COSE474 Deep Learning
Project 2 Report

컴퓨터학과 2022320006 최진혁

1. Description of Code

➢ Question 1

Using the functions conv1x1 and conv3x3, implement bottle neck building block (residual block). The bottle neck block consists of 3 convolution layers, and first and second convolution layer has same output channels. Thus, the block can be implemented as conv1x1 with in/out channel = in/middle, conv3x3 with in/out channel = middle/middle, and conv1x1 with in/out channel = middle/out. The downsample option is used, which determine whether the output size of the block will be downsized to floor of {(H-1)/2 + 1}, which is H/2, where H is input size. If the downsample is True, first layer has stride 2 and padding 0, as same as the downsize layer. Then the height and width of the feature map will be maintained using 3x3 convolution with stride 1 and padding 1, and 1x1 convolution with stride 1 and padding 0. If the downsample is False, first layer has stride 1, while the other things are the same as when it is True.

➢ Question 2

Using the residual block, implement ResNet50. The number of class of CIFAR-10 dataset is 10 and the data image size is 3x32x32. Layer 1 has 7x7 convolution with channel 64 and stride 2. The image size is halved this layer, so padding is 3. Thus the layer 1 has nn.Conv2d(3, 64, 7, 2, 3). Layer 1 also has 3x3 max pooling with stride 2. The output size of layer 1 is 8x8, so padding is 1 and nn.Maxpool2d(3, 2, 1).

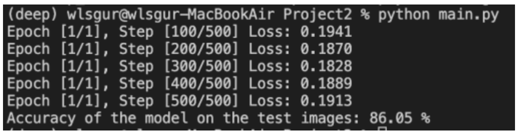
Layer 2 consists of three residual blocks. The output of the layer 1 has 64 channel, so input channel of first block is 64. They all have middle channel 64 and output channel 256, and second, third block has input channel 256, which is same as previous block's output. Last block makeas the feature map half, so the downsample is True.

Layer 3 consists of four residual blocks. The output of the layer 2 has 256 channel, so input channel of first block is 256. They all have middle channel 128 and output channel 512. Like the layer 2, last block performs downsampling, so last parameter is True.

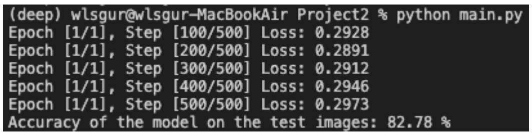
Layer 4 consists of 6 residual blocks. Input channel of first block is 512. They all have middle channel 256 and output channel 1024. In this layer, last block will not perform downsampling.

The FC layer will mapping 1x1x1024 feature map to 10 class score, thus the FC layer is nn.Linear(1024, 10). By average pooling, the feature map size will be halved, thus there is a nn.AvgPool2d(2, 2) with 2x2 kernel and stride 2.

2. Result and Discussions



```
(deep) wlsgur@wlsgur-MacBookAir Project2 % python main.py
Epoch [1/1], Step [100/500] Loss: 0.1941
Epoch [1/1], Step [200/500] Loss: 0.1870
Epoch [1/1], Step [300/500] Loss: 0.1828
Epoch [1/1], Step [400/500] Loss: 0.1889
Epoch [1/1], Step [500/500] Loss: 0.1913
Accuracy of the model on the test images: 86.05 %
```



```
(deep) wlsgur@wlsgur-MacBookAir Project2 % python main.py
Epoch [1/1], Step [100/500] Loss: 0.2928
Epoch [1/1], Step [200/500] Loss: 0.2891
Epoch [1/1], Step [300/500] Loss: 0.2912
Epoch [1/1], Step [400/500] Loss: 0.2946
Epoch [1/1], Step [500/500] Loss: 0.2973
Accuracy of the model on the test images: 82.78 %
```

Left is the test accuracy of VGG16, and Right is that of ResNet50. VGG16 has accuracy 86.05%, and ResNet50 has 82.78%. In both cases, loss is not decreasing well. I think this is because we use the model checkpoint already trained 250 and 285 epochs, and thus current point has small gradient. The accuracy is not very high. Thus if we use larger learning rate, we could get better performance maybe.