Assignment: Feedforward Network Layer Robustness

The goal of this assignment is to investigate the robustness of a layer and how it changes, as we increase the depth of the net.

Case 1: You need to build and train multiple networks of different depths on MNIST data as follows.

1. 2 hidden layers, each having 50 neurons
2. 4 hidden layers, each having 50 neurons
3. 8 hidden layers, each having 50 neurons

Use MNIST train data to train your model. Keep all the hyper-parameters, the optimizer same for all the networks. Use cross-entropy error function. Use Kaiming initialization. Test your networks on MNIST test data and produce the test accuracies for each case.

Case 2: Now take each trained network and for each hidden layer, perturb the parameter (add some noise) associated with the layer (the parameter connecting layer $l\text{-}1$ and l), while keep the other layers intact as in the original trained network. For perturbation, use the Normal distribution used in Kaiming initialization. Test the perturbed model on MNIST test data and measure the deviation of performance compared to the unperturbed trained network. Thus, if a network has two hidden layer and one output layer, then there will three networks with perturbations. Then, rank the layers based on their performance deviation (in descending order).

Submission guidelines:

You have to submit a python file with the source code and it must clearly document the codes for case 1 and case 2 of the problem statement. Make sure that the code produces the following output on the screen while on runs the python file.

1. **For case 1:** It should show the accuracy on the MNIST test data.

2. **For case 2:** For every network of case 1, print the layers based on their performance deviation (in descending order).

You have to submit the code in Moodle.

Deadline: 20th October, 2023, 11.55 PM IST.

NOTE: Plagiarism will be penalized.