Project 2 Report

Sibei Zhou(301416917)

Submission name on Kaggle: Sibei Zhou

Team name: WOW

Best accuracy: **64.6%**

Part 1

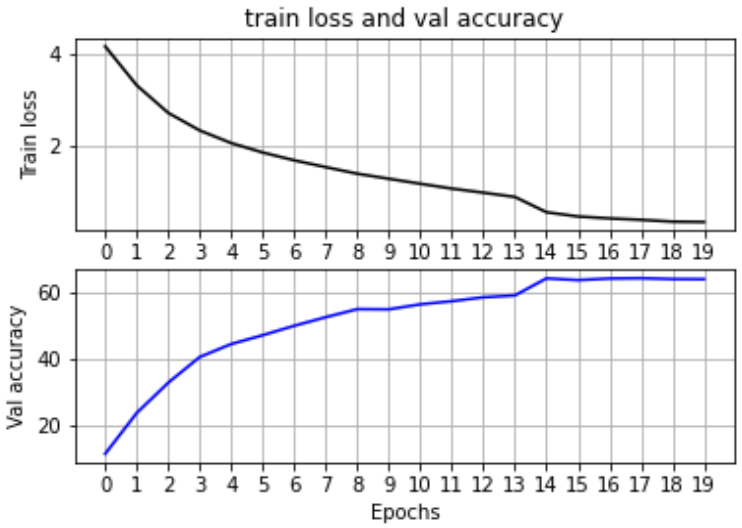
Layer Structure Table (The color of the table is to better distinguish the groups of the layers, like the convolutional modules(conv-relu-maxpool))

Layer No.	Layer Type	Kernel size	Input|Output Dimension	Input|Output Channels
1	Conv2d	3	32|32	3|64
2	Batchnorm	-	32|32	-
3	Relu	-	32|32	-
4	Conv2d	3	32|32	64|64
5	Batchnorm	-	32|32	-
6	Relu	-	32|32	-
7	Conv2d	3	32|32	64|128
8	Batchnorm	-	32|32	-
9	Relu	-	32|32	-

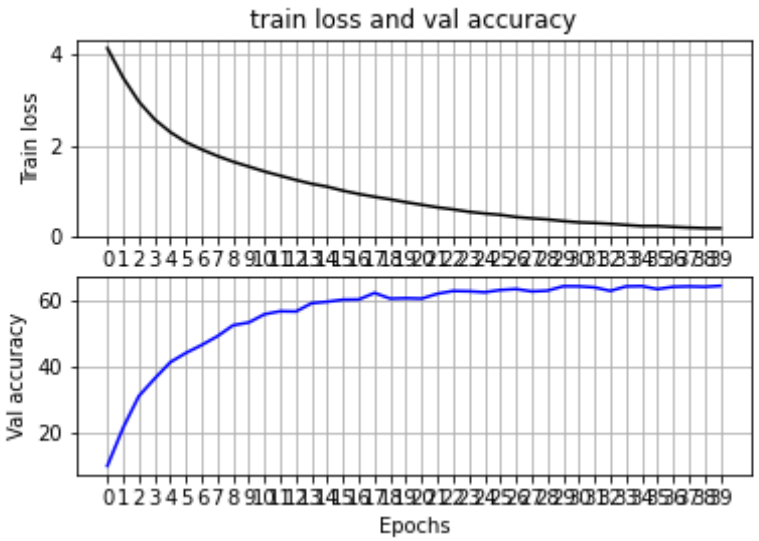
10	Conv2d	3	32|32	128|128
11	Batchnorm	-	32|32	-
12	Relu	-	32|32	-
13	Maxpool2d	2	32|16	-
14	Conv2d	3	16|16	128|192
15	Batchnorm	-	16|16	-
16	Relu	-	16|16	-
17	Conv2d	3	16|16	192|384
18	Batchnorm	-	16|16	-
19	Relu	-	16|16	-
20	Maxpool2d	2	16|8	-
21	Conv2d	3	8|8	384|256
22	Batchnorm	-	8|8	-
23	Relu	-	8|8	-
24	Conv2d	3	8|8	256|256
25	Batchnorm	-	8|8	-
26	Relu	-	8|8	-

27	Maxpool2d	2	8|4	-
28	Linear	-	4096|2048	-
29	Relu	-	2048|2048	-
30	Linear	-	2048|1024	-
31	Relu	-	1024|1024	-
32	Dropout(p=0.2)		1024|1024	
33	Linear	-	1024|512	-
34	Relu	-	512|512	-
35	Linear	-	512|216	-
36	Relu	-	216|216	-
37	Linear	-	216|100	-

Result plot (epoch = 20)



Result plot (epoch = 40)



We can see in the graph that even through the train loss keep decrease, the val accuracy only slightly improved after rising to 60%, and then stabilized on the last 10 epochs.

Structure Explain

Part of the structure is referenced from the AlexNet model, but many modifications have been made to make the net tailored for picture size 32*32.

First, the number of convolution layers is increased to 8 from original AlexNet's 5 convolution layers, with increasing output channels, which prevents the net from overfitting and reduces the training accuracy.

Then, a 2d batch normalizing layer is added before the RELU layer of each convolution layer, and Pooling layers with 2 kernel size and stride = 1 are used instead, and we put it after the 4th, 6th, 8th convolution layer.

Moreover, one more fully connected layer after AlexNet.

Ablation Study

1. Adding 3 more convolution layers with 32,64,128,256 output channels and change all convolution layers' kernel size to 3: accuracy increase **from 26.3% to 35.8%**

2. Batching normalizing layer added after each convolution layer, and Relu layers after it: accuracy increase **from 35.8% to 38%**

3. Adapting AlexNet structure, with 5 convolution layers + Relu laters, 3 pooling layers(after the 1st, 2nd, 5th convolution layer), and 2 linear layers + Relu + Dropout(p=0.5), and the final linear layer which return the result label between 100 (in this step, since the original AlexNet use convolution layer with kernel size = 11, it is too big and reduce the output dimension too fast, and lead to failure, so we change all the size of the kernel to 3): accuracy increase **from 35.8% to 43.4%**

4. Change the potion of the 3 pooling layers and change the convolution layers' filter size: accuracy increase **from 43.4% to 62.4%**

5. Increase the epoch from 20 to 60: accuracy increase **from 62.4% to 64.6%**

Part 2

Model and Hyperparameters

The first time I just use the pre-trainning renet18 without changing the given hyperparameters, I achieve a train accuracy of 14.9%:

```
TRAINING Epoch 1/10 Loss 0.6792 Accuracy 0.0043
TRAINING Epoch 2/10 Loss 0.6617 Accuracy 0.0110
TRAINING Epoch 3/10 Loss 0.6498 Accuracy 0.0180
TRAINING Epoch 4/10 Loss 0.6382 Accuracy 0.0273
TRAINING Epoch 5/10 Loss 0.6279 Accuracy 0.0460
TRAINING Epoch 6/10 Loss 0.6181 Accuracy 0.0547
TRAINING Epoch 7/10 Loss 0.6086 Accuracy 0.0780
TRAINING Epoch 8/10 Loss 0.5989 Accuracy 0.0953
TRAINING Epoch 9/10 Loss 0.5893 Accuracy 0.1220
TRAINING Epoch 10/10 Loss 0.5797 Accuracy 0.1493
Finished Training
```

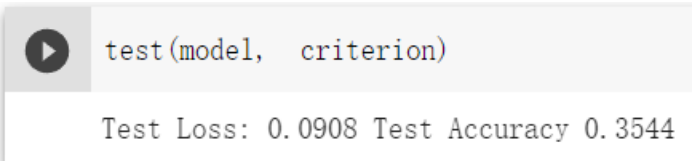
1. First try on slight tweaks to those hyperparameters, that is:

```
NUM_EPOCHS = 50

LEARNING_RATE = 0.001

BATCH_SIZE = 32

RESNET_LAST_ONLY = True
```

I get a train accuracy 53.67%.

```
TRAINING Epoch 40/50 Loss 0.0846 Accuracy 0.5157
TRAINING Epoch 41/50 Loss 0.0836 Accuracy 0.5297
TRAINING Epoch 42/50 Loss 0.0824 Accuracy 0.5313
TRAINING Epoch 43/50 Loss 0.0821 Accuracy 0.5260
TRAINING Epoch 44/50 Loss 0.0821 Accuracy 0.5197
TRAINING Epoch 45/50 Loss 0.0807 Accuracy 0.5217
TRAINING Epoch 46/50 Loss 0.0810 Accuracy 0.5230
TRAINING Epoch 47/50 Loss 0.0793 Accuracy 0.5373
TRAINING Epoch 48/50 Loss 0.0787 Accuracy 0.5303
TRAINING Epoch 49/50 Loss 0.0788 Accuracy 0.5367
TRAINING Epoch 50/50 Loss 0.0781 Accuracy 0.5367
Finished Training
```

And test accuracy at 35.44%:



2. Second try on slight tweaks to those hyperparameters, that is:

```
NUM_EPOCHS = 50

LEARNING_RATE = 0.001

BATCH_SIZE = 32

RESNET_LAST_ONLY = False
```

```
TRAINING Epoch 40/50 Loss 0.0317 Accuracy 0.8053
TRAINING Epoch 41/50 Loss 0.0310 Accuracy 0.8093
TRAINING Epoch 42/50 Loss 0.0296 Accuracy 0.8180
TRAINING Epoch 43/50 Loss 0.0295 Accuracy 0.8213
TRAINING Epoch 44/50 Loss 0.0284 Accuracy 0.8263
TRAINING Epoch 45/50 Loss 0.0282 Accuracy 0.8203
TRAINING Epoch 46/50 Loss 0.0282 Accuracy 0.8247
TRAINING Epoch 47/50 Loss 0.0265 Accuracy 0.8263
TRAINING Epoch 48/50 Loss 0.0265 Accuracy 0.8400
TRAINING Epoch 49/50 Loss 0.0257 Accuracy 0.8450
TRAINING Epoch 50/50 Loss 0.0266 Accuracy 0.8427
Finished Training
```

I get a train accuracy 84.27%.

And test accuracy at 56.78%:

```
[27] test(model, criterion)
```

```
Test Loss: 0.0518 Test Accuracy 0.5678
```