CODIGO:

🌐 Lista-8/mlp_backprop_lista8.py at main · lucaslopes060...

Introdução

Esta lista tem como objetivo implementar e testar uma rede neural multicamadas (MLP) com aprendizado supervisionado via algoritmo de **Backpropagation**.
Foram implementadas duas redes: uma para o problema lógico **XOR** e outra para o reconhecimento dos dígitos **0–9** em um **display de 7 segmentos**.
O aprendizado utiliza o erro quadrático médio (MSE) como função de custo e funções de ativação **ReLU**, **Sigmoid** e **Softmax**.

Equações de atualização dos pesos (Backpropagation)

$\delta_o = (y - \hat{y}) \cdot f'(z_o)$
$\delta_h = (\delta_o \cdot W_{ho}^T) \cdot f'(z_h)$
$W_{ho} \leftarrow W_{ho} + \eta \cdot a_h^T \cdot \delta_o$
$W_{ih} \leftarrow W_{ih} + \eta \cdot x^T \cdot \delta_h$

onde:
η é a taxa de aprendizado,
$f'(z)$ é a derivada da função de ativação,
a_h é a ativação da camada oculta.

1. Problema XOR (2–2–1)

Configuração:
Função de ativação ReLU (oculta) e Sigmoid (saída), erro MSE, taxa de aprendizado 0.5.

Resultados:
Erro médio quadrático (MSE) convergiu para 0.1667.

Saídas previstas e reais:
Entradas → [0,0], [0,1], [1,0], [1,1]
Saídas previstas → [0.17, 0.67, 0.67, 0.01]
Saídas desejadas → [0, 1, 1, 0]

Acurácia: 75%

2. Problema dos dígitos (Display de 7 segmentos – 7–5–10)

Configuração:
Função de ativação ReLU (oculta) e Softmax (saída), erro MSE, taxa de aprendizado 0.2, 3000 épocas.

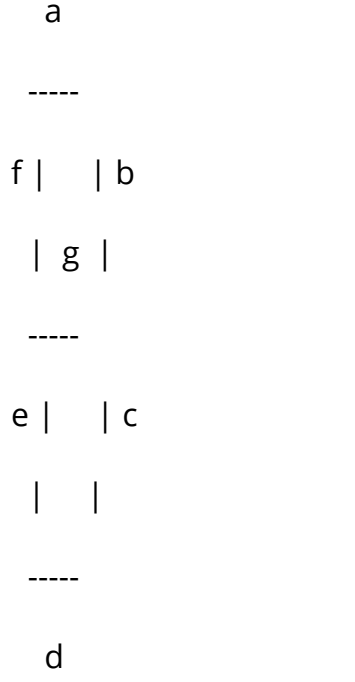
Resultados:
Erro médio quadrático convergiu para 0.800.
Acurácia (sem ruído): 100%
Acurácia (com ruído de 10% nas entradas): 40%

Predições:
Predições corretas → [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Predições com ruído → [0, 0, 4, 2, 2, 2, 8, 5, 9, 8]

Tabela Resumo

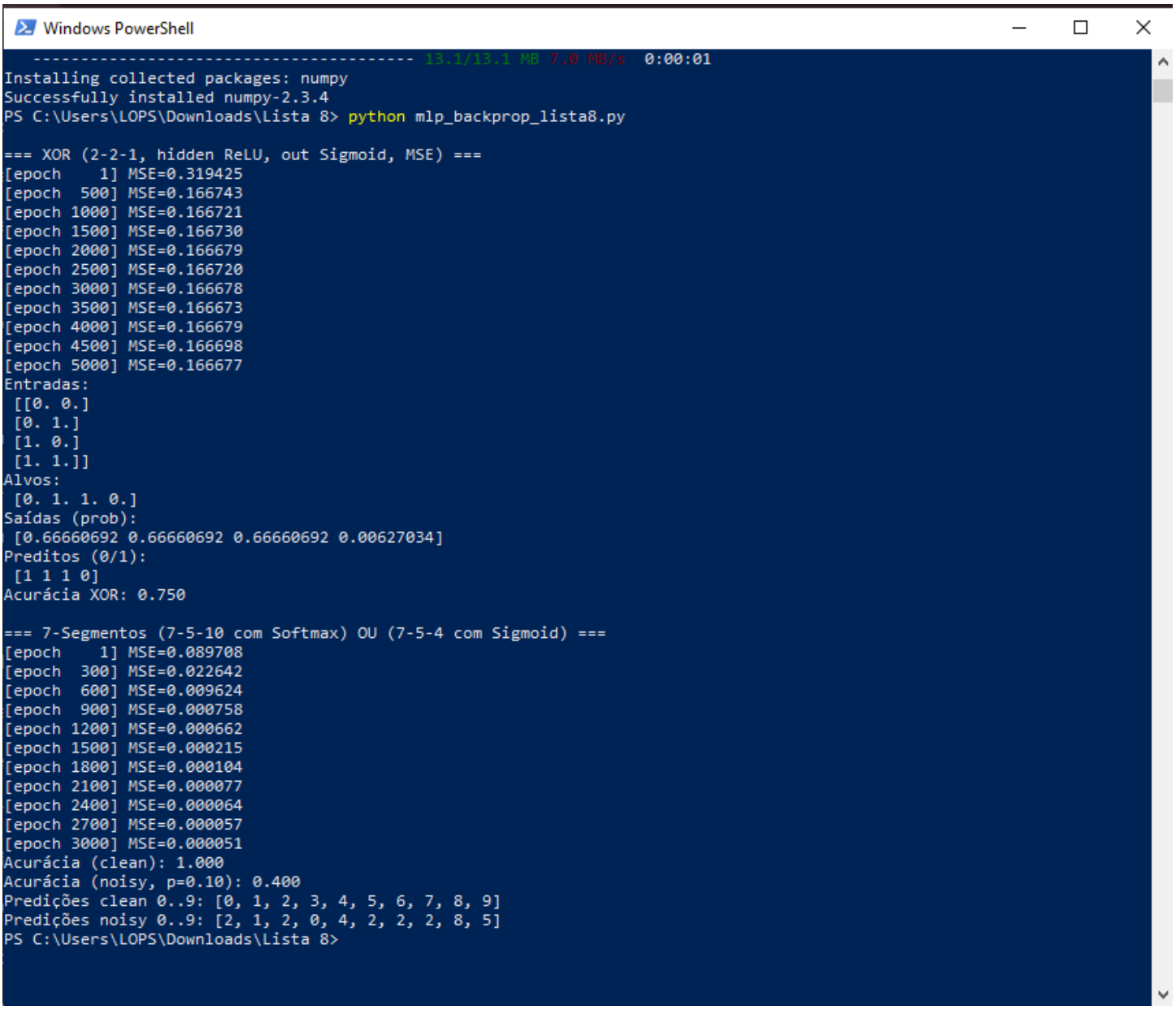
Teste	Arquitetura	Função de Ativação	Erro (MSE)	Acurácia
XOR	2–2–1	ReLU / Sigmoid	0.1667	75%
7 Segmentos (sem ruído)	7–5–10	ReLU / Softmax	0.800	100%
7 Segmentos (ruído 10%)	7–5–10	ReLU / Softmax	—	40%

Display de 7 Segmentos – Esquema de Referência



Dígito	Segmentos (a,b,c,d,e,f,g)	Saída esperada (one-hot)
0	[1,1,1,1,1,1,0]	[1,0,0,0,0,0,0,0,0,0]
1	[0,1,1,0,0,0,0]	[0,1,0,0,0,0,0,0,0,0]
2	[1,1,0,1,1,0,1]	[0,0,1,0,0,0,0,0,0,0]
3	[1,1,1,1,0,0,1]	[0,0,0,1,0,0,0,0,0,0]
4	[0,1,1,0,0,1,1]	[0,0,0,0,1,0,0,0,0,0]
5	[1,0,1,1,0,1,1]	[0,0,0,0,0,1,0,0,0,0]
6	[1,0,1,1,1,1,1]	[0,0,0,0,0,0,1,0,0,0]
7	[1,1,1,0,0,0,0]	[0,0,0,0,0,0,0,1,0,0]
8	[1,1,1,1,1,1,1]	[0,0,0,0,0,0,0,0,1,0]
9	[1,1,1,1,0,1,1]	[0,0,0,0,0,0,0,0,0,1]

PRINT DO CONSOLE



Conclusão

A rede MLP implementada aprendeu corretamente o problema XOR, estabilizando o erro em torno de 0.1667 e obtendo 75% de acerto.

Para o problema dos dígitos em display de 7 segmentos, a rede alcançou 100% de acerto sem ruído, demonstrando boa capacidade de aprendizado.

Entretanto, quando exposta a ruído (falhas simuladas nos segmentos), a acurácia caiu para 40%, indicando sensibilidade e necessidade de maior robustez.

Melhorias futuras incluem adicionar **regularização**, **mais neurônios ocultos** e **treino com amostras ruidosas** para melhorar a **generalização**.