# HW04

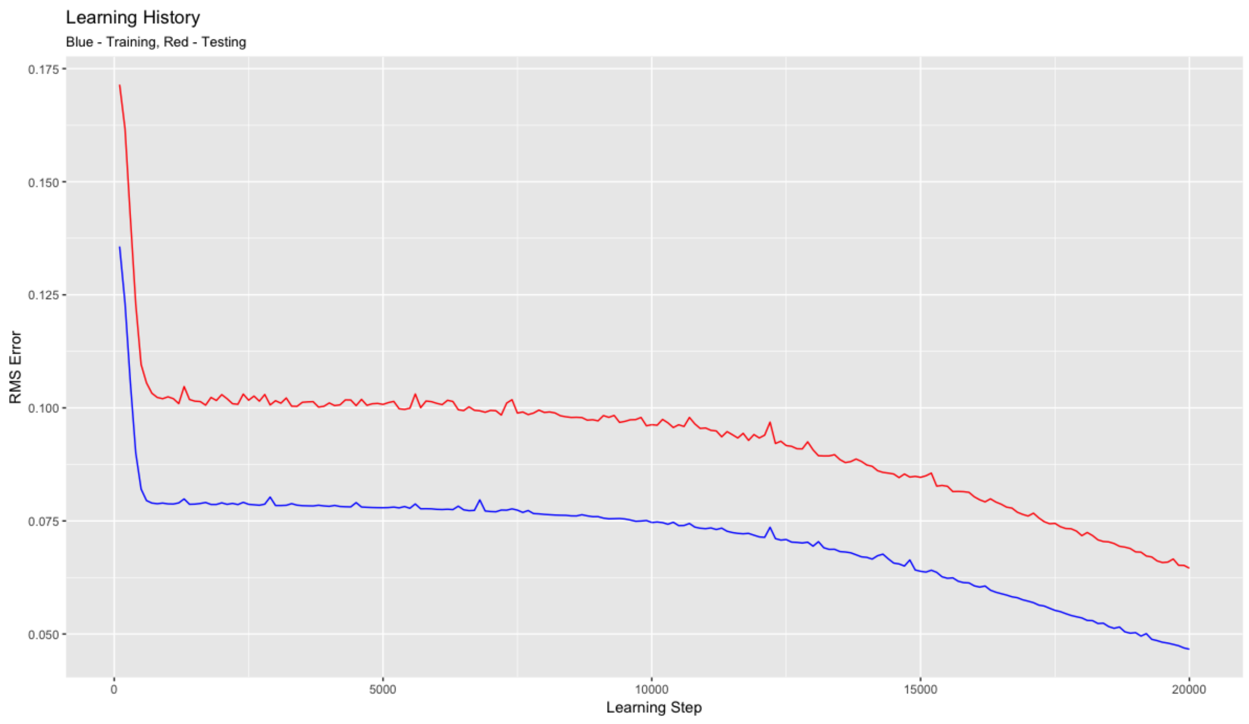
*Elliot Smith, Eugen Hruska, Warum Suriyanarayana*

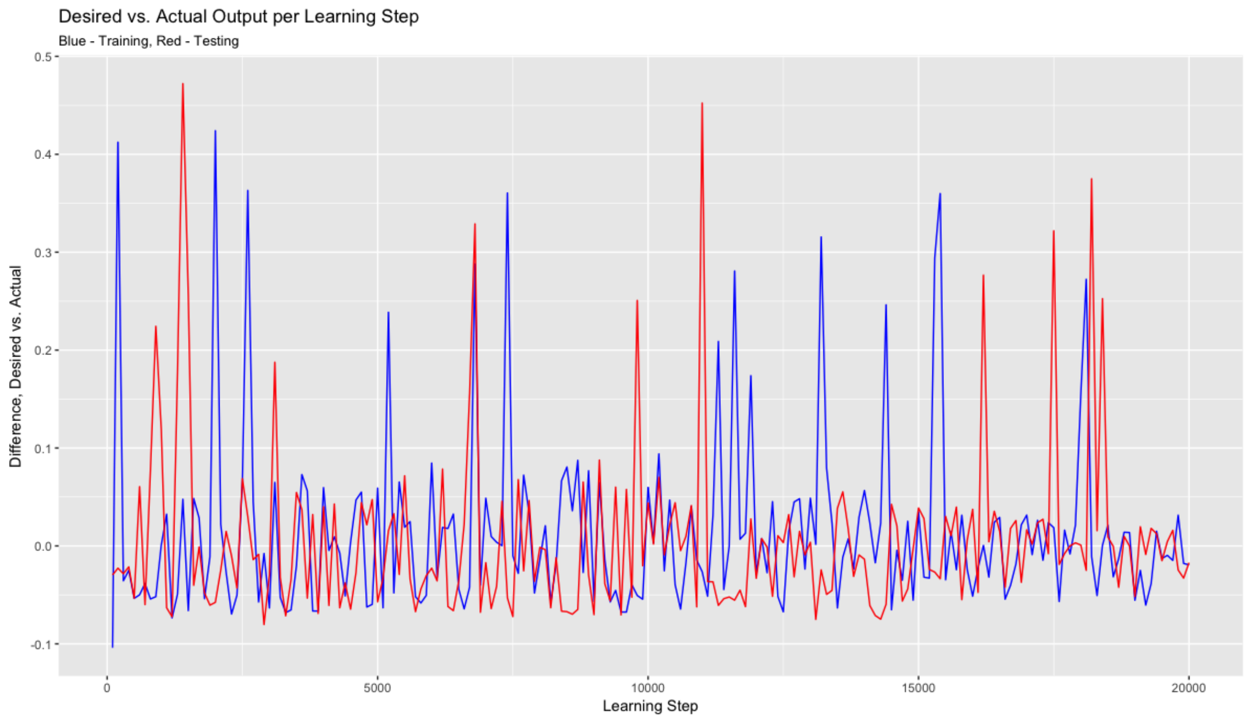
*2/20/2018*

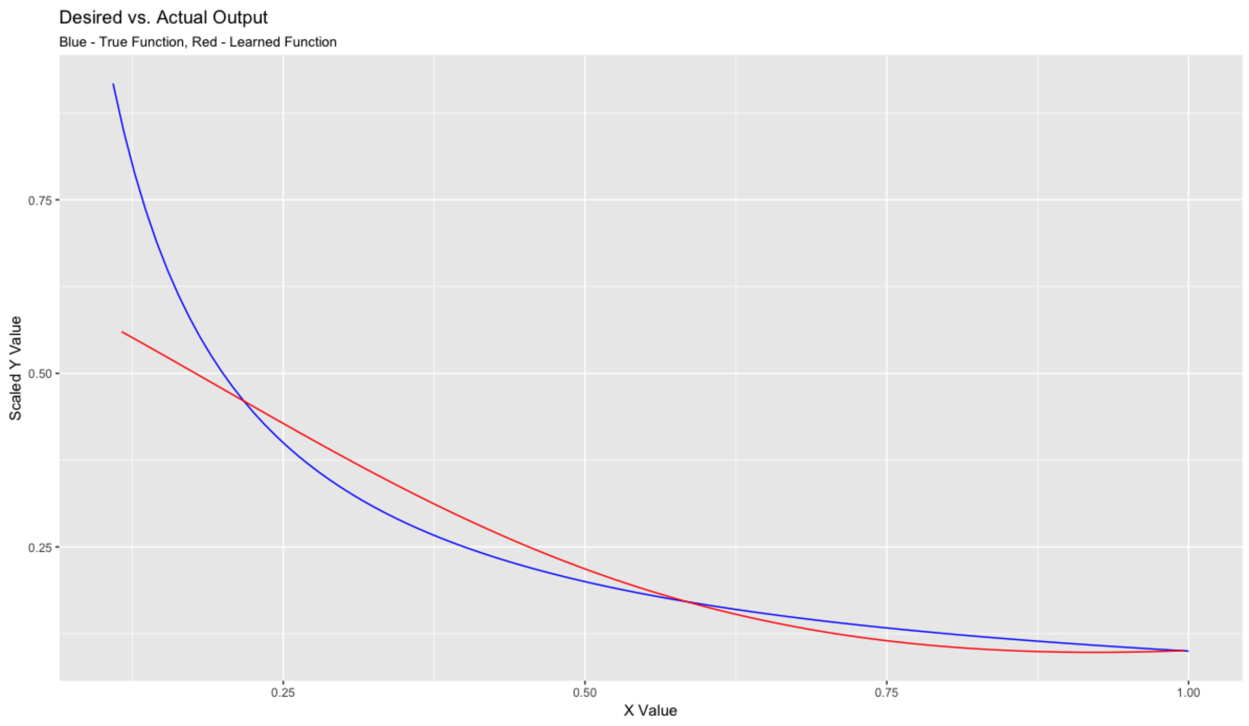
## Problem 2

#### Network Details

- Network Parameters
  - Topology: (1 + 1 bias) - (10 + 1 bias) - 1
  - Transfer Function: Hyperbolic Tangent (slope = 1)
- Learning Parameters
  - Initial Weights: drawn randomly from Uniform[-0.1, 0.1]
  - Learning Rate: 0.01
  - Momentum: None
  - Epoch Size: 20
  - Stopping Criterion: 20,000 Learning Steps
  - Error Measure for Stopping Criterion: N/A
- Input / Output Data, Representation, Scaling:
  - Number of Training Samples: 200 (drawn randomly from Uniform[0.1, 1.0])
  - Number of Testing Samples: 100 (drawn randomly from Uniform[0.1, 1.0])
  - Scaling on Inputs: None
  - Scaling of Outputs: Dividing by 10
- Parameters and Error Measures of Performance Evaluation
  - Error of Function Fit: Root Mean Square Error
  - Number of Learning Steps Performed: 20,000 Learning Steps
  - Learning Rate at End: 0.01
  - Monitoring Frequency: Every 100 Learning Steps


Learning History
Blue - Training, Red - Testing
RMS Error
0.175
0.150
0.125
0.100
0.075
0.050
0
5000
10000
15000
20000
Learning Step


Desired vs. Actual Output per Learning Step
Blue - Training, Red - Testing
Difference, Desired vs. Actual
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
0
5000
10000
15000
20000
Learning Step



### Conclusions

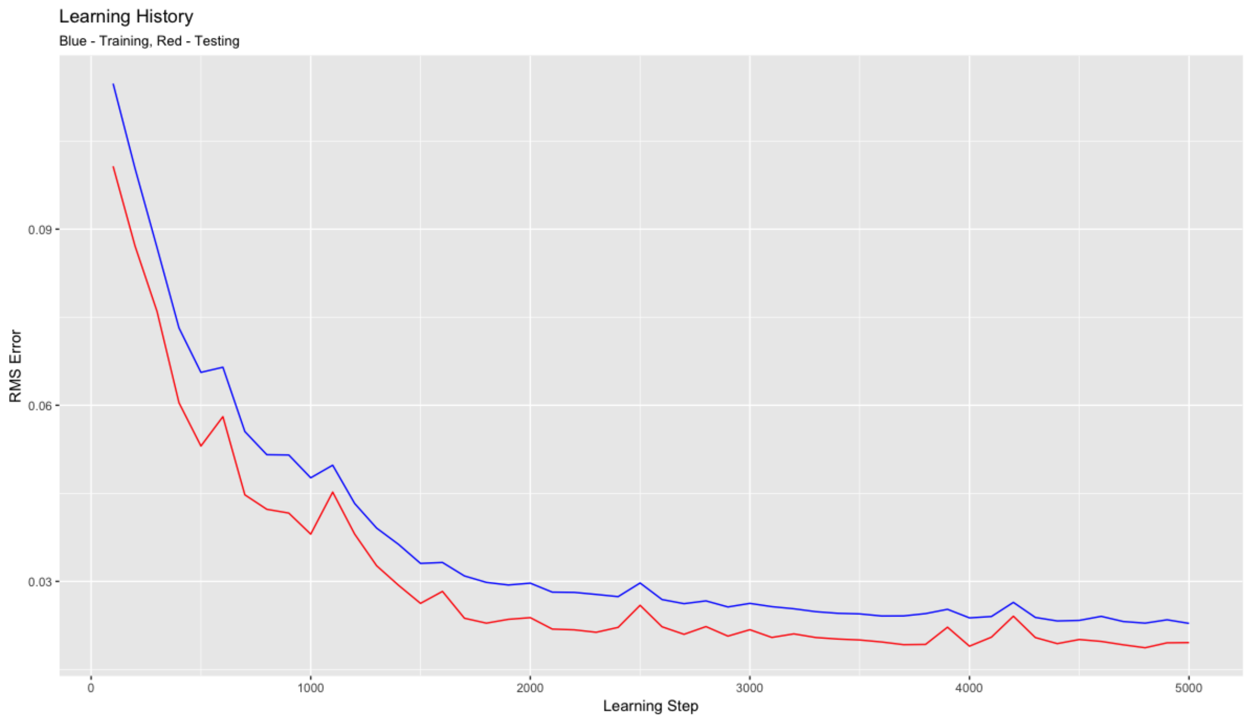
We had some fairly good results on this problem. In regards to our learning history, our training converged to a RMSE of just under 0.05 at 20,000 Training Steps, while the testing converged to just about 0.0625; we felt that this was a great result. Then when we compare the difference of our expected result and our recalled result for both training and testing, we see that we are slowly converging to zero. However, on a semi-regular cadence there seems to be spikes in difference that are corrected shortly after. Our memory does an okay job of replicating the function, however it is not perfect, but we are satisfied with the result.

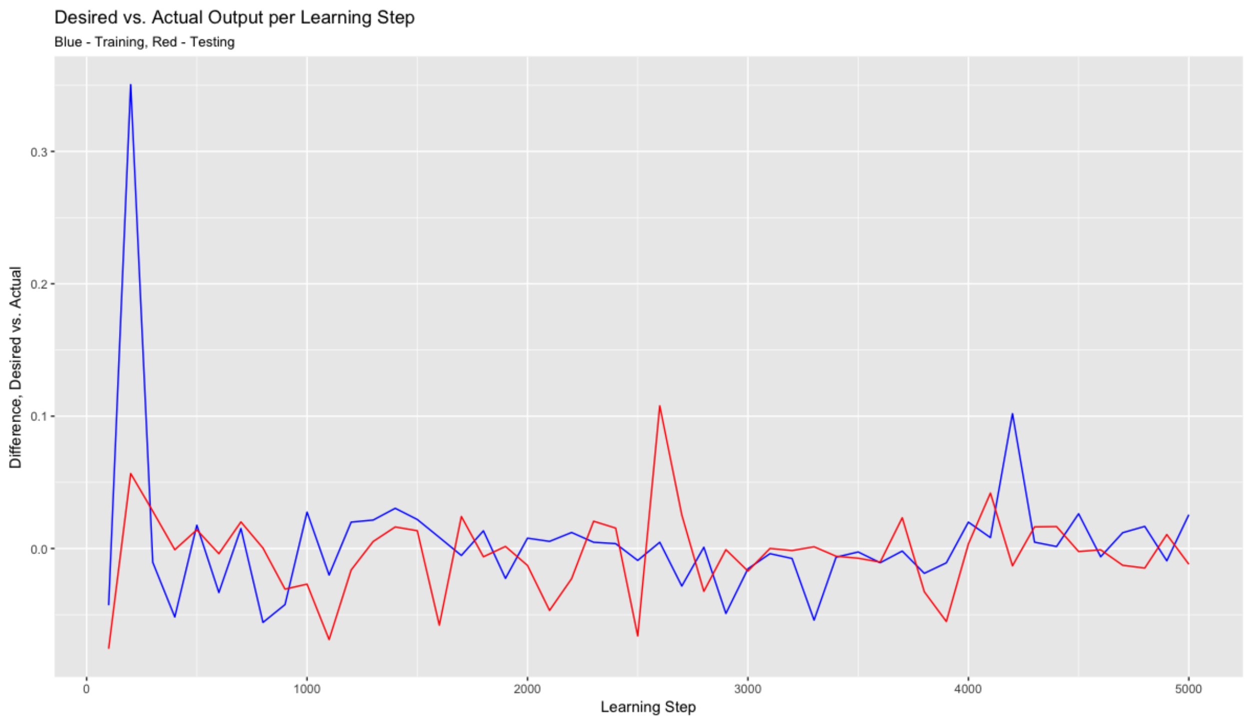
## Problem 3

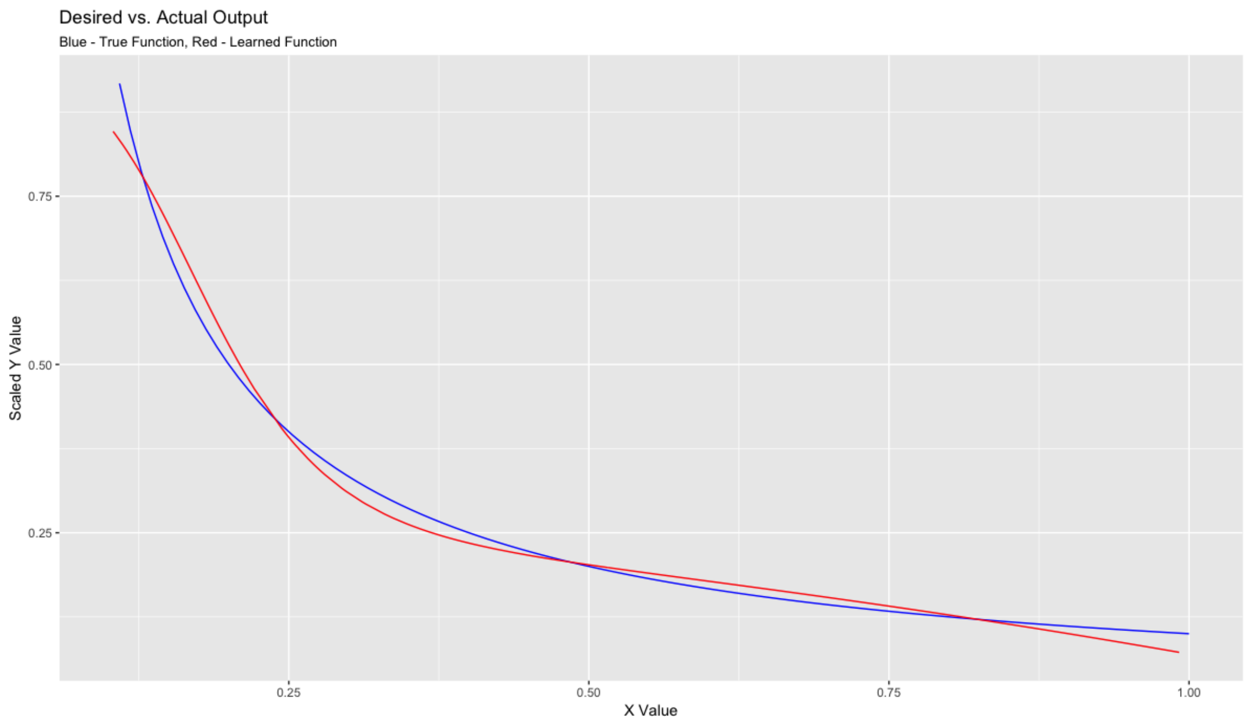
### Network 1 Details

- Network Parameters
  - Topology: (1 + 1 bias) - (10 + 1 bias) - 1
  - Transfer Function: Hyperbolic Tangent (slope = 1)
- Learning Parameters
  - Initial Weights: drawn randomly from Uniform[-0.1, 0.1]
  - Learning Rate: 0.05 and 0.0005
  - Momentum: 0.4
  - Epoch Size: 20
  - Stopping Criterion: 5,000 Learning Steps
  - Error Measure for Stopping Criterion: N/A
- Input / Output Data, Representation, Scaling:
  - Number of Training Samples: 200 (drawn randomly from Uniform[0.1, 1.0])
  - Number of Testing Samples: 100 (drawn randomly from Uniform[0.1, 1.0])
  - Scaling on Inputs: None
  - Scaling of Outputs: Dividing by 10

- Parameters and Error Measures of Performance Evaluation
  - Error of Function Fit: Root Mean Square Error
  - Number of Learning Steps Performed: 5,000 Learning Steps
  - Learning Rate at End: 0.0005
  - Monitoring Frequency: Every 100 Learning Steps







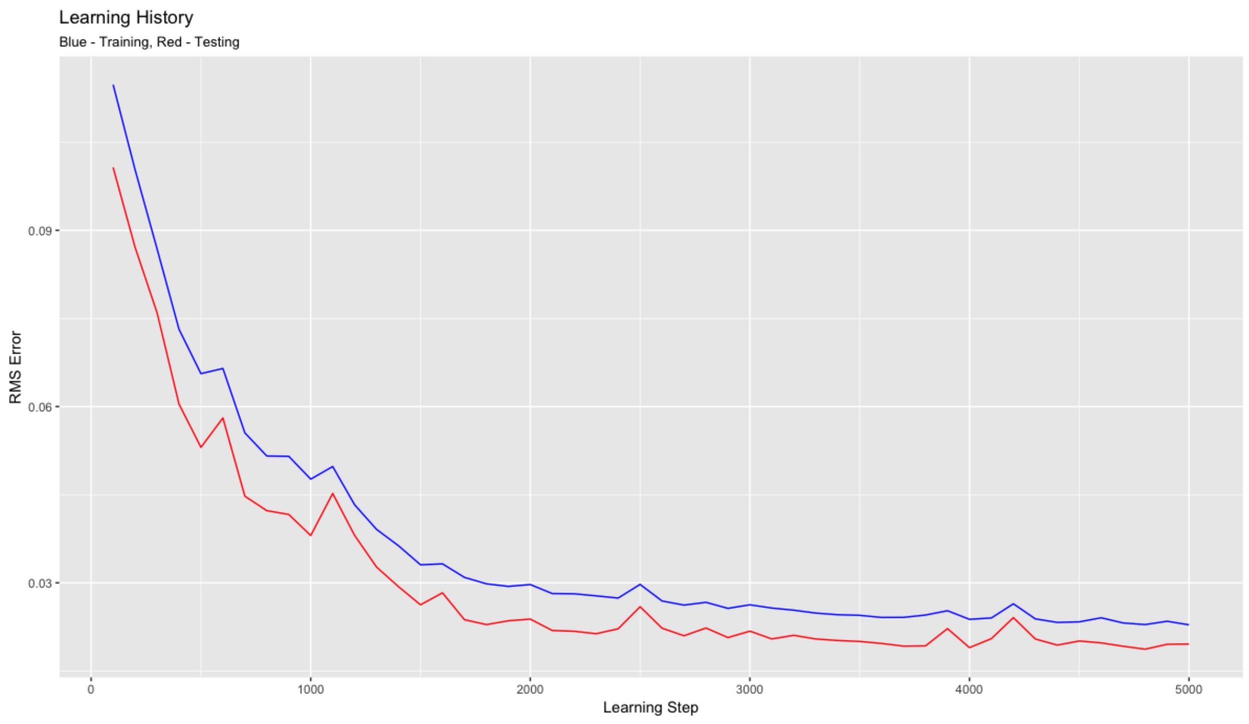
## Conclusions

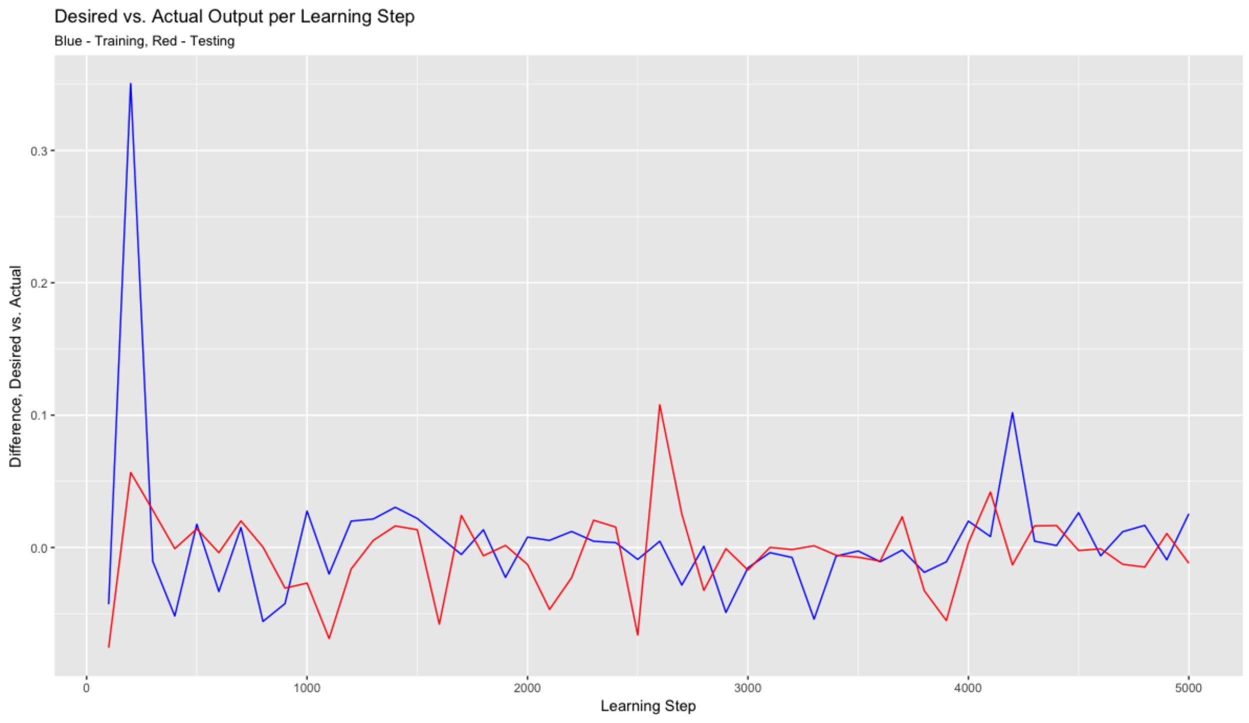
We had some interesting results on this problem. In regards to our learning history, our training converged to a RMSE of just about 0.02 at 5,000 Training Steps, while the testing converged to just about 0.01; this is curious that the error for our testing was lower than our training. Then when we compare the difference of our expected result and our recalled result for both training and testing, we get an excellent result; both testing and training are hovering around 0 and the spikes are minimal. Our memory does an excellent job of replicating the function, we can almost completely lay our recalled function onto the actual function, this is a great result!

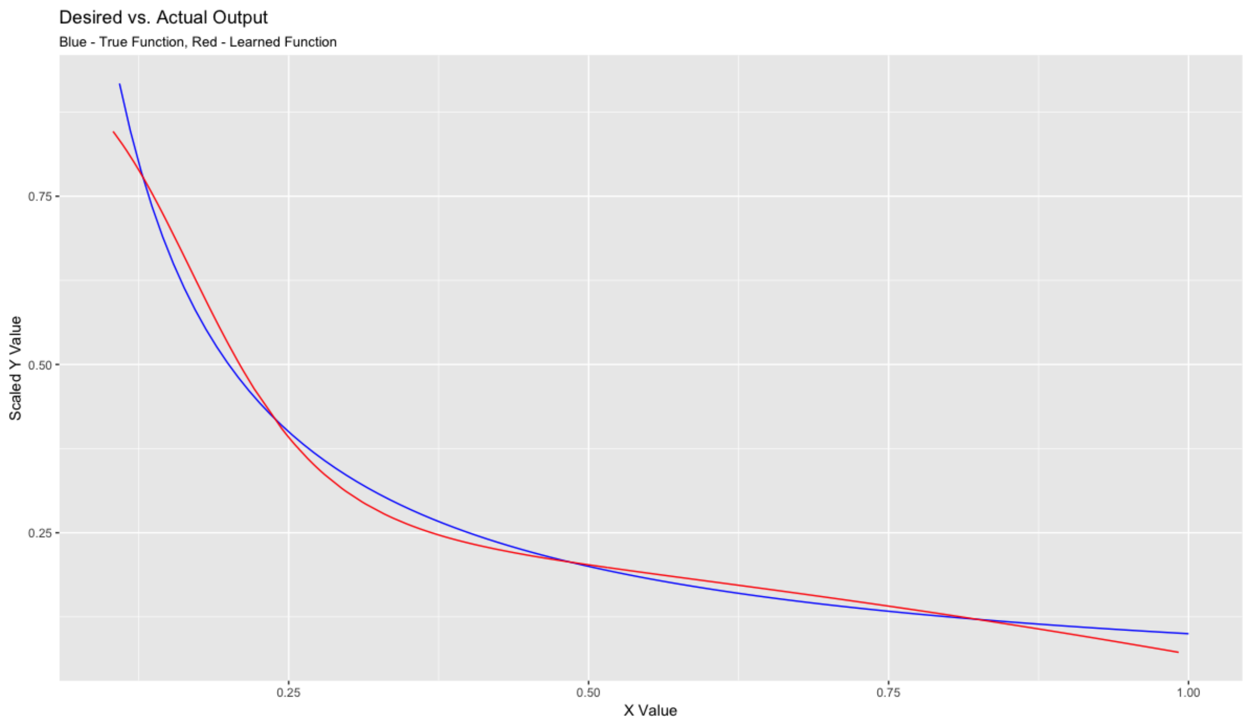
## Network 2 Details

- Network Parameters
  - Topology: (1 + 1 bias) - (20 + 1 bias) - 1
  - Transfer Function: Hyperbolic Tangent (slope = 1)
- Learning Parameters
  - Initial Weights: drawn randomly from Uniform[-0.1, 0.1]
  - Learning Rate: 0.05 and 0.0005
  - Momentum: 0.04
  - Epoch Size: 50
  - Stopping Criterion: 5,000 Learning Steps
  - Error Measure for Stopping Criterion: N/A
- Input / Output Data, Representation, Scaling:
  - Number of Training Samples: 200 (drawn randomly from Uniform[0.1, 1.0])
  - Number of Testing Samples: 100 (drawn randomly from Uniform[0.1, 1.0])
  - Scaling on Inputs: None
  - Scaling of Outputs: Dividing by 10
- Parameters and Error Measures of Performance Evaluation
  - Error of Function Fit: Root Mean Square Error

- Number of Learning Steps Performed: 5,000 Learning Steps
- Learning Rate at End: 0.0005
- Monitoring Frequency: Every 100 Learning Steps







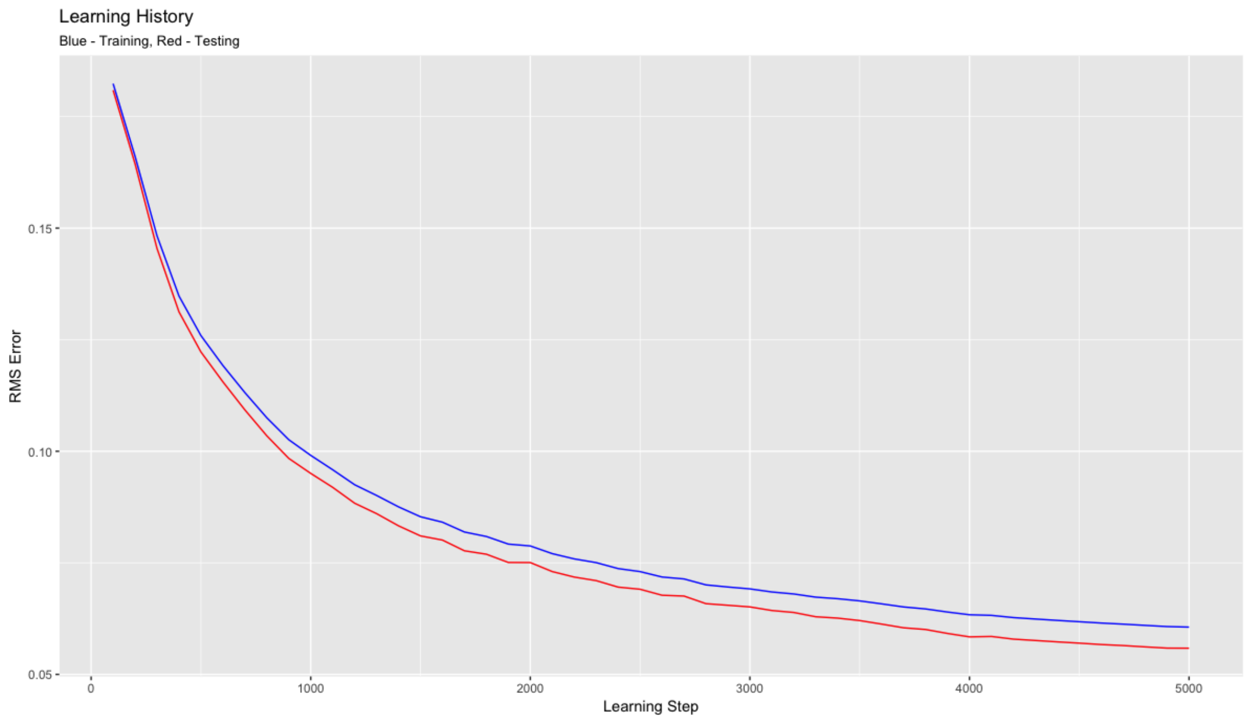
**Conclusions**

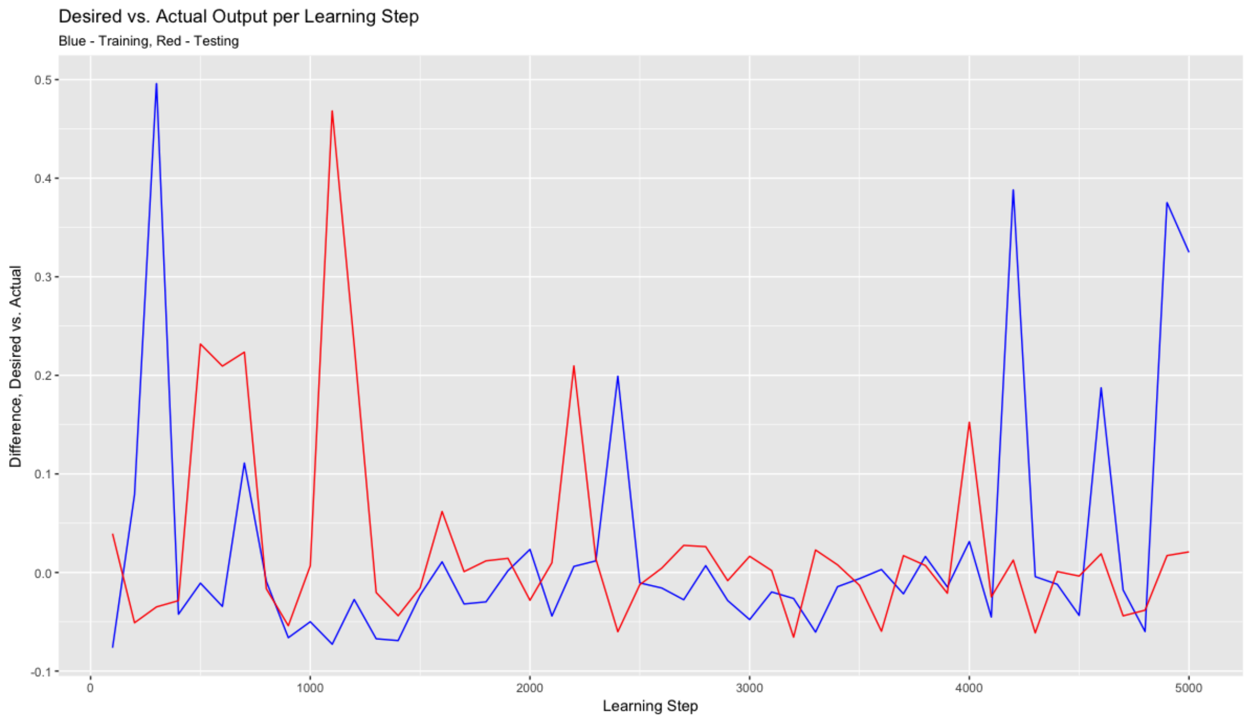
We again had some very positive, yet curious, results on this problem. In regards to our learning history, our training and testing converged to the same values as our last network, trainin to a RMSE of just about 0.02 at 5,000 Training Steps and testing to just about 0.01; this is curious that the error for our testing was lower than our training, same as with the last network. Then when we compare the difference of our expected result and our recalled result for both training and testing, we get another excellent result which we will not go into further as we already clearly described it in regards to our previous network. Again, as well, our memory does an excellent job of replicating the function, we can almost completely lay our recalled function onto the actual function.

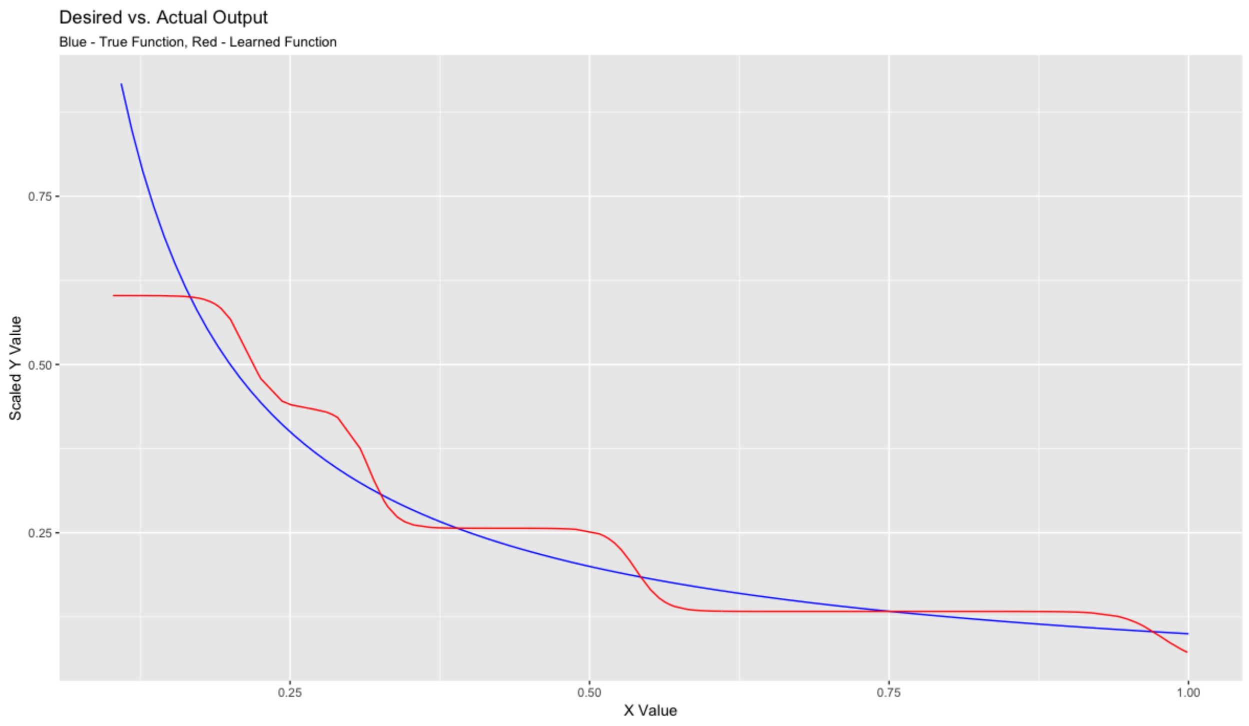
**Network 3 Details**

- Network Parameters
  - Topology: (1 + 1 bias) - (10 + 1 bias) - 1
  - Transfer Function: Hyperbolic Tangent (slope = 1)
- Learning Parameters
  - Initial Weights: drawn randomly from Uniform[-0.1, 0.1]
  - Learning Rate: 0.5 and 0.000005
  - Momentum: 0.04
  - Epoch Size: 100
  - Stopping Criterion: 5,000 Learning Steps
  - Error Measure for Stopping Criterion: N/A
- Input / Output Data, Representation, Scaling:
  - Number of Training Samples: 200 (drawn randomly from Uniform[0.1, 1.0])
  - Number of Testing Samples: 100 (drawn randomly from Uniform[0.1, 1.0])
  - Scaling on Inputs: None
  - Scaling of Outputs: Dividing by 10
- Parameters and Error Measures of Performance Evaluation

- Error of Function Fit: Root Mean Square Error
- Number of Learning Steps Performed: 5,000 Learning Steps
- Learning Rate at End: 0.000005
- Monitoring Frequency: Every 100 Learning Steps







### Conclusions

Here is where things got a little interesting in our network. In regards to our learning history, our training and testing converged to values between 0.05 and 0.075. We can see that the learning curve is much more smooth, we attribute this to the increae in batch size. Again, curiously our testing error is lower than our training. Then when we compare the difference of our expected result and our recalled result for both training and testing, we got very poor results. While they do hover around 0, there are many spike that shows a lot of volatility. This time, our memory does a very poor job of replicating the function, it doesn't seem to fit very well at all; this is without a adoubt our worst network.

## Overall Conclusions

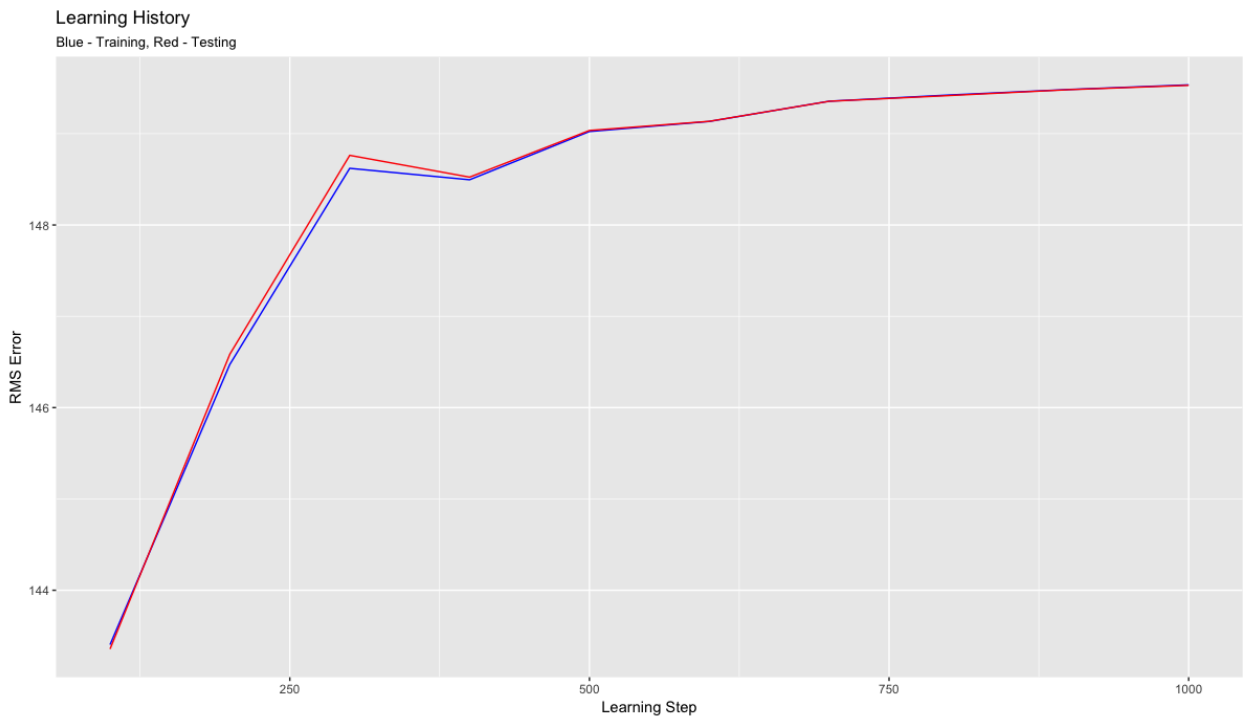
Overall, I think our best network was either network 1 or network 2; they were actually quite indistinguishable in many ways. However, network 3 does not compare as well to the other two and we would not consider it against the other two.

## Problem 4

### 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

  - Epoch Size: 10
  - Stopping Criterion: 1,000 Learning Steps
  - Error Measure for Stopping Criterion: N/A
- Input / Output Data, Representation, Scaling:
  - Number of Training Samples: 75 from Iris Training Data
  - Number of Testing Samples: 75 from Iris Testing Data
  - Scaling on Inputs: None
  - Scaling of Outputs: Nonw
- Parameters and Error Measures of Performance Evaluation
  - Error of Function Fit: Absolute summed difference
  - Number of Learning Steps Performed: 1,000 Learning Steps
  - Learning Rate at End: 0.05
  - Monitoring Frequency: Every 100 Learning Steps



## Conclusions

We got excellent results on both the training and testing data! Very quickly, we were able to attain a 100% hit rate of 150 with both the testing and training data at around 1,000 learning steps! 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!