National University of Singapore
College of Design & Engineering – ECE

EE4400 Data Engineering and Deep Learning
Tutorial 3 - LSTM

Q1. How does an LSTM solve the vanishing or exploding gradient problem?

Q2. What mechanisms does an LSTM network use to learn the appropriate long term and short term states?

Q3. Comparing Simple RNN, RNN with Batch Normalization and LSTM for Univariate Time Series Forecasting

Consider the given univariate sequence:
[10, 20, 30, 40, 50, 60, 70, 80, 90]

a) Convert the time series data given above into the sequence structure that can be used to train the simple RNN and LSTM models. The sequence structure is as follows:

```
[10 20 30] [40 50]
[20 30 40] [50 60]
[30 40 50] [60 70]
[40 50 60] [70 80]
[50 60 70] [80 90]
```

where the input times series sequence length seq_len = 3, and the number of future time steps to predict n_steps = 2. (This is similar to Tutorial 2 Q6.)

b) Create a simple RNN model with 1 hidden layer containing 50 units and 1 output layer with number of units given by the number of future time steps to predict, i.e. n_steps. Use the Adam optimizer and Mean Square Error as the loss function. (This is similar to Tutorial 2 Q6.)
c) Create an RNN model with 1 batch normalization layer after the input, 1 hidden layer containing 50 units, 1 more batch normalization layer and 1 output layer with number of units given by the number of future time steps to predict, i.e. n_steps. Use the Adam optimizer and Mean Square Error as the loss function.
d) Create an LSTM model with 1 hidden layer containing 50 units and 1 output layer with number of units given by the number of future time steps to predict, i.e. n_steps. Use the Adam optimizer and Mean Square Error as the loss function.
e) Determine the number of parameters that will be learned at different layers of each model by printing out `model.summary()` in TensorFlow. Account for the numbers of parameters at each layer of each type of recurrent network based on the structure of the recurrent networks.
f) Train the models using `fit()` for **1,000** epochs. Plot the loss function w.r.t. epochs to observe the model training. Compare the training loss of the 3 models. Comment on the effect of batch normalization for RNN and the performance of LSTM vs RNN.
g) Use the trained models to predict future values for the following test input [70, 80, 90] and comment on the results.