Projeto 1 - Rede MLP

Guilherme Pereira Campos RA:163787
Lucas Oliveira Nery de Araújo RA:158882
Universidade Federal de São Paulo

I. Resumo

Este projeto tem como objetivo a implementação de uma rede neural do tipo MLP (Multi-Layer Perceptron) em Python, sem recorrer a bibliotecas específicas de aprendizado de máquina, como PyTorch ou TensorFlow. A partir dessa implementação, foram desenvolvidos modelos para resolver problemas de classificação e regressão. Esses modelos foram avaliados com base em diferentes configurações de hiperparâmetros, incluindo o número de camadas, neurônios e taxas de aprendizado (eta e momentum). O trabalho também explora o impacto dessas variações no desempenho da rede.

II. Implementação da rede

A implementação da rede neural MLP foi dada por meio de funções de ativação como a sigmoid e a softmax. A função sigmoid tem a derivada

$$f'(x) = f(x) \cdot (1 - f(x)),$$

enquanto a softmax converte as saídas em uma distribuição de probabilidade

$$f(y) = \frac{\exp(y)}{\sum_k \exp(y_k)}.$$

A retropropagação, essencial para redes profundas, permite o ajuste dos pesos das camadas ocultas, que não têm erro diretamente observável. O cálculo do gradiente local, essencial para a atualização dos pesos, depende da função de ativação de cada neurônio. A fórmula de atualização dos pesos é dada por

$$\Delta w_{kj} = \eta \cdot \delta_k \cdot x_j.$$

A retropropagação considera dois casos: para neurônios de saída

$$\delta_k = e_k \cdot f'(v_k),$$

e para neurônios ocultos

$$\delta_j = f'(v_j) \cdot \sum_k \delta_k \cdot w_{kj}.$$

A função de erro é o erro quadrático médio, e o algoritmo de treinamento é dividido entre as fases de feedforward (propagação dos dados) e feedback (ajuste dos pesos).

III. Modelos

Observação: Os modelos foram avaliados no Dataset Iris, que contém 150 amostras de flores de três espécies (Iris setosa, Iris versicolor e Iris virginica), com 4 características medidas.

A. Primeiro Modelo de classificação

O primeiro modelo consiste em uma rede neural MLP com 3 camadas: a camada de entrada possui 4 neurônios, a camada oculta tem 2 neurônios e a camada de saída possui 3 neurônios. A taxa de aprendizado foi definida como 10, e o momentum configurado em 0.3. O modelo foi treinado por 50 épocas e teve uma acurácia de aproximadamente 0.56.
A taxa de aprendizado muito alta (10), causa oscilações no treinamento. A escolha de uma camada oculta com apenas 2 neurônios também limita a capacidade de aprendizado da rede. Com um número insuficiente de neurônios na camada oculta e a taxa de aprendizado elevada, o modelo não aprende de forma eficaz, resultando em uma acurácia baixa.

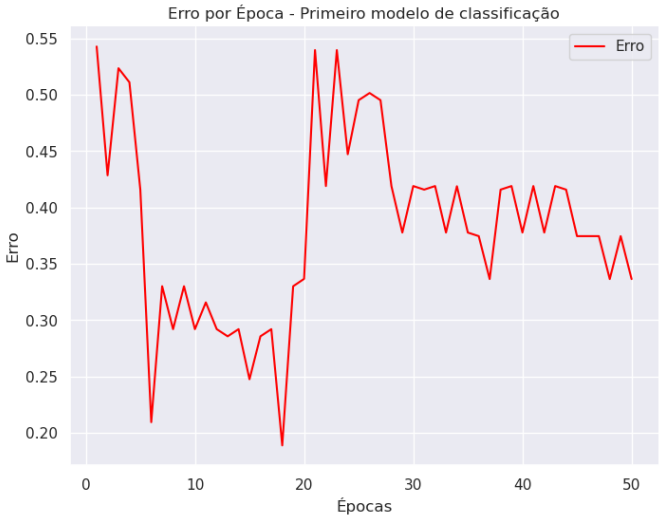

Fig. 1. Primeiro modelo de classificação

B. Segundo Modelo de classificação

O segundo modelo também é uma rede neural MLP com 3 camadas: 4 neurônios na camada de entrada, 2 na camada oculta e 3 na camada de saída. A taxa de aprendizado foi ajustada para 0.8 e o momentum para 0.6. O treinamento foi realizado por 50 épocas e teve uma acurácia de aproximadamente 0.75. O segundo modelo melhorou, alcançando uma acurácia média, graças à taxa de aprendizado de 0.8 e momentum de 0.6, que ajudaram no ajuste dos pesos. No entanto, a configuração ainda não é ideal, pois a taxa de aprendizado pode causar oscilações, e o número de neurônios na camada oculta (2) é baixo, o que limita o desempenho

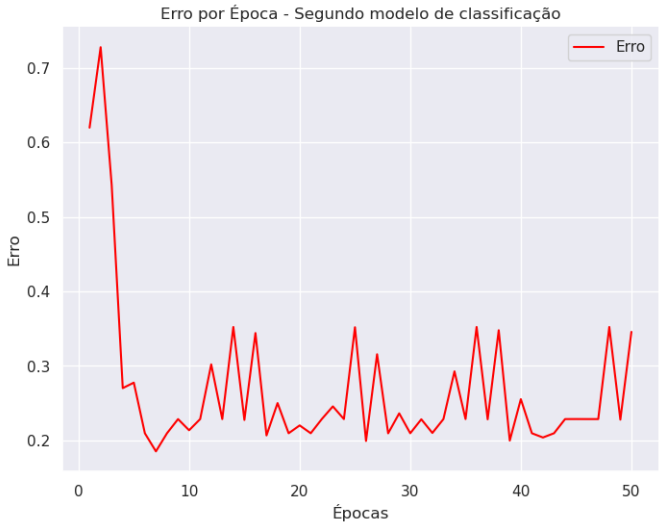

Fig. 2. Segundo modelo de classificação

C. Terceiro Modelo de classificação

O terceiro modelo foi configurado com 4 neurônios na camada de entrada, 10 na camada oculta e 3 na camada de saída. A taxa de aprendizado foi ajustada para 0.05 e o momentum para 0.9. O modelo foi treinado por 200 épocas e teve uma acurácia de aproximadamente 0.96.
Esse modelo teve sua taxa de aprendizado reduzida para 0.05 e o momentum aumentou para 0.9, o que ajudou a estabilizar o treinamento. A camada oculta com 10 neurônios permitiu um melhor aprendizado. Essas mudanças resultaram em uma melhora significativa da acurácia

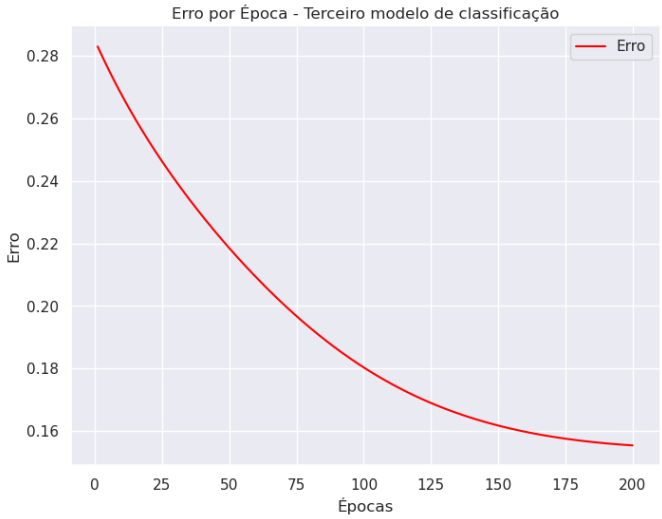

Fig. 3. Terceiro modelo de classificação

Observação: O dataset utilizado para a tarefa de regressão foi criado utilizando a biblioteca `numpy`, simulando a relação $y = x$, com valores de X variando entre 0 e 30. Foi adicionado um ruído aleatório aos dados para introduzir variações.

D. Primeiro modelo de regressão

O primeiro modelo de regressão foi configurado com uma rede neural MLP composta por 3 camadas: a camada de entrada possui 1 neurônio, a camada oculta tem 3 neurônios e a camada de saída possui 1 neurônio. A taxa de aprendizado foi definida como 0.001 e o momentum foi configurado em 0.1. O modelo foi treinado por 50 épocas.
O modelo possui um configuração com apenas 3 neurônios na camada oculta, o que limita sua capacidade de capturar padrões nos dados. Além disso, a taxa de aprendizado muito baixa (0.001) torna o treinamento lento e dificulta um bom resultado em apenas 50 épocas

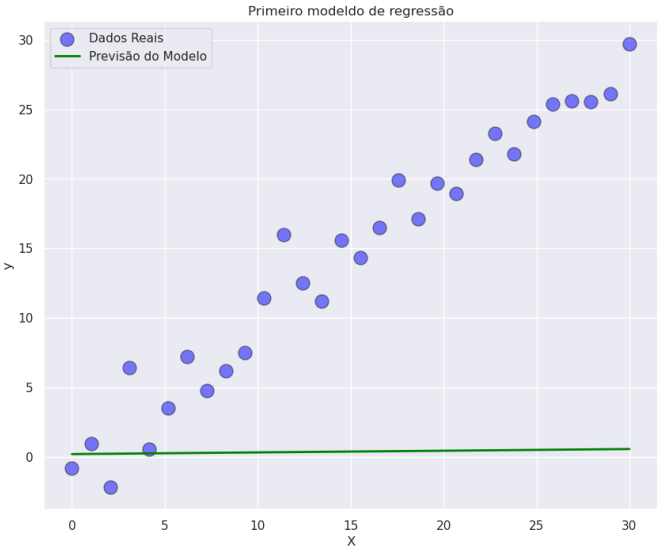

Fig. 4. Primeiro modelo de regressão

E. Segundo modelo de regressão

O segundo modelo de regressão foi configurado com uma rede neural MLP composta por 3 camadas: a camada de entrada possui 1 neurônio, a camada oculta tem 10 neurônios e a camada de saída possui 1 neurônio. A taxa de aprendizado foi definida como 0.05 e o momentum foi configurado em 0.5. O modelo foi treinado por 1000 épocas.
O segundo modelo de regressão tem um desempenho médio. Ele utiliza 10 neurônios na camada oculta, o que aumenta sua capacidade de aprendizado em comparação ao primeiro modelo. A taxa de aprendizado moderada (0.05) e o momentum de 0.5 ajudam a ajustar os pesos de forma mais estável, enquanto as 1000 épocas garantem mais iterações para o aprendizado.

F. Terceiro modelo de regressão

O terceiro modelo de regressão foi configurado com uma rede neural MLP composta por 4 camadas: a camada de entrada possui 1 neurônio, as camadas ocultas possuem 10 neurônios cada e a camada de saída possui 1 neurônio. A taxa de aprendizado foi definida como 0.05 e o momentum foi configurado em 0.9. O modelo foi treinado por 1000 épocas. O terceiro modelo de regressão tem um desempenho bom. Ele

utiliza 10 neurônios em cada camada oculta, o que aumenta sua capacidade de aprendizado em comparação aos outros modelos. A taxa de aprendizado moderada (0.05) e o momentum de 0.9 ajudam a ajustar os pesos de forma mais estável, enquanto as 1000 épocas garantem mais iterações para o aprendizado.

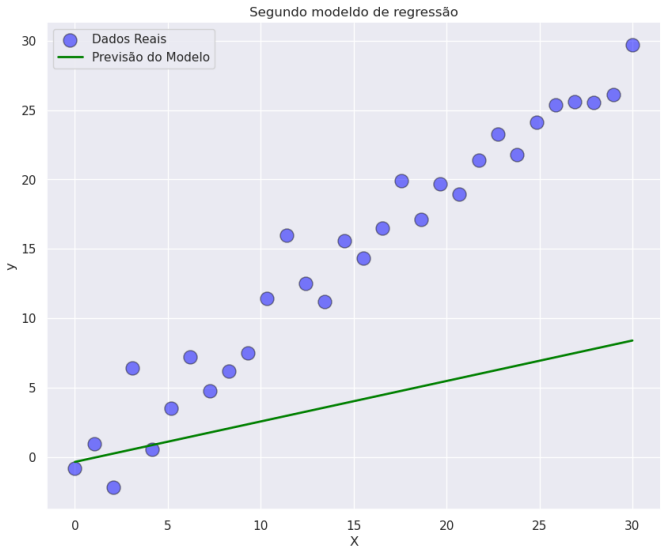

Fig. 5. Segundo modelo de regressão

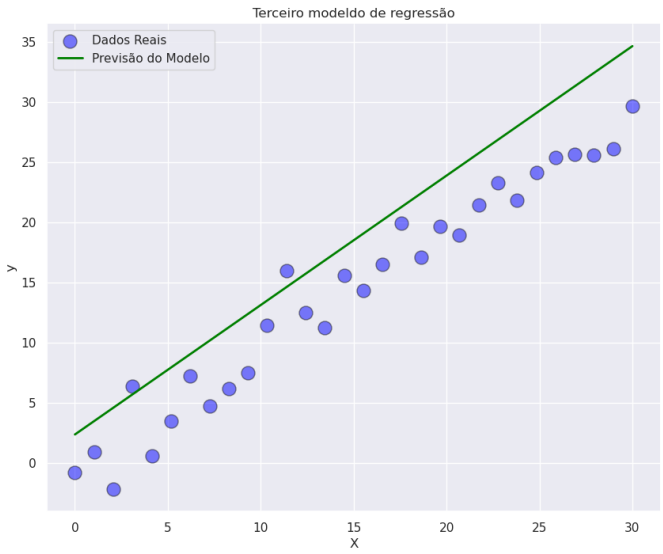

Fig. 6. Terceiro modelo de regressão

IV. Conclusão

Os resultados indicam que a escolha dos hiperparâmetros, como taxa de aprendizado e número de neurônios, impacta diretamente no desempenho dos modelos. O modelo de classificação obteve 96% de acurácia, enquanto os modelos de regressão mostraram boa performance. A otimização dos parâmetros foi crucial para melhorar os resultados.