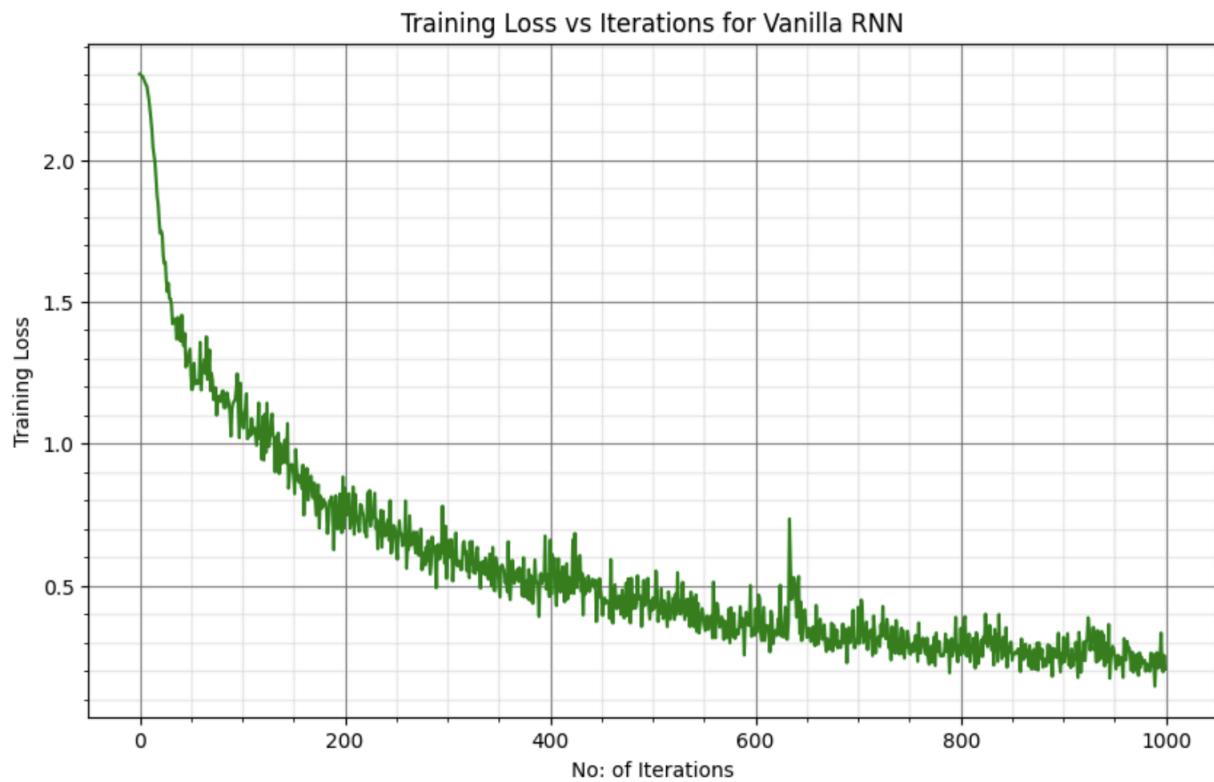


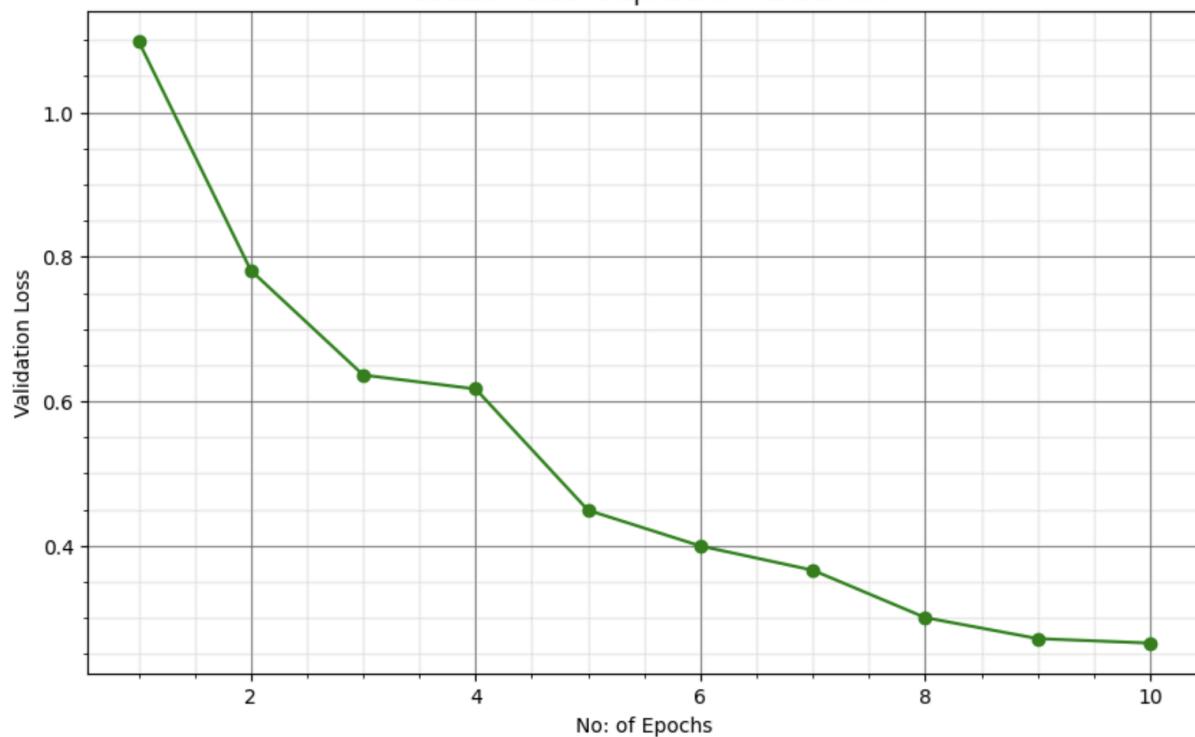
EE5179: Deep Learning for Imaging
Programming Assignment 4: Recurrent Neural Networks

Name : Athira KS
Roll No: EE23D034

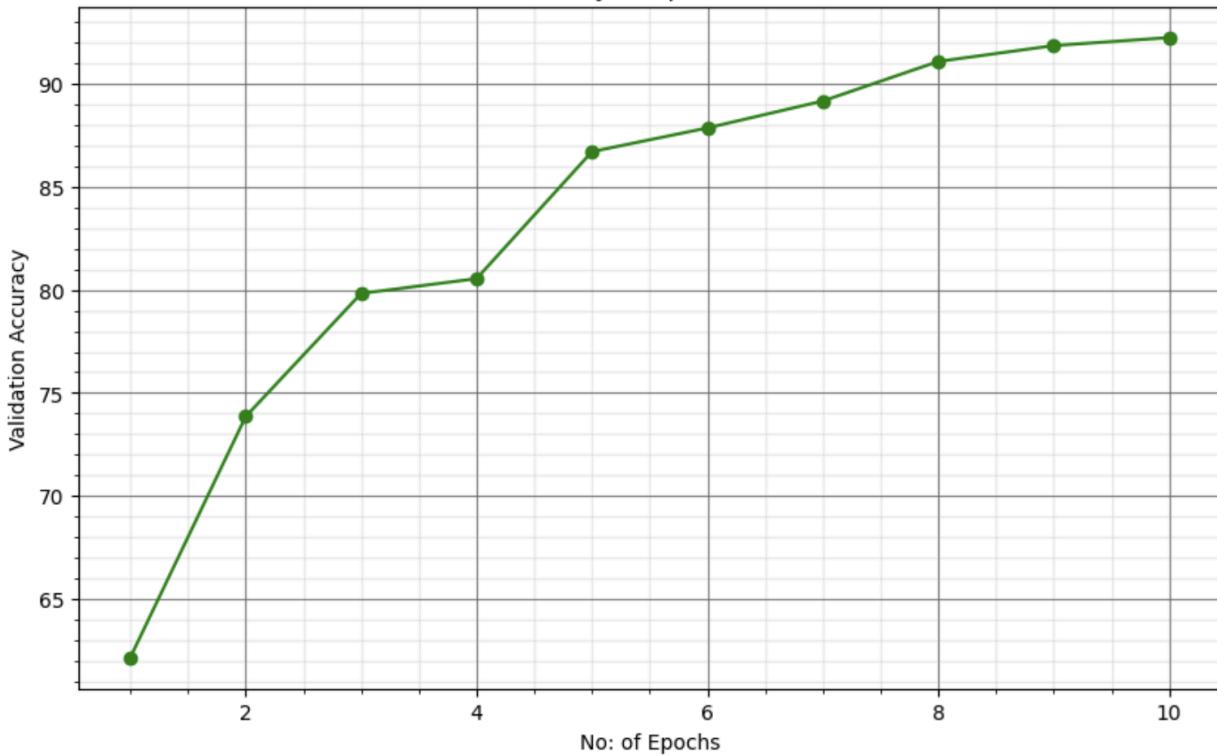
Network with vanilla RNN cells



Validation Loss vs Epochs for Vanilla RNN

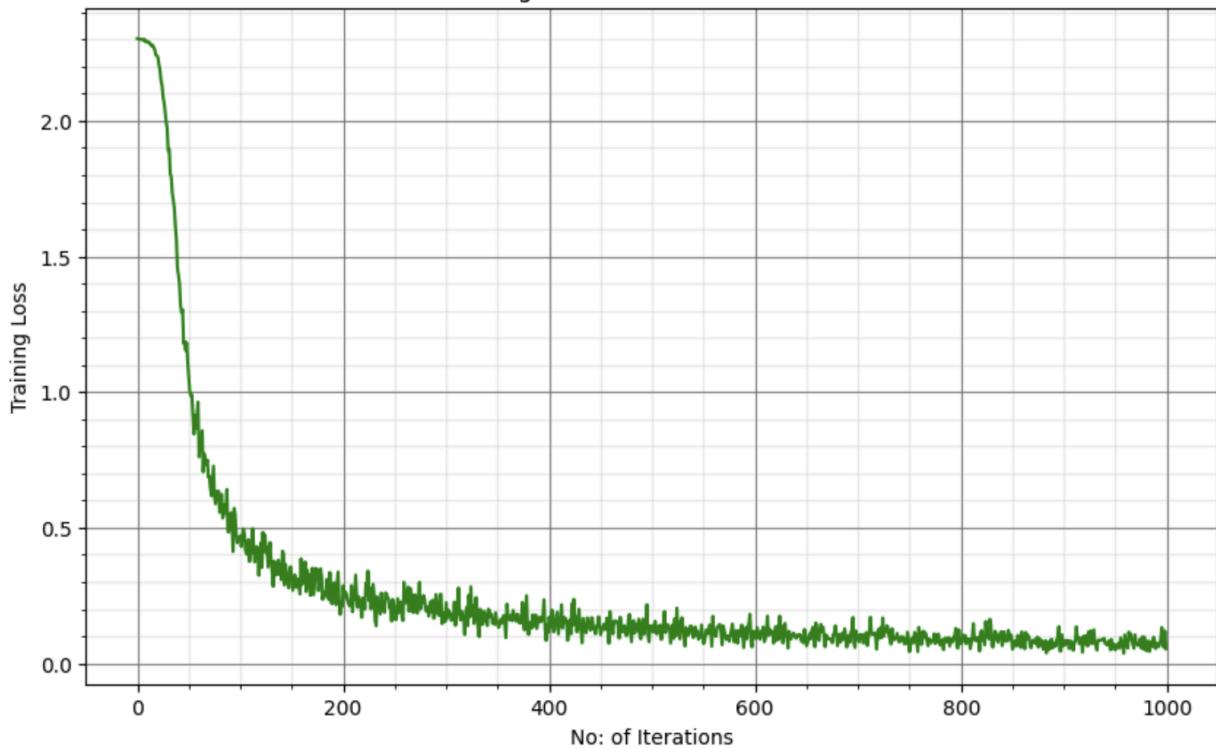


Validation Accuracy vs Epochs for Vanilla RNN

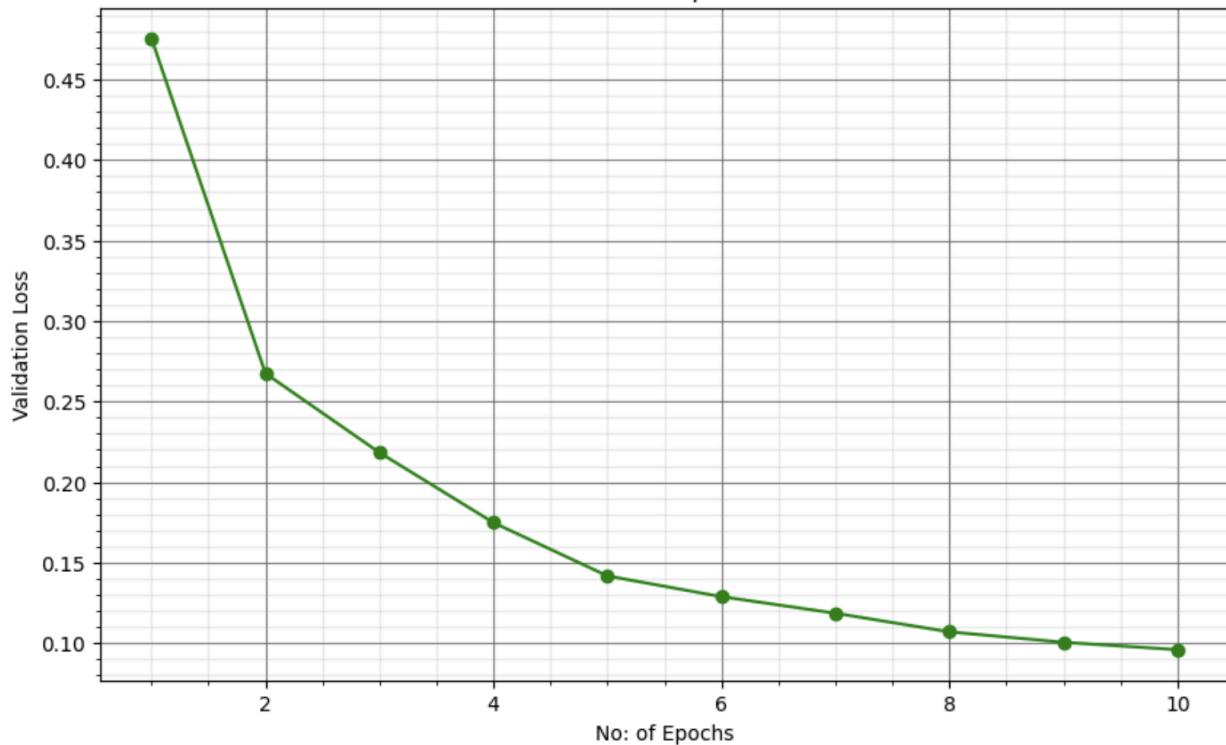


Network with LSTM

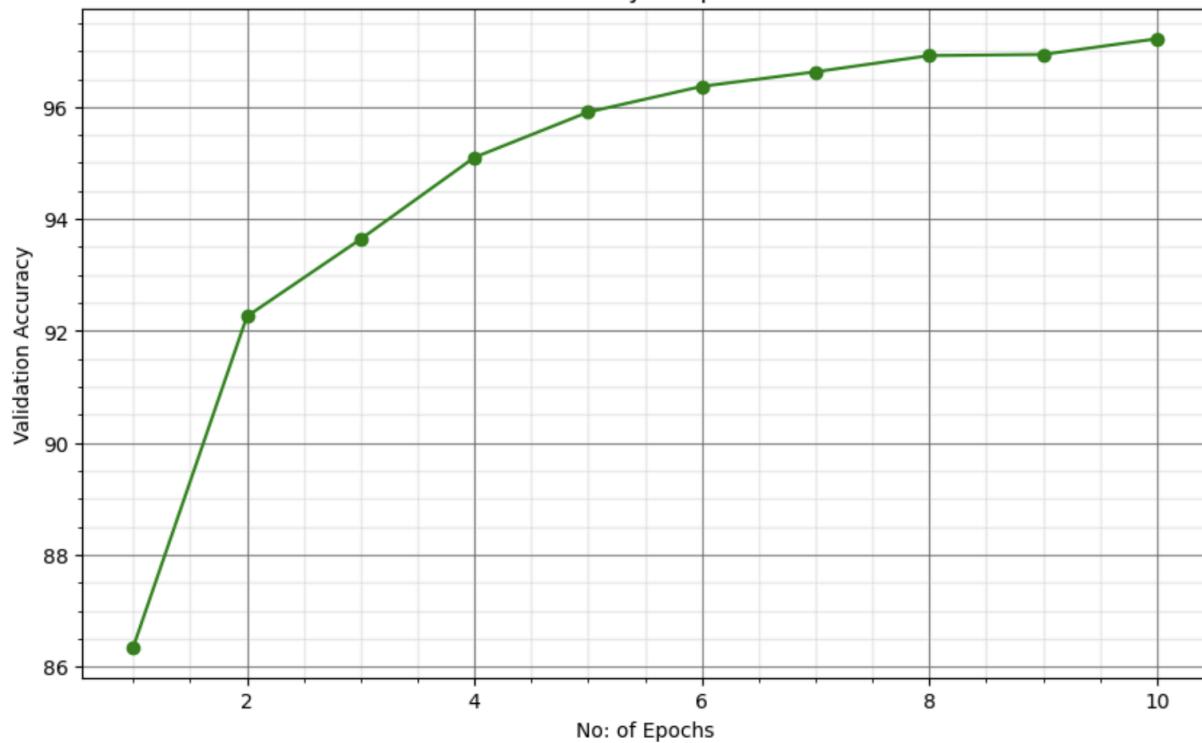
Training Loss vs Iterations for LSTM



Validation Loss vs Epochs for LSTM

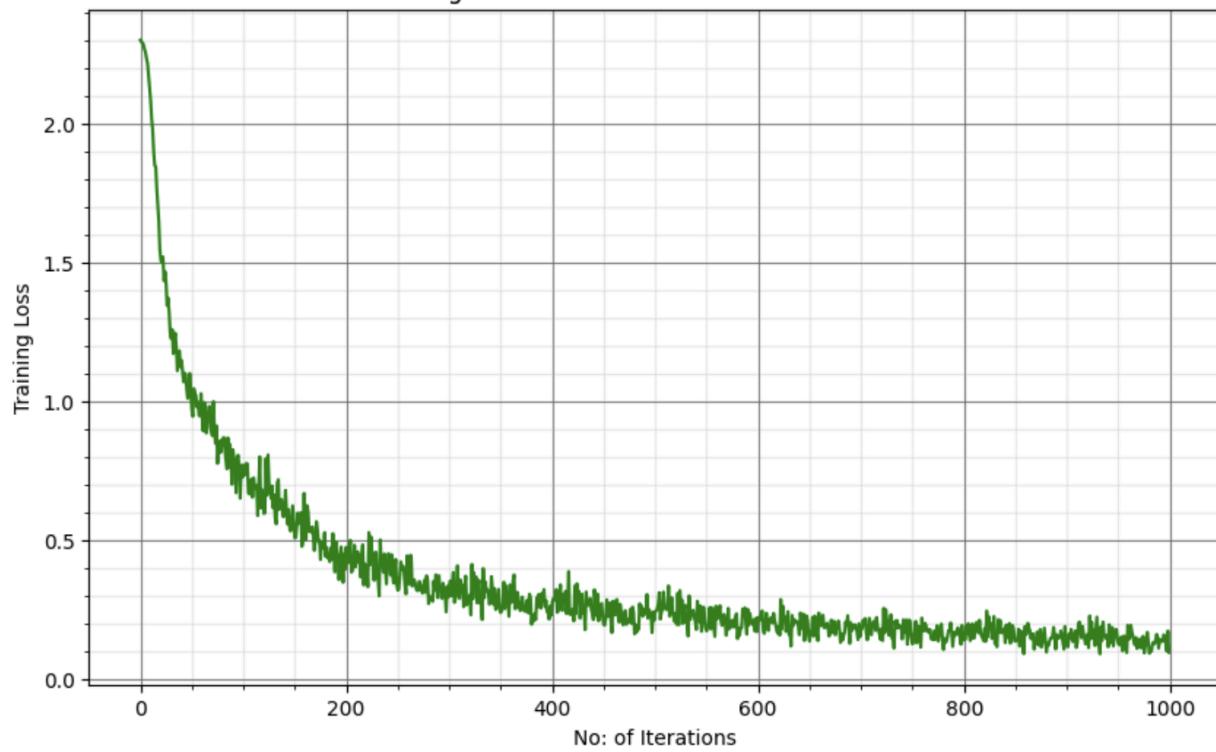


Validation Accuracy vs Epochs for LSTM

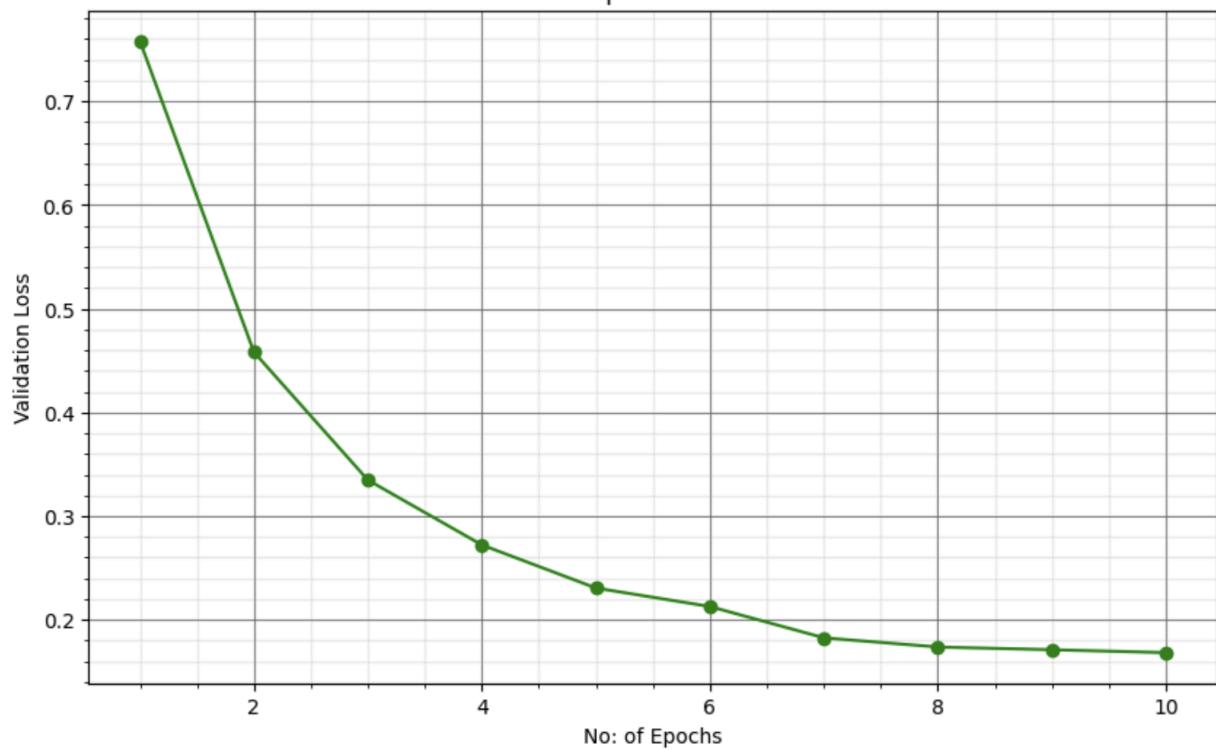


Bidirectional RNN

Training Loss vs Iterations for Bidirectional RNN



Validation Loss vs Epochs for Bidirectional RNN



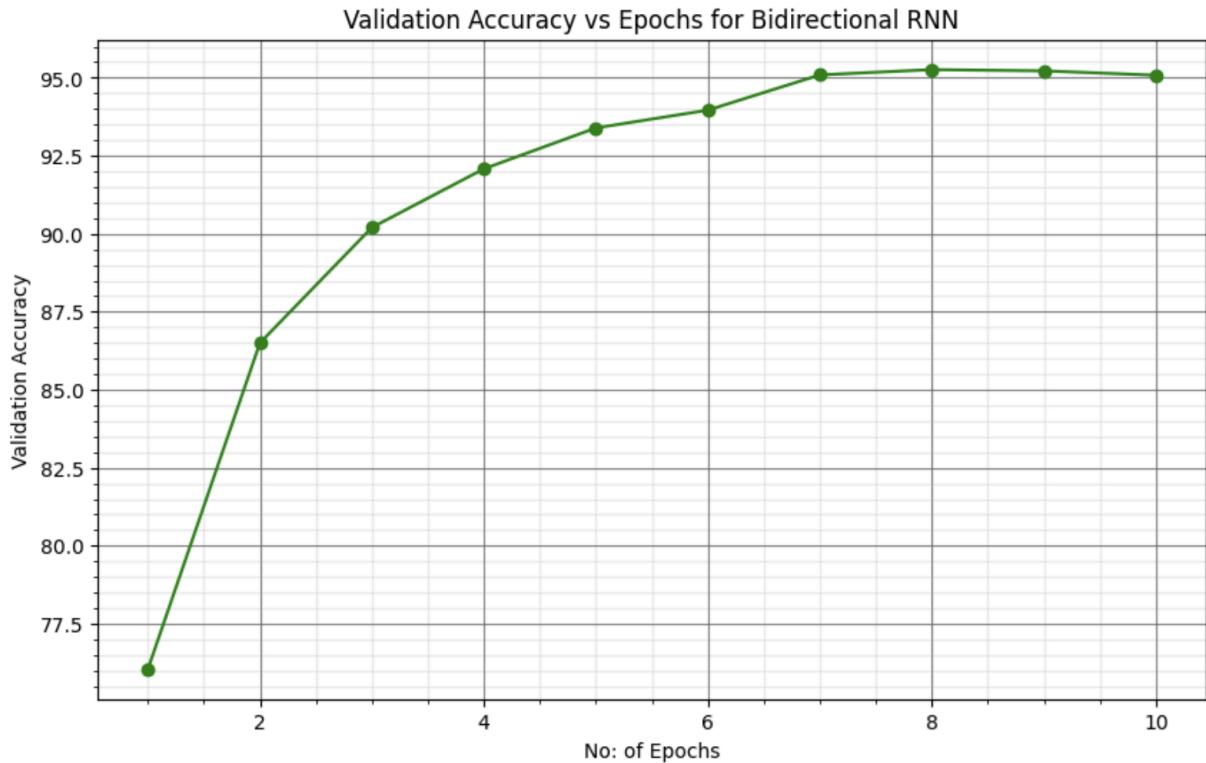


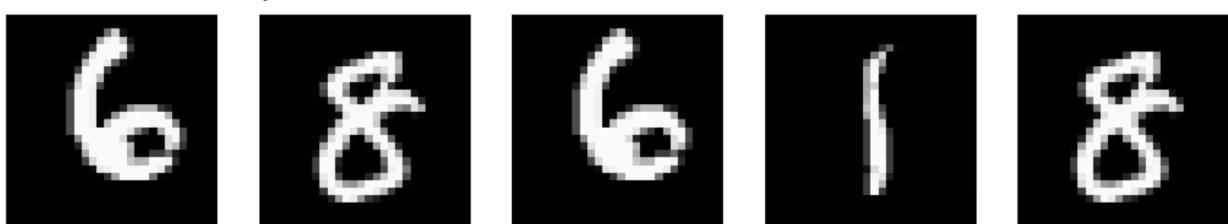
Image 0 - Probability Distribution: [1.3154271e-04 2.3861825e-04 3.1932109e-04 2.2325266e-06 1.3385200e-03
1.5034445e-04 9.9773681e-01 4.7907002e-07 7.2018171e-05 1.0122235e-05]

Image 4 - Probability Distribution: [3.58671969e-05 4.47018137e-05 1.46949715e-05 7.67895777e-04
3.29597469e-06 1.52475783e-04 1.58596667e-05 4.46069544e-06
9.98766541e-01 1.94129927e-04]

Image 0 - Probability Distribution: [1.3154271e-04 2.3861825e-04 3.1932109e-04 2.2325266e-06 1.3385200e-03
1.5034445e-04 9.9773681e-01 4.7907002e-07 7.2018171e-05 1.0122235e-05]

Image 3 - Probability Distribution: [8.0636019e-06 9.9851543e-01 6.7845409e-05 7.6648750e-05 2.5530902e-04
1.0270411e-04 4.1958294e-05 6.0913368e-04 1.5172156e-04 1.7120739e-04]

Image 4 - Probability Distribution: [3.58671969e-05 4.47018137e-05 1.46949715e-05 7.67895777e-04
3.29597469e-06 1.52475783e-04 1.58596667e-05 4.46069544e-06
9.98766541e-01 1.94129927e-04]



Predicted labels: [6. 8. 6. 1. 8.]

Actual labels: [6. 8. 6. 1. 8.]

```

<ipython-input-32-1d3b0dd5d261>:23: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default value), which uses the default pickle module implicit
best_model.load_state_dict(torch.load('/content/drive/MyDrive/PhD/SEM_2/DL/LAB4/lstm_model.pth'))
Raw output for image 0: tensor([-1.4608, -1.4952, -3.3222,  2.7312, -3.1763,  1.0982, -4.5947,  2.0181,
        1.4472,  4.4635]))
Raw output for image 1: tensor([-1.4047, -1.5026, -3.3442,  2.7520, -3.1818,  1.1274, -4.5827,  1.9630,
        1.4722,  4.5063)))
Raw output for image 2: tensor([-1.4531, -1.5535, -3.3077,  2.7489, -3.1962,  1.1324, -4.6181,  2.0524,
        1.4196,  4.4402)))
Raw output for image 3: tensor([-1.4281, -1.5037, -3.3362,  2.7332, -3.1817,  1.1120, -4.5751,  1.9683,
        1.4917,  4.4965)))
Raw output for image 4: tensor([-1.2810, -1.6746, -3.3869,  2.7809, -3.1986,  1.1849, -4.4916,  1.7060,
        1.6954,  4.7254)))
Raw output for image 5: tensor([-1.4803, -1.5964, -3.3515,  2.7272, -3.1802,  1.1530, -4.5930,  1.9641,
        1.5146,  4.5319)))
Raw output for image 6: tensor([-1.6854, -1.8815, -3.3646,  2.6307, -3.0700,  1.0830, -4.6287,  1.9507,
        1.6068,  4.6933)))
Raw output for image 7: tensor([-1.4374, -1.4494, -3.3212,  2.7121, -3.1712,  1.0754, -4.5823,  2.0068,
        1.4582,  4.4587)))
Raw output for image 8: tensor([-1.0029, -2.1182, -2.9943,  3.3793, -3.5736,  1.5761, -4.9384,  2.5167,
        0.5168,  4.0645)))
Raw output for image 9: tensor([-1.3719, -1.5622, -3.3517,  2.7707, -3.2106,  1.1313, -4.5512,  1.8979,
        1.5679,  4.5392)))

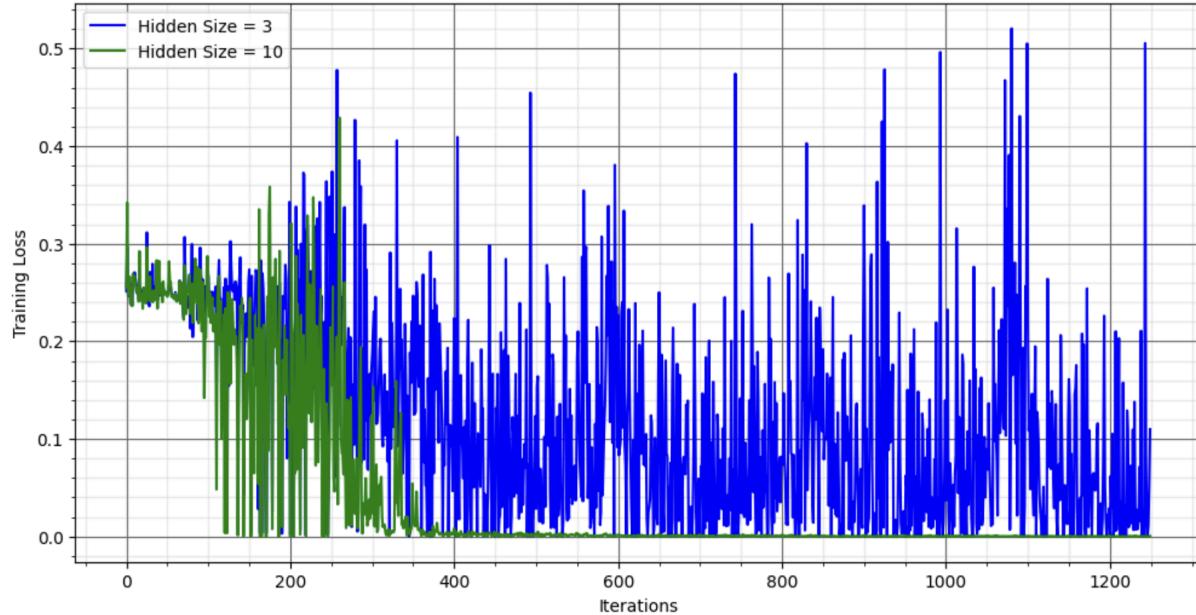
```

Pred: 9 Pred: 9

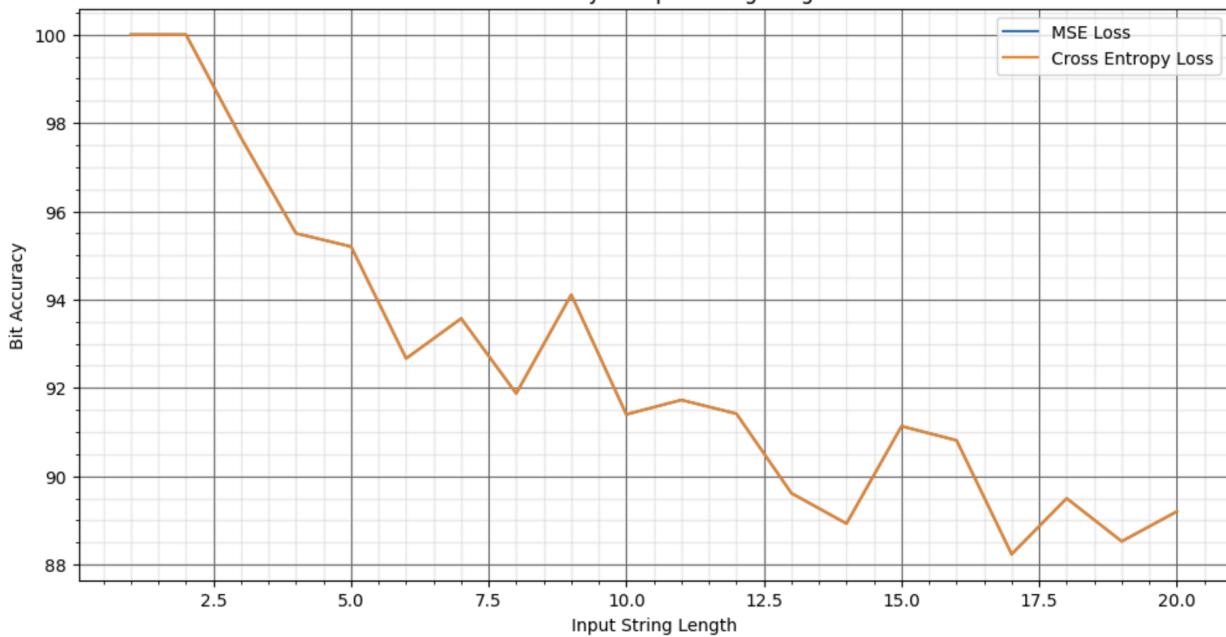
0 1 2 3 4 5 6 7 8 9

Addition of two numbers

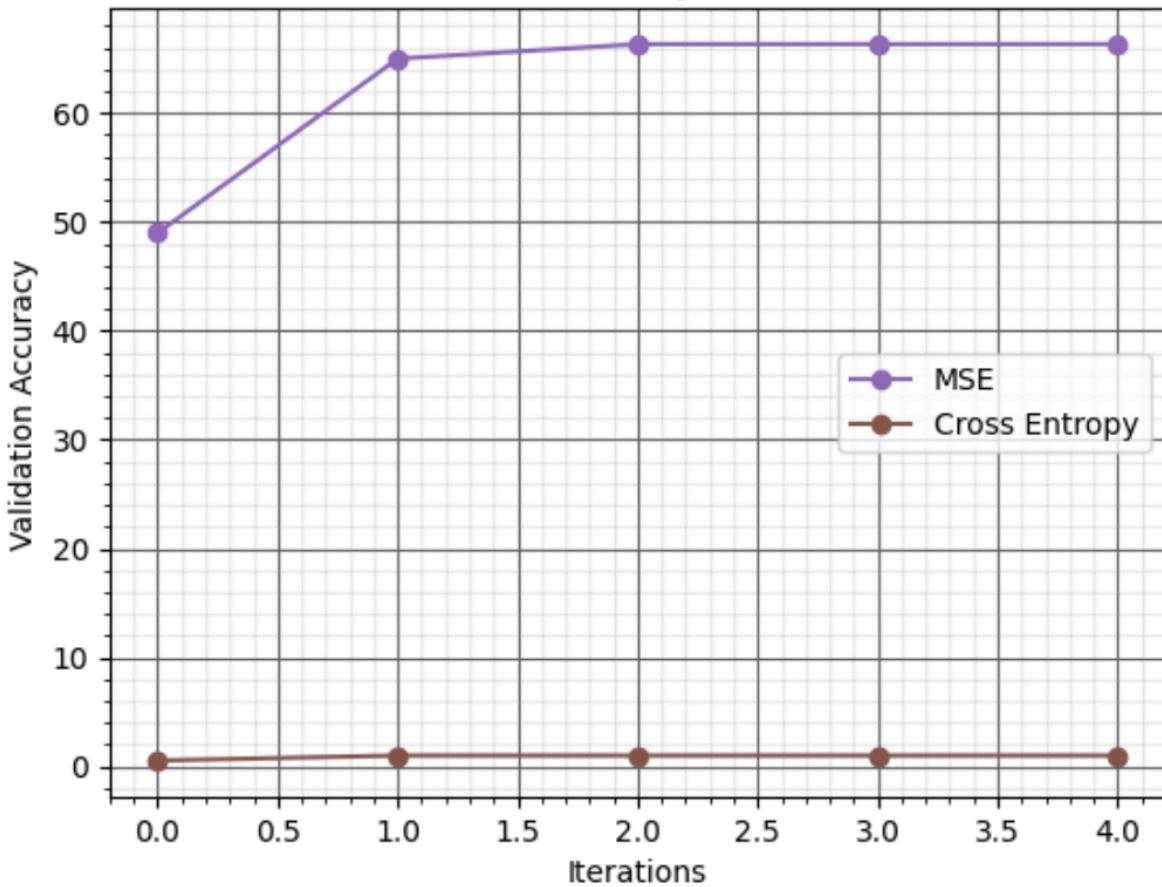
Training Loss vs Iterations



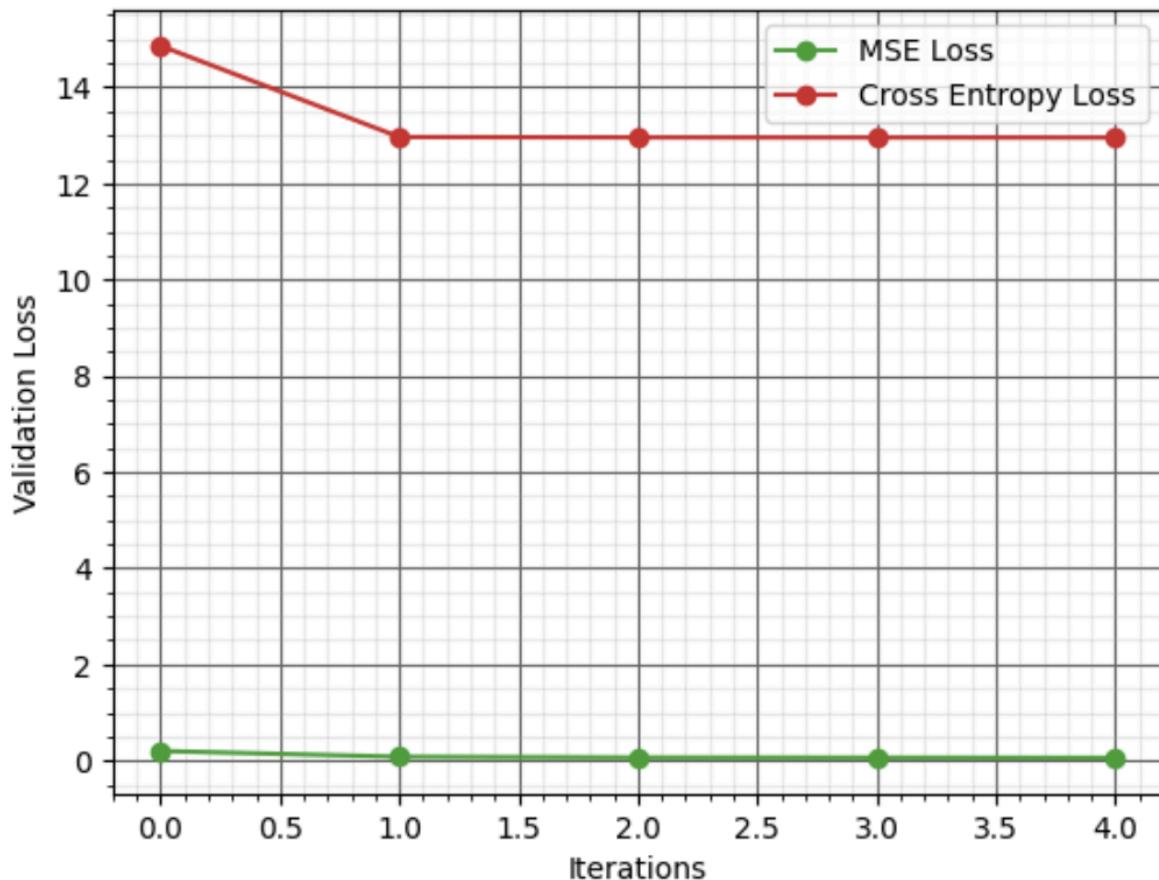
Bit Accuracy vs Input String Length



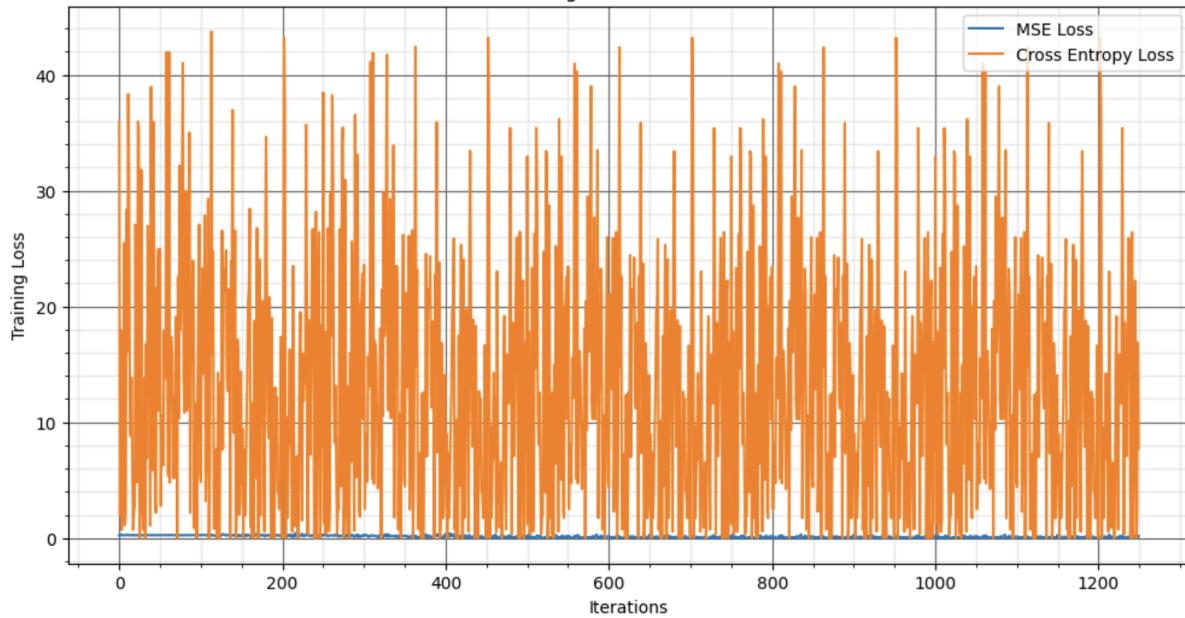
Validation Accuracy vs Iterations

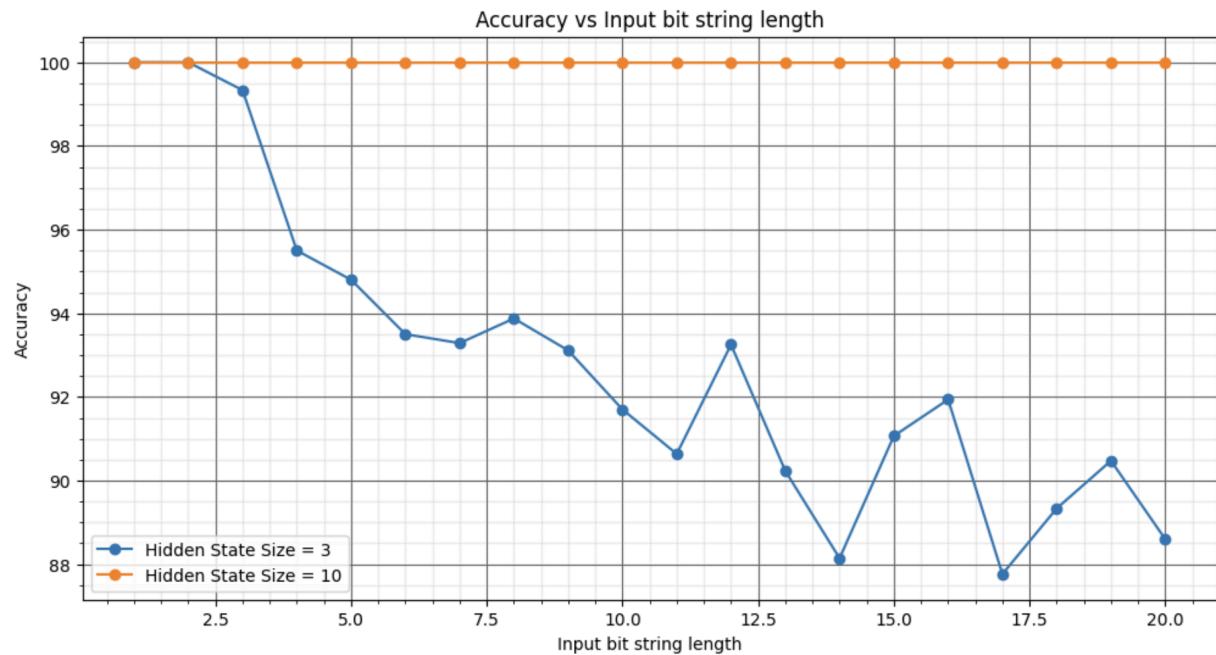


Validation Loss vs Iterations

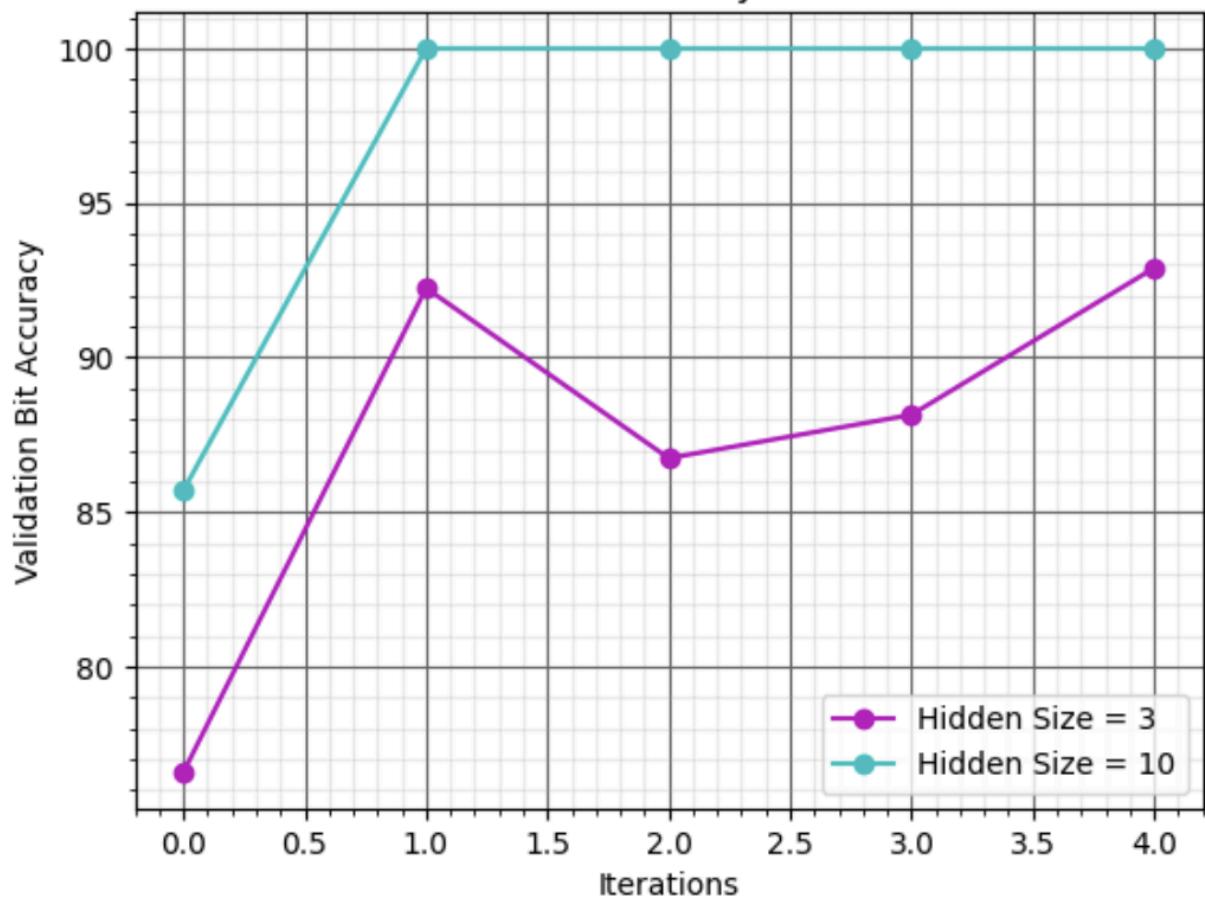


Training Loss vs Iterations

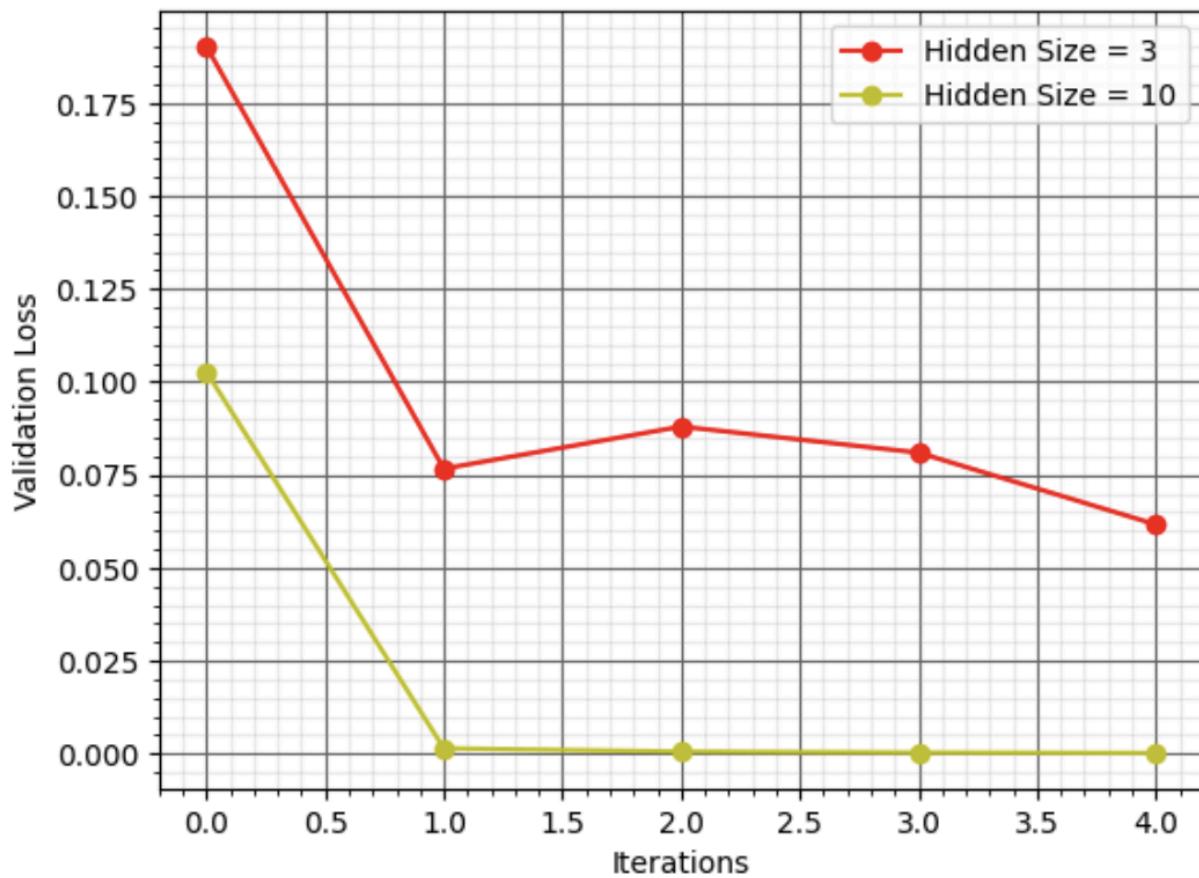




Validation Bit Accuracy vs Iterations



Validation Loss vs Iterations



Fixed Length 3,5,10

