UFMG - Universidade Federal de Minas Gerais
Departamento de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica

Rede perceptron com camadas paralelas (PLP - Parallel Layer Perceptron)

Douglas Alexandre Gomes Vieira

Texto submetido à Banca Examinadora designada pelo colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais como requisito necessário para conclusão do doutorado.

Orientador: *Prof. João A. de Vasconcelos*
Co-Orientador: *Prof. Walmir M. Caminhas*
Advisor at Oxford University: *Prof. Vasile Palade*

Belo Horizonte, 2006.

Esta tese é dedicada aos meus pais, irmãos e à menina de meus olhos, Giulia.

"... ce ne sera qu'avec une ardente patience que
nous conquerrons la ville splendide qui donnera
lumière, justice et dignité à tous les hommes.
Et ainsi la poésie n'aura pas chanté en vain... "

"... só com uma ardente paciência que
conquistaremos a cidade explêndida que dará
luz, justiça e dignidade a todos os homens.
Neste dia a poesia não terá cantado em vão...".

A. Rimbaud

À Família

Poderia facilmente escrever mil palavras para agradecer à minha família pelo apoio incomparável que tive para trilhar estes caminhos tortuosos da vida. Tive de tudo que precisei, principalmente uma vida feliz nestes anos. Minha casa foi meu refúgio onde recarreguei as minhas baterias.

Minha mãe sempre me trouxe de volta à realidade, mesmo em momentos nos quais cheguei a pensar que a realidade era somente uma diversão passageira. Trabalhar com teoria pode ser muito alienante.

No meio das equações do método MGM para a PLP, que é o núcleo desta tese, há um singelo desenho de uma "menina super-poderosa"! Então acho que esta tese tem a co-autoria de minha menina, Giulia, com os super poderes depositados no meu rascunho naquele dia.

Meu pai sempre acreditou, e me fez acreditar que o céu é o limite. É por isto que sempre apostei alto, pois ainda estou longe do meu limite. Sou engenheiro por causa dele, precisa dizer mais?

Em dois shows do DEUK, a banda de rock na qual eu tocava antes de ir para Oxford, meu irmão cantou comigo algumas músicas. A música sempre foi a minha ciência favorita que compartilho com o meu irmão.

Melhor tia e irmã do mundo quebrou um monte de galhos para mim, principalmente no tempo que estive em Oxford. Quando os meus sobrinhos chegarem espero retribuir.

Então, esta é a vida real, o que a minha família me deu de melhor...

Aos Amigos

Aos meus amigos da Engenharia representados pelo Túlio, Sérgio Marvadeza, Rodrigo Valadão, Adriano e Ricardo. Grupo de mentes brilhantes brigando para encontrar um lugar ao sol. O melhor para vocês é meu desejo.

Aos meus amigos guerreiros Marcelo Gerardi, Newton Velloso, Guido Melo e Xuxa por estes mais de 10 anos de convívio em tempos de ondas boas e ruins. Nossos sonhos são os mesmos há muito tempo...

To mine Oxon Friends Claire, Kristell, and Cinthia for the company in the cold, cloudy, grayish and rainy Oxford. You are the proof that good friendship is not a matter of time or geography; you were the sunlight among the clouds.

Aos amigos de infância Eduardo Saldanha e Igor por tudo que aprendemos juntos.

Agradecimentos aos mestres

Prof. João foi meu orientador desde a iniciação científica nos idos de 2001. Motivou me a seguir a carreira científica guiando mas me dando liberdade o suficiente para evoluir. Teve a coragem, que poucos teriam, de me indicar ao doutorado sem passar pelo mestrado. Sei que muitos professores não me deixariam alçar vôos tão altos, mas o Prof. João sempre foi um grande entusiasta. Meus sinceros agradecimentos pelos 5 anos de convívio.

Prof. Walmir me convidou, quando eu ainda era aluno de graduação, para desenvolver uma rede, que no final se transformou na PLP, tema central desta tese. Foi o primeiro professor que notei que se importava e confiava no potencial dos alunos. Este fato fez com que o trabalho final da disciplina "Inteligência Computacional" que cursei com ele 4 anos atrás se transformasse na tese de doutorado que apresento agora. Obrigado pela confiança depositada desde o começo.

Prof. Vasile was my tutor during my one-year stay at Oxford University. He explained me about the cheapest ways to travel in Europe, where to buy food and so on. On the top of these we also researched! Oxford was a remarkable moment in my life, one of these turning points that seldom happens. Thanks for the incentive and advices.

Ao Prof Takahashi pela ajuda sistemática em diversos temas que contribuiram em muito nesta tese.

Ao Prof. Rodney que sempre aberto a conversa contribuiu em muito na minha formação científica e na construção de muitas de minhas idéias. Serei eternamente grato.

Esta tese tem "pedaços" de todos estes mestres.

Gauche

Oh!!! Vasto-mundo
de grandiosa-dimensão
tu és mundo-gigante
mas cabenomeucoração
Oh!!! Geodésico-mundo-insignificante
mesmo mundo-grande
mundo-nada
Solitária imensidão...

Resumo

Este trabalho apresenta uma nova abordagem para lidar com o problema de minimização do risco estrutural (structural risk minimization - SRM) aplicado ao problema geral de aprendizado de máquinas. A formulação é baseada no conceito fundamental de que o aprendizado supervisionado é um problema de otimização bi-objetivo onde dois objetivos conflitantes devem ser minimizados. Estes objetivos estão relacionados ao erro de treinamento, risco empírico (R_{emp}), e à complexidade (capacidade) da máquina de aprendizado (Ω). Neste trabalho uma formulação geral baseada na norma-Q é utilizada para calcular a complexidade da máquina e esta pode ser utilizada para modelar e comparar a maioria das máquinas de aprendizado encontradas na literatura. A principal vantagem da medida proposta é que esta é uma maneira simples de separar as influências dos parâmetros lineares e não-lineares na medida de complexidade, levando a um melhor entendimento do processo de aprendizagem. Uma nova máquina de aprendizado, a rede perceptron com camadas paralelas (Parallel Layer Perceptron -PLP), foi proposta neste trabalho utilizando um treinamento baseado nas definições e estruturas de aprendizado propostas nesta tese, o Método do Gradiente Mínimo (Minimum Gradient Method-MGM). A combinação da PLP com o MGM (PLP-MGM) é feita utilizando o estimador de mínimos quadrados, sendo esta a principal contribuição deste trabalho.

Abstract

This work presents a novel approach to deal with the structural risk minimization (SRM) applied to a general machine learning problem. The formulation is based on the fundamental concept that supervised learning is a bi-objective optimization problem in which two conflicting objectives should be minimized. The objectives are related to the training error, empirical risk (R_{emp}), and the machine complexity (Ω). In this work one general Q-norm like method to compute the machine complexity is presented and it can be used to model and compare most of the learning machines found in the literature. The main advantage of the proposed complexity measure is that it is a simple method to split the linear and non-linear complexity influences, leading to a better understanding of the learning process. One novel learning machine, the Parallel Layer Perceptron (PLP) network was proposed here using a training algorithm based on the definitions and structures of learning, the Minimum Gradient Method (MGM). The combination of the PLP with the MGM (PLP-MGM) is held using a reliable least-squares procedure and it is the main contribution of this work.

Lista de Abreviações

AB	AdaBoost
ABR	Regularized AdaBoost
CV	Cross validation - Validação cruzada
ERM	Empirical Risk Minimization - Minimização do risco empírico
ES	Early stop - parada prematura
FFT	Fast Fourier Transform - Transformada rápida de Fourier
iid	Independente e identicamente distribuída
KFD	Kernel Fisher Discriminant - Discriminante de Fisher usando kernels
LASSO	Generalized Least Absolute Shrinkage and Selection Operator
LM	Levenberg-Marquardt
LSE	Least Squares Estimate - Estimador de mínimos quadrados
LOO	Leave-One-Out KFD
LOOM	Leave-One-Out SVM
MGM	Minimum Gradient Method - Método do gradiente mínimo
MLP	Multi-layer perceptron - Rede perceptron de múltiplas camadas
MOBJ	Multiobjetivo
MSE	Mean Squared Error - Erro Quadrático Médio
OBD	Optimal Brain Damage
PLP	Parallel Layer Perceptron - Rede perceptron com camadas Paralelas
PO	Pareto-ótima
PPSVM	Posterior Probability SVM
RBF	Radial Basis Function - Funções de base radial
SRM	Structural Risk Minimization - Minimização do risco estrutural
SVM	Support Vector Machine - Máquinas de vetores suporte
VC	Vapnik Chervonenkis
Xval	k-fold KFD

Lista de Simbolos

C	Matriz que multiplicada pelos parâmetros lineares resulta na saída da máquina
D	Matriz que multiplicada pelos parâmetros lineares resulta na derivada da saída da máquina
$e(.)$	Erro para um dado padrão
$err(.)$	Erro de generalização
$\mathbf{E}[.]$	Esperança matemática
$f(.)$	Função ou hipótese
$f_1(.)$	Função que representa a minimização do risco empírico
$f_2(.)$	Função que representa a minimização da capacidade da função
$F(.)$	Transformada de Fourier de $f(.)$
$F(x)$	Função Distribuição de probabilidade
$F(y\|x)$	Função de Distribuição condicional de y dado x
$F(y,x)$	Função de Distribuição Conjunta de y e x
$\mathcal{F}(.)$	Espaço das funções ou hipóteses
$fat_{\mathcal{F}}(\gamma)$	A dimensão *fat-shattering*
$G(.)$	Função de crescimento
h	Dimensão Vapinik e Chernonekis (VC dimnesion)
H	Pesos da camada escondida de uma MLP
$H(.)$	Esperança da entropia aleatória
$H1(.)$	Entropia aleatória
$H^{ann}(.)$	Entropia recozida
i	Índice
$I(.)$	Função indicadora
j	Índice e número imaginário na definição da transformada de Fourier
$J(.)$	Jacobiano
k	Número de erros no conjunto de treinamento
$k(.)$	Núcleo (Kernel)
l	Parâmetros lineares
$L(.)$	Medida de perda ou discrepância
$\mathcal{L}(.)$	Lagrangeano
$m(.)$	Menor margem do conjunto
$M(.)$	Margem em relação ao conjunto de treinamento
n	Dimensão do espaço de entradas, $x \in \mathbb{R}^n$
$N(.)$	Diversidade de uma classe de funções
$\Omega(.)$	Medida de complexidade (capacidade)
R	Esfera que contem os vetores $x \in X$

$R(.)$	Funcional de risco
$R_{emp}(.)$	Funcional de risco empírico
S	Conjunto de treinamento
P	Parâmetros lineares (Matriz para a PLP e vetor para os polinômios)
$P(.)$	Probabilidade de um evento
p_{ji}	Componente da matriz de pesos P
Q	Matriz que define a medida de complexidade, $Q = A^T A$
t	Contador referente as amostras do conjunto de treinamento
T	Número de amostras no conjunto de treinamento
U	Pesos da camada de saída de uma MLP
V	Parâmetros não-lineares (Matriz para a PLP)
v_{ji}	Componente da matriz de pesos V
w	Parâmetros da máquina de aprendizado, $w \in \Lambda$
x	Variáveis de entrada ou controle
X	Espaço onde o vetor x está definido, $x \in X$
yd	Resposta obtida do supervisor, saída desejada
z	Variável aleatória, $z = (x, y)$, $z \in Z$
α	Multiplicador de Lagrange, termo de ponderação
δ	Constante positiva ->Probabilidade de um evento ocorrer
Δ	Margem de separação para funções lineares no espaço das variáveis
ϵ	Constante positiva
η	Passo nos métodos baseados no gradiente
γ	Margem de separação no espaço das funções
$\gamma(.)$	Função de ativação
λ	Termo de regularização
Λ	Espaço de hipóteses
ω	Freqüência
$\phi(.)$	Função não linear utilizada para a construção de máquinas de aprendizagem
ϱ	Constante não infinita
$\vartheta(.)$	Medida de perda ou discrepância no espaço $\mathbb{R}^{n+1}$
τ	Ordem dos aproximadores polinomiais
$\theta(.)$	Função degrau
$\xi(.)$	Variável de folga de uma margem (soft margin)
ζ	Limite superior do risco esperado
∇f	Gradiente de f
$\tilde{O}(.)$	Ordem de grandeza assintótica
:	Tal que (such that)

Lista de Figuras

Lista de Tabelas

Sumário

Parte I

Considerações preliminares

Capítulo 1

Introdução

Um dos grandes desejos do homem, que aparece junto com a evolução das máquinas, é a concepção de uma máquina que possa operar em ambientes desconhecidos por ela, utilizando o seu próprio aprendizado, independente do controle humano. Uma máquina que possa ser chamada de autônoma ou cognitiva. A capacidade de lidar com eventos para os quais não foi previamente treinada determinaria o sucesso, ou insucesso desta.

Dentro deste contexto busca-se encontrar modelos que sejam capazes de resolver tarefas complexas, e que sejam eficientes computacionalmente. Entre as possíveis técnicas destacam-se as redes perceptron de múltiplas camadas (MLPs), as Redes de Funções de Base Radial (RBFs) e as Máquinas de Vetores Suporte (SVMs). Para uma referência introdutória sobre o assunto consulte [Hay99].

Ao tentar utilizar técnicas de aprendizado de máquina em problemas reais algumas características são desejáveis:

- bons resultados para o problema dado;
- tempo computacional da máquina de aprendizado (quanto mais rápido melhor);
- e que esta seja pouco dependente do projetista, i.e., seja pouco dependente da interferência humana na definição dos parâmetros de treinamento.

Estas características desejáveis motivaram o desenvolvimento desta tese de doutorado. Para se abordar estes três pontos é necessário estudar tanto

questões teóricas como práticas. As teóricas são: o que é um bom resultado para um dado problema? Quais são os fatores que levam a uma máquina de aprendizado a alcançar tal resultado? Depois nasce a questão prática de como utilizar estas idéias de forma eficiente computacionalmente. Todos estes aspectos são considerados neste trabalho.

O princípio que norteou a área de aprendizado de máquinas supervisionado durante muitos anos foi o princípio indutivo da minimização do risco empírico (Empirical Risk Minimization - ERM). Este é baseado no conceito simples de que a minimização do erro de treinamento, o risco empírico, é uma boa estratégia para se gerar máquinas com bom aprendizado. Entretanto, foi teoricamente provado que a minimização do risco empírico não era suficiente para garantir a convergência da máquina de aprendizagem[Vap98]. Para garantir a consistência do aprendizado surge então o princípio indutivo da minimização do risco estrutural (Structural Risk Minimization - SRM). Este princípio diz que além de se minimizar o risco empírico é necessário também limitar a capacidade (complexidade) da máquina de aprendizagem.

Este trabalho apresenta uma nova abordagem para lidar com o problema de minimização do risco estrutural (structural risk minimization - SRM) aplicado ao problema geral de aprendizado de máquinas. A formulação é baseada no conceito fundamental de que o aprendizado supervisionado é um problema de otimização bi-objetivo onde dois objetivos conflitantes devem ser minimizados. Estes estão relacionados ao erro de treinamento, risco empírico (R_{emp}), e a complexidade (capacidade) da máquina de aprendizado (Ω).

Neste trabalho uma formulação geral baseada na norma-Q é utilizada para calcular a complexidade da máquina de aprendizagem e esta pode ser utilizada para modelar e comparar a maioria das máquinas de aprendizado encontradas na literatura. A principal vantagem da medida de complexidade proposta é que esta é uma maneira simples de separar as influências dos parâmetros lineares e não-lineares na medida de complexidade, levando a um melhor entendimento do processo de aprendizagem. Matematicamente isto significa que, dados os parâmetros lineares l e os não-lineares V, a complexidade pode ser escrita como $\Omega = (A(V)l)^T A(V)l = l^T Q(V)l$. Observe que a complexidade é a norma do vetor $A(V)l$, e, como o produto entre $A(V)$ e l é utilizado, esta pode ser naturalmente implementada em paralelo. Para tal a rede perceptron com camadas paralelas (Parallel Layer Perceptron -PLP) com uma camada linear em paralelo com uma camada não-linear é uma solução natural. A PLP também é uma contribuição deste trabalho.

Muitas definições da matriz Q, dado $x^T Qx > 0$, $\forall x \neq 0$, podem ser uti-

lizadas nesta construção. Nesta tese a minimização da norma do gradiente da saída da rede é utilizada, gerando o Método do Gradiente Mínimo (Minimum Gradient Method-MGM). A combinação da PLP com o MGM (PLP-MGM) é feita utilizando o estimador de mínimos quadrados, sendo esta a contribuição prática deste trabalho.

O texto está dividido em duas partes, sendo a primeira considerações gerais sobre o problema de aprendizado de máquinas (capítulos 2 e 3), e a segunda as contribuições deste trabalho. No capítulo 2 é definido o problema de aprendizado de máquina que será tratado, o aprendizado supervisionado. As idéias apresentadas neste seguirão a linha do aprendizado estatístico. No capítulo 3 serão mostradas de forma sucinta algumas máquinas de aprendizado que aplicam o princípio indutivo da minimização do risco estrutural (SRM). Serão tratadas as MLPs e suas diversas técnicas, muitas heurísticas, para a minimização do risco estrutural, assim como as RBFs e as SVMs. Será dada ênfase nas propriedades de cada método envolvido, e não em detalhes práticos. Estas propriedades serão de suma relevância para o entendimento da técnica proposta.

No capítulo 4 é apresentada a norma-Q como uma medida de complexidade e as suas implicações. Uma medida que emprega a idéia citada acima é gerada utilizando a minimização do gradiente da saída da rede. A derivação desta segue a linha dos resultados apresentados no capítulo 2. O método do gradiente mínimo proposto neste capítulo escreve um problema multi-objetivo, na realidade bi-objetivo, onde se deseja minimizar dois objetivos conflitantes, o erro empírico e a norma do gradiente em um domínio de interesse. O método proposto é comparado do ponto de vista teórico com os métodos descritos no capítulo 3, sendo que existem diversas equivalências entre todos os métodos como mostrado no Capítulo 4.

Utilizando as idéias descritas no capítulo 4, é derivado o aproximador polinomial de gradiente mínimo. A idéia de se aplicar a formulação apresentada no capítulo 4 em polinômios, antes de analisar a aplicabilidade em redes neurais, se deve à simplicidade dos mesmos, simplificando as análises. De fato, para aproximadores polinomiais, a formulação teórica desenvolvida no capítulo 4 se transforma em um problema prático onde as funções envolvidas são convexas. Em alguns experimentos realizados durante este trabalho ficou demonstrado que se trata de uma ferramenta aplicável inclusive em problemas onde as amostras estão contaminadas por ruídos. Entretanto, também é importante ressaltar que os polinômios sofrem do famoso "mal da dimensionalidade" sendo pouco aplicáveis em problemas de alta dimensão (muitas

variáveis de entrada).

Visando resolver os problemas apresentados para os polinômios, no capítulo 6 é proposta a rede perceptron com camadas paralelas (PLP- Parallel Layer Perceptron). Um dos casos particulares da PLP utiliza duas camadas em paralelo, sendo que uma utiliza um aproximador polinomial, de fato um aproximador linear para evitar o mal da dimensionalidade. Para este caso particular é derivada a rede PLP de gradiente mínimo, onde o ajuste do gradiente é realizado somente na camada polinomial para aproveitar a convexidade dos funcionais envolvidos. A rede PLP com gradiente mínimo é a principal contribuição prática deste trabalho. Ainda no capítulo 6 alguns exemplos utilizando dados sintéticos são mostrados e a técnica proposta se mostrou superior às demais testadas.

Finalmente no capítulo 7 a técnica proposta é testada em problemas reais, entre eles problemas benchmark na área de aprendizado de máquinas e um problema eletromagnético. Algumas considerações finais são apresentadas no capítulo 8, entre estas uma visão geral do trabalho e algumas perspectivas futuras.

1.1 Publicações

Neste momento alguns trabalhos derivados deste texto já foram publicados, sendo quatro publicações em revistas e sete em congressos. No presente momento 3 artigos estão submetidos revistas e um para congresso. A lista dos trabalhos é apresentada a seguir:

1. W. M. CAMINHAS, D. A. G. VIEIRA and J. A. VASCONCELOS - "Parallel Layer Perceptron" - Neurocomputing (Elsevier), no. 55, pp. 771-778, October 2003.

2. D. A. G. VIEIRA, R. H. C. TAKAHASHI, V. PALADE, J. A. VASCONCELOS and W. M. CAMINHAS - "The Q-norm complexity measure and the Minimum Gradient Method: a novel approach to the machine learning structural risk minimization problem" - submitted to IEEE Transactions on Neural Networks on September-2005.

3. D. A. G. VIEIRA, J. A. VASCONCELOS and W. M. CAMINHAS - "Controlling the Parallel Layer Perceptron complexity using a multiobjective learning algorithm" - Neural Computing and Applications, accepted to publication (available on-line).

4. D. G. VIEIRA, D. A. G. VIEIRA, W. M. CAMINHAS, J. A. VASCONCELOS - "A Hybrid Approach Combining Genetic Algorithm and Sensitivity Information Extracted from Parallel Layer Network" - IEEE Transactions on Magnetics, v. 41, n. 5, p. 1740-1743, 2005.

5. D. A. G. VIEIRA, W. M. CAMINHAS and J. A. VASCONCELOS - "Extracting Sensitivity Information of Electromagnetic Device Models from a Modified ANFIS Topology" - IEEE Transactions on Magnetics, vol. 40, no. 2, pp. 1188-1191, March 2004.

6. D. A. G. VIEIRA, J. A. VASCONCELOS and W. M. CAMINHAS - "Improving Neural Networks Sensitivity Extraction of Electromagnetic Devices Using the Parallel Layer Perceptron Trained with the Minimum Gradient Method" - 12th CEFC - IEEE Conference on Electromagnetic Field Computation - Abril/2006, Miami, USA.

7. D. A. G. VIEIRA, W. M. CAMINHAS and J. A. VASCONCELOS - "Multiobjective Methodology to Compare Neural Networks Applied in Electromagnetics" - 11th CEFC - IEEE Conference on Electromagnetic Field Computation (Digital proceedings) - June/2004 Seoul, Korea.

8. D. G. VIEIRA, D. A. G. VIEIRA, W. M. CAMINHAS and J. A. VASCONCELOS - "A Hybrid Approach Combining Genetic Algorithm and Sensitivity Information Extracted from a Parallel Layer Perceptron" - 11th CEFC - IEEE Conference on Electromagnetic Field Computation (Digital proceedings) - June/2004 Seoul, Korea.

9. D. A. G. VIEIRA, W. M. CAMINHAS, J. A. RAMIREZ and J. A. VASCONCELOS - "Extracting Sensitivity Information of Electromagnetic Device Models from a Modified ANFIS Topology" - 14th COMPUMAG - IEEE Conference on the Computation of Electromagnetic Fields (Digital proceedings) - P 95279 - July/2003 -Saratoga Springs, New York, USA.

10. T. MEDEIROS, D. A. G. VIEIRA, W. M. CAMINHAS, J. A. VASCONCELOS, A. P. BRAGA, R. R. SALDANHA and R. T. ALBUQUERQUE -"Neural networks trained with multiobjective learning algorithm applied to optimization problems" - 24th CILAMCE -Iberian Latin-American Congress on Computational Methods in Engineering (Digital proceedings) - CIL 625-39 - October/2003 - Ouro Preto, Brazil.

11. D. A. G. VIEIRA, J. A. VASCONCELOS and W. M. CAMINHAS - "Multiobjective Methodology to Compare Neural Networks Applied to Eletromagnetics" - Submitted to Neural Computing and Applications on September-2005.

12. D. A. G. Vieira, J. A. Vasconcelos, W. M. Caminhas and Vasile Palade - "The Minimum Gradient Complexity Control Applied to Sensitivity Extraction of Electromagnetic Devices" - 16th COMPUMAG - IEEE Conference on the Computation of Electromagnetic Fields - Germany.

13. L. Travassos, D. A. G. VIEIRA, N. Ida, C. Vollaire and A. Nicolas - "Characterizing inclusions in a non-homogenous GPR problem by Neural Networks" - 16th COMPUMAG - IEEE Conference on the Computation of Electromagnetic Fields - Germany.

14. L. Travassos, D. A. G. VIEIRA, N. Ida, C. Vollaire and A. Nicolas - "Concrete Inclusion Assessment using Principal Component Analysis and Neural Networks " - submetido para IEEE Geoscience and Remote Sensing Letters.

15. A. James, D. A. G. VIEIRA, D. Ara, and G-Z Yang - "Surgical workflow segmentation using foveal window information" - submetido para Miccai.

Em [CVV03], item 1, é proposto a topologia da rede perceptron com camadas paralelas, alguns de seus algoritmos de treinamento que visam a minimização do risco empírico (note que neste trabalho será enfatizado a minimização do risco estrutural) e o teorema da aproximação universal. O método proposto superou métodos padrões tanto em erro alcançado como em custo computacional, sendo este o artigo seminal desta tese.

Em [VCV04], item 5, o método proposto em [CVV03] é aplicado à extração de sensibilidade de um dispositivo eletromagnético, mais precisamente um alto-falante. Os resultados apresentados neste artigo mostraram a eficiência da topologia proposta para a resolução deste tipo de problema.

Em [VVCV05], item 4, a extração de sensibilidade apresentada em [VCV04] é utilizada como operador adicional em um algoritmo de otimização, melhorando significativamente o desempenho do mesmo. Em [VVC04], item 11, é mostrado um comparativo entre diversas redes para problemas eletromagnéticos, sendo que a rede proposta foi superior às demais na maioria dos

casos. Uma discussão sobre o uso de técnicas de otimização multiobjetivo para o treinamento de redes é apresentada em [VCV06], item 3. Entretanto, o leitor irá notar que este texto terá como ênfase os novos resultados obtidos, especialmente a técnica de treinamento de mínimo gradiente como apresentado em [VTP+05], item 2.

Em cooperação com Ecole Centrale de Lyon (França) e The University of Akron (USA) a rede PLP desenvolvida nesta tese foi utilizada na detecção de falhas em estruturas de concreto, sendo que dois artigos em congressos foram aceitos e um está submetido para revista.

Em cooperação com Imperial College of London (Reino Unido) as idéias propostas nesta tese foram utilizadas na segmentação de um procedimento cirurgico que é o bloco fundamental para se criar uma máquina capaz de prever erros neste tipo de procedimento.

Em conjunto com a Universidade de Oxford (Reino Unido) a rede PLP está sendo aplicada na definição da estrutura de proteinas. Parte da teoria aprensentada nesta tese foi desenvolvida em meu periodo de doutoramento sanduíche em Oxford.

Os outros artigos apresentam aplicações das técnicas propostas neste trabalho em diversos problemas.

Durante o desenvolvimento desta tese, entre Outubro de 2003 e Dezembro 2006, também foram publicados pelo autor desta tese artigos na área de otimização deterministica e estocástica. Destes cinco artigos foram publicados em revistas e doze em congressos.

Capítulo 2

Teoria do aprendizado de máquinas

Neste capítulo serão discutidos alguns pontos importantes para a fundamentação deste trabalho. Para referências gerais sobre o assunto de aprendizado de máquina consulte, por exemplo, [Vap98] [CST00] [Vap01] [HTF01]. O objetivo deste capítulo é prover uma visão do problema e das condições construtivas para se realizar o aprendizado de forma eficiente. Desta forma os detalhes técnicos, em geral, não serão tratados, sendo que os principais resultados apresentados servirão como base para a discussão sobre algoritmos de aprendizado de máquina. A maior parte do material apresentado neste capítulo se baseia em [Vap98] e [Vap01].

De fato, em um problema de aprendizado, o objetivo primal é encontrar a função desejada em um vasto conjunto de funções, sendo que esta busca é realizada tendo como base um número limitado de exemplos.

Na Figura 2.1 é mostrado um gráfico com uma base de dados (conjunto de treinamento) e duas possíveis aproximações que podem ser geradas a partir destes. É importante lembrar que as amostras que compõem a base de dados podem estar contaminadas com alguma forma de ruído, sendo este o caso prático mais geral. O modelo em linha contínua representa um modelo simples, linear, que apresenta algum erro em relação aos dados conhecidos. Em contrapartida, o modelo representado pela linha tracejada aproxima, sem erro, os dados conhecidos, mas utiliza uma função muito mais complexa para tal. Uma pergunta fica no ar, qual das soluções é a mais adequada para representar o problema em questão?

Esta pergunta tem sido tema central nas pesquisas relacionadas ao apren-

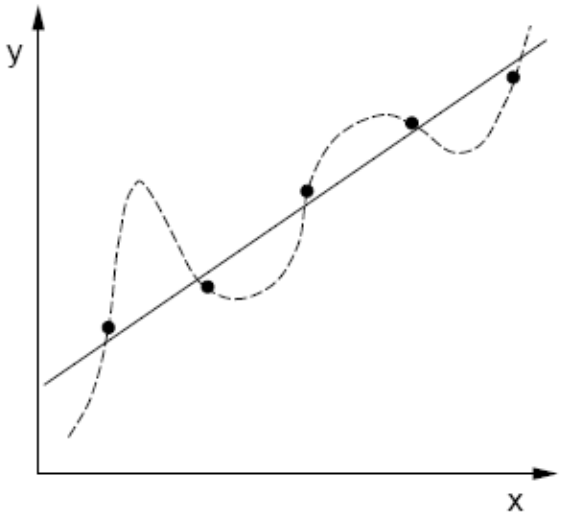

Figura 2.1: Figura mostrando possíveis soluções para o problema de regressão tendo os pontos como o conjunto conhecido. Nota-se que o modelo em linha contínua apresenta alguns erros na aproximação mas em contrapartida é um modelo mais simples. O modelo em linha tracejada representa um modelo com mais complexidade e que ajusta-se aos dados precisamente.

dizado de máquinas nos últimos anos. Uma das abordagens utilizadas para entender este problema, baseado na estatística clássica, é o estudo do dilema da polarização e da variância [GBD92]. Caso sempre fossem utilizados modelos lineares para representar uma distribuição de dados, uma polarização seria inserida no modelo. Caso fosse optado por interpolar os dados, i.e., gerar modelos com zero de resíduo, modelos com muita flutuação poderiam ser gerados, os quais poderiam não representar a resposta desejada. Nesta situação o modelo sofreria de grande variância.

Na comunidade de redes neurais, os modelos complexos demais para os dados são chamados de super-ajustados (overfitted), e os modelos que são simples demais para modelar o fenômeno estudado são chamados de sub-ajustados (underfitted). Na próxima seção será apresentado um modelo estatístico para o problema de aprendizado de máquinas.

2.1 O problema de aprendizado de máquinas

Pode-se pensar no problema de aprendizado de máquinas supervisonado utilizando três componentes básicas [Vap01]:

- Um gerador de vetores aleatórios x, amostrados independentemente de uma distribuição de probabilidade (acumulada) fixa, mas desconhecida, $F(x)$[1];
- um supervisor que retorna um vetor de saída yd para uma dada entrada x, de acordo com a função de distribuição condicional $F(yd|x)$ (este é o caso geral que inclui o caso $yd = f(x)$), também fixa mas desconhecida (a existência do supervisor define o aprendizado como supervisionado);
- e uma máquina capaz de realizar um conjunto de funções $\{f(x, w), w \in \Lambda\}$. O espaço Λ é o espaço onde as funções realizáveis pela máquina de aprendizagem se encontram, o espaço de hipóteses. Este pode ser em alguns casos composto por funções ou por parâmetros destas. Por exemplo w pode ser os coeficientes de um polinômio.

Pode-se definir o problema de aprendizado de máquinas como o problema de escolher em um dado conjunto de funções, $\{f(x, w), w \in \Lambda\}$ a função que consegue acertar de forma mais precisa, seguindo um critério pré-estabelecido, a resposta do supervisor.

2.2 A minimização do risco

Para realizar a escolha da melhor aproximação para a resposta do supervisor é necessário adotar uma medida de perda ou discrepância, $L(yd, f(x, w))$, entre a resposta yd do supervisor para uma dada entrada x e a resposta $f(x, w)$ gerada pela máquina de aprendizagem. O valor esperado da perda pode ser dado pelo funcional de risco (ou erro)[2] [Vap01][3]:

$$R(w) = \int L(yd, f(x, w))dF(x, yd), \tag{2.1}$$

[1] A função de densidade de probabilidade $p(x)$ é a derivada de $F(x)$, i.e., $\frac{dF(x)}{dx}$.

[2] $R(w)$ é o valor esperado do erro de teste.

[3] O risco $R(w) = \int L(yd, f(x, w))dF(x, yd)$ pode ser escrito de forma equivalente utilizando a densidade de probabilidade $p(x, yd)$, $R(w) = \int L(yd, f(x, w))p(x, yd)dx\ dyd$

onde $L(yd, f(x,w))$ é a função de perda ou discrepância.

Desta forma o objetivo do problema de aprendizagem de máquina pode ser descrito como encontrar a função $f(x,w_0)$ que minimiza o funcional de risco apresentado na Equação 2.1 sobre a classe de funções $\{f(x,w),\ w \in \Lambda\}$. É importante ter em mente que a distribuição $F(x,yd)$ é desconhecida e que a única informação disponível sobre o problema são as amostras do conjunto de treinamento. Desta forma não é possível minimizar a integral, definida na Equação 2.1, diretamente. O conjunto de treinamento é formado por T amostras aleatórias independentes e identicamente distribuídas (iid). Independência significa que cada observação traz o máximo de informação e a condição de identicamente distribuídas diz que estas são referentes a um mesmo fenômeno, observadas de acordo com $F(x,yd) = F(x)F(yd|x)$ [4], onde:

$$S_T = \{(x_1, yd_1), ..., (x_T, yd_T)\}, \tag{2.2}$$

é o conjunto de treinamento.

Os problemas de regressão e classificação são particularidades do problema mais geral descrito na Equação 2.1.

Pode-se então definir o problema de regressão como: *Considere a resposta do supervisor yd como um valor real e considere $\{f(x,w),\ w \in \Lambda\}$, como um conjunto de funções reais que contém a função de regressão desejada $f(x,w_0)$*[5]*, logo*[6]*:*

$$f(x,w_0) = \int yd\ dF(yd|x). \tag{2.3}$$

Caso $f(x,w) \in L_2$[7], a função de regressão pode ser a que minimiza o funcional descrito na Equação 2.1 com a seguinte função de perda:

$$L(yd, f(x,w)) = (yd - f(x,w))^2. \tag{2.4}$$

[4]
- $F(x)$ é a Função Distribuição de probabilidade
- $F(yd|x)$ é a Função de Distribuição condicional de yd dado x
- $F(y,x)$ é a Função de Distribuição Conjunta de yd e x

[5] $f(x,w_0)$ é a função que minimiza o risco esperado, $R(w)$, i.e., $R(w_0)$ é mínimo.

[6] A função $f(x,w_0) = \int yd\ dF(yd|x)$ equivale à $f(x,w_0) = \int ydp(yd|x)\ dyd$, que é a esperança conditional de yd dado x, $\mathbf{E}[yd|x]$.

[7] Uma função é pertencente ao L_2 se $\int \|f(r)\|^2 dr < +\infty$.

Desta forma o problema de regressão pode ser entendido como o problema de minimizar o funcional de risco, Equação 2.1, com a função de perda apresentada na Equação 2.4, na situação na qual a medida de probabilidade $F(x, yd)$ é desconhecida e amostras, Equação 2.2, são dadas. Definindo a função desejada, a função que representa a resposta do supervisor, como $f_0(x)$, se esta não pertence ao conjunto de funções $\{f(x, w),\ w \in \Lambda\}$, minimizar o funcional de risco equivale a encontrar a função mais próxima no espaço L_2. De fato pode-se para este caso escrever o risco:

$$\begin{aligned} R(w) &= \int (yd - f(x, w))^2 dF(x, yd) \\ &= \int (f_0(x) - f(x, w))^2 dF(x, yd) + \int (yd - f_0(x))^2 dF(x, yd) \end{aligned} \quad (2.5)$$

Onde o último termo não depende da função aproximada, e pode ser entendido como, por exemplo, um ruído existente nas medidas. Detalhes sobre a derivação da Equação 2.5 podem ser encontrados em [Vap98] e [Vap01].

Pode-se definir o problema de classificação como: *Considere que a resposta do supervisor yd possa ter somente dois valores, $yd = \{0, 1\}$, e considere $\{f(x, w), w \in \Lambda\}$ como um conjunto de funções indicadoras (funções que podem representar somente os valores zero e um).* Para este caso pode-se definir a seguinte função de perda:

$$L(yd, f(x, w)) = \begin{cases} 0 & se\ yd = f(x, w) \\ 1 & se\ yd \neq f(x, w) \end{cases} \quad (2.6)$$

Para esta função de perda, o funcional de risco mostrado na Equação 2.1 representa a probabilidade de classificação incorreta. Desta forma pode-se ver o problema de aprendizado como o problema de encontrar uma função que minimiza a probabilidade de erro de classificação quando $F(x, yd)$ é desconhecido mas um conjunto de treinamento é dado.

Os problemas de aprendizado supra-citados podem ser descritos de uma maneira geral. Considere o conjunto de funções $\{\vartheta(z, w), w \in \Lambda\}$ e a medida de probabilidade $F(z)$ definida no espaço Z. Desta forma o objetivo é minimizar o funcional de risco [Vap01]:

$$R(w) = \int \vartheta(z, w) dF(z),\ w \in \Lambda. \quad (2.7)$$

No problema definido na Equação 2.7 a medida de probabilidade $F(z)$ é desconhecida mas um conjunto *iid* de amostras,

$$S_T = \{z_1, ..., z_T\} \tag{2.8}$$

é dado. Desta forma pode-se entender que o funcional de risco mostrado na Equação 2.7 mede o erro de generalização e é uma variável aleatória dependente da seleção, também aleatória das amostras de treinamento.

2.3 Minimização do risco empírico

Como a medida de probabilidade $F(z)$ é desconhecida, o funcional do risco esperado $R(w)$ mostrado na Equação 2.7 não pode ser diretamente integrado, sendo este substituído pelo risco empírico [8]

$$R_{emp}(w, S_T) = \frac{1}{T}\sum_{t=1}^{T} \vartheta(z_t, w). \tag{2.9}$$

O funcional de risco empírico é construído utilizando o conjunto de treinamento mostrado na Equação 2.8. A idéia é aproximar a função $\vartheta(z, w_0)$ que minimiza o risco apresentado na Equação 2.7, pela função $\vartheta(z, w_T)$ que minimiza o risco empírico apresentado na Equação 2.9. Este princípio é denominado como princípio indutivo da minimização do risco empírico (ERM - Empirical Risk Minimization).

O método de minimização dos mínimos quadrados para problemas de regressão é uma conseqüência do princípio da minimização do risco empírico. De fato, para definir o problema de regressão é introduzida uma variável $n+1$ dimensional, $z = (x, yd) = (x^1, ..., x^n, yd)$, e a função de perda mostrada na Equação 2.4 é utilizada. Utilizando esta função de perda na Equação 2.9 tem-se:

$$R_{emp}(w, S_T) = \frac{1}{T}\sum_{t=1}^{T} (yd_t - f(x_t, w))^2. \tag{2.10}$$

Uma pergunta então aparece, "Quais são as condições de consistência para o princípio de minimização do risco empírico?".

Para responder tal pergunta é necessário mostrar as condições para convergência em probabilidade dos funcionais envolvidos. A convergência em

[8] O R_{emp} é o erro de treinamento.

probabilidade dos valores $R(w, S_T)$ significa que para qualquer $\epsilon > 0$ e qualquer $\delta > 0$, existe um número $T_0 = T_0(\epsilon, \delta)$, de tal forma que para qualquer $T > T_0$, com probabilidade de pelo menos $1 - \delta$, a desigualdade, $R(w, S_T) - R(w_0) < \epsilon$, é verdadeira.

Para responder a questão acima é preciso definir as condições necessárias e suficientes para a convergência em probabilidade das seguintes seqüências de valores aleatórios.

- Os valores dos riscos $R(w_T)$, o esperado risco para um dado conjunto de treinamento S de tamanho T, convergem para o menor valor possível do risco esperado $R(w_0)$

$$\lim_{T \to \infty} P\{R(w, S_T) - \inf R(w_0) > \epsilon\} = 0, \forall \epsilon > 0. \tag{2.11}$$

  onde $T \to \infty$ significa que o número de amostras do conjunto de treinamento tende a infinito.

- E os valores obtidos para o risco empírico $R_{emp}(w, S_T)$, $T = 1, 2, ...$ convergem para o mínimo valor possível de risco $R(w_0)$

$$\lim_{T \to \infty} P\{R_{emp}(w, S_T) - \inf R(w_0) > \epsilon\} = 0, \forall \epsilon > 0. \tag{2.12}$$

A Equação 2.11 mostra que as soluções encontradas utilizando a minimização do risco empírico converge para a melhor possível. A Equação 2.12 mostra que os valores do risco empírico convergem para os menores riscos.

2.4 Teoria da consistência do processo de aprendizado

Um ponto que deve ser considerado é que a existência de um espaço de hipóteses, $\{f(x, w), w \in \Lambda\}$, faz com que a condição de consistência mostrada na Equação 2.12 deva ser reformulada. As condições mostradas nas Equações 2.11 e 2.12 são de natureza estatística, i.e., elas dizem que a probabilidade de um desvio grande entre o erro de teste e de treinamento para função f é pequeno[9], quanto maior o número de amostras. Entretanto estas não tratam

[9]Pode-se definir o erro de treinamento como o erro que a máquina comete para um conjunto de dados conhecidos, i.e., os dados para os quais esta foi treinada. O erro de teste é para um conjunto desconhecido.

o caso no qual o desvio é grande, e de fato a máquina de aprendizagem pode implementar diversas funções com esta característica. Desta maneira precisa-se da teoria de consistência para o processo de aprendizado. Esta teoria traz as condições necessárias e suficientes para a convergência do risco empírico, quando um espaço de hipóteses é dado. Embora seja uma teoria assintótica, qualquer teoria que envolva o princípio de minimização do risco empírico deve satisfazer estas condições necessárias e suficientes para a convergência. Sem a restrição do espaço de funções, a minimização do risco empírico não é consistente. De fato, o pior caso entre todas as funções que a máquina de aprendizagem pode implementar que define a consistência da minimização do risco empírico.

Considere o seguinte teorema [Vap01]: *Considere* $\{\vartheta(z,w), w \in \Lambda\}$ *como um conjunto de funções que têm, para uma medida de probabilidade* $F(z)$*, o funcional de risco limitado da seguinte forma:*

$$A \leq \int \vartheta(z,w) dF(z) \leq B; \ \forall w \in \Lambda. \tag{2.13}$$

Para que o princípio de minimização do risco empírico seja consistente é necessário e suficiente que o risco empírico, $R_{emp}(w, S_T)$, convirja uniformemente para o risco esperado, $R(w)$, dentro do conjunto $\{\vartheta(z,w), w \in \Lambda\}$. Matematicamente:

$$\lim_{T\to\infty} P\{ \sup_{w\in\Lambda} (R(w) - R_{emp}(w, S_T)) > \epsilon\} = 0, \ \forall \epsilon > 0. \tag{2.14}$$

Este teorema mostra que a análise das propriedades de convergência do princípio de minimização do risco empírico deve ser uma análise de pior caso. Nota-se que a convergência depende do maior desvio entre os riscos.

Para descrever as condições necessárias e suficientes para a convergência uniforme da Equação 2.14, é necessário definir o conceito de entropia do conjunto de funções, $\{\vartheta(z,w), w \in \Lambda\}$, com um conjunto de treinamento de tamanho T.

Considere um conjunto de funções indicadoras $\{\vartheta(z,w), w \in \Lambda\}$, i.e., funções que podem indicar somente zero ou um, considerando um conjunto de treinamento como descrito na Equação 2.8. Pode-se caracterizar a diversidade deste conjunto de funções pela quantidade $N(\Lambda, S_T)$ que representa o número máximo de diferentes separações destas amostras que podem ser obtidas usando funções do conjunto indicado. No caso de reconhecimento de

padrões pode-se interpretar esta medida como a quantidade de maneiras que uma classe de funções pode separar um conjunto de padrões em diferentes classes (duas no caso estudado).

Pode-se definir $N(\Lambda, T)$ como o máximo de $N(\Lambda, S_T)$, para todos o possíveis conjuntos S_T de tamanho T. Quando $N(\Lambda, T) = 2^T$, todas as separações possíveis podem ser implementadas pelas funções da classe. Neste caso, a classe de funções é dita como capaz de dividir T amostras. Observe que isto significa que existe um conjunto de T amostras que podem ser divididas em todas as maneiras possíveis. Isto não significa que é aplicável para todos os conjuntos S_T com T amostras.

Defina-se a entropia aleatória como:

$$H1(\Lambda, S_T) = ln\ N(\Lambda, S_T). \tag{2.15}$$

A entropia aleatória descreve a diversidade de um conjunto de funções para um dado conjunto S_T. Observe que $H1(\Lambda, S_T)$ é aleatória devido ao fato que o conjunto S é construído utilizando dados aleatórios. Considerando então a esperança matemática[10], $\mathbf{E}[.]$, da entropia aleatória tem-se:

$$H(\Lambda, T) = \mathbf{E}[ln\ N(\Lambda, S_T)]. \tag{2.16}$$

Esta quantidade é a entropia do conjunto de funções indicadoras $\{\vartheta(z, w), w \in \Lambda\}$ para conjuntos de tamanho T, onde a esperança é calculada utilizando amostragens aleatória do conjunto S_T de uma dada distribuição F. Observe que esta é dependente do conjunto de funções indicadoras, da medida de probabilidade $F(z)$ e do número de amostras T. Desta forma a entropia descreve a diversidade esperada de um conjunto de funções indicadoras quando o conjunto de amostragem é de tamanho T.

Em [VC71] foi mostrado para funções de perda indicadoras (relativas a problemas de classificação de padrões) o seguinte teorema: *Para convergência uniforme das freqüências e suas probabilidades*

$$\lim_{T\to\infty} P\{ \sup_{w\in\Lambda} |(R(w) - R_{emp}(w, S_T))| > \epsilon\} = 0,\ \forall \epsilon > 0 \tag{2.17}$$

é necessário e suficiente que a seguinte condição seja válida:

$$\lim_{T\to\infty} \frac{H(\Lambda, T)}{T} = 0,\ \forall\ \epsilon > 0. \tag{2.18}$$

[10] $\mathbf{E}[x] = \int xp(x)dx = \int xdF(x)$

Esta condição, Equação 2.18, é necessária e suficiente para a consistência do princípio da minimização do risco empírico. Todas as máquinas que minimizam o risco empírico devem satisfazê-la. Embora demonstrado que o princípio da minimização do risco empírico é consistente, é necessário mostrar que a taxa de convergência assintótica do erro não é lenta demais.

Permutando a esperança $\mathbf{E}[.]$ e o logaritmo na Equação (2.16) obtém-se a *entropia recozida*. Matematicamente:

$$H^{ann}(\Lambda, T) = ln\ \mathbf{E}[N(\Lambda, S_T)]. \tag{2.19}$$

Como a função logaritmica é côncava, a entropia recozida é um limitante superior da entropia aleatória, Equação 2.16[11]. Desta maneira se a entropia recozida satisfaz uma condição similar a mostrada na Equação 2.18, a mesma condição é valida para a entropia aleatória. Pode ser mostrado que a convergência

$$\lim_{T\to\infty} \frac{H^{ann}(\Lambda, T)}{T} = 0, \tag{2.20}$$

implica em uma convergência exponencial e rápida da forma [Vap01]

$$P\{\sup_{w\in\Lambda} |(R(w) - R_{emp}(w, S_T))| > \epsilon\} \leq e^{-c\epsilon^2 T},\ \forall\epsilon > 0. \tag{2.21}$$

Sendo que esta condição é necessária e suficiente com $c > 0$.

Para demonstrar que o princípio de minimização do risco empírico é consistente e converge rapidamente, independentemente da medida de probabilidade, considere a seguinte função de crescimento:

$$G(\Lambda, T) \equiv ln \sup_{z_1,\ldots,z_T} N(\Lambda, S_T). \tag{2.22}$$

A condição necessária e suficiente para a convergência é dada pela Equação 2.23.

$$\lim_{T\to\infty} \frac{G(\Lambda, T)}{T} = 0, \tag{2.23}$$

[11] A desigualdade de Jensen é normalmente escrita em termos de uma função convexa f, e é dada por $f(\mathbf{E}[x]) \leq \mathbf{E}[f(x)]$. No caso apresentado na Eq. 2.19, f é uma função concava, logo a desigualdade de Jensen pode ser escrita como $\mathbf{E}[f(x)] \leq f(\mathbf{E}[x])$

Note que a definição de função de crescimento equivale ao logaritmo da função $N(\Lambda, T)$, i.e., $G(\Lambda, T) = ln\ N(\Lambda, T)$. Observa-se que as seguintes desigualdades para qualquer T é válida

$$H(\Lambda, T) \leq H^{ann}(\Lambda, T) \leq G(\Lambda, T). \tag{2.24}$$

O fato que os limites do erro sejam válidos para qualquer distribuição a faz inevitavelmente um limite pessimista, pois algumas distribuições serão mais difíceis de aprender que outras, e os limites têm que ser válidos para todas distribuições. É importante observar que as análises feitas até o momento são mais conceituais que construtivas, pois a teoria não prevê maneiras de avaliar as quantidades utilizadas. Na próxima seção será apresentado o conceito de dimensão VC, que é um limitante superior da função de crescimento, e pode ser utilizada de forma construtiva.

2.4.1 A dimensão VC (Vapnik and Chervonenkis dimension)

Como será visto a função de crescimento, $G(\Lambda, T)$, tem diversas propriedades interessantes. Se o conjunto de funções $\{\vartheta(z, w), w \in \Lambda\}$ é rico o suficiente, logo, para qualquer conjunto de amostras de tamanho T, as amostras podem ser escolhidos de tal forma que eles possam ser separados em todas as 2^T maneiras possíveis. Assim,

$$G(\Lambda, T) = T\ ln\ (2). \tag{2.25}$$

Pode-se observar que neste caso a convergência da Equação 2.23 não ocorre, e o treinamento não será, em geral, bem sucedido. O próximo passo será resumir o comportamento da função de crescimento, indicada na Equação 2.22, por um número. Este número é conhecido como a dimensão VC *(Vapnik and Chervonenkis dimension)* e será denotado como h. Por construção, a dimensão VC é o máximo número de amostras que podem ser separados pelas funções existentes em Λ. A função de crescimento, para $T > h$ é limitada por [Vap01]

$$G(\Lambda, T) \leq\ h\left(ln\ \frac{T}{h}\ + 1\right). \tag{2.26}$$

Desta forma, pode-se ver que a função de crescimento será linear ou limitada por uma função logaritmica, sendo a segunda situação o regime propício para

o aprendizado. Os conceitos mostrados, que podem ser entendidos como a capacidade ou complexidade de um conjunto de funções, podem ser ordenados como:

$$H(\Lambda, T) \leq H^{ann}(\Lambda, T) \leq G(\Lambda, T) \leq h\left(ln\ \frac{T}{h}\ + 1\right). \tag{2.27}$$

Da Equação pode-se observar que da esquerda para a direita as capacidades ficam menos precisas. Entretanto as entropias da esquerda são dependentes da distribuição $F(z)$, e a função de crescimento e a dimensão VC não o são. Considerando o fato que as distribuições não são conhecidas a priori, caso fossem não seria necessário uma máquina para aprendê-las, torna as definições à direita de 2.27 as utilizáveis.

A finitude da dimensão VC do conjunto de funções indicadoras $\{\vartheta(z, w), w \in \Lambda\}$, implementadas pela máquina de aprendizagem é condição suficiente para a consistência do princípio de minimização do risco empírico, e também leva a uma convergência rápida.

Definições equivalentes para funções reais (contínuas)

Na seção anterior foram definidos diversos conceitos de capacidade para funções indicadoras (discretas). Nesta seção será mostrado como obter definições equivalentes para funções reais.

Considere $\{A \leq \vartheta(z, w) \leq B, w \in \Lambda\}$, um conjunto de funções reais limitadas pelas constantes A e B. Das funções reais do conjunto pode-se gerar um conjunto de funções indicadoras da seguinte forma:

$$I(z, w, \varrho) = \theta(\vartheta(z, w) - \varrho),\ w \in \Lambda \tag{2.28}$$

onde $A < \varrho < B$ é uma constante e $\theta(u)$ é a função degrau, i.e., indica 0 se a função for negativa e $+1$ caso contrário. A dimensão VC do conjunto de funções reais $\{\vartheta(z, w), w \in \Lambda\}$, é definida como a dimensão VC do conjunto de funções indicadoras mostradas na Equação 2.28.

2.4.2 Exemplos da dimensão VC

Um dos exemplos mais importantes para exemplificar a dimensão VC é o conjunto de funções indicadoras lineares

$$f(x, w) = \theta(\sum_{i=1}^{n} w_i x_i + w_0) \tag{2.29}$$

no espaço n-dimensional $X = (x_1, ..., x_n)$. Para esta classe de funções a dimensão VC é igual a $n+1$. Desta forma, utilizando funções deste conjunto só poderão ser separados $n+1$ vetores. Observe que a dimensão VC significa que existe um conjunto com T amostras que pode ser divido, não que todos os conjuntos possam ser. Por exemplo, dados três pontos não colineares em $\mathbb{R}^2$, independentemente de que classe estes pertençam, existem parâmetros w que conseguem separar as classes. O mesmo não ocorre se as amostras forem colineares.

Para o caso que considera o conjunto de funções lineares reais a derivação da dimensão VC é equivalente a mostrada acima. Basta transformar a função real em uma função indicadora, utilizando a Equação 2.28. O resultado não é alterado devido ao fato que pode-se utilizar $w_0 - \beta$ ao invés de w_0.

Considere o hiperplano:

$$(w^* x) - b = 0, \ |w^*| = 1. \tag{2.30}$$

Este hiperplano é chamado de hiperplano classificador com margem Δ se este classifica vetores x da seguinte maneira:

$$y = \begin{cases} 1, & se\ (w^* x) - b \geq \Delta \\ -1, & se\ (w^* x) - b \leq -\Delta \end{cases} \tag{2.31}$$

Observem que classificações de vetores x que estão dentro da margem $(-\Delta, \Delta)$ são indefinidas. Pode-se provar que para um conjunto de vetores $x \in X$ pertencentes a uma esfera de raio R, o conjunto dos hiperplanos com margem Δ tem a dimensão VC (h) limitada pela desigualdade:

$$h \leq min\ \left(\frac{R^2}{\Delta^2}, n \right) + 1. \tag{2.32}$$

Estes exemplos mostram que a dimensão VC de um hiperplano é no máximo $n+1$, e esta pode ser reduzida caso a margem de separação Δ exista. Este fato é utilizado para a construção das SVMs (Support Vector Machines), onde a complexidade é controlada controlando a margem de separação. Este resultado é exemplificado para um caso bi-dimensional como mostrado na Figura 2.2. Pode-se entender a margem de um classificador como uma medida da dificuldade imposta pela distribuição.

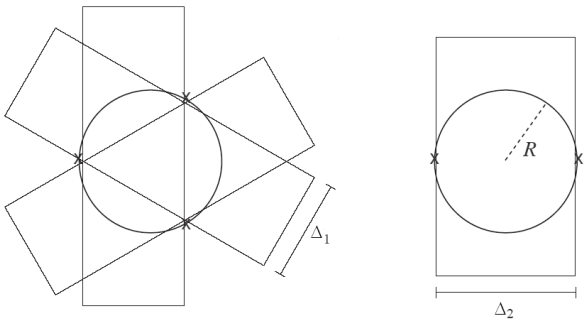

Figura 2.2: Nesta figura observa-se duas situações distintas onde os círculos representam a hiper-esfera de raio R que contém as amostras $x \in X$. Na primeira situação é apresentado hiperplanos com margem Δ_1, e pode-se observar que estes podem classificar até três amostras distintas. Na segunda é mostrado um hiperplano com margem Δ_2, sendo que este só pode distinguir duas amostras distintas. Desta forma é claro que a dimensão VC é dependente da margem de separação.

2.5 Limites na taxa de convergência do processo de aprendizado

Utilizando os conceitos de capacidade definidos anteriormente serão mostrados os limites das taxas de convergência do processo de aprendizado. Estes limites servem para estimar a capacidade de generalização da máquina de aprendizagem. É importante notar que estes são limites superiores do risco esperado, logo não correspondem ao risco real. As análises apresentadas nesta seção serão desenvolvidas para problemas de classificação, sendo que a extensão destas para problemas de regressão seguem a idéia de utilizar a função apresentada na Equação 2.28. Deve ser observado que os limites definidos aqui utilizam um conjunto de treinamento finito, diferentemente dos resultados das equações 2.11 e 2.12 onde $T \rightarrow \infty$.

Os limites definidos nesta seção têm a seguinte forma, $\zeta = \zeta(T, \Lambda, \delta)$, i.e., o erro é função do número de amostras utilizadas (T), do espaço de hipóteses (Λ) e de um parâmetro adicional, (δ), que garante que com a probabilidade de pelo menos $1-\delta$ no conjunto de treinamento aleatório S_T, o erro de generalização $(err(F, w))$, onde $F = F(z)$, da hipótese selecionada, $\{f(x, w), w \in \Lambda\}$,

será limitado por

$$err(F, w) \leq \zeta(T, \Lambda, \delta), \tag{2.33}$$

que significa que é provavelmente aproximadamente correto (PAC- Probably Approximately Correct). Esta afirmativa é equivalente a dizer que a probabilidade de se selecionar hipóteses com valores altos de erro é pequena[12]:

$$P\{S_T : err(F, w) > \zeta(T, \Lambda, \delta)\} < \delta \tag{2.34}$$

Considere o espaço de hipóteses, $\{f(x, w), w \in \Lambda\}$, tendo a dimensão VC igual a h. Para qualquer distribuição de probabilidade F em $X \times \{-1, 1\}$, com a probabilidade $1 - \delta$, qualquer hipótese $w \in \Lambda$ consistente com S tem o erro limitado por

$$err(F, w) \leq \zeta(T, \Lambda, \delta) = \frac{2}{T}\left(h \ln \frac{2eT}{h} + \ln \frac{2}{\delta}\right), \tag{2.35}$$

onde $h \leq T$, $T > 2/\zeta$. Uma hipótese é considerada consistente caso não apresente erro no conjunto de treinamento.

O resultado mostrado na Equação 2.35 é muito interessante e mostra a importância da dimensão VC para a generalização. Entretanto os dados de treinamento podem estar, por exemplo, corrompidos por ruído, e desta forma não é recomendável buscar por hipóteses que sejam consistentes com o conjunto de treinamento.

Considere o espaço de hipóteses, $f(x, w)$, $w \in \Lambda$, tendo a dimensão VC igual a h. Para qualquer distribuição de probabilidade F em $X \times \{-1, 1\}$, com a probabilidade $1-\delta$ no conjunto de treinamento S_T , qualquer hipótese $w \in \Lambda$ que cometa k erros no conjunto de treinamento S tem o erro de generalização limitado por

$$err(F, w) \leq \zeta(T, \Lambda, \delta) = \frac{2k}{T} + \frac{4}{T}\left(h \ln \frac{2eT}{h} + \ln \frac{4}{\delta}\right), \tag{2.36}$$

onde $h \leq T$.

[12] A probabilidade, dado o conjunto S tal que $err(.) > \zeta(.)$, é menor que um δ.

2.5.1 Limites de generalização baseados na margem

Foi mostrado anteriormente que em um problema que utiliza funções indicadoras lineares a dimensão VC depende da margem de separação entre as classes. Nesta seção será generalizada a definição de margem para uma classe arbitrária de funções reais.

Considere a classe $\mathcal{F}$ de funções reais no espaço de entrada X para classificação com limiar em 0. Pode-se definir a margem de um exemplo $(x_t, yd_t) \in X \times \{-1, 1\}$ em relação a uma função $f \in \mathcal{F}$ como a quantidade:

$$\gamma_t = yd_t f(x_t). \tag{2.37}$$

Observe que $\gamma_t > 0$ implica na classificação correta do exemplo. A distribuição de margens de f em relação ao conjunto de treinamento S pode ser definida como:

$$M(S_T, f) = \{\gamma_t = yd_t f(x_t) : t = 1, ..., T\}. \tag{2.38}$$

Na formulação de alguns problemas de aprendizado de máquinas, como no caso das SVMs, a maximização da menor margem, que é definida como

$$m(S, f) = min\ M(S_T, f), \tag{2.39}$$

é utilizada.

Utilizando a idéia de margem a dimensão VC pode ser generalizada. Considere $\mathcal{F}$ um conjunto de funções reais. O conjunto de amostras X é γ-*separável* (γ-*shattered*) por $\mathcal{F}$ se existem números reais r_x indexados por $x \in X$ tal que para todos os vetores binários $b \in \{-1, 1\}^T$, existe uma função $f_b \in \mathcal{F}$ satisfazendo:

$$f_b(x_t) \begin{cases} \geq r_t + \gamma & se\ b_t = 1 \\ < r_t + \gamma & se\ b_t = -1 \end{cases} \tag{2.40}$$

A dimensão *fat-shattering*, $fat_{\mathcal{F}}(\gamma)$, do conjunto de funções $\mathcal{F}$, é o tamanho do maior conjunto que pode ser γ-*shattered*, dado um γ. A dimensão *fat-shattering* foi introduzida em [KS90].

Considere a classe de funções reais $\mathcal{F}$ que tenha a dimensão *fat-shattering* limitada por $fat_{\mathcal{F}}(\gamma)$, sendo utilizada como classificador com a variação $[-R, R]$ e um $\gamma \in \mathbb{R}^+$ fixo. Para qualquer distribuição de probabilidade F em $X \times \{-1, 1\}$, com a probabilidade $1 - \delta$ em T amostras aleatórias S_T,

qualquer hipótese $f \in \mathcal{F}$ que tenha margem $m(S_T, f) \geq \gamma$ em S_T tem o erro limitado por

$$err(F, f) \leq \zeta(T, \mathcal{F}, \delta, \gamma) = \frac{2}{T}\left(d \ln \frac{16elR}{d\gamma} \ln \frac{128TR^2}{\gamma^2} + \ln \frac{4}{\delta}\right), \quad (2.41)$$

onde $l > 2/\zeta$, $d < l$, $d = fat_{\mathcal{F}}(\gamma/8)$. A demonstração deste teorema pode ser encontrada em [STB98].

Observa-se que o resultado mostrado na Equação 2.41 é obtido sem considerar erros no conjunto de treinamento, $\gamma \in \mathbb{R}^+$. Como já discutido anteriormente, gerar hipóteses que são totalmente consistentes com o conjunto de treinamento pode ser não desejado, pois as amostras podem ser não separáveis ou pode haver ruído no conjunto de treinamento.

Em [Bar98] foi utilizada a idéia de percentual da margem. Esta medida tem uma vantagem significante pois inclui casos nos quais as hipóteses não são totalmente consistentes. Ordenando os valores da distribuição da margem, Equação 4.3, tal que $\gamma_1 \leq \gamma_2 \leq ... \leq \gamma_T$, e fixando um número $k < T$, o número k/T é a porcentagem $M_k(S_T, f)$ que $M(S_T, f)$ é γ_k.

Considere o conjunto de funções indicadoras lineares com o vetor de pesos unitários em um espaço e produto interno X e fixe $\gamma \in \mathbb{R}^+$. Existe uma constante c, de tal forma que qualquer distribuição de probabilidade F em $X \times \{-1, 1\}$ com o suporte na hiperesfera de raio R centrada na origem, com a probabilidade de $1 - \delta$ em T amostras aleatórias de S, qualquer hipótese pertencente a esta classe tem o erro limitado por

$$err(F, f) \leq \zeta(T, \mathcal{F}, \delta, \gamma) = \frac{k}{T} + \sqrt{\frac{c}{T}\left(\frac{R^2}{M_{S,k}(f)^2} \ln^2 T + \ln \frac{1}{\delta}\right)}, \quad (2.42)$$

para todo $k < T$. Este resultado sugere que pode-se obter boa generalização minimizando o número de amostras que estão classificados com margem menor que γ. De fato, neste resultado pode-se ignorar as amostras que tornam o problema não separável ou diminuem muito a margem. Este resultado foi substituido pelas margens suaves que serão mostradas na seção seguinte.

2.5.2 Limites com margens suaves (Soft margin)

Considerando uma margem alvo γ, e perguntando quanto cada amostra falha para obtê-la pode-se definir matematicamente as variáveis de folga de uma margem. Dada uma classe de funções reais $\mathcal{F}$ no espaço de entrada X para

classificação com limiar em 0, pode-se definir a variável de folga de uma margem de um exemplo $(x_t, yd_t) \in X \times \{-1, 1\}$ em relação à função $f \in \mathcal{F}$ e à margem alvo γ como a quantidade:

$$\xi_t((x_t, yd_t), f, \gamma) = \xi_t = max(0, \gamma - yd_t f(x_t)). \tag{2.43}$$

Observa-se que para amostras com margem superior a γ o valor é zero. Para amostras que foram classificados incorretamente tem-se que $\xi_t > \gamma$. Pode-se também definir o vetor das variáveis de folga para um dado conjunto de treinamento S como

$$\xi(S, f, \gamma) = \xi = (\xi_1, ..., \xi_T). \tag{2.44}$$

Em [STC02] foram derivados os limites de generalização considerando a variável de folga da margem definida na Equação 2.44. Considere $\mathcal{F}$ uma classe de funções reais com variação $[-a, a]$ e a dimensão *fat-shattering* limitada por $fat_{\mathcal{F}}(\gamma)$. Fixe uma escala da variação da saída como $\kappa \in \mathbb{R}^+$. Considere uma distribuição de probabilidade fixa mas desconhecida no espaço $X \times \{-1, 1\}$. Com a probabilidade de $1 - \delta$ em um conjunto amostrado de forma aleatória S de tamanho T, para todo $0 < \gamma \leq a$, a generalização do conjunto de funções indicadoras derivadas de $\mathcal{F}$ com o limiar em 0 é limitada por:

$$err(F, f) \leq \tilde{O}\left(\frac{fat(\gamma/16) + \|\xi\|_2^2/\gamma^2}{|S|}\right), \tag{2.45}$$

onde $\tilde{O}$ significa ordem de grandeza assintótica ignorando logaritmos.

2.6 Minimização do risco estrutural (SRM - Structural Risk Minimization)

Nas seções anteriores deste capítulo foram discutidos aspectos gerais do aprendizado de máquina. Foram mostradas as condições necessárias e suficientes para a consistência do princípio da indução da minimização do risco empírico assim como limites de generalização para máquinas que utilizam este princípio. Entretanto como ressaltado em [STC02], estes resultados devem ser considerados como uma indicação dos fatores que afetam a generalização, não uma estimativa realista do erro. Desta forma estes podem ser

utilizados como base na construção de algoritmos, pois como citado anteriormente, estes trazem indicações dos fatores que são importantes para uma boa generalização.

Em [Vap92] foi proposto um princípio para minimizar o funcional de risco para situações onde o conjunto de treinamento S é pequeno[13]. Observa-se que os limites do risco apresentados são função do risco empírico (erro de treinamento) e da dimensão VC (e sua variação com a dimensão *fat-shattering*). Para a situação supra-citada foi proposto o princípio indutivo da minimização do risco estrutural (SRM -Structural Risk Minimization), que considera a minimização do risco empírico e do risco estrutural. Por exemplo, observando a Equação 2.42, percebe-se que esta é composta por dois termos, sendo que a minimização de ambos é importante para a generalização. O princípio SRM define uma escolha com compromisso entre a qualidade da aproximação (definida em termos do risco empírico) e a complexidade da função aproximada (que pode ser definida como a dimensão VC, a dimensão *fat-shattering* entre outras).

Considere o conjunto $\mathcal{F}$ de funções $\{\vartheta(z,w),\ w \in \Lambda\}$, estruturado como um sub-conjunto aninhado de funções $\mathcal{F}_k = \{\vartheta(z,w),\ w \in \Lambda_k\}$, de tal forma que:

$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset ... \subset \mathcal{F}_n ... \ . \tag{2.46}$$

Os elementos da Equação 2.46 devem satisfazer às seguintes propriedades:

- A dimensão VC h_k de cada elemento do conjunto $\mathcal{F}_k$ é finito, e

$$h_1 < h_2 < ... < h_n ... \ . \tag{2.47}$$

- Qualquer elemento de $\mathcal{F}_k$ da estrutura tem as funções de perda positivas e limitadas.

Desta forma pode-se entender o método de minimização do risco estrutural como um problema multiobjetivo[14] (mais precisamente bi-objetivo) que tenta encontrar uma solução de compromisso entre os dois objetivos que em

[13] O conjunto de treinamento é considerado pequeno quando $T/h < 20$, i.e., o número de amostras do conjunto dividido pela dimensão VC é menor que 20.

[14] Para detalhes sobre otimização multiobjetivo consulte o apêndice A.

geral são conflitantes. Pode-se escrever o problema de minimização do risco estrutural de forma geral como:

$$min \left\{ \begin{array}{l} f_1 \\ f_2 \end{array} \right. \tag{2.48}$$

onde f_1 representa a minimização de algum funcional de risco, R_{emp}, e f_2 a minimização da complexidade da máquina, Ω, a dimensão VC por exemplo. Quando f_1 e f_2 são convexos as duas formulações são equivalentes, i.e., o problema multiobjetivo é uma seqüência ordenada. Entretanto, quando a convexidade não é garantida, a formulação multiobjetiva é mais completa. Normalmente, não é possível minimizar todos os objetivos simultaneamente, porque o ótimo de um dos objetivos raramente é o ótimo dos outros. Então, não existe um ótimo único, mas um conjunto deles, quando a formulação multiobjetivo é considerada. Para descrever melhor algumas definições são necessárias: i) *Dominância*: Um vetor w_1 domina w_2, $w_1 \prec w_2$, se $f_j(w_1) \leq f_j(w_2)$ $\forall j$, onde $j = 1, .., m$ e $f_j(w_1) \neq f_j(w_2)$ para pelo menos um j. ii) *Otimalidade de Pareto*: Um vetor w^* é denominado Pareto Ótimo (PO) se não existe nenhum outro vetor na região viável que o domina. Utilizando estas definições é possível gerar um conjunto de soluções, no espaço dos objetivos, chamado de Fronteira PO, a qual tem o melhor compromisso entre o erro e a complexidade da máquina.

Algumas informações referentes as Eq. (2.46) e (2.47) podem ser obtidas à luz da otimalidade de Pareto da Eq. (4.8). Na seqüência ordenada $\mathcal{F}_1 \subset \mathcal{F}_2 \subset ... \subset \mathcal{F}_n ...$, o erro empírico mínimo é um conjunto ordenado como $R_{emp}(\mathcal{F}_1) \geq R_{emp}(\mathcal{F}_2) \geq ... \geq R_{emp}(\mathcal{F}_n)...$, enquanto a complexidade é ordenada $h_1 < h_2 < ... < h_n ...$. Esta ordenação emplica que se $f_1(w_1) < f_1(w_2)$, $f_2(w_1) > f_2(w_2)$, então, w_1 não domina w_2. Em um caso mais geral, a situação onde $f_1(w_1) < f_1(w_2)$ e $f_2(w_1) < f_2(w_2)$ pode acontecer (w_2 é um modelo mais complexo e com maior erro de treinamento). Seguindo as definições de Otimalidade de Pareto e o problema definido na Eq. (4.8), a solução w_1 (mais simples e com menor error de treinamento) será escolhida.

O conjunto com todas as soluções PO da Eq. (4.8) formam um conjunto com os melhores compromissos entre R_{emp} e a complexidade Ω. Isto significa que qualquer melhora em R_{emp} implica em piora da complexidade e vice-versa. Neste sentido, este trabalho está buscando modelos PO, w^*. Um exemplo gráfico destes conceitos multiobjetivo estão mostrados na Fig. 2.3 e mais detalhes podem ser encontrados no Apêndice.

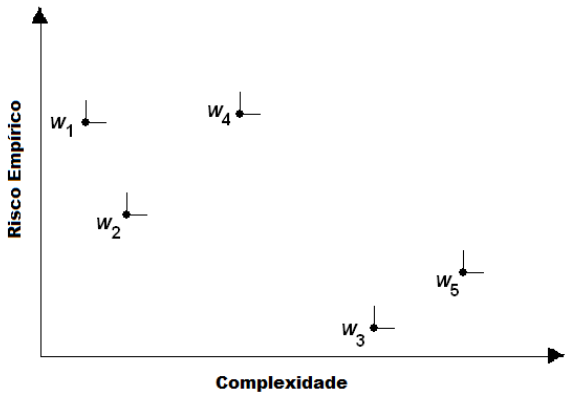

Figura 2.3: Exemplo ilustrando o conceito de Otimalidade de Pareto. No eixo x, está representada a complexidade Ω e em y o risco empírico R_{emp}. Pode-se notar que $\Omega(w_1) < \Omega(w_2)$ e $R_{emp}(w_1) > R_{emp}(w_2)$, logo a máquina de aprendizagem representada por w_1 não domina w_2, $w_1 \not\prec w_2$. Também é claro que $w_2 \not\prec w_1$, $w_1 \not\prec w_3$, $w_3 \not\prec w_1$, $w_2 \not\prec w_3$ e $w_3 \not\prec w_2$, i.e., não há relação de dominância entre w_1, w_2 *e* w_3. A máquina w_3 domina w_5, $w_3 \prec w_5$, e $w_2 \prec w_4$, logo não é interessante utilizar as máquinas w_4 e w_5, pois estas são piores nos dois atributos. A fronteira PO neste exemplo é $PO = \{w_1, w_2, w_3\}$, estas são as máquinas de interesse deste trabalho.

2.7 Problemas mal colocados (Teoria de regularização)

Um problema é dito como bem colocado se a solução da equação

- existe,
- é única, e
- é estável[15].

[15] A soluçao do problema

$$Af(t) = F(x), \tag{2.49}$$

é estável se uma pequena variação em $F(x)$ resulta em uma pequena variação na solução, $f(t)$

Um problema é dito mal colocado se a solução da equação viola pelo menos um dos requerimentos citados acima. O estudo de formas de resolução do problema mal colocado data do início da década de 60. Foram desenvolvidos três métodos para a resolução deste problema quando funcionais lineares do tipo

$$Af(t) = F(x), \tag{2.50}$$

são considerados. Todos os métodos são baseados em um funcional de regularização $\Omega(f)$. O funcional de regularização $\Omega(f)$ é semi-contínuo, positivo e compacto (no espaço da funções definidas por f) para a situação $\Omega(f) \leq c$, $c > 0$. Este é definido no espaço das funções $f \in \mathcal{F}$, o domínio das soluções da equação.

Para impor unicidade à solução é suficiente que $\Omega(f)$ tenha as seguintes propriedades [Tik63]:

- $\Omega(f)$ é um funcional não negativo convexo. Deve-se lembrar que a convexidade é definida para qualquer $0 \leq \lambda \leq 1$ como:

$$\Omega(\lambda f_a + (1-\lambda) f_b) < \lambda\Omega(f_a) + (1-\lambda)\Omega(f_b), \;\; f_a, f_b \in \mathcal{F}; \tag{2.51}$$

- $\Omega(0) = 0$ é verdadeiro,
- e para cada função f a função $r(\rho) = \Omega(\rho f)$ é uma função estritamente crescente de ρ.

Tendo como base este funcional de regularização três métodos foram propostos.

1. O método variacional de Tikhonov [Tik63] [TA77] que é descrito da seguinte forma:

$$min \; \|Af - F\| + \lambda\Omega(f), \tag{2.52}$$

   onde λ é uma constante pré definida.

2. O método residual de Phillips [Phi62] que é descrito da seguinte forma:

$$\begin{array}{ll} min & \Omega(f) \\ s.a. & \|Af - F\| \leq \lambda \end{array} \tag{2.53}$$

   onde $\lambda > 0$ é alguma constante pré-definida.

3. O método das quasi-soluções de Ivanov [Iva62] [Iva76] que pode ser descrito como:

$$\begin{array}{ll} min & \|Af - F\| \\ s.a. & \Omega(f) \leq \lambda \end{array} \tag{2.54}$$

onde $\lambda > 0$ é alguma constante pré-definida.

Os três métodos são equivalentes, i.e., estes são capazes de gerar as mesmas soluções, como mostrado em [Vas70]. De fato considerando a teoria de otimização multiobjetivo, dado que os funcionais envolvidos são convexos, os três métodos mostrados nas Eqs. 2.52, 2.53 e 2.54, são capazes de mapear a fronteira Pareto por completo (a fronteira PO é parametrizada em função de λ). Como também é sabido na teoria de otimização, os métodos ponderados (equivalente a formulação apresentada por Tikhonov), quando aplicáveis, têm melhor desempenho computacional, sendo este talvez o fato que levou a maior popularização do método descrito na Equação 2.52 se comparados aos outros métodos. De fato pode-se ver o problema de regularização como um problema multiobjetivo, similarmente ao problema de minimização de risco estrutural mostrado na Equação 4.8, onde as técnicas descritas acima são somente formas computacionais para resolver o problema multiobjetivo reescrevendo-o como um problema de otimização mono-objetivo.

Como o funcional de regularização $\Omega(f)$ é compacto (no espaço da funções definidas por f) para a situação $\Omega(f) < c$, $c > 0$, a ordenação de c implica em uma seqüência ordenada de funções onde dado um maior valor de $c_1 > c_2$ $\Omega(f) < c_1 \supset \Omega(f) < c_2$. Nota-se que é equivalente ao mostrado na Equação 2.46.

2.8 Discussão

Neste capítulo foi mostrado de forma rigorosa o problema de aprendizado de máquina que será tratado ao longo desta tese (problema de aprendizado supervisionado). Alguns dos aspectos teóricos da taxa de convergência assim como da taxa de generalização de máquina de aprendizagem foram discutidos. A teoria mostrada nesta seção teve o intuito de dar uma visão ampla dos fatores que interferem no aprendizado, e não de entrar em detalhes técnicos da derivação da mesma. Esta escolha foi feita devido ao fato de que os limites de generalização são análise de pior caso, sendo estes muito super

estimados. Desta forma pode-se utilizá-los para indicar formas construtivas para o aprendizado de máquina, sendo assim derivado o princípio da indução da minimização do risco estrutural. Este princípio se baseia na idéia de que existem dois termos, em geral conflitantes, que se deseja otimizar para se obter um baixo erro de generalização (baixo valor para o risco esperado $R(w)$). Também foi discutido o método da regularização, sendo que foi mostrado que este é um caso particular da minimização do risco estrutural sendo também um problema bi-objetivo.

Fica claro pelas considerações feitas ao longo deste capítulo que é necessário considerar a complexidade na construção de máquinas de aprendizagem. Esta tese mostra que de uma forma geral esta complexidade pode ser escrita em termos de uma norma Q, de tal forma que um modelo genérico para o aprendizado de máquinas pode ser escrito. Antes das propostas desta tese serem apresentadas, serão mostradas no próximo capítulo algumas máquina de aprendizagem que aplicam o SRM.

Capítulo 3

Máquinas de aprendizagem que aplicam o princípio da minimização do risco estrutural (SRM)

Neste capítulo serão mostradas de forma sucinta as máquinas de aprendizagem supervisionado que utilizam o princípio da minimização do risco estrutural para obter boas taxas de generalização, i.e., valores pequenos para o funcional de risco $R(\omega)$. Serão consideradas as redes perceptron de múltiplas camadas (MLP- Multi-layers perceptron), as redes com funções de bases radial (RBF - Radial Basis Function) e a máquinas de vetores suporte (SVM- Support Vector Machine). Para uma referência introdutória sobre estas máquinas de aprendizagem consulte [Hay99].

3.1 Perceptron de múltiplas camadas (MLP - Multi-layer perceptron)

A MLP é provavelmente a máquina de aprendizagem supervisionado mais popular, devido principalmente ao algoritmo da retro-propagação (back-propagation) [RHW86] e ao teorema de aproximação universal [Cyb89] [Fun89] que mostra que esta é capaz de aproximar qualquer função contínua definida em uma região compacta.

3.1.1 Topologia

A MLP é uma topologia baseada em conjuntos de neurônios os quais são distribuídos em paralelo por camadas, e as camadas são distribuídas em cascata como mostrado na Figura 8.1. Uma MLP pode tratar problemas de mapeamento da forma, $f(.) \rightarrow \mathbb{R}^n \rightarrow \mathbb{R}^m$, onde m e n podem ser qualquer número natural, i.e., uma MLP pode mapear um número m de funções simultaneamente.

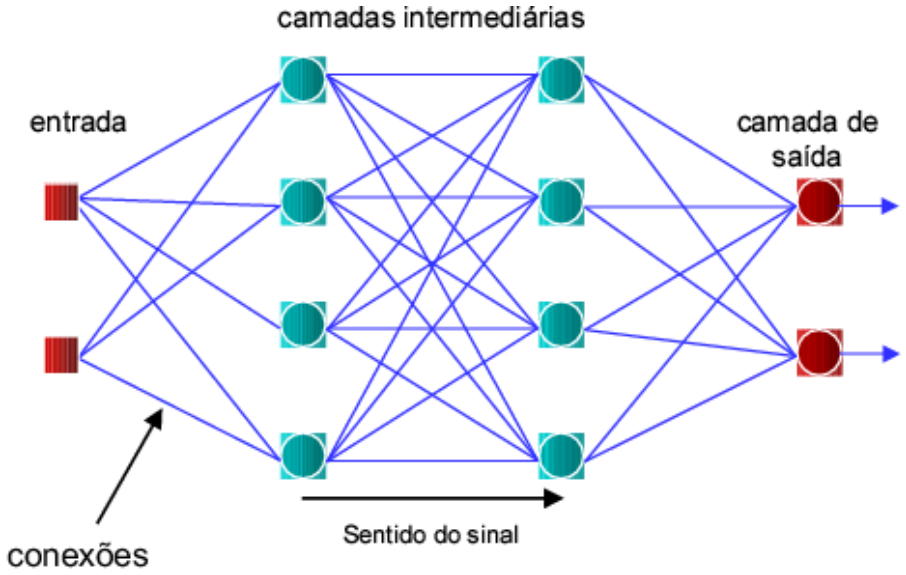

Figura 3.1: Topologia da multi-layer perceptron (MLP)

Nas próximas seções serão explorados os algoritmos de treinamento supervisionado desta topologia. Embora a MLP possa ter mais de uma camada escondida, desta parte do texto em diante serão consideradas somente redes com uma camada escondida, para simplificar as demonstrações. A saída de uma MLP com uma camada escondida, camada de saída linear, m neurônios na camada escondida e n entradas, pode ser escrita como:

$$y_t = \sum_{j=1}^{m} U_j \phi \left(\sum_{i=1}^{n+1} H_{ji} x_{it} \right) \tag{3.1}$$

Onde y_t é a t-ésima saída da rede referente ao t-ésimo padrão de treinamento, U são os pesos da camada de saída, H os pesos da camada escondida, $x_{(n+1)t}$

é a polarização do neurônio e $\phi(.)$ uma função de ativação, sigmoidal por exemplo.

3.1.2 Algoritmos de treinamento para a minimização do risco empírico

Os algoritmos de treinamento supervisionado mais populares para MLPs visando a minimização do risco empírico, em geral, se baseiam na informação do gradiente da função de erro. A função de erro pode ser escrita como:

$$R_{emp}(w, x_t, yd_t) = \sum_{t=1}^{T} e_t(w, x_t, yd_t)^2. \tag{3.2}$$

onde:

$$e_t(w, x_t, yd_t) = (y_t(w, x_t) - yd_t). \tag{3.3}$$

Onde t representa o t-ésimo padrão de treinamento, sendo este composto pelas coordenadas x_t e a saída desejada yd_t. A saída da rede é y_t. Note que esta formulação é facilmente expandida para o caso de mais de uma saída. As épocas (iterações) dos algoritmos de treinamento serão definidas utilizando o índice k. A variável w pode representar um vetor ou uma matriz de pesos, onde a representação dependerá da simplicidade para a explicação tanto para os pesos da camada de saída, U, como os da camada escondida, H.

O algoritmo back-propagation, proposto em 1986, utiliza a técnica do gradiente para a atualização dos pesos da rede [RHW86]. Este método pode ser escrito para um peso da camada escondida como:

$$H_{ji(k+1)} = H_{ji(k)} - \eta \frac{\partial R_{emp(k)}(H, U)}{\partial H_{ji(k)}}. \tag{3.4}$$

A Equação 3.4 descreve a atualização do peso na iteração k como uma perturbação na direção oposta a derivada do erro nesta mesma iteração. A fórmula para a atualização dos pesos U da camada de saída podem ser escritos de forma similar.

Este método tem algumas deficiências, como: i) o algoritmo pode apresentar convergência lenta em regiões muito suaves, ii)não existe metodologia bem definida para a determinação da taxa de aprendizado (η)[1] e iii) para

[1]Na teoria de otimização são propostas algumas técnicas para determinação do passo ótimo (η), como o método da seção áurea. Tradicionalmente, estas não são utilizadas na comunidade de Redes Neurais.

funções unimodais com alta excentricidade o método terá convergência lenta e oscilatória[2].

Uma maneira de melhorar o desempenho do back-propagation é utilizar taxa de aprendizado adaptativa, i.e, η é função de k, $\eta(k)$. Esta técnica é baseada na seguinte regra empírica: o passo deve ser aumentado se o erro decresce de forma consistente, e este deve ser diminuído se o erro apresentar um padrão oscilatório.

Outro algoritmo proposto para melhorar o back-propagation foi o Quick propagation (QPROP). O algoritmo QPROP se baseia em algumas considerações [Fah88]: i)a superfície de erro do conjunto de treinamento é uma parábola em função dos pesos, ii) e esta não é afetada pelo ajuste dos demais pesos da rede. A função de erro é então aproximada pela seguinte fórmula $R_{emp}(k, w) = aw^2(k) + bw(k) + c$. As constantes a, b e c são determinadas em função do erro. Note que a constante a deve ser maior que zero, pois se trata de um problema de minimização. O método se baseia em estimar esta aproximação quadrática do erro e calcular o passo ótimo considerando a função aproximada. Em geral este método é mais rápido que o back-propagation clássico.

Seguindo a idéia de correção de segunda ordem para a direção do gradiente, em [HM94] foi proposta a utilização do método de Levenberg-Marquardt. O método de Levenberg-Marquardt, que é uma modificação do método de Gauss-Newton, é escrito como: $\Delta w = [J^T(w)J(w) + \mu I]^{-1}J^T(w)e(w)$. Onde o parâmetro μ controla o tamanho do passo do método. Este método é reconhecido na literatura como um dos métodos de taxa de convergência mais elevada.

3.1.3 Algoritmos de treinamento para a minimização do risco estrutural

Nesta seção serão discutidas técnicas que foram propostas para a minimização do risco estrutural para MLPs. É importante notar que a maioria destas foram desenvolvidas de forma empírica, mas em geral têm alguma relação com as idéias discutidas no capítulo 2 sobre os limites de generalização.

[2]Para funções quadráticas simples, onde alta excentricidade é percebida o método baseado em gradiente apresenta comportamento oscilatório. A solução para este problema é corrigir a direção do gradiente com uma informação de segunda ordem (Hessiana).

Parada prematura (Early stop)

Uma estratégia para obter redes com boa capacidade de generalização é interromper o treinamento precocemente [WHR90]. Esta técnica foi proposta tendo em vista o fato que o erro de validação passa por um mínimo durante o processo de minimização do erro de treinamento.

Esta é uma técnica interessante que se baseia na hipótese de que a função do erro de validação seja unimodal durante o treinamento, mas infelizmente isto não é sempre verdade. Mas de qualquer forma a parada prematura do treinamento evita que sejam geradas redes com grande complexidade.

Destrição ótima do Cérebro (Optimal Brain Damage)

O algoritmo de poda Optimal Brain Damage [CDS90], faz o ajuste da complexidade da rede alterando a estrutura inicial da mesma. Inicialmente treina-se uma rede superdimensionada para o problema, e após o treinamento, os pesos são eliminados e é calculado como cada peso altera função de erro. Os pesos que alteram menos o erro da rede são eliminados, pois estes são pouco importantes para a mesma. Depois que os pesos foram eliminados, a nova topologia deve ser re-treinada.

Em geral utiliza-se uma série de Taylor truncada no termo de segunda ordem para predizer o efeito da perturbação da função de erro. Observa-se que a rede com alguns pesos eliminados tem a capacidade máxima menor que a da rede inicial.

Validação cruzada (Cross-validation)

A validação cruzada também visa aumentar a capacidade de generalização das redes neurais [Sto74].

Um dos métodos proposto nesta família é o k-fold Cross Validation. Neste, o conjunto de dados é dividido em k partes de forma a constituir k conjuntos diferentes de treinamento e validação, que são utilizados para treinar k redes. Se k é igual ao tamanho do conjunto de dados, este método é denominado leave-one-out Cross-Validation.

Métodos que consideram a minimização da norma dos pesos

Em [Bar97] e [Bar98] foi observado que a dimensão VC de uma rede neural tem como limitante superior o valor da norma dos pesos associados a esta.

Desta forma, limitando a norma dos pesos, também limita-se a dimensão *fat-shattering*. Esta constatação trouxe uma justificativa teórica para o método já existente Decaimento do Peso (Weight Decay -WD) [Hin89] e foi motivação de uma nova família de métodos baseados em idéias multiobjetivo [TBTS00], [CBM+03].

O algoritmo Weight Decay é um método de poda que modifica a função de custo de forma a evitar soluções com normas elevadas [Hin89]. A modificação se baseia na penalização de funções com norma elevada,

$$min\ R_{emp}(w) + \lambda \|w\|^2. \tag{3.5}$$

onde λ representa a importância entre o termo de regularização, $\|w\|$, e o termo do somatório dos erros quadráticos. O termo de penalidade faz com que os pesos tendam a convergir para o menor valor absoluto. Este fenômeno é interessante pois pesos grandes podem comprometer a capacidade de generalização das redes, pois resultam em variância excessiva da saída.

O treinamento multiobjetivo proposto em [TBTS00] re-escreve o problema de treinamento como um problema restrito, de forma similar ao método de regularização apresentado em [Iva62]. Redes neurais treinadas com algoritmo multiobjetivo foram apresentadas utilizando basicamente dois tipos de métodos, o método da relaxação [TPF97], e o baseado em modos deslizantes [CBM+03] (que equivale a um método de otimização por metas). O método da relaxação pode ser visto como uma variação do ϵ-restrito, como mostrado em [TBTS00]. Observe que o método do Decaimento dos Pesos também é uma abordagem multiobjetivo onde o problema a ser resolvido é uma combinação convexa do original [VCV06].

3.2 Funções de base radial (RBF - Radial basis function)

A utilização de redes com funções de base radial foi primeiramente explorada em [BL89]. Uma rede com função base radial (RBF) tem sua forma básica envolvendo três diferentes camadas. A primeira camada é a camada de entrada, onde as entradas, variáveis de controle, do problema estão conectadas. A segunda camada é uma camada escondida de alta dimensionalidade. A terceira camada é a camada de saída, sendo que esta utiliza um neurônio

linear. Uma rede RBF consiste em gerar uma função de saída da seguinte forma:

$$f(w,x) = \sum_{i=1}^{n} w_i \phi(\|x - c_i\|). \tag{3.6}$$

onde $\phi(\|x-c_i\|)$ é um conjunto de n funções, geralmente não-lineares, conhecidas como funções de base radial, c_i's são os centros das funções, e $\|.\|$ denota a norma, que é usualmente Euclidiana. Note que uma rede com função de base radial realiza uma combinação de aproximações locais para resultar em uma aproximação global. Algumas funções podem ser utilizadas na camada escondida, entre elas se destacam a multiquádrica inversa e a Gaussiana, que estão descritas matematicamente nas Equações 3.7 e 3.8, respectivamente.

$$\phi(r) = \frac{1}{(r^2+c^2)^{1/2}} \quad \text{para algum } c > 0 \text{ e r}\geq 0 \ . \tag{3.7}$$

$$\phi(r) = exp(-\frac{r^2}{2\sigma^2}) \quad \text{para algum } \sigma > 0 \text{ e r}\geq 0 \ . \tag{3.8}$$

A topologia desta rede está mostrada na Figura 3.2.

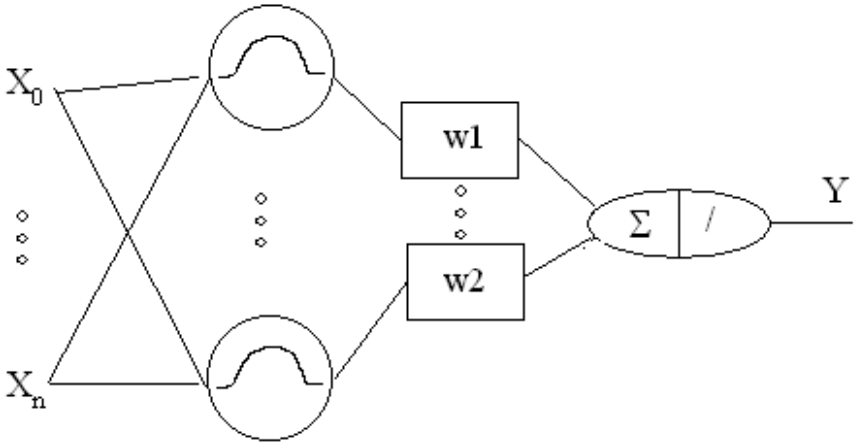

Figura 3.2: Topologia de uma RNA com funções de base radial

Detalhes sobre o teorema de aproximação universal para redes com funções de base radial podem ser encontrados em [PS91].

A idéia utilizada para aumentar a capacidade de generalização das RBFs é baseada na teoria de regularização. Em [GJP95] o funcional de regularização é definido como a suavidade da função. A suavidade foi definida como o comportamento oscilatório de uma função. Desta forma uma função é considerada mais suave que outra se esta tem um comportamento menos oscilatório[3]. No domínio da freqüência pode-se dizer que uma função tem um comportamento menos oscilatório se esta tem menos energia nas altas freqüências. A energia das altas freqüências pode ser medida, passando a função primeiramente em um filtro passa-altas, e depois medindo a norma L_2 do resultado. Matematicamente:

$$\Omega(f) = \int \frac{|F(s)|^2}{G(s)} ds \tag{3.9}$$

onde F representa a transformada de Fourier de f, e G é um filtro passa-altas. A abordagem utilizada para resolver este problema foi a ponderada, como descrito por Tikhonov [Tik63], sendo que pode-se mostrar que a solução é única em termos de uma função de Green associada.

3.3 Máquinas de vetores suporte (SVM - Support vector machine)

A rede SVM (Support Vector Machine) decorre da teoria de aprendizagem estatística, sendo uma aproximação do método de minimização do Risco Estrutural [CV95]. Esta se baseia na idéia mostrada no capítulo 2 que a dimensão VC é função da margem de separação dos dados.

As SVMs mapeiam um vetor de entradas em um espaço de alta dimensão, através de funções não-lineares, de tal forma que as classes sejam linearmente separáveis neste espaço intermediário. Um hiperplano ótimo é então construído de forma a separar estas classes. Este hiperplano ótimo é definido como aquele que maximiza a margem de separação entres as classes, como mostrado na Figura 3.3.

Para se construir uma máquina SVM pode-se, por exemplo, utilizar o núcleo (Kernel) produto interno entre um vetor de suporte e o vetor de entrada. A escolha do núcleo define que tipo da máquina SVM e este pode ser

[3]Observe que desta forma pode-se definir uma seqüência ordenada de espaço em termo da suavidade.

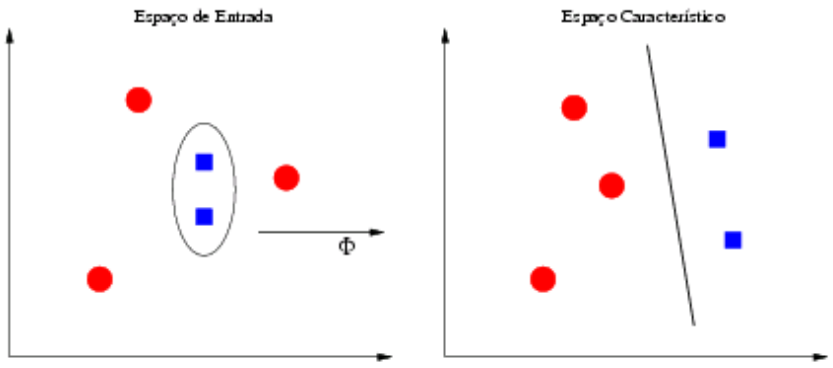

Figura 3.3: Mapeamento não linear realizado pela SVM de tal forma que em um espaço característico, a separação das classes possa ser realizada de forma linear, sendo que o hiperplano de separação ótimo é definido como o que maximiza a margem entre as classes.

de aprendizagem polinomial, base radial, ou perceptron com duas camadas escondidas.

A topologia de uma máquina SVM é mostrada na Figura 3.5.

Para construir o hiperplano ótimo mostrado na Figura 3.4, o seguinte problema de otimização deve ser resolvido:

$$\begin{array}{ll} min & \frac{1}{2}\|w\|^2 \\ s.a. & yd_t((wx_t)+b) \geq 1,\ t=1,...,T. \end{array} . \tag{3.10}$$

O método de Lagrange é utilizado na resolução deste problema, com a introdução dos multiplicadores de Lagrange, $\alpha_i \geq 0$ e o Lagrangeano é dado conforme 3.11.

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{t=1}^{T} \alpha_t (yd_t((wx_t)+b) - 1). \tag{3.11}$$

O Lagrangeano $\mathcal{L}$ deve ser minimizado em relação às variáveis primais w e b, e maximizado em função das variáveis duais, α_t. Se alguma restrição é violada, $yd_t((wx_t)+b) - 1 < 0$, $\mathcal{L}$ pode ser aumentado aumentando-se o

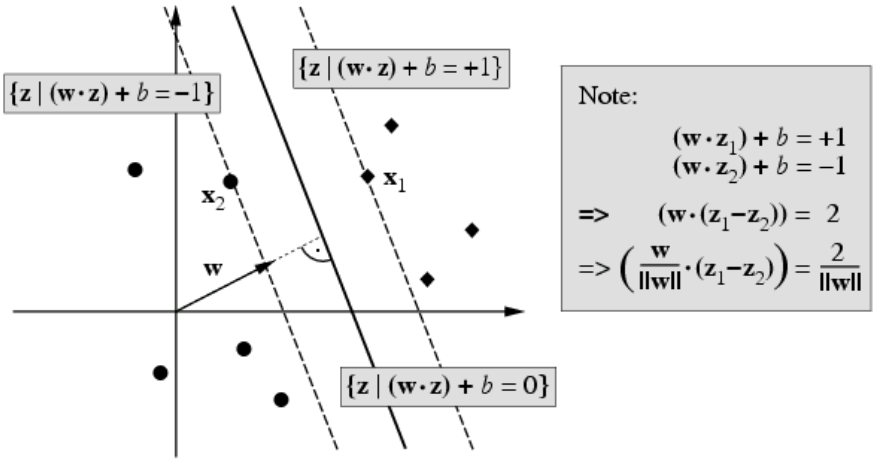

Figura 3.4: Para este problema de classificação binário o hiperplano de separação ótimo é ortogonal à linha mais curta que conecta as cascas convexas das duas classes, sendo que este intercepta esta em sua metade (distância igual entre as duas classes). Para situações onde as classes são separáveis existe um vetor de pesos e uma polarização tal que $yd_t((wx_t) + b) > 0(t = 1, .., T)$. Escalonando w e b de tal forma que os pontos mais próximos do hiperplano satisfaçam $|(wx_t) + b| = 1$, obtém-se a fórmula canônica do hiperplano. Para este caso, a margem perpendicular ao hiperplano é igual a $2/\|w\|$.

respectivo α_t. Desta forma w e b devem ser alterados de forma que $\mathcal{L}$ decresça. Para prevenir que o termo $\alpha_t(yd_t((wx_t)+b)-1)$ fique arbitrariamente grande, as alterações em w e b deverão evitar estas situações, de forma que as res-
trições sejam gradativamente respeitadas, considerando que o problema seja separável. Utilizando as condições de otimalidade de Karush-Kuhn-Tucker(K-K-T)[4], pode-se induzir que os α_t's correspondentes às situações onde as restrições não estão ativas, devem ser feitos iguais a zero. Para a condição ótima as derivadas de $\mathcal{L}$ em relação as variáveis primais devem zerar,

[4]Detalhes sobre as condições de Karush-Kuhn-Tuker podem ser encontradas no apêndice A sobre otimização multiobjetivo.

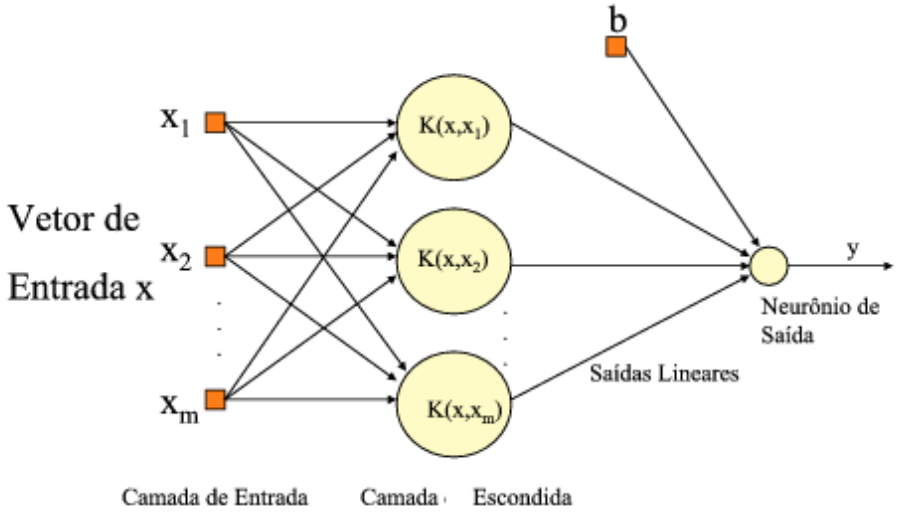

Figura 3.5: Topologia de uma máquina SVM.

$$\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial b} = 0 \Rightarrow \sum_{t=1}^{T} \alpha_t y d_t = 0$$

$$\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial w} = 0 \Rightarrow w = \sum_{t=1}^{T} \alpha_t y d_t x_t = 0. \tag{3.12}$$

O vetor solução é então uma expansão em termos de um subgrupo do conjunto de treinamento, para o qual o α_t correspondente é diferente de zero, e estes pontos do conjunto de treinamento são denominados vetores de suporte (Support Vectors). Os vetores de suporte se encontram na margem. Todos exemplos restantes não são utilizados no processo de otimização. Esta característica está de acordo com a nossa intuição que diz que a separação é definida pelos padrões mais próximos do hiperplano, e que a solução não é dependente de outros exemplos. Utilizando as condições de Karush-Kuhn-Tucker, pode-se escrever que:

$$\alpha_t[yd_t((wx_t) + b) - 1] = 0,\ t = 1, ..., T. \tag{3.13}$$

Utilizando as Equações 3.12 e 3.11, pode-se eliminar as variáveis primais e escrever a equação dual de Wolfe, tal que:

$$\begin{aligned} max \quad & W(\alpha) = \sum_{t=1}^{T} \alpha_t - \frac{1}{2} \sum_{t=1}^{T} \sum_{j=1}^{T} \alpha_t \alpha_j y_t y_j (x_t x_j) \\ s.a. \quad & \alpha_t \geq 0, \ t = 1, ..., T. \ e \ \sum_{t=1}^{T} \alpha_t y_t = 0 \end{aligned} \tag{3.14}$$

Onde o hiperplano de decisão pode ser escrito como:

$$f(x) = sinal(\sum_{t=1}^{T} y_t \alpha_t (x x_t) + b) \tag{3.15}$$

A polarização b é calculada utilizando um vetor suporte e a Equação 3.13.

Existem diversos argumentos teóricos que confirmam a boa capacidade de generalização quando o hiperplano ótimo é utilizado.

3.3.1 Espaço característico e núcleos (Kernels)

Para construir máquinas baseadas em vetores suporte, o algoritmo teve de ser complementado por um método de calcular produto interno em espaços característicos não-linearmente relacionados com o espaço de entrada. A idéia básica é mapear os dados em outro espaço F (denominado de espaço característico), utilizando um mapeamento não linear e utilizar o algoritmo linear em F.

Deve se notar que os problemas descritos nas Equações (3.14) e (3.15) somente necessitam do cálculo dos produtos internos, $(\phi(x).\phi(y))$. Esta informação é essencial, pois estes podem ser, em alguns casos, mapeados utilizando um núcleo simples:

$$k(x, y) = (\phi(x)\phi(y)). \tag{3.16}$$

Por exemplo, pode-se utilizar um kernel polinomial:

$$k(x, y) = (xy)^d. \tag{3.17}$$

Outras opções são o núcleo Gaussiano e o MLP, que utiliza uma função tangente hiperbólica. Considere $K(x, z)$ uma matriz simétrica, $K(i, j) = k(x_i, x_j)$. De acordo com o teorema de Mercer esta é um núcleo se e somente se K é definida positiva, i.e., $x^T K x > 0, \ \forall x \neq 0$.

3.4 Discussão

Neste capítulo foram apresentados diversos métodos de treinamento de máquinas de aprendizagem que visam minimizar, de forma direta ou indireta, o risco estrutural. De fato todas as técnicas discutidas tentam, de alguma forma restringir o espaço onde as funções realizáveis pela máquina se encontram.

Alguns pontos importantes devem ser discutidos. Para as MLPs foi mostrado que a norma dos pesos representa um limite superior para a dimensão *fat-shattering*. De fato em [Bar98] as demonstrações mostram que é suficiente que a norma dos pesos da camada de saída seja limitada para que a máquina tenha uma dimensão VC limitada. Este fato pode ser entendido da seguinte forma. Considere as saídas dos neurônios da camada escondida como um novo espaço dimensional. Desta forma a camada de saída representa um hiperplano neste novo espaço. A minimização da norma dos pesos da camada de saída equivale à construção de um hiperplano de separação ótimo neste novo espaço dimensional, de forma similar às SVMs.

Pode-se analisar as condições utilizadas em [Bar98] para mostrar a importância da norma dos pesos. Considere $\sigma : \mathbb{R} \rightarrow [-M/2, M/2]$ sendo uma função monotônica crescente. Defina a classe $\mathcal{F}$ de funções no $\mathbb{R}^n$ como:

$$\mathcal{F} = \{x \longmapsto \sigma(wx + w_0) : w \in \mathbb{R}^n, w_0 \in \mathbb{R}\} \tag{3.18}$$

e definindo-se a hipótese escolhida pela máquina de treinamento como:

$$\mathcal{H} = \{\sum_{i=1}^{N} \alpha_i f_i : N \in \mathbb{N}, f_i \in \mathcal{F}, \sum_{i=1}^{N} |\alpha_i| \leq A\} \tag{3.19}$$

para $A \geq 1$. Logo para $\gamma \leq MA$

$$fat(\mathcal{H}, \gamma) \leq \frac{cM^2A^2n}{\gamma^2} log\left(\frac{MA}{\gamma}\right) \tag{3.20}$$

Para alguma constante c.

É sabido que, o limite na dimensão *fat-shattering* implica em um limite no erro de generalização. Em [Bar98] também é mostrado que a dimensão $fat - shattering$ de uma função sigmoidal também é limitada pela norma do vetor de pesos da mesma.

Pode-se observar que todos os métodos que implementam a minimização da norma dos pesos são, de fato, mecanismos computacionais para resolver um problema multiobjetivo geral. Pode-se ver que o método WD é equivalente ao método de regularização proposto por Tikhonov [Tik63] [TA77]. O método multiobjetivo proposto em [TBTS00] é equivalente aos métodos de regularização descritos em [Phi62] [Iva62] [Iva76], quando o método computacional utilizado para a solução do mesmo é o ϵ-restrito. Desta forma pode-se observar que os métodos propostos em [TBTS00] são extensões dos métodos de regularização já existentes que foram propostos no começo da década de 60. O grande mérito do trabalho que propôs o treinamento multiobjetivo para redes neurais é que este coloca uma luz, sob a óptica da otimização multiobjetivo, na natureza do problema. No caso citado no capítulo anterior onde os funcionais envolvidos eram convexos, era possível mostrar que as três técnicas de regularização eram equivalentes [Vas70]. Entretanto, sob a luz da teoria de otimização, pode-se mostrar facilmente que para redes neurais os problemas não são necessariamente equivalentes, e ainda que existem resultados que podem ser alcançados utilizando o treinamento multiobjetivo que não podem ser encontrados utilizando o WD, que de fato implementa uma combinação convexa dos objetivos[5].

Em [And02] foi mostrado que existe uma equivalência bi-lateral entre as RBFs regularizadas e as SVMs. Bi-lateral no sentido que dado uma rede RBF consegue-se gerar uma SVM equivalente e vice-versa. Esta demonstração mostra que abordagens diversas podem gerar soluções de mesma qualidade.

De forma geral, neste capítulo foram mostradas as principais técnicas existentes para melhorar a generalização de máquinas de aprendizagem e as características que as diversas técnicas têm em comum. Como já havia sido dito anteriormente, as técnicas de regularização implementam o princípio de minimização do risco estrutural. Outro ponto importante que foi ressaltado é que considerar o problema de regularização como um problema multiobjetivo, embora leve geralmente a mecanismos já conhecidos há mais de 40 anos, traz uma nova perspectiva do problema, que pode ser, e em geral é, muito útil.

As idéias discutidas nestes capítulos iniciais tiveram o propósito de embasar uma discussão mais profunda sobre métodos de treinamento de redes neurais. Pode-se observar que os esforços foram na direção de dar uma visão mais geral do problema de aprendizado e das características construtivas dos

[5]Para maiores informações sobre a otimização multiobjetivo consulte o apêndice A sobre o assunto.

métodos que se mostraram eficientes ao longo dos anos.

Das idéias apresentadas nesta parte introdutória foi derivada a Q-norma como medida de complexidade. Esta medida traz uma estrutura eficiente computacionalmente pois separa as influências da parte linear e não-linear do problema. A Q-norma e o método do gradiente mínimo, contribuições desta tese, são mostrados no capítulo a seguir.

Parte II

Contribuições

Capítulo 4

A medida de complexidade baseada na norma-Q, o método do gradiente mínimo e suas implicações

4.1 A medida de complexidade norma-Q

Para se controlar a complexidade de uma máquina de aprendizagem, a primeira questão a ser tratada é como medi-la. Nas seções anteriores, algumas técnicas que usam a idéia de controle de complexidade foram apresentadas. Do ponto de vista teórico, a principal ferramenta utilizada nos dias atuais é a dimensão VC e as suas generalizações. Em termos práticos, qualquer medida de complexidade para ser utilizável deve ser escrita como função das variáveis de entrada e dos parâmetros da máquina de aprendizagem. Adicionalmente, também é interessante se utilizar funções quadráticas (ou pelo menos uni-modal), pois estas são simples de serem otimizadas. Por exemplo, MLPs usam a norma Euclidiana do vetor de pesos, enquanto as SVMs usam a norma do vetor normal ao hiperplano de separação. O vetor normal ao hiperplano está fortemente conectado com o vetor de pesos; este é, de fato, o vetor de pesos para um modelo linear. Para qualquer norma e qualquer transformação bijetiva linear A, pode-se definir uma nova norma de w como $\|Aw\|$. Em $2D$, com A sendo uma rotação de 45° e algum escalonamento necessário, a norma-1 se transforma em norma-∞. De forma genérica, o

problema de medir a complexidade pode ser escrito como uma norma-Q. Considere $w \in \mathbb{R}^n$ um vetor com dimensão finita, e $Q \in \mathbb{R}^n$ uma matriz simétrica definida positiva[1] . A complexidade em termos de uma norma-Q de w, $\sqrt{w^T Q w}$, pode ser definida como

$$\Omega(w) \equiv w^T Q w. \tag{4.1}$$

À primeira vista a condição $w^T Q w > 0$ parece ser restritiva, mas é de fato uma característica necessária do problema bi-objetivo apresentado aqui, que é um problema de minimização. Todas as técnicas citadas no capítulo anterior podem ser reescritas na forma da norma-Q. Para as MLPs e SVMs, utilizando a norma do vetor de pesos como medida de complexidade, a matriz Q será a matriz identidade. Alterando Q outros métodos podem ser gerados. Como Q é uma matriz simétrica positiva, esta pode ser decomposta na forma $Q = A^T A$, para algum A, logo, a norma da complexidade defnida na Eq. (4.1) é a norma do vetor transformado $\|Aw\|$. A utilidade desta formulação é que esta pode representar a complexidade de diversas máquinas de aprendizagem as diferenciando na matriz Q ou na transformação linear A no caso onde uma norma não Euclidiana é utilizada. A utilização da medida de complexidade proposta traz também uma outra vantagem: a decomposição das partes linear e não-linear do problema. Considere uma máquina de aprendizagem com os parâmetros lineares l e os não-lineares N. A medida de complexidade pode então ser escrita como $l^T Q(N) l$, de maneira que uma decomposição entre as partes lineares e não lineares do problema seja alcançada. Observe que este tipo de decomposição é o principal resultado dos métodos baseados em Kernel. Estes métodos controlam a complexidade controlando somente a parte linear das máquina (as não linearidades são definidas pelos Kernels). Nas próximas seções uma nova máquina de aprendizagem, denominada de Método do Gradiente Mínimo (Minimum Gradient Method- MGM) é definida. A idéia de se utilizar o gradiente como medida de complexidade é derivada do conceito de maximização da margem, mas esta também está relacionada com outras propriedades interessantes como a minimização da energia das altas-freqüências.

[1] $Q = Q^T$ e $x^T Q x > 0, \ \forall x \neq 0$

4.2 O método do Gradiente Mínimo

Neste capítulo será discutida uma maneira de se utilizar a teoria desenvolvida no capítulo 2 de forma construtiva, para se obter uma nova máquina de aprendizagem. O conceito fundamental utilizado para descrever as propriedades de generalização de uma máquina de aprendizagem foi o conceito da dimensão VC, que mede a capacidade de um conjunto de funções. Como foi observado anteriormente a dimensão VC efetiva depende do conceito de margem entre os funcionais e esta pode ser definida para uma dada amostra yd_t como:

$$\gamma_t = yd_t f(x_t). \tag{4.2}$$

A distribuição de margens de f, $M(S, f)$, em relação ao conjunto de treinamento S pode ser definida como:

$$M(S, f) = \{\gamma_t = yd_t f(x_t) : t = 1, ..., T\}. \tag{4.3}$$

A idéia de margem é fundamental para a construção da dimensão $fat - shattering$, que pode ser entendida como a dimensão VC efetiva.

4.3 Maximização da margem de conjunto de exemplos

Como dito anteriormente a margem de uma amostra yd_t, que pode ser definida como em 4.2, deve ser maximizada para se obter melhores limites de generalização para uma máquina.

Para encontrar o máximo desta função de margem para uma dada amostra yd_t, é necessário que o gradiente de 4.2 seja igual a zero. Desta forma obtém-se[2]:

$$\frac{\partial \gamma_t}{\partial x_{it}} = \frac{\partial [yd_t f(x_t)]}{\partial x_{it}} = 0 \tag{4.4}$$

$$= yd_t \frac{\partial f(x_t)}{\partial x_{it}} = 0 \tag{4.5}$$

$$\therefore \|\nabla f(x_t)\| = 0. \tag{4.6}$$

[2]O índice t indica o t-ézimo padrão. Para x_t deseja-se derivar em relação as coordenadas do espaço de entradas, x_{it}, onde $i = 1, ..., n + 1$.

Para um conjunto de exemplos pode-se pensar no problema de maximização da margem como:

$$min \quad \sum_{t=1}^{T} \|\nabla f(x_t)\|. \tag{4.7}$$

onde pode-se concluir que a norma do gradiente da função para um dado ponto pode ser utilizada para realizar o princípio da minimização do risco estrutural. Desta forma uma contribuição importante, do ponto de vista teórico desta tese, se resume em escrever o problema de aprendizado de máquina como o problema de otimização bi-objetivo[3]:

$$min \begin{cases} f_1 = R_{emp}(w) \\ f_2 = \sum_t^N \|\nabla f(x_t)\|^2 \end{cases} \tag{4.8}$$

O problema formulado desta maneira tem como meta conseguir o equilíbrio entre dois fatores responsáveis pela capacidade de generalização de uma máquina que são em geral conflitantes. Este método será denominado como o método do gradiente mínimo.

De fato a idéia de se utilizar a norma da k-ésima derivada parcial como termo de suavidade para problemas de regularização aparecem em alguns momentos na literatura sem justificativas teóricas. Nas seções seguintes será analisado as implicações desta idéia.

4.4 O hiperplano de separação ótimo revisitado

Nesta seção será discutido, sob a óptica das idéias mostradas anteriormente, um dos problemas mais tratados nos últimos anos na literatura sobre aprendizado de máquinas, o hiperplano de separação ótimo. Pode-se definir um hiperplano como:

$$f(x, w, b) = wx + b. \tag{4.9}$$

No capítulo 2 mostrou-se que a dimensão VC efetiva de um hiperplano depende da margem de separação entre as classes. As SVMs, como mostrado

[3]O objetivo f_2 não foi escrito em termos das amostras do conjunto de treinamento, tamanho T, e sim de um conjunto de tamanho N. A derivada pode ser avaliada, sem um custo adicional elevado em pontos diferentes do conjunto de treinamento, como por exemplo nos pontos de validação. Pode-se se pensar assim em reduzir a capacidade da função não só nos pontos de treinamento como em outros pontos de interesse.

no capítulo 3, utilizam esta idéia para gerar as máquinas de aprendizagem. Para se obter um hiperplano de separação ótimo, dado um problema linearmente separável, foi mostrado que basta minimizar o seguinte funcional:

$$\tau(w) = \frac{1}{2}\|w\|^2 \tag{4.10}$$

A minimização do funcional mostrado na Equação 4.10 é a base da construção das SVMs. Pode-se escrever a margem para o hiperplano como:

$$\gamma = ydf(x) = yd[wx + b], \tag{4.11}$$

sendo que a norma do gradiente de γ em relação a x é proporcional a:

$$\|\nabla f(x)\| = \|w\|, \tag{4.12}$$

que é equivalente ao funcional descrito na Equação 4.10. Pode-se observar desta forma que para este caso particular a formulação apresentada é equivalente à das SVMs e que, para um problema linear, a norma do gradiente além do significado de máxima margem no espaço das funções também significa máxima margem no espaço das variáveis. Como se trata de um problema linear o gradiente representa a projeção do ponto de interesse na curva de separação das classes, sendo esta a menor distância, em termos da norma Euclidiana, nos espaço das variáveis.

Pode-se observar que as idéias mostradas até aqui trabalhavam com a margem no espaço das funções. A derivação da fórmula do método do gradiente mínimo também é realizado no espaço das funções. Entretanto é interessante notar que para este caso particular existe equivalência entre os dois espaços.

4.5 Análise para funções monotônicas

Considere uma função monotônica genérica unidimensional, aproximada linearmente em diversos intervalos $\Delta x_i = x_i - x_{i-1}$, onde a variação referente a cada intervalo é definida como $\Delta f_i = f_{i-1} - f_i$, como mostrado na Fig. 4.1.

Para analisar este problema pode-se pensar que a função f_0 tem valor igual a 1, e define uma classe em um problema de classificação, e que a

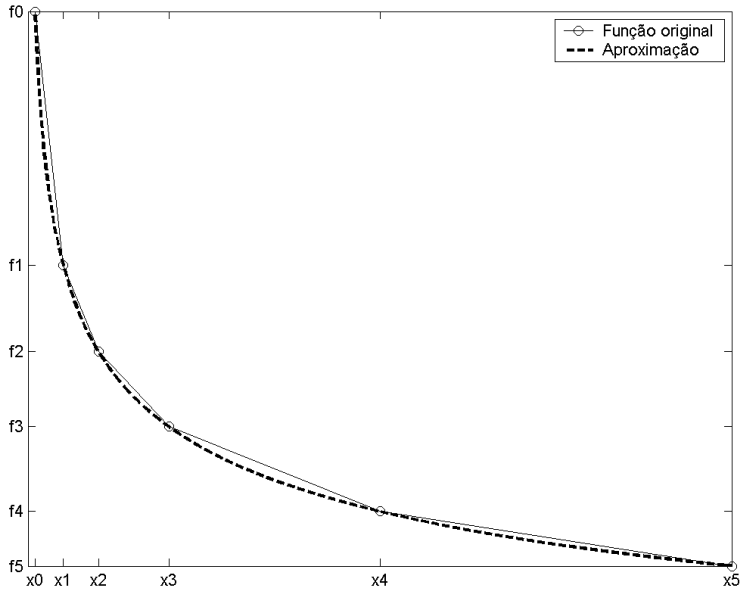

Figura 4.1: Função convexa unidimensional aproximada por cinco segmentos de reta.

função f_n (para o caso da figura $n = 5$) representa a transição entre a classe 1 e -1, i.e., $f_n = 0$. Para esta situação pode-se escrever:

$$f_n = f_0 - \sum_{i=1}^{n} \frac{\Delta f_i}{\Delta x_i} \Delta x_i. \tag{4.13}$$

Considerando $f_n = 0$ e $f_0 = 1$ (observe que esta consideração somente simplifica a demonstração e não particulariza a análise), pode-se reescrever a Equação 4.13 como:

$$1 = \sum_{i=1}^{n} \frac{\Delta f_i}{\Delta x_i} \Delta x_i. \tag{4.14}$$

Para simplificar a análise pode-se definir $\Delta x_i = \Delta x$, $\forall i = 1, ..., n$. Desta

forma pode-se reescrever a Equação (4.14) como:

$$1 = \Delta x \sum_{i=1}^{n} \frac{\Delta f_i}{\Delta x}. \tag{4.15}$$

Logo:

$$\frac{1}{\Delta x} = \sum_{i=1}^{n} \frac{\Delta f_i}{\Delta x}. \tag{4.16}$$

Da Equação 4.15 observa-se que a distância Δx entre as classes é inversamente proporcional à inclinação de cada segmento de reta. Isto é, para sair da região onde a função define a classe 1 e a região de transição, é necessário "andar" um maior Δx se as inclinações das retas forem pequenas. Por exemplo, se $\Delta x = 1$, tem-se que $\gamma = \Delta x n = n$, e $\sum_{i=1}^{n} \frac{\Delta f_i}{\Delta x} = 1$. Se $\Delta x = 1/2$, tem-se que $\gamma = \Delta x n = n/2$, e $\sum_{i=1}^{n} \frac{\Delta f_i}{\Delta x} = 1/2$. Outra observação interessante é que pode-se considerar Δx tão pequeno quanto se quiser, e desta forma aproximar a função com a precisão desejada, e ainda, no limite, $\Delta x \to 0$, o termo $\frac{\Delta f_i}{\Delta x} \to -\frac{df}{dx}$, o que faz esta avaliação equivalente a minimização da norma da derivada (deve-se lembrar que para funções monotônicas as derivadas não mudam de sinal).

Pode-se observar que o caso que utiliza uma reta de separação é um subcaso deste analisado, sendo que no primeiro é considerado $n = 1$.

Desta análise foi mostrado que a maximização da margem no espaço de funções quando funções monotônicas são consideradas, equivale à maximização desta no espaço das variáveis. Isto é, quanto menor a inclinação dos segmentos de retas, maior será o Δx, sendo que o último representa a distância no espaço das funções.

4.6 O problema de minimização das componentes de alta freqüência revisitado

Como dito anteriormente, em [GJP95] foi proposto, para redes RBFs, a minimização de um termo de suavidade que era escrito como a minimização da energia das altas freqüências da função aproximada. Para estudar este pro-

blema e mostrar outras propriedades da idéia de mínimo gradiente, considere a transformada de Fourier n-dimensional:

$$f(t_1, ..., t_n) = \frac{1}{2\pi^n} \int_{-\infty}^{+\infty} F(j\omega_1, ..., j\omega_n) \prod_{i=1}^{n} e^{j\omega_i t_i} d\omega_1 ... d\omega_n \tag{4.17}$$

A derivada de $f(t)$ em relação à t tem a seguinte transformada para quase todos ω:

$$\frac{\partial f(t_1, ..., t_n)}{\partial t_k} = \frac{1}{2\pi^n} \int_{-\infty}^{+\infty} j\omega_k F(j\omega_1, ..., j\omega_n) \prod_{i=1}^{n} e^{j\omega_i t_i} d\omega_1 ... d\omega_n \tag{4.18}$$

Desta forma pode-se escrever a seguinte propriedade que a derivada parcial de $f(.)$ em relação a um t_k corresponde do lado direito da Eq. 4.17 à multiplicação de $F(.)$ por jw_k, isto é[4]:

$$\frac{\partial f(t_1, ..., t_n)}{\partial t_k} \longleftrightarrow j\omega_k F(j\omega_1, ..., j\omega_n) \tag{4.19}$$

Desta forma observa-se que a norma do gradiente é equivalente, em uma grande gama de funções, à minimização da energia das altas freqüências. O termo $F(.)$ representa a energia, sendo que esta está sendo multiplicada pelo ω_k equivalente. Desta forma, quanto maior o ω_k maior será a amplificação do termo $F(.)$ correspondente, sendo assim montada uma filtragem passa-altas. Outro ponto interessante a ser notado é que em casos que a transformada de Fourier exista para derivadas de ordem mais elevadas, é equivalente a escolha de diferentes filtragens no sinal para minimizar a energia das altas freqüências (pode ser entendido como mudar a definição de "alta" freqüência). De fato para a derivada n-ésima tem-se $\omega_k^n F(.)$, sendo então a filtragem realizada por ω_k^n.

Dada esta interpretação, o hiperplano de separação ótimo pode ser visto como uma função suave, definindo-se suave como aquela que possui reduzida energia em freqüência.

4.7 Minimização da norma dos pesos re-visitada

No treinamento multiobjetivo tenta-se diminuir o erro de treinamento e ao mesmo tempo diminuir a norma do vetor de pesos. Considerando que as

[4]Esta propriedade é válida para uma diversidade de funções desde que a função seja integrável [Bar93].

funções de ativação $\phi(.)$ sigmoidal ou tangente hiperbólica têm comportamento praticamente linear quando seus argumentos são pequenos, pode-se então escrever:

$$\phi(w, x) \approx wx \quad \text{se} \parallel w \parallel \rightarrow 0 \tag{4.20}$$

Logo a saída de uma MLP, onde U e H são os pesos da camada de saída e da camada escondida respectivamente, pode ser reescrita como:

$$y_t = \sum_{j=1}^{m} \sum_{i=0}^{n} U_j H_{ij} x_{it} \tag{4.21}$$

Logo, minimizar a norma do gradiente da Equação 4.21 equivale a:

$$\|\nabla f(x_t)\|^2 = \sum_{j=1}^{m} \sum_{i=0}^{n} (U_j H_{ij})^2. \tag{4.22}$$

Deste resultado pode-se notar que se os valores dos pesos forem pequenos a função será suave, como já poderia ser previsto considerando o comportamento linear das funções de ativação. Entretanto pode-se notar que, mesmo quando a linearização é considerada, a norma dos pesos não é a medida mais representativa de suavidade, sendo que esta deve ser substituída pela medida apresentada na Equação 4.22. O resultado mostrado nesta equação é condizente com a expectativa empírica que nos diz que um neurônio pode cancelar o efeito de outro, para um dado ponto no espaço, de tal forma que estes não influenciem a saída, quão menos na suavidade da mesma. Deve-se lembrar que a norma dos pesos da camada não linear deve ser mantida pequena, para que as condições de análise desta seção sejam válidas, i.e., as funções de ativação trabalhem em suas regiões lineares. Observe que um resultado equivalente ao apresentado nesta seção foi apresentado em [MR97]. Outro ponto que também é um pouco intuitivo e é mostrado nesta equação é que não faz sentindo considerar somente a norma de todos os pesos, pois estes, de fato, estão em espaços vetoriais diferentes. O resultado mostrado na Equação 4.22, indica que realmente os pesos têm importâncias diferentes para o comportamento da função aproximada, pois como já discutido anteriormente, estes estão em espaços diferentes. Os pesos para as SVMs, Eq. 4.12, estão em um mesmo espaço, logo, desta forma a norma deste faz sentido.

4.8 SVMs não lineares

Considere o problema primal das SVMs não lineares (soft-margin)

$$\min\ W(\phi) = \phi Q \phi + C \sum_{i=1}^{T} \xi_i \tag{4.23}$$

$$s.a.\ yd_i[x_i\phi + b] \geq 1 - \xi_i,\ i = 1, ..., T \tag{4.24}$$

Em geral a matriz Q é utilizada como uma matriz identidade. Considere então Q como uma matriz simétrica definida positiva. Logo, existe uma matriz W tal que $A = \sqrt{Q}$. A função objetivo pode então ser escrita como:

$$W(\phi) = A\phi A\phi + C \sum_{i=1}^{T} \xi_i. \tag{4.25}$$

Escrevendo $\varphi = A\phi$ e $z_i = A^{-1}x_i$ o treinamento da SVM pode então ser re-escrito como:

$$\min\ W(\varphi) = \varphi\varphi + C \sum_{i=1}^{T} \xi_i \tag{4.26}$$

$$s.a.\ yd_i[z_i\varphi + b] \geq 1 - \xi_i,\ i = 1, ..., T \tag{4.27}$$

Pode-se observar então que existe uma transformação linear dos vetores x em vetores z que o problema de maximização da margem, $Q = I$, se transforma em uma matriz Q positiva definida. De fato a escolha do Kernel implica a estrutura de regularização escolhida, i.e., a matriz Q.

4.9 Discussão

Neste capítulo foi derivado, partindo da idéia de margem de um padrão, o método do gradiente mínimo. Embora existam idéias similares a esta na literatura, estas foram desenvolvidas de forma empírica. O ponto mais importante deste capítulo é que, utilizando a idéia de gradiente mínimo, pode-se derivar as fórmulas mostradas no capítulo anterior para a minimização do risco estrutural. Foi possível obter uma interpretação do método do hiperplano ótimo e também da técnica que trabalha com a redução da energia

das altas freqüências. As idéias apresentadas neste capítulo, derivadas do método proposto ajudam a esclarecer um pouco o problema de aprendizado de máquina e a relacionar as diversas técnicas já existentes, sendo esta uma contribuição importante.

No capítulo seguinte serão apresentados e analisados os aproximadores polinomiais de gradiente mínimo, a primeira contribuição prática deste trabalho. Para aproximadores polinomiais será mostrado como se escrever o MGM na forma de uma norma-Q como proposto neste trabalho.

Capítulo 5

Aproximadores polinomiais de gradiente mínimo

Neste capítulo será estudada a aplicação da minimização da norma do gradiente em aproximadores polinomiais. Trata-se de um caso de estudo simples que servirá para esclarecer alguns pontos sobre o comportamento da técnica proposta.

5.1 Formulação para aproximadores polinomiais de gradiente mínimo

Pode-se escrever um polinômio unidimensional de ordem τ como:

$$y_t = p_{(\tau+1)}x_t^{\tau} + p_{(\tau)}x_t^{\tau-1} + ... + p_{(2)}x_t^1 + p_{(1)}. \tag{5.1}$$

Pode-se então reescrever a Equação 5.1 em forma matricial, i.e. para todos os y_t, $t = 1, ..., T$, como:

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix} \begin{bmatrix} x_1^{\tau} & x_1^{\tau-1} & \dots & 1 \\ \vdots & \ddots & \vdots & \\ x_T^{\tau} & x_T^{\tau-1} & \dots & 1 \end{bmatrix} \begin{bmatrix} p_{(\tau+1)} \\ \vdots \\ p_{(1)} \end{bmatrix} = CP. \tag{5.2}$$

A derivada da Equação 5.1 pode ser escrita como:

$$\frac{dy_t}{dx}|_{x=x_t} = \sum_{i=1}^{\tau} p_{(\tau+2-i)}(\tau+1-i)x_t^{\tau-i}. \tag{5.3}$$

A Equação 5.3 pode ser escrita, de forma similar a Equação 5.2, como:

$$\frac{dy}{dx} = \begin{bmatrix} \tau x_1^{\tau-1} & (\tau-1)x_1^{\tau-2} & \dots & x_1 & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 \\ \tau x_T^{\tau-1} & (\tau-1)x_T^{\tau-2} & \dots & x_T & 0 \end{bmatrix} \begin{bmatrix} p_{(\tau+1)} \\ \vdots \\ p_{(1)} \end{bmatrix} = DP. \tag{5.4}$$

Neste trabalho foi proposta a utilização da norma do gradiente como medida de complexidade para a função aproximada. O seguinte problema bi-objetivo foi proposto para definir uma técnica de minimização do risco estrutural:

$$min \begin{cases} f_1 = R_{emp}(w) \\ f_2 = \sum_t^N \|\nabla f(x_t)\|^2 \end{cases} \tag{5.5}$$

Pode-se então escrever o termo da minimização do risco empírico como a minimização do funcional de erro:

$$f_1 = (CP - yd)^T (CP - yd) =$$

$$= yd^T yd - yd^T CP - P^T C^T yd + P^T C^T CP, \tag{5.6}$$

onde yd é a resposta desejada. Para minimizar a norma do gradiente pode-se escrever:

$$f_2 = (DP)^T (DP) = P^T QP \tag{5.7}$$

onde que é $Q = D^T D$. A Equação 5.6 representa o erro, e a Equação 5.7 representa as derivadas que foram escritas na forma de uma norma-Q. Para os aproximadores polinomiais a matriz Q, dada a formulação do gradiente mínimo, é função somente da ordem do polinômio. Note que as duas funções são quadráticas. Logo pode-se escrever um problema multiobjetivo no qual se deseja minimizar o erro e a derivada da função nos pontos de treinamento como[1]:

$$f = \lambda f_1 + (1-\lambda) f_2$$

$$= \lambda (CP - yd)^T (CP - yd) + (1-\lambda) P^T QP \tag{5.8}$$

[1]Caso as funções não fossem quadráticas seria necessário utilizar uma outra formulação para a resolução do problema multiobjetivo. Para mais detalhes consulte o apêndice A.

Onde o P ótimo é calculado diferenciando a Equação 5.8, e igualando a zero. A derivada da Equação 5.8 em relação aos parâmetros P pode ser calculada como[2]:

$$\frac{df}{dP} = \lambda\frac{df_1}{dP} + (1-\lambda)\frac{df_2}{dP}$$

$$= \lambda[(-yd^TC)^T - C^Tyd + (C^TC + C^TC)P] + (1-\lambda)(Q+Q^T)P$$

$$= \lambda(-2C^Tyd + 2C^TCP) + (1-\lambda)2QP \tag{5.9}$$

Para encontrar o P ótimo, P^*, basta igualar a Equação 5.9 a zero e resolvê-la. Matematicamente:

$$\lambda(-2C^Tyd + 2C^TCP) + (1-\lambda)2QP = 0$$

$$-2\lambda C^Tyd + 2\lambda C^TCP + (1-\lambda)2QP = 0$$

$$[\lambda C^TC + (1-\lambda)Q]P = \lambda C^Tyd$$

$$P^* = [\lambda C^TC + (1-\lambda)Q]^{-1}\lambda C^Tyd. \tag{5.10}$$

Onde o mapeamento do Pareto-Ótimo, o conjunto que contém todos os possíveis P^*, é realizado variando λ entre zero e um. Utilizando a idéia de conjunto de validação pode-se escolher λ de tal forma que o erro seja minimizado neste conjunto. Em [Tei01] foi mostrado que esta escolha é ótima dado

[2]Seja a matriz $A \in \mathbb{R}^{n\times n}$ e os vetores $x \in \mathbb{R}^n$ e $y \in \mathbb{R}^n$. As seguintes relações são verdadeiras:

$$\frac{\partial(x^Ty)}{\partial y} = x;$$

$$\frac{\partial(y^Tx)}{\partial y} = x;$$

$$\frac{\partial(x^TAx)}{\partial x} = (A + A^T)x.$$

um conjunto estatisticamente representativo e a função a ser aproximada esteja contaminada por um ruído Gaussiano. De fato pode-se observar que o erro de validação é uma função quadrática, sendo que, para esta classe de problemas, existe uma única solução dentro do conjunto Pareto, onde o erro é mínimo. Neste trabalho λ ótimo, λ^*, será calculado utilizando o método da seção áurea.

5.2 Experimentos utilizando o aproximador polinomial de gradiente mínimo

Nesta seção serão apresentados alguns resultados utilizando o aproximador descrito na Equação 6.39, onde o decisor atuará considerando a minimização do erro de validação. Considere o seguinte problema de aproximar uma reta contaminada por um ruído Gaussiano com variância igual à 1^2

$$f_{P1}(x) = 2x + ruido, \tag{5.11}$$

por um polinômio de ordem 10.

Na Figura 5.1 são mostrados os pontos de treinamento, validação, a função desejada e a função aproximada. Nas Figuras 5.2 e 5.3 são mostrados respectivamente os erros de treinamento e validação. Como era de se esperar o erro de treinamento tem um comportamento monotônico decrescente. O erro de validação apresenta um ponto onde a função tem um mínimo, e este ponto representa a função escolhida.

A fronteira Pareto-Ótima (PO) é mostrada na Figura 5.4, sendo esta convexa, como previsto. No capítulo anterior foi apontado que, esta formulação é equivalente, dado algumas condições de integrabilidade da transformada de Fourier, à minimização da energia da saída de um filtro passa-altas. Na Figura 5.5 é mostrada a energia na saída deste filtro, e nota-se que esta é monotônica crescente, como previsto.

O segundo exemplo testado trata o problema de aproximar um polinômio de ordem 3 por um de ordem 10, contaminado por um ruído Gaussiano. Matematicamente:

$$f_{P3}(x) = 0,2x^3 + x^2 + x + ruido. \tag{5.12}$$

Resultados similares aos obtidos para o caso onde se desejava aproximar uma reta, foram obtidos para o problema descrito na Equação 5.12.

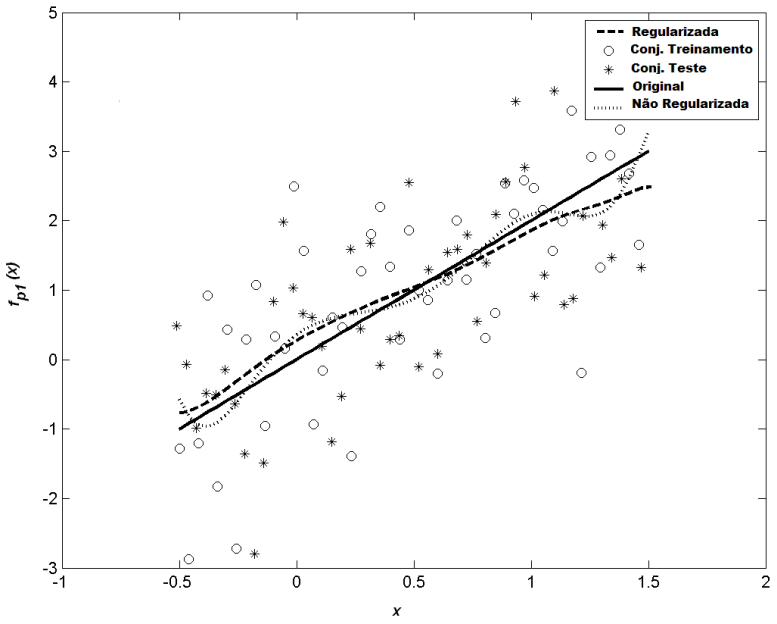

Figura 5.1: Problema de aproximação de uma reta contaminada por um ruído Gaussiano. Pode-se observar que a função aproximada com a regularização é mais próxima da original, e aparentemente mais "suave".

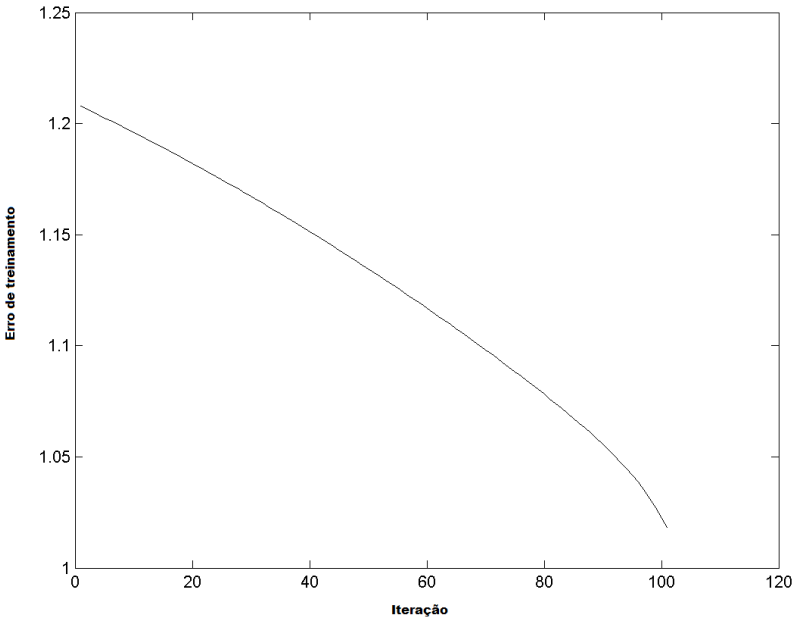

Figura 5.2: Erro de treinamento para f_{P1}. Este é em geral monotônico decrescente, caso não ocorram problemas numéricos.

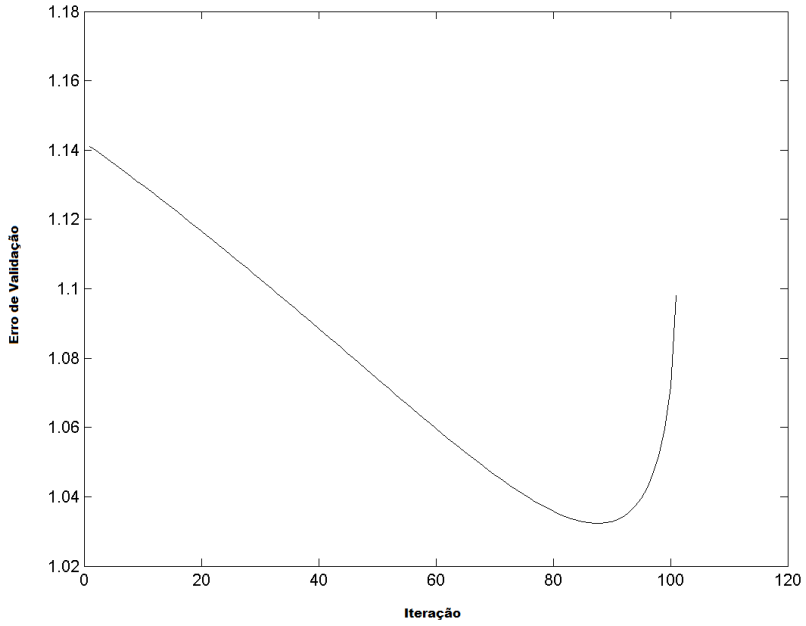

Figura 5.3: Erro de validação para o problema f_{P1}. É interessante notar que o erro de validação tem um mínimo bem definido, como mostrado na figura. Esta situação é geral pois esta é uma função quadrática nos parâmetros (salvo casos onde ocorram problemas numéricos).

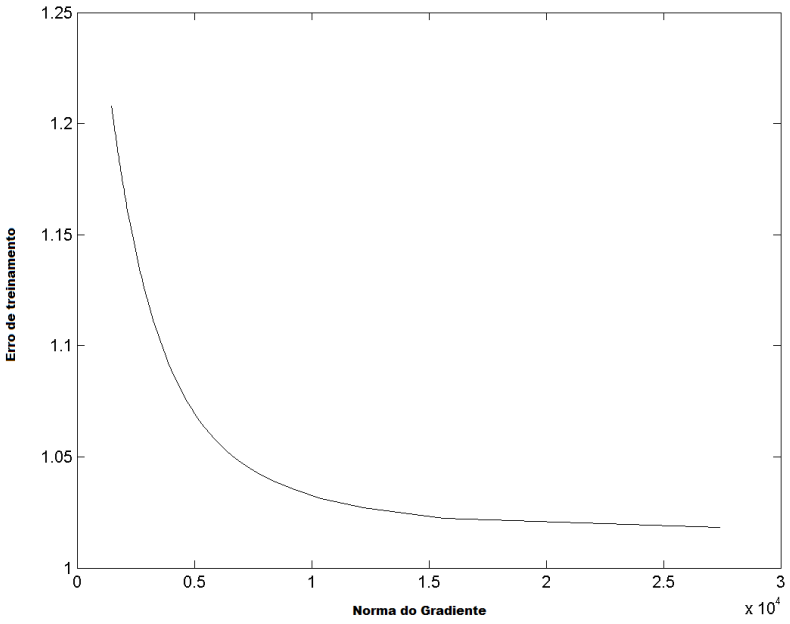

Figura 5.4: Fronteira Pareto-Ótima (PO) para o problema definido na Equação 5.11.

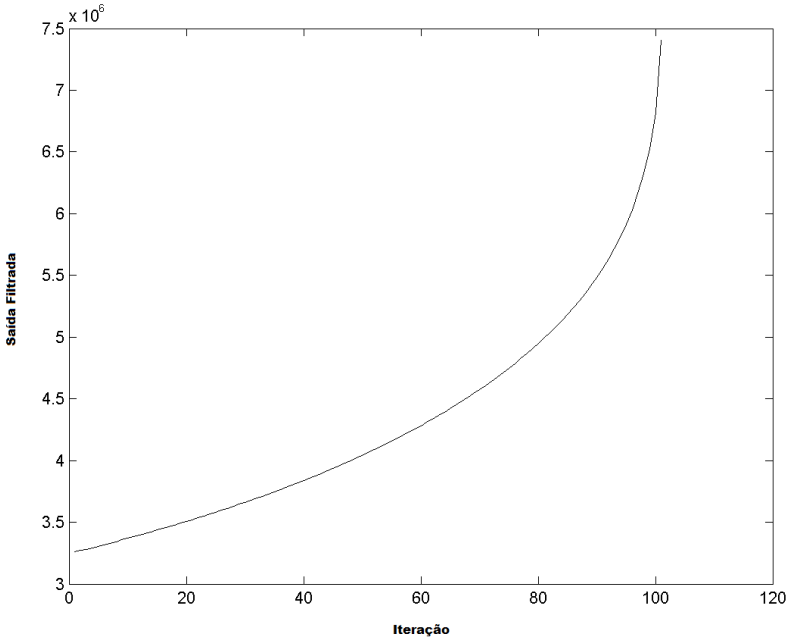

Figura 5.5: Energia da saída do filtro passa-altas equivalente à minimização do gradiente sendo esta uma função monotônica crescente, como esperado teoricamente.

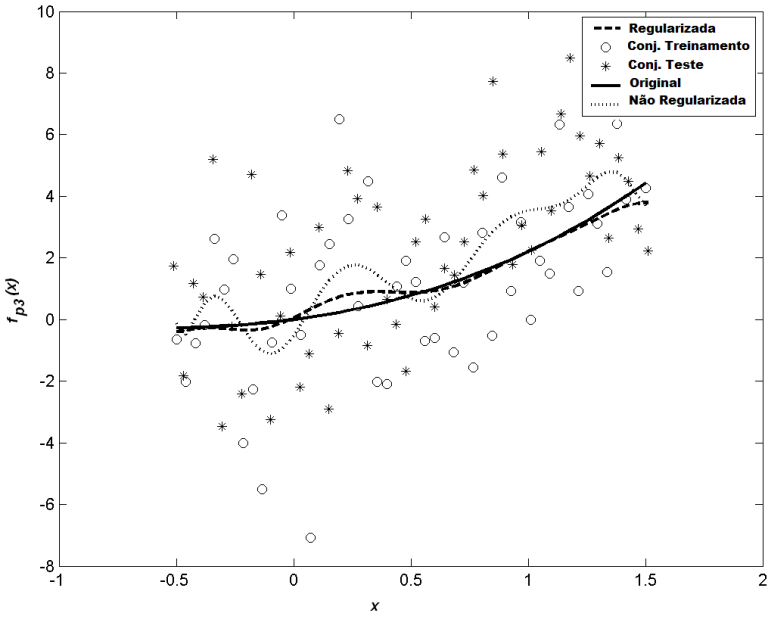

Figura 5.6: Problema de aproximação de um polinômio de terceiro grau contaminado por um ruído Gaussiano. Pode-se observar que a função aproximada com a regularização é mais próxima da original, e aparentemente mais "suave".

Na Figura 5.6 são mostrados os pontos de treinamento, validação, a função desejada e a função aproximada. Nas Figuras 5.7 e 5.8 são mostrados respectivamente os erros de treinamento e validação. O erro de validação apresenta um ponto de mínimo, como no caso anterior. A fronteira Pareto-Ótima (PO) é mostrada na Figura 5.9, sendo esta convexa, como previsto anteriormente. Na Figura 5.10 é mostrado a energia na saída do filtro passa-altas, e observa-se que esta é monotônica crescente, como previsto.

5.3 Discussão

Neste capítulo foi apresentado e analisado o aproximador polinomial de gradiente mínimo. A escolha por iniciar as análises por este aproximador foi devido ao fato de se tratar de um sistema bastante simples, sendo que a

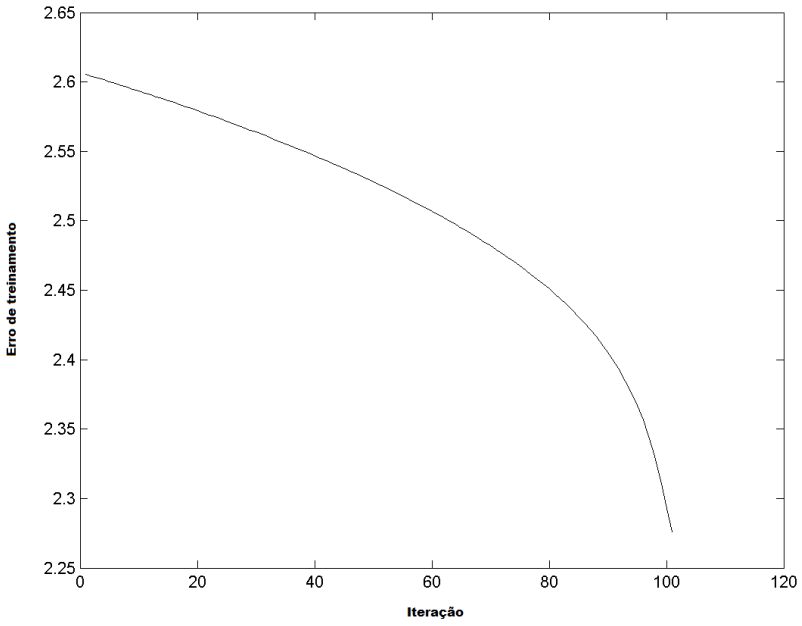

Figura 5.7: Erro de treinamento para f_{P3}. Este é em geral monotônico decrescente, caso não ocorram problemas numéricos.

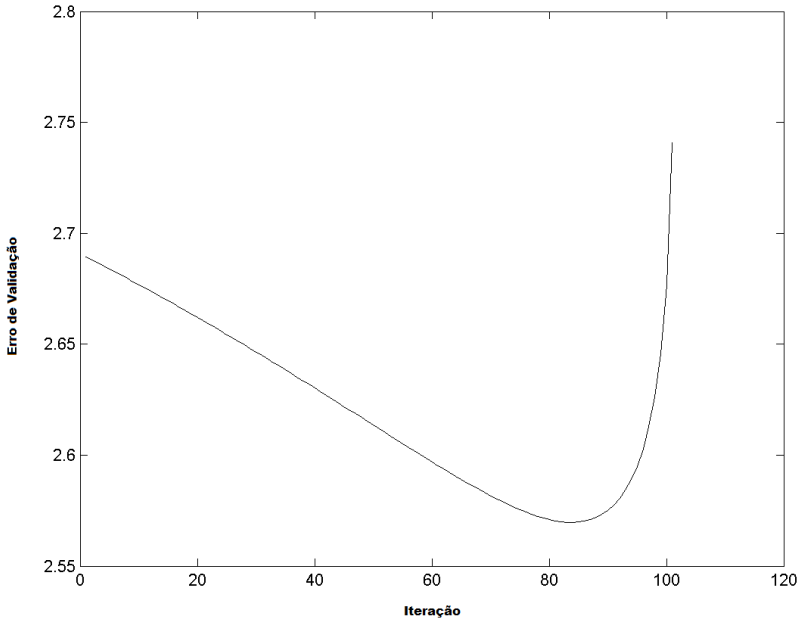

Figura 5.8: Erro de validação para o problema f_{P3}. É interessante notar que o erro de validação tem um mínimo bem definido, como mostrado na figura. Esta situação é geral pois esta é uma função quadrática nos parâmetros (salvo os casos onde ocorram problemas numéricos).

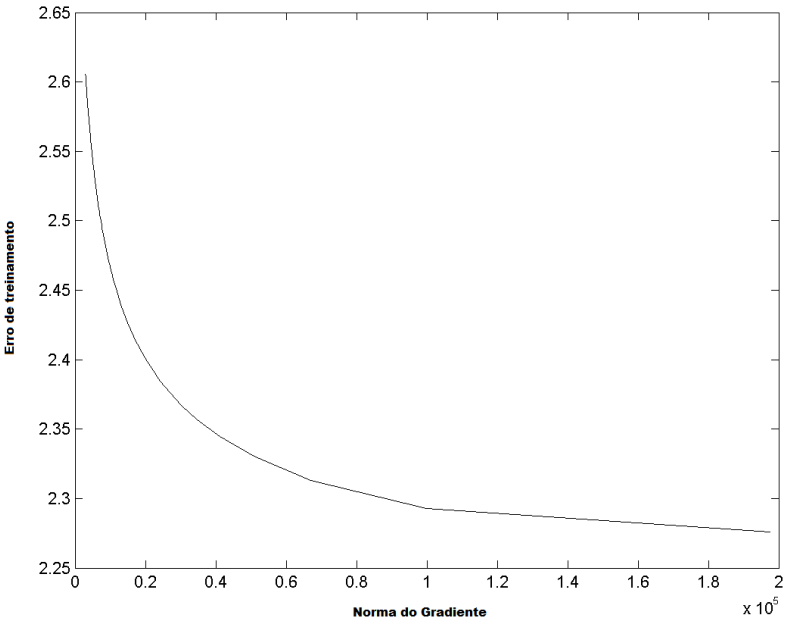

Figura 5.9: Fronteira Pareto-Ótima (PO) para o problema definido na Equação 5.12.

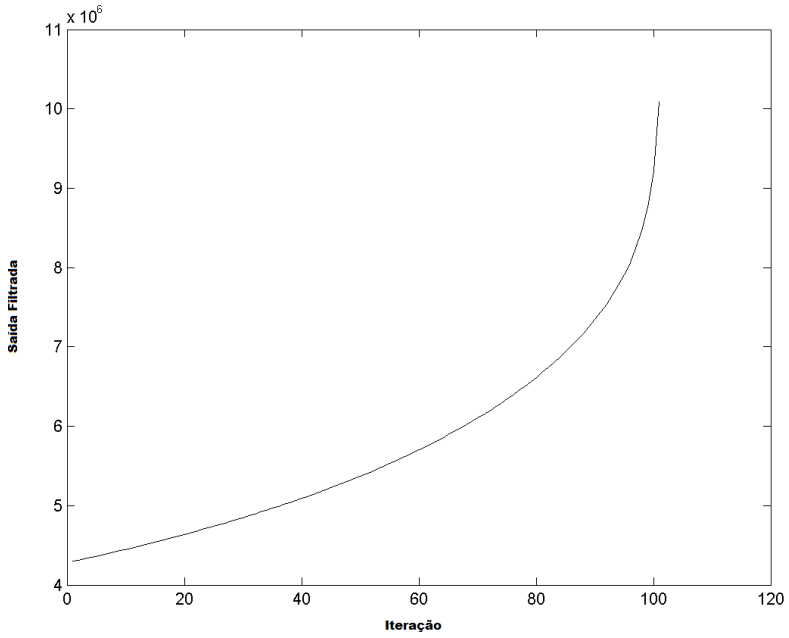

Figura 5.10: Energia da saída do filtro passa-altas equivalente à minimização do gradiente. Pode-se observar que trata-se de uma função monotônica crescente, como esperado teoricamente.

compreensão de alguns fenômenos ficam mais claros. Pode-se observar que para esta classe de aproximadores a derivada da função tem a transformada de Fourier integrável, fazendo com que a metodologia proposta, em geral, funcione como uma filtragem das altas freqüências. Este fato foi confirmado empiricamente utilizando a FFT da função aproximada. Entretanto é importante lembrar que esta classe de aproximadores sofre de problemas de condicionamento e do "mal da dimensionalidade". De fato, como é bem conhecido, a utilização deste tipo de aproximador para problemas de alta dimensão é proibitivo devido à quantidade de parâmetros livres requeridos.

Para concluir observa-se que utilizar aproximadores polinomiais é interessante devido ao fato de estarem envolvidas somente funções quadráticas, mas alguns problemas numéricos e o mal da dimensionalidade, impedem que estes sejam de aplicação mais geral. Considere a expansão polinomial para um mapeamento em $x \in \mathbb{R}^m$[GWW61]:

$$f(x) = a_0 + \sum_{i=1}^{m} a_i x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij} x_i x_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} a_{ijk} x_i x_j x_k ... \quad (5.13)$$

Dada a Eq. 5.13, pode-se notar que o número de parâmetros livres a para um polinômio de ordem τ é $O(m,\tau) = \sum_{i=1}^{\tau} m^{i-1}$. Por exemplo, um problema com 3 dimensões, $m = 3$, utilizando uma série de ordem 3, $\tau = 3$, terá $O(3,3) = 13$ parâmetros livres, com $m = 10$ e $\tau = 10$, $O(10,10) > 1.1e{+}9$!. Observe que para um problema simples com 10 dimensões utilizando um polinômio de ordem 10 o número de parâmetros livres explode, este é o famoso mal da dimensionalidade.

Na próxima seção será apresentada a rede perceptron com camadas paralelas (PLP - Parallel Layer Perceptron) e sua forma de regularização por gradiente mínimo que aproveita as boas características dos aproximadores polinomiais e tenta resolver os problemas citados.

Capítulo 6

Rede perceptron com camadas paralelas (PLP-parallel layer perceptron)

Nesta seção será discutida a principal contribuição do trabalho desenvolvido. Esta contribuição trata-se do desenvolvimento de uma nova topologia de redes neurais denominada rede perceptron com camadas paralelas (PLP-Parallel Layer Perceptron)[CVV03]. Vários artigos já foram publicados em revistas derivados deste trabalho [CVV03], [VCV04], [VVCV05], [VCV06] e alguns outros se encontram submetidos como indicado no Capítulo 1 deste trabalho.

6.1 Topologia

A saída y_t da rede perceptron com camadas paralelas (PLP) considerando n entradas e m perceptrons por camada é calculada como:

$$y_t = \beta\left(\left\{\sum_{j=1}^{m}\left[\gamma_j\left(a_{jt}\right)\phi_j\left(b_{jt}\right)\right]\right\}\right) = \beta(\psi_t) \tag{6.1}$$

Onde:

$$a_{jt} = \sum_{i=1}^{n+1} p_{ji}x_{it} \tag{6.2}$$

$$b_{jt} = \sum_{i=1}^{n+1} v_{ji} x_{it} \tag{6.3}$$

Geralmente não se define uma função γ ou ϕ para cada neurônio j e sim funções γ e ϕ comuns a todos eles. Por isto será utilizado de agora em diante, para simplificar a apresentação, $\gamma_j(.) = \gamma(.)$ e $\phi_j(.) = \phi(.)$, onde $\beta(.)$, $\gamma(.)$ e $\phi(.)$ são funções de ativação (tangente hiperbólica, Gaussiana, linear, etc...), v_{ji} e p_{ji} são os componentes das matrizes de pesos P e V, x_{it} é a i-ésima entrada para o t-ésimo padrão, onde x_{0t} é a polarização do perceptron, e y_t é a t-ésima posição do vetor de saída y. Na Figura 6.1 é mostrada a topologia da PLP.

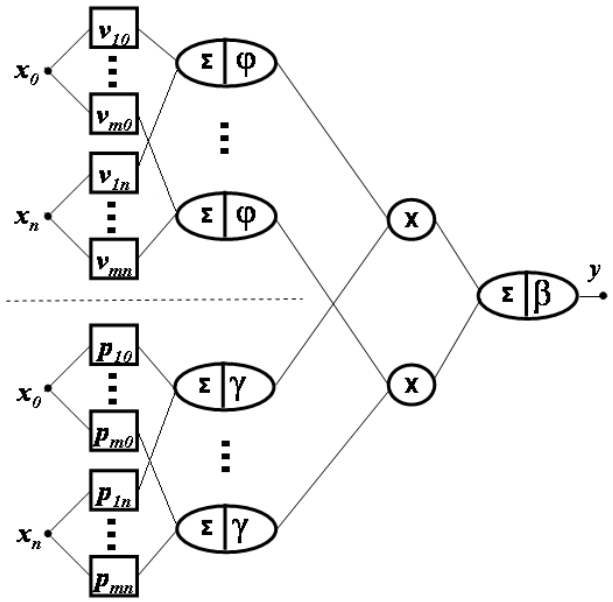

Figura 6.1: Arquitetura de rede perceptron com camadas em paralelo.

Similarmente às tradicionais MLPs, todos os parâmetros da rede podem ser ajustados utilizando o método do gradiente. Entretanto algumas diferenças devem ser consideradas. Primeiramente, no caso da MLP, a rede utiliza função de funções para o mapeamento entrada-saída. A PLP utiliza basicamente o produto de funções. E mais, como pode ser visto na Figura 6.1, a topologia proposta é composta por camadas paralelas. Esta característica simplifica a implementação em máquinas paralelas ou clusters. Um caso par-

ticular da topologia mostrada na Figura 6.1 é gerado considerando $\gamma(.)$ e $\beta(.)$ como funções identidade. Neste caso a saída da rede é calculada como:

$$y_t = \sum_{j=1}^{m}[a_{jt}\phi(b_{jt})] = \sum_{j=1}^{m}[L_{jt}N_{jt}], \tag{6.4}$$

onde $L_{jt} = a_{jt}$ e $N_{jt} = \phi(b_{jt})$. Para um conjunto de pontos y_t, $t = 1, ..., T$, definimos de forma similar os vetores L_j e N_j.

É importante notar que, o caso particular descrito na Equação 6.4 tem algumas características interessantes [CVV03]. A superfície de erro em relação a p_{ji}, que para este caso particular é um parâmetro linear, é uma estrutura quadrática (mono-modal), logo um algoritmo de treinamento mais eficiente pode ser utilizado. Para este caso pode-se utilizar o estimador de mínimos quadrados (LSE) para adaptar os parâmetros lineares da rede, sendo que a convergência para o mínimo local acontece em uma iteração.

6.2 Teorema da aproximação universal

Considere o teorema da aproximação universal escrito como [Cyb89, Fun89]:

Considere $\phi(.)$ uma função contínua, não constante, limitada e monotonicamente crescente. Deixe Ip denotar o hiper cubo unitário p-dimensional. O espaço de funções contínuas em Ip é denotado por $C(Ip)$. Logo, dada qualquer função pertencente a $C(Ip)$ e $\epsilon > 0$, existe um inteiro M e um conjunto de constantes reais α_i, w_{ij} e b_i , onde $i = 1, ..., M$ e $j = 1, ..., p$,de tal forma que pode-se definir:

$$\tilde{f}(x_1, ..., x_p) = \sum_{i=1}^{M} \alpha_i \phi \left(\sum_{j=1}^{p} w_{ij}x_j + b_i \right) \tag{6.5}$$

como uma aproximação da função $f(.)$, i.e., $|\tilde{f}(x_1, ..., x_p) - f(x_1, ..., x_p)| < \epsilon$ para todo $\{x_1, ..., x_p\} \in I_p$.

Este teorema é o utilizado para a prova que uma MLP é um aproximador universal. Por inspeção pode-se escrever a saída do caso particular da PLP, como mostrado na Equação 6.4, onde α é função de x. Desta forma pode-se concluir que a PLP é um aproximador universal e que uma MLP é um caso particular da mesma. A mesma discussão é válida para a situação onde funções radiais são utilizadas na camada não linear, de tal forma que uma

rede RBF também é um caso particular da PLP, sendo que o teorema de aproximação universal desenvolvido para a primeira pode ser utilizado para a segunda.

Uma interpretação gráfica pode ser dada à capacidade de aproximação da PLP. Esta rede pode ser entendida como uma combinação linear de funções não-lineares. Uma rede com um par de camadas paralelas, e dois perceptrons por camada, $m = 2$, é capaz de aproximar dois períodos de uma função senoidal. Na Figura 6.2 são mostradas as funções lineares, $L1$ e $L2$, geradas pelo perceptron linear e na Figura 6.3 é mostrada a saída da camada não-linear, $N1$ e $N2$, considerando $\phi(.)$ como uma função Gaussiana. O produto entre as saídas lineares e não-lineares é mostrado na Figura 6.4 e na Figura 6.5 é mostrada a aproximação resultante. Resultados similares podem ser obtidos para outros tipos de função de ativação. Os resultados mostrados nesta seção estão publicados em [CVV03].

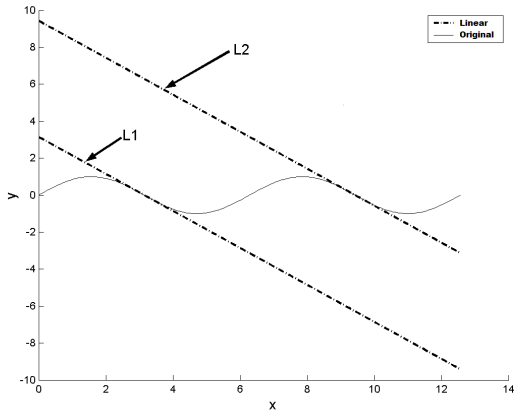

Figura 6.2: Parâmetros lineares

6.3 Algoritmos para a minimização do risco empírico

Nesta seção serão discutidos alguns possíveis algoritmos de treinamento para a PLP. Os métodos propostos nesta seção já se encontram devidamente implementados e testados.

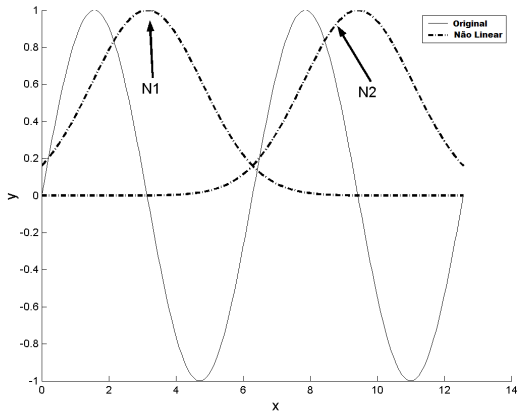

Figura 6.3: Parâmetros não lineares

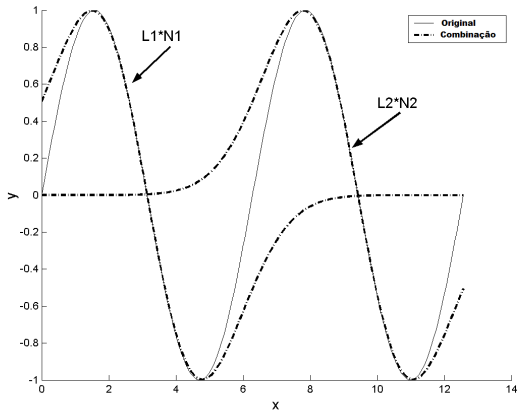

Figura 6.4: Produto entre os parâmetros lineares e não lineares

Os métodos descritos se baseiam na informação do gradiente da função de erro. A função de erro pode ser escrita como:

$$E(w) = \frac{1}{2} \sum_{t=1}^{T} e_t(w, x_t, yd_t)^2. \tag{6.6}$$

Onde:

$$e_t(w, x_t, yd_t) = (y_t(w, x_t) - yd_t). \tag{6.7}$$

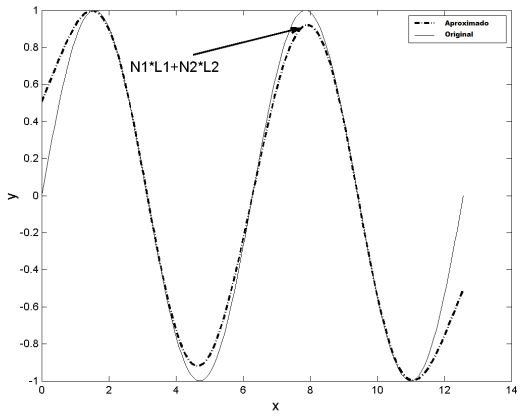

Figura 6.5: Função aproximada

Na Eq. 6.6 $E(.)$ representa também uma medida de risco empírico, t representa o t-ésimo padrão, sendo este composto pelas coordenadas x_t e a saída desejada yd_t. A saída da rede é y_t. As épocas (iterações) dos algoritmos de treinamento serão definidas utilizando o índice k. A variável w pode representar um vetor ou uma matriz de pesos, que para este caso é composto dos pesos P e V, onde a representação dependerá da simplicidade para a explicação.

6.3.1 Gradiente

Para se utilizar os métodos baseados no gradiente, tem-se que calcular o gradiente da Equação 6.7 em relação aos pesos, utilizando a topologia descrita na Equação 6.1:

$$\frac{\partial e_t}{\partial p_{ji}} = \frac{\partial y_t}{\partial p_{ji}} = \frac{\partial \beta}{\partial \psi_t}\frac{\partial \psi_t}{\partial \gamma}\frac{\partial \gamma}{\partial a_{jt}}\frac{\partial a_{jt}}{\partial p_{ji}} = \frac{\partial \beta}{\partial \psi_t}\frac{\partial \gamma}{\partial a_{jt}}\phi(b_{jt})x_{it}. \tag{6.8}$$

$$\frac{\partial e_t}{\partial v_{ji}} = \frac{\partial y_t}{\partial v_{ji}} = \frac{\partial \beta}{\partial \psi_t}\frac{\partial \psi_t}{\partial \phi}\frac{\partial \phi}{\partial b_{jt}}\frac{\partial b_{jt}}{\partial v_{ji}} = \frac{\partial \beta}{\partial \psi_t}\frac{\partial \phi}{\partial b_{jt}}\gamma(a_{jt})x_{it}. \tag{6.9}$$

Para o caso particular descrito na Equação 6.4, obtém-se:

$$\frac{\partial e_t}{\partial p_{ji}} = \frac{\partial y_t}{\partial p_{ji}} = \frac{\partial y_t}{\partial a_{jt}}\frac{\partial a_{jt}}{\partial p_{ji}} = \phi(b_{jt})x_{it}. \tag{6.10}$$

$$\frac{\partial e_t}{\partial v_{ji}} = \frac{\partial y_t}{\partial v_{ji}} = \frac{\partial y_t}{\partial \phi}\frac{\partial \phi}{\partial b_{jt}}\frac{\partial b_{jt}}{\partial v_{ji}} = a_{jt}\frac{\partial \phi}{\partial b_{jt}}x_{it}. \tag{6.11}$$

Logo:

$$\frac{\partial E}{\partial p_{ji}} = \sum_{t=1}^{T} e_t \frac{\partial e_t}{\partial p_{ji}} \tag{6.12}$$

$$\frac{\partial E}{\partial v_{ji}} = \sum_{t=1}^{T} e_t \frac{\partial e_t}{\partial v_{ji}} \tag{6.13}$$

Utilizando o método do gradiente pode-se escrever:

$$p_{ji(k+1)} = p_{ji(k)} - \eta \frac{\partial E}{\partial p_{ji(k)}}. \tag{6.14}$$

$$v_{ji(k+1)} = v_{ji(k)} - \eta \frac{\partial E}{\partial v_{ji(k)}}. \tag{6.15}$$

Onde k representa a k-ésima iteração, como informado anteriormente, e η a taxa de aprendizado (passo do gradiente). Na implementação feita η foi calculado da seguinte maneira:

$$\eta = \triangle / \|\nabla w\|. \tag{6.16}$$

onde $\triangle$ é o tamanho do passo, usualmente $\triangle = 0.01$, e $\|\nabla w\|$ é a norma do gradiente de V ou P dependendo de qual peso está sendo ajustado.

Uma regra empírica foi desenvolvida de forma a atualizar dinamicamente o passo. Considere:

- Se o erro decrescer em quatro iterações consecutivas, o passo é aumentado, $\triangle_{k+1} = \triangle_k * c1$, onde $c1 = 1.1$.
- Caso contrário o passo é diminuído, $\triangle_{k+1} = \triangle_k * c2$, onde $c2 = 0.9$.

Os valores destas constantes, como o passo inicial, foram determinados de forma empírica, sendo que o algoritmo não é muito sensível a estas, sendo definidas geralmente nos intervalos, $c1 = [1.1\ 1.3]$ e $c2 = [0.7\ 0.9]$. Nas seções seguintes são descritos outros métodos de otimização mais robustos de forma que o processo de treinamento seja o mais eficiente possível, i.e., métodos de otimização que utilizam algumas características da topologia para alcançar de forma mais eficiente um mínimo local. Para as próximas situações serão descritos métodos onde a característica da função de erro para o caso particular é explorado, utilizando a atualização por mínimos quadrados do parâmetros P.

6.3.2 Híbrido - Gradiente e mínimos quadrados

Neste treinamento é proposto utilizar a atualização dos termos não-lineares de acordo com a Equação 6.15 e a atualização dos pesos da camada linear com o método dos mínimos quadrados (LSE-least square estimator).

Como a saída y_t é uma função linear dos parâmetros p_{ji}, seus valores ótimos podem ser calculados utilizando um algoritmo simples baseado em álgebra linear. Esta característica é similar à dos conseqüentes de uma ANFIS [Jan93]. Para simplificar a explicação sobre LSE, considere $l_z = p_{ji}$, onde $z = (n+1)(j-1)+i$, l é a transformação da matriz P para um vetor l com os mesmos componentes, onde $j = 1, ..., m$, $i = 1, \ldots, n+1$ e n é a dimensão do espaço de variáveis,

$$l = \left[\begin{array}{cccccccc} p_{11} & \cdots & p_{1(n+1)} & p_{21} & \cdots & p_{2(n+1)} & p_{m1} & \cdots & p_{m(n+1)} \end{array} \right]^T. \quad (6.17)$$

Primeiramente todas as saídas dos perceptrons não-lineares são calculadas. Uma matriz C, que é uma combinação das saídas não lineares com as entradas é gerada. Os componentes c_{tz} da matriz C são $c_{tz} = x_{it}\phi(b_{jt})$, onde $t = 1, ..., T$. Matematicamente:

$$C = \left[\begin{array}{ccc} x_{11}\phi(b_{11}) & \cdots & x_{(n+1)1}\phi(b_{m1}) \\ \vdots & \cdots & \vdots \\ x_{1T}\phi(b_{1T}) & \cdots & x_{(n+1)T}\phi(b_{mT}) \end{array} \right]. \quad (6.18)$$

Pode-se então escrever uma forma equivalente à Equação 6.4 utilizando uma notação matricial. Matematicamente:

$$y = Cl. \quad (6.19)$$

Como já dito, o objetivo do processo de aprendizagem para a maioria dos algoritmos de treinamento supervisionado é a minimização do somatório do erro quadrático (MSE- Mean squarred error) para o conjunto de treinamento:

$$E = \frac{1}{2}(Cl - yd)^T(Cl - yd). \quad (6.20)$$

O valor ótimo para l é obtido utilizando a condição na qual o gradiente da Equação 6.20 é zero. Similarmente ao caso dos polinômios pode-se obter,

$$l^* = (C^T C)^{-1} C^T yd. \quad (6.21)$$

Depois do cálculo de l^*, estas componentes são retornadas para a forma matricial P. A saída é calculada e a matriz V é atualizada de acordo com a Equação 6.15.

6.3.3 Híbrido - Levenberg-Maquardt e mínimos quadrados

Um dos problemas apresentados para a atualização dos pesos utilizando o método do gradiente é que este pode sofrer problemas para superfícies de erro de alta excentricidade. Uma forma de melhorar este seria utilizar uma correção de segunda ordem na direção do gradiente.

Uma maneira de melhorar o desempenho do treinamento é utilizar o algoritmo de Levenberg-Marquardt [HM94], como é feito para as MLPs. Este método realiza uma aproximação do método Newton, que realiza a correção de segunda ordem. Caso o método de Newton fosse utilizado, a forma de atualização ou correção dos pesos seria:

$$\Delta V = -[\nabla^2 E]^{-1} \nabla E. \tag{6.22}$$

onde ∇E é o gradiente e $[\nabla^2 E]^{-1}$ é a inversa da matriz Hessiana da Equação 6.6. Este método converge em um passo para funções quadráticas[1]. Para o caso de funções não quadráticas pode-se fazer a atualização como:

$$\Delta V = -\eta [\nabla^2 E]^{-1} \nabla E. \tag{6.23}$$

onde η é o passo ao longo da direção de descida. Considerando a Equação 6.6 pode-se mostrar que:

$$\nabla E(V) = J^T(V) e(V). \tag{6.24}$$

$$\nabla^2 E(V) = J^T(V) J(V) + S(V) \tag{6.25}$$

onde $J(V)$ é a matriz Jacobiana e $S(V)$ é um resíduo. O Jacobiano $J(V)$ é mostrado na Equação 6.26.

$$J(V) = \begin{bmatrix} \frac{\partial e_1(V)}{\partial v_1} & \cdots & \frac{\partial e_1(V)}{\partial v_{(n+1)*m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_T(V)}{\partial v_1} & \cdots & \frac{\partial e_T(V)}{\partial v_{(n+1)*m}} \end{bmatrix}. \tag{6.26}$$

Observe que foi considerado para a montagem desta matriz a forma vetorial de V (não matricial). Para o método de Gauss-Newton uma aproximação

[1] Observe que o método dos mínimos quadrados é um caso particular do método de Newton. Este considera como posição inicial, o centro do sistema de coordenadas, e utiliza a direção do gradiente corrigida com informações de segunda ordem com o passo unitário.

linear é assumida de forma que $S(V) \approx 0$. A forma de atualização pode então ser descrita como:

$$\Delta V = [J^T(V)J(V)]^{-1}J^T(V)e(V). \tag{6.27}$$

O método de Levenberg-Marquardt, que é uma modificação do método de Gauss-Newton, é escrito como:

$$\Delta V = [J^T(V)J(V) + \mu I]^{-1}J^T(V)e(V). \tag{6.28}$$

O parâmetro μ controla o tamanho do passo do método. Este método tem taxa de convergência elevada pois utiliza informações de ordem mais elevada das funções.

6.4 Minimização do risco estrutural da rede PLP utilizando a idéia do gradiente mínimo

Como discutido anteriormente neste trabalho, o princípio da minimização do risco empírico não é o melhor para treinar uma máquina. Foi mostrado no capítulo 2 que a minimização do risco estrutural é o mais adequado. Neste trabalho foi proposto a utilização da norma do gradiente como medida de complexidade para a função aproximada. O seguinte problema bi-objetivo foi proposto para definir uma técnica de minimização do risco estrutural:

$$min \begin{cases} f_1 = & R_{emp}(w) \\ f_2 = & \sum_t^N \|\nabla f(x_t)\|^2 \end{cases} \tag{6.29}$$

Esta técnica apresenta algumas propriedades interessantes, como mostrado no capítulo anterior para aproximadores polinomiais. Nesta seção a mesma estratégia desenvolvida para os polinômios será aplicada ao caso particular da PLP em dois passos:

1. Treine uma rede PLP super dimensionada de forma a minimizar o risco empírico, utilizando um dos algoritmos descritos acima.

2. Aplique a técnica de minimização da norma do gradiente nos parâmetros lineares (camada polinomial da rede) utilizando a formulação quadrática apresentada no capítulo anterior.

Os passos descritos acima equivalem a gerar uma rede PLP que seja capaz de mapear os ruídos das amostras, e aplicar a regularização nos parâmetros lineares para filtrar estes. A escolha por utilizar a regularização somente nos parâmetros lineares é devido ao fato que estes apresentam funcionais quadráticos, como descritos no capítulo anterior, sendo que algoritmos de treinamento eficientes podem ser gerados.

Primeiramente deve-se definir f_1 para a PLP em relação aos parâmetros lineares. Este pode ser escrito, como já mostrado na Equação 6.20, como:

$$f_1 = (Cl - yd)^T (Cl - yd) \tag{6.30}$$

onde l e C estão definidos respectivamente nas Equações 6.17 e 6.18. O segundo passo necessário é calcular o vetor gradiente da saída da rede. Calculando a derivada primeira da Equação 6.4 em relação à variável de entrada x_i, resulta na seguinte equação [VCV04]:

$$\frac{\partial y_t}{\partial x_{it}} = \sum_{j=1}^{m} \left[\left(\frac{\partial \phi}{\partial b_{jt}} v_{ji} \right) \left(\sum_{i=1}^{n+1} p_{ji} x_{it} \right) + \phi \left(\sum_{i=1}^{n+1} v_{ji} x_{it} \right) p_{ji} \right]. \tag{6.31}$$

Logo o gradiente da saída y pode ser encontrado:

$$\nabla_x y_t = \left[\frac{\partial y_t}{\partial x_{1t}} \frac{\partial y_t}{\partial x_{2t}} ... \frac{\partial y_t}{\partial x_{nt}} \right]^T. \tag{6.32}$$

A derivada parcial em relação ao vetor $x_i = [x_{i1}, ..., x_{it}, ..., x_{iT}]$ pode ser escrita de forma vetorial como:

$$\frac{\partial y}{\partial x_i} = D_i l \tag{6.33}$$

Para exemplificar a construção da matriz D_i, onde $D_i \in \mathbb{R}^{T \times (n+1)m}$ considere os seguintes casos considerando as derivadas parciais em relação a x_1 e x_2. Matematicamente:

$$D_1 = \begin{bmatrix} \frac{\partial \phi}{\partial b_{11}} v_{11} x_{11} + \phi \left(\sum_{i=1}^{n+1} v_{1i} x_{i1} \right) & \cdots & \frac{\partial \phi}{\partial b_{m1}} v_{m1} x_{(n+1)1} \\ \vdots & \cdots & \vdots \\ \frac{\partial \phi}{\partial b_{1T}} v_{1T} x_{1T} + \phi \left(\sum_{i=1}^{n+1} v_{1i} x_{iT} \right) & \cdots & \frac{\partial \phi}{\partial b_{m1}} v_{m1} x_{(n+1)T} \end{bmatrix}. \tag{6.34}$$

$$D_2 = \begin{bmatrix} \frac{\partial \phi}{\partial b_{11}} v_{12} x_{11} & \frac{\partial \phi}{\partial b_{11}} v_{12} x_{21} + \phi \left(\sum_{i=1}^{n+1} v_{1i} x_{i1} \right) \\ \vdots & \cdots \\ \frac{\partial \phi}{\partial b_{1T}} v_{12} x_{1T} & \frac{\partial \phi}{\partial b_{1T}} v_{12} x_{2T} + \phi \left(\sum_{i=1}^{n+1} v_{1i} x_{iT} \right) \end{bmatrix}$$

$$\begin{bmatrix} \cdots & \frac{\partial \phi}{\partial b_{m1}} v_{m2} x_{(n+1)1} \\ \cdots & \vdots \\ \cdots & \frac{\partial \phi}{\partial b_{mT}} v_{m2} x_{(n+1)T} \end{bmatrix}. \tag{6.35}$$

Nas matrizes D_i, $i = k$ as colunas referentes aos pesos p_{jk} são compostas por dois termos no somatório, como pode ser observado na coluna 1 de D_1 e na coluna 2 de D_2, sendo que no restante das colunas somente um termo aparece no somatório, como pode ser observado nas colunas 1 e $m(n+1)$ da matriz D_2.

Considerando a definição das matrizes D_i, a função f_2, que computa a norma do gradiente, pode ser definida como:

$$f_2 = \Omega = \sum_{k=1}^{n} (D_k l)^T (D_k l) = l^T Q l, \tag{6.36}$$

onde $Q \in \mathbb{R}^{(n+1)m \times (n+1)m} = \sum_{k=1}^{n} D_k^T D_k$, tendo a formulação do tipo norma-Q. Uma matriz real simétrica A é positiva se e somente se existe uma matriz não singular M tal que $A = M^T M$. Logo, Q é semi-positiva definida para todas as situações possíveis, lembrando que a soma de matrizes positivas é uma matriz positiva[2]. Neste caso, a função definida na Eq. 6.30 é quasi-convexa. Nas Eqs. 6.30 e 6.36, os dois objetivos definidos na Eq. 6.29 - minimização do risco empírico e da complexidade - são escritas como funções quadráticas de l. Logo a solução do problema multiobjetivo no qual se deseja minimizar o erro e a derivada da função nos pontos de treinamento, pode ser escrita como:

$$min\ f = \lambda f_1 + (1 - \lambda) f_2$$

[2]Uma matriz é semi-positiva definida se $x^T Q x \geq 0$ para qualquer vetor x.

$$= \lambda(Cl - yd)^T(Cl - yd) + (1 - \lambda)l^T Ql \tag{6.37}$$

Onde o P ótimo é calculado diferenciando a Equação 6.37, e igualando a zero o resultado e resolvendo a equação resultante. A derivada da Equação 6.37 em relação aos parâmetros P pode ser calculada como[3]:

$$\frac{df}{dl} = \lambda\frac{df_1}{dl} + (1 - \lambda)\frac{df_2}{dl}$$

$$= \lambda[(-yd^T C)^T - C^T yd + (C^T C + C^T C)l] + (1 - \lambda)(Q + Q)l$$

$$= \lambda(-2C^T yd + 2C^T Cl) + (1 - \lambda)2Ql \tag{6.38}$$

A derivação da Equação 6.38 é similar à derivação do caso polinomial[4]. Para encontrarmos o l ótimo, l^*, basta igualar a Equação 6.38 a zero. Matematicamente:

$$\lambda(-2C^T yd + 2C^T Cl) + (1 - \lambda)2Ql = 0$$

$$-2\lambda C^T yd + 2\lambda C^T Cl + (1 - \lambda)2Ql = 0$$

$$[\lambda C^T C + (1 - \lambda)Q]l = \lambda C^T yd$$

[3]Lembrando que:

$$\frac{\partial(u + v)}{\partial x} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial x}.$$

[4]Seja a matriz $A \in \mathbb{R}^{nxn}$ e os vetores $x \in \mathbb{R}^n$ e $y \in \mathbb{R}^n$. As seguintes relações são verdadeiras:

$$\frac{\partial(x^T y)}{\partial y} = x;$$

$$\frac{\partial(y^T x)}{\partial y} = x;$$

$$\frac{\partial(x^T Ax)}{\partial x} = (A + A^T)x.$$

$$l^* = [\lambda C^T C + (1-\lambda)Q]^{-1} \lambda C^T yd, \tag{6.39}$$

se a matriz $[\lambda C^T C + (1-\lambda)Q]$ é não singular[5]. O algoritmo escalona linearmente com o número de parâmetros livres, $(n+1)m$, o que pode fazer o treinamento ficar caro computacionalmente. Técnicas para reduzir este problema estão sendo estudadas. As técnicas padrões para diminuição do espaço de entradas podem ser aplicadas para reduzir este problema.

O mapeamento das soluções Pareto-Ótimas, o conjunto dos possíveis l^*, é realizado variando λ entre zero e um. Utilizando a idéia de conjunto de validação pode-se escolher λ de tal forma que o erro seja minimizado neste conjunto. O λ^* foi calculado utilizando o método da seção áurea.

6.5 Exemplos numéricos

6.5.1 Problema de regressão

O primeiro problema de teste é um problema de regressão, onde as redes foram testadas com 50 amostras ruidosas amostradas da função 6.40. Um ruído branco com média zero e variância de $0,2^2$ foi adicionado à função original. Os conjuntos de validação e teste foram compostos, respectivamente por 25 e 1000 padrões, sob as mesmas condições previamente mencionadas.

$$f_1(x) = \frac{(x-2)(2x+1)}{(1+x^2)} \tag{6.40}$$

Uma rede com 80 neurônios paralelos foi utilizada nesta simulação. A precisão utilizada para a seção áurea foi igual a $1e-12$, sendo que para alcançá-la são necessárias somente 60 avaliações de λ, i.e., somente 60 redes serão avaliadas. Na Figura 6.6 são mostrados o conjunto de treinamento, validação, a função desejada, e as obtidas com e sem a regularização por gradiente mínimo. Ampliações de algumas regiões da função estudada são mostradas nas Figuras 6.7 e 6.8. Pode-se observar que o resultado obtido foi bem próximo do desejado.

Os resultados encontrados foram comparados com os obtidos utilizando os seguintes algoritmos: i) MLP com 30 neurônios na camada escondida

[5]Este algoritmo de treinamento foi desenvolvido em *Matlab* [Mat]. Ao invés de inverter o sistema da Eq. (6.39) usando o comando *inv*(.), foi utilizado o comando \ que resolve o sistema iterativamente. Esta forma é mais robusta e rápida do que quando a inversão do sistema é utilizada como pode ser visto na página do *Matlab* [Mat].

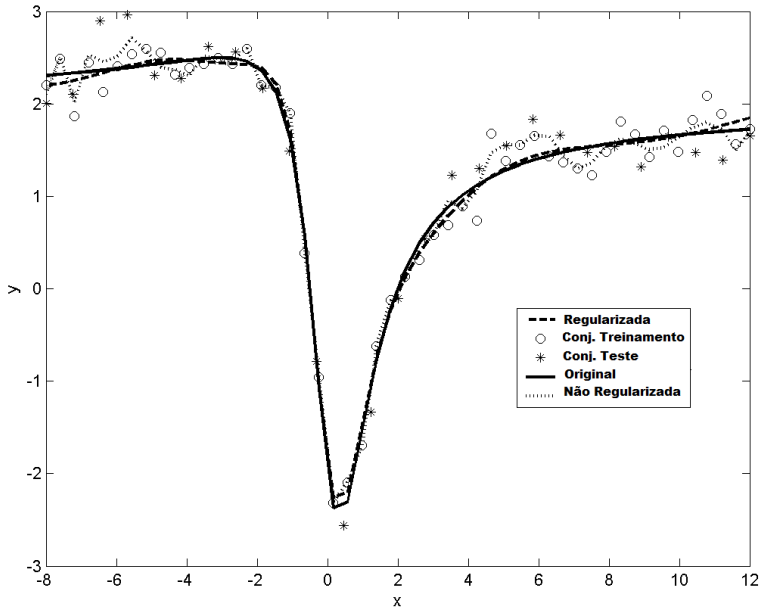

Figura 6.6: Problema de aprendizado da função f_1 contaminada por um ruído Gaussiano.

treinada com o algoritmo de Levenberg-Maquardt (LM) [HM94], ii) MLP treinada com o método do decaimento dos pesos (WD) utilizando a constante de decaimento igual à 0.9, iii) Optimal Brain Damage (OBD) com 30 neurônios, iv) Early Stop (ES) com 30 neurônios, v) 10-Fold Cross-Validation (CV) com 30 neurônios, vi) SVM com kernel RBF, variância igual à 1^2 e limite para os multiplicadores de Lagrange igual à 2 [CV95], vii) MLP com 30 neurônios treinada com o algoritmo MOBJ [Tei01], viii) PLP com 15 neurônios paralelos treinada com um algoritmo MOBJ. A rede PLP treinada com o algoritmo de mínimo gradiente será identificada como PLP-MGM (Parallel Layer Perceptron with Minimum Gradient Method).

Na Tabela 6.1 está mostrado o MSE (mean squared error) médio e o desvio padrão de 100 simulações para um conjunto de teste composto de 1000 exemplos. Alguns resultados mostrados nesta tabela foram extraídos de [Tei01] (somente os resultados que utilizam a PLP foram gerados nesta tese).

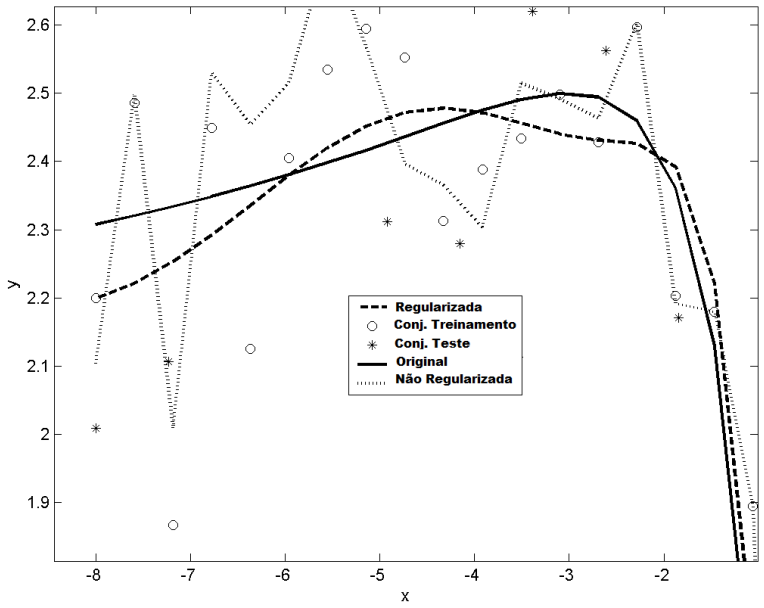

Figura 6.7: Zoom da função (lado esquerdo), mostrando que a solução regularizada "oscila menos" que a não regularizada.

Como pode ser visto na Tabela 6.1 a rede PLP-MGM foi capaz de gerar os melhores resultados, sendo estes muito próximos dos obtidos para as redes treinadas com o método multiobjetivo (MOBJ). Alguns pontos importantes devem ser ressaltados. Primeiro, como já discutido anteriormente o método MOBJ utiliza a minimização da norma dos pesos, e que este pode ser visto como uma aproximação da norma do gradiente. Entretanto o método MOBJ controla tanto os termos da parte linear quanto da não linear, sendo esta uma vantagem conceitual do método. Um ponto importante é que o custo computacional associado ao treinamento MOBJ é muito superior ao custo das inversões que são necessárias para o método do gradiente mínimo.

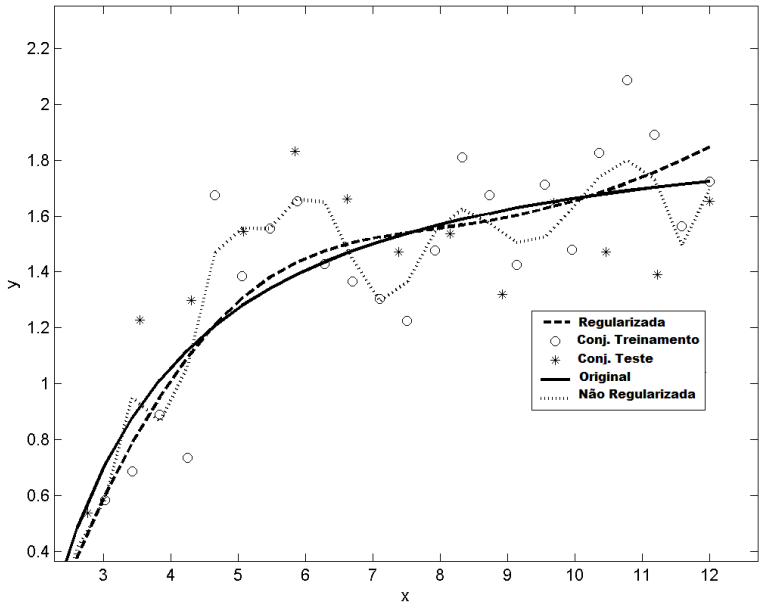

Figura 6.8: Zoom da função (lado direito), mostrando que a solução regularizada "oscila menos" que a não regularizada.

6.5.2 Problema de classificação

Considere os seguintes conjuntos:

$$\begin{aligned} C_1 &= [(x_1, x_2)|(x_1^2 + x_2^2) \leq 0,65] \\ C_2 &= [(x_1, x_2)|0,35 \leq (x_1^2 + x_2^2) \leq 2,15] \\ C_3 &= [(x_1, x_2)|(x_1^2 + x_2^2) \geq 1,85] \end{aligned} \tag{6.41}$$

Seja a primeira classe composta pelo conjunto $C_1 \bigcup C_3$ e a segunda por C_2. O vetor de entrada é composto pelas variáveis x_1 e x_2, com média zero, distribuição normal e variância igual à $0,5^2$, $1,5^2$ e 2^2 para os conjuntos C_1, C_2 *e* C_3, respectivamente. Os conjuntos de validação, treinamento e teste são compostos de 425, 850 e 850 padrões, respectivamente.

Na Tabela 6.2 é mostrado o percentual de classificações corretas considerando o conjunto de teste. Os resultados encontrados foram comparados com os obtidos utilizando os seguintes algoritmos: i) MLP com 30 neurônios na camada escondida treinada com o algoritmo de Levenberg-Maquardt (LM)

Tabela 6.1: Resultados para a função f1 utilizando a média de 100 simulações.

Algorithms	MSE	σ
MLP-LM [Tei01]	0,078	0,003
WD [Tei01]	1,46	0,02
OBD [Tei01]	0,070	0,003
ES [Tei01]	0,059	0,002
CV [Tei01]	0,058	0,002
SVM [Tei01]	0,078	0,003
MLP-MOBJ [Tei01]	0,049	0,002
PLP-MOBJ	0,048	0,002
PLP-MGM	0,047	0,003

[HM94], ii) MLP treinada com o método do decaimento dos pesos (WD) utilizando a constante de decaimento igual à 0.5, iii) Optimal Brain Damage (OBD) com 30 neurônios, iv) Early Stop (ES) com 30 neurônios, v) 10-Fold Cross-Validation (CV) com 30 neurônios, vi) SVM com kernel RBF, variância igual a 5 e limite para os multiplicadores de Lagrange igual à 6 [CV95], vii) MLP com 30 neurônios treinada com o algoritmo MOBJ [Tei01], viii) PLP com 15 neurônios treinada com o algoritmo MOBJ [Tei01].

Tabela 6.2: Resultados para o problema de classificação considerando a média de 100 simulações.

Algorithms	Right answers	σ
MLP-LM [Tei01]	67%	2%
WD [Tei01]	75%	2%
OBD [Tei01]	73%	2%
ES [Tei01]	76%	2%
CV [Tei01]	77%	2%
SVM [Tei01]	75%	2%
MLP-MOBJ [Tei01]	78%	2%
PLP-MOBJ	80%	1%
PLP-MGM	82%	3%

A PLP-MGM superou todas as outras topologias utilizadas neste problema teste, mostrando a sua eficiência, como mostrado na Tabela 6.2. Na Figura 6.9 é mostrado um resultado típico encontrado para este problema.

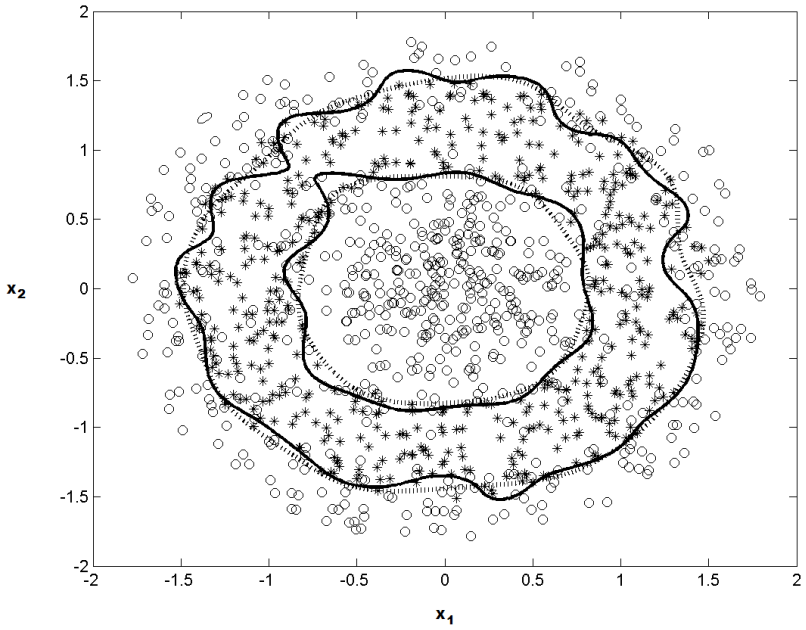

Figura 6.9: Problema de classificação definido na Equação 6.41. A curva contínua representa a resposta encontrada sem a utilização da regularização, e a curva tracejada a resposta regularizada.

6.6 Discussão

Nesta seção foi apresentada a principal contribuição desta tese, a rede perceptron com camadas paralelas. Foram mostradas algumas de suas características principais como algoritmos de treinamento e o teorema da aproximação universal. Primeiramente foram mostrados os algoritmos que utilizam o princípio da indução da minimização do risco empírico. Na seqüência foi apresentada uma técnica, que aplica a minimização do risco estrutural, o método do gradiente mínimo. Esta técnica nasce da idéia de regularização para polinômios apresentados na seção anterior, com a possibilidade de se utilizar uma camada polinomial na rede proposta. Do ponto de vista de custo computacional este método se mostrou mais adequado que o método multiobjetivo anterior, embora nenhum resultado neste sentido tenha sido apresentado formalmente.

A técnica PLP com gradiente mínimo considera a parte não-linear da rede

no cômputo da capacidade da mesma. Este fato é importante, pois embora os parâmetros não-lineares não tenham sido ajustados, estes são levados em consideração na formulação do problema. As SVMs, por exemplo, não levam em consideração este fato, controlam somente a parte linear. De fato, a determinação do espaço característico é um problema tão importante quanto a definição do hiperplano ótimo. Para analisar esta característica das SVMs considere a seguinte função a ser aprendida:

$$f(m_1, m_2, r) = C\frac{m_1 m_2}{r^2} \tag{6.42}$$

que é a lei da gravitação de Newton, expressando a força entre dois corpos de massas m_1 e m_2, separados por r. Uma máquina linear não pode representar esta função de forma adequada. Entretanto com uma mudança simples de coordenadas

$$\begin{aligned} g(x, y, z) &= ln\ f(m_1, m_2, r) \\ &= ln\ C + ln\ m_1 + ln\ m_2 - 2ln\ r = c + x + y - 2z \end{aligned} \tag{6.43}$$

obtém-se uma representação que pode ser aprendida de forma exata por uma máquina linear. A função $g(x, y, z)$ representa $f(m_1, m_2, r)$. Desta forma fica clara a importância da definição do espaço característico, e que este é problema-dependente. Os resultados teóricos relativos as SVMs não prevêem como definir este espaço característico e nem como contabilizá-lo na capacidade da função.

Como o espaço característico é um mapeamento não-linear, qualquer hiperplano de separação em qualquer espaço característico pode ser mapeado em um hiperplano de margem máxima em algum outro espaço característico da mesma dimensão, de forma que ambos representem a mesma separação no espaço de entrada. Uma pergunta fica em aberto, "Qual espaço característico o hiperplano de máxima margem garante os melhores limites de generalização?" [Zha01]. Esta pergunta não tem uma resposta clara.

Algumas técnicas foram propostas ao longo dos anos para tentar responder esta pergunta. Por exemplo, em [SBV95], foi apresentada uma maneira para limitar o número de coeficientes não zero que ocorrem na expansão das SVMs introduzindo uma função de custo específico que elimina a contribuição da função núcleo correspondente a pontos que contribuem para se obter uma margem para um problema dado. Em [AW99] foi proposto um método para

modificar a função núcleo para melhorar o desempenho de SVMs para problemas de classificação. Em [CST00] foi apresentada uma maneira dinâmica de se adaptar os núcleos nas SVMs. Utilizando a validação cruzada foi investigado em [DKP03] a seleção dos parâmetros dos núcleos. Núcleos híbridos foram propostos em [TW04]. De fato, a determinação de núcleos ótimos é um problema ainda em aberto e é, como mostrado no exemplo da gravitação, fundamental para o aprendizado de máquina.

Na formulação mostrada neste capítulo, os parâmetros não-lineares estão explícitos na formulação, embora não tenham sido ajustados. O fato de considerar tanto os elementos lineares como os não-lineares na formulação do problema é uma vantagem conceitual da técnica proposta.

Na próxima seção será mostrada a resolução de alguns problemas mais complexos.

Capítulo 7

Resultados experimentais

Neste capítulo serão apresentados alguns resultados experimentais obtidos para a rede perceptron com camadas paralelas utilizando o método do mínimo gradiente. A versão mono objetivo desta rede já foi testada em diversos problemas [CVV03] [VCV04] [VVCV05] [VVC04], que incluem alguns benchmarks da comunidade de redes neurais, aplicações relacionadas ao projeto de dispositivos eletromagnéticos, sendo que esta foi superior na maioria dos casos tanto em custo computacional como em poder de generalização quando comparada a técnicas que também não utilizam nenhum controle de complexidade (i.e., técnicas que utilizam o princípio de minimização do risco empírico e não do risco estrutural).

Neste capítulo a comparação é feita levando em conta os algoritmos avançados que se encontram na literatura que aplicam o princípio da minimização do risco estrutural.

7.1 Alto-falante

Um alto-falante com duas variáveis de projeto, h_m e w_t, como mostrado na Figura 7.1, é testado. O objetivo deste problema é aprender a densidade de fluxo magnético no air gap (F), o qual é calculado utilizando o método de elementos finitos. A base de dados foi gerada utilizando amostragem uniforme de 13 pontos em cada dimensão, somando um total de 169 amostras.

Na Tabela 7.2 são mostrados os erros, MSE - Mean Squared Error, para as duas topologias testadas, ANFIS e PLP, quando estes são comparados a derivadas numéricas.

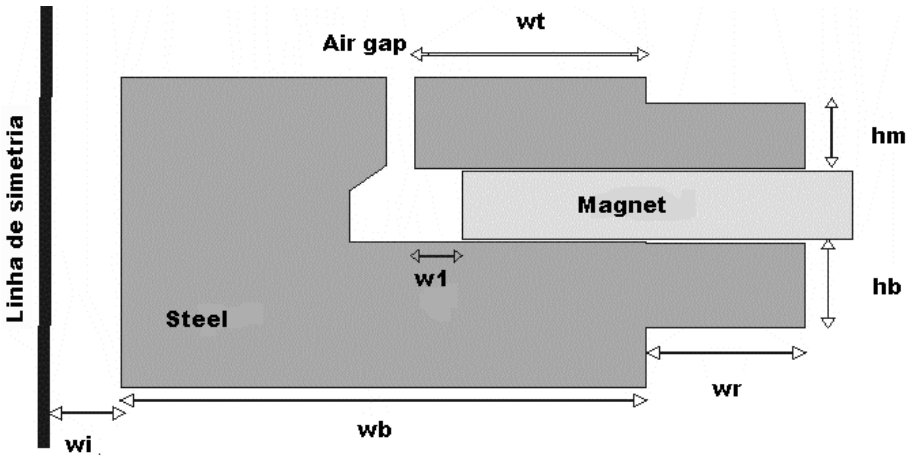

Figura 7.1: O problema do alto falante.

Tabela 7.1: Parâmetros do alto falante.

par(cm)	wi	wt	wl	wr	wb	hm	hb
fix	0,3	-	0,1	0,1	0,1	-	0,2
min	-	0,5	-	-	-	0,3	-
max	-	1,5	-	-	-	1,2	-

Da Tabela 7.2 pode ser notado que os dois modelos apresentam bons resultados se comparados com as derivadas numéricas. Entretanto, como pode ser notado, os resultados quando a PLP é utilizada são mais satisfatórios do que quando a ANFIS o é, mostrando, de novo, a eficiência de rede proposta. Outro ponto importante a ser notado é que os bons resultados obtidos para a PLP convencional e a ANFIS se deveram à escolha apropriada da estrutura testada. Pode-se notar que a PLP com 50 e 100 neurônios apresenta over-fitting, sendo que este problema é resolvido quando a PLP-MGM é utilizada, i.e., mesmo tendo uma estrutura super dimensionada a rede foi capaz de generalizar limitando as funções que esta aproxima. Mesmo se o operador definisse uma topologia inadequada, i.e. com muitos neurônios, a rede seria capaz de encontrar uma resposta adequada.

Tabela 7.2: Resultados para o alto falante.

Topology	MSE de F	MSE de $\frac{\partial F}{\partial x_1}$	MSE de $\frac{\partial F}{\partial x_2}$
ANFIS (9)[RRF00]	0,0011	1,10	1,26
ANFIS (16) [RRF00]	0,00056	0,68	0,84
PLP (9) [VCV04]	0,00055	0,78	1,21
PLP (16) [VCV04]	0,00027	0,55	0,38
PLP (20) [VCV04]	0,00026	0,65	0,46
PLP (50)	$8x10^{-6}$	1,20	0,79
PLP (100)	$6x10^{-6}$	1,48	1,62
PLP-MGM (50)	0,00039	0,62	0,50
PLP-MGM (100)	0,00025	0,56	0,47
PLP-MGM (200)	0,00026	0,66	0,42

7.2 Problemas do IDA benchmark repository

Todos os 13 problemas encontrados no IDA benchmark repository [IDA01] são testados nesta seção. Este problemas foram selecionados entre os problemas utilizados em redes neurais para testar diversas características das máquinas de aprendizagem. Os resultados encontrados para a PLP-MGM são comparados com os resultados obtidos pelas seguintes técnicas:

- SVM-Support Vector Machine,
- KFD- Kernel Fisher Discriminant,
- RBF -Radial Basis Function,
- AB -AdaBoost and
- ABR -Regularized AdaBoost extracted from [MMR+01];
- LOOM - Leave-One-Out SVM [WH00];
- LOO-KFD - Leave-One-Out KFD and
- Xval-KFD - k-fold KFD [CT03];
- Generalized Least Absolute Shrinkage and Selection Operator (LASSO) [Rot04];

- B-KLR - Bayesian KLR [CT05];
- PPSVM - Posterior Probability SVM [TWFYW05]
- PLP- Single objective Parallel Layer Perceptron [CVV03]

Estes resultados são mostrados na Tabela 7.3, 7.4 e 7.5 onde NA significa que os resultado não foi disponibilizado pelos autores. O número de neurônios paralelos (NPN) usados para a PLP e a PLP-MGM para cada conjunto de dados é indicado na ultima linha das tabelas.

Tabela 7.3: Ida repository I (Erros médios).

Method	Banana	B. Cancer	Diabets	German
SVM	11,5±0,7	26 ± 5	23±2	24±2
KFD	10,8±0,5	26 ± 5	23±2	24±2
RBF	10,8±0,6	28 ± 5	24±2	25±2
AB	12,3±0,7	30 ± 5	27±2	28±3
ABR	10,9±0,4	27 ± 5	24±2	24±2
LOO-KFD	10,4±0,4	26 ± 4	23±2	24±2
Xval-KFD	10,4±0,4	26 ± 4	23±2	24±2
LASSO	10,7±0,5	26 ± 5	24±2	24±2
LOOM	10,6±NA	26,3 ± NA	23,4±NA	NA
B-KLR	10,9±NA	27,7±NA	NA	22,7±NA
PPSVM	11,2±0,6	26 ± 5	23±2	NA
PLP	10,7±0,6	27 ± 5	23±2	30±3
PLP-MGM	10,7±0,6	25 ± 4	23±2	24±2
NPN	10	6	1	12

O primeiro ponto a ser notado nos resultados das Tabelas 7.3, 7.4 e 7.5, é que a PLP-MGM obteve melhores resultados que a PLP convencional, como esperado. Também pode ser notado que a PLP-MGM teve resultados similares aos alcançados pelas outras técnicas utilizadas na comparação. Uma comparação estatística destes resultados é então utilizada [Dem06]. Somente os métodos testados em todos os problemas serão considerados nesta comparação, sem considerar o desvio padrão das simulações. Primeiramente todos os algoritmos são classificados para cada conjunto, sendo que o de melhor performance recebe a classificação 1, o segundo 2 e assim por diante. No

Tabela 7.4: Ida repository II (Erros médios).

Method	Heart	Image	Ringnorm	S. Flare
SVM	16±3	3,0±0,6	1,7±0,1	32±2
KFD	16±4	3,3±0,6	1,5±0,1	33±2
RBF	18±3	3,3±0,6	1,7±0,2	34±2
AB	20±3	2,7±0,7	1,9±0,3	36±2
ABR	17±4	2,7±0,6	1,6±0,1	34±2
LOO-KFD	16±4	4,0±0,6	1,4±0,8	34±2
Xval-KFD	16±3	4,0±0,6	1,4±0,4	34±2
LASSO	16±3	NA	1,8±0,3	33±2
LOOM	16,1±NA	NA	NA	NA
B-KLR	NA	4,2±NA	NA	NA
PPSVM	15±3	NA	NA	NA
PLP	19±3	5±4	4±1	37±2
PLP-MGM	16±3	3,3±0,7	4±1	33±2
NPN	1	18	24	17

caso de empate a média das classificações é considerada. A média das classificações é mostrada na Tabela 7.6 e esta que mostra a PLP-MGM possui a melhor classificação média.

A classificação média é uma forma justa de comparar como mostrado em [Dem06]. Entretanto, o desempenho de dois classificadores é significantemente diferente se as classificações médias se diferem de pelo menos a diferença crítica

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{7.1}$$

onde k é o número de algoritmos, N o número de problemas testados. Para detalhes adicionais olhe [Dem06]. No presente caso $CD = 2.4$, o que significa que a PLP-MGM é estatisticamente superior somente em relação ao AB. De fato um maior número de problemas seriam necessários para indicar se existe uma diferença relevante entre as técnicas comparadas.

O número de neurônios da PLP-MGM foi definido empiricamente, e os parâmetros não lineares foram inicializados de forma aleatória. Um trabalho de pesquisa futuro interessante consiste no desenvolvimento de algoritmos para otimizar automaticamente os parâmetros não lineares, definindo então uma matriz Q "ótima" para o problema dado e número de neurônios.

Tabela 7.5: Ida repository III (Erros médios).

Method	Splice	Thyroid	Titanic	Twonorm	Waveform
SVM	10,9±0,7	5±2	22±1	3,0±0,2	9,9±0,4
KFD	10,5±0,6	4±2	23±2	2,6±0,2	10,7±1
RBF	10±1	5±2	23±1	2,9±0,3	10,7±1
AB	10,1±0,5	4±2	23±1	3,0±0,3	10,8±0,6
ABR	9,5±0,7	5±2	23±1	2,7±0,2	9,8±0,8
LOO-KFD	10,8±0,7	5±2	22±1	2,7±0,2	9,7±0,4
Xval-KFD	10,7±0,6	5±2	22±1	2,8±0,2	9,7±0,4
LASSO	NA	5±2	23±1	2,6±0,2	NA
LOOM	NA	5,0	22,7±NA	NA	NA
B-KLR	NA	NA	22,6±NA	NA	10,2±NA
PPSVM	NA	4±2	22±1	2,4±1	9,9±0,6
PLP	12±2	4±2	23±1	2,8±0,3	18±2
PLP-MGM	10±2	4±2	22±1	2,6±0,3	10,7±0,6
NPN	2	2	18	1	19

7.3 Diagnóstico de doença cardíaca

Em [YYN03] foi proposto uma variação das SVMs de margem suave (SVM-S) para as SVMs de margem total (SVM-T). Um dos exemplos testados neste artigo trata-se de um problema de diagnóstico de problemas cardíacos, com dados da Cleveland Clinic Foundation. Esta base de dados é composta por 13 variáveis e 303 observações. A técnica proposta é comparada com os experimentos descritos em [YYN03] e os resultados são mostrados na Tabela 7.7, sendo computado uma média de 100 simulações em todos os casos (infelizmente os autores não mostraram o desvio apresentado nos experimentos). Os resultados também são comparados com os resultados apresentados em [BA01], onde KFA significa Kernel function approximation.

Os resultados obtidos neste experimento mostram o poder de generalização da rede proposta.

7.4 Aplicação à analise de crédito

Da *UCI Machine Learning Repository* foi obtido uma base de dados para a avaliação de risco de crédito. O problema consiste em 690 padrões com

Tabela 7.6: Classificação média considerando os problemas do IDA repository.

Método	Classificação média
SVM	4,5
KFD	3,8
RBF	5,8
AB	6,3
ABR	4,3
LOO-KFD	3,9
Xval-KFD	4,0
PLP-MGM	3,7

15 entradas, sendo que se deseja avaliar se o cliente é apto ou não para receber o crédito. Deste conjunto 50% das amostras constituem o conjunto de treinamento, 25% o de validação e o restante o conjunto de teste.

Na Tabela 7.8 o resultado obtido pela técnica proposta é comparado com diversos resultados encontrados na literatura. De fato a técnica proposta é comparável a estas. Entretanto deve-se notar que o resultado obtido para a MLP-MOBJ foi muito superior aos demais e este fato deve ser investigado mais a fundo. Se comparado com a PLP-MG a MLP-MOBJ tem a vantagem de também ajustar os parâmetros da camada de entrada. Para este problema parece ser necessário definir uma estratégia para o ajuste dos parâmetros da camada não-linear da PLP.

7.5 Discussão

Neste capítulo foram apresentados alguns resultados obtidos aplicando a rede proposta neste trabalho, rede perceptron com camadas paralelas (PLP- Parallel Layer Perceptron) treinada com o método do gradiente mínimo (MGM - Minimum Gradient Method), em problemas reais. As comparações foram realizadas considerando as mais diversas técnicas existentes. A técnica proposta, PLP-MG, obteve, em geral, resultados comparáveis às principais técnicas estado-da-arte existentes. Esta comparação foi feita de forma ampla considerando diversos problemas e diversas máquinas de aprendizagem.

Tabela 7.7: Comparação da PLP-MG, SVM-S e SVM-T para o problema de diagnóstico de doença cardíaca.

Técnica	Média	Desvio
SVM-S (C=1)[YYN03]	20,00	?
SVM-S (C=5)[YYN03]	21,21	?
SVM-S (C=10) [YYN03]	21,32	?
SVM-T (C1=1, C2=0,05) [YYN03]	20,00	?
SVM-T (C1=1, C2=0,10) [YYN03]	19,23	?
SVM-T (C1=1, C2=0,50) [YYN03]	20,22	?
SVM-T (C1=5, C2=0,10) [YYN03]	21,10	?
SVM-T (C1=5, C2=0,50)[YYN03]	20,22	?
SVM-T (C1=5, C2=1,00) [YYN03]	20,99	?
SVM-T (C1=10, C2=0,10) [YYN03]	21,32	?
SVM-T (C1=10, C2=0,50) [YYN03]	19,87	?
SVM-T (C1=10, C2=1,00) [YYN03]	21,10	?
RBF [BA01]	21,6	?
SVM [BA01]	20,3	?
KFA [BA01]	16,8	?
PLP-MG	18	4

Tabela 7.8: Comparação da PLP-MG com diversos algoritmos para o problema de análise de crédito.

Técnica	Média	Desvio
RBF batch K-means [Lly82]	17	4
RBF DF [ISA84]	16	3
RBF IO [DH73]	18	4
RBF DFIO [IK89]	17	4
RBF IODF [IK89]	17	4
RBF on-line K-means [Mac67]	17	4
RBF optimal [CS95]	17	4
RBF GA [LCBL05]	14	4
Backpropagation [Tei01]	17	2
Cascade correlation [FL90]	18	3
Tower [PYH87]	15	3
Pyramid [PYH87]	17	2
SVM [Tei01]	17	3
MLP-MOBJ [Tei01]	9	1
PLP-MG	14	3

Capítulo 8

Considerações finais

Esta tese abordou alguns aspectos teóricos e práticos para o problema de aprendizado de máquinas. Primeiramente foi mostrado que as técnicas clássicas de regularização e o princípio indutivo da minimização do risco estrutural (SRM) são casos particulares de uma formulação mais geral que considera um problema de treinamento bi-objetivo. Este problema bi-objetivo deve ser escrito como a minimização de um funcional de risco empírico, R_{emp}, e a minimização de um funcional de complexidade, Ω. Embora apareçam na literatura alguns mecanismos que utilizam idéias derivadas da otimização multiobjetivo [TBTS00], [CBM+03] e [VCV06], neste trabalho as questões conceituais foram exploradas e ficou demonstrado como os métodos se relacionam. De fato, as idéias de regularização foram desenvolvidas para funcionais convexos embora dois de seus mecanismos também funcionem para funcionais quasi-convexos. O SRM depende somente da uni-modalidade dos funcionais envolvidos para que a ordenação proposta seja válida em todo espaço de interesse. O problema bi-objetivo descrito nesta tese inclui os problemas definidos previamente sem que a uni-modalidade seja necessária.

O segundo ponto desenvolvido neste trabalho foi uma forma genérica de escrever a complexidade de uma máquina de aprendizagem em termos de uma norma-Q. Esta abordagem é uma forma genérica de se separar o problema linear (que pode ser facilmente resolvido), do problema não-linear para o treinamento de máquinas de aprendizagem. Baseado nesta formulação pode-se mostrar que qualquer que seja a característica de complexidade esta será somente uma transformação afim dos vetores de parâmetros lineares. A simplicidade matemática deste modelo pode ser muito útil para o entendimento e para gerar novas técnicas de aprendizado. Pode-se, por exemplo, escrever

uma matriz Q de tal forma que a complexidade é definida como a norma do gradiente da saída da máquina treinada. Esta abordagem foi utilizada neste trabalho para gerar o método do gradiente mínimo (MGM - Minimum Gradient Method). Foi mostrado que o MGM está diretamente relacionado com a maximização das margens na formulação das SVMs e com a minimização da energia de alta-freqüência proposta para as RBFs. Estas contribuições podem ser muito úteis para a modelagem e entendimento do aprendizado para as mais diversas máquinas como mostrado no decorrer do texto.

As idéias conceituais desenvolvidas foram então aplicadas a aproximadores polinomiais. Foram utilizados aproximadores polinomiais com uma dimensão e foi mostrado que, dada esta classe de aproximadores o problema bi-objetivo descrito neste trabalho, considerando a complexidade como a norma do gradiente, é, de fato, um problema convexo. Utilizando a propriedade de convexidade, o algoritmo de treinamento pode ser resolvido utilizando o estimador de mínimos quadrados, que é uma forma robusta e eficiente para se resolver o problema de treinamento. Várias das propriedades teóricas previstas foram comprovadas nos experimentos feitos, como a de que a minimização do gradiente pode ser equivalente a uma filtragem das altas-freqüências. Para concluir observa-se que utilizar aproximadores polinomiais é interessante devido ao fato de estarem envolvidas somente funções quadráticas, mas alguns problemas numéricos e o mal da dimensionalidade, podem impedir que estes sejam de aplicação mais geral.

A maneira desenvolvida nesta tese para utilização de polinômios e evitar (pelo menos reduzir) o mal da dimensionalidade, foi baseado em um modelo onde um polinômio de primeira ordem (neste caso a complexidade cresce linearmente) é colocado em paralelo com um modelo não-linear. Esta é a arquitetura básica da rede proposta nesta tese a rede perceptron com camadas paralelas (PLP - Parallel Layer Perceptron). A PLP é uma nova arquitetura de redes neurais onde as MLPs e RBF, com uma camada escondida, são casos particulares desta. Portanto a PLP divide com elas características desejáveis como a de ser um aproximador universal, com uma flexibilidade extra devido às camadas em paralelo. Desta configuração foi derivada a principal contribuição prática deste trabalho, a rede PLP-MGM.

Considere dois conjuntos Pareto-Ótimo de redes no espaço dos objetivos como mostrado na Figura 8. Claramente as redes geradas no conjunto PO_1 são dominadas pelas pertencentes ao conjunto PO_2, i.e., dada uma rede pertencente à PO_1 existe uma rede em PO_2 que consegue obter menores erros de treinamento com menor complexidade. Desta forma, baseado nas

idéias apresentadas neste texto, seria interessante utilizar o conjunto de redes pertencentes à PO_2. Nesta linha, um dos pontos a se estudar no futuro, é a influência da camada não-linear na generalização da rede. Uma primeira abordagem seria a utilização de polinômios de ordem mais elevada na PLP, de tal forma que esta tenha mais parâmetros lineares e menos não-lineares (sem esquecer de controlar o mal da dimensionalidade). Outro ponto que pode ser abordado é a utilização de técnicas de otimização multiobjetivo não-linear nos pesos desta camada, como o algoritmo Dominating Cone Line Search[VTS06].

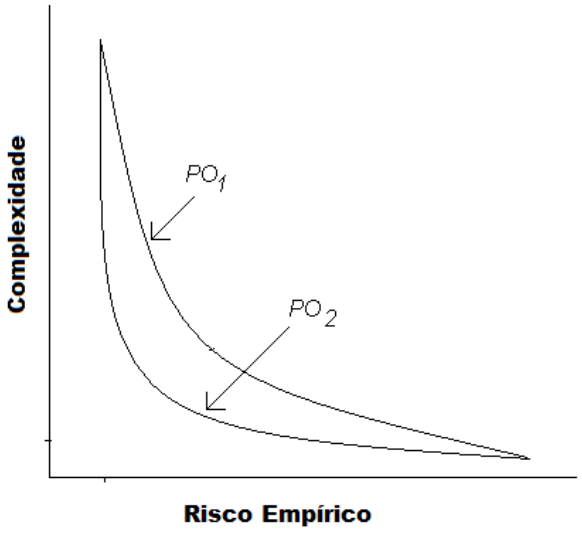

Figura 8.1: Nesta Figura são mostrados dois conjuntos de redes, PO_1 e PO_2, onde as redes em PO_2 dominam as de PO_1. Desta forma é interessante obter o conjunto PO_2 durante o treinamento, sendo este o trabalho futuro mais direto derivado desta tese.

Outro ponto interessante é tentar obter um entendimento mais profundo da constante λ utilizada para ponderar os compromissos entre os objetivos. Esta talvez possa ser utilizada para se realizar a escolha entre diversas estruturas treinadas com este algoritmo. Algumas extensões deste trabalho também podem ser obtidas, como a topologia de múltiplas saídas, e a recursiva. Espera-se que a extensão da técnica do gradiente mínimo pode ser facilmente estendida para outras redes como MLPs, RBFs, ANFIS.

Apêndice A

Problemas multiobjetivo

Neste apêndice serão mostrados alguns conceitos relativos à otimização e aos problemas multiobjetivo. O problema de otimização multiobjetivo (MOP) ou vetorial pode ser escrito como:

$$\begin{aligned} &min\ f(x) \\ &s.a.\ x \in \mathcal{F}_x \end{aligned} \tag{A.1}$$

Onde $f(.) : \mathbb{R}^n \rightarrowtail \mathbb{R}^m$ é o vetor dos objetivos do problema, $\mathcal{F}_x \subset \mathbb{R}^n$ é a região factível e $x \in \mathbb{R}^n$ são as variáveis de controle, espaço dos parâmetros. Os vetores $f(x) \in \mathbb{R}^m$ encontram-se em um espaço vetorial denominado de espaço dos objetivos. Logo neste problema se procura pontos tais que minimizem uma função vetorial.

Usualmente não é possível minimizar todos os objetivos simultaneamente, porque o ótimo de uma função dificilmente é o ótimo das outras, logo não há um ótimo único e sim um conjunto destes. Para definir minimização para problemas vetoriais algumas definições são necessárias.

i) *Dominância*: Um vetor x_1 domina x_2 se $f_j(x_1) \leq f_j(x_2)\ \forall j$ onde $j = 1, .., m$ e $f_j(x_1) \neq f_j(x_2)$ para pelo menos um j, onde $x_1\ e\ x_2 \in \mathcal{F}_x$.

ii) *Solução Pareto-Ótima (PO)*: Um vetor $x^* \in \mathcal{F}_x$ é Pareto-Ótimo (PO) se não existe outro vetor x no espaço viável que domina x^*.

Utilizando estas definições é possível gerar um conjunto de soluções denominadas soluções não dominadas ou PO, formando a fronteira PO, a qual contém os melhores compromissos entre os objetivos.

A.1 As condições de optimalidade de Kuhn-Tucker

Uma solução factível x^* satisfaz as condições necessárias de Kunhn-Tuker para eficiência se:

- todos os f_i's e g_i's são diferenciáveis, onde os g_i's são as restrições do problema, e
- existem vetores multiplicadores, $\mu^* \geq 0$, $\lambda^* \geq 0$, com pelo menos uma desigualdade estrita $\lambda_i^* \geq 0$, tais que:

$$g_k(x^*) \leq 0;$$

$$\mu_k^* g_k(x^*) = 0; \;\; k = 1, ..., p$$

$$\sum_{j=1}^{m} \lambda_j^* \nabla f_j(x^*) + \sum_{k=1}^{p} \mu_j^* \nabla g_k(x^*) = 0 \tag{A.2}$$

Estas condições são necessárias para que x^* seja um mínimo local.

A.2 Métodos para a resolução de problemas multiobjetivo

Nesta seção serão mostrados alguns métodos clássicos para a solução de problemas multiobjetivo.

A.2.1 Método da soma ponderada

O problema ponderado, P_λ, pode ser escrito, para um problema com m objetivos, como:

$$min \sum_{i=1}^{m} \lambda_i f_i(x), \;\; x \in \mathcal{F}_x \tag{A.3}$$

onde λ é o termo de ponderação. Este método gera uma sequência de hiperplanos paralelos, e termina com uma tangente à região viável, como mostrado na Figura A.1. Na Figura A.2 é mostrado um dos problemas desta abordagem, i.e., a parte não convexa do problema não tem hiperplano suporte, logo não pode ser mapeada.

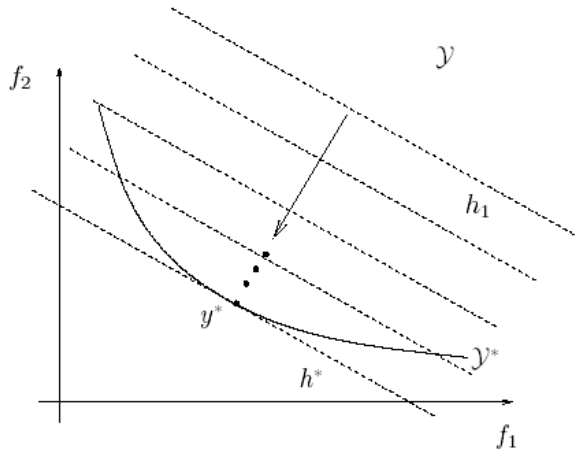

Figura A.1: Interpretação gráfica do método ponderado.

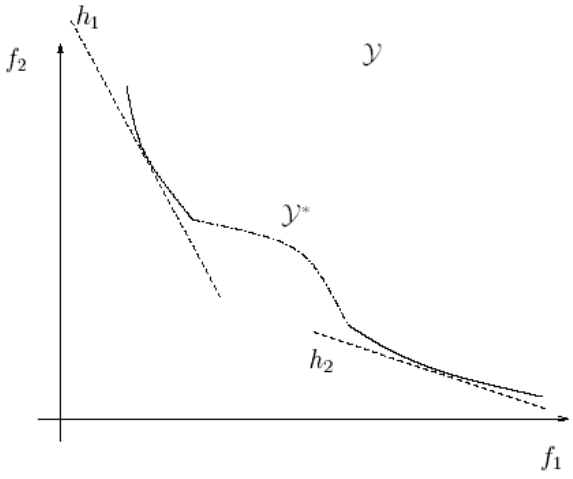

Figura A.2: Interpretação gráfica para o método ponderado quando a fronteira tem regiões não convexas.

A.2.2 Método ϵ-restrito

O método ϵ-restrito, P_ϵ, reescreve o problema multiobjetivo da seguinte forma:

$$\begin{array}{l} min\ f_i(x),\ x \in \mathcal{F}_x \\ s.a.\ f_j \leq \epsilon_j,\ j = 1, ..., m,\ j \neq i \end{array} . \tag{A.4}$$

Desta forma a fronteira PO é descrita como uma função do parâmetro ϵ_j, i.e., variando ϵ_j diversas soluções não dominadas podem ser obtidas, como mostrado na Figura A.3.

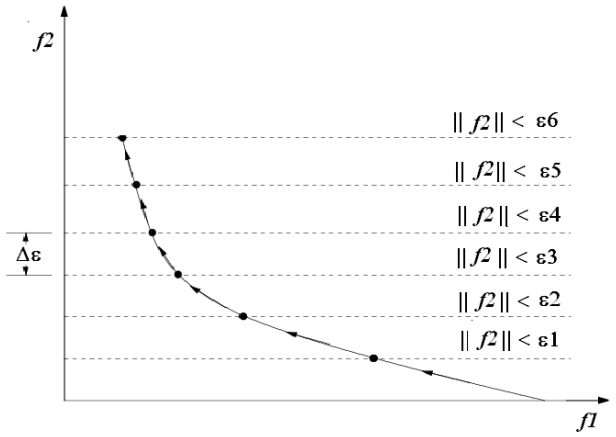

Figura A.3: Interpretação gráfica do método ϵ-restrito.

A.2.3 Programação por metas

O método de programação alvo ou por metas, P_μ, pode ser formulado da seguinte maneira:

$$min\ \|f(x) - M\|_p,\ x \in \mathcal{F}_x \tag{A.5}$$

para um dado $1 \leq p \leq \infty$. Onde M é um vetor composto por M_j que indicam a meta desejada para o objetivo.

A.2.4 Método das relaxações

O método das relaxações pode ser escrito como:

$$min\ \eta \tag{A.6}$$

$$s.a.\ f - f^* - \eta v \leq 0 \tag{A.7}$$

onde f^* é o vetor contento o mínimo de cada função e v_k é um vetor construído utilizando uma combinação convexa dos vetores objetivo. Uma interpretação gráfica deste método é mosttrada na Figura A.4.

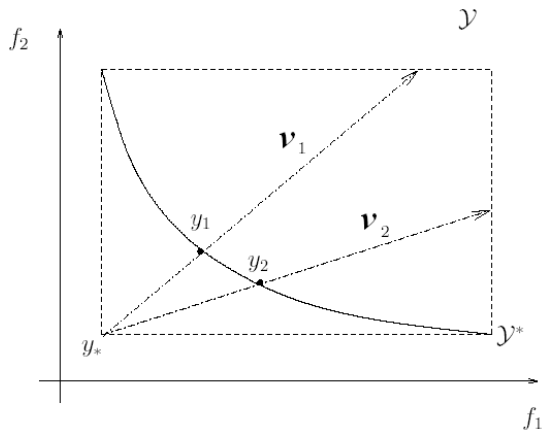

Figura A.4: Interpretação gráfica do método da relaxação.

Referências Bibliográficas

[And02] P. Andras. The equivalence of support vector machine and regularization neural networks. *Neural Processing Letters*, 15(2):97–104, 2002.

[AW99] S. Amari and S. Wu. Improving support vector machine classifier by modifying kernel functions. *Neural Networks*, 12:783–789, 1999.

[BA01] G. Baudat and F. Anouar. Kernel-based methods and function approximation. In Washington DC USA, editor, *International Joint Conference on Neural Networks*, pages 1244–1249, 2001.

[Bar93] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. on Information Theory*, 39(3):930 – 945, 1993.

[Bar97] P. L. Bartlett. For valid generalization the size of the weights is more important than the size of the network. *Advances in Neural Information Processing Systems*, 9:134–141, 1997.

[Bar98] P. L. Bartlett. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Trans. on Information Theory*, 44(2):525–536, 1998.

[BL89] D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1989.

[CBM+03] M. A. Costa, A. P. Braga, B. R. Menezes, R. A.Teixeira, and G. G. Parma. Training neural networks with a multi-objective

sliding mode control algorithm. *Neurocomputing*, 51:467–473, 2003.

[CDS90] Y. L. Cun, J. S. Denker, and S. A. Solla. Optimal brain damage. *Advances in Neural Information Processing Systems*, 2:589–605, 1990.

[CS95] C. Chinrungrueng and C. H. Séquin. Optimal adaptive k-means algorithm with dynamic adjustment of learning rate. *IEEE Trans. on Neural Networks*, 6:157–169, 1995.

[CST00] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and other kernel-based learning methods)*. Cambridge University Press, 2000. ISBN: 0 521 78019 5.

[CT03] Gavin C. Cawley and Nicola L. C. Talbot. Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. *Pattern Recognition*, 36:2585–2592, November 2003.

[CT05] Gavin C. Cawley and Nicola L. C. Talbot. The evidence framework applied to sparse kernel logistic regression. *Neurocomputing*, 65:119–135, March 2005.

[CV95] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–279, 1995.

[CVV03] W. M. Caminhas, D. A. G. Vieira, and J. A. Vasconcelos. Parallel layer perceptron. *Neurocomputing*, 55(3-4):771– 778, October 2003.

[Cyb89] G. Cybenko. Approximation by superpositions of a sigmoid function. *Mathematics of Control Signals and Systems*, 2:303–314, 1989.

[Dem06] Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[DH73] R. O. Duda and P. E. Hart. *Pattern classification and Scene Analysis*. Wiley-Interscience Publication, 1973.

[DKP03] Kaibo Duan, S. Sathiya Keerthi, and Aun Neow Poo. Evaluation of simple performance measures for tuning svm hyperparameters. *Neurocomputing*, 51:41–59, 2003.

[Fah88] S. E. Fahlman. Fast-learning variations on back-propagation: an empirical study. In D. Touretzky, G. Hinton, and T. Senjnowski, editors, *Proceeedings of the 1988 Connectioinst Models Summer School, Pittsburg*, pages 38–51, San Mateo, CA. Morgan Kaufmann, 1988.

[FL90] S. E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 524–532, Denver 1989, 1990. Morgan Kaufmann, San Mateo.

[Fun89] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks, Signals and Systems*, 2:183–192, 1989.

[GBD92] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias-variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[GJP95] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219 – 269, 1995.

[GWW61] D. Gabor, W. Wildes, and R. Woodcock. A universal nonlinear filter, predictor and simulator which. optimizes itself by a learning process. *IEE Proceedings*, 108B:422–438, 1961.

[Hay99] S. Haykin. *Neural Networks: A Comprehensive Fundation.* Prentice-Hall, New Jersey, 2nd edition, 1999.

[Hin89] G. E. Hinton. Connectionist learning procedures. *Artificial inteligence*, 40:185–234, 1989.

[HM94] M. T. Hangan and M. B. Menjah. Training feedforward network with the marquardt algorithm. *IEEE Trans. on Neural Networks*, 5(6):989–993, November 1994.

[HTF01] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, first edition, August 2001.

[IDA01] IDA. Ida bechmark repository used in several boosting, kfd and svm papers. Technical report, 2001. url: ida.first.gmd.de/ raetsch/data/benchmarks.htm.

[IK89] M. A. Ismail and M. S. Kamel. Multimensional data clustering utilizing hybrid strategies. *Pattern Recognition*, 22:75–89, 1989.

[ISA84] M. A. Ismail, S. Z. Selim, and S. K. Aror. Efficient clustering of multimensional data. In *Proc. of the IEEE International Conference on Systems Man and Cybernetics*, pages 120–123, 1984.

[Iva62] V. V. Ivanov. On linear problems which are not well-posed. *Soviet Math. Docl.*, 3(4):981–983, 1962.

[Iva76] V. V. Ivanov. *The theory of approximate methods and their application to the numerical solution of singular integral equations*. Leyden : Noordhoff International, 1976. ISBN: 9028600361.

[Jan93] J. S. R. Jang. Anfis: Adaptive-network-based fuzzy inference systems. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3):665–685, May 1993.

[KS90] Michael J. Kearns and Robert E. Schapire. Efficient distribution-free learning of probabilistic concepts (abstract). In *COLT '90: Proceedings of the third annual workshop on Computational learning theory*, pages 382–391, 1990.

[LCBL05] E. Lacerda, A. Carvalho, A. P. Braga, and T. B. Ludemir. Evolutionary radial basis functions for credit assessment. *Applied Intelligence*, 22(3):167–182, 2005.

[Lly82] S. P. Llyod. Least square quantization in pcm. *IEEE Trans. on Information Theory*, 28(2):129–137, 1982.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the Fifth Berkeley Symposium Math*, volume 1, pages 281–297, 1967.

[Mat] Mathworks. Matlab toolboxes. www.mathworks.com.

[MMR+01] K. Muller, S. Mika, G. Ratsh, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12(2):181–201, March 2001.

[MR97] John E. Moody and Thorsteinn S. Rögnvaldsson. Smoothing regularizers for projective basis function networks. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 585–605. The MIT Press, 1997.

[Phi62] D. Z. Phillips. A technique for numerical solution of certain integral equation of the first kind. *J. Assoc. Comput. Mach*, 9:84–96, 1962.

[PS91] J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3:246–257, 1991.

[PYH87] R. Parekh, J. Yang, and V. Honavar. Construtive neural network learning algorithms for multi-category real-value pattern classification. Technical report, Iowa State University, USA, Departament of Computer Science, 1987.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*, volume 1 of *In D.E. Rumelhart and J.L. McClelland, editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Bradford Books (MIT Press), Cambridge, MA, 1986.

[Rot04] Volker Roth. The generalized lasso. *IEEE Transactions on Neural Networks*, 15(1):16–28, January 2004.

[RRF00] K. Rashid, J. A. Ramirez, and E. M. Freeman. A general approach for extracting sensitivity analysis from neuro-fuzzy model. *IEEE Trans. on Magnetics*, 36(4):1066–1070, 2000.

[SBV95] B. Sholkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In *First International Conference Knowledge Discovery and Data Mining*, 1995.

[STB98] J. Shawe-Taylor and P. L. Bartlett. Structural risk minimization over data-dependent hierarchies. *IEEE Trans. on Information Theory*, 44(5):1926–1940, 1998.

[STC02] J. Shawe-Taylor and N. Cristianini. On the generalization of soft margin algorithms. *IEEE Trans. on Information Theory*, 48(10):2721–2733, October 2002.

[Sto74] M. Stone. Cross-validation choice and assessment os statistical predictions. *Journal of the Royal statistical society*, 36:111–147, 1974.

[TA77] A. N. Tikhonov and V. Y. Arsenin. *Solution of ill-posed problems*. W. H. Winston, Washington, DC, 1977.

[TBTS00] R. A. Teixiera, A. P. Braga, R.H.C Takahashi, and R. R. Saldanha. Improving generalization of mlps with multi-objective optimization. *Neurocomputing*, 35(1-4):189–194, 2000.

[Tei01] R. A. Teixeira. *Treinamento de Redes Neurais Artificias Atraves de Otimização Multi-Objetivo: Uma Nova Abordagem para o Equilibro entre a Polarização e a Variância*. PhD thesis, CPDEE- UFMG, 2001.

[Tik63] A. N. Tikhonov. On solving ill-posed problem and the method of regularization. *Doklady Akademii Nauk USSR*, 153:501–504, 1963.

[TPF97] R. H. Takahashi, P. L. D. Peres, and P. A. V. Ferreira. H2/h-infinity multiobjective pidpid design. *IEEE Control Systems Magazine*, 15(5):37–34, 1997.

[TW04] Y. Tan and J. Wuang. A support vector machine with a hybrid kernel and minimal vapnik-chervonenks dimension. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):385–395, April 2004.

[TWFYW05] Quing Tao, Gao-Wei Wu, Fei-Yue, and Jue Wang. Posterior probability support vector machines for unbalanced data. *IEEE Transactions on Neural Networks*, 16(6):1561–1573, November 2005.

[Vap92] V. N. Vapnik. Principles of structural risk minimization for learning theroy. *Advances in Neural Information Processing Systems*, 4:831–838, 1992.

[Vap98] V. N. Vapnik. *Statistical Learning Theory*. New York: Wiley, 1998.

[Vap01] V. N. Vapnik. *The Nature of Statistical Learning Theory (Statistics for Engineering and Information Science)*. Springer, second edition, September 2001.

[Vas70] V. V. Vasin. Relationship of several variational methods for approximate solutions of ill-posed problems. *Math Notes*, 7:161–166, 1970.

[VC71] V. N. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications,*, 16:264–280, 1971.

[VCV04] D. A. G. Vieira, W. M. Caminhas, and J. A. Vasconcelos. Extracting sensitivity information of electromagnetic devices models from a modified anfis topology. *IEEE Trans. on Magn*, 40(2):1180–1183, 2004.

[VCV06] D. A. G. Vieira, W. M. Caminhas, and J. A. Vasconcelos. Controling the parallel layer perceptron complexity using a multiobjective learning algorithm. *Neural Computing and Applications*, 2006. Available On-line.

[VTP+05] D. A. G. Vieira, R. H. C. Takahashi, V. Palade, J. A. Vasconcelos, and W. M. Caminhas. The q-norm complexity measure and the minimum gradient method: a novel approach to the machine learning structural risk minimization problem. submitted to IEEE Transactions on Neural Networks, September 2005.

[VTS06] D. A. G. Vieira, R. H. C. Takahashi, and R. R. Saldanha. A new vector optimization algorithm based on a cone of efficient directions and multiobjective line search. Submited to Mathematical Programming, November 2006.

[VVC04] D. A. G. Vieira, J. A. Vasconcelos, and W. M. Caminhas. Multiobjective methodology to compare neural networks applied in electromagnetics. In *Proc. of the 11th CEFC - IEEE Conference on Electromagnetic Field Computation*, 2004.

[VVCV05] D. G. Vieira, D. A. G. Vieira, W. M. Caminhas, and J. A. Vasconcelos. A hybrid approach combining genetic algorithm and sensitivity information extracted from a parallel layer perceptron. *IEEE Trans. on Magnetics*, 41(5):1740–1743, May 2005.

[WH00] J. Weston and R. Herbrich. Adaptive margin support vector machines, 2000.

[WHR90] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart. Predicting the future: a connectionist approach. *International Journal of Neural systems*, 1:193–209, 1990.

[YYN03] M. Yoon, Y. Yun, and H. Nakayama. A role on total margin support vector machines. In *IJCNN'03*, pages 2049–2053, 2003.

[Zha01] Bin Zhang. Is the maximal margin hyperplane special in a feature space? Technical report, Hewlett-Packard Labs, USA, April 2001.