

# Estudo de Técnicas de Processamento de Sinais e de Aprendizado de Máquina no Desenvolvimento de Sistemas de Interface Cérebro-Computador

**Palavras-Chave:** Interface Cérebro-Computador; Análise de Correlação Canônica; Aprendizado de Máquina.

**Autores:**

**Daniel Neri de Jesus, FEEC – UNICAMP**

**Prof. Denis Fantinato (orientador), FEEC - UNICAMP**

---

## INTRODUÇÃO:

As chamadas Interfaces Cérebro-Máquina (BCI, do inglês *Brain-Computer Interface*) são uma categoria de dispositivos para a troca de informação direta entre o cérebro humano e sistemas de informática [2]. Devido às importantes aplicações, como em tecnologias assistivas, equipamentos para reabilitação e dispositivos para entretenimento, estudos sobre BCIs estão sendo foco de grande atenção pela comunidade científica e pelo setor produtivo. Os principais processos de um sistema BCI são: (i) a *aquisição dos sinais elétricos* da atividade cerebral, que, no caso de sistemas baseados em EEG, são extraídos através de eletrodos posicionados no escalpo do usuário, em regiões de interesse; (ii) o *pré-processamento dos dados* obtidos na etapa de aquisição, a fim de remover possíveis interferências ao sistema; (iii) a *extração de características*, cujo objetivo principal é mapear os sinais para outro espaço no qual é possível distinguir a informação de interesse com maior eficiência; (iv) a *seleção de características*, que busca a separação das informações mais relevantes para a classificação; e finalmente, a última etapa do sistema BCI é (v) a *classificação dos dados*, para que seja possível definir a que os sinais elétricos cerebrais estão associados. O foco do projeto ocorreu na etapa final, a de classificação dos dados, onde os estudos e aplicações de métodos de aprendizado de máquina foram realizados. Além disso, como diversos dispositivos podem ser considerados sistemas BCI, o enfoque foi em interfaces Cérebro-Máquina baseadas em eletroencefalogramas de superfície (EEG). Esta abordagem, além de bastante difundida, permite a aquisição de dados de maneira não invasiva e com equipamento relativamente simples.

Dessa maneira destacam-se dois paradigmas de BCI: aqueles baseados em potenciais visuais evocados em regime permanente (SSVEP, do inglês *Steady-State Visually Evoked Potential*) e aqueles baseados em imagética motora [2]. O primeiro, SSVEP, está associado com o sincronismo da atividade elétrica do córtex visual mediante estimulação visual externa em certa frequência. Comumente, essas interfaces se baseiam em diversas fontes luminosas que piscam em diferentes frequências, em que cada determinada frequência é associada a um certo comando. No segundo caso, imagética motora, o indivíduo deve imaginar a realização de certos movimentos pré-definidos (e.g., mover a mão direita, mover os pés ou a língua), o que gera certos padrões específicos de atividade neuronal, principalmente no córtex motor. A partir destes padrões de sinais específicos gerados, um sistema BCI deve realizar a classificação destes sinais, e definir corretamente o movimento imaginado. Ambos os casos, apesar de serem a base de tecnologias bastante promissoras, ainda encontram grandes dificuldades para o seu funcionamento pleno e ideal. Os principais desafios destes paradigmas são associados à extração dos sinais cerebrais, bem como ao seu processamento. Por se tratarem de técnicas não invasivas, o ruído (qualquer sinal de interferência que cause prejuízo no processamento) é bastante presente, tanto por atividades internas do cérebro que não servem à categorização específica destes paradigmas, quanto por sinais externos, já que a amplitude dos sinais elétricos cerebrais é extremamente baixa (de 5 - 20  $\mu\text{V}$ ) [5]. Desta forma, nota-se que um importante desafio para que sistemas BCIs se tornem utilizados em ampla escala

se refere ao processamento e classificação corretos dos sinais das interfaces, em tempo hábil. Neste sentido, o presente projeto visa realizar um estudo aprofundado acerca dos métodos de pré-processamento e de classificação, usando redes neurais artificiais para aprimorar os sistemas BCI. Uma análise comparativa entre os principais métodos foi realizada, buscando identificar aquelas capazes de aumentar a taxa de acurácia do classificador.

## METODOLOGIA:

O objetivo geral do projeto foi o aprendizado e comparação de modelos introdutórios à classificação de dados, como o Método dos Mínimos Quadrados (MMQ), Perceptron e a rede MLP (do inglês, *Multilayer Perceptron*) [1][4]. Através da linguagem de programação *Python*, foram realizadas as implementações e testes necessários para o desenvolvimento do projeto. Inicialmente, foram aplicadas as redes mais simples em dados sintéticos e, posteriormente, foram utilizados dados reais de EEG, presentes em repositórios públicos. Os principais métodos estudados e a implementação de classificadores estão descritos abaixo:

- *MMQ*: É uma conhecida técnica matemática, amplamente utilizada no meio científico para tratamento de dados. Seu uso original é para as regressões lineares, onde busca-se, através de amostras reais de duas ou mais variáveis, encontrar relações angulares e lineares entre as mesmas [1]. Apesar da sua natureza não relacionada à classificação de dados, o MMQ mostrou-se uma importante ferramenta introdutória ao aprendizado de máquina. Para o treinamento da rede é necessária uma função de erro, que calcula o quadrado da diferença entre o valor da classe obtido pela função  $f(x_i)$ , onde  $x_i$  é a  $i$ -ésima amostra do conjunto de dados  $X$ , e o valor de classe esperado para esta amostra  $y_i$ , isto é,

$$J(w) = \frac{1}{2} \sum_{N=1}^i (f(x_i) - y_i)^2, \quad (1)$$

onde  $f(x_i)$  é definido por  $f(x) = w^T \cdot X$ , sendo  $w$  a matriz de pesos a ser descoberta. Derivando a equação do erro quadrático, e igualando a 0 para descobrir pontos de mínimos, podemos encontrar para  $W = (X^T X)^{-1} \cdot X^T Y$ , satisfazendo um ponto de mínimo da equação do erro. Vale-se ressaltar, que para o treinamento do modelo como demonstrado, além de um conjunto de amostras  $x_i$ , é necessário um conjunto de resultados esperados previamente descobertos  $y_i$ , só então o modelo está preparado para classificar dados sem resultados descobertos previamente. Tal classificação, então, é decidida por uma função de ativação, que gerará uma saída positiva ou negativa. Em caso negativo, o dado  $x_i$  será classificado como classe 0, e em caso de resultado positivo, retornará o dado  $x_i$  como sendo classe 1, onde o significado de classe 0 e classe 1 pode-se ser previamente definido ao treinar o modelo com os resultados previamente descobertos.

-*Perceptron*: Diferentemente do MMQ, o perceptron é um modelo desenvolvido especificamente para a classificação de dados [1][4]. Apesar de ser uma rede neural simples, há um ganho de complexidade em relação ao método de MMQ, o que em alguns casos pode acarretar em classificações mais precisas em comparação ao método anterior. Para o entendimento do funcionamento do perceptron simples, é necessário se basear na seguinte equação:  $\Delta w = T \cdot (y_i - F(x_i))$ , sendo  $\Delta w$  a taxa de atualização dos pesos,  $T$  a taxa de aprendizado e  $F(x_i)$  a função de ativação. A equação, então, calcula o erro entre a classe esperada  $y_i$  e a classe estimada pela função de ativação  $F(x_i)$ . No início do algoritmo, é escolhida uma taxa de aprendizagem arbitrária (usualmente menor que 1,0) para multiplicação do erro calculado, gerando finalmente a taxa de atualização dos pesos. Para a estimativa inicial dos pesos, há a necessidade da utilização de valores arbitrários para o cálculo da função de ativação, com isto, inicia-se os pesos como nulos, ou como números aleatórios. No geral, o funcionamento do perceptron é (1) Estimar o erro entre a função de ativação e a classe real, (2) Atualizar os pesos com base na taxa de atualização gerada no passo anterior, (3) recomeçar com os novos pesos. Diferentemente do MMQ, percebe-se que o perceptron necessita de diversas iterações, sendo que, por possuir um ciclo fechado, ele

necessita previamente de um número  $n$  de ciclos estipulados manualmente, já que o sistema pode continuar infinitamente a atualizar os pesos caso não haja separação ideal onde  $y_i - F(x_i) = 0$  para todos os dados. A partir disso, entende-se a motivação da taxa de aprendizagem ser menor que 1,0, pois, caso a taxa de atualização a cada interação seja muito alta, a rede pode divergir [4]. Um comparativo de separação de dados artificiais foi feito entre o MMQ e o perceptron, onde justifica-se a maior complexidade e o maior recurso computacional necessário ao perceptron.

*-MLP, Perceptrons de Múltiplas Camadas:* A rede MLP, nada mais é do que um conjunto de perceptrons organizados de maneira a formar uma espécie de rede. Uma rede MLP pode possuir diversas camadas intermediárias, ou seja, perceptrons que não retornam propriamente o valor da classificação dos dados, mas que retorna atributos importantes para a classificação final e, com isso, cria-se uma “trama” que se diferencia do caráter linear de perceptrons simples e do MMQ. Essa diferença também é vista nas classificações, já que os modelos anteriores realizam apenas a separação linear dos dados e a rede MLP é capaz de resolver classificações não lineares. Apesar do uso da rede MLP ser extremamente poderoso, o recurso computacional necessário é consideravelmente maior, a depender da arquitetura da rede (quantidade de camadas intermediárias, e de perceptrons por camada), além do fato de que a não linearidade da rede e das camadas intermediárias tornarem difíceis a sua análise teórica e a visualização do seu processo de aprendizagem [4]. Além disso, o formato da arquitetura da rede depende do problema a ser resolvido, e a sua definição não é tão simples, tornando o seu uso correto relativamente mais difícil.

## RESULTADOS E DISCUSSÃO:

*- Comparativo MMQ e perceptron:* Para este comparativo foram usados 3 conjuntos de dados, linearmente separáveis. Utilizando cada conjunto de dados como uma coordenada no espaço, foi possível plotar as amostras conforme a Fig. 1.



Figura 1: Comparativo MMQ x Perceptron LEGENDA: Verde: Classe 0, Vermelho: Classe 1, Azul: Plano

Após definido o conjunto de dados, foram treinados dois modelos para a sua classificação, o MMQ e o perceptron. Para o treinamento foi utilizado todo o conjunto de dados. Na Fig. 1, nota-se que o MMQ possui um desempenho bastante satisfatório e similar ao perceptron. Porém, o plano feito pelo MMQ erra na classificação de um dado verde, colocando o mesmo abaixo do plano. Diferentemente, o perceptron realiza a separação das duas classes sem erros nos dados. Também foi feito um gráfico de cálculo de erros no perceptron [9], o qual contabiliza o número de vezes que os pesos necessitam de atualização em seu valor, através da taxa  $\Delta w$ . Conforme a Fig. 2 à direita demonstra, o problema de classificação foi resolvido em apenas 1 interação, atualizando duas vezes os valores dos pesos, diferentemente de casos mais complexos que serão abordados posteriormente.

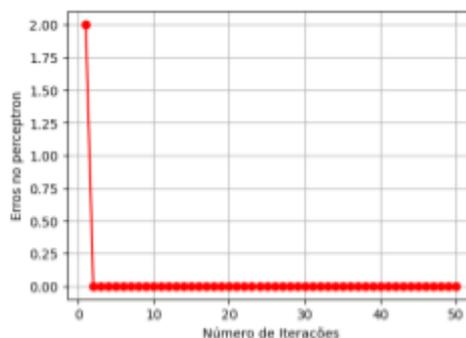
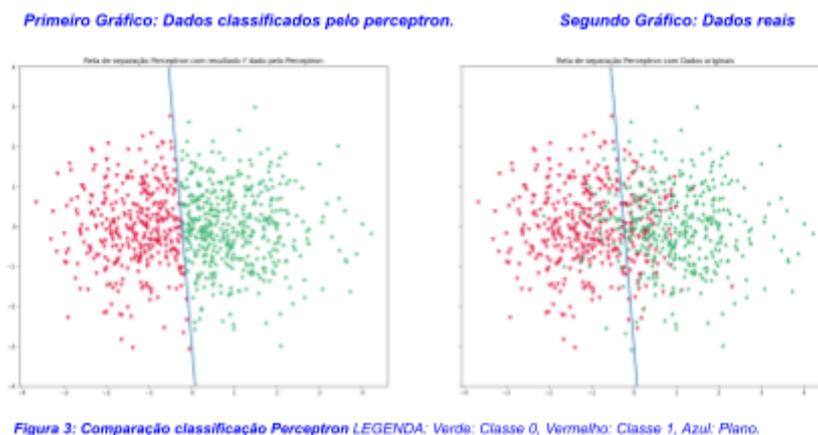


Figura 2: atualização dos pesos por interação

- *Perceptron simples em dados não perfeitamente separáveis*: Também foram feitos testes do perceptron com dados simulativos de dados reais, isto é, dados não classificáveis de maneira exata. O desempenho do perceptron simples pode ser acompanhado na Fig. 3 abaixo:



Como comentado anteriormente, o perceptron pode iterar infinitamente caso não haja uma separação ideal, por tanto, foi utilizada a função custo  $J = \sum_{N=1}^i (f(x_i) - y_i)^2$  que calcula o erro quadrático da estimativa do perceptron  $f(x_i)$  com a saída real  $y_i$ , a fim de se obter um número ideal de interações que minimize este erro.

Pelo gráfico abaixo, nota-se que, neste caso, o número de interações desejável para parada é 170, pois o erro quadrático do perceptron é minimizado. Nota-se também, que o número de erros no perceptron varia constantemente, diferentemente da Fig. 2.

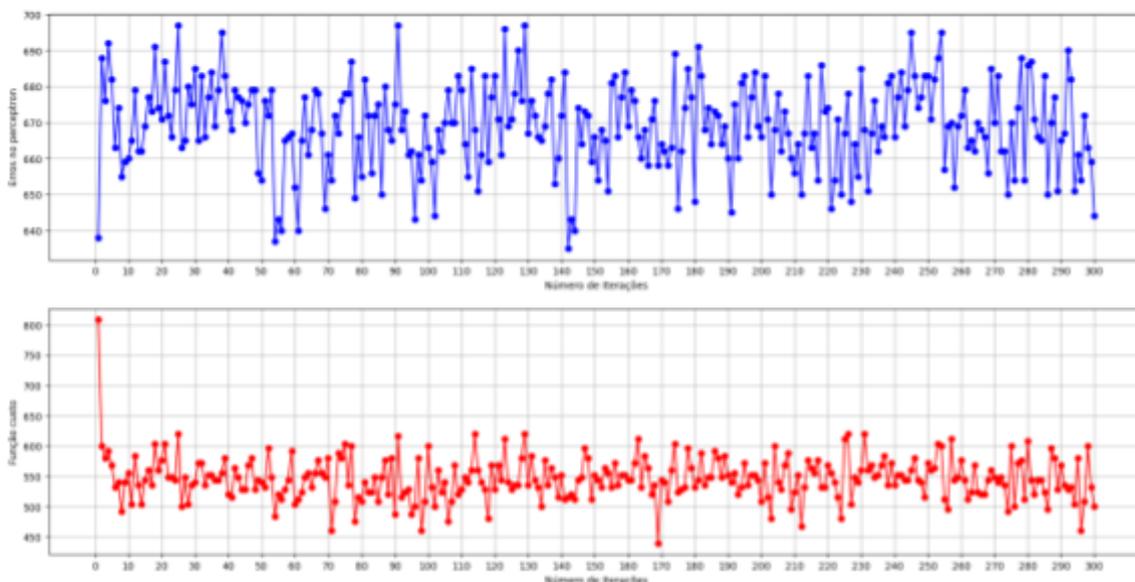


Figura 4: Erros no perceptron (superior), Função Custo (Inferior)

- *MLP em dados de EEG*: Os dados de EEG utilizados [10] possuíam uma variedade muito grande, impossibilitando a visualização gráfica da separação. Para o treinamento das redes foram usados uma proporção de 80% dos dados para treino, enquanto os 20% restantes foram para teste. Diferentes topologias foram comparadas, sendo exibido na Fig. 5 a acurácia de cada arquitetura aplicada, onde cada conjunto de valores entre parênteses representa a quantidade de neurônios por camada intermediária. Nota-se que o perceptron simples e a rede MLP (100) acertaram apenas 25% dos dados. Considerando que os dados possuíam 4 classes de imagética motora (braço direito, braço esquerdo, perna e língua), estes dois modelos não retornaram resultados melhores do que um classificador aleatório. No entanto, a rede de arquitetura (100,100,100) teve um

grande salto de desempenho em comparação ao Perceptron Simples e a rede MLP (100), alcançando 54,3% de acurácia. Também foi utilizada uma técnica de amostragem *K-fold cross validation* [1], a qual selecionou a melhor topologia de rede como sendo a arquitetura (10, 30). Este método separa os dados em 5 partições e, então, treina uma rede com 4 conjuntos, usando a partição restante como validação. As combinações possíveis de conjunto geram 5 configurações distintas, resultando 5 redes treinadas. Ao final, uma acurácia média para a arquitetura da rede é obtida, sendo uma maneira para evitar enviesamento de dados de treino. Com isto, então, foram realizadas diversas simulações, e a arquitetura (10,30) foi a que obteve maior acurácia média (50,1%). Apesar disso, nota-se que ela se saiu ligeiramente pior que a rede MLP (100,100,100) no teste comparativo. Este fato explica-se pois, para o primeiro comparativo, foi utilizada a inicialização de pesos fixa (*seed* fixo) para redes. O processo de treinamento da rede (100,100,100) se saiu particularmente melhor com os pesos utilizados nesse primeiro teste. Por este motivo, foram feitos 100 testes com inicializações de pesos diferentes (porém com mesmo *seed* entre as duas redes), e então foi armazenado o desempenho de cada rede. A rede treinada pelo método k-fold se saiu melhor em 70% dos casos, demonstrando que o método, apesar do primeiro teste, resulta em uma arquitetura mais bem definida de maneira geral ao problema em questão, portanto, é um método válido.

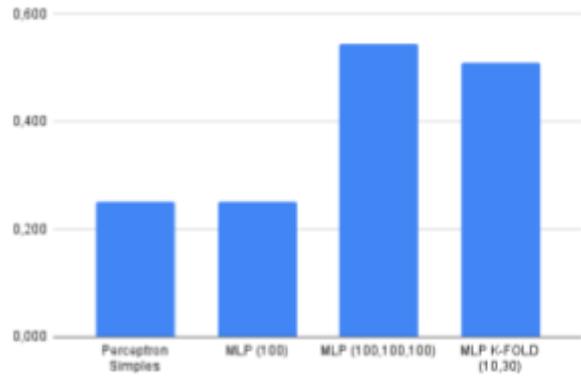


Figura 5: Comparativo Modelos x Acurácia

## CONCLUSÕES:

Foram estudados um conjunto de métodos para a classificação de dados artificiais e reais. Nos dados sintéticos, observou-se uma vantagem no uso do perceptron, mesmo em um cenário simples. No entanto, o método dos mínimos quadrados também apresentou resultados satisfatórios, com desempenho próximo ao do perceptron. No caso dos dados simulados, o perceptron demonstrou dificuldades em problemas não lineares, uma limitação superada pela rede MLP, que possui maior complexidade e capacidade classificatória. Apesar da rede MLP utilizada se aproximar do desempenho em dados de BCI na literatura, os resultados indicam a extrema dificuldade na classificação de dados de interfaces cérebro-computador, um problema que ainda não está completamente resolvido, mesmo com modelos mais complexos.

Algumas limitações da pesquisa incluíram o uso exclusivo de características no domínio do tempo, o que impactou significativamente o desempenho da rede MLP em dados reais, e restrições computacionais que impediram a avaliação de uma maior variedade de arquiteturas de redes MLP. Apesar dessas limitações, o desempenho dos modelos foi satisfatório para a proposta do projeto. Espera-se que modelos mais complexos sejam utilizados em trabalhos futuros.

## BIBLIOGRAFIA

- [1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [2] P. O. de Paula, T. B. da Silva Costa, R. Attux, and D. G. Fantinato. *Classification of image encoded SSVEP-based EEG signals using Convolutional Neural Networks. Expert Systems with Applications*, 214:119096, 2023.
- [3] I. Guyon and A. Elisseeff. *An Introduction to Variable and Feature Selection. Journal of machine learning research*, 3(Mar):1157-1182, 2003.
- [4] S. Haykin. *Neural Networks and Learning Machines*. Pearson Education, 2010.
- [5] A. Kübler and K.-R. Müller. *An Introduction to Brain-Computer Interfacing. Toward brain- computer interfacing*, pages 1-25, 2007.
- [6] C. S. Nam, A. Nijholt, and F. Lotte. *Brain-Computer Interfaces Handbook: Technological and Theoretical Advances*. CRC Press, 2018.
- [7] D. Regan. *Human Brain Electrophysiology: Evoked Potentials and Evoked Magnetic Fields in Science and Medicine*. 1989.
- [8] Wes McKinney. *Python Para Análise de Dados: Tratamento de Dados com Pandas, NumPy & Jupyter*. novatec 2023.
- [9] *Perceptron com Python, uma introdução*, MEDIUM, nov. 2019, disponível em: <https://medium.com/@urapython.community/perceptron-com-python-uma-introdução-f19aaf9e9b64>
- [10] C. Brunner , R. Leeb , G. R. Muller-Putz , A. Schlogl , and G. Pfurtscheller *BCI Competition 2008 – Graz data set A, 2008*