Problem 1)

Inspired by the course example, train and validate rnn.RNN, rnn.LSTM and rnn.GRU for learning the above sequence. Use sequence lengths of 10, 20, and 30 for your training. Feel free to adjust other network parameters. Report and compare training loss, validation accuracy, execution time for training, and computational and mode size complexities across the three models over various lengths of sequence.

RNN sequence of 10:

A screen shot of a computer

AI-generated content may be incorrect.

RNN sequence of 20:

A screen shot of a computer

AI-generated content may be incorrect.

RNN sequence of 30:

A screen shot of a computer

AI-generated content may be incorrect.

The loss value and training accuracy were the best for a RNN sequence of 20.

LTSM sequence of 10:

A screen shot of a computer

AI-generated content may be incorrect.

LTSM sequence of 20:

A screenshot of a computer

AI-generated content may be incorrect.

LTSM sequence of 30:

A screenshot of a computer

AI-generated content may be incorrect.

The sequence of 20 had the best accuracy, the validation loss was best for the sequence of 30.

GRU sequence of 10:

A screen shot of a computer

AI-generated content may be incorrect.

GRU sequence of 20:

A screen shot of a computer

AI-generated content may be incorrect.

GRU sequence of 30:

A screen shot of a computer

AI-generated content may be incorrect.

For Problem 1, it seemed like the training length increased as the sequence length increased. And for all three models the best accuracy was seen for a sequence length of 20.

Problem 2:

Build the model for.LSTM and rnn.GRU for the tiny Shakespeare dataset, the data loader code is already provided.

1. Train the models for the sequence of 20 and 30, report and compare training loss, validation accuracy, execution time for training, and computational and mode size complexities across the two models.
2. Adjust the hyperparameters (fully connected network, number of hidden layers, and the number of hidden states) and compare your results (training and validation loss, computation complexity, model size, training and inference time, and the output sequence). Analyze their influence on accuracy, running time, and computational perplexity.
3. What if we increase the sequence length to 50. Perform the training and report the accuracy and model complexity results.

A screenshot of a computer program

AI-generated content may be incorrect.LSTM sequence of 20:

A computer screen shot of a program

AI-generated content may be incorrect.LSTM sequence of 30:

A screenshot of a computer program

AI-generated content may be incorrect.LSTM sequence of 50:

The best accuracy and training loss was seen on a sequence length of 50, however it took the longest to train.

A computer screen shot of a program

AI-generated content may be incorrect.A computer screen shot of a program

AI-generated content may be incorrect.GRU sequence of 20: GRU sequence of 30:

GRU sequence of 50:

A screenshot of a computer program

AI-generated content may be incorrect.

The best accuracy and training loss was seen on a sequence length of 50, however it took the longest to train. It appears that for both models this time the best sequence length was 50, the only issue is the long run time in comparison to the other model length.

GitHub Link

P1: <https://github.com/Eskdagoat/4106/blob/main/Homework3P1.ipynb>

P2: <https://github.com/Eskdagoat/4106/blob/main/Homework3P2.ipynb>