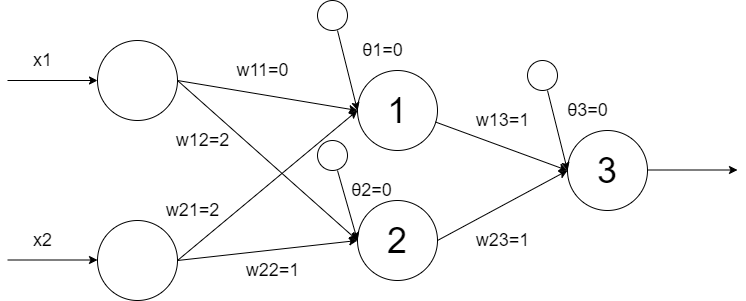
实验二 神经网络实验

姓名：唐川淇 学号：1131190111

# 问题重述

有一BP神经网络，学习参数，激活函数，输入样本，节点最终的期望输出为1，当前的权值与阈值如图所示，求本次训练之后新的权值与阈值。



# 问题的求解

## 计算神经元的输出

首先使用MATLAB建立神经网络，根据题目要求先设定好学习率、输入节点个数、隐藏层节点个数、输出层节点个数。其中输入层到隐藏层权重、隐藏层到输出层权重、三个偏置都可以用矩阵表示。代码如下：

|  |
| --- |
| 初始化 |
| l=1; %学习率  InputNodeNum=2; %输入节点  HiddenNodeNum=2; %隐藏节点  OutputNodeNum=1; %输出节点  W\_IH=zeros(InputNodeNum,HiddenNodeNum); %输入层、隐藏层权重  W\_HO=zeros(HiddenNodeNum,OutputNodeNum); %隐藏层、输出层权重  theta=zeros(3,1); %偏置  W\_IH=[0,2;2,1]; %初始化权重  W\_HO=[1;1];  theta=[0,0,0]; |

设定好参数之后需要继续利用MATLAB计算输出，根据题目中的输入，计算最终节点3的输出

|  |
| --- |
| 计算输出 |
| O=zeros(3);  H=X\*W\_IH;  H=O\_fun(H)+[theta(1),theta(2)];  y=H\*W\_HO;  O(3)=O\_fun(y)+theta(3); |

每个神经元的输出结果如下：

|  |  |  |
| --- | --- | --- |
| 神经元 | 总输入 | 输出 |
| ① | u1 = 0\*1+2\*0+0 = 0 | O1 = 0.5 |
| ② | u2 = 2\*1+1\*0+0 = 2 | O2 = 0.881 |
| ③ | u3 = 1\*0.5+1\*0.881+0 = 1 | O3 = 0.799 |

## 计算误差

|  |
| --- |
| 计算误差 |
| O(1)=H(1);O(2)=H(2);  E(3)=O(3)\*(1-O(3))\*(T3-O(3));  E(2)=O(2)\*(1-O(2))\*W\_HO(2)\*E(3);  E(1)=O(1)\*(1-O(1))\*W\_HO(1)\*E(3); |

计算每个神经元的误差如下：

|  |  |
| --- | --- |
| 神经元 | 误差 |
| ③ | E3 = O3(1-O3)(T3-O3) = 0.799\*(1-0.799)\*(1-0.799) = 0.032 |
| ② | E2 = O2(1-O2) w23E3 = 0.881\*(1-0.881)\*1\*0.032 = 0.003 |
| ① | E1 = O1(1-O1) w13E3 = 0.5\*(1-0.5)\*1\*0.032 = 0.008 |

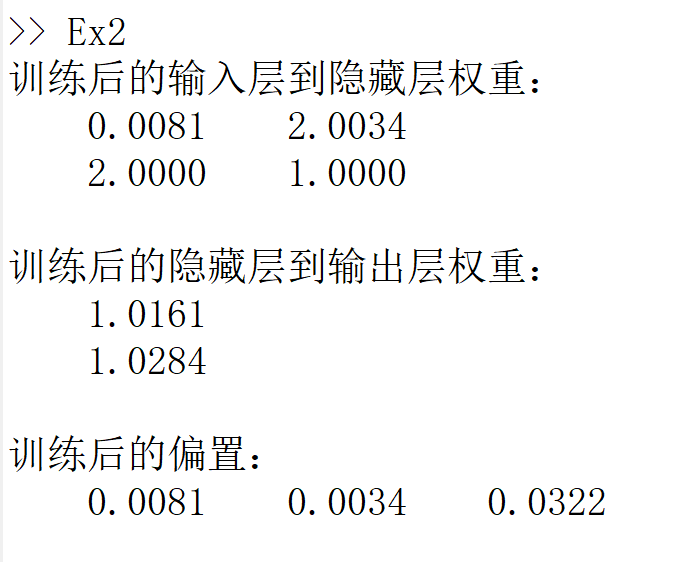
## 调整权值与阈值

|  |
| --- |
| 调整权重 |
| for i=1:2  for j=1:2  W\_IH(i,j)=W\_IH(i,j)+l\*X(i)\*E(j);  end  end  for i=1:2  W\_HO(i)=W\_HO(i)+l\*O(i)\*E(3);  end  for i=1:3  theta(i)=theta(i)+l\*E(i);  end |

调整权值与阈值结果如下：

|  |  |
| --- | --- |
| 神经元 | 更新值 |
| w11 | w11+l\* x1 E1 = 0+1\*0.008 = 0.008 |
| w12 | w12+l\* x1 E2 = 2+1\*0.003 = 2.003 |
| w21 | w21+l\* x2 E1 = 2+1\*0 = 2 |
| w22 | w22+l\* x2 E2 = 1+1\*0 = 1 |
| w13 | w13+l\* O1 E3 = 1+1\*0.5\*0.032 = 1.016 |
| w23 | w23+l\* O2 E3 = 1+1\*0.881\*0.032 = 1.028 |
| θ1 | θ1+l\* E1 = 0+1\*0.008 = 0.008 |
| θ2 | θ2+l\* E2 = 0+1\*0.003 = 0.003 |
| θ3 | θ3+l\* E3 = 0+1\*0.032 = 0.032 |

## 输出结果



附录

|  |
| --- |
| 附录 |
| 介绍：全代码 |
| l=1; %学习率  InputNodeNum=2; %输入节点  HiddenNodeNum=2; %隐藏节点  OutputNodeNum=1; %输出节点    W\_IH=zeros(InputNodeNum,HiddenNodeNum); %输入层、隐藏层权重  W\_HO=zeros(HiddenNodeNum,OutputNodeNum); %隐藏层、输出层权重  theta=zeros(3,1); %偏置    W\_IH=[0,2;2,1]; %初始化权重  W\_HO=[1;1];  theta=[0,0,0];    x=[1,0];T3=1; %样本  [W\_IH,W\_HO,theta]=net\_train(x,W\_IH,W\_HO,theta,l,T3); %训练    disp("训练后的输入层到隐藏层权重：")  disp(W\_IH)  disp("训练后的隐藏层到输出层权重：")  disp(W\_HO)  disp("训练后的偏置：")  disp(theta)    %激活函数  function y=O\_fun(x)  y=1./(1+exp(-x));  end    %训练函数  function [W\_IH,W\_HO,theta]=net\_train(X,W\_IH,W\_HO,theta,l,T3)  O=zeros(3);  E=zeros(3);    H=X\*W\_IH;  H=O\_fun(H)+[theta(1),theta(2)];  y=H\*W\_HO;  O(3)=O\_fun(y)+theta(3);    O(1)=H(1);O(2)=H(2);  E(3)=O(3)\*(1-O(3))\*(T3-O(3));  E(2)=O(2)\*(1-O(2))\*W\_HO(2)\*E(3);  E(1)=O(1)\*(1-O(1))\*W\_HO(1)\*E(3);    for i=1:2  for j=1:2  W\_IH(i,j)=W\_IH(i,j)+l\*X(i)\*E(j);  end  end  for i=1:2  W\_HO(i)=W\_HO(i)+l\*O(i)\*E(3);  end  for i=1:3  theta(i)=theta(i)+l\*E(i);  end  end |