Feed Forward Tasks:
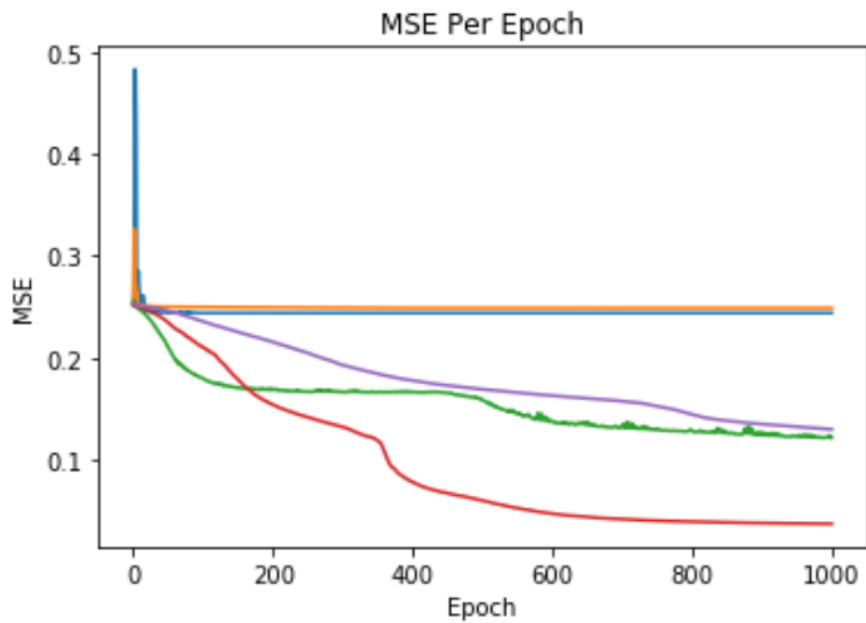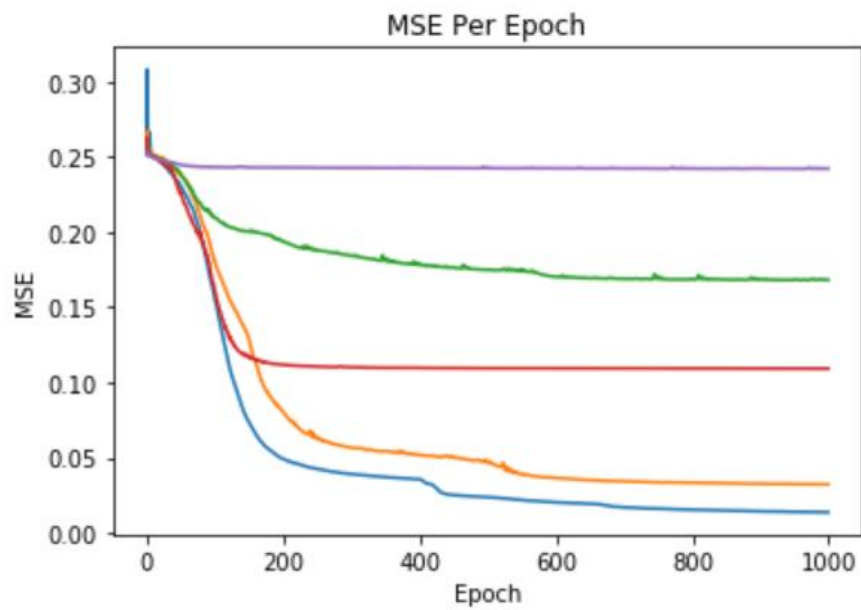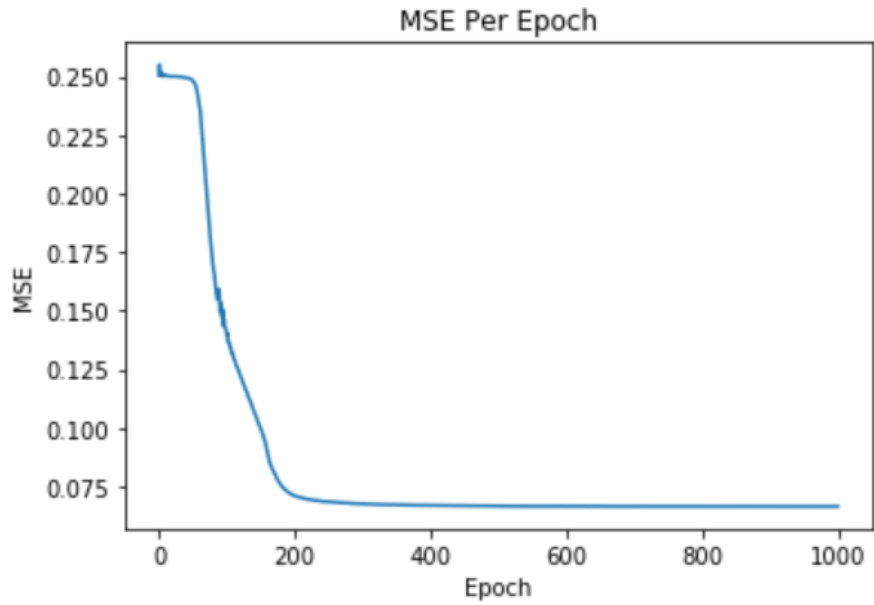
Graph of loss curve for 5 different learning rates:

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

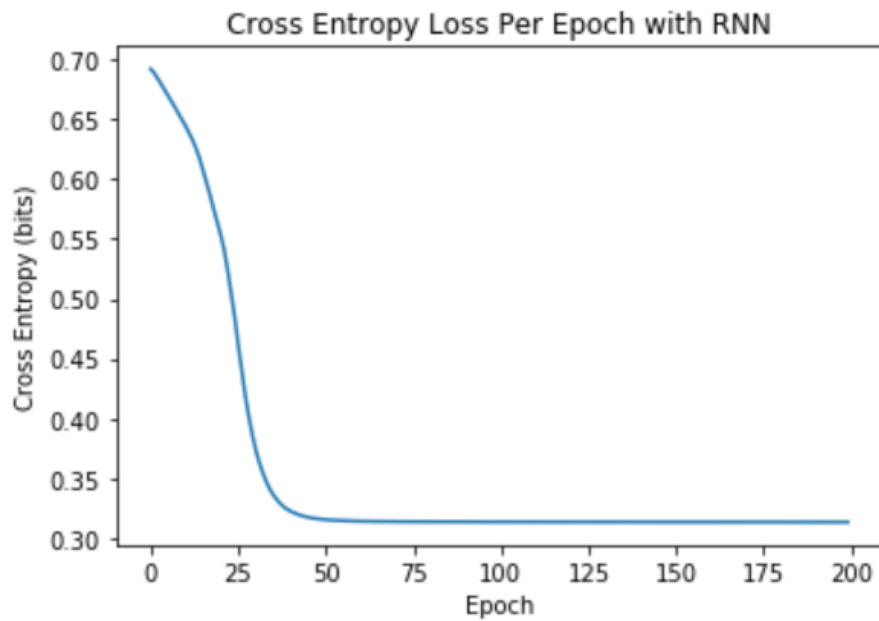Graph of loss curve with an additional layer:

MSE Per Epoch

(graph: MSE vs Epoch)

RNN Task:

The loss curve of your RNN:

Cross Entropy Loss Per Epoch with RNN

(graph: Cross Entropy (bits) vs Epoch)

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

```python
print(mynet.forward_predict(torch.FloatTensor([1,0,0,1,0,1,1,0,0,0,0,1])))

print(mynet.forward_predict(torch.FloatTensor([1,0,0,1,0,1,1,0,0,0,1])))

print(mynet.forward_predict(torch.FloatTensor([1,1])))

print(mynet.forward_predict(torch.FloatTensor([1,0,0,1,0,1,1,1])))

print(mynet.forward_predict(torch.FloatTensor([1,0,0,1,0,1,1,0,0,0,0,1,1,0,1,0,1,0,0])))
```

```
tensor([[2.1727e-05, 9.9998e-01]], grad_fn=<SoftmaxBackward>)
tensor([[2.1727e-05, 9.9998e-01]], grad_fn=<SoftmaxBackward>)
tensor([[9.9957e-01, 4.2921e-04]], grad_fn=<SoftmaxBackward>)
tensor([[1.2078e-05, 9.9999e-01]], grad_fn=<SoftmaxBackward>)
tensor([[9.9999e-01, 1.2155e-05]], grad_fn=<SoftmaxBackward>)
```