Question 1

Part A

Consider using RBFN to approximate the following function:

$$y = 1.2\sin(\pi x) - \cos(2.4\pi x), \quad for\ x \in [-1.6, 1.6]$$

The training set is constructed by dividing the range $[-1.6, 1.6]$ using a uniform step length 0.08, while the test set is constructed by dividing the range $[-1.6, 1.6]$ using a uniform step length 0.01. Assume that the observed outputs in the training set are corrupted by random noise as follows.

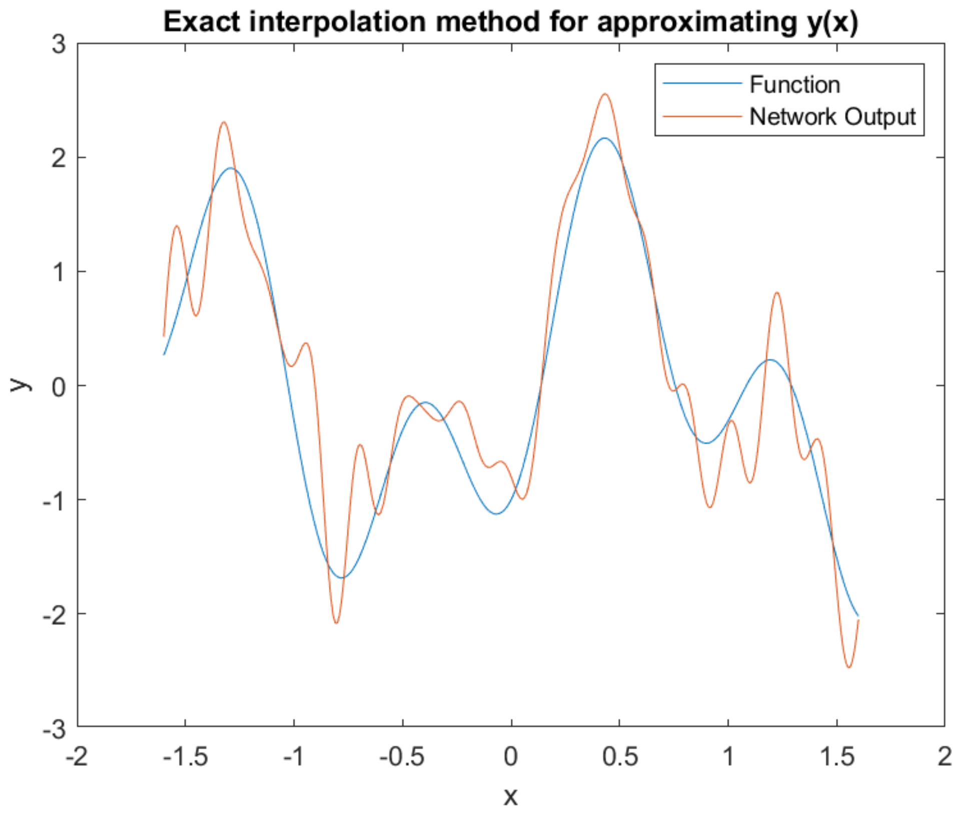


Figure 1 The training of RBFN using Exact Interpolation

Figure 1 shows that there is an overfitting issue in this part. Since number of neurons is equal to number of training samples, which is equal to 41, there are more centers than it is required to approximate this function in this interval.

Part B

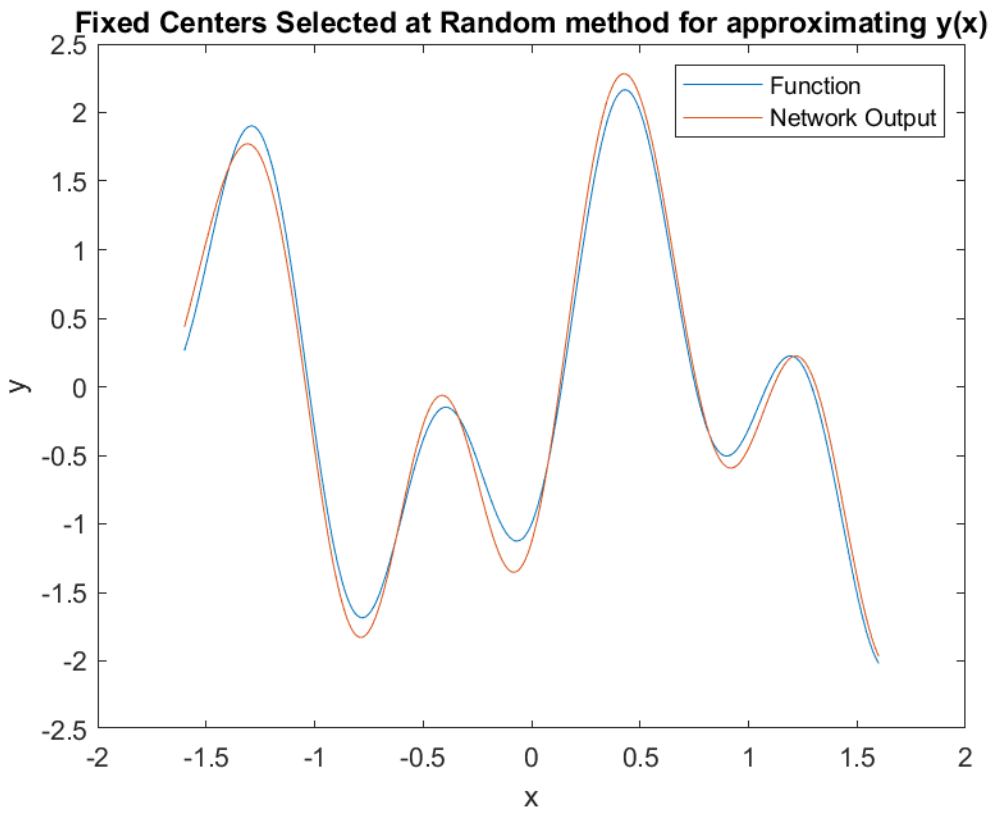


Figure 2 Fixed Centers Selected at Random with 20 Centers

In the part a, the total number of centers are 41. The number of centers is the half of the part a. This method with 20 centers approximate function in the given interval better with less noise. However, there is an issue while finding the weights after calculating the interpolation matrix. Interpolation matrix is (41x20) matrix and to be able to calculate the inverse of it, I used formula at the below.

$$w = (\Phi^T \Phi)^{-1} \Phi^T d$$

It is non-singular but the values at some columns are too close to 0 and they act like linearly dependent. Therefore, this formula is not worked out only by itself. I needed to use pseudo inverse function calculating SVD of the matrix in order to filter out the linear dependency to find the correct inverse to calculate the weights.

Part C

Refer to Figure 3, the existence of regularization decreases the noise in the function's approximation drastically. The change in regularization parameter from 0 to 0.1 achieves a significant improvement in the approximation but increasing the parameter doesn't increase the performance more and more. The performance of this method is almost the same as part b.

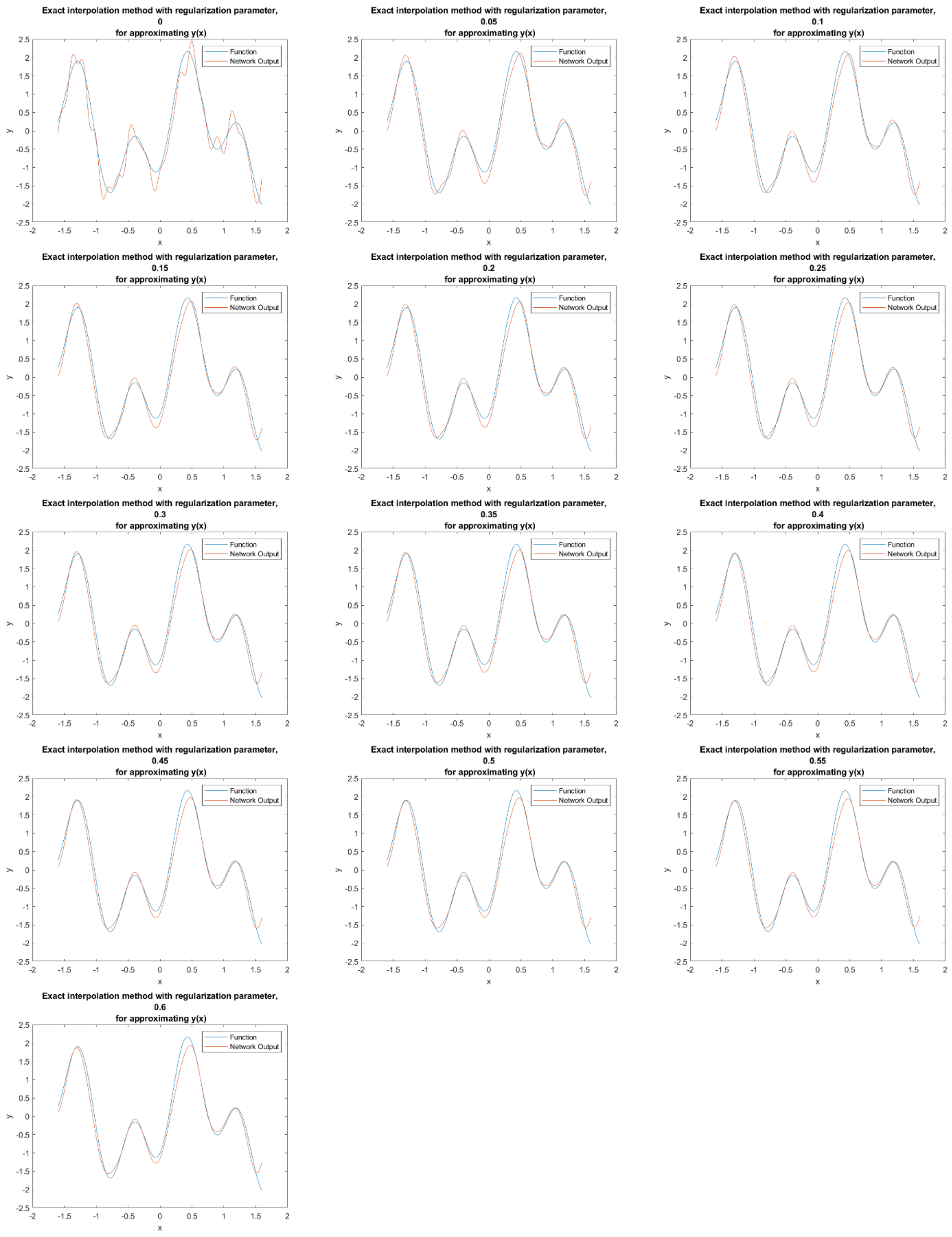


Figure 3 Exact Interpolation with regularization

Question 2

My classes are 2 and 9. I labeled class number 2 and 9 as '1' and the rest are '0'.

Part A

Refer to Figure 4, the performance of the network in the training data is better than the performance of the test data. Regularization decreases the overfitting at the network and it helps us to detect a reasonable threshold value for the digits classification. The interpolation matrix is calculated through each sample in the dataset and it is fixed while varying regularization parameter from 0 to 1.1. Figure 4 shows that the accuracy of network increases when the threshold value is slightly greater than 0.2. The purpose of varying threshold is to find the optimal threshold value for the trained network. The labels are binary but the output of the network is floating point decimal. Therefore, we need to determine a threshold value such that we can decide on the predicted output value. The threshold value for trained network is around 0.2. The further increase in the regularization doesn't improve the performance of the network.

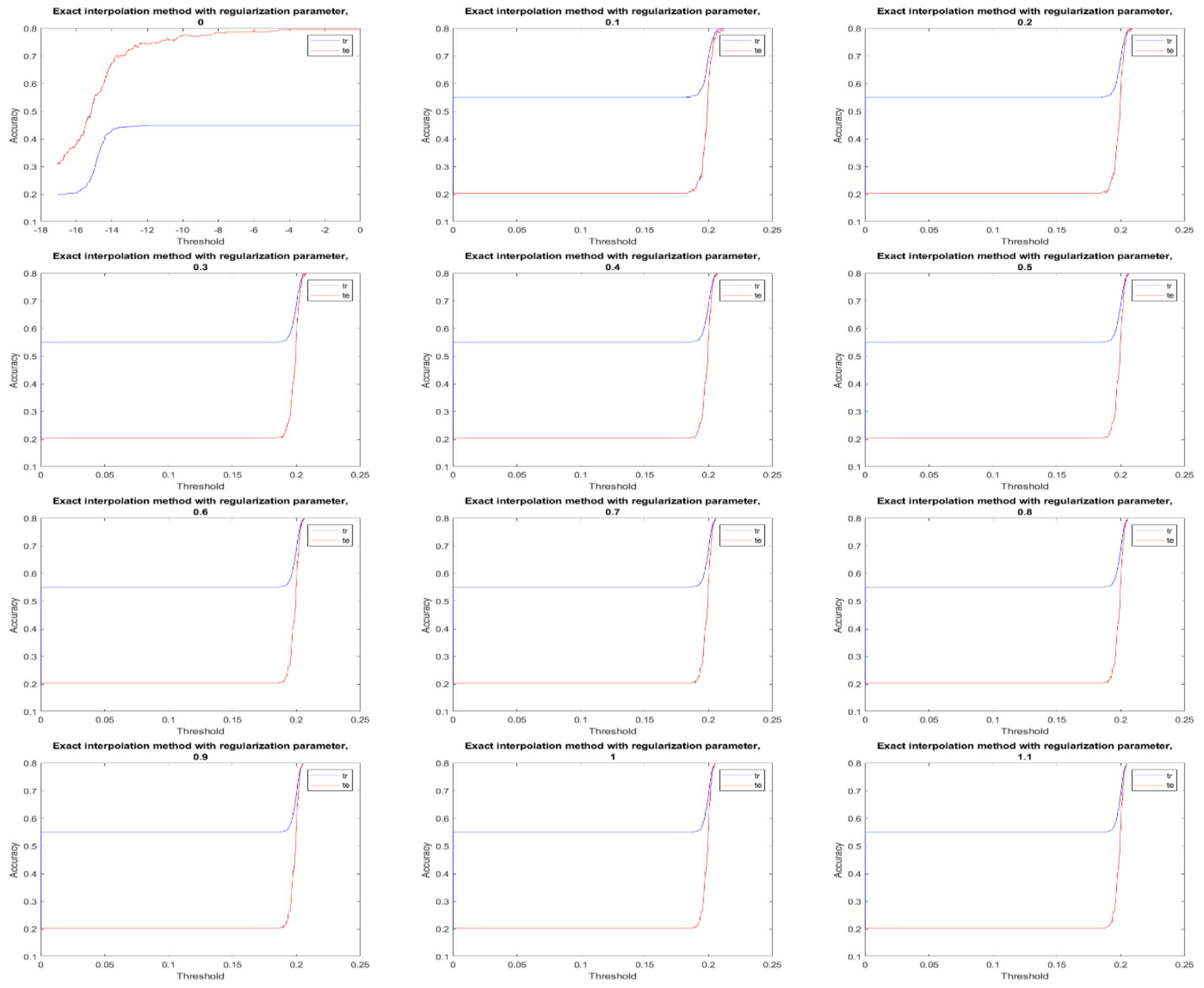


Figure 4 Exact Interpolation with different regularization values for image classification

Part B

Figure 5 demonstrates fixed centers selected at random with standard deviation 100. The result of this method for width = 100 is almost the same as the result of part a.

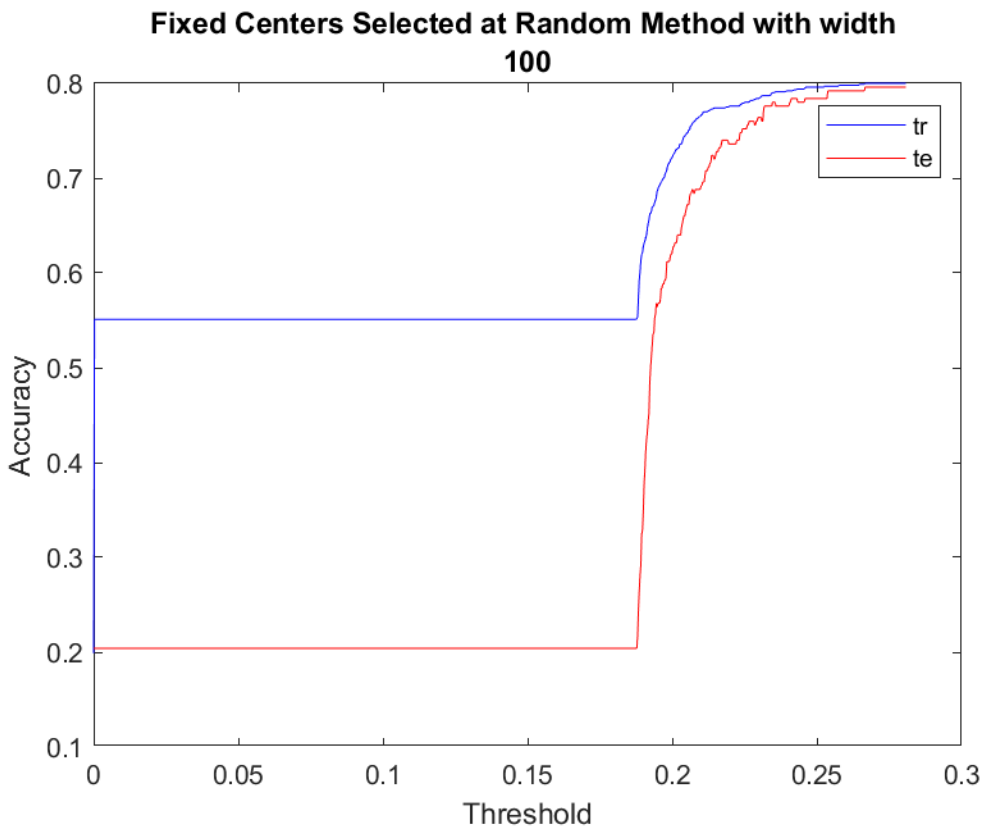
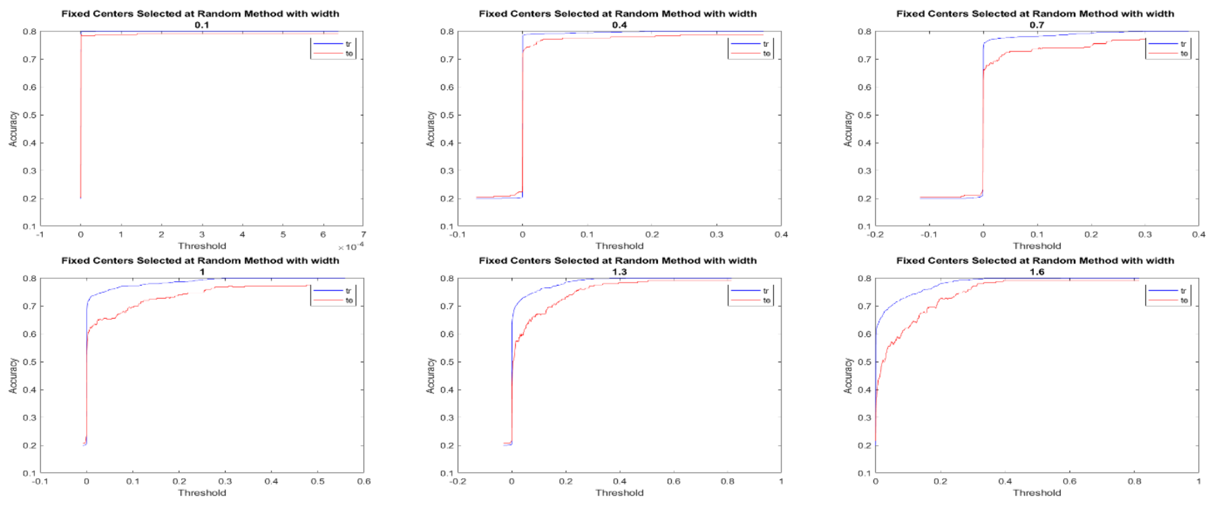
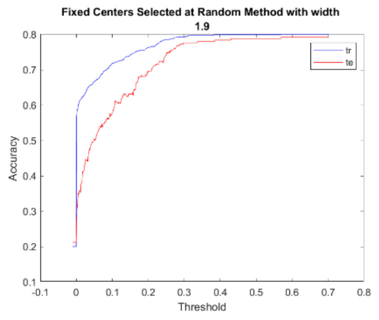
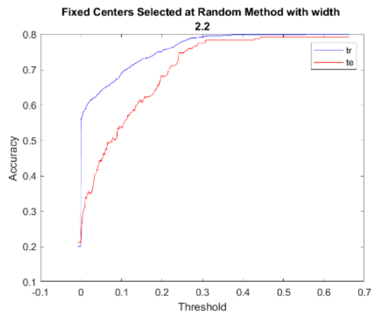


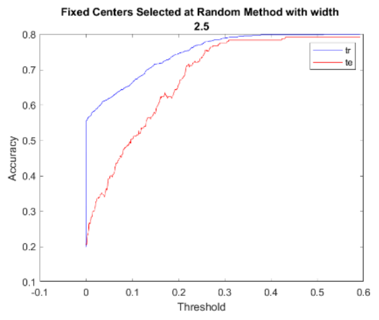
Figure 5 Fixed Centers Selected at Random Method with standard deviation 100

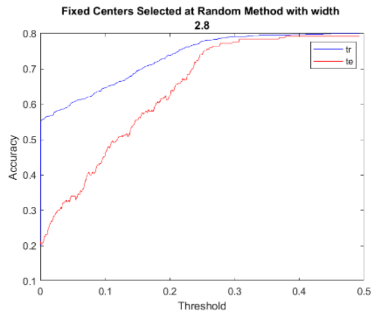
Figure 6 shows that as the standard deviation increases the threshold value tends to increase. This happens because the value of radial basis function evaluated at an arbitrary input increase.

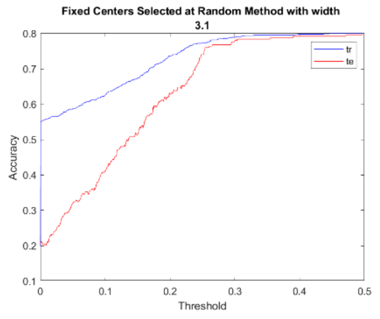


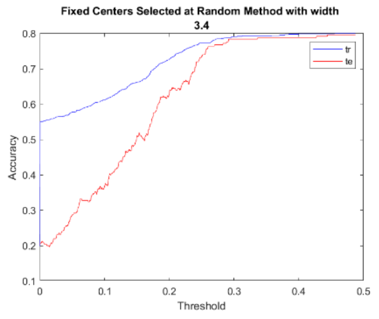

Fixed Centers Selected at Random Method with width 1.9
Accuracy
Threshold
tr
te

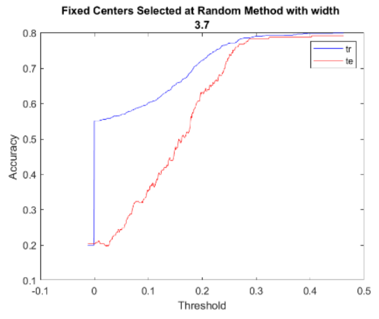

Fixed Centers Selected at Random Method with width 2.2
Accuracy
Threshold
tr
te

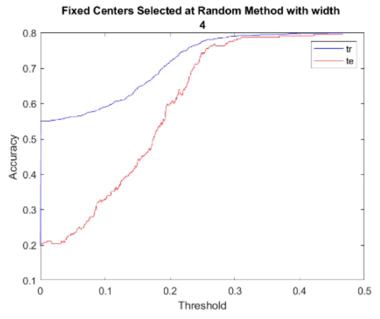

Fixed Centers Selected at Random Method with width 2.5
Accuracy
Threshold
tr
te

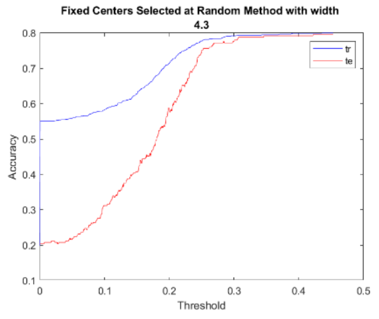

Fixed Centers Selected at Random Method with width 2.8
Accuracy
Threshold
tr
te

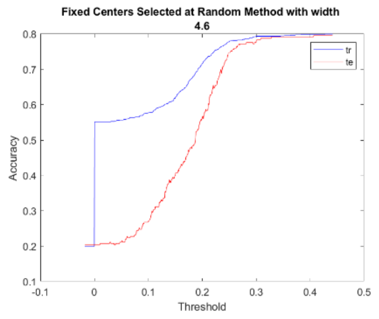

Fixed Centers Selected at Random Method with width 3.1
Accuracy
Threshold
tr
te

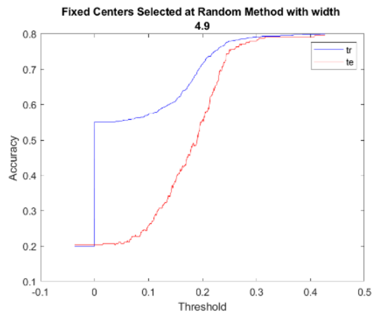

Fixed Centers Selected at Random Method with width 3.4
Accuracy
Threshold
tr
te

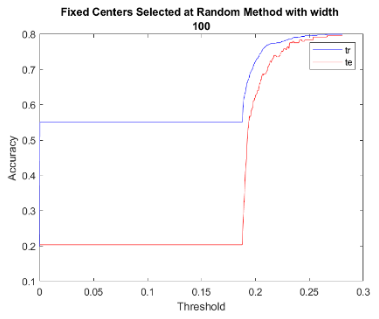

Fixed Centers Selected at Random Method with width 3.7
Accuracy
Threshold
tr
te

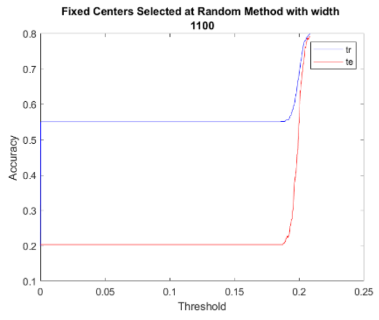

Fixed Centers Selected at Random Method with width 4
Accuracy
Threshold
tr
te

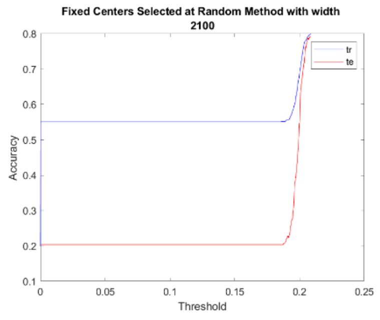

Fixed Centers Selected at Random Method with width 4.3
Accuracy
Threshold
tr
te

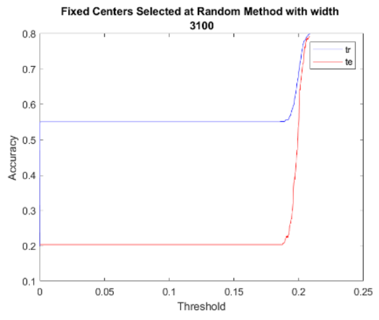

Fixed Centers Selected at Random Method with width 4.6
Accuracy
Threshold
tr
te


Fixed Centers Selected at Random Method with width 4.9
Accuracy
Threshold
tr
te


Fixed Centers Selected at Random Method with width 100
Accuracy
Threshold
tr
te


Fixed Centers Selected at Random Method with width 1100
Accuracy
Threshold
tr
te


Fixed Centers Selected at Random Method with width 2100
Accuracy
Threshold
tr
te


Fixed Centers Selected at Random Method with width 3100
Accuracy
Threshold
tr
te

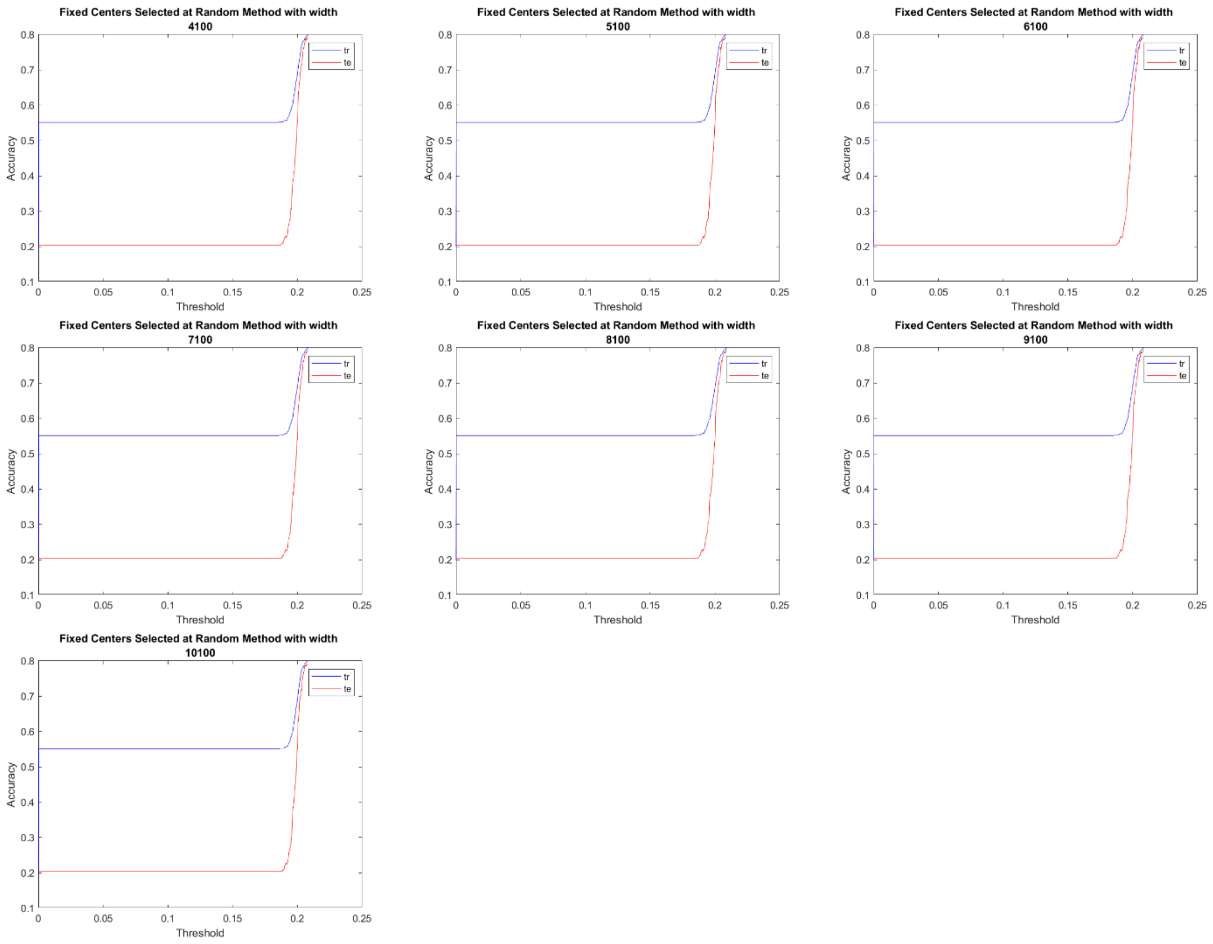


Figure 6 Fixed Centers Selected at Random for different width values

Question 3 – SOM

Part A

In this part, the aim is to build a self-organizing map network to approximate the sinc function in the region from x=-3 to x=3. Map is one dimensional map (40,1) containing 40 neurons. The centers of neurons, namely the weights of neurons, are randomly initialized in the region (x,y) = (-4:4,-4:4). Figure 6 shows the randomly initialized neurons at zeroth iteration of map's training. Figure 6 shows that after 500 iterations neurons are almost aligned with the sinc function and approximated it.

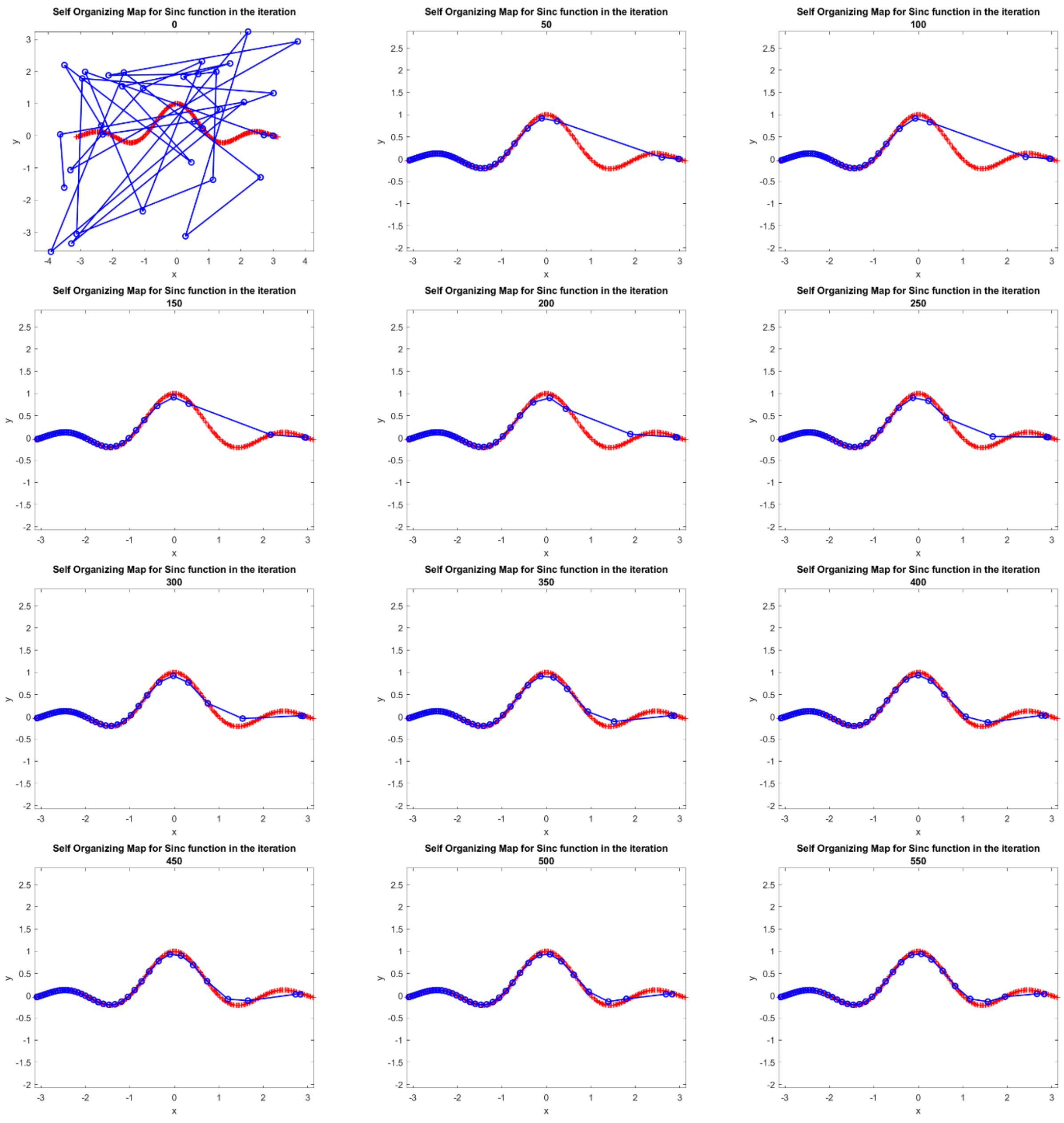


Figure 7 SOM for sinc function

Part B

In part B, it is expected to build a self-organizing map in order to organize neurons with respect to random samples taken from inside of a circle. The shape of map is 8x8 neuron matrix and there is total 64 neurons to organize the map. I couldn't place lines between adjacent neurons; however, the structure of neurons (8x8) can be observed from Figure 7. The map almost converges to its final structure after the fourth itereation as in the Figure 7. It works quite fast compared to other learning algorithms since it is unsupervised learning scheme.

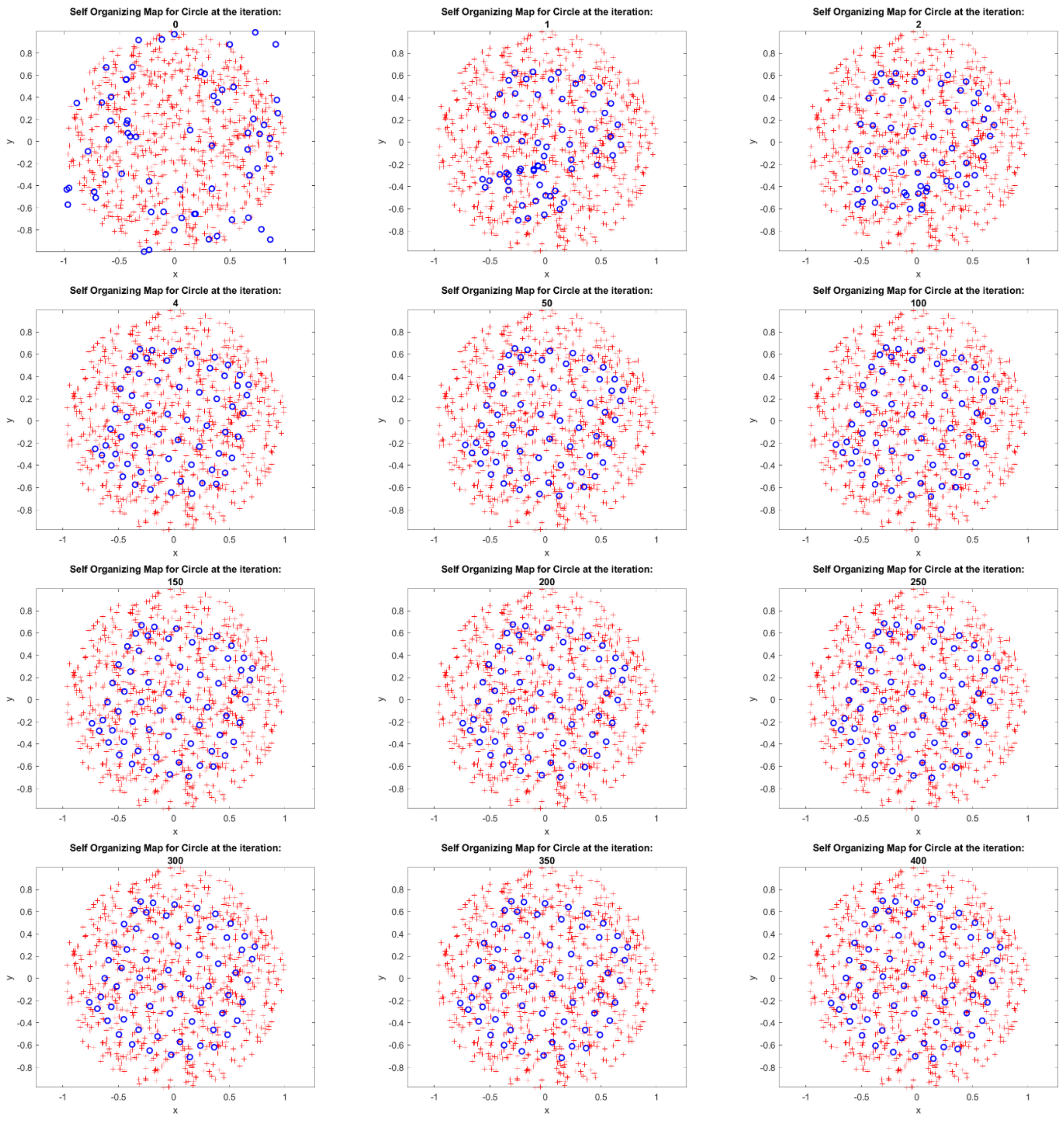


Figure 8 SOM for a Circle