

Resumo

A aplicabilidade da inteligência artificial é multidisciplinar. Várias são as áreas que podem se beneficiar do uso dessa ciência. Entre elas, encontra-se a medicina, a qual proporciona um campo excitante de pesquisa. Este trabalho apresenta um estudo comparativo entre técnicas de aprendizado de máquina aplicadas a bases de dados médicas. Quatro técnicas foram utilizadas, redes neurais do tipo MLP-*Backpropagation* e RBF-DDA, máquinas de vetores suporte (SVMs) e a técnica do vizinho mais próximo (kNN). Seis bases de dados médicas compuseram o estudo. As bases foram referentes a: doença inflamatória intestinal, câncer de mama, diabetes, doença do fígado, câncer pulmonar e glândula tireóide. Para a aplicação das técnicas, três simuladores foram utilizados, SNNS, para redes RBF-DDA, WEKA, para kNN e redes MLP, e LIBSVM para SVM. Validação cruzada em 10 blocos foi utilizada para gerar resultados mais confiáveis e possibilitar melhor comparação entre as técnicas. As técnicas foram comparadas segundo o desempenho de classificação nos conjuntos de teste, tempo de treinamento, tempo de classificação e complexidade do classificador.

Abstract

Artificial intelligence comprises a set of techniques that are successfully applied to several areas of knowledge. Medicine is one of these areas. The application of machine learning techniques to problems in medicine is currently an exciting field of research. This work presents a comparative study between techniques of machine learning applied to medical databases. Four techniques have been used: *MLP-Backpropagation* and RBF-DDA artificial neural networks, support vector machines (SVMs) and the k-nearest neighbour (kNN). Six medical databases were considered in this study: intestinal inflammatory illness, breast cancer, diabetes, illness of the liver, pulmonary cancer and thyroid gland. For the application of the techniques, three simulators had been used, SNNS, for RBF-DDA neural networks, WEKA, for kNN and MLP neural networks, and LIBSVM for SVM. 10-fold cross validation was used to compare the classifiers. The techniques have been compared considering the 10-fold cross validation classification performance, the training time, the classification time and the complexity of the classifier.

Sumário

Índice de Figuras	v
Índice de Tabelas	vii
Tabela de Siglas	viii
Introdução	10
Capítulo 1 - Técnicas de aprendizado de máquina	12
1.1. Redes Neurais Artificiais	12
1.1.1 Neurônio biológico	12
1.1.2 Histórico	13
1.1.3 Características gerais	15
1.1.4 Unidades em redes neurais	15
1.1.5 Funções de ativação	16
1.1.6 Estruturas de redes neurais	18
1.1.7 Formas de aprendizado	19
1.1.8 Redes multilayer perceptron	20
1.1.9 Redes de função de base radial	23
1.2 Máquina de vetor suporte (SVM)	29
1.3 Técnica do vizinho mais próximo (kNN)	30
1.4 Validação cruzada e Normalização	31
1.5 Técnicas de aprendizado de máquina e medicina	32
Capítulo 2 - Simuladores de técnicas de aprendizado de máquina	34
2.1 SNNS (<i>Stuttgart Neural Network Simulator</i>)	34
2.2 LIBSVM (<i>A Library to Support Vector Machine</i>)	39
2.3 WEKA (<i>Waikato Environment for Knowledge Analysis</i>)	41
Capítulo 3 - Bases de dados médicas	44
3.1 Bases de dados	44
3.1.1 Doença inflamatória intestinal	44
3.1.2 Câncer de mama	46
3.1.3 Diabetes	47
3.1.4 Doença do fígado	48
3.1.5 Câncer pulmonar	49
3.1.6 Glândula tireóide	50
Capítulo 4 - Experimentos e resultados	51
4.1 Normalização das bases de dados	51
4.2 Experimentos e resultados utilizando SVM	52
4.3 Experimentos e resultados utilizando kNN	55
4.4 Experimentos e resultados utilizando MLP- <i>backpropagation</i>	56
4.5 Experimentos e resultados utilizando RBF-DDA	57

4.6	Comparações entre as técnicas utilizadas	59
	Capítulo 5 - Conclusões e Trabalhos Futuros	67
	Apêndice A - Glossário	72

Índice de Figuras

Figura1. Constituintes de um neurônio biológico	13
Figura2. Rede de Perceptrons proposta por Rosenblatt	14
Figura3. Redes ADALINE e MADALINE	14
Figura4. Neurônio de McCulloch e Pitts	15
Figura5. Pseudo-código do algoritmo de aprendizagem do perceptron	16
Figura6. Função Linear	17
Figura7. Função rampa	17
Figura8. Função de ativação degrau	18
Figura9. Função sigmóide	18
Figura10. Exemplos de estruturas de RNAs	19
Figura11. Aprendizado supervisionado	20
Figura12. Exemplo de arquitetura de uma rede MLP	20
Figura13. Fluxo do algoritmo de retropropagação de retorno	22
Figura14. Pseudo-código do algoritmo de retropropagação de retorno	23
Figura15. Comportamento das diferentes funções de base radial	24
Figura16. Rede RBF típica	25
Figura17. Exemplo de uma rede RBF-DDA	26
Figura18. Problema de classes conflitantes presente no algoritmo RCE	26
Figura19. Classificação de um novo padrão de entrada pelo algoritmo DDA	27
Figura20. Pseudocódigo do algoritmo DDA	28
Figura21. Exemplo de classificação de padrões com algoritmo DDA	28
Figura22. Mapeamento de características	29
Figura23. Margem geométrica de um ponto x_i e a margem ρ do hiperplano de separação ótimo	30
Figura24. Classificação de padrão utilizando kNN	31
Figura25. Exemplo de validação cruzada em 10 blocos	32
Figura26. Interface gráfica do SNNS	35
Figura27. Criação de rede através do SNNS	36
Figura28. Definição da arquitetura de redes no SNNS	37
Figura29. Exemplo de uma rede RBF-DDA construída através do SNNS	38
Figura30. Painel de manipulação de arquivos do SNNS	38
Figura31. SVMToy antes (a) e após (b) a classificação	39
Figura32. Treinamento e classificação através do SVMTrain e SVMPredict, respectivamente	40

Figura33. Interface gráfica do WEKA	41
Figura34. Interface gráfica da opção Explorer do WEKA	42
Figura35. Classificador MLP presente no WEKA	43
Figura36. Classificador kNN presente no WEKA	43
Figura37. Distribuição percentual das classes na base de dados de doença inflamatória intestinal	44
Figura38. Distribuição percentual das classes na base de dados de câncer de mama	47
Figura39. Distribuição percentual das classes na base de dados Diabetes	48
Figura40. Distribuição percentual das classes na base de dados Doença do Fígado	49
Figura41. Distribuição percentual das classes na base de dados Câncer pulmonar	49
Figura42. Distribuição percentual das classes na base de dados Glândula tireóide	50
Figura43. Comparação entre nº de padrões e taxa de acerto (%) entre as bases de dados de doença inflamatória intestinal e câncer pulmonar	53
Figura44. Comparação entre nº de padrões e taxa de acerto (%) entre as bases de dados de diabetes e doença de fígado	54
Figura45. Comparação entre nº de padrões e taxa de acerto (%) entre as bases de dados de câncer de mama e glândula tireóide	54
Figura46. Visualização 2D da distribuição de dados da base diabetes. Pontos azuis representam teste negativo e os vermelhos, teste positivo	55
Figura47. Visualização 2D da distribuição de dados da base glândula tireóide. Pontos azuis representam Normal, os vermelhos, hipertireoidismo e o verde, hipotireoidismo	56
Figura48. Taxa de acerto (%) versus Nº de épocas	57
Figura49. Diminuição do nº de neurônios da camada oculta da rede RBF-DDA decorrente do aumento do limiar negativo	58
Figura50. Acerto <i>versus</i> θ^-	59
Figura51. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados doença inflamatória intestinal	60
Figura52. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados câncer de mama	61
Figura53. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados diabetes	62
Figura54. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados doença do fígado	63
Figura55. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados câncer pulmonar	64
Figura56. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados glândula tireóide	65

Índice de Tabelas

Tabela 1. Comparação entre redes MLP e RBF	25
Tabela 2. Exemplos de funções kernel utilizadas no algoritmo de aprendizagem de SVMs	30
Tabela 3. Características da base de dados Doença Inflamatória Intestinal	45
Tabela 4. Características da base de dados de câncer de mama	47
Tabela 5. Características da base de dados Diabetes	48
Tabela 6. Características da base de dados Doença do Fígado	49
Tabela 7. Características da base de dados Glândula Tireóide	50
Tabela 8. Características das base de dados presentes no estudo	50
Tabela 9. Codificação das classes das bases de dados	52
Tabela 10. Parâmetros ótimos identificados pelo LIBSVM	52
Tabela 11. Resultados obtidos por SVM nas bases de dados	53
Tabela 12. Classificação das bases de dados utilizando kNN	55
Tabela 13. Melhores resultados obtidos nas bases de dados por redes MLP- <i>Backpropagation</i>	57
Tabela 14. Melhores resultados obtidos nas bases de dados por redes RBF-DDA	58
Tabela 15. Melhores resultados obtidos nas bases de dados	66

Tabela de Siglas

RNA – redes neurais artificiais
SVM – support vector machine (máquina de vetor suporte)
kNN – k-nearest neighbour (técnica do vizinho mais próximo)
ANNIMAB-S – artificial neural networks in medicine and biology society
SNNS – Stuttgart neural network simulator
RBF – radias basis function (função de base radial)
MLP – multy-layer perceptron (perceptron multicamadas)
LIBSVM – library for support vector machines
WEKA – waikato environment for knowledge analysis
DDA – dynamic decay adjustment
MCP – McCulloch e Pitts
IA – inteligência artificial
ADALINE – adaptative linear network
MADALINE – many ADALINE
XOU – ou exclusivo
RCE – restricted coulomb energy
SAD – sistemas de apoio a diagnósticos
IPVR – Institute for Parallel and Distributed High Performance Systems
2D – duas dimensões
3D – três dimensões
URL – uniform resource locator
EUA – Estados Unidos da América
OMS – Organização mundial de saúde
Hg – mercúrio
T3 – teste de captação de resina
TSH – hormônio estimulador de tireoide basal

Agradecimentos

Gostaria de agradecer à todos que contribuíram de alguma forma para que este trabalho se tornasse realidade:

A Deus, a Ele devo tudo que tenho e sou.

Aos meus pais, Petrônio Braga e Lúcia Braga, que me permitiram estar aqui hoje e pelo apoio inconsteste.

Ao meu orientador, Adriano Lorena Inácio de Oliveira, pelo aprendizado, apoio, pela paixão por essa excitante área que é Redes Neurais e pelas cobranças.

Aos meus irmãos, Rodolfo Braga e Petrônio Braga, pela paciência e companheirismo.

A Paloma, minha “co-orientadora”, companheira e ombro amigo.

Ao Dr. Alberto Barros, pelas traduções dos termos médicos, pela disponibilidade e amizade.

Aos meus professores e amigos da POLI, pelo aprendizando e caminhada durante o curso. Sentirei saudades!

Ao meu amigo, Thyago Barbosa, e professores que compõem o DSC, pelo “empréstimo” de seu computador para realização dos experimentos e escrita deste trabalho. Obrigado pela força!

Aos meus familiares e amigos.

Introdução

A proposta de utilização da inteligência artificial no campo da medicina é datada desde a década de 50 [1]. Inúmeras técnicas dessa área da computação têm sido empregadas desde então, tais como: algoritmos genéticos [2], sistemas híbridos [3], lógica difusa [4], redes neurais artificiais [5], dentre outras.

Devido à capacidade de modelagem não-linear, adaptabilidade e possibilidade de aprendizado por meio de exemplos, redes neurais artificiais (RNAs) [6] são extensamente aplicadas em bases de dados médicas complexas [7] para reconhecimento de padrões. A ausência da necessidade de conhecimento prévio sobre o problema a ser solucionado também é um dos principais atrativos dessa técnica.

Não obstante, uma das limitações mais sérias relativas à aplicação de redes neurais é a não-explicabilidade da solução do problema; o que, por consequência, pode interferir na confiabilidade do sistema. No entanto, algoritmos de extração de conhecimento já foram e estão sendo desenvolvidos para abordar esses problemas [8].

A literatura reporta várias aplicações de RNAs na área da medicina. Lisboa [9] fez uma vasta revisão dos benefícios advindos da aplicação dessa técnica em áreas como oncologia, cardiologia, cirurgia, medicina intensiva, etc. Ainda, Baxt [10, 11] contabilizou cerca de 1000 citações de RNAs na literatura biomédica em 1995 e, segundo Barach & Small [12], esse número tenderá a crescer nos próximos anos.

Baseado nas evidências da aplicabilidade de RNAs na medicina [13], realizamos experimentos em seis bases de dados médicas, cinco delas disponíveis no repositório UCI Machine Learning [14] e a outra no Artificial Neural Networks in Medicine and Biology Society (ANNIMAB-S) [15]. As bases, em questão, consideram as seguintes patologias: 1 - Lesão da mama, 2 - Doença inflamatória intestinal, 3 - Diabetes, 4 - Doenças do fígado, 5 - Câncer do Pulmão e 6 - Tireoidismo.

Essa abordagem pode mostrar o potencial de técnicas de aprendizado de máquina na área da medicina. Benefícios advindos desse estudo podem ser considerados significativos, pois abre precedentes para aplicação desde a formação educacional médica [16] até o auxílio no atendimento médico hospitalar [10].

Neste trabalho, consideramos as seguintes técnicas de aprendizado de máquina: redes neurais artificiais (RNAs), máquinas de vetor suporte (SVMs) e técnica do vizinho mais próximo (kNN).

Nos experimentos, foram utilizados três simuladores: o SNNS (*Stuttgart Neural Network Simulator*) [17] para simular redes neurais do tipo RBF [18, 19, 20] e *MultiLayer Perceptron* (MLP) [6], o LIBSVM (*Library For Support Vector Machines*) [21, 22, 23] e o WEKA (*Waikato Environment For Knowledge Analysis*) [24]. O objetivo é comparar o algoritmo RBF-DDA [20, 21, 25] com Máquinas de Vetor Suporte (SVM), além de redes Perceptrons de Múltiplas Camadas (MLP), esta usando o algoritmo *backpropagation* [6], bem como a técnica dos vizinhos mais próximos (kNN).

O presente trabalho encontra-se estruturado em cinco capítulos. O primeiro deles explana os conceitos envolvidos nas técnicas de aprendizado de máquina, bem como faz uma revisão bibliográfica da aplicação de algumas dessas técnicas na área médica. São explicados os algoritmos utilizados nos experimentos, ou seja, RBF-DDA, SVM, kNN e MLP-*Backpropagation*. O capítulo também contém uma revisão bibliográfica das aplicações de técnicas de aprendizagem de máquina na área da medicina

O segundo capítulo apresenta os simuladores utilizados na realização dos experimentos. Para a aplicação do algoritmo RBF-DDA foi utilizado o SNNS, dada sua versatilidade e simplicidade. O simulador usado para aplicação de máquinas de vetor suporte foi o LIBSVM, pelos mesmos motivos da ferramenta anterior. Para os experimentos com MLP-*Backpropagation* e kNN, utilizou-se o WEKA.

O terceiro capítulo apresenta as bases de dados utilizadas nos experimentos, enquanto o quarto capítulo alude aos resultados obtidos em cada uma dessas bases quando da aplicação das técnicas de aprendizado de máquina. A comparação entre tais técnicas considerou o desempenho de classificação, complexidade das redes geradas, tempo de treinamento e tempo de classificação.

Finalmente, o quinto capítulo contém as conclusões e sugestões para trabalhos futuros.

Um glossário, adicionado como apêndice, de alguns termos da área de inteligência artificial encontra-se na seção após o capítulo 5.

Capítulo 1

Técnicas de aprendizado de máquina

No corrente capítulo, são apresentados conceitos referentes a Redes Neurais – RBF-DDA e MLP-*backpropagation*, Máquinas de vetor suporte e técnica do vizinho mais próximo. Validação cruzada e Normalização são também conceituados. Uma revisão da literatura médica relacionada a aprendizagem de máquina também é realizada.

1.1. Redes Neurais Artificiais

Uma rede neural artificial (RNA) consiste de um conjunto de unidades processadoras (neurônios ou nodos) que simulam os neurônios biológicos. Tais unidades são interconectadas através de canais de comunicação, aos quais estão associados pesos (análogo às conexões sinápticas do sistema nervoso) de tal forma que eles permitem sinais serem enviados e recebidos pelos nodos. O poder computacional de uma rede neural artificial não advém da complexidade de cada unidade de processamento, mas da densidade e complexidade das interconexões. O atrativo maior das RNAs reside na capacidade de aprendizado e generalização, bem como na flexibilidade e dinâmica inerentes às suas estruturas. Seu processamento paralelo também contribui para sua utilização em problemas onde métodos tradicionais não têm bom desempenho, como reconhecimento de padrões.

1.1.1 Neurônio biológico

O sistema nervoso humano é constituído por grande quantidade de células, dentre elas, se destacam os neurônios. São eles os responsáveis pelo comportamento e funcionamento do aprendizado, raciocínio e de todo processamento inteligente do cérebro. Os neurônios são formados pelos dendritos, que são os terminais de entrada, e pelos axônios, que por sua vez, são os terminais de saída. Através de um processamento físico-químico, no qual os sinais neurônio-

neurônio são enviados por meio de sinapses nervosas (área de contato entre neurônios), é que ocorre o processo de aprendizagem. A Figura 1 ilustra os constituintes de um neurônio.

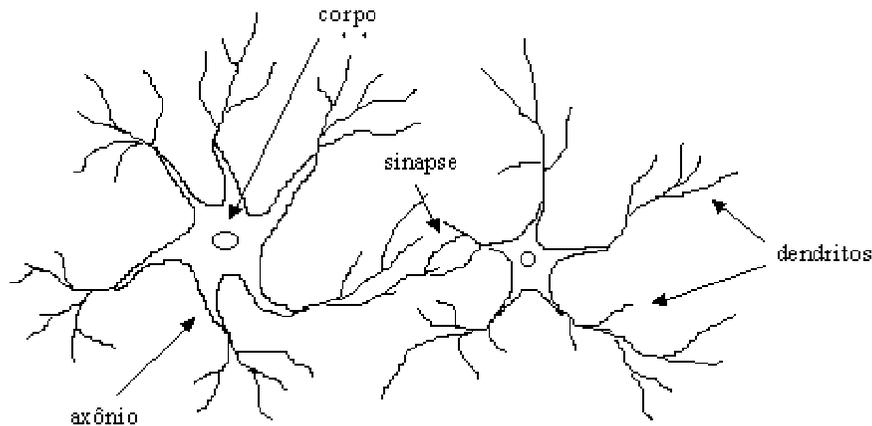


Figura 1. Constituintes de um neurônio biológico

As sinapses são responsáveis pela união funcional entre neurônios, originando as redes neurais, além de controlar o fluxo de impulsos entre eles. Tais fluxos sinápticos são variáveis, e é nesse processo dinâmico que a adaptabilidade neuronal reside. Os sinais advindos dos neurônios pré-sinápticos são enviados para o neurônio pós-sináptico, no qual ocorre as comparações com outros sinais recebidos por este. Se a quantidade desses sinais ultrapassa o limiar de ativação, o neurônio produz um impulso que é transmitido para os neurônios seguintes. A realização de funções complexas pelo cérebro humano se dá pelo processamento massivo paralelo de todos os neurônios ali encontrados (cerca de 10^{11}).

1.1.2 Histórico

As primeiras referências sobre neurocomputação datam de 1943, quando o primeiro modelo matemático de um neurônio foi proposto por McCulloch e Pitts (MCP) [26].

No entanto, no modelo neuronal proposto, encontravam-se alguns problemas associados, tais como a impossibilidade de tratar problemas que não fossem linearmente separáveis, bem como o fato dos pesos serem fixos, o que dificultava seriamente a representação dinâmica que se pretendia simular. Após a detecção de tais fatores limitantes, muitos outros modelos foram propostos a fim de evitá-los.

Sendo assim, muitos outros artigos e livros foram publicados, mas pouca relevância foram dados a eles. Até que em 1949, Hebb demonstrou que a capacidade de aprendizagem está diretamente relacionada com as conexões sinápticas, o que motivou muitos outros pesquisadores a perseguirem o mesmo raciocínio.

Poucos anos depois, em 1951, foi construído o primeiro neuro computador, chamado *SNARK*, por Marvin Minsky. A máquina era capaz de ajustar os seus pesos automaticamente, apesar de nunca ter executado qualquer função relevante. Contudo, o *SNARK* serviu como instrumento norteador para novos estudos.

Em 1956, foram concebidos dois paradigmas no campo da inteligência artificial (IA), o conexionista e o simbólico. O primeiro defende que, construindo-se um sistema que seja capaz de simular a estrutura do cérebro humano, ele pode ser considerado inteligente. O paradigma simbólico propõe a simulação do comportamento inteligente humano desconsiderando os mecanismos responsáveis por tal.

O primeiro neuro computador eficiente, conhecido como *Mark I Perceptron*, foi desenvolvido por Frank Rosenblatt, Charles Wightman e colaboradores entre meados de 1957 e 1958. Nesse computador estava presente o resultado das pesquisas de Rosenblatt: o modelo Perceptron, no qual os neurônios são distribuídos em camadas de entrada e saída, além dos pesos das conexões sinápticas serem adaptáveis. O perceptron encontra-se ilustrado na Figura 2 abaixo.

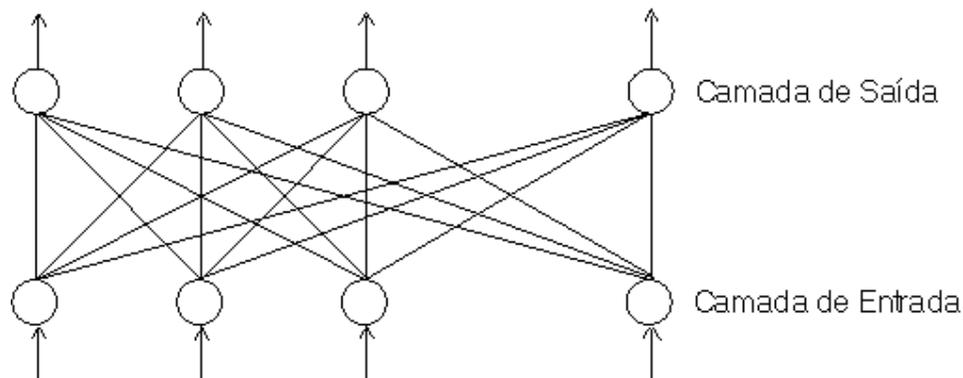


Figura 2. Rede de Perceptrons proposta por Rosenblatt

Em 1960, Bernard Widrow e Hoff propuseram dois novos modelos de neurônio, chamados *ADALINE* (*ADAPtative Linear NETwork*) e *MADALINE* (*MANy ADALINE*). O ADALINE/MADALINE utiliza saídas analógicas em uma arquitetura de três camadas (vide Figura 3).

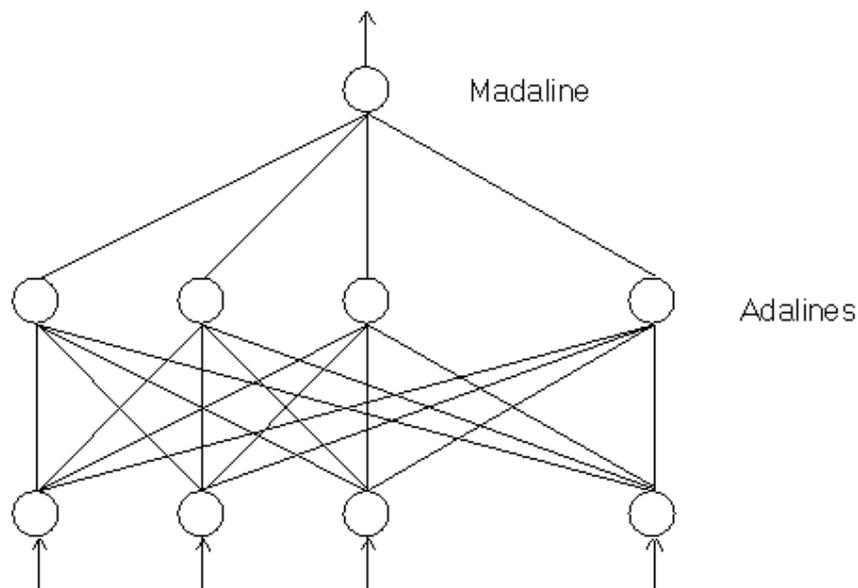


Figura 3. Redes ADALINE e MADALINE

Segundo historiadores e pesquisadores, nos anos 60 e 70, os estudos na área de neurocomputação foi silenciado, devido às várias críticas de Minsky e Papert, que acarretaram no fim das verbas destinadas à pesquisa nesse campo.

A retomada nos estudos é datada de 1982, quando Hopfield, físico e biólogo de renome mundial, relatou a aplicação de redes simétricas para otimização, através de um algoritmo de aprendizagem que estabilizava uma rede binária simétrica com realimentação [26].

Finalmente, em 1986, David Rumelhart e James McClelland publicaram o livro intitulado *Parallel Distributed Processing*, no qual o algoritmo *Backpropagation* foi introduzido, contrariando a visão pessimista de Minsky e Papert. A partir dessa década, houve uma retomada de interesses pelas RNAs no cenário mundial [26].

1.1.3 Características gerais

Segundo Braga *et al* [26], redes neurais artificiais são sistemas paralelos constituídos de unidades (neurônios ou nodos) que processam funções matemáticas quaisquer. A estrutura desses sistemas são dispostas em camadas e interligadas através de conexões. Associadas às conexões, geralmente, encontram-se os pesos, os quais são os responsáveis pelo conhecimento armazenado na rede.

A motivação para a aplicação de redes neurais nas mais diversas áreas advém de sua habilidade para executar computação distribuída, tolerar entradas ruidosas e aprender. Em RNAs, o processo de aquisição de conhecimento passa por uma fase de aprendizagem, na qual exemplos são apresentados à rede, que é capaz de extrair as características necessárias para geração das respostas do problema. A capacidade de generalização da informação aprendida talvez seja o principal atrativo da aplicação da técnica na solução de problemas. Entende-se por generalização a capacidade que a rede tem de apresentar respostas coerentes para dados não conhecidos, após o aprendizado através de um conjunto de exemplos.

1.1.4 Unidades em redes neurais

A unidade constituinte de uma rede neural é o neurônio. É ele o responsável pelo processamento do conjunto de exemplos apresentados à rede. O neurônio artificial mais simples que a literatura reporta é o nodo MCP, e nos valeremos dele para uma ilustração do que um neurônio de uma RNA é capaz de realizar. A Figura 4 abaixo retrata o modelo neuronal MCP.

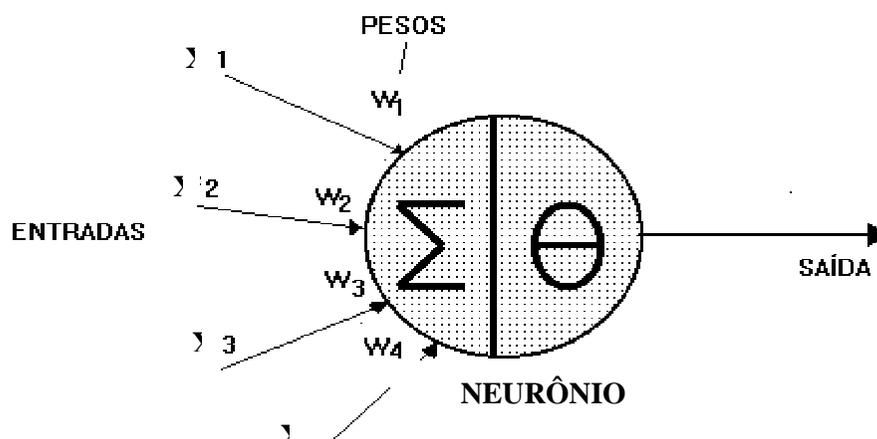


Figura 4. Neurônio de McCulloch e Pitts

Basicamente, o funcionamento de tal neurônio pode ser assim explicado:

- Um vetor X (X_1, X_2, \dots, X_i) é apresentado como entrada da rede.
- Cada elemento do vetor X é multiplicado por um elemento do vetor de pesos W (W_1, W_2, \dots, W_i) correspondente.
- O somatório é então computado e comparado com um limiar de ativação, Θ (*Threshold*)

$$\sum_{i=1}^n x_i w_i \geq \theta$$

Saída = 1, se o somatório for maior que Θ .

Saída = 0, se o somatório for menor que Θ .

Em pseudo-código [6] o algoritmo de aprendizado do perceptron é:

função APRENDIZAGEM-DE-PERCEPTRON(*exemplos, rede*) **retorna** uma hipótese de perceptrons
entrada: *exemplos*, um conjunto de exemplos, cada um com entrada $\mathbf{x} = x_1, \dots, x_n$ e saída y
rede, um perceptron com pesos $W_j, j = 0 \dots n$ e função de ativação g

repita

para cada e **em** *exemplos* **faça**

$$in \leftarrow \sum_{j=0}^n W_j x_j[e]$$

$$Err \leftarrow y[e] - g(in)$$

$$W_j \leftarrow W_j + \alpha \times Err \times g'(in) \times x_j[e]$$

até algum critério de parada ser satisfeito

retornar HIPÓTESE-DA-REDE-NEURAL(*rede*)

Figura 5. Pseudo-código do algoritmo de aprendizagem do perceptron

No pseudo-código da Figura 5, Err é o erro obtido pela diferença entre a saída desejada e $g(in)$ é a saída corrente da rede, α é a taxa de aprendizagem e $g'(in)$ é a derivada da função de ativação.

1.1.5 Funções de ativação

Muitas outras funções de ativação foram propostas na literatura. As mais difundidas na literatura [6, 26] são:

1. Função de ativação linear: é aquela definida pela equação abaixo (ver Figura 5)

$$y = ax$$

onde a é um número real qualquer, x é a entrada e y a saída.

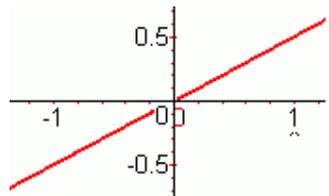


Figura 6. Função linear

2. Função de ativação rampa: um caso particular da função linear, a qual é restringida para produzir valores constantes em um intervalo $[-z, +z]$, ou seja, os valores máximo e mínimo da saída são $+z$ e $-z$, respectivamente. Um exemplo de uma função rampa é a função abaixo.

$$f(x) = \begin{cases} 1 & \text{se } x \geq \frac{1}{2} \\ x, & -\frac{1}{2} < x < \frac{1}{2} \\ 0 & \text{se } x < -\frac{1}{2} \end{cases}$$

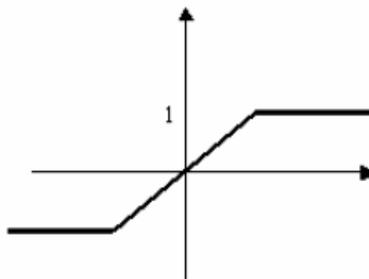


Figura 7. Função rampa

3. Função de ativação degrau: também conhecida como função limiar, sua saída é 1 quando a entrada é positiva e 0 em caso contrário.

$$f(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases}$$

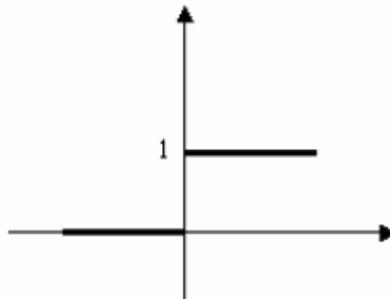


Figura 8. Função de ativação degrau

4. Função de ativação sigmoide: também conhecida como função logística, possui a vantagem de ser diferenciável, o que é de extrema relevância para os algoritmos de aprendizagem. A função logística é dada pela equação abaixo, onde α determina a suavidade (inclinação) da curva. A Figura 9 ilustra graficamente a função sigmoide.

$$f(x) = \frac{1}{1 + \exp(-\alpha x)}$$



Figura 9. Função sigmoide

1.1.6 Estruturas de redes neurais

Com relação à arquitetura de redes, temos que as RNAs podem diferir quanto ao número de camadas, número de neurônios em cada camada, tipo de conexão entre os neurônios e topologia da rede.

O número de camadas presente em uma RNA é um importante parâmetro, porque a sua quantidade está intimamente ligada com a eficiência da mesma, bem como com o tipo de problema que se pretende solucionar. Por exemplo, redes com apenas uma camada de neurônios MCP só resolvem problemas linearmente separáveis (E, OU, NÃO). Problemas como o XOU (ou exclusivo) não podem ser solucionados com uma rede de apenas uma camada. A Figura 10 ilustra vários tipos de arquiteturas de RNAs.

As arquiteturas de RNAs podem apresentar uma (Figura 10 - a, e) ou mais camadas (Figura 10 - b, c, d), nas quais a diferença reside na quantidade de neurônios presente entre as camadas de entrada e saída.

Relativo a conectividade, as redes podem ser totalmente (Figura 10 - a, e) ou parcialmente conectadas (Figura 10 - b, c, d).

Ainda, as redes podem ser classificadas em acíclicas ou redes de alimentação direta e redes cíclicas ou redes recorrentes.

1. Rede de alimentação direta: são aquelas em que não há conexões entre neurônios da n -ésima camadas com quaisquer outros neurônios de camadas com índice inferior a n (Figura 10 – a, b, c).
2. Redes recorrentes: são RNAs que utilizam suas saídas para alimentar de volta suas próprias entradas (Figura 10 – d, e). São geralmente utilizadas para simular processos caóticos, no qual um comportamento dinâmico seja necessário. Sendo assim, redes recorrentes utilizam memória de curto prazo.

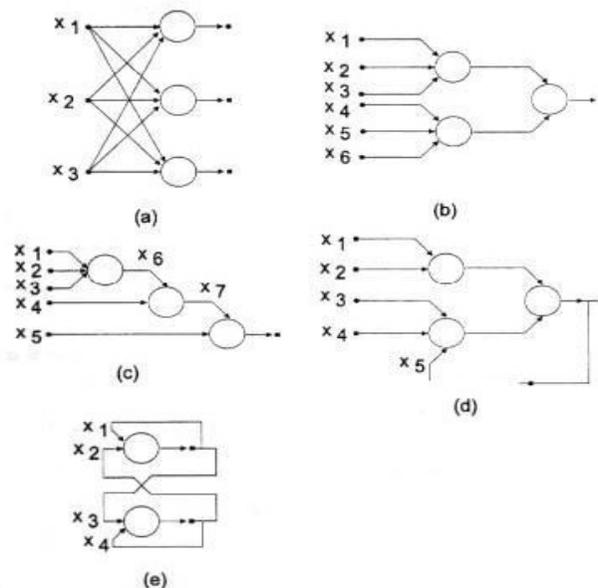


Figura 10. Exemplos de estruturas de RNAs

1.1.7 Formas de aprendizado

Como já citado, RNAs aprendem a partir de um conjunto de exemplos fornecidos ao longo do treinamento. O responsável pela aquisição do conhecimento é o algoritmo de aprendizado, o qual é um conjunto de regras bem definidas para a solução de um problema de aprendizado. Esse conjunto de regras consiste num processo de ajuste dos parâmetros da rede neural, os quais representam o conhecimento adquirido. A área de aprendizagem de máquina costuma distinguir três maneiras do conhecimento ser adquirido: aprendizagem supervisionada, não-supervisionada e por reforço.

1. Aprendizagem supervisionada: a entrada e saída desejadas são fornecidos à rede por um supervisor. O processo de aprendizagem ocorre da seguinte maneira: um padrão de entrada é fornecido à rede e sua saída atual é comparada com a saída desejada, e essa diferença é aplicada no ajuste das variáveis (pesos) da rede. O processo de minimização do erro é que garante a eficiência da RNA. No entanto, o sucesso do aprendizado supervisionado tem relação com a presença do supervisor. Na sua ausência, a rede é

incapaz de progredir em apresentar respostas para o problema. A Figura 11 [26] mostra um diagrama elucidando o aprendizado supervisionado.

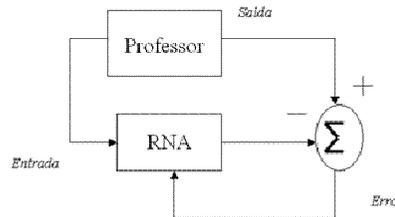


Figura 11. Aprendizado supervisionado

2. Aprendizado não-supervisionado (auto-organização): não utiliza um agente externo que indica a resposta desejada para um determinado padrão de entrada. Utiliza-se entretanto, de padrões redundantes para que a rede responda de maneira semelhante.
3. Aprendizado por reforço: diferente do algoritmo supervisionado, não é fornecida à rede a resposta desejada, a informação apresentada à ela é se a saída está correta ou não. Por exemplo, uma multa causada por uma infração de trânsito fornece ao agente alguma indicação de que seu comportamento é errôneo. Sendo assim, a aprendizagem por reforço considera o funcionamento do ambiente.

1.1.8 Redes multilayer perceptron

As redes neurais de alimentação direta de várias camadas, também conhecidas como redes MLP (Multilayer perceptron) [6, 26], são compostas por neurônios MCP com função de ativação do tipo sigmóide. A maior diferença entre redes MLP e o Perceptron é a quantidade de camadas presentes na arquitetura.

As redes MLP introduzem o conceito de camada escondida ou oculta ou intermediária, a qual seria a camada presente entre as camadas de entrada e saída. A Figura 12 ilustra um exemplo de uma rede MLP.

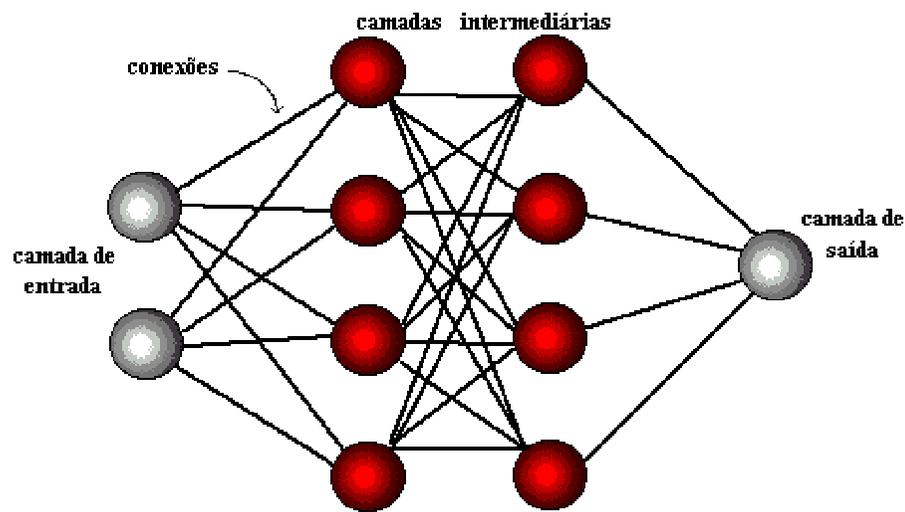


Figura 12. Exemplo de arquitetura de uma rede MLP

A camada de entrada é onde ocorre a apresentação dos dados de entrada (conjunto de exemplos). A camada oculta é responsável pela maior parte do processamento, enquanto a camada de saída é incumbida de apresentar o(s) resultado(s).

O maior benefício advindo da adição da camada oculta é a possibilidade da resolução de problemas não-linearmente separáveis. De fato, os nodos dessa camada são capazes de traçar retas no espaço de entrada formando regiões convexas, de tal forma que se torna possível a resolução daqueles problemas.

Cybenko [27, 28], pesquisador da área, realizou um estudo referente ao número de camadas ocultas e o resultado obtido foi que uma camada intermediária é suficiente para aproximar qualquer função contínua, enquanto duas camadas são capazes de aproximar qualquer função matemática. Apesar do resultado da pesquisa, há certa resistência da comunidade científica quanto ao uso de muitas camadas ocultas, porque o erro propagado para as camadas anteriores tende a ser menos relevante quanto maior for o número de camadas intermediárias. A literatura reporta que uma única camada escondida é suficiente ao longo de experimentos.

Outro ponto relevante na determinação da arquitetura de redes MLP é o número de nodos presentes na(s) camada(s) oculta(s). Estudiosos apontam que não há qualquer regra para a determinação dessa quantidade. Na verdade, o número de nodos depende dos mais variados fatores, tais como [26]:

- Número de exemplos de treinamento;
- Quantidade de ruídos presentes nos exemplos;
- Complexidade da função a ser aprendida;
- Distribuição estatística dos dados de treinamento.

Esses fatores devem ser bem observados para que se evite o problema da superadaptação (*overfitting*), definido como o ato da rede se ajustar (“decorar”) aos padrões de treinamento, reduzindo consideravelmente o poder de generalização.

Na verdade, não há consenso na literatura científica a respeito da determinação dos parâmetros de uma rede MLP, ou seja, número de camadas e número de neurônios presentes nessas camadas. Esses valores tem que ser encontrados através de uma quantidade de testes cansativos, sendo essa, talvez, a maior desvantagem dessa rede.

Os algoritmos de aprendizagem para redes MLP são semelhantes ao algoritmo de aprendizagem do perceptron, com a diferença que pode-se ter várias saídas (e assim teremos um vetor de saída $\mathbf{h}_w(\mathbf{x})$ ao invés de apenas uma saída) e cada padrão de entrada possuir um vetor de saída \mathbf{y} . Muitos algoritmos de aprendizagem foram propostos para o treinamento de redes MLP, sendo o mais conhecido o algoritmo *backpropagation*, bem como variações do mesmo [30 - 35], no entanto em nosso trabalho nos concentraremos apenas no algoritmo de original.

O algoritmo *backpropagation* é do tipo supervisionado, ou seja, pares (entrada, saída desejada) são apresentados por um supervisor ao longo do treinamento. Esse ocorre em duas fases: *forward* e *backward* (Figura 13). A fase *forward* é utilizada para definir a saída para um dado padrão de entrada, enquanto a *backward* é usada para atualizar os pesos da rede através da saída desejada e da saída corrente.

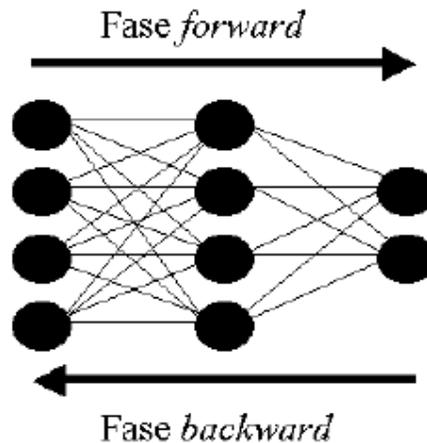


Figura 13. Fluxo do algoritmo *backpropagation*

Na Figura 14 encontra-se ilustrado o pseudo-código do algoritmo *backpropagation* [29]. No pseudo-código temos que a_j é o neurônio j da camada de entrada, M é o número total de camadas (camada (s) oculta (s) + camada de saída), a_i é o neurônio i das M camadas, α é a taxa de aprendizado e $g'(in)$ é a derivada da função de ativação. A classificação utiliza a abordagem *winner-takes-all* [6], na qual a unidade de saída com maior ativação apresenta o resultado. Em suma, o algoritmo recebe como entrada dois vetores, um de entrada (x) e outro de saída (y). Ao j -ésimo nodo da camada de entrada é associado o índice j do vetor x de entrada. Um vetor in é então utilizado para armazenar o resultado do produto escalar do vetor de entrada x e o vetor de pesos W (iniciado com valores aleatórios). A função de ativação, geralmente sigmóide (diferenciável), é aplicada ao vetor in . Após esses processamentos, calcula-se o erro, o qual é obtido pela diferença entre a resposta desejada (vetor y) e a saída corrente (vetor a_i). Terminada a fase *forward*, a fase *backward* é então inicializada, a qual consiste no ajuste dos pesos das conexões entre as camadas de acordo com a taxa de aprendizado e com o erro obtido (Δ_i).

função APRENDIZAGEM-POR-PROPAGAÇÃO-DE-RETORNO(*exemplos, rede*) **retorna** uma rede neural
entradas: *exemplos*, um conjunto de exemplos, cada um com vetor de entrada \mathbf{x} e vetor de saída \mathbf{y}
rede, uma rede de várias camadas com L camadas, pesos $W_{j,i}$, função de ativação g

repita

para cada e em *exemplos* faça
 para cada nó j na camada de entrada faça $a_j \leftarrow x_j[e]$
 para $\ell = 2$ até M faça
 $in_i \leftarrow \sum_j W_{j,i} a_j$
 $a_i \leftarrow g(in_i)$
 para cada nó i na camada de saída faça
 $\Delta_i \leftarrow g'(in_i) \times (y_i[e] - a_i)$
 para $\ell = M - 1$ até 1 faça
 para cada nó j na camada ℓ faça
 $\Delta_j \leftarrow g'(in_j) \sum_i W_{j,i} \Delta_i$
 para cada nó i na camada $\ell + 1$ faça
 $W_{j,i} \leftarrow W_{j,i} + \alpha \times a_j \times \Delta_i$
até algum critério de parada ser satisfeito
retornar HIPÓTESE-DA-REDE-NEURAL(*rede*)

Figura 14. Pseudo-código do algoritmo de retropropagação de retorno

Para evitar o problema de superadaptação, é relevante que se utilize algum critério de parada. O método de parada propõe que o treinamento de uma rede seja abortado antes que ela perca o poder de generalização. Para que isso ocorra, o conjunto de exemplos deve ser dividido em dois conjuntos, um de treinamento e outro de validação. Nesse trabalho, o critério de parada escolhido para o treinamento foi o GL5 [36], o qual utiliza a diferença entre o erro de validação corrente e o erro de validação mínimo, sendo aplicada uma divisão pelo erro de validação mínimo ao resultado dessa subtração. Para o critério GL5, quando a resposta das operações citadas for maior ou igual a 5 por cento, o treinamento é abortado. A fórmula do critério GL5 encontra-se retradada a seguir.

$$GL(t) = 100 \times \left(\frac{E_{val}(t)}{E_{opt}(t)} - 1 \right)$$

Onde $E_{opt}(t)$ é o erro de validação mínimo obtido durante o treinamento até a época t , e o $E_{val}(t)$ é o erro de validação na época t .

1.1.9 Redes de função de base radial

Nesse tipo de rede, os neurônios da camada escondida processam funções de base radiais. Essas são uma classe especial de funções, com a característica de que a resposta cresce ou decresce a partir de um ponto central, como mostra a Figura 15, onde o ponto central é dado pelo par (x, h) . As funções de base radial mais típicas são

- Função Gaussiana: $f(u) = \exp\left(\frac{-v^2}{\sigma_i^2}\right)$
- Função multiquadrática: $f(u) = \sqrt{v^2 + \sigma^2}$
- Função *thin-plate-spline*: $f(u) = v^2 \log(v)$

Onde $v = \|x - \mu\|$ é a distância Euclidiana, x o vetor de entrada, e μ e σ representam respectivamente o centro e a largura da função radial. A distância euclidiana ($dist_e$) entre dois pontos, p e q , cujas coordenadas são (i,j) e (x,y) , respectivamente, é dada por:

$$dist_e(p, q) = dist_e((i, j), (x, y)) = \sqrt{(i - x)^2 + (j - y)^2}$$

Como pode se observar nas equações acima, as redes RBF utilizam em sua função de ativação a distância Euclidiana entre os vetores de entrada e de peso.

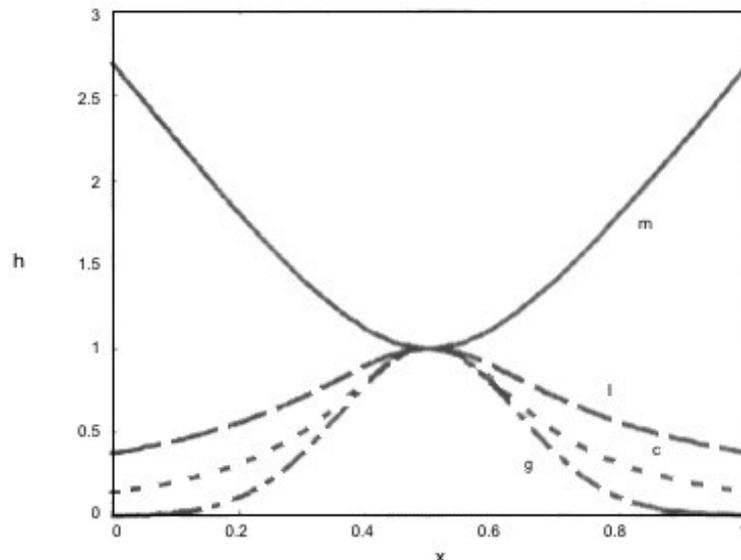


Figura 15. Comportamento das diferentes funções de base radial

A arquitetura de uma rede de função de base radial envolve três camadas com responsabilidades distintas. A camada de entrada apresenta as variáveis de entrada para a camada escondida. A segunda camada, a única camada escondida da rede [6], é responsável pela transformação não-linear do espaço de entrada para um espaço de alta dimensionalidade. Cada neurônio da camada escondida representa uma função radial, a qual define uma hiperelipsóide no espaço de padrões de entrada. A camada de saída é linear, classificando os padrões recebidos da camada escondida. A Figura 16 ilustra uma arquitetura típica de uma rede RBF.

Comparando-se redes MLP com redes RBF, podemos notar muitas diferenças. A Tabela 1 [26] resume as diferenças entre essas duas redes.

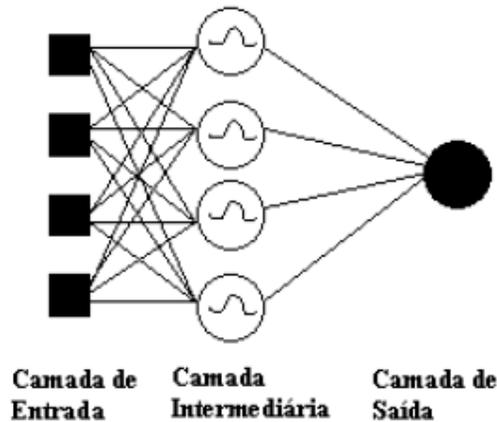


Figura 16. Rede RBF típica

Tabela 1. Comparação entre redes MLP e RBF

Rede MLP versus Rede RBF	
MLP	RBF
Pode ter mais de uma camada escondida	Só possui uma camada escondida
Neurônios da(s) camada(s) escondida(s) e da camada de saída têm funções semelhantes	Neurônios da camada escondida têm funções diferentes dos neurônios da camada de saída
Entrada da função de ativação é o produto escalar dos vetores de entrada e de pesos	Entrada da função de ativação é a distância Euclidiana entre os vetores de entrada e de pesos
Separa padrões de entrada com hiperplanos	Separa padrões de entrada com hiperelipsóides
Melhor em problemas complexos	Melhor em problemas bem-definidos
Constrói aproximadores globais para mapeamento entrada-saída	Constrói aproximadores locais para mapeamento entrada-saída

Com relação ao algoritmo de treinamento de redes RBF, escolheu-se trabalhar com o DDA (*Dynamic Decay Adjustment*) [18], pois o mesmo provê um treinamento construtivo, ou seja, o número de neurônios escondidos é automaticamente determinado ao longo do treinamento. Tais neurônios utilizam função de ativação gaussiana. Nas redes RBF-DDA, cada neurônio da camada de saída representa uma classe. A classificação utiliza a abordagem *winner-takes-all*. O algoritmo possui outra característica motivante, velocidade. O número de épocas necessárias para o treinamento raramente ultrapassa o valor de 5 (cinco). Nas redes RBF-DDA, os nodos da camada de entrada são totalmente conectados com os nodos da camada oculta, que, por sua vez, são conectados a apenas um nodo da camada de saída. A Figura 17 mostra um exemplo de uma arquitetura RBF-DDA. Cada conexão dos nodos da camada escondida com os nodos da camada de saída tem um peso A_i associado. Nodos de saída utilizam a função de ativação linear abaixo.

$$f(\vec{x}) = \sum_{i=1}^m A_i \times R_i(\vec{x})$$

Onde m representa o número de neurônios conectados ao nodo de saída.

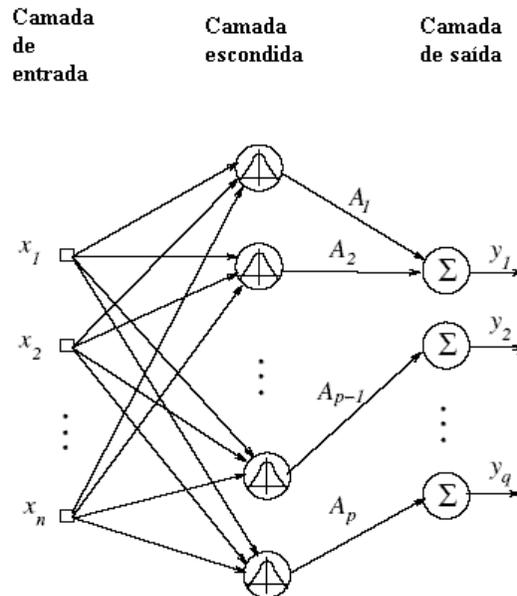


Figura 17. Exemplo de uma rede RBF-DDA

Conforme já citado, o algoritmo propõe um treinamento construtivo. Na fase inicial, a camada escondida encontra-se vazia. Os centros das RBFs, r_i , e suas larguras, σ_i são determinados durante o treinamento, bem como o valor dos pesos A_i .

O DDA utiliza dois parâmetros para tomar a decisão de adicionar ou não um neurônio a camada escondida. O θ^+ (limiar positivo) é utilizado para determinar se um novo protótipo (neurônio) deve ser adicionado a rede. O θ^- (limiar negativo) é usado como limite superior para a ativação de classes conflitantes, problema encontrado no algoritmo RCE (*restricted coulomb energy*) [37] que motivou o desenvolvimento do DDA (vide Figura 18). A Figura 19 ilustra graficamente o uso desses dois limiares para uma correta classificação de um dado padrão de entrada. No caso, o novo padrão x está acima do limiar positivo, $R_B(x)$, e que está abaixo do limiar negativo $R_A(x)$, indicando que esse padrão é da classe B, já que a abordagem utilizada é a *winner-takes-all*.

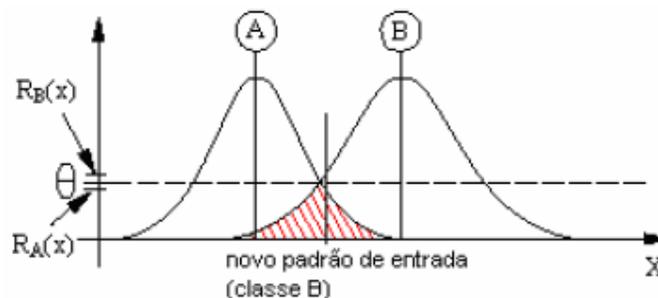


Figura 18. Problema de classes conflitantes presente no algoritmo RCE

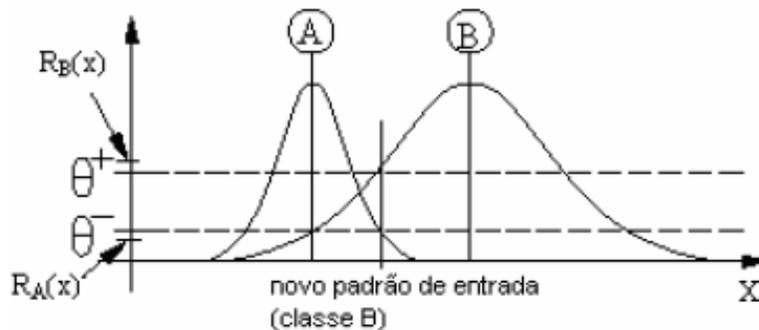


Figura 19. Classificação de um novo padrão de entrada pelo algoritmo DDA

O pseudocódigo a seguir (Figura 20) mostra todo o passo a passo inerente ao algoritmo de treinamento DDA.

O primeiro passo é responsável pela inicialização dos pesos das conexões entre os neurônios da camada escondida e os neurônios da camada de saída.

No segundo passo, os padrões de treinamento são fornecidos à rede. Para cada um deles, se o seu valor de ativação for maior ou igual a θ^+ o peso do protótipo mais próximo ao padrão é incrementado (passo 3), caso contrário, um novo protótipo é introduzido com o centro no novo padrão (passos 4 e 5).

O último passo do algoritmo consiste na redução da largura dos protótipos conflitantes para produzir valores menores que o θ^- para o padrão de treinamento \vec{x} .

A Figura 21 demonstra a aplicação do algoritmo na classificação de padrões. Na Figura 21 (a), um novo padrão da classe A é encontrado e uma gaussiana é adicionada para cobrir o padrão. Na Figura 21 (b), um padrão da classe B é identificado e outra gaussiana é adicionada para cobrir o padrão da classe B, e então é aplicada uma redução na largura da gaussiana da classe A. Ainda, na Figura 21 (c), outro padrão da classe B é encontrado, forçando o incremento da gaussiana da RBF que cobre o padrão B e a redução da largura da RBF da classe A. Finalmente, na Figura 21 (d), outro padrão da classe A é identificado e outra RBF é adicionada para cobrir esse padrão. Note que nesse último passo, houve necessidade da adição de outra RBF, desde que a intersecção do centro desse novo padrão com a gaussiana A já existente é menor que o θ^+ .

```

1: //inicializa os pesos:
   FORALL protótipos  $p_i^k$  DO
      $A_i^k = 0$ 
ENDFOR
2: //Treina uma época completa
   FORALL padrões de treinamento  $(\vec{x}, c)$  DO
     IF  $\exists p_i^c : R_i^c(\vec{x}) \geq \theta^+$  THEN
3:        $A_i^c += 1.0$ 
     ELSE
4:       “//commit”: introduz um novo protótipo
       Adiciona novo protótipo  $p_{m_c+1}^c$  com:
          $\vec{r}_{m_c+1}^c = \vec{x}$ 
          $A_{m_c+1}^c = 1.0$ 
          $m_c += 1$ 
5:        $\sigma_{m_c+1}^c = \max_{k \neq c, 1 \leq j \leq m_k} \{\sigma : R_{m_c+1}^c(\vec{r}_j^k) < \theta^-\}$ 
     ENDIF
6:       “//shrink”: ajustar protótipos conflitantes
       FORALL  $k \neq c, 1 \leq j \leq m_k$  DO
          $\sigma_j^k = \max \{\sigma : R_j^k(\vec{x}) < \theta^-\}$ 
       ENDFOR
     ENDFOR
  
```

Figura 20. Pseudocódigo do algoritmo DDA

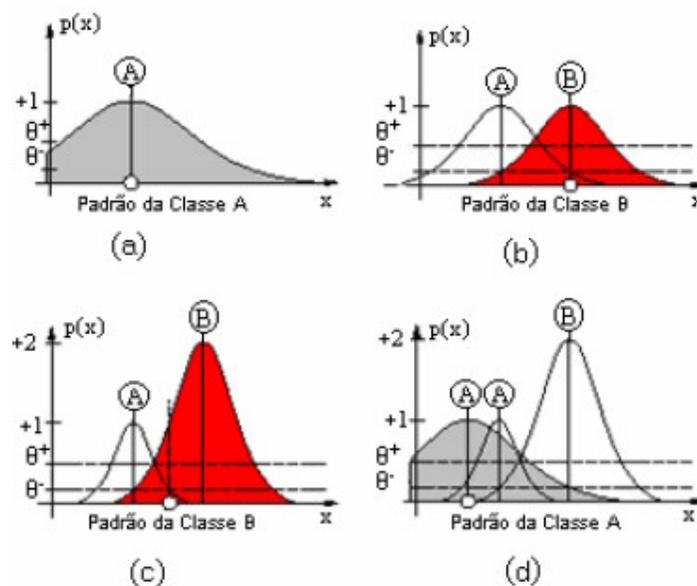


Figura 21. Exemplo de classificação de padrões com algoritmo DDA

1.2 Máquina de vetor suporte (SVM)

Máquinas de vetores suporte (SVMs) ou máquinas de núcleo, foram recentemente propostas como outra possível técnica para classificação de padrões [37]. A idéia proposta por Vapnik e colaboradores está superando muitas das diversas técnicas de aprendizado de máquina [38]. A explicação para a obtenção desses resultados é que, enquanto as técnicas tradicionais tendem otimizar o desempenho sobre o conjunto de treinamento, SVMs minimizam a probabilidade de classificar erroneamente padrões ainda não vistos através de uma distribuição de probabilidade fixa e desconhecida. Existem pelo menos dois motivos que tornam SVMs atrativas:

1. Habilidade de condensar a informação presente no conjunto de treinamento;
2. Uso de superfícies de decisão de dimensão baixa

A Figura 22 mostra um espaço de entrada bidimensional, com duas classes representadas por \bullet e \diamond . Observando a Figura, depreende-se que não existe nenhum separador linear para esse problema. No entanto, se mapearmos cada vetor de entrada x para um novo vetor de características $F(x)$, é possível construir o hiperplano de separação. Então, o que uma máquina de vetor suporte faz é mapear os dados para um espaço de dimensão suficientemente alta.

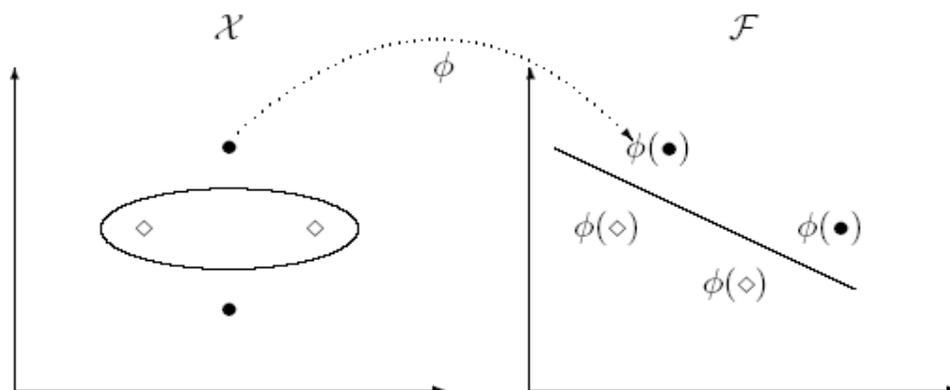


Figura 22. Mapeamento de características

No contexto de reconhecimento de padrões, a idéia principal de uma máquina de vetor suporte é construir um hiperplano ótimo de separação entre exemplos positivos e negativos. De fato, existem casos em que podem existir vários possíveis hiperplanos de separação, mas SVMs buscam apenas encontrar o que maximize a margem entre os exemplos (vide Figura 23 [39]).

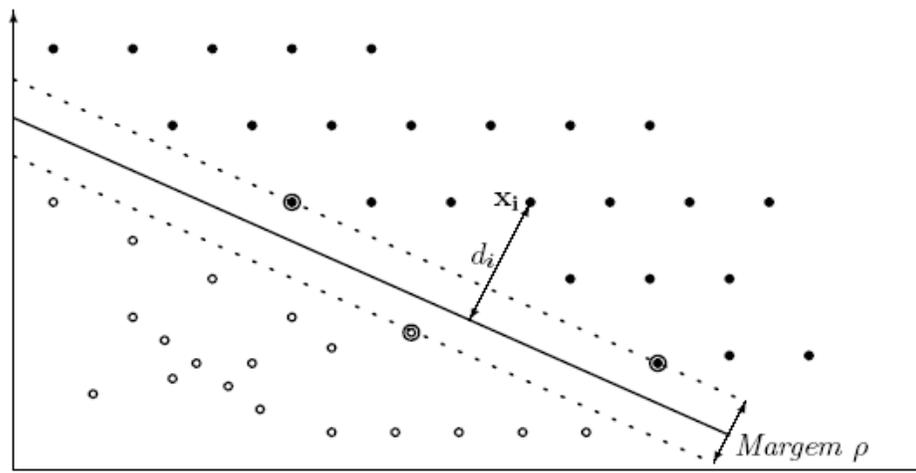


Figura 23. Margem geométrica de um ponto x_i e a margem ρ do hiperplano de separação ótimo

Na Figura 23 os círculos fechados são os exemplos positivos e os círculos abertos são os exemplos negativos. Os círculos que caem sobre as margens (linhas tracejadas) são os vetores suporte (com um círculo mais externo) para esse conjunto de treinamento. Na verdade, os vetores de suporte consistem de um subconjunto dos dados de treinamento extraído pelo algoritmo.

Um ponto relevante no algoritmo de aprendizagem por vetor de suporte é a função kernel ou função núcleo. Dependendo da função utilizada, podemos construir diferentes máquinas de aprendizagem. Pode-se, por exemplo construir máquinas de aprendizagem linear, polinomial, RBFs e MLPs. A Tabela 2 mostra as funções kernel citadas.

Tabela 2. Exemplos de funções kernel utilizadas no algoritmo de aprendizagem de SVMs

Nome da função	Função
Linear	$K(x_i, x_j) = x_i^T x_j$
Polinomial	$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
Sigmóide	$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$
Gaussiana (RBF)	$K(x_i, x_j) = \exp(-\gamma \ x_i - x_j\ ^2), \gamma > 0$

No escopo desse trabalho, utilizamos a função kernel Gaussiana. Sendo assim, as variáveis que nos concentramos em utilizar foram o C ($C > 0$), parâmetro de penalidade do termo de erro, e o γ , largura da função kernel RBF. Em SVMs, a determinação desses dois parâmetros é ponto decisivo para a obtenção de bons resultados, pois estas máquinas são extremamente sensíveis aos seus valores.

1.3 Técnica do vizinho mais próximo (kNN)

Os modelos de vizinho mais próximo, técnica não paramétrica e supervisionada [29], baseiam-se na idéia de que as propriedades de qualquer ponto x tem probabilidade de serem semelhantes às propriedades dos pontos de sua vizinhança. Para identificar essa vizinhança, utiliza-se uma métrica de distância, que geralmente é a distância Euclidiana. Se a vizinhança for muito pequena, ela não conterá quaisquer pontos de dados; se for muito grande, poderá conter todos eles. A solução para esse impasse é dado pela técnica kNN, na qual tenta incluir somente k pontos, ou

seja, se a maioria dos exemplos encontrados entre os k vizinhos possui classe C_i , então atribui-se ao novo exemplo a classe C_i (ver Figura 24). Usualmente valores ímpares para k são utilizados para evitar empate.

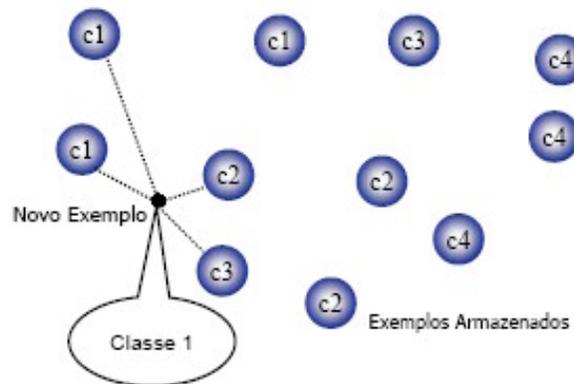


Figura 24. Classificação de padrão utilizando kNN

O kNN não possui fase de treinamento. Nessa técnica, os dados são todos armazenados. Para $k = 1$, para um novo padrão, é calculada a distância Euclidiana (por exemplo) entre esse padrão e todos os padrões de treinamento. A menor distância determinará a classe do novo padrão. Para $k > 1$, são considerados os k vizinhos mais próximos do novo padrão, e dentro desse subconjunto, encontra-se a classe mais numerosa.

A técnica do vizinho mais próximo é bastante simples, no entanto requer o armazenamento de todos os padrões na memória.

1.4 Validação cruzada e Normalização

Validação cruzada, técnica bastante utilizada para testar e comparar classificadores, a base de dados é subdividida em n blocos (n -fold) de tamanhos aproximados, nos quais $n-1$ blocos são utilizados como conjunto de treinamento e o último bloco restante é usado como conjunto de teste. Sendo assim, n rodadas serão necessárias para realizar o treinamento. Em suma, o que acontece durante o treinamento é o seguinte: na primeira rodada, o primeiro bloco é utilizado como conjunto de teste enquanto os $n-1$ blocos restantes são utilizados como conjunto de treinamento, na segunda rodada, o segundo bloco é utilizado como conjunto de teste e os $n-1$ blocos restantes como conjunto de treinamento, e assim sucessivamente (vide Figura 25). Ao final de cada rodada o erro de classificação é calculado, e ao término da última rodada, a média e desvio padrão do erro total é efetuado.

A aplicação de validação cruzada em experimentos é sugerida quando a quantidade de padrões existentes nas bases de dados do estudo é pequena. Os resultados obtidos também são considerados mais precisos, desde que a frequência de exemplos é distribuída ao longo do treinamento.

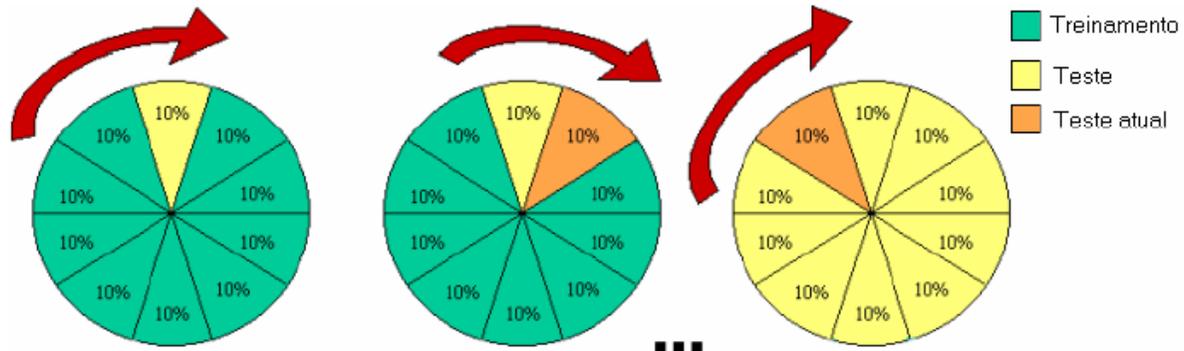


Figura 25. Exemplo de validação cruzada em 10 blocos

O processo de normalização consiste na aplicação de uma fórmula em determinados atributos de um conjunto de dados a fim de colocá-los dentro de um mesmo intervalo de valores $[Y_{\min}, Y_{\max}]$. Estudos mostram que a normalização dos dados aumenta o desempenho de classificação de técnicas de aprendizado de máquina.

1.5 Técnicas de aprendizado de máquina e medicina

A indicação da aplicação da inteligência artificial (IA) na medicina é datada desde a década de 1950. Ledley e Lusted [1] propuseram que ela fosse usada como ferramenta no auxílio de diagnóstico médico. No entanto, atualmente, a IA tem sido utilizada desde o ensino médico superior [16] até no auxílio de diagnósticos médicos [10].

Apesar da sugestão ser datada desde décadas atrás, o uso de técnicas de aprendizado de máquina na medicina de maneira significativa só tem se tornado rotina na última década do século passado. Lisboa [9] faz uma vasta revisão dos benefícios advindos de uma dessas técnicas, redes neurais artificiais. Ele destaca pelo menos três categorias de aplicação de RNAs na medicina:

1. Ferramentas de monitoramento: cuja meta é detectar anormalidades, com o propósito de lembrar os usuários (profissionais de saúde) de diagnósticos que podem ter sido desconsiderados ou mesmo chamar atenção para prováveis doenças raras.
2. Ferramentas de avaliação para pacientes específicos: dado que pacientes tem de ser vistos como indivíduos diferentes entre si, a meta é desenvolver ferramentas capazes de detectar as individualidades de cada um deles. Tais ferramentas podem ser utilizadas para inferir diagnósticos ou prognósticos. Por diagnóstico entende-se como sendo o conhecimento ou determinação de uma doença pela observação dos sintomas. Por prognóstico entende-se como a conjectura sobre o que vai acontecer acerca do curso da doença.
3. ferramentas interativas para planejamento: cujo objetivo é fornecer um cenário de análise capaz de ajudar os médicos a testar hipóteses sobre as condições de uma paciente, e então considerar os possíveis efeitos de diferentes tratamentos.

A motivação para o uso de redes neurais na medicina advém do seu potencial para analisar dados não-lineares, além da capacidade de aprendizado, de processamento paralelo, de tolerância a dados ruidosos e de adaptabilidade a novas manifestações de doenças.

Baxt, em sua pesquisa datada de 1995, identifica aproximadamente 1000 citações de redes neurais na literatura biomédica [10, 11], muitas delas descrevem estudos preditivos em bases de

dados médicas. Além disso, certos domínios médicos foram identificados como potenciais para o auxílio de diagnóstico. Um exemplo disso são pacientes que ingressam em emergências hospitalares com dores agudas no peito, e então, a partir disso, objetiva-se identificar aqueles mais suscetíveis ao infarto do miocárdio. Um estudo envolvendo 706 pacientes [40] retornou um sensibilidade e especificidade de 80 a 96%, respectivamente, quando redes neurais foram utilizadas, comparado com o desempenho de médicos, os quais obtiveram uma sensibilidade e especificidade de 88% e 71%, respectivamente. Por sensibilidade [41] entende-se como uma medida de desempenho que se refere a habilidade de discernimento. Uma sensibilidade perfeita é aquela que tem uma porcentagem de acerto de 100% e de alarme falso (erro de classificação) de 0%. Por especificidade, entende-se como a correta taxa de classificação de casos rejeitados.

Publicações recentes sobre sistemas de apoio a diagnósticos (SAD) auxiliados por computador concluíram que as áreas mais promissoras para o desenvolvimento desses tipos de sistemas são dosagem de drogas (medicações) e cuidado preventivo [42]. Eles servem para alertar contra os riscos das prescrições de determinados tipos de drogas em determinados tipos de patologias. Weiner [43] sinaliza também que o uso de SAD na avaliação de mamografias diminui pela metade o número de lesões não-identificadas sem aumentar a taxa de falsos positivos (classificar algo como positivo, quando na verdade não é), embora haja um aumento de custos em termos de tempo, treinamento e equipamento.

Outras técnicas de aprendizado de máquina são bastante utilizadas, como algoritmos genéticos [2], kNN [44, 45] e máquinas de vetores suporte [46, 47]. Por vezes, combinações das técnicas são propostas para maximização dos resultados.

Enfim, são muitas as aplicações na área médica que utilizam redes neurais artificiais, kNN e SVMs: oncologia, análise para sobrevivência, medicina intensiva, cardiologia, radiologia, medicina física e reabilitação, modelos prognósticos e diagnósticos, previsão de alta de pacientes, farmacologia, etc.

No Brasil, a utilização de técnicas de aprendizado de máquina ainda é pouco empregada. No entanto, alguns pesquisadores trabalham para que o cenário se altere. Fernandes [48], com ajuda de colaboradores, em recente publicação, cita vários sistemas que usam IA para auxílio a diagnósticos e no ensino superior na área da saúde. Dentre os citados, alguns se destacam, tais como:

- Sistema especialista para diagnóstico de doenças periodontais;
- Sistema especialista difuso de apoio ao aprendizado do traumatismo dento-alveolar utilizando recursos multimídia;
- Aplicação de descoberta de conhecimento em malformações congênitas;
- Sistemas tutores inteligentes aplicados em diabetes mellitus.

Talvez os próprios médicos sejam o maior impasse existente na aplicação rotineira de sistemas como os citados acima. O temor de ser “trocado” por uma máquina é real, bem como o potencial constrangimento caso ela supere sua perspicácia e capacidade de julgamento. Problemas de ordem financeira dos hospitais também podem ser fator limitante.

Capítulo 2

Simuladores de técnicas de aprendizado de máquina

No presente capítulo são apresentados os simuladores utilizados nos experimentos deste trabalho. Três simuladores foram utilizados: SNNS [17], para RBF-DDA, LIBSVM [21], para Máquinas de vetor suporte, e WEKA [24], para kNN e MLP-*backpropagation*.

2.1 SNNS (*Stuttgart Neural Network Simulator*)

O SNNS é um simulador de redes neurais artificiais desenvolvido pelo *Institute for Parallel and Distributed High Performance Systems* (IPVR) da Universidade de Stuttgart, em 1989. O simulador fornece um ambiente flexível e eficiente para desenvolvimento de experimentos, pesquisas e aplicações com redes neurais. Em nossos experimentos, utilizamos a versão 4.2, que é a mais recente.

O SNNS, desenvolvido na linguagem ANSI-C, consiste de quatro principais componentes:

1. Kernel: utilizado nas operações das estruturas de dados das RNAs, como também é responsável pelos algoritmos de aprendizado das mesmas.
2. Interface gráfica: permite criação, manipulação e visualização 2D e 3D das redes geradas.
3. Interface de execução em batch
4. Compilador *snns2c*: permite que as redes treinadas sejam compiladas para linguagem C a fim de serem utilizadas em outros aplicativos.

O SNNS dá suporte a vários sistemas operacionais, tais como Windows, Linux, HP-UX, SunOS, Ultrix, OSF1, AIX e NeXTStep. Por ser uma ferramenta freeware e de código aberto, ela permite que qualquer programador incremente funcionalidades ao mesmo.

A Figura 26 ilustra a interface gráfica do simulador, a qual permite manipulação de arquivos, controle do processo de teste e treinamento da rede, informação sobre neurônios e pesos das conexões, diagrama da rede, gráfico de erro, edição de redes, além da ajuda (*Help*) do próprio simulador.

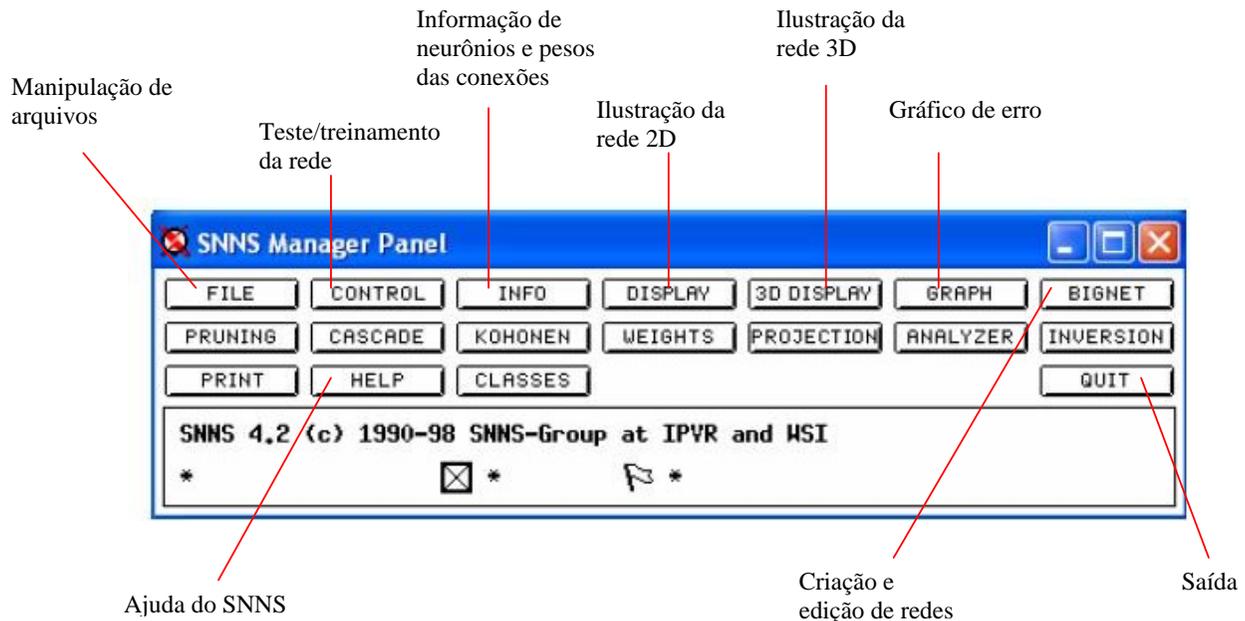


Figura 26. Interface gráfica do SNNS

Como citado, o SNNS permite a criação e manipulação de redes neurais dos mais diversos tipos. No entanto, em nosso trabalho, só utilizamos, junto ao SNNS, redes RBF.

Para a criação de redes RBF, o processo é simples. Através do menu BIGNET → general o SNNS permite a construção desse tipo de rede. A Figura 27 ilustra o processo.

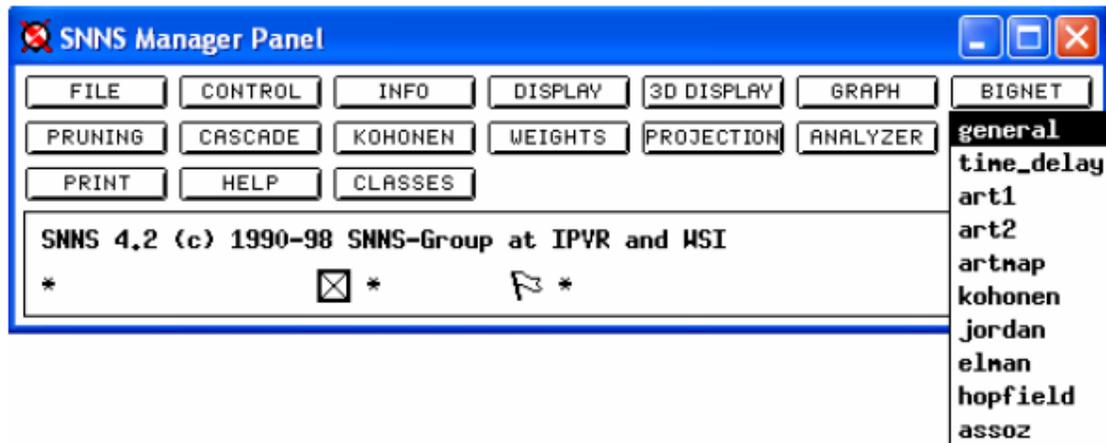


Figura 27. Criação de rede através do SNNS

Após o passo referido, outra tela é aberta, na qual se pode criar de fato a rede, através da definição de parâmetros como tipo de conexões, quantidade de camadas escondidas e número de neurônios presentes em todas as camadas. No presente trabalho, foram utilizadas seis bases distintas, o que ocasionou a criação de seis diferentes redes, dado que o número de nodos da camada de entrada e saída não eram iguais nas bases. Como foi utilizado o algoritmo DDA para o treinamento da rede, os neurônios da camada escondida não foram previamente definidos, bem como o tipo de conexão existente entre eles, pois o algoritmo também é responsável por essa tarefa.

De forma geral, o processo para criação de redes RBF passa somente pela definição do número de neurônios da camada de entrada e saída (Figura 28). Os botões TYPE e POS definem o tipo da camada (entrada, oculta ou saída) e a posição dos neurônios. Sendo assim, basta definir a quantidade de neurônios de ambas camadas e então pressionar o botão CREATE NET. As outras funcionalidades presentes nesse painel fogem do escopo do trabalho.

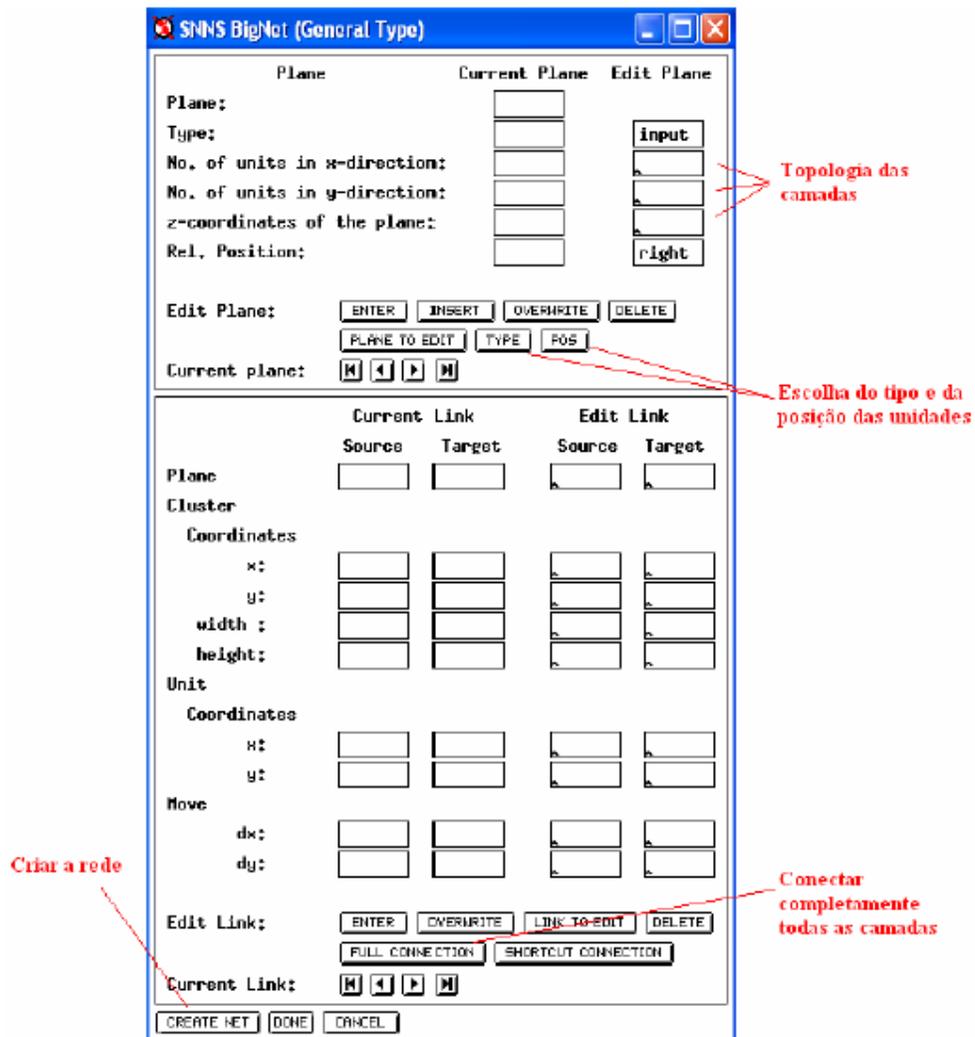


Figura 28. Definição da arquitetura de redes no SNNS

Através da interface gráfica ilustrada na Figura 26, é possível visualizar uma rede criada. A opção para ter esse acesso é feita pelo botão DISPLAY. A Figura 29 mostra uma rede RBF criada através do SNNS. Como citado anteriormente, a rede RBF-DDA, em seu estado inicial, não possui neurônios na camada escondida.

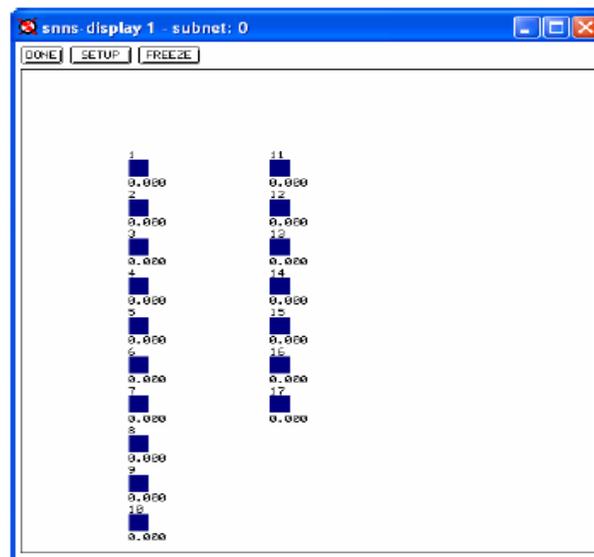


Figura 29. Exemplo de uma rede RBF-DDA construída através do SNNS

Após os passos citados, o que resta é apenas salvar a rede. Por meio do painel principal, através do botão FILE, é possível salvar a rede criada. Conforme a Figura 30, pode-se trabalhar com cinco tipos de arquivos diferentes. Para nossos experimentos, no entanto, só foi necessário a manipulação de três deles: *Network*, *Patterns* e *Result File*. O arquivo do tipo *Network* é usado para salvar ou carregar redes neurais desenvolvidas pelo SNNS. Arquivos *Patterns* contém os conjuntos de dados, normalmente seccionados em conjunto de teste, treinamento e validação. São utilizados ao longo do algoritmo de aprendizagem das redes. Arquivos do tipo *Result File* contém os resultados dos experimentos, tais como padrões classificados corretamente e erroneamente pela rede. Uma ferramenta chamada *Analyze*, contida no SNNS, é utilizada para extrair informações dos arquivos do tipo *Result File*. Ela ilustra, por exemplo, a porcentagem de erro e acerto da rede nos conjuntos de teste, treinamento e validação.

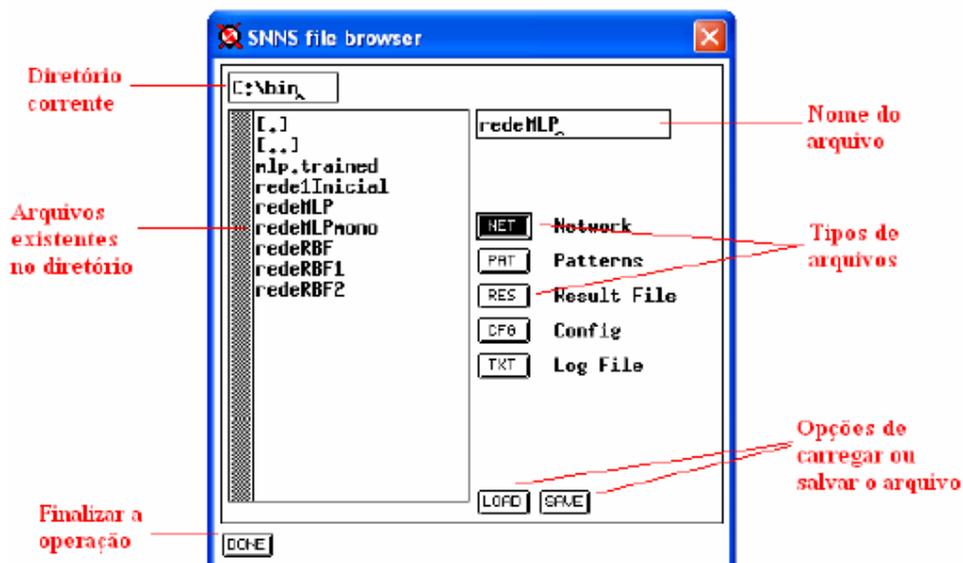


Figura 30. Painel de manipulação de arquivos do SNNS

2.2 LIBSVM (A *Library to Support Vector Machine*)

LIBSVM é uma biblioteca que utiliza máquinas de vetores suporte para classificação e regressão de padrões. LIBSVM dá suporte a vários sistemas operacionais, como também contém implementações em várias linguagens (C++, C# e Java). Ferramenta gratuita e de código aberto, ela permite conexão com outros aplicativos, tais como Matlab, Ruby, LabVIEW e outros. LIBSVM provê ferramentas simples e amigáveis as quais permitem uma fácil execução de experimentos por parte do usuário, a saber:

1. SVMToy: applet simples que demonstra classificação e regressão utilizando SVMs.
2. SVMTrain: permite o treinamento dos dados. Através de um arquivo contendo os dados em formato específico, o SVMTrain treina e gera um arquivo do tipo MODEL. Tal arquivo é utilizado para predição (classificação) dos dados.
3. SVMPredict: através de um arquivo MODEL e um arquivo de entrada contendo dados não apresentados durante o treinamento, tal ferramenta classifica os dados gerando um arquivo de saída.
4. SVMScale: utilizado para normalizar dados. Normalização é importante para eliminar possíveis discrepâncias entre os valores dos dados.

Na Figura 31, encontra-se ilustrado o SVMToy, o qual permite o usuário “brincar” com os dados. Tal ferramenta possui fins didáticos para ensino de como se dá a classificação por meio de SVMs, bem como mostrar como a definição dos parâmetros influi sensivelmente na classificação dos padrões.

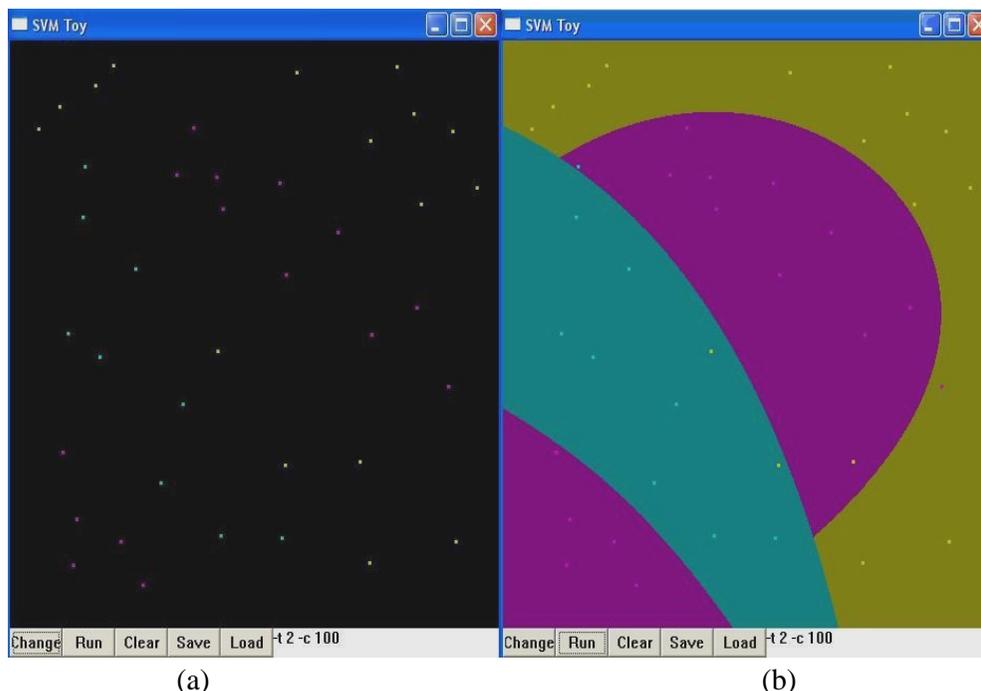


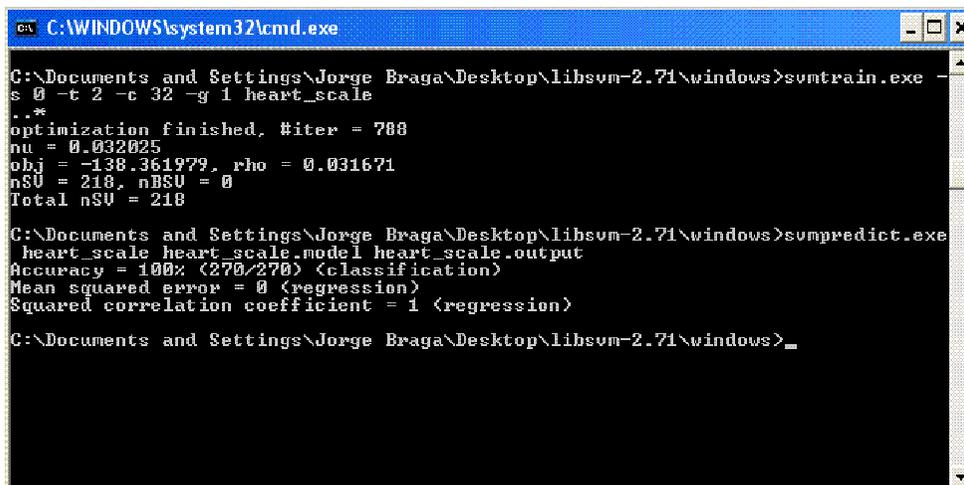
Figura 31. SVMToy antes (a) e após (b) a classificação

Na Figura 31 (a), percebe-se a presença de três classes (pontos roxos, verdes e amarelos). A Figura 31 (b) ilustra as regiões criadas pelas SVMs. Após essa definição de classes, basta

definir os parâmetros para a classificação. São muitos os parâmetros que podem ser testados, no entanto, neste trabalho, descreveremos apenas aqueles que foram utilizados, a saber:

- -s : determina o tipo de SVM a ser utilizada.
 - a. 0 = C-SVC
 - b. 1 = nu-SVC
 - c. 2 = one-class SVM
 - d. 3 = epsilon-SVR
 - e. 4 = nu-SVR
- -t : determina o tipo da função kernel a ser utilizada pelo algoritmo de aprendizagem da SVM.
 - a. 0 = função linear
 - b. 1 = função polinomial
 - c. 2 = função de base radial
 - d. 3 = função sigmóide.
- -c : define o parâmetro C da SVM.
- -p : define o parâmetro γ da SVM.
- -v : define o número de blocos (*folds*) a serem utilizados na validação cruzada.

Referente ao SVMTrain e ao SVMPredict, suas utilizações são bastante simples. Por meio de linhas de comando, via *prompt de comando* (Windows ou Linux), o treinamento e classificação é realizado. A Figura 32 mostra o processo de treinamento e teste.



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Jorge Braga\Desktop\libsvm-2.71\windows>svmtrain.exe
s 0 -t 2 -c 32 -g 1 heart_scale
..**
optimization finished. #iter = 788
nu = 0.032025
obj = -138.361979, rho = 0.031671
nSV = 218, nBSV = 0
Total nSV = 218

C:\Documents and Settings\Jorge Braga\Desktop\libsvm-2.71\windows>svmpredict.exe
heart_scale heart_scale.model heart_scale.output
Accuracy = 100% (270/270) <classification>
Mean squared error = 0 <regression>
Squared correlation coefficient = 1 <regression>

C:\Documents and Settings\Jorge Braga\Desktop\libsvm-2.71\windows>_
```

Figura 32. Treinamento e classificação através do SVMTrain e SVMPredict, respectivamente

Ainda, o LIBSVM provê o aplicativo SVMScale, responsável pela normalização dos dados. Hsu [49], em conjunto com os desenvolvedores do LIBSVM, realizou alguns experimentos para comprovar o benefício advindo da normalização. Em todos os testes, os dados normalizados alcançaram melhor índice de classificação que aqueles dados não-normalizados.

Finalmente, o LIBSVM possui uma ferramenta de seleção de modelo usando a função kernel RBF. A ferramenta utiliza uma busca paralela em conjunto com validação cruzada.

Atualmente, a busca pelos melhores parâmetros é limitada a apenas dois deles: C e γ . À medida que a busca pelos parâmetros é realizada, um gráfico ilustrando todo o processo vai sendo gerado.

2.3 WEKA (*Waikato Environment for Knowledge Analysis*)

O WEKA é uma coleção de algoritmos de aprendizagem de máquina para resolução de problemas de associação, classificação, regressão e clustering. Desenvolvido em Java pela universidade de Waikato, é uma ferramenta gratuita, de código aberto, de fácil manuseio, com interface gráfica simples (Figura 33) e que utiliza um formato de dados específico (*.arff*). Tal formato de arquivo permite que os dados sejam utilizados sem necessidade de normalização. No presente trabalho, utilizamos a versão 3.4.5.

Conforme se observa, o WEKA fornece quatro opções principais: *Simple CLI*, *Explorer*, *Experimenter* e *KnowledgeFlow*. A primeira opção permite a realização de experimentos por meio de comandos (*prompt*), a segunda é utilizada para pré-processamento e aplicação de técnicas de aprendizagem de máquina, a terceira é usada para fins de comparação entre as mais variadas técnicas de inteligência computacional, e finalmente a quarta e última é uma nova interface gráfica para o WEKA. Das opções citadas, no corrente trabalho, somente utilizamos a segunda delas, *Explorer*.



Figura 33. Interface gráfica do WEKA

A tela do *Explorer* (vide Figura 34) permite o pré-processamento dos dados, ou seja, eles podem ser discretizados, normalizados, selecionados, transformados e combinados. As bases de dados podem ser importadas de arquivos em vários outros formatos (*arff*, *csv*, *c4.5*, binário), além de que elas podem ser lidas de uma URL (*Uniform Resource Locator*) ou site da internet.

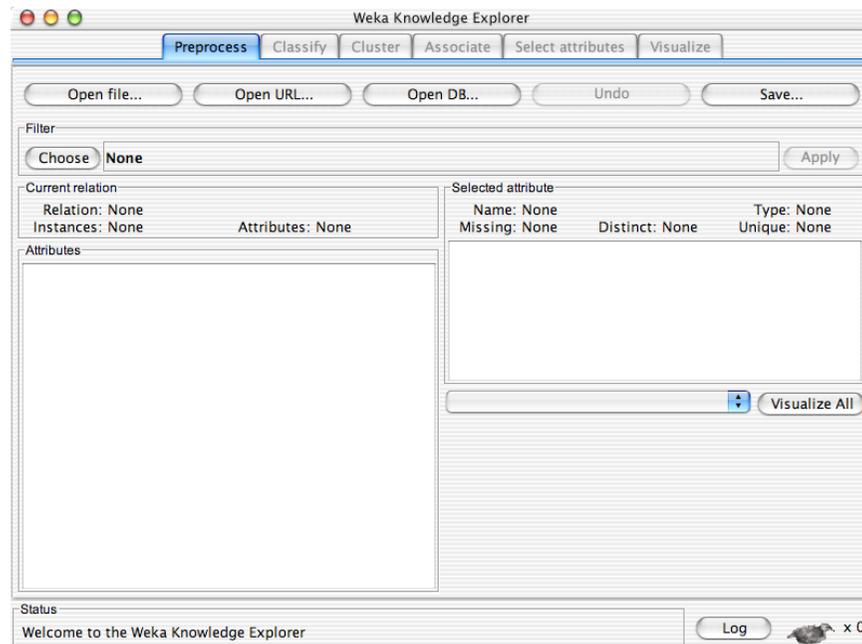


Figura 34. Interface gráfica da opção Explorer do WEKA

Em nosso caso, não utilizamos quaisquer pré-processamentos citados. Da interface ilustrada na Figura 34, só foi usada as opções da aba *Classify*. Por tal motivo, somente serão explicadas as funcionalidades utilizadas no escopo do trabalho.

A aba *Classify* é utilizada para classificação de dados, tanto nominais quanto numéricos. Vários classificadores são fornecidos pelo WEKA, dentre os quais estão árvores e listas de decisão, máquinas de vetores suporte, MLP, kNN, regressão e redes bayesianas. A opção de validação cruzada também é oferecida. Dentre os classificadores citados, em nosso presente trabalho, foram utilizados MLP e kNN.

No classificador MLP, escolhido pelo botão *Choose*→*Functions*→*MultiLayerPerceptron*, várias opções podem ser definidas, tais como número de épocas, número de camadas escondidas, número de neurônios das camadas escondidas, taxa de aprendizado, momentum, porcentagem de dados presentes no conjunto de validação, número de blocos a serem usados na validação cruzada (caso se decida aplicá-la), além de outras variáveis. O classificador utiliza o *backpropagation* como algoritmo de aprendizagem da rede. Em nosso caso, todos os parâmetros citados foram utilizados, menos o termo momentum, desde que a proposta era usar o algoritmo de aprendizagem original. A Figura 35 ilustra a interface que permite definir os parâmetros da rede MLP-*Backpropagation*.

Para o classificador kNN (Figura 36), escolhido pelo botão *Choose*→*lazy*→*IBK*, por ser uma técnica considerada não-paramétrica, somente os valores de *k* foram alterados durante as simulações.

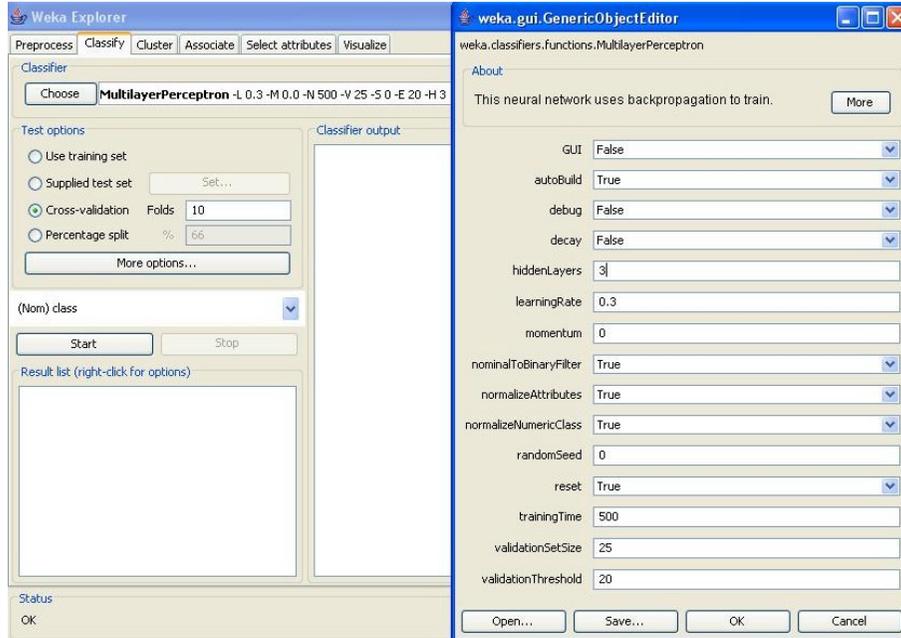


Figura 35. Classificador MLP presente no WEKA

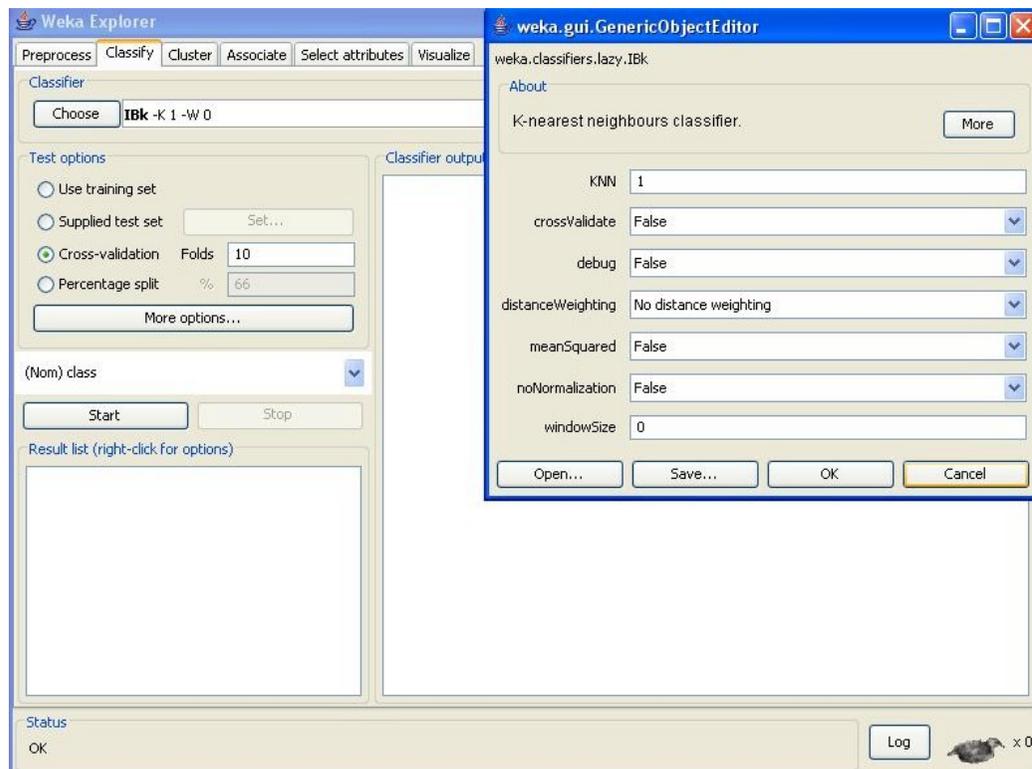


Figura 36. Classificador kNN presente no WEKA

Capítulo 3

Bases de dados médicas

No presente capítulo, as bases de dados utilizadas nos experimentos são explicadas.

3.1 Bases de dados

3.1.1 Doença inflamatória intestinal

A base de dados de doença de estômago [15] é composta de 809 instâncias, com 23 atributos (vide Tabela 3), os quais são utilizados para diagnosticar o paciente. A população que compõe os dados faz parte de um estudo realizado pelo departamento de histopatologia do *Royal Hallamshire Hospital*, da Inglaterra. As confirmações dos diagnósticos foram feitas por exames endoscópicos e por biópsia. A distribuição das classes contidas na base de dados é a apresentada na Figura 37.

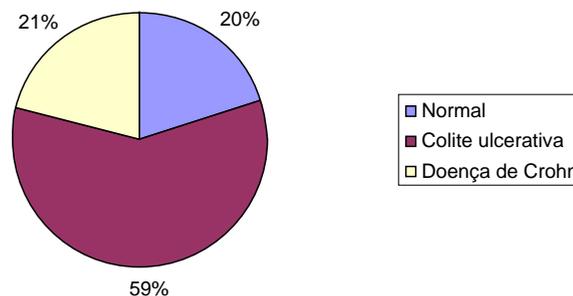


Figura 37. Distribuição percentual das classes na base de dados de doença inflamatória intestinal

A classe Normal possui 165 padrões, a Colite Ulcerativa tem 473 e, finalmente, a Doença de Crohn apresenta 171 padrões.

Tabela 3. Características da base de dados Doença Inflamatória Intestinal

Atributo	Tipo	Intervalo	0	1	2	3
Idade	Inteiro	14-84				
Sexo	Binário	0,1	Homem	Mulher		
Inflamação ativa	Binário	0,1	Não	Sim		
Superfície mucosa	Categoria ordenada	0,1,2	Lisa	Irregular	Projeção vilosa	
Arquitetura da cripta	Categoria ordenada	0,1,2,3	Normal	Leve	Moderada	Severa
Perfil da cripta	Inteiro	2-7				
Celularidade aumentada da lâmina própria	Binário	0,1	Não	Sim		
Discreto e superficial aumento na celularidade da lâmina própria	Binário	0,1	Não	Sim		
Agregados linfóides aumentados na lâmina própria	Binário	0,1	Não	Sim		
Celularidade desigual da lâmina própria	Binário	0,1	Não	Sim		
Acentuado aumento transmucoso na celularidade da lâmina própria	Binário	0,1	Não	Sim		
Extensão da cripta	Categoria ordenada	0,1,2,3	Nenhuma	Pequena	Moderada	Acentuada
Polimorfos na cripta	Categoria ordenada	0,1,2,3	Nenhuma	Pouca	Severa	Muita
Extensão do abscesso da cripta	Categoria ordenada	0,1,2,3	Nenhuma	Pequena	Moderada	Marcada
Polimorfos no abscesso da cripta	Categoria ordenada	0,1,2,3	Nenhuma	Pouca	Moderada	Muita
Polimorfos na lâmina própria	Categoria ordenada	0,1,2	Ausente	Focal	Difusa	
Alterações epiteliais	Categoria ordenada	0,1,2,3	Normal	Achatamento	Degeneração	Erosão

Depleção de mucina	Categoria ordenada	0,1,2,3	Normal	Leve	Moderada	Severa
Linfócitos intraepithelial	Binário	0,1	Normal	Aumentado		
Colágeno subepitelial	Binário	0,1	Normal	Aumentado		
Granulomas da lâmina própria	Binário	0,1	Ausente	Presente		
Granulomas sumucosos	Binário	0,1	Ausente	Presente		
Célula histiocítica basal	Binário	0,1	Ausente	Presente		

3.1.2 Câncer de mama

O câncer de mama é o tipo mais comum de câncer (excluindo o câncer de pele) que afeta mulheres na América do Norte, Europa e Brasil [50]. O diagnóstico para esse tipo de câncer depende do tamanho da área que o tumor ocupa, além do tempo de vida do mesmo. Quanto mais rápida a detecção do câncer, mais chances a paciente tem de sobreviver. Inúmeras técnicas são utilizadas para a detecção de tumores, sejam eles benignos ou malignos. Inglaterra e Estados Unidos da América (EUA) costumam utilizar radiografia da mama (mamografia) para identificar possíveis anormalidades presentes na mama do paciente. Embora a mamografia e o exame do toque serem os métodos mais tradicionais e conhecidos dos pacientes, uma outra técnica é ainda mais utilizada, conhecida como aspiração de lesões mamárias através de agulhas [50]. Nessa técnica, células são extraídas da mama através de uma seringa e então são transferidas para uma solução, que por sua vez é enviada para um laboratório de patologias. Apesar da simplicidade da ideia do método, somente médicos (citopatologistas) experientes conseguem detectar a presença de tumores com alto índice de acerto. A formação de um médico citopatologista experiente leva em média cinco anos. Devido a esses fatores, a aplicação de técnicas de aprendizado de máquina é algo interessante e promissor, as quais podem preencher essas lacunas.

A base de dados, resultado de um estudo que coletou 692 testes de laboratório da técnica citada, foi coletada pelo departamento de patologias do *Royal Hallamshire Hospital*, da Inglaterra [15]. O resultados foram classificados segundo dois grupos: tumores benignos e tumores malignos. A classe de tumores benignos possui 457 exemplos, enquanto a classe de tumores malignos tem 235 padrões em seu conjunto. A Figura 38 ilustra o percentual de distribuição de classes.

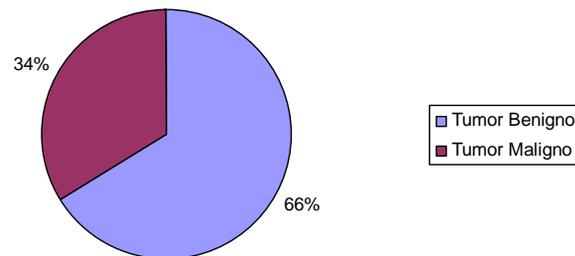


Figura 38. Distribuição percentual das classes na base de dados de câncer de mama

Existem 11 (onze) variáveis de entrada (atributos) presentes na base. Todas as observações foram realizadas por um consultor patologista com 10 anos de experiência na técnica já citada. A Tabela 4 contém as variáveis que compõem a base.

Tabela 4. Características da base de dados de câncer de mama

Atributo	Tipo	Intervalo	0	1
Idade	Inteiro	17-89		
Displasia celular	Binário	0,1	Ausente	Presente
Lâmina intracitoplasmática	Binário	0,1	Ausente	Presente
Tridimensionalidade dos grupos das células epiteliais	Binário	0,1	Ausente	Presente
Núcleo bipolar exposto	Binário	0,1	Ausente	Presente
Macrofagia espumosa	Binário	0,1	Ausente	Presente
Nucléolo	Binário	0,1	Ausente	Presente
Pleiomorfismo nuclear	Binário	0,1	Ausente	Presente
Tamanho do núcleo	Binário	0,1	Ausente	Presente
Células piteliais necróticas	Binário	0,1	Ausente	Presente
Alterações apócrinas	Binário	0,1	Ausente	Presente

3.1.3 Diabetes

A base de dados de diabetes, disponível no site da UCI *machine learning* [14], é constituída de 768 observações, todas elas coletadas de jovens de descendência indiana que residem nos EUA. O estudo foi realizado em 1990 por Vincent Sigillito do *National institute of Diabetes and Digestive and Kidney Diseases*, cujo objetivo era identificar se a paciente mostrava sinais

(sintomas) de diabetes, de acordo com critérios estabelecidos pela Organização Mundial de Saúde (OMS). Oito atributos (ver Tabela 5) determinam a classe ao qual as pacientes pertencem: teste positivo, na qual a presença da doença é constatada, e teste negativo, na qual não há diabetes. Duzentos e sessenta e oito (268) padrões do teste positivo são encontrados na base, enquanto para a outra classe, 500 padrões são identificados. A Figura 39 mostra a distribuição percentual de classes presente na base de dados diabetes.

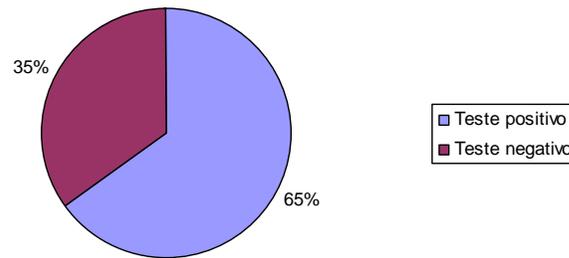


Figura 39. Distribuição percentual das classes na base de dados Diabetes

Tabela 5. Características da base de dados Diabetes

Atributo	Tipo	Intervalo
Quantidade de gravidez	Inteiro	0-17
Concentração plasmática de glicose de 2h no teste oral de tolerância a glicose	Inteiro	0-199
Pressão sanguínea diastólica (mm Hg)	Inteiro	0-122
Espessamento da prega cutânea do tricepes	Inteiro	0-99
Insulina sérica de 2h (mu U/ml)	Inteiro	0-846
Índice de massa corpórea	Real	0-67,1
Função de continuidade de diabetes	Real	0,078-2420
Idade	Inteiro	21-81

3.1.4 Doença do fígado

A base de dados de doença de fígado, disponível no site da UCI *machine learning* [14], é composta por 345 instâncias coletadas de pacientes do sexo masculino. Seis variáveis definem as possíveis classes: ausência ou presença de doença do fígado. O estudo, realizado pelo BUPA Medical Research Ltd., visava a obtenção de relação de consumo de álcool e doenças de fígado. A Figura 40 ilustra a distribuição probabilística das classes contidas na base. A Tabela 6 contém os atributos presentes na base.

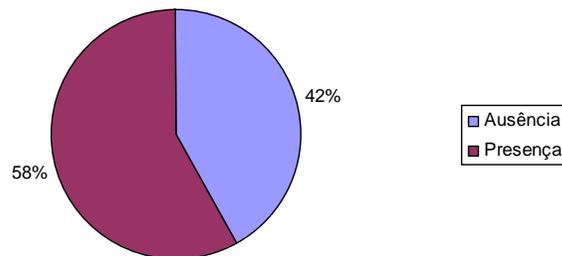


Figura 40. Distribuição percentual das classes na base de dados Doença do Fígado

Tabela 6. Características da base de dados Doença do Fígado

Atributo	Tipo	Intervalo
Volume corpuscular médio	Inteiro	65-103
Fosfatase Alcalina	Inteiro	23-138
Alamine aminotransferase	Inteiro	4-155
Aspartate aminotransferase	Inteiro	5-82
Gamma-glutamyl transpeptidase	Inteiro	5-297
Nº de doses de álcool por dia	Inteiro	0-20

3.1.5 Câncer pulmonar

A classificação correta de três tipos de câncer pulmonar é o objetivo dessa base. Cinquenta e seis atributos predizem o tipo de câncer pulmonar. Ainda, trinta e dois padrões estão presentes na base, disponível no site da UCI *machine learning* [14]. No entanto, não foi encontrada quaisquer descrições sobre o que os atributos de fato representam, bem como os 3 tipos de patologias que se pretende discernir. Apesar desse fator limitante, escolheu-se fazer experimentos com tal base, desde que as técnicas de aprendizado de máquina não tem “consciência” de qual problema está tratando. O objetivo neste trabalho foi verificar o poder de classificação das várias técnicas utilizadas no mesmo. A classe 1 possui 9 padrões, a classe 2 com 13 e a classe 3 com 10 padrões disponíveis. A Figura 41 ilustra o percentual das três classes presentes na base de dados.

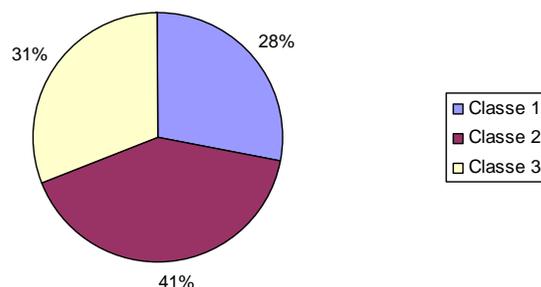


Figura 41. Distribuição percentual das classes na base de dados Câncer pulmonar

3.1.6 Glândula tireóide

Cinco testes de laboratórios (vide Tabela 7) são utilizados para determinar se há presença de problemas na glândula tireóide. Três possíveis classificações existem: Normal (ausência de patologia), com 150 padrões disponíveis, Hipertireoidismo, com 35 e Hipotireoidismo com 30., perfazendo um total de 215 padrões. A base encontra-se disponível no site da UCI *machine learning* [14]. A Figura 42 ilustra o percentual das três classes presentes na base de dados.

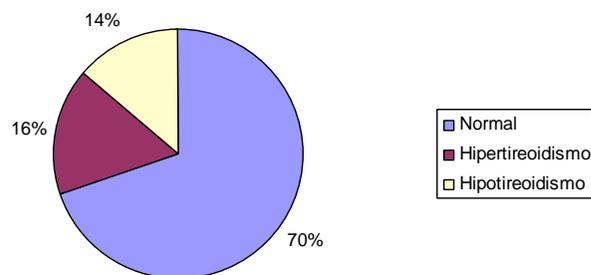


Figura 42. Distribuição percentual das classes na base de dados Glândula tireóide

Tabela 7. Características da base de dados Glândula Tireóide

Atributo	Tipo	Intervalo
Teste de captação de resina – T3	Real	0-1
Tiroxina sérica total	Real	0-1
Triiodotironina sérica total	Real	0-1
Hormônio estimulador de tireoide Basal (TSH)	Real	0-1
Diferença máxima absoluta de TSH	Real	0-1

Finalmente, a Tabela 8 abaixo resume em números as seis bases de dados.

Tabela 8. Características das base de dados presentes no estudo

Base	Nº de entradas	Nº de classes	Nº de padrões
Doença inflamatória intestinal	23	3	809
Câncer de mama	11	2	692
Diabetes	8	2	768
Doença do fígado	6	2	345
Câncer pulmonar	56	3	32
Glândula tireóide	5	3	215

Capítulo 4

Experimentos e resultados

No presente capítulo os experimentos são descritos e os resultados obtidos são comentados.

4.1 Normalização das bases de dados

No presente trabalho, decidiu-se normalizá-los dentro do intervalo [0,1]. A normalização, em nosso caso, foi executada segundo a fórmula abaixo.

$$X_{\text{norm}} = (X - X_{\text{mín}})/(X_{\text{máx}} - X_{\text{mín}})$$

Onde X_{norm} representa o valor normalizado do atributo, $X_{\text{mín}}$ e $X_{\text{máx}}$ representam o menor e o maior valor entre os valores do atributo.

A normalização não foi efetuada para os experimentos de *MLP-Backpropagation* e *kNN*, pois a ferramenta de simulação utilizada para a aplicação de tais técnicas, WEKA, permite que os dados de entrada não necessitem de quaisquer pré-processamentos. Para os outros simuladores, a normalização foi aplicada normalmente. Esse pré-processamento foi regido pelas seguintes diretrizes:

- Dados binários, ou seja, que assumem somente dois possíveis valores, 0 ou 1, não foram normalizados.
- Dados que assumem valores contidos em intervalos contínuos foram normalizados.
- Dados que assumem mais que dois valores, sem estarem contidos em intervalo contínuo, foram também normalizados.
- As possíveis classes presentes em cada base foram codificadas para vetores binários conforme a Tabela 9.

Tabela 9. Codificação das classes das bases de dados

Base de dados	Codificação
Doença inflamatória intestinal	Colite ulcerativa → 001, Doença de Crohn → 010, Normal → 100
Câncer de mama	Tumor maligno → 10, Tumor benigno → 01
Diabetes	Teste positivo → 10, Teste negativo → 01
Doença do fígado	Ausência → 01, Presença → 10
Câncer pulmonar	Classe 1 → 001, Classe 2 → 010, Classe 3 → 100
Glândula tireóide	Normal → 001, Hipertireoidismo → 010, Hipotireoidismo → 100

4.2 Experimentos e resultados utilizando SVM

Os experimentos utilizando máquinas de vetores suporte foram realizados usando o simulador LIBSVM. Como já citado, máquina de vetor suporte é uma técnica de aprendizado de máquina muito sensível aos parâmetros escolhidos. Os resultados podem variar entre extremos, tanto bons como maus desempenhos de classificação são prováveis quando variáveis são alteradas. Por isso, o LIBSVM possui uma ferramenta de seleção de modelo (parâmetros) usando a função kernel RBF. A ferramenta utiliza uma busca paralela em conjunto com validação cruzada em 10 blocos. A busca pelos melhores parâmetros considera os parâmetros: C e γ . Tais variáveis são utilizadas na função kernel RBF, sendo esta a função usada ao longo dos experimentos. A Tabela 10 ilustra o resultado obtido nas bases de dados.

Tabela 10. Parâmetros ótimos identificados pelo LIBSVM

Base de dados	C	Γ	Tempo para encontrar parâmetros ótimos (s)
Doença inflamatória intestinal	16	0,0625	10
Câncer de mama	32	0,125	8
Diabetes	8	0,5	7
Doença do fígado	32	1	5
Câncer pulmonar	2	0,25	5
Glândula tireóide	16	1	5

Observa-se, pela Tabela, que o tempo médio necessário para o LIBSVM encontrar os parâmetros ótimos das bases em questão foi de 6.667 segundos.

Com as variáveis ótimas definidas, efetuou-se o treinamento em cada uma das bases a fim de obter o desempenho de classificação nelas. Validação cruzada em 10 blocos foi utilizada. A Tabela 11, a seguir, ilustra o desempenho de classificação de máquinas de vetores suporte em cada uma das bases de dados.

Como observado, o tempo médio necessário para treinamento e classificação, utilizando validação cruzada em 10 blocos, foi de aproximadamente 1 (um) segundo. Percebe-se, também, a relação entre quantidade de atributos e desempenho de classificação. Notadamente, na base de dados doença inflamatória intestinal, com 25 atributos, e câncer pulmonar, com 56, ambas obtiveram maus desempenhos de classificação. Isso pode ser explicado pela complexidade do problema. Notou-se também que, apesar da diferença na quantidade de padrões de ambas as bases, 777 padrões, o erro de validação cruzada entre elas não foi muito grande, cerca de 7% (ver

Figura 43). Essa observação também tornou-se presente nas bases de diabetes e doença de fígado, apesar de uma diferença de 423 padrões, a diferença no erro de teste foi aproximadamente 4%. A Figura 44, a seguir, ilustra uma comparação entre o n° de padrões e o erro de teste.

Tabela 11. Resultados obtidos por SVM nas bases de dados

Base de dados	N° médio de vetores suporte	Tempo de treinamento (s)	Taxa de acerto (%)
Doença inflamatória intestinal	477,2	2	68,1088
Câncer de mama	74,2	1	96,2428
Diabetes	364,3	1	77,6040
Doença do fígado	211,4	0	73,2258
Câncer pulmonar	27,9	0	61,3903
Glândula tireóide	30,9	1	97,1963

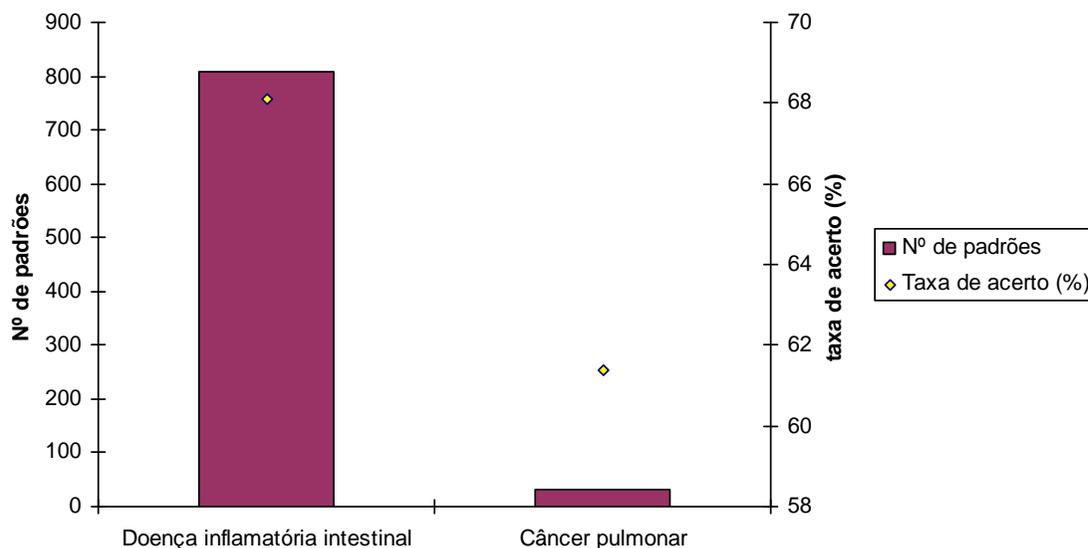


Figura 43. Comparação entre n° de padrões e taxa de acerto (%) entre as bases de dados de doença inflamatória intestinal e câncer pulmonar

Apesar das observações, a literatura não reporta quaisquer evidências sobre o assunto. Antes, acreditava-se que um bom número de exemplos seja um dos fatores responsáveis por uma boa classificação. Com isso, entende-se que o número de atributos também deve ser considerado como um desses fatores, e isso torna-se verdadeiro quando técnicas de seleção de atributos são e estão sendo desenvolvidas. A distribuição de probabilidade de classes num conjunto de exemplos deve ser levada em consideração também, pois uma alta porcentagem de uma classe pode levar a uma má classificação dos exemplos de outra classe com probabilidade bem inferior.

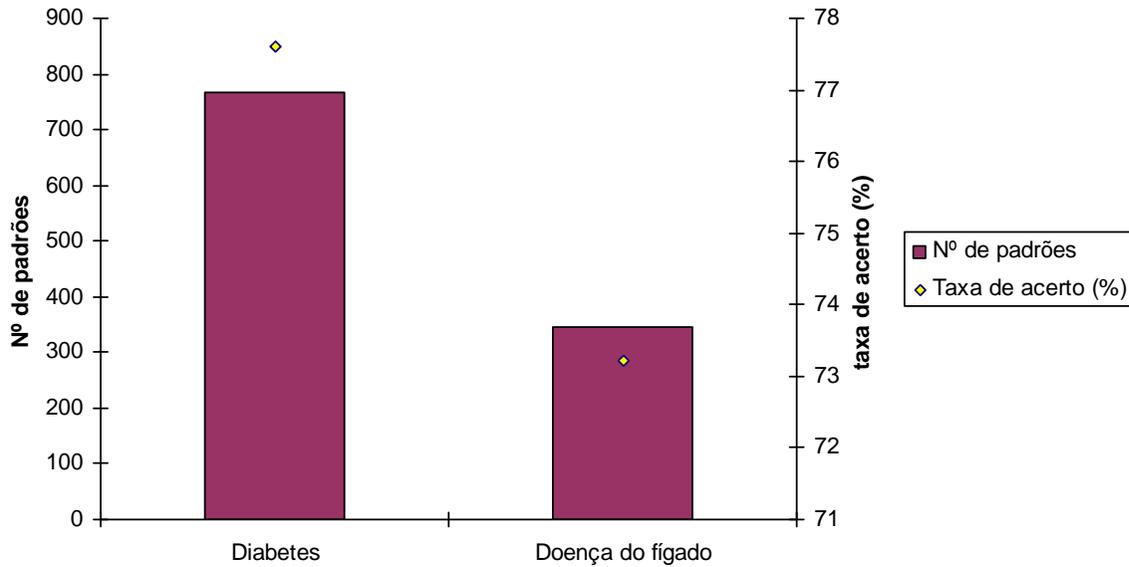


Figura 44. Comparação entre nº de padrões e taxa de acerto (%) entre as bases de dados de diabetes e doença de fígado

Apesar das comparações ilustradas nas Figuras 43 e 44, no caso das bases de câncer de mama e de glândula tireóide, a relação não existiu, pois a diferença entre número de padrões de ambas as bases resulta em 477, o que não prejudicou o desempenho de classificação da base glândula tireóide, a qual obteve maior taxa de corretude no reconhecimento. A Figura 45 faz uma comparação entre as referidas bases.

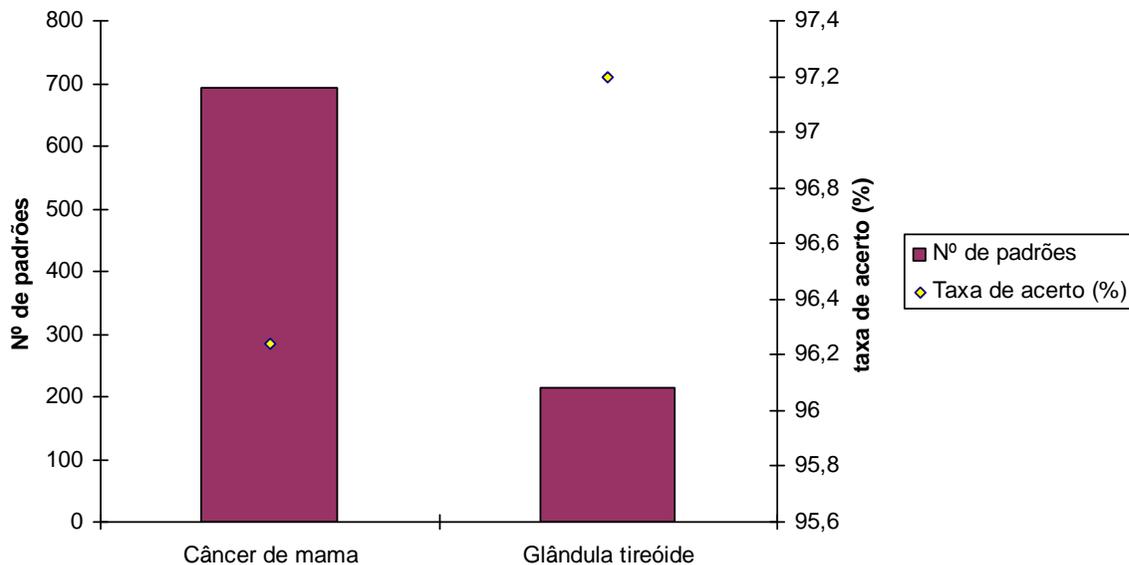


Figura 45. Comparação entre nº de padrões e taxa de acerto (%) entre as bases de dados de câncer de mama e glândula tireóide

4.3 Experimentos e resultados utilizando kNN

Para os experimentos utilizando o modelo do vizinho mais próximo, foi aplicado o simulador WEKA. Por ser uma técnica não-paramétrica, o único valor alterado durante os testes foi o da variável K. A literatura sugere que valores pares não devem ser empregados, pois podem resultar em classificações incorretas, dado que empates podem acontecer. Nos experimentos realizados nas bases de dados, escolheu-se trabalhar com os seguintes valores para $k = 1, 3, 5$ e 7 . Validação cruzada em 10 blocos foi utilizada. A Tabela 12 contém os resultados obtidos para o k no qual a taxa de acerto foi melhor.

O tempo médio para classificar os padrões das bases foi de aproximadamente 11,3 segundos. Provavelmente um dos fatores responsáveis seja o próprio algoritmo, o qual armazena todos os padrões para então efetuar a classificação. O tamanho das bases aliado ao número de atributos de cada uma delas também foram um dos fatores que contribuíram para o aumento do tempo.

Os valores obtidos para k nos revela a complexidade do problema. Por exemplo, na base diabetes, os dados encontram-se muito próximos entre si, portanto, é natural que um k relativamente alto seja usado. O inverso é observado na base de dados de glândula tireóide, na qual classes encontram-se bem definidas no espaço. As Figura 46 e 47, ilustradas a seguir, obtidas através do simulador WEKA, ilustram os dois casos, respectivamente.

Tabela 12. Classificação das bases de dados utilizando kNN

Base de dados	K	Tempo de treinamento (s)	Taxa de acerto (%)
Doença inflamatória intestinal	3	38	63,4116
Câncer de mama	5	13	95,5202
Diabetes	7	14	73,4375
Doença do fígado	3	2	63,4783
Câncer pulmonar	7	0	65,6250
Glândula tireóide	1	1	97,2093

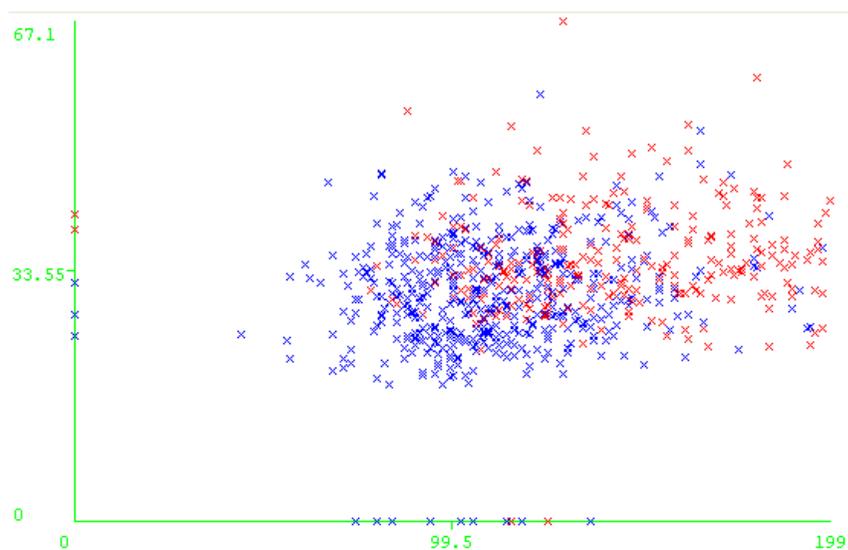


Figura 46. Visualização 2D da distribuição de dados da base diabetes. Pontos azuis representam teste negativo e os vermelhos, teste positivo

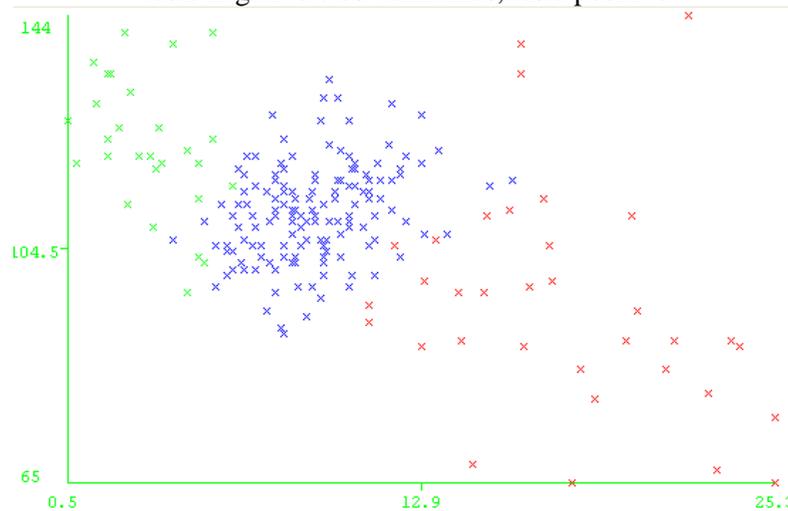


Figura 47. Visualização 2D da distribuição de dados da base glândula tireóide. Pontos azuis representam Normal, os vermelhos, hipertireoidismo e o verde, hipotireoidismo

4.4 Experimentos e resultados utilizando MLP- *backpropagation*

Redes neurais artificiais que usam a topologia MLP e *backpropagation* como algoritmo de aprendizagem apresentam um problema: a determinação do número de camadas ocultas, dos neurônios das dessas camadas, da taxa de aprendizagem e do número de épocas necessárias para o treinamento. Utilizamos apenas uma camada oculta, conforme recomendado na literatura [26]. Quanto aos outros parâmetros, escolhemos os seguintes valores:

- Neurônios da camada escondida: 5, 10 e 50
- Taxa de aprendizagem: 0,01 , 0,05 e 0,1
- Número de épocas: 1000, 10000 e 50000

De posse dos valores acima, 27 (3x3x3) experimentos foram realizados em cada base de dados, ou seja, todas as combinações possíveis dos três parâmetros foram aplicadas. Validação cruzada em 10 blocos foi utilizada. Outra desvantagem dessa técnica advém desses testes, que por vezes são extremamente demorados e cansativos. Na Tabela 13 a seguir encontram-se os melhores resultados obtidos, com os parâmetros ótimos identificados pelo processo de teste-erro.

Conforme se observa a Tabela 13, na base de dados doença inflamatória intestinal, o número de neurônios presentes na camada escondida foi 50, o que é justificado pela complexidade do problema, dado que 25 atributos definem a classificação. Ainda, o tempo necessário para o treinamento atingiu 608 segundos, devido à grande quantidade de padrões e a lentidão do algoritmo de aprendizagem – baixa taxa de aprendizado implica em aumento de tempo de treinamento. Essas observações também explicam os resultados atingidos nas outras bases de dados. A alteração do número de épocas quase não afetou o desempenho de

classificação do algoritmo em nenhuma das bases. A base que sofreu maior variação, nesse sentido, foi a glândula tireóide. A Figura 48, a seguir, confirma essa observação.

Tabela 13. Melhores resultados obtidos nas bases de dados por redes MLP-Backpropagation

Base de dados	Nº de neurônios escondidos	Taxa de aprendizado	Nº de épocas	Tempo de treinamento (s)	Taxa de acerto (%)
Doença inflamatória intestinal	50	0,01	1000	608	67,4907
Câncer de mama	10	0,05	1000	27	95,3757
Diabetes	10	0,01	1000	165	77,4740
Doença do fígado	5	0,1	1000	10	69,8551
Câncer pulmonar	10	0,05	1000	2	46,8750
Glândula tireóide	5	0,1	1000	19	93,0233

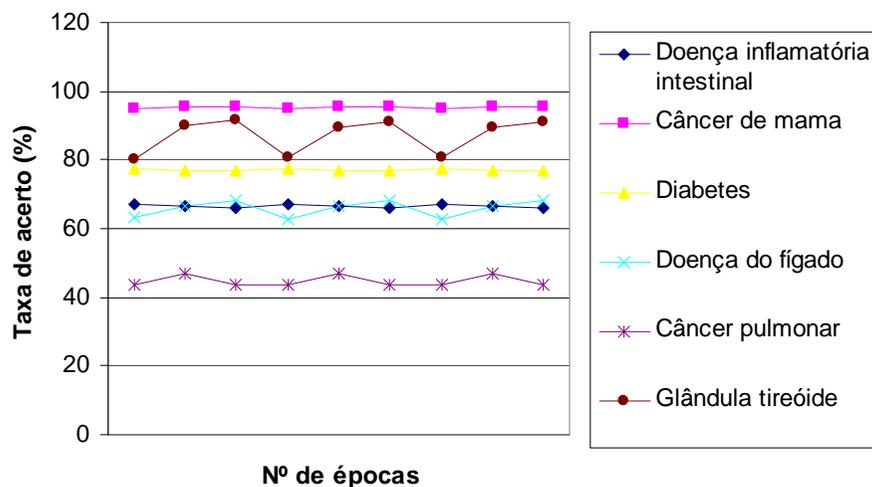


Figura 48. Taxa de acerto (%) versus Nº de épocas

4.5 Experimentos e resultados utilizando RBF-DDA

A aplicação do algoritmo DDA, implica na escolha de dois parâmetros, θ^+ e θ^- . Segundo desenvolvedores do algoritmo, uma boa escolha para o valor do parâmetro θ^+ é 0,4, por isso, escolhemos não variar o parâmetro nas execuções do algoritmo. Para o valor de θ^- , escolheu-se variá-lo de 10^{-10} até 1, ou seja, 10^{-10} , 10^{-9} , 10^{-8} , e assim sucessivamente. Validação cruzada em 10 blocos foi utilizada. Os melhores resultados obtidos são apresentados na Tabela 14 a seguir.

Depreende-se, pela observação da Tabela 14, que redes RBF-DDA obtiveram uma média de tempo para o treinamento de aproximadamente 3,17 segundos (levando-se em consideração apenas os melhores resultados). Observou-se, também, que, ao longo dos experimentos, o valor assumido pelo θ^- foi inversamente proporcional ao número de neurônios da camada escondida, ou seja, quanto menor o valor daquele parâmetro mais neurônios (unidades RBFs) são necessários para cobrir os padrões. Isso realmente era esperado, pois a diminuição da largura das gaussianas

implica necessariamente no aumento da sua quantidade. Esse comportamento encontra-se ilustrado na Figura 49 a seguir. A Figura 50, a seguir, apresenta a diminuição da taxa de acerto quanto maior for o valor assumido pelo θ^- , apesar de que um valor maior de θ^- não implica em pior taxa de acerto, conforme se percebe na figura mencionada. Nessa, percebe-se também que nas bases de dados de doença inflamatória intestinal e doença do fígado, a variação do limiar negativo no intervalo $[0,1, 1]$ quase não alterou a porcentagem de acerto. Por fim, o número de épocas também foi o esperado, pois o algoritmo DDA é rápido, conforme observou o próprio autor do mesmo [18].

Tabela 14. Melhores resultados obtidos nas bases de dados por redes RBF-DDA

Base de dados	Nº médio de neurônios escondidos	θ^-	Nº médio de épocas	Tempo de treinamento (s)	Taxa de acerto (%)
Doença inflamatória intestinal	389,2	0,3	5,2	5	61,22
Câncer de mama	187	0,0001	4	4	94,66
Diabetes	596,4	0,3	4,3	3	75,25
Doença do fígado	268	0,2	4,1	3	58,99
Câncer pulmonar	28,8	0,0001	3	1	42,67
Glândula tireóide	128,2	0,0001	4	3	97,33

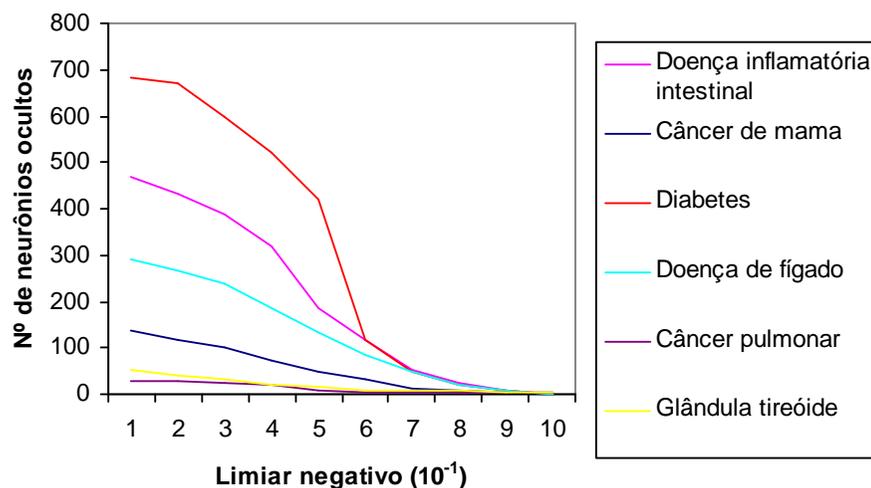


Figura 49. Diminuição do nº de neurônios da camada oculta da rede RBF-DDA decorrente do aumento do limiar negativo

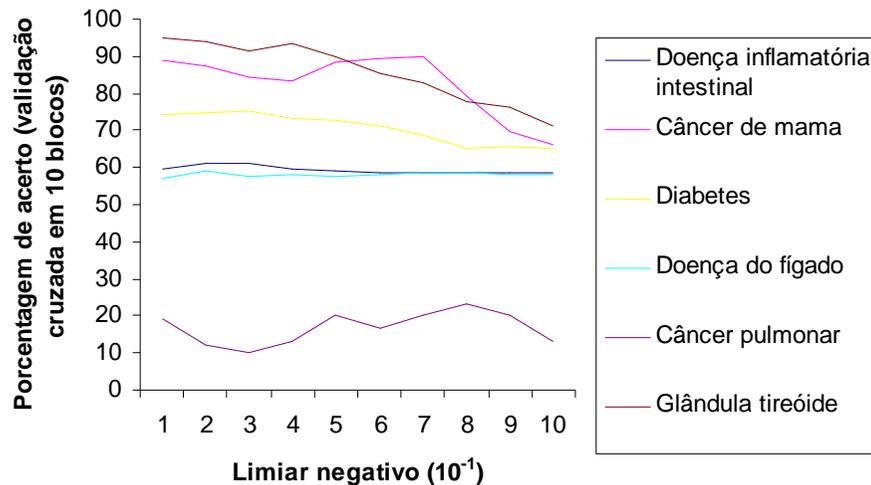


Figura 50. Acerto *versus* θ^-

4.6 Comparações entre as técnicas utilizadas

Analisando o potencial de desempenho de classificação das técnicas utilizadas nas bases de dados do presente estudo, verifica-se que máquinas de vetores suporte tiveram melhor desempenho que as outras técnicas em quatro das seis bases, a saber:

- Doença inflamatória intestinal, com 68,1088% de acerto
- Câncer de mama, com 96,2428%
- Diabetes, com 77,604% e
- Doença de fígado, com 73,2258%

Não apenas o percentual de acerto foi superior, mas também o tempo necessário para o treinamento e classificação, embora a complexidade tenha sido maior. Coincidentemente, os melhores resultados para SVMs foram obtidos naquelas bases que possuíam maior quantidade de padrões, o que não implica necessariamente que essa técnica seja melhor apenas quando o número de exemplos for suficientemente grande. Esse resultado já era esperado, pois com SVM temos a garantia de encontrar uma solução ótima, ao contrário das outras técnicas utilizadas. Outro fator a ser considerado, é que no método SVM, as classes são separadas duas a duas, favorecendo um melhor desempenho, ao custo de maior complexidade.

Na base de dados de doença inflamatória, conforme a Figura 51 ilustrada a seguir, uma comparação é efetuada entre as várias técnicas utilizadas. Máquina de vetor suporte superou as outras técnicas tanto em percentual de acerto, quanto em tempo total do experimento (medido em segundos) ao custo de uma alta complexidade. De fato, SVM gerou uma complexidade apenas inferior à técnica do vizinho mais próximo. No quesito complexidade, ainda, observamos que redes MLP obtiveram o melhor resultado, embora seu tempo de treinamento tenha sido bem superior em relação às outras técnicas, bem como o percentual de acerto. A técnica kNN gerou alta complexidade, desde que ela necessita armazenar todos os exemplos presentes na base para que ocorra o processo classificatório.

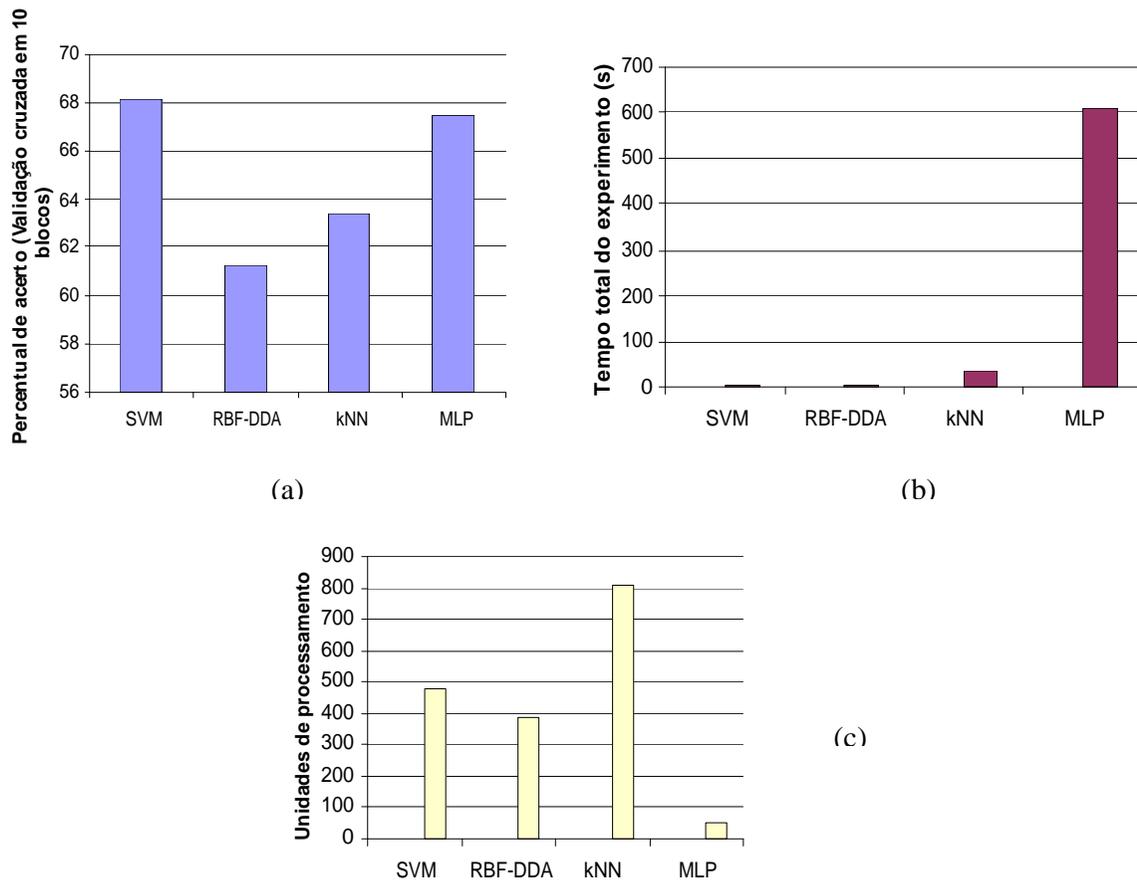


Figura 51. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados doença inflamatória intestinal

Uma comparação entre as técnicas na base de dados de câncer de mama é ilustrada na Figura 52 a seguir. Máquina de vetor suporte novamente superou todas as outras técnicas tanto em percentual de acerto, quanto em tempo total do experimento ao custo de alta complexidade. Em relação à complexidade depreendemos que redes MLP geraram uma menor quantidade de unidades de processamento, embora o tempo de treinamento tenha sido o maior dentre as outras técnicas. A técnica kNN gerou a mais alta complexidade, motivado pelo armazenamento total dos padrões. Essas observações também estão presentes nas figuras 53 e 54, as quais representam as comparações entre as técnicas nas bases de dados diabetes e doença do fígado, respectivamente. As figuras estão ilustradas a seguir.

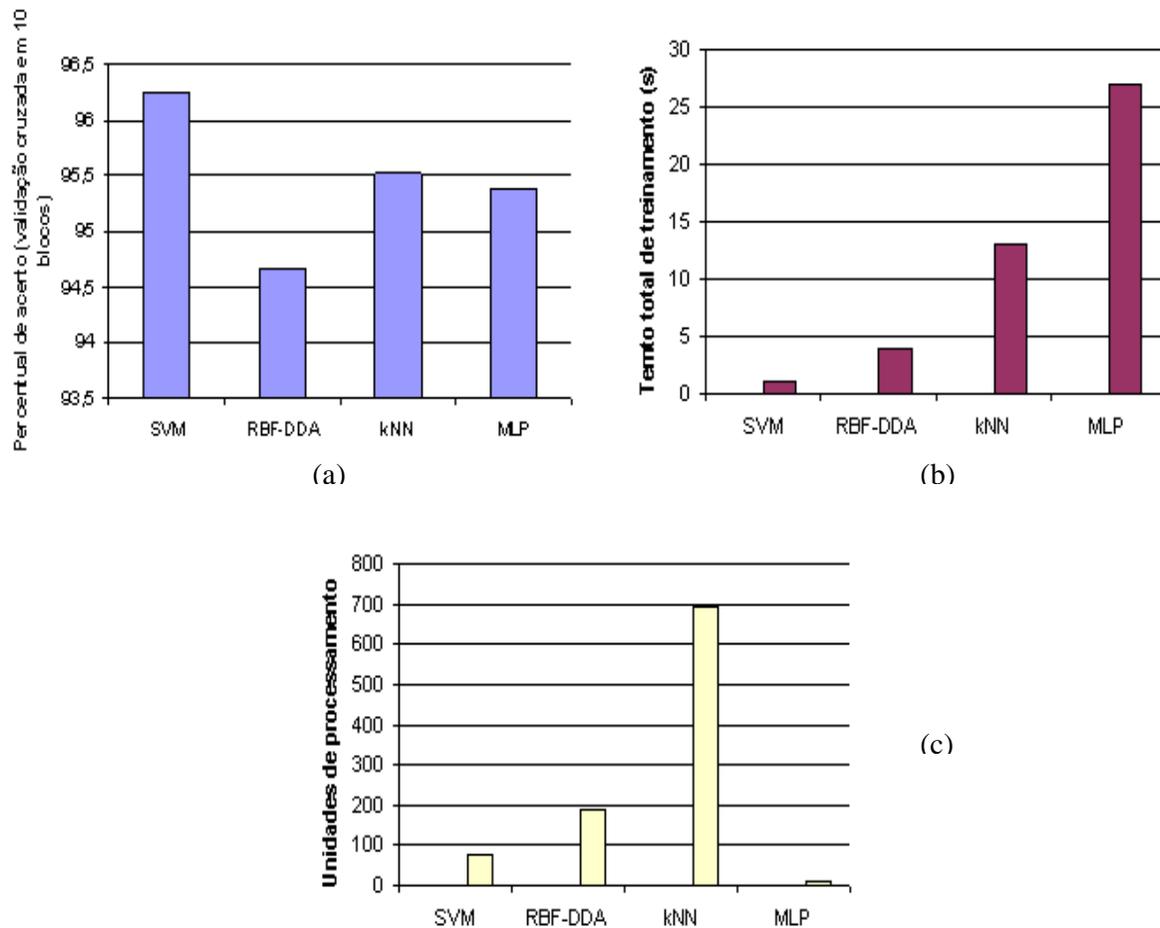


Figura 52. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados câncer de mama

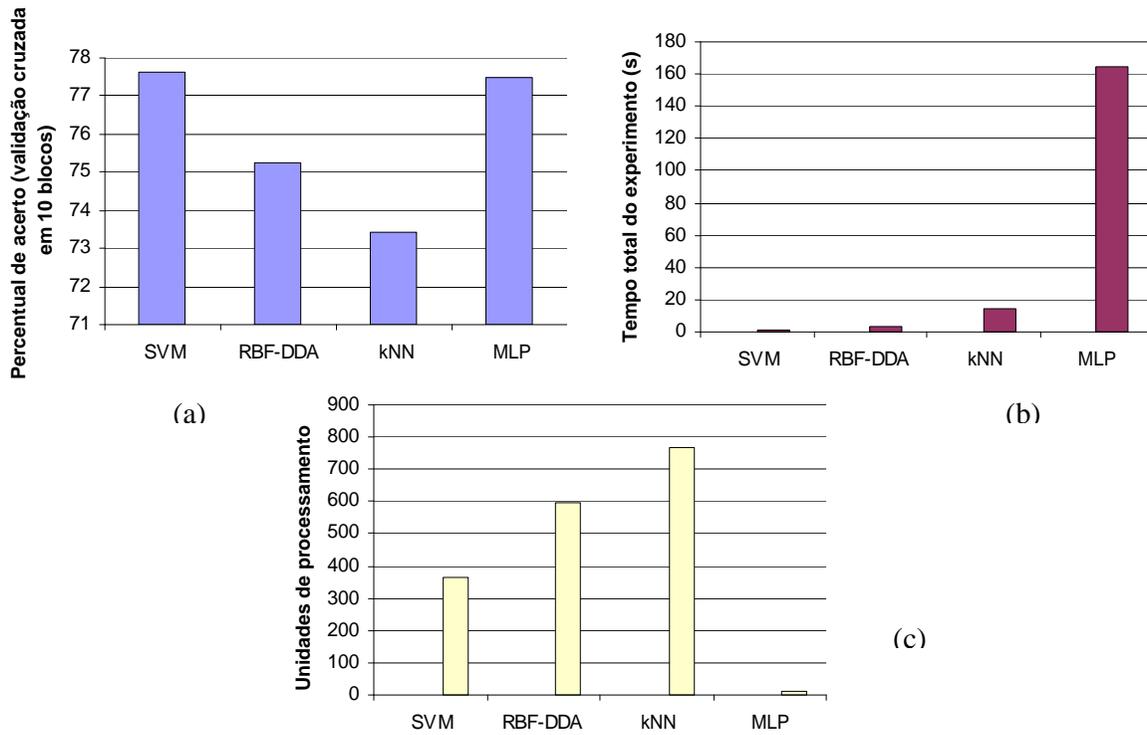


Figura 53. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados diabetes

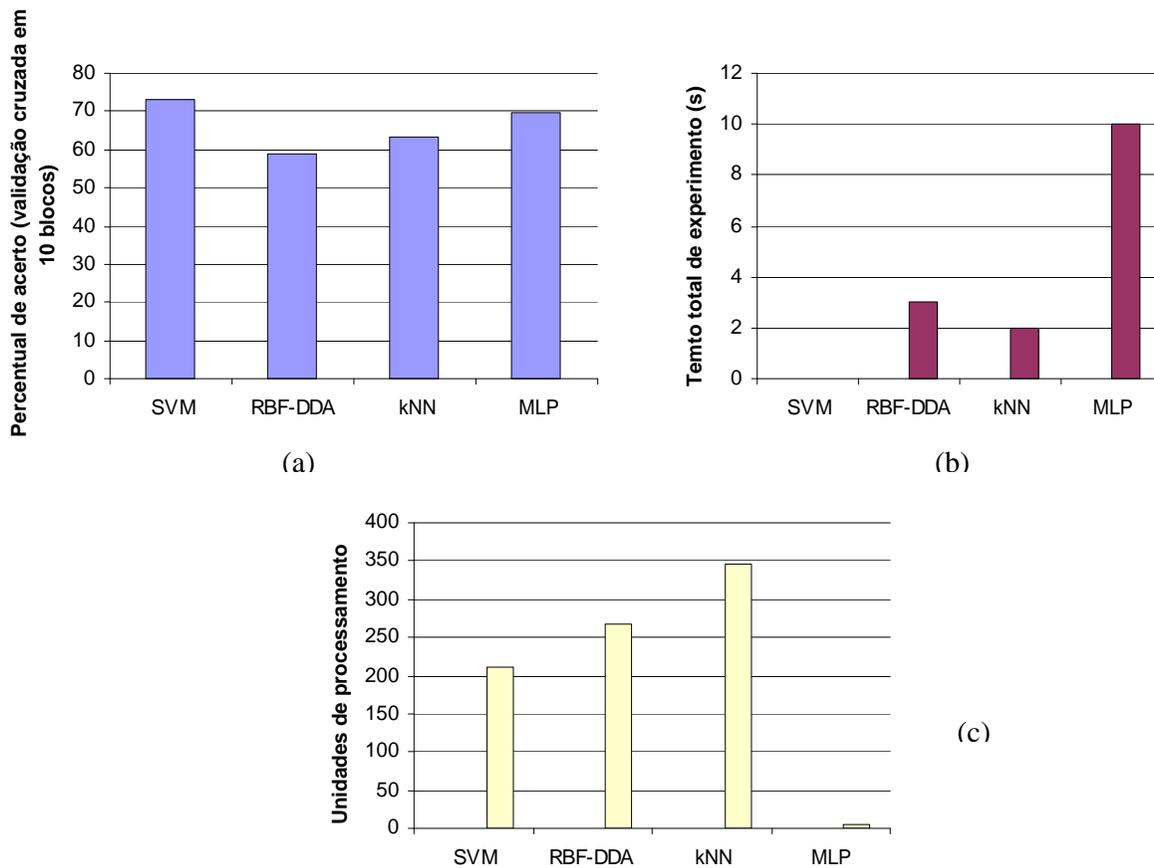


Figura 54. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados doença do fígado

A técnica de RBF-DDA alcançou melhor resultado apenas em uma base de dados, no caso, a da glândula tireóide, com 97,33% de percentual de acerto (validação cruzada). Pode-se creditar esse resultado devido à sua capacidade de tolerância a ruídos, o que torna sua aplicação em bases de dados ruidosas algo interessante. Similar ao resultado obtido com SVMs, houve um acréscimo de complexidade na rede gerada em contrapeso à classificação. A figura 55 a seguir ilustra o citado.

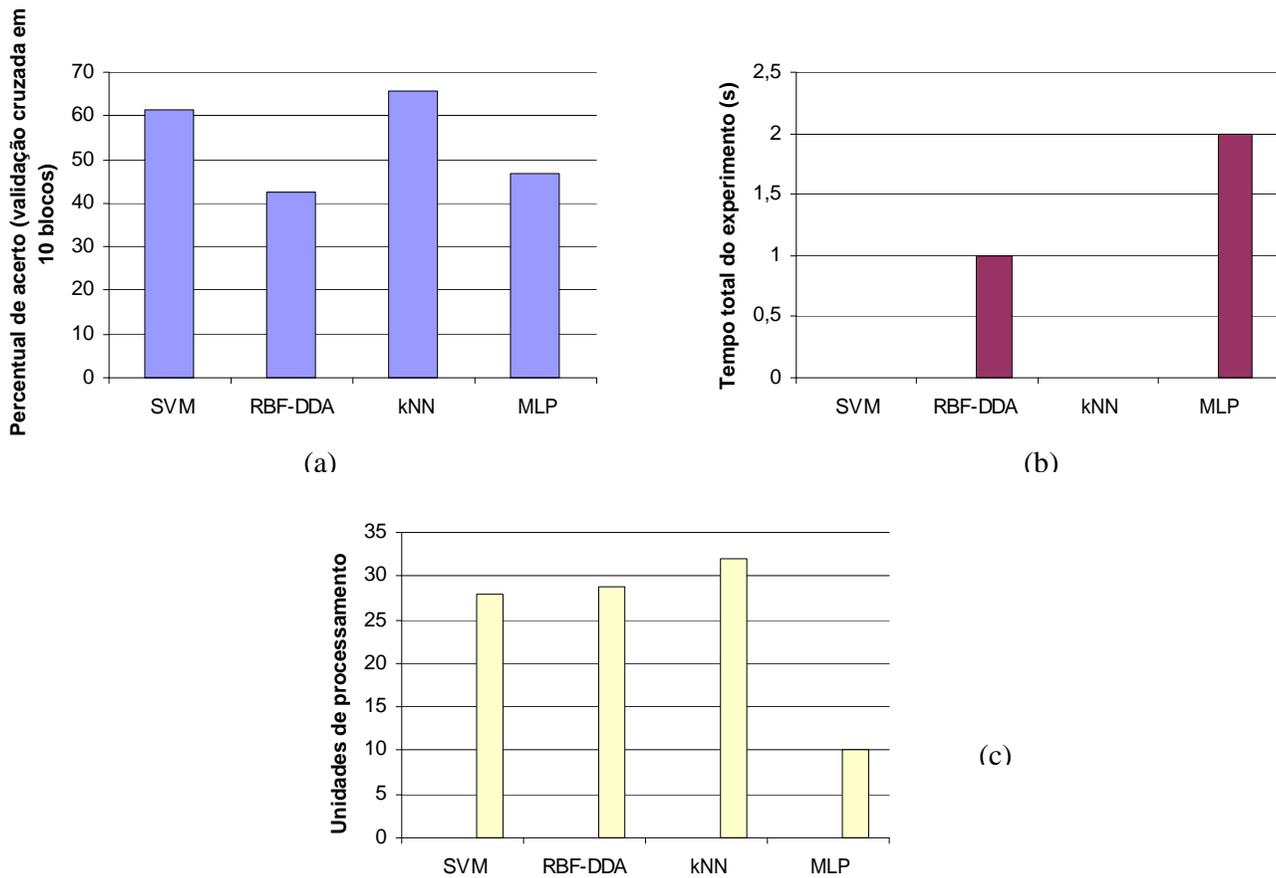


Figura 55. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados câncer pulmonar

Por fim, kNN obteve a melhor solução na base de câncer pulmonar com um percentual de acerto (validação cruzada) de 65,625%. Outra vantagem da técnica, observada nessa base, foi o tempo necessário para o reconhecimento dos padrões, aproximadamente 0 (zero) segundos, ao custo do armazenamento computacional total desses mesmos padrões. A Figura 56, a seguir, compara a base em relação a quantidade de unidades de processamento, tempo de treinamento e taxa de acerto na classificação.

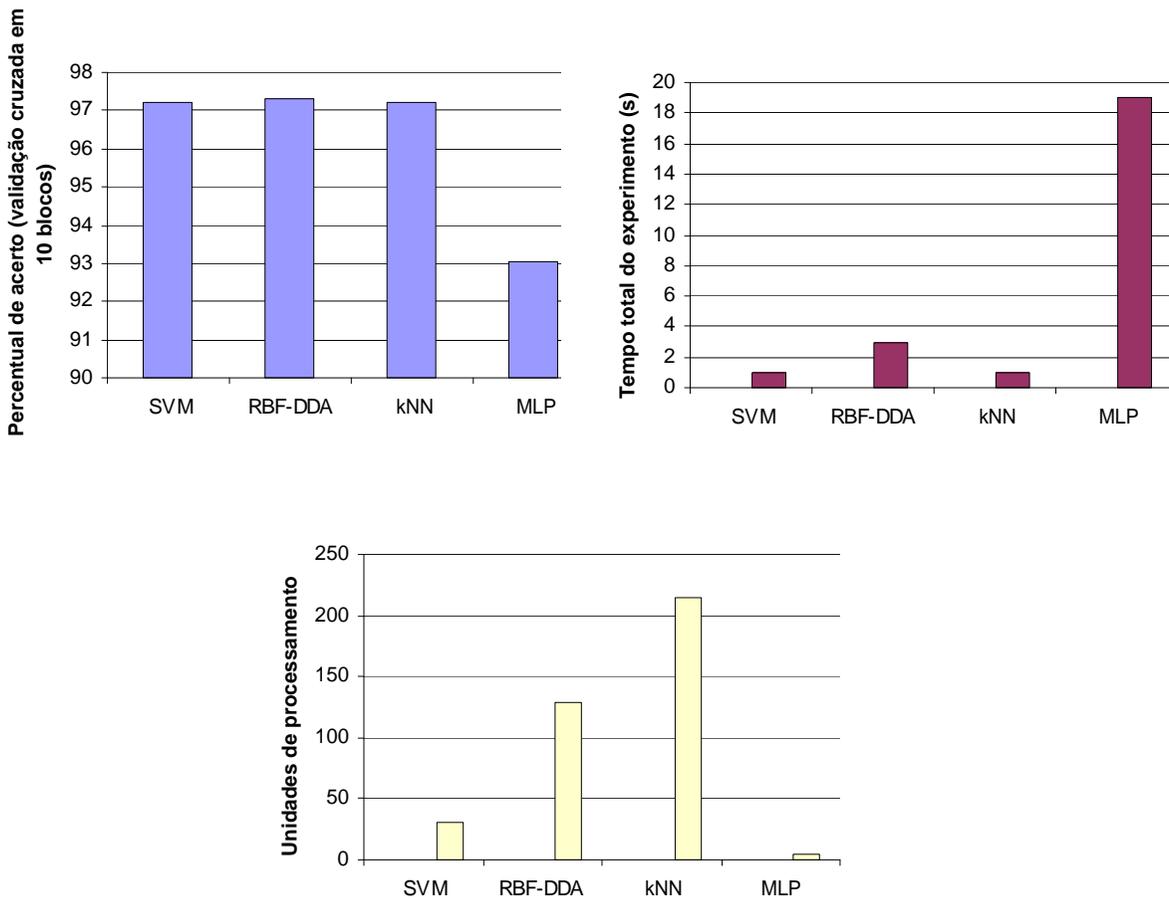


Figura 56. Percentual de acerto de validação cruzada (a), tempo total do experimento em segundos (b) e unidades de processamento (c) na base de dados glândula tireóide

A Tabela 15, a seguir, apresenta as bases de dados e as técnicas que apresentaram melhores resultados. Com ela, percebemos que SVM apresentou melhores resultados em quatro das seis bases. Embora SVM não tenha atingido os melhores resultados em todas as bases, sua porcentagem de acerto de classificação nas duas bases restantes, câncer pulmonar e glândula tireóide, foi bastante próxima das técnicas que obtiveram os melhores índices de reconhecimento, kNN e RBF-DDA, respectivamente. Na questão de tempo total de experimento, SVM superou todas as técnicas em todas as bases. Em relação à complexidade, MLP atingiu melhores resultados em todas as bases, seguida por SVM, RBF-DDA e kNN. Observe que os valores ilustrados na coluna *Nº de padrões armazenados* são aproximações dos resultados obtidos.

Tabela 15. Melhores resultados obtidos nas bases de dados

Base de dados	SVM	RBF-DDA	kNN	Nº de padrões armazenados	Tempo para treinamento (s)	Porcentagem de acerto (validação cruzada)
Doença inflamatória intestinal	X			477	2	68,1048
Câncer de mama	X			74	1	96,2428
Diabetes	X			364	1	77,604
Doença do fígado	X			211	0	73,2258
Câncer pulmonar			X	32	0	65,625
Glândula tireóide		X		128	3	97,33

Conclusões e Trabalhos Futuros

O presente estudo buscou realizar uma comparação de várias técnicas de aprendizado de máquina aplicadas no reconhecimento de padrões em bases de dados médicas. Dentre as técnicas utilizadas, encontram-se máquinas de vetor suporte, técnica do vizinho mais próximo e redes neurais artificiais do tipo RBF- DDA e MLP-*backpropagation*.

Seis bases reais de dados médicas foram utilizadas, todas elas resultados de coletas de pacientes e exames laboratoriais, as quais encontram-se disponíveis online para realização de estudos como este.

Ferramentas de simulação foram empregadas para a execução dos experimentos das técnicas de aprendizado de máquina. Redes RBF-DDA foram utilizadas através do simulador SNNS, enquanto o LIBSVM foi usado para realizar os testes com SVMs. Para redes MLP-*Backpropagation* e para kNN, o simulador WEKA foi aplicado. Para todos os experimentos, validação cruzada em 10-blocos foi utilizada para tornar os resultados mais confiáveis.

Os resultados alcançados com os experimentos foram comparados em relação a: complexidade das redes geradas, tempo de classificação e treinamento e percentual de acerto no conjunto de teste.

No geral, redes MLP obtiveram maus resultados e longo tempo de treinamento, mas geraram complexidade menor que as outras técnicas. A técnica do vizinho mais próximo possui a desvantagem de armazenar todos os dados para executar o algoritmo de aprendizado. No corrente trabalho, kNN realizou treinamentos relativamente curtos com uma boa taxa de classificação. RBF-DDA em média obteve redes de alta complexidade, apesar do baixo tempo de treinamento e classificação e um bom índice de reconhecimento. Máquinas de vetores suporte foi a técnica que obteve melhores resultados, tanto na questão de classificação de padrões, quanto no tempo necessário para treiná-los e classificá-los.

A partir dos resultados obtidos, determinou-se que máquina de vetor suporte foi a técnica que apresentou melhores resultados em quatro das seis bases que compuseram o estudo, a saber: doença inflamatória intestinal, câncer de mama, diabetes e doença de fígado. Para as outras duas bases, câncer pulmonar e glândula tireóide, as técnicas kNN e RBF-DDA obtiveram os melhores resultados, respectivamente. Observou-se também que, apesar de não ter alcançado o melhor índice de classificação em todas as bases, SVMs apresentaram respostas muito próximas das técnicas de kNN e RBF-DDA nas bases de câncer pulmonar e glândula tireóide.

Propomos que, para trabalhos futuros, outras técnicas de aprendizado de máquina sejam empregadas nas bases de dados que compuseram o estudo. Sistemas híbridos e máquinas de comitê [55] podem também ser utilizadas para combinar as mais diversas técnicas a fim de tentar melhorar a taxa de acerto nas bases. Técnicas de seleção de atributos [56] mais relevantes é outra proposta que têm se tornado bastante presente em publicações e que poderiam ser aplicadas em outro estudo desse cunho.

Outras bases de dados também podem e devem ser utilizadas, para afirmar a capacidade que técnicas de inteligência artificial tem para reconhecer padrões das mais variadas áreas médicas.

Sistemas de apoio a diagnóstico é outra alternativa de pesquisas futuras. A real implementação de um SAD e o impacto que ele possa vir a causar no atendimento de pacientes e no auxílio de diagnósticos de patologias é outro tema motivante. Sistemas de auxílio pedagógico podem também ser desenvolvidos com o objetivo de ajudar profissionais da área de saúde que estão em formação acadêmica ou extra-curricular.

Uma proposta final, mas não menos importante, é que o número de padrões presente nas bases de dados seja incrementado, através de um processo de coleta rigoroso e eficaz.

Bibliografia

- [1] LEDLEY, R. S. e LUSTED, L. B. *Reasoning foundations of medical diagnosis*. Science. v. 130, p. 9-21. 1959.
- [2] PAETZ, J. *Finding optimal decision scores by evolutionary strategies*. Artificial Intelligence in medicine. v. 32, p. 85-95. 2004.
- [3] TUNG, W. L. e QUEK, C. *GenSo-FDSS: a neural-fuzzy decision support system for pediatric ALL cancer subtype identification using gene expression data*. Artificial intelligence in medicine. v. 33, p. 61-88. 2005.
- [4] BOEGL, K. et al. *Knowledge acquisition in the fuzzy knowledge representation framework of a medical consultation system*. Artificial intelligence in medicine. v 30, p. 1-26. 2004.
- [5] BAXT, W. G. *Use of an artificial neural network for data analysis in clinical decision making: the diagnosis of acute coronary occlusion*. Neural computing. v. 2, p. 480-489. 1990.
- [6] HAYKIN, S. *Redes Neurais: princípios e prática*. 2. ed. Trad. Paulo Martins Engel. Porto Alegre: Bookman, 2001. 900 p.
- [7] FRIZE, M. et al *Clinical decision support systems for intensive care units: using artificial neural networks*. Medical engineering & physics. v. 23, p. 217-225. 2001.
- [8] MARTINELLI, E. *Extração de conhecimentos de redes neurais*. 1999. 111f. Dissertação (Mestrado em Ciências de computação e matemática computacional) - Escola de Engenharia de São Carlos/EESC, Universidade de São Paulo.
- [9] LISBOA, P.J.G. *A review of evidence of health benefit from artificial neural networks in medical intervention*. Neural Networks. v. 15, p. 11-39. 2002.
- [10] BAXT, W.G. *Application of neural networks to clinical medicine*. Lancet. v. 346, p. 1135-1138. 1995.
- [11] BAXT, W.G. *Artificial neural network to identify acute myocardial infarction-reply*. Lancet. v. 347, p. 551. 1996.
- [12] BARACH, P. e SMALL, P.D. *Reporting and preventing medical mishaps: lessons from non-medical near-miss reporting systems*. British Medical Journal. v. 320, 245-252. 2000.
- [13] HORN, W. *AI in medicine on its way from knowledge-intensive to data-intensive systems*. Artificial intelligence in medicine. v. 23, p. 5-12. 2001.
- [14] BLAKE, C. e MERZ, C. *UCI repository of machine learning databases*. Disponível em <<http://www.ics.uci.edu/~mllearn/MLRepository.html>>. Acesso em 05 de junho de 2005.
- [15] RUHE, D. *Artificial Neural Networks in Medicine and Biology Society*. Disponível em <http://www.phil.gu.se/ann/annworld.html> . Acesso em 05 de junho de 2005.
- [16] LILLEHAUG, S. e LAJOIE, S. P. *AI in medical education - another grand challenge for medical informatics*. Artificial Intelligence in medicine. v. 12, p. 197-225. 1998.

- [17] ZELL, A. et al. *Stuttgart neural network simulator*. Software disponível em: <<http://www-ra.informatik.uni-tuebingen.de/SNNS>>. Acesso em 25 de março de 2005.
- [18] BERTHOLD, M. R. e DIAMOND, J. *Boosting the performance of RBF networks with dynamic decay adjustment*, in: G. Tesauro, D. Touretzky, T. Leen, (Eds.), *Advances in Neural Information Processing*, v. 7, MIT Press, New York, 1995, p. 521–528.
- [19] OLIVEIRA, A. L. I., MELO, B. J. M, MEIRA, S. R. L. *Integrated Method for constructive training of radial basis function networks*. *Electronics Letters*. v. 41, n. 7, p. 429-430. Março, 2005.
- [20] OLIVEIRA, A.L.I., MELO, B. J. M, MEIRA, S. R. L. *Improving constructive training of RBF networks through selective pruning and model selection*. *Neurocomputing*. v. 64, p. 537-541. Março, 2005.
- [21] CHANG, C.C. e LIN, C.J. *LIBSVM: a library for support vector machines*. Software disponível em: <<http://www.csie.ntu.edu.tw/~cjlin/libsvm>>. Acesso em 25 de março de 2005.
- [22] V., D.S.A. *Advanced support vector machines and kernel methods*. *Neurocomputing*. v. 55, p. 5-20. Setembro, 2003.
- [23] CHANG, R. et al. *Support Vector Machines for Diagnosis of Breast Tumors on US Images*. *Academic Radiology*. v. 10, n. 2, p. 189-197. Fevereiro, 2003.
- [24] WAIKATO ENVIRONMENT FOR KNOWLEDGE ANALYSIS. Software disponível em <<http://www.cs.waikato.ac.nz/ml/weka/>>, visitado em 25/10/2005
- [25] OLIVEIRA, A. L. I., MELO, B. J. M, MEIRA, S. R. L. *Improving RBF-DDA Performance on Optical Character Recognition through Parameter Selection*. *Pattern Recognition*, in 17th International Conference on (ICPR'04). v. 4, p. 625-628. 2004.
- [26] BRAGA, A. P., CARVALHO, A. P. L. F., LUDERMIR, T. B. *Redes Neurais Artificiais Teoria e Aplicações*. Livros Técnicos e Científicos Editora, Rio de Janeiro, 2000.
- [27] CYBENKO, G. *Contínuos valued neural networks with two hidden layers are sufficient*. Technical report, Departament of computer science, Tufts University, 1988.
- [28] CYBENKO, G. *Approximation by superpositions of a sigmoid function*. *Mathematics of control, signal and systems*. v. 2, p 303-314. 1989.
- [29] RUSSEL, S. e NORVIG, P. *Inteligência Artificial*. Editora Campus e Editora Elsevier, 2004.
- [30] RUMELHART, D. E. e McCLELLAND, J. L. *Parallel Distributed Processing*. v. 1. The MIT Press, 1986.
- [31] FAHLMAN, S. E. *An empirical study of learning speed in backpropagation networks*. Technical report, Carnegie Mellow University, 1988.
- [32] HAGAN, M e MENHAJ, M. *Training feedforward networks with the Marquardt algorithm*. *IEEE transactions on neural networks*. v. 5, p. 989-993. Novembro. 1994.
- [33] PEARLMUTTER, B. *Gradient descent: second order momentum and saturation error*. Em MOODY, S. et al, *Advances in neural information processing systems 2*. p. 887-894. Morgan Kaufmann, 1992.
- [34] BATTITI, R. *First and second-order methods for learning: between steepest descent and Newton's method*. Technical report, University of Trento, 1991.
- [35] RIEDMILLER, M. *Rprop – description and implementation details*. Technical report, University of Karlsruhe, 1994.
- [36] PRECHELT, L. *Proben1 - a set of neural networks benchmark problems and benchmarking rules*. Technical Report 21/94, Universität Karlsruhe, Germany. 1994.
- [37] PONTIL, M. e VERRI, A. *Properties of support vector machines*. v. 10, p. 955-974. *Neural Computation*. 1998.
- [38] CRISTIANINI, N. e SHAWE-TAYLOR, J. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.

- [39] LIMA, A. R. G. *Máquinas de vetores suporte na classificação de imagens digitais*. 2002. 81f. Dissertação. Universidade Federal do Ceará.
- [40] HUNT, D. L. et al. *Effects of computer-based clinical decision support systems on physician performance and patient outcomes*. JAMA. v. 280, p. 1339-1346. 1998.
- [41] MACMILLAN, N. A. e CREELMAN, C. D. *Detection theory: a User's guide*. Lawrence Erlbaum Associates, 2005.
- [42] BAXT, W. G. e WHITE, H. *Bootstrapping confidence intervals for clinical input variable effects in a network trained to identify the presence of acute myocardial infarction*. Neural computation. v. 7, p. 624-638. 1995.
- [43] WEINER, M. G. e PIFER, E. *Computadorized decision support and the quality of care*. Managed Care. v. 9, p.41-51. 2000.
- [44] DVORKIN, D. et al. *SiMCAL 1 algorithm for analysis of gene expression data related to the phosphatidylserine receptor*. Artificial Intelligence in Medicine. v. 35, p. 49-60. Setembro, 2005.
- [45] NATTKEMPER, T. W. et al. *Evaluation of radiological features for breast tumour classification in clinical screening with machine learning methods*. Artificial Intelligence in Medicine. v. 34, p. 129-139. Setembro, 2005.
- [46] HUANG, T. M. e KECCMAN, V. *Gene extraction for cancer diagnosis by support vector machines-An improvement*. Artificial Intelligence in Medicine. v. 35, p. 185-194. Setembro, 2005.
- [47] LU, C. et al. *Preoperative prediction of malignancy of ovarian tumors using least square support vector machines*. Artificial Intelligence in Medicine. v. 28, p. 281-306. Julho, 2005.
- [48] FERNANDES, A. M. R. et al. *Inteligência artificial aplicada à saúde*. Visual Books, 2004.
- [49] UNDERWOOD, J. C. E. *Tumours: benign and malignant*. General and Systematic Pathology. Underwood, J. C. E. p. 223-246. Underwood JCE Editors. 1992.
- [50] TROTT, P. A. *Aspiration cytodiagnosis of the breast*. Diagn. Oncol. v. 1, p. 79-87. 1991.
- [51] HSU, C., CHANG, C., LIN, C. *A practical guide to support vector classification*. Disponível em < www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>. Acesso em 5 de junho de 2005.
- [52] DELEN, D. et al. *Predicting breast cancer survivability: a comparison of three data mining methods*. Artificial Intelligence in Medicine. v. 34, p. 113-127. Junho, 2005.
- [53] DOUNIAS, G. e LINKENS, D. *Adaptative systems and hybrid computational intelligence in medicine*. Artificial Intelligence in Medicine. v. 32, p. 183-194. Novembro, 2005.
- [54] PILA, A. D. *Seleção de atributos relevantes para aprendizado de máquina utilizando a abordagem rough sets*. 2001. 163f. Dissertação (Mestrado em Ciências de computação e matemática computacional) – Instituto de ciências e matemáticas e de computação/ICM, Universidade de São Paulo.

Apêndice A

Glossário

Algoritmos genéticos (AG) - são uma família de modelos computacionais inspirados na evolução, que incorporam uma solução potencial para um problema específico numa estrutura semelhante a de um cromossomo e aplicam operadores de seleção e "cross-over" a essas estruturas de forma a preservar informações críticas relativas à solução do problema. Normalmente os AG's são vistos como otimizadores de funções, embora a quantidade de problemas para o qual os AG's se aplicam seja bastante abrangente.

Aprendizado de máquina - é uma área da inteligência artificial cuja meta é desenvolver técnicas computacionais e/ou sistemas capazes de adquirir conhecimento e aprender.

Lógica difusa - é uma generalização da lógica clássica que admite valores lógicos fracionários, ao contrário da lógica tradicional que admite apenas o par oposto *falso/verdadeiro, tudo/nada*, etc. A lógica difusa é uma tentativa de implementar níveis intermediários de verdade.

Neurônio artificial - são as unidades básicas de processamento da informação, projetadas para simular o comportamento de neurônios biológicos.

Reconhecimento de padrões - é a ciência que trata da classificação e descrição de objetos. Por exemplo, a partir do Peso e da Altura de um indivíduo, é possível classificá-lo como gordo ou magro.

Sistemas híbridos - é um sistema dinâmico que integra componentes de natureza discreta (ou digital) e/ou componentes de natureza contínua (ou analógica).

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.