Uso de Redes Neurais do Tipo Transformers para Classificação de Estágios do Sono

Resumo: O processo de classificação de estágios do sono é demorado e intensivo e requer especialistas para analisar horas de sinais de EEG. Além disso, está sujeito a erros subjetivos de cada especialista. O uso de rede neurais profundas vem sendo demonstrado resultados promissores, com diversas arquiteturas sendo utilizadas como redes Convolutivas, Long-Short Term Memories (LSTMs) e, mais recentemente, Transformers. Porém, adultos de diferentes idades podem ter diferentes perfis de sono, incluindo duração dos estágios e características do sinal.

Neste trabalho, utilizaremos Transformers para a classificação de sono. Utilizaremos aprendizado auto-supervisionado para permitir que o Transformer aprenda a extrair características relevantes do sinal e gere uma representação que possa ser posteriormente utilizada por um classificador. Utilizaremos esta representação como entrada para um classificador, que pode ser ajustado para diferentes perfis de sujeitos. Iniciaremos o projeto com o uso de um eletrodo e, posteriormente, utilizaremos múltiplos eletrodos. Utilizaremos bases abertas de dados de competições de BCI para realizar o treinamento e avaliação das redes.

Palavras-chave:

Interface Cérebro-Máquina, Redes Neurais Recorrentes, Aprendizado de Máquina.

1 Introdução

O processo de classificação de estágios do sono é demorado e intensivo e requer especialistas para analisar horas de sinais de EEG para cada noite de cada indivíduo. Além disso, está sujeito a erros subjetivos de cada especialista, que podem classificar trechos de um mesmo sinal de modo diferente. O uso de rede neurais profundas vem sendo demonstrado resultados promissores, com diversas arquiteturas sendo utilizadas como redes Convolutivas [14, 10], Long-Short Term Memories (LSTMs) [3] e, mais recentemente, Transformers [6]. A maior dificuldade é que o treinamento dessas redes requer um número muito grande de exemplos rotulados. Isto ocorre porque é necessário a aprender os pesos de um grande número de conexões entre neurônios. Além disso, o espaço de entrada possui alta-dimensionalidade, pois cada as séries temporais possuem milhares de amostras, e a razão sinal ruído é muito pequena em sinais de EEG.

O uso de métodos auto-supervisionados, que permitem extrair propriedades das entradas sem a necessidade de rótulos [11], vem se tornando cada vez mais comum na área de aprendizado profundo. A ideia é criar tarefas de predição envolvendo os dados originais. Por exemplo, em aplicações de Processamento de Linguagem Natural, pode-se eliminar alguma palavra de uma frase e usar a rede para prever qual palavra foi eliminada [5]. No caso de um vídeo, pode-se tentar prever os próximos quadros [18], enquanto em uma imagem, pode-se prever se a imagem foi rotacionada ou não [18]. No caso de EEG, diversos trabalhos já foram publicados, utilizando-se de técnicas como prever se um trecho do sinal se encontra próximo de outro e em qual posição relativa [2, 12].

Outro problema existente no caso de EEGs é a baixa disponibilidade de dados em algumas classes de indivíduos. Por exemplo, no caso de sono, pode ser que tenhamos

poucos dados de indivíduos acima de acima de 80 anos, mas muitos dados em outras idades. Este problema é reduzido com a Transferência de Aprendizado [17, 21, 16, 13], que é popular em múltiplos domínios de redes neurais, e que no caso específico de classificação de sinais de EEG, começou a ser utilizada mais recentemente [20, 7, 22]. Com a transferência de aprendizado, queremos que o novo modelo seja facilmente adaptável a novos indivíduos com características diferentes, como no caso de indivíduos de idade mais avançada. Para estes novos indivíduos, disponibilizamos apenas uma pequena quantidade de dados de treinamento. A ideia é que podemos ter um grupo inicial de indivíduos que realizam longas coletas de dados, utilizados para treinar o modelo.

As redes neurais do tipo Transformers [19] geraram uma revolução na área de processamento de linguagem natural, com o Bert [5] e GPT-3 [4]. Este tipo de arquitetura passou a ser recentemente utilizado no reconhecimento de linguagem em fala, com a arquitetura Wav2Vec 2.0 [1] obtendo resultados excelentes. Nesta arquitetura, uma rede convolutiva é utilizada para transformar o sinal contínuo do som em uma sequência de códigos discretos, que é então processada por um Transformer. Para o treinamento, é utilizado o aprendizado auto-supervisionado, com a apresentação das entradas com algum dos códigos faltando e a tarefa da rede é gerar este código faltante. O uso de Transformers para classificação de estágios de sono foi avaliado recentemente por Eldele *et al.* [6] e Song *et al.* [15], ambos com o uso de aprendizado supervisionado, obtendo resultados do estado da arte. Já o uso de aprendizado auto-supervisionado para gerar representações do sinal de EEG foi avaliado por Kostas *et al.* [9], que focaram na habilidade de arquiteturas com o Transformer de gerarem representações que possam ser utilizadas em diferentes tarefas de classificação. Mas Kostas *et al.* não avaliaram se este tipo de geração de codificação funciona melhor que o aprendizado *end-to-end* com

aprendizado supervisionado em problemas específicos, como a classificação de estágios do sono.

Neste trabalho, utilizaremos redes convolutivas combinadas com Transformers para a classificação de sono. Utilizaremos aprendizado auto-supervisionado para permitir que o Transformer aprenda a extrair características relevantes do sinal e gere uma representação que possa ser posteriormente utilizada por um classificador. Utilizaremos esta representação como entrada para um classificador, que pode ser ajustado para diferentes perfis de sujeitos. Iniciaremos o projeto com o uso de um eletrodo e, posteriormente, utilizaremos múltiplos eletrodos. Utilizaremos bases abertas de dados de competições de BCI para realizar o treinamento e avaliação das redes.

2 Objetivos

Neste trabalho, utilizaremos aprendizado auto-supervisionado em arquiteturas do tipo Transformer para extração de características relevantes de sinais de EEG durante o sono. Utilizaremos esta representação como entrada para um classificador, que poderá ser ajustado para diferentes perfis de sujeitos, em um processo de transferência de aprendizado. O projeto é subdividido nos seguintes objetivos específicos:

- Desenvolvimento de uma arquitetura baseada em Transformers para extração de características relevantes de sinais de EEG;
- Treinamento deste modelo utilizando aprendizado auto-supervisionado com dados de sono;
- Avaliação do uso da representação gerada para a classificação de estágios do sono de diferentes indivíduos;

- Agrupamento dos indivíduos em grupos com sujeitos com características similares e avaliação do desempenho do classificador;
- Atualizar a arquitetura do modelo para o caso com mais de um eletrodo e avaliar seus desempenho;
- Comparar as arquiteturas desenvolvidas com outras do estado da arte presentes no braindecode.

3 Metodologia e Plano de trabalho

O projeto pode ser dividido em 3 partes: (1) Desenvolvimento de arquitetura baseada em Transformer; (2) Classificação de estágios do sono e (3) Agrupamentos de indivíduos para classificação. Nesta seção descrevemos a metodologia e plano de trabalho.

3.1 Desenvolvimento de arquitetura baseada em Transformer

Nesta fase o aluno irá estudar as bases de dados disponível, fazer o download dos datasets e colocá-los em um formato que possam ser facilmente utilizados. O aluno precisará obter os dados e configura-los de modo a criar: (i) as tarefas de aprendizado auto-supervisionado, e (ii) as tarefas de classificação. Usaremos a base de dados Sleep-EDF Database Expanded [8][1], que contém dados de polisonografia de 197 noites de sono de diferentes indivíduos, incluindo sinais de 2 canais de EEG. Os dados estão anotados, com a classificação de cada período do sono, de modo que a organização destes para a tarefa de classificação é simples de ser realizada. Usaremos estes dados também para gerar os exemplos de aprendizado auto-supervisionado.

[1] Disponível em `https://www.physionet.org/content/sleep-edfx/1.0.0/`.

Além de preparar a base de dados, o aluno irá implementar um modelo de Transformers [19], baseado na arquitetura Wav2Vec 2.0 [1]. Nessa arquitetura, feita para processamento de sinais sonoros, cada trecho do sinal é convertido em uma representação latente z (um vetor de valores), utilizando uma rede convolutiva, resultando em uma sequência de códigos. O objetivo da rede convolutiva é extrair características do sinal que sejam úteis para a classificação, além de realizar a discretização da série temporal de um modo que esta possa ser processada pelo Transformer. Esta sequência de códigos z é apresentado como entrada para um Transformer que, por sua vez, gera uma sequência de representações de contexto c, conforme mostrado na Figura 1.

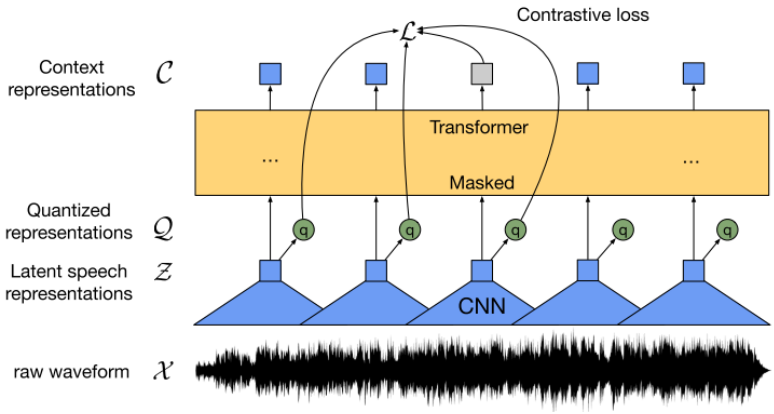

Figura 1: Arquiterua do Wav2Vec 2.0, com o sinal de audio na parte inferior, seguido pela rede convolutiva e do Transformer. Extraída de [1].

No aprendizado auto-supervisionado, apresentaremos as entradas com algum dos códigos faltando e a tarefa da rede é gerar este código faltante, de modo similar ao que é feito em arquiteturas de processamento de linguagem natural, como o BERT [5]. Como isso, o Transformer precisa aprender informações sobre a estrutura do sinal de modo a reconstruir as representações de contexto c perdidas.

O desenvolvimento será realizado utilizando a linguagem Python e a biblioteca Keras, que fornece uma interface de alto-nível para a implementação de redes neurais.

Usaremos uma das 2 implementações disponíveis do Wav2Vec 2.0, a oficial, que é parte da biblioteca FairSeq [2], ou a disponível no pavote Hugging Face [3]. O principal desafio nesta fase é a adaptação do modelo, inicialmente desenvolvido para processamento de fala, para sinais de EEG. Apesar de ambos serem séries temporais, quase todos os parâmetros, como tamanho das janelas e códigos, e as arquiteturas das redes convolutivas e do Transformers, precisarão ser otimizadas para os sinais de EEG.

O aluno utilizará a IDE PyCharm e os experimentos serão realizados utilizadas máquinas dedicadas com GPUs localizadas no laboratório L105, do bloco L, para os experimentos. No PyCharm, é possível realizar a codificação no computador local e a execução em um computador remoto, de modo que o aluno poderá realizar seu projeto tanto no laboratório L105 quanto em qualquer local onde possa conectar seu notebook pessoal.

Faremos a avaliação nos cenários utilizando apenas um ou os dois eletrodos fornecidos no conjunto de dados da Sleep-EDF Database Expanded. Para usar dois eletrodos temos duas alternativas, que são (i) combiná-los já com a rede convolutiva em uma representação z combinada, ou (ii) gerar uma representação z para cada eletrodo e combiná-la no Transformer. A abordagem (ii) dá maior flexibilidade, mas requer maior poder computacional, pois o número de parâmetros do Transformer cresce quadraticamente com o tamanho da entrada.

3.2 Classificação de estágios do sono

A rede Transformer recebe como entrada as representações quantizadas e gera uma sequência de representações de contexto c, conforme Figura 1. O classificador irá atuar

[2] `https://github.com/facebookresearch/fairseq`
[3] `https://huggingface.co/docs/transformers/model_doc/wav2vec2`

sobre estas representações c, que são vetores numéricos, classificando qual o estágio do sono para cada momento.

Testaremos 2 cenários para o treinamento deste classificador. O mais simples será usar um classificador simples como Linear Discriminant Analysis (LDA), Support Vector Machine (SVM) ou Random Florests (RF). Ele será treinado diretamente com os rótulos relativos a cada posição do sinal referente à representação de contexto c. A maior limitação desta abordagem é que o contexto c gerado pode não ser adequado para a classificação.

A segunda abordagem será utilizar uma rede neural na saída do Transformer para realizar a classificação. A vantagem é que neste caso é possível realizar um treinamento *end-to-end* a partir do sinal original. Os pesos do Transformer e da rede convolutiva seriam obtidos utilizando o aprendizado auto-supervisionado da etapa anterior. Além disso, uma vantagem do Transformer é justamente o fato de diferentes parte do sinal, representadas por diferentes representações z poderem ser utilizadas para gerar o contexto c.

Do mesmo modo que na etapa anterior, avaliaremos o uso de um ou dois eletrodos, com as duas abordagens para combiná-los, que é na rede convolutiva ou no Transformer.

3.3 Agrupamentos de indivíduos para classificação

Na terceira etapa iremos nos concentrar na transferência de aprendizado. Para tal, iremos treinar a rede em um grupo de indivíduos de idades até 65 anos e testá-lo em 2 grupos, (i) com idades de 65 a 80 anos, e (ii) acima de 80 anos, conforme Figura 2. Este desafio apareceu como parte do NeurIPS 2021 BEETL Competition: Benchmarks for EEG Transfer Learning.

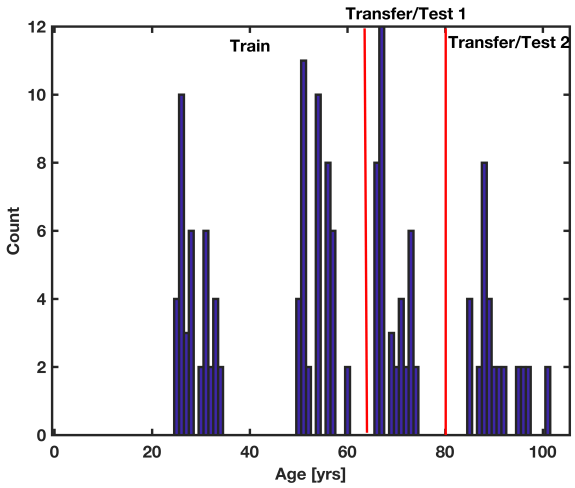

Figura 2: Distribuição de voluntários por idade. Extraído de https://beetl.ai/data.

A ideia é que com apenas um pequeno conjunto de dados desses novos conjuntos, seja possível classificar os indivíduos neles presentes. Para tal, utilizaremos as 2 abordagens da etapa anterior: (a) usar classificadores como LDA, SVM e RF diretamente nas representações c e (b) colocar uma rede neural para fazer treinamento *end-to-end* na rede como um todo. Mas agora, usaremos como pesos iniciais para o Transformer e a rede convolutiva o modelo treinado na etapa anterior, já treinada para classificação com os dados dos voluntários com idade menor que 65 anos.

De modo similar à etapa anterior, a expectativa é que o treinamento *end-to-end* gere melhores resultados, mas como a quantidade de dados neste caso é menor, talvez a vantagem não seja tão clara. Uma possibilidade que iremos avaliar é uma aumentação dos dados de treino desse novos grupos, por exemplo, eliminando partes do sinal, de modo similar ao feito para o aprendizado supervisionado na primeira etapa.

Do mesmo modo que na etapa anterior, avaliaremos o uso de um ou dois eletrodos, com as duas abordagens para combiná-los, que é na rede convolutiva ou no Transformer.

3.4 Cronograma:

O projeto tem duração prevista de 1 ano, conforme cronograma da Tabela 1.

Tabela 1: Cronograma

Atividades	1ª Quad.	2ª Quad.	3ª Quad.
Arquitetura baseada em Transformers	X		
Classificação de estágios do sono	X	X	
Agrupamentos de indivíduos para classificação		X	X

4 Adequação do Projeto ao Aluno e a um projeto de Iniciação Científica

O aluno está finalizando o um projeto de IC, onde está utilizando redes neurais convolutivas para classificação de sinais de imagética motora em Interfaces Cérebro Máquina e decodificação des estágios do sono. Consequentemente, o aluno já possui o conhecimento de como criar e operar redes neurais convolutivas e a trabalhar em múltiplos conjuntos de dados de EEG, já tendo experiência com dados de sono. Assim, apesar do projeto ser ambicioso, o aluno já poderá inicia o projeto trabalhando diretamente com o aprendizado das arquiteturas Transformers e Wav2Vec. Deste modo, o cronograma é compatível com a formação e nível de experiência do aluno.

4.1 Inserção do Projeto

Este projeto de IC será desenvolvido em conjunto com o projeto de um aluno de Doutorado do Programa de Pós-Graduação em Ciência da Computação, e que está fazendo um estágio no King's College, e com dois alunos de Iniciação Científica do Curso de Ciên-

cias Moleculares da USP, um deles já com bolsa FAPESP. Além disso, o docente possui colaborações com outros docentes da UFABC e outras universidades, cujos nomes não foram inclusos aqui para manter o anonimato no projeto. Estes docentes orientam outros alunos de Mestrado e Doutorado também envolvidos na área de decodificação de sinais de EEG.

Referências

[1] A. Baevski, H. Zhou, A. Mohamed, and M. Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 2020-Decem(Figure 1):1–19, 2020.

[2] H. Banville, G. Moffat, I. Albuquerque, D. A. Engemann, A. Hyvarinen, and A. Gramfort. Self-Supervised Representation Learning from Electroencephalography Signals. *IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, 2019-Octob, 2019.

[3] P. Bashivan, I. Rish, M. Yeasin, and N. Codella. Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks. *ICLR*, pages 1–15, 2016.

[4] T. B. Brown, J. Kaplan, N. Ryder, T. Henighan, M. Chen, A. Herbert-voss, D. M. Ziegler, G. Krueger, A. Askell, C. Hesse, and S. Mccandlish. Language Models are Few-Shot Learners. *Arxiv*, 2020.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] E. Eldele, Z. Chen, C. Liu, M. Wu, C. K. Kwoh, X. Li, and C. Guan. An Attention-Based Deep Learning Approach for Sleep Stage Classification with Single-Channel EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29:809–818, 2021.

[7] V. Jayaram, M. Alamgir, Y. Altun, B. Scholkopf, and M. Grosse-Wentrup. Transfer Learning in Brain-Computer Interfaces. *IEEE Computational Intelligence Magazine*, 11(1):20–31, feb 2016.

[8] B. Kemp, A. Zwinderman, B. Tuk, H. Kamphuisen, and J. Oberye. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering*, 47(9):1185–1194, 2000.

[9] D. Kostas, S. Aroca-Ouellette, and F. Rudzicz. BENDR: Using Transformers and a Contrastive Self-Supervised Learning Task to Learn From Massive Amounts of EEG Data. *Frontiers in Human Neuroscience*, 15(June):1–15, 2021.

[10] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, 15(5):056013, oct 2018.

[11] X. Liu, F. Zhang, Z. Hou, Z. Wang, L. Mian, J. Zhang, and J. Tang. Self-supervised learning: Generative or contrastive, 2021.

[12] M. N. Mohsenvand, M. R. Izadi, and P. Maes. Contrastive representation learning for electroencephalogram classification. In E. Alsentzer, M. B. A. McDermott, F. Falck, S. K. Sarkar, S. Roy, and S. L. Hyland, editors, *Proceedings of the Machine Learning for Health NeurIPS Workshop*, volume 136 of *Proceedings of Machine Learning Research*, pages 238–253. PMLR, 11 Dec 2020.

[13] O. Ozdenizci, Y. Wang, T. Koike-Akino, and D. Erdogmus. Transfer Learning in Brain-Computer Interfaces with Adversarial Variational Autoencoders. *International IEEE/EMBS Conference on Neural Engineering, NER*, 2019-March:207–210, 2019.

[14] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, 38(11):5391–5420, 2017.

[15] Y. Song, X. Jia, L. Yang, and L. Xie. Transformer-based Spatial-Temporal Feature Learning for EEG Decoding. *Arxiv*, pages 1–10, 2021.

[16] C. Tan, F. Sun, T. Kong, B. Fang, and W. Zhang. Attention-based Transfer Learning for Brain-computer Interface. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1154–1158. IEEE, may 2019.

[17] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A survey on deep transfer learning. *Lecture Notes in Computer Science*, 11141 LNCS:270–279, 2018.

[18] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding, 2019.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. In *Neural Information Processing Systems (NIPS)*, jun 2017.

[20] P. Wang, J. Lu, B. Zhang, and Z. Tang. A review on transfer learning for brain-computer interface classification. *2015 5th International Conference on Information Science and Technology, ICIST 2015*, pages 315–322, 2015.

[21] X. Wei, P. Ortega, and A. A. Faisal. Inter-subject deep transfer learning for motor imagery eeg decoding. In *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 21–24, 2021.

[22] D. Wu, X. Jiang, R. Peng, W. Kong, J. Huang, and Z. Zeng. Transfer learning for motor imagery based brain-computer interfaces: A complete pipeline, 2021.