UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

MESTRADO ACADÊMICO EM ENGENHARIA DE TELEINFORMÁTICA

NATANAEL RODRIGUES DA SILVA

CLASSIFICADORES DE PADRÕES RANDOMIZADOS PARA DETECÇÃO DE CRISES EPILÉPTICAS: UMA AVALIAÇÃO CRÍTICA

FORTALEZA

2017

NATANAEL RODRIGUES DA SILVA

CLASSIFICADORES DE PADRÕES RANDOMIZADOS PARA DETECÇÃO DE CRISES EPILÉPTICAS: UMA AVALIAÇÃO CRÍTICA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Engenharia de Teleinformática

Orientador: Prof. Dr. Guilherme de Alencar Barreto

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S581c Silva, Natanael Rodrigues da.
Classificadores de padrões randomizados para detecção de crises epilépticas: uma avaliação crítica / Natanael Rodrigues da Silva. – 2017.
132 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2017.
Orientação: Prof. Dr. Guilherme de Alencar Barreto.

1. crises epiléticas. 2. periodograma de Welch. 3. coeficientes LPC. 4. curvas ROC. I. Título.
CDD 621.38

NATANAEL RODRIGUES DA SILVA

CLASSIFICADORES DE PADRÕES RANDOMIZADOS PARA DETECÇÃO DE CRISES EPILÉPTICAS: UMA AVALIAÇÃO CRÍTICA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Engenharia de Teleinformática

Aprovada em: 19 de Dezembro de 2017

BANCA EXAMINADORA

Prof. Dr. Guilherme de Alencar Barreto (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dra. Michela Mulas
Universidade Federal do Ceará (UFC)

Prof. Dr. Pedro Pedrosa Rebouças Filho
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Dedico este trabalho aos meus pais Lucieudo e Emília.

AGRADECIMENTOS

Primeiramente, agradeço à Deus por toda sabedoria e benção em todos os momentos.

Aos meus pais, Lucieudo e Emília, pela exemplar educação, amor, ensinamentos, apoio, carinho e por sempre acreditarem em todos meus sonhos, isso tudo foi fundamental para que eu conseguisse alcançar com sucesso cada um deles.

Ao Prof. Dr. Guilherme de Alencar Barreto, por sua excelente orientação, incentivo e paciência. Estando sempre presente, disposto a ensinar e ajudar, sua orientação e transmissão de ensinamentos foi crucial para o sucesso desse trabalho.

À Prof. Dra. Michela Mulas e ao Prof. Dr. Pedro Pedrosa Rebouças Filho por aceitarem o convite de fazer parte da banca avaliadora.

Aos amigos e familiares, por estarem sempre presente, por dividir as alegrias e tristezas do dia-a-dia, sou muito grato a todos por poder compartilhar tantos momentos.

Aos demais professores, corpo técnico, administrativo e terceirizados da UFC e do laboratório CENTAURO sempre muito prestativos.

À CAPES, pelo apoio financeiro com a manutenção da bolsa de auxílio.

“Nós estamos presos à tecnologia quando o que
nós mais queremos é algo que apenas funcione.”
(Douglas Adams)

RESUMO

Nesta dissertação, avaliamos os desempenhos de classificadores de padrões randomizados na tarefa de detecção de crises epiléticas a partir de sinais de EEG. Nosso objetivo é investigar se essa nova classe de métodos de aprendizagem de máquinas tem desempenho superiores aos de classificadores lineares e não lineares convencionais, como o MQ, MLP e o SVM, em tarefas de classificação de crises epiléticas utilizando sinais de EEG. A motivação para o trabalho vem da observação de que a recente onda de aplicações envolvendo classificadores randomizados tende a reportar somente resultados positivos, nos quais estes métodos sempre alcançam desempenhos equivalentes ou superiores aos obtidos por classificadores convencionais. Uma avaliação abrangente é realizada e os resultados corroboram nossa hipótese de que os classificadores randomizados geralmente não apresentam resultados superiores aos produzidos por classificadores convencionais não lineares bem treinados. Além disso, os desempenhos de classificadores randomizados são mais dependentes do método de extração de características utilizado do que os não randomizados.

Palavras-chave: Crises epiléticas. Periodograma de Welch. Coeficientes LPC. Curvas ROC.

ABSTRACT

In this dissertation, we evaluated the performance of randomized pattern classifiers in the task of detecting epileptic seizures from EEG signals. Our aim is to investigate whether this new class of machine learning methods performs better than conventional linear and nonlinear classifiers such as MQ, MLP and SVM in epileptic seizures recognition tasks with EEG data. The motivation for the work comes from the observation that the recent wave of applications involving random classifiers tends to report only positive results, in which these methods always reach equivalent or superior performances to those obtained by conventional classifiers. A comprehensive assessment is conducted and the results corroborate our hypothesis that randomized classifiers generally do not present better results than those produced by well-trained conventional nonlinear classifiers. In addition, the performances of randomized classifiers are more dependent on the method of extraction of characteristics used than the non-randomized ones.

Keywords: Randomized classifiers. Epileptic seizures. Welch's periodogram. LPC coefficients. ROC curves.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AC	Acurácia
ANN	Rede Neural Artificial
AR	Autoregressivo
DB	Decibel
DFT	Transformada discreta de Fourier
DWT	Transformada discreta de Wavelet
ECG	Eletrocardiograma
EDF	European Data Format
EEG	Eletroencefalograma
EF	Eficiência
ELM	Extreme Learning Machine
EP	Especificidade
FFT	Transformada Rápida de Fourier
FMRI	Ressonância Magnética Funcional
LPC	Codificação de Predição linear
MCC	Coeficiente de Correlação de Matthews
MEG	Magnetoencefalograma
MIT	Instituto de Tecnologia de Massachusetts
MLM	Minimal Learning Machine
MLP	Perceptron Multicamada
MQ	Mínimos Quadrados
NBR	Norma Brasileira Regulamentar
RBF	Função de Base Radial
RKS	Random Kitchen Sinks
ROC	Curva de Característica de Operação do Receptor
PSD	Densidade Espectral de Potência
RVFL	Random Vector Functional Link
SB	Sensibilidade
SNC	Sistema Nervoso Central
SNP	Sistema Nervoso Periférico
SVM	Máquina de Vetor de Suporte
TT	Tempo de Treinamento
TP	Tempo de Processamento
VAR	Variância
VPN	Valor Preditivo Negativo
VPP	Valor Preditivo Positivo

LISTA DE SÍMBOLOS

ρ	Autocorrelação
$\approx$	Aproximado
©	Copyright
f'	Derivada de f
σ	Desvio Padrão
∇	Gradiente
∞	Infinito
$\int$	Integral
®	Marca Registrada
e	Número de Euler
$\prod$	Produtório
$\in$	Pertence ao Conjunto
%	Porcentagem
$\mathbb{R}$	Reais
Ω	Resistência
§	Secção
$\sum$	Somatório
σ^2	Variância

SUMÁRIO

1 INTRODUÇÃO

Há um interesse cada vez maior em algoritmos randomizadas de aprendizagem de máquinas para tarefas de reconhecimento de padrões complexas. Alguns exemplos desses algoritmos são random vector functional link (RVFL) (PAO *et al.*, 1994; ZHANG; SUGANTHAN, 2016a), extreme learning machine (ELM) (HUANG *et al.*, 2015), minimal learning machine (MLM) (SOUZA JUNIOR *et al.*, 2015), no-prop network (WIDROW *et al.*, 2013), random forests (HO, 1998) e o random kitchen sinks (RKS) (RAHIMI; RECHT, 2009).

Todo esse interesse parece ser principalmente motivado pela maneira mais rápida que eles são projetados e executados, sem recorrer a um longo processo de aprendizagem em várias épocas de treinamento, conforme exigido por algoritmos de aprendizagem padrão, como o algoritmo de backpropagation (ZHANG; SUGANTHAN, 2016b).

Essa característica desses classificadores são alcançadas (por exemplo, para os classificadores neurais), simplesmente, ao aleatorizar os pesos e limiares da camada de entrada para a camada oculta. Apenas os pesos da camada oculta para a de saída são estimados a partir dos dados, que podem ser realizados por meio de qualquer técnica padrão de estimação de parâmetros para sistemas lineares (por exemplo, mínimos quadrados (MQ ou LS, do inglês Least Square)).

Uma consequência direta de todo o modismo em torno de classificadores randomizados é que os artigos quase sempre relatam resultados positivos, nos quais os desempenhos dos classificadores randomizados são equivalentes ou melhores do que aqueles alcançados por poderosos classificadores não-lineares, como o MLP e o SVM. Os estudos críticos são muito difícies de encontrar. Assim, um leitor mais experiente pode identificar um viés de confirmação nos resultados relatados, onde algoritmos randomizadas são construídos para seus melhores desempenhos, enquanto os classificadores mais tradicionais não são projetados com tanto empenho.

A partir do exposto, nesta dissertação, procuramos preencher uma lacuna na literatura de classificadores de padrões randomizados. Para isso, selecionamos uma tarefa desafiadora: detecção de crises epiléticas a partir de sinais de eletroencefalograma (EEG) (ADELI; GHOSH-DASTIDAR, 2010).

Epilepsia é um transtorno neurológico que afeta milhares de pessoas mundialmente de todas as raças, sexos, condições socioeconômicas e regiões, podendo gerar consequências profundas, incluindo morte súbita, ferimentos, problemas psicológicos e transtornos mentais. Consequentemente, é um problema de saúde pública, por ser uma das condições neurológicas

crônicas graves mais comum no mundo (BESSA, 2016).

A convulsão epiléptica é um estado em que há uma descarga anormal, excessiva e síncrona de neurônios localizados basicamente no córtex cerebral. Esta atividade anormal é intermitente e geralmente autolimitante, que dura de alguns segundos a alguns minutos e afeta milhões de pessoas em todo o mundo (cerca de 1% da população mundial) (LEHNERTZ *et al.*, 2003; ORGANIZATION, 2017). Essa atividade pode ocorrer em um conjunto de neurônios do encéfalo (crises focais) ou em áreas mais extensas (crises generalizadas), de modo que os sintomas de cada crise dependerá das partes do cérebro envolvida na disfunção (KANASHIRO, 2006).

A detecção de ataques epiléticos e o diagnóstico correspondente são realizados por neurologistas ainda com base no exame visual do EEG. Este foi introduzido no início do século passado por Hans Berger e desde então o exame vem sendo utilizado para o diagnóstico de diversas patologias associadas a transtornos mentais. O fato que a epilepsia é um transtorno neurológico está associado com alterações neuronais, que resultam em potenciais eletromagnéticos detectáveis (descargas epileptiformes) que podem ser mensurados através de eletrodos localizados no escalpo, esses sinais são de baixíssima amplitude, sendo assim, necessário um circuito amplificador para que o sinal seja perceptível.

O sinal de EEG é muito ruidoso, não linear e não estacionário (SUBHA *et al.*, 2010). Tais condições caracterizam o processamento do sinal de EEG (por exemplo, para detecção de crises epiléticas) como uma tarefa desafiadora, mesmo com todos os desenvolvimentos em extração de atributos e métodos de aprendizagem de máquinas. A literatura sobre a aplicação de classificadores não-lineares na detecção / classificação de convulsões, tais como as redes MLP, RBF e máquinas de kernel (SVM e LSSVM), é extensa (ALOTAIBY *et al.*, 2014). Como esperado, há um interesse crescente em aplicar classificadores randomizados para as mesmas tarefas (WANG *et al.*, 2017; DING *et al.*, 2015; ZHAO *et al.*, 2015; DONOS *et al.*, 2015).

1.1 Objetivos

1.1.1 Objetivo geral

Realizar uma ampla comparação de desempenho de classificadores randomizados e não randomizados na tarefa de classificação de crises epilépticas a partir de sinais de EEG.

Além do objetivo principal, com este trabalho, visamos obter maior aprofundamento

matemático nos métodos aplicados e gerar resultados que possam colaborar com outras pesquisas nesta temática que ainda hoje desafia diversos pesquisadores e cientistas.

1.1.2 Objetivos específicos

Como objetivos específicos, esperamos:

- Estudar a teoria relacionada a obtenção de sinais de EEG;
- Analisar e escolher um dentre os vários conjunto de dados de sinais de EEG de uso público para utilização neste trabalho;
- Implementar os seguintes classificadores convencionais: MLP, MQ e SVM;
- Implementar os seguintes classificadores randomizados: RVFL, ELM e RKS;
- Utilizar 2 métodos de extração de atributos para os sinais de EEG;
- Comparar e avaliar a performance dos classificadores.

1.2 Lista de publicações

Ao longo do desenvolvimento deste trabalho foram confeccionados, submetidos e aprovados os seguintes artigos:

- SILVA, N. R.; JUNIOR, J. P. S.; BARRETO, G. A.; SOUZA, R. B. Randomized Pattern Classifiers for Epileptic Seizure Detection: A Critical Assessment. XIII Congresso Brasileiro de Inteligência Computacional - (CBIC), 2017.
- JUNIOR, J. P. S.; SILVA, N. R.; BARRETO, G. A.; SOUZA, R. B. Avaliação de Desempenho da Rede Neural Máquina de Aprendizado Extremo na Detecção de Crises Epilépticas. 7o Simpósio de Instrumentação e Imagens Médicas (SIIM) / 6o Simpósio de Processamento de Sinais da UNICAMP (SPS-UNICAMP), 2017.

1.3 Organização do Restante da Dissertação

O trabalho inicia-se com a descrição dos objetivos e motivação para o desenvolvimento do mesmo, assim como sua aplicação e possíveis resultados.

Ao longo do capítulo dois é realizada uma revisão bibliográfica apresentado os conceitos de eletroencefalografia e epilepsia, entre outros temas de bioengenharia, tais como características das atividades neurais do cérebro, processo de aquisição dos sinais, padrão "10-20"da distribuição da localização dos eletrodos, bandas ou ritmos cerebrais entre outros

assuntos.

No capítulo três, é apresentado os dois métodos de extração de atributos ou características (features) que serão utilizados. Será iniciado com a explicação sobre o método que utiliza os coeficientes da predição linear do sinal em um modelo autorregressivo, seguido pela abordagem utilizando densidade espectral de potência (PSD) através do periodograma, destacando o equacionamento e formalismo matemático necessário e a implementação em código para analisar e executar os dois métodos.

No capítulo quatro, é descrito os seis classificadores utilizados reunidos em cinco grupos distintos: linear contendo o classificador mínimos quadrados (MQ); randomizado baseado em redes neurais com os classificadores Random Vector Functional-Link (RVFL) e extreme learning machine (ELM), randomizado baseado em método de kernel com Random Kitchen Sinks (RKS), Não randomizados baseado em redes neurais com MultiLayer Perceptron (MLP) e não randomizados baseado em método de kernel com Suport Vector Machine (SVM). Para fins organizacionais, dividimos em duas partes, sendo a primeira com a descrição dos classificadores randomizados no capítulo quatro e a segunda parte com a descrição dos classificadores não randomizados no Apêncice A, B e C.

No quinto capítulo é apresentado a as técnicas, parâmetros e configurações utilizadas nos dois métodos de extração de atributos e nos classificadores propostos, apresentando as ferramentas utilizadas.

No sexto capítulo são apresentados os resultados obtidos em cada cenário proposto contendo a descrição detalhada dos mesmos. Nesse capítulo teremos a análise quantitativa e qualitativa dos resultados.

Por fim, apresentaremos a conclusão do trabalho, discutindo sobre os objetivos alcançados, comentários dos resultados e trabalhos futuros a respeito do tema abordado.

2 SINAIS DE ELETROENCEFALOGRAMA

Este capítulo irá apresentar as principais características do sinal de eletroencefalograma a partir da eletroencefalografia, destacando as atividades e os ritmos cerebrais, assim como os métodos de gravação do EEG, padrão de posicionamento dos eletrodos (Sistema 10-20), as características eletrográficas de uma anormalidade detectada no EEG destacando a epilepsia, além de técnicas, características e conceitos de instrumentação e engenharia biomédica úteis para a compreensão do trabalho.

2.1 Fundamentação

Para o completo diagnóstico de distúrbios cerebrais como a epilepsia, é de fundamental importância a compreensão dos princípios básicos de funcionamento, estruturação, desenvolvimento, funções neuronais e a atividades neurofisiológicas do cérebro somado com os principais mecanismos para a gravação e a interpretação da atividade elétrica do cérebro.

Os principais métodos utilizados atualmente para realizar a gravação e verificar as alterações funcionais e fisiológicas do cérebro humano são: EEG, magnetoencefalograma (MEG) e ressonância magnética funcional (FMRI, do inglês Functional Magnetic Ressonance Imaging) (ARAUJO *et al.*, 2004). Cada um desse exames apresentam suas vantagens e desvantagens de acordo com cada tipo de distúrbio, porém de fato a maior proporção dos exames realizados é utilizando o EEG por conta da menor complexidade em relação aos outros, baixo custo ao usuário e maior facilidade de acesso nos hospitais. Assim, o EEG é uma excelente ferramenta para a exploração da atividade neuronal do cérebro associada a mudanças síncronas dos potenciais elétricos da membrana dos neurônios vizinhos.

Hans Berger (1873-1941) iniciou o estudo sobre sinais EEG em humanos em 1920, usando um galvanômetro com uma sensibilidade de 130 $\mu V/cm$ fez a primeira gravação de EEG com cerca de três minutos de duração. Na década de 50 os trabalhos com EEG expandiu em todo o mundo, sendo impulsionada com a popularização de cirurgia para remover focos epiléticos. Nessa década também os eletrodos utilizados no EEG evoluíram utilizando materiais como tungstênio e com eletrólitos como cloreto de potássio e com diâmetros de cerca de $3\mu m$ (SANEI; CHAMBERS, 2007).

Dados os crescentes avanços tecnológicos em bioengenharia, foi possível grandes aprimoramentos no registro do eletroencefalograma, tais como os sistemas digitais, os registros

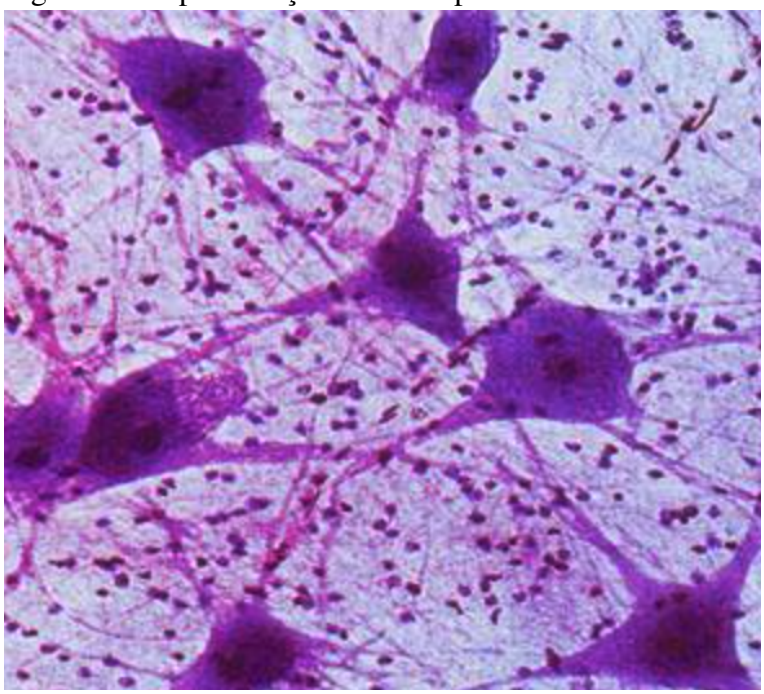
sincronizados com vídeo, a utilização dos sistemas de multicanais, chegando atualmente aos equipamentos com monitoramento remoto e portáteis. A avaliação rotineira do EEG de superfície é o cenário mais comum na prática da epileptologia clínica, sendo o mais utilizado para o diagnóstico e condução do tratamento da maior parte das síndromes epiléticas. O EEG é o meio de diagnóstico mais frequentemente utilizado para estudo da Epilepsia, sendo também o menos dispendioso e abundante tanto de dados como de pesquisa no ambiente acadêmico.

2.1.1 Fundamentos fisiológicos

A atividade cerebral humana inicia-se entre a decima sétima e vigésima terceira semana de formação. O sistema nervoso é composto por uma rede de células especializadas na condução de impulsos elétricos denominado neurônios, figura 1, que juntas comunicam-se e processam informações do corpo e do ambiente externo, tomadas de decisões entre outras funções.

Na maioria dos seres vivos, o sistema nervoso é dividido em sistemas nervoso central (SNC) e sistema nervoso periférico (SNP).

Figura 1 – Representação microscópica de neurônios.



Fonte: Disponível em (INFO ESCOLA, 2017).

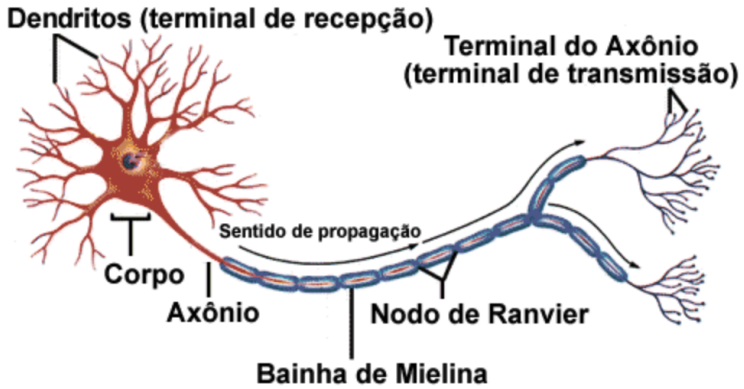
Os neurônios, figura 2, são divididos em três partes: o corpo celular que é a parte onde ficam o núcleo e diversas organelas, como mitocôndrias, que irão produzir algumas substâncias

importantes e energia para o funcionamento correto da célula; Os dendritos que são várias pequenas ramificações que saem do corpo celular, propagam sinais elétricos, retransmitindo-os através do axônio que é uma grande extensão do corpo celular, que se conecta outros neurônios ou células de outros tecidos, como músculos e glândulas. Em torno do axônio geralmente são formadas as bainhas de mielina, compostas de células especializadas chamadas de células de Schwann, que são envoltórios contendo material lipídico. Essa bainha faz com que o transporte de impulsos elétricos seja mais rápido. Alguns axônios podem ultrapassar 1 metro de comprimento.

Além disso, os neurônios podem ser classificados em receptores quando são encarregados de captar informações diretamente das células sensoriais, como aquelas que compõem a retina (olho), o ouvido, tato, a língua, etc. Essa captação é feita utilizando os dendritos; neurônios de conexão ou mistos quando fazem a conexão entre dois neurônios, recebe informação pelo dendrito, e a repassa à célula nervosa seguinte usando o axônio. Esse tipo é o mais encontrado nos sistemas nervosos animais. E por fim, como neurônios efetores que são os neurônios que recebem as informações do cérebro (as respostas aos estímulos captados pelos neurônios receptores) e as repassam para os músculos e glândulas.

A principal função do neurônio é transmitir potenciais elétricos para outras células ao longo das finas fibras denominadas axônios que utilizam substâncias químicas chamadas neurotransmissores para permitir a função neuronal chamada sinapses. Estes potenciais elétricos são chamados de potenciais de ação ou impulso nervoso e pode ser interpretada como a informação transmitida por um nervo a uma célula.

Figura 2 – Representação pictórica da estrutura de um neurônio.

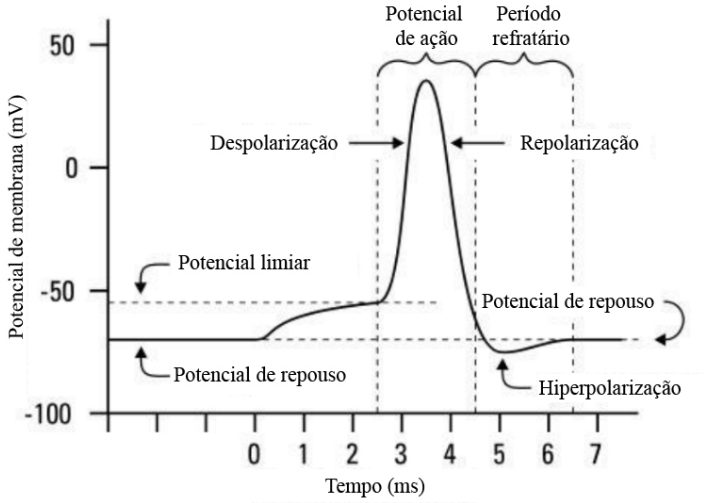


Fonte: Disponível em (INFO ESCOLA, 2017).

Os potenciais de ação são causados por uma troca de íons através da membrana do neurônio, ou seja, é uma mudança temporária no potencial elétrico da membrana que é transmitida ao longo do axônio. Geralmente é iniciado no corpo celular e propaga somente

em uma direção (dendritos, corpo celular e por fim axônio). Ao despolarizar, o potencial da membrana do neurônio, torna-se mais positivo, produzindo um pico de potencial também denominado disparo ou ativação. Após chegar ao ponto máximo do pico ocorrerá a repolarização da membrana, se tornando mais negativa. O potencial elétrico se torna mais negativo do que o potencial elétrico referente ao de repouso e em seguida retorna ao normal. Esse ciclo dura em torno de 5 a 10 ms, na figura 3 é apresentado um exemplo de um disparo.

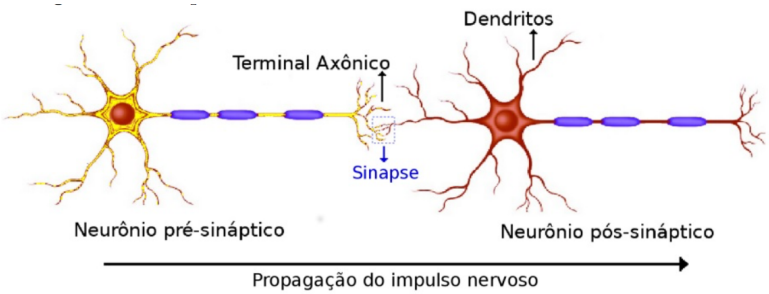
Figura 3 – Representação esquemática do potencial de ação.



Fonte: (SHARMA *et al.*, 2012).

Os valores máximos do potencial elétrico variam devido ao processo na sinapse. A sinapse, apresentada na figura 4 é a transmissão de um sinal elétrico ou químico entre dois neurônios: o neurônio que inicia a transmissão é denominado pré-sináptico e o que sofre a ação é denominado o pós-sináptico.

Figura 4 – Sinapse elétrica com a propagação do impulso nervoso.



Fonte: (BORGES *et al.*, 2015).

Através da sinapse é possível efetuar a medição do EEG, esse sinal e o resultado da medição das correntes de excitação sinápticas que fluem dos dendritos de muitos neurônios piramidais do córtex cerebral. Quando os neurônios estão ativados, as correntes sinápticas geram

um campo magnético mensurável por uma eletromiografia[1] e um campo elétrico secundário sobre o couro cabeludo que pode ser medido através de um EEG.

Assim, a atividade do EEG representa o somatório da atividade síncrona de um conjunto de milhões de neurônios que tem uma orientação espacial semelhantes. Portando a atividade do EEG apresenta as oscilações em uma variedades de frequências de uma rede de neurônios. Quando as ondas de íons no escalpo atinge o eletrodo, a diferença de potencial elétrico entre o eletrodo que se deseja medir e o eletrodo de referência pode ser mensurada utilizando um circuito amplificador com um voltímetro. É necessário compreender que a cabeça humana é composta de várias camadas, dentre elas podemos citar o crânio, couro cabeludo, cérebro e outras membranas (meninge, por exemplo). Cada camada apresenta uma resistência diferente e a condutividade no crânio pode chegar até cem vezes mais que nos outros tecidos envolvidos, então necessita-se de uma grande concentração de neurônios em uma determinada área para gerar sinais mensuráveis por um eletrodo.

O córtex cerebral é responsável por gerar quase que toda a atividade do EEG e os potenciais pós-sináptico são responsáveis por quase a totalidade do registro da atividade elétrica e não somente os potenciais de ação. A atividade do EEG também é dependente de mecanismos do fluxo de corrente, condução de volume, propagação, sincronização e dessincronização.

Portanto, o estudo dos sinais elétricos do cérebro através do EEG é de fundamental importância para o estudo das anormalidades, tal estudo é utilizado principalmente para a investigação da área motora suplementar, campo ocular frontal, área motora primária, área somato-sensorial primária e área pré-motora, Representação esquemática das principais partes do cérebro, monitorar o estado de alerta como coma e morte encefálica, localização de áreas de danos após a lesão cabeça que foram causadas por acidente vascular cerebral e tumor, monitoramento envolvimento cognitivo (ritmo alfa, será introduzido na próxima seção) e a epilepsia (SANEI; CHAMBERS, 2007).

O método mais utilizado atualmente para confirmação de uma crise epilética é o diagnóstico por inspeção visual dos sinais de EEG por um especialista. Essa análise é um processo racional e sistemático, requerendo uma série de etapas ordenadas que caracterizam as atividades elétricas registradas em termos de descritores específicos ou características e medidas, portanto é necessário analisarmos a frequência, a amplitude do sinal, a forma de onda, o modo de

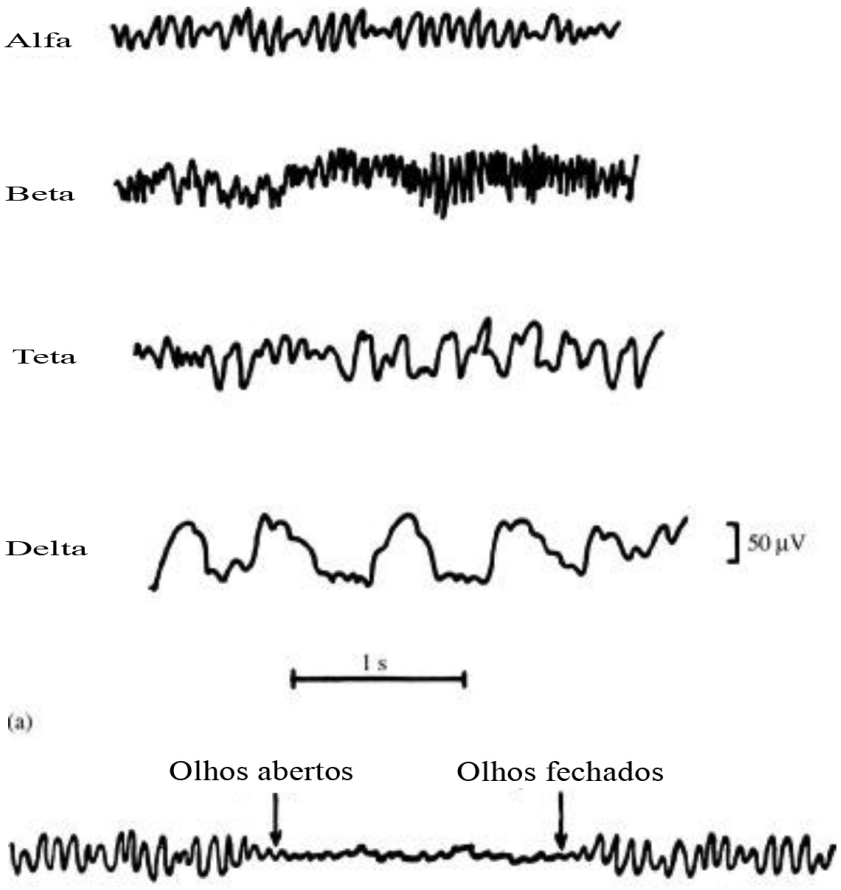
[1] Técnica de monitoramento da atividade elétrica das membranas excitáveis das células musculares, resultando no somatório algébrico de todos os sinais detectados sob a área de alcance dos eletrodos, podendo ser afetado por propriedades musculares, anatômicas e fisiológicas (BASMAJIAN; LUCA, 1985)

ocorrência (aleatório, em série, contínua), seus momentos estatísticos dentre outras características que poderemos utilizar como atributos no decorrer deste trabalho.

2.2 Ritmos cerebrais

Dentre os diversos padrões contidos em uma ampla faixa de componentes de frequência do sinal de EEG, destaca-se o grupo dos ritmos cerebrais com características visuais e estatísticas bem definidas. Muitos desses padrões são diagnosticados por uma inspeção visual do sinal do EEG por um médico especialista da área. É importante salientar que as amplitudes e frequências de um determinado estado (de vigília ou do sono) varia de um ser humano para outro. E as suas características das ondas cerebrais também mudam com a idade de cada indivíduo.

Figura 5 – Exemplos típicos de ritmos cerebrais.



Fonte: (WEBSTER, 2009).

As principais ondas cerebrais podem ser classificadas em cinco grupos distintos com faixas de frequências bem definidas. Destas, quatro delas podemos visualizar na figura 5. Além

disso, na figura é apresentada a mudança que ocorre quando uma pessoa está em um estado relaxado e abre os olhos.

As bandas de frequências dos ritmos cerebrais são chamadas de alfa (α), teta (θ), beta (β), delta (δ) e gama (γ) e serão apresentadas abaixo em ordem crescente de faixa de frequência:

- As ondas delta estão dentro de uma faixa de 0,5-4 Hz, sendo observadas em estágios de atividades cerebrais lentas. Estas ondas estão associadas principalmente com o sono profundo, é facilmente confundível com ruído causado pelos grandes músculos do pescoço ou da mandíbula. No entanto, através da aplicação de métodos de análise do EEG, é fácil de perceber quando a resposta é causada por movimento excessivo.
- As ondas teta, cuja faixa de frequência é de 4 a 7,5Hz, estão associadas com a mudança da consciência em direção à sonolência. Essas ondas também estão associadas com o acesso ao material inconsciente, inspiração criativa e a meditação profunda. Normalmente, esse tipo de faixa de frequência está acompanhado por outras frequências relacionando com o nível de excitação. A onda teta desempenha um papel importante na infância, entretanto contingentes maiores de atividade de tais ondas em adultos em vigília são anormais e são causados por vários problemas patológicos.
- As ondas alfa estando entre 8 e 13 Hz são encontradas normalmente na parte posterior da cabeça na região occipital do cérebro. Em geral, têm forma arrendondada ou forma de um sinal sinusoidal. Raramente podem-se se manifestar como ondas agudas. As ondas alfas indicam uma consciência relaxada, sem qualquer atenção ou concentração, é o ritmo mais proeminente em toda a atividade cerebral. O estado alfa é reduzido ou eliminado através da abertura dos olhos, por ouvir sons desconhecidos, por ansiedade, concentração mental ou aumento da atenção.
- As ondas beta apresentando frequência no intervalo 14 e 30 Hz são associadas com o pensamento ativo, atenção e foco no mundo exterior para resolver um problema concreto e são observadas em adultos normais e também podem estar ligadas ao estado de pânico. Encontrada em sua maioria na região frontal e central com amplitudes menores que os ritmos alfa.
- Já as ondas gama estando acima 30 Hz geralmente não são de interesse clínico e fisiológico. No entanto, a detecção destes ritmos pode ser utilizada para a confirmação de determinadas doenças cerebrais.

Mesmo profissionais com experiência e bastante treinados sentem dificuldades de entender e detectar os ritmos cerebrais do EEG em determinados momentos. Assim, há diversas ferramentas para o processamento de sinais que permitem separar e analisar formas de ondas desejadas dentro do EEG. Portanto, uma análise visual do EEG é subjetiva e dependente da anormalidade que se deseja verificar.

2.3 Instrumentação biomédica

A área de instrumentação biomédica visa o desenvolvimento de sistemas e equipamentos elétricos, eletrônicos e mecânicos destinados ao diagnóstico, tratamento ou monitoração de pacientes, sob supervisão médica.

Nesta seção, iremos abordar especificamente os conceitos, ferramentas e técnicas necessárias para a aquisição do sinal de EEG e atividade relacionadas à detecção de crise epilética.

2.3.1 Eletrodos

O eletrodo de superfície, figura 6, é um dispositivo colocado no escalpo do paciente através do qual a atividade elétrica cerebral é captada e transmitida. Fios blindados são utilizados para diminuir as interferências externas econectados aos amplificadores do aparelho de EEG. O mecanismo de condução da eletricidade do escalpo para o eletrodo consiste na condução da corrente pelos íons presentes na solução (gel ou pasta) condutora. A corrente elétrica é conduzida pelos íons na solução da mesma forma que a corrente é carregada pelos elétrons ligados em um condutor metálico (DUFFY *et al.*, 1999; MISULIS, 1989). As voltagens medidas não são constantes, pois variam conforme várias condições: sono, vigilia, estado nervoso entre outros fatores.

Figura 6 – Exemplos de eletrodos de superfície.

Fonte: Disponível em (KANDEL, 2016)

Quando entra em contato com a solução salina condutora, o metal do eletrodo descarrega íons, formando uma dupla camada elétrica na interface metal-eletrólito. Isso gera um potencial (potencial de meia célula), pois não é a voltagem presente no próprio eletrodo, mas sim o potencial medido em relação a outro eletrodo vizinho. Isso é importante porque, como a maioria dos eletrodos utilizados na rotina é polarizável, eles devem ser feitos do mesmo material para que o potencial gerado em cada um deles seja semelhante e, assim, excluídos do registro como sinal de modo comum. Isso evita que sejam produzidos artefatos no registro eletrencefalográfico (DUFFY *et al.*, 1999; MISULIS, 1989).

Os eletrodos devem ser reversíveis, isto é, devem permitir que o fluxo de cargas passe através da junção (interface) em ambas as direções. Eletrodos não reversíveis podem permitir a polarização entre a pasta e o eletrodo, reduzindo o fluxo de corrente na interface metal-eletrólito (MISULIS, 1989).

Dentre os principais tipos de eletrodos, podemos citar os descartáveis à base de gel, eletrodos de disco reutilizáveis, eletrodos de escalpo ou toucas (figura 7), eletrodos de base salina e eletrodos de agulha. As toucas de eletrodos são normalmente utilizadas para as gravações de multicanais utilizando um número grande de eletrodos, esses eletrodos consistem em discos de Ag-AgCl com menos de 3 mm de diâmetro, com longos fios flexíveis conectados aos amplificadores (SANEI; CHAMBERS, 2007).

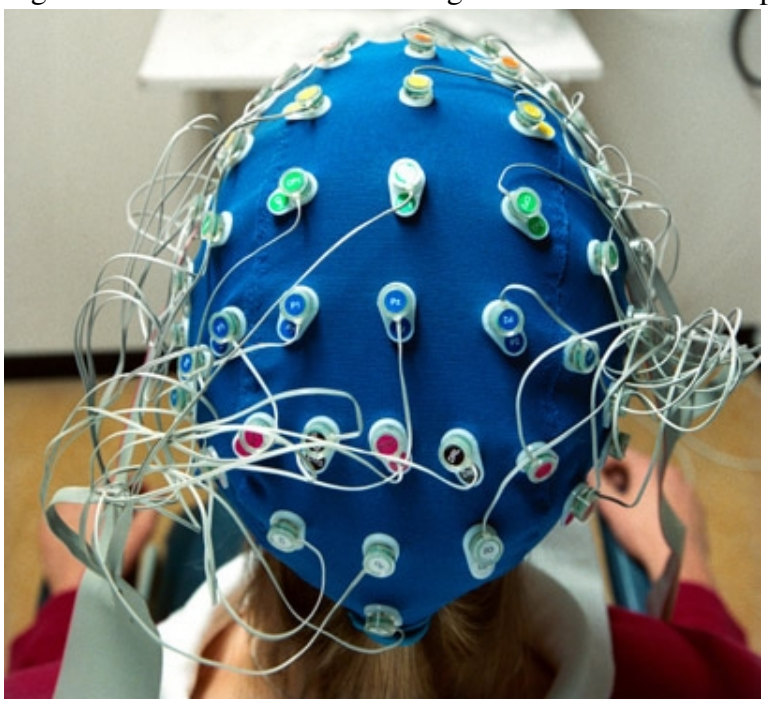
Outro importante conceito é o de canal do EEG, como será visto adiante, o banco de dados utilizado possui 23 canais. Assim, um canal do EEG, ou sinal, é formado pela diferença de potenciais medidas entre dois eletrodos. Por exemplo, considerando o canal 1 referente aos eletrodos P7 e O1, então este canal apresenta o sinal gerado pela diferença entre esses dois eletrodos.

2.3.2 Padrão convencional de posicionamento dos eletrodos

O posicionamento convencional dos eletrodos foi recomendando pela *International Federation of EEG Societies* e é denominado padrão ou sistema 10-20 (JASPER, 1958), de acordo com a figura 8.

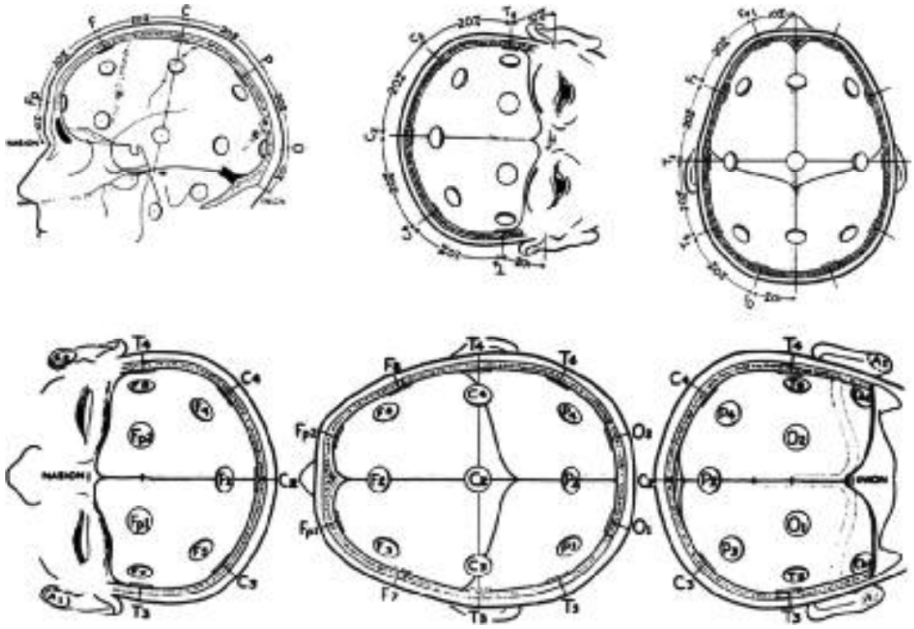
O padrão refere-se à distribuição dos eletrodos considerando algumas distâncias constantes usando marcos anatômicos específicos a partir dos quais as medidas seriam feitas e, em seguida, usa-se 10 ou 20% do que a distância especificada como o intervalo entre os elétrodos. Os números pares referem-se aos eletrodos posicionados no hemisfério direito e os

Figura 7 – Touca de eletroencefalografia com eletrodos acoplados.



números ímpares aos eletrodos do hemisfério esquerdo. As letras apresentam qual a localização do eletrodo na cabeça: frontal (F), temporal (T), central (C), parietal (P), occipital (O).

Figura 8 – Padrão de posicionamento de eletrodos do sistema 10-20.



Fonte: (KLEM *et al.*, 1999).

2.3.3 Técnicas de medição e gravação de EEG

O eletroencefalograma é um exame que permite o estudo do registro gráfico das correntes elétricas espontâneas desenvolvidas no cérebro, através de eletrodos aplicados no couro cabeludo na superfície encefálica ou até mesmo dentro da estrutura encefálica.

Apesar do EEG realizado com o uso de eletrodos dentro da estrutura encefálica ter um menor índice de ruido, pois por o eletrodo ser implantado no interior do cérebro, é menos susceptível a interferência eletromagnética e artefatos, além de captar uma melhor resolução espacial, entretanto caracteriza-se por ser de natureza invasiva. Assim, o EEG utilizando eletrodos de superfície com escalpo no couro cabeludo é o mais comum clinicamente e com ótimos resultados, considerando a aquisição com eletrodos de qualidade para uma maior exatidão na medição do sinal.

Atualmente, os sistemas de aquisição de EEG, de acordo com alguns modelos comuns exibidos na figura 9, consistem em uma série de delicados eletrodos, com um conjunto de amplificadores, um por canal, uma série de filtros de sinais e dispositivos para armazenamento e visualização. Sendo assim, o sinal do EEG foi transformado em um sinal digital, exigindo uma frequência de amostragem, resolução e codificação dos sinais.

Os sistemas informatizados de EEG transformam o sinal analógico para digital por meio de conversores analógico-digitais (AD), sabendo que para a maioria das aplicações do EEG a banda de frequência é limitada em Hz, a frequência mínima de amostragem é de 60 Hz para satisfazer o critério de Nyquist, considerando que a frequência analisada para detectar a crise epilética está dentro do intervalo dos primeiros 30Hz (SHOEB; GUTTAG, 2010). A resolução de cada amostra utilizada em sua maioria é de 16 bits, levando a arquivos de tamanhos grandes, sendo necessário alguma compactação.

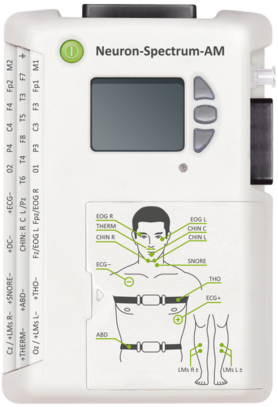
2.4 Materiais e Métodos

Para compreendermos a complexidade e estrutura do banco de dados utilizado neste trabalho, iremos descrever algumas características genéricas sobre o mesmo. Considerando que o sinal de um EEG digitalizado é um conversão de um sinal analógico para digital utilizando um conversor AD, normalmente a resolução utilizada nos sistemas de gravação do EEG é a de 16 bits. Podemos calcular a dimensão do arquivo, dado que um paciente utiliza 23 eletrodos a uma taxa de amostragem de 256 Hz amostras por segundo em um período de uma hora com a

Figura 9 – Equipamentos de Aquisição de EEG

(a) Eletroencefalógrafo Neuron-Spectrum-5

(b) Máquina digital leitora do sinal de EEG com 32 canais



(c) Polissonígrafo Neuron-Spectrum-AM

resolução de 16 bits, o tamanho do arquivo gerado será de $23 \times 60 \times 60 \times 500 \times 16 \approx 339 Mbits$ (SANEI; CHAMBERS, 2007).

Dado a ineficiência de armazenar os dados de todos os pacientes, concluímos a necessidade um processamento digital de sinal com o objetivo de extrair informações, características e comportamentos que melhor represente esse sinal e que possa ser manipulado e armazenado facilmente.

Os arquivos utilizados são do projeto CHB-MIT Scalp EEG Database (GOLDBERGER *et al.*, 2000) e podem ser acessados através do site: (SHOEB, 2016). Dentre outros trabalhos com esse banco de dados, pode-se destacar o (SHOEB, 2009; SHOEB; GUTTAG, 2010).

O conjunto de dados utilizado neste trabalho consiste em gravações continua do sinal de EEG, do tipo escalpo, realizado em 24 pacientes (a maioria pediátricos) após a retirada da medicação para a avaliação de cirurgia de epilepsia no Hospital Infantil de Boston, lista de paciente apresentada no Anexo A. O sinal de EEG apresenta uma amostragem de 256 Hz utilizando em 18 eletrodos. A montagem do escalpo para as gravações seguiu o padrão 10-20.

Os arquivos gerados pela gravação do EEG em média apresentam de 1 hora de duração, em alguns pacientes foram utilizadas gravações de 4 horas, gerando um total de 686 arquivos (aproximadamente 32GB de dados). Os arquivos estão catalogados em com convulsões e sem convulsões, em todos os arquivos foram detectados 197 convulsões distribuídas em 141 arquivos totalizando um total de 195,5 minutos para todos pacientes.

Os arquivos foram disponibilizados no formato *.edf* (European Data Format), seguindo o seguinte padrão: `chb01_03.edf`, onde chb01 identifica o paciente e 03 identifica o

número do arquivo que esta sendo utilizado. A lista dos arquivos utilizados é apresentada no Anexo A. O arquivo de extensão *.edf* é um arquivo de dados composto por um cabeçalho seguido pelos registros de dados, a especificação pode ser vista em (KEMP *et al.*, 1992). O cabeçalho identifica o paciente e especifica as características técnicas do sinal gravado. Os primeiros 256 bytes do cabeçalho especifica o número da versão deste formato, o paciente, a identificação de gravação, informações de tempo sobre a gravação, o número de registros de dados e, finalmente, o número de sinais em cada registro de dados. Em seguida é especificado o tipo de sinal (por exemplo, EEG, temperatura corporal, etc), a calibração de amplitude e o número de amostras em cada registo de dados (KEMP *et al.*, 1992).

Para visualizar os dados presentes nos arquivos utilizando o Matlab foi necessário utilizar a função `edfread.m`. O parâmetro de entrada da função é o endereço e nome do arquivo, e a função retorna um cabeçalho (header) e os dados (recorddata). Um exemplo de como a função é utilizada é apresentado a baixo no código a seguir:

```
clear; clc;
% Exemplo de utilização da função edfread
[header, recorddata] = edfread('chb01\_03.edf');
```

Utilizando essa função já disponível no Matlab, para um arquivo salvo no diretório de nome `chb01_03.edf`, a função retorna os dados gravados em forma de uma matriz de dimensões 23 linhas por 921600 colunas. O número de linhas informa a configuração dos 23 canais utilizados, os sinais dos 18 eletrodos são combinados em pares diferenciais. Na tabela 1 é apresentada a configuração dos canais utilizada para a gravação do arquivo apresentado no exemplo. As colunas apresentam os dados gravados para cada par diferencial de eletrodos, cada coluna é equivalente a uma amostragem do sinal para os 23 canais. Sabendo que a frequência de amostragem utilizada para a gravação é de 256 Hz, temos uma amosta a cada 1/256 segundos. Então 921600 colunas equivale a uma hora, ou 3600 segundos, de gravação. Cada arquivo de uma hora apresenta em média 40.4MB de tamanho em arquivo.

Após a obtenção dos dados citados acima, é realizada a preparação do vetor de atributos para cada um dos dois métodos de extração de características e realizado o treinamento de todos modelos de classificação, seguido pela execução dos algoritmos.

As funções matemáticas, classificadores e técnicas descritas foram implementas no software Matlab utilizando algoritmos descritos nesse trabalho ou referenciados por pesquisas já existentes no meio acadêmico, científico e tecnológico.

Tabela 1 – Configuração dos canais utilizados

Canal 1	FP1 - F7
Canal 2	F7 - T7
Canal 3	T7 - P7
Canal 4	P7 - O1
Canal 5	FP1 - F3
Canal 6	F3 - C3
Canal 7	C3 - P3
Canal 8	P3 - O1
Canal 9	FP2 - F4
Canal 10	F4 - C4
Canal 11	C4 - P4
Canal 12	P4 - O2
Canal 13	FP2 - F8
Canal 14	F8 - T8
Canal 15	T8 - P8
Canal 16	P8 - O2
Canal 17	FZ - CZ
Canal 18	CZ - PZ
Canal 19	P7 - T7
Canal 20	T7 - FT9
Canal 21	FT9 - FT10
Canal 22	FT10 - T8
Canal 23	T8 - P8

2.5 Detecção de ataque epilético no sinal de EEG

O ataque epilético é um distúrbio neurológico cuja expressão comum entre os indivíduos é a recorrência de crises convulsivas não induzidas.

Além da variabilidade de cada indivíduo, alguns fatores podem determinar a variação dos sintomas clínicos, sendo os principais relacionados com a localização da origem desse ataque, com o padrão de distribuição e com a abrangência para outras regiões do cérebro.

A redistribuição da energia espectral, causada pela epilepsia, consiste no surgimento ou no desaparecimento de componentes de frequência dentro de uma faixa que varia de 0 a 25 Hz (SHOEB, 2009). Porém, essas componentes de frequência variam de paciente para paciente e varia também com o local de origem do ataque.

Utilizando os exemplos disponibilizados por (SHOEB, 2009), a figura 10 apresenta um exemplo do comportamento dos sinais de EEG de dois paciente em crise epiléticas.

Analisando a figura 10, com o auxílio de (SHOEB, 2009), no primeiro gráfico a crise

se inicia em 1723 segundos, diminuindo a energia do sinal de EEG em todos os canais, seguido pelo surgimento de um ritmo beta nos canais F3 - C3 e C3 - P3. Em seguida a amplitude deste ritmo aumenta à medida que a sua frequência diminui e se instala dentro da banda de frequência do tipo teta. No exemplo seguinte, a crise epilética inicia-se em 6313 segundos com o surgimento de ritmo teta proeminentemente nos canais F7-T7 e T7-P7, os outros canais apresentam mudanças após o início de crise, além de ser possível verificar alterações da frequência em outro canais.

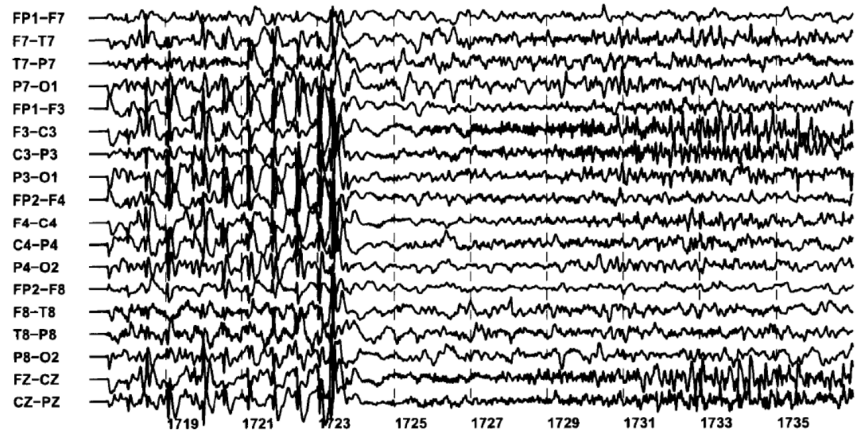
Para cada canal, é necessário conhecer a amplitude, pois no início de um ataque focal, acontece a alteração na atividade em poucos canais do EEG, pela localização do eletrodo mais perto na região de origem da epilepsia, porém é necessário verificar o início de um atividade epilética generalizada que envolve todos os canais utilizados.

Os exemplos acima representam o quão desafiador é a classificação de crise epilética, dado a imensa variabilidade de comportamento e complexidade em classificar esses padrões de forma genérica para pacientes, sendo mais comum realizar um estudo do padrão do comportamento do EEG para cada paciente.

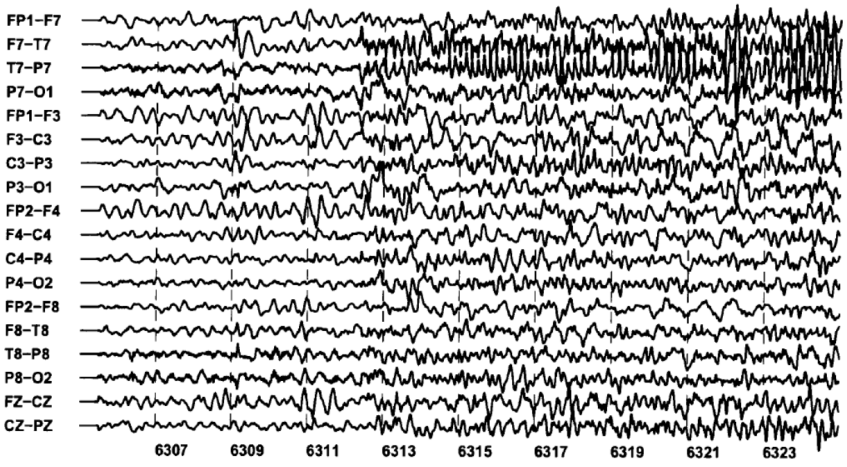
2.6 Resumo do capítulo

Neste capítulo, foram apresentados os principais conceitos que envolvem o desenvolvimento, funcionamento e característica do cérebro e sistema nervoso. Seguida pela apresentação dos conceitos básicos referentes a aquisição do sinal EEG de escalpo e seus fundamentos fisiológicos no qual foi estudado as ativações de um neurônio e a sinapse entre neurônios. Continuamos com o explanação referente ao sinal e suas características, assim como sua aplicabilidade no diagnóstico da epilepsia, apresentando também os ritmos cerebrais, destacando as principais faixas de frequências e suas características e padrão convencional de posição dos eletrodos (10-20). Também foram destacadas a origem e o formato dos dados a serem utilizados nesse trabalho, proveniente de um banco de dados. Para finalizar, mostramos as diferenças entre 2 exemplos de crises epiléticas.

Figura 10 – Amostras de EEG com crise epilética para 2 pacientes.



(a) Paciente A



(b) Paciente B

Fonte: (SHOEB, 2009).

3 EXTRAÇÃO DE ATRIBUTOS DO SINAL DE EEG

O sinal de EEG é uma série temporal complexa, com comportamento aperiódico e altamente não estacionário. Assim, é necessário reparametrizar a série temporal para extrair atributos que preservem informações relevantes contidas nos sinais originais.

Dois dos métodos comumente utilizados para este propósito são os coeficientes de codificação preditiva linear (LPC, do inglês Linear Predictive Coding) (THERRIEN, 1992) ou periodograma de Welch (WELCH, 1967) utilizando a densidade espectral de Potência (PSD, do inglês Power Spectral Density). Deve-se notar que, no entanto, ambos os métodos assumem a estacionariedade das séries temporais. Para lidar com a não-estacionariedade elevada de um sinal EEG, a sequência original é segmentada em subsequências menores, que são então assumidas estacionárias.

Alguns outros métodos também se destacam nesse contexto, sem existir um método padronizado para a extração de atributos nos sinais de EEG. Dentre os quais podemos citar o da transformada discreta de wavelet (DWT, do inglês Discrete Wavelet Transform) (JAHBAKHANI *et al.*, 2006); (SUBASI, 2007), método de modelagem de processo autorregressivo (PENNY *et al.*, 2000; PFURTSCHELLER *et al.*, 1998) e por fim o método da estimação da PSD (CHIAPPA; BENGIO, 2004).

Uma parte importante e fundamental para a classificação, é a extração de atributos. Quando o sinal a ser extraído é o EEG, essa importância é elevada por se tratar de uma série de temporal complexa, com comportamento aperiódico, não estacionário e com dinâmica não linear.

Assim, escolher quais as características mais adequadas para o problema é fundamental para o desempenho desejado de um classificador e além do desejo de ser rápido o suficiente para poder ser utilizado em um plataforma de tempo real.

Neste trabalho, utilizaremos os métodos de coeficientes LPC e PSD com o método de Welch que são métodos lineares de tratamento de sinais tanto no domínio do tempo com coeficientes LPC, quanto no domínio da frequência com PSD.

3.1 Predição Linear

LPC é uma ferramenta oriunda da área de filtragem adaptativa com aplicação principalmente em processamento de fala. A predição linear recebe esse nome por considerar que cada

amostra do sinal pode ser predita a partir de uma combinação linear de amostras passadas; ou seja, assume-se uma correlação significativa entre as observações anteriores. Os pesos dados às amostras passadas nesta combinação são denominados coeficientes de predição linear e definem o chamado filtro de predição linear, cuja ordem é determinada pelo número de amostras passadas utilizadas.

Nesta dissertação, para realizar a predição, utilizaremos o modelo autoregressivo, a ser fundamentado a seguir. Os coeficientes encontrados irão compor o vetor de atributos do classificador.

3.1.1 Estimação de Parâmetros de um processo Autorregressivo

Um processo y_t é chamado de processo estocástico de ordem p, AR(p), se em cada intervalo de tempo t o valor de y_t é determinada pela seguinte expressão:

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \varepsilon_t, \tag{3.1}$$

onde $\phi_0, \phi_1, ..., \phi_p$ são os parâmetros do processo e ε_t é um processo estocástico denominado ruído aditivo gaussiano branco. Este processo é estacionário no sentido amplo cuja função de autocovariância é nula para todo $\tau \neq 0$, ou seja, é um conjunto de dados não relacionados com a variança σ_ε^2, $\varepsilon_n \sim N(0, \sigma_\varepsilon^2)$. Onde τ é uma variável de deslocamento ou atraso para comparar instantes ou amostras diferentes da mesma variável.

Os processos AR podem ser usados como modelos se for razoável assumir que o valor atual de uma série temporal depende da combinação do valor do seu passado mais um erro aleatório (EHLERS, 2009). Em outras palavras, é a combinação linear dos valores anteriores da série com a adição de um ruído branco.

A função de autocorrelação de um modelo AR(p) pode ser escrita de acordo com a seguinte expressão, denominada Equação de Yule-Walker:

$$\rho(\tau) = \phi_1 \rho(\tau - 1) + \phi_2 \rho(\tau - 2) + ... + \phi_p \rho(\tau - p), \tau > 0 \tag{3.2}$$

onde $\rho(\tau)$ é a função de autocorrelação normalizada que é dada por:

$$\rho(\tau) = \frac{R_x(\tau)}{\sigma_x^2} = \frac{E[x(t)x(t-\tau)]}{E[x^2(t)]} \tag{3.3}$$

onde o operador $E[\ ]$ representa o valor esperado de uma variável.

Com isso, objetivamos estimar os coeficientes $\phi_0, \phi_1, ..., \phi_p$, assumindo que o sinal do EEG em um intervalo de t segundos seja estacionário e ergódico. Utilizando o método dos momentos é possível estimar os coeficiente utilizando a equação de Yule-Walker.

O procedimento é iniciado calculando-se a estimativa amostral da função de autocorrelação normalizada $r(\tau)$, que é dada pela seguinte expressão:

$$r(\tau) = \frac{\sum_{k=\tau+1}^{N} x(k)x(k-\tau)}{\sum_{k=1}^{N} x^2(k)} \tag{3.4}$$

onde $x(k)$ é a k-ésima amostra do conjunto de dados.

Reescrevendo a Eq. 3.2 em função de $r(\tau)$ temos

$$r(\tau) = \phi_1 r(\tau-1) + \phi_2 r(\tau-2) + ... + \phi_p r(\tau-p), \tau > 0 \tag{3.5}$$

Dado que a função de autocorrelação é par, uma vez que $r(-\tau) = r(\tau)$ e que $r(0) = 1$, substituindo os valores de τ para $\tau = 1, 2, ..., p$ chegamos ao seguinte sistema de equações:

$$\begin{cases} r(1) &= \phi_1 &+ \phi_2 r(1) &+ \cdots + \phi_p r(p-1), & \tau = 1 \\ r(2) &= \phi_1 r(1) &+ \phi_2 r(2) &+ \cdots + \phi_p r(p-2), & \tau = 2 \\ \vdots & \vdots & \vdots & & \vdots \\ r(p) &= \phi_1 r(p-1) &+ \phi_2 r(p-2) &+ \cdots + \phi_p, & \tau = p \end{cases} \tag{3.6}$$

Escrevendo o sistema de forma matricial tem-se $R\phi = r$.

$$\begin{bmatrix} 1 & r(1) & \cdots & r(p-1) \\ r(1) & 1 & \cdots & r(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & \cdots & 1 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(p) \end{bmatrix} \tag{3.7}$$

onde $\mathbf{R}$ é uma matriz quadrada de dimensão p, e ϕ e $\mathbf{r}$ são vetores de dimensão $p \times 1$. Para calcular os valores estimados dos coeficientes teremos que inverter a matriz R. O vetor de coeficiente é dado por

$$\hat{\phi} = R^{-1} r \tag{3.8}$$

Com um conjunto de dados e utilizando a equação de Yule-Walker, podemos estimar os parâmetros de um processo AR(p). Porém, é importante também estimar qual a ordem do modelo que melhor se encaixa ao processo real a ser modelado.

Com o intuito de estimar a melhor ordem utilizando o método do momentos e a expressão de Yule-Walker é calculada da função de autocorrelação parcial (FACP), para efetuar o cálculo utiliza-se a Eq. 3.8 de forma recursiva.

3.2 Densidade Espectral de Potência

O funcionamento deste método dá-se a partir da aplicação do método da transformada rápida de fourier (FFT, do inglês Fast Fourier Transform) discreta ao sinal para encontrar seu conteúdo no domínio da frequência.

Sabemos que o sinal do EEG é não estacionário, o que significa que seu espectro e propriedades estatísticas mudam com o tempo. Tal sinal pode, entretanto, ser aproximado como sendo estacionário por partes, por meio de uma sequência de segmentos estacionários independentes. Nesta dissertação, será assumido que a duração de um intervalo para o qual pode-se considerar o segmento estacionário é de 2 segundos (SHOEB, 2009).

A PSD é calculada através da FFT, no qual é estimada uma sequência de autocorrelação e pode ser encontrada pela utilização de métodos não paramétricos. O método não paramétrico a ser utilizado nesse trabalho será o método de Welch (WELCH, 1967). Porém antes de discutirmos este método, neste capítulo apresentaremos o conceito de periodograma e as motivações para uso do método de Welch.

Dado que a autocorrelação $r_x(k)$ de um processo estacionário no sentido amplo quantifica a dependência temporal entre amostras sucessivas de uma realização deste processo, $P_x(e^{j\omega})$ pode ser calculada através da transformada de Fourier de tempo discreto:

$$P_x(e^{j\omega}) = \sum_{k=-\infty}^{\infty} r_x(k)e^{-jk\omega} \tag{3.9}$$

onde P_x é chamada de densidade espectral de potência. Dado o espectro de potência para calcularmos a sequência de autocorrelação, aplica-se a inversa da transformada de Fourier, como mostrado a seguir.

$$r_x(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_x(e^{j\omega})e^{-jk\omega}d\omega \tag{3.10}$$

$$r_x(k) = \lim_{N\to\infty} \frac{1}{2N+1} \left(\sum_{n=-N}^{N} x(n+k)x^*(n) \right) \tag{3.11}$$

onde $x^*(n)$ representa o conjugado de $x(n)$

Adiante iremos abordar o uso de métodos não paramétricos, que se baseiam na ideia de estimar a sequencia de autocorrelação de um processo estocástico através de um conjunto de dados medidos e em seguida utilizar a transformada de Fourier para obter uma estimativa do espectro de potência. Assim, será apresentado o periodograma e a sua variação o com o método de Welch.

Quando estamos trabalhando com um processo estacionário fraco para se calcular a estimativa da autocorrelação $(\hat{r})$ a equação 3.11 muda para um número finitos de amostras, a equação é dada por:

$$\hat{r}_x(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n+k)x^*(n) \tag{3.12}$$

para garantir a inclusão dos valores de $x(n)$ que caem fora do intervalo $[0, N-1]$ podemos reescrever a equação da seguinte forma:

$$\hat{r}_x(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} x(n+k)x^*(n) \tag{3.13}$$

para os valores de $k < 0$ será usada a propriedade de simetria do função de autocorrelação e $\hat{r}_x(k)$ é nula para $|k| \geq N$. Então a transformada discreta da autocorrelação estimada é uma estimação da densidade espectral potencia conhecida como periodograma (HAYES, 1996).

$$\hat{P}_{per}(e^{j\omega}) = \sum_{k=-N+1}^{N-1} r_x(k)e^{-jk\omega} \tag{3.14}$$

Embora definido em termos da sequência de autocorrelação estimada $\hat{r}_x(k)$, normalmente se expressa o periodograma diretamente em termos do processo $x(n)$. Para expressar em termos do processo é necessário realizar o seguinte procedimento, seja $x_N(n)$ o sinal finito de comprimento N que é igual a $x(n)$ ao longo do intervalo $[0, N-1]$, e zero caso contrário. Seja

$x_N(n)$ o produto de $x(n)$ com uma janela retangular $w_R(n)$. Agora em termos de $x_N(n)$, a função de autocorrelação estimada é dada por:

$$\hat{r}_x(k) = \frac{1}{N}\sum_{k=-\infty}^{\infty} x_N(n+k)x^*(n) = \frac{1}{N}x_N(k) * x_N^*(-k) \tag{3.15}$$

usando o teorema da convolução e a transformada de Fourier na equação 3.15 temos:

$$\hat{P}_{per}(e^{j\omega}) = \frac{1}{N}X_N(e^{j\omega})X_N^*(e^{j\omega}) = \frac{1}{N}\left|X_N^*(e^{j\omega})\right|^2 \tag{3.16}$$

onde o operador $|\ |$ representa o módulo ou valor absoluto de uma variável e $X_N^*(e^{j\omega})$ é o conjugado da transformada discreta de Fourier para N amostras de $x_N(n)$, como apresentado na equação 3.17.

$$\hat{X}_N(e^{j\omega}) = \sum_{k=-\infty}^{\infty} x_N(n)e^{-jn\omega} = \sum_{n=0}^{N-1} x(n)e^{-jn\omega} \tag{3.17}$$

Podemos verificar que o periodograma é proporcional ao quadrado da magnitude da transformada discreta de Fourier de $x_N(n)$ e pode ser mais fácil de implementar seguindo a seguinte ordem:

- Calcula-se com os dados $x_N(n)$ a transformada discreta de Fourier e acha $x_N(k)$;
- Calcula-se o quadrado da magnitude de $(|X_N(k)|^2)/N$.

Então, pode-se afirmar que o periodograma é proporcional ao quadrado da magnitude da transformada de Fourier de um sinal janelado $x_N(n) = x(n)w_R(n)$.

Uma implementação em Matlab\Octave da função para estimação do periodograma é mostrado no algoritmo 1.

O sinal $x_N(n)$ utilizado no periodograma, foi janelado por uma janela retangular. Porém é comum o uso de outros tipos de janelamento diferente do retangular, destacam-se as janelas de: Bartlett, Hanning, Hamming, Blackman, Flattopwin, Gaussiana e Taylorwin. O cálculo do periodograma modificada é dado por:

$$\hat{P}_M(e^{j\omega}) = \frac{1}{NU}\left[\sum_{n=-\infty}^{\infty} x(n)w(n)e^{-jn\omega}\right]^2 \tag{3.18}$$

Algoritmo 1: Algoritmo para calcular o Periodograma

Entrada: *nargin*,x,inicio,fim
Resultado: Periodograma(Px)

```
1 x←x(:) ;                    /* A variável interna recebe o vetor de dados */
2 se nargin = 1 então ▷ A condição é verdadeira se a única entrada for x
3 |   inicio←1
  |   fim←length(x);                /* length calcula o tamanho da amostra */
4 fim
5 Px←abs(fft(x(inicio : fim),256)).^2/(fim − inicio + 1) ;        /* Janela de 256 */
6 Px(1)←Px(2)
  retorna Px
```

onde $w(n)$ é a janela utilizada e U é dado pela média quadrática da magnitude de $w(n)$.

$$U = \frac{1}{L}\sum_{n=0}^{L-1} |w(n)|^2 \tag{3.19}$$

No algoritmo 2 é apresentado uma função em Matlab para uso no cálculo da estimativa do periodograma modificado com a possibilidade de uso das janelas de Hamming, Hanning, Barlett e Blackman.

Para iniciar o periodograma do método de Welch temos que descrever o método de Bartlett (BARTLETT, 1950), pois o método de Welch é uma variação do método de Bartlett. O método de periodograma de Bartlett, que, diferentemente do periodograma, ele produz uma estimativa do espectro de potência, a motivação para este método vem da observação do valor esperado do periodograma converge para $P_x(e^{j\omega})$ à medida que o comprimento do registro de dados N vai para o infinito.

$$\lim_{N\to\infty} E[\hat{P}_{per}(e^{j\omega})] = P_x(e^{j\omega}) \tag{3.20}$$

Pela Eq. 3.20, conclui-se que se pudermos encontrar uma média da estimativa do periodograma, então esta teremos uma estimativa consistente de $P_x(e^{j\omega})$. Seja, $x_i(n)$ de $i = 1$ até K, onde temos K realizações de um processo não correlacionado de um processo estocástico $x(n)$ sobre o intervalo de $0 < n \leq L$. Sabe-se que a estimativa do periodograma de $x_i(n)$ é dado por

$$\hat{P}^{(i)}_{per}(e^{j\omega}) = \frac{1}{L}\left[\sum_{n=0}^{L-1} x_i(n)e^{-jn\omega}\right] \quad ,i = 1,2,...,k \tag{3.21}$$

Algoritmo 2: Algoritmo para calcular o Periodograma modificado com Janelamento

Entrada: $x, tipo\_janela, inicio, fim$
Resultado: periodograma_modificado(Pxm)

$x \leftarrow x(:)$; /* A variável interna recebe o vetor de dados */
se $nargin = 2$ **então** ▷ A condição é verdadeira se as entradas forem somente x e o tipo_janela
  $inicio \leftarrow 1$
  $fim \leftarrow length(x)$; /* length calcula o tamanho da amostra */
fim
$N \leftarrow fim - inicio + 1$
$w \leftarrow ones(N, 1)$; /* w é a Janela */
se $tipo\_janela = 2$ **então**
  w←hamming(N)
fim
se $tipo\_janela = 3$ **então**
  w←hanning(N)
fim
se $tipo\_janela = 4$ **então**
  w←bartlett(N)
fim
se $tipo\_janela = 5$ **então**
  w←blackman(N)
fim
$xw \leftarrow x(inicio : fim).*w/norm(w)$; /* nova entrada modificada pela Janela normalizada */
$Pxm \leftarrow N * periodograma(xw)$
retorna Pxm

e a média de conjunto para K realizações é dada por

$$\hat{P}_x(e^{j\omega}) = \frac{1}{K}\sum_{i=1}^{K}\hat{P}_{per}^{(i)}(e^{j\omega}) \tag{3.22}$$

Calculando o valor esperado para $\hat{P}_x(e^{j\omega})$

$$E[\hat{P}_x(e^{j\omega})] = E[\hat{P}_{per}^{(i)}(e^{j\omega})] = \frac{1}{2\pi}P_x(e^{j\omega}) * W_B(e^{j\omega}) \tag{3.23}$$

onde $W_B(e^{j\omega})$ é a transformada de Fourier da janela de Bartlett no intervalo de $[-L, L]$. Como assumimos que os dados são não correlacionados a variância de $\hat{P}_x(e^{j\omega})$ é:

$$Var(\hat{P}_x(e^{j\omega})) = \frac{1}{K}Var(\hat{P}_{per}^{(i)}(e^{j\omega})) \approx \frac{1}{K}P_x^2(e^{j\omega}) \tag{3.24}$$

A abordagem recém descrita é complexa na prática pois normalmente não se tem K realizações de um processo e sim uma única realização com N amostras. Então, Bartlett propôs que $x(n)$ seja particionado em K sequencias não sobrepostas de tamanho L, onde $N = LK$. A estimativa da PSD do sinal de acordo com essa proposta é dada então por:

$$\hat{P}_B(e^{j\omega}) = \frac{1}{N}\sum_{i=0}^{K-1}\left[\sum_{n=0}^{L-1} x(n+iL)e^{-jn\omega}\right]^2 \tag{3.25}$$

onde $x_i(n) = x(n+iL)$ para $n = 0,1,...,K$ e $i = 0,1,...,K-1$. Uma implementação desse método utilizando uma função do Matlab é mostrada no Algoritmo 3

Algoritmo 3: Algoritmo para calcular o Periodograma do método de Barlett

Entrada: x, K_seq
Resultado: Periodograma de Barlett(Pxb)

```
1 L←floor(length(x)/K_seq)
  Pxb←0
  inicio←1
  for i ← 1 to K_seq do
2 |  Pxb←Pxb+periodograma(x((inicio):(inicio+L−1)))/K_seq
  |  inicio←inicio+L;            /* atualização do novo valor de início */
3 end
4 retorna Pxb
```

(WELCH, 1967) propôs duas alterações no método de Barlett. A primeira é permitir que a sequencia de dados $x_i(n)$ se sobreponham e a segunda é permitir que o janelamento dos dados $w(n)$ seja aplicado a cada sequência, desta forma produz um conjunto de periodogramas modificados que devem ser calculados pela média.

Com um sinal $x_i(n)$, esse sinal é formado por sucessiva sequências deslocadas de D amostras ao longo de L pontos na sequencia, ou seja, $x_i(n) = x(n+iD)$ para $i = 0,1,...,L-1$. Então, a quantidade de amostras sobrepostos entre $x_i(n)$ e $x_{1+i}(n)$ é $L-D$ pontos. O calculo da PSD estimada de Welch é dado por

$$\hat{P}_W(e^{j\omega}) = \frac{1}{KLU}\sum_{i=0}^{K-1}\left[\sum_{n=0}^{L-1} w(n)x(n+iD)e^{-jn\omega}\right]^2 \tag{3.26}$$

onde U é dado pela Eq. 3.19. O valor esperado da estimativa obtida pelo método de Welch é

$$E(\hat{P}_W(e^{j\omega})) = \frac{1}{2\pi LU}P_x(e^{j\omega}) * \left|W(e^{j\omega})\right|^2 \tag{3.27}$$

onde $\left|W(e^{j\omega})\right|$ é a transformada de Fourier da janela escolhida. Portanto, o periodograma de Welch é um periodograma de Barlett estendido.

Algoritmo 4: Algoritmo para calcular o Periodograma de Welch

```
/* sobreposicao é uma variável de entrada que determina o tamanho da
   sobreposição da amostra no janelamento.                          */
Entrada: x, L, sobreposicao, win
Resultado: Periodograma de welch(Pxw)
se sobreposicao >= 1|sobreposicao < 0 então
    error('sobreposição invalida')
fim
inicio←1
Pxw←0
L_com_sobreposicao←(1 − sobreposicao) ∗ L
K_seq←1 + floor((length(x) − L)/(L_com_sobreposicao))
for i ← 1 to K_seq do
    Pxw←Pxw + periodograma_modificado(x, tipo_janela, inicio, inicio + L −
    1)/K_seq
    inicio←inicio + L_com_sobreposicao
end
retorna Pxw
```

3.3 Construção do vetor de atributos

Consideremos um conjunto de $N = 23$ canais, dos quais obtemos 23 sinais EEG de uma certa duração para cada paciente. Para o conjunto de dados com o qual estamos trabalhando, cada sinal EEG dura 1 hora, amostrado a uma taxa de 256Hz. Para um determinado canal de EEG, os vetores de atributos são construídos a cada dois segundos. Os sucessivos segmentos de dois segundos de duração (chamados de épocas EEG) são processados por uma janela de tempo de duração $L = 2$ segundos. Para uma taxa de amostragem de 256Hz, cada segmento contém 512 amostras.

▷ **Periodograma de Welch** - Os vetores de atributos extraídos usando o método de Welch para um paciente são construídos de acordo com as seguintes etapas de acordo com a Fig. 11, seguindo a metodologia proposta em (SHOEB; GUTTAG, 2010):

- **Passo 1** - Para a época atual de EEG de dois segundos, aplique o método de periodograma de Welch. Repita-o para todos os N canais do EEG.
- **Passo 2** - Aplique uma escala logarítmica aos valores PSD resultantes para convertê-los em decibéis (dB).

• **Passo 3** - Segmente o PSD resultante (em dB) em $M = 8$ bandas de frequência linearmente espaçadas cobrindo o intervalo de 0,5 a 25Hz e, em seguida, compute a energia média dentro de cada banda. Para o canal N, este procedimento leva ao cálculo de $M = 8$ valores de atributos $x_{1,N}$, $x_{2,N}$, ..., $x_{M,N}$.

• **Passo 4** - Dentro de cada época de 2 segundos de EEG no tempo $t = T$, concatene as energias de $M = 8$ extraídas de cada $N = 23$ EEG canais. Esse processo forma um vetor de atributos $\mathbf{X}_T$ de dimensão $M \times N = 184$, definido como

$$\mathbf{X}_T = [x_{1,1} \;\; x_{2,1} \;\; \cdots \;\; x_{M,1} \mid \cdots \mid x_{1,N} \;\; x_{2,N} \;\; \cdots \;\; x_{M,N}]^\top \tag{3.28}$$

• **Passo 5** - Crie um vetor de atributos que seja o resultado da concatenação dos vetores de atributos de $W = 3$ épocas consecutivas de EEG, mas sem nenhumas superposição. O vetor de recurso aumentado $\mathscr{X}_T$ é definido como

$$\mathscr{X}_T = [\mathbf{X}_T \;\; \mathbf{X}_{T-L} \;\; \cdots \;\; \mathbf{X}_{T-(W-1)L}]^\top, \tag{3.29}$$

e tem dimensão $W \times M \times N = 552$.

Em relação ao Passo 5, vale ressaltar que especialistas na área médica consideram como crise epilética uma leitura de EEG anormal que persista por períodos mínimos de 6 a 10 segundos (LOGAR *et al.*, 1994). Para incorporar este fato, estabelecemos $W = 3$ para que os classificadores avaliados levem em consideração a evolução dos vetores de atributos ao longo de pelo menos um período de 6 segundos.

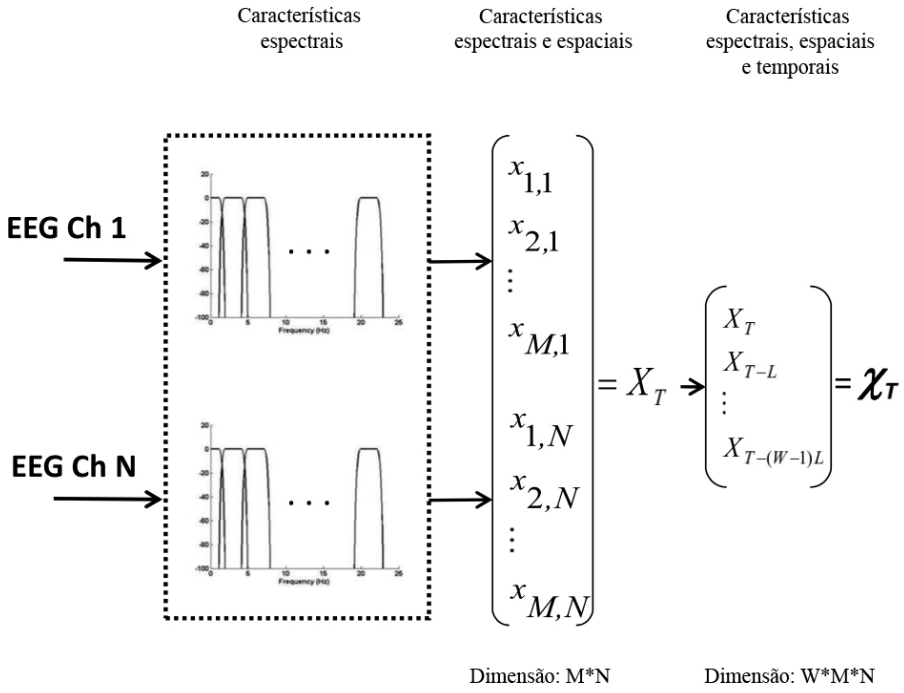
▷ **Coeficientes LPC** - A construção do vetor de atributos usando coeficientes LPC também envolve segmentos de épocas de EEG de dois segundos. No entanto, em vez de especificar o número M de bandas de frequência sobre as quais calculamos a energia por época de EEG, precisamos especificar a ordem p do modelo AR (p). De acordo com a função de autocorrelação parcial (FACP) do sinal, definimos $p = 4$. Valores maiores não melhoraram consideravelmente as acurácias da classificação, enquanto valores menores levaram a uma degradação no desempenho.

Assim, os vetores de atributos construídos por meio do método LPC para um paciente são obtidos da seguinte forma:

• **Passo 1** - Para a atual época de EEG de dois segundos, aplique a equação de Yule-Walker para estimar os coeficientes de p correspondentes do modelo AR. Repita-o para todos os canais N EEG.

• **Passo 2** - Dentro de cada época de EEG no tempo $t = T$, concatene os coeficientes de $p = 4$ estimados para cada um dos canais $N = 23$ EEG. Este processo forma um vetor de recurso

Figura 11 – Construção dos vetores de atributos para a detecção de crises epiléticas a partir dos N canais de EEG para um paciente (SHOEB; GUTTAG, 2010).



$\mathbf{X}_T$ da dimensão $p \times N = 92$, definido como

$$\mathbf{X}_T = [a_{1,1} \;\; a_{1,2} \;\; \cdots \;\; a_{1,p} \mid \cdots \mid a_{N,1} \;\; a_{N,2} \;\; \cdots \;\; a_{N,p}]^\top. \tag{3.30}$$

• **Passo 3** - Crie um vetor de característica que seja o resultado da concatenação dos vetores de atributos de $W = 3$ períodos consecutivos, mas sem superposição. O vetor aumentado $\mathcal{X}_T$ tem dimensão $W \times p \times N = 276$:

$$\mathcal{X}_T = [\mathbf{X}_T \;\; \mathbf{X}_{T-L} \;\; \cdots \;\; \mathbf{X}_{T-(W-1)L}]^\top. \tag{3.31}$$

3.3.1 Rotulação das classes

Deve-se notar que, devido à própria natureza da anomalia a ser detectada, em cada amostra de sinal de EEG, há uma distribuição irregular das classes, constando muito mais vetores de atributos rotulados como *normal* (-1, classe negativa) do que como *crise epilética* (+1, classe positiva).

Em média, para cada amostra, apenas 2 % dos sinais de EEG analisados correspondem a intervalos que contêm convulsões. Isso implica que as tarefas de detecção de crises são altamente desequilibradas, implicando em uma menor perfomance para classificadores sensíveis

a esse desequilíbrio. Para efeito de exemplificação, cada amostra tem duração de 1 hora como já mencionado e cada crise dura em torno de 1 minuto.

Para lidar com as categorias desbalanceadas, foi equalizado propositadamente a proporção de casos positivos e negativos por paciente a cada amostra. Assim, para cada quantidade de dados extraído do momento de ocorrência da crise epilética, uma mesma quantidade de dados normal, imediatamente anterior a esta, foi extraído para compor os vetores de atributos.

3.4 Resumo do capítulo

Esse capítulo abordou os principais métodos de extração de features, iniciando com a estimação de parâmetros usando os coeficientes de uma codificação preditiva linear a partir de um modelo autorregressivo de ordem p (AR(p)), utilizando a expressão de Yule-Walker e o métodos dos momentos. Em seguida foi apresentado o método da estimação de densidade espectral de potência no qual foi realizado uma revisão destacando o periodograma, o periodograma modificado, o periodograma de Barlett, o periodograma de Welch e a autocorrelação parcial. Adicionalmente, foi apresentado toda a metodologia de composição do vetores de atributos pelos dois métodos propostos.

4 CLASSIFICADORES

O problema de interesse, detecção de crises epilépticas, geralmente é tratado como um problema de classificação binária. A este respeito, o classificador tem que diferenciar entre os períodos de atividade normal ou crise epilética.

Diante da complexidade e característica singular do sinal EEG já detalhado nos capítulos anteriores, implementamos nesse trabalho um grupo bem variado de classificadores com o intuito de testar o desempenho de cada um deles diante da classificação do sinal EEG.

Neste capítulo, apresentaremos detalhadamente os seis classificadores que iremos avaliar no capítulo 6. A seguir, iremos justificar a escolha de todos e então detalhar a conceituação matemática de cada um deles. Os classificadores relacionados para avaliação foram os seguintes:

- Modelos lineares
  - mínimos quadrados (MQ) (CHARNES *et al.*, 1976; ALDRICH, 1998)
- Redes neurais randomizadas
  - random vector functional link (RVFL) (PAO *et al.*, 1994; ZHANG; SUGANTHAN, 2016a)
  - extreme learning machine (ELM) (HUANG *et al.*, 2015)
- Métodos de kernel randomizados
  - random kitchen sinks (RKS) (RAHIMI; RECHT, 2009)
- Redes neurais não randomizadas
  - Perceptron Multicamadas (MLP) (HAYKIN, 1998; ROSENBLATT, 1961)
- Métodos de kernel não randomizados
  - Máquinas de Vetor Suporte (SVM) (VAPNIK, 1995; VAPNIK, 1998)

A escolha dos classificadores foi balizada por 3 hipóteses de trabalho, a saber:

- Hipótese 1: A tarefa de detecção de crises epiléticas pode ser satisfatóriamente resolvida por um classificador linear.
- Hipótese 2: Em comparação com o classificador linear, classificadores não lineares convecionais, tais como o MLP e SVM possuem melhor desempenho.
- Hipótese 3: Em relação aos classificadores MLP e SVM, classificadores não lineares randomizados, tais como RVFL, ELM e RKS, apresentam melhor desempenho.

Visando todas as vantagens dos classificadores randomizados, tais como rapidez de projeto e execução, escolhemos três deles: ELM, RVFL e RKS, sendo dois deles de origem, composição e estrutura neural e o terceiro constituído pelo método de kernel. Para referenciar o experimento e comparar a performance desses classificadores randomizados com poderosos classificadores já consagrados na literatura, escolhemos o MLP de origem neural e o SVM com o método de kernel. Além desses, incluimos o classificador linear MQ.

A partir do exposto, iremos descrever cada classificador individualmente, expondo todo o arcabouço teórico necessário para reproduzi-los.

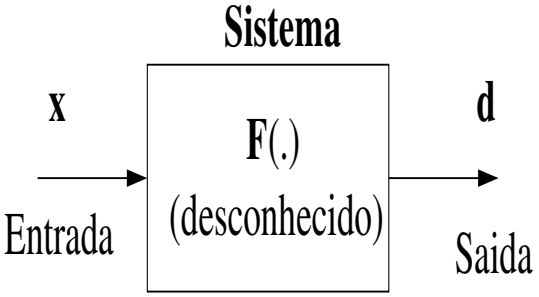
4.1 *Definições Preliminares*

De início, vamos assumir que existe uma lei matemática $\mathbf{F}(\cdot)$, também chamada aqui de função ou mapeamento, que relaciona um vetor de entrada qualquer, $\mathbf{x} \in \mathbb{R}^{p+1}$, com um vetor de saída, $\mathbf{d} \in \mathbb{R}^m$. Esta relação, representada genericamente na Figura 12, pode ser descrita matematicamente da seguinte forma:

$$\mathbf{d} = \mathbf{F}[\mathbf{x}] \tag{4.1}$$

em que se assume que $\mathbf{F}(\cdot)$ é totalmente desconhecida, ou seja, não sabemos de antemão quais são as *fórmulas* usadas para associar um vetor de entrada $\mathbf{x}$ com seu vetor de saída $\mathbf{d}$ correspondente.

Figura 12 – Representação simplificada de um mapeamento entrada-saída genérico.



O mapeamento $\mathbf{F}(\cdot)$ pode ser tão simples quanto um mapeamento linear, tal como

$$\mathbf{d} = \mathbf{M}\mathbf{x} \tag{4.2}$$

em que $\mathbf{M}$ é uma matriz de dimensão $(p+1) \times m$. Contudo, $\mathbf{F}(\cdot)$ pode ser bastante complexo, envolvendo relações não-lineares entre as variáveis de entrada e saída. É justamente o funcionamento da relação matemática $\mathbf{F}(\cdot)$ que se deseja *imitar* através do uso de algoritmos adaptativos, tais como as redes neurais.

Supondo que a única fonte de informação que nós temos a respeito de $\mathbf{F}(\cdot)$ é conjunto finito de N pares entrada-saída observados (ou medidos), ou seja:

$$\begin{array}{ll} \mathbf{x}_1, & \mathbf{d}_1 \\ \mathbf{x}_2, & \mathbf{d}_2 \\ \vdots & \vdots \\ \mathbf{x}_N, & \mathbf{d}_N \end{array} \tag{4.3}$$

Os pares entrada-saída mostrados acima podem ser representados de maneira simplificada como $\{\mathbf{x}_\mu, \mathbf{d}_\mu\}$, em que μ é um apenas índice simbolizando o μ-ésimo par do conjunto de dados. Uma maneira de se adquirir conhecimento sobre $\mathbf{F}(\cdot)$ se dá exatamente através dos uso destes pares.

Para isto pode-se utilizar uma rede neural qualquer para implementar um mapeamento entrada-saída aproximado, representado como $\hat{\mathbf{F}}(\cdot)$, tal que:

$$\mathbf{y}_\mu = \hat{\mathbf{F}}[\mathbf{x}_\mu] \tag{4.4}$$

em que $\mathbf{y}_\mu$ é a saída gerada pela rede neural em resposta ao vetor de entrada $\mathbf{x}_\mu$. Esta saída, espera-se, seja muito próxima da saída real $\mathbf{d}_\mu$. Dá-se o nome de *Aprendizado Indutivo* ao processo de obtenção da relação matemática geral $\hat{\mathbf{F}}(\cdot)$ a partir de apenas alguns pares $\{\mathbf{x}_\mu, \mathbf{d}_\mu\}$ disponíveis.

A seguir serão descritos modelos de aprendizagem indutivo propostos com o intuito de obter uma representação aproximada de um mapeamento entrada-saída genérico.

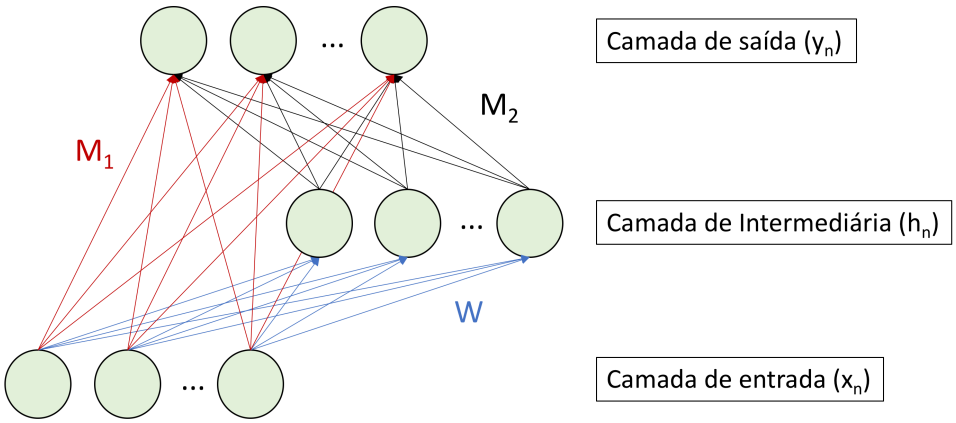
4.2 Random Vector Functional Link Network

As redes randomizadas são variações da rede neural artificial (ANN) que é uma família de métodos de aprendizagem não paramétricos, inspirado na rede neural biológica, para estimar ou aproximar funções que podem depender de um grande número de entradas e saídas, a partir de um mapeamento não-linear. Devido a complexidade e custo matemático para encontrar os melhores pesos para a camada de neurônios não-lineares, os métodos baseados em aleatorização solucionam esse problema, inicializando aleatoriamente as configurações de rede (como as pesos) e alguns parâmetros da rede durante o treinamento.

Assim, como primeira rede randomizada, apresentamos o modelo neural RVFL que é uma rede neural randomizada com uma camada oculta de propagação direta.

Em seu modelo, é projetado dois caminhos para o processamento das informações a partir da camada de entrada até os neurônios da camada de saída. Esses caminhos são então adicionados para formar a saída da rede conforme a figura 13.

Figura 13 – Estrutura da Rede RVFL



Fonte: Construída pelo autor.

Matematicamente, nós temos

$$y_n^{(1)} = \mathbf{m}_1^T \mathbf{x}_n, \tag{4.5}$$

onde $\mathbf{m}_1 \in \mathbb{R}^p$ é vetor de pesos correspondente[1].

O segundo caminho processa os vetores de entrada através de uma camada oculta de q ($q \geq 1$) neurônios não lineares; ou seja,

$$y_n^{(2)} = \mathbf{m}_2^T \mathbf{h}_n, \tag{4.6}$$

onde $\mathbf{m}_2 \in \mathbb{R}^q$ é o vetor de pesos correspondente e $\mathbf{h}_n \in \mathbb{R}^q$ é o vetor de ativação da camada oculta, ou seja o vetor contendo as saídas da camada oculta em resposta ao vetor de entrada atual $\mathbf{x}_n$. O vetor $\mathbf{h}_n$ é calculado como:

$$\mathbf{h}_n = \phi(\mathbf{W}\mathbf{x}_n) = [\phi(\mathbf{w}_1^T \mathbf{x}_n + b_1), \ldots, \phi(\mathbf{w}_q^T \mathbf{x}_n + b_q)]^T, \tag{4.7}$$

onde $\phi(\cdot)$ é uma função de ativação não linear (por exemplo, sigmoidal) que opera em cada componente do seu vetor de argumento, $\mathbf{W}$ é a matriz de pesos $q \times p$ e b_i, $i = 1, .., q$, representa

[1] Assumimos que todos vetores são vetores colunas, a menos que seja afirmado o contrário.

o limiar do i-ésimo neurônio oculto. Os vetores de peso $\mathbf{m}_1$ e $\mathbf{m}_2$ são estimados a partir de dados, enquanto as entradas da matriz $\mathbf{W}$ e os limiares b_i são aleatoriamente amostrados de uma distribuição uniforme ou normal, uma vez que os pesos w_{ij}, $i = 1,\ldots,q$ e $j = 0,\ldots,p$, tenham sido inicializados com valores aleatórios. Formalmente, podemos escrever:

$$w_{ij} \sim U(a,b) \qquad \text{ou} \qquad w_{ij} \sim N(0,\sigma^2) \tag{4.8}$$

em que $U(a,b)$ é um número (pseudo-)aleatório uniformemente distribuído no intervalo (a,b), enquanto $N(0,\sigma^2)$ é um número (pseudo-)aleatório normalmente distribuído com média zero e variância σ^2.

Se adicionarmos as saídas de ambos os caminhos, obtemos

$$y_n = y_n^{(1)} + y_n^{(2)} = \mathbf{m}_1^T \mathbf{x}_n + \mathbf{m}_2^T \mathbf{h}_n = [\mathbf{m}_1^T \mid \mathbf{m}_2^T] \begin{bmatrix} \mathbf{x}_n \\ - \\ \mathbf{h}_n \end{bmatrix} = \mathbf{m}^T \mathbf{z}_n, \tag{4.9}$$

onde $\mathbf{m} = [\mathbf{m}_1^T \mid \mathbf{m}_2^T]^T$ de dimensão $(p+q)\times 1$ é o vetor obtido a partir da concatenação dos vetores de peso $\mathbf{m}_1$ e $\mathbf{m}_2$. Pela mesma razão, $\mathbf{z}_n$ de dimensão $(p+q)\times 1$ é o vetor obtido a partir da concatenação da entrada atual $\mathbf{x}_n$ e o vetor de ativação da camada oculta $\mathbf{h}_n$.

O vetor de pesos $\mathbf{m}$ pode ser prontamente estimado através do método dos mínimos quadrados ordinários (OLS, do inglês do ordinary least squares) por meio da seguinte expressão:

$$\mathbf{m} = (\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\mathbf{d}, \tag{4.10}$$

onde $\mathbf{Z} = [\mathbf{z}_1 \,|\, \mathbf{z}_2 \,|\, \cdots \,|\, \mathbf{z}_{N_1}]$ de dimensão $(p+q)\times N_1$ é uma matriz cujas N_1 colunas são os vetores estendidos $\mathbf{z}_n = [\mathbf{x}_n^T \mid \mathbf{h}_n^T]^T \in \mathbb{R}^{p+q}$, $n = 1,\ldots,N_1$, onde N_1 é o número de padrões de entrada disponíveis para o treinamento. O vetor $\mathbf{d}$ é definido na Eq.(4.3). Para evitar problemas numéricos, uma versão regularizada da Eq.(4.10) é comumente usada, ou seja

$$\mathbf{m} = (\mathbf{Z}\mathbf{Z}^T + \lambda\mathbf{I})^{-1}\mathbf{Z}\mathbf{d}, \tag{4.11}$$

onde a constante $\lambda > 0$ é chamada de parâmetro de regularização.

Em Igelnik & Pao 1995, encontramos algumas justificativas teóricas para o aprimoramento tanto da RVFL como de outras redes neurais com neurônios ocultos implementados como produto de funções univariadas ou funções de base radial. Eles formulam o problema

como um limite - representação integral da função a ser aproximada. Então, a representação integral do limite é aproximada usando o método de Monte-Carlo. Verificou-se que, com pesos e viés, da camada de entrada à camada oculta, amostrada de uma distribuição uniforme com um intervalo adequado, a rede RVFL é um aproximador universal eficiente para funções contínuas em conjuntos de dimensões finitas limitadas. De fato, o erro geral da aproximação pode ser limitado pela soma do erro de aproximação da função pela integral e o erro de aproximação integral pelos métodos de Monte-Carlo.

Já em Zhang & Suganthan 2015, foi realizada uma avaliação abrangente da RVFL, onde foi mostrado que o caminho direto da camada de entrada para saída desempenha um papel fundamental na capacidade de classificação da RVFL. Em Chen 1996, o autor mostra que o número máximo de nós ocultos é $N - r - 1$ para um RVFL com um limiar constante para aprender um mapeamento dentro de uma determinada precisão com base em um conjunto de dados de tamanho N e com vetores de atributos de dimensão n, onde r é o posto do conjunto de dados. Um método de aprendizagem on-line também é proposto. Além disso, um método de mínimos quadrados ponderado robusto é investigado para eliminar outliers. Em Chen & Wan 1999, um algoritmo de atualização passo a passo dinâmico foi proposto quando um novo neurônio ou um novo dado são adicionados com base na mesma solução pseudo-inversa. Em Chen & Wan 1999, os autores também propuseram vários métodos para refinar o modelo para lidar com o problema de valores singulares pequenos da matriz Z, que podem ser causados por uma matriz Z mal-condicionada e resultarão em pesos muito grandes que amplificarão ainda mais o ruído nos dados do teste. Algumas soluções potenciais em Chen & Wan 1999 incluem: i) investigação de um limite superior nos pesos, ii) Poda de valores singulares e investigação da relação entre os valores de corte e o desempenho da rede em termos de erro de predição, iii) Método de aprendizagem de mínimos quadrados ortogonais, um método de regularização ou métodos de validação cruzada.

4.3 Extreme Learning Machine

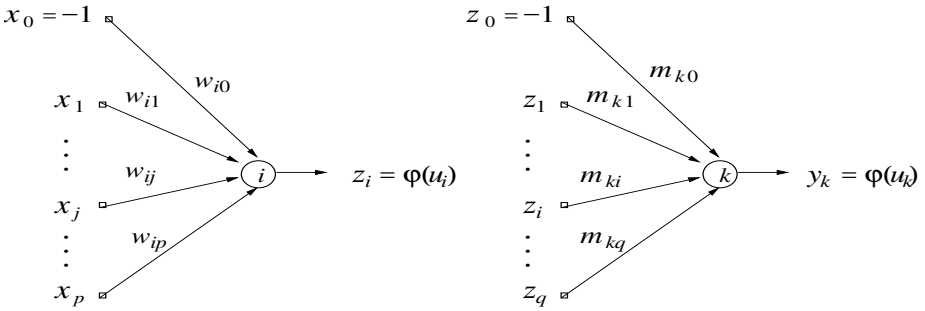
4.3.1 Máquina de Aprendizado Extremo

Estamos considerando nas definições e cálculos a seguir uma arquitetura de rede neural do tipo *feedforward* (i.e. sem realimentação) com apenas uma camada de neurônios ocultos, conhecida como Máquina de Aprendizado Extremo (ELM, do inglês *Extreme Learning*

Machine) (HUANG *et al.*, 2006). Esta arquitetura de rede neural é semelhante à rede MLP, porém apresenta uma fase de aprendizado infinitamente mais rápida que a da rede MLP. Começaremos com a descrição de sua arquitetura, para em seguida escrever sobre o funcionamento e o treinamento da rede ELM.

Os neurônios da camada oculta (primeira camada de pesos sinápticos) são representados conforme mostrado na Figura 14a, enquanto os neurônios da camada de saída segunda camada de pesos sinápticos) são representados conforme mostrado na Figura 14b.

Figura 14 – Camadas de Neurônios: (a) Neurônio da camada escondida. (b) Neurônio da camada de saída.



O vetor de pesos associado a cada neurônio i da camada escondida, também chamada de *camada oculta* ou *camada intermediária*, é representado como

$$\mathbf{w}_i = \begin{pmatrix} w_{i0} \\ \vdots \\ w_{ip} \end{pmatrix} = \begin{pmatrix} \theta_i \\ \vdots \\ w_{ip} \end{pmatrix} \tag{4.12}$$

em que θ_i é o limiar (*bias* ou *threshold*) associado ao neurônio i. Os neurônios desta camada são chamados de neurônios escondidos por não terem acesso direto à saída da rede, onde são calculados os erros de aproximação.

De modo semelhante, o vetor de pesos associado a cada neurônio k da camada de saída é representado como

$$\mathbf{m}_k = \begin{pmatrix} m_{k0} \\ \vdots \\ m_{kq} \end{pmatrix} = \begin{pmatrix} \theta_k \\ \vdots \\ m_{kq} \end{pmatrix} \tag{4.13}$$

em que θ_k é o limiar associado ao neurônio de saída k. O treinamento da rede ELM se dá em duas etapas, que são descritas a seguir.

4.3.2 Fase 1: Inicialização Aleatória dos Pesos dos Neurônios Ocultos

Esta etapa de funcionamento da rede ELM envolve o cálculo das ativações e saídas de todos os neurônios da camada escondida e de todos os neurônios da camada de saída, uma vez que os pesos w_{ij}, $i = 1,\ldots,q$ e $j = 0,\ldots,p$, tenham sido inicializados com valores aleatórios. Formalmente, podemos escrever:

$$w_{ij} \sim U(a,b) \qquad \text{ou} \qquad w_{ij} \sim N(0,\sigma^2) \tag{4.14}$$

em que $U(a,b)$ é um número (pseudo-)aleatório uniformemente distribuído no intervalo (a,b), enquanto $N(0,\sigma^2)$ é um número (pseudo-)aleatório normalmente distribuído com média zero e variância σ^2.

Em ambientes de programação, tais como Matlab ou Octave, esta fase é facilmente implementada em uma linha apenas de código. Para isso, precisamos definir uma matriz de pesos $\mathbf{W}$, com q linhas e $p+1$ colunas:

$$\mathbf{W} = \begin{pmatrix} w_{10} & w_{11} & \cdots & w_{1p} \\ w_{20} & w_{21} & \cdots & w_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ w_{q0} & w_{q1} & \cdots & w_{qp} \end{pmatrix}_{q\times(p+1)} = \begin{pmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_q^T \end{pmatrix} \tag{4.15}$$

em que notamos que i-ésima linha da matriz $\mathbf{W}$ é composta pelo vetor de pesos do i-ésimo neurônio oculto.

Uma vez definida a matriz $\mathbf{W}$, podemos realizar a etapa 1 através das seguintes linhas de código Octave, caso os pesos sejam inicializados com números aleatórios uniformes:

```
» a=0; b=0.1;     % define intervalo dos pesos
» W=a+(b-a).*rand(q,p+1);     % gera números uniformes
```

ou pelas seguintes linhas se preferirmos números aleatórios gaussianos:

```
» sig=0.1;     % define desvio-padrao dos pesos
» W=sig*randn(q,p+1);     % gera números gaussianos
```

4.3.3 Fase 2: Acúmulo das Saídas dos Neurônios Ocultos

O fluxo de sinais (informação) se dá dos neurônios de entrada para os neurônios de saída, passando obviamente pelos neurônios da camada escondida. Por isso, diz-se que o informação está fluindo no sentido **direto** (*forward*), ou seja:

Entrada $\rightarrow$ Camada Intermediária $\rightarrow$ Camada de Saída

Assim, após a apresentação de um vetor de entrada $\mathbf{x}$, na instante n, o primeiro passo é calcular as ativações dos neurônios da camada escondida:

$$u_i(n) = \sum_{j=0}^{p} w_{ij}x_j(n) + b_i = \mathbf{w}_i^T\mathbf{x}(n) + b_i, \quad i = 1,\ldots,q \tag{4.16}$$

em que T indica o vetor (ou matriz) transposto e q indica o número de neurônios da camada escondida.

A operação sequencial da Eq. (4.16) pode ser feita de uma única vez se utilizarmos a notação vetor-matriz. Esta notação é particularmente útil em ambientes do tipo Matlab/Octave. Neste caso, temos que o vetor de ativações $\mathbf{u}_i(n) \in \mathbb{R}^q$ do i-ésimo neurônio oculto na iteração t é calculado como

$$\mathbf{u}(n) = \mathbf{W}\mathbf{x}(n). \tag{4.17}$$

Em seguida, as saídas correspondentes são calculadas como

$$z_i(n) = \phi_i(u_i(n)) = \phi_i\left(\sum_{j=0}^{p} w_{ij}(n)x_j(n) + b_i\right) = \phi_i\left(\mathbf{w}_i^T(n)\mathbf{x}(n) + b_i\right) \tag{4.18}$$

tal que a função de ativação ϕ assume geralmente uma das seguintes formas:

$$\phi_i(u_i(n)) = \frac{1}{1+\exp[-u_i(n)]}, \qquad \text{(Logística)} \tag{4.19}$$

$$\phi_i(u_i(n)) = \frac{1-\exp[-u_i(n)]}{1+\exp[-u_i(n)]}, \qquad \text{(Tangente Hiperbólica)} \tag{4.20}$$

Em notação matriz-vetor, a Eq. (4.18) pode ser escrita como

$$\mathbf{z}(n) = \phi_i(\mathbf{u}_i(n)) = \phi_i(\mathbf{W}\mathbf{x}(n)). \tag{4.21}$$

em que a função de ativação $\phi_i(\cdot)$ é aplicada a cada um dos q componente do vetor $\mathbf{u}(n)$.

Para cada vetor de entrada $\mathbf{x}(n)$, $t = 1,\ldots,N$, tem-se um vetor $\mathbf{z}(n)$ correspondente, que deve ser organizado (disposto) como uma coluna de uma matriz $\mathbf{Z}$. Esta matriz terá q linhas por N colunas:

$$\mathbf{Z} = [\mathbf{z}_1 \mid \mathbf{z}_2 \mid \cdots \mid \mathbf{z}_N]. \tag{4.22}$$

A matriz $\mathbf{Z}$ será usada na Fase 3 para calcular os valores do pesos dos neurônios de saída da rede ELM.

4.3.4 Fase 3: Cálculo dos Pesos dos Neurônios de Saída

Sabemos que para cada vetor de entrada $\mathbf{x}(n)$, $n = 1,\ldots,N$, tem-se um vetor de saídas desejadas $\mathbf{d}(n)$ correspondente. Se organizamos estes N vetores ao longo das colunas de uma matriz $\mathbf{D}$, então temos que esta matriz terá dimensão m linhas e N colunas:

$$\mathbf{D} = [\mathbf{d}_1 \mid \mathbf{d}_2 \mid \cdots \mid \mathbf{d}_N]. \tag{4.23}$$

Podemos entender o cálculo dos pesos da camada de saída como o cálculo dos parâmetros de um mapeamento linear entre a camada oculta e a camada de saída. O papel de vetor de entrada para a camada de saída no instante n é desempenhado pelo vetor $\mathbf{z}(n)$ enquanto o vetor de saída é representado pelo vetor $\mathbf{d}(n)$. Assim, buscamos determinar a matriz $\mathbf{M}$ que melhor represente a transformação

$$\mathbf{d}(n) = \mathbf{M}\mathbf{z}(n). \tag{4.24}$$

Para isso, podemos usar o método dos mínimos quadrados, também conhecido como método da pseudoinversa. Assim, usando as matrizes $\mathbf{Z}$ e $\mathbf{D}$, a matriz de pesos $\mathbf{M}$ é calculada por meio das seguinte expressão:

$$\mathbf{M} = \mathbf{D}\mathbf{Z}^T \left(\mathbf{Z}\mathbf{Z}^T\right)^{-1}. \tag{4.25}$$

Alguns comentários sobre a matriz $\mathbf{M}$ fazem-se necessários:

- Note que para satisfazer a Eq. (4.24) a matriz $\mathbf{M}$ tem dimensão $m \times q$.
- A k-ésima linha da matriz $\mathbf{M}$, denotado aqui por $\mathbf{m}_k$, $k = 1,2,\ldots,m$, corresponde ao vetor de pesos do k-ésimo neurônio de saída.

4.3.5 Teste e Capacidade de Generalização da Rede ELM

Uma vez determinadas as matrizes de pesos $\mathbf{W}$ e $\mathbf{M}$ temos a rede ELM pronta para uso. Durante o uso da rede ELM, calculamos as ativações dos neurônios da camada de saída por meio da seguinte expressão:

$$a_k(n) = \sum_{i=0}^{q} m_{ki}(n) z_i(n) = \mathbf{m}_k^T \mathbf{z}(n), \quad k = 1,\ldots,m \tag{4.26}$$

em que m é o número de neurônios de saída. Note que as saídas dos neurônios da camada oculta, $z_i(n)$, fazem o papel de entrada para os neurônios da camada de saída.

Em notação vetor-matriz, as operações da Eq. (4.26) podem ser executados de uma só vez por meio da seguinte expressão:

$$\mathbf{a}_n = \mathbf{M}\mathbf{z}_n. \tag{4.27}$$

Para a rede ELM, assumimos que os neurônios de saída usam a função identidade como função de ativação, ou seja, as saídas destes neurônios são iguais às suas ativações, calculadas como:

$$y_k(n) = \phi_k(a_k(n)) = a_k(n). \tag{4.28}$$

Por generalização adequada entende-se a habilidade da rede em utilizar o conhecimento armazenado nos seus pesos e limiares para gerar saídas coerentes para novos vetores de entrada, ou seja, vetores que não foram utilizados durante o treinamento. A generalização é considerada boa quando a rede, durante o treinamento, foi capaz de capturar (aprender) adequadamente a relação entrada-saída do mapeamento de interesse.

4.3.6 Dicas para um Bom Desempenho da Rede ELM

O projeto de uma rede neural envolve a especificação de diversos itens, cujos valores influenciam consideravelmente funcionamento do algoritmo. A seguir especificaremos a lista destes itens juntamente com as faixas de valores que os mesmos podem assumir:

Dimensão do vetor de Entrada (p): Este item pode assumir em tese valores entre 1 e ∞. Porém, existe um limite superior que depende da aplicação de interesse e do custo de se medir (observar) as variáveis x_j. É importante ter em mente que um valor alto para p não indica necessariamente um melhor desempenho para a rede neural, pois pode haver redundância no processo de medição. Neste caso, uma certa medida é, na verdade, a combinação linear de outras medidas, podendo ser descartada sem prejuízo ao desempenho da rede. Quando é muito caro, ou até impossível, medir um elevado número de variáveis x_j, deve-se escolher aquelas que o especialista da área considera como mais relevante ou representativas para o problema. O ideal seria que cada variável x_j, $j = 1, \ldots, p$, "carregasse"informação que somente ela contivesse. Do ponto de vista estatístico, isto equivale a dizer que as variáveis são *independentes* ou *não-correlacionadas* entre si.

Dimensão do vetor de saída (m): Assim como o primeiro item, este também depende da aplicação. Se o interesse está em problemas de aproximação de funções, $\mathbf{y} = F(\mathbf{x})$, o número

de neurônios deve refletir diretamente a quantidades de funções de saída desejadas (ou seja, a dimensão de **y**).

Se o interesse está em problemas de classificação de padrões, a coisa muda um pouco de figura. Neste caso, o número de neurônios deve codificar o número de classes desejadas. É importante perceber que estamos chamando as classes às quais pertencem os vetores de dados de uma forma bastante genérica: classe 1, classe 2, ..., etc. Contudo, à cada classe pode estar associado um rótulo (e.g. classe dos empregados, classe dos desempregados, classe dos trabalhadores informais, etc.), cujo significado depende da interpretação que o especialista na aplicação dá a cada uma delas. Estes rótulos normalmente não estão na forma numérica, de modo que para serem utilizados para treinar a rede ELM eles devem ser convertidos para a forma numérica. A este procedimento dá-se o nome de codificação da saída da rede.

A codificação mais comum define como vetor de saídas desejadas um vetor binário de comprimento unitário; ou seja, apenas uma componente deste vetor terá o valor "1", enquanto as outras terão o valor "0" (ou -1). A dimensão do vetor de saídas desejadas corresponde ao número de classes do problema em questão. Usando esta codificação define-se automaticamente um neurônio de saída para cada classe. Por exemplo, se existem três classes possíveis, existirão três neurônios de saída, cada um representando uma classe. Como um vetor de entrada não pode pertencer a mais de uma classe ao mesmo tempo, o vetor de saídas desejadas terá valor 1 (um) na componente correspondente à classe deste vetor, e 0 (ou -1) para as outras componentes. Por exemplo, se o vetor de entrada $\mathbf{x}(n)$ pertence à classe 1, então seu vetor de saídas desejadas é $\mathbf{d}(n) = [1\ \ 0\ \ 0]^T$. Se o vetor $\mathbf{x}(n)$ pertence à classe 2, então seu vetor de saídas desejadas é $\mathbf{d}(n) = [0\ \ 1\ \ 0]^T$ e assim por diante para cada exemplo de treinamento.

Número de neurônios na camada escondida (q): Encontrar o número ideal de neurônios da camada escondida não é uma tarefa fácil porque depende de uma série de fatores, muitos dos quais não temos controle total. Entre os fatores mais importantes podemos destacar os seguintes:

1. Quantidade de dados disponíveis para treinar e testar a rede.
2. Qualidade dos dados disponíveis (ruidosos, com elementos faltantes, etc.)
3. Número de parâmetros ajustáveis (pesos e limiares) da rede.
4. Nível de complexidade do problema (não-linear, descontínuo, etc.).

O valor de q é geralmente encontrado por tentativa-e-erro, em função da capacidade de *generalização* da rede (ver definição logo abaixo). Grosso modo, esta propriedade avalia o desempenho da rede neural ante situações não-previstas, ou seja, que resposta ela dá quando novos dados de entrada forem apresentados. Se muitos neurônios existirem na camada escondida, o desempenho será muito bom para os dados de treinamento, mas tende a ser ruim para os novos dados. Se existirem poucos neurônios, o desempenho será ruim também para os dados de treinamento. O valor ideal é aquele que permite atingir as especificações de desempenho adequadas tanto para os dados de treinamento, quanto para os novos dados.

Existem algumas fórmulas heurísticas (*ad hoc*) que sugerem valores para o número de neurônios na camada escondida da rede ELM, porém estas regras devem ser usadas apenas para dar um valor inicial para q. O projetista deve sempre treinar e testar várias vezes uma dada rede ELM para diferentes valores de q, a fim de se certificar que a rede neural generaliza bem para dados novos, ou seja, não usados durante a fase de treinamento.

Dentre a regras heurísticas citamos a seguir três, que são comumente encontradas na literatura especializada:

Regra do valor médio - De acordo com esta fórmula o número de neurônios da camada escondida é igual ao valor médio do número de entradas e o número de saídas da rede, ou seja:

$$q = \frac{p+M}{2} \tag{4.29}$$

Regra da raiz quadrada - De acordo com esta fórmula o número de neurônios da camada escondida é igual a raiz quadrada do produto do número de entradas pelo número de saídas da rede, ou seja:

$$q = \sqrt{p \cdot M} \tag{4.30}$$

Regra de Kolmogorov De acordo com esta fórmula o número de neurônios da camada escondida é igual a duas vezes o número de entradas da rede adicionado de 1, ou seja:

$$q = 2p+1 \tag{4.31}$$

Perceba que as regras só levam em consideração características da rede em si, como número de entradas e número de saídas, desprezando informações úteis, tais como número de dados disponíveis para treinar/testar a rede e o erro de generalização máximo aceitável.

Uma regra que define um valor inferior para q levando em consideração o número de dados de treinamento/teste é dada por:

$$q \geq \frac{N-1}{p+2} \tag{4.32}$$

A regra geral que se deve sempre ter em mente é a seguinte: *devemos sempre ter muito mais dados que parâmetros ajustáveis*. Assim, se o número total de parâmetros (pesos + limiares) da rede é dado por $Z = (p+1)\cdot q + (q+1)\cdot M$, então devemos sempre tentar obedecer à seguinte relação:

$$N \gg Z \tag{4.33}$$

Um refinamento da Equação (4.33), proposto por Baum & Haussler (1991), sugere que a relação entre o número total de parâmetros da rede (Z) e a quantidade de dados disponíveis (N) deve obedecer à seguinte relação:

$$N > \frac{Z}{\varepsilon} \tag{4.34}$$

em que $\varepsilon > 0$ é o erro percentual máximo aceitável durante o teste da rede; ou seja, se o erro aceitável é 10%, então $\varepsilon = 0{,}1$. Para o desenvolvimento desta equação, os autores assumem que o erro percentual durante o treinamento não deverá ser maior que $\varepsilon/2$.

Para exemplificar, assumindo que $\varepsilon = 0{,}1$, então temos que $N > 10Z$. Isto significa que para uma rede de Z parâmetros ajustáveis, devemos ter uma quantidade dez vezes maior de padrões de treinamento.

Note que se substituirmos Z na Equação (4.34) e isolarmos para q, chegaremos à seguinte expressão que fornece o valor aproximado do número de neurônios na camada oculta:

$$q \approx \left\lceil \frac{\varepsilon N - M}{p+M+1} \right\rceil \tag{4.35}$$

em que $\lceil u \rceil$ denota o menor inteiro maior que u.

A Equação (4.35) é bastante completa, visto que leva em consideração não só aspectos estruturais da rede ELM (número de entradas e de saídas), mas também o erro máximo tolerado para teste e o número de dados disponíveis. Portanto, seu uso é bastante recomendado.

Funções de ativação (ϕ): Em tese, cada neurônio pode ter a sua própria função de ativação, diferente de todos os outros neurônios. Contudo, para simplificar o projeto da rede é comum adotar a mesma para todos os neurônios. Em geral, escolhe-se a função logística

ou a tangente hiperbólica para os neurônios da camada escondida. Aquela que for escolhida para estes neurônios será adotada também para os neurônios da camada de saída. Em algumas aplicações é comum adotar uma função de ativação linear para os neurônios da camada de saída, ou seja, $\phi_k(u_k(n)) = C_k \cdot u_k(n)$, onde C_k é uma constante (ganho) positiva. Neste caso, tem-se que $\phi_k'(u_k(n)) = C_k$. O fato de $\phi_k(u_k(n))$ ser linear não altera o poder computacional da rede, o que devemos lembrar sempre é que os neurônios da camada escondida devem ter uma função de ativação não-linear, obrigatoriamente.

Avaliação de Desempenho: O desempenhado da rede ELM é, em geral, avaliada com base nos valores do erro quadrático médio (ε_{teste}) por padrão de teste:

$$\varepsilon_{teste} = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(n) = \frac{1}{2N}\sum_{t=1}^{N}\sum_{k=1}^{n}e_k^2(n) \tag{4.36}$$

em que $e_k(n) = d_k(n) - y_k(n)$ é o erro do k-ésimo neurônio de saída na iteração t.

Por outro lado, quando se utiliza a rede para classificar padrões, o desempenho da mesma é avaliado pela *taxa de acerto na classificação*, definida como:

$$P_{acerto} = \frac{\text{Número de vetores classificados corretamente}}{\text{Número de total de vetores}} \tag{4.37}$$

Outras métricas de avaliação do desempenho da rede ELM em tarefas de reconhecimento de padrões são a matriz de confusão e os valores de sensibilidade e especificidade para o caso de problemas de classificação binária.

Para validar a rede treinada, ou seja, dizer que ela está apta para ser utilizada, é importante testar a sua resposta (saída) para dados de entrada diferentes daqueles vistos durante o treinamento. Estes novos dados podem ser obtidos através de novas medições, o que nem sempre é viável. Durante o teste os pesos de saída da rede, em geral, não são ajustados.

Para contornar este obstáculo, o procedimento mais comum consiste em treinar a rede apenas com uma parte dos dados selecionados *aleatoriamente*, guardando a parte restante para ser usada para testar o desempenho da rede. Assim, ter-se-á dois conjuntos de dados, um para treinamento, de tamanho $N_1 < N$, e outro de tamanho $N_2 = N - N_1$. Em geral, escolhe-se N_1 tal que a razão N_1/N esteja na faixa de $0,75$ a $0,90$.

Em outras palavras, se $N_1/N \approx 0,75$ tem-se que 75% dos vetores de dados devem ser selecionados aleatoriamente, sem reposição, para serem utilizados durante o treinamento. Os 25% restantes serão usados para testar a rede. O valor de ε_{teste} calculado com os dados de teste é chamado de *erro de generalização* da rede, pois testa a capacidade da mesma em "extrapolar"o conhecimento aprendido durante o treinamento para novas situações. É

importante ressaltar que, geralmente, o erro de generalização é maior do que o erro de treinamento, pois trata-se de um novo conjunto de dados.

4.3.7 Dicas para um Bom Projeto da Rede ELM

Pré-processamento dos pares entrada-saída Antes de apresentar os exemplos de treinamento para a rede ELM é comum mudar a escala original das componentes dos vetores **x** e **d** para a escala das funções de ativação logística (0 e 1) ou da tangente hiperbólica (-1 e 1). As duas maneiras mais comuns de se fazer esta mudança de escala são apresentadas a seguir:

Procedimento 1: Indicado para quando as componentes x_j do vetor de entrada só assumem valores positivos e a função de ativação, $\phi(u)$, é a função logística. Neste caso, aplicar a seguinte transformação a cada componente de **x**:

$$x_j^* = \frac{x_j}{x_j^{max}} \tag{4.38}$$

em que, ao dividir cada x_j pelo seu maior valor $x_j^{max} = \max_{\forall t}\{x_j(n)\}$, tem-se que $x_j^* \in [0,1]$.

Procedimento 2: Indicado para quando as componentes x_j do vetor de entrada assumem valores positivos e negativos, e a função de ativação, $\phi(u)$, é a função tangente hiperbólica. Neste caso, aplicar a seguinte transformação a cada componente de **x**:

$$x_j^* = 2\left(\frac{x_j - x_j^{min}}{x_j^{max} - x_j^{min}}\right) - 1 \tag{4.39}$$

em que $x_j^{min} = \min_{\forall t}\{x_j(n)\}$ é o menor valor de x_j. Neste caso, tem-se que $x_j^* \in [-1,+1]$.

Os dois procedimentos descritos acima também devem ser igualmente aplicados às componentes d_k dos vetores de saída, **d**, caso estes possuam amplitudes fora da faixa definida pelas funções de ativação.

Função Tangente Hiperbólica: Tem sido demonstrado empiricamente, ou seja, através de simulação computacional que o processo de treinamento converge mais rápido quando se utiliza a função de ativação tangente hiperbólica do que quando se usa a função logística. A justificativa para isto está no fato da tangente hiperbólica ser uma função ímpar, ou seja, $\phi(-u_i) = -\phi(u_i)$. Daí sugere-se utilizar a função tangente hiperbólica sempre que o problema permitir.

Classificação de padrões: Quando se treina a rede ELM para classificar padrões é comum usar a codificação de saída descrita na Seção 4.3.6, em que na especificação do vetor de saídas desejadas assume-se o valor de saída unitário (1) para o neurônio que representa a classe e nulo (0) para os outros neurônios. Conforme dito no item anterior este valores são assintóticos e portanto, dificilmente serão observados durante a fase de teste.

Assim para evitar ambiguidades durante o cálculo da taxa de acerto P_{acerto} durante as fases de treinamento e teste define-se como a classe do vetor de entrada atual, $\mathbf{x}(n)$, como sendo a classe representada pelo neurônio que tiver maior valor de saída. Em palavras, podemos afirmar que se o índice do neurônio de maior saída é c, ou seja

$$y_c(n) = \max_{\forall k}\{y_k(n)\} \tag{4.40}$$

então a Classe de $\mathbf{x}(n)$ é a Classe c.

Os dois primeiros classificadores randomizados (ou seja, RVFL e ELM) originaram-se no campo da rede neural. Já o classificador RKS (RAHIMI; RECHT, 2008b), por sua vez, tem origem no campo das máquinas de kernel.

4.4 Random Kitchen Sinks

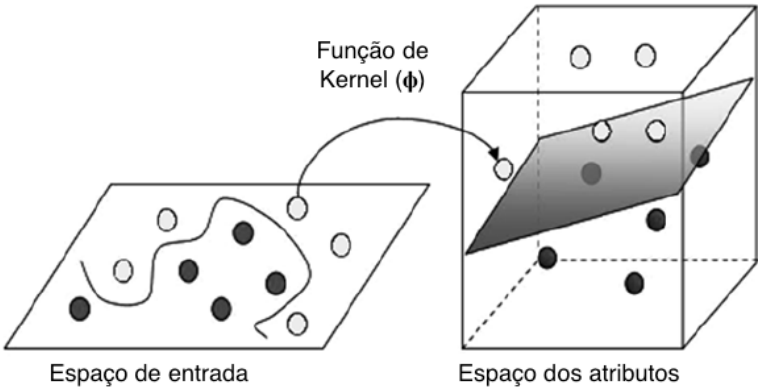
4.4.1 Definição

Os métodos de kernel fornecem uma abordagem elegante, teoricamente bem fundamentada para tratar problemas de aprendizagem. Algoritmos tradicionais requerem o cálculo de uma matriz de kernel de dimensão $N \times N$ para solucionar problemas de aprendizagem para N vetores de entrada. No entanto, a aplicação desses métodos para conjuntos de dados em grande escala contendo milhares de observações tem provado ser um desafio. Algoritmos como o RKS surgem como uma solução alternativa para minimizar tal dificuldadde, ao formalizar a tarefa de aprendizagem no espaço primal em vez do dual.

O método kernel contém uma denominada função kernel. Essa função mapeia o espaço de entrada separável não linear em um espaço de característica separável linear de dimensional maior.

Na Fig. 15, temos um espaço de recursos bidimensional, que não é linear. Com a função kernel, podemos mapear linearmento o espaço de entrada em um espaço de recursos tridimensional.

Figura 15 – Representação da transformação de kernel.



Fonte: (RAHIMI; RECHT, 2009).

Rahimi & Recht 2008a, 2008b desencadearam interesses em uma abordagem que aproxima um kernel invariante ao deslocamento $k(\mathscr{X} \times \mathscr{X}) \to \mathbb{R}$ por

$$k(\mathbf{x},\mathbf{y}) = \langle \phi(x), \phi(y) \rangle \approx z(\mathbf{x})^T z(\mathbf{y}), \tag{4.41}$$

onde $z : \mathscr{X} \to \mathbb{R}^D$. O método primal em $\mathbb{R}^D$ pode ser usado, permitindo a resolução da maioria dos problemas de interesse em tarefas de aprendizagem de máquinas (SUTHERLAND; SCHNEIDER, 2015).

Onde a saída da função de kernel, tem sua forma

$$f(\mathbf{x}) = \sum_{i=1}^{\infty} \alpha_i k(x; x_i) \tag{4.42}$$

o que equivale a uma decisão de uma superfície linear

$$f(\mathbf{x}) = \langle w_i, \phi(x) \rangle \tag{4.43}$$

Partindo do modelo de solução já utilizado por outros classificadores, mas com uma diferente abordagem de treinamento, temos a função de saída representada por uma combinação linear de infinitos kernels:

$$f(\mathbf{x}) = \sum_{i=1}^{\infty} \alpha_i \phi(\mathbf{x}; w_i) \tag{4.44}$$

onde ϕ é uma função não-linear, α_i e w_i são os parâmetros a serem calculados para encontrar a solução. Assumindo apenas T funções, obtemos a seguinte representação aproximada:

$$f(\overrightarrow{x}) \approx f_T(\mathbf{x}) = \sum_{i=1}^{T} \alpha_i \phi(\mathbf{x}; w_i) \tag{4.45}$$

Dado que

$$w_i^*, ..., w_T^* \sim p(\mathbf{w}) \tag{4.46}$$

ou seja, os parâmetros das T funções não lineares são amostradas de uma função de densidade de probabilidades $p(.)$

Pode-se encontrar o vetor de pesos α, resolvendo o seguinte problema de otimização:

$$\alpha^* = \min_{\alpha} \left\| \sum_{i=1}^{T} \alpha_i \phi(x; w_i) - f \right\|_{\mu} \tag{4.47}$$

A partir da solução do problema de otimização para calcular a variável α do RKS (RAHIMI; RECHT, 2009)

$$\alpha = \phi(x)^T \phi(x)^{-1} \phi(x)^T y \tag{4.48}$$

assim a partir do truque de kernel, é possível solucionar a equação a partir de métodos lineares como o método dos mínimos quadrados aplicado no algoritmo 5.

Assim, para cada vetor de entrada $\mathbf{x}_n$, calculamos o mapeamento de atributos aproximado como

$$y^* = \alpha^T \phi(x^*) \tag{4.49}$$

onde o asterisco $(*)$ representa o novo dado do conjunto de teste.

4.4.2 Implementação do Algoritmo

A seguir é descrito um pseudocódigo para implementação em Matlab do algoritmo, destacando sua rapidez de treinamento na ordem de $N \times T \times D$ e o teste de $T \times D$, onde D é a dimensão das funções de não lineares, T é o número de características e N é o número de amostras.

Todo o pseudo código é encontrado descrito em (RAHIMI, 2016)

Algoritmo 5: Algoritmo do classificador RKS

/* Função de Treinamento */

Entrada: $entrada, saida, dimensao, ruido$

Resultado: [alpha,w]

1 $w \leftarrow randn(dimensao, size(entrada, 1))$; /* Amostragem dos atributos */

2 $Z \leftarrow exp(i * w * entrada)$; /* Calcula a Matriz de atributos */

3 $alpha \leftarrow (eye(size(entrada, 1)) * ruido + Z * Z') \backslash (Z * saida)$

retorna *[alpha,w]*

/* Função de Teste */

Entrada: $entrada\_teste, saida\_teste, alpha, w$

Resultado: saida_teste

4 $saida\_teste \leftarrow alpha' * exp(i * w * entrada\_teste)$

retorna *saida_teste*

4.5 Classificadores adicionais

Por uma questão de completude, incluímos as descrições dos outros classificadores nos apêndices A, B e C. Sendo estes o classificador linear de mínimos quadrados e também dois classificadores não lineares supervisionados não randomizados: a rede de perceptron multicamada e a Máquina de vetor de suporte de Vapnik.

4.6 Resumo do capítulo

Neste capítulo, foram especificadas as hipóteses e etapas experimentais dos classificadores, entradas e saídas dos mesmos, assim como a notação matemática utilizada no problema de classificação. Na sequência foram apresentados todos os classificadores utilizados para detecção de crise epilética, iniciando com os randomizados RVFL, ELM e RKS neste capítulo e os terminando com os não randomizados MQ, MLP e SVM nos apêndices A, B e C.

5 METODOLOGIA DE TREINAMENTO E AVALIAÇÃO

Neste capítulo será descrito todos os procedimentos realizados para a obtenção dos resultados, envolvendo toda a preparação do banco de dados, as configurações, estimação dos hiperparâmetros e técnicas aplicadas sobre os classificadores a serem avaliados nesta dissertação (MQ, RVFL, RKS, ELM, MLP e SVM). Assim como as duas técnicas de extração de características já apresentadas no capítulo 3.

5.1 Fundamentação

Uma das características do banco de dados utilizado é que o intervalo de crise epilética é sempre bem menor do que toda a aquisição e irregular de acordo com cada aquisição em cada paciente. Para efeito de exemplificação, cada amostra tem duração de 1 hora como já mencionado e a crise epilética dura em torno de 1 minuto, nesse caso tomamos uma amostra de 2 minutos dividida igualmente com e sem crise epilética. Assim, para lidar com as categorias desbalanceadas, nós equalizamos propositadamente a proporção de casos positivos a negativos por paciente.

Quanto às amostras do banco de dados, dividimos aleatoriamente as instâncias disponíveis por paciente em 3 subgrupos: treinamento (70 %), validação (20 %) e teste (10 %). O conjunto de treinamento será utilizado para a etapa de aprendizado do classificador, o de validação será utilizado para verificar a eficiência da rede quanto a sua capacidade de generalização durante o treinamento e o conjunto de teste será utilizado para verificar a performance de cada classificador para novos dados.

A tarefa de classificação de crise epilética requer uma análise cuidadosa sobre o problema para minimizar ambiguidades e erros nos dados. Além disso, os dados coletados devem ser significativos e cobrir amplamente o domínio do problema; não devem cobrir apenas as operações normais ou rotineiras, mas também as exceções e as condições nos limites do domínio do problema.

Depois de determinados estes conjuntos, eles são, colocados em ordem aleatória para minimizar viéses associados à ordem de apresentação dos dados. Além disso, é necessário um pré-processamento destes dados, através de normalizações, escalonamentos e conversões de formato para torná-los mais apropriados à sua utilização nos classificadores.

A estimação de cada hiperparâmetro, a análise de seu melhor para valor e as boas

técnicas de construção de classificadores são descritas no capítulo de classificadores e no apêndice A, B e C. Já as configurações relativas as técnicas de extração de atributos são apresentadas no capítulo 3.

Um próximo passo é a definição da configuração da rede, que pode ser dividido em duas etapas genericamente para todos os classificadores:

1) Determinação da topologia a ser utilizada. Ou seja, a escolha dos principais hiperparâmetros conforme apresentados adiante.

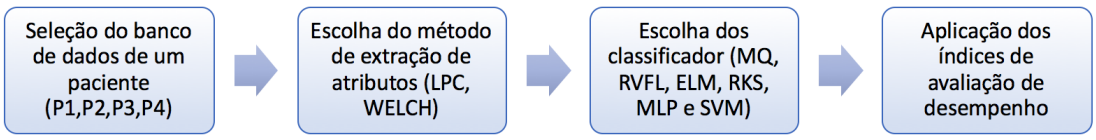
2) Determinação de parâmetros do algoritmo de treinamento, funções de ativação e outras funções específicas para cada classificador.

O passo seguinte é o teste da rede. Durante esta fase o conjunto de teste é utilizado para determinar o desempenho da rede com dados que não foram previamente utilizados. O desempenho do classificador, medida nesta fase, é uma boa indicação de seu desempenho real.

Por último, são gerados os relatórios de resultados e as matrizes de confusão de cada classificador e elaborado os mais diversos gráficos que serão mostrados no próximo capítulo baseados nos índices a serem apresentados adiante.

As etapas e relações são resumidas no gráfico ilustrativo da figura 16.

Figura 16 – Etapas do experimento



Fonte: Construído pelo autor.

5.1.1 Hiperparâmetros

Para cada classificador, há diversos parâmetros e configurações específicas definidos que constam no capítulo 4 na secção de cada classificador. Dentre esses parâmetros, destacam-se os hiperparâmetros que são os parâmetros mais importantes que definirão características cruciais dos modelos, resultando diretamente na performance de cada classificador. Nesta secção, apresentaremos os seguintes hiperparâmetros, a saber:

- q - número de neurônios ocultos;
- η - taxa de aprendizagem;
- α - fator de momento;
- γ - parâmetro de escalonamento do kernel gaussiano no SVM e RKS;
- D - dimensão do mapeamento do RKS;
- C - parâmetro de regularização do SVM.

Os hiperparâmetros dos classificadores estão listados na Tabela 2. Esses valores foram escolhidos após 100 execuções independentes de validação cruzada 5 folds, nos conjuntos de treinamento e validação. Os resultados numéricos apresentados nas tabelas a seguir são de quatro banco de dados utilizados (representados por P1, P2, P3 e P4), cujos resultados numéricos em conjuntos de teste são típicos entre os coletados em todo o conjunto de indivíduos.

Tabela 2 – Hiperparâmetros dos classificadores avaliados.

Classificadores	**Hiperparâmetros selecionados**		
MLP	$q = 250$	$\eta = 0,05$	$\alpha = 0,75$
RKS	$D = 300$	$\gamma = 0,005$	-
ELM	$q = 280$	-	-
SVM	$C = 1000$	$\gamma = 0,005$	-
RVFL	$q = 250$	-	-

Fonte: Elaborada pelo autor.

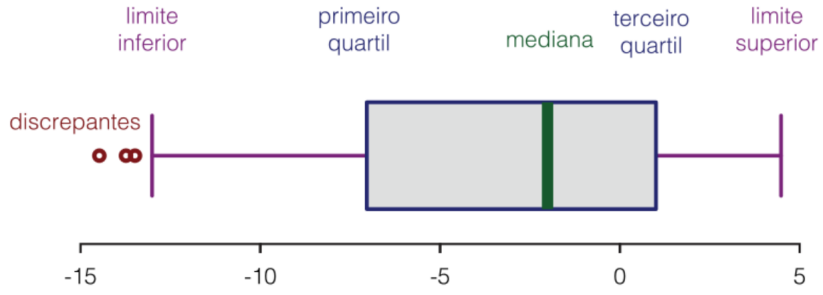
5.2 Índices de avaliação de desempenho

Abaixo serão definidos todos os conceitos e índices utilizados na comparação dos resultados dos algoritmos. Para efeitos de tempo de treinamento e execução, os algoritmos foram executados no software matlab em um computador com processador intel i7 e 8gb de ram, entretanto como sabemos que essa velocidade do classificador depende de algumas outras variável, mais do que o valor absoluto dos índices de performance temporal de cada classificador, desejamos ver a relação entre eles.

Boxplot: Utilizados na apresentação dos resultados desse trabalho, o Boxplot ou diagrama de caixa é uma ferramenta gráfica para representar a variação de dados observados de uma variável numérica.

É formado pelo primeiro (Q_1), terceiro quartil(Q_3) e pela mediana(Q_2). As hastes inferiores e superiores se estendem, respectivamente, do quartil inferior até o menor valor não inferior ao limite inferior e do quartil superior até o maior valor não superior ao limite superior. Os valores atípicos ou outliers (valores discrepantes) podem ser plotados como pontos individuais. O boxplot não é paramétrico, apresentando a variação em amostras de uma população estatística sem fazer qualquer suposição da distribuição estatística subjacente. Os espaços entre as diferentes partes da caixa indicam o grau de dispersão, a obliquidade nos dados e os outliers.

Figura 17 – Estrutura do boxplot



O limite inferior é calculo a seguir:

$$maxmin(dados), Q_1 - 1,5(Q_3 - Q_1) \tag{5.1}$$

Já o superior de maneira análoga é calculo a seguir:

$$maxmin(dados), Q_3 + 1,5(Q_3 - Q_1) \tag{5.2}$$

Matriz de confusão: Utilizada em todos os classificadores para então calcular os indicadores analisados, a matriz de classificação ou confusão é uma ferramenta padrão para avaliação de modelos estatísticos. Ele tem como vantagem facilitar a visualização da performance do classificador a partir da divisão em uma matriz 2×2 de 4 resultados: a quantidade de F_P, F_N, V_P, e V_N.

Validação cruzada: É uma técnica para avaliar a capacidade de generalização de um modelo, a partir de um conjunto de dados. Esta técnica é amplamente utilizada para estimar o quão preciso é o modelo na prática, ou seja, o seu desempenho para um novo conjunto de dados. O conceito central das técnicas de validação cruzada é o particionamento do conjunto de dados em subconjuntos mutualmente exclusivos, e posteriormente, utiliza-se alguns destes subconjuntos para a estimação dos parâmetros do modelo (dados de treinamento) e

o restante dos subconjuntos (dados de validação ou de teste) são empregados na validação do modelo.

Assim, considerando as seguintes variáveis, apresentamos os índices de avaliação de desempenho dos classificadores.

- T_A - total de acertos;
- T_D - total de dados;
- V_P - número de predições verdadeiro positivas;
- V_N - número de predições verdadeiras negativas;
- F_P - número predições falso positivos;
- F_N - número de diagnósticos falso negativo;
- P - número de diagnósticos positivos;
- N - número de diagnósticos negativos;
- A_P - total de acertos positivos;
- A_N - total de acertos negativos;
- T_P - total de positivos;
- T_N - total de diagnósticos negativos.

5.2.1 Acurácia (AC) ou precisão

A Acurácia (do inglês accuracy, também conhecida como repetibilidade da avaliação) de varias amostras em classificação é uma medida da proporção de predições corretas, sem levar em consideração o que é positivo e o que é negativo. Esta medida é altamente suscetível a desbalanceamentos do conjunto de dados e pode facilmente induzir a uma conclusão errada sobre o desempenho do sistema:

$$AC = \frac{T_A}{T_D} = \frac{V_P + V_N}{P + N} \tag{5.3}$$

5.2.2 Sensibilidade(SB)

a proporção de verdadeiros positivos, ou seja, a capacidade do sistema em predizer corretamente a condição para casos que realmente a têm:

$$SB = \frac{A_P}{T_P} = \frac{V_P}{V_P + F_N} \tag{5.4}$$

5.2.3 Especificidade (EP)

A proporção de verdadeiros negativos, ou seja, a capacidade do sistema em predizer corretamente a ausência da condição para casos que realmente não a têm:

$$EP = \frac{A_N}{T_N} = \frac{V_N}{V_N + F_P} \tag{5.5}$$

5.2.4 Eficiência (EF)

A média aritmética da Sensibilidade e Especificidade. Na prática, a sensibilidade e a especificidade variam em direções opostas. Isto é, geralmente, quando um método é muito sensível a positivos, tende a gerar muitos falso-positivos, e vice-versa. Assim, um método de decisão perfeito (100

$$EF = \frac{SB + EP}{2} \tag{5.6}$$

5.2.5 Valor preditivo positivo (VPP) ou Valor preditivo negativo (VPN)

VPP é a proporção de verdadeiros positivos em relação a todas as predições positivas, já o VPN é a proporção de verdadeiros negativos em relação a todas as predições negativas. Estas medidas são altamente suscetíveis a desbalanceamentos do conjunto de dados e pode facilmente induzir a uma conclusão errada sobre o desempenho do sistema.

$$VPP = \frac{V_P}{V_P + F_P} \qquad VPN = \frac{V_N}{V_N + F_N} \tag{5.7}$$

5.2.6 Coeficiente de Correlação de Matthews (MCC)

É uma medida de qualidade de duas classificações binárias que pode ser usada mesmo se as classes possuem tamanhos bastante diferentes. Retorna um valor entre (-1) e (+1), em que um coeficiente de (+1) representa uma predição perfeita, (0) representa uma predição aleatória media, e (-1) uma predição inversa. Esta estatística é equivalente ao coeficiente ϕ, e tenta, assim como a eficiência, resumir a qualidade da tabela de contingência em um único valor numérico passível de ser comparado.

$$\phi = \frac{(V_P * V_N - F_P * F_N)}{\sqrt{(V_P + F_P) * (V_P + F_N) * (V_N + F_P) * (V_N + F_N)}} \tag{5.8}$$

5.2.7 Tempo de treinamento (TT)

é a medição do tempo em segundos de todo o treinamento do algoritmo.

5.2.8 Tempo de Processamento (TP)

é a medição do tempo em segundos de todo o funcionamento do algoritmo, uma vez que os classificadores já estão treinados.

5.2.9 Curva ROC

Quando desenvolvemos sistemas, métodos ou testes que envolvem a detecção, diagnósticos ou previsão de resultados, é importante validar seus resultados de forma a quantificar seu poder discriminativo e identificar um procedimento ou método como bom ou não para determinada análise. No entanto, devemos levar em conta que a simples quantificação de acertos num grupo de teste não necessariamente reflete o quão eficiente um sistema é, pois essa quantificação dependerá fundamentalmente da qualidade e distribuição dos dados neste grupo de teste.

A curva ROC (Receiver Operating Characteristic) foi desenvolvida por engenheiros elétricos e engenheiros para detecção de sinais para sistemas de radar na década de 50. A análise ROC tem sido utilizada em medicina, radiologia, psicologia e outras áreas por muitas décadas e, mais recentemente, foi introduzida à áreas como aprendizado de máquina e mineração de dados como uma ferramenta útil e poderosa para a avaliação de modelos de classificação (SPACKMAN, 1989; BRADLEY, 1997).

Como o resultado de sistemas de classificação em classes geralmente são contínuos, ou seja, produzem um valor situado dentro de um determinado intervalo contínuo, como [0;1], é necessário definir um ponto de corte, ou um limiar de decisão, para se classificar e contabilizar o número de predições positivas e negativas (como diagnósticos verdadeiros e falsos no caso de ocorrência de uma patologia). Como este limiar pode ser selecionado arbitrariamente, a melhor prática para se comparar o desempenho de diversos sistemas é estudar o efeito de seleção de diversos limiares sobre a saída dos dados.

Para cada ponto de corte são calculados valores de sensibilidade e especificidade, que podem então serem dispostos em um gráfico denominado curva ROC, que apresenta no eixo das ordenadas os valores de sensibilidade ou taxa dos verdadeiros positivos e nas abscissas o complemento da especificidade, ou seja, o valor (1-especificidade) ou taxa dos falsos positivos.

Um classificador perfeito corresponderia a uma linha horizontal no topo do gráfico, porém esta dificilmente será alcançada. Na prática, curvas consideradas boas estarão entre a linha diagonal e a linha perfeita, onde quanto maior a distância da linha diagonal, melhor o sistema. A linha diagonal indica uma classificação aleatória, ou seja, um sistema que aleatoriamente seleciona saídas como positivas ou negativas, como jogar uma moeda para cima e esperar cara ou coroa. Definitivamente, não é o tipo de sistema mais confiável possível. No entanto, um sistema cuja curva ROC esteja localizada abaixo da diagonal ainda pode ser convertido num bom sistema, basta inverter suas saídas e então sua curva também será invertida.

5.3 Resumo do capítulo

Neste capítulo, inicialmente, apresentamos uma ordem sequencial de todos os passos necessários para a obtenção dos resultados apresentados a seguir, desde a preparação do banco de dados até as técnicas de extração de características, como também as funções e hiperparâmetros utilizados nos classificadores.

Em seguida, apresentamos os parâmetros do teste de performance que serão utilizado para a comparação dos resultado adquiridos nos classificadores e cenários que serão apresentados no capítulo seguinte. Assim como a fundamentação teórica de cada um deles.

6 RESULTADOS

Neste capítulo iremos apresentar todos os resultados, construídos com esse trabalho, referentes aos questionamentos, hipóteses e perspectivas inicialmente levantadas. Junto com esses resultados, iremos discutir sobre os principais temas relacionados aos mesmos, de modo que possamos contextualizar e interpretar esses resultados com o intuito de colaborar com outras pesquisas nessa área tão importante para a sociedade brasileira e mundial.

6.1 Apresentação

Ao todo, foram realizadas as execuções deste experimento com 16 amostras de dados de 8 pessoas, resultando em 192 cenários experimentais e 320 imagens a serem analisadas.

A título de ilustração, escolhemos 4 amostras independentes de dados de 3 pacientes diferentes, por representarem o conjunto dos resultados. As 4 amostras independentes estão representadas na tabela 6 no Anexo. Para fins didáticos, essas quatros amostras representadas serão referenciadas no trabalho por P1, P2, P3 e P4 ou por Paciente 1, 2, 3 e 4.

Com os 4 conjunto de dados, definimos 48 cenários com a combinação de método de extração de atributos, classificador e conjunto da dados (representado pela a aquisição de um paciente no banco de dados) com a finalidade de comparação dos resultados, conforme mostrado na tabela 3.

Para cada paciente, foram realizadas 50 iterações independentes para cada um dos 48 cenários. No final de cada fase de teste, curva ROC foi construída e foram calculados os indicadores AC, SB, EP, EF, VPP, VPN, MCC, TT e TP a partir da matriz de confusão correspondente e outras variáveis já definidas.

Separamos os resultados por paciente, por método de extração de características e por indicador, assim apresentamos os resultados de um indicador para um dos pacientes utilizando um dos métodos de extração. Em todos os resultados, estarão presentes os valores para todos os classificadores para fins de comparação.

Além disso, apresentamos na tabela 3 o resumo de todos os 48 experimentos realizados. Com o intuito de didaticamente resumir os principais resultados, devido a complexidade dos experimentos, preparamos 4 tabelas em que indicamos numericamente a média dos principais indicadores para cada paciente, organizada por experimentos.

Abaixo, na Figura 18 apresentamos a acurácia dos resultados dos classificadores

Tabela 3 – Cenários definidos para os experimentos

Paciente	Atributos	Classificador					
		MQ	**MLP**	**RKS**	**RVFL**	**ELM**	**SVM**
1	**Welch**	A	B	C	D	E	F
	LPC	G	H	I	J	K	L
2	**Welch**	M	N	O	P	Q	R
	LPC	S	T	U	V	W	X
3	**Welch**	Y	Z	AA	AB	AC	AD
	LPC	AE	AF	AG	AH	AI	AJ
4	**Welch**	AK	AL	AM	AN	AO	AP
	LPC	AQ	AR	AS	AT	AU	AV

Fonte: Elaborada pelo autor.

para os quatro pacientes com o método de Welch para extração de características. Com isso, podemos observar que os classificadores SVM, MLP e RVFL conseguiram os melhores resultados. Também é possível identificar uma maior variabilidade de resultados no P2, P3 e P4, tendo o P1 um melhor valor médio.

Já na Figura 19, apresentamos o experimento com os mesmo pacientes, porém com a utilização do método LPC para extração de características. Com esse método de extração de características, os classificadores obtiveram um melhor valor médio com menor variabilidade para todos resultados, destacando que os classificadores MLP, SVM e RVFL ainda continuaram com os melhores resultados se comparado aos demais no mesmo experimento. Todos esses resultados serão ratificados adiante com a apresentação da curva ROC.

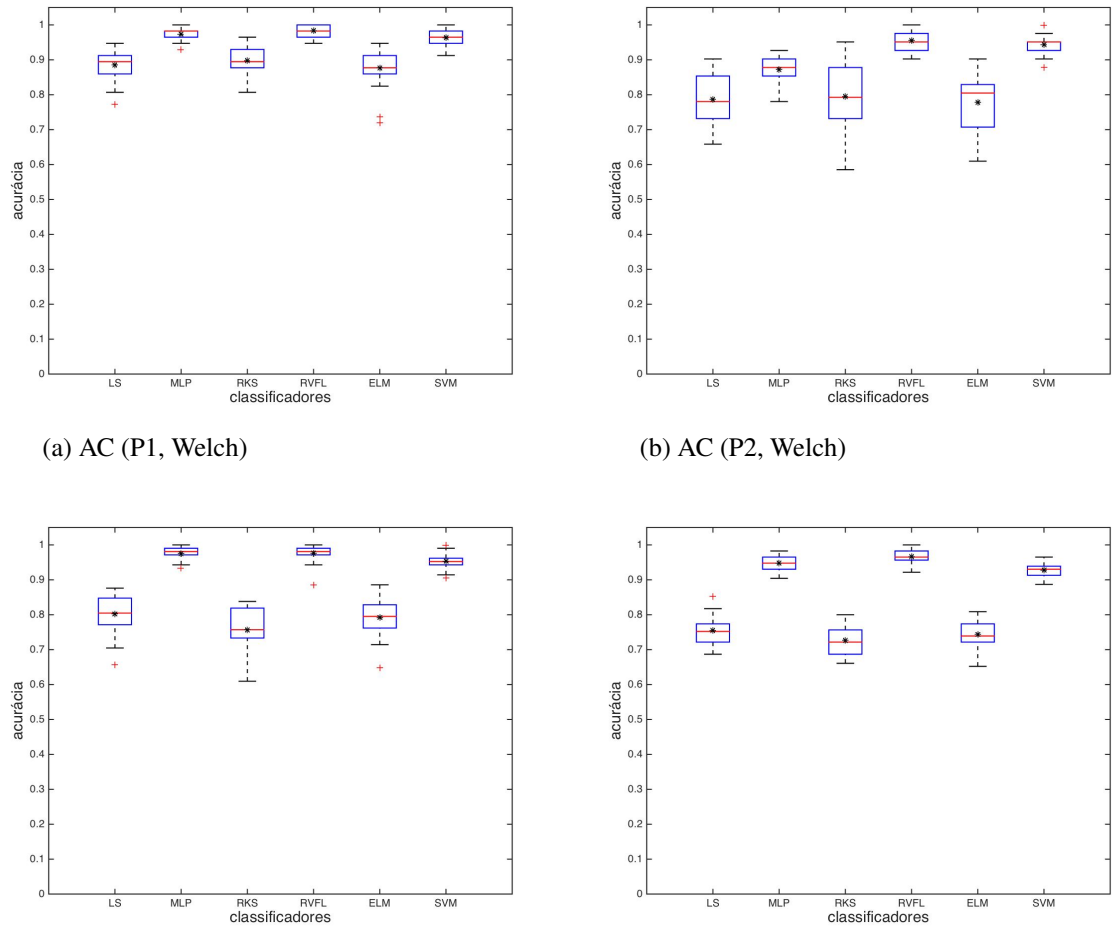
Na Figura 20 apresentamos a sensibilidade dos resultados dos classificadores para os quatro pacientes com o método de Welch para extração de características.

Em um sistema de detecção de crise epilética, é importantíssimo uma sensibilidade elevada, ou seja, o sistema ser capaz de predizer corretamente a condição para casos que realmente tem.

Os resultados para esses experimentos na Figura 20 são semelhantes aos da acurácia desse mesmo experimento. Ou seja, os classificadores MLP, SVM e RVFL se destacaram com menor variância e melhor média dos resultados. Além disso, os classificadores utilizando os dados do paciente P1 obtiveram os melhores resultados.

Já nos cenários da Figura 21, utilizando o método LPC de extração de características, o mesmo padrão de resultado foi observado tanto para os classificadores como para os resultados por paciente, destacando que os resultados obtidos pelos classificadores para o paciente P1 foram ligeiramente inferior em relação ao dos outros pacientes, mas com resultados superiores aos da

Figura 18 – AC para todos os classificadores usando o método de extração de características Welch para os quatro pacientes.



(a) AC (P1, Welch)

(b) AC (P2, Welch)

(c) AC (P3, Welch)

(d) AC (P4, Welch)

Fonte: Elaborada pelo autor.

Figura 20.

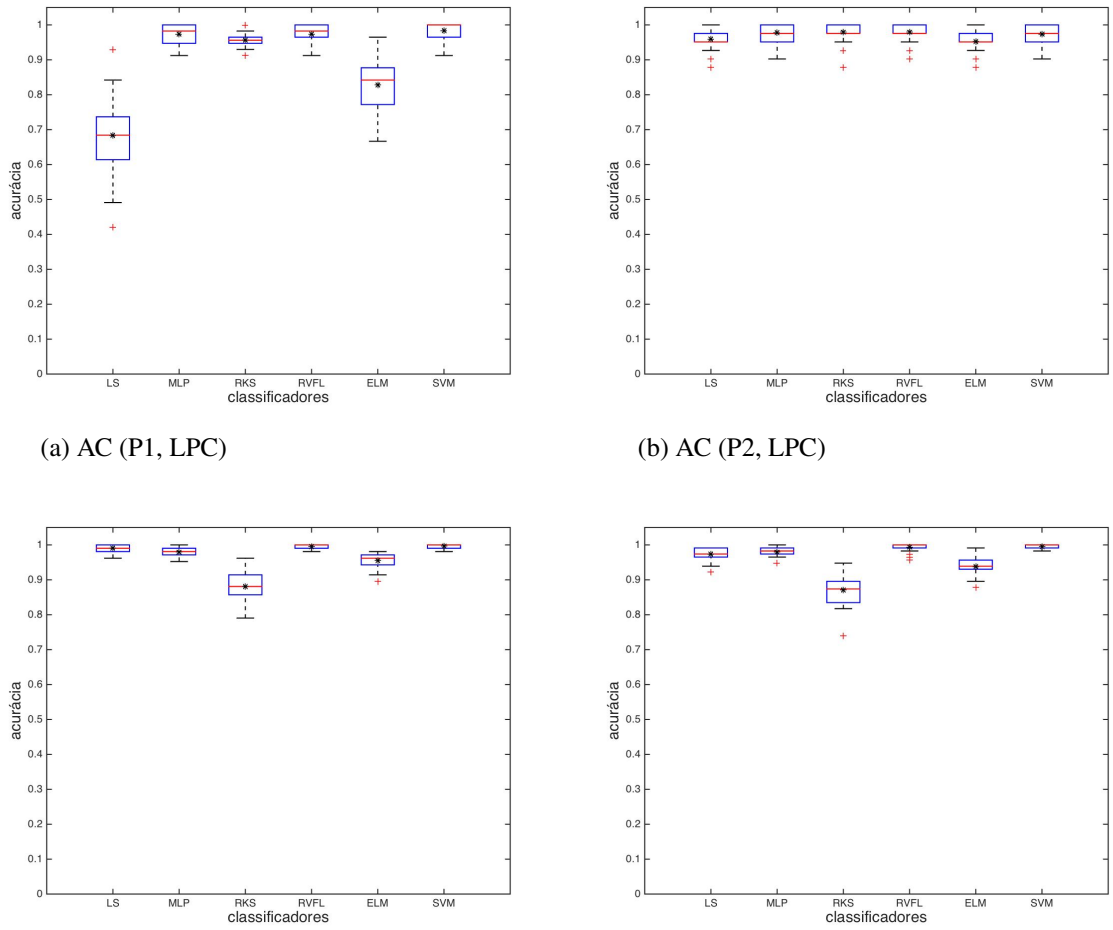
Nas Figuras 22 e 23 apresentamos a especificidade dos resultados dos classificadores para os quatro pacientes com os métodos de Welch e LPC usados para extração de características.

Uma especificidade elevada é de fundamental importância, assim como uma elevada sensibilidade, para esse sistema de detecção predizer corretamente quando um paciente não tem a crise epilética.

Podemos observar na Figura 22 que as especificidade tem uma maior variância, tanto dentro dos resultados de um mesmo classificador, como dentre os resultados médios dos classificadores para todos os pacientes. Já quando é utilizado o método LPC de acordo com a Figura 23, obtivemos uma menor variância em todos os casos.

Nas Figuras 24 e 25 apresentamos a eficiência dos resultados dos classificadores para

Figura 19 – AC para todos os classificadores usando o método de extração de características LPC para os quatro pacientes.



(a) AC (P1, LPC)

(b) AC (P2, LPC)

(c) AC (P3, LPC)

(d) AC (P4, LPC)

Fonte: Elaborada pelo autor.

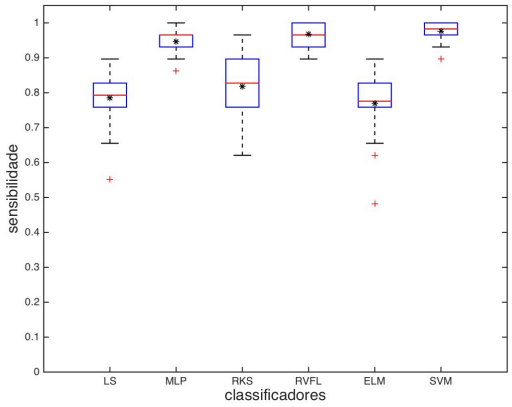
os quatro pacientes com os métodos de Welch e LPC usados para extração de características.

Uma elevada eficiência é obtida a partir de um equilíbrio entre altos valores de sensibilidade e especificidade. Assim, dependendo da aplicação, pode ser mais coerente escolher a utilização de um sistema pelo EF e não pela SB ou EP individualmente.

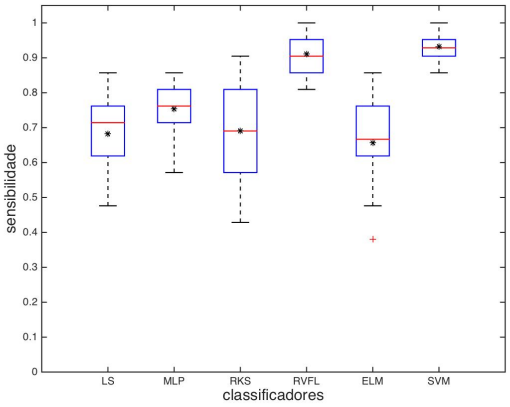
Tanto na Figura 24 como na Figura 25, é possível identificar que, com algumas pequenas diferenças, os indicadores continuam seguindo um padrão, apresentando sempre os mesmos melhores classificadores para os pacientes, assim como pacientes que são melhores independentes dos classificadores.

Nas Figuras 26 e 27 apresentamos a valor preditivo positivo dos resultados dos classificadores para os quatro pacientes com os métodos de Welch e LPC usados para extração de características.

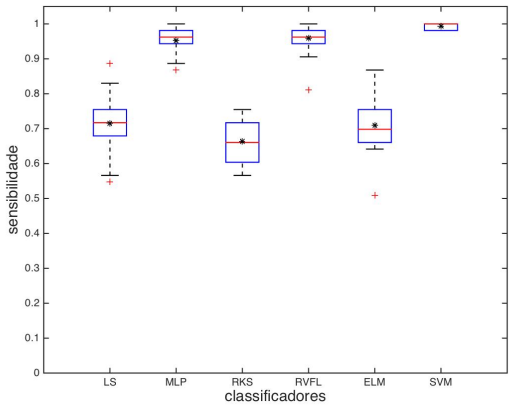
Figura 20 – SB para todos os classificadores usando o método de extração de características Welch para os quatro pacientes.



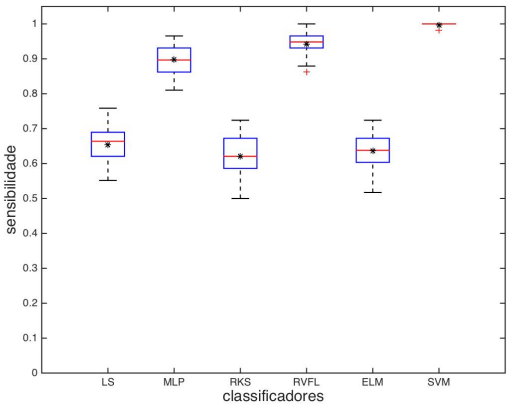
(a) SB (P1, Welch)



(b) SB (P2, Welch)



(c) SB (P3, Welch)



(d) SB (P4, Welch)

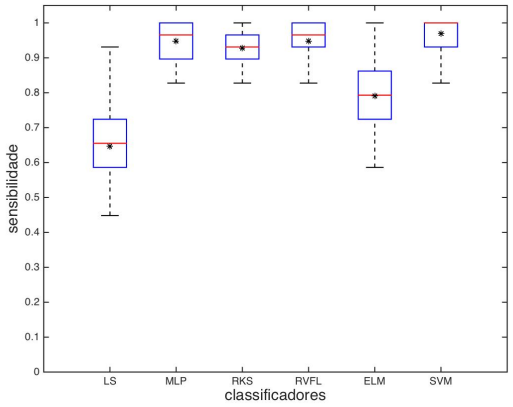
Fonte: Elaborada pelo autor.

Tanto na Figura 26 como na Figura 27, é possível identificar que o VPP apresenta uma menor variância e maior valor médio para os melhores (com maior valores de acurácia) classificadores, ratificando uma boa capacidade do sistema em identificar crises epiléticas quando elas realmente ocorrem.

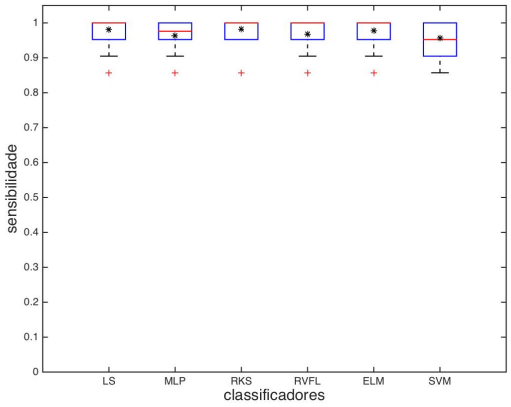
Nas Figuras 28 e 29 apresentamos a valor preditivo negativo dos resultados dos classificadores para os quatro pacientes com os métodos de Welch e LPC usados para extração de características.

Tanto na Figura 28 como na Figura 29, é possível identificar que o VPN apresenta uma menor variância e maior valor médio para os melhores (com maior valores de acurácia) classificadores, ratificando uma boa capacidade do sistema em identificar a não ocorrência de uma crise epilética quando ela realmente não ocorre.

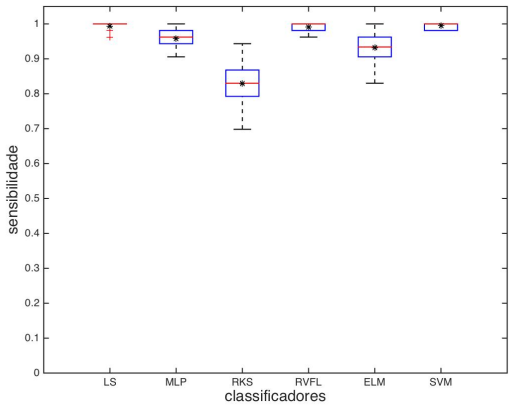
Figura 21 – SB para todos os classificadores usando o método de extração de características LPC para os quatro pacientes.



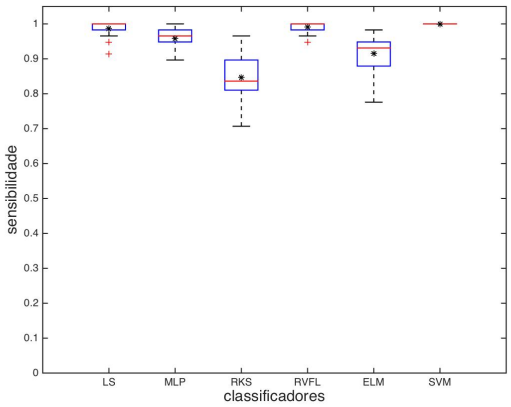
(a) SB (P1, LPC)



(b) SB (P2, LPC)



(c) SB (P3, LPC)



(d) SB (P4, LPC)

Fonte: Elaborada pelo autor.

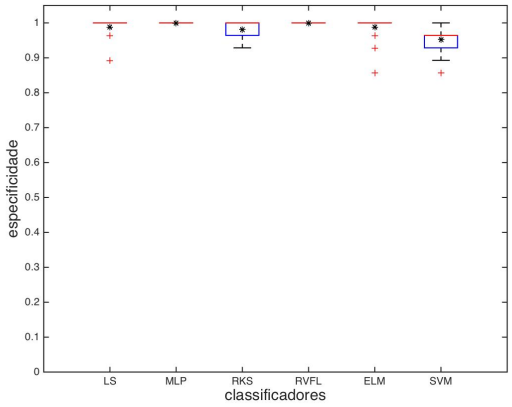
Nas Figuras 30 e 31 apresentamos o coeficiente de correlação de Matthews dos resultados dos classificadores para os quatro pacientes com os métodos de Welch e LPC usados para extração de características. Este indicador é de fundamental importância para um sistema, pois com ele é possível resumir a qualidade do sistema de tamanho variável em um único valor numérico passível de ser comparado.

Na Figura 30, ratificando os resultandos dos indicadores anteriores, nota-se uma vantagem na utilização dos classificadores MLP, SVM e RVFL, seguido pelos demais.

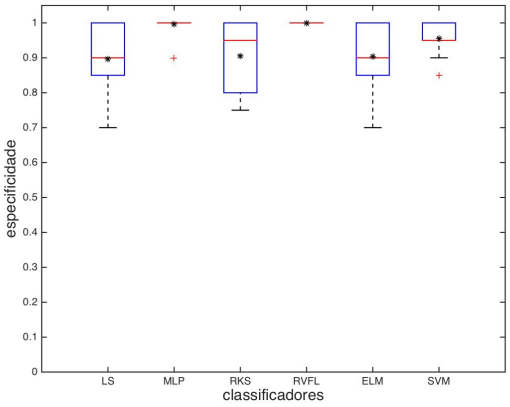
Já Figura 31, a variância dos classificadores foi menor e com um maior valor médio, representando uma vantagem em se utilizar o método LPC de extração de características.

Separamos todos os classificadores em uma etapa de aprendizagem e de teste de desempenho. Por mais que alguns classificadores como o MQ não tenha essa etapa de treinamento

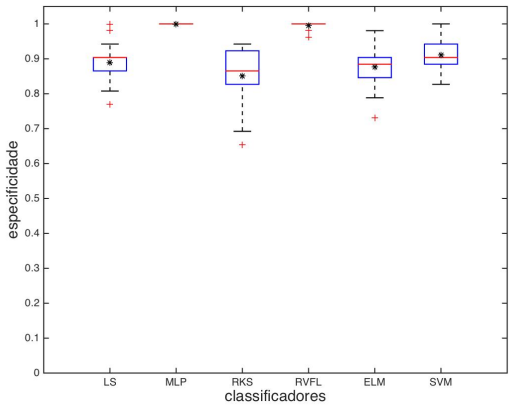
Figura 22 – EP para todos os classificadores usando o método de extração de características Welch para os quatro pacientes.



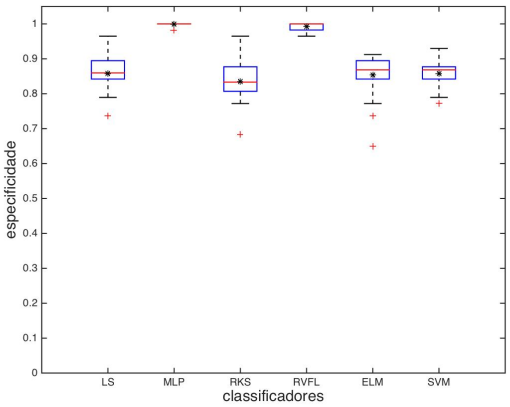
(a) EP (P1, Welch)



(b) EP (P2, Welch)



(c) EP (P3, Welch)



(d) EP (P4, Welch)

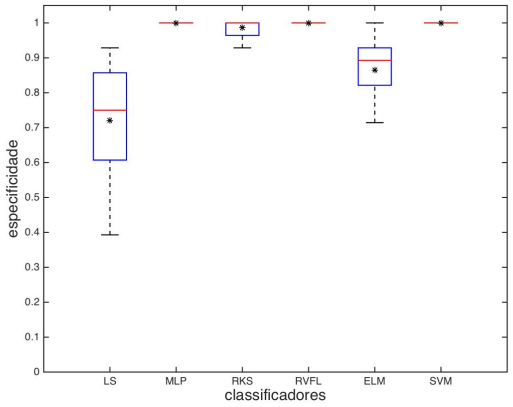
Fonte: Elaborada pelo autor.

ou aprendizagem da mesma forma da MLP por exemplo, será evidenciado a partir do menor valor desse indicador abaixo denominado tempo de treinamento.

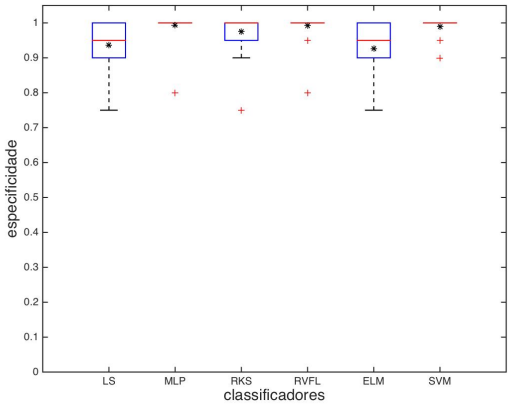
Como sabemos que esses valores são funções de diversas variáveis desconhecidas como características do processador ou do hardware e software em geral, com esse indicador, objetivamos conhecer as relações ou proporções entre os classificadores e não o valor absoluto especificamente.

Nas Figuras 32 e 33, tivemos uma mesma relação com maior tempo para o MLP, seguido pelo SVM e então pelos classificadores randomizados e por último pelo MQ. A principal diferença temporal entre os métodos de extração de características é que cada método gerará um vetor de atributo de tamanho diferente, consequentemente impactará diretamente no cálculo de tempo de execução do algoritmo.

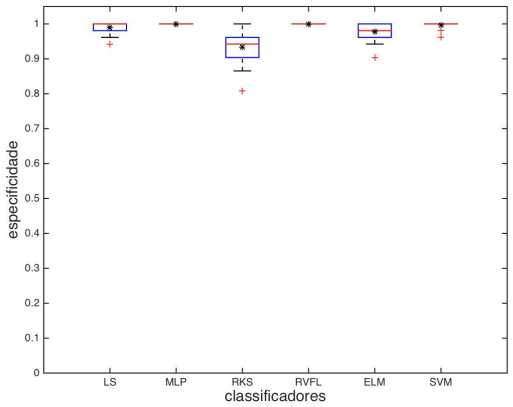
Figura 23 – EP para todos os classificadores usando o método de extração de características LPC para os quatro pacientes.



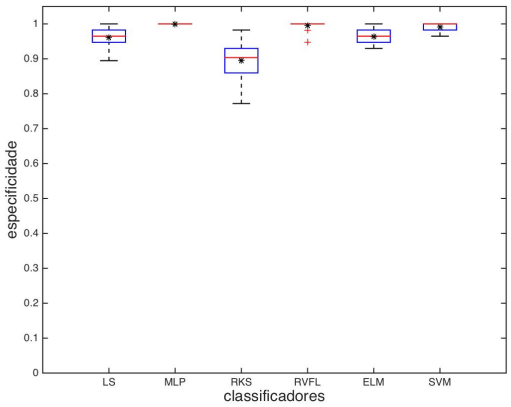
(a) EP (P1, LPC)



(b) EP (P2, LPC)



(c) EP (P3, LPC)



(d) EP (P4, LPC)

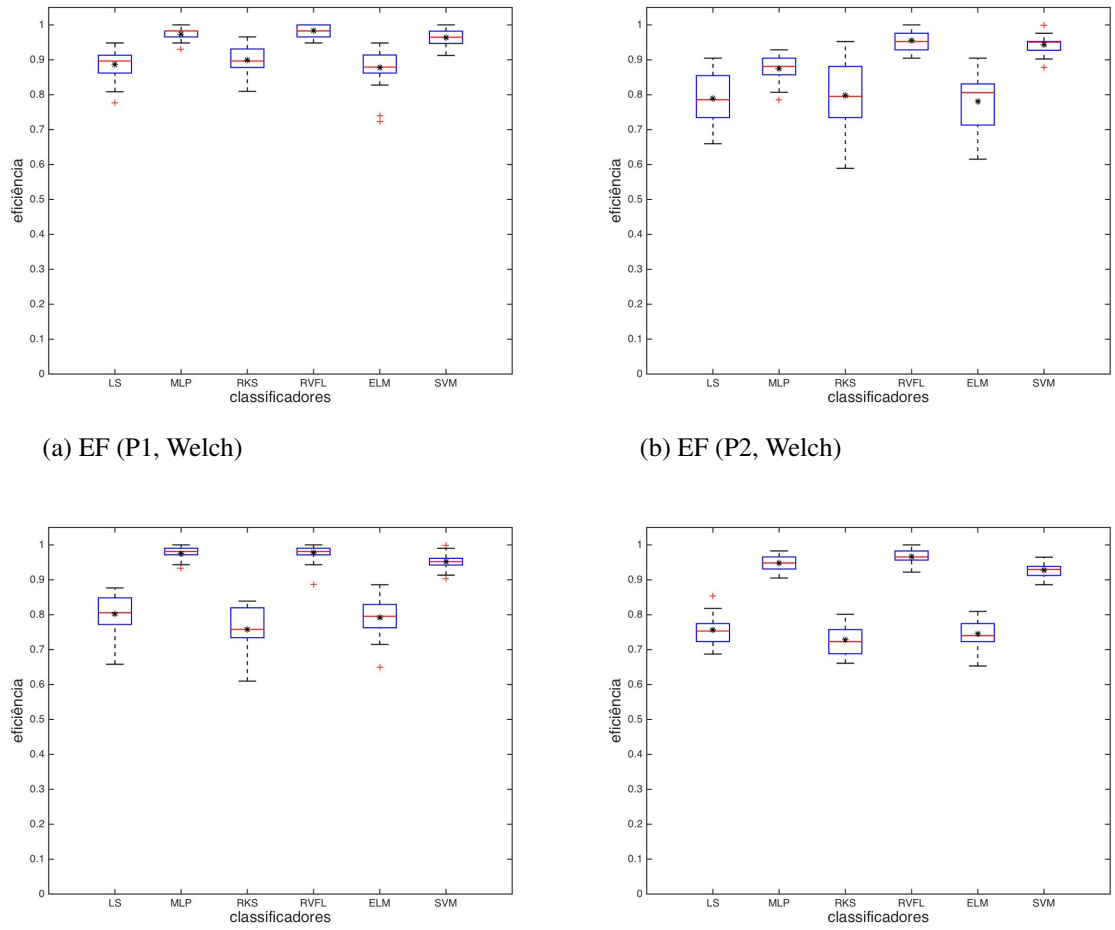
Fonte: Elaborada pelo autor.

Nas Figuras 34 e 35, tivemos como principal diferença em relação às Figuras 32 e 33, um menor tempo, devido este ser a duração de execução do algoritmo quando ele já está treinado. Já a relação temporal teve uma pequena modificação com o SVM em primeiro lugar, a MLP em segundo, seguido pelos classificadores randomizados e por último pelo MQ.

Uma melhor maneira de visualizar as diferenças no desempenho dos vários classificadores para os dois métodos de extração de atributos e para os quatro pacientes é através das curvas ROC correspondentes. Essas curvas são mostradas na Figura 36 para o método do periodograma de Welch e na Figura 37 para o método utilizando LPC.

Indo ao encontro do resultado já reportado nos indicadores anteriores, tanto na Figura 36 como na Figura 37, identificamos uma pior performance para os classificadores MQ, ELM e RKS. Entretanto, estes ainda apresentam um bom resultado. Já quando comparamos os dois

Figura 24 – EF para todos os classificadores usando o método de extração de características Welch para os quatro pacientes.



(a) EF (P1, Welch)

(b) EF (P2, Welch)

(c) EF (P3, Welch)

(d) EF (P4, Welch)

Fonte: Elaborada pelo autor.

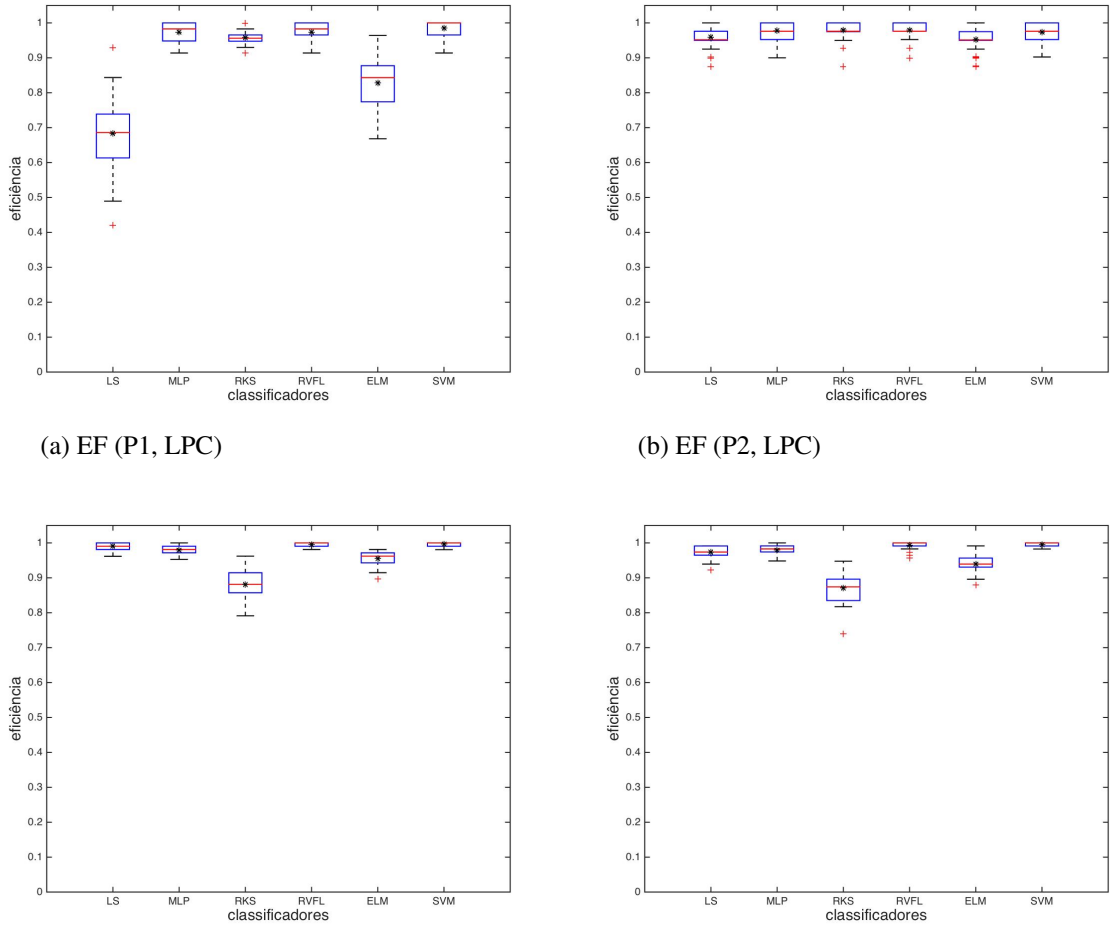
métodos de extração, temos um melhor resultado com a utilização do método LPC para a maioria dos pacientes.

Na Tabela 4, os valores médios das medidas de avaliação para os quatro pacientes nos 48 cenários são resumidamente relatados. Pode-se observar que, para todas as medidas de avaliação, o uso dos coeficientes LPC levou a melhores desempenhos dos classificadores (colunas L para o P1, V para P2, AK para P3 e AX para P4) quando comparados aos resultados alcançados pelo uso do periodograma de Welch.

6.2 Resumo do capítulo

Neste capítulo, iniciamos com uma descrição detalhada dos resultados experimentais e uma discussão sobre a metodologia de obtenção dos indicadores dos resultados. Em seguida,

Figura 25 – EF para todos os classificadores usando o método de extração de características LPC para os quatro pacientes.



(a) EF (P1, LPC)

(b) EF (P2, LPC)

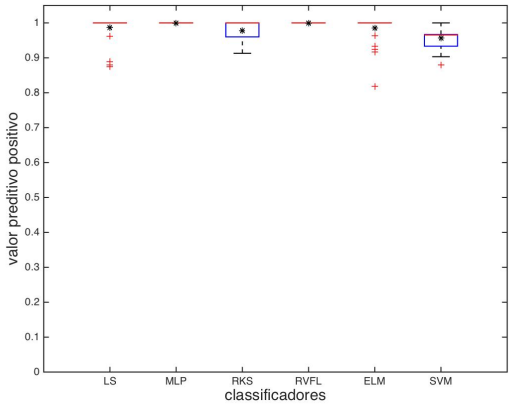
(c) EF (P3, LPC)

(d) EF (P4, LPC)

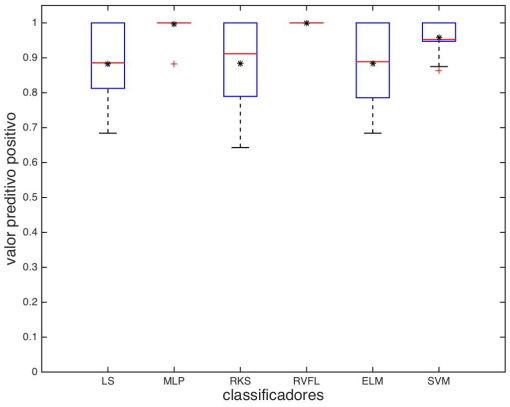
Fonte: Elaborada pelo autor.

apresentamos em uma ordem sequencial os gráficos e discussões sobre os principais resultados individuais. Essa ordem é formada pela sequência de apresentação dos indicadores no capítulo 5, sendo cada indicador apresentado em dois grupos de gráficos, onde o primeiro contém o método de extração de atributos de welch e o segundo o método LPC para os 4 pacientes.

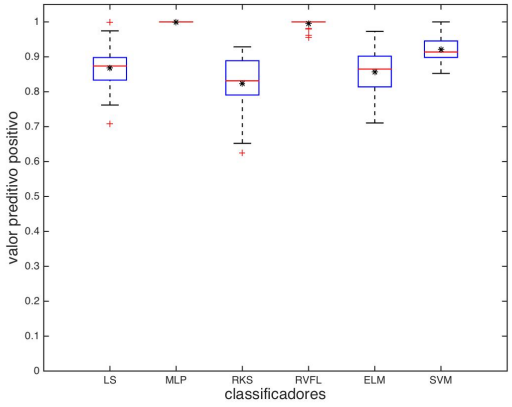
Figura 26 – VPP para todos os classificadores usando o método de extração de características Welch para os quatro pacientes.



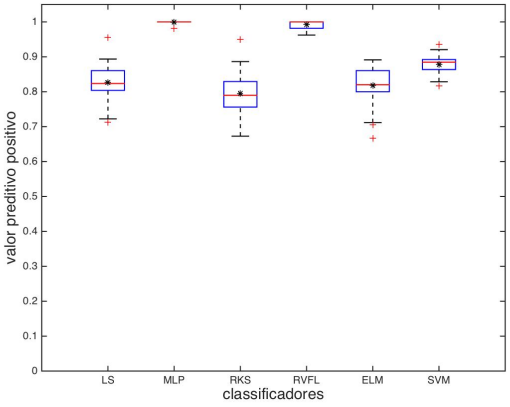
(a) VPP (P1, Welch)



(b) VPP (P2, Welch)



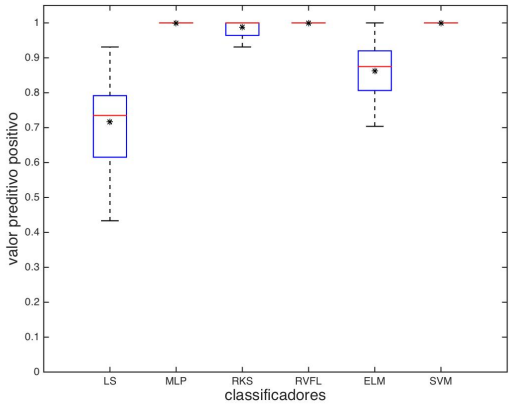
(c) VPP (P3, Welch)



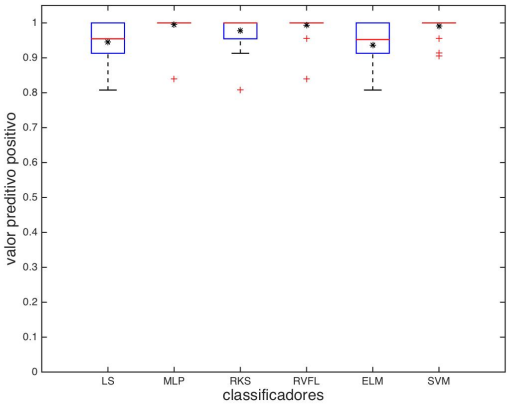
(d) VPP (P4, Welch)

Fonte: Elaborada pelo autor.

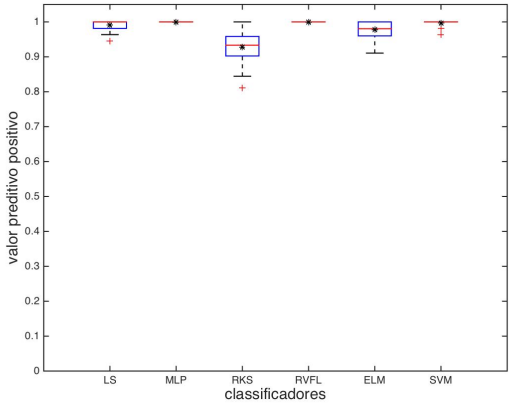
Figura 27 – VPP para todos os classificadores usando o método de extração de características LPC para os quatro pacientes.



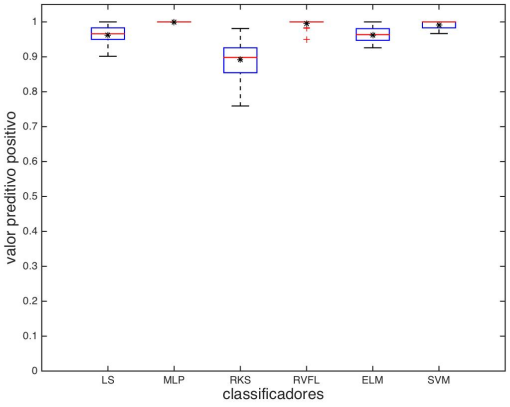
(a) VPP (P1, LPC)



(b) VPP (P2, LPC)



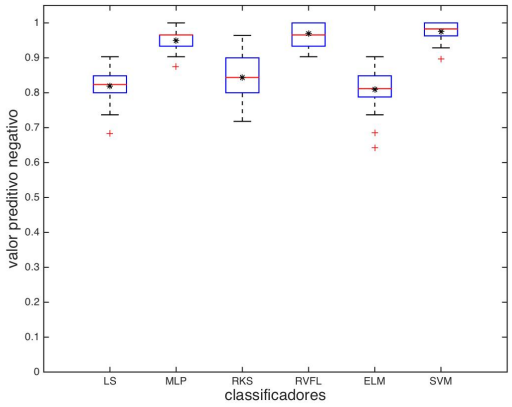
(c) VPP (P3, LPC)



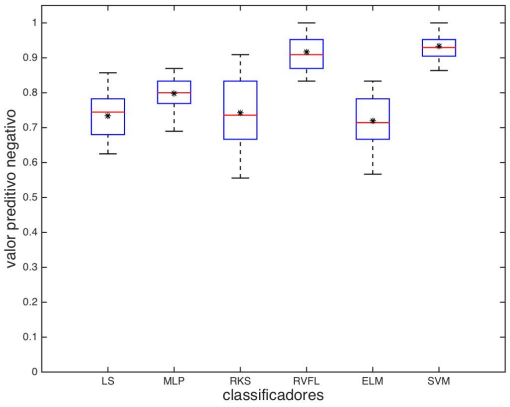
(d) VPP (P4, LPC)

Fonte: Elaborada pelo autor.

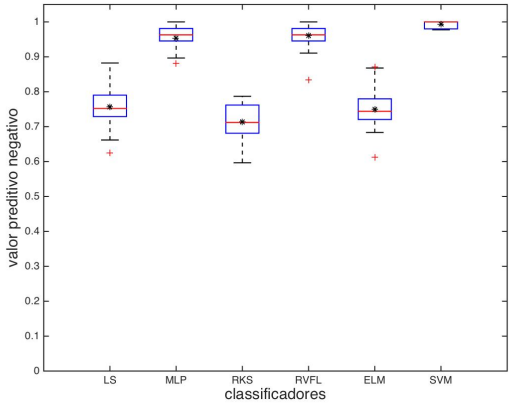
Figura 28 – VPN para todos os classificadores usando o método de extração de características Welch para os quatro pacientes.



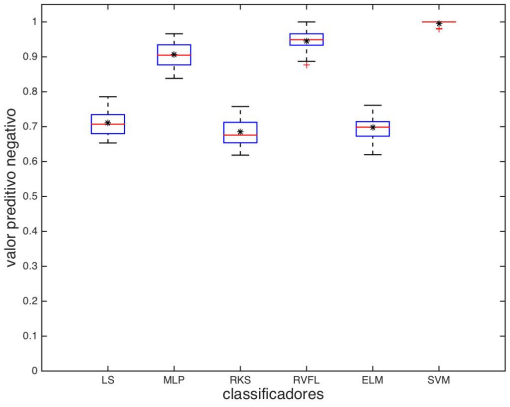
(a) VPN (P1, Welch)



(b) VPN (P2, Welch)



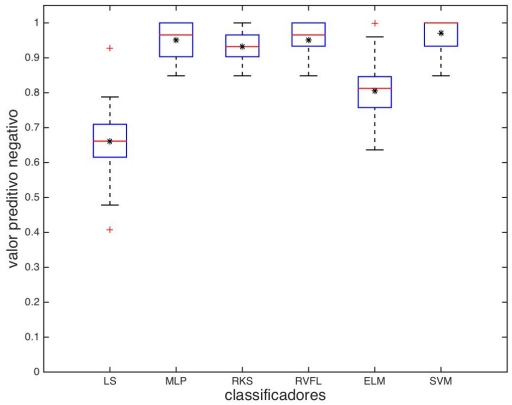
(c) VPN (P3, Welch)



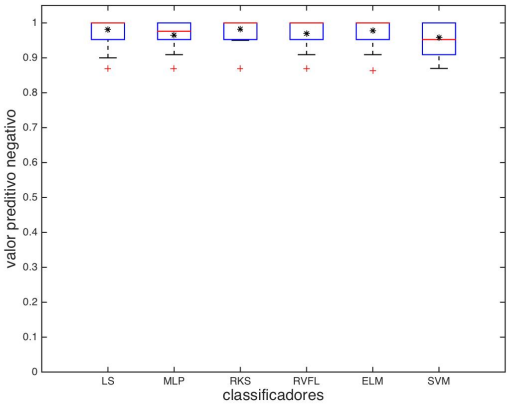
(d) VPN (P4, Welch)

Fonte: Elaborada pelo autor.

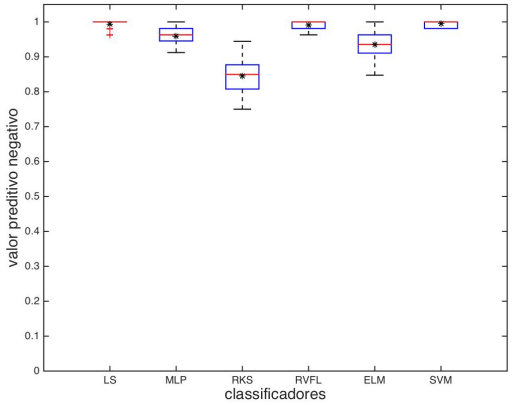
Figura 29 – VPN para todos os classificadores usando o método de extração de características LPC para os quatro pacientes.



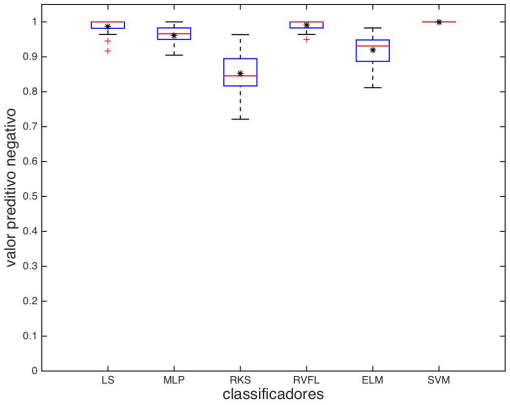
(a) VPN (P1, LPC)



(b) VPN (P2, LPC)



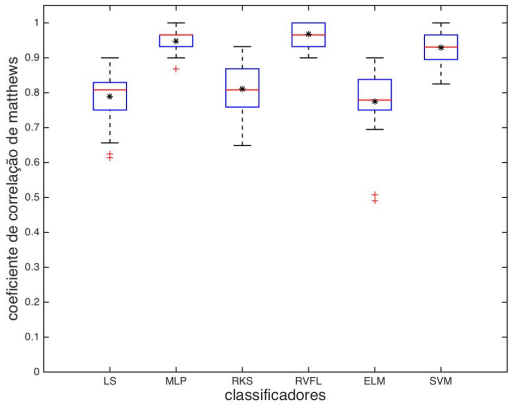
(c) VPN (P3, LPC)



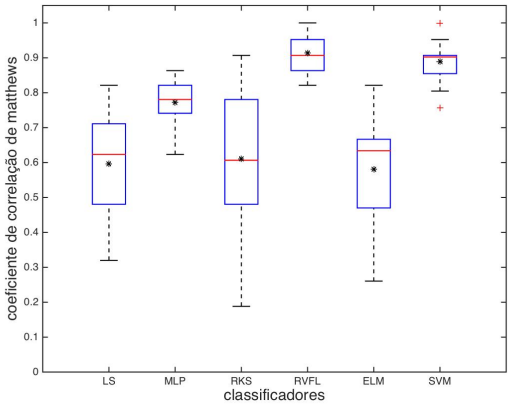
(d) VPN (P4, LPC)

Fonte: Elaborada pelo autor.

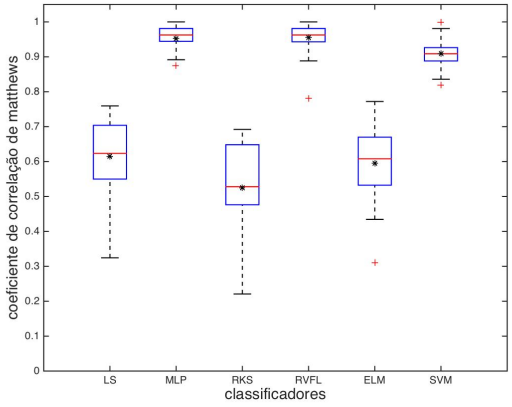
Figura 30 – MCC para todos os classificadores usando o método de extração de características Welch para os quatro pacientes.



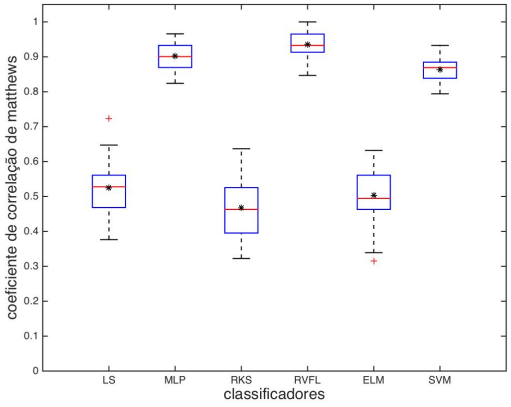
(a) MCC (P1, Welch)



(b) MCC (P2, Welch)



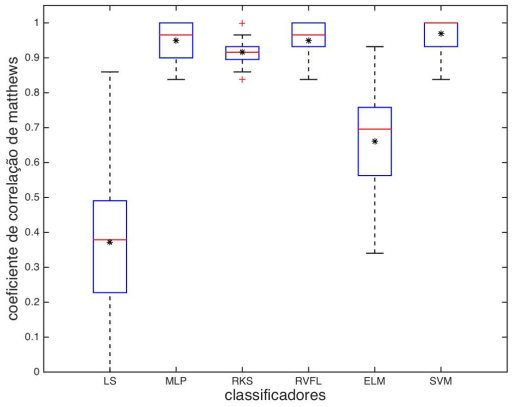
(c) MCC (P3, Welch)



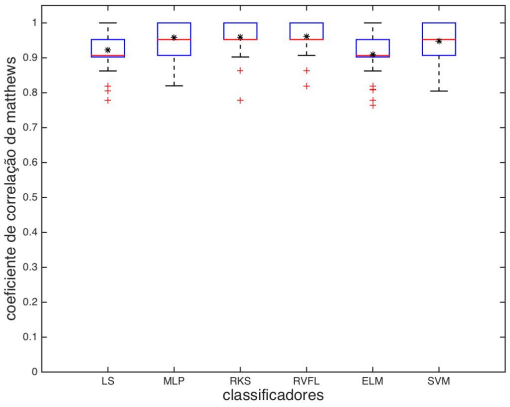
(d) MCC (P4, Welch)

Fonte: Elaborada pelo autor.

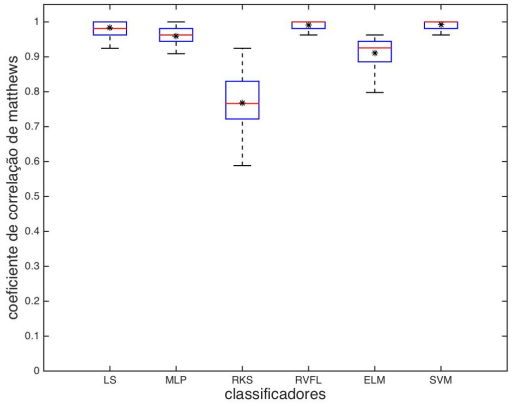
Figura 31 – MCC para todos os classificadores usando o método de extração de características LPC para os quatro pacientes.



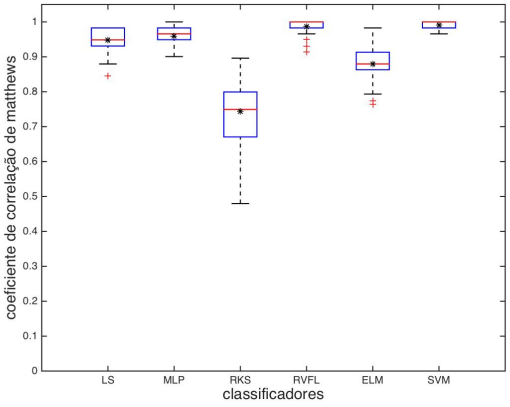
(a) MCC (P1, LPC)



(b) MCC (P2, LPC)



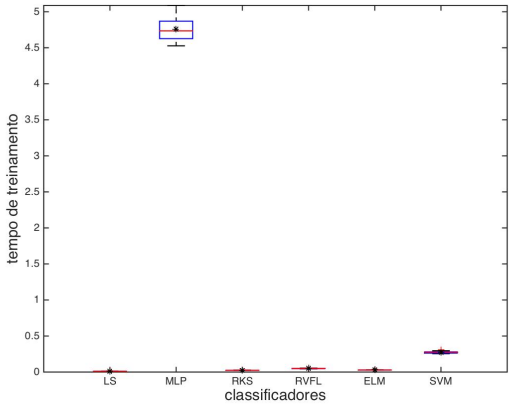
(c) MCC (P3, LPC)



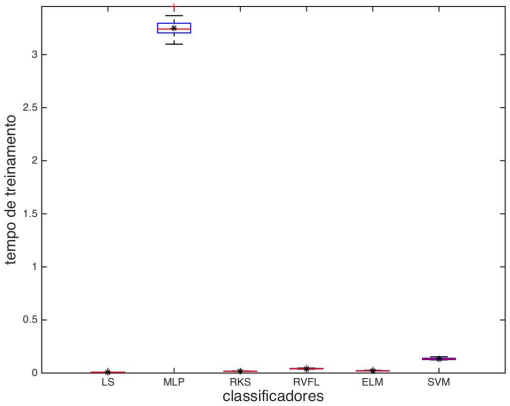
(d) MCC (P4, LPC)

Fonte: Elaborada pelo autor.

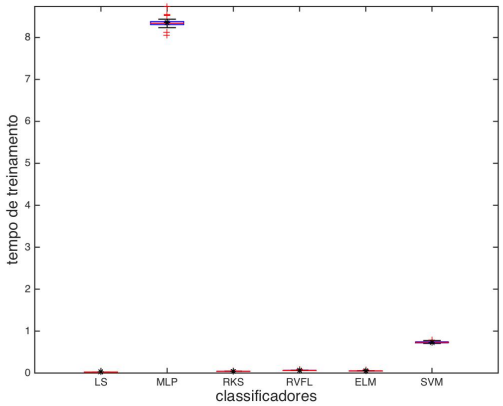
Figura 32 – TT para todos os classificadores usando o método de extração de características Welch para os quatro pacientes.



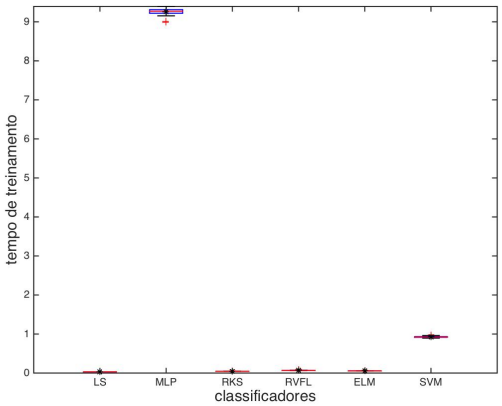
(a) TT (P1, Welch)



(b) TT (P2, Welch)



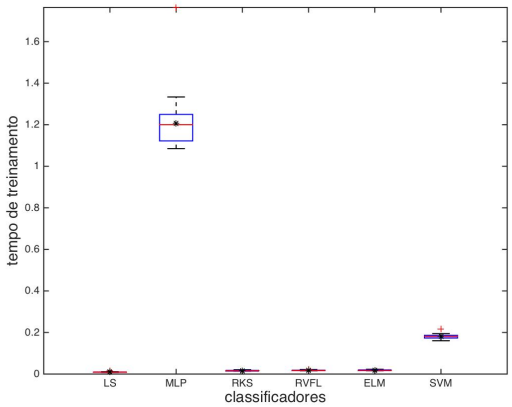
(c) TT (P3, Welch)



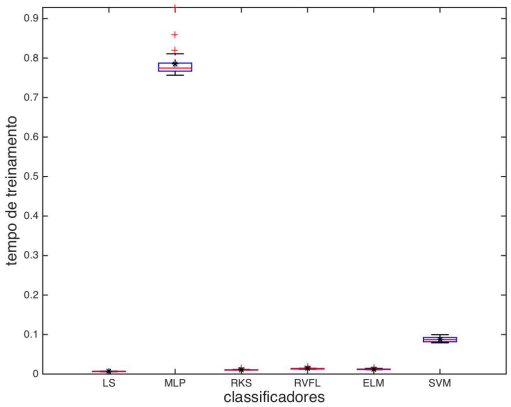
(d) TT (P4, Welch)

Fonte: Elaborada pelo autor.

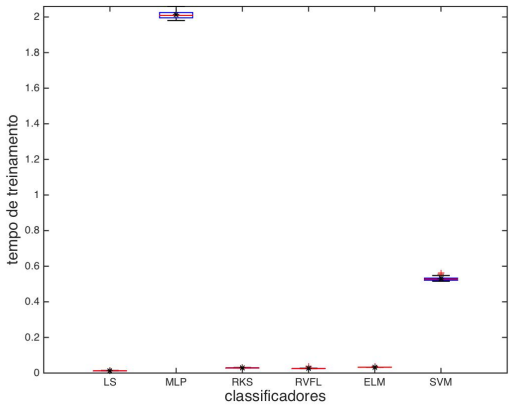
Figura 33 – TT para todos os classificadores usando o método de extração de características LPC para os quatro pacientes.



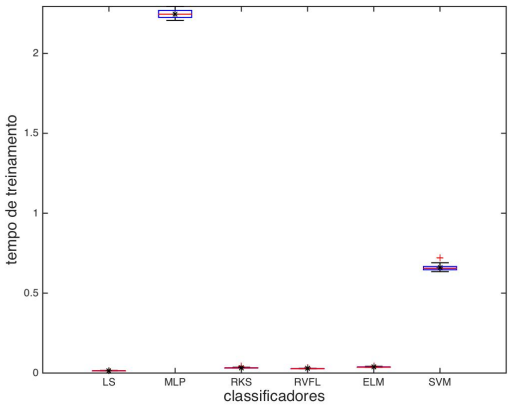
(a) TT (P1, LPC)



(b) TT (P2, LPC)



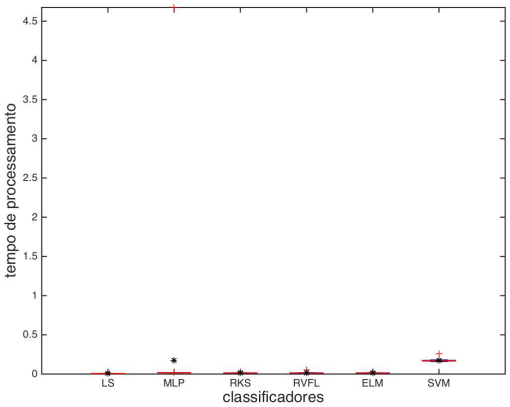
(c) TT (P3, LPC)



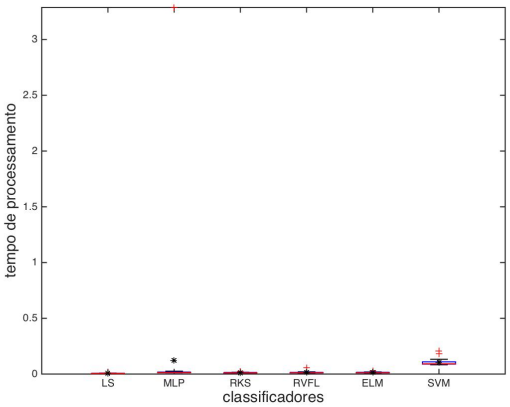
(d) TT (P4, LPC)

Fonte: Elaborada pelo autor.

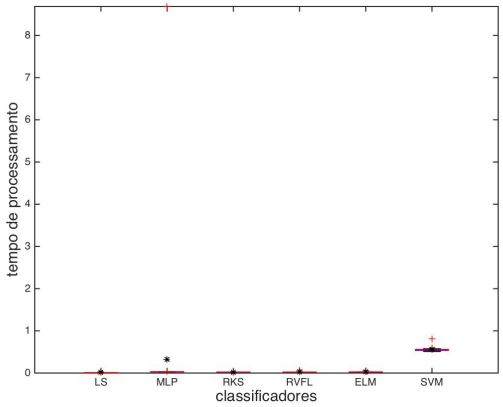
Figura 34 – TP para todos os classificadores usando o método de extração de características Welch para os quatro pacientes.



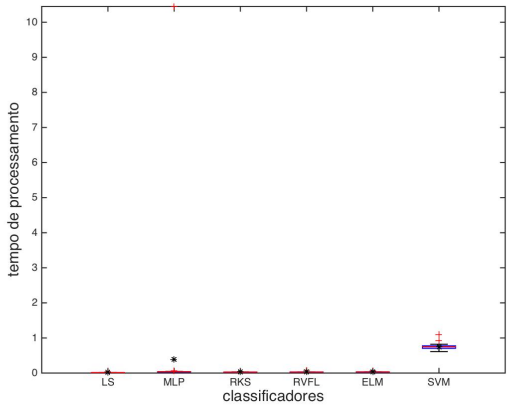
(a) TP (P1, Welch)



(b) TP (P2, Welch)



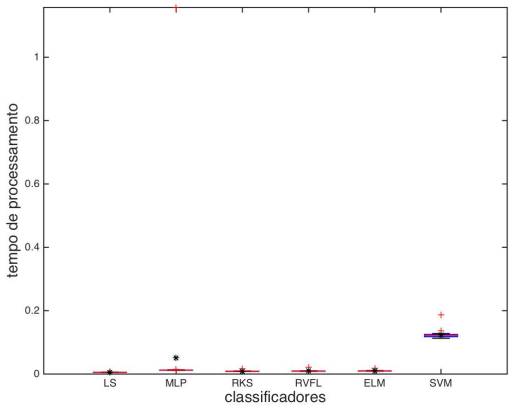
(c) TP (P3, Welch)



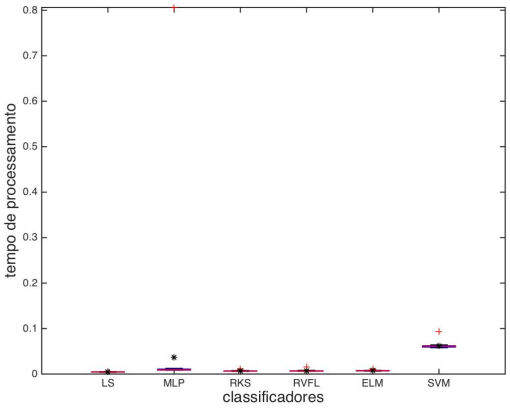
(d) TP (P4, Welch)

Fonte: Elaborada pelo autor.

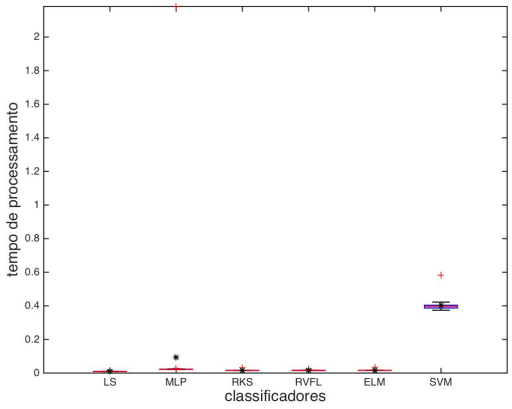
Figura 35 – TP para todos os classificadores usando o método de extração de características LPC para os quatro pacientes.



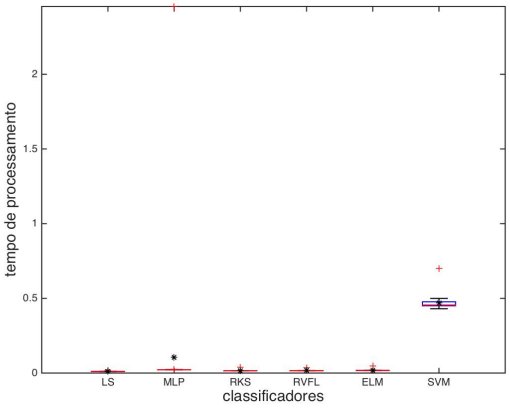
(a) TP (P1, LPC)



(b) TP (P2, LPC)



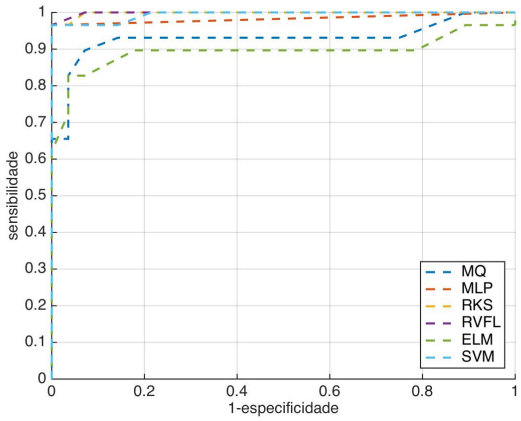
(c) TP (P3, LPC)



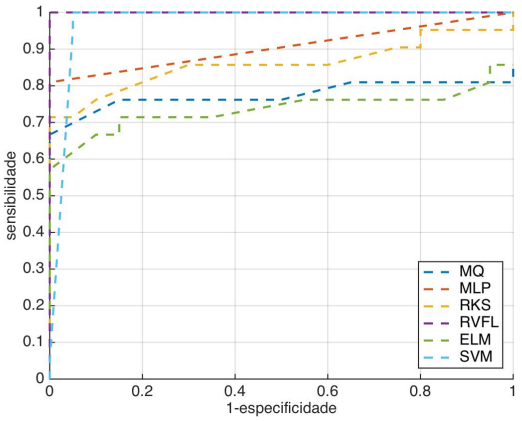
(d) TP (P4, LPC)

Fonte: Elaborada pelo autor.

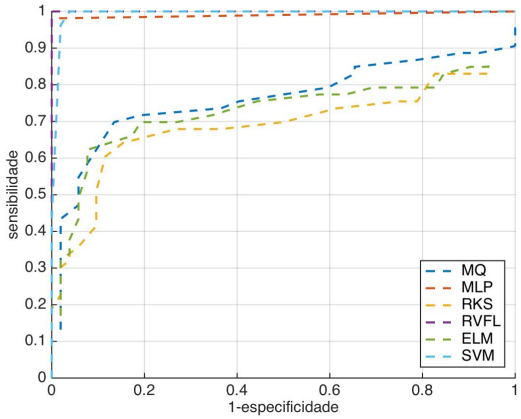
Figura 36 – Curvas ROC para todos os classificadores usando o método de extração de características Welch para os quatro pacientes.



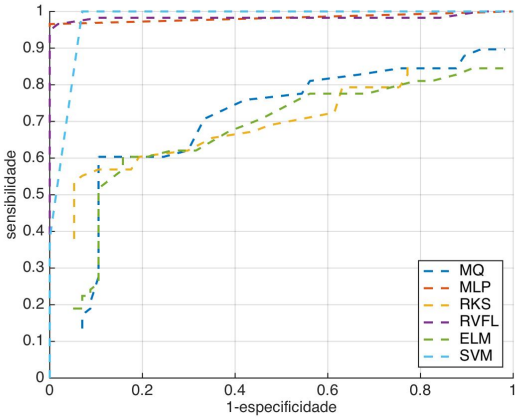
(a) Curvas ROC (P1, Welch)



(b) Curvas ROC (P2, Welch)



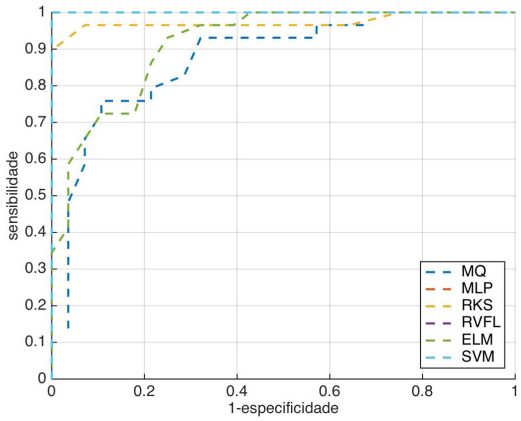
(c) Curvas ROC (P3, Welch)



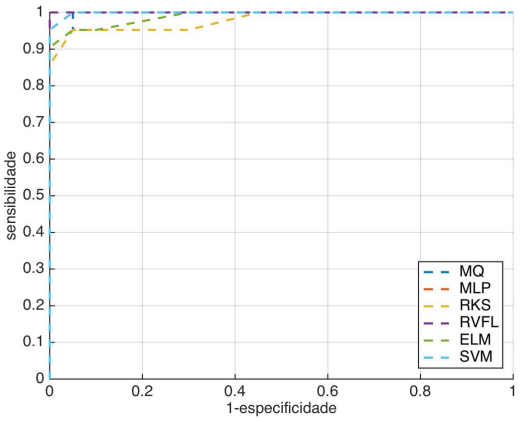
(d) Curvas ROC (P4, Welch)

Fonte: Elaborada pelo autor.

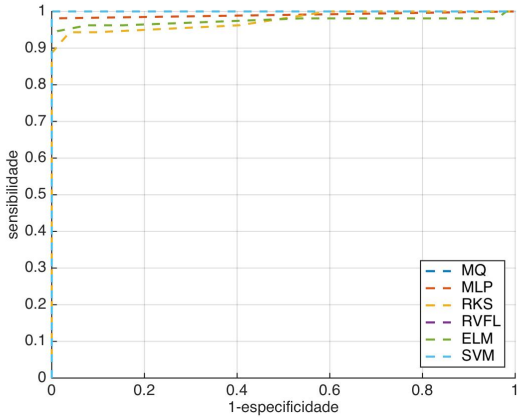
Figura 37 – Curvas ROC para todos os classificadores usando o método LPC para os quatro pacientes.



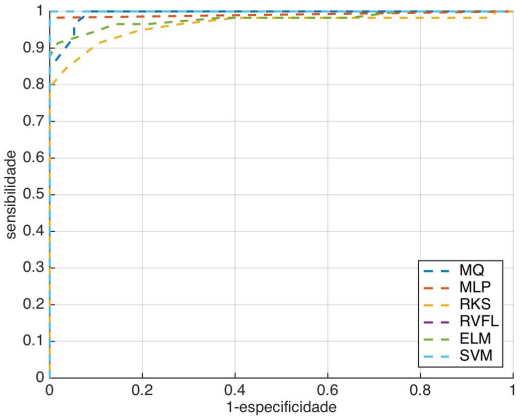
(a) Curvas ROC (P1, LPC)



(b) Curvas ROC (P2, LPC)



(c) Curvas ROC (P3, LPC)



(d) Curvas ROC (P4, LPC)

Fonte: Elaborada pelo autor.

Tabela 4 – Resultados dos pacientes 1, 2, 3 e 4 para os 48 cenários de simulação.

Paciente 1												
Medições	A	B	C	D	E	F	G	H	I	J	K	L
AC	0,885	0,973	0,898	0,984	0,877	0,964	0,684	0,974	0,957	0,974	0,827	0,984
SB	0,785	0,947	0,818	0,968	0,769	0,976	0,647	0,948	0,928	0,948	0,791	0,969
EP	0,988	1,000	0,981	1,000	0,988	0,952	0,721	1,000	0,987	1,000	0,865	1,000
MCC	0,789	0,948	0,810	0,968	0,775	0,930	0,372	0,950	0,917	0,950	0,661	0,970

Paciente 2												
Medições	M	N	O	P	Q	R	S	T	U	V	W	X
AC	0,787	0,872	0,795	0,954	0,777	0,943	0,959	0,978	0,979	0,980	0,953	0,973
SB	0,683	0,754	0,690	0,911	0,657	0,932	0,981	0,963	0,983	0,968	0,978	0,957
EP	0,897	0,997	0,905	1,000	0,903	0,955	0,937	0,993	0,975	0,992	0,927	0,990
MCC	0,597	0,772	0,610	0,914	0,581	0,889	0,922	0,958	0,959	0,961	0,909	0,948

Paciente 3												
Medições	Y	Z	AA	AB	AC	AD	AF	AG	AH	AI	AJ	AK
AC	0,802	0,976	0,757	0,977	0,792	0,953	0,991	0,979	0,881	0,996	0,955	0,996
SB	0,716	0,952	0,664	0,959	0,709	0,994	0,993	0,958	0,829	0,991	0,932	0,995
EP	0,889	1,000	0,851	0,996	0,876	0,911	0,990	1,000	0,934	1,000	0,978	0,997
MCC	0,615	0,953	0,525	0,955	0,595	0,910	0,983	0,959	0,768	0,991	0,911	0,992

Paciente 4												
Medições	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AX
AC	0,755	0,948	0,727	0,967	0,744	0,928	0,974	0,979	0,871	0,993	0,939	0,995
SB	0,654	0,898	0,621	0,943	0,637	0,996	0,987	0,959	0,847	0,991	0,914	1,000
EP	0,858	0,999	0,835	0,992	0,853	0,858	0,960	1,000	0,895	0,995	0,963	0,991
MCC	0,524	0,902	0,467	0,936	0,503	0,864	0,948	0,959	0,744	0,987	0,880	0,991

7 CONCLUSÕES E TRABALHOS FUTUROS

7.1 Conclusões

Neste artigo, comparamos os desempenhos de classificadores randomizados com aqueles obtidos por classificadores tradicionais (MQ, SVM e MLP) utilizando dois métodos de extração de características. A tarefa escolhida foi a detecção de ataques epilépticos de sinais EEG. Mostramos que apenas uma (a rede RVFL) dos três dos classificadores randomizados mais utilizados alcançou desempenhos comparáveis aos fornecidos pelos classificadores SVM e MLP. A vantagem adicional do treinamento rápido dos primeiros torna uma alternativa respeitável aos últimos.

Podemos inferir várias conclusões importantes a partir dos resultados numéricos do capítulo anterior.

(*i*) Em primeiro lugar, os classificadores MLP e SVM manteve em todos os resultados uma ótima performance, independentemente dos métodos de extração de recursos escolhidos.

(*ii*) O único classificador randomizado cujo desempenho é equivalente aos dos classificadores MLP e SVM é a rede RVFL.

(*iii*) Os desempenhos dos classificadores MQ, RKS e ELM são sempre inferiores aos dos classificadores MLP, SVM e RVFL, independentemente do método de extração de recurso usado.

(*iv*) Os desempenhos de todos os classificadores (incluindo o trio ELM / RKS / MQ) melhoram quando os atributos pelo método LPC são usados para ambos os pacientes (de acordo com as figuras 36 e 37).

Como conclusão geral, podemos afirmar que não podemos dar por certo que os desempenhos de redes randomizadas são sempre equivalentes ou superiores aos de classificadores não-lineares. Como mostramos os classificadores MLP e SVM uma ótima performance para o problema de interesse. Entre os classificadores randomizados, houve alta variabilidade entre os resultados, com o classificador RVFL obtendo claramente o melhor desempenho. Na realidade, o desempenho do classificador RVFL foi comparável àqueles apresentados pelos classificadores MLP e SVM. Com respeito a esses dois classificadores, o classificador RVFL oferece a vantagem adicional de treinamento mais rápido, sendo assim uma boa alternativa para eles.

7.2 Trabalhos futuros

Diversas são as ideias e continuidades que essa pesquisa gerou, mas dentre as principais, podemos destacar:

(*i*) Realizar os mesmos experimentos com novos bancos de dados com paciente de perfis mais variados.

(*ii*) Pesquisar e utilizar um novo atributo no vetor de entrada como o sinal de ECG ou algum outro sinal do corpo humano que possa ajudar a identificar o momento anterior ao surgimento de uma crise epilética.

(*iii*) Investigar a possibilidade de uma classe intermediária entre a crise epilética e o período de não crise, ou seja, um sinal anterior a crise que possa antevê-la.

(*iv*) Utilizar novos classificadores randomizados com mecanismos de busca em árvores com o intuito de conseguir novos tempos e dinâmicas de classificação.

(*v*) Utilizar outros extratores de atributos como DWT, análise da amplitude dos sinais e utilização de métodos de agrupamento.

(*vi*) Utilizar um tratamento do sinal por filtragem devido a alta incidência de ruído nos eletrodos na captura do sinal.

(*vii*) A classificação de novas anomalias cerebrais e atividades Mentais.

REFERÊNCIAS

ABE, S. **Support vector machines for pattern classification**. London: Springer-Verlag, 2005. ISBN 1-85233-929-9.

ADELI, H.; GHOSH-DASTIDAR, S. **Automated EEG-Based diagnosis od neurological disorders: Inventing the Future of Neurology**. [S.l.]: CRC Press, 2010.

AIZERMAN, M.; BRAVERMAN, E.; ROZONOER, L. Theoretical foundations of the potential function method in pattern recognition learning. **Automation and Remote Control**, v. 25, p. 821–837, 1964.

ALDRICH, J. Doing least squares: Perspectives from gauss and yule. **International Statistical Review**, Blackwell Publishing Ltd, v. 66, n. 1, p. 61–81, 1998.

ALOTAIBY, T. N.; ALSHEBEILI, S. A.; ALSHAWI, T.; AHMAD, I.; EL-SAMIE, F. E. A. EEG seizure detection and prediction algorithms: a survey. **EURASIP Journal on Advances in Signal Processing**, v. 2014, n. 1, p. 183, 2014.

ARAUJO, D. B. D.; CARNEIRO, A. A. O.; BAFFA, O. Localizando a atividade cerebral via magnetoencefalografia. **Ciencia e Cultura, São Paulo**, v. 56, n. 1, 2004.

BARTLETT, M. S. Periodogram analysis and continuous spectra. **Biometrika**, v. 27, p. 1–16, 1950.

BASMAJIAN, J. V.; LUCA, C. J. D. Muscles alive: their functions revealed by electromyography. **Williams & Wilkins**, 1985.

BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. **Nonlinear Programming Theory and Algorithms**. 2nd. ed. [S.l.]: Wiley, 1993.

BESSA, R. S. **Dinâmica do status epilepticus em dois modelos animais de epilepsia do lobo temporal**. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE, 2016.

BHAYA, A.; KASZKUREWICZ, E. Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method. **Neural Networks**, v. 17, n. 1, p. 65–71, 2004.

BORGES, R.; IAROSZ, K.; BATISTA, A.; CALDAS, I.; BORGES, F.; LAMEU, E. Sincronização de disparos em redes neuronais com plasticidade sináptica. **Revista Brasileira de Ensino de Fisica, v. 37, n. 2**, 2015.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: **Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory**. Pittsburgh, PA: ACM Press, 1992. p. 144–152.

BOTTOU, L.; CORTES, C.; DENKER, J.; DRUCKER, H.; GUYON, I.; JACKEL, L.; LECUN, Y.; MULLER, U.; SACKINGER, E.; SIMARD, P.; VAPNICK, V. Comparison of classifiers methods: a case study in handwriting digit recognition. In: **International Conference on Pattern Recognition**. [S.l.]: IEEE Computer Society Press, 1994.

BRADLEY, A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. **Pattern Recognition**, v. 30, n. 7, p. 1145–1159, 1997.

CHARNES, A.; FROME, E. L.; YU, P. L. The equivalence of generalized least squares and maximum likelihood estimates in the exponential family. **Journal of the American Statistical Association**, Taylor & Francis, v. 71, n. 353, p. 169–171, 1976.

CHEN, C. P. A rapid supervised learning neural network for function interpolation and approximation. **IEEE Transactions on Neural Networks**, v. 7, n. 5, p. 1220?1230, 1996.

CHEN, C. P.; WAN, J. Z. A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics**, v. 29, n. 1, p. 62–72, 1999.

CHIAPPA, S.; BENGIO, S. HMM and IOHMM modeling of EEG rhythms for asynchronous BCI systems. In: **Proceedings of the 12th European Symposium on Artificial Neural Networks (ESANN'2004)**. [S.l.: s.n.], 2004. p. 199–204.

CORTES, C.; VAPNIK, V. Support vector network. **Machine Learning**, v. 20, p. 273–297, 1995.

DING, S.; ZHANG, N.; XU, X.; GUO, L.; ZHANG, J. Deep extreme learning machine and its application in EEG classification. **Mathematical Problems in Engineering**, v. 2015, n. ID 129021, p. 1–11, 2015.

DONOS, C.; DUMPELMANN, M.; SCHULZE-BONHAGE, A. Early seizure detection algorithm based on intracranial EEG and random forest classification. **International Journal of Neural Systems**, v. 25, n. 5, p. 1–11, 2015.

DUFFY, F. H.; IYER, V. G.; SURWILLO, W. W. Eletroencefalografia clínica e mapeamento cerebral topográfico: tecnologia e prática. **Rio de Janeiro : Revinter**, p. 259p, 1999.

EHLERS, R. S. **ANÁLISE DE SÉRIES TEMPORAIS**. 2009.

FERRIS, M. C.; MUNSON, T. S. Interior-point methods for massive support vector machines. **Society for Industrial and Applied Mathematics**, v. 13, n. 3, p. 783–804, 2003.

GOLDBERGER, A. L.; AMARAL, L. A. N.; GLASS, L.; HAUSDORFF, J.; IVANOV, P. C. H.; MARK, R.; MIETUS, J.; MOODY, G.; PENG, C.-K.; STANLEY, H. **PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals.** 2000. CHB-MIT Scalp EEG Database. Disponível em: <http://circ.ahajournals.org/cgi/content/full/101/23/e215>. Acesso em: 10 jan. 2016.

GUNN, S. R. **Support Vector Machines for Classification and Regression**. [S.l.], 1998.

HAYES, M. H. Statistical digital signal processing and modeling. **USA: John Wiley and Sons**, 1996.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. Englewood Cliffs, NJ: Macmillan Publishing Company, 1994.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. 2nd. ed. [S.l.]: Prentice Hall, 1998.

HO, T. K. The random subspace method for constructing decision forests. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 20, n. 8, p. 832–844, 1998.

HUANG, G.; HUANG, G.-B.; SONG, S.; YOU, K. Trends in extreme learning machines: A review. **Neural Networks**, v. 61, n. 1, p. 32–48, 2015.

HUANG, G. B.; ZHU, Q. Y.; SIEW, C. K. Extreme learning machine: Theory and applications. **Neurocomputing**, v. 70, n. 1–3, p. 489–501, 2006.

IGELNIK, B.; PAO, Y.-H. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. **IEEE Trans. Neural Netw.**, v. 6, n. 6, p. 1320–1329, 1995.

INFO ESCOLA. **Tecido Nervoso**. 2017. Disponível em: <http://www.infoescola.com/biologia/tecido-nervoso/>. Acesso em: 5 set. 2017.

JAHBAKHANI, P.; KODOGIANNIS, V.; REVETT, K. Eeg signal classification using wavelet feature extraction and neural networks. **IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing**, p. 52–57, 2006.

JASPER, H. H. The ten-twenty electrode system of the international federation. **Electroencephalography and Clinical Neurophysiology, EEG Journal, 10, 371-375**, 1958.

KANASHIRO, A. L. A. N. **EPILEPSIA: prevalência, características epidemiológicas e lacuna de tratamento farmacológico**. Tese (Doutorado) — Curso de Medicina, Faculdade de Ciências Médicas da Universidade Estadual de Campinas, 2006.

KANDEL. **Eletrodos para Eletroencefalograma**. 2016. Disponível em: <https://kandel.com.br/eletrodos/eeg/>. Acesso em: 03 nov. 2017.

KEMP, B.; VARRI, A.; ROSA, A. C.; NIELSEN, K. D.; GADE, J. A simple format for exchange of digitized polygraphic recordings. **Electroencephalography and Clinical Neurophysiology**, v. 82, p. 391–393, 1992.

KEMP, B.; VäRRI, A.; ROSA, A. C.; NIELSEN, K. D.; GADE, J. **European Data Format**. 1992. Disponível em: <http://www.edfplus.info/specs/edf.html>. Acesso em: 20 nov. 2016.

KLEM, G. H.; LUDERS, H. O.; JASPER, H.; ELGER, C. The ten-twenty electrode system of the international federation. **International Federation of Clinical Neurophysiology**, 1999.

KNERR, S.; PERSONNAZ, L.; DREYFUZ, G. **Single-layer learning revisited: a stepwise procedure for building and training a neural network**. [S.l.]: Springer-Verlag, 1990.

LEHNERTZ, K.; MORMANN, F.; KREUZ, T.; ANDRZEJAK, R.; RIEKE, C.; DAVID, P.; ELGER, C. Seizure prediction by nonlinear EEG analysis. **IEEE Engineering in Medicine and Biology Magazine**, v. 22, n. 1, p. 57–63, 2003.

LOGAR, C.; WALZL, B.; LECHNER, H. Seizure prediction by nonlinear EEG analysis. **Role of long-term EEG monitoring in diagnosis and treatment of epilepsy**, v. 34, n. Suppl 1, p. 29–32, 1994.

MERCER, J. Functions of positive and negative type and their connection with the theory of integral equations. **A Philos. Trans. Roy. Soc.**, v. 209, p. 425–446, 1909.

MISULIS, K. E. Basic electronics for clinical neurophysiology. **Journal of Clinical Neurophysiology**, 1989.

ORGANIZATION, W. H. **Epilepsy**. 2017. Disponível em: <http://www.who.int/mediacentre/factsheets/fs999/en/>. Acesso em: 05 out. 2017.

OSUNA, E.; FREUND, R.; GIROSI, F. An improved training algorithm for support vector machines. In: PRINCIPE, J.; GILE, L.; MORGAN, N.; WILSON, E. (Ed.). **Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing VII**. New York: IEEE, 1997. p. 276–285.

PAO, Y.; PARK, G.; SOBAJIC, D. Learning and generalization characteristics of the random vector functional-link net. **Neurocomputing**, v. 6, p. 163–180, 1994.

PENNY, W. D.; ROBERTS, S. J.; CURRAN, E.; STOKES, M. J. EEG-based communication: a pattern recognition approach. **IEEE Transactions on Rehabilitation Engineering**, v. 8, n. 2, p. 214–215, 2000.

PFURTSCHELLER, G.; NEUPER, C.; SCHLOGL, A.; LUGGER, K. Separability of eeg signals recorded during right and left motor imagery using adaptive autoregressive parameters. **IEEE Trans. Rehabil. Eng**, v. 6, p. 316–355, 1998.

RAHIMI, A. **Random Features**. 2016. Disponível em: <https://keysduplicated.com/~ali/random-features/>. Acesso em: 01 jun. 2017.

RAHIMI, A.; RECHT, B. Random features for large-scale kernel machines. In: PLATT, J. C.; KOLLER, D.; SINGER, Y.; ROWEIS, S. T. (Ed.). **Advances in Neural Information Processing Systems 20**. [S.l.]: Curran Associates, Inc., 2008. p. 1177–1184.

RAHIMI, A.; RECHT, B. Uniform approximation of functions with random bases. In: **Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing**. [S.l.: s.n.], 2008. p. 555–561.

RAHIMI, A.; RECHT, B. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In: KOLLER, D.; SCHUURMANS, D.; BENGIO, Y.; BOTTOU, L. (Ed.). **Advances in Neural Information Processing Systems 21**. [S.l.]: Curran Associates, Inc., 2009. p. 1313–1320.

ROSENBLATT, F. X. principles of neurodynamics: Perceptrons and the theory of brain mechanisms. **Spartan Books**, Washington DC, 1961.

SANEI, S.; CHAMBERS, J. A. Eeg signal processing. **England: John Wiley and Sons**, 2007.

SHARMA, S.; KUMAR, G.; MISHRA, D. K.; MOHAPATRA, D. Design and implementation of a variable gain amplifier for biomedical signal acquisition. **International Journal of Advanced Research in Computer Science and Software Engineering, v. 2(2)**, 2012.

SHAWE-TAYLOR, J.; BARTLETT, P. L.; WILLIAMSON, R. C.; ANTHONY, M. Structural risk minimization over data-dependent hierarchies. **IEEE Transactions on Information Theory**, v. 44, n. 5, p. 1926–1940, 1998.

SHOEB, A. **CHB-MIT Scalp EEG Database**. 2016. Disponível em: <https://www.physionet.org/pn6/chbmit/>. Acesso em: 03 out. 2017.

SHOEB, A.; GUTTAG, J. Application of machine learning to epileptic seizure detection. In: **Proceedings of the 27th International Conference on Machine Learning (ICML 2010)**. [S.l.: s.n.], 2010. p. 1–8.

SHOEB, A. H. **Application of machine learning to epileptic seizure onset detection and treatment**. Tese (Doutorado) — Harvard University, MIT Division of Health Sciences and Technology, 2009.

SMOLA, A. J.; BARTLETT, P. L.; SCHÖLKOPF, B.; SCHUURMANS, D. **Advances in Large Margin Classifiers**. Cambridge, Massachusetts: The MIT Press, 2000.

SOUZA JUNIOR, A. H.; CORONA, F.; BARRETO, G. A.; MICHE, Y.; LENDASSE, A. Minimal learning machine: A novel supervised distance-based approach for regression and classification. **Neurocomputing**, v. 164, n. 21, p. 34–44, 2015.

SPACKMAN, K. A. Signal detection theory: Valuable tools for evaluating inductive learning. **6th Int Workshop on Machine Learning (ICML 1989), Morgan Kaufmann**, p. 160–163, 1989.

STITSON, M. O.; WESTON, J. A. E. **Implementational issues of support vector machines**. [S.l.], 1996.

SUBASI, A. EEG signal classification using wavelet feature extraction and a mixture of expert model. **Expert Systems with Applications**, v. 32, n. 4, p. 1084–1093, 2007.

SUBHA, D. P.; JOSEPH, P. K.; ACHARYA, R.; LIM, C. M. Eeg signal analysis: A survey. **Journal of Medical Systems**, v. 34, n. 2, p. 195–212, 2010.

SUTHERLAND, D. J.; SCHNEIDER, J. On the error of random fourier features. In: **Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI'2015)**. [S.l.: s.n.], 2015. p. 862–871.

THERRIEN, C. W. **Discrete Random Signals ans Statistical Signal Processing**. New Jersey: Prentice-Hall, 1992.

VAPNIK, V. **The Nature of Statistical Learning Theory**. [S.l.]: Springer, 1995.

VAPNIK, V. **Statistical Learning Theory**. [S.l.]: Wiley-Interscience, 1998.

WANG, Y.; LI, Z.; FENG, L.; ZHENG, C.; ZHANG, W. Automatic detection of epilepsy and seizure using multiclass sparse extreme learning machine classification. **Computational and Mathematical Methods in Medicine**, v. 2017, n. ID 6849360, p. 1–10, 2017.

WEBSTER, J. G. Medical instrumentation: Application and design. In: ____. 4th. ed. [S.l.]: John Wiley & Sons, 2009. p. 160–190.

WELCH, P. D. The use of the fast fourier transform for the estimation of power spectra. **IEEE Transactions on Audio Electroacoustics**, v. 15, n. 2, p. 70–73, 1967.

WIDROW, B.; GREENBLATT, A.; KIM, Y.; PARK, D. The No-Prop algorithm: A new learning algorithm for multilayer neural networks. **Neural Networks**, v. 37, p. 182–188, 2013.

ZHANG, L.; SUGANTHAN, P. N. A comprehensive evaluation of random vector functional link networks. **Information Sciences**, 2015.

ZHANG, L.; SUGANTHAN, P. N. A comprehensive evaluation of random vector functional link networks. **Information Sciences**, v. 367–368, p. 1094–1105, 2016.

ZHANG, L.; SUGANTHAN, P. N. A survey of randomized algorithms for training neural networks. **Information Sciences**, v. 364–365, p. 146–155, 2016.

ZHAO, H.; GUO, X.; WANG, M.; LI, T.; PANG, C.; GEORGAKOPOULOS, D. Analyze EEG signals with extreme learning machine based on PMIS feature selection. **International Journal of Machine Learning and Cybernetics**, p. 1–7, 2015.

APÊNDICE A – CLASSIFICADOR DE MÍNIMOS QUADRADOS

Assumindo que N pares de dados $\{(\mathbf{x}_\mu, \mathbf{d}_\mu)\}_{\mu=1}^{N}$ estejam disponíveis para construção e avaliação do modelo, onde $\mathbf{x}_\mu \in \mathbb{R}^{p+1}$ é o μ-ésimo padrão de entrada[1] e $\mathbf{d}_\mu \in \mathbb{R}^K$ é o rótulo da classe alvo correspondente, com K denotando o número de classes. Para os rótulos, assumimos um esquema de codificação 1-de-K, ou seja, para cada vetor de rótulos $\mathbf{d}_\mu$, a componente cujo índice corresponde à classe do padrão $\mathbf{x}_\mu$ é definida como "+1", enquanto as outras $K-1$ componentes são definidas como "-1".

Então, primeiramente deve-se selecionar aleatoriamente N_1 ($N_1 < N$) pares de dados a partir do conjunto de dados disponível e os organizar ao longo das colunas das matrizes $\mathbf{D}$ e $\mathbf{X}$, como segue:

$$\mathbf{X} = [\mathbf{x}_1 \mid \mathbf{x}_2 \mid \cdots \mid \mathbf{x}_{N_1}] \quad \text{e} \quad \mathbf{D} = [\mathbf{d}_1 \mid \mathbf{d}_2 \mid \cdots \mid \mathbf{d}_{N_1}]. \tag{A.1}$$

em que $\dim(\mathbf{X}) = (p+1) \times N_1$ e $\dim(\mathbf{D}) = m \times N_1$. O objetivo é usar as matrizes $\mathbf{X}$ e $\mathbf{D}$ para obter o seguinte mapeamento linear:

$$\mathbf{D} = \beta\mathbf{X} \quad (\text{modo } \textit{batch}), \tag{A.2}$$

que pode também ser escrito em mapeamentos individuais do tipo

$$\mathbf{d}_\mu = \beta\mathbf{x}_\mu \quad (\text{modo padrão-a-padrão}), \tag{A.3}$$

para $\mu = 1, \ldots, N_1$. Para ambos os modos de operação, a dimensão da matriz β é $K \times (p+1)$.

A solução de mínimos quadrados ordinários (MQO) do sistema linear na Equação (A.2) é dada pela inversa generalizada Moore-Penrose, ou seja

$$\hat{\beta} = \mathbf{D}\mathbf{X}^T \left(\mathbf{X}\mathbf{X}^T\right)^{-1}, \tag{A.4}$$

em que o símbolo ($\wedge$) indica uma estimativa do operador matriz β. A solução de norma-mínima para Equação (A.2) é dada pela versão regularizada da Equação (A.4):

$$\hat{\beta} = \mathbf{D}\mathbf{X}^T \left(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I}\right)^{-1}, \tag{A.5}$$

em que $\mathbf{I}$ é a matriz identidade de dimensão $(p+1) \times (p+1)$ e λ é um parâmetro de regularização positivo muito pequeno.

[1] A primeira componente de $\mathbf{x}_\mu$ é igual a 1 para poder incluir o *bias* com parâmetro a ser estimado.

É importante notar que o vetor de parâmetros $\hat{\beta}_i \in \mathbb{R}^{p+1}$, $i = 1,\ldots,m$, pode ser calculado individualmente por meio da seguinte equação:

$$\hat{\beta}_i = \left(\mathbf{X}\mathbf{X}^T\right)^{-1}\mathbf{X}\mathbf{D}_i^T, \tag{A.6}$$

em que o vetor $\mathbf{D}_i$ corresponde à i-ésima linha da matriz $\mathbf{D}$. O vetor estimado $\hat{\beta}_i$ deve ser interpretado como sendo o vetor de pesos (incluindo o *bias*) do i-ésimo neurônio de saída do classificador MQ.

Predição da Classe para um Padrão Desconhecido: Uma vez de posse da matriz de pesos $\hat{\beta}$ estimada, os $N_2 = N - N_1$ pares de dados restantes são usados para validar o modelo. Dessa forma, para o modo de recuperação padrão-a-padrão, a saída do classificador MQ é dada por

$$\mathbf{y}_\mu = \hat{\beta}\mathbf{x}_\mu, \tag{A.7}$$

para $\mu = 1,\ldots,N_2$, enquanto para o modo *batch*, tem-se

$$\mathbf{Y} = \hat{\beta}\mathbf{X}. \tag{A.8}$$

O índice da classe predita i_μ^* para o μ-ésimo padrão de entrada de teste é então dado pela seguinte regra de decisão:

$$i_\mu^* = \arg\max_{i=1,\ldots,K}\{y_{i\mu}\} = \arg\max_{i=1,\ldots,K}\{\hat{\beta}_i^T\mathbf{x}_\mu\}, \tag{A.9}$$

em que $y_{i\mu} = \hat{\beta}_i^T\mathbf{x}_\mu$ é a i-ésima componente do vetor $\mathbf{y}_\mu$ calculado como na Equação (A.7), com o vetor $\hat{\beta}_i^T$ sendo a i-ésima linha da matriz $\hat{\beta}$.

A.1 Implementação em Matlab/Octave

As Equações (A.4) e (A.5) usadas, respectivamente, para estimar a matriz de pesos do classificador MQ e de sua versão regularizada, podem ser facilmente implementadas no ambiente Matlab.

Assumindo que os vetores de atributos $\mathbf{x}_\mu$, $\mu = 1,\ldots,N_1$, usados no treinamento do classificador MQ estejam dispostos ao longo das colunas da matriz `X` e que os rótulos correspondentes estejam dispostos ao longo das colunas da matriz `D`, então a Equação (A.4) pode ser implementada da forma que se lê, ou seja,

```
» B = D*X'*inv(X*X');
```

em que B denota a estimativa MQO da matriz de pesos β. Contudo, esta forma de se estimar β não é recomendada por ter elevado custo computacional e por ser muito susceptível a erros numéricos. Neste caso, recomenda-se usar o operador *barra* (/), ou seja

```
» B = D/X;
```

Uma outra maneira de estimar a matriz β é através do comando PINV:

```
» B = D*pinv(X);
```

Para a versão regularizada do classificador MQ, também é possível estimar $\hat{\beta}$ através da escrita direta da Equação (A.5) no prompt do Matlab:

```
» l = 0.01;
» I=ones(size(X*X'));
» B = D*X'*inv(X*X' + l*I);
```

Porém, pelas mesmas razões apontadas anteriormente, recomenda-se o uso do operador *barra* (/). Neste caso, a sequência de comandos passa ser a seguinte:

```
» l = 0.01;
» I=eye(size(X*X'));
» A=X*X' + l*I;
» R=D*X';
» B = R/A;
```

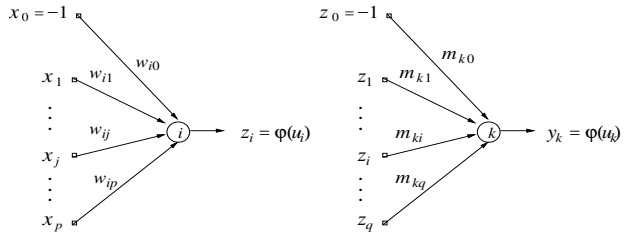
APÊNDICE B – PERCEPTRON MULTICAMADAS E O ALGORITMO DE RETROPROPAGAÇÃO DO ERRO

B.1 Perceptron Multicamadas

Estamos considerando nas definições e cálculos a seguir uma arquitetura de rede neural do tipo **Perceptron Multicamada** (MLP - *Multilayer Perceptron*) com apenas uma camada escondida de neurônios treinados com o algoritmo de **retropropagação do erro** (*Error Backpropagation*).

Os neurônios da camada escondida (primeira camada de pesos sinápticos) são representados conforme mostrado na Figura 38a, enquanto os neurônios da camada de saída segunda camada de pesos sinápticos) são representados conforme mostrado na Figura 38b.

Figura 38 – Camadas de Neurônios: (a) Neurônio da camada escondida. (b) Neurônio da camada de saída.



(a) (b)

O vetor de pesos associado a cada neurônio i da camada escondida, também chamada de *camada oculta* ou *camada intermediária*, é representado como

$$\mathbf{w}_i = \begin{pmatrix} w_{i0} \\ \vdots \\ w_{ip} \end{pmatrix} = \begin{pmatrix} \theta_i \\ \vdots \\ w_{ip} \end{pmatrix} \tag{B.1}$$

em que θ_i é o limiar (*bias* ou *threshold*) associado ao neurônio i. Os neurônios desta camada são chamados de neurônios escondidos por não terem acesso direto à saída da rede MLP, onde são calculados os erros de aproximação.

De modo semelhante, o vetor de pesos associado a cada neurônio k da camada de saída é representado como

$$\mathbf{m}_k = \begin{pmatrix} m_{k0} \\ \vdots \\ m_{kq} \end{pmatrix} = \begin{pmatrix} \theta_k \\ \vdots \\ m_{kq} \end{pmatrix} \tag{B.2}$$

em que θ_k é o limiar associado ao neurônio de saída k.

O treinamento da rede MLP se dá em duas etapas, que são descritas a seguir.

B.2 Fase 1: Sentido Direto

Esta etapa de funcionamento da rede MLP envolve o cálculo das ativações e saídas de todos os neurônios da camada escondida e de todos os neurônios da camada de saída. Assim, o fluxo de sinais (informação) se dá dos neurônios de entrada para os neurônios de saída, passando obviamente pelos neurônios da camada escondida. Por isso, diz-se que o informação está fluindo no sentido **direto** (*forward*), ou seja:

Entrada $\rightarrow$ Camada Intermediária $\rightarrow$ Camada de Saída

Assim, após a apresentação de um vetor de entrada $\mathbf{x}$, na iteração t, o primeiro passo é calcular as ativações dos neurônios da camada escondida:

$$u_i(n) = \sum_{j=0}^{p} w_{ij}(n)x_j(n) = \mathbf{w}_i^T(n)\mathbf{x}(n), \quad i = 1,\ldots,q \tag{B.3}$$

em que T indica o vetor (ou matriz) transposto e q indica o número de neurônios da camada escondida. Em seguida, as saídas correspondentes são calculadas como:

$$z_i(n) = \phi_i(u_i(n)) = \phi_i\left(\sum_{j=0}^{p} w_{ij}(n)x_j(n)\right) = \phi_i\left(\mathbf{w}_i^T(n)\mathbf{x}(n)\right) \tag{B.4}$$

tal que a função de ativação ϕ assume geralmente uma das seguintes formas:

$$\phi_i(u_i(n)) = \frac{1}{1+\exp[-u_i(n)]}, \qquad \text{(Logística)} \tag{B.5}$$

$$\phi_i(u_i(n)) = \frac{1-\exp[-u_i(n)]}{1+\exp[-u_i(n)]}, \qquad \text{(Tangente Hiperbólica)} \tag{B.6}$$

O segundo passo consiste em repetir as operações das Equações (B.3) e (B.4) para os neurônios da camada de saída:

$$u_k(n) = \sum_{i=0}^{q} m_{ki}(n)z_i(n), \quad k = 1,\ldots,M \tag{B.7}$$

em que M é o número de neurônios de saída. Note que as saídas dos neurônios da camada escondida, $z_i(n)$, fazem o papel de entrada para os neurônios da camada de saída.

Em seguida, as saídas dos neurônios da camada de saída são calculadas como:

$$y_k(n) = \phi_k(u_k(n)) = \phi_k\left(\sum_{i=0}^{q} m_{ki}(n)z_i(n)\right) \tag{B.8}$$

tal que a função de ativação ϕ_k assume geralmente uma das formas definidas nas Equações (B.5) e (B.6).

B.3 Fase 2: Sentido Inverso

Esta etapa de funcionamento da rede MLP envolve o cálculo dos gradientes locais e o ajuste dos pesos de todos os neurônios da camada escondida e da camada de saída. Assim, o fluxo de sinais (informação) se dá dos neurônios de saída para os neurônios da camada escondida. Por isso, diz-se que o informação está fluindo no sentido **inverso** (*backward*), ou seja:

Camada de Saída $\rightarrow$ Camada Escondida

Assim, após os cálculos das ativações e saídas levados a cabo na Fase 1, o primeiro passo da Fase 2 consiste em calcular os gradientes locais dos neurônios da camada de saída:

$$\delta_k(n) = e_k(n)\phi'(u_k(n)), \quad k = 1,\ldots,M \tag{B.9}$$

em que $e_k(n)$ é o erro entre a saída desejada $d_k(n)$ para o neurônio k e saída gerada por ele, $o_k(n)$:

$$e_k(n) = d_k(n) - y_k(n), \qquad k = 1,\ldots,M \tag{B.10}$$

A derivada $\phi'(u_k(n))$ assume diferentes formas, dependendo da escolha da função de ativação. Assim, temos as seguintes possibilidades:

$$\phi'_k(u_k(n)) = \frac{d\phi_k(u_k(n))}{du_k(n)} = y_k(n)[1-y_k(n)], \quad \text{Se } \phi_k(u_k(n)) \text{ é a função logística} \tag{B.11}$$

$$\phi'_k(u_k(n)) = \frac{d\phi_k(u_k(n))}{du_k(n)} = 1-y_k^2(n), \quad \text{Se } \phi_k(u_k(n)) \text{ é a tangente hiperbólica} \tag{B.12}$$

O segundo passo da Fase 2 consiste em calcular os gradientes locais dos neurônios da camada escondida:

$$\delta_i(n) = \phi'_i(u_i(n)) \sum_{k=1}^{n} m_{ki}\delta_k(n), \quad i = 1,\ldots,q \tag{B.13}$$

tal que a derivada $\phi'(u_i(n))$ pode ser calculada por uma das seguintes formas:

$$\phi'_i(u_i(n)) = \frac{d\phi_i(u_i(n))}{du_i(n)} = y_i(n)[1-y_i(n)], \quad \text{Se } \phi_i(u_i(n)) \text{ é a função logística} \tag{B.14}$$

$$\phi'_i(u_i(n)) = \frac{d\phi_i(u_i(n))}{du_i(n)} = \frac{1}{2}[1-y_i^2(n)], \quad \text{Se } \phi_i(u_i(n)) \text{ é a tangente hiperbólica} \tag{B.15}$$

O terceiro passo da Fase 2 corresponde ao processo de atualização ou ajuste dos parâmetros (pesos sinápticos e limiares) da rede MLP com uma camada escondida. Assim, para a camada escondida temos que a regra de atualização dos pesos, w_{ij}, é dada por:

$$\begin{aligned} w_{ij}(n+1) &= w_{ij}(n) + \Delta w_{ij}(n) \\ &= w_{ij}(n) + \alpha \delta_i(n) x_j(n) \end{aligned} \tag{B.16}$$

em que $\alpha(n)$ é a taxa de aprendizagem. E para camada de saída temos que a regra de atualização dos pesos, m_{ki}, é dada por:

$$\begin{aligned} m_{ki}(n+1) &= m_{ki}(n) + \Delta m_{ki}(n) \\ &= m_{ki}(n) + \alpha \delta_k(n) z_i(n) \end{aligned} \tag{B.17}$$

B.4 Treinamento, Convergência e Generalização

O projeto de uma rede neural envolve a especificação de diversos itens, cujos valores influenciam consideravelmente funcionamento do algoritmo. A seguir especificaremos a lista destes itens juntamente com as faixas de valores que os mesmos podem assumir:

Dimensão do vetor de Entrada (p): Este item pode assumir em tese valores entre 1 e ∞. Porém, existe um limite superior que depende da aplicação de interesse e do custo de se medir (observar) as variáveis x_j. É importante ter em mente que um valor alto para p não indica necessariamente um melhor desempenho para a rede neural, pois pode haver redundância no processo de medição. Neste caso, uma certa medida é, na verdade, a combinação linear de outras medidas, podendo ser descartada sem prejuízo ao desempenho da rede. Quando é muito caro, ou até impossível, medir um elevado número de variáveis x_j, deve-se escolher aquelas que o especialista da área considera como mais relevante ou representativas para o problema. O ideal seria que cada variável x_j, $j = 1, \ldots, p$, "carregasse" informação que somente ela contivesse. Do ponto de vista estatístico, isto equivale a dizer que as variáveis são *independentes* ou *não-correlacionadas* entre si.

Dimensão do vetor de saída (M): Assim como o primeiro item, este também depende da aplicação. Se o interesse está em problemas de aproximação de funções, $\mathbf{y} = F(\mathbf{x})$, o número de neurônios deve refletir diretamente a quantidades de funções de saída desejadas (ou seja, a dimensão de $\mathbf{y}$).

Se o interesse está em problemas de classificação de padrões, a coisa muda um pouco de figura. Neste caso, o número de neurônios deve codificar o número de classes desejadas.

É importante perceber que estamos chamando as classes às quais pertencem os vetores de dados de uma forma bastante genérica: classe 1, classe 2, ..., etc. Contudo, à cada classe pode estar associado um rótulo (e.g. classe dos empregados, classe dos desempregados, classe dos trabalhadores informais, etc.), cujo significado depende da interpretação que o especialista na aplicação dá a cada uma delas. Estes rótulos normalmente não estão na forma numérica, de modo que para serem utilizados para treinar a rede MLP eles devem ser convertidos para a forma numérica. A este procedimento dá-se o nome de codificação da saída da rede.

A codificação mais comum define como vetor de saídas desejadas um vetor binário de comprimento unitário; ou seja, apenas uma componente deste vetor terá o valor "1", enquanto as outras terão o valor "0" (ou -1). A dimensão do vetor de saídas desejadas corresponde ao número de classes do problema em questão. Usando esta codificação define-se automaticamente um neurônio de saída para cada classe. Por exemplo, se existem três classes possíveis, existirão três neurônios de saída, cada um representando uma classe. Como um vetor de entrada não pode pertencer a mais de uma classe ao mesmo tempo, o vetor de saídas desejadas terá valor 1 (um) na componente correspondente à classe deste vetor, e 0 (ou -1) para as outras componentes. Por exemplo, se o vetor de entrada $\mathbf{x}(n)$ pertence à classe 1, então seu vetor de saídas desejadas é $\mathbf{d}(n) = [1\ \ 0\ \ 0]^T$. Se o vetor $\mathbf{x}(n)$ pertence à classe 2, então seu vetor de saídas desejadas é $\mathbf{d}(n) = [0\ \ 1\ \ 0]^T$ e assim por diante para cada exemplo de treinamento.

Número de neurônios na camada escondida (q): Encontrar o número ideal de neurônios da camada escondida não é uma tarefa fácil porque depende de uma série de fatores, muito dos quais não temos controle total. Entre os fatores mais importantes podemos destacar os seguintes:

1. Quantidade de dados disponíveis para treinar e testar a rede.
2. Qualidade dos dados disponíveis (ruidosos, com elementos faltantes, etc.)
3. Número de parâmetros ajustáveis (pesos e limiares) da rede.
4. Nível de complexidade do problema (não-linear, descontínuo, etc.).

O valor de q é geralmente encontrado por tentativa-e-erro, em função da capacidade de *generalização* da rede (ver definição logo abaixo). Grosso modo, esta propriedade avalia o desempenho da rede neural ante situações não-previstas, ou seja, que resposta ela dá quando novos dados de entrada forem apresentados. Se muitos neurônios existirem na

camada escondida, o desempenho será muito bom para os dados de treinamento, mas tende a ser ruim para os novos dados. Se existirem poucos neurônios, o desempenho será ruim também para os dados de treinamento. O valor ideal é aquele que permite atingir as especificações de desempenho adequadas tanto para os dados de treinamento, quanto para os novos dados.

Existem algumas fórmulas heurísticas (*ad hoc*) que sugerem valores para o número de neurônios na camada escondida da rede MLP, porém estas regras devem ser usadas apenas para dar um valor inicial para q. O projetista deve sempre treinar e testar várias vezes uma dada rede MLP para diferentes valores de q, a fim de se certificar que a rede neural generaliza bem para dados novos, ou seja, não usados durante a fase de treinamento.

Dentre a regras heurísticas citamos a seguir três, que são comumente encontradas na literatura especializada:

Regra do valor médio - De acordo com esta fórmula o número de neurônios da camada escondida é igual ao valor médio do número de entradas e o número de saídas da rede, ou seja:

$$q = \frac{p+M}{2} \tag{B.18}$$

Regra da raiz quadrada - De acordo com esta fórmula o número de neurônios da camada escondida é igual a raiz quadrada do produto do número de entradas pelo número de saídas da rede, ou seja:

$$q = \sqrt{p \cdot M} \tag{B.19}$$

Regra de Kolmogorov De acordo com esta fórmula o número de neurônios da camada escondida é igual a duas vezes o número de entradas da rede adicionado de 1, ou seja:

$$q = 2p+1 \tag{B.20}$$

Perceba que as regras só levam em consideração características da rede em si, como número de entradas e número de saídas, desprezando informações úteis, tais como número de dados disponíveis para treinar/testar a rede e o erro de generalização máximo aceitável. Uma regra que define um valor inferior para q levando em consideração o número de dados de treinamento/teste é dada por:

$$q \geq \frac{N-1}{p+2} \tag{B.21}$$

A regra geral que se deve sempre ter em mente é a seguinte: *devemos sempre ter muito mais dados que parâmetros ajustáveis*. Assim, se o número total de parâmetros (pesos + limiares) da rede é dado por $Z = (p+1)\cdot q + (q+1)\cdot M$, então devemos sempre tentar obedecer à seguinte relação:

$$N \gg Z \tag{B.22}$$

Um refinamento da Equação (B.22), proposto por Baum & Haussler (1991), sugere que a relação entre o número total de parâmetros da rede (Z) e a quantidade de dados disponíveis (N) deve obedecer à seguinte relação:

$$N > \frac{Z}{\varepsilon} \tag{B.23}$$

em que $\varepsilon > 0$ é o erro percentual máximo aceitável durante o teste da rede; ou seja, se o erro aceitável é 10%, então $\varepsilon = 0{,}1$. Para o desenvolvimento desta equação, os autores assumem que o erro percentual durante o treinamento não deverá ser maior que $\varepsilon/2$.

Para exemplificar, assumindo que $\varepsilon = 0{,}1$, então temos que $N > 10Z$. Isto significa que para uma rede de Z parâmetros ajustáveis, devemos ter uma quantidade dez vezes maior de padrões de treinamento.

Note que se substituirmos Z na Equação (B.23) e isolarmos para q, chegaremos à seguinte expressão que fornece o valor aproximado do número de neurônios na camada oculta:

$$q \approx \left\lceil \frac{\varepsilon N - M}{p + M + 1} \right\rceil \tag{B.24}$$

em que $\lceil u \rceil$ denota o menor inteiro maior que u.

A Equação (B.24) é bastante completa, visto que leva em consideração não só aspectos estruturais da rede MLP (número de entradas e de saídas), mas também o erro máximo tolerado para teste e o número de dados disponíveis. Portanto, seu uso é bastante recomendado.

Funções de ativação (ϕ_i) e (ϕ_k): Em tese, cada neurônio pode ter a sua própria função de ativação, diferente de todos os outros neurônios. Contudo, para simplificar o projeto da rede é comum adotar a mesma para todos os neurônios. Em geral, escolhe-se a função logística ou a tangente hiperbólica para os neurônios da camada escondida. Aquela que for escolhida para estes neurônios será adotada também para os neurônios da camada de saída. Em algumas aplicações é comum adotar uma função de ativação linear para os neurônios da camada de saída, ou seja, $\phi_k(u_k(n)) = C_k \cdot u_k(n)$, onde C_k é uma constante (ganho)

positiva. Neste caso, tem-se que $\phi_k'(u_k(n)) = C_k$. O fato de $\phi_k(u_k(n))$ ser linear não altera o poder computacional da rede, o que devemos lembrar sempre é que os neurônios da camada escondida devem ter uma função de ativação não-linear, obrigatoriamente.

Critério de Parada e Convergência: A convergência da rede MLP é, em geral, avaliada com base nos valores do erro médio quadrático (ε_{epoca}) por época de treinamento:

$$\varepsilon_{epoca} = \frac{1}{N}\sum_{n=1}^{N}\varepsilon(n) = \frac{1}{2N}\sum_{n=1}^{N}\sum_{k=1}^{n} e_k^2(n) \tag{B.25}$$

Por outro lado, quando se utiliza a rede para classificar padrões, o desempenho da mesma é avaliado pela *taxa de acerto na classificação*, definida como:

$$P_{epoca} = \frac{\text{Número de vetores classificados corretamente}}{\text{Número de total de vetores}} \tag{B.26}$$

O gráfico $\varepsilon_{epoca} \times$ número de épocas ou o $P_{epoca} \times$ número de epócas é chamado de *Curva de Aprendizagem* da rede neural.

Em geral, o treinamento da rede neural é interrompido quando ε_{epoca} (ou P_{epoca}) atinge um limite inferior considerado adequado para o problema em questão (por exemplo, $\varepsilon_{epoca} \leq 0,001$ ou $P_{epoca} \approx 0,95$), ou quando o número máximo de épocas permitido é alcançado.

Avaliação da Rede Treinada: Para validar a rede treinada, ou seja, dizer que ela está apta para ser utilizada, é importante testar a sua resposta (saída) para dados de entrada diferentes daqueles vistos durante o treinamento. Estes novos dados podem ser obtidos através de novas medições, o que nem sempre é viável. Durante o teste os pesos da rede não são ajustados.

Para contornar este obstáculo, o procedimento mais comum consiste em treinar a rede apenas com uma parte dos dados selecionados *aleatoriamente*, guardando a parte restante para ser usada para testar o desempenho da rede. Assim, ter-se-á dois conjuntos de dados, um para treinamento, de tamanho $N_1 < N$, e outro de tamanho $N_2 = N - N_1$. Em geral, escolhe-se N_1 tal que a razão N_1/N esteja na faixa de $0,75$ a $0,90$.

Em outras palavras, se $N_1/N \approx 0,75$ tem-se que 75% dos vetores de dados devem ser selecionados aleatoriamente, sem reposição, para serem utilizados durante o treinamento. Os 25% restantes serão usados para testar a rede. O valor de ε_{epoca} calculado com os dados de teste é chamado de *erro de generalização* da rede, pois testa a capacidade da mesma em "extrapolar" o conhecimento aprendido durante o treinamento para novas situações.

É importante ressaltar que, geralmente, o erro de generalização é maior do que o erro de treinamento, pois trata-se de um novo conjunto de dados.

B.5 Dicas para um Bom Projeto da Rede MLP

A seguir são dadas algumas sugestões para aumentar a chance de ser bem-sucedido no projeto de uma rede neural artificial.

Pré-processamento dos pares entrada-saída Antes de apresentar os exemplos de treinamento para a rede MLP é comum mudar a escala original das componentes dos vetores **x** e **d** para a escala das funções de ativação logística (0 e 1) ou da tangente hiperbólica (-1 e 1). As duas maneiras mais comuns de se fazer esta mudança de escala são apresentadas a seguir:

Procedimento 1: Indicado para quando as componentes x_j do vetor de entrada só assumem valores positivos e a função de ativação, $\phi(u)$, é a função logística. Neste caso, aplicar a seguinte transformação a cada componente de **x**:

$$x_j^* = \frac{x_j}{x_j^{max}} \tag{B.27}$$

em que, ao dividir cada x_j pelo seu maior valor $x_j^{max} = \max_{\forall t}\{x_j(n)\}$, tem-se que $x_j^* \in [0,1]$.

Procedimento 2: Indicado para quando as componentes x_j do vetor de entrada assumem valores positivos e negativos, e a função de ativação, $\phi(u)$, é a função tangente hiperbólica. Neste caso, aplicar a seguinte transformação a cada componente de **x**:

$$x_j^* = 2\left(\frac{x_j - x_j^{min}}{x_j^{max} - x_j^{min}}\right) - 1 \tag{B.28}$$

em que $x_j^{min} = \min_{\forall t}\{x_j(n)\}$ é o menor valor de x_j. Neste caso, tem-se que $x_j^* \in [-1,+1]$.

Os dois procedimentos descritos acima também devem ser igualmente aplicados às componentes d_k dos vetores de saída, **d**, caso estes possuam amplitudes fora da faixa definida pelas funções de ativação.

Taxa de aprendizagem variável: Nas Equações (B.16) e (B.17) é interessante que se use uma taxa de aprendizagem variável no tempo, $\alpha(n)$, decaindo até um valor bem baixo com o passar das iterações, em vez de mantê-la fixa por toda a fase de treinamento. Duas opções

são dadas a seguir:

$$\alpha(n) = \alpha_0\left(1-\frac{t}{t_{max}}\right), \quad \text{Decaimento linear} \tag{B.29}$$

$$\alpha(n) = \frac{\alpha_0}{1+t}, \quad \text{Decaimento exponencial} \tag{B.30}$$

em que α_0 é o valor inicial da taxa de aprendizagem e t_{max} é o número máximo de iterações:

$$t_{max} = \text{Tamanho do conjunto de treinamento} \times \text{Número máximo de épocas} \tag{B.31}$$

A ideia por trás das duas equações anteriores é começar com um valor alto para α, dado por $\alpha_0 < 0,5$, e terminar com um valor bem baixo, da ordem de $\alpha \approx 0,01$, a fim de estabilizar o processo de aprendizado.

Termo de momento: Também nas Equações (B.16) e (B.17) é interessante que se use um termo adicional, chamado *termo de momento*, cujo objetivo é tornar o processo de modificação dos pesos mais estável. Com este termo, as Equações (B.16) e (B.17) passam a ser escritas como:

$$w_{ij}(n+1) = w_{ij}(n) + \alpha\delta_i(n)x_j(n) + \eta\Delta w_{ij}(n-1) \tag{B.32}$$

$$m_{ki}(n+1) = m_{ki}(n) + \alpha\delta_k(n)z_i(n) + \eta\Delta m_{ki}(n-1) \tag{B.33}$$

em que $\Delta w_{ij}(n-1) = w_{ij}(n) - w_{ij}(n-1)$ e $\Delta m_{ki}(n-1) = m_{ki}(n) - m_{ki}(n-1)$. A constante η é chamada *fator de momento*. Enquanto α deve ser mantida abaixo de 0,5 por questões de estabilidade do aprendizado, o fator de momento é mantido em geral em valores na faixa [0,5 - 1]. É importante destacar que resultados teóricos recentes demostram que a introdução do termo de momento equivale, na verdade, a uma versão estacionária do método do gradiente conjugado (BHAYA; KASZKUREWICZ, 2004).

Função Tangente Hiperbólica: Tem sido demonstrado empiricamente, ou seja, através de simulação computacional que o processo de treinamento converge mais rápido quando se utiliza a função de ativação tangente hiperbólica do que quando se usa a função logística. A justificativa para isto está no fato da tangente hiperbólica ser uma função ímpar, ou seja, $\phi(-u_i) = -\phi(u_i)$. Daí sugere-se utilizar a função tangente hiperbólica sempre que o problema permitir.

Limites menores que os assintóticos: É interessante notar que os valores limites 0 e 1 para a função logística, ou (-1 e $+1$) para a função tangente hiperbólica são valores assintóticos, ou seja, nunca são alcançados na prática. Assim, ao tentarmos forçar a saída rede neural para estes valores assintóticos, os pesos sinápticos, w_{ij} e m_{ki} tendem a assumir valores

absolutos muito altos, ou seja, $w_{ij} \to \infty$ e $m_{ki} \to \infty$.

Para evitar este problema, sugere-se elevar de um valor bem pequeno $0 < \varepsilon \ll 1$ o limite inferior de $\phi(\cdot)$ e diminuir deste mesmo valor o limite superior de $\phi(\cdot)$. Assim, teríamos a seguinte alteração:

$$-1 \quad \to \quad \varepsilon - 1 \tag{B.34}$$

$$0 \quad \to \quad \varepsilon \tag{B.35}$$

$$1 \quad \to \quad 1 - \varepsilon \tag{B.36}$$

É comum escolher valores dentro da faixa $\varepsilon \in [0,01 - 0,05]$.

Classificação de padrões: Quando se treina a rede MLP para classificar padrões é comum usar a codificação de saída descrita na Seção B.4, em que na especificação do vetor de saídas desejadas assume-se o valor de saída unitário (1) para o neurônio que representa a classe e nulo (0) para os outros neurônios. Conforme dito no item anterior este valores são assintóticos e portanto, dificilmente serão observados durante a fase de teste.

Assim para evitar ambiguidades durante o cálculo da taxa de acerto P_{epoca} durante as fases de treinamento e teste define-se como a classe do vetor de entrada atual, $\mathbf{x}(n)$, como sendo a classe representada pelo neurônio que tiver maior valor de saída. Em palavras, podemos afirmar que se o índice do neurônio de maior saída é c, ou seja

$$y_c(n) = \max_{\forall k} \{y_k(n)\} \tag{B.37}$$

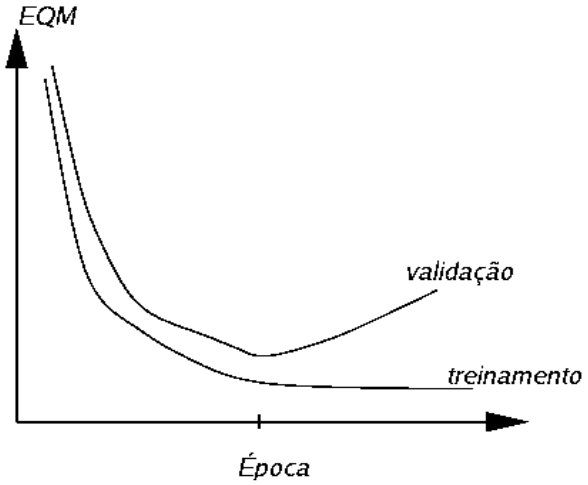
então a Classe de $\mathbf{x}(n)$ é a Classe c.

Generalização: A rede MLP é um dos algoritmos de aproximação mais poderosos que existem, conforme atestado por uma gama de teoremas matemáticos. Contudo, todo este poder computacional, se não for utilizado adequadamente, não necessariamente implica em uma rede que seja capaz de generalizar adequadamente.

Por generalização adequada entende-se a habilidade da rede em utilizar o conhecimento armazenado nos seus pesos e limiares para gerar saídas coerentes para novos vetores de entrada, ou seja, vetores que não foram utilizados durante o treinamento. A generalização é considerada boa quando a rede, durante o treinamento, foi capaz de capturar (aprender) adequadamente a relação entrada-saída do mapeamento de interesse.

O bom treinamento de uma rede MLP, de modo que a mesma seja capaz de lidar com novos vetores de entrada, depende de uma série de fatores, dentre os quais podemos listar os seguintes

Figura 39 – Curvas de aprendizagem para conjuntos de estimação e validação.



1. Excesso de graus de liberdade de uma rede MLP, na forma de elevado número de parâmetros ajustáveis (pesos e limiares).
2. Excesso de parâmetros de treinamento, tais como taxa de aprendizagem, fator de momento, número de camadas ocultas, critério de parada, dimensão da entrada, dimensão da saída, método de treinamento, separação dos conjuntos de treinamento e teste na proporção adequada, critério de validação, dentre outros.

Em particular, no que tange ao número de parâmetros ajustáveis, uma das principais consequências de um treinamento inadequado é a ocorrência de um subdimensionamento ou sobredimensionamento da rede MLP, o que pode levar, respectivamente, à ocorrência de *underfitting* (subajustamento) ou *overfitting* (sobreajustamento) da rede aos dados de treinamento. Em ambos os casos, a capacidade de generalização é ruim.

Dito de maneira simples, o subajuste da rede aos dados ocorre quando a rede não tem poder computacional (i.e. neurônios na camada oculta) suficiente para aprender o mapeamento de interesse. No ouro extremo está o sobreajuste, que ocorre quando a rede tem neurônios ocultos demais (dispostos em uma ou duas camadas ocultas) e passar a memorizar os dados de treinamento. O ajuste ideal é obtido para um número de camadas ocultas e neurônios nestas camadas que confere à rede um bom desempenho durante a fase de teste, quando sua generalização é avaliada.

Uma das técnicas mais utilizadas para treinar a rede MLP, de modo a garantir uma boa generalização é conhecido como parada prematura (*early stopping*). Para este método funcionar, primeiramente devemos separar os dados em dois conjuntos, o de treinamento e

o de teste, conforme mencionado na Seção B.4. Em seguida, o conjunto de treinamento é ainda dividido em duas partes, uma para estimação dos parâmetros da rede propriamente dito e outra para validação durante o treinamento. O conjunto de validação deve ser usado de tempos em tempos (por exemplo, a cada 5 épocas de treinamento) para cálculo do erro quadrático médio de generalização. Durante a validação, os pesos e limiares da rede não são ajustados.

A ideia do método da parada prematura é interromper o treinamento a partir do momento em que o erro quadrático médio calculado para o conjunto de validação assumir uma tendência de crescimento. Argumenta-se que esta tendência de crescimento do erro é um indicativo de que a rede está começando a se especializar demais nos dados usados para estimação dos parâmetros. A avaliação final da generalização é feita usando-se o conjunto de teste. O método de parada prematura pode ser melhor visualizado através das curvas de aprendizagem do erro quadrático médio para o conjunto de estimação e para o conjunto de validação, conforme mostradas na Figura 39.

APÊNDICE C – MÁQUINAS DE VETOR SUPORTE

Máquinas de vetor suporte (*support vector machines - SVM*) são classificadores de padrões que se baseiam na teoria de aprendizado estatístico (VAPNIK, 1995; VAPNIK, 1998) que, grosso modo, consiste na filosofia de projeto que leva em consideração a minimização do erro estrutural e não apenas a minimização do erro empírico[1], como ocorre para as redes MLP e RBF.

Antes de prosseguir, no intuito de melhor compreender classificadores SVM, deve-se ter em mente algumas definições importantes, a saber, generalização, dimensão VC e risco estrutural.

- **Generalização**: Este termo, emprestado da psicologia, é usado aqui para qualificar o aprendizado do classificador. Sua capacidade de generalização é dita tão melhor quanto maior for a taxa de acerto para dados de teste (HAYKIN, 1994).
- **Dimensão VC**: É uma medida da capacidade ou poder de discriminação da família de funções que o algoritmo de aprendizado gerou após a etapa de treinamento. Em outras palavras, se um conjunto genérico contendo N vetores puder ser rotulado de 2^N modos diferentes e, para cada uma destas possibilidades, existir uma função, dentre as geradas pelo algoritmo, que possa discriminá-las corretamente diz-se, então, que este conjunto de vetores pode ser separado por este algoritmo e que a dimensão VC é, portanto, N.
- **Minimização do Risco Estrutural**: Esta é uma indução baseada no fato de a taxa de erro do algoritmo de aprendizagem sobre os dados de teste ser limitada pela soma da taxa de erro de treinamento e um termo que depende da dimensão VC (SHAWE-TAYLOR *et al.*, 1998).

Definidos estes conceitos passa-se, nas linhas seguintes, a uma breve descrição da teoria básica que caracteriza as máquinas de vetor suporte.

C.1 Teoria Básica para SVM

Chama-se o hiperplano

$$\mathbf{w}_o \cdot \mathbf{x} + b_o = 0 \tag{C.1}$$

de ótimo se ele separa o conjunto de treinamento $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\}$ e se a margem entre o hiperplano e o vetor de treinamento mais próximo é máxima. Isto significa que o

[1] Erro quadrático médio calculado para os vetores de treinamento.

hiperplano ótimo tem que satisfazer as desigualdades

$$\mathbf{y}_i(\mathbf{w}_o \cdot \mathbf{x} + b_o) \geq 1, \quad i = 1, \ldots, m. \tag{C.2}$$

e minimizar o funcional

$$R(\mathbf{w}) = \mathbf{w} \cdot \mathbf{w} \tag{C.3}$$

Este problema de otimização quadrática pode ser resolvido no espaço dual dos multiplicadores de Lagrange (BAZARAA *et al.*, 1993). Assim, constrói-se o lagrangiano da seguinte forma

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \tag{C.4}$$

e busca-se minimizá-lo com relação a $\mathbf{w}$ e b e maximizá-lo com relação aos multiplicadores

$$\alpha_i \geq 0, \quad i = 1, \ldots, m. \tag{C.5}$$

Deste modo, ao se minimizar (C.4) com relação a $\mathbf{w}$ a b obtém-se, respectivamente, as equações

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{C.6}$$

e

$$\sum_{i=1}^{m} \alpha_i y_i = 0. \tag{C.7}$$

Substituindo-se (C.6) no Lagrangiano (C.4) e considerando (C.7), obtém-se o funcional

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j. \tag{C.8}$$

Ao se maximizar esta equação (C.8) com relação ao parâmetro α e respeitando-se as restrições (C.5) e (C.7) obtém-se a solução ótima $\alpha^o = (\alpha_1^o, \alpha_2^o, \ldots, \alpha_m^o)$ a qual, por conseguinte, especifica os coeficientes para o hiperplano ótimo desejado

$$\mathbf{w}_o = \sum_{i=1}^{m} \alpha_i^o y_i \mathbf{x}_i \tag{C.9}$$

e

$$\sum_{i=1}^{m} \alpha_i^o y_i \mathbf{x}_i \cdot \mathbf{x} + b_o = 0, \tag{C.10}$$

em que b_o é escolhido de modo a maximizar a margem de separação hiperplano-vetor mais próximo. É importante ressaltar que a solução ótima satisfaz as condições de Kuhn-Tucker

$$\alpha_i^o[y_i(\mathbf{w}_o \cdot \mathbf{x}_i + b_o) - 1] = 0. \tag{C.11}$$

E supondo que $\alpha_i^o \neq 0$, tem-se que

$$y_i(\mathbf{w}_o \cdot \mathbf{x}_i + b_o) = 1. \tag{C.12}$$

Os vetores $\mathbf{x}_i$ que satisfazem (C.12) denominam-se, então, *vetores suporte*. E a norma do vetor $\mathbf{w}_o$ define a margem ρ entre o hiperplano de separação ótima e os vetores suporte

$$\rho = \frac{1}{\|\mathbf{w}_o\|} \tag{C.13}$$

Portanto, levando-se em conta as equações (C.7) e (C.11), obtém-se

$$\frac{1}{\rho^2} = \mathbf{w}_o \cdot \mathbf{w}_o = \sum_{i=1}^{m} y_i \alpha_i^o \mathbf{w}_o \cdot \mathbf{x}_i = \sum_{i=1}^{m} y_i \alpha_i^o (\mathbf{w}_o \cdot \mathbf{x}_i + b_o) = \sum_{i=1}^{m} \alpha_i^o \tag{C.14}$$

Por outro lado, para os casos de não-separabilidade do conjunto de treinamento, uma alternativa é a introdução de variáveis de flexibilização ξ_i, de modo que o funcional (C.3) assume a forma

$$R(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i, \tag{C.15}$$

em que C é um parâmetro de regularização. Sujeito às restrições

$$y_i(\mathbf{w}_o \cdot \mathbf{x}_i + b_o) \geq 1 - \xi_i, \tag{C.16}$$

$$\text{e} \quad \xi_i \geq 0, \tag{C.17}$$

o Lagrangiano deste problema assume a seguinte forma

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i] + C \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \nu_i \xi_i. \tag{C.18}$$

Nestas condições, deve-se minimizar a Eq. (C.18) com relação a $\mathbf{w}$, b e ξ_i e maximizá-lo com relação aos multiplicadores $\alpha_i \geq 0$ e $\nu_i \geq 0$.

Verifica-se que o resultado da minimização com relação a $\mathbf{w}$ e b conduz às restrições (C.6) e (C.7) e o resultado da minimização com relação a ξ_i implica na nova restrição

$$\alpha_i + \nu_i = C. \tag{C.19}$$

Considerando que $\nu_i \geq 0$, obtém-se

$$0 \leq \alpha_i \leq C. \tag{C.20}$$

Quando se utiliza (C.16) e (C.17) no Lagrangiano (C.18) tem-se que, para determinar o hiperplano ótimo, a maximização do funcional (C.8) deve respeitar as restrições (C.7) e (C.20).

Para o caso de não-separabilidade, as condições de Kuhn-Tucker

$$\alpha_i^o[y_i(\mathbf{w}_o \cdot \mathbf{x}_i + b_o) - 1 + \xi_i] = 0 \quad \text{e} \quad \nu_i \xi_i = 0 \tag{C.21}$$

devem ser satisfeitas. E, assim como ocorre para o caso anterior, os vetores $\mathbf{x}_i$, que correspondem aos α_i^o não nulos, são denominados vetores suporte. Neste caso, decorre que

$$y_i(\mathbf{w}_o \cdot \mathbf{x}_i + b_o) = 1 - \xi_i, \tag{C.22}$$

e, portanto, pelas condições (C.19) e (C.21) segue que se $\xi_i > 0$, então, $\nu = 0$ e $\alpha_i = C$. Neste ponto, pode-se distinguir entre dois tipos de vetores suporte: os vetores para os quais $0 < \alpha_i^o < C$ e aqueles para os quais $\alpha_i^o = C$.

Ao se projetar um classificador SVM, usualmente os vetores de entrada $\mathbf{x} \in \mathscr{X}$ são mapeados em um espaço aumentado ou espaço de características, $\phi(\mathbf{x}) \in \mathscr{F}$, com elevada dimensão onde se constroem os hiperplanos de separação ótima. O produto de dois vetores quaisquer neste espaço pode, então, assumir a forma generalizada

$$\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j), \tag{C.23}$$

em que $k(\mathbf{x}_i, \mathbf{x}_j)$ é conhecida como função núcleo, ou simplesmente (*kernel*), que atende as condições de Mercer[2]. Na Tabela 5 algumas opções para a função de kernel são mostradas.

C.2 Projeto de Classificadores SVM

A construção de um classificador SVM exige, em geral, que se resolvam problemas de otimização quadrática, os quais são fortemente dependentes do número e da dimensão dos vetores de treinamento (SMOLA *et al.*, 2000). Neste trabalho, nenhuma técnica adicional de aceleração da resolução de problemas de otimização foi usada. Aos interessados neste tipo de técnica de aceleração recomenda-se a leitura das referências (STITSON; WESTON, 1996) e (OSUNA *et al.*, 1997). Todos os problemas de otimização foram resolvidos com base

[2] Ser uma função definida positiva e simétrica (MERCER, 1909; AIZERMAN *et al.*, 1964; BOSER *et al.*, 1992).

Tabela 5 – Funções de Kernel Típicas.

$k(\mathbf{x}-\mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|^2)$	Gaussiana RBF
$k(\mathbf{x}-\mathbf{y}) = (\|\mathbf{x}-\mathbf{y}\| + c^2)^{\frac{1}{2}}$	Multiquadrática
$k(\mathbf{x}-\mathbf{y}) = (\|\mathbf{x}-\mathbf{y}\| + c^2)^{-\frac{1}{2}}$	Multiquadrática Inversa
$k(\mathbf{x}-\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^{2n+1}$ $k(\mathbf{x}-\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^{2n} ln(\|\mathbf{x}-\mathbf{y}\|)$	Splines
$k(\mathbf{x},\mathbf{y}) = \tanh(\|\mathbf{x}\cdot\mathbf{y}\| - \theta)$	Tangente Hiperbólica
$k(\mathbf{x},\mathbf{y}) = (1 + \|\mathbf{x}\cdot\mathbf{y}\|)^d$	Polinomial de grau d
$k(x,y) = B_{2n+1}(x-y)$	B - splines
$k(x,y) = \frac{\sin(d+\frac{1}{2})(x-y)}{\sin\frac{(x-y)}{2}}$	Polinomial trigonométrico de grau d

nas implementações propostas por (GUNN, 1998) com o uso do método das restrições ativas (BAZARAA *et al.*, 1993).

Outro aspecto importante referente a aplicação de SVM aos problemas de classificação aqui encontrados diz respeito ao fato deste classificador ter sido originalmente projetado para tarefas de classificação binária (CORTES; VAPNIK, 1995). Surge, então, mais um problema a superar: "como adequá-lo a problemas multiclasses?". Para contornar esta situação, no entanto, desenvolveram-se formulações que possibilitam aplicá-lo a problemas multiclasses (FERRIS; MUNSON, 2003; ABE, 2005) e, dentre estas, duas abordagens foram utilizadas nesta dissertação e serão descritas a seguir.

Abordagem Um Contra Todos (*One Against All* - OA)

Este método foi um dos primeiros a tratar o problema de classificação multiclasses (BOTTOU *et al.*, 1994). Ele caracteriza-se por construir c modelos SVM, em que c é o número de classes dos dados. No treinamento do i-ésimo classificador SVM, os exemplos da i-ésima classe recebem rótulos positivos (+1) enquanto que o restante dos dados recebem rótulo negativos (-1).

Deste modo, dado $\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\ldots,(\mathbf{x}_i,y_i),\ldots,(\mathbf{x}_m,y_m)\}$, onde $\mathbf{x}_i \in \mathscr{R}^n, i = 1,2,\ldots,m$ e $y_i \in \{1,2,\ldots,c\}$, o i-ésimo classificador SVM resolve o seguinte problema

$$\min_{\mathbf{w}^i,b^i,\xi^i} \quad \frac{1}{2}(\mathbf{w}^i)^T\mathbf{w}^i + C\sum_{j=1}^{m}\xi_j^i, \quad \text{sujeito às seguintes restrições:}$$
$$(\mathbf{w}^i)^T\phi(\mathbf{x}_j) + b^i \geq 1 - \xi_j^i, \quad \text{se } y_j = i, \tag{C.24}$$
$$(\mathbf{w}^i)^T\phi(\mathbf{x}_j) + b^i \leq \xi_j^i - 1, \quad \text{se } y_j \neq i,$$
$$\xi_j^i \geq 0, \quad j = 1,2,\ldots,m.$$

Minimizar $\frac{1}{2}(\mathbf{w}^i)^T\mathbf{w}^i$ significa maximizar $2/\|\mathbf{w}^i\|$. Quando os dados não são linear-

mente separáveis, o termo $C\sum_{j=1}^{m}\xi_j^i$ busca reduzir o número de erros durante o treinamento.

Após resolver (C.24), dispõem-se de c funções de decisão

$$\begin{pmatrix} (\mathbf{w}^1)^T\phi(\mathbf{x})+b^1 \\ (\mathbf{w}^2)^T\phi(\mathbf{x})+b^2 \\ \vdots \\ (\mathbf{w}^i)^T\phi(\mathbf{x})+b^i \\ \vdots \\ (\mathbf{w}^c)^T\phi(\mathbf{x})+b^c \end{pmatrix}.$$

Diz-se então que $\mathbf{x}$ pertence a classe de cuja função de decisão apresente o maior valor, ou seja

$$\text{classe}(\mathbf{x}) \equiv \arg\max_{i=1,2,\ldots,c}((\mathbf{w}^i)^T\phi(\mathbf{x})+b^i). \tag{C.25}$$

Abordagem Um Contra Um (*One Against One* - OO)

Este método foi proposto inicialmente por (KNERR *et al.*, 1990). Nesta abordagem são construídos $c(c-1)/2$ classificadores em que cada um é treinado sobre duas classes. O problema que cada SVM resolve é formulado como

$$\begin{aligned} min_{\mathbf{w}^{ij},b^{ij},\xi^{ij}} \quad & \frac{1}{2}(\mathbf{w}^{ij})^T\mathbf{w}^{ij}+C\sum_t^m\xi_t^{ij} \\ (\mathbf{w}^{ij})^T\phi(\mathbf{x}_t)+b^{ij} &\geq 1-\xi_t^{ij}, \quad se\ \ y_t=i, \\ (\mathbf{w}^{ij})^T\phi(\mathbf{x}_t)+b^{ij} &\leq \xi_t^{ij}-1, \quad se\ \ y_t=j, \\ \xi_t^{ij} &\geq 0. \end{aligned} \tag{C.26}$$

Durante a etapa de teste, após os $c(c-1)/2$ classificadores serem obtidos, para cada vetor existe um contador que armazena o número de vezes que este foi classificado como pertencente a cada uma das classes existentes. Em outras palavras, se a função de decisão $(\mathbf{w}^{ij})^T\phi(\mathbf{x})+b^{ij}$ indicar o vetor $\mathbf{x}$ como pertencente à i-ésima classe, o número de vitórias desta classe é incrementado, do contrário, incrementa-se o da j-ésima classe. Ao término deste processo, a estratégia de voto majoritário é adotada e o vetor de teste é classificado como pertencendo a classe que apresentar maior número de vitória. Observe que caso ocorra empate entre classes, escolhe-se aleatoriamente uma destas para representar o vetor de teste.

Em termos comparativos, ao levar-se em conta que nesta dissertação cada classe de treinamento apresenta m/c vetores, esta abordagem exige que se resolvam $c(c-1)/2$ problemas

de otimização quadrática, cada um com $2m/c$ variáveis. Na primeira abordagem, no entanto, tem-se c problemas de programação quadrática para se resolver com m variáveis cada.

ANEXO A – INFORMAÇÕES TÉCNICAS SOBRE OS BANCOS DE DADOS UTILIZADOS

Tabela 6 – Informações técnicas sobre os bancos de dados utilizados

Paciente	Nome do Arquivo	Hora		Numero de Convulsões	Tempo da Convulsão (s)		Duração (s)
		Inicio	Fim		Inicio	Fim	
1	chb01_21.edf	07:33:46	08:33:46	1	327	420	93
3	chb03_02.edf	14:23:39	15:23:39	1	731	796	65
8	chb08_02.edf	12:28:57	13:28:57	1	2670	2841	171
8	chb08_05.edf	15:29:14	16:29:14	1	2856	3046	190