CECS 456: Machine Learning (Fall 2018)

Homework #5
Due 11/28/2018, 12:00 AM

For this assignment, we will work on handwritten image detection with Keras using MNIST data. This data set contains 70,000 images of handwritten digits in grayscale (0=black, 255 = white). The images are 28 pixels by 28 pixels for a total of 784 pixels. This is quite small by image standards. Also, the images are well centered and isolated. This assignment is designed to be able to be completed using only CPU, but if you have a GPU, you can also install Keras with GPU capabilities (e.g. CUDA for the backend).

Assignment Overview

Your goal for this assignment is to experiment with some different CNN architectures for image classification problems.

Please answer question in bold on this page in your readme.

Keras Documentation

Keras has quite helpful documentation. For this homework, you may want to look at the core layers (Dense: a dense fully-connected layer, ...), the convolutional layers (Conv2D, ...), the model class (which has methods such as evaluate, which evaluates the loss and summary metrics for the model, and predict, which generates test time predictions from the model), and the sequential class (which has some additional methods such as predict_classes, which generates class predictions), andloading and saving of models.

Training Times

On my laptop it took about ten to twenty minutes to train each model for a few epochs. However, slower computers may run training more slowly. So you may want to plan ahead especially if your computer is slow to avoid running into some training time bottleneck at the 11th hour.

I also highly recommend that while debugging your program, you make the program run more quickly by lowering the number of training samples/epochs / and/or saving your model after training (followed by re-loading it on future runs of the program). This will avoid you having to wait for a long time while training the model, only to discover that you have some minor typo later in your program (which triggers e.g. a run-time exception in Python).

MNIST garnished with a CNN (100 points)

Experiment with the MNIST dataset using different CNN classifiers. You can start with the mnist_cnn.py example in Keras.

Compare the following three architectures:

1. (25 points) a 3x3 convolution layer with 4 convolutions followed by a softmax,
2. (25 points) a 3x3 convolution with 32 convolutions followed by a softmax,
3. (25 points) a 3x3 convolution layer with 32 convolutions followed by a 2x2 max pool followed by softmax,

   Question: (25 points) What are the test accuracies and training times for the above three classifiers?

Submission

Submit your assignment in a zip file named yourname_HW5.zip onto Dropbox. Please include your source code and a readme with answers to the question above.