Aplicação de Máquinas de Aprendizado Extremo ao Problema de Aprendizado Ativo

Euler Guimarães Horta

Aplicação de Máquinas de Aprendizado Extremo ao Problema de Aprendizado Ativo

Euler Guimarães Horta

Orientador: *Prof. Dr. Antônio de Pádua Braga*

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

UFMG - Belo Horizonte
Outubro de 2015

Horta, Euler Guimarães.
H821a Aplicação de máquinas de aprendizado extremo ao problema de aprendizado ativo [manuscrito] / Euler Guimarães Horta. - 2015.
x, 87 f., enc.: il.

Orientador: Antônio de Pádua Braga.

Tese (doutorado) Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f. 81-87.

1. Engenharia elétrica - Teses. 2. Perceptrons - Teses. I. Braga, Antônio de Pádua. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

Agradecimentos

Agradeço primeiramente a Deus pelas inúmeras oportunidades que tive em minha vida e pela proteção nestes árduos anos de doutoramento. Aos meus pais, Waldir e Iranilde, e minha irmã Michelle, que sempre me deram carinho e apoio em todos os momentos que precisei e que sempre me incentivaram a estudar e vencer desafios. A minha noiva Francisca pelo amor, carinho, paciência e compreensão, sobretudo no desgastante final desta tese. Ao meu amigo Ulisses pelos vários conselhos. Ao Braga pelos 11 anos de orientação e paciência (da iniciação científica ao doutorado!). Ao Cristiano pelo apoio que foi fundamental no final desta tese. A todos os colegas do LITC pelos bate-papos e convivência. À UFMG pela oportunidade de estudar gratuitamente em cursos de altíssima qualidade (graduação, mestrado e doutorado). Ao PPGEE por ter me aceitado no programa e por ter concedido a prorrogação do prazo para defesa desta tese. Ao Instituto de Ciência e Tecnologia da UFVJM que me concedeu o afastamento no último semestre do doutoramento, que foi fundamental para a elaboração deste texto. A todos os colegas da UFVJM que me apoiaram neste desafio de cursar o doutorado trabalhando.

Resumo

O Aprendizado Ativo tem o objetivo de escolher apenas os padrões mais informativos para rotulação e aprendizado. No Aprendizado Ativo uma estratégia é utilizada para analisar um padrão não rotulado e decidir se deve ou não ter o seu rótulo solicitado a um especialista. Em geral essa rotulação tem um custo elevado, o que motiva o estudo de técnicas que minimizem o número de rótulos necessários para o aprendizado. As abordagens tradicionais de Aprendizado Ativo geralmente fazem algumas considerações irreais em relação aos dados, como exigir separabilidade linear ou que a distribuição dos dados seja uniforme. Além disso, os modelos tradicionais necessitam de um processo de ajuste fino de parâmetros, o que exige que rótulos sejam reservados para esse fim, aumentando os custos do processo. Nesta tese são apresentadas duas estratégias de Aprendizado Ativo que não fazem nenhuma consideração quanto aos dados e que não necessitam de ajuste fino de parâmetros. Os algoritmos propostos são baseados em Máquinas de Aprendizado Extremo (*Extreme Learning Machines - ELM*) e em um Perceptron com pesos normalizados treinado com aprendizado Hebbiano. As estratégias de Aprendizado Ativo decidem se um padrão deve ser rotulado utilizando um simples teste de convergência. Esse teste é obtido por meio de uma adaptação do Teorema de Convergência do Perceptron. Os modelos propostos permitem o aprendizado incremental e *online*, são práticos e rápidos, e são capazes de obter uma boa solução em termos de complexidade neural e de capacidade de generalização. Os resultados dos experimentos mostram que os modelos desenvolvidos têm desempenho similar às ELMs regularizadas e às SVMs com *kernel* ELM. Entretanto, os modelos propostos utilizam uma quantidade de rótulos muito menor, sem a necessidade de processos de otimização computacionalmente caros e sem a necessidade de ajuste fino de parâmetros.

Abstract

The main objective of Active Learning is to choose only the most informative patterns to be labeled and learned. In Active Learning scenario a selection strategy is used to analyze a non-labeled pattern and to decide whether its label should be queried to a specialist. Usually, this labeling process has a high cost, which motivates the study of strategies that minimize the number of necessary labels for learning. Traditional Active Learning approaches make some unrealistic considerations about the data, such as requiring linear separability or that the data distribution should be uniform. Furthermore, traditional approaches require fine-tuning parameters, which implies that some labels should be reserved for this purpose, increasing the costs. In this thesis we present two Active Learning strategies that make no considerations about the data distribution and that do not require fine-tuning parameters. The proposed algorithms are based on *Extreme Learning Machines* (ELM) with a Hebbian Perceptron with normalized weights in the output layer. Our strategies decide whether a pattern should be labeled using a simple convergence test. This test was obtained by adapting the Perceptron Convergence Theorem. The proposed methods allow online learning, they are practical and fast, and they are able to obtain a good solution in terms of neural complexity and generalization capability. The experimental results show that our models have similar performance to regularized ELMs and SVMs with ELM *kernel*. However, the proposed models learn a fewer number of labeled patterns without any computationally expensive optimization process and without fine-tuning parameters.

Sumário

Lista de Abreviaturas

Abreviatura	**Significado**
Ac	Acurácia
AUC	Área Abaixo da Curva ROC (*Area Under de ROC Curve*)
EALM	Máquinas de Aprendizado Ativo e Extremo (*Extreme Active Learning Machines*)
EALMSS	Máquinas de Aprendizado Ativo e Extremo do Tipo *Stream-based Selective Sampling* (*Extreme Active Learning Machines - Selective Sampling*)
EALMPB	Máquinas de Aprendizado Ativo e Extremo do Tipo *Pool-based Sampling* (*Extreme Active Learning Machines - Pool-based*)
ELM	Máquinas de Aprendizado Extremo (*Extreme Learning Machines*)
ELMPCP	Máquina de Aprendizado Extremo com Perceptron Hebbiano e aprendizado dos padrões mais próximos do separador *(ELM with Hebbian Perceptron learned using the Closest Patterns)*
ELMPRP	Máquina de Aprendizado Extremo com Perceptron Hebbiano e aprendizado dos padrões escolhidos aleatoriamente *(ELM with Hebbian Perceptron learned using Random Patterns)*
ELM2012	*Regularized Extreme Learning Machines - Versão 2012*
OS-ELM	*Online Sequential Extreme Learning Machine*
AELM	*Active Extreme Learning Machine*
AL-ELM	*Active Learning - Extreme Learning Machine*
PDKCM	Perceptron de Dasgupta *et al.* [15]
PCBGZ	Perceptron de Cesa-Bianchi *et al.* [10]
PCBGZ-OPT	Perceptron de Cesa-Bianchi *et al.* [10] com parâmetro ótimo
SVM	Máquinas de Vetores de Suporte (*Support Vector Machine*)
SVMTK	SVM de Tong e Koller [70]

Lista de Símbolos

Símbolo	**Significado**
γ	Margem de segurança
$\mathbf{x}$	Padrão ($\mathbf{x} = [x_1, x_2, ..., x_m]^T$)
$\mathbf{w}$	Vetor de pesos ($\mathbf{w} = [w_1, w_2, ..., w_m]^T$)
$bias$	Limiar de ativação do perceptron
W_{ih}	Matriz de pesos de entrada ($[\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_N]$)
p	Número de neurônios da camada escondida
N	Número de padrões de uma base de dados
H	Matriz de todos os dados de treinamento propagados através da camada escondida
$H^\dagger$	Inversa generalizada de Moore–Penrose da matriz H
$\mathbf{y}$	Vetor de rótulos dos dados de treinamento
y	Rótulo de um padrão
$K(x, x)$	Kernel

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

A indução de modelos de aprendizado supervisionado depende que um conjunto de dados suficientemente grande esteja disponível e que esse conjunto seja composto por pares $(\mathbf{x}_i, y_i)$, que representam um padrão e seu rótulo. Os padrões $\mathbf{x}_i$ são obtidos por uma amostragem do espaço de entrada χ de acordo com uma função de probabilidade $P(\mathbf{X})$, sendo que os rótulos y_i são obtidos a partir de uma função oráculo $f_g(\mathbf{x}_i)$. O objetivo final do aprendizado é obter um conjunto de parâmetros de uma função de aproximação que seja capaz de representar de forma aproximada a função oráculo. No caso de classificação binária a função de aproximação pode realizar a classificação em um processo com duas etapas, uma responsável por realizar uma transformação nos dados e outra responsável pela classificação. A etapa de transformação geralmente tem o objetivo de simplificar o problema para viabilizar o processo de classificação. Nesse contexto, o problema de classificação binária se resume a obter um conjunto de parâmetros $\mathbf{Z}$ e $\mathbf{w}$ que represente cada uma dessas etapas. Dessa forma, a função de aproximação pode ser representada por $f(\mathbf{x}, \mathbf{Z}, \mathbf{w})$ tal que $f(\mathbf{x}, \mathbf{Z}, \mathbf{w}) \approx f_g(\mathbf{x})$.

As condições de convergência que garantem que $f(\mathbf{x}, \mathbf{Z}, \mathbf{w}) \rightarrow f_g(\mathbf{x})$ em χ dependem da representatividade e do tamanho do conjunto de aprendizado $D_L = \{\mathbf{x}_i, y_i\}_{i=1}^{N_L}$. Uma rotulação confiável dos exemplos de entrada $D_u = \{\mathbf{x}_i\}_{i=1}^{N_u}$ é de fundamental importância para garantir que a função de aproximação $f(\mathbf{x}, \mathbf{Z}, \mathbf{w})$ seja robusta. Dessa forma, N_L deve ser suficientemente grande para garantir as condições de convergência.

No Aprendizado Supervisionado assume-se que o modelo a ser treinado não tem o controle da probabilidade de amostragem $P(\mathbf{X})$ e os padrões são amostrados aleatoriamente. Entretanto, o desenvolvimento de máquinas de

aprendizado que podem influenciar e até mesmo controlar $P(\mathbf{X})$ tem atraído muito interesse nos últimos anos. Esse campo da área de Aprendizado de Máquinas é conhecido como *Aprendizado Ativo* [61]. Nesse contexto, o modelo é **ativo** no sentido de que pode decidir quais padrões precisa rotular e aprender. Os padrões podem ser obtidos a partir de um fluxo de dados constante (*stream*) [11] ou utilizando um conjunto de dados de tamanho pré-definido (*pool*) [42].

A estratégia de seleção utilizada por um algoritmo de Aprendizado Ativo determina de forma direta ou indireta a probabilidade $P(\mathbf{X})$ de um padrão ser selecionado, rotulado e aprendido. O objetivo do Aprendizado Ativo é similar ao do Aprendizado Supervisionado: induzir uma função de aprendizado que seja válida para qualquer padrão de entrada e que seja o mais similar possível da função geradora de rótulos $f_g(\mathbf{x})$. A principal diferença do Aprendizado Ativo é que apenas os padrões mais informativos são utilizadas para obter a função de aproximação. Para o problema de classificação binária o Aprendizado Ativo utiliza os padrões mais informativos para obter $f(\mathbf{x},\mathbf{Z},\mathbf{w}) \rightarrow f_g(\mathbf{x})$.

As estratégias de Aprendizado Ativo são particularmente úteis quando muitos padrões não rotulados estão disponíveis, mas obter o rótulo de um padrão tem um custo muito elevado. Dessa forma, deseja-se construir modelos com o mínimo de rótulos possível. Essas estratégias utilizam algum critério que seja capaz de dizer se um padrão é mais informativo que os outros para, então, decidir se o padrão deve ou não ser rotulado. Alguns algoritmos de Aprendizado Ativo para classificação binária consideram que os padrões mais informativos são aqueles mais próximos da margem de separação entre duas classes [70, 58] se modelos como Máquinas de Vetores de Suporte (*Support Vector Machine* - SVM) [71] forem utilizados.

Aprendizado Ativo baseado em margem de separação tem sido realizado para problemas de classificação binária, considerando que os dados são linearmente separáveis no espaço de entrada [10, 15, 49]. Uma vez que um separador linear é obtido a partir de um padrão inicial, as rotulações seguintes são realizadas de acordo com algum critério pré-estabelecido, geralmente relacionado à proximidade de um padrão em relação ao separador, o que é muito simples de calcular se o separador é linear. Porém, em uma abordagem mais realista e geral um separador não linear deve ser considerado. Isso requer que uma linearização seja realizada mapeando-se os dados de entrada em um espaço de características, onde a seleção dos padrões deve ser realizada. Esse processo pode ser realizado utilizando-se um modelo de duas camadas, sendo a primeira responsável pela linearização do problema e a segunda responsável pela separação das classes devidamente linearizadas.

Assim, uma função $f(\mathbf{x},\mathbf{Z},\mathbf{w})$ deve ser composta por uma camada escon-

dida que contém uma função de mapeamento $\psi(\mathbf{x},\mathbf{Z})$ e por uma função de saída $\phi(\psi(\mathbf{x},\mathbf{Z}),\mathbf{w})$. Como cada uma das duas funções possuem apenas uma camada, a função $\phi(\cdot,\cdot)$ realiza uma separação linear e a função $\psi(\mathbf{x},\mathbf{Z})$ lineariza o problema. A maior dificuldade com a abordagem não linear é que para se obter a função $\psi(\mathbf{x},\mathbf{Z})$ geralmente parâmetros livres precisam ser finamente ajustados.

Com o objetivo de vencer a dificuldade em obter um mapeamento não linear que não necessite de ajuste fino de parâmetros livres, nesta tese será apresentado um método baseado nos princípios das Máquinas de Aprendizado Extremo (*Extreme Learning Machines* - ELM) [39] para se obter uma função de mapeamento $\psi(\mathbf{x},\mathbf{Z})$. O princípio básico das ELMs é obter aleatoriamente os elementos de $\mathbf{Z}$ e a partir disso projetar o espaço de entrada em um espaço de características de dimensão muito elevada, obtendo-se assim a função de mapeamento $\psi(\mathbf{x},\mathbf{Z})$. Na prática, o único parâmetro necessário para a projeção ELM é a dimensão (número de neurônios escondidos) do espaço de características. Esse parâmetro não precisa ser finamente ajustado, bastando escolher um valor muito maior que o tamanho do espaço de entrada [23, 26], sendo que o desempenho final do classificador não é muito sensível a esse parâmetro.

Espera-se que o mapeamento $\mathbf{H} = \psi(\mathbf{X},\mathbf{Z})$ de $\mathbf{X}$ no espaço de características transforme o problema não linear em um problema linearmente separável nesse novo espaço, contanto que o número de neurônios escondidos p seja suficientemente grande para que a dimensão do espaço projetado seja suficientemente elevada para garantir as condições de separabilidade linear do Teorema de Cover [14]. Esse teorema prevê que quanto maior for a dimensão do espaço utilizado para representar um problema de classificação, maior será a probabilidade do problema se tornar linearmente separável.

Uma vez que a matriz de mapeamento $\mathbf{H}$ é obtida, o problema de aprendizado se reduz à seleção de padrões adequados para induzir os parâmetros de um separador linear. O método da pseudoinversa de Moore-Penrose adotado pelas ELMs para obter o separador linear resulta em sobreajuste (*overfitting*) quando o número de padrões selecionados tende ao número de neurônios escondidos ($N \rightarrow p$). Nessa situação, o número de equações tende ao número de variáveis e a solução da pseudoinversa pode levar a uma solução de erro zero [39] para o conjunto de treinamento, o que causa o sobreajuste. Como no Aprendizado Ativo o tamanho do conjunto de treinamento pode alcançar o número de neurônios escondidos à medida que mais padrões são selecionados e rotulados, o efeito da solução de erro zero da pseudoinversa é indesejado porque pode resultar em um declínio no desempenho do classificador próximo do limite $N \approx p$. Em virtude disso, uma alternativa à solução da pseudoinversa deve ser adotada em problemas de Aprendizado Ativo.

Além disso, como o Aprendizado Ativo é incremental, realizar um retreinamento utilizando todo o conjunto de treinamento a cada vez que um novo padrão é selecionado pode ser proibitivo. Por isso, nesta tese é proposto o uso de um método de aprendizado incremental para substituir a solução da pseudoinversa. Além disso, o método tem um termo residual inerente que compensa a solução de erro zero da pseudoinversa.

O classificador apresentado nesta tese é composto por uma camada escondida ELM composta por muitos neurônios escondidos e uma camada de saída composta por um Perceptron com pesos normalizados [21] treinado com o aprendizado Hebbiano [31]. Duas estratégias de Aprendizado Ativo serão apresentadas, sendo que ambas são derivadas de uma adaptação do Teorema de Convergência do Perceptron [51, 30]. A primeira estratégia de Aprendizado Ativo é do tipo *stream-based* e cada padrão é analisado apenas uma vez. A segunda estratégia é do tipo *pool-based* e a distância de todos os padrões em relação ao hiperplano separador deve ser analisada para decidir qual padrão será avaliado pelo teste de convergência em cada iteração. Os classificadores obtidos pelas duas estratégias são incrementais e *on-line*, o que possibilita sua atualização constante, seja para aprendizado de novos padrões ou para desaprendizado de padrões que se tornem desnecessários ao longo do tempo. Além disso, será mostrado que as duas estratégias realizam um controle de complexidade do modelo neural obtido, resultando em soluções com boa capacidade de generalização.

Resultados experimentais mostrarão que as estratégias propostas de Aprendizado Ativo têm desempenho similar ao modelo regularizado das ELMs e às SVMs treinadas com kernel ELM. As estratégias apresentadas, contudo, utilizam apenas uma pequena parte de cada conjunto de dados para obter um classificador com boa capacidade de generalização. Além disso nenhum parâmetro precisa ser finamente ajustado e não é utilizado nenhum processo de otimização computacionalmente caro.

1.1 Objetivos

A seguir são apresentados os objetivos desta tese.

1.1.1 Objetivo Geral

O objetivo geral da tese é apresentar um classificador para problemas binários composto por uma rede neural de duas camadas que é treinada com Aprendizado Ativo. Esse classificador é composto por uma camada escondida ELM e por uma camada de saída treinada com um Perceptron. Esse Perceptron tem seus pesos ajustados de acordo com uma variação da regra de Hebb

e com um critério de seleção de padrões que possibilita o Aprendizado Ativo. O classificador não necessita de ajuste fino de parâmetros livres.

1.1.2 Objetivos Específicos

Esta tese foi elaborada a partir de seis objetivos específicos:

1. Realizar o Aprendizado Ativo em problemas de classificação binária e não linearmente separáveis;
2. Desenvolver métodos de Aprendizado Ativo que não necessitem de ajuste fino de parâmetros livres;
3. Que seja possível o aprendizado incremental e *on-line*;
4. Que os métodos sejam práticos e rápidos;
5. Que sejam capazes de obter uma boa solução em termos de complexidade neural e de capacidade de generalização;
6. Que possuam um critério de seleção de rótulos bem definido ou uma condição de parada adequada.

1.2 Contribuições

Nesta tese são propostas duas estratégias de Aprendizado Ativo que podem ser utilizadas tanto para separadores lineares quanto não lineares. Para o treinamento dos classificadores nenhum parâmetro precisa ser finamente ajustado. O separador linear utilizado é um modelo baseado em um Perceptron com pesos normalizados treinado com aprendizado Hebbiano [21] que, segundo os autores, é capaz de minimizar o erro de treinamento e maximizar a margem, se comportando como uma SVM linear, porém com a característica de ser livre de ajuste de parâmetros e de possuir aprendizado *on-line*.

Ao longo desta tese será demonstrado que o Teorema de Convergência do Perceptron [51, 30] pode ser adaptado para o Perceptron com pesos normalizados treinado com aprendizado Hebbiano e que pode ser utilizado tanto como critério de seleção de padrões, na abordagem *stream-based*, quanto condição de parada para um algoritmo de Aprendizado Ativo na abordagem *pool-based*.

O modelo resultante da combinação de uma camada escondida ELM com o Perceptron citado e treinado com uma das duas estratégias de Aprendizado Ativo resultará em um modelo com aprendizado incremental e *on-line* que permite atualizações constantes. O modelo também possibilita o desaprendizado de padrões que se tornem obsoletos com o tempo. Além disso, as estratégias

de Aprendizado Ativo controlam a complexidade do modelo neural obtido, o que resulta em soluções com boa capacidade de generalização e livres de sobreajuste. Por fim, os métodos de Aprendizado Ativo propostos são práticos e rápidos, com resultados similares aos obtidos por ELMs regularizadas ou por SVMs lineares com *kernel* ELM.

1.3 Organização do Texto

Esta tese está organizada em 8 capítulos da seguinte forma:

- O Capítulo 2 apresenta os conceitos básicos de aprendizado de máquina. Além disso, introduz o conceito de Aprendizados Ativo e apresenta os principais trabalhos que estão relacionados a esta tese;
- O Capítulo 3 apresenta as principais dificuldades encontradas para o desenvolvimento de algoritmos de Aprendizado Ativo para problemas não linearmente separáveis. Discute o uso de SVMs e ELMs nesse tipo de problema.
- O Capítulo 4 apresenta o aprendizado Hebbiano e discute como utilizá-lo para realizar regularização e obter um modelo neural sem sobreajuste;
- O Capítulo 5 apresenta o classificador proposto na tese e as duas estratégias de Aprendizado Ativo desenvolvidas;
- O Capítulo 6 apresenta os resultados para quatro experimentos que pretendem mostrar: as limitações do uso de ELMs para realizar o Aprendizado Ativo; a capacidade de controle de complexidade do modelo neural utilizando-se as estratégias de Aprendizado Ativo; a vantagem dos modelos desenvolvidos não necessitarem de ajuste fino de parâmetros livres; e a aplicação dos métodos desenvolvidos em problemas reais e sua comparação com os resultados obtidos por outros modelos de Aprendizado Ativo apresentados na literatura;
- O Capítulo 7 discute os resultados obtidos e faz especulações em relação à capacidade das estratégias apresentadas de controlarem a complexidade e a capacidade de generalização de modelos neurais. Apresenta ainda as principais questões que permanecem abertas;
- O Capítulo 8 apresenta as conclusões e as propostas de trabalhos futuros.

CAPÍTULO 2

Conceitos Básicos

Neste capítulo serão apresentados os conceitos básicos necessários para o desenvolvimento deste trabalho. Além disso serão apresentados o problema de Aprendizado Ativo e as principais técnicas presentes na literatura.

2.1 Aprendizado de Máquina

O principal objetivo dos algoritmos de Aprendizado de Máquina é fazer com que um computador consiga extrair informações de dados fornecidos e a partir disso consiga desenvolver um modelo geral que seja capaz de representar o problema estudado. Esses dados podem representar diversos tipos de problemas e o algoritmo tenta extrair informações que sejam suficientes para criar um modelo capaz de, dado um estímulo em sua entrada, responder de forma apropriada. Essa resposta pode ser na forma de classificação, regressão, agrupamento ou otimização. Diversos modelos são bioinspirados, como as redes neurais artificiais [7], algoritmos genéticos [65], algoritmo clonal [17], etc.

Todos esses algoritmos têm em comum a capacidade de produzir um modelo matemático capaz de representar algum problema. Isso consiste em induzir uma função $f(\mathbf{x},\mathbf{Z},\mathbf{w}) \approx f_g(\mathbf{x})$, ou seja, obter um modelo matemático que melhor represente alguma função oráculo. A função oráculo é a função que representa a resposta de um problema dado algum estímulo $\mathbf{x}$. Para classificação binária esse modelo pode atuar em duas etapas, sendo a primeira etapa responsável por simplificar o problema e a segunda etapa responsável pela classificação. Essas etapas podem ser modeladas através dos parâmetros $\mathbf{Z}$ e $\mathbf{w}$ que compõem a função $f(\mathbf{x},\mathbf{Z},\mathbf{w})$. O objetivo do Aprendizado é encontrar os

parâmetros $\mathbf{Z}$ e w mais adequados de forma que $f(\mathrm{x},\mathbf{Z},\mathrm{w}) \approx f_g(\mathrm{x})$. Os diversos algoritmos de Aprendizado de Máquina diferem basicamente na forma como os parâmetros $\mathbf{Z}$ e w são utilizados e ajustados. Em geral o parâmetro $\mathbf{Z}$ é utilizado para transformar um problema geral em um problema linearmente separável e o parâmetro w é utilizado para realizar a classificação do problema linearizado.

Os métodos de Aprendizado de Máquina geralmente são classificados nos seguintes grupos: Aprendizado Supervisionado, Aprendizado Não Supervisionado e Aprendizado Semi-supervisionado. Além disso, algoritmos desses grupos podem ainda ser subclassificados em mais duas áreas: Aprendizado Passivo e Aprendizado Ativo. Um algoritmo pode ser do tipo Supervisionado e Passivo ou Supervisionado e Ativo, por exemplo.

No Aprendizado Supervisionado existe a figura do "professor". Ele é responsável por estimular as entradas do modelo e observar a saída para então comparar o resultado obtido com aquele desejado para o estímulo utilizado. A diferença entre a resposta desejada e a resposta obtida é utilizada pelo professor para ajustar os parâmetros do modelo, de forma que quando esse estímulo for novamente apresentado ao modelo a resposta deverá ser a correta [7]. A Figura 2.1 ilustra esse conceito. Nessa abordagem é necessário que estejam disponíveis além dos dados utilizados para estímulo os dados que representem as saídas desejadas.

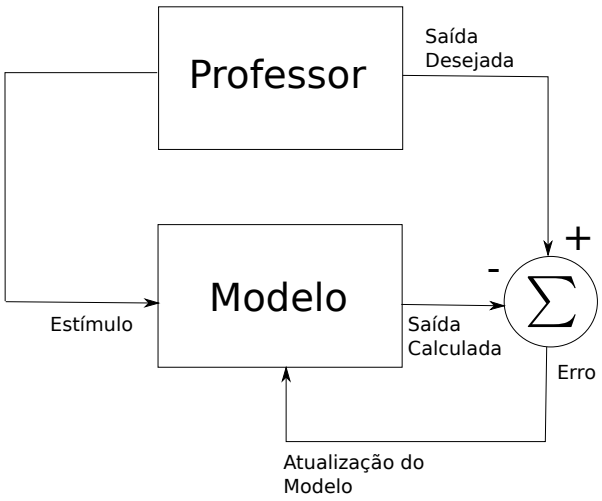


Figura 2.1: Aprendizado Supervisionado. Adaptado de [7]

No Aprendizado Não Supervisionado os dados utilizados para estímulo, ou seja, para treinamento do modelo, podem ou não estar acompanhados dos dados desejados para a saída. Além disso, nessa abordagem não há a figura do "professor" e não é feito nenhum procedimento de correção do erro. Esse conceito é ilustrado na Figura 2.2. Nessa abordagem é extraída informação a partir da estrutura dos dados.

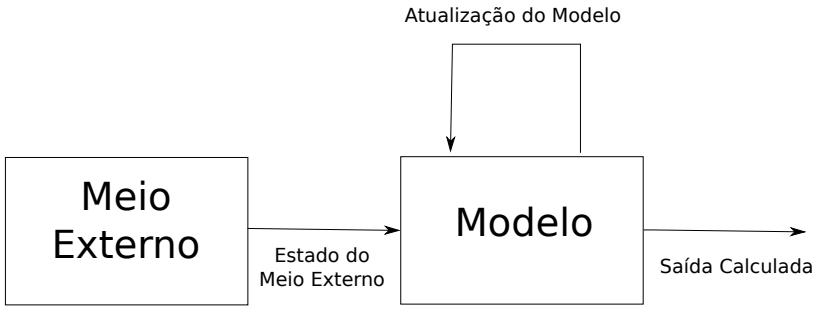


Figura 2.2: Aprendizado Não Supervisionado. Adaptado de [7]

O Aprendizado Semi-Supervisionado utiliza tanto conceitos do Aprendizado Supervisionado quando do Não Supervisionado. A principal característica dos algoritmos desse tipo é que além de utilizar dados rotulados, ou seja, estímulos que possuem uma saída desejada bem definida, também utilizam dados não rotulados durante o processo de aprendizagem. O algoritmo pode, por exemplo, utilizar os dados rotulados para realizar algum tipo de aprendizagem por correção de erro e acrescentar, durante o processo, informações obtidas dos dados não rotulados, como, por exemplo, a distância entre os padrões, a qual agrupamento pertencem, etc.

Esses três tipos de algoritmos podem aprender de forma passiva ou ativa. Na abordagem passiva o algoritmo não tem controle sobre quais dados deve aprender. A probabilidade $P(\mathbf{X})$ de um padrão ser selecionado para aprendizado é igual para todos os padrões. O algoritmo não tem controle e não influencia a probabilidade de amostragem $P(\mathbf{X})$. Nessa abordagem geralmente todos os padrões são rotulados. Esse conceito é ilustrado na Figura 2.3. Isso implica que muitos dados redundantes podem ser rotulados, o que não traz vantagens para o processo de aprendizado. Como geralmente a rotulação dos padrões tem um custo, rotular padrões redundantes pode resultar em desperdício de recursos.

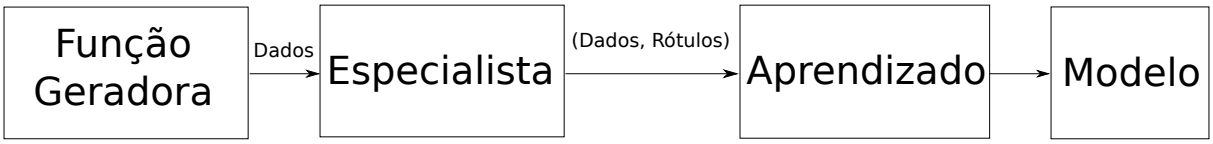


Figura 2.3: Aprendizado Passivo.

Na abordagem ativa o algoritmo tem algum controle sobre a probabilidade de amostragem $P(\mathbf{X})$. Dessa forma, o algoritmo é considerado **ativo** por ser capaz de analisar um padrão e decidir se o mesmo deve ser rotulado e aprendido ou não. A Figura 2.4 ilustra esse conceito. Isso é vantajoso principalmente quando a rotulação dos padrões tem um custo elevado, uma vez que se o algoritmo decidir que um padrão não precisa ser aprendido significa que o mesmo não precisa ser rotulado, o que pode levar a uma economia de recursos. O

Aprendizado Ativo é realizado principalmente em problemas de classificação [62].

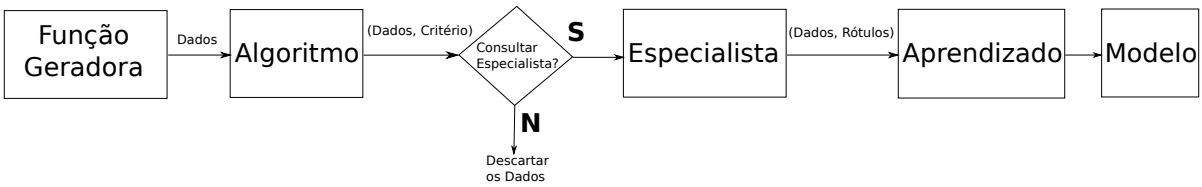


Figura 2.4: Aprendizado Ativo.

O Aprendizado Ativo é o principal tema tratado nesta tese, por isso será detalhado a seguir.

2.2 Aprendizado Ativo

O Aprendizado Ativo é útil quando há grande quantidade de dados não rotulados e o processo de rotulação tem um custo elevado, inviabilizando que todos os padrões sejam rotulados. Esse custo pode ser de natureza econômica, fadiga do especialista, tempo, etc [62]. Escolher para rotulamento padrões de forma aleatória é uma prática comum nos métodos de aprendizado passivo, porém essa abordagem tem a desvantagem de que padrões redundantes têm a mesma probabilidade de serem escolhidos que os padrões mais representativos do problema. O Aprendizado Ativo tem o objetivo de possibilitar que o algoritmo escolha os melhores padrões para serem rotulados por um especialista e utilizados no treinamento de algum modelo de aprendizado de máquina. Dessa forma, padrões redundantes ou pouco representativos não são rotulados, evitando que parte dos recursos disponíveis sejam desperdiçados. Essa característica faz com que o Aprendizado Ativo também seja conhecido como "projeto experimental ótimo" [60], pois busca encontrar o conjunto ótimo de padrões, em termos dos recursos disponíveis, a serem utilizados para o treinamento de modelos de aprendizado de máquina.

Em geral, a forma como os dados são amostrados da função geradora pode ser classificada em pelo menos duas categorias: a "amostragem baseada em fluxo de dados" ou *Stream-based Selective Sampling* [4, 11] e a "amostragem baseada em um conjunto fixo de dados" ou *Pool-based Sampling* [42]. Na primeira forma um padrão é amostrado diretamente da função geradora e o modelo decide se o padrão deverá ser rotulado ou não. Caso o padrão não seja rotulado ele simplesmente é descartado, caso contrário o padrão e seu rótulo são utilizados para atualizar algum modelo de aprendizado de máquina. O processo se repete enquanto existirem recursos disponíveis para a rotulação ou até que algum critério de parada seja alcançado. Na segunda forma uma grande quantidade de padrões é amostrada da função geradora e colocada

em um *pool*. O modelo de Aprendizado Ativo deverá avaliar todos os padrões presentes no *pool* e então decidir, segundo algum critério, qual padrão deverá ser rotulado [60]. O padrão escolhido e seu rótulo são utilizados para atualizar o modelo de aprendizado de máquina com um retreinamento. O processo se repete até que o número máximo de rótulos seja obtido ou até que algum critério de parada seja atingido.

Como observado, os métodos de Aprendizado Ativo necessitam da definição de algum critério para decidir que padrões deverão rotular. Em virtude disso, surge uma questão crucial no Aprendizado Ativo: como selecionar os padrões mais informativos? Muitos autores acreditam que esses padrões são aqueles presentes na região do espaço em que há grande incerteza em relação aos rótulos. Essa região é aquela em que padrões próximos entre si poderão ser de uma classe ou outra, sendo a região com a maior probabilidade de o classificador cometer erros. Dessa forma, diversos métodos de Aprendizado Ativo escolhem e rotulam padrões nessa região, pois eles são, teoricamente, os padrões mais informativos do problema. Em geral, esses métodos são baseados em amostragem por incerteza (*Uncertainty Sampling*) e em busca no espaço de versões.

2.2.1 Amostragem por Incerteza e Busca No Espaço de Versões

Os métodos que realizam amostragem por incerteza têm como principal objetivo selecionar e rotular apenas os padrões para os quais os rótulos são incertos, deixando de lado os padrões que aparentemente terão uma classificação bem definida [60].

Um dos primeiros trabalhos a tentar modelar a região de incerteza foi o trabalho de Cohn *et al.* [11] em que os autores utilizavam duas redes neurais a fim de explorar o espaço de versões [48], sendo uma treinada a fim de ser mais geral e outra treinada a fim de ser mais específica, ou seja, a região do espaço definida por esta deveria estar englobada pela região definida por aquela. Toda vez que um padrão amostrado é classificado com rótulos diferentes pelas duas redes seu rótulo real é solicitado ao especialista, pois isso indica que o padrão está na região de incerteza. De posse do novo rótulo os modelos são retreinados para definir a nova região de incerteza. Um problema dessa abordagem é o custo computacional, pois toda vez que um rótulo é solicitado duas redes neurais devem ser retreinadas.

Essa abordagem também pode ser encarada como um comitê de duas redes neurais, pois utilizam dois modelos treinados separadamente para tomar uma decisão, como nos métodos apresentados a seguir.

Aprendizado Ativo Baseado em Comitês

Uma abordagem muito importante para a área de Aprendizado Ativo é a "*Query by Committe*" [63, 1, 47] em que a região de incerteza é definida explorando-se diretamente o espaço de versões utilizando um comitê de classificadores. Cada classificador gera uma superfície de separação e quando um padrão é amostrado ele é classificado por todos os classificadores do comitê e terá seu rótulo solicitado somente se metade dos classificadores classificá-lo como positivo e a outra metade classificá-lo como negativo. Isso indicaria que o padrão estaria na região de incerteza, pois há um grande desacordo entre os modelos do comitê. Essa abordagem tem a desvantagem de ter um elevado custo computacional, pois a cada vez que um padrão é rotulado todos os classificadores do comitê devem ser retreinados.

Uma abordagem mais simples e prática para explorar a região de incerteza é amostrar os padrões mais próximos de um hiperplano separador, como apresentado a seguir.

Aprendizado Ativo Utilizando SVMs

Uma abordagem prática para amostrar padrões contidos na região de incerteza foi proposta por Schohn e Cohn [58] e por Tong e Koller [70] em que, para problemas linearmente separáveis, um padrão é escolhido de um *pool* para ser rotulado se ele for o mais próximo de um hiperplano separador construído a priori. Quanto mais próximo um padrão estiver do hiperplano separador, maior a chance de uma classificação ser incorreta, o que o torna um forte candidato a ser rotulado e inserido no processo de treinamento. Essa abordagem tem a vantagem de ser simples e de ter custo computacional menor que aquelas apresentadas anteriormente. A maior desvantagem é ter que calcular a distância entre todos os padrões presentes no *pool* e o hiperplano separador em cada iteração do algoritmo. Outra desvantagem é que os modelos propostos são baseados em SVMs o que exige a solução de um problema de programação quadrática cada vez que um padrão é rotulado. O método pode ser aplicado em problemas não linearmente separáveis se um *kernel* for utilizado, porém isso exige que uma quantidade de padrões rotulados seja separado para realizar o ajuste de parâmetros livres.

A heurística proposta por [58] e por [70] pode ser aplicada a outros classificadores lineares, como Perceptrons, por exemplo.

Aprendizado Ativo Utilizando Perceptrons

Outros trabalhos também utilizam a distância de um padrão ao hiperplano separador de forma indireta, utilizando Perceptrons para a construção do hi-

perplano em problemas linearmente separáveis [10, 15, 49]. Essas abordagens têm baixo custo computacional, pois são baseadas em Perceptrons, e também porque avaliam os padrões à medida que eles chegam, decidindo se deverão ser rotulados ou não, em um esquema de Aprendizado Ativo *on-line*. Apesar de parecerem práticos, como relatado por Monteleoni *et al.* [49], eles têm a desvantagem de necessitarem de ajuste de parâmetros livres, o que implica a reserva de uma quantidade de padrões rotulados a fim de se realizar a validação-cruzada. Como apontado por Guillory *et al.* [29] isso eleva o custo do processo, pois a quantidade de rótulos efetivamente utilizada na aprendizagem será a quantidade de rótulos reservados para o ajuste de parâmetros somados à quantidade de rótulos utilizados na fase de Aprendizado Ativo. Dessa forma, a realização do Aprendizado Ativo sem a necessidade de ajuste de parâmetros livres surge como um importante desafio para novos métodos de aprendizado de máquina.

Em geral, as abordagens baseadas em Perceptrons são do tipo *Stream-based Selective Sampling*. Um exemplo é o trabalho de Cesa-Bianchi *et al.* [10] onde é proposta uma modificação no algoritmo clássico do Perceptron para a realização do Aprendizado Ativo sendo definido o limite teórico de rótulos necessários para resolver um problema linearmente separável. A ideia consiste em realizar a solicitação do rótulo de um padrão com uma probabilidade $b/(b + |p|)$, sendo b um parâmetro definido pelo usuário e $|p|$ é a margem do padrão corrente, sendo, portanto, um filtro baseado em margem. Caso o rótulo seja solicitado e a classificação realizada pelo Perceptron seja incorreta, é realizado o ajuste clássico dos pesos do Perceptron, caso contrário o padrão não é utilizado. O valor ótimo para b é $b = (max_{\mathbf{x}\in C}||\mathbf{x}||^2)/2$, fazendo com que o número de erros limite seja igual ao do Perceptron clássico, o que pode fazer com que o algoritmo utilize quase todos os rótulos.

Dasgupta *et al.* [15] realizaram uma análise do Aprendizado Ativo baseado em Perceptron, definindo limites teóricos para o número de erros para o caso particular onde a distribuição dos padrões de entrada é uniforme e o problema é linearmente separável. Eles propuseram uma modificação na forma de ajuste dos pesos do Perceptron, $\mathbf{w}_{t+1} = \mathbf{w}_t - 2(\mathbf{w}_t \cdot \mathbf{x}_t)\mathbf{x}_t$, sendo o ajuste realizado quando o algoritmo solicitar um rótulo e o mesmo for diferente da classificação gerada pelo Perceptron. O algoritmo verifica se o padrão corrente possui margem menor que um limite s_t e, em caso afirmativo, solicita o rótulo do mesmo. Caso o Perceptron acerte a classificação de R padrões consecutivos, o valor da margem limite s_{t+1} é reduzido para a metade $s_{t+1} = s_{t+1}/2$. O parâmetro R deve ser ajustado pelo usuário. O algoritmo é definido para situações onde $||\mathbf{x}|| = 1$ e é definido que $||\mathbf{w}|| = 1$. Os autores definem que o valor inicial de s_t é igual a $1/\sqrt{d}$, onde a d é a dimensão do padrão.

No trabalho de Monteleoni *et al.* [49] é realizada uma comparação empírica dos dois Perceptrons citados anteriormente, realizando pequenas modificações no Perceptron de Dasgupta *et al.* [15] para que o mesmo pudesse funcionar em situações em que os dados não estão distribuídos uniformemente. Eles definiram que o valor máximo de s_t é 1, pois os padrões e o vetor de pesos possuem norma igual a 1. Para o ajuste do parâmetro b do algoritmo de Cesa-Bianchi *et al.* [10] e para o parâmetro R do algoritmo de Dasgupta *et al.* [15] é realizada validação cruzada do tipo *10-fold* em uma parte do conjunto de dados separados e rotulados para esse fim.

Como pode ser observado, os métodos de Aprendizado Ativo baseados em Perceptrons também necessitam de ajuste de parâmetros livres, o que pode tornar o processo caro, pois uma quantidade de rótulos deve ser reservada para este fim. Além disso, esses métodos foram desenvolvidos apenas para trabalharem com problemas linearmente separáveis e com distribuição uniforme dos dados, o que limita a possibilidade de uso dos mesmos. Eles podem ser utilizados com um *kernel*, a fim de linearizar os problemas, mas isso implica que os parâmetros livres do *kernel* também sejam ajustados.

No próximo capítulo serão discutidas as dificuldades para a realização do Aprendizado Ativo em problemas não linearmente separáveis e possíveis caminhos para a solução desse problema.

2.3 Conclusões do Capítulo

Neste capítulo foram apresentados os conceitos básicos necessários para o entendimento do problema de Aprendizado Ativo e os principais trabalhos nessa área. Algumas das estratégias de amostragem por incerteza apresentadas serão utilizadas no Capítulo 6, onde serão comparadas com os métodos desenvolvidos nesta tese. No próximo capítulo será discutido o problema da realização do Aprendizado Ativo utilizando dados não linearmente separáveis.

CAPÍTULO 3

Aprendizado Ativo em Problemas Não linearmente Separáveis

Um mapeamento adequado dos dados de entrada em um espaço de características é um problema central quando se necessita realizar o Aprendizado Ativo em problemas não linearmente separáveis. Esse mapeamento tem o objetivo de encontrar um conjunto de parâmetros que quando aplicado aos dados de entrada consegue fazer com que os dados se tornem linearmente separáveis no espaço projetado. Máquinas de Vetores de Suporte (*Support Vector Machines* - SVM) e redes neurais artificiais de múltiplas camadas têm a capacidade de realizar esse mapeamento e de realizar a classificação. Entretanto, as SVMs precisam de ajuste de parâmetros livres, tanto para a versão linear quanto para a versão não linear que faz uso de um *kernel*, o que pode dificultar o Aprendizado Ativo. Redes neurais treinadas com algoritmos baseados no *Backpropagation* [7] também necessitam do ajuste de parâmetros livres, sendo o número de neurônios escondidos o mais importante. Essas dificuldades serão discutidas nas seções a seguir.

3.1 Máquinas de Vetores de Suporte

Uma Máquina de Vetores de Suporte (*Support Vector Machine - SVM*) consiste em um algoritmo de separação linear que tem como objetivo separar duas classes. Para tanto, o algoritmo gera um hiperplano separador que tem a característica de maximizar a margem de separação entre duas classes [30]. O processo de treinamento consiste em minimizar o risco estrutural por meio da variação da dimensão VC (dimensão Vapnik e Chervonenkis), de modo que

o erro de treinamento e a dimensão VC sejam minimizados simultaneamente [72, 59]. O hiperplano ótimo será aquele capaz de maximizar a margem de separação entre as classes.

Encontrar esse hiperplano consiste em resolver o problema dual ao problema de minimização da dimensão VC. Minimizar a dimensão VC é o mesmo que maximizar a margem de separação [72, 59]. Para um problema de classificação binária onde ($\mathbf{x}_i$,y_i) forma o par entrada-saída para cada padrão de treinamento e $y_i \in \{+1, -1\}$ deve-se encontrar o vetor de pesos ótimo $\mathbf{w}$ resolvendo o seguinte problema de otimização [30]:

$$\text{Minimizar: } \Phi(\mathbf{w},\xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \tag{3.1}$$

Sujeito a:

$$y_i(\mathbf{w}^T\psi(\mathbf{x}_i) + b) \geq 1 - \xi_i \qquad \text{para } i = 1, 2, ..., N$$
$$\xi_i \geq 0 \qquad \text{para todo } i,$$

em que C é um parâmetro de regularização positivo escolhido pelo usuário, ξ_i são variáveis de folga que medem o desvio de um padrão da condição ideal de separabilidade de padrões [30] e $\psi(\mathbf{x}_i)$ consiste em uma função não linear aplicada ao espaço de entrada com o objetivo de projetar o padrão $\mathbf{x}_i$ em um espaço de características onde os padrões se tornem linearmente separáveis. O produto $\psi(\mathbf{x})^T\psi(\mathbf{x}_i)$ induz uma função de *kernel* $K(\mathbf{x},\mathbf{x}_i)$. Essa função possibilita construir o hiperplano ótimo no espaço de características sem ter que considerar o próprio espaço de características de forma explícita [30].

Para solucionar o problema de otimização é utilizado o método dos multiplicadores de Lagrange, formulando-se o problema dual apresentado na Equação 3.2

$$\text{Maximizar: } Q(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i,\mathbf{x}_j) \tag{3.2}$$

$$\text{Sujeito a: } \sum_{i=1}^{N} y_i\alpha_i = 0;$$
$$\forall_{i=1}^{N} : 0 \leq \alpha_i \leq C,$$

em que α_i é o multiplicador de Lagrange, $K(\mathbf{x}_i,\mathbf{x}_j)$ é o *kernel* utilizado. O parâmetro C controla a relação entre a complexidade do algoritmo e o número de padrões de treinamento classificados incorretamente. Ele pode ser visto como um parâmetro de penalização [59].

O vetor de pesos ótimo $\mathbf{w}^*$ é dado por:

$$\mathbf{w}^* = \sum_{i=1}^{N} y_i \alpha_i^* \psi(\mathbf{x}_i) \tag{3.3}$$

e o intercepto b^* é dado por:

$$b^* = -\frac{1}{2}\left[\max_{\{i|y_i=-1\}}\left(\sum_{j=1}^{N_{SV}} y_j \alpha_j K(\mathbf{x}_i,\mathbf{x}_j)\right) + \min_{\{i|y_i=+1\}}\left(\sum_{j=1}^{N_{SV}} y_j \alpha_j K(\mathbf{x}_i,\mathbf{x}_j)\right)\right]. \tag{3.4}$$

A função de decisão é dada por [59]:

$$f(\mathbf{x}) = \hat{y} = \mathbf{sign}\left(\sum_{i=1}^{N_{SV}} y_i \alpha_i^* K(\mathbf{x}_i,\mathbf{x}) + b^*\right) \tag{3.5}$$

ou seja, se $f(\mathbf{x}) \geq 0$, então x pertence à classe +1, senão x pertence à classe -1. Nessa equação N_{SV} é o número de vetores de suporte. Um vetor de suporte é um padrão para o qual o multiplicador de Lagrange α_i^* associado é diferente de zero.

Para solucionar o problema de otimização apresentado na Equação 3.2 utiliza-se programação quadrática. A solução Lagrangeana do problema de programação quadrática das SVMs consiste em encontrar multiplicadores de Lagrange que, na prática, indicam os padrões (vetores de suporte) que são usados para calcular a saída da SVM. A saída $\hat{y}_j$ da SVM para um padrão de entrada $\mathbf{x}_j$ é uma combinação linear de rótulos y_j ponderados pelo *kernel* $K(\mathbf{x}_i,\mathbf{x}_j)$ entre $\mathbf{x}_j$ e todos os outros padrões de treinamento $\mathbf{x}_i$, conforme apresentado na Equação 3.5.

Como somente os padrões com multiplicadores de Lagrange diferentes de zero efetivamente contribuem para o cálculo de $\hat{y}_j$, o treinamento de uma SVM pode ser visto como um problema de seleção de padrões. Dado um *kernel* com parâmetros apropriados, a seleção de padrões marginais $\mathbf{x}_i$ e de multiplicadores de Lagrange α_i levam à minimização do erro e a maximização da margem [71].

Nesse cenário, padrões "descartados" são aqueles multiplicados por multiplicadores de Lagrange nulos $\alpha_i = 0$ e os padrões "selecionados", os vetores de suporte, são aqueles para as quais os multiplicadores de Lagrange estão na faixa de $0 < \alpha_i \leq C$.

Uma dificuldade em realizar o Aprendizado Ativo utilizando SVMs é que todo o conjunto de treinamento deve ser utilizado para retreinar a SVM à medida que novos padrões são selecionados. Isso implica que para cada novo padrão selecionado um problema de programação quadrática deve ser resolvido. Métodos de aprendizado ativo devem ser capazes de lidar com aprendizado

incremental e *on-line* [61, 49, 29].

Além disso, as SVMs necessitam que o parâmetro C seja ajustado o que exige que padrões sejam reservados para esse fim, o que pode aumentar o custo do processo de aprendizado em um cenário onde os rótulos dos padrões têm custo de aquisição elevado. A utilização de funções de *kernel* dificultam o processo, uma vez que também exigem ajuste de parâmetros.

Um modelo de aprendizado de máquina que se mostra promissor para a realização do Aprendizado Ativo são as Máquinas de Aprendizado Extremo (*Extreme Learning Machines - ELM*). Esse modelo não exige ajuste fino de parâmetros livres e utiliza um algoritmo de aprendizagem relativamente simples.

Tanto as ELMs quanto as SVMs são baseadas em um mapeamento de duas camadas, sendo a primeira para linearização do problema e a segunda para realizar a separação linear. O *kernel* das SVMs realiza um mapeamento implícito enquanto as ELMs são baseadas em um mapeamento explícito usando uma camada escondida com função sigmoidal [44, 23]. As ELMs serão detalhadas a seguir, e serão discutidas as dificuldades do seu uso para a realização do Aprendizado Ativo.

3.2 Máquinas de Aprendizado Extremo

Uma das principais dificuldades em se treinar redes neurais artificiais de duas camadas do tipo *Multilayer Perceptron (MLP)* é o cálculo dos pesos da camada escondida. Em geral, abordagens tradicionais realizam a retropropagação do erro de treinamento através das camadas da rede neural artificial. O algoritmo mais conhecido para realizar esse tipo de treinamento é o *Back-propagation* que serve como base para diversos outros algoritmos apresentados na literatura [7]. Entretanto, é sabido desde a década de 1990 que os parâmetros da camada escondida (pesos e *bias*) podem ser definidos aleatoriamente, bastando realizar o treinamento dos parâmetros (pesos) da camada de saída [57]. Essa forma de treinamento se tornou popular somente após o ano de 2006 com o trabalho de Huang et al. [39] que apresentou as Máquinas de Aprendizado Extremo (*Extreme Learning Machine - ELM*).

As Máquinas de Aprendizado Extremo consistem basicamente em um algoritmo para treinamento de redes neurais artificiais de duas camadas. A Figura 3.1 apresenta a topologia típica das ELMs. As principais características do algoritmo são:

1. O número de neurônios escondidos é grande;

2. O treinamento dos pesos da camada escondida e da camada de saída é feito separadamente;

3. Os pesos da camada escondida são ajustados aleatoriamente;

4. Os pesos da camada de saída não são ajustados iterativamente, mas obtidos diretamente usando o método da pseudoinversa.

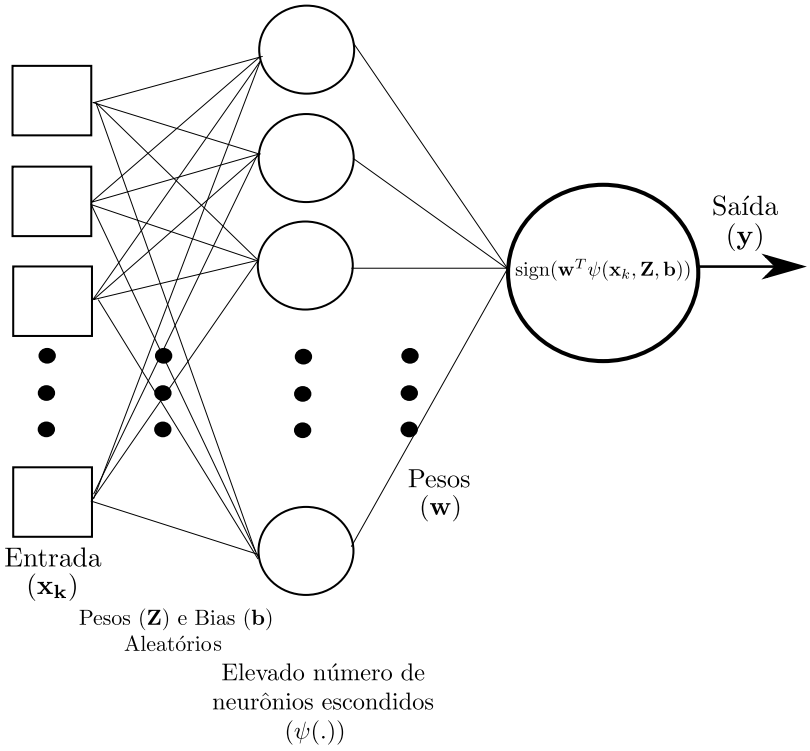


Figura 3.1: Topologia típica das ELMs

A matriz de entrada $\mathbf{X}$ com N linhas e n colunas contém os dados de treinamento, em que N é o número de padrões e n é a dimensão do espaço de entrada. As linhas do vetor $\mathbf{y}$ de dimensão $N \times 1$ contém os rótulos correspondentes para cada um dos N padrões de entrada de $\mathbf{X}$, como visto a seguir:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \vdots & \vdots \\ x_{N1} & \cdots & x_{Nn} \end{bmatrix}_{N \times n}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}. \tag{3.6}$$

A função $\psi(\mathbf{x},\mathbf{Z},\mathbf{b})$, com argumento $\mathbf{x}$, matriz de parâmetros $\mathbf{Z}$ e vetor de *bias* $\mathbf{b}$, mapeia cada uma das linhas de $\mathbf{X}$ nas linhas da matriz de mapeamento $\mathbf{H}$ de dimensão $N \times p$, como visto a seguir:

$$\mathbf{H} = \begin{bmatrix} \psi(\mathbf{x}_1^T,\mathbf{Z},\mathbf{b}) \\ \vdots \\ \psi(\mathbf{x}_N^T,\mathbf{Z},\mathbf{b}) \end{bmatrix} = \begin{bmatrix} \text{sigmoid}(\mathbf{x}_1^T \cdot \mathbf{z}_1 + b_1) & \cdots & \text{sigmoid}(\mathbf{x}_1^T \cdot \mathbf{z}_p + b_p) \\ \vdots & \vdots & \vdots \\ \text{sigmoid}(\mathbf{x}_N^T \cdot \mathbf{z}_1 + b_1) & \cdots & \text{sigmoid}(\mathbf{x}_N^T \cdot \mathbf{z}_p + b_p) \end{bmatrix}_{N \times p}. \tag{3.7}$$

Nessa equação p é o número de neurônios escondidos ($\mathbf{H} = \psi(\mathbf{x},\mathbf{Z},\mathbf{b})$). A função de ativação de todos os neurônios corresponde à função tangente sigmoidal.

No caso particular das ELMs, como os elementos de $\mathbf{Z}$ e $\mathbf{b}$ são escolhidos aleatoriamente, o número de neurônios escondidos p deve ser grande o bastante para satisfazer o Teorema de Cover [14], ou seja, quanto maior for a dimensão do espaço utilizado para representar um problema de classificação, maior será a probabilidade do problema se tornar linearmente separável. Dessa forma, os dados projetados de $\mathbf{X}$ em $\mathbf{H}$ serão considerados linearmente separáveis nesse novo espaço.

A matriz $\mathbf{H}$ é mapeada no espaço de saída pela função $\phi(\mathbf{H},\mathbf{w})$ para aproximar o vetor de rótulos $\mathbf{y}$. O vetor $\mathbf{w}$ de dimensão $p \times 1$ contém os parâmetros do separador linear na camada escondida e é obtido pela solução de um sistema de equações lineares de N equações:

$$\mathbf{H}\mathbf{w} = \mathbf{y}. \tag{3.8}$$

A solução de norma mínima desse sistema de equações é:

$$\mathbf{w} = \mathbf{H}^{\dagger}\mathbf{y} \tag{3.9}$$

em que $\mathbf{H}^{\dagger}$ é a *pseudoinversa de Moore-Penrose*. A resposta da rede $\hat{\mathbf{y}}$ a um padrão de entrada $\mathbf{x}$ é obtida calculando-se $\mathbf{H}$ e estimando a saída como $\hat{\mathbf{y}} = \text{sign}(\mathbf{H}\mathbf{w})$.

Considerando que os valores presentes na matriz $\mathbf{H}$ são reais e que $(\mathbf{H}^T\mathbf{H})^{-1}$ existe, a *pseudoinversa de Moore-Penrose* pode ser calculada da seguinte forma [3]:

$$\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T. \tag{3.10}$$

Como pode ser observado na Equação 3.10, caso a pseudoinversa possa ser calculada ela existirá mesmo se o número de padrões for menor que o número de neurônios. O vetor $\mathbf{w}$ calculado pelo método da pseudoinversa é a solução de norma mínima para um sistema de equações lineares.

Como em qualquer problema de aproximação, espera-se que a função geral resultante $f(\mathbf{x},\mathbf{Z},\mathbf{b},\mathbf{w})$ seja robusta, ou generalize, para $\mathbf{x}_i \notin \mathbf{X}$. Entretanto, a solução de erro zero quando $N = p$ resulta em sobreajuste. Como o conjunto de treinamento é formado incrementalmente no Aprendizado Ativo, o valor de N eventualmente se igualará ao valor de p durante o processo de aprendizado. Isso tornaria impraticável o uso de ELM nesse contexto, mesmo utilizando as heurísticas de Schohn e Cohn [58] e Tong e Koller [70], discutidas no Capítulo 2.

Com o objetivo de mostrar a degradação no desempenho de uma ELM

treinada pelo método da pseudoinversa, uma estratégia de seleção aleatória de padrões foi aplicada ao conjunto de dados da Figura 3.2, utilizando 100 neurônios na camada escondida. A Figura 3.2 apresenta um problema de classificação binária não linearmente separável com 180 padrões para cada classe. O experimento consiste em dez execuções de uma validação cruzada do tipo *10-fold*. Esse tipo de validação cruzada consiste em dividir aleatoriamente o conjunto de treinamento em dez partes e utilizar nove partes para o treinamento e a parte restante para validação. Na etapa seguinte a parte utilizada para validação é introduzida no conjunto de treinamento e outra parte é removida do conjunto de treinamento e utilizada para validação. O processo continua até que todas as dez partes tenham sido utilizadas na etapa de validação. A resposta final de cada execução da validação cruzada foi o valor médio da área abaixo da curva ROC (*Area Under the ROC Curve* - AUC) [2] calculada sobre a parte do conjunto utilizada para validação.

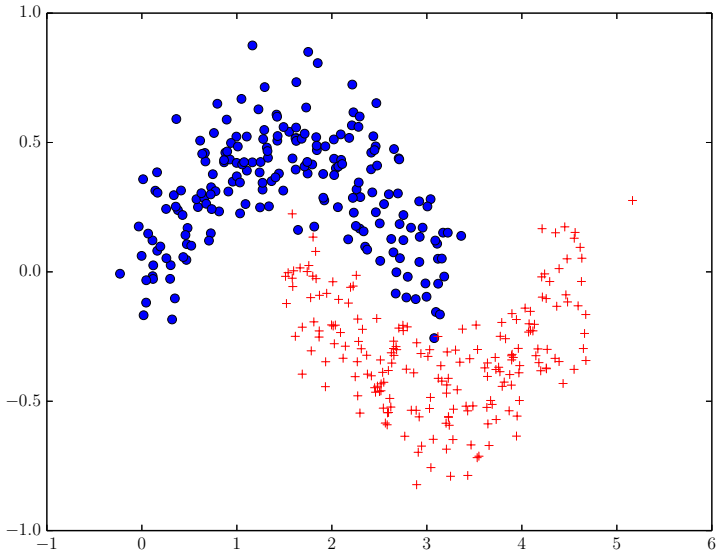


Figura 3.2: Problema de classificação binária utilizado no exemplo de seleção aleatória da Figura 3.3.

O processo de aprendizado é iniciado utilizando somente um padrão escolhido aleatoriamente. O padrão é propagado através da camada escondida que é composta por pesos aleatórios. Os pesos de saída são obtidos pelo cálculo da pseudoinversa. O aprendizado utiliza uma estratégia de seleção aleatória de padrões e à medida que novos padrões são inseridos no processo de aprendizado, a propagação do padrão e o cálculo da pseudoinversa são repetidos. A Figura 3.3 mostra a AUC média dos experimentos. Como pode ser observado, o desempenho da AUC é fortemente degradado na região $N \approx p$ quando o número de equações se aproxima do número de variáveis. Nessa situação

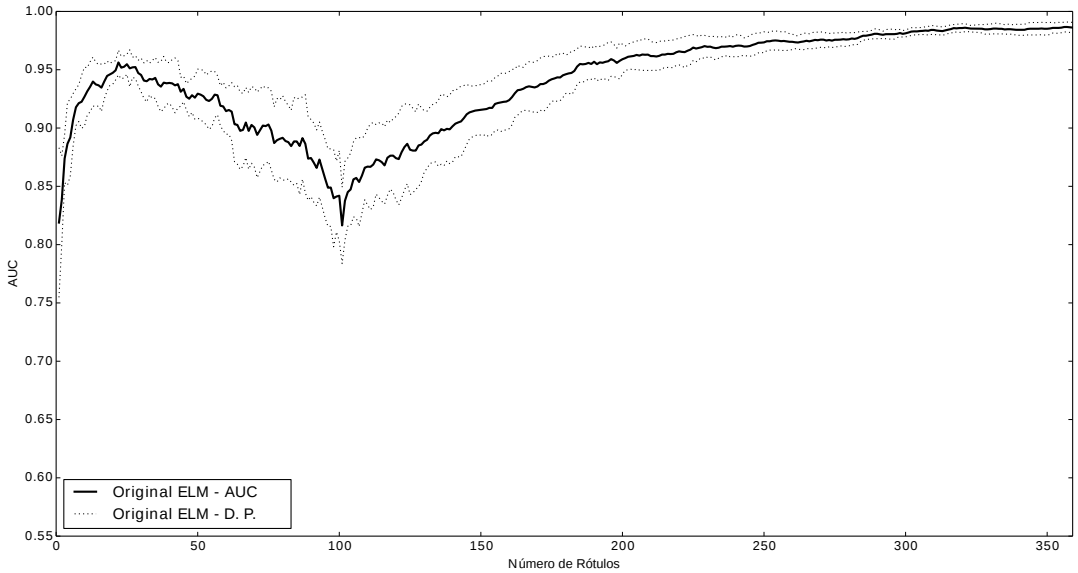


Figura 3.3: O treinamento das ELMs baseado na solução de um sistema de equações lineares pelo método da pseudoinversa causa um declínio abrupto na performance do classificador quando o número de padrões selecionados tende ao número de neurônios escondidos ($N \approx p$).

a solução do sistema equações lineares através da pseudoinversa tem erro mínimo.

O Aprendizado Ativo pode se beneficiar da linearização fornecida pela projeção na camada escondida das ELMs, uma vez que esse modelo de linearização é praticamente um mapeamento que não necessita de ajuste fino de parâmetros. O número de neurônios escondidos p não precisa ser finamente ajustado e os pesos da camada escondida são definidos aleatoriamente [23, 40]. Dessa forma nenhuma otimização de parâmetros é necessária e nenhuma iteração com o usuário é realizada durante o aprendizado. Em compensação, o elevado número de neurônios escondidos e a solução da pseudoinversa resultarão no comportamento indesejado da Figura 3.3.

Uma variação *online* e sequencial das ELMs denominada *Online Sequential Extreme Learning Machine* (OS-ELM) [43] pode parecer um bom candidato para o Aprendizado Ativo, uma vez que esse modelo pode aprender dados apresentados um a um. Entretanto, sua formulação exige que o modelo inicial seja calculado utilizando pelo menos $N = p$ padrões. Como para se obter uma boa separação linear p deve ser grande, isso implica que o conjunto de treinamento inicial também deve ser grande. Dessa forma, esse modelo não é a melhor escolha, uma vez que o principal objetivo do Aprendizado Ativo é minimizar o número de padrões rotulados necessários para o treinamento [61].

Um modelo de ELM capaz de evitar o comportamento indesejado da Fi-

gura 3.3 é a nova formulação regularizada proposta por Huang et al. [40]. Essa nova formulação permite que as ELMs generalizem bem mesmo se $N \leq p$. O problema de classificação binária pode ser definido da seguinte forma [40]:

$$\begin{aligned} &\text{Minimizar: } L_{Primal} = \frac{1}{2}||\mathbf{w}||^2 + C\frac{1}{2}\sum_{i=1}^{N}\xi_i^2 \\ &\text{Subjeito a: } \psi(\mathbf{x}_i^T,\mathbf{Z},\mathbf{b}) \cdot \mathbf{w} = y_i - \xi_i, \qquad i = 1,...,N \end{aligned} \tag{3.11}$$

em que ξ_i é o erro de treinamento referente ao padrão de treinamento $\mathbf{x}_i$ e C é um parâmetro de regularização.

Nessa formulação, o treinamento da ELM é equivalente a resolver o seguinte problema de otimização dual [40]:

$$L_{Dual} = \frac{1}{2}||\mathbf{w}||^2 + C\frac{1}{2}\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N}\alpha_i(\psi(\mathbf{x}_i^T,\mathbf{Z},\mathbf{b}) \cdot \mathbf{w} - y_i + \xi_i) \tag{3.12}$$

em que cada multiplicador de Lagrange α_i corresponde a um padrão de treinamento [40].

Huang et al. [40] propõem duas soluções para esse problema de otimização:

1. Para o caso em que o conjunto de treinamento tem tamanho pequeno ou médio:
$$\mathbf{w} = \mathbf{H}^T(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{y}. \tag{3.13}$$
2. Para o caso em que o conjunto de treinamento tem tamanho elevado:
$$\mathbf{w} = (\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}. \tag{3.14}$$

Para classificação binária a saída para um padrão de entrada $\mathbf{x}_i$ é estimada como sendo $\hat{\mathbf{y}} = \text{sign}(\psi(\mathbf{x}_i,\mathbf{Z},\mathbf{b})\mathbf{w})$ [40]. Para essa formulação um parâmetro de regularização deve ser finamente ajustado. Nesse caso, alguns padrões devem ser separados e rotulados para essa finalidade, o que pode aumentar os custos do Aprendizado Ativo. O principal objetivo do Aprendizado Ativo é induzir um modelo de aprendizado utilizando o menor número possível de padrões rotulados, o que torna proibitivo o ajuste fino de parâmetros.

Com o objetivo de mostrar as características dessas formulações alternativas das ELMs [1] o mesmo problema da Figura 3.3 foi aplicado aos métodos OS-ELM e a formulação regularizada das ELMs, que aqui foi chamada de ELM2012. Para ELM2012 trinta porcento do conjunto de treinamento foi uti-

[1]Todas as implementações das ELMs foram baseadas nos códigos fonte disponíveis no *website* http://www.ntu.edu.sg/home/egbhuang/elm_codes.html.

lizado para realizar o ajuste fino do parâmetro de regularização C. Para OS-ELM foi utilizado na fase de inicialização um número de padrões rotulados igual ao número de neurônios escondidos (100). Os padrões subsequentes foram aprendidos um a um sendo amostrados aleatoriamente. Os resultados estão apresentados na Figura 3.4. Como pode ser observado, OS-ELM sofre com a solução sobreajustada da pseudoinversa na fase de inicialização ($N = p$). Esse problema é propagado durante a fase de aprendizado um a um. O método ELM2012 resolve as limitações da pseudoinversa quando $N \approx p$, mas utiliza um parâmetro que deve ser finamente ajustado, o que pode ser proibitivo para o Aprendizado Ativo [29].

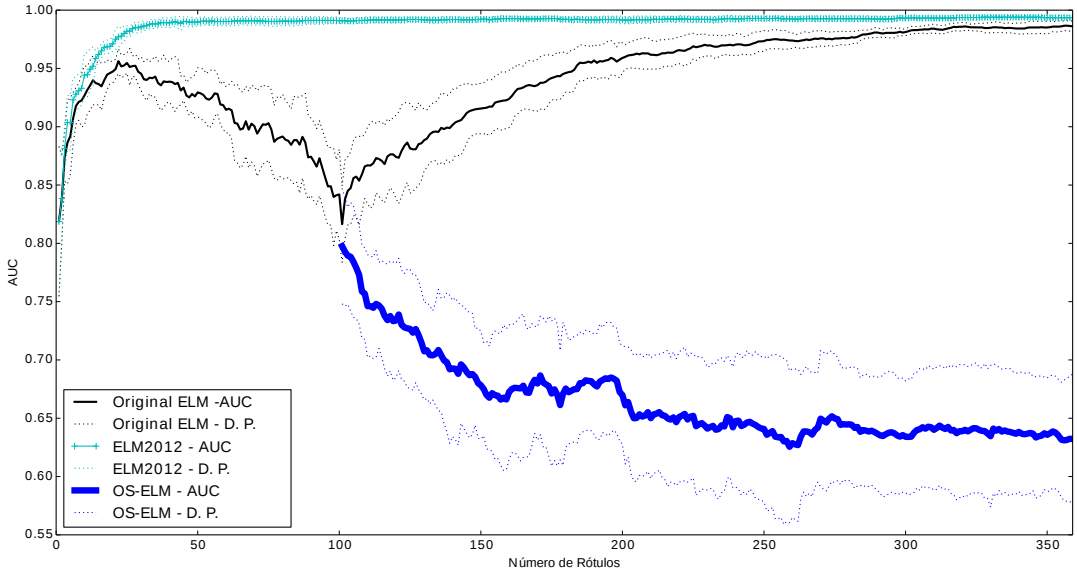


Figura 3.4: Comparação entre a formulação original das ELMs [39], OS-ELM [43] e a formulação regularizada [40] aqui chamada ELM2012. Trinta porcento do conjunto de treinamento foi utilizado para realizar o ajuste fino do parâmetro de regularização do modelo ELM2012

3.2.1 Aprendizado Ativo Utilizando Máquinas de Aprendizado Extremo

Na literatura já existem alguns métodos recentes de Aprendizado Ativo que fazem uso das ELMs [56, 74], entretanto, todos eles exigem ajuste fino de parâmetros, o que, do ponto de vista prático, limita a aplicação do Aprendizado Ativo por exigir que uma quantidade de dados rotulados esteja disponível para esse fim [29]. O método proposto por Samat et al. [56] denominado AELM (*Active Extreme Learning Machine*) consiste no uso de um comitê formado por diversas redes treinadas com a formulação original das ELMs. O número de

neurônios escondidos de cada modelo é definido utilizando validação cruzada, o que exige a reserva de uma quantidade de padrões rotulados para essa finalidade. O autor não utilizou a principal característica das ELMs de possibilitar uma boa separabilidade linear quando o número de neurônios é muito maior que a dimensão do espaço de entrada. Isso é devido ao fato de o conjunto inicial não ser muito grande, pois no aprendizado ativo o número de rótulos é limitado. Nesse caso o autor não pode utilizar um número elevado de neurônios escondidos, pois a capacidade de generalização pode ser prejudicada devido à solução do sistema de equações lineares utilizando o método da pseudoinversa, conforme discutido anteriormente. A seleção dos padrões mais informativos é feita baseada em uma estratégia de voto majoritário. Essa abordagem têm diversas limitações, a saber: o número de neurônios escondidos de cada rede do comitê deve ser finamente ajustado utilizando, por exemplo, o método da validação cruzada; é necessário o uso de um número elevado de redes no comitê; o método sofre de um problema crônico em estratégias do tipo *pool-based*: não possui critério de parada. Todas essas limitações reduzem a aplicabilidade desse modelo em problemas reais de Aprendizado Ativo.

Outra abordagem recente foi proposta por Yu et al. [74] denominada AL-ELM (*Active Learning ELM*) na qual os autores fazem uso do modelo regularizado das ELMs citado anteriormente. A estratégia utilizada pelos autores consiste em estimar a probabilidade a posteriori de um padrão pertencer a uma classe ou outra baseada nas saídas lineares da ELM regularizada. Nessa abordagem cada classe é representada por um neurônio de saída, sendo que para um problema de classificação binária são utilizados dois neurônios na saída. A classificação de um padrão corresponde ao neurônio com maior valor de saída. Essa abordagem também é do tipo *pool-based*, sofrendo do mesmo problema crônico: não possui um critério de parada. Outra limitação consiste na necessidade de se ajustar o parâmetro C do modelo regularizado das ELMs, o que exige que uma quantidade de padrões rotulados seja separado para esse fim. Esse ajuste pode ser realizado utilizando, por exemplo, o método da validação cruzada. Além disso, os autores também não fizeram uso das vantagens referentes a uma camada escondida de dimensão elevada, necessitando ajustar também o número de neurônios escondidos. Todas essas limitações tornam pouco prático o seu uso em problemas reais de Aprendizado Ativo.

Nesta tese serão propostos métodos de Aprendizado Ativo baseados em ELMs que serão capazes de solucionar todas as limitações dos métodos citados nesta seção. No próximo capítulo será apresentada uma abordagem baseada em aprendizado Hebbiano [31, 21] para compensar o comportamento indesejado apresentado na Figura 3.3 e para escapar do ajuste fino de parâ-

metros de regularização.

3.3 Conclusões do Capítulo

Neste capítulo foram discutidas as dificuldades de se realizar o Aprendizado Ativo em problemas não linearmente separáveis. Verificou-se que as SVMs parecem ser adequadas ao Aprendizado Ativo, mas o ajuste de parâmetros livres e o retreinamento utilizando todo o conjunto de treinamento tornam o seu uso pouco prático e pouco eficiente. Apresentou-se como alternativa o uso de ELMs, já que esse modelo funciona praticamente como um mapeamento não linear sem ajuste fino de parâmetros. Discutiu-se a dificuldade em utilizar o treinamento de ELMs através do método da pseudoinversa, uma vez que essa tende a levar ao sobreajuste quando o número de padrões tende ao número de neurônios escondidos. No próximo capítulo será discutido como utilizar o aprendizado Hebbiano para evitar o sobreajuste nas ELMs e permitir o treinamento *on-line*.

CAPÍTULO 4

Aprendizado Hebbiano e Regularização

O aprendizado Hebbiano é uma forma de aprendizado não supervisionado cujo resultado é proporcional ao produto cruzado dos padrões pelos seus rótulos. Essa forma de aprendizado não se baseia em correção do erro, não exigindo, portanto, a comparação do rótulo desejado pelo obtido. Um termo residual devido a não ortogonalidade dos dados estará presente na maioria das aplicações. Neste capítulo será mostrado que esse termo residual, que tradicionalmente é considerado uma limitação do aprendizado Hebbiano, pode ser utilizado para realizar um tipo de regularização não supervisionada. Esse termo será crucial para escapar da solução sobreajustada das ELMs. O controle desse termo residual será um dos objetivos dos métodos de Aprendizado Ativo que serão apresentados no próximo capítulo.

4.1 Aprendizado Hebbiano

No aprendizado de redes neurais artificiais, a atualização dos pesos w_{ij} que conectam neurônios i e j de acordo com a regra de Hebb [31] é proporcional ao produto cruzado de seus valores de ativação para cada associação entrada-saída k ou, em outras palavras, $w_{ij} \propto x_{ik}y_{jk}$. A regra pode ser escrita na seguinte forma matricial:

$$\mathbf{w} = \mathbf{X}^T\mathbf{y}, \tag{4.1}$$

em que $\mathbf{w}$ é o vetor de pesos $n \times 1$, $\mathbf{X}$ é a matriz $N \times n$ de dados de entrada e $\mathbf{y}$ é o vetor $N \times 1$ que contém os valores de saída.

A estimação $\hat{\mathbf{y}}$ do vetor $\mathbf{y}$ é dada por $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$. Substituindo-se $\mathbf{w}$ pela Equação 4.1 nessa expressão tem-se que $\hat{\mathbf{y}} = \mathbf{X}\mathbf{X}^T\mathbf{y}$. Para recuperar perfeitamente $\mathbf{y}$ deve-se ter $\mathbf{X}\mathbf{X}^T = \mathbf{I}$. Se essa condição é obtida o erro de treinamento é zero, já que $\hat{\mathbf{y}} = \mathbf{y}$, e a regra de aprendizado seria equivalente ao método da pseudoinversa apresentado na Equação 3.9. Essa condição ocorre somente se $\mathbf{x}_i \perp \mathbf{x}_j$ e $\mathbf{x}_i^T\mathbf{x}_i = 1 \ \forall ij$. Nessa situação $\mathbf{X}$ forma uma base ortonormal e $\mathbf{X}^T = \mathbf{X}^{-1}$ [3] o que leva às Equações 4.1 e 3.9 serem equivalentes. Entretanto, na maioria das situações reais $\mathbf{X}$ não será uma base ortonormal e um termo residual devido ao produto $\mathbf{X}\mathbf{X}^T$ causará um deslocamento de $\hat{\mathbf{y}}$ em relação a $\mathbf{y}$.

Como $\hat{y}_k = \sum_{j=1}^{N} \mathbf{x}_k^T\mathbf{x}_j y_j$, o termo correspondente a $j = k$ pode ser separado da soma o que leva à Equação 4.2. Na situação particular em que os dados de entrada são normalizados, ou seja $\mathbf{x}_k^T\mathbf{x}_k = 1$, a Equação 4.2 pode ser simplificada na Equação 4.3 [32]:

$$\hat{y}_k = \mathbf{x}_k^T\mathbf{x}_k y_k + \sum_{j=1, j\neq k}^{N} \mathbf{x}_k^T\mathbf{x}_j y_j \tag{4.2}$$

$$\hat{y}_k = y_k + \overbrace{\sum_{j=1, j\neq k}^{N} \mathbf{x}_k^T\mathbf{x}_j y_j}^{crosstalk} . \tag{4.3}$$

A interpretação da Equação 4.3 é que a resposta estimada $\hat{y}_k$ para o padrão de entrada $\mathbf{x}_k$ é a resposta exata y_k mais o termo do *crosstalk* residual.

O *crosstalk* da Equação 4.3 é inerente ao aprendizado Hebbiano e é devido à não ortogonalidade dos dados de entrada, dependendo de como os padrões estão espacialmente relacionados uns aos outros. Nas abordagens usuais de aprendizado indutivo os dados são fornecidos para aprendizado sem nenhum controle na probabilidade de amostragem de um dado $\mathbf{x}_k$. Portanto, o *crosstalk* é inerente a um dado conjunto de dados e pode ser calculado diretamente dos padrões.

Da Equação 4.3 pode ser observado que o *crosstalk* representa um deslocamento em relação à solução de erro zero da pseudoinversa, que é indesejada no aprendizado ELM já que leva ao sobreajuste. O grau no qual a solução Hebbiana difere da solução de erro zero pode ser controlado por uma estratégia de aprendizado ativo, escolhendo apenas os padrões mais adequados para serem aprendidos. Isso tem um efeito de penalização em relação ao resultado da pseudoinversa, o que se espera que resulte em uma curva de aprendizado mais suave que aquela apresentada na Figura 3.3.

Nesta tese propõe-se que o uso do aprendizado Hebbiano apresentado na Equação 4.1 no lugar da Equação 3.9 leva a uma melhor generalização devido

ao termo residual (*crosstalk*), o que evita os efeitos do *overfitting* resultantes do cálculo da pseudoinversa, como será discutido na próxima seção. Além disso, o aprendizado Hebbiano é particularmente adequado ao aprendizado ativo, pois ao selecionar um novo padrão não é necessário que os padrões previamente selecionados sejam reapresentados ao processo de treinamento, sendo, portanto, um processo de aprendizado *on-line*. A contribuição de um novo padrão é simplesmente somada ao vetor de pesos corrente, como mostrado a seguir:

$$\mathbf{w}_{hebb} = \overbrace{\mathbf{x}_0 y_0 + \mathbf{x}_1 y_1 + \cdots + \mathbf{x}_t y_t}^{\text{Vetor de pesos corrente}} + \overbrace{\mathbf{x}_{t+1} y_{t+1}}^{\text{Novo padrão}} . \tag{4.4}$$

O aprendizado Hebbiano é do tipo não supervisionado [7], não sendo necessário controlar de forma iterativa a minimização do erro. Esse controle está implícito na escolha dos padrões que se deseja aprender. Ao selecionar para aprendizado apenas os padrões mais informativos o erro será automaticamente minimizado. Dessa forma fica evidente que utilizar uma técnica de aprendizado ativo adequada potencializará a capacidade de classificação de um Perceptron treinado com a regra de Hebb. Além disso, o desaprendizado, que é apropriado para remover padrões redundantes em problemas dinâmicos onde um padrão pode se tornar obsoleto com o tempo, pode também ser realizado no aprendizado Hebbiano, bastando remover os dados indesejados do somatório. Essa abordagem de desaprendizado será objeto de estudo em trabalhos futuros.

O *crosstalk* pode aumentar à medida que novos padrões são selecionados para serem aprendidos e inseridos no somatório da Equação 4.4. A estimação do termo de *crosstalk* em diferentes cenários foi tema de muitos trabalhos que tinham como objetivo a estimação da capacidade de armazenamento de Redes de Hopfield [33] nos anos 1980 [45, 53, 54, 27]. O número limite de associações que podem ser armazenadas é claramente dependente do *crosstalk*. No aprendizado ativo, entretanto, a magnitude do *crosstalk* é resultado da seleção dos padrões para o treinamento, sendo que dessa forma a estratégia de seleção realiza um controle indireto de quanto $\hat{y}_k$ desviará de y_k. No Capítulo 5 uma estratégia de seleção e um método para estimar o número máximo de padrões necessários para o treinamento de ELMs com aprendizado Hebbiano serão apresentadas para o contexto do Aprendizado Ativo. A seguir será mostrado que o termo de *crosstalk* tem um efeito de regularização no aprendizado Hebbiano.

4.1.1 Evitando Overfitting com Aprendizado Hebbiano

Como discutido no Capítulo 3, o treinamento de ELMs tende a resultar em redes sobreajustadas devido ao cálculo da pseudoinversa quando o número de padrões é muito próximo ou igual ao número de neurônios escondidos [39]. Isso acontece devido à solução da pseudoinversa ser ótima e resultar em erro zero quando $N = p$. Para evitar os efeitos do *overfitting* resultante da minimização do erro de treinamento, alguns métodos realizam um deslocamento da solução da rede da região de erro zero. Métodos de regularização [28], por exemplo, incluem um termo de penalização na função objetivo, o que é usualmente representado como uma combinação linear do erro quadrático e de uma função de penalização adicional. A função objetivo é usualmente descrita na forma $J(\mathbf{w}) = \sum e(\mathbf{w})^2 + \lambda||\mathbf{w}||$ com a norma dos pesos $||\mathbf{w}||$ frequentemente usada como penalização [6]. O efeito de funções objetivo regularizadas é o de deslocar a solução de erro zero de treinamento para $\lambda \neq 0$. Em outras palavras, o erro de treinamento deve aumentar para evitar o efeito de *overfitting* do cálculo da pseudoinversa.

O termo de *crosstalk* da Equação 4.3 implicitamente contribui como um termo de penalização para a solução de erro zero, o que pode ser visto na proposição a seguir.

Proposição 1. *O erro quadrático $\epsilon_k^2 = (y_k - \hat{y}_k)^2$ devido à associação arbitrária k treinada com aprendizado Hebbiano pode ser representado por:*

$$\epsilon_k^2 = \overbrace{[2y_k - \hat{y}_k]^2}^{Erro} + \overbrace{y_k(2\mathbf{x}_k^T\mathbf{x}_k \sum_{i=1, i\neq k}^{N} y_i\mathbf{x}_i^T\mathbf{x}_k - y_k)}^{Penalização}. \tag{4.5}$$

Prova. O erro $\epsilon_k^2 = (y_k - \hat{y}_k)^2$ devido a um padrão arbitrário k, depois de uma expansão é dada por:

$$\epsilon_k^2 = y_k^2 - 2y_k\hat{y}_k + \hat{y}_k^2. \tag{4.6}$$

Substituindo $\hat{y}_k = (\sum_{i=1}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k$ na Equação 4.6 resulta em:

$$\epsilon_k^2 = y_k^2 - 2y_k\left(\sum_{i=1}^{N} y_i\mathbf{x}_i\right)^T\mathbf{x}_k + \left((\sum_{i=1}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k\right)^2. \tag{4.7}$$

Removendo do somatório o termo $y_k\mathbf{x}_k^T\mathbf{x}_k$ devido ao k_{esima} padrão resulta em:

$$\epsilon_k^2 = y_k^2 - 2y_k(y_k\mathbf{x}_k)^T\mathbf{x}_k - 2y_k(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k + [(y_k\mathbf{x}_k)^T\mathbf{x}_k + (\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k]^2. \tag{4.8}$$

Expandindo o termo quadrático no final da Equação 4.8 obtém-se:

$$\epsilon_k^2 = y_k^2 - 2y_k(y_k\mathbf{x}_k)^T\mathbf{x}_k - 2y_k(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k + [(y_k\mathbf{x}_k)^T\mathbf{x}_k]^2 + 2(y_k\mathbf{x}_k)^T\mathbf{x}_k(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k + [(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k]^2. \tag{4.9}$$

Combinando o termo $y_k^2 - 2y_k(y_k\mathbf{x}_k)^T\mathbf{x}_k$ com o termo $[(y_k\mathbf{x}_k)^T\mathbf{x}_k]^2$ da Equação 4.9 e reduzindo para a sua forma quadrática obtém-se:

$$\epsilon_k^2 = [y_k - (y_k\mathbf{x}_k)^T\mathbf{x}_k]^2 + [y_k - (\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k]^2 - y_k^2 + 2(y_k\mathbf{x}_k)^T\mathbf{x}_k(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k. \tag{4.10}$$

Os termos $-y_k^2 + 2(y_k\mathbf{x}_k)^T\mathbf{x}_k(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k$ da Equação 4.10 podem ser reduzidos para a forma quadrática $-[y_k - \mathbf{x}_k^T\mathbf{x}_k(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k]^2 + [\mathbf{x}_k^T\mathbf{x}_k(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k]^2$, obtendo-se:

$$\epsilon_k^2 = [y_k - (y_k\mathbf{x}_k)^T\mathbf{x}_k]^2 + [y_k - (\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k]^2 - [y_k - \mathbf{x}_k^T\mathbf{x}_k(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k]^2 + [\mathbf{x}_k^T\mathbf{x}_k(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k]^2. \tag{4.11}$$

Considerando que os dados de entrada são normalizados ($\mathbf{x}_k^T\mathbf{x}_k = 1$) obtém-

se:

$$\epsilon_k^2 = \overbrace{[y_k - (y_k\mathbf{x}_k)^T\mathbf{x}_k]^2}^{=0} + \overbrace{[y_k - (\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k]^2}^{(y_k-\hat{y}_k+y_k)^2} - \overbrace{[y_k - \mathbf{x}_k^T\mathbf{x}_k(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k]^2}^{(y_k-(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k)^2} + \overbrace{[\mathbf{x}_k^T\mathbf{x}_k(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k]^2}^{((\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k)^2}. \tag{4.12}$$

Como crosstalk = $(\sum_{i=1,i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k)$, obtém-se:

$$\epsilon_k^2 = [2y_k - \hat{y}_k]^2 - (y_k - \text{crosstalk})^2 + \text{crosstalk}^2 \tag{4.13}$$

$$\epsilon_k^2 = [2y_k - \hat{y}_k]^2 - y_k^2 + 2y_k\text{crosstalk} - \text{crosstalk}^2 + \text{crosstalk}^2 \tag{4.14}$$

$$\epsilon_k^2 = [2y_k - \hat{y}_k]^2 - y_k^2 + 2y_k\text{crosstalk} \tag{4.15}$$

$$\epsilon_k^2 = [2y_k - \hat{y}_k]^2 + y_k(2\text{crosstalk} - y_k). \tag{4.16}$$

□

O termo de penalização da Equação 4.5 foi obtido do erro original $(y_k - \hat{y}_k)^2$ com o objetivo de mostrar o efeito residual do aprendizado Hebbiano. O primeiro termo é proporcional ao erro quadrático, enquanto o segundo tem um efeito de penalização devido à relação espacial dos padrões dentro do conjunto de treinamento, uma vez que é relacionado ao *crosstalk* da Equação 4.3. À medida que novos padrões são selecionados para aprendizado, a interferência entre eles tende a aumentar assim como a magnitude do termo de penalização da Equação 4.5. A estratégia de seleção no aprendizado ativo tem, portanto, um impacto direto na penalização e na suavização do *overfitting*.

O gráfico da Figura 4.1 mostra a evolução do erro médio quadrático de treinamento de uma ELM devido aos aprendizados Hebbiano e baseado em pseudoinversa, em relação ao número de padrões aleatoriamente escolhidos para treinamento. O treinamento foi realizado utilizando as Equações 3.9 e 4.1.

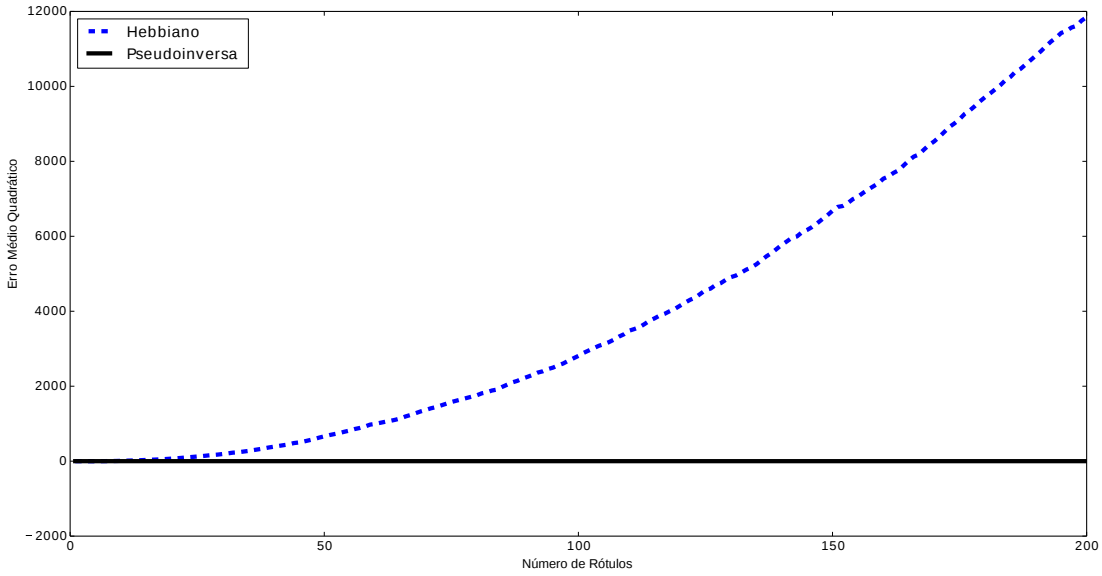


Figura 4.1: Erro médio quadrático no conjunto de treinamento para ELMs treinadas com as Equações 3.9 e 4.1, método da pseudoinversa e aprendizado Hebbiano, respectivamente.

Para evidenciar o efeito do *crosstalk*, a saída da rede foi calculada como sendo $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$ para os dois modelos, ou seja, não foi utilizada a função sinal. Como esperado, o erro do aprendizado Hebbiano aumenta quadraticamente, devido ao primeiro termo da Equação 4.5, enquanto o erro da pseudoinversa é muito próximo de zero independentemente do número de padrões selecionados para aprendizado, mesmo sem o uso da função sinal na saída da rede. Esse comportamento muda drasticamente quando o erro é calculado para o conjunto de teste, como pode ser visto na Figura 4.2. É importante enfatizar que as Figuras 4.1 e 4.2 estão em escalas diferentes no eixo y, por causa da discrepância da magnitude dos erros, porém as escalas não afetam a análise qualitativa. Pode ser observado que o erro do método da pseudoinversa aumenta drasticamente próximo de $N = p$, um comportamento que é compatível com aquele apresentado pela AUC na Figura 3.3.

Enquanto o erro de teste do aprendizado Hebbiano continua com comportamento quadrático, o erro da pseudoinversa para o conjunto de teste é muito maior, principalmente nas proximidades de $N = p$ devido ao efeito do *overfitting*.

Esse tipo de comportamento indica que o *crosstalk*, que é frequentemente considerado como uma limitação do aprendizado Hebbiano [45, 53, 54, 27], pode ter um efeito positivo no aprendizado de ELMs. A sua contribuição para a suavização da saída dependerá, contudo, da habilidade do método de aprendizado de controlar a sua magnitude, que é um dos objetivos do método proposto

nesta tese, e será detalhado no Capítulo 5.

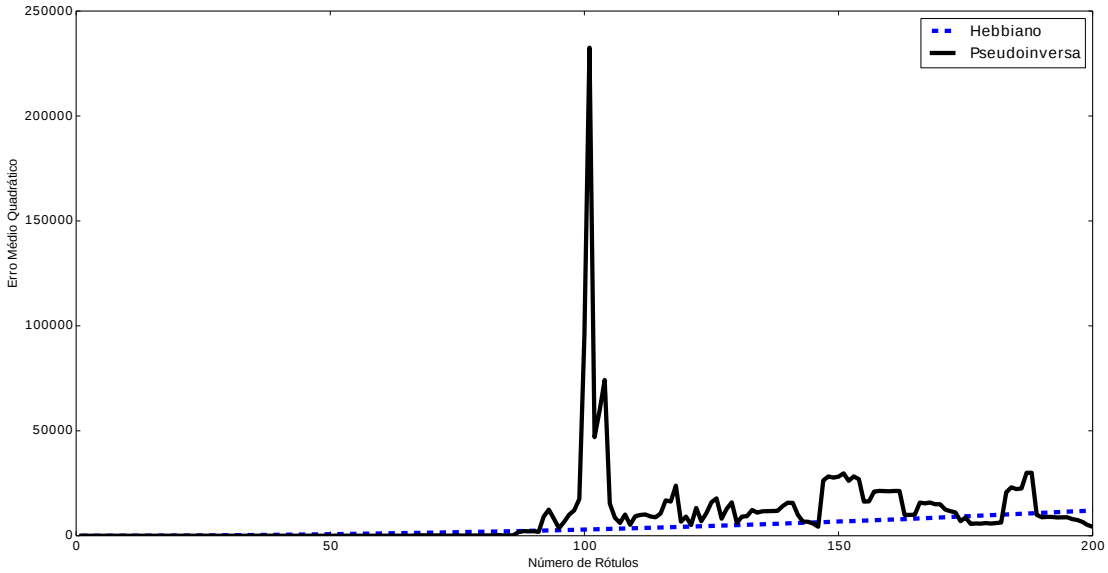


Figura 4.2: Erro no conjunto de teste para ELMs treinadas com as Equações 3.9 e 4.1, método da pseudoinversa e aprendizado Hebbiano, respectivamente. O erro da pseudoinversa aumenta bruscamente próximo de $N = p$.

O modelo resultante da Equação 4.1 é simplesmente um Perceptron [55] treinado com aprendizado Hebbiano. Uma variação com pesos normalizados, como apresentado na Equação 4.20, mostrou uma tendência à maximização da margem [21], podendo ser considerada uma versão simplificada das SVMs onde todos os multiplicadores de Lagrange seriam iguais a 1. Essa característica tende a tornar o aprendizado Hebbiano menos susceptível aos possíveis efeitos negativos do *crosstalk*, sendo, portanto, o modelo de separador linear escolhido para a realização do Aprendizado Ativo nas ELMs. Esse Perceptron será detalhado na seção a seguir.

4.2 Perceptron Hebbiano com Pesos Normalizados

Auer et al. [5] propuseram um algoritmo que consiste em atualizar o vetor de pesos de um Perceptron de forma que o produto escalar $\mathbf{w}^T\mathbf{x}_k$ tenha o sinal correto[1] e também que garanta que os padrões ficarão distantes do hiperplano por pelo menos uma certa margem de segurança $\gamma > 0$ a fim de garantir uma boa capacidade de generalização. O algoritmo proposto é denominado regra P-Delta e os pesos são atualizados nas seguintes condições:

[1] $\mathbf{x}_k$ representa um padrão de treinamento e $k = 1,...,N$, em que N é o número de padrões do conjunto de treinamento. Os vetores $\mathbf{x}$ e $\mathbf{w}$ são vetores aumentados, ou seja, $\mathbf{x} = [x_1, x_2, ..., x_m, 1]^T$ e $\mathbf{w} = [w_1, w_2, ..., w_m, bias]^T$, em que $bias$ é o limiar de ativação do Perceptron. Essa notação será utilizada ao longo de todo o trabalho.

- Quando há uma classificação incorreta, ou seja, quando o sinal de $\mathbf{w}^T\mathbf{x}_k$ for diferente de y_k (saída desejada para cada padrão, que deve ser 1 ou -1);

- Quando a classificação é correta, mas o valor da margem do padrão é menor que uma certa margem de segurança $|\mathbf{w}^T\mathbf{x}_k| < \gamma$, o que significa que uma pequena variação no produto escalar pode fazer com que o sinal fique incorreto.

O vetor de pesos deve ser normalizado ($\|\mathbf{w}\| = 1$) a fim de que a comparação entre $\mathbf{w}^T\mathbf{x}_k$ e γ não seja influenciada pelo norma dos pesos. Esse Perceptron é utilizado em uma rede de Perceptrons paralelos, sendo que o valor de γ deve ser ajustado ao longo do processo, o que é indesejado.

Utilizando essas ideias Fernandez-Delgado et al. [21] reescreveram os critérios de ajuste dos pesos da seguinte forma:

- Os pesos são atualizados quando $y_k = +1$ e $\mathbf{w}^T\mathbf{x}_k < \gamma$;

- Os pesos são atualizados quando $y_k = -1$ e $\mathbf{w}^T\mathbf{x}_k > -\gamma$;

Como pode ser observado, os critérios continuam os mesmos, porém a forma como foram reescritos possibilita que a expressão para o erro seja escrita como uma função linear sendo $\gamma - \mathbf{w}^T\mathbf{x}_k$ para $y_k = +1$ e $\gamma + \mathbf{w}^T\mathbf{x}_k$ para $y_k = -1$. Dessa forma, a expressão para o erro para um padrão $\mathbf{x}_k$ pode ser escrita da seguinte forma:

$$\text{erro}(\mathbf{x}_k) = \begin{cases} \gamma - y_k\mathbf{w}^T\mathbf{x}_k & \text{se} \quad \gamma - y_k\mathbf{w}^T\mathbf{x}_k \geq 0 \\ 0 & \text{caso contrário.} \end{cases} \tag{4.17}$$

Quando o Perceptron acerta e a distância do ponto em relação ao hiperplano é maior que a margem limite o erro é zero, caso contrário será um valor positivo. Somando essa expressão de erro para todos os padrões de treinamento obtém-se o valor exato da soma dos erros cometidos pelo Perceptron:

$$E^0(\mathbf{w}) = \sum_{k=1}^{N}[\gamma - y_k\mathbf{w}^T\mathbf{x}_k]^+, \quad \text{em que } [z]^+ = \max(z,0) \tag{4.18}$$

Segundo os autores, essa medida de erro mede duas contribuições. A primeira é referente à margem de um padrão $\mathbf{x}_k$ em relação ao hiperplano separador, uma vez que, se o padrão é corretamente classificado, quanto maior for o valor de $\mathbf{w}^T\mathbf{x}_k$ menor será o erro e se a margem for maior que γ o erro para esse padrão será zero. A segunda refere-se ao erro de treinamento uma vez que $y_k\mathbf{w}^T\mathbf{x}_k$ é positivo somente quando y_k e $\mathbf{w}^T\mathbf{x}_k$ tem o mesmo sinal o que reduz o erro pela subtração presente na equação 4.18.

Para calcular o valor de w que minimiza $E^0(\mathbf{w})$ é necessário que $\nabla E^0(\mathbf{w}) = \mathbf{0}$, porém a função $[z]^+$ não é diferenciável em z = 0. Em virtude disso os autores propuseram utilizar uma aproximação linear $[z]^+ \rightarrow z$. Os padrões em que $y_k \mathbf{w}^T \mathbf{x}_k > \gamma$ passam a ser levados em consideração e ajudam a reduzir o valor do erro. Dessa forma, o erro aproximado passa a ser:

$$E(\mathbf{w}) = \sum_{k=1}^{N} (\gamma - y_k \mathbf{w}^T \mathbf{x}_k) = \gamma N - \sum_{k=1}^{N} (y_k \mathbf{w}^T \mathbf{x}_k). \tag{4.19}$$

Como γN é um valor constante, para minimizar $E(\mathbf{w})$ basta encontrar w que maximiza o valor de $\sum_{k=1}^{N} y_k \mathbf{w}^T \mathbf{x}_k$ e que $\|\mathbf{w}\| = 1$ conforme explicado anteriormente. Retirando o termo $\mathbf{w}^T$ do somatório tem-se que $\mathbf{w}^T \sum_{k=1}^{N} y_k \mathbf{x}_k$. Segundo os autores para maximizar esse valor basta encontrar w que tenha norma 1 e que seja paralelo a $\sum_{k=1}^{N} y_k \mathbf{x}_k$. Dessa forma tem-se que:

$$\mathbf{w}_0 = \frac{\sum_{k=1}^{N} y_k \mathbf{x}_k}{\| \sum_{k=1}^{N} y_k \mathbf{x}_k \|}. \tag{4.20}$$

Como pode ser observado, a equação 4.20 apresenta uma abordagem analítica para o treinamento de Perceptrons. Segundo os autores, o algoritmo proposto funciona como uma SVM em que todos os padrões de treinamento são vetores de suporte e possuem multiplicadores de Lagrange iguais a 1. Para se obter a saída do Perceptron para um padrão desejado basta fazer $y(\mathbf{x}) = \mathbf{sign}(\mathbf{w}_0^T \mathbf{x})$.

Essa abordagem, além de dispensar o ajuste do parâmetro C das SVMs, permite também o treinamento *on-line*, pois basta somar os novos padrões, multiplicados por sua saída desejada, ao vetor de pesos e normalizar novamente o vetor resultante dividindo-o por sua norma, a fim de se obter $||\mathbf{w}_0|| = 1$.

Como pode ser observado, o Perceptron proposto por Fernandez-Delgado *et al.* [21] utiliza de forma analítica os padrões de treinamento para ajustar os pesos do Perceptron, multiplicados pelos seus rótulos. Esse cálculo nada mais é do que a regra de Hebb com pesos normalizados e com a função sinal como regra de decisão, o que pode ser visto comparando-se a Equação 4.4 com a Equação 4.20. No próximo capítulo será mostrado como controlar a evolução do *crosstalk* e realizar o Aprendizado Ativo por meio de um simples teste de convergência desenvolvido para o Perceptron Hebbiano com pesos normalizados.

4.3 Conclusões do Capítulo

Neste capítulo foi discutido o uso do aprendizado Hebbiano no lugar da pseudoinversa para evitar o sobreajuste nas ELMs e permitir o treinamento

on-line. Foi demonstrado que o termo residual do aprendizado Hebbiano, denominado *crosstalk*, tem um efeito regularizador, deslocando a solução obtida no treinamento da solução de erro zero. Por fim, foi apresentado um Perceptron treinado com aprendizado Hebbiano e com pesos normalizados que, segundo os autores, funciona como uma SVM em que todos os padrões utilizados são vetores de suporte e que todos os multiplicadores de Lagrange são iguais a 1. No próximo capítulo serão apresentados os métodos de Aprendizado Ativo desenvolvidos nesta tese.

CAPÍTULO 5

Métodos Propostos

Como exposto anteriormente, vários autores demonstraram que métodos simples de Aprendizado Ativo podem ter bons resultados em problemas linearmente separáveis. O objetivo desta tese é estender essa abordagem para problemas não linearmente separáveis, em que não seja necessário o ajuste fino de parâmetros livres e que seja possível o treinamento *on-line*. Neste capítulo será proposto um modelo neural baseado em uma camada escondida do tipo ELM (*Extreme Learning Machine*) para realizar a linearização necessária nos dados, sem a necessidade de ajuste fino de parâmetros livres, de forma que possam ser utilizados métodos lineares de Aprendizado Ativo. Ao longo do capítulo serão apresentadas duas estratégias de Aprendizado Ativo baseadas em um Perceptron treinado com a regra de Hebb [31] e com pesos normalizados [21].

Para realizar o Aprendizado Ativo é necessário determinar o número de padrões que seriam suficientes para se obter uma boa aproximação da função geradora dos dados. Na seção a seguir será determinado o número de padrões necessários para que o Perceptron Hebbiano com pesos normalizados [21] convirja para uma solução com boa capacidade de generalização em problemas de classificação binária.

5.1 Número Limite de Padrões de Treinamento

No contexto deste trabalho, uma questão que ainda precisa ser respondida é: como estimar o número mínimo de padrões necessários para realizar o aprendizado e obter boa capacidade de generalização? Isso é particularmente importante para suavizar o efeito do *crosstalk*, controlar o custo de rotulação

no Aprendizado Ativo e reduzir o tamanho do conjunto de treinamento.

Para um Perceptron treinado com o algoritmo de Rosenblatt [55], que é baseado em correção do erro, existe um teorema que define que o algoritmo converge com um número limite de iterações que é menor ou igual a um número máximo de classificações incorretas, caso o problema seja linearmente separável [30]. No contexto deste trabalho é assumido que o problema é linearmente separável, em virtude da propagação dos padrões pela camada escondida ELM. Baseado nessa premissa, é mostrado a seguir que a prova de convergência do Perceptron proposta por Nilsson [51, 30] pode ser estendida para um Perceptron treinado com aprendizado Hebbiano e com pesos normalizados.

A prova do teorema a seguir garante a estimativa de um número limite de padrões que é suficiente para encontrar um separador linear e, consequentemente, produzir um limite superior que garanta a convergência [35, 36, 37]. Considerando um conjunto de treinamento reduzido, o aprendizado pode ser interrompido para evitar o aumento no termo de penalização da Equação 4.5 devido ao *crosstalk*. Como será discutido nas próximas seções, o número limite de padrões fornecido pelo teorema a seguir na verdade resulta em um valor de *crosstalk* com boa relação custo-benefício em termos do erro de treinamento e da complexidade do modelo, o que leva a um classificador com boa capacidade de generalização.

Teorema 1. *Para duas classes linearmente separáveis o número máximo de rótulos necessários para a convergência do Perceptron Hebbiano com pesos normalizados é dado por:*

$$t_{max} = \frac{\beta + 2\theta}{\alpha^2},$$

em que β é a norma quadrática máxima entre os padrões usados para treinamento:

$$\beta = \max_{\mathbf{x} \in \zeta} ||\mathbf{x}||^2,$$

θ é a margem do padrão mais distante do hiperplano separador:

$$\theta = \max_{\mathbf{x} \in \zeta} |\mathbf{w}^T \mathbf{x}|,$$

e α é a margem do padrão mais próximo do hiperplano separador:

$$\alpha = \min_{\mathbf{x} \in \zeta} |\mathbf{w}^T \mathbf{x}|.$$

Prova. Suponha que se deseja classificar duas classes linearmente separáveis ζ_1 e ζ_2 pertencentes ao conjunto de dados ζ e que o vetor de pesos inicial é

$\mathbf{w}_0 = \mathbf{0}$. Para padrões $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t$ com rótulos $y_1, y_2, ..., y_t \in \{-1, +1\}$ o vetor de pesos atualizados $\mathbf{w}_{t+1}$ é dado por:

$$\mathbf{w}_{t+1} = \mathbf{x}_1 y_1 + \mathbf{x}_2 y_2 + ... + \mathbf{x}_t y_t. \tag{5.1}$$

Como as classes ζ_1 e ζ_2 são linearmente separáveis, existe uma solução $\mathbf{w}^*$ que corretamente classifica todos os padrões. Dessa forma, para uma solução fixa $\mathbf{w}^*$ pode-se definir um número positivo α:

$$\alpha = \min_{\mathbf{x}_t \in \zeta} |\mathbf{w}^{*T} \mathbf{x}_t y_t| = \min_{\mathbf{x}_t \in \zeta} |\mathbf{w}^{*T} \mathbf{x}_t|. \tag{5.2}$$

Multiplicando os dois lados da Equação 5.1 por $\mathbf{w}^{*T}$, obtém-se:

$$\mathbf{w}^{*T} \mathbf{w}_{t+1} = \mathbf{w}^{*T} \mathbf{x}_1 y_1 + \mathbf{w}^{*T} \mathbf{x}_2 y_2 + ... + \mathbf{w}^{*T} \mathbf{x}_t y_t. \tag{5.3}$$

De acordo com a definição da Equação 5.2 pode-se garantir que:

$$\mathbf{w}^{*T} \mathbf{w}_{t+1} \geq t\alpha. \tag{5.4}$$

Utilizando a desigualdade de Cauchy-Schwarz [66] obtém-se:

$$||\mathbf{w}^*||^2 ||\mathbf{w}_{t+1}||^2 \geq [\mathbf{w}^{*T} \mathbf{w}_{t+1}]^2. \tag{5.5}$$

Observando a Equação 5.4 pode-se concluir que $[\mathbf{w}^{*T} \mathbf{w}_{t\,+\,1}]^2 \geq t^2 \alpha^2$, ou de forma equivalente:

$$||\mathbf{w}_{t+1}||^2 \geq \frac{t^2 \alpha^2}{||\mathbf{w}^*||^2}. \tag{5.6}$$

A Equação 5.1 pode ser escrita como $\mathbf{w}_{k+1} = \mathbf{w}_k + y_k \mathbf{x}_k$, onde $k = 1,...,t$. Tomando o quadrado da norma Euclidiana nos dois lados dessa equação obtém-se:

$$||\mathbf{w}_{k+1}||^2 = ||\mathbf{w}_k||^2 + ||\mathbf{x}_k y_k||^2 + 2\mathbf{w}_k^T \mathbf{x}_k y_k. \tag{5.7}$$

A Equação 5.7 pode agora ser reescrita da seguinte forma:

$$||\mathbf{w}_{k+1}||^2 - ||\mathbf{w}_k||^2 = ||\mathbf{x}_k y_k||^2 + 2\mathbf{w}_k^T \mathbf{x}_k y_k. \tag{5.8}$$

Somando os termos $k = 1,...,t$ da Equação 5.8 e considerando a condição inicial $\mathbf{w}_0 = \mathbf{0}$ tem-se:

$$||\mathbf{w}_{t+1}||^2 = \sum_{k=1}^{t} ||\mathbf{x}_k y_k||^2 + 2 \sum_{k=1}^{t} \mathbf{w}_k^T \mathbf{x}_k y_k. \tag{5.9}$$

Como $||\mathbf{x}_k y_k||^2$ = $||\mathbf{x}_k||^2$, uma vez que $y_k \in \{-1, +1\}$, pode-se definir:

$$\beta = \max_{\mathbf{x}_k \in \zeta} ||\mathbf{x}_k||^2 \tag{5.10}$$

e:

$$\theta = \max_{\mathbf{x}_k \in \zeta} |\mathbf{w}_k^T \mathbf{x}_k y_k| = \max_{\mathbf{x}_k \in \zeta} |\mathbf{w}_k^T \mathbf{x}_k|. \tag{5.11}$$

Usando as definições das Equações 5.10 e 5.11, pode-se garantir que a seguinte desigualdade é verdadeira:

$$||\mathbf{w}_{t+1}||^2 \leq (\beta + 2\theta)t. \tag{5.12}$$

A partir da Equação 5.12 pode-se concluir que o quadrado da norma do vetor de pesos cresce linearmente com o valor de t. Esse resultado é conflitante com aquele apresentado na Equação 5.6 para valores suficientemente grandes de t.

Nesse caso, o valor de t não pode ser maior que um certo valor t_{max} para o qual as Equações 5.6 e 5.12 são iguais:

$$\frac{t_{max}^2 \alpha^2}{||\mathbf{w}^*||^2} = t_{max}(\beta + 2\theta). \tag{5.13}$$

Assim, o valor de t_{max} pode ser obtido pela Equação 5.14:

$$t_{max} = \frac{(\beta + 2\theta)||\mathbf{w}^*||^2}{\alpha^2}. \tag{5.14}$$

Considerando que o classificador utilizará o vetor de pesos final $\mathbf{w}^*$ normalizado [21] tem-se que $||\mathbf{w}^*|| = 1$ e a Equação 5.14 pode ser reescrita da seguinte forma:

$$t_{max} = \frac{\beta + 2\theta}{\alpha^2}. \tag{5.15}$$

□

A Equação 5.15 indica que o Perceptron treinado com aprendizado Hebbiano e com o vetor de pesos normalizado [21] converge utilizando no máximo $\frac{\beta+2\theta}{\alpha^2}$ rótulos. Como será mostrado na próxima seção, o valor de t_{max} pode ser estimado durante o treinamento e utilizado como parte de um critério de seleção de rótulos e, além disso, também pode ser utilizado para determinar o número de padrões que devem ser aprendidos.

5.2 Estratégia de Aprendizado Ativo Baseada em um Teste de Convergência

O teorema apresentado na seção anterior pode ser utilizado como critério de seleção de rótulos para Aprendizado Ativo, uma vez que indica a quantidade de rótulos necessária para garantir a convergência de um Perceptron treinado com aprendizado Hebbiano. A decisão em rotular ou não um padrão é baseada em estimativas sucessivas de t_{max} durante o processo de aprendizado. Como a convergência deve ocorrer com um número reduzido de padrões, a influência do *crosstalk* pode ser controlada e uma suavização pode ser obtida.

O classificador proposto é composto por uma camada escondida ELM e uma camada de saída treinada utilizando-se o aprendizado Hebbiano com pesos normalizados [21]. A Figura 5.1 apresenta a topologia proposta.

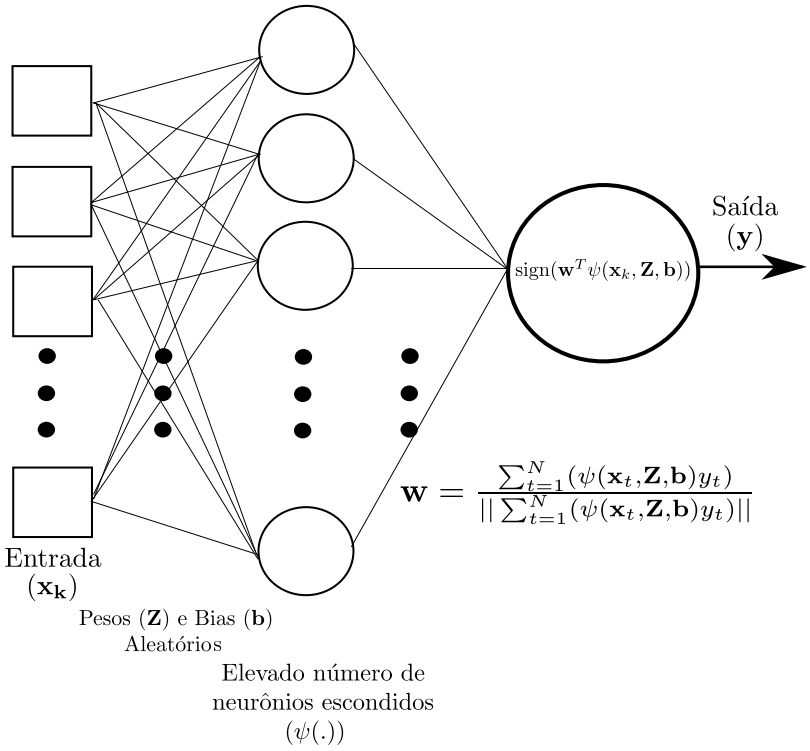


Figura 5.1: Topologia do classificador proposto

Com o objetivo de selecionar somente os padrões mais informativos, a estratégia de Aprendizado Ativo utiliza um teste de convergência como um critério para seleção de rótulos cada vez que um novo padrão é apresentado. Cada novo padrão é propagado pela camada escondida ELM e então sua margem é calculada em relação ao hiperplano corrente. Esse valor é atribuído ao α da Equação 5.15. A variável θ corresponde ao valor máximo entre o α calculado a os αs anteriores. A variável β é o valor máximo entre o quadrado da norma do padrão corrente e o quadrado da maior norma entre os padrões de trei-

namento anteriores. Utilizando essas variáveis, t_{max} é calculado utilizando a Equação 5.15.

A partir disso, pode-se definir dois algoritmos de Aprendizado Ativo, sendo um do tipo *Stream-based Selective Sampling* e o outro do tipo *Pool-based Sampling*:

- **Extreme Active Learning Machine Selective Sampling - EALMSS** [36, 37]: Nesta versão o teste de convergência funciona como um critério para seleção de rótulos e é realizado utilizando o valor calculado de t_{max} para o padrão corrente. Se t_{max} é maior que o número de padrões utilizados para treinamento, então o algoritmo não convergiu do ponto de vista do padrão corrente, sendo que dessa forma o padrão deve fornecer informações novas e por isso deve ser aprendido. Seu rótulo é solicitado ao especialista e os pesos do Perceptron são ajustados utilizando a Equação 5.16. Se t_{max} é menor ou igual ao número de padrões utilizados para treinamento, então o algoritmo convergiu e não é necessário solicitar o rótulo do padrão corrente, pois provavelmente ele é redundante, podendo ser descartado. O processo continua enquanto novos padrões são apresentados ou até que o número máximo de rótulos disponíveis é alcançado. A Figura 5.2 apresenta o fluxograma do método EALMSS e a Figura 5.3 apresenta o pseudocódigo.

- **Extreme Active Learning Machine Pool-based Sampling - EALMPB** [35]: Nesta versão o teste de convergência funciona como um critério de parada para o algoritmo. Todos os dados não rotulados a serem analisados devem estar disponíveis em um conjunto de dados C (um *pool* de tamanho limitado). Todos os padrões do conjunto de dados são propagados pela camada escondida, formando o conjunto C_{ELM}, e cada vetor resultante terá calculada a sua margem em relação ao hiperplano corrente. O vetor que tiver a menor margem, ou seja, o mais próximo do hiperplano separador utilizado na análise, será utilizado para calcular o valor de t_{max}. O padrão terá seu rótulo solicitado e será utilizado para ajustar os pesos do perceptron utilizando a Equação 5.16 somente se o t_{max} calculado for maior que o número de padrões já utilizados para treinamento, indicando que o número de padrões utilizados até o momento é insuficiente para garantir a convergência. Caso contrário o padrão é descartado e o algoritmo termina, pois o perceptron terá convergido. A Figura 5.4 apresenta o fluxograma do método EALMPB e Figura 5.5 apresenta o pseudocódigo.

$$\mathbf{w}_{t+1} = \frac{\mathbf{w}_t + \psi(\mathbf{x}_t, \mathbf{Z}, \mathbf{b})y_t}{||\mathbf{w}_t + \psi(\mathbf{x}_t, \mathbf{Z}, \mathbf{b})y_t||} \tag{5.16}$$

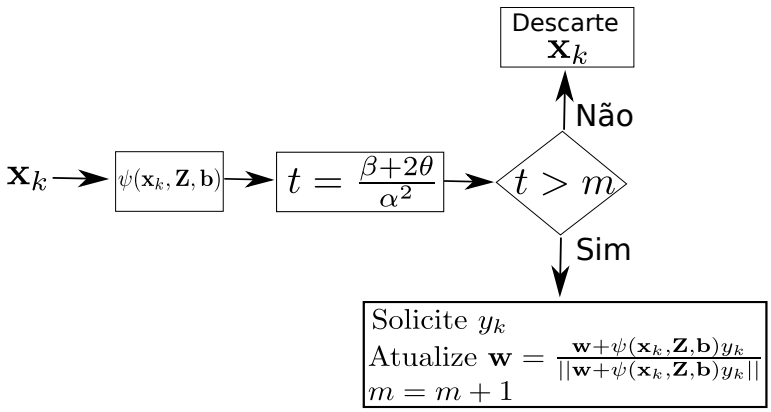


Figura 5.2: Fluxograma do método EALMSS

Entrada: Tamanho inicial do conjunto de treinamento m, número máximo de rótulos disponíveis L, matriz de pesos aleatórios ELM $\mathbf{Z}$, vetor de *bias* aleatórios $\mathbf{b}$, função geradora F
Saída : Vetor de pesos $\mathbf{w}$
Método :

Selecione aleatoriamente m padrões $\mathbf{x}_k$ da função geradora F;
Propague os m padrões pela camada escondida ELM ($\psi(\mathbf{x}_k,\mathbf{Z},\mathbf{b})$) e solicite seus rótulos y_k;
$\mathbf{w} = \frac{\sum_{k=1}^{m} \psi(\mathbf{x}_k,\mathbf{Z},\mathbf{b})y_k}{||\sum_{k=1}^{m} \psi(\mathbf{x}_k,\mathbf{Z},\mathbf{b})y_k||}$;
$\beta = \max_{k=1,\ldots,m}(||\psi(\mathbf{x}_k,\mathbf{Z},\mathbf{b})||^2)$;
$\theta = \max_{k=1,\ldots,m}(|\mathbf{w}^T\psi(\mathbf{x}_k,\mathbf{Z},\mathbf{b})|)$;
repita
    Seleciona aleatoriamente um padrão $\mathbf{x}$ de F;
    $\alpha = |\mathbf{w}^T\psi(\mathbf{x},\mathbf{Z},\mathbf{b})|$;
    $\beta = \max(||\psi(\mathbf{x},\mathbf{Z},\mathbf{b})||^2, \beta)$;
    $\theta = \max(|\mathbf{w}^T\psi(\mathbf{x},\mathbf{Z},\mathbf{b})|,\theta)$;
    $t = \frac{\beta+2\theta}{\alpha^2}$;
    se $t > m$ **então**
        Solicite o rótulo y;
        Atualize $\mathbf{w} = \frac{\mathbf{w}+\psi(\mathbf{x},\mathbf{Z},\mathbf{b})y}{||\mathbf{w}+\psi(\mathbf{x},\mathbf{Z},\mathbf{b})y||}$;
        $m = m + 1$;
    fim
até $(m = L)$ or $(F = \emptyset)$;

Figura 5.3: Estratégia de Aprendizado Ativo - *Selective Sampling*

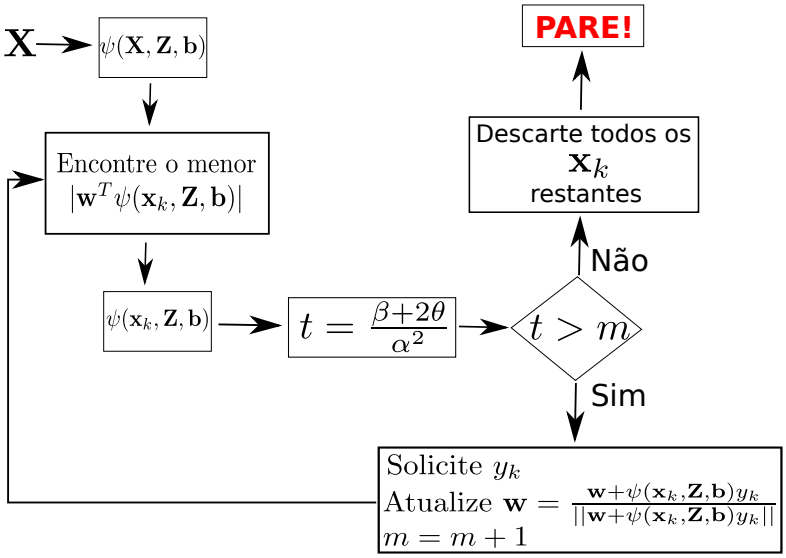


Figura 5.4: Fluxograma do método EALMPB

Entrada: Conjunto de padrões C, tamanho inicial do conjunto de treinamento m, matriz de pesos aleatórios ELM $\mathbf{Z}$ e vetor de *bias* aleatórios $\mathbf{b}$
Saída : Vetor de pesos $\mathbf{w}$
Método :

Propague todos os padrões de C pela camada escondida formando o conjunto C_{ELM};
Retire aleatoriamente m padrões de C_{ELM} e solicite os rótulos;
$\mathbf{w} = \frac{\sum_{k=1}^{m} \psi(\mathbf{x}_k,\mathbf{Z},\mathbf{b})y_k}{||\sum_{k=1}^{m} \psi(\mathbf{x}_k,\mathbf{Z},\mathbf{b})y_k||}$;
$\beta = \max_{k=1,\ldots,m}(||\psi(\mathbf{x}_k,\mathbf{Z},\mathbf{b})||^2)$;
$\theta = \max_{k=1,\ldots,m}(|\mathbf{w}^T\psi(\mathbf{x}_k,\mathbf{Z},\mathbf{b})|)$;
Faça n = Inf;
enquanto $(t > m)$ e $(C_{ELM} \neq \emptyset)$ **faça**
    Calcule $\alpha = |\mathbf{w}^T\psi(\mathbf{x},\mathbf{Z},\mathbf{b})|$ para todos os padrões de C_{ELM} e escolha o menor α;
    $\beta = \max(||\psi(\mathbf{x},\mathbf{Z},\mathbf{b})||^2, \beta)$;
    $\theta = \max(|\mathbf{w}^T\psi(\mathbf{x},\mathbf{Z},\mathbf{b})|,\theta)$;
    $t = \frac{\beta+2\theta}{\alpha^2}$;
    se $t > m$ **então**
        Retire $\psi(\mathbf{x},\mathbf{Z},\mathbf{b})$ de C_{ELM};
        Solicite o rótulo y;
        Atualize $\mathbf{w} = \frac{\mathbf{w}+\psi(\mathbf{x},\mathbf{Z},\mathbf{b})y}{||\mathbf{w}+\psi(\mathbf{x},\mathbf{Z},\mathbf{b})y||}$;
        $m = m + 1$;
    fim
fim

Figura 5.5: Estratégia de Aprendizado Ativo - *Pool-Based Sampling*

A estratégia de Aprendizado Ativo proposta neste trabalho é capaz de encontrar uma solução que maximize a capacidade de generalização do modelo e que controla o termo de *crosstalk*, tanto para a versão *Stream-based* quanto para a *Pool-based*. Para ilustrar essas características, quatro estratégias de seleção de padrões foram aplicadas ao conjunto de dados da Figura 3.3. Em todos os casos foi utilizada a mesma camada escondida ELM com 100 neurônios escondidos. A primeira estratégia utiliza o aprendizado Hebbiano com pesos normalizados para aprender, iterativamente, o padrão mais próximo do hiperplano separador, como proposto na heurística de Schohn *et al.* [58] (apresentada na Figura 5.6 como *ELMPCP - ELM with Hebbian Perceptron and Closest Patterns*). A segunda estratégia usa o aprendizado Hebbiano com pesos normalizados para aprender padrões selecionados aleatoriamente (apresentados na Figura 5.6 como *ELMPRP - ELM with Hebbian Perceptron and Random Patterns*). As duas últimas estratégias utilizam os métodos de Aprendizado Ativo baseados no teste de convergência, apesentadas como *Extreme Active Learning Machine (EALM)* nas versão *Stream-based Selective Sampling* (EALMSS) e *Pool-based Sampling* (EALMPB). Para fins de comparação, também foi incluído um classificador ELM com o uso da pseudoinversa e padrões selecionados aleatoriamente. O experimento foi realizado utilizando validação cruzada do tipo *10-fold* dez vezes. Os resultados são apresentados na Figura 5.6 em termos da AUC média.

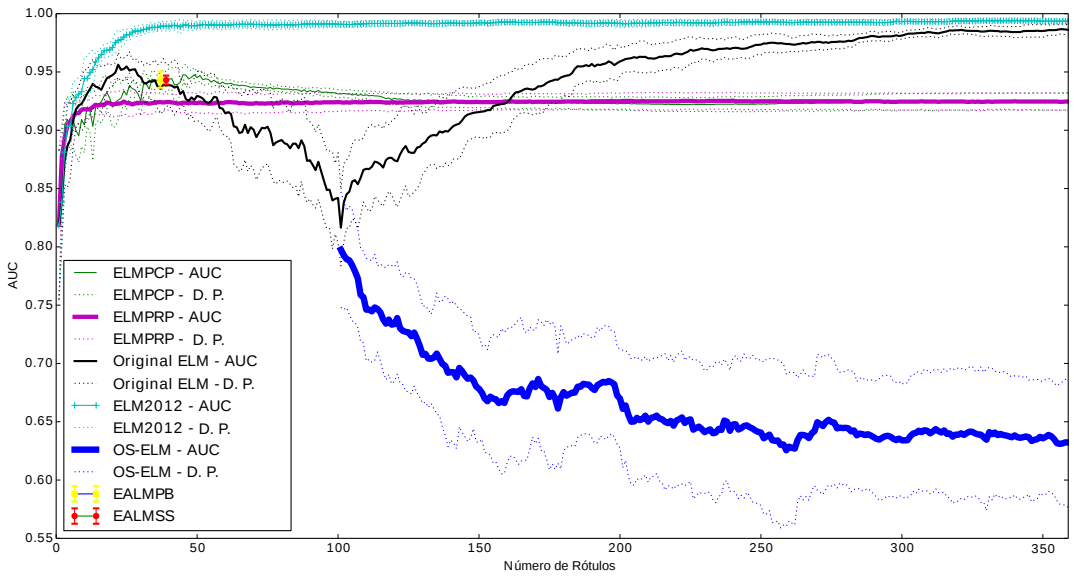


Figura 5.6: Resultados para validação cruzada do tipo *10-fold* para diferentes estratégias de seleção de padrões aplicadas no conjunto de dados da Figura 3.3

Considerando um conjunto de treinamento reduzido, a heurística de

Schohn *et al.* [58] obtém melhores resultados do que a seleção aleatória. Entretanto, a heurística de Schohn é consideravelmente mais custosa em termos de tempo de treinamento, uma vez que requer que em cada iteração a distância em relação ao hiperplano seja calculada para todos os padrões não rotulados. A estratégia de Aprendizado Ativo proposta neste trabalho, em suas duas versões, obtém um bom desempenho em termos de AUC. A primeira (EALMSS) calcula a margem apenas para o padrão analisado em cada iteração. A segunda (EALMPB) calcula a margem de todos os padrões disponíveis em cada iteração, utilizando o teste de convergência como condição de parada.

Uma vantagem da versão EALMSS é que ela não necessita de um critério de parada, pois basta analisar cada um dos dados disponíveis apenas uma vez. Além disso, é importante notar que apesar de o algoritmo EALMSS ser do tipo *Stream-based* [11], seu desempenho é similar ao do algoritmo EALMPB que, por sua vez, é próxima ao melhor classificador que pode ser obtido utilizando a estratégia de Schohn, sendo ambas do tipo *pool-based* [58]. A versão regularizada das ELMs obteve melhores resultados que as outras estratégias, mas para esse método um parâmetro de regularização teve que ser finamente ajustado utilizando trinta porcento do conjunto de treinamento, o que pode ser proibitivo em aplicações reais de Aprendizado Ativo.

5.3 Interpretação do teste de convergência

Como a estratégia de Aprendizado Ativo apresentada neste trabalho é baseada no Perceptron Hebbiano torna-se importante verificar qual é o efeito do *crosstalk* no cálculo de t_{max}. Para tanto, deve-se encontrar os valores de y_k e $\hat{y}_k$ em função do *crosstalk* e do valor de α, que correspondem à contribuição do padrão analisado ao cálculo de t_{max}.

O rótulo y_k do padrão analisado é desconhecido no início do Aprendizado Ativo. O objetivo é determinar se esse rótulo deve ou não ser solicitado ao especialista. Uma forma de estimá-lo é considerar que o hiperplano separador analisado é capaz de classificá-lo corretamente, ou seja, que tem boa capacidade de generalização, sendo capaz de classificar corretamente um padrão ainda desconhecido. Dessa forma pode-se encontrar o valor estimado de y_k em função do *crosstalk* estimado:

$$\mathbf{w}^T\mathbf{x}_k = (\sum_{i=1}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k \tag{5.17}$$

$$\mathbf{w}^T\mathbf{x}_k = y_k\mathbf{x}_k^T\mathbf{x}_k + \overbrace{(\sum_{i\neq k}^{N} y_i\mathbf{x}_i)^T\mathbf{x}_k}^{\text{crosstalk}} \tag{5.18}$$

$$\mathbf{w}^T\mathbf{x}_k = y_k\mathbf{x}_k^T\mathbf{x}_k + \text{crosstalk} \tag{5.19}$$

$$y_k\mathbf{x}_k^T\mathbf{x}_k = \mathbf{w}^T\mathbf{x}_k - \text{crosstalk} \tag{5.20}$$

$$y_k = \frac{\mathbf{w}^T\mathbf{x}_k - \text{crosstalk}}{\mathbf{x}_k^T\mathbf{x}_k} \tag{5.21}$$

$$\mathbf{x}_k^T\mathbf{x}_k = ||\mathbf{x}_k||^2 \tag{5.22}$$

$$y_k = \frac{\mathbf{w}^T\mathbf{x}_k - \text{crosstalk}}{||\mathbf{x}_k||^2}. \tag{5.23}$$

A equação 5.23 apresenta uma estimativa do rótulo de $\mathbf{x}_k$ em função da sua norma e do *crosstalk* estimado, ou seja, o *crosstalk* devido aos padrões utilizados para a construção do hiperplano representado por $\mathbf{w}^T$. Essa estimativa foi possível ao se considerar que a informação trazida por $\mathbf{x}_k$ já está presente no vetor $\mathbf{w}^T$, de forma que esse padrão seria redundante. Tal hipótese é verificada no teste de convergência da equação 5.14 e, caso seja confirmada, o padrão é descartado e seu rótulo não é solicitado.

Além das considerações apresentadas, é necessário encontrar o valor de $\hat{y}_k$, ou seja, a classificação fornecida pelo Perceptron:

$$\hat{y}_k = \text{sign}(\mathbf{w}^T\mathbf{x}_k) \tag{5.24}$$

$$\text{sign}(\mathbf{w}^T\mathbf{x}_k) = \frac{\mathbf{w}^T\mathbf{x}_k}{|\mathbf{w}^T\mathbf{x}_k|} \tag{5.25}$$

$$\hat{y}_k = \frac{\mathbf{w}^T\mathbf{x}_k}{|\mathbf{w}^T\mathbf{x}_k|}. \tag{5.26}$$

Dessa forma, o erro de classificação para o padrão $\mathbf{x}_k$ será obtido comparando-se o valor resultante da hipótese de que a informação de $\mathbf{x}_k$ já está presente no hiperplano corrente $\mathbf{w}^T$, ou seja, que o padrão é redundante, e o valor resultante da classificação realizada pelo Perceptron:

$$\epsilon_k = y_k - \hat{y}_k \tag{5.27}$$

$$\epsilon_k = \frac{\mathbf{w}^T\mathbf{x}_k - \text{crosstalk}}{||\mathbf{x}_k||^2} - \frac{\mathbf{w}^T\mathbf{x}_k}{|\mathbf{w}^T\mathbf{x}_k|} \tag{5.28}$$

$$\epsilon_k = \frac{(\mathbf{w}^T\mathbf{x}_k - \text{crosstalk})|\mathbf{w}^T\mathbf{x}_k| - \mathbf{w}^T\mathbf{x}_k||\mathbf{x}_k||^2}{||\mathbf{x}_k||^2|\mathbf{w}^T\mathbf{x}_k|}. \tag{5.29}$$

Deseja-se que $\epsilon_k = 0$, ou seja, que o rótulo y_k seja devidamente recuperado, logo:

$$(\mathbf{w}^T\mathbf{x}_k - \text{crosstalk})|\mathbf{w}^T\mathbf{x}_k| - \mathbf{w}^T\mathbf{x}_k||\mathbf{x}_k||^2 = 0. \tag{5.30}$$

O método de Aprendizado Ativo proposto considera que o valor de α é correspondente ao valor absoluto da margem do padrão analisado, ou seja, a margem do padrão $\mathbf{x}_k$ em relação ao hiperplano corrente, logo:

$$\alpha = |\mathbf{w}^T\mathbf{x}_k|. \tag{5.31}$$

Substituindo na equação 5.30 obtém-se:

$$(\mathbf{w}^T\mathbf{x}_k - \text{crosstalk})\alpha - \mathbf{w}^T\mathbf{x}_k||\mathbf{x}_k||^2 = 0. \tag{5.32}$$

Isolando o valor de α:

$$\alpha = \frac{\mathbf{w}^T\mathbf{x}_k||\mathbf{x}_k||^2}{(\mathbf{w}^T\mathbf{x}_k - \text{crosstalk})}. \tag{5.33}$$

Substituindo 5.33 em 5.14 tem-se:

$$t_{max} = \frac{(\beta + 2\theta)||\mathbf{w}||^2(\mathbf{w}^T\mathbf{x}_k - \text{crosstalk})^2}{(\mathbf{w}^T\mathbf{x}_k||\mathbf{x}_k||^2)^2}. \tag{5.34}$$

Como o vetor de pesos do Perceptron utilizado é normalizado, ou seja $||\mathbf{w}|| = 1$, a Equação 5.34 resume-se na Equação 5.35.

$$t_{max} = \frac{(\beta + 2\theta)(\mathbf{w}^T\mathbf{x}_k - \text{crosstalk})^2}{(\mathbf{w}^T\mathbf{x}_k||\mathbf{x}_k||^2)^2}. \tag{5.35}$$

Dessa forma, o valor de t_{max} foi obtido em função do *crosstalk* necessário para que o erro de classificação seja zero. Isso leva às seguintes conclusões:

- O controle do erro e do *crosstalk* estão implícitos no processo de escolha do padrão durante o Aprendizado Ativo, conforme pode ser verificado na Equação 5.35;
- Se a distância do padrão analisado ao hiperplano separador for muito grande e o *crosstalk* também, o valor de t_{max} poderá ser pequeno, dependo dos valores de β, θ e da norma do padrão. Dessa forma, o padrão terá uma maior probabilidade de ser rejeitado, dependendo do número de padrões utilizados previamente para treinamento do hiperplano corrente;
- Se a distância for pequena, mas o *crosstalk* for grande, o valor de t_{max} poderá ser grande. Dessa forma, o padrão terá maior probabilidade de ser escolhido para ser inserido no processo de aprendizado, dependendo

do número de padrões utilizados previamente para treinamento do hiperplano corrente;

- Se a distância for grande e o *crosstalk* pequeno, t_{max} poderá ser pequeno. Dessa forma, o padrão terá uma maior probabilidade de ser rejeitado, dependendo do número de padrões utilizados previamente para treinamento do hiperplano corrente;

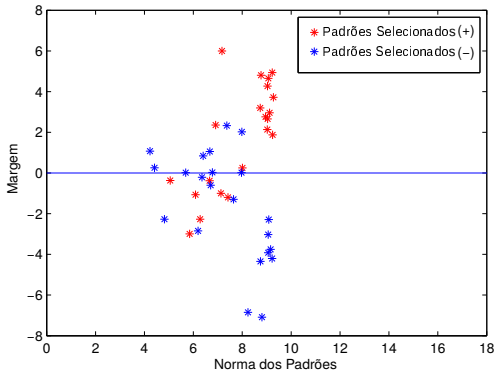
- Se a distância e o *crosstalk* tiverem valores próximos, então t_{max} poderá ser pequeno. Dessa forma, o padrão terá uma maior probabilidade de ser rejeitado, dependendo do número de padrões utilizados previamente para treinamento do hiperplano corrente;

- Os valores de β, θ e do quadrado da norma do padrão analisado ($||\mathbf{x}_k||^2$) completam a equação 5.35 fornecendo informações indiretas da distribuição dos dados, uma vez que: β corresponde ao valor máximo do quadrado da norma dos padrões de treinamento, o que corresponde ao quadrado da distância do padrão mais afastado da origem do sistema de coordenadas; θ corresponde ao valor máximo entre o α do padrão corrente e o maior α entre os padrões de treinamento, ou seja, corresponde à distância do padrão mais afastado do hiperplano corrente. $||\mathbf{x}_k||^2$ corresponde ao quadrado da distância do padrão analisado em relação à origem do sistema de coordenadas.

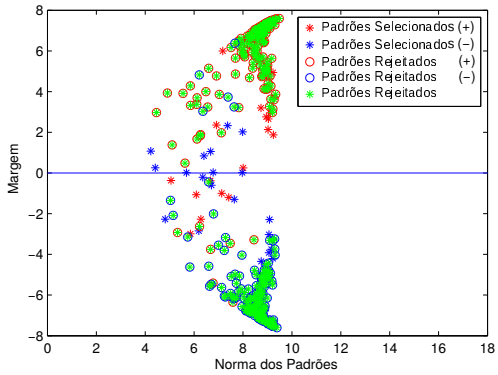
Essas conclusões explicam o porquê do método minimizar o erro controlando o *crosstalk*. Dessa forma, são evidenciados os motivos que levam a um padrão ser rejeitado ou não, pois de acordo com o número de padrões já treinados, a distância do padrão analisado e o seu *crosstalk* ele terá uma probabilidade maior ou menor de ser rejeitado. Os valores de β e θ complementam a equação, pois fornecem informações da distribuição espacial dos dados, sendo que essa distribuição também está intimamente ligada com o *crosstalk*, conforme discutido no capítulo anterior.

O cálculo de t_{max} utiliza informações de norma e margem extraídas do conjunto de treinamento. Dessa forma, é interessante observar a relação entre esses duas grandezas de forma gráfica. Para tanto, foi realizado um experimento utilizando o problema da Figura 3.2. Foram gerados classificadores utilizando os métodos EALMSS e EALMPB apresentados na Seção 5.2. A camada escondida ELM continha 100 neurônios. Os resultados apresentados a seguir consistem em uma execução, tendo apenas o objetivo de possibilitar uma análise qualitativa dos padrões escolhidos no Aprendizado Ativo. O mesmo comportamento apresentado ocorreu em outras execuções do mesmo experimento. As Figuras 5.7(a) e 5.8(a) apresentam gráficos que relacionam a

norma e a margem de cada padrão utilizado no treinamento para os métodos EALMSS e EALMPB, respectivamente. As Figuras 5.7(b) e 5.8(b) apresentam a norma e margem tanto dos padrões selecionados quanto daqueles rejeitados durante o processo de aprendizado dos métodos EALMSS e EALMPB, respectivamente.

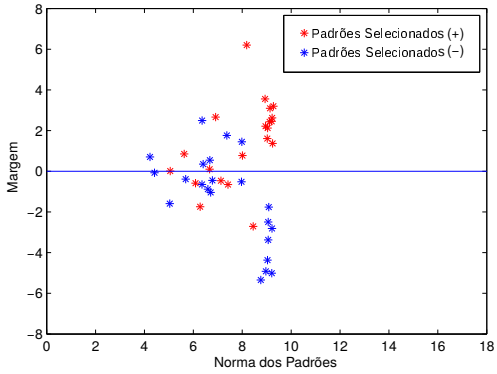


(a) Norma x Margem - Padrões Selecionados (EALMSS)

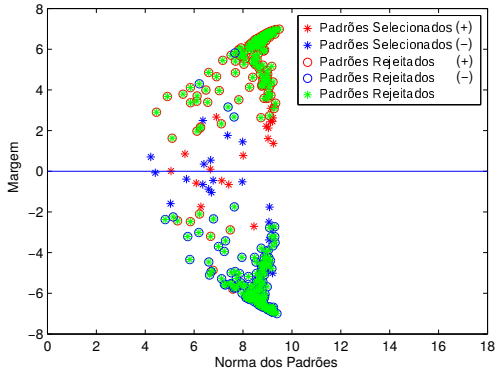


(b) Norma x Margem - Padrões Selecionados e Rejeitados (EALMSS)

Figura 5.7: Relação entre Norma e Margem dos padrões selecionados (5.7(a)) e dos padrões rejeitados 5.7(b) para o método EALMSS



(a) Norma x Margem - Padrões Selecionados (EALMPB)



(b) Norma x Margem - Padrões Selecionados e Rejeitados (EALMPB)

Figura 5.8: Relação entre Norma e Margem dos padrões selecionados (5.7(a)) e dos padrões rejeitados (5.7(b)) para o método EALMPB

Como pode ser observado, o método de Aprendizado Ativo seleciona padrões capazes de representar a distribuição dos dados no espaço, tanto na versão EALMSS quanto na EALMPB. Essa representação é obtida utilizando a margem dos padrões em relação ao hiperplano separador e a norma do vetor de dados, que representa a distância do padrão em relação à origem do sistema de coordenadas. O método seleciona: alguns poucos padrões que representam os dados mais distantes do hiperplano; seleciona a maioria dos padrões próximos ao hiperplano separador (região entre -2 e 2 no eixo y para

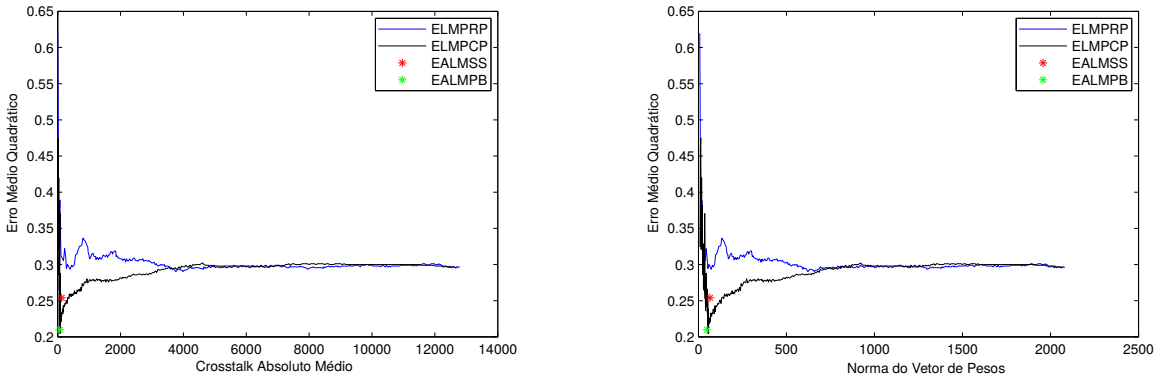
os dois algoritmos), que certamente são os padrões mais informativos, uma vez que a incerteza em relação ao rótulo dos mesmos é maior; e seleciona padrões que também fornecem informações da distância dos dados em relação à origem. Esse comportamento é devido ao cálculo de t_{max}, pois o mesmo leva em consideração a distância do padrão analisado em relação ao hiperplano separador (α), a distância do padrão mais distante da origem (β) e a distância do padrão mais distante do hiperplano separador (θ). Isso garante que um número mínimo de padrões seja selecionado para representar de forma suficiente a distribuição dos padrões no espaço, fornecendo, portanto, um modelo sem redundância nos dados e com boa capacidade de generalização, por conter informação da distribuição espacial dos dados.

De certa forma as Figuras 5.7(a) e 5.8(a) mostram que os dois algoritmos de Aprendizado Ativo conseguem extrair o "esqueleto" da distribuição dos dados, conforme pode ser verificado nas Figuras 5.7(b) e 5.8(b). Nessas figuras os padrões rejeitados são representados por um asterisco verde com bordas circulares que representam a classe do padrão rejeitado, sendo a borda vermelha referente aos padrões da classe positiva e a borda azul referente aos padrões da classe negativa. Esse comportamento dos algoritmos é extremamente vantajoso para o Aprendizado Ativo, pois evita que recursos importantes sejam gastos na rotulação de padrões redundantes. Contribui ainda para encontrar um modelo com capacidade suficiente para generalizar bem, conforme será discutido na próxima seção.

Outra observação importante é que o modelo EALMSS tem comportamento similar ao EALMPB, porém com custo computacional bem menor, uma vez que não precisa analisar a distância de todos os padrões disponíveis, analisando apenas um padrão por vez, podendo ser utilizado também com conjuntos de dados de tamanho fixo (*pool*) tendo que analisar o conjunto por inteiro apenas uma única vez. No Capítulo 6 os dois algoritmos serão comparados em problemas reais e essa conclusão será aprofundada.

5.4 Complexidade do Modelo Extreme Active Learning Machine e Sua Capacidade de Generalização

Com o objetivo de verificar a evolução da complexidade neural do modelo durante o processo de treinamento, o problema da Figura 3.2 foi utilizado em um experimento que consiste no treinamento de uma rede com uma camada escondida ELM com 100 neurônios e um Perceptron Hebbiano com pesos normalizados na saída. Essa rede foi treinada de quatro formas diferentes, a



(a) Crosstalk Absoluto Médio X Erro Médio Quadrático

(b) Norma Média X Erro Médio Quadrático

Figura 5.9: Evolução do erro médio quadrático em função de duas medidas de complexidade

saber: utilizando a cada iteração os padrões mais próximos do hiperplano separador (ELMPCP); utilizando padrões escolhidos aleatoriamente (ELMPRP); e utilizando os algoritmos EALMSS e EALMPB.

O experimento foi executado 10 vezes e foram coletados os valores médios do erro médio quadrático calculado sobre todos os dados disponíveis, o valor médio da soma do valor absoluto do *crosstalk* calculado utilizando apenas os padrões efetivamente utilizados para treinamento (padrões rotulados) e o valor médio da norma do vetor de pesos. Utilizando os valores coletados foram geradas as Figuras 5.9(a) e 5.9(b). A Figura 5.9(a) apresenta a relação entre erro médio quadrático e o valor médio da soma dos *crosstalks* absolutos. A Figura 5.9(b) apresenta a relação entre o erro médio quadrático e a norma do vetor de pesos.

A norma do vetor de pesos foi utilizada como medida de complexidade do modelo uma vez que a classificação gerada por esse vetor e pelo vetor normalizado é exatamente a mesma. A normalização consiste simplesmente na divisão do vetor por um número positivo. A norma do vetor de pesos é considerada uma boa medida de complexidade para modelos neurais e já foi amplamente estudada em diversos trabalhos [6, 69, 8, 67, 41, 46, 68, 13, 12]. Ela está relacionada com a capacidade de generalização de um modelo neural, sendo que quanto mais complexo o modelo, maior é a norma e maior é a chance do modelo sofrer com sobreajuste [69].

Como pode ser observado nas Figuras 5.9(a) e 5.9(b) a evolução do erro médio quadrático em relação ao *crosstalk* médio absoluto e à norma média do vetor de pesos tem evolução similar, indicando que o *crosstalk* é uma medida de complexidade similar à norma do vetor de pesos. Esse resultado mostra que encontrar uma solução com boa relação custo-benefício entre o erro médio quadrático e o *crosstalk* tem o mesmo efeito que encontrar a solução de

melhor relação entre erro médio quadrático e norma do vetor de pesos. Como pode ser observado nas duas figuras, a heurística de Aprendizado Ativo de seleção do padrão mais próximo do hiperplano separador tem o comportamento de encontrar soluções com menor erro médio quadrático em modelos com complexidade menor (*crosstalk* e norma) quando comparado com o modelo passivo de aprendizado que utiliza dados escolhidos aleatoriamente.

Também é possível observar que o algoritmo EALMPB possui uma condição de parada capaz de encontrar exatamente a solução com melhor relação custo benefício entre o erro médio quadrático e a complexidade do modelo, uma vez que encontra a solução com menor erro e com baixa complexidade. Observa-se ainda que a solução EALMSS encontra uma solução intermediária, porém que também possui uma boa relação custo-benefício. Entretanto essa última encontra tal solução com um custo computacional muito menor que o modelo EALMPB.

Observando a Figura 5.6 é possível verificar que de fato a solução encontrada pelo modelo EALMPB tem uma capacidade de generalização ligeiramente melhor que a solução EALMSS, o que condiz com as Figuras 5.9(a) e 5.9(b). Mostra também que as soluções com boa relação entre erro médio quadrático e complexidade têm melhor capacidade de generalização. No Capítulo 6 essa capacidade de encontrar uma solução com boa capacidade de generalização será verificada para bases de dados de problemas reais e confirmarão as conclusões apresentadas nesta seção.

5.5 Conclusões do Capítulo

Neste capítulo foram apresentadas duas estratégias de Aprendizado Ativo que são baseadas em um simples teste de convergência. Ao longo do capítulo foi demonstrado que o Aprendizado Ativo é capaz de controlar a evolução do *crosstalk* durante o processo de aprendizado e que esse termo age como um regularizador. Além disso, foi verificada a influência da distribuição espacial dos dados e do *crosstalk* no cálculo do teste de convergência. Foi discutido como o teste é capaz de eliminar os padrões redundantes e de encontrar um vetor de pesos calculado a partir do "esqueleto" da distribuição espacial dos dados.

Por fim, foi feita uma análise da relação entre erro e complexidade dos modelos obtidos pelos dois algoritmos de Aprendizado Ativo apresentados neste capítulo. Foi verificado que os dois algoritmos são capazes de encontrar uma solução com boa relação entre erro e complexidade o que explica o porquê dessas soluções terem boa capacidade de generalização utilizando para aprendizado um conjunto reduzido de padrões. Verificou-se que o algo-

ritmo EALMSS, cujo aprendizado é feito por um fluxo contínuo de dados, tem comportamento semelhante ao modelo EALMPB, cujo aprendizado é computacionalmente muito mais custoso, pois a cada iteração a distância de todos os padrões do conjunto de dados não rotulados deve ser calculada em relação ao hiperplano corrente.

No próximo capítulo serão apresentados resultados experimentais para problemas reais e os algoritmos de Aprendizado Ativo serão comparados com outras estratégias apresentadas na literatura.

CAPÍTULO 6

Experimentos e Resultados

Neste capítulo serão apresentados os resultados para quatro experimentos. O experimento descrito na Seção 6.1 apresenta as limitações das ELMs quando o número de padrões de treinamento é próximo do número de neurônios da camada escondida. Também são apresentadas as vantagens da utilização do aprendizado Hebbiano com pesos normalizados [21] no lugar da solução da pseudoinversa. O experimento descrito na Seção 6.2 complementa o experimento da Seção 6.1 ao apresentar a relação de custo-benefício das soluções dos métodos EALMSS e EALMPB em termos do erro médio quadrático de treinamento e da complexidade do modelo, sendo essa última representada tanto pelo *crosstalk* médio absoluto quanto pela norma média dos pesos do Perceptron não normalizado.

O experimento descrito na Seção 6.3 mostra que o número de neurônios escondidos não precisa ser finamente ajustado se esse valor for muito maior que o número de características do espaço de entrada. Na Seção 6.4 os algoritmos de aprendizado ativo apresentados nesta tese são comparados com outros métodos de aprendizado ativo conhecidos na literatura, com uma SVM linear[1] e com uma ELM regularizada. Todos esses 4 experimentos foram realizados utilizando bases de dados oriundas do repositório UCI [22]. As características dessas bases de dados estão listadas na Tabela 6.1.

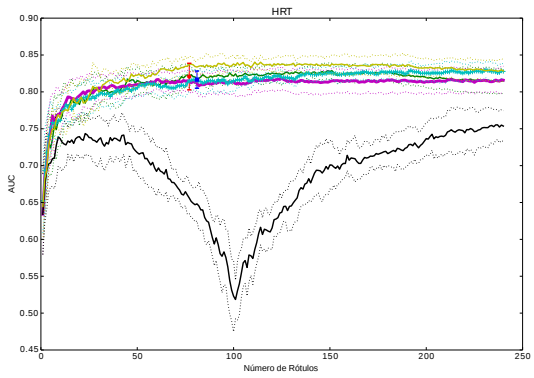
[1]Todas as implementações das SVMs lineares foram realizadas utilizando a biblioteca LibLinear [20] disponível para diversas linguagens de programação no pacote Shogun [64].

Tabela 6.1: Conjuntos de Dados Utilizados

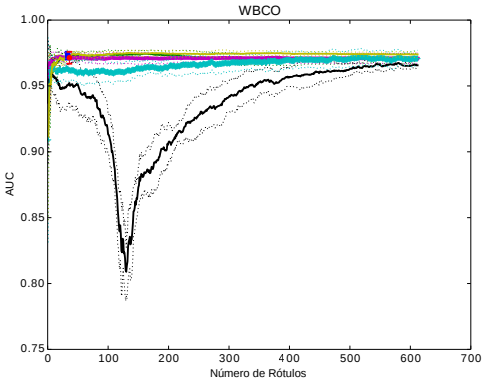
Nome	Sigla	No. Padrões	No. Entradas	No. Classes
Heart Disease	HRT	297	13	2
Wisc. Breast Cancer Original	WBCO	699	9	2
Wisc. Breast Cancer Diagnostic	WBCD	569	31	2
Pima Diabetes	PIMA	768	8	2
Sonar	SNR	208	60	2
Ionosphere	ION	351	34	2
Australian Credit	AUST	690	14	2
Liver Disorder	LIV	345	6	2
German Credit	GER	1000	24	2
Spam	SPAM	4601	58	2

6.1 Experimento: Limitações das ELMs para o Aprendizado Ativo

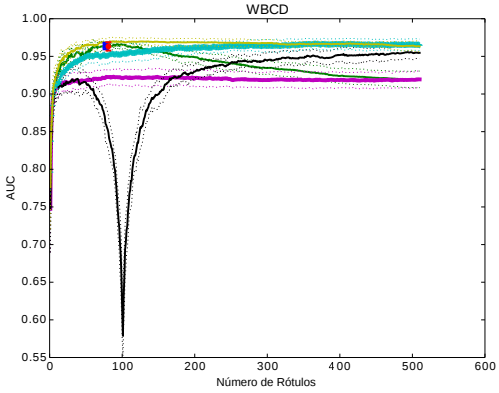
Os seguintes modelos foram comparados no primeiro experimento: ELM original com treinamento utilizando o método da pseudoinversa (Original ELM) [39]; modelo composto por uma camada escondida ELM e um Perceptron na saída treinado com aprendizado Hebbiano e pesos normalizados [21] (ELMP); modelos treinados utilizando a estratégia de aprendizado ativo apresentada nesta tese nas versões *Stream-based Selective Sampling* (*Extreme Active Learning Machine Selective Sampling - EALMSS*) e *Pool-based Sampling* (*Extreme Active Learning Machine Pool-based - EALMPB*); ELM regularizada (ELM2012) [40]; e o modelo AL-ELM [74]. Os dados foram apresentados de forma aleatória para todos os modelos ELM, ELM2012 e EALMSS. No caso particular do modelo ELMP, dois testes foram realizados: (i) aprendizado utilizando dados apresentados aleatoriamente em cada iteração (*ELMP Random Patterns* - ELMPRP); e (ii) aprendizado utilizando o padrão mais próximo do hiperplano separador em cada iteração, como proposto por Schohn e Cohn [58] (*ELMP Closest Patterns* - ELMPCP). Para todos os modelos a camada escondida ELM utilizada foi a mesma, possuindo 100 neurônios escondidos. Esse número foi escolhido para demonstrar as limitações das ELMs quando o número de padrões de treinamento é próximo do número de neurônios escondidos. Para os modelos ELM2012 e AL-ELM trinta porcento do conjunto de treinamento foi rotulado e utilizado para realizar o ajuste fino do parâmetro de regularização.



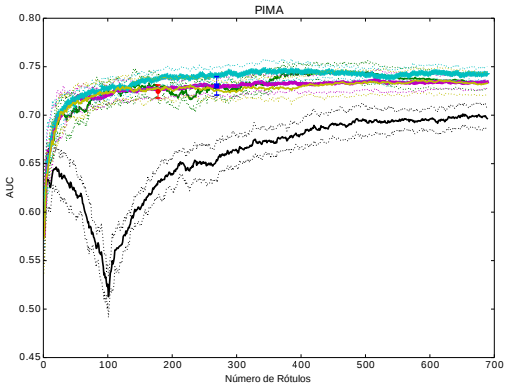
(a) HRT



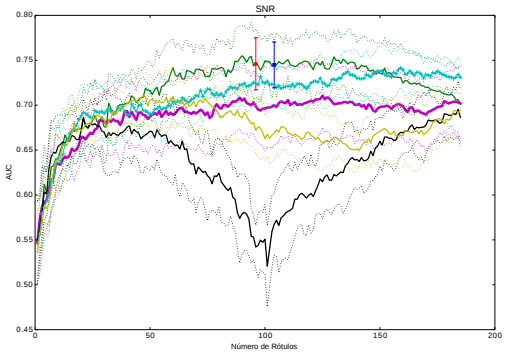
(b) WBCO



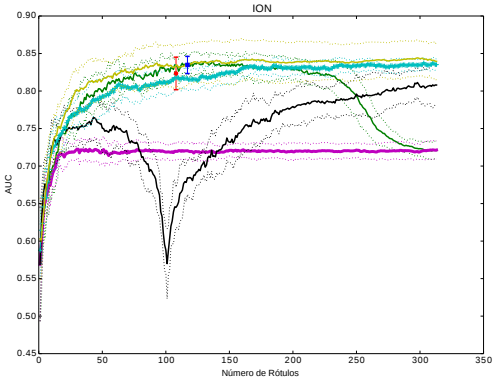
(c) WBCD



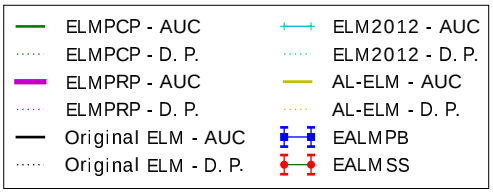
(d) PIMA



(e) SNR

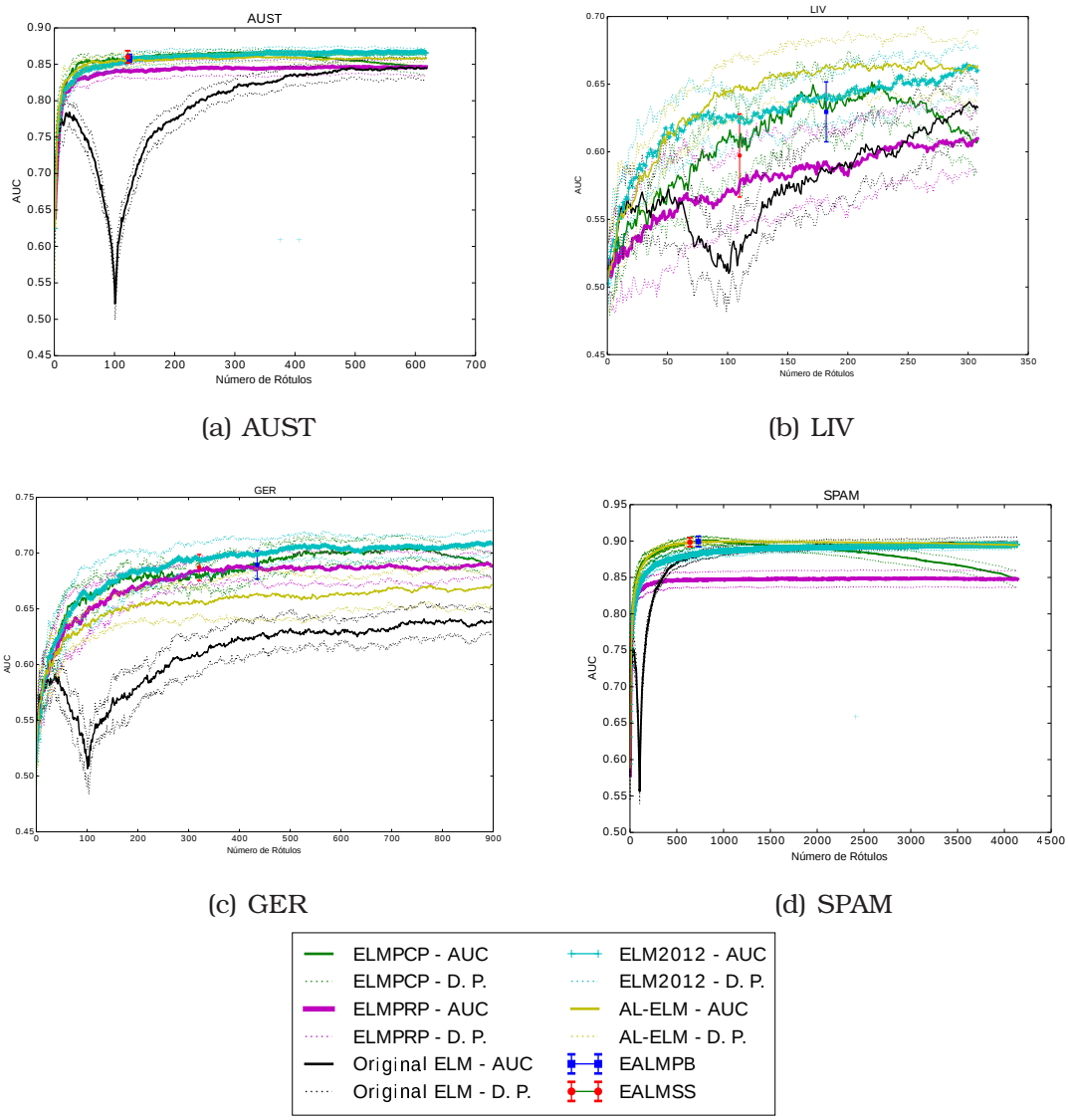


(f) ION



(g) Legenda

Figura 6.1: Resultados médios de 10 execuções da validação cruzada do tipo *10-fold* para diferentes métodos ELM aplicados nas bases de dados: HRT, WBCO, WBCD, PIMA, SNR and ION



(a) AUST (b) LIV

(c) GER (d) SPAM

(e) Legenda

Figura 6.2: Resultados médios de 10 execuções da validação cruzada do tipo *10-fold* para diferentes métodos ELM aplicados nas bases de dados: AUST, LIV, GER, SPAM

O desempenho médio de cada algoritmo em termos de AUC foi calculada após 10 execuções da validação cruzada do tipo *10-fold*. Os resultados são apresentados nas Figuras 6.1 e 6.2. Como esperado, a formulação original das ELMs não apresentou boa generalização quando o tamanho do conjunto de treinamento ficou próximo do número de neurônios da camada escondida, uma vez que a saída da ELM é calculada pela solução de um sistema de equações lineares [39] utilizando o cálculo da pseudoinversa. Como pode ser observado, a estratégia de aprendizado ativo obtém desempenho similar aos melhores classificadores que podem ser obtidos com o método ELMPCP que é treinado utilizando os padrões mais próximos do hiperplano separador. Isso mostra que o algoritmo EALMSS funciona como um filtro "baseado em margem", mas sem a necessidade de calcular a distância de todos os padrões em

relação ao hiperplano separador em cada iteração, como feito pelo algoritmo EALMPB.

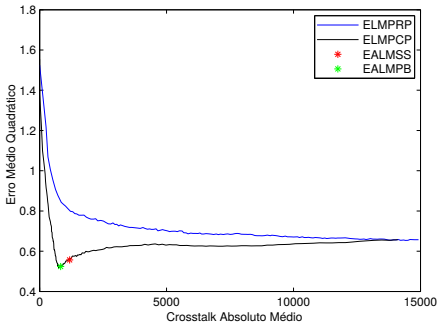
Como pode ser verificado, a versão regulariza das ELMs (ELM2012) e sua versão treinada com aprendizado ativo (AL-ELM) solucionam a limitação da pseudoinversa ao realizar o ajuste do parâmetro de regularização. Entretanto, uma parte do conjunto de dados teve que ser rotulada para esse fim, o que torna seu uso pouco prático no contexto do aprendizado ativo. Além disso, as soluções EALMSS e EALMPB são muito próximas das melhores soluções que podem ser obtidas com os métodos ELM2012 e AL-ELM, com a vantagem de não serem sensíveis ao ajuste fino de parâmetros e de possuírem uma condição de parada bem definida, que leva a soluções com boa capacidade de generalização. O modelo EALMSS obtém essas soluções com baixo custo computacional, uma vez que não precisa calcular a distância de todos os padrões em relação ao separador e não realiza nenhum tipo de inversão de matriz.

6.2 Experimento: Complexidade do Modelo Neural Obtido Pelo Aprendizado Ativo

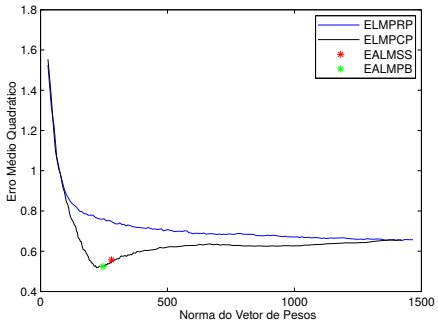
Para avaliar a evolução da complexidade dos modelos neurais treinados com o aprendizado Hebbiano foram utilizados os mesmos experimentos de validação cruzada da seção anterior, sendo coletadas a média do erro médio quadrático calculado utilizando todos os dados disponíveis para treinamento (os nove *folds* em cada execução da validação cruzada), o *crosstalk* absoluto médio calculado utilizando os padrões efetivamente aprendidos em cada iteração e a norma média do vetor de pesos do Perceptron sem normalização. Assim como o experimento anterior todos os resultados são oriundos da média de 10 execuções da validação cruzada. O experimento foi executado para todas as bases da tabela 6.1.

Os resultados são apresentados nas Figuras 6.3, 6.4 e 6.5. Como pode ser observado, em todos os casos os métodos EALMSS e EALMPB tendem a encontrar uma boa relação custo-benefício em termos de erro médio quadrático e complexidade, que pode ser medida tanto como *crosstalk* quanto como norma do vetor de pesos. Esses resultados corroboram com os resultados apresentados nas Figuras 6.1 e 6.2 da seção anterior, uma vez que elas demonstram que para quase todas as bases utilizadas as soluções EALMSS e EALMPB possuem boa capacidade de generalização, ou seja, classificam bem dados desconhecidos. Vale ressaltar que as Figuras 6.3, 6.4 e 6.5 foram geradas utilizando apenas os dados de treinamento. Isso confirma que soluções com boa relação custo-benefício entre erro de treinamento e complexidade do modelo tendem a ser soluções que generalizam bem, conforme relatado em

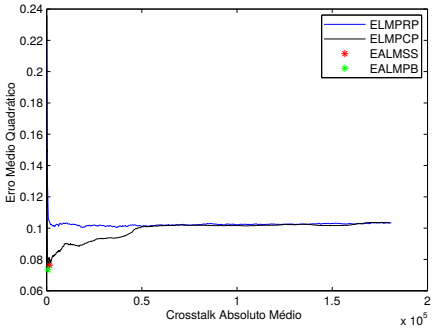
diversos trabalhos [6, 69, 8, 67, 41, 46, 68, 13, 12].



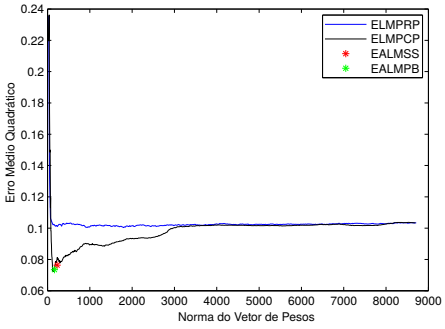
(a) Crosstalk Absoluto Médio X Erro Médio Quadrático - HRT



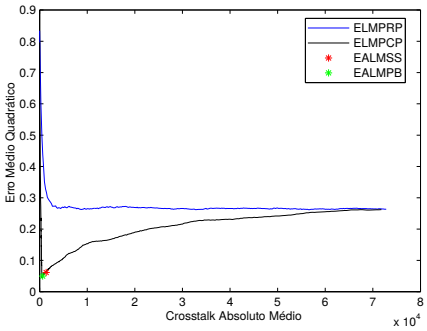
(b) Norma Média X Erro Médio Quadrático - HRT



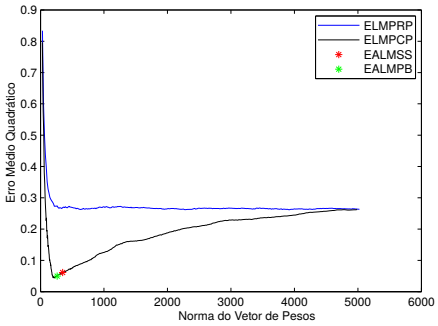
(c) Crosstalk Absoluto Médio X Erro Médio Quadrático - WBCO



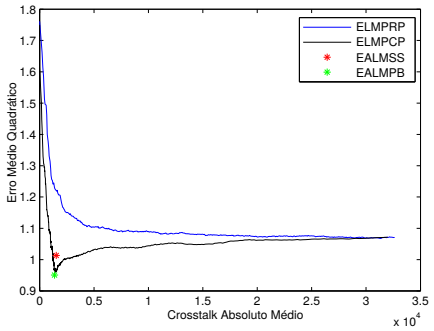
(d) Norma Média X Erro Médio Quadrático - WBCO



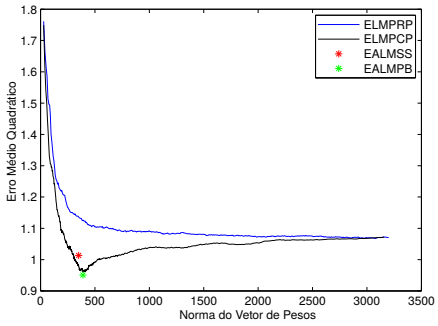
(e) Crosstalk Absoluto Médio X Erro Médio Quadrático - WBCD



(f) Norma Média X Erro Médio Quadrático - WBCD

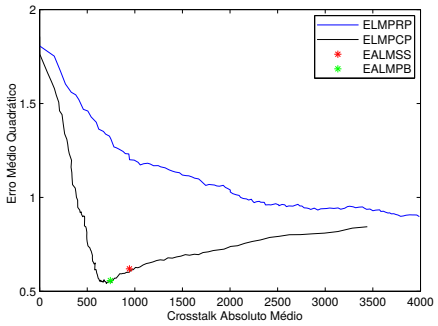


(g) Crosstalk Absoluto Médio X Erro Médio Quadrático - PIMA

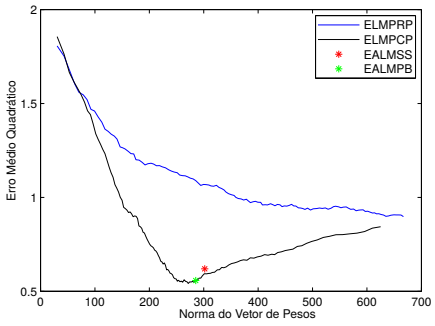


(h) Norma Média X Erro Médio Quadrático - PIMA

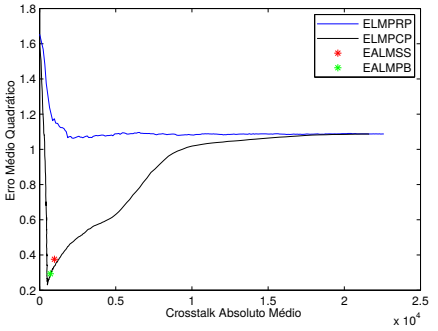
Figura 6.3: Evolução do erro médio quadrático em função de duas medidas de complexidade no conjuntos de dados: HRT, WBCO, WBCD, PIMA



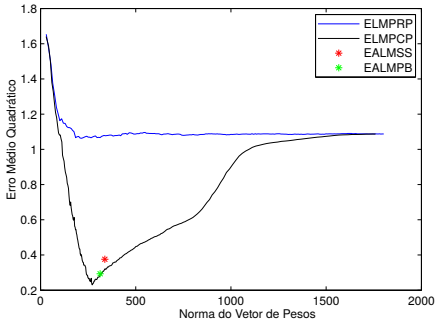
(a) Crosstalk Absoluto Médio X Erro Médio Quadrático - SNR



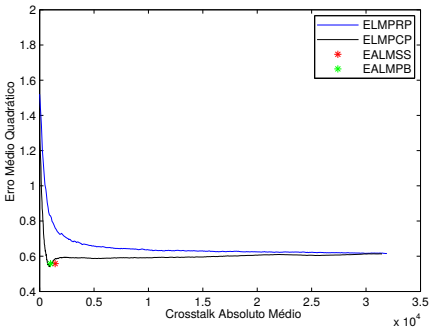
(b) Norma Média X Erro Médio Quadrático - SNR



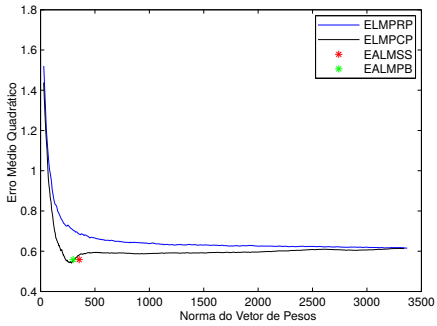
(c) Crosstalk Absoluto Médio X Erro Médio Quadrático - ION



(d) Norma Média X Erro Médio Quadrático - ION

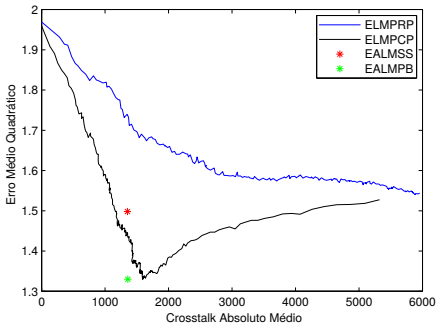


(e) Crosstalk Absoluto Médio X Erro Médio Quadrático - AUST

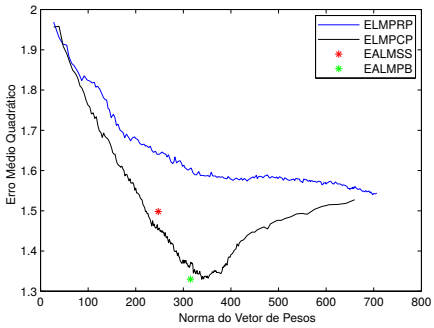


(f) Norma Média X Erro Médio Quadrático - AUST

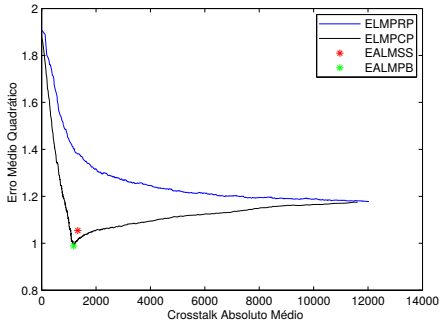
Figura 6.4: Evolução do erro médio quadrático em função de duas medidas de complexidade no conjuntos de dados: SNR, ION e AUST



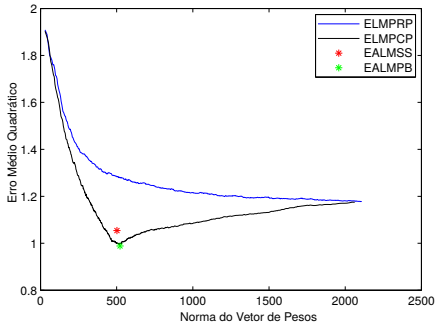
(a) Crosstalk Absoluto Médio X Erro Médio Quadrático - LIV



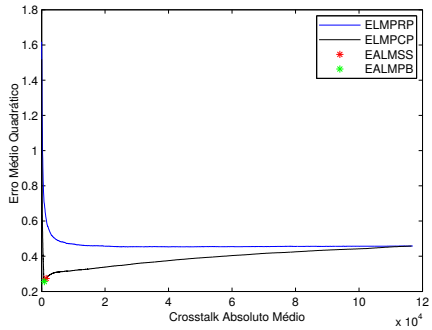
(b) Norma Média X Erro Médio Quadrático - LIV



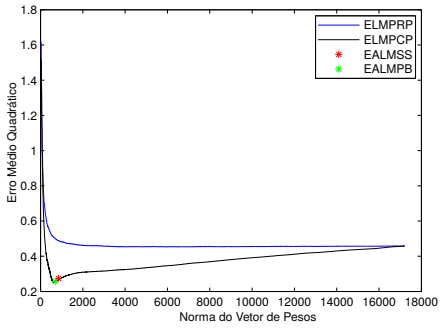
(c) Crosstalk Absoluto Médio X Erro Médio Quadrático - GER



(d) Norma Média X Erro Médio Quadrático - GER



(e) Crosstalk Absoluto Médio X Erro Médio Quadrático - SPAM



(f) Norma Média X Erro Médio Quadrático - SPAM

Figura 6.5: Evolução do erro médio quadrático em função de duas medidas de complexidade no conjuntos de dados: LIV, GER, SPAM

Certamente esse resultado foi alcançado pela capacidade do aprendizado ativo selecionar para aprendizado apenas os padrões mais informativos, impedindo que padrões redundantes sejam utilizados. Isso reduz o *crosstalk* e a norma, ou seja, a complexidade do modelo não precisa ser aumentada para absorver os dados redundantes, uma vez que a informação contida nesses dados provavelmente está presente nos dados previamente utilizados. Essa é a capacidade de extrair o "esqueleto" dos dados, conforme discutido na Seção 5.3 do capítulo anterior.

Uma vantagem da abordagem ativa apresentada nesta tese é que a solução é encontrada realizando o treinamento apenas uma vez, enquanto outras

técnicas, como as multiobjetivo, necessitam gerar diversos classificadores e posteriormente realizar um procedimento de escolha de uma solução com boa relação custo-benefício. Uma comparação mais aprofundada entre os algoritmos propostos nesta tese e modelos multiobjetivo será realizada em trabalhos futuros.

6.3 Experimento: Influência do Número de Neurônios Escondidos no Resultado do Aprendizado Ativo

Neste experimento foram comparados classificadores treinados com os métodos EALMSS e EALMPB, mas que possuem diferentes tamanhos para a camada escondida, a saber: 100, 500, 1000, 2500, 5000 e 10000 neurônios. Foram feitas 10 execuções da validação cruzada do tipo *10-fold*. Os resultados são listados na Tabela 6.2 em termos do número médio de rótulos selecionados pelos métodos (Rótulos) e a AUC média (AUC). Os melhores valores estão marcados em negrito. A partir da Tabela 6.2 pode ser verificado que quando o número de neurônios escondidos é maior que 500, o desempenho de todos os modelos se torna similar. Isso demonstra que os métodos EALMSS e EALMPB têm pouca sensibilidade ao tamanho da camada escondida quando o número de neurônios é suficientemente grande, não sendo, portanto, necessário realizar o ajuste fino do número de neurônios escondidos. O uso de um número elevado de neurônios como 1000 é suficiente, conforme sugerido em [40, 23, 26]. Para comprovar essa conclusão um teste de significância estatística foi aplicado aos resultados da Tabela 6.2, conforme será apresentado a seguir.

6.3.1 Teste de Significância Estatística

A Tabela 6.2 apresenta a comparação entre 6 classificadores do tipo *Stream-based* aplicados em 10 bases de dados diferentes e a comparação entre 6 classificadores do tipo *Pool-based* nas mesmas 10 bases. Algum teste de significância estatística deve ser realizado para verificar de fato que o tamanho da camada escondida não afeta de maneira significativa nem a qualidade da classificação das diferentes redes nem a quantidade de rótulos selecionados. O teste deve ser capaz de responder se as seguintes hipóteses podem ser rejeitadas: (i) Os classificadores têm desempenho equivalente em termos de AUC independente do tamanho da camada escondida; (ii) Os classificadores utilizam um número equivalente de rótulos para o aprendizado.

Como o teste estatístico deve comparar vários classificadores treinados em várias bases diferentes os testes *Paired T-Test* [16] e de Wilcoxon [73] não são

Tabela 6.2: Comparação entre Diferentes Tamanhos da Camada Escondida ELM para os Métodos EALMSS e EALMPB

EALMSS						
	100 Neurônios		500 Neurônios		1000 Neurônios	
Key	Rótulos	AUC	Rótulos	AUC	Rótulos	AUC
HRT	76,92 ± 3,36	0,82 ± 0,02	80,09 ± 1,58	0,83 ± 0,01	78,17 ± 1,09	0,83 ± 0,01
WBCO	36,47 ± 2,82	0,97 ± 0,01	36,98 ± 0,93	0,97 ± 0,00	37,16 ± 0,70	0,97 ± 0,00
WBCD	81,28 ± 5,36	0,96 ± 0,01	76,71 ± 1,01	0,97 ± 0,00	**73,65 ± 0,97**	**0,97 ± 0,00**
PIMA	177,63 ± 8,64	0,72 ± 0,01	172,96 ± 3,42	0,71 ± 0,01	168,69 ± 2,38	0,72 ± 0,01
SNR	95,97 ± 4,19	0,75 ± 0,03	100,00 ± 1,34	0,78 ± 0,01	98,42 ± 1,74	0,75 ± 0,01
ION	107,83 ± 4,44	0,82 ± 0,02	102,05 ± 1,04	0,85 ± 0,01	103,59 ± 1,37	0,84 ± 0,01
AUST	122,04 ± 5,80	0,86 ± 0,01	**119,59 ± 2,30**	**0,86 ± 0,01**	124,88 ± 1,88	0,86 ± 0,01
LIV	**109,73 ± 6,23**	**0,60 ± 0,03**	107,20 ± 4,14	0,58 ± 0,03	109,84 ± 7,10	0,59 ± 0,04
GER	291,21 ± 4,64	0,63 ± 0,01	302,07 ± 3,09	0,67 ± 0,01	315,55 ± 3,11	0,68 ± 0,01
SPAM	675,86 ± 4,33	0,89 ± 0,00	597,17 ± 4,89	0,92 ± 0,00	557,18 ± 4,68	0,92 ± 0,00
	2500 Neurônios		5000 Neurônios		10000 Neurônios	
HRT	76,94 ± 1,96	0,83 ± 0,01	76,53 ± 1,75	0,83 ± 0,01	**76,34 ± 1,11**	**0,84 ± 0,01**
WBCO	36,90 ± 0,50	0,97 ± 0,00	36,70 ± 1,07	0,97 ± 0,00	**36,12 ± 0,87**	**0,97 ± 0,00**
WBCD	75,44 ± 1,14	0,97 ± 0,00	75,35 ± 1,15	0,97 ± 0,00	75,76 ± 0,95	0,97 ± 0,01
PIMA	**168,63 ± 2,73**	**0,72 ± 0,01**	171,63 ± 3,59	0,73 ± 0,01	172,43 ± 2,83	0,71 ± 0,01
SNR	**97,54 ± 1,53**	**0,78 ± 0,02**	98,08 ± 1,49	0,78 ± 0,02	97,75 ± 2,40	0,77 ± 0,02
ION	**101,56 ± 1,65**	**0,85 ± 0,01**	101,72 ± 1,67	0,85 ± 0,01	102,97 ± 0,78	0,85 ± 0,01
AUST	119,91 ± 4,19	0,86 ± 0,01	120,75 ± 3,46	0,86 ± 0,00	121,18 ± 3,17	0,86 ± 0,01
LIV	109,22 ± 6,94	0,59 ± 0,03	106,53 ± 5,12	0,58 ± 0,03	105,84 ± 5,22	0,58 ± 0,02
GER	**313,04 ± 3,09**	**0,68 ± 0,01**	313,45 ± 3,11	0,68 ± 0,02	314,67 ± 2,90	0,68 ± 0,01
SPAM	559,81 ± 3,67	0,92 ± 0,00	561,52 ± 6,31	0,92 ± 0,00	**549,60 ± 2,95**	**0,92 ± 0,00**
EALMPB						
	100 Neurônios		500 Neurônios		1000 Neurônios	
Key	Rótulos	AUC	Rótulos	AUC	Rótulos	AUC
HRT	81,00 ± 5,32	0,82 ± 0,01	82,07 ± 2,04	0,84 ± 0,01	81,02 ± 1,52	0,83 ± 0,01
WBCO	**32,87 ± 2,41**	**0,97 ± 0,01**	34,08 ± 0,38	0,97 ± 0,00	34,23 ± 0,35	0,97 ± 0,00
WBCD	76,65 ± 4,56	0,96 ± 0,00	69,57 ± 0,60	0,97 ± 0,00	**69,08 ± 0,73**	**0,97 ± 0,00**
PIMA	268,91 ± 2,56	0,73 ± 0,01	234,11 ± 6,36	0,73 ± 0,01	**225,89 ± 3,91**	**0,73 ± 0,01**
SNR	104,46 ± 8,36	0,74 ± 0,03	105,09 ± 1,85	0,78 ± 0,02	105,11 ± 1,76	0,75 ± 0,02
ION	116,81 ± 6,17	0,83 ± 0,01	103,56 ± 1,05	0,85 ± 0,01	105,29 ± 1,60	0,84 ± 0,01
AUST	**124,67 ± 9,40**	**0,86 ± 0,01**	129,66 ± 3,23	0,86 ± 0,01	135,40 ± 3,55	0,86 ± 0,00
LIV	**182,36 ± 15,21**	**0,63 ± 0,02**	176,58 ± 7,85	0,62 ± 0,01	196,49 ± 8,46	0,63 ± 0,02
GER	386,13 ± 6,65	0,65 ± 0,01	374,76 ± 2,68	0,68 ± 0,01	393,72 ± 4,47	0,68 ± 0,01
SPAM	776,83 ± 9,13	0,90 ± 0,00	653,46 ± 6,00	0,92 ± 0,00	602,38 ± 4,77	0,92 ± 0,00
	2500 Neurônios		5000 Neurônios		10000 Neurônios	
HRT	80,37 ± 1,62	0,84 ± 0,01	**79,33 ± 1,41**	**0,84 ± 0,01**	80,61 ± 1,25	0,84 ± 0,01
WBCO	34,12 ± 0,62	0,97 ± 0,00	33,68 ± 0,90	0,97 ± 0,00	33,16 ± 0,27	0,97 ± 0,00
WBCD	70,17 ± 0,79	0,97 ± 0,00	69,70 ± 1,30	0,97 ± 0,00	70,48 ± 0,73	0,97 ± 0,00
PIMA	226,86 ± 3,22	0,72 ± 0,01	229,74 ± 4,38	0,73 ± 0,01	226,62 ± 5,44	0,73 ± 0,01
SNR	103,51 ± 1,84	0,78 ± 0,01	**102,28 ± 2,07**	**0,78 ± 0,02**	103,06 ± 2,44	0,78 ± 0,02
ION	**101,29 ± 1,37**	**0,85 ± 0,01**	102,68 ± 1,70	0,84 ± 0,01	102,65 ± 1,35	0,85 ± 0,00
AUST	128,57 ± 3,03	0,86 ± 0,00	128,47 ± 3,35	0,86 ± 0,01	130,06 ± 2,95	0,86 ± 0,00
LIV	189,73 ± 9,85	0,63 ± 0,03	191,71 ± 9,53	0,63 ± 0,02	185,87 ± 8,63	0,63 ± 0,02
GER	384,90 ± 5,32	0,69 ± 0,01	**383,86 ± 4,01**	**0,69 ± 0,01**	384,82 ± 3,56	0,69 ± 0,01
SPAM	603,97 ± 4,57	0,92 ± 0,00	607,25 ± 5,77	0,92 ± 0,00	**601,39 ± 5,84**	**0,92 ± 0,00**

adequados, pois devem ser utilizados para realizar a comparação entre apenas 2 classificadores. Demšar [18] recomenda que seja utilizado o teste de Friedman [24, 25] para realizar a comparação entre vários classificadores em várias bases de dados. Caso a hipótese nula seja rejeitada é necessário utilizar algum teste *post-hoc* para identificar para quais classificadores a hipótese nula é rejeitada. Demšar [18] sugere o uso do teste de Bonferroni-Dunn [19] para comparações do tipo "um-contra-todos" ou o teste de Nemenyi [50] para comparações do tipo "um-contra-um". Esses testes têm a vantagem de serem não-paramétricos e não exigirem que os dados analisados tenham distribuição normal.

Considerando que devem ser comparados L algoritmos aplicados em M bases de dados e assumindo como hipótese nula a possibilidade de que todos

os classificadores são equivalentes, o teste de Friedman pode ser calculado a partir das Equações 6.1 e 6.2 [9]:

$$F_F = \frac{(M-1)\chi_F^2}{M(L-1)-\chi_F^2} \tag{6.1}$$

$$\chi_F^2 = \frac{12M}{L(L+1)}(\sum_{t=1}^{L} R_t^2 - \frac{L(L+1)^2}{4}) \tag{6.2}$$

onde R_t, $t = 1,...,L$, consiste nos *ranks* médios para os algoritmos utilizados. F_F é distribuída de acordo com a distribuição F possuindo $L-1$ e $(L-1)(M-1)$ graus de liberdade [9].

O teste de Friedman consiste em comparar o valor de F_F com um valor crítico (C) obtido de uma tabela de valores críticos para a distribuição F. Se $F_F > C$ então a hipótese nula pode ser rejeitada e se conclui que os algoritmos comparados não são equivalentes. Caso a hipótese nula seja rejeitada deve ser utilizado um teste *post-hoc* para verificar em quais comparações a hipótese é rejeitada.

Para os experimentos realizados nesta seção foram testados 6 modelos e 10 bases de dados, logo $L = 6$ e $M = 10$, tanto para o modelo EALMSS quanto para o modelo EALMPB. Nessa situação os graus de liberdade para os dois modelos são $L - 1 = 5$ e $(L - 1)(M - 1) = 45$. Considerando um nível de significância de 5% e os graus de liberdade apresentados, o valor crítico para os dois modelos é $C = 2{,}4221$. Dessa forma, a hipótese nula será rejeitada se $F_F > 2{,}4221$, sendo esse valor calculado para cada modelo e para cada experimento, ou seja, em termos de AUC e do número de rótulos selecionados.

Para o modelo EALMSS o valor de F_F calculado a partir da comparação do valor de AUC obtido por cada classificador em cada base de dados foi de $1{,}1810$. Isso significa que a hipótese de que os classificadores são equivalentes em termos de AUC não pode ser rejeitada. Para o mesmo modelo o valor de F_F calculado a partir da comparação do número de rótulos utilizado por cada classificador em cada base de dados foi de $1{,}8771$. Como esse valor é menor que o valor crítico conclui-se que a hipótese de que os classificadores utilizam um número equivalente de rótulos não pode ser rejeitada.

O valor de F_F calculado a partir da comparação do valor de AUC para o modelo EALMPB é $1{,}8062$. Como esse valor é menor que o valor crítico significa que a hipótese de que o número de neurônios não afeta o desempenho em termos de AUC não pode ser rejeitada. O valor calculado para a comparação do número de rótulos utilizados foi de $1{,}3482$ e novamente a hipótese de que o número de neurônios não influencia na quantidade de rótulos selecionados não pode ser rejeitada.

Como em todos os experimentos desta seção a hipótese nula não pôde ser

rejeitada, não há indícios estatísticos de que os classificadores não são equivalentes. Isso indica que a conclusão de que o número de neurônios escondidos não precisa ser finamente ajustado para os modelos EALMSS e EALMPB não pode ser rejeitada. Como foi discutido anteriormente, os resultados apresentados na Tabela 6.2 indicam que quando o número de neurônios escondidos é maior que 500, o desempenho de todos os modelos se torna similar.

6.4 Experimento: Comparação entre Métodos de Aprendizado Ativo

Neste experimento são comparados os métodos EALMSS e EALMPB com os seguintes métodos: Perceptron de Dasgupta *et al.* (PDKCM) [15], Perceptron de Cesa-Bianchi *et al.* (PCBGZ) [10]; SVM de Tong e Koller (SVMTK) [70], uma SVM linear treinada com todos os padrões de treinamento (SVMALL) e a versão regularizada das ELMs treinada com todos os padrões de treinamento (ELM2012). Todos esses métodos foram usados como saídas lineares para uma camada escondida ELM com 1000 neurônios e com pesos aleatoriamente selecionados na faixa de [-3, 3], como proposto por [23]. Aproximadamente trinta porcento de cada base de dados foi separada com o objetivo de ajustar os parâmetros livres dos métodos PDKCM, PCBGZ, ELM2012 e as SVMs. Isso foi feito utilizando a validação cruzada do tipo *10-fold*, como sugerido por Monteleoni *et al.* [49], porém a medida de desempenho escolhida foi a AUC. Para o modelo PCBGZ também foi testado o valor ótimo do parâmetro $b = (max_{\mathbf{x}\in C}||\mathrm{x}||^2)/2$ [10] e essa versão foi chamada de PCBGZ-OPT. O parâmetro de regularização C das SVMs foi escolhido a partir da faixa de valores $\{2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 1, 2, 2^2, ..., 2^{14}\}$, como sugerido em [21]. O parâmetro de regularização C do modelo ELM2012 foi escolhido a partir da faixa de valores $\{2^{-24}, 2^{-23}, ..., 2^{24}, 2^{25}\}$, como sugerido em [40]. Foram feitas dez execuções da validação cruzada do tipo *10-fold* com o objetivo de calcular a acurácia média (Ac), a AUC média (AUC) e o número médio de padrões selecionados pelo aprendizado ativo (Rótulos AA). Todas as bases de dados foram normalizadas para média 0 e desvio padrão 1. EALM, PDKCM, PCBGZ e PCBGZ-OPT foram inicializadas com um padrão selecionado aleatoriamente. SVMTK foi inicializado utilizando um conjunto composto por dois padrões, um de cada classe e escolhidos aleatoriamente, como proposto por [70].

Os modelos PDKCM, PCBGZ e PCBGZ-OPT foram adaptados para trabalharem tanto na versão *stream-based* (original) quanto *pool-based*, sendo que nessa versão são escolhidos para serem avaliados a cada iteração os padrões mais próximos do hiperplano corrente conforme definido nas heurísticas de Schon e Cohn [58] e Tong e Koller [70], com o objetivo de observar se existe

algum ganho em termos de número de rótulos selecionados ou em termos de AUC.

Para os métodos do tipo *stream-based* todos os dados não rotulados disponíveis foram avaliados apenas uma única vez. Nesse caso não há critério de parada e os algoritmos continuam avaliando os padrões à medida que eles chegam. A execução termina quando cada padrão da base de dados tiver sido avaliado. Para os métodos *pool-based* alguma condição de parada deve ser definida. Para as adaptações de PDKCM, PCBGZ e PCBGZ-OPT o critério de parada foi interromper o aprendizado depois que todos os padrões da base de dados fossem analisados. A única diferença para a versão *stream-based* foi que a cada execução o padrão mais próximo do hiperplano era avaliado, ao invés de avaliar um padrão escolhido ao acaso.

Para o modelo SVMTK foi utilizado o critério de parada proposto por Schon e Cohn [58]. O critério consiste em verificar se o padrão mais próximo do hiperplano tem distância menor que o vetor de suporte mais distante do separador, caso contrário considera-se que nenhuma informação nova é trazida pelos padrões disponíveis e o algoritmo termina, pois todos os candidatos a vetores de suporte teriam sido aprendidos. Para o algoritmo EALMPB o critério de parada foi o teste de convergência, conforme discutido no Capítulo 5.

Nesta seção o método AL-ELM [74] não pôde ser utilizado, pois o mesmo não possui condição de parada, o que limita o seu uso prático e restringe sua comparação com os outros métodos apresentados.

Os resultados do experimento estão apresentados na Tabela 6.3. Para cada modelo a quantidade de rótulos efetivamente utilizada para treinamento consiste naqueles utilizados para ajustar os parâmetros livres somados aos rótulos selecionados pela estratégia de aprendizado ativo escolhida. Dessa forma, a Tabela 6.3 mostra o número de rótulos obtidos pelo aprendizado ativo e o número de rótulos efetivamente utilizados (Rótulos Efetivos). Como pode ser observado, os melhores resultados foram obtidos pelos métodos EALMSS, EALMPB, SVMTK, SVMALL e ELM2012. Apesar do método SVMTK ter selecionado um número menor de rótulos durante o aprendizado ativo, seu custo computacional é muito maior que o do método EALMSS. Além disso, pode ser verificado que a quantidade de rótulos efetivos utilizados pelos métodos EALMSS e EALMPB são menores que aqueles utilizados por todos os outros métodos, uma vez que eles não exigem ajuste fino de parâmetros livres. A seguir serão apresentados resultados de testes de significância estatística para confirmar essas conclusões.

Tabela 6.3: Comparação entre diferentes métodos de Aprendizado Ativo e Passivo

	Stream-based Selective Sampling											
	EALMSS						PDKCM					
Sigla	Rótulos AA	%	Rótulos Efetivos	%	Ac	AUC	Rótulos AA	%	Rótulos Efetivos	%	Ac	AUC
HRT	88,58 ± 5,97	35,29	88,58	35,29	0,85 ± 0,01	0,84 ± 0,01	65,18 ± 1,43	25,97	146,18	58,24	0,77 ± 0,02	0,77 ± 0,02
WBCO	39,10 ± 2,79	6,16	39,10	6,16	0,98 ± 0,00	0,98 ± 0,00	63,46 ± 2,50	9,99	268,46	42,28	0,97 ± 0,00	0,97 ± 0,01
WBCD	80,07 ± 3,96	15,14	80,07	15,14	0,97 ± 0,01	0,97 ± 0,01	36,01 ± 0,87	6,81	207,01	39,13	0,93 ± 0,02	0,92 ± 0,02
PIMA	188,16 ± 6,18	26,32	188,16	26,32	0,76 ± 0,01	0,72 ± 0,01	131,02 ± 3,86	18,32	361,02	50,49	0,72 ± 0,02	0,70 ± 0,02
SNR	103,89 ± 2,88	53,83	103,89	53,83	0,71 ± 0,03	0,72 ± 0,03	77,01 ± 1,85	39,90	139,01	72,03	0,71 ± 0,03	0,72 ± 0,03
ION	105,30 ± 5,86	32,30	105,30	32,30	0,89 ± 0,01	0,87 ± 0,02	51,01 ± 1,77	15,65	156,01	47,86	0,83 ± 0,03	0,80 ± 0,03
AUST	137,92 ± 8,34	21,52	137,92	21,52	0,86 ± 0,01	0,86 ± 0,01	145,56 ± 3,18	22,71	352,56	55,00	0,83 ± 0,01	0,83 ± 0,01
LIV	163,54 ± 5,43	50,95	163,54	50,95	0,60 ± 0,02	0,60 ± 0,02	142,22 ± 3,56	44,31	246,22	76,70	0,61 ± 0,02	0,61 ± 0,02
GER	295,71 ± 8,26	31,80	295,71	31,80	0,75 ± 0,01	0,68 ± 0,01	247,85 ± 5,07	26,65	547,85	58,91	0,71 ± 0,01	0,64 ± 0,01
SPAM	515,48 ± 18,56	12,05	515,48	12,05	0,92 ± 0,00	0,91 ± 0,00	314,13 ± 6,07	7,34	1694,13	39,60	0,87 ± 0,01	0,87 ± 0,01
	PCBGZ						PCBGZ-OPT					
Sigla	Rótulos AA	%	Rótulos Efetivos	%	Ac	AUC	Rótulos AA	%	Rótulos Efetivos	%	Ac	AUC
HRT	50,83 ± 2,10	20,25	131,83	52,52	0,78 ± 0,02	0,78 ± 0,02	168,47 ± 0,36	67,12	249,47	99,39	0,77 ± 0,03	0,76 ± 0,03
WBCO	182,32 ± 2,98	28,71	387,32	61,00	0,97 ± 0,01	0,97 ± 0,01	424,06 ± 0,74	66,78	629,06	99,06	0,97 ± 0,01	0,96 ± 0,01
WBCD	203,67 ± 2,79	38,50	374,67	70,83	0,96 ± 0,01	0,96 ± 0,01	353,04 ± 0,54	66,74	524,04	99,06	0,96 ± 0,01	0,96 ± 0,01
PIMA	193,01 ± 4,35	26,99	423,01	59,16	0,71 ± 0,01	0,69 ± 0,02	480,08 ± 0,51	67,14	710,08	99,31	0,69 ± 0,02	0,67 ± 0,02
SNR	93,28 ± 1,77	48,33	155,28	80,46	0,71 ± 0,03	0,70 ± 0,02	130,46 ± 0,30	67,60	192,46	99,72	0,71 ± 0,03	0,70 ± 0,02
ION	116,28 ± 3,33	35,67	221,28	67,88	0,86 ± 0,01	0,83 ± 0,02	219,55 ± 0,33	67,35	324,55	99,56	0,86 ± 0,03	0,83 ± 0,03
AUST	170,40 ± 2,75	26,58	377,40	58,88	0,82 ± 0,02	0,81 ± 0,02	430,49 ± 0,61	67,16	637,49	99,45	0,79 ± 0,01	0,79 ± 0,01
LIV	156,05 ± 2,65	48,61	260,05	81,01	0,59 ± 0,03	0,59 ± 0,03	215,38 ± 0,29	67,10	319,38	99,50	0,59 ± 0,03	0,59 ± 0,03
GER	180,43 ± 3,37	19,40	480,43	51,66	0,70 ± 0,01	0,63 ± 0,01	624,64 ± 0,63	67,17	924,64	99,42	0,70 ± 0,01	0,64 ± 0,02
SPAM	1168,10 ± 12,43	27,30	2548,10	59,56	0,89 ± 0,01	0,89 ± 0,01	2860,56 ± 1,69	66,87	4240,56	99,12	0,89 ± 0,01	0,89 ± 0,01
	Pool-based Sampling											
	EALMPB						PDKCM					
Sigla	Rótulos AA	%	Rótulos Efetivos	%	Ac	AUC	Rótulos AA	%	Rótulos Efetivos	%	Ac	AUC
HRT	79,97 ± 3,80	31,86	79,97	31,86	0,84 ± 0,01	0,83 ± 0,01	21,76 ± 1,88	8,67	102,76	40,94	0,65 ± 0,05	0,65 ± 0,05
WBCO	33,23 ± 1,64	5,23	33,23	5,23	0,98 ± 0,00	0,98 ± 0,00	14,63 ± 9,89	2,30	219,63	34,59	0,91 ± 0,05	0,91 ± 0,04
WBCD	71,96 ± 5,66	13,60	71,96	13,60	0,98 ± 0,00	0,97 ± 0,01	19,16 ± 2,96	3,62	190,16	35,95	0,82 ± 0,03	0,82 ± 0,03
PIMA	203,55 ± 3,65	28,47	203,55	28,47	0,77 ± 0,01	0,73 ± 0,01	76,36 ± 16,86	10,68	306,36	42,85	0,56 ± 0,04	0,55 ± 0,04
SNR	104,19 ± 2,55	53,98	104,19	53,98	0,71 ± 0,02	0,72 ± 0,02	16,26 ± 1,35	8,42	78,26	40,55	0,51 ± 0,05	0,52 ± 0,04
ION	99,69 ± 4,40	30,58	99,69	30,58	0,89 ± 0,01	0,86 ± 0,01	15,85 ± 3,20	4,86	120,85	37,07	0,56 ± 0,08	0,57 ± 0,07
AUST	122,02 ± 4,72	19,04	122,02	19,04	0,86 ± 0,01	0,86 ± 0,01	83,14 ± 8,08	12,97	290,14	45,26	0,60 ± 0,07	0,60 ± 0,07
LIV	170,35 ± 6,78	53,07	170,35	53,07	0,60 ± 0,02	0,61 ± 0,02	23,80 ± 2,70	7,41	127,80	39,81	0,51 ± 0,03	0,51 ± 0,04
GER	308,08 ± 7,13	33,13	308,08	33,13	0,75 ± 0,01	0,66 ± 0,01	157,48 ± 7,80	16,93	457,48	49,19	0,52 ± 0,04	0,52 ± 0,03
SPAM	520,10 ± 9,37	12,16	520,10	12,16	0,92 ± 0,00	0,91 ± 0,00	447,00 ± 40,64	10,45	1827,00	42,71	0,59 ± 0,06	0,59 ± 0,05
	PCBGZ						PCBGZ-OPT					
Sigla	Rótulos AA	%	Rótulos Efetivos	%	Ac	AUC	Rótulos AA	%	Rótulos Efetivo	%	Ac	AUC
HRT	57,75 ± 1,35	23,01	138,75	55,28	0,74 ± 0,02	0,72 ± 0,03	161,50 ± 0,86	64,34	242,50	96,61	0,81 ± 0,00	0,80 ± 0,01
WBCO	135,67 ± 3,51	21,37	340,67	53,65	0,96 ± 0,00	0,95 ± 0,01	300,65 ± 2,14	47,35	505,65	79,63	0,97 ± 0,00	0,96 ± 0,00
WBCD	124,29 ± 2,17	23,50	295,29	55,82	0,94 ± 0,00	0,93 ± 0,01	292,31 ± 1,16	55,26	463,31	87,58	0,94 ± 0,00	0,94 ± 0,01
PIMA	89,41 ± 2,89	12,50	319,41	44,67	0,65 ± 0,00	0,51 ± 0,01	393,16 ± 1,63	54,99	623,16	87,16	0,68 ± 0,00	0,56 ± 0,01
SNR	88,35 ± 1,87	45,78	150,35	77,90	0,66 ± 0,02	0,68 ± 0,02	128,81 ± 0,39	66,74	190,81	98,87	0,64 ± 0,02	0,66 ± 0,02
ION	136,38 ± 4,78	41,83	241,38	74,04	0,68 ± 0,04	0,73 ± 0,02	215,92 ± 0,50	66,23	320,92	98,44	0,69 ± 0,01	0,72 ± 0,01
AUST	133,04 ± 4,49	20,76	340,04	53,05	0,70 ± 0,02	0,67 ± 0,02	386,64 ± 1,67	60,32	593,64	92,61	0,80 ± 0,00	0,78 ± 0,01
LIV	78,68 ± 2,42	24,51	182,68	56,91	0,50 ± 0,01	0,51 ± 0,01	193,55 ± 0,72	60,30	297,55	92,69	0,50 ± 0,01	0,51 ± 0,01
GER	102,06 ± 1,63	10,97	402,06	43,23	0,71 ± 0,01	0,51 ± 0,01	561,83 ± 2,56	60,41	861,83	92,67	0,72 ± 0,00	0,55 ± 0,00
SPAM	751,68 ± 12,81	17,57	2131,68	49,83	0,78 ± 0,00	0,72 ± 0,01	2229,68 ± 2,14	52,12	3609,68	84,38	0,81 ± 0,00	0,76 ± 0,00
	Pool-based Sampling						SVM com todos os padrões					
	SVMTK						SVMALL					
Sigla	Rótulos AA	%	Rótulos Efetivos	%	Ac	AUC	Rótulos AA	%	Rótulos Efetivos	%	Ac	AUC
HRT	45,70 ± 4,24	18,21	126,70	50,48	0,82 ± 0,01	0,82 ± 0,01	170,00	67,73	251,00	100,00	0,81 ± 0,01	0,83 ± 0,08
WBCO	18,70 ± 1,83	2,94	223,70	35,23	0,97 ± 0,00	0,97 ± 0,00	430,00	67,72	635,00	100,00	0,97 ± 0,00	0,97 ± 0,02
WBCD	40,00 ± 2,79	7,56	211,00	39,89	0,98 ± 0,00	0,97 ± 0,00	358,00	67,67	529,00	100,00	0,98 ± 0,00	0,99 ± 0,02
PIMA	138,60 ± 26,06	19,38	368,60	51,55	0,75 ± 0,01	0,72 ± 0,01	485,00	67,83	715,00	100,00	0,74 ± 0,01	0,71 ± 0,08
SNR	59,60 ± 9,71	30,88	121,60	63,01	0,75 ± 0,02	0,75 ± 0,02	131,00	67,88	193,00	100,00	0,81 ± 0,01	0,82 ± 0,10
ION	55,80 ± 5,77	17,12	160,80	49,33	0,90 ± 0,01	0,87 ± 0,01	221,00	67,79	326,00	100,00	0,91 ± 0,01	0,90 ± 0,07
AUST	68,90 ± 13,90	10,75	275,90	43,04	0,85 ± 0,01	0,85 ± 0,01	434,00	67,71	641,00	100,00	0,84 ± 0,01	0,84 ± 0,04
LIV	96,40 ± 26,01	30,03	200,40	62,43	0,64 ± 0,02	0,64 ± 0,02	217,00	67,60	321,00	100,00	0,67 ± 0,02	0,62 ± 0,11
GER	207,40 ± 17,33	22,30	507,40	54,56	0,74 ± 0,01	0,66 ± 0,01	630,00	67,74	930,00	100,00	0,75 ± 0,01	0,64 ± 0,04
SPAM	340,00 ± 25,26	7,95	1720,00	40,21	0,92 ± 0,00	0,92 ± 0,00	2898,00	67,74	4278,00	100,00	0,93 ± 0,00	0,93 ± 0,02
	ELM2012											
Sigla	Rótulos AA	%	Rótulos Efetivos	%	Ac	AUC						
HRT	170,00	67,73	251,00	100,00	0,82 ± 0,01	0,82 ± 0,01						
WBCO	430,00	67,72	635,00	100,00	0,97 ± 0,00	0,97 ± 0,00						
WBCD	358,00	67,67	529,00	100,00	0,97 ± 0,00	0,97 ± 0,00						
PIMA	484,00	67,69	715,00	100,00	0,73 ± 0,00	0,74 ± 0,00						
SNR	131,00	67,88	193,00	100,00	0,76 ± 0,01	0,76 ± 0,02						
ION	221,00	67,79	326,00	100,00	0,87 ± 0,01	0,84 ± 0,01						
AUST	434,00	67,71	641,00	100,00	0,85 ± 0,00	0,85 ± 0,00						
LIV	217,00	67,60	321,00	100,00	0,64 ± 0,02	0,63 ± 0,02						
GER	630,00	67,74	930,00	100,00	0,70 ± 0,01	0,72 ± 0,01						
SPAM	2898,00	67,74	4278,00	100,00	0,93 ± 0,00	0,92 ± 0,00						

6.4.1 Teste de Significância Estatística

Para uma melhor interpretação dos resultados desta seção será utilizado o teste de Friedman, conforme discutido na seção anterior. As comparações desta seção consistem em: (i) comparar o método EALMSS com 3 métodos do tipo *stream-based* e com os método SVMTK, SVMALL e ELM2012 a fim de

comparar os métodos *stream-based* com os métodos baseados em SVMs e com uma ELM regularizada treinada com todos os padrões, totalizando 7 modelos e 10 bases; (ii) comparar o método EALMPB com as adaptações dos métodos *stream-based* para o contexto *pool-based*, com os modelos baseados em SVMs e com uma ELM regularizada treinada com todos os padrões, totalizando, novamente, 7 modelos e 10 bases.

Comparação entre os métodos stream-based, os modelos baseados em SVMs e a ELM regularizada

Como foram comparados 7 modelos em 10 bases, $L = 7$ e $M = 10$, sendo os graus de liberdade $L - 1 = 6$ e $(L - 1)(M - 1) = 54$. Considerando um nível de significância de 5% e os graus de liberdade apresentados, o valor crítico para os dois modelos é $C = 2{,}2720$ e a hipótese nula será rejeitada somente se $F_F > 2{,}2720$, sendo esse valor calculado para cada modelo e para cada desempenho avaliado, ou seja, em termos de acurácia, AUC, do número de rótulos selecionados e do número de rótulos efetivo.

A Tabela 6.4 apresenta o valor crítico C e o valor calculado de F_F para cada medida de desempenho. Como pode ser observado, a hipótese de que os modelos comparados são equivalentes é rejeitada para todas as medidas. Dessa forma, um teste *post-hoc* deve ser utilizado para cada medida de desempenho para verificar em quais comparações os classificadores não são equivalentes. Como o objetivo é comparar o classificador EALMSS com os demais, deve ser utilizado um teste *post-hoc* do tipo "um-contra-todos". Segundo Demšar [18] para essa situação o teste mais adequado é o de Bonferroni-Dunn [19]. Para esse teste o desempenho entre dois classificadores é significativamente diferente se os *ranks* médios diferem em pelo menos o valor da diferença crítica (DC)[18]:

$$DC = q_\alpha \sqrt{\frac{L(L+1)}{6M}} \tag{6.3}$$

onde o valor crítico q_α é obtido da Tabela 6.5. Dessa forma, dois classificadores serão estatisticamente diferentes se $|R_i - R_j| > DC$, onde R é o *rank* médio de cada classificador comparado. No teste de Bonferroni-Dunn é escolhido um algoritmo de controle, ou seja, o algoritmo que se deseja comparar com os demais. No caso dos experimentos apresentados nesta subseção o algoritmo de controle é o modelo EALMSS.

Os resultados do teste de Bonferroni-Dunn são apresentados na Tabela 6.6. Como pode ser observado o modelo EALMSS em termos de Ac e AUC tem resultados estatisticamente diferentes dos modelos PDKCM, PCBGZ e PCBGZ-OPT, uma vez que a hipótese nula é rejeitada para esses modelos. Em relação

Tabela 6.4: Resultados do teste de Friedman: Modelos *stream-based*, modelos baseados em SVM e ELM regularizada

	F_F	C	Rejeitar ?
Ac	11,6473	2,2720	sim
AUC	15,4068	2,2720	sim
Rótulos AL	71,8989	2,2720	sim
Rótulos EF	137,9388	2,2720	sim

Tabela 6.5: Valores críticos para o teste de Bonferroni-Dunn. O número de modelos inclui o de controle [18]

Nº modelos	2	3	4	5	6	7	8	9	10
$q_{0,05}$	1,960	2,241	2,394	2,498	2,576	2,638	2,690	2,724	2,773
$q_{0,10}$	1,645	1,960	2,128	2,241	2,326	2,394	2,450	2,498	2,539

aos modelos SVMTK, SVMALL e ELM2012 a hipótese nula em termos de Ac e AUC não pode ser rejeitada. Observando-se novamente a Tabela 6.3 e comparando o modelo EALMSS com os modelos SVMTK, SVMALL e ELM2012, verifica-se que mesmo EALMSS sendo tipo *stream-based* tem resultados similares a abordagens do tipo *pool-based*, porém com custo computacional muito mais baixo uma vez que analisa os padrões apenas uma única vez e não realiza nenhum processo de otimização. Além disso, os resultados do modelo EALMSS são similares à SVM treinada com todos os padrões e a ELM regularizada treinada com todos os padrões, o que parece indicar que é verdadeira a hipótese de que o *crosstalk* funciona implicitamente como um termo de regularização, com a vantagem de ser ajustado automaticamente pelo processo de Aprendizado Ativo escolhendo-se os padrões mais informativos.

Em termos dos rótulos selecionados durante a fase de aprendizado ativo não há diferença estatística entre os valores obtidos pelo método EALMSS e os demais métodos de aprendizado ativo comparados. A diferença é encontrada apenas quando comparado com o modelos SVMALL e ELM2012 que utilizam todos os dados disponíveis. Isso mostra que apesar dos outros métodos de aprendizado ativo selecionarem um número médio de rótulos ligeiramente menor para algumas bases, do ponto de vista estatístico não há diferença em relação aos valores obtidos pelo método EALMSS. Novamente essa é uma vantagem do método, uma vez que nenhum ajuste fino de parâmetros foi necessário.

Por fim, não há diferença estatística em termos do número de rótulos efetivos utilizados pelo método EALMSS em relação aos métodos PDKCM e SVMTK. Porém, o método EALMSS não realiza nenhum ajuste de parâmetros. Além disso, existe diferença estatística entre os resultados obtidos pelo modelo EALMSS em termos de Ac e AUC em relação ao modelo PDKCM, sendo os resultados do modelo EALMSS melhores, conforme pode ser observado na

Tabela 6.6: Resultados do teste de Bonferroni-Dunn. Modelo de controle: EALMSS

	Diferença entre *ranks* médios						
	PDKCM	PCBGZ	PCBGZ-OPT	SVMTK	SVMALL	ELM2012	DC
Ac	2,9500	3,0000	3,3500	0,1000	0,3000	0,7000	2,5486
Rejeitar?	sim	sim	sim	**não**	**não**	**não**	
AUC	2,8000	3,1000	3,6000	0,1000	0,0500	0,1000	2,5486
Rejeitar?	sim	sim	sim	**não**	**não**	**não**	
Rótulos AA	1,2000	0,1000	1,8000	1,7000	3,3500	3,2500	2,5486
Rejeitar?	**não**	**não**	**não**	**não**	sim	sim	
Rótulos Ef	1,8000	2,7000	4,0000	1,5000	5,5500	5,4500	2,5486
Rejeitar?	**não**	sim	sim	**não**	sim	sim	

Tabela 6.3.

Observa-se na Tabela 6.6 que a hipótese de que o modelo EALMSS tem resultados similares ao modelo SVMTK em todas as medidas de desempenho não pode ser rejeitada. Pela Tabela 6.3 pode-se concluir que o modelo EALMSS tem resultados similares ao SVMTK, porém com a vantagem de analisar cada padrão apenas uma única vez, não precisar realizar ajuste fino de parâmetros e não realizar nenhum processo de otimização computacionalmente caro.

Comparação entre os métodos Pool-based

Novamente foram comparados 7 modelos em 10 bases resultando em $L = 7$ e $M = 10$. Foram comparados os modelos EALMPB com as versões adaptadas *pool-based* dos métodos PDKCM, PCBGZ e PCBGZ-OPT, com os modelos baseados em SVMs e com a ELM regularizada. Novamente, considerando um nível de significância de 5%, o valor crítico é $C = 2{,}2720$. Os modelos são comparados em termos de acurácia, AUC, do número de rótulos selecionados e do número de rótulos efetivo.

A Tabela 6.7 apresenta o valor crítico C e o valor calculado de F_F para cada medida de desempenho. Como pode ser observado a hipótese de que os modelos comparados são equivalentes é rejeitada para todas as medidas e, conforme discutido na subseção anterior, o teste de Bonferroni-Dunn precisa ser utilizado. O modelo de controle utilizado foi o modelo EALMPB. Os resultados desse teste são apresentados na Tabela 6.8.

Como pode ser observado, a hipótese de que os resultados do modelo EALMPB em termos de Ac e AUC são estatisticamente similares aos dos modelos SVMTK, SVMALL e ELM2012, não pode ser rejeitada nesses três casos. Com relação ao número de rótulos utilizados pelo aprendizado ativo a hipótese nula é rejeitada para os modelos SVMALL e ELM2012, que utilizam todos os rótulos.

Assim como os resultados do modelo EALMSS, o modelo EALMPB também teve resultados similares ao modelo SVMTK em todas as medidas de desem-

penho analisadas, como pode ser visto na Tabela 6.3. A vantagem do modelo EALMPB em relação ao modelo SVMTK é que não necessita de ajuste fino de parâmetros e não utiliza nenhum processo de otimização computacionalmente caro.

Tabela 6.7: Resultados do teste de Friedman: Modelos *pool-based*

	F_F	C	Rejeitar ?
Ac	25,0311	2,2720	sim
AUC	33,1757	2,2720	sim
Rótulos AA	90,4083	2,2720	sim
Rótulos Ef	112,4458	2,2720	sim

Tabela 6.8: Resultados do teste de Bonferroni-Dunn. Modelo de controle: EALMPB

	Diferença entre *ranks* médios						
	PDKCM	PCBGZ	PCBGZ-OPT	SVMTK	SVMALL	ELM2012	DC
Ac	4,6000	3,6000	2,8000	0,4500	0,1000	1,0500	2,5486
Rejeitar?	sim	sim	sim	**não**	**não**	**não**	
AUC	4,1000	3,4000	2,7000	0,0000	0,2500	0,1500	2,5486
Rejeitar?	sim	sim	sim	**não**	**não**	**não**	
Rótulos AA	2,2000	0,4000	1,5000	1,4000	3,0500	2,9500	2,5486
Rejeitar?	**não**	**não**	**não**	**não**	sim	sim	
Rótulos Ef	0,9000	2,4000	3,8000	1,9000	5,3500	5,2500	2,5486
Rejeitar?	**não**	**não**	sim	**não**	sim	sim	

6.5 Conclusões do Capítulo

Neste capítulo foram apresentados resultados para quatro experimentos. O primeiro teve o objetivo de mostrar as limitações das ELMs na realização do aprendizado ativo. O segundo mostrou a evolução da relação entre erro de treinamento e complexidade do modelo treinado utilizando as estratégias de aprendizado ativo propostas nesta tese. O experimento seguinte apresentou resultados que demonstraram que o desempenho dos modelos de aprendizado ativo propostos não dependem do número de neurônios escondidos se a camada escondida for suficientemente grande. Por fim, foi apresentada a comparação entre os métodos propostos com outros presentes na literatura. Foi verificado que os dois modelos propostos têm resultados em termos de acurácia e AUC similares àqueles obtidos pelos modelos baseados em SVMs e a ELM regularizada. Além disso, a proximidade desses resultados indica que o termo de *crosstalk* parece funcionar como um termo de regularização implícito e com ajuste automático realizado pelo aprendizado ativo.

No próximo capítulo serão discutidas todas as contribuições desta tese e serão levantados questionamentos importantes que levarão a possíveis trabalhos futuros.

CAPÍTULO 7

Discussão

Os métodos desenvolvidos neste trabalho resolveram de forma muito simples a dificuldade existente em realizar o Aprendizado Ativo em problemas não linearmente separáveis. Ao utilizar uma camada escondida ELM para linearizar o problema, foi possível realizar um mapeamento do espaço de entrada em um espaço de características que é praticamente livre de ajuste fino de parâmetros. Ao utilizar um Perceptron treinado com aprendizado Hebbiano e com pesos normalizados foi possível escapar das limitações da pseudoinversa, sobretudo no que diz respeito à relação entre o número de neurônios escondidos e o número de padrões utilizados. Além disso, o teste de convergência do Perceptron pôde ser estendido para a versão treinada com aprendizado Hebbiano e pesos normalizados, mostrando-se um excelente critério de seleção de rótulos para um modelo *Stream-based* (EALMSS) ou uma excelente condição de parada para um modelo *Pool-based* (EALMPB).

A análise do Perceptron treinado com aprendizado Hebbiano mostrou que uma boa solução seria obtida somente se o *crosstalk* fosse devidamente controlado, tendo como efeito benéfico a capacidade de regularizar o modelo, fazendo com que a solução obtida fosse desviada da solução de erro zero que seria similar à solução da pseudoinversa. Ao utilizar o teste de convergência para definir que padrões devem ou não ser aprendidos, foi possível controlar automaticamente a evolução do *crosstalk* e, como consequência, a complexidade do modelo, ao selecionar apenas os padrões que não são redundantes. A expressão encontrada para o número de padrões necessários para convergência mostrou ter a capacidade de extrair o "esqueleto" da distribuição dos dados quando utilizada no teste de convergência. Dessa forma, os padrões selecionados representam de forma compacta as regiões do espaço ocupadas

pelos dados de entrada, de forma que ao deixar de aprender padrões redundantes, ou seja, que representem a mesma região do espaço, a complexidade do modelo não precisa ser aumentada. Ao escolher apenas os padrões que satisfazem o teste de convergência em cada iteração, automaticamente o modelo converge para uma solução com boa relação custo-benefício entre erro de treinamento e complexidade do modelo.

Essas conclusões indicam que os modelos desenvolvidos nesta tese tendem a minimizar o erro empírico (erro de treinamento) controlando ao mesmo tempo a complexidade do modelo, o que talvez indique que o teste de convergência também controla o tamanho da dimensão VC. Essa possibilidade parece adequada, pois explicaria o porquê do modelo convergir para uma boa solução em termos de capacidade de generalização. Esse hipótese abre um novo caminho para o entendimento do Aprendizado Ativo, que pode passar a ser vista como uma forma de controlar a dimensão VC para obter modelos com boa capacidade de generalização.

Os resultados experimentais suportam essa hipótese, uma vez que é mostrado que os modelos desenvolvidos obtém soluções estatisticamente similares àquelas obtidas pelos modelos baseados em SVMs. Vale destacar que o objetivo dos modelos baseados em SVMs é exatamente minimizar o risco empírico controlando a dimensão VC para garantir boa capacidade de generalização. A vantagem dos métodos propostos é que o custo computacional é baixo e nenhum processo de otimização é necessário.

A aparente capacidade dos modelos propostos em controlar o erro de treinamento juntamente com a complexidade do modelo indica que uma comparação com modelos de aprendizado de redes neurais baseados em treinamento multiobjetivo deve ser realizada, pois aparentemente a solução encontrada tem as mesmas características que aquelas que podem ser obtidas pelo método multiobjetivo, com a vantagem de não ser necessário nenhum processo de seleção de modelos, pois um Pareto não precisa ser criado para se obter uma solução com boa relação de custo-benefício.

Outro tema que precisa ser tratado em trabalhos futuros diz respeito ao tamanho do espaço de entrada. Se o espaço de entrada for muito grande, como em problemas de microarrajos [34] que podem ter mais de 22000 entradas, por exemplo, uma alternativa ao uso de neurônios escondidos deve ser considerada, pois obter uma camada escondida grande o suficiente poderia tornar o processo lento. Uma alternativa que parece promissora para esse problema é o *kernel* ELM assintótico infinito [26] que é uma função de *kernel* equivalente a ter infinitos neurônios na camada escondida. Entretanto, ao utilizar esse *kernel* o vetor de pesos deixaria de ser calculado diretamente, ficando implícito no espaço de características. Em virtude disso a norma do vetor de pesos

nesse espaço de características precisaria ser definida a fim de se realizar o cálculo de t_{max}, o que até o presente momento permanece como um problema em aberto.

Uma questão teórica interessante que permanece aberta é a possibilidade do Perceptron Hebbiano com pesos normalizados [21] ter alguma relação com a regra de Oja [52], uma vez que o cálculo das duas é semelhante. Tal análise será objeto de trabalhos futuros.

Considerando a semelhança do Perceptron utilizado com as SVMs, um ponto importante a ser explorado é a possibilidade de adaptar os métodos propostos para trabalharem com problemas multiclasses. Tal adaptação pode ser baseada em modelos de SVMs multiclasses existentes na literatura [38]. Outro ponto de interesse é a adaptação dos modelos propostos para a realização de Aprendizado Ativo em problemas de classes desbalanceadas. Uma dificuldade desse tipo de problema é que os padrões têm uma probabilidade maior de pertencerem à classe majoritária, o que pode prejudicar o desempenho de modelos baseados em fluxo de dados.

Por fim, o presente trabalho apresentou uma alternativa simples e prática para a realização do Aprendizado Ativo em problemas não linearmente separáveis, que poderá servir de base para vários outros trabalhos, conforme discutido.

CAPÍTULO
8

Conclusão

O Aprendizado Ativo tem atraído a atenção de vários pesquisadores nos últimos anos já que grandes quantidades de dados tem sido geradas para várias classes de problemas, sendo que para algumas delas o custo da rotulação é elevado. Isso motiva o estudo de técnicas que possibilitem desenvolver bons classificadores com o menor número de rótulos possível. Muitos trabalhos se dedicaram ao estudo de classificadores lineares, mas pouco foi feito em relação a problemas não linearmente separáveis. Além disso, a maioria dos modelos encontrados na literatura exige ajuste fino de parâmetros, sendo que essa necessidade impacta diretamente no objetivo principal do Aprendizado Ativo no que se refere ao número de rótulos utilizados, uma vez que esse ajuste é feito reservando-se uma quantidade de rótulos para esse fim.

Dessa forma, esta tese teve o objetivo de encarar essas duas questões, sendo que foram desenvolvidos modelos capazes de classificar problemas não linearmente separáveis minimizando o número de rótulos necessários e sem a necessidade de ajuste fino de parâmetros. Os modelos desenvolvidos são baseados em uma rede neural de duas camadas. A camada escondida é do tipo ELM, que é capaz de projetar os dados em um espaço de dimensão elevada, aumentado a probabilidade de que os mesmos se tornem linearmente separáveis nesse novo espaço. Já a camada de saída é composta por um Perceptron treinado com aprendizado Hebbiano e com pesos normalizados. O Perceptron utilizado se comporta como uma SVM em que todos os padrões utilizados para treinamento são considerados vetores de suporte com multiplicadores de Lagrange iguais a um. Os modelos têm a capacidade de controlar o *crosstalk* de um Perceptron Hebbiano e, como consequência, obtém uma solução com boa

relação custo-benefício entre erro de treinamento e complexidade do modelo treinado.

Foi demonstrado que o teorema de convergência do Perceptron clássico pode ser adaptado para o Perceptron Hebbiano. Essa adaptação foi utilizada como critério de seleção para o Aprendizado Ativo, selecionando para rotulação apenas os padrões capazes de representar o "esqueleto" da distribuição espacial dos dados em relação ao hiperplano separador. Essa escolha parece maximizar a capacidade de generalização do modelo proposto. Foi demonstrado ainda que o Aprendizado Ativo baseado no cálculo da pseudoinversa para as ELMs não é adequado, uma vez que o seu uso exige que o número de padrões de treinamento seja muito maior que o número de neurônios utilizados.

O modelo proposto foi comparado com outros modelos presentes na literatura, tanto nas versões *Selective Sampling* quanto *Pool-based Sampling*. Foi verificado que os resultados obtidos em termos da acurácia e da AUC são muito próximos dos obtidos utilizando-se SVMs com aprendizado ativo, SVMs com aprendizado passivo e próximo dos resultados obtidos com uma ELM regularizada. Foi verificado que o número de rótulos utilizado para a construção do modelo é menor que o utilizado pelos outros métodos, uma vez que devem ser considerados tanto os rótulos separados para o ajuste de parâmetros quanto os rótulos obtidos pelo Aprendizado Ativo.

Por fim, os métodos propostos se mostraram uma boa alternativa não somente para problemas de Aprendizado Ativo, mas também para problemas passivos em geral, uma vez que os métodos possuem uma forma de aprendizado simples, prática e eficiente e que ainda fornecem boas soluções em termos de erro de treinamento e da complexidade dos modelos, sendo, portanto, soluções com boa capacidade de generalização. A seguir, serão apresentadas possíveis propostas de continuidade para este trabalho, com foco principal na solução das questões que permanecem abertas após o fim desta tese.

8.1 Trabalhos Futuros

Esta tese apresentou dois modelos de Aprendizado Ativo extremamente simples. Ao longo do trabalho foram verificadas características importantes dos modelos, como, por exemplo, a capacidade de encontrar uma solução com boa relação custo-benefício em termos do erro de treinamento e da complexidade do modelo. Foi observado ainda que o uso do aprendizado Hebbiano possibilita que seja realizado o desaprendizado em problemas dinâmicos, que o Perceptron proposto parece ter relação com a Regra de Oja [52] e que o Aprendizado Ativo parece controlar a dimensão VC. Essas questões somadas

à outras questões comuns aos problemas de aprendizado de máquina motivam os seguintes trabalhos futuros:

1. Comparar os métodos de aprendizado ativo propostos com o aprendizado multiobjetivo para verificar as semelhanças e diferenças em relação à minimização do erro com controle da complexidade do modelo;

2. Verificar a relação entre o Perceptron Hebbiano com pesos normalizados e a regra de Oja [52]. Essa última é capaz de extrair a primeira componente principal dos dados;

3. Verificar a aplicação dos modelos propostos em problemas dinâmicos que necessitem o desaprendizado de padrões obsoletos ao longo do tempo;

4. Buscar alternativas para lidar com espaços de entrada muito grandes. Nesse caso a camada escondida ELM deixa de ser adequada, pois o número de neurônios necessários para realizar a linearização do problema poderia ser muito elevada. Uma alternativa já apresentada na literatura para esse problema é o *kernel* ELM assintótico infinito [26] que é uma função de *kernel* que equivale a utilizar um número infinito de neurônios na camada escondida;

5. Desenvolver métodos para lidar com problemas multimodais. Os algoritmos precisam ser inicializados de alguma forma que possibilite que padrões representativos de todos os agrupamentos presentes no problema estejam disponíveis na etapa inicial;

6. Desenvolver uma versão multiclasse para os algoritmos propostos;

7. Buscar formas de lidar com problemas de classes desbalanceadas, uma vez que a classe majoritária poderia ser priorizada pelo modelo, já que é utilizado no treinamento um conjunto reduzido de dados;

8. Como o modelo EALMSS tem um treinamento muito simples e eficiente, torna-se interessante a implementação do mesmo em *hardware*, como em FPGAs por exemplo, possibilitando que o Aprendizado Ativo possa ser executado diretamente do *hardware*. Isso seria importante para lidar com problemas de visão computacional, por exemplo.

Produção Científica

Durante o curso de doutorado foram geradas as seguintes produções científicas:

- Relacionadas diretamente com a tese:

  1. HORTA, E. G. ; CASTRO, C. L. ; BRAGA, A. P. *Stream-based Extreme Learning Machine Approach for Big Data Problems.* Mathematical Problems in Engineering, vol. 2015, p. 1-17, 2015. doi:10.1155/2015/126452.
  2. HORTA, E. G. ; BRAGA, A. P. *An Extreme Learning Approach to Active Learning.* In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2014, Bruges, Bélgica. ESANN 2014 proceedings, 2014. p. 613-618.
  3. HORTA, E. G. ; BRAGA, A. P. *Aplicação de Máquinas de Aprendizado Extremo ao Problema de Aprendizado Ativo.* In: 1st BRICS Countries Congress (BRICS-CCI) and 11th Brazilian Congress (CBIC) on Computational Intelligence, 2013, Porto de Galinhas.

- Não relacionada com a tese:

  1. HORTA, E. G. ; SHIGUEMORI, E. H. ; VELHO, H. F. C. ; BRAGA, A. P. *Extração de características e casamento de padrões aplicados à estimação de posição de um VANT.* In: XIX Congresso Brasileiro de Automática (CBA2012), 2012, Campina Grande. Anais do XIX Congresso Brasileiro de Automática, CBA 2012. Campina Grande, 2012. p. 5045-5050.

Referências Bibliográficas

[1] Abe, N., and Mamitsuka, H. Query Learning Strategies using Boosting and Bagging. In Fifteenth International Conference on Machine Learning (San Francisco, USA, 1998).

[2] Andrus, W. S., and Bird, K. T. Radiology and Receiver Operating Characteristic ROC Curve. CHEST 67, 4 (1975), 378–379.

[3] Anton, H., and Busby, R. Contemporary Linear Algebra. Wiley, 2002.

[4] Atlas, L., Cohn, D., Ladner, R., El-Sharkawi, M. A., and Marks, R. J. Training Connectionist Networks with Queries and Selective Sampling. In Advances in neural information processing systems 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990, pp. 566—-573.

[5] Auer, P., Burgsteiner, H., and Maass, W. A Learning Rule for Very Simple Universal Approximators Consisting of a Single Layer of Perceptrons. Neural networks : the official journal of the International Neural Network Society 21, 5 (June 2008), 786–95.

[6] Bartlett, P. L. For Valid Generalization the Size of the Weights is More Important than the Size of the Network. In Advances in Neural Information Processing Systems (1997), M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., vol. 9, The MIT Press.

[7] Braga, A. P., Carvalho, A. P. L. F., and Ludemir, T. B. Redes Neurais Artificiais: Teoria e Aplicações. Livros Técnicos e Científicos, Rio de Janeiro, RJ, 2007.

[8] Braga, A. P., Takahashi, R. H. C., Teixeira, R. A., and Costa, M. A. Multiobjective Learning. Springer, 2006, ch. Multi-Objective Algorithms for Neural Networks Learning, pp. 151–172.

[9] CASTRO, C. L. D. Novos Critérios para Seleção de Modelos Neurais em Problemas de Classificação com Dados Desbalanceados. Ph. D., Programa de Pós-Graduação em Engenharia Elétrica, UFMG, 2011.

[10] CESA-BIANCHI, N., GENTILE, C., AND ZANIBONI, L. Worst-Case Analysis of Selective Sampling for Linear Classification. Journal of Machine Learning Research 7 (2006), 1205–1230.

[11] COHN, D., ATLAS, L., AND LADNER, R. Improving Generalization with Active Learning. Machine Learning 15, 2 (May 1994), 201–221.

[12] COSTA, M. A., BRAGA, A. P., AND MENEZES, B. R. Improving Generalization of MLPs with Sliding Mode Control and the Levenberg-Marquadt Algorithm. Neurocomputing 70 (2007), 1342–1347.

[13] COSTA, M. A., BRAGA, A. P., MENEZES, B. R., TEIXEIRA, R. A., AND PARMA, G. G. Training Neural Networks with a Multi-Objective Sliding Mode Control Algorithm. Neurocomputing, 51 (2003), 467–473.

[14] COVER, T. M. Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. IEEE Transactions On Electronic Computers EC-14, 3 (1965), 326–334.

[15] DASGUPTA, S., KALAI, A. T., AND MONTELEONI, C. Analysis of Perceptron-Based Active Learning. Journal of Machine Learning Research 10 (2009), 281–299.

[16] DAVID, H. A., AND GUNNINK, J. L. The Paired t Test Under Artificial Pairing. The American Statistician 51, 1 (1997), pp. 9–12.

[17] DE CASTRO, L., AND VON ZUBEN, F. Learning and Optimization Using the Clonal Selection Principle. Evolutionary Computation, IEEE Transactions on 6, 3 (Jun 2002), 239–251.

[18] DEMŠAR, J. Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res. 7 (dec 2006), 1–30.

[19] DUNN, O. J. Multiple Comparisons among Means. Journal of the American Statistical Association 56 (1961), 52–64.

[20] FAN, R.-E., CHANG, K.-W., HSIEH, C.-J., WANG, X.-R., AND LIN, C.-J. LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research 9 (June 2008), 1871–1874.

[21] FERNANDEZ-DELGADO, M., RIBEIRO, J., CERNADAS, E., AND AMENEIRO, S. B. Direct Parallel Perceptrons (DPPs): Fast Analytical Calculation of

the Parallel Perceptrons Weights with Margin Control for Classification Tasks. IEEE Transactions on Neural Networks 22, 11 (Nov. 2011), 1837–48.

[22] FRANK, A., AND ASUNCION, A. UCI Machine Learning Repository.

[23] FRÉNAY, B., AND VERLEYSEN, M. Using SVMs with Randomised Feature Spaces: an Extreme Learning Approach. In European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning (2010), no. April, pp. 315–320.

[24] FRIEDMAN, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. Journal of the American Statistical Association 32, 200 (dec 1937), 675–701.

[25] FRIEDMAN, M. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. The Annals of Mathematical Statistics 11, 1 (1940), 86–92.

[26] FRÉNAY, B., AND VERLEYSEN, M. Parameter-Insensitive Kernel in Extreme Learning for Non-Linear Support Vector Regression. Neurocomputing 74, 16 (2011), 2526–2531.

[27] GARDNER, E. Maximum Sotorage Capacity in Neural Networks. Europhysics Letters 4 (1987), 481–485.

[28] GIROSI, F., JONES, M., AND POGGIO, T. Regularization Theory and Neural Networks Architectures. Neural Computation 7 (1995), 219–269.

[29] GUILLORY, A., CHASTAIN, E., AND BILMES, J. Active Learning as Non-Convex Optimization. In 12th International Conference on Artificial Intelligence and Statistics (AISTATS) (Clearwater Beach, Florida, 2009), vol. 5.

[30] HAYKIN, S. Neural Networks: A Comprehensive Foundation. Macmillan, New York, 1994.

[31] HEBB, D. O. The Organization of Behavior. Wiley, New York, 1949.

[32] HERTZ, J., KROGH, A., AND PALMER, R. Introduction to the Theory of Neural Computation. Advanced book program: Addison-Wesley. Addison-Wesley Publishing Company, 1991.

[33] HOPFIELD, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proceedings of the National Academy of Sciences 79 (1982), 2554–2558.

[34] HORTA, E. G. Previsores para a Eficiência da Quimioterapia Neoadjuvante no Câncer de Mama. M. Sc., Universidade Federal de Minas Gerais (UFMG), Novembro 2008.

[35] HORTA, E. G., AND BRAGA, A. P. Aplicação de Máquinas de Aprendizado Extremo ao Problema de Aprendizado Ativo. In Congresso Brasileiro de Inteligência Computacional (Recife, PE, Brasil, 2013).

[36] HORTA, E. G., AND BRAGA, A. P. An Extreme Learning Approach to Active Learning. In European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (Bruges, Belgium, 2014), i6doc.com, pp. 613–618.

[37] HORTA, E. G., DE CASTRO, C. L., AND BRAGA, A. P. Stream-Based Extreme Learning Machine Approach for Big Data Problems. Mathematical Problems in Engineering 2015 (2015), 17.

[38] HSU, C.-W., AND LIN, C.-J. A Comparison of Methods for Multiclass Support Vector Machines. Neural Networks, IEEE Transactions on 13, 2 (Mar 2002), 415–425.

[39] HUANG, G., ZHU, Q., AND SIEW, C. Extreme Learning Machine: Theory and Applications. Neurocomputing 70, 1-3 (Dec. 2006), 489–501.

[40] HUANG, G.-B., ZHOU, H., DING, X., AND ZHANG, R. Extreme Learning Machine for Regression and Multiclass Classification. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42, 2 (Apr. 2012), 513–529.

[41] KOKSHENEV, I., AND BRAGA, A. P. Complexity Bounds for Radial Basis Functions and Multi-Objective Learning. In European Symposium on Neural Networks (ESANN07) (2007), pp. 73–78.

[42] LEWIS, D. D., AND GALE, W. A. A Sequential Algorithm for Training Text Classifiers. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (Dublin, Ireland, 1994), Springer-Verlag New York, Inc.

[43] LIANG, N.-Y., HUANG, G.-B., SARATCHANDRAN, P., AND SUNDARARAJAN, N. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. Trans. Neur. Netw. 17, 6 (Nov. 2006), 1411–1423.

[44] LIU, Q., HE, Q., AND SHI, Z. Extreme Support Vector Machine Classifier. In Advances in Knowledge Discovery and Data Mining, T. Washio, E. Suzuki, K. Ting, and A. Inokuchi, Eds., vol. 5012 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 222–233.

[45] McEliece, R. J., Posner, E. C., Rodemich, E. R., and Venkatesh, S. S. The Capacity of the Hopfield Associative Memory. IEEE Transactions on Information Theory 33, 4 (July 1987), 461–482.

[46] Medeiros, T., and Braga, A. P. A New Decision Strategy in Multi-Objective Training of Artificial Neural Networks. In European Symposium on Neural Networks (ESANN07) (2007), pp. 555–560.

[47] Melville, P., and Mooney, R. J. Diverse Ensembles for Active Learning. In 21st International Conference on Machine Learning (Banff, Canada, 2004).

[48] Mitchell, T. M. Generalization as Search. Artificial Intelligence 18, 2 (Mar. 1982), 203–226.

[49] Monteleoni, C., and Kääriäinen, M. Practical Online Active Learning for Classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2007), vol. 328.

[50] Nemenyi, P. Distribution-Free Multiple Comparisons. PhD thesis, New Jersey, USA, 1963.

[51] Nilsson, N. Learning Machines. McGraw-Hill, New York, 1965.

[52] Oja, E. Simplified Neuron Model as a Principal Component Analyzer. Journal of Mathematical Biology 15, 3 (1982), 267–273.

[53] Peretto, P. On Learning Rules and Memory Storage Abilities of Asymmetrical Neural Networks. Journal de Physique 49 (1988), 711–726.

[54] Personnaz, L., Guyon, I., and Dreyfus, G. Information Storage and Retrieval in Spin-Glass-Like Neural Networks. Journal de Physique Lettres 46 (1985), 359–365.

[55] Rosenblatt, F. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan, Washington, DC, 1962.

[56] Samat, A., Gamba, P., Du, P., and Luo, J. Active Extreme Learning Machines for Quad-Polarimetric SAR Imagery Classification. International Journal of Applied Earth Observation and Geoinformation 35, Part B (2015), 305 – 319.

[57] Schmidt, W., Kraaijveld, M., and Duin, R. Feedforward Neural Networks with Random Weights. In Pattern Recognition, 1992. Vol.II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on (Aug 1992), pp. 1–4.

[58] SCHOHN, G., AND COHN, D. Less is More: Active Learning with Support Vector Machines. In Seventeenth International Conference on Machine Learning (San Francisco, USA, 2000).

[59] SEMOLINI, R. Support Vector Machines, Inferência Transdutiva e o Problema de Classificação. M. Sc., Universidade Estadual de Campinas (Unicamp), Dezembro 2002.

[60] SETTLES, B. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[61] SETTLES, B. Active Learning Literature Survey. Tech. rep., University of Wisconsin–Madison, 2010.

[62] SETTLES, B. Active Learning. Morgan & Claypool Publishers, 2012.

[63] SEUNG, H. S., OPPER, M., AND SOMPOLINSKY, H. Query by Committee. Proceedings of the fifth annual workshop on Computational learning theory - COLT '92 (1992), 287–294.

[64] SONNENBURG, S., RÄTSCH, G., HENSCHEL, S., WIDMER, C., BEHR, J., ZIEN, A., DE BONA, F., BINDER, A., GEHL, C., AND FRANC, V. The SHOGUN Machine Learning Toolbox. Journal of Machine Learning Research 11 (June 2010), 1799–1802.

[65] SRINIVAS, M., AND PATNAIK, L. Genetic Algorithms: a Survey. Computer 27, 6 (June 1994), 17–26.

[66] STEELE, J. M. The Cauchy-Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities. Cambridge University Press, New York, NY, USA, 2004.

[67] SUTTORP, T., AND IGEL, C. Multi-Objective Optimization of Support Vector Machines. Springer, 2006, ch. Multi-Objective Algorithms for Neural Networks Learning, pp. 199–220.

[68] TEIXEIRA, R. A., BRAGA, A. P., SALDANHA, R. R., TAKAHASHI, R. H. C., AND MEDEIROS, T. H. The Usage of Golden Section in Calculating the Efficient Solution in Artificial Neural Networks Training by Multi-objective Optimization. In International Conference on Neural Networks (ICANN07) (2007).

[69] TEIXEIRA, R. A., BRAGA, A. P., TAKAHASHI, R. H. C., AND SALDANHA, R. R. Improving Generalization of MLPs with Multi-Objective Optimization. Neurocomputing, 35 (2000), 189–194.

[70] TONG, S., AND KOLLER, D. Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research (2001), 45–66.

[71] VAPNIK, V. N. An Overview of Statistical Learning Theory. IEEE Transactions on Neural Networks 10, 5 (Jan. 1999), 988–99.

[72] VAPNIK, V. N. The Nature of Statistical Learning Theory. Springer, 2000.

[73] WILCOXON, F. Individual Comparisons by Ranking Methods. Biometrics Bulletin 1, 6 (Dec. 1945), 80–83.

[74] YU, H., SUN, C., YANG, W., YANG, X., AND ZUO, X. AL-ELM: One Uncertainty-Based Active Learning Algorithm Using Extreme Learning Machine. Neurocomputing 166 (2015), 140 – 150.