



Universidade Federal de Ouro Preto - UFOP
Escola de Minas
Colegiado do curso de Engenharia de
Controle e Automação - CECAU



Vinícius Augusto Pereira Queiroz

**Aplicação de Redes Neurais Artificiais Profundas em
Reconhecimento de Padrões Motores de Eletroencefalograma
Utilizando Redes Complexas**

Monografia de Graduação em Engenharia de Controle e Automação

Ouro Preto, 2017

Vinícius Augusto Pereira Queiroz

**Aplicação de Redes Neurais Artificiais Profundas em
Reconhecimento de Padrões Motores de Eletroencefalograma
Utilizando Redes Complexas**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Eduardo José da Silva Luz

Ouro Preto, 2017

cutte

Queiroz, Vinicius Augusto Pereira.

Aplicação de Redes Neurais Artificiais Profundas em Reconhecimento de Padrões Motores de Eletroencefalograma Utilizando Redes Complexas [manuscrito] / Vinicius Augusto Pereira Queiroz. - 2017.

56f.: il.: color; grafs; tabs; Equações; Links.

Orientador: Prof. MSc. Eduardo José da Silva Luz.

Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais.

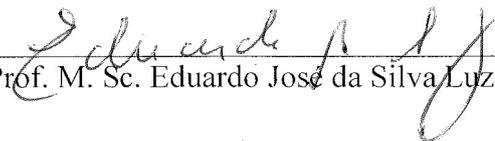
Área de concentração: Engenharia de Controle e Automação.

1. BCI. 2. EEG. 3. Imageria Motora. 4. Aprendizagem de Máquina. 5. Redes Complexas. I. Luz, Eduardo José da Silva. II. Universidade Federal de Ouro Preto. III. Título.

CDU:

Catálogo: ficha@sisbin.ufop.br

Monografia defendida e aprovada, em 11 de dezembro de 2017, pela comissão avaliadora constituída pelos professores:


Prof. M. Sc. Eduardo José da Silva Luz - Orientador


Prof. Dr. Gladston Juliano Prates Moreira – Professor Convidado


Prof. M. Sc. Reinaldo Silva Fortes – Professor Convidado

Dedico este trabalho à minha família, amigos e professores, que nunca deixaram de acreditar em mim.

Resumo

Os projetos de Interface Cérebro-Máquina têm se destacado na comunidade da Engenharia Biomédica na última década, devido aos avanços dos recursos computacionais e dos algoritmos de Aprendizagem de Máquina e também às diversas aplicações que este tipo de sistema pode proporcionar aos usuários. Entretanto, as pesquisas apresentadas na literatura ainda apresentam barreiras impedindo a utilização destes sistemas pelo público em geral, sejam por limites de praticidade ou de eficácia. Neste trabalho, propomos um estudo sobre as Interfaces Cérebro-Máquina, apresentando os conceitos básicos para sua implementação e para o entendimento do funcionamento dos mesmos. Um projeto prático também foi realizado e avaliado utilizando um banco de dados de eletroencefalograma de Imageria Motora, testando um método proposto recentemente que ainda não havia sido utilizado para este propósito, com o intuito de expandir as pesquisas nesta área. Os resultados mostram que ainda há muitos testes a serem realizados com este método, e que a definição dos valores dos hiper-parâmetros do algoritmo é a maior dificuldade de implementação deste modelo de Interface Cérebro-Máquina.

Palavras-chaves: BCI, EEG, imageria motora, aprendizagem de máquina, redes complexas.

Abstract

Brain-Computer Interfaces projects have been highlighted in the Biomedical Engineering community over the last decade, due to improvements in computational resources and Machine Learning algorithms and also due to the diverse applications these systems may provide to the users. However, the researches presented in the literature still present barriers preventing general public use of these systems, either by limits in practicality or efficacy. In this paper we propose a study about Brain Computer Interfaces, presenting the basic concepts needed for its implementation and understanding how they work. A practical project was also elaborated and evaluated using a Motor Imagery electroencephalogram database, testing a recently proposed method which had not been utilized for this purpose yet, with the aim to expand the researches in this field. Results show that there is plenty of tests yet to be made with this method, and that the definition of the hyperparameters values of the algorithm is the most difficult task of implementation of this model of Brain-Computer Interface.

Key-words: BCI, EEG, motor imagery, machine learning, complex networks.

Lista de ilustrações

Figura 1	Equipamento de Imageria por Ressonância Magnética (MRI)	12
Figura 2	Representação de Eletrocorticografia (ECoG)	13
Figura 3	Equipamento de Eletroencefalografia (EEG)	13
Figura 4	Soletrador baseado em P300-EVP	15
Figura 5	Representação de um estimulador computadorizado de SSVEP	15
Figura 6	Gráfico representativo de um EEG gravado durante uma seção de SSVEP	16
Figura 7	Transporte de íons de potássio e cloreto através da membrana celular, pelos Canais Proteicos	20
Figura 8	A: Difusão de íons de potássio para fora da célula, através de membrana que só é permeável a íons de potássio. B: Difusão de íons de sódio para dentro da célula, através de membrana que só é permeável a íons de sódio.	21
Figura 9	Bomba de sódio-potássio. ATP, Adenosina Trifosfato; ADP, Adenosina Difosfato; Na^+ , íon de sódio, K^+ , íon de potássio	22
Figura 10	Ativação dos canais regulados por voltagem. Canal de sódio acima e canal de potássio abaixo.	23
Figura 11	Propagação do potencial de ação ao longo de uma fibra condutora. . .	24
Figura 12	Ilustração de um neurônio típico do sistema nervoso central	25
Figura 13	Processo de sinapse química	26
Figura 14	Estados do neurônio. A: Neurônio em repouso; B: Neurônio excitado; C: Neurônio inibido.	27
Figura 15	Somação espacial de um neurônio. Os números representam o potencial de membrana naquele ponto do neurônio, e os terminais com letras representam as terminações nervosas do axônio pré-sináptico. E: Excitação; I: Inibição	28
Figura 16	Áreas funcionais do cérebro	29
Figura 17	Áreas motoras e somáticas do córtex cerebral	30
Figura 18	Sistema 10-20 de posicionamento de eletrodos com 64 canais.	31
Figura 19	Exemplo de uma Rede Neural Artificial, com $L = 4$, $K = 4$, $s_1 = 3$, $s_2 = 5$, etc.	35
Figura 20	A: Rede Neural Artificial totalmente conectada. B: Rede Neural Artificial após a aplicação da técnica de <i>Dropout</i>	35
Figura 21	Representação prática de comparação de <i>throughput</i> entre Programação Serial (carro) e GPGPU (ônibus).	36

Figura 22	Comparação entre Programação Paralela e Programação Convencional aplicada ao reconhecimento de	37
Figura 23	Posicionamento dos eletrodos no sistema de aquisição de dados do EEGMMIDB.	40
Figura 24	Interface Gráfica do Usuário do treinamento do classificador.	45
Figura 25	GUI em funcionamento.	46
Figura 26	Resultados apresentados na GUI.	47

Lista de tabelas

Tabela 1	Ondas cerebrais e respectivas faixas de frequência.	31
Tabela 2	Matriz de Confusão do Teste 8	49
Tabela 3	Resultados dos testes com o algoritmo implementado, variando-se os parâmetros em destaque	50

Sumário

1	Introdução	11
1.1	Justificativas e Relevância	11
1.2	Aquisição de Sinais Cerebrais	12
1.3	Modelos de Interfaces Cérebro-Máquina	14
1.4	Metodologia	16
1.5	Objetivos	17
1.6	Organização e estrutura	17
2	Revisão Teórica	19
2.1	Neurofisiologia	19
2.1.1	Potencial de Membrana	19
2.1.2	Potencial de Ação	22
2.1.3	Sinapses	24
2.1.4	Funções Motoras	28
2.2	Eletroencefalograma	29
2.2.1	Eletroencefalograma de Imageria Motora	32
2.3	Aprendizagem de Máquina	32
2.3.1	Redes Neurais Artificiais Profundas	34
2.4	Programação Paralela	36
3	Desenvolvimento	38
3.1	Banco de Dados	38
3.1.1	Descrição do EEG Motor Movement/Imagery Dataset	38
3.1.2	Pré-Processamento do Banco de Dados: Sliding Window	40
3.2	Extração de Características: Redes Complexas aplicadas à Vectorencefalograma Temporal	41
3.3	Interface Gráfica do Usuário	43
3.4	Treinamento do Classificador	44
4	Análise do Algoritmo	48
4.1	Métricas de Interesse	48
4.2	Resultados	49
5	Conclusão	51
	Referências	52

1 Introdução

1.1 Justificativas e Relevância

A Interface Cérebro-Máquina (do inglês, Brain-Machine Interface, ou BMI), também conhecida como Interface Cérebro-Computador (do inglês, Brain-Computer Interface, ou BCI) é um dos tópicos de pesquisa mais valorizados hoje em dia na área da Engenharia Biomédica. Ela consiste em fazer uma aquisição de sinais do cérebro, e, a partir destes, enviar comandos à algum tipo de máquina, dispositivo ou computador, utilizando métodos de processamento de sinais ou imagens, extração de características automatizadas e aprendizagem de máquina (também conhecido como reconhecimento de padrões).

Os maiores beneficiados por esse tipo de tecnologia são pacientes que têm sua coordenação motora prejudicada, como os pacientes vítimas de derrame, paralisia infantil, Esclerose Lateral Amiotrófica (do inglês, Amyotrophic Lateral Sclerosis, ou ALS), paraplégicos ou tetraplégicos. Dentre as aplicações práticas existentes atualmente, destacam-se:

- Sistemas de comunicação em pacientes com Síndrome de Encarceramento; (SCHE-RER et al., 2004; ULLAH et al., 2011)
- Sistemas de posicionamento 2D e clique de *mouse*; (BLACK et al., 2003)
- Sistemas de navegação de cadeiras-de-rodas; (CARLSON; Del R. Millan, 2013)
- Sistemas de controle de próteses robóticas; (MÜLLER-PUTZ; PFURTSCHELLER, 2008)
- Controle de membros periféricos em pacientes paraplégicos ou tetraplégicos, por Estímulo Elétrico Funcional; (PFURTSCHELLER et al., 2005)
- Sistemas de fisioterapia assistida por Realidade Virtual; (ZHANG et al., 2015)
- Sistemas de verificação de consciência de pacientes comatosos; (PAN et al., 2014)
- Sistemas de verificação do nível de concentração em aplicações críticas; (KHAN; HONG, 2015)
- Reconhecimento de emoções para sistemas de avaliação e recomendação; (LIN et al., 2010)
- Controle de dispositivos com as mãos livres; (LAFLEUR et al., 2013)

- Imersão em Jogos de Realidade Virtual; (LÉCUYER et al., 2008)

A partir desta análise, é possível verificar que não somente pacientes com alguma deficiência motora, mas também pessoas saudáveis podem se beneficiar com o uso de BCI's.

1.2 Aquisição de Sinais Cerebrais

Os sinais a serem adquiridos podem ser sinais elétricos provenientes do couro cabeludo (eletroencefalograma, ou EEG), sinais elétricos obtidos diretamente do córtex cerebral (eletrocorticograma, ou ECoG), ou até mesmo Imagens de Ressonância Magnética (do inglês, Magnetic Resonance Imaging, ou MRI), entre outros.

Utilizar um MRI costuma prover um sistema com acurácia maior do que os sistemas utilizando os outros dois tipos de aquisição de sinais, além não necessitar de cirurgias para sua utilização. Porém, o equipamento é muito grande, excessivamente caro, e o paciente precisa estar deitado e imóvel, o que impossibilita a sua utilização em situações práticas do dia-a-dia. A Figura 1 evidencia a importabilidade de equipamentos de MRI.



Figura 1: Equipamento de Imageria por Ressonância Magnética (MRI)

Sistemas que utilizam ECoG's têm precisão um pouco abaixo do MRI, mas são mais baratos, relativamente portáteis, e não necessitam que o paciente esteja totalmente imóvel e deitado, abrindo novas possibilidades de aplicações práticas. Entretanto, os sensores são invasivos (i.e., é necessário realizar uma operação cirúrgica intracraniana no paciente, para introduzir os eletrodos nas posições corretas, que ficarão em contato diretamente com o córtex cerebral). Na Figura 2, vemos uma representação gráfica do posicionamento de uma grade de eletrodos no córtex sensorimotor do paciente.

Por outro lado, sistemas baseados em EEG, apesar de terem ainda menos precisão e serem mais suscetíveis a ruídos elétricos, magnéticos e mecânicos do que MRI's ou ECoG's, são incomparavelmente mais baratos, relativamente portáteis e são não-invasivos - já que

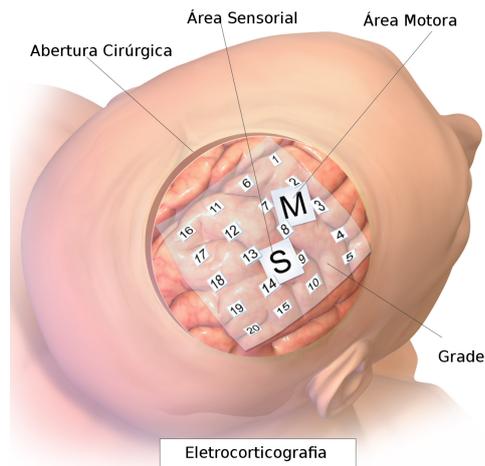


Figura 2: Representação de Eletrocorticografia (ECoG)

os eletrodos são posicionados na superfície do couro cabeludo do paciente. Portanto, sua utilização no cotidiano e comercialização para o público é viável. Desta forma, muitos sistemas de BCI existentes atualmente utilizam sinais de EEG como forma de comunicação entre o humano e a máquina. Na Figura 3 podemos ver um paciente com eletrodos posicionados no couro cabeludo.



Figura 3: Equipamento de Eletroencefalografia (EEG)

É importante notar que existem outros tipos de sistemas de aquisição de dados tais como Microeletrocorticografia (μ ECoG), Magnetoencefalografia (MEG) e Microeletrodos Intracorticais, mas não serão discutidos neste trabalho. Mais informações sobre estas e outras formas de aquisição de sinais cerebrais podem ser encontradas em Nakasatp et al. (1994) e Kellis et al. (2016). Neste trabalho, utilizaremos informações provenientes de eletroencefalogramas (EEG), e o tema será discutido com maior detalhe nas Seções 2.2 e 3.1.

1.3 Modelos de Interfaces Cérebro-Máquina

Diferentes aplicações de BCI requerem diferentes formas de abordagem ao processamento de sinais do cérebro. Por exemplo: Se o objetivo do sistema é o controle de próteses robóticas ou do braço em pacientes paraplégicos, o ideal é que o sistema saiba reconhecer a intenção de se movimentar o braço, e faça com que este se movimente. Por outro lado, se o objetivo do sistema é saber se um motorista de automóvel está com sono enquanto dirige, o sistema deve ser capaz de continuamente verificar o estado de consciência do motorista, e efetuar alguma ação caso este esteja com sono. Estes sistemas têm suas peculiaridades, em relação à faixa de frequência relevante dos sinais, posicionamento dos eletrodos, se devem ser síncronos ou assíncronos, se o paciente estará concentrado em outra atividade, etc. Desta forma, abordagens diferentes de BCI têm se mostrado eficazes em diferentes aplicações.

Uma das abordagens de BCI é conhecido como P300. Este tipo de BCI é fundamentado nos Potenciais Evocados Relacionados (ERP), à partir do Efeito *Oddball*, o qual evidencia que dado um certo estímulo visual, auditivo ou tátil, que o paciente esteja esperando que aconteça, e não aconteça em intervalos regulares, poderá ser observado um pico no sinal do EEG 300ms após o estímulo (GORDON; LEDOUX, 2008). Desta forma, é possível construir sistemas em que o paciente se concentre no resultado que quer, e quando este resultado for apresentado, o pico no sinal dos eletrodos localizados na região do córtex sensorial-somático será computado, fazendo com que o dispositivo tome aquele resultado como escolhido.

Uma aplicação do modelo de BCI P300 é o sistema de comunicação utilizando P300-Potenciais Evocados Visuais (EVP) para pacientes com Esclerose Lateral Amiotrófica (ALS), tal como feito por Kaper et al. (2004). O paciente se concentra na letra que ele quer escolher para soletrar a palavra que quer comunicar. Enquanto isso, o sistema pisca as letras de forma aleatória, uma de cada vez. Quando a letra que o paciente está querendo soletrar é destacada, um pico no sinal de EEG é observado após 300ms. Quando as outras letras são destacadas, não ocorre este pico. Desta forma, o algoritmo consegue saber qual letra o paciente está querendo soletrar, armazena esta letra, e repete o processo para decodificar as próximas letras, até finalizar uma palavra ou sentença. O painel de letras piscando pode ser visto na Figura 4, onde a letra M está destacada no instante.

Outro modelo de BCI é conhecido como Potenciais Evocados Visuais de Estado Estacionário (do inglês, Steady-State Visual Evoked Potentials, ou SSVEP). Seu fundamento parte do princípio que, ao se ter um estímulo visual que pisca em frequências maiores que 6Hz, o córtex occipital emitirá sinais elétricos senoidais com a mesma frequência que o estímulo (AMIRI et al., 2013). A fonte dos EVP's podem ser provenientes de uma tela de computador ou de um painel de LED's.



Figura 4: Soletrador baseado em P300-EVP

Na Figura 5, vemos uma representação de um estimulador de SSVEP, que pode ser utilizado para o controle de uma cadeira-de-rodas, por exemplo. Os blocos vermelhos piscam em frequências constantes e diferentes entre si. O usuário se foca no bloco correspondente à função que quer realizar, e o córtex occipital emitirá EVP's com a mesma frequência que o bloco que está sendo focado. Os eletrodos captam os potenciais elétricos do cérebro e o enviam a um computador, que irá realizar o processamento de sinais e extração de características para destacar a frequência predominante no EEG, como representado pela Figura 6. O resultado do processo anterior é alimentado a um classificador, que definirá qual a ação pretendida pelo usuário, e, finalmente, tomará o curso de ação correspondente.



Figura 5: Representação de um estimulador computadorizado de SSVEP

BCI's baseadas em SSVEP têm se mostrado muito eficazes, em relação à acurácia do classificador, tempo de treinamento e taxa de transmissão de informação, principalmente quando se fazem sistemas híbridos de SSVEP e P300 (AMIRI et al., 2013). Entretanto, sistemas baseados em EVP causam fadiga rapidamente, já que o usuário precisa estar focado em luzes piscando à uma frequência baixa. Além disso, o paciente não pode ter os

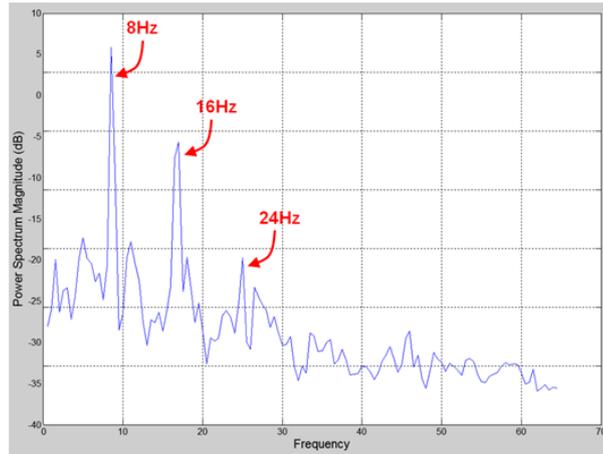


Figura 6: Gráfico representativo de um EEG gravado durante uma sessão de SSVEP

músculos oculares nem sua visão debilitados, e ainda há a necessidade do paciente ficar concentrado somente nesta tarefa. Isto faz com que suas aplicações sejam limitadas.

Para contornar as limitações dos sistemas baseados em P300 e SSVEP, existem também os sistemas baseados em Imageria Motora (do inglês, *Motor Imagery*, ou MI). Este método consiste em decodificar sinais cerebrais provenientes de uma intenção de se movimentar músculos específicos do corpo. Este método será o foco deste trabalho, e será discutido com maior detalhe na Seção 2.2.1.

1.4 Metodologia

O trabalho foi dividido em 3 etapas principais: teórica, de implementação, e de análise:

- Na etapa teórica, um estudo de neurofisiologia foi realizado, utilizando Guyton e Hall (2013) como referência principal. Em seguida, uma revisão bibliográfica sobre BCI foi realizada, analisando diferentes formas de implementação de uma BCI, e seus conceitos pertinentes.
- Na etapa de implementação, primeiramente foi escolhido um banco de dados para os testes. Em seguida, foi modificado um programa para a extração de características baseadas em Vectorencefalograma Temporal (TVEG), e implementado um algoritmo de aprendizagem de máquina baseado em Redes Neurais Artificiais Profundas, utilizando Programação Paralela para agilizar o processo de aprendizagem, com uma Interface Gráfica de Usuário.
- A parte de análise consistiu-se em realizar alguns testes iterativos dos algoritmos implementados, alterando-se os hiper-parâmetros, e anotando-se as métricas de interesse.

1.5 Objetivos

Este trabalho será focado no estudo tanto teórico quanto prático da implementação de Interfaces Cérebro-Máquina (BCI). Uma abrangente revisão teórico-bibliográfica será introduzida nas seções posteriores, e a análise de um método preliminar baseado em Vectorencefalograma Temporal (TVEG) também será apresentada. Esta pesquisa tem o objetivo de servir como base para futuras pesquisas relacionadas à implementação de BCI's, com o intuito de mostrar diferentes métodos de implementação e testar uma abordagem inovadora, que potencialmente reduzirá os custos de recursos computacionais e monetários deste tipo de sistema.

De forma específica, todo o estudo foi feito com a intenção de implementar um algoritmo capaz de classificar, de forma assíncrona, quando um paciente imagina estar movimentando a mão direita, a mão esquerda, ou nenhum dos dois, tendo como entrada um banco de dados de gravações de EEG.

1.6 Organização e estrutura

Este trabalho será dividido em 5 capítulos principais. Cada capítulo conterá múltiplas seções, nas quais um tópico específico será discutido em cada uma.

- No presente Capítulo 1, a introdução ao trabalho, apresentamos conceitos fundamentais para o entendimento do mesmo, assim como delimitando os objetivos, metodologia e organização e estrutura.
- No Capítulo 2, faremos uma revisão teórica abrangente em cima dos principais conceitos necessários para a compreensão deste trabalho como um todo. Este capítulo irá explicar: alguns conceitos básicos de Neurofisiologia; como funciona um eletroencefalograma; o que é Imageria Motora; como funciona um método de aprendizagem de máquina para reconhecimento de padrões conhecido como Redes Neurais Artificiais Profundas; e a importância de utilizar Programação Paralela em algoritmos de aprendizagem de máquina;
- No Capítulo 3, falaremos sobre o desenvolvimento dos algoritmos utilizados para o teste do método proposto. Conterá uma descrição do banco de dados utilizado, explicará o método de extração de características que nomeamos como Vectorencefalograma Temporal, descreverá os passos tomados para o reconhecimento de padrões dos sinais de EEG, e apresentará a Interface Gráfica de Usuário desenvolvida para a utilização do programa de aprendizagem de máquina;

- Na Capítulo 4, explicaremos como foi definido o método de testes dos modelos propostos de Redes Neurais Artificiais, quais as métricas que utilizamos para a avaliação dos resultados, e os resultados obtidos dos testes;
- Por último, no Capítulo 5, discutiremos sobre os resultados obtidos, e apresentaremos propostas para trabalhos futuros.

2 Revisão Teórica

2.1 Neurofisiologia

Para entender como uma Interface Cérebro-Máquina funciona, é necessário ter um conhecimento básico dos fundamentos de Neurofisiologia - o estudo sobre o funcionamento do cérebro e terminações nervosas e seus papéis na conversão de pensamentos em ações, e dos sentidos em reações. Por esta razão, esta seção será uma revisão de Neurofisiologia, baseada no livro de Guyton e Hall (2013). Os conceitos aqui apresentados terão o intuito de trazer inspirações para novas técnicas e métodos para a implementação de uma Interface Cérebro-Máquina. Ressalta-se que a presente seção pode ser ignorada pelos leitores que não buscam um conhecimento mais detalhado sobre o assunto em análise.

2.1.1 Potencial de Membrana

A transmissão de informações através do corpo humano é feita de forma elétrica, química, ou eletroquímica. Nas fibras nervosas (extensões das células nervosas, como axônios e dendritos) e fibras musculares (células que compõem os músculos), essa transmissão acontece primariamente de forma elétrica, e será descrita a seguir.

Uma diferença de potencial entre o exterior da célula e seu interior existe mesmo quando a célula está em repouso (não há transmissão de sinais). Este potencial de repouso é chamado de Potencial de Membrana, e pode ser calculado segundo a Equação de Goldman (Equação 2.1):

$$F.Em(\text{milivolts}) = -61 \times \log \frac{C_{Na^+}_I P_{Na^+} + C_{K^+}_I P_{K^+} + C_{Cl^-}_O P_{Cl^-}}{C_{Na^+}_O P_{Na^+} + C_{K^+}_O P_{K^+} + C_{Cl^-}_I P_{Cl^-}} \quad (2.1)$$

onde $F.Em$ é a Força Eletromotriz, C é a concentração da substância em subscrito, P é a permeabilidade da membrana da célula para a substância em subscrito, Na^+ significa íons de sódio, K^+ significa íons de potássio, Cl^- significa íons de cloreto, I significa dentro da célula (*inside*) e O significa fora da célula (*outside*). Desta forma, $C_{Na^+}_I$ significa concentração de íons de sódio dentro da célula.

Analisando a Equação 2.1, pode-se observar alguns pontos-chave:

- A concentração de íons de sódio, de potássio e de cloreto no interior e no exterior da célula e a permeabilidade da membrana celular para estas substâncias determinam o Potencial de Membrana da célula em repouso.

- Um gradiente de íons positivos (Na^+ , K^+) fluindo de dentro da membrana para fora da célula causa uma diferença de potencial negativa, enquanto íons negativos (Cl^-) fluindo para fora da célula causa um potencial voltaico positivo.

Além disso, é importante saber que o gradiente de íons de sódio e de potássio são muito maiores que os de cloreto. Portanto, íons de cloreto não têm um peso significativo no cálculo do Potencial de Membrana de Repouso. O gradiente de concentração destes íons ocorre devido ao fenômeno de Difusão e pela Bomba de Sódio-Potássio, que serão discutidos a seguir.

A Figura 7 ilustra Canais Proteicos, presentes na maioria das membranas celulares, que permitem uma Permeabilidade Seletiva. A partir de diferenças no diâmetro do canal, do tamanho da molécula, das cargas elétricas na superfície interior da comporta, e de substâncias químicas presentes no Canal Proteico, o Filtro de Seletividade determina quais tipos de substância atravessarão a membrana celular através das comportas do Canal Proteico. Estes Canais Proteicos têm comportas que são controladas a partir de variações de voltagem no interior da célula e são o meio principal de transporte passivo (sem uso de energia) de íons, através do fenômeno da Difusão. A Difusão consiste em uma tendência das substâncias a tentar atingir o equilíbrio de concentração ao longo do volume em que estão presentes.

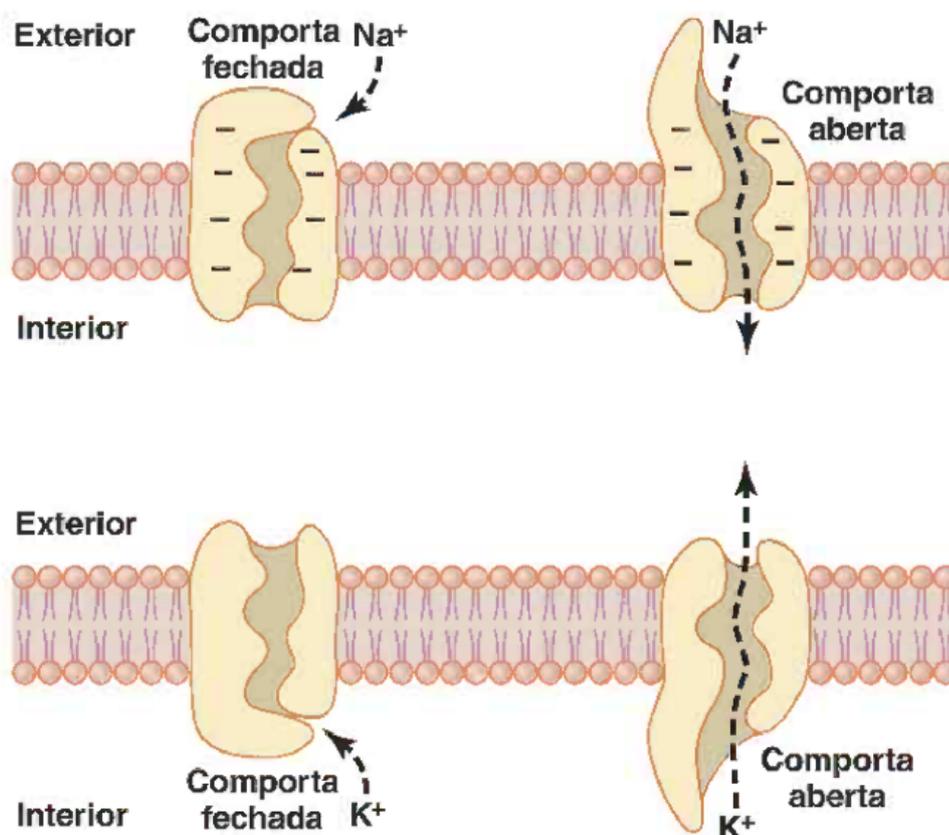


Figura 7: Transporte de íons de potássio e cloreto através da membrana celular, pelos Canais Proteicos

A Figura 8A apresenta uma maior concentração de íons de potássio no interior da membrana da fibra nervosa (que, neste caso, é permeável somente à íons de potássio) do que no exterior da célula. Desta forma, íons de potássio fluem para fora da célula através dos Canais Proteicos da membrana, levando consigo cargas positivas, e causando uma eletronegatividade no interior da célula. Após alguns milissegundos, a diferença de potencial se torna suficientemente grande para fechar as comportas dos Canais Proteicos, impedindo a passagem de íons de potássio, e mantendo sua eletronegatividade. Na Figura 8B, o mesmo fenômeno ocorre para uma membrana permeável somente a íons de sódio, onde a concentração de íons sódio é maior no exterior da membrana do que no interior, e os íons de sódio tendem a fluir para dentro da fibra nervosa.

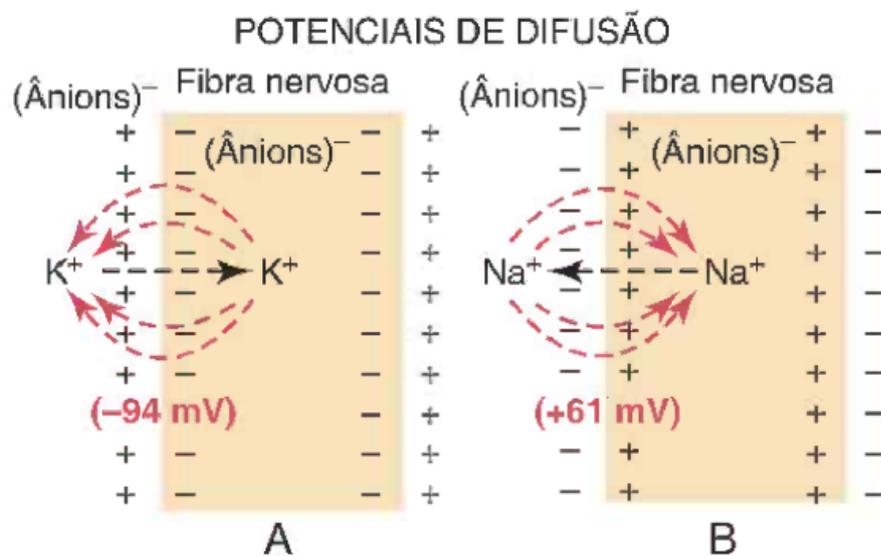


Figura 8: **A:** Difusão de íons de potássio para fora da célula, através de membrana que só é permeável a íons de potássio. **B:** Difusão de íons de sódio para dentro da célula, através de membrana que só é permeável a íons de sódio.

Além do fenômeno da Difusão, íons de sódio são transferidos para fora e íons de potássio são transferidos para dentro da célula de forma contínua. Para que isto ocorra, é necessário que haja uma quebra de ATP em ADP, o que libera energia para a célula. Este processo de transporte ativo de íons sódio e potássio pela membrana é realizado pela Bomba de Sódio-Potássio (ou Bomba de $\text{Na}^+ - \text{K}^+$), conforme ilustrado na Figura 9. Como se pode ver, mais íons de sódio são transferidos para o interior da célula do que íons de potássio para o exterior. Isto causa uma diferença de potencial negativo no interior da célula, que é adicionada ao potencial gerado pela Difusão.

Utilizando-se valores padrões de gradientes de concentração e permeabilidade (que não fazem parte do escopo deste trabalho) e aplicando-se a Equação de Goldman, conclui-se que uma fibra nervosa periférica possui -90 mV em repouso.

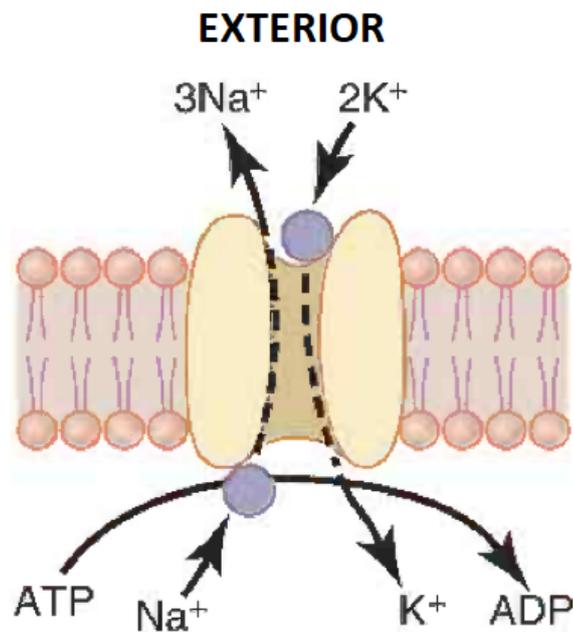


Figura 9: Bomba de sódio-potássio. ATP, Adenosina Trifosfato; ADP, Adenosina Difosfato; Na^+ , íon de sódio, K^+ , íon de potássio

2.1.2 Potencial de Ação

Para que ocorra a transmissão de sinais através das fibras nervosas, é necessário que ocorra uma grande variação no Potencial de Membrana da célula. Esta variação é chamada de Potencial de Ação.

Como mencionado na Seção 2.1.1, nas membranas das fibras nervosas existem Canais Proteicos de Permeabilidade Seletiva regulados por voltagem. O efeito de um Potencial de Ação nestes canais é ilustrado na Figura 10.

Quando a voltagem no interior da célula varia positivamente até determinado ponto (aproximadamente -65mV nas fibras nervosas periféricas), as comportas de íons de sódio abrem subitamente, aumentando o influxo destes íons carregados positivamente para dentro da célula, o que aumenta positivamente ainda mais o potencial da célula. O potencial da célula continua a aumentar até que ela fique despolarizada chegando a $+35\text{mV}$ em células do sistema nervoso. Neste ponto de despolarização, as comportas de íons de sódio se fecham lentamente, e as comportas de íons de potássio se abrem lentamente, fazendo com que íons de potássio fluam para fora da célula, e esta retome a polarização de -90mV . O ponto de aproximadamente -65mV é chamado de limiar de excitação - se o potencial da célula não atinge o limiar de excitação, não ocorre o Potencial de Ação.

A Figura 11 representa graficamente a propagação do potencial de ação ao longo da fibra nervosa. Na Figura 11A, a célula está em repouso. Na Figura 11B, um potencial de ação é gerado no meio da fibra. Como o interior da célula neste ponto ficou carregado

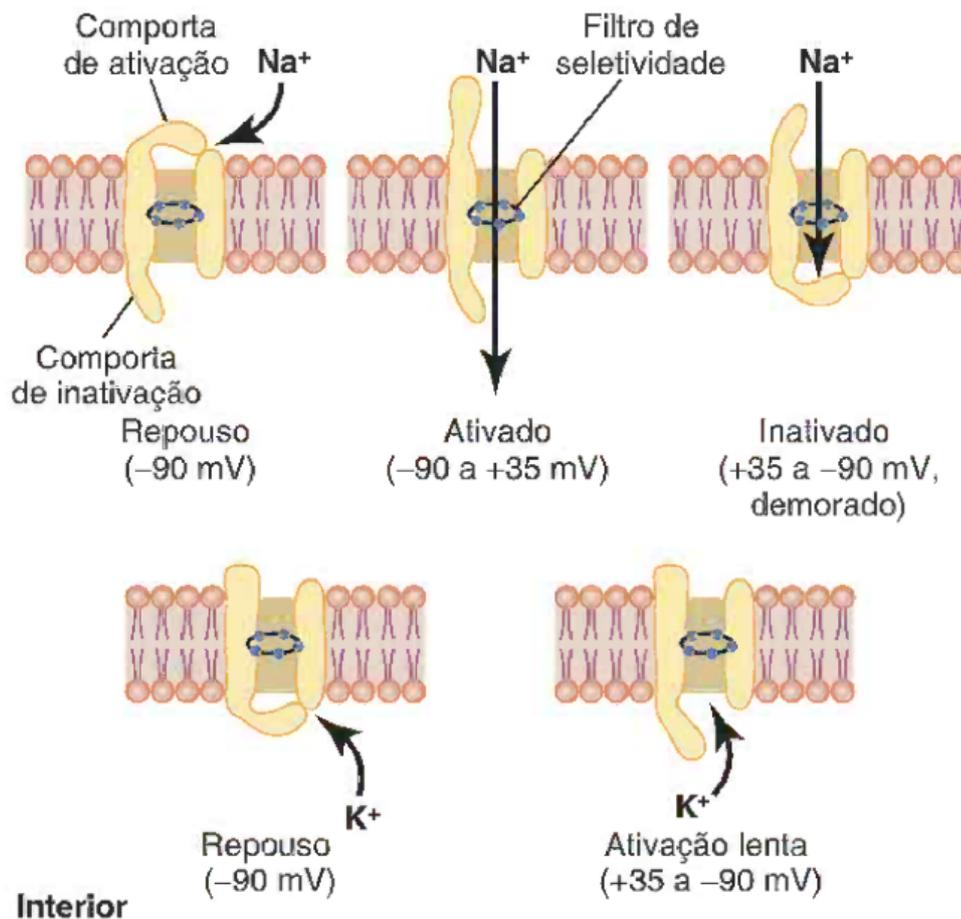


Figura 10: Ativação dos canais regulados por voltagem. Canal de sódio acima e canal de potássio abaixo.

positivamente, nos pontos adjacentes a voltagem aumenta à ponto de atingir o limiar de excitação, abrindo os canais de sódio nestes pontos adjacentes, até que o interior da célula fique despolarizado também nestes pontos, representado pela Figura 11C. Este processo se repete continuamente através da membrana, propagando o potencial de ação nas duas direções da fibra nervosa, como visto na Figura 11D.

O processo descrito reduz minimamente a diferença de concentração de sódio e de potássio. Entretanto, após longos períodos contínuos de estímulo, a diferença de concentração destas substâncias no interior e no exterior da célula pode chegar perto do equilíbrio, impedindo o processo de excitação. Para regular estas concentrações, a Bomba de Sódio-Potássio age continuamente da mesma forma como descrito na Seção 2.1.1. Entretanto, como dito, este processo requer o uso de ATP; ou seja, há um gasto de energia para a utilização contínua destas fibras.

O potencial de ação se propaga ao longo das fibra nervosa até que esta chegue ao seu fim. Deste ponto, duas coisas podem ocorrer: ou a fibra nervosa está conectada à fibras musculares, ou à outra fibra nervosa. Se no fim da fibra nervosa, ocorre uma junção neuromuscular, que é composta por terminações da fibra nervosa conectadas a várias fibras

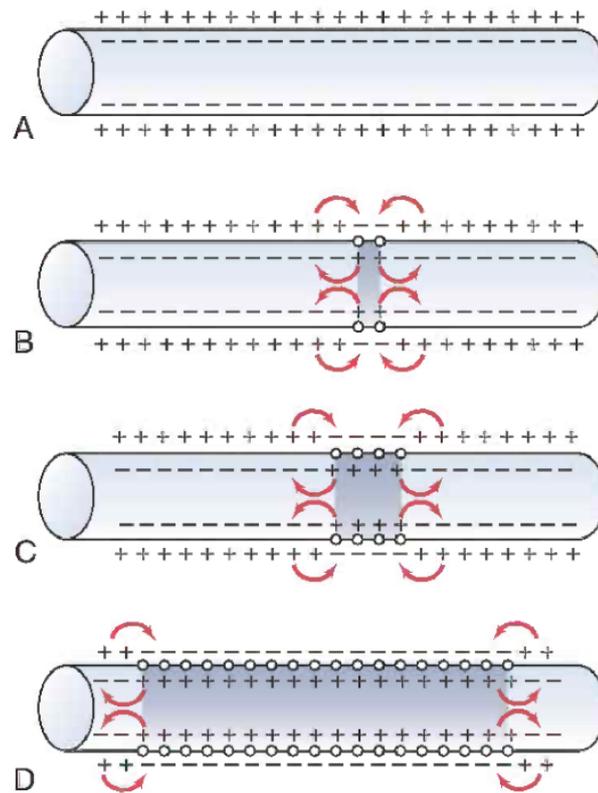


Figura 11: Propagação do potencial de ação ao longo de uma fibra condutora.

musculares, o mesmo processo de potencial de ação ocorre nesta, aumentando o fluxo de íons de sódio no interior da fibra muscular, e se propagando ao longo desta. Este potencial de ação, por sua vez, faz com que íons de cálcio sejam liberados, o que desencadeia forças atrativas entre filamentos do músculo e fazendo com que estes filamentos deslizem. Este deslizamento dos filamentos é o processo de contração do músculo. A fisiologia muscular não entra no escopo deste trabalho, e é descrita com maiores detalhes nos capítulos 6 e 7 do livro de referência desta seção (GUYTON; HALL, 2013).

2.1.3 Sinapses

Quando um potencial de ação se propaga até o fim de uma fibra nervosa e há outra fibra nervosa conectada à primeira, ocorre o que chamamos de sinapse. A sinapse é a transmissão de informações de uma fibra nervosa à outra, e esta pode ser de feita elétrica, como no caso do sistema nervoso autônomo, para contrair músculos lisos (músculos dos órgãos internos) e músculos cardíacos; e pode ser feita de forma química, como no caso do processamento dos sentidos e movimentos de membros. A maior diferença entre uma sinapse elétrica e uma sinapse química é que, no primeiro caso a transmissão de informações é bidirecional, e no segundo caso é unidirecional. Como as sinapses químicas são as mais utilizadas nas funções motoras, daremos maior ênfase à esta.

A Figura 12 ilustra um neurônio típico do sistema nervoso central, apontando

cada uma de suas partes, e o sentido do impulso nervoso nesta. O neurônio é a célula responsável pela transmissão de informações através do corpo. Um corpo humano contém mais de 100 bilhões de neurônios. As fibras nervosas discutidas nas seções anteriores são as extensões destes neurônios, e podem ser axônios ou dendritos. Os dendritos são também chamados de fibras de entrada ou fibras aferentes, e os axônios podem ser chamados de fibras de saída ou fibras eferentes. Como se pode observar na figura, o impulso nervoso começa pelos dendritos, passa pelo corpo celular, e segue até o final do axônio. No fim do axônio encontram-se terminações nervosas que, por sua vez, se conectam aos dendritos ou corpo celular de outro neurônio, mas nunca encostam nestes.

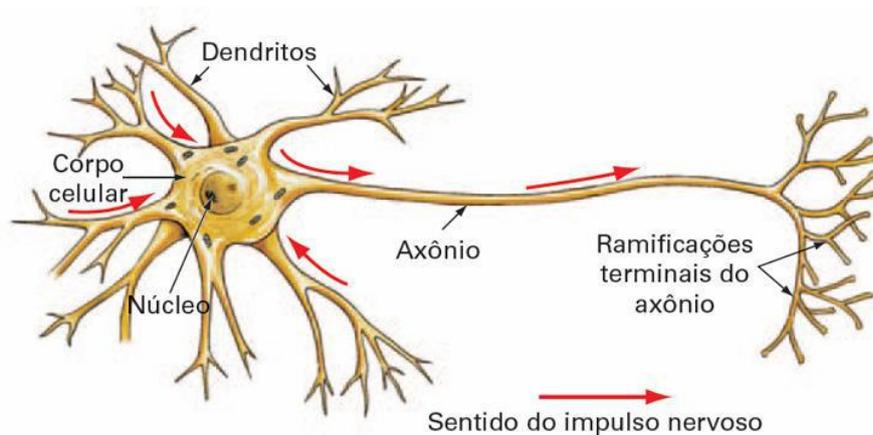


Figura 12: Ilustração de um neurônio típico do sistema nervoso central

Um neurônio motor típico tem seu valor de potencial de membrana em aproximadamente -65mV , quando em repouso. Além disso, no interior do corpo celular neuronal, esta voltagem é praticamente uniforme, devido à propriedade eletrolítica do seu líquido intracelular. Além disso, com um diâmetro grande, o corpo celular do neurônio possui uma baixa resistência, o que faz com que alterações no potencial em um ponto do corpo celular provavelmente irão afetar todo o corpo celular igualmente.

A Figura 13 demonstra um processo de sinapse química. Quando um potencial de ação é conduzido até o final de uma ramificação da terminação nervosa do axônio no neurônio pré-sináptico, seus canais de íons de cálcio regulados por voltagem são abertos, fazendo com que haja um fluxo destes íons para o interior da membrana. Estes íons fazem com que as vesículas sinápticas - que carregam os neurotransmissores - liberem estes neurotransmissores para a fenda sináptica (espaçamento entre um neurônio conectado ao outro). Estes neurotransmissores são substâncias químicas que, quando acopladas às proteínas receptoras (neuroreceptores) contidas na membrana dos dendritos ou do corpo celular do neurônio pós-sináptico, induzem a abertura ou fechamento de canais de íons de sódio, potássio, e/ou cloreto nesta célula pós-sináptica, e podem até modificar a estrutura funcional do neurônio pós-sináptico (e.g. aumentar ou diminuir o número de neuroreceptores).

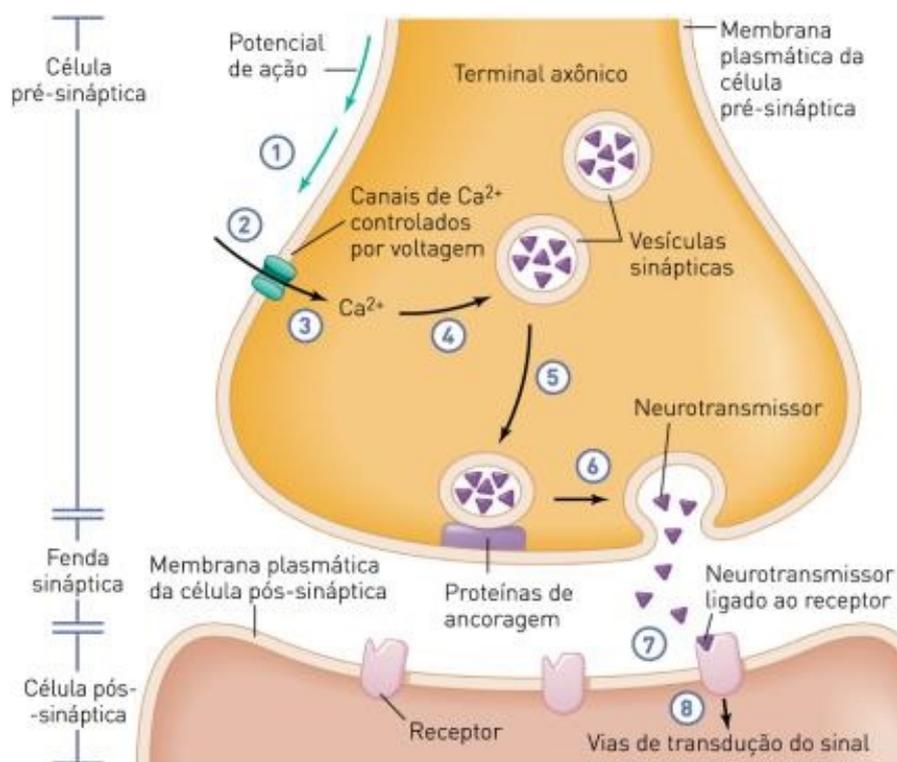


Figura 13: Processo de sinapse química

Quando os neurotransmissores induzem a abertura dos canais de íons de sódio na célula pós-sináptica, a sinapse é chamada de Excitatória. Os resultados da abertura dos respectivos canais fazem com que o neurônio pós-sináptico tenha seu potencial de membrana mais positivo, deixando-o mais próximo do limiar de excitação, facilitando a ativação do mesmo. Por outro lado, quando o neurotransmissor induz a abertura dos canais de cloreto e/ou potássio, a sinapse é Inibitória. A Sinapse Inibitória faz com que o potencial de membrana do neurônio pós-sináptico se torne mais negativo, o que faz com que outras sinapses provindas de outros neurônios tenham mais dificuldade de aumentar o potencial de membrana deste neurônio até o limiar de excitação. Além disso, há neurotransmissores que causam efeitos de longo prazo no neurônio pós-sináptico, como aqueles que causam modificações na estrutura da célula pós-sináptica. Existem mais de 50 neurotransmissores catalogados, e eles são agrupados de acordo com a função que exercem no neurônio pós-sináptico. A Figura 14 demonstra um neurônio motor típico, em três estados diferentes (repouso, excitado e inibido), e seus respectivos potenciais de membrana do corpo celular. Um detalhe que também pode ser observado na Figura 14 é que, quando o neurônio é excitado até atingir o limiar de ativação (-45mV , neste caso), um potencial de ação é originado no segmento inicial do axônio, que percorre ao longo de toda a fibra nervosa, causando o que chamamos anteriormente de impulso nervoso neste neurônio, que irá originar uma outra sinapse. Como a maioria dos dendritos não possuem canais de sódio, eles não propagam o potencial de ação da célula pós-sináptica para a pré-sináptica.

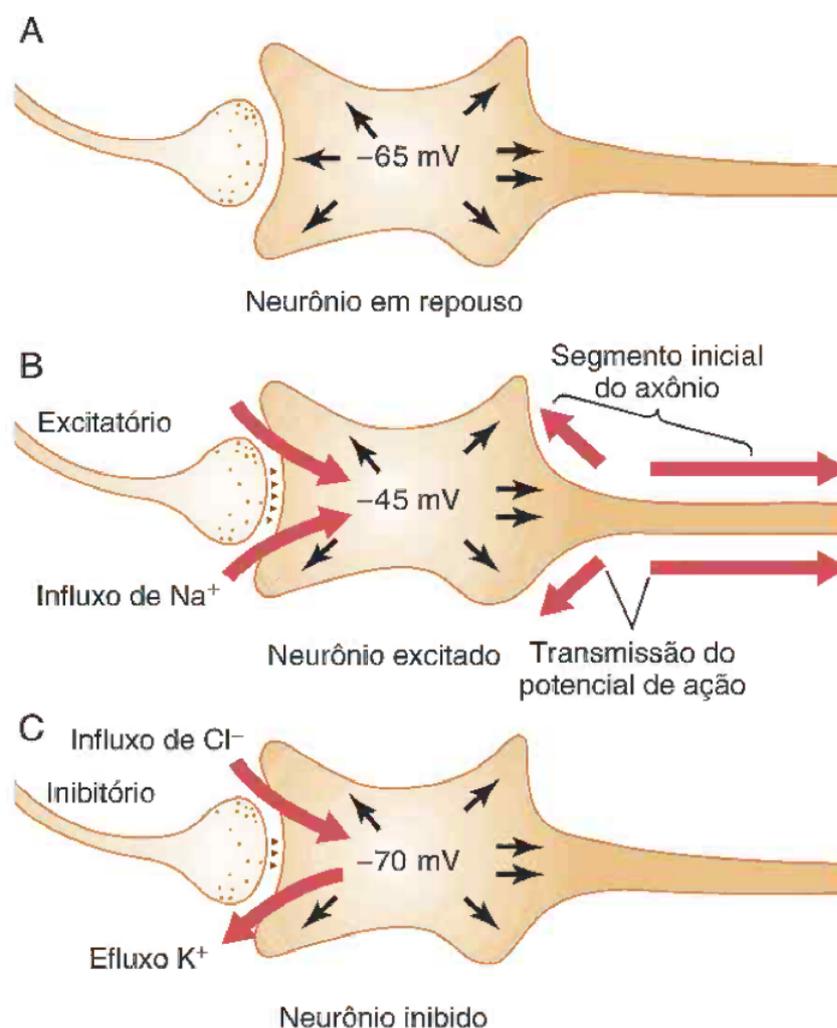


Figura 14: Estados do neurônio. **A:** Neurônio em repouso; **B:** Neurônio excitado; **C:** Neurônio inibido.

Como visto na Figura 15, várias sinapses podem ocorrer simultaneamente. Uma sinapse dura aproximadamente 15ms, e qualquer outra sinapse que acontecer neste meio tempo (provinda da mesma terminação nervosa, ou em outra parte do neurônio), terá seu valor somado à primeira. O resultado desta soma é então adicionado ao valor de potencial de membrana em repouso do corpo celular neuronal. Além disso, uma sinapse ocorrida nos dendritos terão uma resistência até chegar na região do corpo celular, diminuindo seu efeito ao longo da distância dos mesmos. Desta forma, uma só sinapse não tem capacidade o suficiente para fazer um neurônio em repouso elevar seu potencial de membrana até o limiar de ativação (uma sinapse causa uma diferença de potencial de aproximadamente 0,5mV a 1.0mV, sendo este valor positivo se for uma sinapse excitatória, ou negativo se for inibitória).

Por fim, um agrupamento de neurônios é chamado de Circuito Neuronal, e existem diversos tipos de circuitos, exercendo funções diferentes no processamento de sinais no cérebro. Por exemplo, existem circuitos de amplificação de sinal, de oscilação através de *feedback* positivo, de estabilização dos circuitos (evitando convulsões epiléticas e excesso de

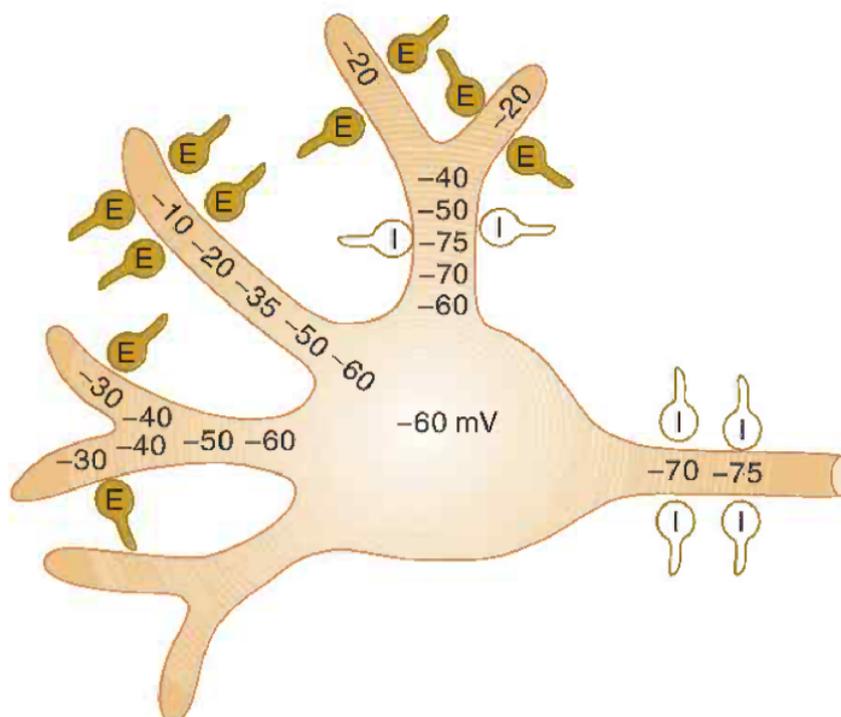


Figura 15: Somação espacial de um neurônio. Os números representam o potencial de membrana naquele ponto do neurônio, e os terminais com letras representam as terminações nervosas do axônio pré-sináptico. **E**: Excitação; **I**: Inibição

tensionamento de músculos), dentre outros. Um conjunto de circuitos neuronais possibilita a transmissão de informações dos cinco sentidos e os movimentos dos membros do corpo humano.

2.1.4 Funções Motoras

O córtex cerebral é dividido em várias áreas diferentes, cada um com suas funções específicas. Na Figura 16 observamos algumas destas principais áreas. Entretanto, observando o propósito deste trabalho, nos atemos a mencionar que o planejamento de ações e movimentos se localiza no lobo frontal, que o processamento do tato ocorre no lobo parietal, que o lobo occipital é responsável pelo processamento de imagens provindas dos olhos, e que o lobo temporal processa os sinais audíveis provindos dos ouvidos. Além disso, importa notar que o cerebelo também possui importância no planejamento de ações e movimentos do corpo.

O lobo frontal contém o córtex motor, que é dividido em três subáreas - o córtex motor primário, a área pré-motora e a área motora suplementar - conforme na Figura 17. Acredita-se que o córtex motor suplementar é responsável por movimentos bilaterais (como o movimento de agarrar algo com as duas mãos e rotação da cabeça e olhos). O córtex motor primário é responsável pela excitação dos conjuntos de músculos necessários para a realização de movimentos específicos. Quando se imagina um movimento complexo ou

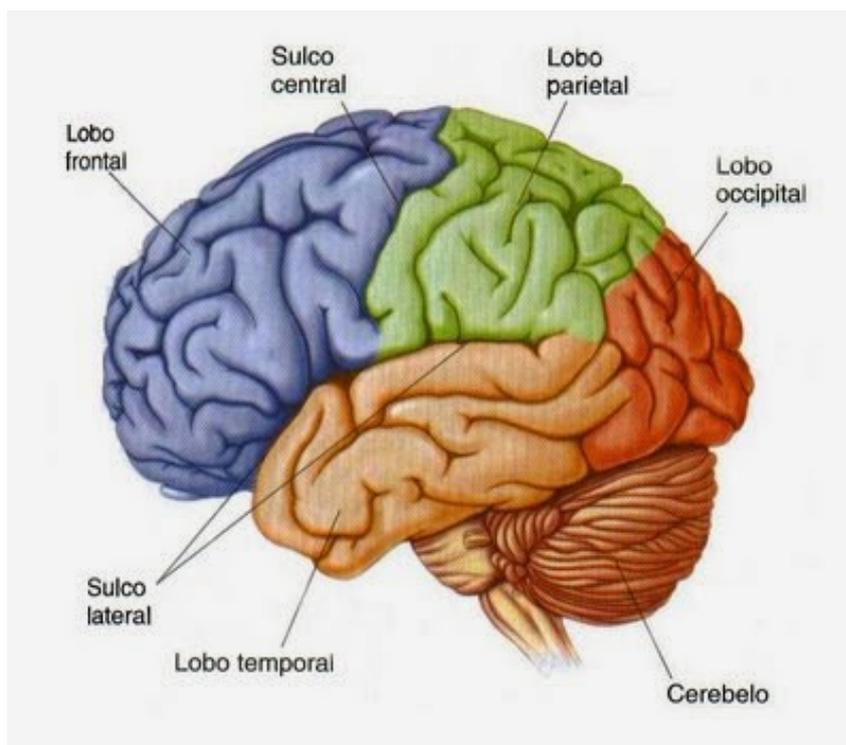


Figura 16: Áreas funcionais do cérebro

delicado de algum membro do corpo, este se origina na parte anterior do córtex pré-motor e envia a “imagem motora” para a parte posterior do córtex pré-motor; no córtex pré-motor posterior, a “imagem motora” é decodificada em sinais sucessivos de movimentação dos músculos para que se atinja o movimento imaginado; então, este padrão de movimentos é enviado ao córtex motor primário, que excita os neurônios responsáveis por ativar os circuitos neuronais de cada um destes movimentos.

O sinal proveniente do córtex motor pode ser enviado diretamente à medula espinhal (que redistribuirá os sinais para os músculos a serem excitados), ou pode passar antes por outros caminhos, tal como pelo cerebelo. Acredita-se que a parte lateral do cerebelo, também conhecida como cerebrocerebelo, recebe informações providas do córtex pré-motor, e ajuda no planejamento e temporização de transições de uma sequência de movimentos para outra, em movimentos complexos dos dedos e das mãos. O sinal é então reenviado ao córtex sensorimotor, funcionando como *feedback*. Os detalhes desta transmissão de informações ainda estão em discussão na comunidade científica.

2.2 Eletroencefalograma

Como mencionado na Seção 1.2, um dos métodos para monitoramento das atividades cerebrais é o Eletroencefalograma, ou EEG. Este método consiste em medir a voltagem emitida por agrupamentos de neurônios através de eletrodos posicionados no couro cabeludo.

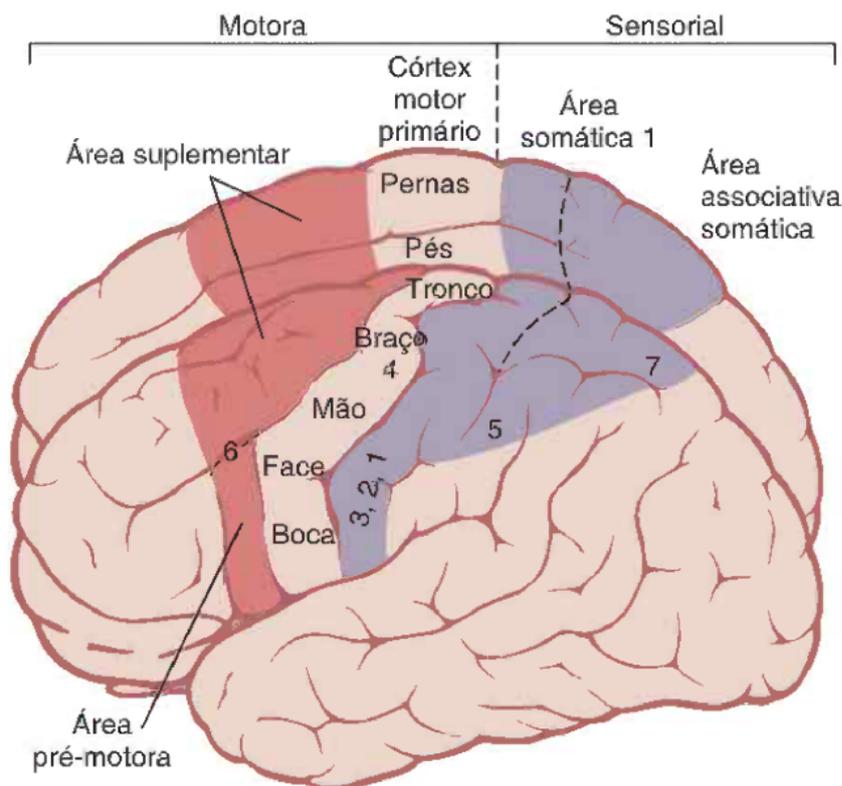


Figura 17: Áreas motoras e somáticas do córtex cerebral

Na Seção 2.1, explicamos como as atividades cerebrais (tais como o processamento dos sentidos e planejamento do movimento de membros) resultam em correntes elétricas no encéfalo e outras partes do corpo. Entretanto, somente diferenças de potencial causadas por impulsos de agrupamentos de milhares a milhões de neurônios disparando simultaneamente podem ser observadas através do couro cabeludo. Este potencial medido varia normalmente entre 0 e $100\mu V$. Como se trata de um valor muito pequeno, torna-se necessário amplificar este sinal, utilizando amplificadores operacionais em série, ou amplificadores instrumentais (de alta taxa de amplificação). Como o sinal será amplificado várias vezes, os ruídos também podem ser amplificados, tornando necessária a utilização de um filtro analógico de sinal, antes da amplificação do mesmo (TEPLAN, 2002).

O cérebro humano emite o que é chamado de ondas cerebrais, que são oscilações rítmicas na voltagem lida pelos eletrodos do eletroencefalograma. Estas ondas cerebrais são classificadas de acordo com a faixa de frequência em que aparecem. É possível separar as ondas cerebrais de diferentes frequências, passando as ondas cerebrais do domínio do tempo para o domínio da frequência, utilizando, por exemplo, a Transformada de Fourier (TEPLAN, 2002). Quatro grupos básicos de ondas cerebrais são definidos de acordo com a faixa de frequência em que atuam, e cada uma demonstra padrões diferentes de acordo com o estado funcional instantâneo do cérebro humano, como descrito na Tabela 1. É importante notar que diferentes trabalhos utilizam valores levemente diferentes

de faixas de frequência para categorizar as ondas cerebrais. A Tabela 1 apresenta as faixas descritas no trabalho de Amiri et al. (2013).

Tabela 1: Ondas cerebrais e respectivas faixas de frequência.

Onda Cerebral	Frequência(Hz)
Delta	1 a 4
Theta	4 a 7
Alpha	8 a 12
Mu	8 a 13
Beta	12 a 30
Gamma	25 a 100

Como discutido também na Seção 2.1, diferentes áreas do cérebro atuam em diferentes funções. Por este motivo, e para uma padronização do posicionamento físico dos eletrodos, foi desenvolvido o Sistema 10-20 de posicionamento dos eletrodos, ilustrado na Figura 18, o qual utiliza pontos de referência protuberantes no couro cabeludo, e proporções padronizadas de referência de distâncias. Desta forma, funções específicas são representadas por grupos específicos de eletrodos (e.g., os eletrodos posicionados no lobo occipital, nomeados com o prefixo “O” e “PO”, representam informações relacionadas à visão e processamento de imagens).

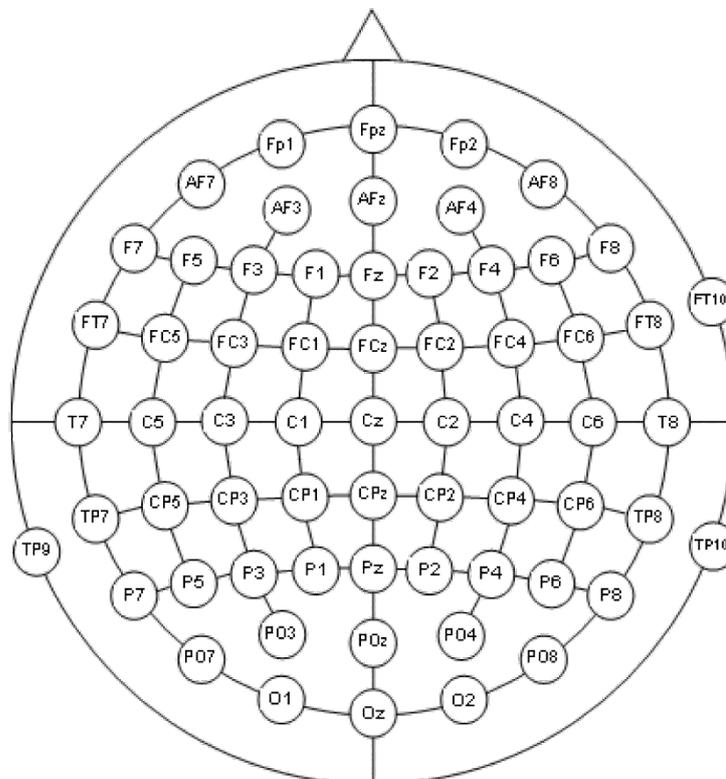


Figura 18: Sistema 10-20 de posicionamento de eletrodos com 64 canais.

2.2.1 Eletroencefalograma de Imageria Motora

Os ritmos Beta e Mu são amplamente utilizados na literatura para o reconhecimento de padrões de Imageria Motora: o ato de imaginar a movimentação de membros do corpo (PALANIAPPAN, 2014; AMIRI et al., 2013; HE et al., 2015; ANG et al., 2011; GUGER et al., 2003; KIM et al., 2016; LANCE et al., 2012; MAKEIG et al., 2012; MAYE et al., 2011). Tais estudos mostram que ao filtrar os sinais provenientes do eletroencefalograma nas frequências dos ritmos beta e mu - apresentados na Tabela 1 - melhora-se de forma significativa a eficácia de uma Interface Cérebro-Máquina (BCI) baseada em Imageria Motora (MI).

2.3 Aprendizagem de Máquina

A Aprendizagem de Máquina é uma subárea da Inteligência Artificial. Dado um conjunto de dados, o algoritmo de Aprendizagem de Máquina classifica os dados de entrada em outros conjuntos pré-definidos. E.g., em um conjunto de imagens, um algoritmo de aprendizagem de máquina pode ser implementado para se rotular as imagens em que um carro está presente. Ela está presente em algoritmos de reconhecimento de objetos (KRIZHEVSKY; SUTSKEVER; Geoffrey E., 2012) e faces (CHEN et al., 2000) em imagens, algoritmos de previsão de demanda (HO; HSU, 1992), classificação de arritmias cardíacas (GARCIA et al., 2017), BCI's (Jiralerspong, Trongmun, Liu, Chao, Ishikawa, 2014), entre outras várias aplicações.

Existem vários métodos de implementação de um algoritmo de Aprendizagem de Máquina, entre eles: Regressão Logística, Análise de Discriminantes Lineares (LDA), *Random Forest*, *Support Vector Machines* (SVM), Redes Neurais Artificiais (ANN), Redes Neurais Convolutivas (CNN), Redes Neurais Artificiais Profundas (DNN), *K-Nearest Neighbors* (KNN), entre outros. erir, se colocar o nome em Inglês tb, ou as iniciais em Português...ou elas são sempre No presente trabalho, utilizaremos as Redes Neurais Artificiais Profundas, devido à sua capacidade de classificar dados altamente não-lineares, e sua alta eficácia em problemas de aprendizagem de máquina inerentemente difíceis (KRIZHEVSKY; SUTSKEVER; Geoffrey E., 2012).

A Aprendizagem de Máquina se divide ainda em duas sub-áreas: aprendizagem supervisionada e Aprendizagem não-supervisionada. Na aprendizagem supervisionada, o algoritmo recebe, além dos dados de entrada, os rótulos de cada dado, e na aprendizagem não-supervisionada, o algoritmo não sabe os rótulos. Como as Redes Neurais Artificiais Profundas são utilizadas primariamente como aprendizagem supervisionada, daremos ênfase à esta área.

De forma geral, um algoritmo de aprendizagem de máquina supervisionada funciona da seguinte forma: antes de iniciar o algoritmo, o banco de dados é embaralhado e dividido

em três subconjuntos: conjunto de treinamento, conjunto de validação (ou avaliação) e conjunto de teste. Em seguida, o conjunto de treinamento é alimentado ao classificador em uma primeira etapa, denominada de fase de treino. Nesta etapa, o classificador aprenderá (de acordo com o algoritmo implementado) a separar os dados de entrada (também chamados de exemplos ou amostras) em conjuntos diferentes de classes. Na segunda etapa, chamada de fase de avaliação, o algoritmo tenta prever os rótulos de dados não vistos na fase de treinamento, sem saber os seus rótulos. Ainda nesta segunda fase, compara-se os resultados com os rótulos verdadeiros, para se calcular a eficiência do algoritmo. Caso o resultado não seja satisfatório, muda-se os hiper-parâmetros do algoritmo, embaralha-se novamente o banco de dados, divide-o em três subconjuntos e repete-se as duas primeiras fases. Quando um resultado satisfatório é obtido, utiliza-se o mesmo modelo treinado para prever os rótulos dos dados do conjunto de teste, e mede-se as métricas de eficácia para avaliação do algoritmo neste conjunto de teste. É importante que não se tome como melhor modelo aquele que deu o melhor resultado no conjunto de teste, e sim aquele que deu o melhor resultado no conjunto de validação, pois isso garante que as métricas de teste não sejam enviesadas (ou, em inglês, *biased*).

Alguns pontos importantes sobre a Aprendizagem de Máquina:

- De uma forma geral, utilizando os classificadores atuais, quanto maior o banco de dados, melhor será o resultado;
- Um teorema conhecido como *No Free Lunch Theorem* afirma que não existe um classificador que será melhor que todos os outros em todas as situações. (WOLPERT; MACREADY, 1997);
- Dependendo do problema de aprendizagem de máquina, pode-se realizar operações de pré-processamento (tais como filtragem e extração de características específicas) no banco de dados, antes de alimentá-lo ao classificador;
- Pode-se utilizar múltiplos classificadores em sucessão, seja para agilizar o processo ou para refinar os resultados;
- Um classificador pode ser considerado adaptativo ou *online*, quando ele continua aprendendo mesmo enquanto está em uso;
- Métodos de otimização (tais como *Particle-Swarm Optimization*, ou Genetic Algorithms) podem ser utilizados tanto para se definir os hiper-parâmetros do classificador - o que de forma geral, pode aumentar a eficácia do algoritmo - quanto para se definir as características mais relevantes para a classificação - o que pode agilizar tanto o processo de treinamento quanto de classificação.

2.3.1 Redes Neurais Artificiais Profundas

O modelo de Redes Neurais Artificiais (do inglês, *Artificial Neural Networks* ou ANN) é uma analogia feita com os neurônios humanos e como eles são conectados entre si. É importante notar que não há relação no uso de Redes Neurais Artificiais com uma Interface Cérebro-Máquina, além de sua possível utilização como um método de Aprendizagem de Máquina. Os neurônios de uma ANN não correspondem aos neurônios físicos do paciente, e os nomes utilizados são somente analogias.

Em uma ANN, para estimular as diferentes áreas do “cérebro”, um **peso** é designado a cada neurônio. Treinamos uma ANN para encontrar quais pesos são recomendados para cada problema de classificação.

Para treinar a ANN, primeiramente é necessário inicializar os pesos de cada neurônio. É importante que estes pesos sejam assimétricos, já que isto previne que o algoritmo falhe. Desta forma, ao invés de inicializar todos os pesos como zeros, é recomendado atribuir pesos aleatórios a cada neurônio.

Em seguida, é calculada a hipótese de saída, aplicando os valores de entrada $x^{(i)}$ às funções de ativação de cada camada/*layer* ($a_l(x^{(i)})$) multiplicada pelo peso de cada neurônio. Este processo é conhecido como propagação direta (ou, do inglês, *forward-propagation*).

O algoritmo posteriormente avalia o erro causado por cada neurônio, ao avaliar a função de custo que compara a hipótese de saída e o valor verdadeiro. Por último, subtrai-se o erro de cada peso. Este processo é chamado de propagação inversa (ou, do inglês, *back-propagation*). O algoritmo repete este processo de forma iterativa várias vezes, com o objetivo de minimizar a função de custo, dada como

$$J(\Theta) = \frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K \left[-y_k^{(i)} \log((h_{\Theta}(x^{(i)}))_k) - (1 - y_k^{(i)}) \log(1 - (h_{\Theta}(x^{(i)}))_k) \right] + \frac{\lambda}{2m} \left[\sum_{l=1}^{L-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (\Theta_{ji}^{(l)})^2 \right] \quad (2.2)$$

onde $\Theta_{ji}^{(l)}$ é a matriz de pesos da camada l , contendo os pesos de cada conexão entre o neurônio j na camada l e o neurônio i na camada $(l + 1)$; m é o número de exemplos; K é o número de rótulos, ou saídas; s_l é o número de neurônios (ou unidades) na camada l ; $x^{(i)}$ é o i^{th} exemplo; $y_k^{(i)}$ é o rótulo do $i^{\text{ésimo}}$ exemplo, correspondendo à classe k ; $h_{\Theta}(x^{(i)})_k$ pode ser visto como a probabilidade de um exemplo $x^{(i)}$ pertencer à classe k , dado um conjunto de pesos específicos Θ (se a função de ativação for uma sigmoide; e λ é o coeficiente de regularização).

O termo mais à direita da Equação (2.2) é chamado de termo de regularização, parametrizado por λ , e é utilizado para evitar que o algoritmo sofra de *overfitting* (i.e.,

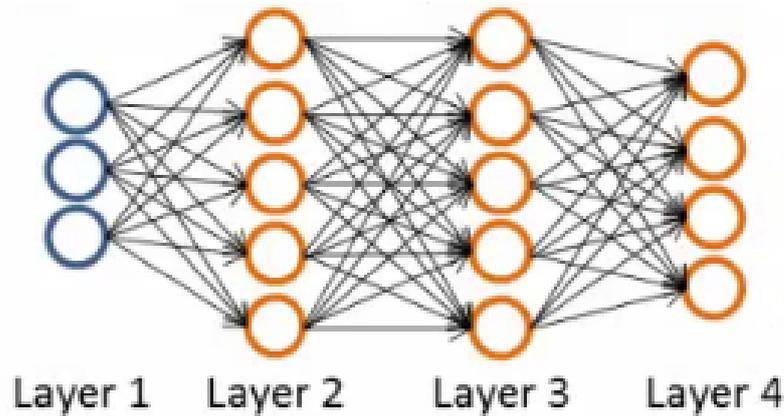


Figura 19: Exemplo de uma Rede Neural Artificial, com $L = 4$, $K = 4$, $s_1 = 3$, $s_2 = 5$, etc...

quando o classificador rotula bem a maioria dos dados de treinamento mas falha na predição dos conjuntos de teste).

Um exemplo de arquitetura de uma ANN é ilustrada na Figura 19. Como todos os neurônios de uma camada está conectada à todos os neurônios das camadas próximas, esta ANN é classificada como uma ANN totalmente conectada (ou, do inglês, *fully connected*).

Quando uma ANN possui várias camadas de neurônios, esta passa a se chamar Rede Neural Artificial Profunda (do inglês, *Deep Neural Network*, ou DNN). Dois problemas das DNN's são: o resultado da *back-propagation* passa a se tornar um valor cada vez menor, devido às diferenciações sucessivas na otimização, dificultando a classificação da função de custo; e uma maior possibilidade de ocorrer *overfitting*. Para resolver este problema, alguns métodos são utilizados. Um deles é conhecido como *Dropout*. O *Dropout* consiste em desconectar alguns neurônios aleatoriamente, de acordo com uma probabilidade pré-definida. Desta forma, reduz-se os casos de *overfitting* e de desaparecimento do resultado da *back-propagation* (SRIVASTAVA et al., 2014). A Figura 20 ilustra uma representação de Rede Neural com 2 camadas escondidas antes e depois da aplicação da técnica de *Dropout*.

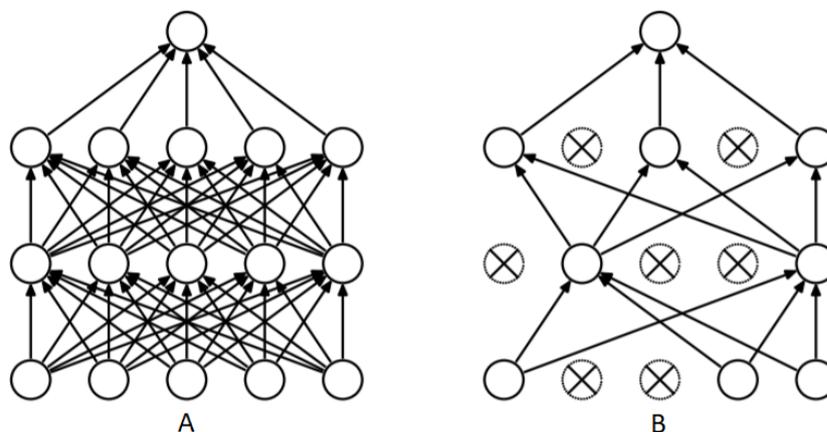


Figura 20: **A:** Rede Neural Artificial totalmente conectada. **B:** Rede Neural Artificial após a aplicação da técnica de *Dropout*.

2.4 Programação Paralela

A Programação Paralela (também conhecida como Programação de Unidade de Processamento Gráfico de Propósitos Gerais ou *GPGPU Programming*) é utilizada em diversos ramos da computação, tais como processamento de imagens, simulações e aprendizagem de máquina (OWENS et al., 2007). Seu conceito básico parte do princípio de que, ao realizar múltiplas vezes a mesma operação com dados diferentes, um processador com múltiplos núcleos lentos entrega os resultados mais rapidamente do que um processador com somente um núcleo muito rápido. Este conceito é chamado de *throughput*. Como um exemplo, observe a Figura 21. O carro representa um processador convencional (mais rápido) com 5 núcleos, e o ônibus, um processador gráfico (mais lento) com 40 núcleos. Digamos que ambos precisam realizar a tarefa de levar 40 pessoas (40 operações) por uma distância de 160km (a distância representa o tempo que cada operação demora para ser realizada). Calculamos então, que o ônibus demorará 4h para chegar ao seu destino (terminar de realizar todas as 40 operações, sendo realizadas simultaneamente) e - se não contarmos o tempo de retorno, para facilitar a analogia com os processadores - o carro precisará de 8h.

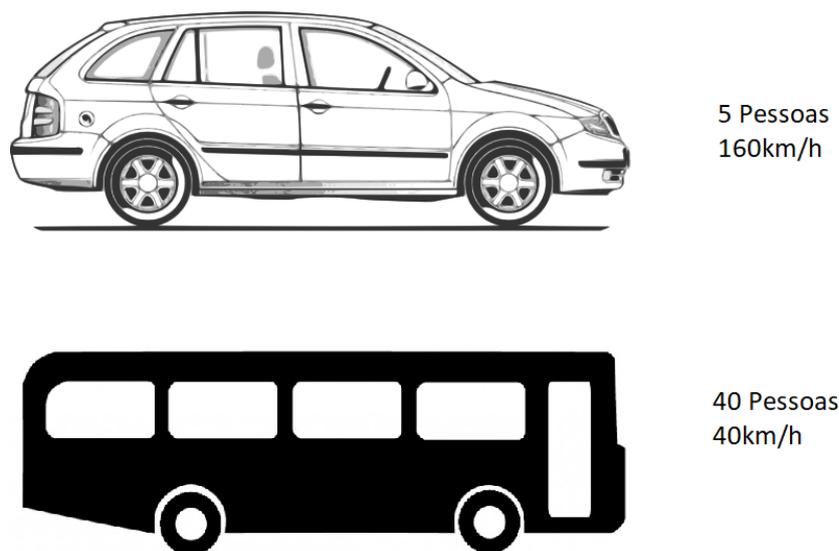


Figura 21: Representação prática de comparação de *throughput* entre Programação Serial (carro) e GPGPU (ônibus).

Na prática, a transferência de dados da memória da CPU para a memória da GPU e vice-versa toma relativamente muito tempo. Assim, só é recomendada a utilização de Programação Paralela quando houver muito processamento similar em vários dados de uma vez. Como a aprendizagem de máquina envolve muitos dados e, em sua maioria, operações matriciais repetidas, há um grande aumento de performance ao se utilizar Programação de GPGPU (STRIGL; KOFLER; PODLIPNIG, 2010). Como pode ser visto

na Figura 22, a performance de execução de uma CNN aumenta consideravelmente quando se utiliza Programação Paralela (*GPU*), em comparação com Programação em CPU Trivial (*CPU_{triv.}*) e Programação em CPU Otimizada (*CPU_{opt.}*). Pode-se observar que tanto um aumento nos dados de entrada quanto um aumento na complexidade da arquitetura das Redes Neurais (número de neurônios nas camadas) resultam em uma maior diferença entre a utilização da GPU e da CPU no processamento. Por este motivo, tanto ao utilizar DNN's, quanto ao processar muitos dados (como no caso do Banco de Dados utilizado neste trabalho, que será descrito detalhadamente na Seção 3.1.1), recomenda-se fortemente o uso de Programação Paralela.

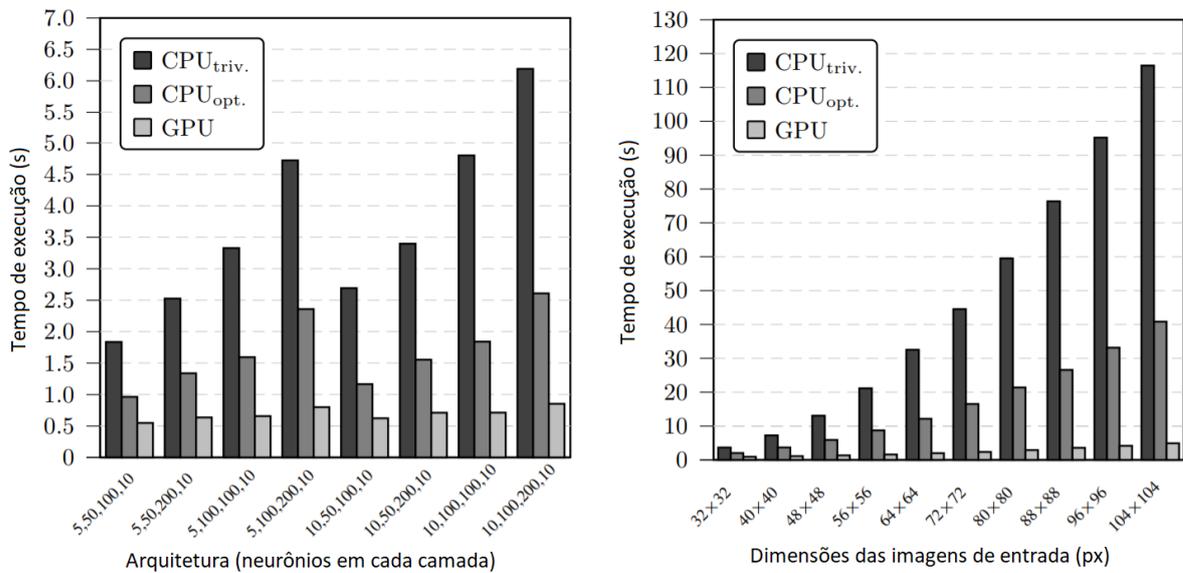


Figura 22: Comparação entre Programação Paralela e Programação Convencional aplicada ao reconhecimento de

Existem duas principais linguagens de programação paralela utilizadas atualmente: OpenCL - baseada em OpenGL, multi-plataforma, e de arquitetura-aberta; e CUDA - baseada em C, atualmente também multi-plataforma e proprietária da NVIDIA. Existem diversas bibliotecas de Aprendizagem de Máquina que utilizam Programação Paralela, e sua grande maioria são implementadas em CUDA, tais como Theano, Caffe, Torch, Keras e TensorFlow. Estas bibliotecas facilitam de forma significativa a implementação de sistemas de aprendizagem de máquina utilizando a GPU. Neste trabalho, optou-se por utilizar o TensorFlow, devido à sua alta performance, flexibilidade e facilidade de uso (ABADI et al., 2016).

3 Desenvolvimento

3.1 Banco de Dados

3.1.1 Descrição do EEG Motor Movement/Imagery Dataset

Para o treinamento do classificador, foi utilizado o banco de dados EEG Motor Movement/Imagery Dataset (ou EEGMMIDB) (GOLDBERGER et al., 2000). Este banco de dados consiste numa coletânea de gravações de EEG obtidas a partir de 109 pacientes. Enquanto os pacientes realizavam **tarefas** específicas, um sistema de EEG com 64 canais, conhecido como BCI-2000 (SCHALK et al., 2004) digitalizava os dados provenientes dos eletrodos. Cada paciente executou 14 **experimentos**: 2 experimentos de base de 1 minuto cada (um com os olhos fechados e outro com os olhos abertos), e 3 experimentos de dois minutos cada com as seguintes tarefas:

1. Um alvo aparece ou no lado esquerdo ou no lado direito da tela. O paciente fecha e abre o punho correspondente até o alvo desaparecer. Então, o paciente relaxa.
2. Um alvo aparece ou no lado esquerdo ou no lado direito da tela. O paciente imagina estar fechando e abrindo o punho correspondente até o alvo desaparecer. Então, o paciente relaxa.
3. Um alvo aparece ou na parte de cima ou na parte de baixo da tela. O paciente fecha e abre os dois punhos (se o alvo estiver em cima) ou fecha e abre os dois pés (se o alvo estiver embaixo) até o alvo desaparecer. Então, o paciente relaxa.
4. Um alvo aparece ou na parte de cima ou na parte de baixo da tela. O paciente imagina estar fechando e abrindo os dois punhos (se o alvo estiver em cima) ou imagina estar fechando e abrindo os dois pés (se o alvo estiver embaixo) até o alvo desaparecer. Então, o paciente relaxa.

Em suma, os 14 experimentos de cada paciente eram:

1. Base, olhos abertos
2. Base, olhos fechados
3. Tarefa 1

4. Tarefa 2
5. Tarefa 3
6. Tarefa 4
7. Tarefa 1
8. Tarefa 2
9. Tarefa 3
10. Tarefa 4
11. Tarefa 1
12. Tarefa 2
13. Tarefa 3
14. Tarefa 4

O banco de dados é fornecido em formato EDF+ (*European Data Format Plus*), contendo os 64 canais do EEG digitalizados a 160 amostras por segundo, além da informação de qual paciente está realizando qual experimento e um cabeçalho, que contém informações como a frequência de amostragem, o pré-filtro utilizado, a data de gravação, as anotações (rótulos), e outras informações secundárias. As **anotações** contém informações do que o paciente fazia em cada momento, contendo o momento de início do evento (seja ele o movimento ou imaginação de mover a mão direita, a mão esquerda, ou manter-se em repouso) e a duração do mesmo. Estas anotações são classificadas em três códigos:

- **T0** corresponde ao descanso;
- **T1** corresponde ao início da movimentação (real ou imaginada) do punho esquerdo (nos experimentos 3, 4, 7, 8, 11 e 12) ou dos dois punhos (nos experimentos 5, 6, 9, 10, 13, e 14);
- **T2** corresponde ao início da movimentação (real ou imaginada) do punho direito (nos experimentos 3, 4, 7, 8, 11 e 12) ou dos dois pés (nos experimentos 5, 6, 9, 10, 13, e 14).

O sistema de aquisição de sinais de EEG deste banco de dados utiliza o Sistema 10-20 de posicionamento dos eletrodos (como descrito na Seção 2.2), mas exclui os eletrodos Nz, F9, F10, FT9, FT10, A1, A2, TP9, TP10, P9, e P10, como pode-se observar na Figura 23

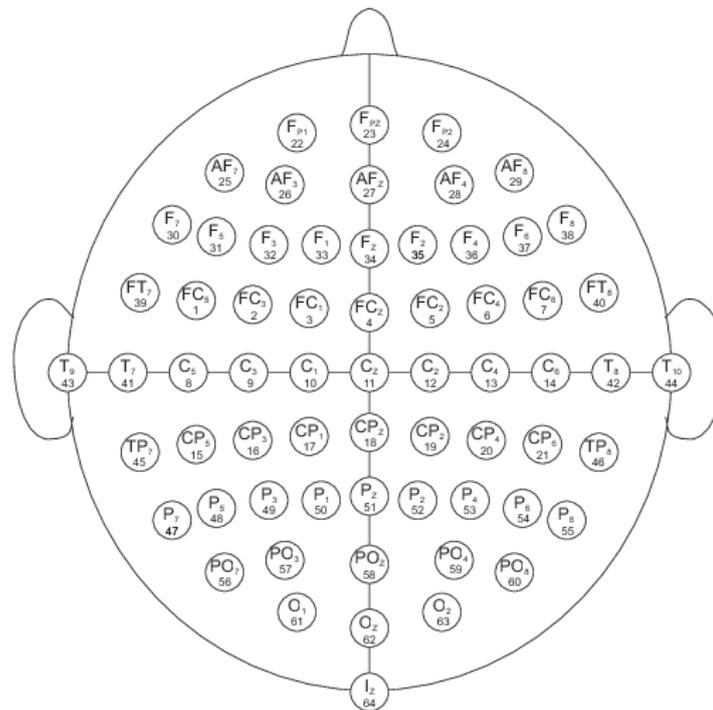


Figura 23: Posicionamento dos eletrodos no sistema de aquisição de dados do EEGMMIDB.

3.1.2 Pré-Processamento do Banco de Dados: Sliding Window

Como o foco deste trabalho é implementar eventualmente uma BCI capaz de classificar assincronamente quando um paciente imagina movimentar os punhos, e qual punho ele está imaginando movimentar, utilizamos em nosso algoritmo os dados dos experimentos 1, 2, 3, 4, 7, 8, 11 e 12 e descartamos os outros. Utilizamos os dados de todos os 109 pacientes, e utilizamos somente os eletrodos Cz, C3, e C4, já que, como determinado experimentalmente, são os mais relevantes para a classificação de Imagaria Motora das mãos (SLEIGHT; PILLAI; MOHAN, 2009; NEUPER; PFURTSCHELLER, 2001; DEECKE; WEINBERG; BRICKETT, 1982) o que permite uma grande redução no tempo de processamento e necessidade de memória para execução do algoritmo de treinamento do classificador. Além disso, como apresentado na Seção 2.2.1, é comum filtrar os sinais nas faixas de frequência mu e beta. Entretanto, analisando o trabalho de (QUEIROZ et al., 2015), o melhor resultado obtido foi quando não foi utilizado nenhum filtro digital. Por este motivo, não realizamos nenhum tipo de filtragem digital no presente trabalho.

Para a leitura, organização, exclusão dos dados não necessários, e reformatação do banco de dados, foi implementado um algoritmo em MATLAB. Como o banco de dados tem um tamanho relativamente grande (aproximadamente 3,4GB) para os computadores pessoais atuais, foi necessária cautela com o gerenciamento de memória RAM, para que não haja problemas de falta de memória durante a execução, o que interromperia o programa

e poderia travar todo o sistema. O arquivo reformatado (agora em formato `.mat`), tem a seguinte estrutura: o banco de dados é dividido em duas células do MATLAB: uma chamada de **data** - que contém as gravações dos 3 eletrodos selecionados do EEG, dos 109 pacientes executando as 8 tarefas selecionadas (totalizando $109 \times 8 = 872$ tarefas) - e outra chamada **header**, que contém as informações do cabeçalho e das anotações.

Para que os dados sejam utilizados em um algoritmo de aprendizagem de máquina, ainda é necessário transformar o banco de dados em formato matricial: uma matriz contendo as características dos dados das gravações, chamada de X (em que cada coluna é uma amostra) e um vetor contendo os rótulos, chamado de y . Para isso, primeiramente aplicamos uma técnica conhecida como Sliding Window (SW) para separar cada amostra - o conjunto de dados do EEG que foram gravados durante cada **evento**, sejam eles imaginar o movimento das mão esquerda, da mão direita, ou manter-se em repouso naquele momento. Para se aplicar o SW, primeiramente define-se os valores de 3 hiper-parâmetros: **time_after_cue**, que é o tempo (em segundos) a ignorar a partir de quando o alvo aparece na tela, devido ao *delay* na reação do paciente; **window_length**, que é o tamanho (ou a duração em segundos) de cada amostra; e **window_padding**, que é a distância (ou tempo, em segundos) que a janela será deslizada em cada iteração.

Em seguida, toma-se o momento de início (**start_point_sec**) e a duração de cada de cada evento (**headertask.annotation.starttime(event)**) - que são informações contidas nas anotações provindas do cabeçalho - e segmenta-se estes eventos de acordo com o tamanho de *window_length*, descartando os sinais até o momento definido em *time_after_cue*, criando-se uma outra célula, que chamamos de **window**, com 4 vetores de atributos: Cz, C3, C4 e t, contendo os dados dos respectivos eletrodos no segmento definido, e o vetor t sendo um vetor de números inteiros em ordem crescente, incrementados de 1 em 1, com seu primeiro elemento igual a 1, e do mesmo tamanho que os outros 3 vetores, representando o eixo do tempo da gravação do EEG. Cada elemento da célula *window* é uma **amostra**, e sua quantidade é diretamente proporcional ao número de eventos, e inversamente proporcional aos 3 hiper-parâmetros definidos para a SW. O processo é iterado até que todo a janela deslizando percorra por todo o evento e, depois, repete-se para as outras tarefas.

3.2 Extração de Características: Redes Complexas aplicadas à Vectorencefalograma Temporal

Após cada iteração do SW, o processo de extração de características é realizado para o segmento daquela iteração. Para efetuar a extração de características, implementamos um método inovador, que nomeamos de Vectorencefalograma Temporal (ou TVEG, do inglês, *Temporal Vectorencephalogram*), baseado em TVCG (GARCIA et al., 2016; GARCIA et

al., 2017).

O Vectorencefalograma é baseado no conceito de Vectorcardiograma (GUYTON; HALL, 2013). Essencialmente, cada ponto do Vectorencefalograma (VEG) traçado nos mostra o ângulo e magnitude do vetor instantâneo médio da polarização do cérebro. Para traçarmos um VEG, basta colocarmos cada canal do EEG em eixos ortogonais. Para que um VEG se torne um TVEG, basta adicionarmos o eixo do tempo no gráfico, que podem ser representados por amostra, ou em segundos. Desta forma, o TVEG formado tem 4 dimensões: uma com os valores do eletrodo Cz, outra com os valores do eletrodo C3, outra com os valores do eletrodo C4, e a última com os valores do tempo (representados em amostras, como discutido na Seção 3.1.2).

Com o TVEG formado, para extrairmos as características relevantes, primeiramente transformamos o TVEG em uma Rede Complexa. O método das Redes Complexas foi desenvolvido a partir de duas áreas: análise estatística e teoria dos grafos. Na área de processamento de sinais biomédicos, as Redes Complexas já foram utilizadas para a descrição e extração de características de Vectorcardiogramas e Vectorcardiogramas Temporais (LUZ, 2012; QUEIROZ et al., 2015; GARCIA et al., 2016; GARCIA et al., 2017). No presente trabalho, propomos aplicar este procedimento na descrição de Vectorencefalogramas Temporais.

Primeiro, consideremos o conjunto de pontos do TVEG, $V = [p_1, p_2, \dots, p_{n_v}]$, no qual cada ponto $p_i = [Cz_i, C3_i, C4_i, t_i]$ é um vetor de 4 dimensões, com as componentes Cz_i , $C3_i$ e $C4_i$ correspondendo ao valor instantâneo dos eletrodos com os respectivos prefixos, e t_i o instante em que estes componentes apresentaram tal valor. Em seguida, determinamos a Distância Euclidiana $d(p_i, p_j)$ entre cada par de pontos contidos em V e atribuímos este valor como o peso da aresta $e_{i,j}$ entre estes dois pontos. A rede corresponde ao grafo $G = (V, E)$, onde $E = \{e_{i,j} \mid i, j \in \{1, \dots, n_v\}\}$. Do passo anterior, obtemos a matriz $W = (w_{ij})$, de ordem n_v , i.e.,

$$w_{ij} = d(p_i, p_j) = \sqrt{(Cz_i - Cz_j)^2 + (C3_i - C3_j)^2 + (C4_i - C4_j)^2 + (t_i - t_j)^2}, \quad (3.1)$$

onde $i, j \in \{1, \dots, n_v\}$. Então, normalizamos os valores da matriz no intervalo $[0, 1]$.

Pode-se notar que todos os vértices estão conectados a todos os outros vértices. Isso classifica a rede como uma Rede Regular. Uma das formas de se obter uma Rede Complexa à partir de uma Rede Regular é aplicando um método iterativo conhecido como Evolução Dinâmica. Em suma, em cada iteração, todas as arestas com peso w_{ij} que forem maior que um limiar T_l são removidas da rede. Deste passo, obtemos uma nova rede complexa, de mundo-pequeno. Em seguida, extraímos cinco características estatísticas da rede gerada, baseada em determinadas propriedades do grafo. O limiar é então incrementado, e o processo é repetido.

Precisamos então, definir as variáveis que determinam como o limiar T_l é calculado: T_0 é o primeiro limiar, T_Q é o último limiar, m é o número finito total de iterações e l é a iteração atual. Então, considerando que T_l é incrementada linearmente, definimos

$$T_l = T_0 + l \frac{(T_Q - T_0)}{(m - 1)}, \quad (3.2)$$

onde T_0 , T_Q e m são determinadas empiricamente, e $0 \leq l \leq m - 1$.

Como mencionado anteriormente nesta seção, extraímos cinco características de cada rede complexa gerada, seguindo o trabalho de Backes, Casanova e Bruno (2009):

- Grau médio de conectividade.
- Grau máximo de conectividade.
- Entropia do grau das juntas.
- Energia do grau das juntas.
- Média do grau das juntas.

As primeiras duas características são obtidas ao se verificar quantos vértices estão conectados à cada vértice (i.e., analisando o grau dos vértices). As últimas três são obtidas ao se calcular a probabilidade de um vértice de determinado grau estar conectado a outro vértice do mesmo grau.

Após este último passo, obtivemos um vetor de características com $m \times 5$ elementos para cada TVEG. Organizamos estes vetores em uma matriz X , onde cada linha corresponde a um vetor de características obtidas do TVEG - que, por sua vez, corresponde a uma janela (ou amostra) do procedimento de SW, descrito na Seção 3.1.2. Construimos também um vetor coluna y , em que cada elemento corresponde ao rótulo - obtido à partir do cabeçalho do Banco de Dados - da amostra de mesmo índice da matriz X .

3.3 Interface Gráfica do Usuário

Enquanto os passos anteriores (de pré-processamento do banco de dados e extração de características) foram implementados em MATLAB, implementamos um protótipo de algoritmo em Python para o treinamento do classificador. Para facilitar o uso do algoritmo, a definição dos hiper-parâmetros do classificador pode ser feita através de uma Interface Gráfica do Usuário (GUI), desenvolvida em Qt, utilizando-se a biblioteca PyQt (SUMMERFIELD, 2008). A GUI desenvolvida é ilustrada na Figura 24. Um botão de “Procurar” (*Browse*) aciona o explorador de arquivos do Sistema Operacional, para que se selecione o banco de dados pré-processado. Em seguida, seleciona-se o tamanho de

cada lote a ser alimentado ao classificador durante o treinamento (*Batch Size*) e o número de iterações de otimização a ser realizado durante o treinamento do classificador (*Steps*). Um outro botão pode ser acionado para que se adicione uma nova camada escondida na arquitetura da Rede Neural (*Add Hidden Layer*), e pode ser ativado múltiplas vezes, adicionando um número arbitrário de camadas. No agrupamento (*Hidden Layer 1*) logo abaixo do botão *Add Hidden Layer*, pode-se selecionar o número de neurônios (*Neurons*) e a probabilidade de se manter as conexões dos neurônios daquela camada (*Dropout*). Para cada vez que o botão *Add Hidden Layer* for acionado, um novo agrupamento aparece para o usuário, para se selecionar a quantidade de neurônios e probabilidade de manter as conexões daquela nova camada. Ao se clicar no botão *Begin Learning*, o treinamento do classificador é iniciado e, como se pode ver na Figura 25 o botão *Begin Learning* fica indisponível até o término do treinamento, e todo o processo é registrado na aba *Log*. Pode-se também limpar os registros da aba *Log* clicando-se no botão *Clear Log* e salvar o conteúdo da aba de registros em formato *.txt* clicando-se no botão *Save Log*, que abre o explorador de arquivos do sistema para se selecionar o nome do arquivo e o diretório de destino. Por último, como visto na Figura 26, os resultados também são apresentados na aba de registros.

Para que a interface não travasse quando se iniciava o processo de treinamento, foi necessário programar o algoritmo de treinamento como uma *Thread*, de tal forma que a *Thread* da GUI não fosse a mesma *Thread* do classificador.

3.4 Treinamento do Classificador

Utilizamos Redes Neurais Artificiais Profundas (DNN) como o classificador para todos os testes, como apresentado na Seção 2.3.1. O primeiro passo foi separar a matriz X e o vetor y em três subconjuntos, cada: os conjuntos de treinamento (X_{train} e y_{train}), os conjuntos de validação (X_{valid} e y_{valid}) e os conjuntos de teste (X_{test} e y_{test}). Estes conjuntos devem tomar elementos aleatórios, tomando-se o cuidado para que os índices dos rótulos continuem os mesmos das amostras. Em seguida, para resolver o problema de desbalanceamento do banco de dados (ou seja, ter mais amostras de dados de um rótulo do que dos outros), removemos os dados excedentes. Isto é, primeiro verificamos quantas amostras tínhamos do rótulo com menos amostras, e retiramos aleatoriamente amostras dos outros dois rótulos até que para todo rótulo, tivéssemos o mesmo número de amostras nos conjuntos de treinamento e de validação.

Para que o classificador implementado consiga diferenciar os rótulos, utilizamos um método chamado de One-Hot-Encoding. Este método consiste em codificar os rótulos em diferentes vetores unitários ortogonais. Em outras palavras, o rótulo T0 se torna $[1,0,0]$, o rótulo T1 se torna $[0,1,0]$ e o rótulo T2 se torna $[0,0,1]$.

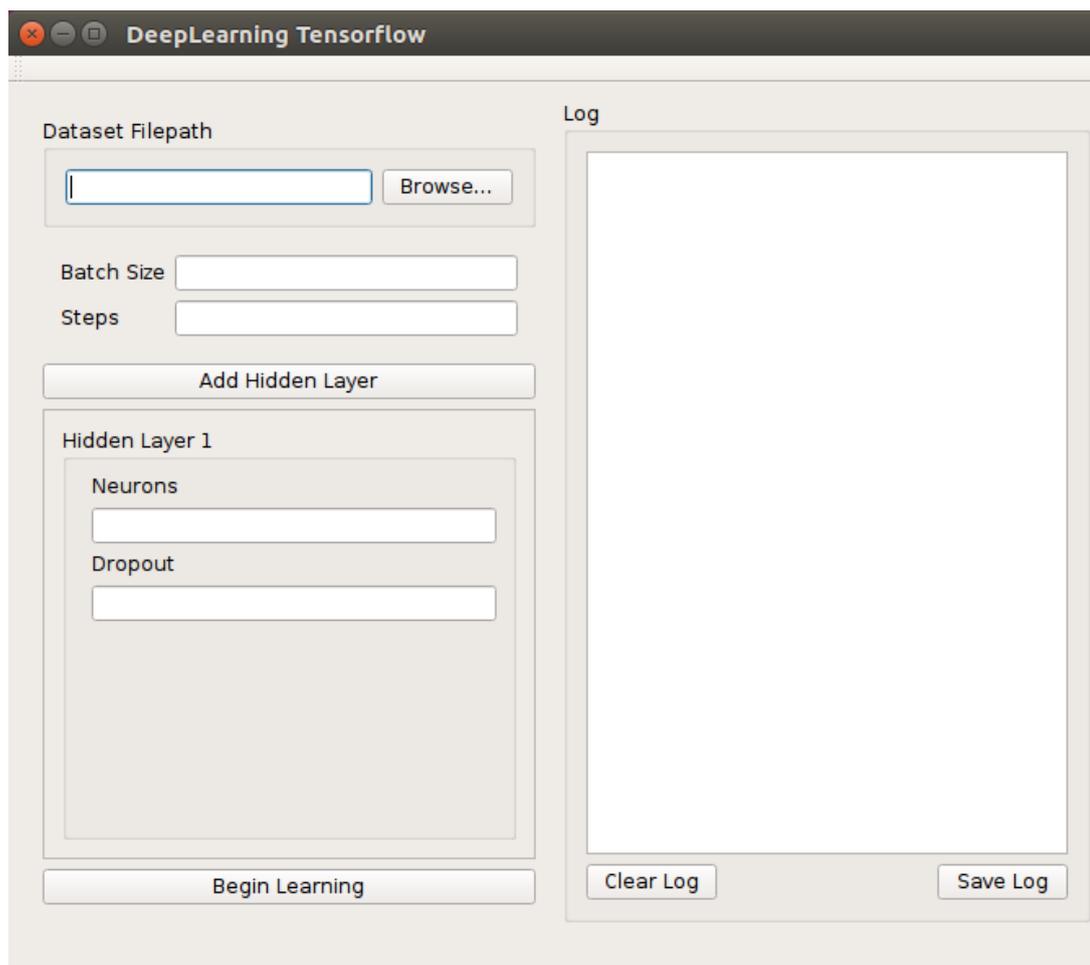


Figura 24: Interface Gráfica do Usuário do treinamento do classificador.

Como mencionado na Seção 2.4, utilizamos a biblioteca TensorFlow para o treinamento do classificador. Para se utilizar a biblioteca, primeiro define-se um grafo, que conterá as informações de todas as operações, variáveis, constantes e entradas do algoritmo de treinamento. Aplicamos o método Dropout (SRIVASTAVA et al., 2014) (Seção 2.3.1) de regularização, em todas as camadas da DNN, com a probabilidade definida pelo usuário na GUI. A função de ativação utilizada entre cada camada da rede foi a Unidade de Retificação Linear (ReLU) (LECUN; BENGIO; HINTON, 2015). Para se calcular o erro das iterações de *back-propagation* (ou seja, a função objetivo a ser minimizada), calculamos a média do Softmax da Entropia Cruzada entre os resultados obtidos e os desejados (DUNNE; CAMPBELL, 1997). Finalmente, como método de otimização, utilizamos o *Adam Optimizer* (KINGMA; BA, 2015).

As definições do grafo não são computadas na hora de sua declaração, mas sim quando são necessárias para se computar algum resultado necessário. Por exemplo, as funções relativas ao treinamento do classificador só serão computadas quando se desejar a acurácia de treinamento do classificador, e as funções relativas à validação só serão computadas quando se desejar a acurácia de validação. Após a definição do grafo do Ten-

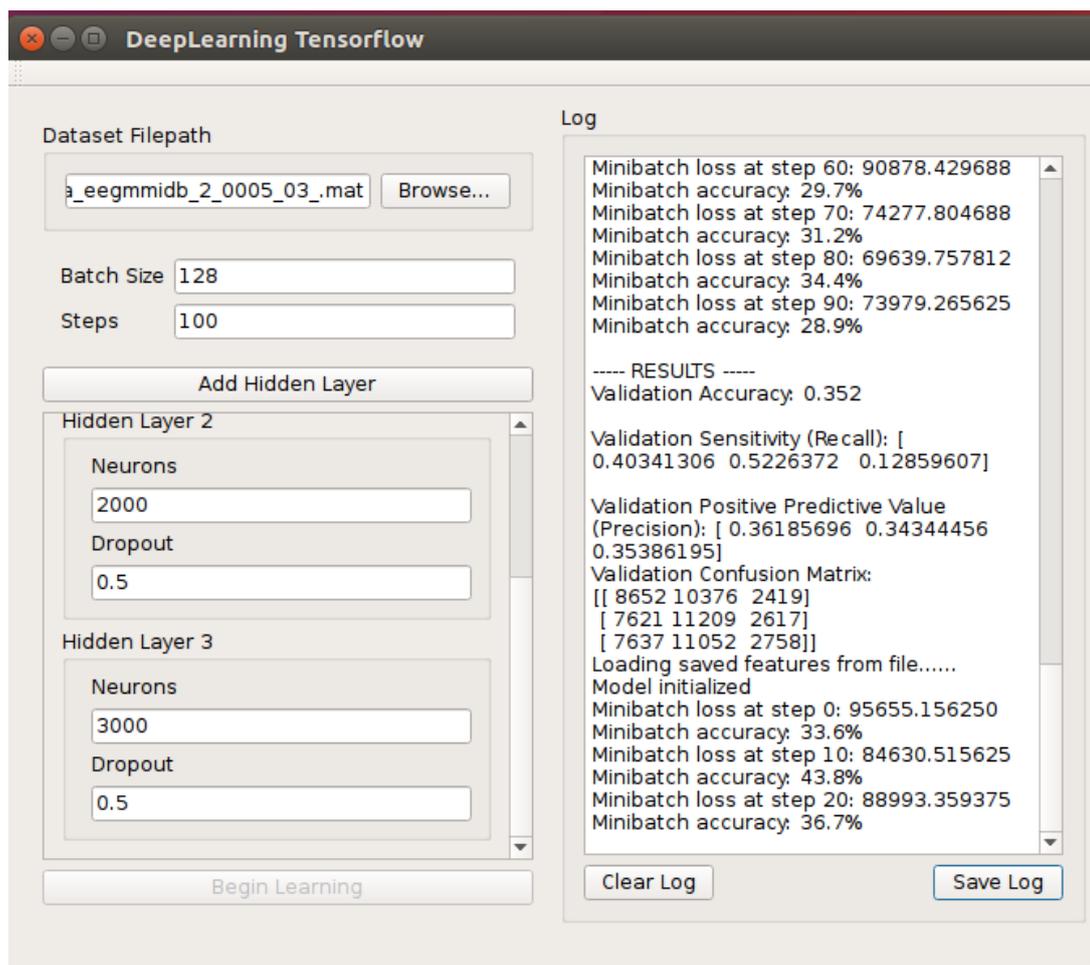


Figura 25: GUI em funcionamento.

sorFlow, é iniciada uma sessão do TensorFlow utilizando o grafo definido como parâmetro. Efetuamos o treinamento do classificador em lotes, ou seja, para alimentar o algoritmo de treinamento, selecionamos uma quantidade (*Batch Size*, definida pelo usuário na GUI) de elementos aleatórios a serem utilizados para aprendizagem em cada passo - o que aumenta a performance computacional e diminui a quantidade de memória necessária para o algoritmo. Por último, calculamos a acurácia, a sensibilidade e a precisão do algoritmo no conjunto de validação, (e de teste, quando necessário), e traçamos a Matriz de Confusão (Ver Seção 4.1).

Os códigos-fonte implementados estão disponíveis em <https://github.com/godofred/BCI-UFOP>.

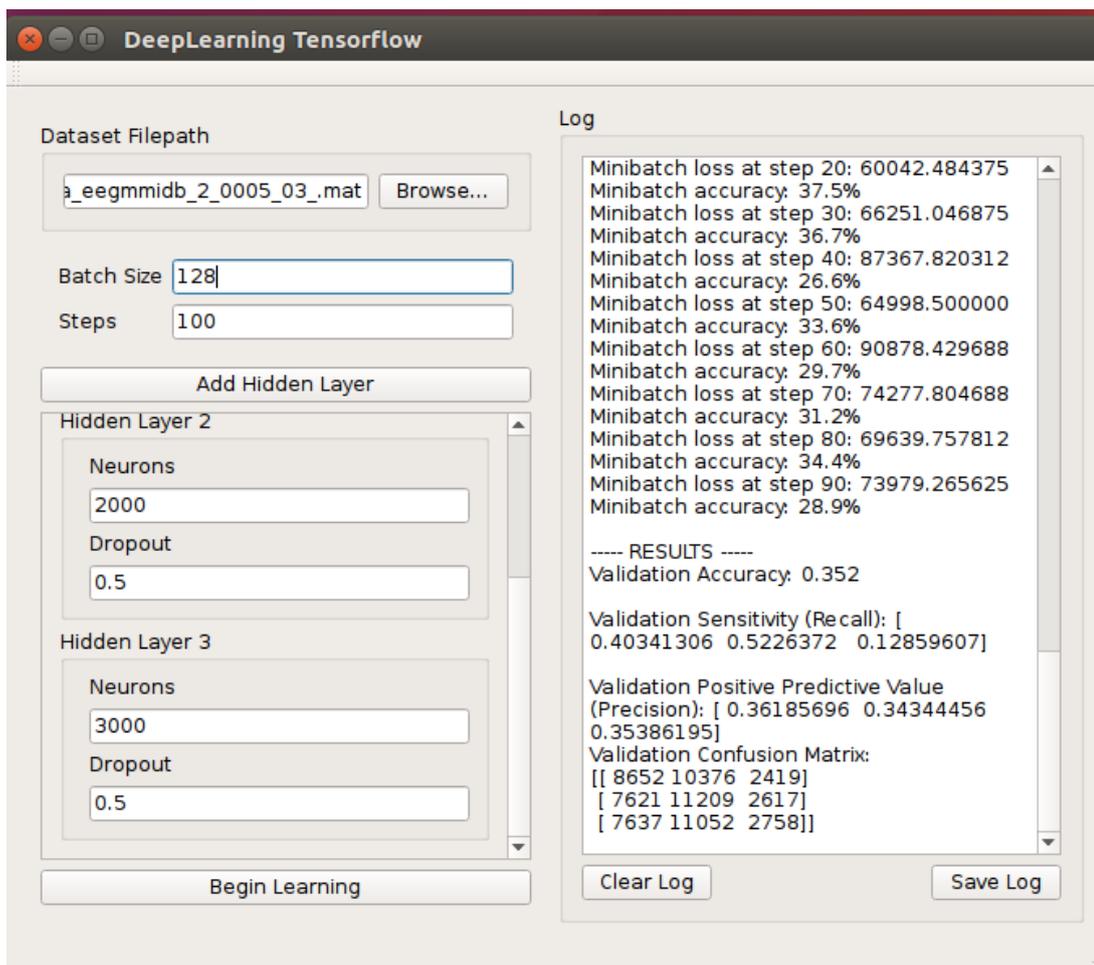


Figura 26: Resultados apresentados na GUI.

4 Análise do Algoritmo

4.1 Métricas de Interesse

Como discutido na Seção 2.3, é necessário avaliar a eficácia do classificador através do conjunto de validação. Para efetuar esta avaliação, utilizamos três métricas quantitativas:

- **Acurácia:** Mede a proporção de classificações corretas.
- **Sensitividade** (ou *Recall*): Mede a probabilidade do método classificar corretamente um resultado positivo. Um teste pode fraudar obtendo 100% de sensibilidade ao classificar todos os resultados como positivos.
- **Precisão** (ou *Positive Predictive Value*): Mede qual a probabilidade dos resultados classificados como positivos serem realmente positivos. Um teste pode fraudar maximizando a precisão classificando como positivos só aqueles resultados que tiver mais certeza do resultado.

Pode-se observar que analisando somente a acurácia, não se pode garantir de que o algoritmo irá classificar corretamente com a mesma probabilidade, já que o conjunto de treinamento pode ser desbalanceado. Pode-se observar também que ao analisar em conjunto a sensibilidade e a precisão, o teste não poderá “trapacear”, já que aumentar propositalmente a sensibilidade irá diminuir a precisão, e vice-versa.

No caso do presente trabalho, temos 3 rótulos - descansando, ou T0; imaginando a movimentação da mão direita, ou T1; e imaginando a movimentação da mão esquerda, ou T2, como apresentando na Seção 3.1.1. Desta forma, os rótulos não são somente valores positivos e negativos. Para se calcular a precisão e sensibilidade, precisaremos analisar a capacidade do algoritmo de classificar cada classe, ou seja, teremos um valor de precisão e um de sensibilidade para cada rótulo, totalizando 6 métricas, além da acurácia. Para se calcular estes valores, para cada classe, considera-se como valor “positivo” os referentes àquela classe, e “negativo” como de qualquer outra classe.

Apresentamos também uma Matriz de Confusão representativa. Uma Matriz de Confusão é uma matriz quadrada C de $n \times n$ elementos, onde n é o número de classes (ou rótulos). O elemento $C_{i,j}$ é igual ao número de amostras que eram da classe i , e que foram classificadas como sendo da classe j .

4.2 Resultados

Durante o desenvolvimento do trabalho, precisamos definir vários hiper-parâmetros no nosso algoritmo - ou seja, valores definidos empiricamente que afetam direta ou indiretamente no resultado final do classificador. Utilizamos 3 parâmetros para a técnica de Sliding Window, para a segmentação do banco de dados: *time_after_cue*, *window_padding* e *window_length*; 3 parâmetros da Evolução Dinâmica para extração de características das Redes Complexas provindas dos TVEG's: o limiar inicial T_0 , o limiar final T_Q e o número de iterações da Evolução Dinâmica m ; os parâmetros das Redes Neurais Artificiais Profundas: o número de camadas escondidas l ou *hidden_layers*, o número de neurônios em cada camada e a probabilidade de se manter a conexão para o método Dropout em cada camada; e os parâmetros do algoritmo de otimização do treinamento do classificador: o tamanho do lote *batch_size* e o número de iterações de otimização *num_steps*.

Os testes realizados consistiram em modificar manualmente os hiper-parâmetros citados, com o objetivo de maximizar as métricas de interesse no conjunto de validação. Alguns destes resultados são apresentados na Tabela 3. Algumas observações relevantes desta tabela: Os Testes 2 e 3 possuem exatamente os mesmos valores de parâmetros, e atingem resultados um pouco diferentes. Isto se deve à inicialização aleatória dos pesos da Rede Neural; Os Testes 8, 9, 13, 14 e 15 demonstram que o algoritmo passou a classificar quaisquer amostras como a mesma classe; O Teste 10 demonstrou praticamente o mesmo comportamento que os Testes citados, mas classificou corretamente uma amostra como da classe T2, obtendo 100% de precisão nesta classe; Os Testes 13 e 14 também possuem os mesmos valores de hiper-parâmetros, mas o Teste 13 classifica todas as amostras como sendo T1 e o Teste 14 classifica todas as amostras como sendo T0. A Tabela 2 mostra a Matriz de Confusão do conjunto de validação do Teste 8.

Tabela 2: Matriz de Confusão do Teste 8

	t0	t1	t2
T0	5965	0	0
T1	5965	0	0
T2	5965	0	0

Tabela 3: Resultados dos testes com o algoritmo implementado, variando-se os parâmetros em destaque

Teste	m	T0	TQ	Window Length	Time After Cue	Window Padding	Batch Size	Steps	Hidden Layers	Neurons	Dropout	T0 Se.	T1 Se.	T2 Se.	T0 Pre.	T1 Pre.	T2 Pre.	Acc.		
1	8	0.1	0.6	0.5	1.5	0.4	128	1000	1	[100]	[0.5]	0.42	0.29	0.34	0.36	0.34	0.33	0.35		
2	8	0.1	0.6	0.5	1.5	0.4	128	1000	1	[1000]	[0.5]	0.54	0.27	0.3	0.38	0.36	0.36	0.37		
3	8	0.1	0.6	0.5	1.5	0.4	128	1000	1	[1000]	[0.5]	0.50	0.40	0.23	0.39	0.36	0.36	0.38		
4	8	0.1	0.6	0.5	1.5	0.4	128	1000000	1	[1000]	[0.5]	0.46	0.36	0.33	0.41	0.47	0.36	0.38		
5	8	0.1	0.6	0.5	1.5	0.4	128	1000	2	[1000 1000]	[0.5 0.5]	0.46	0.30	0.29	0.36	0.34	0.34	0.35		
6	8	0.1	0.6	0.5	1.5	0.4	1024	1000	2	[1000 1000]	[0.5 0.5]	0.43	0.41	0.21	0.36	0.34	0.34	0.35		
7	8	0.1	0.6	0.5	1.5	0.4	1024	10000	2	[1000 1000]	[0.5 0.5]	0.1	0.02	0.91	0.48	0.47	0.33	0.35		
8	8	0.1	0.6	0.5	1.5	0.4	128	10000	3	[1000 1000 1000]	[0.5 0.5 0.5]	1	0	0	0.33	0	0	0.33		
9	8	0.05	0.3	0.5	1.5	0.4	128	10000	3	[1000 1000 1000]	[0.5 0.5 0.5]	1	0	0	0.33	0	0	0	0.33	
10	16	0.05	0.3	0.5	1.5	0.4	128	10000	3	[1000 1000 1000]	[0.5 0.5 0.5]	1	0	0	0.33	0	1	0	0.33	
11	16	0.05	0.3	0.5	1.5	0.4	128	10000	3	[1000 1000 1000]	[0.05 0.05 0.05]	0.27	0.17	0.47	0.34	0.33	0.34	0.34		
12	16	0.05	0.3	0.5	1.5	0.4	128	10000	3	[3000 1000 5000]	[0.05 0.05 0.05]	0.98	0.01	0.01	0.33	0.33	0.34	0.34		
13	16	0.05	0.3	0.5	1.5	0.4	128	10000	8	[100 50 20 50 100 50 20 5000]	[0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5]	0	1	0	0	0	0.33	0	0.33	
14	16	0.05	0.3	0.5	1.5	0.4	128	10000	8	[100 50 20 50 100 50 20 5000]	[0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5]	1	0	0	0.33	0	0	0	0	0.33
15	32	0.25	0.75	0.5	0.5	0.1	128	10000	8	[100 50 20 50 100 50 20 5000]	[0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5]	0	0	1	0	0	0	0.33	0.33	
16	8	0.1	0.6	0.5	1.5	0.4	128	1000000	1	[10000]	[0.5]	0.39	0.38	0.35	0.41	0.36	0.36	0.37		

5 Conclusão

Todos os resultados obtidos demonstram que o classificador falhou em classificar corretamente as amostras dos banco de dados utilizado, apresentando previsões aleatórias. É importante notar que o presente trabalho não comprovou a ineficiência dos métodos utilizados, e sim que deve-se explorar mais a utilização do método proposto.

Ao todo, considerando uma arquitetura de Rede Neural com apenas uma camada escondida no modelo proposto, temos 10 hiper-parâmetros não-lineares a serem otimizados de forma combinatória - ou seja, pode-se encontrar um ótimo local ao se alterar somente um dos hiper-parâmetros, mas é possível que para se encontrar o ótimo global, este hiper-parâmetro tenha um outro valor. O maior desafio deste projeto é encontrar os valores ótimos de cada parâmetro, e que estes valores não prejudiquem as aplicações práticas do algoritmo. Por exemplo, um valor alto para m aumentaria o número de características extraídas do TVEG, o que provavelmente aumentaria a eficácia do classificador, mas ao custo de mais recursos computacionais, o que poderia inviabilizar o uso em “tempo-real” ou impedindo a portabilidade do sistema.

Além dos parâmetros citados, poderia-se considerar também mais alguns parâmetros, tais como: quantos e quais eletrodos a serem utilizados, quais experimentos deveriam ser utilizados, a função de ativação da DNN (sigmoideal, softmax...), métodos e bandas de filtragem digital (IIR, FIR; ritmos beta e mu, filtro rejeita-banda de 50/60Hz, filtro passa-baixa para remoção de artefatos de movimento, filtros espaciais de EEG...), técnicas mais sofisticadas de balanceamento do banco de dados (SMOTE, peso de classificação...), o método de otimização do treinamento do classificador (Gradiente Estocástico, AdaGrad...) e até mesmo o próprio classificador (SVM, CNN...). Todas estas variações são propostas plausíveis de pesquisa para se aumentar a eficácia dos algoritmos de Interface Cérebro-Máquina por Imageria Motora. Desta forma, ainda há muito espaço para a utilização de TVEG com Redes Complexas para a classificação de sinais de EEG de Imageria Motora.

Propomos também a utilização de métodos meta-heurísticos para a otimização combinatória dos hiper-parâmetros do algoritmo, tais como Algoritmos Genéticos, Busca Dispersa, Colônia de Formigas ou Enxame de Partículas.

Por conseguinte, o presente trabalho apresentou uma revisão teórica extensa, discutindo os conceitos necessários para a compreensão do funcionamento de uma Interface Cérebro-Máquina, e testou, de forma preliminar, um método de classificação de sinais de EEG não apresentado na literatura para o objetivo apresentado.

Referências

ABADI, M.; BARHAM, P.; CHEN, J.; CHEN, Z.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; IRVING, G.; ISARD, M.; KUDLUR, M.; LEVENBERG, J.; MONGA, R.; MOORE, S.; MURRAY, D. G.; STEINER, B.; TUCKER, P.; VASUDEVAN, V.; WARDEN, P.; WICKE, M.; YU, Y.; ZHENG, X.; BRAIN, G. TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning. In: **12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)**. [S.l.: s.n.], 2016. p. 265–284. ISBN 978-1-931971-33-1. ISSN 0270-6474. Citado na página 37.

AMIRI, S.; RABBI, A.; AZINFAR, L.; FAZEL-REZAI, R. A Review of P300, SSVEP, and Hybrid P300/SSVEP Brain- Computer Interface Systems. **Brain-Computer Interface Systems - Recent Progress and Future Prospects**, 2013. v. 2013, p. 1–8, 2013. ISSN 1687-5893. Citado 4 vezes nas páginas 14, 15, 31 e 32.

ANG, K. K.; GUAN, C.; WANG, C.; PHUA, K. S.; TAN, A. H. G.; CHIN, Z. Y. Calibrating EEG-based motor imagery brain-computer interface from passive movement. **Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS**, 2011. p. 4199–4202, 2011. ISSN 1557170X. Citado na página 32.

BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. A complex network-based approach for boundary shape analysis. **Pattern Recognition**, 2009. v. 42, n. 1, p. 54–67, 2009. ISSN 00313203. Citado na página 43.

BLACK, M. J.; BIENENSTOCK, E.; DONOGHUE, J. P.; SERRUYA, M.; WU, W.; GAO, Y. Connecting brains with machines: the neural control of 2D cursor movement. **International IEEE/EMBS Conference on Neural Engineering, NER**, 2003. v. 2003-January, p. 580–583, 2003. ISSN 19483554. Citado na página 11.

CARLSON, T.; Del R. Millan, J. Brain-controlled wheelchairs: A robotic architecture. **IEEE Robotics and Automation Magazine**, 2013. v. 20, n. 1, p. 65–73, 2013. ISSN 10709932. Citado na página 11.

CHEN, L. F.; LIAO, H. Y. M.; KO, M. T.; LIN, J. C.; YU, G. J. New LDA-based face recognition system which can solve the small sample size problem. **Pattern Recognition**, 2000. v. 33, n. 10, p. 1713–1726, 2000. ISSN 00313203. Citado na página 32.

DEECKE, L.; WEINBERG, H.; BRICKETT, P. Magnetic fields of the human brain accompanying voluntary movement: Bereitschaftsmagnetfeld. **Exp Brain Res**, 1982. v. 48, n. 1, p. 144–148, 1982. Citado na página 40.

DUNNE, R.; CAMPBELL, N. On the pairing of the Softmax activation and cross-entropy penalty functions and the derivation of the Softmax activation function. **Proc. 8th Aust. Conf. on the Neural Networks**, 1997. p. 1–5, 1997. Citado na página 45.

GARCIA, G.; MOREIRA, G.; LUZ, E.; MENOTTI, D. Improving automatic cardiac arrhythmia classification: Joining temporal-VCG, complex networks and SVM classifier. In: **Proceedings of the International Joint Conference on Neural Networks**. [S.l.: s.n.], 2016. v. 2016-October, p. 3896–3900. ISBN 9781509006199. Citado 2 vezes nas páginas 41 e 42.

GARCIA, G.; MOREIRA, G.; MENOTTI, D.; LUZ, E. Inter-Patient ECG Heartbeat Classification with Temporal VCG Optimized by PSO. **Scientific Reports**, 2017. v. 7, n. 1, 2017. ISSN 20452322. Citado 3 vezes nas páginas 32, 41 e 42.

GOLDBERGER, A. L.; AMARAL, L. A. N.; GLASS, L.; HAUSDORFF, J. M.; IVANOV, P. C.; MARK, R. G.; MIETUS, J. E.; MOODY, G. B.; PENG, C.-K.; STANLEY, H. E. PhysioBank, PhysioToolkit, and PhysioNet : Components of a New Research Resource for Complex Physiologic Signals. **Circulation**, 2000. v. 101, n. 23, p. e215–e220, 2000. ISSN 0009-7322. Disponível em: <<http://circ.ahajournals.org/cgi/doi/10.1161/01.CIR.101.23.e215>>. Citado na página 38.

GORDON, B.; LEDOUX, K. **Handbook of the Neuroscience of Language**. [S.l.: s.n.], 2008. 105–114 p. ISBN 9780080453521. Citado na página 14.

GUGER, C.; EDLINGER, G.; HARKAM, W.; NIEDERMAYER, I.; PFURTSCHELLER, G. How many people are able to operate an EEG-based brain-computer interface (BCI)? *IEEE Trans Neural Syst Rehabil Eng* 2003; 11: 145-147. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, 2003. v. 11, n. 2, p. 145–147, 2003. Disponível em: <http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=1214>. Citado na página 32.

GUYTON, A. C.; HALL, J. E. **Textbook of Medical Physiology**. [S.l.: s.n.], 2013. Citado 4 vezes nas páginas 16, 19, 24 e 42.

HE, B.; BAXTER, B.; EDELMAN, B. J.; CLINE, C. C.; YE, W. W. Noninvasive brain-computer interfaces based on sensorimotor rhythms. **Proceedings of the IEEE**, 2015. v. 103, n. 6, p. 907–925, 2015. ISSN 00189219. Citado na página 32.

HO, K.-l.; HSU, Y.-y. Short Term Load Forecasting Using a Multilayer Neural Network. **IEEE Transactions on Power Systems**, 1992. v. 7, n. 1, p. 141–149, 1992. ISSN 08858950. Citado na página 32.

Jiralerspong, Trongmun, Liu, Chao, Ishikawa, J. Identification of Three Mental States Using a Motor Imagery Based Brain Machine Interface. **IEEE Symposium on computational Intelligence in Brain Computer Interfaces (CIBCI)**, 2014. p. 2081–2089, 2014. Citado na página 32.

KAPER, M.; MEINICKE, P.; GROSSEKATHOEFER, U.; LINGNER, T.; RITTER, H. BCI competition 2003 - Data set Iib: Support vector machines for the P300 speller paradigm. **IEEE Transactions on Biomedical Engineering**, 2004. v. 51, n. 6, p. 1073–1076, 2004. ISSN 00189294. Citado na página 14.

KELLIS, S.; SORENSEN, L.; DARVAS, F.; SAYRES, C.; O'NEILL, K.; BROWN, R. B.; HOUSE, P.; OJEMANN, J.; GREGER, B. Multi-scale analysis of neural activity in humans: Implications for micro-scale electrocorticography. **Clinical Neurophysiology**, 2016. v. 127, n. 1, p. 591–601, 2016. ISSN 18728952. Citado na página 13.

- KHAN, M. J.; HONG, K.-S. Passive BCI based on drowsiness detection: an fNIRS study. **Biomedical optics express**, 2015. v. 6, n. 10, 2015. ISSN 2156-7085. Citado na página 11.
- KIM, Y.; RYU, J.; KIM, K. K.; TOOK, C. C.; MANDIC, D. P.; PARK, C. Motor Imagery Classification Using Mu and Beta Rhythms of EEG with Strong Uncorrelating Transform Based Complex Common Spatial Patterns. **Computational Intelligence and Neuroscience**, 2016. v. 2016, 2016. Citado na página 32.
- KINGMA, D. P.; BA, J. L. Adam: a Method for Stochastic Optimization. **International Conference on Learning Representations 2015**, 2015. p. 1–15, 2015. ISSN 09252312. Citado na página 45.
- KRIZHEVSKY, A.; SUTSKEVER, I.; Geoffrey E., H. ImageNet Classification with Deep Convolutional Neural Networks. **Advances in Neural Information Processing Systems 25 (NIPS2012)**, 2012. p. 1–9, 2012. ISSN 10495258. Citado na página 32.
- LAFLEUR, K.; CASSADY, K.; DOUD, A.; SHADES, K.; ROGIN, E.; HE, B. Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface. **Journal of Neural Engineering**, 2013. v. 10, n. 4, p. 046003, 2013. ISSN 1741-2560. Citado na página 11.
- LANCE, B. J.; KERICK, S. E.; RIES, A. J.; OIE, K. S.; MCDOWELL, K. Brain-computer interface technologies in the coming decades. **Proceedings of the IEEE**, 2012. v. 100, n. SPL CONTENT, p. 1585–1599, 2012. ISSN 00189219. Citado na página 32.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, 2015. v. 521, n. 7553, p. 436–444, 2015. ISSN 0028-0836. Citado na página 45.
- LÉCUYER, A.; LOTTE, F.; REILLY, R. B.; LEEB, R.; HIROSE, M.; SLATER, M. Brain-computer interfaces, virtual reality, and videogames. **Computer**, 2008. v. 41, n. 10, p. 66–72, 2008. ISSN 00189162. Citado na página 12.
- LIN, Y. P.; WANG, C. H.; JUNG, T. P.; WU, T. L.; JENG, S. K.; DUANN, J. R.; CHEN, J. H. EEG-based Emotion Recognition. **IEEE Transactions on Biomedical Engineering**, 2010. v. 57, n. 7, p. 1798–1806, 2010. ISSN 00189294. Citado na página 11.
- LUZ, E. **CLASSIFICAÇÃO AUTOMÁTICA DE ARRITMIAS: UM NOVO MÉTODO USANDO CLASSIFICAÇÃO HIERÁRQUICA**. Dissertação (Mestrado) — UFOP, 2 2012. Citado na página 42.
- MAKEIG, S.; KOTHE, C.; MULLEN, T.; BIGDELY-SHAMLO, N.; ZHANG, Z.; KREUTZ-DELGADO, K. Evolving signal processing for brain-computer interfaces. **Proceedings of the IEEE**, 2012. v. 100, n. SPL CONTENT, p. 1567–1584, 2012. ISSN 00189219. Citado na página 32.
- MAYE, A.; ZHANG, D.; WANG, Y.; GAO, S.; ENGEL, A. K. Multimodal brain-computer interfaces. **Tsinghua Science and Technology**, 2011. v. 16, n. 2, p. 133–139, 2011. ISSN 10070214. Citado na página 32.
- MÜLLER-PUTZ, G. R.; PFURTSCHELLER, G. Control of an electrical prosthesis with an SSVEP-based BCI. **IEEE Transactions on Biomedical Engineering**, 2008. v. 55, n. 1, p. 361–364, 2008. ISSN 00189294. Citado na página 11.

NAKASATP, N.; LEVESQUE, M. F.; BARTH, D. S.; BAUMGARTNER, C.; ROGERS, R. L.; SUTHERLING, W. W. Comparisons of MEG, EEG, and ECoG source localization in neocortical partial epilepsy in humans. **Electroencephalography and Clinical Neurophysiology**, 1994. v. 91, n. 3, p. 171–178, 1994. ISSN 00134694. Citado na página 13.

NEUPER, C.; PFURTSCHELLER, G. Evidence for distinct beta resonance frequencies in human EEG related to specific sensorimotor cortical areas. **Clinical Neurophysiology**, 2001. v. 112, n. 11, p. 2084–2097, 2001. ISSN 13882457. Citado na página 40.

OWENS, J. D.; LUEBKE, D.; GOVINDARAJU, N.; HARRIS, M.; KRÜGER, J.; LEFOHN, A. E.; PURCELL, T. J. **A survey of general-purpose computation on graphics hardware**. 2007. 80–113 p. Citado na página 36.

PALANIAPPAN, R. Electroencephalogram-based Brain–Computer Interface: An Introduction. **Springer-Verlag**, 2014. p. 321, 2014. Disponível em: <<http://link.springer.com/10.1007/978-1-4471-6584-2>>. Citado na página 32.

PAN, J.; XIE, Q.; HE, Y.; WANG, F.; DI, H.; LAUREYS, S.; YU, R.; LI, Y. Detecting awareness in patients with disorders of consciousness using a hybrid brain-computer interface. **Journal of neural engineering**, 2014. IOP Publishing, v. 11, n. 5, p. 056007, 2014. ISSN 1741-2552. Disponível em: <<http://orbi.ulg.ac.be/handle/2268/185857>>. Citado na página 11.

PFURTSCHELLER, G.; MÜLLER-PUTZ, G. R.; PFURTSCHELLER, J.; RUPP, R. EEG-based asynchronous BCI controls functional electrical stimulation in a tetraplegic patient. **Eurasip Journal on Applied Signal Processing**, 2005. v. 2005, n. 19, p. 3152–3155, 2005. ISSN 11108657. Citado na página 11.

QUEIROZ, V.; LUZ, E.; MOREIRA, G.; GUARDA, A.; MENOTTI, D. Automatic cardiac arrhythmia detection and classification using vectorcardiograms and complex networks. In: **Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS**. [S.l.: s.n.], 2015. v. 2015-November, p. 5203–5206. ISBN 9781424492718. ISSN 1557170X. Citado 2 vezes nas páginas 40 e 42.

SCHALK, G.; MCFARLAND, D. J.; HINTERBERGER, T.; BIRBAUMER, N.; WOLPAW, J. R. BCI2000: A general-purpose brain-computer interface (BCI) system. **IEEE Transactions on Biomedical Engineering**, 2004. v. 51, n. 6, p. 1034–1043, 2004. ISSN 00189294. Citado na página 38.

SCHERER, R.; MÜLLER, G. R.; NEUPER, C.; GRAIMANN, B.; PFURTSCHELLER, G. An asynchronously controlled EEG-based virtual keyboard: Improvement of the spelling rate. **IEEE Transactions on Biomedical Engineering**, 2004. v. 51, n. 6, p. 979–984, 2004. ISSN 00189294. Citado na página 11.

SLEIGHT, J.; PILLAI, P.; MOHAN, S. Classification of executed and imagined motor movement EEG signals. **Research Report**, 2009. p. 1–10, 2009. Disponível em: <<http://www-personal.umich.edu/~shiwali/images/MohanPillaiSleight.p>>. Citado na página 40.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout. A Simple Way to Prevent Neural Networks from Overfitting. **Journal of Machine Learning Research**, 2014. v. 15, p. 1929–1958, 2014. ISSN 15337928. Citado 2 vezes nas páginas 35 e 45.

STRIGL, D.; KOFLER, K.; PODLIPNIG, S. Performance and Scalability of GPU-Based Convolutional Neural Networks. In: **2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing**. [S.l.: s.n.], 2010. p. 317–324. ISBN 978-1-4244-5672-7. ISSN 1066-6192. Citado na página 36.

SUMMERFIELD, M. **Rapid Gui Programming with Python and Qt: The Definite Guid to PyQt Programming**. [S.l.: s.n.], 2008. NP p. ISSN 1537-6613. ISBN 9780132354189. Citado na página 43.

TEPLAN, M. Fundamentals of EEG measurement. **Measurement Science Review**, 2002. v. 2, n. 2, p. 1–11, 2002. ISSN 15353893. Citado na página 30.

ULLAH, K.; ALI, M.; RIZWAN, M.; IMRAN, M. Low-cost single-channel EEG based communication system for people with lock-in syndrome. **Proceedings of the 14th IEEE International Multitopic Conference 2011, INMIC 2011**, 2011. p. 120–125, 2011. Citado na página 11.

WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. **IEEE Transactions on Evolutionary Computation**, 1997. v. 1, n. 1, p. 67–82, 1997. ISSN 1089778X. Citado na página 33.

ZHANG, X.; XU, G.; XIE, J.; LI, M.; PEI, W.; ZHANG, J. An EEG-driven Lower Limb Rehabilitation Training System for Active and Passive Co-stimulation. In: **Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference**. [S.l.: s.n.], 2015. v. 2015, p. 4582–4585. ISBN 978-1-4244-9270-1. ISSN 1557-170X (Print). Citado na página 11.