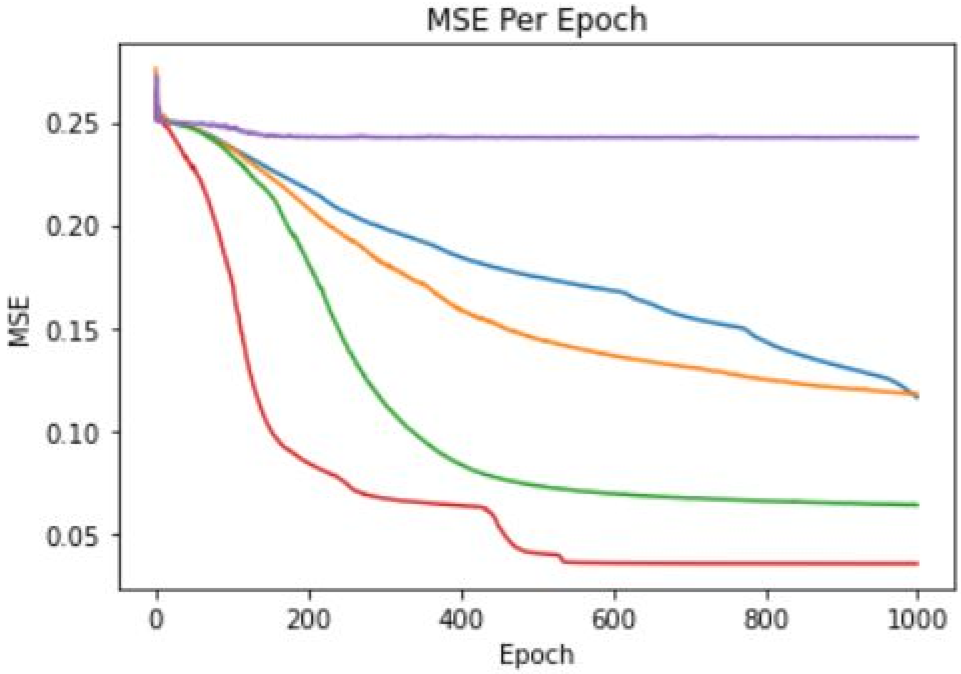
Shanni Lam
Math 189Z
30 April 2020

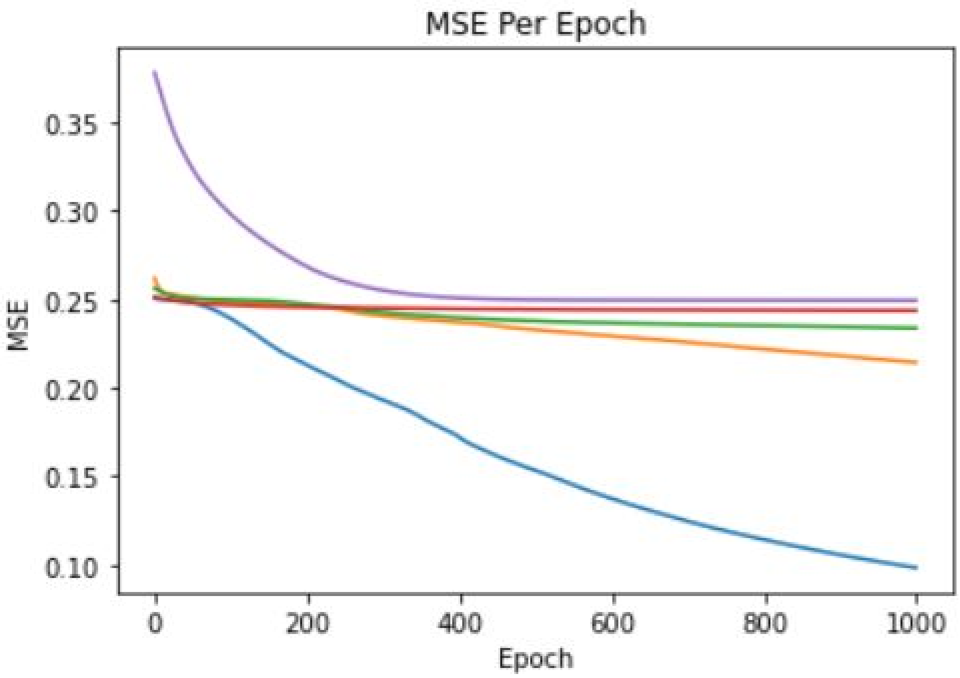
Homework 4

Feed Forward Tasks:

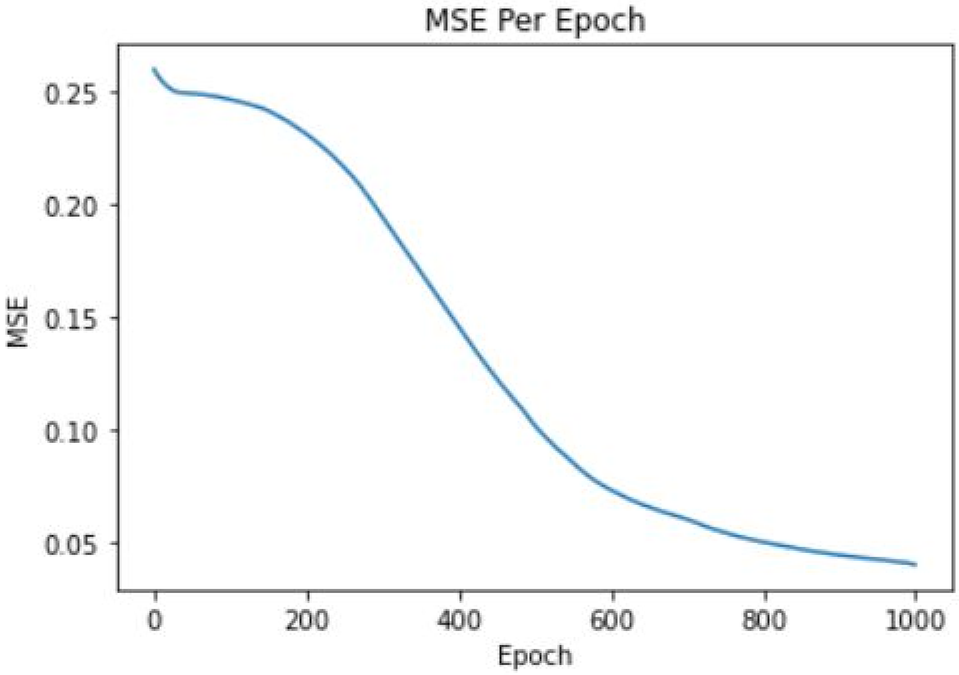
Graph of loss curve for 5 different learning rates:



Graph of loss curve for 5 different numbers of hidden nodes:

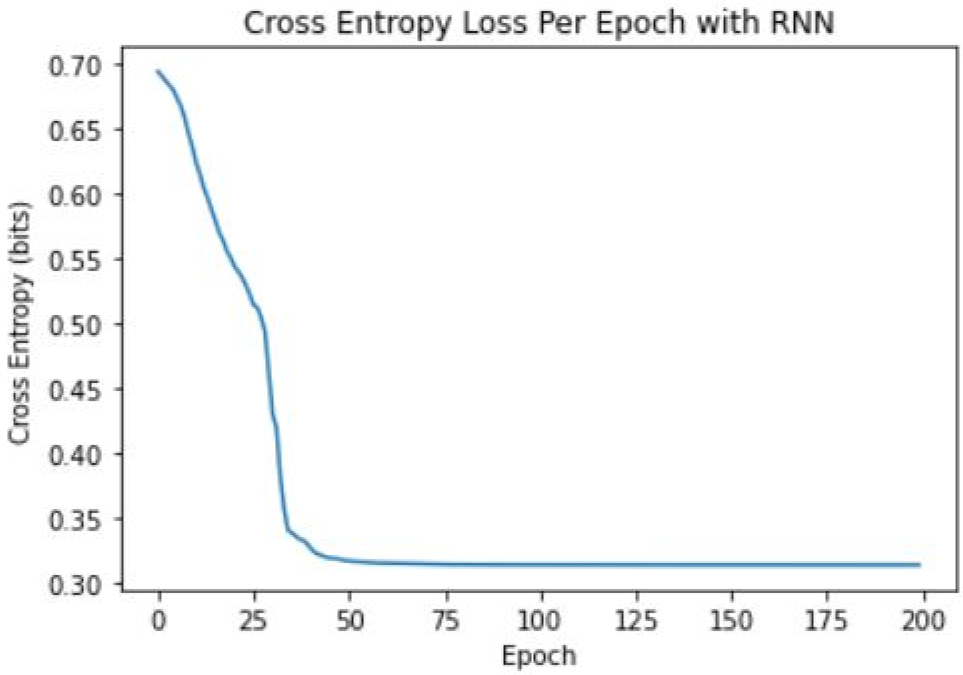


Graph of loss curve with an additional layer:

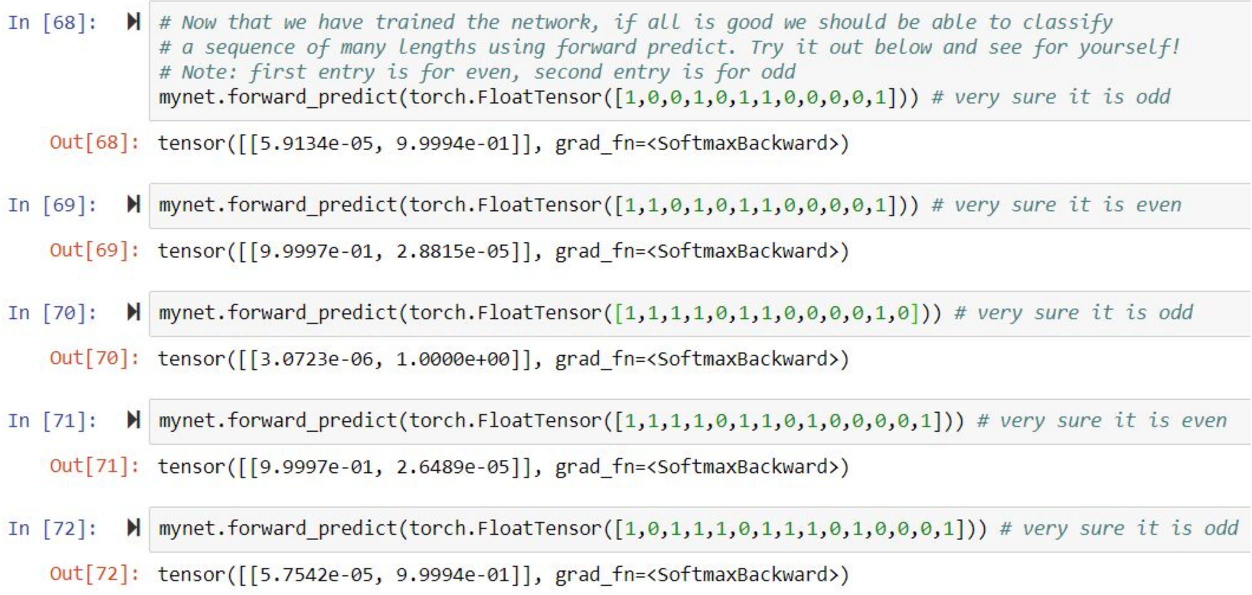


RNN Tasks:

The loss curve of your RNN:



A screenshot of your RNN classifying at least 5 sequences of different lengths:



```
In [68]:  # Now that we have trained the network, if all is good we should be able to classify
          # a sequence of many lengths using forward predict. Try it out below and see for yourself!
          # Note: first entry is for even, second entry is for odd
          mynet.forward_predict(torch.FloatTensor([1,0,0,1,0,1,1,0,0,0,0,1])) # very sure it is odd

Out[68]:  tensor([[5.9134e-05, 9.9994e-01]], grad_fn=<SoftmaxBackward>)

In [69]:  mynet.forward_predict(torch.FloatTensor([1,1,0,1,0,1,1,0,0,0,0,1])) # very sure it is even

Out[69]:  tensor([[9.9997e-01, 2.8815e-05]], grad_fn=<SoftmaxBackward>)

In [70]:  mynet.forward_predict(torch.FloatTensor([1,1,1,1,0,1,1,0,0,0,0,1,0])) # very sure it is odd

Out[70]:  tensor([[3.0723e-06, 1.0000e+00]], grad_fn=<SoftmaxBackward>)

In [71]:  mynet.forward_predict(torch.FloatTensor([1,1,1,1,0,1,1,0,1,0,0,0,0,1])) # very sure it is even

Out[71]:  tensor([[9.9997e-01, 2.6489e-05]], grad_fn=<SoftmaxBackward>)

In [72]:  mynet.forward_predict(torch.FloatTensor([1,0,1,1,1,0,1,1,1,0,1,0,0,0,1])) # very sure it is odd

Out[72]:  tensor([[5.7542e-05, 9.9994e-01]], grad_fn=<SoftmaxBackward>)
```