Neural Networks Projects

Cancer Detection

Tom Makkink

2018

**1 Introduction**

Cancer is a group of diseases that involve irregular cell growth, that may invade or spread to the other parts of the body. Breast cancer develops from breast tissue, and the first noticeable symptoms are usually a lump in the breast tissue. Breast cancer is most commonly diagnosed by a biopsy of the affected area of the breast.

Serum proteomic pattern diagnostics is a technology that has the potential to improve clinical diagnostics tests for cancer. We would like to assist researchers in cancer diagnostics by constructing a neural network that can identify whether a patient has cancer based on protein profiles.

**2 The Data**

The data set was obtained from the UCL machine learning dataset: “Wisconsin breast cancer database”, and is arranged as follows:

9 input attributes, with values on a scale from 1 – 10, and one binary output (benign = 2, malignant = 4). The values are then scaled to a 0…1 range.

Class distribution:

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Benign | Malign | Total |
| Total Number | 458 | 241 | 699 |
| Total % | 66 | 34 | 100 |

% Import the data

D = importdata('cancerdata.txt');

% Write to a excel file

xlswrite('\cancerdata.xls',D,1);

An analysis of the data in the excel file revealed that there were 16 missing values for attribute 6. This problem was solved by encoding the missing values as 0.3, the average value of that attribute was around 3.5.

**3 Method**

We load and organise the data, and then convert the targets from 2 (benign) to 1 and 4 (malignant) to -1:

% Import the data

D = importdata('cancerdata.txt');

% Import patterns and Targets

p=xlsread('cancerdata.xls','B4:J702');

t=xlsread('cancerdata.xls','K4:K702');

% Organise data

p = p';

t = t';

%change 2(benign) to 1 and 4(malignant) to -1

t(t==2)=1;

t(t==4)=-1;

m=size(p,2);

The neural net used in cancer\_train was a feed-forward network and had the following structure:

* Two hidden layers with 20 neurons in each
* The data was divided into 60% training set, 20% test set and 20% validation set
* The transfer functions used were tansig in the hidden layers and purelin in the output layer
* The Levenberg-Marquardt back-propagation algorithm (trainlim) was used in the training method.
* The gradient descent with momentum was used as the learning function.

%----------------------------------------------------------------------

% Set up and propagate the network

%----------------------------------------------------------------------

% Number of neurons in each layer

s1 = 20;

s2 = 20;

% Deploy the net

net = feedforwardnet([s1,s2]);

% Display net

display(net);

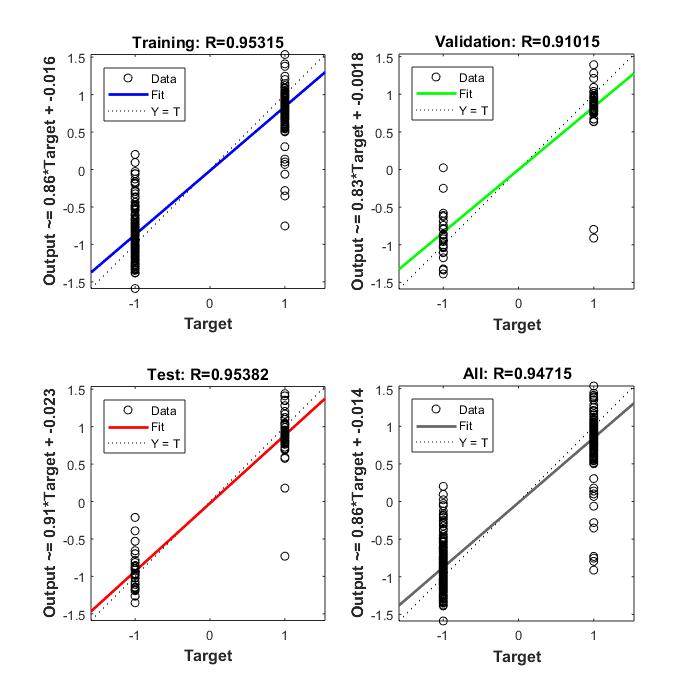
% Set training ratios

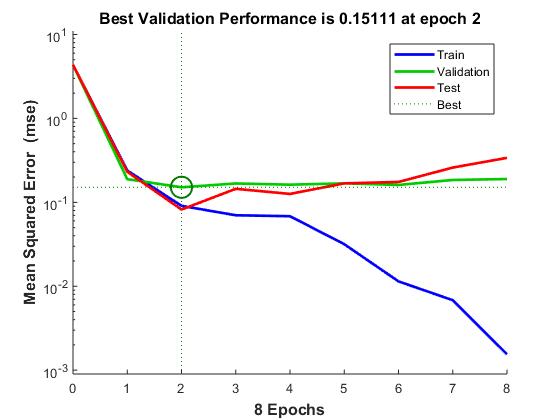
[ptrain,pval,ptest,trainInd,valInd,testInd] = dividerand(p,0.6,0.2,0.2);

[ttrain,tval,ttest] = divideind(t,trainInd,valInd,testInd);

net = init(net);

[net,tr] = train(net, p, t);





Now that we have trained the net, we can test it:

% Training set

ttrain=t(:,tr.trainInd);

ptrain=p(:,tr.trainInd);

atrain=sim(cancerNet,ptrain);

% Test set

ttest=t(:,tr.testInd);

ptest=p(:,tr.testInd);

atest=sim(cancerNet,ptest);

% Validation set

tval=t(:,tr.valInd);

pval=p(:,tr.valInd);

aval=sim(cancerNet,pval);

The diagnosis was made using the hardlim function:

% Make diagnosis

atest=hardlims(atest);

[atest' ttest'];

The performance was calculated by accessing what percentage the net diagnosed correctly:

% Find accuracy

% Number correct

nc=sum(atest==ttest);

% Percentage correct

pc=nc/length(ttest)\*100;

The final percentage accuracy was 95.13%, which is a very good.

**4 Conclusion**

The correct classification percentage of 95.13% is very good, and indicates that Serum proteomic pattern diagnostics could be a very useful technology in assisting the diagnosis of breast cancer in patients.

**5 References**

*https://www.mathworks.com/help/nnet/examples/cancer-detection.html*