CPSC 501 Assignment 2

Report

Part 2

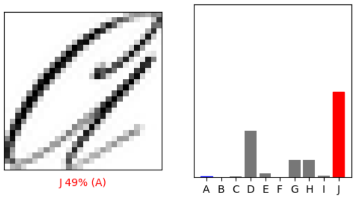
Partial Model

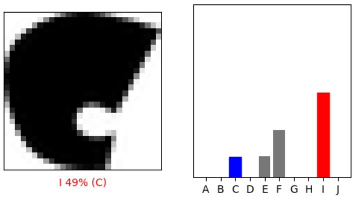
notMNIST-Partial uses a model identical to MNIST-Complete from part 1, having a hidden layer of 450 neurons with ReLU activation, Adam optimizer, and 10 epochs. As the input shape is the same, I felt no need to further edit the hyper-parameters or structures. The accuracy is lower than the accuracy in MNIST from part 1, with train/test accuracy of around 95%/93% compared to approximately 99%/98% in part 1.

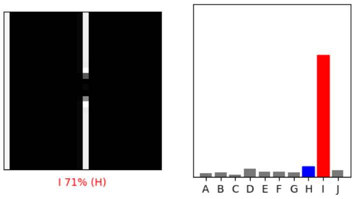
Modification on predict_test.py

In order to utilize predict_test, I modified 3 lines: line 14, line 28, and line 30. Line 14 is the line in which model was initialized as None, so I modified it to load whatever is in sys.argv[2] instead, which would load either MNIST.h5 or nonMNIST.h5 depending on the mode. Line 28 had an array of all 1/10s, and I used the model's predict function to get probability values from the img array, then flattened it to ensure I get an one-dimensional array. Finally, line 30, predicted_label, was modified to store the highest probability from the prediction array.

Three wrong images







The wrong images are found from the test data using predict_test, specifically index 1, 21, and 31, respectively. They are all "non-standard" characters, such as in a handwriting style or similar to a comic style.

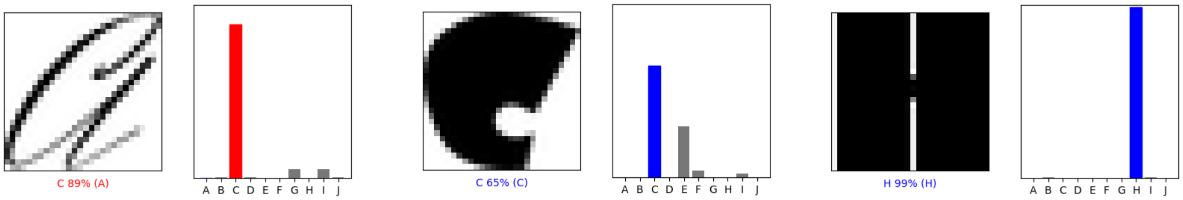
Changes on the Complete model

First of all, convolution neural network learned in the lecture was implemented. CNN detects and compresses certain patterns, which I thought would help with the test accuracy. The one shown in the class was implemented with a minor modification to deal with errors (instead of 25, 3, 3 on Convolution2D layer, it is changed to 25, 3), the train/test accuracy increased to around 98%/93.5%.

Afterwards, it was learned from https://keras.io/api/layers/convolution_layers/convolution2d/ that Convolution2D has activation function as an option, and ReLU activation function was implemented, which further increased accuracy up to around 99%/94.5%.

Finally, a dropout layer after the hidden layer copied from notMNIST-Partial was implemented to further increase accuracy. Various values were tested, and the dropout rate of 0.6 had the best result, with the train/test accuracy of around 95.5%/95%.

New results on the previously wrong images



The one in the handwriting style still gives a wrong answer, but it detects the character as C instead of J. The remaining two gives the correct answer with a high probability. I believe implementing CNN improved detection of features.