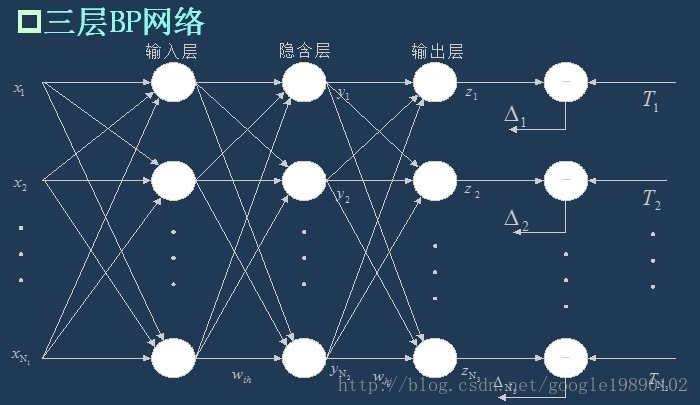
# BP神经网络

来源：

[https://blog.csdn.net/google19890102/article/details/32723459#reply](https://blog.csdn.net/google19890102/article/details/32723459%23reply)

## 一、BP神经网络的概念

BP神经网络是一种多层的前馈神经网络，其主要的特点是：信号是前向传播的，而误差是反向传播的。具体来说，对于如下的只含一个隐层的神经网络模型：



(三层BP神经网络模型)

BP神经网络的过程主要分为两个阶段，第一阶段是信号的前向传播，从输入层经过隐含层，最后到达输出层；第二阶段是误差的反向传播，从输出层到隐含层，最后到输入层，依次调节隐含层到输出层的权重和偏置，输入层到隐含层的权重和偏置。

## 二、BP神经网络的流程

在知道了BP神经网络的特点后，我们需要依据信号的前向传播和误差的反向传播来构建整个网络。

### 1、网络的初始化

假设输入层的节点个数为http://latex.codecogs.com/gif.latex?n，隐含层的节点个数为http://latex.codecogs.com/gif.latex?l，输出层的节点个数为http://latex.codecogs.com/gif.latex?m。输入层到隐含层的权重http://latex.codecogs.com/gif.latex?\omega_%7bij%7d，隐含层到输出层的权重为http://latex.codecogs.com/gif.latex?\omega_%7bjk%7d，输入层到隐含层的偏置为http://latex.codecogs.com/gif.latex?a_j，隐含层到输出层的偏置为http://latex.codecogs.com/gif.latex?b_k。学习速率为http://latex.codecogs.com/gif.latex?\eta，激励函数为http://latex.codecogs.com/gif.latex?g\left&space;(&space;x&space;\right&space;)。其中激励函数为http://latex.codecogs.com/gif.latex?g\left&space;(&space;x&space;\right&space;)取Sigmoid函数或者线性函数。Sigmoid函数形式为：

http://latex.codecogs.com/gif.latex?g\left&space;(&space;x&space;\right&space;)=\frac%7b1%7d%7b1+e%5e%7b-x%7d%7d

### 2、隐含层的输出

如上面的三层BP网络所示，隐含层的输出http://latex.codecogs.com/gif.latex?H_j为

http://latex.codecogs.com/gif.latex?H_j=g\left&space;(&space;\sum_%7bi=1%7d%5e%7bn%7d\omega&space;_%7bij%7dx_i+a_j&space;\right&space;)

### 3、输出层的输出

http://latex.codecogs.com/gif.latex?O_k=\sum_%7bj=1%7d%5e%7bl%7dH_j\omega&space;_%7bjk%7d+b_k

### 4、误差的计算

我们取误差公式为：

http://latex.codecogs.com/gif.latex?E=\frac%7b1%7d%7b2%7d\sum_%7bk=1%7d%5e%7bm%7d\left&space;(&space;Y_k-O_k&space;\right&space;)%5e2

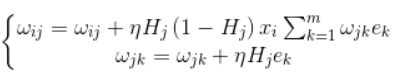
其中http://latex.codecogs.com/gif.latex?Y_k为期望输出。我们记http://latex.codecogs.com/gif.latex?Y_k-O_k=e_k，则http://latex.codecogs.com/gif.latex?E可以表示为

http://latex.codecogs.com/gif.latex?E=\frac%7b1%7d%7b2%7d\sum_%7bk=1%7d%5e%7bm%7de_k%5e2

以上公式中，http://latex.codecogs.com/gif.latex?i=1\cdots&space;n，http://latex.codecogs.com/gif.latex?j=1\cdots&space;l，http://latex.codecogs.com/gif.latex?k=1\cdots&space;m。

### 5、权值的更新

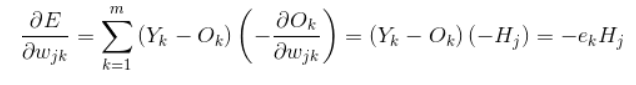
权值的更新公式为：



**这里需要解释一下公式的由来：**

这是误差反向传播的过程，我们的目标是使得误差函数达到最小值，即http://latex.codecogs.com/gif.latex?min&space;E，我们使用梯度下降法：

* 隐含层到输出层的权重更新



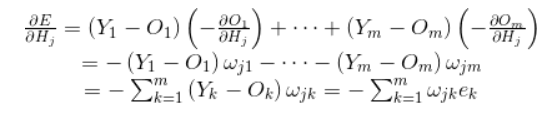
则权重的更新公式为：

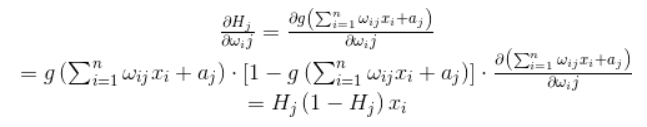
http://latex.codecogs.com/gif.latex?w_%7bjk%7d=w_%7bjk%7d+\eta&space;H_je_k

* 输入层到隐含层的权重更新

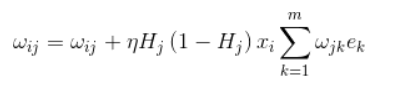
http://latex.codecogs.com/gif.latex?\frac%7b\partial&space;E%7d%7b\partial&space;w_%7bij%7d%7d=\frac%7b\partial&space;E%7d%7b\partial&space;H_j%7d\cdot&space;\frac%7b\partial&space;H_j%7d%7b\partial&space;\omega&space;_%7bij%7d%7d

其中



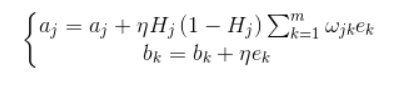


则权重的更新公式为：



### 6、偏置的更新

偏置的更新公式为：



* 隐含层到输出层的偏置更新

http://latex.codecogs.com/gif.latex?\frac%7b\partial&space;E%7d%7b\partial&space;b_k%7d=\left&space;(&space;Y_k-O_k&space;\right&space;)\left&space;(&space;-\frac%7b\partial&space;O_k%7d%7b\partial&space;b_k%7d&space;\right&space;)=-e_k

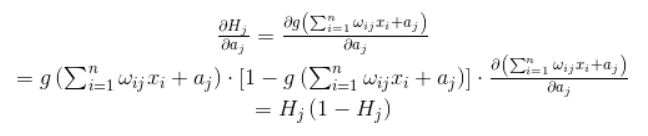
则偏置的更新公式为：

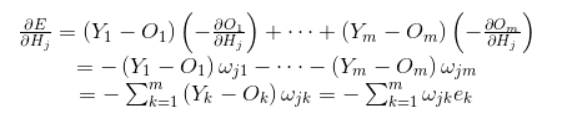
http://latex.codecogs.com/gif.latex?b_k=b_k+\eta&space;e_k

* 输入层到隐含层的偏置更新

http://latex.codecogs.com/gif.latex?\frac%7b\partial&space;E%7d%7b\partial&space;a_j%7d=\frac%7b\partial&space;E%7d%7b\partial&space;H_j%7d\cdot&space;\frac%7b\partial&space;H_j%7d%7b\partial&space;a_j%7d

其中





则偏置的更新公式为：

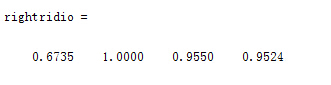
http://latex.codecogs.com/gif.latex?a_k=a_k+\eta&space;H_j\left&space;(&space;1-H_j&space;\right&space;)\sum_%7bk=1%7d%5e%7bm%7d\omega&space;_%7bjk%7de_k

### 7、判断算法迭代是否结束

有很多的方法可以判断算法是否已经收敛，常见的有指定迭代的代数，判断相邻的两次误差之间的差别是否小于指定的值等等。

## 三、实验的仿真

在本试验中，我们利用BP神经网络处理一个四分类问题，最终的分类结果为：



## 附录

MATLAB代码

### 1、主程序

%% BP的主函数

% 清空

clear all;

clc;

% 导入数据

load data;

%从1到2000间随机排序

k=rand(1,2000);

[m,n]=sort(k);

%输入输出数据

input=data(:,2:25);

output1 =data(:,1);

%把输出从1维变成4维

for i=1:2000

switch output1(i)

case 1

output(i,:)=[1 0 0 0];

case 2

output(i,:)=[0 1 0 0];

case 3

output(i,:)=[0 0 1 0];

case 4

output(i,:)=[0 0 0 1];

end

end

%随机提取1500个样本为训练样本，500个样本为预测样本

trainCharacter=input(n(1:1600),:);

trainOutput=output(n(1:1600),:);

testCharacter=input(n(1601:2000),:);

testOutput=output(n(1601:2000),:);

% 对训练的特征进行归一化

[trainInput,inputps]=mapminmax(trainCharacter');

%% 参数的初始化

% 参数的初始化

inputNum = 24;%输入层的节点数

hiddenNum = 50;%隐含层的节点数

outputNum = 4;%输出层的节点数

% 权重和偏置的初始化

w1 = rands(inputNum,hiddenNum);

b1 = rands(hiddenNum,1);

w2 = rands(hiddenNum,outputNum);

b2 = rands(outputNum,1);

% 学习率

yita = 0.1;

%% 网络的训练

for r = 1:30

E(r) = 0;% 统计误差

for m = 1:1600

% 信息的正向流动

x = trainInput(:,m);

% 隐含层的输出

for j = 1:hiddenNum

hidden(j,:) = w1(:,j)'\*x+b1(j,:);

hiddenOutput(j,:) = g(hidden(j,:));

end

% 输出层的输出

outputOutput = w2'\*hiddenOutput+b2;

% 计算误差

e = trainOutput(m,:)'-outputOutput;

E(r) = E(r) + sum(abs(e));

% 修改权重和偏置

% 隐含层到输出层的权重和偏置调整

dw2 = hiddenOutput\*e';

db2 = e;

% 输入层到隐含层的权重和偏置调整

for j = 1:hiddenNum

partOne(j) = hiddenOutput(j)\*(1-hiddenOutput(j));

partTwo(j) = w2(j,:)\*e;

end

for i = 1:inputNum

for j = 1:hiddenNum

dw1(i,j) = partOne(j)\*x(i,:)\*partTwo(j);

db1(j,:) = partOne(j)\*partTwo(j);

end

end

w1 = w1 + yita\*dw1;

w2 = w2 + yita\*dw2;

b1 = b1 + yita\*db1;

b2 = b2 + yita\*db2;

end

end

%% 语音特征信号分类

testInput=mapminmax('apply',testCharacter',inputps);

for m = 1:400

for j = 1:hiddenNum

hiddenTest(j,:) = w1(:,j)'\*testInput(:,m)+b1(j,:);

hiddenTestOutput(j,:) = g(hiddenTest(j,:));

end

outputOfTest(:,m) = w2'\*hiddenTestOutput+b2;

end

%% 结果分析

%根据网络输出找出数据属于哪类

for m=1:400

output\_fore(m)=find(outputOfTest(:,m)==max(outputOfTest(:,m)));

end

%BP网络预测误差

error=output\_fore-output1(n(1601:2000))';

k=zeros(1,4);

%找出判断错误的分类属于哪一类

for i=1:400

if error(i)~=0

[b,c]=max(testOutput(i,:));

switch c

case 1

k(1)=k(1)+1;

case 2

k(2)=k(2)+1;

case 3

k(3)=k(3)+1;

case 4

k(4)=k(4)+1;

end

end

end

%找出每类的个体和

kk=zeros(1,4);

for i=1:400

[b,c]=max(testOutput(i,:));

switch c

case 1

kk(1)=kk(1)+1;

case 2

kk(2)=kk(2)+1;

case 3

kk(3)=kk(3)+1;

case 4

kk(4)=kk(4)+1;

end

end

%正确率

rightridio=(kk-k)./kk

### 2、****激活函数****

%% 激活函数

function [ y ] = g( x )

y = 1./(1+exp(-x));

end