Report

Data Preparation, I split my data into a training section and testing section, each of them is roughly half of the dataset. Beside that there are some NULL values in the dataset, so I basically replace them with some values. Google Colab is very helpful when reading the csv file since it skips the first row which is just name for attribute.

Network configuration, I setup my network with 4 layers which is one input layer, one output layer, and two hidden layers. There are thirteen neurons which represent the thirteen features in the input layer, eights neuron in each hidden layer, and two neurons in the output layer.

Validation strategy, I did normalize the training dataset and it roughly improve the percentage by one percent. It is not much but I am sure that the quality of accuracy is mainly affected by my choice of loss function. Beside that I uses SoftMax function for the activation function for the output layer. I try to lower the risk of overtraining by shuffle the training dataset before put the model into training.

Results, after gradually adding more layers into the model, I found that the sweat point is two layers and no less than three neurons per layers since they seem to perform worsts, around twenty to thirty percent accuracy. The learning rate seems to work well between 0.1 and 0.3, larger learning rates does not seem to add any more benefit.

References

“Load CSV Data  :   Tensorflow Core.” *TensorFlow*, https://www.tensorflow.org/tutorials/load\_data/csv.

“Tf.keras.layers.normalization  :   Tensorflow V2.11.0.” *TensorFlow*, https://www.tensorflow.org/api\_docs/python/tf/keras/layers/Normalization.

Wwsalmon. “Simple Mnist NN from Scratch (Numpy, No TF/Keras).” *Kaggle*, Kaggle, 24 Nov. 2020, https://www.kaggle.com/code/wwsalmon/simple-mnist-nn-from-scratch-numpy-no-tf-keras/notebook.

“How to Build a Neural Network with Tensorflow and Keras in 10 Minutes.” *YouTube*, YouTube, 5 Nov. 2019, https://www.youtube.com/watch?v=\_c\_x8A3mNDk.

Link to Google Colaboratory

<https://colab.research.google.com/drive/1PKFXOuE4lL2W-WcZzwwll6vWeWvV6qbl?usp=sharing>

Output

Estimated Classification

* Typical Run with non-normalize data

Epoch 73/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2353 - accuracy: 0.4667

Epoch 74/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2351 - accuracy: 0.5267

Epoch 75/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2348 - accuracy: 0.5467

Epoch 76/100

5/5 [==============================] - 0s 3ms/step - loss: 2.2346 - accuracy: 0.5533

Epoch 77/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2346 - accuracy: 0.5533

Epoch 78/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2342 - accuracy: 0.5467

Epoch 79/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2342 - accuracy: 0.5133

Epoch 80/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2340 - accuracy: 0.4533

Epoch 81/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2339 - accuracy: 0.4867

Epoch 82/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2338 - accuracy: 0.4867

Epoch 83/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2336 - accuracy: 0.4733

Epoch 84/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2335 - accuracy: 0.5067

Epoch 85/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2336 - accuracy: 0.5200

Epoch 86/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2334 - accuracy: 0.4800

Epoch 87/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2334 - accuracy: 0.4067

Epoch 88/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2331 - accuracy: 0.3667

Epoch 89/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2331 - accuracy: 0.3800

Epoch 90/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2331 - accuracy: 0.4067

Epoch 91/100

5/5 [==============================] - 0s 4ms/step - loss: 2.2330 - accuracy: 0.3867

Epoch 92/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2330 - accuracy: 0.4333

Epoch 93/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2330 - accuracy: 0.4600

Epoch 94/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2330 - accuracy: 0.4467

Epoch 95/100

5/5 [==============================] - 0s 3ms/step - loss: 2.2331 - accuracy: 0.4400

Epoch 96/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2330 - accuracy: 0.4267

Epoch 97/100

5/5 [==============================] - 0s 3ms/step - loss: 2.2331 - accuracy: 0.4333

Epoch 98/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2330 - accuracy: 0.3867

Epoch 99/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2331 - accuracy: 0.4333

Epoch 100/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2330 - accuracy: 0.4333

* Typical Run with normalize data
* Epoch 73/100
* 5/5 [==============================] - 0s 2ms/step - loss: 4.5321 - accuracy: 0.3600
* Epoch 74/100
* 5/5 [==============================] - 0s 2ms/step - loss: 3.7160 - accuracy: 0.3667
* Epoch 75/100
* 5/5 [==============================] - 0s 2ms/step - loss: 5.1983 - accuracy: 0.3600
* Epoch 76/100
* 5/5 [==============================] - 0s 2ms/step - loss: 4.3839 - accuracy: 0.4000
* Epoch 77/100
* 5/5 [==============================] - 0s 2ms/step - loss: 3.8471 - accuracy: 0.3467
* Epoch 78/100
* 5/5 [==============================] - 0s 2ms/step - loss: 3.7381 - accuracy: 0.4333
* Epoch 79/100
* 5/5 [==============================] - 0s 2ms/step - loss: 3.9244 - accuracy: 0.3867
* Epoch 80/100
* 5/5 [==============================] - 0s 2ms/step - loss: 3.8047 - accuracy: 0.3267
* Epoch 81/100
* 5/5 [==============================] - 0s 2ms/step - loss: 4.3675 - accuracy: 0.3733
* Epoch 82/100
* 5/5 [==============================] - 0s 2ms/step - loss: 3.3166 - accuracy: 0.3867
* Epoch 83/100
* 5/5 [==============================] - 0s 2ms/step - loss: 3.0577 - accuracy: 0.3400
* Epoch 84/100
* 5/5 [==============================] - 0s 2ms/step - loss: 3.6613 - accuracy: 0.3400
* Epoch 85/100
* 5/5 [==============================] - 0s 2ms/step - loss: 2.9179 - accuracy: 0.3933
* Epoch 86/100
* 5/5 [==============================] - 0s 2ms/step - loss: 4.0198 - accuracy: 0.3200
* Epoch 87/100
* 5/5 [==============================] - 0s 2ms/step - loss: 3.3429 - accuracy: 0.4133
* Epoch 88/100
* 5/5 [==============================] - 0s 2ms/step - loss: 3.0470 - accuracy: 0.4333
* Epoch 89/100
* 5/5 [==============================] - 0s 2ms/step - loss: 2.8820 - accuracy: 0.3400
* Epoch 90/100
* 5/5 [==============================] - 0s 2ms/step - loss: 2.8711 - accuracy: 0.3933
* Epoch 91/100
* 5/5 [==============================] - 0s 3ms/step - loss: 3.0686 - accuracy: 0.3467
* Epoch 92/100
* 5/5 [==============================] - 0s 2ms/step - loss: 2.7717 - accuracy: 0.3867
* Epoch 93/100
* 5/5 [==============================] - 0s 4ms/step - loss: 2.5458 - accuracy: 0.4333
* Epoch 94/100
* 5/5 [==============================] - 0s 3ms/step - loss: 2.8460 - accuracy: 0.3333
* Epoch 95/100
* 5/5 [==============================] - 0s 4ms/step - loss: 2.6966 - accuracy: 0.3933
* Epoch 96/100
* 5/5 [==============================] - 0s 2ms/step - loss: 2.6937 - accuracy: 0.3533
* Epoch 97/100
* 5/5 [==============================] - 0s 2ms/step - loss: 2.6391 - accuracy: 0.4333
* Epoch 98/100

5/5 [==============================] - 0s 2ms/step - loss: 2.8253 - accuracy: 0.3800

Epoch 99/100

5/5 [==============================] - 0s 2ms/step - loss: 2.2223 - accuracy: 0.3600

Epoch 100/100

5/5 [==============================] - 0s 2ms/step - loss: 2.5245 - accuracy: 0.3933

* Run with validation set

5/5 [==============================] - 0s 2ms/step - loss: 1.4647 - accuracy: 0.4967