UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

Detecção e classificação de arritmias cardíacas utilizando técnicas de aprendizado de máquina

Relatório Final do projeto na modalidade Iniciação Científica submetido à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Projeto FAPESP: 2019/26911-6

Período de Vigência do Projeto: 01/05/2020 a 30/04/2021

Período Coberto por este Relatório: 10/10/2020 a 30/04/2021

Bolsista: Natália Nagata

Orientador: Magno Teófilo Madeira da Silva

Coorientador: Renato Candido

São Paulo
Maio de 2021

Resumo

Este relatório final tem como propósito apresentar o estudo de métodos computadorizados aplicados ao problema de detecção e classificação de arritmias cardíacas por meio da análise do sinal de eletrocardiograma (ECG). Ele se inicia com os objetivos e o cronograma do plano de pesquisa e um resumo do relatório parcial para a comparação das atividades previstas e realizadas no período coberto. Um resumo contendo uma breve explicação de cada atividade realizada é apresentado. Em seguida, as atividades são detalhadas, iniciando-se com as descrições do sistema cardiovascular e do sinal de ECG para o entendimento dos elementos que compõem esse sinal. Também são descritos os bancos de dados disponíveis para o problema de classificação de arritmias e são estudadas as etapas de preparação do sinal para a classificação. Além disso, é realizada uma revisão bibliográfica sobre os diferentes tipos de sinais usados como entrada das redes neurais e sobre os diferentes tipos de classificadores empregados. Redes *perceptron* multicamada (*multilayer perceptron* - MLP), redes convolucionais (*convolutional neural networks* - CNN), redes recorrentes (*recurrent neural network* - RNN) e análise de discriminantes lineares (*linear discriminant analysis* - LDA) são desenvolvidas para o problema proposto e as simulações realizadas são descritas. Por fim, os resultados são comparados com os da literatura. O relatório termina com as conclusões do projeto. Cabe observar que os conceitos fundamentais de redes MLP, de redes CNN, de redes RNN, da técnica de LDA, o problema da detecção de arritmias em uma classificação binária e o efeito das entradas na rede neural MLP foram organizados em apêndices. A partir dos resultados desse projeto, um artigo foi aceito no *Simpósio Internacional de Iniciação Científica e Tecnológica da USP* e dois outros artigos foram submetidos, um para o *Simpósio Brasileiro de Telecomunicações e Processamento de Sinais* e outro para a *Sociedade Brasileira para o Progresso da Ciência*. Esses artigos encontram-se anexos ao relatório.

Sumário

1 Resumo do Plano de Pesquisa e do Relatório Parcial

Neste Capítulo, descrevem-se os objetivos e o cronograma de atividades do plano de pesquisa inicial. Em seguida, um resumo das atividades e dos principais resultados do Relatório Parcial é apresentado.

1.1 Objetivos

Na literatura, muitos métodos computadorizados para detecção automática de arritmias cardíacas a partir de sinais de eletrocardiograma (ECG) foram propostos. O diagnóstico dessas doenças, em geral, demanda muito tempo do cardiologista e é dificultado pelas características morfológicas variáveis do sinal. No entanto, devido às altas taxas de erro das técnicas computadorizadas e ao crescimento da área de aprendizado de máquina, a pesquisa em automatização do diagnóstico ressurgiu [1–16].

No presente trabalho de Iniciação Científica, propõe-se o uso de redes neurais para a identificação e a classificação de arritmias cardíacas. Particularmente, serão considerados três tipos de rede: (i) perceptron multicamada (*multilayer perceptron* - MLP), (ii) convolucional (*convolutional neural network* - CNN) e (iii) recorrente (*recurrent neural network* - RNN).

Os principais objetivos deste Projeto de Pesquisa são:

1. Realizar um estudo de técnicas de aprendizado de máquina aplicadas a detecção e classificação de arritmias cardíacas;

2. Estudar e implementar diferentes soluções baseadas em redes neurais, focando principalmente nas redes MLP, CNN e RNN;

3. Fazer uma revisão bibliográfica das diferentes entradas do sinal de ECG para treinar as redes neurais;

4. Realizar uma análise comparativa que exponha a metodologia, as técnicas estudadas e os resultados do Projeto.

1.2 Cronograma de Atividades

As atividades a serem realizadas estão listadas a seguir:

1. Estudo de processamento de sinais. A ideia é estudar conceitos básicos de processamento de sinais como: sinais e sistemas de tempo discreto, convolução, filtragem, resposta em frequência e transformada discreta de Fourier. Esses conceitos são importantes para que a aluna consiga entender e implementar soluções para o pré-processamento e extração de características do sinal de ECG, como o realizado pelo algoritmo Pan-Tompkins para detecção do complexo QRS.

2. Estudo de técnicas de aprendizagem de máquina. A ideia é que a aluna tenha contato com diferentes técnicas abordadas no curso *Introduction to Deep Learning* ministrado por Andrew Ng disponível na plataforma *Coursera*.

3. Estudo das possíveis entradas utilizadas na literatura para detecção e classificação de arritmias cardíacas.

4. Estudo e implementação das redes MLP, CNN e RNN para detecção e classificação de arritmias cardíacas. Pretende-se primeiramente implementar as redes em Matlab e posteriormente em Python utilizando o TensorFlow. Uma primeira implementação em Matlab (ambiente que a aluna já está acostumada) é importante para que ela entenda melhor o funcionamento das redes e de seus algoritmos de treinamento. A ideia aqui é que a aluna consiga repetir alguns resultados publicados na literatura para se preparar para fazer uma comparação criteriosa das soluções recentemente propostas.

5. Análise dos resultados.

6. Redação do Relatório Final.

Como a bolsa iniciou-se em maio, o cronograma de atividades previsto foi ajustado, como mostrado na Tabela 1.1.

Tabela 1.1: Cronograma de atividades.

Atividade	2020				2021		
Meses	mai/jun	jul/ago	set/out	nov/dez	jan	fev/mar	abr
1	X	X					
2	X	X	X				
3		X	X				
4		X	X	X	X	X	
5			X	X	X	X	X
6						X	X

1.3 Resumo do Relatório Parcial

No Relatório Parcial, foram estudados os conceitos fundamentais de processamento digital de sinais, como o somatório de convolução e as propriedades de sistemas lineares e invariantes no tempo (SLIT). Em seguida, foram também compreendidas a filtragem de sinais e a representação no domínio da frequência com o aprofundamento na resposta em frequência e na transformada discreta de Fourier. Foram feitos estudos adicionais na teoria de banco de filtros, em particular o *Quadrature Mirror Filter* (QMF), que não estavam previstos no cronograma inicial.

O estudo dos conceitos básicos das redes neurais e de técnicas de Aprendizado de Máquina (*Machine learning* - ML) foi feito por meio do curso *Introduction to Deep Learning* [17] e das referências [18–20]. Compreenderam-se o funcionamento da regressão logística, o método do gradiente estocástico, o algoritmo de retropropagação, as funções de ativação e as funções custo. O aprofundamento desse estudo incluiu redes neurais rasas (com poucas camadas ocultas), redes neurais profundas (com várias camadas), ajuste de hiperparâmetros e técnicas de regularização e de inicialização dos parâmetros. Foram também vistos algoritmos de otimização, como o uso de *mini-batches* e de médias ponderadas aplicadas aos parâmetros para otimizar a velocidade do aprendizado. Nesse contexto, destaca-se a compreensão dos algoritmos Gradiente Descendente com Momento, RMSprop (*Root Mean Square Propagation*) e Adam [21]. Esses conceitos, detalhados no Relatório Parcial, foram colocados no Apêndice A.

Por meio de uma revisão bibliográfica dos métodos computadorizados da literatura, foi escolhido o banco de dados *MIT-BIH Arrhythmia Database* [22,23], que permitiu a comparação dos resultados desta IC com os de diversos trabalhos publicados. Percebeu-se que muitos resultados da literatura utilizam batimentos dos mesmos pacientes nas fases de treinamento e teste. No entanto, esse tipo de separação não é realista e os os resultados de classificação não são confiáveis. Nesta IC, decidiu-se seguir a recomendação da *Association for the Advancement of*

Medical Instrumentation (AAMI), que estabeleceu que os resultados de classificação são mais confiáveis quando os batimentos de um mesmo paciente não são repetidos no treinamento e teste da rede. A mesma recomendação foi seguida pelas referências [24–27] que nortearam o trabalho. A abordagem desse banco de dados e do efeito da divisão dos dados é essencial para a definição do problema. Por isso, foi colocada novamente neste relatório.

Posteriormente, foram estudadas as etapas das técnicas computadorizadas para reconhecimento do sinal de ECG antes da classificação: o pré-processamento, a determinação dos limites do complexo QRS e das ondas P e T e a extração das características. Para a primeira etapa, foi implementado parte do Algoritmo de Pan-Tompkins para a detecção do complexo QRS, estimando-se o período médio de um batimento. Na segunda etapa, foram usados algoritmos de segmentação do sinal de ECG para delimitar alguns eventos importantes do batimento. Por último, investigou-se na literatura as características do sinal de ECG mais utilizadas para o problema de classificação, gerando possíveis entradas da rede. Nessa última etapa, foram feitos estudos adicionais da transformada contínua de wavelet, por uma formulação multi-resolução, e da transformada discreta de wavelet, por um banco de filtros [28], que também não estavam previstos no cronograma. Assim como a compreensão do banco de dados, o entendimento dessas etapas é muito importante para o desenvolvimento dos métodos computadorizados de identificação do sinal de ECG. Portanto, esses conceitos também foram repetidos neste relatório.

No Relatório Parcial, foram feitas duas abordagens para a implementação de redes MLP. Na primeira, cujos resultados são apresentados no Apêndice F, teve-se um primeiro contato com o banco de dados e limitou-se apenas à detecção da presença ou não de arritmias. Na segunda o objetivo foi classificar o sinal entre as cinco classes de arritmia definidas pela AAMI. Para isso, foram simuladas redes MLP com diferentes arquiteturas e entradas. Esses resultados são apresentados no Apêndice G. As simulações com a entrada de 960 amostras do sinal de ECG e a entrada de características extraídas de [24] obtiveram os melhores resultados e por isso essas entradas foram utilizadas para dar continuidade ao projeto durante o período coberto por este relatório.

Por fim, os resultados do Relatório Parcial foram comparados com os da literatura, encontrando-se próximos aos dos trabalhos usados como referência [24–26]. Foi possível concluir que a extração de características na MLP influencia no desempenho da rede, diminuindo a quantidade de parâmetros usados e auxiliando a classificação de determinadas classes de arritmia. Além disso, concluiu-se que a derivação escolhida e o desbalanceamento dos dados também influ-

enciam fortemente a classificação. Os resultados do período coberto pelo Relatório Parcial podem ser acessados pelo GitHub no link `https://github.com/natnagata/Resultados_MLP_Arritmia.git`.

2 Resumo das Atividades Realizadas

As atividades realizadas no período coberto por este relatório estão listadas neste Capítulo.

Estudo e implementação das redes neurais convolucionais

Inicialmente, os conceitos fundamentais das redes neurais convolucionais foram estudados por meio do curso *Introduction to Deep Learning* [17] e das referências [18, 19]. Compreenderam-se as operações de convolução e de subamostragem (*pooling*) que compõem uma extração de características automática realizada pela CNN. No Apêndice C são apresentados alguns desses conceitos. Foram vistas arquiteturas clássicas dessas redes como a *LeNet-5* de LeCun [29] e arquiteturas de redes residuais (*residual neural network* - ResNet). Inspirando-se em [29], redes neurais convolucionais foram implementadas para a identificação de dígitos do banco de dados do MNIST [30], como um primeiro contato com as CNNs.

Em seguida, realizou-se uma revisão bibliográfica do uso de CNNs para a classificação de arritmias. Investigaram-se as arquiteturas de CNN que utilizaram o sinal original do ECG como entrada, aproveitando-se da propriedade de extração de características intrínseca a esse tipo de rede. No entanto, diversos resultados da literatura em que redes neurais convolucionais foram propostas não seguiram as recomendações de divisão dos dados da AAMI. O trabalho procurou analisar a utilidade das CNNs no problema de identificação de arritmias seguindo a divisão realista dos dados. Foram feitas simulações com essas arquiteturas por meio dos módulos *Tensorflow* e *Keras* [31] do *Python*. Os resultados obtidos com as CNNs foram considerados pouco promissores, mostrando novamente como a escolha do paradigma de divisão dos dados é determinante. Esses resultados estão apresentados na Seção 3.5.3.

Estudo e implementação da técnica de análise de discriminantes lineares

A revisão bibliográfica de trabalhos considerados como estado da arte indicou que os melhores resultados da literatura seguindo as recomendações da AAMI foram obtidos com o uso de análise de discriminantes lineares. Uma das principais razões para isso é a compensação do desbalanceamento da quantidade de dados das classes feita por meio de pesos. Assim,

considerou-se interessante aprofundar o entendimento sobre esse método, que é clássico em técnicas de redução de dimensionalidade e tem sido muito usado para problemas de classificação de arritmias [24, 32–35]. Vale ressaltar que essa atividade não estava prevista no cronograma inicial.

Os conceitos sobre essa técnica foram estudados por meio das referências [20, 36, 37] e estão detalhados na Seção E. Foram vistos os discriminantes lineares de Fisher [38] e a sua generalização para mais de duas classes, com a decomposição em autovalores para encontrar a melhor projeção dos dados. Estudaram-se também os modelos com fronteiras de decisão lineares que surgem a partir do pressuposto de distribuição dos dados e a regra de máxima verossimilhança para encontrar os parâmetros dessas distribuições. Por fim, compreendeu-se a modificação da LDA com o uso de pesos para lidar com o desbalanceamento de dados.

Em seguida, LDAs foram implementadas usando a biblioteca *Scikit-learn* [39] para o problema de classificação de arritmias. Inspirando-se em [24], classificadores usados nas duas derivações do sinal foram combinados de modo a integrar os conhecimentos adquiridos por cada um. Os resultados apresentados na Seção 3.5.4 indicaram que essa combinação auxiliou no desempenho da rede. Essa ideia inspirou a pesquisa sobre a combinação da LDA com as redes neurais, o que não estava proposto no plano de pesquisa inicial.

Estudo e implementação das redes neurais recorrentes

Assim como na CNN, os conceitos fundamentais das redes neurais recorrentes foram estudados por meio do curso *Introduction to Deep Learning* [17] e das referências [18, 19]. Foram vistas RNNs clássicas como o modelo de espaço de estados (*State-Space Model*) e a rede *Recurrent Multilayer Perceptron* (RMLP). Essas redes podem ser utilizadas para problemas de processamento de sinais em tempo real e em um fluxo temporal. Porém, como os batimentos usados não são sempre consecutivos, nem pertencem a somente um paciente, foi necessária uma abordagem em que a recorrência ocorre em um determinado número de iterações no tempo.

Após uma revisão bibliográfica do uso de RNNs no problema de classificação de arritmias, verificou-se que o uso de estruturas de *Long Short-Term Memory* (LSTM) [40] foi o mais comum para essa abordagem [41–46]. A LSTM foi então estudada com maiores detalhes, compreendendo-se as funções das portas (*gates*) internas ao bloco que permitem o aprendizado de dependências de longo prazo. Esses estudos estão detalhados na Seção D.

Em seguida, as redes recorrentes com blocos LSTM foram implementadas com os módulos *Tensorflow* e *Keras* [31] do *Python*. As quantidades de neurônios na camada oculta da RNN,

o número de camadas, o *dropout* e o número de épocas foram ajustados usando *grid search* a fim de se obter o melhor desempenho. Os resultados foram descritos no Apêndice 3.5.5.

Implementação de combinações de classificadores

Explorou-se o efeito de combinar diferentes classificadores para aumentar o desempenho do modelo. Para isso, a rede MLP, estudada em maiores detalhes no Relatório Parcial, foi novamente simulada para a obtenção da melhor estrutura considerando o problema de classificação entre quatro classes, N, S, V e F. A arquitetura dessa rede está descrita na Seção 3.5.2.

Dessa forma, na Seção 3.5.6 foram simuladas as combinações MLP–LDA, MLP–RNN, RNN–LDA e MLP–RNN–LDA. Concluiu-se que o uso de modelos combinados ajuda a melhorar o desempenho na identificação de arritmias para determinadas classes. Os programas utilizados e os resultados das simulações podem ser acessados pelo GitHub no link `https://github.com/natnagata/ArrhythmiaClassification`.

Análise dos Resultados

Por fim, dentre as combinações de classificadores, o modelo da MLP–LDA apresentou o melhor desempenho e foi selecionado para ser comparado com os demais trabalhos da literatura. Usaram-se como referência os trabalhos considerados como estado da arte [24, 33–35, 47, 48]. Essa comparação apresentada na Seção 3.5.7 mostra que os resultados obtidos no trabalho alcançaram os valores relatados na literatura e atingiram valores superiores de métricas para determinadas classes de arritmia.

3 Detalhamento das Atividades Realizadas

Parte das atividades listadas no Capítulo 2 são descritas com maiores detalhes neste Capítulo. Na Seção 3.1 descrevem-se os conceitos fundamentais do sistema cardiovascular, da atividade elétrica do coração e do do sinal de ECG. Em seguida, o banco de dados utilizado no projeto é detalhado e aborda-se o paradigma de divisão dos dados dos pacientes. Na Seção 3.4, abordam-se as etapas anteriores à classificação de arritmias, necessárias para a identificação dos elementos do sinal de ECG. Na Seção 3.5, as arquiteturas das redes neurais e da LDA são descritas e os resultados para cada classificador são apresentados. Como o foco do trabalho está nos resultados atingidos para classificação de arritmias cardíacas, os conceitos fundamentais das redes neurais MLP, CNN, RNN e da LDA foram descritos nos Apêndices A, C, D e E, respectivamente. O ajuste de hiperparâmetros das redes neurais foi abordado no Apêndice B.

3.1 O Sistema Cardiovascular e o Sinal de ECG

Antes de estudar qualquer tipo de processamento no sinal ECG ou técnica para a classificação de doenças, é importante primeiro entender a base fisiológica do sinal. O funcionamento do sistema cardiovascular e a representação da atividade elétrica do coração pelo sinal ECG são descritos a seguir.

O coração pode ser interpretado como uma bomba sanguínea distribuidora de nutrientes para todo o corpo humano. Ele é dividido em quatro cavidades distintas: os átrios esquerdo e direito, responsáveis por receber o sangue a partir das veias, e os ventrículos esquerdo e direito, responsáveis por enviar o sangue para as artérias [49]. Nos lados direito e esquerdo, respectivamente, as válvulas tricúspide e mitral separam as duas cavidades, e as válvulas pulmonar e aórtica separam o ventrículo direito e esquerdo da artéria pulmonar e da artéria aorta. [50]

A atividade desse órgão é basicamente composta por duas fases: a sístole, que compreende à contração para o envio do sangue, e a diástole, que corresponde ao relaxamento para o recebimento do sangue. O sangue venoso do corpo chega ao átrio direito por meio das veias

cavas superior e inferior, passando para o ventrículo direito e sendo enviado para os pulmões pela artéria pulmonar. Após a troca gasosa, o sangue arterial chega ao átrio esquerdo pela veia pulmonar, passando para o ventrículo esquerdo e sendo enviado ao corpo pela artéria aorta.

Segundo [51], o entendimento básico do ECG está dividido em 4 fenômenos: a eletrofisiologia de uma única célula, a condução de corrente elétrica através do miocárdio, a fisiologia das estruturas do coração por onde o sinal passa e a tomada das medidas desse sinal na superfície do corpo por meio dos eletrodos.

Assim como em outros tecidos eletricamente ativos, como, por exemplo, o tecido nervoso e o muscular esquelético, as células do tecido muscular cardíaco possuem um potencial de repouso da ordem de -80 mV, ou seja, um valor negativo com relação ao fluido extracelular. Nesse estado, sua membrana plasmática encontra-se polarizada. Porém, a membrana dessas células em particular é capaz de controlar a permeabilidade de íons como sódio e potássio por meio da abertura e do fechamento de canais, que geram mudanças no potencial de ação ao longo do tempo [51]. Dessa forma, a membrana alterna seu estado durante os processos de repolarização e despolarização e permite a condução de sinais elétricos.

Antes do potencial de ação ser propagado, ele deve ser inicializado pelas células de marca-passo, que possuem a propriedade de despolarização espontânea. Essas células encontram-se no nó sinoatrial (SA), no nó atrioventricular (AV) e em certos sistemas de condução especializados dentro dos átrios e ventrículos. Sob condições normais, o nó SA é responsável por determinar a taxa de batimentos cardíacos. Porém, em circunstâncias patológicas, é possível que essa taxa seja determinada pelas células marcapasso inferiores. Uma vez inicializado, o potencial de ação propaga-se pela membrana até a despolarização completa da célula, que o transmite às células adjacentes, como uma corrente contínua [51].

Em um coração normal, o potencial de ação inicia-se no nó SA, é conduzido pelo músculo atrial e passa dos átrios para os ventrículos apenas através do nó AV, cuja função é providenciar um atraso na condução, para que os átrios sejam contraídos completamente antes dos ventrículos iniciarem a contração [51, 52]. Após o nó AV, o impulso cardíaco direciona-se ao feixe de His, dividindo-se entre os ramos esquerdo e direito. O ramo esquerdo divide-se então em dois fascículos, e ambos os ramos se dividem na rede de fibras de Purkinje [52]. Esse processo de condução pode ser visualizado na Figura 3.1.

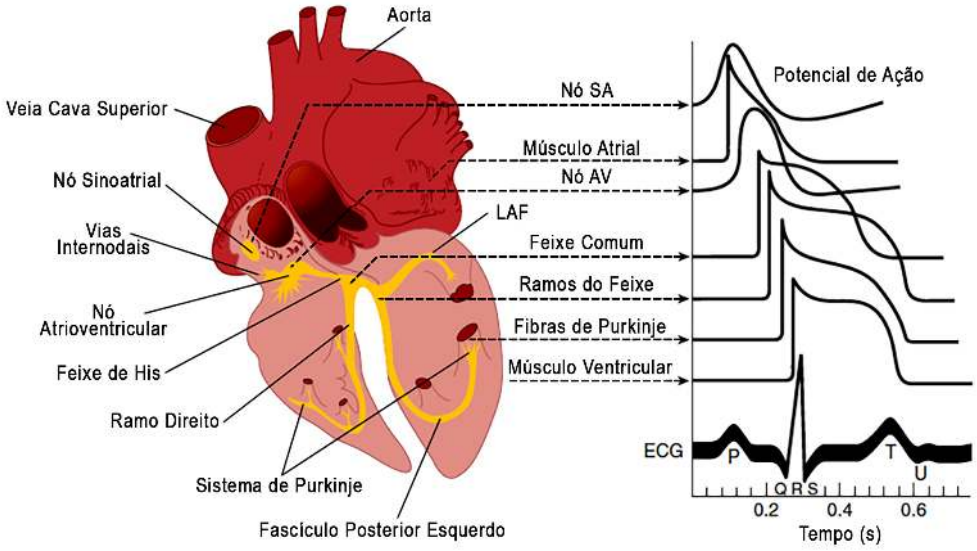


Figura 3.1: Potencial de Ação ao longo do tempo.
Fonte: Adaptado de [53].

Doenças nesse sistema de condução ou em outras regiões podem provocar uma variedade de distúrbios no ritmo do coração. A condução pode ser atrasada em uma região ou completamente bloqueada, e batimentos espontâneos podem surgir em lugares anormais [52].

A gravação dos potenciais superficiais do corpo ao longo do tempo, desenvolvidos em uma derivação, produz o que é conhecido como eletrocardiograma [51]. Esses potenciais podem ser captados por eletrodos posicionados no tórax ou nos membros de um indivíduo. O termo derivação (*lead*) na área médica refere-se ao arranjo particular de eletrodos usados para esse processo, em alguns casos com resistores [52]. No total, 12 derivações são possíveis, cada uma com uma perspectiva elétrica diferente da atividade do coração.

As derivações periféricas são medidas com eletrodos posicionados nos quatro membros e com um terminal central. Elas permitem a visualização da atividade elétrica no plano frontal [49] e podem ser divididas em dois grupos: as bipolares e as aumentadas. As derivações bipolares, I, II, e III, correspondem, respectivamente, à diferença de potencial dos eletrodos entre o braço esquerdo e o direito, entre a perna esquerda e o braço direito, e entre a perna esquerda e o braço esquerdo [51]. Elas correspondem às derivações mais antigas e são conhecidas como o "Triângulo de Einthoven" [52].

Em contrapartida, as derivações aumentadas, aVF, aVR e aVL, fornecem os potenciais de um dado membro com respeito à média dos potenciais dos outros dois membros. Assim, a aVF registra a diferença entre a perna esquerda e a média das derivações dos braços; a aVR, entre o braço direito e a média da perna e do braço esquerdos; e a aVL, entre o braço esquerdo

e a média da perna esquerda e do braço direito [51]. A Figura 3.2 apresenta o Triângulo de Einthoven, com as seis derivações formadas a partir dos membros.

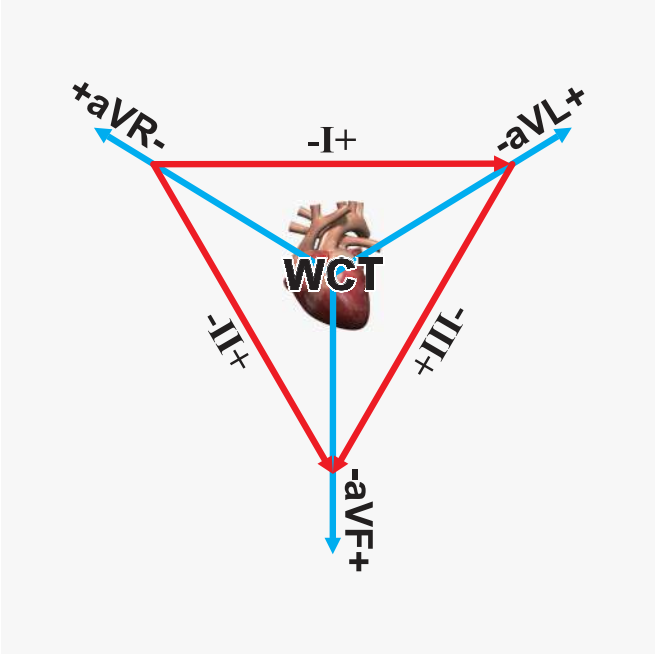


Figura 3.2: Triângulo de Einthoven.

As derivações precordiais, V1, V2, V3, V4, V5 e V6, indicam o potencial de regiões torácicas, em um plano horizontal. Elas representam a diferença de potencial entre cada um dos seis eletrodos, V1 a V6, posicionados no tórax como mostrado na Figura 3.3, e o terminal central. O terminal central, tanto para as derivações periféricas, quanto para as derivações precordiais, é determinado pela média dos potenciais dos três membros usados para as derivações bipolares: braços esquerdo e direito e perna esquerda. Ele é chamado de terminal central de Wilson (*Wilson central terminal* - WCT) e é obtido pela implementação de três resistores iguais em cada um dos eletrodos ligados aos membros, como ilustrado na Figura 3.4.

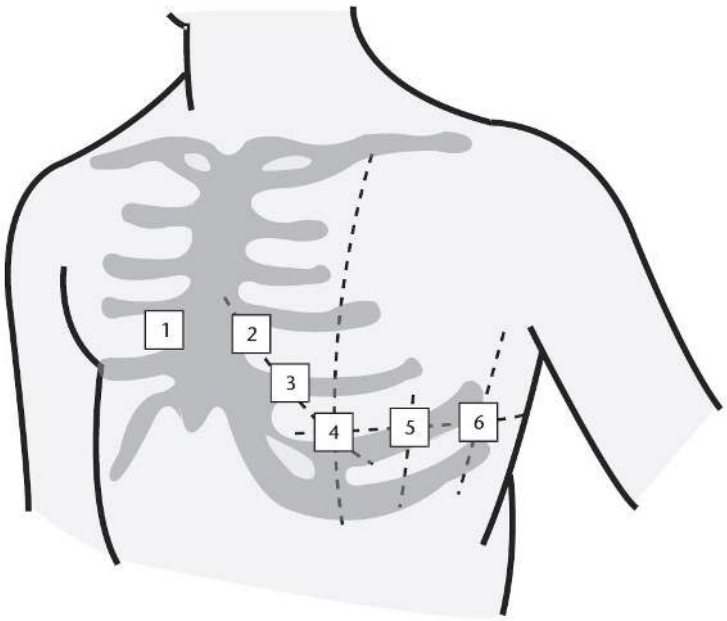


Figura 3.3: Posicionamento das derivações precordiais.
Fonte: [54].

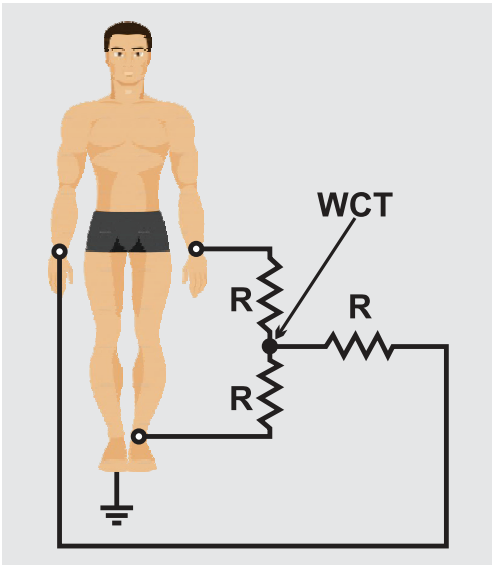


Figura 3.4: Terminal Central de Wilson.

Portanto, para a utilização do sinal de ECG como entrada de uma rede neural, deve-se observar o tipo de derivação na qual o sinal foi captado, pois o padrão da forma de onda é diferente para cada uma.

Um batimento normal no sinal de ECG tem a forma de onda apresentada na Figura 3.5 com seus intervalos e segmentos mais importantes delimitados [52]. O sinal foi gerado pelo programa `ecgsyn.m` [22, 55, 56].

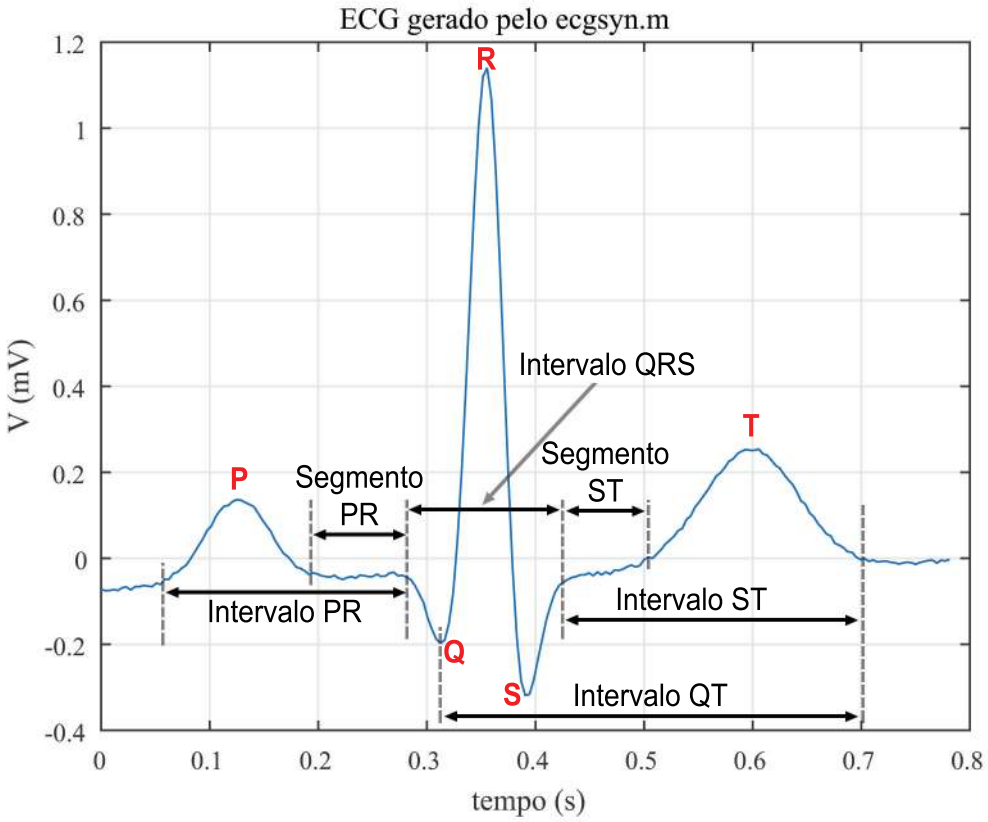


Figura 3.5: Intervalos e segmentos de um ECG.

Os componentes principais do eletrocardiograma são listados a seguir [49, 51, 57]:

- **Onda P:** marca a despolarização atrial associada à contração do átrio e inicia-se com o impulso do nó SA. Possui uma duração de cerca de 110 ms e uma amplitude bem menor do que a do complexo QRS, pois os átrios são menores do que os ventrículos.

- **Segmento PR:** trecho da linha de base, contendo o nível isoelétrico de referência para a amplitude geral do ECG, medido entre a onda P e o complexo QRS. Marca, de forma mais estável, o ponto de 0 V do sinal, uma vez que há uma pausa entre a condução da corrente dos átrios para os ventrículos.

- **Intervalo PR:** corresponde à duração do sinal entre o começo da onda P e o começo do complexo QRS. Representa o tempo necessário para o impulso viajar do nó SA ao ventrículo e possui valores considerado normais entre 120 e 200 ms.

- **Complexo QRS:** é um dos elementos mais importantes para a análise do sinal e corresponde às ondas Q, R e S juntas, que indicam um único evento. Sua largura corresponde ao tempo necessário para o ventrículo despolarizar-se, durando normalmente de 80 a 120 ms.

Quanto menor a taxa dos batimentos cardíacos, maior é o tempo do QRS, devido à diminuição da velocidade de condução através do ventrículo. O complexo geralmente ascende ou decai de 1 a 2 mV em relação à linha de base, e batimentos anormais podem ser muito maiores em amplitude.

- **Segmento ST:** inicia-se no ponto de inflexão após a onda S e termina na onda T, durando normalmente de 60 a 80 ms. Compreende à pausa entre a despolarização e a repolarização ventricular e, em batimentos normais, espera-se que o segmento também esteja no nível isoelétrico.

- **Onda T:** marca a repolarização ventricular, em que o músculo cardíaco se prepara para um novo ciclo do ECG.

- **Intervalo QT:** medido entre o início do complexo QRS e o final da onda T, representa o tempo entre o início da despolarização ventricular e o fim da repolarização ventricular. Esse tempo varia dependendo da taxa de batimentos cardíacos, da idade e do gênero.

As variações das características normais dos elementos do ECG, como as mudanças no tempo de duração e na amplitude, podem indicar distúrbios e, por isso, a análise desses elementos é muito importante. O complexo QRS, em particular, possui um papel de destaque: uma vez detectado, é possível reconhecer os demais trechos e elementos integrantes dos batimentos cardíacos, possibilitando a análise isolada de parâmetros como a distância RR ou o segmento ST [52].

3.2 Os Bancos de Dados de Arritmias

O desenvolvimento de métodos computadorizados para a classificação do sinal de ECG depende do uso de um banco de dados com as anotações desejadas. Nesta seção, os bancos de dados mais utilizados pelos pesquisadores são descritos e aquele escolhido para o projeto é detalhado.

O problema de classificar as arritmias pode ser, genericamente, dividido em duas abordagens: a classificação de batimentos consecutivos e a análise de segmentos de longa duração do sinal de ECG [58]. A última abordagem foi explorada por [9, 59–62] e um exemplo recente dessa proposta pode ser encontrado no desafio proposto pela PhysioNet/Cinc, *Classification of 12-lead ECGs: the PhysioNet - Computing in Cardiology Challenge 2020* [22, 63, 64], para a classificação do sinal das 12 derivações de um paciente em um conjunto de uma ou mais classes de diagnóstico.

A primeira abordagem é a mais pesquisada e é a utilizada neste projeto. A maioria das arritmias manifesta-se como uma sequência de batimentos de intervalos incomuns e a classificação desses batimentos é uma etapa importante para a identificação dessas doenças [65]. Para isso, o banco de dados a ser escolhido deve conter as anotações de cada tipo de batimento.

Alguns dos bancos de dados mais utilizados na literatura e disponibilizados gratuitamente para a classificação e detecção de arritmias cardíacas são:

1. *MIT-BIH Arrhythmia Database* [22, 23] – que contém 48 gravações ambulatoriais de 30 minutos e dois canais, com anotações manuais de cada batimento feitas por cardiologistas, além de anotações adicionais indicando o início e o fim dos diferentes ritmos cardíacos presentes no sinal de ECG. É o banco de dados mais explorado para a detecção e classificação de arritmias.

2. *MIT-BIH Atrial Fibrillation Database* [22, 66] – que contém 25 gravações ambulatoriais de ECG de longa duração e anotações dos ritmos fibrilação atrial, flutter atrial, ritmo juncional, e demais ritmos, além de anotações das localizações de cada complexo QRS, realizadas por um detector automático. Esse banco foi utilizado por [1, 67–69].

3. *CU Ventricular Tachyarrhythmia Database* [22, 70] – composto por 35 gravações de oito minutos de episódios de taquicardia ventricular, flutter ventricular e fibrilação ventricular. Foi explorado recentemente por [1, 2, 71].

4. *UCI Arrhythmia Data Set* [72] – que contém 279 atributos de 452 pacientes diferentes, como, por exemplo, idade, gênero, taxa de batimentos cardíacos, duração e amplitude dos componentes principais do sinal de ECG das 12 derivações, para a classificação entre 16 grupos de arritmias. Esse banco de dados foi utilizado por [73–76].

5. *Physionet Challenge 2017* [22] – que possui 12186 gravações curtas, no intervalo de 9 a 60 segundos, de uma derivação, para a classificação do sinal entre as classes ritmo sinusal normal, fibrilação atrial, ritmo alternativo ou sinal com muito ruído para ser classificado. Esses dados foram usados por [42, 77].

6. *St Petersburg INCART 12-lead Arrhythmia Database* [22] – composto por 75 gravações de 30 minutos das 12 derivações de pacientes com doença arterial coronariana (DAC), isquemia, anormalidades na condução de impulsos elétricos e arritmias, e composto por anotações de cada batimento. Foi usado por [45, 78].

Por ser o mais explorado na literatura e conter as anotações para a classificação de cada batimento, o banco de dados *MIT-BIH Arrhythmia Database* (MITDB) [22, 23] foi escolhido neste projeto para permitir comparações dos métodos desenvolvidos com os diversos estudos realizados na área. O MITDB foi o primeiro banco disponibilizado publicamente para o estudo de detectores de arritmias e suas anotações já foram corrigidas diversas vezes ao longo dos anos, diferentemente de outros bancos, com anotações não corrigidas manualmente.

Nesse banco, 48 gravações foram escolhidas a partir de um conjunto de 4000 registros de 24 horas ambulatoriais de pacientes estudados nos laboratórios do *Boston's Beth Israel Hospital*, entre os anos de 1975 e 1979. Esses sinais foram obtidos de 47 pacientes diferentes, sendo que os registros '201' e '202' pertencem a um único paciente. Desses pacientes, 25 foram selecionados para incluir arritmias menos comuns, mas clinicamente importantes, que não seriam bem representadas em uma pequena amostra, e os outros 23 foram selecionados aleatoriamente. O primeiro grupo representa a "série 200", enquanto o segundo, a "série 100" [23].

Na maioria das gravações, o sinal do primeiro canal é de uma derivação bipolar II modificada, obtida com o posicionamento dos eletrodos no peito, uma prática padrão para o registro de ECG ambulatorial. Já o sinal do canal 2 é, em geral, proveniente de uma derivação V1 modificada, sendo, em alguns pacientes, a derivação V2, V4 ou V5 [23]. As derivações de cada gravação podem ser consultadas na Tabela 3.1.

Tabela 3.1: Derivações de cada gravação.

MLII e V1	101, 106, 108, 109, 112, 115, 116, 118, 119, 122, 201, 203, 205, 207, 208, 209, 215, 220, 223, 230, 105, 111, 113, 121, 200, 202, 210, 212, 213, 214, 219, 221, 222, 228, 231, 232, 233, 234, 107, 217
MLII e V2	103, 117
MLII e V4	124
V5 e MLII	114
MLII e V5	100, 123
V5 e V2	102, 104

As gravações foram digitalizadas com uma frequência de amostragem de 360 Hz, com conversores analógico-digitais de 11 bits de resolução e uma tensão de fundo de escala de 10 mV. Após a digitalização, um detector sensível à inclinação do complexo QRS foi usado para encontrar os batimentos e dois cardiologistas elaboraram as anotações de forma independente, adicionando batimentos, registrando os batimentos anormais, apagando falsas detecções e adicionando informações dos ritmos e qualidade dos sinais. As discordâncias das anotações foram

resolvidas pelo consenso entre os cardiologistas, mas um total de 33 batimentos permaneceu não classificado pela ausência de um acordo [23].

A Tabela 3.2 apresenta os significados e as quantidades de cada símbolo em todo o banco. A maioria dos símbolos representa um tipo de batimento diferente, identificado pelos cardiologistas, enquanto outros são reservados para sinalizar eventos, como, por exemplo, a mudança de ritmo ou de qualidade do sinal.

Tabela 3.2: Significados dos símbolos presentes no MITDB.

Símbolo	Significado	Quantidade
N	Batimento normal	75052
L	Batimento de bloqueio do ramo esquerdo	8075
R	Batimento de bloqueio do ramo direito	7259
V	Contração ventricular prematura	7130
/	Batimento de marca-passo	7028
A	Batimento atrial prematuro	2546
+	Mudança de ritmo	1291
f	Fusão de batimentos de marca-passo e normal	982
F	Fusão de batimentos ventricular e normal	803
∼	Mudança na qualidade do sinal	616
!	Flutter ventricular	472
”	Comentários	437
j	Batimento de escape juncional	229
x	Onda P não conduzida	193
a	Batimento atrial prematuro aberrante	150
|	Artefato semelhante ao QRS isolado	132
E	Batimento de escape ventricular	106
J	Batimento juncional prematuro	83
Q	Batimentos não classificados	33
e	Batimento de escape atrial	16
[	Início da fibrilação/flutter ventricular	6
]	Fim da fibrilação/flutter ventricular	6
S	Batimento prematuro supraventricular	2

Escolhido o banco de dados, é necessário estabelecer um paradigma para a divisão dos dados nos conjuntos de teste e de treinamento. Essa separação será tratada na seção a seguir.

3.3 Separação dos Dados

A separação dos dados entre os conjuntos de teste e de treinamento é determinante para o desempenho das redes. No caso da classificação de arritmias, é importante que o modelo proposto seja clinicamente viável. Nesta seção, descreve-se a separação dos dados adotada neste projeto.

As características morfológicas e temporais do sinal de ECG dependem do paciente e de sua condição física [57]. Essas características são exclusivas e distintas para cada indivíduo, como consequência de fatores como gênero, orientação da massa cardíaca, condutividade e ordem de ativação dos músculos cardíacos [79].

Por isso, o sinal de ECG tem sido cada vez mais usado para a identificação biométrica em conjunto com outros sinais [79–81]. Estudos como [82] têm demonstrado que métodos de autenticação biométrica baseados no sinal de ECG são robustos, necessitando apenas de uma derivação para atingir bons resultados, devido à singularidade desse sinal em cada indivíduo.

Assim, a presença de batimentos de um mesmo paciente tanto no conjunto de treinamento quanto no de teste para a detecção e a classificação de arritmias influencia positivamente o desempenho da rede, mas não é clinicamente interessante pois prejudica a generalização do modelo [24–27,48,83]. Isso acontece porque as redes aprendem particularidades desses pacientes e, consequentemente, apresentam maiores acurácias do que quando sinais de um mesmo paciente não se repetem no treinamento e teste.

Na Figura 3.6 estão ilustradas as gravações de dois pacientes distintos e ambos os trechos contêm batimentos considerados como normais. Observa-se que, apesar de pertencerem à mesma classe, há diferenças significativas entre os dois sinais que podem ser aprendidas pelas redes durante o treinamento.

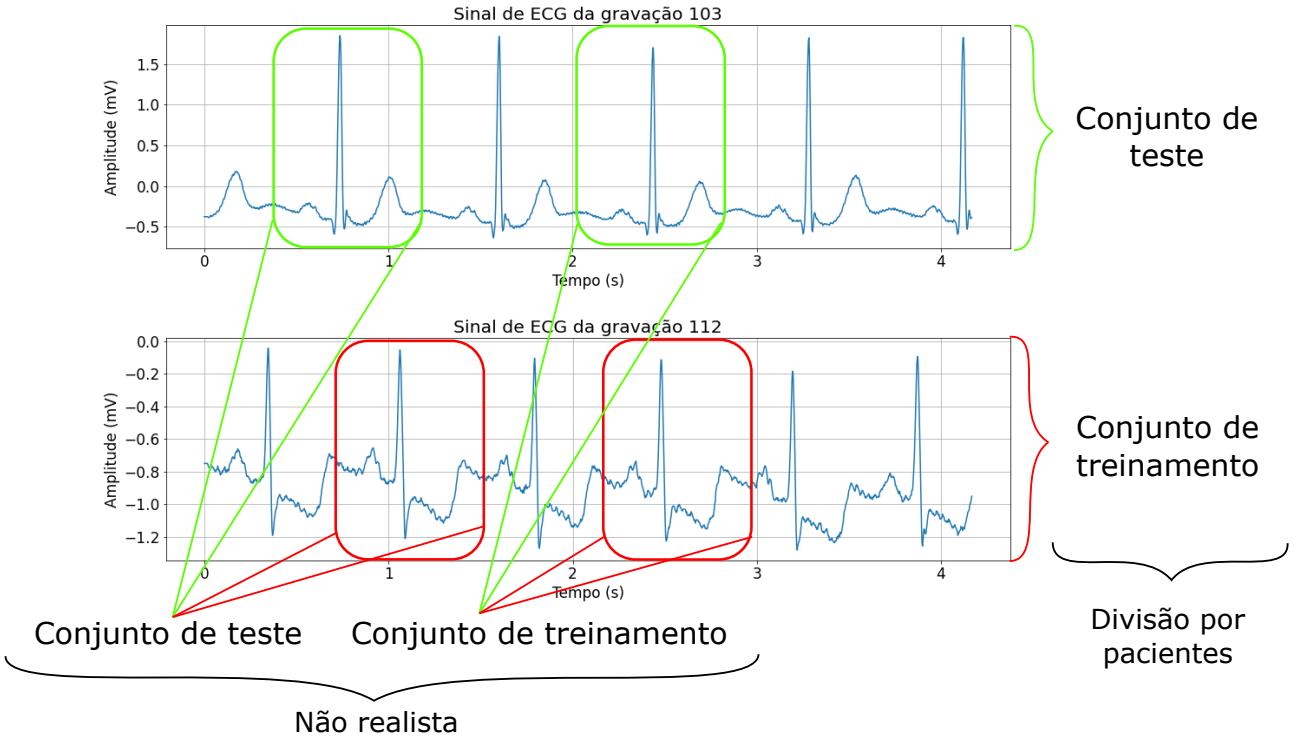


Figura 3.6: Diferentes abordagens de divisão de dados.

Diante disso, a *Association for the Advancement of Medical Instrumentation* (AAMI) [84] estabeleceu o padrão ANSI/AAMI EC57:1998/(R)2008 para a avaliação dos desempenhos dos algoritmos na classificação de arritmias, permitindo a comparação justa entre diferentes métodos [26]. O padrão recomenda a divisão das gravações de modo que batimentos de um mesmo paciente não sejam simultaneamente usados nos conjuntos de treinamento e de teste. Ele também especifica os tipos de instrumentos que devem ser usados para o registro do sinal e como esse registro deve ser realizado, além de indicar os bancos de dados que seguiram esses procedimentos e que podem ser utilizados, como o MITDB [27].

No entanto, poucos pesquisadores seguem as recomendações da AAMI, originando resultados não confiáveis, uma vez que muitos são favorecidos pelo viés de possuir a informação de um mesmo paciente nos conjuntos de treinamento e de teste. Essa prática dificulta a verificação dos méritos relativos aos diferentes algoritmos.

A AAMI não especifica quais pacientes devem ser usados para o treinamento ou para o teste. E esse problema foi tratado por [24], que propôs a divisão dos pacientes em dois grupos, DS1 e DS2, de modo que ambos os conjuntos tivessem um número de amostras aproximadamente balanceado [27]. Os conjuntos DS1 e DS2 contêm aproximadamente 50000 batimentos, compostos por uma mistura dos sinais de ECG de rotina da "série 100" e dos sinais de arritmias ventriculares, juncionais e supraventriculares complexas da "série 200".

Muitos outros estudos [25–27, 83] basearam-se na divisão proposta por [24] e essa separação, apresentada na Tabela 3.3, foi usada neste projeto. A AAMI também recomenda ignorar as gravações provenientes de pacientes com marca-passos no MITDB, correspondentes aos arquivos '102', '104', '107' e '217', sugestão que foi seguida por [24] e também considerada aqui.

Tabela 3.3: Separação dos pacientes baseada em [24].

DS1	101	106	108	109	112	114	115	116	118	119	122
	124	201	203	205	207	208	209	215	220	223	230
DS2	100	103	105	111	113	117	121	123	200	202	210
	212	213	214	219	221	222	228	231	232	233	234

Em um primeiro contato com o banco de dados do MITBD, as redes iniciais desenvolvidas neste projeto tiveram como objetivo apenas detectar a presença ou não de um batimento característico de arritmia, sem a distinção entre as diferentes classes. Para isso, as anotações foram agrupadas como indicado na Tabela 3.4.

Tabela 3.4: Separação dos batimentos para a detecção de arritmias.

Grupo	Símbolos	Quantidade total
Batimentos normais	N	75052
Batimentos anormais	S, e, Q, J, E, a, j, F, f, A, /, V, R, L	34442
Não batimentos	[,], !, x, |, ∼, +, ”	3153

Assim, para o problema da detecção da arritmia, a rede desenvolvida teve apenas um neurônio de saída, que fornecia o valor 0, caso a rede não detectasse arritmia, e 1, caso contrário. Para balancear os dados entre as classes batimentos normais e batimentos anormais, usou-se metade da quantidade de batimentos normais de cada paciente.

Posteriormente, foram desenvolvidas as redes para a classificação das diferentes arritmias. Segundo a recomendação da AAMI [84], os dados do MITDB devem ser divididos em cinco classes:

1. Classe N: contém os batimentos que se originam no nó SA (normais e de bloqueio de ramo);

2. Classe S: contém os batimentos supraventriculares ectópicos;

3. Classe V: engloba os batimentos ventriculares ectópicos;

4. Classe F: contém batimentos que resultam da fusão de batimentos normais e ventriculares ectópicos;

5. Classe Q: possui batimentos desconhecidos e os batimentos de marca-passo.

As redes desenvolvidas no Relatório Parcial classificaram cada batimento em uma das cinco classes definidas pelo AAMI, com cinco neurônios na camada de saída. A Tabela 3.5 apresenta a divisão dos batimentos do MITDB para essas classes.

No período coberto por este relatório, adotou-se o problema de classificação entre quatro classes, as classes N, S, V e F. A identificação da classe Q não foi considerada devido à ausência de resultados promissores tanto na literatura quanto neste projeto. Além desse agrupamento, muitos autores otimizam um problema de três classes, classificando N, S e V, e removendo a classe F devido à menor quantidade de dados dessa classe [33, 35, 47, 48].

Tabela 3.5: Classes de batimentos cardíacos segundo a AAMI.

Classe AAMI	N	S	V	F	Q
MIT-BIH	Batimento normal (N) Batimento de bloqueio do ramo esquerdo (L) Batimento de bloqueio do ramo direito (R) Batimento de escape atrial (e) Batimento de escape juncional (j)	Batimento atrial prematuro (A) Batimento atrial prematuro aberrante (a) Batimento juncional prematuro (J) Batimento prematuro supraventricular (S)	Contração ventricular prematura (V) Batimento de escape ventricular (E)	Fusão de batimentos ventricular e normal (F)	Batimento de marca-passo (/) Batimentos não classificados (Q) Fusão de batimentos de marca-passo e normal (f)

3.4 Preparando o Sinal para a Classificação

Em geral, o reconhecimento do sinal ECG por técnicas computadorizadas possui quatro passos: o pré-processamento do sinal, a detecção do complexo QRS e das ondas P e T, a extração das características e finalmente a classificação [85]. A Figura 3.7 ilustra as etapas de pré-processamento, de detecção do complexo QRS e de extração das características, que são descritas em maiores detalhes nesta Seção.

3.4.1 Etapa de Pré-processamento

O pré-processamento do sinal tem por objetivo reduzir os ruídos e artefatos provenientes de várias origens, como a interferência causada pela rede elétrica (na faixa de 60 Hz), os ruídos devido à respiração do paciente (entre 0,15 a 0,3 Hz), os ruídos atribuídos ao mal posicionamento dos eletrodos na pele e aos aparelhos médicos usados durante a captura do sinal e o artefato da variação da impedância entre o eletrodo e a pele [57].

A etapa de pré-processamento, largamente explorada na literatura [86–89], engloba processos como filtragem, reamostragem, remoção de artefatos e normalização. Os métodos de pré-processamento a serem utilizados dependem do objetivo final da pesquisa. Assim, para a segmentação automática do sinal os métodos podem diferir daqueles usados para a classificação de arritmias [25].

Uma das propostas mais simples para a filtragem se dá com filtros de resposta ao impulso de duração finita (*Finite Impulse Response* - FIR) [89]. Tal método funciona bem para a atenuação

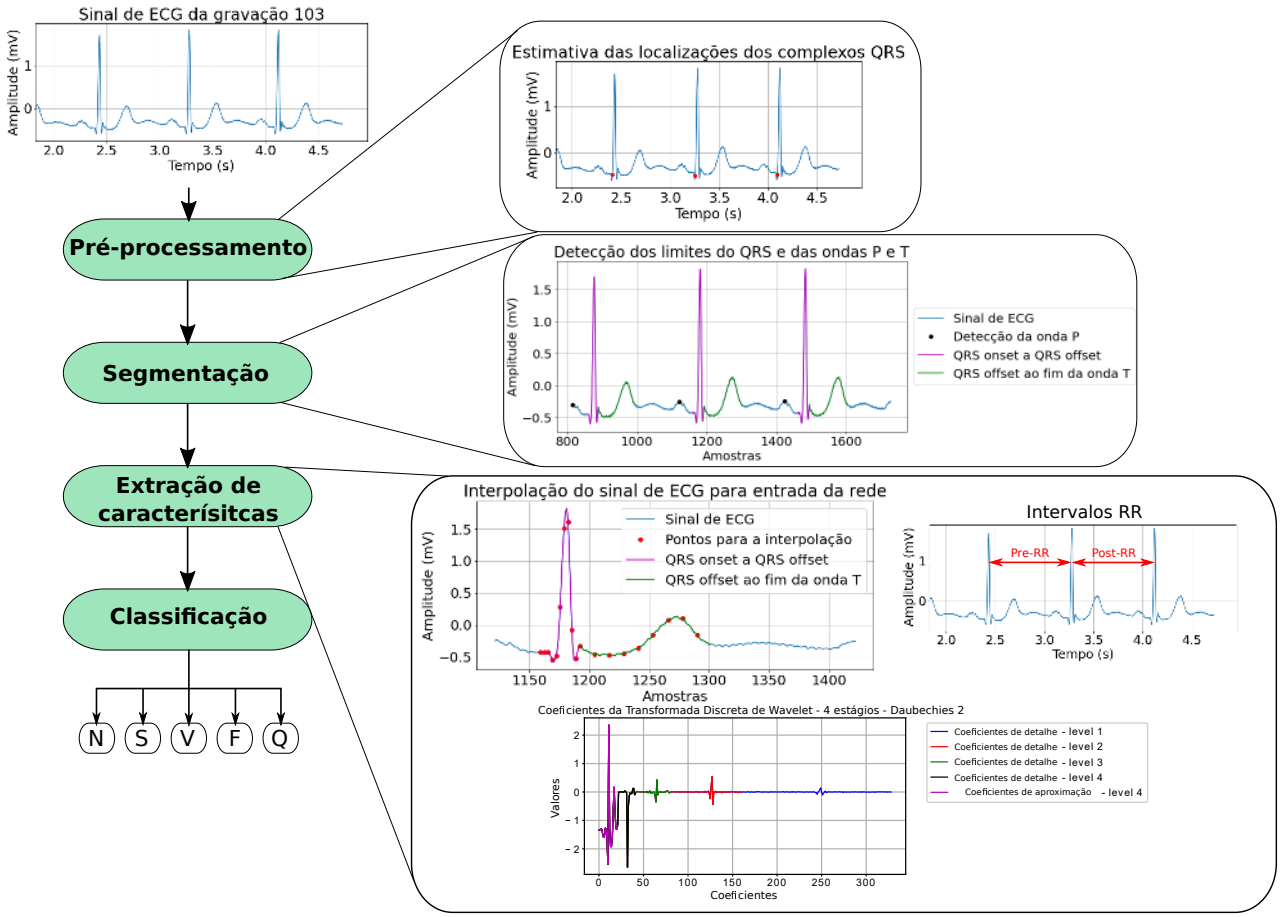


Figura 3.7: Etapas da classificação automática do sinal de ECG.

de frequências conhecidas, como a da interferência da rede elétrica, mas o uso indiscriminado de filtros para as frequências desconhecidas distorcem a morfologia do sinal [25]. Dessa forma, outras arquiteturas foram propostas, como os filtros adaptativo, redes neurais [90] e os métodos baseados na transformada de wavelet [86, 91].

O impacto das técnicas de pré-processamento na classificação automática do sinal ECG não é claro, e mesmo os métodos considerados como estado da arte na classificação não realizam o pré-processamento do sinal [25, 57].

Assim, para as redes neurais desenvolvidas neste projeto, o sinal não foi filtrado na etapa de pré-processamento. Dessa forma, o sinal original foi considerado como entrada das redes, visando padronizar o processo de avaliação de desempenho, como proposto em [25], por exemplo. Além disso, foram avaliados os desempenhos dos algoritmos desenvolvidos com e sem a normalização do sinal.

Nessa etapa, uma parte do Algoritmo de Pan e Tompkins [92], considerado como um método de pré-processamento por [57], foi desenvolvida para estimar o intervalo RR médio entre os batimentos de todas as gravações. O período médio foi usado para estabelecer o tamanho fixo

da entrada da rede.

O algoritmo de Pan e Tompkins detecta o complexo QRS por meio de uma série de filtros. A detecção do complexo QRS é uma tarefa trabalhosa, dificultada não apenas pela variabilidade fisiológica do sinal, como também pelos diferentes tipos de ruídos com características semelhantes ao complexo QRS. Tipicamente, o processamento do sinal para a detecção do complexo ocorre em três etapas principais: filtragem linear, transformação não-linear do sinal e uso de um algoritmo de decisão, com limiares dinâmicos [92]. O algoritmo proposto por Pan e Tompkins [92] abrange as três etapas, empregando os filtros passa-faixas, derivativo e integrador de janela móvel, para a primeira etapa; a potenciação do sinal, para a segunda; e os limiares adaptativos e as técnicas de discriminação da Onda T, para a terceira [92].

Para a estimativa do intervalo RR médio neste projeto, foram usados o filtro passa-faixas, o filtro derivativo e a transformação não linear do sinal sugeridos por [92], e um código simples de detecção por meio de um limiar estático. O algoritmo de decisão com limiares dinâmicos não foi implementado. Um diagrama esquemático do algoritmo desenvolvido está mostrado na Figura 3.8.

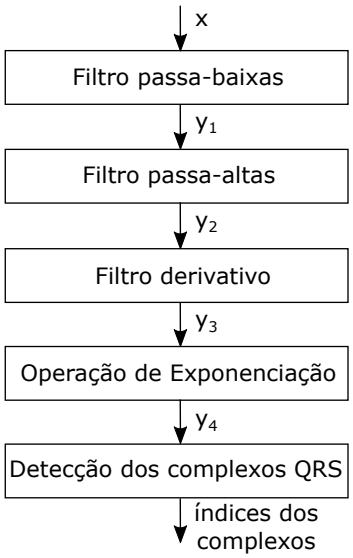


Figura 3.8: Diagrama esquemático da Etapa de Pré-Processamento.

Para ilustrar, foi escolhido o trecho inicial da gravação '103', apresentado na Figura 3.9, para observar os efeitos de cada etapa no processamento do sinal.

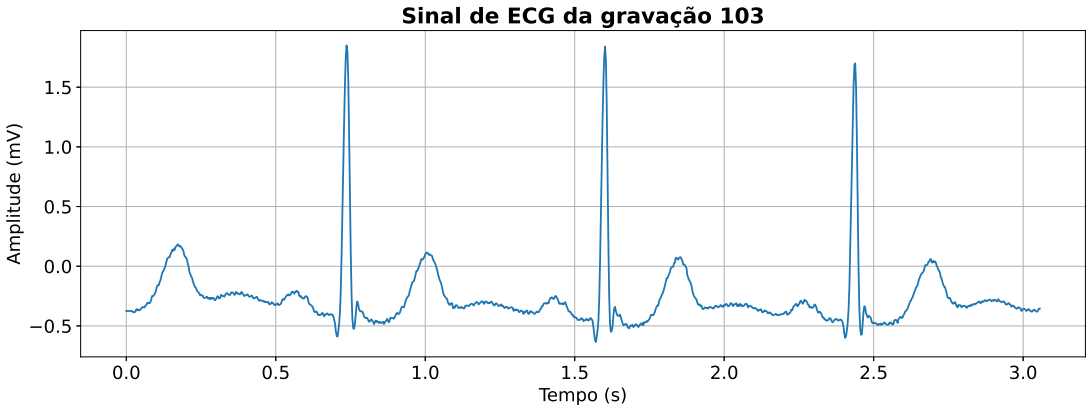


Figura 3.9: Trecho do sinal de ECG da gravação '103'.

O filtro passa-faixas tem como objetivo reduzir os ruídos e a interferência da rede elétrica, maximizando a energia do QRS nas frequências desejadas que se encontram na faixa de 5 a 15 Hz [92]. Para isso, um filtro passa-baixas, com frequência de corte de aproximadamente 11 Hz, é colocado em cascata com um filtro passa-altas, com frequência de corte de 5 Hz.

A função de transferência do filtro passa-baixas é descrita por

$$H_{lp}(z) = (1 + z^{-10}) + 2(z^{-1} + z^{-9}) + 3(z^{-2} + z^{-8}) + 4(z^{-3} + z^{-7}) + 5(z^{-4} + z^{-6}) + 6z^{-5}. \quad (3.1)$$

A resposta em frequência e o diagrama de polos e zeros desse filtro estão apresentados nas Figuras 3.10 e 3.11. Nota-se que se trata de um filtro FIR de fase linear por trechos [93].

Figura 3.10: Resposta em frequência do filtro passa-baixas.

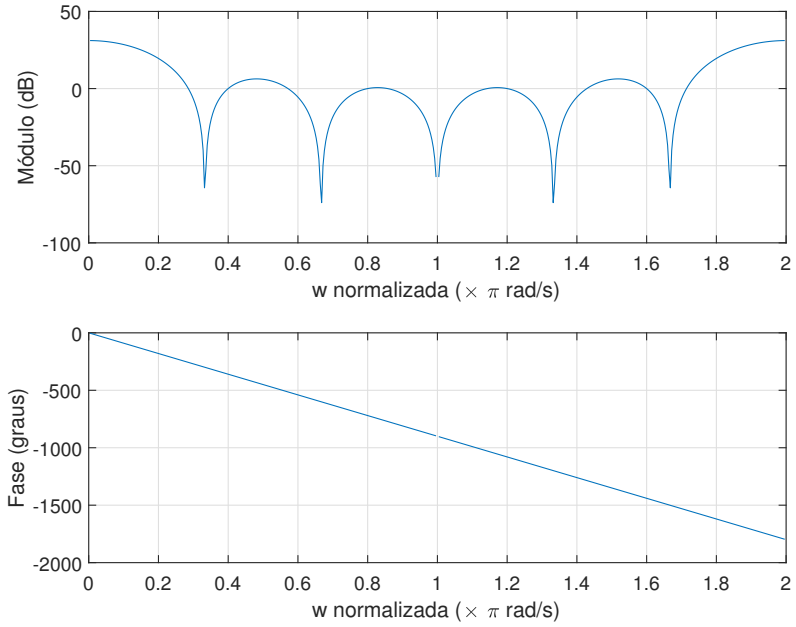
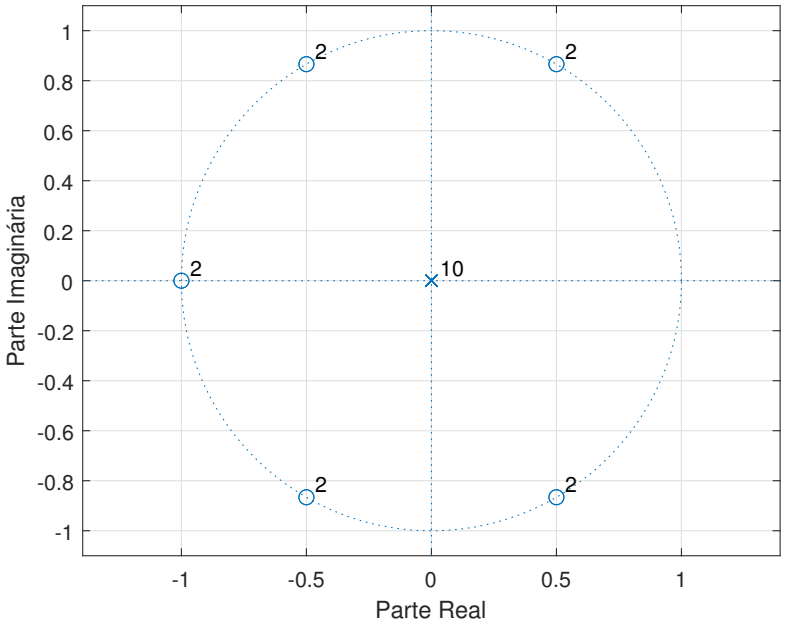


Figura 3.11: Diagrama de polos e zeros do filtro passa-baixas.



O atraso provocado por esse filtro é de 5 amostras e o ganho é de 36. Na Figura 3.12, é possível observar o sinal de saída desse filtro, considerando o sinal da Figura 3.9 como entrada. Cabe observar que esse sinal foi dividido por 36 para compensar o ganho do filtro.

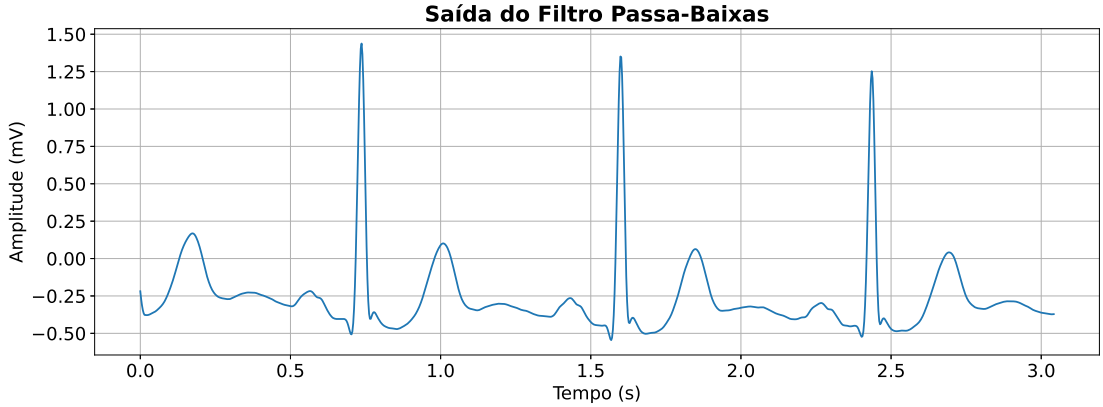


Figura 3.12: Saída y_1 do filtro passa-baixas.

O filtro passa-altas possui função de transferência

$$H_{hp}(z) = \frac{-1/32 + z^{-16} - z^{-17} + z^{-32}/32}{1 - z^{-1}}. \tag{3.2}$$

A resposta em frequência e o diagrama de polos e zeros desse filtro estão apresentados nas Figuras 3.13 e 3.14. A forma da função de transferência de (3.2) é conveniente por ser compacta. No entanto, ela dá a impressão que se trata de um filtro IIR. Manipulações algébricas em (3.2) permitem mostrar que se trata de um filtro FIR com fase linear por trechos, o que pode ser comprovado pelo diagrama de polos e zeros.

Figura 3.13: Resposta em frequência do filtro passa-altas.

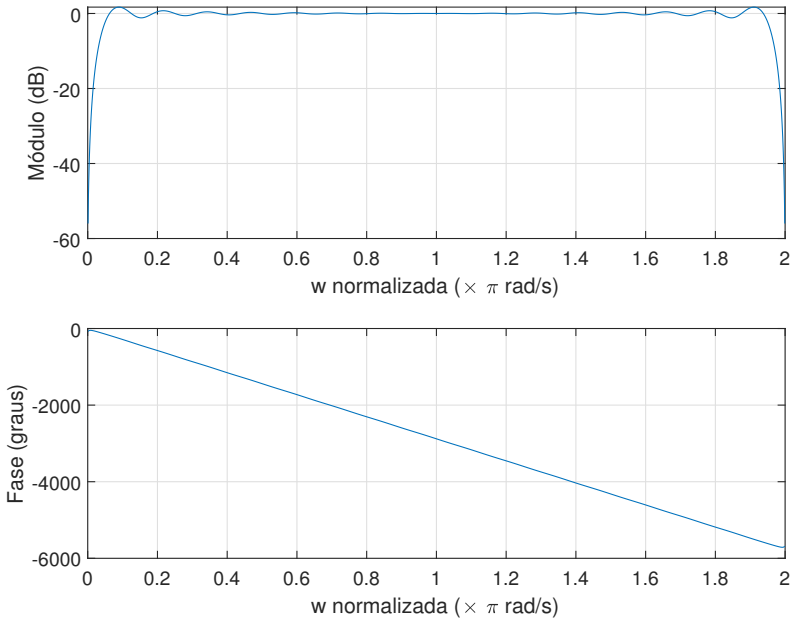
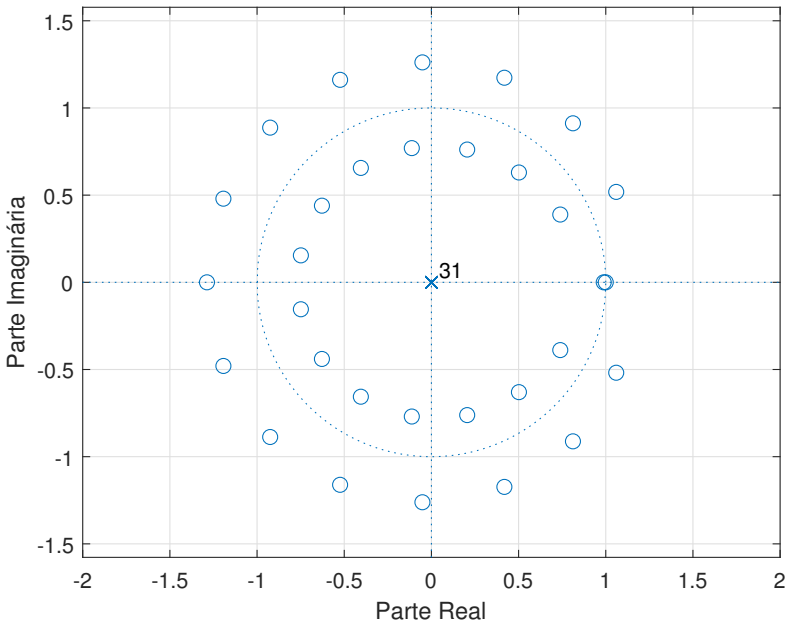


Figura 3.14: Diagrama de polos e zeros do filtro passa-altas.



Esse filtro possui ganho 1 e provoca um atraso de 16 amostras. A saída y_2 desse filtro está apresentada na Figura 3.15.

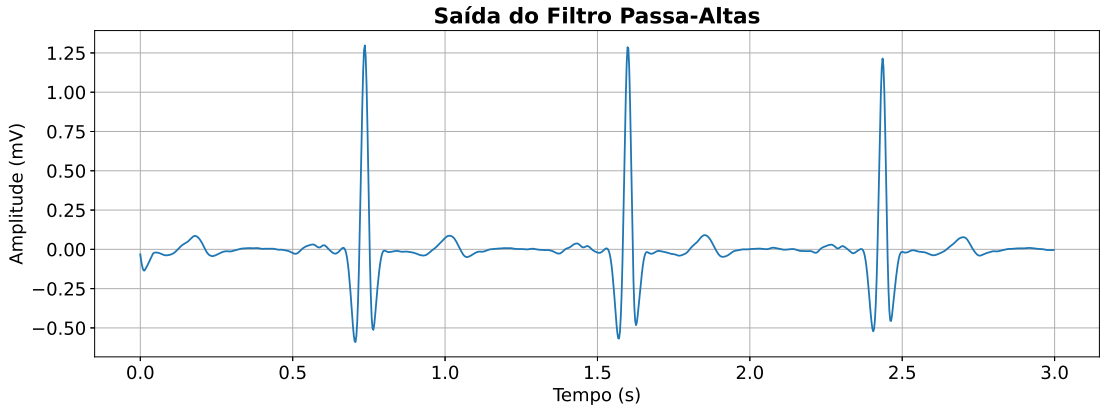


Figura 3.15: Saída y_2 do filtro passa-altas.

Após a etapa de filtragem, a derivada do sinal é utilizada para providenciar informações sobre a inclinação do complexo QRS [92]. Para isso, utiliza-se um filtro FIR de quarta ordem, com função de transferência dada por

$$H(z) = 0.1(-2z^{-2} - z^{-1} + z^{1} + 2z^{2}). \tag{3.3}$$

Esse filtro aproxima uma derivada ideal na faixa de 0 a 30 Hz e possui um atraso de 2 amostras [92]. A Figura 3.16 ilustra a saída y_3 obtida para a gravação escolhida.

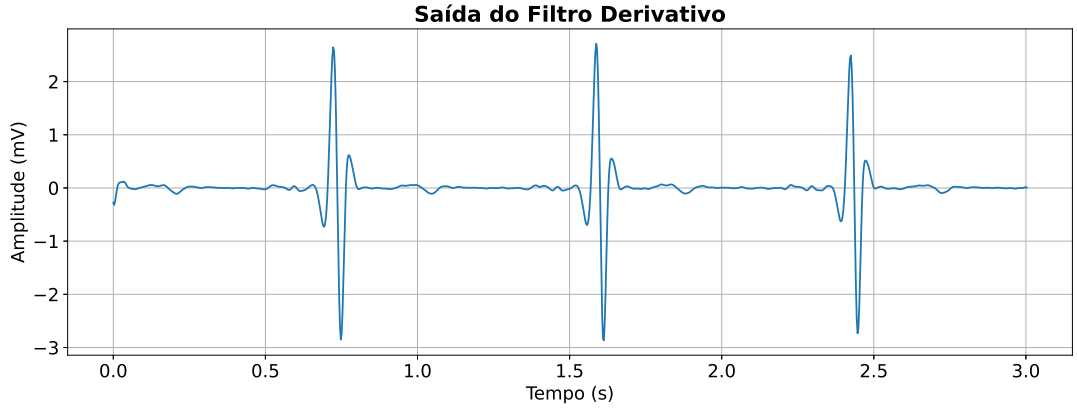


Figura 3.16: Saída y_3 do filtro derivativo.

A saída do filtro derivativo é amplificada, de modo a ressaltar as maiores frequências, que são predominantemente as frequências do sinal de ECG. Para tanto, utiliza-se uma transformação não linear: a operação de potenciação dada por

$$y_4(n) = (y_3(n))^2. \tag{3.4}$$

A Figura 3.17 apresenta a saída y_4 dessa operação.

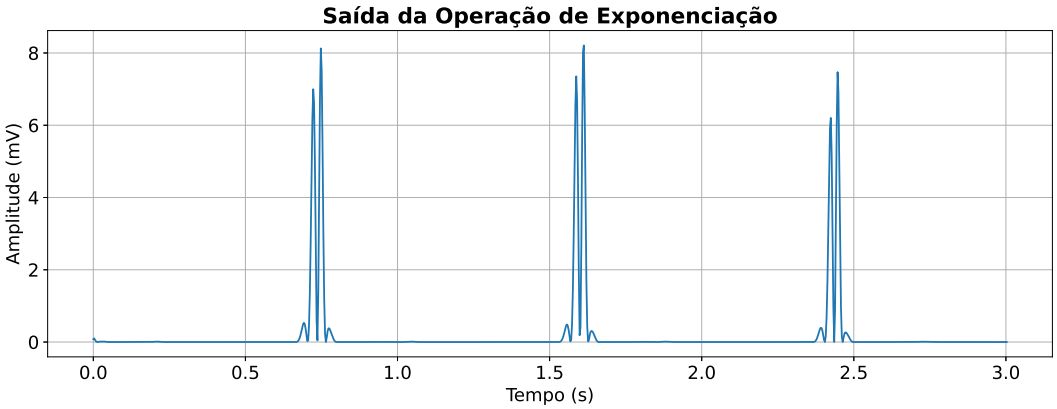


Figura 3.17: Saída y_4 da operação de exponenciação.

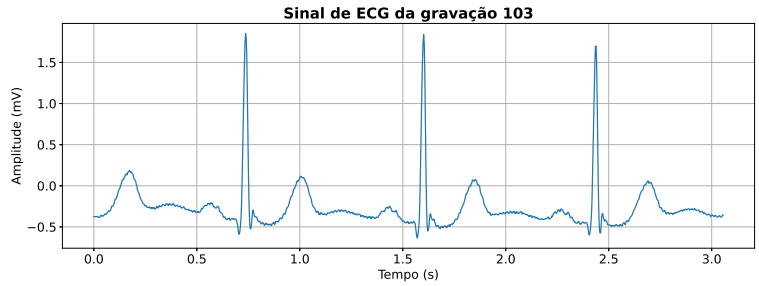
Na Figura 3.18 estão mostrados novamente os gráficos de entrada e de saída das etapas para a observação do efeito de cada filtro.

O projeto de pesquisa não tem como objetivo aperfeiçoar os métodos de detecção do complexo QRS, sendo necessária apenas uma estimativa do período médio dos batimentos de todos os pacientes para o estabelecimento da entrada da rede. A partir da saída y_4, é possível obter os tempos de início dos complexos QRS por meio do pseudocódigo descrito no Algoritmo 1.

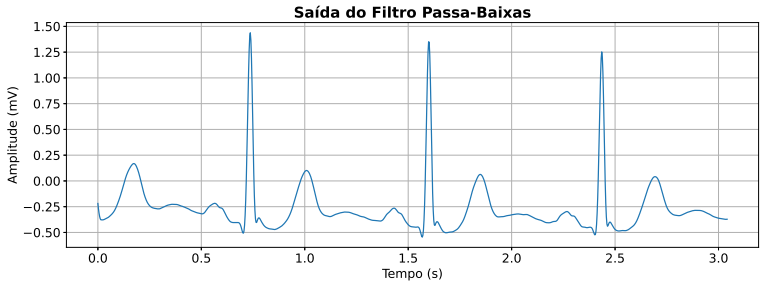
Algoritmo 1: Estimativa do período médio.

```
limiar ← max(y4) * 0.1;
subida ← 0;
ultimo ← 0;
para i = 0; i ← i + 1; i < N faça
    se y4[i] > limiar então
        se subida = 0 então
            RRi ← i − ultimo;
            ultimo ← i
        fim
        subida ← 30;
    senão
        se subida > 0 então
            subida = subida − 1;
        fim
    fim
fim
```

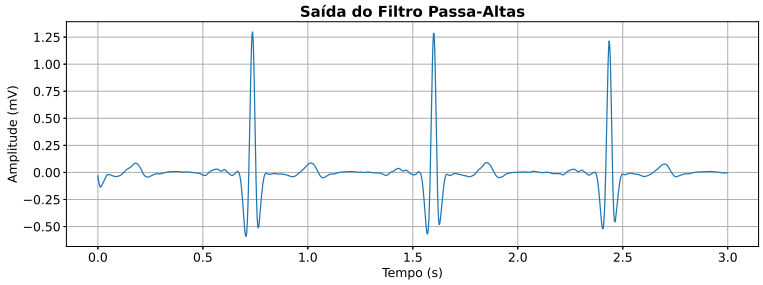
A Figura 3.19 ilustra a detecção dos inícios dos complexos QRS por meio do Algoritmo 1 para a realização da estimativa do período médio dos batimentos.



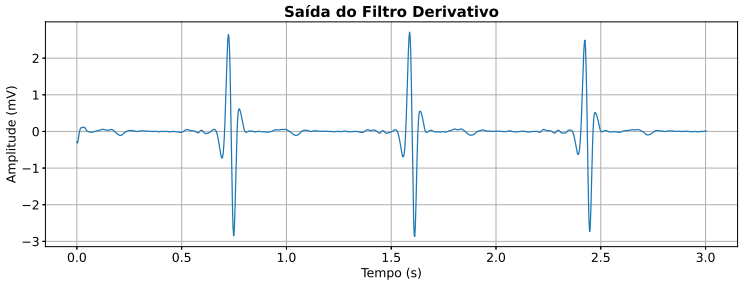
(a) x



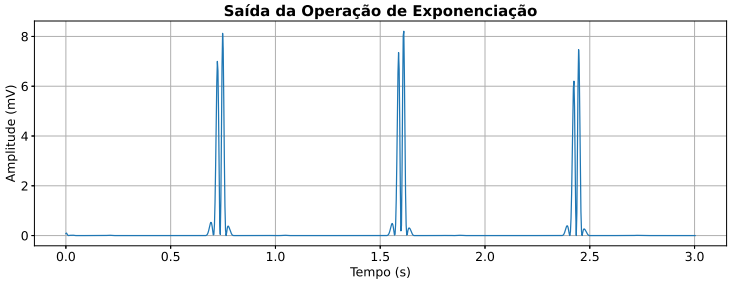
(b) y_1



(c) y_2



(d) y_3



(e) y_4

Figura 3.18: Comparação do efeito de cada etapa do algoritmo de Pan e Tompkins.

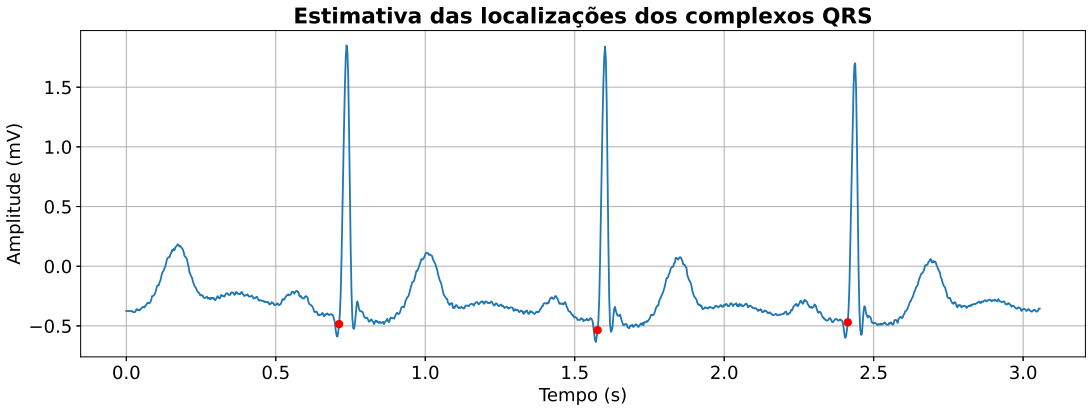


Figura 3.19: Detecção dos complexos QRS.

Com o Algoritmo 1, foram calculados os períodos médios de um batimento de cada paciente. Para verificar a validade dos resultados obtidos, usou-se a função de identificação do complexo QRS do pacote `wfdb-python` [94] para comparação dos resultados. Realizando-se a média geral dos batimentos de todos os pacientes e excluindo-se os pacientes 102, 104, 107 e 217, como recomendado pela AAMI, obteve-se o valor de 322,77 amostras para o período médio geral, o que corresponde a um trecho de 0,897s do sinal. O módulo [94] apresentou um período médio de 0,821s do sinal, indicando um erro de 9,2% entre o resultado obtido neste trabalho sem o uso dos limiares adaptativos de [92] e o resultado desse módulo. Como os períodos médios dos pacientes obtidos pelo módulo variam de 0,537 s a 1,177 s e é necessário fixar um mesmo valor para todos os pacientes para o tamanho de entrada fixo das redes neurais, os resultados obtidos foram considerados satisfatórios para esse propósito.

Estabeleceu-se, por fim, o valor aproximado de 320 amostras (0,889 s) como sendo a estimativa do período médio geral dos batimentos para todos os métodos e propósitos do projeto. Vale lembrar que essa é apenas uma estimativa do período médio dos batimentos para o estabelecimento da entrada da rede, sendo suficiente para esse objetivo. Para a segmentação dos batimentos, uma detecção mais precisa dos complexos é necessária, o que será comentado a seguir.

3.4.2 Etapa de Segmentação dos Batimentos

A etapa de determinação do pico e dos limites do complexo QRS e das ondas P e T (conhecida como *delineation*), assim como a etapa de pré-processamento, foi muito explorada na literatura [95, 96], tendo sido estudada por mais de três décadas [25]. A segmentação dos batimentos, quando não há anotações das ocorrências dos eventos, é um problema de alta complexidade.

Assim, neste trabalho não se investigou a implementação de algoritmos para esse fim. Em vez disso, decidiu-se utilizar o método proposto em [97] e implementado em MATLAB pelo pacote `ECGkit` [98].

A segmentação dos batimentos proposta por [97] é baseada na transformada de wavelet. O primeiro passo na segmentação e na determinação do pico e dos limites das ondas é a detecção do complexo QRS. Encontrado o complexo QRS, é possível identificar os picos das ondas individuais, P e T, e encontrar o início (*onset*) e o fim (*offset*) do complexo. Em geral, algoritmos desenvolvidos para esse fim partem de um complexo QRS anterior e definem janelas temporais de busca antes e depois do pico R para procurar outras ondas. Uma vez definida a janela, algumas técnicas são aplicadas para realçar características específicas de cada onda, como, por exemplo, a sua faixa de frequência [97].

Devido aos ruídos e à baixa amplitude dos limites entre as ondas, como visto na Seção 3.1, a identificação do início e do fim de cada onda é uma tarefa difícil. Além disso, não existe uma regra "universal" para a localização desses limites. Este fato pode ser observado na grande variedade de abordagens da literatura, que englobam modelos matemáticos, critérios de inclinação, transformada de wavelet, filtros adaptativos e até mesmo redes neurais, como a implementada em [16].

No caso do algoritmo proposto por [97], foi usada a transformada de wavelet. O erro médio obtido para essa determinação não excedeu um intervalo de amostra e o desvio padrão ficou em torno da tolerância aceita por especialistas, atingindo melhores resultados do que outros algoritmos conhecidos, principalmente na determinação do fim da Onda T. A Figura 3.20 ilustra a identificação da Onda P e a delimitação do complexo QRS e da Onda T.

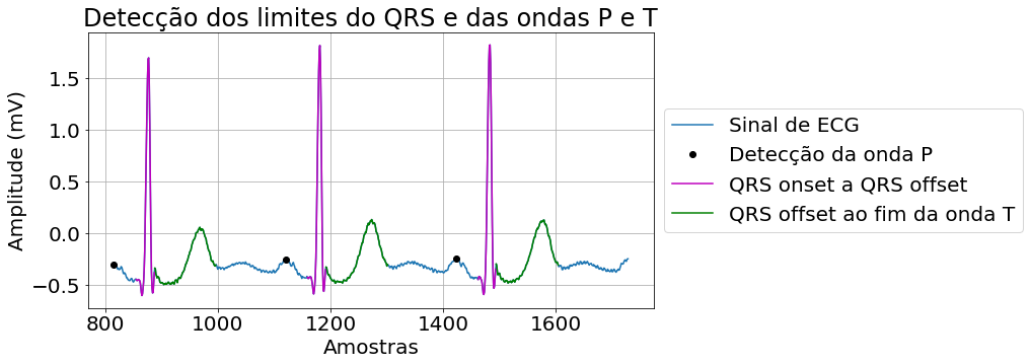


Figura 3.20: Segmentação de um batimento.

3.4.3 Etapa de Extração das Características

A etapa de extração de características é usada para encontrar a menor quantidade de características do sinal de ECG que permitem taxas de classificação aceitáveis [85].

Várias técnicas de extração de características foram propostas na literatura, como as técnicas de análise de componentes independentes (*independent component analysis* - ICA), os intervalos RR, a análise de discriminantes lineares (*linear discriminant analysis* - LDA), a transformada de Fourier, a transformada de wavelet e a técnica que usa estatísticas de ordem superior (*high order statistics* - HOS) [27]. No entanto, a maioria dos algoritmos utiliza a transformada de wavelet.

Nos últimos anos, o uso da transformada de wavelet como uma ferramenta de processamento e de análise cresceu em comparação com o uso da transformada discreta de Fourier. A transformada discreta de Fourier apenas fornece a informação espectral e não temporal dos sinais, enquanto a transformada de wavelet fornece uma representação no tempo e na frequência e é apropriada para sinais não estacionários como o ECG [85, 97].

A transformada de wavelet é uma alternativa à transformada de Fourier de tempo curto, que usa janelas pequenas para a identificação de altas frequências e janelas maiores para a identificação de baixas frequências [99]. A seguir, a extração de características de [24] utilizada no projeto é descrita em maiores detalhes.

3.4.4 A Extração de Características da Literatura

Chazal *et al.* [24] propuseram um classificador baseado em discriminantes lineares para 12 configurações diferentes de características. O classificador tem seus parâmetros determinados por um estimador de máxima verossimilhança. O estudo também não investigou o problema de segmentação do ECG, usando o algoritmo `ecgpuwave` da Physionet [22, 100]. Segundo o artigo, a extração de características é necessária porque desempenhos melhores são atingidos se, primeiramente, um pequeno número de características for extraído.

Os pesquisadores investigaram a influência de diferentes características como entrada. As informações usadas foram:

1. Intervalos RR: intervalo RR entre o batimento atual e o anterior (*Pre-RR*), intervalo RR entre o batimento atual e o posterior (*Post-RR*), intervalo RR médio de toda a gravação de um paciente (*Average RR*) e o intervalo RR médio local entre 10 batimentos adjacentes (*Local average RR*);

2. Durações de intervalos (1ª e 2ª derivação): duração do QRS (intervalo de tempo entre o QRS *onset* e o QRS *offset*), duração da Onda T (intervalo de tempo entre o QRS *offset* e o fim da Onda T) e um *booleano* indicando a presença ou não da Onda P;

3. Morfologia 1A/1B (1ª e 2ª derivação): 10 amostras em uma janela que se inicia no QRS *onset* e termina no QRS *offset*, 9 amostras em uma janela que se inicia no QRS *offset* e termina no final da Onda T;

4. Morfologia 2A/2B (1ª e 2ª derivação): mesmas características da Morfologia 1A/1B, com o sinal sendo normalizado antes da tomada dos pontos;

5. Morfologia 3A/3B (1ª e 2ª derivação): 10 amostras em uma janela que se inicia 50 ms antes do pico R e termina 100 ms depois (engloba parte do complexo QRS), 9 amostras em uma janela que se inicia 150 ms depois do pico R e termina 500 ms depois do pico R (engloba parte da Onda T); e

6. Morfologia 4A/4B (1ª e 2ª derivação): mesmas características da Morfologia 3A/3B, com o sinal sendo normalizado antes da tomada dos pontos.

Atribuiu-se a nomenclatura A para a 1ª derivação e B para a 2ª. Essas informações foram agrupadas em oito grupos de características (*Feature Sets*), FS1 a FS8, mostrados na Tabela 3.6.

Tabela 3.6: Grupos de características de [24].

FS1	26	Intervalos RR, duração de intervalos de A e morfologia 1A
FS2	26	Intervalos RR, duração de intervalos de A e morfologia 2A
FS3	22	Intervalos RR e morfologia 3A
FS4	22	Intervalos RR e morfologia 4A
FS5	26	Intervalos RR, duração de intervalos de B e morfologia 1B
FS6	26	Intervalos RR, duração de intervalos de B e morfologia 2B
FS7	22	Intervalos RR e morfologia 3B
FS8	22	Intervalos RR e morfologia 4B

Além de usar os grupos FS1 a FS8 separadamente, os pesquisadores também estudaram o processamento de informações de múltiplos grupos simultaneamente, com a probabilidade condicional obtida de cada grupo sendo combinada por meio de saídas de diferentes classificadores. Seja $P_m(k \mid \mathbf{x})$ a probabilidade de uma entrada $\mathbf{x}$ pertencer a uma classe k fornecida como uma posição do vetor de saída do m-ésimo classificador, o vetor final de saída de um modelo combinado para a entrada $\mathbf{x}$ é dado por [24]

$$P(k \mid \mathbf{x}) = \frac{\prod_{m=1}^{M} P_m(k \mid \mathbf{x})}{\sum_{l=1}^{K} \prod_{m=1}^{M} P_m(l \mid \mathbf{x})}, \tag{3.5}$$

sendo $K = 5$ o número de classes e M o número de classificadores. Essa operação é ilustrada pela combinação de classificadores esquematizada na Figura 3.21. A classificação global é obtida pela escolha da classe com a maior probabilidade condicional resultante dessa operação.

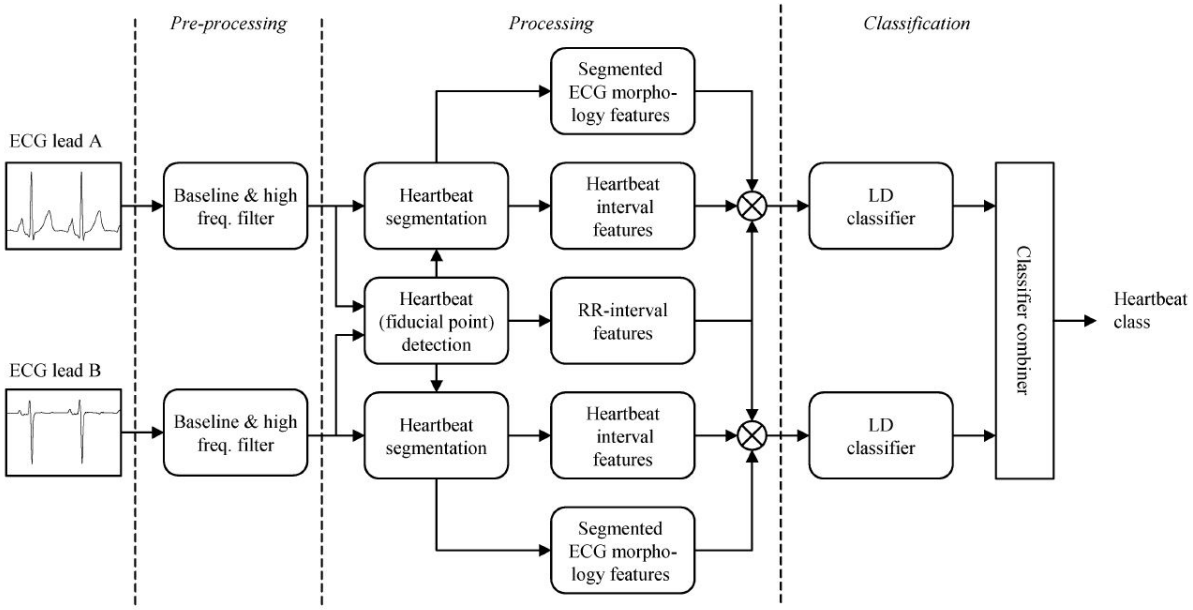


Figura 3.21: Representação da combinação de classificadores.
Fonte: [24].

Assim, além de usar configurações de uma derivação, foram usadas configurações de mais de uma derivação, combinando FS1 e FS5, FS2 e FS6, FS3 e FS7, e, por fim, FS4 e FS8. Após um primeiro estudo baseado em uma validação cruzada com 22 *folds*, cada um contendo uma gravação do conjunto DS1, escolheu-se a configuração com melhor desempenho dentre as 12 possíveis. Essa configuração foi a combinação de FS1 e FS5. Em um segundo estudo, seguindo a divisão DS1 e DS2, o modelo foi testado novamente com essa configuração, usando todo o conjunto DS1 para estimar os parâmetros e o conjunto DS2 para testá-los. A matriz de confusão de [24] para DS2 é apresentada na Tabela 3.7.

Tabela 3.7: Matriz de confusão de [24].

		Classes Preditas				
		N	S	V	F	Q
Classes Verdadeiras	N	38444	1904	303	3509	98
	S	173	1395	252	16	1
	V	117	321	2504	176	103
	F	33	1	7	347	0
	Q	4	0	3	0	0

3.5 Classificação das Arritmias

Nesta Seção definem-se as métricas de desempenho utilizadas para avaliar os classificadores. Em seguida, descrevem-se as arquiteturas das redes MLP, CNN e RNN, além do método estatístico da LDA, otimizadas para o problema de classificação das classes N, S, V e F. Por fim, os principais resultados para cada classificador e para as combinações dos classificadores são descritos. Os programas utilizados e os resultados das simulações podem ser acessados pelo GitHub no link `https://github.com/natnagata/ArrhythmiaClassification`.

3.5.1 Definição das Métricas

Algumas das métricas mais usadas na literatura para avaliar os desempenhos de classificadores de arritmias são a Acurácia, a Sensibilidade (*recall* ou *sensitivity*) e a Precisão (*positive predictive*). O cálculo dessas métricas é dado, respectivamente, por

$$Acc = \frac{VP + VN}{VP + VN + FP + FN} \times 100, \tag{3.6}$$

$$Se = \frac{VP}{VP + FN} \times 100, \tag{3.7}$$

e

$$P = \frac{VP}{VP + FP} \times 100, \tag{3.8}$$

em que VP representa o número de verdadeiros positivos; VN, o de verdadeiros negativos; FP, o de falsos positivos; e FN, o de falsos negativos.

A Acurácia mede a taxa de classificações corretas realizadas dentre todas as classificações feitas e é um indicativo do desempenho geral da rede. No entanto, é clinicamente interessante avaliar se o classificador está de fato cumprindo seu objetivo de detecção de determinada classe de arritmia. Para isso, é necessário medir quantos resultados são corretos nas classificações de dados com valor verdadeiro positivo para uma dada classe, métrica conhecida como Sensibilidade. Por outro lado, é necessário verificar se o classificador fornece muitos falsos alarmes de arritmias. Para isso, mede-se quantas classificações corretas foram realizadas dentre todas as saídas consideradas como positivas pelo modelo para determinada classe, que corresponde à métrica conhecida por Precisão.

A AAMI estabelece o cálculo das métricas de modo diferente, dando maior enfoque para as classes V e S, como pode ser observado nas Figuras 3.22 (a) e 3.22 (b). As métricas recomendadas pela AAMI focam na habilidade de um algoritmo em distinguir batimentos da classe V (chamada de VEB na Figura 3.22 (a)) de batimentos que não pertencem à classe V, e em

distinguir batimentos da classe S (chamada de SVEB na Figura 3.22 (b)) de batimentos que não pertencem à classe S [24].

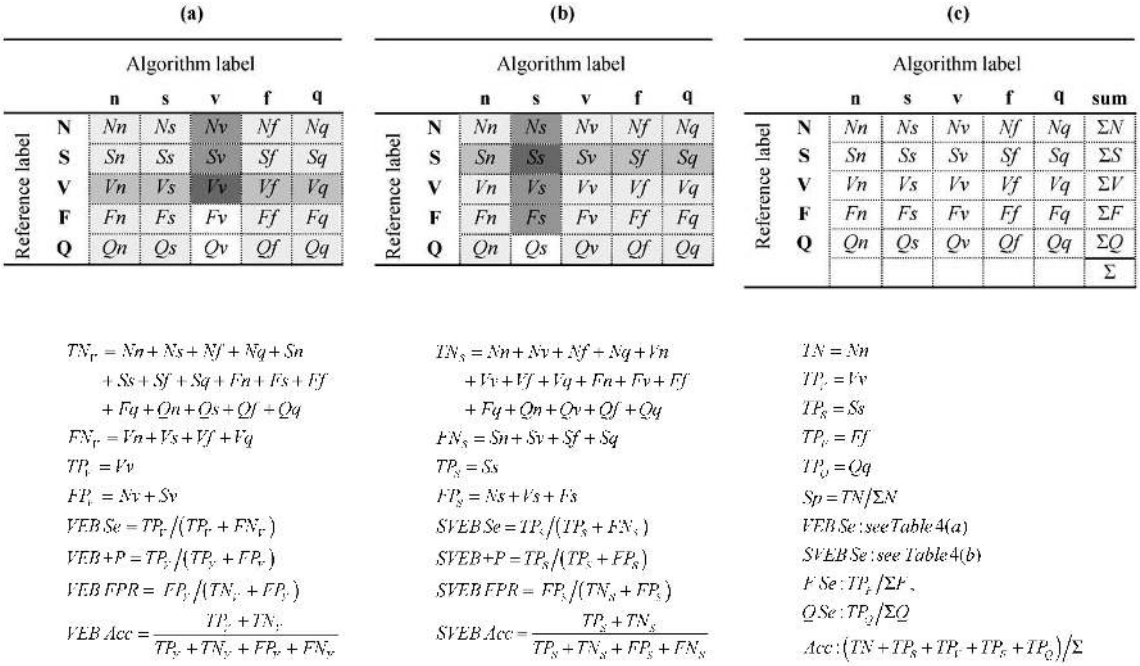


Figura 3.22: Métricas recomendadas pela AAMI.
Fonte: [24].

Vale notar que as métricas sugeridas não recompensam ou penalizam um classificador por classificar um batimento do tipo F ou do tipo Q como um do tipo V, assim como não recompensam ou penalizam por classificar um batimento do tipo Q como um do tipo S. Por isso, essas classificações são desconsideradas nos cálculos dos verdadeiros negativos de V e de S. Como as métricas listadas na Figura 3.22 (a) e 3.22 (b) não medem a habilidade de um classificador em distinguir batimentos em múltiplas classes simultaneamente, [24] também propõe as métricas da Figura 3.22 (c).

Neste trabalho, tanto as métricas recomendadas pela AAMI na Figura 3.22 (a) e 3.22 (b), quanto as métricas recomendadas por [24] na Figura 3.22 (c) foram utilizadas. Para as comparações finais entre os classificadores e para as comparações com a literatura, calcularam-se também para cada classe a métrica de $F1$-score, dada por

$$F1 = 2 \times \frac{Se \times P}{Se + P}. \tag{3.9}$$

Como no problema de classificação de arritmias a AAMI e diversos autores consideram ambas as métricas de Se e de P importantes, o cálculo do valor de $F1$-score é interessante, pois não prioriza uma sobre a outra, levando em conta tanto FP quanto FN. Quando as classes são

desbalanceadas, as métricas de $F1$-score fornecem resultados mais confiáveis do que a própria Acurácia, uma vez que uma classe muito representada pode dominar esses resultados, como ocorre com a classe N.

Assim, além da Acurácia, foram calculadas também métricas globais a partir do $F1$-score de cada classe para se obter uma ideia do desempenho geral da rede. Usaram-se as métricas gerais de *macro*-$F1$-score e de $F1$-score ponderado, definidas respectivamente por

$$mF1 = \frac{\sum_{k=1}^{K} F1_k}{K} \tag{3.10}$$

e

$$wF1 = \frac{\sum_{k=1}^{K} n_k F1_k}{\sum_{k=1}^{K} n_k}, \tag{3.11}$$

em que $F1_k$ é o $F1$-score obtido para cada classe k, K é o número de classes considerado e n_k é o número de elementos da classe k.

3.5.2 Classificação de Arritmias usando a rede MLP

No problema de classificação de arritmias cardíacas, os dados foram separados nas classes sugeridas pela AAMI. As cinco classes possíveis de arritmia são: batimentos do nó SA (N), supraventriculares ectópicos (S), ventriculares ectópicos (V), fusão de batimentos normais e ventriculares ectópicos (F) e desconhecidos ou de marca-passo (Q). A classe Q foi desconsiderada no período coberto por este relatório, devido à ausência de resultados promissores, tanto no trabalho, quanto na literatura. Como a classe normal possui um número muito maior de dados em relação às outras, a quantidade de batimentos normais foi reduzida, como indicado na Tabela 3.8.

Tabela 3.8: Número de batimentos com a classe N reduzida para os conjuntos de teste e de treinamento no problema de classificação de arritmias cardíacas.

Classe	Conjunto de treinamento	Conjunto de teste
N	8492	8359
S	755	1319
V	2538	2034
F	396	376

A partir dos resultados do Relatório Parcial, concluiu-se que a melhor entrada para a rede MLP no desempenho da classe S é o uso de 960 amostras do sinal original da primeira derivação. Assim, para classificar um determinado batimento, foram utilizados também o batimento anterior e o batimento posterior. Para otimizar a rede MLP para o problema de quatro classes,

variaram-se o número de neurônios e o número de camadas em um *grid search* a fim de se obter o melhor desempenho.

A estrutura da rede MLP que obteve as maiores métricas é composta por duas camadas ocultas, a primeira com 32 neurônios e a segunda com 16, e funções de ativação ReLU em ambas. Na camada de saída, foram utilizados 4 neurônios com função *Softmax*. Para trabalhar com as classes desbalanceadas, usou-se a função custo de entropia cruzada categórica com pesos calculados na proporção inversa do número de dados de cada classe. Considerou-se ainda o algoritmo de otimização de Adam [21] com $\beta_1 = 0{,}9$ e $\beta_2 = 0{,}99$, e o algoritmo de retropropagação com passo de aprendizado $\eta = 0{,}001$, 250 épocas, mini-batches de tamanho $k = 2048$ e inicialização de Xavier [101] para os pesos. As métricas obtidas para essa MLP estão indicadas na Tabela 3.9 e um resumo da estrutura é apresentado na Tabela 3.10.

Tabela 3.9: Resultados (%) obtidos para a rede MLP para o problema de quatro classes.

Modelo	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
32-16-4, 250 épocas, sem *dropout*	77,15	82,39	88,72	57,24	57,59	82,06	71,05	3,99	2,77

Tabela 3.10: Rede MLP utilizada para o problema de quatro classes.

	Caracterização da Rede MLP
Camada de entrada	960
Camada oculta 1	32 neurônios
Camada oculta 2	16 neurônios
Camada de saída	4 neurônios
Função de ativação	ReLU (ocultas) Softmax (saída)
Função custo	Entropia cruzada
Passo de aprendizado	0,001
Tamanho do *mini* batch	2048
Otimizador	Adam, com $\beta_1 = 0{,}9$ e $\beta_2 = 0{,}99$

Realizou-se também um teste para avaliar como a divisão dos dados dos pacientes influencia no desempenho. Batimentos de um mesmo paciente foram usados tanto no conjunto de teste quanto no de treinamento em uma rede MLP com os mesmo parâmetros da Tabela 3.10. O desempenho obtido está mostrado na Tabela 3.11.

Tabela 3.11: Resultados (%) obtidos para a rede MLP com as mesmas configurações da Tabela 3.10 para a divisão que considera cada batimento independente do outro.

Modelo	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
32-16-4 250 épocas, sem *dropout*	93,90	95,85	96,82	85,03	75,04	94,06	96,37	74,87	82,78

Os resultados obtidos com essa divisão são superiores aos da Tabela 3.9 para todas as métricas. Isso mostra como a mistura dos batimentos aumenta significativamente os acertos da classificação, mesmo utilizando uma rede MLP com configurações idênticas.

Além disso, aumentando-se suficientemente o número de épocas, pode-se melhorar o desempenho de P da classe S como mostrado na Tabela 3.12.

Tabela 3.12: Resultados (%) obtidos para a rede MLP com as mesmas configurações da Tabela 3.10 e 1000 épocas para a divisão que considera cada batimento independente do outro.

Modelo	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
32-16-4 1000 épocas, sem *dropout*	94,45	97,03	96,43	82,09	81,77	93,71	96,67	76,13	81,23

Os valores obtidos com essa rede não são realistas. Em geral, ao treinar-se as redes MLP com número de épocas maior do que 500, seguindo a divisão recomendada pela AAMI, foi observado um *overfitting* que prejudicava significativamente o desempenho da rede para o conjunto de teste. A Tabela 3.12, no entanto, apresenta os resultados de uma rede treinada com 1000 épocas que obteve métricas com valores acima de 75%, evidenciando como algoritmos que usam batimentos de um mesmo paciente nos dois conjuntos são beneficiados.

Por fim, a rede MLP e o efeito das entradas sobre o desempenho foram explorados com maiores detalhes no Relatório Parcial, e essas atividades estão descritas nos Apêndices F e G.

3.5.3 Classificação de Arritmias usando CNN

As configurações de CNN simuladas foram arquiteturas encontradas na literatura para o uso do sinal original como entrada da rede. Nessas arquiteturas, utiliza-se uma CNN 1D para aplicação em sequências temporais, em que as operações de convolução e subamostragem (*pooling*) ocorrem em apenas uma dimensão, que é o vetor de valores de tensão do sinal de ECG. Como a CNN possui essas operações, é possível aproveitar a propriedade intrínseca de extração de características dessas redes. No entanto, diversos trabalhos em que essa abordagem foi utilizada

não seguiram as recomendações de divisão dos dados da AAMI, apresentando métricas de até 99% de Acurácia.

Em sua maioria [1,2,6,10,13,102,103], os conjuntos de dados foram divididos aleatoriamente, com uma certa porcentagem para o treinamento e o restante para teste. Em alguns casos, realizou-se a técnica de *k-fold cross validation*, dividindo-se o banco de dados em k conjuntos de mesmo tamanho e usando-se (k-1) para o treinamento e 1 para o teste. Assim, a utilidade de se empregar CNNs para o problema de classificação de arritmias é questionável. O trabalho procurou avaliar o desempenho dessas estruturas para a divisão realista dos dados.

Dessa forma, as arquiteturas de [1, 102] foram simuladas. A configuração da CNN que obteve o melhor desempenho, com as modificações de alguns hiperparâmetros das arquiteturas originais, está indicada na Tabela 3.13 e ilustrada na Figura 3.23. Nessa arquitetura, usou-se a função custo de entropia cruzada categórica com pesos calculados na proporção inversa do número de dados de cada classe. Considerou-se o algoritmo de otimização de Adam [21] com $\beta_1 = 0{,}9$ e $\beta_2 = 0{,}99$, e o algoritmo de retropropagação com passo de aprendizado $\eta = 0{,}001$, 100 épocas, mini-batches de tamanho $k = 2048$ e inicialização de Xavier [101] para os pesos. Os resultados dessa rede estão apresentados na Tabela 3.14.

Tabela 3.13: Arquitetura da CNN utilizada.

Camadas	Tamanho do *Kernel*	Stride	Saída	*Dropout*
Convolucional	27	1	934×3	–
Max-Pooling	2	2	467×3	–
Convolucional	14	1	454×10	–
Max-Pooling	2	2	227×10	–
Convolucional	3	1	225×10	–
Max-Pooling	2	2	112×10	–
Convolucional	4	1	109×10	–
Max-Pooling	2	2	54×10	–
Totalmente conectada	–	–	30	0,2
Totalmente conectada	–	–	10	0,1
Totalmente conectada	–	–	4	–

Tabela 3.14: Resultados (%) obtidos para a CNN para o problema de 4 classes.

Modelo	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
CNN	77,99	84,03	87,70	31,84	41,46	89,68	74,39	42,55	31,75

Em comparação com a MLP, os valores de métricas da classe S obtidos com a CNN foram piores. Assim como apontado por outros autores [15, 25, 35, 104], a classe S é a que apresenta a

maior dificuldade em ser identificada corretamente, por assemelhar-se à morfologia da classe N e pela menor quantidade de dados da classe S. De fato, nas simulações com as CNNs, percebeu-se que muitas vezes batimentos pertencentes à classe S eram classificados equivocadamente como pertencentes à classe N.

Em seguida, testou-se a concatenação das características extraídas pela CNN com as características extraídas manualmente da FS1 de [24], como indicado na Figura 3.24. Os hiperparâmetros das camadas convolucionais, das camadas de *Max-Pooling* e das camadas totalmente conectadas foram os mesmos, modificando-se apenas a entrada das camadas totalmente conectadas. A Tabela 3.15 indica os resultados dessa modificação.

Tabela 3.15: Resultados (%) obtidos para a CNN com a inclusão das características de [24].

Modelo	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
CNN	77,82	82,21	89,14	43,21	37,75	83,43	93,55	71,28	25,67

Novamente, os resultados da classe S foram piores do que na MLP. Como muitos artigos consideram as classes N, S e V como as principais e a AAMI prioriza a classificação das classes S e V [24,84], decidiu-se por não se prosseguir com o uso das CNNs, uma vez que seu desempenho foi considerado baixo para a classe S.

Encontraram-se também trabalhos em que espectogramas obtidos a partir da transformada de Fourier de curto prazo (*short-term Fourier transform* - STFT) foram usados como entrada de redes convolucionais 2D [4,11,105,106]. Do mesmo modo que a literatura de redes convolucionais citada anteriormente, esses artigos não seguiram as recomendações da AAMI. Porém, como é uma proposta recente, o trabalho investigou a possibilidade do uso de imagens de espectograma como entrada da rede. Os desempenhos seguindo as recomendações da AAMI não se mostraram promissores e, portanto, essa abordagem foi descartada.

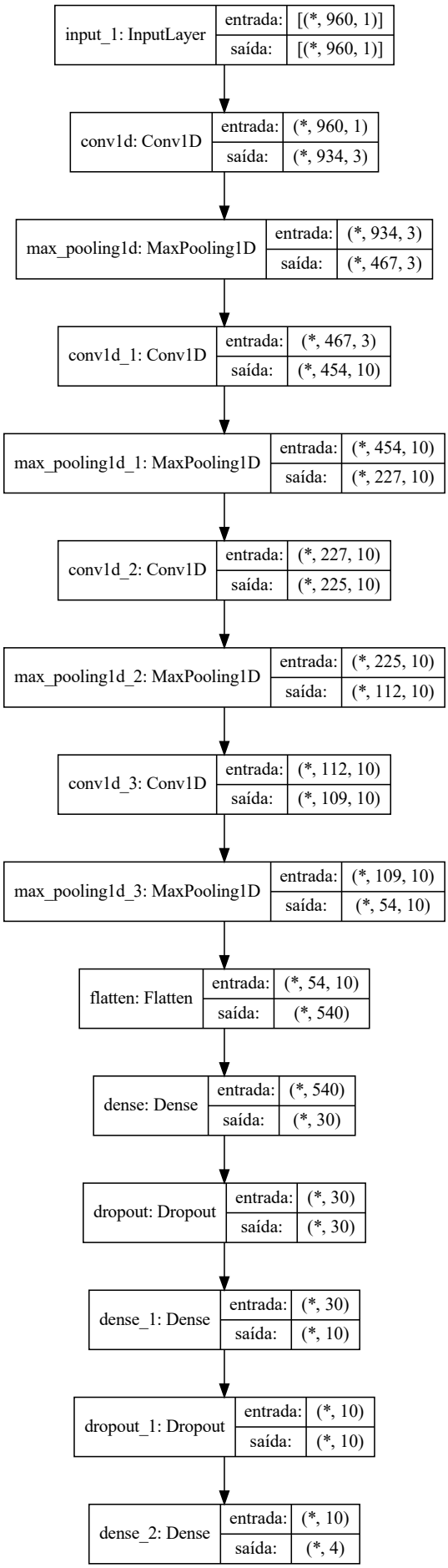


Figura 3.23: Arquitetura da CNN implementada para os resultados da Tabela 3.14. Imagem gerada com *Tensorflow* [31]. A primeira dimensão dos tensores, indicada com *, varia dependendo da quantidade de exemplos do conjunto utilizado.

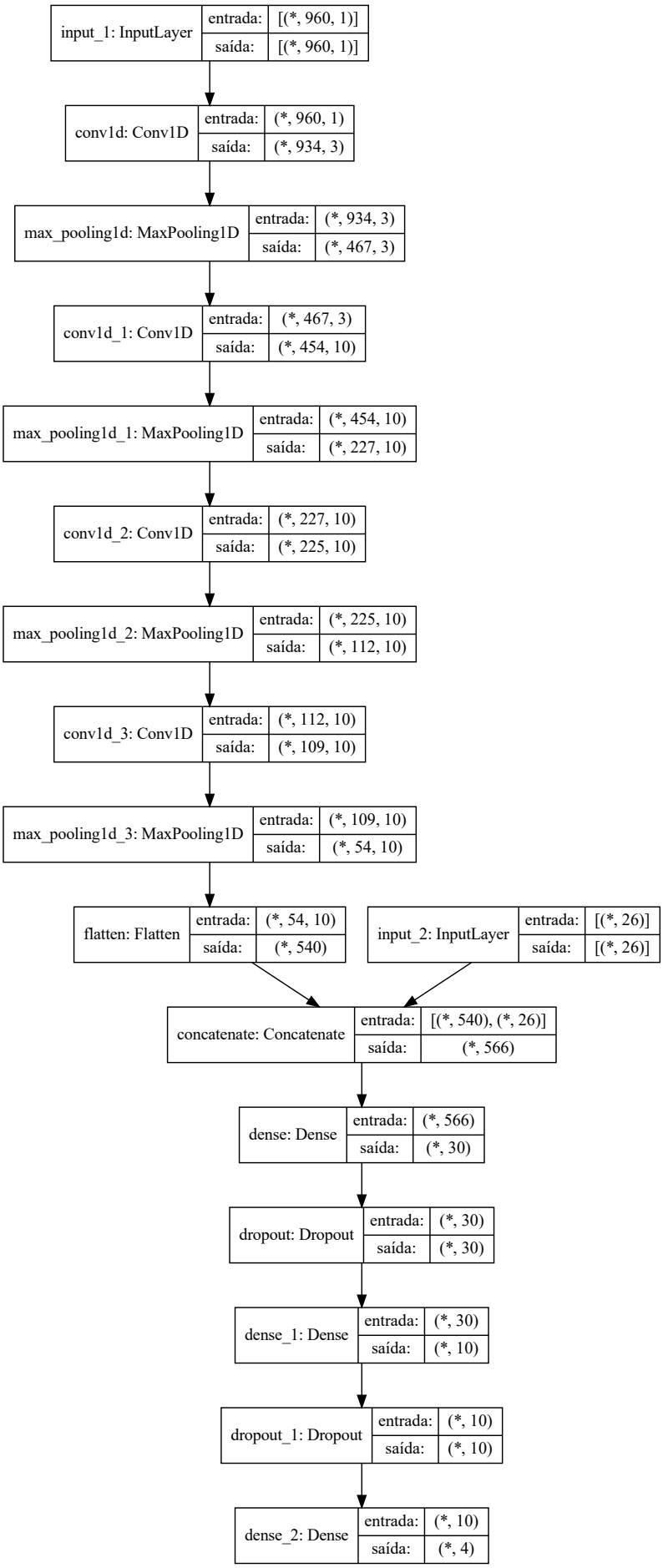


Figura 3.24: Arquitetura da CNN implementada para os resultados da Tabela 3.15. Imagem gerada com *Tensorflow* [31]. A primeira dimensão dos tensores, indicada com *, varia dependendo da quantidade de exemplos do conjunto utilizado.

3.5.4 Classificação de Arritmias usando LDA

Dentre os classificadores que seguem uma divisão realista dos pacientes, métodos baseados em LDA são os mais comuns, e também têm sido propostos em estudos recentes (ver, e.g., as referências de [25, 48]). Ao usar a divisão por pacientes proposta por [24], é um desafio otimizar MLPs e SVMs para obter resultados promissores nas classes S e V menos representadas. Uma das maiores vantagens da LDA é a facilidade em lidar com problemas gerados pelo desbalanceamento do número de dados das classes [25].

Foram feitas 3 simulações com a LDA, para analisar o seu comportamento quanto ao desbalanceamento dos dados. Para cada simulação, foram usadas as entradas FS1 e FS5 propostas por [24], para dois classificadores LDA, e as saídas dos classificadores foram combinadas por meio da operação descrita na Equação (3.5). A Figura 3.25 ilustra o modelo desenvolvido. Também testaram-se as entradas FS2 e FS6, mas as métricas obtidas foram menores, conclusão que também foi relatada por Chazal *et al* [24]. Nas LDAs, usou-se o método de decomposição em autovalores, com um estimador de covariância por máxima verossimilhança, probabilidades marginais iguais entre as classes, ou seja, de 1/4, e a menor redução de dimensionalidade possível. Como o número de classes do problema é 4, a transformação com a menor redução de dimensionalidade fornece 3 dimensões ($K - 1 = 3$).

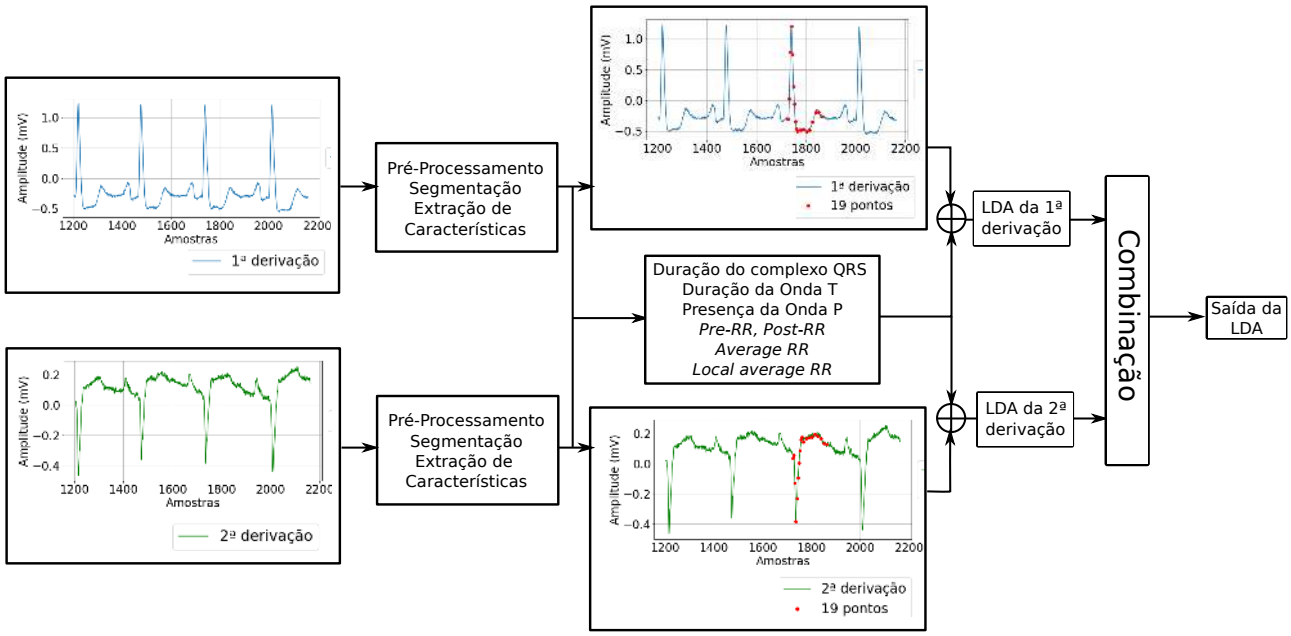


Figura 3.25: Modelo de classificação de arritmias usando LDA.

Primeiramente, foram selecionados apenas alguns dados para equilibrar o número de exemplos das classes, usando como referência a quantidade da classe F, que era a menos representada. Foi realizada a subamostragem dos dados para aproximadamente 376 exemplos com o cuidado

de permitir que todos os pacientes contribuíssem com as classes presentes em suas gravações e evitar que os dados selecionados pertencessem a um mesmo indivíduo, o que não seria realista e excluiria pacientes. Além disso, os dados foram escolhidos de forma aleatória, para evitar que fossem usados batimentos sequenciais. As quantidades obtidas após essa etapa para os conjuntos de teste e de treinamento estão indicadas na Tabela 3.16, e os resultados das simulações são apresentados na Tabela 3.17.

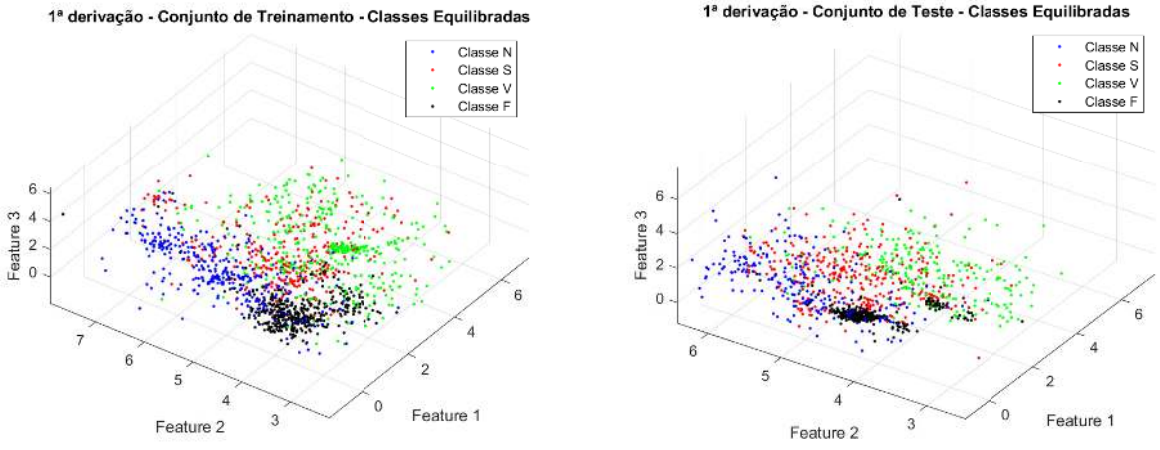
Tabela 3.16: Número equilibrado de batimentos para os conjuntos de teste e de treinamento no problema de classificação por LDA.

Classe	Conjunto de treinamento	Conjunto de teste
N	352	352
S	393	399
V	428	237
F	396	376

Tabela 3.17: Resultados (%) obtidos no problema de classificação por LDA para os dados com quantidades equilibradas.

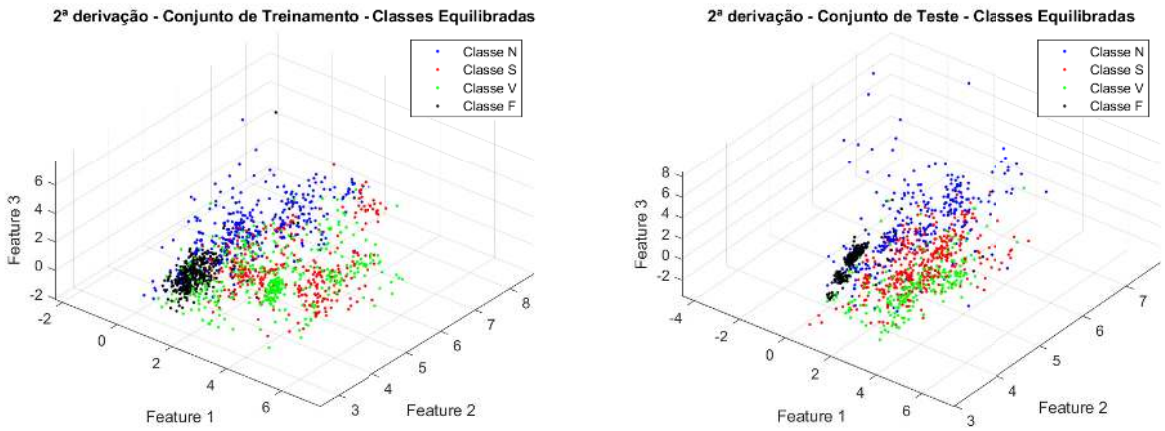
Modelo com dados equilibrados	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
Classificador da 1ª derivação	51,69	75,28	41,60	56,39	69,23	49,79	67,82	25,80	43,11
Classificador da 2ª derivação	65,03	72,16	66,15	38,10	62,81	53,16	58,06	94,41	68,40
Classificador final	69,57	79,55	64,97	48,37	68,20	57,38	66,02	90,43	76,92

A projeção dos dados para 3 dimensões usando a LDA calculada com as quantidades de dados equilibradas entre as classes pode ser visualizada na Figura 3.26 para a primeira derivação e na Figura 3.27 para a segunda derivação.



(a) Conjunto de Treinamento (b) Conjunto de Teste

Figura 3.26: Projeções da primeira derivação com a LDA para o número equilibrado de dados.



(a) Conjunto de Treinamento

(b) Conjunto de Teste

Figura 3.27: Projeções da segunda derivação com a LDA para o número equilibrado de dados.

A projeção dos dados da segunda derivação mostra uma sobreposição maior da classe S com as classes N e V em comparação com a projeção da primeira derivação. Esse fato também pode ser notado pelos valores das métricas da classe S na Tabela 3.17, que foram maiores na primeira derivação. Por outro lado, a projeção da segunda derivação mostra uma separação mais visível da classe F com relação às demais, o que também é notável pelo aumento da Se de 25,80% para 94,41% e da P de 43,11% para 68,40%. Assim, a combinação de LDAs de diferentes derivações é interessante para incorporar as contribuições de cada derivação, que auxiliam de forma mais ou menos relevante a identificação de cada classe.

Simulou-se então o caso em que a classe N tem seus dados reduzidos, com as quantidades da Tabela 3.8, enquanto utilizam-se todos os dados disponíveis das outras classes. Os resultados estão apresentados na Tabela 3.18.

Tabela 3.18: Resultados (%) obtidos no problema de classificação por LDA para a classe N reduzida.

Modelo com classe N reduzida	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
Classificador da 1ª derivação	69,82	80,00	93,34	40,41	33,09	51,08	65,39	48,14	10,52
Classificador da 2ª derivação	62,15	70,57	92,24	43,67	23,63	33,87	63,56	92,82	16,13
Classificador final	69,85	79,23	93,28	46,25	30,95	44,05	70,61	83,78	18,06

Por fim, usaram-se todos os dados disponíveis, com as quantidades da Tabela 3.19 e os resultados da Tabela 3.20.

Tabela 3.19: Número total de batimentos para os conjuntos de teste e de treinamento no problema de classificação por LDA.

Classe	Conjunto de treinamento	Conjunto de teste
N	40098	40052
S	755	1319
V	2538	2034
F	396	376

Tabela 3.20: Resultados (%) obtidos no problema de classificação por LDA para todos os dados.

Modelo com todos os dados	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
Classificador da 1ª derivação	77,04	79,90	98,45	40,33	16,86	51,47	47,53	39,10	2,49
Classificador da 2ª derivação	67,35	69,61	98,23	43,59	10,14	33,63	41,91	93,09	4,33
Classificador final	75,71	78,18	98,50	46,25	15,84	44,84	49,89	82,18	4,90

Comparando-se as Tabelas 3.18 e 3.20, é possível perceber que, apesar das sensibilidades de todas as classes serem próximas nos dois casos, as precisões das classes S, V e F são menores para a última simulação, com o maior desbalanceamento. Assim, mesmo atribuindo maiores pesos às classes menos representadas, não é recomendável usar quantidades tão diferentes de dados para cada classe.

Por outro lado, escolher apenas 400 batimentos de um banco tão diverso de modo a incluir todos os pacientes piora significativamente as métricas da classe N. Isso pode ser notado comparando-se as Tabelas 3.17 e 3.18. Essa piora ocorre pois essa classe é a que mais sofre redução de dados. Considerando que 40098 exemplos de batimentos cardíacos normais são reduzidos a 352 e que a escolha inclui morfologias de diferentes pacientes, é de se esperar que o desempenho piore. Pode-se concluir que, ao lidar com o problema de desbalanceamento de classes, não é interessante usar todos os exemplos possíveis, assim como não é interessante apenas subamostrar e descartar grandes quantias de exemplos. Cabe observar que as Figuras 3.26 e 3.27 ilustram como essa escolha de dados pode tornar as projeções do conjunto de treinamento e de teste diferentes.

3.5.5 Classificação de Arritmias usando RNN

Assim como foi feito na MLP, 960 amostras do sinal original da primeira derivação foram usadas como entrada para as RNNs implementadas no projeto. Foram simuladas RNNs compostas por três blocos LSTM, ou seja, três passos de tempo, e cada LSTM com uma entrada de 320

amostras, como ilustrado na Figura 3.28. Adotou-se o problema de 4 classes, com a classe N reduzida como na Tabela 3.18.

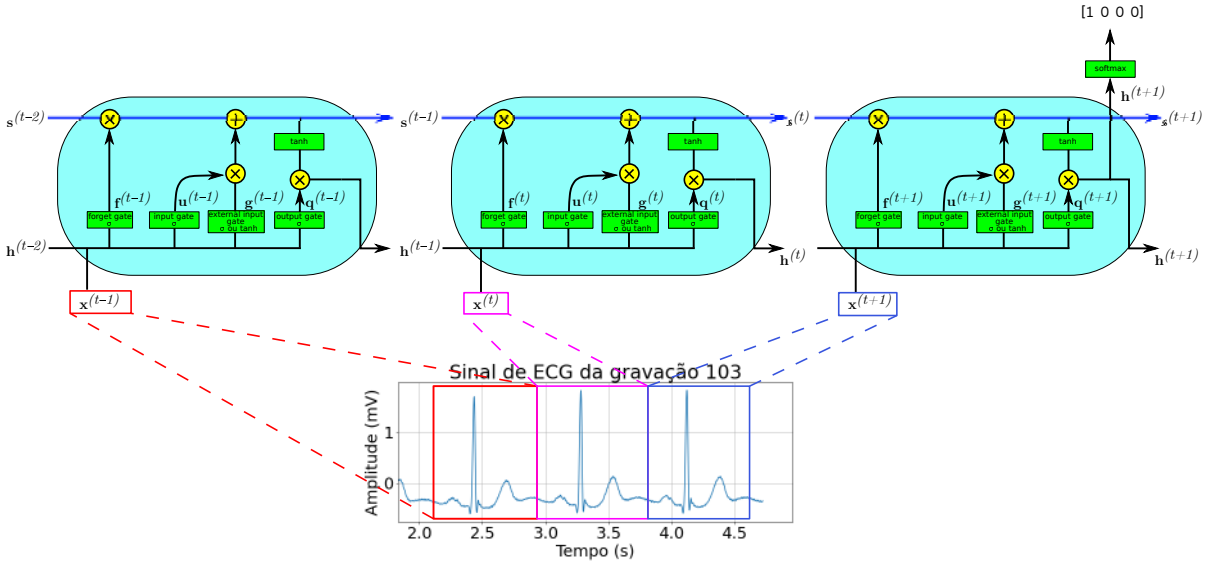


Figura 3.28: RNN com três passos de tempos.

O número de camadas e o número de épocas foram ajustados usando *grid search* a fim de se obter o melhor desempenho. Primeiramente, usando o algoritmo Adam [21], com $\beta_1 = 0{,}9$ e $\beta_2 = 0{,}99$, inicialização de Xavier [101] para os pesos da entrada, inicialização ortogonal para os pesos recorrentes, 100 épocas, passo de aprendizado de 0,001, e sem o uso de *dropout*, a rede foi treinada variando-se o número de neurônios de 8 a 256, acrescentando 4 neurônios em cada nova simulação.

Os resultados obtidos com um número pequeno de neurônios, entre 8 e 64, foram de sensibilidades e precisões baixas para todas as classes. As classes N e V apresentaram sensibilidades menores do que 75%, e a acurácia da rede foi menor do que 70%. Essas métricas aumentaram com o aumento do número de neurônios, como pode ser observado na Tabela 3.21. O melhor desempenho registrado foi com 72 neurônios. As redes com número de neurônios maior do que 100 apresentaram métricas das classes N e V semelhantes às obtidas para a de 72 neurônios, às vezes com uma pequena melhora nos resultados da classe N, mas apresentaram métricas menores para a classe S.

Tabela 3.21: Resultados (%) obtidos para diferentes números de neurônios em uma RNN de uma camada, treinada com 100 épocas.

Número de neurônios	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
8	47,80	44,24	75,75	27,98	15,65	81,86	42,46	12,23	5,88
32	67,18	74,46	81,20	29,34	22,62	72,17	68,37	11,17	8,71
72	74,43	79,34	84,42	55,42	47,65	77,53	71,13	15,16	12,31
100	72,95	77,17	84,92	56,33	42,80	78,07	67,81	9,57	9,02
128	71,21	80,50	79,68	21,91	19,41	77,43	85,37	3,99	5,15
256	72,83	83,24	81,48	20,92	32,43	74,58	69,75	14,10	10,27

A partir desses resultados, a rede com 72 neurônios foi escolhida para modificar outros hiperparâmetros e comparar os resultados. Variou-se o número de épocas de 100 a 700, acrescentando 100 épocas a cada nova simulação. Observou-se uma diminuição gradativa da sensibilidade da classe S com o aumento do número de épocas, apesar de resultados um pouco melhores para a classe F, como indicado na Tabela 3.22. Os maiores valores das métricas foram atingidos com 100 épocas.

Tabela 3.22: Resultados (%) obtidos para diferentes números de épocas em uma RNN de uma camada com 72 neurônios.

Número de épocas	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
100	74,43	79,34	84,42	55,42	47,65	77,53	71,13	15,16	12,31
300	68,08	72,56	82,00	36,24	29,18	80,14	62,07	15,16	14,92
500	69,28	76,54	81,28	26,46	28,42	76,60	62,52	18,62	14,96

O mesmo procedimento foi realizado usando-se dois batimentos normais como entrada dos dois primeiros blocos LSTM e o batimento com arritmia como entrada do último. No entanto, percebeu-se que as métricas de Sensibilidade e Precisão da classe S foram menores para a maioria dos casos. As métricas para 72 neurônios estão apresentadas na Tabela 3.23. Como o desempenho foi pior, voltou-se a utilizar a entrada anterior.

Tabela 3.23: Resultados (%) obtidos para a mudança de abordagem da ordem de batimentos na entrada.

Número de neurônios	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
72	73,16	83,30	82,15	16,92	28,81	80,07	74,50	7,71	5,19

Implementando um *dropout* de 0,2 na camada da rede, mas sem usar *dropout* nas conexões recorrentes, para 100 épocas e 72 neurônios, os resultados foram menores do que para o caso sem *dropout*. Porém, para maiores números de épocas, os resultados com *dropout* são melhores

do que sem, como esperado, uma vez que o *dropout* evita o *overfitting*. A sensibilidade e a precisão da classe F foram maiores. Os resultados para essas simulações estão mostrados na Tabela 3.24 e podem ser comparados com os da Tabela 3.22.

Tabela 3.24: Resultados (%) obtidos para diferentes números de épocas em uma RNN de uma camada com 72 neurônios e *dropout* de 0,2.

Número de épocas	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
100	65,41	67,48	83,03	47,23	27,08	76,06	68,94	25,53	13,026
300	71,40	76,41	83,61	40,03	34,49	80,38	68,61	21,54	15,58
500	77,14	85,78	84,81	32,90	48,93	78,81	67,69	31,38	32,42

Foram simuladas também RNNs compostas por duas camadas de 3 blocos LSTM, variando-se o número de neurônios nas duas camadas. No entanto, as métricas obtidas foram menores, e não se notou uma melhora com o aumento do número de camadas.

Por fim, para as RNNs as melhores estruturas encontradas foram de 72 neurônios, 100 épocas e sem *dropout* e de 72 neurônios, *500* épocas e com *dropout* de 0,2.

3.5.6 Classificação de Arritmias usando Combinações de Classificadores

Em geral, classificadores usados em uma aplicação específica atingem diferentes graus de sucesso, apoiados em conjuntos de características distintos. Segundo [107], a combinação de classificadores em um único sistema de reconhecimento de arritmias ajuda a integrar o conhecimento adquirido pelos diferentes modelos. Apesar disso, a combinação de classificadores foi pouco explorada na literatura para o diagnóstico automático de arritmias seguindo as recomendações da AAMI [25].

Em [24], as combinações dos classificadores da primeira e da segunda derivação obtiveram maiores acurácias em todos os resultados em comparação ao uso de apenas um classificador. Neste projeto, usou-se a operação de multiplicação elemento a elemento proposta em [24] e descrita na Equação (3.5) para combinar os classificadores. Essa combinação fornece resultados mais confiáveis em probabilidades que foram estimadas como zero, reduzindo o erro total das redes e a incidência de falsos negativos [107].

Foram feitas combinações da rede MLP com a LDA (MLP–LDA), da rede MLP com a RNN (MLP–RNN), da RNN com a LDA (RNN–LDA) e desses três classificadores juntos (MLP–RNN–LDA), utilizando-se as estruturas que obtiveram melhores desempenhos. A arquitetura da MLP utilizada foi a da Tabela 3.10. Já na RNN, testaram-se duas arquiteturas

na combinação com a LDA para decidir qual estrutura resultaria no melhor desempenho para as combinações.

Combinando-se a rede recorrente de uma camada, 72 neurônios, 100 épocas e sem *dropout*, da Tabela 3.21, e a rede de 72 neurônios, 500 épocas, com *dropout*, da Tabela 3.24, com a saída combinada dos modelos da LDA, da Tabela 3.18, foram obtidos os resultados da Tabela 3.25.

Tabela 3.25: Resultados (%) obtidos para combinações das RNNs com a LDA.

Modelo que foi combinado com a LDA	*Acc*	N		S		V		F	
		Se	*P*	*Se*	*P*	*Se*	*P*	*Se*	*P*
72 neurônios, 100 épocas, sem *dropout*	81,68	89,58	89,36	55,72	56,11	73,60	86,13	40,69	23,47
72 neurônios, 500 épocas, com *dropout*	82,83	91,60	89,07	37,15	55,37	79,99	79,56	63,56	42,91

As métricas de *Se* e *P* da classe S foram menores no segundo caso, com uma diferença significativa na *Se*, mas as métricas da classe F foram maiores. Como muito artigos consideram as classes N, S e V como as principais, optou-se por utilizar a estrutura da RNN com 72 neurônios, 100 épocas e sem *dropout* para ser comparada com os demais classificadores.

As métricas obtidas para cada simulação das combinações estão apresentadas na Tabela 3.26, junto ao desempenho dos classificadores individuais, para comparação dos resultados.

Tabela 3.26: Resultados (%)obtidos para as combinações dos classificadores com melhores desempenhos, junto ao desempenho dos classificadores separadamente para comparação. Os dois melhores resultados para cada métrica estão em negrito.

Métodos propostos	*Acc*	N			S			V			F			wF1 4 classes	wF1 3 classes	mF1
		Se	*P*	*F*1	*Se*	*P*	*F*1	*Se*	*P*	*F*1	*Se*	*P*	*F*1			
MLP	77,2	82,4	88,7	85,4	**57,2**	57,6	57,4	82,1	71,1	76,2	4,0	2,8	3,3	78,3	80,7	73,0
RNN	74,4	79,3	84,4	81,8	55,4	47,7	51,2	77,5	71,1	74,2	15,2	12,3	13,6	75,1	77,0	69,1
LDA	69,9	79,2	**93,3**	85,7	46,3	31,0	37,1	44,1	70,6	54,3	**83,8**	**18,1**	**29,7**	73,4	74,8	59,0
MLP–LDA	**84,2**	**90,4**	**91,1**	**90,8**	56,8	**68,0**	**61,9**	**89,3**	84,1	**86,6**	15,4	12,6	13,9	**84,5**	**86,8**	**79,8**
MLP–RNN	78,8	84,3	87,1	85,7	56,8	57,7	57,2	83,9	73,4	78,3	4,52	5,7	5,0	78,8	81,2	73,7
RNN–LDA	81,7	89,6	89,4	89,5	55,7	56,1	55,9	73,6	**86,1**	79,4	**40,7**	**23,5**	**29,8**	82,3	83,9	74,9
MLP–RNN–LDA	**84,4**	**91,1**	89,2	**90,1**	**57,2**	**71,2**	**63,5**	**87,9**	**85,1**	**86,5**	12,2	13,3	12,7	**84,2**	**86,5**	**80,0**

Em geral, a combinação apresentou métricas maiores em comparação com os modelos individuais. Analisando-se os classificadores propostos neste trabalho individualmente, é possível observar que a MLP apresenta as métricas gerais de *Acc* e de $wF1$ maiores do que a RNN e a LDA, além de $F1$ maiores para as classe N, S e V. Para essas classes, a MLP também atinge os valores de $F1$ usuais da literatura, porém, para a classe S, obtém um valor de $F1$ de 57,4%, que é maior do que o valor obtido em todos os outros trabalhos considerados. Para a classe V, o valor de $F1$ é de 76,2%, que se encontra na média dos demais trabalhos. Com relação à classe N, o desempenho da MLP é um pouco pior. Observa-se que o melhor classificador individual

para a classe F é a LDA, com a maior métrica de Se de 83,8%, e de $F1$ de 29,7%. Nota-se que as entradas propostas por [24] e utilizadas na LDA auxiliam a identificação da classe F.

Os resultados dos classificadores combinados foram melhores do que os dos classificadores individuais, em termos das métricas Acc e $wF1$. Comparando-se a combinação RNN–LDA com a RNN individual, por exemplo, aumentou-se a Acc de 74,4% da RNN para 81,7%, e o valor de $wF1$ de 75,1% da RNN para 82,3%. Além disso, o valor de $F1$ de todas as classes foi maior na combinação da RNN–LDA do que na RNN e na LDA separadamente.

Observando-se os valores em negrito, é possível perceber que a MLP–LDA e a MLP–RNN–LDA apresentaram os melhores desempenhos gerais dentre todas as simulações. Combinando-se a MLP com a LDA, a Acc de 77,2% da MLP aumentou para 84,2% e o valor de $wF1$ de 78,3% aumentou para 84,5%. Porém, ao realizar-se a combinação MLP–RNN–LDA, não é possível notar variação significativa das métricas em relação ao modelo MLP–LDA, com o valor de Acc aumentando apenas 0,2% e de $wF1$ aumentando 0,3%. Assim, a contribuição da RNN na combinação não foi relevante, além de levar a um aumento de 113476 parâmetros no modelo. Portanto, decidiu-se comparar o modelo MLP–LDA com os resultados da literatura. A Figura 3.29 ilustra essa combinação.

3.5.7 Comparação com a literatura

No Relatório Parcial, os resultados foram comparados com os artigos [24–26] que nortearam boa parte das decisões do projeto, sendo [26] uma implementação dos métodos de [108–112] seguindo as recomendações da AAMI. Em uma nova revisão bibliográfica, outros artigos considerados como estado da arte foram encontrados e foram considerados neste relatório.

Para realizar essa análise comparativa, foram utilizados os resultados da combinação MLP–LDA. A Tabela 3.27 apresenta a matriz de confusão obtida para esse modelo. A comparação desse modelo com os propostos na literatura é feita na Tabela 3.28.

Tabela 3.27: Matriz de confusão obtida para MLP–LDA.

		Classes Preditas			
		N	S	V	F
Classes Verdadeiras	N	7557	317	161	324
	S	370	749	183	17
	V	127	28	1817	62
	F	240	7	71	58

A seguir, os trabalhos da literatura cujos resultados aparecem na Tabela 3.28 são descritos brevemente. Em [24], desenvolveu-se uma LDA para classificar os batimentos em um problema de cinco classes. A estrutura desse modelo e as características utilizadas foram detalhadas na

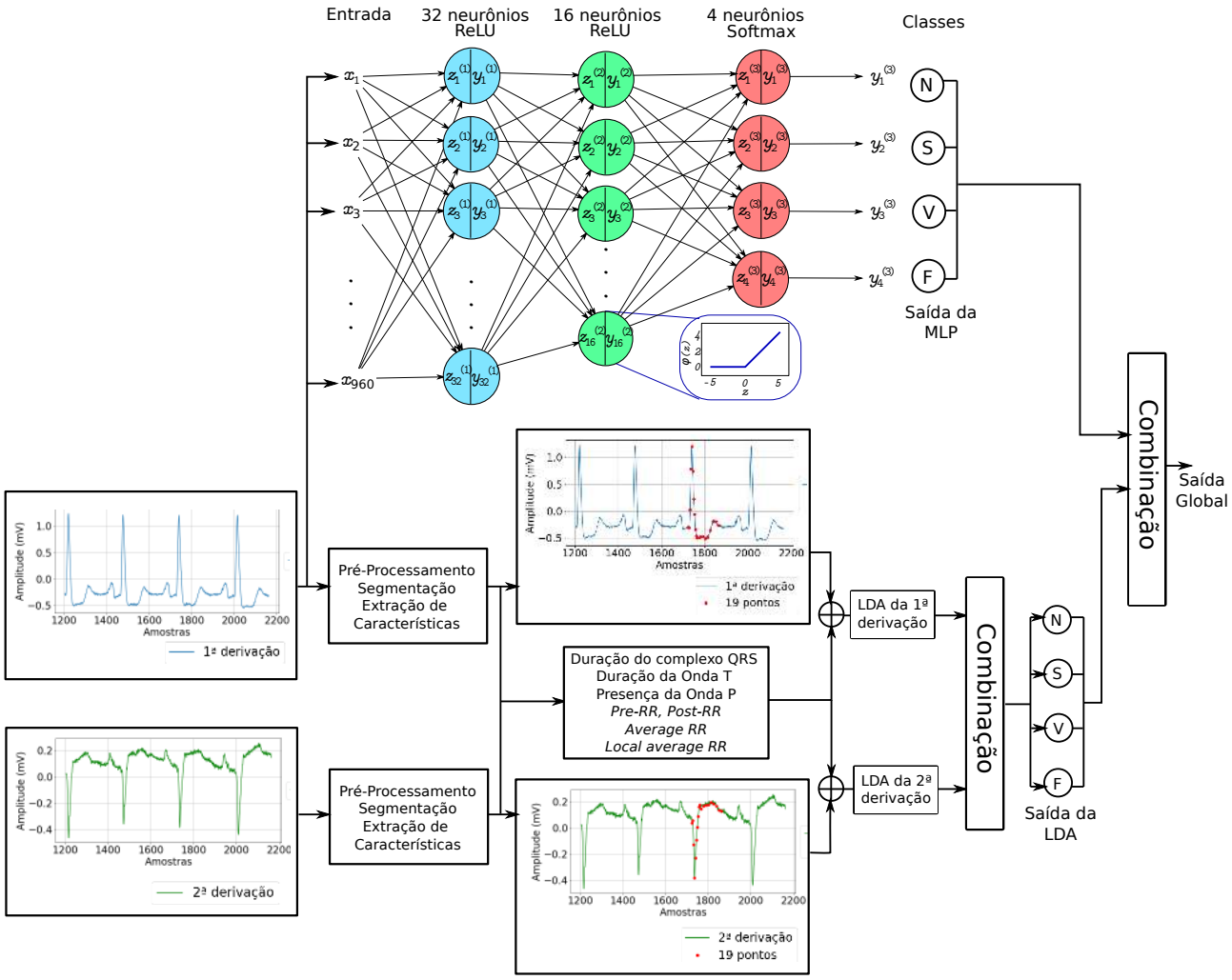


Figura 3.29: Modelo MLP–LDA.

Seção . Em [34], características temporais, morfológicas e estatísticas do sinal foram utilizadas junto a um algoritmo de *sequential forward floating search* para encontrar combinações de características ótimas, com classificadores baseados em LDA e MLP. Os resultados de [34] apresentados na Tabela 3.28 foram obtidos com uma rede MLP.

Em [35], foi proposta uma LDA com pesos e extração de características de intervalos RR e características morfológicas usando transformada de wavelet. Em [47], o ECG foi representado em três dimensões por meio do *temporal vectorcardiogram* (TVCG). Essa representação foi usada em redes complexas para extração de características, que, por sua vez, foram consideradas como entrada de um classificador SVM. Por fim, [48] aprimorou o modelo de [47] com o uso de *particle swarm optimization* para seleção de características. Para realizar as métricas de *wF1* para três e quatro classes, investigou-se o número de dados utilizado em cada trabalho para o cálculo da Equação (3.11).

A combinação MLP–LDA obtém valores de $F1$ para as classe S e V de 61,9% e 86,6% respectivamente, maiores do que todos os outros da literatura. Além disso, obtém o valor de

Tabela 3.28: Métricas (%) dos resultados das simulações e comparação com os valores da literatura. Os dois maiores resultados para cada métrica calculada com os resultados dos métodos propostos estão em negrito.

Métodos propostos	*Acc*	N			S			V			F			$wF1$ 4 classes	$wF1$ 3 classes	$mF1$
		Se	*P*	$F1$	*Se*	*P*	$F1$	*Se*	*P*	$F1$	*Se*	*P*	$F1$			
MLP–LDA	84,2	90,4	91,1	90,8	56,8	68,0	61,9	89,3	84,1	86,6	15,4	12,6	13,9	84,5	86,8	79,8

Métodos da literatura	*Acc*	N			S			V			F			$wF1$ 4 classes	$wF1$ 3 classes	$mF1$
		Se	*P*	$F1$	*Se*	*P*	$F1$	*Se*	*P*	$F1$	*Se*	*P*	$F1$			
Chazal [24]	85,9	86,9	99,2	92,6	75,9	38,5	51,1	77,7	81,9	79,7	89,4	8,6	15,7	89,7	90,3	74,5
Mar [34]	90,0	89,6	99,1	94,11	83,2	33,5	47,8	86,8	75,9	81,0	61,1	16,6	26,1	91,0	91,5	74,3
Llamedo[†][33]	93,0	95,0	98,0	96,5	77,0	39,0	51,8	81,0	87,0	83,9	–	–	–	–	93,9	77,4
Lin[†][35]	90,8	91,6	99,3	95,3	81,4	31,6	45,5	86,2	73,7	79,5	–	–	–	–	92,4	73,4
Garcia[†][47]	91,0	95,0	96,0	95,5	30,0	26,0	27,9	85,0	66,0	74,3	–	–	–	–	91,6	65,9
Luz[†][48]	92,4	94,0	98,0	96,0	62,0	53,0	57,2	87,3	59,4	70,7	–	–	–	–	92,9	74,6

[†] Autores otimizaram os métodos para o problema de três classes: N, S e V.

$F1$ de N de 90,8%, com uma diferença de 5,7% do maior valor dos demais trabalhos. No entanto, o valor de $F1$ da classe F é menor do que os de [24] e [34], que consideram o problema de quatro classes. Em relação ao desempenho geral, o valor de $wF1$ foi menor dos que os demais, com uma diferença máxima de 6,8%, no caso de quatro classes. Considerando três classes, essa diferença passa a ser de 7,1%. Dentre os classificadores que usaram as quatro classes, a diferença máxima desconsiderando a classe F foi de 4,7%.

Apesar dos valores de $F1$ maiores nas classes S e V, a combinação MLP–LDA obteve $wF1$ menor devido ao peso da classe N durante a ponderação no cálculo dessa métrica. Neste trabalho, não foram considerados todos os batimentos da classe N, pois o desbalanceamento dos dados na proporção original prejudicaria muito o treinamento dos classificadores, mesmo com a correção realizada pelos pesos na função custo. Avaliando-se a métrica $mF1$, ao atribuir pesos iguais às métricas $F1$ de todas as classes, o método proposto alcança valores maiores do que os relatados nos demais trabalhos.

4 Conclusões

Neste trabalho, foram estudados métodos computadorizados para a detecção e classificação de arritmias cardíacas. Os resultados dos classificadores desenvolvidos foram satisfatórios, alcançando os valores relatados na literatura e apresentando desempenhos superiores para determinadas classes de arritmia. A arquitetura que apresentou o melhor desempenho foi a combinação da rede MLP com a LDA, com o uso do sinal original da primeira derivação como entrada da rede MLP e de características extraídas das duas derivações como entrada da LDA. Esse modelo obteve melhores resultados do que os demais artigos nas métricas de *F1*-score das classes de batimentos supraventriculares ectópicos e de batimentos ventriculares ectópicos.

É importante ressaltar que as simulações realizadas e os artigos utilizados para comparação consideraram a divisão mais realista dos dados dos pacientes entre os conjuntos de treinamento e de teste. A divisão que considera os batimentos independentes uns dos outros influencia fortemente os resultados, gerando desempenhos muito melhores, porém, sob um viés clínico, não realistas.

Comparando-se os resultados dos diferentes métodos, também foi possível observar que a combinação de classificadores aumentou a maioria das métricas em comparação com os métodos individuais. Concluiu-se que essa combinação auxilia efetivamente a melhorar o desempenho geral do modelo na classificação de arritmias.

Avaliando-se o cronograma de atividades do projeto, mostrado na página 6, as atividades propostas foram concluídas no período previsto. Também foram realizados estudos adicionais sobre a teoria de transformada de wavelet, sobre banco de filtros e sobre análise de discriminantes lineares, além de simulações usando combinações dos classificadores, atividades que não estavam previstas no cronograma inicial.

Este projeto de pesquisa auxiliou no entendimento de disciplinas da graduação, reforçando conceitos de *Sistemas e Sinais*, *Processamento Digital de Sinais* e *Métodos Numéricos e Aplicações*. Além disso, foi possível aprimorar os conhecimentos da linguagem de programação *Python* e das bibliotecas *Tensorflow*, *Keras* e *Scikit-learn*, muito utilizadas dentro da área de

aprendizado de máquina.

A partir dos resultados obtidos neste trabalho, um resumo intitulado "Classificação de arritmias usando redes neurais perceptron multicamada" foi apresentado no *28º Simpósio Internacional de Iniciação Científica e Tecnológica da USP* (SIICUSP 2020) na área de Engenharia e recebeu menção honrosa pela Pró-Reitoria de Pesquisa da USP. Outro artigo sobre classificação de arritmias usando redes MLP foi submetido à *73ª Reunião Anual da Sociedade Brasileira para o Progresso da Ciência* (SBPC 2021). Um artigo sobre combinações de redes neurais e discriminantes lineares para classificação de arritmias foi submetido ao *XXXIX Simpósio Brasileiro de Telecomunicações e Processamento de Sinais* (SBrT). Estes artigos encontram-se anexos ao Relatório.

A Redes Neurais

O cérebro humano pode ser interpretado como um sistema de processamento de informações altamente complexo, não-linear e paralelo. Ele tem a capacidade de organizar as suas estruturas, os neurônios, para realizar determinados processos, como reconhecimento de padrões, percepções e controle motor, de modo muito mais rápido do que um computador digital [18]. A plasticidade de um sistema nervoso em desenvolvimento permite a sua adaptação ao meio ambiente ao redor. Essa característica é essencial para o funcionamento dos neurônios como unidades de processamento de informações, assim como também é fundamental para os neurônios de redes neurais artificiais [18].

Uma rede neural artificial é um processador massivo paralelo e distribuído, composto por unidades de processamento simples que possuem uma capacidade natural para guardar os conhecimentos provenientes da experiência e torná-los disponíveis para o uso [18]. A sua analogia com o cérebro decorre de dois aspectos:

- A rede adquire conhecimento do meio por um processo de aprendizado, chamado de algoritmo de aprendizado;
- Os coeficientes associados às conexões entre os neurônios, conhecidos como pesos sinápticos, são usados para guardar o conhecimento adquirido.

O algoritmo de aprendizado modifica os pesos sinápticos de uma rede para atender determinado objetivo, sendo uma abordagem que se aproxima da teoria de filtros lineares adaptativos. Porém, existe também a possibilidade de uma rede modificar a sua própria topologia, ideia motivada pelo fato de que neurônios no cérebro humano podem perecer e novas conexões sinápticas podem surgir [17, 20].

O poder computacional das redes neurais provém, além da sua estrutura paralela e distribuída, da sua propriedade de generalização, capaz de fornecer saídas razoáveis para entradas que não foram encontradas durante o aprendizado. Assim, as redes neurais possibilitam soluções aproximadas para problemas complexos de larga escala, oferecendo também diversas

propriedades úteis, como a não linearidade, o mapeamento de entrada e saída (aprendizado supervisionado) e a adaptabilidade [20].

O neurônio é uma unidade de processamento de informação fundamental para o funcionamento de uma rede neural. O modelo de um neurônio é composto por 3 elementos básicos:

1. Um conjunto de sinapses, com cada sinapse i conectada a um neurônio k, caracterizada por um peso próprio w_{ki} que multiplica um sinal de entrada x_i;

2. Um somador dos sinais de entrada multiplicados por suas respectivas contribuições, que realiza uma operação de combinação linear; e

3. Uma função de ativação φ, que geralmente é não linear e pode limitar a amplitude de saída de um neurônio.

O modelo também possui um nível de ajuste (*bias*) aplicado externamente que tem o efeito de aumentar ou diminuir a entrada da função de ativação, dependendo se é positivo ou negativo. O modelo de um neurônio está apresentado na Figura A.1.

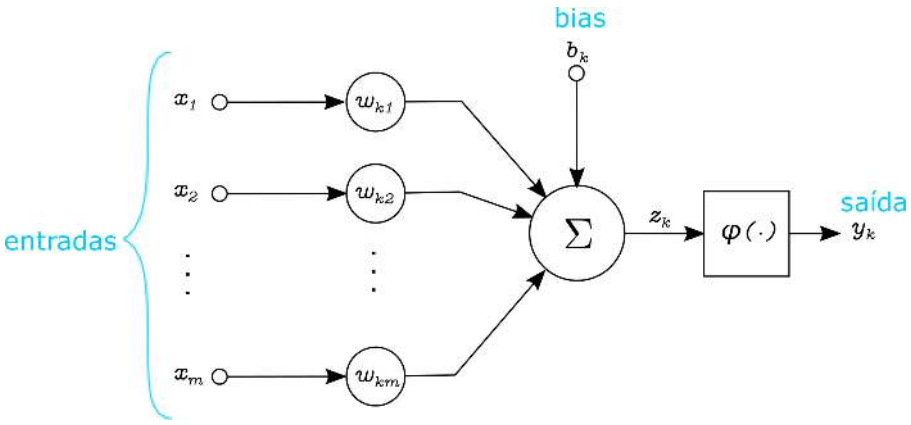


Figura A.1: Modelo de um neurônio.

As entradas do neurônio k podem ser agrupadas no vetor

$$\mathbf{x} = [x_1, x_2, \ldots, x_m]^T \tag{A.1}$$

e seus pesos sinápticos no vetor

$$\mathbf{w}_k = [w_{k1}, w_{k2}, \ldots, w_{km}]. \tag{A.2}$$

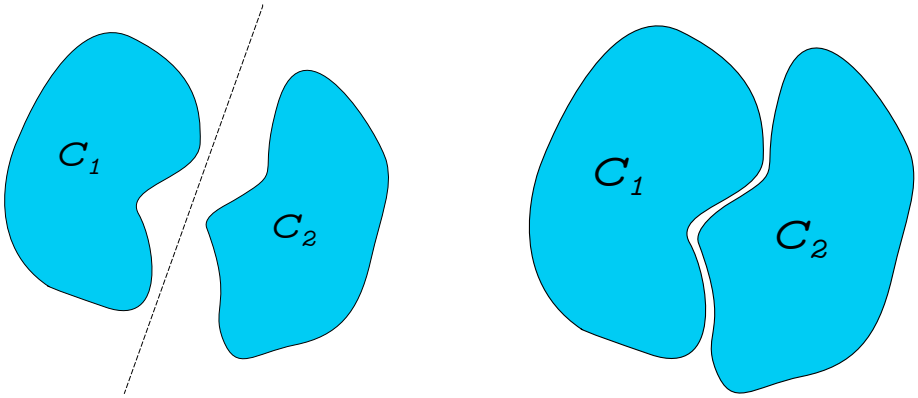
No modelo, a saída do somador é dada por

$$z_k = \sum_{i=1}^{m} w_{ki} x_i + b_k = \mathbf{w}_k \cdot \mathbf{x} + b_k, \tag{A.3}$$

e a aplicação da função de ativação a z_k fornece a saída y_k do neurônio, dada por

$$y_k = \varphi(z_k). \tag{A.4}$$

O *perceptron* de Rosenblatt é a forma mais simples de uma rede neural desenvolvida para a classificação de padrões ditos *linearmente separáveis*, consistindo de um único neurônio com uma saída binária. Em 1958, ao desenvolver um algoritmo para ajustar os parâmetros de sua rede neural, Rosenblatt provou que se os padrões usados para o treinamento de sua rede *perceptron* fossem retirados de duas classes C_1 e C_2 linearmente separáveis, o algoritmo convergia e era possível formar um hiperplano entre essas duas classes, como ilustrado na Figura A.2, prova que ficou conhecida como *Teorema de Convergência do Perceptron* [18].



(a) Linearmente separáveis (b) Não linearmente separáveis

Figura A.2: Classes linearmente e não linearmente separáveis.

Assim, por possuir uma única camada de rede, o *perceptron* de Rosenblatt possui limitações práticas, classificando apenas padrões linearmente separáveis. Essas limitações foram superadas com a rede *perceptron* multicamada (*Multilayer Perceptron* - MLP), caracterizada por três pontos principais [18]:

1. O modelo de cada neurônio da rede inclui uma função de ativação não linear que é *diferenciável*;

2. A rede possui uma ou mais camadas que são *ocultas* do ponto de vista da entrada e da saída; e

3. A rede exibe um alto grau de *conectividade*, determinada pelos pesos sinápticos.

No entanto, essas mesmas características dificultam a análise teórica da MLP, devido à presença de uma estrutura distribuída e não linear e de neurônios ocultos que tornam o processo de aprendizado complexo de se visualizar [20].

A Figura A.3 ilustra um neurônio k, com as definições dadas pelas Equações (A.1) a (A.4).

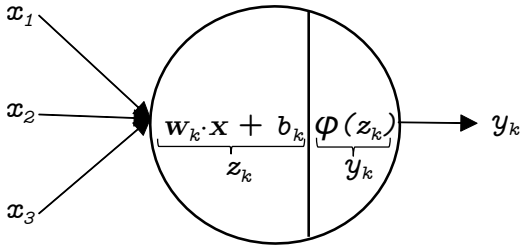


Figura A.3: Neurônio k da rede MLP.
Fonte: Adaptado de [17].

Uma arquitetura em grafo da rede MLP pode ser observada na Figura A.4. Essa rede possui quatro camadas, sendo três delas ocultas e uma de saída.

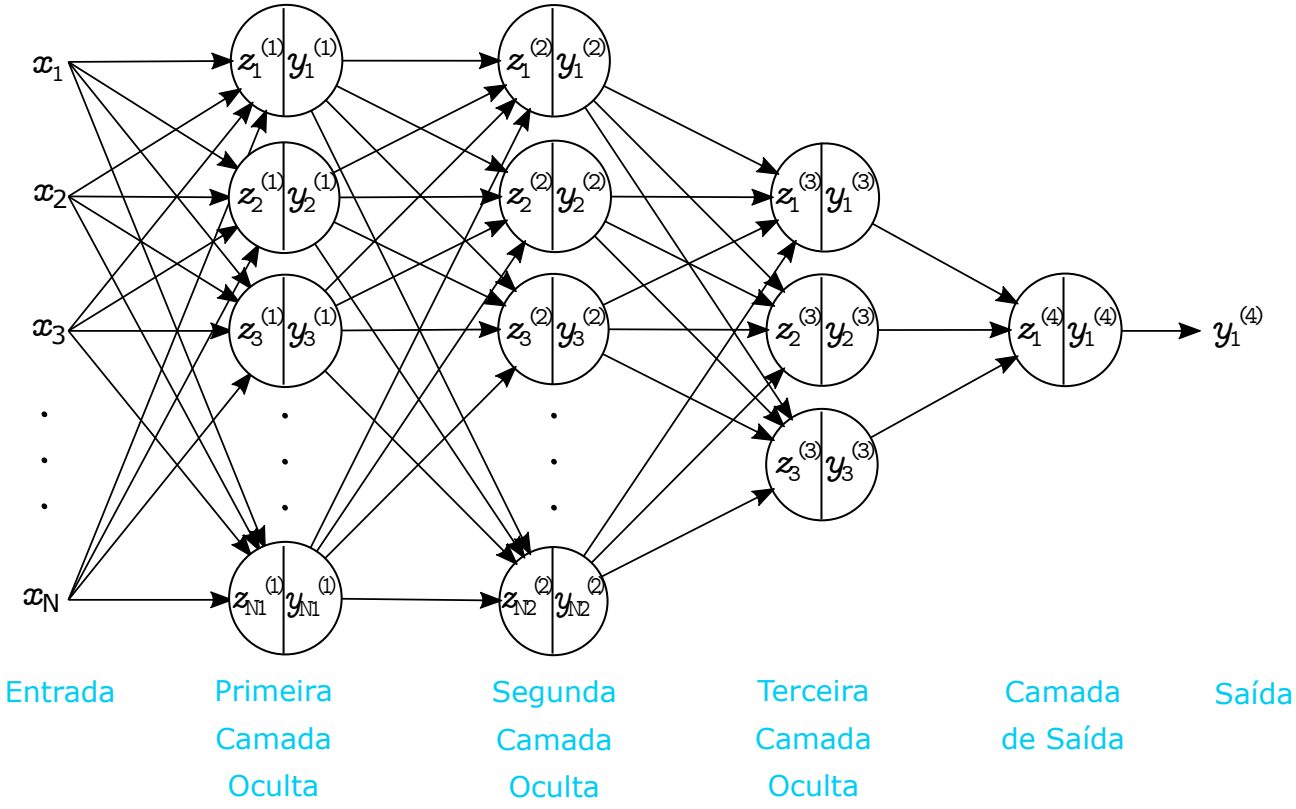


Figura A.4: Exemplo de uma rede MLP com quatro camadas (três ocultas e uma de saída).

Genericamente, cada camada j possui N_j neurônios, com $j = 1,2,\ldots,L$. No exemplo ilustrado na Figura A.4, $L = 4$. É possível definir os vetores $\mathbf{b}^{(j)}$, $\mathbf{z}^{(j)}$ e $\mathbf{y}^{(j)}$ de modo a representar o *bias*, a saída do somador e a saída de cada neurônio da camada j [113], sendo esses vetores dados respectivamente por

$$\mathbf{b}^{(j)} = \begin{bmatrix} b_1^{(j)} \\ b_2^{(j)} \\ \vdots \\ b_{N_j}^{(j)} \end{bmatrix}, \quad \mathbf{z}^{(j)} = \begin{bmatrix} z_1^{(j)} \\ z_2^{(j)} \\ \vdots \\ z_{N_j}^{(j)} \end{bmatrix} \quad \text{e} \quad \mathbf{y}^{(j)} = \begin{bmatrix} y_1^{(j)} \\ y_2^{(j)} \\ \vdots \\ y_{N_j}^{(j)} \end{bmatrix}. \tag{A.5}$$

Dessa forma, a matriz de pesos da camada j possui N_j linhas e N_{j-1} colunas, com cada linha i correspondente ao vetor de pesos do i-ésimo neurônio dessa camada, ou seja,

$$\mathbf{W}^{(j)} = \begin{bmatrix} \mathbf{w}_1^{(j)} \\ \mathbf{w}_2^{(j)} \\ \vdots \\ \mathbf{w}_{N_j}^{(j)} \end{bmatrix} = \begin{bmatrix} w_{11}^{(j)} & w_{12}^{(j)} & \dots & w_{1N_{j-1}}^{(j)} \\ w_{21}^{(j)} & w_{22}^{(j)} & \dots & w_{2N_{j-1}}^{(j)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N_j1}^{(j)} & w_{N_j2}^{(j)} & \dots & w_{N_jN_{j-1}}^{(j)} \end{bmatrix}. \tag{A.6}$$

Por fim, o vetor $\mathbf{z}^{(j)}$ pode ser obtido por

$$\mathbf{z}^{(j)} = \mathbf{W}^{(j)} \cdot \mathbf{y}^{(j-1)} + \mathbf{b}^{(j)}, \tag{A.7}$$

e a saída do neurônio por

$$\mathbf{y}^{(j)} = \varphi(\mathbf{z}^{(j)}). \tag{A.8}$$

Por definição, para $j = 0$, $\mathbf{y}^{(0)} = \mathbf{x}$.

As funções de ativação, denotadas por $\varphi(z)$, definem uma relação entrada-saída não linear [17]. Uma das funções de ativação mais comuns é a função sigmoidal, sendo um exemplo a função logística definida por

$$\varphi(z) = \frac{1}{1 + e^{-az}}, \tag{A.9}$$

em que a é um escalar real positivo, chamado de parâmetro de inclinação. Essa função é diferenciável, assume valores contínuos entre 0 e 1, e é, em geral, usada em classificações binárias, devido à interpretação probabilística de pertencimento (saída 1) ou não (saída 0) à classe em questão.

Por vezes, é interessante usar uma função de ativação com um intervalo de -1 a 1, como a função tangente hiperbólica, definida por

$$\varphi(z) = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{A.10}$$

Há também a função de ativação *Rectified Linear Unit* (ReLU) [17], dada por

$$\varphi(z) = max(0,z). \tag{A.11}$$

Redes neurais profundas que usam essa função treinam muito mais rápido quando comparadas com as redes que usam a função tangente hiperbólica. Atualmente, a recomendação padrão para a função de ativação é a *ReLU*. Essa função foi baseada no princípio de que os modelos são mais facilmente otimizados quando o seu comportamento é próximo do linear [19]. Sua

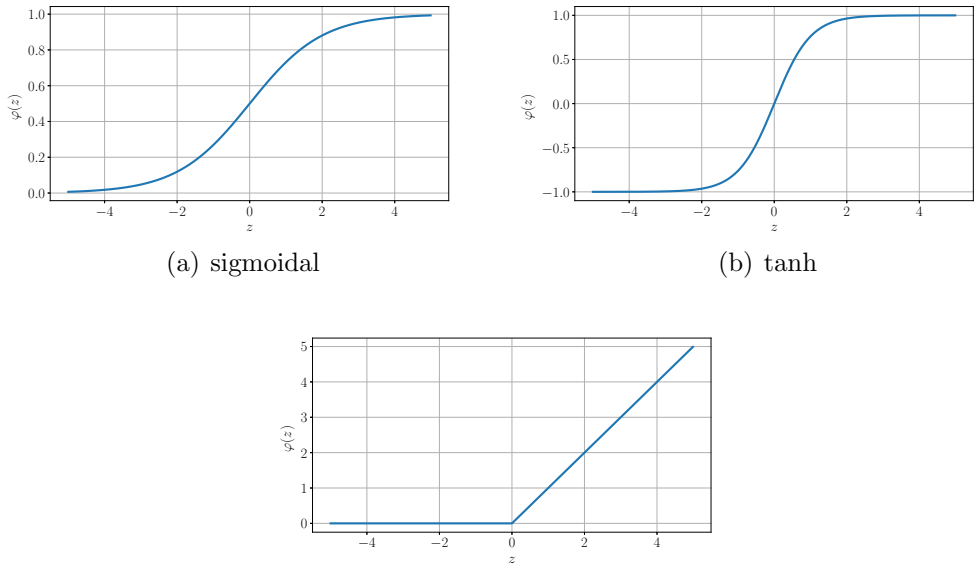
propriedade principal é de permitir que o algoritmo de aprendizado evite ficar parado em mínimos locais [17, 114].

Nos problemas de classificação entre diversas classes utiliza-se uma generalização da regressão logística. Nesses casos, a rede possui mais de um neurônio na saída, e utiliza-se a função de ativação *softmax* [19], definida por

$$\varphi(z_k^{(L)}) = \frac{e^{z_k^{(L)}}}{\sum_{i=1}^{N_L} e^{z_i^{(L)}}}. \tag{A.12}$$

Em cada exemplo de treinamento n, essa função é aplicada para cada neurônio k da última camada L, gerando, na saída da rede, um vetor $\mathbf{y}^{(L)}(n)$ de tamanho igual ao número de classes do problema.

As funções de ativação sigmoidal, tangente hiperbólica e *ReLU* são ilustradas na Figura A.5.



(a) sigmoidal

(b) tanh

(c) ReLU

Figura A.5: Funções de ativação.

Em geral, para treinar a MLP, usa-se o algoritmo de retropropagação (*back-propagation*), que é dividido em duas fases [18]:

1. A fase progressiva (*forward*), em que os pesos sinápticos da rede são fixados e a entrada é propagada camada a camada até atingir a saída;

2. A fase regressiva (*backward*), em que a saída obtida $\mathbf{y}^{(L)}$ é comparada com a resposta desejada. Dessa comparação, é produzido um sinal de erro que é propagado no sentido contrário da primeira fase. Nessa etapa, diversos ajustes são realizados nos pesos sinápticos da rede.

As amostras da resposta desejada $d_1(n), d_2(n), \cdots d_{N_L}(n)$ correspondentes à n-ésima entrada $\mathbf{x}(n)$ podem ser agrupadas no vetor

$$\mathbf{d}(n) = \begin{bmatrix} d_1(n) \\ d_2(n) \\ \vdots \\ d_{N_L}(n) \end{bmatrix}. \tag{A.13}$$

Assim, para cada exemplo no conjunto de treinamento

$$\Im = \{\mathbf{x}(n), \mathbf{d}(n)\}_{n=1}^{M}, \tag{A.14}$$

$\mathbf{d}(n)$ é comparado à $\mathbf{y}^{(L)}(n)$ e um sinal de erro é produzido. Para isso, é possível escolher diferentes funções custo, denotadas genericamente de J. Uma função custo compara o vetor $\mathbf{d}(n)$ com o vetor de saída $\mathbf{y}(n)^{(L)}$ por meio de uma função $\mathcal{L}(y_k^L(n), d_k(n))$, aplicada em cada neurônio k da camada de saída [17]. Assim, a função custo pode ser definida como

$$J(w,b) = \frac{1}{M}\sum_{n=1}^{M}\sum_{k=1}^{N_L}\mathcal{L}(y_k^{(L)}(n), d_k(n)). \tag{A.15}$$

O objetivo da rede é encontrar os valores de pesos e *bias* que minimizem J.

A função custo mais comum para problemas de regressão é a função do Erro Quadrático Médio (*Mean Square Error* - MSE) [18], dada por

$$J_{MSE} = \frac{1}{M}\sum_{n=1}^{M}\sum_{k=1}^{N_L}(y_k^{(L)}(n) - d_k(n))^2. \tag{A.16}$$

Para aplicações de classificação binária, isto é, para $N_L = 1$, é comum utilizar a função custo de Entropia Cruzada (*Cross-Entropy*) definida por

$$J_{CE} = -\frac{1}{M}\sum_{n=1}^{M}(d_1(n)log(y_1^{(L)}(n)) + (1 - d_1(n))log(1 - y_1^{(L)}(n))). \tag{A.17}$$

Na classificação entre múltiplas classes, usa-se a função custo de Entropia Cruzada Categórica (*Categorical Cross-Entropy*), dada por

$$J_{CCE} = -\frac{1}{M}\sum_{n=1}^{M}\sum_{k=1}^{N_L}(y_k^{(L)}(n)log(d_k(n))). \tag{A.18}$$

Observa-se que, apesar da saída referente à cada exemplo do conjunto de treinamento ser um vetor, as funções custo resultam em um escalar.

Escolhida a função custo a ser minimizada, inicia-se então o algoritmo de retropropagação. Por meio do método do gradiente estocástico, os pesos são atualizados de modo a minimizar o

valor de J. Utilizando o método do gradiente estocástico, aplica-se a correção $\Delta w_{kj}(n)$ definida por

$$\Delta w_{kj}(n) = -\eta \frac{\partial J(n)}{\partial w_{kj}(n)}, \tag{A.19}$$

em que η é o passo de aprendizado [18]. A partir de $\mathbf{W}^{(j)}(n)$, a matriz de pesos é atualizada por

$$\mathbf{W}^{(j)}(n+1) = \mathbf{W}^{(j)}(n) - \eta \frac{\partial J(n)}{\partial \mathbf{W}^{(j)}(n)}. \tag{A.20}$$

Analogamente, a partir de $\mathbf{b}^{(j)}(n)$, o vetor de *bias* deve ser atualizado por

$$\mathbf{b}^{(j)}(n+1) = \mathbf{b}^{(j)}(n) - \eta \frac{\partial J(n)}{\partial \mathbf{b}^{(j)}(n)}. \tag{A.21}$$

Para um problema de classificação, com o uso da função custo de Entropia Cruzada Categórica, o pseudocódigo do algoritmo de retropropagação é descrito no Algoritmo 4, com as

etapas de cálculo progressivo e regressivo descritas nos Algoritmos 2 e 3, respectivamente.

Algoritmo 2: Algoritmo da fase progressiva.

Entrada: $\mathbf{x}, \mathbf{W}, \mathbf{b}, \mathbf{y}, \mathbf{z}, \varphi$
Saída: $\mathbf{y}, \mathbf{z}$

Funcao `Propaga` $(\mathbf{x}, \mathbf{W}, \mathbf{b}, \mathbf{y}, \mathbf{z}, \varphi)$ **:**
  $\mathbf{y}^{(0)} \leftarrow \mathbf{x}$
  para $j = 0$; $j \leftarrow j + 1$; $j < L$ **faça**
    $\mathbf{z}^{(j+1)} \leftarrow \mathbf{W}^{(j+1)} \cdot \mathbf{y}^{(j)} + \mathbf{b}^{(j+1)}$
    $\mathbf{y}^{(j+1)} \leftarrow \varphi(\mathbf{z}^{(j+1)})$
  fim
  retorna $\mathbf{y}, \mathbf{z}$;
Fim da Funcao

Algoritmo 3: Algoritmo da fase regressiva.

Entrada: $\mathbf{x}, \mathbf{d}, \mathbf{W}, \mathbf{b}, \mathbf{dW}, \mathbf{db}, \mathbf{y}, \mathbf{z}, L, \varphi$
Saída: $\mathbf{dW}, \mathbf{db}$

Funcao `Retropropaga` $(\mathbf{x}, \mathbf{d}, \mathbf{W}, \mathbf{b}, \mathbf{dW}, \mathbf{db}, \mathbf{y}, \mathbf{z}, L, \varphi)$ **:**
  $\mathbf{dz}^{(L)} = \mathbf{y}^{(L)} - \mathbf{d}$
  $\mathbf{dW}^{(L)} \leftarrow \frac{1}{M}\mathbf{dz}^{(L)} \cdot (\mathbf{y}^{(L-1)})^T$
  $\mathbf{db}^{(L)} \leftarrow \frac{1}{M} \sum \mathbf{dz}^{(L)}$
  para $j = L - 2$; $j \leftarrow j - 1$; $j \geq 0$ **faça**
    $\mathbf{dz}^{(j+1)} \leftarrow (\mathbf{W}^{(j+2)})^T \cdot \mathbf{dz}^{(j+2)} * d\varphi(\mathbf{z}^{(j+1)})$
    $\mathbf{dW}^{(j+1)} \leftarrow \frac{1}{M}\mathbf{dz}^{(j+1)} \cdot (\mathbf{y}^{(j)})^T$
    $\mathbf{db}^{(j+1)} \leftarrow \frac{1}{M} \sum \mathbf{dz}^{(j+1)}$
  fim
  retorna $\mathbf{dW}, \mathbf{db}$;
Fim da Funcao

Algoritmo 4: Algoritmo de Retropropagação.

Entrada: $\mathbf{x}, \mathbf{y}, L, \varphi, \eta$
Saída: $\mathbf{W}, \mathbf{b}$, **custo**

Inicializar $\mathbf{W}, \mathbf{b}, \mathbf{z}, \mathbf{y}, \mathbf{dW}, \mathbf{db}$;
para $n = 0$; $n \leftarrow n + 1$; $n < M$ **faça**
  $\mathbf{y}, \mathbf{z} \leftarrow$ Propaga$(\mathbf{x}, \mathbf{W}, \mathbf{b}, \mathbf{y}, \mathbf{z}, \varphi)$;
  $\mathbf{dW}, \mathbf{db} \leftarrow$ Retropropaga$(\mathbf{x}, \mathbf{d}, \mathbf{W}, \mathbf{b}, \mathbf{dW}, \mathbf{db}, \mathbf{y}, \mathbf{z}, L, \varphi)$;
  para $j = 1$; $j \leftarrow j + 1$; $j \leq L$ **faça**
    $\mathbf{W}^{(j)} \leftarrow \mathbf{W}^{(j)} - \eta\mathbf{dW}$;
    $\mathbf{b}^{(j)} \leftarrow \mathbf{b}^{(j)} - \eta\mathbf{db}$;
  fim
  custo $\leftarrow J_{CCE}$;
fim

As parcelas $\frac{\partial J}{\partial \mathbf{W}^{(j)}}$ e $\frac{\partial J}{\partial \mathbf{b}^{(j)}}$ são obtidas pela regra da cadeia, e os cálculos de $\mathbf{dW}$ e $\mathbf{db}$ para a função custo considerada são mostrados no Algoritmo 3. A parcela $d\varphi(\mathbf{z}^{(j+1)})$ corresponde à derivada das funções de ativação de cada camada que surge pela regra da cadeia, sendo diferente para cada função φ escolhida.

Nos algoritmos descritos, quando omitidos os índices j para os vetores $\mathbf{y}, \mathbf{z}, \mathbf{dW}, \mathbf{db}$ eles

podem ser interpretados como matrizes que guardam em cada coluna j o vetor da camada j. Em *Python*, a implementação desses vetores ocorreu com o uso de dicionários, em que a chave usada foi o índice j de cada camada.

B Ajustes de Hiperparâmetros e Algoritmos de Otimização

Os hiperparâmetros de uma rede neural artificial são os parâmetros que são fornecidos às redes e são determinantes para a obtenção dos parâmetros $\mathbf{W}$ e $\mathbf{b}$. Alguns exemplos de hiperparâmetros são a taxa de aprendizado, o número de iterações, o número de camadas ocultas e de neurônios dessas camadas, além da escolha de funções de ativação [17]. O ajuste desses parâmetros faz parte do processo altamente iterativo e empírico das aplicações de aprendizado de máquina [17]. A Seção a seguir descreve brevemente alguns aspectos práticos do ajuste de hiperparâmetros.

Um primeiro aspecto a ser considerado durante o desenvolvimento de aplicações de aprendizado de máquina é a troca entre *bias* e variância (*variance*). O processo de aprendizado pode ser interpretado como um problema de "ajuste de curva" e é possível visualizar uma rede neural artificial como um bom interpolador não linear das entradas [18]. A capacidade de generalização de uma rede pode ser provada quando uma entrada de teste que nunca foi usada durante o treinamento obtém um mapeameto de entrada-saída correto.

No entanto, é comum que as redes guardem informações dos dados de treinamento que não são características da função verdadeira a ser modelada. Esse fenômeno é conhecido como sobreajuste (*overfitting*) e corresponde a um resultado com alta variância, em que a capacidade de generalização da rede é baixa. É possível também que a rede apresente um *underfitting*, comportando-se como um ajuste linear e demonstrando um alto *bias* [17], caso em que a capacidade de classificação é baixa. No segundo caso, é possível treinar por mais tempo ou aumentar a profundidade da rede para melhorar o desempenho. No primeiro, um ajuste possível é a regularização.

Uma das regularizações mais usadas é a ℓ^2. Essa regularização adiciona um termo na função custo, afetando a retropropagação e, consequentemente, a atualização dos pesos. A função custo torna-se

$$J = \frac{1}{M}\sum_{n=1}^{M}\sum_{k=1}^{N_L}\mathcal{L}(y_k^{(L)}(n), d_k(n)) + \frac{\lambda}{2M}\sum_{j=1}^{L}\left\|\mathbf{W}^{(j)}\right\|_F^2 \tag{B.1}$$

em que λ é o hiperparâmetro de regularização. A norma de Frobenius presente em (B.1) é

definida por

$$\left\|\mathbf{W}^{(j)}\right\|_F^2 = \sum_{i=1}^{(N_{j-1})} \sum_{k=1}^{(N_j)} (w_{ki}^{(j)})^2. \tag{B.2}$$

Essa regularização penaliza a matriz de pesos, evitando normas elevadas e diminuindo o efeito dos neurônios, fazendo com que o "ajuste" complexo e não linear da rede se aproxime de um linear [17].

Outra forma de regularização usada é o *dropout*. O *dropout* consiste na eliminação de determinados neurônios de forma aleatória em cada iteração do processo de aprendizado. Cada neurônio possui uma probabilidade p de ser desativado. Ao zerar os neurônios de forma aleatória, a rede não pode depender exclusivamente de características específicas, tendo que "espalhar" os seus pesos [17]. A Figura B.1 ilustra o uso do *dropout* para uma rede com $L = 2$, $N_1 = 6$, $N_2 = 4$, $p_1 = 0{,}5$ e $p_2 = 0{,}25$.

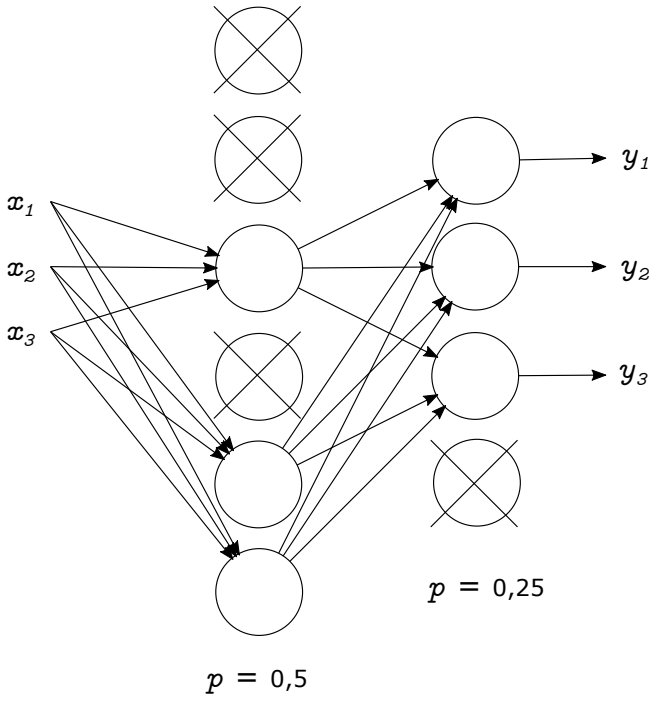


Figura B.1: Exemplo de *dropout*.

Além do ajuste de hiperparâmetros, existem diversos algoritmos de otimização usados para melhorar ou acelerar o processo de aprendizado. O Algoritmo de Adam é o mais conhecido e empregado atualmente e será descrito a seguir.

O Algoritmo de Adam é uma mistura de dois outros Algoritmos, o Gradiente Descendente com Momento e o RMSprop (*Root Mean Square Propagation*). Do Gradiente Descendente com Momento, o Adam adquire sua característica de usar uma média ponderada de modo exponencial dos valores dos parâmetros da iteração anterior. O hiperparâmetro β_1 estabelece a

contribuição da média ponderada de valores de pesos e *bias* anteriores para a nova iteração [17]. Em geral, usa-se $\beta_1 = 0{,}9$. Define-se a matriz $\mathbf{V}_{dw}^{(j)}$

$$\mathbf{V}_{dw}^{(j)}(t+1) = \beta_1 \mathbf{V}_{dw}^{(j)}(t) + (1-\beta_1)\frac{\partial J}{\partial \mathbf{W}^{(j)}}(t+1) \tag{B.3}$$

e o vetor $\mathbf{v}_{db}^{(j)}$

$$\mathbf{v}_{db}^{(j)}(t+1) = \beta_1 \mathbf{v}_{db}^{(j)}(t) + (1-\beta_1)\frac{\partial J}{\partial \mathbf{b}^{(j)}}(t+1), \tag{B.4}$$

que guardam informações das iterações anteriores. Do RMSprop, o Adam adquire sua característica de adaptação do passo de aprendizado, ditado pelo hiperparâmetro β_2. Um valor comum para β_2 é 0,999. Definem-se, também, uma matriz

$$\mathbf{S}_{dw}^{(j)}(t+1) = \beta_2 \mathbf{S}_{dw}^{(j)}(t) + (1-\beta_2)\left[\frac{\partial J}{\partial \mathbf{W}^{(j)}}(t+1)\right]^2 \tag{B.5}$$

e um vetor

$$\mathbf{s}_{db}^{(j)}(t+1) = \beta_2 \mathbf{s}_{db}^{(j)}(t) + (1-\beta_2)\left[\frac{\partial J}{\partial \mathbf{b}^{(j)}}(t+1)\right]^2. \tag{B.6}$$

Além disso, o Adam possui uma correção de *bias* que evita as diferenças dos vetores definidos nas Equações (B.3) a (B.5) no início das iterações. O pseudocódigo do Algoritmo de Adam está apresentado no Algoritmo 5.

Algoritmo 5: Algoritmo de Adam.

Inicializar $\mathbf{V}_{dw}^{(j)}, \mathbf{v}_{db}^{(j)}, \mathbf{S}_{dw}^{(j)}, \mathbf{s}_{db}^{(j)}$;
para $t = 0$; $t \leftarrow t+1$; $t < n_{iter}$ **faça**
  $\mathbf{V}_{dw}^{(j)} \leftarrow \beta_1 \mathbf{V}_{dw}^{(j)} + (1-\beta_1)\frac{\partial J}{\partial \mathbf{W}^{(j)}}$;
  $\mathbf{v}_{db}^{(j)} \leftarrow \beta_1 \mathbf{v}_{db}^{(j)} + (1-\beta_1)\frac{\partial J}{\partial \mathbf{b}^{(j)}}$;
  $\mathbf{S}_{dw}^{(j)} \leftarrow \beta_2 \mathbf{S}_{dw}^{(j)} + (1-\beta_2)\left[\frac{\partial J}{\partial \mathbf{W}^{(j)}}\right]^2$;
  $\mathbf{s}_{db}^{(j)} \leftarrow \beta_2 \mathbf{s}_{db}^{(j)} + (1-\beta_2)\left[\frac{\partial J}{\partial \mathbf{b}^{(j)}}\right]^2$;
  $\mathbf{V}_{dw}^{c} \leftarrow \frac{\mathbf{V}_{dw}^{(j)}}{1-\beta_1^t}$;
  $\mathbf{v}_{db}^{c} \leftarrow \frac{\mathbf{v}_{db}^{(j)}}{1-\beta_1^t}$;
  $\mathbf{S}_{dw}^{c} \leftarrow \frac{\mathbf{S}_{dw}^{(j)}}{1-\beta_2^t}$;
  $\mathbf{s}_{db}^{c} \leftarrow \frac{\mathbf{s}_{db}^{(j)}}{1-\beta_2^t}$;
  $\mathbf{W}^{(j)} \leftarrow \mathbf{W}^{(j)} - \eta\frac{\mathbf{V}_{dw}^{c}}{\sqrt{\mathbf{S}_{dw}^{c}}+\epsilon}$;
  $\mathbf{b}^{(j)} \leftarrow \mathbf{b}^{(j)} - \eta\frac{\mathbf{v}_{db}^{c}}{\sqrt{\mathbf{s}_{db}^{c}}+\epsilon}$;
fim

No Algoritmo, os vetores e matrizes com índice c representam os vetores e matrizes corrigidos no processo de correção do *bias* por médias ponderadas exponencialmente [17]. A constante ϵ foi usada para evitar a divisão por zero.

Por último, o Aprendizado Sensível ao Custo (*Cost-Sensitive Learning*) e uso do SMOTE (*Synthetic Minority Over-Sampling Technique*) em redes neurais, para trabalhar com classes

desbalanceadas, são descritos a seguir. Nas configurações usuais do Aprendizado de Máquina, os classificadores são projetados para minimizar o número de erros cometidos, isto é, de classificações incorretas realizadas. Porém, quando os custos dos erros de classificação diferem entre as classes, os classificadores devem ser avaliados comparando-se o total dos custos dos erros [115]. É nesse contexto de modificação dos pesos de cada erro que se encontra o Aprendizado Sensível ao Custo.

Para classificações binárias, a função custo de entropia cruzada pode ter um peso aplicado ao caso probabilístico de falsos negativos, penalizando mais os falsos negativos caso o peso seja maior do que um, ou penalizando menos caso contrário [116]. Já para classificações multiclasses, a função de entropia cruzada categórica pode possuir pesos por classes, aumentando ou diminuindo os custos de um provável falso negativo para cada uma das classes. Em [116], a função de entropia cruzada categórica com pesos é dada por

$$J_{WCCE} = -\frac{1}{M}\sum_{n=1}^{M}\sum_{k=1}^{N_L} p_k \times y_k^{(L)}(n) log(d_k(n)), \tag{B.7}$$

em que M é o número de exemplos de treinamento, N_L é o número de classes, p_k é o peso da classe k, $d^k(n)$ é o valor real do exemplo n de treinamento para a classe k e $x(n)$ é a entrada para o n-ésimo exemplo de treinamento. Os pesos são calculados na proporção inversa do número de dados por classe.

Além de distinguir os custos de cada erro de classificação, é possível melhorar o desempenho dos algoritmos no contexto dos dados desbalanceados através da sobreamostragem (*oversampling*) das classes minoritárias ou da subamostragem (*undersampling*) das classes majoritárias. A abordagem proposta por [117], denominada SMOTE, realiza uma sobreamostragem criando exemplos sintéticos das classes minoritárias, ao invés de apenas repetir aleatoriamente os dados já existentes dessas classes.

C Redes Neurais Convolucionais

As redes neurais convolucionais (*Convolutional Neural Network* – CNN) são uma classe especial de MLP usada para reconhecer formas bidimensionais, como séries temporais ou imagens, com um alto grau de invariância com relação à translação, ao dimensionamento e a distorções no geral [19]. Elas são muito empregadas para problemas de classificação de padrões e têm sido usadas com sucesso em diversas aplicações práticas. Apesar de ser originalmente proposta para o processamento de imagens, a CNN tem sido utilizada recentemente no processamento de séries temporais, como é o caso do sinal de ECG [1, 118–120].

Da mesma forma que a MLP, a CNN calcula os pesos a partir de um algoritmo de retropropagação. A CNN é composta por camadas convolucionais para a extração de características, que servem como entrada para uma rede MLP totalmente conectada, responsável pela classificação. Nessas camadas, cada neurônio possui entradas provenientes de um *campo local* da camada anterior, forçando a extração de características locais, e a localização exata dessas características perde importância [18].

Cada camada convolucional da rede é composta por mapas de características, dentro dos quais os neurônios são forçados a compartilhar o mesmo conjunto de pesos sinápticos. O mapeamento dessas características permite efeitos como a invariância ao deslocamento, por meio da convolução com um filtro (*kernel*) de tamanho reduzido, seguido de uma função sigmoidal, e a redução no número de parâmetros livres, com o compartilhamento de pesos.

As camadas convolucionais também envolvem estágios com operações de subamostragem, conhecidas como *pooling*. Cada camada convolucional é seguida por uma camada computacional que realiza uma média local e uma subamostragem, por meio da qual se reduz a resolução do mapa de características. Isso diminui a sensibilidade da saída do mapa a deslocamentos e outras formas de distorção. Após a extração, os mapas de características são vetorizados e a MLP realiza a classificação utilizando as características aprendidas durante a etapa anterior [121]. A Figura C.1 ilustra um exemplo de CNN para séries temporais.

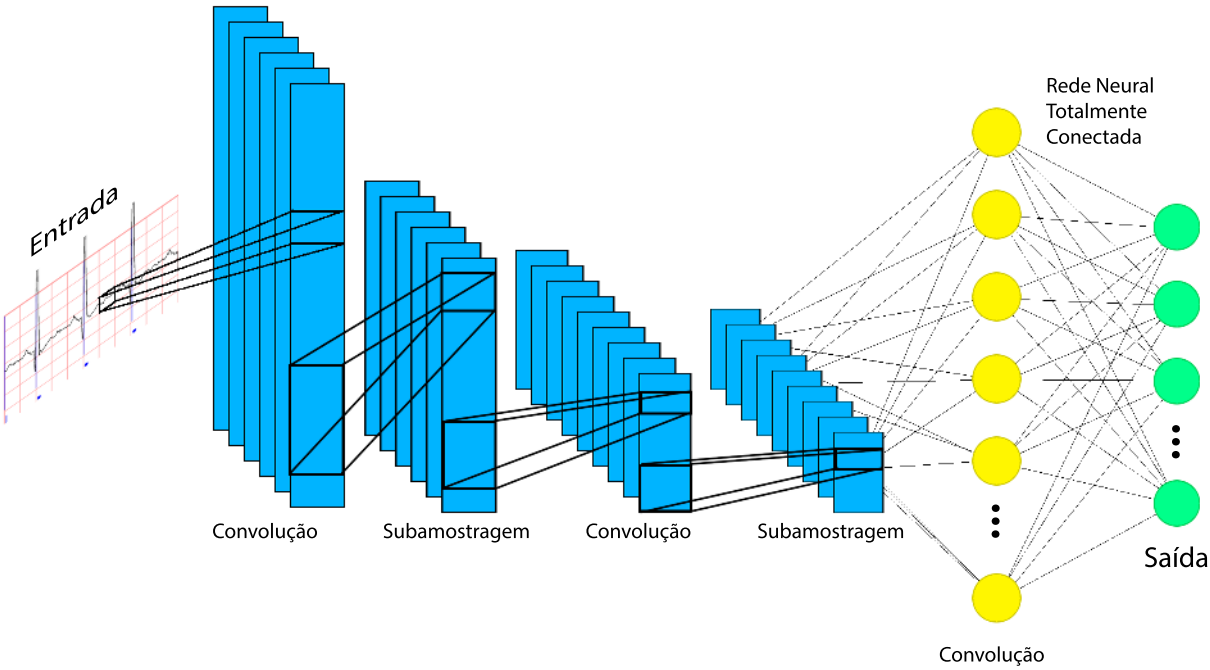


Figura C.1: Rede Neural Convolucional

Para o exemplo de uma matriz bidimensional de entrada $\mathbf{I}$ e para um *kernel* multidimensional $\mathbf{K}$ de parâmetros que são atualizados pelo algoritmo de aprendizado, a operação de convolução é definida por [19]

$$\mathbf{S}(i,j) = (\mathbf{I} * \mathbf{K})(i,j) = \sum_m \sum_n \mathbf{I}(m,n)\mathbf{K}(i-m, j-n). \tag{C.1}$$

A saída $\mathbf{S}(i,j)$ dessa operação é comumente denominada de mapa de características. A convolução apresenta uma propriedade cumulativa, que decorre da inversão do *kernel* em relação à entrada. No entanto, essa não é uma propriedade relevante para as redes neurais e, por isso, muitas redes são implementadas com a função de correlação cruzada, dada por

$$\mathbf{S}(i,j) = (\mathbf{I} * \mathbf{K})(i,j) = \sum_m \sum_n \mathbf{I}(i+m, j+n)\mathbf{K}(m,j). \tag{C.2}$$

No contexto de aprendizado de máquina, essa operação em geral é chamada de convolução ou convolução sem inversão de filtro [19], como ilustrado na Figura C.2.

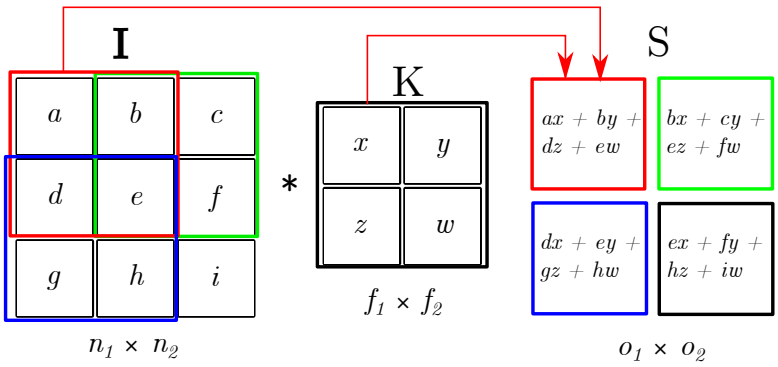


Figura C.2: Operação de convolução sem inversão de filtro.

Como exemplificado na Figura, a convolução de uma entrada **I** de dimensão $(n_1 \times n_2) = (3 \times 3)$ com o filtro **K** de dimensão $(f_1 \times f_2) = (2 \times 2)$ gera uma saída **S** de dimensão $(o_1 \times o_2) = (2 \times 2)$. De forma genérica, a dimensão de **S** é dada por [17]

$$(o_1, o_2) = (n_1 - f_1 + 1, n_2 - f_2 + 1). \tag{C.3}$$

Muitas vezes, para não se perder a informação das bordas ou para se obter **S** com a mesma dimensão da entrada, é comum realizar a operação de *padding*, preenchendo-se com zeros as bordas no entorno da entrada e realizando a convolução com essas bordas adicionais, operação denominada de *same convolution*. Seja p a quantidade de colunas ou linhas adicionais, a dimensão da saída com o uso de *padding* é dada por

$$(o_1, o_2) = (n_1 + 2p - f_1 + 1, n_2 + 2p - f_2 + 1), \tag{C.4}$$

e, para que a dimensão da saída seja a mesma da entrada, tem-se que

$$p = \frac{f - 1}{2}. \tag{C.5}$$

Além do *padding*, também é possível aplicar o *stride*, em que o deslocamento s do filtro sobre a entrada passa a ser maior do que apenas uma linha ou uma coluna. Assim, a dimensão de **S** é dada por [17]

$$(o_1, o_2) = \left(\left\lfloor \frac{n_1 + 2p - f_1}{s} + 1 \right\rfloor, \left\lfloor \frac{n_2 + 2p - f_2}{s} + 1 \right\rfloor \right). \tag{C.6}$$

Os estágios de *pooling* ajudam a tornar a rede aproximadamente invariante a pequenas translações na entrada e reduzem as dimensões das matrizes. Uma das funções mais utilizadas para isso é a chamada *max-pooling*, que retorna a saída de maior valor dentro de uma vizinhança [19]. Um exemplo numérico com essa função é ilustrado na Figura C.3.

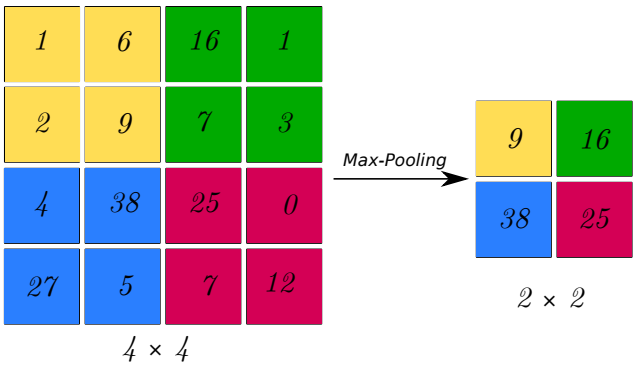


Figura C.3: Função de *max pooling* aplicada a um exemplo numérico.

Assim como nas camadas convolucionais, pode-se variar o tamanho do filtro usado nessa operação de *pooling*, o número de filtros e o *stride*. Porém, ao contrário das camadas convolucionais, não existem parâmetros a serem aprendidos nessa etapa de subamostragem, sendo apenas uma operação computacional fixa.

D Redes Neurais Recorrentes

Assim como as CNNs são uma classe especial de MLP usada para reconhecer formas bidimensionais, as Redes Neurais Recorrentes (*Recurrent Neural Network* – RNN) são uma família de redes neurais para processamento de dados sequenciais que apresentam laços em suas conexões, tendo a saída de pelo menos um neurônio realimentada à rede. A realimentação das RNNs possibilita uma natureza mais dinâmica, o que facilita a caracterização de sinais variantes no tempo e a classificação de sinais que apresentam dependência temporal de longo prazo [19, 122].

O compartilhamento de parâmetros na RNN é diferente e mais profundo do que o da CNN. Na RNN, cada parte da saída é uma função de saídas anteriores, e cada nova saída é produzida usando as mesmas regras aplicadas às saídas anteriores, enquanto na CNN a saída é apenas uma função de uma pequena vizinhança da entrada [19].

Um modelo de RNN é o de estado de espaços (*State-Space Model*) mostrado na Figura D.1, em que a camada oculta define o estado da rede e sua saída é realimentada à camada de entrada por blocos atrasadores. A quantidade de atrasos aplicada determina a ordem do modelo. Considerando um vetor de entrada $\mathbf{u}_n$ e um vetor de saída $\mathbf{x}_n$, o comportamento dinâmico do modelo pode ser descrito por

$$\begin{aligned} \mathbf{x}_{n+1} &= \varphi(\mathbf{x}_n, \mathbf{u}_n) \\ \mathbf{y}_n &= \mathbf{B}\mathbf{x}_n, \end{aligned} \tag{D.1}$$

em que $\varphi(\cdot, \cdot)$ representa a função não-linear da camada oculta e $\mathbf{B}$ é a matriz de pesos da camada de saída. Esse modelo inclui outras estruturas, como, por exemplo, a rede de Elman, cuja arquitetura é idêntica com exceção da camada de saída poder ser não-linear e do banco de atrasos da saída ser omitido [18]. A rede de Elman é por vezes também chamada de rede neural recorrente simples (*Simple Recurrent Network* – SRN), e possui neurônios da camada oculta que "reciclam" a informação de tempos passados.

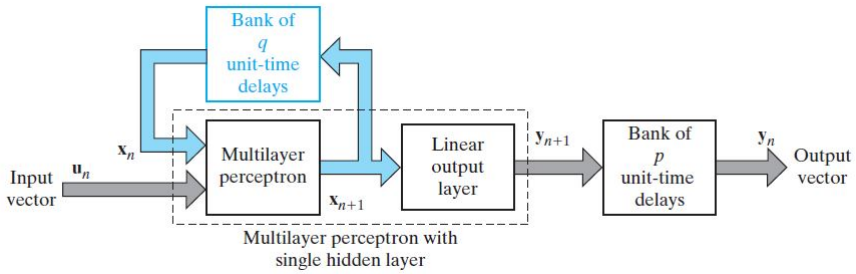


Figura D.1: Modelo *State-Space*.
Fonte: [18]

Uma rede recorrente com comportamento dinâmico pode ser descrita pelas equações de estado e de medida dadas, respectivamente, por

$$\mathbf{x}_{n+1} = \varphi(\mathbf{W}_a\mathbf{x}_n + \mathbf{W}_b\mathbf{u}_n) \tag{D.2}$$

$$\mathbf{y}_n = \mathbf{W}_c\mathbf{x}_n, \tag{D.3}$$

com $\mathbf{W}_a$ de dimensão $(q \times q)$, $\mathbf{W}_b$ de dimensão $(q \times m)$ e $\mathbf{W}_c$ de dimensão $(p \times q)$, sendo q denominado de ordem do sistema. Dessa forma, a Figura D.1 é um modelo de m entradas, p saídas e de ordem q. A matriz $\mathbf{W}_a$ representa os pesos dos q neurônios da camada oculta que estão conectados por realimentação à entrada. $\mathbf{W}_b$ é a matriz de pesos dos neurônios da camada oculta que estão conectados aos nós de alimentação da entrada. Já $\mathbf{W}_c$ é a matriz de pesos dos p neurônios na camada de saída que estão conectados aos neurônios da camada oculta.

A RNN possui a importante propriedade de ser uma aproximadora universal de sistemas dinâmicos não-lineares, satisfazendo o Teorema de Aproximação Universal de Kolmogorov [123], com grande poder de computação em processamento de sinais e aplicações de controle [18]. Ela também apresenta a habilidade de simular máquinas de estado finito (*Finite State Machine* – FSM), tendo sido historicamente usada para prever a próxima letra ou palavra de uma frase, ou seja, mantendo uma dependência do contexto.

A recursão traz propriedades interessantes como o armazenamento de memórias em estruturas cíclicas e a produção de diferentes saídas para uma mesma configuração de entrada, uma vez que a saída passa a depender também do valor atual dos estados internos. Ela também tem a propriedade de gerar saídas que evoluem com o tempo mesmo quando a entrada é estática [122]. Além disso, RNNs com conexões padrões, isto é, sem a necessidade de serem de maior ordem, são suficientes para simular qualquer Máquina de Turing (*Turing machine*) [124], o que demonstra o poder computacional dessas redes.

Nas RNNs há dois algoritmos de aprendizado. O algoritmo de BPTT (*back-propagation-through-time*) opera com a premissa de que a operação temporal de uma rede recorrente seja "desdobrada" (*unfolded*) em uma MLP, tornando-se o algoritmo de retropropagação comum. Já o algoritmo de RTRL (*real-time-recurrent-learning*) usa o modelo descrito pelas Equações (D.2), em que os ajustes dos pesos são feitos em tempo real, enquanto a rede continua a desempenhar a função de processamento do sinal. Ambos os métodos são baseados no gradiente descendente. O BPTT possui menor complexidade computacional do que o RTRL, mas demanda um espaço de memória grande, que é proporcional ao tamanho das sequências. O RTRL é mais adequado a treinamentos contínuos *on-line*, enquanto o BPTT pode ser usado para os dois modos de treinamento e não depende somente dos passos de tempo mais recente, mas sim de múltiplos passos dentro de um determinado tempo [18].

Um grafo computacional é um modo de formalizar a estrutura de um conjunto de cálculos que envolvam o mapeamento de entradas e de parâmetros para saídas e custo. Nas RNNs, os grafos possuem uma estrutura repetitiva, que resulta do "desdobramento" (*unfolding*) de um cálculo. Para um número de passos de tempo (*time steps*) finito τ, um grafo pode ser desdobrado aplicando-se uma mesma definição $\tau - 1$ vezes. Por exemplo, para um sistema dinâmico

$$\mathbf{s}^{(t)} = f(\mathbf{s}^{(t-1)}; \boldsymbol{\theta}), \tag{D.4}$$

em que $\mathbf{s}^{(t)}$ é o estado do sistema e $\boldsymbol{\theta}$ é o parâmetro que melhor minimiza a função custo, e para $\tau = 3$, tem-se

$$\begin{aligned} \mathbf{s}^{(3)} &= f(\mathbf{s}^{(2)}; \boldsymbol{\theta}) \\ &= f(f(\mathbf{s}^{(1)}; \boldsymbol{\theta}); \boldsymbol{\theta}), \end{aligned} \tag{D.5}$$

que pode ser representado em um grafo acíclico [19].

Se existir um sinal externo $\mathbf{x}^{(t)}$, o grafo de uma rede neural

$$\mathbf{h}^{(t)} = f(\mathbf{h}^{(t-1)}; \mathbf{x}^{(t)}; \boldsymbol{\theta}), \tag{D.6}$$

pode ser representado como indicado na Figura D.2. A variável $\mathbf{h}$ é o estado da camada oculta da rede.

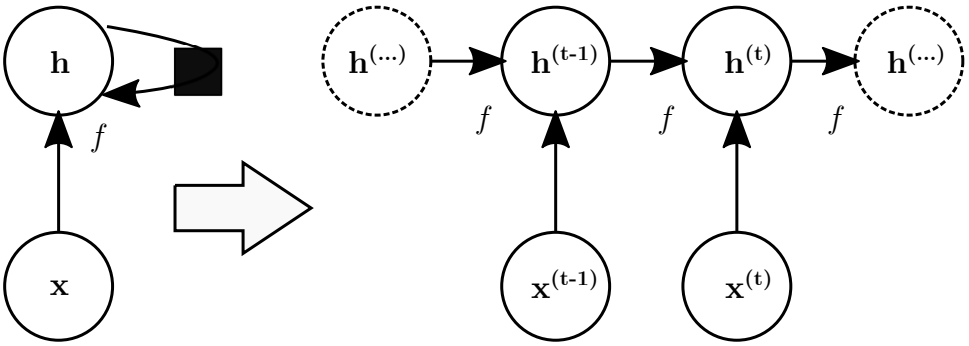


Figura D.2: Representação da operação de *unfolding*.

A Equação (D.6) pode ser representada de duas formas. A primeira é a apresentada do lado esquerdo da Figura D.2, contendo um nó para cada componente que existe em uma implementação física de um modelo, em que o quadrado preto indica que uma interação ocorre com um atraso de um passo de tempo, do estado do tempo t ao de $t + 1$. A segunda é o grafo do lado direito, em que cada componente é representado por muitas variáveis diferentes, com cada uma por passo de tempo, e o tamanho do grafo depende do tamanho da sequência [19].

O processo de *unfolding* introduz duas vantagens principais:

1. Independentemente do tamanho da sequência, o modelo aprendido tem sempre o mesmo tamanho de entrada, porque ele é especificado em termos da transição de um estado para outro, ao invés de ser especificado em termos do histórico de tamanhos variáveis; e

2. É possível usar a mesma função f de transição com os mesmo parâmetros a cada passo de tempo.

Assim, não é necessário criar um modelo separado para todos os possíveis passos de tempo, sendo possível aprender um único modelo f que opera em todos os passos de tempo e com todos os tamanhos de sequência, permitindo a generalização para tamanhos que nunca apareceram no conjunto de treinamento [19].

Há diferentes estruturas de RNNs com respeito à quantidade de entradas e saídas da rede. Existem RNNs que produzem uma saída a cada passo de tempo (*many-to-many*) com conexões recorrentes entre unidades ocultas, ou com conexões recorrentes da saída de um passo para as unidades ocultas do próximo; outras produzem apenas uma saída final após ler uma sequência inteira (*many-to-one*). É possível também gerar uma sequência a partir de uma única entrada (*one-to-many*). Um exemplo de aplicação para uma rede *many-to-many* é a de encontrar um nome de uma pessoa em uma frase, fornecendo 0 na saída quando a palavra em determinado instante não é um nome e 1 caso contrário. Uma rede (*many-to-one*), por outro lado, pode ser

usada para classificar sentimentos a partir de uma frase, como fornecer uma nota de um a cinco a partir de um comentário sobre um filme [17].

Um modelo de RNN *many-to-many* de tamanho finito com conexões recorrentes entre unidades ocultas, como mostrada na Figura D.3, é universal no sentido de que pode calcular qualquer função executável por uma máquina de Turing.

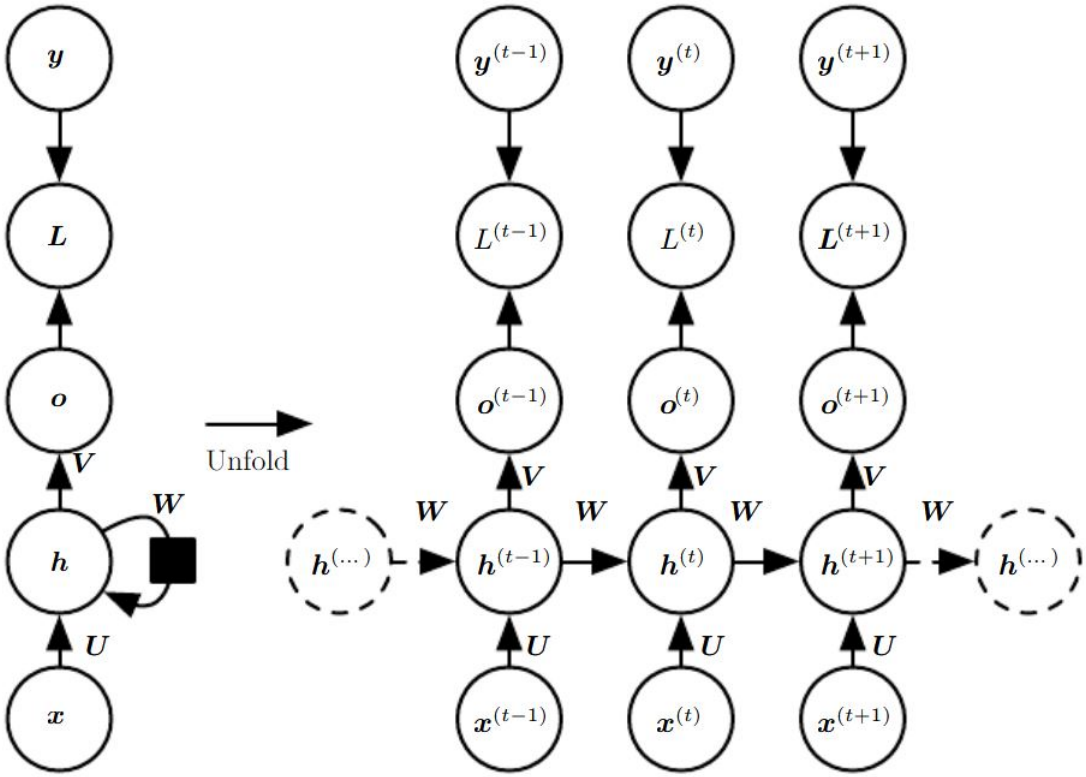


Figura D.3: Grafo de uma RNN.
Fonte: [19]

A propagação da RNN começa com um estado inicial $\mathbf{h}^{(0)}$, e, com $t = 1, \ldots, \tau$, aplicam-se as equações

$$\mathbf{a}^{(t)} = \mathbf{b} + \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} \tag{D.7}$$

$$\mathbf{h}^{(t)} = \varphi(\mathbf{a}^{(t)}) \tag{D.8}$$

$$\mathbf{o}^{(t)} = \mathbf{c} + \mathbf{V}\mathbf{h}^{(t)} \tag{D.9}$$

$$\hat{\mathbf{y}}^{(t)} = \text{softmax}(\mathbf{o}^{(t)}), \tag{D.10}$$

em que $\mathbf{U}$ é a matriz de pesos da entrada para a camada oculta, $\mathbf{W}$ é a matriz de pesos entre camadas ocultas, $\mathbf{V}$ é a matriz de pesos da camada oculta para saída, e $\mathbf{b}$ e $\mathbf{c}$ são vetores de *bias*. A função custo L mede quão longe está a saída $\mathbf{o}$ de seu valor verdadeiro $\mathbf{y}$.

A função custo total é então dada pela soma da função custo para cada passo de tempo,

$$L\left(\left\{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(\tau)}\right\},\left\{\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(\tau)}\right\}\right)=\sum_{t}L^{(t)}. \tag{D.11}$$

O cálculo do gradiente envolve propagar a entrada da esquerda para a direita no grafo *unfolded*, seguido de uma retropropagação do erro da direita para a esquerda, pelo BPTT.

No BPTT, a recursão inicia-se com os nós imediatamente anteriores ao custo final. É necessário calcular, primeiramente, os gradientes dos nós internos do grafo, para obter enfim os gradientes dos parâmetros. O gradiente $\nabla_{\boldsymbol{o}^{(t)}}L$ nas saídas $\boldsymbol{o}^{(t)}$ do passo de tempo t, para cada neurônio i da camada oculta é

$$\left(\nabla_{\boldsymbol{o}^{(t)}}L\right)_i=\frac{\partial L}{\partial o_i^{(t)}}=\frac{\partial L}{\partial L^{(t)}}\frac{\partial L^{(t)}}{\partial o_i^{(t)}}=\hat{y}_i^{(t)}-\mathbf{1}_{i=y^{(t)}}. \tag{D.12}$$

Em $t=\tau$, o gradiente de $\boldsymbol{h}^{(\tau)}$ é

$$\nabla_{\boldsymbol{h}^{(\tau)}}L=\boldsymbol{V}^{\top}\nabla_{\boldsymbol{o}^{(\tau)}}L, \tag{D.13}$$

e, fazendo a retropropagação, de $t=\tau-1$ a $t=1$, deve-se levar em consideração também os valores de $\boldsymbol{h}^{(t+1)}$,

$$\nabla_{\boldsymbol{h}^{(t)}L}=\left(\frac{\partial\boldsymbol{h}^{(t+1)}}{\partial\boldsymbol{h}^{(t)}}\right)^{\top}\left(\nabla_{\boldsymbol{h}^{(t+1)}}L\right)+\left(\frac{\partial\boldsymbol{o}^{(t)}}{\partial\boldsymbol{h}^{(t)}}\right)^{\top}\left(\nabla_{\boldsymbol{o}^{(t)}}L\right), \tag{D.14}$$

em que $\left(\frac{\partial\boldsymbol{o}^{(t)}}{\partial\boldsymbol{h}^{(t)}}\right)^{\top}$ é $\boldsymbol{V}^{\top}$, e $\left(\frac{\partial\boldsymbol{h}^{(t+1)}}{\partial\boldsymbol{h}^{(t)}}\right)^{\top}$ depende da função de ativação usada.

Os gradientes dos parâmetros são encontrados por [19]

$$\nabla_{\boldsymbol{c}}L=\sum_{t}\left(\frac{\partial\boldsymbol{o}^{(t)}}{\partial\boldsymbol{c}}\right)^{\top}\nabla_{\boldsymbol{o}^{(t)}}L=\sum_{t}\nabla_{\boldsymbol{o}^{(t)}}L \tag{D.15}$$

$$\nabla_{\boldsymbol{b}}L=\sum_{t}\left(\frac{\partial\boldsymbol{h}^{(t)}}{\partial\boldsymbol{b}^{(t)}}\right)^{\top}\nabla_{\boldsymbol{h}^{(t)}}L \tag{D.16}$$

$$\nabla_{\boldsymbol{V}}L=\sum_{t}\sum_{i}\left(\frac{\partial L}{\partial o_i^{(t)}}\right)\nabla_{\boldsymbol{V}^{(t)}}o_i^{(t)}=\sum_{t}\left(\nabla_{\boldsymbol{o}^{(t)}}L\right)\boldsymbol{h}^{(t)^{\top}} \tag{D.17}$$

$$\nabla_{\boldsymbol{W}}L=\sum_{t}\sum_{i}\left(\frac{\partial L}{\partial h_i^{(t)}}\right)\nabla_{\boldsymbol{W}^{(t)}}h_i^{(t)} \tag{D.18}$$

$$\nabla_{\boldsymbol{U}}L=\sum_{t}\sum_{i}\left(\frac{\partial L}{\partial h_i^{(t)}}\right)\nabla_{\boldsymbol{U}^{(t)}}h_i^{(t)}. \tag{D.19}$$

A rede da Figura D.4 é menos poderosa e expressa uma menor quantidade de funções do que a da Figura D.3, não podendo simular uma máquina de Turing, pois a única informação que é mandada para o futuro é a da saída, e, a menos que esta tenha uma alta dimensão, há

significativa perda de informação do passado. A vantagem dessa rede, porém, é a de que os passos de tempos são desacoplados, pois para um determinado tempo, o conjunto de treinamento já pode prover o valor ideal da saída anterior, não sendo necessário computá-lo.

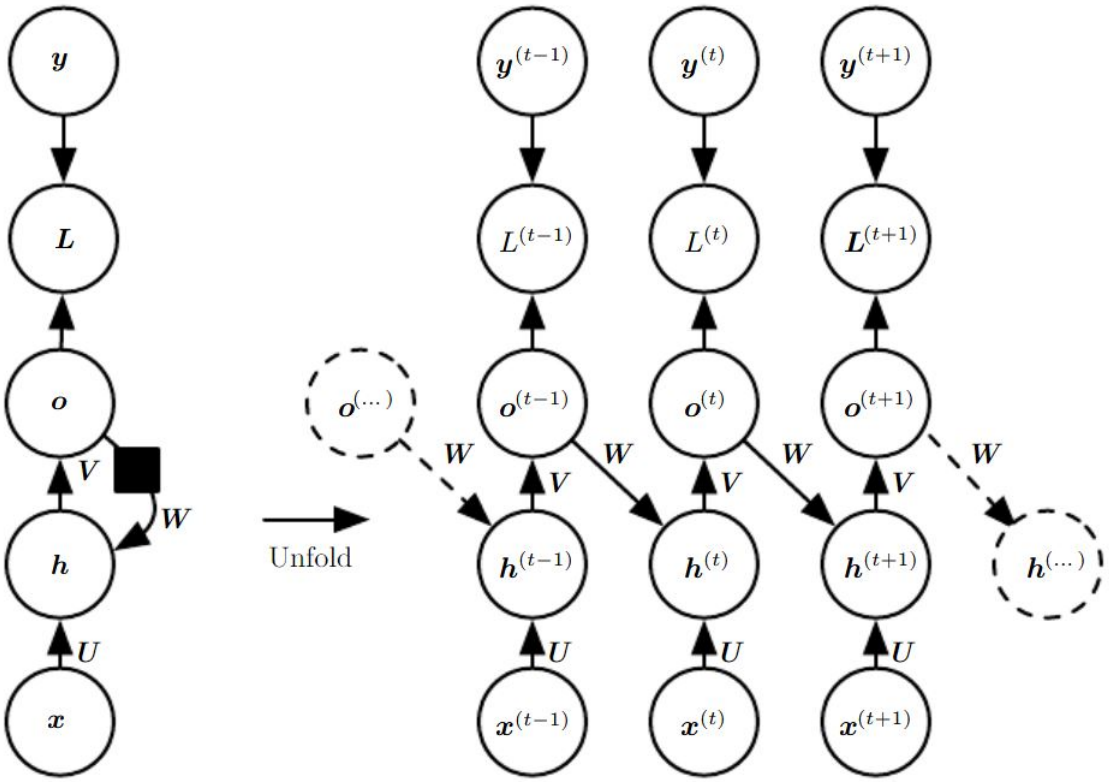


Figura D.4: Grafo de uma RNN com conexões entre a saída de um passo de tempo e a camada oculta do próximo.

Fonte: [19]

Nesse modelo, pode-se usar o *teacher forcing*, baseado no critério de máxima verossimilhança, em que durante o treinamento o modelo usa o valor verdadeiro $y^{(t)}$ como entrada do tempo $t+1$, evitando usar o BPTT, que é custoso [18].

Atualmente, as estruturas de RNN mais utilizadas são compostas por blocos com mecanismos internos, as portas (*gates*), e essas redes são chamadas de *gated* RNNs. Exemplos desses blocos são o LSTM (*Long Short-Term Memory*) [40] e o GRU (*Gated Recurrent Units*) [125]. As *gated* RNNs são baseadas na ideia de criar caminhos ao longo dos passos para evitar os problemas de desvanecimento do gradiente (*vanishing gradient*) ou de explosão do gradiente (*exploding gradient*).

As RNNs básicas apresentam dificuldades para capturar dependências de longo prazo, e, assim como ocorre com MLPs muito profundas ao tentar modificar as computações realizadas em camadas iniciais, sofre o problema de (*vanishing gradient*). Isso afeta aplicações como a linguagem, que possui termos no final de um frase dependentes dos primeiros [17]. Por exemplo,

na frase “Os estudantes de engenharia, que apresentaram o projeto da Feira de Ciências no último sábado, esperam pelos resultados”, há uma oração subordinada adjetiva explicativa longa entre o sujeito “estudantes” e a forma verbal “esperam”. Assim, a rede deve memorizar por vários passos de tempo a condição de conjugação no plural do verbo “esperar”. Com os caminhos criados nas *gated* RNNs, é possível memorizar informações por longos períodos, e, no caso do bloco LSTM, é também possível esquecê-las, após serem usadas.

D.1 O Bloco *Long Short-Term Memory*

O LSTM foi originalmente proposto em [40] para resolver o problema de desvanecimento do gradiente. Sua capacidade de guardar informações em intervalos arbitrários possibilita o aprendizado de dependências de longo prazo e a classificação de séries temporais com durações desconhecidas. O uso desse bloco tem se mostrado muito bem-sucedido em diversas aplicações, como reconhecimento de voz [126], tradução automática [127], reconhecimento de palavras cursivas [128], entre outros.

Uma LSTM possui células com as mesmas entradas e saídas de uma RNN normal, mas com mais parâmetros e com portas que controlam o fluxo de informação. Essas células são conectadas de forma recorrente umas às outras, substituindo as unidades ocultas de RNNs básicas. Uma célula de LSTM é ilustrada na Figura D.5. Um dos componentes da LSTM é a unidade de estado $s_i^{(t)}$ que tem um *self-loop*, controlado por uma porta $f_i^{(t)}$, para cada passo de tempo t e unidade i, denominada de *forget gate*. O número de unidades i é o número que se deseja de neurônios na camada oculta, no estado da célula e na saída da célula.

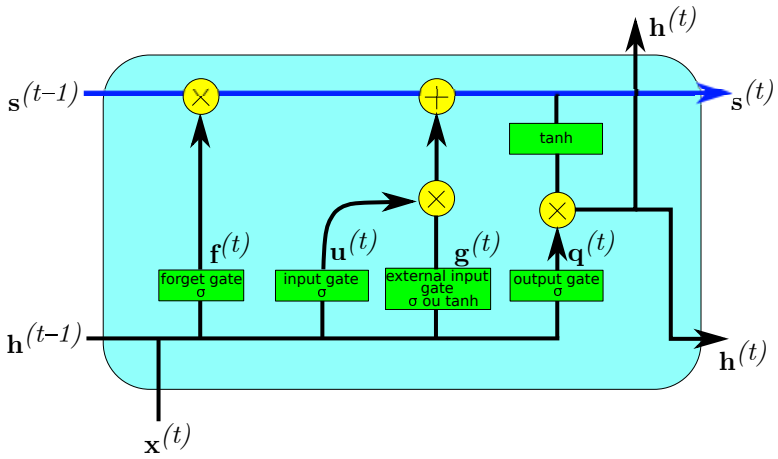


Figura D.5: Bloco LSTM.

A *forget gate* decide se a informação do estado da célula anterior deve passar adiante para

diferentes passos de tempo, sendo dada por

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right), \tag{D.20}$$

em que $\mathbf{x}^{(t)}$ é a entrada atual, $\mathbf{h}^{(t)}$ é o vetor da camada oculta atual, contendo as saídas de todas as células LSTM, e $\mathbf{b}^f, \mathbf{U}^f, \mathbf{W}^f$ são, respectivamente, o *bias*, os pesos das entradas e os pesos das conexões recorrentes para a *forget gate*. Essa unidade usa uma função de ativação sigmoidal para dar como saída 1, caso queira manter a informação do estado $s_i^{(t-1)}$, e 0, caso contrário.

Uma entrada $x_i^{(t)}$ passa por um neurônio artificial normal, e seu valor $u_i^{(t)}$ pode ser "acumulado" no estado se a porta de entrada externa (*external input gate*) $g_i^{(t)}$ permitir. A saída $u_i^{(t)}$ da *input gate* é dada por

$$u_i^{(t)} = \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right), \tag{D.21}$$

em que $\mathbf{b}, \mathbf{U}, \mathbf{W}$ são, respectivamente, o *bias*, os pesos das entradas e os pesos das conexões recorrentes da própria célula LSTM.

A *external input gate* é calculada de forma semelhante a *forget gate*, sendo dada por

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right), \tag{D.22}$$

em que a função de ativação pode ser tanto a sigmoidal quanto a tangente hiperbólica.

O estado interno da célula LSTM (*cell internal state*) é representado pela linha horizontal no topo diagrama da Figura D.5, e ele é atualizado por

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + u_i^{(t)} g_i^{(t)}. \tag{D.23}$$

O estado da célula passa por uma tangente hiperbólica, e a saída da célula é dada pela multiplicação

$$h_i^{(t)} = \tanh\left(s_i^{(t)}\right) q_i^{(t)}, \tag{D.24}$$

em que $q_i^{(t)}$ é a porta de saída (*output gate*), que pode controlar a ocorrência ou não de saída na célula, e é descrita por

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right). \tag{D.25}$$

E Análise de Discriminantes Lineares

Nesta Seção, os conceitos do uso da análise de discriminantes lineares para classificação são detalhados. Inicialmente, abordam-se os discriminantes lineares de Fisher e a sua generalização. Em seguida, uma visão de como modelos com fronteiras de decisão lineares surgem a partir do pressuposto de distribuição dos dados é apresentada. O uso da LDA para classificação é então explicado para a regra de máxima verossimilhança. Por fim, apresenta-se uma modificação da LDA para trabalhar com classes em que o número de dados é desbalanceado.

E.1 Discriminantes Lineares de Fisher

A análise de discriminantes lineares (*linear discriminant analysis* – LDA) é uma técnica bem conhecida para extração de características, redução de dimensionalidade e classificação, sendo uma generalização do discriminante linear de Fisher [38]. Apesar de discriminantes lineares serem comumente chamados de discriminantes lineares de Fisher, o desenvolvimento realizado por Fisher [38] é um pouco diferente dos métodos que modelam funções discriminantes, uma vez que Fisher não assume distribuições gaussianas ou covariâncias iguais entre as classes. A seguir, o discriminante de Fisher e a sua generalização para a LDA são descritos.

A LDA foi largamente explorada em aplicações de reconhecimento facial e muitas vezes é comparada a análise de componentes principais (*principal component analysis* – PCA) [129–131], um dos métodos multivariados mais conhecidos de redução de dimensionalidade, proposto por Pearson [132] e Hotelling [133]. O objetivo da PCA é reduzir ao máximo a dimensão dos dados e produzir a menor perda possível das informações. Nesse método, as variáveis correlacionadas e com redundância são transformadas em variáveis não correlacionadas e sem redundância [134].

No entanto, a PCA pode auxiliar ou não a classificação, uma vez que essa análise não leva em consideração os valores verdadeiros das classes, realizando a redução por coordenadas. Assim, apesar da PCA encontrar componentes úteis para representar os dados, não é possível afirmar que essas componentes sejam úteis para fazer a discriminação dos dados em diferentes classes.

Enquanto a PCA procura por direções que são eficientes para representações, a LDA procura por direções que são eficientes para classificações [36].

Considere um problema de classificação de $K = 2$ classes, com n entradas $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ de dimensão D, das quais n_1 entradas pertencem à classe C_1 e n_2 pertencem à classe C_2. Projetando os vetores de entrada $\mathbf{x}_i$, $i = 1, 2, \ldots, n$, para uma dimensão por meio do produto escalar

$$y_i = \mathbf{w}^T \mathbf{x}_i, \tag{E.1}$$

obtêm-se n exemplos y_i, $i = 1, 2, \ldots n$ divididos em dois subconjuntos $\mathcal{Y}_1$ e $\mathcal{Y}_2$. É possível classificar a entrada como pertencente a classe C_k por meio da comparação com um limiar, classificando, por exemplo, a entrada como pertencente a classe C_1, se $y_i \geq -w_0$, e como pertencente a C_2, caso contrário.

Em geral, a projeção em uma dimensão leva a uma considerável perda de informação e classes que eram bem separáveis no espaço D-dimensional original podem tornar-se sobrepostas. Porém, ajustando os componentes do vetor de pesos $\mathbf{w}$, é possível escolher uma projeção que maximiza a separação de classes. A Figura E.1 ilustra uma direção de projeção que maximiza essa separação.

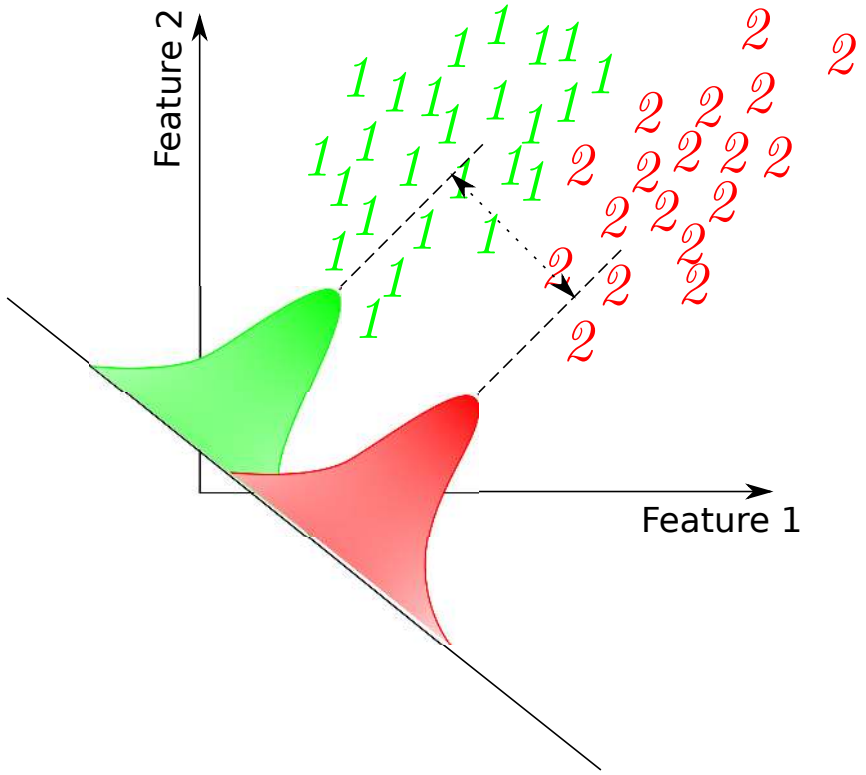


Figura E.1: Ilustração da direção de projeção que maximiza a separação de classes.

Se $\|\mathbf{w}\| = 1$, cada y_i é a projeção de seu correspondente $\mathbf{x}_i$ em uma linha que segue a direção

de $\mathbf{w}$. A magnitude de $\mathbf{w}$, na verdade, não tem uma importância real, sendo apenas um valor de escala de y. Porém, determinar a melhor direção de $\mathbf{w}$ é fundamental para uma boa separação das classes [20].

Uma medida de separação das classes é a diferença das projeções das médias. A partir das médias de cada classe,

$$\begin{aligned}\mathbf{m}_1 &= \frac{1}{n_1}\sum_{\mathbf{x}\in C_1}\mathbf{x} \\ \mathbf{m}_2 &= \frac{1}{n_2}\sum_{\mathbf{x}\in C_2}\mathbf{x},\end{aligned} \tag{E.2}$$

pode-se obter as médias dos pontos projetados

$$\tilde{m}_k = \mathbf{w}^T\mathbf{m}_k \tag{E.3}$$

para cada classe C_k. Essa medida de separação sugere que devemos escolher $\mathbf{w}$ de forma a maximizar

$$|\tilde{m}_1 - \tilde{m}_2| = \left|\mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)\right|, \tag{E.4}$$

e podemos fazer essa diferença tão grande quanto desejarmos apenas mudando a escala de $\mathbf{w}$ [36].

Há ainda um problema de sobreposição das classes quando elas são projetadas na linha que interliga as suas médias. Essa dificuldade surge das covariâncias não diagonais na distribuição de classes. Para obter uma boa separação dos dados projetados, queremos que a diferença entre as médias seja grande de forma relativa a alguma medida de desvio-padrão para cada classe. Assim, a ideia proposta por Fisher é de maximizar a função que fornece a maior separação entre as médias das classes projetadas e que fornece também a menor variância dentro de cada classe, minimizando a sobreposição de classes [20].

Com a variância dentro da classe C_k dos dados projetados dada por

$$\tilde{s}_k^2 = \sum_{y\in\mathcal{Y}_k}(y - \tilde{m}_k)^2, \tag{E.5}$$

é possível definir a variância total dentro de cada classe (*total within-class scatter*) por $\tilde{s}_1^2 + \tilde{s}_2^2$. O critério de Fisher é definido como a razão da variância entre classes pela variância dentro de cada classe, sendo dada por

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}. \tag{E.6}$$

Para reescrever $J(\cdot)$ como uma função explícita de $\mathbf{w}$, definem-se as matrizes de covariância $\mathbf{S}_W$ e $\mathbf{S}_B$. A matriz $\mathbf{S}_W$ é a matriz de covariância dentro de cada classe, dada por

$$\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2, \tag{E.7}$$

em que

$$\mathbf{S}_k = \sum_{\mathbf{x} \in C_k} (\mathbf{x} - \mathbf{m}_k)(\mathbf{x} - \mathbf{m}_k)^T, \tag{E.8}$$

e a matriz $\mathbf{S}_B$ é a matriz de covariância entre classes definida por

$$\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T. \tag{E.9}$$

Com essas definições, é possível escrever

$$\begin{aligned} \tilde{s}_k^2 &= \sum_{\mathbf{x} \in C_k} (\mathbf{w}^T \mathbf{x} - \mathbf{w}^T \mathbf{m}_k)^2 \\ &= \sum_{\mathbf{x} \in C_k} \mathbf{w}^T (\mathbf{x} - \mathbf{m}_k)(\mathbf{x} - \mathbf{m}_k)^T \mathbf{w} \\ &= \mathbf{w}^T \mathbf{S}_k \mathbf{w} \end{aligned} \tag{E.10}$$

e obtém-se

$$\tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}^T \mathbf{S}_W \mathbf{w}. \tag{E.11}$$

Analogamente, obtém-se a diferença das médias projetadas como

$$\begin{aligned} (\tilde{m}_1 - \tilde{m}_2)^2 &= (\mathbf{w}^T \mathbf{m}_1 - \mathbf{w}^T \mathbf{m}_2)^2 \\ &= \mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{w} \\ &= \mathbf{w}^T \mathbf{S}_B \mathbf{w}. \end{aligned} \tag{E.12}$$

A Equação (E.6) pode ser reescrita como

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}, \tag{E.13}$$

que é denominada quociente de Rayleigh generalizado.

Para encontrar o maior valor do quociente generalizado, considere um problema da forma

$$L(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{A} \mathbf{w}}{\mathbf{w}^T \mathbf{w}}, \tag{E.14}$$

em que $\mathbf{A} \in \mathbb{R}^{n \times n}$ é uma matriz simétrica. Essa expressão é denominada quociente de Rayleigh e representa a forma quadrática normalizada da matriz $\mathbf{A}$. É possível demonstrar que

$$\max_{\mathbf{w} \in \mathbb{R}^n : \mathbf{w} \neq 0} \frac{\mathbf{w}^T \mathbf{A} \mathbf{w}}{\mathbf{w}^T \mathbf{w}} = \lambda_{\max}, \tag{E.15}$$

quando $\mathbf{w}$ é o autovetor associado ao maior autovalor de $\mathbf{A}$ [135].

Para provar esse teorema, primeiramente, o quociente de Rayleigh é simplificado impondo-se $\|\mathbf{w}\| = 1$. Como a função $L(\mathbf{w})$ a ser maximizada é invariante com respeito a multiplicação

de $\mathbf{w}$ por uma constante α, sempre é possível escolher $\mathbf{w}$ tal que o denominador $\mathbf{w}^T\mathbf{w} = 1$ [37]. O problema torna-se então resolver

$$\max_{\mathbf{w}\in\mathbb{R}^n:\|\mathbf{w}\|=1} \mathbf{w}^T\mathbf{A}\mathbf{w}. \tag{E.16}$$

Seja $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ a decomposição em autovalores e autovetores, em que $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ é ortogonal e $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ é diagonal com valores ordenados de forma decrescente. Então, a Equação (E.16) pode ser desenvolvida como

$$\begin{aligned}\mathbf{w}^T\mathbf{A}\mathbf{w} &= \mathbf{w}^T(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T)\mathbf{w} \\ &= (\mathbf{w}^T\mathbf{V})\mathbf{\Lambda}(\mathbf{V}^T\mathbf{w}) \\ &= \mathbf{z}^T\mathbf{\Lambda}\mathbf{z},\end{aligned} \tag{E.17}$$

em que $\mathbf{z} = \mathbf{V}^T\mathbf{w}$ também é um vetor unitário, pois

$$\|\mathbf{z}\|^2 = \mathbf{z}^T\mathbf{z} = \mathbf{w}^T\mathbf{V}\mathbf{V}^T\mathbf{w} = 1, \tag{E.18}$$

sendo $\mathbf{V}$ ortogonal (e, portanto, $\mathbf{V}$ vezes sua transposta fornece a matriz identidade), e $\|\mathbf{w}\| = 1$ como imposto anteriormente.

Assim, o problema de otimização original se reduz a encontrar

$$\max_{\mathbf{z}\in\mathbb{R}^n:\|\mathbf{z}\|=1} \mathbf{z}^T\mathbf{\Lambda}\mathbf{z}. \tag{E.19}$$

Definindo $\mathbf{z}$ como $\mathbf{z} = (z_1, \ldots, z_2)^T$, a Expressão (E.19) pode ser descrita como

$$\mathbf{z}^T\mathbf{\Lambda}\mathbf{z} = \sum_{i=1}^{n} \lambda_i z_i^2, \tag{E.20}$$

pois $\mathbf{\Lambda}$ possui apenas elementos em suas diagonais. Essa forma quadrática é sujeita à imposição de $z_1^2 + \ldots + z_n^2 = 1$, e, como $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ quando $z_1^2 = 1$ e $z_2^2 = \ldots = z_n^2 = 0$, o maior valor da Expressão (E.20) é igual ao próprio maior autovalor λ_1.

Voltando ao problema do quociente generalizado, como $\mathbf{S}_W$ e $\mathbf{S}_B$ são simétricas e positivas semi-definidas [36], é possível encontrar uma matriz denominada raiz quadrada $\mathbf{S}_W^{\frac{1}{2}}$ cuja decomposição em autovalores e autovetores é

$$\mathbf{S}_W^{\frac{1}{2}} = \mathbf{V}_{S_W}\mathbf{\Lambda}_{S_W}^{\frac{1}{2}}\mathbf{V}_{S_W}^T, \tag{E.21}$$

com autovalores que resultam da raiz quadrada dos autovalores de $\mathbf{S}_W$.

Seja $\mathbf{b} = \mathbf{S}_W^{\frac{1}{2}}\mathbf{w}$. É possível escrever o denominador da Equação (E.13) como

$$\mathbf{w}^T\mathbf{S}_W\mathbf{w} = \mathbf{w}^T\mathbf{S}_W^{\frac{1}{2}}\mathbf{S}_W^{\frac{1}{2}}\mathbf{w} = \mathbf{b}^T\mathbf{b}, \tag{E.22}$$

e substituir $\mathbf{w} = \mathbf{S}_W^{-\frac{1}{2}}\mathbf{b}$ no numerador transformando o problema do quociente generalizado na Equação (E.14).

O objetivo é então maximizar

$$\max_{\mathbf{b}\in\mathbb{R}^n:\|\mathbf{b}\|=1} \mathbf{b}^T(\mathbf{S}_W^{-\frac{1}{2}})^T\mathbf{S}_B\mathbf{S}_W^{-\frac{1}{2}}\mathbf{b}, \tag{E.23}$$

encontrando o maior autovalor da matriz $\mathbf{A} = (\mathbf{S}_W^{-\frac{1}{2}})^T\mathbf{S}_B\mathbf{S}_W^{-\frac{1}{2}}$. Pode-se diagonalizar $\mathbf{A}$ por

$$\begin{aligned} \mathbf{V}_A^T A\mathbf{V}_A &= \mathbf{\Lambda} \\ \mathbf{V}_A^T\left((\mathbf{S}_W^{-\frac{1}{2}})^T\mathbf{S}_B\mathbf{S}_W^{-\frac{1}{2}}\right)\mathbf{V}_A &= \mathbf{\Lambda} \\ \mathbf{V}^T\mathbf{S}_B\mathbf{V} &= \mathbf{\Lambda}, \end{aligned} \tag{E.24}$$

sendo $\mathbf{V} = (\mathbf{S}_W^{-\frac{1}{2}})\mathbf{V}_A$ e $\mathbf{V}_A$ os autovalores da matriz $\mathbf{A}$.

Sabe-se também que $(\mathbf{S}_W^{-\frac{1}{2}})^T\mathbf{S}_W\mathbf{S}_W^{-\frac{1}{2}} = \mathbf{I}$, e, portanto,

$$\begin{aligned} \mathbf{V}_A^T\left((\mathbf{S}_W^{-\frac{1}{2}})^T\mathbf{S}_W\mathbf{S}_W^{-\frac{1}{2}}\right)\mathbf{V}_A &= \mathbf{I} \\ \mathbf{V}^T\mathbf{S}_W\mathbf{V} &= \mathbf{I}. \end{aligned} \tag{E.25}$$

Com isso, temos o sistema

$$\begin{cases} \mathbf{V}^T\mathbf{S}_B\mathbf{V} = \mathbf{\Lambda} \\ \mathbf{V}^T\mathbf{S}_W\mathbf{V} = \mathbf{I}. \end{cases}$$

Multiplicando a segunda equação à direita por Λ e igualando as duas equações, o sistema torna-se um problema de resolução de autovalores dado por

$$\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{V} = \mathbf{V}\mathbf{\Lambda}. \tag{E.26}$$

Apesar da Equação (E.13) não ser um discriminante, mas sim uma escolha específica de direção da projeção dos dados em uma dimensão, é possível usar os dados projetados para construir um discriminante, escolhendo um limiar y_0 tal que um ponto é classificado como pertencente à classe C_1 se $y(\mathbf{x}) \geq y_0$, e como pertencente à classe C_2 caso contrário. Por exemplo, pode-se modelar as funções de densidade de probabilidade usando distribuições gaussianas e depois encontrar os parâmetros dessas distribuições por máxima verossimilhança. A hipótese de distribuição gaussiana é justificada pelo Teorema do Limite Central, uma vez que a Equação (E.1) é a soma de um conjunto de variáveis aleatórias [20].

É possível evitar a extração dos discriminantes, que envolvem um problema de autovalores e autovetores generalizado, pelas funções de classificação de Fisher calculadas sobre as variáveis diretamente. Essa abordagem é diferente do uso de funções discriminantes canônicas da LDA,

exceto na classificação de apenas 2 classes, em que o discriminante canônico é o mesmo que o de Fisher. Assim, o discriminante linear de Fisher em essência é uma técnica de redução de dimensão, não um discriminante. Para classificação binária, estabelece-se um limiar ótimo t e, para classificação de múltiplas classes, modela-se uma distribuição gaussiana, encontram-se as probabilidades marginais e utiliza-se Bayes para encontrar a probabilidade condicional.

A generalização natural do discriminante de Fisher para $K > 2$ classes envolve $K - 1$ funções discriminantes. Assumindo que a dimensão D do espaço das entradas é maior do que K, a projeção do espaço D-dimensional para um espaço $(K-1)$-dimensional é conseguida com $K - 1$ funções discriminantes $y_k = \mathbf{w}_k^T \mathbf{x}$, em que $k = 1, \ldots, K - 1$. Essas funções podem ser agrupadas em um vetor $\mathbf{y}$, e $\mathbf{w}_k$ podem ser agrupados como colunas de uma matriz $\mathbf{W}$ de dimensão $D \times (K - 1)$, com $\mathbf{y} = \mathbf{W}^T \mathbf{x}$. A matriz de covariância dentro de cada classe é dada por

$$\mathbf{S}_W = \sum_{k=1}^{K} \mathbf{S}_k, \tag{E.27}$$

em que

$$\mathbf{S}_k = \sum_{n \in C_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T, \tag{E.28}$$

$$\mathbf{m}_k = \frac{1}{n_k} \sum_{n \in C_k} \mathbf{x}_n, \tag{E.29}$$

e n_k é o número de padrões na classe C_k. A matriz total de covariância é

$$\mathbf{S}_T = \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^T, \tag{E.30}$$

em que $\mathbf{m}$ é a média do conjunto total de dados

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n = \frac{1}{N} \sum_{k=1}^{K} n_k \mathbf{m}_k, \tag{E.31}$$

e $N = \sum_k n_k$ é o total de dados. A matriz $\mathbf{S}_T$ pode ser decomposta como

$$\mathbf{S}_T = \mathbf{S}_W + \mathbf{S}_B, \tag{E.32}$$

em que

$$\mathbf{S}_B = \sum_{k=1}^{K} n_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T. \tag{E.33}$$

As matrizes de covariância foram definidas no espaço original, mas podem ser definidas matrizes semelhantes no espaço $(K-1)$-dimensional de y,

$$\tilde{\mathbf{S}}_W = \sum_{k=1}^{K} \sum_{n \in C_k} (\mathbf{y}_n - \tilde{\mathbf{m}}_k)(\mathbf{y}_n - \tilde{\mathbf{m}}_k)^T, \tag{E.34}$$

e

$$\tilde{\mathbf{S}}_B = \sum_{k=1}^{K} n_k(\tilde{\mathbf{m}}_k - \tilde{\mathbf{m}})(\tilde{\mathbf{m}}_k - \tilde{\mathbf{m}})^T, \tag{E.35}$$

em que

$$\tilde{\mathbf{m}}_k = \frac{1}{n_k} \sum_{n \in C_k} \mathbf{y}_n$$

$$\tilde{\mathbf{m}} = \frac{1}{N} \sum_{k=1}^{K} n_k \tilde{\mathbf{m}}_k. \tag{E.36}$$

Analogamente ao caso de duas classes, é possível mostrar que

$$\tilde{\mathbf{S}}_W = \mathbf{W}^T \mathbf{S}_W \mathbf{W}^T \tag{E.37}$$

e

$$\tilde{\mathbf{S}}_B = \mathbf{W}^T \mathbf{S}_B \mathbf{W}^T. \tag{E.38}$$

Essas equações mostram como as matrizes de covariância entre classes e dentro de cada classe são transformadas pela projeção para o espaço de dimensão menor. Novamente, procura-se a matriz de transformação $\mathbf{W}$ que maximiza a razão entre a matriz de covariância entre classes e a matriz de covariância dentro de cada classe. Uma medida possível é o determinante dessas matrizes, uma vez que o determinante é o produto de autovalores e, portanto, é o produto de "variâncias" nas direções principais. Usando essa medida, obtém-se a função

$$\mathbf{J}(\mathbf{W}) = \frac{\left|\tilde{\mathbf{S}}_B\right|}{\left|\tilde{\mathbf{S}}_W\right|} = \frac{\left|\mathbf{W}^T \mathbf{S}_B \mathbf{W}\right|}{\left|\mathbf{W}^T \mathbf{S}_W \mathbf{W}\right|}. \tag{E.39}$$

Analogamente ao caso de duas classes, maximizar tal critério é resolver um problema de autovalores, com $\mathbf{W}$ dado pelos autovetores de $\mathbf{S}_W^{-1}\mathbf{S}_B$ que correspondem aos maiores autovalores [37].

Em geral, a solução de $\mathbf{W}$ não é única e as transformações permitidas incluem rotação e mudança na escala dos eixos de várias formas, mas essas transformações são todas lineares de um espaço $(K-1)$-dimensional para um espaço $(K-1)$-dimensional.

E.2 Métodos Probabilísticos Generativos

Seja a densidade de probabilidade condicional $p(\mathbf{x} \mid C_k)$ e a probabilidade marginal de uma classe $p(C_k)$. Usando o Teorema de Bayes é possível calcular as probabilidades condicionais $p(C_k \mid \mathbf{x})$. Considerando um problema de $K = 2$ classes, a probabilidade condicional (ou *a posteriori*) da classe C_1 pode ser escrita como

$$
\begin{aligned}
p(C_1 \mid x) &= \frac{p(\mathbf{x} \mid C_1)p(C_1)}{p(\mathbf{x} \mid C_1)p(C_1) + p(\mathbf{x} \mid C_2)p(C_2)} \\
&= \frac{1}{1 + \exp(-a)} = \sigma(a),
\end{aligned} \tag{E.40}
$$

em que

$$
\begin{aligned}
a &= \ln \frac{p(\mathbf{x} \mid C_1)p(C_1)}{p(\mathbf{x} \mid C_2)p(C_2)} \\
a &= \ln \frac{p(\mathbf{x} \mid C_1)}{p(\mathbf{x} \mid C_2)} + \ln \frac{p(C_1)}{p(C_2)},
\end{aligned} \tag{E.41}
$$

e $\sigma(a)$ é a função sigmoidal. A inversa da função sigmoidal é a função *logit*

$$
a = \ln \left(\frac{\sigma}{1 - \sigma} \right), \tag{E.42}
$$

que representa a razão das probabilidades $\ln [p(C_1 \mid \mathbf{x})/p(C_2 \mid \mathbf{x}))]$, conhecida também como *log odds* [20].

Para o caso de $K > 2$ classes, temos a função *softmax*

$$
\begin{aligned}
p(C_k \mid \mathbf{x}) &= \frac{p(\mathbf{x} \mid C_k)p(C_k)}{\sum_j p(\mathbf{x} \mid C_j)p(C_j)} \\
&= \frac{\exp(a_k)}{\sum_j \exp(a_j)},
\end{aligned} \tag{E.43}
$$

em que

$$
a_k = \ln \left(p(\mathbf{x} \mid C_k)p(C_k) \right). \tag{E.44}
$$

Assumindo que os dados são provenientes de uma distribuição gaussiana multivariada $\mathbf{X} \sim \mathcal{N}(\mu, \mathbf{V})$, e supondo matrizes de covariância $\mathbf{V}_k$ idênticas para todas as classes ($\mathbf{V}_k = \mathbf{V}$), temos para a classe C_k

$$
p(\mathbf{x} \mid C_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\mathbf{V}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \mathbf{V}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)}. \tag{E.45}
$$

Considerando, novamente, o caso de duas classes, a probabilidade condicional de C_1 pode ser escrita como

$$
p(C_1 \mid x) = \sigma(\mathbf{w}^T \mathbf{x} + w_0), \tag{E.46}
$$

em que, aplicando-se a Equação (E.45) na Equação (E.41), definem-se

$$
\mathbf{w} = \mathbf{V}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{E.47}
$$

$$
w_0 = -\frac{1}{2} \boldsymbol{\mu}_1^T \mathbf{V}^{-1} \boldsymbol{\mu}_1 + \frac{1}{2} \boldsymbol{\mu}_2^T \mathbf{V}^{-1} \boldsymbol{\mu}_2 + \ln \frac{p(C_1)}{p(C_2)}. \tag{E.48}
$$

A igualdade das matrizes de covariância ocasiona o cancelamento dos fatores de normalização e da parte quadrática dos expoentes. Assim, para qualquer par de classes, o limite de decisão é linear no espaço das entradas, com uma função de x no argumento da função sigmoidal que é linear. Sem a igualdade de covariância, o termo quadrático não é cancelado, e obtém-se uma função discriminante quadrática em x (*quadratic discriminant analysis* – QDA), que permite limites de decisão mais flexíveis, porém exige um maior número de parâmetros a serem estimados [37].

As probabilidades marginais $p(C_k)$ apenas aparecem no *bias* w_0, tendo o efeito de deslocar de forma paralela os limites de decisão e, de forma geral, os contornos das probabilidades condicionais.

Para o caso geral de K classes, tem-se

$$a_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0}, \tag{E.49}$$

em que, usando a Equação (E.45) nas Equações (E.43) e (E.44), definem-se

$$\begin{aligned} \mathbf{w}_k &= \mathbf{V}^{-1}\boldsymbol{\mu}_k \\ w_0 &= -\frac{1}{2}\boldsymbol{\mu}_k^T \mathbf{V}^{-1}\boldsymbol{\mu}_k + \ln p(C_k). \end{aligned} \tag{E.50}$$

Novamente, $a_k(\mathbf{x})$ são funções lineares de $\mathbf{x}$ como consequência do cancelamento do termo quadrático. A fronteira de decisão resultante, correspondente à mínima taxa de erro de classificação, ocorre quando duas das maiores probabilidades condicionais são iguais.

E.3 Máxima Verossimilhança

Dadas as densidades de probabilidade condicionais $p(\mathbf{x} \mid C_k)$, é possível determinar os valores dos parâmetros, junto às probabilidades marginais $p(C_k)$, usando a máxima verossimilhança. Considere um problema de duas classes, com distribuições gaussianas e matrizes de covariância idênticas, e um conjunto de dados $\{\mathbf{x}_n, t_n\}$, com $n = 1, \ldots, N$, em que $t_n = 1$ indica a classe C_1 e $t_n = 0$, a classe C_2. Definem-se as probabilidades marginais $p(C_1) = p$ e $p(C_2) = 1 - p$. Então, para um dado da classe C_1 tem-se

$$p(\mathbf{x}_n, C_1) = p(C_1)p(\mathbf{x}_n \mid C_1) = p\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_1, \mathbf{V}), \tag{E.51}$$

e, para um dado da classe C_2,

$$p(\mathbf{x}_n, C_2) = p(C_2)p(\mathbf{x}_n \mid C_2) = (1 - p)\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_2, \mathbf{V}). \tag{E.52}$$

A função de verossimilhança é dada por

$$p(\boldsymbol{t} \mid p, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \mathbf{V}) = \prod_{n=1}^{N} [p\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_1, \mathbf{V})]^{t_n} [(1-p)\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_2, \mathbf{V})]^{1-t_n}, \tag{E.53}$$

em que $\boldsymbol{t} = (t_1, \ldots, t_N)^T$. Em geral, é interessante maximizar o logaritmo da função de verossimilhança.

Considerando, primeiro, a maximização com respeito a p, os termos da função de log-verossimilhança que dependem de p são

$$\sum_{n=1}^{N} \{t_n \ln(p) + (1-t_n) \ln(1-p)\}, \tag{E.54}$$

e igualando a derivada com respeito a p a zero, obtém-se

$$p = \frac{1}{N} \sum_{n=1}^{N} t_n = \frac{n_1}{N}, \tag{E.55}$$

em que n_1 é o número total de pontos da classe C_1 e n_2, da classe C_2. Assim, a estimativa da máxima verossimilhança para p é simplesmente a fração de pontos que são da classe C_1, como esperado. Esse resultado é facilmente generalizado para o caso de múltiplas classes [20].

Realizando a maximização com respeito a $\boldsymbol{\mu}_1$, os termos da função de log-verossimilhança que dependem de $\boldsymbol{\mu}_1$ são

$$\sum_{n=1}^{N} t_n \ln(\mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_1, \mathbf{V})) = -\frac{1}{2} \sum_{n=1}^{N} t_n (\mathbf{x}_n - \boldsymbol{\mu}_1)^T \mathbf{V}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_1) + \text{constante}, \tag{E.56}$$

e igualando a derivada com respeito a $\boldsymbol{\mu}_1$ a zero, obtém-se

$$\boldsymbol{\mu}_1 = \frac{1}{n_1} \sum_{n=1}^{N} t_n \mathbf{x}_n, \tag{E.57}$$

que é simplesmente a média de todas as entradas referentes à classe C_1. Analogamente,

$$\boldsymbol{\mu}_2 = \frac{1}{n_2} \sum_{n=1}^{N} (1-t_n) \mathbf{x}_n. \tag{E.58}$$

Maximizando, por último, a solução para a matriz $\mathbf{V}$,

$$\begin{aligned}
&-\frac{1}{2} \sum_{n=1}^{N} t_n \ln|\mathbf{V}| - \frac{1}{2} \sum_{n=1}^{N} t_n (\mathbf{x}_n - \boldsymbol{\mu}_1)^T \mathbf{V}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_1) \\
&-\frac{1}{2} \sum_{n=1}^{N} (1-t_n) \ln|\mathbf{V}| - \frac{1}{2} \sum_{n=1}^{N} (1-t_n) (\mathbf{x}_n - \boldsymbol{\mu}_2)^T \mathbf{V}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_2) \\
&= -\frac{N}{2} \ln|\mathbf{V}| - \frac{N}{2} \mathrm{Tr}\left\{\mathbf{V}^{-1} \mathbf{S}\right\},
\end{aligned} \tag{E.59}$$

em que

$$\mathbf{S} = \frac{n_1}{N}\mathbf{S}_1 + \frac{n_2}{N}\mathbf{S}_2, \tag{E.60}$$

e

$$\mathbf{S}_k = \frac{1}{n_k}\sum_{n\in C_k}(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T, \tag{E.61}$$

para $k = 1, 2$. Usando o resultado padrão de máxima verossimilhança para uma distribuição gaussiana, tem-se que $\mathbf{V} = \mathbf{S}$, que representa uma média com pesos das matrizes de covariância associadas com cada uma das duas classes separadamente. Esse resultado pode ser estendido para um problema de K classes para a obtenção dos parâmetros.

E.4 A LDA com Pesos

Em aplicações em que os dados são desbalanceados entre as classes, é comum utilizar técnicas de subamostragem ou de sobreamostragem dos dados, como explicado no Relatório Parcial e no Apêndice B. No entanto, no caso da LDA, muitos autores [24, 32–35] têm usado pesos aplicados às médias e às matrizes de covariância para evitar desperdiçar dados das classes mais representadas.

Assim, atribuem-se pesos p_k para cada classe k nas matrizes $\mathbf{S}_W$ e $\mathbf{S}_B$

$$\mathbf{S}_W = \sum_{k=1}^{K} p_k \sum_{n\in C_k}(\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T, e \tag{E.62}$$

$$\mathbf{S}_B = \sum_{k=1}^{K} p_k n_k(\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T, \tag{E.63}$$

em que n_k é o número de dados na classe C_k, $\mathbf{m}_k$ é a média desses dados, e $\mathbf{m}$ é a média global ponderada

$$\mathbf{m} = \frac{\sum_{k=1}^{K} p_k \sum_{n\in C_k}\mathbf{x}_n}{\sum_{k=1}^{K} p_k n_k}. \tag{E.64}$$

Para maximizar $J(\mathbf{W})$, deve-se encontrar a decomposição de $\mathbf{S}_W^{-1}\mathbf{S}_B$

$$\mathbf{Q}\mathbf{D}\mathbf{Q}^T = \mathbf{S}_W^{-1}\mathbf{S}_B, \tag{E.65}$$

em que $\mathbf{D}$ é uma matriz diagonal contendo os autovalores e $\mathbf{Q}$ possui as colunas com os autovetores associados. A matriz $\mathbf{W}$ usada para projetar os dados é composta pelos autovetores associados aos maiores autovalores, sendo a menor redução de dimensionalidade possível com o uso de $K - 1$ autovetores. Além disso, as colunas de $\mathbf{Q}$ devem ser redimensionadas de modo que

$$\mathbf{Q}^T\frac{\mathbf{S}_W}{\sum_{k=1}^{K} p_k n_k}\mathbf{Q} = \mathbf{Q}^T\mathbf{V}\mathbf{Q} = \mathbf{I}, \tag{E.66}$$

em que p_k é o peso da k-ésima classe, n_k é o número de dados dessa classes, $\mathbf{I}$ é a matriz identidade e $\mathbf{V}$ é a matriz de covariância da distribuição gaussiana.

Por fim, encontram-se as probabilidade condicionais de determinada entrada pertencer a uma classe k por meio das Equações (E.43) e (E.50).

F Classificação Binária

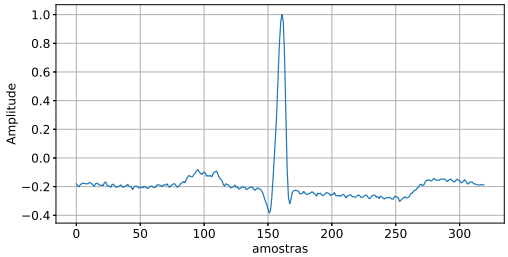
As primeiras redes desenvolvidas no projeto tiveram como objetivo apenas a detecção das arritmias cardíacas. Para isso, foram testadas diversas entradas possíveis para a rede MLP, estudando-se o efeito de cada tipo de entrada de forma separada. Posteriormente foram selecionadas as entradas que apresentaram os melhores resultados para realizar combinações entre elas.

Basicamente, as entradas foram originadas considerando três naturezas distintas: o sinal de ECG original, o módulo dos coeficientes da transformada discreta de Fourier de um batimento e os coeficientes da transformada discreta de wavelet de três batimentos, com diferentes funções *wavelet*. Os coeficientes das transformadas foram normalizados. As configurações dessas entradas estão listadas na Tabela F.1, em que "Tamanho" representa o número de amostras.

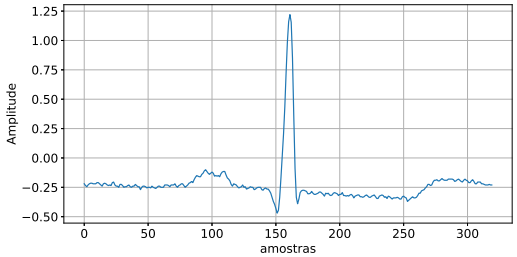
Tabela F.1: Configurações das entradas de naturezas distintas.

Configuração	Descrição da entrada	Tamanho
A1	Sinal original de ECG com 1 batimento e valores normalizados	320
A2	Sinal original de ECG com 1 batimento, sem normalizar os valores	320
A3	Sinal original de ECG com 3 batimentos e valores normalizados	960
A4	Sinal original de ECG com 3 batimentos, sem normalizar os valores	960
A5	Módulo dos Coeficientes da transformada discreta de Fourier até a frequência de 60 Hz de 1 batimento	54
A6	Coeficientes da transformada discreta de wavelet para *haar*	961
A7	Coeficientes da transformada discreta de wavelet para *db2*	978
A8	Coeficientes da transformada discreta de wavelet para *db4*	1005
A9	Coeficientes da transformada discreta de wavelet para *db6*	1022
A10	Coeficientes da transformada discreta de wavelet para *sym6*	1022

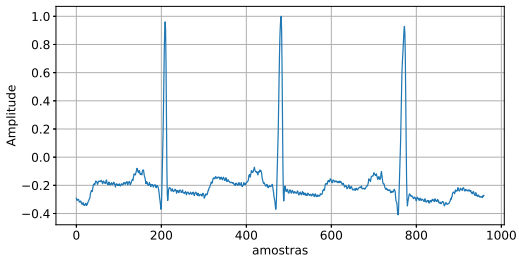
Exemplos dessas entradas são mostrados na Figura F.1, considerando apenas a primeira derivação dos sinais de ECG dos pacientes. A segunda derivação não foi utilizada nessa etapa inicial do projeto.



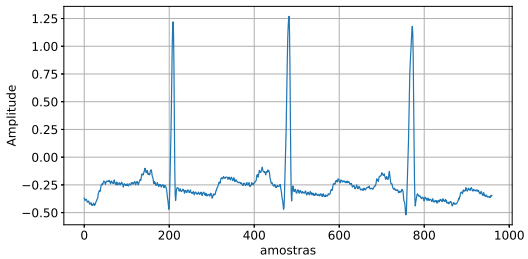
(a) A1



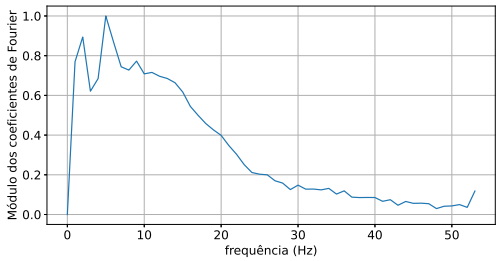
(b) A2



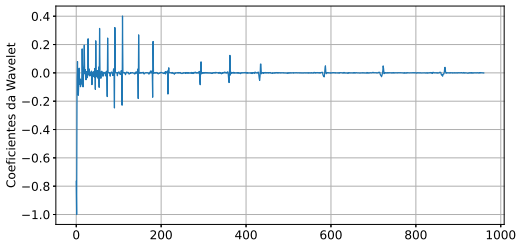
(c) A3



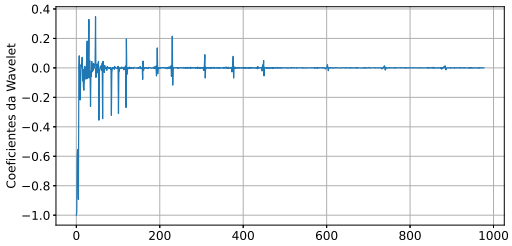
(d) A4



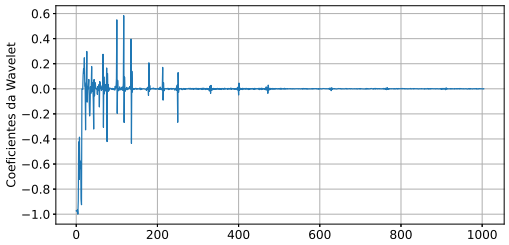
(e) A5



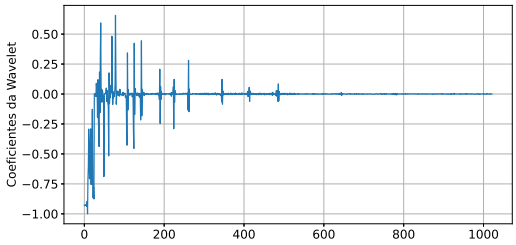
(f) A6



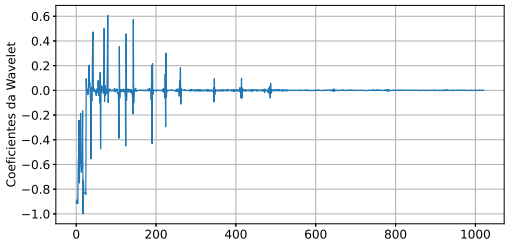
(g) A7



(h) A8



(i) A9



(j) A10

Figura F.1: Entradas para cada configuração da Tabela F.1.

Como explicado na Seção 3.3, a quantidade de batimentos normais no conjunto de dados

é aproximadamente o dobro da quantidade de batimentos com arritmia. Para trabalhar com os dados, metade do número de batimentos normais de cada paciente foi desconsiderado nos conjuntos de teste e de treinamento. O número de batimentos considerado para cada classe está apresentado na Tabela F.2.

Tabela F.2: Número de batimentos de cada classe para os conjuntos de teste e de treinamento.

Classe	Conjunto de treinamento	Conjunto de teste
Normal	19075	36386
Arritmia	12904	13256

A saída da classe normal foi considerada como 0 e a da classe de arritmia como 1. Em todas as configurações de entrada, a rede neural foi construída como indicado na Tabela F.3.

Tabela F.3: Rede Neural utilizada para a detecção de arritmias.

	Caracterização da Rede I
Camada de entrada	Variável de acordo com a configuração de entrada
Camada oculta 1	64 neurônios
Camada oculta 2	64 neurônios
Camada de saída	1 neurônio
Função de ativação	ReLU (ocultas) Sigmoidal (saída)
Função custo	Entropia cruzada
Passo de aprendizado	0,001
Tamanho do *mini* batch	2048
Dropout	0,45 (camada 1) 0,35 (camada 2)
Otimizador	Adam, com $\beta_1 = 0{,}9$ e $\beta_2 = 0{,}999$

A rede foi treinada com 500 épocas e os resultados da Acurácia (Acc), da Sensibilidade (Se) e da Precisão (P) para as classes normal e de arritmia são apresentados na Tabela F.4. O número de parâmetros da rede para cada configuração de entrada também é apresentado para comparação.

Os melhores resultados em termos de acurácia foram obtidos com as configurações A3, A6 e A10. Nessas configurações, a sensibilidade e a precisão da classe de arritmia também são maiores do que nos demais casos, ainda que as métricas da classe normal sejam um pouco menores. A configuração que demonstrou o pior desempenho foi a A5, caso em que a rede é alimentada com o módulo dos coeficientes da transformada discreta de Fourier, que possuem apenas a informação da frequência.

Nota-se que a transformada discreta de wavelet mantém boa parte da informação do sinal de ECG, atingindo resultados equivalentes ou superiores aos obtidos quando a rede é alimentada

Tabela F.4: Resultados do treinamento das redes, considerando as configurações da Tabela F.1 e a estrutura de MLP da Tabela F.3. Para cada métrica, os três melhores resultados foram colocados em negrito.

Configuração	Acurácia(%)	Normal		Arritmia		Parâmetros
		Se(%)	P(%)	Se(%)	P(%)	
A1	80,42	**91,86**	80,41	**80,44**	59,93	24769
A2	74,10	75,20	87,71	71,06	51,06	24769
A3	**81,74**	**83,56**	**90,79**	76,73	**62,97**	65729
A4	79,64	81,34	89,92	74,98	59,42	65729
A5	68,25	72,12	82,38	57,65	42,96	7745
A6	**81,90**	83,52	**91,04**	**77,45**	**63,12**	65793
A7	78,25	78,60	90,48	**77,29**	56,82	66881
A8	80,96	82,44	**90,74**	76,89	61,47	68609
A9	80,32	81,55	90,67	76,96	60,31	69697
A10	**83,44**	**86,73**	90,29	67,14	**74,41**	69697

com o sinal original. Isso mostra a importância da etapa de extração das características para alimentar a rede MLP.

Aumentando-se a profundidade da rede, obtiveram-se desempenhos piores. Usando a configuração A10, e realizando-se testes para uma rede com um maior número de camadas ($L = 3, 4$) e com um maior número de neurônios ($N_1 = 128, N_2 = 128$), os resultados obtidos foram equivalentes ou piores do que os apresentados na Tabela F.4. Da mesma forma, usando-se a função de ativação tangente hiperbólica, ao invés de $ReLU$, as métricas diminuíram.

Assim, mantendo a mesma configuração da Tabela F.3, realizou-se a combinação das três naturezas distintas, com as entradas A3, A5 e A10, como mostrado na Tabela F.5. As configurações combinadas dessas entradas são ilustradas na Figura F.2.

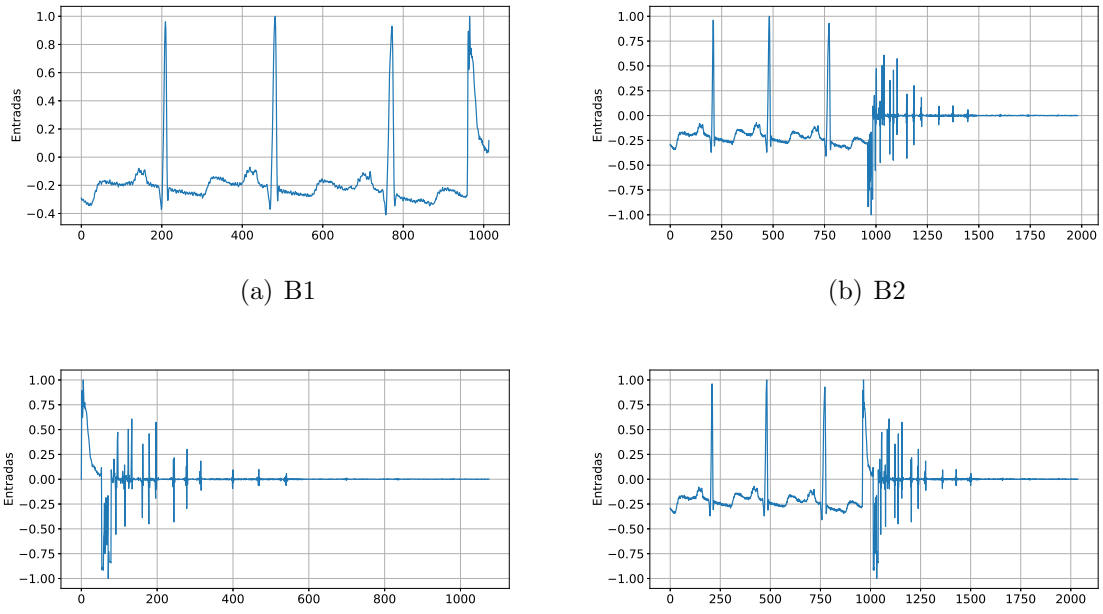
Tabela F.5: Configurações das combinações de entradas.

Configuração	Descrição da entrada	Tamanho
B1	A3 e A5	1014
B2	A3 e A10	1982
B3	A5 e A10	1076
B4	A3, A5 e A10	2036

Os resultados obtidos no treinamento da rede MLP para cada configuração da Tabela F.5 e seguindo a estrutura da Tabela F.3 estão mostrados na Tabela F.6. É possível observar que a entrada A5 não acrescenta informações que melhorem significativamente o desempenho da rede, uma vez que B1 mostrou um desempenho semelhante ao que foi obtido em A3. Já B2, B3 e B4 apresentaram resultados piores que os apresentados na Tabela F.4, indicando o *overfitting* do treinamento devido à quantidade de informações usadas.

Tabela F.6: Resultados do treinamento das redes, considerando as configurações da Tabela F.5 e a estrutura de MLP da Tabela F.3. Para cada métrica, o melhor resultado foi colocado em negrito.

Configuração	Acurácia(%)	Normal		Arritmia		Parâmetros
		Se(%)	P(%)	Se(%)	P(%)	
B1	**81,96**	**84,50**	90,26	74,97	**63,80**	69185
B2	79,81	80,38	**91,03**	**78,25**	59,23	131137
B3	77,19	80,50	87,39	68,11	55,99	73153
B4	79,94	80,83	90,79	77,49	59,55	134593



(a) B1

(b) B2

(c) B3

(d) B4

Figura F.2: Entradas para cada configuração da Tabela F.5.

G Efeito das Entradas em uma Rede MLP

Após a classificação binária da existência ou não de arritmia, desenvolveram-se redes neurais MLP para a classificação entre os diferentes tipos de arritmia, avaliando-se o efeito das entradas no desempenho. Para isso, as entradas consideradas foram baseadas em [24], permitindo a comparação do desempenho da rede proposta neste trabalho em relação aos classificadores de [24] e [25]. Assim, foram testadas diferentes entradas baseadas na combinação de três tipos de informação: os valores do sinal de ECG; as características dos intervalos RR e das durações do complexo QRS e da onda T, além da informação da existência ou não da onda P; e os valores da interpolação do sinal de ECG sugerida por [24].

Em [24], um dos elementos da extração de características usado é uma amostragem do sinal de ECG por meio de 19 pontos, sendo dez deles uniformemente espaçados em uma janela que se inicia no *QRS onset* e termina no *QRS offset*, e os outros nove, também uniformemente espaçados, em uma janela que se inicia no *QRS offset* e termina na Onda T. Como o sinal é discreto, a tomada dos pontos deve ser feita como uma reamostragem, para que eles sejam de fato uniformemente espaçados na janela em que estão presentes. Para isso, foi feita uma interpolação dos dados existentes. A Figura G.1 ilustra os pontos obtidos pela interpolação do sinal.

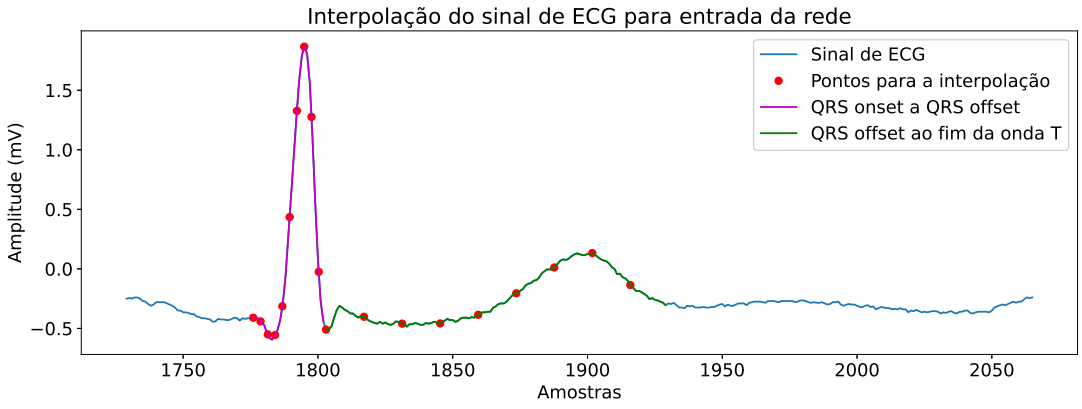


Figura G.1: Interpolação do sinal de ECG como entrada da rede.

Outras entradas usadas por [24] são referentes aos intervalos RR, às durações dos segmentos QRS e da onda T, e a um *booleano* indicando a presença ou não da Onda P. Os intervalos RR são formados por quatro informações: o valor do intervalo RR entre o batimento atual e o anterior; o valor do intervalo RR entre o batimento atual e o posterior; o intervalo RR médio de toda a gravação de um paciente; e o valor do intervalo RR médio local entre 10 batimentos adjacentes.

Os dados foram separados nas classes sugeridas pela AAMI, e a quantidade de batimentos total do banco de dados para cada classe está apresentado na Tabela G.1. As cinco classes possíveis de arritmia são: batimentos do nó SA (N), supraventriculares ectópicos (S), ventriculares ectópicos (V), fusão de batimentos normais e ventriculares ectópicos (F) e desconhecidos ou de marca-passo (Q).

Tabela G.1: Número de batimentos total do banco de dados para os conjuntos de teste e de treinamento no problema de classificação.

Classe	Conjunto de treinamento
N	90125
S	2781
V	7009
F	803
Q	15

Durante a segmentação dos batimentos, etapa que foi tratada na Subseção 3.4.2, foi necessário identificar os elementos do sinal de ECG, como o *QRS onset*, o *QRS offset* e a Onda T de forma automática, para que fosse possível realizar a interpolação desejada. Como esses elementos não foram identificados em todos os batimentos pelo módulo `ECGkit`, os batimentos que não tiveram essas janelas encontradas foram descartados. Isso reduziu a quantidade de batimentos disponíveis em cada classe. Como a classe normal possui um número muito maior de dados em relação às outras, a quantidade de batimentos normais foi reduzida selecionando-se um quarto do total para o uso na rede. A Tabela G.2 mostra o número de batimentos usados para cada classe.

Apesar de eliminar boa parte da classe normal, percebe-se ainda uma alta disparidade no número de cada classe, o que é prejudicial para o treinamento da rede. Para evitar reduzir ainda mais a quantidade de dados, trabalhou-se com a classificação em dados desbalanceados por meio do Aprendizado Sensível ao Custo, atribuindo-se pesos maiores na função custo para as classes minoritárias.

Foram consideradas as entradas listadas na Tabela G.3 para o problema de classificação

Tabela G.2: Número de batimentos usados para os conjuntos de teste e de treinamento no problema de classificação.

Classe	Conjunto de treinamento	Conjunto de teste
N	8534	8303
S	755	1319
V	2538	2034
F	396	376
Q	1	5

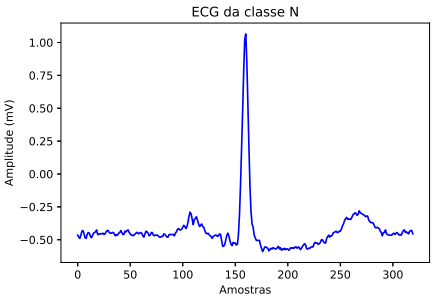
entre os cinco tipos de arritmia, usando-se apenas o sinal original.

Tabela G.3: Configurações das entradas na classificação de arritmias.

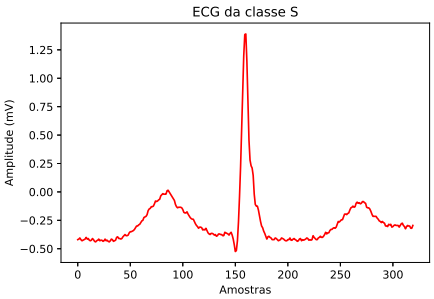
Configuração	Descrição da entrada	Tamanho
C1	Sinal original de ECG da 1ª derivação com 1 batimento e valores normalizados	320
C2	Sinal original de ECG da 1ª derivação com 1 batimento e sem normalizar os valores	320
C3	Sinal original de ECG da 1ª derivação com 3 batimentos e valores normalizados	960
C4	Sinal original de ECG da 1ª derivação com 3 batimentos e sem normalizar os valores	960
C5	Sinal original de ECG da 2ª derivação com 1 batimento e valores normalizados	320
C6	Sinal original de ECG da 2ª derivação com 1 batimento e sem normalizar os valores	320
C7	Sinal original de ECG da 2ª derivação com 3 batimentos e valores normalizados	960
C8	Sinal original de ECG da 2ª derivação com 3 batimentos e sem normalizar os valores	960

Alguns exemplos do sinal original, correspondentes às configurações C2 e C4 de cada classe, podem ser observados na Figura G.2.

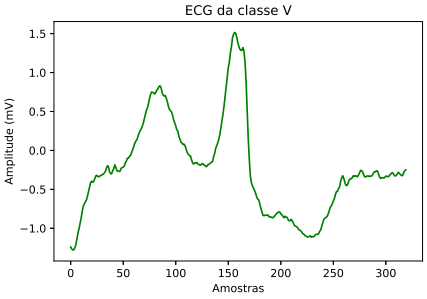
Em todas as configurações de entrada, a rede neural foi construída como indicado na Tabela G.4 e ilustrada na Figura G.3, tendo sido treinada com 1000 épocas.



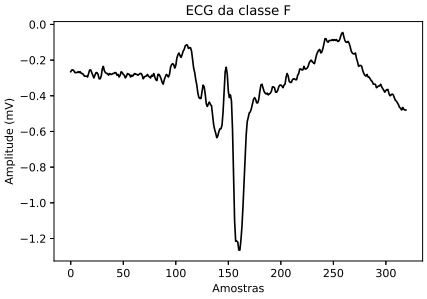
(a) C2 para Classe N



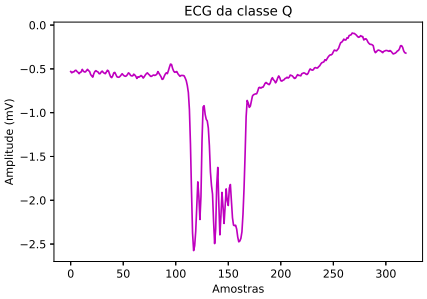
(b) C2 para Classe S



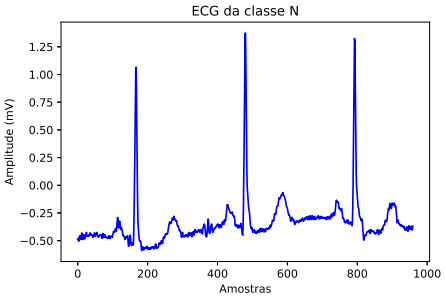
(c) C2 para Classe V



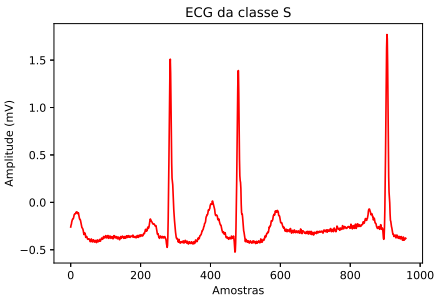
(d) C2 para Classe F



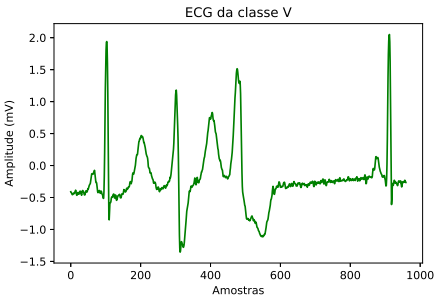
(e) C2 para Classe Q



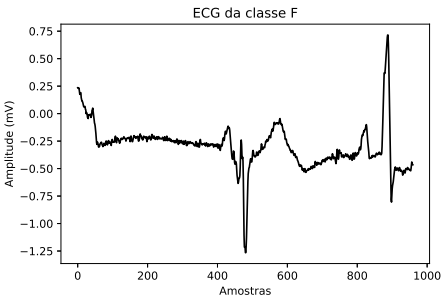
(f) C4 para Classe N



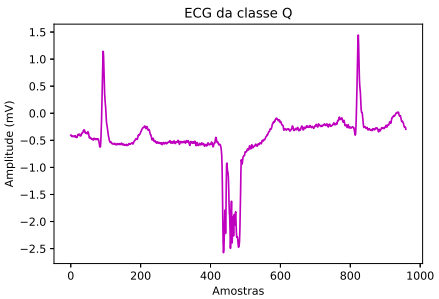
(g) C4 para Classe S



(h) C4 para Classe V



(i) C4 para Classe F



(j) C4 para Classe Q

Figura G.2: Entradas das configurações C2 e C4 da Tabela G.3.

Tabela G.4: Rede Neural utilizada para a classificação de arritmias.

	Caracterização da Rede II
Camada de entrada	Variável de acordo com a configuração de entrada
Camada oculta 1	64 neurônios
Camada oculta 2	64 neurônios
Camada de saída	5 neurônios
Função de ativação	ReLU (ocultas) Softmax (saída)
Função custo	Entropia cruzada
Passo de aprendizado	0,001
Tamanho do *mini* batch	2048
Dropout	0,25 (camada 1) 0,15 (camada 2)
Otimizador	Adam, com $\beta_1 = 0{,}9$ e $\beta_2 = 0{,}99$

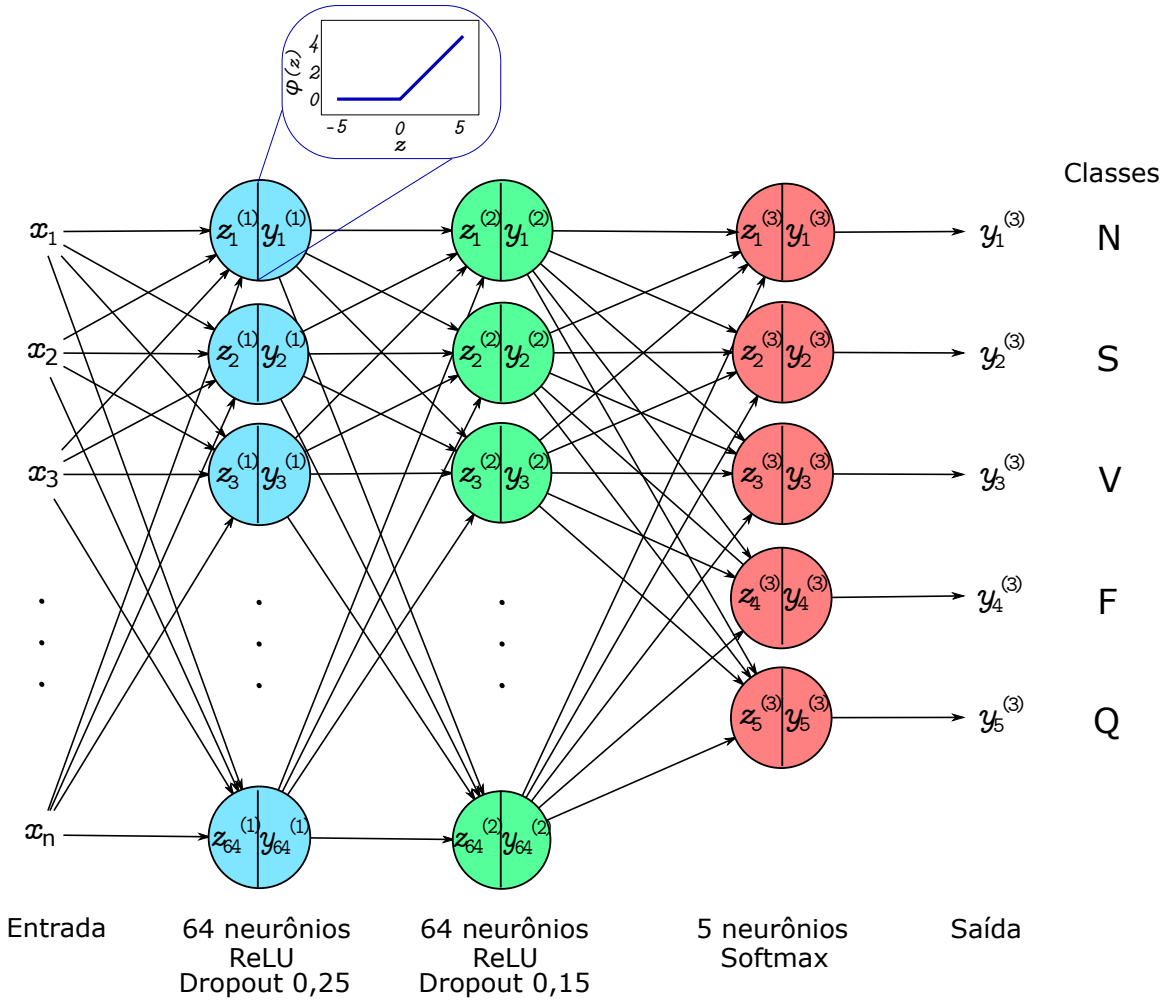


Figura G.3: Rede Neural MLP proposta.

Os resultados da Acurácia (Acc), da Sensibilidade (Se) e da Precisão (P) para cada classe, calculados segundo a recomendação da AAMI, são apresentados na Tabela G.5, em que i denota a configuração de entrada. O número de parâmetros da rede para cada i também é apresentado para comparação.

Tabela G.5: Resultados obtidos para cada entrada da Tabela G.3 e para a rede da Tabela G.4. Para cada métrica, os dois melhores resultados foram colocados em negrito.

i	Acc	N		S		V		F		Q		Param.
		Se	P	Se	P	Se	P	Se	P	Se	P	
C1	65,84	70,70	**88,98**	14,56	14,92	**91,10**	55,25	2,66	1,33	0	0	25029
C2	68,10	73,02	83,48	**23,81**	25,32	**88,94**	**71,99**	2,93	1,37	0	0	25029
C3	**76,97**	**89,02**	82,09	12,05	**47,75**	83,78	70,68	2,93	4,15	0	0	65989
C4	**76,67**	**82,74**	86,90	**51,55**	**63,08**	79,89	**70,99**	14,36	7,62	0	0	65989
C5	44,27	40,50	86,23	6,06	4,63	86,28	35,03	**24,84**	**10,07**	0	0	25029
C6	53,20	55,11	**90,69**	5,38	8,99	82,01	32,12	**23,67**	**12,13**	0	0	25029
C7	55,24	60,40	86,09	2,65	4,17	78,61	33,23	0	0	0	0	65989
C8	55,85	62,34	86,82	4,93	7,88	72,81	32,74	0,27	0,24	0	0	65989

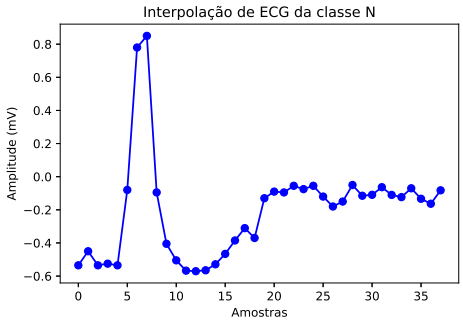
É possível notar que, apesar da primeira derivação fornecer resultados muito melhores de acurácia, sensibilidade e precisão das classes N, S e V, a segunda derivação apresentou melhores resultados que a primeira derivação para a classe F, quando foi utilizado apenas um batimento.

Usando-se a entrada que apresentou o melhor resultado, ou seja, a configuração C4, foram feitas combinações de C4 com C8 (ambas derivações com 3 batimentos e sem normalizar os valores), e de C4 com as informações dos intervalos RR, das durações dos segmentos QRS e da onda T, e da presença ou não da Onda P. Também foi feita uma entrada com a interpolação do batimento central de C4 e de C8 juntamente com as informações dos intervalos. Essas combinações estão listadas na Tabela G.6.

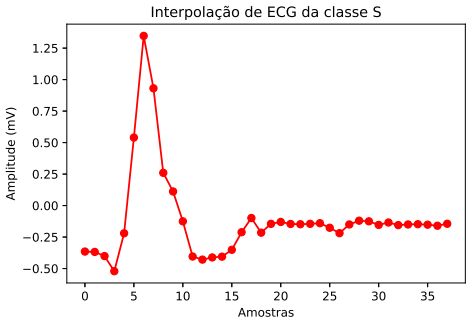
Tabela G.6: Configurações de combinações das entradas na classificação de arritmias.

Configuração	Descrição da entrada	Tamanho
D1	C4 e informações dos intervalos	967
D2	C4 e C8	1920
D3	Interpolação de um batimento das duas derivações e informações dos intervalos	45

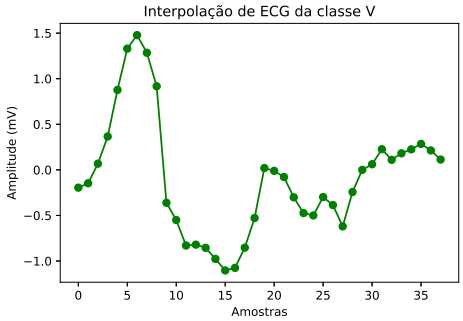
Alguns exemplos dos pontos obtidos pela interpolação feita na configuração D3 podem ser observados na Figura G.4. Os pontos da primeira e da segunda derivação estão colocados em série.



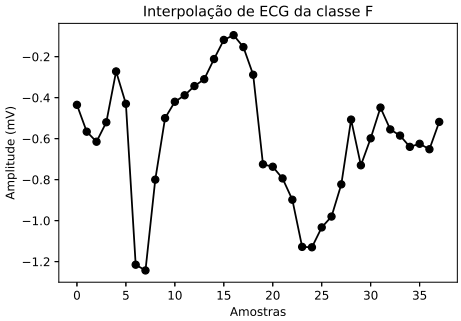
(a) D3 para Classe N



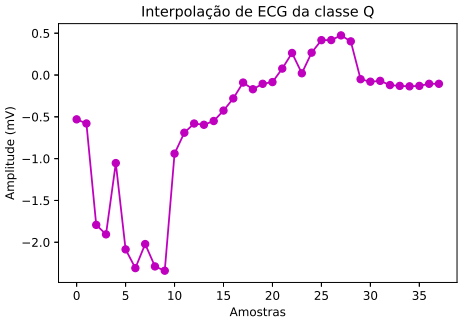
(b) D3 para Classe S



(c) D3 para Classe V



(d) D3 para Classe F



(e) D3 para Classe Q

Figura G.4: Entradas utilizadas na configuração D3 da Tabela G.6.

Na Tabela G.7 encontram-se os resultados da Acurácia, da Sensibilidade e da Precisão obtidos para as configurações da Tabela G.6.

Tabela G.7: Resultados obtidos para cada entrada da Tabela G.6 e para a rede da Tabela G.4. Para cada métrica, o melhor resultado foi colocado em negrito.

i	Acc	N		S		V		F		Q		Param.
		Se	P	Se	P	Se	P	Se	P	Se	P	
D1	**75,43**	**81,07**	86,16	26,99	51,22	**91,25**	**78,35**	36,44	12,38	0	0	66437
D2	72,40	75,86	84,77	**46,85**	**59,14**	87,76	68,21	3,46	1,40	0	0	127429
D3	71,15	74,25	**92,41**	35,94	33,91	80,73	72,11	**75,27**	**17,13**	0	0	7429

Comparando-se as Tabelas G.5 e G.7, nota-se que as informações dos intervalos RR e das durações das ondas na configuração D1 influenciam positivamente os resultados da classe F, mas

pioram os da classe S. Percebe-se também que a informação adicional da segunda derivação, no caso de D2, não afeta significativamente o desempenho da rede comparado com quando ela é alimentada apenas pela configuração C4.

Por fim, apesar do desempenho obtido com D3 ser menor do que o obtido com C4 para a classe S e para a acurácia geral, os resultados para as classes N, V e F foram equivalentes ou superiores. Deve ser levado em conta ainda que o número de parâmetros de D3 é muito menor do que o de C4, indicando um custo computacional menor, mas com resultados equivalentes. Portanto, a extração de características, como a interpolação sugerida por [24] é uma etapa relevante para a rede MLP.

A matriz de confusão do conjunto de treinamento para D3 está apresentada na Tabela G.8, e a matriz de confusão do conjunto de teste está indicada na Tabela G.9.

Tabela G.8: Matriz de confusão obtida para a configuração D3 no treinamento.

		Classes Preditas				
		N	S	V	F	Q
Classes Verdadeiras	N	8088	250	100	95	1
	S	19	734	1	1	0
	V	32	21	2415	70	0
	F	7	1	4	384	0
	Q	0	0	0	0	1

Tabela G.9: Matriz de confusão obtida para a configuração D3 no teste.

		Classes Preditas				
		N	S	V	F	Q
Classes Verdadeiras	N	6165	751	150	1237	0
	S	330	474	485	30	0
	V	120	171	1642	101	0
	F	56	2	35	283	0
	Q	0	1	3	1	0

Referências Bibliográficas

[1] U. R. Acharya *et al.*, "Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network," *Information Sciences*, vol. 405, pp. 81–90, 2017.

[2] U. R. Acharya *et al.*, "Automated identification of shockable and non-shockable life-threatening ventricular arrhythmias using convolutional neural network," *Future Generation Computer Systems*, vol. 79, pp. 952–959, 2018.

[3] S. Savalia and V. Emamian, "Cardiac arrhythmia classification by multi-layer perceptron and convolution neural networks," *Bioengineering*, vol. 5, no. 2, pp. 35, 2018.

[4] J. Huang *et al.*, "ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network," *IEEE Access*, vol. 7, pp. 92871–92880, 2019.

[5] T. Mahmud, S. A. Fattah, and M. Saquib, "DeepArrNet: An efficient deep CNN architecture for automatic arrhythmia detection and classification from denoised ECG beats," *IEEE Access*, vol. 8, pp. 104788–104800, 2020.

[6] A. Rajkumar, M. Ganesan, and R. Lavanya, "Arrhythmia classification on ECG using deep learning," in *2019 5th International Conference on Advanced Computing Communication Systems*, pp. 365–369, 2019.

[7] X. Xu and H. Liu, "ECG heartbeat classification using convolutional neural networks," *IEEE Access*, vol. 8, pp. 8614–8619, 2020.

[8] X. Zhai and C. Tin, "Automated ECG classification using dual heartbeat coupling based on convolutional neural network," *IEEE Access*, vol. 6, pp. 27465–27472, 2018.

[9] Ö. Yıldırım *et al.*, "Arrhythmia detection using deep convolutional neural network with long duration ECG signals," *Computers in biology and medicine*, vol. 102, pp. 411–420, 2018.

[10] F. Y. O. Abdalla *et al.*, "Deep convolutional neural network application to classify the ECG arrhythmia," *Signal, Image and Video Processing*, vol. 14, n. 7, p. 1431-1439, 2020.

[11] A. Ullah *et al.*, "Classification of arrhythmia by using deep learning with 2-D ECG spectral image representation," *Remote Sensing*, vol. 12, no. 10, pp. 1685, 2020.

[12] J. Zhang *et al.*, "MLBF-Net: A multi-lead-branch fusion network for multi-class arrhythmia classification using 12-lead ECG," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 9, pp. 1–11, 2021.

[13] X. Hua *et al.*, "A novel method for ECG signal classification via one-dimensional convolutional neural network," *Multimedia Systems*, pp. 1–13, 2020.

[14] S. Vijayarangan *et al.*, "Interpreting deep neural networks for single-lead ECG arrhythmia classification," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, pp. 300–303, 2020.

[15] H. Tung *et al.*, "Multi-lead ECG classification via an information-based attention convolutional neural network," *arXiv preprint arXiv:2003.12009*, 2020.

[16] V. Moskalenko, N. Zolotykh, and G. Osipov, "Deep learning for ECG segmentation," in *International Conference on Neuroinformatics*, Springer, pp. 246–254, 2019.

[17] A. Y. Ng, K. Katanforoosh, and Y. B. Mourri, "Neural networks and deep learning," deeplearning.ai. Disponível em: https://www.coursera.org/learn/neural-networks-deep-learning. Acesso em: 21 jan. 2020.

[18] S. Haykin, *Neural networks and learning machines*, vol. 3, Pearson Upper Saddle River, NJ, USA, 2009.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016. Disponível em: http://www.deeplearningbook.org. Acesso em: 15 fev. 2020.

[20] C. M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000 (June 13), Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

[23] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[24] P. de Chazal, M. O'Dwyer, and R. B. Reilly, "Automatic classification of heartbeats using ECG morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 7, pp. 1196–1206, 2004.

[25] E. J. S. Luz *et al.*, "ECG-based heartbeat classification for arrhythmia detection: A survey," *Computer methods and programs in biomedicine*, vol. 127, pp. 144–164, 2016.

[26] E. Luz and D. Menotti, "How the choice of samples for building arrhythmia classifiers impact their performances," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4988–4991, 2011.

[27] L. B. Marinho *et al.*, Rebouças F., and V. H. C. de Albuquerque, "A novel electrocardiogram feature extraction approach for cardiac arrhythmia classification," *Future Generation Computer Systems*, vol. 97, pp. 564–577, 2019.

[28] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice Hall, 1998.

[29] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[30] Y. LeCun *et al.*, "Mnist handwritten digit database," vol. 7, pp. 23, 2010. Disponível em: http://yann. lecun. com/exdb/mnist. Acesso em: 21 fev. 2020.

[31] Martín Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems." Software disponível em tensorflow.org, 2015.

[32] M. L. Soria and J. P. Martínez, "Analysis of multidomain features for ECG classification," in *2009 36th Annual Computers in Cardiology Conference*, pp. 561–564, 2009.

[33] M. Llamedo and J. P. Martínez, "Heartbeat classification using feature selection driven by database generalization criteria," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 3, pp. 616–625, 2011.

[34] T. Mar *et al.*, "Optimization of ECG classification by means of feature selection," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 8, pp. 2168–2177, 2011.

[35] C. Lin and C. Yang, "Heartbeat classification using normalized RR intervals and morphological features," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[36] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, John Wiley & Sons, second edition, 2001.

[37] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media, 2009.

[38] R. A. Fisher, "The statistical utilization of multiple measurements," *Annals of eugenics*, vol. 8, no. 4, pp. 376–386, 1938.

[39] F. Pedregosa et. al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[41] S. Saadatnejad, M. Oveisi, and M. Hashemi, "LSTM-based ECG classification for continuous monitoring on personal wearable devices," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 2, pp. 515–523, 2020.

[42] R. Banerjee, A. Ghose, and S. Khandelwal, "A novel recurrent neural network architecture for classification of atrial fibrillation using single-lead ECG," in *Proc. of 27th European Signal Processing Conference*, pp. 1–5, 2019.

[43] Shraddha Singh *et al.*, "Classification of ECG arrhythmia using recurrent neural networks," *Procedia computer science*, vol. 132, pp. 1290–1297, 2018.

[44] C. Zhang *et al.*, "Patient-specific ECG classification based on recurrent neural networks and clustering technique," in *2017 13th IASTED International Conference on Biomedical Engineering*, pp. 63–67, 2017.

[45] G. Wang *et al.*, "A global and updatable ECG beat classification system based on recurrent neural networks and active learning," *Information Sciences*, vol. 501, pp. 523–542, 2019.

[46] J. V. Zaen *et al.*, "Classification of cardiac arrhythmias from single lead ECG with a convolutional recurrent neural network," *arXiv preprint arXiv:1907.01513*, 2019.

[47] G. Garcia *et al.*, "Improving automatic cardiac arrhythmia classification: Joining temporal-VCG, complex networks and SVM classifier," in *2016 International Joint Conference on Neural Networks*, pp. 3896–3900, 2016.

[48] G. Garcia *et al.*, "Inter-patient ECG heartbeat classification with temporal VCG optimized by PSO," *Scientific Reports*, vol. 7, no. 1, pp. 1–11, 2017.

[49] R. M. Rangayyan, *Biomedical signal analysis*, vol. 33, John Wiley & Sons, 2015.

[50] A. Noordergraaf, *Circulatory system dynamics*, vol. 1, Elsevier, 2012.

[51] G. D. Clifford *et al.*, *Advanced methods and tools for ECG data analysis*, Artech house Boston, 2006.

[52] D. B. Geselowitz, "On the theory of the electrocardiogram," *Proceedings of the IEEE*, vol. 77, no. 6, pp. 857–876, 1989.

[53] K. E. Barret, S. Boitano, and S. M. Barman, *Ganong's review of medical physiology*, 2012.

[54] J. Venegas and R. Mark, "Quantitative physiology: Organ transport systems," Open Courseware, 2004. Disponível em https://ocw.mit.edu/courses/health-sciences-and-technology/hst-542j-quantitative-physiology-organ-transport-systems-spring-2004/readings/. Acesso em: 9 set. 2020.

[55] P. E. McSharry et al., "ECGSYN - A realistic ECG waveform generator (version 1.0.0)," 2003. Disponível em https://physionet.org/content/ecgsyn/1.0.0/. Acesso em: 5 mar. 2020.

[56] P. E. et al. McSharry, "A dynamical model for generating synthetic electrocardiogram signals," *IEEE transactions on biomedical engineering*, vol. 50, no. 3, pp. 289–294, 2003.

[57] S. K. Berkaya *et al.*, "A survey on ECG analysis," *Biomedical Signal Processing and Control*, vol. 43, pp. 216–235, 2018.

[58] P. Pławiak, "Novel methodology of cardiac health recognition based on ECG signals and evolutionary-neural system," *Expert Systems with Applications*, vol. 92, pp. 334 – 349, 2018.

[59] J. H. Abawajy, A. V. Kelarev, and M. Chowdhury, "Multistage approach for clustering and classification of ECG data," *Computer Methods and Programs in Biomedicine*, vol. 112, no. 3, pp. 720 – 730, 2013.

[60] K. Padmavathi and K. S. Ramakrishna, "Classification of ECG signal during atrial fibrillation using autoregressive modeling," *Procedia Computer Science*, vol. 46, pp. 53 – 59, 2015.

[61] I. Romero and L. Serrano, "ECG frequency domain features extraction: a new characteristic for arrhythmias classification," in *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2, pp. 2006–2008 vol.2, 2001.

[62] M. H. Vafaie, M. Ataei, and H. R. Koofigar, "Heart diseases prediction based on ECG signals' classification using a genetic-fuzzy system and dynamical model of ECG signals," *Biomedical Signal Processing and Control*, vol. 14, pp. 291 – 296, 2014.

[63] E. A. P. Alday *et al.*, "Classification of 12-lead ECGs: the PhysioNet - Computing in Cardiology Challenge 2020 (version 1.0.1)," *PhysioNet*, 2020. Disponível em: https://doi.org/10.13026/f4ab-0814. Acesso em: 21 jun. 2020.

[64] E. A. P. Alday *et al.*, "Classification of 12-lead ECGs: the Physionet/Computing in Cardiology Challenge 2020," *Physiological measurement*, vol. 41, no. 12, pp. 124003, 2020.

[65] S. I. Niwas, R. S. S. Kumari, and V. Sadasivam, "Artificial neural network based automatic cardiac abnormalities classification," in *Sixth International Conference on Computational Intelligence and Multimedia Applications*, pp. 41–46, 2005.

[66] G. B. Moody and R. G. Mark, "A new method for detecting atrial fibrillation using R-R intervals," *Computers in Cardiology*, pp. 227–230, 1983.

[67] Y. Hagiwara *et al.*, "Computer-aided diagnosis of atrial fibrillation based on ECG signals: A review," *Information Sciences*, vol. 467, pp. 99–114, 2018.

[68] O. Faust *et al.*, "Automated detection of atrial fibrillation using long short-term memory network with RR interval signals," *Computers in biology and medicine*, vol. 102, pp. 327–335, 2018.

[69] R. J. Martis *et al.*, "Application of higher order statistics for atrial arrhythmia classification," *Biomedical signal processing and control*, vol. 8, no. 6, pp. 888–900, 2013.

[70] F. M. Nolle *et al.*, "CREI-GARD, a new concept in computerized arrhythmia monitoring systems," *Computers in Cardiology*, vol. 13, pp. 515–518, 1986.

[71] Y. Xu *et al.*, "Detection of ventricular tachycardia and fibrillation using adaptive variational mode decomposition and boosted-CART classifier," *Biomedical Signal Processing and Control*, vol. 39, pp. 219–229, 2018.

[72] D. Dua and C. Graff, "UCI machine learning repository," 2017. Disponível em: http://archive.ics.uci.edu/m. Acesso em: 12 abr. 2020.

[73] S. M. Jadhav, S. L. Nalbalwar, and A. A. Ghatol, "Arrhythmia disease classification using artificial neural network model," in *2010 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–4, 2010.

[74] S. M. Jadhav, S. L. Nalbalwar, and A. A. Ghatol, "ECG arrhythmia classification using modular neural network model," in *2010 IEEE EMBS conference on biomedical engineering and sciences*, pp. 62–66, 2010.

[75] S. M. Jadhav, S. L. Nalbalwar, and A. A. Ghatol, "Artificial neural network based cardiac arrhythmia classification using ECG signal data," in *2010 International Conference on Electronics and Information Engineering*, vol. 1, pp. V1–228–V1–231, 2010.

[76] R. D. Raut and S. V. Dudul, "Arrhythmias classification with MLP neural network and statistical analysis," in *2008 First International Conference on Emerging Trends in Engineering and Technology*, pp. 553–558, 2008.

[77] P. Warrick and M. N. Homsi, "Cardiac arrhythmia detection from ECG combining convolutional and long short-term memory networks," in *2017 Computing in Cardiology*, pp. 1–4, 2017.

[78] F. Liu *et al.*, "A LSTM and CNN based assemble neural network framework for arrhythmias classification," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1303–1307, 2019.

[79] J. S. Arteaga-Falconi, H. Al Osman, and A. El Saddik, "ECG authentication for mobile devices," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 591–600, 2016.

[80] I. Odinaka *et al.*, "Cardiovascular biometrics: Combining mechanical and electrical signals," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 16–27, 2015.

[81] S. Gutta and Q. Cheng, "Joint feature extraction and classifier design for ECG-based biometric recognition," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 2, pp. 460–468, 2016.

[82] L. Biel *et al.*, "ECG analysis: a new approach in human identification," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 3, pp. 808–812, 2001.

[83] S. Mousavi and F. Afghah, "Inter- and intra- patient ECG heartbeat classification for arrhythmia detection: A sequence to sequence deep learning approach," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1308–1312, 2019.

[84] Association for the Advancement of Medical Instrumentation et al., "ANSI/AAMI EC57:1998/(R)2008 - Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms," *American National Standards Institute, Arlington, VA, USA*, 2008, Association for the Advancement of Medical Instrumentation (AAMI), ANSI/AAMI/ISO EC57,1998-(R)2008, 2008.

[85] N. K. Dewangan and S. P. Shukla, "ECG arrhythmia classification using discrete wavelet transform and artificial neural network," in *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*, pp. 1892–1896, 2016.

[86] O. Sayadi and M. B. Shamsollahi, "Multiadaptive bionic wavelet transform: Application to ECG denoising and baseline wandering reduction," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–11, 2007.

[87] O. Sayadi and M. B. Shamsollahi, "ECG denoising and compression using a modified extended Kalman filter structure," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 9, pp. 2240–2248, 2008.

[88] N. V. Thakor and Y. S. Zhu, "Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection," *IEEE Transactions on Biomedical Engineering*, vol. 38, no. 8, pp. 785–794, 1991.

[89] P. A. Lynn, "Recursive digital filters for biological signals," *Medical & biological engineering*, vol. 9, no. 1, pp. 37–43, 1971.

[90] Q. Xue, Y. H. Hu, and W. J. Tompkins, "Neural-network-based adaptive matched filtering for QRS detection," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 4, pp. 317–329, 1992.

[91] B. N. Singh and A. K. Tiwari, "Optimal selection of wavelet basis function applied to ECG signal denoising," *Digital signal processing*, vol. 16, no. 3, pp. 275–287, 2006.

[92] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, 1985.

[93] A. V. Oppenheim, *Discrete-time signal processing*, Pearson Education India, 1999.

[94] MIT Laboratory for Computational Physiology, "Python waveform-database (WFDB) (version 2.2.0)," 2018. Disponível em: https://github.com/MIT-LCP/wfdb-python. Acesso em: 13 fev. 2020.

[95] C. Li, C. Zheng, and C. Tai, "Detection of ECG characteristic points using wavelet transforms," *IEEE Transactions on biomedical Engineering*, vol. 42, no. 1, pp. 21–28, 1995.

[96] M. Bahoura, M. Hassani, and M. Hubin, "DSP implementation of wavelet transform for real time ECG wave forms detection and heart rate analysis," *Computer methods and programs in biomedicine*, vol. 52, no. 1, pp. 35–44, 1997.

[97] J. P. Martínez *et al.*, "A wavelet based ECG delineator: evaluation on standard databases," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 4, pp. 570–581, 2004.

[98] A. Demski and M. L. Soria, "ecg-kit: a Matlab toolbox for cardiovascular signal processing," *Journal of open research software*, vol. 4, no. 1, 2016.

[99] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE signal processing magazine*, vol. 8, no. 4, pp. 14–38, 1991.

[100] PhysioNet, "ECGPUWAVE (version 1.3.4)," 2018. Disponível em: https://physionet.org/content/ecgpuwave/1.3.4/. Acesso em: 8 jun. 2020.

[101] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[102] U. R. Acharya *et al.*, "A deep convolutional neural network model to classify heartbeats," *Computers in biology and medicine*, vol. 89, pp. 389–396, 2017.

[103] K. Ochiai, S. Takahashi, and Y. Fukazawa, "Arrhythmia detection from 2-lead ECG using convolutional denoising autoencoders," in *Proc. KDD*, pp. 1–7, 2018.

[104] H. Huang *et al.*, "A new hierarchical method for inter-patient heartbeat classification using random projections and RR intervals," *Biomedical engineering online*, vol. 13, no. 1, pp. 90, 2014.

[105] S. Y. Şwn and N. Özkurt, "ECG arrhythmia classification by using convolutional neural network and spectrogram," in *2019 Innovations in Intelligent Systems and Applications Conference*, pp. 1–6, 2019.

[106] M. Salem, S. Taheri, and J. Yuan, "ECG arrhythmia classification using transfer learning from 2-dimensional deep CNN features," in *2018 IEEE Biomedical Circuits and Systems Conference*, pp. 1–4, 2018.

[107] S. Osowski, T. Markiewicz, and L. T. Hoai, "Recognition and classification system of arrhythmia using ensemble of neural networks," *Measurement*, vol. 41, no. 6, pp. 610–617, 2008.

[108] C. Ye, M. T. Coimbra, and B. V. K. V. Kumar, "Arrhythmia detection and classification using morphological and dynamic features of ECG signals," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pp. 1918–1921, 2010.

[109] S.-N. Yu and K.-T. Chou, "Integration of independent component analysis and neural networks for ECG beat classification," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2841-2846, 2008.

[110] S.-N. Yu and Y.-H. Chen, "Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1142-1150, 2007.

[111] İ. Güler and E. D. Übeylı, "ECG beat classifier designed by combined neural network model," *Pattern recognition*, vol. 38, no. 2, pp. 199–208, 2005.

[112] M. H. Song *et al.*, "Support vector machine based arrhythmia classification using reduced features," *International Journal of Control, Automation, and Systems*, vol. 3, no. 4, pp. 571–579, 2005.

[113] G. A. Lizarzaburu, M. T. M. Silva, and R. Candido, "Restauração de imagens com técnicas de aprendizado de máquina," Tech. Rep., Fundação de Amparo à Pesquisa do Estado de São Paulo, 2020.

[114] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, 2011.

[115] M. Kukar *et al.*, "Cost-sensitive learning with neural networks," in *ECAI*, vol. 98, pp. 445–449, 1998.

[116] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020.

[117] N. V. Chawla *et al.*, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[118] S. Kiranyaz *et al.*, "1D convolutional neural networks and applications: A survey," *arXiv preprint arXiv:1905.03554*, 2019.

[119] U. R. Acharya *et al.*, "Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network," *Knowledge-Based Systems*, vol. 132, pp. 62–71, 2017.

[120] A. Y. Hannun *et al.*, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Medicine*, vol. 25, pp. 65–69, 2019.

[121] Y. Zheng *et al.*, "Time series classification using multi-channels deep convolutional neural networks," in *International Conference on Web-Age Information Management*. Springer, pp. 298–310, 2014.

[122] E. Del Moral Hernández and M. T. M. Silva, "Recurrent neural networks," in *Computational Intelligence*, H. Ishibuchi, Ed., vol. 1, pp. 97–127. UNESCO/Eolss Publishers Co. Ltd., London, 2015.

[123] D. Mandic and J. Chambers, *Recurrent neural networks for prediction: learning algorithms, architectures and stability*, Wiley, 2001.

[124] H. T. Siegelmann and E. D. Sontag, "Turing computability with neural nets," *Applied Mathematics Letters*, vol. 4, no. 6, pp. 77–80, 1991.

[125] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[126] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, 2013.

[127] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *arXiv preprint arXiv:1409.3215*, 2014.

[128] A. Graves, "Offline arabic handwriting recognition with multidimensional recurrent neural networks," in *Guide to OCR for Arabic scripts*, Springer, pp. 297–313, 2012.

[129] J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," *Advances in neural information processing systems*, vol. 17, pp. 1569–1576, 2004.

[130] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis-a brief tutorial," in *Institute for Signal and information Processing*, vol. 18, pp. 1–8, 1998.

[131] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[132] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[133] H. Hostelling, "Analysis of a complex statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[134] I. T. Jolliffe, *Principal component analysis*, Springer, second edition, 2002.

[135] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge University Press, 1991.

Anexo - Trabalhos de Simpósios Nacionais

Resumo apresentado no 28º SIICUSP

CLASSIFICAÇÃO DE ARRITMIAS UTILIZANDO REDES NEURAIS PERCEPTRON MULTICAMADA

Natália Nagata, Renato Candido e Magno T. M. Silva
Escola Politécnica, USP, SP
nagata.natalia@usp.br, renatocan@lps.usp.br, magno.silva@usp.br

Objetivos

Este trabalho apresenta os resultados obtidos para a classificação de arritmias cardíacas usando uma rede neural perceptron multicamada (*multilayer perceptron* – MLP). As arritmias foram divididas em cinco classes, estabelecidas pela AAMI (*Association for the Advancement of Medical Instrumentation*) [1] . Foram considerados diferentes tipos de entrada da rede e estudou-se a influência das características do sinal de eletrocardiograma (ECG) no desempenho.

Métodos e Procedimentos

As redes MLP foram treinadas com 44 gravações de 30 min. do sinal de ECG de duas derivações, obtidas do banco de dados *MIT-BIH Arrhythmia Database* [2,3]. Foram testadas sete entradas diferentes, baseadas na combinação de três tipos de informação: (i) os valores (em mV) do sinal de ECG; (ii) as características dos intervalos RR e das durações do complexo QRS e da onda T, informação da existência ou não da onda P; e (iii) os valores da interpolação do sinal de ECG sugerida por [4].

Na camada de saída, foram considerados cinco neurônios com a função Softmax, cada um para uma das cinco classes possíveis: batimentos do nó SA (N), supraventriculares ectópicos (S), ventriculares ectópicos (V), fusão de batimentos normais e ventriculares ectópicos (F) e desconhecidos ou de marca-passo (Q). Consideraram-se duas camadas ocultas, ambas com 64 neurônios e com funções de ativação ReLU.

Para trabalhar com as classes desbalanceadas usou-se a função custo de entropia cruzada categórica com pesos calculados na proporção inversa do número de dados de cada classe. Consideraram-se o algoritmo de otimização Adam com $\beta_1 = 0{,}9$ e $\beta_2 = 0{,}99$, o algoritmo de retropropagação (*backpropagation*) com passo $\eta = 0{,}001$, 1000 épocas para o treinamento e *mini-batches* de tamanho $k = 2048$.

Os dados foram separados nos conjuntos de teste e de treinamento pela divisão entre os pacientes, como sugerido pela AAMI [1]. Isso evita resultados não confiáveis favorecidos pelo viés de um mesmo paciente sendo usado em ambos conjuntos.

Resultados

Na Tabela 1, encontram-se os resultados da Acurácia (Acc), da Sensibilidade (Se), da Precisão (P) e o número de parâmetros (Param.), obtidos para as diversas entradas. As configurações usadas nas entradas foram de um e três batimentos da primeira derivação (i=1,2) e da segunda derivação (i=3,4), de 3 batimentos das duas derivações (i=5), de 3 batimentos da primeira derivação junto às informações dos intervalos (i=6) e da interpolação de um batimento das duas derivações junto às informações dos intervalos (i=7).Devido à ausência do padrão Q, essa classificação não foi considerada na tabela.

Tabela 1: Resultados obtidos para cada entrada.

i	Acc	N		S		V		F		Param.
		Se	P	Se	P	Se	P	Se	P	
1	68,10	73,02	83,48	23,81	25,32	88,94	71,99	2,93	1,37	25029
2	76,67	82,74	86,90	51,55	63,08	79,89	70,99	14,36	7,62	65989
3	53,20	55,11	90,69	5,38	8,99	82,01	32,12	23,67	12,13	25029
4	55,85	62,34	86,82	4,93	7,88	72,81	32,74	0,27	0,24	65989
5	72,40	75,86	84,77	46,85	59,14	87,76	68,21	3,46	1,40	127429
6	75,43	81,07	86,16	26,99	51,22	91,25	78,35	36,44	12,38	66437
7	71,15	74,25	92,41	35,94	33,91	80,73	72,11	75,27	17,13	7429

Conclusões

Os resultados de classificação de arritmias da Tabela 1 estão dentro da média dos valores relatados na literatura, seguindo a divisão por pacientes nos conjuntos de treinamento e teste. Vale ressaltar que quando essa divisão não é considerada, a rede consegue atingir resultados de classificação muito melhores. No entanto, isso não é realista sob uma perspectiva clínica. Além disso, nota-se que a extração de características para a rede MLP é uma etapa importante, uma vez que permite alcançar bons resultados de classificação com menos parâmetros. Em um trabalho futuro, pretende-se considerar outros tipos redes, como a rede neural convolucional (*convolutional neural network* - CNN) e a recorrente (*recurrent neural network* - RNN) para esse problema.

Referências Bibliográficas

[1] *ANSI/AAMI EC57:2012 - Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms*, 2013.

[2] G.B. Moody and R.G. Mark, *The impact of the MIT-BIH Arrhythmia Database*, IEEE Eng in Med and Biol 20(3):45-50 (May-June 2001).

[3] A. Goldberger *et al.*, *PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals*, Circulation [Online]. 101 (23), pp. e215–e220, 2000.

[4] P. Chazal, M. O'Dwyer, and R. B. Reilly, *Automatic classification of heartbeats using ECG morphology and heartbeat interval features*, IEEE Trans. Biomedical Eng., v. 51, n. 7, p. 1196-1206, 2004.

Trabalho submetido a 73ª Reunião da SBPC

3.13.99 – Engenharia Biomédica.

CLASSIFICAÇÃO DE ARRITMIAS CARDÍACAS COM REDES NEURAIS PERCEPTRON MULTICAMADA

Natália Nagata[1], Renato Candido[2] , Magno T. M. Silva[3]
1. Estudante de Engenharia Elétrica da Escola Politécnica da Universidade de São Paulo (EPUSP)
2. Pesquisador Associado da EPUSP - Departamento de Engenharia de Sistemas Eletrônicos/Coorientador
3. Professor da EPUSP - Departamento de Engenharia de Sistemas Eletrônicos/Orientador

Resumo

O desenvolvimento de sistemas de diagnóstico automatizado a partir do sinal de eletrocardiograma (ECG) é um assunto de interesse desde a década de 1980. Recentemente, com o surgimento de novas técnicas de aprendizado de máquina, a pesquisa em automatização do diagnóstico de arritmias cardíacas atraiu novamente a atenção da comunidade científica. Neste trabalho, utilizam-se redes neurais do tipo perceptron multicamada (*multilayer perceptron* - MLP) para classificação automática de arritmias, considerando a abordagem mais realista de separação dos dados dos pacientes durante as fases de treinamento e teste. Os objetivos da pesquisa são: (i) encontrar as melhores estruturas de redes MLP para o problema em termos de métricas de classificação, (ii) estudar o efeito da entrada e da extração de características no desempenho da rede e (iii) realizar uma análise comparativa dos resultados obtidos com os da literatura.

Palavras-chave: sinal de ECG; diagnóstico automático; cardiologia.

Apoio financeiro: FAPESP (2019/26911-6 e 2017/20378-9)

Trabalho selecionado para a JNIC: Pró-Reitoria de Pesquisa da USP

Introdução

Segundo a Organização Mundial da Saúde (OMS), as doenças cardiovasculares são a principal causa de morte no mundo e as arritmias cardíacas são algumas das doenças cardiovasculares mais comuns [1]. As arritmias correspondem a qualquer distúrbio na taxa, na regularidade e nos locais de origem ou condução dos impulsos elétricos cardíacos [2]. O diagnóstico das arritmias é feito pela análise do sinal de eletrocardiograma (ECG), que registra a atividade elétrica do coração pelos potenciais superficiais do corpo ao longo do tempo.

A análise manual do ECG demanda muito tempo do especialista e é dificultada pelas características morfológicas variáveis do sinal. Algumas arritmias aparecem raramente e pode ser necessário gravar até uma semana de atividade do ECG, o que impossibilita a obtenção de resultados imediatos [3]. Assim, muitos métodos computadorizados para detecção automática têm sido propostos na literatura [2, 4]. Devido às altas taxas de erro desses métodos e ao crescimento da área de aprendizado de máquina [5], a pesquisa em automatização do diagnóstico de arritmias cardíacas ressurgiu [3, 6-9].

Recentemente, soluções baseadas em redes neurais, como redes perceptron multicamada (MLP) [3], convolucionais [6, 7] e recorrentes [8, 9] têm sido propostas na literatura. Em vários desses trabalhos, os autores utilizam dados dos mesmos pacientes tanto no conjunto de teste, quanto no de treinamento. Essa abordagem não é clinicamente realista, já que na prática o sistema treinado será utilizado em pacientes cujos dados não foram usados no treinamento. Neste artigo, propõe-se o uso de redes neurais do tipo MLP para detecção e classificação de arritmias cardíacas, usando a separação dos pacientes nas fases de treinamento e teste do banco de dados *MIT-BIH Arrhythmia Database* (MITDB) [10, 11]. Os objetivos do trabalho são: (i) propor melhores estruturas de redes MLP como classificadores em termos de métricas de desempenho, (ii) identificar a influência das características usadas como entrada e (iii) realizar uma comparação com a literatura.

Metodologia

O MITDB é composto por 48 gravações ambulatoriais de duas derivações de 30 minutos de pacientes do *Boston's Beth Israel Hospital*, com anotações manuais de cada batimento feitas por cardiologistas. Como o sinal de ECG é característico de cada indivíduo e de sua condição física, para não prejudicar a generalização dos métodos de automatização de diagnóstico, a *Association for the Advancement of Medical Instrumentation* (AAMI) [12] recomenda a divisão das gravações de modo que batimentos de um mesmo paciente não sejam simultaneamente usados nos conjuntos de treinamento e de teste. Apesar disso, poucos pesquisadores seguem as recomendações da AAMI, originando resultados favorecidos e não realistas, o que dificulta a verificação dos méritos relativos aos diferentes algoritmos. Neste trabalho, seguiu-se a divisão por pacientes proposta por De Chazal et al. [13]. Além disso, os dados foram classificados entre as cinco classes sugeridas pela AAMI: batimentos do nó sinoatrial (N), supraventriculares ectópicos (S), ventriculares ectópicos (V), fusão de batimentos normais e ventriculares ectópicos (F) e desconhecidos ou de marca-passo (Q). A implementação do classificador foi dividida em quatro etapas conforme esquematizado na Figura 1.

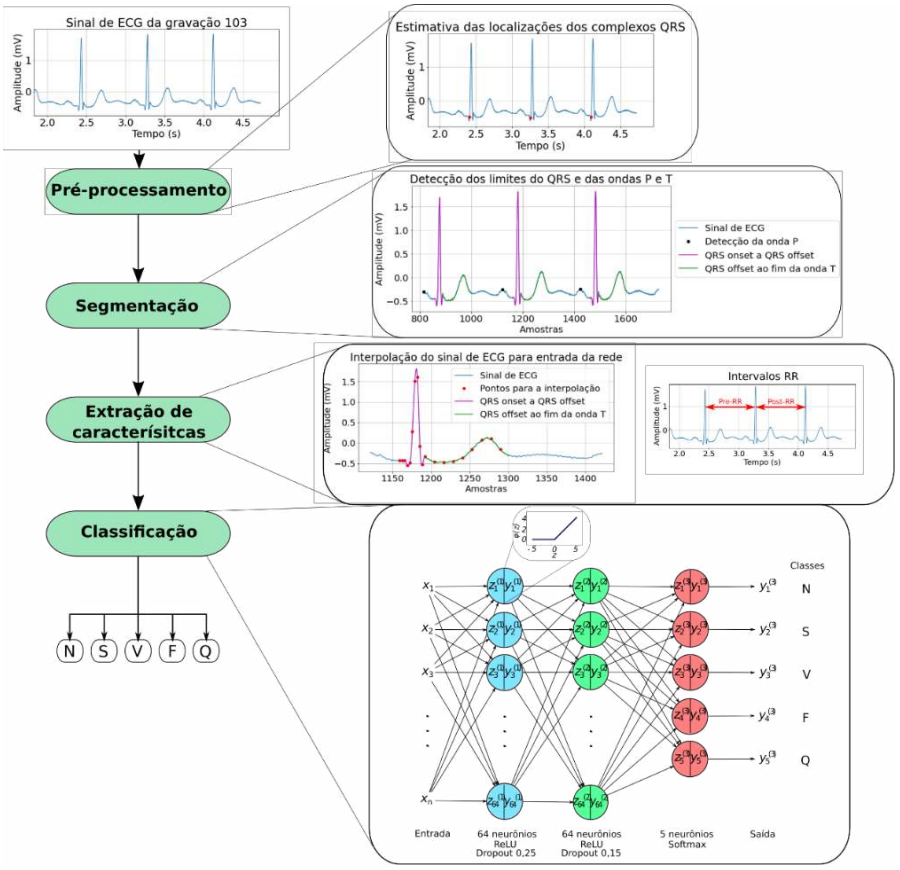


Figura 1 – Diagrama das quatro etapas de técnicas computadorizadas para análise de ECG.

A etapa de pré-processamento reduz os ruídos e artefatos provenientes de várias origens, como a rede elétrica. A segmentação delimita o complexo QRS e as ondas P e T. Para isso, usou-se o método proposto em [14], baseado em Transformada de Wavelet e implementado em MATLAB pelo pacote ecg-kit [15]. Já a etapa de extração de características determina a menor quantidade de características do sinal de ECG que permite taxas de classificação aceitáveis. Um exemplo de extração foi proposto por [13], em que são tomadas 10 amostras do complexo QRS, 9 amostras entre o fim do QRS e o fim da Onda T, informações dos intervalos RR, da duração do QRS e da Onda T e a presença da Onda P. Além dessas características, considera-se aqui o uso de um ou três batimentos do sinal original como entrada.

Na etapa de classificação, considerou-se uma MLP de duas camadas ocultas, ambas com 64 neurônios, funções de ativação ReLU, e *dropout* de 0,15 e 0,25 respectivamente. Na camada de saída, foram utilizados 5 neurônios com função Softmax. Para trabalhar com as classes desbalanceadas, usou-se a função custo de entropia cruzada categórica com pesos calculados na proporção inversa do número de dados de cada classe. Considerou-se ainda o algoritmo de otimização Adam com $\beta_1 = 0{,}9$ e $\beta_2 = 0{,}99$, e o algoritmo de retropropagação (*backpropagation*) com passo de aprendizado $\eta = 0{,}001$, 1000 épocas para o treinamento e *mini-batches* de tamanho $k = 2048$. Todas as etapas, com exceção da extração de características, foram implementadas em Python e as redes foram implementadas usando as bibliotecas Tensorflow e Keras [16].

Resultados e Discussão

As configurações usadas nas entradas foram de um ou três batimentos do sinal original da primeira derivação (A1, A2) e da segunda derivação (B1, B2), de três batimentos das duas derivações (AB), de três batimentos da primeira derivação junto às informações dos intervalos RR (A3) e de características extraídas como em [13] a partir das duas derivações (C). Calcularam-se as métricas de Acurácia geral (Acc),

$$Acc = \frac{VP + VN}{VP + VN + FP + FN},$$

e de Sensibilidade (Se) e Precisão (P) para cada classe,

$$Se = \frac{VP}{VP+FN} \text{ e } P = \frac{VP}{VP+FP},$$

em que VP, VN, FP e FN são, respectivamente, as quantidades de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos. Seguindo a recomendação da AAMI [12, 13], os falsos positivos devido à classe Q foram desconsiderados para a classe S e os falsos positivos devido às classes F e Q foram desconsiderados para a classe V.

Os resultados e o número de parâmetros para cada configuração estão apresentados na Tabela 1, junto aos resultados de [13], considerado como estado da arte. O número de parâmetros da solução de [13] não foi especificado e, portanto, não foi considerado na tabela. Devido à ausência do padrão Q, tanto na literatura quanto no presente trabalho, essa classificação não foi considerada na tabela.

Tabela 1 – Acurácia da rede (Acc), Sensibilidade (Se) e Precisão (P) de cada classe considerando diferentes entradas. Os maiores valores de cada métrica estão em negrito.

i	Acc	N		S		V		F		Número de Parâmetros
		Se (%)	P (%)	Se (%)	P (%)	Se (%)	P (%)	Se (%)	P (%)	
A1	68,10	73,02	83,48	23,81	25,32	88,94	71,99	2,93	1,37	25029
A2	**76,67**	**82,74**	86,90	**51,55**	**63,08**	79,89	70,99	14,36	7,62	65989
B1	53,20	55,11	90,69	5,38	8,99	82,01	32,12	23,67	12,13	25029
B2	55,85	62,34	86,82	4,93	7,88	72,81	32,74	0,27	0,24	65989
AB	72,40	75,86	84,77	46,85	59,14	87,76	68,21	3,46	1,40	127429
A3	75,43	81,07	86,16	26,99	51,22	**91,25**	**78,35**	36,44	12,38	66437
C	71,15	74,25	**92,41**	35,94	33,91	80,73	72,11	**75,27**	**17,13**	7429
[13]	85,9	86,9	99,2	75,9	38,5	77,7	81,9	89,4	8,6	–

Para a primeira derivação, as métricas da configuração A2 são significativamente maiores do que as da A1, com um aumento da Acurácia, da Sensibilidade e da Precisão das classes N, S e F. Observou-se que os batimentos anterior e posterior ao batimento atual auxiliam a classificação, fornecendo melhores resultados para essa derivação. Já na segunda derivação, ambas as configurações B1 e B2 apresentam métricas baixas para classe S, Acurácias menores do que a primeira derivação, e resultados semelhantes entre si, com exceção da classe F, que foi melhor classificada na B1.

O uso de três batimentos das duas derivações (AB) não apresenta um ganho significativo em relação ao uso de A2, com um aumento apenas na Sensibilidade de V. Além disso, requer aproximadamente o dobro de parâmetros, sendo mais custoso. Em A3, o acréscimo da informação dos intervalos RR à configuração A2 auxiliou as classes V e F, mas prejudicou S. Por fim, a extração de características na configuração C permitiu aumentar a Sensibilidade de F, levando aos maiores valores de métricas dessa classe, e à maior Precisão de N, apesar de utilizar uma quantidade de parâmetros menor.

Na literatura, o desempenho da classe S é, em geral, muito baixo. Luz et al. implementaram em [17] os métodos propostos em [18-22] seguindo as recomendações da AAMI, o que levou a valores de no máximo 27,0% para Sensibilidade e de no máximo 48,3% para Precisão. Os resultados de [17] são inferiores aos da rede proposta com A2 na maioria das métricas. Considerando-se também os artigos [23-26], que otimizam classificadores para N, S e V, em um problema de apenas 3 classes, a rede proposta com A2 atinge uma Precisão da classe S melhor do que o maior dos valores da literatura, que é 53%, mas apresenta as métricas da classe N menores. Ainda assim, foi possível atingir os valores relatados pela literatura para as classes S e V, mesmo sem uma otimização para o problema de 3 classes.

Conclusões

Diversos trabalhos que utilizam redes neurais para classificação de arritmias cardíacas não separam os dados dos pacientes de modo realista e atingem desempenhos muito bons, com métricas acima de 95% de acerto. Como sugerido por [17], para permitir idenficar os méritos dos diferentes métodos de diagnóstico automático propostos na literatura, é importante o estabelecimento de padrões de separação dos pacientes em treinamento e teste como os da AAMI. Além disso, observou-se que a eliminação de dados pode prejudicar a generalização da rede, descartando exemplos que poderiam contribuir no treinamento. Esse fato também foi constatado por [13] e evitado por uma função custo com pesos, que lida com o desbalanceamento dos dados e obtém resultados melhores do que subamostrar extensivamente algumas classes ou sobreamostrar dados.

Diversas simulações com a rede proposta mostraram que, para a primeira derivação, uma MLP com entrada de três batimentos atinge resultados melhores para a maior parte das métricas do que uma entrada com apenas um batimento. Isso se deve à informação do batimento anterior e posterior ao analisado, e, consequentemente, à informação do intervalo RR que entra na rede, o que pode ser justificado pela irregularidade desses intervalos em classes de arritmia, como a S, e pela sua regularidade na classe N.

A comparação dos resultados mostra que a rede proposta com A2 alcança os valores usuais apresentados na literatura para as classes N, S e V, e atinge uma Precisão de S maior do que [13,18-26], apesar de obter Acurácia e métricas de N menores. Similarmente, a rede proposta com C possibilita desempenhos melhores na classe F, com valor de Precisão maior do que os demais, e utiliza uma quantidade bem menor de parâmetros. Percebeu-se que as entradas de [13], também utilizadas por outros artigos, beneficiam as métricas da classe F na MLP proposta. Concluiu-se, principalmente, que a extração de características é uma etapa essencial na MLP, uma vez que permite alcançar bons resultados utilizando uma menor quantidade de parâmetros.

Referências bibliográficas

[1] **World Health Organization:** The top 10 causes of death, 8 de dez. 2020. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>. Acesso em: 9 de jan. de 2021.

[2] BERKAYA, S. K. et al. A survey on ECG analysis. **Biomedical Signal Processing and Control**, v. 43, p. 216-235, 2018.

[3] DEWANGAN, N. K.; SHUKLA, S. P. ECG arrhythmia classification using discrete wavelet transform and artificial neural

network. In: **2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)**. IEEE, 2016. p. 1892-1896.

[4] MARINHO, L. B. et al. A novel electrocardiogram feature extraction approach for cardiac arrhythmia classification. **Future Generation Computer Systems**, v. 97, p. 564-577, 2019.

[5] GOODFELLOW, I. et al. **Deep learning**. Cambridge: MIT press, 2016.

[6] HANNUN, A. Y. et al. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. **Nature medicine**, v. 25, n. 1, p. 65-69, 2019.

[7] ACHARYA, U. R. et al. A deep convolutional neural network model to classify heartbeats. **Computers in biology and medicine**, v. 89, p. 389-396, 2017.

[8] WANG, G. et al. A global and updatable ECG beat classification system based on recurrent neural networks and active learning. **Information Sciences**, v. 501, p. 523-542, 2019.

[9] BANERJEE, R.; GHOSE, A.; KHANDELWAL, S. A Novel Recurrent Neural Network Architecture for Classification of Atrial Fibrillation Using Single-lead ECG. In: **2019 27th European Signal Processing Conference (EUSIPCO)**. IEEE, 2019. p. 1-5.

[10] GOLDBERGER, A. L. et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. **Circulation**, v. 101, n. 23, p. e215–e220, 2000.

[11] MOODY, G. B.; MARK, R. G. The impact of the MIT-BIH arrhythmia database. **IEEE Engineering in Medicine and Biology Magazine**, v. 20, n. 3, p. 45-50, 2001.

[12] AAMI, ANSI; EC57, A. A. M. I. (R) 2008-Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms. **American National Standards Institute**, Arlington, VA, USA, 2008.

[13] DE CHAZAL, P.; O'DWYER, M.; REILLY, R. B. Automatic classification of heartbeats using ECG morphology and heartbeat interval features. **IEEE Transactions on Biomedical Engineering**, v. 51, n. 7, p. 1196-1206, 2004.

[14] MARTÍNEZ, J. P. et al. A wavelet-based ECG delineator: evaluation on standard databases. **IEEE Transactions on Biomedical Engineering**, v. 51, n. 4, p. 570-581, 2004.

[15] DEMSKI, A. J.; SORIA, M. L. ecg-kit: a Matlab Toolbox for Cardiovascular Signal Processing. **Journal of Open Research Software**, v. 4, n. 1, 2016.

[16] ABADI, Martín et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Disponível em: < https://www.tensorflow.org>. Acesso em: 17 de jan. de 2020.

[17] LUZ, E.; MENOTTI, D. How the choice of samples for building arrhythmia classifiers impact their performances. In: **2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society**. IEEE, 2011. p. 4988-4991.

[18] YE, C.; COIMBRA, M. T.; KUMAR, B. V. K. V. Arrhythmia detection and classification using morphological and dynamic features of ECG signals. In: **2010 Annual International Conference of the IEEE Engineering in Medicine and Biology**. IEEE, 2010. p. 1918-1921.

[19] YU, S.-N.; CHOU, K.-T. Integration of independent component analysis and neural networks for ECG beat classification. **Expert Systems with Applications**, v. 34, n. 4, p. 2841-2846, 2008.

[20] YU, S.-N.; CHEN, Y.-H. Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network. **Pattern Recognition Letters**, v. 28, n. 10, p. 1142-1150, 2007.

[21] GÜLER, I.; ÜBEYLI, E. D. ECG beat classifier designed by combined neural network model. **Pattern recognition**, v. 38, n. 2, p. 199-208, 2005.

[22] SONG, M.-H. et al. Support vector machine based arrhythmia classification using reduced features. **International Journal of Control, Automation, and Systems**, v. 3, n. 4, p. 571-579, 2005.

[23] LLAMEDO, M.; MARTÍNEZ, J. P. Heartbeat classification using feature selection driven by database generalization criteria. **IEEE Transactions on Biomedical Engineering**, v. 58, n. 3, p. 616-625, 2010.

[24] LIN, C.-C.; YANG, C.-M. Heartbeat classification using normalized RR intervals and morphological features. **Mathematical Problems in Engineering**, v. 2014, 2014.

[25] GARCIA, G. et al. Improving automatic cardiac arrhythmia classification: Joining temporal-VCG, complex networks and SVM classifier. In: **2016 International Joint Conference on Neural Networks (IJCNN).** IEEE, 2016. p. 3896-3900.

[26] GARCIA, G. et al. Inter-patient ECG heartbeat classification with temporal VCG optimized by PSO. **Scientific Reports**, v. 7, n. 1, p. 1-11, 2017.

Trabalho submetido ao XXXIX SBrT

Combinações de redes neurais e discriminantes lineares para classificação de arritmias cardíacas

Natália Nagata, Renato Candido e Magno T. M. Silva

***Resumo*— Neste trabalho, utilizam-se redes neurais *perceptron* multicamada, redes neurais recorrentes, análise de discriminantes lineares e combinações desses métodos para classificação automática de arritmias cardíacas. A fim de se obter resultados clinicamente realistas para o diagnóstico, dados de um mesmo paciente não foram usados simultaneamente nas fases de treinamento e de teste. Os resultados indicam que a combinação da rede *perceptron* multicamada com a análise de discriminantes lineares apresenta um desempenho melhor em relação aos modelos individuais. Além disso, esse esquema combinado alcança métricas de classificação superiores às da literatura para as classes de arritmia S e V.**

***Palavras-Chave*— Aprendizado de máquina, redes neurais, análise de discriminante linear, eletrocardiograma, arritmias cardíacas.**

***Abstract*— In this work, we use multilayer neural networks (MLP), recurrent neural networks, linear discriminant analysis (LDA) and combinations of these methods for automatic classification of cardiac arrhythmias. In order to obtain clinically realistic results for the diagnosis, patients records used during the training phase were not used in the test set. The results indicate that the combination of MLP with LDA presents a better performance compared to those of individual models. Furthermore, this combined scheme achieves classification metrics superior to those reported in the literature for the S and V arrhythmia classes.**

***Keywords*— Machine learning, neural networks, linear discriminant analysis, electrocardiogram, cardiac arrhythmias.**

I. Introdução

O eletrocardiograma (ECG) registra a atividade elétrica do coração por meio de um arranjo de eletrodos que captam os potenciais superficiais do corpo ao longo do tempo. É um teste efetivo de baixo custo e não invasivo que se tornou uma ferramenta padrão na identificação de doenças do coração [1].

Segundo a Organização Mundial da Saúde (OMS), as doenças cardiovasculares são a principal causa de morte no mundo [2]. Dentre essas doenças, as arritmias cardíacas são as mais comuns e a sua classificação precisa é de grande interesse em estudos biomédicos [2]. As arritmias correspondem a qualquer distúrbio na taxa, na regularidade e nos locais de origem ou condução dos impulsos elétricos cardíacos. Um importante passo no seu diagnóstico é a classificação de batimentos consecutivos do sinal de ECG [3].

A análise manual desses batimentos demanda muito tempo do cardiologista. Em alguns casos, é necessário gravar o sinal de ECG por até uma semana para identificar determinadas arritmias, possibilitando a perda de informações importantes e dificultando resultados imediatos [4], [5]. Uma alternativa é o uso de métodos computadorizados para a identificação automática das arritmias. A partir da década de 1990, surgiram vários trabalhos em que soluções baseadas em redes neurais *perceptron* multicamada (*multilayer perceptron* – MLP) [6], análise de discriminantes lineares (*linear discriminant analysis* – LDA) [7] e máquinas de vetores de suporte (*support vector machines* – SVM) [8] foram propostas para esse problema.

As altas taxas de erro desses métodos em conjunto com o crescimento da área de aprendizado de máquina [9] fizeram com que a pesquisa em automatização do diagnóstico de arritmias cardíacas atraísse novamente a atenção da comunidade científica. Trabalhos recentes consideram o uso de redes neurais convolucionais (*convolutional neural network* – CNN) [10] e redes neurais recorrentes (*recurrent neural networks* – RNN) [11] na classificação de arritmias. Porém, em vários desses trabalhos, os autores não seguem as recomendações da *Association for the Advancement of Medical Instrumentation* (AAMI) [12] e utilizam dados dos mesmos pacientes tanto no conjunto de teste, quanto no de treinamento. Com exceção de métodos direcionados para o monitoramento de pacientes específicos (*patient-specific*) [13], essa abordagem não é clinicamente realista, já que na prática o sistema será utilizado em pacientes cujos dados não foram usados no treinamento.

Dentre os classificadores que seguem uma divisão realista dos pacientes, métodos baseados em LDA são os mais comuns, e também têm sido propostos em estudos recentes (ver, e.g., as referências de [14], [15]). Ao usar a divisão por pacientes proposta por [4], é um desafio otimizar MLPs e SVMs para obter resultados promissores nas classes menos representadas. Uma das maiores vantagens da LDA é a facilidade em lidar com problemas gerados pelo desbalanceamento do número de dados das classes [14].

Como a divisão de pacientes em treinamento e teste nem sempre é considerada, torna-se difícil avaliar e comparar os métodos de classificação de arritmias cardíacas contidos na literatura. Além disso, a combinação de classificadores foi pouco explorada para o diagnóstico automático de arritmias seguindo as recomendações da AAMI [14].

Neste artigo, propõe-se o uso de redes neurais dos tipos MLP e RNN, do método estatístico da LDA e de combinações desses classificadores para o diagnóstico automático de arritmias. Foram considerados sinais de ECG do banco de dados *MIT-BIH Arrhythmia Database* (MITDB) [16], [17], levando-se em conta a separação dos pacientes nas fases de treinamento e teste. Dessa forma, foi possível comparar o desempenho dos diferentes classificadores propostos em termos de métricas de classificação, avaliar o efeito da combinação desses classifica-

Natália Nagata, Renato Candido, Magno T. M. Silva, Depto. de Engenharia de Sistemas Eletrônicos, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil, e-mails: nagata.natalia@usp.br; renatocan@lps.usp.br; magno.silva@usp.br. Este trabalho foi financiado pela FAPESP (2019/26911-6 e 2017/20378-9) e CAPES (código de financiamento 001).

dores e comparar os resultados obtidos com os da literatura.

O artigo está organizado da seguinte forma. Na Seção II, descreve-se o banco de dados utilizado e as classes de arritmia consideradas. Em seguida, apresentam-se as etapas de reconhecimento do sinal de ECG anteriores à classificação. Na Seção IV, as estruturas dos classificadores são descritas em maiores detalhes. As Seções V e VI contêm respectivamente os resultados de simulação e as principais conclusões do trabalho.

II. Banco de Dados

O MITDB é composto por 48 gravações ambulatoriais de duas derivações de 30 minutos de pacientes do *Boston's Beth Israel Hospital*, com anotações manuais de cada batimento feitas por cardiologistas. Neste trabalho, seguiu-se a recomendação da AAMI e dividiu-se os conjuntos de treinamento e de teste conforme a separação proposta por De Chazal *et al.* [4]. A AAMI também recomenda o agrupamento das anotações em cinco classes: batimentos do nó sinoatrial (N), supraventriculares ectópicos (S), ventriculares ectópicos (V), fusão de batimentos normais e ventriculares ectópicos (F) e desconhecidos ou de marca-passo (Q). No entanto, devido à ausência de resultados promissores da classe Q, tanto na literatura quanto neste trabalho, essa classificação não foi considerada. Além desse agrupamento em um problema de quatro classes, muitos autores otimizam um problema de três classes, classificando N, S e V, e removendo a classe F devido à menor quantidade de dados [15], [18]–[20].

Para lidar com o desbalanceamento do número de dados entre as classes, os dados da classe N foram subamostrados e as funções custo foram ponderadas de forma proporcional ao inverso da quantidade de dados de cada classe. A Tabela I mostra a quantidade total disponível de dados e a quantidade utilizada no trabalho.

TABELA I: Número total de batimentos e número utilizado para os conjuntos de teste e de treinamento.

Classe	Número total de batimentos		Número utilizado	
	Conjunto de treinamento	Conjunto de teste	Conjunto de treinamento	Conjunto de teste
N	40098	40052	8492	8359
S	755	1319	755	1319
V	2538	2034	2538	2034
F	396	376	396	376

III. Etapas de Reconhecimento do Sinal

Em geral, o diagnóstico a partir do sinal de ECG por técnicas computadorizadas é dividido em quatro etapas: o pré-processamento do sinal, a segmentação, a extração das características e a classificação. A etapa de pré-processamento reduz os ruídos e artefatos provenientes de várias origens, como a interferência da rede elétrica. A etapa de segmentação delimita o complexo QRS e as ondas P e T. Para isso, usou-se o método proposto em [21], baseado em transformada de wavelet e implementado em MATLAB pelo pacote ecg-kit [22]. Já a etapa de extração de características determina a menor quantidade de características do sinal de ECG que permite taxas de classificação aceitáveis [5]. A etapa de classificação é explorada na seção seguinte.

IV. Classificadores

Na etapa de classificação, foram considerados três classificadores: uma rede MLP, uma RNN e uma LDA. Cada método foi implementado individualmente e, posteriormente, suas saídas foram combinadas. As redes foram implementadas utilizando as bibliotecas Tensorflow e Keras [23] e a LDA, utilizando a biblioteca scikit-learn [24]. Na sequência, abordam-se os parâmetros e a entrada da rede MLP considerada. Detalha-se também o bloco da RNN utilizada e os seus parâmetros. Por fim, descreve-se o método de balanceamento da LDA por pesos e a extração de características usada nesse método.

A. Rede neural MLP

Inspirando-se nos resultados de [25], para classificar um determinado batimento, foram utilizados também o batimento anterior e o batimento posterior. Considerou-se que cada batimento compreendia 320 amostras, totalizando 960 amostras da primeira derivação do sinal como entrada da rede.

A rede MLP utilizada é composta por duas camadas ocultas, a primeira com 32 neurônios e a segunda com 16, e funções de ativação ReLU em ambas. Na camada de saída, foram utilizados 4 neurônios com função *Softmax*. O número de camadas e o número de épocas foram ajustados usando *grid search* a fim de se obter o melhor desempenho. Para trabalhar com as classes desbalanceadas, usou-se a função custo de entropia cruzada categórica com pesos calculados na proporção inversa do número de dados de cada classe. Essa função é dada por

$$J_{WCCE} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} p_k \times y_k^{(L)}(n) log(d_k(n)), \quad (1)$$

em que N é o número de exemplos de treinamento, K é o número de classes, p_k é o peso da classe k, $d_k(n)$ é o valor real do n-ésimo exemplo de treinamento e $y_k^{(L)}(n)$ é a saída da rede para o n-ésimo exemplo. Considerou-se ainda o algoritmo de otimização Adam [26] com $\beta_1 = 0,9$ e $\beta_2 = 0,99$, e o algoritmo de retropropagação (*backpropagation*) com passo de aprendizado $\eta = 0,001$, 250 épocas, mini-batches de tamanho $k = 2048$ e inicialização de Xavier [27] para os pesos.

B. Rede neural recorrente

A rede RNN utilizada possui uma camada com três blocos de LSTM (*Long Short-Term Memory*), o que corresponde a três passos de tempo. Como na MLP, usou-se uma entrada de 960 amostras da primeira derivação do sinal de ECG, com 320 amostras de entrada para cada passo de tempo, como ilustrado na Figura 1. A LSTM foi originalmente proposta em [28] para resolver o problema de desvanecimento do gradiente. Ela possui portas que controlam o fluxo de informação e permitem o aprendizado de dependências de longo prazo.

O estado interno da célula LSTM (*cell internal state*) $s_i^{(t)}$ é representado pela linha horizontal azul no topo do bloco LSTM da Figura 1. Essa unidade é controlada pela porta *forget gate* $f_i^{(t)}$, para cada passo de tempo t e unidade i da camada oculta, sendo o número total de unidades igual a 72 na rede considerada. A *forget gate* usa uma função de ativação sigmoidal tendo como saída um valor entre 0 e 1 que controla o fluxo de informação do estado $s_i^{(t-1)}$ para o estado $s_i^{(t)}$

A entrada $x_i^{(t)}$ de um bloco LSTM fornece a saída $u_i^{(t)}$ que pode ser "acumulada" no estado se a porta de entrada externa

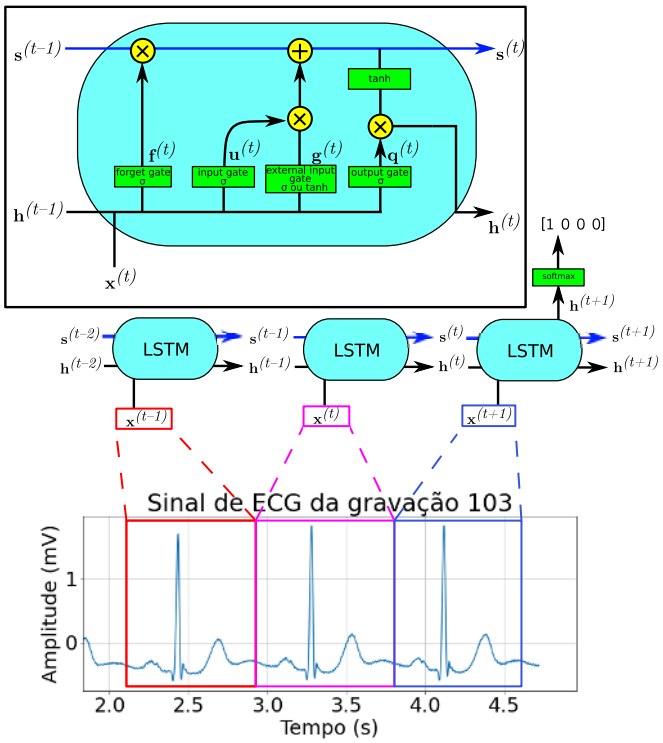


Fig. 1: Rede RNN utilizada com três passos de tempos.

(*external input gate*) $g_i^{(t)}$ permitir. A porta de saída (*output gate*) $q_i^{(t)}$ controla a ocorrência ou não de saída na célula. O estado $s_i^{(t)}$ é atualizado a cada passo de tempo, e, após passar por uma tangente hiperbólica e ser multiplicado pela porta de saída, fornece o vetor de estado oculto $h_i^{(t)}$ da célula. No último passo de tempo, esse vetor é conectado a uma camada com 4 neurônios de saída e função *Softmax*.

Na RNN considerada neste trabalho, usou-se a função custo de entropia cruzada categórica com pesos da Equação (1), o algoritmo de otimização Adam com $\beta_1 = 0,9$ e $\beta_2 = 0,99$, e o algoritmo de retropropagação com passo de aprendizado $\eta = 0,001$, 100 épocas, mini-batches de tamanho $k = 32$, inicialização de Xavier para os pesos da entrada e inicialização ortogonal para os pesos recorrentes.

C. Análise de discriminante linear (LDA)

A LDA é uma técnica bem conhecida para extração de características, redução de dimensionalidade e classificação, sendo uma generalização do discriminante linear de Fisher [29] para K classes. Na LDA, uma entrada $\mathbf{x}$ de um espaço D-dimensional é projetada para um espaço $(K-1)$-dimensional por meio de $K-1$ funções discriminantes $y_k = \mathbf{w}_k^T\mathbf{x}$, com $k = 1, \ldots, K-1$. Essas funções podem ser agrupadas em um vetor $\mathbf{y}$, e $\mathbf{w}_k$ podem ser agrupados como colunas de uma matriz $\mathbf{W}$ de dimensão $D \times (K-1)$, com $\mathbf{y} = \mathbf{W}^T\mathbf{x}$. Após a projeção, é possível classificar a entrada como pertencente à classe C_k por meio da comparação com um limiar [30]. Denotando a matriz de covariância entre classes e a matriz de covariância dentro de cada classe por $\mathbf{S}_B$ e $\mathbf{S}_W$, respectivamente, a função custo para o caso de $K = 2$ é dada por $J(\mathbf{w}) = (\mathbf{w}^T\mathbf{S}_B\mathbf{w})/(\mathbf{w}^T\mathbf{S}_W\mathbf{w})$. Essa função custo busca maximizar a separação entre classes ao mesmo tempo em que reduz a sobreposição entre elas, fornecendo a melhor direção de projeção dos dados. Para $K > 2$ classes, a solução que maximiza $J(\mathbf{w})$ é obtida quando $\mathbf{W}$ é composta pelos autovetores associados aos maiores autovalores de $\mathbf{S}_W^{-1}\mathbf{S}_B$ [30].

Para levar em conta o desbalanceamento da quantidade de dados das classes, calculam-se $\mathbf{S}_B$ e $\mathbf{S}_W$ com pesos p_k na proporção inversa ao número de dados das classes, ou seja,

$$\mathbf{S}_W = \sum_{k=1}^{K} p_k \sum_{n \in C_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T, e \qquad (2)$$

$$\mathbf{S}_B = \sum_{k=1}^{K} p_k n_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T, \qquad (3)$$

em que n_k é o número de dados na classe C_k, $\mathbf{m}_k$ é a média desses dados, e $\mathbf{m}$ é a média ponderada

$$\mathbf{m} = \frac{\sum_{k=1}^{K} p_k \sum_{n \in C_k} \mathbf{x}_n}{\sum_{k=1}^{K} p_k n_k}. \qquad (4)$$

Os vetores de entrada de cada batimento cardíaco utilizado na LDA são compostos por 10 amostras do complexo QRS, 9 amostras entre o fim do QRS e o fim da Onda T, informações dos intervalos RR, da duração do QRS e da Onda T e a presença da Onda P, como proposto por [4] e ilustrado na Figura 2. Os intervalos RR usados foram os intervalos entre o batimento atual e o anterior (*Pre-RR*), entre o batimento atual e o posterior (*Post-RR*), o intervalo médio de toda a gravação de um paciente (*Average RR*) e o intervalo médio local entre 10 batimentos adjacentes (*Local average RR*), totalizando 26 características extraídas para cada batimento.

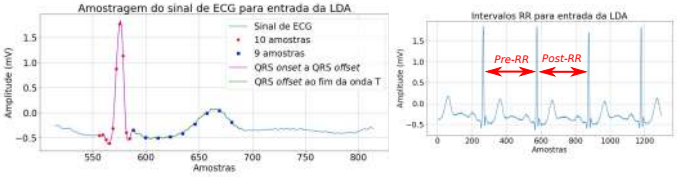


Fig. 2: Extração de características da LDA.

Para cada derivação (*lead A* e *lead B*), foi considerado um classificador. As saídas desses classificadores foram combinadas para fornecer a saída global como explicado na Subseção IV-D abaixo. A LDA usada possui um estimador de covariância por máxima verossimilhança, probabilidades marginais iguais entre as classes, ou seja, de $1/4$, e a menor redução de dimensionalidade possível ($K - 1 = 3$).

D. Combinação de classificadores

Em geral, classificadores usados em uma aplicação específica atingem diferentes graus de sucesso, apoiados em conjuntos de características distintos. A combinação de classificadores em um único sistema de reconhecimento de arritmias ajuda a integrar o conhecimento adquirido pelos diferentes modelos [31]. Em [4], as configurações de classificadores combinados obtiveram maiores acurácias em todos os resultados em comparação ao uso de apenas um classificador. Neste trabalho, usou-se a operação de multiplicação elemento a elemento proposta em [4] para combinar os classificadores.

Seja $P_m(k \mid \mathbf{x})$ a probabilidade de uma entrada $\mathbf{x}$ pertencer a uma classe k fornecida como uma posição do vetor de saída do m-ésimo classificador, o vetor final de saída de um modelo combinado para a entrada $\mathbf{x}$ é dado por [4]

$$P(k \mid \mathbf{x}) = \frac{\prod_{m=1}^{M} P_m(k \mid \mathbf{x})}{\sum_{l=1}^{K} \prod_{m=1}^{M} P_m(l \mid \mathbf{x})}, \quad (5)$$

sendo $K = 4$ o número de classes e M o número de classificadores. A classificação global é obtida pela escolha da classe com a maior probabilidade condicional resultante dessa operação. Essa combinação fornece resultados mais confiáveis em probabilidades que foram estimadas como zero, reduzindo o erro total das redes e a incidência de falsos negativos.

V. Resultados e Discussão

Foram feitas simulações considerando a MLP, a RNN e a LDA individualmente e as combinações MLP–LDA, MLP–RNN, RNN–LDA e MLP–RNN–LDA. Para cada simulação, calcularam-se as métricas de Acurácia geral $Acc = (VP + VN)/(VP + VN + FP + FN)$ e de Sensibilidade $Se = VP/(VP + FN)$ e Precisão $P = VP/(VP + FP)$ para cada classe, sendo VP, VN, FP e FN as quantidades de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos, respectivamente. Seguindo a recomendação da AAMI [4], [12], os falsos positivos devido à classe F foram desconsiderados para a classe V. Calcularam-se também as métricas de $F1$-score $F1 = 2 \times Se \times P/(Se + P)$ para cada classe e as métricas gerais de *macro*-$F1$-score $mF1 = \sum_{k=1}^{K} F1_k/K$ e de $F1$-score ponderado para 3 e 4 classes $wF1 = \sum_{k=1}^{K} n_k F1_k / \sum_{k=1}^{K} n_k$, em que n_k é o número de elementos da classe k da Tabela I. Os resultados para cada simulação estão apresentados na Tabela II, junto a outros resultados da literatura que seguem as recomendações da AAMI. Os dois maiores valores das simulações para cada métrica são apresentados em negrito.

Os trabalhos da literatura cujos resultados aparecem na Tabela II são considerados como estado da arte. Em [4], desenvolveu-se uma LDA para classificar os batimentos em um problema de cinco classes, a partir da extração de características descrita na Subseção IV-C. Em [32], características temporais, morfológicas e estatísticas do sinal foram utilizadas junto a um algoritmo de *sequential forward floating search* para encontrar combinações de características ótimas, com classificadores baseados em LDA e MLP. Os resultados de [32] apresentados na Tabela II foram obtidos com uma rede MLP. Em [19], foi proposta uma LDA com pesos e extração de características de intervalos RR e características morfológicas usando transformada de wavelet. Em [20], o ECG foi representado em três dimensões por meio do *temporal vectorcardiogram* (TVCG). Essa representação foi usada em redes complexas para extração de características, que, por sua vez, foram consideradas como entrada de um classificador SVM. Por fim, [15] aprimorou o modelo de [20] com o uso de *particle swarm optimization* para seleção de características.

Analisando-se os classificadores propostos neste trabalho individualmente, é possível observar que a MLP apresenta as métricas gerais de Acc e de $wF1$ maiores do que a RNN e a LDA, além de $F1$ maiores para as classe N, S e V. Para essas classes, a MLP também atinge os valores de $F1$ usuais da literatura, porém, para a classe S, obtém um valor de $F1$ de $57,4\%$, que é maior do que o valor obtido em todos os outros trabalhos considerados. Para a classe V, o valor de $F1$ é de $76,2\%$, que se encontra na média dos demais trabalhos. Com relação à classe N, o desempenho da MLP é um pouco pior. Observa-se que o melhor classificador individual para a classe F é a LDA, com a maior métrica de Se de $83,8\%$, e de $F1$ de $29,7\%$. Nota-se que as entradas propostas por [4] e utilizadas na LDA auxiliam a identificação da classe F.

Os resultados dos classificadores combinados foram melhores do que os dos classificadores individuais, em termos das métricas Acc e $wF1$. Comparando-se a combinação RNN–LDA com a RNN individual, por exemplo, aumentou-se a Acc de 74,4% da RNN para 81,7%, e o valor de $wF1$ de 75,1% da RNN para 82,3%. Além disso, o valor de $F1$ de todas as classes foi maior na combinação da RNN–LDA do que na RNN e na LDA separadamente.

Observando-se os valores em negrito, é possível perceber que a MLP–LDA e a MLP–RNN–LDA apresentaram os melhores desempenhos gerais dentre todas as simulações. Combinando-se a MLP com a LDA, a Acc de 77,2% da MLP aumentou para 84,2% e o valor de $wF1$ de 78,3% aumentou para 84,5%. Porém, ao realizar-se a combinação MLP–RNN–LDA, não é possível notar variação significativa das métricas em relação ao modelo MLP–LDA, com o valor de Acc aumentando apenas 0,2% e de $wF1$ aumentando 0,3%. Assim, a contribuição da RNN na combinação não foi relevante, além de levar a um aumento de 113476 parâmetros no modelo. Portanto, decidiu-se comparar o modelo MLP–LDA com os resultados da literatura. A matriz de confusão obtida por esse modelo está na Tabela III.

A combinação MLP–LDA obtém valores de $F1$ para as classe S e V de 61,9% e 86,6% respectivamente, maiores do que todos os outros da literatura. Além disso, obtém o valor de $F1$ de N de 90,8%, com uma diferença de 5,7% do maior valor dos demais trabalhos. No entanto, o valor de $F1$ da classe F é menor do que os de [4] e [32], que consideram o problema de quatro classes. Em relação ao desempenho geral, o valor de $wF1$ foi menor dos que os demais, com uma diferença máxima de 6,8%, no caso de quatro classes. Considerando três classes, essa diferença passa a ser de 7,1%. Dentre os classificadores que usaram as quatro classes, a diferença máxima desconsiderando a classe F foi de 4,7%.

Apesar dos valores de $F1$ maiores nas classes S e V, a combinação MLP–LDA obteve $wF1$ menor devido ao peso da classe N durante a ponderação no cálculo dessa métrica. Neste trabalho, não foram considerados todos os batimentos da classe N, pois o desbalanceamento dos dados na proporção original prejudicaria muito o treinamento dos classificadores, mesmo com a correção realizada pelos pesos na função custo. Avaliando-se a métrica $mF1$, ao atribuir pesos iguais às métricas $F1$ de todas as classes, o método proposto alcança valores maiores do que os relatados nos demais trabalhos.

VI. Conclusões

A classificação de arritmias cardíacas por métodos computadorizados é uma área com muitas possibilidades de melhoria. Neste trabalho, avaliou-se o efeito de combinar os classificadores MLP, RNN e LDA. Dentre os classificadores individuais,

TABELA II: Métricas (%) dos resultados das simulações e comparação com os valores da literatura. Os dois maiores resultados para cada métrica calculada com os resultados dos métodos propostos estão em negrito.

Métodos propostos	Acc	N			S			V			F			$wF1$ 4 classes	$wF1$ 3 classes	$mF1$
		Se	P	F1	Se	P	F1	Se	P	F1	Se	P	F1			
MLP	77,2	82,4	88,7	85,4	**57,2**	57,6	57,4	82,1	71,1	76,2	4,0	2,8	3,3	78,3	80,7	73,0
RNN	74,4	79,3	84,4	81,8	55,4	47,7	51,2	77,5	71,1	74,2	15,2	12,3	13,6	75,1	77,0	69,1
LDA	69,9	79,2	**93,3**	85,7	46,3	31,0	37,1	44,1	70,6	54,3	**83,8**	**18,1**	**29,7**	73,4	74,8	59,0
MLP–LDA	**84,2**	**90,4**	**91,1**	**90,8**	56,8	**68,0**	**61,9**	**89,3**	84,1	**86,6**	15,4	12,6	13,9	**84,5**	**86,8**	**79,8**
MLP–RNN	78,8	84,3	87,1	85,7	56,8	57,7	57,2	83,9	73,4	78,3	4,52	5,7	5,0	78,8	81,2	73,7
RNN–LDA	81,7	89,6	89,4	89,5	55,7	56,1	55,9	73,6	**86,1**	79,4	**40,7**	**23,5**	**29,8**	82,3	83,9	74,9
MLP–RNN–LDA	**84,4**	**91,1**	89,2	**90,1**	**57,2**	**71,2**	**63,5**	**87,9**	**85,1**	**86,5**	12,2	13,3	12,7	**84,2**	**86,5**	**80,0**

Métodos da literatura	Acc	N			S			V			F			$wF1$ 4 classes	$wF1$ 3 classes	$mF1$
		Se	P	$F1$	Se	P	$F1$	Se	P	$F1$	Se	P	$F1$			
Chazal [4]	85,9	86,9	99,2	92,6	75,9	38,5	51,1	77,7	81,9	79,7	89,4	8,6	15,7	89,7	90,3	74,5
Mar [32]	90,0	89,6	99,1	94,11	83,2	33,5	47,8	86,8	75,9	81,0	61,1	16,6	26,1	91,0	91,5	74,3
Llamedo†[18]	93,0	95,0	98,0	96,5	77,0	39,0	51,8	81,0	87,0	83,9	–	–	–	–	93,9	77,4
Lin†[19]	90,8	91,6	99,3	95,3	81,4	31,6	45,5	86,2	73,7	79,5	–	–	–	–	92,4	73,4
Garcia†[20]	91,0	95,0	96,0	95,5	30,0	26,0	27,9	85,0	66,0	74,3	–	–	–	–	91,6	65,9
Luz†[15]	92,4	94,0	98,0	96,0	62,0	53,0	57,2	87,3	59,4	70,7	–	–	–	–	92,9	74,6

† Autores otimizaram os métodos para o problema de três classes: N, S e V.

TABELA III: Matriz de confusão obtida para MLP–LDA.

		Classes Preditas			
		N	S	V	F
Classes Verdadeiras	N	7557	317	161	324
	S	370	749	183	17
	V	127	28	1817	62
	F	240	7	71	58

a MLP apresentou os melhores resultados para as classes N, S e V, consideradas como principais pela maioria dos trabalhos. Além disso, verificou-se que as combinações ajudam efetivamente a melhorar o desempenho geral do classificador, aumentando a maioria das métricas. Observou-se ainda que a combinação MLP–LDA atingiu bons valores de $F1$-score para as classes S e V, apresentando valores de $wF1$ e $mF1$ próximos aos dos classificadores concebidos como estado da arte.

REFERÊNCIAS

[1] L. B. Marinho *et al.*, "A novel electrocardiogram feature extraction approach for cardiac arrhythmia classification," *Future Generation Computer Systems*, vol. 97, pp. 564–577, 2019.

[2] S. K. Berkaya *et al.*, "A survey on ECG analysis," *Biomedical Signal Processing and Control*, vol. 43, pp. 216–235, 2018.

[3] S. I. Niwas, R. S. S. Kumari, and V. Sadasivam, "Artificial neural network based automatic cardiac abnormalities classification," in *Proc. of ICCIMA'05*. IEEE, 2005, pp. 41–46.

[4] P. de Chazal, M. O'Dwyer, and R. B. Reilly, "Automatic classification of heartbeats using ECG morphology and heartbeat interval features," *IEEE Trans. Biomed. Eng.*, vol. 51, pp. 1196–1206, 2004.

[5] N. K. Dewangan and S. P. Shukla, "ECG arrhythmia classification using discrete wavelet transform and artificial neural network," in *Proc. of RTEICT*. IEEE, 2016, pp. 1892–1896.

[6] L. Edenbrandt, B. Devine, and P. W. Macfarlane, "Neural networks for classification of ECG ST-T segments," *J. Electrocardiol.*, vol. 25, pp. 167–173, 1992.

[7] Yun-Chi Yeh, Wen-June Wang, and Che Wun Chiou, "Cardiac arrhythmia diagnosis method using linear discriminant analysis on ECG signals," *Measurement*, vol. 42, pp. 778–789, 2009.

[8] M. H. Song *et al.*, "Support vector machine based arrhythmia classification using reduced features," *Int. J. Control, Automation, and Systems*, vol. 3, pp. 571–579, Dec. 2005

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[10] U. R. Acharya *et al.*, "A deep convolutional neural network model to classify heartbeats," *Computers in Biology and Medicine*, vol. 89, pp. 389–396, 2017.

[11] G. Wang *et al.*, "A global and updatable ECG beat classification system based on recurrent neural networks and active learning," *Information Sciences*, vol. 501, pp. 523–542, 2019.

[12] Association for the Advancement of Medical Instrumentation (AAMI), "ANSI/AAMI EC57:1998/(R)2008 - Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms," *American National Standards Institute, Arlington, VA, USA*, 2008.

[13] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," *IEEE Trans. Biomed. Eng.*, vol. 63, pp. 664–675, 2015.

[14] E. J. S. Luz *et al.*, "ECG-based heartbeat classification for arrhythmia detection: A survey," *Computer Methods and Programs in Biomedicine*, vol. 127, pp. 144–164, 2016.

[15] G. Garcia *et al.*, "Inter-patient ECG heartbeat classification with temporal VCG optimized by PSO," *Scientific Reports*, vol. 7, pp. 1–11, 2017.

[16] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. e215–e220, Jun. 13, 2000.

[17] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, pp. 45–50, 2001.

[18] M. Llamedo and J. P. Martínez, "Heartbeat classification using feature selection driven by database generalization criteria," *IEEE Trans. Biomed. Eng.*, vol. 58, pp. 616–625, 2011.

[19] C. Lin and C. Yang, "Heartbeat classification using normalized RR intervals and morphological features," *Mathematical Problems in Engineering*, 2014.

[20] G. Garcia *et al.*, "Improving automatic cardiac arrhythmia classification: Joining temporal-VCG, complex networks and SVM classifier," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 3896–3900.

[21] J. P. Martínez *et al.*, "A wavelet based ECG delineator: evaluation on standard databases," *IEEE Trans. Biomed. Eng.*, vol. 51, pp. 570–581, 2004.

[22] A. Demski and M. L. Soria, "ecg-kit: a Matlab toolbox for cardiovascular signal processing," *J. Open Research Software*, vol. 4, 2016.

[23] Martín Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[24] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[25] N. Nagata, R. Candido, and M. T. M. Silva, "Classification of arrhythmias using multilayer perceptron neural networks," in *28th USP International Symposium of Undergraduate Research*, 2020.

[26] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] X. Glorot, and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proc. of AISTATS*, p. 249–256, 2010.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[29] R. A. Fisher, "The statistical utilization of multiple measurements," *Annals of Eugenics*, vol. 8, pp. 376–386, 1938.

[30] C. M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.

[31] S. Osowski, T. Markiewicz, and L. T. Hoai, "Recognition and classification system of arrhythmia using ensemble of neural networks," *Measurement*, vol. 41, pp. 610–617, 2008.

[32] T. Mar *et al.*, "Optimization of ECG classification by means of feature selection," *IEEE Trans. Biomed. Eng.*, vol. 58, pp. 2168–2177, 2011.