Group 07

23100035

In phase 3 of the Neural Network assignment, we primarily tested the accuracy of our neural network by testing it with a completely new image set, separate from the dataset that it was first trained and then tested on.

We had developed the code that could take user defined hidden nodes and learning rate to train on the data. For this phase as well, we chose 3 hidden layers with 128 nodes each. We kept the number of nodes in each layer the same as to minimize the loss of any information, large jumps in number of nodes between layers can cause loss of information.

We trained the model on the entire data set for a learning rate of 0.009 as to not over train it and ran it for 30 epochs. We chose 30 epochs as it was giving us a good accuracy of 80 + and was also completing in less amount of time.

We then loaded our own dataset of 50 images, with 5 images for each member of the dataset into the notebook. We read the images using the cv2 library with the greyscale flag, and append them into my_X_test list. Similarly, my_Y_test list contains all the one-hot encodings for each image. These are done within one loop, as the images are loaded into the list.

In the preprocessing phase, we primarily do two things. Firstly, we must standardize the images dimensions. Since our Neural Network takes in 28 by 28 dimensional images, we must explicitly resize the image to follow these dimensions, or directly resize them to a matrix of 1 by 784, which is the true input size of the NN. We have directly resized them to 1x784.

We have also used INTER_CUBIC interpolation, which interpolates over a 4x4 pixel neighborhood in order to fill in gaps that might be there (in case image less then 28 by 28). Secondly, we convert the image array into a numpy array and the normalize by dividing the entire array by 255.

The accuracy of our own dataset came out to be 8% which is quite low, but this could be due to a variety of reasons. We list them below:

1. The original images were not of high quality, and pixel boundaries were blurry/not defined well.
2. Image quality was lost when converting from normal RBG to greyscale, translation could have resulted in data loss, and hence messier boundaries between clothing boundry and background
3. All the images were not of the right dimensions, as you can see in pre-processing, we are explicitly reshaping the image to a 1 by 784 array. The interpolation copies neighboring 4x4 pixels to fill in the empty space (extrapolates) them. This could cause further errors in the data set
4. Images could be too zoomed in (hence some portion cropped out) or too zoomed out.
5. Our model itself has been trained on a very strict and uniform data set, as well as tested on a uniform data set as well. In pavlovian terms, one could say our model has been conditioned to only accurately judge images of specific idealistic properties that have been provided by the training dataset.

Improvement ideas:

- Training of dataset could consist of many different images from different sources, where we preprocess them ourselves and not have them preprocessed out of the box.
- Following a more standardized way of collecting images to run through our Neural Network