

Uso de Informações Estruturais da
Matriz de Projeção para
Regularização de *Extreme Learning
Machines*

Lourenço Ribeiro Grossi Araújo

Programa de Pós Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais

Orientador: Antônio de Pádua Braga
Co-Orientador: Luiz Carlos Bamberra Torres

Dissertação de Mestrado

Fevereiro de 2019

Universidade Federal de Minas Gerais

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

**USO DE INFORMAÇÕES ESTRUTURAIS DA MATRIZ DE
PROJEÇÃO PARA REGULARIZAÇÃO DE EXTREME LEARNING
MACHINES**

Versão Final

Lourenço Ribeiro Grossi Araújo

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

Orientador: Prof. Antônio de Pádua Braga
Coorientador: Luiz Carlos Bamberira Torres

Belo Horizonte
Fevereiro de 2019

A663u	<p>Araújo, Lourenço Ribeiro Grossi. Uso de informações estruturais da matriz de projeção para regularização de Extreme Learning Machines [recurso eletrônico] / Lourenço Ribeiro Grossi Araújo. – 2019. 1 recurso online (x, 45 f. : il., color.) : pdf.</p> <p>Orientador: Antônio de Pádua Braga. Coorientador: Luiz Carlos Bambirra Torres.</p> <p>Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.</p> <p>Bibliografia: f. 32-35.</p> <p>Apêndices: f. 36-45. Exigências do sistema: Adobe Acrobat Reader.</p> <p>1. Engenharia elétrica - Teses. 2. Aprendizado do computador - Teses. 3. Redes neurais (Computação) - Teses. I. Braga, Antônio de Pádua. II. Torres, Luiz Carlos Bambirra. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.</p> <p style="text-align: right;">CDU: 621.3(043)</p>
-------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

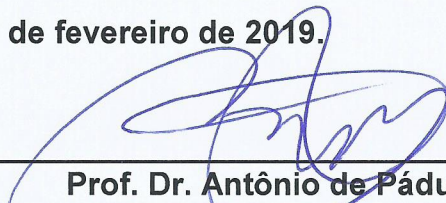
"Uso de Informações Estruturais da Matriz de Projeção para Regularização de Extreme Learning Machines"

Lourenço Ribeiro Grossi Araujo

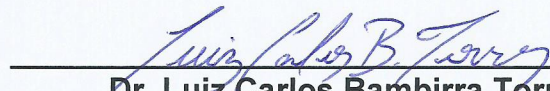
Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 14 de fevereiro de 2019.

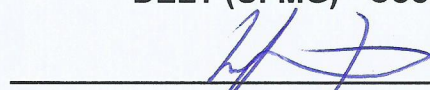
Por:



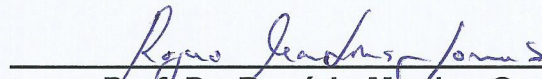
Prof. Dr. Antônio de Pádua Braga
DELT (UFMG) - Orientador



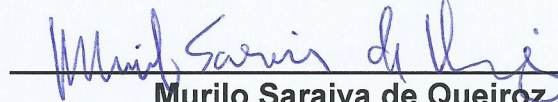
Dr. Luiz Carlos Bambirra Torres
DELT (UFMG) - Coorientador



Prof. Dr. Leonardo José Silvestre
Departamento de Computação e Eletrônica (UFES)



Prof. Dr. Rogério Martins Gomes
DECOM (CEFET-MG)



Murilo Saraiva de Queiroz
(NVIDIA)

Agradecimentos

Agradeço ao Professor Antônio de Pádua Braga, meu orientador, por me ter aberto as portas do LITC e por seus ensinamentos e disponibilidade, sem os quais este trabalho não existiria.

Agradeço, também, ao meu co-orientador, Luiz Carlos Bambirra Torres, pelas valiosas discussões que muito me ensinaram e enriqueceram este trabalho.

Agradeço à FAPEMIG, cujos recursos, na forma de bolsa, permitiram minha dedicação ao trabalho.

Devo agradecer também aos professores Marcelo Cardoso e Gustavo Almeida, que me permitiram encontrar o caminho da Inteligência Computacional vindo da Engenharia Química.

Agradeço a todos os membros do LITC com os quais compartilhei os 2 últimos anos pela convivência e por todo o aprendizado.

Finalmente, agradeço aos meus pais, Marcelo e Mônica, e à minha irmã, Ana, por todo o apoio e exemplo que sempre foram e são.

”Tenho o privilégio de não saber quase tudo,

E isso explica o resto”

Manoel de Barros

Resumo

O estudo tem como objetivo principal avaliar a possibilidade de se utilizar alguma informação a respeito da separabilidade linear dos dados projetados na camada oculta de uma rede neural do tipo ELM como informação para obtenção automática de um parâmetro de Regularização de Tikhonov. Redes neurais do tipo ELM são redes que podem ser treinadas de forma muito rápida e que apresentam a propriedade de aproximação universal. Alguma forma de regularização é necessária para que redes neurais do tipo ELM sejam capazes de generalizar e a regularização de Tikhonov é uma possibilidade. No entanto, tal técnica envolve a escolha de um parâmetro que pondera entre a minimização do erro de treinamento e a minimização da norma dos pesos. Tal escolha é geralmente feita por meio de um processo de validação cruzada, que é caro e contraditório a um dos princípios das ELM, que é, justamente, a alta velocidade de treinamento. As metodologias propostas geram modelos regularizados em tempo muito menor que o gasto para obter parâmetros por validação cruzada e com desempenho (medido em termos de acurácia) muito semelhante. Foram ainda, brevemente, desenvolvidas ideias estudando a possibilidade de se utilizar a matriz de distância da camada oculta de uma rede neural do tipo ELM para o treinamento e a respeito da regularização (sem parâmetros) de redes ELM construídas com spiking neurons.

Abstract

This work aims at evaluating the usage of some linear separability measure taken from the structure of a hidden layer projected matrix of an Extreme Learning Machine as prior information for automatic obtention of a regularization parameter for a Tikhonov Regularization. Extreme Learning Machines (ELM) are networks that can be trained very quickly and present universal approximation property. Some regularization is usually necessary in order to stop ELMs from overfitting and Tikhonov Regularization is a straightforward option. Such technique, however, demands the selection of a regularization parameter that weights the training error minimization and the network weights norm minimization. This selection is usually carried out by cross validation, which increases training times and in fact goes against ELM philosophy. Proposed methodologies are capable of generating regularized models with similar performance to those obtained through cross validation and in much shorter times.

The distance matrix calculated from the hidden layer of an ELM is also briefly explored and a proposal of parameterless regularization of Spiking Neurons ELMs is introduced.

Lista de Figuras

1.1	Superfícies de Separação obtidas por uma ELM de 100 neurônios com regularização com norma l_2 (linha contínua) e sem regularização (área sombreada)	2
2.1	Estrutura de uma rede do tipo SLFN, com uma entrada \mathbf{x} e saída \hat{y}	5
2.2	Curva de erro em forma U (James et al., 2013)	13
3.1	Comportamento dos erros de teste e treinamento e também do ângulo de separabilidade linear para o problema de classificação de 2 espirais	19
4.1	Superfícies de separação obtidas com ELMs de 20, 100 e 500 neurônios na camada escondida: sem regularização (ELM) e com as regularizações baseadas na Silhueta (ELM-sil)	25
A1	Superfícies de Decisão obtidas com ELM e com pesos obtidos pelo gradiente descendente considerando-se as matrizes de distância	41

Lista de Tabelas

4.1	Bases de dados Estudadas	26
4.2	Acurácia de teste ($\%media \pm \sigma$)	28
4.3	Tempo de treinamento (em segundos)	29
4.4	Resultados do teste t para a acurácia de teste	29

Lista de Abreviaturas

RNAS	Redes Neurais Artificiais
SLFN	<i>Single Layer Feedforward Network</i>
ELM	<i>Extreme Learning Machine</i>
SRM	<i>Structural Risk Minimization</i>
MDS	<i>Multidimensional Scaling</i>
SNN	<i>Spiking Neural Network</i>
ESN	<i>Echo State Network</i>
LSM	<i>Liquid State Machine</i>

Lista de Símbolos

X	Matriz de dados de entrada
H	Matriz de projeção
Z	Matriz de pesos da camada oculta
W	Matriz de pesos da camada de saída
y	Vetor de resposta
$h(\cdot)$	Função de ativação
b	Termo de <i>bias</i>
λ	Parâmetro de regularização
I	Matriz Identidade

Sumário

Lista de Figuras	v
Lista de Tabelas	vi
1 Introdução	1
1.1 Publicações	3
1.2 Organização do Trabalho	3
2 Revisão Bibliográfica	4
2.1 <i>Single Layer Feedforward Networks</i>	4
2.2 Extreme Learning Machines	7
2.3 Regularização	9
2.3.1 Regularização sob uma perspectiva do aprendizado estatístico	10
2.4 Regularização de ELMs	14
2.4.1 ELM regularizada(Deng et al., 2009)	14
2.4.2 OP-ELM e TROP-ELM	15
2.4.3 Regularização de ELMs com Matrizes de Afinidade	16
3 Separabilidade Linear e Regularização	17
3.1 Separabilidade Linear	17
3.2 Silhueta	20
3.3 Fisher Score	20
3.4 Cálculo de λ	21
3.4.1 Relação entre o Critério de Fisher e o Ângulo de Separabilidade	22
4 Resultados e Discussões	24
4.1 Bases de dados reais	24

5	Considerações Finais	30
	Referências Bibliográficas	32
	Apêndices	36
A	<i>Multidimensional Scaling</i> e Treinamento de Redes Neurais	37
A.1	Formulação Matemática	37
A.2	Treinamento de redes Neurais a partir do MDS	38
A.3	Resultados Preliminares	40
B	Regularização com Memória Associativa de Extreme Learning Machines de Neurônios de Pulso	42
B.1	Motivação	43

Capítulo 1

Introdução

Embora eficientes para solução de problemas de aprendizado de máquinas, as Máquinas de Aprendizado Extremo (do inglês, *Extreme Learning Machines*, ou ELMs) perdem a capacidade de generalização com relativa facilidade, problema conhecido como *overfitting* (Huang et al., 2004), e, portanto, beneficiam-se de estratégias de regularização. Por serem redes do tipo SLFN (*Single Layer Feedforward Network*) (Haykin, 1994), uma solução usual é a aplicação de uma regularização com norma l_2 (Deng et al., 2009), também conhecida como regularização de Tikhonov (Tikhonov, 1963). A maioria das estratégias de regularização, no entanto, envolve o ajuste de parâmetros por métodos iterativos, o que acaba por comprometer a velocidade de treinamento do modelo. No trabalho de Huang et al. (2012), é apresentada uma metodologia de regularização que envolve a seleção, por validação cruzada, de um parâmetro de regularização com norma l_2 . Já a estratégia conhecida como **OP-ELM**, proposta por Miche et al. (2010), envolve uma penalização com norma l_1 , que tem como efeito a poda da rede. Finalmente, a estratégia conhecida como **TROP-ELM** (Miche et al., 2011), envolve a aplicação, inicialmente de uma penalização com norma l_1 , seguida de uma penalização com norma l_2 . A Figura 1.1 apresenta o efeito da regularização com norma l_2 sobre a superfície de separação obtida após o treinamento de uma ELM para separação de duas distribuições normais.

É possível observar que a regularização com norma l_2 impõe suavidade à solução, com-

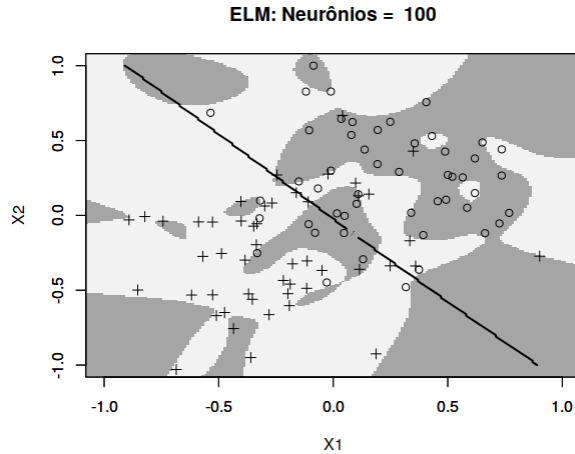


Figura 1.1: Superfícies de Separação obtidas por uma ELM de 100 neurônios com regularização com norma l_2 (linha contínua) e sem regularização (área sombreada)

batendo o problema de *overfitting*.

A introdução da etapa de validação cruzada para seleção de parâmetros de regularização na construção das ELMs acaba por aumentar o tempo necessário para o treinamento, o que reduz o efeito de uma das principais vantagens do modelo, que é justamente o tempo reduzido de treinamento (Silvestre et al., 2015).

No trabalho de Silvestre et al. (2015), utiliza-se a informação espacial *a priori*, formalizada como uma matriz de afinidade, para regularização de ELMs. É provado que a utilização da matriz de afinidade é similar à regularização de Tikhonov, e, quando utilizada uma matriz de afinidade independente de parâmetros, não é necessário o ajuste de nenhum parâmetro.

No presente trabalho, investiga-se a possibilidade de, a partir da estrutura dos dados, determinar, de forma automática, uma maneira de se regularizar a rede ELM construída. Diferentemente da metodologia apresentada por Silvestre et al. (2015), explora-se, neste trabalho, a estrutura dos dados projetados, valendo-se da ideia de separabilidade linear, uma das consequências do Teorema de Cover (Cover, 1965). A metodologia proposta é comparada a uma medida de separabilidade linear dos dados (Ben-Israel & Levin, 2006), com a finalidade de se conferir robustez teórica ao trabalho.

1.1 Publicações

- ARAUJO, L. R. G. ; TORRES, L. C. B. ; SILVESTRE, L. J. ; BRAGA, A. P. . Extreme Learning Machines regularizadas de forma automática a partir das informações estruturais da matriz de projeção. Congresso Brasileiro de Automática, 2018, João Pessoa. XXII Congresso Brasileiro de Automática, 2018. v. XXII. p. 1-6.
- ARAUJO, L. R. G. ; TORRES, L. C. B. ; SILVESTRE, L. J. ; BRAGA, A. P. . Regularization of Extreme Learning Machines with information of spatial relations of the projected data. Codit 2019, França (*Aceito*)

1.2 Organização do Trabalho

O trabalho está organizado da seguinte forma: o Capítulo 2 apresenta a revisão bibliográfica, e contém a teoria necessária para se compreender o processo de regularização de *Extreme Learning Machines*. São, também, apresentadas as metodologias existentes de regularização de ELMs e alguns conceitos sobre separabilidade linear.

O Capítulo 3 apresenta e desenvolve a ideia de se utilizar informações de separabilidade linear para regularização sem hiperparâmetros de redes ELM. O Capítulo 4 apresenta e discute os resultados e o Capítulo 5 traz as conclusões.

Os Apêndices A e B apresentam ideias que foram trabalhadas ao longo do curso de Mestrado, e que serão investigadas em trabalhos futuros.

O primeiro apêndice apresenta uma proposta de treinamento de Redes Neurais inspirada na técnica de *Multidimensional Scaling*. São apresentados os fundamentos da técnica, bem como um desenvolvimento matemático para a atualização dos pesos de uma Rede Neural.

O segundo apêndice traz uma proposta de algoritmo baseado em Redes Neurais Pulsadas e ELM para classificação de séries temporais.

Capítulo 2

Revisão Bibliográfica

2.1 *Single Layer Feedforward Networks*

As chamadas *Single Layer Feedforward Networks* (SLFN) são redes neurais utilizadas para solução de problemas não lineares de classificação de padrões e regressão. São redes compostas por três camadas, sendo apenas uma camada oculta, daí seu nome. A primeira camada é a camada de entrada, que conecta a rede neural ao ambiente seguida de uma camada oculta que é responsável por transformar os dados de forma não linear para um espaço (tipicamente) de maior dimensão (Haykin, 1994).

Um caso particular de rede SLFN são as redes neurais com camada de saída linear: são redes treinadas em duas etapas distintas, sendo a primeira uma transformação não linear dos dados para um espaço de maior dimensão que o espaço de entrada e a segunda a solução de um problema de mínimos quadrados para minimização do erro (Huang et al., 2004). Finalmente, tem-se a camada de saída linear. A Figura 2.1 traz a estrutura de uma rede do tipo SLFN com saída linear. Exemplos de redes SLFN com camada de saída linear incluem as redes RBF e as redes ELM (Haykin, 1994; Huang et al., 2004)

Na Figura 2.1, uma entrada \mathbf{x} de n dimensões passa pelos p neurônios da camada oculta da rede neural. A matriz de pesos \mathbf{Z} ligando a entrada à camada escondida apresenta dimensões $n \times p$. A camada escondida é responsável por projetar o vetor \mathbf{x} em dimensão p , somando-se

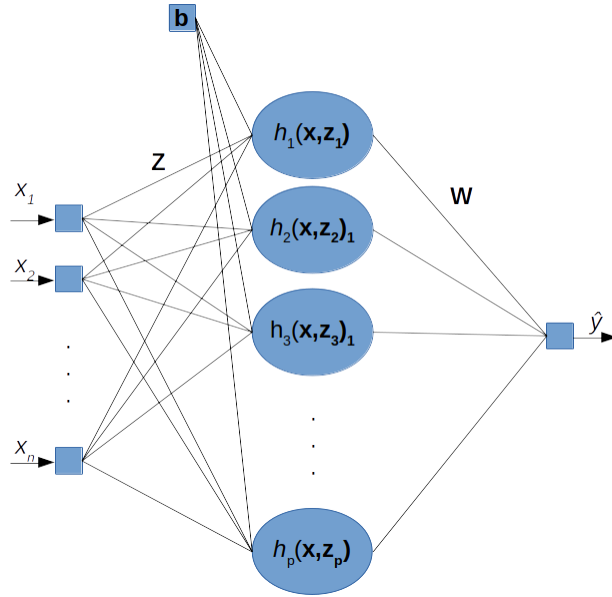


Figura 2.1: Estrutura de uma rede do tipo SLFN, com uma entrada \mathbf{x} e saída \hat{y}

ainda um vetor \mathbf{b} de *bias*, resultando em um vetor \mathbf{h} . Finalmente, o vetor \mathbf{h} passa pela matriz \mathbf{W} , que leva à resposta \hat{y} do modelo. Para uma resposta \hat{y} de m dimensões, a matriz \mathbf{W} apresenta dimensão $p \times m$. Para N amostras, apenas substituem-se os vetores \mathbf{x} , \mathbf{h} e \hat{y} pelas respectivas matrizes de N linhas \mathbf{X} , \mathbf{H} e \mathbf{Y} .

A i -ésima observação em uma SLFN será modelada por:

$$\hat{y}_i = \mathbf{W}h_i = \sum_{j=1}^p W_j g(\mathbf{z}_j \mathbf{x}_i + b_j) \quad (2.1)$$

em que a função $g(\cdot)$ corresponde à função de ativação na camada oculta, de tal forma que

$$h_i = g(\mathbf{z}_j \mathbf{x}_i + b_j)$$

O teorema de Cover traz as justificativas teóricas para o funcionamento das redes SLFN com camada de saída linear. O teorema é enunciado da seguinte forma:

Teorema 1. *Um problema complexo de classificação de padrões, projetado de forma não linear em um espaço de alta dimensão, tem maior probabilidade de ser linearmente separável que em um espaço de baixa dimensão, desde que o espaço não seja densamente povoado.*

O papel da camada oculta de uma SLFN é projetar de forma não linear os dados em um espaço de alta dimensionalidade. A camada de saída, por sua vez, constitui o hiperplano que permite separar os dados projetados (Haykin, 1994).

Diferentemente das redes neurais do tipo *Multilayer Perceptron* (MLP) em que os pesos de todas as camadas são aprendidos a partir da minimização recursiva de um erro de classificação ou regressão (retropropagação do erro), para as redes SLFN, apenas os pesos da camada de saída são calculados, a partir de um valor de erro.

O erro de uma SLFN pode ser calculado de acordo com a Eq. (2.2).

$$J = (\mathbf{Y} - \hat{\mathbf{Y}})^2 = \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.2)$$

Porém, como $\hat{y}_i = \mathbf{h}_i \mathbf{w}$, então:

$$J = \sum_{i=1}^N (y_i - \mathbf{h}_i \mathbf{w})^2 \quad (2.3)$$

Deseja-se encontrar os valores de \mathbf{W} que minimizam o valor do erro J . Os valores podem ser calculados pelas Eq. (2.4)-(2.7)

$$\arg \min_{\mathbf{w}} J \quad (2.4)$$

$$\frac{\partial J}{\partial w_j} = \frac{\partial}{\partial w_j} \sum_{i=1}^N (y_i - \mathbf{h}_i \mathbf{w})^2 = 0 \quad (2.5)$$

$$2 \sum_{i=1}^N (y_i - \mathbf{h}_i \mathbf{w}) h_{ij} = 0 \quad (2.6)$$

Que na forma matricial, pode ser reescrita como:

$$\mathbf{w} = \mathbf{H}^+ \mathbf{y} \quad (2.7)$$

O vetor de pesos, é, portanto, encontrado multiplicando-se a pseudoinversa da matriz de projeção (\mathbf{H}^+) pelo vetor de respostas, \mathbf{y} . A solução apresentada na Equação 2.7 é a solução de mínimos quadrados com norma mínima.

2.2 Extreme Learning Machines

As *Extreme Learning Machines* são um caso particular de SLFN que apresenta a propriedade de aproximação universal (Huang et al., 2004).

A matriz \mathbf{H} , construída a partir da Eq. (2.1), é dada por:

$$\mathbf{H} = \begin{bmatrix} h(\mathbf{z}_1 \cdot \mathbf{x}_1 + b_1) & \dots & h(\mathbf{z}_p \cdot \mathbf{x}_1 + b_p) \\ \vdots & \dots & \vdots \\ h(\mathbf{z}_1 \cdot \mathbf{x}_N + b_1) & \dots & h(\mathbf{z}_p \cdot \mathbf{x}_N + b_p) \end{bmatrix}. \quad (2.8)$$

A Eq. (2.1) na forma matricial é reescrita como:

$$\mathbf{HW} = \hat{\mathbf{Y}}. \quad (2.9)$$

Para que a rede seja capaz de aproximar as N observações, é necessário que:

$$\mathbf{HW} = \hat{\mathbf{Y}} = \mathbf{Y}. \quad (2.10)$$

O algoritmo de *Extreme Learning Machine* propõe que os pesos \mathbf{z}_j e os valores de bias b_j sejam inicializados de forma aleatória para $j = 1, \dots, p$. Torna-se desnecessário o aprendizado dos pesos por propagação reversa dos erros, o que dispensa a utilização de algoritmos de otimização não linear, que implicam em um consumo elevado de tempo (Huang et al., 2004; Miche et al., 2011).

Para que a rede seja capaz de aproximar bem os dados, é necessário um número elevado de neurônios. Desde que atendidas algumas condições com relação à inicialização dos pesos e às funções de ativação utilizadas, conforme mostrado por Huang, as ELMs se comportam como aproximadores universais (Huang et al., 2006a).

A matriz de pesos pode ser obtida, então, a partir da Eq. (2.11):

$$\mathbf{W} = \mathbf{H}^+ \mathbf{Y} \quad (2.11)$$

em que \mathbf{H}^+ é a pseudoinversa da matriz \mathbf{H} .

Como as *Extreme Learning Machines* são casos particulares de SLFNs, a obtenção de \mathbf{W} como apresentado na Eq. (2.11) leva à minimização de erro de treinamento (trata-se de uma das soluções do problema de mínimos quadrados dado por $\|\mathbf{HW} - \mathbf{Y}\|$) (Huang et al., 2004).

Um algoritmo para o treinamento de ELMs é fornecido por Huang et al. (2004) e é descrito a seguir:

Algoritmo 1: ALGORITMO PARA TREINAMENTO DE ELMs

Entrada: $(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbf{R}^n, \mathbf{t}_i \in \mathbf{R}^m, i = 1, \dots, N$; funções de ativação $g(x)$; p neurônios na camada oculta

Saída: Número esperado de nodos atingidos

1 início

2 | Atribuir valores arbitrários a \mathbf{z}_j e b_j para $j = 1, \dots, p$;

3 | Calcular o valor da matriz \mathbf{H}

4 | Calcular o vetor \mathbf{W} :

$\mathbf{W} = \mathbf{H}^+ \mathbf{Y}$

em que \mathbf{H}^+ é a pseudoinversa de \mathbf{H}

5 fim

6 retorna $\sigma(S)$

Uma das principais vantagens do algoritmo proposto é a sua alta velocidade de treinamento. O fato de que se obtém um problema de otimização para o qual é possível encontrar o mínimo global também é uma vantagem quando se compara com as redes MLP treinadas por retropropagação do erro: nestas é comum a ocorrência de mínimos locais que prejudicam os resultados do treinamento.

Quando se compara redes ELM a outras redes do tipo SLFN, como as redes RBF, por exemplo, tem-se uma vantagem relativa à simplicidade de configuração dos modelos. Se as redes RBF demandam a seleção prévia de centros e raios, as redes ELM demandam apenas a seleção da quantidade de neurônios.

As ELMs, no entanto, apresentam tendência ao *overfitting* (Deng et al., 2009), o que pode ser explicado pela etapa de treinamento, que se dá pela minimização do risco empírico (Vapnik, 2013), sem considerar o risco estrutural, como pode ser visto na Eq. (2.11).

2.3 Regularização

O treinamento de redes neurais é equivalente ao problema de estimação de um sistema que transforma entradas em saídas, a partir de um conjunto de pares entrada-saída (Poggio & Girosi, 1990). Trata-se de um problema inverso, em que, a partir de observações (efeitos), buscam-se as causas (Velho, 2008).

Problemas inversos, tipicamente, violam as 3 condições de Hadamard, o que os classifica como problemas mal postos. As condições de Hadamard são mencionadas abaixo:

I A solução existe;

II A solução é única;

III A solução tem dependência contínua (e suave) com os dados de entrada.

A resolução numérica de problemas mal postos é difícil e devem, portanto, ser encontradas estratégias que possibilitem seu tratamento. A regularização é a formulação proposta por Tikhonov para a solução de problemas mal postos. Em um problema de reconstrução de superfícies, a regularização de Tikhonov impõe suavidade à solução, o que transforma o problema mal posto em bem posto.

Tomando-se o exemplo do problema de otimização a ser resolvido para as SLFN (e especificamente para as ELM), a partir da Eq. (2.4), aplicar a regularização é equivalente a impor restrições do tipo $\Omega[w] \leq \rho$, em que $\Omega[w]$ é o operador de regularização e ρ é um limiar. Valendo-se dos multiplicadores de Lagrange, o problema regularizado passa a ser escrito de acordo com a Eq. (2.12).

$$\arg \min_{\mathbf{w}} J + \alpha \|\Omega[w]\| \quad (2.12)$$

onde α é o multiplicador de Lagrange, também conhecido como parâmetro de regularização, associado ao operador de regularização, que pode assumir várias formas funcionais. A regularização com norma l_2 dos pesos resulta na chamada Regularização de Tikhonov, conhecida também como *Ridge Regression* ou *Weight Decay*.

2.3.1 Regularização sob uma perspectiva do aprendizado estatístico

Seja $\mathbf{z} = \{\mathbf{x}, y\}$, um vetor contendo entradas e saídas, e $\mathbf{X}_N = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ um conjunto de N amostras independentes e identicamente distribuídas amostradas a partir de uma função de densidade de probabilidade conhecida $p(\mathbf{x})$. Dada uma função de perda $Q(\mathbf{x}, w)$, o objetivo de um problema de aprendizado é encontrar a função $Q(\mathbf{x}, w^*)$ que minimiza o funcional de Risco $R(w)$ dado pela Eq. (2.13).

$$R(w) = \int Q(\mathbf{x}, w)p(\mathbf{x})d\mathbf{x} \quad (2.13)$$

Sendo finito o conjunto de dados \mathbf{X}_N disponível, e sem conhecimento da função $p(\mathbf{x})$ fica-se limitado a minimizar o valor de risco aproximado, conhecido como Risco Empírico.

$$R_{emp}(x) = \sum_{i=1}^N Q(\mathbf{x}_i, w) \quad (2.14)$$

É possível perceber que a Eq. (2.4) apresentada para as SLFN corresponde exatamente à minimização do risco empírico apresentado na Eq. (2.14).

De acordo com Vapnik (2013), a minimização do Risco Empírico leva à minimização do Risco Real somente quando o conjunto de treinamento é suficientemente grande. Com efeito, para um problema de classificação, existe um limite superior para o Risco Real, dado pela Eq. (2.15).

$$R(w) \leq R_{emp}(w) + \sqrt{\frac{h(\log(2N/h)) - \log(\eta/4)}{N}} \quad (2.15)$$

onde N é a quantidade de amostras, η é um intervalo de confiança e o parâmetro h , conhecido como Dimensão de Vapnik-Chervonenkis (VC), é uma medida da complexidade do modelo construído. Para valores finitos de h , é fácil perceber que, quando N tende a infinito (conjuntos de amostras muito grandes), o valor de $R_{emp}(w)$ aproxima-se do valor real de $R(w)$.

A minimização do Risco Empírico, quando se trabalha com conjuntos finitos (e geralmente limitados) de amostras, pode levar ao problema conhecido como sobreajuste (do

inglês, *overfitting*). Tal problema se manifesta quando, para os dados de treinamento, o modelo apresenta grande acurácia mas, quando testado em um conjunto de dados inéditos, o chamado conjunto de teste, o desempenho deteriora, ou seja, o modelo não é capaz de generalizar.

A partir de conjuntos finitos de amostras, uma alternativa proposta por Vapnik para se estimar o Risco (e obter, assim, uma função a se minimizar que leve a resultados capazes de generalização) é a chamada Minimização do Risco Estrutural (do inglês, *Structural Risk Minimization*, SRM). A Minimização do Risco Estrutural é geralmente obtida ao se fixar *a priori* a estrutura (impedindo o crescimento da complexidade) do modelo e, só então, é minimizado o erro a partir da amostra dos dados.

Fixar a estrutura do modelo é semelhante a impor certo grau de suavidade à solução, de tal forma que a SRM é equivalente à minimização do Risco Empírico adicionado de um termo de regularização, como mostrado na Eq. (2.16).

$$R_{reg}(w) = R_{emp}(w) + \lambda\Omega(w) \quad (2.16)$$

Dilema Viés-Variância e Regularização

Quando se treina um modelo de aprendizado de máquina, dois tipos de erro existem: o erro de treinamento e o erro de teste. O erro de treinamento (ϵ_{treino}) diz respeito ao erro obtido para os dados utilizados no treinamento do modelo e, para um problema de regressão, o caso extremo de $\epsilon_{treino} = 0$ equivaleria à interpolação dos dados. Já o erro de teste (ϵ_{teste}) é o erro resultante quando amostras que não foram utilizadas durante a etapa de treinamento do modelo são apresentadas.

Se o erro de treinamento está relacionado ao Risco Empírico, o erro de teste está relacionado ao Risco Real, e diretamente relacionado à capacidade de generalização do modelo construído. É sabido que o erro de treinamento não é um bom estimador do erro de teste. Um exemplo são os modelos sobreajustados: erro de treinamento próximo a zero, mas erros de teste elevados (e conseqüentemente, baixa capacidade de generalização).

Segundo James et al. (2013), o erro de teste é composto de três termos: a variância do erro, o quadrado do viés do modelo e a variância do modelo. A variância do erro constitui o chamado erro irreduzível, e é o mínimo erro de teste que se pode obter. Já o erro redutível é composto pela soma do viés do modelo ao quadrado e da variância do modelo.

A variância do modelo diz respeito ao quanto o modelo varia ao ser estimado a partir de diferentes subconjuntos dos dados de treinamento. O viés, por sua vez, está relacionado ao erro introduzido ao se escolher um modelo para um problema real, muitas vezes extremamente complexo.

Como regra geral, modelos mais flexíveis levarão a baixos valores de viés e altos valores de variância (e vice-versa). Quando se pensa no erro de teste variando em função da flexibilidade do modelo, observa-se, tipicamente, um comportamento em U : quando a flexibilidade é muito baixa, o viés ao quadrado predomina na composição do erro de teste, enquanto a variância é baixa. Conforme a flexibilidade aumenta, o viés cai, e a variância sobe. Em determinado momento, o erro de teste atinge um valor mínimo (um valor ótimo no dilema viés-variância). A partir de então, o viés passa a cair mais lentamente, enquanto a variância continua a aumentar de forma que o valor de erro aumenta. Tal dinâmica é apresentada na Figura 2.2.

Na Figura 2.2, o erro de teste é representado em vermelho, constituído da soma do viés ao quadrado (em azul), variância do modelo (em amarelo) e variância do erro (linha tracejada horizontal). A linha tracejada vertical aponta para a flexibilidade ótima, que minimiza o erro de teste.

A regularização, pensada dentro do contexto do dilema viés-variância, tem por objetivo limitar a flexibilidade do modelo (e conseqüentemente o efeito da sua variância sobre o erro de teste), de tal forma que o treinamento leve a uma melhor estimativa do erro de teste.

Quando se pensa em *Extreme Learning Machines*, é fácil perceber que são modelos de alta flexibilidade (são aproximadores universais), portanto, tendem a ser modelos de baixo viés e grande variância. É necessário, portanto, limitar a flexibilidade dos modelos.

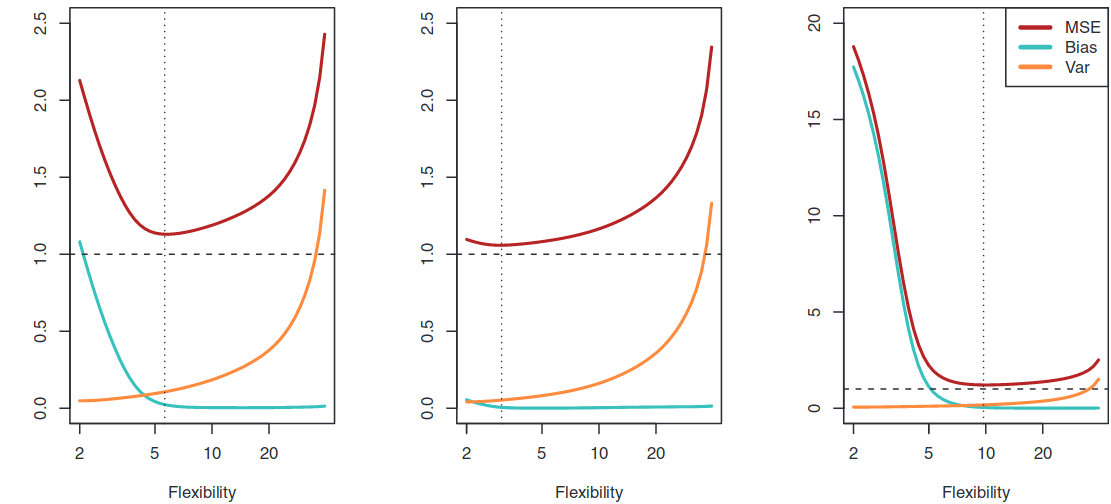


Figura 2.2: Curva de erro em forma U (James et al., 2013)

A figura apresenta o erro médio de teste (MSE) de regressão obtido com diferentes modelos para 3 conjuntos de dados diferentes. O gráfico à esquerda trata de dados que apresentam uma relação não linear com a resposta e com uma quantidade considerável de ruído, sendo assim, modelos de baixa flexibilidade apresentam grande viés (por tentar aproximar uma função geradora não linear por um modelo linear) enquanto que os modelos de grande flexibilidade apresentam grande variância (por aproximar apenas o ruído). Já o gráfico central trata de dados lineares, que, portanto, apresentam baixo viés para modelos de baixa flexibilidade e grande variância para modelos de maior flexibilidade. Finalmente, o gráfico à direita trata de dados com alta não linearidade, de tal forma que, mesmo para modelos de grande flexibilidade, a variância não cresce de forma explosiva.

2.4 Regularização de ELMs

A regularização de ELMs deve ser interpretada como uma etapa de minimização do risco estrutural (Vapnik, 2013), que impede o crescimento excessivo da rede (controlando ou a quantidade de neurônios, ou o módulo dos pesos).

As estratégias de regularização mais comuns envolvem tanto uma penalização por norma l_2 (regularização de Tikhonov), como apresentada por Deng et al. (2009) e Huang et al. (2012), quanto a seleção de neurônios mais importantes, graças a uma penalização com norma l_1 , como os métodos **OP-ELM** (Miche et al., 2010) e **TROP-ELM** (Miche et al., 2011), sendo que o último envolve a aplicação de ambas as penalidades: inicialmente aplica-se uma penalidade com norma l_1 , seguida de uma penalização com norma l_2 .

Processos de regularização baseados em penalização com norma l_2 (também conhecidos como regularização de Tikhonov) levam a uma diminuição no módulo dos pesos como um todo, o que proporciona maior suavidade à superfície de separação. Já a aplicação de penalidade com norma l_1 leva à seleção dos neurônios mais importantes, com os pesos dos neurônios menos importantes tendendo a zero (James et al., 2013).

2.4.1 ELM regularizada(Deng et al., 2009)

Quando busca-se a minimização tanto do risco empírico quanto do risco estrutural (regularização de Tikhonov com norma l_2), representado pela norma dos pesos da rede (o que é equivalente à busca por uma margem máxima de separação), obtêm-se as equações apresentadas por Deng et al. (2009). Para uma matriz de erros não ponderada (*unweighted*), a Eq. (2.17) apresenta o cálculo do vetor β de pesos.

$$\beta = \left(\frac{\mathbf{I}}{\gamma} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T Y \quad (2.17)$$

O parâmetro γ presente na Eq. (2.17) representa a ponderação entre Risco Empírico e Risco Estrutural no cálculo dos pesos de saída de uma ELM.

O efeito da regularização de Tikhonov proposta por Deng et al. (2009) é de diminuição da

norma dos pesos, o que é coerente com o resultado apresentado por Bartlett (1997), de que, para redes neurais, a magnitude dos pesos é mais importante que a quantidade de pesos.

2.4.2 OP-ELM e TROP-ELM

As redes ELM com poda ótima propostas por Miche et al. (2010, 2011) são redes obtidas após a seleção dos neurônios de maior relevância. A OP-ELM (Miche et al., 2010) busca lidar com a existência de variáveis irrelevantes no conjunto de dados, que são responsáveis por deteriorar a qualidade dos resultados. A metodologia de OP-ELM é constituída de três etapas:

I Treinamento de uma rede ELM

II Ordenação dos neurônios de maior relevância

III Remoção (utilizando *leave one out cross validation*) dos neurônios menos relevantes

As etapas II e III acima mencionadas são equivalentes a uma regularização com norma l_1 .

Um aprimoramento da OP-ELM, proposto por Miche et al. (2011) e conhecido como TROP-ELM (*Tikhonov Regularized Optimally Pruned ELM*), envolve a combinação de duas penalidades, l_1 e l_2 , de forma que exista, ao mesmo tempo, uma ordenação e seleção de neurônios e, também, um controle da magnitude dos pesos.

A metodologia resultante permite a construção de redes numericamente estáveis e com boa capacidade de generalização (resultante da penalidade com norma l_2) e robustas à existência de variáveis irrelevantes (resultado da seleção de neurônios mais importantes).

Nas três metodologias de regularização já citadas, existe a necessidade de se determinar a extensão da regularização aplicada: para as ELMs regularizadas de Deng et al. (2009), é necessário determinar o valor de γ , enquanto, para as metodologias OP-ELM e TROP-ELM, é necessário selecionar valores de λ_1 , associado à penalidade com norma l_1 e λ_2 , associado à penalidade com norma l_2 . A seleção de tais parâmetros é, tipicamente, feita de

forma iterativa, por validação cruzada, o que causa um aumento no tempo necessário para treinamento dos modelos (Silvestre et al., 2015).

2.4.3 Regularização de ELMs com Matrizes de Afinidade

No trabalho de Silvestre et al. (2015), é proposta uma estratégia de regularização de ELMs baseada na informação espacial disponível *a priori* a respeito dos dados, que é incorporada ao modelo na forma de uma matriz de afinidade. As etapas da metodologia proposta por Silvestre et al. (2015) são descritas a seguir.

- I Constrói-se uma rede neural do tipo ELM e computa-se a matriz \mathbf{H} da camada oculta;
- II A partir dos dados, calcula-se uma matriz de afinidade \mathbf{P} ;
- III Transforma-se a matriz \mathbf{H} em uma matriz $\mathbf{H}' = \mathbf{PH}$.
- IV Calculam-se os pesos de saída \mathbf{W}' a partir da pseudoinversa da matriz \mathbf{H}' .

É possível demonstrar que o cálculo dos pesos a partir da metodologia descrita acima é equivalente à Regularização de Tikhonov - disponível em (Silvestre et al., 2015). Se a matriz de afinidade \mathbf{P} for construída sem a necessidade de seleção de parâmetros, a metodologia proposta por Silvestre et al. (2015) leva à regularização de ELMs livre de parâmetros. Os ganhos de tempo obtidos quando a regularização é executada sem seleção de um parâmetro de regularização são consideráveis.

Capítulo 3

Separabilidade Linear e Regularização

Por apresentar uma camada de saída linear, uma rede neural do tipo ELM só conseguirá classificar corretamente a totalidade das amostras de treinamento quando os dados forem linearmente separáveis no espaço de projeção, ou seja, sempre que, para um problema de classificação binária existir um hiperplano no espaço projetado dado pela Equação 3.1

$$\mathbf{W}^T \mathbf{x} + b = 0 \tag{3.1}$$

de tal forma que para toda amostra i tal que $y_i = +1$, $\mathbf{W}^T \mathbf{x}_i + b \geq 0$ e vice-versa.

A existência de tal plano é equivalente a dizer que os dados apresentados na matriz \mathbf{H} são linearmente separáveis.

3.1 Separabilidade Linear

O conceito de separabilidade linear é frequentemente utilizado na formulação de algoritmos de aprendizado de máquina. Os exemplos incluem *Support Vector Machines* (Vapnik, 2013) e redes neurais (tanto do tipo *Perceptron* quanto as SLFN em geral e as ELM em particular (Haykin, 1994)). O trabalho de Elizondo (2006) apresenta uma revisão de diferentes estratégias para teste de separabilidade linear de conjuntos. O autor classifica os métodos para teste de separabilidade linear em quatro grupos: 1) Programação Linear; 2) Geometria

Computacional; 3) Redes Neurais; 4) Programação Quadrática. É apresentado também o Discriminante Linear de Fisher.

De modo geral, os métodos tratados no trabalho de Elizondo (2006) apresentarão problemas quando a quantidade de dimensões for elevada, quando os conjuntos de dados forem grandes e quando as distribuições de probabilidade dos dados não obedecerem a critérios como normalidade e ausência de *outliers*. Quando se lida com problemas de otimização, pode ocorrer também a inexistência de garantia de globalidade dos pontos ótimos, o que compromete os resultados.

Para redes neurais do tipo ELM a separabilidade linear na projeção dos dados é obtida com o acréscimo de neurônios na camada oculta. Por ser um aproximador universal, é sabido que uma ELM será capaz de obter erro de treino zero, desde que uma quantidade suficiente de neurônios seja utilizada na construção do modelo.

Ben-Israel & Levin (2006) propõem uma metodologia que permite medir o grau de separabilidade linear de um conjunto binário de classificação. Trata-se de um ângulo (variável de 0 a $\frac{\pi}{2}$), dado pela equação 3.2

$$\theta(\kappa) = \arctan \frac{\kappa \|\hat{S}^{-1} \hat{\mathbf{d}}\|}{2} \quad (3.2)$$

em que \hat{S} é a matriz de covariância obtida a partir das amostras, $\hat{\mathbf{d}}$ é a distância entre as médias de cada classe e κ é um fator de regularização. Para valores pequenos de κ , se as classes forem bem separadas (linearmente), o ângulo se aproxima de $\frac{\pi}{2}$. Se as classes não forem bem separadas, o valor de θ se aproxima de 0.

Para o problema de classificação de duas espirais utilizando-se ELMs, ao variar-se a complexidade (quantidade de neurônios) é possível observar que a separabilidade linear dos dados projetados (para o conjunto de treinamento) aumenta conforme cai o erro de treinamento, enquanto que o erro de teste segue a curva U, conforme o esperado: em um primeiro momento, o erro é dominado pelo termo de viés, que cai com o aumento da complexidade; em seguida, o aumento da variância leva ao crescimento do erro de teste. Todos os comportamentos mencionados podem ser visualizados na Figura 3.1

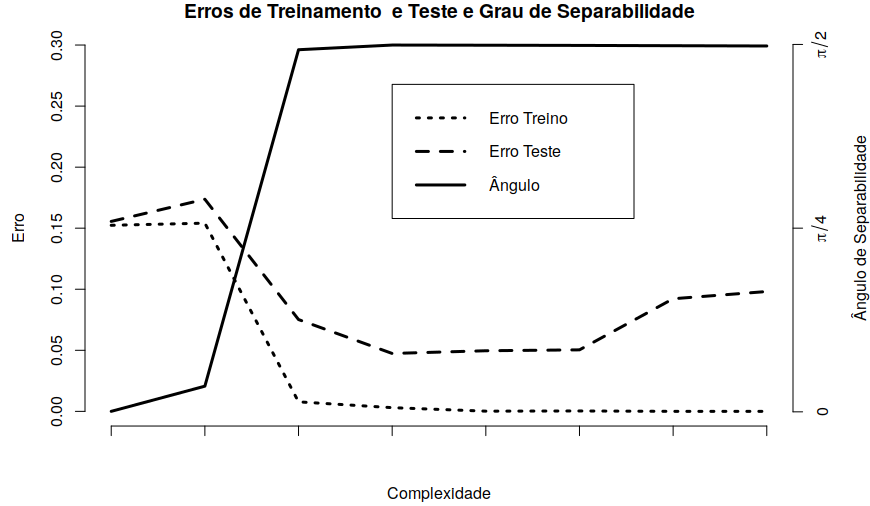


Figura 3.1: Comportamento dos erros de teste e treinamento e também do ângulo de separabilidade linear para o problema de classificação de 2 espirais

Como discutido no Capítulo 2, a regularização de Tikhonov é uma alternativa para controle da complexidade do modelo, o que leva a uma maior capacidade de generalização de modelos de maior flexibilidade.

Uma vez observada a relação entre separabilidade linear e complexidade (e, consequentemente, necessidade de regularização), e mantendo-se em mente o custo da seleção de parâmetros de regularização, torna-se promissora a busca por uma relação direta entre medidas de separabilidade linear e valores para o parâmetro de regularização de Tikhonov (λ).

Tipicamente, os valores de λ selecionados por validação cruzada para a regularização de Tikhonov, variam de 0 (nenhuma regularização) a infinito (máxima regularização). A metodologia proposta por Deng et al. (2009), por exemplo, propõe valores de λ que variam de $2^{-50} \approx 10^{-16}$ a $2^{50} \approx 10^{15}$. É necessário, portanto, encontrar relações envolvendo as medidas de separabilidade linear que levem a valores de λ na faixa desejada (\mathbb{R}^+). É desejado, também, que não existam parâmetros que devam ser ajustados pelo usuário, caso contrário, é mais simples utilizar alguma das metodologias já propostas de regularização com norma l_2 de redes ELM. Sendo assim, não faz sentido utilizar a medida proposta por Ben-Israel & Levin (2006), uma vez que o ajuste do parâmetro κ é necessário.

No presente trabalho são propostas e testadas duas metodologias que não envolvem parâmetros e que espera-se que sejam capazes de descrever a separabilidade linear da matriz de projeção do conjunto de treinamento.

3.2 Silhueta

O conceito de Silhueta (Kaufman & Rousseeuw, 2009) foi desenvolvido para técnicas de *clustering* e fornece uma medida da capacidade que os agrupamentos propostos teriam para separar de forma coerente os dados.

O valor de Silhueta para um dado grupo (classe) i pode ser obtido a partir da Equação (3.3):

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))} \quad (3.3)$$

em que i indica o índice dos grupos, $a(i)$ é a distância média intragrupos e $b(i)$ é a mínima distância média intergrupos. Nota-se que o valor de Silhueta pode assumir valores entre $-1 \leq s(i) \leq 1$. Quanto mais o valor de silhueta $s(i)$ é próximo de 1, mais o elemento i é similar aos demais elementos de seu próprio grupo, quando comparado aos elementos dos demais grupos (a similaridade é medida em função das distâncias $a(i)$ e $b(i)$). Já valores próximos de -1 indicam um alto grau de dissimilaridade intragrupo.

Tomando-se a Silhueta média (\mathcal{S}), tem-se uma única medida que representa a qualidade dos agrupamentos. É esperado que, quanto maior a separabilidade linear, maior será o valor de Silhueta média, e vice-versa.

3.3 Fisher Score

A ideia motivadora do *Fisher score* (Duda et al., 1973) é encontrar a combinação de características que maximiza a distância interclasse e minimiza a distância intraclasse. Uma heurística comum consiste em analisar cada característica de forma independente: para cada

uma das dimensões do problema, calcula-se a distância euclidiana entre as médias, ponderada pela dispersão dos dados (variância).

$$\mathcal{F}(x^j) = \frac{\text{dist}(\mu_+^j, \mu_-^j)}{(\sigma_+^j)^2 + (\sigma_-^j)^2} \quad (3.4)$$

em que μ_+^j e $(\sigma_+^j)^2$ são respectivamente a média e a variância da classe positiva ao longo da característica j , enquanto μ_-^j e $(\sigma_-^j)^2$ representam a classe negativa. Quanto maior o valor de $\mathcal{F}(x^j)$, maior a importância da dimensão avaliada.

3.4 Cálculo de λ

A partir de bases de dados sintéticas, de fácil visualização, foram obtidas, de forma empírica, relações entre os valores de Silhueta e *Fisher Score* e valores de λ adequados para a Regularização de Tikhonov.

Como os valores de Silhueta média variam de -1 até 1 , foi necessário obter uma função que, para tais valores de entrada, retornasse valores de λ entre 0 e $+\infty$. Considerou-se que valores negativos de silhueta representam dados com baixo grau de separabilidade linear, e, portanto, a necessidade de regularização não existe. Já para valores positivos de silhueta, optou-se por uma função do tipo $f(x) = \frac{-1}{\log(x)}$, que retorna valores próximos de 0 para valores de \mathcal{S} próximos de 0 e virtualmente infinitos para valores de \mathcal{S} próximos de 1 .

O valor de λ a partir da Silhueta média é definido a partir da Equação 3.5.

$$\lambda(\mathcal{S}) = \begin{cases} 0, & \text{se } \mathcal{S} \leq 0 \\ \frac{-1}{10\log(\mathcal{S})}, & \text{caso contrário.} \end{cases} \quad (3.5)$$

o multiplicador 10 no denominador foi definido empiricamente a partir de bases de dados sintéticas.

Já os valores de *Fisher Score* variam de 0 a $+\infty$ sem a necessidade de ajustes. Quando a distância intragrupos é grande, o valor de σ^2 domina e o valor do critério tende a zero. Quando a distância intergrupos é grande, por sua vez, a diferença entre as médias domina e

o valor do critério tende a infinito. Tal comportamento do critério é coerente com o comportamento que se espera do valor de λ : grandes valores de distância intergrupos (e/ou baixos valores de distância intragrupo) implicariam em fácil separabilidade, e consequentemente na necessidade de um alto valor de λ , enquanto que grandes distâncias intragrupos (e/ou baixas distâncias intergrupo) implicam em baixa separabilidade e em um baixo valor de λ . Optou-se, portanto, por calcular o valor de λ a partir da Equação 3.6.

$$\lambda(\mathcal{F}) = \text{mean}(\mathcal{F}) \quad (3.6)$$

Assim como a Equação 3.5 foi definida e ajustada empiricamente em bases de dados sintéticas, também a Equação 3.6 foi selecionada após testes e visualização em bases de dados sinteticamente geradas.

3.4.1 Relação entre o Critério de Fisher e o Ângulo de Separabilidade

É interessante mostrar como o valor de λ obtido a partir da Equação 3.6 se relaciona com o valor de $\tan(\theta)$ conforme proposto na Equação 3.2. Ben-Israel & Levin (2006) consideram que a variância para as duas classes são iguais. Considerando-se também que a matriz $\hat{\mathbf{S}}$ é diagonal da forma

$$\hat{\mathbf{S}} = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_p^2 \end{bmatrix}$$

o que é coerente com a forma como se calcula o critério de Fisher, o desenvolvimento das Equações 3.2 e 3.6 leva às Equações 3.7 e 3.8.

$$\lambda(\mathcal{F}) = \text{mean}(F) = \frac{1}{2p} \sum_{i=1}^p \frac{\sqrt{(\mu_{i,+} - \mu_{i,-})^2}}{\sigma_i^2} \quad (3.7)$$

$$\lambda = \tan(\theta) = \frac{\kappa}{2} \sqrt{\sum_{i=1}^p \frac{(\mu_{i,+} - \mu_{i,-})^2}{(\sigma_i^2)^2}} \quad (3.8)$$

Embora as Equações 3.7 e 3.8 não levem às mesmas formulações, é possível perceber a similaridade entre os resultados, de forma que não parece absurdo concluir que, em alguma medida, a Equação 3.7 representa, de fato, a separabilidade linear dos dados a partir, exclusivamente, de informações contidas na matriz de projeção dos dados \mathbf{H} .

Capítulo 4

Resultados e Discussões

A Figura 4.1 mostra o efeito da regularização aplicada utilizando as metodologias propostas sobre a superfície de separação obtida com ELMs de 20, 100 e 500 neurônios na camada escondida.

O conjunto de 100 amostras foi gerado sinteticamente a partir de duas distribuições normais com mesma variância.

Observa-se na Figura 4.1 que os valores propostos de λ levam a uma suavização das superfícies de decisão. Sabe-se que, para duas distribuições normais, com mesma variância, a superfície ótima de separação é dada por um hiperplano entre as classes. As duas ELMs regularizadas, com os valores propostos de λ foram capazes de se aproximar melhor da superfície ótima.

4.1 Bases de dados reais

Foram realizados testes envolvendo sete das dez bases de dados testadas por Silvestre et al. (2015), e compararam-se os resultados obtidos quando o parâmetro de regularização λ foi selecionado por validação cruzada, **ELM-reg**, com os resultados obtidos quando λ foi selecionado baseado na silhueta, **ELM-sil**, e quando λ foi selecionado baseado no critério de Fisher, **ELM-fis**. Não foram testadas todas as bases utilizadas por Silvestre et al. (2015)

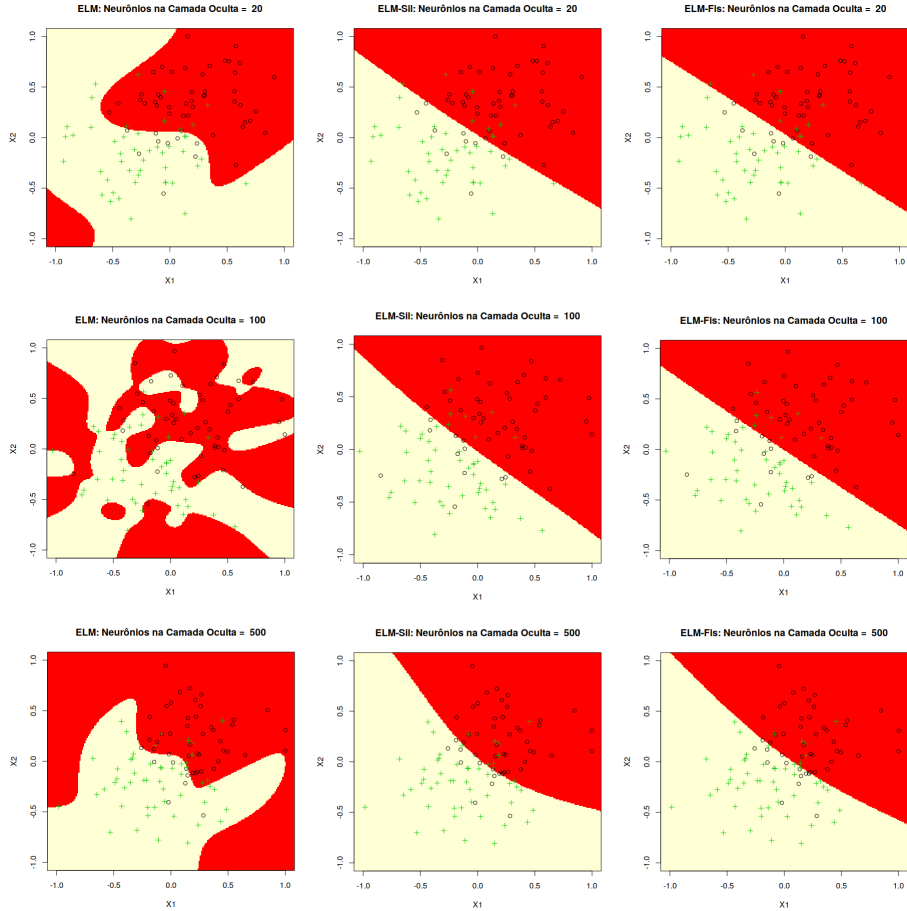


Figura 4.1: Superfícies de separação obtidas com ELMs de 20, 100 e 500 neurônios na camada escondida: sem regularização (ELM) e com as regularizações baseadas na Silhueta (ELM-sil) e no Critério de Fisher (ELM-fis)

pois três dentre as dez bases são originariamente não binárias e optou-se por não realizar a binarização das bases.

Para o método **ELM-reg**, o parâmetro λ foi selecionado por validação cruzada do tipo $10 - fold$, conforme a metodologia apresentada por Huang et al. (2012).

Para cada uma das sete bases de dados foram realizadas 30 repetições, com divisão entre conjunto de teste e treinamento na proporção 70% para treinamento e 30% para teste. Foram treinadas Redes ELM com 10, 30, 100, 500 e 1000 neurônios na camada escondida.

As bases de dados utilizadas na condução dos experimentos foram obtidas no repositório

UCI (Dheeru & Karra Taniskidou, 2017), são elas: *Australian Credit* (acr), *QSAR biodegradation* (bio), *BUPA Liver Disorders* (bld), Statlog Heart (hea), *Pima Indian Diabetes* (pid), *Congressional Voting Records Data Set* (vot) e *Wisconsin Breast Cancer* (wbc).

As características de cada uma das bases são apresentadas na Tabela 4.1.

Tabela 4.1: Bases de dados Estudadas

Base de Dados	Amostras	Atributos
acr	690	14
bio	1055	41
bld	345	6
hea	270	13
pid	768	8
vot	435	16
wbc	683	10

Todos os testes apresentados neste trabalho foram executados em um *notebook* Dell Inspiron, com sistema operacional Debian, processador Intel Core i7 4510U, com 8GB de memória RAM. Todos os testes foram realizados apenas com o *Desktop Environment* e o software RStudio abertos. Foi descartada a pior medida de tempo para cada uma das abordagens.

Os testes foram todos realizados utilizando-se a linguagem de programação R (R Core Team, 2015).

As Tabelas 4.2 e 4.3 apresentam os resultados obtidos para as duas estratégias testadas de obtenção de λ comparados aos resultados obtidos quando se seleciona o valor de λ por validação cruzada (**ELM-reg**). A Tabela 4.2 apresenta os resultados com relação à acurácia de classificação e a Tabela 4.3 apresenta os resultados obtidos para o tempo de execução do processo de treinamento. Os maiores valores de acurácia e os menores valores de tempo estão destacados em negrito.

Foram realizados testes estatísticos comparando o desempenho (acurácia de teste e tempo de execução) dos métodos propostos (**ELM-sil** e **ELM-fis**) com o **ELM-reg**. Para cada uma das medidas de desempenho, foram realizados dois *t-testes* subsequentes (Montgomery,

2017), com correção de Bonferroni (McDonald, 2009) para o valor de $\alpha = 0.05$, o que leva a $\alpha_{bonferroni} = 0.025$.

Para todos os testes, a hipótese nula consistiu em:

$$H_0 : \mu_{ELM-reg} = \mu_{ELM-sil} = \mu_{ELM-fis}$$

Já a hipótese alternativa, consistiu em:

$$\begin{cases} H_{A1} : \mu_{ELM-reg} > \mu_{ELM-sil} \\ H_{A2} : \mu_{ELM-reg} > \mu_{ELM-fis} \end{cases} \quad (4.1)$$

Em termos de acurácia de teste, a técnica **ELM-sil** foi diferente (inferior) da técnica **ELM-reg** com significância estatística. Já para a técnica **ELM-fis**, não foram observados indícios suficientes para a rejeição da hipótese nula de igualdade. Os resultados (média das diferenças e *p-valor*) podem ser observados na Tabela 4.4.

Uma justificativa possível, no entanto, para a utilização do método ELM-sil ao invés do método ELM-reg, consiste no ganho em termos de velocidade (especialmente para um maior número de neurônios) ao custo de uma perda pequena em termos de acurácia: em média, quando se utilizou a regularização a partir da silhueta, a acurácia foi 0.65% pior.

Para a estratégia baseada no critério de Fisher, é possível que resultados melhores sejam obtidos com a ortogonalização da matriz **H**. Já para a abordagem baseada em Silhueta, será interessante buscar uma função $\lambda(sil)$ melhor que aquela apresentada na Equação 3.5, que, como mencionado, foi obtida de forma empírica e mostrou resultados promissores, mas ainda não equivalentes àqueles obtidos por validação cruzada.

Tabela 4.2: Acurácia de teste ($\%media \pm \sigma$)

p	ELM-reg	ELM-sil	ELM-fis
acr			
10	84.7 ± 2.7	83.7 ± 2.7	84.4 ± 1.8
30	85.7 ± 2.0	86.4 ± 2.2	86.4 ± 2.0
100	85.5 ± 2.7	86.0 ± 2.2	86.6 ± 2.2
500	86.2 ± 2.1	85.6 ± 2.1	86.5 ± 1.9
1000	86.8 ± 1.9	85.4 ± 1.5	85.5 ± 1.9
bio			
10	74.9 ± 3.6	74.8 ± 4.1	75.8 ± 4.4
30	84.0 ± 2.0	83.5 ± 2.1	83.7 ± 2.6
100	84.6 ± 1.8	85.7 ± 1.7	85.7 ± 1.6
500	85.7 ± 1.5	85.4 ± 2.4	87.1 ± 1.4
1000	*	84.9 ± 3.8	86.7 ± 1.9
bld			
10	71.7 ± 4.6	67.6 ± 3.5	67.4 ± 4.3
30	71.3 ± 3.9	68.8 ± 4.6	68.4 ± 3.6
100	72.3 ± 3.7	71.8 ± 5.5	70.3 ± 3.2
500	72.4 ± 4.1	73.8 ± 3.6	73.7 ± 3.7
1000	72.4 ± 3.9	70.8 ± 3.9	71.4 ± 3.3
hea			
10	81.7 ± 4.0	79.2 ± 5.8	77.9 ± 4.6
30	83.7 ± 3.4	82.5 ± 2.5	83.3 ± 2.9
100	83.3 ± 3.9	82.6 ± 4.0	82.4 ± 3.1
500	83.7 ± 3.0	81.0 ± 3.8	82.7 ± 3.6
1000	83.7 ± 3.9	81.1 ± 4.5	83.3 ± 3.3
pid			
10	76.9 ± 2.5	76.4 ± 2.4	75.9 ± 2.7
30	76.8 ± 2.4	76.9 ± 2.0	77.7 ± 2.2
100	75.6 ± 2.4	76.1 ± 2.1	77.3 ± 2.3
500	*	76.7 ± 2.0	77.2 ± 2.4
1000	76.5 ± 2.5	76.7 ± 2.6	76.6 ± 2.2
vot			
10	89.5 ± 3.2	88.3 ± 3.2	90.6 ± 3.2
30	95.2 ± 2.0	94.5 ± 1.5	94.4 ± 1.5
100	93.9 ± 1.9	94.4 ± 1.8	95.2 ± 1.5
500	95.7 ± 1.5	92.7 ± 2.2	94.9 ± 1.3
1000	95.4 ± 1.6	93.5 ± 1.6	94.1 ± 1.9
wbc			
10	95.7 ± 1.5	95.3 ± 1.4	94.6 ± 1.9
30	96.6 ± 1.2	96.3 ± 1.0	96.0 ± 1.2
100	96.3 ± 1.1	96.5 ± 1.0	96.3 ± 1.1
500	96.4 ± 0.9	96.0 ± 1.0	96.5 ± 1.1
1000	96.4 ± 1.1	96.2 ± 1.0	96.5 ± 1.2

Tabela 4.3: Tempo de treinamento (em segundos)

p	ELM-reg	ELM-sil	ELM-fis
acr			
10	2.06 ± 0.17	0.006 ± 0.002	0.001 ± 0.000
30	3.03 ± 0.08	0.011 ± 0.002	0.004 ± 0.000
100	12.39 ± 0.65	0.046 ± 0.004	0.023 ± 0.002
500	365.28 ± 12.64	0.820 ± 0.016	0.68 ± 0.016
1000	2515.78 ± 33.50	4.957 ± 0.061	4.69 ± 0.040
bio			
10	2.55 ± 0.39	0.015 ± 0.006	0.002 ± 0.000
30	4.11 ± 0.33	0.023 ± 0.005	0.007 ± 0.000
100	16.44 ± 0.73	0.086 ± 0.005	0.035 ± 0.002
500	431.44 ± 3.07	1.091 ± 0.019	0.805 ± 0.020
1000	*	7.861 ± 0.043	5.385 ± 0.026
bld			
10	1.83 ± 0.35	0.002 ± 0.000	0.001 ± 0.000
30	2.61 ± 0.42	0.003 ± 0.000	0.004 ± 0.002
100	8.89 ± 1.11	0.017 ± 0.001	0.019 ± 0.004
500	277.67 ± 4.41	0.549 ± 0.017	0.546 ± 0.011
1000	2135.44 ± 10.03	4.110 ± 0.261	4.223 ± 0.039
hea			
10	1.78 ± 0.26	0.001 ± 0.000	0.001 ± 0.000
30	2.28 ± 0.36	0.003 ± 0.000	0.003 ± 0.001
100	8.06 ± 0.39	0.015 ± 0.001	0.017 ± 0.001
500	269.89 ± 4.85	0.512 ± 0.007	0.518 ± 0.013
1000	2119.78 ± 32.10	4.075 ± 0.013	4.084 ± 0.024
pid			
10	2.03 ± 0.08	0.005 ± 0.000	0.002 ± 0.000
30	2.89 ± 0.22	0.011 ± 0.001	0.005 ± 0.000
100	12.89 ± 0.70	0.055 ± 0.008	0.025 ± 0.001
500	*	0.820 ± 0.010	0.688 ± 0.018
1000	2545.00 ± 27.96	5.319 ± 0.063	4.804 ± 0.019
vot			
10	1.94 ± 0.17	0.004 ± 0.004	0.001 ± 0.000
30	2.67 ± 0.35	0.006 ± 0.001	0.005 ± 0.001
100	9.89 ± 0.55	0.024 ± 0.002	0.021 ± 0.003
500	303.67 ± 3.83	0.622 ± 0.002	0.567 ± 0.005
1000	2265.17 ± 43.49	4.346 ± 0.034	4.258 ± 0.027
wbc			
10	2.11 ± 0.33	0.008 ± 0.005	0.001 ± 0.000
30	3.00 ± 0.43	0.010 ± 0.001	0.004 ± 0.000
100	11.61 ± 0.49	0.050 ± 0.013	0.023 ± 0.001
500	344.89 ± 2.19	0.778 ± 0.016	0.737 ± 0.132
1000	2431.72 ± 9.55	4.987 ± 0.053	4.660 ± 0.043

Tabela 4.4: Resultados do teste t para a acurácia de teste

Estratégia	Média	p -valor
(ELM-reg - ELM-fis)	0.0020	0.234
(ELM-reg - ELM-sil)	0.0065	0.004

Capítulo 5

Considerações Finais

Neste trabalho, foram apresentadas duas metodologias para seleção de parâmetros de regularização de Tikhonov de *Extreme Learning Machines*. As metodologias fundamentam-se na aplicação do Teorema de Cover às matrizes de projeção geradas pelas ELM. As metodologias propostas visam à seleção do parâmetro de regularização de forma automática e independente de validação cruzada a partir de informações de separabilidade linear extraídas da matriz de projeção gerada pela camada oculta das redes ELM.

Duas medidas foram utilizadas para se estabelecer o grau de separabilidade linear (*Fisher-Score* e Silhueta) e foram estabelecidas relações entre as medidas e o parâmetro de regularização. Ambas as medidas foram aplicadas para a separação de bases de dados reais e os resultados obtidos foram comparados aos resultados obtidos quando o parâmetro de regularização é escolhido por validação cruzada, metodologia considerada controle.

A relação entre o valor de Silhueta encontrado e o valor do parâmetro de regularização foi desenvolvida de forma empírica para bases de dados sintéticas (de fácil visualização) e não apresenta, no momento, uma justificativa teórica para sua aplicação. Já a relação estabelecida entre o critério de Fisher e o parâmetro de regularização, embora também tenha sido desenvolvida empiricamente para bases sintéticas, pode ser melhor justificada em níveis teóricos, já que, como mostrado no Capítulo 3, trata-se de uma relação que resulta (dependendo de suposições relativamente simples a respeito da matriz covariância dos da-

dos da matriz de projeção) em uma medida similar à medida de separabilidade linear já desenvolvida na literatura.

A seleção dos parâmetros de regularização sem a necessidade de validação cruzada é, como esperado, muito mais rápida (de 470 a 2000 vezes) e, em termos de acurácia de classificação, parece promissora. A metodologia baseada no critério de Fisher levou a resultados sem perda de acurácia estatisticamente significativa quando comparados aos resultados obtidos por validação cruzada. Já a estratégia baseada em silhueta levou a resultados estatisticamente piores que a metodologia controle, mas, ainda assim, muito próximos (menos de 1% piores em média).

Para trabalhos futuros, restam algumas frentes a serem abordadas. A primeira é a necessidade de se testar as metodologias em um número maior de bases de dados, para se atestar, de fato, a capacidade das propostas de gerar bons resultados de forma consistente. Uma segunda via a ser explorada diz respeito à busca por uma maior formalização nas metodologias, especialmente naquela relacionada à silhueta, que, embora apresente resultados promissores, ainda é, neste momento, fundamentalmente empírica.

Referências Bibliográficas

- P. L. Bartlett. For valid generalization the size of the weights is more important than the size of the network. In *Advances in neural information processing systems*, pages 134–140, 1997.
- A. Ben-Israel & Y. Levin. The geometry of linear separability in data sets. *Linear algebra and its applications*, 416(1):75–87, 2006.
- C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.
- T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334, 1965.
- W. Deng, Q. Zheng, & L. Chen. Regularized extreme learning machine. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, pages 389–395. IEEE, 2009.
- D. Dheeru & E. Karra Taniskidou. UCI machine learning repository, 2017.
- R. O. Duda, P. E. Hart, & D. G. Stork. *Pattern classification*. Wiley, New York, 1973.
- D. Elizondo. The linear separability problem: Some testing methods. *IEEE Transactions on neural networks*, 17(2):330–344, 2006.

- T.-c. Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.
- A. Grüning & S. M. Bohte. Spiking neural networks: Principles and challenges. In *ESANN*, 2014.
- S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.
- G.-B. Huang, Q.-Y. Zhu, & C.-K. Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990. IEEE, 2004.
- G.-B. Huang, L. Chen, C. K. Siew, et al. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Networks*, 17(4):879–892, 2006a.
- G.-B. Huang, Q.-Y. Zhu, & C.-K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006b.
- G.-B. Huang, H. Zhou, X. Ding, & R. Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2012.
- H. Jaeger. The “echo state” approach to analysing and training recurrent neural networks—with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001a.
- H. Jaeger. *Short term memory in echo state networks*, volume 5. GMD-Forschungszentrum Informationstechnik, 2001b.
- G. James, D. Witten, T. Hastie, & R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.
- L. Kaufman & P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

- S. Kok. *Liquid state machine optimization*. PhD thesis, Master Thesis, Utrecht University, 2007.
- J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964a.
- J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964b.
- W. Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.
- W. Maass, T. Natschläger, & H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.
- J. H. McDonald. *Handbook of biological statistics*, volume 2. Sparky House Publishing Baltimore, MD, 2009.
- Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, & A. Lendasse. Op-elm: optimally pruned extreme learning machine. *IEEE transactions on neural networks*, 21(1):158–162, 2010.
- Y. Miche, M. Van Heeswijk, P. Bas, O. Simula, & A. Lendasse. Trop-elm: a double-regularized elm using lars and tikhonov regularization. *Neurocomputing*, 74(16):2413–2421, 2011.
- D. C. Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.
- J. M. Nageswaran, N. Dutt, J. L. Krichmar, A. Nicolau, & A. V. Veidenbaum. A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors. *Neural networks*, 22(5-6):791–800, 2009.
- H. Paugam-Moisy & S. Bohte. Computing with spiking neuron networks. In *Handbook of natural computing*, pages 335–376. Springer, 2012.

- T. Poggio & F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.
- R. V. Rullen & S. J. Thorpe. Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural computation*, 13(6):1255–1283, 2001.
- L. J. Silvestre, A. P. Lemos, J. P. Braga, & A. P. Braga. Dataset structure as prior information for parameter-free regularization of extreme learning machines. *Neurocomputing*, 169:288–294, 2015.
- S. Thorpe, A. Delorme, & R. Van Rullen. Spike-based strategies for rapid processing. *Neural networks*, 14(6-7):715–725, 2001.
- A. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Meth. Dokl.*, 4:1035–1038, 1963.
- V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.
- H. F. d. C. Velho. *Introdução aos problemas inversos: Aplicações em pesquisa espacial*, 2008.

Apêndices

Os dois apêndices apresentados a seguir correspondem a ideias que foram desenvolvidas ao longo do período de dois anos e ficam como propostas de trabalhos futuros

A *Multidimensional Scaling* e Treinamento de Redes Neurais

O problema de *Multidimensional Scaling* consiste em representar n objetos em t dimensões, de modo que as distâncias dos pontos na representação sejam o mais próximo possível das dissimilaridades experimentais dos objetos (Kruskal, 1964a).

Segundo Kruskal (1964a), o problema pode ser resolvido a partir da minimização de um critério chamado de *Stress*. O *Stress* é definido como a soma dos quadrados das diferenças (resíduos) entre os valores de distâncias reais entre os pontos e os valores de distância observados na representação.

Em um segundo trabalho, Kruskal (1964b) detalha o método numérico para minimização da medida de *Stress*. A estratégia de otimização adotada consiste na aplicação do Gradiente Descendente, com gradientes obtidos de forma analítica. As limitações conhecidas do Gradiente Descendente, especificamente a presença de mínimos locais e a demora para convergência, são verificadas também na execução do *Multidimensional Scaling*.

A técnica de *Multidimensional Scaling* é comumente utilizada para a resolução de problemas de redução de dimensionalidade, como uma alternativa às técnicas baseadas em componentes principais.

A.1 Formulação Matemática

O *Stress*, a ser minimizado, é definido de acordo com a Eq. (1).

$$S = \sqrt{\frac{(d_{ij} - \hat{d}_{ij})^2}{\sum d_{ij}^2}} \quad (1)$$

Na Eq. (1), d_{ij} representa uma medida da distância entre os pontos i e j nos dados reais, enquanto que \hat{d}_{ij} representa a medida de distância entre os pontos no espaço projetado, t -dimensional.

O problema de *Multidimensional Scaling* pode, então, ser formalizado como um problema de otimização definido pela Eq. (2).

$$\arg \min_{\hat{d}_{ij}} S \quad (2)$$

Deseja-se encontrar o conjunto de distâncias \hat{d}_{ij} que minimiza o valor de *Stress*, o que corresponde a encontrar um conjunto de vetores \hat{x}_i no espaço projetado que leve a tais distâncias.

A.2 Treinamento de redes Neurais a partir do MDS

A ideia desenvolvida neste trabalho consiste em tratar o problema de treinamento de redes neurais como um problema de projeção em alta dimensão (Cover, 1965), para o qual deseja-se conservar a estrutura de uma matriz de distâncias conforme o algoritmo de MDS.

Dada uma matriz de distâncias $\mathbf{M}_{N \times N}$, construída a partir de N amostras, separáveis, deseja-se encontrar uma matriz $\mathbf{D}_{N \times N}$, construída a partir da matriz de projeção de uma rede neural ($\mathbf{H}_{N \times p}$), de forma que seja minimizada a função de perda dada pela Eq.(3).

$$L(\mathbf{M}, \mathbf{D}) = \frac{1}{2} \sum_{i=2}^N \sum_{j=1}^{i-1} (m_{ij}^2 - d_{ij}^2)^2 \quad (3)$$

Para construção da matriz de projeção \mathbf{H} considera-se a topologia de rede das ELM, de forma que, dada uma matriz de dados de entrada $\mathbf{X}_{N \times n}$, e uma matriz de pesos $\mathbf{Z}_{p \times n}$, a matriz de projeção será dada pela Eq. (4).

$$\mathbf{H}_{N \times p} = \tanh(\mathbf{Z}\mathbf{V}^T) \quad (4)$$

Considerando-se, um caso hipotético em que $N = 3$, $n = 2$ e $p = 2$, as matrizes \mathbf{X} , \mathbf{Z} e \mathbf{H} serão dadas por:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix}$$

$$\mathbf{Z} = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} \tanh(x_{11}z_{11} + x_{12}z_{12}) & \tanh(x_{11}z_{21} + x_{12}z_{22}) \\ \tanh(x_{21}z_{11} + x_{22}z_{12}) & \tanh(x_{21}z_{21} + x_{22}z_{22}) \\ \tanh(x_{31}z_{11} + x_{32}z_{12}) & \tanh(x_{31}z_{21} + x_{32}z_{22}) \end{bmatrix}$$

A Eq. (3) pode ser reescrita, portanto, como:

$$L = \frac{1}{2} \left[(m_{21}^2 - d_{21}^2)^2 + (m_{31}^2 - d_{31}^2)^2 + (m_{32}^2 - d_{32}^2)^2 \right] \quad (5)$$

$$\begin{aligned} L = \frac{1}{2} & \left([m_{21}^2 - (h_{21} - h_{11})^2 - (h_{22} - h_{12})^2]^2 \right. \\ & + [m_{31}^2 - (h_{31} - h_{11})^2 - (h_{32} - h_{12})^2]^2 \\ & \left. + [m_{32}^2 - (h_{31} - h_{21})^2 - (h_{32} - h_{22})^2]^2 \right) \end{aligned} \quad (6)$$

$$\begin{aligned} L = \frac{1}{2} & \left([m_{21}^2 - (\tanh(x_{21}z_{11} + x_{22}z_{12}) - \tanh(x_{11}z_{11} + x_{12}z_{12}))^2 \right. \\ & \quad \left. - (\tanh(x_{21}z_{21} + x_{22}z_{22}) - \tanh(x_{11}z_{21} + x_{12}z_{22}))^2]^2 \right. \\ & + [m_{31}^2 - (\tanh(x_{31}z_{11} + x_{32}z_{12}) - \tanh(x_{11}z_{11} + x_{12}z_{12}))^2 \\ & \quad \left. - (\tanh(x_{31}z_{21} + x_{32}z_{22}) - \tanh(x_{11}z_{21} + x_{12}z_{22}))^2]^2 \right. \\ & \left. + [m_{32}^2 - (\tanh(x_{31}z_{11} + x_{32}z_{12}) - \tanh(x_{21}z_{11} + x_{22}z_{12}))^2 \right. \\ & \quad \left. - (\tanh(x_{31}z_{21} + x_{32}z_{22}) - \tanh(x_{21}z_{21} + x_{22}z_{22}))^2]^2 \right) \end{aligned} \quad (7)$$

Assim como, no problema de MDS, deseja-se minimizar o *Stress*, deseja-se neste caso, minimizar a função de perda L . Como a matriz de projeção \mathbf{H} é construída em função dos pesos, minimizar a função de perda é equivalente a encontrar o conjunto ótimo de pesos. O ótimo para a função L pode ser encontrado buscando-se soluções a partir do termo de atualização $\Delta z_{kl} = -\eta \frac{\partial L}{\partial z_{kl}}$, em que η é um termo de passo.

Para o exemplo trabalhado, calcula-se $\frac{\partial L}{\partial z_{21}}$:

$$\begin{aligned}
\frac{\partial L}{\partial z_{21}} = & \frac{1}{2} \frac{\partial}{\partial z_{21}} \left[m_{21}^2 - (\tanh(x_{21}z_{11} + x_{22}z_{12}) - \tanh(x_{11}z_{11} + x_{12}z_{12}))^2 \right. \\
& \left. - (\tanh(x_{21}z_{21} + x_{22}z_{22}) - \tanh())^2 \right]^2 \\
& + \frac{1}{2} \frac{\partial}{\partial z_{21}} \left[m_{31}^2 - (\tanh(x_{31}z_{11} + x_{32}z_{12}) - \tanh(x_{11}z_{11} + x_{12}z_{12}))^2 \right. \\
& \left. - (\tanh(x_{31}z_{21} + x_{32}z_{22}) - \tanh(x_{11}z_{21} + x_{12}z_{22}))^2 \right]^2 \\
& + \frac{1}{2} \frac{\partial}{\partial z_{21}} \left[m_{32}^2 - (\tanh(x_{31}z_{11} + x_{32}z_{12}) - \tanh(x_{21}z_{11} + x_{22}z_{12}))^2 \right. \\
& \left. - (\tanh(x_{31}z_{21} + x_{32}z_{22}) - \tanh(x_{21}z_{21} + x_{22}z_{22}))^2 \right]^2
\end{aligned} \tag{8}$$

A aplicação repetida da regra da cadeia leva ao seguinte resultado:

$$\begin{aligned}
\frac{\partial L}{\partial z_{21}} = & (-2)(m_{21}^2 - d_{21}^2)(h_{22} - h_{12})[x_{21} \operatorname{sech}^2(x_{21}z_{21} + x_{22}z_{22}) \\
& - x_{11} \operatorname{sech}^2(x_{11}z_{21} + x_{12}z_{22})] \\
& + (-2)(m_{31}^2 - d_{31}^2)(h_{32} - h_{12})[x_{31} \operatorname{sech}^2(x_{31}z_{21} + x_{32}z_{22}) \\
& - x_{11} \operatorname{sech}^2(x_{11}z_{21} + x_{12}z_{22})] \\
& + (-2)(m_{32}^2 - d_{32}^2)(h_{32} - h_{22})[x_{31} \operatorname{sech}^2(x_{31}z_{21} + x_{32}z_{22}) \\
& - x_{21} \operatorname{sech}^2(x_{21}z_{21} + x_{22}z_{22})]
\end{aligned} \tag{9}$$

A partir da Eq. (9), é possível extrair uma fórmula geral para atualização dos pesos, para casos gerais de dimensionalidade das matrizes \mathbf{X} e \mathbf{Z} :

$$\begin{aligned}
\frac{\partial L_1}{\partial z_{kl}} = & (-2) \sum_{i=2}^N \sum_{j=1}^{i-1} (m_{ij}^2 - d_{ij}^2)(h_{ik} - h_{jk}) \\
& [x_{il} \operatorname{sech}^2(\mathbf{X}_i \cdot \mathbf{Z}_p^T) - x_{jl} \operatorname{sech}^2(\mathbf{X}_j \cdot \mathbf{Z}_p^T)]
\end{aligned} \tag{10}$$

A.3 Resultados Preliminares

A metodologia proposta foi utilizada para o treinamento de uma rede neural do tipo ELM. A matriz \mathbf{X} foi gerada a partir de duas espirais, enquanto a matriz \mathbf{M} , imposta, foi calculada

a partir de duas distribuições normais com sobreposição (a presença de sobreposição parece ter um efeito de regularização, semelhante à regularização por acréscimo de ruído proposta por Bishop (1995)).

A estrutura de rede neural utilizada consistiu de duas *Extreme Learning Machines* com 10 neurônios na camada oculta. Para uma delas, os pesos foram apenas gerados de forma aleatória, como devem ser as ELM. Para a outra rede, os pesos, após serem gerados, foram ajustados com o gradiente proposto na Eq. (10). Utilizou-se o método do gradiente descendente para minimização da função de perda apresentada na Eq. (3).

O resultado preliminar obtido para a superfície de separação é apresentado na Figura A1.

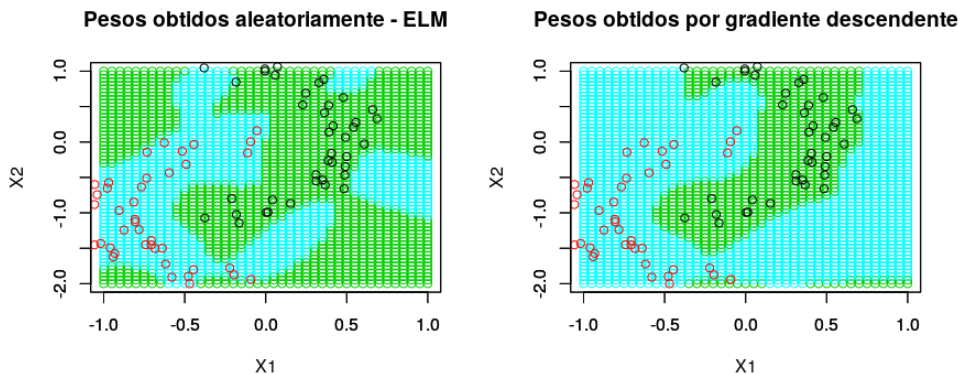


Figura A1: Superfícies de Decisão obtidas com ELM e com pesos obtidos pelo gradiente descendente considerando-se as matrizes de distância

Pela Figura A1, tanto o método apresentado para cálculo dos pesos quanto a ELM com pesos iniciados aleatoriamente foram capazes de separar os dados. No entanto, o método proposto parece gerar uma superfície de separação mais suave (*i.e.* regularizada) que a superfície gerada pela ELM. Tal resultado é particularmente promissor e merece ser mais explorado.

Ainda não foi possível solucionar os problemas de implementação (principalmente a ocorrência frequente de mínimos locais e a demora no treinamento), e, portanto, não foram realizados testes em mais bases de dados.

B Regularização com Memória Associativa de Extreme Learning Machines de Neurônios de Pulso

Modelos tradicionais de neurônio levam em consideração a taxa de disparos do neurônios, mas não consideram nem a presença nem o momento dos disparos individualmente, sendo que a importância de ambos já foi demonstrada para a representação de dinâmica (Rullen & Thorpe, 2001). Uma representação alternativa de neurônios consiste em codificar a informação como pulsos, dos quais tanto a presença quanto o momento de ocorrência são significativos (Paugam-Moisy & Bohte, 2012). O processo sináptico, temporal em sua natureza, é melhor representado pelos chamados trens de pulso, estruturas utilizadas em redes neurais pulsadas no lugar dos vetores reais para transmissão da informação. Neurônios de pulso são capazes de transmitir uma quantidade de informação muito superior às representações clássicas (Thorpe et al., 2001).

Assim como as redes neurais do tipo ELM, as redes de Estado de Eco (ESN) (Jaeger, 2001a) e as Máquinas de Estado Líquido (Maass et al., 2002) são construídas com inicialização aleatória de pesos. Tanto as ESN quanto as LSM são parte do campo conhecido como Computação de Reservatório, e são estabelecidas ao redor de uma estrutura recursiva de alta dimensionalidade, conhecida como reservatório, que é responsável por projetar os dados de entrada em um espaço não linear de maior dimensionalidade. A camada de saída das ESN e LSM é tipicamente linear, e, assim como acontece nas ELM, trata-se da única camada treinada. A alta dimensionalidade dos modelos garante uma alto poder computacional. Uma outra característica importante das máquinas de reservatório é sua memória de curta duração (Jaeger, 2001b), que confere aos modelos a capacidade de lidar de forma muito natural com séries temporais (Jaeger, 2001b; Maass et al., 2002). As redes ESN e LSM diferem entre si no tipo de neurônio utilizado nas computações do reservatório: as ESN utilizam neurônios sigmoidais e as LSM utilizam neurônios de pulso.

O principal empecilho na aplicação da Computação de Reservatório está relacionado ao alto custo computacional envolvido (Kok, 2007). Sobretudo quando se trabalha com LSM

e neurônios de pulso, o custo dos cálculos envolvidos na construção do reservatório torna-se proibitivo.

Este trabalho tem como objetivo a criação de um modelo baseado em *Extreme Learning Machines*, e que herde, portanto, sua velocidade de treinamento, mas que incorpore, também, propriedades de LSMs e ESNs, de forma que a rede neural resultante seja capaz de lidar de forma simples com dados temporais. Como mostrado por Silvestre et al. (2015), é possível incorporar informação estrutural (representada por uma matriz de afinidade) no treinamento de ELMs de forma que é obtida uma rede regularizada sem necessidade de parâmetros. Neste trabalho, constrói-se o argumento de que memórias associativas podem ser utilizadas da mesma maneira.

É proposta uma implementação de ELM construída a partir de neurônios de pulso, cujos neurônios de saída são regularizados pela incorporação de uma matriz de memória associativa. A natureza temporal dos neurônios de pulso combinada à memória associativa fornecem à ELM modificada a capacidade de lidar com dados temporais, o que leva a um modelo de rápido treinamento, adequado para a classificação de séries temporais.

B.1 Motivação

Durante a década de 1990, diversos trabalhos mostraram a capacidade de aproximação de redes neurais constituídas por neurônios de pulso (Redes Neurais Pulsadas, ou, do inglês, SNN) (Maass, 1997). Tais redes apresentam as mesmas (ou até melhores) propriedades de aproximação que as redes mais tradicionais. Não faltam evidências de que os neurônios de pulso são modelos mais precisos dos neurônios reais que os neurônios baseados em taxa de disparo (como o linear ou o sigmoidal) (Rullen & Thorpe, 2001).

No entanto, o desenvolvimento de Redes Neurais Pulsadas foi limitado pelo custo computacional envolvido nos cálculos necessários para simulações de larga escala (Nageswaran et al., 2009). Ainda que existam diferentes modelos de neurônios de pulso, com diferentes níveis de complexidade, o treinamento de Redes Neurais Pulsadas demanda a solução de equações diferenciais, o que torna os modelos muito caros com o crescimento das redes.

Além do tempo de computação dos neurônios de pulso, uma outra dificuldade relativa às SNNs diz respeito ao treinamento eficiente das redes (Grüning & Bohte, 2014). Algoritmos de treinamento desenvolvidos para as redes neurais tradicionais, como o algoritmo de propagação reversa do erro, não são diretamente aplicáveis para SNNs. Muitos algoritmos de treinamento de SNNs foram propostos ao longo dos últimos 20 anos, com diferentes níveis de complexidade. No entanto, todos os algoritmos já propostos apresentam alguma limitação, seja decorrente de simplificações necessárias, seja decorrente de custos proibitivos (Grüning & Bohte, 2014).

Extreme Learning Machines apresentam uma topologia de rede interessante para se trabalhar com neurônios de pulso. O treinamento é extremamente simples e rápido, uma vez que os pesos da camada escondida são gerados aleatoriamente e o treinamento é realizado apenas para os pesos da camada de saída linear (Huang et al., 2006b).

Ainda que ELMs sejam similares a ESNs e LSMs, é fundamental observar uma diferença: as máquinas de reservatório apresentam memória de curta duração (Kok, 2007). Tal memória é uma das propriedades das máquinas de reservatório que as torna adequadas ao processamento de dados temporais. É fundamental que o modelo proposto seja capaz de incorporar tal propriedade para que possa lidar com dados temporalmente correlacionados. A proposta deste trabalho consiste em combinar uma memória associativa com uma ELM composta de neurônios de pulso. A informação contida na matriz de memória associativa é responsável por suavizar e regularizar a resposta da rede.

A rede neural pulsada proposta é treinada a partir das mesmas equações utilizadas no trabalho de Silvestre et al. (2015). A principal diferença reside na matriz utilizada para transformar a matriz de projeção \mathbf{H} . Neste trabalho, é utilizada uma matriz de memória \mathbf{M} , obtida dos dados de entrada, enquanto que no trabalho de Silvestre et al. (2015), é utilizada uma matriz de afinidade \mathbf{P} . A combinação das propriedades de uma ELM com a natureza temporal de neurônios de pulso e matrizes de memória leva a uma possibilidade de aplicação na classificação de séries temporais.

A formulação da matriz de memória a ser utilizada é, ainda, um desafio, já que, para que a

utilização da matriz seja equivalente à regularização, é necessário que algumas características sejam respeitadas, como pode ser visto no trabalho de Silvestre et al. (2015).

O problema de classificação de séries temporais é recorrente em diferentes áreas de conhecimento, de séries financeiras a sinais médicos, incluído reconhecimento de fala e gestos (Fu, 2011). Uma área de interesse que se destaca é a área de sinais médicos, em que, por exemplo, tem-se os exames de eletrocardiografia (ECG) e eletroencefalografia (EEG).

Para a presente proposta de trabalho ainda não existem resultados a serem apresentados.