Project Part 3: Classification Using Neural Networks and Deep Learning

Due Dec 9, 2019 by 11:59pm **Points** 10 **Submitting** a file upload
Available until Dec 10, 2019 at 2:59am

This assignment was locked Dec 10, 2019 at 2:59am.

Project Overview:

In this part, we will revisit the Handwritten Digits Recognition task in Part 1, using a

convolutional neural network. The basic dataset is the same MNIST dataset from Part I,

but you may choose to use only a subset for training and testing, if speed performance with the entire dataset becomes a bottleneck. For example, you may use only 6000 samples for training (each digit with 600 samples) and 1000 samples for testing (each digit with 100 samples).

The basic requirement of this part is to experiment with a convolutional neural network with the following parameter settings:

(1) The input size is the size of the image (28x28).

(2) The first hidden layer is a convolutional layer, with 6 feature maps. The convolution

kernels are of 3x3 in size. Use stride 1 for convolution.

(3) The convolutional layer is followed by a max pooling layer. The pooling is 2x2 with stride 1.

(4) After max pooling, the layer is connected to the next convolutional layer, with 16 feature maps. The convolution kernels are of 3x3 in size. Use stride 1 for convolution.

(5) The second convolutional layer is followed by a max pooling layer. The pooling is 2x2 with stride 1.

(6) After max pooling, the layer is fully connected to the next hidden layer with 120 nodes and relu as the activation function.

(7) The fully connected layer is followed by another fully connected layer with 84 nodes and relu as the activation function, then connected to a softmax layer with 10 output nodes (corresponding to the 10 classes).

We will train such a network with the training set and then test it on the testing set.

You are required to plot the training error and the testing error as a function of the learning epochs. You are also required to change some of the hyper-parameters (the kernel size, the number of feature maps, etc), and then repeat the experiment and plot training and testing errors under the new setting.

These are the minimum requirements. Additional requirements may be added (like

experimenting with different kernel sizes, number of feature maps, ways of doing pooling, or even introducing drop-out in training, etc.).

Algorithm:

Convolutional Neural Network

Resources:

MNIST dataset, Google CoLab

Workspace:

Google CoLab (see file intro_to_colab.docx for more details)

Software:

Google CoLab

Language(s):

Python

Getting Started:

Read this document carefully, as well as additional files included (intro_to_colab.docx and **baseline.docx**). For more details about Colab, please go to **https://colab.research.google.com/notebooks/welcome.ipynb (https://colab.research.google.com/notebooks/welcome.ipynb)**

Required Tasks:

1. Read **intro_to_colab.docx** to get familiar with the platform.

2. Run the baseline code (**baseline.docx**) and report the accuracy.

3. Change the kernel size to 5*5, redo the experiment, plot the learning errors along with the epoch, and report the testing error and accuracy on the test set.

4. Change the number of the feature maps in the first and second convolutional layers, redo the experiment, plot the learning errors along with the epoch, and report the testing error and accuracy on the test set.

5. Submit a brief report summarizing the above results, along with your code.

Optional Tasks:

1. Change the kernel size to 9*9 and redo the experiment and report your results.

2. Use another optimizer (it can be chosen from **https://keras.io/optimizers/ (https://keras.io/optimizers/)**) and at least 3 different learning rate to redo the experiment, and report the accuracy vs learning rate on the test set.

3. Use average pooling and redo the experiment an0d report your results.

Optional tasks are to be explored on your own if you are interested and have extra time for them. No submission is required on the optional tasks. No grading will be done even if you submit any work on the optional tasks. No credit will be assigned to them even if you submit them. (So, please do not submit any work on optional tasks.)

What to Submit and Due Dates

1. Code: please add comment properly to explain what you do.
2. A report including the plots for the learning/testing errors and the final classification accuracy.

Code and Report are due at the end of Week 7.