

Inteligência Artificial para Robótica Móvel

CT-213

Instituto Tecnológico de Aeronáutica

Relatório do Laboratório 5 - Estratégias Evolutivas

Leonardo Peres Dias

2 de maio de 2025





Sumário

1	Breve Explicação em Alto Nível da Implementação	3
2	Figuras Comprovando Funcionamento do Código	4
2.1	Função de Classificação sum_gt_zeros	4
2.2	Função de Classificação XOR	5
2.3	Segmentação de Cores	6
3	Discussões	7
3.1	Classificação “Soma > 0”	7
3.2	Classificação XOR	7
3.3	Segmentação de Imagem	7

1 Breve Explicação em Alto Nível da Implementação

1. Inicialização:

- Recebe dimensões de entrada, da hidden layer e da saída.
- Pesos são inicializados com distribuição gaussiana ($0.001 * \text{randn}$).
- biases começam em zero.

2. Propagação Direta (`forward_propagation`):

- Ativações da camada 0 são as próprias entradas.
- Para cada camada $l = 1, 2$: $z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$, $a^{(l)} = \sigma(z^{(l)})$.

3. Função de Custo (`compute_cost`):

- Mean binary cross entropy sobre todas as amostras:

$$J = -\frac{1}{m} \sum_i [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)].$$

4. Cálculo dos Gradientes (`compute_gradient_back_propagation`):

- Erro de saída: $\delta^{(2)} = \hat{y} - y$.
- $\delta^{(1)} = W^{(2)T} \delta^{(2)} * \sigma'(z^{(1)})$.
- Gradiente dos pesos: $\nabla_{W^{(l)}} = \frac{1}{m} (\delta^{(l)} a^{(l-1)T})$.
- Gradiente dos bias: média de $\delta^{(l)}$ ao longo das amostras.

5. `back_propagation`:

- Para cada camada l : $W^{(l)} \leftarrow W^{(l)} - \alpha \nabla_{W^{(l)}}$ e $b^{(l)} \leftarrow b^{(l)} - \alpha \nabla_{b^{(l)}}$.

2 Figuras Comprovando Funcionamento do Código

2.1 Função de Classificação `sum_gt_zeros`

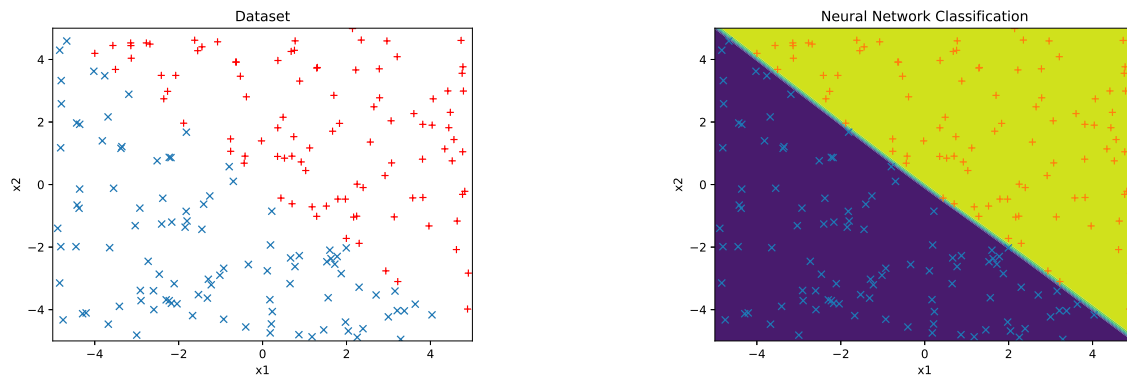


Figura 1: Tarefa de classificação.

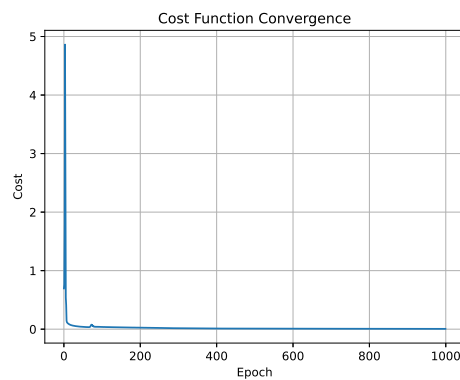


Figura 2: Convergência da função de custo.

2.2 Função de Classificação XOR

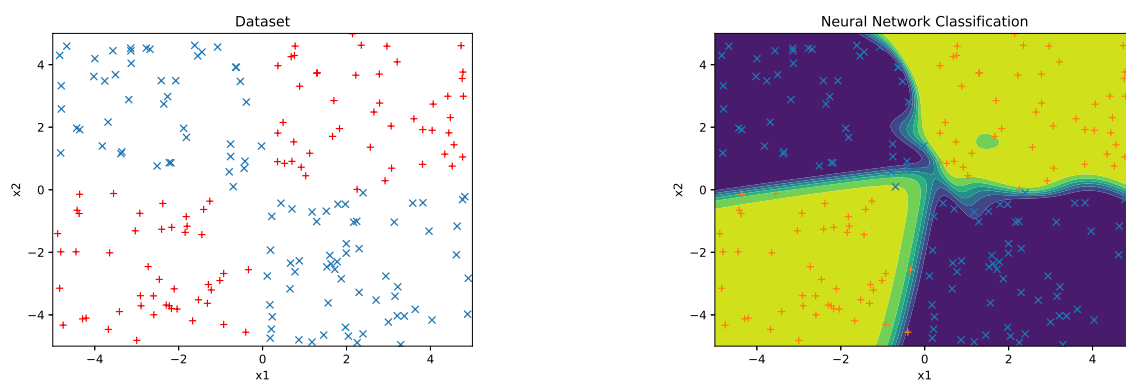


Figura 3: Tarefa de classificação.

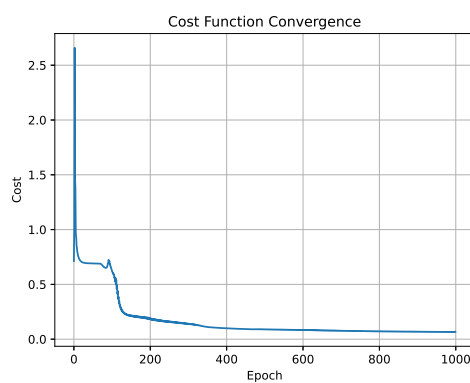


Figura 4: Convergência da função de custo.

2.3 Segmentação de Cores

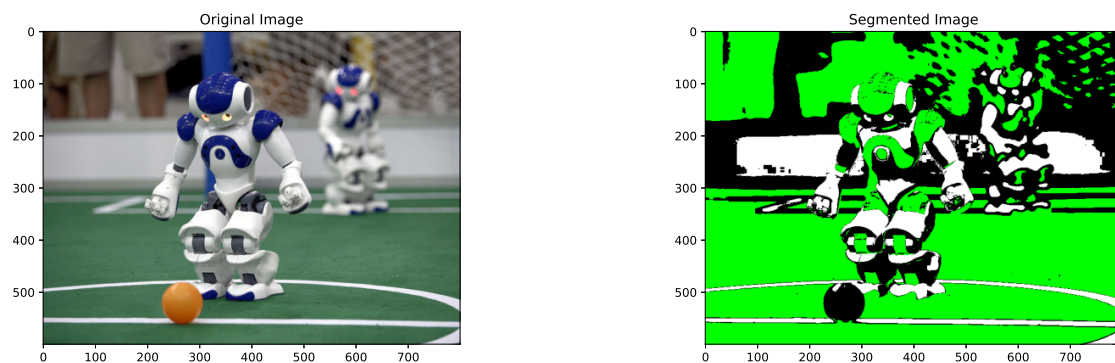


Figura 5: Tarefa de segmentação.

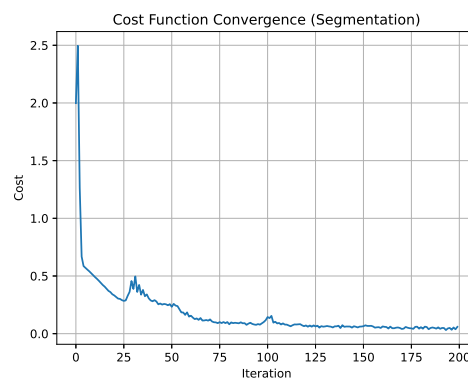


Figura 6: Convergência da função de custo.

3 Discussões

3.1 Classificação “Soma > 0”

O dataset é linearmente separável e, portanto, observa-se que:

- A função de custo converge rapidamente a valores próximos de zero, refletindo a separabilidade.
- O gráfico de *cost function* apresenta curva suave e decrescente sem oscilações, indicando estabilidade do gradiente.
- A fronteira de decisão resultante classifica corretamente todos os pontos de teste, evidenciando que a arquitetura é suficiente para esse problema.

3.2 Classificação XOR

O problema XOR é não linearmente separável e, portanto, nota-se que:

- A convergência do custo é mais lenta e atinge um mínimo acima de zero.
- Apesar disso, a rede consegue aprender bem o XOR errando minimamente a classificação.

3.3 Segmentação de Imagem

A tarefa de segmentação utiliza a mesma rede para classificar cada pixel entre branco e verde.

- A convergência do custo é mais ruidosa.
- A segmentação da imagem, classifica como verde cores que são próximas.