Problem 1

Table 1: Parameters of Training BP Network to Fit Iris data

Network parameters	
Topology	$(4+1_{Bias})$ — $(2+1_{Bias})$ — 3
Transfer function	tanh with slope of 1
Learning parameters	
Initial weights	drawn from U$[\frac{-1}{\sqrt{NPE}}, \frac{1}{\sqrt{NPE}}]$
Learning rate (α)	0.01
Momentum	0.7
Epoch size ($Epoch$)	100
Stopping criteria	RMSE$<$ 0.2 or learn count (t) $>$ 500 $\times$ 100
Error measure	$1 - \frac{Number\ of\ CorrectlyClassified\ Inputs}{Number\ of\ All\ Inputs}$(1-Accurancy) and RMSE
Input / output data, representation, scaling	
# training samples (N_{tr})	100
# test samples (N_{tst})	50
Scaling of inputs	already scaled
Scaling of outputs	set the maximum element of each column to 1, others to 0

Figure 1: Training and Test Accurancy for Each Fold

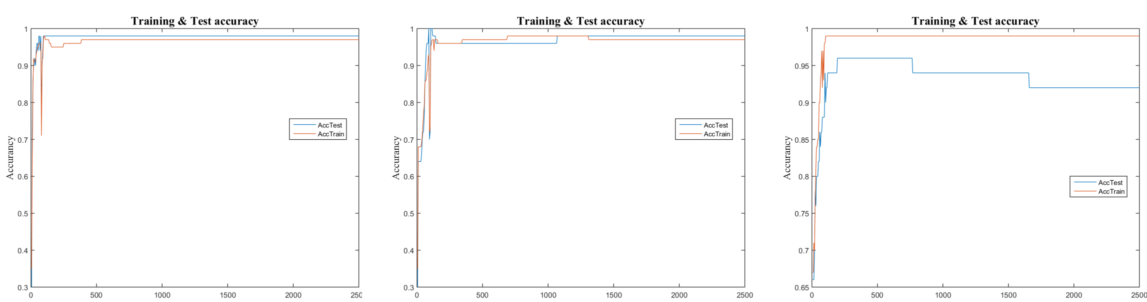
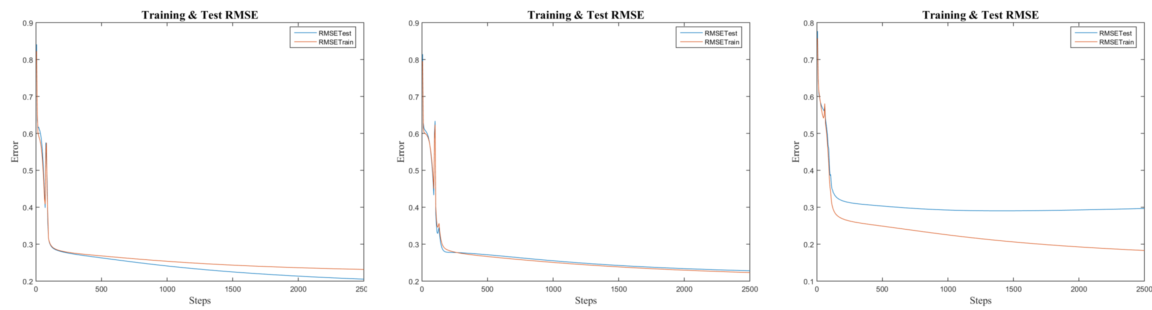


Figure 2: Training and Test RMSE for Each Fold



This time we used same parameters (except epoch size) as HW4. For cross-validation, all the data is randomly splited into 3 sets. For each fold one of these sets is used as test set, and other sets are merged and used as training set. Then we perform training and test on each fold seperately.

In HW4 we had already used confusion matrix to display the result. This time we just used the same procedure and format. We can see that the performance of each fold is similar, which indicates that the network is well generalized.

Note that I didn't give the "actual vs desired" plot. That's because I think the plot cannot give any information which is not given by confusion matrix, and is much more messive. In the text of HW5, it's said that the plot could show the "localization" of misclassified data, but this "localization" doesn't make any sense. The order of the data is trivial in the classification, furthermore, the order is rearranged in the cross validation. If we want to know the real "localization" of misclassifications, we need to see the INPUT of them, but we cannot plot that since the Input is 4-dimensional.

Table 2: Results of the cross validation(training set)

	Fold 1			Fold 2			Fold 3		
Desired / Actual	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
Class 1	35	0	0	33	0	0	32	0	0
Class 2	0	34	2	0	30	1	0	32	0
Class 3	0	1	28	0	2	34	0	1	35
Overall Accurancy	0.97			0.97			0.99		

Mean Accurancy of Each Fold: 0.978
Std of Accurancy: 0.0115

Table 3: Results of the cross validation(test set)

	Fold 1			Fold 2			Fold 3		
Desired / Actual	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
Class 1	15	0	0	17	0	0	18	0	0
Class 2	0	14	0	0	18	1	0	16	3
Class 3	0	1	20	0	0	14	0	1	12
Overall Accurancy	0.98			0.98			0.92		

Mean Accurancy of Each Fold: 0.96
Std of Accurancy: 0.0346

*Class 1 is Setosa, Class 2 is Versacolor, Class 3 is Virginica

Problem 2

Table 4: Parameters of Training BP Network to perform the equalization of the communication channel

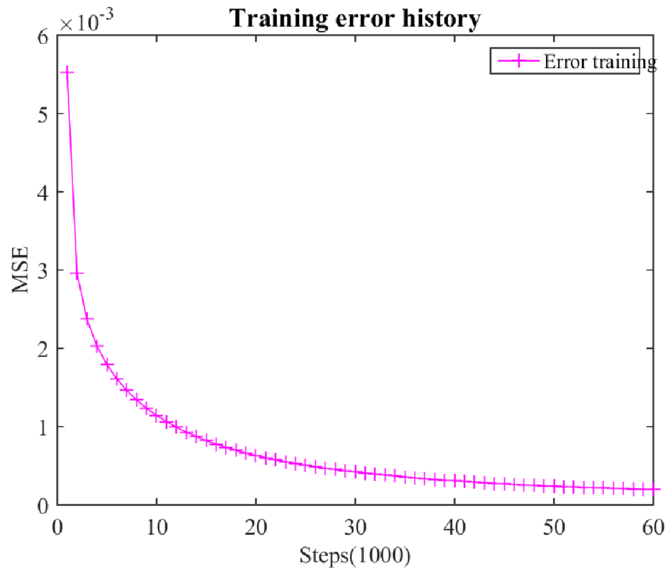
Network parameters	
Topology	$(1 + 1_{Bias})$ — $(10 + 1_{Bias})$ (otherwise notified) — 1
Transfer function	tanh with slope of 1(hidden layer) or linear(output layer)
Learning parameters	
Initial weights	drawn from U$[\frac{-1}{\sqrt{NPE}}, \frac{1}{\sqrt{NPE}}]$
Learning rate (α)	0.01, otherwise notified
Momentum	0.9
Epoch size ($Epoch$)	20
Stopping criteria	error (Err_{RMSD}) < 0.005 or learn steps =60,000
Monitoring frequency of error measure	Every 1000 learn steps
Error measure(Err_{RMSD})	Square root of the sum of $(D - y)^2$ that averaged over all training or testing samples (see formula (1) in problem 2)
Input / output data, representation, scaling	
Training samples ($S(n)$)	$2sin(\frac{2\pi n}{20})$, n=1:20
Test sample set 1 ($s_1(n)$)	$0.8sin(\frac{2\pi n}{10}) + 0.25cos(\frac{2\pi n}{25})$,n=1:50
Test sample set 2 ($s_2(n)$)	50 random numbers drawn from a zero mean, unit variance normal distribution
Scaling	none

For all the output verses desired output plots, we used z(nT) as input(x axis), and $\hat{s}$(nT), s(nT) as output(y axis).
It's widely accepted that when doing regression, use linear output layer could achieve better performance. So this time we didn't scale the data, and use a linear transfer function in output layer correspondingly.
In this problem, we only care the performance of the neural network, so we just use the input and output to

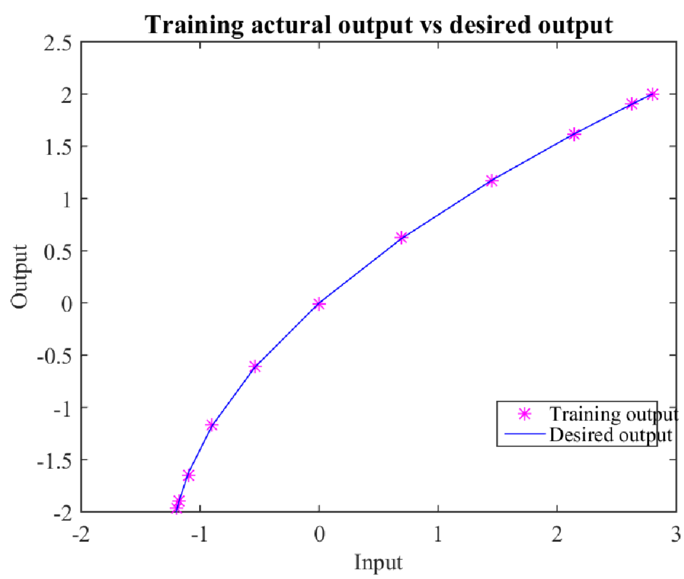
display the result, instead of the "original signal".

2.1) Training

The MSE history is shown below:

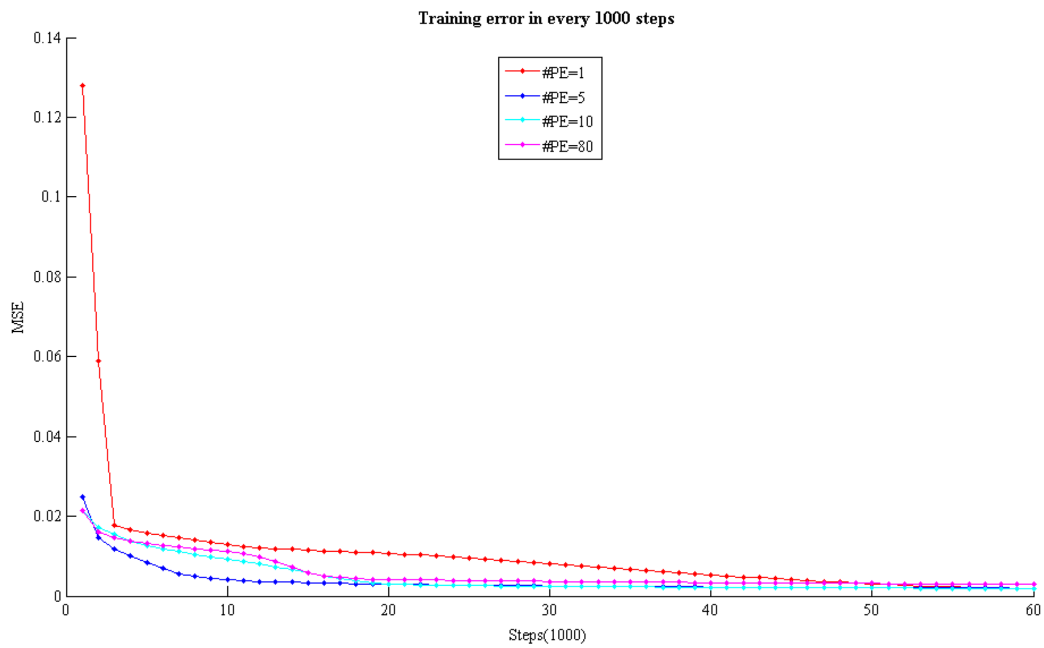


The training output versus desired output is shown as figure below.



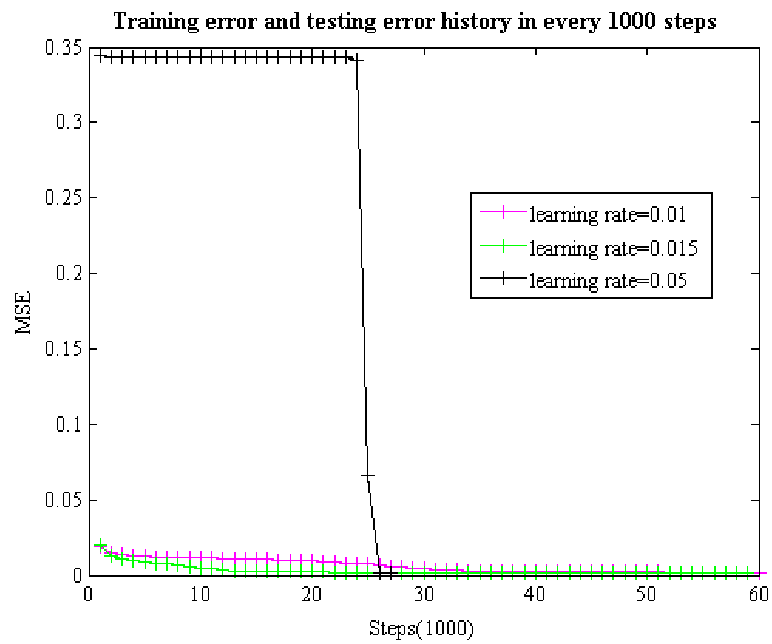
2.1.1) Different hidden PE number

We first tried to use different number of hidden PE. By comparing the history, increasing the number of hidden PE did not significantly change the reduction of MSE, except for #PE=1. For #PE=1, it took longer time to reduce the MSE. The final error rate for #PE=1, #PE=5, #PE=10, #PE=80 are 0.0004, 0.0003, 0.0006, 0.0009. We noticed that the final MSE varied during repeating the training, so these values can only show that their final result were qualitatively the same.
In this case, the final output of training were almost overlapped to #PE=10, so the training data vs desired data for each condition is not shown here.

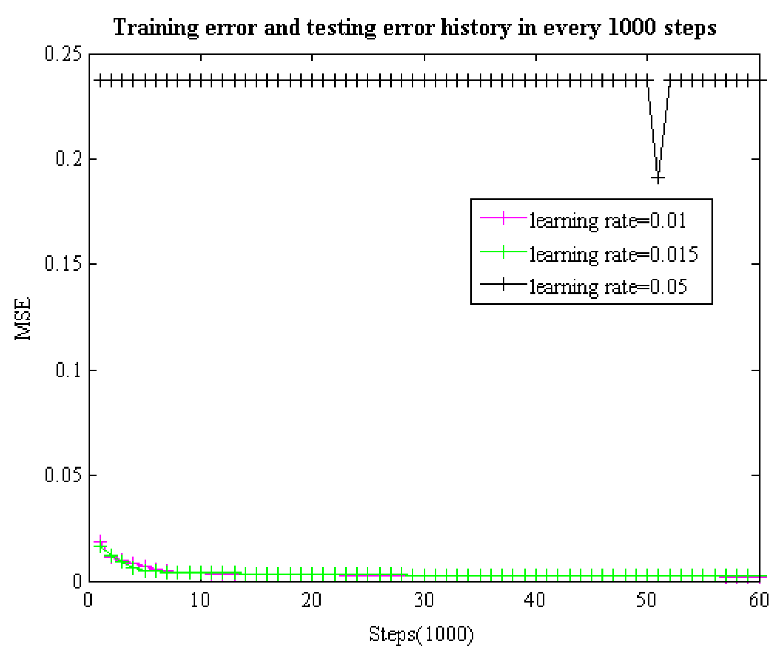


2.1.2) Different learning rate

We used 0.01, 0.015 and 0.05 as learning rate. Interestingly, we found two patterns when learning rate is equal to 0.05. As shown below, the large learning rate can help the neuron network get to a small MSE much faster than the others and meet the criteria of stopping. The learning met the MSE less than or equal to 0.0015 criteria at 27000 step.



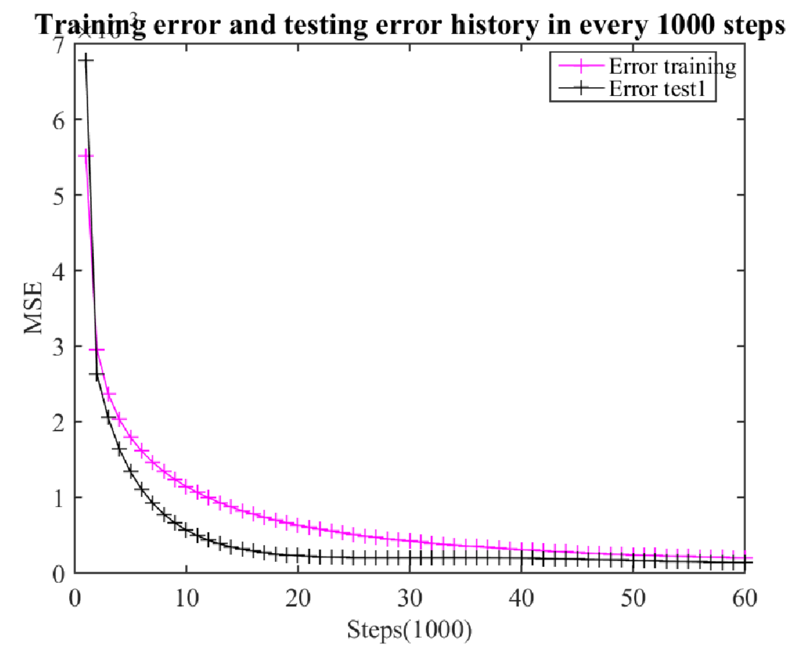
However, in some other cases, using 0.05 as learning rate cannot converge the MSE, which is shown as figure below:



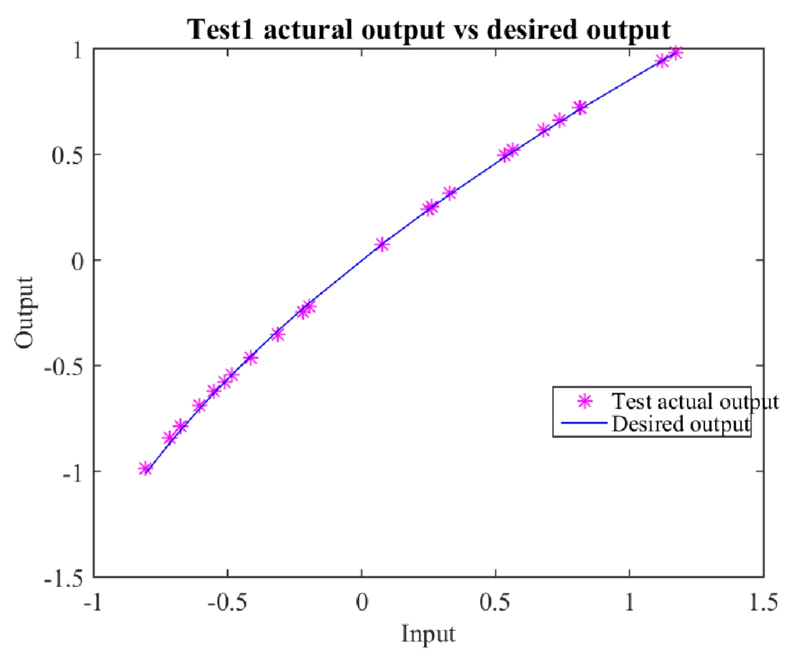
And the learning failed.

2.2.1) Test the memory using $s_1(n)0.8sin(\frac{2\pi n}{10}) + 0.25cos(\frac{2\pi n}{25})$

Test group 1: $s_1(n)0.8sin(\frac{2\pi n}{10}) + 0.25cos(\frac{2\pi n}{25})$
The MSE history is shown as below.

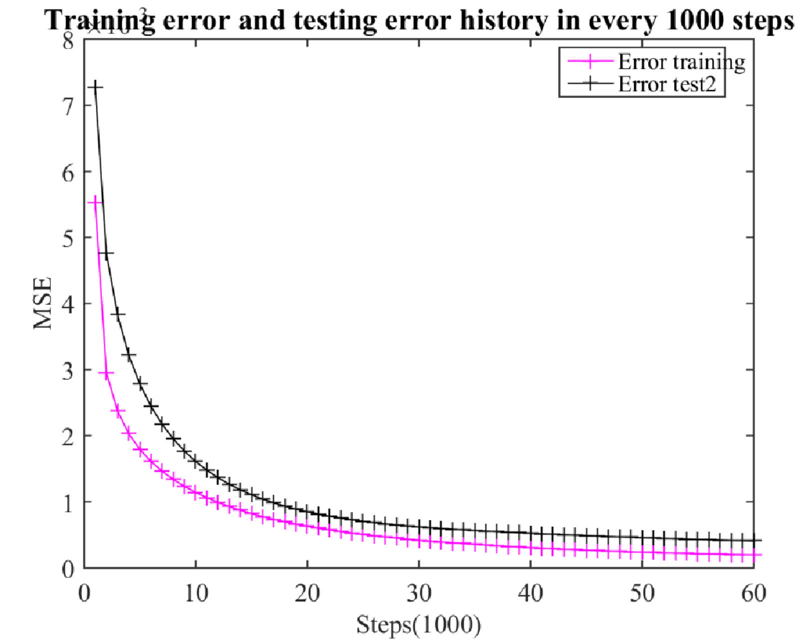


The training, testing output versus desired output is shown as figure below, where the final MSE is 0.0003 for testing, 0.0004 for training.



2.2.1) Test the memory using test group 2

Test group 2: $s_2(n)$), 50 random numbers drawn from a zero mean, unit variance normal distribution.
The MSE history is shown as below. The final MSE for testing group is 0.0006, for training group is 0.0004.



The training, testing output versus desired output is shown as figure below: