Assignment 2
CS7.601 (Monsoon 2023)
Deep Learning: Theory and Practices
Submission Deadline:
11:55 PM, October 7$^{\text{th}}$ 2023

Max. Marks : 20

Instructions

1. Please submit your code for Q1 and Q2 in **Jupyter Notebooks**. Analysis should be included in the notebook itself using markdown cells. **Note that unlike assignment 1, analysis and reasoning of the results obtained will carry a significant weightage in the marks distribution**.

2. Only the following libraries are allowed:

   - Numpy
   - Pandas
   - Matplotlib
   - Pytorch (version $\geq$ 2.0)
   - Tensorflow (version $\geq$ 2.0) (Only allowed for assignment 2, assignments 3 and 4 will be pytorch only)
   - tqdm
   - scikit-learn

3. Question 3 will be released by 28$^{\text{th}}$ September. Instructions for Q3 will be released alongside it.

4. **Only submissions made on Moodle will be considered for evaluations.**

1 RNN For Auto Regressive Models [3 Marks]

In this exercise you are asked to generate an Auto Regressive model and then create an RNN that predicts it. Generate samples of an Auto Regressive model of the form

$$X(t) = a_1 X(t-1) + a_2 X(t-2) + a_3 X(t-3) + U(t)$$

where $U(t) \sim \text{Uniform}(0, 0.1)$, $a_1 = 0.6, a_2 = -0.5, a_3 = -0.2$. Generate 2000 training and 2000 test examples using this model. Now train an RNN that predicts the sequence. Apply the training algorithm on new samples and calculate the averaged cost square error cost function.

1. Investigate RNN with 1,2 and 3 hidden layers. Describe the architecture used for the network.
2. Plot the epoch-MSE curve during training.
3. Report MSE (mean square error), MAE (mean absolute error) and R^2 (R-square) on the test data.

Reference : https://en.wikipedia.org/wiki/Coefficient_of_determination

2 Learning Long Term Dependencies [7 Marks]

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. Error signals occur at every single time steps. To emphasize the long term lag problem, we use a training set consisting of only two sets of sequences: $\{(x, a_{i_1}, a_{i_2}, \ldots, a_{i_{p-1}}, x) \mid 1 \leq i_1 \leq i_2 \leq \ldots \leq i_{p-1} \leq p-1\}$ and $\{(y, a_{i_1}, a_{i_2}, \ldots, a_{i_{p-1}}, y) \mid 1 \leq i_1 \leq i_2 \leq \ldots \leq i_{p-1} \leq p-1\}$. 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. Also mention the activation functions, optimizer and other parameters you choose. Experiment around with multiple architectures and report your observations.
2. Plot the number of input sequences passed through the network versus training error (for both LSTM and RNN).
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 (for both LSTM and RNN).

Example : Consider $p = 5$. Suppose the set of random integers (i_1, i_2, i_3, i_4) are $\{1, 1, 3, 4\}$ respectively. Then, the sequence $\{(x, i_1, i_2, i_3, i_4, x)\}$ will have matrix representation as

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$