Neural Networks and Deep Learning: Lab Assignment 2

Summary: In this assignment, you will demonstrate you understand how to train, evaluate, and analyze convolutional neural networks (CNNs) with regularization. Your submission should include one PDF file with two separate, self-contained parts: one being a report describing your methods, results, and analysis and the other part showing your code.

1. **Influence of Regularization [15 points]:**

   (a) Design and conduct your experiment (Code)

   - Load a real dataset not covered in class that is designed for an image classification problem. If you are seeking assistance for how to find a dataset, one good option is to revisit the TA's programming tutorials. You can pre-process the dataset if you wish.
   - If the dataset you choose does not already come in a train/validation/test split, then divide it into a 70/15/15 train/validation/test split of the dataset.
   - Train at least eight CNN models using all combinations of at least two different numbers of convolutional layers with (a) at least three regularization techniques covered in the course and (b) no regularization. Select and set all other hyperparameters to be identical when training each model; e.g., number of iterations for training, optimization approach, number of filters, etc. Recall that regularization techniques covered in the lecture about regularization include parameter norm penalties, early stopping, data augmentation, dropout, and batch normalization. While you can select any architecture you want, recall that one popular, proven approach is to alternate between convolutional layers and pooling layers followed by fully connected layers until the output layer.
   - For each trained model, compute how long it takes to train it and produce a plot that shows the learning curves on both the training and validation splits with respect to the accuracy metric (i.e., each plot should show two curves indicating the computed accuracy with respect to the number of training iterations/epochs).
   - Train a new model on all of the training and validation data using the top-performing model from all models tested on the validation set. Then, report the resulting model's performance on the test set using the accuracy metric.

   (b) Report your methods, results, and analysis (Report)

   - Describe the methods you used for your experiment such that the reader could reproduce your experiments. This should include a discussion of the dataset (e.g., source? number of examples?), what neural network architectures and hyperparameters were used to train all the models, and what type of hardware was used during training.

- Report how long each model took to train and show the plots that visualize the learning curves for every tested model.
- Indicate which model configuration led to the top-performing model on the validation set and report the performance of the final model that was evaluated on the test split.
- Discuss your analysis of general trends that emerge from your results. Your discussion should consist of 2-4 paragraphs. To format each paragraph, please first identify one general trend observed from the results and then offer insights/speculations into **why** you think the trend/results may occur (regardless of whether you deem the results good or bad). Possible trends to consider include: What is observed when comparing the use of regularization techniques to not using any regularization during training? Did a certain number regularization techniques or network depth (i.e., from different numbers of convolutional layers) lead to consistently better results? What, if any, insights are gained by looking at the learning curves (e.g., overfitting vs underfitting)? What do you see as the trade-offs between training time and different choices for the number of convolutional layers and regularization techniques? How does the performance compare for the top-performing model configuration when tested on the validation set and test set?

2. **Interpreting CNN Representations [10 points]:**

   (a) Design and conduct your experiment (Code)

   - From the previous problem, use the same dataset with splits and the same final model.
   - Implement at least one technique for interpreting what your image classification model learned. Recall that techniques covered in the course include (but are not limited to) visualizing convolutional filters, visualizing activation maps, retrieving images with similar feature representations, identifying images that maximally activate neurons in a network, and visualizing the separability of classes using pretrained features extracted from the model. Apply this technique to at least two different layers of your network to support comparative analysis.

   (b) Report your methods, results, and analysis (Report)

   - Describe the methods you used such that the reader could reproduce your experiments.
   - Discuss your analysis of general trends that emerge from your results. Your discussion should consist of 2-4 paragraphs. To format each paragraph, please first identify one general trend observed from the results and then offer insights/speculations into **why** you think the trend/results may occur (regardless of whether you deem the results good or bad). Possible trends to consider include: What, if anything, can you infer about what the model learned? How does what is learned differ at different layers of the network?

How to Submit Lab Assignment 2: Please submit a pdf named with your first and last name; i.e., firstname_lastname.pdf. A successful submission will consist of two self-contained, separate contributions. First, it should include a report describing all methods, results, and analysis (i.e., portions indicated by "Report") **as the first part of the PDF file**. Second, it should include the source code of your implementation **as the second part of the PDF file** (i.e., portions indicated by "Code").[1] We will only review the code in detail when the report is not a self-contained document, in order to provide partial credit.

Collaboration versus Academic Misconduct: Collaboration with other students is permitted, but the work you submit must be your own. Copying/plagiarizing work from another student is not permitted and is considered academic misconduct. For more information about University of Colorado Boulder's Honor Code and academic misconduct, please visit the course syllabus.

[1] We require submitting the code as a PDF to avoid many issues that we have observed in the past with being able to access submitted code. These issues have arisen, in part, because we make no programming language requirements. Issues also have arisen from students not providing read permissions for links to their files; e.g., for Google Colab.