**CSC 4760 – Homework 6**

**Objectives:**

1. Design a feed forward neural network for classifying handwritten digits.
2. Show the network architecture of the design.
3. Show the code of the design.
4. Show the training and testing information.
5. Show the test results.

**Results:**A diagram of a network

Description automatically generated

Figure 1 – Architecture Diagram depicting how the neural network is structured. The number of neurons per layer is dependent on the variable input size determined by the user. It should be noted however that the input layer MUST be the size of the tensor created for each image. This tensor is roughly the size of total number of pixels in the image.

A screenshot of a computer program

Description automatically generated

Figure 2 – NN Code lines 1-33. This section of code mostly involves importing the libraries and pointing the code towards running in the correct location. The training size and learning rate for the neural network is also established, along with the base parameters that determine the size of the neural network as well.

A screenshot of a computer code

Description automatically generated

Figure 3 – NN code lines 34-57. This section of code downloads external datasets to the user’s machine. It also transforms members of the datasets into tensors so that the linear regression model of the NN can handle it for internal calculations. This section SHOULD have been pointed to training models for handwriting recognition, however it is NOT. This was due to time limitations on my part after caring for a friend all week, and attending a hackathon the weekend prior (I built a neural network using basic linear regression for heartbeat using my project idea, and won best interactive media! I have no clue what to do with the gaming monitor I won).

A screenshot of a computer program

Description automatically generated

Figure 4 – NN Code lines 60-71. This is the actual core of the neural network, where the three layers are defined. The first layer consists of a linear regression model between the total numer of neurons on the input layer, and the hidden layer. Any model that lacks any relation (returns zero or less) is then returned as zero by the relu function. The next layer is then created as the linear regression between the total number of neurons in the hidden layer, and the number of defined classes (in this case 10, for 10 digits). Using simple linear regression like this even with a large number of neurons seems fairly inefficient, I cannot help but wonder if it would have been more efficient for me to using a convolution between the input layer and the second layer to get stronger analysis. However, when in doubt, simply increase the number of neurons and hope for the best.

A screenshot of a computer program

Description automatically generated

Figure 5 – NN lines 80-102. This is the training section of the neural network, the number of times the neural network is trained is based on the number of epochs defined by the user, for each epoch, the neural network combs through ever image label classification presented in the training data. A forward pass is performed through the neural network, generating expected classifications for the training data, the loss is then determined based on the results compared to the training data. Optimization of weights for neurons is then created based on the loss.  
  
Just noticed that the images are reshaped to the device for every pass – this seems wasteful, we could probably speed up the code if we defined the images as a variable outside of the training loop. This runs fairly fast currently, but I imagine with larger training sets it could make a huge difference.

A screenshot of a computer program

Description automatically generated

Figure 6 – NN lines 103 to 119. This is the testing section of the model. Images are loaded from the testing data, outputs are generated based on the model, and the accuracy of the outputs is compared testing data labels.

A screenshot of a computer

Description automatically generated

Figure 7 – Testing results. It should be noted that this is NOT to my knowledge applied to handwritten digit classification, but only for the image classification presented in the model. However, googling the exact images downloaded I vaguely suspect they are.