In this assignment, you will be addressing one application of recurrent neural networks (RNN), implementing in Python a multi-layer RNN for training/sampling from character-level language models. The model takes one text file as input and trains an RNN to learn to predict the next character in a sequence. The RNN can then be used to generate text character by character that will look like the original training data. The echo state network is addressed, as well as long short-term memory and gate recurrent units.

Get the available code “min-char-rnn.py” and “grad-check.py”. Using any IDE environment of your choosing, copy and paste the code. Download the character-level RNN, "karpathy/char-rnn," located in the Topic 7 resources, and train it on the “tiny Shakespeare” data set available at the same location. Consider the following

1. What is the RNN architecture used for reading the code “min-char-rnn.py”?
2. Create outputs of the language model after training for 5 epochs.
3. Create outputs of the language model after training for 50 epochs.
4. Create outputs of the language model after training for 500 epochs.
5. Check the gradient descent using the code “grad-check.py”.
6. What significant differences do you see between the three outputs?
7. What are the challenges encountered for training the RNN?

Recurrent Neural Networks are