Assignment

Note: Experiments (1 to 6) may be implemented in bare python or pytorch.

1. Consider a neural network with 4 layers of 500 neurons each. Initialize the weights of each of these layers to samples from Gaussian with mean 0 and standard deviation 0.01. The i/p layer consists of 1000 features randomly sampled from the standard Gaussian. Do only the forward propagation once of the input with tanh as the activation function in all the layers. At the end of forward propagation, plot layer-wise (including input layer) histogram of activation values. What do you observe?

2. Repeat the above experiment for sigmoid activation. What do you observe?

3. Repeat the above experiment for ReLU activation. What do you observe?

4. Repeat the above experiment again for tanh, sigmoid and ReLU, but now with weights initialized to samples from standard Gaussian. What do you observe?

5. Now repeat the above experiment with Xavier Weight Initialization on all the 3 activations. What do you observe?

6. Now repeat the above experiment with He Weight Initialization on all the 3 activations. What do you observe?

7. In the first lab assignment, use ReLU activation (in the hidden layers) and He weight initialization and train the model. Does the model train faster? Does it perform better on valid/test set?

8. In the first lab assignment, use ReLU activation (in the hidden layers) but allow the weights to be automatically initialized by pytorch linear module. Add batch normalization layer after every application of ReLU and train the model. Does the model train faster? Does it perform better on valid/test set? Compare the results with results from experiment 7.

9. In the first lab assignment, use ReLU activation (in the hidden layers) but allow the weights to be automatically initialized by pytorch linear module. Deploy drop-out in every hidden layer. Retain/drop probabilities may be chosen by trial and error. Does the model perform better?

10. Instead of drop-out, use L2 regularization. Regularization param may be chosen by trial and error. Does the model perform better?

The code and observations should be done in colab and submitted. Also, go through the tutorial link given in lab assignment (if you have not already done so), with particular focus on datasets and data loaders. Spend some time on going through torchvision documentation.