



## Interface Cérebro-Computador Baseada em P300 Resumo - Lucas Costa e Lima

- **Descrição de uma Interface P300**

Uma Interface Cérebro-Computador (*Brain-Computer Interface - BCI*) é um sistema para comunicação direta entre o cérebro e um computador [1]. A comunicação se estabelece através de comandos dados por padrões cerebrais capturados e processados através de algum paradigma específico e utilizados para a comunicação. O paradigma escolhido durante a Iniciação Científica em questão utiliza sinais cerebrais extraídos através de Eletroencefalografia (EEG), especificamente no Lobo Occipital, com amostragem temporal do sinal.

A interface escolhida para reprodução foi feita em 1988 por Farwell e Donchin [2], consistente de uma matriz 6x6 com letras a serem escolhidas uma a uma, após seguidos períodos de excitação aleatória de linhas e colunas da matriz (tendo em vista o ganho de tempo na seleção das letras).

A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	1	2	3	4
5	6	7	8	9	-

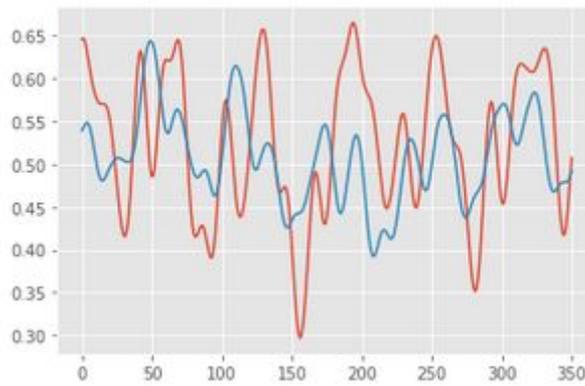
**Figura 1:** Representação gráfica da matriz a ser vista pelo sujeito.

O sinal a ser analisado na interface escolhida é o potencial P300. Tal potencial é evocado no cérebro cerca de 300ms após o seu estímulo. Este ocorre, na situação em questão, cada vez que a letra desejada é iluminada em sua linha ou coluna, então o estímulo visual gera o tal potencial, podendo ser classificado e utilizado para escrever.

Após a captura dos sinais para treino, utiliza-se um algoritmo de pré-processamento e subsequente classificação de sinais onde há ou não a presença de P300. Após um modelo de algoritmo treinado para o reconhecimento e classificação, a versão “*online*” pode ser utilizada, onde não mais se conhece a saída esperada para efetivar a comunicação.

- **Pré-processamento**

Os dados utilizados viram de uma base de dados aberta encontrada em [P300 Dataset](#). Filtrados utilizando um *notch* em 50Hz e um filtro passa-banda 0.1-30Hz. Entretanto, o sinal de P300 é muito menor, em amplitude, que a maior parte dos sinais basais cerebrais de maior frequência implementou-se uma média acerca dos estímulos de cada linha/coluna visando aumentar o sinal P300 e diminuir o ruído basal do cérebro. Além disso, um filtro *butterworth* passa-baixas foi implementado também na intenção de diminuir os ruídos de frequências que não interessam ao sinal em questão, como ondas gama.



**Figura 2:** Em vermelho um sinal positivo pós-processamento, em azul um sinal negativo pós-processamento.

- **Escolha do Algoritmo Inteligente para análise dos dados**

A primeira abordagem foi feita sem um algoritmo inteligente propriamente dito. Utilizou-se um filtro casado na expectativa de perceber a correlação entre os dados. Para tal, estabeleceu-se um sinal “ideal”, feito das médias de todos os sinais positivos de P300 disponíveis entre os 7 sujeitos e, em seguida, o filtro era aplicado entre o sinal ideal e o sinal a ser avaliado. Este processo sofreu algumas alterações de abordagem, como um filtro para cada indivíduo ao invés de utilizar os dados de todos, uma vez que o tempo de resposta é particularmente diferente para cada um. Entretanto não houveram resultados satisfatórios na classificação dos sinais, assim levando a abordagem com algoritmos inteligentes. O aluno era previamente capacitado na utilização de redes neurais multicamada para classificação (MLP) por ter feito um curso com o orientador na pós-graduação como aluno especial, assim, a abordagem escolhida foi utilizar um MLP construído no *Keras*[3] para a classificação dos dados.

```
Model: "sequential_7"
```

Layer (type)	Output Shape	Param #
conv1d_11 (Conv1D)	(None, 351, 4)	24
global_average_pooling1d_7 (GlobalAveragePooling1D)	(None, 4)	0
dense_10 (Dense)	(None, 2)	10
activation_7 (Activation)	(None, 2)	0

Total params: 34  
Trainable params: 34  
Non-trainable params: 0

**Figura 3:** Arquitetura minimalista da MLP.

A arquitetura escolhida variou no número de camadas lineares e convolucionais na tentativa de otimizar a acurácia do modelo, mas observou-se uma falha de evadir um mínimo local da função de custo. Por causa disso um modelo minimalista com apenas uma camada convolucional com máscara de tamanho 5 e uma camada linear para a classificação foi adotado na tentativa de não recair no caso do mínimo local em questão, que será abordado no próximo tópico.

Foi decidido utilizar ao menos uma camada convolucional uma vez que o sinal possui características, ainda que pequenas, no seu espectro que diferenciam o P300 de um sinal não-P300. A

partir disso, também foi decidido utilizar os dados de forma categórica para melhorar o desempenho de classificação. O otimizador da rede é o Adaptive Momentum (Adam)[4]. O Adam tem se mostrado o otimizador mais robusto para redes neurais e sua escolha é praticamente inquestionável na maioria das soluções, exceto algumas muito específicas.

- **Abordagens adotadas para a resolução dos problemas de classificação**

Muito embora o poder de generalização dos algoritmos inteligentes seja conhecido e exaltado pela maioria, a estrutura existente nos dados ainda é prevalecente sobre esta capacidade, uma vez que este generaliza sobre àquele. Existiu um problema de classificação durante o trabalho que tentou, de diversas formas, ser resolvido, sem sucesso, mantendo a integralidade dos dados, fazendo diferentes formas de médias e filtragens. Após diversas tentativas de convergência da rede para apenas um indivíduo fracassarem sem a eliminação de dados, isto é, abortada a hipótese de poucos dados para a convergência da rede, a abordagem foi diferente.

Já é conhecido o efeito de *datasets* desbalanceados em problemas de classificação, que era o caso, numa proporção de **5:1** dos dados negativos em comparação aos positivos. Uma estratégia possível seria fazer K-Folding com as 5 partes de sinais negativos e a parte positiva [6], entretanto, a hipótese foi descartada por não querer permitir um *overfitting* da rede ao indivíduo. Então, arbitrariamente eliminou-se 80% dos dados negativos de P300 tentando retirar o mínimo local da função de custo, o que foi completamente positivo e os resultados convergiram de imediato.

- **Comparação de estratégias de *Deep Learning***

Após a resolução do pré-processamento, as estratégias para a classificação de P300 mais usuais eram comitês de máquinas de vetores suporte (SVM) e redução de dimensionalidade (PCA ou ICA) aplicados aos eletrodos, na tentativa de selecionar os melhores canais para a classificação. Na tentativa de abordar o problema de uma outra perspectiva foram considerados:

- i. Dependência espacial dos dados:

- Cada eletrodo se localiza em uma parte diferente do crânio, permitindo inferir informação espacial sobre o sinal de P300, suscitando fortemente a hipótese de alimentar a rede como uma rede de classificação de imagens;

- ii. Dependência temporal dos dados:

- Em BCI é comum analisar o espectro de frequências do sinal, entretanto, parte relevante do sinal precisa estar no domínio do tempo;

- iii. *Transfer Learning*

- Um conceito aplicado para um ajuste fino de redes deve ser fortemente considerado aqui, uma vez que podemos deixar uma arquitetura pré-treinada em dados de vários sujeitos por um tempo maior e, para o sujeito final, e.g. um consumidor, o tempo de treino seria bem menor e a rede convergiria bem para mais pessoas.

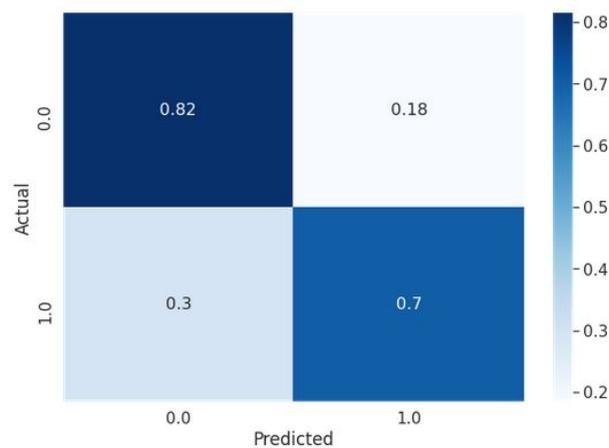
- 1. Considerações sobre *Transfer Learning*

- Primeiro achou-se por ser de grande valia a rede ser treinada de

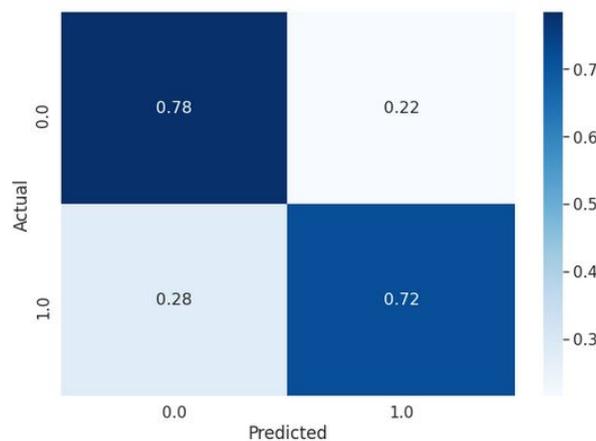
indivíduo em indivíduo, para analisar como a função de custo convergia, mas percebeu-se por bem avaliá-la em vários indivíduos primeiro e aplicar o ajuste fino apenas no sujeito final;

- **Resultados**

A seguir estão as matrizes de confusão de abordagens consideradas relevantes para a análise:



**Figura 6:** Matriz de confusão com *Transfer Learning* para sinais 2D



**Figura 7:** Melhor matriz de confusão apenas com ajuste fino para último indivíduo

Além dos resultados supracitados, houveram performances muito desbalanceadas quando:

- Aumentava-se o número de épocas de treino de forma desproporcional (*Overfitting*)
- Utilizava-se funções de ativação lineares nas camadas completamente conectadas no fim
- Inseriam-se camadas de normalização (*Batch Normalization*) ou redução de dados (*Dropout*)

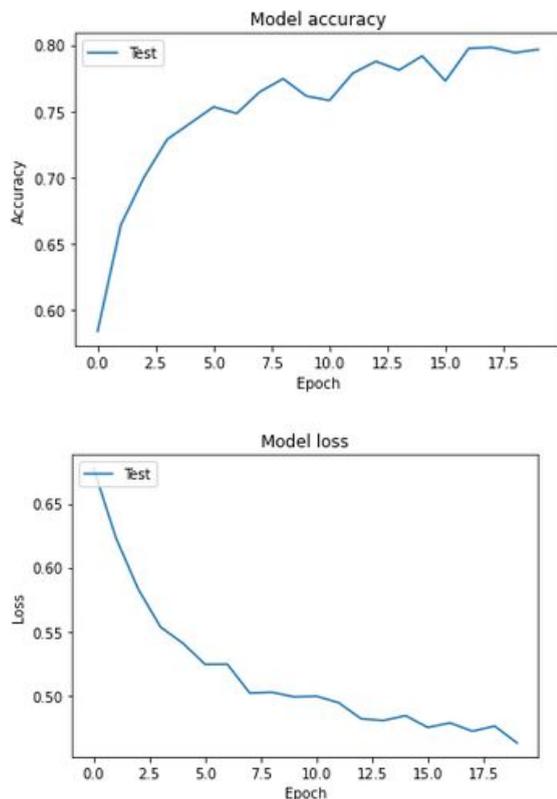
A explicação para a queda de desempenho do *Overfitting*, principalmente em uma queda de desempenho nos dados que continham P300, provém da conformação do modelo às características

basais dos sinais, comuns em ambos, que provavelmente reduziram a sensibilidade do modelo ao sinal de P300. A queda de desempenho na utilização de camadas lineares (*ReLU*) na ativação no final da rede, após camadas convolucionais, escala as frequências de forma monotônica, assim, reduz o impacto das pequenas frequências, responsáveis pelo sinal de P300.

Uma última discussão válida é a aplicação do *Transfer Learning* na fase de treinos ou não. Como descrito em seções anteriores, duas abordagens foram feitas:

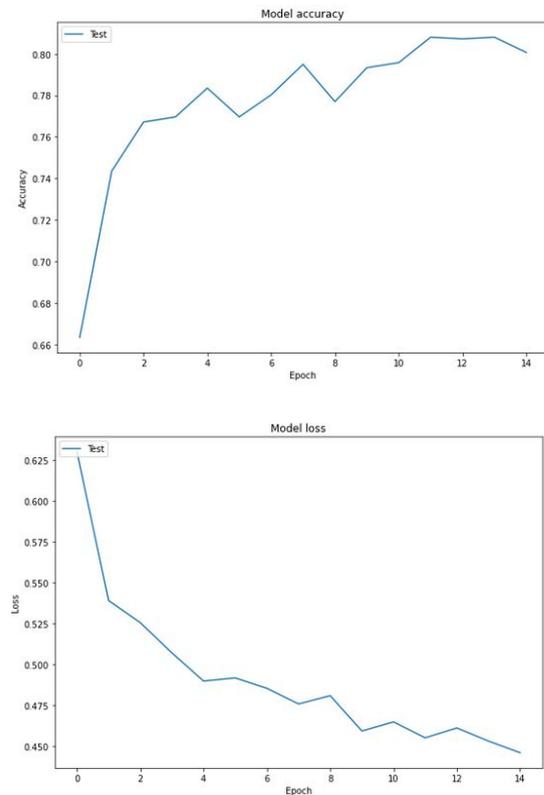
- 1 - *Transfer learning* de indivíduo para indivíduo durante o treino e de novo no ajuste fino;
- 2 - *Transfer learning* apenas no ajuste fino, misturando os dados dos indivíduos na fase de treino;

A seguir veremos que a diferença na velocidade de convergência no ajuste fino é mínima, respectivamente as situações 1 e 2:



**Figuras 8 e 9:** Acurácia e Custo para o Caso

1



**Figuras 10 e 11:** Acurácia e custo para o Caso

2

Sendo possível perceber que não há diferença prática de convergência no ajuste fino entre os casos.

## Referências

- 1 - H. Serby , E. Yom-Tov - “An Improved P300-Based Brain-Computer Interface”
- 2 - L. A. Farwell and E. Donchin, “Talking off the top of your head: A mental prosthesis utilizing event-related brain potentials,” *Electroencephalogr. Clin. Neurophysiol.*, vol. 70, pp. 510–523, 1988.
- 3 - [Keras: the Python deep learning API](#)
- 4 - <https://arxiv.org/pdf/1412.6980.pdf>