Assignment-1

Neural

Network

Training

In this assignment you will use and train a fully connected neural network and a CNN to classify the digits in the MNIST dataset.

1) The neural network should incorporate the following features. (Use backpropagation)

   a) No. of hidden layers = 1
   b) No. of neurons in the hidden layer1 = 100
   c) Activation function of hidden layer: Sigmoid function
   d) No. of neurons in the output layer = 10
   e) Activation function in output layer: SoftMax function
   f) Input: MNIST data (28*28)

2) The convolution neural network should have the following with descriptions. (LeNet model). (Use backpropagation)
   a) Convolution layer 1: 5*5, with 6 output channels
   b) MaxPool1: 2*2, stride=2
   c) Convolution layer 2: 5*5, with 16 output channels
   d) MaxPool1: 2*2, stride=2
   e) Linear Layer1: input (calculate yourself based on conv and pooling layer), output=120
   f) Linear Layer2: input = 120, output=84
   g) Linear Layer1: input=84, output=10
   h) Use ReLU as activation function in each convolution and linear layer
   i) Input: MNIST data (28*28)

3) Classify the digits (0-9) in an MNIST data set (28*28). Use Stochastic Gradient Descent (SGD) for optimization.

4) Calculate following:
   a) Training accuracy
   b) Testing accuracy
   c) Training loss
   d) Testing loss

5) Draw graph showing the loss vs epoch.

6) Visualize your model's behavior:
   a) Draw the learning curve showing how train and test error vary with increase in batch size.

   b) Plot the test error vs learning rate for each of the model. Vary the learning rates as 0.025, 0.05, 0.1, 0.2 and 0.5. Which learning rate would you prefer for this problem and why? (write this answer in your python notebook itself.)

   c) Now instead of keeping learning rate fixed for the complete training, vary your learning rate with number of epochs (you can take any learning rate value and decrease it as per your choice). Observe rate of convergence of model. Write your observation in brief.

   Description: Basically, you have to build a Stochastic Gradient Descent with Restarts (SGDR), a variant of learning rate annealing, which gradually decreases the learning rate through training.
   You have to decrease learning rate by steps: use one learning rate for the first few iterations, then drop to another learning rate for the next few iterations and then drop the learning rate further for the next few iterations.

Instruction for submission:
- You have to submit separate python notebooks for each model.
- Attempt question 3-6 for both the models.
- Give heading of respective question number in notebook and then show the output.
- Write answer of 6.b and 6.c in python notebook itself at the end.