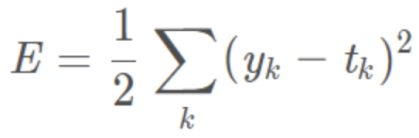
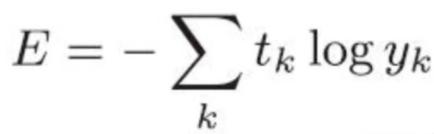
Q1.

Convolutional kernel: Since the direct use of full connection between the input layer and the hidden layer for the input image would result in too many parameters, the convolution kernel was created to reduce the computation [1]. The essence of convolutional kernel is to extract local features from input images by imitating convolutional neurons, thus greatly reducing the number of parameters while maintaining the original image features. Specifically, a convolution kernel has three properties: kernel size, stride, and padding. The size of the kernel determines the receptive field of the convolution kernel on the orginal input. A larger convolution kernel means that more image information can be withdrawn, whereas the parameter size and the calculation amount will increase. The stride determines the accuracy of extraction. If the step size is too large, many pixels of the input image will be missed, which will make the image extracted by features less accurate. The padding is related to the case that the original input image is not proportional to the size of the convolution kernel, and this problem is solved by boundary filling. In practice, multiple similar convolution kernels are set to specific receptive fields and locate at different parts of the image. The outputs of such set of kernels form the feature map [2].

The loss function: The role of the loss function is to measure the difference between the predicted result and the target value. Researchers can reduce the value of the loss function by assessing the difference between these two values and adjusting the parameters of the model using optimization algorithms. The loss functions commonly used in convolutional neural networks are Mean Square Error(MSE) function and Cross-entropy function [3]. The fundamental form of MSE function is:



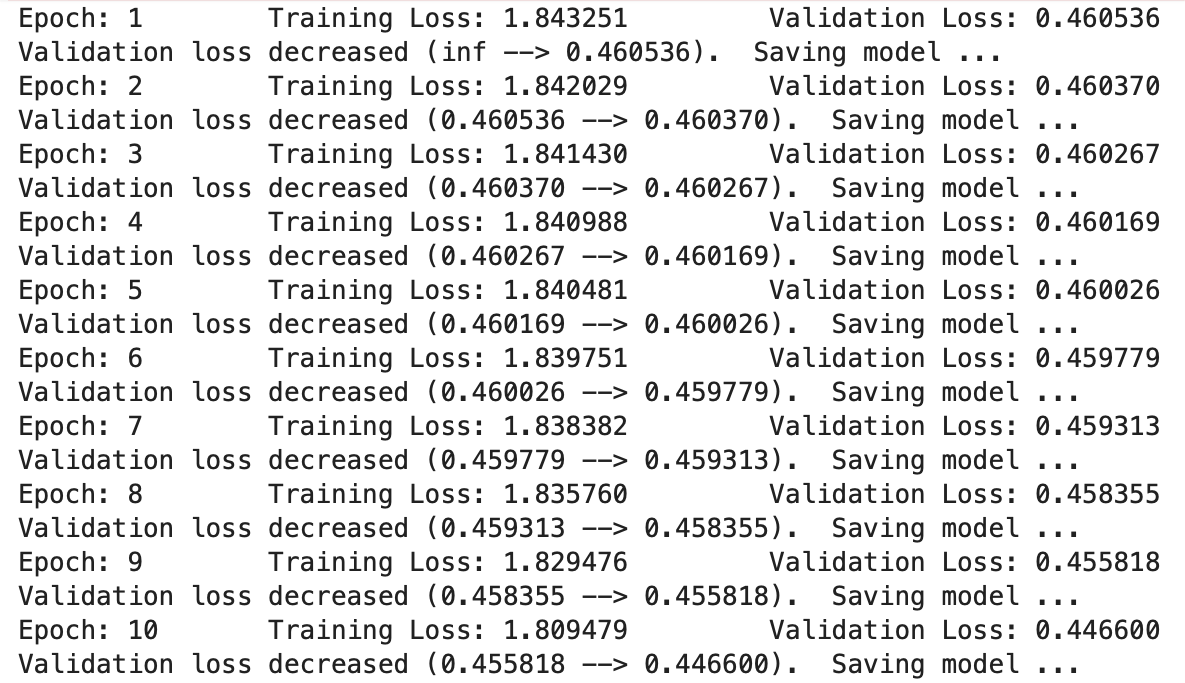
It is obviously that this function focuses on calculating the difference between predicted value ‘y’ and the target value ‘t’. Meanwhile, the basic form of Cross-entropy function is as follows:



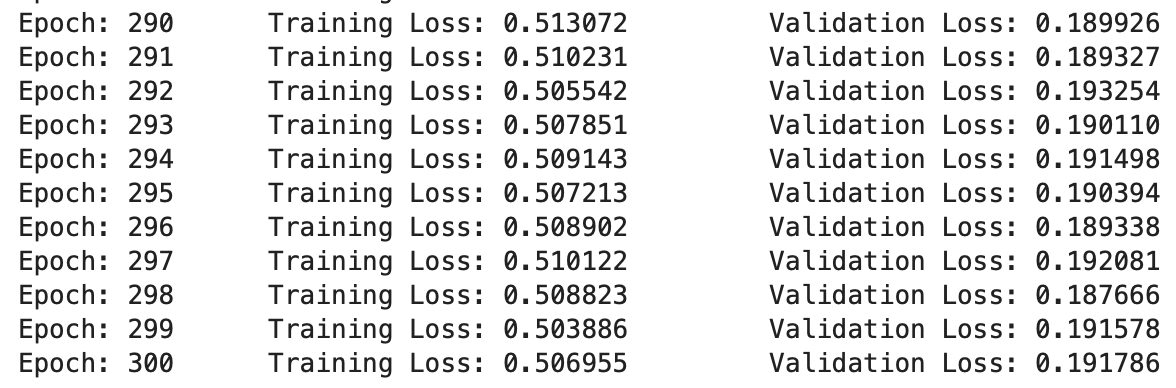
In this function, ‘t’ represents the target value for each category. Another essential parameter in the loss function is ‘y’, and it represents the predicted value of each input data. For the part within the log, it has gone through the traditional normalization process using the activate function like softmax function and sigmoid function.

Q2.

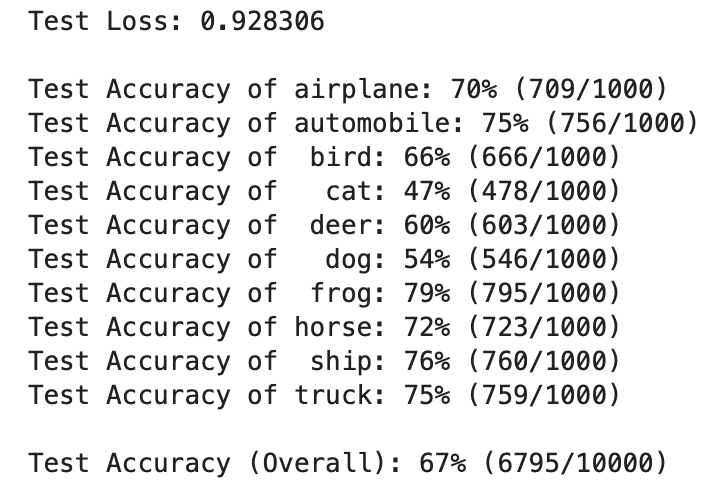
The training process of the CNN model on CIFAR-10:



In the early stage of training, the model converges rapidly, and the value of the loss function decreases after almost every iteration. But after two hundred iterations, the model is no longer convergent. As shown in the picture below:

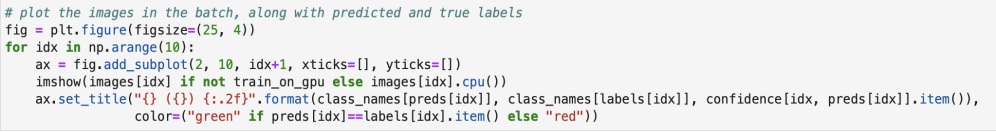


The final accuracy performance on test set:



Before I output the test results, I use the softmax function to calculate the confidence of each picture relating to all ten categories. Since the ‘torch.max’ function has already sifted out the most likely categories for each sample, confident value can be eventually found based on this maximum prediction value. The code changes are shown as follows:





The sample test results are as follows:



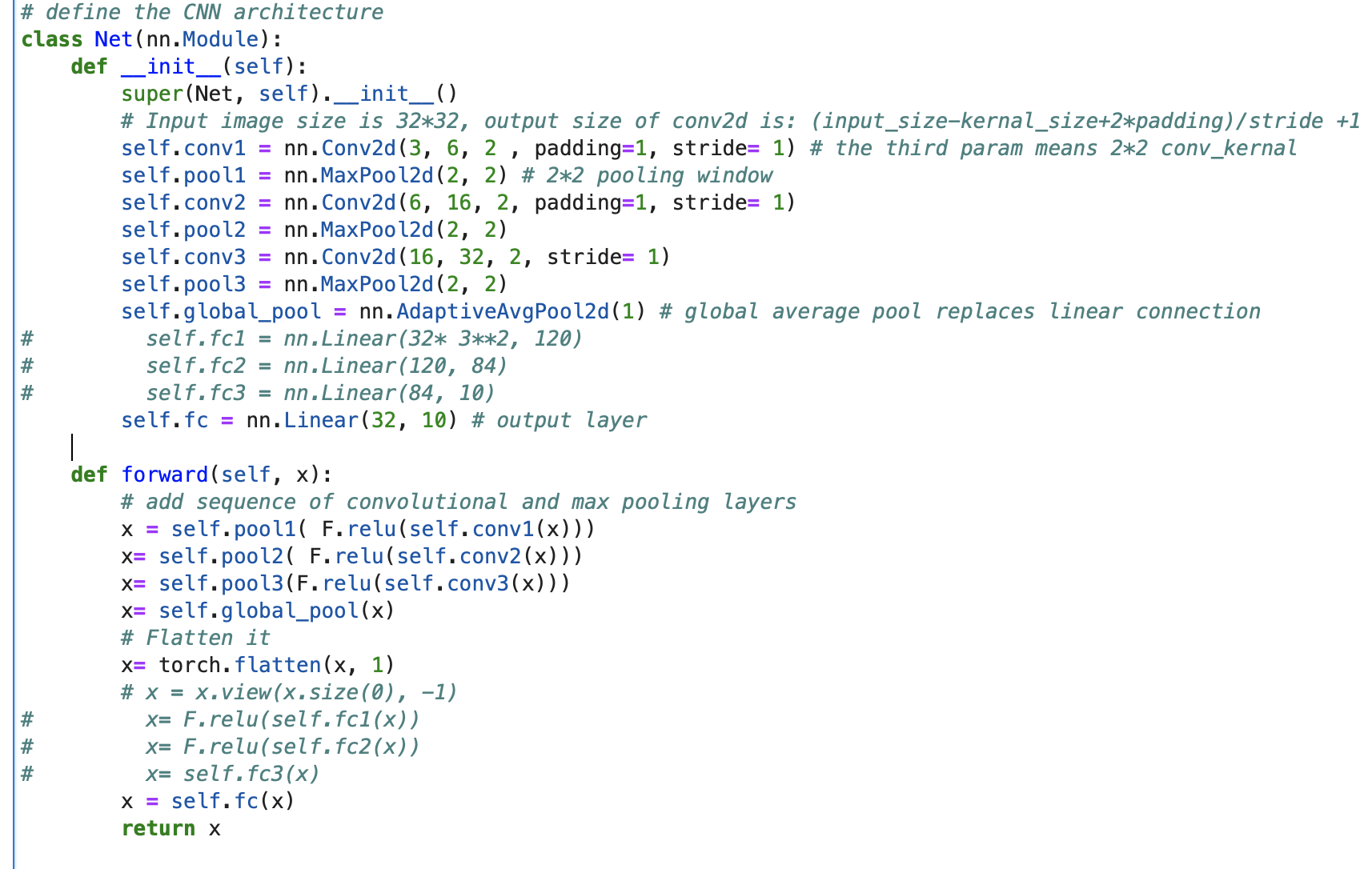
Q3

为了方便测试不同的方法，Q3的所有方案均只测试了30代。这是缺陷要在最后提及！

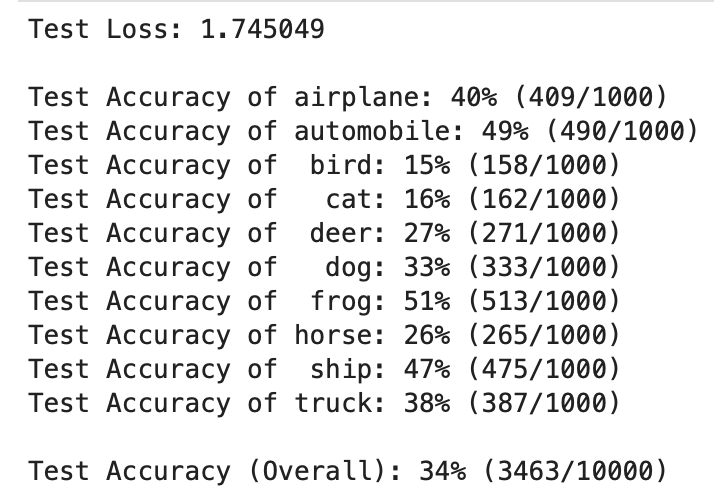
Method 1: Reduce one convolution layer and one corresponding maximum pooling layer

Method 2: Using global average pooling

Global average pooling is used instead of linear connection layer, which can effectively reduce the number of parameters in the model. At the same time, it can improve the speed of model training and is easier to avoid overfitting [4]. The codes are shown below:



The result of using this network is as follows:



Method 3: Using label smoothing

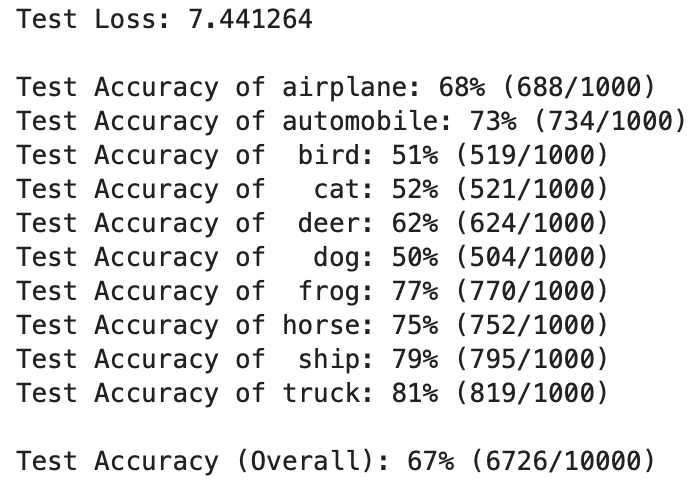
The advantage of label smoothing is that it can reduce the sensitivity of the model to noise and abnormal labels in the training data, helping the model to better generalize to previously unseen data [5].

The definition of Label Smoothing class is presented below:



其中的KL散度函数介绍见GPT

And the result of such loss function into the test set is:



Method 4: Changing optimizer to Adam and its parameter like learning rate and momentum

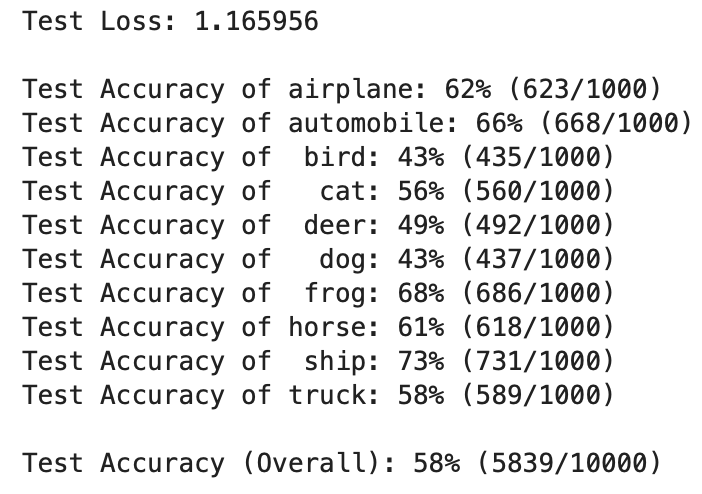
Adam优化器原理介绍见GPT

Adaptive Moment Estimation(Adam) optimization algorithm optimizes traditional optimization algorithms such as stochastic gradient descent by combining first-order and second-order estimation matrix. It is able to adjust the learning rate in an adaptive way, so that the model converges to a better solution faster [6].

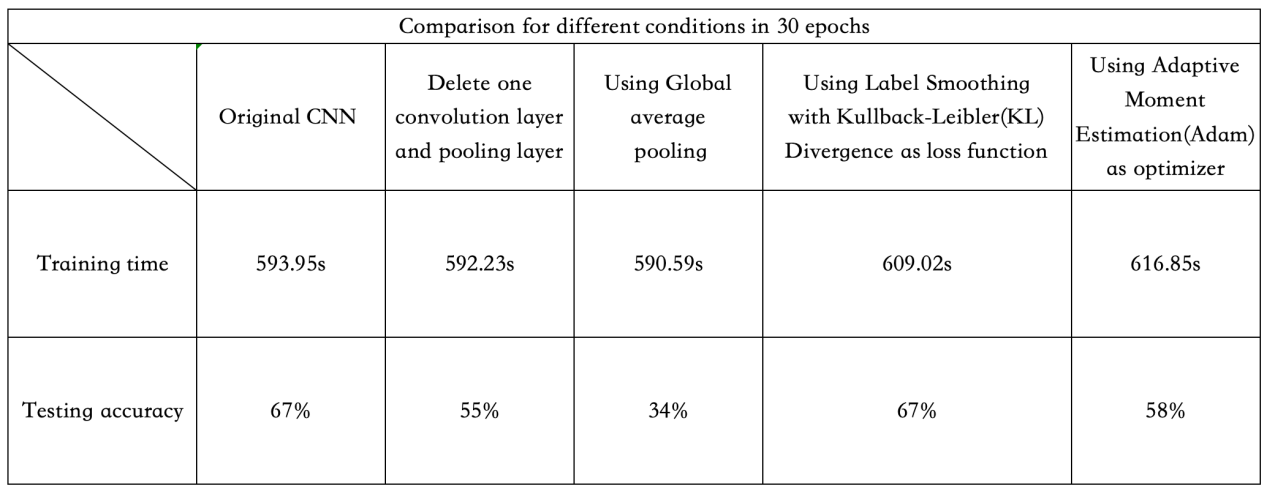
The rewritten code is shown below:



The testing result is:



Overall comparison relating to training time and accuracy:



Reference list

1. S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
2. Y. LeCun, Y. Bengio et al., “Convolutional networks for images, speech, and  
   time series,” The handbook of brain theory and neural networks, vol. 3361,  
   no. 10, p. 1995, 1995.
3. Q. Zhu, P. Zhang, Z. Wang and X. Ye, "A New Loss Function for CNN Classifier Based on Predefined Evenly-Distributed Class Centroids," in IEEE Access, vol. 8, pp. 10888-10895, 2020, doi: 10.1109/ACCESS.2019.2960065.
4. M. Lin, Q. Chen, and S. Yan, “Network in network,” 2014.
5. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking  
   the inception architecture for computer vision,” 2015.
6. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,”  
   2017.