1}^Q (d_m^2 - 2 \cdot d_m \cdot y_{m(k)} + y_{m(k)}^2) - 2 \cdot E_t \end{aligned}$$

$$\begin{aligned} [E_{(k+1)} + E_{(k)} - 2 \cdot E_t] &= \frac{1}{2} \sum_{m=1}^Q (d_m^2 - 2 \cdot d_m \cdot y_{m(k+1)} + y_{m(k+1)}^2 + d_m^2 \\ &\quad - 2 \cdot d_m \cdot y_{m(k)} + y_{m(k)}^2) - 2 \cdot E_t \end{aligned}$$

$$[E_{(k+1)} + E_{(k)}] = \frac{1}{2} \sum_{m=1}^Q \{2 \cdot d_m^2 - 2 \cdot d_m \cdot (y_{m(k)} + y_{m(k+1)}) + y_{m(k+1)}^2 + y_{m(k)}^2\} - 2 \cdot E_t \quad (4.68)$$

A partir das Equações 4.67 e 4.68, pode-se identificar o termo  $y_{m(k+1)}$  como sendo o objeto de estudo através do qual deseja-se estabelecer uma relação entre os ganhos  $\alpha$  e  $\beta$ .

**Corolário 1** *Seja a seguinte equação para a saída, no instante  $k + 1$ , de um nodo  $m$  de uma rede MLP com  $Q$  saídas:*

$$y_{m(k+1)} = y_{m(k)} + a_m \alpha + b_m \beta \quad (4.69)$$

onde  $a_m$  e  $b_m$  são, respectivamente, os coeficientes reais das variáveis  $\alpha$  e  $\beta$  em relação à saída  $m$ :  $\{a_m, b_m, y_{m(k+1)}, y_{m(k)}\} \in \mathbb{R}$ . Então, as condições de convergência da superfície do Erro,  $S_{E(k)}$ , definidas pelas Equações 4.22 e 4.23, podem ser escritas na seguinte forma:

$$\text{sgn}(S_{E(k)}) \cdot \left[ \frac{1}{2} \sum_{m=1}^Q \{a_m^2 \alpha^2 + 2 a_m b_m \alpha \beta + b_m^2 \beta^2 - 2 a_m e_{m(k)} \alpha - 2 b_m e_{m(k)} \beta\} \right] < 0 \quad (4.70)$$

$$\begin{aligned} \text{sgn}(S_{E(k)}) \cdot \left[ \frac{1}{2} \sum_{m=1}^Q \{a_m^2 \alpha^2 + 2 a_m b_m \alpha \beta + b_m^2 \beta^2 - 2 a_m e_{m(k)} \alpha - 2 b_m e_{m(k)} \beta + 2 e_{m(k)}^2\} \right. \\ \left. - 2 E_t \right] > 0 \end{aligned} \quad (4.71)$$

onde  $e_{m(k)} = d_m - y_{m(k)}$ .

**DEMONSTRAÇÃO:** A partir da Equação 4.69, o termo  $y_{m(k+1)}^2$  pode ser escrito como:

$$y_{m(k+1)}^2 = a_m^2 \alpha^2 + 2 a_m b_m \alpha \beta + b_m^2 \beta^2 + 2 a_m y_{m(k)} \alpha + 2 b_m y_{m(k)} \beta + y_{m(k)}^2 \quad (4.72)$$

Substituindo as Equações 4.69 e 4.74 nas Equações 4.67 e 4.68 referentes aos termos de convergência (Teorema 1) da superfícies do erro,  $S_{E(k)}$ , obtém-se os seguintes termos:

$$[E_{(k+1)} - E_{(k)}] = \frac{1}{2} \sum_{m=1}^Q \{a_m^2 \alpha^2 + 2 a_m b_m \alpha \beta + b_m^2 \beta^2 - 2 a_m e_{m(k)} \alpha - 2 b_m e_{m(k)} \beta\}$$

$$\begin{aligned} [E_{(k+1)} + E_{(k)} - 2 E_t] = \frac{1}{2} \sum_{p=1}^Q \{a_p^2 \alpha^2 + 2 a_p b_p \alpha \beta + b_p^2 \beta^2 - 2 a_p e_{p(k)} \alpha - 2 b_p e_{p(k)} \beta \\ + 2 e_{p(k)}^2\} - 2 E_t \end{aligned}$$

presentes nas Equações 4.70 e 4.71.

■

A forma simplificada para as equações de 2ª ordem definidas nas Equações 4.70 e 4.71, pode ser escrita na forma:

$$\text{sgn}(S_{E(k)}) \cdot [a_1 \cdot \alpha^2 + a_2 \cdot \alpha \cdot \beta + a_3 \cdot \beta^2 + a_4 \cdot \alpha + a_5 \cdot \beta + a_6] \quad (4.73)$$

com os respectivos coeficientes:

$$\begin{aligned} a_1 &= \frac{1}{2} \sum_{m=1}^Q a_m^2 & a_2 &= \sum_{m=1}^Q a_m \cdot b_m \\ a_3 &= \frac{1}{2} \sum_{m=1}^Q b_m^2 & a_4 &= - \sum_{m=1}^Q a_m \cdot e_{m(k)} \\ a_5 &= - \sum_{m=1}^Q b_m \cdot e_{m(k)} & a_6 &= \begin{cases} 0, & \text{para Condição 4.70} \\ 2 \cdot S_{E(k)} & \text{para Condição 4.71} \end{cases} \end{aligned} \quad (4.74)$$

$$\text{onde } S_{E(k)} = \frac{1}{2} \left( \sum_{p=1}^Q e_{p(k)}^2 \right) - E_t.$$

#### • Comentário do Corolário 1:

O objetivo do Corolário 1 é descrever uma metodologia para a determinação dos intervalos de convergência para a superfícies do erro,  $S_{E(k)}$ , garantindo a existência dos mesmos uma vez que o Teorema 2 pode ser aplicado junto ao desenvolvimento deste corolário.

Mesmo que o Corolário 1 seja aplicado à superfície do Erro, a mesma está relacionada à topologia utilizada ou à saída  $y_m$ . Portanto, para verificar a existência de uma região de convergência positiva para os ganhos  $\alpha$  e  $\beta$ , é necessário analisar o comportamento das condições de convergência para as diversas topologias de MLP em questão. São essas: o perceptron linear, o perceptron não-linear, a rede MLP de duas camadas com saída linear.

#### 4.2.2.1 O Perceptron Linear

Seja a equação de saída de um nodo perceptron com função de ativação linear:

$$y = \sum_{j=1}^N x_j \cdot w_j \quad (4.75)$$

onde  $N$  é o número de entradas do nodo. A análise para a determinação dos intervalos dos ganhos será realizada para cada padrão de entrada (treinamento convencional).

O gradiente para a correção dos pesos é calculado da seguinte forma:

$$g_{j(k)} = \frac{\partial E(k)}{\partial w_{j(k)}} = 2 \cdot \frac{1}{2} \cdot (d - y) \cdot (-1) \cdot \frac{\partial y}{\partial w_{j(k)}}$$

$$g_{j(k)} = \frac{\partial E_{(k)}}{\partial w_{j(k)}} = x_j$$

$$g_{j(k)} = -e_{(k)} \cdot x_{j(k)}$$

onde:  $e_{(k)} = d - y_{(k)}$ .

A saída do nodo no instante  $k + 1$  é da forma:

$$y_{(k+1)} = \sum_{j=1}^N x_j \cdot w_{j(k+1)}$$

onde:

$$w_{j(k+1)} = w_{j(k)} + \Delta w_{j(k)}$$

Reescrevendo a equação de saída do nodo tem-se:

$$y_{(k+1)} = \sum_{j=1}^N x_j \cdot (w_{j(k)} + \Delta w_{j(k)}) \quad (4.76)$$

Uma vez que  $\Delta w_{j(k)}$  representa o ajuste segundo o método SMC-MOBJ (ver Equação 4.7) e conhecida a equação do gradiente do nodo, pode-se expandir a Equação 4.76:

$$\Delta w_{j(k)} = -\alpha \cdot \text{sgn}(S_{E(k)}) \cdot g_{j(k)} - \beta \cdot \text{sgn}(S_{\|w_{(k)}\|}) \cdot w_{j(k)}$$

$$\Delta w_{j(k)} = +\alpha \cdot \text{sgn}(S_{E(k)}) \cdot e_{(k)} \cdot x_{j(k)} - \beta \cdot \text{sgn}(S_{\|w_{(k)}\|}) \cdot w_{j(k)}$$

$$y_{(k+1)} = \sum_{j=1}^N x_j \cdot \left( w_{j(k)} + \alpha \cdot \text{sgn}(S_{E(k)}) \cdot e_{(k)} \cdot x_{j(k)} - \beta \cdot \text{sgn}(S_{\|w_{(k)}\|}) \cdot w_{j(k)} \right) \quad (4.77)$$

Simplificando a Equação 4.77:

$$y_{(k+1)} = y_{(k)} + \alpha \cdot \text{sgn}(S_{E(k)}) \cdot e_{(k)} \sum_{j=1}^N (x_{j(k)}^2) - \beta \cdot \text{sgn}(S_{\|w_{(k)}\|}) \sum_{j=1}^N (x_j \cdot w_{j(k)}) \quad (4.78)$$

Aplicando-se o Corolário 1 à Equação 4.78, os coeficientes das equações de convergência do Erro (Equações 4.73 e 4.74) são definidos da seguinte forma:

$$\begin{aligned}
a_1 &= \frac{1}{2} e_{(k)}^2 \cdot \left( \sum_{j=1}^N x_{j(k)}^2 \right)^2 & a_2 &= -y_{(k)} \cdot \text{sgn}(S_{\|w_{(k)}\|}) \cdot \text{sgn}(S_{E(k)}) \cdot e_{(k)} \cdot \sum_{j=1}^N x_{j(k)}^2 \\
a_3 &= \frac{1}{2} y_{(k)}^2 & a_4 &= -\text{sgn}(S_{E(k)}) \cdot e_{(k)}^2 \cdot \sum_{j=1}^N x_{j(k)}^2 \\
a_5 &= \text{sgn}(S_{\|w_{(k)}\|}) \cdot y_{(k)} \cdot e_{(k)} & a_6 &= \begin{cases} 0, & \text{para Condição 4.70} \\ 2 \cdot S_{E(k)} & \text{para Condição 4.71} \end{cases}
\end{aligned} \tag{4.79}$$

A partir dos coeficientes, verifica-se que expressões para as condições de convergência e as restrições definidas pelo Teorema 2 se aplicam a esse caso garantindo a existência de uma região de convergência para os ganhos e, segundo a propriedade A.2.1 (Apêndice), conclui-se que as equações de convergência são do tipo parabólico ( $I = 0$ ).

Para determinar a existência de uma região positiva para os ganhos é necessário analisar a situação especificada pela Equação 4.48:

- $b < 0$ , para  $S_{E(k)} > 0$  ( $\text{sgn}(S_{E(k)}) = +1$ ) :

$$\begin{aligned}
b &= a_4 + a_5 \cdot p \\
a_4 &= -e_{(k)}^2 \cdot \sum_{j=1}^N x_{j(k)}^2 \quad (a_4 < 0)
\end{aligned}$$

A condição que garante a existência de um intervalo positivo para os ganhos é da forma:

$$\text{Se } a_5 > 0, \quad \text{então: } p < \frac{-a_4}{a_5} \tag{4.80}$$

#### 4.2.2.2 O Perceptron Não-Linear

A Equação 4.81 mostra a saída característica desse tipo de estrutura

$$y = f \left( \sum_{j=1}^N x_j \cdot w_j \right) \tag{4.81}$$

onde  $f(\cdot)$  corresponde à função de ativação do nodo, contínua e diferenciável.

Nesta seção serão apresentadas as condições de convergência para o nodo perceptron utilizando o treinamento segundo o método SMC-MOBI proposto. Serão apresentadas duas abordagens possíveis para o problema. A primeira abordagem consiste na propagação do sinal de saída do nodo para a saída linear do mesmo. Tal procedimento é obtido através da *inversa* da função de ativação  $f(u)$  ( $u = \sum_{j=1}^N x_j \cdot w_j$ ). A partir de tal procedimento a análise de convergência utilizada é semelhante à descrita para o nodo linear (Seção 4.79).

A segunda proposta utiliza uma aproximação da função de ativação através da sua decomposição em Série de Taylor. O uso dos primeiros termos da série de Taylor limita a capacidade de aproximação da função para regiões próximas do ponto de interesse mas, proporcionam uma significativa simplificação e redução do custo matemático para a definição dos intervalos de convergência.

### 1. Retropropagação de Sinal em um Nodo Perceptron

Seja o nodo perceptron apresentado na Figura 4.6

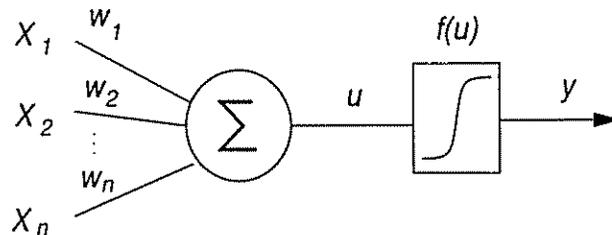


Figura 4.6: Nodo perceptron com Função de Ativação  $f(u)$

onde:

$$y = f(u)$$

$$u = \sum_{j=1}^N x_j \cdot w_j$$

É possível definir a seguinte função inversa para a função  $f(u)$ :

$$\begin{cases} \text{Se: } y = f(u) \\ \text{Então: } u = f^{-1}(y) \end{cases} \quad (4.82)$$

onde  $f^{-1}(y)$  é a função inversa de  $f(u)$ . Para que a Equação 4.82 seja válida é preciso que a função  $f(u)$  seja uma função unimodal, ou seja, associada a cada entrada  $u$ , existe apenas uma única saída  $y$  e vice-versa, ou seja, a uma saída específica seja associada uma única entrada. Outro fator importante para a aplicação da função inversa ao conjunto de treinamento de um nodo perceptron consiste no fato de que os dados a serem *transformados* devem estar definidos no intervalo da Imagem da função  $f(u)$ .

A Tabela 4.1 apresenta as funções inversas das funções de ativação *tansig* e *logsig* normalmente utilizadas como funções de ativação do nodo perceptron.

Tabela 4.1: Funções de ativação e suas inversas

| Função de Ativação | Equação                              | Domínio                 | Imagem        | Inversa   |
|--------------------|--------------------------------------|-------------------------|---------------|---|
| tansig             | $y = \frac{2}{1+e^{-2 \cdot u}} - 1$ | $-\infty < u < +\infty$ | $-1 < y < +1$ | $u = -\frac{1}{2} \ln \left( \frac{2}{y+1} - 1 \right)$ |
| logsig             | $y = \frac{1}{1+e^{-u}}$             | $-\infty < u < +\infty$ | $0 < y < +1$  | $u = -\ln \left( \frac{1}{y} - 1 \right)$               |

## 2. Decomposição da Função de Ativação em Série de Taylor

Uma segunda opção para a análise e definição de intervalos de convergência, para um nodo que apresenta uma função de ativação não-linear, consiste na decomposição da função de ativação em série de Taylor.

Seja a expansão em série de Taylor de uma função  $f(x)$  no ponto  $x_o + \Delta x$ :

$$f_{(x_o+\Delta x)} = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_o)}{n!} (\Delta x)^n \quad (4.83)$$

onde  $\Delta x = x - x_o$ .

É possível estimar uma aproximação da função desejada utilizando apenas alguns termos da série, como uma aproximação de primeira ordem:

$$f_{(x_o+\Delta x)} = f(x_o) + f'(x_o) \cdot \Delta x \quad (4.84)$$

Para a aproximação expressa na Equação 4.84 existe um erro associado que pode ser minimizado com a redução do valor de  $\Delta x$ . Este erro depende da complexidade da função a ser aproximada. Sucintamente, a aproximação de uma função utilizando uma decomposição de primeira ordem é descrita na Equação 4.85

$$f_{(x_o+\Delta x)} = f(x_o) + f'(x_o) \cdot \Delta x \quad (4.85)$$

$$\text{para } |\Delta x| \leq \xi$$

onde  $\xi$  é um valor positivo que representa o intervalo no qual a aproximação é válida. O valor de  $\xi$  varia dependendo do tipo de função a ser aproximada e do erro de aproximação desejado.

A partir da Tabela 4.2 é possível identificar os valores dos erros médios de aproximação da função tangente hiperbólica bem como os valores de  $\xi$  utilizados.

Tabela 4.2: Erros médios e intervalos de confiança para a aproximação da função Tangente Hiperbólica utilizando a expansão de primeira ordem em Série de Taylor

| $\xi$      | 0.001    | 0.01     | 0.05    | 0.1     | 0.2     | 0.5     | 1.0     | 1.5     |
|------------|----------|----------|---------|---------|---------|---------|---------|---------|
| Erro médio | 1.33e-14 | 1.33e-10 | 8.33e-8 | 1.33e-6 | 2.12e-5 | 8.01e-4 | 1.15e-2 | 4.92e-2 |

A análise da convergência do nodo perceptron com função de ativação não linear será descrito a seguir uma vez que, utilizando o método de retropropagação do sinal de saída, a análise é semelhante ao nodo linear.

Seja a saída de um nodo perceptron com uma função de ativação não-linear:

$$y^{(k)} = f \left( \sum_j x_j \cdot w_{j(k)} \right)$$

A saída no instante  $k + 1$ , no qual os pesos foram ajustados, pode ser descrita na forma:

$$y^{(k+1)} = f \left[ \sum_j x_j \cdot (w_{j(k)} + \Delta w_{j(k)}) \right] \quad (4.86)$$

Aplicando-se a decomposição de primeira ordem em série de Taylor na Equação 4.86 obtém-se:

$$y^{(k+1)} = f \left( \sum_j x_j \cdot w_{j(k)} \right) + f' \left( \sum_j x_j \cdot w_{j(k)} \right) \cdot \sum_j x_j \cdot \Delta w_{j(k)} \quad (4.87)$$

onde:

$$\left| \sum_j x_j \cdot \Delta w_{j(k)} \right| \leq \xi$$

Através da decomposição do valor de  $\Delta w_{j(k)}$  e, assumindo a relação  $\beta = p \cdot \alpha$ , é possível definir o seguinte intervalo para o ganho  $\alpha$  em relação à expansão em série de Taylor:

$$\alpha \leq \frac{\xi}{\left| \text{sgn}(S_{E(k)}) \cdot \sum_j x_j \cdot g_{j(k)} + p \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot \sum_j x_j \cdot w_{j(k)} \right|} \quad (4.88)$$

A Equação 4.88 restringe o intervalo dos ganhos dos parâmetros  $\alpha$  e  $\beta$  ( $\beta = p \cdot \alpha$ ) de acordo com a precisão desejada ( $\xi$ ) para a aproximação da função de ativação do nodo.

Decompondo-se a Equação 4.87:

$$y^{(k+1)} = y^{(k)} + y'_{(k)} \cdot \left[ \sum_j x_j \left( -\alpha \cdot \text{sgn}(S_{E(k)}) \cdot g_{j(k)} - \beta \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot w_{j(k)} \right) \right]$$

onde  $g_j(k) = -e(k) \cdot y'(k) \cdot x_j$ , representa a derivada do erro em relação ao parâmetro de ajuste,  $w_j(k)$ .

$$y_{(k+1)} = y_{(k)} - \alpha \cdot \text{sgn}(S_{E(k)}) \cdot y'(k) \cdot \sum_j x_j \cdot g_j(k) - \beta \cdot \text{sgn}(S_{\|w(k)\|}) \cdot y'(k) \cdot \sum_j x_j \cdot w_j(k) \quad (4.89)$$

Aplicando-se o Corolário 1 à Equação 4.89. os seguintes coeficientes referentes às condições de convergência da superfície do erro são definidos:

$$\begin{aligned} a_1 &= +\frac{1}{2} \left( y'(k) \cdot \sum_j x_j \cdot g_j(k) \right)^2 \\ a_2 &= +y'(k)^2 \cdot \text{sgn}(S_{E(k)}) \cdot \text{sgn}(S_{\|w(k)\|}) \cdot \left( \sum_j x_j \cdot g_j(k) \right) \cdot \left( \sum_j x_j \cdot w_j(k) \right) \\ a_3 &= +\frac{1}{2} \left( y'(k) \cdot \sum_j x_j \cdot w_j(k) \right)^2 \\ a_4 &= +e(k) \cdot \text{sgn}(S_{E(k)}) \cdot y'(k) \cdot \left( \sum_j x_j \cdot g_j(k) \right) \\ a_5 &= +e(k) \cdot \text{sgn}(S_{\|w(k)\|}) \cdot y'(k) \cdot \left( \sum_j x_j \cdot w_j(k) \right) \\ a_6 &= \begin{cases} 0, & \text{para Condição 4.70} \\ 2 \cdot S_{E(k)} & \text{para Condição 4.71} \end{cases} \end{aligned} \quad (4.90)$$

Sendo que os coeficientes obtidos caracterizam equações de 2ª ordem do tipo parabólico ( $I = 0$ , ver Seção A.2.1). Uma vez que o Teorema 2 também se aplica aos resultados obtidos, a análise segundo a condição 4.48 torna-se necessária para verificar a existência de uma região positiva para os ganhos  $\alpha$  e  $\beta$ :

- $b < 0$ , para  $S_{E(k)} > 0$  ( $\text{sgn}(S_{E(k)}) = +1$ ):

$$b = a_4 + a_5 \cdot p$$

$$a_4 = +e(k) \cdot y'(k) \cdot \left( \sum_j x_j \cdot g_j(k) \right)$$

onde :

$$g_j(k) = -e(k) \cdot y'(k) \cdot x_j$$

logo,

$$a_4 = -e_{(k)}^2 \cdot (y'(k))^2 \cdot \sum_j x_j^2 \quad (a_4 < 0)$$

A condição que garante a existência de um intervalo positivo para os ganhos é da forma (Caso especial,  $S_{E(k)} > 0$ ):

$$\text{Se } a_5 > 0, \quad \text{então: } p < \frac{-a_4}{a_5} \quad (4.91)$$

Utilizando os dados referentes ao nodo linear:  $y'_{(k)} = 1$ ,  $g_{j(k)} = -e_{(k)} \cdot x_j$  e  $y_{(k)} = \sum_j x_j \cdot w_{j(k)}$ , obtém-se os coeficientes referentes ao nodo linear, conforme demonstrado anteriormente.

#### 4.2.2.3 Rede MLP de duas camadas

Seja a saída de linear de uma rede MLP de duas camadas em relação a um vetor de entrada,  $\mathbf{x}$ :

$$y_{p(k)} = \sum_{i=1}^H w_{2ip(k)} \cdot f_i^h \left[ \sum_{j=1}^N (x_j \cdot w_{1ji(k)} + b_{1i(k)}) \right] + b_{2p(k)} \quad (4.92)$$

onde:

- Q** : número de saídas da rede
- H** : número de nodos da camada escondida
- N** : número de entradas da rede
- w** : vetor de pesos da rede
- b** : pesos de polarização
- $f^h$  : função de ativação da camada escondida

sendo  $w_1$ ,  $b_1$ ,  $w_2$  e  $b_2$  os pesos e os termos de polarização referentes às camadas de entrada (escondida) e saída da MLP, respectivamente.

Com o objetivo de simplificar a Equação 4.92, será considerada a seguinte expressão:

$$y_{p(k)} = \sum_{i=1}^{H+1} w_{2ip(k)} \cdot f_i^h \left[ \sum_{j=1}^{N+1} (x_j \cdot w_{1ji(k)}) \right] \quad (4.93)$$

onde:

$$x_{N+1} = 1$$

$$f_{H+1}^h = 1$$

$$\dot{f}_{H+1}^h = 0$$

sendo  $\dot{f}_i^h$ , a derivada da função de ativação do nodo  $i$  da camada escondida.

A região de convergência para os ganhos do algoritmo SMC-MOBJ para a rede MLP pode ser definida através da interseção das regiões individuais referentes à análise em

relação aos pesos da camada de saída e em relação aos pesos da camada de entrada.

A seguir serão identificadas as regiões de convergência referentes às camadas de uma rede MLP de duas camadas com saída linear.

- Camada de saída:

Considerando somente os pesos da camada de saída,  $w_2$ , como sendo os parâmetros de interesse, a equação da saída  $p$  de uma rede MLP pode ser escrita na seguinte forma:

$$y_{p(k)} = \sum_{i=1}^{H+1} w_{2ip(k)} \cdot f_{i(k)}^h \quad (4.94)$$

Supondo que o ajuste dos pesos seja realizado, inicialmente, somente nos pesos da camada de saída, a Equação 4.94 pode ser comparada à equação de um nodo perceptron linear, como pode ser visto na Figura 4.7.

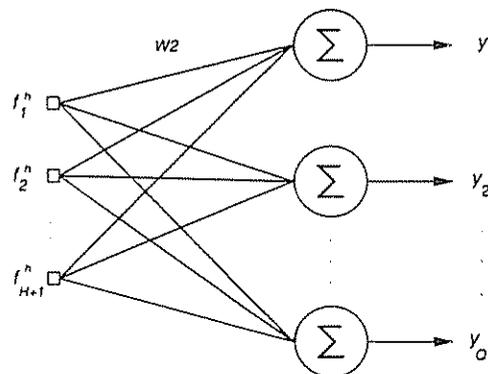


Figura 4.7: Camada de saída de uma rede MLP

Nesse caso, as entradas dos nodos correspondem aos valores de saída da função de ativação da camada escondida,  $f_{i(k)}^h$ , e os pesos, pelo vetor  $w_2$ . Os coeficientes são então obtidos a partir do uso das expressões relativas ao nodo linear aplicando-se a análise referente à rede com múltiplas saídas. Os coeficientes da equação do 2º grau associada às condições de convergência podem ser definidos como:

$$\begin{aligned}
a_1 &= \frac{1}{2} \sum_{p=1}^Q \left\{ e_{p(k)}^2 \cdot \left[ \sum_{i=1}^{H+1} (f_{i(k)}^h)^2 \right]^2 \right\} \\
a_2 &= -\text{sgn}(S_{E(k)}) \cdot \text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot \sum_{p=1}^Q \left\{ e_{p(k)} \cdot y_{p(k)} \sum_{i=1}^{H+1} (f_{i(k)}^h)^2 \right\} \\
a_3 &= \frac{1}{2} \sum_{p=1}^Q y_{p(k)}^2 \\
a_4 &= -\text{sgn}(S_{E(k)}) \cdot \sum_{p=1}^Q \left\{ e_{p(k)}^2 \cdot \sum_{i=1}^{H+1} (f_{i(k)}^h)^2 \right\} \\
a_5 &= -\text{sgn}(S_{\|\mathbf{w}(k)\|}) \cdot \sum_{p=1}^Q y_{p(k)} \cdot e_{p(k)} \\
a_6 &= \begin{cases} 0, & \text{para Condição 4.70} \\ 2 \cdot S_{E(k)} & \text{para Condição 4.71} \end{cases}
\end{aligned} \tag{4.95}$$

Segundo a análise realizada para o nodo perceptron linear (Seção 4.2.2.1), é garantida a existência de um intervalo positivo para os ganhos  $\alpha$  e  $\beta$  segundo os coeficientes definidos em 4.95.

- Camada de entrada ou escondida:

Seja agora considerado o ajuste dos pesos da camada de entrada ou escondida,  $w_{1ji(k)}$ . Para isso, os pesos da camada de saída serão mantidos constantes. A equação da saída  $p$  da rede assume a forma:

$$y_{p(k)} = \sum_{i=1}^{H+1} w_{2ip} \cdot f_i^h \left[ \sum_{j=1}^{N+1} x_j \cdot w_{1ji(k)} \right] \tag{4.96}$$

A saída no instante  $k + 1$  é então definida por:

$$y_{p(k+1)} = \sum_{i=1}^{H+1} w_{2ip} \cdot f_i^h \left[ \sum_{j=1}^{N+1} x_j \cdot (w_{1ji(k)} + \Delta w_{1ji(k)}) \right] \tag{4.97}$$

Aplicando-se a decomposição em Série de Taylor de primeira ordem tem-se:

$$\begin{aligned}
y_{p(k+1)} &= \sum_{i=1}^{H+1} w_{2ip} \cdot \left[ f_{i(k)}^h + \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot \Delta w_{1ji(k)} \right] \\
y_{p(k+1)} &= y_{p(k)} + \sum_{i=1}^{H+1} \left[ w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot \Delta w_{1ji(k)} \right]
\end{aligned} \tag{4.98}$$

expandindo-se o termo  $\Delta w_{1ji(k)}$  na Equação 4.98 obtém-se:

$$y_{p(k+1)} = y_{p(k)} + \left\{ -\text{sgn}(S_{E(k)}) \cdot \sum_{i=1}^{H+1} \left[ w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot g_{1ji(k)} \right] \right\} \cdot \alpha$$

$$+ \left\{ -\text{sgn}(S_{\|w(k)\|}) \cdot \sum_{i=1}^{H+1} \left[ w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot w_{1ji(k)} \right] \right\} \cdot \beta \quad (4.99)$$

Deve-se observar o limite de aproximação imposto ao ganho  $\alpha$  referentes à expansão em série de Taylor.

$$\alpha \leq \frac{\xi}{\left| \text{sgn}(S_{E(k)}) \cdot \sum_{j=1}^{N+1} x_j \cdot g_{1ji(k)} + p \cdot \text{sgn}(S_{\|w(k)\|}) \sum_{j=1}^{N+1} x_j \cdot w_{1ji(k)} \right|} \quad (4.100)$$

onde, apesar da Equação 4.100 estar atribuída a um único nodo, o limite para os ganhos pode ser definido em função de toda a rede, escolhendo o menor limite associado a um dos nodos da rede.

Os seguintes coeficientes referentes aos parâmetros  $\alpha$  e  $\beta$  são então obtidos:

$$a_p = -\text{sgn}(S_{E(k)}) \cdot \sum_{i=1}^{H+1} \left[ w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot g_{1ji(k)} \right]$$

$$b_p = -\text{sgn}(S_{\|w(k)\|}) \cdot \sum_{i=1}^{H+1} \left[ w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot w_{1ji(k)} \right]$$

Uma vez definidos os coeficientes e aplicando-se o Corolário 1, obtém-se os coeficientes das equações de convergência do 2º grau:

$$a_1 = \frac{1}{2} \sum_{p=1}^Q \left\{ \sum_{i=1}^{H+1} \left[ w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot g_{1ji(k)} \right] \right\}^2$$

$$a_2 = \text{sgn}(S_{E(k)}) \cdot \text{sgn}(S_{\|w(k)\|}) \cdot \sum_{p=1}^Q \left\{ \sum_{i=1}^{H+1} \left( w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot g_{1ji(k)} \right) \right.$$

$$\left. \cdot \sum_{i=1}^{H+1} \left( w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot w_{1ji(k)} \right) \right\}$$

$$\begin{aligned}
a_3 &= \frac{1}{2} \sum_{p=1}^Q \left\{ \sum_{i=1}^{H+1} \left[ w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot w_{1ji(k)} \right] \right\}^2 \\
a_4 &= \operatorname{sgn}(S_{E(k)}) \cdot \sum_{p=1}^Q \left\{ e_{p(k)} \cdot \sum_{i=1}^{H+1} \left[ w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot g_{1ji(k)} \right] \right\} \\
a_5 &= \operatorname{sgn}(S_{\|w(k)\|}) \cdot \sum_{p=1}^Q \left\{ e_{p(k)} \cdot \sum_{i=1}^{H+1} \left[ w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot w_{1ji(k)} \right] \right\} \\
a_6 &= \begin{cases} 0, & \text{para Condição 4.70} \\ 2 \cdot S_{E(k)}, & \text{para Condição 4.71} \end{cases}
\end{aligned}$$

onde, a partir do Teorema 2, é garantida a existência de uma região de convergência para os ganhos  $e$ , aplicando-se a Propriedade A.2, verifica-se que as regiões de convergência apresentam um comportamento elipsoidal para  $Q \neq 1$  ou parabólico para  $Q = 1$ .

A condição que garante a existência de um intervalo positivo para os ganhos é da forma (Caso especial,  $S_{E(k)} > 0$ ):

$$-b < 0, \text{ para } S_{E(k)} > 0 \text{ (} \operatorname{sgn}(S_{E(k)}) = +1 \text{):}$$

$$\begin{aligned}
b &= a_4 + a_5 \cdot p \\
a_4 &= + \sum_{p=1}^Q \left\{ e_{p(k)} \cdot \sum_{i=1}^{H+1} \left[ w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot \sum_{j=1}^{N+1} x_j \cdot g_{1ji(k)} \right] \right\}
\end{aligned}$$

onde :

$$g_{1ji(k)} = - \sum_{p=1}^Q e_{p(k)} \cdot w_{2ip} \cdot \dot{f}_{i(k)}^h \cdot x_j$$

logo,

$$a_4 = - \sum_{p=1}^Q \left\{ e_{p(k)}^2 \cdot \sum_{i=1}^{H+1} \left[ w_{2ip}^2 \cdot (\dot{f}_{i(k)}^h)^2 \cdot \sum_{j=1}^{N+1} x_j^2 \right] \right\} \quad (a_4 < 0)$$

$$\text{Se } a_5 > 0, \text{ então: } p < \frac{-a_4}{a_5} \quad (4.101)$$

A análise realizada para a superfície do Erro garante a existência de um intervalo positivo para os ganhos  $\alpha$  e  $\beta$ . As Figuras 4.8 e 4.9 exemplificam as regiões de convergência para  $S_{E(k)} > 0$  e  $S_{E(k)} < 0$ , respectivamente.

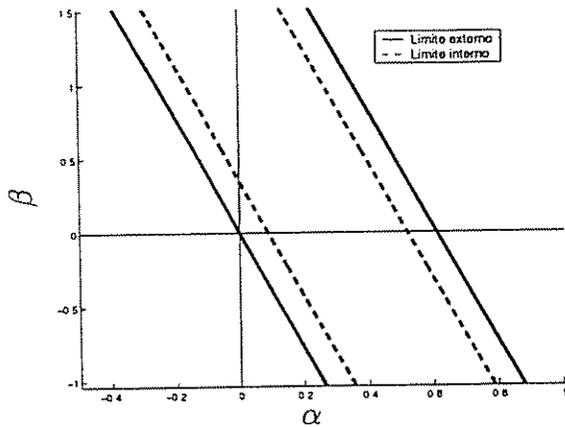


Figura 4.8: Região de convergência para  $S_{E(k)} > 0$

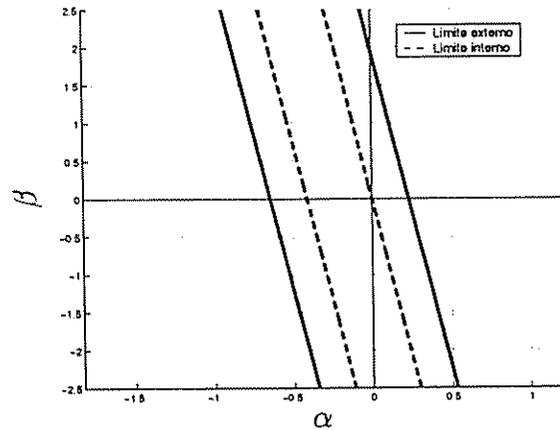


Figura 4.9: Região de convergência para  $S_{E(k)} < 0$

A região de convergência do Erro para uma rede MLP de duas camadas é obtida através da interseção das regiões obtidas separadamente para a camada escondida e para a camada de entrada. A Figura 4.10 exemplifica o método apresentando o comportamento das regiões obtidas (a) e a respectiva interseção (b). Deve-se observar a existência de uma limitação imposta pela aproximação utilizando a expansão em Série de Taylor (Equação 4.100) e a restrição obtida para a função Norma para que seja definida a solução final para os ganhos  $\alpha$  e  $\beta$ .

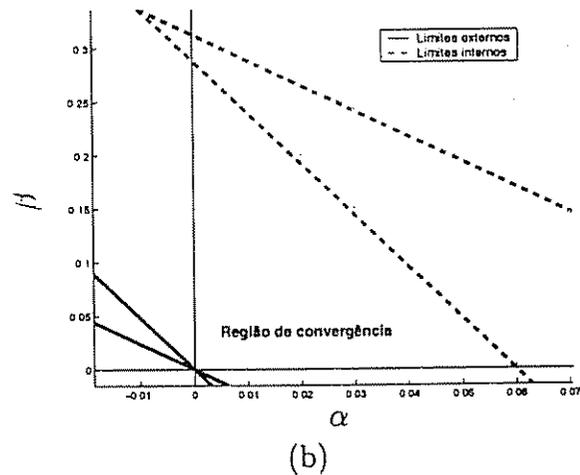
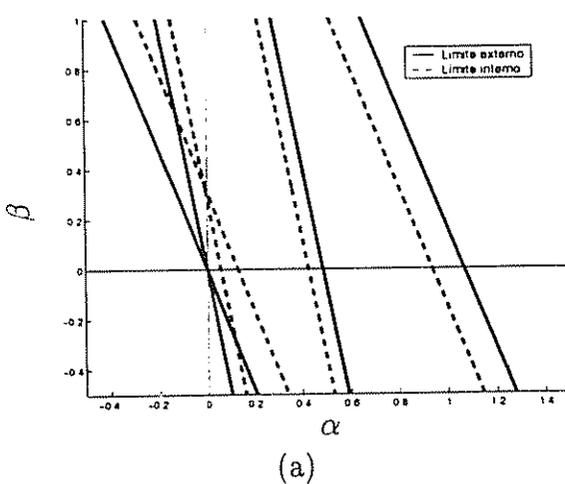


Figura 4.10: Região de convergência para uma rede MLP de duas camadas

A Figura 4.11 mostra a interseção das regiões definidas pelas equação referentes às superfícies do erro, camadas de entrada e escondida, e superfície da norma. Nesse caso, não existem limites internos em relação às regiões definidas pelas equação referentes à convergência do Erro.

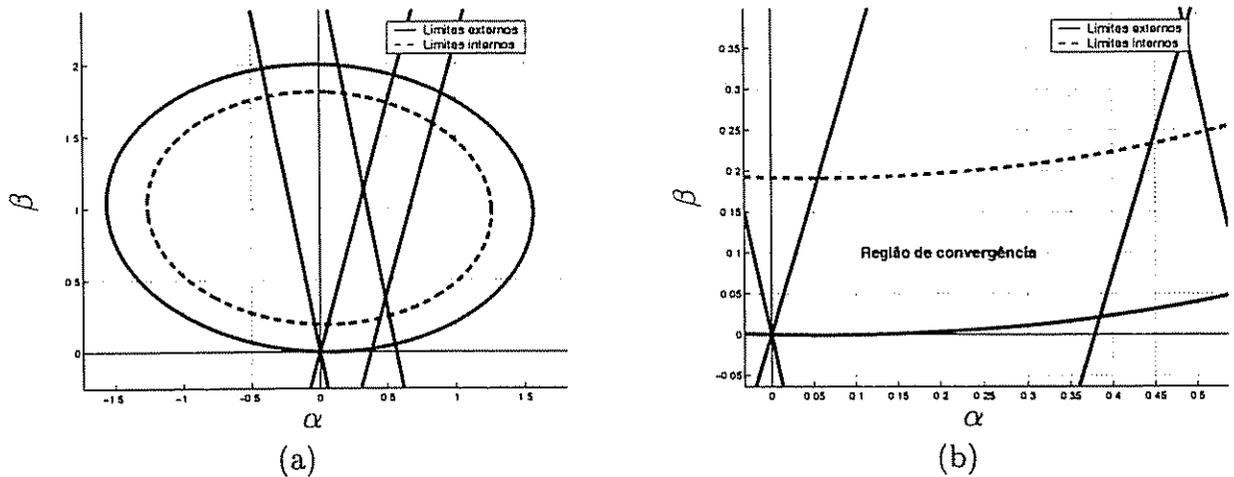


Figura 4.11: Região de convergência para uma rede MLP de duas camadas

### 4.2.3 Considerações sobre a Convergência do Método

Como foi demonstrado, as equações de convergência para os ganhos  $\alpha$  e  $\beta$ , segundo o algoritmo proposto (SMC-MOBJ), podem ser definidas por equações de 2ª ordem em relação aos ganhos, seja para a superfície do Erro ou em relação à superfície da Norma. Dessa forma, o problema em questão, foi abordado de forma a determinar os coeficientes dessas equações.

Em relação à determinação dos coeficientes para as equações de convergência do Erro para o nodo perceptron não-linear e a rede MLP de duas camadas, a decomposição em Série de Taylor possibilitou estabelecer uma relação linear entre a saída do nodo ou da rede no instante  $k + 1$ , os ganhos e a saída no instante  $k$ . Uma vez que foi utilizada uma decomposição de primeira ordem, a aproximação da função encontra-se limitada às regiões próximas do ponto de operação. Mas, uma vez que a equação de ajuste dos pesos também utiliza uma aproximação de primeira ordem para a superfície do Erro, o gradiente, o uso de aproximações de ordem superior para a função de ativação representa um alto custo computacional e não garante um aumento significativo na velocidade de convergência do método.

Uma vez obtidos os coeficientes das equações de convergência do erro e da norma e observado o limite imposto pela decomposição em série de Taylor, a região final para os ganhos é determinada pela interseção das regiões definidas pelas equações de convergência da norma e do erro. Essa região pode ser representada em um plano ( $\mathbb{R}^2$ ) possibilitando uma análise gráfica, conforme demonstrado nas Figuras 4.10 e 4.11.

A região de convergência depende do valor atual do vetor de pesos e do gradiente no ponto, os ganhos podem ser então determinados iterativamente, a cada passo de atualização dos pesos ou através de uma heurística como, por exemplo, ajustando os pesos somente quando a convergência às superfícies de deslizamento não apresenta um comportamento assintótico.

### 4.3 Conclusões do Capítulo

Um novo algoritmo para o treinamento de MLPs foi apresentado. Partindo do princípio de que um plano de soluções pode ser definido em função do Erro e da Norma, o treinamento é descrito em função de um ponto em  $\mathbb{R}^2$  a ser alcançado. Definidas as direções de atualização dos pesos segundo o método do gradiente descendente e adicionando-se superfícies de controle, o algoritmo proposto possibilita a convergência para qualquer solução factível do plano.

A partir da análise segundo a Teoria de Modos Deslizantes, foi possível obter um conjunto de equações que definem valores ou intervalos para os ganhos  $\alpha$  e  $\beta$  capazes de garantir a convergência do método proposto. A metodologia apresentada não garante que o estado final desejado para o erro ( $E_t$ ) e para a norma dos pesos ( $\|w_t\|$ ) será alcançado, mas garante uma convergência assintótica para o mesmo. Dessa forma, mesmo que a solução desejada esteja definida na região de soluções não-existentes, o algoritmo é capaz de convergir para uma solução factível próxima da solução desejada (não-existente).

# Capítulo 5

## Determinação das Soluções Eficientes e Variações para o Algoritmo Proposto

Neste capítulo são definidos alguns métodos para a obtenção das soluções eficientes, ou pertencentes ao conjunto Pareto-ótimo, utilizando o algoritmo proposto (SMC-MOBJ). A partir de uma análise do comportamento da equação de ajuste dos pesos segundo o método SMC-MOBJ e utilizando métodos de otimização, são desenvolvidos algoritmos multi-objetivo capazes de promover o aumento da velocidade de convergência.

### 5.1 Obtenção do Conjunto Pareto através de Trajetórias no Plano de Soluções

Conforme a Equação 4.7, o algoritmo proposto procura por um ponto específico,  $m(E_t, \|\mathbf{w}_t\|)$ , do plano de soluções. Uma vez que o algoritmo seja capaz de convergir ao ponto desejado, podem-se definir trajetórias no plano de soluções a partir de um ponto inicial. Lembrando que o Conjunto Pareto-ótimo representa o limite entre as soluções factíveis e as soluções não-existentes, mesmo que os pontos objetivo estejam definidos na região de soluções não-existentes, o algoritmo é capaz de achar uma solução factível próxima ou sob o Pareto. Dessa forma, o método proposto é capaz de estimar a curva Pareto-ótimo.

Conhecendo-se o comportamento do conjunto Pareto no espaço solução: **Erro**  $\times$  **Norma** (ver Figura 3.2), um ponto inicial para o treinamento da MLP com norma praticamente nula pode ser obtido através da multiplicação dos pesos gerados aleatoriamente por um fator de escala, por exemplo, da ordem de  $10^{-6}$ . Uma vez que esse ponto possa ser identificado como um limite inferior da curva Pareto, uma trajetória pode ser traçada utilizando critérios para a definição dos próximos pontos a serem *alcançados* pelo algoritmo.

A seguinte técnica foi elaborada para a definição dos pontos a serem alcançados pelo

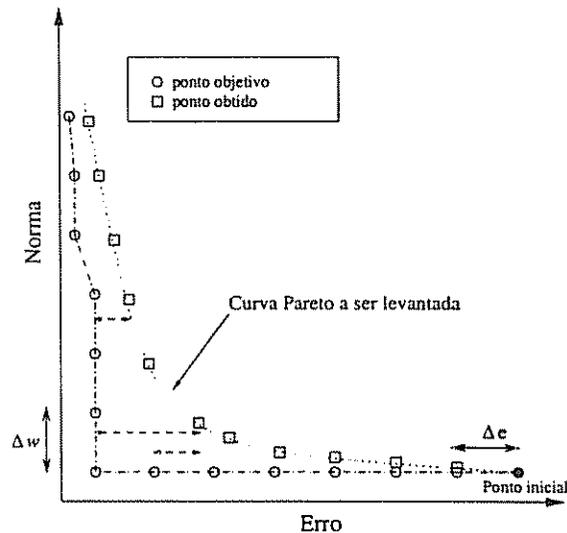


Figura 5.1: Método para a obtenção da Curva Pareto

algoritmo:

1. A partir de um ponto inicial, que representa uma rede MLP, define-se uma trajetória cujos pontos apresentam valores de norma constantes e iguais à norma do ponto inicial. Em relação à coordenada referente ao erro, subtrai-se gradativamente um  $\Delta e$  (positivo). Uma vez que essa trajetória ultrapasse a fronteira das soluções eficientes, o erro da rede irá convergir para um determinado valor superior àquele estabelecido pela trajetória.
2. Após a convergência do erro, partindo dessa solução, uma nova trajetória é estabelecida na qual são definidos pontos com um incremento gradual e positivo da coordenada referente à norma dos pesos. O erro desejado é mantido em um valor positivo, constante e inferior àquele obtido ao final da primeira etapa.
3. Uma vez que a convergência do erro seja verificada, os pontos referentes às trajetórias são ajustados de forma a manter o erro desejado na região de soluções não-existent. O método então, proporciona a convergência do estado da rede a uma solução factível próxima desse ponto.

O processo de obtenção do conjunto Pareto-ótimo descrito é demonstrado na Figura 5.1. O número de pontos a ser levantado é estabelecido como critério de parada desse método. Deve-se especificar também o número máximo de iterações do algoritmo a ser realizado para cada ponto definido. Neste método, o algoritmo proposto é utilizado  $q$ -vezes, onde  $q$  é o número de soluções Pareto a serem obtidas.

Outro método, partindo de um ponto superior, pode ser definido. Os critérios de ajuste das trajetórias são semelhantes. Entretanto, é necessário um ponto inicial superior que pode ser obtido utilizando, por exemplo, o algoritmo de Levenberg-Marquardt.

É possível, também, obter uma aproximação parcial do conjunto Pareto partindo de um dos extremos da curva, conforme pode ser observado nas Figuras 5.2 e 5.3:

1. Partindo de um ponto superior da curva, obtido através do uso de um algoritmo de minimização do erro, pode-se especificar um conjunto de pontos em uma trajetória retilínea cujo erro é definido constante e a norma variando do valor inicial até norma nula.
2. De modo semelhante, partindo de um ponto de norma nula, a trajetória retilínea apresentará valores com norma constante e valores, para o erro, variando entre o valor inicial e o valor nulo.

O processo descrito acima é definido como *parcial* devido ao fato de abranger diferentes regiões da curva Pareto.

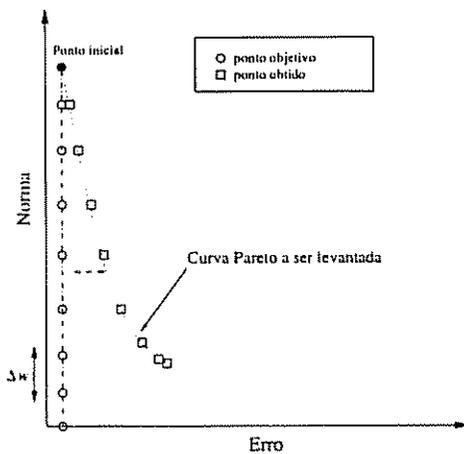


Figura 5.2: Aproximação de soluções eficientes de norma elevada

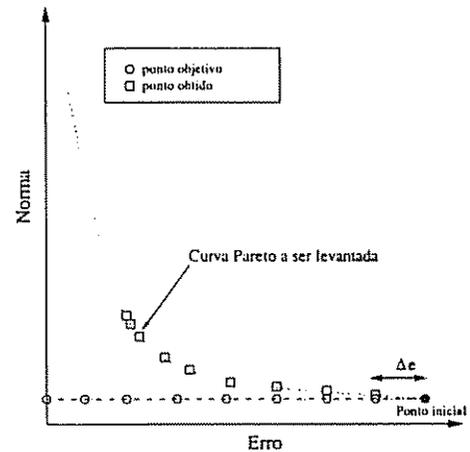


Figura 5.3: Aproximação de soluções eficientes de norma reduzida

## 5.2 Obtenção do Conjunto Pareto através da Restrição da Norma

A Equação 4.7 orienta o ajuste dos pesos de forma a alcançar uma situação onde  $E_{(k)} = E_t$  e  $\|\mathbf{w}_{(k)}\| = \|\mathbf{w}_t\|$ . Portanto,  $E_t$  e  $\|\mathbf{w}_t\|$  representam objetivos ou restrições impostas à solução desejada. Sendo o conjunto Pareto-ótimo definido pelos pontos pertencentes a uma região na qual a função do erro é mínima para um dado valor da norma ou vice-versa, o objetivo do treinamento pode ser re-definido de forma a representar uma minimização do Erro sujeita a uma restrição da norma.

$$\min_{\mathbf{w} \in W} \frac{1}{N_T} \sum_{j=1}^{N_T} (d_j - f(\mathbf{x}_j; \mathbf{w}))^2 \quad (5.1)$$

sujeito a:  $\|\mathbf{w}_{(k)}\| \approx \|\mathbf{w}_t\|$

Tal objetivo pode ser incorporado à Equação 4.7 através da definição de um erro nulo desejado ( $E_t = 0$ ):

$$\begin{aligned} E_t &= 0 \\ S_{E(k)} &= E(k) \end{aligned} \quad (5.2)$$

sendo a equação da função do erro definida em 4.1, tem-se:

$$\begin{aligned} E_{(k)} &> 0, \forall k \\ S_{E(k)} &> 0 \\ \text{sgn}(S_{E(k)}) &= +1 \end{aligned} \quad (5.3)$$

A equação de correção dos pesos é então definida da seguinte forma:

$$\Delta w_{ji(k)} = -\alpha \cdot \frac{\partial E(k)}{\partial w_{ji(k)}} - \beta \cdot \text{sgn}(S_{\|w(k)\|}) \cdot w_{ji(k)} \quad (5.4)$$

Através da Equação 5.4 é possível realizar um treinamento que obtenha o menor valor da função do erro para um determinado valor de norma. Esse método não é capaz de alcançar soluções arbitrárias, localizadas na região de soluções factíveis e não pertencentes ao conjunto Pareto-ótimo. Seu comportamento é descrito pela Figura 5.4.

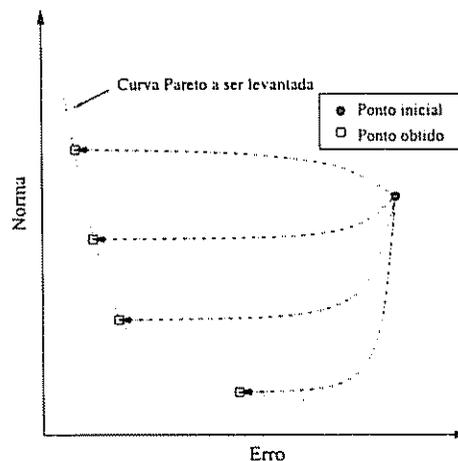


Figura 5.4: Método de Geração do Conjunto Pareto através da Restrição da Norma

Neste método, a escolha do parâmetro  $\beta$  determina a velocidade de convergência da norma para o valor desejado ( $\|w_t\|$ ).

### 5.3 Variações do algoritmo Multi-Objetivo proposto

A partir da Equação 4.7 e do uso de técnicas de controle por modos deslizantes, é possível estender o estudo das superfícies de deslizamento aplicando-as junto a outras técnicas de

otimização. Como resultado, tem-se algoritmos *híbridos* capazes de alcançar as soluções do conjunto Pareto em um número menor de iterações.

### 5.3.1 O Algoritmo Multi-Objetivo Utilizando o Método de Levenberg-Marquardt

O algoritmo SMC-MOBJ proposto define um ponto objetivo no espaço solução  $(E_t, ||\mathbf{w}_t||)$  que deverá ser alcançado durante o treinamento. Uma vez que esse ponto encontra-se na região factível, o algoritmo proposto será capaz de alcançá-lo.

Para alcançar soluções do conjunto Pareto-ótimo, que representam as soluções de erro mínimo para um determinado valor da norma do vetor de pesos, um erro nulo pode ser especificado o que torna a equação de ajuste na forma descrita pela Equação 5.4.

A Equação 5.4 é semelhante à descrita para o algoritmo *backpropagation* mas, com um termo de restrição para a norma. Esta pode ser descrita através da seguinte operação:

$$\begin{aligned} & \text{minimizar } E(\mathbf{w}) \\ & \text{sujeito a: } S_{||\mathbf{w}||} \approx 0 \end{aligned} \quad (5.5)$$

O algoritmo multi-objetivo utilizando o método de Levenberg-Marquardt (MOBJ-SMC-LM) [Costa et al., 2002b] é capaz de realizar a operação descrita acima (Equação 5.5). Comparando a Equação 5.4 com a equação padrão do algoritmo de Levenberg-Marquardt (Equação 2.15) percebe-se que a restrição da norma é alcançada através do uso do gradiente descendente da função da norma controlado pela respectiva superfície de deslizamento. O mesmo princípio pode ser aplicado à equação de ajuste dos pesos segundo o método de Levenberg-Marquardt, o que permite obter a seguinte equação de ajuste:

$$\Delta \mathbf{w} = [J^T(\mathbf{w})J(\mathbf{w}) + \mu I]^{-1} J(\mathbf{w})e(\mathbf{w}) - \beta \cdot \text{sign}(S_{||\mathbf{w}||}) \cdot w_{ji} \quad (5.6)$$

Para evitar efeitos indesejáveis de chaveamento na superfície da norma, a função sinal pode ser substituída pela função linear da superfície, o que torna a Equação 5.6 na forma:

$$\Delta \mathbf{w} = [J^T(\mathbf{w})J(\mathbf{w}) + \mu I]^{-1} J(\mathbf{w})e(\mathbf{w}) - \beta \cdot (S_{||\mathbf{w}||}) \cdot w_{ji} \quad (5.7)$$

onde  $\beta$  é um ganho positivo para a superfície de deslizamento e o parâmetro  $\mu$  é ajustado de acordo com a distância euclidiana no espaço bi-dimensional entre o ponto desejado  $(E_t, ||\mathbf{w}_t||)$  e os valores atuais das funções do erro e norma. O parâmetro  $\mu$  é multiplicado por um fator  $\gamma$  quando essa distância é aumentada durante uma iteração e é dividido pelo mesmo fator quando a distância é reduzida.

$$\mu = \begin{cases} \mu/\gamma, & \text{se } dist(k+1, t) \leq dist(k, t) \\ \mu \cdot \gamma, & \text{se } dist(k+1, t) > dist(k, t) \end{cases} \quad (5.8)$$

onde  $dist(k, t)$  é a distância euclideana entre o erro atual, função de norma e seus valores desejados.

$$dist(k, t) = \sqrt{(\|\mathbf{w}_{(k)}\| - \|\mathbf{w}_t\|)^2 + (E(\mathbf{w}_{(k)}) - E_t)^2} \quad (5.9)$$

## 5.4 Treinamento Multi-Objetivo através do Controle da Taxa de Aprendizado

A equação de ajuste dos pesos segundo o algoritmo SMC-MOBJ permite o controle simultâneo do erro e da norma dos pesos de uma rede MLP possibilitando a convergência da rede a um ponto desejado no plano de soluções, desde que esse ponto seja definido na região factível. Uma vez que a superfície da norma seja alcançada, alguns fatores podem provocar a perda da velocidade de convergência do algoritmo em relação à superfície do erro.

De acordo com a equação de correção dos pesos segundo o método SMC-MOBJ (Equação 4.7):

$$\Delta w_{ji(k)} = -\alpha \cdot sgn(S_{E(k)}) \cdot \frac{\partial E(k)}{\partial w_{ji(k)}} - \beta \cdot sgn(S_{\|\mathbf{w}(k)\|}) \cdot w_{ji(k)}$$

exceto nos casos onde  $S_{E(k)} = 0$ , o termo referente à função do erro pode ser nulo se:  $\frac{\partial E(k)}{\partial w_{ji(k)}} = 0$ , o que não impede o algoritmo de efetuar o ajuste dos pesos uma vez que o termo de ajuste dos pesos referente à norma da MLP seja não-nulo. Por outro lado, se  $S_{\|\mathbf{w}(k)\|} = 0$ , a equação de ajuste é semelhante à equação de correção dos pesos do método *backpropagation* (ver Equação 2.6), ou seja, o algoritmo pode estar sujeito às regiões de gradientes nulos ou mínimos locais. Tal fenômeno tem como consequência a perda de velocidade de convergência do algoritmo. Mas, mesmo que os casos em questão não ocorram, sendo ambos os termos não-nulos, é improvável que a direção de minimização do erro seja sempre coerente com a direção de minimização da norma, podendo ocorrer a seguinte situação:

$$sgn\left(\frac{\partial E(k)}{\partial w_{ji(k)}}\right) \neq sgn(w_{ji(k)})$$

indicando que um dos termos:  $\left\{-\alpha \cdot sgn(S_{E(k)}) \cdot \frac{\partial E(k)}{\partial w_{ji(k)}}\right\}$  ou  $\left\{-\beta \cdot sgn(S_{\|\mathbf{w}(k)\|}) \cdot w_{ji(k)}\right\}$  possa influenciar ou se sobrepor ao outro e vice-versa, para determinado conjunto de parâmetros de ajuste,  $w_{ji(k)}$ . Considerando então esses fatores, é possível identificar três

possibilidades de interação entre os termos de ajuste mencionados (erro e norma). Estabelecendo a convergência da função do erro como referência, tem-se:

1. Interação Construtiva: a soma dos termos de ajuste referentes ao gradiente da norma e do erro produzem um vetor  $\Delta w_{ji(k)}$ , que permite uma convergência mais rápida do erro e/ou da norma em relação ao ajuste utilizando somente o gradiente referente à função do erro.
2. Interação Destrutiva: a soma dos ajustes de norma e erro resultam em um vetor  $\Delta w_{ji(k)}$  que apresenta uma velocidade de convergência do erro reduzida em relação ao ajuste segundo o gradiente do erro.
3. Interação Imparcial: a combinação dos gradientes referentes às funções do erro e da norma não influencia a convergência do erro.

De acordo com a Figura 3.2, a relação entre o erro e a norma no conjunto Pareto é semelhante à de uma função inversa. Analisando-se o treinamento segundo os tradicionais algoritmos baseados somente no gradiente descendente da função do erro e observando a relação Erro  $\times$  Norma da curva Pareto, pode-se concluir que a minimização do erro implica no aumento da função da norma (ver Figura 3.20). Essa característica indica que a possibilidade de ocorrência de uma interação imparcial, conforme descrito acima, é pequena quando a norma desejada é superior à norma atual da rede.

A seguir é abordada uma nova técnica baseada na análise da norma para o controle da minimização do erro definindo-se intervalos para a taxa de aprendizagem ou o parâmetro  $\alpha$ . A equação de ajuste dos pesos é idêntica à do algoritmo *backpropagation*, entretanto, o objetivo do treinamento é minimizar o erro mantendo o valor da norma constante ou variando-a dentro de um determinado intervalo. O uso desse algoritmo permite obter pontos pertencentes, ou muito próximos, ao conjunto Pareto-ótimo.

#### 5.4.1 Controle da norma de MLPs através da limitação da taxa de aprendizado: $\alpha$

Sendo um algoritmo que permite minimizar a função de custo do erro mantendo o valor da norma dos pesos dentro de uma determinada faixa, é necessário obter, inicialmente, uma rede MLP cujo valor da norma dos pesos esteja contido dentro do intervalo especificado. Para isso, os pesos de uma rede podem ser multiplicados por um escalar  $c$  de forma que a norma desejada,  $\|\mathbf{w}_t\|$ , seja alcançada em um único passo. Partindo da norma inicial da rede,  $\|\mathbf{w}_{(k)}\|$  onde  $k = 0$ , a relação entre a norma desejada e a norma da rede pode ser expressa por:

$$\|\mathbf{w}_t\| = c \cdot \|\mathbf{w}_{(k)}\|$$

$$c = \frac{\|\mathbf{w}_t\|}{\|\mathbf{w}(k)\|} \quad (5.10)$$

Substituindo a equação da norma (3.16) em (5.10) tem-se:

$$\begin{aligned} \|\mathbf{w}_t\| &= c \cdot \sqrt{\sum_{ji} w_{ji(k)}^2} \\ \|\mathbf{w}_t\| &= \sqrt{c^2 \cdot \sum_{ji} w_{ji(k)}^2} = \sqrt{\sum_{ji} b_{ji(k)}^2} \end{aligned}$$

onde  $b_{ji(k)}$  representa os pesos da rede com norma  $\|\mathbf{w}_t\|$

$$\begin{aligned} \|\mathbf{w}_t\| &= \sqrt{\sum_{ji} c^2 \cdot w_{ji(k)}^2} = \sqrt{\sum_{ji} b_{ji(k)}^2} \\ b_{ji(k)}^2 &= c^2 \cdot w_{ji(k)}^2 \end{aligned}$$

$$b_{ji(k)} = c \cdot w_{ji(k)} \quad (5.11)$$

As Equações (5.10) e (5.11) permitem alcançar a norma desejada utilizando uma única operação de multiplicação em qualquer instante de tempo,  $k$ . Uma vez ajustada a norma da MLP, é necessário definir os critérios para o uso do algoritmo proposto.

1. A análise parte do princípio de que a norma da rede MLP se encontra inicialmente contida no seguinte intervalo:

$$|(\|\mathbf{w}(k)\| - \|\mathbf{w}_t\|)| < \varphi, \quad (\varphi > 0)$$

2. O algoritmo descrito a seguir procura minimizar o erro sem alterar significativamente o valor da norma. Portanto, o seu uso limita-se à estimação das soluções pertencentes ao conjunto Pareto-ótimo, não sendo capaz de alcançar soluções arbitrárias do plano de soluções.

A equação padrão para o ajuste de pesos de MLPs é da forma:

$$w_{ji(k+1)} = w_{ji(k)} + \Delta w_{ji(k)} \quad (5.12)$$

Para o ajuste segundo o método SMC-MOBJ onde o termo de restrição do erro não é positivo ( $sgn(S_{E(k)}) = +1$ ), a equação do ajuste pode ser escrita como:

$$\Delta w_{ji(k)} = -\alpha \cdot \frac{\partial E}{\partial w_{ji}} - \beta \cdot sgn(\|\mathbf{w}(k)\| - \|\mathbf{w}_t\|) \cdot w_{ji} \quad (5.13)$$

o que é equivalente a atribuir ao erro desejado o valor nulo:

$$\begin{aligned}
E_t &= 0 \\
\text{sgn}(E_{(k)} - E_0) &= +1 \\
-\alpha \cdot \text{sgn}(E_{(k)} - E_0) \cdot \frac{\partial E}{\partial w_{ji}} &= -\alpha \cdot \frac{\partial E}{\partial w_{ji}}
\end{aligned}$$

A equação da norma dos pesos de uma MLP descrita na Equação 3.16 pode ser expandida utilizando-se a Equação 5.12:

$$\|\mathbf{w}_{(k+1)}\| = \sqrt{\sum_{ji} (w_{ji(k)} + \Delta w_{ji(k)})^2} \quad (5.14)$$

Analisando-se a Equação 5.14 é possível determinar as situações nas quais ocorrem o crescimento ou o decrescimento da norma dos pesos em relação aos ajustes referentes a cada peso,  $w_{ji(k)}$  e  $\Delta w_{ji(k)}$ .

- 1) Se  $\Delta w_{ji(k)} > 0$  e  $w_{ji(k)} > 0$ , Então  $\|\mathbf{w}_{(k+1)}\| > \|\mathbf{w}_{(k)}\|$
- 2) Se  $\Delta w_{ji(k)} > 0$  e  $w_{ji(k)} < 0$ , Então  $\|\mathbf{w}_{(k+1)}\| < \|\mathbf{w}_{(k)}\|$
- 3) Se  $\Delta w_{ji(k)} < 0$  e  $w_{ji(k)} > 0$ , Então  $\|\mathbf{w}_{(k+1)}\| < \|\mathbf{w}_{(k)}\|$
- 4) Se  $\Delta w_{ji(k)} < 0$  e  $w_{ji(k)} < 0$ , Então  $\|\mathbf{w}_{(k+1)}\| > \|\mathbf{w}_{(k)}\|$

O objetivo principal do método consiste na decomposição dos gradientes dos pesos segundo o seu efeito final na função norma dos pesos: crescimento ou decrescimento. A seguir, para cada tipo de gradiente, um valor para a taxa de aprendizado é encontrada de forma que os ajustes segundo os gradientes definidos pelo método limitem a norma dos pesos a uma determinada região, sem impedir a minimização da função do erro.

É importante observar que, na análise da norma dos pesos descrita acima, supõe-se que somente os pesos em questão são ajustados. Dessa forma, pode-se subdividir o ajuste dos pesos segundo os seguintes critérios:

- Ajustes que promovem o crescimento da Norma (1 e 4)
- Ajustes que promovem o decrescimento da Norma (2 e 3)

A seguir os casos apresentados são analisados. Para melhor análise desses casos é necessário expandir a Equação 5.14:

$$\|\mathbf{w}_{(k+1)}\| = \sqrt{\sum_{ji} (w_{ji(k)}^2 + 2 \cdot w_{ji(k)} \cdot \Delta w_{ji(k)} + \Delta w_{ji(k)}^2)} \quad (5.16)$$

agrupando os somatórios:

$$\|\mathbf{w}_{(k+1)}\| = \sqrt{\left[ \sum_{ji} w_{ji(k)}^2 + \sum_{ji} (2 \cdot w_{ji(k)} \cdot \Delta w_{ji(k)}) + \sum_{ji} (\Delta w_{ji(k)})^2 \right]} \quad (5.17)$$

sendo o termo de ajuste para esse método,  $\Delta w_{ji(k)}$ , descrito em função da taxa de aprendizado,  $\alpha$ , e do gradiente representado pelo vetor  $\mathbf{g}(k)$  ( $\mathbf{g}(k) \in \mathfrak{R}^n$  :  $g_{ji(k)} = \frac{\partial E}{\partial w_{ji(k)}}$ ),

$$\Delta w_{ji(k)} = -\alpha \cdot g_{ji(k)} \quad (5.18)$$

a equação final é escrita na forma:

$$\|\mathbf{w}_{(k+1)}\| = \sqrt{\left[ \sum_{ji} w_{ji(k)}^2 - 2 \cdot \alpha \cdot \sum_{ji} (w_{ji(k)} \cdot g_{ji(k)}) + \alpha^2 \cdot \sum_{ji} g_{ji(k)}^2 \right]} \quad (5.19)$$

onde  $\alpha > 0$ .

#### 5.4.1.1 Casos onde ocorre o decrescimento da norma dos pesos

- $w_{ji(k)} > 0$ ,  $\Delta w_{ji(k)} < 0$  ( $g_{ji(k)} > 0$ )

Seja definido o seguinte vetor,  $\mathbf{g}_{(k)}^-$ , pertencente ao espaço  $\mathfrak{R}^n$ , onde  $n$  representa o número de pesos ou parâmetros de ajuste da rede MLP:

$$g_{ji(k)}^- = \begin{cases} g_{ji(k)}, & \text{Se } w_{ji(k)} > 0 \text{ e } g_{ji(k)} > 0 \\ 0, & \text{caso contrário} \end{cases} \quad (5.20)$$

Realizando-se o ajuste dos pesos utilizando o vetor  $\mathbf{g}_{(k)}^-$  ( $\Delta w_{ji(k)} = -\alpha \cdot g_{ji(k)}^-$ ), é possível afirmar que:

$$\|\mathbf{w}_{(k+1)}\| < \|\mathbf{w}_{(k)}\| \quad (5.21)$$

Substituindo-se a Equação (5.21) na Equação 5.19, considerando o vetor  $\mathbf{g}_{(k)}^-$  e elevando-se ambos os termos ao quadrado, é possível cancelar a raiz quadrada de modo que:

$$\sum_{ji} w_{ji(k)}^2 - 2 \cdot \alpha \cdot \sum_{ji} (w_{ji(k)} \cdot g_{ji(k)}^-) + \alpha^2 \cdot \sum_{ji} (g_{ji(k)}^-)^2 < \|\mathbf{w}_{(k)}\|^2 \quad (5.22)$$

A partir da definição da função norma dos pesos expressa na Equação 3.16, é possível definir a seguinte relação:

$$\|\mathbf{w}_{(k)}\|^2 = \sum_{ji} w_{ji(k)}^2 \quad (5.23)$$

o que permite simplificar a Equação 5.22 obtendo a expressão:

$$-2.\alpha. \sum_{ji} \left( w_{ji(k)} \cdot g_{ji(k)}^- \right) + \alpha^2 \cdot \sum_{ji} (g_{ji(k)}^-)^2 < 0 \quad (5.24)$$

Colocando o termo  $\alpha$  em evidência e escrevendo a inequação nessa forma, tem-se:

$$\alpha \cdot \left( \alpha \cdot \sum_{ji} (g_{ji(k)}^-)^2 - 2 \cdot \sum_{ji} w_{ji(k)} \cdot g_{ji(k)}^- \right) < 0 \quad (5.25)$$

lembrando que  $\alpha > 0$ , é possível definir um intervalo para a taxa de aprendizado de tal forma a garantir o decrescimento da norma:

$$\alpha < \frac{2 \cdot \sum_{ji} w_{ji(k)} \cdot g_{ji(k)}^-}{\sum_{ji} (g_{ji(k)}^-)^2} \quad (5.26)$$

$$0 < \alpha < \frac{2 \cdot \sum_{ji} w_{ji(k)} \cdot g_{ji(k)}^-}{\sum_{ji} (g_{ji(k)}^-)^2} \quad (5.27)$$

Entretanto, o intervalo expresso na inequação 5.27 não indica o valor de  $\alpha$  que garante a minimização da função do erro e sim da função da norma. A heurística adotada para a escolha do ganho  $\alpha$  final consiste na definição de aproximadamente cinco valores para o ganho  $\alpha$  distribuídos uniformemente no intervalo encontrado, ou na forma:

$$\alpha \in \{ \alpha^*, \alpha^* \cdot 10^{-2}, \alpha^* \cdot 10^{-4}, \alpha^* \cdot 10^{-6}, \alpha^* \cdot 10^{-8} \} \quad (5.28)$$

onde  $\alpha^* = \frac{2 \cdot \sum_{ji} w_{ji(k)} \cdot g_{ji(k)}^-}{\sum_{ji} (g_{ji(k)}^-)^2}$ .

O valor do ganho  $\alpha$  que minimiza a função do erro é então utilizado para o ajuste dos pesos, caso contrário o ajuste não é realizado. Dessa forma, é possível reduzir a norma dos pesos da rede e a função do erro simultaneamente.

- $w_{ji(k)} < 0, \Delta w_{ji(k)} > 0$  ( $g_{ji(k)} < 0$ )

Nesse caso, o vetor  $\mathbf{g}_{(k)}^-$  é definido como:

$$g_{ji(k)}^- = \begin{cases} g_{ji(k)}, & \text{Se } w_{ji(k)} < 0 \text{ e } g_{ji(k)} < 0 \\ 0, & \text{caso contrário} \end{cases} \quad (5.29)$$

A análise de convergência para este caso é idêntica à descrita anteriormente, sendo o intervalo 5.27 também obtido.

Suscintamente, o cálculo do intervalo para o ganho  $\alpha$  pode ser realizado agrupando-se os casos onde ocorre o decrescimento da norma dos pesos e efetuando-se os cálculos descritos em (5.27) uma única vez. A Equação 5.33 demonstra a forma geral para definir o vetor  $\mathbf{g}_{(k)}^-$ :

$$g_{ji(k)}^- = \begin{cases} g_{ji(k)}, & \text{Se } w_{ji(k)} \cdot g_{ji(k)} > 0 \\ 0, & \text{caso contrário} \end{cases} \quad (5.30)$$

Para este caso, o ajuste dos pesos é realizado da seguinte forma:

$$w_{ji(k+1)} = w_{ji(k)} - \alpha \cdot g_{ji(k)}^-$$

onde o ganho  $\alpha$  pode ser definido pela Equação 5.28.

#### 5.4.1.2 Casos onde ocorre o crescimento da norma dos pesos

Supondo que o ajuste dos pesos segundo o critério de decrescimento da norma tenha sido realizado. O estudo dos casos que promovem o crescimento da norma tem como objetivo implementar um novo ajuste dos pesos com as seguintes características:

1. Ajustar a norma dos pesos para o valor desejado,  $\|\mathbf{w}_t\|$
2. Reduzir a função do erro

Com base nesses objetivos, é possível estabelecer um ganho da norma,  $\Delta\|\mathbf{w}_{(k)}\|$ , que deverá ser adicionado à norma atual da MLP para que a norma desejada seja então novamente alcançada.

$$\|\mathbf{w}_{(k)}\| + \Delta\|\mathbf{w}_{(k)}\| \leq \|\mathbf{w}_t\| \quad (5.31)$$

A razão de a Equação 5.31 ser descrita na forma de uma inequação se deve ao fato de que o método é iterativo, ou seja, a norma desejada poderá ser alcançada, ou não, em uma única iteração mas, no decorrer das iterações, a norma da MLP irá convergir para a norma desejada. O ganho da norma é então definido por:

$$\Delta\|\mathbf{w}_{(k)}\| \approx \|\mathbf{w}_t\| - \|\mathbf{w}_{(k)}\| \quad (5.32)$$

onde:  $\|\mathbf{w}_{(k)}\| < \|\mathbf{w}_t\|$ , devido ao efeito da redução da norma ( $\Delta\|\mathbf{w}_{(k)}\| > 0$ ).

Analisando a influência de cada elemento do vetor de pesos da rede e do vetor gradiente, os casos que promovem o crescimento da norma podem ser definidos da seguinte forma:

- $w_{ji(k)} > 0, \Delta w_{ji(k)} > 0$  ( $g_{ji(k)} < 0$ )
- $w_{ji(k)} < 0, \Delta w_{ji(k)} < 0$  ( $g_{ji(k)} > 0$ )

O vetor  $n$ -dimensional,  $\mathbf{g}_{(k)}^+$ , pode então ser definido como:

$$g_{ji(k)}^+ = \begin{cases} g_{ji(k)}, & \text{Se } w_{ji(k)} \cdot g_{ji(k)} < 0 \\ 0, & \text{caso contrário} \end{cases} \quad (5.33)$$

Uma vez que:

$$\|\mathbf{w}_{(k+1)}\| = \|\mathbf{w}_{(k)}\| + \Delta \|\mathbf{w}_{(k)}\| \quad (5.34)$$

substituindo a Equação 5.19 em 5.34 e considerando-se o vetor  $\mathbf{g}_{(k)}^+$ , tem-se:

$$\sqrt{\left[ \sum_{ji} w_{ji(k)}^2 - 2 \cdot \alpha \cdot \sum_{ji} (w_{ji(k)} \cdot g_{ji(k)}^+) + \alpha^2 \cdot \sum_{ji} (g_{ji(k)}^+)^2 \right]} = \|\mathbf{w}_{(k)}\| + \Delta \|\mathbf{w}_{(k)}\| \quad (5.35)$$

Considerando a Equação 5.31 e sendo a relação expressa pela Equação 5.34 aplicada, tem-se:

$$\sqrt{\left[ \sum_{ji} w_{ji(k)}^2 - 2 \cdot \alpha \cdot \sum_{ji} (w_{ji(k)} \cdot g_{ji(k)}^+) + \alpha^2 \cdot \sum_{ji} (g_{ji(k)}^+)^2 \right]} \leq \|\mathbf{w}_t\| \quad (5.36)$$

Para permitir uma maior flexibilidade do algoritmo para o ajuste da norma em relação ao valor desejado,  $\|\mathbf{w}_t\|$ , será considerada uma variância,  $\varphi/2$ , de tal forma que a norma desejada será definida por uma variável  $\zeta$ , calculada da seguinte forma:

$$\zeta = \|\mathbf{w}_t\| + \frac{\varphi}{2} \quad (5.37)$$

Então, a Equação 5.36 assume a forma:

$$\sqrt{\left[ \sum_{ji} w_{ji(k)}^2 - 2 \cdot \alpha \cdot \sum_{ji} (w_{ji(k)} \cdot g_{ji(k)}^+) + \alpha^2 \cdot \sum_{ji} (g_{ji(k)}^+)^2 \right]} \leq \zeta \quad (5.38)$$

Elevando-se os termos ao quadrado e agrupando-os segundo a taxa de aprendizagem,  $\alpha$ , a seguinte inequação de segundo grau é obtida:

$$\alpha^2 \left[ \sum_{ji} (g_{ji}^+)^2 \right] + \alpha \left[ -2 \cdot \sum_{ji} w_{ji} \cdot g_{ji}^+ \right] + \left[ \sum_{ji} w_{ji}^2 - \zeta^2 \right] \leq 0 \quad (5.39)$$

podendo ser reescrita na forma:

$$a \cdot \alpha^2 + b \cdot \alpha + c \leq 0 \quad (5.40)$$

onde

$$a = \sum_{ji} (g_{ji}^+)^2,$$

$$b = -2 \cdot \sum_{ji} w_{ji} \cdot g_{ji}^+$$

$$c = \sum_{ji} w_{ji}^2 - \zeta^2$$

Analisando os coeficientes da equação de 2<sup>o</sup> grau, pode-se concluir que:

1.  $a \geq 0$
2.  $b \geq 0$ . pois o subconjunto em questão:  $\{w_{ji} \cdot g_{ji} < 0\}$ , garante tal afirmação.
3.  $c \leq 0$ . pois  $\zeta^2 > \sum_{ji} w_{ji}$  (ver Equação 5.37). O objetivo nessa etapa do treinamento é elevar a norma a um valor próximo do desejado.

Uma vez que as soluções complexas associadas à Equação 5.39 não serão tratadas, a parábola resultante apresenta uma convexidade positiva.

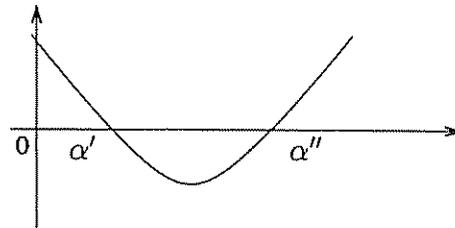


Figura 5.5: Formato da parábola para  $a > 0$

Uma vez identificado o comportamento da parábola ilustrada na Figura 5.5, os valores para o ganho  $\alpha$  que satisfazem a Equação 5.39 estão situados no intervalo entre as duas raízes:  $\alpha'$  e  $\alpha''$  (onde:  $\alpha'' > \alpha'$ ). Sendo o valor do ganho positivo uma condição necessária,  $\alpha > 0$ , é preciso que pelo menos uma das raízes seja real e positiva. Os intervalos para a escolha do ganho  $\alpha$  são então definidos da seguinte forma:

1. Se:  $\alpha' < 0$  e  $\alpha'' > 0$ ,  $\alpha \in \{0 < \alpha < \alpha''\}$ ,
2. Senão se:  $\alpha' > 0$  e  $\alpha'' > 0$ ,  $\alpha \in \{\alpha' < \alpha < \alpha''\}$ ,
3. Caso contrário:  $\alpha \in \{\emptyset\}$ , ajuste não é realizado.

Uma vez que o intervalo de soluções para o parâmetro  $\alpha$  foi encontrado, o método de escolha do ganho final que será utilizado para a correção dos pesos é semelhante àquele descrito no processo de decrescimento da norma: cinco valores são escolhidos no intervalo

e o valor que proporcionar a redução da função do erro é então utilizado. Caso nenhum dos valores escolhidos seja capaz de reduzir o erro, ou se existirem raízes complexas, a correção dos pesos não será realizada.

#### 5.4.1.3 Descrição do Algoritmo

O método de decrescimento e crescimento da norma é realizado em um loop que é finalizado quando um número máximo de iterações é alcançado ou segundo um critério de erro. A vantagem desse método consiste na convergência mais rápida do erro em relação ao algoritmo SMC-MOBJ padrão e na independência da especificação do ganho  $\alpha$ , ou do termo de restrição da norma,  $\beta$ . O único parâmetro a ser ajustado é o intervalo de variação para a norma,  $\varphi$ . Entretanto, o algoritmo só é funcional quando a norma desejada é inicialmente alcançada. Isso é possível através da multiplicação dos pesos da MLP por uma constante (Equações 5.10 e 5.11). Dessa forma, o algoritmo é ideal para o levantamento do conjunto Pareto uma vez que permite localizar, para uma dada norma, o menor valor do erro possível. Por outro lado, o método não é capaz de convergir para soluções arbitrárias, que não se encontram no conjunto Pareto-ótimo.

**Algoritmo 1** Controle do Ganho  $\alpha$  através da Restrição da Norma dos Pesos

---

Inicializa\_os\_pesos\_da\_rede\_randomicamente( $w_{ji(k)}$ );  
 $\varphi \leftarrow$  Largura da faixa de operação da norma;  
{ Ajusta a norma dos pesos para a norma desejada:  $\|w_t\|$  }  
 $c \leftarrow \frac{\|w_t\|}{\|w_{(k)}\|}$ ;  
 $w_{ji(k)} \leftarrow c \cdot w_{ji(k)}$ ;  
**Para** passos = 1 : MAX\_PASSOS, **faça**  
  Apresentar\_todo\_o\_conjunto\_de\_treinamento\_e\_calcular\_os\_gradientes();  
  { ( $g_{ji(k)}$ ) - BATCH }  
  { Casos onde ocorre o decrescimento da norma }  
   $w_{ji(k)}^+ \leftarrow$  pesos onde:  $j, i \in \{(w_{ji(k)} \cdot g_{ji(k)} > 0)\}$ ;  
   $g_{ji(k)}^+ \leftarrow$  gradientes onde:  $j, i \in \{(w_{ji(k)} \cdot g_{ji(k)} > 0)\}$ ;  
   $\alpha_{max} \leftarrow \frac{2 \sum_{ji} w_{ji(k)}^+ g_{ji(k)}^+}{\sum_{ji} (g_{ji(k)}^+)^2}$ ;  
   $\alpha \leftarrow \alpha_{max}$ ;  
  contador  $\leftarrow$  0;  
  **Enquanto** ( $E_{(k+1)} > E_{(k)}$ )|(contador < MAX\_CONT)), **faça**  
     $w_{ji(k+1)} \leftarrow w_{ji(k)} - \alpha \cdot g_{ji(k)}^+$ ;  
     $\alpha \leftarrow \alpha/10$ ; { Define um novo valor do ganho  $\alpha$  dentro do intervalo especificado }  
    contador  $\leftarrow$  contador + 1;  
    calcula o valor de  $E_{(k+1)}$ ;  
  **fim Enquanto**  
  { Casos onde ocorre o crescimento da norma }  
   $w_{ji(k)}^- \leftarrow$  pesos onde:  $j, i \in \{(w_{ji(k)} \cdot g_{ji(k)} < 0)\}$ ;  
   $g_{ji(k)}^- \leftarrow$  gradientes onde:  $j, i \in \{(w_{ji(k)} \cdot g_{ji(k)} < 0)\}$ ;  
   $[\alpha', \alpha''] =$  raízes  $\left\{ \alpha^2 \left[ \sum_{ji} (g_{ji}^-)^2 \right] + \alpha \left[ -2 \cdot \sum_{ji} w_{ji}^- \cdot g_{ji}^- \right] + \left[ \sum_{ji} (w_{ji}^-)^2 - \zeta^2 \right] = 0 \right\}$ ;  
   $\alpha_{max} \leftarrow$  maior\_raiz\_real\_positiva( $\alpha', \alpha''$ ):  
   $\alpha \leftarrow \alpha_{max}$ ;  
  contador  $\leftarrow$  0;  
  **Enquanto** ( $E_{(k+1)} > E_{(k)}$ )|(contador < MAX\_CONT)), **faça**  
     $w_{ji(k+1)} \leftarrow w_{ji(k)} - \alpha \cdot g_{ji(k)}^-$ ;  
     $\alpha \leftarrow \alpha/10$ ; { Define um novo valor do ganho  $\alpha$  dentro do intervalo especificado }  
    contador  $\leftarrow$  contador + 1;  
    calcula o valor de  $E_{(k+1)}$ ;  
  **fim Enquanto**  
  **Se** ( $E_{(k)} < \text{ERRO\_MÍNIMO}$ ), **então**  
    interromper\_loop();  
  **fim Se**  
**fim Para**

---

## 5.5 Conclusões do Capítulo

A capacidade do método de se deslocar no plano de soluções possibilita definir trajetórias, nesse plano, e estimar soluções arbitrárias e soluções eficientes. Através da especificação de um objetivo nulo para a função do erro, o algoritmo é capaz de alcançar soluções específicas localizadas sob o conjunto Pareto-ótimo ou próximas do mesmo.

A implementação de um método multi-objetivo utilizando o algoritmo de Levenberg-Marquardt (MOBJ-SMC-LM) foi possível através da especificação de um termo de restrição da norma à equação padrão de ajuste dos pesos. O método permite obter soluções Pareto em um número menor de iterações.

Um método multi-objetivo que utiliza a decomposição dos gradientes da função do erro também foi apresentado (MOBJ-BCKPRP). A função norma é utilizada para definir os intervalos para os valores das taxas de aprendizado. A decomposição e controle das taxas de aprendizado permitem minimizar o erro restringindo a solução a uma região onde a norma é praticamente constante.

A Tabela 5.1 apresenta as principais características do algoritmo SMC-MOBJ proposto bem como as variações apresentadas neste capítulo.

Tabela 5.1: Características do Algoritmo Proposto (SMC-MOBJ) e variações

| Algoritmo    | Características  |
|--------------|--|
| SMC-MOBJ     | Utiliza o controle por modos deslizantes para direcionar o estado definido pelo erro e norma da rede MLP a um estado pré-determinado. O ajuste dos pesos é realizado utilizando-se os valores dos gradientes do erro e da norma no ponto.  |
| MOBJ-SMC-LM  | O ajuste dos pesos é realizado segundo a equação de ajuste do algoritmo de Levenberg-Marquardt, que utiliza uma aproximação quadrática da superfície do erro, permitindo um aumento na velocidade de convergência do método. A restrição da norma dos pesos é realizada através da adição do termo de restrição da norma (SMC-MOBJ) à equação de ajuste.   |
| MOBJ-BCKPROP | O método propõe a decomposição do vetor gradiente em duas componentes: a primeira permite reduzir a norma dos pesos e a segunda, permite aumentá-la. O ajuste é então realizado utilizando-se, alternadamente, essas componentes mas, através de uma análise matemática, o método permite obter as taxas de aprendizado associadas a cada componente de forma que a variação da norma dos pesos seja controlada. |



## Capítulo 6

# Aplicação do Método Proposto: Simulações e Resultados

Neste capítulo são apresentados os principais resultados obtidos a partir de simulações utilizando o algoritmo proposto e suas variações. Algumas simulações foram inicialmente realizadas utilizando conjuntos de dados provenientes dos resultados alcançados por Teixeira [Teixeira, 2001]. Esses dados permitiram testar o algoritmo quanto à sua capacidade de gerar soluções eficientes. O algoritmo também foi analisado em relação à sua capacidade de deslocamento no plano de soluções.

Dados provenientes de problemas reais também foram utilizados para demonstrar a capacidade do algoritmo de encontrar soluções com boa capacidade de generalização. Resultados de simulações utilizando algoritmos conhecidos como: Backpropagation, Modos Deslizantes (Parma), Levenberg-Marquardt, RPROP, Early Stopping, MOBJ e Weight Decay são apresentados no Apêndice B e permitem comparar os resultados obtidos.

Também são apresentados resultados nos quais o treinamento de redes MLPs é realizado mantendo um dos funcionais, erro ou norma, constante (soluções fora do conjunto Pareto) a fim de exemplificar esse tipo de comportamento.

### 6.1 Descrição dos Problemas

Os problemas propostos para a validação do algoritmo SMC-MOBJ podem ser subdivididos em três categorias:

- Classificação
- Regressão
- Predição

**Classificação** A primeira base de dados utilizada no processo de classificação foi obtida a partir de um problema de classificação constituído por duas classes cujos padrões

foram amostrados aleatoriamente utilizando duas distribuições normais bivariadas com médias distintas e variâncias iguais. Para os demais problemas, foram utilizadas bases de dados provenientes de casos reais que serão descritas à medida que sejam utilizadas.

**Regressão** Cinco diferentes bases de dados foram utilizadas para avaliar a capacidade da rede de realizar aproximações de funções. As bases de dados utilizadas bem como os pontos referentes ao conjunto Pareto para cada função foram inicialmente obtidas por Teixeira [Teixeira, 2001]. As seguintes funções foram utilizadas:

- Função seno ao qual foi adicionado um ruído gaussiano.
- Função seno amortecido (com ruído gaussiano)
- Função  $\text{sinc}(x)$  (com ruído gaussiano)
- Função  $d(x)$  descrita pela seguinte equação:

$$d(x) = \frac{(x - 2)(2x + 1)}{(1 + x^2)} + \xi \quad (6.1)$$

definida no intervalo  $[-8, 12]$ , onde  $\xi$  representa um ruído gaussiano com média zero e variância  $\sigma^2 = 0.2^2$ .

Outras bases de dados utilizadas referem-se a problemas reais como os dados de regressão *Building* dispostas na base de dados PROBEN [Blake and Merz, 1998]

**Predição** Somente uma base de dados foi utilizada nesse processo. A série econômica do Preço do Boi Gordo no Estado de São Paulo foi escolhida para avaliar o desempenho da predição para diferentes pontos do conjunto Pareto. O uso de redes MLPs para a predição dessa série é descrita na literatura [Costa et al., 2000].

Para cada conjunto de dados utilizado, foram definidos o conjunto de treinamento e o conjunto de validação da seguinte forma: 60% dos dados para treinamento e o restante (40%) para validação.

## 6.2 Problemas de Classificação

Para os problemas de classificação, foram utilizadas funções de ativação tangente hiperbólica em ambas as camadas das redes MLPs treinadas. Para a identificação de duas classes, apenas uma saída foi utilizada ao passo que, para a classificação de mais de duas classes, foram associadas  $n$  saídas, onde  $n$  representa o número de classes a serem identificadas. À saída de maior valor é atribuída a classe correspondente.

### 6.2.1 Separação de duas Classes no Espaço Bi-dimensional

Os padrões referentes a duas classes foram amostrados aleatoriamente segundo a distribuição normal bivariada. As classes possuem médias distintas ( $\mu_{c1} = [2, 2]$ ,  $\mu_{c2} = [4, 4]$ ) e mesma variância ( $\sigma_{c1}^2 = \sigma_{c2}^2 = 1.5^2$ ). Para a primeira classe, a resposta desejada foi definida como  $d = 1$  e para a segunda,  $d = -1$ . Uma vez treinada a rede, que irá apresentar uma resposta contida no intervalo:  $(-1, +1)$ , utilizou-se uma função de limiar de forma a saturar a saída:

$$Y(x) = \begin{cases} +1, & \text{se } x \geq 0 \\ -1, & \text{se } x < 0 \end{cases} \quad (6.2)$$

A Figura 6.1 mostra os resultados obtidos a partir da geração do conjunto Pareto. A Figura 6.1 (b) demonstra, no plano, as soluções Pareto encontradas pelos algoritmos MOBJ e SMC-MOBJ. As diferenças de erro entre as soluções de mesma norma pode ser analisada segundo o critério de parada do método SMC-MOBJ, que é definido por um número máximo de iterações por solução (ou ponto). Como pode ser observado, a melhor solução para esse caso subdivide o espaço de padrões através de uma reta. Por se tratar de uma solução linearmente separável, a rede com melhor capacidade de generalização obtida a partir do Pareto, apresenta um baixo valor da norma dos pesos. Conforme será visto posteriormente, redes ou nodos com valores reduzidos de norma tendem a apresentar um comportamento linear. Esse fenômeno está relacionada à função de ativação.

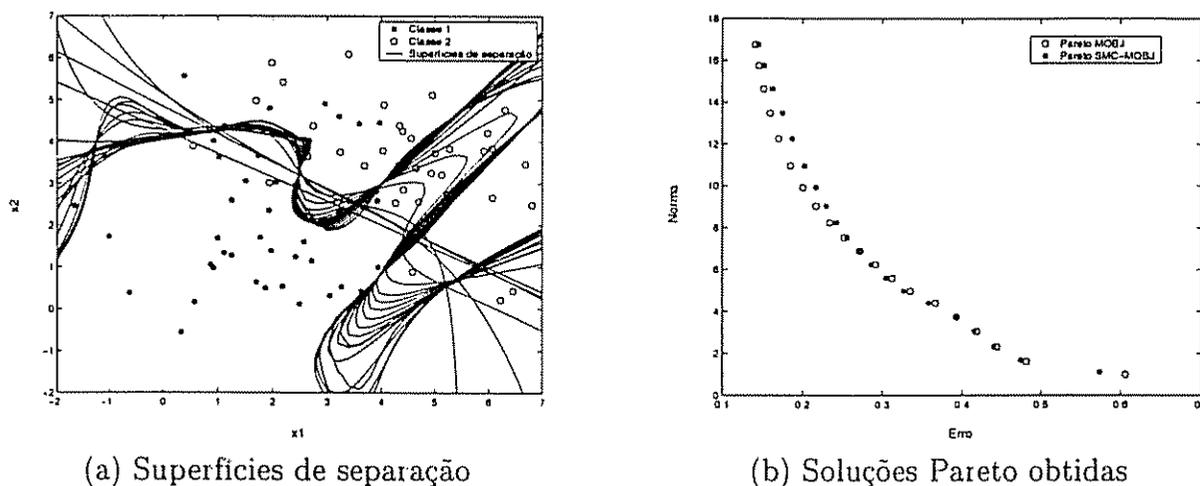


Figura 6.1: Classificação de duas classes com distribuição bivariada

### 6.2.2 Classificação: Adult Database

A seguir, são apresentados os resultados de classificação de um conjunto de dados denominado *Adult Database* [Blake and Merz, 1998], os dados são provenientes de um estudo sócio-econômico, classificando indivíduos segundo a sua renda anual. Por se tratar de um banco de dados muito utilizado para *benchmark*, resultados obtidos com o método *Support*

*Vector Machine* (SVMs) [Mangasarian and Musicant, 1999] foram utilizados para analisar a capacidade de generalização do algoritmo proposto (SMC-MOBY).

Tabela 6.1: Número de pares entrada-saída (padrões) dos conjuntos de treinamento e validação

| índice do conjunto | Número de padrões para o conjunto de treinamento | Número de padrões para o conjunto de validação |
|--------------------|--|--|
| 1                  | 1605   | 30958  |
| 2                  | 2265   | 30298  |
| 3                  | 3185   | 29378  |
| 4                  | 4781   | 27782  |
| 5                  | 6414   | 26149  |
| 6                  | 11222  | 21341  |
| 7                  | 16102  | 16461  |
| 8                  | 22698  | 9865   |

Os dados para treinamento encontram-se subdivididos segundo a quantidade de pares ou padrões entrada-saída. Ao todo foram utilizados 8 (oito) conjuntos, cada conjunto possuindo os respectivos dados de treinamento e validação com tamanhos variados. A distribuição desses conjuntos é descrita na Tabela 6.1

Para cada conjunto de treinamento, um conjunto de soluções pertencentes à curva Pareto-ótimo foi obtido a partir do uso do algoritmo SMC-MOBY proposto e, utilizando os dados de validação, um ponto do Pareto com a maior taxa de generalização ou a solução com o menor erro de validação foi escolhida. Foram então obtidas oito soluções finais (uma para cada conjunto de treinamento). Em seguida, os resultados foram agrupados em uma tabela na qual foram inseridos os resultados obtidos para o algoritmo SOR (Successive Overrelaxation - variação do método SVM) e SVM<sup>l</sup> (Support Vector Machine *light*) [Mangasarian and Musicant, 1999].

Tabela 6.2: Resultados do problema de classificação com os dados provenientes do *Adult Database*

| índice do conjunto | Resultados de classificação (% acertos) |                    |          |        |        |
|--------------------|---|--------------------|----------|--------|--------|
|                    | SOR %                                   | SVM <sup>l</sup> % | SMC-MOBY |        |        |
|                    |   |                    | %        | Erro   | Norma  |
| 1                  | 84.06                                   | 84.25              | 84.38    | 0.4222 | 1.854  |
| 2                  | 84.24                                   | 84.43              | 84.66    | 0.4416 | 2.241  |
| 3                  | 84.13                                   | 84.40              | 84.58    | 0.4287 | 1.9715 |
| 4                  | 84.28                                   | 84.47              | 84.7     | 0.4127 | 2.4144 |
| 5                  | 84.3                                    | 84.43              | 84.71    | 0.4118 | 2.4617 |
| 6                  | 84.37                                   | 84.68              | 84.95    | 0.4184 | 2.6715 |
| 7                  | 84.62                                   | 84.83              | 85.11    | 0.4209 | 2.2201 |
| 8                  | 85.06                                   | 85.17              | 85.4     | 0.4218 | 2.1744 |

Segundo a tabela 6.2, o algoritmo proposto apresentou uma capacidade de generalização compatível e ligeiramente superior em relação aos métodos SVM e SOR, o que significa que o método proposto foi capaz de obter, neste caso, uma solução final com boa generalização, uma vez que os métodos SVM e SOR são capazes de obter soluções para problemas de classificação com elevadas taxas de acertos.

Alguns algoritmos de minimização da função do erro (derivados do *backpropagation*), como o Resilient Propagation (Rprop) [Riedmiller and Braun, 1993], Scaled Conjugate Gradient (SCG) [Moller, 1993] e o próprio algoritmo *backpropagation* (Backprop), foram utilizados para a análise da capacidade de generalização do método proposto, bem como das características das funções de erro e norma alcançadas para cada caso. Para cada algoritmo, um conjunto composto por 10 redes de mesma topologia foi treinado. Os resultados obtidos encontram-se subdivididos em três tabelas: a primeira tabela apresenta os valores médios e os desvios para a capacidade de generalização das redes (Tabela 6.3), ou seja, a porcentagem de acertos para o conjunto de validação, a segunda tabela fornece os valores para a função de erro (Tabela 6.4) e a terceira, a norma dos pesos com os respectivos valores médios e desvios (Tabela 6.5).

Tabela 6.3: Tabela de porcentagem de acertos: médias e desvios

| índice<br>do<br>conjunto | Validação do Treinamento (% acertos) |        |       |        |       |        |
|--------------------------|--------------------------------------|--------|-------|--------|-------|--------|
|                          | Backprop                             |        | SCG   |        | Rprop |        |
|                          | média                                | desvio | média | desvio | média | desvio |
| 1                        | 82.62                                | 0.3845 | 80.79 | 1.5533 | 80.72 | 1.4068 |
| 2                        | 82.69                                | 1.4635 | 80.94 | 1.5894 | 81.15 | 1.1149 |
| 3                        | 82.97                                | 1.033  | 80.17 | 3.5777 | 81.24 | 1.3092 |
| 4                        | 81.50                                | 2.1015 | 78.48 | 5.0881 | 81.75 | 2.1880 |
| 5                        | 83.71                                | 0.2624 | 79.89 | 2.9771 | 83.01 | 0.6717 |
| 6                        | 82.47                                | 3.3113 | 83.52 | 1.0611 | 82.27 | 1.9242 |
| 7                        | 84.40                                | 0.3286 | 83.52 | 1.4837 | 83.52 | 1.8806 |
| 8                        | 84.62                                | 0.2491 | 84.00 | 1.9710 | 84.10 | 2.0286 |

Tabela 6.4: Erro obtido para o conjunto de treinamento ( $E = \sum e^2$ ): médias e desvios

| índice<br>do<br>conjunto | Erro da MLP para o conjunto de treinamento |        |        |        |        |        |
|--------------------------|--|--------|--------|--------|--------|--------|
|                          | Backprop                                   |        | SCG    |        | Rprop  |        |
|                          | média                                      | desvio | média  | desvio | média  | desvio |
| 1                        | 0.4863                                     | 0.0319 | 0.4997 | 0.1192 | 0.4846 | 0.0818 |
| 2                        | 0.5371                                     | 0.0333 | 0.5245 | 0.1138 | 0.4859 | 0.1435 |
| 3                        | 0.4902                                     | 0.0230 | 0.5508 | 0.1967 | 0.4428 | 0.1191 |
| 4                        | 0.4887                                     | 0.0188 | 0.6103 | 0.1801 | 0.5085 | 0.1221 |
| 5                        | 0.4598                                     | 0.0136 | 0.5379 | 0.1263 | 0.4379 | 0.0415 |
| 6                        | 0.5077                                     | 0.1311 | 0.4377 | 0.0298 | 0.4861 | 0.0928 |
| 7                        | 0.4406                                     | 0.0114 | 0.4504 | 0.0514 | 0.4356 | 0.0368 |
| 8                        | 0.4605                                     | 0.0221 | 0.4530 | 0.0636 | 0.4366 | 0.0389 |

Tabela 6.5: Norma da MLP obtida ( $\|w\|$ ): médias e desvios

| índice<br>do<br>conjunto | Norma da MLP obtida |        |       |        |         |        |
|--------------------------|---------------------|--------|-------|--------|---------|--------|
|                          | Backprop            |        | SCG   |        | Rprop   |        |
|                          | média               | desvio | média | desvio | média   | desvio |
| 1                        | 107.86              | 10.71  | 41.95 | 32.20  | 30827.0 | 1248.0 |
| 2                        | 126.0               | 12.61  | 70.07 | 37.21  | 30941.0 | 1200.0 |
| 3                        | 118.24              | 11.17  | 78.41 | 64.14  | 22839.0 | 4754.0 |
| 4                        | 129.91              | 36.05  | 36.04 | 20.68  | 10737.0 | 1924.0 |
| 5                        | 75.97               | 3.32   | 23.87 | 18.40  | 13599.0 | 2778.0 |
| 6                        | 52.94               | 29.42  | 13.90 | 4.22   | 12159.0 | 2257.0 |
| 7                        | 19.77               | 2.0    | 11.04 | 2.50   | 7081.1  | 1731.8 |
| 8                        | 40.0                | 42.73  | 11.49 | 6.51   | 7566.0  | 2625.0 |

Em relação ao algoritmo proposto (SMC-MOBY), a capacidade de generalização dos algoritmos testados é inferior mesmo em relação aos métodos SVM e SOR (Tabela 6.2). Outro aspecto importante é o valor da norma alcançado. Apesar de minimizar somente o erro, o Rprop apresentou valores de norma muito elevados, seguidos pelo algoritmo backpropagation e SCG (Scaled Conjugate Gradient) [Moller, 1993]. A solução obtida pelo algoritmo SMC-MOBY apresentou um valor final da norma dos pesos inferior em relação aos demais resultados mas, uma capacidade de generalização superior.

### 6.3 Regressão

O algoritmo SMC-MOBY foi utilizado para gerar o conjunto Pareto-ótimo e trajetórias no plano de soluções para os seguintes casos: função seno, *sinc*, seno amortecido e função

$d(x)$ . Os resultados foram agrupados de forma a comparar o conjunto Pareto gerado pelo algoritmo SMC-MOBY com os resultados obtidos pelo algoritmo MOBY. As soluções de regressão obtidas também são demonstradas. Todas as redes utilizadas foram superdimensionadas para demonstrar a capacidade do método de obter soluções com boa capacidade de generalização. As redes apresentam a seguinte topologia: 1-50-1 com saída linear e funções tangente hiperbólica na camada escondida.

A Figura 6.2 mostra os resultados obtidos a partir da função  $d(x)$ . O conjunto Pareto obtido, bem como as soluções de regressão são muito semelhantes às obtidas pelo algoritmo MOBY, o que exemplifica a capacidade do algoritmo proposto de gerar as soluções pertencentes ao conjunto Pareto-ótimo.

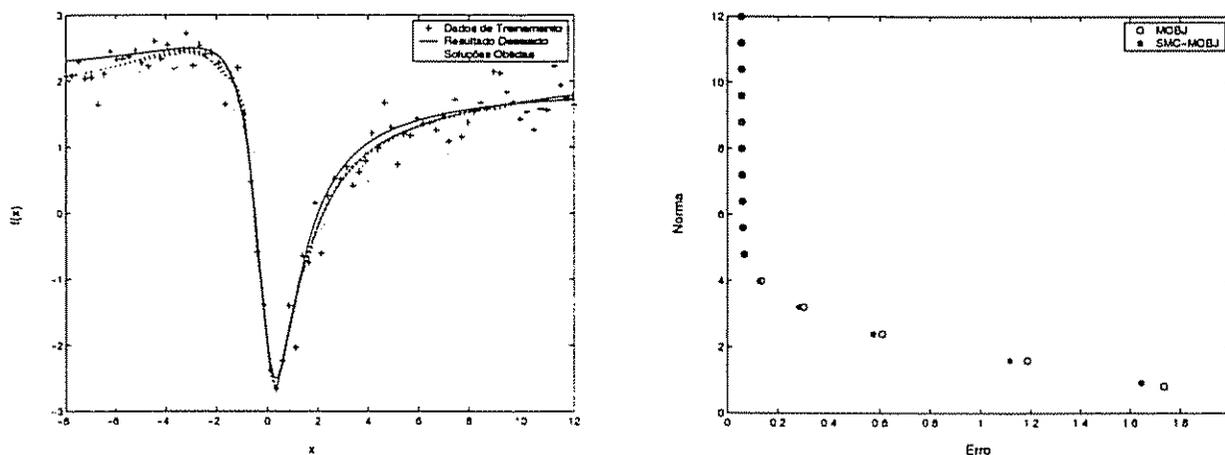


Figura 6.2: Resultados obtidos para a regressão da função  $d(x)$

A função  $d(x)$  foi utilizada porque apresenta regiões de resposta suave em seus extremos seguida de uma região central íngreme. As respostas obtidas em relação a essas regiões podem ser subdivididas em dois grupos: soluções de baixa norma, incapazes de modelar a função e soluções de normas mais elevadas capazes de modelar ambas as regiões. Tal comportamento justifica o fato de que controlando a norma dos pesos, a complexidade da rede pode ser ajustada. Nesse caso, para valores reduzidos de norma, a rede apresenta um comportamento de *underfitting*.

A seguir, a função *seno* é utilizada. O comportamento demonstrado na Figura 6.3 é semelhante ao descrito para a função  $d(x)$  no qual uma rede superdimensionada é capaz de obter uma resposta com alta capacidade de generalização através do controle da norma dos pesos.

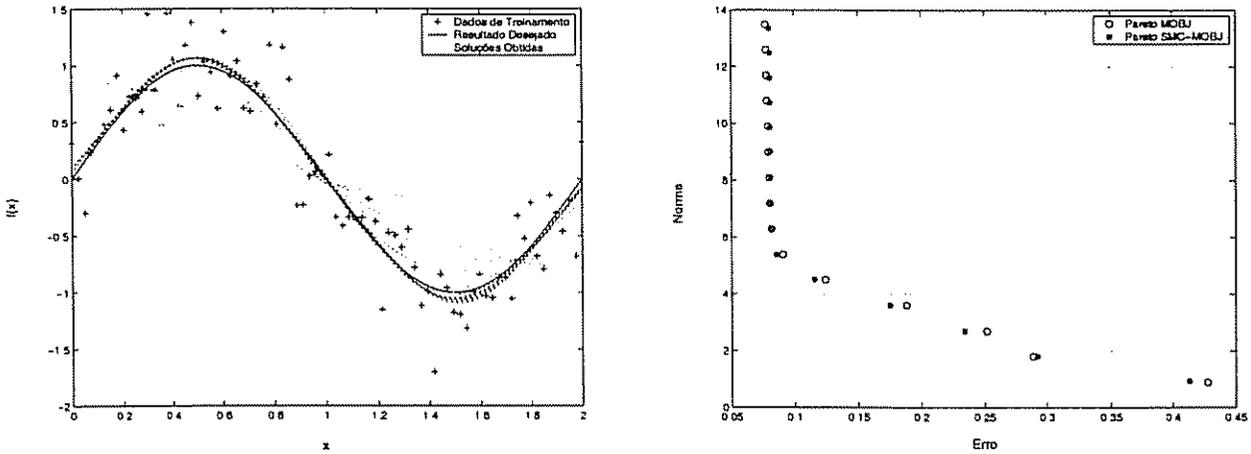


Figura 6.3: Resultados obtidos para a regressão da função seno

As funções *sinc* e *seno amortecido* são mais complexas em relação às funções  $d(x)$  e *seno*. Os resultados obtidos para a regressão dessas funções utilizando redes MLPs são apresentados nas Figuras 6.4 e 6.5

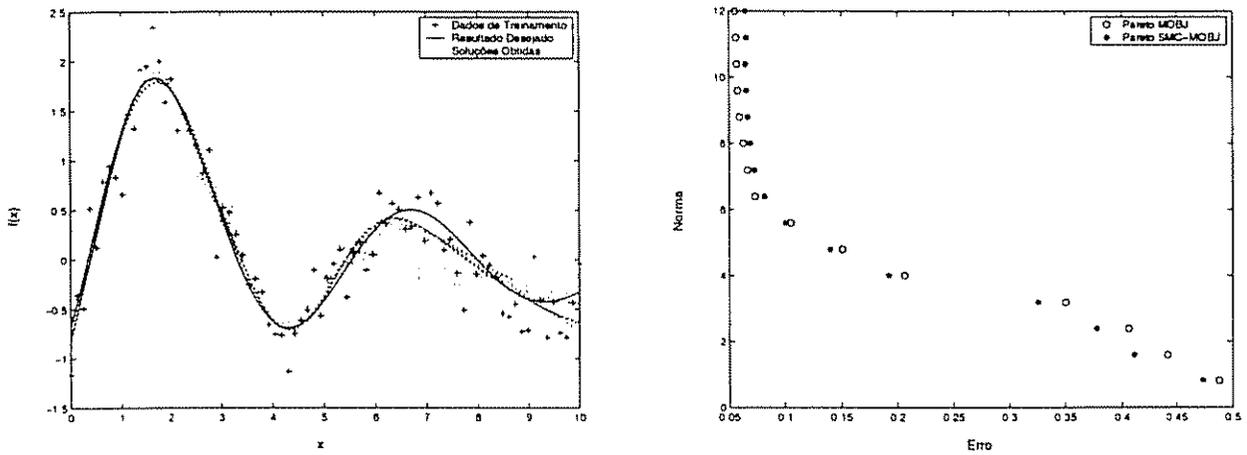


Figura 6.4: Resultados obtidos para a regressão da função seno amortecido

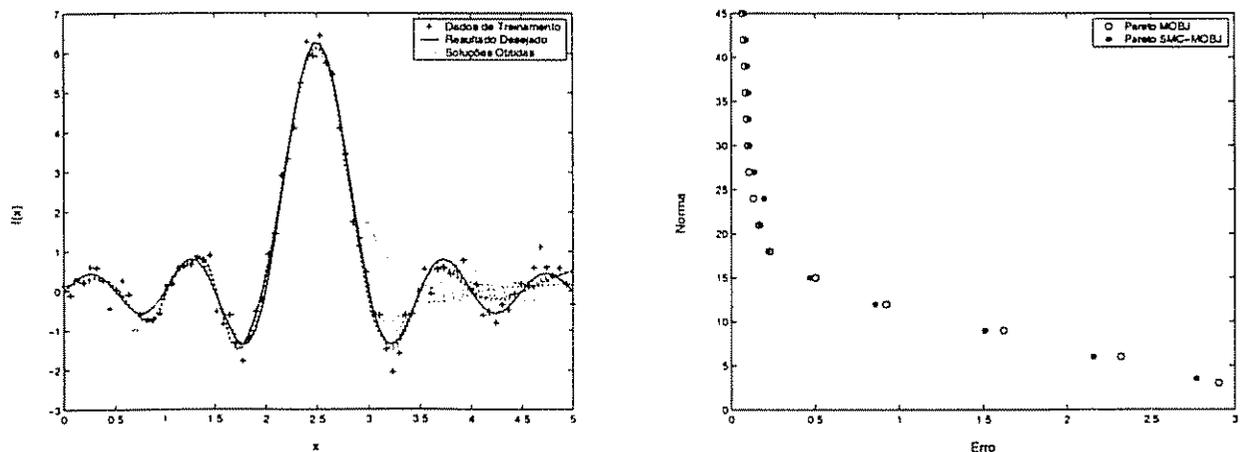


Figura 6.5: Resultados obtidos para a regressão da função *sinc*

Os resultados demonstram que, para valores elevados de norma, o algoritmo SMC-MOBJ foi capaz de convergir para o ponto desejado mas não foi capaz de alcançá-lo. Isso se deve ao fato de que o algoritmo utiliza o método do gradiente descendente, que apresenta uma convergência mais lenta se comparado com algoritmos mais complexos. Em virtude dessa característica, o número de iterações utilizadas não foi suficiente para a completa convergência do algoritmo ao ponto desejado.

É possível perceber um aumento da norma dos pesos no conjunto Pareto obtido para esses casos, em especial para a função *sinc*. Conforme será descrito na seção 7.3, nodos que apresentam função de ativação tangente hiperbólica na camada escondida dependem do valor da norma dos pesos para que esse nodo *opere* na região não-linear da função de ativação. Conseqüentemente, para uma rede super-dimensionada, quanto maior o valor da norma, maior será a quantidade de nodos operando nessa região. Tal característica, uma vez treinada a rede utilizando um algoritmo multi-objetivo e selecionada a solução com a melhor generalização, fornece uma medida da *complexidade* do modelo neural obtido e, conseqüentemente, da função geradora.

## 6.4 Controle de trajetórias no Espaço Solução

Uma vez definido um ponto objetivo no plano de soluções, o algoritmo proposto (SMC-MOBJ) é capaz de convergir em direção a esse ponto. Tal comportamento agrega ao algoritmo a capacidade de *desenhar* trajetórias no plano de soluções a partir de um conjunto de pontos pré-estabelecidos. Com isso, é possível analisar o comportamento de soluções específicas, como soluções com normas muito próximas ou que apresentam um mesmo erro.

O objetivo dessa seção é exemplificar a capacidade do algoritmo de deslocar-se no plano de soluções. Para isso, diversos conjuntos de pontos definindo trajetórias no plano de soluções foram utilizados.

### 6.4.1 Controle de Trajetórias na Região Factível

Os resultados que se seguem foram obtidos a partir do uso de trajetórias pré-estabelecidas na região factível. O objetivo dessas simulações consiste em verificar a capacidade do algoritmo SMC-MOBJ de se deslocar nessa região.

Foram utilizados os dados de treinamento referentes ao problema de classificação bi-dimensional e a função *sinc*. Os resultados podem ser verificados na Figura 6.6

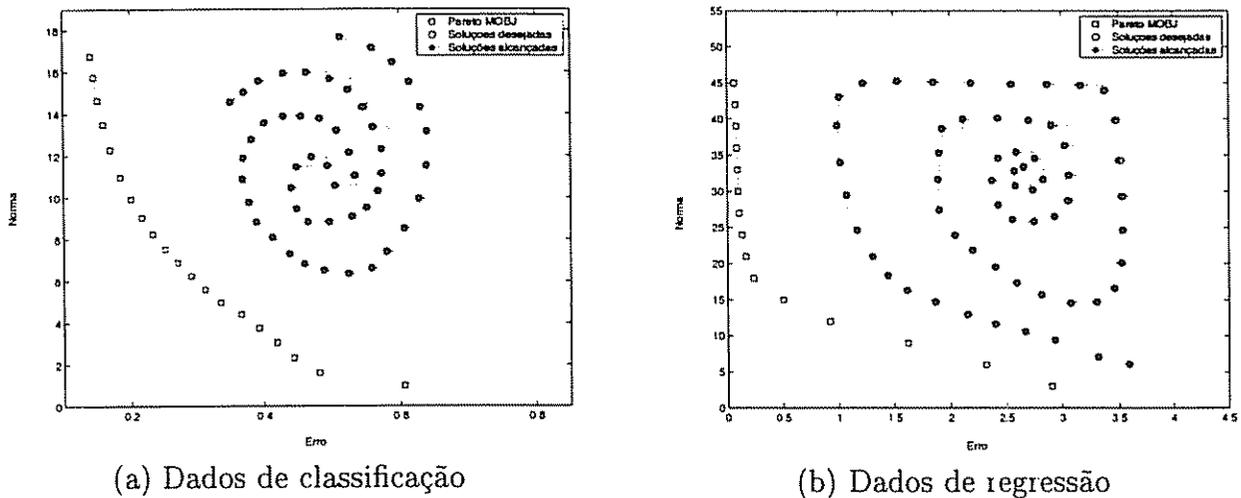


Figura 6.6: Exemplos de trajetórias na região factível

É possível verificar, através das curvas Pareto obtidas para os casos de regressão (seno, sinc) e classificação, que as características da curva Pareto-ótimo: inclinação, posição da solução ótima, estão relacionadas ao conjunto de treinamento em questão. Mas, uma vez conhecida a característica da curva Pareto como sendo o limite entre as soluções factíveis e as soluções não-existent e aplicando-se a Teoria de Modos Deslizantes para determinar os ganhos  $\alpha$  e  $\beta$  referentes à equação de ajuste dos pesos, o método SMC-MOBJ é capaz de convergir para qualquer solução factível mapeada no plano de soluções.

A seguir, foram elaboradas algumas trajetórias no plano de soluções que ultrapassam a curva Pareto convergindo para a solução *utópica*, não-existente (erro e norma nulos). Os resultados podem ser analisados através da Figura 6.7 onde, apesar de o algoritmo alcançar pontos com valores de erro mais próximos do eixo das ordenadas do que as soluções MOBJ, a existência de uma *barreira* entre a região factível e o conjunto de soluções não-existent é verificada.

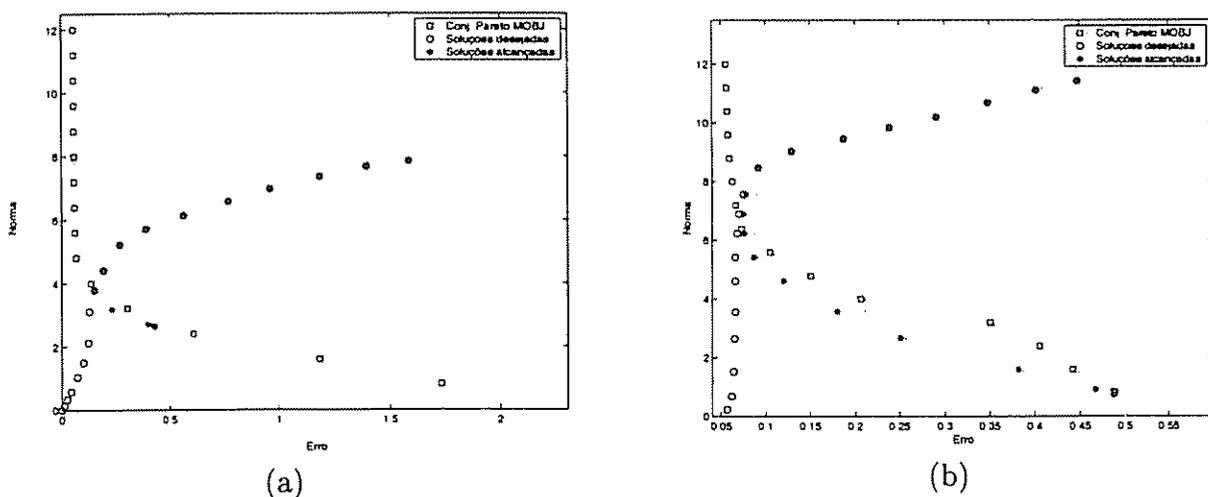


Figura 6.7: Exemplos de trajetórias no Plano de Soluções

Na Figura 6.7 (a), o algoritmo proposto convergiu para um ponto ótimo. A convergência a esse ponto está diretamente relacionada aos ganhos  $\alpha$  e  $\beta$ , especialmente quando se refere à convergência da norma. Valores elevados de  $\beta$  em relação ao ganho  $\alpha$  promovem um aumento da velocidade de convergência da norma em relação à convergência do erro. Tal efeito resulta em um treinamento onde a norma desejada é facilmente alcançada em poucas iterações o que, no caso da Figura 6.7 (b), provocou uma convergência das soluções para pontos do conjunto Pareto muito próximos do valor da norma desejada. Por outro lado, em casos onde o ganho  $\alpha$  é inferior ao ganho  $\beta$ , a convergência do algoritmo SMC-MOBJ para pontos localizados em uma região inferior do conjunto Pareto é prejudicada, (ver Figura 6.7 (a)) provocando uma súbita redução na velocidade de convergência da norma. Em ambos os casos, o uso da Teoria de Modos Deslizantes garante a convergência do erro e da norma mas, a velocidade de convergência em relação às superfícies de deslizamento depende da razão entre esses ganhos.

#### 6.4.2 Análise da Variação do Erro e da Norma a Partir de um Ponto Definido no Conjunto Pareto

O objetivo dos experimentos propostos nesta seção consiste em testar a resposta da rede a partir do ponto de maior generalização escolhido no conjunto Pareto, permitindo a análise de soluções de mesma norma ou mesmo erro em relação ao ponto ótimo. Para isso, foram utilizados os resultados provenientes da aproximação de uma função por uma MLP (dimensão - 1:50:1)

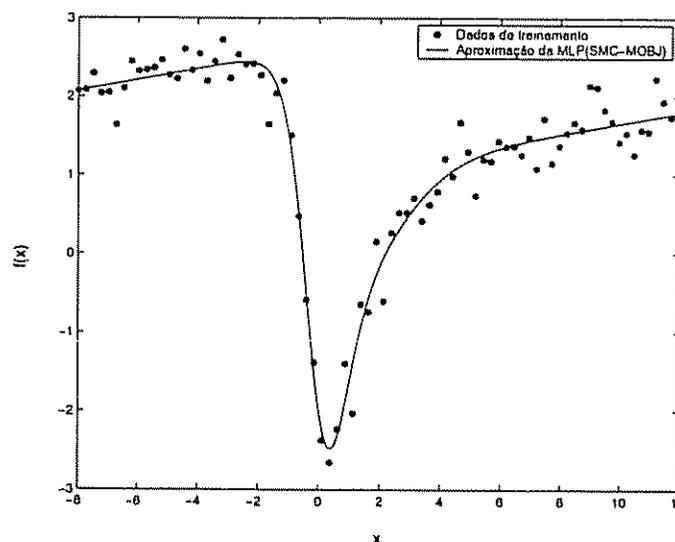


Figura 6.8: Resposta da MLP após treinamento utilizando o algoritmo SMC-MOBJ

A partir do erro ( $E_{(k)}$ ) e da norma ( $\|w_{(k)}\|$ ) obtida para a solução demonstrada na Figura 6.8, foram realizadas duas simulações:

1. A norma foi mantida constante e o valor do erro foi progressivamente elevado para demonstrar a influência do erro na resposta da MLP.
2. O valor do erro foi definido constante e a norma foi progressivamente elevada para demonstrar a influência da norma na resposta da MLP.

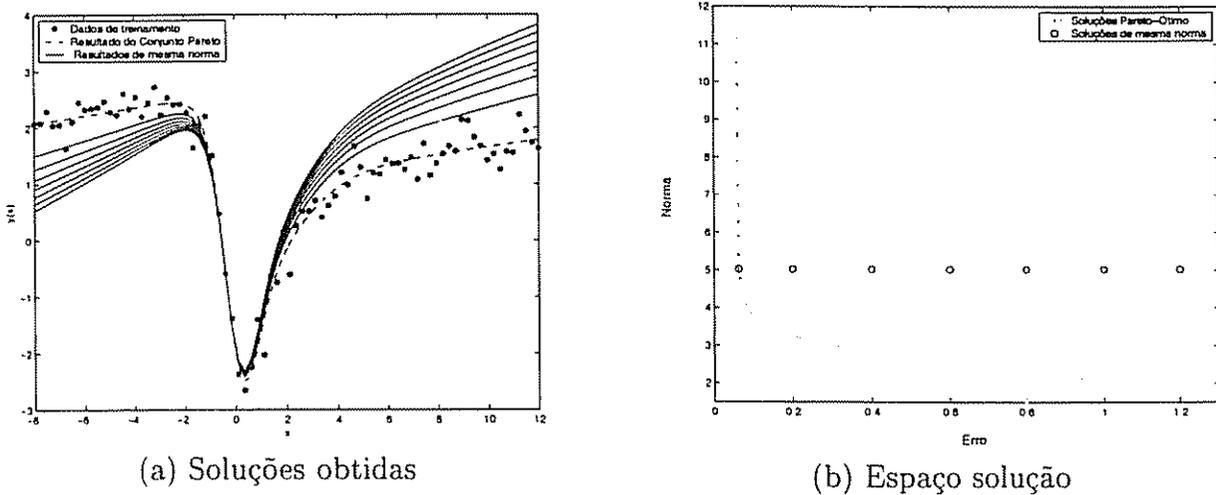


Figura 6.9: Soluções da MLP para  $\|w_{(k)}\|$  constante

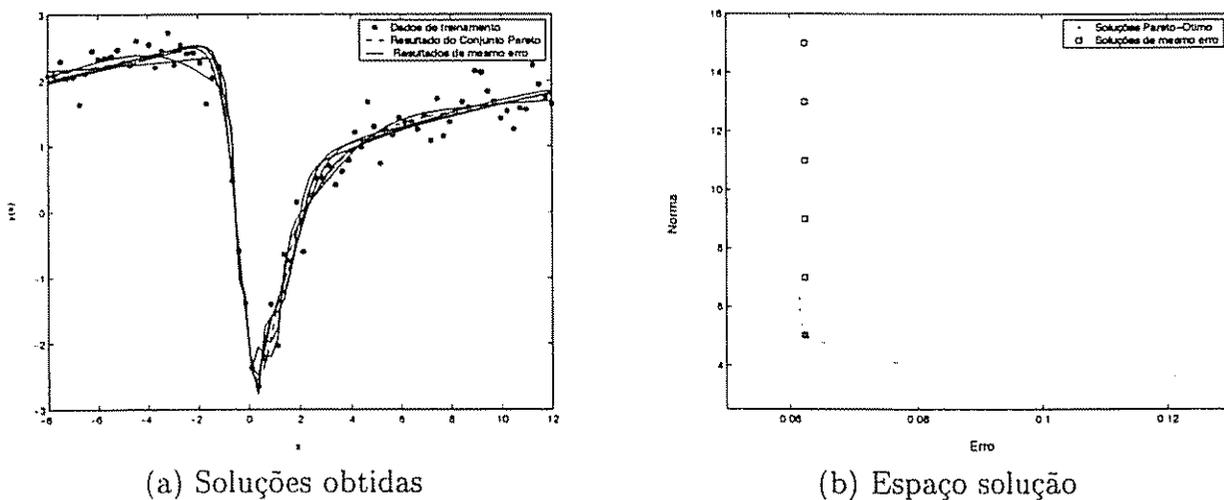


Figura 6.10: Soluções da MLP para  $E_{(k)}$  constante

Os resultados expressos nas Figuras 6.9 e 6.10 demonstram que:

1. Para um mesmo valor de norma, o aumento do erro provoca a perda da capacidade de assimilação da rede (MLP) ao conjunto de treinamento e, como consequência, a perda da capacidade de aproximação da função desejada. Este comportamento pode ser visto na Figura 6.9 (a) na qual a resposta da rede apresentou um deslocamento progressivo, em relação à solução inicial, com o aumento do erro.

2. Para um mesmo valor do erro, o aumento progressivo da norma provoca um *ajuste* da MLP ao conjunto de treinamento. Analisando o resultado da Figura 6.10 (a), percebe-se que à medida em que os valores da norma dos pesos aumentam, a rede ajusta a sua resposta em relação aos dados, minimizando o erro em relação aos dados de determinadas regiões e compensando através de um aumento do erro em relação às demais, de forma que a soma final dos erros quadráticos não é alterada.

Dessa forma, é evidente que, a partir da solução de maior generalização (menor erro de validação), o aumento da norma e posterior ajuste do erro na direção de sua minimização faz com que a rede apresente um comportamento de *overfitting*.

### 6.4.3 Variações do algoritmo multi-objetivo: o algoritmo MOBJ-SMC-LM

Os resultados de regressão utilizando o algoritmo multi-objetivo de Levenberg-Marquardt são apresentados a seguir. Os dados provenientes da função *sinc* foram inicialmente utilizados para definir o conjunto Pareto. Na Figura 6.11 é possível verificar as respostas obtidas bem como o conjunto Pareto estimado. Os resultados de regressão alcançados pelo método MOBJ-SMC-LM são semelhantes àqueles obtidos com o uso do algoritmo MOBJ.

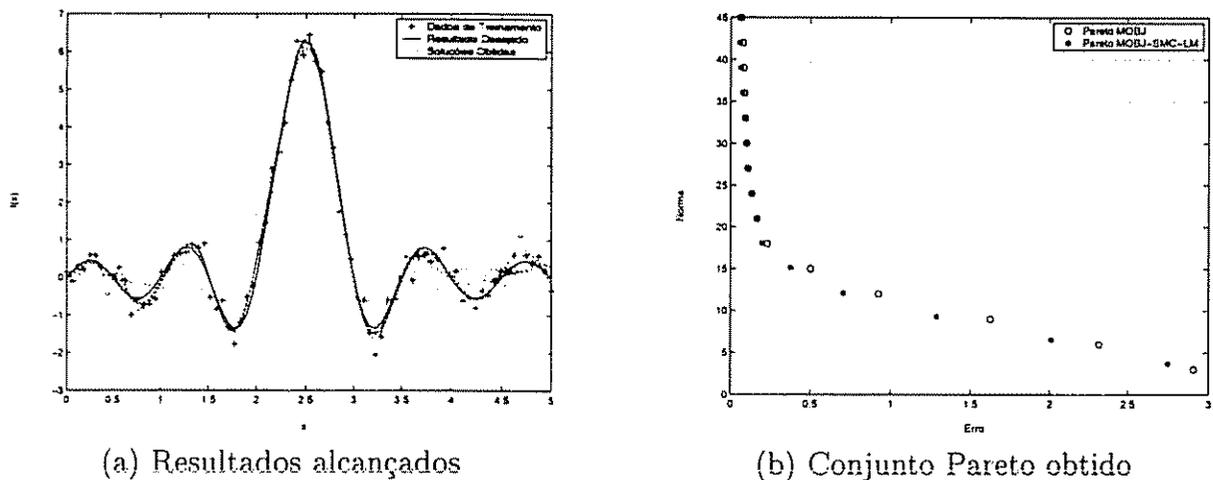


Figura 6.11: Soluções obtidas com o algoritmo MOBJ-SMC-LM

Através da Figura 6.11 é possível concluir que o algoritmo é capaz de alcançar as soluções Pareto com uma melhor resolução do erro, ou seja, alcançando valores reduzidos da função do erro, especialmente para os pontos que apresentam valores de norma mais elevados. Tal comportamento pode ser explicado pela capacidade do método de aproximação da superfície do erro por uma superfície quadrática, o que permite que o ponto de erro mínimo para um determinado valor de norma seja alcançado em um número menor de iterações.

A velocidade de convergência do método MOBJ-SMC-LM pode ser medida em relação aos demais algoritmos multi-objetivo (MOBJ/SMC-MOBJ) especificando-se um ponto comum a ser alcançado a partir de um determinado vetor de pesos. Entretanto, o método MOBJ utilizado neste trabalho para estimar as soluções Pareto ( $\epsilon$ -restrito), não apresenta a característica de convergência para soluções arbitrárias do plano de soluções partindo de qualquer solução definida no plano. Dessa forma, para medir a velocidade do método MOBJ-SMC-LM será utilizado apenas o método SMC-MOBJ, cujas características de convergência são semelhantes.

Para analisar a velocidade de convergência do método em relação ao algoritmo SMC-MOBJ, foi especificado um mesmo valor de norma para as superfícies de deslizamento. A seguir ambos os algoritmos foram utilizados, partindo de um mesmo ponto inicial gerado a partir de pesos aleatórios e convergindo para o erro mínimo condicionado ao valor especificado da norma. A convergência, em número de iterações, do erro e da norma podem ser visualizadas na Figura 6.12.

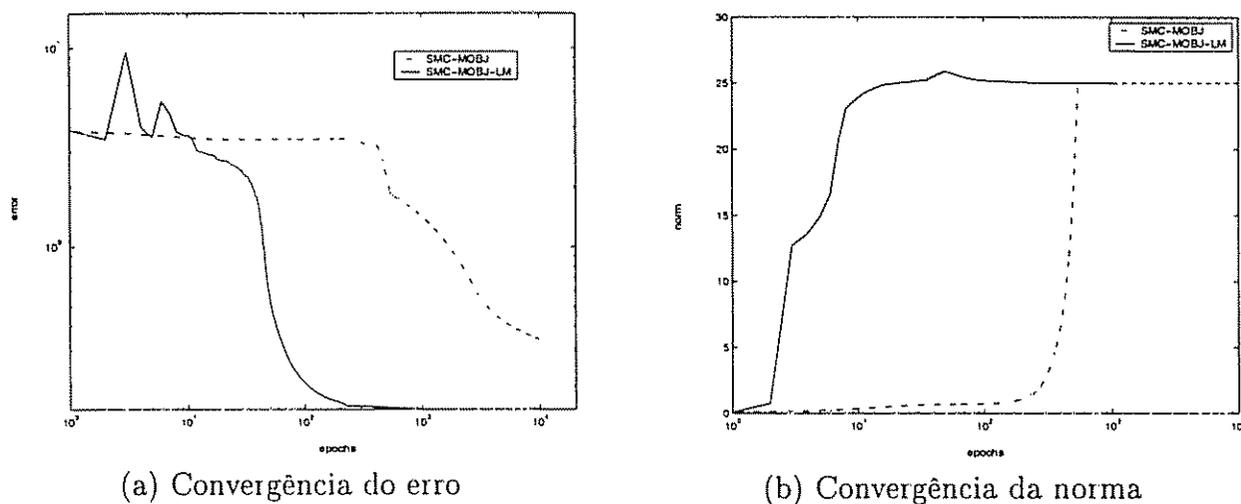


Figura 6.12: Análise da velocidade de convergência do algoritmo MOBJ-SMC-LM

Os resultados demonstram a capacidade do algoritmo MOBJ-SMC-LM de convergir para o mínimo valor do erro, dado um determinado valor da norma, em um número menor de iterações em relação ao algoritmo SMC-MOBJ. A principal vantagem do método em relação ao algoritmo de Levenberg-Marquardt consiste na sua capacidade de obter uma solução final com alta capacidade de generalização. O método de Levenberg-Marquardt é capaz de alcançar soluções de erro mínimo e norma elevada podendo apresentar uma rede com *overfitting*. A Figura 6.13 demonstra esse comportamento no qual a solução do método de Levenberg-Marquardt apresenta uma valor de norma elevado e erro mínimo, sendo que a solução ideal, escolhida a partir da geração do conjunto Pareto, apresenta um valor reduzido da norma dos pesos, com um erro superior.

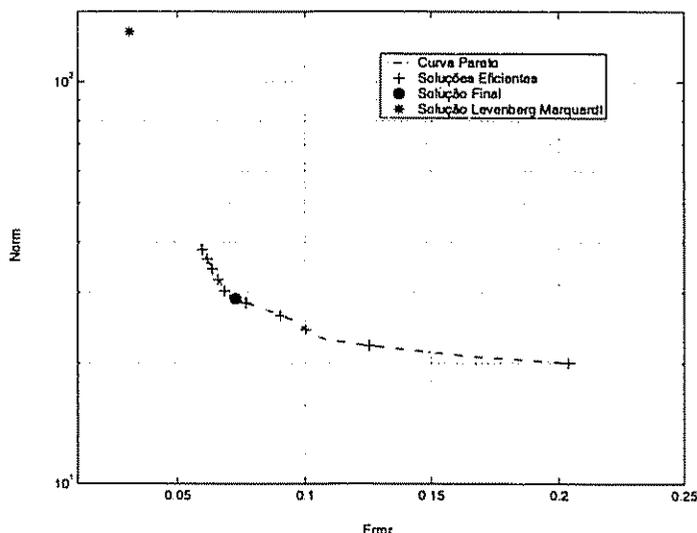
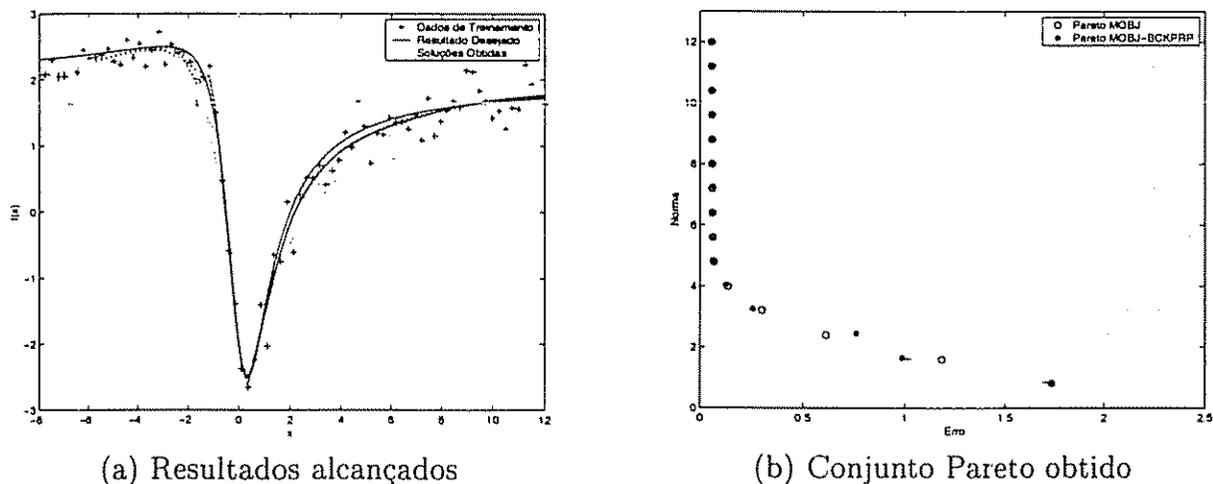


Figura 6.13: O conjunto Pareto e a solução de Levenberg-Marquardt

#### 6.4.4 O Controle da norma através da taxa de aprendizado

Apesar de o algoritmo *backpropagation* não apresentar um comportamento multi-objetivo capaz de restringir o seu treinamento a um determinado valor da norma, modificações na manipulação dos gradientes e nos ganhos da taxa de aprendizado,  $\alpha$ , possibilitam tal operação conforme descrito na Seção 5.4. Como resultado, esse método é capaz de restringir o valor da norma dos pesos a uma determinada região à medida em que o ponto de operação é deslocado em direção ao conjunto Pareto-ótimo, gerando, assim, as soluções ótimas das quais é possível extrair a rede com a melhor capacidade de generalização.



(a) Resultados alcançados

(b) Conjunto Pareto obtido

Figura 6.14: Soluções obtidas com o algoritmo MOBJ-BP (1)

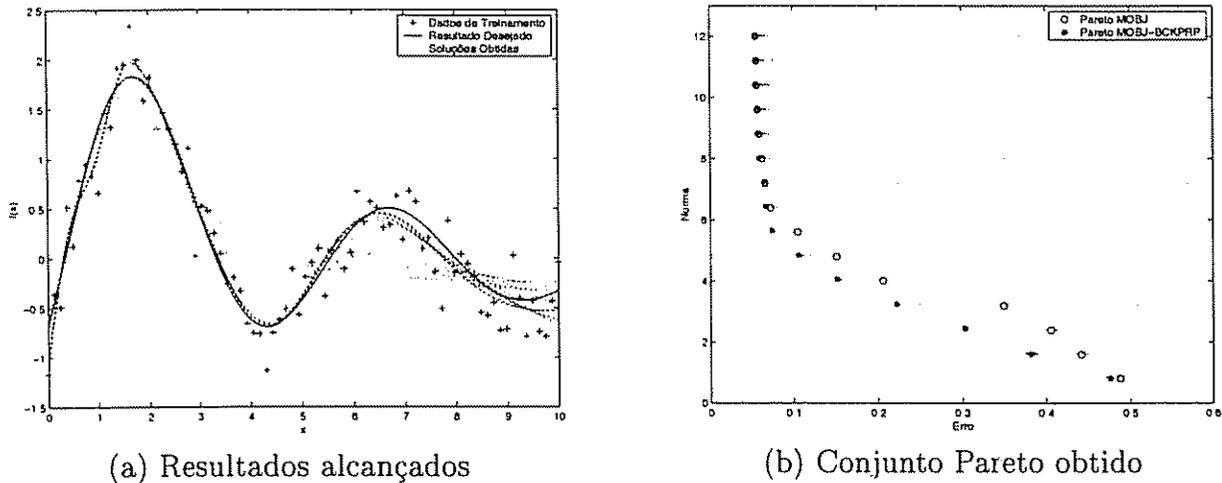


Figura 6.15: Soluções obtidas com o algoritmo MOBJ-BP (2)

As Figuras 6.14 e 6.15 apresentam os resultados obtidos a partir da geração do conjunto Pareto utilizando o algoritmo *backpropagation* com controle de norma (MOBJ-BP). Os resultados demonstram que o algoritmo também é capaz de gerar as soluções multi-objetivo. Entretanto, a velocidade de convergência é comprometida uma vez que o método utiliza o gradiente descendente para a realização dos ajustes dos pesos. A principal vantagem do método consiste na escolha da taxa de aprendizado por parte do algoritmo, uma vez que o método é capaz de determinar os intervalos para a taxa de aprendizagem  $\alpha$ , escolhendo o ganho capaz de reduzir a função do erro da rede.

## 6.5 Predição de Séries Temporais

Nesta seção será demonstrado o uso do algoritmo multi-objetivo para a escolha de MLPs capazes de realizar a predição de séries temporais. Para isso, a série de Preços do Boi Gordo no Estado de São Paulo [Costa et al., 2000] será utilizada. Na Figura 6.16 é possível verificar o comportamento da série. Uma análise detalhada dessa série pode ser obtida junto à literatura [Costa et al., 2000].

A série apresenta 538 amostras que foram subdivididas em 450 amostras para o treinamento e 88 amostras para a validação. Foi utilizada uma rede MLP de duas camadas apresentando 20 entradas, constituídas por valores atrasados da série, 20 nodos na camada escondida e uma saída representando a predição do próximo valor. As funções de ativação da camada de entrada e de saída foram definidas pela tangente hiperbólica e linear, respectivamente.

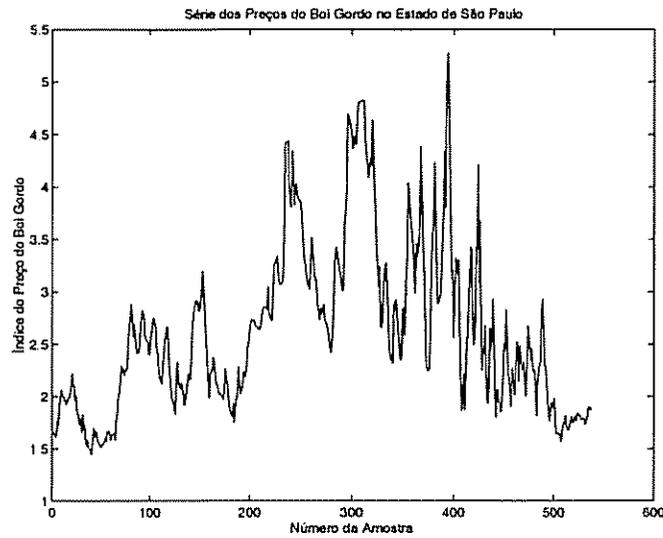
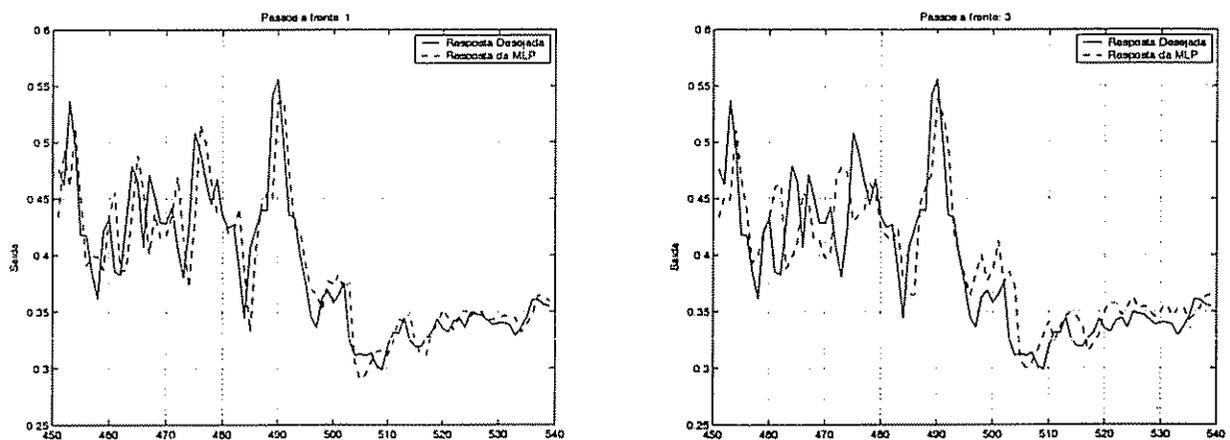


Figura 6.16: Série temporal utilizada

O método multi-objetivo foi utilizado para obter o conjunto Pareto-ótimo para a série de dados. Uma vez obtidas as soluções ótimas, foram escolhidas as soluções que apresentam o melhor desempenho para a predição um passo à frente, três-passos, seis-passos e predição livre. Os resultados encontram-se a seguir:

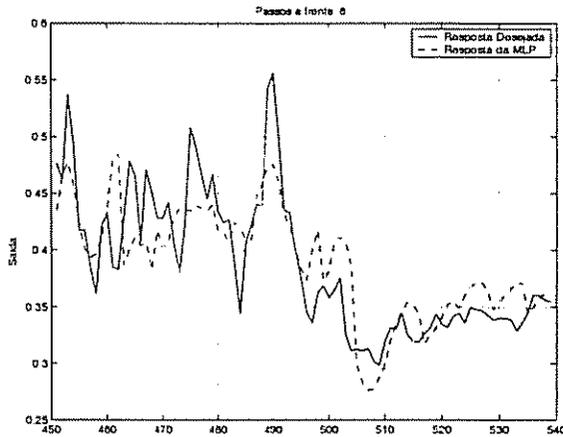


(a) Predição 1-passo à frente

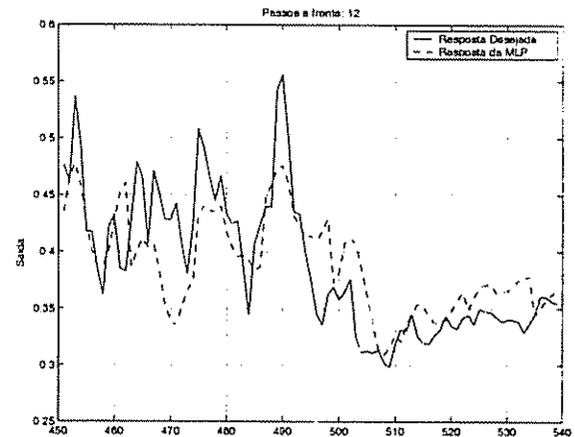
(b) Predição 3-passos à frente

Figura 6.17: Soluções multi-objetivo para a predição da Série do Boi Gordo (1)

Na Figura 6.17 encontra-se os resultados da predição para 1-passo (a) e 3-passos (b) à frente representando uma predição mensal e trimestral do índice de preços.



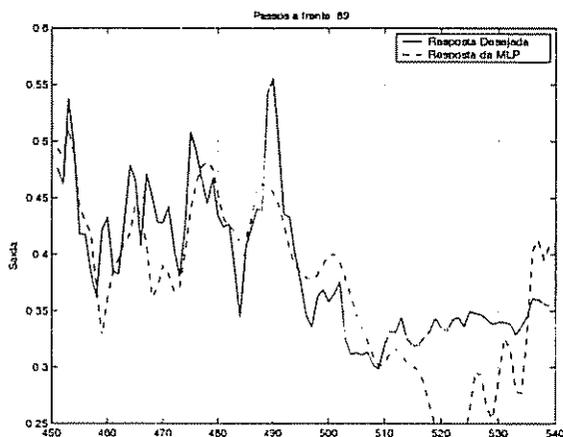
(a) Predição 6-passos à frente



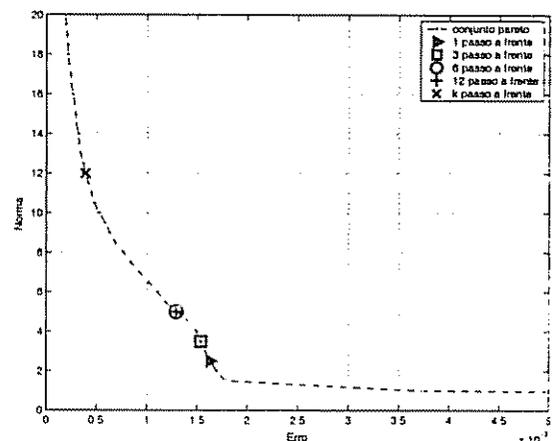
(b) Predição 12-passos à frente

Figura 6.18: Soluções multi-objetivo para predição da Série do Boi Gordo (2)

A Figura 6.18 demonstra a predição semestral (a) e anual (b) do índice. Apesar da perda da capacidade de predição mensal da rede, a resposta é coerente com a tendência dos seus valores, especialmente em relação à predição a partir da amostra 505.



(a) Predição livre (k-passo)



(b) Conjunto Pareto Obtido

Figura 6.19: Soluções multi-objetivo para predição da Série do Boi Gordo (3)

Na Figura 6.19 (a) é possível verificar a predição livre e as soluções obtidas para cada caso mapeadas no conjunto Pareto (b). De acordo com os resultados obtidos, o aumento do horizonte de predições da rede provocou um aumento das normas das soluções encontradas.

Em um treinamento convencional, a rede é treinada utilizando um algoritmo mono-objetivo na tentativa de obter a solução com o menor erro em relação ao conjunto de treinamento. Normalmente, diversas redes são treinadas sendo escolhida aquela que apresenta a melhor resposta em relação a um determinado horizonte de predição. O treinamento segundo o método multi-objetivo permite definir um conjunto de soluções eficientes

sendo então selecionadas as redes que apresentam o melhor desempenho para os diferentes horizontes de predição desejados. O problema de predição de séries temporais pode ser analisado como um problema de regressão de função, para um determinado vetor de entrada deseja-se prever uma determinada saída. Para obter diferentes horizontes de predição é realizada a realimentação da saída da rede. Dessa forma, o conjunto de treinamento é o mesmo independentemente do horizonte de predição desejado indicando que as soluções ou redes podem ser obtidas a partir de um mesmo conjunto Pareto, sendo escolhidas utilizando-se o erro de validação em relação ao respectivo horizonte de predição. Utilizando os métodos multi-objetivo, a procura por modelos neurais de predição é realizada através da geração das soluções do conjunto Pareto-ótimo.

## 6.6 Conclusões do Capítulo

Os resultados obtidos neste capítulo demonstram que é possível estabelecer trajetórias no espaço solução através do algoritmo SMC-MOBJ, possibilitando a obtenção de qualquer solução factível mapeada no plano de soluções.

A análise do treinamento de RNAs no plano de soluções: Erro  $\times$  Norma, demonstra que as soluções com alta capacidade de generalização, resultantes do treinamento de uma RNA, podem ser obtidas através do equilíbrio entre o erro e a norma do vetor de pesos. Exceto em casos onde a rede é sub-dimensionada, os algoritmos multi-objetivos: MOBJ, SMC-MOBJ, MOBJ-SMC-LM e MOBJ-BP (controle da taxa de aprendizado), são capazes de obter a solução na qual o modelo neural apresenta uma resposta próxima da função geradora expressa no conjunto de treinamento.

O método MOBJ-SMC-LM, desenvolvido a partir do algoritmo de Levenberg-Marquardt, ao qual foi adicionado um termo de restrição da norma, permite obter as soluções do conjunto Pareto em um número menor de iterações com uma boa resolução da função do erro. O método é capaz de alcançar valores mínimos para o erro a partir de qualquer ponto do plano de soluções desde que seja especificado o valor da norma desejada.

O algoritmo multi-objetivo com controle da taxa de aprendizado (MOBJ-BP), desenvolvido a partir do método *backpropagation*, permite alcançar as soluções ótimas através da decomposição e manipulação dos gradientes e taxas de aprendizado. O método é capaz de restringir o valor da norma dos pesos a uma determinada região minimizando, simultaneamente, o erro da rede.



# Capítulo 7

## Métodos de Pruning Aplicados a Soluções Multi-Objetivo

### 7.1 Introdução

Vários métodos para a simplificação de topologias MLP são abordados na literatura [Reed, 1993]. O objetivo desses métodos consiste no aprimoramento da capacidade de generalização das soluções obtidas através da simplificação da topologia. O algoritmo MOBJ, SMC-MOBJ e suas variações permitem obter uma solução de equilíbrio entre o erro de validação e o erro de treinamento, promovendo o equilíbrio entre a polarização e a variância da solução. Outra importante característica desses métodos é a possibilidade de obter uma topologia que apresente o menor valor da norma dos pesos para um determinado erro de treinamento. Desde que a topologia original da rede seja adequada aos dados e os dados estatisticamente representativos, o método garante uma boa solução, com boa capacidade de generalização.

Semelhante aos métodos de regularização, o método MOBJ/SMC-MOBJ não altera a topologia original da rede MLP. Mas, uma vez obtida uma solução multi-objetivo, é desejável que também seja alcançada uma simplificação da topologia original sem perda de desempenho.

Neste capítulo serão abordados alguns métodos de pruning de MLPs com ênfase nos métodos multi-objetivo [Costa et al., 2002a]. Para isso, uma análise do comportamento da norma dos pesos e sua influência no desempenho de cada elemento da rede, seja nodo, peso ou função de ativação será realizada.

Sendo concluído o treinamento multi-objetivo de uma MLP, conhecido o conjunto Pareto correspondente e definida a solução que apresente a maior capacidade de generalização, o uso de métodos de pruning tem o propósito de simplificar a dimensão da MLP. Por outro lado, algumas características da curva Pareto podem ser utilizadas na tentativa de encontrar a topologia reduzida durante o treinamento. Algumas técnicas

que serão apresentadas nesse capítulo são descritas na literatura. Neste capítulo, alguns métodos de simplificação de topologias serão novamente descritos, mas sob a perspectiva do conjunto Pareto.

A generalização de uma RNA pode ser definida como a capacidade de resposta coerente da rede à um padrão ou conjunto de padrões que não foram apresentados à rede durante o treinamento. Tal fato justifica a necessidade de definir um conjunto de validação: para testar a capacidade de generalização da rede.

Segundo a literatura [Haykin, 1999], a generalização é influenciada por três fatores:

1. dimensão do conjunto de treinamento e sua representatividade;
2. arquitetura da RNA;
3. complexidade física do problema a ser tratado.

Segundo os resultados apresentados no Capítulo 6, a capacidade de generalização de uma rede MLP está relacionada ao erro quadrático e à norma da matriz dos pesos. Na prática, uma vez que a topologia de uma RNA seja superior a uma topologia mínima (definida empiricamente), a capacidade de generalização da rede pode ser definida através de um erro ( $E_{(k)}$ ) e uma norma ( $\|\mathbf{w}_{(k)}\|$ ), que correspondem a uma solução eficiente ou pertencente ao conjunto Pareto-ótimo. Uma vez garantida a capacidade de generalização para uma determinada topologia de RNA (nesse caso uma MLP) pelo algoritmo multi-objetivo, a ação de *pruning* ou poda pode ser definido como o processo de otimização da dimensão da MLP com o propósito de manter o grau de generalização alcançado no treinamento.

## 7.2 Identificação Multi-Objetivo de Topologias

Uma variação do método conhecido como *método da força bruta* [Engelbrecht, 2001] para a escolha de topologias, no qual inúmeras topologias são treinadas segundo o algoritmo backpropagation ou suas variações e, em seguida, a solução que apresenta a menor dimensão de rede e a melhor performance é escolhida, pode ser utilizada na escolha da dimensão da rede através do uso do algoritmo MOBJ/SMC-MOBJ. A principal vantagem consiste na capacidade do método de obter uma solução de boa generalização para uma dada topologia em um único treinamento. Uma análise do comportamento do conjunto Pareto-ótimo permite concluir que o seu formato bem como o seu comportamento é bem característico a partir de uma dada topologia mínima, ou seja, para topologias com dimensões superiores, o conjunto Pareto-ótimo praticamente não é afetado. Uma vez conhecida essa característica, é possível definir um conjunto relativamente pequeno de topologias (aproximadamente 3 ou 5 topologias) e treiná-las utilizando um método multi-objetivo,

selecionando a menor topologia capaz de manter a melhor característica Erro  $\times$  Norma do conjunto Pareto-ótimo em relação às topologias mais complexas.

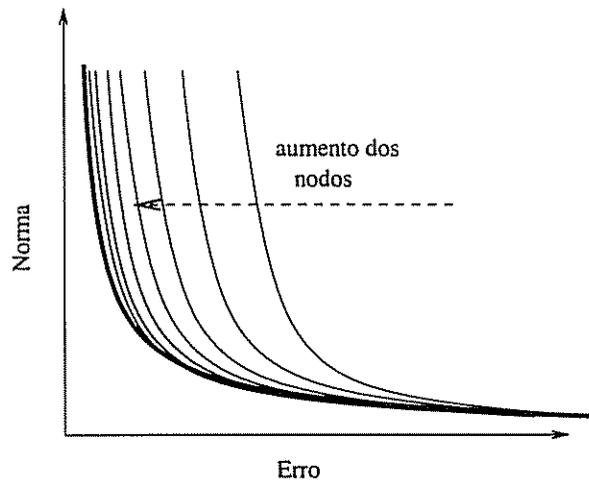


Figura 7.1: Comportamento do conjunto Pareto-ótimo com o aumento dos nodos da camada escondida

A Figura 7.1 demonstra o comportamento do conjunto Pareto em função do aumento dos nodos na camada escondida de uma MLP de duas camadas. Redes sub-dimensionadas apresentam uma curva Pareto deslocada à direita em relação ao eixo da ordenadas. Com o aumento progressivo da quantidade de nodos na camada escondida, a curva Pareto converge para um determinado limite. Redes superdimensionadas apresentam curvas Pareto semelhantes.

Como exemplo, cinco topologias foram escolhidas para teste nos dados referentes à aproximação da função *sinc*. As MLPs apresentam a seguinte quantidade de nodos na camada escondida: 50, 25, 15, 10 e 5. Cada topologia foi treinada utilizando o algoritmo MOBJ padrão. Os resultados referentes ao erro de treinamento, erro de validação e norma da matriz dos pesos encontram-se na tabela 7.1

Tabela 7.1: Desempenho das soluções MOBJ em função da variação dos nodos da camada escondida

| Nodos Escondidos    | 50     | 25     | 15     | 10     | 5      |
|---------------------|--------|--------|--------|--------|--------|
| Erro de Treinamento | 0.0610 | 0.0716 | 0.0715 | 0.0743 | 0.1646 |
| Erro de Validação   | 0.0226 | 0.0159 | 0.0154 | 0.0131 | 0.0871 |
| Norma dos Pesos     | 38.23  | 31.25  | 32.17  | 37.45  | 28.01  |

As curvas Pareto obtidas para cada caso podem ser visualizadas na Figura 7.2. A rede com 5 nodos na camada escondida está deslocada em relação às demais curvas, representando um comportamento de um modelo sub-dimensionado. As demais topologias apresentam curvas Pareto muito próximas mas, com variações nas regiões inferiores (baixa

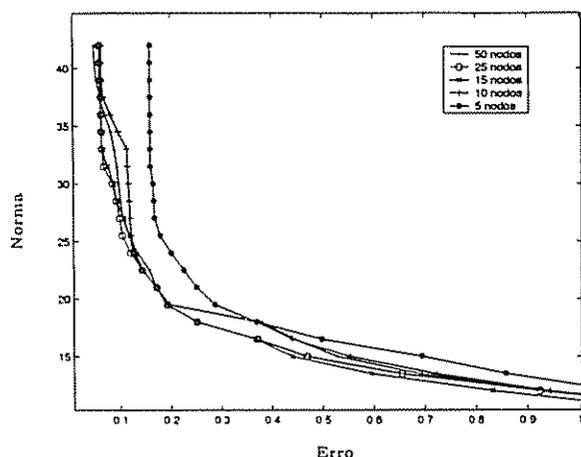


Figura 7.2: Curvas Pareto obtidas

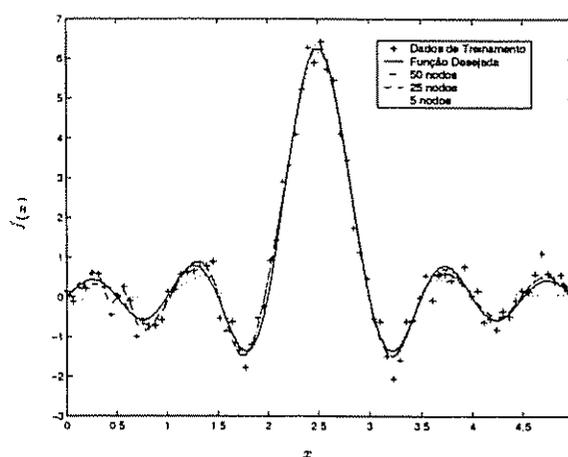


Figura 7.3: Soluções obtidas

Tabela 7.2: Identificação de topologias de RNAs utilizando um treinamento multi-objetivo

| Base de Dados | $E_t$  | $E_v$  | Topologia Final | $\ w\ $ |
|---------------|--------|--------|-----------------|---------|
| cancer        | 99.52% | 98.93% | 9-1-2           | 3.029   |
| card          | 88.89% | 89.13% | 51-2-2          | 2.460   |
| gene          | 95.07% | 91.73% | 120-4-3         | 5.613   |
| $d(x)$        | 0.0626 | 0.0113 | 1-8-1           | 7.008   |
| sinc          | 0.0738 | 0.0199 | 1-11-1          | 39.989  |
| building      | 0.0159 | 0.0106 | 14-3-3          | 4.990   |

norma) e em uma estreita região compreendida entre os valores 25 e 35 para a norma dos pesos. A melhor solução apresenta a melhor capacidade de generalização, ou seja, o menor erro em relação ao conjunto de validação. No exemplo proposto, a topologia ideal para o problema apresenta 10 nodos na camada escondida.

Uma maneira de minimizar as variações das curvas Pareto obtidas, consiste em realizar múltiplos treinamentos para uma mesma topologia, calculando a curva média. Entretanto, tal procedimento tem como consequência o aumento do tempo total de treinamento e do custo computacional.

A Tabela 7.2 contém os resultados obtidos a partir do uso do método proposto em diferentes bases de dados. Os resultados referentes à classificação de padrões são apresentados sob a forma de percentagem de acerto para o conjunto de treinamento e de validação. Os demais problemas referem-se ao uso dos modelos neurais para a regressão de funções sendo demonstrados os respectivos erros de treinamento ( $E_t$ ) e validação ( $E_v$ ).

Novamente, analisando a Figura 7.2, é possível perceber que à medida que a topologia da rede se torna sub-dimensionada, ocorre o chamado *underfitting*. Esse fenômeno provoca o deslocamento da curva Pareto-ótimo, afastando-a progressivamente do eixo das ordenadas. O método, descrito anteriormente, obtém todo o conjunto Pareto para uma

posterior análise e identificação da estrutura ideal da rede. Uma vez observado o efeito do deslocamento da curva Pareto-ótimo, é possível definir uma estratégia para encontrar a topologia ideal utilizando apenas uma referência para o valor da norma. A Figura 7.4 exemplifica o método: Definido um valor de norma,  $\|w^*\|$ , o algoritmo SMC-MOBJ é utilizado para minimizar o erro restringindo a norma dos pesos a um único valor. De acordo com o comportamento do conjunto Pareto, espera-se uma redução do valor do erro alcançado com o aumento da topologia ou dos nodos na camada escondida. A partir de determinado valor, o erro não sofrerá grandes variações convergindo para um limite. A menor topologia que apresentar um valor do erro igual ou muito próximo ao limite será então escolhida. A vantagem do método em relação ao método anterior, no qual todo o conjunto Pareto-ótimo é gerado para cada topologia analisada, consiste na restrição do treinamento a apenas um ponto do conjunto Pareto, reduzindo o custo computacional. Entretanto, uma vez definida a melhor topologia, é necessário um treinamento para a obtenção de todo o conjunto Pareto a fim de escolher o ponto de melhor equilíbrio entre o erro e a norma (máxima generalização). Inicialmente, é preciso estimar o valor da referência da norma,  $\|w^*\|$ , evitando valores localizados próximos à região inferior do Pareto, no qual a variação do erro é maior (ver Figuras 7.2 e 7.4).

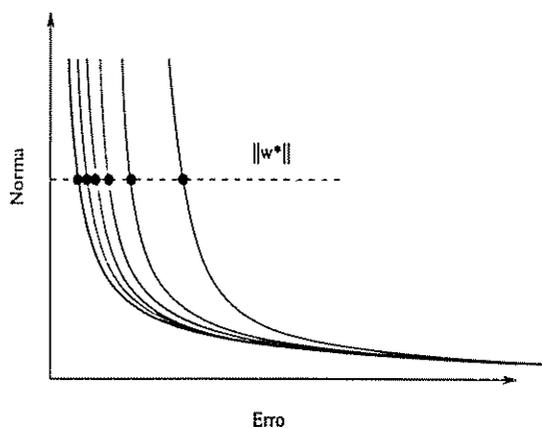


Figura 7.4: Comportamento da Curva Pareto em função da Topologia

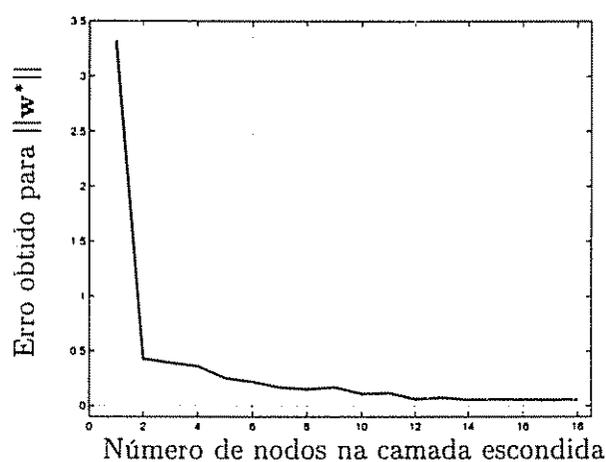


Figura 7.5: Soluções obtidas

A Figura 7.5 exemplifica o método em questão. Os dados utilizados referem-se à função *sinc* e o valor da norma de referência utilizada,  $\|w^*\| = 50$ . A curva demonstra que a partir de 12 nodos na camada escondida, a rede é capaz de apresentar um baixo valor do erro, com pouca variabilidade, indicando que o conjunto Pareto-ótimo, obtido a partir dessa topologia, apresenta características semelhantes. Tal variabilidade, que também é verificada no treinamento de topologias idênticas, pode ser explicada pela resolução ou passos de treinamento insuficientes para o alcance do ponto desejado no Pareto. Tal efeito produz uma curva Pareto muito próxima à de seus valores reais, mas com um pequeno desvio.

### 7.2.1 Descrição Matemática do Método

O método construtivo de RNAs denominado *Identificação Multi-Objetivo de Topologias* realiza a busca de soluções em um espaço  $\beta$ -dimensional definido pelos seguintes funcionais:

1.  $E_{(\mathbf{w})}$  : Soma dos erros quadráticos da RNA (MLP)
2.  $\|\mathbf{w}\|$  : Norma do vetor de pesos da RNA
3.  $H$  : Número de nodos na camada escondida da RNA

O objetivo do método consiste em obter um vetor de pesos  $N_{(H)}$ -dimensional, onde  $N$  é a dimensão do vetor de pesos escrito em função do número de nodos da camada escondida da rede, de modo que a solução ou rede desejada apresente uma topologia reduzida com boa capacidade de generalização. Sendo a capacidade de generalização de uma rede MLP relacionada à minimização dos funcionais  $E_{(\mathbf{w})}$  e  $\|\mathbf{w}\|$ , o problema de otimização desses funcionais pode ser escrito conforme a Equação 7.1.

$$\mathbf{w}^* = \underset{\mathbf{w} \in \mathbb{R}^{N(H)}}{\text{arg min}} \begin{cases} E_{(\mathbf{w})} \\ \|\mathbf{w}\| \\ H \end{cases} \quad (7.1)$$

onde os funcionais  $H$  e  $N_{(H)}$  são definidos em  $\mathcal{Z}^+$ , que representa o conjunto dos números inteiros positivos, ou seja, um conjunto discreto de soluções. Os funcionais  $E_{(\mathbf{w})}$  e  $\|\mathbf{w}\|$  são definidos em  $\mathbb{R}^+$  (conjunto de soluções reais positivas).

O conjunto de soluções *eficientes* ou Pareto-ótimo, nesse caso, é definido como:

$$\mathcal{W}^* \triangleq \{ \mathbf{w}^* \in \mathbb{R}^{N(H)} \mid \nexists \mathbf{w} \text{ tal que } \mathbf{f}(\mathbf{w}) \leq \mathbf{f}(\mathbf{w}^*) \text{ e } \mathbf{f}(\mathbf{w}) \neq \mathbf{f}(\mathbf{w}^*) , \text{ para } H = 1, 2, 3, \dots, n \} \quad (7.2)$$

onde  $\mathbf{f}(\mathbf{w}) = [E_{(\mathbf{w})}, \|\mathbf{w}\|]^T$  e  $n$  é o número de nodos que representa uma rede *superdimensionada*, definido empiricamente.

O conjunto Pareto é então contínuo em relação aos funcionais  $E_{(\mathbf{w})}$  e  $\|\mathbf{w}\|$  e discreto em relação ao funcional  $H$ . A Figura 7.6 exemplifica o comportamento desse conjunto. Os resultados foram obtidos a partir de um conjunto de treinamento, utilizando a função *sinc*, ao qual foi adicionado um ruído gaussiano.

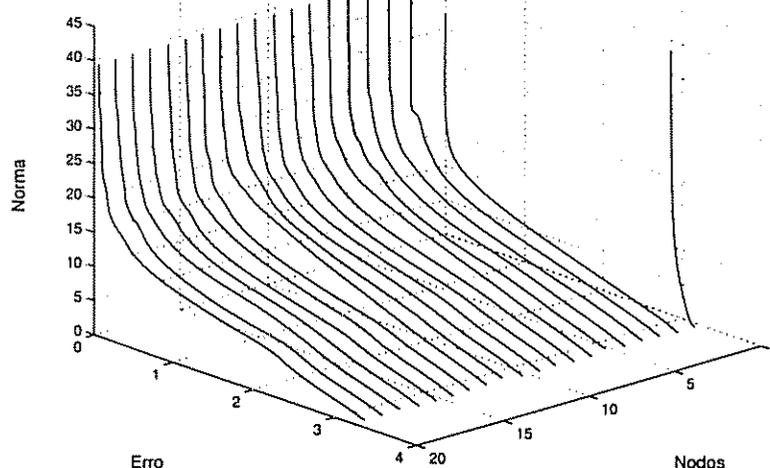


Figura 7.6: Conjunto tridimensional de soluções *eficientes*

A definição do valor de  $n$ , que representa o número de nodos na camada escondida de uma rede MLP *superdimensionada*, conforme apresentado anteriormente, é empírico. Segundo a análise realizada em relação ao comportamento do conjunto Pareto para a rede MLP, o valor de  $n$  pode ser estimado a partir da convergência da curva Pareto a um limiar no plano Erro  $\times$  Norma com o aumento do funcional  $H$  (ver Figura 7.1). A Figura 7.7 demonstra o comportamento do conjunto Pareto no plano  $\mathbb{R}^2$  em função do aumento dos nodos na camada escondida da MLP para o exemplo utilizando a função *sinc*.

O critério de decisão utilizado para selecionar a solução final é o mesmo descrito para o método MOBJ [Teixeira, 2001], ou seja, o erro de validação. Dessa forma, a rede que apresentar a menor topologia com boa capacidade de generalização (menor erro de validação) é escolhida como solução final.

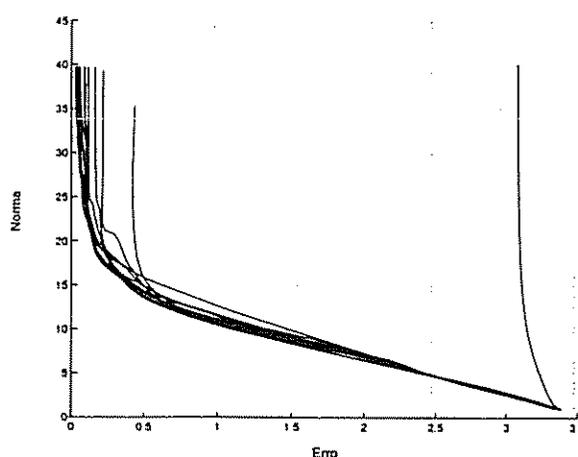


Figura 7.7: Convergência do conjunto Pareto com o aumento dos nodos *escondidos*

Diversas heurísticas podem ser definidas para analisar subconjuntos de soluções eficientes definidas em  $\mathcal{W}^*$  (Equação 7.1), como o método que analisa a convergência do erro em relação a um único valor da norma dos pesos.

### 7.3 Comportamento Linear/não-Linear do nodo Perceptron

O método *Comportamento Linear/não-Linear do nodo Perceptron* proposto [Costa et al., 2002a] procura indentificar nodos pertencentes à camada escondida de uma rede MLP superdimensionada já treinada utilizando um algoritmo multi-objetivo, substituindo esses nodos por conexões lineares e, dessa forma, simplificando a topologia original sem perda da capacidade de resposta ou generalização da rede.

Seja um nodo  $i$  localizado na camada escondida de uma rede MLP de duas camadas. A saída do nodo é representada pela seguinte função:

$$y_i = f_i \left\{ \left[ \sum_{j=1}^N (x_j \cdot w_{1ji}) \right] + b_{1i} \right\} \quad (7.3)$$

As funções de ativação para um nodo na camada escondida são, normalmente, na forma sigmoidal. Caso a função de ativação seja linear, conforme será visto posteriormente, o nodo pode ser substituído por pesos lineares interligados diretamente entre a entrada e a saída da rede. Na Figura 7.8 é possível visualizar o comportamento de uma função sigmoidal (tangente hiperbólica) e uma função linear.

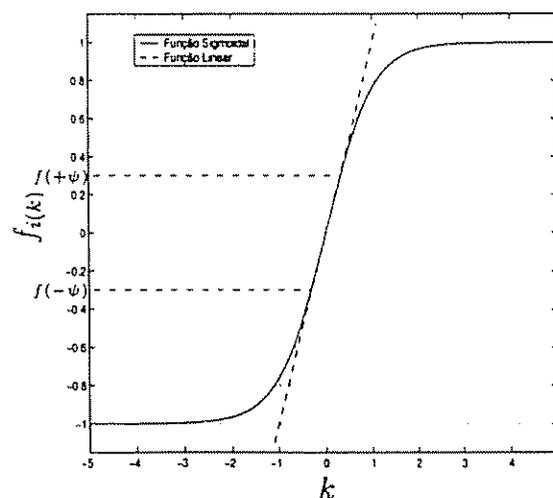


Figura 7.8: Funções de ativação: tangente hiperbólica e linear

onde:  $k = \left[ \sum_{j=1}^N (x_j \cdot w_{1ji}) \right] + b_{1i}$

Dada a Figura 7.8, é possível observar que o comportamento da função tangente hiperbólica próxima da origem é semelhante ao comportamento da função linear. Dessa forma, é possível decompor a função sigmoideal em uma região que apresenta um comportamento linear e duas outras regiões com comportamento não-linear. Para isso, o parâmetro  $\psi$  foi definido de forma que, no intervalo  $[-\psi < k < \psi]$  o nodo apresenta um comportamento linear. Entretanto, a princípio não há como ser estimado o valor atribuído ao parâmetro  $\psi$ .

Uma vez que um nodo não-linear da camada escondida apresente um comportamento linear, é possível substituí-lo por conexões lineares interligando diretamente a entrada da rede à camada de saída. Tal fato pode ser justificado utilizando propriedades algébricas.

Uma vez conhecida a Equação 7.3 que representa a saída de um nodo na camada escondida, a equação para um nodo linear é da forma:

$$y_i^t = \left[ \sum_{j=1}^N (x_j \cdot w_{1ji}) \right] + b_{1i} \quad (7.4)$$

Expandindo a equação de saída de uma rede MLP (Equação 3.24) de forma a evidenciar o nodo linear  $t$ , obtém-se a seguinte expressão:

$$y_p = f_p \left\{ \sum_{i=1, i \neq t}^H \left( w_{2ip} \cdot f_i \left[ \sum_{j=1}^N (x_j \cdot w_{1ji}) + b_{1i} \right] \right) + w_{2tp} \cdot \left[ \sum_{j=1}^N (x_j \cdot w_{1jt}) + b_{1t} \right] + b_{2p} \right\} \quad (7.5)$$

o termo  $w_{2tp}$  pode ser inserido no somatório do nodo linear resultando em:

$$y_p = f_p \left\{ \sum_{i=1, i \neq t}^H \left( w_{2ip} \cdot f_i \left[ \sum_{j=1}^N (x_j \cdot w_{1ji}) + b_{1i} \right] \right) + \sum_{j=1}^N (x_j \cdot w_{1jt} \cdot w_{2tp}) + (b_{1t} \cdot w_{2tp} + b_{2p}) \right\} \quad (7.6)$$

O ganho  $w_{1jt} \cdot w_{2tp}$  pode ser substituído por um parâmetro  $w_{ln_{jp}}$  e o bias do nodo da saída pela nova soma:  $b_{2p}' = b_{1t} \cdot w_{2tp} + b_{2p}$ , resultando na expressão:

$$y_p = f_p \left\{ \sum_{i=1, i \neq t}^H \left( w_{2ip} \cdot f_i \left[ \sum_{j=1}^N (x_j \cdot w_{1ji}) + b_{1i} \right] \right) + \sum_{j=1}^N (x_j \cdot w_{ln_{jp}}) + b_{2p}' \right\} \quad (7.7)$$

A análise acima foi realizada supondo que um único nodo apresenta um comportamento linear. Caso outros nodos apresentem o mesmo comportamento, o mesmo processo pode ser aplicado somando-se aos pesos lineares,  $w_{ln}$ , o produto da combinação entre os pesos da camada escondida e de saída do nodo em questão. Semelhantemente, o termo de bias da saída receberá um acréscimo resultante da combinação entre o termo de bias

do nodo e o peso de sua saída. A Figura 7.9 mostra o perfil, em um exemplo, da nova topologia da rede MLP. As Equações 7.8 e 7.9 indicam a atualização das conexões lineares e o ajuste do termo de bias da saída.

$$wl_{jp} = \sum_{t=1}^L w1_{jt} \cdot w2_{tp} \quad (7.8)$$

$$b2_p = b2_p + \sum_{t=1}^L b1_t \cdot w2_{tp} \quad (7.9)$$

onde  $L$  é o número de nodos que podem ser substituídos por conexões lineares (apresentam um comportamento linear).

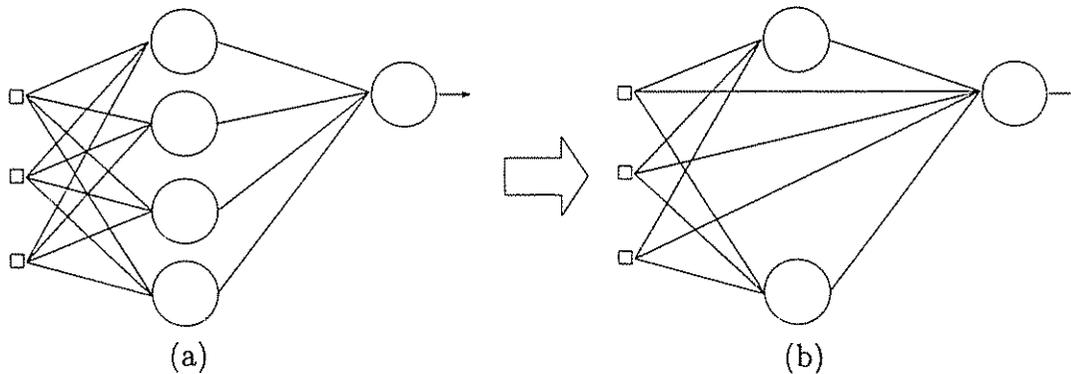


Figura 7.9: Rede MLP antes (a) e após (b) simplificação

Retornando à Equação 7.3, referente ao nodo da camada escondida e considerando um sinal de entrada,  $x_j$ , que apresenta uma distribuição normal de média zero e desvio padrão unitário, o comportamento do nodo está associado à dimensão dos pesos na sua entrada,  $w1_{ji}$  e  $b1_i$ . A função Norma (Equação 4.2) permite agrupar os valores dos pesos representando-os em uma forma unidimensional. Nesse caso, a raiz quadrada da soma dos valores absolutos ou quadráticos dos pesos. Se a função norma for pequena, significa que a amplitude dos pesos do nodo ou da rede também encontra-se reduzida, o que não indica que a distribuição das normas individuais seja homogênea. Mas, quanto menor a norma dos pesos de um nodo, menor será a amplitude do sinal de entrada na função de ativação. Conseqüentemente, a redução do sinal de entrada do nodo,  $k$ , pode impor o comportamento tipicamente linear na sua função de ativação,  $f_{(k)}$ .

Considerando-se uma topologia MLP de duas camadas com função de ativação tangente hiperbólica nos nodos da camada escondida, o mesmo princípio observado anteriormente ao nodo pode ser aplicado. Se a norma da rede for suficientemente baixa, é possível que todos os nodos apresentem um comportamento linear. Nesse caso, o processo de simplificação da rede irá resultar em uma rede MLP de uma única camada. Por outro lado, também é correto afirmar que, se a norma da rede for suficientemente alta, é provável que

todos os nodos apresentem um comportamento não-linear. De posse dessa informação e sendo o conjunto Pareto-ótimo formado por soluções com variação progressiva de norma, é correto afirmar que existe no conjunto Pareto-ótimo um conjunto de soluções representadas por soluções com um comportamento tipicamente linear (norma baixa) até soluções com uma forte tendência não-linear (norma elevada). A solução ótima, ou que apresenta a melhor capacidade ou coeficiente de generalização, também apresenta o melhor equilíbrio linear/não-linear de seus nodos, sendo o método MOBJ/SMC-MOBJ capaz de determinar soluções ou uma solução muito próxima a essa, uma vez definida a topologia da rede.

Uma vez que a rede MLP obtida através dos algoritmos MOBJ/SMC-MOBJ apresenta a menor norma possível e garante a generalização, pode-se analisar o seu desempenho na tentativa de identificar e simplificar nodos com comportamento linear. Como resultado de tal operação, espera-se determinar uma rede constituída por conexões lineares entre a entrada e a saída mais um conjunto de nodos não-lineares na camada escondida (supondo uma MLP com uma única camada escondida) promovendo, assim, a simplificação da topologia sem perda significativa da sua capacidade de generalização. Tal operação consiste na determinação do parâmetro  $\psi$  e, conseqüentemente, do intervalo no qual o nodo não-linear apresenta um comportamento linear. Uma vez que a topologia resultante pode apresentar uma pequena variação na sua capacidade de generalização, que é medida pela média do erro quadrático para o conjunto de validação, pode-se determinar, empiricamente, o valor do parâmetro  $\psi$ , definindo-se o desvio aceitável para o erro de validação, aqui representado por  $\sigma_v$ . O algoritmo utilizado nesse processo é descrito a seguir.

**Algoritmo 2** Simplificação de Topologia através da Identificação de Nodos Lineares

---

```

rede ← Treina_rede_MLP_com_algoritmo_MOBJ/SMC-MOBJ();
ev ← erro_de_validação_da_rede_treinada(rede);
σv ← desvio aceitável ;
[max_nodoi; min_nodoi] ← valores máximos e mínimos da resposta dos nodos i's;
{Hidden-treinamento}
Para ψ = 0.01 : 0.01 : 0.99, faça
  aux_rede = rede;
  Para j = 1:NODOS_ESCONDIDOS, faça
    Se (max_nodoj < ψ) e (min_nodoj > -ψ) então
      nodoj = linear;
      simplificar_nodo(aux_rede);
      aux_ev ← calcular_saida_da_rede_para_validacao();
      Se aux_ev ≤ (ev + σv) então
        ev = aux_ev;
        salva_rede(aux_rede);
        aux_ψ = ψ; {obtem o valor de ψ de melhor resultado}
      fim Se
    fim Se
  fim Para
fim Para

```

---

O algoritmo desenvolvido é capaz de determinar o valor de  $\psi$  que apresenta o resultado desejado. Entretanto, em alguns casos, pode ocorrer a inexistência de nodos com comportamento linear. Nesses casos, o algoritmo não efetua nenhum ajuste na topologia original. A seguir são demonstrados alguns exemplos nos quais o método de simplificação foi aplicado.

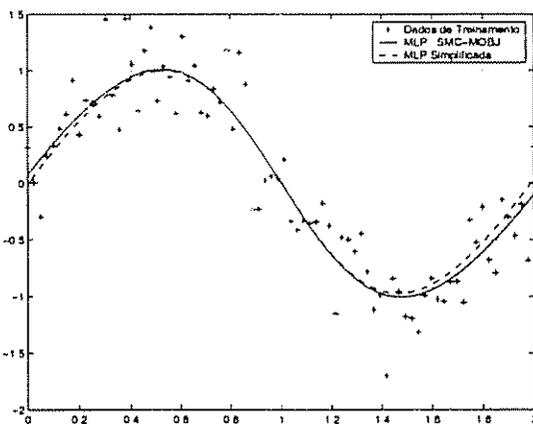


Figura 7.10: Resposta da MLP antes e após simplificação

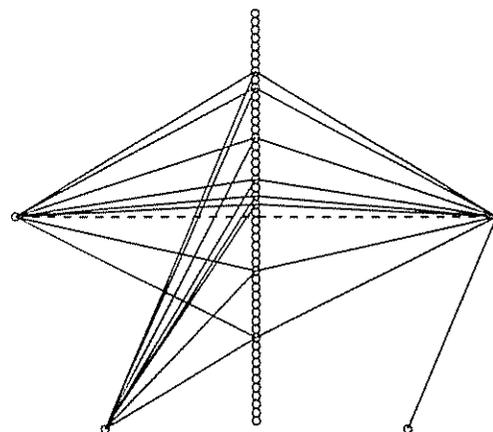


Figura 7.11: Topologia simplificada

As Figuras 7.10 e 7.11 foram geradas a partir de uma MLP 1-50-1 treinada utilizando o algoritmo MOBJ-SMC. A Figura 7.11 mostra a topologia resultante, os nodos eliminados (vazios) e a conexão linear interligando a entrada à saída. A tabela 7.3 mostra as

Tabela 7.3: Performance do Método de Simplificação - Regressão Função Seno

| $\sigma_v$ | $E_t$  | $E_v$  | Topologia  |        | $\psi$ |
|------------|--------|--------|------------|--------|--------|
|            |        |        | não linear | linear |        |
| 0          | 0.0835 | 0.0010 | 1-15-1     | 1-1    | 0.411  |
| 1e-4       | 0.0838 | 0.0010 | 1-14-1     | 1-1    | 0.421  |
| 1e-3       | 0.0860 | 0.0018 | 1-8-1      | 1-1    | 0.441  |
| 1e-2       | 0.0937 | 0.0065 | 1-1-1      | 1-1    | 0.7910 |

topologias obtidas para diversos valores de  $\sigma_v$ . A topologia original apresenta um erro de treinamento,  $e_t = 0.0829$ , erro de validação,  $e_v = 0.0014$  e topologia original 1-50-1.

Novas bases de dados foram utilizadas para exemplificar o desempenho desse método na simplificação de topologias MLPs. Os resultados encontram-se dispostos nas tabelas 7.4 e 7.5.

Tabela 7.4: Características iniciais após treinamento MOBJ

| Base de Dados | Topologia Inicial | $E_t$  | $E_v$  | $\ w\ $ |
|---------------|-------------------|--------|--------|---------|
| cancer        | 9-15-2            | 98.09% | 98.57% | 14.263  |
| card          | 51-15-2           | 88.89% | 88.04% | 4.413   |
| gene          | 120-15-3          | 98.85% | 89.53% | 8.831   |
| $d(x)$        | 1-50-1            | 0.0635 | 0.0085 | 5.064   |
| sinc          | 1-50-1            | 0.0729 | 0.0128 | 32.383  |
| building      | 14-15-3           | 0.0196 | 0.0228 | 6.992   |

A Tabela 7.4 apresenta os resultados obtidos a partir de topologias superdimensionadas de MLPs. O método multi-objetivo foi utilizado para obter o ponto de máxima generalização a partir da geração do conjunto Pareto-ótimo. Os resultados de classificação estão expressos em função da percentagem de acertos e os problemas de regressão, em função do somatório dos erros quadráticos. Os conjuntos de dados utilizados encontram-se subdivididos em treinamento ( $t$ ) e validação ( $v$ ). Os resultados foram inicialmente obtidos a fim de indicar uma referência em relação à capacidade de simplificação de topologias dos algoritmos propostos nesta seção.

Tabela 7.5: Resultados provenientes do processo de simplificação proposto

| Base de Dados | $\sigma_v$ | $E_t$  | $E_v$  | Topologia Inicial | Topologia Final |        | $\psi$ |
|---------------|------------|--------|--------|-------------------|-----------------|--------|--------|
|               |            |        |        |                   | não linear      | linear |        |
| cancer        | 0          | 96.42% | 98.57% | 9-15-2            | 9-2-2           | 9-2    | 0.999  |
| card          | 0          | 88.89% | 88.04% | 51-15-2           | -               | 51-2   | 0.999  |
| gene          | 0          | 98.85% | 89.53% | 120-15-3          | 120-15-3        | -      | 0      |
| $d(x)$        | 0          | 0.0635 | 0.0085 | 1-50-1            | 1-46-1          | 1-1    | 0.031  |
|               | $1e-5$     | 0.0635 | 0.0085 | 1-50-1            | 1-28-1          | 1-1    | 0.701  |
|               | $1e-4$     | 0.0635 | 0.0086 | 1-50-1            | 1-27-1          | 1-1    | 0.801  |
|               | $1e-3$     | 0.0634 | 0.0088 | 1-50-1            | 1-26-1          | 1-1    | 0.841  |
| sinc          | 0          | 0.0754 | 0.0110 | 1-50-1            | 1-48-1          | 1-1    | 0.6110 |
|               | $1e-2$     | 0.0909 | 0.0214 | 1-50-1            | 1-45-1          | 1-1    | 0.6310 |
| building      | 0          | 0.0234 | 0.0224 | 14-15-3           | 14-7-3          | 14-3   | 0.8110 |
|               | $1e-2$     | 0.0505 | 0.0253 | 14-15-3           | 14-4-3          | 14-3   | 0.9110 |

A Tabela 7.5 contém os principais resultados obtidos a partir da simplificação das redes geradas através do treinamento MOBJ/SMC-MOBJ. As bases de dados utilizadas abrangem problemas de classificação e regressão sendo quatro bases: cancer, card, gene e building provenientes de situações reais [Blake and Merz, 1998].

Para classificação, o processo de simplificação foi capaz de diminuir a quantidade de conexões não-lineares para o conjunto *cancer* sem alterar a capacidade de classificação e generalização da rede. No *card*, o processo resultou em uma rede puramente linear, ou seja, com uma única camada de pesos, sem perda de desempenho. Finalmente, no conjunto *gene*, nenhum nodo pôde ser simplificado.

Em relação aos conjuntos de regressão, dois deles, o  $d(x)$  e o  $sinc(x)$ , foram gerados a partir de equações ou funções conhecidas e o conjunto *building*, foi obtido a partir de uma base de dados reais. Nesses casos, verifica-se a variação da topologia em função do desvio,  $\sigma_v$ , imposto ao problema. Indicando que o processo de simplificação está fortemente relacionado à escolha adequada desse parâmetro, como no exemplo da simplificação do conjunto de dados provenientes da função  $d(x)$  no qual uma pequena variação desse parâmetro provoca um aumento significativo dos nodos simplificados. Como contra-exemplo, os conjuntos *sinc* e *building* apresentam uma baixa sensibilidade a esse parâmetro limitando o conjunto solução e restringindo-o à soluções onde o desvio é nulo. A sensibilidade ao parâmetro  $\sigma_v$  pode ser analisada em relação aos limites do intervalo de linearização indicado pelo parâmetro  $\psi$ . Se a rede apresenta um valor de  $\psi$  bem próximo da unidade quando o valor de  $\sigma_v$  é nulo, o(s) nodo(s) linearizado(s) atuam numa região quase não-linear indicando que os demais nodos já apresentam essa característica. Por outro lado, se  $\psi$  apresenta um valor muito pequeno, a probabilidade de encontrar outros nodos que possam ser linearizados, na região intermediária ( $0 < \psi < 1$ ), é maior.

## 7.4 Identificação de Nodos com Semelhança de Respostas

Os vários métodos apresentados neste trabalho têm como objetivo a simplificação de topologias inicialmente treinadas com um algoritmo multi-objetivo (MOBJ/SMC-MOBY). Alguns desses métodos, como o método de linearização, procuram características específicas na MLP que possam ser utilizadas no processo de simplificação. O método apresentado nesta seção parte do princípio de que redes com muitos nodos na camada escondida possam apresentar, em alguns desses nodos, respostas semelhantes. A cada par de nodos que apresenta essa característica, um único nodo pode ser associado, simplificando a topologia da rede. Por outro lado, esse método só permite a simplificação dos nodos *escondidos*. Métodos de simplificação que utilizam informações referentes à correlação entre as saídas de nodos são descritos na literatura [Sietsma and Dow, 1991].

O Método apresentado utiliza informações referentes à diferença entre as saídas de pares de nodos em relação ao conjunto de treinamento para identificar nodos com semelhança de respostas e, uma vez identificado o par de nodos que apresenta a maior semelhança (ou a menor diferença), um critério de decisão que utiliza a norma dos pesos de saída referente a cada nodo é utilizado para definir o nodo a ser removido e ajustar os pesos restantes da rede de forma a minimizar possíveis perdas de generalização.

Sejam dois nodos,  $r$  e  $t$ , pertencentes ao conjunto de nodos da camada escondida de uma MLP:  $(r, t) \in H$ , onde  $H$  é o número de nodos na camadas escondida. Sejam também dois vetores,  $\vec{f}_r^H$  e  $\vec{f}_t^H$ ,  $K$ -dimensionais, representando as saídas dos respectivos nodos em relação ao conjunto de treinamento.

Dois nodos apresentam respostas semelhantes se:

$$\|\vec{f}_r^H - \vec{f}_t^H\| = 0 \quad (7.10)$$

A partir de dois nodos semelhantes é possível definir uma resposta,  $\vec{f}_{rt}^H$ , como a média dos seus sinais. Se:  $\vec{f}_r^H = \vec{f}_t^H = \vec{a}$ ,

$$\vec{f}_{rt}^H = \frac{1}{2} (\vec{f}_r^H + \vec{f}_t^H) = \frac{1}{2} (\vec{a} + \vec{a}) = \vec{a} \quad (7.11)$$

Segundo as funções sigmoidais ímpares, a uma saída é atribuída apenas uma entrada. Dessa forma, para saídas iguais, as entradas lineares serão iguais:

$$\vec{f}_r^H = f_r^H \left[ \sum_{j=1}^N (\vec{x}_j \cdot w_{1j_r}) + b_{1r} \right]$$

$$\begin{aligned} \vec{f}_t^H &= f_t^H \left[ \sum_{j=1}^N (\vec{x}_j \cdot w_{1jt}) + b_{1t} \right] \\ (\vec{f}_r^H = \vec{f}_t^H) &\Rightarrow \sum_{j=1}^N (\vec{x}_j \cdot w_{1jr}) + b_{1r} = \sum_{j=1}^N (\vec{x}_j \cdot w_{1jt}) + b_{1t} \end{aligned}$$

Logo, também é possível definir um vetor resultante da média desses sinais:

$$\begin{aligned} \vec{f}_{rt}^l &= \frac{1}{2} \left\{ \sum_{j=1}^N (\vec{x}_j \cdot w_{1jr}) + b_{1r} + \sum_{j=1}^N (\vec{x}_j \cdot w_{1jt}) + b_{1t} \right\} \\ \vec{f}_{rt}^l &= \sum_{j=1}^N \left[ \vec{x}_j \cdot \left( \frac{w_{1jr} + w_{1jt}}{2} \right) \right] + \left( \frac{b_{1r} + b_{1t}}{2} \right) \end{aligned} \quad (7.12)$$

Na realidade, a Equação 7.12 representa uma simplificação de dois nodos que apresentam uma resposta semelhante. Contudo, a probabilidade de ocorrência de nodos semelhantes, mesmo que a rede possua vários nodos na camada escondida, é pequena. Normalmente, é possível que nodos apresentem respostas quase semelhantes:

$$\|\vec{f}_r^H - \vec{f}_t^H\| \approx 0 \quad (7.13)$$

onde

$$\vec{f}_r^H = \vec{a} \quad (7.14)$$

$$\vec{f}_t^H = \vec{a} + \vec{\xi}$$

sendo  $\vec{\xi}$  o vetor de diferenças de  $\vec{f}_r^H$  em relação a  $\vec{f}_t^H$  ou vice-versa. Nessa situação, o sinal médio será:

$$\vec{f}_{rt}^H = \vec{a} + \frac{\vec{\xi}}{2} \quad (7.15)$$

O algoritmo proposto nessa seção, para nodos quase semelhantes, procura manter um nodo,  $r$  ou  $t$ , e retirar o outro *conectando* os pesos do nodo retirado ao nodo escolhido. Tal fato se justifica porque, de acordo com a Equação 7.15, o uso da média das respostas irá resultar em um termo de ruído,  $\vec{\xi}/2$ , que irá se propagar para os nodos da camada seguintes. Se um nodo for mantido, o efeito desse termo de ruído pode ser minimizado através da escolha adequada do nodo a ser removido.

O método de escolha do nodo a ser mantido deve minimizar a influência do termo  $\vec{\xi}$  na camada seguinte da MLP. Para isso, os vetores  $\vec{f}_r^H$  e  $\vec{f}_t^H$  devem ser analisados em relação à saída dos nodos da próxima camada. Sendo o caso estudado, uma MLP de duas

camadas, a saída  $p$  da rede apresenta o seguinte comportamento:

$$\vec{y}_p = f_p \left\{ \sum_{i=1}^H w_{2ip} \cdot \vec{f}_i^H \right\} \quad (7.16)$$

Expandindo a Equação 7.16 de forma a evidenciar um par de nós,  $(r, t)$ , com respostas quase semelhantes,

$$\vec{y}_p = f_p \left\{ \sum_{i=1, i \neq r, i \neq t}^H w_{2ip} \cdot \vec{f}_i^H + w_{2rp} \cdot \vec{f}_r^H + w_{2tp} \cdot \vec{f}_t^H \right\} \quad (7.17)$$

é possível definir um vetor,  $\vec{v}_p$ , da seguinte forma:

$$\vec{v}_p = w_{2rp} \cdot \vec{f}_r^H + w_{2tp} \cdot \vec{f}_t^H \quad (7.18)$$

Substituindo a relação 7.14 em 7.18:

$$\vec{v}_p = (w_{2rp} + w_{2tp}) \cdot \vec{a} + w_{2tp} \cdot \vec{\xi} \quad (7.19)$$

se a relação 7.14 for contrária:

$$\begin{aligned} \vec{f}_r^H &= \vec{a} + \vec{\xi} \\ \vec{f}_t^H &= \vec{a} \end{aligned}$$

o vetor  $\vec{v}_p$  terá a seguinte forma:

$$\vec{v}_p = (w_{2rp} + w_{2tp}) \cdot \vec{a} + w_{2rp} \cdot \vec{\xi} \quad (7.20)$$

Em ambas as Equações 7.19 e 7.20, aparece o termo:  $(w_{2rp} + w_{2tp}) \cdot \vec{a}$ . Já os termos referentes ao ruído,  $\vec{\xi}$ , variam entre  $w_{2tp} \cdot \vec{\xi}$  e  $w_{2rp} \cdot \vec{\xi}$ , dependendo do vetor de referência. De posse desses resultados, é possível concluir que, para minimizar o ruído no nó de saída, é preciso determinar o peso de menor amplitude.

$$\min(\vec{v}_p) \propto \min(|w_{2rp}|, |w_{2tp}|) \quad (7.21)$$

Ao peso de menor amplitude não se atribui a saída de referência.

$$\begin{cases} \text{Se } |w_{2rp}| \leq |w_{2tp}|, \text{ então } \vec{f}_t^H = \vec{a} \\ \text{Se } |w_{2tp}| < |w_{2rp}|, \text{ então } \vec{f}_r^H = \vec{a} \end{cases}$$

Para múltiplas saídas ou nós na camada seguinte, utiliza-se a norma dos pesos conectados aos nós  $r$  e  $t$ ,  $\|w_{2r}\|$  e  $\|w_{2t}\|$ . A relação final para a escolha do nó de referência é a seguinte:

$$\begin{cases} \text{Se } \|w_{2_r}\| \leq \|w_{2_t}\|, \text{ então } \vec{f}_t^H = \vec{a} \\ \text{Se } \|w_{2_t}\| < \|w_{2_r}\|, \text{ então } \vec{f}_r^H = \vec{a} \end{cases} \quad (7.22)$$

o ajuste dos pesos das conexões do nodo de referência é obtida através do comportamento expresso na Equação 7.20:

$$\begin{cases} \text{Se } \|w_{2_r}\| \leq \|w_{2_t}\|, \text{ então: } w_{2_{tp}} = w_{2_{tp}} + w_{2_{rp}} \\ \text{Se } \|w_{2_t}\| < \|w_{2_r}\|, \text{ então: } w_{2_{rp}} = w_{2_{rp}} + w_{2_{tp}} \end{cases} \quad (7.23)$$

O ruído máximo,  $\vec{\xi}_{max}$ , aceitável é obtido empiricamente através da análise dos erros de treinamento e validação. Se após a retirada do nodo e ajuste dos pesos da rede o erro de validação for menor ou igual ao seu valor original, a simplificação é válida. O algoritmo utilizado nesse método é descrito a seguir.

---

### Algoritmo 3 Simplificação de Topologia através da Identificação de Nodos com Respostas Semelhantes

---

rede = Treina\_rede\_MLP\_com\_algoritmo\_MOBJ/SMC-MOBJ();

$H \Leftarrow$  número de nodos na camada escondida;

$e_{val} \Leftarrow$  obtém o erro de validação da rede treinada;

$e_{ref} = e_v$  {Erro de referência}

$\sigma_v \Leftarrow$  desvio aceitável;

**Enquanto**  $e_{val} \leq (e_{ref} + \sigma_v)$ , **faça**

$(r, t) \Leftarrow$  mínimo( $\|\vec{f}_r^H - \vec{f}_t^H\|, (r, t) \in H$ );

**Se**  $\|w_{2_r}\| \leq \|w_{2_t}\|$  **então**

$w_{2_{tp}} = w_{2_{tp}} + w_{2_{rp}}$ ;

retira\_nodo(r, rede);

**senão**

$w_{2_{rp}} = w_{2_{rp}} + w_{2_{tp}}$ ;

retira\_nodo(t, rede);

**fim Se**

$e_{val} \Leftarrow$  resposta\_da\_rede(rede);

**fim Enquanto**

---

Os resultados do processo de simplificação utilizando o método de identificação de nodos com semelhança de respostas são apresentados na Tabela 7.6. A topologia original do problema *cancer* foi reduzida para a topologia 9-6-2, que representa uma redução de 9 nodos na camada escondida. No problema *card*, o método foi capaz de reduzir 11 nodos. Nenhuma redução foi obtida para o problema *gene* ao passo que para a função  $d(x)$ , houve uma redução de 40 nodos, 4 nodos para a regressão da função *sinc* e apenas um nodo foi simplificado para o problema *building*. Em todos os casos, o desvio aceitável,  $\sigma_v$ , foi definido como nulo.

Tabela 7.6: Resultados provenientes da simplificação por semelhança de respostas

| Base de Dados | $E_t$  | $E_v$  | Topologia Inicial | Topologia Final | $\ w\ $ |
|---------------|--------|--------|-------------------|-----------------|---------|
| cancer        | 96.18% | 98.57% | 9-15-2            | 9-6-2           | 7.108   |
| card          | 88.89% | 88.04% | 51-15-2           | 51-4-2          | 7.147   |
| gene          | 98.85% | 89.53% | 120-15-3          | 120-15-3        | 8.831   |
| $d(x)$        | 0.0659 | 0.0069 | 1-50-1            | 1-10-1          | 5.461   |
| sinc          | 0.0730 | 0.0127 | 1-50-1            | 1-46-1          | 32.287  |
| building      | 0.0189 | 0.0227 | 14-15-3           | 14-14-3         | 6.9899  |

## 7.5 Simplificação de Topologias pela Simples Extração de Conexões

Este é o método mais simples de extração de conexões enunciado nesse capítulo. Também conhecido como método de pruning de *força bruta* [Reed, 1993], o método consiste na avaliação da resposta da rede quando uma conexão é anulada. Se o erro de treinamento aumenta com a retirada do peso, a conexão é re-estabelecida.

Dada uma topologia contendo  $n$  entradas,  $h$  nodos na camada escondida e  $p$  saídas, o número total de conexões ou pesos dessa rede pode ser calculado como:

$$C = n.h + h.p + h + p$$

O método da simplificação de topologia pela simples extração de conexões [Costa et al., 2002a], tem como objetivo extrair a quantidade máxima de conexões sem que haja alteração substancial do erro de validação em relação à topologia original. Para cada conexão ou peso extraído, o valor do erro de validação,  $e_v$ , é calculado. Se o erro de validação for menor ou igual ao seu valor inicial (rede sem simplificação) a respectiva conexão é retirada, caso contrário, a conexão é mantida e a próxima é então avaliada. O processo é repetido até que todas as  $C$  conexões sejam abrangidas. Nesse método, nenhum re-treinamento é realizado, o que o torna rápido, apesar de abranger todas as conexões.

A tabela 7.7 apresenta os resultados para o método de simplificação proposto. O sinal (\*), apresentado na coluna: Topologia Final, indica que a topologia final obtida também apresenta conexões extraídas entre as camadas e os termos de bias. Para o problema *cancer*, a estrutura final obtida apresenta 11 nodos extraídos além de pesos entre os nodos resultantes. A topologia final para o problema *card* apresentou, além de uma redução do número de nodos, uma simplificação da quantidade de entradas da rede, proporcionando um ligeiro aumento da capacidade de classificação em relação ao conjunto de validação. O mesmo comportamento pode ser observado em relação ao problema *gene*,

do qual foram extraídas 76 entradas e 3 nodos. Para os problemas de regressão, foram extraídos: 22 nodos para a função  $d(x)$ , somente conexões para a função *sinc* e 10 entradas e 6 nodos para função *building*. Em todos os casos, as respostas em relação aos conjuntos de validação são melhores que os resultados iniciais (Tabela 7.4).

Tabela 7.7: Resultados provenientes da simplificação pela simples extração de conexões

| Base de Dados | $E_t$  | $E_v$  | Topologia Inicial | Topologia Final | $\ \mathbf{w}\ $ |
|---------------|--------|--------|-------------------|-----------------|------------------|
| cancer        | 96.18% | 99.64% | 9-15-2            | 9-4-2*          | 6.172            |
| card          | 85.99% | 88.41% | 51-15-2           | 14-9-2*         | 3.185            |
| gene          | 90.08% | 90.31% | 120-15-3          | 44-12-3*        | 8.831            |
| $d(x)$        | 0.0702 | 0.0081 | 1-50-1            | 1-28-1*         | 5.050            |
| sinc          | 0.0729 | 0.0127 | 1-50-1            | 1-50-1*         | 32.382           |
| building      | 0.0257 | 0.0165 | 14-15-3           | 4-8-3*          | 6.007            |

## 7.6 Simplificação de Topologias Utilizando um Método de Saliência

Também conhecido como método de pruning baseado no cálculo da Hessiana [Haykin, 1999], o método utiliza informações da derivada de segunda ordem da superfície de erro para realizar um equilíbrio entre a complexidade da rede e o erro de treinamento. Um modelo para a superfície do erro é obtido para se avaliar os efeitos de perturbações nos pesos da rede. Esse modelo é obtido através da expansão da função de custo do erro, no ponto de operação, utilizando-se a Série de Taylor.

$$E(\mathbf{w} + \Delta\mathbf{w}) = E(\mathbf{w}) + g^T(\mathbf{w})\Delta\mathbf{w} + \frac{1}{2}\Delta\mathbf{w}^T\mathbf{H}\Delta\mathbf{w} + O(\|\Delta\mathbf{w}\|^3) \quad (7.24)$$

onde  $\Delta\mathbf{w}$  é a perturbação aplicada ao ponto de operação  $\mathbf{w}$ ,  $g(\mathbf{w})$  é o gradiente calculado no ponto  $\mathbf{w}$  e  $\mathbf{H}(\mathbf{w})$  a hessiana também definida no ponto  $\mathbf{w}$ .

O objetivo consiste em identificar um conjunto de parâmetros da MLP cuja extração produza o menor aumento no valor da função de custo do erro,  $E(\mathbf{w})$ . Mas, para isso, algumas condições são necessárias para diminuir a complexidade da Equação 7.24:

1. Assume-se que a rede foi treinada e encontra-se em um região de mínimo local onde:  
 $g^T(\mathbf{w})\Delta\mathbf{w} \approx 0$
2. A superfície da função de erro pode ser aproximada por uma superfície quadrática. Dessa forma, os termos de ordem mais elevadas da expansão podem ser ignorados:  
 $O(\|\Delta\mathbf{w}\|^3) \approx 0$

O método em questão será utilizado para determinar quais conexões serão eliminadas em uma rede MLP já treinada utilizando um algoritmo MOBJ/SMC-MOBJ. Mesmo que a rede inicial represente uma solução pertencente ao conjunto Pareto, ou seja, possuindo uma valor de erro e norma não nulos, a condição 1 será assumida. Sob essas condições a Equação 7.24 pode se simplificada:

$$\Delta E(\mathbf{w}) = E(\mathbf{w} + \Delta \mathbf{w}) - E(\mathbf{w}) \simeq \frac{1}{2} \Delta \mathbf{w}^T \mathbf{H} \Delta \mathbf{w} \quad (7.25)$$

A partir da Equação 7.25 foi desenvolvido o método OBS (*optimal brain surgeon*) [Hassibi and Stork, 1993]. O objetivo do método OBS é anular pesos para minimizar o aumento da função de erro,  $E(\mathbf{w})$ . Sendo  $w_i$  um determinado peso, a sua eliminação é equivalente à seguinte situação:

$$\Delta w_i + w_i = 0$$

$$\mathbf{1}_i^T \Delta \mathbf{w} + w_i = 0$$

onde  $\mathbf{1}_i$  é um vetor cujo único elemento não nulo e unitário é o elemento  $i$ . Pode-se quantificar o objetivo do método OBS da seguinte forma: minimizar a Equação 7.25 sujeito à restrição  $\mathbf{1}_i^T \Delta \mathbf{w} + w_i = 0$

Para resolver esse problema de otimização, utiliza-se o método de *Lagrange*

$$S = \frac{1}{2} \Delta \mathbf{w}^T \mathbf{H} \Delta \mathbf{w} - \lambda (\mathbf{1}_i^T \Delta \mathbf{w} + w_i) \quad (7.26)$$

onde  $\lambda$  é o multiplicador de Lagrange. Calculando a derivada de  $S$  em relação à  $\Delta \mathbf{w}$ , aplicando a restrição e utilizando inversão de matrizes, encontra-se o valor ótimo de  $\Delta \mathbf{w}$ :

$$\Delta \mathbf{w} = - \frac{w_i}{[\mathbf{H}^{-1}]_{i,i}} \mathbf{H}^{-1} \mathbf{1}_i \quad (7.27)$$

e, conseqüentemente, o valor ótimo do Lagrangiano  $S$  em relação ao elemento  $w_i$ :

$$S_i = \frac{w_i^2}{2[\mathbf{H}^{-1}]_{i,i}} \quad (7.28)$$

onde  $\mathbf{H}^{-1}$  é a matriz inversa da hessiana e  $[\mathbf{H}^{-1}]_{i,i}$  o elemento da sua diagonal  $i$ . O valor de  $S_i$  é também chamado de *saliência* de  $w_i$  e representa o aumento na função do erro em consequência da retirada da conexão ou peso  $w_i$ . No método OBS, o peso que possui o menor valor de saliência é removido sendo possível ajustar os demais pesos segundo a Equação 7.27. Apesar de ser um método com princípios matemáticos bem definidos, o OBS apresenta um alto custo computacional no cálculo da inversa da Hessiana. Para minimizar esse custo computacional, foi desenvolvido o método OBD (*optimal brain damage*) [Cun et al., 1989]. O método OBD assume uma matriz hessiana diagonal reduzindo os

cálculos para apenas esses elementos.

Nesta seção, o método OBS foi utilizado para a simplificação de topologias MLPs de duas camadas. Utilizando um pacote de identificação de sistemas [Norgaard, 1997], onde o erro de validação também foi avaliado, procurou-se reduzir a topologia da rede sem que houvesse aumento do erro de validação ( $\sigma_v = 0$ ). Os resultados obtidos bem como as topologias finais encontram-se na Tabela 7.8.

Tabela 7.8: Resultados provenientes da simplificação utilizando o método OBD

| Base de Dados | $E_t$  | $E_v$  | Topologia Inicial | Topologia Final | $\ w\ $ |
|---------------|--------|--------|-------------------|-----------------|---------|
| cancer        | 96.66% | 98.57% | 9-15-2            | 9-15-2*         | 7.489   |
| card          | 84.06% | 87.68% | 51-15-2           | 1-2-2*          | 2.123   |
| gene          | -      | -      | 120-15-3          | -               | -       |
| $d(x)$        | 0.0663 | 0.0070 | 1-50-1            | 1-28-1*         | 5.058   |
| sinc          | 0.0732 | 0.0127 | 1-50-1            | 1-50-1*         | 32.381  |
| building      | 0.0182 | 0.0196 | 14-15-3           | 14-12-3*        | 6.9694  |

O método apresentou um custo computacional muito elevado para o conjunto *gene*, sendo incapaz de realizar a simplificação.

Segundo a Tabela 7.8, os resultados para os problemas *sinc* e *cancer* apresentam somente simplificação de conexões. Para o problema *card*, o método foi capaz de classificar 87.68% do conjunto de validação utilizando apenas uma entrada, sendo que a rede original apresenta 51 entradas. Apenas 3 nodos foram simplificados para o problema *building*. O resultado obtido para o problema  $d(x)$  é semelhante àquele obtido utilizando o método da simples extração dos pesos.

## 7.7 Associação de Técnicas para Simplificação de Topologias

Várias técnicas apresentadas nesse capítulo permitem simplificar a topologia original de uma MLP através da análise de determinadas características que esta apresente. É o caso dos métodos de simplificação através da identificação de nodos com comportamento linear e com semelhança de respostas. Já os métodos de retirada de conexões como o OBS e o método de simples extração, apresentam resultados muito semelhantes mas, o custo computacional do método OBS é muito alto. Nessa seção, uma combinação de três métodos foi utilizada para a simplificação das topologias. São esses:

1. Identificação de Nodos com Semelhança de Respostas
2. Simplificação de Topologia pela Simples Extração de Conexões

## 3. Simplificação de Topologia através da Identificação de Nodos Lineares

Os resultados e as topologia finais encontram-se na Tabela 7.9

Tabela 7.9: Resultados provenientes da associação dos métodos de simplificação

| Base de Dados | $E_t$  | $E_v$  | Topologia Inicial | Topologia Final |        | $\ w\ $ |
|---------------|--------|--------|-------------------|-----------------|--------|---------|
|               |        |        |                   | não linear      | linear |         |
| cancer        | 96.18% | 99.64% | 9-15-2            | 9-2-2*          | 3-2    | 6.106   |
| card          | 85.27% | 88.77% | 51-15-2           | -               | 8-2    | 2.909   |
| gene          | 91.18% | 91.26% | 120-15-3          | 20-8-3*         | 1-1    | 6.435   |
| $d(x)$        | 0.0677 | 0.0074 | 1-50-1            | 1-6-1*          | -      | 5.490   |
| sinc          | 0.0759 | 0.0112 | 1-50-1            | 1-44-1*         | 1-1    | 32.236  |
| building      | 0.0257 | 0.0164 | 14-15-3           | 4-7-3*          | 1-2    | 5.9885  |

Os resultados obtidos pela combinação das técnicas enunciadas possibilitaram uma redução significativa das topologias originais se comparados ao uso dos métodos individuais. A topologia final para o problema *cancer* apresenta uma redução de 13 nodos, alguns pesos, além da existência de conexões lineares entre a saída e a entrada da rede. A topologia original do problema *card* foi completamente simplificada resultando em uma rede com uma única camada. Para o problema *gene*, a topologia final apresenta uma redução de 100 entradas, 7 nodos, alguns pesos, sendo uma conexão linear estabelecida entre os seus extremos. A simplificação das redes referentes à regressão das funções  $d(x)$  e *sinc* resultou na extração de 54 e 6 nodos escondidos, respectivamente, além da extração de pesos, sendo uma conexão linear estabelecida na topologia referente a função *sinc*. O problema de regressão *building* obteve uma redução de 10 entradas, 8 nodos e pesos, além de apresentar duas conexões lineares na topologia final.

Comparando os resultados obtidos com as Tabelas 7.8, 7.7, 7.6 e 7.5, verifica-se que os menores erros de validação foram obtidos através da combinação dos métodos de simplificação. Entretanto, principalmente em relação aos problemas de classificação, o favorecimento do erro de validação provocou uma queda da capacidade de classificação em relação ao conjunto de treinamento. Tal característica pode ser um indicativo de que o conjunto de validação ou de treinamento, ou ambos, não sejam estatisticamente representativos. Destaca-se que tal fenômeno favoreceu os dados referentes à função  $d(x)$  e *sinc*, uma vez que o conjunto de validação, em ambos os casos, trata-se da função geradora sem ruído. Surge, então, a necessidade de um ajuste *fino* para equilibrar os erros de treinamento e validação. Para que tal ajuste fosse realizado, o mesmo método aplicado aos dados da Tabela 7.9 foi utilizado sendo o conjunto de validação definido pela união dos conjuntos de treinamento e validação, ou seja, todo o conjunto de dados. A Tabela 7.10 apresenta os resultados.

Tabela 7.10: Resultados provenientes dos processos de simplificação combinados e com ajuste do conjunto de validação

| Base de Dados | $E_t$  | $E_v$  | Topologia Inicial | Topologia Final |        | $\ \mathbf{w}\ $ |
|---------------|--------|--------|-------------------|-----------------|--------|------------------|
|               |        |        |                   | não linear      | linear |                  |
| cancer        | 96.42% | 98.57% | 9-15-2            | 7-2-2*          | 1-1    | 6.223            |
| card          | 86.71% | 89.86% | 51-15-2           | 25-4-2*         | 10-2   | 3.152            |
| gene          | 97.64% | 91.34% | 120-15-3          | 111-15-3*       | -      | 8.598            |
| $d(x)$        | 0.0677 | 0.0074 | 1-50-1            | 1-6-1*          | -      | 5.490            |
| sinc          | 0.0737 | 0.0112 | 1-50-1            | 1-44-1*         | 1-1    | 32.247           |
| building      | 0.0131 | 0.0213 | 14-15-3           | 6-7-3*          | 2-2*   | 6.045            |

Segundo os resultados obtidos na Tabela 7.10, o uso do conjunto total de dados como sendo o conjunto de validação permitiu um melhor equilíbrio das respostas dos conjuntos individuais de validação e de treinamento. Os resultados provenientes dos problemas de classificação apresentam taxas de classificação próximas, o que era esperado. Houve uma perda da capacidade de aproximação das funções *sinc* e  $d(x)$  em virtude do conjunto de validação ser agora definido pela própria função geradora e pela função com ruído. As topologias resultantes possuem dimensões superiores àquelas obtidas utilizando o método anterior (Tabela 7.9).

## 7.8 Identificação de Topologias Utilizando re-Treinamento

Dentre os vários métodos utilizados na identificação de estruturas de redes MLPs, o método *Identificação Multi-Objetivo de Topologias* apresenta um caráter construtivo. Esse método, como visto, procura a topologia mínima capaz de apresentar o menor erro de treinamento para um determinado valor de norma,  $\|\mathbf{w}^*\|$ . É um método eficiente, mas que necessita de uma inicialização aleatória dos pesos para cada topologia testada.

O algoritmo proposto nesta seção parte do princípio de que uma rede superdimensionada foi treinada e apresenta uma resposta com boa capacidade de generalização, com baixa norma, ou seja, a rede foi treinada utilizando um algoritmo multi-objetivo. A partir desse ponto, é utilizado o método de simplificação por semelhança de respostas para a retirada de um nodo da camada escondida. Uma vez retirado o nodo, o valor da norma da rede restante é calculado e um treinamento restringindo a rede a esse valor da norma é realizado armazenando-se o erro final obtido. O processo é repetido até que todos os nodos sejam retirados. A topologia que apresentar o menor erro para o conjunto de validação é definida como a rede de topologia mínima capaz de obter uma boa resposta para o problema.

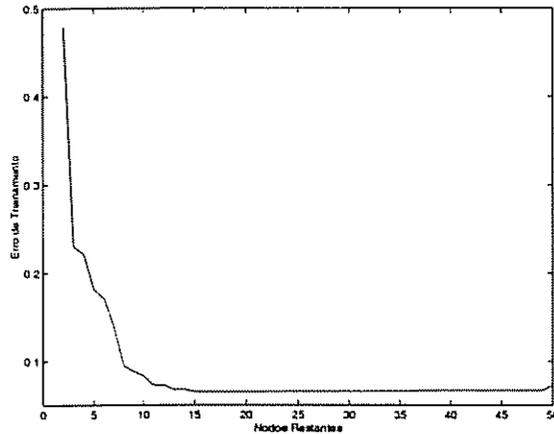


Figura 7.12: Erro de Treinamento em Função dos Nodos Restantes

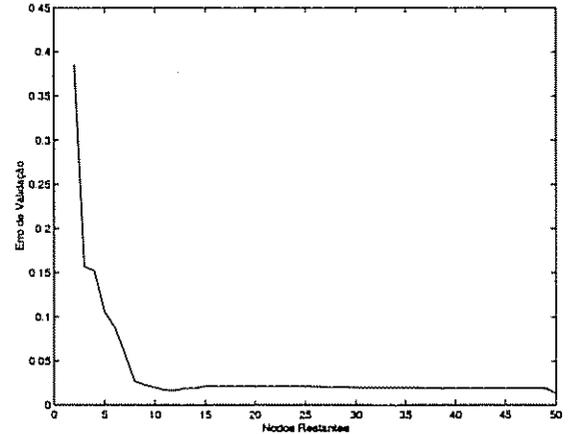


Figura 7.13: Erro de Validação em Função dos Nodos Restantes

As Figuras 7.12 e 7.13 devem ser analisadas da direita para a esquerda pois, a rede utilizada nesse exemplo apresenta inicialmente 50 nodos na camada escondida. A cada nodo retirado, a rede é re-treinada utilizando o algoritmo SMC-MOBJ. A principal vantagem do método consiste na sua capacidade de manter informação armazenada na rede após a retirada do nodo. Como a rede inicial já apresenta uma boa resposta, a retirada de um nodo não implica na perda total da informação armazenada, exceto quando há nodos insuficientes na camada escondida. Essa *memória* da rede permite uma melhor convergência do algoritmo ao ponto desejado.

Após a retirada do primeiro nodo, a rede apresenta uma ligeira queda do erro de treinamento e um aumento no erro de validação. Tal comportamento indica que a solução que está sendo alcançada está ligeiramente deslocada em relação à melhor solução ou solução ótima. Entretanto, tal característica não impede o perfeito funcionamento do algoritmo. Uma vez determinada a topologia ideal é possível *corrigir* tal solução obtendo-se pontos do conjunto Pareto próximos a essa solução e determinando o ponto de melhor resposta. De acordo com a Figura 7.13, a topologia mínima para o problema em questão apresenta 11 nodos na camada escondida. O exemplo utilizado corresponde à regressão da função *sinc*.

Tabela 7.11: Resultados do processo de identificação de topologias utilizando re-treinamento

| Base de Dados | $E_t$  | $E_v$  | Topologia Final | $\ w\ $ |
|---------------|--------|--------|-----------------|---------|
| cancer        | 97.14% | 97.86% | 9-8-2           | 7.431   |
| card          | 89.86% | 87.32% | 51-2-2          | 2.821   |
| gene          | 98.85% | 89.53% | 120-15-3        | 8.831   |
| $d(x)$        | 0.0614 | 0.0137 | 1-7-1           | 5.600   |
| sinc          | 0.0764 | 0.0163 | 1-11-1          | 26.862  |
| building      | 0.003  | 0.0248 | 14-5-3          | 6.9694  |

A Tabela 7.11 mostra os resultados do processo de simplificação das redes MLPs utilizando o método de re-treinamento. Os resultados finais podem ser aprimorados utilizando o treinamento multi-objetivo a fim de determinar todo o conjunto Pareto para cada base de dados, selecionando o ponto de máxima generalização. A principal vantagem desse método consiste no baixo esforço computacional em relação ao método Identificação Multi-Objetivo de Topologias, no qual todo o conjunto Pareto é obtido para cada topologia proposta.

## 7.9 Conclusões do Capítulo

Neste capítulo foram apresentados alguns métodos para a simplificação de redes MLPs treinadas a partir do uso de algoritmos multi-objetivo (MOBJ/SMC-MOBJ). Os algoritmos multi-objetivo: MOBJ e SMC-MOBJ, permitem obter soluções de equilíbrio entre o erro e a norma garantindo uma rede com boa capacidade de generalização. Por outro lado, os algoritmos multi-objetivo propostos não possibilitam a simplificação da topologia original durante o treinamento.

A simplificação das redes obtidas utilizando os algoritmos multi-objetivo, visa a definição de uma estrutura mínima que apresente o mesmo desempenho da estrutura original. Este é o motivo pelo qual a maior parte dos métodos apresentados não promovem o treinamento e sim a análise de características específicas da rede. Os métodos apresentados nesta seção estão listados a seguir:

1. Identificação Multi-Objetivo de Topologias
2. Comportamento Linear/não-Linear do nodo Perceptron
3. Identificação de Nodos com Semelhança de Respostas
4. Simplificação de Topologia pela Simples Extração de Conexões
5. Simplificação de Topologia Utilizando um Método de Saliência
6. Combinação de Técnicas para Simplificação de Topologias
7. Identificação de Topologias Utilizando re-Treinamento

Dentre esses métodos, apenas o primeiro e o último necessitam do uso do algoritmo SMC-MOBJ. Os demais utilizam a resposta das redes e as características físicas das conexões e nodos para simplificar a topologia. Os melhores resultados foram obtidos a partir da combinação dos métodos 2, 3 e 4. O método 5, apesar de apresentar um comportamento semelhante ao método 4, apresenta um custo computacional muito elevado.

Foi analisado que, em alguns exemplos, um ajuste fino no desempenho da rede é necessário através da adição do conjunto de treinamento ao conjunto de validação. Essa técnica foi aplicada somente ao caso 6 onde não foi realizado nenhum treinamento. Tal fato pode ser explicado uma vez que, se o conjunto de validação apresentar peculiaridades em relação ao conjunto de treinamento, o teste para a minimização desse conjunto pode provocar uma *suave* discrepância em relação ao conjunto de treinamento durante a retirada das conexões, favorecendo-o. Essa técnica de ajuste *fino* permite obter um melhor equilíbrio da resposta da rede em relação aos conjuntos de dados.

Em relação aos métodos de identificação de topologias através de treinamento, o método 1 apresentou o melhor resultado na identificação do número mínimo de nodos na camada escondida de uma MLP, em relação ao método 7.

A partir dos resultados obtidos, pode-se concluir que é possível simplificar a topologia de uma rede MLP treinada por um algoritmo multi-objetivo sem que haja perda da sua capacidade de generalização. Através do método de simplificação e extração de conexões, é possível identificar as principais entradas e conexões que atuam no modelo neural, extraindo-se informações da rede simplificada.



## Capítulo 8

# Conclusões e Propostas de Continuidade

O método MOBJ (multi-objetivo) para treinamento de RNAs [Teixeira et al., 2000] é capaz de gerar modelos neurais com alta capacidade de generalização, desde que a topologia utilizada seja suficientemente flexível. As soluções multi-objetivo são obtidas através da minimização do erro restringindo a norma dos pesos a determinados valores possibilitando, assim, o controle da flexibilidade do modelo. Uma vez minimizado o erro referente a uma determinada norma, e definido um intervalo discreto de normas, a solução que apresenta o menor erro em relação a um conjunto de validação é escolhida.

O método SMC-MOBJ [Costa et al., 2002c, Costa et al., 2003] proposto para o treinamento multi-objetivo de MLPs é capaz de obter as soluções pertencentes ao conjunto Pareto-ótimo do qual é possível extrair a solução de máxima generalização. Apesar de apresentar uma velocidade de convergência compatível com o algoritmo *backpropagation* padrão, uma vez que utiliza o ajuste segundo os gradientes das funções do erro e da norma, o método é capaz de convergir para a solução desejada desde que essa se encontre na região factível de soluções. O método não utiliza técnicas complexas de otimização, sendo simples e viável a sua implementação computacional.

Uma vez definidos os gradientes referentes à função do erro e da norma, é possível impor um controle ao ajuste dos pesos através da especificação de duas *superfícies de deslizamento*. Essas superfícies controlam a direção dos gradientes impondo comportamentos de minimização e maximização de forma a convergir o estado da rede (erro e norma) a um estado pré-estabelecido.

Uma análise dos ganhos utilizando a Teoria de Modos Deslizantes [Itkis, 1976] permitiu definir uma região para os ganhos do algoritmo proposto de modo que a convergência da rede às superfícies de deslizamento, que representam a solução desejada, seja garantida. A partir da equação de ajuste dos pesos segundo o método SMC-MOBJ foi possível obter variações através do uso de técnicas de otimização. Consequentemente, obteve-se o ajuste

dos pesos segundo o método multi-objetivo de Levenberg-Marquardt (MOBJ-SMC-LM) [Costa et al., 2002b] que possibilita a redução do número de iterações durante a geração das soluções pertencentes ao Pareto. O método proposto foi obtido através da adição de um termo de restrição da norma junto à equação padrão de ajuste dos pesos segundo o método de Levenberg-Marquardt.

Uma terceira variação do método SMC-MOBY foi proposta através da decomposição dos gradientes e utilizando a função norma dos pesos para a definição de intervalos para a taxa de aprendizado (MOBY-BP). O resultados demonstram a capacidade do método de obter as soluções eficientes. O método MOBY-BP realiza o controle da norma através da decomposição dos gradientes e o ajuste é realizado definindo-se ganhos para esses gradientes de forma que a norma final dos pesos esteja condicionada a um pequeno intervalo no qual se encontra a norma desejada, mantendo a convergência do erro. A principal característica do método consiste no uso da equação de ajuste segundo o algoritmo *back-propagation* padrão.

Uma vez caracterizada a capacidade dos métodos propostos de obter redes superdimensionadas com alta capacidade de generalização, foram propostos métodos multi-objetivo construtivos para encontrar topologias reduzidas de MLPs. Dentre os algoritmos desenvolvidos, o método *Identificação Multi-Objetivo de Topologias* gera o conjunto Pareto para um conjunto de topologias. A menor topologia que apresenta o menor erro em relação ao conjunto de validação é escolhida. Como os métodos multi-objetivo são capazes de obter a solução de melhor generalização para uma determinada topologia, e sendo o comportamento do conjunto Pareto-ótimo estável a partir de uma topologia mínima, o método é capaz de obter uma topologia reduzida com o melhor desempenho. A desvantagem desse método consiste no custo computacional para a geração das curvas Pareto. Para cada topologia testada é necessário gerar todo o respectivo conjunto Pareto. A fim de reduzir esse custo computacional, foi proposta uma variação na qual somente um ponto do conjunto Pareto é analisado. Essa solução é gerada a partir do método SMC-MOBY que permite minimizar a função do erro restringindo a norma dos pesos a um único valor. Dessa forma, várias topologias são testadas restringindo-se a norma dos pesos. O mesmo comportamento do conjunto Pareto é analisado definindo-se a solução de erro mínimo e menor topologia. Para cada topologia testada somente um ponto do conjunto Pareto é gerado.

Métodos de *pruning* [Costa et al., 2002a] foram utilizados na análise de soluções multi-objetivo para simplificar a topologia obtida. Os métodos propostos procuram simplificar topologias já treinadas utilizando o método multi-objetivo, reduzindo o custo computacional do processo de simplificação. Uma análise da resposta da função de ativação dos nodos de uma MLP permite obter um método de simplificação capaz de substituir um nodo por conexões lineares sem perda de desempenho. Esse método de redução de topologia possibilita, em alguns casos, a completa simplificação de uma rede MLP de duas camadas,

resultando em uma topologia com uma única camada. Uma vez utilizado o método, é possível identificar saídas que sejam linearmente relacionadas às entradas.

Os demais métodos utilizam características ou propriedades dos nodos ou pesos da rede para realizar a *poda* de conexões: o método de identificação de nodos com semelhança de respostas, procura por pares de nodos que apresentam respostas semelhantes, que são medidas pelo erro quadrático de suas saídas. Pares de nodos semelhantes são substituídos por um único nodo. O método da simplificação de topologia pela simples extração de conexões permite extrair não somente nodos mas entradas da rede e pesos entre as camadas. Tal característica permite identificar as principais entradas para o modelo neural que garantem a sua generalização. Dentre os métodos de pruning propostos, os melhores resultados foram obtidos a partir da combinação dos métodos, resultando em topologias híbridas com redução de entradas, pesos, nodos e conexões lineares estabelecidas entre as entradas e as saídas dos modelos neurais. O uso associado dos métodos propostos possibilita, em alguns casos, obter topologias finais compatíveis com aquelas obtidas utilizando o método construtivo *Identificação Multi-Objetivo de Topologias*.

## 8.1 Propostas de continuidade

Vários métodos para o treinamento multi-objetivo de MLPs foram apresentados neste trabalho bem como diversas técnicas para a redução de topologias e análise dos parâmetros de RNAs. Sob esse aspecto, podem ser identificados diversos tópicos de pesquisa. Alguns estão relacionados a seguir:

- Na seção 3.4 foi demonstrado um método de decomposição de pesos em coordenadas polares. A princípio, o método foi utilizado para gerar soluções de mesma norma em um nodo perceptron constituído por um peso e um termo de polarização (bias). Uma vez que sejam definidas restrições de busca para os pesos, é possível utilizar este método para gerar as soluções do conjunto Pareto-ótimo. As restrições podem ser obtidas através de uma análise detalhada das equações dos gradientes dos pesos. O método de decomposição também pode ser utilizado para medir as variações dos ângulos nas soluções do conjunto Pareto-ótimo, na tentativa de identificar padrões de comportamento.
- Variações do algoritmo SMC-MOBJ foram propostas, como o método multi-objetivo de Levenberg-Marquardt (MOBJ-SMC-LM) e o controle da norma dos pesos através da taxa de aprendizado (MOBJ-BP). Diversos métodos de otimização podem ser aplicados ao problema multi-objetivo utilizando o método de controle por Modos Deslizantes para direcionar a convergência da rede.

- Outros métodos de pruning podem ser utilizados na simplificação de topologias além de uma análise dos métodos já propostos.
- Os algoritmos propostos e os métodos de simplificação podem ser implementados em outros tipos de redes como, por exemplo, redes do tipo RBF.
- Avaliação estatística da capacidade dos métodos multi-objetivo no aumento da capacidade de generalização de RNAs.

# Referências Bibliográficas

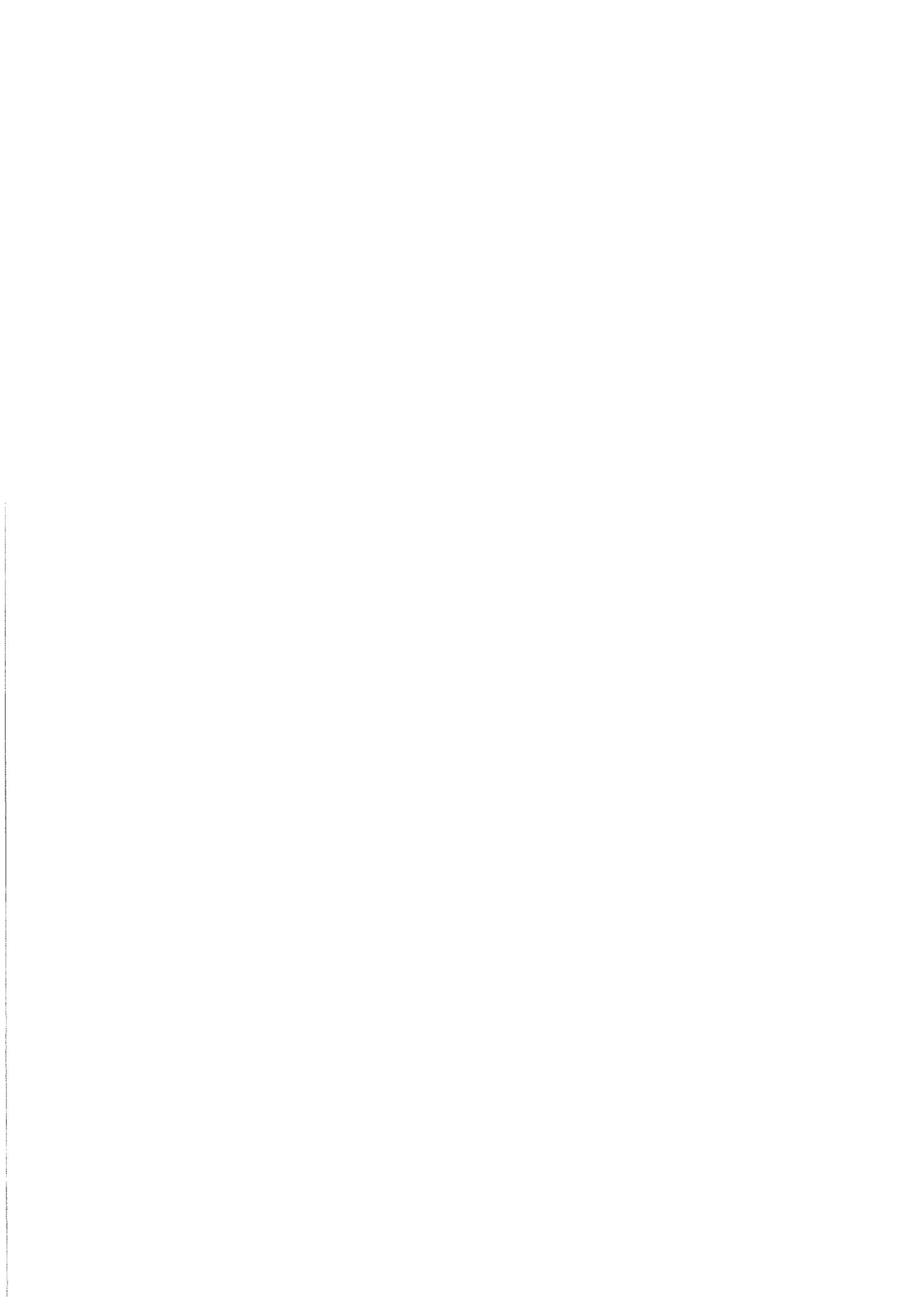
- [Barron, 1993] Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoid function. *IEEE Transactions on Information Theory*, 39:930–945.
- [Bartlett, 1997] Bartlett, P. L. (1997). For valid generalization, the size of the weights is more important than the size of the network. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*, pages 134–140, Cambridge, MA: The MIT Press.
- [Battiti and Massuli, 1990] Battiti, R. and Massuli, F. (1990). BFGS optimization for faster and automated supervised learning. In *Proceedings of the International Neural Network Conference*, volume 2, pages 757–760, Paris.
- [Blake and Merz, 1998] Blake, C. and Merz, C. (1998). UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- [Braga et al., 2000] Braga, A., Carvalho, A., and Ludermir, T. (2000). *Redes Neurais Artificiais: teoria e aplicações*. Livros Técnicos e Científicos (LTC).
- [Butkov, 1978] Butkov, E. (1978). *Física Matemática*.
- [Chankong and Haimes, 1983] Chankong, V. and Haimes, Y. (1983). *Optimization-based Methods For Multiobjective Decision-making - an Overview*, volume 5, chapter 1, pages 1–33. ELSEVIER SCIENCE BV, AMSTERDAM.
- [Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20:273–279.
- [Costa et al., 2002a] Costa, M., Braga, A., and de Menezes, B. (2002a). Constructive and pruning methods for neural network design. In *XVI Brazilian Symposium on Neural Networks*.
- [Costa et al., 2002b] Costa, M., Braga, A., and de Menezes, B. (2002b). Improved generalization learning with sliding mode control and the levenberg-marquadt algorithm. In *XVI Brazilian Symposium on Neural Networks*.

- [Costa et al., 2002c] Costa, M., Braga, A., de Menezes, B., Parma, G., and Teixeira, R. (2002c). Control of generalization with a bi-objective sliding mode control algorithm. In *XVI Brazilian Symposium on Neural Networks*.
- [Costa et al., 2003] Costa, M. A., Braga, A., de Menezes, B., Parma, G., and Teixeira, R. (2003). Training neural networks with a multi-objective sliding mode control algorithm. *Neurocomputing*. accepted.
- [Costa et al., 2000] Costa, M. A., de Pádua Braga, A., and Aguirre, A. (2000). 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*.
- [Cun et al., 1989] Cun, Y. L., Denker, J. S., and Solla, S. A. (1989). Optimal brain damage. In *Proceedings of NIPS-89*, pages 598–605.
- [Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314.
- [Engelbrecht, 2001] Engelbrecht, A. (2001). A new pruning heuristic based on variance analysis of sensitivity information. *IEEE Transactions on Neural Networks*, 12:1386–1399.
- [Fahlman, 1988] Fahlman, S. E. (1988). Faster-learning variations on backpropagation: an empirical study. In Touretzky, D., Hinton, G., and Sejnowski, T., editors, *Proceedings of the 1988 Connectionist Models Summer School, Pittsburg*, pages 38–51, San Mateo, CA: Morgan Kaufmann.
- [Geman et al., 1992] Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58.
- [Girosi et al., 1995] Girosi, F., Jones, M., and Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269.
- [Hagan and Menhaj, 1994a] Hagan, M. T. and Menhaj, M. B. (1994a). Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993.
- [Hagan and Menhaj, 1994b] Hagan, M. T. and Menhaj, M. B. (1994b). Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993.
- [Hassibi and Stork, 1993] Hassibi, B. and Stork, D. G. (1993). Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems (5)*, pages 164–171.

- [Haykin, 1999] Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Tom Robbins - Prentice Hall, Inc.
- [Hinton and Nowlan, 1987] Hinton, G. E. and Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems*, 1(1):495–502. 1987.
- [Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational facilities. *Proceedings of the National Academy of Science of the USA*, 79:2554–2558.
- [Itkis, 1976] Itkis, U. (1976). *Control Systems of Variable Structure*. Keter Publishing House Jerusalem LTD.
- [Kramer and S.-Vincentelli, 1989] Kramer, A. H. and S.-Vincentelli, A. (1989). Efficient parallel learning algorithms for neural networks. In in *Neural Information Processing Systems*, A., editor, *D. S. Touretzky (Ed.)*, pages 40–48, San Mateo. CA: Morgan Kaufmann.
- [Kreyszig, 1993] Kreyszig, E. (1993). *Advanced Engineering Mathematics*. John Wiley and Sons, Inc., 7 edition.
- [Lehmann, 1987] Lehmann, C. H. (1987). *Geometria Analítica*. Editora Globo.
- [Mangasarian and Musicant, 1999] Mangasarian, O. L. and Musicant, D. R. (1999). Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks*. 10(5):1032–1037.
- [Marquardt, 1963] Marquardt, D. W. (1963). An algorithm for least squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441.
- [McCulloch and Pitts, 1943] McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin OF Mathematical Biophysics*, 5:115–133.
- [Minsky and Papert, 1969] Minsky, M. L. and Papert, S. A. (1969). *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA:MIT Press.
- [Moller, 1993] Moller, M. (1993). A scaled conjugate-gradient algorithm for fast supervised learning. *Neural Networks*, 6:525–533.
- [Norgaard, 1997] Norgaard, M. (1997). Neural network based system identification toolbox. Technical report, Department of Automation, Technical University of Denmark.

- [Pareto, 1896] Pareto, V. (1896). *Cours D'Economie Politique*, volume I and II. Rouse, Lausanne.
- [Parma et al., 1998] Parma, G. G., de Menezes, B. R., and Braga, A. P. (1998). Sliding mode algorithm for training multilayer artificial neural networks. *Electronics Letters*, 34(1):97–98.
- [Parma et al., 1999] Parma, G. G., de Menezes, B. R., and Braga, A. P. (1999). Neural networks learning with sliding mode control: The sliding mode backpropagation algorithm. *International Journal of Neural Systems*, 9(3):187–193.
- [Reed, 1993] Reed, R. (1993). Pruning algorithms - a survey. *IEEE Transactions on Neural Networks*, 4(5):740–746.
- [Riedmiller and Braun, 1993] Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Proceedings of the IEEE Intl. Conf. on Neural Networks*, pages 586–591, San Francisco, CA.
- [Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage in the brain. *Psychological review*, 65:386–408.
- [Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representation by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. Cambridge, MA:MIT Press.
- [Shor, 1977] Shor, N. Z. (1977). Cut-off method with space extension in convex programming problems. *Cybernetics*, 12:94–96.
- [Sietsma and Dow, 1991] Sietsma, J. and Dow, R. (1991). Creating artificial neural networks that generalize. *Neural Networks*, 4:67–79.
- [Silva and Almeida, 1990] Silva, F. M. and Almeida, L. B. (1990). Speeding up backpropagation. In Eckmiller, R., editor, *Advanced Neural Computers*, pages 151–158, Amsterdam:Elsevier North Holland.
- [Sira-Ramírez and Colina-Morles, 1995] Sira-Ramírez, H. and Colina-Morles, E. (1995). A sliding mode strategy for adaptive learning in adalines. *IEEE Transactions on Circuits and Systems*, 42(12):1001–1012.
- [Sira-Ramírez et al., 2000] Sira-Ramírez, H., Colina-Morles, E., and Rivas-Echeverría, F. (2000). Sliding mode-based adaptive learning in dynamical-filter-weights neurons. *International Journal of Control*, 73(8):678–685.

- [Stone, 1974] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, B36:111–133.
- [Takahashi et al., 1997] Takahashi, R. H. C., Peres, P. L. D., and Ferreira, P. A. V. (1997). H<sub>2</sub>/H-infinity multiobjective PID design. *IEEE Control Systems Magazine*, 17(5):37–47.
- [Teixeira, 2001] Teixeira, R. A. (2001). *Treinamento de Redes Neurais Artificiais Através de Otimização Multi-Objetivo: Uma Nova Abordagem para o Equilíbrio entre a Polarização e a Variância*. PhD thesis, CPDEE, UFMG.
- [Teixeira et al., 2000] Teixeira, R. A., Braga, A. P., Takahashi, R. H. C., and Saldanha, R. R. (2000). Improving generalization of MLPs with multi-objective optimization. *Neurocomputing*, 35(1–4):189–194.
- [Tollenaere, 1990] Tollenaere, T. (1990). SuperSAB: Fast adaptive back propagation with good scaling properties. *Neural Networks*, 3(5):561–573.
- [Utkin, 1978] Utkin, V. I. (1978). *Sliding modes and their application in variable structure systems*. MIR, Moscow.
- [Vapnik, 1995] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.
- [Weigend et al., 1990] Weigend, A. S., Huberman, B. A., and Rumelhart, D. E. (1990). Predicting the future: a connectionist approach. *International Journal of Neural Systems*. 1:193–209.
- [Widrow and Hoff, 1960] Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. In *IRE WESCON Convention Record*. volume 4, pages 96–104. New York: IRE.



# Apêndice A

## Apêndice: Equações Matemáticas e Propriedades

### A.1 Transformação de Coordenadas em $\mathbb{R}^2$

Sejam as seguintes propriedades em relação à **Transformação de Coordenadas** em um plano  $\mathbb{R}^2$  definido pelos eixos  $x$  e  $y$  [Lehmann, 1987]:

**Propriedade 1:** Translação dos eixos coordenados

Se os eixos coordenados são transladados para uma nova origem  $O'(h, k)$  e se as coordenadas de qualquer ponto  $P$  antes e depois da translação são  $(x, y)$  e  $(x', y')$ , respectivamente, então as equações de transformação das antigas para as novas são dadas por:

$$x = x' + h$$

$$y = y' + k$$

Da mesma forma, as equações de transformação das coordenadas novas para as antigas são dadas por:

$$x' = x - h$$

$$y' = y - k$$

**Propriedade 2:** Rotação dos eixos coordenados

Se os eixos coordenados são girados de um ângulo  $\theta$  em torno de sua origem  $O$  como ponto fixo e se as coordenadas de qualquer ponto  $P$  são  $(x, y)$  e  $(x', y')$  antes e depois da rotação, respectivamente, então as equações de transformação das antigas para as novas coordenadas são dados por:

$$x = x' \cos \theta - y' \sin \theta$$

$$y = x' \operatorname{sen} \theta - y' \operatorname{cos} \theta$$

Essa transformação também pode ser escrita na forma matricial:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \operatorname{cos} \theta & -\operatorname{sen} \theta \\ \operatorname{sen} \theta & -\operatorname{cos} \theta \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}$$

## A.2 A equação geral do segundo grau: o Indicador $I = b^2 - 4ac$

A equação geral do segundo grau em relação às variáveis  $x, y$ :

$$a \cdot x^2 + b \cdot xy + c \cdot y^2 + d \cdot x + e \cdot y + f,$$

onde  $\{a, b, c, d, e, f\} \in \mathfrak{R}$

é do tipo parabólico, elíptico ou hiperbólico, conforme seu indicador,

$$I = b^2 - 4ac,$$

seja zero, negativo ou positivo, respectivamente [Lehmann, 1987].

### A.2.1 Caso particular

Seja a equação do segundo grau em relação às variáveis  $\alpha, \beta$ :

$$a_1 \cdot \alpha^2 + a_2 \cdot \alpha\beta + a_3 \cdot \beta^2 + a_4 \cdot \alpha + a_5 \cdot \beta + a_6 \tag{A.1}$$

onde  $\{a_1, a_2, a_3, a_4, a_5, a_6\} \in \mathfrak{R}$ . Se os coeficientes reais  $a_1, a_2$  e  $a_3$  forem definidos pelas Equações A.2, A.3 e A.4,

$$a_1 = \frac{1}{2} \sum_{m=1}^Q x_m^2 \tag{A.2}$$

$$a_2 = \sum_{m=1}^Q x_m \cdot y_m \tag{A.3}$$

$$a_3 = \frac{1}{2} \sum_{m=1}^Q y_m^2 \tag{A.4}$$

onde  $x$  e  $y$  são vetores reais definidos em  $\mathfrak{R}^Q$ . O indicador,  $I$  ( $I = a_2^2 - 4 \cdot a_1 a_3$ ) apresenta a seguinte característica:

- $I = 0$ , para  $Q = 1$

Definindo:  $x_1 = x$  e  $y_1 = y$ ,

$$I = (x \cdot y)^2 - 4 \cdot \frac{1}{2} x^2 \cdot \frac{1}{2} y^2$$

$$I = x^2 \cdot y^2 - x^2 \cdot y^2 = 0$$

A Equação A.1 é do tipo parabólico.

- $I \leq 0$ , para  $Q \neq 1$  ( $Q > 1$ )

Utilizando a notação vetorial:

$$I = (\mathbf{x} \cdot \mathbf{y})^2 - 4 \cdot \frac{\mathbf{x} \cdot \mathbf{x}}{2} \cdot \frac{\mathbf{y} \cdot \mathbf{y}}{2}$$

$$I = (|\mathbf{x}| \cdot |\mathbf{y}| \cdot \cos\theta)^2 - |\mathbf{x}|^2 \cdot |\mathbf{y}|^2$$

$$I = |\mathbf{x}|^2 \cdot |\mathbf{y}|^2 \cdot \cos^2\theta - |\mathbf{x}|^2 \cdot |\mathbf{y}|^2$$

como:  $0 \leq \cos\theta \leq 1$ ,

$$I \leq 0$$

A Equação A.1, nesse caso, é do tipo elipsoidal se  $\mathbf{x} \neq \mathbf{y}$ , ou parabólica se  $\mathbf{x} = \mathbf{y}$ ,  $\mathbf{x} = -\mathbf{y}$ .

### A.3 Determinação das raízes de um polinômio de 2º grau

A equação de um polinômio de ordem 2 em relação à variável  $x$  pode ser descrita como:

$$a \cdot x^2 + b \cdot x + c$$

onde  $a$ ,  $b$ ,  $c$  são coeficiente reais. Para o cálculo das raízes, deve-se determinar o valor de  $\Delta$ :

$$\Delta = b^2 - 4 \cdot a \cdot c$$

As raízes podem ser definir de acordo com o valor de  $\Delta$ :

- $\Delta \geq 0$ , as raízes são calculadas utilizando a seguinte equação:

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

- $\Delta < 0$ , não existem raízes reais distintas
- $\Delta = 0$ , existe uma raiz real dupla definida como:

$$x = \frac{-b}{2a}$$

O comportamento da concavidade da parábola está associado ao sinal do coeficiente  $a$ , como pode ser visualizado na Figura A.1.

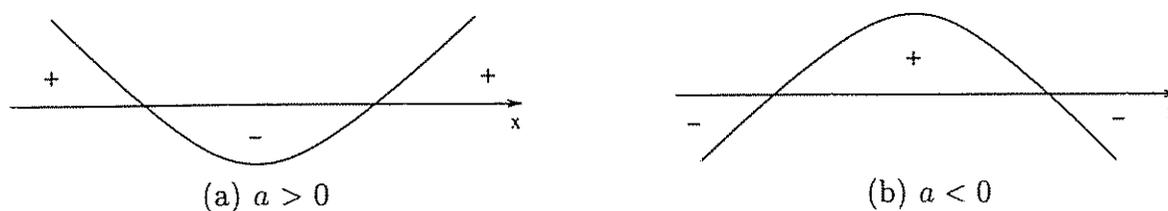


Figura A.1: Comportamento da concavidade da parábola em relação ao coeficiente  $a$

## Apêndice B

# Apêndice: Resultados para o Banco de Dados Proben - Classificação (Cancer, Card, Gene)

Os testes foram realizados considerando uma amostragem de 10 redes MLP por algoritmo de treinamento, exceto para os algoritmos MOBJ e SMC-MOBJ. As tabelas expressam os valores médios:  $\mu_{E_t}$ ,  $\mu_{E_v}$  e  $\mu_{||w||}$ ; seus respectivos desvios:  $\sigma_{E_t}$ ,  $\sigma_{E_v}$  e  $\sigma_{||w||}$ ; e o melhor resultados obtido para cada algoritmo. Em todos os casos o número de nodos na camada escondida foi mantido constante e igual a 10 (dez), sendo o número de entradas e saídas definido pelo respectivo conjunto de dados. O número de entradas e saídas bem como os casos utilizados são mostrados a seguir:

**Cancer** : 9 entradas, 2 saídas, 699 amostras

**Card** : 51 entradas, 2 saídas, 960 amostras

**Gene** : 120 entradas, 3 saídas, 3175 amostras

O conjunto de treinamento e validação forma gerados a partir do fracionamento do conjunto total de dados na seguinte proporção: 60% para o conjunto de treinamento e 40%, validação. Os parâmetro de treinamento para cada algoritmo são listados a seguir:

**Parma(1)** - (SMC1)  $\alpha = \beta = 0.01$ ,  $C = 1$ ,  $C_H = 2$ ;

**Parma(2)** - (SMC2)  $\alpha = \beta = 0.001$ ,  $C = 1$ ,  $C_H = 3$ ;

**BckPrg** - (BackPropagation)  $\alpha = 0.1$ ;

**LevenMrqt** - (Levenberg-Marquardt)  $\mu = 0.001$ ,  $\mu^- = 0.1$ ,  $\mu^+ = 10$ ;

**Rprop** - (Resilient BackPropagation)  $\alpha = 0.1$ ;

Early Stopping - (Rprop)  $\alpha = 0.1$ ;

WD - (Weight Decay)  $\alpha = 0.003, \beta = 0.001$ ;

Os algoritmo SMC-MOBY e MOBY são capazes de *adaptar* os seus ganhos de modo a melhorar a sua convergência, não sendo necessária a especificação de parâmetros de ganho para o treinamento.

| Algoritmo | Treinamento     |                    | Validação       |                    | Norma         |                  | Melhor Resultado |         |            |
|-----------|-----------------|--------------------|-----------------|--------------------|---------------|------------------|------------------|---------|------------|
|           | $\mu_{E_t}(\%)$ | $\sigma_{E_t}(\%)$ | $\mu_{E_v}(\%)$ | $\sigma_{E_v}(\%)$ | $\mu_{  w  }$ | $\sigma_{  w  }$ | $E_{(k)}$        | $  w  $ | Acertos(%) |
| Parma(1)  | 98.04           | 0.19               | 96.25           | 0.30               | 11.97         | 0.53             | 0.2948           | 12.18   | 96.79      |
| Parma(2)  | 99.69           | 0.32               | 95.68           | 0.36               | 25.39         | 1.98             | 0.0049           | 27.28   | 96.07      |
| BckPrg    | 85.23           | 21.83              | 84.57           | 20.69              | 16.37         | 0.65             | 0.0544           | 16.81   | 97.50      |
| LevenMrqt | 99.36           | 0.67               | 95.43           | 0.96               | 390.99        | 727.24           | 0.0095           | 163.71  | 96.79      |
| Rprop     | 99.31           | 0.61               | 94.96           | 1.43               | 75.37         | 33.08            | 0.0382           | 39.62   | 96.79      |
| Early Stp | 97.33           | 0.54               | 97.29           | 0.35               | 20.06         | 8.07             | 0.0418           | 42.06   | 97.86      |
| SMC-MOBY  | 97.61           | -                  | 96.79           | -                  | 4.84          | -                | 0.0472           | 4.84    | 96.79      |
| MOBY      | 96.42           | -                  | 98.57           | -                  | 7.49          | -                | 0.0659           | 7.49    | 98.57      |
| WD        | 97.28           | 0.23               | 97.21           | 0.15               | 4.03          | 0.15             | 0.2846           | 4.11    | 97.50      |

Tabela B.1: Resultados obtidos através dos dados: Cancer/Proben

| Algoritmo | Treinamento     |                    | Validação       |                    | Norma         |                  | Melhor Resultado |         |            |
|-----------|-----------------|--------------------|-----------------|--------------------|---------------|------------------|------------------|---------|------------|
|           | $\mu_{E_t}(\%)$ | $\sigma_{E_t}(\%)$ | $\mu_{E_v}(\%)$ | $\sigma_{E_v}(\%)$ | $\mu_{  w  }$ | $\sigma_{  w  }$ | $E_{(k)}$        | $  w  $ | Acertos(%) |
| Parma(1)  | 96.84           | 1.53               | 83.62           | 0.78               | 26.87         | 0.75             | 0.1126           | 27.00   | 85.14      |
| Parma(2)  | 97.22           | 0.56               | 84.13           | 1.78               | 39.53         | 1.07             | 0.0697           | 39.80   | 86.59      |
| BckPrg    | 90.65           | 0.68               | 83.26           | 0.76               | 13.00         | 0.40             | 0.1574           | 12.70   | 84.78      |
| LevenMrqt | 99.18           | 2.09               | 79.49           | 3.15               | 126.86        | 82.58            | 0.0048           | 117.91  | 84.06      |
| Rprop     | 99.35           | 0.28               | 78.88           | 1.66               | 469.34        | 369.70           | 0.0145           | 537.63  | 80.80      |
| Early Stp | 90.31           | 1.31               | 86.23           | 1.17               | 37.81         | 23.91            | 0.1702           | 26.71   | 88.78      |
| SMC-MOBY  | 90.34           | -                  | 86.59           | -                  | 4.30          | -                | 0.1666           | 4.30    | 86.59      |
| MOBY      | 87.44           | -                  | 88.04           | -                  | 2.60          | -                | 0.2355           | 2.60    | 88.04      |
| WD        | 93.48           | 0.75               | 87.10           | 0.92               | 7.83          | 0.32             | 0.1241           | 8.12    | 88.04      |

Tabela B.2: Resultados obtidos através dos dados: Card/Proben

| Algoritmo | Treinamento     |                    | Validação       |                    | Norma         |                  | Melhor Resultado |         |            |
|-----------|-----------------|--------------------|-----------------|--------------------|---------------|------------------|------------------|---------|------------|
|           | $\mu_{E_t}(\%)$ | $\sigma_{E_t}(\%)$ | $\mu_{E_v}(\%)$ | $\sigma_{E_v}(\%)$ | $\mu_{  w  }$ | $\sigma_{  w  }$ | $E_{(k)}$        | $  w  $ | Acertos(%) |
| Parma(1)  | 94.84           | 0.98               | 78.65           | 1.62               | 38.07         | 0.74             | 0.0967           | 37.22   | 81.41      |
| Parma(2)  | 92.49           | 6.06               | 77.90           | 3.90               | 107.71        | 5.51             | 0.0745           | 103.95  | 80.32      |
| BckPrg    | 96.65           | 0.69               | 87.74           | 0.38               | 10.37         | 0.05             | 0.0937           | 10.28   | 88.50      |
| LevenMrqt | -               | -                  | -               | -                  | -             | -                | -                | -       | -          |
| Rprop     | 99.87           | 0.06               | 85.92           | 1.07               | 55.43         | 10.12            | 0.0016           | 48.66   | 87.56      |
| Early Stp | 98.47           | 0.86               | 86.79           | 0.77               | 15.77         | 0.77             | 0.0853           | 15.76   | 88.43      |
| SMC-MOBY  | 99.84           | -                  | 89.76           | -                  | 8.42          | -                | 0.0319           | 8.42    | 89.76      |
| MOBY      | 98.85           | -                  | 89.53           | -                  | 8.83          | -                | 0.0438           | 8.83    | 89.53      |
| WD        | 99.82           | 0.06               | 87.25           | 0.66               | 11.65         | 0.14             | 0.0108           | 11.61   | 88.04      |

Tabela B.3: Resultados obtidos através dos dados: Gene/Proben

Na Tabela B.3, o alto custo computacional do algoritmo Levenberg-Marquardt impediu a sua aplicação nesse caso.

Nos resultados apresentados, o método SMC-MOBY foi utilizado para gerar um conjunto discreto de soluções provenientes do conjunto Pareto-ótimo ao passo que, uma variação no método MOBY  $\epsilon$ -restrito, permitiu obter a melhor solução encontrada em um intervalo do conjunto Pareto-ótimo. Ou seja, o método MOBY utilizado gerou um número

maior de soluções com um conseqüente aumento do tempo de convergência do método em relação às soluções SMC-MOBJ geradas. Tal fato justifica a pequena discrepância entre as normas e os valores dos erros obtidos para os dois métodos.

