

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Matheus Felipe Calisto

**Reconhecimento da Doença de Parkinson por  
meio de sinais EMG utilizando Inteligência  
Artificial**

**Uberlândia, Brasil**

**2024**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Matheus Felipe Calisto

**Reconhecimento da Doença de Parkinson por meio de  
sinais EMG utilizando Inteligência Artificial**

Trabalho de conclusão de curso apresentado  
à Faculdade de Computação da Universidade  
Federal de Uberlândia, como parte dos requi-  
sitos exigidos para a obtenção título de Ba-  
charel em Sistemas de Informação.

Orientador: Prof. Dr. Daniel S. Duarte Caetano

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2024

Matheus Felipe Calisto

## **Reconhecimento da Doença de Parkinson por meio de sinais EMG utilizando Inteligência Artificial**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

---

**Prof. Dr. Daniel S. Duarte Caetano**  
Orientador

---

**Profa. Dra. Alessandra A. Paulino**  
Membro da Banca

---

**Profa. Dra. Fernanda M. C. Santos**  
Membro da Banca

Uberlândia, Brasil  
2024

*Dedico a minha família e amigos, por toda a compreensão, motivação e apoio incondicional durante o desenvolvimento deste trabalho.*

# Agradecimentos

Em primeiro lugar, agradeço a Deus, que meu deu a oportunidade, a força e a saúde necessárias para concluir essa trajetória.

Aos meus pais, Carlos e Marceli, por todo apoio e incentivo que tornaram tantos sonhos possíveis.

Ao meu amigo Heitor, por estar ao meu lado em todos os momentos e por me dar forças para não desistir.

Ao meu orientador Daniel e a todos meus professores, que durante toda essa jornada me auxiliaram e compartilharam seus conhecimentos, tornando esse trabalho possível.

# Resumo

Os indivíduos que sofrem com a doença de Parkinson (DP), possuem muitas dificuldades em sua rotina, tais como, problemas na fala, dificuldade na escrita, tremores e desequilíbrios. Isso pode trazer diversas limitações nas atividades diárias como também profissionais. Várias abordagens são investigadas para tentar detectar a doença (sinal eletromiográfico (EMG), ressonância magnética (RM)). Para complementar os métodos tradicionais de diagnóstico, o uso de algoritmos que utilizam inteligência artificial (IA) tem sido empregado para realizar a detecção da DP. A literatura apresenta diversos trabalhos que buscam analisar padrões e fazer o reconhecimento da DP, para isso utilizam diversas formas, cada estudo mostra um nível de eficácia diferente. Diante disso, se torna necessário o uso de métodos que utilizam IA para realizar a detecção da DP, buscando ter o melhor desempenho possível e auxiliar esses pacientes. O objetivo deste estudo é verificar o desempenho da rede neural Learning Vector Quantization (LVQ) para detectar quais indivíduos presentes em uma base de dados pública possuíam a DP, nessa base o sinal EMG já foi processado e tivemos acesso as características já extraídas e através disso foi verificado quais eram os parâmetros considerados ótimos. Com isto, utilizando-se a rede LVQ, os parâmetros considerados ótimos e 80% dos dados para treinamento, alcançou-se 68.80% de eficiência na a detecção de indivíduos que possuíam a DP.

Palavras Chave: Doença de Parkinson, rede neural LVQ, sinais EMG, Inteligência Artificial.

# Lista de ilustrações

|                                                                 |    |
|-----------------------------------------------------------------|----|
| Figura 1 – Exemplo de sinal EMG (CAETANO et al., 2012). . . . . | 15 |
| Figura 2 – Esquema - Rede LVQ. . . . .                          | 18 |
| Figura 3 – Dados do banco. . . . .                              | 26 |

# Lista de tabelas

|          |   |                                                             |    |
|----------|---|-------------------------------------------------------------|----|
| Tabela 1 | – | Resumo Comparativo dos Trabalhos Relacionados . . . . .     | 25 |
| Tabela 2 | – | Resultados de variação de Alpha . . . . .                   | 33 |
| Tabela 3 | – | Resultados de variação de Decaimento Alpha . . . . .        | 34 |
| Tabela 4 | – | Resultados de variação de Tolerância . . . . .              | 34 |
| Tabela 5 | – | Resultados de variação dos Clusters . . . . .               | 34 |
| Tabela 6 | – | Resultados dos clusters em diferentes percentuais . . . . . | 35 |

# Lista de abreviaturas e siglas

|              |                                                                           |
|--------------|---------------------------------------------------------------------------|
| ANN          | Rede Neural Artificial                                                    |
| APQ3         | Quociente de Perturbação de Amplitude 3                                   |
| D2           | Dimensão de Correlação                                                    |
| <i>DBS</i>   | Estimulação Cerebral Profunda                                             |
| DFA          | Análise de Flutuação de Tendência                                         |
| DP           | Doença de Parkinson                                                       |
| <i>FD</i>    | Frequency Domain                                                          |
| HNR          | Relação Harmônico-ruído                                                   |
| IA           | Inteligência Artificial                                                   |
| <i>IEMG</i>  | Integrated EMG                                                            |
| INTERQ RANGE | Intervalo Interquartil do Sinal                                           |
| KNN          | K-vizinhos mais próximos                                                  |
| LDA          | Análise Discriminante Linear                                              |
| LOG          | Detector de Log                                                           |
| LVQ          | Learning Vector Quantization                                              |
| MAR          | Modelo Autoregressivo                                                     |
| <i>MAV</i>   | Mean Absolute Value                                                       |
| MAVFD        | Média dos Valores Absolutos das Primeiras Diferenças do Sinal             |
| MAVFDN       | Média dos Valores Absolutos das Primeiras Diferenças do Sinal Normalizado |
| MAVS         | Inclinação Média do Valor Absoluto                                        |
| MAVSD        | Média dos Valores Absolutos das Segundas Diferença                        |
| MAVSDN       | Média dos Valores Absolutos das Segundas Diferenças do Sinal Normalizado  |

|              |                                                               |
|--------------|---------------------------------------------------------------|
| MDVP         | Programa de Voz Multidimensional                              |
| <i>MDF</i>   | Most Discriminant Features                                    |
| <i>MNF</i>   | Minimum Noise Fraction                                        |
| <i>MNP</i>   | Mean Power                                                    |
| NHR          | Relação Ruído-harmônico                                       |
| <i>PEAK</i>  | Máximo Global                                                 |
| PPE          | Entropia do período de pitch                                  |
| PPQ          | Quociente de Perturbação de Período                           |
| RANGE        | Média dos Valores Absolutos das Primeiras Diferenças do Sinal |
| RBF          | Função de Base Radial                                         |
| RM           | Ressonância Magnética                                         |
| RPDE         | Entropia de Densidade de Período Recorrente                   |
| <i>RMS</i>   | Root Mean Square                                              |
| SVM          | Máquina de Vetores de Suporte                                 |
| STD          | Desvio Padrão                                                 |
| <i>TDF</i>   | Time Domain Features                                          |
| <i>T-SNE</i> | T-Distributed Stochastic Neighbor Embedding                   |
| TTP          | Potência Total                                                |
| <i>VAR</i>   | Variance                                                      |
| <i>WL</i>    | Waveform Length                                               |
| <i>ZC</i>    | Zero Crossing                                                 |

# Sumário

|            |                                                                                                                                                    |           |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>1</b>   | <b>INTRODUÇÃO</b>                                                                                                                                  | <b>12</b> |
| <b>1.1</b> | <b>Objetivos</b>                                                                                                                                   | <b>13</b> |
| 1.1.1      | Objetivo Geral                                                                                                                                     | 13        |
| 1.1.2      | Objetivos específicos                                                                                                                              | 13        |
| <b>1.2</b> | <b>Organização da monografia</b>                                                                                                                   | <b>13</b> |
| <b>2</b>   | <b>FUNDAMENTAÇÃO TEÓRICA</b>                                                                                                                       | <b>14</b> |
| <b>2.1</b> | <b>Doença de Parkinson e Sinais EMG</b>                                                                                                            | <b>14</b> |
| <b>2.2</b> | <b>Fundamentos de Sinais Biomédicos</b>                                                                                                            | <b>14</b> |
| 2.2.1      | O sinal EMG                                                                                                                                        | 14        |
| 2.2.2      | Processamento de sinal EMG                                                                                                                         | 15        |
| <b>2.3</b> | <b>Inteligência Artificial</b>                                                                                                                     | <b>18</b> |
| 2.3.1      | Learning Vector Quantization (LVQ)                                                                                                                 | 18        |
| <b>3</b>   | <b>TRABALHOS RELACIONADOS</b>                                                                                                                      | <b>21</b> |
| <b>3.1</b> | <b>Comparison between MLP and LVQ neural networks for virtual upper limb prosthesis control</b>                                                    | <b>21</b> |
| <b>3.2</b> | <b>Classification of Parkinson's Disease Using EMG Signals from Different upper Limb Movements Based on Multiclass Support Vector Machine</b>      | <b>22</b> |
| <b>3.3</b> | <b>Parkinson Disease Detection Based on Voice and EMG Pattern Classification Method for Indonesian Case Study</b>                                  | <b>23</b> |
| <b>3.4</b> | <b>On the Use of t-Distributed Stochastic Neighbor Embedding for Data Visualization and Classification of Individuals with Parkinson's Disease</b> | <b>24</b> |
| <b>3.5</b> | <b>Considerações Finais</b>                                                                                                                        | <b>24</b> |
| <b>4</b>   | <b>MATERIAIS E MÉTODOS</b>                                                                                                                         | <b>26</b> |
| <b>4.1</b> | <b>Base de Dados</b>                                                                                                                               | <b>26</b> |
| <b>4.2</b> | <b>Extração de características</b>                                                                                                                 | <b>26</b> |
| <b>4.3</b> | <b>Variação dos parâmetros</b>                                                                                                                     | <b>27</b> |
| <b>4.4</b> | <b>Testes com parâmetros ótimos</b>                                                                                                                | <b>27</b> |
| <b>4.5</b> | <b>Análise dos testes</b>                                                                                                                          | <b>27</b> |
| <b>5</b>   | <b>IMPLEMENTAÇÃO</b>                                                                                                                               | <b>28</b> |
| <b>5.1</b> | <b>Leitura dos dados</b>                                                                                                                           | <b>28</b> |

|     |                                             |    |
|-----|---------------------------------------------|----|
| 5.2 | Seleção dos melhores parâmetros . . . . .   | 28 |
| 5.3 | Realização do Treinamento . . . . .         | 29 |
| 5.4 | Realização dos testes . . . . .             | 31 |
| 5.5 | Testes com os melhores parâmetros . . . . . | 32 |
| 6   | RESULTADOS E DISCUSSÃO . . . . .            | 33 |
| 7   | CONCLUSÃO . . . . .                         | 36 |
|     | REFERÊNCIAS . . . . .                       | 37 |

# 1 Introdução

A Doença de Parkinson é uma doença neurológica que acontece pela degeneração das células na região de substância negra no cérebro. Essa enfermidade causa problemas no movimento dos indivíduos. Existem cerca de 10 milhões de pessoas no mundo que sofrem com essa doença, podendo ter problemas na fala, escrita, tremores e desequilíbrio ([GARCIA; COUTO; NETO, 2021](#)).

Esses sintomas causam diversos impedimentos na vida dessas pessoas, a detecção da DP em estágio precoce é uma tarefa muito importante para o tratamento dos pacientes ([RAMOS et al., 2014](#)).

Diferentes abordagens têm sido empregadas para realização do diagnóstico da doença, tais como avaliações clínicas feitas em consultórios e exames de ressonância magnética ([VEDOLIN; MARCHIORI; RIEDER, 2004](#)). Porém, estas abordagens não são capazes de antecipar o diagnóstico da doença. Por esta razão, vários pesquisadores tem empregado o uso de IA para conseguir reconhecer pequenos padrões que deem indícios da manifestação da doença [Yadav, Singh e Pal \(2023\)](#), [Belić et al. \(2019\)](#), [Alkhatib et al. \(2020\)](#). Tal processo tem sido aplicado sinais de voz, sinais EMG e nas imagens dos exames de RM ([SRIRAM et al., 2013](#)).

Sinais EMG é uma técnica para fazer análise de atividade elétrica nos músculos, podendo ser utilizada para fazer análises através da voz, marcha e outros. No trabalho de [Oliveira et al. \(2018\)](#), por exemplo, é demonstrado o uso de uma luva com sensores inerciais integrados e sensores EMG para caracterizar os sintomas da DP através dos movimentos do punho. Para realizar essa detecção foi utilizado um algoritmo de IA. Similarmente em [Machado et al. \(2016\)](#) foram comparados algoritmos para distinguir pacientes com DP de pacientes normais, um dos algoritmos utilizados foi o algoritmo de machine learning t-distributed stochastic neighbor embedding (t-SNE). Segundo o autor, o t-SNE tenta colocar um ponto de um espaço de alta dimensão em um de baixa dimensão, de modo a preservar a identidade da vizinhança, e o algoritmo SNE converte distâncias euclidianas entre pontos de dados em probabilidades condicionais representando semelhanças.

Uma vez que não é tão simples encontrar bases de dados públicas com informações que permitam a exploração das combinações de diferentes técnicas este trabalho irá abordar a análise dos sinais EMG.

A motivação para esse trabalho é conseguir auxiliar as pessoas que sofrem com a DP e conseguir apoiar as pesquisas que utilizam IA para resolver problemas. Atualmente ainda não é possível saber qual a melhor forma de utilizar IA para fazer a detecção da DP.

Sendo assim, neste trabalho será analisada a influência dos parâmetros da rede na predição da DP baseado em uma técnica de extração de características selecionada.

## 1.1 Objetivos

### 1.1.1 Objetivo Geral

Este trabalho tem como objetivo analisar a influência da variação dos principais parâmetros da rede neural e seus impactos sobre o desempenho na classificação da DP aplicados a uma base de dados pública.

### 1.1.2 Objetivos específicos

- Treinar/testar a rede com uma base de dados de sinais EMG, publicamente disponível;
- Estudar a influência dos principais parâmetros da rede no desempenho durante a classificação;
- Testar a rede com os parâmetros que apresentaram maior eficiência na predição da DP.

## 1.2 Organização da monografia

A organização desta monografia será a seguinte: O Capítulo 2 aborda a fundamentação teórica, explicando conceitos necessários para o entendimento do trabalho. O Capítulo 3 detalha os trabalhos que possuem alguma relação com o que será feito. No Capítulo 4 é abordado a metodologia utilizada para realizar a detecção da DP, como a rede neural e o banco de dados que foram utilizados. No Capítulo 5 é apresentado detalhes sobre a implementação do código realizado para atingir os objetivos do trabalho. No capítulo 6 é apresentado os resultados e discussões que foram obtidos ao realizar os testes e treinamentos da rede nos diferentes cenários. No capítulo 7 é apresentado a conclusão do trabalho realizado, destacando as principais contribuições, limitações e também trabalhos futuros que podem ser feitos a partir deste trabalho.

## 2 Fundamentação Teórica

Neste capítulo serão apresentados os principais conceitos relacionados ao trabalho que serão a base para sua execução. Apresentar os trabalhos relacionados, que serão utilizados como base para a proposta de contribuição deste trabalho.

### 2.1 Doença de Parkinson e Sinais EMG

A DP é uma doença neurológica que acontece pela degeneração das células na região de substância negra no cérebro. Essa enfermidade causa problemas nos movimentos dos indivíduos. Uma das maneiras de auxiliar a pessoa que tem a DP a sofrer menos com a doença é fazer um diagnóstico precoce, para tentar retardar a evolução da doença no decorrer do tempo. No trabalho de [Jr e Cardoso \(2004\)](#) é demonstrado várias formas de fazer esse tratamento inicial, que pode variar conforme o nível dos sintomas. Uma das formas tradicionais de fazer a avaliação da DP é por meio de RM, segundo [Vedolin, Marchiori e Rieder \(2004\)](#). A RM, por outro lado, é um método relativamente simples, acessível e de menor custo em relação a outros métodos funcionais. Diversos estudos demonstraram o papel do método no diagnóstico da doença. Com a evolução da computação, surgiram métodos mais avançados para fazer essa avaliação, um deles é através do sinal EMG. Este sinal pode ser utilizado para fazer a predição da doença de diversas maneiras, uma dessas é feita pela autor [Putri et al. \(2018\)](#) onde foi utilizado sinais de voz e sinal EMG para a predição. No trabalho citado foram utilizados reconhecimento de padrões, extração de características e classificação, através dos sinais de voz, teve uma acurácia de 94,4%, e do sinal EMG, de 71%.

### 2.2 Fundamentos de Sinais Biomédicos

#### 2.2.1 O sinal EMG

O sinal EMG, conforme apresentado na Figura 1 é a somação algébrica de todos os sinais detectados em certa área, podendo ser afetados por propriedades musculares, anatômicas e fisiológicas, assim como pelo controle do sistema nervoso periférico e a instrumentação utilizada para a aquisição de sinais ([ENOKA, 2000](#)). Segundo o trabalho de [Andrade \(2007\)](#), a primeira dedução de que um músculo gerava potenciais elétricos foi feita por Francesco Redi em 1666 e as primeiras investigações a respeito da relação entre a eletrecidade e a contração muscular foram realizadas por Luigi Galvani em 1791. Esses sinais podem ser obtidos de diferentes áreas, de acordo com os movimentos que são

realizados.

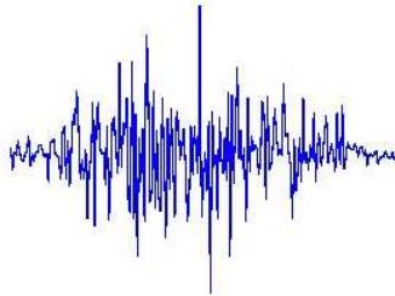


**Figura 1** – Exemplo de sinal EMG ([CAETANO et al., 2012](#)).

Para se trabalhar com o sinal EMG, várias técnicas de pré-processamento são aplicadas, tais como, normalização do sinal, janelamento, extração de características, dentre outras. Isto é feito com o objetivo de otimizar os resultados em diferentes abordagens em que o sinal EMG é empregado. A extração de característica do sinal puro é importante, pois, além de reduzir a quantidade de informação apresentada, por exemplo a uma rede neural, também pode selecionar apenas parte da informação que é relevante em cada contexto.

### 2.2.2 Processamento de sinal EMG

Ao longo do tempo tivemos evoluções nos métodos de extração de características dos sinais EMG para potencializar e facilitar o processamento da IA no reconhecimento de um determinado padrão, tais como, características no domínio do tempo(*Time Domain Features* (TDF)) ou características no domínio da frequência(*Frequency domain*(FD)) como o modelo autoregressivo(MAR).

A TDF é uma representação matemática que descreve como sinais variam ao longo do tempo, e é uma forma mais antiga de realizar o processamento. Um exemplo desse uso é citado em [Ervilha \(1999\)](#), onde foi apresentado padrões do sinal EMG de músculos dos membros inferiores com a marcha em ambiente aquático e comparado com o padrão da marcha em ambiente terrestre.

Dentre as características empregadas existem 12 diferentes, que são abordadas no trabalho de [Adem, Tessema e Simegn \(2022\)](#) e [Caetano et al. \(2012\)](#), em todas as equações  $X$  indica o sinal de entrada,  $N$  o número de sinais e  $i$  é a posição da amostra no sinal, representando a posição atual no conjunto, as formulas são:

- **Máximo Global (PEAK)**

Representa o valor máximo absoluto encontrado em um conjunto de dados.

$$\text{PEAK} = \max(|x_1|, |x_2|, \dots, |x_N|) \quad (2.1)$$

- **Média dos Valores Absolutos das Segundas Diferenças do Sinal Normalizado (MAVSDN)**

Representa o cálculo da média dos valores absolutos das segundas diferenças em um sinal normalizado.

$$\text{MAVSDN} = \frac{1}{N-2} \sum_{i=1}^{N-2} |x_{i+2} - 2x_{i+1} + x_i| \quad (2.2)$$

- **Média dos Valores Absolutos das Segundas Diferenças (MAVSD)**

Refere-se ao Cálculo da média dos valores absolutos das segundas diferenças em um sinal.

$$\text{MAVSD} = \frac{1}{N-2} \sum_{i=1}^{N-2} |x_{i+2} - 2x_{i+1} + x_i| \quad (2.3)$$

- **Média dos Valores Absolutos das Primeiras Diferenças do Sinal Normalizado (MAVFDN)**

Representa o Cálculo da média dos valores absolutos das primeiras diferenças em um sinal normalizado.

$$\text{MAVFDN} = \frac{1}{N-1} \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (2.4)$$

- **Média dos Valores Absolutos das Primeiras Diferenças (MAVFD)**

Calcula a média dos valores absolutos das primeiras diferenças em um sinal.

$$\text{MAVFD} = \frac{1}{N-1} \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (2.5)$$

- **Intervalo Interquartil do Sinal (INTERQ RANGE)**

Refere-se à diferença entre os valores do terceiro e primeiro quartis de um conjunto de dados.

$$\text{INTERQ RANGE} = Q_3 - Q_1 \quad (2.6)$$

- **Média dos Valores Absolutos das Primeiras Diferenças do Sinal (RANGE)**

Representa o cálculo da diferença entre o maior e o menor valor absoluto em um conjunto de dados.

$$\text{RANGE} = Q_3 - Q_1 \quad (2.7)$$

- **Desvio Padrão (STD)**

Mede a dispersão dos valores em relação à média, indicando o quanto os valores estão distantes da média.

$$\text{STD} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (2.8)$$

- **Variância (Variance - VAR)**

Quantifica a variabilidade de um conjunto de dados.

$$\text{VAR} = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (2.9)$$

- **Entropia Aproximada**

Mede a incerteza ou desordem em um conjunto de dados, comumente calculada usando a entropia de Shannon, que considera a probabilidade de ocorrência de cada valor.

$$H = - \sum_{i=1}^N p(x_i) \log_2(p(x_i)) \quad (2.10)$$

- **Média dos valores absolutos (Mean Absolute Value - MAV)**

Representa a média dos valores absolutos das amostras do segmento analisado.

$$\text{MAV} = \frac{1}{N} \sum_{i=1}^N ||x_i|| \quad (2.11)$$

- **Raiz quadrada média (Root Mean Square (RMS))**

Representa a magnitude média de um conjunto de valores.

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (2.12)$$

A importância da aplicação de tais técnicas é relevante, porque não é apresentado aos diferentes algoritmos de IA o sinal em sua forma discreta, mas sim, um valor numérico que representa uma característica. Então, ao invés de se apresentar 1000 diferentes valores de amplitude do sinal puro, apresenta-se por vez apenas um conjunto de características, que pode variar conforme a técnica utilizada.

## 2.3 Inteligência Artificial

### 2.3.1 Learning Vector Quantization (LVQ)

A rede LVQ de Kohonen é largamente utilizada para mapear um espaço contínuo de dados que determinam classes distintas de informações, conforme é apresentado pela Figura 2.

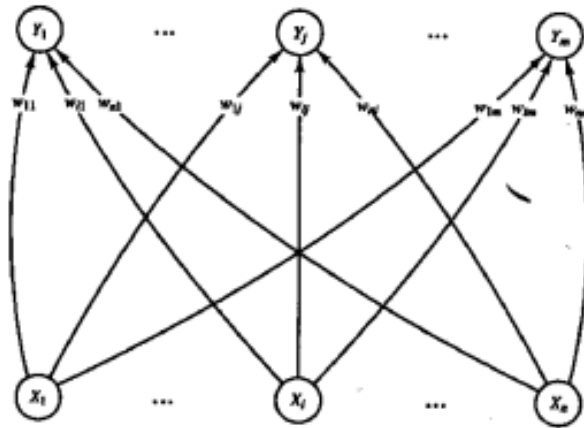


**Figura 2** – Esquema - Rede LVQ.

Na Figura 2 temos:

- de  $X_i$  até  $X_n$ : a representação dos neurônios da camada de entrada da rede, ou seja, as unidades de entrada. Cada neurônio está associado a uma informação, contida no vetor de entrada  $\mathbf{x}$  que é compartilhada com cada neurônio na camada de saída ou unidades de saída;
- de  $Y_i$  até  $Y_n$ : a representação dos neurônios na camada de saída da rede ou unidades de saída. Cada unidade de saída representa uma classe, porém pode-se ter várias unidades de saída (*cluster*) associada a uma classe de informação;
- $W_j$ : representa o vetor de peso existente entre cada unidade de entrada e as unidades de saídas;

O algoritmo de treinamento para a rede LVQ tem como principal objetivo encontrar a unidade de saída que mais se aproxima de um determinado vetor de entrada. Definida

a unidade mais próxima, caso seu vetor de pesos  $\mathbf{w}_c$  e o vetor de entrada  $\mathbf{x}$  pertençam à mesma classe, altera-se os pesos da unidade de saída na direção do vetor de entrada; se  $\mathbf{w}_c$  e  $\mathbf{x}$  pertencem a classes diferentes, os pesos da unidade de saída são ajustados afastando-a do vetor de entrada  $\mathbf{x}$ .

A seguir é apresentado um algoritmo de atualização dos pesos para uma rede LVQ.

1. Inicialização dos vetores de referência e da taxa de aprendizagem
2. Enquanto a condição de parada for falsa, execute os passos 3-7.
3. Para cada vetor de treinamento  $\mathbf{x}$ , execute os passos 4-5.
4. Encontre  $J$  tal que  $\|\mathbf{x} - \mathbf{w}_j\|$  é um mínimo.
5. Atualize  $\mathbf{w}_J$  da seguinte forma:

se  $T = C_J$ , então:

$$w_J(novo) = w_J(anterior) + \alpha[x - w_J(anterior)]$$

se  $T \neq C_J$ , então:

$$w_J(novo) = w_J(anterior) - \alpha[x - w_J(anterior)]$$

6. Reduza a taxa de aprendizagem.
7. Teste a condição de parada (esta condição pode especificar um número fixo de iterações ou um limite mínimo para a taxa de aprendizagem).

Existem na literatura diferentes formas de se inicializar os pesos da rede (KOHONEN, 1989; FAUSETT, 1994) :

- utilizar os valores dos  $m$  primeiros vetores de treinamento como pesos;
- Inicialização randômica e;
- Usar o K-means (algoritmo clássico de clusterização) ou uma rede autoorganizável.

Durante a realização deste treinamento alguns parâmetros estruturais da rede podem ter seus valores ajustados afim de colaborar com o melhor ajustamento dos valores dos pesos da rede tais como taxa de aprendizado (alpha), decaimento de alpha e tolerância.

A taxa de aprendizado determina a velocidade do aprendizado, determina o passo deste aprendizado e também regula a convergência.

O decaimento de alpha estabelece um valor proporcional com que a taxa de aprendizado será reduzida após cada ciclo de treinamento para reduzir a velocidade de ajuste dos valores dos pesos.

Por fim, o valor da tolerância pode ser utilizado para auxiliar na determinação de até quando o treinamento será realizado comparando com um valor de erro para que ela seja interrompida antes da execução de todos os ciclos o que indica mais rápida convergência da rede.

## 3 Trabalhos Relacionados

### 3.1 Comparison between MLP and LVQ neural networks for virtual upper limb prosthesis control

Neste trabalho, o principal objetivo segundo [Caetano et al. \(2012\)](#), é comparar o desempenho das redes neurais LVQ e MLP no reconhecimento de padrões EMG, utilizando técnicas de extração de característica, proporcionando um melhor controle da prótese aos indivíduos que sofreram com perda de membros superiores. Isso é importante para um indivíduo que sofre com a perda de um membro pois é necessário um grande esforço mental para conseguir se adaptar a uma prótese real.

Foram utilizadas duas técnicas de extração, TDF e MAR baseadas no domínio da frequência. Para o TDF foram utilizadas cinco feições, MAV, Inclinação média do valor absoluto(MAVS), ZC, Mudança de sinal de inclinações(SSC) e WL, já o MAR representa um sinal que depende dos valores de saída armazenados pelo sistema, ambos utilizaram segmentos de 40 amostras.

Para os testes realizados na rede LVQ, o uso de TDF em movimentos com as mãos atingiu resultados de 97% de acurácia, com 72% dos padrões de treinamento; já com o uso de TDF para movimentos dos braços foram atingidos resultados de 80% de acurácia; utilizando a técnica ARM para movimentos com as mãos foram obtidas acurácia de 99% com 11% dos padrões utilizados para treinamento e 97% de acurácia para movimentos dos braços com 10% dos padrões usados para treinamento.

Em relação a rede MLP em todos os testes, o desempenho ficou abaixo dos 85% para movimento das mãos e o tempo de treinamento foi maior que 100 segundos. Nos resultados para a rede MLP para movimentos de mão usando TDF e rede LVQ utilizando movimentos de braços com o método ARM, a taxa de erro é muito grande, portanto sua utilização não é recomendada.

Esse trabalho utiliza IA e sinais EMG, além disso esse é um tema que trabalha na área cerebral, assim como a DP.

## 3.2 Classification of Parkinson's Disease Using EMG Signals from Different upper Limb Movements Based on Multiclass Support Vector Machine

De acordo com [Adem, Tessema e Simegn \(2022\)](#), o objetivo geral desse trabalho é realizar a detecção da DP utilizando sinais mioelétricos dos músculos de membros superiores.

Foram registrados os sinais de pacientes com DP de diferentes níveis e de indivíduos que não possuem tal doença; após isso, foi realizado o pré processamento, no qual foi utilizada uma técnica para aumento e eliminação de ruídos nos dados de sinais EMG bruto. O aumento do sinal EMG bruto foi feito através de uma adição de ruído gaussiano branco e, após isso, foram utilizados os filtros IRR NOTCH para rejeitar energia na linha de 50hz e high pass para rejeitar ruídos de uma frequência muito baixa(ruídos inferiores a 20hz) e muito altas(ruídos superiores a 500hz).

Foram extraídas 14 características diferentes, 12 de domínio de tempo e 2 de domínio de frequência. Para tornar o algoritmo eficiente em relação a tempo, foram escolhidas 9 características, essa classificação foi feita através do algoritmo Relief-F esse algoritmo determina os pesos dos recursos nos casos de variável categorica multiclasse, penalizando os recursos que fornecem valores diferentes para vizinhos da mesma classe e recompensando recursos que fornecem valores diferentes para vizinhos de classes diferentes, os 9 melhores classificados e que foram utilizados foram Entropy, Mean Frequency, Skewness, Median frequency, Kurtosis, Maximum fractal length, RMS, IEMG e MAV.

Para realizar a classificação, foi utilizada a máquina de vetores de suporte (SVM) multiclasse com função de base radial(RBF), foi utilizada a abordagem um contra todos, foram divididos em conjuntos de treinamento com 480 sinais e um conjunto de testes com 120 sinais, por fim, foi feito o treinamento dos algoritmos K-vizinhos mais próximos(KNN) e Analise Discriminante Linear(LDA) para realizar a comparação.

Os resultados foram positivos, o algoritmo RBF apresentou resultados de acurácia, especificidade e sensibilidade (parâmetros de desempenho) superiores ao KNN e ao LDA. No exercício de pronação de pulso, a precisão do algoritmo RBF foi de 96.6%, especificidade de 98.5% e sensibilidade de 96.7%; já no exercício de tocar o ombro a acurácia foi de 95%, especificidade de 98.3% e sensibilidade de 95%; na ação de flexionar o cotovelo a 90° sem carga a acurácia foi de 90%, especificidade de 96.7% e sensibilidade de 90% e por último na ação de flexionar o cotovelo a 90° com carga a acurácia foi de 91.7%, especificidade de 97.1% e sensibilidade de 91,7%. Esse é um estudo semelhante ao que será apresentado, ele também busca fazer a detecção da DP utilizando algoritmos de IA e sinal EMG, porém se diferencia nas questões da utilização de músculos de membros

superiores.

### 3.3 Parkinson Disease Detection Based on Voice and EMG Pattern Classification Method for Indonesian Case Study

O objetivo do autor [Putri et al. \(2018\)](#) nesse trabalho é realizada a detecção da DP utilizando recursos de voz, eletromiografia(EMG) e redes neurais.

Foram utilizados 22 métodos de extração de recursos de voz, Programa de Voz Multidimensional (MDVP): Fo(Hz), MDVP:Fhi(Hz), MDVP:Flo(Hz), MDVP: jitter((RAP), MDVP: quociente de perturbação de período (PPQ), MDVP: shimmer, MDVP: shimmer (dB), MDVP: APQ, jitter: DDP, shimmer: quociente de perturbação de amplitude 3 (APQ3) ), shimmer: APQ5, shimmer: DDA, relação ruído-harmônico (NHR), relação harmônico-ruído (HNR), entropia de densidade de período recorrente (RPDE), análise de flutuação de tendência (DFA), spread1, spread2, dimensão de correlação (D2) , Entropia do período de pitch (PPE).

Além desses, foram utilizados 12 recursos para sinais EMG, IEMG, MAV, VAR, RMS, detector de log (LOG), WL, curtose, assimetria, frequência média ( MNF), Frequência Mediana (MDF), Potência Total (TTP), Potência Média (MNP).

Foi utilizada Rede Neural Artificial (ANN) para fazer a classificação, a ANN é composta por três camadas: camada de entrada, camada oculta e camada de saída, onde temos respectivamente os resultados dos cálculos das características, uma rede feedforward com função de transferência logarítmica-sigmóide e função de transferência sigmóide tangente hiperbólica.

Segundo [Putri et al. \(2018\)](#) Os resultados mostram que a classificação dos sinais de voz é melhor que o resultado da classificação dos sinais EMG. A precisão do teste de sinais de voz é 94,4% e sinais EMG de e 71%. Esse é um estudo semelhante ao que será apresentado, ele também busca fazer a detecção da DP, porém ele utiliza sinais de voz e faz a comparação com sinais EMG.

### 3.4 On the Use of t-Distributed Stochastic Neighbor Embedding for Data Visualization and Classification of Individuals with Parkinson's Disease

No trabalho de [Oliveira et al. \(2018\)](#), O objetivo é utilizar o algoritmo de método de incorporação estocástica de t-SNE para conseguir distinguir pacientes com DP de indivíduos que não sofrem com esse problema.

Para o trabalho foram selecionados 38 indivíduos, dentre eles indivíduos saudáveis, pacientes da DP tratados com levodopa e pacientes da DP tratados com estimulação cerebral profunda. Para realizar essa distinção, foram utilizadas características estimadas a partir da projeção de dados(PCA, mapeamento de Sammon e t-SNE) e que foram classificados por um classificador de SVM.

A extração de características foi realizada em sinais filtrados, amplitude instantânea e frequência instantânea, que foram estimadas através da transformada de Hilbert. Foram utilizados os calculos de MAV, RMS, máximo global (PEAK), média dos valores absolutos das segundas diferenças do sinal normalizado (MAVSDN), média dos valores absolutos das segundas diferenças (MAVSD), média dos valores absolutos das primeiras diferenças do sinal normalizado (MAVFDN), média dos valores absolutos das primeiras diferenças do sinal (MAVFD), intervalo interquartil do sinal, diferença entre os valores máximo e mínimo de um sinal (RANGE), desvio padrão (STD), VAR e entropia aproximada.

Por fim, foi criada uma matriz de recursos para cada método, com objetivo de verificar qual seria melhor fazendo a distinção dos pacientes com DP e indivíduos saudáveis. O algoritmo t-SNE obteve o melhor desempenho em todos os cenários com a relação de qualidade(qr) média de 99,42%, em segundo lugar o mapeamento de Sammon com QR média de 90,72%, em último lugar ficou o PCA com qr média de 81,36%. Esse trabalho também utiliza algoritmos para fazer a detecção da DP, sendo semelhante ao que será feito, além do uso de sinais EMG.

### 3.5 Considerações Finais

Com base nas pesquisas correlatas, buscou-se destacar quais as redes neurais e algoritmos de IA que foram utilizados, técnicas de pré-processamento, características extraídas para cada tipo de sinal voltadas para DP. A Tabela 1 detalha as informações relacionadas.

Através da Tabela 1, observa-se que apenas o estudo de [Caetano et al. \(2012\)](#)

**Tabela 1** – Resumo Comparativo dos Trabalhos Relacionados

| Trabalhos Relacionados        | Doença de Parkinson | LVQ | MLP | SVM | TDF | FD | Sinal EMG | Sinal de Voz |
|-------------------------------|---------------------|-----|-----|-----|-----|----|-----------|--------------|
| (ADEM; TESSEMA; SIMEGN, 2022) | ✓                   | ✗   | ✗   | ✓   | ✓   | ✓  | ✓         | ✗            |
| (OLIVEIRA et al., 2018)       | ✓                   | ✗   | ✓   | ✓   | ✓   | ✓  | ✓         | ✗            |
| (PUTRI et al., 2018)          | ✓                   | ✗   | ✓   | ✗   | ✓   | ✗  | ✓         | ✓            |
| (CAETANO et al., 2012)        | ✗                   | ✓   | ✓   | ✗   | ✓   | ✓  | ✓         | ✗            |

utiliza a rede LVQ, que possui uma arquitetura mais simplificada e de fácil implementação. Quase todos os demais trabalhos com exceção de Adem, Tessema e Simegn (2022) não utilizaram a rede MLP, que possui uma arquitetura mais complexa. O SVM foi empregado nos trabalhos de Adem, Tessema e Simegn (2022) e Oliveira et al. (2018) com o objetivo de comparação de performances. Todos os autores buscaram utilizar TDF para simplificar as informações a serem mostradas para os diferentes algoritmos durante a fase de treinamento e teste, assim como o uso do sinal EMG.

Com base nos trabalhos correlatos é que se define como objetivo do trabalho, que é a utilização da rede LVQ no diagnóstico da DP. Para isso, utilizaremos uma combinação de características no domínio do tempo, apresentados nos trabalhos durante todo o período de observação da variação dos parâmetros da rede em busca de uma melhor eficiência na predição da DP.

## 4 Materiais e Métodos

Nesse estudo foi escolhido utilizar a rede LVQ, com base no trabalho relacionado [Caetano et al. \(2012\)](#).

### 4.1 Base de Dados

Será utilizada a base de dados coletada e disponibilizada por ([OLIVEIRA et al., 2018](#)). Nela foram utilizados 4 sensores, que são acelerômetro, giroscópio, magnetômetro e eletromiográfico para a coleta dos dados, além disso, foram selecionados 38 indivíduos, 10 saudáveis, 16 com DP tratados com leodolva e 12 com DP tratados com estimulação cerebral profunda (DBS). Cada indivíduo foi instruído a repetir um conjunto de quatro atividades simples: tocar os dedos, levar o dedo ao nariz, realizar movimentos de pronação e supinação, e repouso, com cada atividade sendo repetida cinco vezes. A base contém 760 registros, onde 560 são de indivíduos que possuem DP e 200 de indivíduos que não possuem, já publicamente distribuída para comunidade científica.

Então, para cada um dos 760 registros do banco de dados teremos: 408 colunas de entrada de dados, onde serão apresentadas a rede 34 conjuntos de 12 características. Totalizando 25840 registros de entrada.

Conforme demonstrado na Figura 3, os dados não foram normalizados.

| Pathological | Gyro1MAV | Gyro1RMS | Gyro1PEAK | Gyro1MAV | Gyro1MAV | Gyro1MAV | Gyro1MAV | Gyro1INTE | Gyro1RAN | Gyro1STD | Gyro1VAR | Gyro1APPE |
|--------------|----------|----------|-----------|----------|----------|----------|----------|-----------|----------|----------|----------|-----------|
| 1            | 0,071441 | 0,087553 | 0,240604  | 1,032614 | 0,090452 | 7,121679 | 0,054644 | 0,127302  | 0,433149 | 0,087595 | 0,007673 | 1,12848   |
| 0            | 0,109702 | 0,135168 | 0,452398  | 0,608175 | 0,082245 | 2,399313 | 0,043878 | 0,180205  | 0,677468 | 0,135232 | 0,018288 | 0,810382  |

**Figura 3** – Dados do banco.

### 4.2 Extração de características

Nessa base, o sinal bruto já foi processado e as características já foram extraídas, portanto não é possível pegar os sinais e aplicar outra técnica de pré-processamento. Considerando essas informações, serão exploradas variações nos parâmetros da rede selecionada e comparados os resultados da eficiência alcançada com o trabalho [Oliveira et al. \(2018\)](#).

A extração de características foi realizada sobre os sinais filtrados (FS), a amplitude instantânea (IA) e a frequência instantânea (IF), estimada a partir da transformada de Hilbert, representada em cada um dos bancos.

Foram estimadas as seguintes características: MAV, RMS, PEAK, MAVSDN, MAVSD, MAVFDN, MAVFD, INTERQ RANGE, RANGE, STD, VAR e Entropia Aproximada.

### 4.3 Variação dos parâmetros

Para observar a influência dos parâmetros na performance da rede, a abordagem a seguir é descrita:

- Durante a variação, os parâmetros abaixo permanecerão fixos: número de classes: 2; número de entradas: 12; número de clusters ou número de neurônios na camada de saída da rede: 3; taxa de aprendizado(alpha): 0.025; decaimento de alpha: 0.95; tolerância: 0.0001; número de épocas: 100.
- Para variação de cada um dos parâmetros (taxa de aprendizagem(alpha), decaimento de alpha, tolerância e núcleo de clusters(unidades de saída), variar-se-a um parâmetro mantendo os demais fixos, dentro da faixa:
  1. Valores de alpha: 0.01 a 0.99;
  2. Valores de dec. Alpha: 0.01 a 0.99;
  3. Valores de tolerância: 0.001 a 0.099;
  4. Clusters: 7 a 70 (10 intervalos);

### 4.4 Testes com parâmetros ótimos

Após verificado quais os parâmetros ótimos, será realizado um teste utilizando todos de uma única só vez.

### 4.5 Análise dos testes

A fim de observar como será o comportamento da rede para a base de dados selecionada, serão extraídas informações sobre a **eficiência** (Equação 4.1) durante a variação de cada um deles. As informações extraídas poderão ser exibidas por meio de gráficos ou tabelas, para devidas análises.

$$E = \frac{N_{acertos}}{N_{total}} \quad (4.1)$$

## 5 Implementação

Este capítulo apresenta detalhes da implementação realizada a fim de atingir o objetivo geral do trabalho, desde a leitura dos dados, definição das configurações para o teste de cada parâmetro, até a coleta e processamento dos dados gerados.

### 5.1 Leitura dos dados

Para a leitura dos dados, foram necessárias a realização de alguns ajustes, pois, os dados estavam em uma tabela com colunas que não seriam utilizadas. Para isso, elas foram desconsideradas no momento da leitura. O trecho de código exibido no bloco 5.1, ilustra os detalhes.

```
1 def ler_excel(file_path, sheet_name=0):
2     # Ler os dados do arquivo Excel
3     df = pd.read_excel(file_path, sheet_name=sheet_name)
4     # Converter o DataFrame em uma lista de listas
5     conteudo = df.values.tolist()
6     return conteudo
7
```

**Listing 5.1** – Trecho de código utilizado para leitura dos dados.

Após o carregamento dos dados, é necessário definir as configurações de cada processo (treinamento/teste) a ser realizado.

### 5.2 Seleção dos melhores parâmetros

Para selecionar a quantidade de padrões de treinamento a ser utilizado na etapa de treinamento e testes, foi implementada a rotina exibida no bloco de código 5.2. Além disso, também tem se que informar qual o parâmetro será avaliado por vez (Alpha, Decaimento de Alpha, Tolerância e Clusters).

```
1 print("Qual o percentual de dados usados para treinamento")
2 opcao1 = input("Informe um valor(ex:70):")
3 qtd_totaldados = len(dados)
4 per_trei = round(((int(opcao1)/100)*qtd_totaldados)
5 print(f"O valor do percentual foi: {per_trei}")
6
7 print("Selecione a opcao de variacao de parametros:")
8 print("1-Alpha")
9 print("2-Decaimento de alpha")
10 print("3-Tolerancia")
```

```

11 print("4-Clusters")
12 opcao = input("Digite uma das opcoes:")
13 print(f"Opcao selecionada: {opcao}. ")
14

```

**Listing 5.2** – Leitura dos dados de Parametrização

Uma vez definidas as configurações, os dados são separados dinamicamente conforme exibido no bloco de código 5.3, para serem apresentados a rede para o devido treinamento e teste.

```

1 dados_embaralhado = dados[:]
2 random.shuffle(dados_embaralhado)
3 treinamento = dados_embaralhado[:per_trei]
4 n = qtd_totaldados - per_trei
5 teste = dados_embaralhado[-n:]
6 input_data = np.array([sublist[:-1] for sublist in treinamento])
7 test_data = np.array([sublist[:-1] for sublist in teste])
8 labels = np.array([sublist[-1] for sublist in treinamento])
9 test_labels = np.array([sublist[-1] for sublist in teste])
10

```

**Listing 5.3** – Separação dos dados de Teste e Treinamento

## 5.3 Realização do Treinamento

Na realização do treinamento, para realizar a variação dos parâmetros que foram escolhidos, foi utilizado uma estrutura de repetição, que varia o valor do parametro entre cada uma das 10 possibilidades citadas na metodologia, conforme mostrado no bloco 5.4.

```

1 while contador < 10:
2     print(f"Contador:{contador}.")
3     t1 = time()
4     prev_error = float('inf')
5     if opcao == '1':
6         self.learning_rate = parVar[contador]
7         print(f"##### Variando taxa de aprendizado:{self.
learning_rate}")
8     if opcao == '2':
9         self.alpha_decay = parVar[contador]
10        print(f"##### Variando Decaimento de Alpha:{self.
alpha_decay}")
11    if opcao == '3':
12        self.tolerance = parVar[contador]
13        print(f"##### Variando a Tolerancia:{self.tolerance}
")
14    if opcao == '4':

```

```

15         self.num_clusters = parVar[contador]
16         self.weights = np.random.rand(self.num_classes, self.
num_input, self.num_clusters)
17         print(f"##### Variando a Clusters:{self.num_clusters
}")
18

```

**Listing 5.4** – Variando os parâmetros de acordo com a configuração definida

O trecho de código do bloco 5.5 está relacionado com o algoritmo de treinamento, nele é verificado se ocorreu convergência e qual foi a convergência alcançada naquela determinada época.

```

1         for epoch in range(epochs):
2             error = 0
3             for i in range(len(input_data)):
4                 input_vector = input_data[i]
5                 label = labels[i]
6                 winner_index, cluster_index = self._find_winner(
input_vector)
7                 self._update_weights(input_vector, label,
winner_index, cluster_index)
8                 error += self._calculate_error(input_vector, label,
winner_index, cluster_index)
9                 error /= len(input_data) # Media do erro por epoca
10
11                 print(f"DifErros: {abs(error - prev_error)},
ErroAnterior: {prev_error}, Erro: {error}.")
12
13                 if abs(prev_error - error) < self.tolerance:
14                     print(f"Convergencia alcançada na epoca {epoch+1}.")
15                     break
16                 prev_error = error
17                 self.learning_rate *= self.alpha_decay # Decaimento do
alpha a cada epoca
18
19             if epoch==99 :
20                 print("")
21                 print(f"Nao houve convergencia.")
22

```

**Listing 5.5** – Algoritmo de Treinamento

Finalizado o treinamento, automaticamente o processo de teste é iniciado, conforme descrito na seção a seguir.

## 5.4 Realização dos testes

A identificação de qual é o grupo que o conjunto de dado apresentado pertence é parte importante tanto na fase de treinamento quanto na de teste. O trecho de código exibido pelo bloco 5.6 é utilizado para calcular a distância de um vetor de entrada e os restante dos pesos de uma rede, ele é utilizado para determinar o índice do vencedor e qual seu *cluster*, para que isso seja feito é utilizado a função *findwinner*.

```
1 def _find_winner(self, input_vector):
2     distances = np.zeros((self.num_classes, self.num_clusters))
3     for i in range(self.num_classes):
4         for j in range(self.num_clusters):
5             distances[i, j] = np.linalg.norm(input_vector - self.weights
6 [i, :, j])
7     winner_index, cluster_index = np.unravel_index(distances.argmin(),
8 distances.shape)
9     return winner_index, cluster_index
```

**Listing 5.6** – Cálculo da Distância Euclidiana

A função *predict* apresentada no bloco 5.7 identifica o índice do vencedor. Para que a identificação seja feita é utilizada a função *findwinner*, que calcula o índice da classe e do cluster mais próximo ao vetor de entrada.

```
1 def predict(self, input_vector):
2     winner_index, _ = self._find_winner(input_vector)
3     return winner_index
4
```

**Listing 5.7** – Identificação da Classe

O cálculo eficiência do algoritmo após treinamento é realizado por meio da função *calcular\_eficiencia*(bloco 5.8). Nela é utilizada uma estrutura de repetição que verifica por meio da função *predict* se os dados de teste são iguais ao das *labels* adquiridas, caso sejam, eles são contabilizados e por fim é verificado quantos estão corretos.

```
1 def calcular_eficiencia(self, test_data, test_labels, parametro):
2     corretos = 0
3     for i in range(len(test_data)):
4         prediction = lvq.predict(test_data[i])
5         if test_labels[i] == prediction:
6             corretos = corretos + 1;
7         print("Input:", test_data[i], "Predicted Class:", prediction
8 )
9     eficiencia =0;
10    print("i:", i+1, "corretos:",corretos)
11    eficiencia = corretos / (i+1);
```

```
11     print("Eficiencia alcançada:", eficiencia*100)
12     print("")
13
```

**Listing 5.8** – Cálculo da Eficiência.

Esta função é importante pois, por meio da extração destes dados durante a variação de cada parâmetro, será possível identificar quais os melhores em cada caso isolado, ou seja, os parâmetros ótimos.

## 5.5 Testes com os melhores parâmetros

Após obter os parâmetros com melhor eficiência, os mesmos serão utilizados para verificar qual a eficiência da rede quando combinados. Neste caso, após a descoberta, a função de treinamento não utiliza mais a variação de parametros, pois os mesmos já são definidos previamente.

```
1     lvq = LVQ(num_input=12, num_classes=2, num_clusters=28,
2             initial_learning_rate=0.911, alpha_decay=0.911, tolerance=0.0911)
```

**Listing 5.9** – Definindo instância para Treinamento e Teste com os parâmetros ótimos

## 6 Resultados e Discussão

A fim de descobrir quais eram os melhores parâmetros, foi realizado a variação de cada um deles conforme apresentado na seção 4.3. Além disso, também foi observado a variação da eficiência alterando o percentual de padrões apresentados para treinamento e teste. Para cada um deles, foram escolhidos os valores de 1%, 10%, 40%, 50%, 80% e 90% para a quantidade de valores de treino.

As Tabelas 2, 3, 4 e 5 apresentam a eficiência obtida com a variação de cada um dos parâmetros, esses representados na primeira coluna, e da porcentagem de padrões de entrada utilizados para treinamento, representados na primeira linha.

Pode-se observar por meio da Tabela 2 que não foram obtidas diferenças ao variar os valores de Alpha.

| %            | 1      | 10     | 40     | 80     | 95            |
|--------------|--------|--------|--------|--------|---------------|
| <b>0.011</b> | 31.60% | 31.54% | 31.37% | 32.02% | <b>67.87%</b> |
| <b>0.111</b> | 31.60% | 31.54% | 31.37% | 32.02% | <b>67.87%</b> |
| <b>0.211</b> | 31.60% | 31.54% | 31.37% | 32.02% | <b>67.87%</b> |
| <b>0.311</b> | 31.60% | 31.54% | 31.37% | 32.02% | <b>67.87%</b> |
| <b>0.411</b> | 31.60% | 31.54% | 31.37% | 32.02% | <b>67.87%</b> |
| <b>0.511</b> | 31.60% | 31.54% | 31.37% | 32.02% | <b>67.87%</b> |
| <b>0.611</b> | 31.60% | 31.54% | 31.37% | 32.02% | <b>67.87%</b> |
| <b>0.711</b> | 31.60% | 31.54% | 31.37% | 32.02% | <b>67.87%</b> |
| <b>0.811</b> | 31.60% | 31.54% | 31.37% | 32.02% | <b>67.87%</b> |
| <b>0.911</b> | 31.60% | 31.54% | 31.37% | 32.02% | <b>67.87%</b> |

**Tabela 2** – Resultados de variação de Alpha

Somente ao variar o percentual de dados utilizado para testes e treino é que se pode observar um ganho na eficiência. São diversas as razões que colaboram para este comportamento da rede. A natureza dos padrões de entrada (tamanho da grandezas). Conforme estabelecido por Fausett (2006) o valor de Alpha determina o tamanho dos passos que o algoritmo de otimização dá na direção do gradiente, o decaimento de alpha ajusta o processo de aprendizado diminuindo alpha ao longo do tempo e a tolerância atua como um critério de parada, limitando o treinamento a um erro aceitável permitindo um ajuste adequado aos pesos da rede. O *overfitting* é uma das hipóteses pelo qual os parâmetros não alteraram a eficiência após etapa de treinamento. Nele, a rede foi treinada a tal ponto que, mesmo passando poucos padrões, ela não consegue generalizar o que é ou não é, comprometendo o desempenho. No caso em questão, mesmo utilizando tolerância para verificar a convergência na rede, isso não garantiu que ela não pudesse chegar ao *overfitting*, uma vez que o erro (diferença entre as atualizações dos pesos entre uma época

e outra) não foi menor que a tolerância e somente ser interrompido quando alcançar 100 épocas. O mesmo comportamento pode ser observado nas Tabelas 3 e 4.

| %            | 1      | 10     | 40     | 80     | 95            |
|--------------|--------|--------|--------|--------|---------------|
| <b>0.011</b> | 68.38% | 68.43% | 68.71% | 68.84% | <b>70.97%</b> |
| <b>0.111</b> | 68.38% | 68.43% | 68.71% | 68.84% | <b>70.97%</b> |
| <b>0.211</b> | 68.38% | 68.43% | 68.71% | 68.84% | <b>70.97%</b> |
| <b>0.311</b> | 68.38% | 68.43% | 68.71% | 68.84% | <b>70.97%</b> |
| <b>0.411</b> | 68.38% | 68.43% | 68.71% | 68.84% | <b>70.97%</b> |
| <b>0.511</b> | 68.38% | 68.43% | 68.71% | 68.84% | <b>70.97%</b> |
| <b>0.611</b> | 68.38% | 68.43% | 68.71% | 68.84% | <b>70.97%</b> |
| <b>0.711</b> | 68.38% | 68.43% | 68.71% | 68.84% | <b>70.97%</b> |
| <b>0.811</b> | 68.38% | 68.43% | 68.71% | 68.84% | <b>70.97%</b> |
| <b>0.911</b> | 68.38% | 68.43% | 68.71% | 68.84% | <b>70.97%</b> |

**Tabela 3** – Resultados de variação de Decaimento Alpha

| %             | 1      | 10     | 40     | 80            | 95     |
|---------------|--------|--------|--------|---------------|--------|
| <b>0.0011</b> | 68.43% | 31.54% | 67.96% | <b>69.40%</b> | 68.18% |
| <b>0.0111</b> | 68.43% | 31.54% | 67.96% | <b>69.40%</b> | 68.18% |
| <b>0.0211</b> | 68.43% | 31.54% | 67.96% | <b>69.40%</b> | 68.18% |
| <b>0.0311</b> | 68.43% | 31.54% | 67.96% | <b>69.40%</b> | 68.18% |
| <b>0.0411</b> | 68.43% | 31.54% | 67.96% | <b>69.40%</b> | 68.18% |
| <b>0.0511</b> | 68.43% | 31.54% | 67.96% | <b>69.40%</b> | 68.18% |
| <b>0.0611</b> | 68.43% | 31.54% | 67.96% | <b>69.40%</b> | 68.18% |
| <b>0.0711</b> | 68.43% | 31.54% | 67.96% | <b>69.40%</b> | 68.18% |
| <b>0.0811</b> | 68.43% | 31.54% | 67.96% | <b>69.40%</b> | 68.18% |
| <b>0.0911</b> | 68.43% | 31.54% | 67.96% | <b>69.40%</b> | 68.18% |

**Tabela 4** – Resultados de variação de Tolerância

Já na Tabela 5 podemos observar uma variação da eficiência ao variar o *cluster*.

| %         | 1             | 10            | 40            | 80            | 95            |
|-----------|---------------|---------------|---------------|---------------|---------------|
| <b>7</b>  | 31.57%        | 31.63%        | <b>68.51%</b> | 31.81%        | 31.88%        |
| <b>14</b> | <b>68.42%</b> | <b>68.36%</b> | <b>68.51%</b> | 31.81%        | 31.88%        |
| <b>21</b> | 65.46%        | <b>68.36%</b> | 31.48%        | 68.18%        | <b>68.11%</b> |
| <b>28</b> | 61.53%        | 33.52%        | 34.33%        | <b>68.42%</b> | 34.59%        |
| <b>35</b> | 62.80%        | 59.29%        | 62.86%        | 65.71%        | 67.95%        |
| <b>42</b> | 42.24%        | 36.82%        | 59.29%        | 38.64%        | 31.73%        |
| <b>49</b> | 42.76%        | 37.55%        | 37.53%        | 67.93%        | 63.69%        |
| <b>56</b> | 66.19%        | 38.32%        | 48.71%        | 53.28%        | 34.90%        |
| <b>63</b> | 61.89%        | 46.24%        | 47.72%        | 62.11%        | 66.56%        |
| <b>70</b> | 42.32%        | 56.60%        | 53.45%        | 56.19%        | 34.13%        |

**Tabela 5** – Resultados de variação dos Clusters

Conforme estabelecido por Fausett (2006) a variação no número de *clusters* está diretamente relacionada a variação no número de camadas de saída da rede neural. Assim

podemos perceber que em alguns casos a variação da quantidade destas camadas permitiu alcançar melhor classificação do padrão apresentado aumentando a eficiência. Pode-se observar que, variando tanto a quantidade de padrões apresentados para treinamento e o valor do próprio parâmetro, as eficiências mais elevadas alcançadas foram de: 68.51% para 7 *clusters*, 68.51% para 14 *clusters*, 68.36% para 21 *clusters* e 68.42% para 28 *clusters*.

Por fim, após apresentados os resultados de todos os parâmetros variados isoladamente, são apresentados na Tabela 6 os resultados da aplicação dos melhores parâmetros encontrados combinados, nos melhores casos.

| %  | 7      | 14            | 21            | 28            |
|----|--------|---------------|---------------|---------------|
| 10 | 31.50% | 31.64%        | 31.46%        | <b>68.47%</b> |
| 40 | 68.43% | 32.03%        | <b>68.54%</b> | 31.61%        |
| 50 | 31.94% | <b>68.45%</b> | 31.27%        | 31.38%        |
| 80 | 68.38% | 31.48%        | 31.28%        | <b>68.80%</b> |
| 90 | 31.46% | 32.46%        | <b>67.91%</b> | 31.23%        |

**Tabela 6** – Resultados dos clusters em diferentes percentuais

De acordo com a Tabela 6, quando comparado ao trabalho de [Oliveira et al. \(2018\)](#), onde o autor conseguiu atingir eficiência de até 99%, os resultados não foram tão expressivos, o melhor resultado obtido foi ao utilizar 28 *clusters* e 80% dos dados para treino e a sua precisão foi de 68.80%. Quando analisado individualmente e pensado na utilização desse algoritmo.

Diante dos resultados, fica claro que o algoritmo não conseguiu atingir o desempenho equivalente ao alcançado no trabalho de [Oliveira et al. \(2018\)](#). No entanto, já que não é possível aplicar técnicas de pré-processamento as características extraídas presentes na base de dados, uma abordagem relevante está em verificar a correlação existente entre as características existentes, pois, algumas delas podem não ser relevantes para que a rede consiga generalizar conseguindo discernir entre diferentes entradas a classificação correta([ARAUJO; COELHO, 2009](#); [YAMADA, 2021](#)).

## 7 Conclusão

Este estudo teve como objetivo identificar quais indivíduos presentes em uma base de dados pública possuíam a DP utilizando a rede LVQ. Na literatura, nem sempre são claramente mencionadas nas publicações, quais os parâmetros do algoritmo de inteligência artificial utilizado, permitiu alcançar tal eficiência. Por esta razão, foi apresentada uma metodologia que permitiu variar os parâmetros da rede LVQ durante a fase de treinamento e teste e observar qual o comportamento desta para os diferentes casos.

A base de dados utilizada neste estudo foi fornecida por [Oliveira et al. \(2018\)](#) cujo o sinal EMG já foi processado e no qual tivemos acesso as características extraídas. Para validar a metodologia seguinte abordagem adotada: variar cada um dos parâmetros, mantendo os demais fixos (taxa de aprendizagem(alpha), decaimento de alpha, tolerância e clusters) e também quantidade de padrões apresentados na fase de treinamento([CAETANO et al., 2012](#)). Após verificado quais são os parâmetros ótimos, foi realizado um teste utilizando todos de uma única vez.

Com isto, a rede LVQ utilizando os parâmetros ótimos conseguiu atingir eficiência de até 68.80% ao utilizar 28 *clusters* e 80% dos dados para treino. Uma das razões para o valor não ter conseguido ser maior, pode se dar por conta de um possível overfitting, no qual a rede foi treinada a tal ponto que, mesmo passando poucos padrões, ela não consegue generalizar o que é ou não é, comprometendo o desempenho e nesse caso, mesmo utilizando tolerância para verificar a convergencia na rede.

Para trabalhos futuros, para verificar se a rede LVQ pode ainda ter um resultado melhor utilizando essa base de dados, é importante verificar quais características são realmente relevantes a serem apresentadas como padrões de entrada para o treinamento da rede. Para isto, técnicas análises de componentes principais(PCA) podem ser exploradas ([ARAUJO; COELHO, 2009](#); [YAMADA, 2021](#)).

# Referências

- ADEM, H. M.; TESSEMA, A. W.; SIMEGN, G. L. Classification of parkinson's disease using emg signals from different upper limb movements based on multiclass support vector machine. **International Journal Bioautomation**, Bulgarska Akademiya na Naukite/Bulgarian Academy of Sciences, v. 26, n. 1, p. 109, 2022. DOI: <<https://doi.org/10.7546/ijba.2022.26.1.000849>>. Citado 3 vezes nas páginas 15, 22 e 25.
- ALKHATIB, R.; DIAB, M. O.; CORBIER, C.; BADAUI, M. E. Machine learning algorithm for gait analysis and classification on early detection of parkinson. **IEEE Sensors Letters**, IEEE, v. 4, n. 6, p. 1–4, 2020. DOI: <<https://doi.org/10.1109/LSENS.2020.2994938>>. Citado na página 12.
- ANDRADE, N. A. Desenvolvimento de um sistema de aquisição e processamento de sinais eletromiográficos de superfície para a utilização no controle de próteses motoras ativas. 2007. Citado na página 14.
- ARAUJO, W. O. de; COELHO, C. J. Análise de componentes principais (pca). **University Center of Anápolis, Anápolis**, 2009. Citado 2 vezes nas páginas 35 e 36.
- BELIĆ, M.; BOBIĆ, V.; BADŽA, M.; ŠOLAJA, N.; ĐURIĆ-JOVIČIĆ, M.; KOSTIĆ, V. S. Artificial intelligence for assisting diagnostics and assessment of parkinson's disease—a review. **Clinical neurology and neurosurgery**, Elsevier, v. 184, p. 105442, 2019. DOI: <<https://doi.org/10.1016/j.clineuro.2019.105442>>. Citado na página 12.
- CAETANO, D. S. D. et al. Comparação entre as redes lvq e mlp no controle de próteses virtuais para membros superiores. Universidade Federal de Uberlândia, 2012. Citado 7 vezes nas páginas 6, 15, 21, 24, 25, 26 e 36.
- ENOKA, R. M. As bases neuromecânicas da cinesiologia. In: **As bases neuromecânicas da cinesiologia**. [S.l.: s.n.], 2000. p. 450–450. Citado na página 14.
- ERVILHA, U. F. **Estudo da variação angular da articulação do joelho e do sinal eletromiográfico no domínio temporal durante a marcha humana em ambiente aquático**. Tese (Doutorado) — Universidade de São Paulo, 1999. Citado na página 15.
- FAUSETT, L. (Ed.). **Fundamentals of neural networks: architectures, algorithms, and applications**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994. ISBN 0-13-334186-0. Citado na página 19.
- FAUSETT, L. V. **Fundamentals of neural networks: architectures, algorithms and applications**. [S.l.]: Pearson Education India, 2006. Citado 2 vezes nas páginas 33 e 34.
- GARCIA, A. C. C.; COUTO, I. A.; NETO, L. G. Doença de parkinson e transtorno depressivo: uma revisão bibliográfica parkinson's disease and depressive disorder: a

literatura review. **Brazilian Journal of Health Review**, v. 4, n. 5, p. 18921–18930, 2021. DOI: <<https://doi.org/10.34119/bjhrv4n5-036>>. Citado na página 12.

JR, A. L. T.; CARDOSO, F. Tratamento inicial da doença de parkinson. **Revista neurociências**, v. 12, n. 3, p. 141–146, 2004. DOI: <<https://doi.org/10.34024/rnc.2004.v12.8861>>. Citado na página 14.

KOHONEN, T. **Self-organization and Associative Memory**. 3rd. ed. [S.l.]: Springer-Verlag, 1989. DOI: <<https://doi.org/10.1007/978-3-642-88163-3>>. Citado na página 19.

MACHADO, A. R.; ZAIDAN, H. C.; PAIXÃO, A. P. S.; CAVALHEIRO, G. L.; OLIVEIRA, F. H. M.; JÚNIOR, J. A. F. B.; NAVES, K.; PEREIRA, A. A.; PEREIRA, J. M.; POURATIAN, N. et al. Feature visualization and classification for the discrimination between individuals with parkinson's disease under levodopa and dbs treatments. **Biomedical engineering online**, Springer, v. 15, p. 1–22, 2016. DOI: <<https://doi.org/10.1186/s12938-016-0290-y>>. Citado na página 12.

OLIVEIRA, F. H. M.; MACHADO, A. R.; ANDRADE, A. O. et al. On the use of t-distributed stochastic neighbor embedding for data visualization and classification of individuals with parkinson's disease. **Computational and mathematical methods in medicine**, Hindawi, v. 2018, 2018. DOI: <<https://doi.org/10.1155/2018/8019232>>. Citado 6 vezes nas páginas 12, 24, 25, 26, 35 e 36.

PUTRI, F.; CAESARENDRA, W.; PAMANASARI, E. D.; ARIYANTO, M.; SETIAWAN, J. D. Parkinson disease detection based on voice and emg pattern classification method for indonesian case study. **JEMMME (Journal of Energy, Mechanical, Material, and Manufacturing Engineering)**, v. 3, n. 2, p. 87–98, 2018. DOI: <<https://doi.org/10.22219/jemmme.v3i2.6977>>. Citado 3 vezes nas páginas 14, 23 e 25.

RAMOS, M. L.; NEVES, D. R.; LIMA, V. P.; ORSINI, M.; MACHADO, D.; BASTOS, V. H. d. V.; SANTOS, A. P. Análise de parâmetros pneumofuncionais em pacientes com doença de parkinson: estudo piloto. **Rev Bras Neurol**, v. 50, n. 2, p. 38–43, 2014. Citado na página 12.

SRIRAM, T.; RAO, M. V.; NARAYANA, G.; KALADHAR, D.; VITAL, T. P. R. Intelligent parkinson disease prediction using machine learning algorithms. **Int. J. Eng. Innov. Technol**, v. 3, p. 212–215, 2013. Citado na página 12.

VEDOLIN, L.; MARCHIORI, E.; RIEDER, C. Avaliação da doença de parkinson pela ressonância magnética. **Radiologia Brasileira**, SciELO Brasil, v. 37, p. 83–90, 2004. Citado 2 vezes nas páginas 12 e 14.

YADAV, S.; SINGH, M. K.; PAL, S. Artificial intelligence model for parkinson disease detection using machine learning algorithms. **Biomedical Materials & Devices**, Springer, p. 1–13, 2023. DOI: <<https://doi.org/10.1007/s44174-023-00068-x>>. Citado na página 12.

YAMADA, F. T. **Um estudo sobre a influência de métodos imparciais de análise de componentes principais em problemas de classificação**. Tese (Doutorado) — [sn], 2021. Citado 2 vezes nas páginas 35 e 36.