Q1 Function Approximation with RBFN (10 Marks)

a) In the first case, RBFN is trained using the 'Exact Interpolation' method, and the fitting result is displayed in Figure 1.

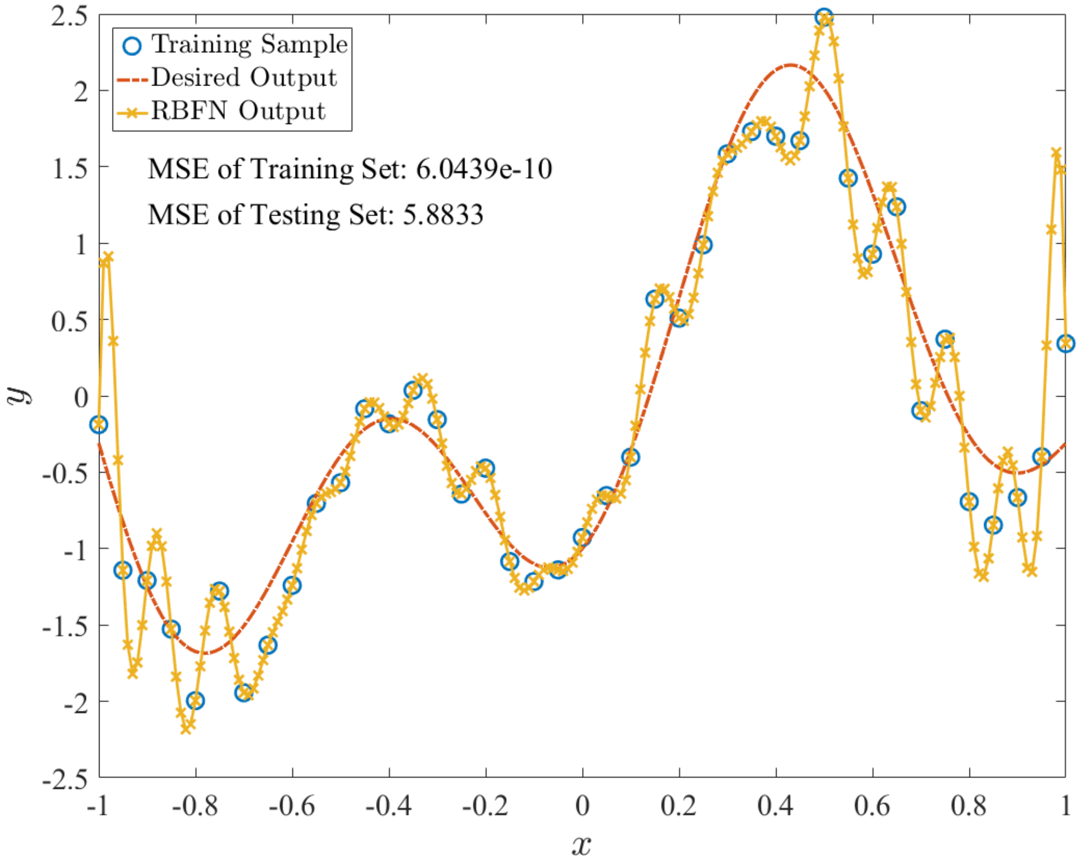


Figure 1 Fitting result obtained via the method of Exact Interpolation.

Mean Square Error (MSE) upon the training set is 6.04×10^{-10}, which is approximately 0, since the weights of RBFN are derived to perfectly fit all the training samples. In the meanwhile, MSE upon the test set is 5.88, which is because of the inability of RBFN to model the underlying function corrupted by noise.

b) In this case, only 20 training samples are randomly selected as the centers of RBFN. Moreover, the bias term is introduced following the fashion given in Page34 of Lecture5. The fitting result is shown in Figure 2. Compared with a), MSE drops to 2.48 for the test set while increases to 1.47 for the training set. Additionaly, the fitting curve obtained in this case is much smoother than that obtained in section a).

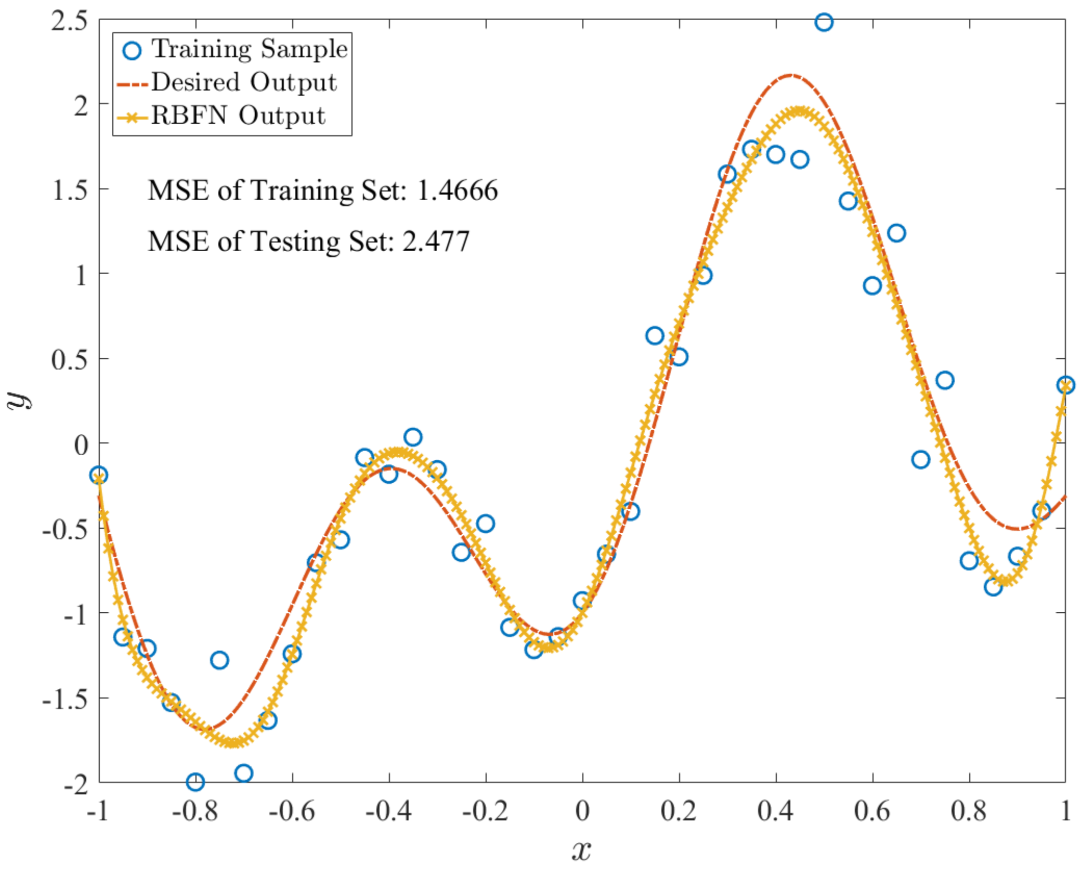
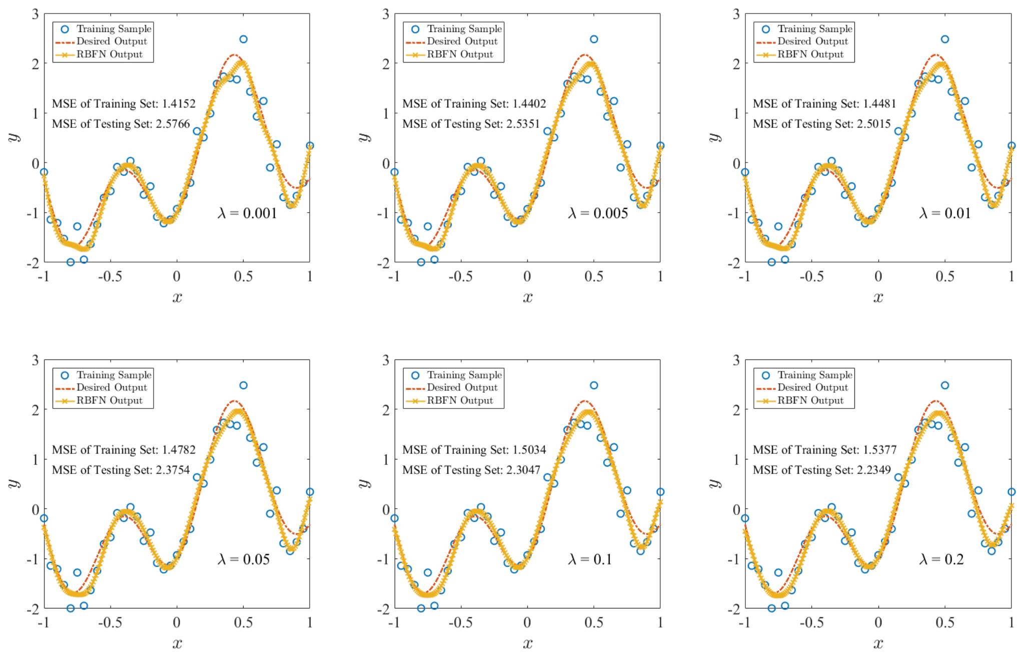


Figure 2 Fitting result obtained via the strategy of Fixed Centers (20) Selected at Random.

c) Use the same centers and widths as those determined in a) and apply the regularization with varying strength (λ) to RBFN. The obtained fitting results are given in Figure 3, and the relation between regularization strength and the fitting performance of RBFN is illustrated in Figure 4.



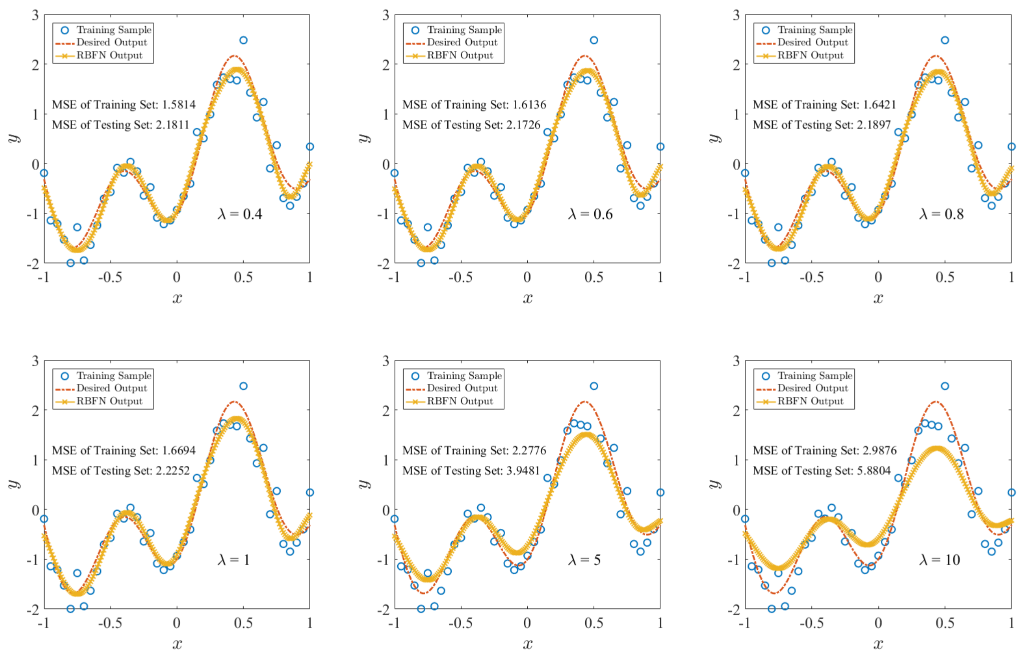


Figure 3 Fitting results obtained via the method of Exact Interpolation with different regularization strength.

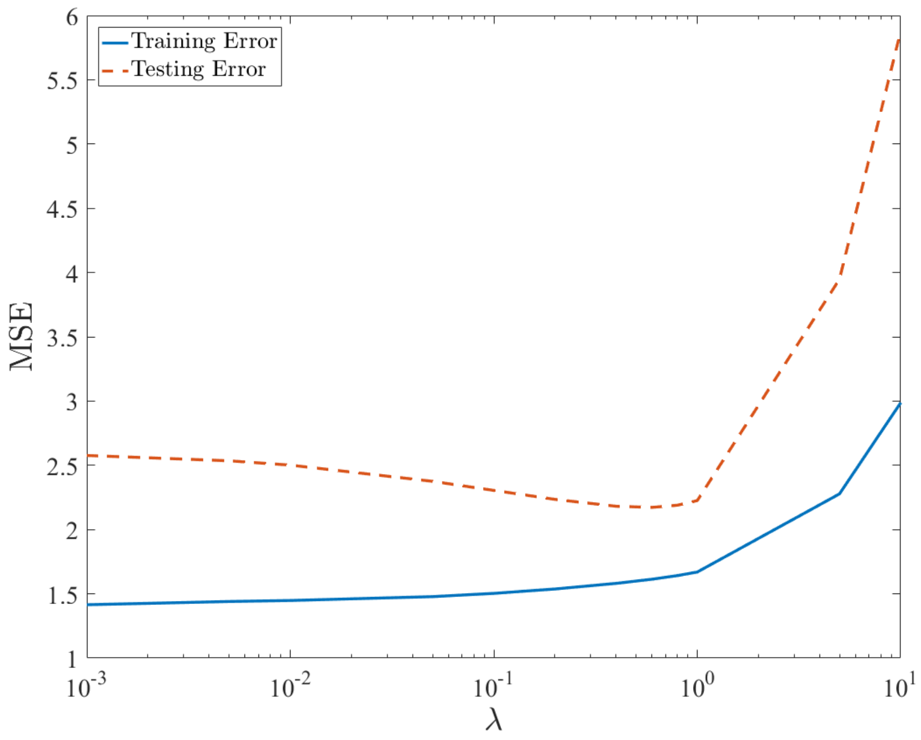


Figure 4 MSE upon the training/test set versus regularization strength.

It can be found from Figure 3-4 that the fitting curve becomes smoother as the regularization strength increases. In particular, when $\lambda = 0.6$, the performance of resulted RBFN reaches its best. However, if the regularization strength continues to grow, slopes of the obtained curve would become too flat to properly fit the underlying function and the MSE starts to rise.

Q2 Handwritten Digits Classification using RBFN (20 Marks)

The following experiments are performed upon class 1 (‘E’) and class 6 (‘T’) for demonstration.

a) Use the ‘Exact Interpolation’ method and apply regularization with different strength to determine the weights of RBFN. Classification performance of the obtained RBFNs is compared in the following figures.

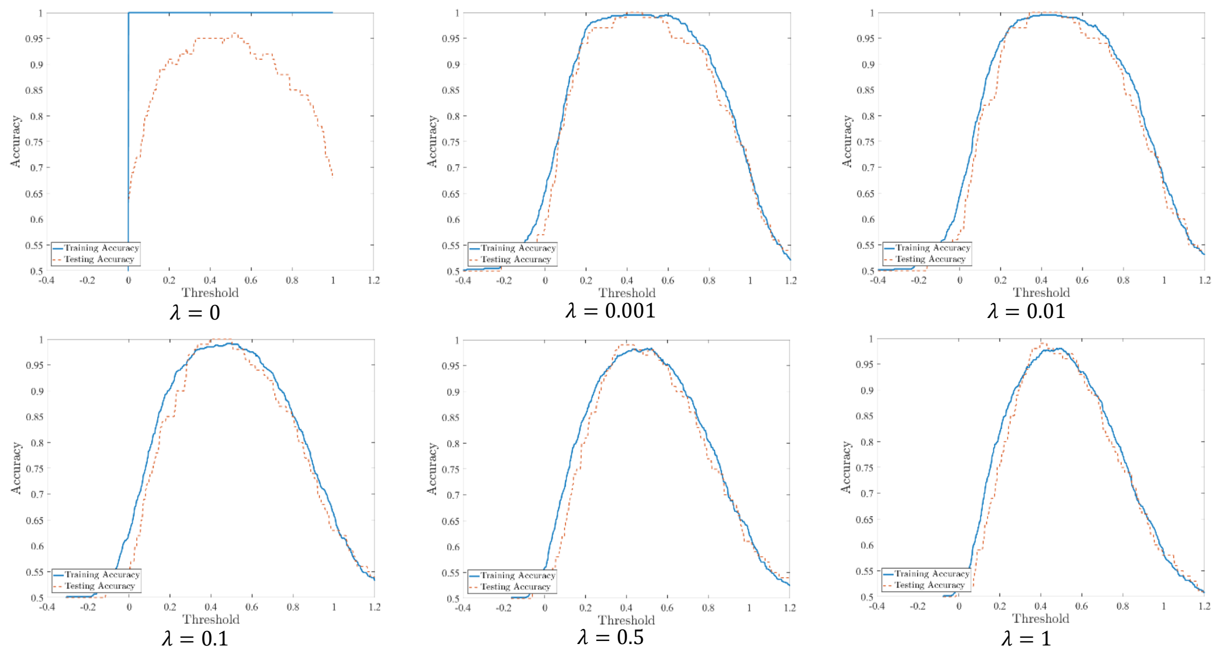


Figure 5 Comparison of the performance of RBFNs that are yield under different regularization strength.

It can be found from the above figures that, in the case of no regularization ($\lambda = 0$), the training accuracy jumps to and remains at 100% once the threshold passes 0. This confirms the fact that RBFN obtained via Exact Interpolation does pass through each individual training sample exactly. However, such RBFN performs poorly on the test set, i.e. RBFN overfits on the training set.

After applying regularization, the overfitting issue is largely reduced, and the test accuracy could also achieve 100% when the threshold is around 0.4~0.6. Meanwhile, it is also noted that if the regularization strength goes too large, e.g. when $\lambda \geq 0.1$, the margin of threshold within which the RBFN could completely separate the test set becomes narrower.

b) When randomly select 100 training samples as the centers of RBFN and fix the widths according to $\sigma = d_{max}/\sqrt{2M}$, the classification performance of the yield RBFN is displayed in Figure 6. It can be observed from this figure that RBFN obtained under this setting performs much worse than those derived in a).

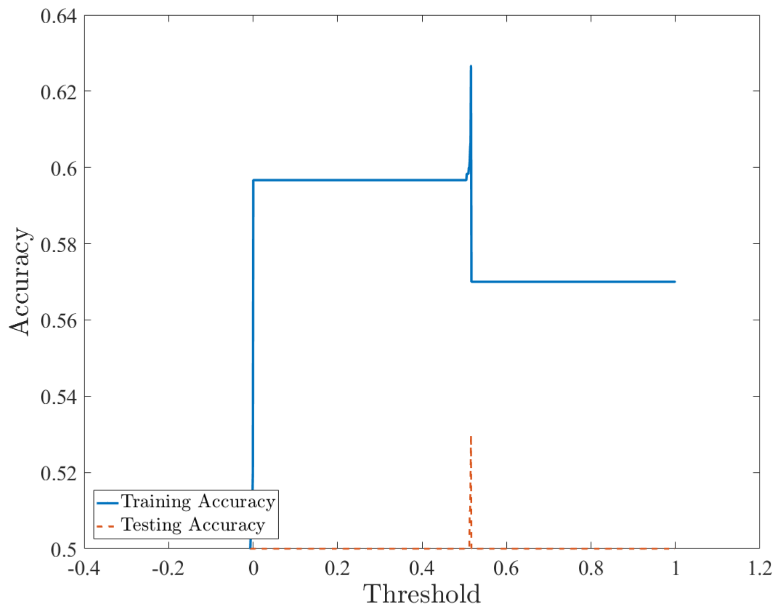


Figure 6 Performance of the RBFN with 100 centers and width $\sigma = 2.16$.

We then vary the value of width from 0.1 to 10000 and plot the resulted best classification accuracy under each width in Figure 7. It is found that the classification results are directly affected by the selection of σ: either a small or large σ would deteriorate the performance greatly. Furthermore, $d_{max}/\sqrt{2M}$ might not be a good universal width estimation especially when the data is of high dimension.

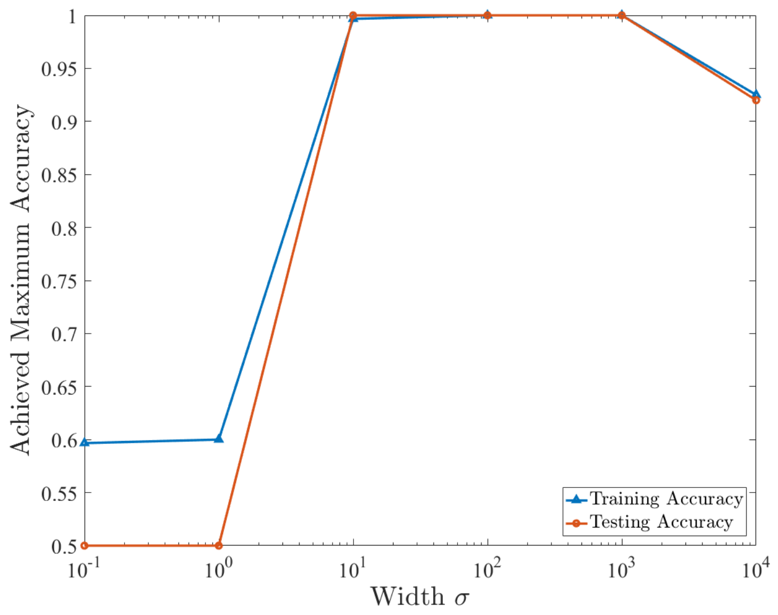


Figure 7 The performance of RBFN is sensitive to the selection of width.

c)

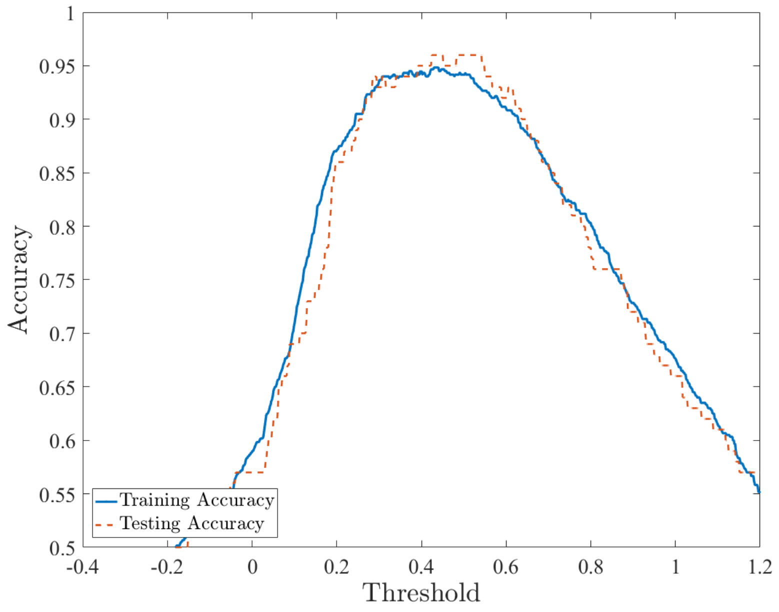


Figure 8 Performance of the RBFN with 2 centers and width $\sigma = 100$.

Comparing Figure 8 with Figure 6, it is clear that despite having only 2 centers, the resulted RBFN outperforms the 100-center RBFN in section b). This is because the 100 centers in b) are randomly selected, while here the 2 centers are determined in a meaningful manner using 'k-mean clustering', and it turns out that the obtained 2 centers are very close to the mean of training images as shown in the following figure.

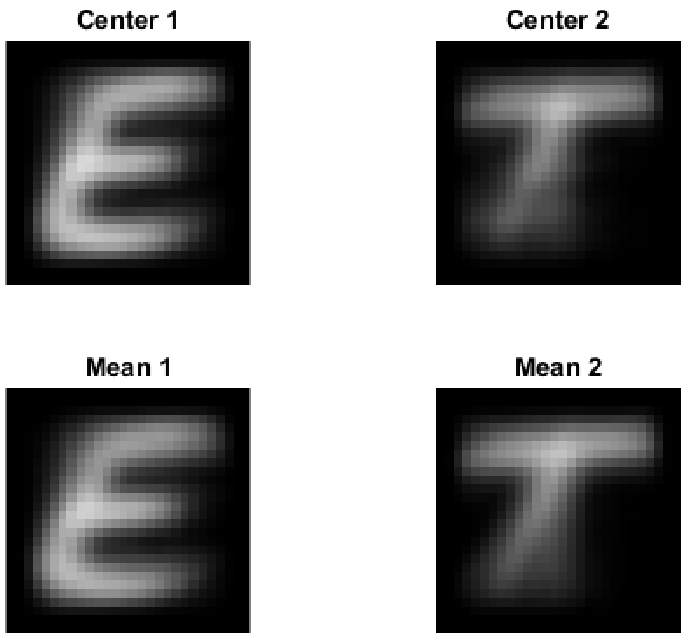


Figure 9 Comparison between the centers discovered by K-mean clustering and the mean of training samples.

Q3 Self-Organizing Map (SOM) (20 Marks)

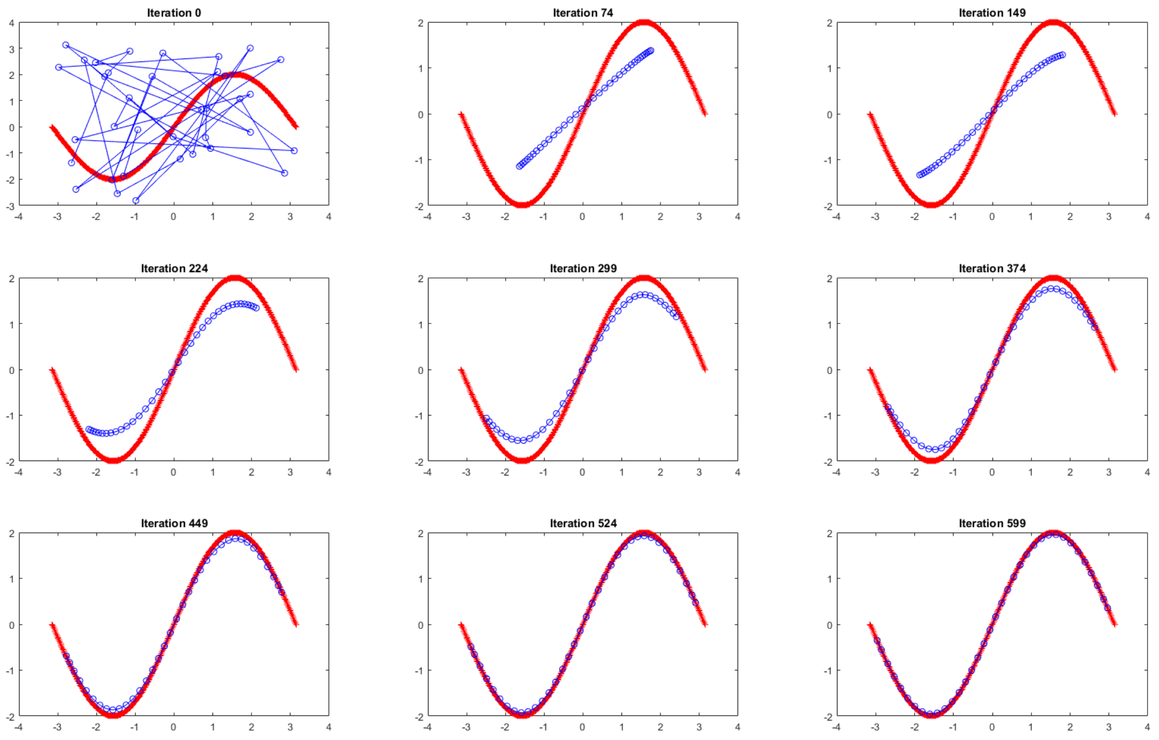
For n-dimensional input space (n=2 in a & b), n=784 in c)) and m (m=1×36 in a), m=6×6 in b), m=10×10 in c)) output neurons:

1. Sampling: choose an input vector x from the training set;
2. Determine winner neuron $i(x)$: $i(x) = \text{argmin}_k ||w_k - x||$ (Euclidean distance)
3. Update the weights of all the neighborhood neurons j of the winner neuron $i(x)$:

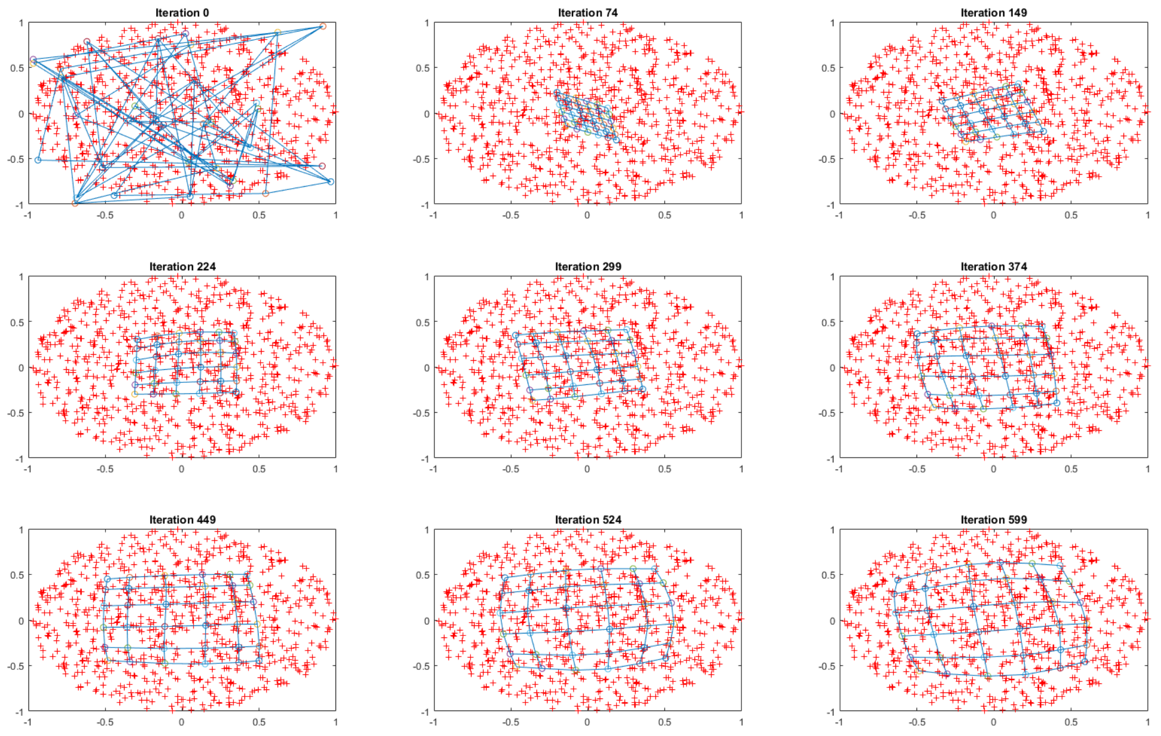
$$w_j(n+1) = w_j(n) + \eta(n)h_{j,i(x)}(n)\big(x - w_j(n)\big), j = 1,2,\dots,m$$

4. If stopping criterion met, STOP; Otherwise, go to (1).

a)



b)



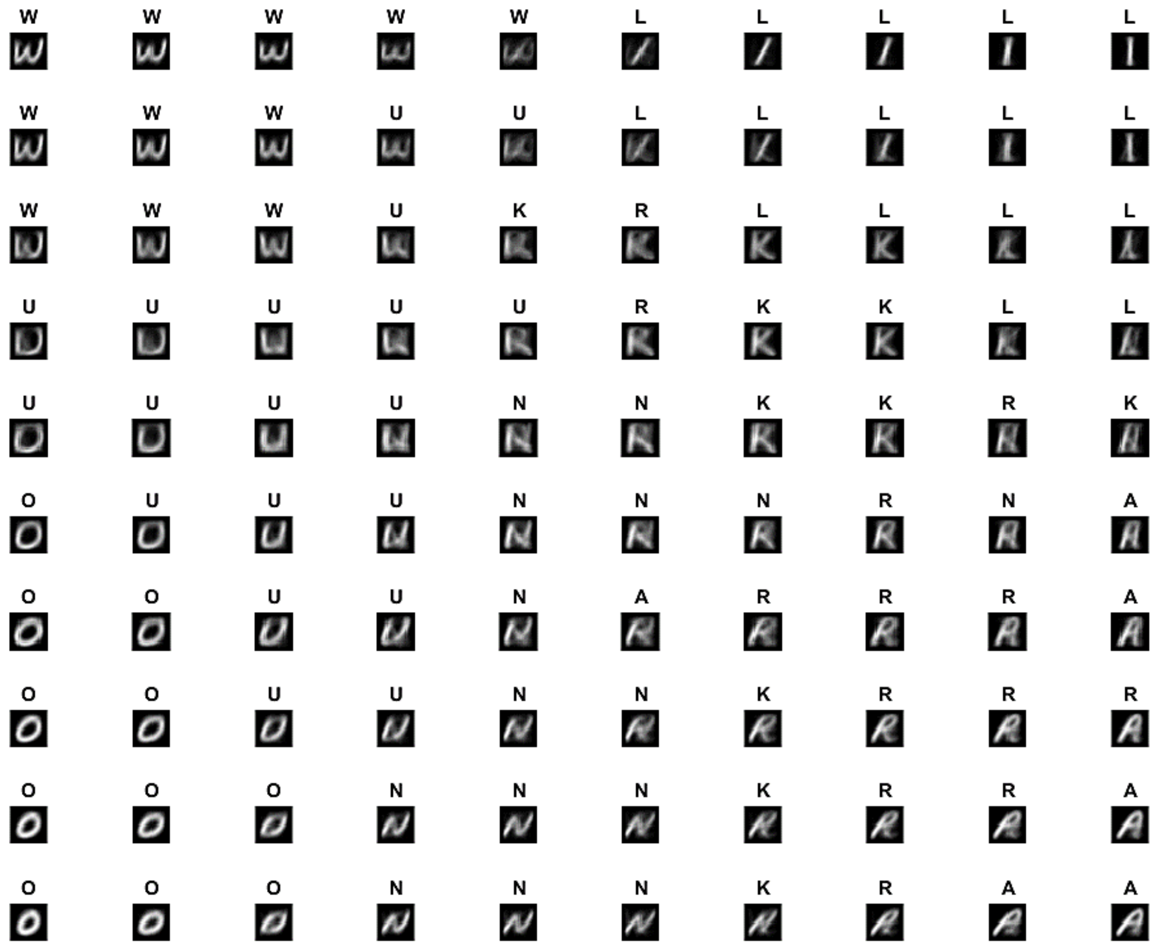
c) Generating the conceptual/semantic map:

- Choose one neuron from the output layer;
- Search the training set and determine which sample is closest to the particular neuron;
- Label the neuron using the class label of the training sample;
- Repeat the above procedure for all output neurons.

W	W	W	W	W	L	L	L	L	L
W	W	W	U	U	L	L	L	L	L
W	W	W	U	K	R	L	L	L	L
U	U	U	U	U	R	K	K	L	L
U	U	U	U	N	N	K	K	R	K
O	U	U	U	N	N	N	R	N	A
O	O	U	U	N	A	R	R	R	A
O	O	U	U	N	N	K	R	R	R
O	O	O	N	N	N	K	R	R	A
O	O	O	N	N	N	K	R	A	A

Clustered regions can be seen clearly from this map (class 1 ‘E’ and class 6 ‘T’ are omitted in this demonstration). Characters ‘W’, ‘U’, ‘O’, ‘L’ are all compactly clustered, while ‘N’, ‘K’, ‘R’, ‘A’ are relatively messed with each other since they share more visual characteristics, i.e. resemble each other. Due to random initialization, the results are different for different runs.

The corresponding weights are visualized in the following figure. There is a clear connection between the weight of a neuron and the label assigned to it. It is especially illuminating to observe the vagueness of neurons at the boundaries between clusters.



Test the SOM:

- Choose one sample form the test set;
- Find out the winner neuron in the SOM which is closest to that particular sample;
- Label the test sample using the label of the winner neuron;
- Repeat the above procedure for all test samples.

Accuracy on training set: 62.62%

Accuracy on test set: 65.25%

Character	N	U	R	A	L	W	O	K
NO. of Test Errors	23	8	26	29	7	10	7	29

It is noted from the above Table that handwritten characters ‘A’, ‘K’, ‘R’, ‘N’ are mostly mis-classified. This is due to fact these handwritten characters are quite similar to each other, making it more difficult to separate them. Increase the epochs can enhance the average accuracy. Tuning other parameters like the SOM lattice size, learning rate, etc. would also yield different results. You are encouraged to conduct more trials and discuss possible findings.