Part B- California Housing Dataset

# Introduction:

The California Housing Prices Dataset consists of 8 different attributes of housing complexes in California like location, number of bedrooms, number of bathrooms etc. The data set also contains prices of the houses. The objective of this project is to predict housing prices by training the neural networks with the given attributes. The following experiments give an insight into choosing parameters like the right learning rate, number of hidden neurons, number of hidden layers for obtaining the best model to predict the housing prices.

## Train data and Test data:

The California housing price dataset has been divided into subsets for training and testing in the ratio of 70:30 respectively

## Five-Fold Cross Validation:

The training data has been further divided into a training subset and the validation subset in the ratio of 4:1. This is called as the five-fold cross validation method where 1/5 of the data is reserved for validation.

## Epochs:

A 1000 epochs are used for the experiments.

## Batch-Size:

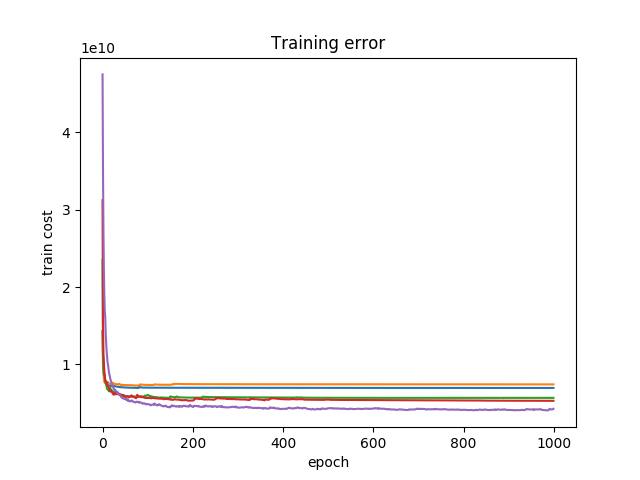
A batch size of 32 has been used. So, the weights and the biases are updated every 32 inputs in an iteration. This prevents updating too frequently but also converges faster. Though a batch size of 32 has been used, for the training cost, the mean of the batch costs is taken and plotted. Thus, the training error has been plotted against the number of epochs.

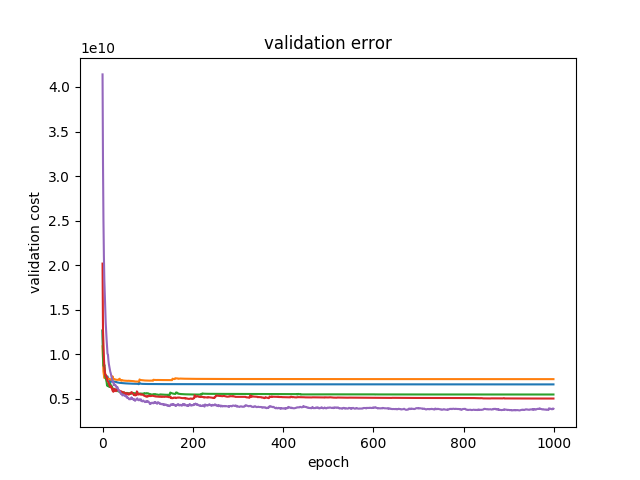
# 3- layer network:

# Different Learning rates:

The 3-layer network designed in the stage above was trained with different learning rates using 5-fold cross validation in order to find the optimal learning rates. The different learning rates that were tried were [0.001,0.5\*0.001,0.0001,0.5\*0.0001,0.00001].

\*\* Though the neural network was trained according to a batch size of 32 to increase the rate of converge, the training error of each epoch is the mean of the batch errors in one epoch.





## Index:

|  |  |
| --- | --- |
| 0.001 | Blue |
| 0.0005 | Orange |
| 0.0001 | Green |
| 0.00005 | Red |
| 0.00001 | Purple |

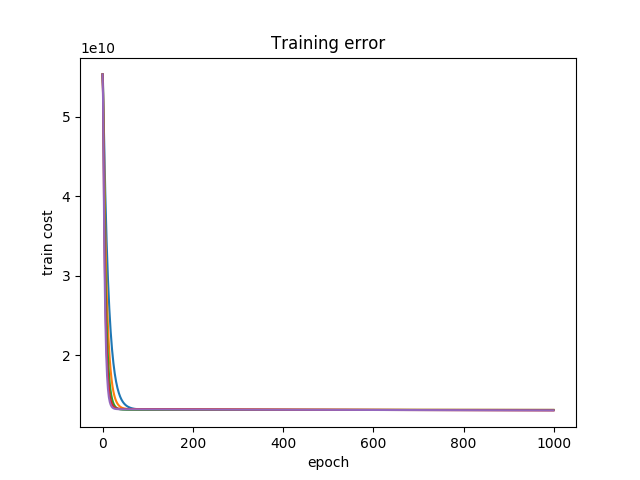
As we can see that alpha= 0.00001 converges faster than the other rates and hence that should be chosen as the optimal learning rate = 0.00001 (1e-5) or 10 raised to -5

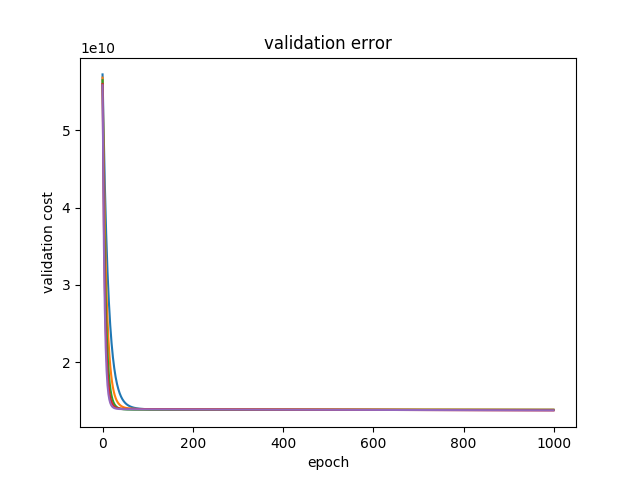
We trained the network again using the optimal training rate and plotted the predictions against the outcomes.

# Different number of hidden neurons:

The 3-layer network designed in the stage above was trained with different neurons in the hidden layer using 5-fold cross validation in order to find the optimal hidden neurons. The different number of neurons that were tried were [20,30,40,50,60].

\*\* Though the neural network was trained according to a batch size of 32 to increase the rate of converge, the training error of each epoch is the mean of the batch errors in one epoch.





## Index:

|  |  |
| --- | --- |
| 20 | Blue |
| 30 | Orange |
| 40 | Green |
| 50 | Red |
| 60 | Purple |

As we can see the training converges faster when there are more number of neurons in the hidden layer. Hence n=60 should be chosen as the optimum amount of neurons.

# Different number of layers:

# Final model: