Home Work 2: CAP 6610 Spring 2017

[Programming Assignment]

Due Date: March 15th,Wed

Please submit submit a zip file which has all the dataset, report and codes for the homework.We will not accept any submission without a zip file.

1. Single Node Neural Network.

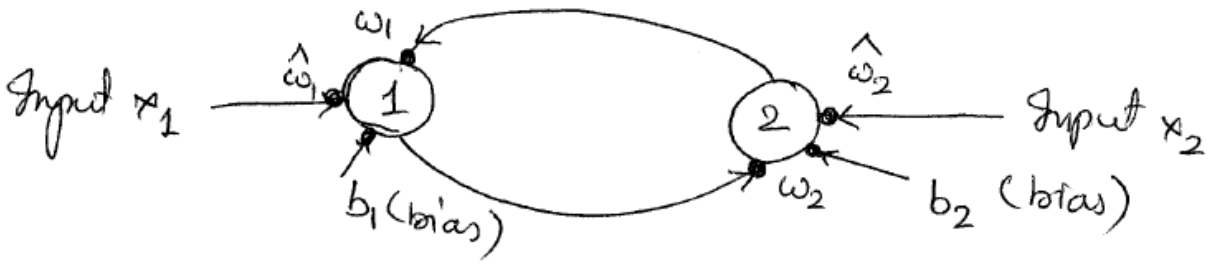
   (a) Here, you would implement a Single Node Neural Network using the Perceptron Learning Algorithm that we have proved in the class. The activation function is the $sign$ function. Use the Bank-Marketing dataset for this part. Choose 50-50 training-testing split.

   Please refer to the page 192 for detailed description.
   (goo.gl/iLCH7l This is the website link)
   https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
   Download The folder "bank.zip" from the Following link.
   https://archive.ics.uci.edu/ml/machine-learning-databases/00222/
   and use The "bank.csv" dataset.

   (b) Use Gradient Descent on a Single Sigmoid Neuron (e.g. $y = sigmoid(\sum w_i x_i)$) instead of $sign$ function (e.g. $sign(\sum w_i x_i)$) to learn the discriminant. use the same dataset of prev. question.

2. Code and test a two layer feed-forward net of sigmoidal nodes with two input units, ten hidden units and one output unit that learns the concept of a circle in 2D space. The concept is: $< x, y >$ is labeled $+$ if $(x-a)^2 + (y-b)^2 < r^2$ and is labeled $-$ otherwise.
   Draw all data from the unit square $[0,1]^2$. Set a = 0.5, b = 0.6, r = 0.4.
   Generate 100 random samples uniformly distributed on $[0,1]^2$ to train the network using error back-propagation and 100 random samples to test it. Repeat the procedure multiple epochs and with multiple initial weights. Report the changing accuracy and the hyperplanes corresponding to the hidden nodes (when the sigmoid is turned into a step function).

3. Consider the following 2 node Recurrent Neural Network(RNN).Each node receives input from that time step, and the output/state of the other node form the previous time step. Each node has a bias (that can be included by changing an additional input to 1, as described in the class.) your goal is to learn $w_1, \hat{w}_1, b_1, w_2, \hat{w}_2, b_2$ from a given dataset using error Back Propagation Through Time, as described in the class.
   The dataset format is $[x_1, x_2, y_1, y_2]$. It was created by an (unkown to you) RNN in the following manner.



- at $t = 0$: (initialization) the state/output of both node is set to 0.
- at $t = 1$: the nodes receives inputs x_1, x_2 respectively and the state/output of the other node from $t = 0$.
- at $t = 2$: the nodes receive inputs 0,0 respectively. (that is the inputs are turned off) and the state/output of the other node from $t = 1$.
- at $t = 3$: identical to the above, state from $t = 2$.

The state/output at $t = 3$, is recorded as y_1, y_2 (two nodes)