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 that merges two separate PDF documents: one showing your code and another showing your methods, results, and analysis.

1. **Influence of Network Depth and Regularization Techniques [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.
   - Create (or load) a 60/20/20 train/val/test split of the dataset.
   - Train at least nine CNN models using all combinations of at least three different numbers of convolutional layers with (a) at least two 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 course include parameter norm penalty, 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 epochs used during training).
   - 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 (Write-up)

   - 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 parameters were used to train all the models, and what type of hardware was used during training. For full credit, ensure that your write-up is written formally such that it could be included in a research report/publication.

- 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 tested on the evaluation split.
- Discuss what general trends emerge from your results. Some suggestions for topics to discuss are listed here. What is observed when comparing the use of regularization techniques to not using any regularization during training? Did a certain number of convolutional layers or regularization techniques 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? For full credit, offer insights/speculations into why your results may be turning out the way they are. Your discussion should consist of 2-4 paragraphs.

2. **Impact of Training Data Amount [10 points]:**

   (a) Design and conduct your experiment (Code)

   - Use the same dataset with splits from the previous problem.
   - You should train at least six models in this experiment by using all combinations of at least two different regularization techniques covered in the course with the following amounts of training data: train with 50%, 75%, and 100% of the training data respectively. As done for the previous problem, select and set all other hyperparameters to be identical when training each model.
   - 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.

   (b) Report your methods, results, and analysis (Write-up)

   - Describe the methods you used for your experiment such that the reader could reproduce your experiments. For full credit, ensure that your write-up is written formally such that it could be included in a research report/publication.
   - Report how long each model took to train and show the plots that visualize the learning curves for all tested models.
   - Discuss what general trends emerge from your results. Some suggestions for topics to discuss are listed here. How is the performance of each regularization approach affected by the amount of training data available? 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 performance when using different amounts of training data? For full credit, offer insights/speculations into why your results may be turning out the way they are. Your discussion should consist of 2-4 paragraphs.

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 contributions. First, it should include the source code of your implementation **as the first part of the PDF file** (i.e., portions indicated by "Code").[1] Second, it should include a report with all results and analysis (i.e., portions indicated by "Write-up") **as the second part of the PDF file**. All material that you submit must be your own.

[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.