数据挖掘——构建分类器

一、小组成员

杨林青

白艺冲 1000010160

于晨 1000012770

二、任务分工

杨林青：

白艺冲： 神经网络

于晨： 构建分类器以区分不同的iris品种

三、分类器构建

1、思路

构建神经网络，将花的数据作为输入，品种分类作为输出，训练神经网络，得出分类器，进行分类。

2、数据预处理及读入

将字符串形式的品种转换为品种Index。之后正常读入matlab。

取85%作为训练集，15%作为测试集。

file = fopen('classdata.txt');

data = fscanf(file, '%f,%f,%f,%f,%d', [5, inf]);

trainIndicator = rand(1, size(data, 2));

trainPortion = 0.15;

input = data(1:4,trainIndicator > trainPortion);

output = data(1:4,trainIndicator > trainPortion);

3、关键代码

全部代码见”train.m” 及 “predict.m”

a. 初始化，迭代训练神经网络，固定次数以及误差阈限

weight = cell(depth);

threshold = cell(depth);

for k = 1 : depth

weight{k} = rand(levelNum(k+1), levelNum(k)) - 2 \* rand(levelNum(k+1) , levelNum(k));

threshold{k} = rand(levelNum(k+1) , 1) - 2 \* rand(levelNum(k+1) , 1);

end

while(runCount < 100000 && sumMSE > minError)

sumMSE = 0; % sum of MSE

for i = 1 : sampleNum % sample loop

……

end

sumMSE = sqrt(sumMSE) / sampleNum;

end

b.计算网络激活值

% calculat the network

for k = 2 : depth

netValue{k} = weight{k-1} \* netValue{k-1} + threshold{k-1}; %calculate each layer

netValue{k} = 1 ./ (1 + exp(-netValue{k})); %apply logistic function

end

netValue{depth+1} = weight{depth} \* netValue{depth} + threshold{depth}; %output layer

c.反向传播

% update the weights

e = output(:,i) - netValue{depth + 1}; %calc error

assistS{depth} = diag(ones(size(netValue{depth+1})));

s{depth} = -2 \* assistS{depth} \* e;

for k = depth - 1 : -1 : 1

assistS{k} = diag((1-netValue{k+1}).\*netValue{k+1});

s{k} = assistS{k} \* weight{k+1}' \* s{k+1};

end

d.引入动量项，增强训练结果

for k = 1 : depth

dW = (1-mAlpha) \* s{k} \* netValue{k}' + mAlpha \* oldDW{k};

dT = (1-mAlpha) \* s{k} + mAlpha \* oldDT{k};

oldDW{k} = dW; oldDT{k} = dT;

weight{k} = weight{k} - afa \* dW;

threshold{k} = threshold{k} - afa \* dT;

end

4、测试分类器准确度

% test output

for i = 1 : length(x)

netValue{1} = x(:,i);

for k = 2 : depth

netValue{k} = weight{k-1} \* netValue{k-1} + threshold{k-1};

netValue{k} = 1 ./ ( 1 + exp(-netValue{k}));

end

y(:, i) = weight{depth} \* netValue{depth} + threshold{depth};

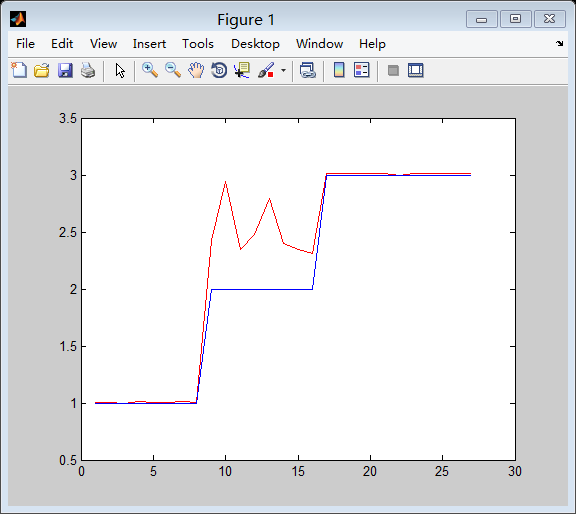
end

% predict

class = round(y);

correct = (class==z);

由于每次训练有随机因素，所以每次结果不同。如下是某次训练结果：



>> predict

correct :25 in 27 test samples

correct ratio: 0.92593

四、代码使用方法

将predict.m第一句load('HW\_train.mat');去掉注释，运行。此时可以读入预训练好的结果，并给出最终评测。

训练请执行train.m，之后将predict.m第一句注释掉，将评测新训练的结果。