Term Project for Final Exam

Name:리카이신 (Li Kaixin)

Student ID:2021227496

I. INTRODUCTION

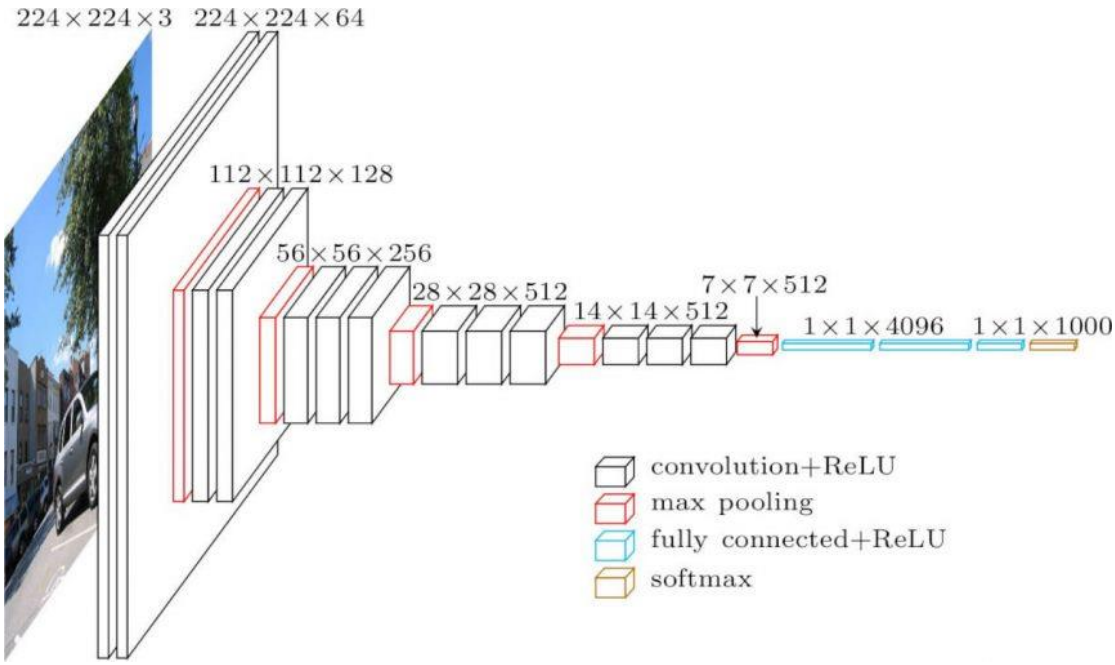
In this term project, I used VGG-16 model to achieve good performance on CIFAR-10 dataset. The rest of this report is organized as follows. I introduce the VGG16 model in Section II. In Section III, the technology used in the study and the parameters modified by me is discussed, and the result is given in Section IV.

II. VGG16 MODEL

The architecture depicted below is VGG16. VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition".

An improvement of VGG16 compared to AlexNet is to use consecutive 3x3 convolution kernels to replace the larger convolution kernels in AlexNet. For a given receptive field, using a stacked small convolution kernel is better than using a large convolution kernel, because multiple

non-linear layers can increase the depth of the network to ensure more complex learning The model, and the cost is relatively small (fewer parameters).



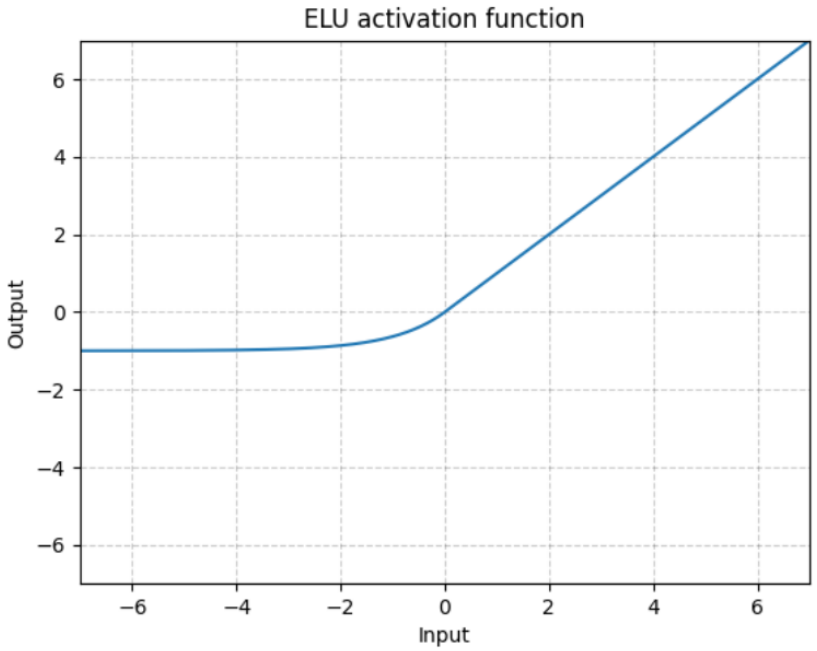
III. TECHNOLOGY AND PARAMETERS

During training, the input to our ConvNets is a fixed-size 32 × 32 RGB image. The image is passed through a stack of conv layers, where I use filters with a very small receptive field: 3 × 3.

The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv layers. Max-pooling is performed over a 2 × 2pixel window, with stride 2.

A stack of convolutional layers is followed by three Fully-Connected (FC) layers: the first one has 512*1*1 channels, the second one has 256 channels, the third performs 10-way classification and thus contains 10 channels. The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks, and all hidden layers are equipped with the rectification ELU non-linearity. The Exponential Linear Unit (ELU) is an activation function for neural networks. In contrast to ReLUs, ELUs have negative values which allows them to push mean unit activations closer to zero like batch normalization but with lower computational complexity.

$$\mathrm{ELU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha * (\exp(x) - 1), & \text{if } x \leq 0 \end{cases}$$



I also used cutout technology and adjusted learning rate. (Sets the learning rate to the initial LR decayed by 0.2 every steep step).

IV. RESULTS

According to the github provided by the professor, I first chose VGG, but the accuracy was only 89.5%. I changed VGG model to VGG16 model and changed some setting and these setting and parameters are as follow:

model	VGG16
framework	pytorch
learning rate	0.001
epoch	150
batch_size	128
transforms	RandomCrop, RandomHorizontalFlip, Normalize, cutout
optimizer	SGD(momentum = 0.9, weight_decay=1e-4)

Top – 1 classification rate of this model by these settings is 91.07%.

```
Epoch: 94 | Batch_idx: 380 |  Loss: (0.1954) | Acc: (93.31%) (45504/48768)
Epoch: 94 | Batch_idx: 390 |  Loss: (0.1951) | Acc: (93.34%) (46668/50000)
=> saving checkpoint
# TEST : Loss: (0.2921) | Acc: (91.07%) (9107/10000)
Epoch: 95 | Batch_idx: 0 |  Loss: (0.1672) | Acc: (92.19%) (118/128)
Epoch: 95 | Batch_idx: 10 |  Loss: (0.1623) | Acc: (94.11%) (1325/1408)
```

Through this course offered by the professor, I have mastered the basic knowledge about deep learning, and I trained some deep learning models and learned to modify some settings and parameters. I benefit a lot from the course. Thank you, professor.