Homework 4
600.482/682 Deep Learning
Spring 2019

February 28, 2020

Due Sunday 3/15 11:59pm.
Please submit a report (LaTeX generated PDF)
and the code to Gradescope with entry code 9G83Y7
Please start early as model training might take a long time
and Kaggle will not accept late submission

1 Problem 1

Implement CNNs that classify images using PyTorch with the tricks and architectures you learned in the class.

Dataset Train and tune your network based on our modified version of CIFAR-100. Load the dataset with the code snippet provided in `https://www.kaggle.com/t/f63012ec31444360a7ae3afa3c6c884e`.

The modified CIFAR-100 is a subset of the original CIFAR-100. Please do not train your network with the original CIFAR-100 dataset because we reserve some samples to test your network. Training on the original dataset disqualifies you for extra credit.

Implementation Implement 3 variants of models. You may come up with your own or use state-of-the-art models as a starting point. If you start with someone else's models, please do more than hyperparameter tuning. Cite anything that inspired you in your report.

You must use at least 2 of the following in each model and at least 5 across all models. You are free to use anything not listed here.

1. dropout
2. batchnorm
3. skip connection
4. transfer learning
5. data augmentation
6. regularization
7. batch size, learning rate, learning rate schedulers and different optimizers

Please use a fixed random seed for reproducibility.

Note The dataset is difficult. Do not be discouraged if you get 10% initial accuracy. Anything above 50% is very good. You will not be graded based on the test accuracy.

Deliverables and grading

1. An informal report. Briefly explain the highlights of each model and your intuitions. Include a dev accuracy and loss plot for your best-performing model. Present how the architecture and tricks affect the performance with *abalation study tables*. Your report will be graded based on the depth of your analysis. Do not write an introduction. Max 2 pages.

   Tips: An ablation study refers to removing some part of the model, and showing how that affects performance. For example, to show the benefit of using optimizer A vs B, train two identical models except using optimizer A or B, and see how it effect *dev* set performance.

2. Code. You can submit Jupyter Notebook (Google Colab) `.ipynb` or a Python project or both. If you have dependencies other than numpy and pytorch, please install them inline in Jupyter Notebook (for Google Colab) or provide a `requirements.txt`. Document how to run the training loop in your report. Provide high-level comments on what is not obvious. Your code will be graded based on consistency with the report, correctness, and good practice.

3. Kaggle. Submit your test result to the Kaggle competition https://www.kaggle.com/t/f63012ec31444360a7ae3afa3c6c884e. You will receive a participation grade. The top 30% participants will receive a extra credit. Please report your team name in the report.

   Tips: If you want to be anonymous on the leader board, please create a new temporary Kaggle account, ***not revealing your name****.*