



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

AYRTON SOUSA MARINHO

CLASSIFICAÇÃO AUTOMÁTICA DA QUALIDADE DE SINAL DE ECG COM
REDES NEURASIS CONVOLUCIONAIS

CRATEÚS

2021

AYRTON SOUSA MARINHO

CLASSIFICAÇÃO AUTOMÁTICA DA QUALIDADE DE SINAL DE ECG COM REDES
NEURAS CONVOLUCIONAIS

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Crateús da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Orientador: Prof. Msc. Lívio Antônio
Melo Freire

Coorientador: Prof. Msc. Marciel Barros
Pereira

CRATEÚS

2021

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

S696c Sousa Marinho, Ayrton.
Classificação automática da qualidade de sinal de ECG com redes neurais convolucionais / Ayrton Sousa Marinho. – 2021.
46 f. : il.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, , Crateús, 2021.
Orientação: Prof. Me. Lívio Antônio Melo Freire.
Coorientação: Prof. Me. Marciel Barros Pereira.

1. Deep Learning. 2. ECG. 3. Machine Learning. 4. Redes Neurais Convolucionais. I. Título.
CDD

AYRTON SOUSA MARINHO

CLASSIFICAÇÃO AUTOMÁTICA DA QUALIDADE DE SINAL DE ECG COM REDES
NEURAIS CONVOLUCIONAIS

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Crateús da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Msc. Lívio Antônio Melo Freire (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Msc. Marciel Barros Pereira (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Msc. Lisieux Marie Marinho dos Santos
Andrade
Universidade Federal do Ceará (UFC)

Prof. Msc. Amanda Drielly Pires Venceslau
Universidade Federal do Ceará (UFC)

À minha família, e amigos.

AGRADECIMENTOS

Ao Prof. Msc. Lívio Antônio Melo Freire por me orientar em meu trabalho de conclusão de curso.

Ao Prof. Msc. Marciel Barros Pereira , que me auxiliou e teve grande contribuição no processo de idealização e construção do TCC.

À banca examinadora pelo tempo e disposição para ler e compreender o trabalho abordado e fornecer críticas a fim de melhorar o presente projeto.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

Aos meus familiares e amigos, que em momentos em que precisei de ajuda estavam disponíveis para conversar e ajudar no que fosse necessário para alcançar os objetivos, em especial, Saori Pereira da Costa pelo incentivo e apoio.

Agradeço a todos os professores por me proporcionar o conhecimento no processo de formação profissional, e pelo esforço para tentar repassar da melhor forma possível o conteúdo.

“Quando seu temperamento aumentar, abaixe seus punhos - quando seus punhos levantarem, abaixe seu temperamento.”

(Miyagi Chojun)

RESUMO

De acordo com a Organização Mundial da Saúde (OMS) Organization (2017), muitas mortes no mundo ocorrem devido a problemas cardíacos e exames de diagnóstico, como Eletrocardiograma (ECG), são usados para tentar identificar possíveis problemas no coração. Soluções automatizadas vêm sendo testadas através de diversos estudos, nos quais buscam prever, através da análise inteligente de um sinal de ECG, se um indivíduo possui alguma doença cardíaca. Contudo, alguns sinais de ECG são capturados através de dispositivos que, muitas vezes, não possuem eficiência na entrega da qualidade do sinal, gerando, comumente, dados com ruídos ou inconsistências. Este trabalho apresenta como tema central o uso de técnicas de aprendizagem profunda para classificação de qualidade de sinais de ECG em três possíveis categorias: Aceitável, Semi-Aceitável e Não Aceitável. Assim, o presente trabalho busca verificar a viabilidade de uma solução automática, com uso de técnicas de Deep Learning (DL), com foco em Convolutional Neural Network (CNN), para a classificação da qualidade desses sinais, a fim de garantir melhor filtragem de dados de entrada para treinamentos de modelos de previsão de doenças. A metodologia, trata-se de uma pesquisa exploratória com natureza qualitativa, na coleta de dados, foi utilizado uma base de dados já criada especificamente para a tarefa de classificação de sinais de ECG. Como métrica para a classificação do algoritmo é utilizado o F-Score, Precision e Recall obtidos através da técnica de random search em 100 amostras diferentes de modelos. Os resultados foram positivos e indicam um F-SCORE de 0.9270, Precision de 0.9278 e um Recall de 0.9264 para a melhor combinação de hiperparâmetros baseado nos dados utilizados para o treinamento, teste e validação dos modelos.

Palavras-chave: Eletrocardiograma. Deep Learning. Redes Neurais Convolucionais. Machine Learning.

ABSTRACT

According to World Health Organization (WHO) Organization (2017), many deaths worldwide occur due to heart problems and tools, such as, Electrocardiogram (EKG) are used to try to identify possible heart problems. An automated solution has been tested through several studies in which they seek to predict, through an intelligent analysis of a EKG signal, if an individual has any heart disease. However, some EKG signals are captured using devices that are often not efficient in delivering the signal quality that is commonly expressed by noise or inconsistencies in the signal. This paper presents as a central theme the quality classification of EKG signals using Deep Learning techniques to learn the pre-existing signals and classify them in three possible categories: Acceptable, Semi-Acceptable and Not Acceptable. Thus, the present paper seeks to verify the feasibility of using an automatic solution using DL techniques with a focus on CNN to classify the quality of these signals, in order to ensure better filtering and guarantee quality of data for training in disease prediction models. The methodology, it is an exploratory research with a qualitative nature, a database already created specifically for the task of classification of EKG signals was used. As a metric for the classification of the algorithm, the F-Score, Precision and Recall metrics obtained through the random search technique in 100 different samples of models are used. The results were positive and indicate an F-SCORE of 0.9270, Precision of 0.9278 and a Recall of 0.9264 for the best combination of hyperparameters based on the data used for the training, testing and validation of the models.

Keywords: Electrocardiogram. Deep Learning. Convolutional Neural Network. Machine Learning.

LISTA DE FIGURAS

Figura 1 – Exemplo de um sinal ECG	15
Figura 2 – Estrutura de uma rede neural artificial	17
Figura 3 – Estrutura de um neurônio artificial	18
Figura 4 – Principais funções de ativação existentes	19
Figura 5 – Modelo de arquitetura de uma Rede Neural Convolutacional	21
Figura 6 – Processo de convolução	22
Figura 7 – Arquitetura de um filtro extraindo características da imagem de um gato	23
Figura 8 – Exemplo de uma operação de MPL	23
Figura 9 – Exemplo de uma camada fully-connected	24
Figura 10 – Estrutura de uma Rede Neural Convolutacional	26
Figura 11 – Exemplo de um sinal de boa qualidade	31
Figura 12 – Exemplo de um sinal de qualidade ruim	32
Figura 13 – Exemplo de sinal ECG 100001/100001ECG presente no dataset Nemcova Radovan Smisek (2020)	32
Figura 14 – Estrutura da CNN para a classificação de sinais	34
Figura 15 – Arquitetura do modelo para a classificação de sinais	35
Figura 16 – Intervalos ECG	37
Figura 17 – Comparação resultado de modelos F1_Score, Precision, Recall	41
Figura 18 – Comparação resultado de loss nos modelos na base de teste e validação	42
Figura 19 – Comparação resultado de acurácia nos modelos na base de teste e validação	42

LISTA DE ABREVIATURAS E SIGLAS

ANN	Artificial Neural Network
APL	Average Pooling Layers
ATI-CNN	Attention-based Time-incremental Convolutional Neural Network
BIDMC	Beth Israel Deaconess Medical Centre
CC	Camada Convolutiva
CL	Convolutiva Layer
CLs	Convolutivas Layers
CNN	Convolutiva Neural Network
CNNs	Convolutivas Neural Networks
CP	Camadas de Pooling
DDNN	Deep Densely Neural Network
DL	Deep Learning
ECG	Eletrocardiograma
EKG	Electrocardiogram
FC	Filtro Convolutiva
FCs	Filtros Convolutivos
GPL	Global Pooling Layers
IA	Inteligência Artificial
IoT	Internet of Things
MFSC	Mel-frequency Spectral Coefficients
ML	Machine Learning
MPL	Max-Pooling Layer
MSE	Mean Squared Error
NSRDB	MIT-BIH Normal Sinus Rhythm Database
OMS	Organização Mundial da Saúde
ReLU	Rectified Linear Unit
RNA	Rede Neural Artificial
RNC	Rede Neural Convolutiva
RNN	Recurrent Neural Network
SQA	Signal-quality Awareness
WHO	World Health Organization

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Contextualização	13
1.2	Justificativa	13
1.3	Objetivo Geral	14
1.4	Objetivos Específicos	14
1.5	Organização Deste Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Eletrocardiograma	15
2.2	Inteligencia Artificial	16
2.3	Redes Neurais Artificiais	16
2.3.1	<i>Neurônios</i>	17
2.3.2	<i>Pesos</i>	18
2.3.3	<i>Funções de Ativação</i>	18
2.3.4	<i>Função de Custo</i>	19
2.3.5	<i>Regularização em Aprendizagem Profunda</i>	20
2.4	CNN - Convolutional Neural Network	21
2.4.1	<i>Filtros Convolucionais</i>	22
2.4.2	<i>Camadas de Pooling</i>	23
2.4.3	<i>Camada Fully Connected</i>	24
3	TRABALHOS RELACIONADOS	26
4	METODOLOGIA	31
4.1	Obtenção e Pré-Processamento de Dados de Entrada	31
4.2	Arquitetura de Rede Neural	32
4.2.1	<i>CNN</i>	33
4.2.2	<i>Fully Connected</i>	33
4.2.3	<i>Classificação</i>	33
4.3	Metodologia de validação da arquitetura	34
5	ARQUITETURA	35
6	PREPROCESSAMENTO DOS DADOS	37
6.1	Treinamento e Validação	38

7	RESULTADOS	40
7.1	Análise de Precisão	40
8	CONCLUSÕES E TRABALHOS FUTUROS	43
	REFERÊNCIAS	44

1 INTRODUÇÃO

1.1 Contextualização

De acordo com a OMS Organization (2017), doenças do coração são as principais causas de mortes todos os anos, por exemplo, cerca de 17.9 milhões de pessoas morreram de alguma doença cardiovascular no ano de 2016, isso demonstra que essas doenças estão em um nível alarmante de saúde global. Utilizar sinais de ECG para tomar decisões médicas e auxiliar no diagnóstico pode colaborar para mitigar esse número.

A análise de sinais de ECG para a previsão e prevenção de doenças cardiovasculares tem sido tópico de estudos em Inteligencia Artificial (IA) aplicada à medicina, mais especificamente à cardiologia. Com isso, técnicas de previsão de doenças ou reconhecimento de disfunções cardíacas têm sido automatizadas e cada vez mais assertivas em diagnósticos de doenças cardiovasculares.

1.2 Justificativa

Sinais de ECG são dados complexos de serem obtidos e analisados. Dependendo do tipo de coleta, os dados podem ser alterados ou apresentar perdas significativas. A captura desses dados sofre efeitos de ruído⁰ com muita facilidade. Assim, a leitura e interpretação desses dados por sistemas que dependem de entradas com boa qualidade pode ser prejudicada.

É possível encontrar na literatura diversas soluções que classificam doenças cardiovasculares usando técnicas DL visando ao reconhecimento de padrões de sinais de ECG.

Tendo em vista que esses sinais estão sujeitos a ruídos durante o processo de coleta, é necessário verificar se modelos de DL conseguem extrair características capazes de determinar a qualidade do sinal. Portanto, antes de classificar doenças automaticamente, a classificação da qualidade de sinais de ECG precisa ser realizada. Assim, será possível verificar se a qualidade do sinal é aceitável e se ele pode ser utilizado no processo automático de classificação de doenças.

⁰ Ruído no sentido deste trabalho pode ser entendido como irregularidades em um sinal de ECG, tais como falhas de captura ou sinais maiores do que deveriam ser.

1.3 Objetivo Geral

Analisar a aplicação de modelo de DL chamado CNN e avaliar sua capacidade de extrair as características necessárias para classificar a qualidade de sinal de ECG.

1.4 Objetivos Específicos

- Aplicar modelos de CNN e determinar seu desempenho para classificação da qualidade de sinais ECG.
- Avaliar o grau de aceitabilidade de um sinal de ECG em termos de sua qualidade.

1.5 Organização Deste Trabalho

Serão introduzidos conceitos teóricos que fundamentaram este trabalho na seção 2. Na Seção 3, será descrito os trabalhos relacionados ao presente estudo. Na seção 4, é apresentada a proposta do trabalho,

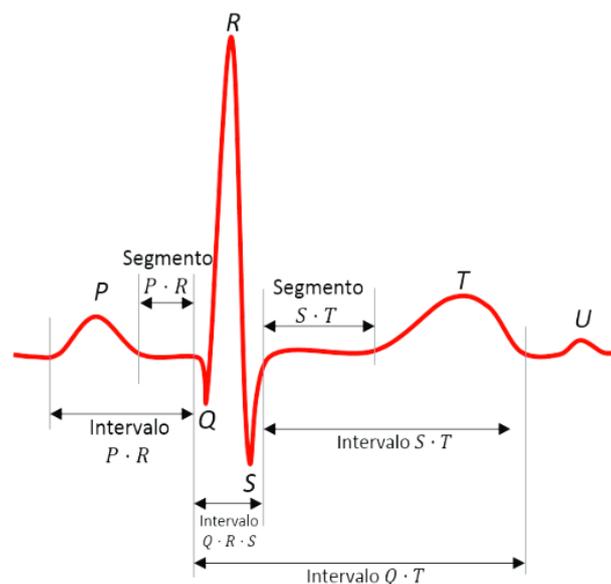
2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentadas as definições e métodos necessários para a compreensão da presente pesquisa. São abordados de forma pontual a seção de Eletrocardiograma, seguido da seção Inteligência Artificial que demonstrarão breves definições. Logo após, será abordada a seção Redes Neurais Artificiais, que compreenderá uma visão geral de como é construída uma Rede Neural Artificial (RNA) do inglês Artificial Neural Network (ANN). Após isso, será demonstrada a base teórica das Convolutional Neural Networks (CNNs) e suas respectivas configurações.

2.1 Eletrocardiograma

Um ECG é um teste que mede atividade cardiovascular de um indivíduo, o funcionamento se dá através de coleta de pulsos elétricos quando os músculos do coração se comprimem e se expandem (ASSOCIATION, 2015). Um sinal de ECG demonstra algumas informações como ondas⁰, o tempo e a intensidade elétrica que o coração produziu. Um profissional de saúde pode observar estes dados e determinar se existe algum fator determinante para saúde do paciente, A figura 1 demonstra um exemplo de um sinal de ECG demonstrando de forma compreensiva como é dividido um sinal de ECG entre intervalos e segmentos.

Figura 1 – Exemplo de um sinal ECG



Fonte: (REGIS *et al.*, 2016).

⁰ Uma onda em um sinal de ECG representa um pulso elétrico gerado pelo batimento cardíaco

Um sinal de ECG é dividido em ondas, essas ondas são identificadas pelas letras: *P, Q, R, S, T, U*. Muitas técnicas de Machine Learning (ML) utilizam intervalos entre essas ondas para tentar prever doenças cardiovasculares. Um exemplo de doença muito discutida é *Fibrilação Atrial*, que é um dos principais objetos de estudos entre os artigos de previsão no segmento de IA descritos na seção 3. A seguir será descrito a seção que descreve sobre IA em que mais detalhes serão descritos para a compreensão deste trabalho.

2.2 Inteligencia Artificial

IA é a ciência e a engenharia de fazer máquinas inteligentes, especialmente programas de computador inteligentes (MCCARTHY, 2012). Está relacionado à tarefa de usar computadores para entender e imitar a inteligência humana, mas a IA não precisa se limitar a métodos biologicamente observáveis (MCCARTHY, 2012).

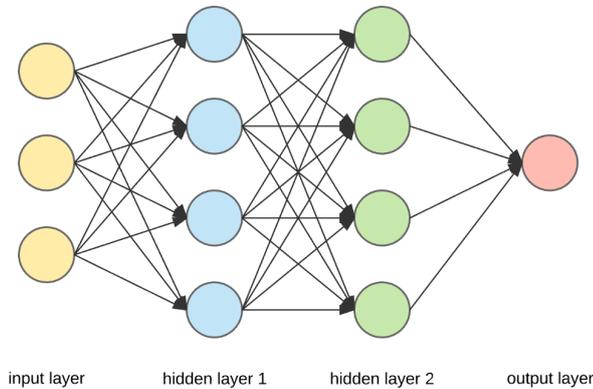
Assim, a medicina tem encarado desafios em modernizar os processos de aquisição de dados, processamento e a aplicação a grande quantidade de conhecimento necessário para resolver problemas complexos (RAMESH *et al.*, 2004). Tem-se hoje diversos tipos de equipamentos Internet of Things (IoT) ou *Wearables*, como relógios inteligentes, capazes de coletar dados de atividade física e batimentos cardíacos (LUI; CHOW, 2018). Uma quantidade enorme assim de dados é inviável para análise humana, dados o tempo e a capacidade de processamento do cérebro. Dito isso, existem algumas tarefas comumente conhecidas que são realizadas por algoritmos inteligentes no ramo da medicina: O reconhecimento de padrões e classificação de dados. Em alguns casos, o diagnóstico é tão preciso quanto o de um médico, como relatado em (HANNUN *et al.*, 2019). Em (RAMESH *et al.*, 2004), o autor cita que as ANN são a técnica mais utilizada nas últimas duas décadas. Atualmente, a ANN é a técnica mais utilizada na medicina para reconhecimento de padrões.

2.3 Redes Neurais Artificiais

Esta seção abordará conceitos importantes para o estudo de redes neurais. Em cada subseção apresenta-se pontos específicos da ANN. Na Seção 2.3.1 são debatidos os conceitos sobre neurônio artificial. Na Seção 2.3.2, são tratados sobre os pesos. Na Seção 2.3.3, são tratadas as funções de ativação e, por fim, a seção 2.3.4 apresenta as funções de custo. Para

finalizar, serão descritas na seção 2.3.5 as técnicas de regularização, no intuito de gerar modelos capazes de generalizar para dados não vistos no treinamento. A seguir é representada uma ANN.

Figura 2 – Estrutura de uma rede neural artificial



Fonte: (OGNJANOVSKI, 2020).

A figura 2 demonstra de forma simplista e direta a arquitetura de uma ANN em que na primeira camada obtêm-se os dados, e então esses valores são repassados para camadas mais internas até por fim obter o resultado da classificação na última camada.

2.3.1 Neurônios

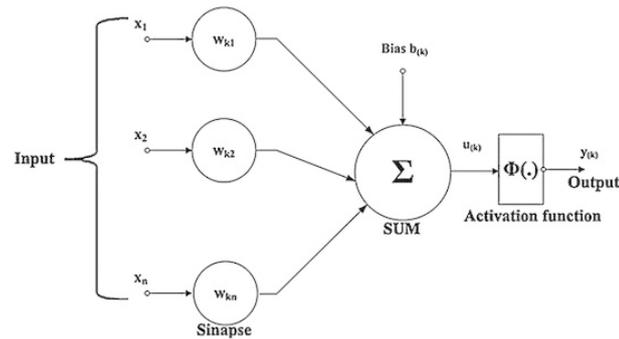
Neurônios (com os *perceptrons*) em uma ANN são baseados nos neurônios cerebrais, em que cada neurônio recebe sinais de outros neurônios. Se a soma dessas entradas elétricas for suficientemente poderosa para ativar o neurônio, ele transmite um sinal ao longo do axônio ¹. Esses neurônios anexados podem então disparar sinais elétricos (CS.STANFORD.EDU, a). O processo em uma ANN é similar. O *perceptron* ² que calcula a soma ponderada dos valores de entrada (CS.STANFORD.EDU, b) e então é aplicada uma função de ativação 2.3.3 no *perceptron* produzindo um novo resultado que é repassado para novos *perceptrons* em camadas mais internas da ANN.

A figura 3 representa um neurônio, é possível perceber que existem cálculos envolvidos, tal como um somatório e uma função de ativação. Ambos posteriormente descritos nas próximas subseções.

¹ É um prolongamento único de uma célula nervosa e passa esse sinal para os outros neurônios

² Um Perceptron é um modelo matemático que recebe várias entradas, x_1, x_2, \dots, x_k e produz uma única saída binária Academy (2021)

Figura 3 – Estrutura de um neurônio artificial



Fonte: (OGNJANOVSKI, 2020).

2.3.2 Pesos

A ANN utiliza pesos para nivelar o conhecimento e encontrar o conjunto ótimo de valores que maximizam o seu aprendizado. Os pesos são um conjunto de valores $W = \{w_i\}$: $w_i \in \mathbb{R}$. A sua função é transformar os dados de entrada a fim de refinar o aprendizado. Os w_i são inicializados com valores arbitrários e sofrem atualizações no processo de treinamento. A equação 2.1 representa a saída de um neurônio com base no produto das entradas x_i , em que $x_i \in X$ e $x_i \in \mathbb{R}$ por seus respectivos pesos w_i , na forma a seguir:

$$\sum_{i=1}^n x_i \cdot w_i \quad (2.1)$$

A equação descrita acima define a soma dos produtos dos valores de entradas, dados pela variável x , e por seus respectivos pesos, representados pela variável w . O resultado desta operação é então repassado à função de ativação do neurônio.

2.3.3 Funções de Ativação

Funções de ativação recebem a soma, aplicam uma transformação e retornam o valor de ativação para aquele determinado neurônio. Existem muitas funções de ativações, sendo a mais simples a *Step Function*. Essa é função geralmente relacionada a problemas binários: resulta em ativação 1, se $soma \geq 1$; resulta 0 caso contrário. Dentre as funções de ativação existentes, destacam-se:

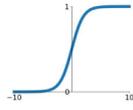
- *Rectified Linear Unit (ReLU)* é uma função de ativação muito utilizada, e o seu funcionamento é simples. É verificado se o valor da entrada é menor ou igual a 0, caso afirmativo,

Figura 4 – Principais funções de ativação existentes

Activation Functions

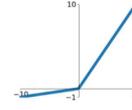
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



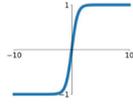
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

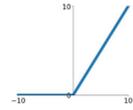


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

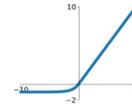
ReLU

$$\max(0, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Fonte: (CAI *et al.*, 2020)

então é retornado 0, caso contrário a função retorna o próprio valor.

- *Softmax* é uma função de ativação muito utilizada na última camada da rede neural quando o problema proposto possui características de classificação multi-classe. Esta função retorna uma lista de valores de probabilidades para cada classe.

$$\text{softmax}(x) = \frac{e^{x_i}}{\sum_{j=1} e^{x_j}} \quad (2.2)$$

2.3.4 Função de Custo

Funções de Custo ou *Loss Functions* são funções utilizadas no processo de minimizar o erro ³ de um modelo . O seu resultado é usado em uma técnica chamada *Backpropagation* ⁴, que efetua novos cálculos de pesos a fim de melhorar o aprendizado de uma ANN. Existem algumas funções de custo populares em *ML*, como *Cross Entropy Loss* e *Mean Squared Error (MSE)*.

A *MSE* é uma função de custo usada em *ML* para problemas de regressão. $X = x_i : x_i \in \mathbb{R}$ e representa os valores preditos pelo modelo. Já $Y = y_i : y_i \in \mathbb{R}$ refere-se valores de referência, ou seja, o valor que o modelo deveria prever. O cálculo da *MSE* é mostrado na equação 2.3:

$$MSE = \frac{1}{N} \sum_{i=1} (y_i - x_i)^2 \quad (2.3)$$

³ Um erro, ou também chamado de Loss é uma penalização que o modelo sofre por ter feito uma previsão ruim do resultado.

⁴ Backpropagation se refere à atualização de pesos de uma rede neural a fim de otimizar o modelo e guiar para a melhor solução

Além dessa, existe a função chamada *Cross Entropy Loss*, que é usado para problemas de classificação. O cálculo da *Cross Entropy Loss* pode ser verificado na equação 2.4:

- θ_z = probabilidade de ser uma determinada classe X no vetor de predição.
- y = probabilidade de ser uma determinada classe X no vetor principal.

$$C = - \sum_{i=1}^N y_i \log \theta_z \quad (2.4)$$

2.3.5 Regularização em Aprendizagem Profunda

Um problema central no aprendizado de máquina é como fazer um algoritmo ter desempenho aceitável não apenas nos dados de treinamento, mas também em novas entradas. Muitas estratégias usadas no aprendizado de máquina são explicitamente projetados para reduzir o erro de teste, possivelmente às custas do aumento do erro de treinamento. Essas estratégias são conhecidas como regularização (GOODFELLOW *et al.*, 2016).

Por vezes, torna-se necessário regularizar o modelo de ML buscando reduzir problemas de *Overfitting*⁵. Dentre algumas técnicas tem-se como destaque a técnica conhecida como *Dropout*.

Dropout é a técnica de regularização proposta para lidar com o problema de *Overfitting* (GOODFELLOW *et al.*, 2016). Essa técnica consiste em retirar determinada quantidade de neurônios de uma certa camada da ANN. Ao remover essas unidades, novo treinamento da ANN é exigido, para atualizar os pesos sem considerar as unidades retiradas. Assim, durante o processo de treinamento, reduz-se a possibilidade dos pesos da rede memorizarem os dados de entrada. Fica a cargo do responsável pela arquitetura da ANN definir quantas unidades e quais camadas passarão pelo processo. De maneira geral, em situações reais, o *Dropout* é representado pelo uso de uma porcentagem de neurônios de cada camada que deverão ser retiradas para lidar com o problema de *Overfitting*.

Apresentada o que é uma regularização em um modelo de aprendizado de máquina, torna-se necessário avançar e apresentar o conceito por trás das CNN

⁵ é um problema de superajuste do modelo perante os dados. Este ocorre quando o espaço entre os erros de treinamentos e erros de testes são muito grandes (GOODFELLOW *et al.*, 2016). Em outras palavras, o modelo passa a ficar viciado nos dados de treinamento. Assim, se for disponibilizado alguma informação que seja diferente do que o modelo foi treinado, ele terá um desempenho inferior.

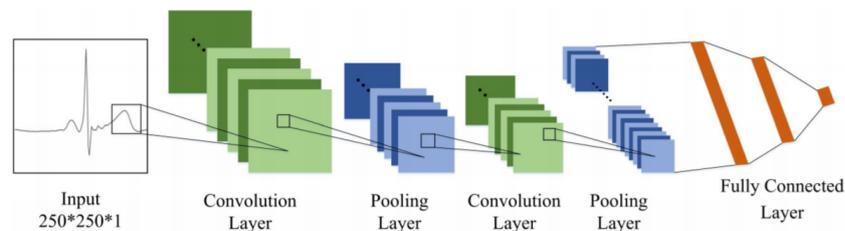
2.4 CNN - Convolutional Neural Network

Problemas como reconhecer se uma dada imagem é um gato ou um cachorro, é um problema complexo que depende de muitas Convolutional Layers (CLs) para extrair diferentes tipos de características da imagem. No campo da saúde, essa abordagem é usada, dentre diversos objetivos, o reconhecimento de cânceres ou problemas do coração, ao aprender características de imagens de tumores, ou batimentos cardíacos através de sinais de ECG.

CNN é um tipo de rede neural *feed-forward* que tem obtido um bom desempenho no aprendizado de problemáticas que envolvem imagens, sons ou textos. Além disso, elas são utilizadas em diversas outras tarefas que aplicam a classificação, também conhecidas como ConvNets, as CNN trabalham de maneira sistematizada com vetores multidimensionais chamados Filtros Convolucionais (FCs) utilizada.

A figura 5 representa uma CNN no desenvolvimento para a solução do problema proposto de análise de qualidade de sinais ECG. É possível observar inúmeros

Figura 5 – Modelo de arquitetura de uma Rede Neural Convolutional



Fonte: (HUANG; WU, 2020).

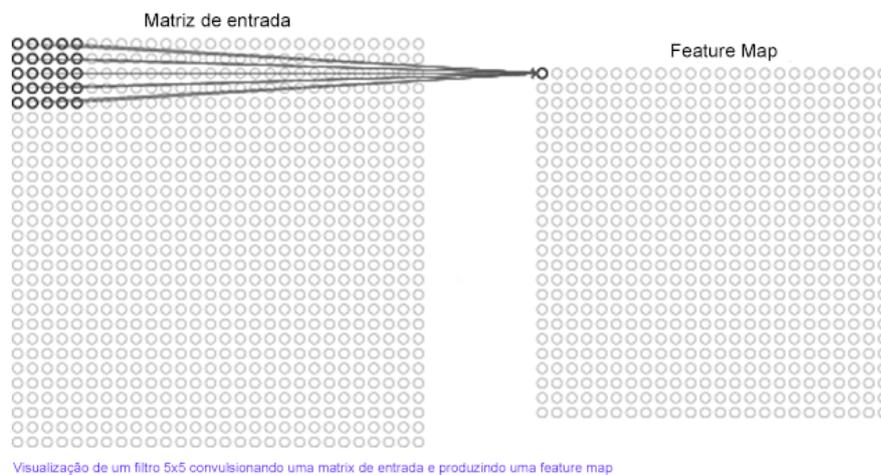
Nesta abordagem, geralmente são encontrando padrões na imagem ou amostras de áudio como resultado da classificação. Dessa forma, ela busca extrair características únicas para cada entrada dada. É comum, também, algumas tarefas de reconhecimento de sinais, sejam audios ou imagens. Ela também conta com o auxílio dos FCs e outras técnicas de otimização computacional. Além disso, cada Convolutional Layer (CL) possui seu Filtro Convolutional

(FC) associado.

2.4.1 Filtros Convolucionais

Os filtros em uma CNN tem como objetivo transformar os dados de entrada de um neurônio. Os dados de entrada são geralmente matrizes multidimensionais de dados. Para realizar o reconhecimento de características específicas de um conjunto de dados (GOODFELLOW *et al.*, 2016). Tal filtro, conhecido também como *kernel*, é geralmente uma matriz multidimensional de parâmetros que são adaptados pelo algoritmo de aprendizado (GOODFELLOW *et al.*, 2016). O produto entre as matrizes resulta em um valor escalar, que será inserido na posição da representação da imagem de entrada denominada *feature map*. Após o primeiro cálculo, o filtro se move para a direita ou para baixo, selecionando mais uma seção de mesma dimensão do filtro, realizando nova operação e gerando novo valor. Esse processo é chamado de *Convolução*. A figura 6 demonstra como funciona o processo de convolução, multiplicando por um filtro e gerando um feature map.

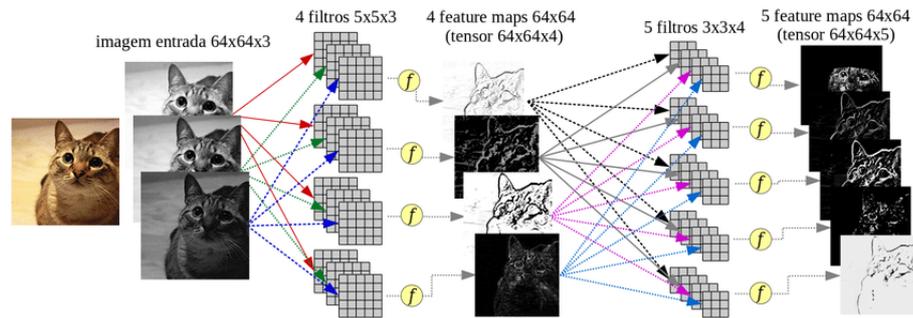
Figura 6 – Processo de convolução



Fonte: (DESHPANDE, 2016).

Após toda imagem ter sido convoluída com o novo filtro, uma nova entrada será calculada e o processo continua para a Camadas de Pooling (CP). A figura 7 demostra a extração de características de um gato sendo retirada a cada camada convolucional.

Figura 7 – Arquitetura de um filtro extraindo características da imagem de um gato



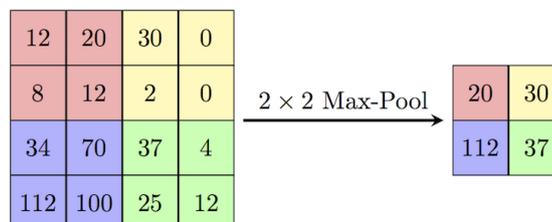
Fonte: (PONTI; COSTA, 2018).

2.4.2 Camadas de Pooling

CNNs possuem necessariamente CLs, dessa forma as entradas de cada CL são aplicadas devidos filtros iniciados de forma arbitrária produzindo, assim, uma nova saída a partir dos dados de entrada. Existem alguns tipos de camadas de *pooling* como Average Pooling Layers (APL), Global Pooling Layers (GPL), e por fim a Max-Pooling Layer (MPL), (BROWNLEE, 2019)

A MPL reduz o tamanho da saída após o filtro aplicado na matriz de entrada da CL assim como todas as camadas de *pooling*. Nesta abordagem, cria-se um novo filtro, geralmente uma matriz de tamanho 2, e também é definido uma quantidade de movimentos que o filtro de *pooling* faz na imagem de saída do FC é o que leva o nome de *Stride*. Então, o filtro de *pooling* é iniciado selecionando uma região de tamanho 2x2 da saída da matriz da CL, verificando dentre todos os valores qual maior, sendo portanto, alocado em uma nova matriz de saída.

Figura 8 – Exemplo de uma operação de MPL



Fonte: (COMPUTERSCIENCEWIKI, 2018).

No exemplo da figura 8, é usado um filtro de tamanho 2x2, cada quadrante é uma operação de *Stride* = 2 pois o filtro seleciona a cada duas linhas, ou 2 colunas, um novo conjunto. Sendo assim, a matriz da esquerda possuía tamanho 4x4 após a aplicação da MPL, a imagem foi

reduzida em 4 vezes.

Após diversas reduções de dimensão na imagem, é chegada o momento de treinar propriamente a ANN, passando, portanto, para a Camada Fully Connected.

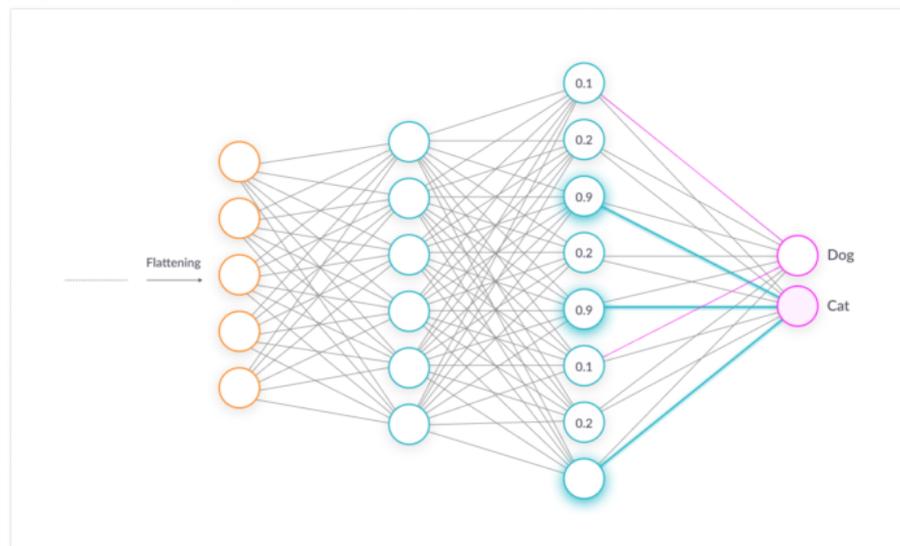
2.4.3 Camada Fully Connected

Esta camada é uma ANN de tipo *Feed-forward*, da qual é possível alcançar a aprendizagem de um determinado problema. Esse tipo de denominação se dá pelo fato que nesta camada, todos os neurônios são conectados aos neurônios da camada mais interna. Nas CNNs, elas se localizam após as CLs, ou de pooling, tomando assim os dados de saída destas camadas como dados de entrada para si.

O processo conhecido como *Flatten* converte a matriz de saída da camada de *pooling*, em vetores. Cada um desses vetores contribui com a alimentação da rede, e cada valor dos vetores sofre operações matemáticas de multiplicação por pesos em cada neurônio. A aplicação da função de ativação resulta em um valor que, por sua vez, será propagado para as camadas internas.

A figura 9 é mostrada como é arquitetada uma camada fully connected, onde todos os nós se ligam a todos os nós da camada à frente, repassando os cálculos entre camadas e neurônios.

Figura 9 – Exemplo de uma camada fully-connected



Fonte: (MISSINGLINK.AI,)

O algoritmo *Backpropagation* é utilizado como forma de recalculer os valores dos pesos de todas as camadas da rede e, assim, refinar a aprendizagem da rede por um determinado

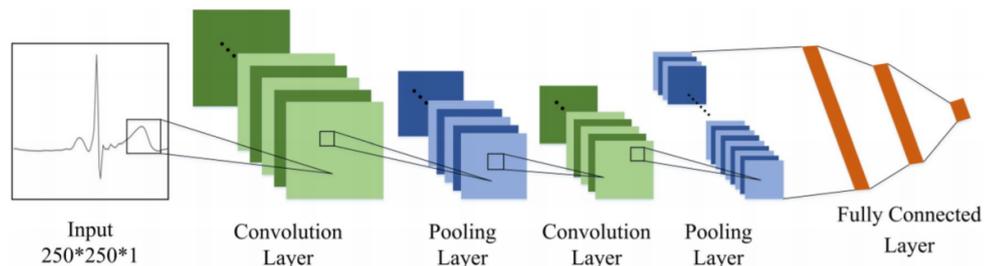
número de *épocas* . No término do período de aprendizagem, é aplicada na última camada, uma função de ativação diferente das demais já aplicadas, geralmente as funções *softmax* ou *sigmoid*, pois estas funções geram um conjunto de valores probabilísticos para serem usados na classificação.

3 TRABALHOS RELACIONADOS

Este capítulo discorre sobre trabalhos relevantes que abordaram o tema de reconhecimento de padrões usando uma ANN. Com isso, serão descritas brevemente suas respectivas soluções para os temas propostos.

Huang e Wu (2020) usa uma combinação dos banco de sinais MIT-BIH AF e o banco MIT-BIH NSR. A base de dados MIT-BIH AF como verificado pelo autor possui 25 sinais onde 2 não foram utilizados por conterem apenas o ritmo (.atr) e os batimentos inaudíveis (.qrs), com isso ele usou 23 sinais válidos dentro deste banco. Assim, a classificação é iniciada utilizando uma CNN e como entrada uma imagem de sinal de ECG é servida para a rede neural.

Figura 10 – Estrutura de uma Rede Neural Convolucional



Fonte: Huang e Wu (2020).

Assim como no trabalho anterior, Hannun *et al.* (2019) utiliza uma CNN para a classificação de sinais de ECG. O trabalho busca classificar 12 ritmos baseados em seus eletrocardiogramas ambulatoriais e então comparar com previsões de cardiologistas. Os autores utilizam uma nova base de dados com maior tamanho para a classificação de todos os 12 ritmos cardíacos. O pré-processamento de imagens é dispensado e dado um sinal ECG como entrada o algoritmo consegue dar probabilidades dos diagnósticos usando uma CNN de 34 camadas. Dessa forma, o trabalho de Awni Y. Hannun demonstra que o uso de uma CNN pode ser incorporada no presente trabalho. Em outro trabalho relacionado a IoT Deepu *et al.* (2016) desenvolveram uma nova abordagem para redução de energia usando compressão de dados em sensores de monitoramento

sem fio para sinais ECG, com isso reduzindo custos em cuidados de saúde e possibilitando que levam a uma qualidade de vida melhor. Neste trabalho, o autor usa uma forma de compressão de dados chamado *fan algorithm*, um algoritmo utilizado para compressão de sinais ECG para redução de espaço e possibilitar um menor custo nos envios dos dados via *bluetooth*. Este algoritmo pode ser útil para o presente trabalho, devido a compressão de sinais possibilitar uma transmissão mais eficiente destes dados através de dispositivos IoT Cai *et al.* (2020). Utiliza uma Deep Density Neural Network (DDNN) para a distinção de Atrial fibrillation e outras anormalidades. Foram combinados de acordo com o autor 3 base de dados diferentes que são 12-lead ECG datasets, Chinese PLA General Hospital (301 Hospital), um dispositivo wearable ECG (Mason-Likar ECG 12-lead system, esse dispositivo incluem 4 terminais de membro integrados em uma faixa no peito) (CardioCloud Medical Technology (Beijing) Co. Ltd); e outros 11 hospitais (The China Physiological Signal Challenge 2018). O autor utiliza uma CNN para o treinamento dos sinais de ECG. Por também utilizar CNN para reconhecimento de sinais de ECG este trabalho contribuirá positivamente para o presente trabalho. Ele utiliza o conceito de blocos de CNN dentro da DDNN, foi mostrado uma forte capacidade de generalização para o reconhecimento de anormalidades cardíacas Acharya *et al.* (2019).

Os autores apresentaram uma rede neural convulacional de 11 camadas para a classificação de CHF. Diferente de alguns trabalhos citados acima, este usa o artifício de um pré-processamento dos sinais ECG. O presente trabalho, com o intuito de verificar se é possível, usando uma CNN, classificar a qualidade de sinais. Este artigo pode contribuir para este trabalho como uma etapa anterior a de treinamento eliminando a necessidade de pré-processamento de sinais. De acordo com os autores, "os sinais ECG usados neste trabalho foram obtidos de banco de dados públicos: PhysioBank, Beth Israel Deaconess Medical Centre (BIDMC) Congestive Heart Failure Database, Fantasia Database e MIT-BIH Normal Sinus Rhythm Database (NSRDB)" Acharya *et al.* (2019). No trabalho de Baloglu *et al.* (2019) é utilizado técnicas de DL para a classificação de infarto do miocárdio. Os autores do trabalho utilizaram uma CNN com 10 camadas para o treinamento e com o dataset 12-Lead ECG data, previamente reportado nos trabalhos Hannun *et al.* (2019) Cai *et al.* (2020), eles afirmam que ao contrário de de criar um modelo para cada classe, foi instanciado um modelo mais flexível que reconheceria os sinais de infarto de miocardio para todos os sinais. No trabalho de Lui e Chow (2018), os autores buscam a classificação de sinais ECG de dispositivos portáteis, e que possuem muito ruído devido a

baixa qualidade dos equipamentos de captura ao ser comparado a equipamentos hospitalares. É utilizado uma CNN em combinação com uma Recurrent Neural Network (RNN) para realizar esta tarefa, porém o número de dados é limitado. Em outros trabalhos como Hannun *et al.* (2019), Cai *et al.* (2020) Baloglu *et al.* (2019) são utilizados sinais de 12-leads, porém dispositivos portáteis conseguem produzir de 3 a 4 leads, alguns apenas 1. O presente trabalho se caracteriza como uma abordagem de classificação multi-classe de qualidade de sinais, e o texto do trabalho de Lui e Chow (2018) demonstra a possibilidade de um caminho a ser seguido. No trabalho também utilizam de técnicas de redução de ruídos para melhor classificação dos sinais.

Em Fotiadou *et al.* (2020) é apresentada uma técnica com CNN de apurar um sinal de eletrocardiograma de um feto. Sinais ECG de fetos possuem diversos ruídos entre eles a idade de gestão, posição dos eletrodos e o próprio corpo da mãe. Neste trabalho, é utilizado uma CNN para reduzir o ruído de um sinal ECG. A arquitetura é dividida em duas partes, uma parte chamada de Encoder e a outra Decoder, cada uma com 8 camadas convolucionais. De acordo com o autor essa redução de ruídos em fetos é de extrema importância para a prevenção e previsão de possíveis problemas com o feto. O trabalho Maknickas e Maknickas (2017) apresenta uma forma de reconhecimento de sinais sonoros de batimentos cardíacos para classificação de sinais normais e anormais. É utilizada uma abordagem usando redes neurais convolucionais, dividindo os sinais em sinais normais e anormais, com 70% para treinamento 30% para teste, o que também será uma característica importante para o presente trabalho. Utiliza-se Mel-frequency Spectral Coefficients (MFSC) como uma forma de divisão do sinal fonocardiaco e assim possibilitar que pedaços do sinal sejam melhor interpretados.

O trabalho Satija *et al.* (2017) usa sinais recebidos de dispositivos IoT e informa a importância de Signal-quality Awareness (SQA) tem um papel importante que incluem reconhecimento automático de arritmias anteriormente à análise do ECG para o diagnóstico de doenças. A qualidade do sinal neste trabalho é classificada em 3 classes: Bom, intermediário e ruim. Este tipo de método usa dois passos importantes que são extração de *features* e avaliação de qualidade de sinal usando regras heurísticas, e abordagens usando ML. Segundo os autores, neste artigo o ECG SQA é implementado baseado na Transformada Discreta de Fourier, *turning points* e *decision rules*. As fases demonstradas no trabalho são: Remoção de desvio de linha de base e detecção abrupta de alterações, Detecção de ausência de sinal de ECG, Detecção de ruído HF, Classificação da qualidade do sinal de ECG. O trabalho Yao *et al.* (2020) em questão é

similar aos outros que buscam a classificação de sinais de ECG usando 12-lead, contudo este trabalho afirma que CNNs não conseguiam aceitar sinais ECG de tamanhos variados e tinham performances limitadas em detectar arritmia paroxísticas. Assim, no trabalho, o autor decidiu modificar a CNN para uma Attention-based Time-incremental Convolutional Neural Network (ATI-CNN), um modelo de DL que alcança uma fusão espaço-temporal das informações do sinal de ECG. O dataset utilizado para o treinamento pertence ao "1st China Physiological Signal Challenge". Em relação ao pré-processamento, o autor afirma que por motivos de sua base de dados ser relativamente pequena em comparação outros trabalhos que usam DL diversas estratégias de aumento de dados foram utilizadas no treinamento de modelos.

Nemcova *et al.* (2020) é um estudo sobre compressão de sinais de ECG especialmente em transmissão. Um novo algoritmo de compressão é demonstrado nesse trabalho, mas este usa o mesmo dataset que o presente trabalho busca utilizar.

A tabela abaixo indica a quantidade de uso de cada dataset nos trabalhos

1. MIT-BIH
2. Physiobank (PTB) ECG
3. 12-lead ECG dataset
4. Chinese PLA General Hospital
5. 11 hospitals (The China Physiological Signal Challenge 2018)
6. Wearable ECG devices
7. 1st China Physiological Signal Challenge
8. Real-time signal quality-aware ECG telemetry system for IoT-based health care monitoring
9. The University of Haute Alsace heart sounds database (UHAHSDB)
10. Brno University of Technology ECG Quality Database (BUT QDB)

Datasets	Artigos
1	HUANG; WU, HANNUN <i>et al.</i> , DEEPU <i>et al.</i> , ACHARYA <i>et al.</i> , SATIJA <i>et al.</i>
2	LUI; CHOW
3	HANNUN <i>et al.</i> , CAI <i>et al.</i> , BALOGLU <i>et al.</i>
4	CAI <i>et al.</i>
5	CAI <i>et al.</i>
6	LUI; CHOW
7	YAO <i>et al.</i>
8	SATIJA <i>et al.</i>
9	MAKNICKAS; MAKNICKAS
10	NEMCOVA <i>et al.</i>

4 METODOLOGIA

Usando os *datasets* que foram aplicados em diversos trabalhos como (MIT-BIH) Huang e Wu (2020), Hannun *et al.* (2019), Deepu *et al.* (2016), Acharya *et al.* (2019), Satija *et al.* (2017) e (12-lead ECG Dataset) & Hannun *et al.* (2019), Cai *et al.* (2020), Baloglu *et al.* (2019), e o mais recente dentre os *datasets* criado por Nemcova Radovan Smisek (2020). Pretende-se investigar se uma rede neural convolucional é capaz de extrair as características necessárias para classificar a qualidade de sinal de ECG. O processo utilizado neste trabalho abordará o tipo de técnica conhecida como Aprendizagem Supervisionada.

4.1 Obtenção e Pré-Processamento de Dados de Entrada

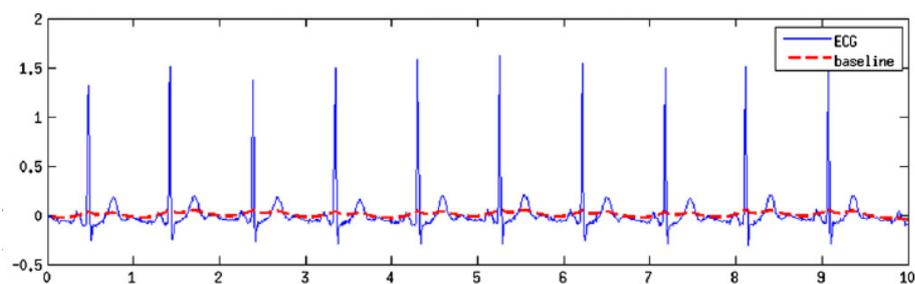
O dataset que será utilizado é o de Nemcova Radovan Smisek (2020) um dataset exclusivo de qualidade de sinais de ECG, contendo 3 classes bem definidas.

Na Figura 11 e Figura 12 são apresentados sinais a serem classificados em termos da qualidade. Observe que o sinal representado pelo quadrante superior possui uma qualidade que representa a sua aceitação para diagnóstico enquanto o quadrante abaixo ilustra um sinal de baixa qualidade para diagnóstico.

Será utilizado um conjunto de sinais ECG representados por vetores unidimensionais, previamente rotuladas nas classes a seguir:

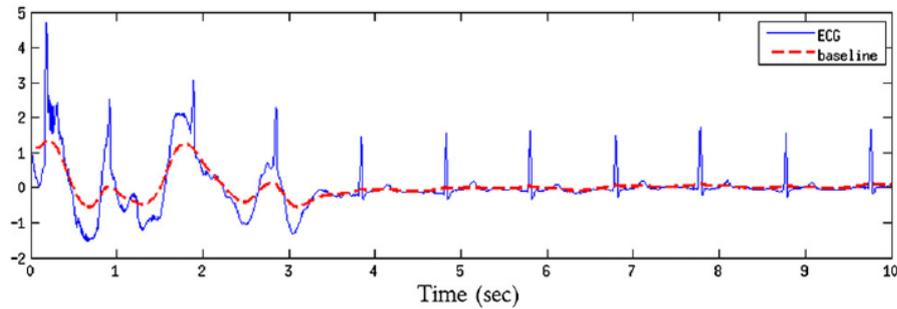
- **Classe 1:** Sinais que apresentam boa qualidade para diagnóstico médico;
- **Classe 2:** Sinais que apresentam qualidade inferior que os da Classe 1, no entanto, que ainda podem ser utilizados para diagnósticos;
- **Classe 3:** Sinais que apresentam baixa qualidade e, portanto, não podem ser utilizados para diagnóstico.

Figura 11 – Exemplo de um sinal de boa qualidade



Fonte: Clifford *et al.* (2012)

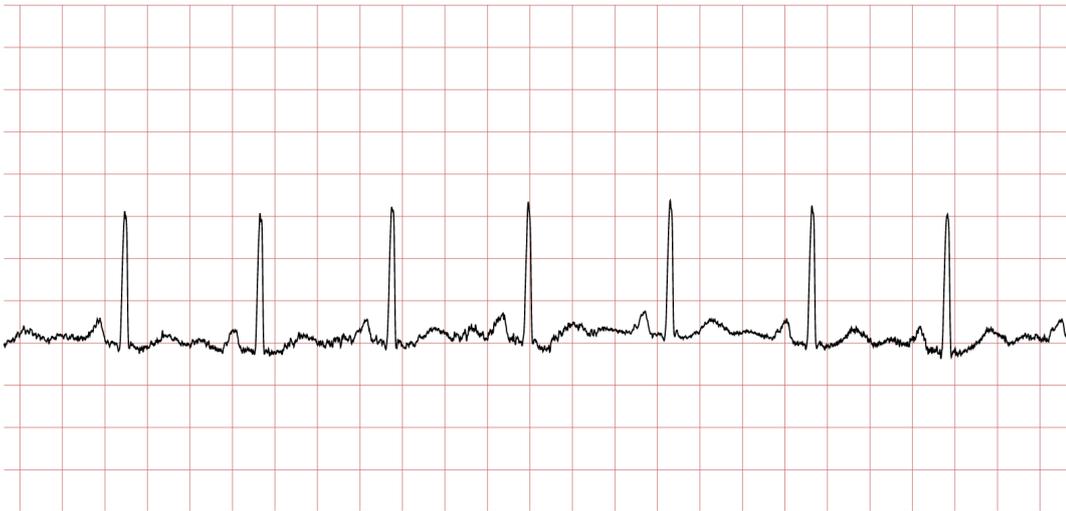
Figura 12 – Exemplo de um sinal de qualidade ruim



Fonte: Clifford *et al.* (2012)

A Figura 13 exibe um exemplo de um sinal de ECG de entrada retirada diretamente da base de dados que será utilizada neste trabalho.

Figura 13 – Exemplo de sinal ECG 100001/100001ECG presente no dataset Nemcova Radovan Smisek (2020)



Fonte: Clifford *et al.* (2012)

4.2 Arquitetura de Rede Neural

Pode-se separar essa arquitetura em 4 processos a fim de entendê-los separadamente. No primeiro processo temos a camada de entrada, em que irá receber vetores unidimensionais. No segundo processo, inicia-se a convolução, a fim de determinar características do ECG dentro do dataset. Esta fase irá conter inicialmente 3 camadas de convolução. O terceiro desta arquitetura está em através do resultado obtido até o segundo processo, passá-lo para uma rede neural fully connected, em que ocorrerá todos cálculos, necessário para o aprendizado. Na quarta, e última fase, será utilizada uma função de ativação representada pela equação 2.2.

4.2.1 CNN

Para representação dos sinais nas classes propostas na Seção 4.1, será utilizada uma arquitetura CNN, como descrita na Seção 2.4. Nas camadas convolucionais serão ativados os FCs de ordem 1×3 . Selecionando um array $M \in B$ em que B é a entrada da camada anterior e seja F o filtro da primeira camada convolucional de ordem 1×3 , então o *feature map* será calculado por:

$$\text{feature map} = M \cdot F \quad (4.1)$$

Para cada combinação com a matriz M serão gerados novos valores que, formarão a *feature map*, que irá produzir entrada para a camada de *pooling*. Será utilizada a MPL reduzindo a dimensionalidade dos dados de entrada. Seleciona-se uma região de tamanho 1×2 na *feature map*, e por sua vez, será definido o *Stride* que indica a variação dos valores da *feature map*. Essa operação irá reduzir o tamanho do sinal, produzindo um melhor custo computacional para os calculos mais internos.

De acordo com Hannun *et al.* (2019) e Huang e Wu (2020) o uso de 3 camadas convolucionais é suficiente para a tarefa de classificação de sinas de ECG.

4.2.2 Fully Connected

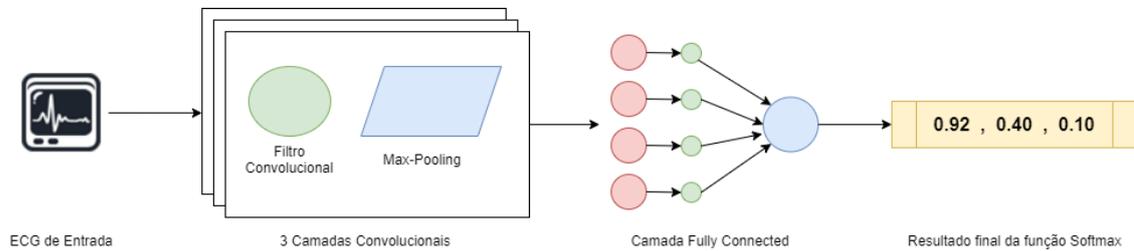
Esta etapa contará com pelo menos 10 (dez) camadas internas, que se trata de uma quantidade aceitável de camadas, de acordo com estudos como (HANNUN *et al.*, 2019) e (HUANG; WU, 2020). Contudo, como se trata de outro problema, estas camadas participarão de treinamento e teste a fim de que o próprio algoritmo decida entre as opções disponíveis qual a que trará resultados positivos para a abordagem tratada neste trabalho. É também importante afirmar que as camadas escondidas utilizarão como ativação a função ReLU, com exceção da última camada que utilizará a função de ativação *Softmax* representada na equação 2.2

4.2.3 Classificação

O problema de reconhecimento de qualidade de sinal como aceitável ou não aceitável, se trata de um problema de classificação binária, portanto, seria utilizada a função *Sigmoid*. No entanto, a função *Softmax* será utilizada, pois no presente trabalho o objetivo é classificar um

total de três classes, para que sejam definidas probabilidades para todas as classes.

Figura 14 – Estrutura da CNN para a classificação de sinais



Fonte: Autoria Própria

O presente trabalho utilizará a função *Cross Entropy Loss* no treinamento do modelo, a razão para isso é que para problemas de classificação, *Cross Entropy Loss* é uma boa escolha função de custo, diferentemente do MSE que é mais utilizado para problemas de regressão. Como dito anteriormente θ_z é probabilidade de ser uma determinada classe X no vetor de predição, dessa forma, θ_z é a soma ponderada das entradas somado ao viés. (DSA,). Quando consideramos $(1 - \theta_z)$ indicamos que está sendo verificando a probabilidade da classe não está no vetor de predição, já $(1 - y_i)$ se refere a probabilidade de uma classe não estar no vetor principal. Dessa forma a junção dessas partes pode ser lido como calcular a probabilidade da classe estar no vetor principal, a probabilidade da classe estar no vetor de predição menos a probabilidade de não estar na classe principal e nem estar no vetor de predição.

4.3 Metodologia de validação da arquitetura

O conjunto de dados, de maneira geral, precisa ser devidamente separado para algumas fases, são elas: Treinamento, Teste e Validação. Alguns problemas não exigem tanto, separando em apenas duas fases: Treinamento e teste. Como este trabalho envolve uma boa precisão, este terá 3 fases.

- No treinamento serão separados 70% dos dados, estes serão separados exclusivamente para o treino da rede neural.
- Na fase de teste e validação serão separados 15% dos dados, assim, se passar nos testes ainda ocorrerá uma nova fase que é a de validação. Nesta, será utilizado a técnica *K-Fold Cross-validation* que é responsável por avaliar o quão eficiente é a generalização do modelo de ML.

5 ARQUITETURA

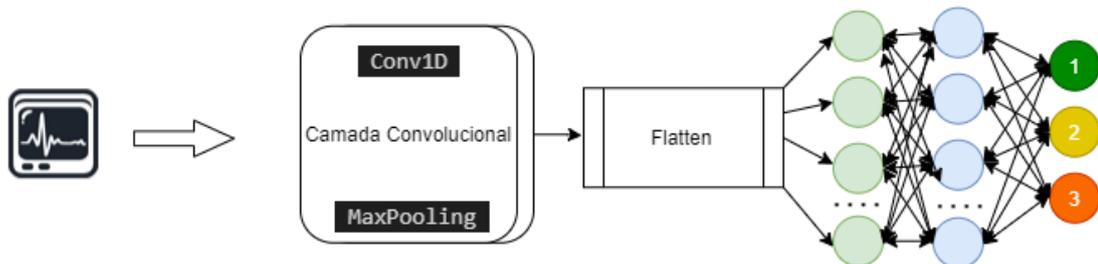
Neste capítulo serão apresentados aspectos relativos à arquitetura proposta para o modelo de aprendizagem, considerando o tipo de estratégia e os principais parâmetros que devem ser utilizados

A construção do modelo utilizou a separação entre dados de treinamento, teste e validação. A tabela a seguir demonstra quantitativamente como foram separados os dados.

Tabela 1 – Divisão dos Dados

	Porcentagem
Treinamento	80%
Teste	10%
Validação	10%

Figura 15 – Arquitetura do modelo para a classificação de sinais



Fonte: Autoria Própria

A figura 15 descreve o modelo de DL do presente trabalho foi definido com as seguintes características: O trabalho utilizou a arquitetura de uma Rede Neural Convolutiva (RNC), em que foi utilizado uma Camada Convolutiva (CC) uni-dimensional, seguida de uma MPL uni-dimensional finalizando aplicando a operação de achatamento do inglês *Flatten*. Para a camada *Fully-connected*, foi utilizada uma arquitetura simples, em que possui apenas uma camada escondida, uma camada extra do tipo *LeakyReLU* e por fim a camada de saída responsável por determinar a classificação do sinal.

O modelo foi compilado utilizando alguns hiperparâmetros fixos. Abaixo segue uma tabela de valores utilizados para os seus respectivos hiperparâmetros:

O modelo propõe resolver o problema de que dado um sinal de ECG ele classifique pedaços de tamanhos específicos apresentando uma previsão para cada um dos intervalos. Então,

Hiper parâmetro fixado	valor
Optimizer	Adam
Learning Rate	0.0030000006482499
Decay	0.0003000006482499
Clipvalue	0.23
Loss	Sparse Categorical Crossentropy
Metrics	Sparse Categorical Accuracy

dado um sinal de 1 minuto, o modelo entregará uma lista de previsões para cada segundo do sinal.

6 PREPROCESSAMENTO DOS DADOS

Como dito previamente na seção de cenário deste capítulo, os dados não possuíam um formato apropriado para tarefas de aprendizado de máquina, assim, observou-se a necessidade de redefinir os intervalos e suas classificações.

Cada ECG possui 4 anotadores (do inglês *annotators*), em que possuem 3 colunas cada. A primeira coluna informa o início do intervalo, já a segunda coluna informa o fim do intervalo da qualidade do sinal. A terceira coluna, informa a classificação do intervalo em classes de tipo 0, 1, 2, 3. Classes do tipo 0, foram ignoradas por se tratarem de intervalos que não foram possíveis obter uma classificação apropriada.

Figura 16 – Intervalos ECG

1	47323379	3	1	47323871	3	1	47323500	3	1	47323500	3
47323380	47388207	2	47323872	47329547	1	47323501	47323539	1	47323501	47326788	2
47388208	47404719	1	47329548	47333899	2	47323540	47326788	2	47326789	47329547	1

Fonte: https://physionet.org/content/butqdb/1.0.0/105001/105001_ANN.csv

Como é possível ver na figura 16, cada *annotator* criaram diferentes intervalos, e em um primeiro momento foi decidido utilizar esses dados já criados, usando, portanto, os intervalos já disponíveis. Contudo, problemas emergiram após a tentativa de carregar os dados, os intervalos eram muito grandes e eram em maior parte do tempo de tamanho diferentes. Foi necessário criar um novo dataset a partir desses intervalos de cada classificação dada para cada *annotator*.

Como solução para o problema de tamanhos diferentes, foi elaborado uma solução para criação de um novo dataset com intervalos menores do intervalo principal. De forma direta, para uma explicação mais clara, será utilizado a figura 16. Tomando a primeira classificação do primeiro *annotator* é visto o intervalo [1, 47323379] com classificação 3, ou seja, uma sinal de qualidade ruim. Esse intervalo está descrito em milissegundos, ou utilizando a nomenclatura do dataset, *ticks*. A cada 1000 ticks (1000 milissegundos) é passado 1 segundo, dessa forma pensou-se em uma oportunidade para separar um intervalo muito grande, em fragmentos menores. Como o intervalo maior tinha uma classificação associada a ele, então, por herança todos os seus fragmentos poderiam também ter essa mesma classificação. Assim, foram gerados novos arquivos .csv para cada intervalo e sua respectiva classificação.

Em um primeiro momento foi decidido dividir para cada *annotator*, intervalos de 10

mil, representando 10 segundos de duração do sinal, contudo essa abordagem percebeu-se um alto consumo de recursos computacionais. O mesmo ocorreu para intervalos de 5 mil ticks, em que a precisão melhorava um pouco, mas não tanto e ainda existia o problema e alto consumo nos recursos computacionais. Por último, utilizou-se uma abordagem em que o modelo seria treinado apenas com 1000 ticks. Isso reduziu de certa forma o gasto computacional mas ainda assim não o suficiente.

Decidiu-se, portanto, seguir apenas com 1000 ticks, e utilizar outras abordagens para o treinamento, pois ainda existiam problemas a serem resolvidos nos dados. Contudo, embora o problema de geração dos novos dados estivesse resolvido, o modelo de ML não conseguia aprender de maneira satisfatória.

Foi utilizado a biblioteca Scikit-Learn (PEDREGOSA *et al.*, 2011) para realizar o pré-processamento das informações a fim de normalizar os dados do sinal que possuíam valores não normalizados. Ao construir o novo dataset foi decidido que seria melhor aproveitado se os novos dados já fossem salvos de maneira normalizada.

A última etapa ocorreu enquanto construindo a nova base de dados. Nessa etapa foi percebido que os intervalos para cada *annotator* se sobrepunham, ou seja, alguns *annotators* diziam que um dado intervalo $[x,y]$ possuía classificação diferente para o mesmo intervalo. Para resolver esse problema, foi escolhido o *annotator* que continha os menores intervalos possíveis pois isso tem acarretado a uma melhor precisão na classificação. Utilizando a figura 16 como referência, tomando o primeiro e o segundo *annotator*, segunda linha, tem-se $[47323380, 47388207]$ com classificação 3 e $[47323872, 47329547]$ com classificação 2, se interceptam. Assim gerando uma inconsistência e repetição dos dados com classes diferentes, diminuindo a acurácia do modelo e mantendo o erro do modelo muito alto. Assim, foram escolhidos os *annotators* com menos interseção, no caso os *annotators* 3 e 4.

6.1 Treinamento e Validação

Para realizar o treinamento dos dados aplicou-se o carregamento dos dados de treinamento, validação e testes. A separação dos dados ocorreu a partir da construção do dataset gerado. Cada tarefa foi dividido em pastas, dessa forma nenhum dado conseguiria ter acesso a outro, portanto, evitando a mistura dos datasets.

O modelo foi treinado aplicando diversas combinações de hiper parâmetros utilizando a técnica de Random Search, que consiste em dado um conjunto de hiper parâmetros, escolher

diversas combinações por um número K de repetições.

De acordo com (ZABINSKY *et al.*, 2009) o algoritmo refere-se a um algoritmo que usa algum tipo de aleatoriedade ou probabilidade (normalmente na forma de um gerador de números pseudo-aleatórios) na definição do método.

O modelo adotado utiliza a técnica de *EarlyStopping* a fim de parar o treinamento do modelo quando não existir melhora do atributo "erro de validação", cada modelo treina sob 30 iterações utilizando a combinação de hiper parâmetros escolhidos pelo algoritmo de random search. A medida que o modelo treina, a validação é feita após o termino de cada iteração, assim, quanto menor o erro nos dados de validação, melhores são os resultados e é possível observar a existência de *Underfitting* ou *Overfitting*.

Cada modelo após o fim do treinamento é avaliado também com os dados de teste, assim, após a obtenção de todos esses dados são criadas diversas métricas a fim de demonstrar a qualidade do modelo descrito na próxima seção.

7 RESULTADOS

7.1 Análise de Precisão

Uma outra forma de validação é a verificação da precisão de um modelo. Nessa Seção, será analisado a quantidade de respostas corretas o modelo obteve baseado em cada combinação de hiperparâmetros. Por exemplo, é interessante saber qual melhor tamanho de batch para o modelo.

Dentre algumas métricas definidas, serão demonstradas as seguintes: precision, recall, f1_score, scores de teste e validação.

Para descrever os resultados do modelo, será utilizada principalmente os insights dados pela métrica F1_SCORE pois os dados são muito desbalanceados com relação às classes para classificação, por essa razão foi utilizado essa métrica a fim de melhor avaliar os modelos em questão.

Tabela 2 – Métrica F1_SCORE

	F1_SCORE
Melhor	0.9270
Média	0.9198
Pior	0.8927

Tabela 3 – Dados gerais de validação baseado no f1_score

	f1_score	val_accuracy
Máximo	0.9270	0.8766
Média	0.9198	0.8759
Mínima	0.8927	0.8439

Tabela 4 – Dados gerais de teste baseado no f1_score

	f1_score	test_accuracy
Máximo	0.927078	0.928165
Média	0.9198	0.9216
Mínima	0.892709	0.90137

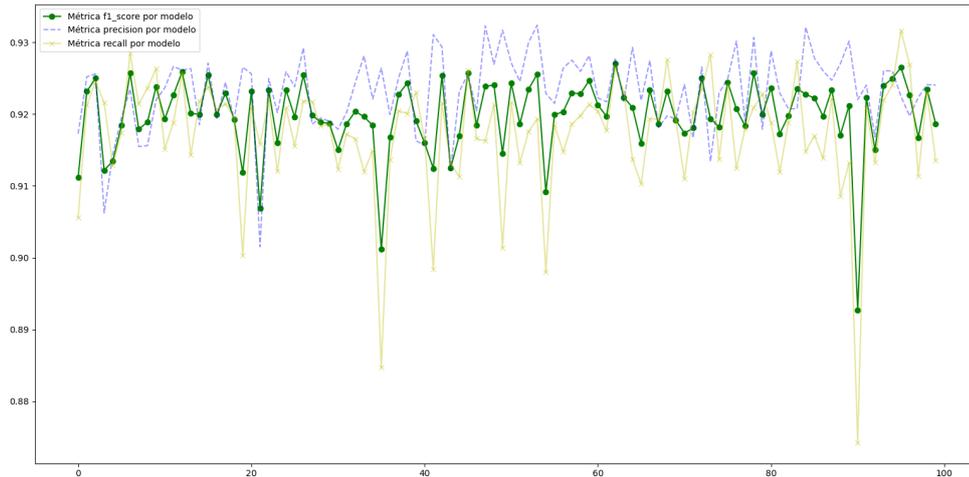
Tabela 5 – Dados gerais de precision e recall baseado no f1_score

	f1_score	precision	recall
Máxima	0.9270	0.9278	0.9264
Média	0.919	0.9231	0.9173
Mínima	0.8927	0.9220	0.8742

Tabela 6 – Dados gerais de precision e recall baseado no f1_score

	val_accuracy	test_accuracy	f1_score	precision	recall
Máxima	0.8766	0.9281	0.9270	0.9278	0.9264
Média	0.8759	0.9216	0.9198	0.9231	0.9173
Mínima	0.8439	0.9013	0.8927	0.9220	0.8742

Figura 17 – Comparação resultado de modelos F1_Score, Precision, Recall



Fonte: Autoria Própria

Como visto, o f1_score demonstra que o modelo é preciso e gera bons resultados com relação aos dados de validação e teste.

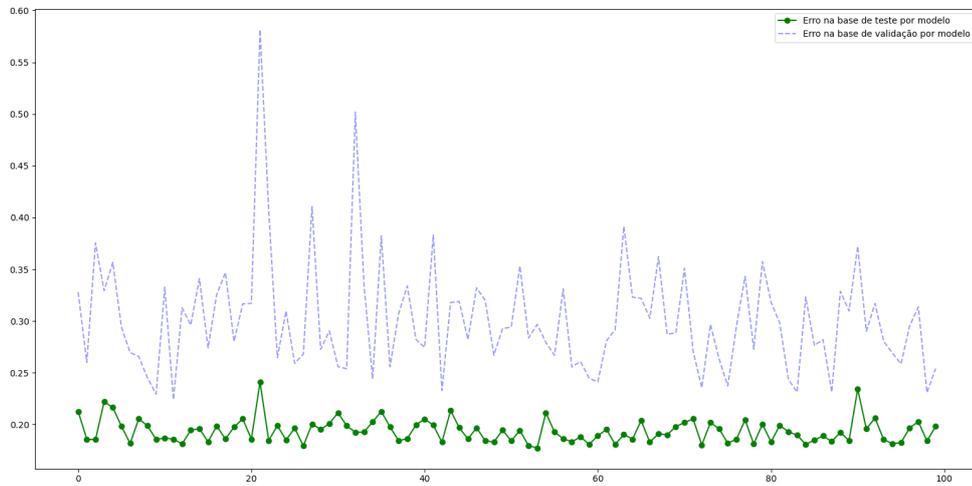
Para a construção deste trabalho, foi utilizado o para selecionar de forma aleatória diferentes combinações de hiperparâmetros, a seguir será descrito a faixa de valores usado para realizar as combinações entre os hiperparâmetros.

Tabela 7 – Faixa de valores para cada Hiperparâmetro

	batch_size	neurons	dropout	filter_size	kernel_size	stride
	50	500	0.2	32	4	4
	150	1250	0.4	64	6	6
	300	1750	-	128	8	-
	650	-	-	-	-	-
	-	-	-	-	-	-
	-	-	-	-	-	-

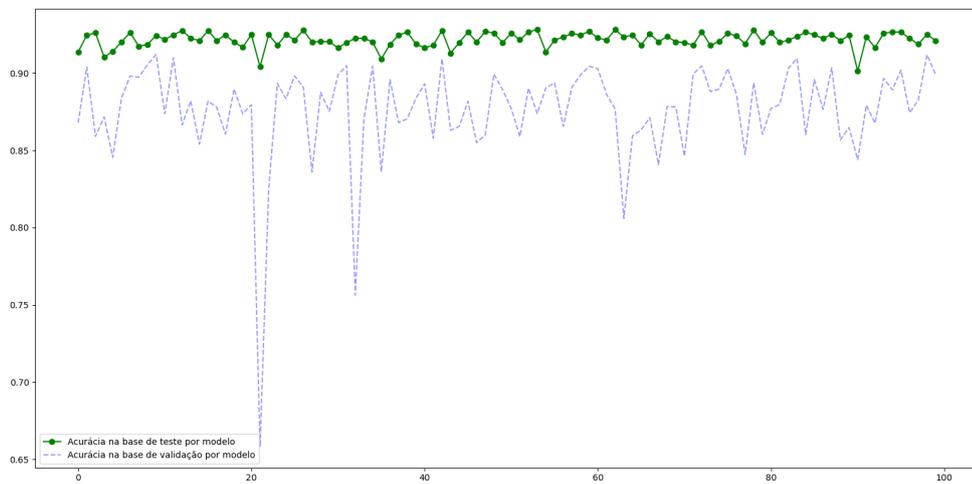
A seguir, na tabela 8 será demonstrado a melhor combinação de hiperparâmetros, ou seja, a combinação que maximiza o f1_score, bem como a pior combinação, ou em outras palavras, a combinação que obteve o pior resultado possível para o modelo.

Figura 18 – Comparação resultado de loss nos modelos na base de teste e validação



Fonte: Autoria Própria

Figura 19 – Comparação resultado de acurácia nos modelos na base de teste e validação



Fonte: Autoria Própria

Tabela 8 – Combinações de hiperparâmetros para o modelo

	batch_size	neurons	dropout	filter_size	kernel_size	stride
<i>Melhores Hiperparâmetros</i>	300	500	0.2	64	8	4
<i>Piores Hiperparâmetros</i>	650	250	0.4	128	4	4

8 CONCLUSÕES E TRABALHOS FUTUROS

Os resultados obtidos demonstram que a arquitetura e a forma como os dados foram construídos trazem resultados bons para classificação de sinal ECG. A arquitetura de uma RNC consegue extrair e é possível aplicá-la e determinar o desempenho para classificação da qualidade de sinais ECG. É possível observar também que a RNC é capaz de determinar o grau de aceitabilidade de um sinal de ECG em termos de sua qualidade. Contudo, é perceptível que ainda a muito a melhorar para obter uma classificação mais precisa. É possível que o acréscimo de novas features aumente a precisão pois isso significaria aumentar o tamanho do sinal observado, contudo isso causaria um aumento no uso dos recursos computacionais.

Observando os dados coletados e transformados, o modelo foi desenvolvido e verificado por meio de análises estatísticas que demonstraram resultados pertinentes à área. Com a análise, foi possível verificar que embora o dataset seja desbalanceado em sua quantidade de exemplos para algumas classes, o modelo consegue prever com uma precisão considerável um sinal de ECG.

Como trabalho futuro, observa-se a necessidade de refinamento do modelo, para que seja capaz de lidar com predições mais precisas das classes 2 e 3 que são as mais confusas para o modelo, dado a características parecidas. Outro possível trabalho de pesquisa pode se dar através da área de otimização da geração da nova base de dados, visto que muitos sinais se sobrepunham nos diferentes *Annotators* buscar uma forma otimizada de relacionar e separar os dados devidamente intersectando todos os *annotators*. Outro trabalho futuro se enquadra na classificação desses sinais mas em forma de imagens, ou seja, gerar imagens de X segundos, essa tem uma aplicação mais real, visto que a classificação de imagens pode ser tratada no mundo real de forma mais barata e acessível para esse problema. E por ultimo, como complemento do presente trabalho, estudar quais algoritmos para cálculo de similaridade melhor se adequa ao cenário atual, para então prever de forma completa a qualidade integral de um sinal de ECG.

REFERÊNCIAS

- ACADEMY, D. S. **O Perceptron**. 2021. Disponível em: <<https://www.deeplearningbook.com.br/o-perceptron-parte-1/>>.
- ACHARYA, U. R.; FUJITA, H.; OH, S. L.; HAGIWARA, Y.; TAN, J. H.; ADAM, M.; TAN, R. S. Deep convolutional neural network for the automated diagnosis of congestive heart failure using ecg signals. **Applied Intelligence**, Springer, v. 49, n. 1, p. 16–27, 2019.
- ASSOCIATION, A. H. **Electrocardiogram (ECG or EKG)**. 2015. Disponível em: <<https://www.heart.org/en/health-topics/heart-attack/diagnosing-a-heart-attack/electrocardiogram-ecg-or-ekg>>.
- BALOGLU, U. B.; TALO, M.; YILDIRIM, O.; TAN, R. S.; ACHARYA, U. R. Classification of myocardial infarction with multi-lead ecg signals and deep cnn. **Pattern Recognition Letters**, Elsevier, v. 122, p. 23–30, 2019.
- BROWNLEE, J. A **Gentle Introduction to Pooling Layers for Convolutional Neural Networks**. 2019. Disponível em: <<https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>>.
- CAI, W.; CHEN, Y.; GUO, J.; HAN, B.; SHI, Y.; JI, L.; WANG, J.; ZHANG, G.; LUO, J. Accurate detection of atrial fibrillation from 12-lead ecg using deep neural network. **Computers in biology and medicine**, Elsevier, v. 116, p. 103378, 2020.
- CLIFFORD, G.; BEHAR, J.; LI, Q.; REZEK, I. Signal quality indices and data fusion for determining clinical acceptability of electrocardiograms. **Physiological measurement**, v. 33, p. 1419–33, 08 2012.
- COMPUTERSCIENCEWIKI. **Max-pooling / Pooling**. 2018. Disponível em: <https://computersciencewiki.org/index.php/Max-pooling/_/_Pooling>.
- CS.STANFORD.EDU. **Biological Neurons**. Disponível em: <<https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Biology/index.html>>.
- CS.STANFORD.EDU. **Perceptron**. Disponível em: <<https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Neuron/index.html>>.
- DEEPU, C. J.; HENG, C.-H.; LIAN, Y. A hybrid data compression scheme for power reduction in wireless sensors for iot. **IEEE transactions on biomedical circuits and systems**, IEEE, v. 11, n. 2, p. 245–254, 2016.
- DESHPANDE, A. **A Beginner’s Guide To Understanding Convolutional Neural Networks**. 2016. Disponível em: <<https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>>.
- DSA, E. **Introduction to Different Activation Functions for Deep Learning**. Disponível em: <<http://deeplearningbook.com.br/cross-entropy-cost-function/>>.
- FOTIADOU, E.; KONOPCZYŃSKI, T.; HESSER, J.; VULLINGS, R. End-to-end trained encoder–decoder convolutional neural network for fetal electrocardiogram signal denoising. **Physiological Measurement**, IOP Publishing, v. 41, n. 1, p. 015005, feb 2020. Disponível em: <<https://doi.org/10.1088%2F1361-6579%2Fab69b9>>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.

HANNUN, A. Y.; RAJPURKAR, P.; HAGHPANAHI, M.; TISON, G. H.; BOURN, C.; TURAKHIA, M. P.; NG, A. Y. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. **Nature medicine**, Nature Publishing Group, v. 25, n. 1, p. 65, 2019.

HUANG, M.-L.; WU, Y.-S. Classification of atrial fibrillation and normal sinus rhythm based on convolutional neural network. **Biomedical Engineering Letters**, Springer, p. 1–11, 2020.

LUI, H. W.; CHOW, K. L. Multiclass classification of myocardial infarction with convolutional and recurrent neural networks for portable ecg devices. **Informatics in Medicine Unlocked**, Elsevier, v. 13, p. 26–33, 2018.

MAKNICKAS, V.; MAKNICKAS, A. Recognition of normal–abnormal phonocardiographic signals using deep convolutional neural networks and mel-frequency spectral coefficients. **Physiological measurement**, IOP Publishing, v. 38, n. 8, p. 1671, 2017.

MCCARTHY, P. J. **What is AI? / Basic Questions**. 2012. Disponível em: <<http://jmc.stanford.edu/artificial-intelligence/what-is-ai/index.html>>.

MISSINGLINK.AI. **Fully Connected Layers in Convolutional Neural Networks: The Complete Guide**. Disponível em: <<https://missinglink.ai/guides/convolutional-neural-networks/fully-connected-layers-convolutional-neural-networks-complete-guide/>>.

NEMCOVA, A.; VITEK, M.; NOVAKOVA, M. Complex study on compression of ecg signals using novel single-cycle fractal-based algorithm and spiht. **Scientific reports**, Nature Publishing Group, v. 10, n. 1, p. 1–15, 2020.

NEMCOVA RADOVAN SMISEK, K. O. M. V. L. S. L. M. A. **Brno University of Technology ECG Quality Database (BUT QDB)**. 2020. Disponível em: <<https://physionet.org/content/butqdb/1.0.0/>>.

OGNJANOVSKI, G. **Everything you need to know about Neural Networks and Backpropagation — Machine Learning Easy and Fun**. 2020. Disponível em: <<https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e>>.

ORGANIZATION, W. H. **Cardiovascular diseases (CVDs)**. 2017. Disponível em: <[https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))>.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PONTI, M.; COSTA, G. De Barros Paranhos da. **Como funciona o Deep Learning**. 2018.

RAMESH, A.; KAMBHAMPATI, C.; MONSON, J. R.; DREW, P. Artificial intelligence in medicine. **Annals of The Royal College of Surgeons of England**, Royal College of Surgeons of England, v. 86, n. 5, p. 334, 2004.

REGIS, C.; CALDEIRA, L.; GURJAO, E. Avaliação da amostragem compressiva em sinais de ecg e imagens digitais. **Revista Principia - Divulgação Científica e Tecnológica do IFPB**, v. 1, p. 95, 06 2016.

SATIJA, U.; RAMKUMAR, B.; MANIKANDAN, M. S. Real-time signal quality-aware ecg telemetry system for iot-based health care monitoring. **IEEE Internet of Things Journal**, IEEE, v. 4, n. 3, p. 815–823, 2017.

YAO, Q.; WANG, R.; FAN, X.; LIU, J.; LI, Y. Multi-class arrhythmia detection from 12-lead varied-length ecg using attention-based time-incremental convolutional neural network. **Information Fusion**, Elsevier, v. 53, p. 174–182, 2020.

ZABINSKY, Z. B. *et al.* Random search algorithms. **Department of Industrial and Systems Engineering, University of Washington, USA**, Citeseer, 2009.