CS-512 Assignment 4: Report

Russi Sinha
Department of Computer Science
Illinois Institute of Technology

Abstract

This is a report for Assignment 4 for CS512. The task given at hand is to implement a custom CNN using MNIST dataset and measuring difference in performance by varying the parameters. The custom CNN then needs to be used to predict hand written digits.

1. Problem Statement

The program must read an image of a hand-written digit from an input file and must detect whether the digit is 'odd' or 'even' using the custom CNN implemented using the MNIST dataset.

2. Proposed Solution

For solving this problem statement a custom CNN needs to be trained and validated using a dataset and a prediction is to be done on an image from the user and detect whether the digit is even or odd.

3. Implementation Details

For the program to be executed, the following needs to be installed:

- Tensorflow
- Keras

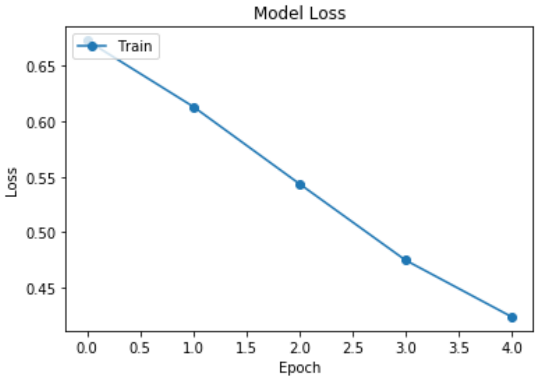
On execution of the file, the program loads a MNIST dataset and splits them into train and test data. A couple of convolutional layers are added to the model and a pooling layer is added to both the layers. The model is then compiled and trained using the train data and validated using the test data. The model is saved in a file for later use and need not be trained again. This model is then used to predict an image of a hand-written digit provided by the user. The console displays the value of the prediction whether the digit is even or odd.

The file 'cnn.py' needs to be executed first to train and validate the data and the model is saved in a file. For predicting an image, the file 'cnn_test.py' needs to be executed which loads the model which was saved before and generates a prediction.

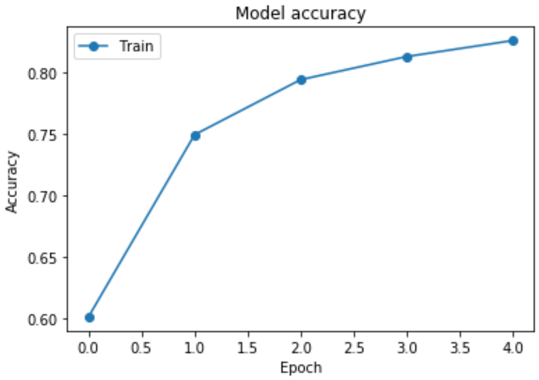
4. Results and Discussion

Deliverable 1:

Training Loss

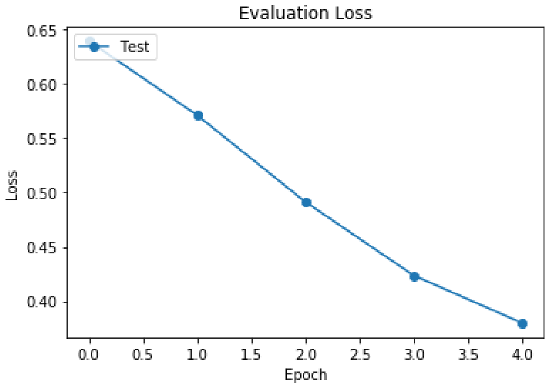


Training Accuracy

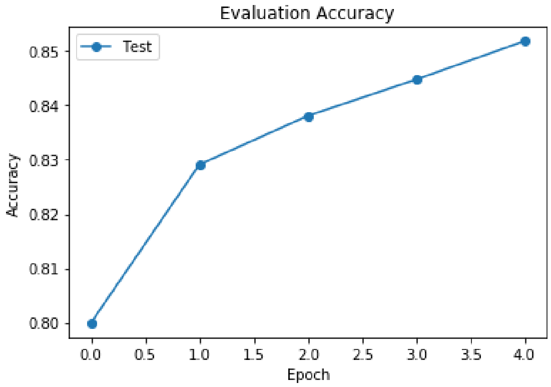


Final training step
Loss: 0.4386
Accuracy: 0.8143

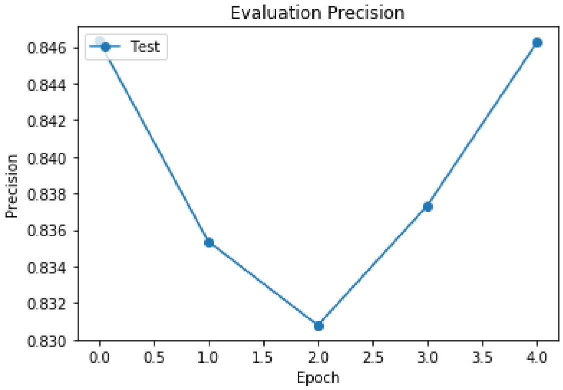
Evaluation Loss



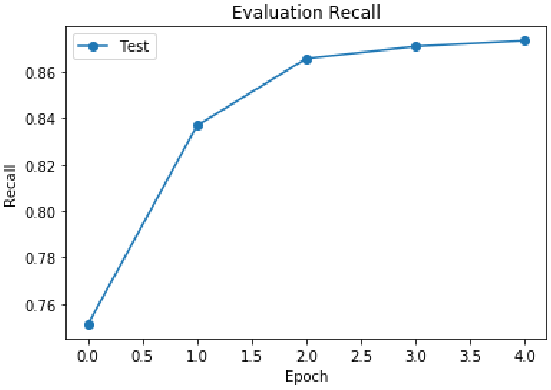
Evaluation Accuracy



Evaluation Precision



Evaluation Recall



Final evaluation step
Loss: 0.3637
Accuracy: 0.8526
Precision: 0.8513
Recall: 0.8675

Deliverable 2:

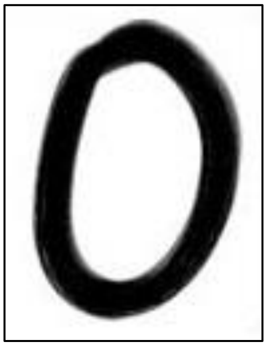
Change kernel size of convolutional filters to 7*7
Result: Loss metric decreases and accuracy increases
Change kernel size of convolutional filters to 9*9
Result: Loss metric further decreases and accuracy further increases
Conclusion: As we increase the kernel size of the convolutional filters, the accuracy of the classifier is better.

Change number of epochs to 10
Result: Loss metric decreases and accuracy increases
Change number of epochs to 15
Result: Loss metric further decreases and accuracy further increases
Conclusion: As we increase the number of epochs, the accuracy of the classifier is better.

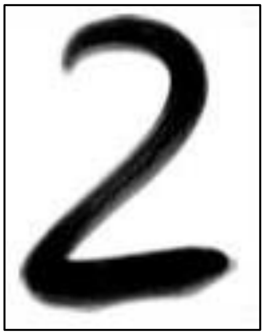
Change learning rate to 0.01
Result: Steep decrease in Loss metric and steep increase in accuracy
Change learning rate to 0.09
Result: Drastic decrease in Loss metric and drastic increase in accuracy
Conclusion: As we increase the learning rate, there is a tremendous increase in the accuracy of the classifier.

Deliverable 3:

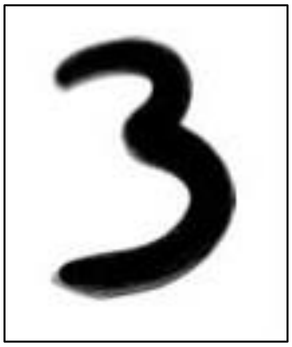
Below are few of the images whose results have been predicted using the above model.



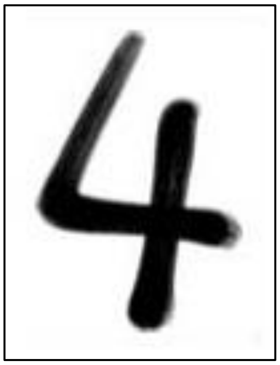
Result: Odd



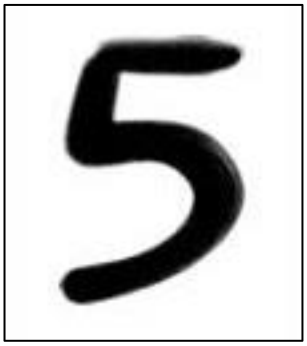
Result: Even



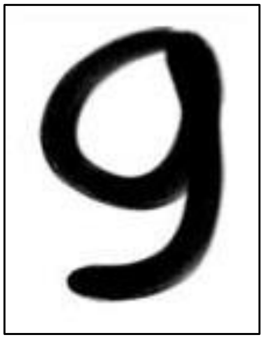
Result: Odd



Result: Even



Result: Even



Result: Odd

5. References

http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html
https://keras.io/
https://www.python36.com/mnist-handwritten-digits-classification-using-keras/
https://www.tensorflow.org/tutorials/