

Estudo de Modelos Neurais Supervisionados em Regressão e Classificação Multiclasse: ADALINE, Perceptron Simples e MLP com Validação Monte Carlo

Bruno Matos de Araújo - 2225673, João Pedro Rego Magalhães - 2517985

Abstract — Abstract — Este relatório apresenta uma atividade prática sobre a aplicação de técnicas de regressão e classificação utilizando redes neurais artificiais, com base em dois conjuntos de dados distintos. Na primeira etapa, aborda-se um problema bidimensional: para regressão, dados de um aerogerador (com a velocidade do vento como variável independente e a potência gerada como dependente) e, para classificação, um problema binário utilizando o conjunto Spiral3d.csv. Na segunda etapa, o foco é um problema multiclasse de reconhecimento de condições da coluna vertebral, onde seis preditores clínicos fornecem informações para identificar três estados (normal, hérnia de disco e espondilolistese) por meio de codificação one-hot. Os modelos implementados – ADALINE, Perceptron de Múltiplas Camadas (MLP) e Rede de Função de Base Radial (RBF) – foram validados utilizando simulações de Monte Carlo, permitindo uma análise comparativa detalhada do desempenho, além de investigações sobre underfitting e overfitting.

Palavras-chave — Redes Neurais Artificiais, Regressão, Classificação Multiclasse, ADALINE, MLP, RBF, Monte Carlo, One-Hot Encoding, Underfitting, Overfitting.

I. INTRODUÇÃO

Modelos de inteligência artificial têm desempenhado um papel fundamental na replicação de processos decisórios humanos, permitindo a solução de problemas complexos em diversas áreas. No presente trabalho, exploramos a aplicação de redes neurais artificiais—tanto em suas versões lineares quanto não lineares—para abordar duas etapas distintas. A primeira etapa consiste na resolução de problemas bidimensionais, envolvendo tarefas de regressão e classificação, enquanto a segunda etapa foca em um problema multiclasse para o reconhecimento de condições da coluna vertebral.

Na primeira etapa, aplicamos técnicas preditivas a partir de dois conjuntos de dados. Para a tarefa de regressão, utilizamos o arquivo aerogerador.dat, no qual a velocidade do vento (variável independente) é empregada para estimar a potência gerada (variável dependente) em um contexto de dados unidimensionais ($p = 1$, $N = 2250$). Paralelamente, avaliamos a tarefa de classificação usando o conjunto de dados Spiral3d.csv – composto por três preditores e rótulos binários – por meio de algoritmos como ADALINE, Perceptron de Múltiplas Camadas (MLP) e Rede de Função de Base Radial (RBF). Estratégias de validação robusta, utilizando simulações de Monte Carlo ($R = 250$ rodadas), possibilitam a análise das métricas de desempenho (erro quadrático médio para regressão e acurácia, sensibilidade e especificidade para classificação) e a investigação de cenários de underfitting e overfitting através da variação das topologias dos modelos.

A segunda etapa concentra-se em um problema real de reconhecimento multiclasse, utilizando o conjunto de dados coluna_vertebral.csv. Nesse contexto, seis preditores clínicos – incluindo incidência pélvica, inclinação pélvica, ângulo da lordose lombar, ângulo sacral, raio pélvico e grau de espondilolistese – são empregados para classificar os

pacientes em três categorias: Normal (NO), Hérnia de Disco (DH) e Espondilolistese (SL). A codificação dos rótulos é realizada via one-hot encoding, e os modelos ADALINE, MLP e RBF são ajustados e avaliados por meio de simulações de Monte Carlo ($R = 100$ rodadas), com 80% dos dados destinados ao treinamento e 20% ao teste. Essa abordagem permite uma análise aprofundada do comportamento dos modelos em diferentes configurações, comparando as métricas de acurácia, sensibilidade e especificidade, bem como a análise das matrizes de confusão e curvas de aprendizado.

II. METODOLOGIA

O presente trabalho foi dividido em duas etapas principais, abordando problemas de regressão e classificação por meio da implementação “do zero” de modelos de redes neurais artificiais lineares e não lineares. Os algoritmos desenvolvidos foram baseados nos pseudocódigos clássicos de ADALINE, Perceptron Simples e Perceptron de Múltiplas Camadas (MLP). Cada modelo foi configurado por meio da definição dos hiperparâmetros essenciais (taxa de aprendizagem, número máximo de épocas e critério de parada) e validado por meio de simulações de Monte Carlo.

I. Primeira Etapa – Regressão e Classificação para Problemas Bidimensionais

1. Tarefa de Regressão

Objetivo: Prever a potência gerada por um aerogerador a partir da velocidade do vento (arquivo “aerogerador.dat”, $p = 1$, $N = 2250$).

Procedimentos:

- **Organização dos dados e visualização:** Os dados foram organizados em uma matriz X – com a inclusão de uma coluna constante para o bias – e em um vetor y , de modo que cada instância experimental corresponda à medição da velocidade do vento e à potência gerada. Um gráfico de dispersão foi gerado para explorar a relação entre a velocidade e a potência, permitindo a análise exploratória dos padrões presentes.
- **Implementação dos Modelos:** Foram aplicados dois modelos preditivos:
 - **ADALINE:** Implementado seguindo os pseudocódigos apresentado em sala de aula, o ADALINE foi configurado definindo a taxa de aprendizagem (η), o número máximo de épocas e a precisão (ϵ). Inicializados os

pesos (incluindo o bias), o modelo atualiza-os iterativamente com a regra:

$$w(t+1) = w(t) + \eta [d(t) - u(t)] x(t)$$

onde $u(t) = w^T(t)x(t)$.

O treinamento é interrompido quando a variação do Erro Quadrático Médio (EQM) fica abaixo de ϵ ou quando o número máximo de épocas é atingido.

- **Perceptron de Múltiplas Camadas (MLP):** O MLP foi configurado definindo o número de camadas ocultas e o número de neurônios em cada camada, juntamente com funções de ativação não lineares (como tangente hiperbólica ou sigmoide).

Durante o treinamento, os dados com bias são propagados forward para calcular as ativações e, em seguida, o erro é retropropagado (backward) para atualizar os pesos, visando minimizar o EQM. Diversas topologias foram testadas para identificar cenários de underfitting e overfitting.

- **Validação via Monte Carlo:** Foram realizadas 250 rodadas de particionamento dos dados (80% para treinamento e 20% para teste). Em cada rodada, o desempenho dos modelos foi avaliado utilizando o MSE, e os resultados estatísticos (média, desvio-padrão, valor máximo e mínimo) foram consolidados em tabelas para comparação.

2. Tarefa de Classificação (Bidimensional)

Objetivo: Resolver um problema de classificação binária utilizando o conjunto “Spiral3d.csv” ($p = 3$, $N = 2000$; os rótulos estão na 4ª coluna).

Procedimentos:

- **Organização dos Dados e Visualização:** Os dados foram organizados extraindo os preditores e os rótulos, com as conversões necessárias. Foi gerado um gráfico de dispersão (utilizando duas das três dimensões dos preditores) para visualizar a distribuição das classes.
- **Implementação dos Modelos:** Foram implementados os seguintes modelos:
 - **Perceptron Simples:** Implementado para atualizar os pesos de forma iterativa, onde cada amostra incorretamente classificada desencadeia uma atualização. O treinamento encerra quando nenhuma atualização é necessária em uma época.
 - **MLP:** Configurado de forma semelhante à tarefa de regressão, mas com a etapa de teste adaptada para classificação, onde a classe é definida pelo índice do valor máximo na saída do modelo.
- **Validação via Monte Carlo:** Utilizando 250 rodadas de Monte Carlo (80% treino / 20% teste), foram calculadas as métricas de acurácia, sensibilidade e especificidade. Para as rodadas com

os melhores e piores desempenhos, as matrizes de confusão foram construídas (plotadas com Seaborn) e as curvas de aprendizado foram geradas. Os resultados estatísticos finais foram apresentados em tabelas e gráficos.

II. Segunda Etapa – Classificação Multiclasse para Problema Multidimensional

Objetivo: Identificar condições clínicas da coluna vertebral a partir do conjunto “coluna_vertebral.csv” ($p = 6$, $N = 310$). As classes a serem reconhecidas são: Normal (NO), Hérnia de Disco (DH) e Espondilolistese (SL).

Procedimentos:

- **Organização dos Dados:** Os preditores foram organizados em uma matriz X , e com a inclusão do vetor de bias, a dimensão passou a ser $(p+1) \times N$. Os rótulos foram convertidos via one-hot encoding, conforme a seguinte regra: Normal: $[+1, -1, -1]$ Hérnia de Disco: $[-1, +1, -1]$ Espondilolistese: $[-1, -1, +1]$ Assim, o conjunto Y possui dimensão $C \times N$.
- **Implementação dos Modelos:** Nesta etapa, foram utilizados dois modelos:
 - **ADALINE:** Adaptado para problemas de classificação multiclasse, o ADALINE atualiza os pesos conforme o mesmo princípio utilizado na tarefa de regressão, com a decisão final baseada no índice do maior valor da saída.
 - **MLP:** O MLP foi configurado para lidar com a classificação multiclasse, com a definição adequada de camadas ocultas e funções de ativação. A fase de teste utiliza o argmax da saída para determinar a classe.
- **Validação via Monte Carlo:** Foram realizadas 100 rodadas de particionamento (80% treino / 20% teste). As métricas avaliadas foram acurácia, sensibilidade e especificidade. Para as rodadas com os melhores e piores desempenhos, matrizes de confusão foram geradas (por meio do Seaborn) e as curvas de aprendizado foram plotadas. Em seguida, os resultados estatísticos (média, desvio-padrão, valor máximo e mínimo) foram compilados em uma tabela.

Observações importantes sobre a metodologia aplicada:

- **Normalização:** Em todas as etapas, os dados foram normalizados ou padronizados para garantir a estabilidade numérica dos modelos.
- **Critério de Parada:** Todos os algoritmos implementados utilizam um critério de parada baseado tanto no número máximo de épocas quanto na tolerância do erro (EQM ou erro de classificação).
- **Hiperparâmetros:** Parâmetros como taxa de aprendizagem (η), número máximo de épocas, número de camadas ocultas, quantidade de neurônios por camada e critério de parada (ϵ) foram ajustados com base em experimentos preliminares e na análise das curvas de aprendizado.

Algoritmos Utilizados

Algoritmo ADALINE (Treinamento) – Algorithm 1:

```
1: Definir o valor de  $\eta$ , número máximo de épocas e precisão ( $\epsilon$ ).
2: Inicializar o vetor de pesos  $w(t)$  com valores nulos ou aleatórios.
3: Iniciar o contador de épocas ( $epoch \leftarrow 0$ )
4: repeat
5:    $EQM\_anterior \leftarrow EQM(x, d, w)$ 
6:   for todas as  $N$  amostras de treinamento do
7:      $u(t) \leftarrow w^T(t)x(t)$ 
8:      $w(t+1) \leftarrow w(t) + \eta(d(t) - u(t))x(t)$ 
9:   end for
10:   $epoch \leftarrow epoch + 1$ 
11:   $EQM\_atual \leftarrow EQM(x, d, w)$ 
12: until  $|EQM\_atual - EQM\_anterior| \leq \epsilon$  OU Número máximo de épocas atingido
```

Algoritmo ADALINE (Cálculo do EQM) – Algorithm 2:

```
1:  $EQM \leftarrow 0$ 
2: for todas as amostras de treinamento do
3:    $u(t) \leftarrow w^T(t)x(t)$ 
4:    $EQM \leftarrow EQM + (d(t) - u(t))^2$ 
5: end for
6:  $EQM \leftarrow EQM / (2N)$ 
```

Algoritmo ADALINE (Operação – Fase de Teste) – Algorithm 3:

```
1: Obter uma amostra ( $x\_desconhecido$ ) a ser classificada
2: Utilizar o vetor  $w$  já estimado
3: Realizar as seguintes operações:
4:    $u \leftarrow w^T x\_desconhecido$ 
5:    $y(t) \leftarrow \text{signal}(u(t))$ 
6:   if  $y == -1$  then
7:     amostra pertence a classe A
8:   else
9:     amostra pertence a classe B
10:  end if
```

Algoritmo MLP – Configuração (Algorithm 4):

```
1: Definir a quantidade  $L$  de camadas escondidas.
2: Definir a quantidade de neurônios em cada uma das  $L$  camadas escondidas:  $[q_1, q_2, q_3, \dots, q_L]$ .
3: Definir a quantidade de neurônios  $m$  na camada de saída.
4: Definir o valor da taxa de aprendizagem  $\eta$ .
5: Definir a quantidade máxima de épocas  $maxEpoch$ .
6: Definir o critério de parada em função do erro (EQM).
7: Criar uma lista (list) dos elementos:  $W, i, y, \delta$  cada uma com  $L+1$  posições.
8: Inicializar as  $L+1$  matrizes  $W$  com valores aleatórios pequenos  $(-0.5, 0.5)$ .
9: Receber os dados de treinamento com a ordem  $X\_treino \in \mathbb{R}^{(p \times N)}$  e os rótulos de treinamento com ordem  $Y\_treino \in \mathbb{R}^{(c \times N)}$ .
10: Adicionar o vetor linha de  $-1$  na primeira linha da matriz de dados  $X\_treino$ , resultando em  $X\_treino \in \mathbb{R}^{((p+1) \times N)}$ .
```

Algoritmo MLP – Treinamento (Algorithm 5):

```
1:  $EQM \leftarrow 1$ .
2:  $Epoch \leftarrow 0$ .
3: while  $EQM > CritérioParada$  &&  $Epoch < MaxEpoch$  do
4:   for cada amostra em  $X\_treino$  do
```

```
5:      $x\_amostra \leftarrow N$ -ésima amostra de  $X\_treino$ .
6:     Forward( $x\_amostra$ )
7:      $d \leftarrow N$ -ésimo rótulo de  $X\_treino$ .
8:     Backward( $x\_amostra, d$ )
9:   end for
10:   $EQM \leftarrow EQM()$ 
11:   $Epoch \leftarrow Epoch + 1$ .
12: end while
```

Algoritmo MLP – Forward (Algorithm 6):

```
1: Receber a amostra  $x\_amostra \in \mathbb{R}^{((p+1) \times 1)}$ .
2:  $j \leftarrow 0$ 
3: for cada matriz de peso  $W$  em cada uma das  $L+1$  camadas do
4:   if  $j == 0$  then
5:      $i[j] \leftarrow W[j] \cdot x\_amostra$ 
6:      $y[j] \leftarrow g(i[j])$ 
7:   else
8:      $y\_bias \leftarrow y[j-1]$  com adição de  $-1$  na primeira posição do vetor.
9:      $i[j] \leftarrow W[j] \cdot y\_bias$ 
10:     $y[j] \leftarrow g(i[j])$ 
11:   end if
12:    $j \leftarrow j + 1$ 
13: end for
```

Algoritmo MLP – Backward (Algorithm 7):

```
1: Receber a amostra  $x\_amostra \in \mathbb{R}^{((p+1) \times 1)}$  e seu rótulo  $d \in \mathbb{R}^{(c \times 1)}$ .
2:  $j \leftarrow$  Quantidade de matrizes  $W - 1$ .
3: while  $j \geq 0$  do
4:   if  $j + 1 ==$  Quantidade de matrizes  $W$ , then
5:      $\delta[j] \leftarrow g'(i[j]) \circ (d - y[j])$ 
6:      $y\_bias \leftarrow y[j-1]$  com adição de  $-1$  na primeira posição do vetor.
7:      $W[j] \leftarrow W[j] + \eta(\delta[j] \otimes y\_bias)$ 
8:   else if  $j == 0$  then
9:      $Wb[j+1]$  recebe a matriz  $W[j+1]$  transposta sem a coluna que multiplica pelos limiares de ativação.
10:     $\delta[j] \leftarrow g'(i[j]) \circ (Wb[j+1] \cdot \delta[j+1])$ 
11:     $W[j] \leftarrow W[j] + \eta(\delta[j] \otimes x\_amostra)$ 
12:   else
13:      $Wb[j+1]$  recebe a matriz  $W[j+1]$  transposta sem a coluna que multiplica pelos limiares de ativação.
14:     $\delta[j] \leftarrow g'(i[j]) \circ (Wb[j+1] \cdot \delta[j+1])$ 
15:     $y\_bias \leftarrow y[j-1]$  com adição de  $-1$  na primeira posição do vetor.
16:     $W[j] \leftarrow W[j] + \eta(\delta[j] \otimes y\_bias)$ 
17:   end if
18:    $j \leftarrow j - 1$ 
19: end while
```

Algoritmo MLP – Cálculo do EQM (Algorithm 8):

```
1:  $EQM \leftarrow 0$ 
2: for cada amostra em  $X\_treino$  do
3:    $x\_amostra \leftarrow N$ -ésima amostra de  $X\_treino$ .
4:   Forward( $x\_amostra$ )
5:    $d \leftarrow N$ -ésimo rótulo de  $X\_treino$ .
6:    $EQI \leftarrow 0$ 
7:    $j \leftarrow 0$ 
8:   for cada neurônio na camada de saída do
9:      $EQI \leftarrow EQI + (d[j] - y[QTD\_L - 1][j])^2$ 
10:     $j \leftarrow j + 1$ 
11:   end for
12:    $EQM \leftarrow EQM + EQI$ 
13: end for
```

14: $EQM \leftarrow EQM / (2 * QtdAmostrasTreino)$

Algoritmo MLP – Operação (Teste) (Algorithm 9):

```
1: for cada amostra em X_teste do
2:   x_amostra ← N-ésima amostra de X_teste.
3:   Forward(x_amostra)
4:   Realizar a atribuição desta amostra para a classe cujo
   índice do vetor de saída possuir maior valor.
5: end for
```

Algoritmo Perceptron Simples – Treinamento (Algorithm 10):

```
1: Início (t = 0)
2: Definir o valor de  $\eta$  entre 0 e 1.
3: Inicializar o vetor de pesos w(t) com valores nulos ou
   aleatórios.
4: ERRO ← 'EXISTE'
5: while ERRO == 'EXISTENTE' do
6:   ERRO ← 'INEXISTE'
7:   for todas as amostras em x do
8:      $u(t) \leftarrow w^T(t)x(t)$ 
9:      $y(t) \leftarrow \text{signal}(u(t))$ 
10:     $w(t+1) \leftarrow w(t) + \eta (d(t) - y(t)) x(t)$ 
11:    if  $d(t) \neq y(t)$  then
12:      ERRO ← 'EXISTENTE'
13:    end if
14:  end for
15:  t ← t + 1
16: end while
17: FIM TREINAMENTO
```

Algoritmo Perceptron Simples – Operação (Teste) (Algorithm 11):

```
1: Obter uma amostra (x_desconhecido) a ser classificada.
2: Utilizar o vetor w já estimado.
3: Realizar as seguintes operações:
4:    $u \leftarrow w^T x_{\text{desconhecido}}$ 
5:    $y(t) \leftarrow \text{signal}(u(t))$ 
6:   if  $y == -1$  then
7:     a amostra pertence à classe A
8:   else
9:     a amostra pertence à classe B
10:  end if
```

III. RESULTADOS E DISCUSSÕES

- Etapa I – Regressão e Classificação para problemas bidimensionais

Para a tarefa de regressão, os resultados apresentados na **Tabela 1** abaixo foram obtidos após a execução de 250 rodadas de simulações de Monte Carlo, onde, em cada iteração, os dados foram particionados aleatoriamente em 80% para treinamento e 20% para teste. As métricas foram avaliadas por meio do Mean Squared Error (MSE) para cada modelo.

Modelo	Média	Desvio Padrão	Maior Valor	Menor Valor
ADALINE	0,0764	0,0167	0,1277	0,0361
MLP	0,1681	0,0183	0,2172	0,1228

Tabela 1 - Tabela da tarefa de Regressão

Observa-se que o modelo ADALINE apresentou um desempenho superior em termos de ajuste, com um MSE médio significativamente menor (≈ 0.0764) quando comparado ao MLP, cujo MSE médio foi aproximadamente 0.1681. A menor variabilidade observada no ADALINE, conforme evidenciada pelo desvio-padrão (0.0167), indica maior consistência na performance do modelo ao longo das simulações. Em contrapartida, o MLP, apesar de demonstrar maior capacidade de modelar relações não lineares e, consequentemente, capturar padrões mais complexos, apresentou um erro médio mais elevado e uma amplitude maior dos valores de MSE, sugerindo uma maior sensibilidade às configurações da topologia e aos hiperparâmetros.

Além dos resultados numéricos, temos a **Figura 1** que apresenta o gráfico de dispersão dos dados de entrada, demonstrando a ampla variabilidade e a distribuição dos pontos que compõem o conjunto de dados utilizado na regressão. Por sua vez, a **Figura 2** exibe as curvas de aprendizado dos modelos MLP Subdimensionado e Superdimensionado, evidenciando que, na evolução do MSE ao longo das épocas, o Subdimensionado convergiu para um EQM mais alta.

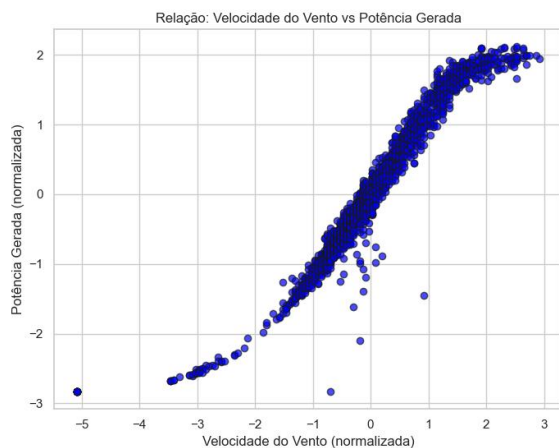


Figura 1 - Gráfico de dispersão

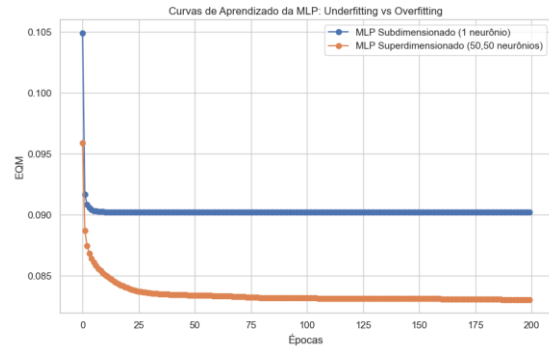


Figura 2 - Curvas de aprendizado do MLP Subdimensionado e Superdimensionado

Em síntese, os resultados obtidos via simulação Monte Carlo confirmam que, embora ambos os modelos tenham sido implementados “do zero”, diferenças significativas no desempenho evidenciam a importância da escolha adequada do modelo e da otimização dos parâmetros de treinamento. A análise combinada dos resultados numéricos, das curvas de aprendizado e da visualização gráfica dos dados proporciona uma avaliação robusta e detalhada dos métodos aplicados. Em contrapartida, o MLP, apesar de demonstrar maior capacidade de modelar relações não lineares e, consequentemente, potencialmente capturar padrões mais complexos nos dados, apresentou um erro médio maior e uma maior amplitude dos valores de MSE ao longo das simulações, evidenciando maior sensibilidade às configurações da topologia e aos hiperparâmetros.

Esses resultados ressaltam a importância da escolha do modelo e dos parâmetros para a tarefa de regressão: enquanto o ADALINE se mostrou robusto e eficiente para o ajuste do problema em questão, o MLP demanda uma afinação mais precisa para alcançar um desempenho comparável, especialmente em contextos onde relações não lineares são relevantes.

De maneira geral, os resultados da validação Monte Carlo confirmam que modelos implementados “do zero” podem apresentar variações significativas no desempenho dependendo da partição dos dados, mas a análise estatística (média, desvio padrão, e extremos) fornece uma avaliação robusta comparando os métodos propostos.

Na tarefa de classificação, foram avaliadas duas abordagens: Perceptron Simples e Perceptron Multicamadas (MLP). Os algoritmos foram testados por meio de 250 rodadas de simulações de Monte Carlo, onde os dados do conjunto “Spiral3d.csv” foram particionados aleatoriamente em 80% para treinamento e 20% para teste. As métricas de desempenho foram avaliadas com base em acurácia, sensibilidade e especificidade (**Tabelas 2 e Tabela 3**).

Os resultados quantitativos estão resumidos a seguir:

- Resumo das Métricas para o Perceptron Simples:

Métrica	Média	Desvio Padrão	Maior Valor	Menor Valor
ACURÁCIA	0,5775	0,0607	0,6875	0,3700
SENSIBILIDADE	0,5930	0,2602	0,9953	0,0000
ESPECIFICIDADE	0,5622	0,2602	1,000	0,0052

Tabela 2 - Resumo das Métricas para Perceptron Simples

- Resumo das Métricas para o MLP:

Métrica	Média	Desvio Padrão	Maior Valor	Menor Valor
ACURÁCIA	0,6819	0,0411	0,8250	0,5550
SENSIBILIDADE	0,7083	0,1453	0,9906	0,2042
ESPECIFICIDADE	0,6564	0,1291	0,9384	0,2780

Tabela 3 - Resumo das Métricas para o MLP

Observa-se que o Perceptron Simples obteve uma acurácia média de 57,75%, enquanto a abordagem MLP apresentou uma acurácia média superior, de 68,19%. Notavelmente, as melhores e piores rodadas foram: para o Perceptron Simples, a melhor rodada (rodada 12) alcançou 68,75% de acurácia e a pior (rodada 211) 37,00%; para o MLP, a melhor rodada (rodada 128) atingiu 82,50% de acurácia, em contraposição à pior (rodada 144), com 55,50%.

Além das estatísticas, como informação complementar, foram gerados gráficos que ilustram o comportamento dos modelos durante o treinamento e a fase de teste:

A **Figura 3**, por exemplo, apresenta o gráfico de dispersão 2D dos dados do “Spiral3d.csv”, onde os pontos azuis (classe -1) e vermelhos (classe +1) formam dois anéis concêntricos intercalados. Essa estrutura deixa claro que as classes não são separáveis por um único hiperplano linear, o que já antecipa a limitação do Perceptron Simples e a maior capacidade de modelagem do MLP.

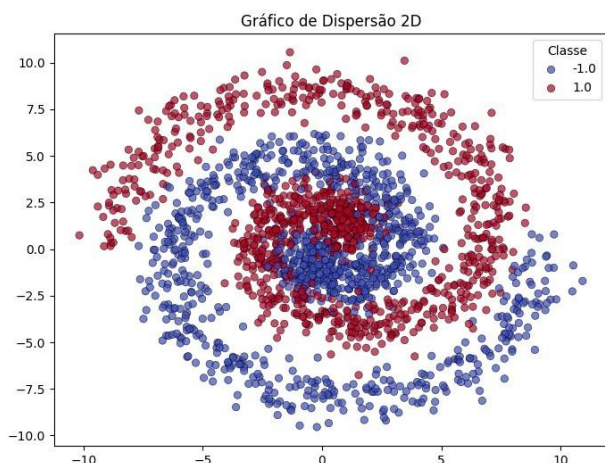


Figura 3 - Gráfico de Dispersão 2D

Na **Figura 4**, observa-se a curva de aprendizado do Perceptron referente à melhor execução (rodada 12). O erro inicial próximo a 720 decresce até cerca de 660, mas mantém oscilações significativas após as primeiras épocas, refletindo a dificuldade do modelo em convergir para uma fronteira linear numa tarefa não linear. A acurácia máxima registrada nessa execução foi de 0,6875.

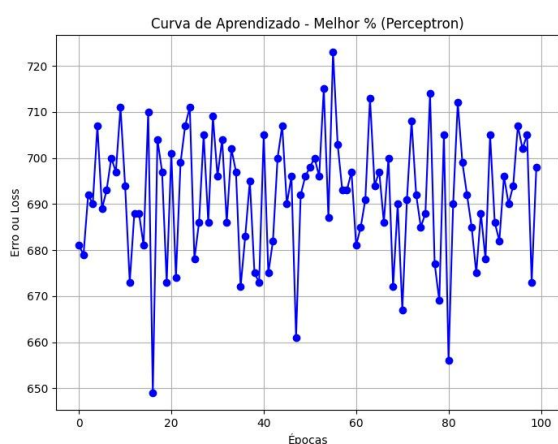


Figura 4 - Curva de aprendizado do Perceptron Simples (melhor rodada)

A **Figura 5** exibe a matriz de confusão do Perceptron na melhor rodada. Destacam-se 134 verdadeiros negativos e 141 verdadeiros positivos, com 75 e 50 classificações incorretas de cada tipo, respectivamente. Isso se traduz em uma sensibilidade de aproximadamente 0,738 e especificidade de 0,641, mostrando que mesmo no melhor caso o Perceptron tende a favorecer o reconhecimento de uma classe em detrimento da outra.

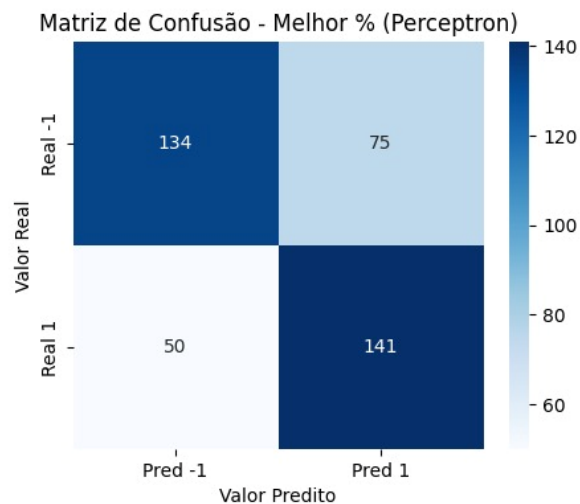


Figura 5 - Matriz de Confusão do Perceptron Simples (melhor rodada)

Em contraste, a **Figura 6** mostra a curva de aprendizado do Perceptron na pior execução (rodada 211). Aqui o erro varia sem padrão claro entre 670 e 715, evidenciando completa falta de adaptação do modelo à fronteira espiral. A acurácia mínima alcançada foi de apenas 0,3700.

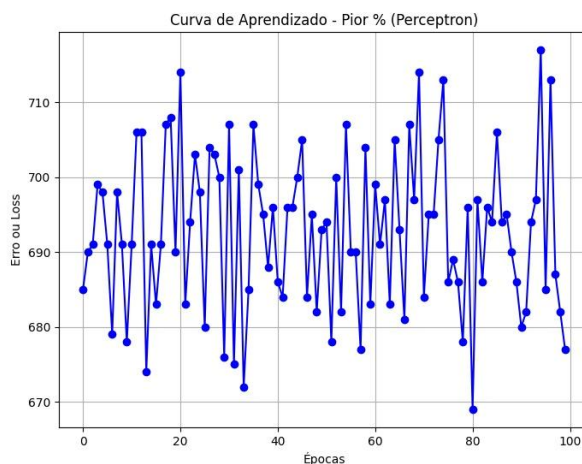


Figura 6 - Curva de Aprendizado do Perceptron Simples (pior rodada)

A **Figura 7** traz a matriz de confusão do Perceptron na pior rodada. Com 77 verdadeiros negativos e 71 verdadeiros positivos, somados a 107 falsos positivos e 145 falsos negativos, o modelo praticamente classifica tudo como -1 (especificidade $\approx 0,418$ e sensibilidade $\approx 0,329$), comprovando seu desempenho insatisfatório nesse cenário.

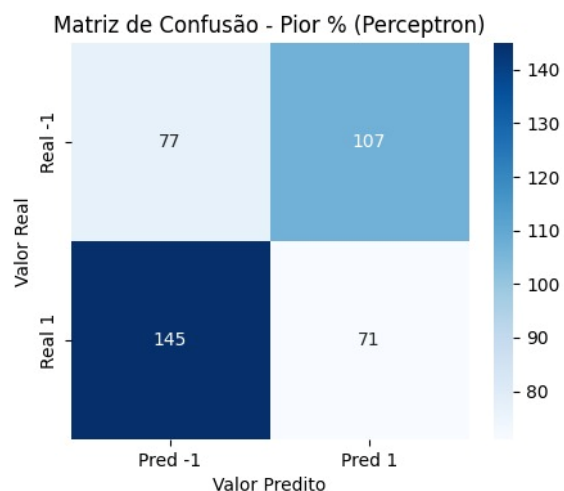


Figura 7 - Matriz de Confusão do Perceptron Simples (pior rodada)

Já a **Figura 8** exibe a curva de aprendizado do MLP na melhor execução (rodada 128). Aqui o erro decai de aproximadamente 0,67 para valores em torno de 0,40 ao

longo de 200 épocas, com poucos picos de oscilação. Esse comportamento revela a capacidade do MLP de ajustar a fronteira de decisão não linear e estabilizar o aprendizado, culminando em acurácia máxima de 0,8250.

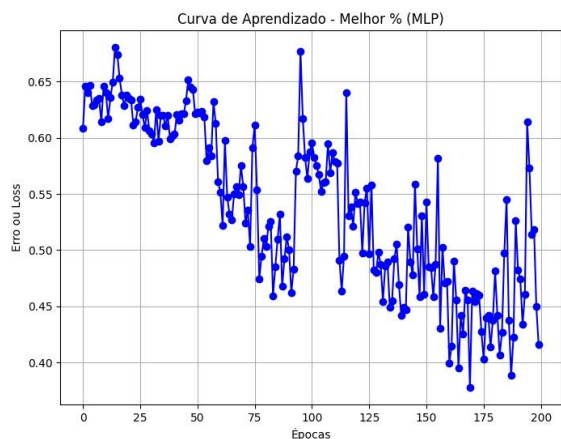


Figura 8 - Curva de Aprendizado da MLP (melhor rodada)

A **Figura 9** apresenta a matriz de confusão do MLP na melhor rodada. Com 160 verdadeiros negativos e 170 verdadeiros positivos, contra apenas 34 falsos positivos e 36 falsos negativos, o modelo atinge sensibilidade e especificidade equilibradas em $\approx 0,825$, evidenciando classe-balance e alto poder discriminativo.

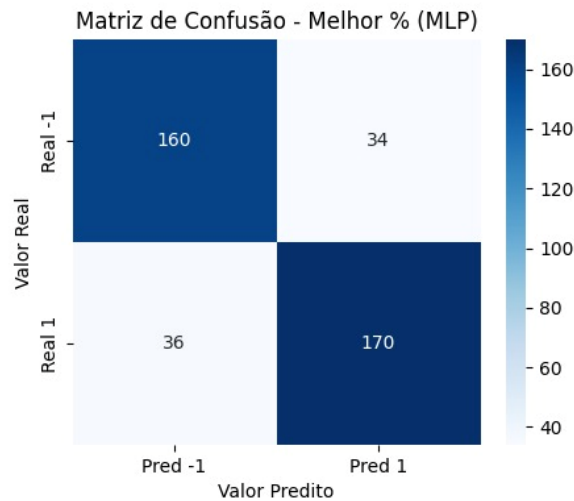


Figura 9 - Matriz de Confusão da MLP (melhor rodada)

A **Figura 10** ilustra a curva de aprendizado do MLP na pior execução (rodada 144). Apesar de reduzir o erro de cerca de 0,62 para a faixa de 0,56–0,57, o modelo apresenta menor refinamento na fronteira e menor estabilidade, explicando sua acurácia mínima de 0,5550.

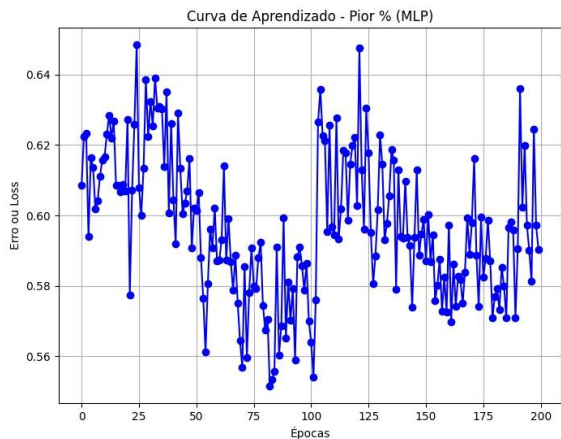


Figura 10 - Curva de Aprendizado da MLP (pior rodada)

Por fim, a **Figura 11** mostra a matriz de confusão do MLP na pior rodada, onde aparecem 183 verdadeiros negativos e apenas 39 verdadeiros positivos, com 26 falsos

positivos e 152 falsos negativos. Esses resultados indicam alto viés para a classe -1 (especificidade $\approx 0,875$ vs. sensibilidade $\approx 0,204$), sinalizando que ajustes adicionais de hiperparâmetros ou topologia seriam necessários para recuperar o desempenho completo do MLP.

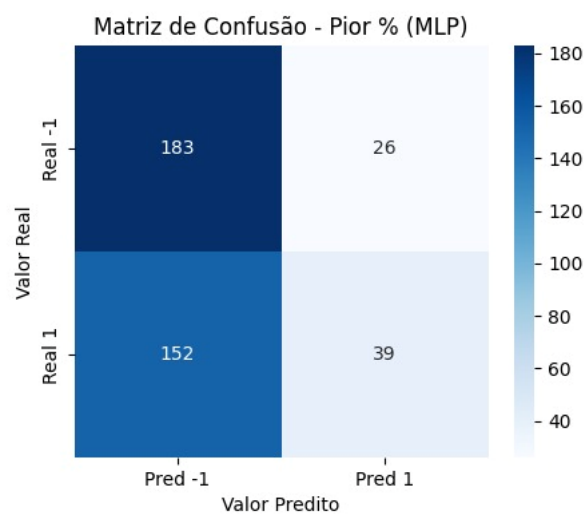


Figura 11- Matriz de Confusão da MLP (pior rodada)

Em síntese, os resultados da validação por Monte Carlo confirmam que, para o problema de classificação abordado, a abordagem MLP tende a superar o Perceptron Simples, refletindo a sua capacidade aprimorada de capturar relações não lineares e melhorar a separação das classes. A combinação dos dados estatísticos, das curvas de aprendizado e das matrizes de confusão oferece uma avaliação robusta dos métodos propostos, destacando a importância da escolha do modelo e da calibração dos hiperparâmetros para alcançar um desempenho consistente.

• Etapa II - Classificação Multiclasse para Problema Mudimensional

Os resultados para a tarefa de classificação multiclasse, utilizando o conjunto “coluna vertebral.csv”, foram obtidos por meio de 100 simulações de Monte Carlo, com a divisão de 80% dos dados para treinamento e 20% para teste. A Tabela abaixo resume as principais métricas para os modelos ADALINE e MLP, considerando acurácia, sensibilidade e especificidade (**Tabelas 4 e Tabela 5**):

○ Resumo das métricas para modelo ADALINE:

Métrica (ADALINE)	Média	Desvio Padrão	Maior Valor	Menor Valor
ACURÁCIA	0,6819	0,0411	0,8250	0,5550
SENSIBILIDADE	0,7083	0,1453	0,9906	0,2042
ESPECIFICIDADE	0,6564	0,1291	0,9384	0,2780

Tabela 4 - Resumo das Métricas para ADALINE

○ Resumo das métricas para modelo MLP:

Métrica (ADALINE)	Média	Desvio Padrão	Maior Valor	Menor Valor
ACURÁCIA	0,6819	0,0411	0,8250	0,5550
SENSIBILIDADE	0,7083	0,1453	0,9906	0,2042
ESPECIFICIDADE	0,6564	0,1291	0,9384	0,2780

Tabela 5 - Resumo das métricas para modelo MLP

Adicionalmente, a análise individual das rodadas revelou:

Para o ADALINE, a melhor rodada apresentou acurácia de 0,9516, enquanto a pior atingiu 0,6774. Para o MLP, a melhor rodada registrou acurácia de 0,9355 e a pior, 0,7097.

Para investigar o impacto da complexidade do modelo,

também foram avaliadas configurações alternativas do MLP:

- Underfitting MLP: Obtendo acurácia de 0.7258, sensibilidade de 0.6377 e especificidade de 0.8635, essa configuração demonstra que uma rede com baixa complexidade falha em capturar os padrões relevantes dos dados.
- Overfitting MLP: Apresentando acurácia de 0.8387, sensibilidade de 0.8364 e especificidade de 0.9258, essa topologia indicou um ajuste muito próximo aos dados de treinamento, potencialmente comprometendo a generalização, embora resulte em melhores métricas para a amostra de teste.

Em síntese, os resultados demonstram que ambos os modelos – ADALINE e MLP – obtiveram desempenho satisfatório para o problema de classificação multiclasse. O MLP, com média de acurácia de 0.8258, ligeiramente supera o ADALINE (média de 0.8161), além de apresentar incrementos modestos nas medidas de sensibilidade e especificidade. Todavia, as variações entre as rodadas ressaltam que a escolha dos hiperparâmetros e da topologia da rede tem papel crucial na performance final. Enquanto o ADALINE mostrou maior estabilidade em termos de especificidade, o MLP demonstrou uma capacidade aprimorada de discriminar as classes, especialmente quando configurado com uma complexidade que balanceia adequadamente underfitting e overfitting. Esses achados reforçam a importância de uma calibração criteriosa dos modelos para maximizar a generalização em problemas de classificação multiclasse.

Na **Figura 12**, o gráfico de dispersão bidimensional das amostras mostra a distribuição das três classes (NO, DH e SL) em relação às variáveis “Incidência Pélvica” e “Inclinação Pélvica”. Observa-se que, embora exista algum grau de sobreposição entre as classes, a classe SL tende a ocupar regiões de valores mais elevados em ambas as variáveis, enquanto NO e DH ficam mais concentradas em torno da origem e em regiões negativas. Essa análise preliminar justifica a aplicabilidade de métodos lineares (como o ADALINE) e não lineares (como o MLP) para tentar separar as categorias.

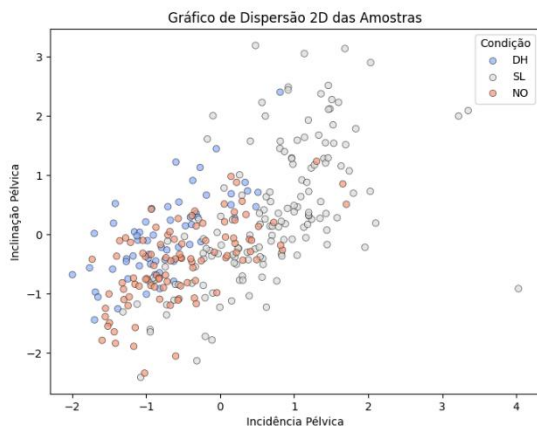


Figura 12 - Gráfico de dispersão 2D das amostras

Os resultados agregados para o ADALINE apresentaram média de acurácia de 0,8161 ($\pm 0,0516$), sensibilidade de 0,7665 ($\pm 0,0620$) e especificidade de 0,9048 ($\pm 0,0281$). A **Figura 13** mostra a curva de erro quadrático médio para a melhor iteração do ADALINE: percebe-se uma convergência rápida nos primeiros 10 épocas, com estabilização do erro ao redor de 0,65 após cerca de 20 épocas. A **Figura 14** detalha a matriz de confusão correspondente à melhor rodada do ADALINE, revelando que a classe SL foi perfeitamente discriminada, enquanto houve algumas confusões entre NO e DH.

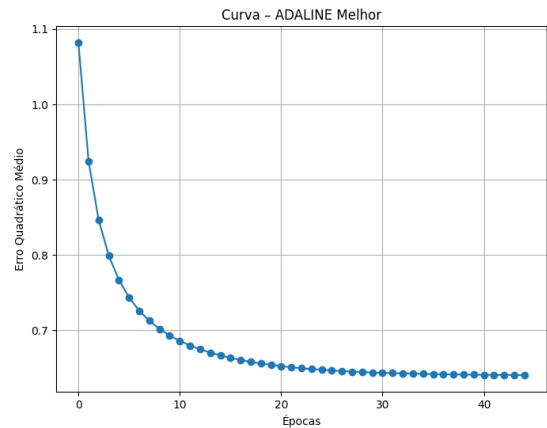


Figura 13 - Curva de Aprendizado ADALINE (melhor rodada)

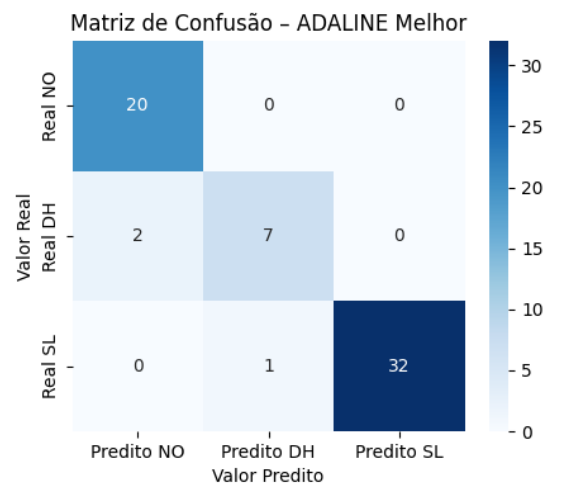


Figura 14 - Matriz de Confusão do modelo ADALINE (melhor rodada)

Para a pior rodada do ADALINE, foi observada uma acurácia de 0,6774 (**Figura 15**). A matriz de confusão indicou 8 amostras de NO classificadas como DH e 5 amostras de DH classificadas como NO, reforçando a dificuldade do modelo linear em separar essas duas classes quando os dados de entrada se sobrepõem mais intensamente. Isso demonstra que o ADALINE é sensível à variabilidade nos particionamentos dos dados, assim como evidenciado na **Figura 16**, da curva de aprendizagem para a pior rodada.

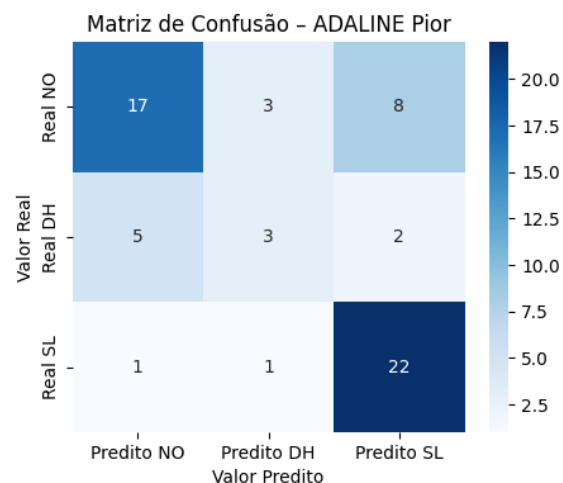


Figura 15 - Matriz de Confusão ADALINE (pior rodada)

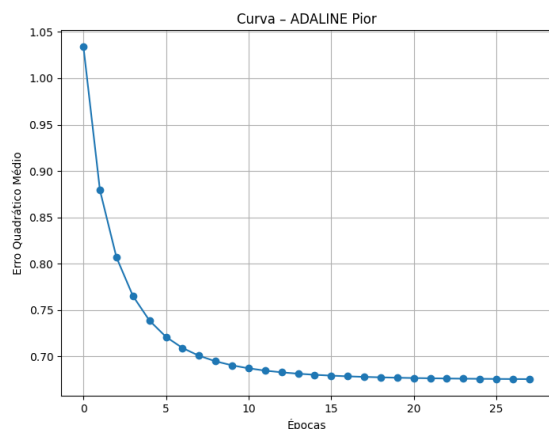


Figura 16 - Curva de Aprendizagem ADALINE (pior caso)

O MLP, em sua melhor configuração encontrada, alcançou média de acurácia de 0,8258 ($\pm 0,0446$), sensibilidade de 0,7845 ($\pm 0,0549$) e especificidade de 0,9172 ($\pm 0,0229$). A **Figura 17** ilustra a matriz de confusão do MLP na melhor rodada, na qual foram identificados apenas dois falsos negativos para SL e nenhum falso positivo para DH, indicando excelente capacidade de discriminação entre as três classes. Na **Figura 18**, a curva de aprendizado do MLP melhor demonstra um comportamento de queda consistente do erro quadrático médio até valores próximos de 0,30 após 300 épocas, embora com pequenas oscilações devido ao gradiente estocástico.

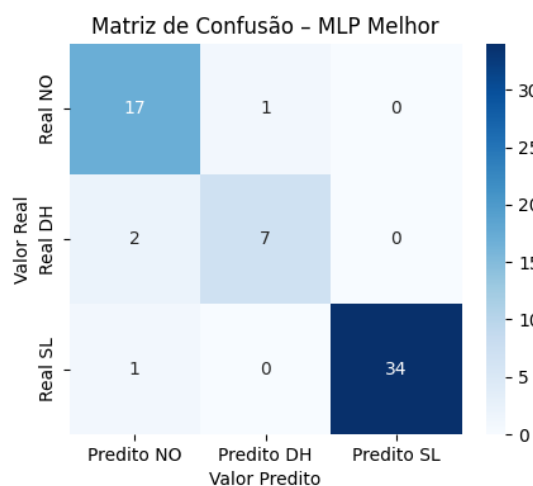


Figura 17 - Matriz de confusão do MLP (melhor rodada)

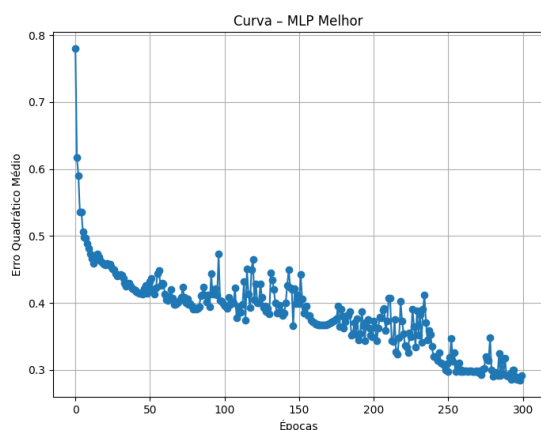


Figura 18 - Curva de Aprendizagem do MLP (melhor rodada)

Para a pior rodada do MLP, evidenciada pela **Figura 19** (matriz de confusão) e **Figura 20** (curva de aprendizado), o modelo apresentou acurácia de 0,7097. Esse valor foi obtido diretamente a partir da matriz de confusão, onde foram corretamente classificadas 44 amostras de um total de 62 ($44/62 \approx 0,7097$). A matriz de confusão mostra 6 amostras de

NO classificadas erroneamente como DH e 9 amostras de DH classificadas como NO, indicando dificuldade em separar essas duas classes nesse particionamento mais desafiador. A curva de aprendizado revela que o erro se estabilizou em torno de 0,35, sem convergir tão eficientemente quanto no melhor caso, sugerindo sensibilidade à inicialização dos pesos e à ordem de apresentação dos dados.

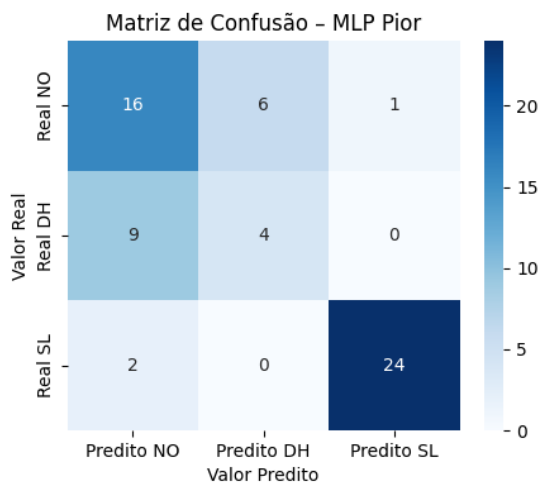


Figura 19 - Matriz de Confusão (pior rodada)

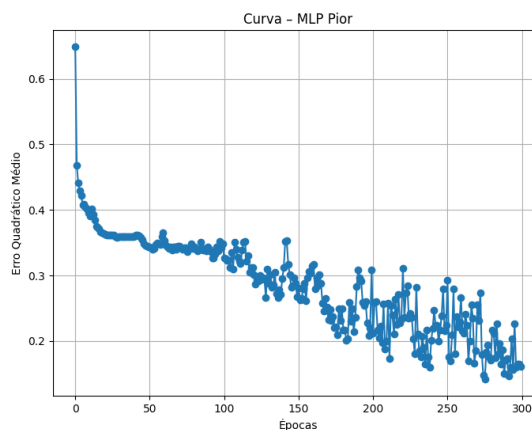


Figura 20 - Curva de Aprendizagem (pior rodada)

Para avaliar os extremos de complexidade, testaram-se duas configurações adicionais do MLP:

- **Underfitting**: rede com pouca capacidade (menos camadas/neurônios) apresentou acurácia de 0,7258, sensibilidade de 0,6377 e especificidade de 0,8635. A **Figura 21** (matriz de confusão do caso underfitting) evidencia muitas amostras da classe DH sendo classificadas como NO, indicando que o modelo não capturou os padrões mais sutis entre as classes e a **Figura 22** confirma o comportamento.
- **Overfitting (Figuras 23 e 24)**: rede excessivamente complexa atingiu acurácia de 0,8387, sensibilidade de 0,8364 e especificidade de 0,9258. Ainda que tenha melhorado as métricas na amostra de teste, há indícios de ajuste muito próximo ao conjunto de treinamento, o que pode comprometer a generalização em dados não vistos.

Em resumo, tanto o ADALINE quanto o MLP forneceram desempenho satisfatório na classificação multiclasse, com vantagem ligeira para o MLP em média de acurácia e em todas as demais métricas. O ADALINE, por ser um modelo linear, mostrou maior estabilidade na especificidade, enquanto o MLP, quando calibrado adequadamente, revelou maior sensibilidade e capacidade de discriminar a classe SL. As variações de resultado entre as rodadas destacam a importância da escolha criteriosa de hiperparâmetros e topologia da rede. Modelos subdimensionados (underfitting) falham em aprender padrões relevantes, ao passo que modelos excessivamente complexos (overfitting) podem comprometer a robustez em cenários reais. Portanto, para maximizar a generalização, recomenda-se empregar estratégias de validação e regularização, além de um cuidadoso balanceamento entre capacidade de representação e simplicidade do modelo.

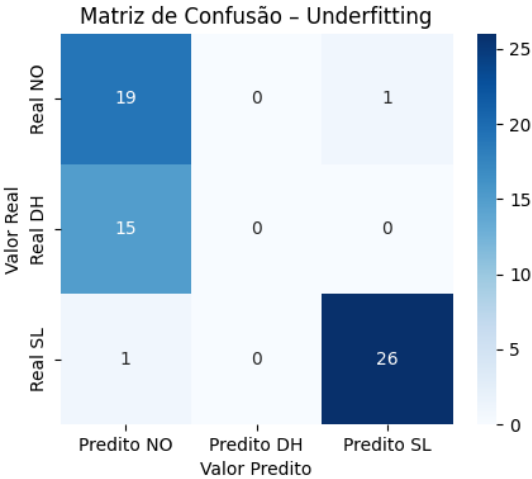


Figura 21 - Matriz de Confusão Underfitting

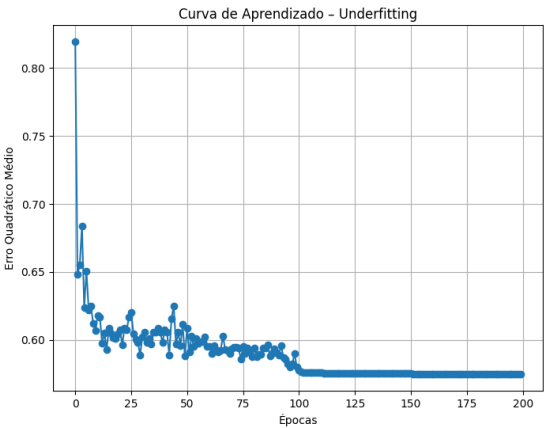


Figura 22 - Curva de Aprendizado Underfitting

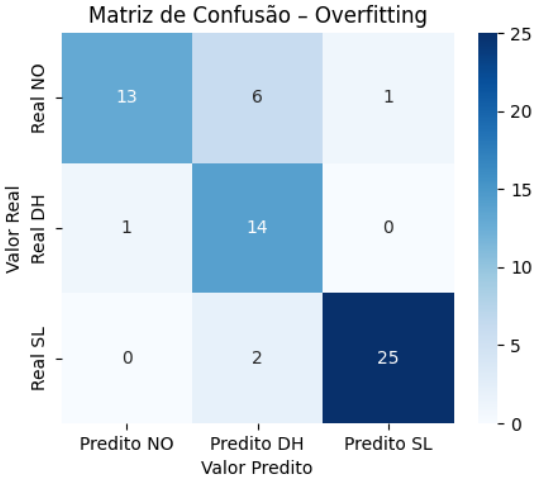


Figura 23- Matriz de Confusão - Overfitting

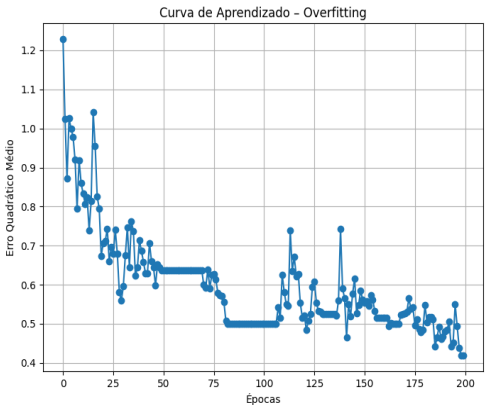


Figura 24 - Curva de Aprendizado - Overfitting

IV. CONCLUSÃO

O presente trabalho demonstrou, de forma prática e robusta, a viabilidade e a eficácia dos modelos de redes neurais artificiais implementados “do zero” em duas tarefas distintas.

Na primeira etapa, voltada a problemas bidimensionais, o ADALINE obteve erro médio (MSE) de 0,0764 ($\pm 0,0167$), variando entre 0,0361 e 0,1277, enquanto o MLP apresentou MSE médio de 0,1681 ($\pm 0,0183$), com extremos 0,1228 e 0,2172. Na tarefa de classificação binária (Spiral3d), o Perceptron Simples alcançou acurácia média de 57,8% ($\pm 6,1\%$), sensibilidade média de 59,3% e especificidade de 56,2% (variações amplas, 0–100%), com melhor rodada em 68,8% e pior em 37,0%. Já o MLP de classificação bidimensional atingiu acurácia média de 68,2% ($\pm 4,1\%$), sensibilidade de 78,8% e especificidade de 65,6%, variando entre 55,5% e 82,6%.

Na Segunda Etapa, focada no diagnóstico das condições da coluna vertebral, o ADALINE registrou acurácia média de 81,6% ($\pm 5,2\%$), sensibilidade de 76,6% e especificidade de 90,5%, com extremos de 67,7% a 95,2%. O MLP superou ligeiramente, com 82,6% de acurácia média ($\pm 4,5\%$), sensibilidade de 78,5% e especificidade de 91,7%, variando entre 70,9% e 93,6%. Os experimentos de underfitting (MLP subdimensionada: 1 neurônio) e overfitting (MLP superdimensionada: 50-50 neurônios) evidenciaram trade-offs claros: topologias muito simples caíram para 72,6% de acurácia, enquanto redes maiores chegaram a 83,9%, mas com maior risco de ajuste excessivo.

Em suma, todos os modelos se mostraram capazes de aprender tanto regressões simples quanto classificações mais complexas, e a validação por Monte Carlo provou-se essencial para estimar a robustez e variabilidade de desempenho. A escolha cuidadosa de topologias e hiperparâmetros — aliada à normalização dos dados e ao critério de convergência — garantiu equilíbrio entre precisão e generalização em todas as etapas do trabalho.