Instituto Tecnológico de Aeronáutica - ITA
Inteligência Artificial para Robótica Móvel - CT213
Aluno: Arthur de Sousa Vianna

Relatório do Laboratório 6 - Redes Neurais

1 Breve Explicação em Alto Nível da Implementação

Uma rede neural é um conceito de *machine learning* que, simplificadamente, tem como objetivo a realização de um "*fitting*" dos dados de um *dataset* em uma função de tal forma que os *outputs* sejam coerentes para quaisquer *inputs*, inclusive foras do *dataset*. O código teve como objetivo implementar uma rede neural de classificação multiclasse de três camadas: a camada de entrada, uma camada escondida e a camada de saída. Todos os neurônios tinham a sigmoide como função de ativação. Como se trata de um problema de classificação multiclasse, a *loss function* utilizada foi a regressão logística multiclasse. Na implementação do *forward propagation* da rede neural, na função `forward_propagation`, determinaram-se as matrizes $\mathbf{Z}$ e $\mathbf{A}$, de tal forma que a primeira dimensão corresponde ao número da camada (l) e a segunda, à matriz relativa à cada sessão de treinamento em m colunas, com m sendo o número de amostras em cada treinamento. Note que no cálculo de $\mathbf{Z}^{[l]}$ (matriz $\mathbf{Z}$ para a l-ésima camada), o vetor de *biases* de dimensão $n_l \times 1$ (n_l número de neurônios da camada l) sofre um *broadcasting* para $n_l \times m$, permitindo a adição de matrizes. Na função `compute_gradient_back_propagation`, calcularam-se a matriz dos gradientes dos pesos para cada neurônio e dos vieses, calculados a partir da *loss function*. Um dos pontos importantes da implementação é a lembrança de se calcular a média de cada *sample* de um treinamento, atentando-se às dimensões das matrizes $\mathbf{Z}$, $\mathbf{A}$ e dos *outputs*. Ainda, no vetor de gradiente dos *biases*, foi necessário realizar um *reshape* do *array* para permitir o *broadcasting* no cálculo da descida do gradiente. Por fim, na função `back_propagation`, implementou-se a descida do gradiente para otimização dos parâmetros (pesos e vieses) da rede neural em cada treinamento, utilizando-se a taxa de aprendizagem α (hiperparâmetro) e os gradientes calculados na função anterior.

2 Figuras Comprovando Funcionamento do Código

2.1 Função de Classificação *sum_gt_zeros*

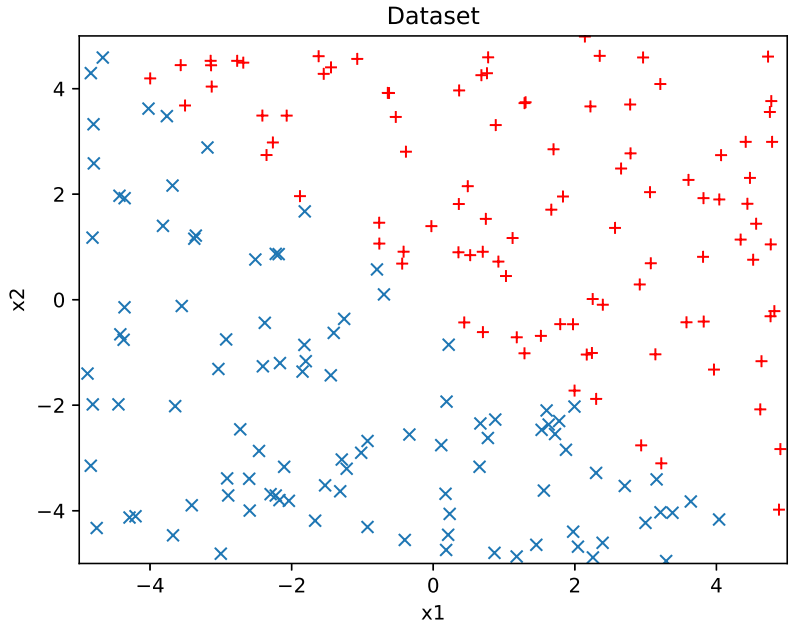

Figura 1: *Dataset* gerado para a função de classificação *sum_gt_zeros*.

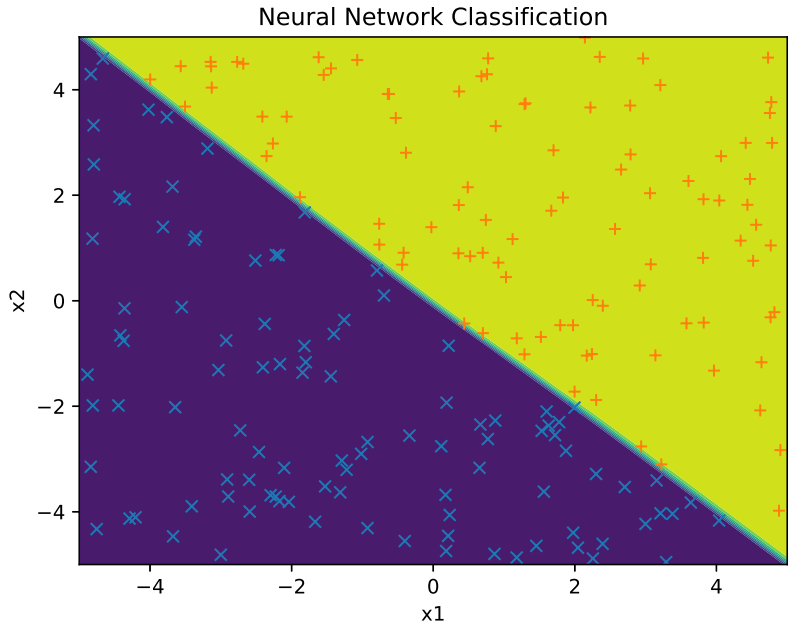

Figura 2: Regiões de classificação feitas pela rede neural para a função *sum_gt_zeros*.

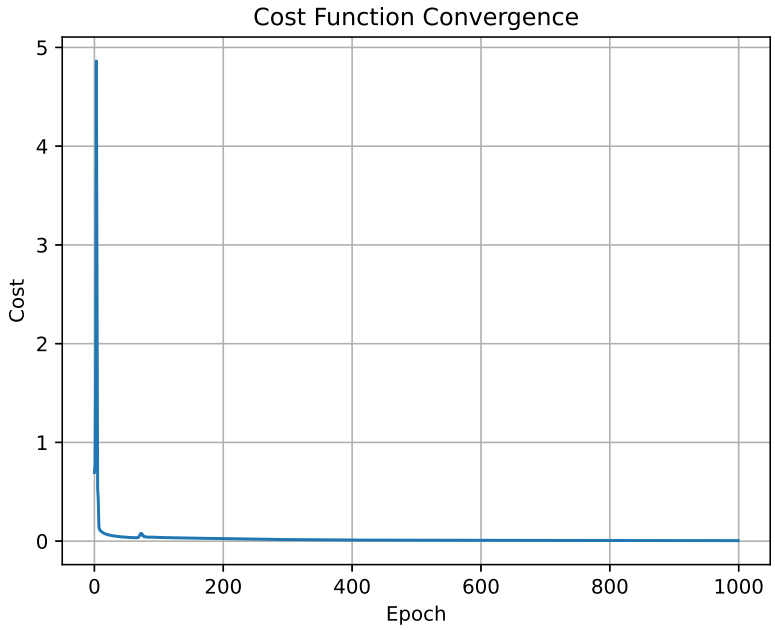

Figura 3: Convergência da função de custo para a função de classificação *sum_gt_zeros*.

2.2 Função de Classificação *XOR*

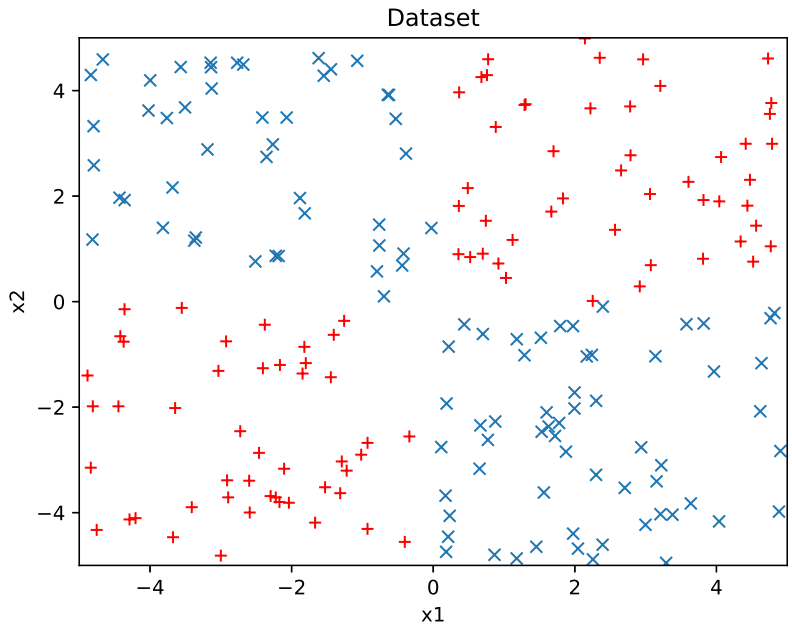

Figura 4: *Dataset* gerado para a função de classificação *XOR*.

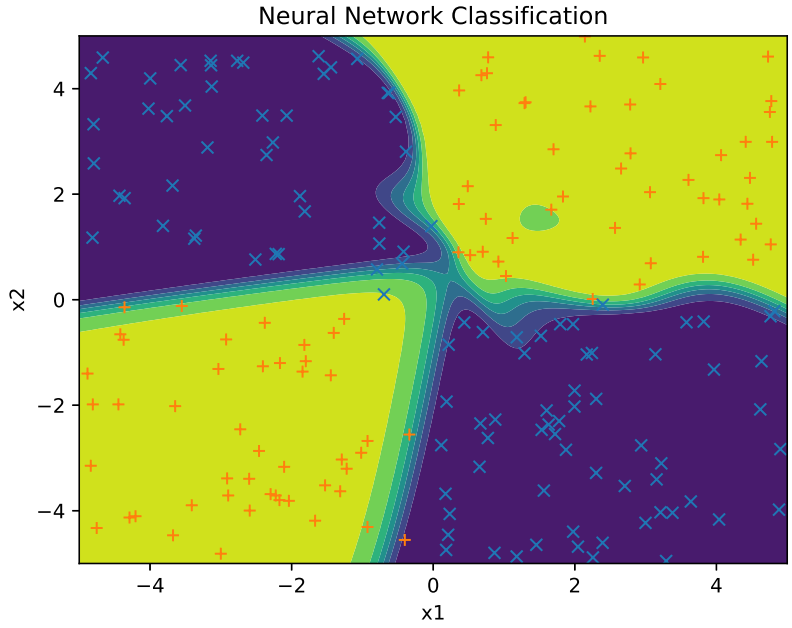

Figura 5: Regiões de classificação feitas pela rede neural para a função *XOR*.

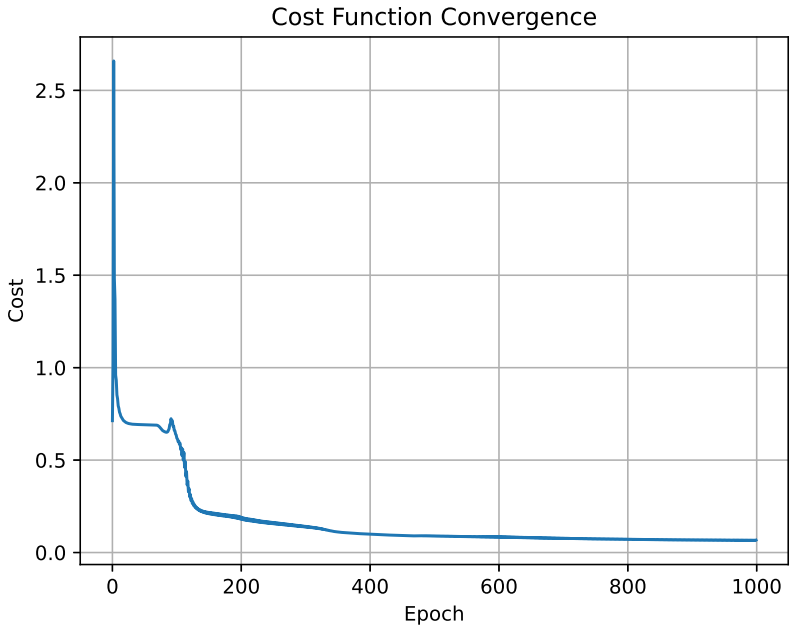

Figura 6: Convergência da função de custo para a função de classificação *XOR*.

2.3 Segmentação de Cores

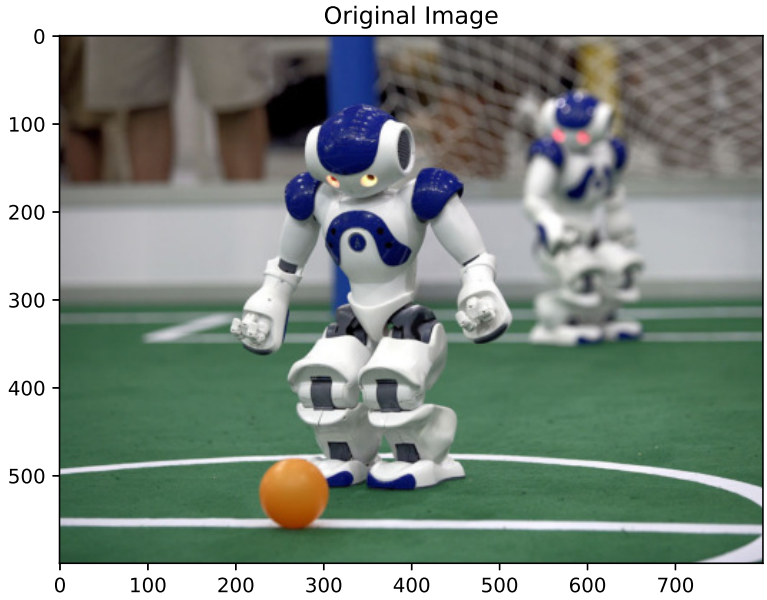

Figura 7: Imagem original utilizada na segmentação de cores.

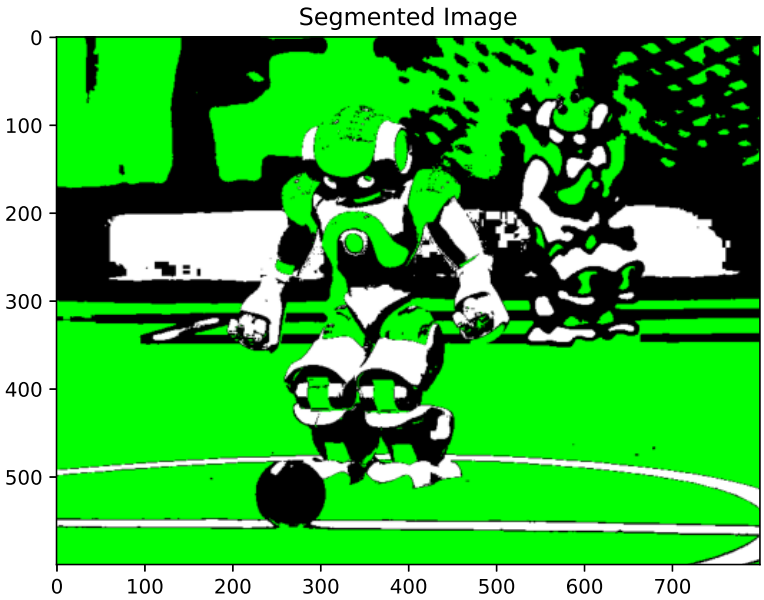

Figura 8: Imagem segmentada obtida pela rede neural.

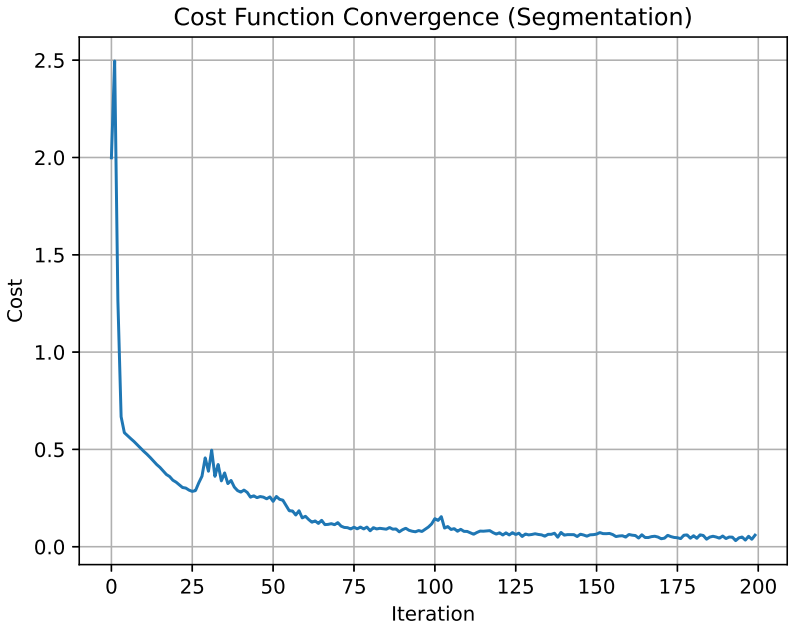

Figura 9: Convergência da função de custo para a segmentação da imagem.

3 Discussões

Na função de classificação *sum_gt_zeros*, como ela é linearmente separável, a convergência foi mais suave e a região de classificação obtida pela rede neural foi coerente aos *labels* do *dataset*, sendo visível uma clara separação linear. Por outro lado, na função *XOR*, a convergência foi menos suave a houve a ocorrência de um pico, provavelmente devido à natureza não-linear da função, tornando mais complexo o ajuste dos parâmetros, principalmente no início. Note ainda que a região de classificação obtida para essa função apresenta curvas coerentes com o *dataset*. Em relação à segmentação de cores, nota-se, novamente, picos irregulares na função de custo, o que pode ser explicado devido à natureza mais complexa de distinção de classificação do problema, envolvendo ajustes mais finos dos parâmetros da rede neural, além de os códigos de cores apresentarem ruídos nas imagens do *dataset*, dificultando a convergência. O resultado final da imagem segmentada foi coerente com o esperado.