Team Members: Guanxiang Diao, Hanlin Yang, Songyan Ma, Wangshuang Xu, Boliang Chen

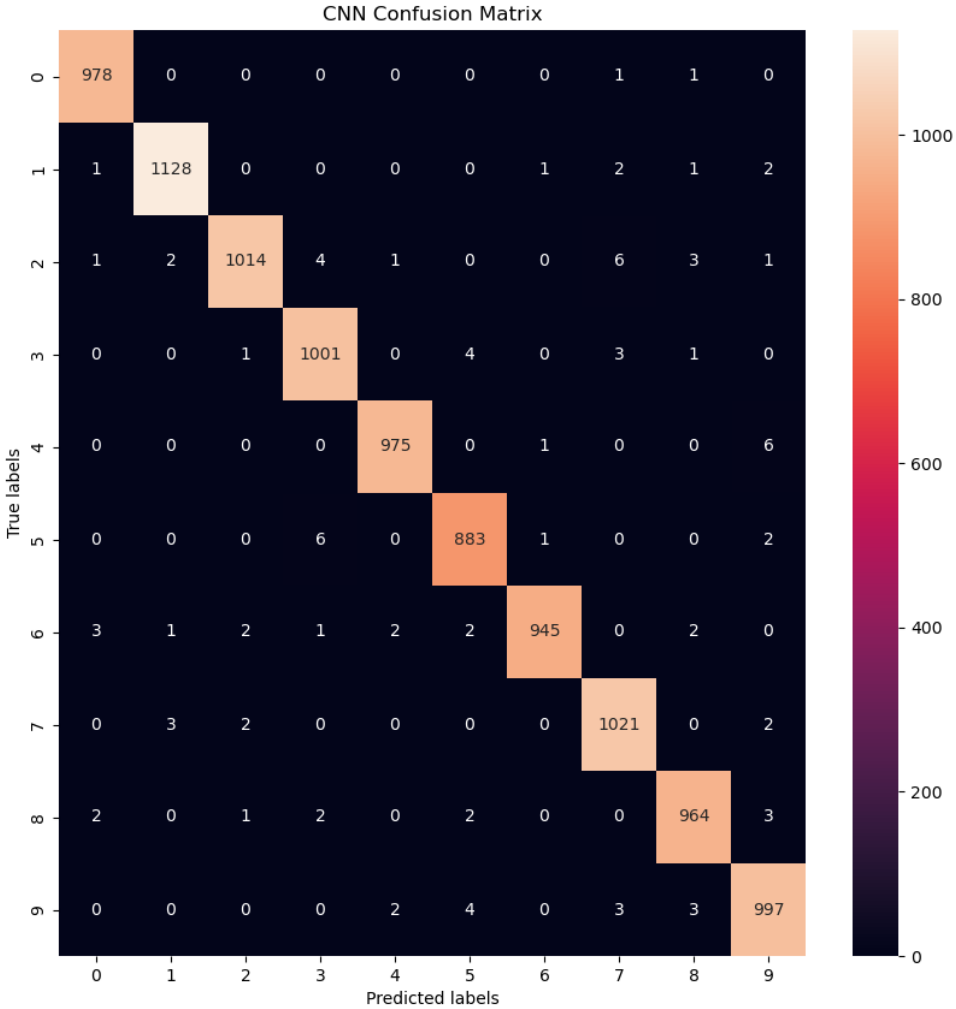
1. Tasks, datasets, and preprocessing (16 points)
This task involves developing deep learning models to recognize handwritten numbers. The data set used may be the MNIST data set, which is the standard data set for evaluating the performance of such tasks. It contains grayscale images of handwritten numbers 0 to 9, each with a size of 28x28 pixels.
Pre-processing steps may include normalization, scaling pixel values to a range of 0-1 to help the network converge during training. In addition, for compatibility with the output layer of the model, one-hot encoding may be applied to the labels, converting them to a binary matrix.

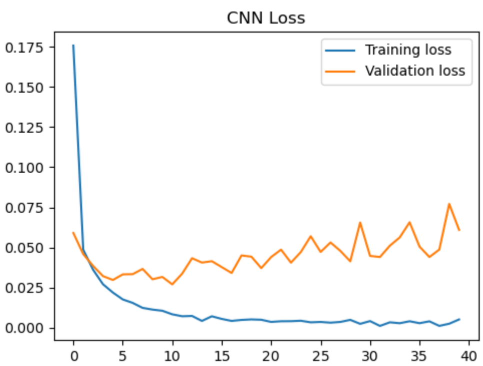
2. Implementation and architecture of two deep learning systems (16 points)
Two architectures are implemented: convolutional neural network (CNN) and recurrent neural network (RNN).
The CNN architecture usually consists of a convolutional layer that extracts features from the image, and then pools the layer to reduce the space size of the representation, reducing the number of parameters in the network and the amount of computation. This is followed by fully connected layers that perform classification based on features extracted by the convolutional layer.
The RNN architecture is designed to process data sequences. For image classification tasks, this may involve processing image rows as a sequence, allowing the model to capture dependencies along horizontal pixel rows.

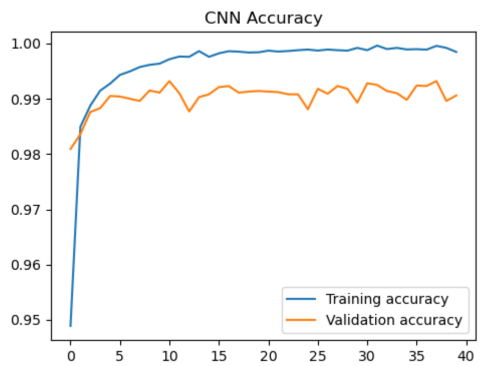
3. Training details of two deep learning systems (16 points)
Both systems have been trained many times, which is proved by the loss and accuracy graphs. Adjust the learning rate and other hyperparameters, such as batch size, to ensure convergence. The figure shows two periods of training where validation losses and accuracy are tracked along with training losses and accuracy to monitor overfitting.
The CNN model has a consistent improvement in training and verification accuracy, indicating that learning is effective, while the RNN model has a more unstable training process, and the verification loss increases at the second age, indicating possible overfitting.

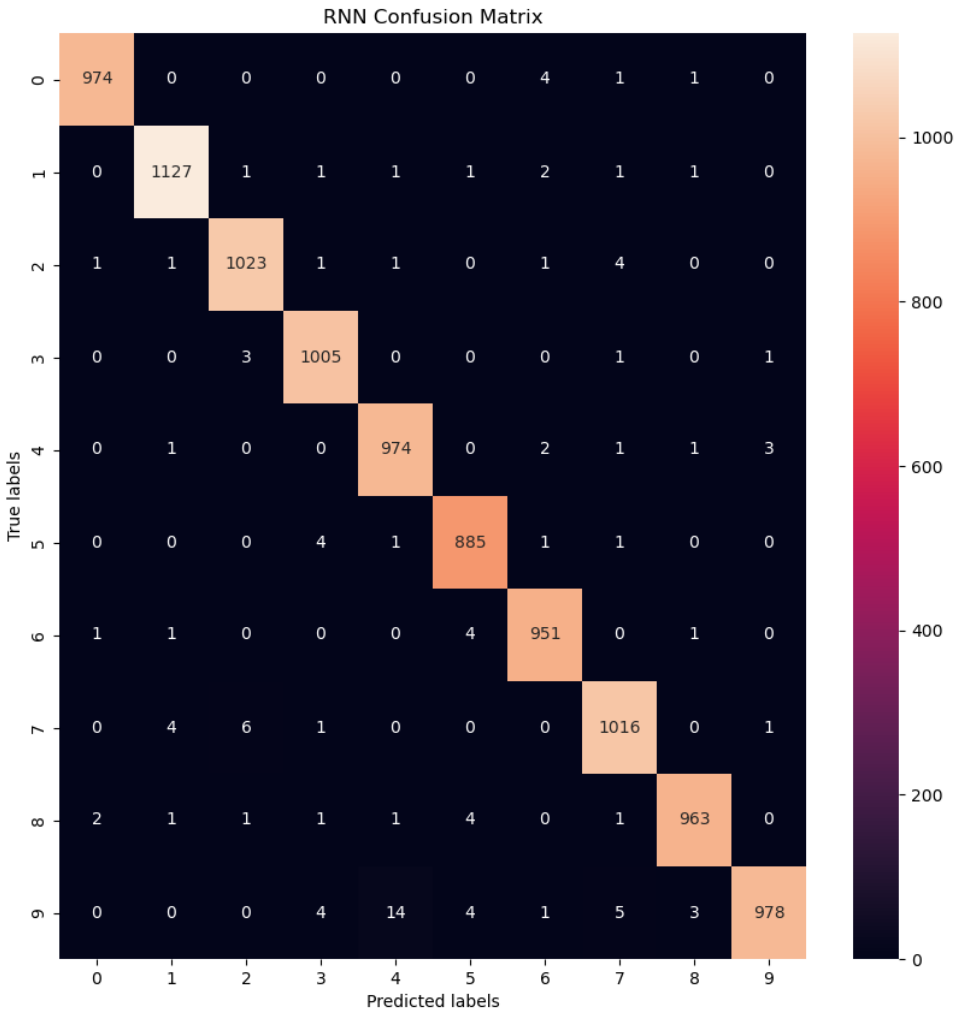
4. Results, Observations and Conclusions (16 points)
The performance of the model can be evaluated based on the accuracy and loss graphs provided, as well as the confusion matrix.
The CNN model showed high accuracy, with a training accuracy of nearly 99.4% and a correspondingly low loss, indicating that the CNN architecture is well suited for image-based tasks.
The RNN model, while still showing high accuracy, did not perform as well as the CNN, which is consistent with the general understanding that the CNN is better at processing image data due to its ability to capture spatial layers.
The confusion matrix indicates that the classification accuracy is high, but there are some misclassifications, and the RNN model is more serious than the CNN model.

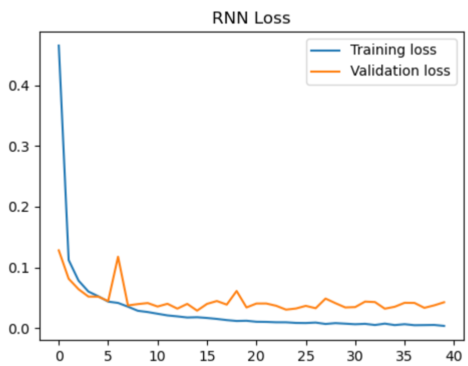
5. Challenges and obstacles you encountered and your solutions (16 points)
Major challenges may include model selection, hyperparameter tuning, and preventing overfitting.
For RNN models, validation losses begin to increase after the initial epoch, indicating overfitting. Techniques such as exit, early stop, or regularization can be employed to mitigate this.
For CNNS, the challenge may be to build an architecture that can capture complex patterns in images without overfitting, and ensure that the model generalizes well to previously unseen data.
The training process needs to be carefully monitored to ensure that no model overfits the training data. The solution here will include the use of validation sets to tune the model and possibly implement techniques such as data enhancement to improve generalization.

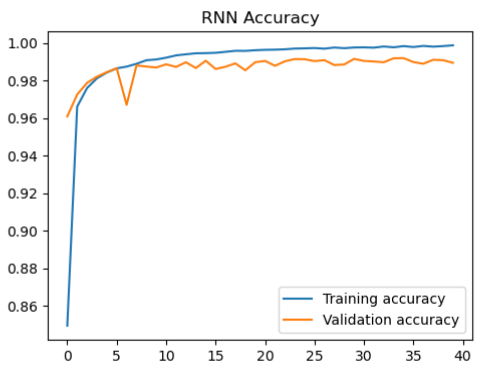

CNN Confusion Matrix
True labels
Predicted labels
0
1
2
3
4
5
6
7
8
9
0
200
400
600
800
1000
978 0 0 0 0 0 0 1 1 0
1 1128 0 0 0 0 1 2 1 2
1 2 1014 4 1 0 0 6 3 1
0 0 1 1001 0 4 0 3 1 0
0 0 0 0 975 0 1 0 0 6
0 0 0 6 0 883 1 0 0 2
3 1 2 1 2 2 945 0 2 0
0 3 2 0 0 0 0 1021 0 2
2 0 1 2 0 2 0 0 964 3
0 0 0 0 2 4 0 3 3 997


CNN Loss
Training loss
Validation loss
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0
5
10
15
20
25
30
35
40


CNN Accuracy
Training accuracy
Validation accuracy
0.95
0.96
0.97
0.98
0.99
1.00
0
5
10
15
20
25
30
35
40


RNN Confusion Matrix
974 0 0 0 0 0 4 1 1 0
0 1127 1 1 1 1 2 1 1 0
1 1 1023 1 1 0 1 4 0 0
0 0 3 1005 0 0 0 1 0 1
0 1 0 0 974 0 2 1 1 3
0 0 0 4 1 885 1 1 0 0
1 1 0 0 0 4 951 0 1 0
0 4 6 1 0 0 0 1016 0 1
2 1 1 1 1 4 0 1 963 0
0 0 0 4 14 4 1 5 3 978
True labels
Predicted labels
0 1 2 3 4 5 6 7 8 9
1000
800
600
400
200
0


RNN Loss
Training loss
Validation loss
0.4
0.3
0.2
0.1
0.0
0 5 10 15 20 25 30 35 40


RNN Accuracy
Training accuracy
Validation accuracy
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0 5 10 15 20 25 30 35 40