**National University of Singapore**
**College of Design & Engineering - ECE**

# EE4400 Data Engineering and Deep Learning
## Tutorial 2 - RNN

Q1. What is the unstable gradients problem in deep learning and how can it be overcome?

Q2. Add on batch normalization to Tutorial 1 Question 3 and comment on the differences observed in the performance and number of parameters. Why are there some non-trainable parameters?

Q3. What enables a recurrent neural network (RNN) to remember input signals or patterns that occur over several time steps?

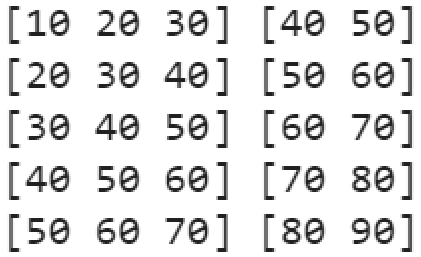
Q4. What input and output sequence type is useful for time series prediction? What should the output be in this application?

Q5. Why does batch normalization not work well in an RNN? What is a better approach for RNN?

Q6. Univariate Time Series Forecasting using RNN:

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

a) Convert the time series data given above into a sequence structure that can be used to train the simple RNN model. 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 time series sequence length seq_len = 3, and the number of future time steps to predict n_steps = 2.

For example:
Consider the first sequence = [10, 20, 30, 40, 50]
For this sequence, X = [10, 20, 30] and Y = [40, 50]

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.
c) Determine the number of parameters that will be learned by the model using `model.summary()` in TensorFlow. Explain the number.
d) Reshape X such that it can be fed as input to the model. TensorFlow models require the input shape to be of the form: (no. of sequences, sequence length, no. of features). Our dataset is a univariate time series so number of features = 1.
e) Train the model using `fit()` for 200 epochs. Plot the loss function w.r.t. epochs to observe the model training.
f) Predict future values for the following test input [70, 80, 90].

© CK Tham & R Rajagopalan ECE NUS