Name: Kuimu Ren
USC ID Number:1473482531
USC Email:kuimuren@usc.edu
Submission Date: 04/13/2024

Problem1

CNN Training on LeNet-5

Abstract and Motivation:

LeNet-5 is one of the pioneering convolutional neural network (CNN) architectures. It was primarily designed for handwritten digit recognition tasks, notably for use by the United States Postal Service. LeNet-5 features a concise structure comprising multiple convolutional and pooling layers, along with fully connected layers for classification. The network incorporates convolution and pooling operations, which effectively extract local features from images while reducing the dimensionality. By employing parameter sharing, LeNet-5 significantly reduces the number of parameters, enhancing its generalization capability.

(a) CNN Architecture

Homework Answer and Discussion:

1. A fully connected layer is a type of layer in a traditional neural network where each neuron is connected to every neuron in the previous layer. It combines the features learned from the previous layer to make a prediction. In classification tasks, it is common to convert the output of a fully connected layer into class probabilities via a softmax activation function.

   A convolutional layer consists of a set of learnable filters that slide over the input image for element-wise multiplication and aggregation. It extracts a hierarchy of spatial features from the input data, which it does by capturing patterns such as edges, textures or shapes.

   The max-pooling layer reduces the spatial size of the feature map generated by the convolutional layer by preserving the maximum value in the sliding window. It helps to reduce computational complexity, control overfitting, and achieve translation invariance by preserving the most important features while discarding irrelevant details.

   Activation functions introduce nonlinearities into the network by mapping input signals to output signals. It enables the network to learn complex patterns and relationships in the data by introducing nonlinear transformations. Common activation functions include ReLU and sigmoid.

   Softmax function is usually applied to the output layer of neural networks, used in multi-class classification tasks. It converts raw output scores into probabilities through normalization, ensuring that they sum to one. This enables the network to predict probability distributions over multiple classes, suitable for classification tasks.

2. Overfitting refers to the phenomenon that a machine learning model performs well on training data, but poorly on test data. It indicates that the model has overlearned the noise

and details in the training data, resulting in reduced generalization ability and unable to make effective predictions on unseen data.

There are a number of ways to effectively solve the overfitting problem. First, we can increase the size of the training dataset. This will result in a more average training model and prevent overfitting. The second is using Data Augmentation, It involves applying random transformations such as rotation, translation, scaling, and flipping to the training data to generate additional training samples. This helps the model learn more robust features and reduces its sensitivity to variations in the input data. The third is to use Regularization. L1 and L2 regularization are two common regularization methods that limit the size of the model parameters by adding a penalty term to the loss function. This helps prevent excessive growth of model parameters and reduces the risk of overfitting.

3. The ReLU function is defined as follows, the output is zero for negative inputs and stays the same for positive inputs. It is computationally efficient and widely used in deep learning due to its simplicity and effectiveness to effectively solve the vanishing gradient problem. However, ReLU neurons may cause them to output zero for any input less than zero during training.

$$f(x) = \max(0, x)$$

Leaky ReLU is an extension of ReLU that allows a small nonzero gradient when the input is negative. It is defined as follows. By allowing a small gradient for negative inputs, Leaky ReLU solves the problem of ReLU and prevents neurons from becoming inactive. Leaky ReLU has been observed to perform better than ReLU in some cases.

$$f(x) = \max(\alpha x, x)$$

α is a small constant.

ELU is another extension of ReLU that uses the negative part of the exponential curve smoothing function. The ELU is defined as follows.

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases}$$

ELU has negative values for x≤0 and helps push the average activation towards zero, similar to batch normalization, leading to better learning dynamics. ELU has been shown to outperform ReLU and Leaky ReLU in performance in some tasks, especially when dealing with vanishing gradients.

4. L1 loss is commonly used in regression problems such as linear regression or pixel-wise image reconstruction tasks. Because L1 loss is more robust to outliers because it penalizes the absolute value of the error instead of the squared error. This makes it more applicable for tasks that need to resist the influence of outliers.

The MSE loss is similarly commonly used in regression problems, especially when training neural networks to make numerical predictions. Because the MSE loss gives a higher penalty for smaller errors, which makes it more suitable for tasks that are highly sensitive to errors, it is also one of the standard regression loss functions.

BCE loss is commonly used in binary classification problems, such as image binary classification tasks or text classification tasks. BCE loss measures the difference between the probability distribution of the neural network output and the actual label, and it is one of the commonly used loss functions when training binary classification models.

(b) Compare classification performance on different datasets

Approach and Procedures:

First, We should load the dataset according to the parameters and preprocess the dataset, such as transform to tensors, normalize and create data loader.

Second, we need to design the convolutional layer and the full connection layer. After each convolutional layer, we also add a Max pooling layer, apply an activation function after each convolutional and fully connected layer, and for the output layer we use the Softmax activation function.

Third, we need to set the learning rate, momentum, maximum number of training rounds, loss function and optimizer for the training process.

Last, we used the model to start training the dataset, and then calculated and recorded the accuracy after each training round.

Experimental Results:

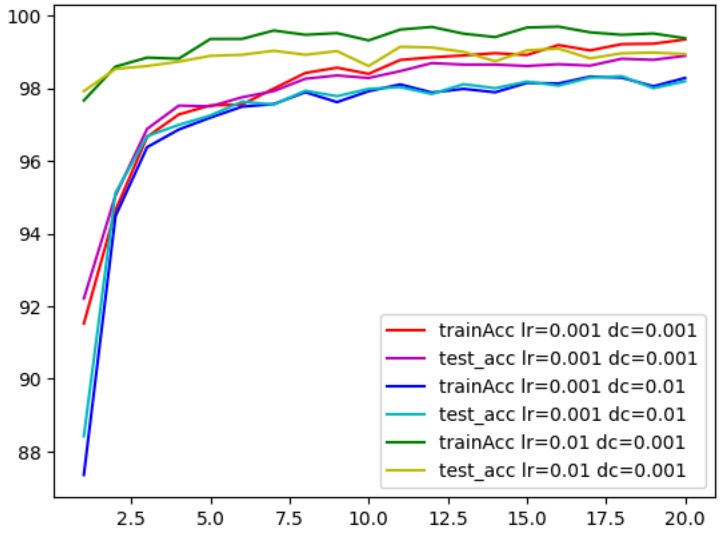

Figure1: train and test accuracy using SGD optimizer with different parameters

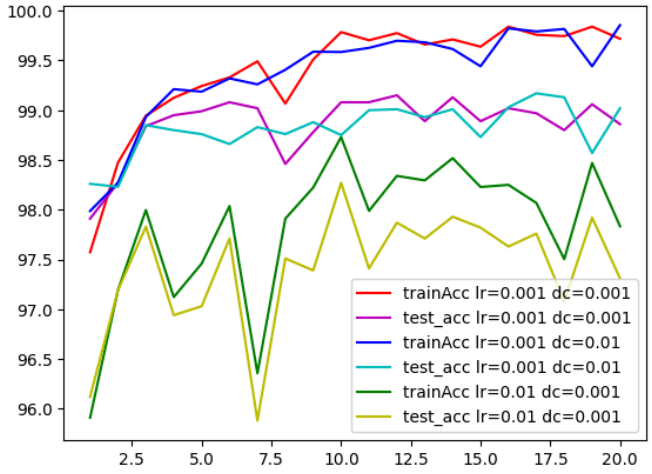

Figure2: train and test accuracy using AdamW optimizer with different parameters

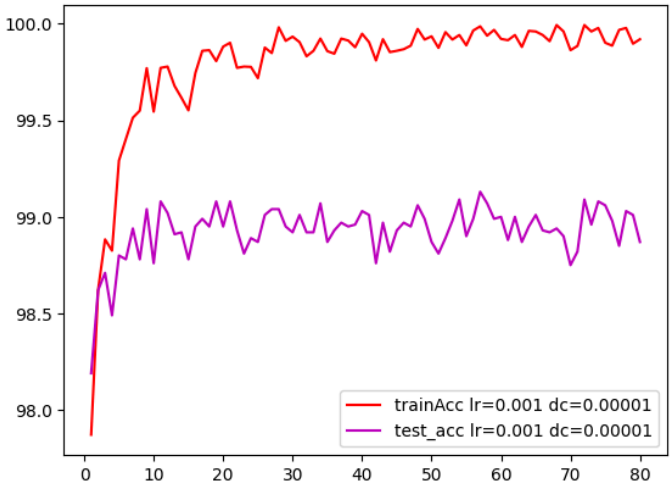

Figure3: train and test accuracy for MNIST

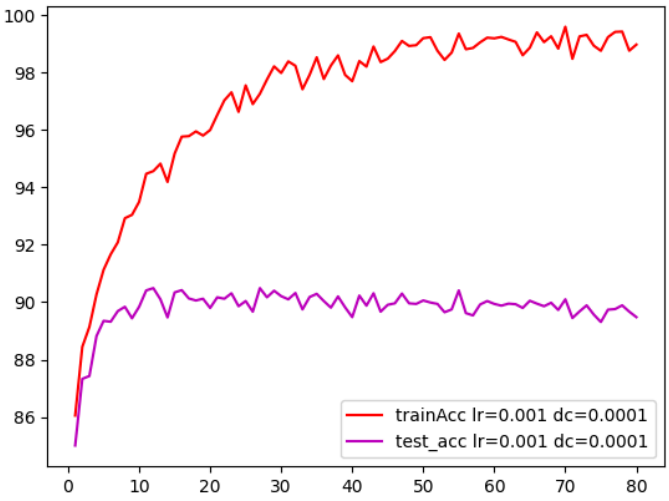

Figure4: train and test accuracy for FashionMNIST

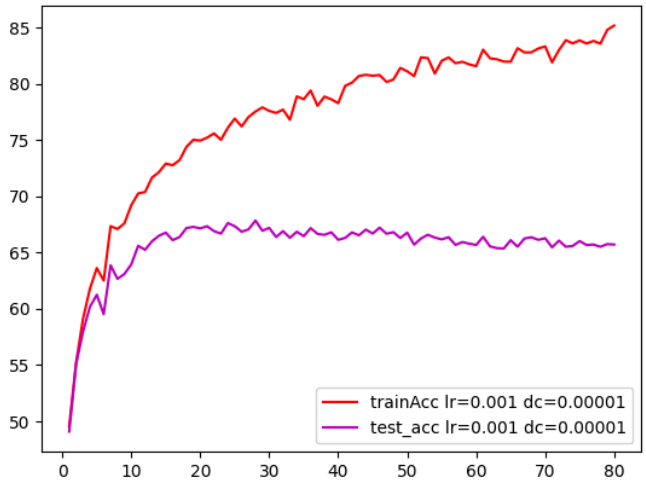

Figure5: train and test accuracy for CIFAR10

Homework Answer and Discussion:

1. In figure1, we use SGD optimizer and then plot the change of the accuracy of training set and test set with the increase of epoch under different learning rate and decay.
   In figure2, we use AdamW optimizer and then plot the change of the accuracy of training set and test set with the increase of epoch under different learning rate and decay.

   From the figure above, we can observe that because SGD optimizer uses an exponentially weighted moving average of the gradients to update the parameters, the weighted sum of the current gradient and the momentum component from the previous iteration to update the parameters in each iteration, the accuracy converges faster and the oscillations are reduced. Comparing the cases under different learning rate and decay, it can be found that increasing the learning rate can make the accuracy converge earlier, but it does not improve the maximum accuracy of the final convergence. An increase in decay will result in low initial accuracy, which will also affect the final accuracy

   From the above figure we can observe that AdamW optimizer uses a penalty parameter to prevent overfitting, so the change in accuracy is relatively more dramatic compared to SGD optimizer, and we can clearly see that AdamW optimizer's accuracy is higher from the beginning, and the final training set accuracy is close to 100%, but the final performance of the test set is similar to SGD optimizer. Under different learning rate and decay, we find that larger learning rate leads to a slight decrease in accuracy, while decay does not have a great impact on accuracy.

2. Show in figure3. The best test accuracy among five runs is 99.180, the mean test accuracy of 5 runs is 99.124, the standard deviation of test accuracy among five runs is about 0.05.

3. Show in figure4. The best test accuracy among five runs is 91.180, the mean test accuracy of 5 runs is 90.844, the standard deviation of test accuracy among five runs is about 0.225.

4. Show in figure5. The best test accuracy among five runs is 69.160, the mean test accuracy of 5 runs is 68.9, the standard deviation of test accuracy among five runs is about 0.214.

5. We can obviously find that the accuracy of the three datasets is different, MNIST dataset performs the best, then FashionMNIST dataset, and CIFAR-10 is the worst. The reasons are as follows:
   There may be differences in the distribution of categories for different datasets. For example, the MNIST dataset is handwritten digit images, the FashionMNIST dataset is clothing images, and the CIFAR-10 dataset is real images containing different kinds of objects. The data complexity may also be different for different datasets. For example, the CIFAR-10 dataset is relatively more complex than the MNIST dataset because it contains more kinds of objects and the appearance and background of the objects are more diverse. And the number and quality of samples may be different for different datasets, which will affect the training effect of the model. For example, the CIFAR-10 dataset is different in size, number, and number of channels compared to the other two.

(c) Evaluation and Ablation Study

Approach and Procedures:

First, Generates a standardized confusion matrix on the test set, for 10 categories. Then find the top three pairs of categories with the most confusion. Show an example for each category pair.

Second, For the CIFAR-10 data set, the ROC curve is drawn using the ROC method. For each category, the corresponding True Positive Rate and False Positive Rate are calculated, and then the ROC curve for each category is drawn.

Third, Calculate the area under ROC curve (AUC) for the CIFAR-10 dataset. For each category, its corresponding area under the ROC curve is calculated. The total AUC value of the CIFAR-10 data set is obtained by weighted summations of the areas under the ROC curves of all categories.

Experimental Results:

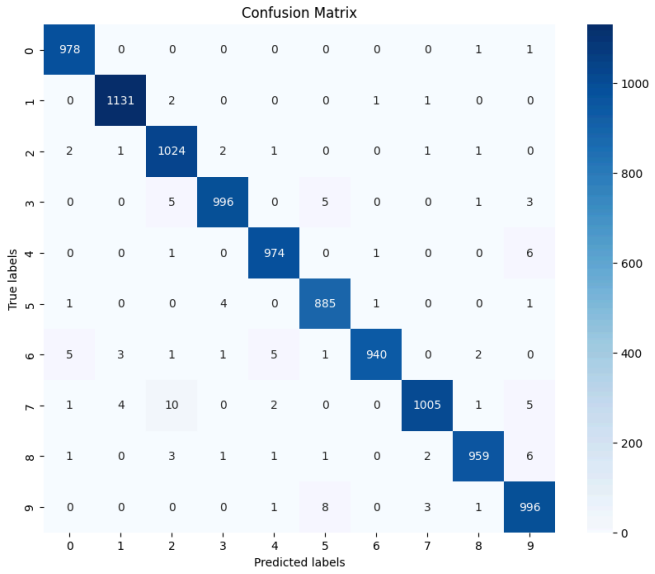

Figure1: Confusion Matrix for MNIST

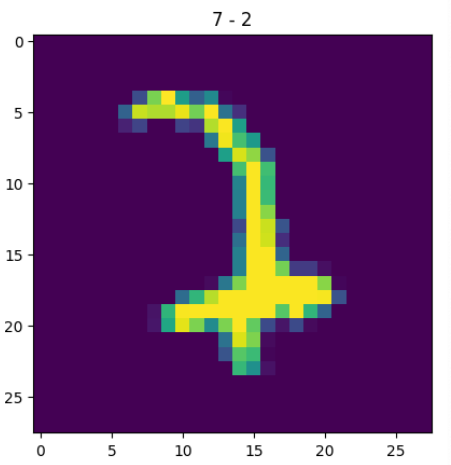

Figure2: The wrong picture treats 7 as 2

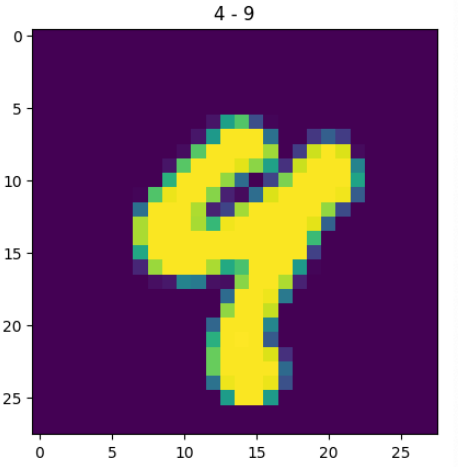

Figure3: The wrong picture treats 4 as 9

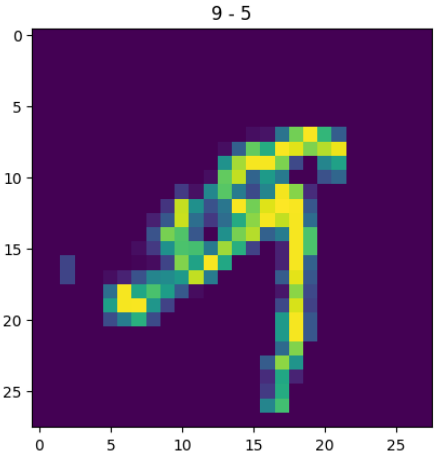

Figure4: The wrong picture treats 9 as 5

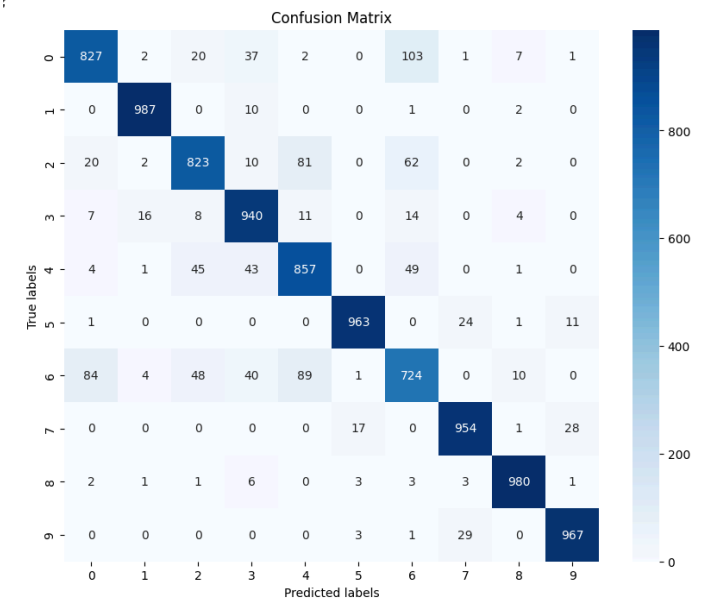

Figure5: Confusion Matrix for FashionMNIST

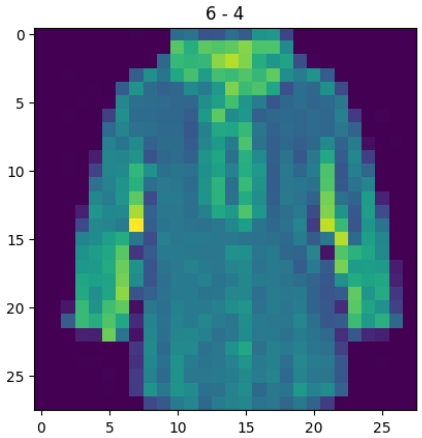

Figure6: The wrong picture treats shirt as coat

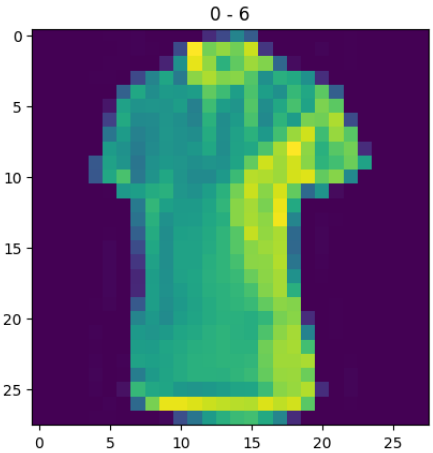

Figure7: The wrong picture treats T-shirt as shirt

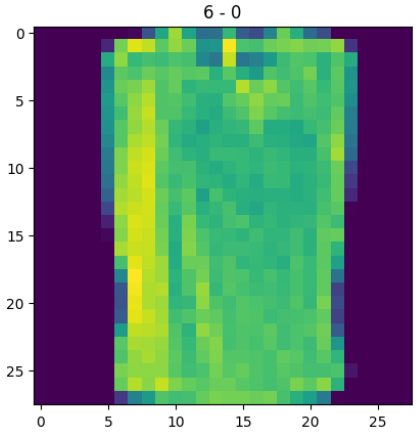

Figure8: The wrong picture treats shirt as T-shirt

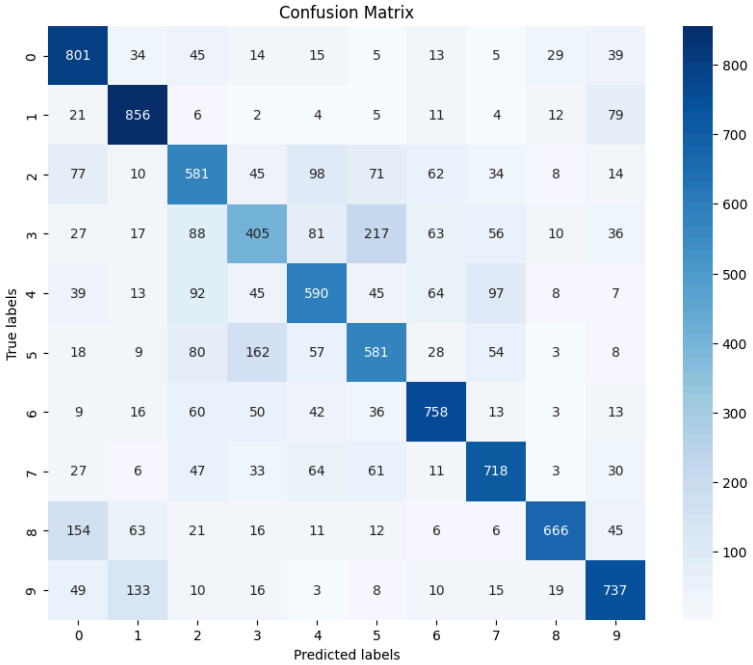

Figure9: Confusion Matrix for CIFAR10

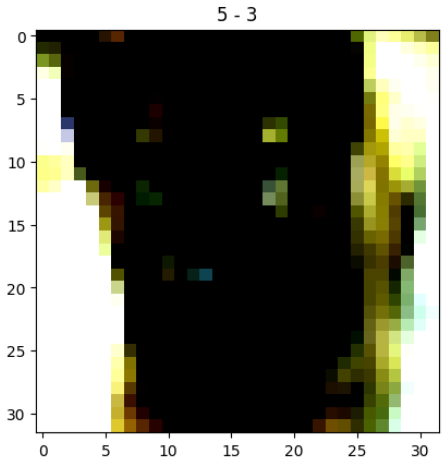

Figure10: The wrong picture treats dog as cat

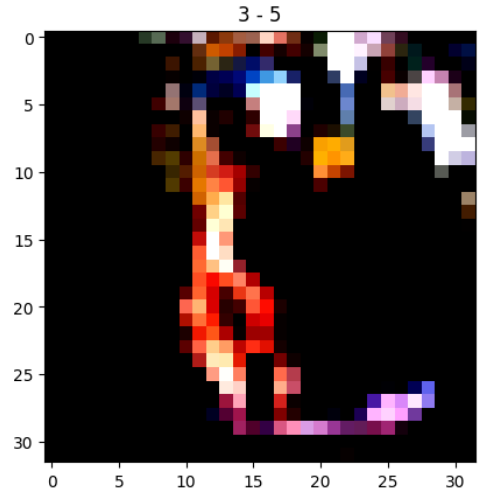

Figure11: The wrong picture treats cat as dog

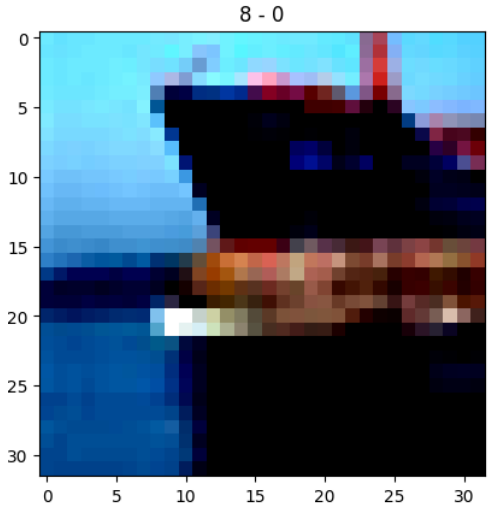

Figure12: The wrong picture treats ship as airplane

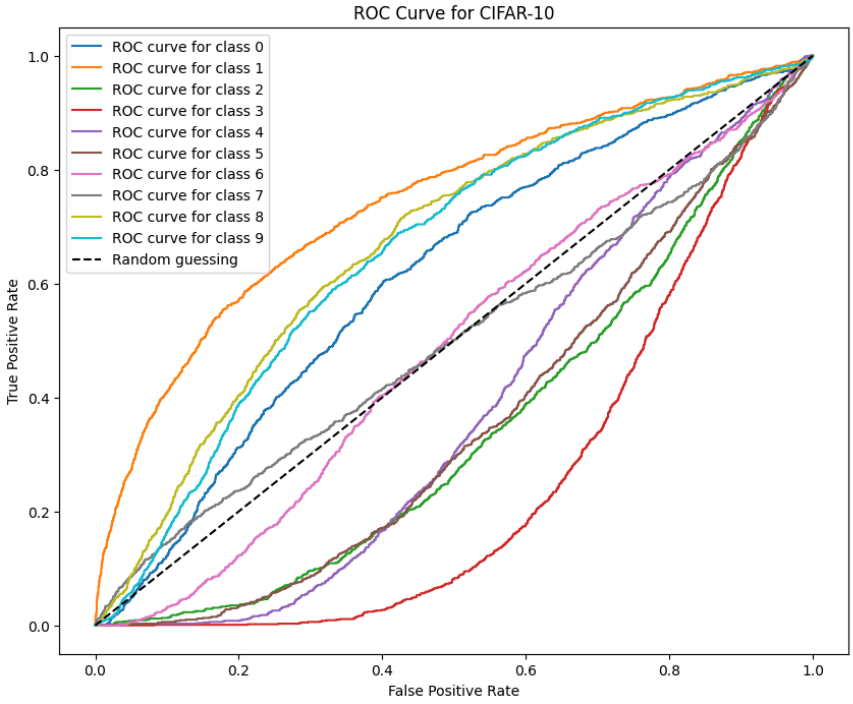

Figure13: the ROC curve for each class

```
ROC AUC for class 0: 0.61
ROC AUC for class 1: 0.74
ROC AUC for class 2: 0.35
ROC AUC for class 3: 0.25
ROC AUC for class 4: 0.38
ROC AUC for class 5: 0.36
ROC AUC for class 6: 0.48
ROC AUC for class 7: 0.50
ROC AUC for class 8: 0.67
ROC AUC for class 9: 0.66
```

Figure14: the AUC for each class

Homework Answer and Discussion:

1. Figure1 show the normalized confusion matrix for MNIST. We can find the top three confused pairs of classes are (7,2), (9,5), (4,9). The example for each of these three pairs show in figure 2,3,4. We can find from the picture that for the three pictures of 7 and 2,9 and 5,4 and 9, the style of their writing is different from the standard number, even human beings are not easy to distinguish the correct number.

   Figure5 show the normalized confusion matrix for FashionMNIST. We can find the top three confused pairs of classes are (0,6), (6,4), (6,0). The example for each of these three pairs show in figure 6,7,8. From the picture, we can find that the difference between T-shirt, shirt and coat is not very big, especially for such small pictures with low pixels, it is more difficult to distinguish, which is also the reason why FashionMNIST is not as accurate as MNIST.

Figure9 show the normalized confusion matrix for CIFAR-10. We can find the top three confused pairs of classes are (3,5), (5,3), (8,0). The example for each of these three pairs show in figure 10,11,12. From the above three pictures, we can find that for CIFAR-10, the classification is no longer some simple characters, but some very different items, and for small pictures with low pixels, it is difficult to distinguish what the image is, even if people to distinguish it is difficult to classify correctly.

2. The ROC curve for each class shows in figure13.
   The difference between the ROC curve and the accuracy is that the ROC curve assesses the performance of the classifier by plotting the relationship between the true rate (TPR) and the false positive rate (FPR), while the accuracy is simply the proportion of the classifier that is correctly classified. The ROC curve is particularly suitable for unbalanced datasets because it takes into account trade-offs between positive and negative categories. However, accuracy may be biased in unbalanced data sets, because it only focuses on the overall classification accuracy rate, and does not consider the balance between positive and negative categories. The ROC curve can also help us understand the performance of the classifier under different thresholds, while the accuracy is only the performance under one threshold.

3. The AUC on the whole CIFAR-10 dataset is 0.5 and AUC for each class is show in figure14.

(d) Classification with noisy data
Experimental Results:

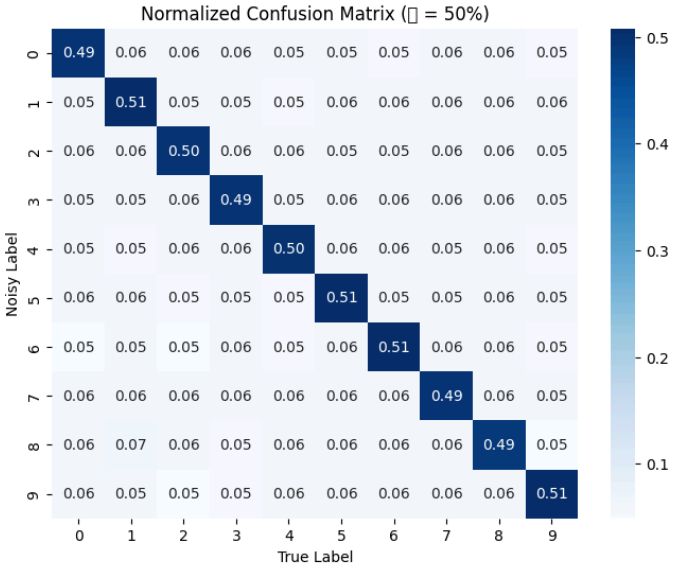

Figure1: normalized confusion matrix for ϵ = 50%

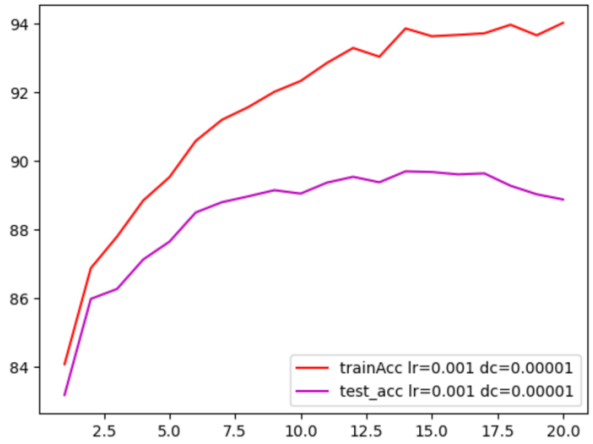

Figure2: testing accuracy with noisy level $\epsilon = 0\%$

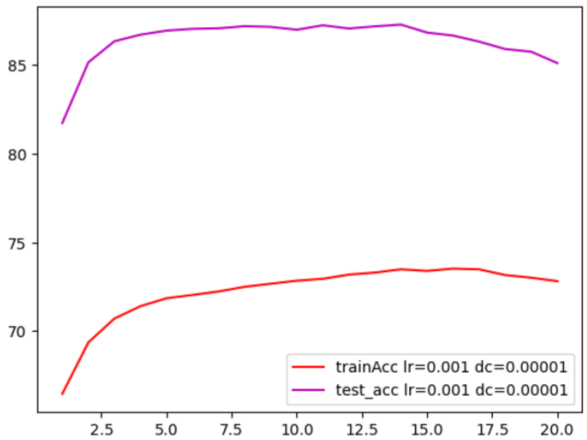

Figure3: testing accuracy with noisy level $\epsilon = 20\%$

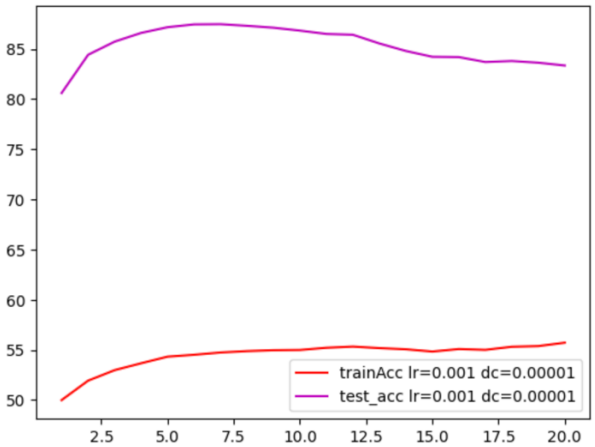

Figure4: testing accuracy with noisy level $\epsilon = 40\%$

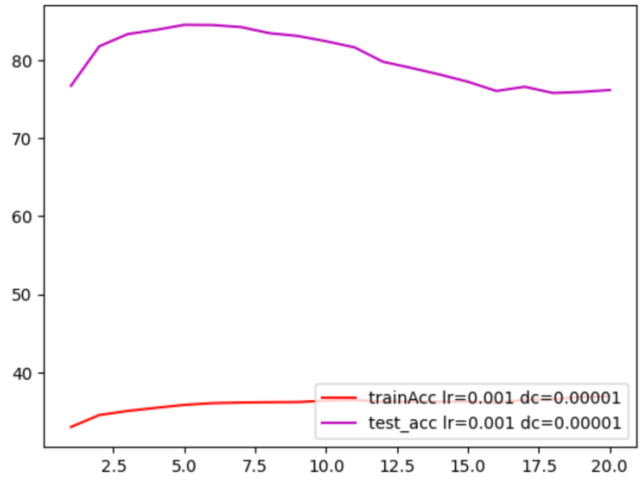

Figure5: testing accuracy with noisy level $\epsilon = 60\%$

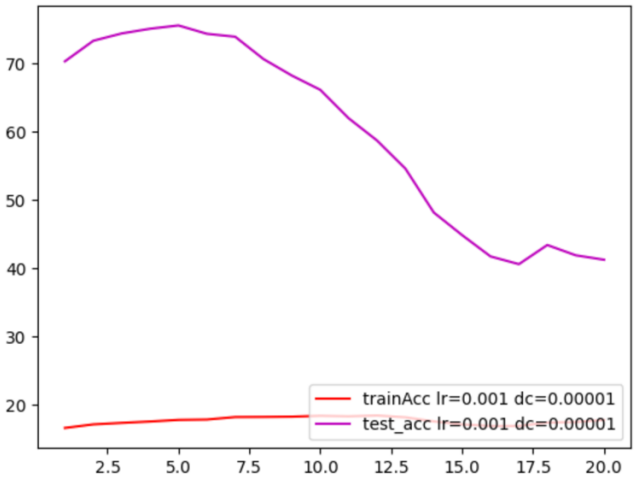

Figure6: testing accuracy with noisy level $\epsilon = 80\%$

Homework Answer and Discussion:

1. We generate the confusion matrix according to the given noise level ϵ. The generated confusion matrix is then used to convert the original label into a noise label to introduce symmetric label noise. Finally, the confusion matrix is calculated, taking the real label and the label with added noise as input, and the confusion matrix is calculated to evaluate the effect after the introduction of noise.
   The normalized confusion matrix shows in Figure1.

2. Result show form figure2 to figure6.
3. From the figure above, we can clearly find that more false labels will lead to a significant decline in the accuracy of the training set, but have little impact on the accuracy of the test set. I think this is because when error labels appear in the training set, the model may overfit those error labels, trying to fit mislabeled data. This results in degraded performance of the model on the training set because the model is no longer able to properly generalize to the real data in the training set. However, the test set differs from

the training set and generally provides better generalization, so there is less performance degradation on the test set.

Problem2

Vision Transformer

Abstract and Motivation:

The Vision Transformer (ViT) is a model that applies the self-attention mechanism to process image data. Originally used in Transformer models for natural language processing, the self-attention mechanism achieved great success in processing sequential data. ViT adapts this mechanism to image data by partitioning the image into a sequence of image patches and treating these patches as a sequence input to the Transformer model, thereby enabling global context modeling of the image.

A) ViT Architecture

Homework Answer and Discussion:

1. Patch embedding is the process of decomposing the input image into smaller, fixed-size blocks and converting each block into an embedded vector.
   Location coding is crucial for preserving spatial information of blocks in a sequence. Since ViT treats image blocks as tokens in a sequence, there is a lack of inherent positional information. These codes represent the spatial coordinates or positions of the blocks in the original image.
   Multi-head attention is responsible for capturing global dependencies and relationships in the block embedding sequence. It does this by calculating the attention score between block pairs and using it to aggregate information from other tokens.

B) Compare classification performance over MNIST dataset

Approach and Procedures:

First, all the steps are the same as in A.

Second, before we perform K-means, we need to perform PCA dimensionality reduction on the 24-d future vector, and then we can obtain the 3-d future vector, which can then be classified using K-means algorithm.

Third, we can also apply morphological closing operation on image result. This operation expands and then corrodes the regions in the image, effectively filling small holes or gaps within the regions.

Experimental Results:

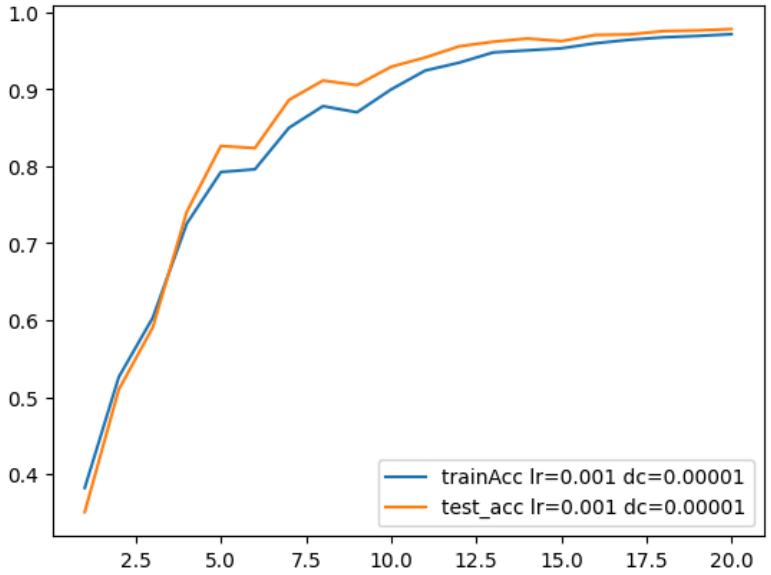

Figure1: The training and testing acc curve of the 20 epochs

Homework Answer and Discussion:

1. Test accuracy is 97.83%, and the curve shows in figure1.
2. The model file size is 774KB.
3. First, ViT may perform better with large and diverse data sets, as the Transformer structure helps models learn global information and long-distance dependencies. In contrast, CNNS are better suited for dealing with local features and may perform better for small and simple data sets. Then, for complex visual tasks, ViT may perform better because it is better able to capture semantic information in images. For simple tasks, the simple structure of the CNN may be more efficient. Finally, the ViT model is usually larger than the CNN model for the same task and requires more parameters to train. Therefore, if training data is insufficient or training time is limited, CNNS may have an advantage because they are generally easier to train and adjust.