CSL712– Advanced Machine Learning
Lab 1

Due on 1/2/2017 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. This lab should be attempted individually.

2. 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.

3. Include a separate folder named as 'code' containing the scripts for the homework along with the necessary data files.

4. Include a README file explaining how to execute the scripts.

5. Name the ZIP file using the following convention rollnumberhwnumber.zip

---

You will be experimenting with logistic regression and multi-layer perceptron using different activation and loss functions on the CIFAR10 dataset. Implement the lab in Python.
Request access to the cloud server using this form. This machine has sufficient resources to execute this lab. You do not need GPU access for this lab. http://goo.gl/forms/1WLWjBC5CZEBVWms2

Download the CIFAR10 dataset.

Split the train set further into a train and validation set.

Perform preprocessing on the data set

- Does the choice of activation function decide the data pre-processing operation? If Sigmoid is the activation function, would you prefer standardization or only subtracting the mean of the data? What about in the case of a ReLU activation function?

Implement multinomial logistic regression (with regularization) using mini-batch stochastic gradient descent

- Start with a batch size of 200, and a learning rate of 1e-7

- What type of relationship exists between the learning rate and batch size?

Implement the multinomial logistic regression as a one layer neural network consisting of a linear transformation layer (adder) and softmax layer. The neural network must also perform regularization on the weights. Ensure that the implementation of the neural network is modular. This will assist in the next set of implementations

- You should be able to get the same output as before for the same set of initial parameters.
- Tune the hyper parameters of the network (learning rate and weight decay)

Implement a two-layer neural network with ReLU as the activation function.

- Train a neural network with the following architecture – linear – relu-linear-softmax. Report the accuracy obtained by this network on the test set

Implement the following activation functions – tanh, Sigmoid and Leaky ReLU. For each activation function, you must also ensure that the derivatives are computed correctly for the backward pass.

- Train a two-layer neural network same as before, but replace the relu layer with each of the new activation functions. Compare the performance of these activation functions

The default loss function so far has been the cross-entropy error. Change the loss function to hinge loss. Again, this should be modular that you can re-train the old networks to use hinge loss with minimal changes.

- It might be advantageous to first write the weight update equations using the hinge loss.