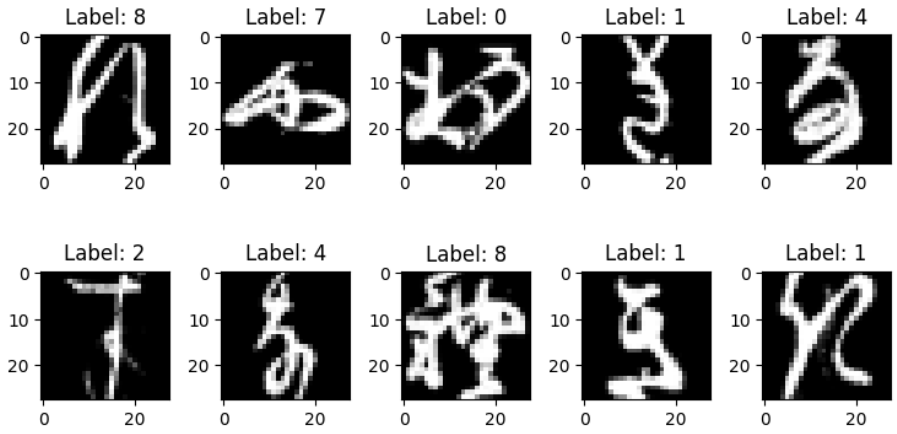
Challenge 3

Kevin Cazzolato

April 2024

1 Data Exploration

In this section, we aim to explore the dataset without rigorous statistical analysis; hence, we will assume that there are no anomalies, leverage points, missing values, or null values (which indeed is the case). After importing the dataset and splitting it into training and test sets, we observe that the dataset is composed of Japanese characters, resulting in 10 labels, as shown in the summary image below.



Additionally, we note that there are 32 images, in one color, with a resolution of $28x28$.

2 Model Creation

2.1 FCNN

We aim to create a fully connected neural network model where each neuron is connected to every neuron in the next layer. Thus, the first layer of neurons (input layer) receives the "flattened" image (in one dimension), resulting in $28x28 = 784$ neurons for the first layer. There will be 3 hidden layers with 256, 128, and 64 neurons, respectively. The final layer will comprise the labels to predict, thus 10. Using

$$f(x) = W_4 \cdot \sigma(W_3\sigma(W_2\sigma(W_1x + b_1) + b_2) + b_3)$$

with $\sigma(x) = \max(0, x)$, the ReLU function was chosen as the activation function, while the loss function is cross-entropy, which uses the entropy function

$$H(p) = -\sum_{i\in\mathbb{X}} p(i)\log p(i)$$

to more significantly penalize incorrect predictions with high model confidence. This choice was entirely arbitrary, as were the choices of epoch number, learning rate in GD, and optimizer usage.

2.2 CNN

We intend to implement the CNN algorithm, specifically composed of two convolutional layers followed by max pooling to reduce the output dimensions. Subsequently, data is passed through two fully connected layers for final classification (in the last layer). The activation function used is ReLU, as previously applied. The hyperparameters are as follows: the loss function is cross-entropy, the Adam algorithm with a learning rate of 0.001 is used, the number of epochs is 10, padding is set to 1, and kernel size to 3.

3 Results Analysis

From the models listed previously, the following results were obtained:

Model	Epoch	Loss
Fully Connected Neural Network	1	0.407937988901635
	2	0.17004608266701302
	3	0.10913533537338177
	4	0.08026481196895863
	5	0.06063051574470786
	6	0.05046937800451803
	7	0.039479444689747956
	8	0.03444179630789828
	9	0.02749231288600919
	10	0.025984076039476712
CNN	1	0.2481
	2	0.0746
	3	0.0435
	4	0.0292
	5	0.0185
	6	0.0143
	7	0.0116
	8	0.0087
	9	0.0096
	10	0.0051

Accuracy on test data:

- Fully Connected Neural Network: 0.9108
- CNN: 0.9514

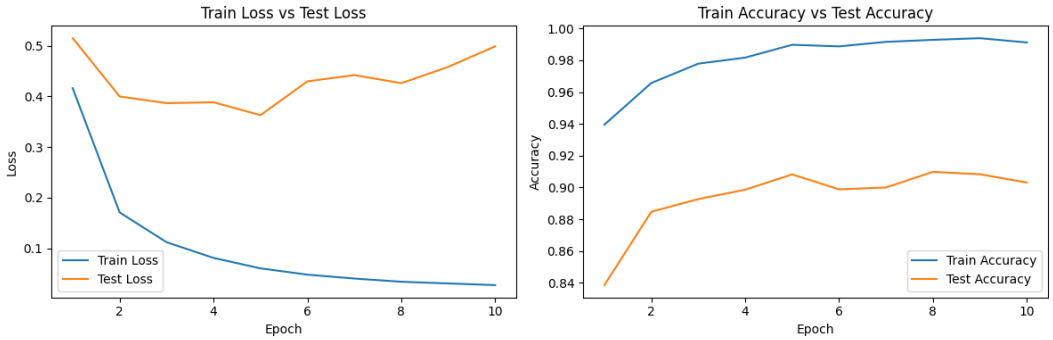
Accuracy on training data:

- FCNN: 0.9937
- CNN: 0.9979

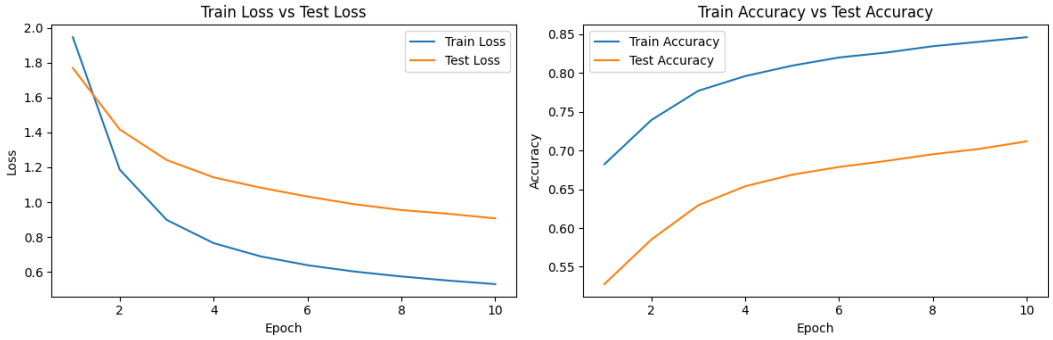
We can observe that the loss values decrease, as expected, in both models, with a faster decay in the CNN loss, suggesting it performs better. This is also evident from the test data accuracy values of the two models; using the CNN algorithm yields a higher accuracy. This is likely due to the fact that this algorithm operates locally, using convolutional layers to search for patterns, which is very suitable for images, as it is expected that two identical objects' images contain many common patterns. On the other hand, FCNN is also a good model but slower in reaching the minimum Loss (it is not possible to determine if it is a global or relative minimum). We can thus conclude that CNN is the most suitable model for this dataset.

4 Plot

In this final section, the Loss and accuracy for test and training data relative to the FCNN algorithm are plotted. Initially, for a learning rate of 0.001:



We can observe that the Loss for the training set decreases rather quickly, even though the loss for the test set does not show the same trend. The oscillatory trend of the test set loss could indicate that the learning rate is too high, preventing the loss function from reaching a local/global minimum, or the model is overfitting. From the accuracy graph, we can exclude the second hypothesis since the test accuracy appears to stabilize at 90%, an excellent value. Therefore, we aim to decrease the learning rate to approach the local/global minimum with greater regularity.



From this last graph, with a learning rate of 0.00001, we can see that the previously encountered issue is no longer present, as there is more consistency with what is expected of an optimal loss. We also note that we are approaching the local/global minimum more slowly, suggesting that the number of epochs should be increased to get closer to the minimum. Finally, from the accuracy graph, we observe that the two accuracies exhibit a monotonic increasing trend, which is optimal, indicating that more computational power would be needed to compute more epochs with this learning rate or alternatively, slightly increase it.