Neural Networks Projects

Body Fat Percentage Estimation

Tom Makkink

2018

**1 Introduction**

Body fat percentage is a topic that has received much limelight in the media in recent years. Although, it has long been established that body fat in humans is essential to maintaining life and reproductive health, the rise of careers such as bodybuilding and modelling paralleled with a dramatic increase in obesity, has created a demand for a viable method of accurately determining this value.

We would like to construct a neural network that assists in the calculation of body fat percentage, based on the input of 13 physical attributes.

**2 The Data**

The data was acquired from the Matlab Neural Network Toolbox, and includes 13 physical attributes which map to a target value of a body fat percentage. The physical attributes are as follows:

* Age (years)
* Weight (lbs)
* Height (inches)
* Neck circumference (cm)
* Chest circumference (cm)
* Abdomen 2 circumference (cm)
* Hip circumference (cm)
* Thigh circumference (cm)
* Knee circumference (cm)
* Ankle circumference (cm)
* Biceps (extended) circumference (cm)
* Forearm circumference (cm)
* Wrist circumference (cm)

The data set contains 252 entries, and was imported as follows:

% Import the data

[p,t] = bodyfat\_dataset;

p is a 13 x 252 double matrix, while t is 1 x 252, representing 252 physical characters and their associated body fat percentage.

**3 Method**

The neural net used in body\_fat\_estimation\_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 tansig in the output layer
* The Bayesian regularization training algorithm (trainbr) was used in the training method, as it performed better than the default Levenberg-Marquardt algorithm. It was also found that setting a maximum epoch count of 500, performed just as well as the default value of 1000.
* The gradient descent with momentum was used as the learning function.

% Number of neurons in hidden layers

s1 = 20;

s2 = 20;

% Set up net

net = fitnet([s1, s2], 'trainbr');

% Set maximum number of epochs

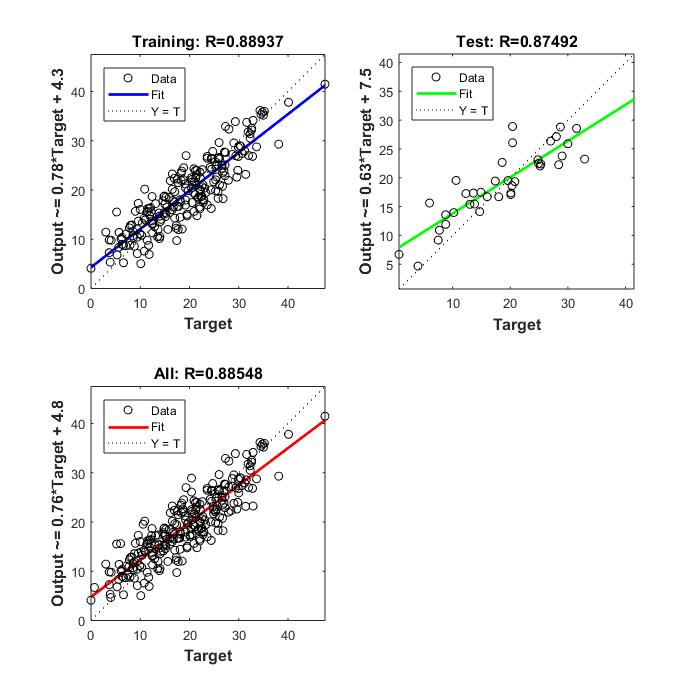
net.trainParam.epochs = 500;

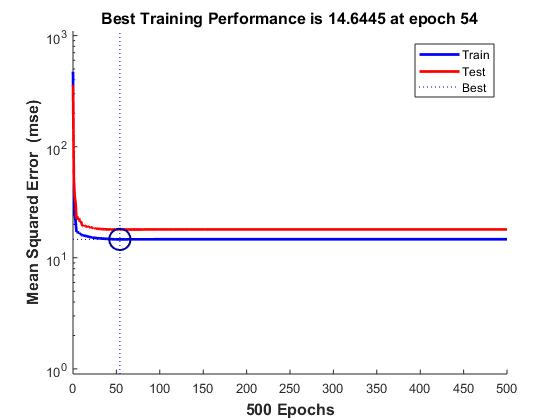
net.layers{1}.transferFcn = 'tansig';

net.layers{2}.transferFcn = 'tansig';

net.layers{3}.transferFcn = 'tansig';

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





Once we had successfully trained the net, it was tested:

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

% Load the data and set up training indices

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

load bodyFat.mat

% Training set

ttrain=t(:,tr.trainInd);

ptrain=p(:,tr.trainInd);

atrain=sim(bodyFatNet,ptrain);

% Test set

ttest=t(:,tr.testInd);

ptest=p(:,tr.testInd);

atest=sim(bodyFatNet,ptest);

% Validation set

tval=t(:,tr.valInd);

pval=p(:,tr.valInd);

aval=sim(bodyFatNet,pval);

%%

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

% Access performance

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

[r2train Rtrain]= correlation(atrain, ttrain)

[r2test Rtest]=correlation(atest,ttest)

% Compare:

[atrain' ttrain'];

% Compare

[atest' ttest'];

The final values were as follows:

r2train = 0.8032

Rtrain = 1.0000

r2test = 0.7407

Rtest = 0.9993

In addition to the training and test functions, a function was created to take in 13 attributes, and output a body fat estimation, using the following script:

function bodyFatPercentage = body\_fat\_fcn (x)

load bodyFat.mat

bodyFatPercentage = sim(bodyFatNet,x)

end

**4 Conclusion**

The final results were good, proving that this neural network could provide a reasonable approximation of body fat percentage based on the 13 input attributes. It must be noted, however, that these estimates are an approximation, and in testing the output of the body fat function to the actual target value, it was found that they are on average between 1% and 2% off. It is likely that using a larger dataset to train the network would improve performance.

**5 References**

[*https://www.mathworks.com/help/nnet/examples/body-fat-estimation.html*](https://www.mathworks.com/help/nnet/examples/body-fat-estimation.html)

*https://en.wikipedia.org/wiki/Body\_fat\_percentage*