

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS  
ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO  
CIÊNCIAS DA COMPUTAÇÃO  
INTELIGÊNCIA ARTIFICIAL  
CLARIMAR JOSE COELHO

ATIVIDADE EXTRA DISCIPLINAR IV

**MARCOS RODOLFO CRUVINEL GOULART QUERINO**

GOIÂNIA  
2020

## INSTRUÇÕES

Escreva um programa em linguagem *Python* para implementar um modelo de rede neural artificial para um *perceptron* de camada única para classificar uma porta lógica or.

## CÓDIGO

```
# Iniciando os valores dos pesos, taxa de aprendizado e degrau
# 'degrau' é a variavel para verificar se o neuronio vai ou nao ativar
peso1= -0.2 # valor aleatorio inicial
peso2= 0.4
tax_aprendizado= 0.2 # o qual rapido quer que o algoritmo aprenda (entre 0 a 1)
degrau= 0 # igual a 0 por padrão, para a porta logica OR
        # se fosse AND, degrau deveria ser igual a 1

# base de dados de treinamento dos possiveis resultados da porta logica OR
treino= [0,1,1,1]

# função Step Function
def stepFunction(num):
    if(num>degrau):
        num= 1 # peso positivo, sinapse excitadora
    elif(num<=degrau):
        num= 0 # peso negativo, sinapse inibidora
    return num

# função de calculo de erro, algoritmo mais simples:
# erro = respostaCorreta - respostaCalculada
def erroCalc(treino, respCalc):
    erro= treino - respCalc
    return erro
```

```

# função de atualização dos pesos

# peso(n+1)= peso(n) + (taxa de aprendizado * entrada * erro)
def atualizaPeso(pesoN, entrada, erro):
    pesoN1= pesoN +(taxa_aprendizado * entrada * erro)
    return pesoN1

# rodando os treinamentos, epoch= 10 (10 treinamentos)
for epoch in range (10):
    print("\nEPOCH N° {}".format(epoch+1))
    it= 0
    # cada treino testa as 4 possibilidades de entradas da porta logica OR

    for entrada1 in range(2): # atribui valores 0 e 1 para entrada1
        for entrada2 in range(2): # atribui valores 0 e 1 para entrada2
            print("\nEntrada 1 = {} , Entrada 2 = {}".format(entrada1,entrada2))
            print("Peso 1 = {} , Peso 2 = {}".format(peso1,peso2))

            # faz somatorio das entradas * pesos
            soma = (peso1*entrada1)+(peso2*entrada2)

            # chama stepFunction para verificar essa sinapse
            soma = stepFunction(soma)

            # stepFunction converte para 1 quaisquer valores maiores que 0
            # e para 0, quaisquer valores menores ou iguais a esse.
            # O valor retornado por stepFunction é a resposta do algoritmo
            # para as entradas 1 e 2 da porta lógica OR
            print("Valor de saída atual = {}".format(soma))

            # imprime o valor correto (valor almejado como resultado)
            print("Valor de saída esperado = {}".format(treino[it]))

            # calcula o erro

```

```

Erro = erroCalc(treino[it], soma)
print("Erro = {}".format(Erro))

# Erro = 0 significa que o algoritmo não errou
# Se errou, chama a função para atualizar os pesos
if(Erro!=0):
    peso1= atualizaPeso(peso1,entrada1,Erro)
    print("Peso 1 atualizado = {}".format(peso1))
    peso2= atualizaPeso(peso2,entrada2,Erro)
    print("Peso 2 atualizado = {}".format(peso2))
    it+= 1

```

## RESULTADOS

Os resultados foram copiados para o presente trabalho em formato de 2 colunas.

### EPOCH N° 1

Entrada 1 = 0 , Entrada 2 = 0

Peso 1 = -0.2 , Peso 2 = 0.4

Valor de saída atual = 0

Valor de saída esperado = 0

Erro = 0

Entrada 1 = 0 , Entrada 2 = 1

Peso 1 = -0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 0

Peso 1 = -0.2 , Peso 2 = 0.4

Valor de saída atual = 0

Valor de saída esperado = 1

Erro = 1

Peso 1 atualizado = 0.0

Peso 2 atualizado = 0.4

Entrada 1 = 1 , Entrada 2 = 1

Peso 1 = 0.0 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

### EPOCH N° 2

Entrada 1 = 0 , Entrada 2 = 0

Peso 1 = 0.0 , Peso 2 = 0.4

Valor de saída atual = 0

Valor de saída esperado = 0

Erro = 0

Entrada 1 = 0 , Entrada 2 = 1

Peso 1 = 0.0 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 0

Peso 1 = 0.0 , Peso 2 = 0.4

Valor de saída atual = 0

Valor de saída esperado = 1

Erro = 1

Peso 1 atualizado = 0.2

Peso 2 atualizado = 0.4

Entrada 1 = 1 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

EPOCH N° 3

Entrada 1 = 0 , Entrada 2 = 0

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 0

Valor de saída esperado = 0

Erro = 0

Entrada 1 = 0 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 0

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

EPOCH N° 4

Entrada 1 = 0 , Entrada 2 = 0

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 0

Valor de saída esperado = 0

Erro = 0

Entrada 1 = 0 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 0  
Peso 1 = 0.2 , Peso 2 = 0.4  
Valor de saída atual = 1  
Valor de saída esperado = 1  
Erro = 0

Entrada 1 = 1 , Entrada 2 = 1  
Peso 1 = 0.2 , Peso 2 = 0.4  
Valor de saída atual = 1  
Valor de saída esperado = 1  
Erro = 0

#### EPOCH N° 5

Entrada 1 = 0 , Entrada 2 = 0  
Peso 1 = 0.2 , Peso 2 = 0.4  
Valor de saída atual = 0  
Valor de saída esperado = 0  
Erro = 0

Entrada 1 = 0 , Entrada 2 = 1  
Peso 1 = 0.2 , Peso 2 = 0.4  
Valor de saída atual = 1  
Valor de saída esperado = 1  
Erro = 0

Entrada 1 = 1 , Entrada 2 = 0  
Peso 1 = 0.2 , Peso 2 = 0.4  
Valor de saída atual = 1  
Valor de saída esperado = 1  
Erro = 0

Entrada 1 = 1 , Entrada 2 = 1  
Peso 1 = 0.2 , Peso 2 = 0.4  
Valor de saída atual = 1  
Valor de saída esperado = 1  
Erro = 0

#### EPOCH N° 6

Entrada 1 = 0 , Entrada 2 = 0  
Peso 1 = 0.2 , Peso 2 = 0.4  
Valor de saída atual = 0  
Valor de saída esperado = 0  
Erro = 0

Entrada 1 = 0 , Entrada 2 = 1  
Peso 1 = 0.2 , Peso 2 = 0.4  
Valor de saída atual = 1  
Valor de saída esperado = 1  
Erro = 0

Entrada 1 = 1 , Entrada 2 = 0  
Peso 1 = 0.2 , Peso 2 = 0.4  
Valor de saída atual = 1  
Valor de saída esperado = 1  
Erro = 0

Entrada 1 = 1 , Entrada 2 = 1  
Peso 1 = 0.2 , Peso 2 = 0.4  
Valor de saída atual = 1  
Valor de saída esperado = 1  
Erro = 0

EPOCH N° 7

Entrada 1 = 0 , Entrada 2 = 0

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 0

Valor de saída esperado = 0

Erro = 0

Entrada 1 = 0 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 0

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

EPOCH N° 8

Entrada 1 = 0 , Entrada 2 = 0

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 0

Valor de saída esperado = 0

Erro = 0

Entrada 1 = 0 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 0

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

EPOCH N° 9

Entrada 1 = 0 , Entrada 2 = 0

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 0

Valor de saída esperado = 0

Erro = 0

Entrada 1 = 0 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 0

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

EPOCH N° 10

Entrada 1 = 0 , Entrada 2 = 0

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 0

Valor de saída esperado = 0

Erro = 0

Entrada 1 = 0 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 0

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

Entrada 1 = 1 , Entrada 2 = 1

Peso 1 = 0.2 , Peso 2 = 0.4

Valor de saída atual = 1

Valor de saída esperado = 1

Erro = 0

## BIBLIOGRAFIA

COELHO, Clarimar Jose. **Inteligência artificial: Redes neurais – Backpropagation**. Sem data de publicação. Disponibilizado pelo autor. Acesso em 02 de dezembro de 2020.

GRANATYR, Jones. **Machine Learning e Data Science com Python de A a Z**. Seção 10: redes neurais artificiais. Udemy. 2020. Disponível em: <https://www.udemy.com/course/machine-learning-e-data-science-com-python-y/>. Acesso em 02 de dezembro de 2020.



LEONEL, Jorge S. **Inteligência Artificial – Perceptron**. Inteligência artificial Brasil. 2019.  
Disponível em: <https://inteligenciaartificialbrasil.wordpress.com/2019/03/17/perceptron/>  
Acesso em 02 de dezembro de 2020.

PALMIERE, Sergio Eduardo. **Rede Perceptron de uma única camada**. Embarcados. 2016.  
Disponível em: <https://www.embarcados.com.br/rede-perceptron-de-uma-unica-camada/>.  
Acesso em 02 de dezembro de 2020.

\_\_\_\_\_. **Rede Neural Perceptron de Camada Única**. Tecnologia Eficiente. 2016.  
Disponível em: <https://www.youtube.com/watch?v=1Q0FYfUIGuY>. Acesso em 02 de dezembro de 2020.