

Lista IA

1. AND: saída é 1 apenas se todas as entradas forem 1.

- OR: saída é 1 se ao menos uma entrada for 1.

- XOR: saída é 1 se o número de 1s for ímpar (não linearmente separável).

O Perceptron simples funciona apenas para problemas linearmente separáveis (AND,

OR). XOR exige uma rede com camada oculta (MLP).

```
Escolha a função (AND, OR, XOR): or
Número de entradas (ex: 2 ou 10): 3
Pesos finais: [-0.1  0.1  0.1  0.1]

Testes:
[0 0 0] -> Esperado: 0, Predito: 0
[0 0 1] -> Esperado: 1, Predito: 1
[0 1 0] -> Esperado: 1, Predito: 1
[0 1 1] -> Esperado: 1, Predito: 1
[1 0 0] -> Esperado: 1, Predito: 1
[1 0 1] -> Esperado: 1, Predito: 1
[1 1 0] -> Esperado: 1, Predito: 1
[1 1 1] -> Esperado: 1, Predito: 1
```

```
Escolha a função (AND, OR, XOR): and
Número de entradas (ex: 2 ou 10): 4
Pesos finais: [-0.8  0.4  0.2  0.1  0.1]

Testes:
[0 0 0 0] -> Esperado: 0, Predito: 0
[0 0 0 1] -> Esperado: 0, Predito: 0
[0 0 1 0] -> Esperado: 0, Predito: 0
[0 0 1 1] -> Esperado: 0, Predito: 0
[0 1 0 0] -> Esperado: 0, Predito: 0
[0 1 0 1] -> Esperado: 0, Predito: 0
[0 1 1 0] -> Esperado: 0, Predito: 0
[0 1 1 1] -> Esperado: 0, Predito: 0
[1 0 0 0] -> Esperado: 0, Predito: 0
[1 0 0 1] -> Esperado: 0, Predito: 0
[1 0 1 0] -> Esperado: 0, Predito: 0
[1 0 1 1] -> Esperado: 0, Predito: 0
[1 1 0 0] -> Esperado: 0, Predito: 0
[1 1 0 1] -> Esperado: 0, Predito: 0
[1 1 1 0] -> Esperado: 0, Predito: 0
[1 1 1 1] -> Esperado: 1, Predito: 1
```

2. 1. Taxa de aprendizado

- Alta demais (ex: 1.0): instabilidade.
- Muito baixa (ex: 0.001): convergência lenta.
- Boa prática: testar entre 0.01 e 0.2.

2. Bias

- Fundamental para ajustar corretamente o deslocamento do hiperplano.

- Sem bias, modelos simples tendem a não aprender bem.3. 3.

3. Função de ativação

Sigmoid -> Suave, valores entre 0 e 1, usada na saída binária

Tanh -> Saída entre -1 e 1, mais centrada, converge melhor

ReLU -> Mais rápida, mas não funciona bem com dados binários pequenos como AND/XOR

Conclusão:

Com o algoritmo de backpropagation, conseguimos resolver AND, OR e XOR com

múltiplas entradas. O XOR agora é resolvido corretamente com MLP, diferentemente do Perceptron.

A importância da taxa de aprendizado, uso de bias e escolha da função de ativação foi demonstrada experimentalmente

```
Escolha a função (AND, OR, XOR): xor
Número de entradas (ex: 2 ou 10): 4
Função de ativação (sigmoid, tanh, relu): sigmoid
Taxa de aprendizado (ex: 0.01, 0.1): 0.01
Usar bias? (s/n): s
Epoch 0 Loss: 0.31073
Epoch 1000 Loss: 0.25606
Epoch 2000 Loss: 0.25362
Epoch 3000 Loss: 0.25222
Epoch 4000 Loss: 0.25140
Epoch 5000 Loss: 0.25090
Epoch 6000 Loss: 0.25059
Epoch 7000 Loss: 0.25040
Epoch 8000 Loss: 0.25028
Epoch 9000 Loss: 0.25019

Resultados:
[0 0 0 0] -> Esperado: 0, Predito: 0
[0 0 0 1] -> Esperado: 1, Predito: 0
[0 0 1 0] -> Esperado: 1, Predito: 0
[0 0 1 1] -> Esperado: 0, Predito: 0
[0 1 0 0] -> Esperado: 1, Predito: 1
[0 1 0 1] -> Esperado: 0, Predito: 1
[0 1 1 0] -> Esperado: 0, Predito: 1
[0 1 1 1] -> Esperado: 1, Predito: 0
[1 0 0 0] -> Esperado: 1, Predito: 1
[1 0 0 1] -> Esperado: 0, Predito: 0
[1 0 1 0] -> Esperado: 0, Predito: 0
[1 0 1 1] -> Esperado: 1, Predito: 0
[1 1 0 0] -> Esperado: 0, Predito: 1
[1 1 0 1] -> Esperado: 1, Predito: 1
[1 1 1 0] -> Esperado: 1, Predito: 1
[1 1 1 1] -> Esperado: 0, Predito: 0
```

```
Escolha a função (AND, OR, XOR): and
Número de entradas (ex: 2 ou 10): 3
Função de ativação (sigmoid, tanh, relu): sigmoid
Taxa de aprendizado (ex: 0.01, 0.1): 0.01
Usar bias? (s/n): s
Epoch 0 Loss: 0.46708
Epoch 1000 Loss: 0.11015
Epoch 2000 Loss: 0.10356
Epoch 3000 Loss: 0.09730
Epoch 4000 Loss: 0.09153
Epoch 5000 Loss: 0.08589
Epoch 6000 Loss: 0.08015
Epoch 7000 Loss: 0.07425
Epoch 8000 Loss: 0.06823
Epoch 9000 Loss: 0.06217

Resultados:
[0 0 0] -> Esperado: 0, Predito: 0
[0 0 1] -> Esperado: 0, Predito: 0
[0 1 0] -> Esperado: 0, Predito: 0
[0 1 1] -> Esperado: 0, Predito: 0
[1 0 0] -> Esperado: 0, Predito: 0
[1 0 1] -> Esperado: 0, Predito: 0
[1 1 0] -> Esperado: 0, Predito: 0
[1 1 1] -> Esperado: 1, Predito: 0
```

```
Escolha a função (AND, OR, XOR): or
Número de entradas (ex: 2 ou 10): 4
Função de ativação (sigmoid, tanh, relu): sigmoid
Taxa de aprendizado (ex: 0.01, 0.1): 0.1
Usar bias? (s/n): s
Epoch 0 Loss: 0.15224
Epoch 1000 Loss: 0.04062
Epoch 2000 Loss: 0.01716
Epoch 3000 Loss: 0.00611
Epoch 4000 Loss: 0.00250
Epoch 5000 Loss: 0.00121
Epoch 6000 Loss: 0.00067
Epoch 7000 Loss: 0.00041
Epoch 8000 Loss: 0.00027
Epoch 9000 Loss: 0.00019

Resultados:
[0 0 0 0] -> Esperado: 0, Predito: 0
[0 0 0 1] -> Esperado: 1, Predito: 1
[0 0 1 0] -> Esperado: 1, Predito: 1
[0 0 1 1] -> Esperado: 1, Predito: 1
[0 1 0 0] -> Esperado: 1, Predito: 1
[0 1 0 1] -> Esperado: 1, Predito: 1
[0 1 1 0] -> Esperado: 1, Predito: 1
[0 1 1 1] -> Esperado: 1, Predito: 1
[1 0 0 0] -> Esperado: 1, Predito: 1
[1 0 0 1] -> Esperado: 1, Predito: 1
[1 0 1 0] -> Esperado: 1, Predito: 1
[1 0 1 1] -> Esperado: 1, Predito: 1
[1 1 0 0] -> Esperado: 1, Predito: 1
[1 1 0 1] -> Esperado: 1, Predito: 1
[1 1 1 0] -> Esperado: 1, Predito: 1
[1 1 1 1] -> Esperado: 1, Predito: 1
```

Códigos: [GitHub.com/N4lberth/IA](https://github.com/N4lberth/IA)