COMP576 Assignment 2

Rocky Wang(rw62)

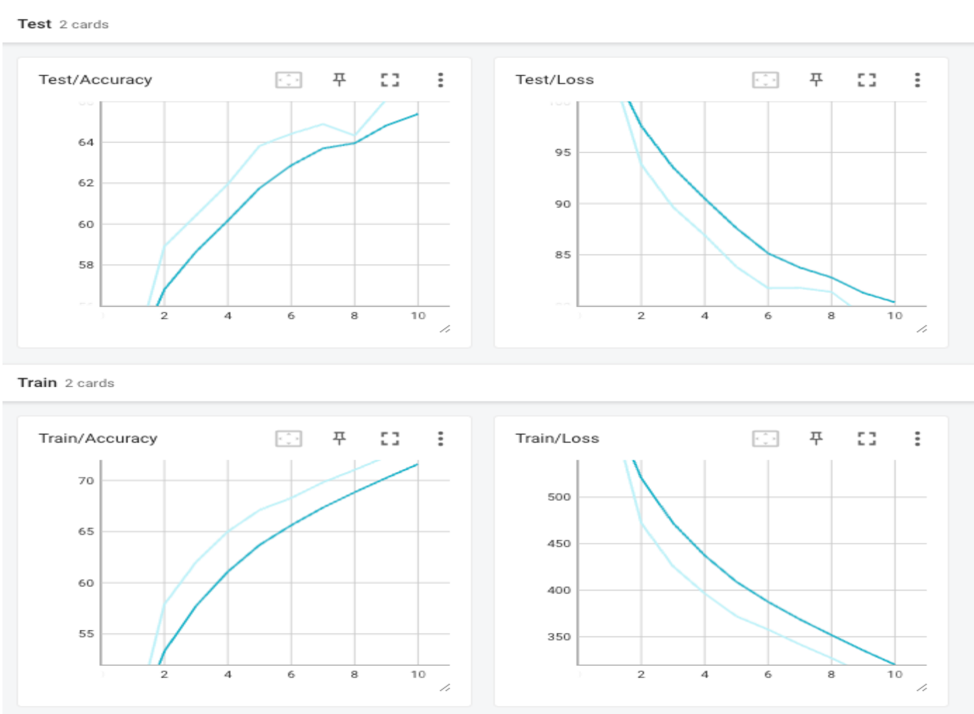
1b

I have trained the model using different sets of hyper parameters:

- Set 1: Learning rate: 0.001. Train method: Adam Optimizer
- Set 2: Learning rate: 0.001. Train method: Stochastic Gradient Descent Optimizer
- Set 3: Learning rate: 0.01. Train method: Adam Optimizer

Based on the results of train/test accuracy and loss, Set 1 has the best test accuracy of 66.23%. Adam optimizer seems to be the better optimizer then Stochastic Gradient Descent Optimizer to be used when training the CNN model. It might because Adam uses adaptive learning rates for each parameter, which helps it converge faster and more reliably compared to the fixed learning rate of SGD.

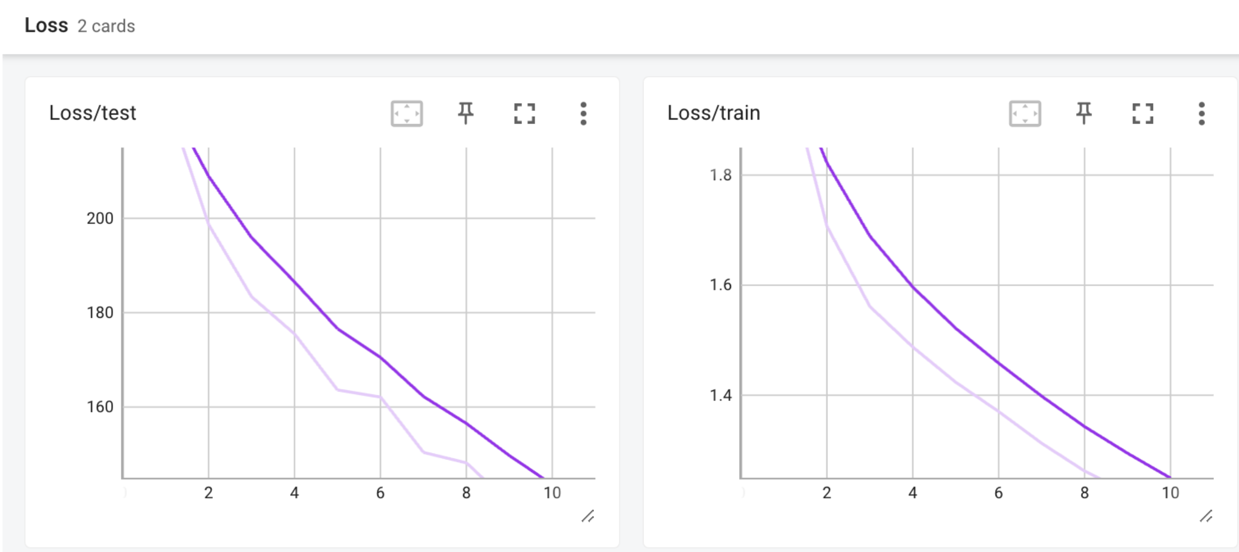
Trained with given hyper parameters.
Learning rate: 0.001
Train method: Adam Optimizer
Test accuracy: 66.23%



Learning rate: 0.001

Train method: Stochastic Gradient Descent Optimizer

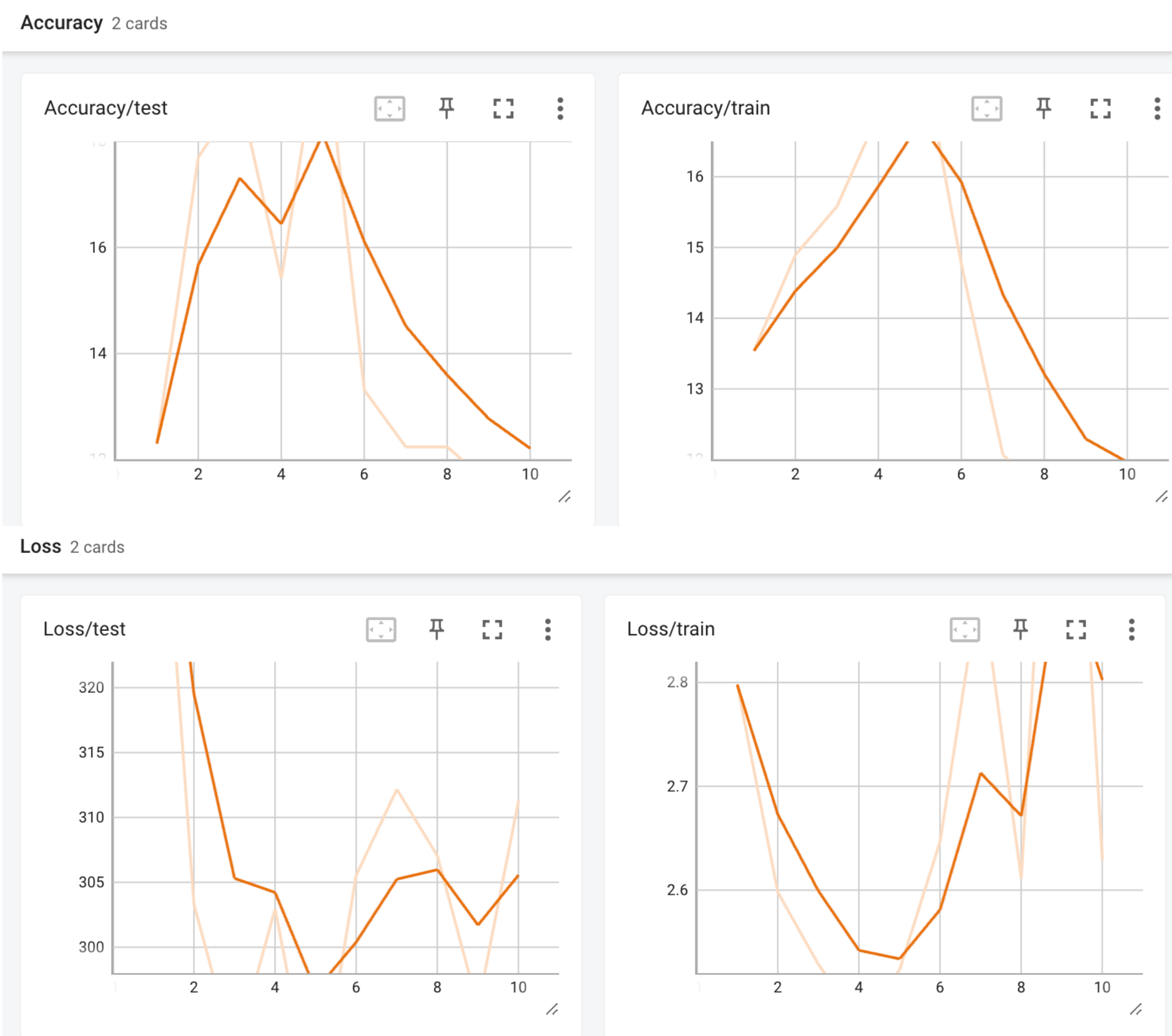
Test accuracy: 62.87%



Learning rate: 0.01

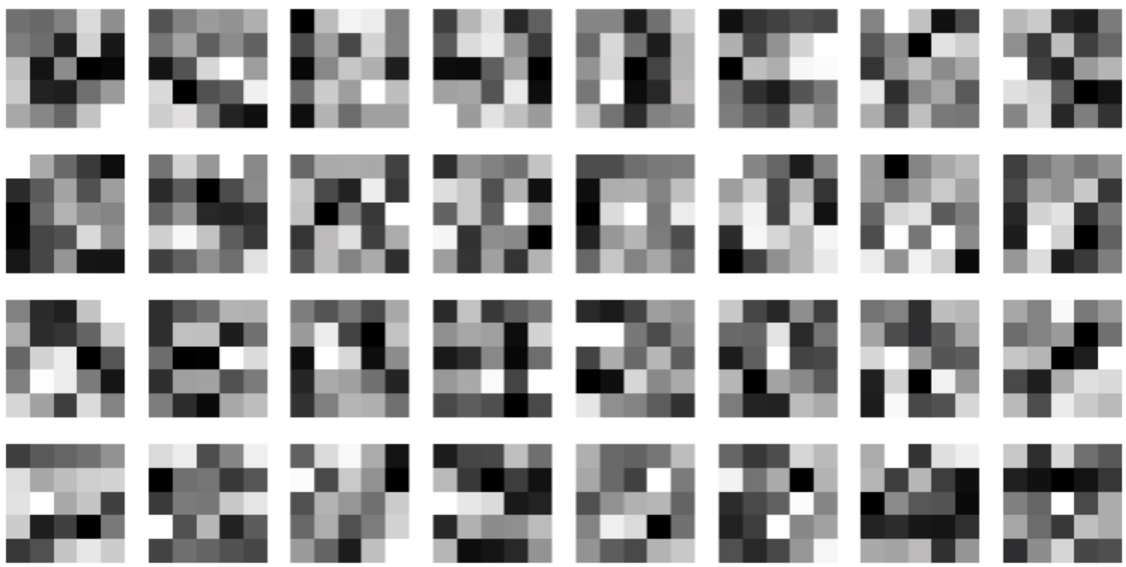
Train method: Adam Optimizer

Test accuracy: 11.38%



1C

Here is the visualization of the first convolutional layer's weights.



```
Conv Layer 1:
Mean activation: -0.07
Std activation: 0.18
Max activation: 1.68
Min activation: -1.77
---------------------------------
Conv Layer 2:
Mean activation: -0.60
Std activation: 0.68
Max activation: 2.11
Min activation: -9.35
---------------------------------
```

2

The paper talks about the large Convolutional Network (ConvNet) models. It particularly addresses the existing gap in understanding why these models are so effective and how they can be further enhanced.

The paper introduces a unique visualization method. This technique offers insights into the functions of intermediate feature layers and the operations of the classifier. By leveraging this visualization, researchers and practitioners can gain a deeper understanding of the inner workings of ConvNets.

Another significant achievement of the authors is the design of improved model architectures. By utilizing their visualization technique, they identified and crafted models that surpassed previous results on the ImageNet classification benchmark. This showcases the practical implications and benefits of their proposed method.

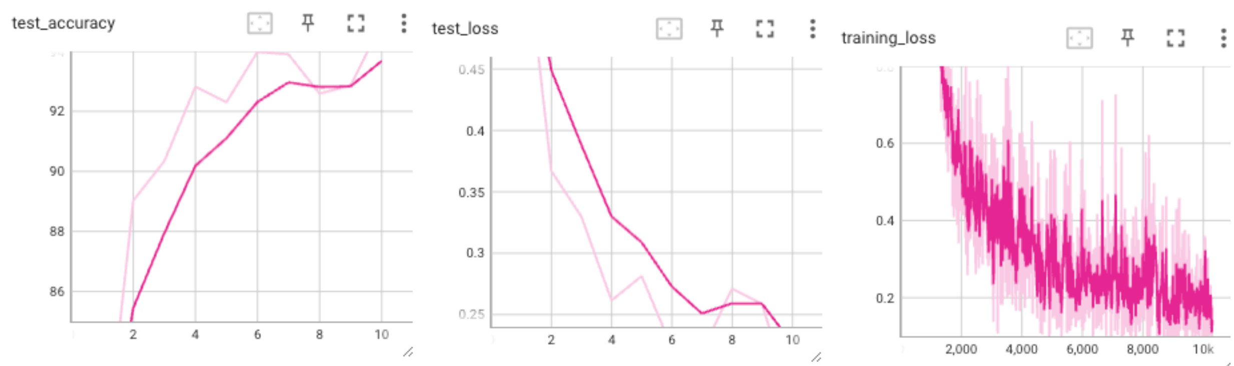
The paper also emphasizes the versatility of their ImageNet model. They demonstrate that it's not limited to just one dataset but can generalize effectively to other datasets. This generalization capability is crucial for the broader application of ConvNets in various domains.

Most importantly, the authors introduces the Deconvolutional Network (deconvnet) approach. This method maps feature activities in reverse, tracing them back to the input pixel space. By doing so, it provides a clearer understanding of what features the ConvNet detects and how they relate to the original input.

3a

Learning_rate = 0.001 Adam Optimizer. Hidden_layer_units = 64 Epochs = 10

Test Accuracy = 94.93%

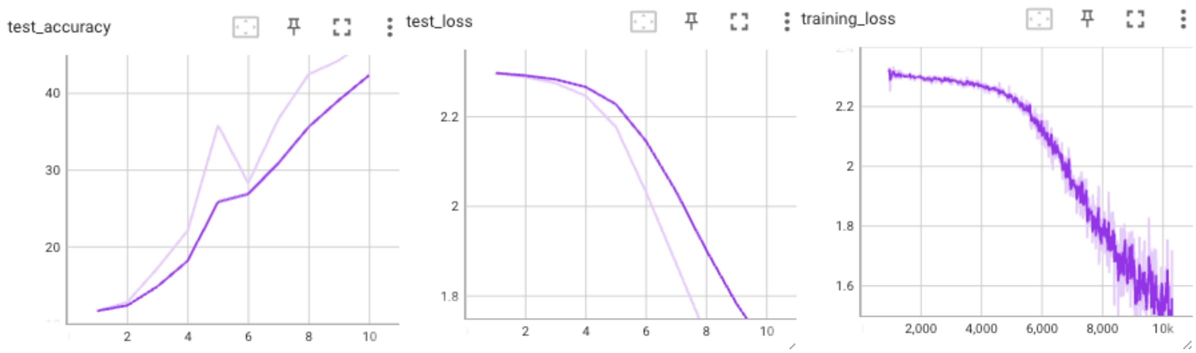


Learning_rate = 0.01 Adam Optimizer. Hidden_layer_units = 64 Epochs = 10

Test Accuracy = 22.89%

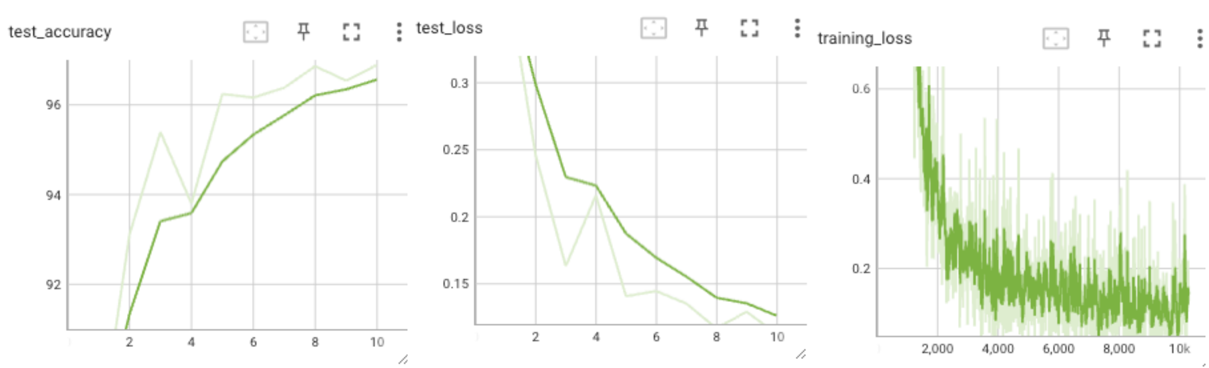
Learning_rate = 0.001 SGD Optimizer. Hidden_layer_units = 64 Epochs = 10

Test Accuracy = 42.37%



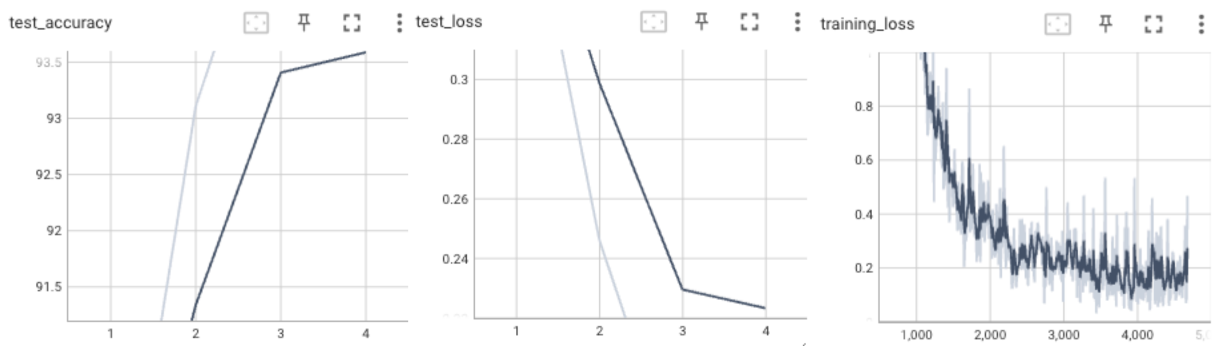
Learning_rate = 0.001 Adam Optimizer. Hidden_layer_units = 128 Epochs = 10

Test Accuracy = 96.89%



Learning_rate = 0.001 Adam Optimizer. Hidden_layer_units = 128 Epochs = 4

Test Accuracy = 93.8%



1. When learning rate is increased from 0.001 to 0.01, the test accuracy decreased significantly. The drop in test accuracy is mainly caused by the high learning rate (0.01). It makes the model overfitting the training data, and thus result in low test accuracy

2. When the hidden layer units increase from 64 to 128, the test accuracy slightly increased. The higher number of hidden layer units increase the capacity of the model to better fit the complex dataset. Thus, it can extract more useful features. However, if the hidden layer units increase too much, there is the possibility of overfitting.

3. I tried using 2 different optimizers (Adam and SGD) for training the model. Adam optimizer results in higher test accuracy. It might be because of Adam's adaptive learning rate characteristic. Adam optimizer can adjust the learning rate so that the model can converge faster on dataset like MNIST.

4. When I decrease the number of iterations, the model results in lower test accuracy. Based on the graph, I can see that the test accuracy still has the trend of going up even at the end of iteration. It is clearly that the model needs more iteration in order to reach its best test accuracy performance.

3b

I have tested LSTM and GRU models using different sets of hyper parameters. For LSTM models, the model with 0.001 learning rate and 128 hidden units perform the best. For GRU models, the model with 0.01 learning rate, 128 hidden units and 2 hidden layers perform the best. All in all, both of the model can reach a test accuracy that is higher than 99% on MNIST dataset when the hyper parameters are monitored. Both of the models can perform better then RNN model.

LSTM:

Learning_rate = 0.01

Num_layers = 1

Hidden_size = 64

Test accuracy after 10 epochs = 96%

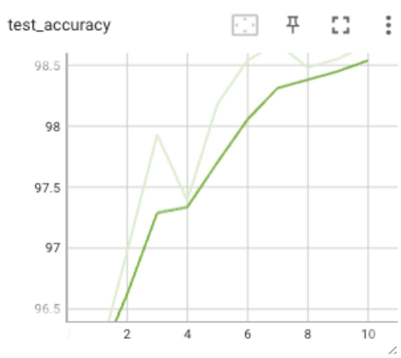
LSTM:

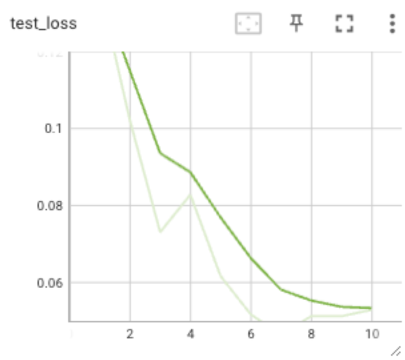
Learning_rate = 0.001

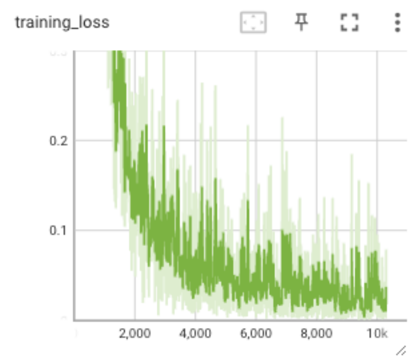
Num_layers = 1

Hidden_size = 128

Test accuracy after 10 epochs = 99%







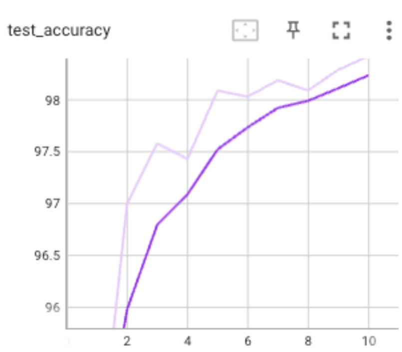
GRU:

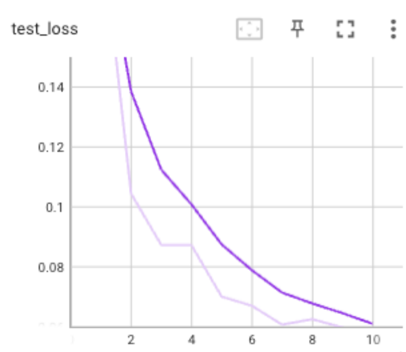
Learning_rate = 0.001

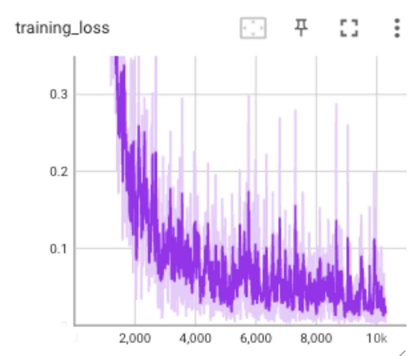
Num_layers = 1

Hidden_size = 64

Test accuracy after 10 epochs = 98.42%







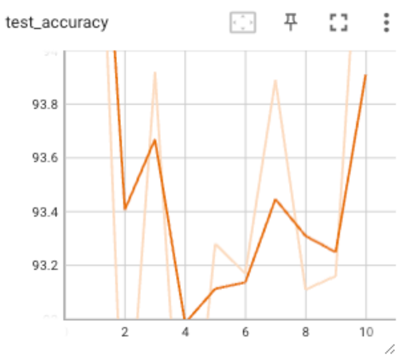
GRU:

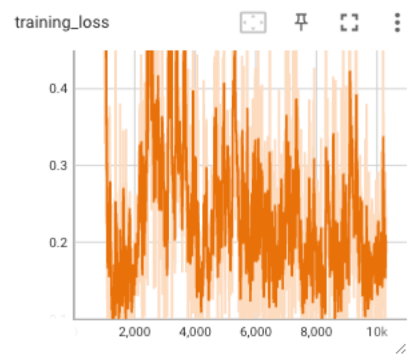
Learning_rate = 0.01

Num_layers = 1

Hidden_size = 128

Test accuracy after 10 epochs = 94.89%





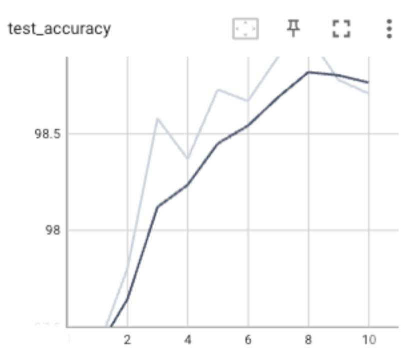
GRU:

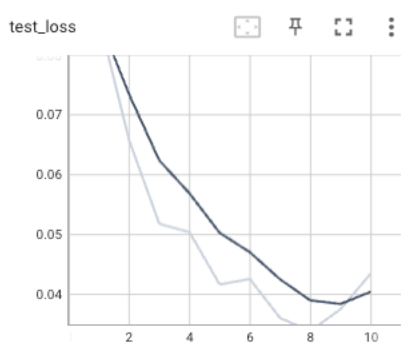
Learning_rate = 0.01

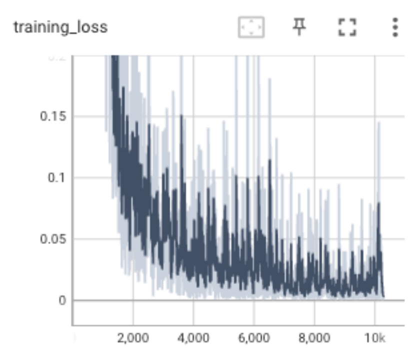
Num_layers = 2

Hidden_size = 128

Test accuracy after 10 epochs = 99.01%







3C

Based on the training experiments above, the CNN model outperforms the RNN model on the MNIST dataset. LSTM and GRU models also outperform CNN model on the MNIST dataset. CNNs are designed to learn spatial hierarchies automatically and adaptively from data, which is beneficial when working with image data like the MNIST dataset. They capture spatial relationships in data by applying convolutional filters which are excellent for detecting local patterns and their hierarchical structures in images.

Although LSTM and GRU are designed to perform efficiently on sequential data. They could perform well and reach high test accuracy on image dataset if the images in MNIST dataset has some sequential features that are detected by LSTM and GRU during the training and are not captured by CNN.