Answer HW05

Eugen Hruska, Elliot Smith

Problem1

1.A)

Network Details

• Network Parameters – Topology: (4 + 1 bias) - (3 + 1 bias) - 3 – Transfer Function: Hyperbolic Tangent (slope = 1)

• Learning Parameters – Initial Weights: drawn randomly from Uniform[-0.1, 0.1] – Learning Rate: 0.05 – Momentum: 0.4 9 – Epoch Size: 10 – Stopping Criterion: 10,000 Learning Steps – Error Measure for Stopping Criterion: N/A

• Input / Output Data, Representation, Scaling: - 3 fold cross validation – Number of Training Samples: 100 randomly sampled from Iris Training and Test Data – Number of Testing Samples: 50 from Iris Testing and Test Data, which were not used in Training – Scaling on Inputs: None – Scaling of Outputs: None

• Parameters and Error Measures of Performance Evaluation – Error of Function Fit: Absolute summed difference – Number of Learning Steps Performed in each fold: 10,000 Learning Steps – Learning Rate: 0.05 – Monitoring Frequency: Every 100 Learning Steps

We got excellent results in each fold for both the training and testing data! Very quickly, we were able to attain a small error. The Classification accurracy is not 100 %, but between 97 and 98 %. We are very happy with this result. We experimented with many different buildings, particularly tweaking the learning rate,number of PEs in the hidden layer and the batch size; however this network gave us the best result!

1.B)

learning history plot in figure 1. Description on plot. The total error descreases fast and than stays low.

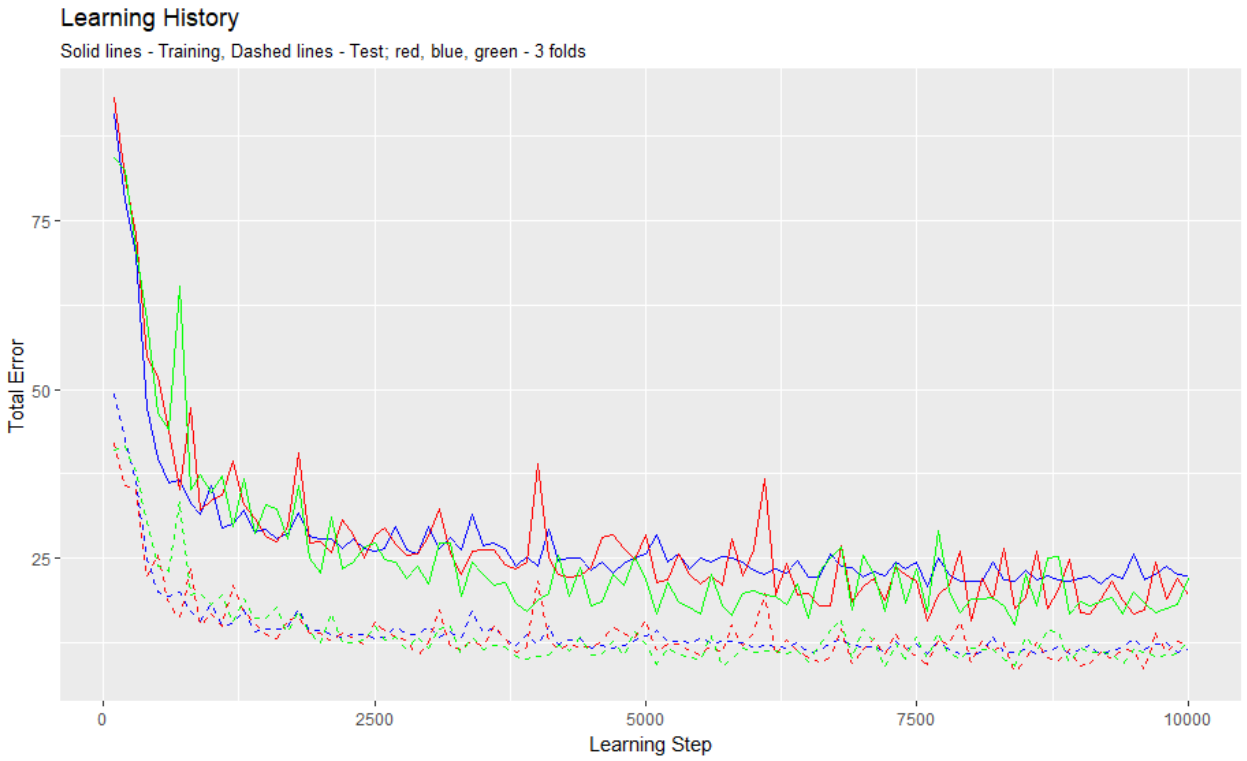

Figure 1: Learning history

Description figure 2: desired output (top three rows) vs actual output (bottom three rows)", x axis are the 50 randomly sampled test data, yellow - no output, blue color - output, one plot for each of the 3 folds

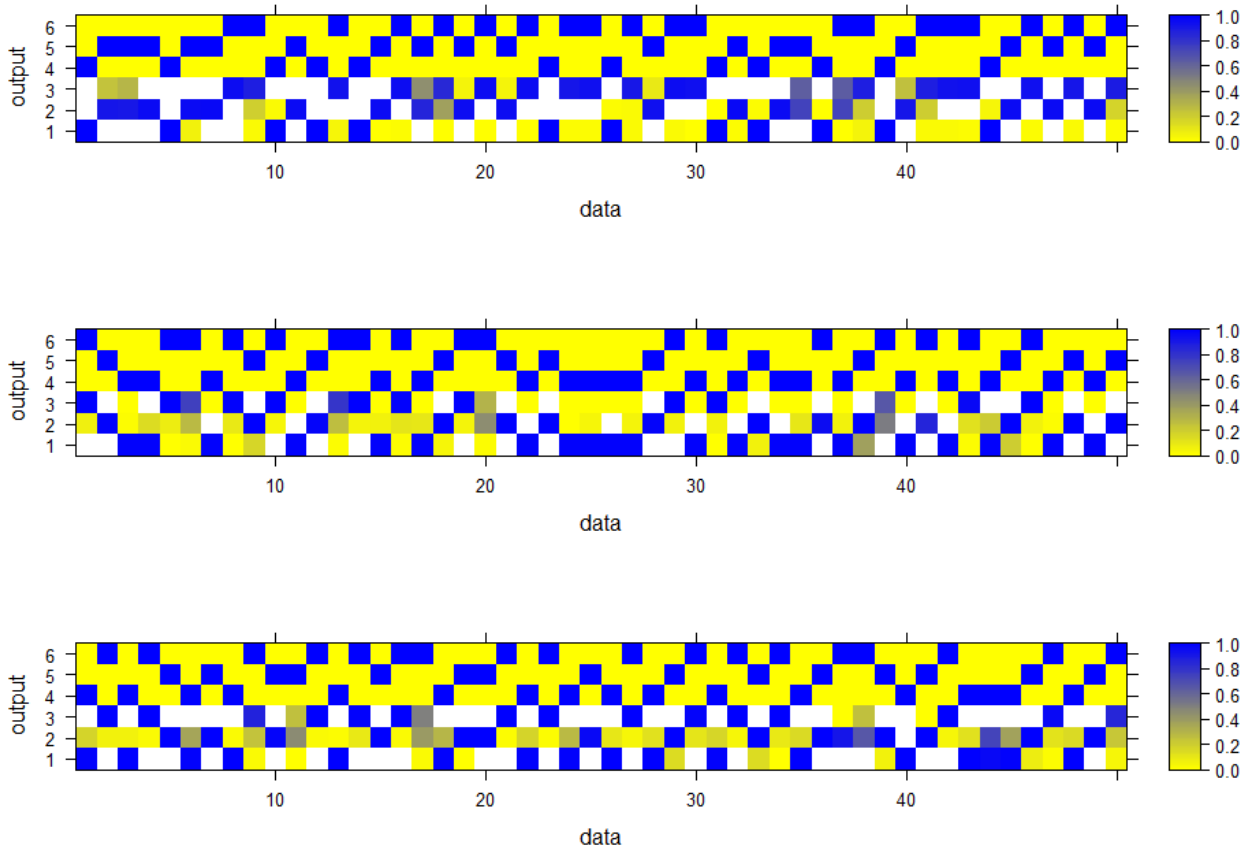

Figure 2: Learning history

Show 3 plots for desired vs actual output "HW05-P1-B-desired-vs-actual-fold*"
Confusion matrices, column - predicted class, row - True class class 1 -Setosa, class 2 - Versacolor, class 3 - Virginica :

confusion matrix 1. fold training:

true class/class predicted	1	2	3
1	38	0	0
2	0	31	1
3	0	2	28

confusion matrix 1. fold testing:

true class/class predicted	1	2	3
1	39	0	0
2	0	32	0
3	0	3	29

confusion matrix 2. fold training:

true class/class predicted	1	2	3
1	29	0	0
2	0	38	0
3	0	2	31

confusion matrix 2. fold testing:

true class/class predicted	1	2	3
1	20	0	0
2	0	39	0
3	0	3	32

confusion matrix 3. fold training:

true class/class predicted	1	2	3
1	33	0	0
2	0	34	1
3	0	1	31

confusion matrix 3. fold testing:

true class/class predicted	1	2	3
1	34	0	0
2	0	35	0
3	0	2	32

The 1. class (Setosa) has a perfect classification, no misclassifications, but there are misclassifications between the 2. (Versacolor) and 3. class (Virginica).

1.C)

Classification accuracies:

classification accuracy 1. fold training: 0.97
classification accuracy 1. fold testing: 0.971
classification accuracy 2. fold training: 0.98
classification accuracy 2. fold testing: 0.971
classification accuracy 3. fold training: 0.98
classification accuracy 3. fold testing: 0.981

The classification accuracy of the testing is almost identical to the classification accuracy of the training.

classification accuracy average and standard deviation training: mean 0.977 standard deviation:0.0058
classification accuracy average and sd testing: mean 0.974 standard deviation 0.0055

The accuracy of the testing is slightly smaller than for training, as explected. The standard deviation is small.

The three fold give very similar results, that means the generalization of the network is good.

Problem2

2.1

Network Details (general)

• Network Parameters – Topology: (1) input + (200) + (1)output – Transfer Function: Hyperbolic Tangent (slope = 1)

• Learning Parameters – Initial Weights: drawn randomly from Uniform[-0.1, 0.1] – Learning Rate: [0.18, 0.03] (for the different layers) Momentum: alpha - 0.9 – Batch Size: 20 – Stopping Criterion: 10,000 Learning Steps – Error Measure for Stopping Criterion: N/A • Scaling: - divide original input and output by 3 – Number of Training Samples: 10001 train and test data as described in Problem

• Parameters and Error Measures of Performance Evaluation – Error: MSE– Monitoring Frequency: Every 100 Learning Steps

We got excellent results in each fold the training and testing data! The MSE error drop in the learning history quickly and stays slow, we were able to attain a small MSE error.

In figure 3-5 the learning histories for three different network settings. Only the difference to the general network settings specified above are described. In each plot the MSE Error for the training data s(nT) and the test data(s1(nT) are plotted, the color legend is on top of the plot.

Network settings, 1. parameter set: Topology: (1) input + (200) + (1)output, Learning Rate: [0.18, 0.03] (for the different layers)

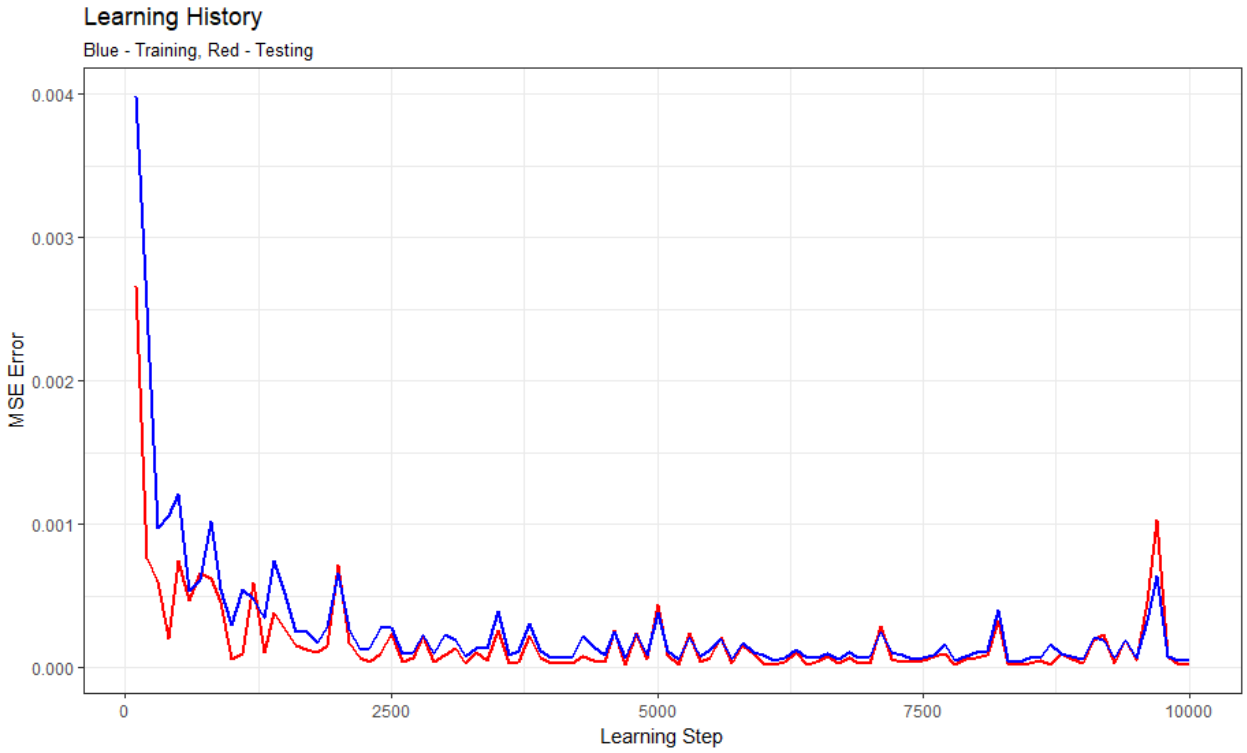

Figure 3:

Network settings, 2. parameter set: Topology: (1) input + (100) + (1)output, Learning Rate: [0.18, 0.03] (for the different layers)

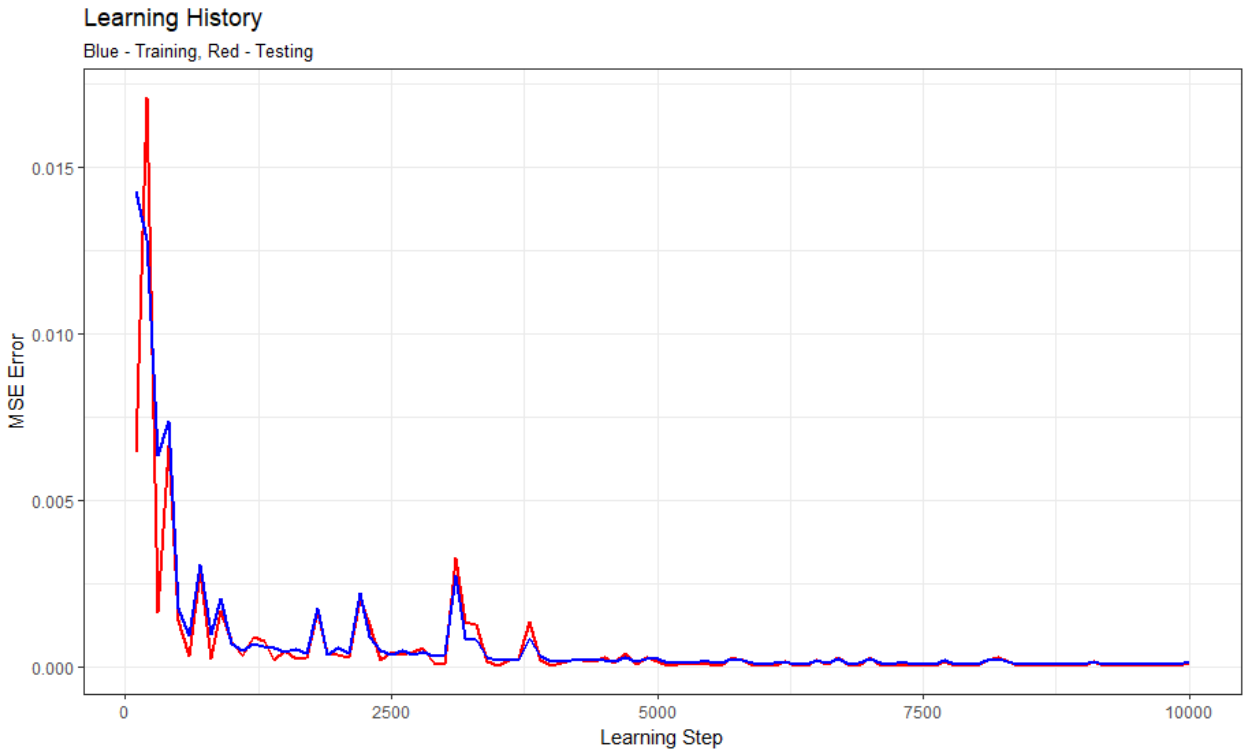

Figure 4:

Network settings, 3. parameter set: Topology: (1) input + (100) + (1) output, Learning Rate: [0.05, 0.01] (for the different layers)

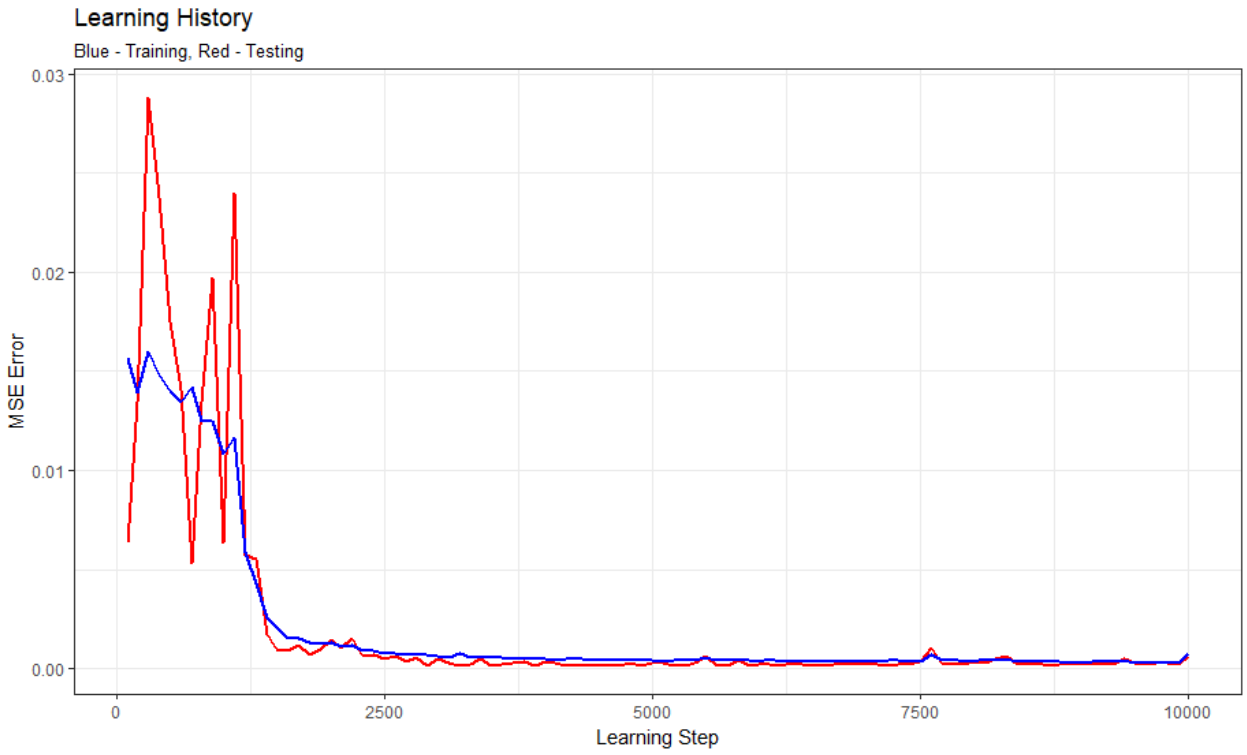

Figure 5:

All three parameters setting converge fast, but the second setting converges fastest.

2.2)

In figure 7, the desired vs Actual output for s1(nT), the color legend is on top of the plot. All three inputs/outputs are scaled, by dividing by 3.

The True Function and the Learned function are almost indistiquishable.

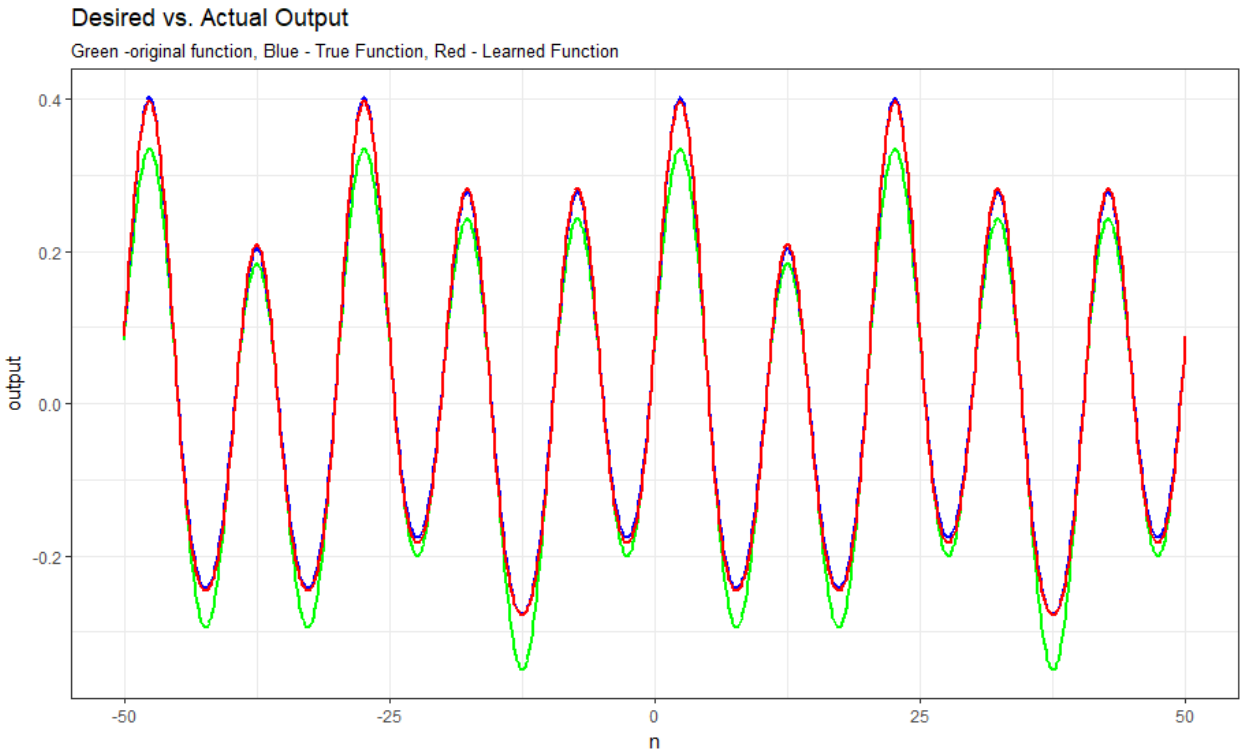

Figure 6:

In figure 8, the desired vs Actual output for s2(nT), the color legend is on top of the plot. All three inputs/outputs are scaled, by dividing by 3.

The True Function and the Learned function are almost indistiquishable, except when the input is above 1, despite being scaled (divided by 3). Is the scaling would be stronger, then even these couple points would fit in, but then most of the data points would have a very small variance.

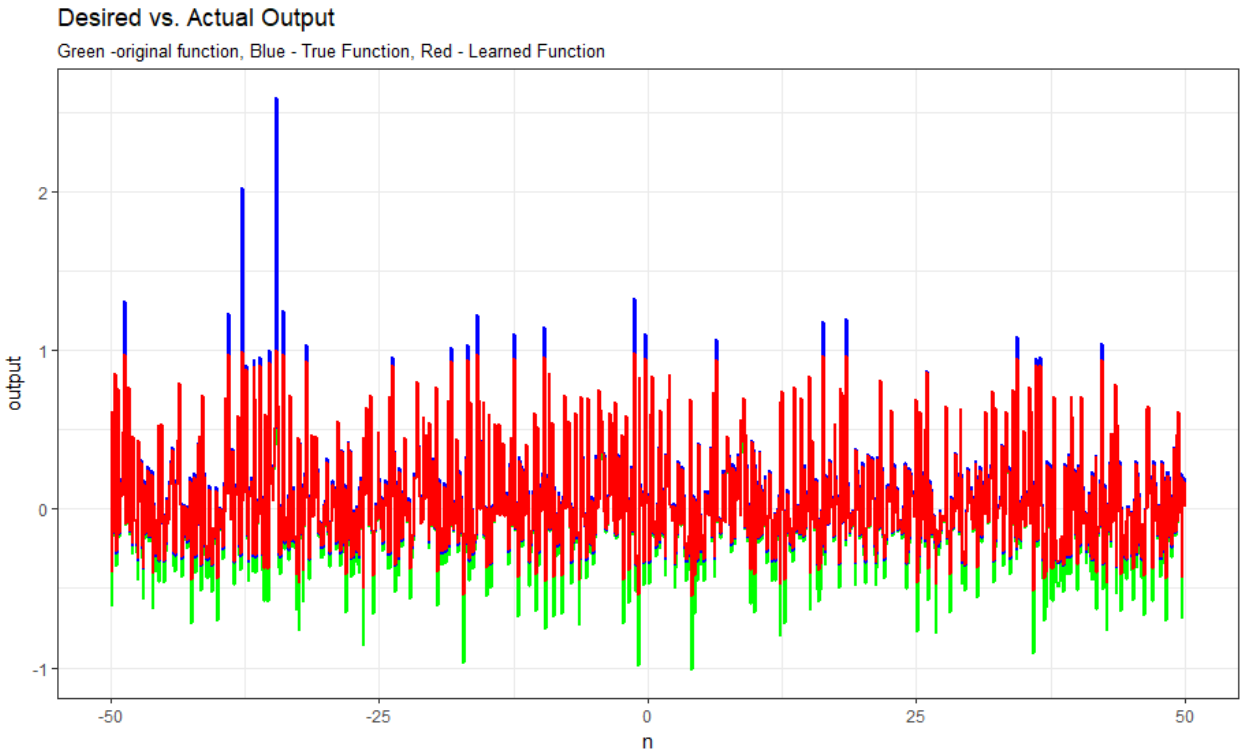

Figure 7:

Since both s1 and s2 were almost perfectly approximated we can conclude the Multilayer Perceptron is good way to equalize a memoryless communications channel.