**GABRIEL CARNEIRO GONÇALVES**

**11611ECP013**

**5° TRABALHO DE APRENDIZAGEM DE MÁQUINAS**

**TREINAMENTO ADELINE E ERRO QUADRATICO TOTAL**

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**

**UBERLÂNDIA**

**2023**

Foi realizado o treinamento de uma rede neural Adeline calculando o erro quadrático a cada ciclo e colocando-os em um gráfico. Foram utilizados 200 ciclos para treinar a rede neural, e após o treinamento, podemos perceber q a cada ciclo, o erro quadrático de aproxima cada vez mais de 0, até um ponto que basicamente não tem mais alteração, como podemos ver na seguinte figura.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

A linguagem utilizada para tal treinamento foi Python, que foi escrito o seguinte código.

import matplotlib.pyplot as plt

import numpy as np

x = [[2.215,0.224,0.294,2.327,2.497,0.169,1.274,1.526,2.009,1.759,1.367,2.173,0.856,2.21,1.587,0.35,1.441,0.185,2.764,1.947],

     [2.063,1.586,0.651,2.932,2.322,1.943,2.428,0.596,2.161,0.342,0.938,2.719,1.904,1.868,1.642,0.84,0.09,1.327,1.149,1.598]]

t = [-1,1,1,-1,-1,1,-1,1,-1,1,1,-1,1,-1,-1,1,1,1,-1,-1]

#w\_ant =  np.random.uniform(-0.5,0,5)

w\_ant = 0.5-np.random.rand(1,2)

b\_ant = 0.5-np.random.rand(1)

wnovo = np.zeros((1,2))

bnovo = np.zeros((1))

print(w\_ant)

print(b\_ant)

teta = 0

alfa = 0.05

numciclos = 200

ciclos = 0

erroquadratico = 0

ciclos = ciclos+1

while ciclos<=numciclos:

    erroquadratico=0

    ciclos = ciclos+1

    for i in range(20):

      #y\_liquido = w\_ant[0][0]\*x[0][i]+ w\_ant[0][1]\*x[1][i] + b\_ant[0]

      y\_liquido = w\_ant[0][0]\*x[0][i] + w\_ant[0][1]\*x[1][i] + b\_ant[0]

      # Função de ativação linear

      y = y\_liquido

      # Cálculo do erro quadratico

      erroquadratico = erroquadratico + (t[i]-y)\*\*2

      # Atualização dos pesos

      wnovo[0][0] = w\_ant[0][0] + alfa \* (t[i]-y)\*x[0][i]

      wnovo[0][1] = w\_ant[0][1] + alfa \* (t[i]-y)\*x[1][i]

      bnovo[0] = b\_ant[0] + alfa \* (t[i]-y)

      w\_ant = wnovo

      b\_ant = bnovo

    plt.plot(ciclos,erroquadratico,'ro')

plt.show()

  # Teste da rede treinada

for i in range(20):

  y\_liquido = wnovo[0][0] \* x[0][i] + wnovo[0][1] \* x[1][i] + bnovo[0]

  # Função de ativação para o teste: degrau

  if y\_liquido >= teta:

    y = 1

  else:

    y = -1

  print(i,"Target:",t[i],"Saída:",y)

  print(i,"Target:",t[i],"Saída liquida: ",y\_liquido)

Onde os pesos iniciais e o b foram iniciados de maneira aleatória. Após todo treinamento, foi feita a testagem da rede neural treinada, onde podemos observar que o objetivo foi alcançado com sucesso, como podemos observar a seguir.

Texto

Descrição gerada automaticamente com confiança média

Onde target é a meta desejada, Saída é o que foi obtido com a rede neural após o treinamento e, Saída liquida é o valor calculado pela rede com os pesos e b novo, que deve ser comparado com o teta, para associar um valor a Y (Saída), onde teta foi assumido como 0, se y\_liquido for maior que 0, Saída recebe 1, se não, Saída recebe -1.