Laboratório 6 - Redes Neurais

Marcelo Buga Martins da Silva

CT-213 - Professor Marcos Ricardo Omena de Albuquerque Máximo

09/05/2021

1 Implementação da Rede Neural

Nesse laboratório foi implementado um algoritmo um algoritmo de rede neural de três camadas com intenção de realizar a segmentação de cores de imagens de futebol de robôs. Para tal implementação, foram utilizadas duas funções: *forward_propagation* e *back_propagation*. Uma visão gerada da implementação de cada função é dada a seguir:

1.1 Forward Propagation

O Forward Propagation foi implementado de forma vetorizada, simplesmente aplicando a fórmula da sigmoide (função de ativação escolhida) e atualizando o vetor Z para cada camada da rede neural a partir dos vetores de pesos e *biases*.

1.2 Back Propagation

O Back Propagation também foi implementado de forma vetorizada de forma a acelerar a execução do algoritmo. Nessa implementação, utilizou-se de vetores δ para a programação dinâmica do algoritmo e poupar

poder computacional. Foram então calculados os vetores das derivadas parciais em relação a cada parâmetro e atualizado os valores dos vetores de pesos e de *biases* com a média dos gradientes calculados. Essa parte do algoritmo configura a etapa de treinamento da rede neural.

2 Teste da Rede Neural

Para testar a implementação, foram utilizadas duas funções simples. A rede neural foi submetida a 100 épocas de treinamento a partir de um *dataset* de 200 casos para cada função. Primeiramente, testou-se com a função *sum_gt_zero* que retorna 1 se a soma das coordenas de um ponto x é maior que 0 e 0 caso contrário. As Figuras 1 e 2 representam a convergência do custo e a classificação das regiões pela rede neural, respectivamente:

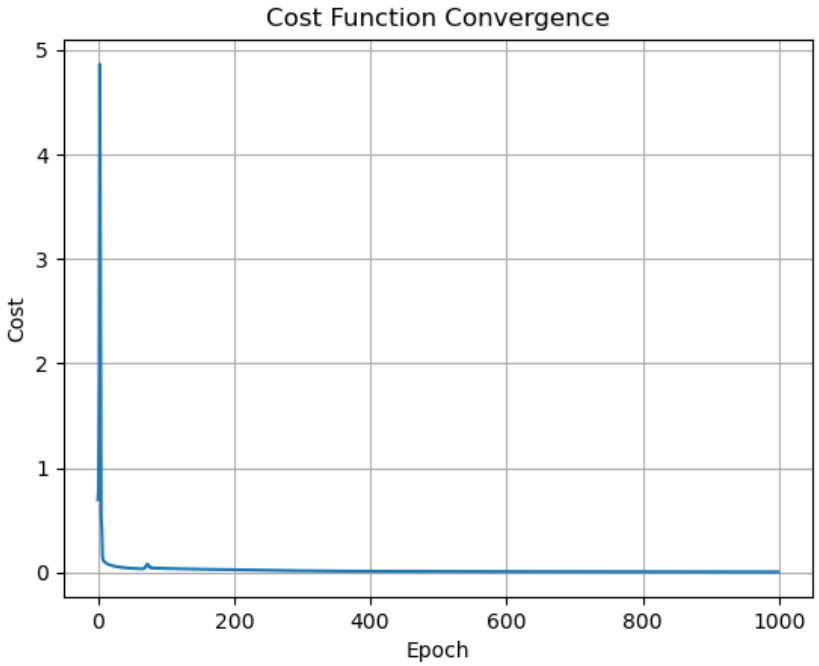

Figura 1: Convergência da função de custo para *sum_gt_zero*

Por sua vez, para a função *xor* que retorna 1 se o sinal das coordenadas forem iguais e 0 se forem distintos, teve-se o resultado representado nas Figuras 3 e 4:

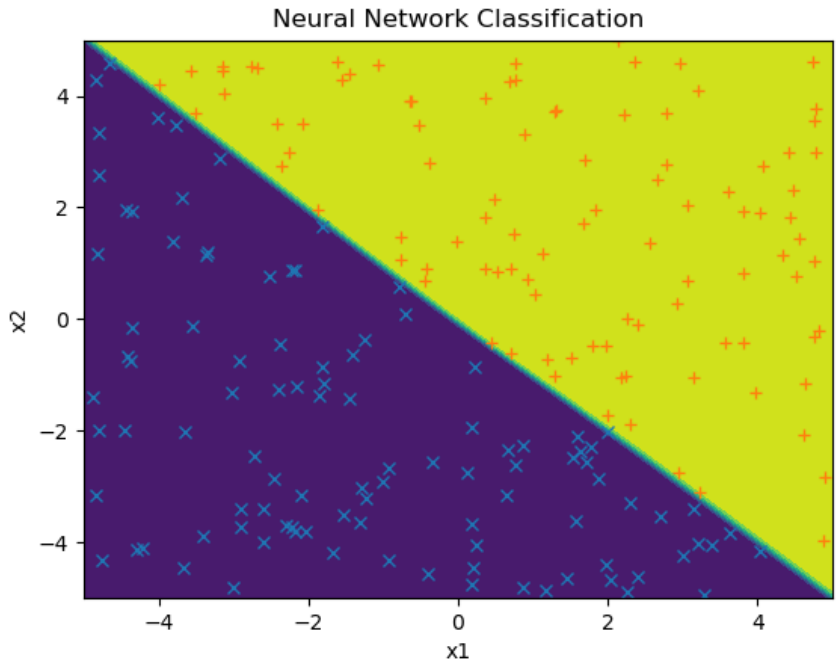

Figura 2: Classificação de regiões de *sum_gt_zero*

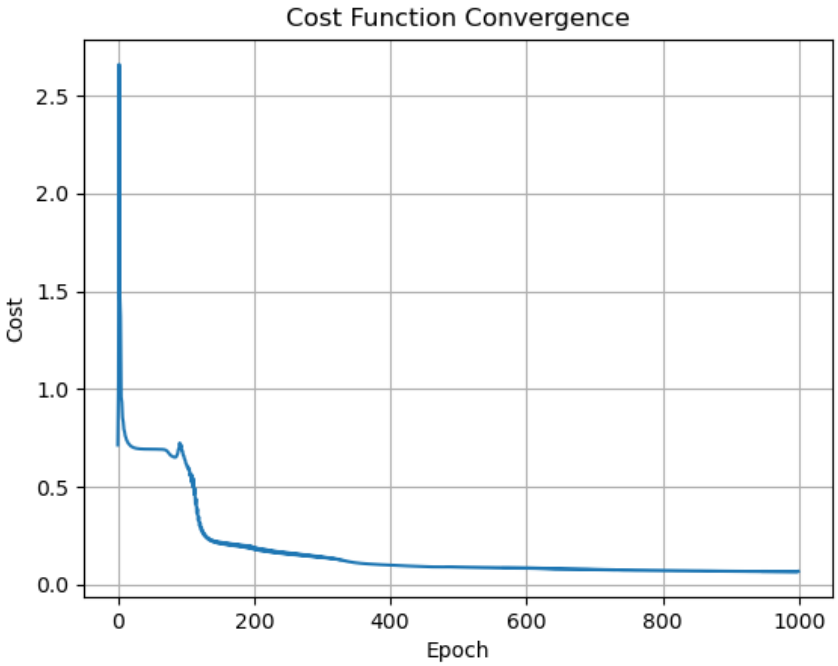

Figura 3: Convergência da função de custo para *xor*

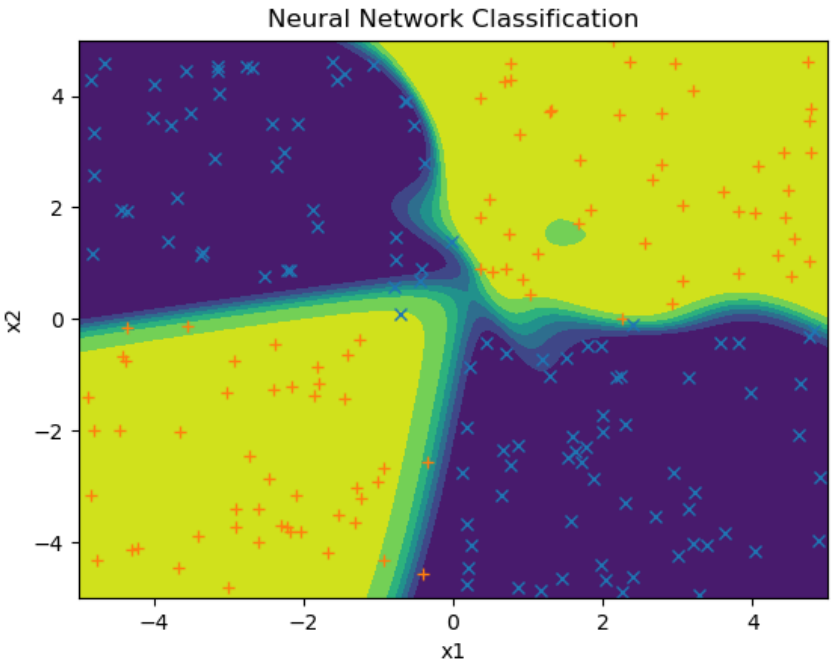

Figura 4: Classificação de regiões de *xor*

Percebe-se que o resultado para a função *sum_gt_zero* foi muito satisfatório, enquanto que, para a função *xor* um número maior de treinamentos ou um aumento no *dataset* é necessário para melhorar a descrição das regiões. De qualquer forma, a implementação provou-se adequada e foi utilizada para o teste da segmentação de cores.

3 Teste da Segmentação de Cores

Por fim, utilizou-se da implementação da rede neural para testar a segmentação das cores verde e branco de uma imagem de futebol de robôs. As Figuras 5 e 6 representam a imagem original e a imagem segmentada após 300 iterações de treinamento de *mini-batches* de 100 *samples*, que teve convergência do custo como apresentado na Figura 7:

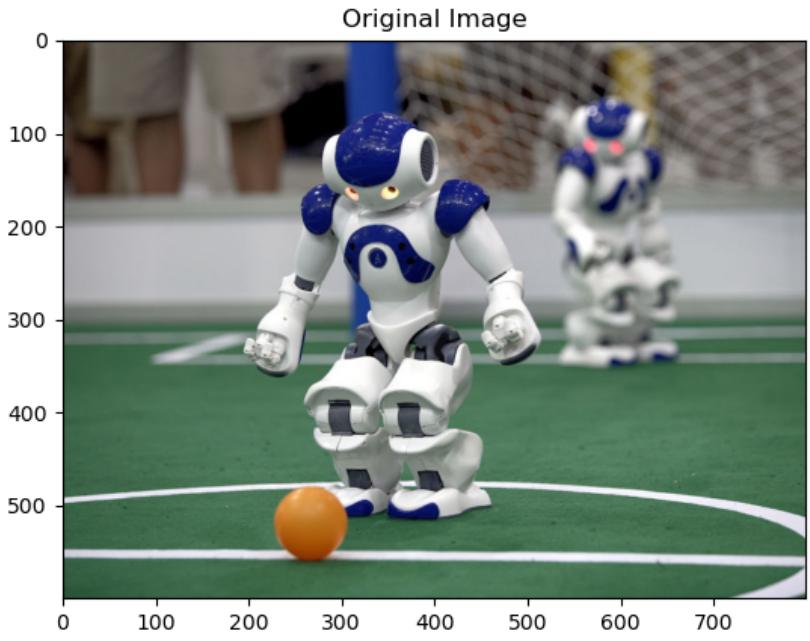

Figura 5: Imagem Original

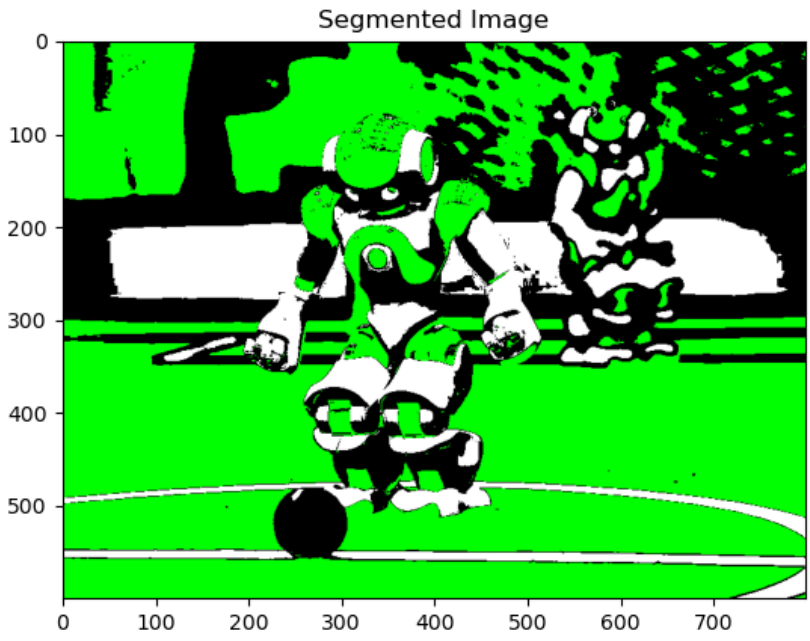

Figura 6: Imagem Segmentada

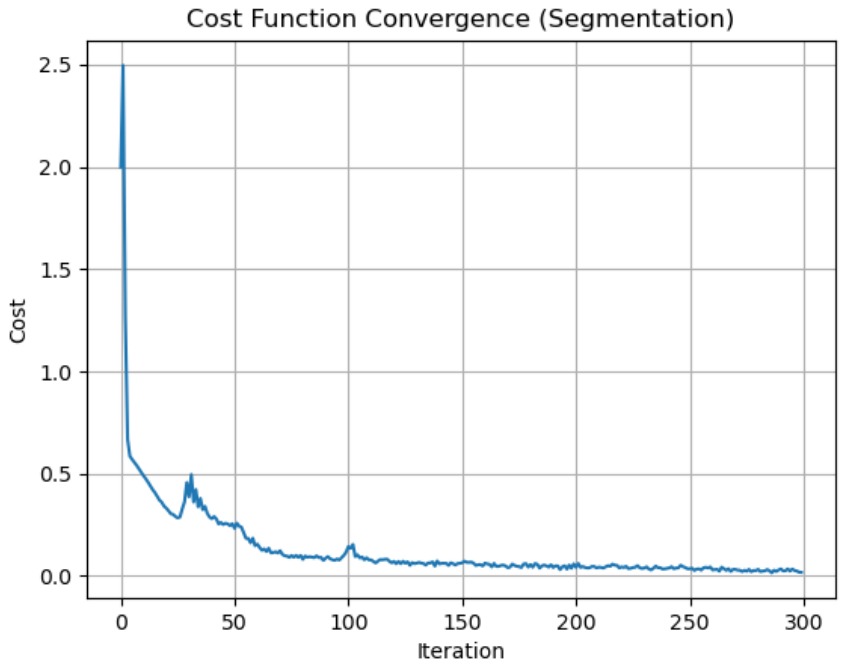

Figura 7: Convergência da função de custo para segmentação de cores

Pode-se perceber que o resultado da segmentação foi bem satisfatório para as identificações mais importantes: o campo e as linhas. A função de custo também conseguiu convergir de forma relativamente rápida, provando que a utilização de redes neurais é bem viável para a tarefa de segmentação de cores.