CPSC 585 - Artificial Neural Networks - Spring 2020

Project 3, due May 6

Last updated Wednesday, April 21, 12:25 am PDT

Platforms and Libraries

This project has the same platform and library requirements as Project 2, but CNNs are more computationally intensive than MLPs (minutes per epoch on a CPU rather than seconds), so access to a GPU or TPU is even more important than in the previous project.

Dataset

As with Project 2, this project uses the EMNIST Letters dataset. You may continue to use the code from the previous project, but the latest version of the TensorFlow Datasets project, released a few days ago, also includes the EMNIST dataset, so you may wish to use that version instead.

Validation Set

While using a validation set is a best practice when tuning hyperparameters, this was not explicitly required in the previous project. If you did not use a validation set in Project 2, you must do so for this project. The test set should only be used for final evaluation of your trained and tuned network.

Warmup

The Keras examples also include `mnist_cnn.py`, so you may wish to start there, in a manner similar to the previous project. Compare this architecture to the one Nielsen describes at the end of Convolutional neural networks in practice in Chapter 6.

A CNN for EMNIST Letters

Modify the code to apply the same convolutional network architecture to the EMNIST Letters dataset. How does the accuracy compare to its peformance on MNIST? How does the accuracy compare to the MLP you trained in Project 2?

Visualizing Training Progress

While the `fit()` method's optional `verbose` argument provides a progress bar and some metrics for each epoch, you may find it easier to visualize the training process by plotting a loss curve. The `fit()` method on a Keras `Model` returns a `History` object that can be used for this purpose, but it does not return until training is complete.

In order to avoid going down dead ends while adjusting your architecture and tuning its hyperparameters, you may prefer to visualize metrics during the process. TensorFlow includes the TensorBoard tool for this purpose.

Experimenting with EMNIST Letters

Now that you have a baseline convolutional network for comparison, begin experimenting with alternative architectures, optimizers, and hyperparameters for the EMNIST Letters dataset. How much can you improve the accuracy over Project 2?

Tips

- Be sure to train and evaluate with separate training and validation sets. The examples included with Keras aren't always careful about what they set the `validation_data` parameter to.
- Try viewing some of the misclassified letters with `matplotlib.pyplot.imshow()` or TensorBoard.
- Recall that EMNIST Letters merges upper- and lowercase letters into the same class, even when they look very different. How might you design a network that accounts for that difference?

When finished, evaluate your results on the test set. Compare the performance on the test set with your previous networks.

Submission

Submit your project by uploading your `.ipynb` file and any other relevant artifacts to the `project3/` subdirectory of the folder that will be shared with you on Dropbox. Do **not** submit copies of the dataset. You may also share a link to a version of your notebook hosted in the cloud, but you must download and submit the `.ipynb` file as well. A submission sheet will be provided in the same Dropbox folder.

You may work alone, or make a single submission for a team of 2-3 students. If you work in a team, only one submission is required, but for safety consider uploading copies to each team

member's submission folder. (Make certain, however, that the copies are the same in each case; the instructor will not attempt to ascertain which is the “real” submission.)

To finalize your submission, upload your files and fill out the sheet with the requested information by the end of class on the due date. Do not respond to the questions in the submission sheet using the Dropbox commenting feature; modify the document itself. Failure to follow any of these submission instructions exactly will incur a **10%** penalty on the project grade for all team members.