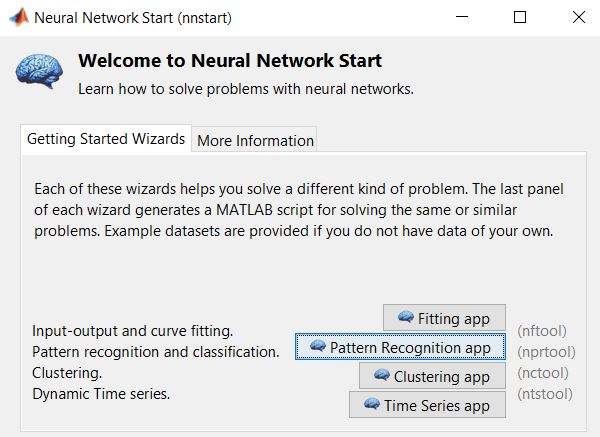
Q5 hw3data.mat contains X and Y-target. There are two classes in this data set. Specifically, in Y-target, there are two line, each of them represent a class. For instance, if the value of the element yi,j is 1, it means the instance i belongs to class j. Apply ANN to this dataset. (Refer to neural network toolbox in matlab).

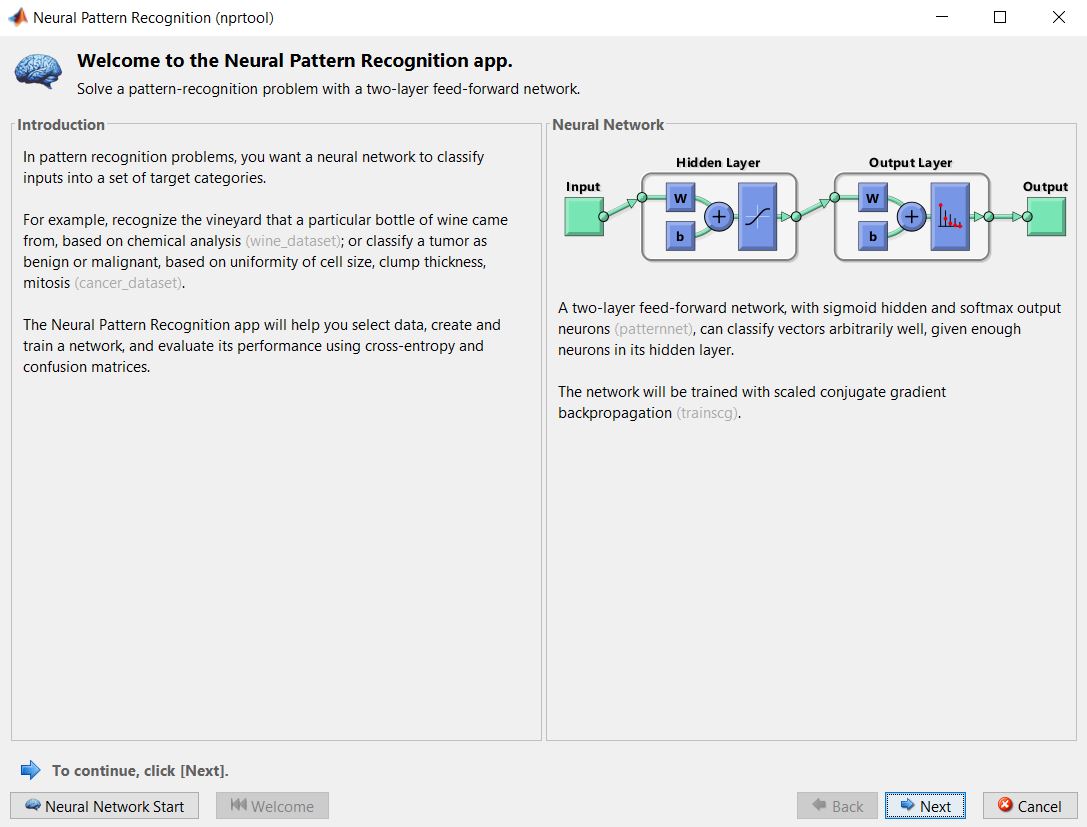
Don't split the data into training, valid and test, just use the default setting for the split. For each time, vary the number of hidden neurons in the order of 2,4,6,8,10. After that, plot a figure, where the horizontal line is the number of hidden neurons, and the vertical line is the error rate. Provide a simple analysis of your results. Please use matlab for this problem and include step by step screenshots of the toolbox in your answer.

Solution:

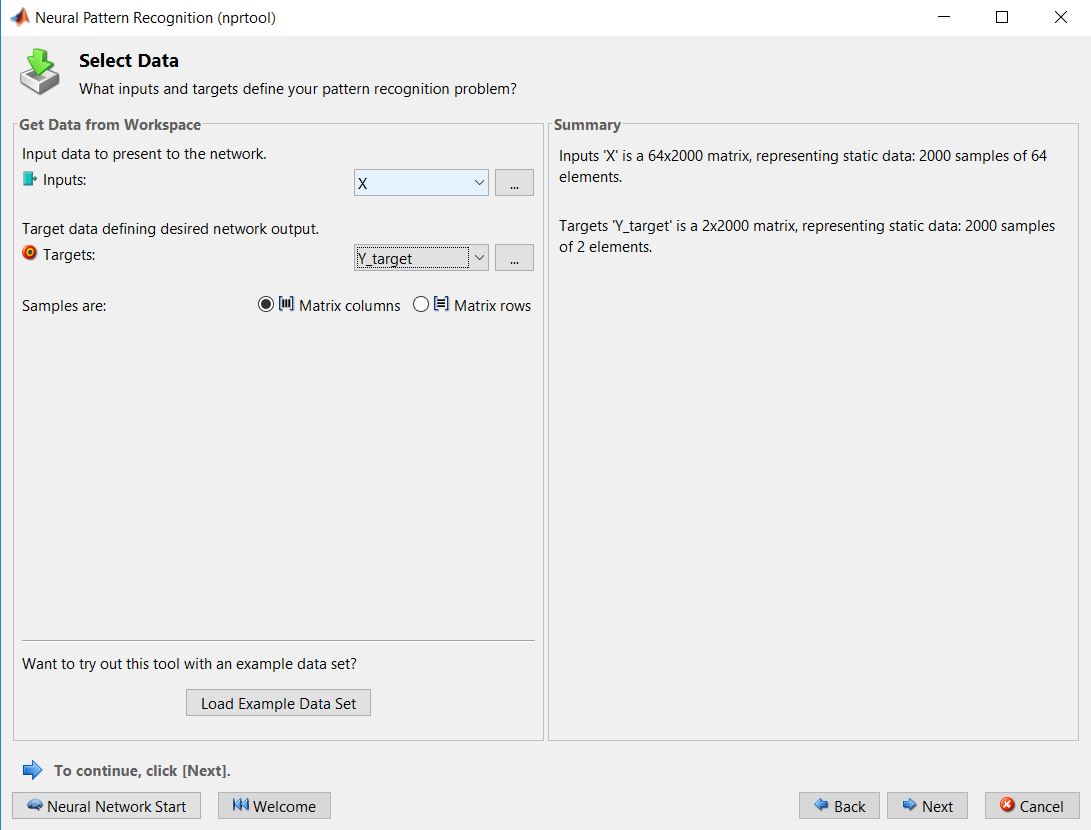
1. Enter *nnstart* in console to start the Neural Network Toolbox. Click on the *Pattern Recognition app* or type *nprtool* on the console to start the application.



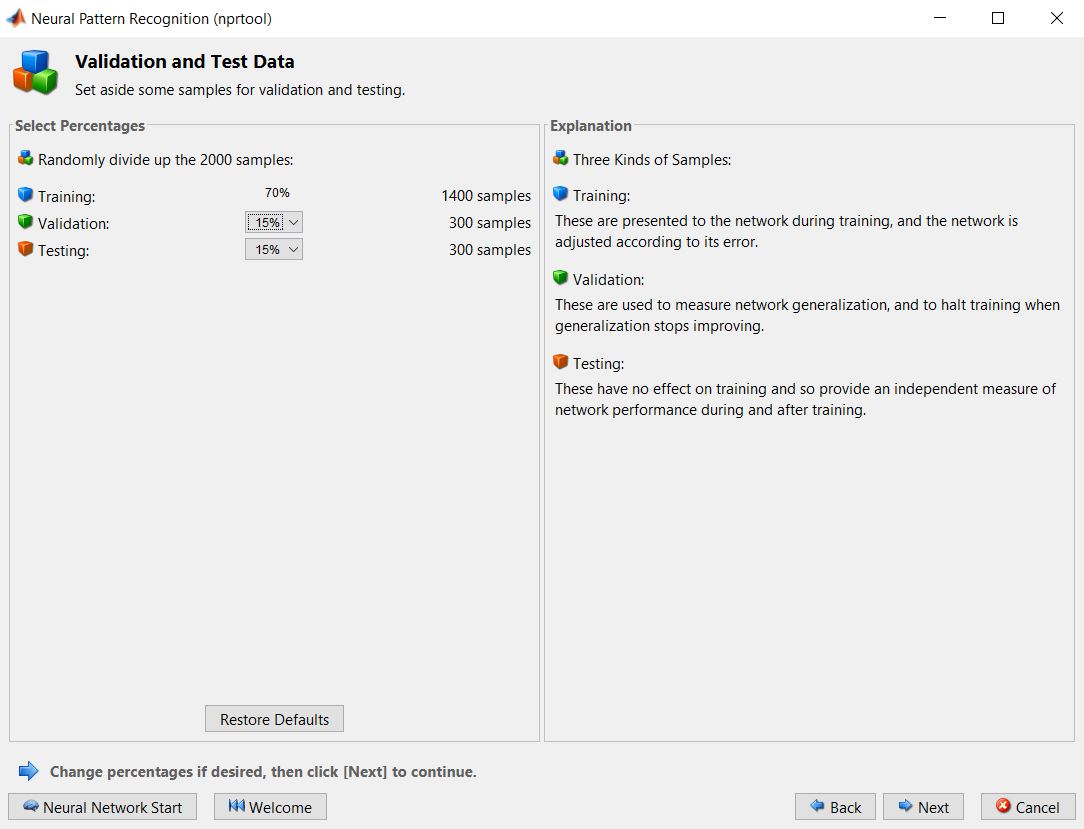
1. This screen explains the type of activity the app is performing and how it is going to be performed. There is also a diagram of the network being made. In this screen all that you have to do is click next to enter the next screen to select the data on which to perform Pattern recognition.



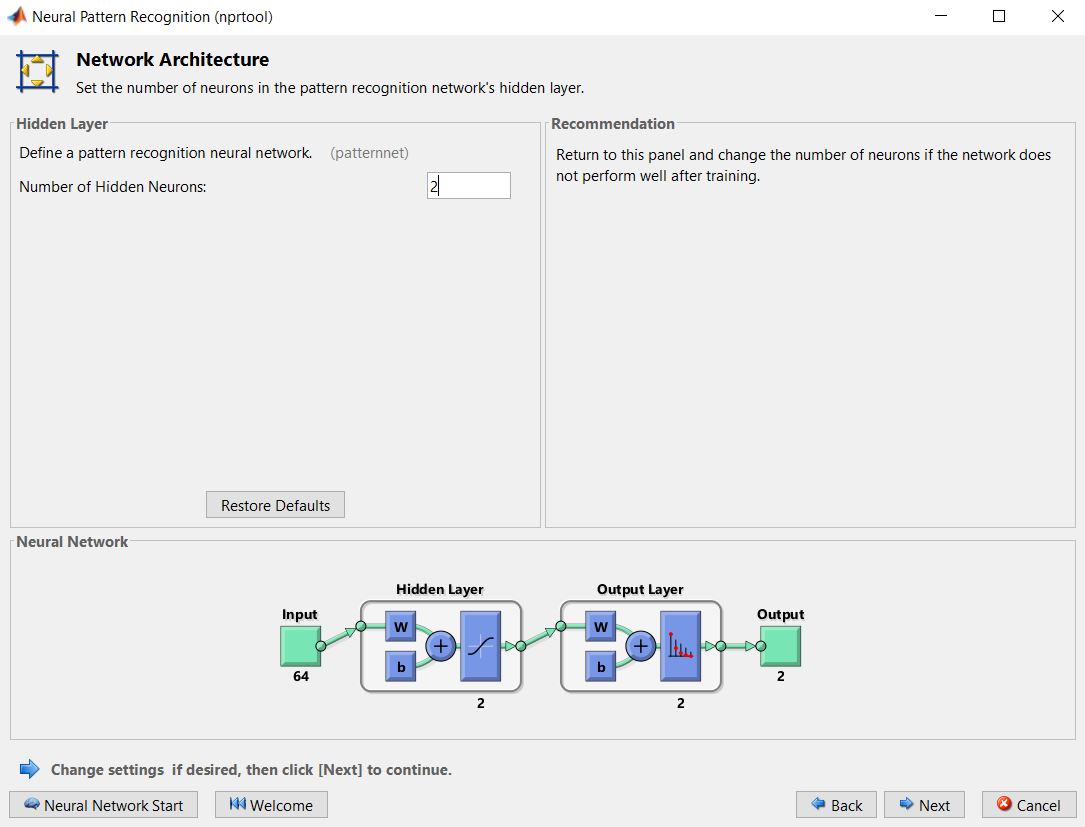
1. In the *select data* screen, we choose the input data we have from the *Inputs* drop down and their desired target outputs from the *Targets* drop down. Here we have chosen a 64x2000 variable X as input and 2x2000 variable Y\_target as targeted output. Click next.



1. In this screen, we decide the ratio in which the input data is going to divided into Training, Validation and Testing sets. We are using the default values for Training, Validation and Testing sets i.e. 70%, 15% and 15%. Click next.

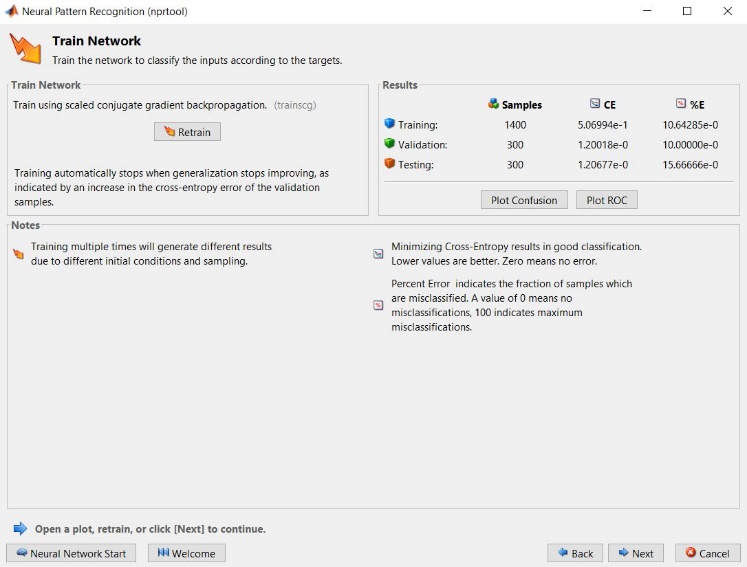
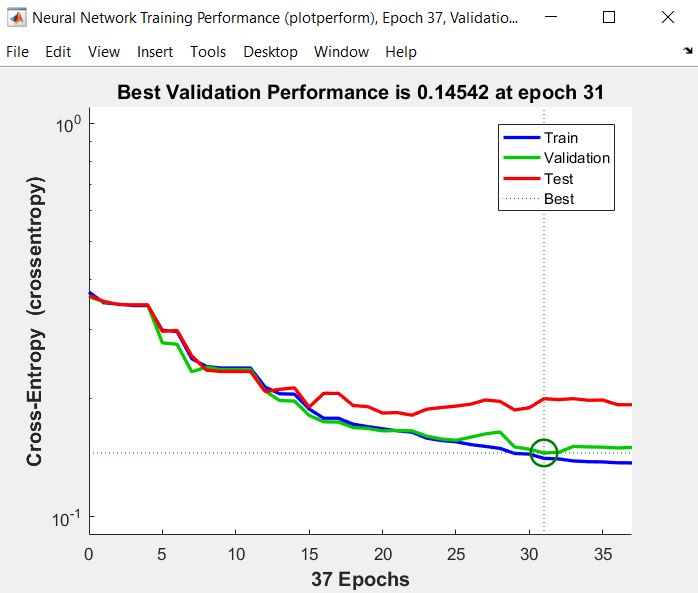


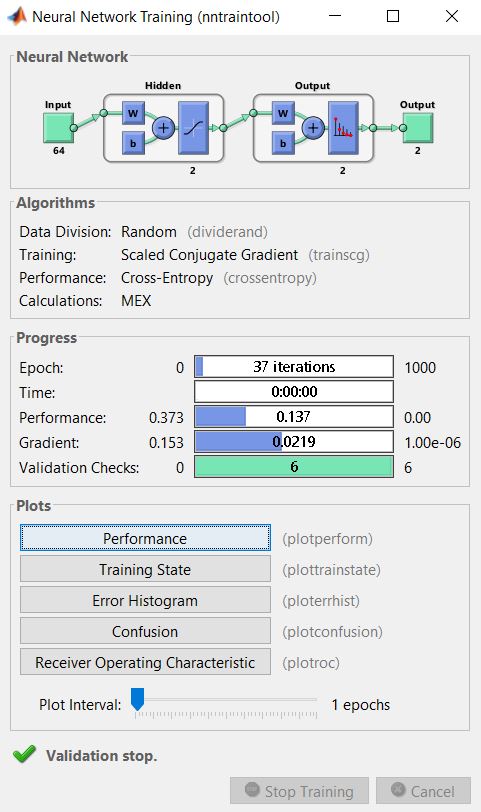
1. In this screen, we choose the hidden neurons that should be in the hidden network. For this question the values should be 2, 4, 6, 8 and 10. Below we will show the result for different values in the hidden neurons field. Click next and train the network by clicking the train button.



* 1. Hidden Neurons: 2

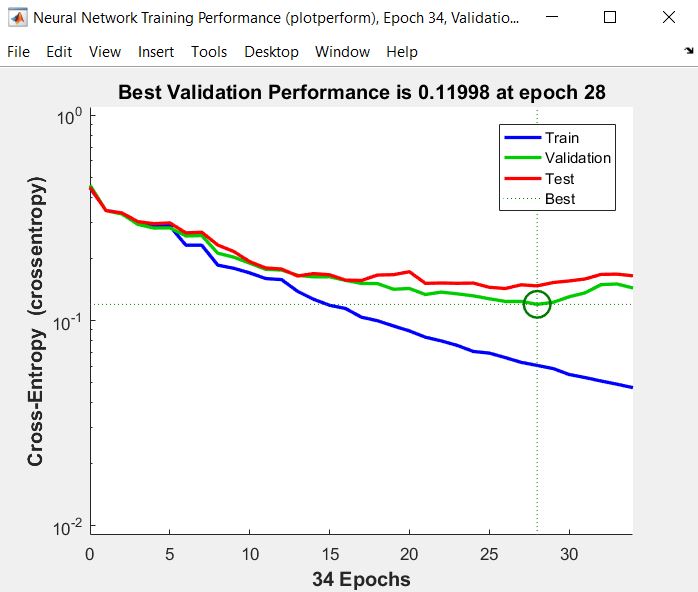
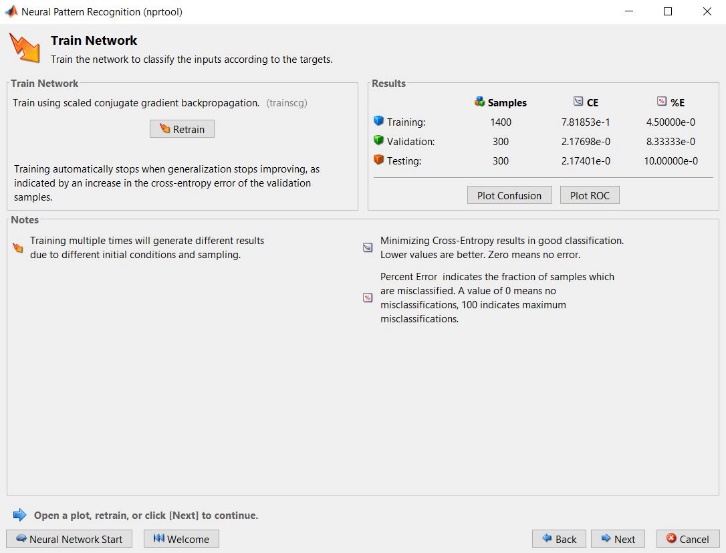
The below images show the data found after training the input with 2 hidden neurons. The training stops at epoch 37 where we can see that validation performance has been rising for the last 6 epochs. This shows that at epoch 31 we have found the minimum of cross entropy. The errors are shown in the top left figure.

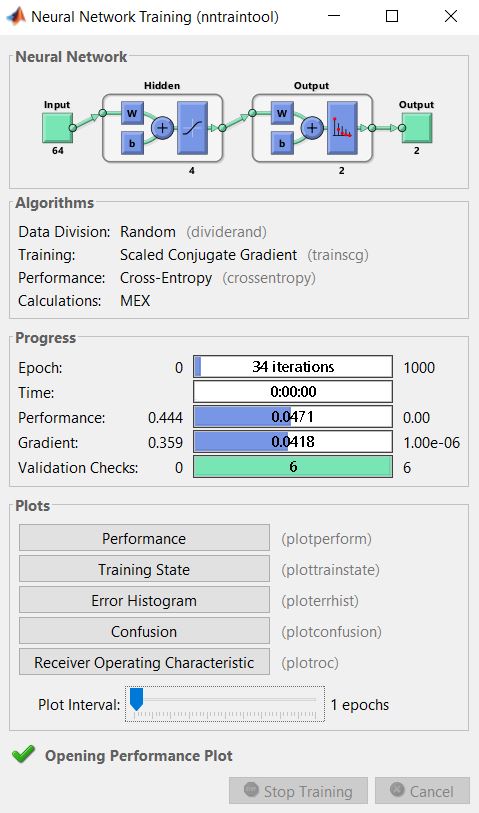
 



* 1. Hidden Neurons: 4

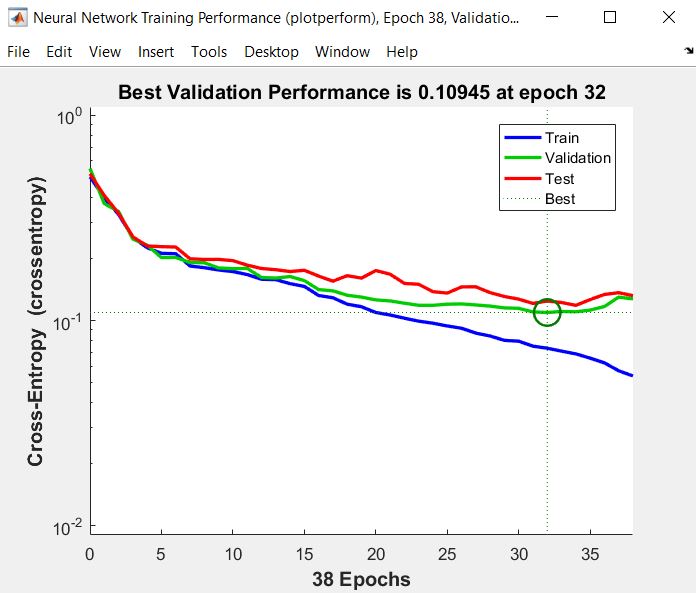
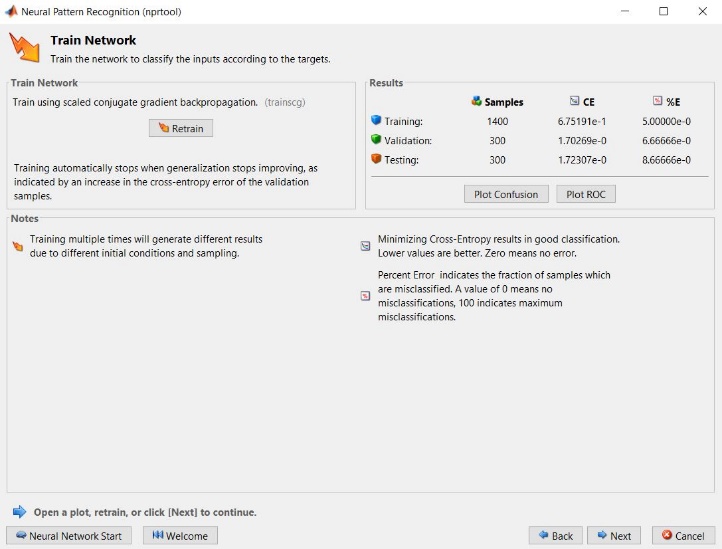
The below screen shots show the data found after training the input with 4 hidden neurons. The training stops at epoch 34 where we can see that validation performance has been rising for the last 6 epochs. This shows that at epoch 28 we have found the minimum of cross entropy. The errors are shown in the top left figure.

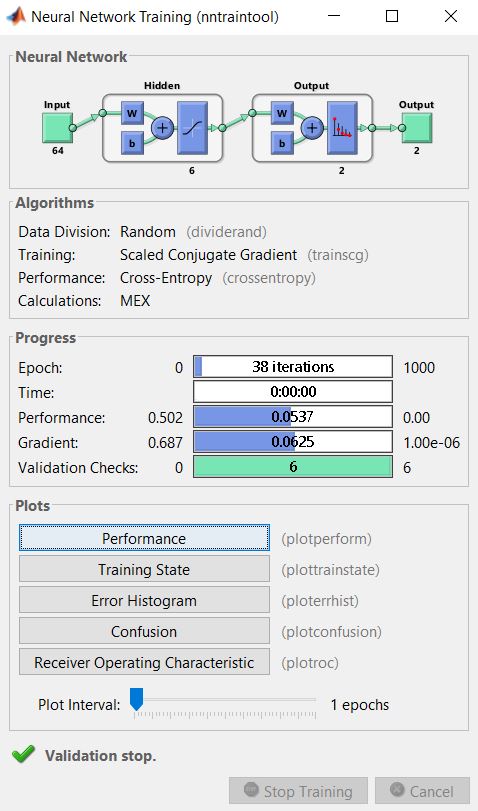




* 1. Hidden Neurons: 6

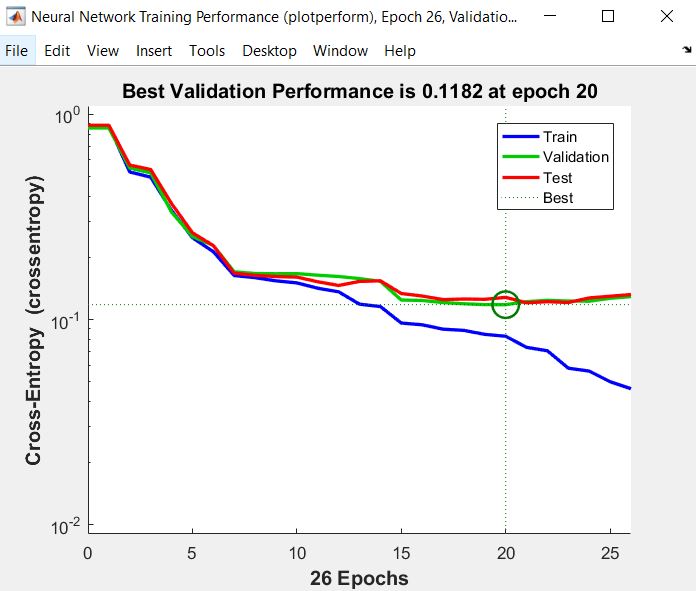
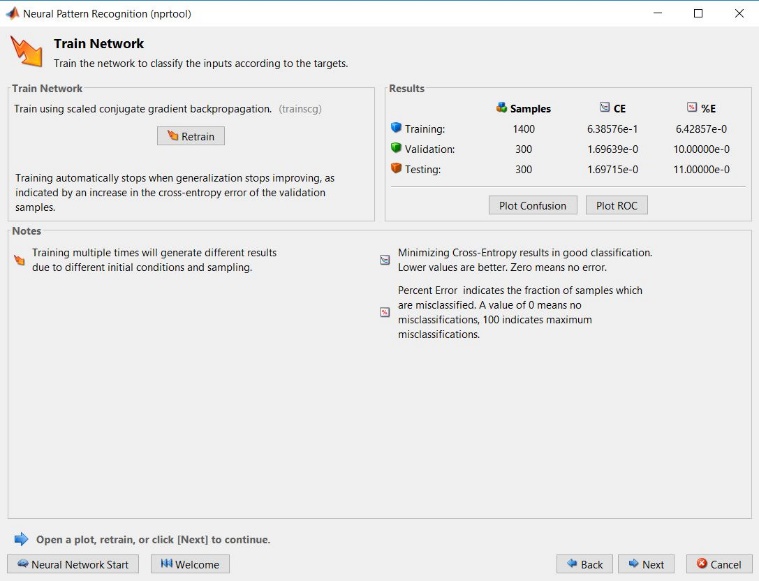
The below screen shots show the data found after training the input with 6 hidden neurons. The training stops at epoch 38 where we can see that validation performance has been rising for the last 6 epochs. This shows that at epoch 32 we have found the minimum of cross entropy. The errors are shown in the top left figure.

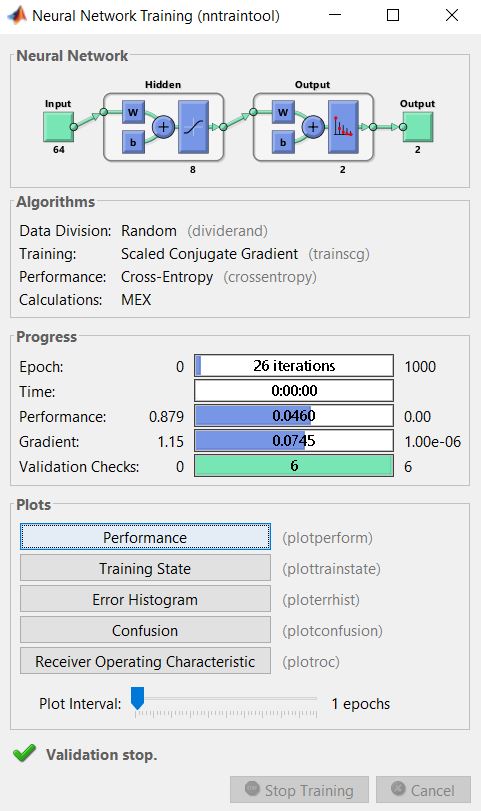




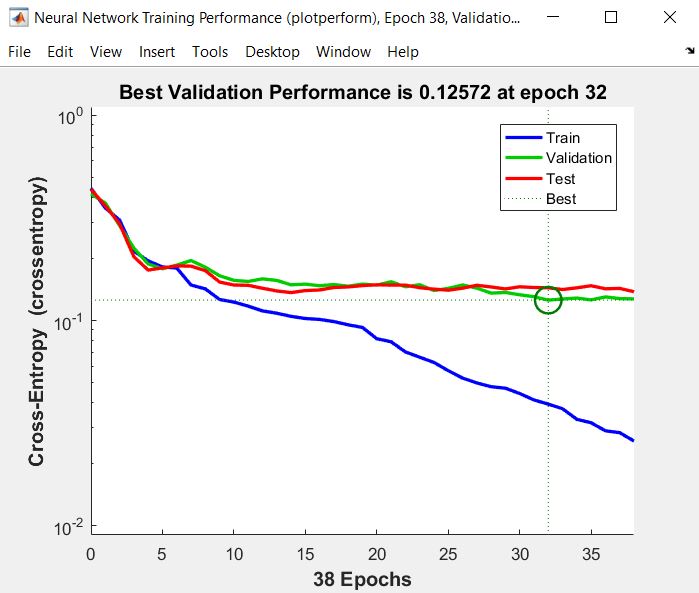
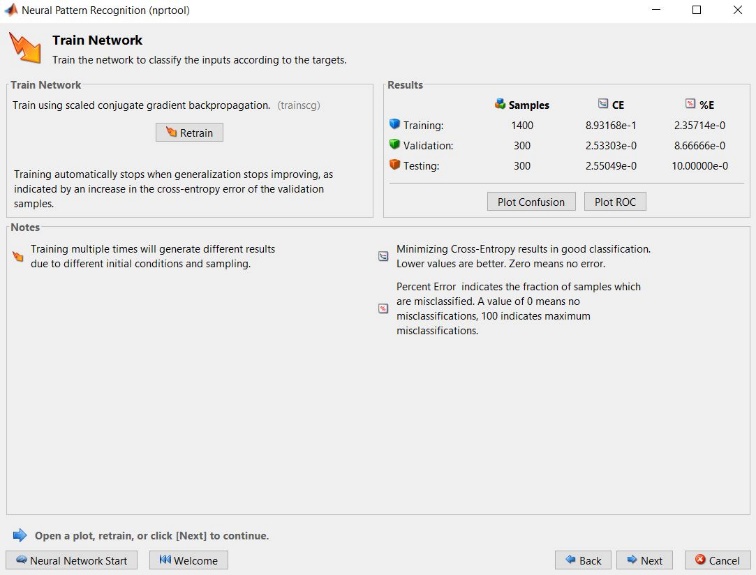
* 1. Hidden Neurons: 8

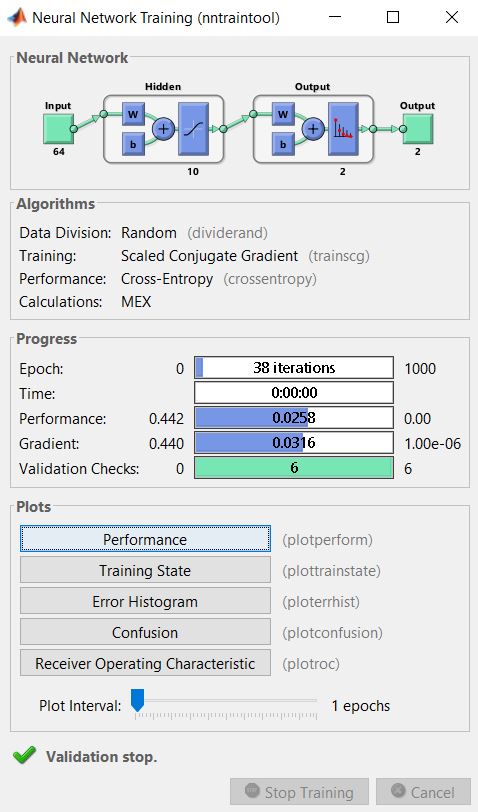
The below screen shots show the data found after training the input with 8 hidden neurons. The training stops at epoch 26 where we can see that validation performance has been rising for the last 6 epochs. This shows that at epoch 20 we have found the minimum of cross entropy. The errors are shown in the top left figure.





* 1. Hidden Neurons: 10  
     The below screen shots show the data found after training the input with 2 hidden neurons. The training stops at epoch 38 where we can see that validation performance has been rising for the last 6 epochs. This shows that at epoch 32 we have found the minimum of cross entropy. The errors are shown in the top left figure.





The below plot compares the error percentage with the number of hidden neurons. Error percentage deals with the fraction of samples that were misclassified. Using below plot we can decide that 6 hidden neurons will give the best result. Above that we will start overfitting our solution.

