Instructor: Dr. Naresh Manwani	Homework Assignment 2 Deep Learning: Theory and Practices (CS7.601), Monsoon 2022 Submission Deadline: **November 6th, 2022**	Total Marks : 20 Submit at: courses.iiit.ac.in

Instructions

1. This is a programming assignment and all questions are compulsory to attempt. Total Marks for the assignment is 30.
2. Your submissions are expected to be a .tar.gz file containing the folowing:
   - main.ipynb - A Jupyter notebook with Code and explaination. Please ensure that you include a **detailed report** of the results within the notebook using **markdown cells**.
   - Auxiliary python scripts (plotting, helper functions which need not appear in the main notebook.)
3. It is advised that you try out Google Colaboratory to ensure the code is reproducible.
4. **Use of Pytorch, Tensorflow or any other deep learning framework is allowed.**

Learning Long Term Dependencies [7 Marks]

Problem 1. There are $p+1$ input symbols denoted $a_1, a_2, \ldots, a_{p-1}, a_p = x, a_{p+1} = y$. a_i is represented by $p+1$ dimensional vector whose i^{th} component is 1 and all other are 0. A net with $p+1$ input units and $p+1$ output units sequentially observes input symbol sequences, one at a time, trying to predict the next symbols. To emphasize the long term lag problem, we use a training set consisting of only two sets of sequences: $\{(x, a_{i_1}, a_{i_2}, \ldots, a_{i_{p-1}}, x) | 1 \leq i_1 \leq i_2 \leq \cdots \leq i_{p-1} \leq p-1\}$ and $\{(y, a_{i_1}, a_{i_2}, \ldots, a_{i_{p-1}}, y) | 1 \leq i_1 \leq i_2 \leq \cdots \leq i_{p-1} \leq p-1\}$. In this experiment take $p = 100$. The only totally predictable targets, however, are x and y, which occur at sequence ends. Training sequences are chosen randomly from the two sets with probability 0.5. Compare how RNN and LSTM perform for this prediction problem. Report the following.

1. Describe the architecture used for LSTM and for RNN. Please also describe activation functions used, learning rate. **[2 Marks]**
2. Plot the number of input sequences passed through the network versus training error (for both LSTM and RNN). **[2 Marks]**
3. Does RNN based model takes longer time to converge than GRU or LSTM. Show using the error convergence plot. **[1 Mark]**
4. Once the training stops, generate 3000 sequences for test set.
5. Report the average number of wrong predictions on the test set in 10 different trials (for both LSTM and RNN). **[2 Marks]**

Detecting Temporal Order [13 Marks]

Problem 2. The goal is to classify sequences. Elements/ targets are represented locally. The sequence starts with 'E' and ends with 'B' and otherwise consists of randomly chosen symbols from the set a, b, c, d except for the three elements at positions t_1, t_2 and t_3 that are either X or Y . The sequence length is randomly chosen between 100 and 110. t_1 is randomly chosen between 10 and 20, t_2 is randomly chosen between 33 and 43 and t_3 is randomly chosen between 66 and 76. There are 8 sequence classes Q, R, S, U, V, A, B, C which depend on the temporal order of Xs and Ys. The rules are $XXX \longrightarrow Q, XXY \longrightarrow R, XYX \longrightarrow S, XYY \longrightarrow U, YXX \longrightarrow V, YXY \longrightarrow A, YYX \longrightarrow B, YYY \longrightarrow C$.

LSTM Use LSTMs for sequence classification with the following specifications.

1. Use 1-hot encoding (vector having only 1 non-zero element) for each input symbol.
2. Use 3 layer network with 8 input units. Use 3 cell blocks of size 2, 4 and 8 output units.
3. Use cross-entropy loss at the output.
4. While deciding the training/ test accuracy, a sequence is classified correctly if final absolute error of all output units is below 0.3.
5. The learning rate is 0.1.
6. Training is stopped once the average training error falls below 0.1 and 2000 most recent sequences were classified correctly.
7. All weights should be initialized between [-0.1,0.1]. The first input gate bias is initialized with -2.0. second input gate bias is initialized with -4.0 and third input gate with -6.0.

Report the following:

1. How many input sequences were generated in the training phase before it meets the stopping condition. **[1 Mark]**
2. Plot the number of input sequences passed through the network versus training error. **[2 Marks]**
3. Once the training stops, generate 3000 sequences for test set.
4. Report the average number of wrong predictions on the test set in 10 different trials. **[2 Marks]**

RNN Use RNN for classifying above sequences. Use the same specifications as mentioned above. Report the following.

1. Plot the number of input sequences passed through the network versus training error. **[2 Marks]**
2. Once the training stops, generate 3000 sequences for test set.
3. Report the average number of wrong predictions on the test set in 10 different trials. **[2 Marks]**

Attention Network Use RNN with attention for the above problem. You can use the same architecture of RNN as earlier. Report the following.

1. Plot the number of input sequences passed through the network versus training error. **[2 Marks]**
2. Once the training stops, generate 3000 sequences for test set.
3. Report the average number of wrong predictions on the test set in 10 different trials. **[2 Marks]**