CS503– Machine Learning
Lab 2

Due on 22/2/2019 11.55pm

Instructions: Upload to your moodle account one zip file containing the following. Please do not submit hardcopy of your solutions. In case moodle is not accessible email the zip file to the instructor at ckn@iitrpr.ac.in. Late submission is not allowed without prior approval of the instructor. You are expected to follow the honor code of the course while doing this homework.

1. **You are to work individually for this lab.**
2. This lab can be implemented in any language that you are comfortable.
3. A neatly formatted PDF document with your answers for each of the questions in the homework. You can use latex, MS word or any other software to create the PDF.
4. Include a separate folder named as ‘code’ containing the scripts for the homework along with the necessary data files. Ensure the code is documented properly.
5. Include a README file explaining how to execute the scripts.
6. Name the ZIP file using the following convention rollnumberhwnumber.zip

---

In this question you will implement an MLP to classify digits from the MNIST dataset. More specifically, you will compute the gradient using Backpropagation and perform stochastic gradient descent on the data. Let us consider a network with H hidden units, K outputs, and inputs of size D; we can then define the following parameters:

- W - a $H \times (D+1)$ matrix of weights for the connections between input to hidden layer units. The h^{th} row of this matrix corresponds to the weights of the input connections starting from the bias term to the D^{th} input term coming into the h^{th} hidden layer unit.
- V - a $K \times (H+1)$ matrix of weights for the connections between the hidden layer to output layer units. Similarly, the i^{th} row of this matrix corresponds to the weights of the hidden layer connections starting from the bias term to the H^{th} hidden layer term coming into the i^{th} output layer unit.

We will use $\tanh$ as the activation function for the hidden units instead of the sigmoid function. $\tanh$ has profile similar to the sigmoid function but has a range [-1 1]. Now for some input x we can write the output of the hidden layer units z as

$$z = \tanh(Wx)$$

where the $\tanh$ operation is performed element wise on the vector Wx. As the goal is to classify digits, the softmax function will be used in the output layer units and therefore the output of the i^{th} unit will be

$$y_i' = \frac{\exp v_i^T z}{\sum_{k=1}^{K} \exp v_k^T z}$$

The output y_i' can be interpreted as the probability that the input is of class i. Given N data samples $\{(x_i, y_i)\}_{i=1}^{N}$, let y_{ni}' be the i^{th} output evaluated for the n^{th} sample. Then we can write the error on this set as

$$-\sum_{n=1}^{N}\sum_{i=1}^{K} y_{ni} \log y_{ni}'$$

a. Derive the gradient of this error function with respect to each weight parameter. Note that the derivative of $h(a) = \tanh(a)$ is given by $h'(a) = 1 - h(a)^2$.
b. Implement a MLP to classify the MNIST digits. The dataset is included in the accompanying zip file ("data.txt" and "label.txt"). Set the number of hidden layer units to be 500, learning rate to be 0.01. We will use a modified form of stochastic gradient descent, where instead of updating the weights after every input, the updates are made in batches of input data. Set the batch size to be 25. Set the number of epochs to be 100. Divide the data into train, validation, and test splits using a preset ratio.
c. Include a plot of the training and validation error as a function of epochs and at most 2 instances for each of the 10 digits that are misclassified from the test set. These plots must be the average over 5 trials. Plot the mean and variance.
d. Extend this a convolutional neural network by implementing the convolution operation with stride. The convolution network will consist of two convolution layers; the first convolution layer has of 50 filters of size 3x3 with a stride of 2, the second convolution layer has again 50 filters of size 3x3 with a stride of 2. The output of the second convolution layer is vectorized and connected to the final output layer. Assume that the convolution operation is following by a sigmoid activation.

Note that the training process might take a while, so make sure that you don't wait till the last minute to start working on the homework.

An important aspect of machine learning is reproducibility of the results presented in a paper/report. Therefore, we will run your code to see if the results are closely matching with what you have presented in the report. Any deviation beyond a reasonable threshold will be considered as fudging of results and will invite severe penalty.