

Atividade Disciplina: Deep Learning

Essa atividade está dividida em duas partes. Ao término, poste a atividade no link disponibilizado no moodle

Parte 1:

1) Abra o colab.research.google.com e crie um novo notebook.

2) Implemente o algoritmo abaixo para atualização dos pesos da rede perceptron que considera a atualização de pesos:

$$w_{i,j}^{(\text{next step})} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$$

e a função de ativação:

$$\text{heaviside}(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}$$

Predição(w, x, bias)

ativação = $w^T * x$ + bias

retorna 1 se ativação ≥ 0 senão 0

Repita (até a quantidade de épocas)

Para cada instância (x,y)

erro = y_i - predição(w, x, bias)

bias = bias + α * erro

Para cada peso w_k

$w[k] = w[k] + \alpha * \text{erro} * x[k]$

Considere os pesos e o bias inicial iguais a zero, taxa de aprendizado = 0.1 e número de épocas=5.

Ainda, considere o seguinte conjunto de entrada:

2,781083600	2,550537003	0
1,465489372	2,362125076	0
3,396561688	4,400293529	0
1,38807019	1,850220317	0
3,06407232	3,005305973	0
7,627531214	2,759262235	1
5,332441248	2,088626775	1

6,922596716	1,77106367	1
8,675418651	-0,242068655	1
7,673756466	3,508563011	1

●O resultado esperado é:

- Peso= [0.20653640140000007, -0.23418117710000003]
- Bias: -0.1

●Sua implementação deve seguir o preenchimento do código descrito abaixo:

```
import numpy as np
def predicao(w,x,bias):
    #SEU CODIGO AQUI

def main():
    x_instances=[[2.7810836, 2.550537003],[1.465489372, 2.362125076],[3.396561688,
4.400293529], [1.38807019, 1.850220317],[3.06407232, 3.005305973],[7.627531214,
2.759262235],[5.332441248, 2.088626775],[6.922596716, 1.77106367],[8.675418651,
-0.242068655],[7.673756466, 3.508563011]]
    yi_label_instances=[0,0,0,0,0,1,1,1,1,1]
    #SEU CODIGO AQUI

if __name__ == "__main__":
    main()
```

Parte 2:

3) Seja a seguinte alteração do algoritmo anterior:

Predição(w , x , bias)

ativação = $\text{sigmoid}(w^T * x + \text{bias})$

retorna ativação

Repita (até convergência dos pesos)

Para cada instância (x, y)

bias = bias - $\alpha * \text{gradiente_bias}$

Para cada peso w_k

$w[k] = w[k] - \alpha * \text{gradiente_x}[k]$

Assumindo que, no algoritmo anterior, o cálculo dos gradientes para os pesos seja:

- a) $\text{gradiente_bias} = (y - \hat{y}) * (\text{sigmoid}(w[k] * x[k] + \text{bias}) * (1 - \text{sigmoid}(w[k] * x[k] + \text{bias})))$
- b) $\text{gradiente} = (y - \hat{y}) * (\text{sigmoid}(w[k] * x[k] + \text{bias}) * (1 - \text{sigmoid}(w[k] * x[k] + \text{bias}))) * x[k]$

Crie uma segunda versão do exercício contemplando o algoritmo alterado.

4) Faça uma pesquisa sobre como obter tais gradientes, e explique, com suas palavras, o que deve ser feito para tal.

Referência:

Aurélien Géron. Neural networks and deep learning
(cap. 1)

<https://www.oreilly.com/library/view/neural-networks-and/9781492037354/ch01.html>