

Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA  
Engenharia Eletrônica

**PROPOSTA DE FERRAMENTA DE  
MONTAGEM DE HIPNOGRAMAS  
UTILIZANDO DEEP LEARNING E  
ELETROENCEFALOGRAMA DE CANAL  
ÚNICO**

Autor: Thássio Gabriel Farias dos Santos  
Orientador: Prof. Dr. Gerardo Antonio Idrobo Pizo

Brasília, DF  
2019





Thássio Gabriel Farias dos Santos

**PROPOSTA DE FERRAMENTA DE MONTAGEM DE  
HIPNOGRAMAS UTILIZANDO DEEP LEARNING E  
ELETROENCEFALOGRAMA DE CANAL ÚNICO**

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Gerardo Antonio Idrobo Pizo

Brasília, DF

2019

---

Thássio Gabriel Farias dos Santos

PROPOSTA DE FERRAMENTA DE MONTAGEM DE HIPNOGRAMAS  
UTILIZANDO DEEP LEARNING E ELETROENCEFALOGRAMA DE CANAL  
ÚNICO/ Thássio Gabriel Farias dos Santos. – Brasília, DF, 2019-

77 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Gerardo Antonio Idrobo Pizo

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA , 2019.

1. sono. 2. redes neurais artificiais. 3. processamento de sinais. I. Prof. Dr.  
Gerardo Antonio Idrobo Pizo. II. Universidade de Brasília. III. Faculdade UnB  
Gama. IV. PROPOSTA DE FERRAMENTA DE MONTAGEM DE HIPNO-  
GRAMAS UTILIZANDO DEEP LEARNING E ELETROENCEFALOGRAMA  
DE CANAL ÚNICO

CDU 02:141:005.6

---

Thássio Gabriel Farias dos Santos

# **PROPOSTA DE FERRAMENTA DE MONTAGEM DE HIPNOGRAMAS UTILIZANDO DEEP LEARNING E ELETROENCEFALOGRAMA DE CANAL ÚNICO**

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Trabalho aprovado. Brasília, DF, 28 de novembro de 2019:

---

**Prof. Dr. Gerardo Antonio Idrobo  
Pizo**  
Orientador

---

**Prof. Dr. Marcelino Monteiro de  
Andrade**  
Convidado 1

---

**Prof. Dr. Daniel Mauricio Muñoz  
Arboleda**  
Convidado 2

Brasília, DF  
2019



# Resumo

A detecção de doenças que alteram a fisiologia do ser humano, principalmente as que alteram o sono, pode ser feita por meio do estudo do comportamento do sono de uma pessoa, que apresenta estágios que são categorizados como NREM1, NREM2, NREM3 e REM. A categorização destes estágios podem ser feitos por meio de polissonografias, exames que utilizam de vários sinais biomédicos como EEG, EMG e EOG. Como forma de reduzir o número de sinais biomédicos necessários para a determinação dos estágios do sono, este trabalho faz um estudo de como utilizar somente um canal de EEG para determinar os estágios do sono observando somente este sinal e o comportamento de seu espectro. A forma proposta é descarregar o conjunto de dados de EEG dos canais Pz-Oz e os dados de estágios do sono de exames de polissonografia advindos do banco de dados da PhysioNet, realizar com o Python(numpy) a análise da frequência destes sinais de EEG utilizando a Fast Fourier Transform e utilizar os resultados dos sinais processados como forma de parâmetro de entrada para a rede neural artificial, descrita em Python(Keras) rodando como backend a biblioteca TensorFlow, que receberá treinamento supervisionado. Após o treino e validação da rede neural, executar a montagem de hipnogramas, gráfico que mostra o progresso dos estágios do sono durante a noite. A montagem eficiente do hipnograma neste trabalho de forma generalizada, se torna limitada devido a insuficiência de amostras presentes no banco de dados. Neste trabalho há apenas a validação da possibilidade da criação da ferramenta devido aos resultados obtidos.

**Palavras-chaves:** sono, doença, redes neurais, sinais de EEG.





# Abstract

Detection of diseases that changes the human physiology, mainly the ones that affects sleep, can be done by studying the behavior of a person's sleep, which presents stages that are categorized as NREM1, NREM2, NREM3 and REM. The categorization of these stages can be done through of polysomnograms, exams that use of several biomedical signals like EEG, EMG and EOG. As a way to reduce the number of biomedical signals required for the determination of sleep stages, this work studies how to use only one EEG channel to determine the sleep stages by observing only this signal and the behavior of its spectrum. The proposed method is to download the EEG data of Pz-Oz channels and the sleep stages data from polysomnography exams from the PhysioNet's database, perform with Python(numpy) the frequency analysis of these EEG signals using Fast Fourier Transform and use the results of the processed signals as an input parameter for the artificial neural network described in Python(Keras) running TensorFlow as backend, which will receive supervised training. After the training and validation of the neural network, perform the assembly of the hypnograms, a graph that shows the progress of sleep stages at night. The efficiency of assembly the hypnogram in this work in a general way becomes limited due to insufficient samples present in the database. In this work there is only the validation of the possibility of creating the tool due to the results obtained.

**Key-words:** sleep, disease, artificial neural network, EEG signals.



# Lista de ilustrações

Figura 1 – Hipnograma no modelo de Dement e Kleitman, modificada de (FERNÁNDEZ-LEAL et al., 2017) . . . . .	22
Figura 2 – Hipnograma no modelo da Academia Americana de Medicina do Sono, modificada de (FERNÁNDEZ-LEAL et al., 2017) . . . . .	25
Figura 3 – Hipnograma de dois pacientes, o de cima normal e o de baixo com depressão modificada de (NUTT; WILSON; PATERSON, 2008) . . . . .	27
Figura 4 – Atividade de ondas lentas durante uma noite de sono de dois pacientes, o de cima normal e o de baixo com depressão modificada de (NUTT; WILSON; PATERSON, 2008) . . . . .	27
Figura 5 – Diagrama de blocos resumido do funcionamento de um EEG . . . . .	28
Figura 6 – Forma de montagem dos eletrodos para uma EEG (DUFFY et al., 1990) . . . . .	29
Figura 7 – EEG de canal único Pz-Oz modificado de (DUFFY et al., 1990) . . . . .	30
Figura 8 – Representação gráfica de (a) sinais de tempo contínuo e (b) Sinal de tempo discreto gerado por amostragem do sinal (a) . . . . .	31
Figura 9 – Sinal contínuo discretizado com uma taxa de 10 amostragens por segundo . . . . .	32
Figura 10 – Sinal discretizado com a presença de sinais com 2, 1 e 0.5 Hz . . . . .	33
Figura 11 – Magnitude do sinal da Fig. 10 no domínio da frequência . . . . .	34
Figura 12 – Funcionamento do sistema nervoso . . . . .	36
Figura 13 – Composição de um neurônio modificado de (HAYKIN et al., 2009) . . . . .	36
Figura 14 – Camadas de um perceptron modificado de (HAYKIN et al., 2009) . . . . .	37
Figura 15 – Comportamento da função tanh . . . . .	38
Figura 16 – Comportamento da função ELU. A curva em azul representa o comportamento não linear da função para números negativos, possuindo sua saturação em -1. A linha em vermelho representa o comportamento linear da função . . . . .	39
Figura 17 – Camadas de uma rede MLP . . . . .	40
Figura 18 – Diagrama do funcionamento do Aprendizado Supervisionado . . . . .	41
Figura 19 – Sinal de EEG no domínio do tempo (a) e frequência(b) em um período de 30 segundos . . . . .	46
Figura 20 – Fluxo de funcionamento do código do cálculo da FFT e potência média do sinal . . . . .	48
Figura 21 – Grafo da rede neural . . . . .	50
Figura 22 – Fluxograma do funcionamento da ferramenta . . . . .	51
Figura 23 – Acurácia dos modelos treinados . . . . .	53
Figura 24 – Perda dos modelos treinados . . . . .	53
Figura 25 – Hipnogramas do 1º modelo . . . . .	54

Figura 26 – Matriz Confusão do 1º modelo . . . . .	55
Figura 27 – Hipnogramas do 2º modelo . . . . .	55
Figura 28 – Matriz Confusão do 2º modelo . . . . .	56
Figura 29 – Hipnogramas do 3º modelo . . . . .	56
Figura 30 – Matriz Confusão do 3º modelo . . . . .	57
Figura 31 – Hipnogramas do 4º modelo . . . . .	57
Figura 32 – Matriz Confusão do 4º modelo . . . . .	58
Figura 33 – QRcode Github . . . . .	66

# Lista de tabelas

Tabela 1 – Diferenças entre os estágios do sono NREM (LEE-CHIONG, 2008) . . .	23
Tabela 2 – Atividades durante o sono REM (LEE-CHIONG, 2008) . . . . .	24
Tabela 3 – Diferenças entre os modelos . . . . .	25
Tabela 4 – Quantidade de amostras por estágio do sono . . . . .	44
Tabela 5 – Separação do dataset . . . . .	45
Tabela 6 – Separação do dataset . . . . .	47
Tabela 7 – Acurácia e perda para cada modelo durante a validação . . . . .	52
Tabela 8 – Acurácia para cada modelo durante os testes . . . . .	54



# Lista de abreviaturas e siglas

EEG	Eletroencefalografia
EMG	Eletromiografia
ECG	Eletrocardiografia
REM	Rapid Eye Movement
NREM	Non-Rapid Eye Movement
FFT	Fast Fourier Transform





# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>1.1</b>	<b>Definição do Problema</b>	<b>18</b>
<b>1.2</b>	<b>Objetivos</b>	<b>18</b>
1.2.1	Objetivo Geral	18
1.2.2	Objetivos Específicos	18
<b>1.3</b>	<b>Organização do trabalho</b>	<b>19</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>20</b>
<b>2.1</b>	<b>Sono</b>	<b>20</b>
2.1.1	Ondas cerebrais	21
2.1.2	Estágios do sono	21
2.1.2.1	Sono NREM	23
2.1.2.2	Sono REM	23
2.1.3	Ajuste nos Estados do Sono	24
2.1.3.1	Aplicação do Estudo do Comportamento do Sono	25
<b>2.2</b>	<b>Eletroencefalografia</b>	<b>26</b>
2.2.1	Fixação dos eletrodos	28
2.2.2	Uso do EEG durante o sono	28
2.2.2.1	Utilização de somente um canal	28
<b>2.3</b>	<b>Análise dos sinais</b>	<b>29</b>
2.3.1	Análise dos sinais no domínio do tempo	29
2.3.2	Discretização de um sinal contínuo no tempo	31
2.3.3	Análise de sinais discretos no domínio da frequência	32
2.3.3.1	Transformada Rápida de Fourier	32
<b>2.4</b>	<b>Redes Neurais Artificiais</b>	<b>35</b>
2.4.1	O Neurônio	36
2.4.1.1	Função de ativação do perceptron	37
2.4.2	Rede Neural Multilayer Perceptron(MLP)	40
2.4.3	Aprendizagem da rede neural	41
<b>3</b>	<b>METODOLOGIA</b>	<b>43</b>
<b>3.1</b>	<b>Aquisição dos Dados</b>	<b>43</b>
<b>3.2</b>	<b>Processamento de Dados</b>	<b>44</b>
3.2.1	Organização e formatação dos dados	44
3.2.2	Extração de Informação do Sinais de EEG	45
3.2.3	Determinação do Estágio do Sono	47

3.2.4	Treinamento e Validação da Rede Neural . . . . .	49
3.3	Montagem do Hipnograma . . . . .	50
4	<b>RESULTADOS</b> . . . . .	<b>52</b>
4.1	Resultados de Treino e Validação . . . . .	52
4.2	Resultados de Teste . . . . .	54
5	<b>CONCLUSÃO</b> . . . . .	<b>60</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>62</b>
	<b>APÊNDICES</b> . . . . .	<b>65</b>
	<b>APÊNDICE A – LINK GITHUB</b> . . . . .	<b>66</b>
	<b>APÊNDICE B – CÓDIGO REDE NEURAL</b> . . . . .	<b>67</b>
	<b>APÊNDICE C – CÓDIGO MONTAGEM DATASET</b> . . . . .	<b>71</b>
	<b>APÊNDICE D – CÓDIGO MONTAGEM HIPNOGRAMA E MA- TRIZ CONFUSÃO</b> . . . . .	<b>74</b>

# 1 Introdução

Os avanços na área da ciência e da computação são feitos que afetam as mais diversas áreas do conhecimento, que utilizam destes avanços como ferramentas para alcançar seus objetivos. Uma área que se beneficia bastante destes avanços é a área biomédica, que tenta sempre descobrir ou aprimorar seus métodos de detecção de doenças.

Das doenças que existem, algumas delas alteram a fisiologia de um ser. Esta alteração acaba se tornando em sinal biológico dado pelo ser humano como um indicativo para a presença de doenças. Um dos aspectos fisiológicos de um ser humano que pode se alterar é o seu sono, sendo perceptível então notar algumas diferenças no comportamento do sono entre a pessoa saudável e a pessoa doente.

O auxílio na detecção destas alterações do comportamento do sono pode ser feito por meio da tecnologia ao se utilizar dados de exames de eletroencefalografia, estudo do sono, processamento de sinais e rede neurais artificiais.

A eletroencefalografia é um exame que visa monitorar a atividade elétrica do cérebro. Este exame era utilizado inicialmente para detecção de patologias e estudar o comportamento das atividades elétricas cerebrais. Ao decorrer dos anos estudos foram feitos para observar o comportamento do sono, o que se estendeu também em tentativas de identificação e diagnóstico de patologias que alteram o comportamento do sono.

No entanto, fazer a identificação do comportamento do sono requer treinamento e conhecimento técnico. Uma forma de automatizar esta identificação é fazer o uso da Transformada Rápida de Fourier, do inglês *Fast Fourier Transform* (FFT), e o uso de Redes Neurais Artificiais sobre os sinais de eletroencefalografia.

A FFT é uma ferramenta poderosa no processamento de sinais, pois, o espectro de um determinado sinal pode ser analisado rapidamente por meio da aplicação deste algoritmo. Assim, o espectro do sinal da eletroencefalografia, que é um dos aspectos mais importante na determinação do comportamento do sono, pode ser finalmente extraído e utilizado como informação para a rede neural.

A rede neural por sua vez, por meio de um aprendizado supervisionado, conseguiria aprender como o espectro dos sinais de eletroencefalografia são relacionados com cada estágio do sono. Assim, é possível ser montado com poucos recursos um gráfico que mostra o comportamento do sono de um indivíduo, o que pode ser utilizado como auxílio para especialistas no assunto diagnosticarem doenças.

Para este aprendizado se faz necessário o uso de um grande *dataset* para que a rede consiga aprender da forma mais generalizada possível, conseguindo assim aplicar seu

modelo para qualquer pessoa que possua dados de EEG coletados. Esta conclusão se dá ao fato deste trabalho não conseguir apresentar o melhor modelo generalizado para todas as redes devido a limitação de banco de dados, porém, consegue validar a possibilidade de utilizar a mesma abordagem utilizada aqui para realizar a categorização dos estágios do sono.

## 1.1 Definição do Problema

A identificação de doenças que alteram a fisiologia humana, principalmente as que alteram de forma significativa o comportamento do sono, podem ser identificadas se observamos este comportamento, como por exemplo, a depressão (STEIGER; KIMURA, 2010)(NUTT; WILSON; PATERSON, 2008). Porém, a identificação do comportamento do sono é feita por meio de polissonografia, um exame que além de conter a eletroencefalografia, exige também o uso de sensores para medição de atividade muscular, eletromiografia e a medição da atividade dos olhos, eletro-oculografia. Como forma de reduzir o número de recursos necessários, este trabalho propõe a criação de uma ferramenta que consiga classificar os estágios do sono com base somente em dados de um canal de eletroencefalografia, pois, dependendo da região da cabeça na qual os eletrodos são posicionados, os dados de canal único de eletroencefalografia possuem a presença de todas as frequências de ondas cerebrais, mesmo que em menor intensidade. Assim, após processar este sinal e extrair as informações de interesse presentes nele, estes dados podem ser utilizados como parâmetros de entrada para uma rede neural que classificará o estágios do sono para posteriormente ser realizada a montagem hipnogramas.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Propor a possibilidade de desenvolvimento de ferramenta que utiliza redes neurais artificiais para determinar os estágios do sono somente com dados de canal único de eletroencefalografias para realizar a montagem de hipnogramas.

### 1.2.2 Objetivos Específicos

Para atingir o objetivo geral, foram definidos os seguintes objetivos específicos:

- Processamento dos sinais de EEG;
- Montagem de *dataset* com *features* e labels para rede neural;
- Estruturação de rede neural de classificação utilizando aprendizado supervisionado;

- Montagem de hipnograma.

## 1.3 Organização do trabalho

O trabalho está dividido em 4 capítulos. O capítulo 2 é sobre o referencial teórico coletado, ele está dividido em 5 subseções que discorrem sobre o sono, depressão, eletroencefalografia, análise de sinais e redes neurais artificiais. o capítulo 3 define a metodologia que será utilizada para a integração das informações coletadas. O capítulo 4 é sobre o resultado das pesquisas e testes feitos ao longo do trabalho. Por fim, o capítulo 5 apresenta a conclusão do trabalho.

## 2 Referencial Teórico

### 2.1 Sono

Uma característica que é intrínseca de todos os seres humanos é a necessidade fisiológica que se tem de dormir, definido como o ato de entrar em estado de sono. Os seres humanos possuem três estados comportamentais e fisiológicos, sendo eles, o estado de vigília, quando se está totalmente desperto, o estado de sono no qual os olhos não se mexem rapidamente, *non-rapid eye movement* (NREM) sleep, e o estado de sono no qual os olhos se mexem rapidamente, *rapid eye movement* (REM) (LEE-CHIONG, 2008).

Entrar em estado de sono é uma característica fisiológica dos seres humanos no qual se nota características como:

- Olhos fechados;
- Corpo deitado;
- Respiração com ritmo regular e lento;
- Diminuição dos movimentos;
- Pouca ou nenhuma resposta a estímulos exteriores.

Estas características comuns implicam que um sono regular deve possuí-las para que a noite de sono seja no mínimo satisfatória e que conseqüentemente uma pessoa consiga obter qualidade em seu sono. A qualidade do sono é um fator que implica em fatores que conseguem alterar características no comportamento de uma pessoa ao longo do dia. Isso porque o sono possui as determinadas funções que são obrigatórias para o correto funcionamento do corpo humano (LEE-CHIONG, 2008).

Dentre as funções do sono, a que mais é perceptível pelos seres humanos é a de restauração da energia após o período de vigília. Porém, o sono tem mais funções que foram constatadas ao longo do tempo por pesquisadores. Algumas funções do sono são:

- Liberação do hormônio do crescimento;
- Crescimento e processamento neural;
- Consolidação de memória;
- Termorregulação;
- Conservação de energia.

### 2.1.1 Ondas cerebrais

Antes de continuar a tratar sobre o sono, é importante dar uma breve explicação sobre as ondas cerebrais que são produzidas pelas oscilações e conexões neurais (DOBORJEH et al., 2019). Estas ondas cerebrais são detectadas por meio de eletroencefalografia, método utilizado para fazer as detecções destas ondas. As ondas já conhecidas e importantes na caracterização do comportamento do sono são categorizadas de acordo com sua frequência, são elas:

- delta (0.4–4 Hz);
- teta (4–8 Hz);
- alfa (8–12 Hz);
- beta (12–28 Hz).

A predominância destas ondas se alteram de acordo com estágio de sono no qual uma pessoa se encontra, devido a isso, é de extrema importância que este tópico seja introduzido brevemente e melhor explicado posteriormente.

### 2.1.2 Estágios do sono

Como já citado anteriormente, dos três estados comportamentais e fisiológicos, o sono cobre dois deles. Estes estados chamados de sono REM e sono NREM recebem estes nomes devido a movimentação do olho humano durante estes estados. O sono NREM que possui 4 estágios nas definições de Dement e Kleitman, os olhos não se movimentam ou se movimentam devagar, justificando o nome *Non-rapid Eye movement(NREM)*, que em português pode ser traduzido como movimento não rápido do olhos. O outro estado é o sono *Rapid Eye Movement(REM)* no qual pode ser traduzido para o português como movimento rápido do olho, como o próprio nome sugere, o olho humano faz movimentos rapidamente durante este estado (LEE-CHIONG, 2008).

A passagem pelos 4 ciclos do sono NREM até a finalização do sono REM é definida como um ciclo de sono. O primeiro ciclo de sono de uma pessoa adulta tem uma duração que vai de 70 a 100 minutos, do segundo ciclo em diante, as durações ficam entre 90 e 120 minutos, a divisão de tempo para cada ciclo tem o seguinte aspecto(LEE-CHIONG, 2008) (HARVARD, 2007):

- estágio 1 NREM entre 2–5% do ciclo;
- estágio 2 NREM entre 45–55% do ciclo;
- estágio 3 NREM entre 5–20% do ciclo;

- estágio 4 NREM entre 5–20% do ciclo;
- estágio REM entre 20–25% do ciclo.

O número de ciclos que uma pessoa adulta necessita idealmente é de 4 a 5 ciclos, devido a essa necessidade, a recomendação de horas na cama é de no mínimo de 6 horas por dia, um tempo de menor que esse já pode ser considerado com insuficiência de sono. Estes padrões citados nem sempre são seguidos exatamente dessa forma para todos os humanos, pois além de cada ser humano ser único, e conseqüentemente apresentar comportamentos um pouco diferentes, aspectos como, idade, quantidade de tempo de sono que se teve na noite anterior, relógio biológico, praticar exercícios, estresse, condições ambientais, ingestão de remédios e doenças (LEE-CHIONG, 2008) (HARVARD, 2007).

A disposição dos estados do sono nos modelos de Dement and Kleitman, descobridores dos padrões dos sonos NREM e REM, está de acordo com a Fig. 1 na qual é apresentado um hipnograma, uma representação gráfica dos estágios do sono. Nesta representação, os estágios do sono são apresentados verticalmente e o tempo em horas apresentado horizontalmente. Os retângulos pretos demarcam os períodos nos quais foi observado movimento rápido dos olhos, ou seja, períodos nos quais o paciente estava no estado de sono REM. O tamanho das linhas definem a quantidade de tempo que se permaneceu naquele estágio. A setas que aparecem logo após os retângulos são para indicar o começo de um novo ciclo.

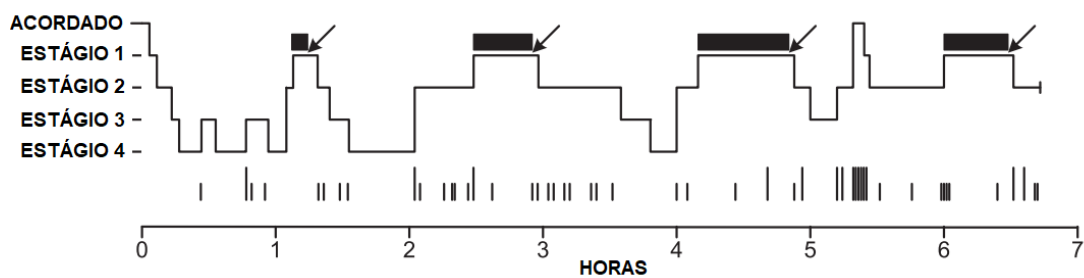


Figura 1 – Hipnograma no modelo de Dement e Kleitman, modificada de (FERNÁNDEZ-LEAL et al., 2017)

Este modelo foi utilizado fortemente após a Federação Internacional de Eletroencefalografia e Comitê de Neurofisiologia Clínica padronizar as terminologias utilizadas com o intuito de facilitar a classificação e identificação dos estágios dos sonos em diferentes estudos. Esta padronização incluía parâmetros, técnicas e padrões de ondas obtidas por Eletroencefalografia, eletro-oculografia e eletromiografia (FERNÁNDEZ-LEAL et al., 2017).



### 2.1.2.1 Sono NREM

Ao se deitar na cama para dormir, uma pessoa atinge o estado de sono geralmente pelo sono NREM, que durante a primeira parte da noite tem muita predominância, em termos do tempo total do ciclo, sobre o sono REM (LEE-CHIONG, 2008).

Durante um ciclo de sono, não necessariamente ocorrem todos os 4 estágios do sono NREM. Os estados possuem características e tempos de duração diferentes, para a determinação correta da divisão desses estágios que ocorrem durante o sono NREM, são utilizados critérios que possuem variáveis provindas de análises de sinais biomédicos durante o sono. Os sinais biomédicos avaliados são (LEE-CHIONG, 2008):

- Eletroencefalografia;
- Eletro-oculografia;
- Eletromiografia do queixo;
- Comparação de um estágio com a outro.

A diferença entre os estágios utilizando esses critérios pode ser visto na Tab. 1.

Estágio do Sono NREM	Eletroencefalografia	Eletro-oculografia	Eletromiografia
1	Baixa amplitude, frequências variadas e predominância de ondas teta	Rolagem lenta dos olhos	Alta atividade muscular no queixo
2	Baixa amplitude, frequências variadas e ondas Delta ocupam menos de 20 % do período desta fase	Sem movimento dos olhos	Pouca atividade muscular no queixo
3	Ondas Delta ocupam entre 20 e 50 % do período desta fase	Sem movimento dos olhos	Menor atividade muscular no queixo comparado as fases 1 e 2
4	Ondas Delta ocupam mais da metade do período desta fase	Sem movimento dos olhos	Menor atividade muscular no queixo comparado as fases 1 e 2

Tabela 1 – Diferenças entre os estágios do sono NREM (LEE-CHIONG, 2008)

### 2.1.2.2 Sono REM

Após ocorrer a passagem das fases do sono NREM, entra-se no estado de sono REM que durante a segunda parte da noite está mais presente no tempo total de um ciclo do que o sono NREM. O sono Rem ocorre de 3 a 5 vezes durante um sono que vai 6 a 8 horas por dia, compondo em média cerca de 20 a 25 % do sono total de um humano adulto (LEE-CHIONG, 2008).

Durante o período do sono REM, os sonhos ficam mais complexos se comparadas com o que ocorre durante o sono NREM, para prevenir que durante os sonhos uma pessoa se mexa e se machuque, a atividade muscular dos braços e pernas no geral se cessam. A percepção de estímulos externos diminui bastante, o que torna a ação de acordar alguém mais difícil, sendo necessário sons ou movimentação o corpo da pessoa com uma intensidade maior (LEE-CHIONG, 2008).

Como apresentado anteriormente, o sono NREM possui características distintas nos sinais de eletroencefalografia, eletro-oculografia e eletromiografia, a análise destes sinais biomédicos durante o sono REM apresentam diferenças quando comparados com os sinais do estágio de NREM, as características estão citadas na Tab. 2.

<b>Eletroencefalografia</b>	<b>Eletro-oculografia</b>	<b>Eletromiografia</b>
Baixa amplitude, frequências variadas, frequência das ondas alfas menores que as que ocorrem durante estado de vigília e estágio 1 do sono NREM	Alternância entre a não rolagem e a rolagem rápida dos olhos	O queixo apresenta pouca ou nenhuma atividade que pode ser seguida de uma intensa e rápida atividade muscular

Tabela 2 – Atividades durante o sono REM (LEE-CHIONG, 2008)

### 2.1.3 Ajuste nos Estados do Sono

Por mais que o modelo mostrado anteriormente já estivesse padronizado, a forma na qual ele era proposto, apresentava dificuldades para se fazer classificações corretas. Assim, estudiosos do sono em diferentes partes do mundo aprimoraram os métodos utilizados fazendo uma melhoria no refinamento da identificação dos padrões dos sinais cerebrais obtidos por meio de eletroencefalografia. Com estas melhorias, foi possível fazer a adição de mais critérios para a classificação dos estágios do sono.

Estas classificações são feitas ao se fazer uma amostragem de intervalo de 30 segundos, dependendo do padrão observado, é possível determinar em qual estágio de sono uma pessoa se encontra, se durante este intervalo for detectado dois comportamentos distintos, ou seja, a amostragem foi feita em um período de transição, determina-se que o estado no qual a pessoa está naquele momento é o que durou mais durante o intervalo de 30 segundos (DOBORJEH et al., 2019).

A classificação dos estágios do sono foram aprimorados pela Academia Americana de Medicina do Sono e agora são referidos como:

- Acordado
- Estágio N1 (NREM 1)

- Estágio N2 (NREM 2)
- Estágio N3 (NREM 3)
- Estágio R (REM)

O estágio N3 é o momento de sono profundo que se tinha antes categorizado como estágio 3 e 4 pelo modelo de Dement e Kleitman. Uma comparação entre os modelos pode ser visto na Tab. 3.

Dement e Kleitman	Academia Americana de Medicina do Sono	Atividade Cerebral
Estágio 1	Estágio N1	Ondas Alfa, Teta e com vale e crista pontiagudos
Estágio 2	Estágio N2	Ondas Teta, ondas com alta frequência e baixo período e presença de ondas
Estágio 3	Estágio N3	Ondas Teta e Delta
Estágio 4	-	-
Estágio REM	Estágio R	Ondas Teta e Alfa

Tabela 3 – Diferenças entre os modelos

O hipnograma da Fig. 1 conseqüentemente sofre algumas alterações devido a reestruturação dos estágios do sono. Um hipnograma nos moldes da Academia Americana de Medicina do Sono pode ser visto na Fig. 2.

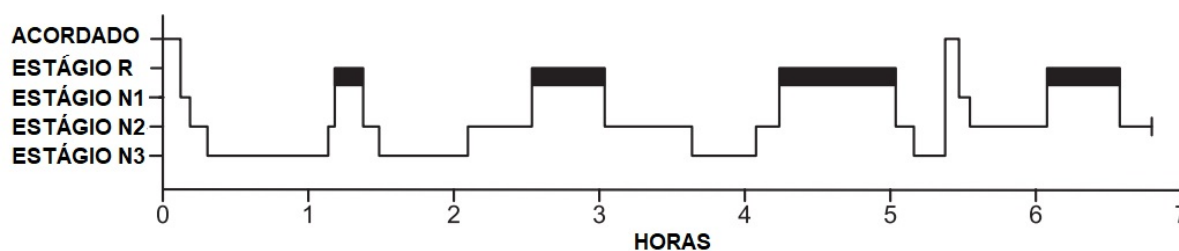


Figura 2 – Hipnograma no modelo da Academia Americana de Medicina do Sono, modificada de (FERNÁNDEZ-LEAL et al., 2017)

### 2.1.3.1 Aplicação do Estudo do Comportamento do Sono

O comportamento do sono visto nestes hipnogramas ocorre de forma diferente em pessoas doentes, como por exemplo nas pessoas que sofrem com transtornos de humor. Usando a depressão como exemplo, é visto que os ciclos do sono em pessoas depressivas e pessoas não depressivas possuem comportamentos diferentes (STEIGER; KIMURA, 2010) (NUTT; WILSON; PATERSON, 2008). Desta forma, o comportamento do sono poderia

ser um possível biomarcador, sinal médico que tem como objetivo indicar um quadro clínico, na identificação da presença de depressão em um indivíduo.

Estudos mostram que durante uma noite normal de sono, uma pessoa com depressão apresenta estados de sono com períodos diferentes de uma pessoa normal. Os sintomas a serem observados são (STEIGER; KIMURA, 2010) (NUTT; WILSON; PATERSON, 2008):

- Continuidade do sono prejudicada (aumento no número de vezes que se desperta durante a noite e períodos maiores estando acordado);
- Latência entre o estado REM encurtada;
- Presença de pouco sono profundo (Estado NREM3);

É possível observar nos hipnogramas da Fig. 3, o comportamento do sono em pessoas depressivas. Estes hipnogramas estão montados fora do padrão da Academia Americana de Medicina do Sono por ser de um estudo realizado antes do padrão ser adotado, ou seja, em 2015. Vale ressaltar que a diminuição da aparição do estado de sono profundo em pessoas com depressão pode estar relacionado com o fluxo de sangue que passa por áreas do cérebro que são afetadas pela depressão (STEIGER; KIMURA, 2010) (NUTT; WILSON; PATERSON, 2008). Para uma melhor compreensão de como são captados estes dados, o entendimento do funcionamento do EEG se faz necessário.

## 2.2 Eletroencefalografia

O comportamento da atividade elétrica do cérebro pode ser observado utilizando dois parâmetros, a amplitude das ondas geradas em microvolts e a frequência delas em Hertz. As frequências sinusoidais observadas durante um EEG tem origem na superfície do córtex cerebral e podem ser medidas por meio do couro cabeludo de uma pessoa (DUFFY et al., 1990).

Durante uma eletroencefalografia, é medida a atividade elétrica do cérebro por aparelhos que possuem de 8 a 24 canais, sendo esta medição feita simultaneamente por todos os canais. Os componentes de um circuito de EEG são basicamente eletrodos, amplificadores e filtros com as respectivas funções de captação da diferença de potencial elétrico, amplificação do sinal elétrico de forma que o sinal não seja distorcido e eliminação de espectros de frequência indesejáveis. O funcionamento de um aparelho de EEG pode ser descrito no diagrama de blocos da Fig. 5 (DUFFY et al., 1990).

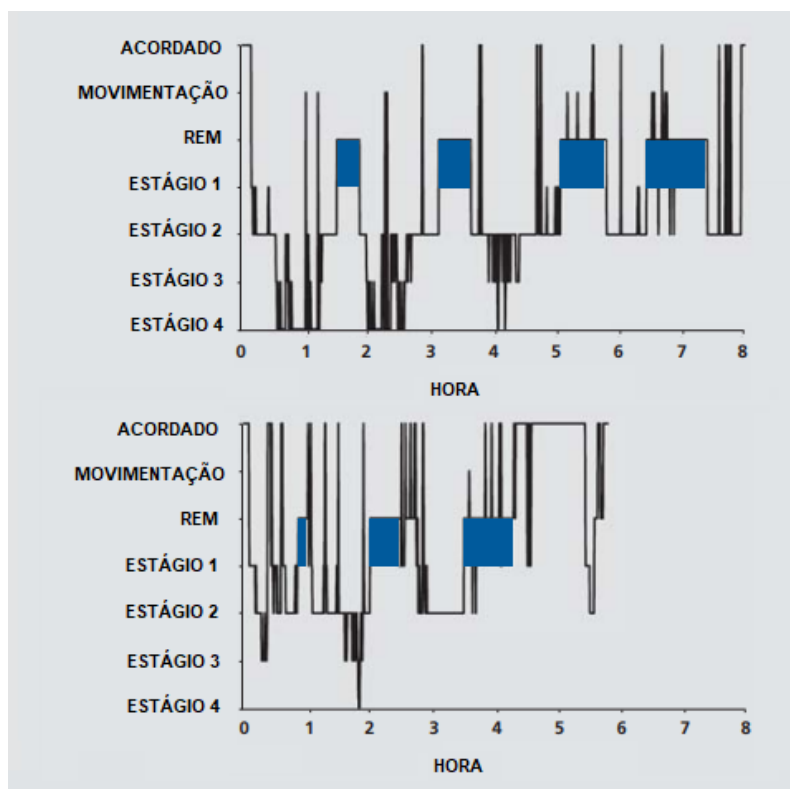


Figura 3 – Hipnograma de dois pacientes, o de cima normal e o de baixo com depressão modificada de (NUTT; WILSON; PATERSON, 2008)

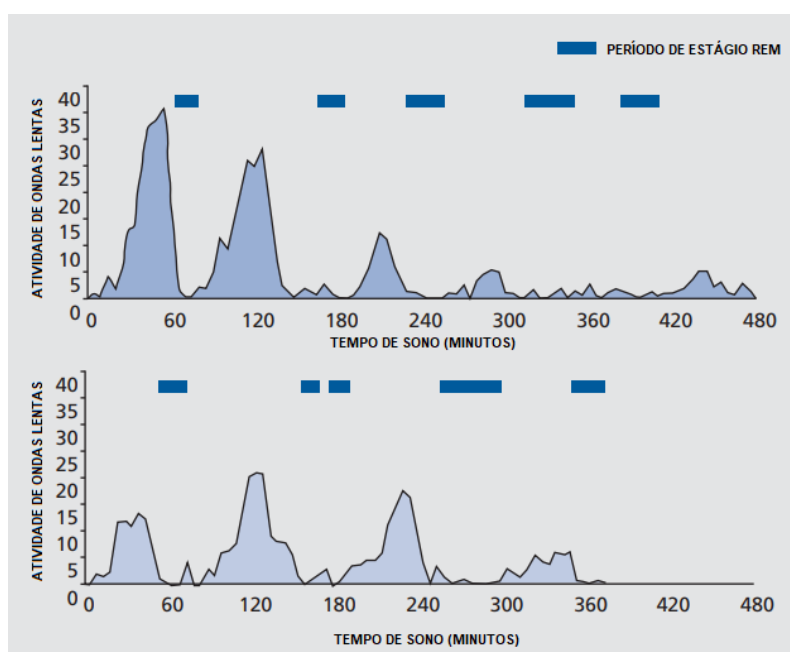


Figura 4 – Atividade de ondas lentas durante uma noite de sono de dois pacientes, o de cima normal e o de baixo com depressão modificada de (NUTT; WILSON; PATERSON, 2008)

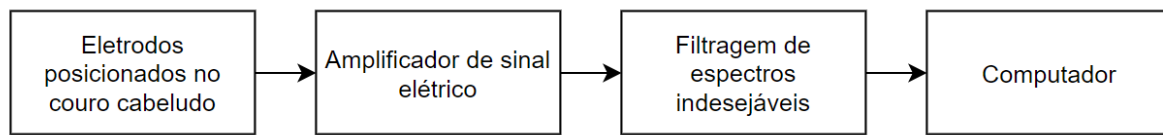


Figura 5 – Diagrama de blocos resumido do funcionamento de um EEG

### 2.2.1 Fixação dos eletrodos

A disposição dos eletrodos depende da medida que será feita. Um bom método de dispor os eletrodos pela cabeça é o sistema internacional 10-20 que recebe este nome em função das distâncias entre os eletrodos que estão posicionados com uma distância de 10 ou 20% de um até o outro. Este método consegue ser bastante eficiente, pois consegue cobrir todas as áreas do cérebro de forma apropriada em crianças e adultos com a cabeça grande (DUFFY et al., 1990). A Fig. 6 mostra a disposição destes eletrodos.

A nomenclatura de cada eletrodo é dada por letra e número. As letras representam as iniciais de quatro partes do crânio, sendo elas a frontal (F), central (C), parietal (P) e occipital (O). Os números são utilizados para representar a posição de cada eletrodo, sendo os números pares os eletrodos da direita e os números ímpares os eletrodos da esquerda. Existem ainda três eletrodos que possuem nomenclaturas um pouco diferentes, são eles: A1, A2, Fp1, Fp2, Fz, Cz e Oz. Os eletrodos Fp estão localizados na testa. Os eletrodos que possuem terminação em z por estarem localizados exatamente entre os hemisférios do crânio, não recebem números. Os eletrodos com a letra 'A' estão localizados nos lóbulos das orelhas (DUFFY et al., 1990).

### 2.2.2 Uso do EEG durante o sono

O uso do EEG durante o sono é utilizado em exames de polissonografia, exame no qual são adquiridos e analisados vários sinais biomédicos como EEG, EMG e ECG.

A polissonografia é considerado o padrão de ouro quando o assunto é monitoramento do sono, sendo possível definir de forma detalhada cada estágio do sono e dar o diagnóstico de distúrbios do sono como por exemplo a apneia. Porém, a utilização da polissonografia requer o uso de muitos sensores, consequentemente aumentando o custo de fabricação do produto o que aumenta também o custo de se realizar um exame deste.

#### 2.2.2.1 Utilização de somente um canal

Como solução de barateamento e menor uso computacional, existe a proposta já estudada de utilizar somente um canal no exame de EEG para definir o estado do sono no qual o paciente se encontra. Estudos feitos já constataram que é possível obter bons

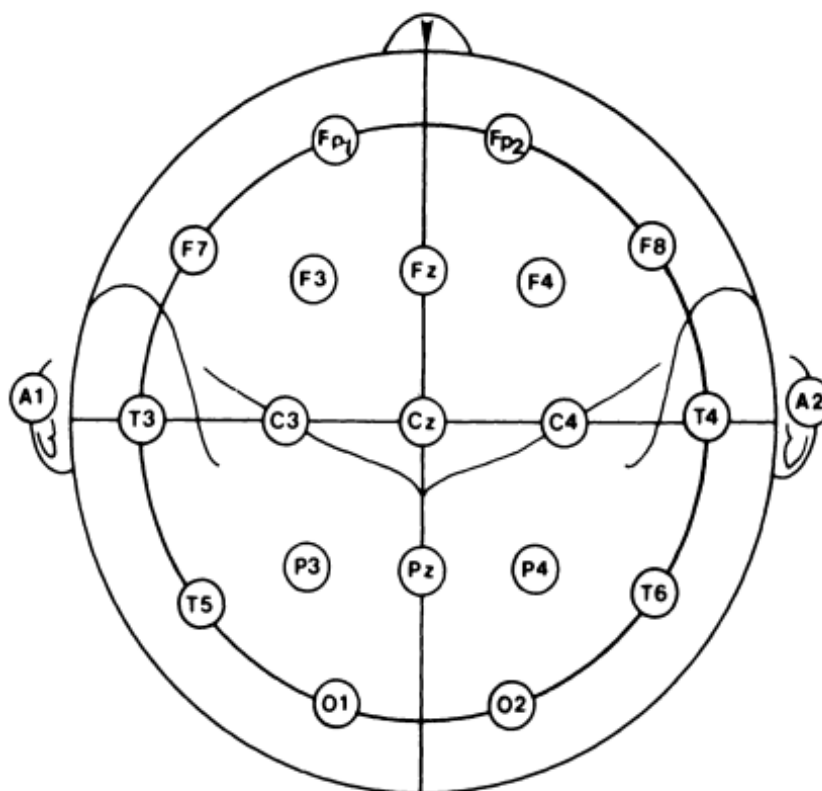


Figura 6 – Forma de montagem dos eletrodos para uma EEG (DUFFY et al., 1990)

resultados na detecção dos estágios do sono utilizando o canal Pz-Oz como pode ser visto na Fig. 7 (LUCEY et al., 2016) (HASSAN; SUBASI, 2017).

As leituras deste canal de nada serviriam se não fosse possível identificar as frequências de ondas presentes no sinal. Desta forma é necessário utilizar de ferramentas matemáticas como a transformada de Fourier para extrair informações do espectro do sinal.

## 2.3 Análise dos sinais

### 2.3.1 Análise dos sinais no domínio do tempo

Os sinais descrevem algum fenômeno físico, como por exemplo o sinal elétrico. Este sinal elétrico é representado matematicamente como uma função na qual a sua principal variável dependente a ser considerada é o tempo. O sinal elétrico quando captado, é descrito no domínio do tempo, ou seja, a forma como esse sinal se comporta conforme o decorrer de um determinado tempo. Os sinais enquanto no domínio do tempo, podem ser classificados em dois tipos (OPPENHEIM; WILLSKY; NAWAB, 1996):

- Sinal de tempo contínuo: A variável independente neste caso é contínua, ou seja,

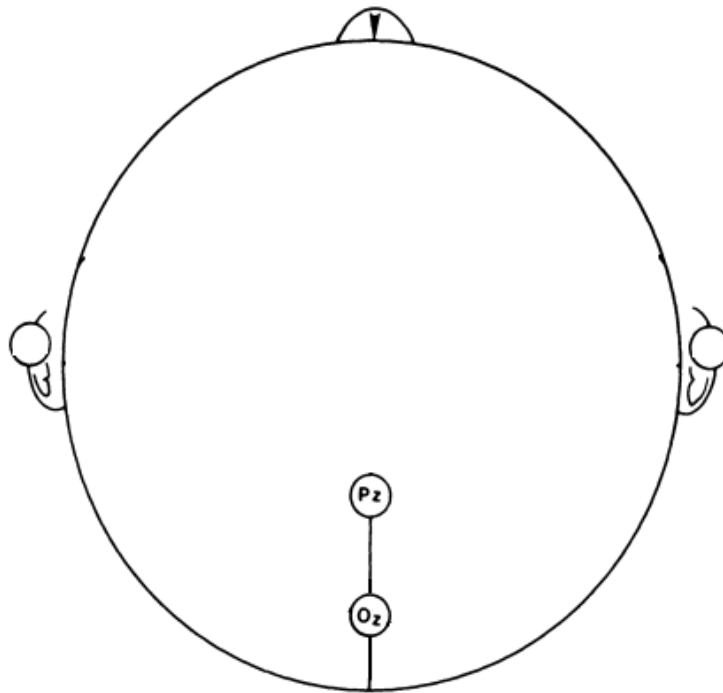


Figura 7 – EEG de canal único Pz-Oz modificado de (DUFFY et al., 1990)

para qualquer instante de tempo escolhido, a função que descreve o comportamento do sinal retornará um valor. Exemplo: Fig. 8a.

- Sinal de tempo discreto: A variável independente neste caso é discreta, ou seja, ela percorre o sinal apenas em valores inteiros, assim, somente é possível escolher determinados períodos de tempo para saber o valor retornado nesta função. Um sinal discreto pode ser proveniente de um sinal de tempo contínuo que passou por um processo de amostragem. Exemplo: Fig. 8b.

Um parâmetro importante que é possível se retirar de qualquer tipo de sinal contínuo ou discreto é a energia e potência média deste sinal. A energia de um sinal é a integral, para sinais contínuos, ou o somatório, para sinais discretos, das amplitudes do sinal obtido. A potência do sinal pode ser calculada dividindo a energia obtida pelo comprimento do intervalo, em sinais contínuos, e pelo número de pontos no intervalo, em sinais discretos. As equações para obtenção da energia e potência dos sinais contínuo e discreto, estão respectivamente descritos nas Eq. 2.1, 2.2, 2.3, 2.4 (OPPENHEIM; WILLSKY; NAWAB, 1996).

$$E = \int_{t_1}^{t_2} |x(t)|^2 dt \quad (2.1)$$



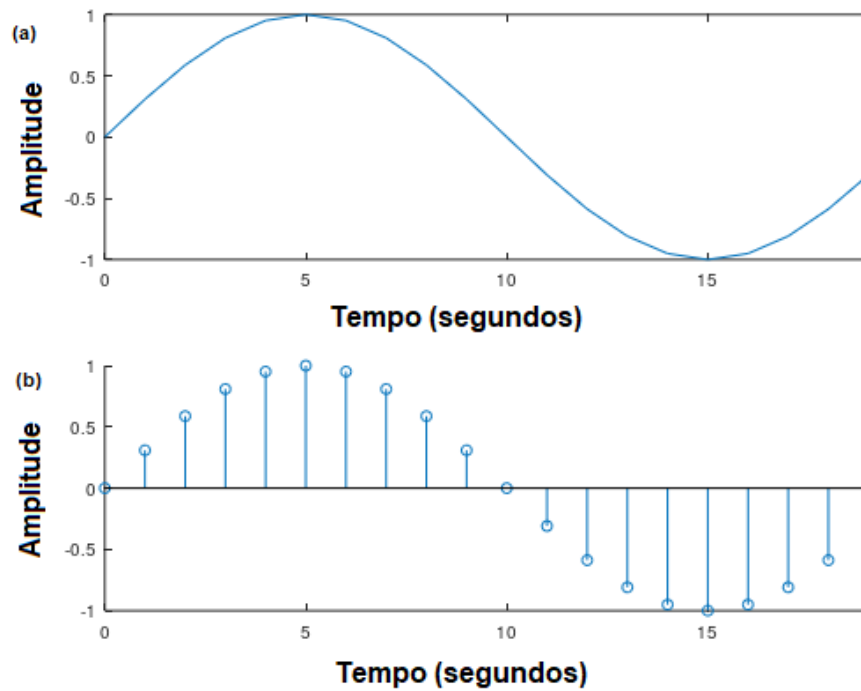


Figura 8 – Representação gráfica de (a) sinais de tempo contínuo e (b) Sinal de tempo discreto gerado por amostragem do sinal (a)

$$P = \int_{t_1}^{t_2} \frac{1}{t_2 - t_1} |x(t)|^2 dt \quad (2.2)$$

$$E = \sum_{n=n_1}^{n_2} |x(n)|^2 \quad (2.3)$$

$$P = \sum_{n=n_1}^{n_2} \frac{1}{(n_2 - n_1) + 1} |x(n)|^2 \quad (2.4)$$

### 2.3.2 Discretização de um sinal contínuo no tempo

Um sinal de tempo contínuo pode ser totalmente representado por um sinal de tempo discreto se aplicado se a discretização for aplicada de forma correta. Se um sinal for limitado em banda e seu número de amostras for maior que o dobro da frequência mais alta presente no sinal, então as amostras especificam unicamente tal sinal, e podemos reconstruí-lo perfeitamente (OPPENHEIM; WILLSKY; NAWAB, 1996).

Uma das formas de discretização de um sinal é a multiplicação deste sinal por um trem de impulsos como pode ser visto na Fig. 9. Sendo possível transformar um sinal que está em um tempo contínuo para um sinal em tempo discreto, a forma mais rápida computacionalmente de se obter as frequências presentes nesta onda é a utilização da *Fast*

*fourier Transform* (FFT), um algoritmo que retorna de forma rápida o sinal representado no domínio da frequência (OPPENHEIM; WILLISKY; NAWAB, 1996).

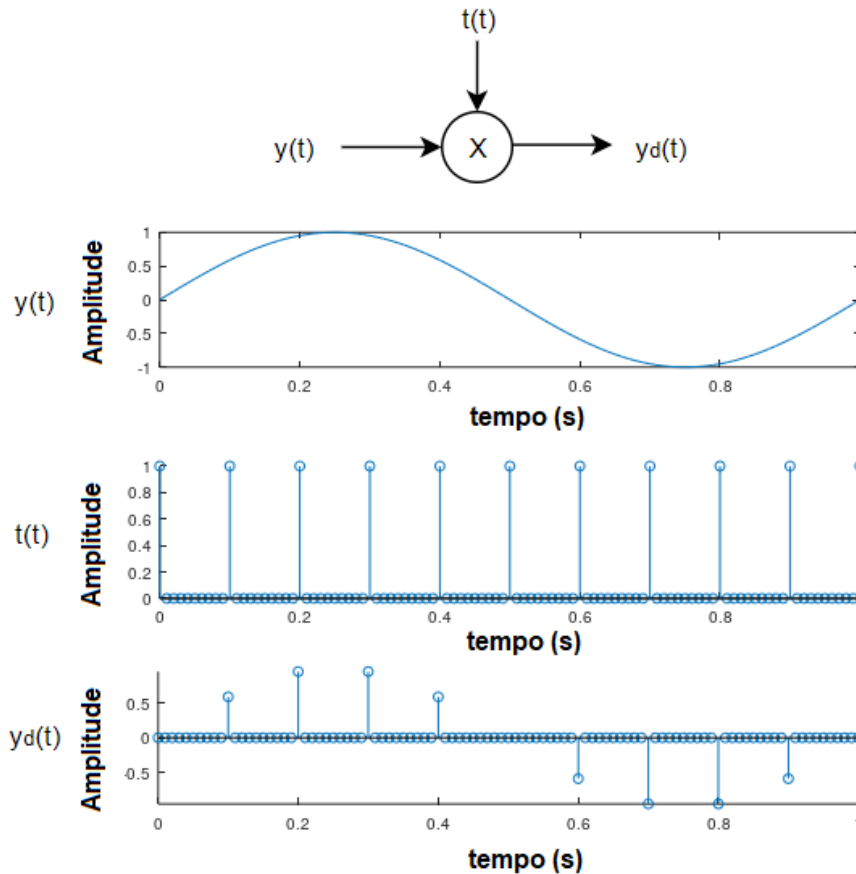


Figura 9 – Sinal contínuo discretizado com uma taxa de 10 amostragens por segundo

### 2.3.3 Análise de sinais discretos no domínio da frequência

O uso do domínio do tempo para tratar de sinais ondulatórios pode ser um tanto quanto confuso, pois dentro de um único sinal, é possível que esteja contido diversas ondas com frequências e amplitudes diferentes. Este sinal com várias ondas somadas pode gerar um sinal com características visíveis na Fig. 10. Para entender melhor as características deste sinal é possível utilizar a Transformada Rápida de Fourier

#### 2.3.3.1 Transformada Rápida de Fourier

A decomposição de um sinal periódico em outros sinais periódicos, pode ser feito com a Transformada Discreta de Fourier Eq. 2.5. Porém a execução desta equação para a obtenção do sinal no domínio da frequência pode ser muito demorada dependendo do número de termos ou da função sendo calculada. Para se obter agilidade nos cálculos,

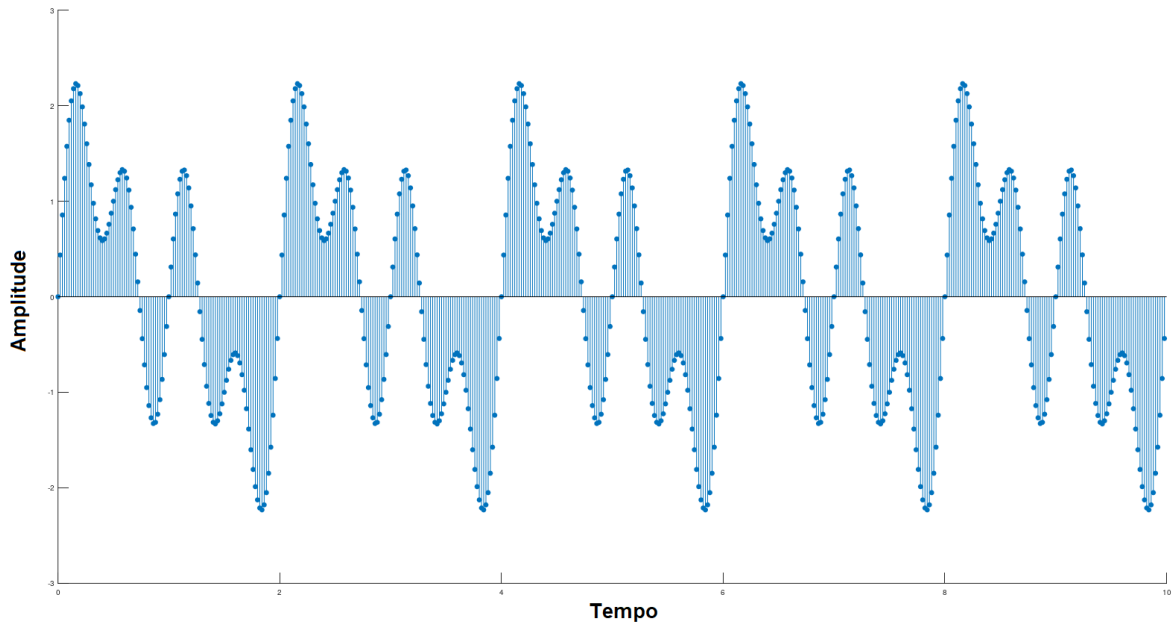


Figura 10 – Sinal discretizado com a presença de sinais com 2, 1 e 0.5 Hz

utiliza-se a Transformada Rápida de Fourier, do inglês *Fast Fourier Transform* (OPPENHEIM; WILLSKY; NAWAB, 1996).

$$X_k = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad (2.5)$$

A *Fast Fourier Transform* (FFT), é um algoritmo altamente utilizado no processamento de sinais, pois permite o cálculo da transformada de Fourier de modo ágil. Ela possui aplicação em vários campos, como por exemplo no da Engenharia Biomédica (COOLEY; LEWIS; WELCH, 1969).

Com a utilização desta ferramenta, é possível avaliar o sinal no domínio da frequência. O resultado obtido com a FFT geralmente são números complexos nos quais a magnitude deles com a devida manipulação, ficam situadas exatamente sobre as frequências presentes em um sinal, assim, conseguindo fornecer informações de quais frequências estão presentes (COOLEY; LEWIS; WELCH, 1969).

Quando se utiliza a Eq. 2.5 para realizar o cálculo da transformada, é necessário calcular  $k \times N$  vezes, pois para  $k$  da transformada, são feitas  $N$  operações, e como  $k = N$ , o número de operações calculadas será  $O(N^2)$ , sendo  $O(x)$  uma forma de descrever que se está avaliando na notação Grande-O, notação na qual somente interessa os extremos de uma função. Com o uso da FFT o número de operações se torna  $O(N \log_2(N))$ . Para uma somatória com um número pequenos de termos, o tempo para a realização do cálculo da transformada utilizando o método da Eq. 2.5 não é muito grande, porém, quando se possui vários termos, a velocidade é prejudicada (ALLSIGNALPROCESSING, 2015).

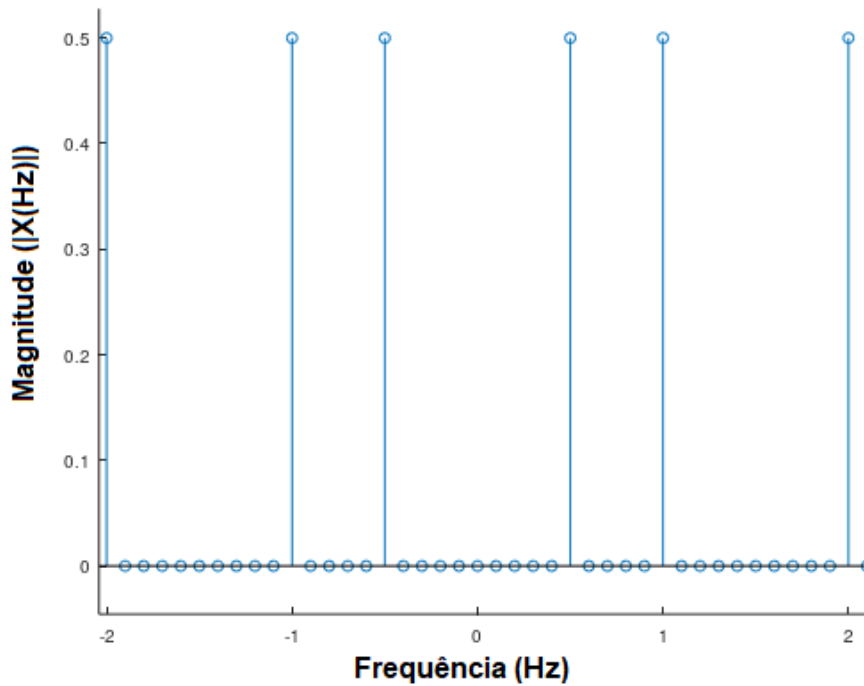


Figura 11 – Magnitude do sinal da Fig. 10 no domínio da frequência

Essa redução de tempo é possível devido ao algoritmo de Cooley-Tukey. Este algoritmo tem como proposta decompor a transformada de Fourier em versões com menos termos que podem ser calculadas rapidamente. No cálculo deste algoritmo o primeiro passo é considerar que,

$$W_N = e^{\frac{-j2\pi}{N}} \quad (2.6)$$

Resultando em,

$$X_k = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad (2.7)$$

O segundo passo é dividir esta transformada em duas, separando-a em termos pares e ímpares, para isso, se considera que os termos pares podem ser descritos na Eq. 2.8 e os termos ímpares descritos na Eq. 2.9,

$$m = 0, 1, 2, 3, 4, \dots, \left(\frac{N}{2} - 1\right) \quad (2.8)$$

$$n_p = 2m \quad (2.9)$$

$$n_i = 2m + 1 \quad (2.10)$$

O resultado na transformada está conforme a Eq. 2.11

$$X_k = \sum_{m=0}^{\frac{N}{2}-1} x[n]W_N^{k(2m)} + \sum_{m=0}^{\frac{N}{2}-1} x[n]W_N^{k(2m+1)} \quad (2.11)$$

Neste ponto o esforço computacional para realizar o cálculo da transformada de Fourier diminuiu. Nele é realizado o máximo de decomposições possíveis das transformadas até que se obtenha várias transformadas nas quais o número de termos seja igual a 1. O número de divisões possíveis é dado pela Eq. 2.12,

$$p = \log_2(N) \quad (2.12)$$

O cálculo do custo computacional é dado por 2.13. Quando um sinal é amostrado com um número muito grande de amostras,  $N \log_2(N)$  se torna muito maior que  $N$ , assim, o custo computacional pode ser aproximado por  $N \log_2(N)$  (COOLEY; LEWIS; WELCH, 1969) (ALLSIGNALPROCESSING, 2015).

$$Custo = \frac{N}{2^p} + pN = N + N \log_2(N) \quad (2.13)$$

## 2.4 Redes Neurais Artificiais

O cérebro humano é altamente complexo, não linear e consegue processar de forma paralela vários tipos de informação e os responsáveis por isso são os neurônios. Tendo como base o cérebro, as redes neurais artificiais tentam traduzir para o mundo digital o que as redes neurais biológicas fazem, ou seja, tentam mimetizar a forma de processar as informações (HAYKIN et al., 2009).

O cérebro humano é composto por uma quantidade imensa de neurônios, o que se aproxima de uma média de 10 bilhões de neurônios, além deste número massivo de neurônios, a interconexão entre eles é muito alta, possuindo em torno de 60 trilhões de interconexões entre si. Essas interconexões são basicamente o que justificam o fato de o cérebro conseguir processar várias informações complexas do meio ambiente ao mesmo tempo com uma velocidade muito superior a de um computador que apesar de possuir porta lógicas que realizam operações em nanossegundos, não possuem esta quantidade de interconexões entre si (HAYKIN et al., 2009).

Os estímulos recebidos pelo corpo humano, que podem ser considerados como sinais de entrada de um sistema, são convertidos em sinal elétrico que servem como informação para o cérebro. Estes sinais elétricos fazem com que os neurônios tenham diferentes tipos de resposta que podem ser consideradas como a saída do sistema. O diagrama de blocos da Fig 12 mostra como ocorre este processo, nele, as setas da esquerda pra direita indicam o

processo de resposta, e as setas da direita pra esquerda indicam o processo de realimentação do sistema, ou seja, a resposta de saída do sistema também tem influência sobre o sistema (HAYKIN et al., 2009).

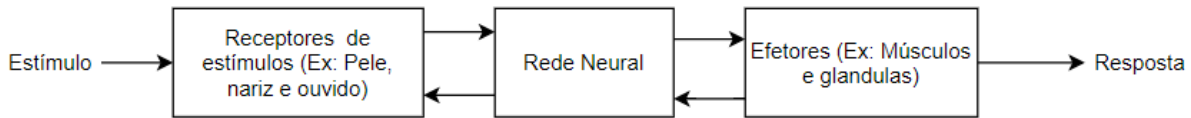


Figura 12 – Funcionamento do sistema nervoso

Dito isto, para o entendimento de como funciona uma rede neural é necessário falar sobre o funcionamento do neurônio.

### 2.4.1 O Neurônio

O neurônio é um processador unitário de informações composto por três unidades básicas que estão de acordo com a Fig. 13: O corpo celular, o axônio e os dendritos. O Axônio é a linha de transmissão dos sinais elétricos dos dendritos para o terminal sináptico. Os dendritos são os receptores dos sinais elétricos provenientes de outros neurônios conectados a ele, transmitindo para o corpo da célula. O corpo celular é onde é feito o processamento dos sinais elétricos provenientes dos dendritos (HAYKIN et al., 2009).

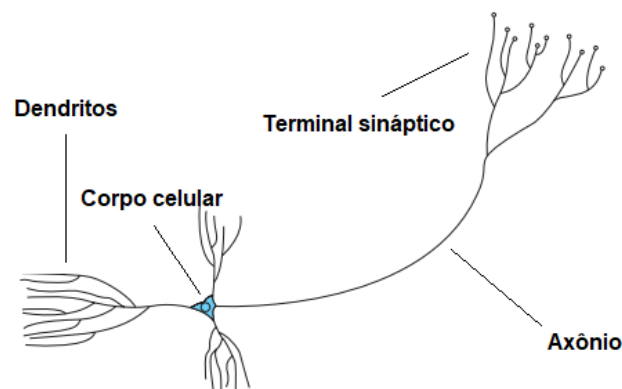


Figura 13 – Composição de um neurônio modificado de (HAYKIN et al., 2009)

O processamento feito dentro do núcleo de um neurônio biológico pode assumir o modelo matemático descrito de acordo com o diagrama da Fig. 14 e a equação resultante deste diagrama, Eq. 2.14. Este neurônio artificial que visa simular um neurônio biológico é chamada de perceptron.

Dentro do núcleo deste perceptron encontram-se:

- Pesos sinápticos: Um sinal  $x_j$  é multiplicado por um peso  $w_{kj}$  e então chega em um neurônio  $k$ . Estes pesos contribuem para a ativação do neurônio, e podem ser reajustados com a realimentação do sistema.
- Off-set: Valor que faz parte da somatória e não tem influência de outros sinais. Funciona como um valor mínimo para a função de ativação.
- Somador: Faz a soma do Off-set com todos os sinais  $x_j$  que já foram multiplicados por seus respectivos pesos  $w_{kj}$ .
- Função de ativação: Regular amplitude do sinal que o neurônio transmitirá para o próximo.

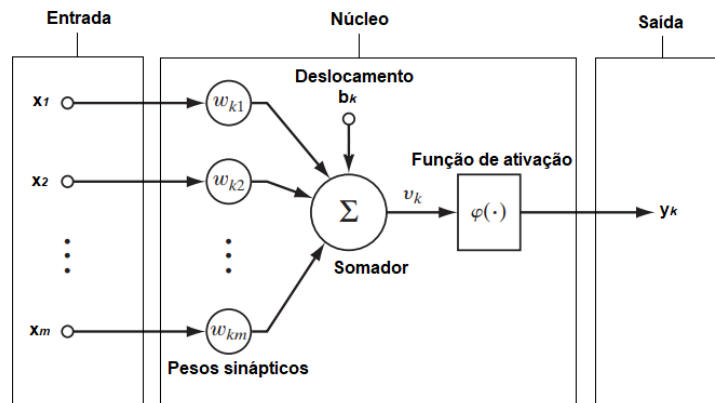


Figura 14 – Camadas de um perceptron modificado de (HAYKIN et al., 2009)

$$y_k = \varphi\left(\sum_{j=1}^m w_{kj}x_j + b_k\right) \quad (2.14)$$

#### 2.4.1.1 Função de ativação do perceptron

Das função de ativação a serem implementadas em um neurônio artificial, as que reproduzem resultados não binários são as mais comumente utilizadas pelo fato de o seu resultado da saída poder receber mais de um valor, como por exemplo entre -1 e 1 na função tanh, gerando uma melhor adaptabilidade da rede aos parâmetros de entrada. A função tanh possui um bom equilíbrio entre um comportamento linear e não linear (HAYKIN et al., 2009). O gráfico mostrado na Fig. 15 mostra como é o comportamento desta função.

A função de ativação tanh é a tangente hiperbólica de um determinado valor, logo, temos:

$$\varphi = \tanh(p) \quad (2.15)$$

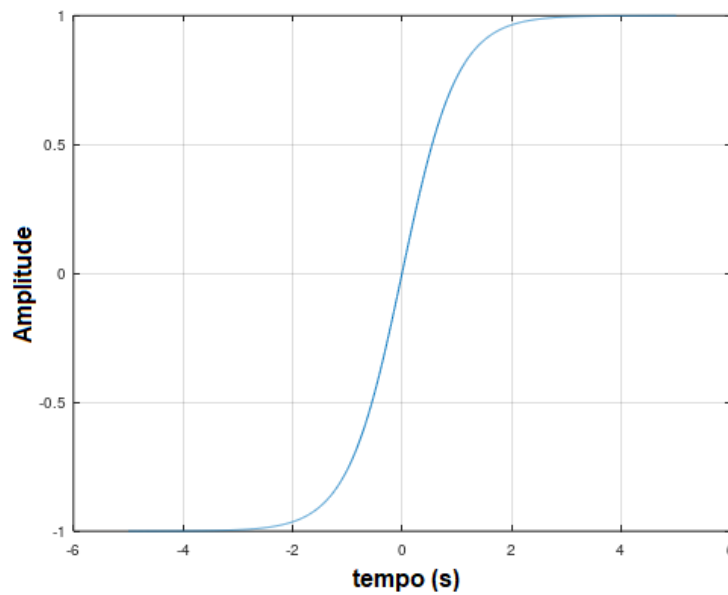


Figura 15 – Comportamento da função tanh

Se observarmos o limite inferior e o limite superior desta função, os valores possíveis de se obter estão dentro do intervalo que vai de -1 até +1. Por receber valores com magnitudes diferentes como parâmetros de entrada, após a normalização dos dados, alguns valores se tornam muito próximos a zero, desta forma, utilizar a função tanh ao invés da sigmoide se torna mais vantajoso, pois, como a função sigmoide satura em 0, algumas entradas podem rapidamente saturar devido ao valor das *features* estarem perto de 0.

$$\lim_{p \rightarrow +\infty} \tanh(p) = 1 \quad (2.16)$$

$$\lim_{p \rightarrow -\infty} \tanh(p) = -1 \quad (2.17)$$

Desta forma, a forma final para a equação que descreve o neurônio para este tipo de função está conforme a Eq. 2.18.

$$y_k = \tanh\left(\sum_{j=1}^m w_{kj}x_j + b_k\right) \quad (2.18)$$

Outro tipo de função de ativação que se encaixa neste trabalho é a função ELU, pois apresenta bons resultados quando os valores de entrada a serem trabalhados pela rede neural são provenientes de resultados gerados pela FFT (BAUMANN et al., 2012). De forma análoga a função RELU, a função ELU tem sua vantagem em não ser retificada, assim, as chances do perceptron deixar de ajustar seus pesos durante o treinamento são



menores. A função ELU possui um comportamento de acordo com a Eq. 2.19 e a Fig.16.

$$f(p) = \begin{cases} p & \text{se } p > 0 \\ (e^p - 1) & \text{se } p < 0 \end{cases} \quad (2.19)$$

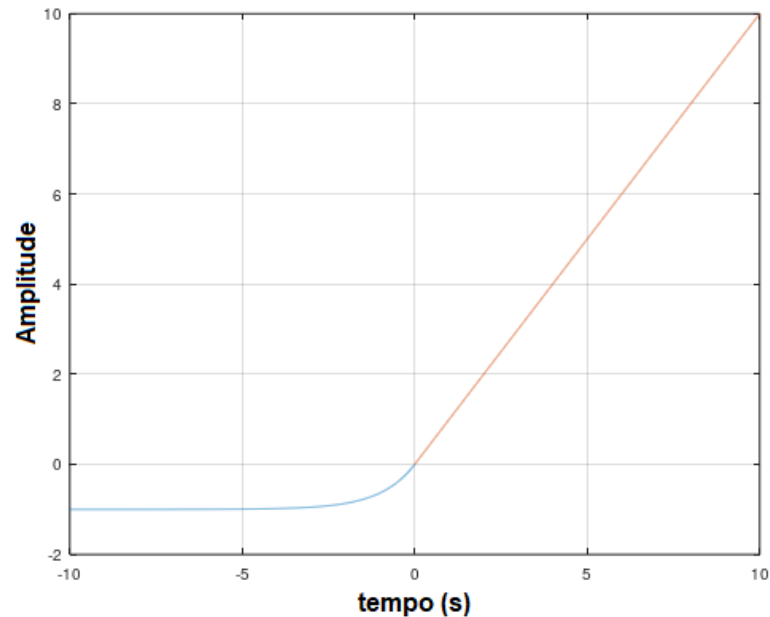


Figura 16 – Comportamento da função ELU. A curva em azul representa o comportamento não linear da função para números negativos, possuindo sua saturação em -1. A linha em vermelho representa o comportamento linear da função

Logo, se utilizarmos a função ELU como função de ativação, a equação matemática para o neurônio será,

$$f\left(\sum_{j=1}^m w_{kj}x_j + b_k\right) = \begin{cases} \sum_{j=1}^m w_{kj}x_j + b_k & \text{se } \sum_{j=1}^m w_{kj}x_j + b_k > 0 \\ (e^{\sum_{j=1}^m w_{kj}x_j + b_k} - 1) & \text{se } \sum_{j=1}^m w_{kj}x_j + b_k < 0 \end{cases} \quad (2.20)$$

Outra função de ativação a ser utilizada pensando em uma rede neural do tipo *Multilayer Perceptron* (MLP), é a função *softmax*. Esta função apresenta aos neurônios de saída da rede MLP a probabilidade de cada um deles ser o resultado esperado para aqueles parâmetros de entrada. Possuindo como base funções sigmoidais, sendo então possível assumir qualquer valor entre 0 e 1, é escolhido como a resposta da rede neural o perceptron da camada de saída que se aproximar mais do número 1. Desta forma, a função *softmax* se torna essencial para realização da classificação dos possíveis estágios do sono.

Para exemplificar melhor o que seriam os perceptrons presentes na camada de saída, é necessário explicar um pouco sobre redes neurais multilayer perceptrons.

## 2.4.2 Rede Neural Multilayer Perceptron(MLP)

Como o próprio nome sugere, são várias camadas de perceptrons, onde estes perceptrons fazem várias conexões entre si que se entrelaçam, dando então a razão de ser chamado de rede. Este modelo de rede pertence a uma classe chamada feedforward, pois o sinal de entrada sempre é propagado para frente.

A forma na qual os neurônios se conectam pode ser visto na Fig. 17, nela é possível ver que uma rede neural MLP é dividida em três partes. A primeira parte é composta pela camada de entrada, é por ela que os sinais externos adentram na rede neural. A segunda parte é composta por camadas ocultas, recebem esse nome por não serem vistas diretamente, e funcionam como intermédio entre a camada de entrada e a camada de saída. A terceira parte é a camada de saída, ela é a apresentação do resultado final das propagações de sinal que aconteceram ao longo da camada oculta (HAYKIN et al., 2009).

O modelo de rede MLP segue três regras:

- perceptrons da mesma camada não se conectam
- Todos os perceptrons de uma camada se conectam com todos os perceptrons da camada seguinte mas não se conectam com a camada anterior.
- É composto por ao menos duas camadas, entrada e saída.

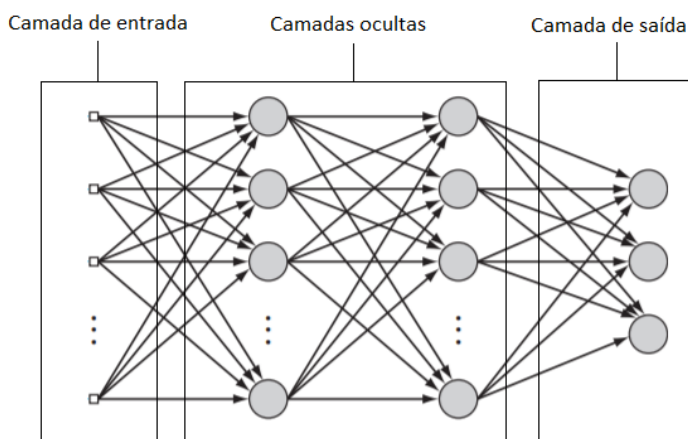


Figura 17 – Camadas de uma rede MLP

De forma geral, uma rede neural possui dois tipos de sinal que são de interesse, um deles é o sinal que é propagado camada a camada pelos perceptrons a partir dos sinais de entrada, o outro tipo de sinal é o sinal de erro. O sinal serve como parâmetro para o ajuste dos pesos  $w_{kj}$ , ou seja, é o sinal que é utilizado para que seja feita a aprendizagem na rede neural.

### 2.4.3 Aprendizagem da rede neural

Redes neurais apresentam dois tipos de aprendizagem, são elas, a aprendizagem não supervisionada e a aprendizagem supervisionada.

A aprendizagem não supervisionada visa pegar os elementos apresentados a rede neural e separá-los em grupos que apresentem as mesmas características, o nome deste processo é chamado de clusterização.

No aprendizado supervisionado as saídas geradas pela rede neural são supervisionadas para saber se o resultado está de acordo com a resposta desejada, este processo visa ou tentar prever um resultado para aquela entrada, ou classificar os parâmetros de entrada. Por atender melhor aos requisitos deste trabalho, o tipo que será adotado é o de classificação.

Na classificação, se o resultado não estiver como o desejado, é feito um ajuste nos pesos de forma que o resultado na saída seja o mais próximo do desejado. Este ajuste é feito várias vezes por meio da propagação do erro para as camadas anteriores até que não seja mais notado mudanças significativas nos pesos, o que seria quando uma rede neural chega ao limite de seu aprendizado. Uma forma de apresentar o comportamento deste tipo de aprendizagem é por meio do diagrama da Fig. 18.

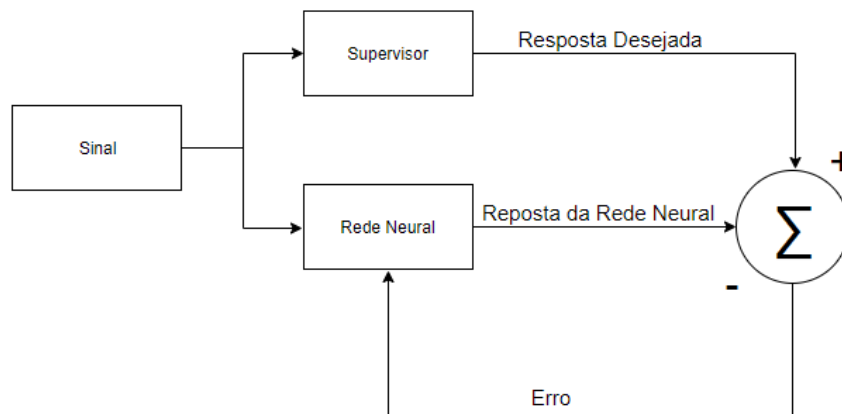


Figura 18 – Diagrama do funcionamento do Aprendizado Supervisionado

A forma utilizada para propagar este erro de volta para a rede neural é chamado de backpropagation (HAYKIN et al., 2009). Para questões de simplificação, os cálculos serão demonstrados para uma rede neural sem camadas ocultas.

Primeiramente é necessário calcular o erro gerado pelo sistema. O erro  $e_i$  de um neurônio  $j$  é a diferença do resultado esperado  $d_j$  para aquele neurônio com o resultado obtido  $o_j$  naquele neurônio,

$$e_j = d_j - o_j \quad (2.21)$$

em seguida é necessário calcular o gradiente de cada neurônio que é o parâmetro que define se o peso deve aumentar ou diminuir e com qual intensidade isso deve ocorrer (HAYKIN et al., 2009).

$$\delta_j = f'_j \left( \sum_{j=1}^m w_{kj} x_j + b_k \right) e_j \quad (2.22)$$

Com o gradiente calculado, é necessário calcular a taxa de aprendizado que é melhor aproximada como uma constante arbitrária que varia com o tempo, ou melhor, com as iterações,

$$\eta(n) = \frac{c}{n} \quad (2.23)$$

Com tudo definido, é possível calcular um novo valor para o peso da seguinte forma,

$$w'_{kj} = w_{kj} + \eta * \delta_j * x_j \quad (2.24)$$

$w'_{kj}$  é o novo valor do peso,  $w_{kj}$  é o valor antigo do peso,  $\eta$  é a taxa de aprendizagem,  $\delta_j$  é o gradiente local daquele peso e  $x_j$  é o sinal de entrada do neurônio.

Com o peso recalculado, deve-se também recalculer o *bias* da seguinte forma

$$b'_j = b_j + \eta * \delta_j \quad (2.25)$$

Como dito, esse ajuste de peso deve ser feito até que os valores convirjam a um ponto no qual a variação do peso novo com o peso antigo seja muito pequena, admitindo que a rede neural não consegue evoluir mais em sua aprendizagem (HAYKIN et al., 2009).

## 3 Metodologia

Este capítulo tem como objetivo desenvolver a forma de integração de todo o conhecimento que foi descrito anteriormente para a concretização dos resultados esperados.

Para realizar a caracterização comportamental do sono, é necessário fazer a análise de sinais biomédicos, sendo o sinal escolhido para a definição, o sinal obtido por meio da eletroencefalografia. Em posse de dados de um EEG, é necessário dar estes dados um tratamento para que a categorização do estágio do sono seja facilitada. A forma encontrada para análise do sinal é observar quais as frequências dominantes no sinal durante cada ciclo do sono por meio da utilização da transformada de Fourier, além de realizar cálculos da energia do sinal em um determinado período de tempo. As informações retiradas do sinal serão analisadas por redes neurais que por meio de uma forma de aprendizagem supervisionada, aprenderão e conseguirão categorizar os estágios do sono de uma pessoa, sendo possível então, realizar a montagem de um hipnograma.

Com uma visão geral do fluxo do processo descrita, é necessário exemplificar e demonstrar como cada passo será realizado.

### 3.1 Aquisição dos Dados

O banco de dados dos sinais que serão utilizados para treino e teste das redes neurais são provenientes do site *PhysioNet* (GOLDBERGER et al., 2000). Este site possui um banco de dados de polissonografia no qual é possível extrair somente os dados de sinal de EEG do exame, e o estágio do sono em determinado período do sono do paciente. Os dados presentes no site estão em formato *.EDF*, para que seja possível analisá-los, é necessário fazer uso de softwares como o *EDFbrowser* que permite fazer a conversão dos dados presentes em um arquivo *.EDF* para um arquivo de texto no formato *.txt*.

Os dados de interesse são o estágio no qual o paciente estava em determinado horário, e sinais de EEG advindos de um canal no qual se tem uma boa performance na categorização dos estágios do sono, o canal Pz-Oz. Os dados de sinal de EEG já em formato de texto, são modificados de forma a atender os requisitos para a leitura de arquivos com Python, utilizando as bibliotecas *scipy* e *numpy*.

As variáveis presentes no arquivo de texto com dados de EEG são: (a) tempo em s; (b) intensidade do sinal elétrico em  $\mu V$ . As variáveis que estão presente no arquivo de texto com os estágios de sono são: (a) Os estágios do sono; (b) O horário no qual determinado estágio do sono foi atingido.

Os dados utilizados provenientes da *Physionet*, são de pessoas com ausência de

sinais de doenças e sem o uso de qualquer tipo de medicação em uma faixa etária entre 21 e 35 anos, sendo o sinal de EEG destes pacientes provenientes do canal Pz-Oz, e com taxa de amostragem de 100Hz. Nestes pacientes, a classificação dos estágios do sono foram feitas por especialistas de forma manual(Kemp et al., 2000).

## 3.2 Processamento de Dados

Esta parte do processamento dos dados adquiridos do banco de dados da *PhysioNet* pode ser dividido em dois momentos. No primeiro momento será feita a extração de informação dos sinais de EEG(GOLDBERGER et al., 2000). No segundo momento será feito por parte da rede neural, a determinação do estágio do sono.

### 3.2.1 Organização e formatação dos dados

Após baixar os dados em formato *.edf* do banco de dados da *Physionet*, iniciou-se uma alteração destes dados de forma que ficassem melhor utilizáveis neste trabalho.

O primeiro passo a ser dado nas alterações dos dados é reduzir o tamanho de amostras contidas para cada paciente. Por não haver interesse em observar o comportamento do paciente ao longo do dia, as bases de dados foram resumidas apenas ao período no qual o paciente se encontra dormindo. Assim, por paciente analisado foram gerados arquivos de texto em formato *.txt* com um número de amostras que abrange um período que começa minutos antes do paciente entrar em estado de sono até alguns minutos depois do paciente sair do estado de sono, dessa forma, além das amostras referentes ao estado de sono, tem-se também várias amostras do paciente em estado de vigília, o que também é necessário para para o aprendizado da rede neural.

Desta forma, das classificações feitas durante o sono de cada um dos pacientes, o número de amostras para cada estágio de sono fica dividido de acordo com a Tabela 4.

Paciente	Quantidade de amostras por estágio				
	Vigília	N1	N2	N3	REM
1	570	59	373	297	215
2	491	92	660	96	176
3	566	117	607	25	109
4	839	18	396	111	151
<b>Total</b>	2466	286	2036	529	651

Tabela 4 – Quantidade de amostras por estágio do sono

Os dados dos pacientes 1,2,3 e 4 foram unificados para a criação de um dataset maior que tem como objetivo treinar a rede de forma que seu aprendizado fique generalizado. Após esta unificação, os dados serão separados entre treino, validação e teste, sendo que cada um representa respectivamente 75%, 12% e 13% do dataset.

	Treino	Validação	Teste	Total
<b>Número de Amostras</b>	4448	746	774	5968
<b>Parcela total do Dataset (%)</b>	75	12	13	100

Tabela 5 – Separação do dataset

Após separar, organizar e as amostras, iniciou-se o processo de processamento de sinais para servir como dado de entrada para a rede neural.

### 3.2.2 Extração de Informação do Sinais de EEG

O sinal de EEG apresenta informações sobre o comportamento elétrico da superfície do cérebro humano. A primeira informação a ser extraída é proveniente do domínio do tempo, aqui, será calculada a potência média do sinal elétrico do cérebro em um determinado período de tempo, pois ela apresenta variações durante as diferentes fases do sono, sendo então, um bom parâmetro para a rede neural conseguir definir qual estágio do sono está relacionado àquele período, vide Eq. 2.4. A segunda informação a ser extraída é o espectro do sinal elétrico do cérebro, pois como já dito anteriormente, fases diferentes do sono apresentam frequências dominantes diferentes.

Para o cálculo da potência média e a extração do espectro do sinal, é necessário fazer uso de métodos computacionais para realizar estes cálculos. O método a ser utilizado neste trabalho é com o uso do Python e suas bibliotecas *scipy* e *numpy*.

Ao carregar o arquivo de texto para para um código feito em *Python*, é criada uma matriz de elementos que possuem as informações de tempo e amplitude do sinal de EEG. Com essa matriz é possível começar a realizar a extração de das informações do sinal. A Fig.19a mostra o comportamento da potência média do sinal de EEG.

Como já citado no referencial teórico, a análise do estágio do sono deve ser feita utilizando um intervalo de 30 segundos, assim, o tempo total de duração do exame no qual os dados foram extraídos, será dividido em subconjuntos de 30 segundos. Destes subconjuntos serão analisados a potência média e a densidade espectral, servindo como sinais de entrada para a rede neural. A Fig.19.b mostra o sinal no domínio da frequência, nela é possível notar que as maiores magnitudes que possuem um certo destaque vão de 0.4 até 5Hz, ou seja, um indicativo de ondas delta e teta neste intervalo.

Como produto destas análises no domínio da frequência, as informações que serão utilizadas como entrada para as redes neurais serão as somas das magnitudes em um determinado intervalo de frequência. Serão definidos 4 somatórios correspondentes ao nome e frequência das ondas cerebrais, sendo elas então, Delta  $\delta$ , Teta  $\theta$ , Alfa  $\alpha$  e beta  $\beta$ , respectivamente nos intervalos entre 0-4 Hz, 4-8 Hz, 8-12 Hz e 12-28 Hz, tal como

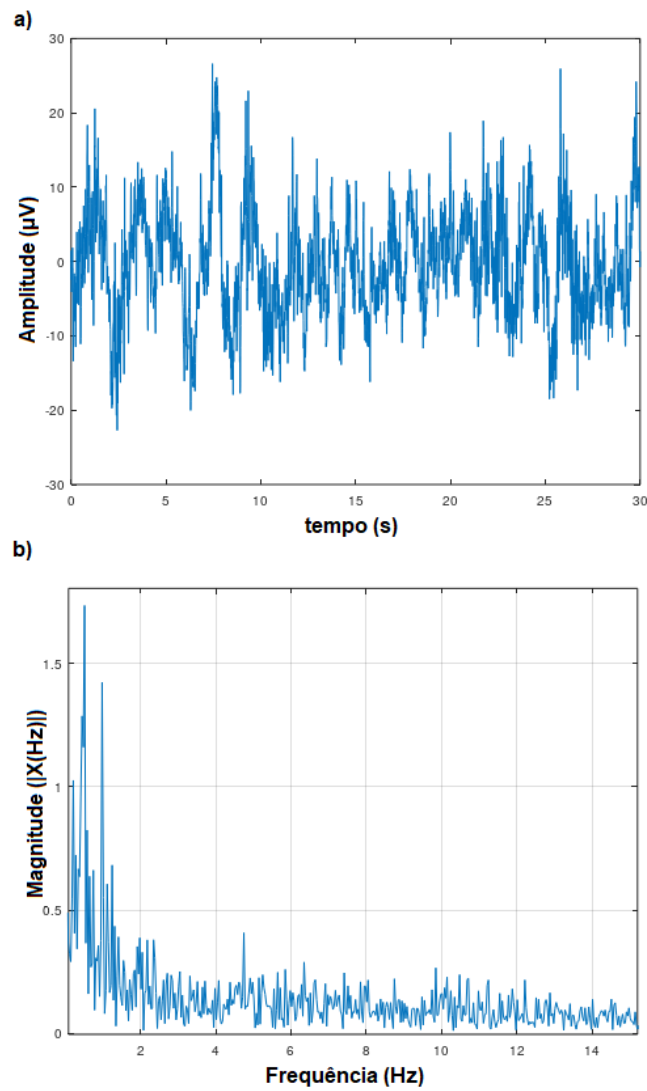


Figura 19 – Sinal de EEG no domínio do tempo (a) e frequência(b) em um período de 30 segundos

indicado para as seguintes equações,

$$\delta = \sum_{k=1501}^{1620} X_k \quad (3.1)$$

$$\theta = \sum_{k=1621}^{1740} X_k \quad (3.2)$$

$$\alpha = \sum_{k=1741}^{1860} X_k \quad (3.3)$$

$$\beta = \sum_{k=1861}^{2340} X_k \quad (3.4)$$



A explicação para os valores do intervalo ocorre da seguinte forma. Um intervalo de 30 segundo possui 3000 amostras, ou seja, a frequência de amostragem é de 100 amostras por segundo. Cada  $Hz$  possui 30 amostras, logo, para um intervalo de 4  $Hz$  são necessárias 120 amostras. Assim, é possível fazer a soma das magnitudes em um determinado intervalo de frequência. O motivo da contagem começar em 1501 se dá ao fato do uso da função *fft* da biblioteca *numpy*, com ela a posição 1500 marca a origem do eixo de frequências, o que vem antes da origem é um espelho dos valores presentes na parte positiva do eixo.

Utilizando *Python* e a biblioteca *numpy*, os dados foram carregados, processados e salvos em um arquivo em formato *.txt* para posterior leitura da rede neural. Os dados processados são relacionados com os estágios do sono, assim, é possível que a rede neural realize estágio de aprendizagem supervisionada. Um exemplo dos dados gerados para o *dataset* estão de acordo com Tabela 6.

Potência Média do Sinal	Somatórios				Saída
	0.4-4 Hz	4-8 Hz	8-12 Hz	12-28 Hz	Desejada
133.3958	26609.1839	26611.619	29476.1128	116935.0584	Acordado
128.7192	6231.0765	5840.2204	6221.4399	30617.7935	N1
394.7344	6967.2297	6605.4192	6950.048	33593.6217	N2
1256.6708	5901.8337	4807.1876	5488.1916	26788.5579	N3
139.9285	6787.2236	6518.6203	7039.2165	33447.8547	R

Tabela 6 – Separação do dataset

O código utilizado para realizar o processamento destes sinais se encontra presente no apêndice C. O fluxo de funcionamento do código ocorre de acordo com a Figura 20.

### 3.2.3 Determinação do Estágio do Sono

Com os parâmetros de entrada definidos, para finalizar o processamento dos dados, resta descrever uma rede neural de forma que ela consigo aprender a identificar os estágio do sono. A forma de implementação da rede neural se dará por meio da linguagem de programação *Python* com utilização do *Keras/Tensorflow*. O motivo da escolha desta linguagem em conjunto com estas bibliotecas é a sua grande participação na área de inteligência artificial e a grande comunidade desta área que participa de forma assídua para sempre promover melhorias em suas bibliotecas.

A biblioteca a ser utilizada é o *Keras/TensorFlow*, ela permite que as operações matemáticas e as conexões entre os neurônios seja feita de forma mais intuitiva e visual. Desta forma a implementação da rede neural ocorrerá da melhor forma possível, aumentando as chances dos resultados serem os melhores possíveis.

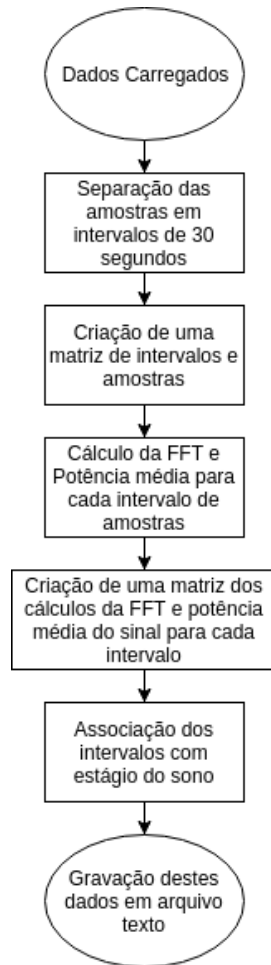


Figura 20 – Fluxo de funcionamento do código do cálculo da FFT e potência média do sinal

A forma que a rede neural trabalhará será da seguinte forma. Os sinais de entrada serão a potência média do sinal e as magnitudes dos quatro tipos de ondas cerebrais, o que leva a uma rede neural com 5 entradas. Essas entradas serão processadas na camada oculta que por fim mandará sinais para a camada de saída que poderá ativar 5 neurônios que determinam qual estado a pessoa se encontra, ou seja, cada neurônio representa os estados: Acordado, NREM1, NREM2, NREM3 e REM. Como o treinamento da rede neural é supervisionado, é necessário utilizar o arquivo de texto que descreve em qual estado a pessoa estava e em qual horário, desta forma, sabendo qual a saída desejada, a rede neural pode ajustar os pesos e *bias* de forma a encontrar na maioria das vezes a saída desejada.

Durante o treinamento, ao final de cada época, é feita uma validação para saber se a rede neural ainda está distante do resultado desejado. Se na validação o resultado não for bom o suficiente, a rede neural ajusta os valores de pesos e vieses, embaralha a ordem dos sinais utilizados para treino e faz o teste novamente. O número de épocas a serem rodadas depende da acurácia que se deseja obter.

### 3.2.4 Treinamento e Validação da Rede Neural

A rede neural descrita no *Keras* faz o uso dos arquivos de texto gerados pelos sinais processados do banco de dados. Estes arquivos de texto são carregados no código criado em *Python* e utilizados na camada de entrada da rede neural. Para tentar obter os melhores resultados possíveis, a rede foi treinada utilizando funções e tamanhos diferentes em sua estrutura.

A estrutura da rede neural é composta de 5 neurônios presentes na camada de entrada, representando os intervalos dos espectros de frequência e a potência média do sinal, 3 camadas ocultas com 64 neurônios em cada uma e 5 neurônios na camada de saída representando cada um dos estágios do sono e o estado de vigília. O Valor do *batch size* foi definido em 8 para que a propagação do erro para a rede fosse feita em sua maioria com labels do mesmo tipo. A quantidade de camadas ocultas e de neurônios presentes em cada uma delas, assim como os valores de épocas e *batch size* se deu pelo ajuste de forma empírica visando a melhora do aprendizado da rede em conjunto com a viabilidade do tempo de execução do treinamento da rede neural.

Devido a utilização da função *softmax*, todos os neurônios de saída receberam alguma valor de ativação. Para a classificação do estágio do sono, o neurônio que possuir o maior valor determina qual o estágio do sono para aqueles dados de entrada.

Assim, foram criados 4 grupos com diferentes funções de ativação em suas camadas ocultas como forma de tentar encontrar o modelo que melhor consiga categorizar os estágios do sono de um paciente. Os modelos são:

- 1º modelo: 3 Camadas ocultas com as respectivas funções de ativação para cada camada oculta: tanh, tanh e tanh;
- 2º modelo: 3 Camadas ocultas com as respectivas funções de ativação para cada camada oculta: elu, elu e elu;
- 3º modelo: 3 Camadas ocultas com as respectivas funções de ativação para cada camada oculta: elu, elu e tanh;
- 4º modelo: 3 Camadas ocultas com as respectivas funções de ativação para cada camada oculta: tanh, elu e elu.

Estas funções de ativação de cada modelo serão inseridos respectivamente em *activation*, *activation\_1* e *activation\_2* do grafo apresentado na Fig. 21.

Além do valor da perda e acurácia obtidos no treino, outro valor bastante importante é o valor obtido na acurácia e perda da validação. Esta observação do valor da

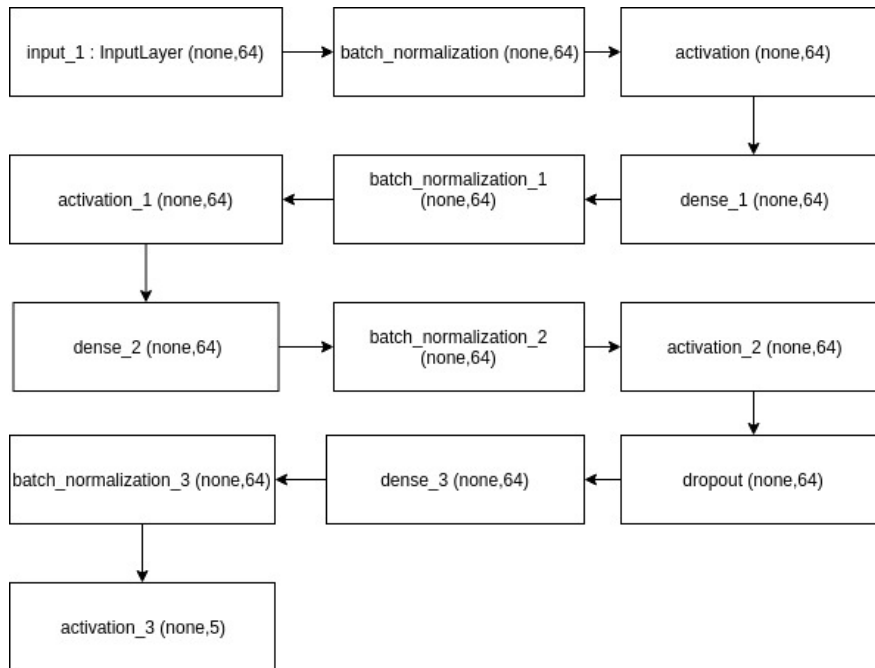


Figura 21 – Grafo da rede neural

acurácia da validação é imprescindível, pois, por meio de sua observação, é possível identificar se a rede está sofrendo *overfitting*, o que seria o caso do modelo estar se ajustando muito bem apenas para os dados de treino.

### 3.3 Montagem do Hipnograma

Após a estruturação, treino e validação da rede neural, a parcela do *dataset* separado para teste é carregada em um código em *Python*, que carregará o modelo salvo da rede neural treinada e realizará um *plot* do estágio do sono em função do tempo, realizando então, a montagem do hipnograma.

O fluxo completo a ser seguido para que seja possível fazer a montagem do hipnograma está descrito na Fig. 22.

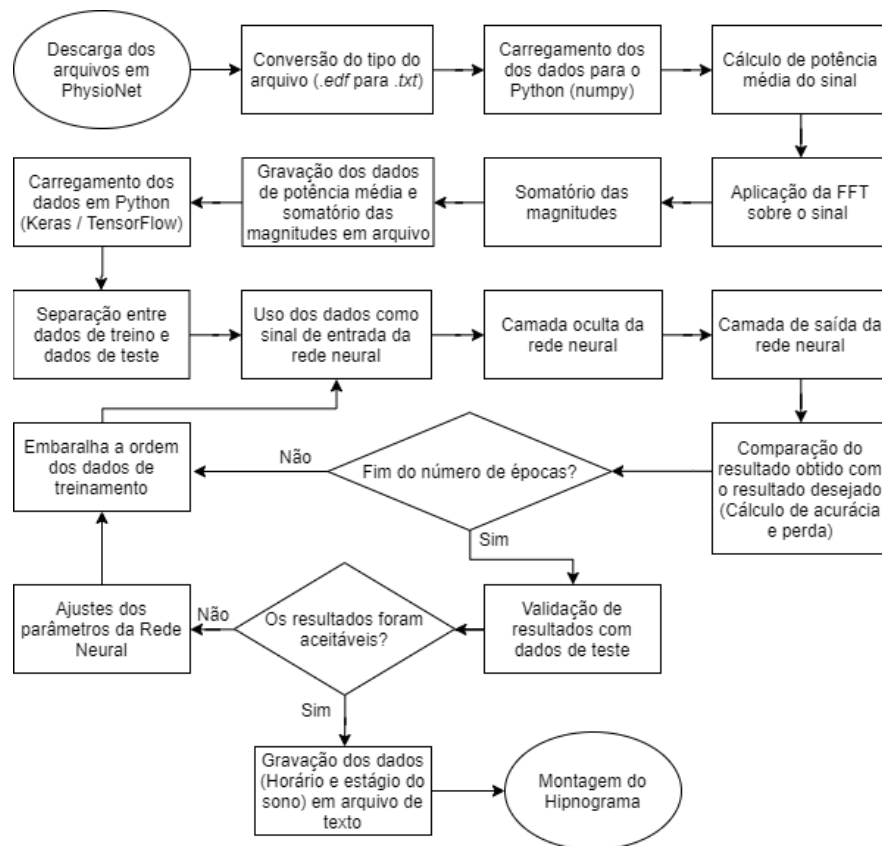


Figura 22 – Fluxograma do funcionamento da ferramenta

## 4 Resultados

Este capítulo tem como objetivo apresentar os resultados obtidos da execução da metodologia proposta.

### 4.1 Resultados de Treino e Validação

Obteve-se os seguintes resultados para cada modelo adotado.

Modelo	Acurácia Treino	Perda Treino	Acurácia Validação	Perda Validação
1	0.7069	0.8056	0.6649	0.8794
2	0.7431	0.7298	0.6769	0.9309
3	0.7455	0.7331	0.6850	0.8134
4	0.7351	0.7405	0.6944	0.9799

Tabela 7 – Acurácia e perda para cada modelo durante a validação

Os treinos e validações da rede neural mantiveram acurácias de treino na faixa de 70% à 74% e acurácias de validação na faixa de 66% à 69%. Por possuírem valores próximos, é esperado que as redes não sofreram de *overfitting*, porém, seu aprendizado pelos valores de perda e acurácia obtidos, é possível perceber que estes modelos de rede ficaram com seu aprendizado limitado.

Os valores de acurácia saturam em torno de 6 épocas, sendo então necessário realizar treinos com pouco mais de 6 épocas para que a rede não comece a ficar tendenciosa e tenha *overfitting*.

Os valores de perda oscilam para todos os modelos, estando com valores próximos à para todos.

A validação do sistema teve seus valores de acurácia e perda oscilando de forma abrupta para os modelos 1,2 e 4, sendo o terceiro modelo o que se manteve mais estável durante o treino.

Ao analisar os valores de cada modelo, espera-se que os modelos 3 e o 4 desempenhem melhor pela acurácia obtida em seus treinos e suas validações. O modelo 3 ainda se mostra o favorito em questão de desempenho, pois obteve os menores valores de perda tanto para treino quanto para validação.

Assim, após a criação dos 4 modelos treinados com o mesmo *dataset*, iniciou-se a fase de testes.

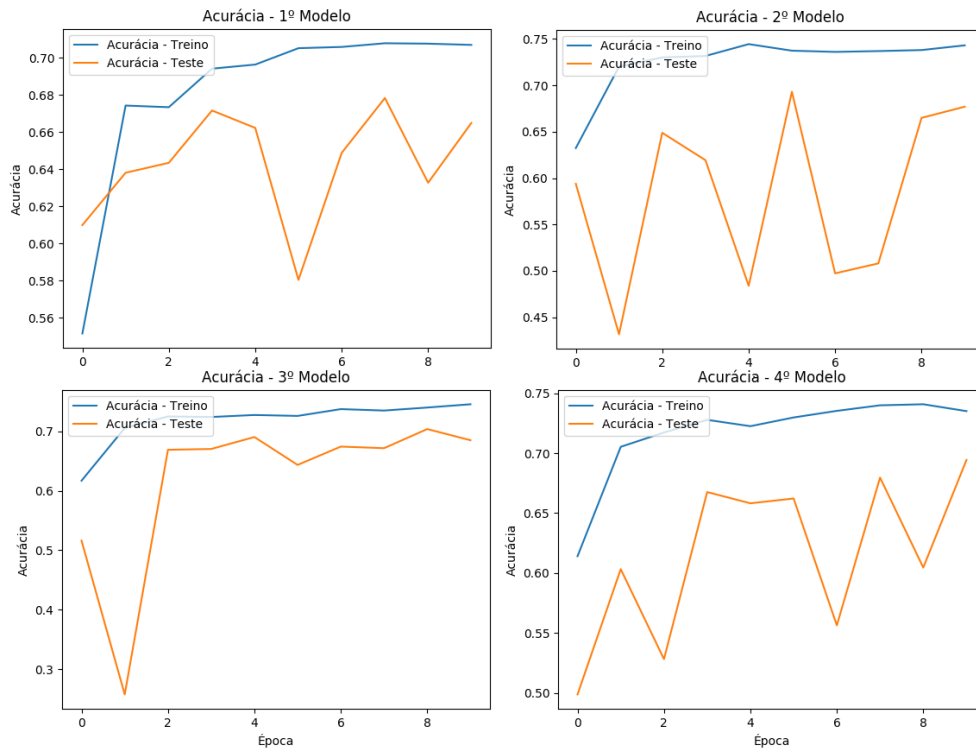


Figura 23 – Acurácia dos modelos treinados

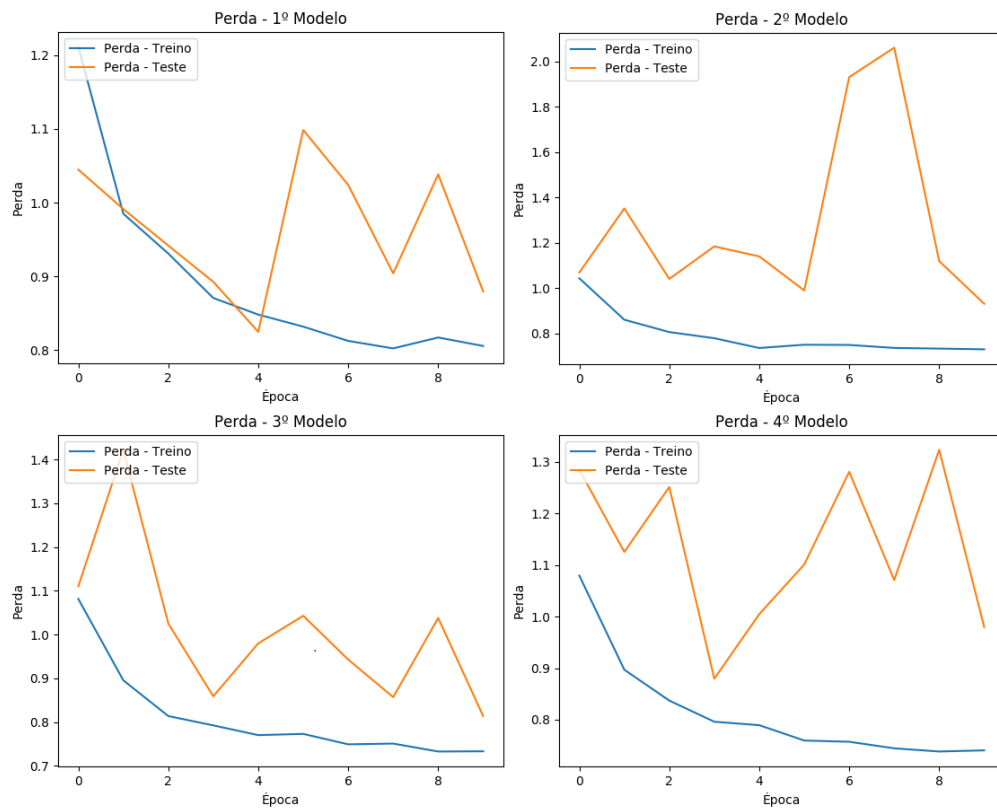


Figura 24 – Perda dos modelos treinados





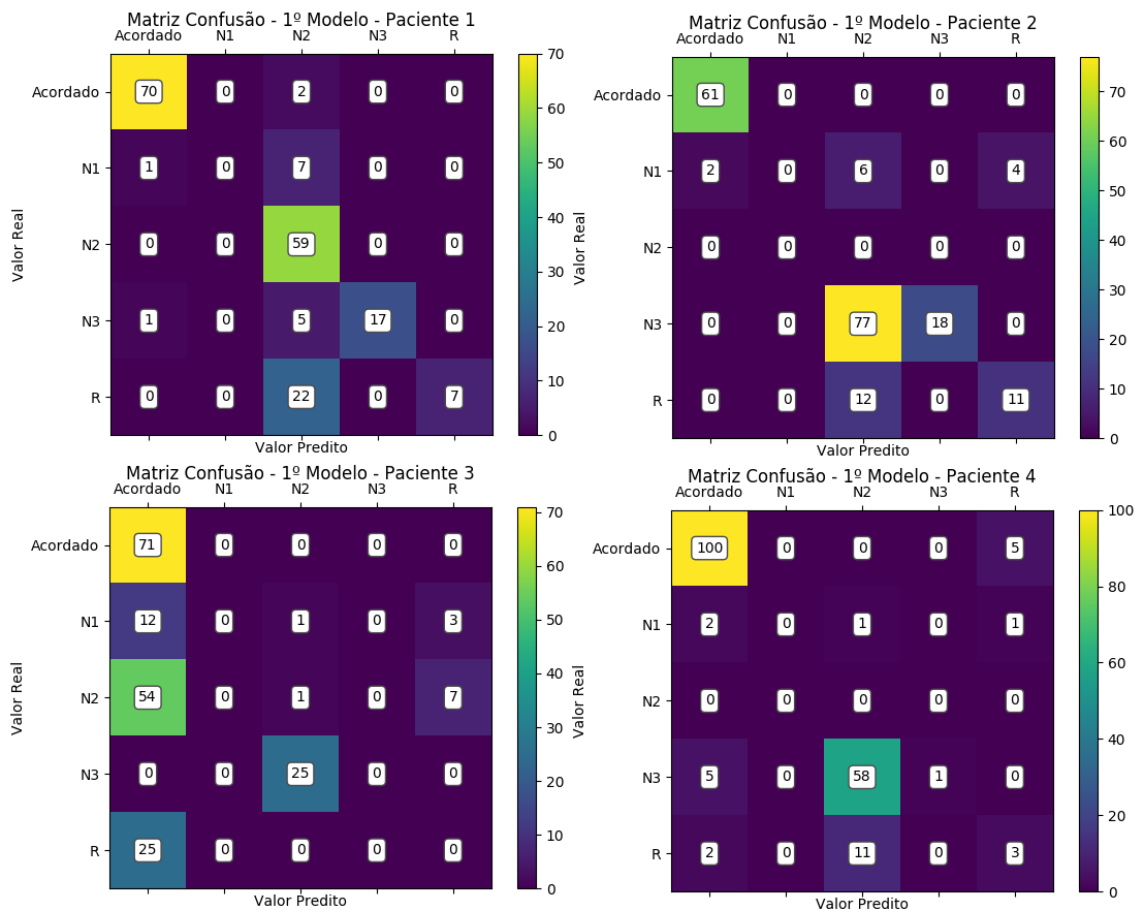


Figura 26 – Matriz Confusão do 1º modelo

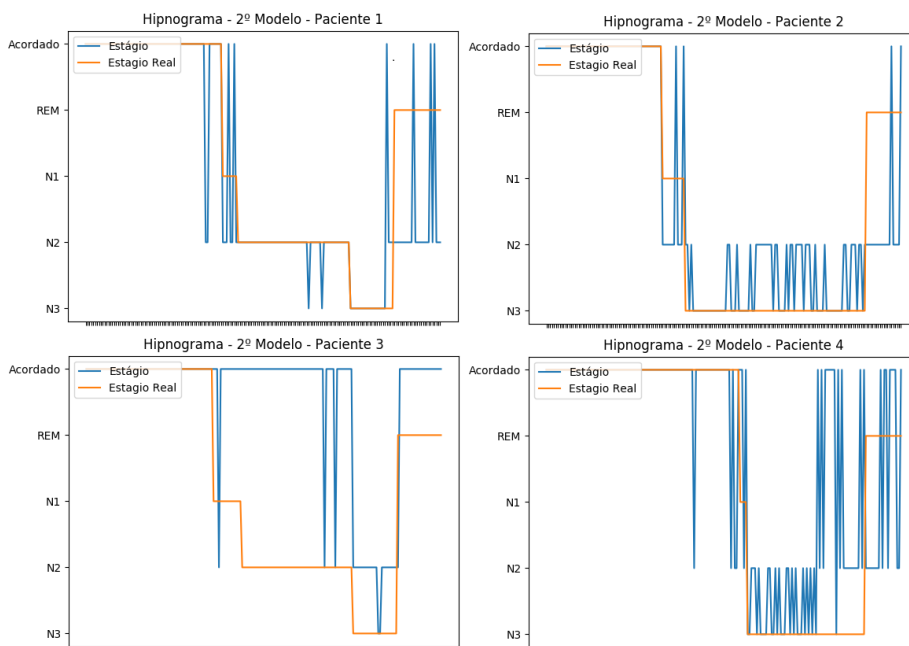


Figura 27 – Hipnogramas do 2º modelo

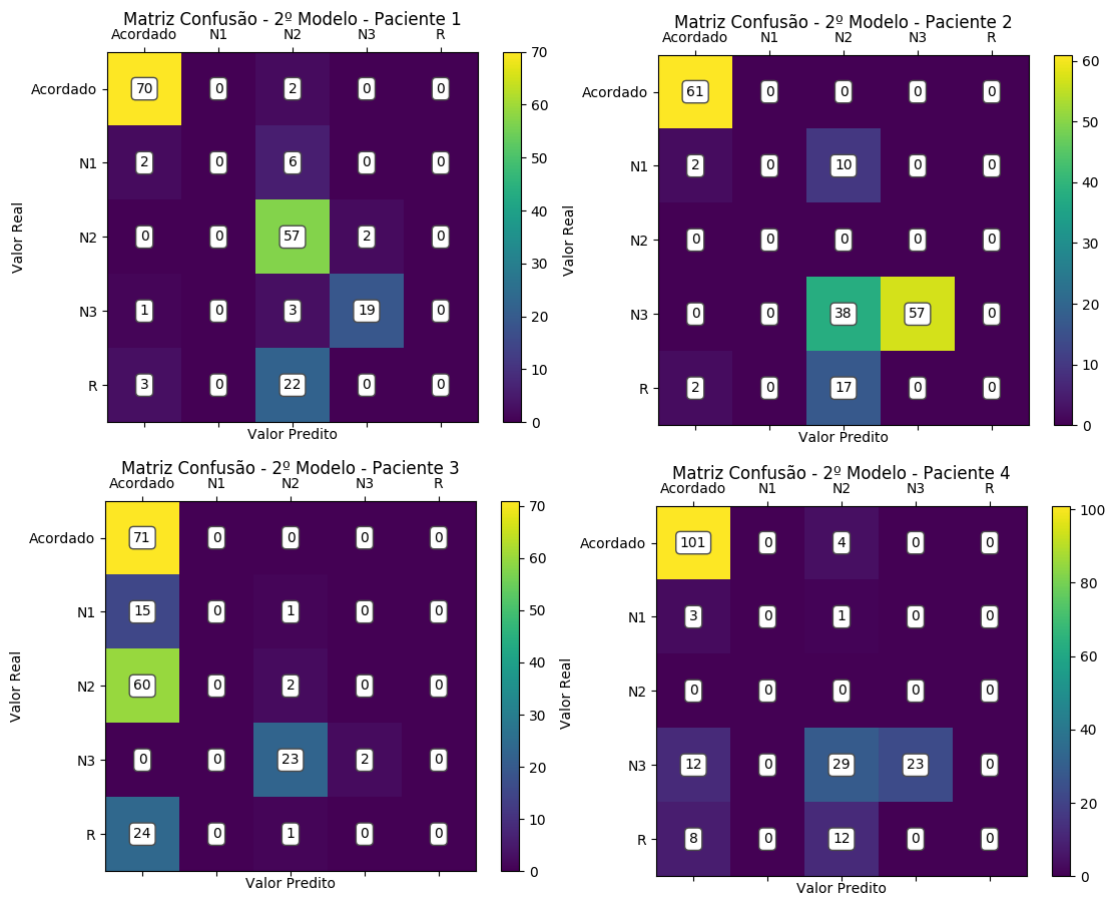


Figura 28 – Matriz Confusão do 2º modelo

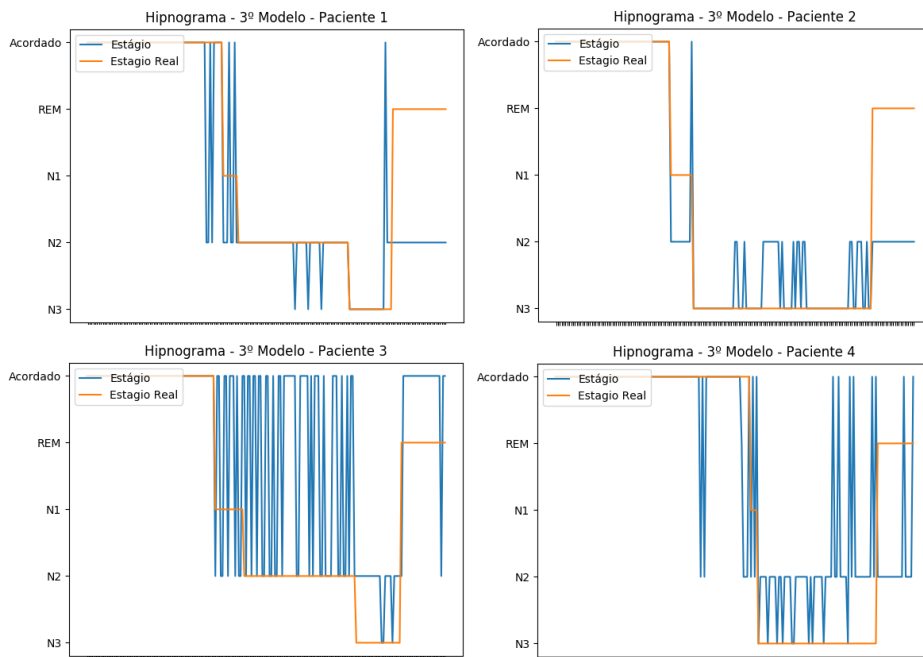


Figura 29 – Hipnogramas do 3º modelo

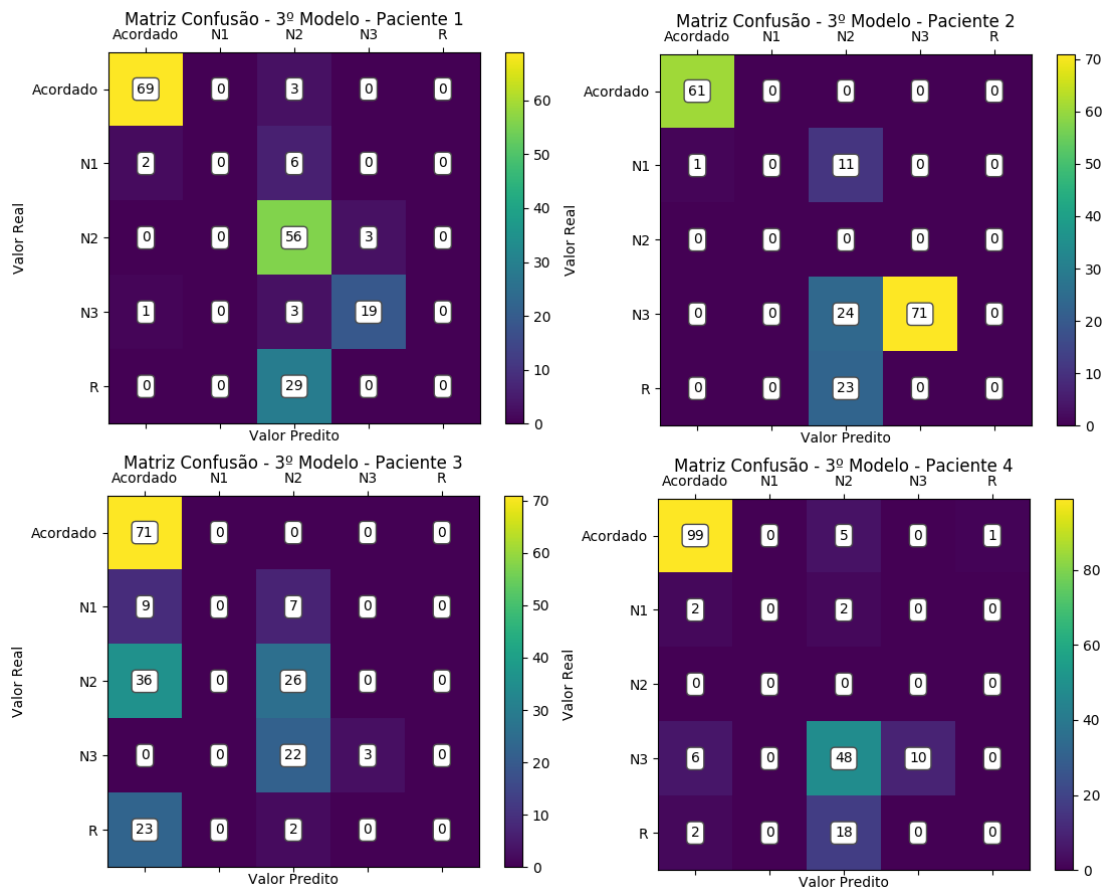


Figura 30 – Matriz Confusão do 3º modelo

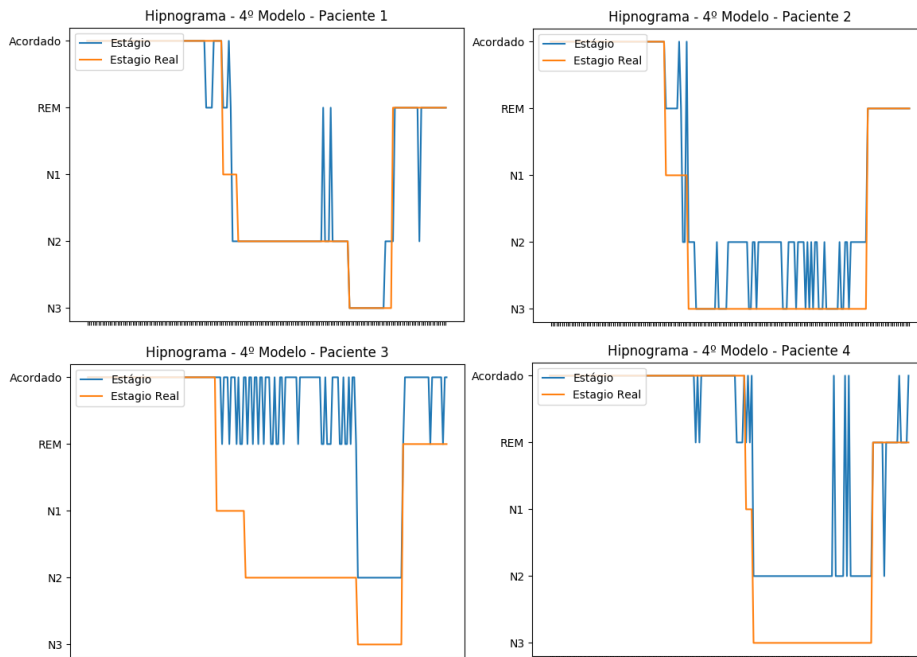


Figura 31 – Hipnogramas do 4º modelo

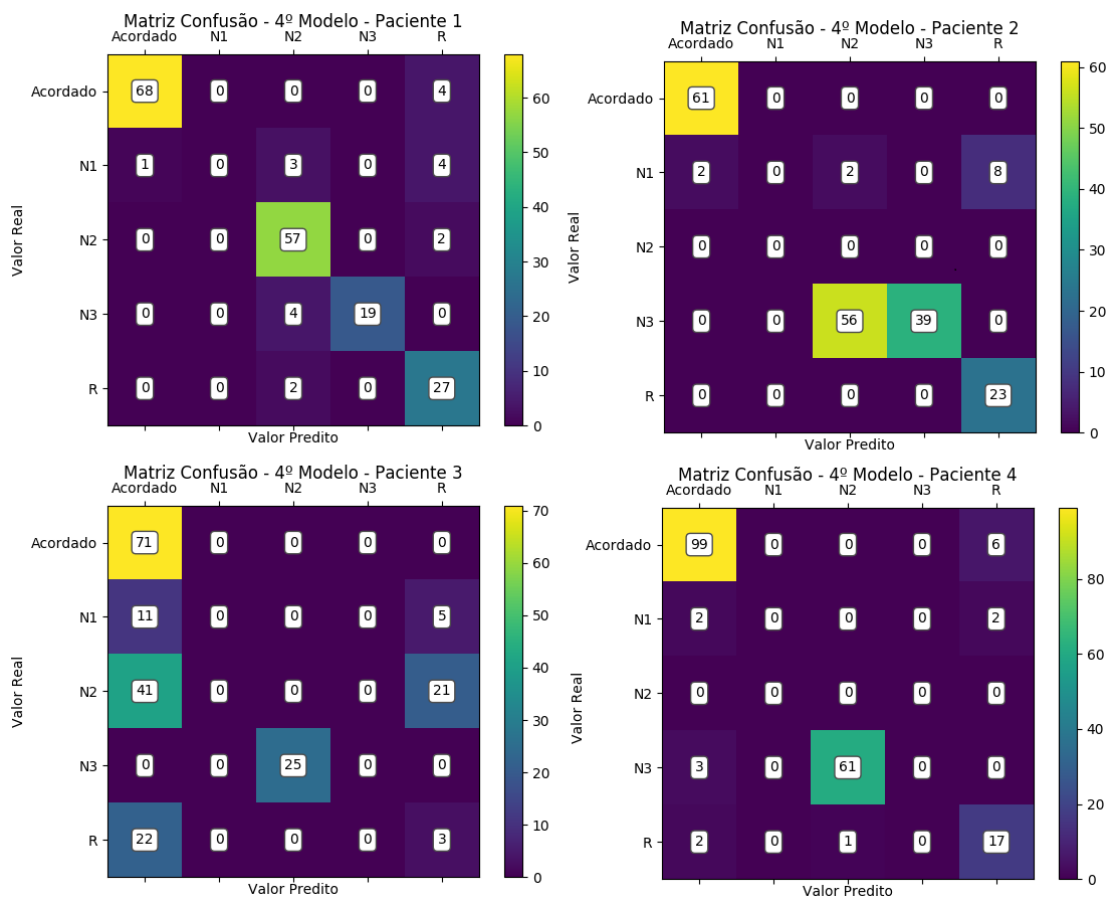


Figura 32 – Matriz Confusão do 4º modelo

Os valores de acurácia de cada modelo possuem valores diferentes devido as funções de ativação ao longo de suas camadas ocultas, porém a diferença mais perceptível entre os resultados é a acurácia entre um paciente e outro.

O paciente com os melhores resultados para todos os modelos é o paciente 1. A provável causa deste ocorrido se deve ao seu *dataset*. A quantidade de amostras para cada estágio do sono no paciente 1 é a maior e mais homogênea dentre os pacientes.

O paciente 3 é o que possui a maior taxa de erros. Após analisar a possível causa para este problema, é possível perceber que o estágio N3 está dentre as saídas nas quais a rede melhor consegue melhor realizar a categorização. Assim, por possuir poucas amostras para treino, e a rede não possuir um número de amostras suficientes para que o aprendizado fique o mais generalizado possível, a acurácia para este paciente se torna baixa.

Apenas um tipo de modelo não conseguiu se sair melhor dentre todos os pacientes. O modelo 2 foi o que se saiu melhor com os dados do paciente 4, o modelo 3 foi o que se saiu melhor com os dados dos pacientes 2 e 3, e o modelo 4 foi o que gerou o melhor resultado com o paciente 1 e o melhor resultado de acurácia da rede neural.

Estar acordado e estar dormindo, por serem estados opostos, apresentam uma maior diferença nos dados captados durante a eletroencefalografia. Assim, observando as matrizes de confusão geradas, é perceptível como a rede consegue, independente do modelo e do paciente, conseguir acertar com alta acurácia o estado de acordado, ficando então apenas com uma maior dificuldade em diferenciar os estágios do sono entre si.

De forma geral, os estados Acordado, N2 e N3 foram os que possuíram melhor categorização pela rede neural. O estado N1 foi o estágio do sono com a menor acurácia, mas isso se deve às baixas quantidades de amostras de N1 utilizadas durante o treino. Apenas no modelo 4 que a categorização do estágio R se saiu bem.

## 5 Conclusão

Com o objetivo de aumentar as formas de diagnosticar doenças, o ser humano com o auxílio da tecnologia realiza pesquisas para desenvolver novos métodos de detecção destas doenças. Um dos métodos já pesquisados é observar as características do comportamento do sono de um indivíduo, comportamento este que pode ser observado por meio de hipnogramas.

Os hipnogramas podem ser utilizados como mais um parâmetro no exame do sono de uma pessoa. Este exame mostrará o tempo no qual a pessoa passará em cada estágio do sono. O período em cada estágio do sono e a latência entre eles pode ser um indicativo de doença.

Este trabalho sugeriu que a classificação dos estágios do sono fossem feitos utilizando somente dados de um canal de eletroencefalografia, o canal Pz-Oz, dados estes que passariam por processamento, com o objetivo de extrair informações sobre seu espectro de frequência e potência média do sinal, que por fim, seria utilizado como parâmetro de entrada de uma rede neural.

Para encontrar uma solução de rede neural que melhor se aplicasse na resolução do problema proposto, foram apresentados 4 modelos de redes neurais. Destes modelos, o que se saiu melhor na classificação dos estágios do sono foi o 4º modelo.

De forma geral, a categorização dos estágios do sono não ocorreu da forma esperada. Apenas o paciente 1 apresentou desempenho satisfatório devido ao seu *dataset* mais homogenizado.

O modelo 4 de rede neural foi o modelo que apesar de não apresentar maior acurácia para todos os pacientes, foi o modelo que melhor conseguiu diferenciar os estágios do sono entre si. Neste caso, o modelo que deve ser utilizado durante a construção de um categorizador de estágios do sono, deve seguir os mesmos moldes ou moldes similares com mais camadas que estejam de acordo com o modelo 4. O bom desempenho deste modelo mostra como os dados normalizados são melhores utilizados pela rede quando sua entrada possui uma função de ativação sigmoideal ( $\tanh$ ), que assim como os valores normalizados, está limitada à 1. Porém o processamento dos valores ao longo das camadas ocultas se dá melhor com o uso de funções sem saturação para os valores positivos ( $\text{elu}$ ).

Ainda é necessário que seja feito treinos com base de dados maiores para que seja possível generalizar de uma melhor forma o aprendizado da rede neural, para que a categorização da rede neural se torne a mais próxima possível da mesma feita por médicos e especialistas. Entretanto, este trabalho valida a possibilidade da criação de um

ferramenta mais robusta que se utilize de uma abordagem similar.

# Referências

- ALLSIGNALPROCESSING. *The Discrete-Time Fourier Transform*. 2015. <<https://allsignalprocessing.com/category/lessons/fourier-series-and-transforms/>>. Citado 2 vezes nas páginas 33 e 35.
- BAUMANN, D. et al. Cosmological non-linearities as an effective fluid. *Journal of Cosmology and Astroparticle Physics*, IOP Publishing, v. 2012, n. 07, p. 051, 2012. Citado na página 38.
- COOLEY, J. W.; LEWIS, P. A.; WELCH, P. D. The fast fourier transform and its applications. *IEEE Transactions on Education*, IEEE, v. 12, n. 1, p. 27–34, 1969. Citado 2 vezes nas páginas 33 e 35.
- DOBORJEH, Z. et al. Spiking neural network modelling approach reveals how mindfulness training rewires the brain. *Scientific reports*, Nature Publishing Group, v. 9, n. 1, p. 6367, 2019. Citado 2 vezes nas páginas 21 e 24.
- DUFFY, F. et al. Clinical electroencephalography and topographic brain mapping. *Journal of Clinical Neurophysiology*, LWW, v. 7, n. 2, p. 296–299, 1990. Citado 5 vezes nas páginas 9, 26, 28, 29 e 30.
- FERNÁNDEZ-LEAL, Á. et al. A knowledge model for the development of a framework for hypnogram construction. *Knowledge-Based Systems*, Elsevier, v. 118, p. 140–151, 2017. Citado 3 vezes nas páginas 9, 22 e 25.
- GOLDBERGER, A. L. et al. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, Am Heart Assoc, v. 101, n. 23, p. e215–e220, 2000. Citado 2 vezes nas páginas 43 e 44.
- HARVARD. *Natural Patterns of Sleep*. 2007. <<http://healthysleep.med.harvard.edu/healthy/science/what/sleep-patterns-rem-nrem>>. Citado 2 vezes nas páginas 21 e 22.
- HASSAN, A. R.; SUBASI, A. A decision support system for automated identification of sleep stages from single-channel eeg signals. *Knowledge-Based Systems*, Elsevier, v. 128, p. 115–124, 2017. Citado na página 29.
- HAYKIN, S. S. et al. *Neural networks and learning machines*. [S.l.]: Pearson education Upper Saddle River, 2009. v. 3. Citado 7 vezes nas páginas 9, 35, 36, 37, 40, 41 e 42.
- Kemp, B. et al. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering*, v. 47, n. 9, p. 1185–1194, Sep. 2000. Citado na página 44.
- LEE-CHIONG, T. *Sleep medicine: Essentials and review*. [S.l.]: Oxford University Press, 2008. Citado 6 vezes nas páginas 11, 20, 21, 22, 23 e 24.
- LUCEY, B. P. et al. Comparison of a single-channel eeg sleep study to polysomnography. *Journal of sleep research*, Wiley Online Library, v. 25, n. 6, p. 625–635, 2016. Citado na página 29.



NUTT, D.; WILSON, S.; PATERSON, L. Sleep disorders as core symptoms of depression. *Dialogues in clinical neuroscience*, Les Laboratoires Servier, v. 10, n. 3, p. 329, 2008. Citado 5 vezes nas páginas 9, 18, 25, 26 e 27.

OPPENHEIM, A. V.; WILLSKY, A. S.; NAWAB, S. H. *Signals & Systems (2Nd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996. ISBN 0-13-814757-4. Citado 5 vezes nas páginas 29, 30, 31, 32 e 33.

STEIGER, A.; KIMURA, M. Wake and sleep eeg provide biomarkers in depression. *Journal of psychiatric research*, Elsevier, v. 44, n. 4, p. 242–252, 2010. Citado 3 vezes nas páginas 18, 25 e 26.



# Apêndices

## APÊNDICE A – Link GitHub



Figura 33 – QRcode Github

Link para acesso dos códigos: <https://github.com/thassio23/TCC>

## APÊNDICE B – Código Rede Neural

```
import tensorflow as tf
import matplotlib.pyplot as plt
from numpy import loadtxt
import numpy as np
import datetime
from sklearn.metrics import confusion_matrix
from keras.callbacks import EarlyStopping
from keras.callbacks import ModelCheckpoint

dataset_treino = loadtxt("megaset.txt", delimiter='')
dataset_teste = loadtxt("setTeste1.txt", delimiter='')
dataset_val = loadtxt("setVal.txt", delimiter='')

Entradas_Treino = dataset_treino[:,0:5]
Saidas_Treino = dataset_treino[:,5:11]

Entradas_Testes = dataset_teste[:,0:5]
Saidas_Testes = dataset_teste[:,5:11]

Entradas_Val = dataset_val[:,0:5]
Saidas_Val = dataset_val[:,5:11]

Entradas_Treino = tf.keras.utils.normalize(Entradas_Treino, axis=1)
Entradas_Testes = tf.keras.utils.normalize(Entradas_Testes, axis=1)
Entradas_Val = tf.keras.utils.normalize(Entradas_Val, axis=1)

with tf.device('/gpu:0'):
    model = tf.keras.models.Sequential()
    model.add(tf.keras.layers.Dense(5, input_dim=5))

    model.add(tf.keras.layers.Dense(64))
    model.add(tf.keras.layers.BatchNormalization())
```

```
model.add(tf.keras.layers.Activation('tanh'))

model.add(tf.keras.layers.Dense(64))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('elu'))

model.add(tf.keras.layers.Dense(64))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('tanh'))
model.add(tf.keras.layers.Dropout(0.5))

model.add(tf.keras.layers.Dense(5))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('softmax'))
print(model.summary())
model.compile(loss='categorical_crossentropy',
              optimizer='adam', metrics=['accuracy'])

mcp_save = ModelCheckpoint("modelo1.h5",
                           save_best_only=True, monitor='val_acc', mode='max')

history = model.fit(Entradas_Treino, Saidas_Treino,
                   epochs=10, validation_data=(Entradas_Val, Saidas_Val),
                   batch_size=8, callbacks=[mcp_save])

plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Acurácia - 4º Modelo')
plt.ylabel('Acurácia')
plt.xlabel('Época')
plt.legend(['Acurácia - Treino', 'Acurácia - Teste'], loc='upper left')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
```

```
plt.title('Perda - 4º Modelo')
plt.ylabel('Perda')
plt.xlabel('Época')
plt.legend(['Perda - Treino', 'Perda - Teste'], loc='upper left')
plt.show()
```

```
labels = ['Acordado', 'N1', 'N2', 'N3', 'R']
```

```
acerto = 0
```

```
diferente = 0
```

```
prediction = model.predict(Entradas_Testes)
```

```
for i in range(191):
```

```
if np.argmax(prediction[i]) == np.argmax(Saidas_Testes[i]):
```

```
acerto = acerto + 1
```

```
else:
```

```
diferente = diferente + 1
```

```
print('acerto:')
```

```
print(acerto)
```

```
print('\n')
```

```
print('erro:')
```

```
print(diferente)
```

```
acc2 = acerto/(acerto+diferente)
```

```
print('\n')
```

```
print('acuracia real:')
```

```
print(acc2)
```

```
a = []
```

```
b = []
```

```
for i in range(191):
```

```
a.append(np.argmax(Saidas_Testes[i]))
```

```
b.append(np.argmax(prediction[i]))
```

```
cm = confusion_matrix(a, b)

# Show confusion matrix in a separate window

fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(cm, interpolation='nearest')
fig.colorbar(cax)
plt.title('Matriz Confusão - 4º Modelo')
plt.ylabel('Valor Real')
plt.xlabel('Valor Predito')
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
for (i, j), z in np.ndenumerate(cm):
    ax.text(j, i, '{:d}'.format(z), ha='center', va='center',
            bbox=dict(boxstyle='round', facecolor='white', edgecolor='0.3'))
plt.show()
```



## APÊNDICE C – Código Montagem DataSet

```

import matplotlib.pyplot as plt
import numpy as np
from scipy import fftpack

NumeroDeLinhas = 4548000
ColunaSelecionada = 1 #0 Hora / 1 - Canal

f = open("Diretório/DataSet.txt","r+")
f1 = open("Diretório/DataSet_Classificação.txt","r+")

linha = []
linha1 = []

AmostrasDe30Segundos = []

for x in range (0, NumeroDeLinhas):
linha.append([])

for x in range(0, NumeroDeLinhas):
linha[x] = f.readline().split(",")

for x in range (0, 1516):
linha1.append([])

for x in range(0, 1516):
linha1[x] = f1.readline().split(",") linha

for x in range (0, 1516):
AmostrasDe30Segundos.append([])

fim = 0

for x in range (0, 1516):
for y in range (0, 3000):
AmostrasDe30Segundos[x].append(float(linha[y+fim][ColunaSelecionada]))

```

```
#Faz uma matriz 1516 linhas para amostras diferentes
fim = fim + 3000

#Calculo da Transformada de Fourier e dataset

FrequenciaDeAmostragem = 100
Dataset = []

for x in range (0, 1516):
Dataset.append([])

for x in range (0, 1516):

SinalAnalisado = AmostrasDe30Segundos[x]
QuadradooSinal = np.square(SinalAnalisado)
PotenciaMediadoSinal = (np.sum(QuadradooSinal)/1516)
X = fftpack.fft(SinalAnalisado)
freqs = fftpack.fftfreq(len(SinalAnalisado)) * FrequenciaDeAmostragem
Dataset[x].append(PotenciaMediadoSinal)
Dataset[x].append(np.abs(X) [1501:1621].sum())
Dataset[x].append(np.abs(X) [1621:1741].sum())
Dataset[x].append(np.abs(X) [1741:1861].sum())
Dataset[x].append(np.abs(X) [1861:2341].sum())

file = open("Diretório/DataSet.txt","a+")

for x in range (0, 1516):
file.write(str(round(Dataset[x] [0], 1)))
file.write(" ")
file.write(str(round(Dataset[x] [1], 1)))
file.write(" ")
file.write(str(round(Dataset[x] [2], 1)))
file.write(" ")
file.write(str(round(Dataset[x] [3], 1)))
file.write(" ")
file.write(str(round(Dataset[x] [4], 1)))
file.write(" ")
if (str(linha1[x] [1]) == "0.000000\n"):
file.write(str("1 0 0 0 0\n"))
```

```
if (str(linha1[x][1]) == "1.000000\n"):  
file.write(str("0 1 0 0 0\n"))
```

```
if (str(linha1[x][1]) == "2.000000\n"):  
file.write(str("0 0 1 0 0\n"))
```

```
if (str(linha1[x][1]) == "3.000000\n"):  
file.write(str("0 0 0 1 0\n"))
```

```
if (str(linha1[x][1]) == "4.000000\n"):  
file.write(str("0 0 0 1 0\n"))
```

```
if (str(linha1[x][1]) == "5.000000\n"):  
file.write(str("0 0 0 0 1\n"))
```

# APÊNDICE D – Código Montagem Hipnograma e Matriz Confusão

```
import tensorflow as tf
import matplotlib.pyplot as plt
from numpy import loadtxt
import numpy as np
import datetime
from sklearn.metrics import confusion_matrix

dataset_teste = loadtxt("setTeste4.txt", delimiter=' ')

model = tf.keras.models.load_model("modelo4.h5")

Entradas_Testes = dataset_teste[:,0:5]
Saidas_Testes = dataset_teste[:,5:11]

Entradas_Testes = tf.keras.utils.normalize(Entradas_Testes, axis=1)

labels = ['Acordado', 'N1', 'N2', 'N3', 'R']

acerto = 0
diferente = 0

prediction = model.predict(Entradas_Testes)

for i in range(193):

    if np.argmax(prediction[i]) == np.argmax(Saidas_Testes[i]):
        acerto = acerto + 1
    else:
        diferente = diferente + 1

print('acerto:')
```

```
print(acerto)
print('\n')
print('erro:')
print(diferente)

acc2 = acerto/(acerto+diferente)
print('\n')
print('acuracia real:')
print(acc2)

a = []
b = []
for i in range(193):
    a.append(np.argmax(Saidas_Testes[i]))

    b.append(np.argmax(prediction[i]))

cm = confusion_matrix(a, b)

# Show confusion matrix in a separate window

fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(cm, interpolation='nearest')
fig.colorbar(cax)
plt.title('Matriz Confusão - 4º Modelo - Paciente 4')
plt.ylabel('Valor Real')
plt.xlabel('Valor Predito')
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
for (i, j), z in np.ndenumerate(cm):
    ax.text(j, i, '{:d}'.format(z), ha='center', va='center',
            bbox=dict(boxstyle='round', facecolor='white', edgecolor='0.3'))
plt.show()

conta = 0
hora = 1
```

```
estagio = [1]*193
estagiori = [1]*193
compara = [1]*193
labelx = [' ']*193

for i in range(193):

    if np.argmax(prediction[i]) == 0:
        estagio[i] = 4
    if np.argmax(prediction[i]) == 1:
        estagio[i] = 2
    if np.argmax(prediction[i]) == 2:
        estagio[i] = 1
    if np.argmax(prediction[i]) == 3:
        estagio[i] = 0
    if np.argmax(prediction[i]) == 4:
        estagio[i] = 3

    if np.argmax(Saidas_Testes[i]) == 0:
        estagiori[i] = 4
    if np.argmax(Saidas_Testes[i]) == 1:
        estagiori[i] = 2
    if np.argmax(Saidas_Testes[i]) == 2:
        estagiori[i] = 1
    if np.argmax(Saidas_Testes[i]) == 3:
        estagiori[i] = 0
    if np.argmax(Saidas_Testes[i]) == 4:
        estagiori[i] = 3

    if conta == 122:
        conta = 0
        labelx[i] = str(hora) + ":00"
        hora = hora + 1

    conta = conta + 1
    compara[i] = i
    labels = ('N3', 'N2', 'N1', 'REM', 'Acordado')
```

```
plt.yticks([0,1,2,3,4], labels)
```

```
plt.xticks(compara, labelx)
```

```
plt.plot(estagio)
```

```
plt.plot(estagiori)
```

```
plt.title('Hipnograma - 4º Modelo - Paciente 4')
```

```
plt.ylabel('Estágio')
```

```
plt.xlabel('Tempo')
```

```
plt.legend(['Estágio', 'Estagio Real'], loc='upper left')
```

```
plt.show()
```