Instituto Tecnológico de Aeronáutica - ITA
Inteligência Artificial para Robótica Móvel - CT213
Aluno: Matheus Felipe Ramos Borges

Relatório do Laboratório 6 - Redes Neurais

1 Breve Explicação em Alto Nível da Implementação

O objetivo deste laboratório é implementar uma rede neural com tres camadas (1 camada escondida) para realizar a segmentação de cores em imagens. A rede é treinada para classificar cada pixel da imagem em uma das seguintes classes: verde, branco ou outras cores (representadas pela cor preta).

Foi utilizada a função de ativação sigmoide em todas as camadas, o que permite interpretar as saídas como probabilidades. Como se trata de um problema de classificação multiclasse, a função de custo escolhida foi a função de perda logística multiclasse:

$$L\left(y^{(i)}, \hat{y}^{(i)}\right) = -\sum_{c=1}^{C} \left[\left(1 - y_c^{(i)}\right) \log\left(1 - \hat{y}_c^{(i)}\right) + y_c^{(i)} \log\left(\hat{y}_c^{(i)}\right)\right] \tag{1}$$

A função de custo total da rede é então definida como a média da função de perda individual para cada exemplo de treinamento:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L\left(y^{(i)}, \hat{y}^{(i)}\right) \tag{2}$$

O treinamento da rede é realizado via back propagation, que calcula os gradientes da função de custo em relação aos parâmetros da rede (pesos e vieses), permitindo a atualização desses parâmetros por meio de algoritmos de otimização, como o gradiente descendente. Os gradientes para cada camada são dados por:

$$\frac{\partial L}{\partial w_{ck}^{[2]}} = \delta_c^{[2]} a_k^{[1]} \tag{3}$$

$$\frac{\partial L}{\partial b_c^{[2]}} = \delta_c^{[2]} \tag{4}$$

$$\frac{\partial L}{\partial w_{kj}^{[1]}} = \delta_k^{[1]} a_j^{[0]} \tag{5}$$

$$\frac{\partial L}{\partial w_{kj}^{[1]}} = \delta_k^{[1]} a_j^{[0]} \tag{6}$$

$$\frac{\partial L}{\partial b_k^{[1]}} = \delta_k^{[1]} \tag{7}$$

onde os termos de erro são definidos como:

$$\delta_c^{[2]} = (\hat{y}_c - y_c) \tag{8}$$

$$\delta_k^{[1]} = \sum_{c=1}^{C} w_{ck}^{[2]} \delta_c^{[2]} \sigma'(z_k^{[1]}) \tag{9}$$

A etapa de forward propagation foi implementada de forma vetorizada para garantir maior eficiência computacional. O processo segue as equações abaixo:

$$\boldsymbol{a}^{[0]} = \boldsymbol{x} \tag{10}$$

De $l = 1$ até L

$$\boldsymbol{z}^{[l]} = \boldsymbol{W}^{[l]} \boldsymbol{a}^{[l-1]} + \boldsymbol{b}^{[l]} \tag{11}$$

$$\boldsymbol{a}^{[l]} = g^{[l]}(\boldsymbol{z}^{[l]}) \tag{12}$$

$$\hat{y} = \boldsymbol{a}^{[L]} \tag{13}$$

2 Figuras Comprovando Funcionamento do Código

2.1 Função de Classificação *sum_ gt_ zeros*

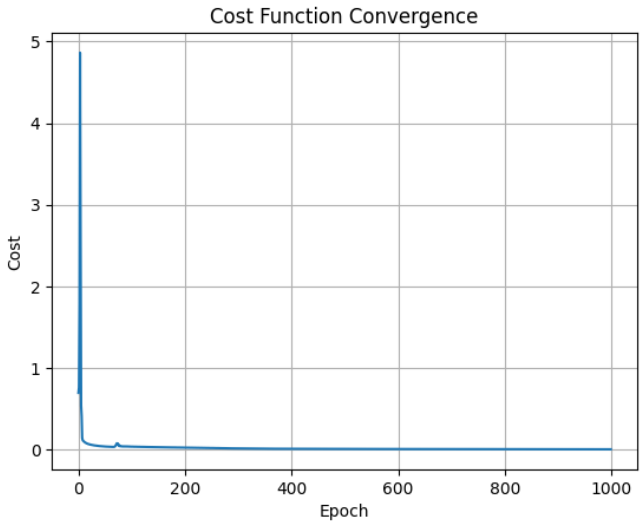

Figura 1: gráfico de convergência da função custo para o *sum_ gt_ zeros*.

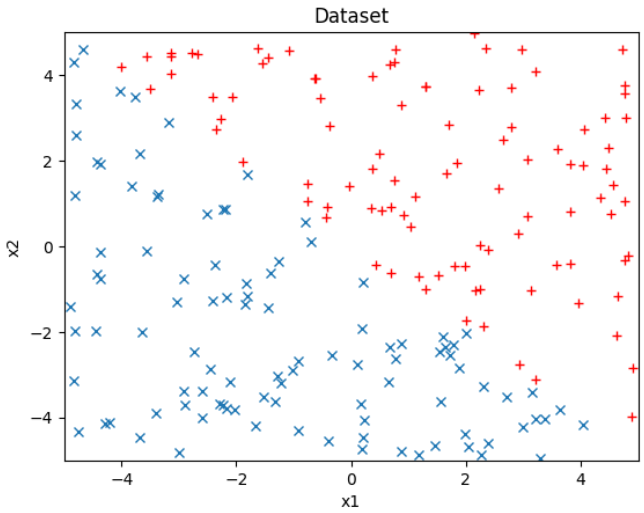

Figura 2: gráfico do dataset para o *sum_gt_zeros*.

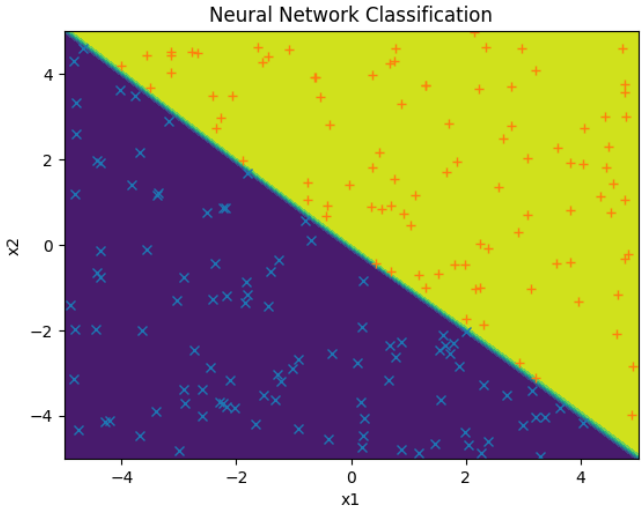

Figura 3: gráfico de classificação da rede neural *fitness* para o *sum_gt_zeros*.

2.2 Função de Classificação *XOR*

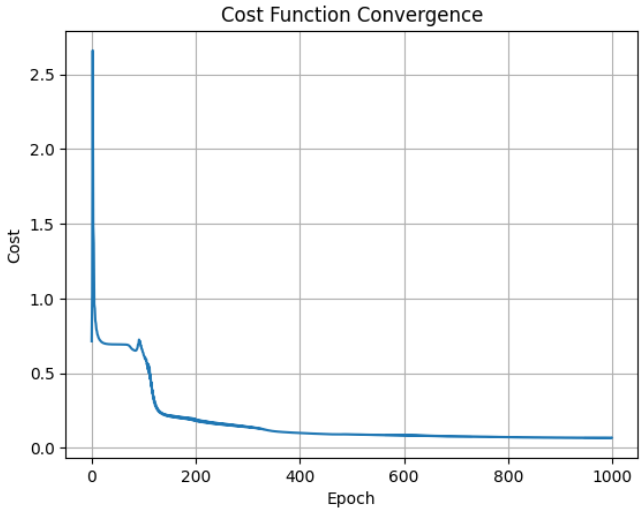

Figura 4: gráfico de convergência da função custo para o *XOR*.

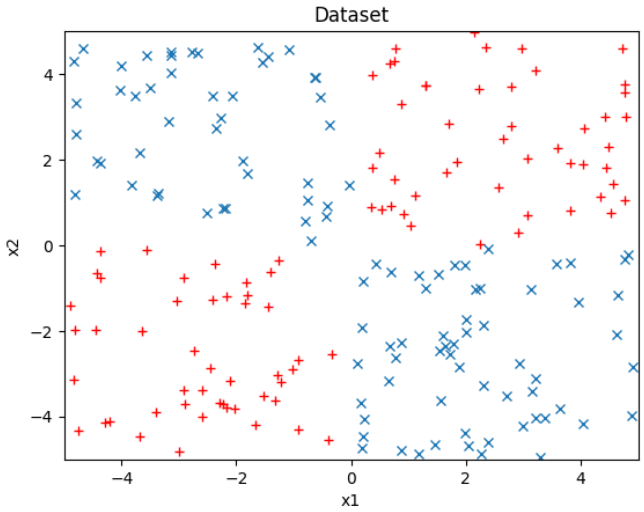

Figura 5: gráfico do dataset para o *XOR*.

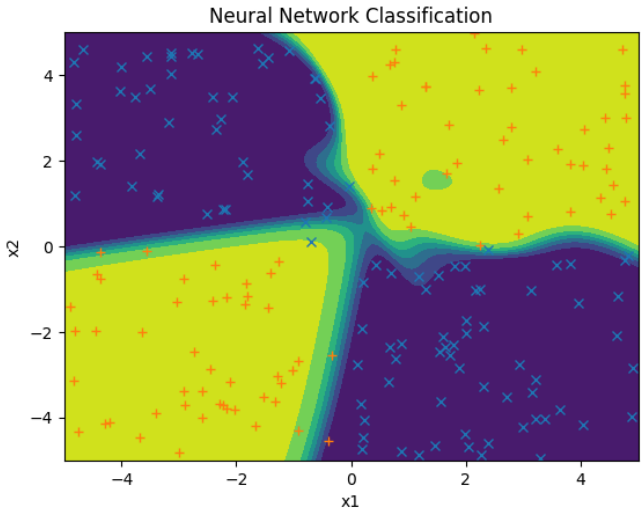

Figura 6: gráfico de classificação da rede neural *fitness* para o *XOR*.

2.3 Segmentação de Cores

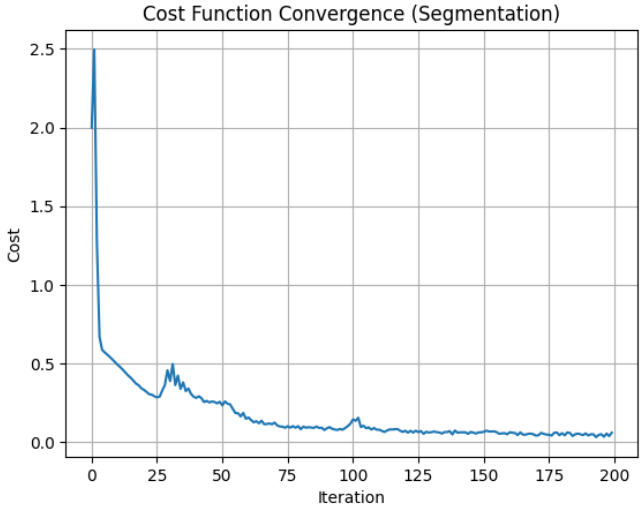

Figura 7: gráfico de convergência da função custo para a segmentação por cores.

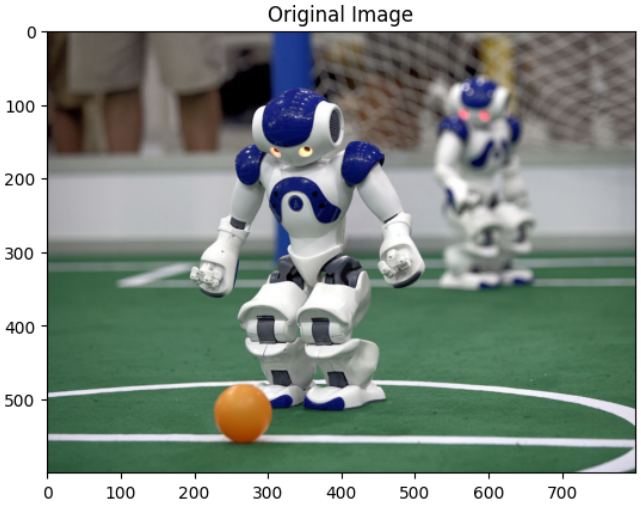

Figura 8: Imagem do robô NAO.

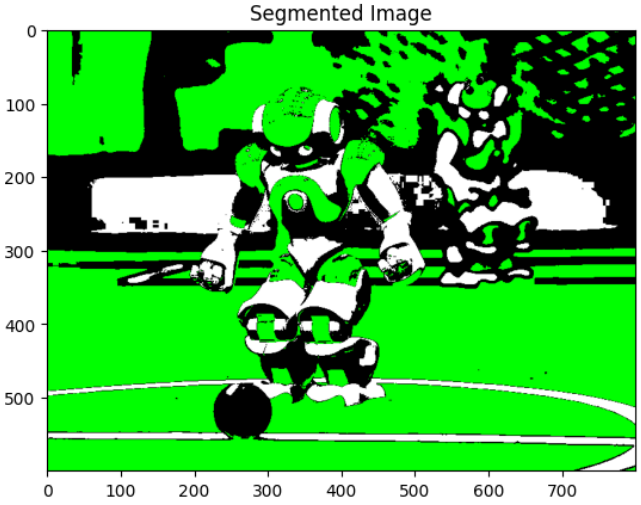

Figura 9: Imagem segmentada por cores do robô NAO.

3 Discussões

A rede neural apresentou boa convergência nos testes realizados, como nos problemas *sum_gt_zeros* e *XOR*, demonstrando capacidade de modelar relações não lineares e separações complexas no espaço de entrada. Isso indica que os mecanismos de foward propagation e back propagation foram corretamente implementados e que a arquitetura, mesmo simples, é suficiente para tarefas básicas de classificação.

Na segmentação de cores do robô NAO, a rede obteve resultados satisfatórios ao distinguir verde, branco e outras cores. A principal limitação observada foi na diferenciação entre o azul do corpo do robô e a classe verde, o que é esperado devido à proximidade espectral e à capacidade restrita do modelo.