



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Engenharia Elétrica e de Computação

Willian Rampazzo

# **Aprendizado Profundo e Interfaces Cérebro-Computador**

Campinas

2019



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Engenharia Elétrica e de Computação

Willian Rampazzo

## **Aprendizado Profundo e Interfaces Cérebro-Computador**

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Engenharia de Computação.

Orientador: Prof. Dr. Romis Ribeiro de Faissol Attux

Este exemplar corresponde à versão final da dissertação defendida pelo aluno Willian Rampazzo, e orientada pelo Prof. Dr. Romis Ribeiro de Faissol Attux

---

Campinas

2019

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Rose Meire da Silva - CRB 8/5974

R147a Rampazzo, Willian, 1983-  
Aprendizado profundo e interfaces cérebro-computador / Willian Rampazzo. – Campinas, SP : [s.n.], 2019.

Orientador: Romis Ribeiro de Faissol Attux.  
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Aprendizado profundo. 2. Interfaces cérebro-computador. 3. Redes neurais (Computação). 4. Potenciais evocados. I. Attux, Romis Ribeiro de Faissol, 1978-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Deep learning and brain-computer interfaces

**Palavras-chave em inglês:**

Computers / neural networks

Brain-computer interfaces

Machine learning

Evoked potentials

**Área de concentração:** Engenharia de Computação

**Titulação:** Mestre em Engenharia Elétrica

**Banca examinadora:**

Romis Ribeiro de Faissol Attux [Orientador]

André Kazuo Takahata

Fernando José Von Zuben

**Data de defesa:** 13-08-2019

**Programa de Pós-Graduação:** Engenharia Elétrica

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0003-1395-5589>

- Currículo Lattes do autor: <http://lattes.cnpq.br/5365232553576872>

## Comissão Julgadora – Dissertação de Mestrado

**Candidato:** Willian Rampazzo RA 095284

**Data de defesa:** 13 de Agosto de 2019

**Título da dissertação:** “Aprendizado Profundo e Interfaces Cérebro-Computador”

Prof. Dr. Romis Ribeiro de Faissol Attux (Presidente, FEEC/UNICAMP)

Prof. Dr. André Kazuo Takahata (CECS/UFABC)

Prof. Dr. Fernando José Von Zuben (FEEC/UNICAMP)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Tese) e na secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

*Dedico esta dissertação à todo mundo.*

# Agradecimentos

Inicialmente, gostaria de agradecer a Deus pela oportunidade de conhecer pessoas fantásticas durante minha jornada no mestrado. Pessoas que fazem o que fazem pela paixão que têm pela área acadêmica e pela educação. Pessoas que deixam de olhar para si e dedicam grande parte de suas vidas na tarefa de fazer com que a educação atinja a todos, sem exceção, da melhor forma e com a melhor qualidade possível.

Agradeço aos meus pais, Estela e Luiz, pela dedicação, pelo suporte e pelo apoio para que eu pudesse ter acesso à uma educação acima da média. Hoje eu entendo que somente pela educação podemos evoluir. Vocês são os melhores pais que eu poderia desejar.

Agradeço à Priscila, por estar sempre ao meu lado, por me apoiar, por me dar suporte e me ajudar a tomar as melhores decisões para minha vida, sempre que precisei, e sempre da forma mais amorosa e mais carinhosa que alguém poderia desejar. Sem você eu não teria chegado até aqui.

Agradeço ao Prof. Romis Attux, por aceitar me orientar e por confiar em mim quando muitos outros já haviam me negado uma oportunidade. Eu sei que não foi fácil, mas a sua perseverança e a sua paixão pelo que faz me fizeram ter você como um dos meus exemplos de pessoa a seguir na vida. Que seu coração seja sempre grande e que você nunca perca as esperanças pois, mesmo sem saber, você transforma a vida das pessoas para melhor.

Agradeço ao Prof. Rafael Ferrari, pelas conversas, pelo suporte e por toda paciência e tempo dedicado a me explicar coisas que não tinham relação com meu mestrado, mas que me fascinam. Sua dedicação e amor pelo que faz deveriam se tornar o modelo básico a ser seguido por quem quer exercer a função de professor.

Agradeço aos amigos do DSPCom, pelas conversas, discussões e reflexões. Vocês foram fundamentais para o meu crescimento pessoal. Conhecer pessoas com as mais variadas experiências de vida me ajudou a perceber como o respeito ao próximo é fundamental para uma vida em sociedade. Muito obrigado por toda ajuda nos momentos que precisei. Obrigado também pelo que pode ser considerado como o motor da ciência, o café, ahhh, que delícia.

Por fim, agradeço a todos os funcionários da Faculdade de Engenharia Elétrica e de Computação por todo auxílio. Agradeço à Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP - processo 2018/04100-3, pelo suporte financeiro e aos cidadãos do estado de São Paulo pela contribuição que permitiu o desenvolvimento de todo esse

projeto de pesquisa. O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código 001.

# Resumo

Uma Interface Cérebro-Computador (BCI, do inglês, *Brain-Computer Interface*) é um sistema artificial de comunicação direta entre o cérebro e um dispositivo externo. Ela opera de forma independente de nervos e músculos periféricos. Sua finalidade é traduzir a intenção do usuário, associada à medição da atividade originada no cérebro, em sinais de controle correspondentes aos da aplicação. Para tal, ela faz uso de técnicas de processamento de sinais e reconhecimento de padrões. Diferentes áreas podem se beneficiar das BCIs, como a área de entretenimento e a área de saúde. Especialmente na área de saúde, estes sistemas podem ter um grande impacto em pessoas com doenças que levam à síndrome do encarceramento, sendo, possivelmente, a única opção para comunicação em tais casos.

As redes neurais profundas têm sido aplicadas a problemas de diferentes áreas, como visão computacional, processamento de linguagem natural, diagnósticos médicos, entre outras, obtendo resultados que, muitas vezes, ultrapassam o estado-da-arte. Uma característica interessante destas redes, em particular as redes convolucionais profundas (DCNNs, do inglês, *Deep Convolutional Neural Networks*), é a capacidade de extração de características de forma automática, dispensando, frequentemente, a engenharia de atributos manual, etapa quase obrigatória para um grande número de técnicas de aprendizado de máquina.

Os resultados promissores obtidos por DCNNs em diferentes áreas indicam que há potencial para empregá-las na tarefa de processamento de sinais cerebrais em sistemas BCI. Este trabalho explora o uso destas redes em uma BCI baseada em potenciais visualmente evocados em estado estacionário (SSVEP, do inglês, *Steady State Visually Evoked Potentials*), investigando, inicialmente, se formatos diferentes de entrada, como o sinal bruto ou uma transformação do sinal, a exemplo, a transformada de Fourier de tempo curto, influenciam no desempenho das DCNNs. Para esta investigação, são propostas novas arquiteturas de DCNNs, avaliadas sobre diferentes tipos de entrada, e seus desempenhos são comparados aos de abordagens usualmente adotadas na etapa de classificação de BCIs. Este trabalho também investiga se é possível utilizar a técnica de transferência de conhecimento para ajustar DCNNs pré-treinadas aos dados de BCIs. Os resultados experimentais indicam que as DCNNs são uma opção a ser aplicada ao processamento de sinais cerebrais em BCIs.

**Palavras-chaves:** Aprendizado profundo; interfaces cérebro-computador; redes neurais artificiais; SSVEP.

# Abstract

A Brain-Computer Interface (BCI) is an artificial system of direct communication between the brain and an external device. It operates independently of peripheral nerves and muscles. Its purpose is to translate the user's intention, associated with the measurement of brain-originated activity, into control signals corresponding to those of the application. To do so, it makes use of signal processing and pattern recognition techniques. Different areas can benefit from BCIs, such as the entertainment area and the healthcare area. Especially in healthcare, these systems can have a significant impact on people with diseases that lead to locked-in syndrome and are possibly the only option for communication in such cases.

Deep neural networks have been applied to problems in different areas, such as computer vision, natural language processing, medical diagnostics, among others, obtaining results that often surpass the state-of-the-art. An exciting feature of these networks, in particular, Deep Convolutional Neural Networks (DCNNs), is the ability to extract features automatically, often without the need for manual attribute engineering, an almost mandatory step for a large number of machine learning techniques.

The promising results obtained by DCNNs in different areas indicate that there is potential to employ them in the task of processing brain signals in BCI systems. In this work, we explore the use of these networks in a Steady-State Visually Evoked Potentials (SSVEP) -based BCI investigating, initially, whether different input formats such as the raw signal or a signal transformation, for example, the short-time Fourier transform, influence the performance of DCNNs. For such, we propose new DCNN architectures and evaluate them in different input formats, comparing their performances to the approaches usually adopted in the BCI classification step. This work also investigates whether it is possible to use the knowledge transfer technique to adjust pre-trained DCNNs to BCI data. Experimental results indicate that DCNNs are an option to apply to brain signal processing in BCIs.

**Keywords:** Deep learning; brain-computer interfaces; artificial neural networks; SSVEP.

# Lista de ilustrações

Figura 1 – Representação das etapas de uma BCI segundo Wolpaw <i>et al.</i> (2002). . . . .	20
Figura 2 – Distribuição dos eletrodos no sistema internacional 10-10 (CHATRIAN <i>et al.</i> , 1985). . . . .	22
Figura 3 – Onda P300. . . . .	24
Figura 4 – Estímulos visuais para uma BCI SSVEP. . . . .	25
Figura 5 – Sinal de EEG proveniente do eletrodo Oz no domínio do tempo e da frequência para um estímulo de 15Hz. . . . .	25
Figura 6 – Modelo não-linear de um neurônio. . . . .	31
Figura 7 – Representação gráfica do modelo perceptron de múltiplas camadas (MLP). . . . .	32
Figura 8 – Representação gráfica da operação de correlação cruzada para uma entrada 2D. . . . .	36
Figura 9 – Representação gráfica da conectividade em redes totalmente conectadas e em redes convolucionais. . . . .	37
Figura 10 – Representação gráfica de um passo do treinamento usando a função custo <i>triplet</i> . . . . .	44
Figura 11 – Representação gráfica da operação da arquitetura DNN usando a função custo <i>triplet</i> apresentada por Schroff <i>et al.</i> (2015). . . . .	45
Figura 12 – Disposição dos eletrodos no escalpo (CARVALHO <i>et al.</i> , 2015b). . . . .	54
Figura 13 – Exemplo da divisão de uma sessão de 12 segundos em janelas de 3 segundos. . . . .	55
Figura 14 – Exemplos de amostras da base de dados de espectrogramas. . . . .	57
Figura 15 – Ilustração da arquitetura DCNN e seus filtros para dados no formato de módulos da STFT ou imagens de espectrogramas. . . . .	59
Figura 16 – Ilustração da arquitetura DCNN e seus filtros para dados no formato de sinais brutos. . . . .	59
Figura 17 – Bloco básico da rede residual, adaptado de He <i>et al.</i> (2016). . . . .	60
Figura 18 – Rede residual com 34 camadas, adaptado de He <i>et al.</i> (2016). . . . .	62
Figura 19 – Proposta de arquitetura DCNN para mapeamento no espaço euclidiano de amostras no formato de imagem de espectrograma nas frequências fundamentais. . . . .	63
Figura 20 – Proposta de arquitetura DCNN para mapeamento no espaço euclidiano de amostras no formato de imagem de espectrograma nas frequências fundamentais mais uma harmônica superior. . . . .	63
Figura 21 – Representação gráfica da divisão dos conjuntos de treinamento, validação e teste. . . . .	65
Figura 22 – Exemplo da soma das amplitudes nas frequências de interesse. . . . .	66

Figura 23 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de espectrogramas sem harmônica superior usando informação dos eletrodos O1, O2, Oz, POz. . . . .	67
Figura 24 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de espectrogramas sem harmônica superior usando informação do eletrodo Oz. . . . .	67
Figura 25 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de módulos da STFT usando informação dos eletrodos O1, O2, Oz, POz. . . . .	68
Figura 26 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de módulos da STFT usando informação do eletrodo Oz. . . . .	69
Figura 27 – Acurácia da DCNN e da SVM linear na base de dados de sinais brutos usando informação dos eletrodos O1, O2, Oz, POz. . . . .	69
Figura 28 – Acurácia da DCNN e da SVM linear na base de dados de sinais brutos usando informação do eletrodo Oz. . . . .	70
Figura 29 – Gráfico das evoluções do erro no conjunto de treinamento (curva inferior) e do erro no conjunto de validação (curva superior) durante o treinamento da DCNN na base de dados de sinais brutos para o sujeito 2 usando informação dos eletrodos O1, O2, Oz, POz. . . . .	72
Figura 30 – Gráfico das evoluções do erro no conjunto de treinamento (curva inferior) e do erro no conjunto de validação (curva superior) durante o treinamento da DCNN na base de dados de sinais brutos para o sujeito 2 usando informação do eletrodo Oz. . . . .	72
Figura 31 – Acurácias dos diferentes arranjos da arquitetura ResNet na base de dados de espectrogramas sem harmônica superior usando informação dos eletrodos O1, O2, Oz, POz. . . . .	75
Figura 32 – Acurácias dos diferentes arranjos da arquitetura ResNet na base de dados de espectrogramas sem harmônica superior usando informação do eletrodo Oz. . . . .	77
Figura 33 – Acurácia da DCNN com função de custo <i>triplet</i> e da SVM linear na base de dados de espectrogramas sem harmônica superior e com uma harmônica superior usando informação dos eletrodos O1, O2, Oz, POz. . . . .	81
Figura 34 – Acurácia da DCNN com função de custo <i>triplet</i> e da SVM linear na base de dados de espectrogramas sem a primeira harmônica e com a primeira harmônica usando informação do eletrodo Oz. . . . .	83

# Lista de tabelas

Tabela 1	–	Variações da arquitetura ResNet. Entre chaves estão os blocos básicos com o tamanho dos filtros e a quantidade de filtros da convolução, seguido pelo número de blocos básicos. As subamostragens das entradas são feitas nas camadas conv2_1, conv3_1, conv4_1 e conv5_1. FLOPs, do inglês, <i>Floating Point Operations per second</i> . Adaptado de He <i>et al.</i> (2016). . . . .	61
Tabela 2	–	Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de espectrogramas sem harmônica superior usando informação dos eletrodos O1, O2, Oz, POz. . . . .	97
Tabela 3	–	Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de espectrogramas sem harmônica superior usando informação do eletrodo Oz. . . . .	97
Tabela 4	–	Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados dos módulos da STFT usando informação dos eletrodos O1, O2, Oz, POz. . . . .	97
Tabela 5	–	Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados dos módulos da STFT usando informação do eletrodo Oz. . . . .	98
Tabela 6	–	Acurácia da DCNN e da SVM linear na base de dados de sinais brutos usando informação dos eletrodos O1, O2, Oz, POz. . . . .	98
Tabela 7	–	Acurácia da DCNN e da SVM linear na base de dados de sinais brutos usando informação do eletrodo Oz. . . . .	98
Tabela 8	–	Acurácias dos diferentes arranjos da arquitetura ResNet na base de dados de espectrogramas sem harmônica superior usando informação dos eletrodos O1, O2, Oz, POz. . . . .	98
Tabela 9	–	Acurácias dos diferentes arranjos da arquitetura ResNet na base de dados de espectrogramas sem harmônica superior usando informação do eletrodo Oz. . . . .	99
Tabela 10	–	Acurácia da DCNN com função de custo <i>triplet</i> e da SVM linear na base de dados de espectrogramas sem harmônica superior e com uma harmônica superior usando informação dos eletrodos O1, O2, Oz, POz. . . . .	99
Tabela 11	–	Acurácia da DCNN com função de custo <i>triplet</i> e da SVM linear na base de dados de espectrogramas sem harmônica superior e com uma harmônica superior usando informação do eletrodo Oz. . . . .	99

# Sumário

<b>1</b>	<b>Introdução</b>	<b>15</b>
1.1	Motivação	16
1.2	Objetivos	16
1.3	Organização da Dissertação	17
1.4	Produções Acadêmicas	18
<b>2</b>	<b>Interfaces Cérebro-Computador</b>	<b>19</b>
2.1	Definição de Interfaces Cérebro-Computador	19
2.2	Aquisição de Sinais Cerebrais	20
2.2.1	Aquisição de Sinais Cerebrais via EEG	21
2.3	Paradigmas de BCI	22
2.3.1	Sistemas BCI Baseados em Imagética	23
2.3.2	Sistemas BCI Baseados em Atenção Seletiva	23
2.4	Processamento de Sinais Cerebrais	26
2.4.1	Pré-Processamento	26
2.4.2	Extração de Características	26
2.4.3	Seleção de Características	27
2.4.4	Classificação	28
2.5	Controle	29
<b>3</b>	<b>Aprendizado Profundo</b>	<b>30</b>
3.1	Redes Neurais Artificiais	30
3.2	Aprendizado Profundo	33
3.3	Redes Convolucionais	34
3.3.1	A Operação de Convolução	35
3.3.2	Conectividade Esparsa, Campos Receptivos Locais e Compartilhamento de Parâmetros	36
3.3.3	Agregação	37
3.4	Regularização	38
3.4.1	Desligamento	38
3.4.2	Normalização em lotes	39
3.4.3	Parada Antecipada	40
3.5	Transferência de Conhecimento	41
3.6	A Função de Custo Triplet	43
<b>4</b>	<b>Trabalhos Relacionados</b>	<b>46</b>
<b>5</b>	<b>Redes Convolucionais Profundas Aplicadas a Sinais SSVEP</b>	<b>52</b>
5.1	Descrição dos Dados de SSVEP	53
5.2	Procedimento para Gerar as Bases de Dados em Diferentes Formatos	54

5.3	Arquiteturas DCNNs para Classificação de Sinais SSVEP . . . . .	57
5.3.1	Uma Arquitetura DCNN para Classificação do Módulo da STFT / Espectrograma . . . . .	58
5.3.2	Uma Arquitetura DCNN para Classificação de Sinais Brutos . . . . .	58
5.4	Transferência de Conhecimento com ResNet . . . . .	59
5.5	Proposta de Arquiteturas DCNNs para Gerar o Mapeamento no Espaço Euclidiano . . . . .	61
<b>6</b>	<b>Experimentos . . . . .</b>	<b>64</b>
6.1	Experimentos com as Arquiteturas DCNNs . . . . .	64
6.1.1	Protocolo para os Experimentos com a DCNN . . . . .	64
6.1.2	Avaliação de Espectrogramas como Entrada da DCNN . . . . .	66
6.1.3	Avaliação de Módulos da STFT como Entrada da DCNN . . . . .	68
6.1.4	Avaliação de Sinais Brutos como Entrada da Rede . . . . .	68
6.1.5	Discussão dos Resultados . . . . .	70
6.2	Experimentos de Transferência de Conhecimento com a Arquitetura ResNet	73
6.2.1	Protocolo para os Experimentos de Transferência de Conhecimento	73
6.2.2	Avaliação de Transferência de Conhecimento com a ResNet . . . . .	74
6.2.3	Discussão dos Resultados . . . . .	78
6.3	Experimentos com a DCNN e a Função de Custo Triplet . . . . .	79
6.3.1	Protocolo para os Experimentos com a Função de Custo Triplet . . . . .	79
6.3.2	Avaliação de Arquitetura DCNN com a Função de Custo Triplet . . . . .	80
6.3.3	Discussão dos Resultados . . . . .	84
	<b>Conclusão . . . . .</b>	<b>85</b>
	<b>Referências . . . . .</b>	<b>87</b>
	<b>ANEXO A Tabelas de Resultados Numéricos . . . . .</b>	<b>97</b>
A.1	Resultados Numéricos dos Experimentos com Espectrogramas como En- trada da Rede . . . . .	97
A.2	Resultados Numéricos dos Experimentos com Módulos da STFT como En- trada da Rede . . . . .	97
A.3	Resultados Numéricos dos Experimentos com Sinais Brutos como Entrada da Rede . . . . .	98
A.4	Resultados Numéricos dos Experimentos com Transferência de Conhecimento	98
A.5	Resultados Numéricos dos Experimentos com a Arquitetura DCNN e a Função de Custo Triplet . . . . .	99

# 1 Introdução

As formas habituais de comunicação ou controle por parte do ser humano requerem nervos e músculos periféricos. Este processo complexo, no qual certas áreas cerebrais são ativadas, em consequência da intenção do indivíduo, envia sinais através do sistema nervoso periférico para os músculos correspondentes, responsáveis por realizar o movimento necessário para a ação desejada (DORNHEGE *et al.*, 2007).

Uma Interface Cérebro-Computador é um sistema artificial que independe de nervos e músculos periféricos, oferecendo uma alternativa à comunicação e ao controle habituais (WOLPAW *et al.*, 2002). Ela opera traduzindo a intenção do usuário, associada à atividade originada e medida diretamente do cérebro, e não dos sistemas periféricos ou dos músculos, em sinais de controle correspondentes aos da aplicação. Esta operação de tradução envolve processamento de sinais e reconhecimento de padrões, geralmente feitos por um computador (NICOLAS-ALONSO; GOMEZ-GIL, 2012). Este sistema tem, tradicionalmente, quatro características (GRAIMANN *et al.*, 2009): ele deve registrar a atividade originada no cérebro, deve fornecer *feedback* ao usuário, deve operar em tempo real e deve operar através do controle intencional, ou seja, ele deve responder somente às tarefas mentais do usuário condizentes com os sinais de controles da aplicação.

Diferentes áreas podem se beneficiar das Interfaces Cérebro-Computador, com aplicações que vão desde a área da saúde até a área de entretenimento (WOLPAW *et al.*, 2002). Alguns exemplos de aplicações que fazem uso destes sistemas são dispositivos de digitação (POSTELNICU; TALABA, 2012), jogos de computador (BONNET *et al.*, 2013), navegação em realidade virtual (LEEB *et al.*, 2007), aplicativos genéricos de controle de cursor (FABIANI *et al.*, 2004) e produção musical cooperativa (TAVARES *et al.*, 2015). Apesar da simplicidade das tarefas executadas por estas aplicações, elas podem ter um grande impacto em pessoas com doenças que levam à síndrome do encarceramento, como a Esclerose Lateral Amiotrófica (ELA), sendo possivelmente a única opção para comunicação em tais casos (GRAIMANN *et al.*, 2009).

O desenvolvimento e o avanço da tecnologia de Interface Cérebro-Computador podem torná-la mais acessível para novos grupos de usuários. Estes sistemas podem fornecer comunicação e controle para usuários com limitações menos severas ou até mesmo usuários saudáveis em algumas situações, como na área de entretenimento (KAPLAN *et al.*, 2013). Podem também fornecer novos meios de tratamento na área médica (GUGER *et al.*, 1999; AFLALO *et al.*, 2015; DONATI *et al.*, 2016). Aplicações voltadas a estes usuários podem demandar novos componentes inteligentes para superar diferentes desafios, como assegurar a fluidez no funcionamento do sistema, garantindo que o *feedback*

visual, ou de outro tipo, chegue corretamente ao usuário (GRAIMANN *et al.*, 2009).

## 1.1 Motivação

As redes neurais profundas têm sido aplicadas a problemas de diferentes áreas como a visão computacional, processamento de linguagem natural, processamento de áudio, sistemas autônomos, diagnósticos médicos, cibersegurança, previsão de séries temporais, entre outras (GOODFELLOW *et al.*, 2016; PATTERSON; GIBSON, 2017; HATCHER; YU, 2018), obtendo resultados que, muitas vezes, ultrapassam o estado-da-arte. Um atributo interessante destas redes, em particular as redes convolucionais profundas, é a capacidade de extração de características de forma automática, dispensando, frequentemente, a engenharia de características manual, etapa quase que obrigatória para um grande número de técnicas de aprendizado de máquina (PATTERSON; GIBSON, 2017).

Um exame da etapa de processamento do sinal na segmentação da operação das Interfaces Cérebro-Computador, apresentada por Wolpaw *et al.* (2002), aponta para uma subetapa dedicada ao pré-processamento do sinal de entrada, uma dedicada à extração de características, uma dedicada à seleção de características e uma dedicada à classificação. Um cenário atraente é a substituição de parte ou todas as subetapas por uma única etapa, por exemplo, empregando tão somente uma rede convolucional profunda. O sucesso que estas redes têm obtido em outras áreas indica que há potencial de simplificação do sistema e melhoria de seu desempenho, seja diminuindo o tempo de treinamento para um usuário ao sistema, seja diminuindo a taxa de erro de classificação dos sinais cerebrais.

Empregar uma rede convolucional profunda em um problema pode demandar uma quantidade massiva de dados para um bom desempenho (GOODFELLOW *et al.*, 2016), fator que pode dificultar seu uso em Interfaces Cérebro-Computador. É habitual ajustar estes sistemas individualmente aos usuários (CARVALHO *et al.*, 2015a), empregando algoritmos lineares com uma quantidade reduzida, mas suficiente, de dados. Cenários sujeitos à avaliação são o treinamento das redes, ou ajuste de redes pré-treinadas, usando os dados de vários usuários de forma a generalizar para um usuário que nunca participou da etapa de treinamento. Estas abordagens podem trazer novas perspectivas para estes sistemas em relação à etapa de ajuste a novos usuários.

## 1.2 Objetivos

Motivado pelos resultados promissores das redes profundas em diferentes áreas, pela possibilidade de substituir uma ou mais subetapas geralmente adotadas na síntese de Interfaces Cérebro-Computador e pela possibilidade de treinar um modelo de forma mais robusta, capaz de generalizar para um novo usuário sem um ajuste prévio, con-

forme apontado anteriormente, este trabalho explora, como objetivo principal, as redes convolucionais profundas na tarefa de processamento de sinais cerebrais em Interfaces Cérebro-Computador.

Para as investigações deste trabalho, são empregados dados de uma Interface Cérebro-Computador baseada em potenciais visualmente evocados em estado estacionário. Dentre as interfaces mais estudadas, estas se destacam por sua proposta simples e direta (CARVALHO *et al.*, 2015a). Essa simplicidade de projeto, no entanto, não exime este tipo de interface da necessidade de treinamento, em tese, a cada utilização.

O primeiro passo é investigar se formatos diferentes de entrada, como o sinal bruto ou alguma transformação do sinal, por exemplo, a transformada de Fourier de tempo curto, influenciam no desempenho das redes convolucionais profundas. Para esta investigação, são propostas novas arquiteturas de redes convolucionais profundas, avaliadas sobre diferentes tipos de entrada, e seus desempenhos são comparados aos de abordagens usualmente adotadas na etapa de classificação de Interfaces Cérebro-Computador. Presentemente, os trabalhos encontrados na literatura não fazem uma comparação direta entre formatos diferentes de entrada. Estes usam apenas um formato para comparar o desempenho das redes profundas aos métodos tradicionais mais usados.

O segundo passo é investigar se é possível ajustar, aos dados de Interfaces Cérebro-Computador, redes convolucionais profundas pré-treinadas em bases de dados diversas. Para esta investigação, é interessante usar uma arquitetura consolidada que permita avaliar diferentes profundidades do modelo, como a rede residual. Até onde pudemos observar, este trabalho é o primeiro a avaliar a técnica de transferência de conhecimento com redes convolucionais profundas em Interface Cérebro-Computador baseada em potenciais visualmente evocados em estado estacionário.

### 1.3 Organização da Dissertação

Esta dissertação está organizada da seguinte maneira:

**Capítulo 2:** este Capítulo apresenta os principais conceitos envolvidos em Interfaces Cérebro-Computador; apresenta também as etapas destes sistemas e uma descrição breve de algumas técnicas empregadas em cada etapa, assim como apresenta os paradigmas mais comuns para estes sistemas.

**Capítulo 3:** este Capítulo apresenta os principais conceitos envolvidos no aprendizado profundo, apresenta também as redes convolucionais, um dos tipos de rede empregado nesta abordagem, assim como apresenta algumas técnicas de regularização aplicadas na tentativa de melhorar a generalização dos modelos.

**Capítulo 4:** a revisão da literatura sobre redes convolucionais profundas aplicadas a dados de Interface Cérebro-Computador baseadas no paradigma de potenciais visualmente evocados em estado estacionário é apresentada neste Capítulo.

**Capítulo 5:** uma discussão sobre a aplicação de redes neurais convolucionais profundas a dados de uma Interface Cérebro-Computador, usando o paradigma de potenciais visualmente evocados em estado estacionário. Também são apresentadas as arquiteturas de rede propostas e uma abordagem para transferência de conhecimento usando a arquitetura de rede residual.

**Capítulo 6:** são apresentados o protocolo experimental e os resultados e discussões relacionados aos experimentos com redes neurais convolucionais profundas e transferência de conhecimento.

**Conclusão:** por fim, são apresentadas as conclusões gerais sobre o trabalho, assim como as questões em aberto e as perspectivas para trabalhos futuros.

**Anexo A:** são descritos os resultados numéricos referentes aos gráficos de resultados do Capítulo 6.

## 1.4 Produções Acadêmicas

Durante o período de desenvolvimento do mestrado, foi publicado um trabalho diretamente ligado à temática desta dissertação:

- P. R. A. S. Bassi, W. Rampazzo, R. Attux, "Redes Neurais Profundas Triplet Aplicadas à Classificação de Sinais em Interfaces Cérebro-Computador", XXXVII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais - SBrT 2019, Petrópolis, RJ.

Além disso, um trabalho foi publicado em parceria com pesquisadores ligados ao Laboratório de Bioinformática e Computação Bio-Inspirada (LBiC, *Laboratory of Bioinformatics and Bio-Inspired Computing*):

- A. O. C. Ayres, W. Rampazzo, E. R. S. Faria, P. M. S. Bezerra, L. Bocato, R. Attux, F. J. Von Zuben, "Adaptive Radius Immune Algorithm for Multimodal Function Optimization in Dynamic Environments", XX International Conference on Artificial Intelligence - ICAI 2018, Las Vegas, Nevada, USA.

## 2 Interfaces Cérebro-Computador

As Interfaces Cérebro-Computador são sistemas que possibilitam a comunicação direta entre o cérebro e o computador, permitindo a interação sem a necessidade de movimentos musculares, tornando-a atrativa tanto para tecnologias assistivas quanto para aplicações de entretenimento (WOLPAW *et al.*, 2002).

Em um dos primeiros trabalhos sobre Interfaces Cérebro-Computador, Vidal (1973) demonstrou que era factível a construção deste tipo de sistema processando sinais cerebrais em tempo de execução com os recursos computacionais disponíveis na época. Este trabalho pode ser considerado como um dos incentivadores para as posteriores pesquisas, que, desde então, se ramificaram em diferentes áreas relacionadas à Interfaces Cérebro-Computador, seus componentes e suas aplicações.

Neste Capítulo, serão apresentados os principais conceitos e etapas de uma Interface Cérebro-Computador. A Seção 2.1 apresenta a definição de uma Interface Cérebro-Computador. A Seção 2.2 descreve a etapa de aquisição de sinais cerebrais. A Seção 2.3 apresenta os paradigmas mais comuns. A Seção 2.4 descreve a etapa de processamento de sinais cerebrais e a Seção 2.5 descreve a etapa de controle relacionada a uma Interface Cérebro-Computador.

### 2.1 Definição de Interfaces Cérebro-Computador

O termo Interfaces Cérebro-Computador (BCI, do inglês, *Brain-Computer Interfaces*) foi formalmente definido em 1999, na primeira reunião internacional para investigação e desenvolvimento de BCI, como "um sistema de comunicação que não depende dos canais de saída convencionais do cérebro, dos nervos periféricos e músculos"(WOLPAW *et al.*, 2002). Assim, um usuário de uma BCI interage com o meio através de mensagens ou comandos traduzidos das medidas diretas de sua atividade cerebral, sem o uso de suas referências biológicas clássicas, como a fala ou o movimento através dos músculos.

A pesquisa e o desenvolvimento de BCIs visam fornecer aos usuários um meio de comunicação ou controle através de informações adquiridas diretamente dos sinais cerebrais. De modo geral, uma BCI consiste em um sistema de controle ou comunicação em malha fechada em que o usuário interage através de suas intenções, gerando sinais cerebrais associados aos comandos de interesse. O sinal cerebral é adquirido, processado e resulta em um sinal de controle que pode ser enviado a um dispositivo. Estas etapas são os principais componentes de qualquer sistema de BCI, como exposto por Wolpaw *et al.* (2002). A Figura 1 apresenta uma representação gráfica destas etapas, com o detalhamento

da etapa de processamento do sinal. Os processos envolvidos em cada uma das etapas serão detalhados nas Seções seguintes.

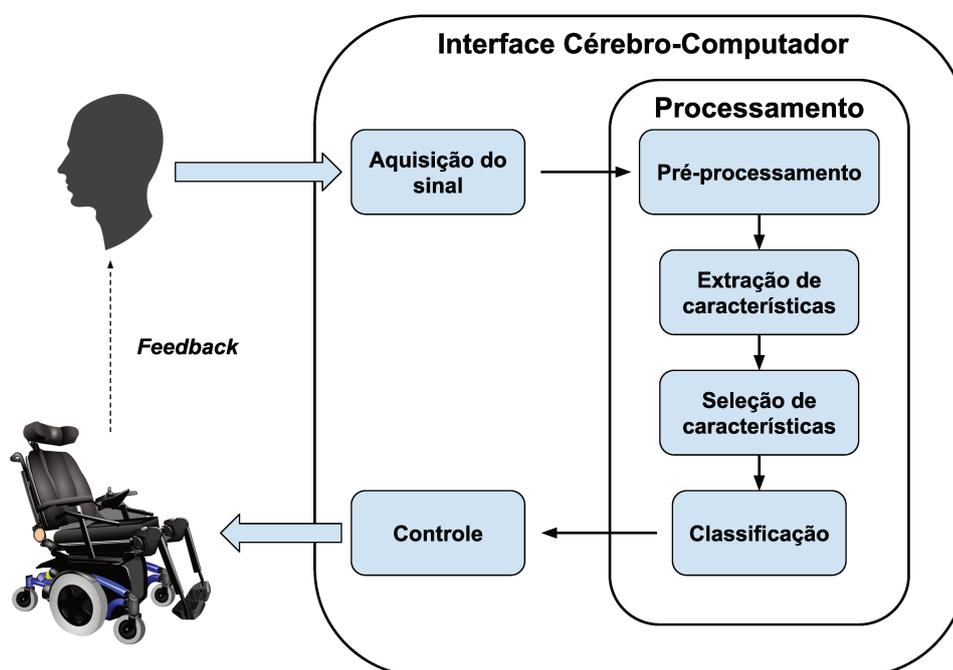


Figura 1 – Representação das etapas de uma BCI segundo Wolpaw *et al.* (2002).

## 2.2 Aquisição de Sinais Cerebrais

Em uma BCI, a aquisição dos sinais cerebrais pode ser considerada uma etapa crítica, dado que a medição da atividade cerebral deve condizer com a intenção de comando do usuário. A atividade cerebral pode ser registrada por meio de diferentes técnicas, com variadas resoluções temporais e espaciais, e de forma invasiva ou não invasiva (NICOLAS-ALONSO; GOMEZ-GIL, 2012).

O registro dos sinais por meio de técnicas invasivas consiste em implantar cirurgicamente os eletrodos no usuário e capturar o sinal neural diretamente do parênquima cerebral. O eletrocorticograma (ECoG) (SCHALK; LEUTHARDT, 2011) e a gravação intracortical (NICOLELIS, 2001) são as técnicas invasivas mais comuns. A primeira registra a atividade de um conjunto de neurônios através de eletrodos posicionados na superfície do córtex. Já a segunda registra a atividade de um conjunto espacialmente limitado de neurônios, ou até mesmo de um único neurônio através de eletrodos posicionados diretamente sobre o tecido cortical. Apesar da boa relação sinal-ruído e alta resolução espacial, técnicas invasivas oferecem o risco da neurocirurgia e o risco pós-cirúrgico, nem sempre viável, e um grande impacto financeiro, dado que intervenções cirúrgicas costumam ser caras.

Nas técnicas não invasivas, não há a necessidade de intervenções cirúrgicas ou procedimentos que ofereçam risco ao usuário para o registro das atividades cerebrais. As técnicas não invasivas mais comuns são a eletroencefalografia (EEG), na qual eletrodos são dispostos sobre o escalpo do usuário para monitorar a atividade elétrica do cérebro (NIDAL; MALIK, 2014), a ressonância magnética funcional (fMRI, do inglês, *functional Magnetic Resonance Imaging*), na qual as atividades cerebrais, identificadas através das alterações no fluxo sanguíneo, são medidas via ressonância magnética (WEISKOPF *et al.*, 2004), a magnetoencefalografia (MEG), na qual os campos magnéticos gerados naturalmente pelas correntes elétricas no cérebro são monitorados por magnetômetros (MELLINGER *et al.*, 2007) e a espectroscopia de infravermelho próximo (NIRS, do inglês, *Near InfraRed Spectroscopy*), na qual o fluxo de oxigenação cerebral é monitorado via luz infravermelha (COYLE *et al.*, 2007). Apesar do eventual risco em casos raros e específicos, técnicas não invasivas apresentam baixa relação sinal-ruído em razão das barreiras físicas entre o córtex e o equipamento de aquisição.

Neste trabalho, são utilizados dados de sinais cerebrais adquiridos via EEG; desta forma, a próxima Seção detalha este tipo de aquisição e, nas Seções seguintes, os assuntos são introduzidos levando em consideração aquisições feitas através de EEG.

### 2.2.1 Aquisição de Sinais Cerebrais via EEG

Para a aquisição de potencial elétrico do sinal cerebral via EEG de forma padronizada, os eletrodos são dispostos sobre o escalpo considerando algum dos padrões disponíveis. Um destes padrões é conhecido como sistema internacional 10-10, proposto em Chatrian *et al.* (1985) como uma extensão do sistema 10-20 (JASPER, 1958). A base para o posicionamento dos eletrodos nos dois sistemas é a mesma. São tomadas duas medidas de referência passando pelo ponto superior central do escalpo, uma entre o násio e o ínio, indicados por N e I na Figura 2, e outra entre os pontos pré-auriculares esquerdo e direito, indicados por T1 e T2 na Figura 2. O sistema internacional 10-10 permite a disposição de até 81 eletrodos no escalpo. A distribuição dos eletrodos no sistema internacional 10-10 pode ser vista na Figura 2.

Os equipamentos mais modernos utilizam toucas com marcações dos padrões de referência para o posicionamento dos eletrodos. Os eletrodos são fixados nas posições desejadas da touca e, seguindo as medidas de referência entre o násio e o ínio e os pontos pré-auriculares, o usuário veste a touca que garante o contato e o posicionamento correto dos eletrodos no escalpo. Uma opção prática e flexível é usar eletrodos secos, porém há degradação na qualidade da aquisição com a redução na relação sinal-ruído. Uma forma de melhorar o contato entre o escalpo e os eletrodos é por meio da aplicação de gel, porém este processo pode diminuir a praticidade.

A verificação da atividade cerebral pode ser realizada medindo a diferença de

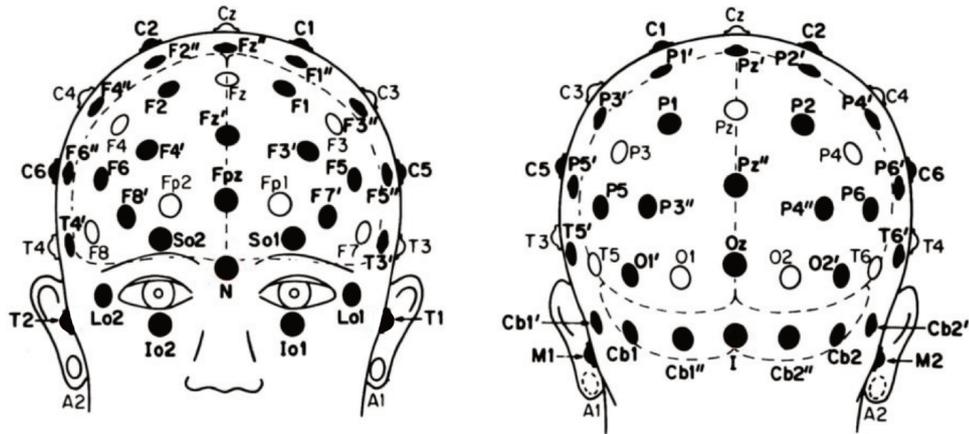


Figura 2 – Distribuição dos eletrodos no sistema internacional 10-10 (CHATRIAN *et al.*, 1985).

potencial entre os eletrodos de dois métodos diferentes, conhecidos como referencial e bipolar (MULERT; LEMIEUX, 2009). No referencial, os eletrodos são posicionados na região de interesse do escalpo e a diferença de potencial é medida entre eles e um eletrodo de referência, posicionado em uma região neutra, com baixa atividade cerebral, como os lóbulos das orelhas, por exemplo. No bipolar, a diferença de potencial é medida entre pares de eletrodos posicionados sobre regiões específicas do escalpo.

Em virtude de o sinal cerebral adquirido através de EGG estar limitado à ordem de algumas dezenas de microvolts, é necessário amplificá-lo por meio de amplificadores de biosinais ou técnicas computacionais. O processo de aquisição do sinal cerebral, aliado a sua baixa ordem de grandeza, tornam inevitável a presença de artefatos, instrumentais ou fisiológicos, no sinal de EEG (NOLAN *et al.*, 2010). Artefatos instrumentais estão relacionados ao meio e podem ser, por exemplo, interferência eletromagnética de equipamentos ao redor, interferência da rede elétrica, entre outros. Artefatos fisiológicos estão relacionados ao próprio usuário durante a aquisição e podem ser, por exemplo, batimentos cardíacos, movimentos musculares, o piscar dos olhos, entre outros.

## 2.3 Paradigmas de BCI

Uma BCI pode fazer aquisição de sinais cerebrais explorando diferentes fenômenos neurofisiológicos através dos mecanismos endógenos ou exógenos de geração de sinais (CHAN *et al.*, 2015). Mecanismos endógenos estão relacionados à imaginação de tarefas e dependem diretamente da capacidade do usuário para induzir uma resposta cerebral, imaginando, por exemplo, o movimento de uma das mãos, ou fazendo um cálculo mentalmente. Mecanismos exógenos estão relacionados à atenção seletiva do usuário e dependem de estímulos externos para induzir uma resposta cerebral, por meio, por exemplo, de uma fonte de luz oscilante ou de um sinal sonoro.

### 2.3.1 Sistemas BCI Baseados em Imagética

Os sistemas BCI endógenos estão diretamente relacionados com os processos cognitivos voluntários do usuário. Paradigmas de BCI que adotam este mecanismo usam a imaginação do usuário para estimular estes processos cognitivos, podendo ser imagética motora ou não motora.

Como o próprio nome sugere, na imagética motora, o usuário imagina o movimento de um membro, como o pé ou a mão, por exemplo, sem propriamente executá-lo. A imaginação do movimento de um membro gera uma atividade no córtex motor com o mesmo perfil da atividade gerada quando o usuário efetivamente executa o movimento daquele membro, permitindo que, através de análise do sinal, esta atividade seja identificada (HAMEDI *et al.*, 2016). A imagética motora tem aplicações práticas que vão de jogos eletrônicos (BONNET *et al.*, 2013) à movimentação de cadeira de rodas (ZHANG *et al.*, 2015) e próteses (AFLALO *et al.*, 2015).

Na imagética não motora, o usuário realiza tarefas mentais, como imaginação de figuras geométricas, imaginação de uma música ou realização de cálculos mentais. Estas tarefas geram atividades em áreas específicas do cérebro, permitindo que, novamente através da análise do sinal, a atividade seja identificada (ABO-ZAHHAD *et al.*, 2015). A região do cérebro a ser monitorada na imagética não motora pode variar e é diretamente dependente do tipo de tarefa mental.

### 2.3.2 Sistemas BCI Baseados em Atenção Seletiva

Os sistemas BCI exógenos estão relacionados com a habilidade de atenção seletiva, que permite ao usuário se concentrar em um único estímulo, dentre inúmeros estímulos presentes no ambiente. A atenção seletiva pode ser ativada através de estímulos visuais, auditivos ou somatossensoriais que são inconscientemente traduzidos em sinais cerebrais com padrões específicos, permitindo, através de análise do sinal, a identificação destes padrões (KELLY *et al.*, 2005). Dentre as abordagens que se enquadram como sistemas BCI exógenos, P300 e Potenciais Visualmente Evocados em Estado Estacionário (SSVEP, do inglês, *Steady State Visually Evoked Potentials*) são comumente utilizadas (SETARE *et al.*, 2013).

Na abordagem P300, um potencial gerado de maneira inconsciente acontece, principalmente na região do lobo parietal, cerca de 300ms após o disparo de um evento esperado, isto é, quando uma expectativa do usuário é atendida (KAPLAN *et al.*, 2013). A Figura 3 representa uma onda com vários componentes de potenciais relacionados a eventos, sendo o ponto P3 o pico do P300.

A primeira BCI a utilizar o P300 foi proposta por Farwell e Donchin (1988) e consistia de um teclado projetado com caracteres dispostos em uma matriz de 6 linhas por

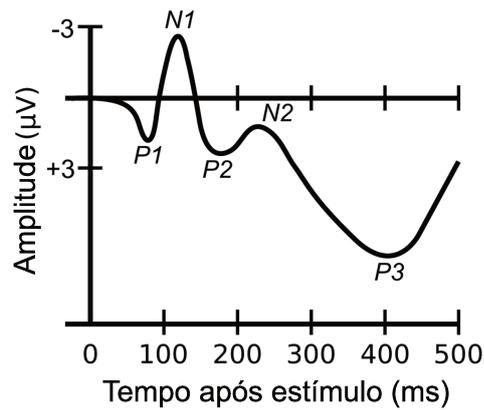


Figura 3 – Onda P300.

6 colunas. O seu princípio de funcionamento consiste de destaques aleatórios das linhas e colunas enquanto o usuário fixa o olhar em um dos caracteres. Quando a intersecção entre a linha e a coluna de destaque corresponde ao caractere desejado pelo usuário, uma onda P300 é gerada permitindo a identificação da letra pela BCI através da sincronia entre os destaques e a detecção da onda P300.

BCIs baseadas em P300 podem ser consideradas robustas, dada a confiabilidade na detecção da resposta P300. Tendo em vista que a onda P300 é gerada inconscientemente e de forma natural pelo cérebro, estas BCIs também podem ser consideradas genéricas, permitindo a utilização independente do usuário, com nenhuma ou poucas sessões de treinamento prévio (POSTELNICU; TALABA, 2012).

Na abordagem SSVEP, o interesse é identificar um padrão de atividade cerebral que ocorre quando o usuário é exposto a estímulos oscilatórios em determinada frequência. Estes estímulos podem ser induzidos, por exemplo, através de imagens ou de emissores de luz oscilando. O padrão de atividade cerebral, representado pelos disparos dos neurônios, ocorre involuntariamente, principalmente na região do córtex visual, quando um usuário se concentra no estímulo oscilante. Assim, os disparos dos neurônios ocorrem na mesma frequência em que oscila o estímulo visualizado pelo usuário, gerando um aumento da energia do sinal na frequência correspondente (MIDDENDORF *et al.*, 2000).

A Figura 4 apresenta uma interface de estímulos visuais para 4 frequências em que cada um dos quadrados oscila entre a cor preta e a cor branca em uma frequência determinada. Quando o usuário se concentra em um dos estímulos visuais e a sincronização dos disparos dos neurônios no córtex visual é detectada, uma moldura na cor cinza é colocada no quadrado oscilando na frequência correspondente como forma de *feedback* para o usuário. Na Figura 4, o quadrado superior exibe a moldura na cor cinza representando o estímulo visual no qual o usuário está se concentrando.

As técnicas de extração de informações espectrais são uma opção relevante para análise do sinal na abordagem SSVEP (JOHANSSON; JAKOBSSON, 2000). A Figura 5

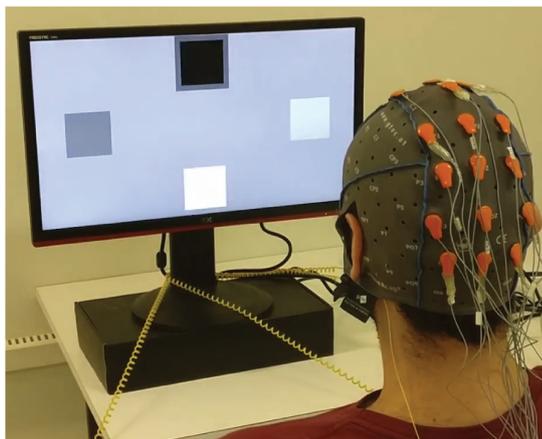


Figura 4 – Estímulos visuais para uma BCI SSVEP.

apresenta as representações no domínio do tempo e no domínio da frequência de um sinal SSVEP adquirido via EEG, proveniente do eletrodo Oz, posicionado usando o sistema internacional 10-10, de um usuário exposto a um estímulo visual oscilando na frequência de 15Hz. Embora a informação do sinal cerebral esteja disponível nas duas representações, é possível identificar, no espectro de potência de forma quase imediata, o estímulo visual ao qual o usuário está concentrado, observando o aumento de magnitude em torno da frequência de 15Hz.

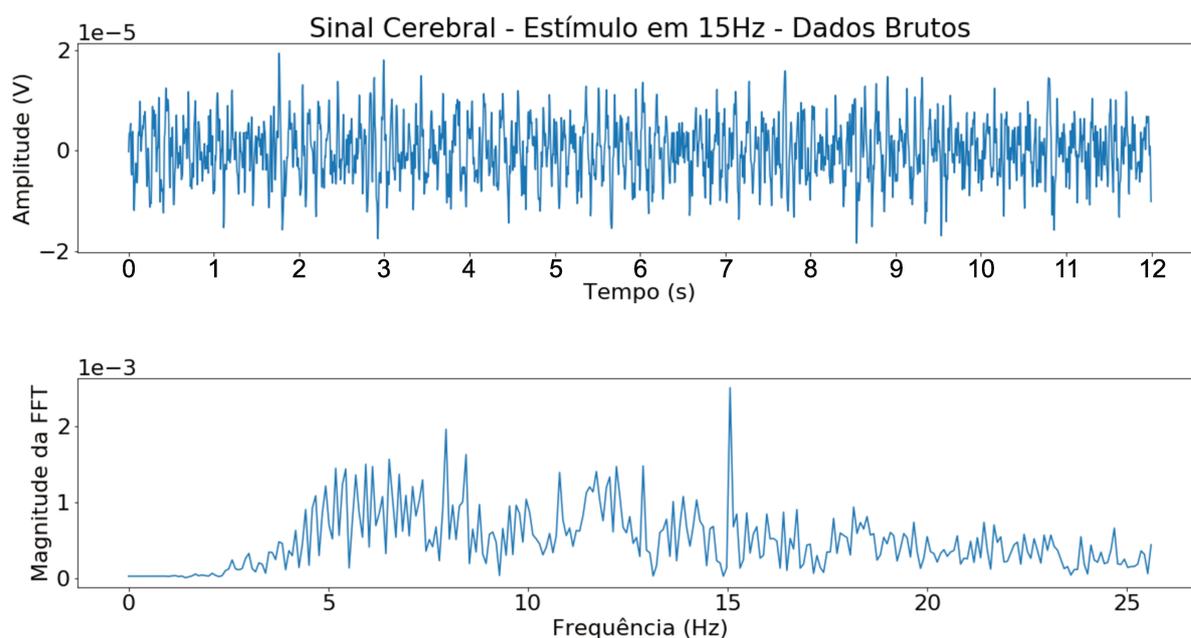


Figura 5 – Sinal de EEG proveniente do eletrodo Oz no domínio do tempo e da frequência para um estímulo de 15Hz.

Sistemas BCI endógenos dependem de um maior esforço cognitivo do usuário quando comparados a sistemas BCI exógenos. As tarefas mentais exigem treinamento e um alto nível de concentração do usuário, podendo ter interferência de outras operações mentais, enquanto em sistemas BCI exógenos os usuários precisam apenas se concentrar

no estímulo externo, sem a dependência de tarefas cognitivas (MCFARLAND; WOLPAW, 2011).

## 2.4 Processamento de Sinais Cerebrais

A etapa de processamento dos sinais cerebrais pode ser dividida nas subetapas de pré-processamento, extração de características, seleção de características e classificação. Nestas subetapas, o sinal de entrada é interpretado e traduzido para sinais de controle e/ou mensagens. Cada uma das subetapas pode variar de acordo com o paradigma de BCI implementado. Este trabalho leva em consideração BCIs SSVEP para apresentar, nas próximas subseções, as subetapas do processamento.

### 2.4.1 Pré-Processamento

Conforme apontado previamente, os sinais de EEG durante sua aquisição estão suscetíveis a interferências. A finalidade da etapa de pré-processamento é aumentar a relação sinal-ruído, por meio de técnicas de filtragem, por exemplo, para as próximas subetapas do processamento de sinais. Como as informações de interesse se apresentam de forma bem definida nas frequências do sinal de SSVEP, interferências que estão fora destas frequências acabam não impactando o desempenho da BCI (BRUNNER *et al.*, 2010).

A Referência de Média Comum (CAR, do inglês, *Common Average Reference*) é um exemplo de filtragem espacial aplicada durante o pré-processamento (DIEN, 1998). Esta técnica, matematicamente simples e eficiente, reduz as componentes que aparecem simultaneamente e com intensidade semelhante em todos os eletrodos. Ela é aplicada subtraindo a média dos potenciais de todos os eletrodos de cada um dos eletrodos conforme a Equação 2.1:

$$V_i^{CAR} = V_i^{ER} - \frac{1}{N_e} \sum_{j=1}^{N_e} V_j^{ER} \quad (2.1)$$

no qual  $V_i^{ER}$  refere-se ao potencial do  $i$ -ésimo eletrodo e  $N_e$  ao número total de eletrodos. Cabe observar que o CAR pode ser aplicado somente em aquisições feitas pelo método referencial, apresentado anteriormente.

### 2.4.2 Extração de Características

Durante a etapa de extração de características, podem ser empregadas técnicas com base no conhecimento a priori sobre as características dos fenômenos neurofisiológicos explorados (DORNHEGE *et al.*, 2007), de forma a mapear o sinal de entrada

em um espaço de dimensão reduzida resultando em uma forma compacta e eficiente da representação das informações relevantes. Estas características podem ser extraídas no domínio do tempo, no domínio da frequência e no domínio do espaço (RAMOSER *et al.*, 2000; FABIANI *et al.*, 2004; GROSSE-WENTRUP; BUSS, 2008). Exemplos de técnicas que podem ser empregadas nesta etapa são a transformada de Fourier (OPPENHEIM; VERGHESE, 2015), a transformada *Wavelet* (MALLAT, 1999), a Análise de Componentes Principais (PCA, do inglês, *Principal Component Analysis*) e a análise de recorrência de dados (CHANG; PARK, 2013).

Conforme discutido anteriormente, no paradigma SSVEP é comum empregar técnicas que exploram informações espectrais (CARVALHO *et al.*, 2015a), como a transformada de Fourier, ou alguma de suas variações, como a transformada rápida de Fourier (FFT, do inglês, *Fast Fourier Transform*) (COOLEY; TUKEY, 1965) para computação mais eficiente da transformada de Fourier e a transformada de Fourier de tempo curto (STFT, do inglês, *Short-Time Fourier Transform*) (DUHAMEL; VETTERLI, 1990).

A STFT permite o cálculo da transformada de Fourier sobre segmentos curtos do sinal, sendo possível obter informações de magnitude e fase em cada ponto no tempo e na frequência mediante a Equação 2.2.

$$X(m, \omega) = \sum_{n=-\infty}^{+\infty} x(n)w(n-m)e^{-j\omega n} \quad (2.2)$$

na qual  $x(n)$  representa o sinal e  $w(n-m)$  representa a janela.

Uma outra forma de analisar a variação do espectro de frequências de um sinal ao longo do tempo é pelo espectrograma, calculado a partir da STFT conforme a Equação 2.3.

$$\text{espectrograma} \equiv |X(m, \omega)|^2 \quad (2.3)$$

onde  $X(m, \omega)$  representa a STFT do sinal  $x(n)$  para um segmento de tamanho  $m$  e frequência angular  $\omega$ .

Um fluxo possível para a etapa de extração de características seria receber uma janela do sinal cerebral filtrado pela etapa de pré-processamento, aplicar a STFT nesta janela e transformar em um espectrograma representando a forma compacta das informações relevantes.

### 2.4.3 Seleção de Características

Durante a etapa de seleção de características, são determinadas as características mais informativas a serem usadas na elaboração do classificador, de forma que ele seja rápido, eficiente e que as características selecionadas melhorem o seu desempenho

(BISHOP, 1995). Esta etapa pode ajudar a simplificar o modelo para um melhor entendimento do problema, pode facilitar a visualização e o próprio entendimento dos dados, pode reduzir o tempo de treinamento do classificador, pode aprimorar a generalização do modelo, reduzindo o sobreajuste, entre outros (BISHOP, 1995). Para a seleção de características, as abordagens comumente empregadas são filtros e *wrappers* (KOHAVI; JOHN, 1997; GUYON; ELISSEEFF, 2003).

No espectrograma de saída do fluxo de exemplo anterior, uma possibilidade para seleção de características seria separar somente as frequências de interesse, correspondentes às frequências dos estímulos visuais, e adicioná-las a uma matriz de características. Uma outra possibilidade para a seleção de características seria aproveitar os sinais de somente alguns dos eletrodos do conjunto usado para aquisição empregando *wrappers* ou privilegiando eletrodos de uma determinada região do escalpo.

#### 2.4.4 Classificação

Na última subetapa no fluxo do processamento de sinais, a informação de entrada é mapeada em sinais de controle e/ou mensagens. Duas possíveis abordagens para esta tarefa seriam a regressão e a classificação. Na regressão, o sinal de entrada é avaliado para gerar um sinal de controle contínuo. Na classificação, o sinal de entrada é mapeado para uma das possíveis classes, associadas aos sinais de controle, sendo esta abordagem a mais comum para sistemas BCI (DORNHEGE *et al.*, 2007).

Na construção do classificador, é usual o emprego de abordagens de aprendizado de máquina, como técnicas de reconhecimento de padrões. Algumas das técnicas aplicadas no projeto de classificadores, supondo classes linearmente separáveis, são a análise discriminante de Fisher (FISHER, 1936), a função discriminante de quadrados mínimos e a máquina de vetor-suporte (SVM, do inglês, *Support-Vector Machine*) com *kernel* linear (BURGES, 1998), ou supondo classes não-linearmente separáveis, as redes neurais artificiais (BISHOP, 1995) e a SVM com *kernel* não-linear.

Uma estratégia para o ajuste supervisionado dos parâmetros do classificador é a validação cruzada. Nela os dados são separados e avaliados em duas etapas: o treinamento e o teste (BISHOP, 1995). Durante o treinamento, os parâmetros, ou a superfície de decisão, do classificador são ajustados com os dados de treinamento disponíveis e suas classes correspondentes. O classificador pode ser avaliado em um conjunto de dados que não foi usado durante o treinamento, conhecido como conjunto de validação, obtendo um indicativo da capacidade de generalização do classificador e possibilitando eventuais ajustes nos seus parâmetros. No teste, após o ajuste dos parâmetros do classificador, dados que nunca foram considerados na etapa de treinamento e de validação são usados para avaliar sua capacidade de generalização.

Na validação cruzada do tipo *holdout*, os dados são divididos em dois conjuntos mutuamente exclusivos, geralmente na proporção de 2/3 para treinamento (conjunto de treinamento e conjunto de validação) e 1/3 para teste. Na validação cruzada do tipo *k-fold*, os dados são divididos em  $k$  subconjuntos mutuamente exclusivos de mesmo tamanho. Cada um dos subconjuntos é utilizado como conjunto de teste. O modelo é avaliado  $k$  vezes alternando o subconjunto de teste. O desempenho do modelo é calculado pela média dos erros dos  $k$  modelos avaliados.

## 2.5 Controle

A forma como o usuário interage com a BCI está relacionada ao paradigma adotado na sua construção. Podem ser consideradas pelo menos duas formas de interação do usuário com a BCI (BORISOFF *et al.*, 2004). No controle assíncrono, o usuário decide quando aciona a interação com o sistema BCI, que fica disponível de forma contínua. Neste caso, períodos de controle intencional se alternam com períodos sem nenhum controle, ou de repouso, em que o sistema se mantém inalterado, sendo regido pelo próprio usuário. Diferentemente, no controle sincronizado, caso o usuário não execute um comando, ou seja, não responda ao estímulo, o sistema pode responder com uma ação indesejada, causada por uma classificação aleatória.

Sistemas BCIs assíncronos são um campo de estudo com aplicações diversas (MILLAN; MOURIÑO, 2003; LEEB *et al.*, 2007; GALÁN *et al.*, 2008). Porém, este trabalho se restringe à etapa de processamento dos sinais cerebrais considerando sinais de controle síncronos do sistema BCI.

## 3 Aprendizado Profundo

O aprendizado profundo tem atraído a atenção da comunidade científica e do público em geral de forma crescente (HATCHER; YU, 2018). De fato, ele tenta responder às mesmas questões fundamentais de regressão e classificação do aprendizado de máquina. A diferença é que ele explora implicitamente a propriedade de representação hierárquica interna, na qual as características de mais alto nível podem ser obtidas a partir da composição de características de níveis mais baixos (GOODFELLOW *et al.*, 2016). Explorar esta propriedade em arquiteturas profundas em conjunto com outras técnicas, como a regularização e a transferência de conhecimento, têm se mostrado promissor para responder a algumas daquelas questões (PATTERSON; GIBSON, 2017).

Neste Capítulo, serão apresentados os principais conceitos usados no aprendizado profundo. A Seção 3.1 apresenta, de forma simplificada, as redes neurais artificiais. A Seção 3.2 conceitua o aprendizado profundo. A Seção 3.3 descreve as redes convolucionais e suas principais características. A Seção 3.4 descreve o conceito e algumas abordagens de regularização. A Seção 3.5 apresenta a abordagem de transferência de conhecimento e a Seção 3.6 contém uma descrição detalhada da função de custo *triplet*.

### 3.1 Redes Neurais Artificiais

O primeiro modelo matemático inspirado no neurônio biológico foi introduzido por McCulloch e Pitts (1943). Este trabalho não só deu origem ao neurônio artificial, mas também pode ser considerado como um dos motivadores para os estudos subsequentes de neurocomputação. O postulado de Hebb (HEBB, 1949), por sua vez, considerado como uma das primeiras propostas de aprendizado neurobiológico, assim como uma das primeiras regras para treinamento de redes neurais artificiais (ANN, do inglês, *Artificial Neural Networks*) (KURISCAK *et al.*, 2015), descreve a capacidade de aprendizado através das correlações entre neurônios.

Inspirado no neurônio artificial, Rosenblatt (1958) apresenta o perceptron, um classificador linear com a inclusão de pesos adaptativos. Matematicamente, o perceptron aplica uma função não-linear  $f : \mathbb{R}^n \mapsto \mathbb{R}$  à saída de um modelo linear dado pela combinação ponderada de valores de entrada mais um limiar, conforme a Equação 3.1.

$$f(\mathbf{x}) = \phi(\mathbf{w}^T \mathbf{x} + b) \quad (3.1)$$

A função não-linear  $\phi : \mathbb{R} \mapsto \mathbb{R}$  é chamada de função de ativação,  $\mathbf{w} \in \mathbb{R}^n$  é o vetor de pesos adaptativos,  $b \in \mathbb{R}$  é um limiar,  $\mathbf{x} \in \mathbb{R}^n$  é o vetor de entrada e  $f(\mathbf{x})$  é a

saída escalar. A Figura 6 apresenta uma representação gráfica de um perceptron.

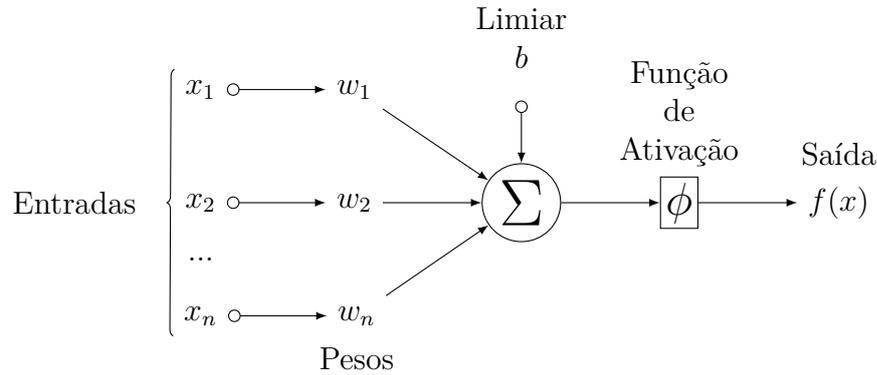


Figura 6 – Modelo não-linear de um neurônio.

Uma organização de perceptrons em camada  $f : \mathbb{R}^n \mapsto \mathbb{R}^n$  é uma aplicação paralela da função de ativação representada pela Equação 3.2:

$$f(\mathbf{x}) = \phi(\mathbf{W}\mathbf{x} + \mathbf{b}) \quad (3.2)$$

onde  $\mathbf{W} \in \mathbb{R}^{n \times n}$  agora representa uma matriz de pesos,  $\mathbf{b} \in \mathbb{R}^n$  um vetor de limiares e  $f(\mathbf{x})$  um vetor de saída. A função de ativação  $\phi : \mathbb{R}^n \mapsto \mathbb{R}^n$  é aplicada elemento a elemento.

Uma ANN *feedforward* organizada em um arranjo com uma camada de entrada, pelo menos uma camada intermediária, que realiza um mapeamento não-linear entre a entrada e a saída, e uma camada de saída é conhecida como perceptron de múltiplas camadas (MLP, do inglês, *Multilayer Perceptron*). A representação matemática de uma MLP com uma camada de entrada, uma camada intermediária e uma camada de saída é dada pela Equação 3.3:

$$\hat{\mathbf{y}} = f_1(f_2(\mathbf{x})) \quad (3.3)$$

onde  $f_2 : \mathbb{R}^n \mapsto \mathbb{R}^n$  é o mapeamento feito pela camada intermediária e  $f_1 : \mathbb{R}^n \mapsto \mathbb{R}^n$  é o mapeamento feito pela camada de saída. A Figura 7 ilustra esta MLP. Cada círculo representa um neurônio. Neurônios da camada de entrada simplesmente propagam seus sinais. Neurônios da camada intermediária realizam o mapeamento não-linear  $f_2$  e propagam as características geradas até a camada de saída, que, neste exemplo, realiza o mapeamento não-linear  $f_1$ .

Exemplos de funções de ativação são a função sigmoide, também conhecida como logística

$$\phi(z) = \frac{1}{1 + e^{-z}}, \quad (3.4)$$

a tangente hiperbólica

$$\phi(z) = \tanh(z) \quad (3.5)$$

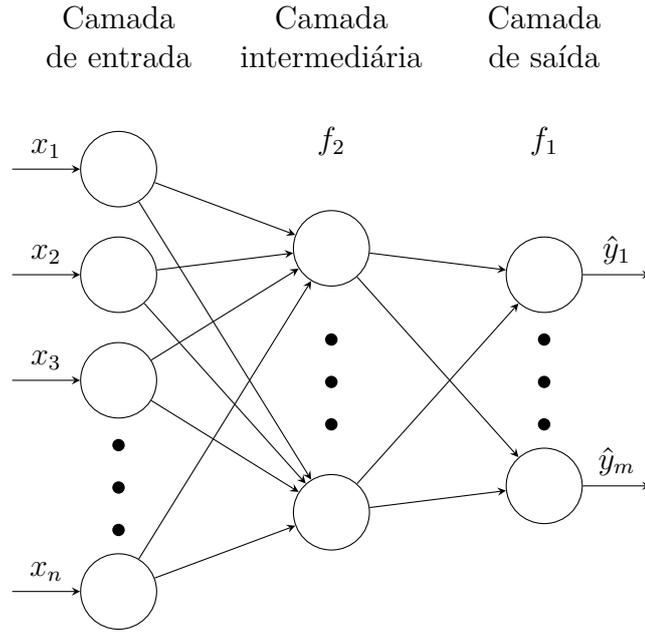


Figura 7 – Representação gráfica do modelo perceptron de múltiplas camadas (MLP).

e a unidade linear retificada (ReLU, do inglês, *Rectified Linear Unit*) (KRIZHEVSKY *et al.*, 2012)

$$\phi(z) = \max(0, z) \tag{3.6}$$

O erro de uma ANN para uma entrada  $x$  avalia o quanto a saída  $\hat{y}$  se aproxima da saída esperada  $y$ . A função de erro, ou função de custo, avalia o erro global, considerando o erro de todas as  $N$  amostras de entrada do conjunto de treinamento. Uma função comumente empregada em problemas de regressão é o erro quadrático médio (MSE, do inglês, *Mean Squared Error*), dado pela Equação 3.7 e uma função comumente empregada para problemas de classificação com duas classes é a entropia cruzada binária (BCE, do inglês, *Binary Cross-Entropy*), dada pela Equação 3.8, onde  $p : \mathbb{R}^n \mapsto \mathbb{R}^n$  é uma função que retorna a probabilidade de  $\hat{y}$  para cada uma das classes, como, por exemplo, o *softmax*, dado pela Equação 3.9.

$$J_{MSE}(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \tag{3.7}$$

$$J_{BCE}(\mathbf{W}) = -\frac{1}{N} \sum_{i=1}^N (y_i \log(p(\hat{y}_i)) + (1 - y_i) \log(1 - p(\hat{y}_i))) \tag{3.8}$$

$$p(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}}, \text{ para } i = 1, \dots, n \text{ e } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^n \tag{3.9}$$

Encontrar um conjunto ótimo de pesos para uma ANN pode ser tratado como um problema de otimização em que se deseja minimizar a função de custo em relação aos

pesos. Dentre os métodos de otimização que podem ser aplicados estão os de primeira e segunda ordem. Cabe enfatizar que estes métodos de otimização são métodos de busca local, assim não há garantia de atingir o ótimo global (GOODFELLOW *et al.*, 2016).

Uma forma de ajustar os pesos de uma ANN, ou seja, de minimizar a função de custo, usando um método de otimização de primeira ordem, é caminhar de maneira iterativa na direção contrária do gradiente (LIANG; BOSE, 1996). Quando este algoritmo usa subconjuntos do conjunto de amostras disponíveis, abordagem conhecida como treinamento em mini-lotes, ele é chamado gradiente descendente estocástico, expresso segundo a Equação 3.10:

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \eta \nabla J(\mathbf{W}_k) \quad (3.10)$$

onde  $\eta$  representa a taxa de aprendizado,  $\nabla(\cdot)$  o operador gradiente e  $J(\mathbf{W}_k)$  a função de custo aplicada a saída correspondente ao subconjunto  $k$  de amostras. Variações deste algoritmo com adaptação da taxa de aprendizado comumente usadas incluem o Adagrad (DUCHI *et al.*, 2011) e o Adam (KINGMA; BA, 2014).

Um algoritmo para ajuste dos pesos de ANNs baseado na ideia de gradientes descendente foi apresentado em Werbos (1974), porém somente em meados dos anos 80 Rumelhart *et al.* (1986) propõem o algoritmo de retropropagação do erro (do inglês, *backpropagation*), muito semelhante ao trabalho de Werbos, demonstrando como obter de forma computacionalmente eficiente o vetor gradiente em redes de múltiplas camadas. No resultado da operação do *backpropagation*, cada elemento do vetor gradiente está diretamente associado a um peso e indica como a variação deste influencia na saída do neurônio (LEE *et al.*, 1993).

## 3.2 Aprendizado Profundo

Arquiteturas de redes profundas derivam desde o modelo proposto por Fukushima (1975), que pode ser considerado como uma rede neural profunda (DNN, do inglês, *Deep Neural Network*). Porém, o termo aprendizado profundo (DL, do inglês, *Deep Learning*) passou a ser amplamente adotado após a publicação de Hinton *et al.* (2006). O DL atingiu um significativo grau de relevância na comunidade científica e junto ao público em geral em 2012, após Krizhevsky *et al.* (2012) reduzirem drasticamente a taxa de erro de classificação na base de dados Imagenet.

Nos últimos anos, o DL tem ganhado atenção pelos resultados que vem obtendo em diferentes áreas como visão computacional, reconhecimento de fala e reconhecimento de texto (HATCHER; YU, 2018). Dois fatores podem ser considerados quando se buscam as razões da expansão de DNNs: o desenvolvimento do *hardware*, com destaque para as unidades de processamento gráfico (GPU, do inglês, *Graphic Processing Units*) para uso

geral (OH; JUNG, 2004) e a massiva disponibilidade de dados para treinamento, como a base de dados Imagenet (DENG *et al.*, 2009), atualizada em 2010 com mais de 14 milhões de imagens distribuídas entre mais de 21 mil classes.

Apesar do nome DNN, arquiteturas profundas formadas por camadas totalmente conectadas continuam sendo MLPs que se diferenciam apenas em sua profundidade. Não há, ainda, uma determinação exata de onde termina uma arquitetura rasa e começa uma arquitetura profunda, sendo a diferença entre elas subjetiva (SCHMIDHUBER, 2015).

### 3.3 Redes Convolucionais

Para falar de DNN neste trabalho, é importante introduzir as redes neurais convolucionais (CNN, do inglês, *Convolutional Neural Network*). Apesar de várias DNNs consideradas relevantes serem construídas com camadas convolucionais (KRIZHEVSKY *et al.*, 2012; HE *et al.*, 2016; SIMONYAN; ZISSERMAN, 2014), seu uso não é regra para determinar uma DNN.

O trabalho de LeCun *et al.* (1989) une a ideia de ANN com camadas de operação de convolução propostas em Fukushima (1975) ao algoritmo *backpropagation* de Rumelhart *et al.* (1986) na tarefa de reconhecimento de números manuscritos. O sucesso nesta tarefa foi um grande passo para o surgimento das CNNs atuais.

As CNNs são um tipo particular de ANN com inspiração neurobiológica na organização do córtex visual dos animais apresentada nos trabalhos de Hubel e Wiesel (1962) e Hubel e Wiesel (1977). Estes trabalhos demonstram como os neurônios corticais, organizados de forma hierárquica, respondem individualmente a estímulos em regiões restritas do campo de visão, conhecidas como campos receptivos. A região ativada depende diretamente do padrão de estímulo apresentado. A sobreposição parcial dos campos receptivos de diferentes neurônios forma o campo de visão completo.

De forma similar à organização do córtex visual, as CNNs são projetadas para reconhecer padrões, sendo favorecidas por dados organizados em grade. Exemplos deste tipo de organização vão desde séries temporais em arranjos de uma dimensão contendo amostras em intervalos regulares de tempo, passando por imagens, com pixels organizados em duas dimensões, podendo chegar a dados organizados em múltiplas dimensões. As CNNs são particularmente bem sucedidas em aplicações práticas de reconhecimento de padrões por serem capazes de implementar soluções que exibem um alto grau de invariância a distorções, como translação, dimensionamento e inclinação (LECUN; BENGIO, 1998).

A operação de convolução, um tipo específico de operação linear, é responsável

por dar nome às CNNs, que podem ser consideradas ANNs nas quais a operação de multiplicação geral de matrizes é substituída pela operação de convolução em pelo menos uma de suas camadas (GOODFELLOW *et al.*, 2016).

### 3.3.1 A Operação de Convolução

Em sua forma mais geral, a operação de convolução entre duas funções ( $f$  e  $g$ ) gera uma terceira função que expressa como a forma de uma é modificada pela outra. A operação de convolução entre  $f$  e  $g$  é escrita como  $f * g$ . Na terminologia de redes convolucionais, o argumento  $f$  da convolução é referido como a entrada, e o argumento  $g$  é referido como o *kernel* ou filtro. A saída da operação de convolução é referida como o mapa de características.

Os dados de entrada de uma CNN podem ser organizados em múltiplas dimensões, variando usualmente de uma a três dimensões, com a possibilidade de usar a operação de convolução em mais de um eixo por vez. O filtro também pode ser representado com múltiplas dimensões tendo, no máximo, dimensão igual ou menor à dimensão do dado de entrada, com seus valores correspondendo aos parâmetros livres ajustados durante o processo de aprendizado.

Como exemplo, para uma imagem  $I$  bidimensional de entrada e filtro  $K$  também bidimensional, a operação de convolução entre eles é representada pela Equação 3.11:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (3.11)$$

Para facilitar a implementação, as bibliotecas computacionais geralmente usam a operação de correlação cruzada, dada pela Equação 3.12, como sinônimo da convolução. Embora a correlação cruzada não apresente a propriedade comutativa, como a operação de convolução, ambas são equivalentes para sinais simétricos (VON ZUBEN; VERONEZE, 2019). Apesar de as bibliotecas computacionais implementarem a correlação cruzada, é comum na área de aprendizado profundo, assim como neste trabalho, se referir a esta operação com o termo convolução.

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (3.12)$$

A Figura 8 representa graficamente a operação de correlação cruzada para uma entrada bidimensional de 5 linhas por 5 colunas e filtro de 3 linhas por 3 colunas. A saída é representada pelo mapa de características. Os destaques em azul e vermelho representam exemplos de operações sobre regiões da entrada com suas saídas correspondentes.

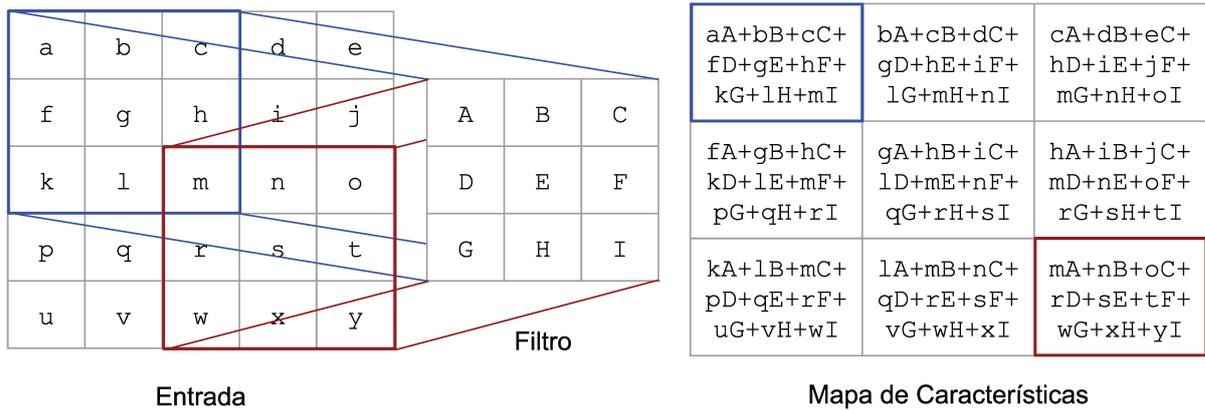


Figura 8 – Representação gráfica da operação de correlação cruzada para uma entrada 2D.

### 3.3.2 Conectividade Esparsa, Campos Receptivos Locais e Compartilhamento de Parâmetros

Uma rede totalmente conectada, como a MLP, opera através de multiplicação de matrizes entre as entradas das camadas e seus parâmetros. Neste tipo de operação, cada neurônio de uma camada interage com a saída de todos os neurônios da camada anterior. Nas CNNs, cada neurônio está conectado somente à saída de um filtro da camada anterior. Adotando filtros de tamanho menor que a entrada, as interações se tornam esparsas, permitindo uma correlação espacial local (LECUN; BENGIO, 1998). A Figura 9 apresenta as conexões de uma MLP e de uma CNN para exemplificar as interações esparsas. Em uma MLP, o neurônio de cor cinza da camada da esquerda afeta todos os neurônios de mesma cor da camada da direita. Já em uma CNN com filtro de tamanho 3, o neurônio de cor cinza da camada da esquerda afeta somente três neurônios, de mesma cor, da camada da direita. O neurônio de cor azul na camada da direita é afetado por todos os neurônios de mesma cor da camada da esquerda em uma MLP e somente por três neurônios, de mesma cor, da camada da esquerda em uma CNN.

À região da entrada demarcada por um filtro se dá o nome de campo receptivo local. Assim, cada campo receptivo local está associado a um único neurônio. Cada deslocamento do filtro pela entrada demarca um novo campo receptivo local e uma associação a um novo neurônio (LECUN; BENGIO, 1998). Na Figura 8, os destaques em azul e em vermelho na entrada representam dois campos receptivos locais e estão associados às saídas de mesma cor no mapa de características.

No cálculo da saída de uma camada em uma rede totalmente conectada, cada parâmetro participa da operação uma única vez, não sendo mais utilizado até que uma nova entrada passe pela camada. Nas CNNs, os filtros compartilham seus parâmetros, assim, quando percorrem a entrada, aprendem os padrões que ocorrem com frequência em diferentes localizações de forma mais robusta (LECUN; BENGIO, 1998).

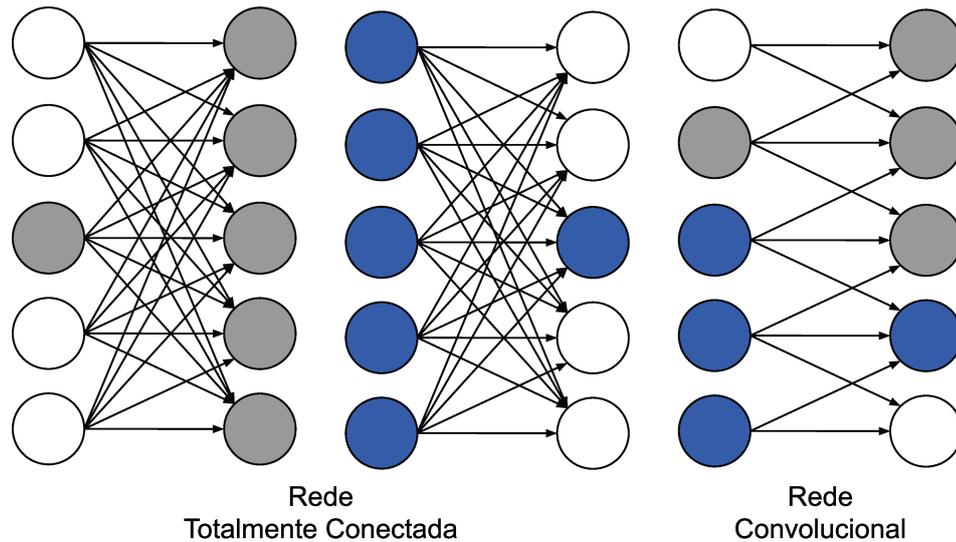


Figura 9 – Representação gráfica da conectividade em redes totalmente conectadas e em redes convolucionais.

A forma particular de compartilhamento de parâmetros em uma camada convolucional faz com que elas tenham uma propriedade interessante, chamada equivariância à translação. Uma função é equivariante quando sua saída é alterada de mesma maneira que a alteração feita em sua entrada. Assim, caso a entrada de uma camada convolucional sofra uma translação por um determinado valor, a mesma translação acontecerá para a saída desta camada convolucional. Isto significa que, em uma função  $f(x)$  equivariante para uma função  $g(x)$ ,  $f(g(x)) = g(f(x))$  (LECUN; BENGIO, 1998).

Segundo LeCun e Bengio (1998), estas três características (conectividade esparsa, campos receptivos locais e compartilhamento de parâmetros) permitem que uma CNN armazene menos parâmetros quando comparadas aos parâmetros armazenados em redes totalmente conectadas, reduzindo o requisito de memória para o modelo e o número de operações para calcular a saída. Assim, a convolução é mais eficiente em comparação com as camadas totalmente conectadas, em termos dos requisitos de memória e eficiência estatística.

### 3.3.3 Agregação

Em uma CNN, é comum inserir, entre sucessivas camadas convolucionais, camadas de agregação (do inglês, *pooling*), também conhecidas como camadas de subamostragem. O objetivo destas camadas é buscar por informações mais relevantes, ou significativas, nas saídas das camadas convolucionais. Segundo Goodfellow *et al.* (2016), camadas de agregação substituem determinadas regiões da saída das camadas convolucionais por uma estatística resumida desta vizinhança.

Uma agregação pode ser definida através de diferentes operações que, quando

aplicadas sobre todos os elementos de uma região, resultam em uma nova saída, geralmente um único valor. Exemplos de operações de agregação são a agregação máxima (do inglês, *max pooling*) (ZHOU; CHELLAPPA, 1988), que retorna o valor máximo da vizinhança, e a agregação média (do inglês, *average pooling*), que retorna a média aritmética da vizinhança.

Independentemente da operação de agregação aplicada, é certo que seu uso ajuda a tornar a representação invariante a pequenas translações, visto que a operação de agregação tende a acontecer dentro da mesma vizinhança, ou pelo menos na maior parte dela, sem alterar bruscamente sua saída (SCHERER *et al.*, 2010). A invariância à translação local pode ser uma propriedade útil quando o mais importante é saber se uma característica está presente na entrada, ao invés de saber onde exatamente ela se encontra.

A operação de agregação resume os valores de uma vizinhança em um único valor, armazenando somente as características mais relevantes, resumindo a entrada em uma saída de tamanho reduzido. Com a redução do número de características, há também a redução do número de parâmetros necessários nas camadas posteriores, melhorando a eficiência computacional da rede e reduzindo os requisitos de memória para armazenar os parâmetros (LECUN; BENGIO, 1998).

## 3.4 Regularização

Segundo Goodfellow *et al.* (2016), regularização é “qualquer modificação que fizermos em um algoritmo de aprendizado que se destina a reduzir seu erro de generalização, mesmo que em detrimento de seu erro de treinamento”. Existem variadas formas de se aplicar regularização: algumas adicionam restrições ao modelo, outras adicionam novas variáveis na função objetivo, outras são ainda projetadas para incluir um conhecimento prévio sobre o problema. As estratégias de regularização variam de problema para problema na tarefa de ajudar na generalização do modelo, sendo necessária uma análise caso a caso.

Por ser um campo amplo de pesquisa, esta Seção descreve algumas das abordagens de regularização para modelos profundos adotadas neste trabalho.

### 3.4.1 Desligamento

A técnica de regularização conhecida como desligamento (do inglês, *dropout*) tem por objetivo desligar, com probabilidade  $p$ , neurônios em uma rede (SRIVASTAVA *et al.*, 2014). Esta técnica é aplicada por camada e determina se um neurônio receberá entradas durante a etapa de treinamento, no passo de propagação dos dados pela rede. Quando um neurônio não participa do passo de propagação dos dados, conseqüentemente, não tem seus parâmetros ajustados.

Na proposta original de Hinton *et al.* (2012), os autores definem um valor fixo de  $p = 0.5$ . Porém, um estudo mais detalhado sobre a influência do desligamento nas camadas das redes convolucionais profundas sugere que este valor seja adaptado para obter um melhor desempenho do modelo (SRIVASTAVA *et al.*, 2014).

Ainda segundo Hinton *et al.* (2012), pelo fato de os neurônios desligados não terem influência nos demais neurônios, pode haver uma redução das coadaptações complexas na rede, fazendo com que o modelo aprenda características de forma mais robusta. Desta maneira, o modelo é pressionado a aprender as características da entrada usando unicamente os neurônios disponíveis. Caso o modelo tenha capacidade de generalização suficiente, este treinamento com a restrição de uso dos neurônios pode melhorar sua generalização.

Uma outra visão possível da técnica de desligamento é fornecer um método poderoso de regularizar derivações de um modelo, tal como um comitê de redes, de forma computacionalmente barata (SRIVASTAVA *et al.*, 2014). Isto só é possível pois o desligamento treina um conjunto que consiste de, praticamente, todas as combinações possíveis de sub-redes formadas pela remoção de neurônios da rede original. Assim, seu efeito seria o de treinar todos os modelos possíveis a partir da rede original, fornecendo uma aproximação computacionalmente barata em relação ao treinamento e avaliação de um conjunto de modelos individuais.

### 3.4.2 Normalização em lotes

A normalização em lotes (do inglês, *batch normalization*) é uma técnica de reparametrização adaptativa para modelos profundos que opera através da inserção de ruído aditivo e multiplicativo durante o treinamento. Esta técnica foi proposta em Ioffe e Szegedy (2015), motivada pela dificuldade de treinamento destes modelos. Segundo Ioffe e Szegedy (2015), adotar a técnica de normalização em lotes pode melhorar o processo de otimização e, em alguns casos, descartar o uso de desligamento.

Durante o treinamento de modelos muito profundos, envolvendo a composição de várias funções ou camadas, o gradiente informa como atualizar cada parâmetro considerando que as outras camadas permanecerão constantes, sem alteração. Na prática todas as camadas são atualizadas simultaneamente, podendo ter consequências inesperadas no treinamento. De forma elegante, a normalização faz a reparametrização destes modelos, reduzindo significativamente o problema da atualização simultânea de várias camadas (GOODFELLOW *et al.*, 2016).

A normalização em lotes pode ser aplicada a qualquer entrada ou camada oculta em uma rede, porém, como é mais provável obter uma distribuição simétrica e não esparsa antes da ativação das camadas (HYVÄRINEN; OJA, 2000). Aplicar a normaliza-

ção em lotes neste ponto pode produzir uma ativação com distribuição estável (IOFFE; SZEGEDY, 2015).

Para aplicar a normalização em lotes a um mini-lote  $\mathbf{H}$  da saída de uma camada, organizado com uma amostra por linha de  $\mathbf{H}$ , usamos:

$$BN_{\gamma,\beta}(\mathbf{H}) = \gamma\mathbf{H}' + \beta \quad (3.13)$$

onde  $\gamma$  e  $\beta$  são parâmetros livres ajustados durante o treinamento do modelo para cada ativação, ou no caso de camadas convolucionais, para cada filtro, e  $\mathbf{H}'$  é dado por

$$\mathbf{H}' = \frac{\mathbf{H} - \mu}{\sigma} \quad (3.14)$$

onde  $\mu$  é a média do mini-lote subtraída de cada uma das amostras de  $\mathbf{H}$  e  $\sigma$  é o desvio padrão do mini-lote que divide cada um dos resultados da subtração. Assim, a operação é aplicada em cada uma das linhas, ou amostras, de  $H$ . As camadas subsequentes operam sobre os dados normalizados exatamente da mesma maneira que operavam sobre  $H$ .

Durante o treinamento do modelo, para um mini-lote com  $m$  amostras, a média é dada por

$$\mu = \frac{1}{m} \sum_i^m \mathbf{H}_i \quad (3.15)$$

e o desvio padrão é dado por

$$\sigma = \sqrt{\frac{1}{m} \sum_i^m (\mathbf{H} - \mu)_i^2}. \quad (3.16)$$

Durante a fase de teste, após o treinamento do modelo, para permitir que o modelo seja avaliado em uma única amostra sem depender de um mini-lote,  $\mu$  e  $\sigma$  podem ser substituídos pelos seus valores médios obtidos durante o treinamento.

É importante para CNNs aplicar a mesma normalização  $\mu$  e  $\sigma$  em cada localização espacial dentro de um mapa de características, para que as estatísticas do mapa de características permaneçam as mesmas, independentemente da localização espacial.

### 3.4.3 Parada Antecipada

A técnica de parada antecipada se refere à ação de interromper o treinamento do modelo antes que ele comece a sobreajustar (BISHOP, 1995). Uma métrica sobre o conjunto de validação é monitorada para decidir quando parar o treinamento. Um hiperparâmetro, conhecido como paciência, determina após quantas iterações sem melhora no valor da métrica o treinamento deve ser interrompido.

Tomando o valor do erro na saída da função de custo como exemplo de métrica, durante o treinamento de modelos com capacidade de representação suficiente, é

comum observar com o passar das iterações o erro no conjunto de treinamento continuar diminuindo, mas o erro no conjunto de validação começar a divergir. Este é um indicativo do início do sobreajuste. Quando o modelo começa a sobreajustar ao conjunto de treinamento, é pouco provável que o erro no conjunto de validação volte a diminuir. Isto significa que o modelo com parâmetros ajustados até o ponto em que os erros começaram a divergir tem maior capacidade de generalização em comparação com qualquer ajuste posterior de parâmetros (SJÖBERG; LJUNG, 1995).

Outra forma de olhar para a técnica de parada antecipada é considerar que ela faz uma busca eficiente por hiperparâmetros. Considerando que o número de iterações total para o treinamento é um hiperparâmetro que deve ser escolhido antes do início do treinamento, a técnica de parada antecipada controla a capacidade efetiva do modelo, determinando quantas iterações podem ser necessárias para seu ajuste nas amostras disponíveis (GOODFELLOW *et al.*, 2016). Assim, mesmo que a escolha inicial do número de iterações para o treinamento do modelo seja muito grande, a técnica de parada antecipada interrompe o treinamento antecipadamente se não houver melhora no valor da métrica no conjunto de validação por um número pré-determinado de iterações.

## 3.5 Transferência de Conhecimento

O conceito de transferência de conhecimento refere-se à situação em que os parâmetros ajustados para um modelo em um ambiente, problema, ou base de dados, são explorados para melhorar a generalização de um outro modelo em outro ambiente, problema, ou base de dados (PAN; YANG, 2009).

A técnica de transferência de conhecimento é comumente aplicada em situações de escassez de dados, visto que o treinamento de modelos profundos tende a sobreajustar em um conjunto de dados de tamanho insuficiente (BENGIO *et al.*, 2013). Aproveitar os parâmetros de um modelo ajustados em uma base suficientemente grande pode contornar este problema.

A transferência de conhecimento pode ainda beneficiar o treinamento de modelos em problemas distintos. Parâmetros de um modelo com uma boa capacidade de generalização em um problema podem passar por um ajuste fino durante uma nova etapa de treinamento, com novos dados, com o objetivo de resolver um problema diferente. O modelo ajustado pode se beneficiar de representações que são úteis para generalizar rapidamente a partir de poucos dados do novo problema (GOODFELLOW *et al.*, 2016).

Os dois usos mais comuns da transferência de conhecimento são (BENGIO *et al.*, 2013):

- Modelos extratores de características. Quando as camadas totalmente conectadas de

um modelo são removidas, as saídas do novo modelo passam a ser as características das amostras, assim, este novo modelo passa a se comportar como um extrator de características. É possível então gerar um novo conjunto de dados de características extraídas de uma base de dados usando um modelo pré-treinado sem suas camadas totalmente conectadas. Este conjunto de características é usado para treinar um novo modelo, menor em comparação com o modelo completo, responsável somente pela classificação ou regressão, dependendo do problema. O tempo de treinamento deste novo modelo costuma ser muito menor comparado com o tempo de treinamento de um modelo completo.

- Ajuste do modelo. Uma outra estratégia possível é usar uma nova base de dados para ajustar os parâmetros de um modelo pré-treinado completo, através de mais algumas iterações de treinamento. Para esta abordagem, é necessário decidir se o modelo será totalmente ajustado ou se somente camadas mais próximas do final do modelo e antes das camadas totalmente conectadas serão ajustadas. Esta decisão se relaciona com a similaridade entre o novo conjunto de dados e o conjunto de dados que deu origem ao modelo pré-treinado. Modelos usados em conjuntos de dados similares terão suas camadas iniciais reconhecendo características em comum. Se os conjuntos de dados forem muito distintos pode ser necessário ajustar todas as camadas do modelo.

Treinar um modelo profundo inicializado com parâmetros aleatórios é custoso em bases de dados muito grandes. Assim, é comum que modelos que foram treinados nestas bases de dados tenham seus parâmetros disponibilizados publicamente. Consequentemente é possível aplicar um modelo pré-treinado a mesma base de dados, ajustá-lo para outros problemas, ou ainda empregá-lo como extrator de características.

Para decidir quando a transferência de conhecimento melhor se encaixa em um novo conjunto de dados, pelo menos dois fatores devem ser analisados: o tamanho do novo conjunto de dados e sua similaridade com o conjunto de dados original. Sabendo que as camadas do início do modelo são mais genéricas e as camadas do final do modelo são específicas para o conjunto de dados original, existem quatro possibilidades (YOSINSKI *et al.*, 2014):

1. O novo conjunto de dados é pequeno e similar ao conjunto de dados original. Como os dados podem não ser suficientes para ajustar o modelo todo e considerando que as camadas do modelo reconheçam características do novo conjunto de dados por ele ser semelhante ao original, ajustar somente as camadas totalmente conectadas pode ser mais prudente e ter melhores resultados.

2. O novo conjunto de dados é grande e similar ao conjunto de dados original. Com dados suficientes e semelhantes aos dados do conjunto original, é possível ajustar o modelo sem se preocupar com ajustes em excesso.
3. O novo conjunto de dados é pequeno e muito diferente do conjunto de dados original. Este acaba sendo o caso mais complexo pois não há dados suficientes para ajustar as camadas totalmente conectadas do modelo e os dados são muito diferentes dos dados originais. Uma opção seria remover as camadas totalmente conectadas e algumas camadas do final do modelo, responsáveis por reconhecer características específicas do conjunto de dados, adicionando um novo modelo, linear ou camadas totalmente conectadas, ao final na tentativa de ajustá-lo.
4. O novo conjunto de dados é grande e muito diferente do conjunto de dados original. Com dados suficientes, pode ser melhor treinar o modelo todo a partir do início. Ainda assim pode ser benéfico inicializar o modelo com parâmetros pré-treinados pois a quantidade de dados do novo conjunto seria suficiente para ajustar o modelo.

Adicionalmente, quando a transferência de conhecimento for a escolha para abordar um novo problema, é importante lembrar que pode ser mais difícil flexibilizar a arquitetura escolhida. O modelo pré-treinado escolhido pode ter restrições de tamanho das entradas, por exemplo. Outro detalhe importante é em relação a taxa de aprendizado. É comum usar taxas de aprendizado menores quando os parâmetros do modelo estão passando por ajustes, já que é esperado que o modelo pré-treinado tenha parâmetros relativamente bons (YOSINSKI *et al.*, 2014).

O uso de transferência de conhecimento pode trazer benefícios como tempo reduzido de treinamento de um modelo, ou a possibilidade de aplicar um modelo pré-treinado a um conjunto de dados diferente do original (GOODFELLOW *et al.*, 2016). Em todo caso, seu uso exige uma cuidadosa análise dos conjuntos de dados e um ajuste minucioso para que funcione da forma esperada.

## 3.6 A Função de Custo Triplet

DNNs que operam com a função de custo *triplet* foram propostas para problemas de classificação por Schroff *et al.* (2015) com o objetivo de separar amostras de diferentes classes. Elas são, até o momento, estado-da-arte para reconhecimento facial, com acurácia de 99.63% na base de dados LFW (do inglês, *Labeled Faces in the Wild*) (HUANG *et al.*, 2008), base de rostos geralmente utilizada para este tipo de avaliação (LEARNED-MILLER *et al.*, 2016).

Nesta combinação de DNN e função de custo *triplet*, a DNN tem o objetivo de criar um mapeamento de suas amostras de entrada em um espaço vetorial euclidiano

de dimensão relativamente pequena e a função de custo *triplet* tem o objetivo de calcular a distância entre as amostras neste espaço. O mapeamento é construído minimizando a função custo *triplet* durante a etapa de treinamento da DNN.

Como amostras de mesma classe tendem a criar agrupamentos e agrupamentos distintos tendem a se distanciar por uma margem mínima, o problema de classificação, quando se usa a função de custo *triplet*, passa a ser de calcular a distância euclidiana entre as entradas da função e determinar se elas pertencem à mesma classe.

A função de custo *triplet* opera sobre uma tripla de entradas enviada à DNN contendo uma amostra âncora,  $x_a$ , uma amostra positiva de mesma classe da âncora,  $x_p$ , e uma amostra negativa de classe diferente da âncora,  $x_n$ . A DNN gera o mapeamento  $f(x_a)$ ,  $f(x_p)$  e  $f(x_n)$  para as três entradas, respectivamente. O erro,  $J$ , para o mapeamento das três entradas é calculado segundo a Equação 3.17.

$$J_{triplet} = \|f(x_a) - f(x_p)\|_2^2 - \|f(x_a) - f(x_n)\|_2^2 + \alpha \quad (3.17)$$

Ao minimizar a função custo *triplet*, minimiza-se também a distância euclidiana entre os mapeamentos de mesma classe  $f(x_a)$  e  $f(x_p)$ , ao mesmo tempo em que se reforça a margem  $\alpha$ , um hiperparâmetro da rede, entre os mapeamentos de mesma classe  $f(x_a)$  e  $f(x_p)$  e o mapeamento de classe distinta  $f(x_n)$ . A Figura 10 exemplifica o efeito da função custo *triplet* durante o treinamento em uma amostra âncora, em azul, uma amostra positiva, em verde, e uma amostra negativa, em vermelho. Durante as iterações do treinamento, a amostra positiva é movida para uma posição mais próxima da âncora em um espaço euclidiano de duas dimensões, enquanto que a amostra negativa é movida para uma posição mais distante da âncora, de forma a reforçar a margem mínima  $\alpha$ .

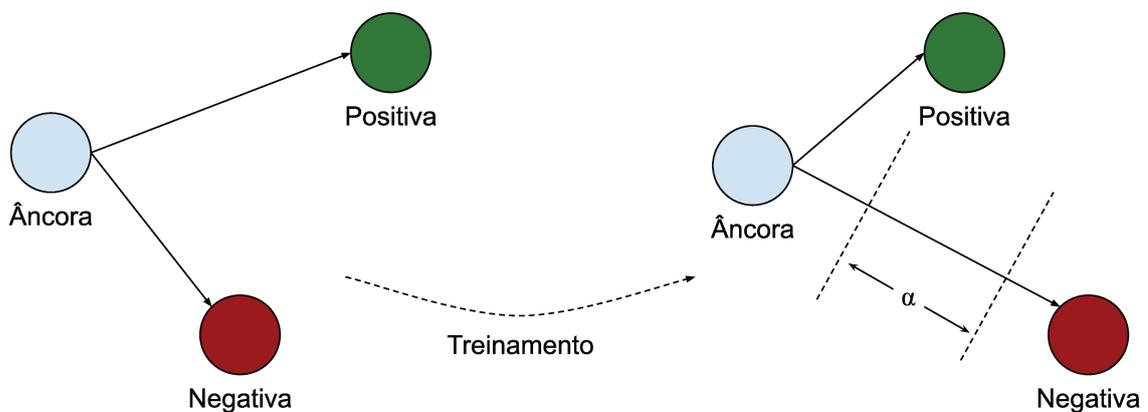


Figura 10 – Representação gráfica de um passo do treinamento usando a função custo *triplet*.

A Figura 11 ilustra a operação da arquitetura proposta por Schroff *et al.* (2015)

para o problema de reconhecimento facial. Nela, uma imagem de entrada do rosto de uma pessoa é definida como amostra âncora, uma outra imagem do mesmo rosto em posição diferente é definida como amostra positiva e uma terceira imagem do rosto de uma pessoa diferente é definida como amostra negativa. As três imagens passam pela mesma DNN, gerando um mapeamento para cada. A função custo *triplet* calcula o erro a partir dos três mapeamentos gerados e só então os pesos da DNN são ajustados.

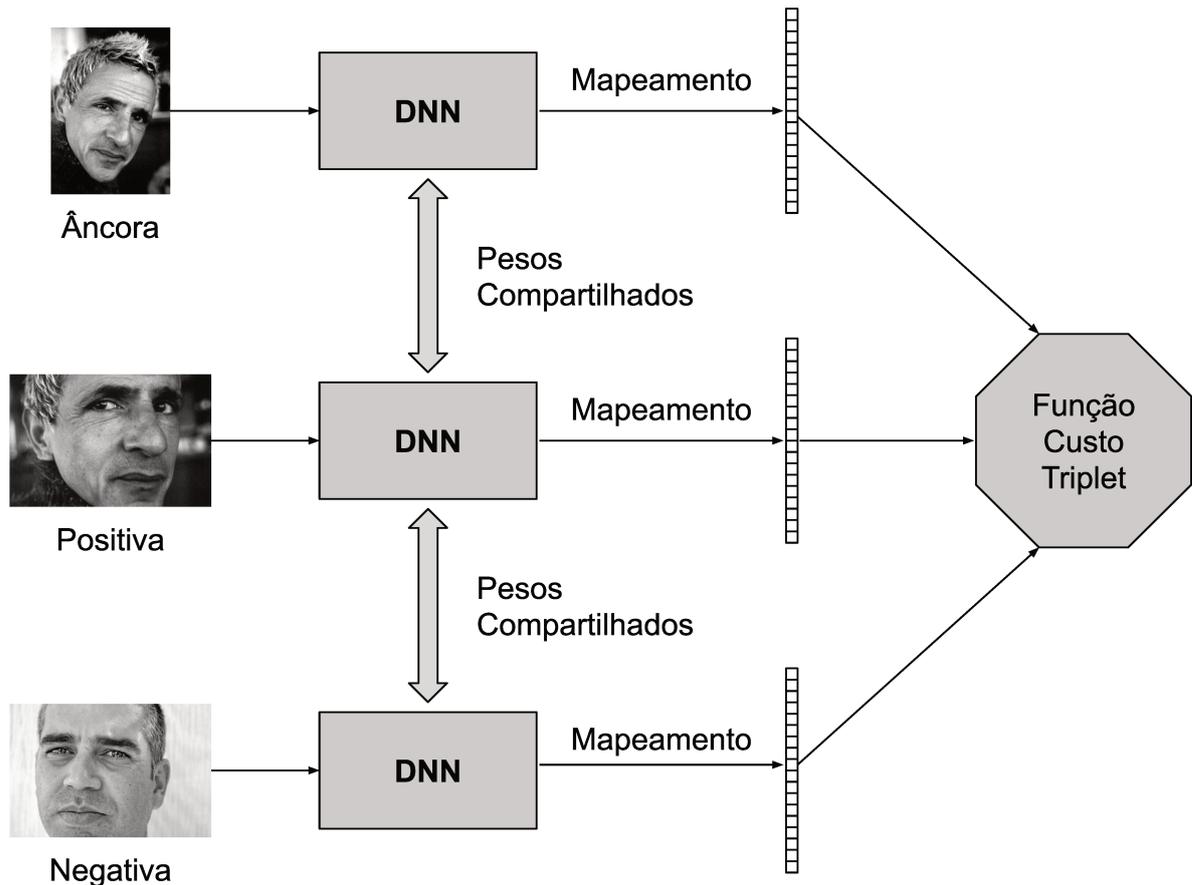


Figura 11 – Representação gráfica da operação da arquitetura DNN usando a função custo *triplet* apresentada por Schroff *et al.* (2015).

Após a etapa de treinamento, a classificação de uma nova amostra de entrada é determinada medindo-se a distância euclidiana quadrática média entre todos os mapeamentos gerados e assumindo que o menor valor de distância quadrática média obtido entre o mapeamento da amostra de entrada e o mapeamento das amostras de outras classes corresponde à classe a qual a amostra de entrada pertence.

Uma das formas de gerar o mapeamento descrito acima é através de DNNs convolucionais, com camadas altamente regularizadas e com normalização L2 na camada de saída, garantindo que  $\|f(x)\|_2 = 1$  (SCHROFF *et al.*, 2015).

## 4 Trabalhos Relacionados

O interesse por empregar DNNs, em especial as convolucionais (DCNNs, do inglês, *Deep Convolutional Neural Networks*), no problema de classificação de sinais SSVEP tem aumentado nos últimos anos. Os trabalhos revisados aqui concentram-se, principalmente, nos anos de 2018 e 2019. Os resultados divulgados por estes trabalhos se mostram promissores para o problema abordado.

O trabalho de Cecotti e Graeser (2008) pode ser considerado como o pioneiro a aplicar CNN ao problema de classificação de sinais SSVEP. A proposta dos autores é de uma arquitetura convolucional de 4 camadas com uma das camadas sendo a operação da transformada de Fourier. Esta arquitetura recebe como entrada sinais brutos de 6 eletrodos, privilegiando a região do córtex visual, em janelas de 1 segundo normalizadas pelo escore padronizado. A camada de entrada da arquitetura simplesmente propaga a janela do sinal, a primeira camada intermediária aplica a operação de convolução com 10 filtros de tamanho 5. Os mapas de características passam, então, pela segunda camada intermediária responsável por aplicar a operação da transformada de Fourier. Da saída da transformada de Fourier, somente as amplitudes das frequências na faixa das classes e suas harmônicas superiores são propagadas para uma camada de 100 neurônios, totalmente conectados com a camada anterior, que propaga o sinal para a camada de saída, com 6 neurônios representando 5 frequências e uma classe de transição. Todas as camadas usam a tangente hiperbólica como função de ativação. Para que o sistema seja confiável, os autores também adotam uma abordagem de rejeição, com um limiar definido para cada uma das classes. Os resultados obtidos pelos autores sobre 2 sujeitos não são suficientes para generalizar sua proposta. A classificação correta das classes é de 51,71% na média e a rejeição da classificação é de 46,18% na média, para os 2 sujeitos. Mesmo com a taxa de classificação baixa, a taxa de rejeição ajuda a tornar o sistema mais confiável.

Um dos autores do trabalho anterior sugere algumas modificações àquela arquitetura e expande a avaliação para 10 sujeitos em Cecotti (2011). A arquitetura proposta adiciona uma camada convolucional ao seu início, altera o número de filtros de 10 para 6, o tamanho dos filtros de 5 para 128 e troca a função de ativação das camadas totalmente conectadas pela sigmoide. Para este trabalho, o autor abandona a abordagem de rejeição da classificação e a classe de transição. São consideradas 5 frequências e mais duas harmônicas superiores para cada uma das frequências. O desempenho médio da arquitetura proposta para os 10 sujeitos é de 95,61%, melhorando drasticamente os resultados em relação ao trabalho anterior.

Nos dois trabalhos anteriores, os autores avaliam o desempenho de classificação

da arquitetura CNN proposta treinando a rede para os dados de cada um dos sujeitos. Em Cecotti (2017), o autor compara o treinamento de arquiteturas CNNs individualmente, por sujeito, com arquiteturas CNNs treinadas com os dados de um conjunto de sujeitos e o desempenho de classificação avaliado em um novo sujeito que não participou do treinamento. São definidas 6 arquiteturas, sendo a primeira uma ANN com uma camada de entrada e uma de saída, representando um modelo linear, a segunda uma MLP com uma camada intermediária de 10 neurônios, a terceira uma CNN de duas camadas convolucionais com 8 filtros de tamanho 64 e 16 filtros de tamanho 48, respectivamente, e as três últimas arquiteturas sendo CNNs de uma camada convolucional variando o número de filtros de tamanho 64 entre 4, 8 e 16. Todas as camadas usaram a função de ativação ReLU. Foram usadas somente 2 classes. Cada arquitetura foi treinada e avaliada em cada um dos 16 sujeitos, assim como treinada com os dados de 14 sujeitos e avaliada em outros 2 sujeitos. As médias das arquiteturas treinadas por sujeito ficaram abaixo das médias das arquiteturas treinadas com os dados de um conjunto de sujeitos, demonstrando que as CNNs têm capacidade de generalização.

Em Bevilacqua *et al.* (2014), os autores usam a mesma CNN proposta em Cecotti (2011), porém sem considerar as harmônicas superiores dos sinais como entrada, para controlar o deslocamento por um ambiente virtual através de sinais SSVEP. No experimento, um sujeito usando um óculos de realidade virtual precisa se deslocar entre dois pontos em um ambiente virtual. Três estímulos em diferentes frequências estão disponíveis no campo de visão do sujeito, indicando a direção em que ele deve se deslocar no ambiente. São usados 5 eletrodos para aquisição do sinal cerebral, posicionados na região occipital. O sinal de entrada é dividido em janelas de 2 segundos e normalizados pelo escore padronizado. Com um desempenho médio da CNN de 87,5%, os 3 participantes do experimento foram capazes de se deslocar entre os dois pontos em 333 segundos em média.

No trabalho de Thomas *et al.* (2017), os autores comparam 6 algoritmos de classificação: k vizinhos mais próximos (k-NN, do inglês *k-Nearest Neighbors*), árvores de decisão, MLP, Máquina de Vetor-Suporte (SVM, do inglês, *Support-Vector Machine*), CNN e Redes Neurais Recorrentes (RNN, do inglês, *Recurrent Neural Networks*). A base de dados utilizada é uma base pública com 11 sujeitos e 5 frequências. A avaliação é feita sobre os sinais brutos, deixando os dados de um sujeito fora do treinamento e testando o algoritmo treinado nos dados deste sujeito. A arquitetura da CNN é composta por uma camada de convolução com 16 filtros de tamanho 4, função de ativação ReLU, desligamento, uma camada de agregação máxima, função de ativação ReLU e uma camada totalmente conectada de 2000 neurônios com ativação ReLU. O desempenho da arquitetura convolucional supera o desempenho dos outros algoritmos de classificação avaliados com uma média de 69,03%, cabendo destacar que dos dados de 11 sujeitos, a CNN obteve acurácia acima de 90% para 6 deles.

Em Kwak *et al.* (2017), os autores compararam o desempenho de três algoritmos frequentemente empregados em BCI SSVEP, a Análise de Correlação Canônica (CCA, do inglês, *Canonical Correlation Analysis*), o Índice de Sincronização Multivariado (MSI, do inglês, *Multivariate Synchronization Index*) e o CCA-kNN, com o desempenho de duas CNNs e uma ANN propostas no trabalho para controlar um exoesqueleto em ambiente ambulatorial. No primeiro teste, os sujeitos ficaram parados enquanto os estímulos foram apresentados em um óculos com 5 conjuntos de LEDs. No segundo ambiente, os estímulos foram apresentados da mesma forma, porém o exoesqueleto ficou livre para se movimentar, aumentando os ruídos na entrada dos algoritmos. Foram usados 8 eletrodos para aquisição do sinal cerebral posicionados na região occipital. As duas arquiteturas CNN propostas tem 2 camadas convolucionais com 8 filtros e tamanhos 120 e 110, respectivamente. As arquiteturas se diferem no número de camadas totalmente conectadas. A primeira arquitetura tem apenas a camada de saída com 5 neurônios e a segunda arquitetura tem uma camada com 3 neurônios e a camada de saída com 5 neurônios. A função de ativação para todas as camadas é a sigmoide. Os algoritmos foram treinados para cada um dos sujeitos e a CNN com 3 camadas foi a que obteve a melhor média dentre todos os algoritmos avaliados, com 99,28% para o ambiente estático e 94,03% para o ambiente em que o exoesqueleto se movimentou.

Em Aznan *et al.* (2018), os autores avaliam uma arquitetura CNN nos sinais brutos de SSVEP. A arquitetura é composta por uma camada convolucional com 16 filtros de tamanho 10 com deslocamento de 4, uma camada de normalização do lote e uma camada de agregação máxima, seguida por camadas totalmente conectadas compostas por uma camada de 600 neurônios e uma camada de saída de 4 neurônios representando as 4 frequências usadas. A função de ativação das camadas é a ReLU. As entradas da rede são representadas por janelas de 3 segundos. A arquitetura treinada para um sujeito individualmente resultou em 96% de acurácia. Ao treinar a arquitetura com os dados dos 3 sujeitos e tentar classificar os dados de um quarto sujeito a rede não obteve uma boa acurácia ficando em 59%. Desta forma, os autores aumentaram a profundidade da CNN adicionando mais 4 camadas convolucionais com 32, 64, 128 e 256 filtros de mesmo tamanho acompanhadas da normalização do lote e de uma camada de agregação máxima. Esta nova arquitetura treinada com os dados de 3 sujeitos e avaliadas em um quarto sujeito obteve acurácia de 69%.

A arquitetura de uma camada convolucional descrita anteriormente é usada pelos mesmos autores em Aznan *et al.* (2019) para controlar um robô. No experimento proposto, os sujeitos visualizam no monitor a imagem capturada pela câmera do robô. Nesta imagem, a área dos objetos detectados por uma segunda CNN é ocupada por retângulos oscilando em diferentes frequências. O sujeito pode direcionar sua visão para um dos retângulos, enviando um comando ao robô para que ele se dirija até o objeto ou movimentar o robô pelo ambiente direcionando a visão para setas oscilando fora da

imagem. As CNNs que controlam a classificação dos sinais SSVEP foram treinadas para cada um dos sujeitos antes de usar o sistema de navegação do robô. A acurácia média para 3 sujeitos controlando o robô em tempo real ficou em 85%.

Uma abordagem para gerar dados sintéticos de SSVEP usando redes generativas é proposta por Aznan *et al.* (2019). Os dados sintéticos são gerados a partir da base de dados criada pelos autores em Aznan *et al.* (2019). Os dados originais e os dados sintéticos são usados para treinar a arquitetura de uma camada convolucional proposta pelos autores em Aznan *et al.* (2018). Aplicando a transformada de Fourier aos sinais sintéticos, os autores demonstram que estes sinais conseguem replicar um sinal de SSVEP original verificando a semelhança entre eles. Os resultados mostram que o uso dos dados sintéticos em conjunto com os dados originais ajuda na convergência da CNN durante o treinamento, melhorando o desempenho delas tanto no treinamento para os dados de cada um dos sujeitos, individualmente, quanto no treinamento com dados de vários sujeitos e o teste em dados de um sujeito que ficou de fora do treinamento.

A arquitetura híbrida apresentada por Attia *et al.* (2018a) usa uma CNN para extrair as características do sinal bruto de 8 eletrodos, uma rede recorrente do tipo memória de curto e longo prazo (LSTM, do inglês, *Long Short-Term Memory*) para correlacionar as características e uma rede totalmente conectada para classificar o sinal dentre 4 frequências. A CNN usa 24 filtros convolucionais de tamanho 3 e a rede recorrente usa 128 unidades. A função de ativação usada nas redes é a tangente hiperbólica. O sinal de entrada da rede representa uma janela de 2 segundos. Os dados de 4 sujeitos são avaliados com a arquitetura híbrida e o algoritmo CCA. Os resultados da arquitetura híbrida tiveram um desempenho médio 6% maior em comparação ao CCA.

Inspirados pelo desempenho de DCNNs em bases de dados de imagens, em Attia *et al.* (2018b) os autores transformam os sinais brutos de SSVEP em espectrogramas em escala de cinza a partir de janelas de 2 segundos do sinal e usam uma DCNN inspirada na arquitetura conhecida como VGG de Simonyan e Zisserman (2014), para classificar os sinais. A arquitetura consiste de uma camada convolucional de 2.048 filtros com tamanho 3, uma camada convolucional de 128 filtros de tamanho 3, uma camada totalmente conectada com 32.768 neurônios seguida por 256 neurônios, 64 neurônios, 16 neurônios e a camada de saída com 4 neurônios. Todas as camadas têm função de ativação ReLU. Os espectrogramas são gerados dos sinais de 8 eletrodos. A arquitetura é treinada para cada um dos 4 sujeitos. A acurácia média da CNN entre os sujeitos foi de 94.67%. As 4 frequências escolhidas representam 4 direções e são usadas para controlar um robô remotamente. Os sujeitos têm no monitor a imagem capturada pela câmera do robô e 4 conjuntos de LEDs distribuídos ao redor do monitor que oscilam nas 4 frequências relacionadas à direção em que o robô pode seguir.

Uma CNN de 2 camadas convolucionais, ambas com 96 filtros, porém com

tamanhos 256 e 16, respectivamente, e camada totalmente conectada de saída de 12 neurônios é proposta em Waytowich *et al.* (2018) para classificar sinais SSVEP para 12 frequências representando botões em um teclado de 4 linhas por 3 colunas. A arquitetura usa função de ativação unidade linear exponencial (ELU, do inglês, *Exponential Linear Unit*), desligamento e normalização do lote após suas camadas convolucionais. A arquitetura foi treinada para cada um dos 10 sujeitos avaliados, usando os dados de 8 eletrodos divididos em janelas de 1 segundo. A acurácia média da CNN entre os 10 sujeitos foi de 80%, maior quando comparada com o CCA que obteve 60% de acurácia na média entre os 10 sujeitos. Os autores também demonstram como a arquitetura proposta é capaz de criar agrupamentos representando as frequências, assim como decompor os sinais, de forma similar à transformada de Fourier.

Dois trabalhos demonstram que é possível construir um teclado virtual utilizando DCNN, Nguyen e Chung (2018) demonstra um teclado virtual de 58 caracteres e Podmore *et al.* (2019) um teclado virtual com 40 caracteres.

Na proposta de teclado virtual de Nguyen e Chung (2018), os autores definem uma DCNN que opera sobre os espectrogramas das janelas de 2 segundos dos sinais. Os 58 caracteres são divididos em 5 quadrantes na tela e oscilam em frequências diferentes. A DCNN classifica, desta forma, somente 5 frequências e não uma frequência para cada caractere. A DCNN é composta por 2 camadas convolucionais com 10 filtros de tamanho 5, ambas seguidas por uma camada de agregação máxima. As camadas totalmente conectadas contêm uma camada de 128 neurônios, seguida por uma camada de 3 neurônios e a camada de saída com 5 neurônios. As camadas convolucionais e a primeira camada totalmente conectada usam a função de ativação ReLU e a segunda camada totalmente conectada usa função de ativação tangente hiperbólica. Para avaliação em tempo real, a DCNN obteve um desempenho médio entre os 8 sujeitos avaliados de 97,37%.

Na proposta de teclado virtual de Podmore *et al.* (2019), os autores definem uma DCNN com 5 camadas convolucionais. Todas as camadas convolucionais são compostas por 100 filtros. O tamanho dos filtros da primeira camada convolucional é de 10 por 30. Para as outras camadas convolucionais, os filtros têm tamanho 30. Todas as camadas convolucionais são seguidas por desligamento, normalização do lote, função de ativação ReLU e agregação máxima. A camada totalmente conectada é composta somente pela saída de 40 neurônios representando as 40 frequências. Os sinais de 10 eletrodos são divididos em janelas de 1,5 segundos e 5 segundos em duas avaliações distintas. A arquitetura é treinada com dados de 30 sujeitos e avaliada em dados de 5 sujeitos de um total de 35 sujeitos. A rede alcança uma acurácia média de 86,19% com janelas de 5 segundos e 75,64% com janelas de 1,5 segundos. Os autores também avaliaram se a rede pré-treinada permitia um ajuste fino com os dados dos usuários que não participaram do treinamento. Para 2 sujeitos, das 6 sessões disponíveis, 4 foram usadas para o ajuste fino da rede, 1

para validação e 1 para teste. Para os dados do primeiro sujeito, o desempenho da DCNN foi de 75% para 90%. Para os dados do segundo sujeito, o desempenho da DCNN foi de 35% para 70%.

Três trabalhos demonstram o uso de sinais brutos de SSVEP como uma forma de biometria para autenticação de pessoas. Em Puengdang *et al.* (2019) os autores definem uma rede recorrente do tipo LSTM com 100 unidades e função de ativação sigmoide para identificar uma frequência em dados de 20 sujeitos. A arquitetura é treinada para cada um dos sujeitos. A média de falsa rejeição do modelo para os 20 sujeitos ficou em 10,53% e a média de falsa aceitação ficou em 6,58%. Em Yu *et al.* (2019) os autores definem uma CNN com 2 camadas convolucionais. Ao final, a rede classifica o sinal de entrada como de um sujeito autorizado ou não autorizado. A arquitetura é treinada com dados de 8 sujeitos, separando alguns dos dados para teste. A média de falsa rejeição do modelo para os 8 sujeitos ficou em 3,15% e a média de falsa aceitação ficou em 0,06%. Já em El-Fiqi *et al.* (2018) os autores definem uma DCNN com 4 camadas convolucionais. Ao final, a rede classifica a qual sujeito pertence o sinal de entrada. A arquitetura é treinada com duas bases de dados diferentes, uma com 4 sujeitos e outra com 10 sujeitos. São separados 80% dos dados para treinamento e 20% dos dados para teste em cada uma das bases. A média de falsa rejeição do modelo para a base de dados com 4 sujeitos ficou em 1,95% e a média de aceitação positiva ficou em 98,27%. A média de falsa rejeição do modelo para a base com 10 sujeitos ficou em 1,11% e a média de aceitação positiva ficou em 95,91%

Uma das dificuldades em BCI SSVEP é detectar quando o sujeito não está com sua visão focada nos estímulos, ou seja, quando o sujeito não deseja executar qualquer dos comandos disponíveis naquele momento. Para tentar resolver este problema, Zhang *et al.* (2019) usam duas abordagens diferentes. Na primeira, define-se uma CNN com uma classe a mais, de controle não intencional. Na segunda, define-se uma CNN que classifica o sinal como comando intencional ou não. Caso o sinal seja de um comando intencional, este sinal é encaminhado para o algoritmo CCA para ser classificado entre as frequências disponíveis. As redes classificam espectrogramas do sinal. Os autores usaram 4 frequências para comandos e avaliaram 10 sujeitos, treinando as redes para cada um dos sujeitos. As redes propostas pelos autores se mostraram mais eficientes para classificar os comandos não intencionais, aumentando a média de acertos nas 4 frequências disponíveis. O desempenho médio da primeira abordagem foi de 89,76% enquanto da segunda abordagem foi de 90,09%.

## 5 Redes Convolucionais Profundas Aplicadas a Sinais SSVEP

As pesquisas na área de BCI se diversificam pelas etapas de seu ciclo (WOLPAW *et al.*, 2002). Tradicionalmente, na etapa de processamento do sinal, há um grande esforço para melhorar as técnicas de extração e seleção de características com a perspectiva de melhora no desempenho do algoritmo de classificação (CARVALHO *et al.*, 2015a). Este esforço evidencia a atenção especial à etapa de extração de características do sinal requerida por parte das BCIs.

A capacidade intrínseca das DCNNs de extrair características distintas dos dados de forma automática (PATTERSON; GIBSON, 2017), dispensando, frequentemente, as técnicas manuais de extração de características, pode beneficiar a etapa de processamento do sinal em BCIs. Desta forma, como mencionado anteriormente, este trabalho tem como objetivo principal explorar as DCNNs na tarefa de processamento de sinais cerebrais em BCIs SSVEP, substituindo parte ou até mesmo todas as subetapas do processo. Com a finalidade de segmentar a exploração do objetivo principal, são apresentadas diferentes propostas relacionadas ao domínio de DL.

A primeira proposta se relaciona ao formato do dado e tem como objetivo investigar se formatos diferentes influenciam o desempenho das DCNNs. Para tal, são geradas três novas bases de dados a partir dos dados de SSVEP descritos na Seção 5.1. A primeira base de dados é composta por imagens de espectrogramas, abordagem similar à usada em Attia *et al.* (2018b), a segunda base de dados é composta pelo módulo da transformada de Fourier de tempo curto (STFT, do inglês *Short-Time Fourier Transform*), abordagem similar à usada em Cecotti e Graeser (2008), e a terceira base de dados é composta pelo sinal bruto, abordagem similar à usada em Aznan *et al.* (2018), Waytowich *et al.* (2018) e Podmore *et al.* (2019). Os espectrogramas e o módulo da STFT representam a mesma informação, porém o primeiro é composto por valores de pixels e o segundo por valores de amplitudes. A Seção 5.2 apresenta os passos detalhados para gerar as três novas bases de dados.

Arquiteturas DCNN têm sido aplicadas ao problema de classificação de dados de SSVEP e obtido sucesso (KWAK *et al.*, 2017; NGUYEN; CHUNG, 2018; PODMORE *et al.*, 2019). Logo, a segunda proposta se relaciona à construção de novas arquiteturas DCNN para os diferentes formatos de dados. Os desempenhos destas arquiteturas são comparados aos desempenhos de algumas técnicas usualmente empregadas em BCIs. As arquiteturas propostas são detalhadas na Seção 5.3.

Apesar de a técnica de transferência de conhecimento ser usualmente empregada em diferentes problemas de classificação (YOSINSKI *et al.*, 2014; GOODFELLOW *et al.*, 2016), até o momento não há reportado na literatura um trabalho que aplique a técnica aos dados de SSVEP. Assim, a terceira proposta é investigar a possibilidade de aplicar a técnica de transferência de conhecimento em dados de SSVEP. Para esta investigação, foi adotada a arquitetura ResNet (HE *et al.*, 2016). Esta arquitetura conta com 5 variações no número de camadas cobrindo uma ampla faixa de profundidade. Os detalhes desta arquitetura estão descritos na Seção 5.4.

Comumente, as propostas que envolvem o treinamento de DCNNs para dados de SSVEP usam a função de custo da entropia cruzada (AZNAN *et al.*, 2018; EL-FIQI *et al.*, 2018; NGUYEN; CHUNG, 2018; WAYTOWICH *et al.*, 2018) e Zhang *et al.* (2019). Investigar se outra função de custo influencia no desempenho das DCNNs é o objetivo da quarta proposta. A função de custo *triplet*, estado-da-arte para o problema de reconhecimento facial (SCHROFF *et al.*, 2015), é avaliada ao tratar o problema de classificação de dados de SSVEP. A função de custo *triplet* está descrita na Seção 3.6 e a proposta para seu uso está detalhada na Seção 5.5.

Como já mencionado, é habitual ajustar as BCIs individualmente aos usuários, de forma que, para cada novo usuário do sistema, seja necessário um tempo dedicado ao seu treinamento (CARVALHO *et al.*, 2015a). Desta forma, a quinta e última proposta é investigar a viabilidade de conceber um esquema de treinamento mais robusto às DCNNs, que se beneficie do treinamento sobre uma base de múltiplos usuários e reduza a necessidade de treinamento a um mínimo para novos usuários, abordagem similar as usadas em Cecotti (2017), Aznan *et al.* (2018) e Podmore *et al.* (2019).

## 5.1 Descrição dos Dados de SSVEP

Para este trabalho não foram feitas aquisições de dados. Os dados adquiridos para o trabalho de Carvalho *et al.* (2015b) foram empregados nos experimentos descritos no Capítulo 6. Uma breve descrição do protocolo de aquisição é apresentada nesta Seção.

A aquisição contou com 10 voluntários saudáveis em um experimento de SSVEP aprovado pelo Comitê de Ética da Unicamp (CAAE 0617.0.146.000-10). Conforme descrito em Carvalho *et al.* (2015b), todos os voluntários foram adequadamente esclarecidos sobre a pesquisa e o protocolo experimental, e assinaram o termo de consentimento. O protocolo de aquisição dos dados consistiu na visualização de estímulos visuais repetitivos (EVRs) nas frequências de 6; 7, 5; 12; 15; 20 e 30Hz. Foram coletadas 8 sessões de 12 segundos para cada uma das frequências, mais o período de repouso, somando 7 estados no total. A interface de estimulação apresentava dois EVRs quadrados de 3,8cm, em padrão xadrez, um à esquerda e outro à direita em um fundo preto. A taxa de atualização do

monitor era de 60Hz.

Durante as sessões, os dados de EEG foram capturados com o amplificador de sinais biológicos g<sup>®</sup>.USBamp e o sistema de eletrodos secos g<sup>®</sup>.SAHARAsys. Juntos eles possibilitam o registro de até 16 derivações eletrofisiológicas. Os eletrodos foram posicionados privilegiando a região occipital em O1, O2, Oz, PO3, PO4, PO7, PO8, POz, P1, P2, Pz, CPz, C1, C2, Cz, FCz, de acordo com o sistema internacional 10-10. A distribuição dos eletrodos no escalpo pode ser vista na Figura 12. Os sinais cerebrais foram digitalizados a uma taxa de amostragem de 256Hz com 24 bits para quantização. Durante a aquisição, foi aplicado um filtro passa-banda do tipo Butterworth de oitava ordem no intervalo de 5 a 60 Hz e um filtro do tipo *notch* de 58 a 62 Hz para eliminar a interferência da rede elétrica. Uma descrição detalhada do protocolo de aquisição está disponível em Carvalho *et al.* (2015b).

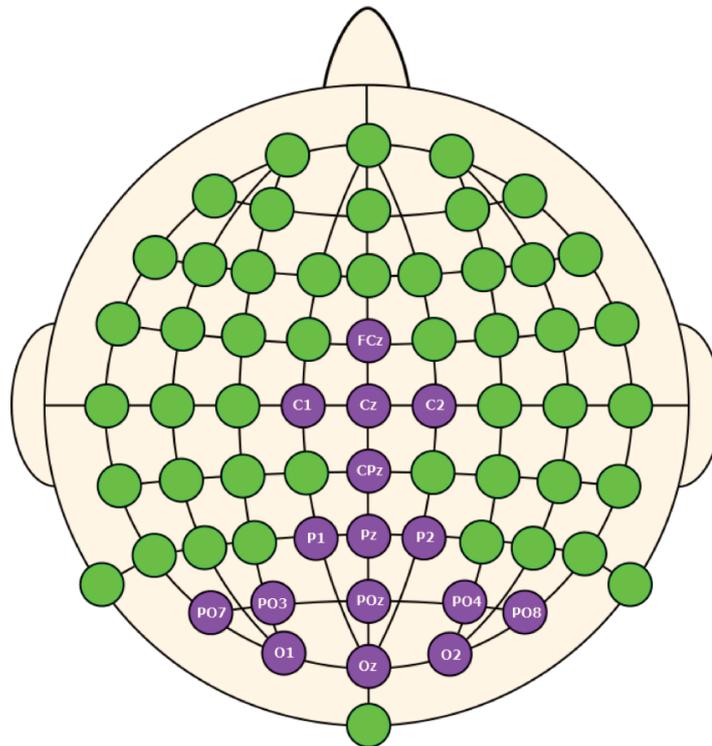


Figura 12 – Disposição dos eletrodos no escalpo (CARVALHO *et al.*, 2015b).

## 5.2 Procedimento para Gerar as Bases de Dados em Diferentes Formatos

No formato original, os dados estavam divididos em 10 pastas, uma para cada sujeito. Cada pasta continha 56 arquivos, totalizando as 8 sessões. Cada sessão contava com 6 arquivos, um para cada uma das frequências mais um arquivo com o repouso. Cada

um dos arquivos continha 12 segundos de aquisição. O nome do arquivo era formado pela frequência, o número do sujeito e o número da sessão.

As frequências dos sinais de SSVEP, representando as classes, adotadas para os experimentos foram as de 12Hz e 15Hz para um estudo mais detalhado do comportamento das arquiteturas de DCNN propostas na tarefa de classificação destes sinais. Os dados dos 16 eletrodos foram espacialmente filtrados pelo CAR, descrito na Seção 2.4. Cada uma das 8 sessões de 12 segundos dos 10 sujeitos foi dividida em janelas de 3 segundos, com 1 segundo de deslocamento. Por conseguinte, cada janela tem 2 segundos de sobreposição dos dados conforme ilustra a Figura 13. Esta divisão tem como objetivo simular um sistema BCI durante a aquisição de dados em tempo real, sendo uma abordagem frequentemente aplicada (CARVALHO *et al.*, 2015b; ATTIA *et al.*, 2018b; AZNAN *et al.*, 2018; NGUYEN; CHUNG, 2018; WAYTOWICH *et al.*, 2018; PODMORE *et al.*, 2019). Nesta situação, uma janela de dados de 3 segundos disponível é classificada a cada segundo.

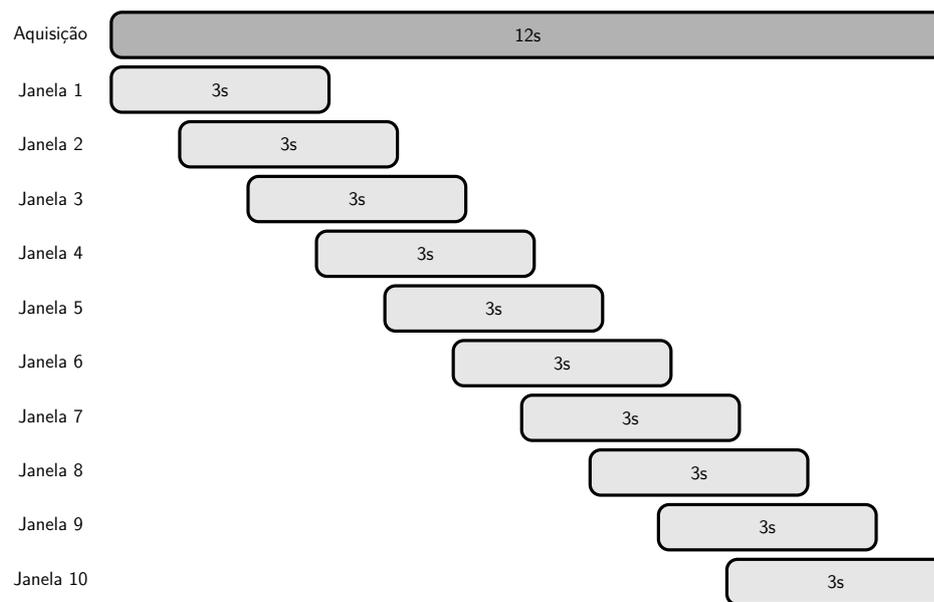


Figura 13 – Exemplo da divisão de uma sessão de 12 segundos em janelas de 3 segundos.

Dividir as sessões de 12 segundos em janelas de 3 segundos gerou 1.280 janelas para cada uma das frequências (10 janelas para cada um dos 16 eletrodos, para cada uma das 8 sessões). Considerando 10 sujeitos e 2 frequências de interesse, no total foram geradas 25.600 janelas. As janelas foram, então, organizadas em 4 bases de dados:

- **base de dados dos módulos da STFT.** Em cada uma das janelas de dados foi aplicada a STFT com janelas de 1 segundo e deslocamento de meio segundo como parâmetro. De todas as frequências de saída da STFT, somente as frequências de interesse e sua vizinhança foram adicionadas à amostra final, concatenando os dados da frequência de interesse e uma faixa de 1Hz abaixo e 1Hz acima. O módulo da

saída da STFT foi convertido para decibéis. Todas as amostras foram interpoladas pela técnica do vizinho mais próximo na proporção de 1:2 no eixo das frequências. As amostras ficaram com a dimensão final de 12 linhas por 7 colunas com os valores representando o módulo da STFT.

- **base de dados de espectrogramas sem harmônica superior do sinal.** Esta base foi gerada a partir da base de dados anterior, com cada uma das amostras transformada em uma imagem de espectrograma em escala de cinza. As amostras ficaram com a mesma dimensão das amostras da base de dados anterior, porém os valores representam pixels entre 0 e 255. A exceção é para o experimento de transferência de conhecimento, no qual as amostras foram geradas com 224 linhas por 224 colunas, pois a arquitetura exige entradas com, no mínimo, este tamanho.
- **base de dados de espectrogramas com uma harmônica superior do sinal.** Esta base de dados foi gerada com o mesmo procedimento da base anterior, mas a cada uma das amostras também foi concatenada uma harmônica superior para as frequências de interesse. A dimensão final de cada amostra é de 24 linhas por 7 colunas com valores representando pixels entre 0 e 255.
- **base de dados de sinais brutos.** Esta base contempla a saída da operação de janelamento, contendo os dados brutos do sinal. A dimensão final de cada amostra é de 1 linha por 768 colunas com valores representando o sinal em volts.

A escolha somente das frequências de interesse, ao invés do espectro todo, nos experimentos com os módulos da STFT e os espectrogramas, pode ser considerada como extração e seleção de características nas subetapas de processamento de BCI. Para estes dois formatos de dados, usar o espectro completo ainda seria considerado como extração de características, porém não selecionar as frequências de interesse resultou em um desempenho muito abaixo dos desempenhos obtidos por técnicas comumente aplicadas em BCI. Desta forma, optou-se por selecionar as frequências de interesse, resultando em uma diminuição drástica da quantidade de informação desnecessária enviada às redes. Estas amostras com informação específica e focada ajudaram a melhorar a convergência no treinamento das redes. Outro fator que pode ser considerado como um tipo de extração e seleção de característica é a conversão dos módulos da STFT para decibéis. Avaliando empiricamente, esta conversão resultou em um ganho de 2% em média nos resultados.

Um exemplo de amostra da base de dados de espectrogramas sem harmônica superior representando a classe de 12Hz está ilustrado na Figura 14a. Cada amostra dessa base de dados é composta pelas duas frequências de interesse e a concatenação da sua vizinhança, 1Hz abaixo e 1Hz acima. Assim, representadas na Figura 14a estão a frequência de interesse de 12Hz e suas vizinhas, 11Hz e 13Hz, e a frequência de 15Hz e suas vizinhas, 14Hz e 16Hz. Um exemplo de amostra da base de dados de espectrogramas com

uma harmônica superior representando a classe de 12Hz está ilustrado na Figura 14b. Cada amostra dessa base de dados é composta pela mesma informação da base de dados sem uma harmônica superior, porém são adicionadas uma harmônica superior para cada frequência de interesse. No caso da frequência de interesse de 12Hz, uma harmônica superior é representada pela frequência de 24Hz, sendo concatenadas também as suas vizinhas, 23Hz e 25Hz. No caso da frequência de interesse de 15Hz, uma harmônica superior é representada pela frequência de 30Hz, sendo concatenadas também as suas vizinhas, 29Hz e 31Hz. Nas figuras, cores mais claras representam uma maior amplitude do sinal, enquanto que, cores mais escuras representam uma menor amplitude do sinal.



(a) Exemplo de uma amostra da base de dados de espectrogramas sem harmônica superior representando a classe de 12Hz. (b) Exemplo de uma amostra da base de dados de espectrogramas com uma harmônica superior representando a classe de 12Hz.

Figura 14 – Exemplos de amostras da base de dados de espectrogramas.

### 5.3 Arquiteturas DCNNs para Classificação de Sinais SSVEP

Para investigar o desempenho das DCNNs no problema de classificação de sinais SSVEP, são propostas duas arquiteturas. A primeira arquitetura pode receber entradas no formato de módulo da STFT ou imagens de espectrograma. A segunda arquitetura recebe uma entrada no formato de sinal bruto. Com esta abordagem, é possível investigar o desempenho das arquiteturas propostas, assim como a influência do formato do dado nas arquiteturas.

As arquiteturas foram construídas com blocos básicos como uma forma de padronizá-las. Um bloco básico é composto por uma camada de convolução seguida por normalização do lote, desligamento, uma camada de agregação máxima e a função de ativação ReLU. O processo para determinar a estrutura das arquiteturas foi empírico. Foram usados os dados do sujeito 1 e do sujeito 5 da base de dados descrita em 5.1, como conjunto de teste em avaliações distintas, para avaliar as arquiteturas durante sua construção. Em Carvalho *et al.* (2015b), os dados do sujeito 1 representam um exemplo

em que os algoritmos de classificação têm acurácia menor e os dados do sujeito 5 representam um exemplo em que os algoritmos de classificação têm acurácia maior. Os dados dos outros nove sujeitos foram divididos entre conjunto de treinamento e validação, na proporção de 70% e 30%, respectivamente, para avaliar o desempenho das arquiteturas. Este desempenho foi avaliado incluindo blocos básicos em suas estruturas, variando a quantidade e o tamanho dos filtros das convoluções e ajustando a entrada das camadas totalmente conectadas, sempre com 2 camadas intermediárias. As arquiteturas com os melhores desempenhos para os dados dos sujeitos 1 e 5, nos formatos de dados avaliados, foram as escolhidas para os experimentos e estão detalhadas nas Seções 5.3.1 e 5.3.2.

### 5.3.1 Uma Arquitetura DCNN para Classificação do Módulo da STFT / Espectrograma

A arquitetura DCNN proposta para dados no formato de módulos da STFT ou imagens de espectrogramas, descritas em 5.2, consiste de 4 blocos básicos com camada de convolução 2D que se diferenciam pelo número de filtros e pelo tamanho dos filtros. Para as operações de convolução, não é aplicado preenchimento, ou seja, não são adicionados zeros fora da área da amostra (operação conhecida em inglês como *padding*) e o deslocamento do filtro (operação conhecida em inglês como *stride*) é unitário. Assim, os mapas de características têm tamanho sempre menor que a entrada da convolução.

Após cada um dos blocos, o mapa de características é interpolado pela técnica do vizinho mais próximo em escala de 1:2. Como as amostras de entrada têm tamanho reduzido, a interpolação é necessária para criar uma arquitetura profunda, já que a operação de convolução sempre reduz o tamanho dos mapas de características.

Ao final dos quatro blocos básicos, a rede totalmente conectada recebe como entrada os mapas de características da última camada convolucional no formato de um vetor. Os 30 mapas de características de tamanho 4 linhas por 2 colunas são concatenados e transformados em um vetor de 240 linhas ( $30 \times 4 \times 2$ ). Desta forma, a camada de entrada da rede totalmente conectada tem 240 neurônios, uma camada intermediária com 250 neurônios e uma camada intermediária com 125 neurônios, ambas com desligamento e ativação ReLU. A camada de saída tem 2 neurônios, representando as duas classes, de 12Hz e 15Hz, com ativação logaritmo do *softmax* (*log-softmax*). A Figura 15 ilustra a arquitetura da DCNN e detalha os filtros em cada bloco.

### 5.3.2 Uma Arquitetura DCNN para Classificação de Sinais Brutos

A arquitetura DCNN proposta para dados no formato de sinais brutos consiste de 4 blocos básicos com camada de convolução 1D que se diferenciam pelo número de filtros e pelo tamanho dos filtros. Para as operações de convolução, não é aplicado preenchimento

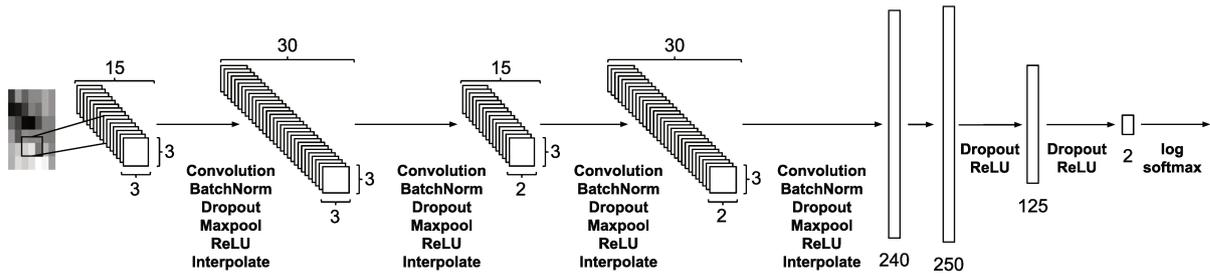


Figura 15 – Ilustração da arquitetura DCNN e seus filtros para dados no formato de módulos da STFT ou imagens de espectrogramas.

e o deslocamento dos filtros é de uma unidade. Assim os mapas de características têm tamanho sempre menor que a entrada da convolução.

Ao final dos quatro blocos básicos, a rede totalmente conectada recebe como entrada os mapas de características da última camada convolucional no formato de um vetor. Os 30 mapas de características de tamanho 18 linhas são concatenados e transformados em um vetor de 540 linhas ( $30 \times 18$ ). Desta forma, a camada de entrada da rede totalmente conectada tem 540 neurônios, uma camada intermediária de 250 neurônios e uma camada intermediária de 125 neurônios, ambas com desligamento e ativação ReLU. A camada de saída tem 2 neurônios, representando as duas classes, de 12Hz e 15Hz, com ativação logaritmo do *softmax* (*log-softmax*). A Figura 16 ilustra a arquitetura da DCNN e detalha os filtros em cada bloco.

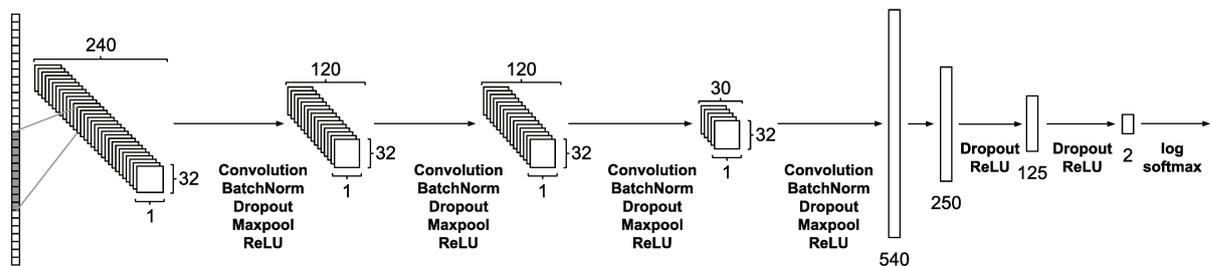


Figura 16 – Ilustração da arquitetura DCNN e seus filtros para dados no formato de sinais brutos.

## 5.4 Transferência de Conhecimento com ResNet

Para os experimentos com transferência de conhecimento, a ResNet foi a arquitetura adotada. Esta arquitetura foi proposta por He *et al.* (2016) com o objetivo de minimizar o problema da perda de informação do gradiente, conhecido em inglês como *vanishing gradient* (BENGIO *et al.*, 1994; HOCHREITER, 1998), obtendo o primeiro lugar na competição *ImageNet Large Scale Visual Recognition Competition* (ILSVRC) de 2015.

O nome dado à arquitetura, rede residual, vem da abordagem empregada na sua construção. Esta arquitetura é composta por blocos básicos de duas ou três convoluções, dependendo da variação de camadas da arquitetura, com uma ligação direta entre a entrada do bloco e sua saída, gerando um curto-circuito, levando, assim, um resíduo da entrada adiante na arquitetura. A Figura 17, ilustra a construção do bloco básico.

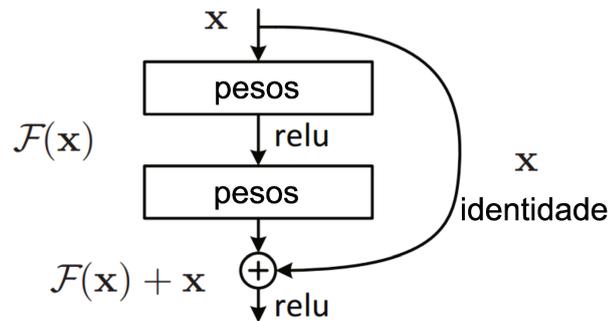


Figura 17 – Bloco básico da rede residual, adaptado de He *et al.* (2016).

As operações de convolução usam preenchimento para preservar o tamanho da entrada. A subamostragem é feita com deslocamento dos filtros em 2 unidades em camadas específicas da arquitetura. Após cada uma das operações de convolução, são aplicadas a normalização do lote e a função de ativação ReLU. Antes de aplicar a função de ativação após a última operação de convolução e normalização do lote, a entrada do bloco é somada à saída da normalização do lote, elemento a elemento.

Para demonstrar a capacidade da arquitetura em minimizar o problema da perda de informação do gradiente, o trabalho avaliou 5 variações no número de blocos básicos da arquitetura ResNet. Os pesos destas 5 variações ajustados pelo treinamento na base de dados ImageNet são disponibilizados publicamente e amplamente utilizados com sucesso na tarefa de transferência de conhecimento para problemas de classificação (NIBALI *et al.*, 2017; LEI *et al.*, 2018; O'SHEA *et al.*, 2018). Um dos maiores motivadores para adotar esta arquitetura na tarefa de transferência de conhecimento para classificação de sinais SSVEP, além do sucesso que ela obteve em outros problemas, é não haver aplicação deste tipo na literatura até o momento.

A Tabela 1 ilustra as 5 variações da arquitetura ResNet usadas na abordagem de transferência de conhecimento deste trabalho. A primeira coluna da tabela apresenta os nomes de cada camada, a segunda coluna apresenta o tamanho da saída de cada camada. As colunas de 2 a 6 apresentam as diferentes arquiteturas em relação a profundidade de cada uma, sendo o tamanho dos filtros e a quantidade de filtros da convolução de cada bloco básico apresentado entre chaves em cada linha.

A Figura 18, adaptada de He *et al.* (2016), ilustra a arquitetura ResNet com 34 camadas. Nela, é possível observar as subamostragens sempre na primeira camada de convolução do bloco básico. As flechas sólidas representam o resíduo sendo levado adi-

camadas	saída	18-camadas	34-camadas	50-camadas	101-camadas	152-camadas
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1,8×10 <sup>9</sup>	3,6×10 <sup>9</sup>	3,8×10 <sup>9</sup>	7,6×10 <sup>9</sup>	11,3×10 <sup>9</sup>

Tabela 1 – Variações da arquitetura ResNet. Entre chaves estão os blocos básicos com o tamanho dos filtros e a quantidade de filtros da convolução, seguido pelo número de blocos básicos. As subamostragens das entradas são feitas nas camadas conv2\_1, conv3\_1, conv4\_1 e conv5\_1. FLOPs, do inglês, *Floating Point Operations per second*. Adaptado de He *et al.* (2016).

ante. Nas flechas pontilhadas, os resíduos têm seus tamanhos reduzidos para corresponder ao tamanho do mapa de características subamostrado. A camada totalmente conectada é formada por mil neurônios, representando as mil classes disponíveis na base de dados ImageNet. Nos experimentos de transferência de conhecimento com a ResNet, os mil neurônios da camada totalmente conectada foram substituídos por dois neurônios, representando as duas classes, de 12Hz e 15Hz, do problema abordado.

## 5.5 Proposta de Arquiteturas DCNNs para Gerar o Mapeamento no Espaço Euclidiano

Para os experimentos com a função de custo *triplet*, as arquiteturas de DCNNs propostas para gerar o mapeamento das entradas no espaço euclidiano têm como base a arquitetura da DCNN proposta na Seção 5.3 para dados no formato de imagem de espectrograma, com pequenas modificações. São propostas duas arquiteturas, uma para entrada no formato de imagem de espectrograma nas frequências fundamentais e outra para entrada no formato de imagem de espectrograma nas frequências fundamentais mais uma harmônica superior para cada frequência.

Ambas as arquiteturas têm a última camada modificada para gerar o mapeamento das amostras de entrada em um espaço euclidiano de dimensão 64 e incluem a normalização L2, conforme proposto por Schroff *et al.* (2015). A arquitetura DCNN para dados no formato de imagem de espectrograma nas frequências fundamentais não sofreu alteração nas camadas convolucionais. Sua representação está ilustrada na Figura 19.

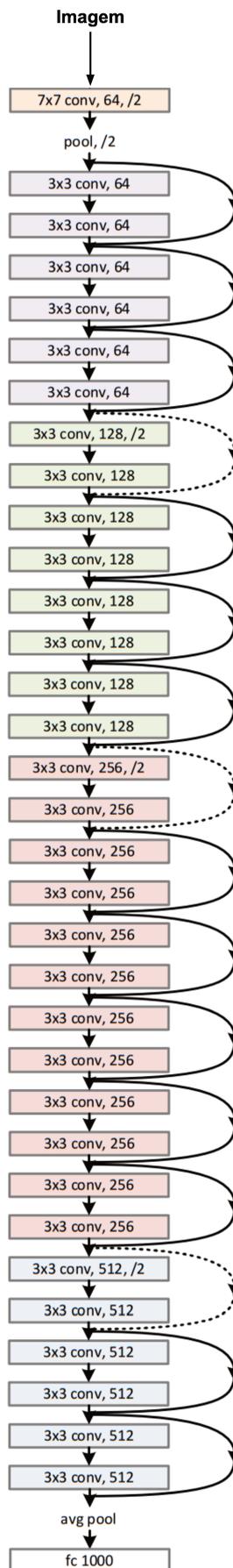


Figura 18 – Rede residual com 34 camadas, adaptado de He *et al.* (2016).

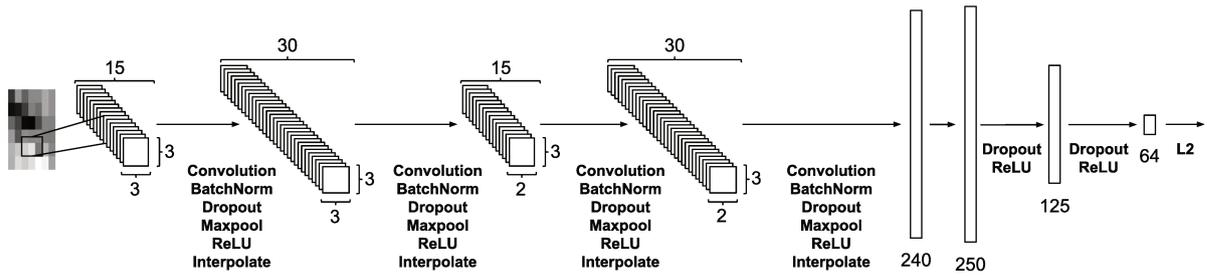


Figura 19 – Proposta de arquitetura DCNN para mapeamento no espaço euclidiano de amostras no formato de imagem de espectrograma nas frequências fundamentais.

A arquitetura DCNN para dados no formato de imagem de espectrograma, nas frequências fundamentais mais uma harmônica superior, precisou de alterações no tamanho dos filtros das camadas convolucionais para manter o número de neurônios da entrada das camadas totalmente conectadas. Sua representação está ilustrada na Figura 20.

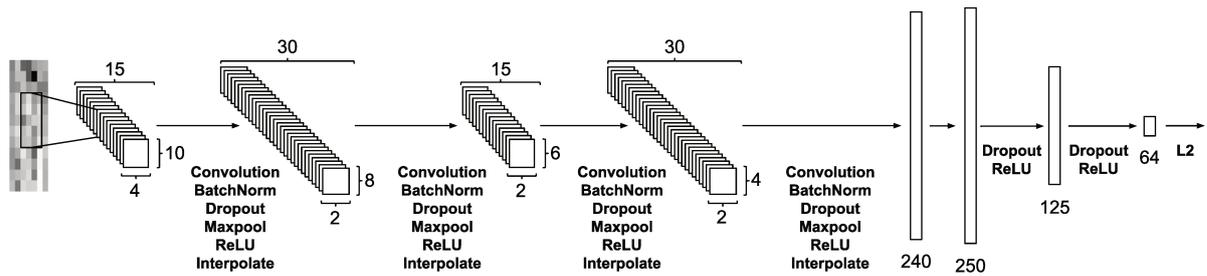


Figura 20 – Proposta de arquitetura DCNN para mapeamento no espaço euclidiano de amostras no formato de imagem de espectrograma nas frequências fundamentais mais uma harmônica superior.

## 6 Experimentos

Para investigar as propostas enumeradas no Capítulo 5, este Capítulo descreve os experimentos, seus resultados e discussões relacionadas. A estrutura adotada para as Seções dos experimentos é a seguinte: é apresentado o protocolo experimental, seguido pelos resultados de todos os experimentos relacionados e, ao final, é apresentada a discussão sobre os experimentos e seus resultados.

As Seções seguintes são dedicadas a três experimentos: o experimento da Seção 6.1 investiga o desempenho das arquiteturas DCNN propostas nas bases de dados geradas, o experimento da Seção 6.2 investiga a técnica de transferência de conhecimento com a arquitetura ResNet e o experimento da Seção 6.3 avalia a arquitetura DCNN proposta usando a função de custo *triplet* durante o treinamento.

### 6.1 Experimentos com as Arquiteturas DCNNs

Esta Seção descreve os experimentos realizados com as arquiteturas DCNN, descritas na Seção 5.3. As arquiteturas foram avaliadas em três bases de dados: a base de dados de espectrogramas sem harmônica superior, a base de dados dos módulos da STFT e a base de dados de sinais brutos.

São apresentados o protocolo experimental, os resultados obtidos com cada uma das arquiteturas e, no final da Seção, é feita uma discussão sobre os experimentos e seus resultados.

#### 6.1.1 Protocolo para os Experimentos com a DCNN

Os experimentos com as arquiteturas DCNN avaliaram duas variações no número de eletrodos das bases de dados descritas na Seção 5.2, ambas privilegiando eletrodos da região occipital. A primeira variação usa informação dos eletrodos O1, O2, Oz, POz, e a segunda variação usa informação somente do eletrodo Oz.

As bases de dados foram divididas em conjunto de treinamento, conjunto de validação e conjunto de teste. O conjunto de dados de cada um dos sujeitos foi avaliado individualmente como conjunto de teste enquanto os dados dos outros 9 sujeitos foram aleatoriamente divididos em 70% para o conjunto de treinamento e 30% para o conjunto de validação e usados no treinamento das arquiteturas. O conjunto de teste com 4 eletrodos contém 640 amostras da base de dados e com 1 eletrodo contém 160 amostras da base de dados. O conjunto de treinamento com 4 eletrodos contém 4.032 amostras da base de dados e com 1 eletrodo contém 1.008 amostras da base de dados e o conjunto de

validação com 4 eletrodos contém 1.728 amostras da base de dados e com 1 eletrodo contém 432 amostras da base de dados. A Figura 21 exemplifica a divisão dos conjuntos. Cada quadrado representa os dados de um sujeito. Os quadrados na cor azul representam os conjuntos de teste e os quadrados na cor verde representam os conjuntos de treinamento e validação.

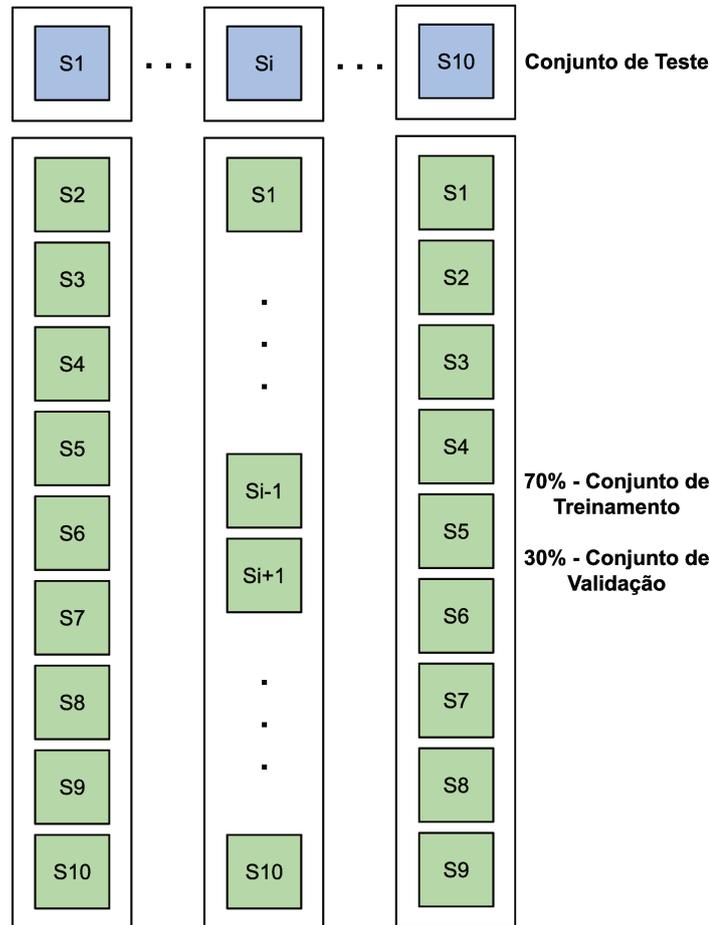


Figura 21 – Representação gráfica da divisão dos conjuntos de treinamento, validação e teste.

As arquiteturas DCNN foram treinadas para cada uma das 10 triplas de conjuntos de treinamento, validação e teste, gerando 10 modelos individuais para cada arquitetura, para cada sujeito avaliado como conjunto de teste.

O treinamento da arquitetura ocorreu por, no máximo, 1500 épocas. O treinamento foi interrompido usando a técnica de parada antecipada, descrita na Seção 3.4, quando o erro no conjunto de validação não diminuiu por 100 épocas consecutivas. Somente o modelo com menor erro no conjunto de validação foi armazenado para comparação. Para tornar computacionalmente eficiente o uso dos recursos de memória durante a etapa de treinamento, o conjunto de treinamento usou mini-lotes de 1024 amostras e o conjunto de validação usou mini-lotes de 32 amostras. O Adam foi o algoritmo de otimiza-

ção usado com taxa de aprendizado inicial de 0,0001. Os outros parâmetros do algoritmo foram mantidos conforme descrito no trabalho original. A função custo escolhida foi o logaritmo negativo da verossimilhança (NLL, do inglês *Negative Log-Likelihood*). A função de ativação *log-softmax* da última camada das arquiteturas DCNN e a função de custo NLL em conjunto equivalem à entropia cruzada como função custo.

Para garantir reprodutibilidade do experimento, no início de sua execução, a semente do gerador de números pseudo-aleatórios foi inicializada com o valor 12345.

Os resultados de classificação nos conjuntos de teste para as DCNNs treinadas foram comparados aos resultados de classificação de uma SVM linear por ser um algoritmo robusto, de simples implementação e comumente utilizado em BCIs (THOMAS *et al.*, 2017; CARVALHO *et al.*, 2015b). Para os experimentos que usaram as bases de dados de espectrogramas sem harmônica superior e os módulos da STFT, os resultados também foram comparados a uma abordagem baseada na soma das amplitudes de todas as frequências de mesma classe e na decisão pela máxima entre elas conforme exemplificado na Figura 22.

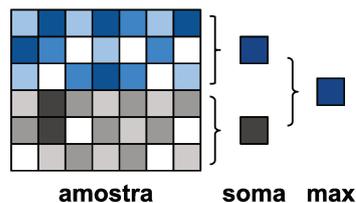


Figura 22 – Exemplo da soma das amplitudes nas frequências de interesse.

### 6.1.2 Avaliação de Espectrogramas como Entrada da DCNN

Este experimento avalia o desempenho da DCNN descrita na Seção 5.3.1 para dados de entrada no formato de espectrogramas sem harmônica superior, gerados conforme descrito na Seção 5.2.

A Figura 23 consolida os resultados de classificação na base de dados de espectrogramas sem harmônica superior usando informação dos eletrodos O1, O2, Oz, POz. Nesta figura, temos as acurácias da DCNN, da SVM linear e da abordagem de soma das amplitudes, assim como a média e o desvio padrão de cada uma.

A Figura 24 consolida os resultados de classificação na base de dados de espectrogramas sem harmônica superior usando informação do eletrodo Oz. Nesta figura, temos as acurácias da DCNN, da SVM linear e da abordagem de soma das amplitudes, assim como a média e o desvio padrão de cada uma.

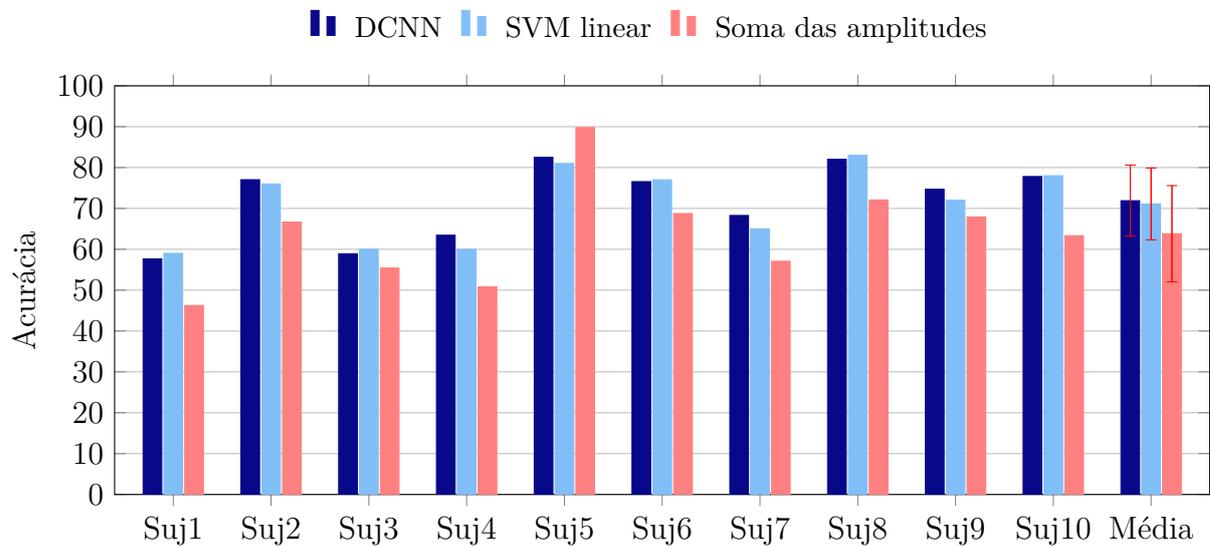


Figura 23 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de espectrogramas sem harmônica superior usando informação dos eletrodos O1, O2, Oz, POz.

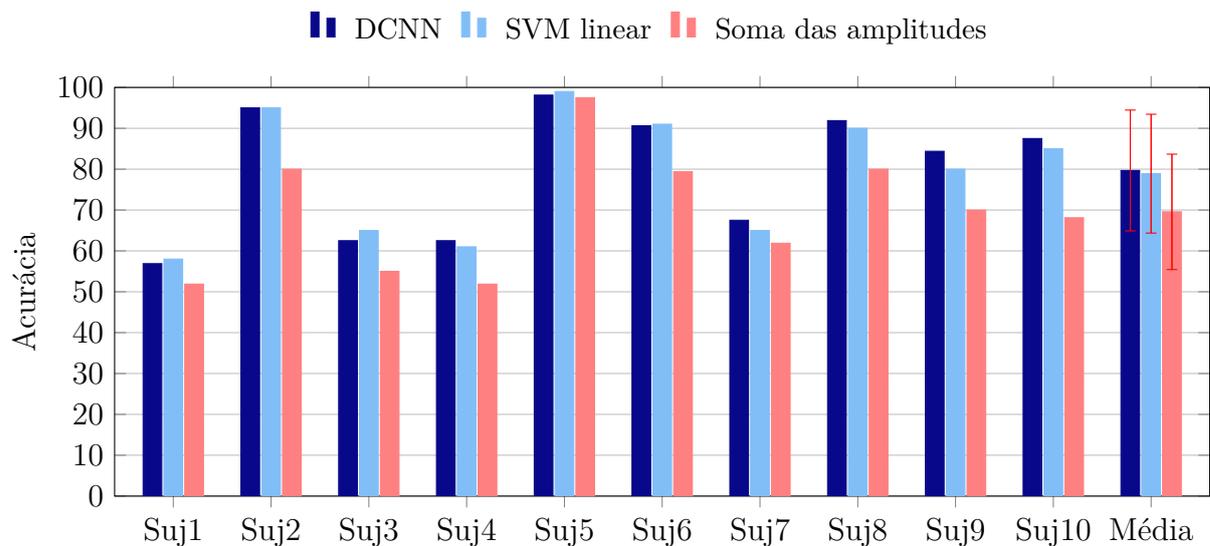


Figura 24 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de espectrogramas sem harmônica superior usando informação do eletrodo Oz.

### 6.1.3 Avaliação de Módulos da STFT como Entrada da DCNN

Este experimento avalia o desempenho da DCNN descrita na Seção 5.3.1 para dados de entrada no formato de módulos da STFT, gerados conforme descrito na Seção 5.2.

A Figura 25 consolida os resultados de classificação na base de dados de módulos da STFT usando informação dos eletrodos O1, O2, Oz, POz. Nesta figura, temos as acurácias da DCNN, da SVM linear e da abordagem de soma das amplitudes, assim como a média e o desvio padrão de cada uma.

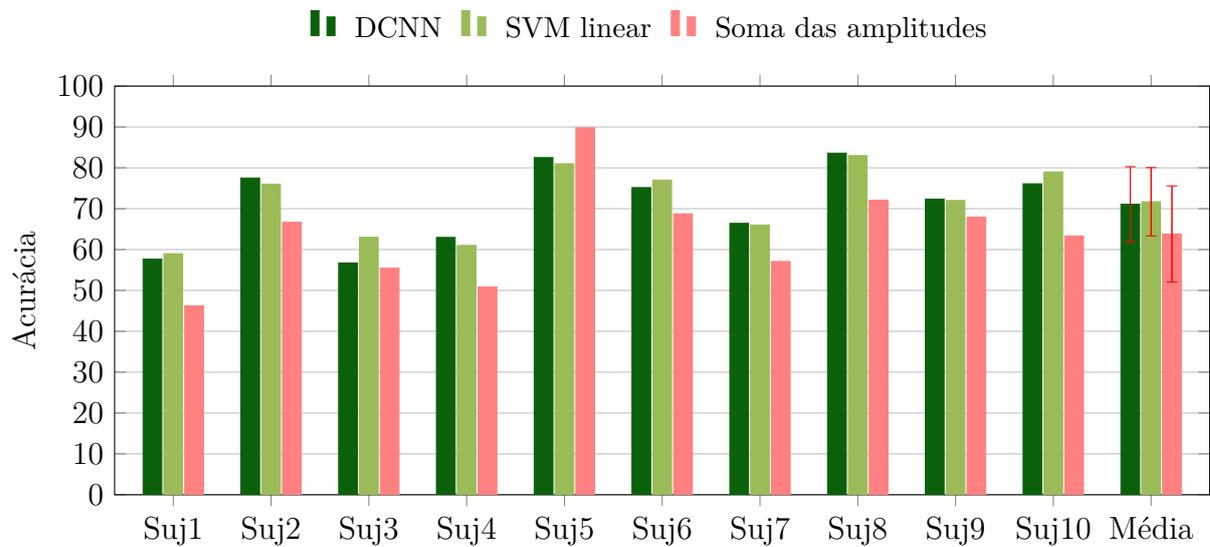


Figura 25 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de módulos da STFT usando informação dos eletrodos O1, O2, Oz, POz.

A Figura 26 consolida os resultados de classificação na base de dados de módulos da STFT usando informação do eletrodo Oz. Nesta figura, temos as acurácias da DCNN, da SVM linear e da abordagem de soma das amplitudes, assim como a média e o desvio padrão de cada uma.

### 6.1.4 Avaliação de Sinais Brutos como Entrada da Rede

Este experimento avalia o desempenho da DCNN descrita na Seção 5.3.2 para dados de entrada no formato de sinais brutos, gerados conforme descrito na Seção 5.2.

A Figura 27 consolida os resultados de classificação na base de dados de sinais brutos usando informação dos eletrodos O1, O2, Oz, POz. Nesta figura, temos as acurácias da DCNN e da SVM linear, assim como a média e o desvio padrão de cada uma.

A Figura 28 consolida os resultados de classificação na base de dados de sinais brutos usando informação do eletrodo Oz. Nesta figura, temos as acurácias da DCNN e da SVM linear, assim como a média e o desvio padrão de cada uma.

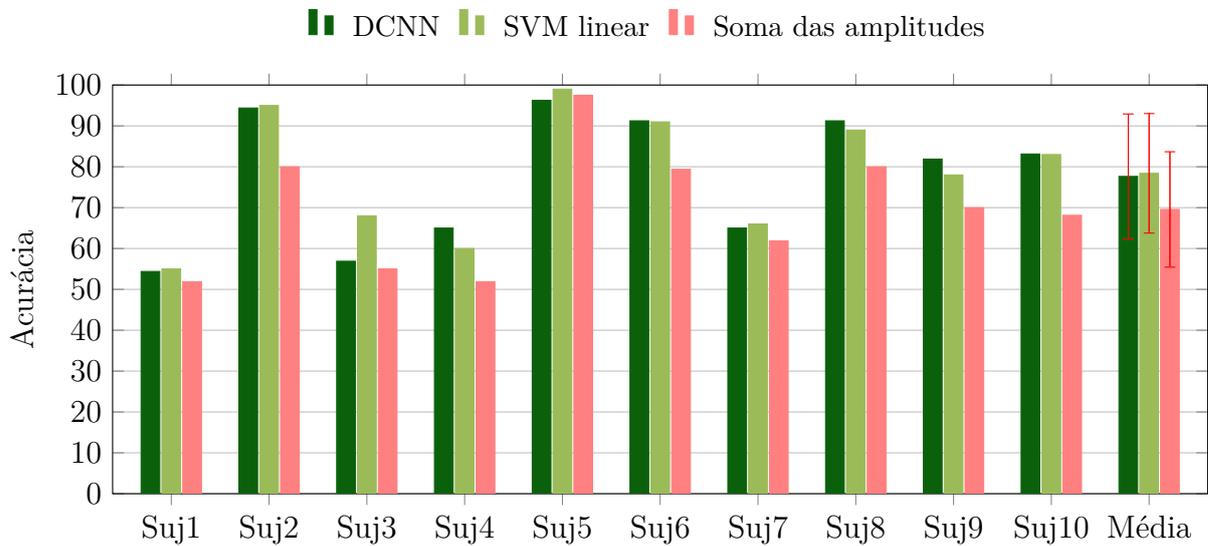


Figura 26 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de módulos da STFT usando informação do eletrodo Oz.

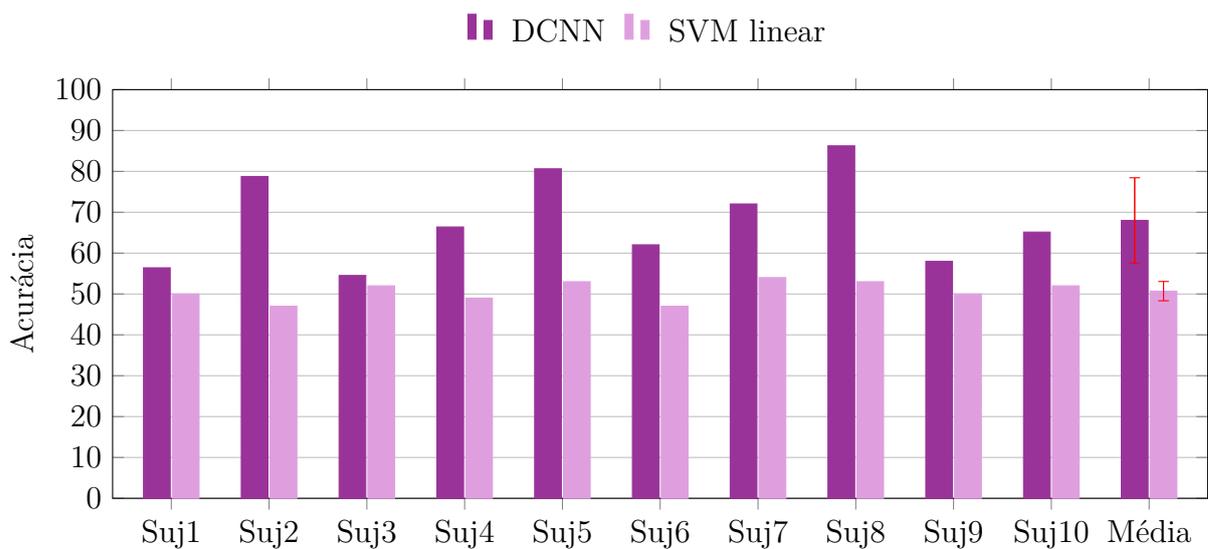


Figura 27 – Acurácia da DCNN e da SVM linear na base de dados de sinais brutos usando informação dos eletrodos O1, O2, Oz, POz.

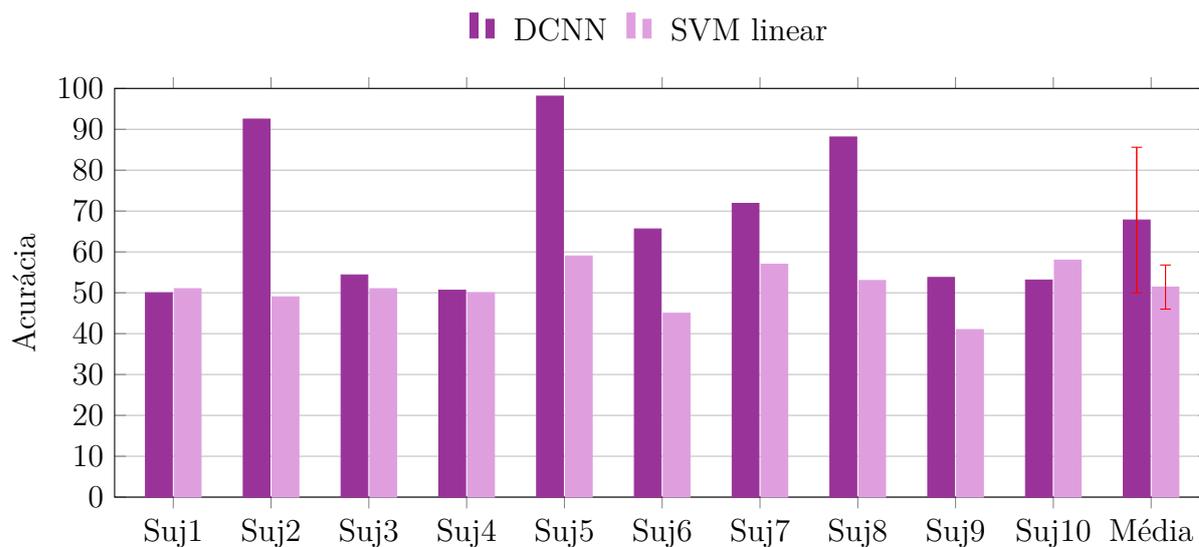


Figura 28 – Acurácia da DCNN e da SVM linear na base de dados de sinais brutos usando informação do eletrodo Oz.

### 6.1.5 Discussão dos Resultados

Comparando as médias das acurácias para a DCNN e para a SVM linear, nas bases de dados de espectrogramas sem harmônica superior e na base de dados de módulos da STFT, é perceptível a proximidade entre elas para as duas variações de números de eletrodos. Para a base de dados de espectrogramas sem harmônica superior, as médias para a DCNN e para a SVM linear são 71,89% e 71,10%, respectivamente, usando a informação dos eletrodos O1, O2, Oz, POz e 79,68% e 78,90%, respectivamente, usando a informação do eletrodo Oz. Para a base de dados de módulos da STFT as médias para a DCNN e para a SVM linear são 71,09% e 71,70%, respectivamente, usando a informação dos eletrodos O1, O2, Oz, POz e 77,64% e 78,40%, respectivamente, usando a informação do eletrodo Oz.

Uma análise da proximidade das médias pode indicar duas hipóteses: a primeira delas descreve o problema como essencialmente linear; a segunda indica a existência de relações não-lineares que ainda podem ser exploradas de maneira mais elaborada. Apesar de a segunda hipótese ainda requerer investigação, cabe destacar a possibilidade de explorar a DCNN treinada para uma abordagem de ajuste fino (do inglês, *fine-tuning*) por sujeito, reduzindo o tempo de treinamento de um sujeito em um sistema BCI SSVEP. De forma exemplificada, o sujeito 1, sujeito com as piores acurácias em praticamente todos os experimentos, seria um candidato para uma abordagem de ajuste fino destes modelos pré-treinados. Já o sujeito 5 tem boa acurácia em praticamente todos os experimentos, não havendo necessidade de ajuste fino, estando este sujeito apto para usar o sistema BCI SSVEP sem precisar de uma etapa de treinamento prévio.

Em relação ao uso de espectrogramas ou módulos da STFT como entrada da

DCNN, as médias dos experimentos demonstram que a DCNN é capaz de se ajustar a ambos os formatos de dados sem variação brusca no desempenho. Como os formatos só diferem na faixa de valores numéricos que os representam, os resultados são bem semelhantes para os dois formatos. Considerando que estes formatos de dados são equivalentes, é computacionalmente mais eficiente optar pelos módulos da STFT como entrada da DCNN, por exigir uma operação de transformação a menos, não sendo necessária a conversão dos seus valores em uma imagem do espectrograma.

Já para os experimentos com a base de dados de sinais brutos, as médias para a DCNN e para a SVM linear são 68,01% e 50,70%, respectivamente, usando a informação dos eletrodos O1, O2, Oz, POz e 67,81% e 51,40%, respectivamente, usando a informação do eletrodo Oz. A grande diferença entre as médias pode ser explicada pela capacidade da DCNN extrair as características do sinal de entrada antes de enviá-las à rede densa para serem classificadas. Na SVM linear, uma entrada de dimensão 768 não permite que o modelo se ajuste adequadamente, ficando com desempenho em torno de 50% para este caso de estudo envolvendo duas classes. Comparando com os desempenhos da SVM linear nas bases de dados de módulos da STFT e espectrogramas sem harmônica superior, nota-se que, para entradas de tamanho reduzido (12 linhas por 7 colunas, 84 valores de entrada), ela se ajusta a ponto de obter resultados próximos aos de algoritmos estado-da-arte usualmente empregados em BCI, como o CCA e seus derivados.

Em relação ao uso de informação de diferentes números de eletrodos para a base de dados de sinais brutos, as médias indicam que não há um grande impacto. Porém, o desvio padrão demonstra que, para alguns usuários, somente a informação do eletrodo Oz é suficiente para a classificação do sinal. Já para outros usuários, a informação de um maior número de eletrodos é essencial, como no caso do sujeito 4.

Ponto interessante a ser destacado na avaliação da base de dados de sinais brutos é a rápida divergência entre o erro no conjunto de treinamento e o erro no conjunto de validação demonstrando que, apesar dos resultados promissores, a DCNN sobreajusta muito rápido aos dados do conjunto de treinamento. O gráfico de treinamento do sujeito 2 com a evolução do erro no conjunto de treinamento e a evolução do erro no conjunto de validação da Figura 29 exhibe o treinamento da DCNN usando informação dos eletrodos O1, O2, Oz, POz. O gráfico da Figura 30 exhibe o treinamento da DCNN usando informação do eletrodo Oz.

Dois fatores podem influenciar o rápido sobreajuste da DCNN para a base de dados de sinais brutos e precisam de uma análise mais profunda: O primeiro fator está relacionado à quantidade insuficiente de dados disponível para o treinamento da DCNN. Existe a hipótese de que, com uma quantidade maior de dados, a DCNN não sobreajustaria tão rápido. Um indicativo deste comportamento é o menor desvio padrão dos resultados obtidos com a informação dos eletrodos O1, O2, Oz, POz, indicando uma me-

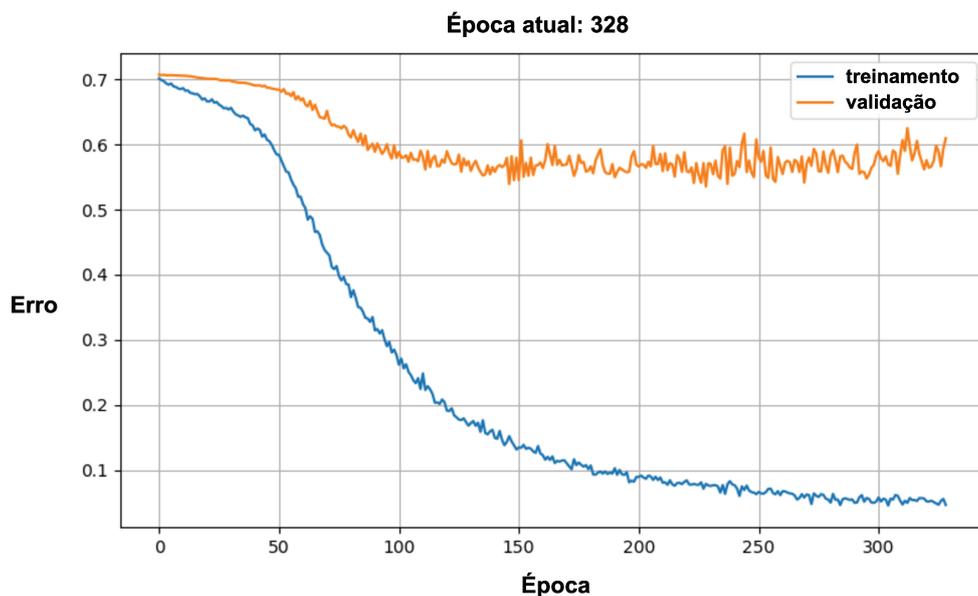


Figura 29 – Gráfico das evoluções do erro no conjunto de treinamento (curva inferior) e do erro no conjunto de validação (curva superior) durante o treinamento da DCNN na base de dados de sinais brutos para o sujeito 2 usando informação dos eletrodos O1, O2, Oz, POz.

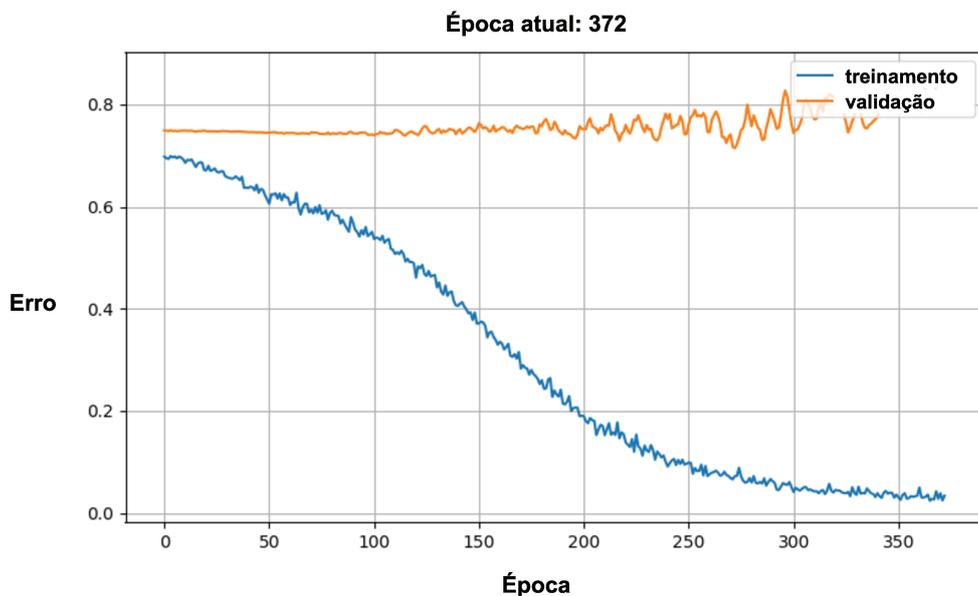


Figura 30 – Gráfico das evoluções do erro no conjunto de treinamento (curva inferior) e do erro no conjunto de validação (curva superior) durante o treinamento da DCNN na base de dados de sinais brutos para o sujeito 2 usando informação do eletrodo Oz.

lhor generalização do modelo com uma quantidade maior de dados disponível. O segundo fator está relacionado à regularização da arquitetura DCNN. Um estudo mais amplo de outros tipos de regularização aplicados à arquitetura DCNN do experimento podem trazer benefícios ao seu treinamento, sem a necessidade de uma base de dados com um número maior de amostras.

## 6.2 Experimentos de Transferência de Conhecimento com a Arquitetura ResNet

Esta Seção descreve os experimentos realizados com a técnica de transferência de conhecimento na arquitetura ResNet, descrita na Seção 5.4. O desempenho da transferência de conhecimento na arquitetura ResNet foi avaliado na base de dados de espectrogramas sem harmônica superior.

São apresentados o protocolo experimental, os resultados obtidos e, no final da Seção, é feita uma discussão sobre os experimentos e seus resultados.

### 6.2.1 Protocolo para os Experimentos de Transferência de Conhecimento

O experimento de transferência de conhecimento avaliou somente a base de espectrogramas sem harmônica superior, descrita na Seção 5.2, com duas variações no número de eletrodos, ambas privilegiando eletrodos da região occipital. A primeira variação usa informação dos eletrodos O1, O2, Oz, POz, e a segunda variação usa informação somente do eletrodo Oz.

As bases de dados foram divididas em conjunto de treinamento, conjunto de validação e conjunto de teste. O conjunto de dados de cada um dos sujeitos foi avaliado individualmente como conjunto de teste enquanto os dados dos outros 9 sujeitos foram aleatoriamente divididos em 70% para o conjunto de treinamento e 30% para o conjunto de validação e usados no ajuste da arquitetura. O conjunto de teste com 4 eletrodos contém 640 amostras da base de dados e com 1 eletrodo contém 160 amostras da base de dados. O conjunto de treinamento com 4 eletrodos contém 4.032 amostras da base de dados e com 1 eletrodo contém 1.008 amostras da base de dados e o conjunto de validação com 4 eletrodos contém 1.728 amostras da base de dados e com 1 eletrodo contém 432 amostras da base de dados.

O modelo pré-treinado da arquitetura ResNet foi ajustado para cada uma das 10 triplas de conjuntos de treinamento, validação e teste, gerando 10 modelos individuais, um para cada sujeito avaliado como conjunto de teste.

Conforme discutido por Yosinski *et al.* (2014) e descrito na Seção 3.5, aplicar a transferência de conhecimento na base de dados avaliada se enquadra no caso de o novo

conjunto de dados ser pequeno e muito diferente do conjunto de dados original. A decisão tomada para ajustar os modelos da arquitetura ResNet, neste caso, foi treinar todas as camadas da arquitetura por algumas poucas épocas com a perspectiva de ajuste dos pesos aos novos dados.

O ajuste do modelo ocorreu por 25 épocas, porém o modelo com o menor erro no conjunto de validação durante o ajuste foi armazenado para comparação. A escolha do número de épocas se deu empiricamente. Durante os ajustes os modelos alcançaram acurácia de 100% no conjunto de treinamento entre as épocas 16 e 22 e o erro no conjunto de validação passou a não diminuir a partir daí. Para tornar computacionalmente eficiente o uso dos recursos de memória durante a etapa de treinamento, o conjunto de treinamento usou mini-lotes de 32 amostras e o conjunto de validação usou mini-lotes de 32 amostras. O Adam foi o algoritmo de otimização usado com taxa de aprendizado inicial de 0,001. Este parâmetro foi ajustado por um escalonador (do inglês, *scheduler*) de taxa de aprendizado, regulando seu valor para 10% do valor anterior a cada 10 épocas. Assim, durante as 10 primeiras épocas o Adam usou a taxa de aprendizado de 0,001, nas 10 próximas épocas 0,0001 e nas últimas 5 épocas 0,00001. Os outros parâmetros do algoritmo foram mantidos conforme descrito no trabalho original. A função custo escolhida foi a entropia cruzada.

Para garantir reprodutibilidade do experimento, no início de sua execução, a semente do gerador de números pseudo-aleatórios foi inicializada com o valor 12345.

### 6.2.2 Avaliação de Transferência de Conhecimento com a ResNet

Este experimento avalia o desempenho da transferência de conhecimento com a arquitetura ResNet para dados de entrada no formato de espectrogramas sem harmônicas superiores, gerados conforme descrito na Seção 5.2.

A Figura 31 consolida os resultados de classificação na base de dados de espectrogramas sem harmônica superior usando informação dos eletrodos O1, O2, Oz, POz. Nesta figura, temos as acurácias dos arranjos da arquitetura ResNet avaliados, assim como a média e o desvio padrão de cada uma.

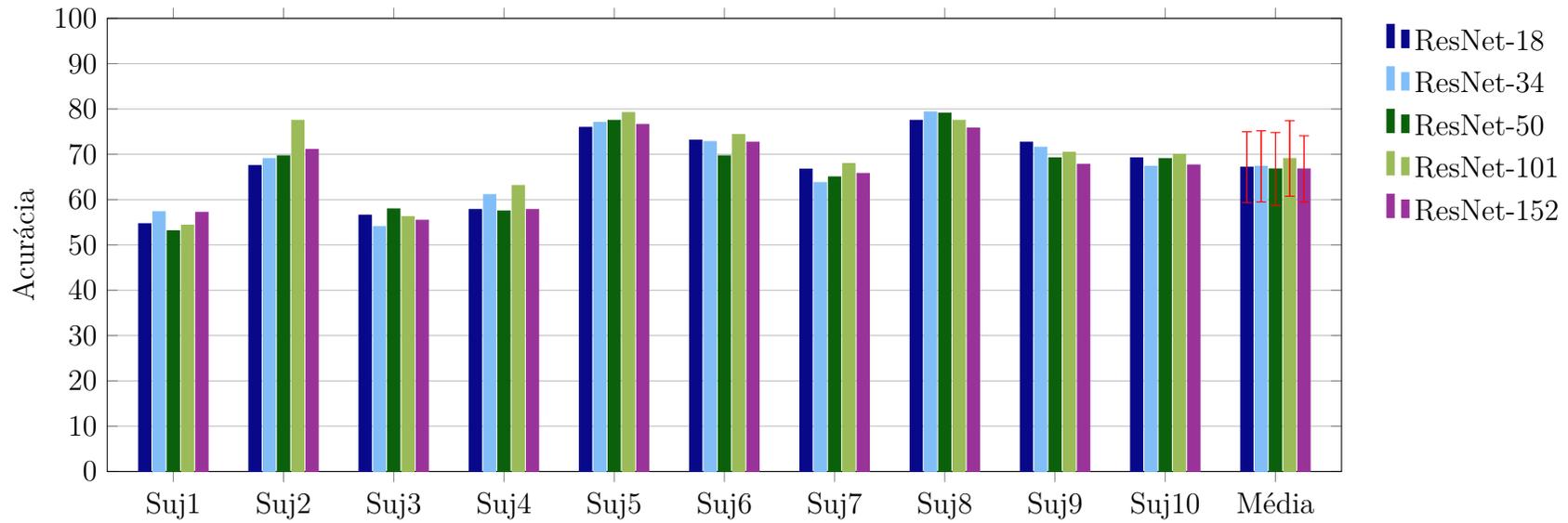


Figura 31 – Acurácias dos diferentes arranjos da arquitetura ResNet na base de dados de espectrogramas sem harmônica superior usando informação dos eletrodos O1, O2, Oz, POz.

A Figura 32 consolida os resultados de classificação na base de dados de espectrogramas sem harmônica superior usando informação do eletrodo Oz. Nesta figura, temos as acurácias dos arranjos da arquitetura ResNet avaliados, assim como a média e o desvio padrão de cada uma.

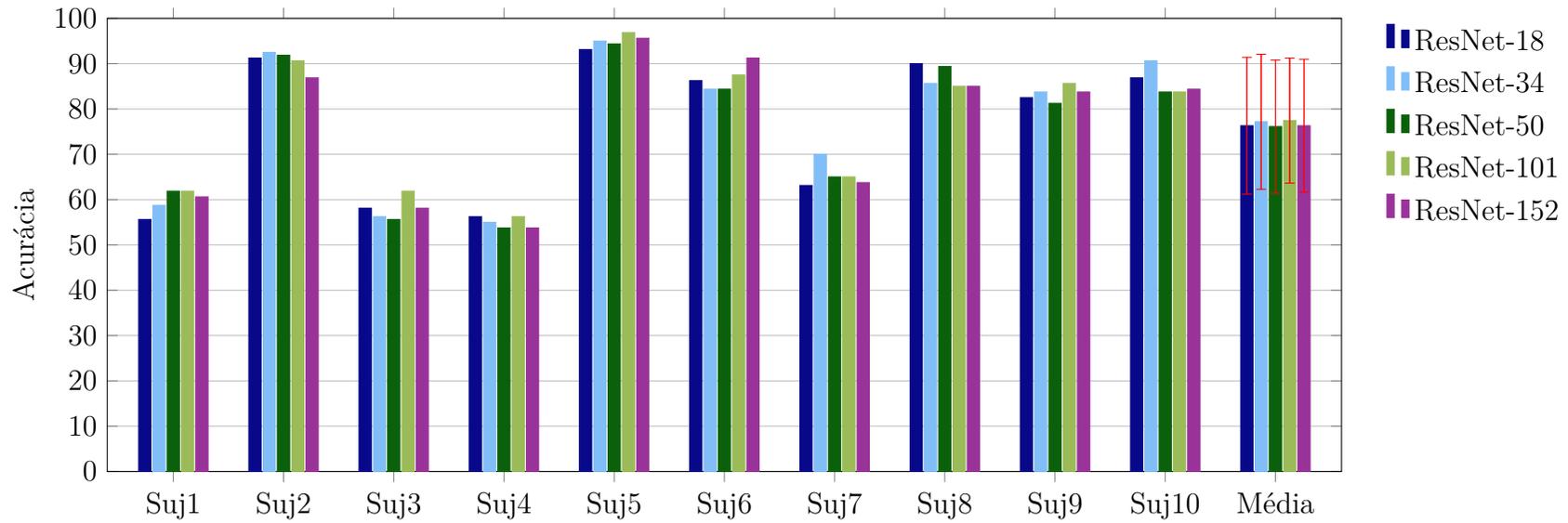


Figura 32 – Acurácias dos diferentes arranjos da arquitetura ResNet na base de dados de espectrogramas sem harmônica superior usando informação do eletrodo Oz.

### 6.2.3 Discussão dos Resultados

Comparando as médias das acurácias para a ResNet na base de dado de espectrogramas sem harmônica superior, é perceptível a proximidade entre elas dentro de cada um dos experimentos, mas, quando as médias são comparadas em relação ao número de eletrodos, há uma diferença grande entre elas. Para a base de dados de espectrogramas sem harmônica superior usando a informação dos eletrodos O1, O2, Oz, POz, as médias para a ResNet em ordem crescente de arranjo são 67,17%, 67,34%, 66,78%, 69,07% e 66,78%. Para a base de dados de espectrogramas sem harmônica superior usando a informação do eletrodo Oz, as médias para a ResNet em ordem crescente de arranjo são 76,31%, 77,18%, 76,12%, 77,43% e 76,31%.

Fazendo uma análise inicial das médias, a primeira impressão é a de que os modelos ajustados somente com a informação do eletrodo Oz sempre se saíram melhores comparados aos modelos ajustados com a informação dos eletrodos O1, O2, Oz, POz. Porém, é importante analisar também o desvio padrão para cada uma das variações do número de eletrodos. Para a base de dados de espectrogramas sem harmônica superior usando a informação dos eletrodos O1, O2, Oz, POz, os valores de desvio padrão para a ResNet em ordem crescente de arranjo são 7,80%, 7,85%, 8,04%, 8,32% e 7,30%. Para a base de dados de espectrogramas sem harmônica superior usando a informação do eletrodo Oz, os valores de desvio padrão para a ResNet em ordem crescente de arranjo são 15,08%, 14,90%, 14,68%, 13,80% e 14,65%. Os valores de desvio padrão maiores para a ResNet ajustada com a informação do eletrodo Oz são claramente influenciados pelos desempenhos dos modelos ajustados aos dados dos sujeitos 1, 3, 4 e 7. O desempenho da ResNet ajustada para os outros sujeitos fica sempre acima de 80%. Um ponto interessante a se destacar no caso da ResNet com desempenho menor é a proximidade dos seus resultados, independente da variação no número de eletrodos. Uma hipótese para este desempenho inferior é o número insuficiente de dados para o ajuste da rede. Desta forma, ela não é capaz de generalizar para estes sujeitos.

Diferentemente da DCNN treinada sobre a base de dados de sinais brutos, para este experimento, não há a possibilidade de explorar novas técnicas de regularização sobre a arquitetura por se tratar de uma abordagem de transferência de conhecimento. Porém, sempre é possível avaliar novas arquiteturas com pesos pré-treinados que melhor se adaptem aos dados destes sujeitos.

Analisando a influência da profundidade das arquiteturas ResNet no desempenho, não há uma relação direta entre arquiteturas mais profundas e melhor desempenho para o tipo de dados utilizado nos experimentos. O que é possível afirmar é que, quanto mais profunda é a arquitetura, maior é o tempo computacional para o seu ajuste, assim como para a classificação de uma amostra, o que pode influenciar diretamente a operação de um sistema BCI.

## 6.3 Experimentos com a DCNN e a Função de Custo Triplet

Esta Seção descreve os experimentos realizados com a arquitetura de DCNN descrita na Seção 5.5 usando a função de custo *triplet*. A arquitetura foi avaliada em duas bases de dados: a base de dados de espectrogramas sem harmônica superior e a base de dados de espectrogramas com uma harmônica superior.

São apresentados o protocolo experimental, os resultados obtidos e, no final da Seção, é feita uma discussão sobre os experimentos e seus resultados.

### 6.3.1 Protocolo para os Experimentos com a Função de Custo Triplet

O experimento com a função de custo *triplet* avaliou duas variações no número de eletrodos das bases de dados de espectrogramas descritas na Seção 5.2, ambas privilegiando eletrodos da região occipital. A primeira variação usa informação dos eletrodos O1, O2, Oz, POz, e a segunda variação usa informação somente do eletrodo Oz.

As bases de dados foram divididas em conjunto de treinamento, conjunto de validação e conjunto de teste. O conjunto de dados de cada um dos sujeitos foi avaliado individualmente como conjunto de teste enquanto os dados dos outros 9 sujeitos foram aleatoriamente divididos em 70% para o conjunto de treinamento e 30% para o conjunto de validação e usados no treinamento da arquitetura. O conjunto de teste com 4 eletrodos contém 640 amostras da base de dados e com 1 eletrodo contém 160 amostras da base de dados. O conjunto de treinamento com 4 eletrodos contém 4.032 amostras da base de dados e com 1 eletrodo contém 1.008 amostras da base de dados e o conjunto de validação com 4 eletrodos contém 1.728 amostras da base de dados e com 1 eletrodo contém 432 amostras da base de dados.

A arquitetura DCNN foi treinada para cada uma das 10 combinações de conjuntos de treinamento, validação e teste, gerando 10 modelos individuais, um para cada sujeito avaliado como conjunto de teste.

Para treinar uma rede com função de custo *triplet*, são necessárias triplas de amostras, conforme descrito na Seção 3.6. Para gerar as triplas, decide-se por uma amostra âncora no conjunto de dados e, aleatoriamente, pela amostra positiva e pela amostra negativa. Em uma época do treinamento, todas as amostras dos conjuntos de treinamento e de validação são consideradas âncora uma única vez. Os testes efetuados com outras formas de gerar as triplas, como as descritas por Schroff *et al.* (2015), não resultaram na melhora do desempenho ou na redução do tempo de treinamento para o experimento.

O treinamento da arquitetura ocorreu por 3000 épocas, porém o modelo com menor erro no conjunto de validação durante o treinamento foi armazenado para compa-

ração. Para tornar computacionalmente eficiente o uso dos recursos de memória durante a etapa de treinamento, o conjunto de treinamento usou mini-lotes de 1024 amostras e o conjunto de validação usou mini-lotes de 1024 amostras. O Adam foi o algoritmo de otimização usado com taxa de aprendizado inicial de 0,001. Os outros parâmetros do algoritmo foram mantidos conforme descrito no trabalho original.

Para garantir reprodutibilidade do experimento, no início de sua execução, a semente do gerador de números pseudo-aleatórios foi inicializada com o valor 12345.

Para avaliar os modelos treinados, as amostras do conjunto de teste foram classificadas conforme descrito na Seção 3.6. Foram comparadas as distâncias euclidianas entre o valor de saída da rede para a amostra do conjunto de teste e os valores de saídas para amostras de cada uma das classes, retiradas do conjunto de treinamento. Os resultados de classificação nos conjuntos de teste para as DCNNs treinadas foram comparados aos resultados de classificação de uma SVM linear.

### 6.3.2 Avaliação de Arquitetura DCNN com a Função de Custo Triplet

Este experimento avalia o desempenho da DCNN com a função de custo *triplet* descrita na Seção 3.6 para dados de entrada no formato de espectrogramas sem harmônicas superiores, assim como dados de entrada no formato de espectrogramas com uma harmônica superior, gerados conforme descrito na Seção 5.2.

A Figura 33 consolida os resultados de classificação na base de dados de espectrogramas sem harmônica superior e com uma harmônica superior usando informação dos eletrodos O1, O2, Oz, POz. Nesta figura, temos as acurácias da DCNN e da SVM linear, assim como a média e o desvio padrão de cada uma.

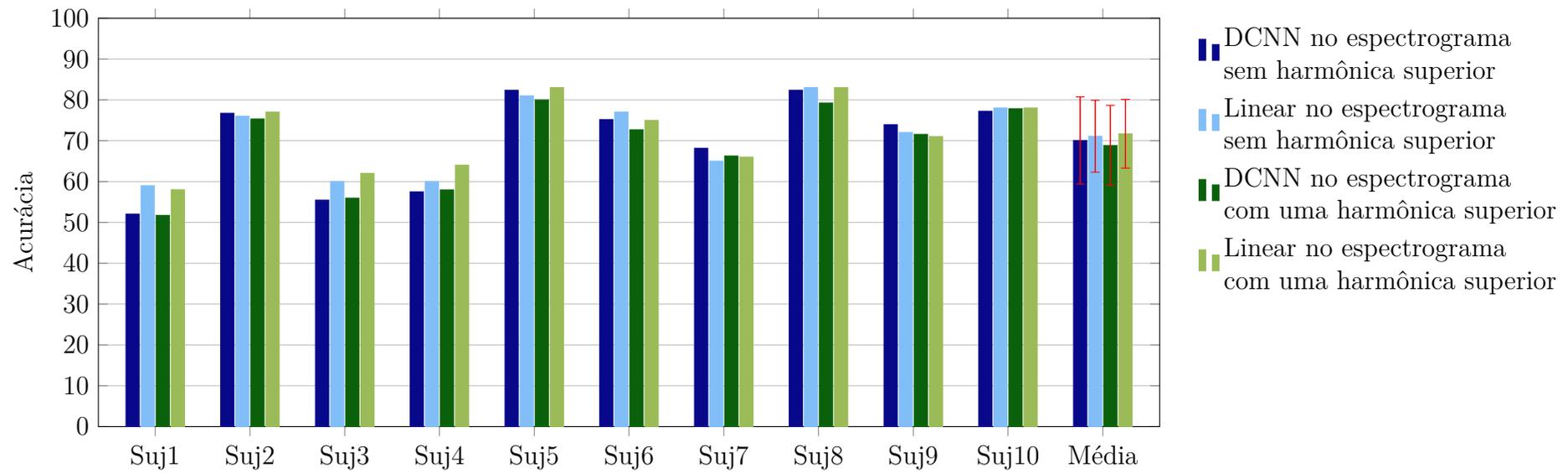


Figura 33 – Acurácia da DCNN com função de custo *triplet* e da SVM linear na base de dados de espectrogramas sem harmônica superior e com uma harmônica superior usando informação dos eletrodos O1, O2, Oz, POz.

A Figura 34 consolida os resultados de classificação na base de dados de espectrogramas sem harmônica superior e com uma harmônica superior usando informação do eletrodo Oz. Nesta figura, temos as acurácias da DCNN e da SVM linear, assim como a média e o desvio padrão de cada uma.

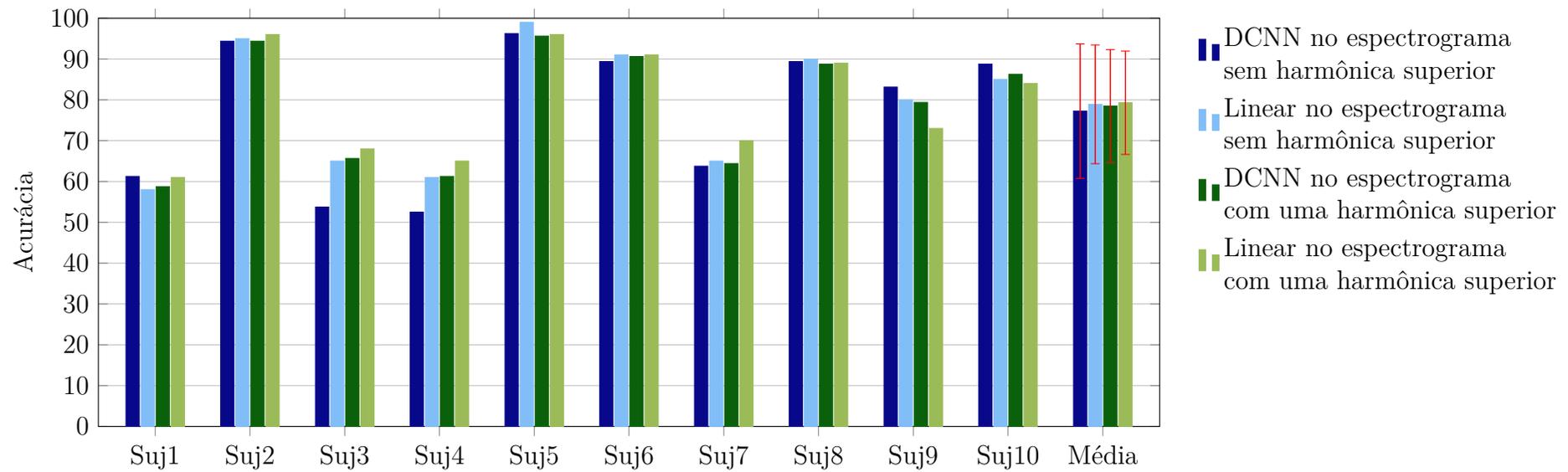


Figura 34 – Acurácia da DCNN com função de custo *triplet* e da SVM linear na base de dados de espectrogramas sem a primeira harmônica e com a primeira harmônica usando informação do eletrodo Oz.

### 6.3.3 Discussão dos Resultados

O objetivo deste experimento foi avaliar se uma abordagem usando a função de custo *triplet* traria algum benefício ao treinamento da DCNN na base de dados de espectrogramas ou um ganho de desempenho, assim como avaliar a influência de uma harmônica superior do sinal de interesse no desempenho dos modelos.

Um dos fatores que afetam diretamente a etapa de treinamento da DCNN usando a função de custo *triplet* é o uso de regularização. As regularizações propostas para a DCNN e descritas na Seção 5.3.1 se mostraram essenciais para tornar possível o treinamento e a convergência da arquitetura.

Em relação ao desempenho, não houve uma diferença significativa entre a DCNN treinada usando a função de custo *triplet*, independente da inclusão de uma harmônica superior do sinal de interesse ao espectrograma, e a DCNN treinada com a função de custo da entropia cruzada. As médias para os modelos DCNN com função de custo da entropia cruzada na base de dados de espectrogramas sem harmônica superior são 71,89% usando a informação dos eletrodos O1, O2, Oz, POz e 79,68% usando a informação do eletrodo Oz. As médias para os modelos DCNN com a função de custo *triplet* na base de dados de espectrogramas sem harmônica superior são 70,07% usando a informação dos eletrodos O1, O2, Oz, POz e 77,25% usando a informação do eletrodo Oz. As médias para os modelos DCNN com a função de custo *triplet* na base de dados de espectrogramas com uma harmônica superior são 68,84% usando a informação dos eletrodos O1, O2, Oz, POz e 78,50% usando a informação do eletrodo Oz.

Apesar da proximidade de desempenho, o fator mais impactante para o uso da função de custo *triplet* é a dificuldade de convergência do modelo e o elevado tempo computacional tanto para a etapa de treinamento quanto para classificação de uma amostra. Durante a etapa de treinamento com a função de custo *triplet*, foi necessário usar um número de épocas 2 vezes maior para chegar ao mesmo desempenho de um modelo treinado com a função de custo da entropia cruzada. Pelo fato de o tempo computacional da função de custo *triplet* para avaliar uma amostra ser maior, dada a necessidade de avaliar não somente a amostra âncora, mas também uma amostra positiva e uma amostra negativa, e a convergência ser mais lenta, o tempo total de treinamento de uma arquitetura usando a função de custo *triplet* se torna de 2 a 3 vezes maior quando comparada ao treinamento da mesma arquitetura usando a função de custo da entropia cruzada.

# Conclusão

O objetivo principal deste trabalho foi explorar as DCNNs na tarefa de processamento de sinais cerebrais em BCIs SSVEP. Para tal, o trabalho investigou diferentes abordagens usadas em DL, como a avaliação de novas arquiteturas definidas para o problema, a avaliação da transferência de conhecimento e a avaliação de uma função de custo que não havia sido reportada ainda na literatura aplicada a este problema. As contribuições deste trabalho podem ser relacionadas aos resultados obtidos para cada uma das propostas associadas ao objetivo principal.

A primeira proposta era investigar diferentes formatos de dados e a influência sobre as DCNNs. Os resultados dos experimentos mostraram que o módulo da STFT e a imagem do espectrograma são equivalentes quando empregados na mesma arquitetura de rede. Já os dados do sinal bruto precisam de uma arquitetura mais robusta, que seja capaz de extrair as características corretas deste tipo de dado.

A segunda proposta era investigar diferentes arquiteturas de DCNN aplicadas aos diferentes formatos de dados. Estas arquiteturas se mostraram eficientes para extrair características e classificar dados no formato do módulo da STFT e imagens do espectrograma. Para os dados do sinal bruto, apesar da arquitetura se mostrar eficiente para alguns sujeitos, para outros ela não obteve o mesmo sucesso. Neste caso, cabe uma investigação destas arquiteturas neste formato específico de dado. Os resultados da literatura indicam a possibilidade de melhorar esta abordagem, seja alterando a arquitetura, avaliando novas formas de regularização, ou treinando a mesma arquitetura com mais dados.

A terceira proposta era investigar a técnica de transferência de conhecimento aos dados de uma BCI SSVEP. Esta técnica se mostrou eficiente para um conjunto de usuários. Porém, para os demais usuários, esta abordagem ficou muito aquém de um desempenho minimamente satisfatório. Cabe aqui avaliar se outras arquiteturas consolidadas na tarefa de transferência de conhecimento podem ter desempenho melhor, ou se a quantidade de dados disponível na base de dados empregada neste trabalho é insuficiente, independente do modelo usado nesta abordagem.

A quarta proposta era investigar se uma função de custo diferente das comumente usadas para problema de classificação seria vantajosa neste problema. A ideia de uma função de custo que cria um agrupamento das classes e para uma nova amostra calcula qual a distância para estes agrupamentos, intuitivamente parece uma boa abordagem para se aplicar a dados de BCIs SSVEP. Porém, o uso da função de custo *triplet* se mostrou inferior em relação a tempo computacional de treinamento e complexidade na avaliação das amostras quando comparado com funções de custo comuns para classificação

em DL, como a entropia cruzada.

A quinta proposta era investigar se era possível conceber um esquema de treinamento mais robusto às DCNNs, capaz de generalizar aos dados de um usuário que não participou do treinamento. Para a base de dados usada neste trabalho, treinar qualquer uma das DCNNs para os dados de um único indivíduo se mostrou inviável. Por outro lado, usar os dados de vários indivíduos para treinar as arquiteturas e avaliá-las em um usuário que não participou do treinamento se mostrou eficiente, abrindo novas possibilidades de investigação. Um delas seria avaliar se é possível, para os dados dos usuários que obtiveram desempenho baixo, aplicar uma abordagem de ajuste fino com menos sessões de treinamento. O sucesso nesta tarefa pode diminuir drasticamente o tempo de treinamento de um novo usuário de um sistema BCI.

## Referências

- ABO-ZAHHAD, M.; AHMED, S. M.; ABBAS, S. N. State-of-the-art methods and future perspectives for personal recognition based on electroencephalogram signals. *IET Biometrics*, IET, v. 4, n. 3, p. 179–190, 2015. Citado na página 23.
- AFLALO, T.; KELLIS, S.; KLAES, C.; LEE, B.; SHI, Y.; PEJSA, K.; SHANFIELD, K.; HAYES-JACKSON, S.; AISEN, M.; HECK, C. *et al.* Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science*, American Association for the Advancement of Science, v. 348, n. 6237, p. 906–910, 2015. Citado 2 vezes nas páginas 15 e 23.
- ATTIA, M.; HETTIARACHCHI, I.; HOSSNY, M.; NAHAVANDI, S. A time domain classification of steady-state visual evoked potentials using deep recurrent-convolutional neural networks. In: IEEE. *2018 15th International Symposium on Biomedical Imaging (ISBI 2018)*. [S.l.], 2018. p. 766–769. Citado na página 49.
- ATTIA, M.; HETTIARACHCHI, I.; MOHAMED, S.; HOSSNY, M.; NAHAVANDI, S. A frequency domain classifier of steady-state visual evoked potentials using deep separable convolutional neural networks. In: IEEE. *2018 International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.], 2018. p. 2134–2139. Citado 3 vezes nas páginas 49, 52 e 55.
- AZKAN, N. K. N.; ATAPOUR-ABARGHOUEI, A.; BONNER, S.; CONNOLLY, J.; MOUBAYED, N. A.; BRECKON, T. Simulating brain signals : creating synthetic EEG data via neural-based generative models for improved SSVEP classification. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2019. Citado na página 49.
- AZKAN, N. K. N.; BONNER, S.; CONNOLLY, J.; MOUBAYED, N. A.; BRECKON, T. On the Classification of SSVEP-Based Dry-EEG Signals via Convolutional Neural Networks. In: IEEE. *2018 International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.], 2018. p. 3726–3731. Citado 5 vezes nas páginas 48, 49, 52, 53 e 55.
- AZKAN, N. K. N.; CONNOLLY, J. D.; MOUBAYED, N. A.; BRECKON, T. P. Using variable natural environment brain-computer interface stimuli for real-time humanoid robot navigation. In: *2019 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2019. Citado 2 vezes nas páginas 48 e 49.
- BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. *Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 35, n. 8, p. 1798–1828, 2013. Citado na página 41.
- BENGIO, Y.; SIMARD, P.; FRASCONI, P. *et al.* Learning long-term dependencies with gradient descent is difficult. *Transactions on Neural Networks*, IEEE, v. 5, n. 2, p. 157–166, 1994. Citado na página 59.
- BEVILACQUA, V.; TATTOLI, G.; BUONGIORNO, D.; LOCONSOLE, C.; LEONARDIS, D.; BARSOTTI, M.; FRISOLI, A.; BERGAMASCO, M. A novel

- BCI-SSVEP based approach for control of walking in virtual environment using a convolutional neural network. In: IEEE. *2014 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2014. p. 4121–4128. Citado na página 47.
- BISHOP, C. M. Regularization and complexity control in feed-forward networks. In: *Proceedings International Conference on Artificial Neural Networks ICANN*. [S.l.: s.n.], 1995. v. 95, p. 141–148. Citado 2 vezes nas páginas 28 e 40.
- BONNET, L.; LOTTE, F.; LÉCUYER, A. Two brains, one game: design and evaluation of a multiuser BCI video game based on motor imagery. *Transactions on Computational Intelligence and AI in games*, IEEE, v. 5, n. 2, p. 185–198, 2013. Citado 2 vezes nas páginas 15 e 23.
- BORISOFF, J. F.; MASON, S. G.; BASHASHATI, A.; BIRCH, G. E. Brain-computer interface design for asynchronous control applications: improvements to the LF-ASD asynchronous brain switch. *Transactions on Biomedical Engineering*, IEEE, v. 51, n. 6, p. 985–992, 2004. Citado na página 29.
- BRUNNER, C.; ALLISON, B. Z.; KRUSIENSKI, D. J.; KAISER, V.; MÜLLER-PUTZ, G. R.; PFURTSCHELLER, G.; NEUPER, C. Improved signal processing approaches in an offline simulation of a hybrid brain–computer interface. *Journal of Neuroscience Methods*, Elsevier, v. 188, n. 1, p. 165–173, 2010. Citado na página 26.
- BURGES, C. J. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, Springer, v. 2, n. 2, p. 121–167, 1998. Citado na página 28.
- CARVALHO, S. N.; COSTA, T. B.; URIBE, L. F.; SORIANO, D. C.; YARED, G. F.; CORADINE, L. C.; ATTUX, R. Comparative analysis of strategies for feature extraction and classification in SSVEP BCIs. *Biomedical Signal Processing and Control*, Elsevier, v. 21, p. 34–42, 2015. Citado 5 vezes nas páginas 16, 17, 27, 52 e 53.
- CARVALHO, S. N.; COSTA, T. B.; URIBE, L. F.; SORIANO, D. C.; ALMEIDA, S. R.; MIN, L. L.; CASTELLANO, G.; ATTUX, R. Effect of the combination of different numbers of flickering frequencies in an SSVEP-BCI for healthy volunteers and stroke patients. In: IEEE. *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*. [S.l.], 2015. p. 78–81. Citado 6 vezes nas páginas 10, 53, 54, 55, 57 e 66.
- CECOTTI, H. A time–frequency convolutional neural network for the offline classification of steady-state visual evoked potential responses. *Pattern Recognition Letters*, Elsevier, v. 32, n. 8, p. 1145–1153, 2011. Citado 2 vezes nas páginas 46 e 47.
- CECOTTI, H. Convolutional neural networks for event-related potential detection: impact of the architecture. In: IEEE. *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. [S.l.], 2017. p. 2031–2034. Citado 2 vezes nas páginas 47 e 53.
- CECOTTI, H.; GRAESER, A. Convolutional neural network with embedded Fourier transform for EEG classification. In: IEEE. *2008 19th International Conference on Pattern Recognition*. [S.l.], 2008. p. 1–4. Citado 2 vezes nas páginas 46 e 52.

- CHAN, A. T.; QUIROZ, J. C.; DASCALU, S.; HARRIS, F. C. An overview of brain computer interfaces. In: *Proc. 30th Int. Conf. on Computers and Their Applications*. [S.l.: s.n.], 2015. Citado na página 22.
- CHANG, M. H.; PARK, K. S. Frequency recognition methods for dual-frequency SSVEP based brain-computer interface. In: IEEE. *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. [S.l.], 2013. p. 2220–2223. Citado na página 27.
- CHATRIAN, G.; LETTICH, E.; NELSON, P. Ten percent electrode system for topographic studies of spontaneous and evoked EEG activities. *American Journal of EEG Technology*, Taylor & Francis, v. 25, n. 2, p. 83–92, 1985. Citado 3 vezes nas páginas 10, 21 e 22.
- COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, JSTOR, v. 19, n. 90, p. 297–301, 1965. Citado na página 27.
- COYLE, S. M.; WARD, T. E.; MARKHAM, C. M. Brain–computer interface using a simplified functional near-infrared spectroscopy system. *Journal of neural engineering*, IOP Publishing, v. 4, n. 3, p. 219, 2007. Citado na página 21.
- DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 Conference on Computer Vision and Pattern Recognition*. [S.l.], 2009. p. 248–255. Citado na página 34.
- DIEN, J. Issues in the application of the average reference: review, critiques, and recommendations. *Behavior Research Methods, Instruments, & Computers*, Springer, v. 30, n. 1, p. 34–43, 1998. Citado na página 26.
- DONATI, A. R.; SHOKUR, S.; MORYA, E.; CAMPOS, D. S.; MOIOLI, R. C.; GITTI, C. M.; AUGUSTO, P. B.; TRIPODI, S.; PIRES, C. G.; PEREIRA, G. A. *et al.* Long-term training with a brain-machine interface-based gait protocol induces partial neurological recovery in paraplegic patients. *Scientific Reports*, Nature Publishing Group, v. 6, p. 30383, 2016. Citado na página 15.
- DORNHEGE, G.; MILLAN, J. d. R.; HINTERBERGER, T.; MCFARLAND, D. J.; MÜLLER, K.-R. *Toward brain-computer interfacing*. [S.l.]: MIT Press, 2007. Citado 3 vezes nas páginas 15, 26 e 28.
- DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, v. 12, n. Jul, p. 2121–2159, 2011. Citado na página 33.
- DUHAMEL, P.; VETTERLI, M. Fast Fourier transforms: a tutorial review and a state of the art. *Signal Processing*, Elsevier, v. 19, n. 4, p. 259–299, 1990. Citado na página 27.
- EL-FIQI, H.; WANG, M.; SALIMI, N.; KASMARIK, K.; BARLOW, M.; ABBASS, H. Convolution neural networks for person identification and verification using steady state visual evoked potential. In: IEEE. *2018 International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.], 2018. p. 1062–1069. Citado 2 vezes nas páginas 51 e 53.

- FABIANI, G. E.; MCFARLAND, D. J.; WOLPAW, J. R.; PFURTSCHELLER, G. Conversion of EEG activity into cursor movement by a brain-computer interface (BCI). *Transactions on Neural Systems and Rehabilitation Engineering*, IEEE, v. 12, n. 3, p. 331–338, 2004. Citado 2 vezes nas páginas 15 e 27.
- FARWELL, L. A.; DONCHIN, E. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, Elsevier, v. 70, n. 6, p. 510–523, 1988. Citado na página 23.
- FISHER, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936. Citado na página 28.
- FUKUSHIMA, K. Cognitron: a self-organizing multilayered neural network. *Biological Cybernetics*, Springer, v. 20, n. 3-4, p. 121–136, 1975. Citado 2 vezes nas páginas 33 e 34.
- GALÁN, F.; NUTTIN, M.; LEW, E.; FERREZ, P. W.; VANACKER, G.; PHILIPS, J.; MILLÁN, J. d. R. A brain-actuated wheelchair: asynchronous and non-invasive brain-computer interfaces for continuous control of robots. *Clinical Neurophysiology*, Elsevier, v. 119, n. 9, p. 2159–2169, 2008. Citado na página 29.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. *Deep learning*. [S.l.]: MIT Press Cambridge, 2016. v. 1. Citado 10 vezes nas páginas 16, 30, 33, 35, 37, 38, 39, 41, 43 e 53.
- GRAIMANN, B.; ALLISON, B.; PFURTSCHELLER, G. Brain-computer interfaces: a gentle introduction. In: *Brain-Computer Interfaces*. [S.l.]: Springer, 2009. p. 1–27. Citado 2 vezes nas páginas 15 e 16.
- GROSSE-WENTRUP, M.; BUSS, M. Multiclass common spatial patterns and information theoretic feature extraction. *Transactions on Biomedical Engineering*, IEEE, v. 55, n. 8, p. 1991–2000, 2008. Citado na página 27.
- GUGER, C.; HARKAM, W.; HERTNAES, C.; PFURTSCHELLER, G. Prosthetic control by an EEG-based brain-computer interface (BCI). In: CITESEER. *Proc. 5th European Conference for the Advancement of Assistive Technology*. [S.l.], 1999. p. 3–6. Citado na página 15.
- GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, v. 3, p. 1157–1182, 2003. Citado na página 28.
- HAMEDI, M.; SALLEH, S.-H.; NOOR, A. M. Electroencephalographic motor imagery brain connectivity analysis for BCI: a review. *Neural Computation*, MIT Press, v. 28, n. 6, p. 999–1041, 2016. Citado na página 23.
- HATCHER, W. G.; YU, W. A survey of deep learning: platforms, applications and emerging research trends. *Access*, IEEE, v. 6, p. 24411–24432, 2018. Citado 3 vezes nas páginas 16, 30 e 33.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 770–778. Citado 8 vezes nas páginas 10, 12, 34, 53, 59, 60, 61 e 62.

HEBB, D. The organization of behavior. a neuropsychological theory. John Wiley, 1949. Citado na página 30.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. *Neural Computation*, MIT Press, v. 18, n. 7, p. 1527–1554, 2006. Citado na página 33.

HINTON, G. E.; SRIVASTAVA, N.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. Citado na página 39.

HOCHREITER, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, World Scientific, v. 6, n. 02, p. 107–116, 1998. Citado na página 59.

HUANG, G. B.; MATTAR, M.; BERG, T.; LEARNED-MILLER, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In: *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*. [S.l.: s.n.], 2008. Citado na página 43.

HUBEL, D. H.; WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, Wiley Online Library, v. 160, n. 1, p. 106–154, 1962. Citado na página 34.

HUBEL, D. H.; WIESEL, T. N. Ferrier lecture-functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, The Royal Society London, v. 198, n. 1130, p. 1–59, 1977. Citado na página 34.

HYVÄRINEN, A.; OJA, E. Independent component analysis: algorithms and applications. *Neural networks*, Elsevier, v. 13, n. 4-5, p. 411–430, 2000. Citado na página 39.

IOFFE, S.; SZEGEDY, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. Citado 2 vezes nas páginas 39 e 40.

JASPER, H. H. The ten-twenty electrode system of the International Federation. *Electroencephalogr. Clin. Neurophysiol.*, v. 10, p. 370–375, 1958. Citado na página 21.

JOHANSSON, B.; JAKOBSSON, P. Fourier analysis of steady-state visual evoked potentials in subjects with normal and defective stereo vision. *Documenta Ophthalmologica*, Springer, v. 101, n. 3, p. 233–246, 2000. Citado na página 24.

KAPLAN, A. Y.; SHISHKIN, S. L.; GANIN, I. P.; BASYUL, I. A.; ZHIGALOV, A. Y. Adapting the P300-based brain-computer interface for gaming: a review. *Transactions on Computational Intelligence and AI in Games*, IEEE, v. 5, n. 2, p. 141–149, 2013. Citado 2 vezes nas páginas 15 e 23.

KELLY, S. P.; LALOR, E. C.; REILLY, R. B.; FOXE, J. J. Visual spatial attention tracking using high-density ssvp data for independent brain-computer communication. *Transactions on Neural Systems and Rehabilitation Engineering*, IEEE, v. 13, n. 2, p. 172–178, 2005. Citado na página 23.

- KINGMA, D. P.; BA, J. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 33.
- KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. *Artificial Intelligence*, Elsevier, v. 97, n. 1-2, p. 273–324, 1997. Citado na página 28.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2012. p. 1097–1105. Citado 3 vezes nas páginas 32, 33 e 34.
- KURISCAK, E.; MARSALEK, P.; STROFFEK, J.; TOTH, P. G. Biological context of Hebb learning in artificial neural networks, a review. *Neurocomputing*, Elsevier, v. 152, p. 27–35, 2015. Citado na página 30.
- KWAK, N.-S.; MÜLLER, K.-R.; LEE, S.-W. A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. *PloS One*, Public Library of Science, v. 12, n. 2, p. e0172578, 2017. Citado 2 vezes nas páginas 48 e 52.
- LEARNED-MILLER, E.; HUANG, G. B.; ROYCHOWDHURY, A.; LI, H.; HUA, G. Labeled faces in the wild: a survey. In: *Advances in Face Detection and Facial Image Analysis*. [S.l.]: Springer, 2016. p. 189–248. Citado na página 43.
- LECUN, Y.; BENGIO, Y. The handbook of brain theory and neural networks. In: ARBIB, M. A. (Ed.). Cambridge, MA, USA: MIT Press, 1998. cap. Convolutional Networks for Images, Speech, and Time Series, p. 255–258. ISBN 0-262-51102-9. Citado 4 vezes nas páginas 34, 36, 37 e 38.
- LECUN, Y. *et al.* Generalization and network design strategies. In: *Connectionism in Perspective*. [S.l.]: Citeseer, 1989. v. 19. Citado na página 34.
- LEE, Y.; OH, S.-H.; KIM, M. W. An analysis of premature saturation in back propagation learning. *Neural Networks*, Elsevier, v. 6, n. 5, p. 719–728, 1993. Citado na página 33.
- LEEB, R.; FRIEDMAN, D.; MÜLLER-PUTZ, G. R.; SCHERER, R.; SLATER, M.; PFURTSCHELLER, G. Self-paced (asynchronous) BCI control of a wheelchair in virtual environments: a case study with a tetraplegic. *Computational Intelligence and Neuroscience*, Hindawi, v. 2007, 2007. Citado 2 vezes nas páginas 15 e 29.
- LEI, H.; HAN, T.; ZHOU, F.; YU, Z.; QIN, J.; ELAZAB, A.; LEI, B. A deeply supervised residual network for HEP-2 cell classification via cross-modal transfer learning. *Pattern Recognition*, Elsevier, v. 79, p. 290–302, 2018. Citado na página 60.
- LIANG, P.; BOSE, N. Neural network fundamentals with graphs, algorithms and applications. *Mc Graw-Hill*, 1996. Citado na página 33.
- MALLAT, S. *A wavelet tour of signal processing*. [S.l.]: Elsevier, 1999. Citado na página 27.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943. Citado na página 30.

- MCFARLAND, D. J.; WOLPAW, J. R. Brain-computer interfaces for communication and control. *Communications of the ACM*, NIH Public Access, v. 54, n. 5, p. 60, 2011. Citado na página 26.
- MELLINGER, J.; SCHALK, G.; BRAUN, C.; PREISSEL, H.; ROSENSTIEL, W.; BIRBAUMER, N.; KÜBLER, A. An MEG-based brain-computer interface (BCI). *Neuroimage*, Elsevier, v. 36, n. 3, p. 581–593, 2007. Citado na página 21.
- MIDDENDORF, M.; MCMILLAN, G.; CALHOUN, G.; JONES, K. S. Brain-computer interfaces based on the steady-state visual-evoked response. *Transactions on Rehabilitation Engineering*, IEEE, v. 8, n. 2, p. 211–214, 2000. Citado na página 24.
- MILLAN, J. R.; MOURIÑO, J. Asynchronous BCI and local neural classifiers: an overview of the adaptive brain interface project. *Transactions on Neural Systems and Rehabilitation Engineering*, IEEE, v. 11, n. 2, p. 159–161, 2003. Citado na página 29.
- MULERT, C.; LEMIEUX, L. *EEG-fMRI: physiological basis, technique, and applications*. [S.l.]: Springer Science & Business Media, 2009. Citado na página 22.
- NGUYEN, T.-H.; CHUNG, W.-Y. A single-channel SSVEP-based BCI speller using deep learning. *Access*, IEEE, v. 7, p. 1752–1763, 2018. Citado 4 vezes nas páginas 50, 52, 53 e 55.
- NIBALI, A.; HE, Z.; WOLLERSHEIM, D. Pulmonary nodule classification with deep residual networks. *International Journal of Computer Assisted Radiology and Surgery*, Springer, v. 12, n. 10, p. 1799–1808, 2017. Citado na página 60.
- NICOLAS-ALONSO, L. F.; GOMEZ-GIL, J. Brain computer interfaces, a review. *Sensors*, Molecular Diversity Preservation International, v. 12, n. 2, p. 1211–1279, 2012. Citado 2 vezes nas páginas 15 e 20.
- NICOLELIS, M. L. Actions from thoughts. *Nature*, Nature Publishing Group, v. 409, n. 6818, p. 403, 2001. Citado na página 20.
- NIDAL, K.; MALIK, A. S. *EEG/ERP analysis: methods and applications*. [S.l.]: CRC Press, 2014. Citado na página 21.
- NOLAN, H.; WHELAN, R.; REILLY, R. FASTER: fully automated statistical thresholding for EEG artifact rejection. *Journal of Neuroscience Methods*, Elsevier, v. 192, n. 1, p. 152–162, 2010. Citado na página 22.
- OH, K.-S.; JUNG, K. GPU implementation of neural networks. *Pattern Recognition*, Elsevier, v. 37, n. 6, p. 1311–1314, 2004. Citado na página 34.
- OPPENHEIM, A. V.; VERGHESE, G. C. *Signals, systems and inference*. [S.l.]: Pearson, 2015. Citado na página 27.
- O'SHEA, T. J.; ROY, T.; CLANCY, T. C. Over-the-air deep learning based radio signal classification. *Journal of Selected Topics in Signal Processing*, IEEE, v. 12, n. 1, p. 168–179, 2018. Citado na página 60.
- PAN, S. J.; YANG, Q. A survey on transfer learning. *Transactions on Knowledge and Data Engineering*, IEEE, v. 22, n. 10, p. 1345–1359, 2009. Citado na página 41.

- PATTERSON, J.; GIBSON, A. *Deep learning: A practitioner's approach*. [S.l.]: "O'Reilly Media, Inc.", 2017. Citado 3 vezes nas páginas 16, 30 e 52.
- PODMORE, J. J.; BRECKON, T. P.; AZNAN, N. K.; CONNOLLY, J. D. *et al.* On the relative contribution of deep convolutional neural networks for SSVEP-based bio-signal decoding in BCI speller applications. *Transactions on Neural Systems and Rehabilitation Engineering*, IEEE, 2019. Citado 4 vezes nas páginas 50, 52, 53 e 55.
- POSTELNICU, C.-C.; TALABA, D. P300-based brain-neuronal computer interaction for spelling applications. *Transactions on Biomedical Engineering*, IEEE, v. 60, n. 2, p. 534–543, 2012. Citado 2 vezes nas páginas 15 e 24.
- PUENGDANG, S.; TUAROB, S.; SATTABONGKOT, T.; SAKBOONYARAT, B. EEG-based person authentication method using deep learning with visual stimulation. In: IEEE. *2019 11th International Conference on Knowledge and Smart Technology (KST)*. [S.l.], 2019. p. 6–10. Citado na página 51.
- RAMOSER, H.; MULLER-GERKING, J.; PFURTSCHELLER, G. Optimal spatial filtering of single trial EEG during imagined hand movement. *Transactions on Rehabilitation Engineering*, IEEE, v. 8, n. 4, p. 441–446, 2000. Citado na página 27.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citado na página 30.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, Nature Publishing Group, v. 323, n. 6088, p. 533, 1986. Citado 2 vezes nas páginas 33 e 34.
- SCHALK, G.; LEUTHARDT, E. C. Brain-computer interfaces using electrocorticographic signals. *Reviews in Biomedical Engineering*, IEEE, v. 4, p. 140–154, 2011. Citado na página 20.
- SCHERER, D.; MÜLLER, A.; BEHNKE, S. Evaluation of pooling operations in convolutional architectures for object recognition. In: SPRINGER. *International Conference on Artificial Neural Networks*. [S.l.], 2010. p. 92–101. Citado na página 38.
- SCHMIDHUBER, J. Deep learning in neural networks: an overview. *Neural Networks*, Elsevier, v. 61, p. 85–117, 2015. Citado na página 34.
- SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 815–823. Citado 7 vezes nas páginas 10, 43, 44, 45, 53, 61 e 79.
- SETARE, A.; 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. Citado na página 23.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Citado 2 vezes nas páginas 34 e 49.

- SJÖBERG, J.; LJUNG, L. Overtraining, regularization and searching for a minimum, with application to neural networks. *International Journal of Control*, Taylor & Francis, v. 62, n. 6, p. 1391–1407, 1995. Citado na página 41.
- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citado 2 vezes nas páginas 38 e 39.
- TAVARES, T. F.; RIMOLDI, G.; PONTES, V. E.; MANZOLLI, J. Cooperative musical creation using Kinect, WiiMote, Epoc and microphones: a case study with MinDSounds. In: CITESEER. *IEEE International Conference on Systems, Man, and Cybernetics*. [S.l.], 2015. Citado na página 15.
- THOMAS, J.; MASZCZYK, T.; SINHA, N.; KLUGE, T.; DAUWELS, J. Deep learning-based classification for brain-computer interfaces. In: IEEE. *2017 International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.], 2017. p. 234–239. Citado 2 vezes nas páginas 47 e 66.
- VIDAL, J. J. Toward direct brain-computer communication. *Annual Review of Biophysics and Bioengineering*, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 2, n. 1, p. 157–180, 1973. Citado na página 19.
- VON ZUBEN, F. J.; VERONEZE, R. *Deep Learning: parte I*. 2019. Notas de aula do curso de pós-graduação IA353 - Redes Neurais, Primeiro Semestre, FEEC, Unicamp. Citado na página 35.
- WAYTOWICH, N.; LAWHERN, V. J.; GARCIA, J. O.; CUMMINGS, J.; FALLER, J.; SAJDA, P.; VETTEL, J. M. Compact convolutional neural networks for classification of asynchronous steady-state visual evoked potentials. *Journal of Neural Engineering*, IOP Publishing, v. 15, n. 6, p. 066031, 2018. Citado 4 vezes nas páginas 50, 52, 53 e 55.
- WEISKOPF, N.; MATHIAK, K.; BOCK, S. W.; SCHARNOWSKI, F.; VEIT, R.; GRODD, W.; GOEBEL, R.; BIRBAUMER, N. Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI). *Transactions on Biomedical Engineering*, IEEE, v. 51, n. 6, p. 966–970, 2004. Citado na página 21.
- WERBOS, P. Beyond regression: New tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*, 1974. Citado na página 33.
- WOLPAW, J. R.; BIRBAUMER, N.; MCFARLAND, D. J.; PFURTSCHELLER, G.; VAUGHAN, T. M. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, Elsevier, v. 113, n. 6, p. 767–791, 2002. Citado 6 vezes nas páginas 10, 15, 16, 19, 20 e 52.
- YOSINSKI, J.; CLUNE, J.; BENGIO, Y.; LIPSON, H. How transferable are features in deep neural networks? In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2014. p. 3320–3328. Citado 4 vezes nas páginas 42, 43, 53 e 73.
- YU, T.; WEI, C.-S.; CHIANG, K.-J.; NAKANISHI, M.; JUNG, T.-P. EEG-based user authentication using a convolutional neural network. In: IEEE. *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*. [S.l.], 2019. p. 1011–1014. Citado na página 51.

ZHANG, R.; LI, Y.; YAN, Y.; ZHANG, H.; WU, S.; YU, T.; GU, Z. Control of a wheelchair in an indoor environment based on a brain–computer interface and automated navigation. *Transactions on Neural Systems and Rehabilitation Engineering*, IEEE, v. 24, n. 1, p. 128–139, 2015. Citado na página 23.

ZHANG, X.; XU, G.; MOU, X.; RAVI, A.; LI, M.; WANG, Y.; JIANG, N. A convolutional neural network for the detection of asynchronous steady state motion visual evoked potential. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, IEEE, 2019. Citado 2 vezes nas páginas 51 e 53.

ZHOU, Y.-T.; CHELLAPPA, R. Computation of optical flow using a neural network. In: *IEEE International Conference on Neural Networks*. [S.l.: s.n.], 1988. v. 1998, p. 71–78. Citado na página 38.

# ANEXO A – Tabelas de Resultados Numéricos

Este Anexo apresenta as tabelas com os resultados numéricos relacionados aos gráficos reportados no Capítulo 6.

## A.1 Resultados Numéricos dos Experimentos com Espectrogramas como Entrada da Rede

Tabelas de resultados referentes ao experimento descrito na Seção 6.1.2.

Sujeito	1	2	3	4	5	6	7	8	9	10
Rede no espectrograma	57,66	77,03	58,91	63,44	82,50	76,56	68,28	82,03	74,69	77,81
Linear no espectrograma	59,00	76,00	60,00	60,00	81,00	77,00	65,00	83,00	72,00	78,00
Soma das amplitudes	46,25	66,67	55,42	50,83	89,79	68,75	57,08	72,08	67,92	63,33

Tabela 2 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de espectrogramas sem harmônica superior usando informação dos eletrodos O1, O2, Oz, POz.

Sujeito	1	2	3	4	5	6	7	8	9	10
Rede no espectrograma	56,88	95,00	62,50	62,50	98,12	90,62	67,50	91,88	84,38	87,50
Linear no espectrograma	58,00	95,00	65,00	61,00	99,00	91,00	65,00	90,00	80,00	85,00
Soma das amplitudes	51,88	80,00	55,00	51,88	97,50	79,38	61,88	80,00	70,00	68,12

Tabela 3 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados de espectrogramas sem harmônica superior usando informação do eletrodo Oz.

## A.2 Resultados Numéricos dos Experimentos com Módulos da STFT como Entrada da Rede

Tabelas de resultados referentes ao experimento descrito na Seção 6.1.3.

Sujeito	1	2	3	4	5	6	7	8	9	10
Rede no módulo da STFT	57,66	77,50	56,72	62,97	82,50	75,16	66,41	83,59	72,34	76,09
Linear no módulo da STFT	59,00	76,00	63,00	61,00	81,00	77,00	66,00	83,00	72,00	79,00
Soma das amplitudes	46,25	66,67	55,42	50,83	89,79	68,75	57,08	72,08	67,92	63,33

Tabela 4 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados dos módulos da STFT usando informação dos eletrodos O1, O2, Oz, POz.

Sujeito	1	2	3	4	5	6	7	8	9	10
Rede no módulo da STFT	54,38	94,38	56,88	65,00	96,25	91,25	65,00	91,25	81,88	83,12
Linear no módulo da STFT	55,00	95,00	68,00	60,00	99,00	91,00	66,00	89,00	78,00	83,00
Soma das amplitudes	51,88	80,00	55,00	51,88	97,50	79,38	61,88	80,00	70,00	68,12

Tabela 5 – Acurácia da DCNN, da SVM linear e da soma das amplitudes na base de dados dos módulos da STFT usando informação do eletrodo Oz.

### A.3 Resultados Numéricos dos Experimentos com Sinais Brutos como Entrada da Rede

Tabelas de resultados referentes ao experimento descrito na Seção 6.1.4.

Sujeito	1	2	3	4	5	6	7	8	9	10
Rede no bruto	56,40	78,75	54,53	66,40	80,62	62,03	72,03	86,25	57,96	65,15
Linear no bruto	50,00	47,00	52,00	49,00	53,00	47,00	54,00	53,00	50,00	52,00

Tabela 6 – Acurácia da DCNN e da SVM linear na base de dados de sinais brutos usando informação dos eletrodos O1, O2, Oz, POz.

Sujeito	1	2	3	4	5	6	7	8	9	10
Rede no bruto	50,00	92,50	54,37	50,62	98,12	65,62	71,87	88,12	53,75	53,12
Linear no bruto	51,00	49,00	51,00	50,00	59,00	45,00	57,00	53,00	41,00	58,00

Tabela 7 – Acurácia da DCNN e da SVM linear na base de dados de sinais brutos usando informação do eletrodo Oz.

### A.4 Resultados Numéricos dos Experimentos com Transferência de Conhecimento

Tabelas de resultados referentes ao experimento descrito na Seção 6.2.

Sujeito	1	2	3	4	5	6	7	8	9	10
ResNet-18	54,68	67,50	56,56	57,81	75,93	73,12	66,71	77,50	72,65	69,21
ResNet-34	57,34	69,06	54,06	61,09	77,03	72,81	63,75	79,37	71,56	67,34
ResNet-50	53,12	69,68	57,96	57,50	77,50	69,68	65,00	79,06	69,21	69,06
ResNet-101	54,37	77,50	56,25	63,12	79,21	74,37	67,96	77,50	70,46	70,00
ResNet-152	57,18	71,09	55,46	57,81	76,56	72,65	65,78	75,78	67,81	67,65

Tabela 8 – Acurácias dos diferentes arranjos da arquitetura ResNet na base de dados de espectrogramas sem harmônica superior usando informação dos eletrodos O1, O2, Oz, POz.

Sujeito	1	2	3	4	5	6	7	8	9	10
<b>ResNet-18</b>	55,62	91,25	58,12	56,25	93,12	86,25	63,12	90,00	82,50	86,87
<b>ResNet-34</b>	58,75	92,50	56,25	55,00	95,00	84,37	70,00	85,62	83,75	90,62
<b>ResNet-50</b>	61,87	91,87	55,62	53,75	94,37	84,37	65,00	89,37	81,25	83,75
<b>ResNet-101</b>	61,87	90,62	61,87	56,25	96,87	87,50	65,00	85,00	85,62	83,75
<b>ResNet-152</b>	60,62	86,87	58,12	53,75	95,62	91,25	63,75	85,00	83,75	84,37

Tabela 9 – Acurácias dos diferentes arranjos da arquitetura ResNet na base de dados de espectrogramas sem harmônica superior usando informação do eletrodo Oz.

## A.5 Resultados Numéricos dos Experimentos com a Arquitetura DCNN e a Função de Custo Triplet

Tabelas de resultados referentes ao experimento descrito na Seção 6.3.

Sujeito	1	2	3	4	5	6	7	8	9	10
<b>Rede no espectrograma com harmônica</b>	51,72	75,31	55,94	57,97	80,00	72,66	66,25	79,22	71,56	77,81
<b>Linear no espectrograma com harmônica</b>	58,00	77,00	62,00	64,00	83,00	75,00	66,00	83,00	71,00	78,00
<b>Rede no espectrograma sem harmônica</b>	52,03	76,72	55,47	57,50	82,34	75,16	68,13	82,34	73,91	77,19
<b>Linear no espectrograma sem harmônica</b>	59,00	76,00	60,00	60,00	81,00	77,00	65,00	83,00	72,00	78,00

Tabela 10 – Acurácia da DCNN com função de custo *triplet* e da SVM linear na base de dados de espectrogramas sem harmônica superior e com uma harmônica superior usando informação dos eletrodos O1, O2, Oz, POz.

Sujeito	1	2	3	4	5	6	7	8	9	10
<b>Rede no espectrograma com harmônica</b>	58,75	94,38	65,63	61,25	95,63	90,63	64,38	88,75	79,38	86,25
<b>Linear no espectrograma com harmônica</b>	61,00	96,00	68,00	65,00	96,00	91,00	70,00	89,00	73,00	84,00
<b>Rede no espectrograma sem harmônica</b>	61,25	94,38	53,75	52,50	96,25	89,38	63,75	89,38	83,13	88,75
<b>Linear no espectrograma sem harmônica</b>	58,00	95,00	65,00	61,00	99,00	91,00	65,00	90,00	80,00	85,00

Tabela 11 – Acurácia da DCNN com função de custo *triplet* e da SVM linear na base de dados de espectrogramas sem harmônica superior e com uma harmônica superior usando informação do eletrodo Oz.