Ethan Daugherty

SNHU

CS-370-H5862 Current/Emerging Trends in CS

2-2 Assignment: Identifying Hand-written Digits

05/15/2020

**Identifying Hand-written Digits**

1. Once in the Apporto environment, create a new Jupyter Notebook and configure it using the following naming convention:
   * <YourLastName>\_<YourFirstName>\_Assignment1.ipynb

Graphical user interface, text, application, email

Description automatically generated

1. Read through the MNIST handwritten digits example on pages 16-24 of Deep Learning with Keras. Copy the code from pages 22-23 into your Jupyter Notebook.

Graphical user interface, text, application

Description automatically generated

1. Run the code in your Jupyter Notebook. Follow the examples in the book to establish an accuracy rate for the training, validation, and test data sets with two hidden layers. Table

   Description automatically generated
2. The remainder of the chapter provides examples of how to modify different parameters within the code (number of hidden layers, hidden neurons, BATCH\_SIZE, number of epochs, and so on). Pick one parameter and run two or three different experiments, modifying the parameter values to establish accuracy scores with different parameter values. Make sure that the experiments result in significant changes in accuracy rates. Be sure to place each experiment in a different code block so that your instructor can view all of your changes.

Background pattern

Description automatically generated with low confidence

Graphical user interface, text, application, email

Description automatically generated

1. Create a Markdown cell in your Jupyter Notebook after your code and its outputs. In this cell, explain the changes in accuracy rates by comparing and contrasting your results from Steps 3 and 4. What happens to the accuracy rates for the training, validation, and test data sets as you change the parameters? Why?

## Analysis:

The \*\*batch\_size\*\* has been changed for all three models. For \*\*model1\*\* the \*\*batch\_size\*\* is \*\*16\*\*.

For \*\*model2\*\* has a \*\*batch\_size\*\* of \*\*32\*\* and \*\*model3\*\* has a batch size of \*\*256\*\*

The test accuracy for all three models differs only slightly:

\*\*model1\*\*Test accuracy: \*\*0.9761999845504761\*\*

\*\*model2\*\* Test accuracy: \*\*0.9693999886512756\*\*

\*\*model3\*\* Test accuracy: \*\*0.9275000095367432\*\*

\*\*model1\*\* Provided the best results compared to other two test models. Though all three \*\*model1\*\*, \*\*model2\*\*, and \*\*model3\*\* did not provide statistically any significant change.

While changing the data within \*\*model2\*\* and \*\*model3\*\* the accuracy fell only slightly but was still a drop in accuracy. By increasing the batch size you increase the generalization error within the test, this is why \*\*model3\*\* is lower in test accuracy than \*\*model1\*\* or \*\*model2\*\*. In these tests the smaller batch sizes work best.