Deep Learning Assignment2

RE6111024 葉嘉浤

***Abstract*—This assignment is to implement a two-layer neural network and a LeNet5 from scratch, and modified the referenced LeNet5. We compared the three mentioned models with the accuracy on image classification. The code can be found at https://github.com/chyeh1126/DeepLearning-2023/tree/main/HW2.**

***Index Terms*—Deep Learning, Neural Network, Image Classification**

I. Introduction

In the field of computer vision, common tasks include image classification, object detection, and so on. The goal of image classification is to use the features presented in an image to let the model output probability values of the presence of certain objects in the image, thereby determining whether certain objects exist in the image.

II. Method

In this section, we will introduce the model used in this assignment and the related hyperparameter settings.

A. Two-layer perceptron

In this two-layer perceptron, we apply two fully-connected layer, and use relu function between the two layer.At the output layer, we use softmax function to transform our predicted value from second layer into the probability of each class. The following table shows the hyperparameter setting for two-layer perceptron.

TABLE I
Hyperparameter setting of Two-layer perceptron

Layer	Setting
Linear 1	in dim = 64, out dim = 50
Linear 2	in dim = 50, out dim = 50

B. LeNet5

LeNet5 is a convolutional neural network model for image classification proposed by Yann LeCun and others in the year 1998. The following figure and table show the architecture of LeNet5 and the related hyperparameter setting.

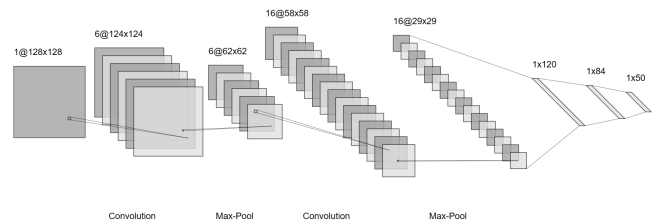

Fig. 1. The structure of LeNet5

TABLE II
Hyperparameter setting of LeNet5

Layer	Setting
Convolution 1	kernel size = 5×5, in channels = 1, out channels = 6
Pooling 1	kernel size = 2×2
Convolution 2	kernel size = 5×5, in channels = 6, out channels = 16
Pooling 2	kernel size = 2×2
Linear 1	in dim = $29 \times 29 \times 16$, out dim = 120
Linear 2	in dim = 120, out dim = 84
Linear 3	in dim = 84, out dim = 50

C. The modified LeNet5

The following table shows the the structure of the modified LeNet5. It is similar to the original LeNet5, only adjust the kernel size, change the activation function $sigmoid(x)$ to $x \times sigmoid(x)$, and add a convolution layer.

TABLE III
Hyperparameter setting of the modified LeNet5

Layer	Setting
Convolution 1	kernel size = 5×5, in channels = 1, out channels = 6
Pooling 1	kernel size = 2×2
Convolution 2	kernel size = 5×5, in channels = 6 out channels = 16, padding = 1
Pooling 2	kernel size = 2×2, padding = 1
Convolution 3	kernel size = 3×3, in channels = 16 out channels = 32
Linear 1	in dim = $29 \times 29 \times 32$, out dim = 120
Linear 2	in dim = 120, out dim = 84
Linear 3	in dim = 84, out dim = 50

III. Experiment

The dataset for this experiment consists of 63,325 images for training, 450 for validation, and 450 for testing, all of which belong to 50 different categories.The images are read as gray-scaled images, then resized to 128×128 before feature extraction. In two-layer perceptron, we use global color histogram to extract image feature, but do not use any feature extractor in LeNet5 since the input for LeNet5 is a image.After extracting image feature, we normalize the pixel value and get the final input for each model. We train two-layer perceptron for 50 epochs, and 10 epochs for LeNet5 and the modified LeNet5, then use top-1 accuracy and top-5 accuracy to evaluate the performance of each model.

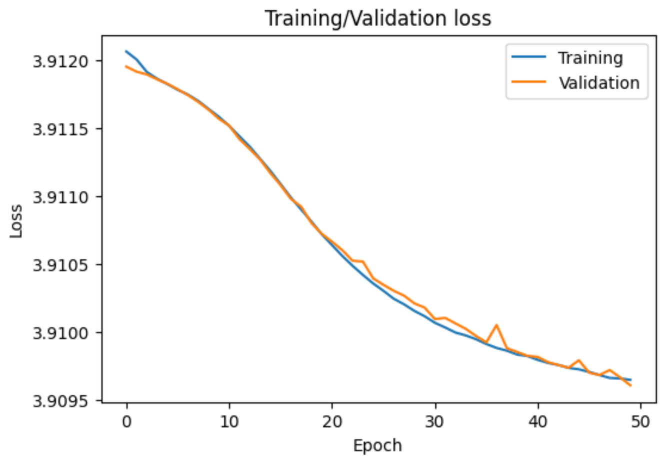

Fig. 2. Training and Validation loss of two-layer perceptron

The above figure shows the training and validation loss for two-layer perceptron in each epoch. We can see that both the two types of loss decreases as the epoch increases.

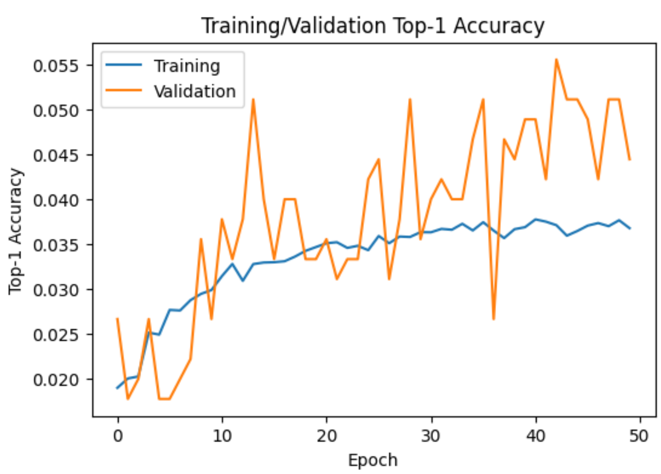

Fig. 3. Top-1 accuracy of two-layer perceptron

The above figure shows the training and validation top-1 accuracy for two-layer perceptron in each epoch. We can see that both the two types of top-1 accuracy change significantly between each epoch, while they increases as the epoch increases.

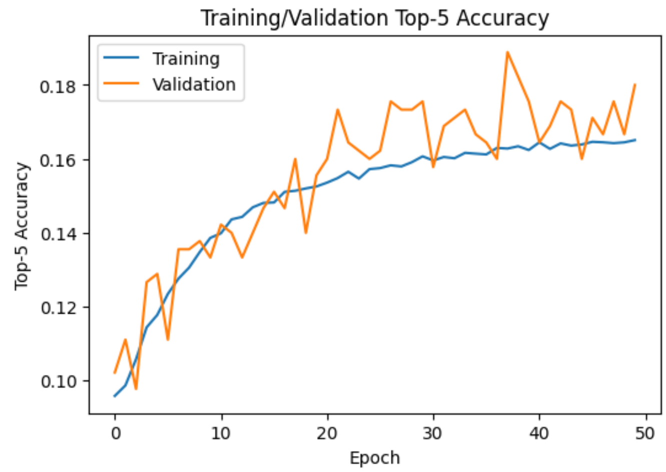

Fig. 4. Top-5 accuracy of two-layer perceptron

The above figure shows the training and validation top-5 accuracy for two-layer perceptron in each epoch. The training top-5 accuracy will converge to around 0.18, while the validation accuracy changes significantly between each epoch, and finally converges to around 0.18.

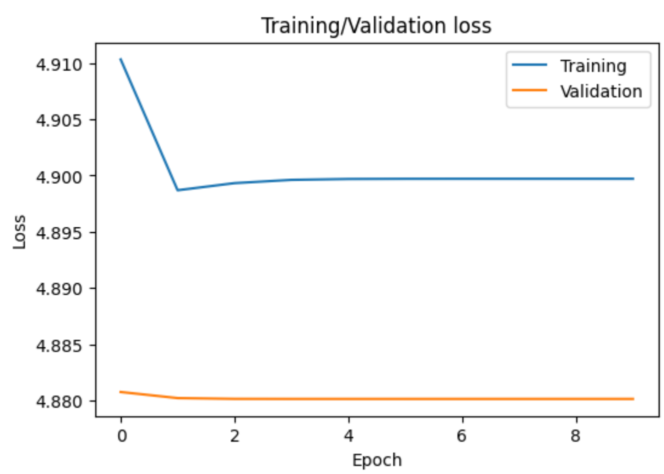

Fig. 5. Training and Validation loss of LeNet5

The above figure shows the training and validation loss for LeNet5 in each epoch. For the training loss, it decreases at the first epoch, but slightly increases in the second epoch, and become stable at following epochs; For the validation loss, it also decreases in the first epoch, and become stable at following epochs.

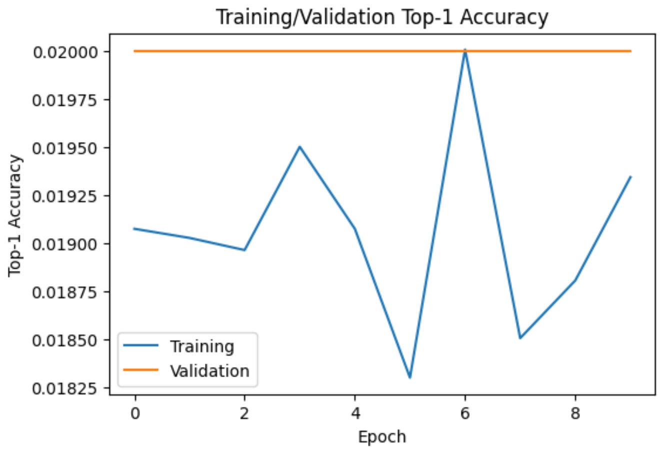

Fig. 6. Top-1 accuracy of LeNet5

The above figure shows the training and validation top-1 accuracy for two-layer perceptron in each epoch. The training top-1 accuracy is approximately 0.019 and it changes significantly, while the validation top-1 accuracy is almost 0.02.

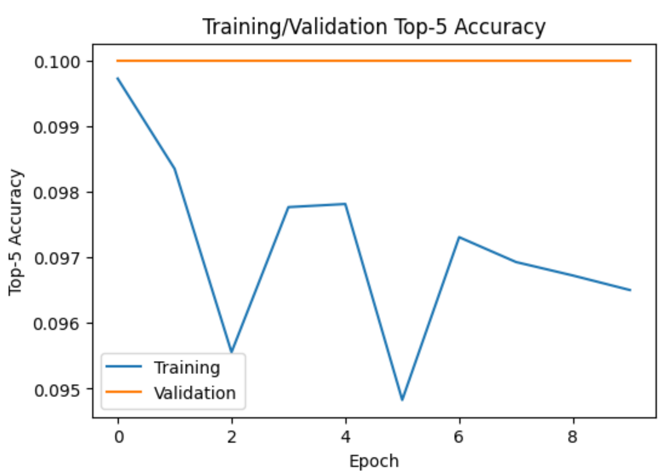

Fig. 7. Top-5 accuracy of LeNet5

The above figure shows the training and validation top-5 accuracy for two-layer perceptron in each epoch. The training top-5 accuracy decreases as the epoch increases and the final value is approximately 0.097, while the validation top-5 accuracy is almost 0.1.

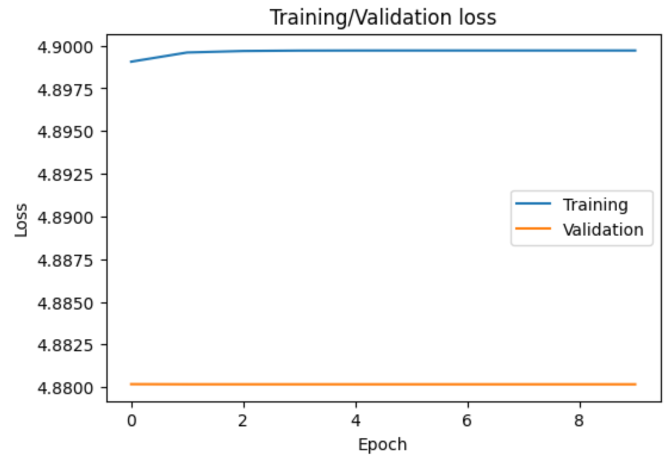

Fig. 8. Training and Validation loss of the modified LeNet5

The above figure shows the training and validation loss for modified LeNet5 in each epoch. For the training loss, it increases at the first epoch, and become stable at following epochs; For the validation loss, it does not change obviously in the graph.

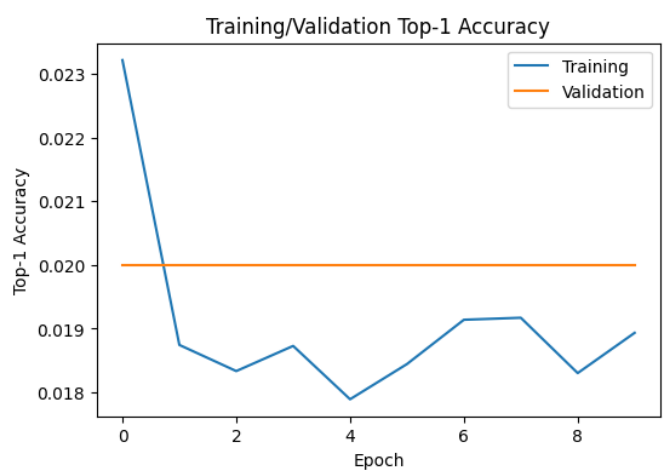

Fig. 9. Top-1 accuracy of the modified LeNet5

The above figure shows the training and validation top-1 accuracy for two-layer perceptron in each epoch. The training top-1 accuracy is highest at first epoch, at around 0.023, and then decreases as the epoch increases. The validation is almost the same, around 0.02.

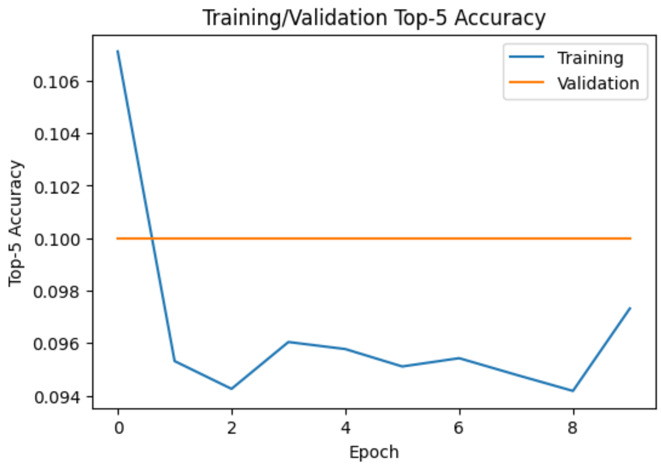

Fig. 10. Top-5 accuracy of the modified LeNet5

The above figure shows the training and validation top-5 accuracy for two-layer perceptron in each epoch. The training top-5 accuracy decreases as the epoch increases and the final value is approximately 0.106, while the validation top-5 accuracy is almost the same, around 0.1.

TABLE IV
Top-1 accuracy

Model	Top-1 accuracy	
	Validation	***Testing***
Two-layer perceptron	0.033 $\pm$0.01	0.016
LeNet5	0.019 $\pm$0.00	0.02
Modified LeNet5	0.019 $\pm$0.00	0.02

The above table records the validation and testing Top-1 accuracy for each model. The two-layer perceptron reach the highest accuracy in validation, while LeNet5 and the modified LeNet5 obtain the highest accuracy in testing.

TABLE V
Top-5 accuracy

Model	Top-5 accuracy	
	Validation	***Testing***
Two-layer perceptron	0.149 $\pm$0.02	0.098
LeNet5	0.1 $\pm$0	0.1
Modified LeNet5	0.1 $\pm$0	0.1

The above table records the validation and testing Top-5 accuracy for each model. Like the result in Top-1 accuracy, the two-layer perceptron reach the highest accuracy in validation, while LeNet5 and the modified LeNet5 obtain the highest accuracy in testing.

IV. Conclusion

In this assignment, we try to implement two-layer perceptron and a LeNet5, and modify the LeNet5, and evaluate their performance with an image classification task. We record the loss, top-1 accuracy, top-5 accuracy on the train and validation dataset, and organize the top-1 accuracy and top-5 accuracy on the validation and test dataset.

According to the experiment results, we can see that the loss of the two-layer model can steadily decrease and the overall accuracy improves. For LeNet5 and the modified LeNet5, the loss converges in the early stage, but the accuracy will fluctuate or decrease with the increase of epoch. Further compare LeNet5 and the modified LeNet5, we observe a slight improvement in accuracy and loss with the modified one.

Comparing the performance of three models on the validation and test dataset,two-layer perceptron achieves the highest accuracy on the validation dataset, while the performances of LeNet5 and the modified LeNet5 are the same, both of them achieve the highest accuracy on the test dataset.

Looking back this assignment, I think the backpropagation. As for the performance of model, there are still many techniques for improvement, such as the generation of initial weights, the selection of activation function, the dimension setting in fully connected layer, and the size of kernel in convolution layer.

References

- Two-layer perceptron : https://towardsdatascience.com/math-neural-network-from-scratch-in-python-d6da9f29ce65
- LeNet5 : https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/
- Calculate output size in CNN : https://androidkt.com/calculate-output-size-convolutional-pooling-layers-cnn/