AI42001 – MLFA

1. Download CIFAR10 dataset from torchvision. Use the official train and validation/test split for training and validation/testing.
2. Create a Convolutional Neural Network (CNN). You are free to create your own model, free to choose any optimizer, learning rate and batch size. Try to build the model and select hyper-parameters such that you get good accuracy (above 70%). Try creating networks with/without batchnorm and/or dropout.
3. Re-Run all the experiments shown in the notebook that was demoed in class.
4. These experiments are:
    a. Run the CNN for 25 epochs to get baseline results.
    b. Perform Lottery Ticket pruning for 5 rounds with each round consisting of 5 epochs.
    c. Run point (b) for 5 different sparsity levels, i.e., 0.1, 0.2, 0.3, 0.4, 0.5
    d. Now perform random pruning as described during the lab demo for 5 rounds with each round consisting of 5 epochs.
    e. Run point (c) for different sparsity levels, i.e., 0.1, 0.2, 0.3, 0.4, 0.5
5. Plot a single multi-line graph as shown in the notebook.
6. The Lottery Ticket style pruning has three hyper-parameters. Mainly the number of rounds, the number of epochs per round and the sparsity level. Create a new experiment where you look at the effect of the number of rounds on model performance. Specifically, fix the sparsity level to 50%, and perform the lottery ticket style training and pruning for the following configuration:

Number of Rounds	Number of Epochs per Round
1	30
2	15
3	10
4	7
5	6

    Plot a single multi-line graph showing the validation accuracy for each of the configuration listed above.
7. Finally, run an experiment wherein, instead of resetting the weights back to the initial weights after pruning, you initialize the weights randomly. You can use the following code *“torch.nn.init.kaiming_uniform_ (layer1.weight)”* for initializing your weights. Report the validation accuracy. Set sparsity level to 50%, number of rounds to 5 and number of epochs per round to 5.

IMP: Do not use any pruning library