Projeto Prático – Redes Neurais – CC01

Aluno: Helton Lucas do Carmo Sousa

1º Questão

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

**int** main**(){**

**double** ErroPrecisao**=**0.000001**;**

**double** EqmAnterior**;**

**double** EqmAtual**;**

**double** EqmFinal**=**0**;**

**double** EQM**=**0**;**

**int** maxInt **=** 5000**;** *//Interacoes*

**int** limiar**=-**1**;** *//limiar de ativação*

**double** w0 **=** **-**limiar**;**

**int** coluna **=** 5**;** *//amostras*

**int** linha **=** 35**;** *//entradas*

**int** tamanho\_x **=**linha**;**

**int** quantelex **=** coluna**;**

**double** w**[**coluna**];** *//pesos*

**int** tam\_w **=** coluna**;**

**int** d**[]={**1**,-**1**,-**1**,-**1**,**1**,**1**,**1**,**1**,-**1**,**1**,-**1**,-**1**,**1**,**1**,-**1**,-**1**,**1**,-**1**,-**1**,**1**,**1**,-**1**,-**1**,**1**,-**1**,**1**,-**1**,**1**,-**1**,**1**,**1**,**1**,-**1**,-**1**,-**1**};** *//desejadas*

**float** taxa\_aprendizagem **=** 0.0025**;**

**int** y **=** 0**;** *//saida*

**double** u **=** 0**;** *//somatorio*

**int** e **=**0**;**

**int** itera**=**0**;**

**int** acertos**[**linha**];**

**int** k **=** 0**;**

**double** classe**[**linha**];**

*//X0 X1 X2 X3*

**double** x **[**linha**][**coluna**]** **=** **{{-**1**,**0.4329**,-**1.3719**,** 0.7022**,-**0.8535**},**

**{-**1**,** 0.3024**,** 0.2286**,** 0.8630**,** 2.7909**},**

**{-**1**,** 0.1349**,** **-**0.6445**,** 1.0530**,** 0.5687**},**

**{-**1**,** 0.3374**,** **-**1.7163**,** 0.3670**,-**0.6283**},**

**{-**1**,** 1.1434**,** **-**0.0485**,** 0.6637**,** 1.2606**},**

**{-**1**,** 1.3749**,** **-**0.5071**,** 0.4464**,** 1.3009**},**

**{-**1**,** 0.7221**,** **-**0.7587**,** 0.7681**,-**0.5592**},**

**{-**1**,** 0.4403**,** **-**0.8072**,** 0.5154**,-**0.3129**},**

**{-**1**,** **-**0.5231**,** 0.3548**,** 0.2538**,**1.5776**},**

**{-**1**,** 0.3255**,** **-**2.0000**,** 0.7112**,-**1.1209**},**

**{-**1**,** 0.5824**,** 1.3915**,** **-**0.2291**,**4.1735**},**

**{-**1**,** 0.1340**,** 0.6081**,** 0.4450**,**3.2230**},**

**{-**1**,** 0.1480**,** **-**0.2988**,** 0.4778**,**0.8649**},**

**{-**1**,** 0.7359**,** 0.1869**,** **-**0.0872**,**2.3584**},**

**{-**1**,** 0.7115**,** **-**1.1469**,** 0.3394**,**0.9573**},**

**{-**1**,** 0.8251**,** **-**1.2840**,** 0.8452**,**1.2382**},**

**{-**1**,** 0.1569**,** 0.3712**,** 0.8825**,**1.7633**},**

**{-**1**,** 0.0033**,** 0.6835**,** 0.5389**,**2.8249**},**

**{-**1**,** 0.4243**,** 0.8313**,** 0.2634**,**3.5855**},**

**{-**1**,** 1.0490**,** 0.1326**,** 0.9138**,**1.9792**},**

**{-**1**,** 1.4276**,** 0.5331**,** **-**0.0145**,**3.7286**},**

**{-**1**,** 0.5971**,** 1.4865**,** 0.2904**,**4.6069**},**

**{-**1**,** 0.8475**,** 2.1479**,** 0.3179**,**5.8235**},**

**{-**1**,** 1.3967**,** **-**0.4171**,** 0.6443**,**1.3927**},**

**{-**1**,** 0.0044**,** 1.5378**,** 0.6099**,**4.7755**},**

**{-**1**,** 0.2201**,** **-**0.5668**,** 0.0515**,**0.7829**},**

**{-**1**,** 0.6300**,** **-**1.2480**,** 0.8591**,**0.8093**},**

**{-**1**,** **-**0.2479**,** 0.8960**,** 0.0547**,**1.7381**},**

**{-**1**,** **-**0.3088**,** **-**0.0929**,** 0.8659**,**1.5483**},**

**{-**1**,** **-**0.5180**,** 1.4974**,** 0.5453**,**2.3993**},**

**{-**1**,** 0.6833**,** 0.8266**,** 0.0829**,**2.8864**},**

**{-**1**,** 0.4353**,** **-**1.4066**,** 0.4207**,-**0.4879**},**

**{-**1**,** **-**0.1069**,** **-**3.2329**,** 0.1856**,-**2.4572**},**

**{-**1**,** 0.4662**,** 0.6261**,** 0.7304**,**3.4370**},**

**{-**1**,** 0.8298**,** **-**1.4089**,** 0.3119**,**1.3235**}** **};**

*//amostras de sinais*

*//double x [linha][coluna] = {{-1, 0.9694, 0.6909, 0.4334, 3.49651},*

*// {-1, 0.5427, 1.3832, 0.6390, 4.0352},*

*// {-1, 0.6081, -0.9196, 0.5925, 0.1016} ,*

*// {-1, -0.1618, 0.4694, 0.2030, 3.0117},*

*// {-1, 0.1870, -0.2578, 0.6124, 1.7749},*

*// {-1, 0.4891, -0.5276, 0.4378, 0.6439},*

*// {-1, 0.3777, 2.0149, 0.7423, 3.3932},*

*// {-1, 1.1498, -0.4067, 0.2469, 1.5866},*

*// {-1, 0.9325, 1.0950, 1.0359, 3.3591},*

*// {-1, 0.5060, 1.3317, 0.9222, 3.7174},*

*// {-1, 0.0497, -2.0656, 0.6124, -0.6585},*

*// {-1, 0.40004, 3.5369, 0.9766, 5.3532},*

*// {-1, -0.1874, 1.3343, 0.5373, 3.2189},*

*// {-1, 0.5060, 1.3317, 0.9222, 3.7174},*

*// {-1, 1.6375, -0.7911, 0.7537, 0.5515}}*

*// iniciar wights entre 0 e 1*

printf**("\nPESO INICIAIS:\n"** **);**

**for** **(int** i **=** 0**;** i **<** coluna**;** i**++){**

*// w[i] = rand() % 101;*

*// w[i]=w[i]/100;*

w**[**i**]=** **(float)(**rand**())/(float)(**RAND\_MAX**)\***1.0**;** *//gerar numero randômico de 0 a 1*

printf**("Peso %d: %.4f ",** i**+**1**,** w**[**i**]);**

**}**

**for(int** t**=**0**;** t**<**linha**;**t**++){**

**for(int** j**=**0**;**j**<**quantelex**;**j**++){**

u **+=** x**[**t**][**j**]\***w**[**j**];**

EQM**+=** pow**((**d**[**j**]-**u**),**2**);**

**}**

**}**

u**=**0**;** *//somatorio*

EqmAtual**=** EQM**/**linha**;**

*// printf("\n EQM Atual: %.4f",EqmAtual);*

EQM**=**0**;**

**do{**

EqmAnterior**=** EqmAtual**;**

**for(int** t**=**0**;** t**<**linha**;**t**++){**

u**=**0**;**

**for(int** j**=**0**;**j**<**quantelex**;**j**++){**

u **+=** x**[**t**][**j**]\***w**[**j**];**

w**[**j**]** **+=** taxa\_aprendizagem**\*(**d**[**t**]-**u**)\***x**[**t**][**j**];** *//Atualizar pesos*

**}**

**}**

k**++;**

EQM**=**0**;**

**for(int** t**=**0**;** t**<**linha**;**t**++){**

**for(int** j**=**0**;**j**<**quantelex**;**j**++){**

u **+=** x**[**t**][**j**]\***w**[**j**];**

EQM**+=** pow**((**d**[**j**]-**u**),**2**);**

**}**

**}**

u **=** 0**;**

EqmAtual**=** EQM**/**linha**;**

printf**("\nEQM atual: %f",**EqmAtual**);**

EQM**=**0**;**

**if((**EqmAnterior**-**EqmAtual**)<**0**){**

EqmFinal **=(**EqmAnterior**-**EqmAtual**)\*-**1**;**

**}else{**

EqmFinal **=**EqmAnterior**-**EqmAtual**;**

**}**

printf**("\nEQM Final: %.4f",**EqmFinal**);**

**}**

**while(**EqmFinal**>**ErroPrecisao**);**

**for(int** t**=**0**;** t**<**linha**;**t**++){**

u**=**0**;**

**for(int** j**=**0**;**j**<**quantelex**;**j**++){**

u **+=** x**[**t**][**j**]\***w**[**j**];**

**}**

**if(**u**>=**0**){**

y**=**1**;**

**}else{**

y**=-**1**;**

**}**

**if(**y**==**1**){**

classe**[**t**]=**0**;**

**}else{**

classe**[**t**]=**1**;**

**}**

**}**

**for(int** z **=** 0 **;** z **<** linha **;** z**++){**

printf**("%.1f\n",** classe**[**z**]);**

**}**

**for(int** j**=**0**;**j**<**tam\_w**;**j**++){**

printf**("\nPesos finais[%d]: %f",** j**+**1**,** w**[**j**]);**

**}**

printf**("\nRede Treinada : Interacoes [%d]",**k**);**

**}**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Treinamento | Vetor de pesos iniciais | | | | | Vetor de pesos finais | | | | | Número de épocas | |
| w0 | w1 | w2 | w3 | w4 | w0 | w1 | w2 | w3 | w4 |  |
| 1ºT | 0.8402 | 0.3944 | 0.7831 | 0.7984 | 0.9116 | 0.03044 | 0.2850 | -0.0323 | -0.2571 | -0.0819 | 524 |
| 2ºT | 0.0840 | 0.0394 | 0.0783 | 0.0798 | 0.0912 | 0.0304 | 0.2850 | -0.0323 | -0.2571 | -0.0819 | 527 |
| 3ºT | 0.1680 | 0.0789 | 0.1566 | 0.1597 | 0.1823 | 0.0304 | 0.2850 | -0.0323 | -0.2571 | -0.0819 | 527 |
| 4ºT | 0.0924 | 0.0434 | 0.0861 | 0.0878 | 0.1003 | 0.0304 | 0.2850 | -0.0323 | -0.2571 | -0.0819 | 527 |
| 5ºT | 0.0933 | 0.0438 | 0.0869 | 0.0886 | 0.1012 | 0.0304 | 0.2850 | -0.0323 | -0.2571 | -0.0819 | 527 |

2º Questão

3º Questão

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Treinamento | Vetor de pesos iniciais | | | | | Vetor de pesos finais | | | | | Número de épocas | |
| w0 | w1 | w2 | w3 | w4 | w0 | w1 | w2 | w3 | w4 |  |
| 1ºT | 0.8402 | 0.3944 | 0.7831 | 0.7984 | 0.9116 | 0.03044 | 0.2850 | -0.0323 | -0.2571 | -0.0819 | 524 |
| 2ºT | 0.0840 | 0.0394 | 0.0783 | 0.0798 | 0.0912 | 0.0304 | 0.2850 | -0.0323 | -0.2571 | -0.0819 | 527 |

Linearmente separáveis não são, devido convergir somente para algumas amostras e não para todas. Mas a aproximação do EQM é o que permite essa classificação se tornar possível a divisão do problema em classes.

4º Questão

A = 0.0; B=1.0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Y T1 | Y T2 | Y T1 | Y T1 | Y T1 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

Não mudam porque a rede é levada a um estado estável, onde estes valores não mudam mais (Estático e Off-line). A não ser que novas amostras sejam inseridas em um possível treinamento.