

Final Report

Name: ZhaoKe (짜오커)

Students Number: 2021223529

ResNet:

First, I used the traditional ResNet architecture and followed the codes that professor gave in github. The accuracy of this architecture was about 89.6%.

The parameters of this architecture I chosen:

Optimizer: SGD

Learning rate =0.01(after 80 comes to 0.001, after 120 comes to 0.0001)

.....

What I changed:

1. Change the optimizer from SGD to Adam

Keep other parameters unchanged, only change the optimizer from SGD to Adam, the accuracy rate has been increased from 89.6% to 92.6%.

The parameters of this architecture:

Optimizer: Adam

Learning rate =0.01(after 80 comes to 0.001, after 120 comes to 0.0001)

eps=1e-08,

weight decay=1e-4

2. Change the optimizer and layers

Keep other parameters unchanged, change optimizer from SGD to Adam and change the layers:

```
def forward(self, x):
    out = F.relu(self.bn1(self.conv1(x)))
    out = self.layer1(out)
    out = self.layer2(out)
    out = self.layer3(out)
    out = self.layer4(out)
    out = F.avg_pool2d(out, 4)
    out = out.view(out.size(0), -1)
    out = self.linear(out)
    return out
```

from

```
def forward(self, x):
    out = F.relu(self.bn1(self.conv1(x)))
    out = F.avg_pool2d(out, 2)
    out = self.layer1(out)
    out = self.layer2(out)
    out = self.layer3(out)
    out = self.layer4(out)
    out = F.avg_pool2d(out, 2)
    out = out.view(out.size(0), -1)
    out = self.linear(out)
    return out
```

to

This makes the training speed faster, but the accuracy is not good, it drops to about 87%.

And I also changed type of pool from the avg_pool to the max_pool, The results show that on this data set, avg_pool is better than max_pool.

3. Change the optimizer from SGD to NAdam

Keep other parameters unchanged, only change the optimizer from SGD to NAdam, The training speed is much slower than Adam, and it is difficult for me to complete it on colab in a short time.

But I found in some articles that when I want to use the momentum-driven RMSprop, or Adam, NAdam can be used to achieve better results.

I believe that using NAdam training in sufficient time, the result will be higher than 92.6%.

ResNet50:

I used the pre-trained Resnet50 from torchvision.models to solve this.

The parameters of this architecture I chosen:

Optimizer: SGD

Learning rate = 0.01

momentum= 0.9

After ResNet50 training 5 epochs, the result increased to **95.2%** much better than traditional ResNet.

What I changed:

1. Change the optimizer

Keep other parameters unchanged, only change the optimizer from SGD to **Adam**, accuracy rate **dropped a lot**.

It may be due to the use of the SGD optimizer in the pre-training process, or Adam is not suitable for ResNet50.

2. **BiT**

But the latest ResNet-based architecture, called **BiT**, is much better than ResNet50. I tried to train the code on colab and found that it can reach **97.8%**.

```
10/10 [=====] - 24s 2s/step - loss: 0.0230 - accuracy: 0.9920 - val_loss: 0.0668 - val_accuracy: 0.9782
Epoch 49/50
10/10 [=====] - 23s 2s/step - loss: 0.0206 - accuracy: 0.9953 - val_loss: 0.0668 - val_accuracy: 0.9782
Epoch 50/50
10/10 [=====] - 24s 2s/step - loss: 0.0266 - accuracy: 0.9918 - val_loss: 0.0668 - val_accuracy: 0.9782
```