ECE421: Introduction to Machine Learning — Fall 2024

Assignment 4 – RNN, MDP, and RL

Due Date: Wednesday, Dec 4, 11:59 PM

General Notes

1. Programming assignments can be done in groups of up to 2 students. Students can be in different sections.
2. Only one submission from a group member is required.
3. Group members will receive the same grade.
4. Please post assignment-related questions on Piazza.

Group Members

Name (and Name on Quercus)	UTORid
Edwin Chacko	chackoed
Bala Kannan Murali	muralib1

1 Part 1: Recurrent Neural Network (RNN)

1.1 Dataset

1.2 Implementing Single-layer Elman RNN and multi-layer LSTM RNN

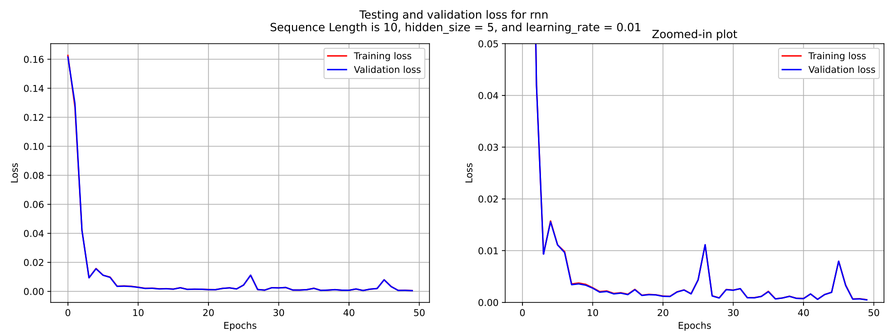
1.3 Implementing the Train Loop

1.4 Hyperparameter Tuning

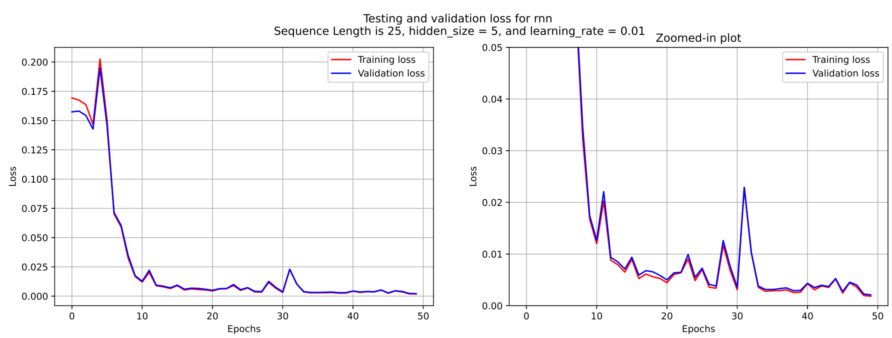
1.4.1 Question 1 (5 points): Report the Result of your Hyperparameter Tuning

In your report, include the loss figures of the best RNN and LST model that you could train. If you use the Google Colab Notebook to generate these figures, note that they are automatically saved under the `figures` folder of the assignment directory in your Google Drive. The figures are saved in eps format, which is vector file format, suitable for LaTeX. Use your favorite eps viewer/converter to view/convert these figures, if needed.

Answer.

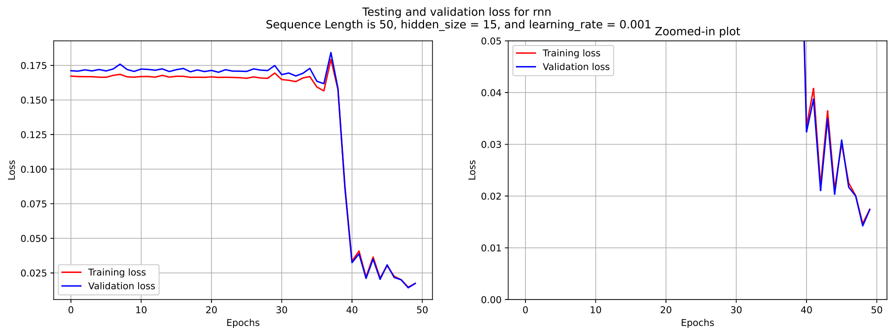


(a) RNN for sequence_length=10

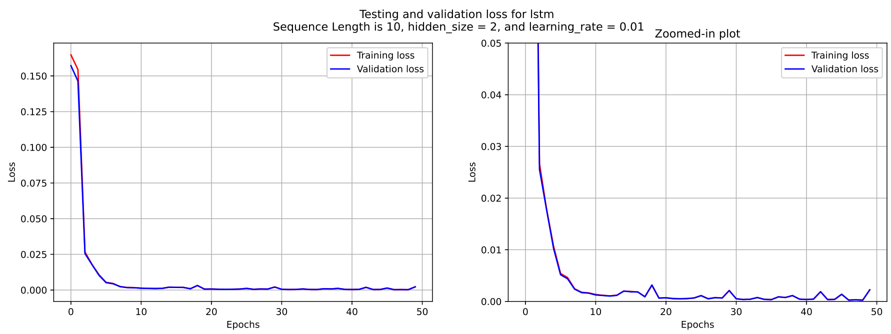


(b) RNN for sequence_length=25

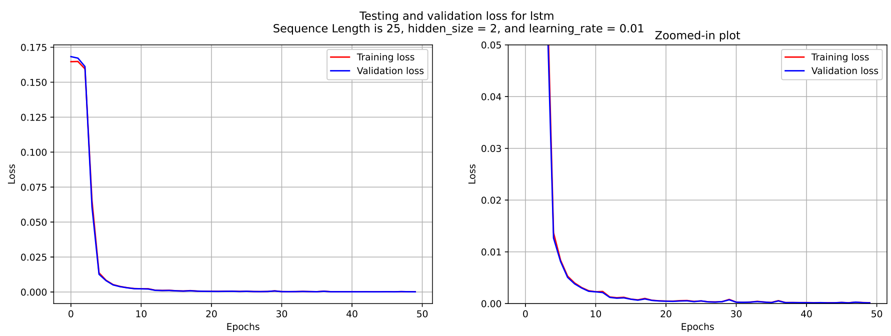
Figure 1: Loss figures of the best RNN and LST models.



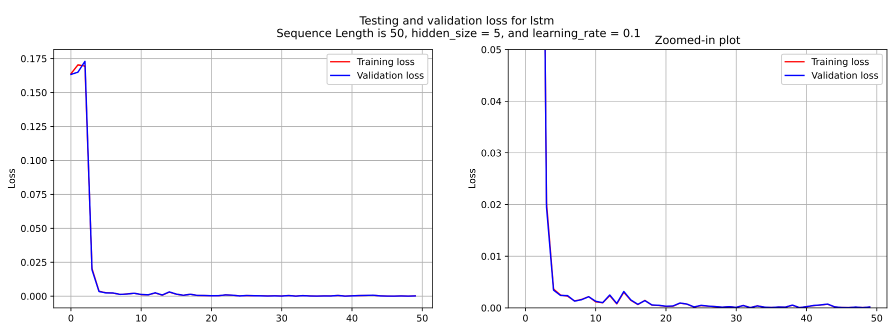
(c) RNN for sequence_length=50



(d) LSTM for sequence_length=10



(e) LSTM for sequence_length=25



(f) LSTM for sequence_length=50

Figure 1: Loss figures of the best RNN and LST model (cont.)

1.4.2 Question 2 (2 points): Why did Vanilla RNN fail?

Explain why the vanilla RNN fails to learn this simple task for the dataset with sequence length of 50? [**HINT:** You may find this article helpful.]

Answer. Long sequence jobs, particularly those with a sequence length of 50, are difficult for the vanilla RNN to handle because of the vanishing and expanding gradient issues.These problems occur during backpropagation through time. The error gradients in a standard RNN propagate back across the sequence's time steps. The vanishing gradient problem is caused by the tiny values (from the derivatives of the activation function) being multiplied repeatedly when the gradients are sent back through each time step. As a result, the gradients diminish rapidly, which prevents the network from learning long-term relationships. The exploding gradient problem, on the other hand, can arise during backpropagation if the weights are big because the gradients may expand out of control. As a result, learning is hampered and weight updates become erratic. In contrast, LSTMs are designed to handle long sequences by using a more sophisticated architecture. LSTMs include gates like the forget, input, and output gates, which control the flow of information into and out of the cell state. The cell state allows the network to retain important information across many time steps without suffering from vanishing or exploding gradients. The presence of the forget gate helps prevent the gradients from vanishing by allowing the network to selectively forget or retain information as needed, and the additive nature of the cell state gradient ensures better control over the gradient flow, preventing it from becoming too small or too large.

1.4.3 Question 3 (5 points): Code Uploading

No written part.

2 Part 2: Markov Decision Process (MDP) and Reinforcement Learning (RL)

No written part.

3 Turning It In

You need to submit your version of the following files:

- `models.py` and `train.py`
- The modified Google Colab notebook named as `PA4.ipynb`
- `valueIterationAgents.py`, `qlearningAgents.py`, and `analysis.py`
- `PA4_qa.pdf` that answer questions related to the implementation in part 1.
- The cover file with your name and student ID filled. If you use the LaTeX template, you don't need to include the cover file separately.

Please pack them into a single folder, compress into a .zip file and name it as PA4.zip.