E0 270: Assignment 2

Due date: April 13, 2024

Instructions

1. There is no assignment template this time. You are free to manage you codebase as you deem fit. However, please note that your code is written in python and not Jupyter (iPython) notebooks.
2. We are not restricting usage of any libraries for this assignment. However, setting up the environment, running into technical problems, and figuring out their solutions is a part of the learning process. Use Google, Stack overflow, and discuss with each other. Unfortunately, we will not be able to help you.
3. The report must be typeset in LaTeX, Ensure that everything is in a single pdf file and all pages are in correct order.
4. If you feel any particular problem in under-specified, you can safely assume that it has been done intentionally. You are free to make assumptions, but please specify the assumptions in your report. No further clarifications on the problem will be provided. However feel free to contact us if you believe that there is an error in the problem specification.
5. Copying codes from others verbatim, along with the use of LLM assisted code generators are strictly forbidden, and will be penalized heavily if detected.
6. We understand that the assignment might be computationally intensive. In case you want access to computational resources, kindly send an email to Akshay Nath (akshaynath@iisc.ac.in and CC Prof Ambedkar) asking for account on the Amazon server until April 30, 2025.

Data

Consider the CIFAR-10 dataset https://www.cs.toronto.edu/ kriz/cifar.html. Each data point is a $3 \times 32 \times 32$ image with Red/Green/Blue channels.

Image Classification

Consider the code in github.com/parag1604/Basic-Image-Classification as your starting point.

0. Run the version 9 by executing "python main.py 9". This should automatically download the CIFAR dataset. After the model has been trained, save the CNN model by giving appropriate name. Experiment by changing the hyper-parameters, loss function, etc. and see if you can achieve a better test set accuracy.
1. Replace the CNN model by a ResNet (arxiv.org/abs/1512.03385) and retrain the model on the training set. Do not use the resnet models from torchvision, rather take a look at the following repository: github.com/techxzen/pytorch-residual-networks and use the ResNet model class defined there. You need to train three ResNet models (until convergence) with depth 20, 56 and 110 respectively. Ensure to save the models by giving appropriate names.

2. In the aforementioned repository, along with ResNet you will also find a PlainNet model class. Train three PlainNet models with depth 20, 56, and 110 and save the model checkpoints.

3. Report the test set loss and accuracies for each of the 7 models trained. Optionally, you can also attach the training plots in case you log the training loss and accuracies.

Visualizing the Loss Landscapes

Consider the following paper: arxiv.org/pdf/1712.09913.

1. Refer to page 3 of the paper and look into the methods described in Section 3 and 4.

2. Implement the "Contour Plots & Random Directions" visualization method after applying the "Filter-Wise Normalization" method proposed in the paper. Upon running the implemented visualization code for any of the saved models, you should get 3D loss landscapes as shown in page 1 of the paper. Report these 3D landscapes for all the 7 models.

3. Report the contour plots of the 7 saved models as shown in page 7 of the paper. Which model is better (PlainNet vs ResNet) and why?

Hint: You may want to look at the video: youtube.com/watch?v=78vq6kgsTa8 to know more about the method proposed in the aforementioned paper.

Deliverables

- The complete codebase for training and visualization.
- A report containing the appropriate details of your implementation, along with justification for any extra assumptions made if necessary.
- Report should also contain accuracies, 3D landscapes, and contour plots for all 7 models.

Submission

Attach **a single zip file** named in the format `Asst2_FirstName_LastName_5DigitsOfSRNo.zip` to the assignment in Teams, before the due date. The zip file should contain your code, your report (in a pdf file named `Report_<5digitSR>.pdf`), and any plots that you put in the report. Ensure to upload the zip file on Teams submission portal on or before April 13, 2025, 11:59 pm. Ensure that you do **not** include the data files and **pycache** files in the code that you submit.