Redes Neurais

Samara Ribeiro Silva
Instituto Tecnológico de Aeronáutica, Laboratório de Inteligência Artificial para Robótica Móvel (CT-213). Professor Marcus Ricardo Omena de Albuquerque Máximo, São José dos Campos, São Paulo, 09 de maio de 2021.

Foi implementada uma rede neural utilizando descida de gradiente.

Os valores de z e a foram calculados:
para $i > 0$

$$z_i = w_i a_{i-1} + b_i$$
$$a_i = g(z_i)$$

a função g é a função de ativação (neste laboratório utilizaremos a função Sigmóide $\sigma(x) = \frac{e^x}{e^x + 1}$).

```
z[i] = self.weights[i]@a[i-1] + self.biases[i]
a[i] = sigmoid(z[i])
```

para $i = 0$ z e a são iguais aos inputs.

Os valores de w e b são atualizados da seguinte maneira:

$$W = W - \alpha \frac{\partial L}{\partial W}$$
$$b = b - \alpha \frac{\partial L}{\partial b}$$

```
self.weights[i]=self.weights[i] - self.alpha * weights_gradient[i]
self.biases[i] = self.biases[i] - self.alpha * biases_gradient[i]
```

O weights_gradient e biases_gradientes são calculados utilizando:

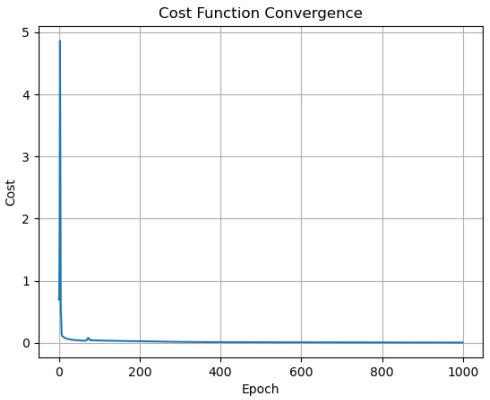
weights_gradient	$\frac{\partial L}{\partial w^{[1]}_{kj}} = \delta^{[1]}_{k} a^{[0]}_{j}$	$\frac{\partial L}{\partial w^{[2]}_{ck}} = \delta^{[2]}_{c} a^{[1]}_{k}$
biases_gradientes	$\frac{\partial L}{\partial b^{[1]}_{k}} = \delta^{[1]}_{k}$	$\frac{\partial L}{\partial b^{[2]}_{c}} = \delta^{[2]}_{c}$

delta	$\delta_k^{[1]} = \sum_{c=1}^{C} w_{ck}^{[2]} \delta_c^{[2]} \sigma'(z_k^{[1]})$	$\delta_c^{[2]} = (\hat{y}_c - y_c)$

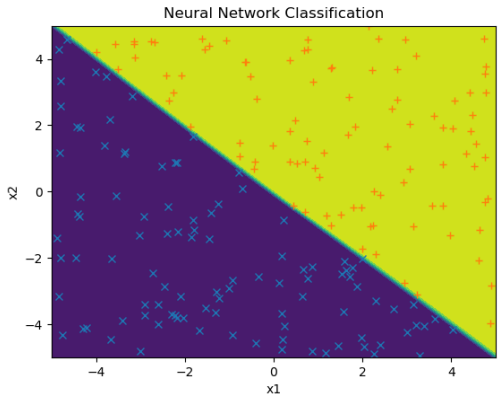
```
m = inputs.shape[1]

delta[2] = (a[2] - expected_outputs)
weights_gradient[2] = (delta[2]@a[1].T) / m
biases_gradient[2] = np.array(np.mean(delta[2], axis=1))
biases_gradient[2] =
biases_gradient[2].reshape(biases_gradient[2].shape[0], 1)

delta[1] = (self.weights[2].T@delta[2])*sigmoid_derivative(z[1])
weights_gradient[1] = (delta[1] @ a[0].T) / m
biases_gradient[1] = np.array(np.mean(delta[1], axis=1))
biases_gradient[1] =
biases_gradient[1].reshape(biases_gradient[1].shape[0], 1)
```

Nas figuras 1, 2 e 3 pode-se observar o funcionamento do algoritmo. Analisando as figuras 1 (b) e (c) nota-se que houve uma boa classificação. O mesmo ocorre no teste das figuras 2 (b) e (c) na qual é possível observar que não houve overfitting. Por fim, observe na figuras 3 a segmentação de cores foi realizada com sucesso atingindo assim o objetivo desta prática laboratorial.

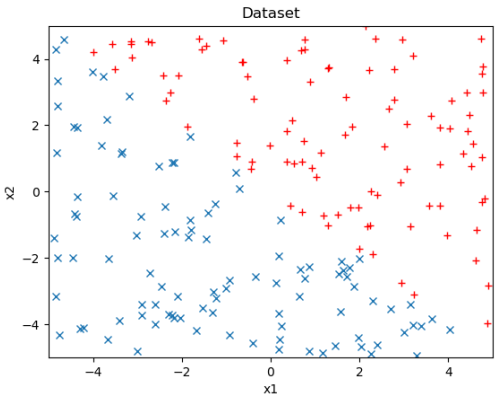
Figura 1: Teste Neural Network (sum_gt_zero)



(a)

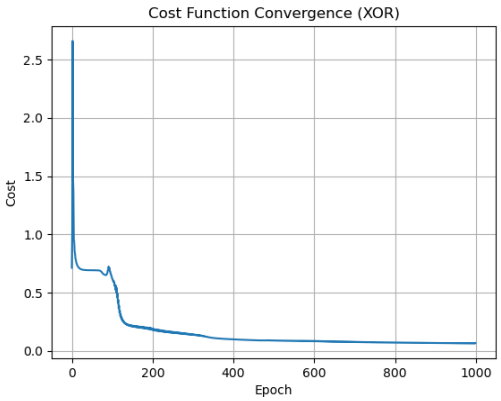


(b)

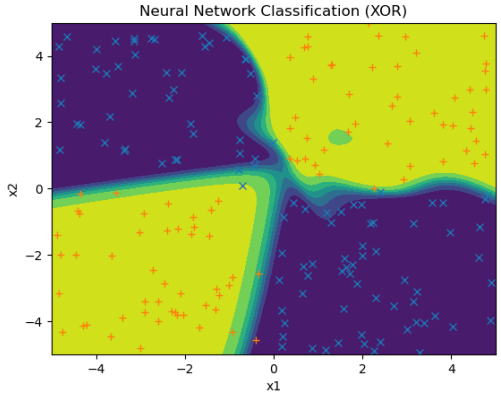


(c)

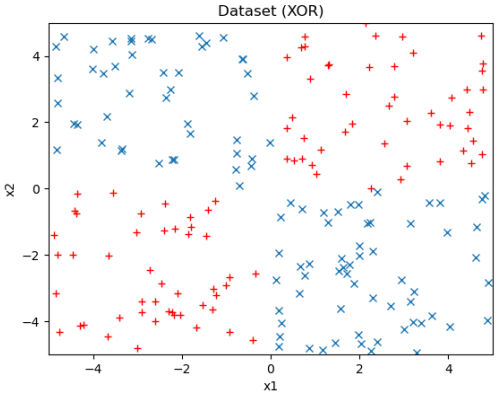
Figura 2: Teste Neural Network (XOR)



(a)

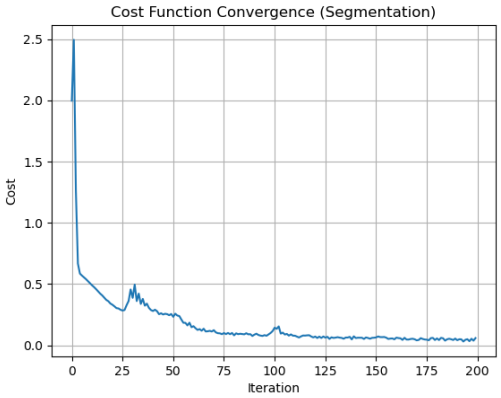


(b)

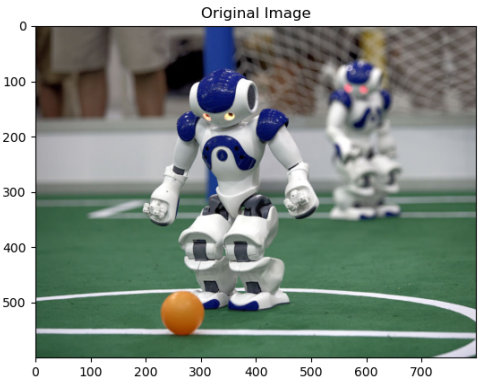


(c)

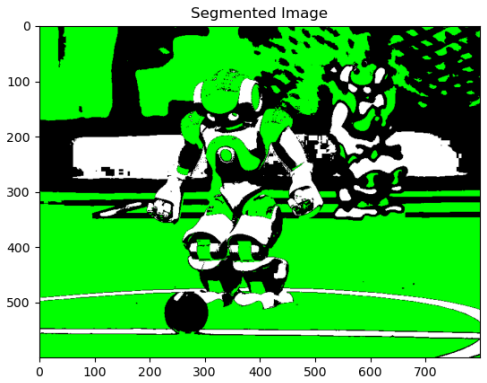
Figura 3:Teste Color Segmentation



(a)



(b)



(c)