

Assignment 4  
CS7.601 (Spring 2020)  
Deep Learning: Theory and Practices  
Submission Deadline: 6.00 PM, 09/05/2020

Max. Marks : 30

## Instructions

1. Please submit your code in **Jupyter Notebook**. We will only consider submissions made using jupyter notebook with **detailed report** of the results included within the notebook using **markdown cells**.
2. You may use Python (Numpy, Pandas and Matplotlib) for your implementation.
3. Use of Pytorch, Tensorflow or any other deep learning frameworks is allowed.
4. Write your own codes.
5. You have to turn in the well documented code along with a detailed report of the results (zip of .ipynb files) electronically in Moodle. The report should be precise and concise.

## 1 RNN For Auto Regressive Models [5 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

$$X(t) = a_1X(t-1) + a_2X(t-2) + a_3X(t-3) + U(t)$$

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.
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.

## 2 RNN For Moving Average Models [5 Marks]

In this exercise you are asked to generate moving average model and then create an RNN that predicts it. Generate samples of a moving average sequence as follows.

$$X(t) = U(t) + a_1U(t-1) + a_2U(t-2) + a_3U(t-3) + a_4U(t-4) + a_5U(t-5)$$

where  $a_1 = 5$ ,  $a_2 = a_3 = a_4 = a_5 = -1$  and  $U(t) \sim \text{Norm}(0,1)$ . 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.
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.

## 3 Detecting Temporal Order [8 Marks]

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  $X$ s and  $Y$ s. The rules are  $XXX \rightarrow Q$ ,  $XXY \rightarrow R$ ,  $XYX \rightarrow S$ ,  $XY Y \rightarrow U$ ,  $YXX \rightarrow V$ ,  $YXY \rightarrow A$ ,  $YYX \rightarrow B$ ,  $YYY \rightarrow C$ .

### 3.1 Part1

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.

**Report the following.**

1. How many input sequences were generated in the training phase before it meets the stopping condition.
2. Plot the number of input sequences passed through the network versus training error.
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.

### 3.2 Part 2

Use RNN for classifying above sequences. Report the following.

1. Plot the number of input sequences passed through the network versus training error.
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.

## 4 Learning Long Term Dependencies [8 Marks]

There are  $p + 1$  input symbols denoted  $a_1, a_2, \dots, a_{p-1}, a_p = x, a_{p+1} = y$ .  $a_i$  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:  $\{(x, a_{i_1}, a_{i_2}, \dots, a_{i_{p-1}}, x) \mid 1 \leq i_1 \leq i_2 \leq \dots \leq i_{p-1} \leq p - 1\}$  and  $\{(y, a_{i_1}, a_{i_2}, \dots, a_{i_{p-1}}, y) \mid 1 \leq i_1 \leq i_2 \leq \dots \leq i_{p-1} \leq p - 1\}$ . 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. 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).

## 5 GANs [4 Marks]

Consider the following univariate distribution.

$$p(x) = 0.4\mathcal{N}(0, 4) + 0.3\mathcal{N}(-6, 4) + 0.3\mathcal{N}(6, 4), \quad (1)$$

where  $\mathcal{N}(\mu, \sigma^2)$  is a Gaussian distribution with mean  $\mu$  and variance  $\sigma^2$ . Generate 10000 samples i.i.d. from the distribution above. Train a GAN so that it can generate data from above distribution. Report the following.

1. Plot the epoch versus error curve.
2. Once the training stops, generate 3000 points from the GAN. Generate histogram plot of the these 3000 points and report your observations.
3. Use 3000 points generated above and fit a Gaussian mixture model (GMM) with three components using EM algorithm. Find the parameters of the GMM and compare them with the original distribution described above in equation(1). Find the KL divergence between the two distributions.