CMPT412

Xiaohong Xu 301417289

Lab 2 Report

Group

group name: **Sky Cloud**
team member name: **Hao Sun, Heyuan Wang, Xiaohong Xu.**

Network design description

We modified the VGG network, and used 8 convolution layers and 4 max pooling layers with kernel size 3 to design our network architecture.

Network Architecture

Layer No.	Layer Type	Kernel size (conv)	Input | Output dimension	Input | Output channels (conv)
1	conv2d	3	32|32	3|64
2	relu	-	32|32	-
3	conv2d	3	32|32	64|128
4	relu	-	32|32	-
5	maxpool2d	2	32|16	-
6	conv2d	3	16|16	128|256
7	relu	-	16|16	-
8	conv2d	3	16|16	256|256
9	relu	-	16|16	-
10	maxpool2d	2	16|8	-
11	conv2d	3	8|8	256|512
12	relu	-	8|8	-
13	conv2d	3	8|8	512|512
14	relu	-	8|8	-
15	maxpool2d	2	8|4	-
16	conv2d	3	4|4	512|512
17	relu	-	4|4	-
18	conv2d	3	4|4	512|512

19	relu	-	4|4	-
20	maxpool2d	2	4|2	-

21	linear	-	2048|4096	-
22	relu	-	4096|4096	-
23	linear	-	4096|4096	-
24	relu	-	4096|4096	-
25	linear	-	4096|4096	-

Ablation study

The first time I tried the architecture by following the instructions and added 2 layers of conv and max pooling, the accuracy rate was only 20%-25%. So I found that I had to learn a new model. By learning Convolutional Neural Networks (CNNs / ConvNets),we found that VGG16 is a good choice.[1]

We learned and tried the VGG-16 architecture at first, but the accuracy is not good enough, our accuracy is about 60% - 62%.

Because the image size is 32x32, it is small. We consider **removing a max pool and 2 conv layers**. and use kernel size 3 filter and padding 1 to make sure the size will not change.

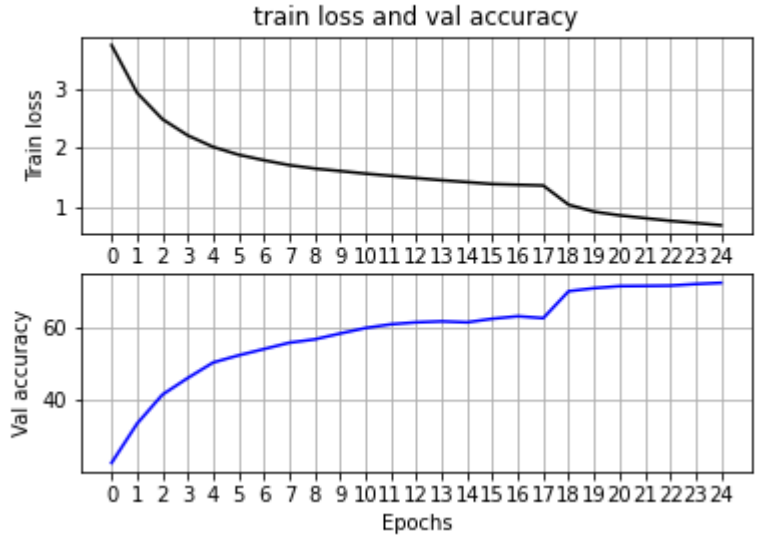
After we remove the max pool layer and 2 conv layer, our accuracy is increased to 71%.
for my personal architecture accuracy is about 65% while using 25 EPOCHS.
and then we tried to adjust some other conv and maxpool layers day by day, the **best accuracy** for my team is **75.8% which is uploaded on kaggle**, is 8 convolution layers and 4 max pooling layers.

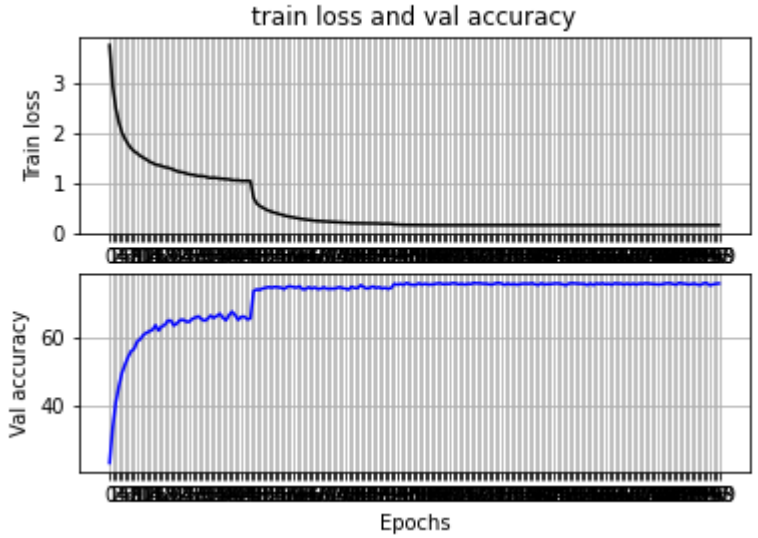
the accuracy show below is run on Colab for my teammate

```
[1] loss: 3.748
Accuracy of the network on the val images: 22 %
[2] loss: 2.939
Accuracy of the network on the val images: 33 %
[3] loss: 2.488
Accuracy of the network on the val images: 41 %
[4] loss: 2.211
Accuracy of the network on the val images: 46 %
[5] loss: 2.014
Accuracy of the network on the val images: 50 %
[6] loss: 1.882
Accuracy of the network on the val images: 52 %
[7] loss: 1.788
Accuracy of the network on the val images: 54 %
[8] loss: 1.704
Accuracy of the network on the val images: 55 %
[9] loss: 1.646
Accuracy of the network on the val images: 56 %
[10] loss: 1.604
Accuracy of the network on the val images: 58 %
[11] loss: 1.559
Accuracy of the network on the val images: 59 %
[12] loss: 1.521
Accuracy of the network on the val images: 60 %
[13] loss: 1.484
...
Accuracy of the network on the val images: 71 %
[25] loss: 0.679
Accuracy of the network on the val images: 72 %
Finished Training
```

illustrating the training loss and the validation accuracy

below is the diagram of our performance vs deeper net improved performance





Part 2

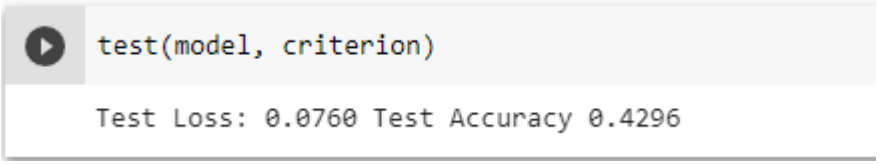
Model and Hyperparameters

1)
batch_size = 64 ,learning_rate = 0.001, **num_epochs** = 50, **resnet_last_only** = true

Training accuracy for resnet_last_only = true (fixed feature extractor)

```
TRAINING Epoch 37/50 Loss 0.0381 Accuracy 0.8940
TRAINING Epoch 38/50 Loss 0.0374 Accuracy 0.8933
TRAINING Epoch 39/50 Loss 0.0361 Accuracy 0.8990
TRAINING Epoch 40/50 Loss 0.0353 Accuracy 0.9043
TRAINING Epoch 41/50 Loss 0.0347 Accuracy 0.9063
TRAINING Epoch 42/50 Loss 0.0335 Accuracy 0.9140
TRAINING Epoch 43/50 Loss 0.0329 Accuracy 0.9117
TRAINING Epoch 44/50 Loss 0.0319 Accuracy 0.9187
TRAINING Epoch 45/50 Loss 0.0316 Accuracy 0.9180
TRAINING Epoch 46/50 Loss 0.0310 Accuracy 0.9203
TRAINING Epoch 47/50 Loss 0.0298 Accuracy 0.9237
TRAINING Epoch 48/50 Loss 0.0293 Accuracy 0.9297
TRAINING Epoch 49/50 Loss 0.0285 Accuracy 0.9310
TRAINING Epoch 50/50 Loss 0.0283 Accuracy 0.9333
Finished Training
----------
```

Test accuracy for resnet_last_only = true (fixed feature extractor)



2)
batch_size = 64 ,learning_rate = 0.001, **num_epochs** = 20, **resnet_last_only** = false

Training accuracy for resnet_last_only = false(fine tune)

```
TRAINING Epoch 8/20 Loss 0.0523 Accuracy 0.8987
TRAINING Epoch 9/20 Loss 0.0436 Accuracy 0.9363
TRAINING Epoch 10/20 Loss 0.0365 Accuracy 0.9500
TRAINING Epoch 11/20 Loss 0.0300 Accuracy 0.9743
TRAINING Epoch 12/20 Loss 0.0248 Accuracy 0.9840
TRAINING Epoch 13/20 Loss 0.0206 Accuracy 0.9900
TRAINING Epoch 14/20 Loss 0.0170 Accuracy 0.9937
TRAINING Epoch 15/20 Loss 0.0139 Accuracy 0.9960
TRAINING Epoch 16/20 Loss 0.0119 Accuracy 0.9963
TRAINING Epoch 17/20 Loss 0.0101 Accuracy 0.9963
TRAINING Epoch 18/20 Loss 0.0086 Accuracy 0.9980
TRAINING Epoch 19/20 Loss 0.0075 Accuracy 0.9970
TRAINING Epoch 20/20 Loss 0.0066 Accuracy 0.9977
Finished Training
```

Test accuracy for resnet_last_only = false(fine tune)

```
test(model, criterion)
```

```
Test Loss: 0.0667 Test Accuracy 0.5196
```

Reference

[1] Convolutional Neural Networks (CNNs / ConvNets)
http://cs231n.github.io/convolutional-networks/#layers