Neural Networks Projects

Housing Fitting Neural Network

Tom Makkink

2018

**1 Introduction**

The real estate industry is an important component of any modern day economy, and is a critical driver of economic growth. Real estate involves purchasing and selling residential, commercial and industrial property. There are a variety of factors that affect the cost of real estate, such as the location of the property, the crime rate of the area, the number of nearby schools and shops etc.

We would like to assist in designing a neural network that is able to accurately estimate the price of a house in an area based on 13 demographic attributes.

**2 The Data**

The data was sourced from the Matlab Neural Networks example folder, and was loaded as follows:

load housing.mat

The data contained 506 entries, and was divided into a 506x13 pattern matrix, which contained values for 13 different demographic attributes, and a 506x1 target matrix, which contained the price of the corresponding house.

The 13 demographic attributes were as follows:

1. Per capita crime rate per town
2. Proportion of residential land zoned for lots over 25,000 sq. ft.
3. Proportion of non-retail business acres per town
4. 1 if tract bounds Charles river, 0 otherwise
5. Nitric oxides concentration (parts per 10 million)
6. Average number of rooms per dwelling
7. Proportion of owner-occupied units built prior to 1940
8. Weighted distances to five Boston employment centres
9. Index of accessibility to radial highways
10. Full-value property-tax rate per $10,000
11. Pupil-teacher ratio by town
12. 1000(Bk - 0.63)ˆ2
13. Percent lower status of the population

**3 Method**

A supervising script (housing\_sup) was first created to find the number of neurons in each hidden layer to best solve the problem. A range from 5 to 30 neurons were tested, in intervals of 5, in each hidden layer. The correlation coefficient and R-squared statistic were then calculated for each net, and compared.

The neural network that performed the best had 20 neurons in the first hidden layer, and 25 in the second. The network was saved as follows:

% Save the variables

save housing\_sup.mat housingnet

The housing network was then loaded in a training script, which creates and simulates a neural network with the following attributes:

* Two hidden layers, the first was 20 neurons and the second with 25.
* 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.

The net was then simulated as follows:

%simulate

atrain=sim(housingnet,ptrain); %train

atest=sim(housingnet,ptest); %test

a=sim(housingnet,p); %all

The r-squared statistic and correlation coefficient was then computed for the training set, the test set and both sets combined as followed:

%train

r2=rsq(ttrain,atrain);

[R,PV]=corrcoef(ttrain,atrain);

%test:

r2=rsq(ttest,atest);

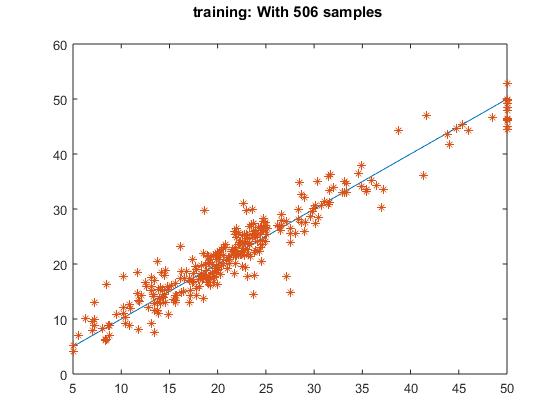
[R,PV]=corrcoef(ttest,atest);

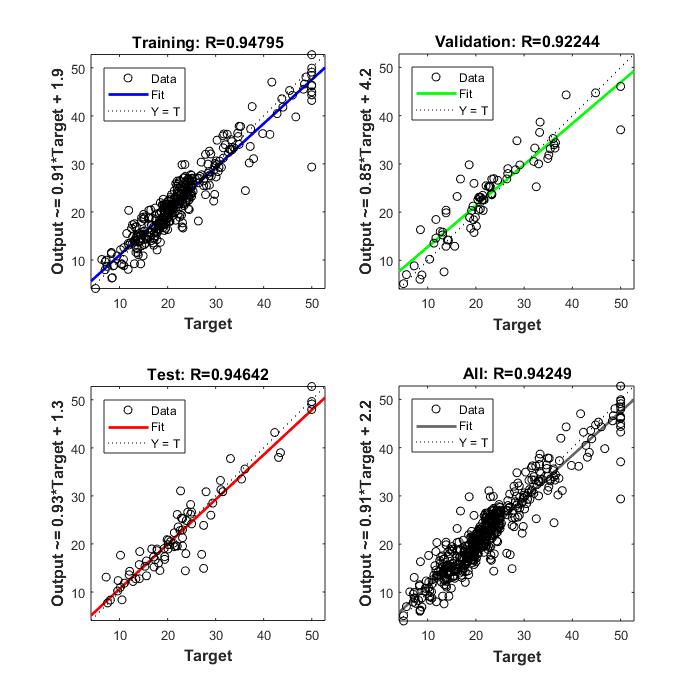
%all

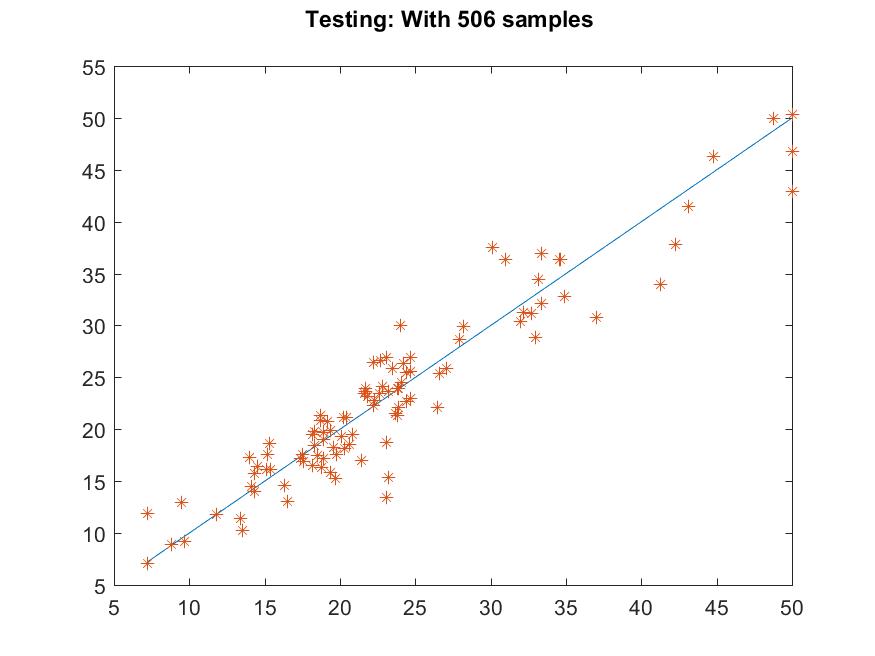
r2=rsq(t,a);

[R,PV]=corrcoef(t,a);

The results were then plotted, and produced the following results:







The Matlab post regression functions were then computed, and produced the following results:

train r= 0.925705

test r= 0.929663

all r= 0.930598

Finally, a housing function was created, which takes in a 13 x 1 input, and returns the estimated price of the house by propagating it through the network. The script is as follows:

% function housingfcn which accepts 13 × 1 inputs and returns

% the house value.

function price = housing\_fcn (x)

load housing\_sup.mat

price = sim(housingnet,x);

end

**4 Conclusion**

The overall results of testing the network were good, and it can be stated that it does provide a decent approximation of what a house will cost given a set of demographics.

**5 References**

*https://www.mathworks.com/help/nnet/gs/neural-network-toolbox-sample-data-sets.html*

**4 Conclusion**

**5 References**