CMPT 762

Computer Vision

Assignment 2

Gokul Mohanarangan

gma56@sfu.ca

301436162

Free-Late day: 3 late days given exclusively for

Assignment 2 are intended to be used here.

1 free late day from the pool of 5 free late days is also used here.

The name on Kaggle : Naruzu

Best accuracy : 0.68500

**BASENET ARCHITECTURE:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Layer number | Layer type | Kernel size  (For conv layers) | Input | Output  Dimensions | Input | Output  Channels  (For conv layers) |
| 1 | Conv2d | 3 | 32 | 32 | 3 | 64 |
| 2 | BatchNorm2d |  | 32 | 32 |  |
| 3 | Relu |  | 32 | 32 |  |
| 4 | Conv2d | 3 | 32 | 32 | 64 | 128 |
| 5 | BatchNorm2d |  | 32 | 32 |  |
| 6 | Relu |  | 32 | 32 |  |
| 7 | Pool |  | 32 | 16 |  |
| 8 | Conv2d | 3 | 16 | 16 | 128 | 128 |
| 9 | BatchNorm2d |  | 16 | 16 |  |
| 10 | Relu |  | 16 | 16 |  |
| 11 | Conv2d | 3 | 16 | 16 | 128 | 128 |
| 12 | BatchNorm2d |  | 16 | 16 |  |
| 13 | Relu |  | 16 | 16 |  |
| 14 | Conv2d | 3 | 16 | 16 | 128 | 256 |
| 15 | BatchNorm2d |  | 16 | 16 |  |
| 16 | Relu |  | 16 | 16 |  |
| 17 | Pool |  | 16 | 8 |  |
| 18 | Conv2d | 3 | 8 | 8 | 256 | 512 |
| 19 | BatchNorm2d |  | 8 | 8 |  |
| 20 | Relu |  | 8 | 8 |  |
| 21 | Conv2d | 3 | 8 | 8 | 512 | 512 |
| 22 | BatchNorm2d |  | 8 | 8 |  |
| 23 | Relu |  | 8 | 8 |  |
| 24 | Conv2d | 3 | 8 | 8 | 512 | 512 |
| 25 | BatchNorm2d |  | 8 | 8 |  |
| 26 | Relu |  | 8 | 8 |  |
| 27 | Pool |  | 8 | 4 |  |
| 28 | Conv2d | 3 | 4 | 4 | 512 | 1024 |
| 29 | BatchNorm2d |  | 4 | 4 |  |
| 30 | Relu |  | 4 | 4 |  |
| 31 | Conv2d | 3 | 4 | 4 | 1024 | 1024 |
| 32 | BatchNorm2d |  | 4 | 4 |  |
| 33 | Relu |  | 4 | 4 |  |
| 34 | Conv2d | 3 | 4 | 4 | 1024 | 1024 |
| 35 | BatchNorm2d |  | 4 | 4 |  |
| 36 | Relu |  | 4 | 4 |  |
| 37 | Linear (FC) |  | 1024 \* 4 \* 4| 100 |  |
| 38 | BatchNorm1d |  | 100 |  |
| 39 | Relu |  | 100 |  |
| 40 | Linear (FC) |  | 100 | 100 |  |
| 41 | BatchNorm1d |  | 100 |  |

**MODEL OVERVIEW:**

To decrease the variance of the model, many data augmentations have been performed as part of the training process. Example : Random resized crop, random horizontal flip, random rotation have been included in the pipeline. The required mean and std values for the entire dataset has been calculated and all data has been normalized before model building. The model has 2 main components to it, the CONV-RELU-POOL and the linear fully connected layers. Every CONV layer is also followed by a BatchNorm layer for normalization and helping the training better. The conv layer has been padded to not reduce dimensions at every step and this enables the model to be deeper. Not every conv layer is followed by a pool layer to prevent loss of information, especially in the deeper layers. Hyperparameters like learning rate, epochs, momentum, and batch sizes have been experimented on. All kernel sizes are fixed at 3 and at every conv layer, the number of channels are increased, usually by a factor of 2. Some residual layers have been mixed in which have the same input and output channels but the previous layer being added to the next layer. This enables the model to ‘retain’ some information from the previous layers.

Best validation accuracy : 68%

Best testing accuracy on Kaggle : 0.68500

**Loss plot and validation accuracy plot:**

**Chart

Description automatically generated**

**Ablation study:**

Model edits :

1. Add residual layers. Since the model does not have a means to “remember” the previous layers, adding residual layers where the output of a certain layer is also added with the a previous layer output mimics the retention capabilities of the model and increases model accuracy.
2. Adding more conv layers. To create a deeper network to better capture the features, several new conv layers were added. Some with same input/output channels to enable residual layers.
3. It also means that conv layers have to be padded so that we don’t lose the dimensions too quickly, thus enabling us to create deeper networks. The padding has been set to 1 from all conv layers.

Before making these series of changes, the accuracy was validation accuracy was 61% and the test accuracy was 0.615. After making the series of changes, the validation accuracy was 68% and the test accuracy was 0.68500. The test accuracies can be verified on Kaggle. To address memory needs, the batch size was also reduced to 32 but it did not significantly affect the accuracies that much, but was kept for ease of computation when memory was occupied in the GPU RAM.

**PART 2 :**

**HYPERPARAMETERS :**

**Fine tune whole network :**

Batch size : 16

Epochs : 50

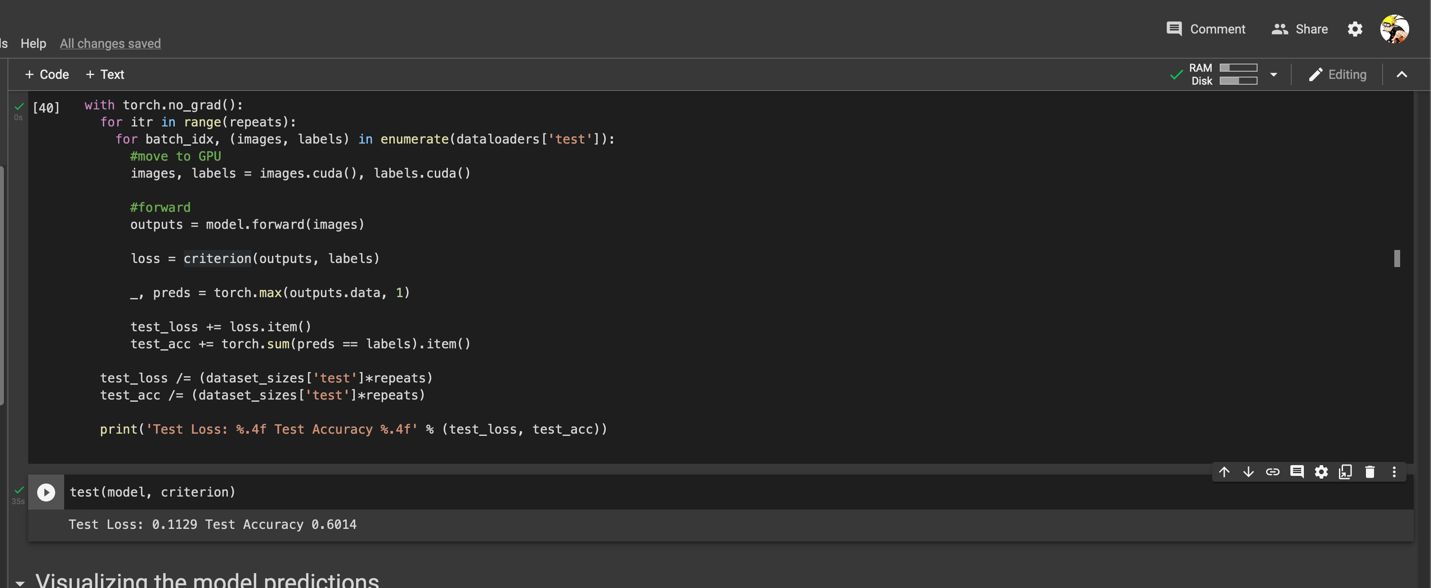
Learning rate : 0.001

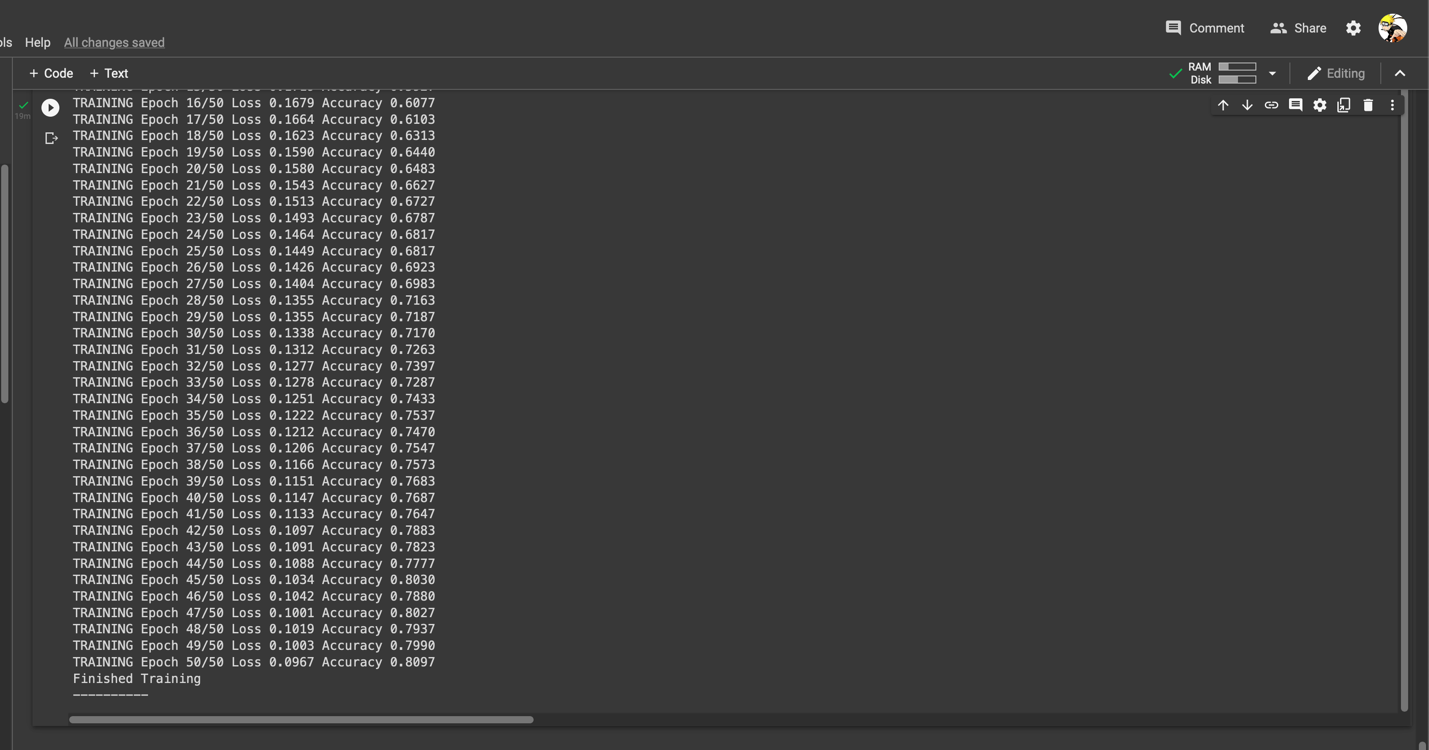
Momentum : 0.9

Restnet last only : False

Training accuracy : 0.8097

Test accuracy : 0.6014





**[The GPU quota ended and I had to switch to a local machine for fixed feature extraction part.]**

**Fixed feature extraction:**

Batch size : 16

Epochs : 50

Learning rate : 0.001

Momentum : 0.9

Restnet last only : True

Training accuracy : .05417

Test accuracy : 0.3864

Graphical user interface, text, application

Description automatically generated

