Homework Set on RNN
Sequential prediction using Recurrent Neural Networks (RNN)

1. **AR Model.** In this exercise you are asked to generate an AR model and then create an RNN (such as LSTM) that predicts it. Generate samples of an Auto Regressive model of the form

$$X_t = a_1 X_{t-1} + a_2 X_{t-2} + a_3 X_{t-3} + U_t \tag{1}$$

where $a_1 = 0.6, a_2 = -0.5, a_3 = -0.2$ and $U_t \overset{i.i.d}{\sim} \text{Uniform}(0, 0.1)$. Now train an RNN that predicts the sequence. Apply the training algorithm on new samples and calculate the averaged cost square error cost function.

   (a) Investigate different RNN structures and different number of training samples. (Important: You are not allowed to use your model knowledge when you design the RNN.)

   (b) Suggest a different kind of predictor and compare to the RNN.

   (c) Compare your RNN predictor to an estimator based on Yule–Walker equations with different model assumptions.

2. **MA Model.** Consider the following MA process:

$$X_t = U_t + a_1 U_{t-1} + a_2 U_{t-2} + a_3 U_{t-3} + a_4 U_{t-4} + a_5 U_{t-5} \tag{2}$$

where $a_1 = 5$, $a_2 = a_3 = a_4 = a_5 = -1$ and $U_t \overset{i.i.d}{\sim} \text{Norm}(0, 1)$.

   (a) Repeat the exercise above with a moving average model

   (b) Any MA process can be approximated using AR model. Assume an AR model with 5,10,50,100 and 250 coefficients and compare its prediction to the RNN prediction.

3. **Markov Process.** Generate a sequence of 10^5 samples from a Binary Markov random process with the following conditional PMF as given in Fig. 1.

   (a) Calculate the stationary distribution of this Markov process.

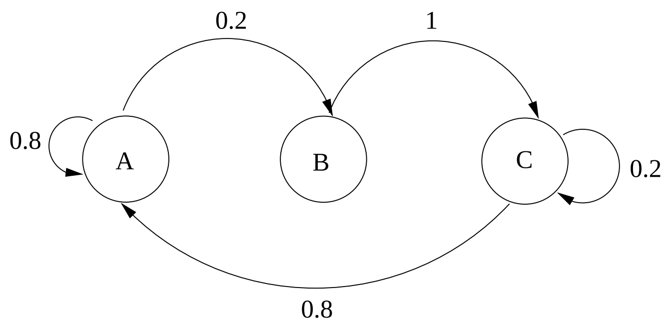

Figure 1: Markov process with three states. The edges indicates the conditional probability of the Markov process, i.e., $p(x_t|x_{t-1})$

(b) What is the entropy rate of a Markov process of order one. You may use the following lecture
`http://www.ee.bgu.ac.il/~haimp/multi/Lec1/lec1.pdf` and Chapter 4 in Cover book.

(c) Calculate the entropy rate of this process.

(d) Check if the empirical distribution of the samples that you have generated match the stationary distribution.

(e) Calculate empirically the entropy rate and compare to the actual one

(f) Generate a train and a test set, and using neural networks (try few configurations) to predict your next symbol. The cost is 0 if the prediction is correct and 1 if its wrong.

(g) Compare log-loss to the entropy rate.

(h) Prove that for i.i.d case the entropy is a lower bound for the negative log los and similarly for stationary processes entropy rate is a lower bound for negative log loss.

4. **Is it Markov or HMM** Consider the Markov process that is described by Fig. 2.

(a) Is it Markov or HMM.

(b) In general, entropy rate of HMM is an open problem and there exists only lower and upper bound. However in this case we can compute the entropy rate.

i. Show that there is a one-to-one between the HMM process in Fig. 2 and the process in Fig. 1.

ii. Prove that if two r.v. X^n and Z^n have a one to one mapping (i.e., there is a bijective function between X^n and Z^n) than $H(X^n) = H(Z^n)$.

iii. Conclude that the entropy rate of the HMM process in Fig. 2 and of the Markov process in Fig. 1 are equal.

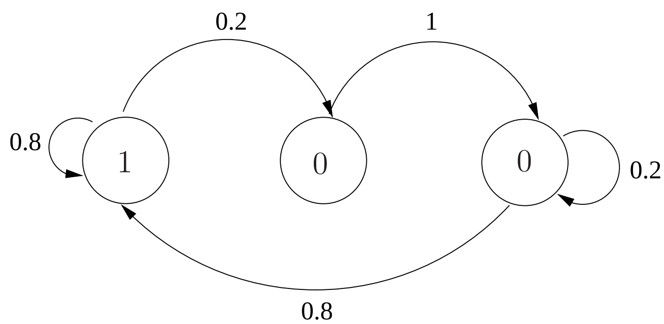

Figure 2: A Binary process that is described by a graph

(c) Generate a train and a test set, and using Neural networks (try few configuration) predict your next symbol. The cost is 0 if the prediction is correct and 1 if its wrong.

(d) Compare the negative log loss to entropy rate. Explain your result.

5. **HMM** Consider the Markov process given in Fig. 1. If the state is A the output of the channel Y is distributed according to $\sim$ Bernouli (p) and if the state is B or C the output is distributed according to $\sim$ Bernouli $(1-p)$. Generate data set for training and test and using a Neural network predict what is the correct state. Repeat the exercise for $p = 1, 0.9, 0.8, 0.7, 0.6, 0.5$.

6. **Parity check** Build a neural network that gets an arbitrary sequence of binary numbers and the output is a Binary sequence such that it indicates if the number of '1' till now is even or odd. For example:

Time	input	ouput
1	0	0
2	1	1
3	1	0
4	1	1
5	0	1
⋮	⋮	⋮

7. **Summing two numbers** Build a neural network that gets two sequence of binary number and the output is a binary sequence that is a sum of the two (the length of the sequence is not fixed). For example:

Time	input 1	input 2	output
1	0	1	1
2	1	1	0
3	1	1	1
4	0	0	1

   At time 1 the network gets the least significant bit and at time 4 it gets the most significant bit. Evaluate you NN on large sequences.

8. **Text Prediction (optional)** Do the assignment 4 that is given in https://piazza.com/ucsd/fall2015/190/resources, including the last assignment where an LSTM needs to be implemented.