**Logistic regression on a replacement MNIST dataset Report**

In order to get the predict.ipynb file to work I added a line in main() in order to load the model:

model = tf.keras.models.load\_model(sys.argv[2])

In predict() I added lines to reshape the image so input is compatible for my convolutional layers

    img\_reshaped = img.reshape(img.shape[0], 28, 28, 1)

    img\_reshaped = img\_reshaped.astype('float32')

Following those lines I added two lines to get my array of percent confidences in each class for my image and the index of the highest prediction in the previous array.

    prediction = model.predict(img\_reshaped).tolist()[0]

    predicted\_label = np.argmax(prediction)

The image that I created (representing the letter “A”) for this part was:



My initial model predicted this letter to be “A” at 1.2 percent.

I attempted several changes to increase the prediction accuracy of the letter, however the first big increase in accuracy occurred when I increased the initial filter of my convolutional layer to 32 from 28 and specified the strides at 1, increasing the accuracy to 31%.

Following this I used the keras ImageDataGenerator class to train with data with letters shifted and rotated. This would eventually increase my accuracy to 97%.

My initial model had an accuracy of 94% with a 1.2% prediction of my drawn letter. My final model had an accuracy of 95.5% with a 97% prediction of my drawn letter.

In order to run the code for the model, simply upload the notMNIST.ipynb as a new notebook in google colab. Run each cell, uploading the notMNIST.npz when prompted.

In order to run the code for the predict.ipynb, upload the file as a new notebook in google colab. Run each cell, uploading the saved notMNIST.h5 model, the notMNIST.npz, and the image.png when prompted. Enter “notMNIST” when prompted for dataset, “notMNIST.h5” when prompted for model, and “image.png” when prompted for image.