**Project 2**

**Deep learning by PyTorch**

Ruixuan Liu

301383988

Part 1

Team Name: Ultraman PyTorch

Team member:

Zhihao Zhao 301416908

Lingxuan Fan 301384783

In part1, we borrowed and modified the bottleneck block from the resent structure. we added a total of 3 bottleneck blocks inside 12 convolutional layers. They are used on the fifth convolutional layer, the ninth convolutional layer and the tenth convolutional layer respectively. We finally used 4 linear layers to reduce the output channel to 100.For image processing, the normalized range was changed to [-1,1] using mean and standard deviation equal to 0.5 on all three channels.

The figure below shows the structure of our network model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Layer No. | Layer Type | Kernel Size  (for Conv Layer) | Inoput|Ouput  dimension | Input|Output  Channels  (for Conv Layer) |
| 1 | Conv2d | 3 | 32|32 | 3|64 |
| 2 | Bn | - | 32|32 |  |
| 3 | Relu | - | 32|32 |  |
| 4 | Conv2d | 3 | 32|32 | 64|128 |
| 5 | Bn | - | 32|32 |  |
| 6 | Relu | - | 32|32 |  |
| 7 | Conv2d | 3 | 32|32 | 128|256 |
| 8 | Bn | - | 32|32 |  |
| 9 | Relu | - | 32|32 |  |
| 10 | Conv2d | 3 | 32|32 | 256|512 |
| 11 | Bn | - | 32|32 |  |
| 12 | Relu | - | 32|32 |  |
| 13 | Maxpool2d | 2 | 32|16 |  |
| 14 | Conv2d | 3 | 16|16 | 512|256 |
| 15 | Bn | - | 16|16 |  |
| 16 | Relu | - | 16|16 |  |
| 17 | Conv2d | 3 | 16|16 | 256|512 |
| 18 | Bn | - | 16|16 |  |
| 19 | Relu | - | 16|16 |  |
| 20 | Conv2d | 3 | 16|16 | 512|1024 |
| 21 | Bn | - | 16|16 |  |
| 22 | Relu | - | 16|16 |  |
| 23 | Conv2d | 3 | 16|16 | 1024|512 |
| 24 | Bn | - | 16|16 |  |
| 25 | Relu | - | 16|16 |  |
| 26 | Maxpool2d | 2 | 16|8 |  |
| 27 | Conv2d | 3 | 8|8 | 512|1024 |
| 28 | Bn | - | 8|8 |  |
| 29 | Relu | - | 8|8 |  |
| 30 | Conv2d | 3 | 8|8 | 1024|512 |
| 31 | Bn | - | 8|8 |  |
| 32 | Relu | - | 8|8 |  |
| 33 | Conv2d | 3 | 8|8 | 512|1024 |
| 34 | Bn | - | 8|8 |  |
| 35 | Relu | - | 8|8 |  |
| 36 | Conv2d | 3 | 8|8 | 1024|2048 |
| 37 | Bn | - | 8|8 |  |
| 38 | Relu | - | 8|8 |  |
| 39 | Linear | - | 2048\*8\*8|1024 |  |
| 40 | Bn1d | - | 1024|1024 |  |
| 41 | Relu | - | 1024|1024 |  |
| 42 | Linear | - | 1024|512 |  |
| 43 | Bn1d | - | 512|512 |  |
| 44 | Relu | - | 512|512 |  |
| 45 | Linear | - | 512|256 |  |
| 46 | Bn1d | - | 256|256 |  |
| 47 | Relu | - | 256|256 |  |
| 48 | Linear | - | 256|100 |  |

We ran 100 epochs and 300 epochs respectively using this network structure model. it is clear to see that overfitting occurs when the accuracy runs up to roughly 70%, and although the loss is still decreasing, the accuracy does not improve much.

A graph of a train loss and a normal curve

Description automatically generated

A graph of a train loss and a graph of a train loss

Description automatically generated

A screenshot of a computer

Description automatically generated

Ablation Study: We only trained until the accuracy reached 63.9% on Kaggle in the first training, when we used only 5 convolutional layers. For the second training, we boosted the convolutional layers to 9 layers and the accuracy increased to 70.9% on Kaggle. We finally boosted the convolutional layers to 12 layers and the accuracy reached up to 71.7%. Additionally, we experimented with boosting to 15 convolutional layers, but the accuracy did not improve with the increase in convolutional layers.

Part 2

ResNet as a fixed feature extractor:

A table of training results

Description automatically generated with medium confidence

A close-up of a computer screen

Description automatically generated

Fine-tuning the ResNet:

I changed the hyper parameter.

A math equation with numbers and symbols

Description automatically generated

Using learning rate as 0.01:

A table of numbers and letters

Description automatically generated with medium confidence

A close-up of a test

Description automatically generated

Using learning rate as 0.0001:

A table of numbers with text

Description automatically generated with medium confidence

A close-up of a test

Description automatically generated

The best performance learning rate is 0.001:

A table of training results

Description automatically generated with medium confidence

A close up of a number

Description automatically generated