COL341 Spring 2023
Assignment 4: Neural Network
(To be done Individually)

Due Date: 29th April 2023, Saturday, 11:55 PM (No extensions)
Demo Date: 30th April, 2023 Sunday (Fixed)
Total Points: 100

1 Introduction

In this assignment, you will train a convolutional neural network (CNN) to classify images. In the first part, you will implement a CNN from scratch including the forward propagation, and back propagation using only `Numpy`. In the second and third parts, you will use the `PyTorch` framework. The details are given in following sections.

2 Dataset description

The CIFAR-10 dataset consists of 60,000 32×32 color images in 10 classes, with 6,000 images per class. The classes are mutually exclusive and there is no overlap between them. You need to download the dataset from official site [1]. The data is already split into training and testing sets of size 50000 and 10000, respectively. Since this is already a widely used public dataset, we won't use any held-out part in this assignment. You will be graded based on your analysis and efforts. You need to use the testing split as the validation data.

3 Part 1: Implement a Neural Network [40 Marks]

In this part you will implement the following neural network model from scratch.

- `CONV1:` Kernel size (3×3), In channels 3, Out channels 32.
- `POOL1:` Kernel size (2×2).
- `CONV2:` Kernel size (5×5), In channels 32, Out channels 64.
- `POOL2:` Kernel size (2×2).
- `CONV3:` Kernel size (3×3), In channels 64, Out channels 64.
- `FC1:` Fully connected layer (also known as Linear layer) with 64 output neurons.
- `FC2:` Fully connected layer with 10 output neurons.

Use `ReLU` as the activation function for all layers apart from the max-pooling layers, and the `FC2` layer. You need to flatten the `CONV3` output before passing as input to `FC1`. Implement the forward and the backward passes and complete the code for the `Conv2D, MaxPool2D, Linear,` and `ReLU` layers and use these to implement the network. Complete the training code, and train for 20

[1] https://www.cs.toronto.edu/~kriz/cifar.html

epochs using an Adam optimizer with learning rate 0.001 and batch size 32. Use the categorical cross entropy loss as the loss function. Plot the training epochs vs. the training and validation losses (loss curve [2]). Report the class-wise validation accuracy. Also plot the epoch vs. overall validation accuracy. Implement your source code in an interactive **Jupyter Notebook** (`.ipynb` format).

4 Part 2: Implement a PyTorch-based Solution [35 Marks]

You need to re-implement the network defined in the previous part using `PyTorch`. You can look into the official tutorial [3]. You may use the training code from this example with required modifications. Use `PyTorch v1.12.1` for your assignments. Installation instructions are here [4]. You can use GPUs from Google Colab [5] for faster completion of the experiments. Colab supports notebook by default. If you are working on your laptop then try setting up the following environment variables in your local machine if the training is slow:

```
export OMP_NUM_THREADS=4
export MKL_NUM_THREADS=4
```

The purpose of this part is to get you familiar with the interplay between different hyper-parameters involved in training a neural network. The final outcome of a neural network model is influenced heavily by the choice of the loss function, optimizer, learning rate, batch size, etc. You will perform the following experiments to understand the role of different hyper-parameters:

4.1 Hyper-parameter Tuning [25 Marks]

For all experiments, analyse and report the effect of varying the following hyper-parameters. Choose Cross-Entropy as the loss function. Use both Adam and SGD optimizers. Use any suitable data augmentation [6] [7] scheme.

1. `Learning Rate (LR):` Use initial LR of `0.001`. Use two other LR values according to your judgement and report your observations on the training and validation loss curves. Also discuss the effect on class-wise accuracy. You may use a suitable number of epochs and any batch size.

2. `Variation in LR:` Apart from using a fixed LR, use a learning rate scheduler of your choice to vary the LR over the epochs. Here is a good starting tutorial on LR schedulers [8]. Study the effect of fixed LR vs. varying LR over epochs using any scheduler and report your analysis. You may use any LR value of your choice. Report the training and validation loss curves, and class-wise accuracy. What are your observations? You may use a suitable number of epochs and any batch size for this experiment.

3. `Number of Training Epochs:` You may choose a suitable value of the training epochs. Does increasing the number of training epochs always help? Hint: how does loss and overall accuracy (both training and validation) change over the epochs? You may use suitable LR and any batch size for this experiment.

[2] https://niruhan.medium.com/drawing-loss-curves-for-deep-neural-network-training-in-pytorch-ac617b24c388

[3] https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

[4] https://pytorch.org/get-started/previous-versions/

[5] https://colab.research.google.com/

[6] https://pytorch.org/vision/main/transforms.html

[7] https://towardsdatascience.com/a-comprehensive-guide-to-image-augmentation-using-pytorch-fb162f2444be

[8] https://towardsdatascience.com/a-visual-guide-to-learning-rate-schedulers-in-pytorch-24bbb262c863

4. `Batch Size`: Vary the batch size $\{4, 8, 16, 32\}$ and plot the batch-size vs. overall validation accuracy curves. You may use suitable LR and number of epochs for this experiment.

4.2 Effect of Loss Function [5 Marks]

Choose any hyper-parameter set-up from the previous experiments. This time, vary the loss function. Experiment with KL Divergence Loss as well. Compare the performance with the Cross-Entropy Loss in terms of convergence, and class-wise accuracy.

4.3 Effect of Data Augmentation [5 Marks]

Choose any hyper-parameter set-up and any loss function. Now turn off the data augmentation. What effect do you observe on the class-wise and overall validation accuracy as compared to training with data augmentations? Hint: find the training accuracy as well. Does it ring a bell?

5 Part 3: Improve the CNN Model [25 Marks]

In this part, you need to improve the neural network model to boost classification accuracy. This part can be fully `PyTorch`-based. You can choose to add more layers, change the layer/ kernel parameters, add different architectural components, change the training regime and hyper-parameters – whatever you deem fit. However, you need to fully justify your design choices. Why did you choose to add/ remove a particular component? Explain in detail. Like before, show the loss curves and the class-wise accuracy. You may also use your observation on these metrics to justify your design choices quantitatively.

6 Submission Instructions

Submit all source code, and a detailed typed report. **The source code should be in a compiled notebook (`.ipynb`) format showing all computation and output for assessment**. Download colab workbook as a `.ipynb` format notebook file. You may use a single notebook for all parts with clear demarcation between them, or a different notebook for each part. The report should clearly define each experiment set-up, parameters, and the relevant observation and analysis (and plots if required). You must zip the source code, and the other files (including the report) in a single zip file, rename the zip file as `<Your-Entry-Number>.zip` (e.g., `2019CSZ8406.zip`), and submit in Moodle.

7 Note

- The assignment is not very difficult, but requires careful planning and continuous execution effort from the beginning. Trying at the last moment is a guaranteed way of messing up.
- Plagiarism or academic integrity violations will attract penalty as per the course policy. If you use any public resources, do not forget to cite them.
- You are required to be prepared for a demo on April 30th, 2023. If you miss the demo due to an emergency, then it is your responsibility to contact the TA with justification and complete your demo before May 12th, 2023. Without the demo, the submission will not be evaluated.
- The deadline of the assignment is fixed and will not be extended. Medical issues or other emergency cases will be dealt on a case-to-case basis. You need to contact the course instructor at the earliest for such issues.