**CMPT 412**

**Computational Vision**

**Report- Project2**

***Image Classification and PyTorch Introduction***

**Kaggle**  **Best accuracy** – 67.7 %

**Part 1: Improving BaseNet on CIFAR100**

In the first part of the project, it is asked to improve the deep net by architectures like Data Normalization, Data augmentation, Deeper Network, adding Normalization Layers, and setting the values of Epochs.

Below is the result of the final network. This is a 27 layered model including 6 conv layers, 2 Maxpool, 3 linear, 8 ReLU, and the rest 8 are batchnorms.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Layer No. | Layer Type | Kernel size  (for conv layers) | Input | Output Dimension | Input | Output Channels |
| 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 | conv2d | 3 | 32 | 32 | 128 | 192 |
| 8 | Batchnorm2d | - | 32 | 32 | - |
| 9 | ReLU | - | 32 | 32 | - |
| 10 | Maxpool2d | 2 | 32 | 16 | - |
| 11 | conv2d | 3 | 16 | 16 | 192 | 256 |
| 12 | Batchnorm2d | - | 16 | 16 | - |
| 13 | ReLU | - | 16 | 16 | - |
| 14 | conv2d | 3 | 16 | 16 | 256 | 384 |
| 15 | Batchnorm2d | - | 16 | 16 | - |
| 16 | ReLU | - | 16 | 16 | - |
| 17 | conv2d | 3 | 16 | 16 | 384 | 512 |
| 18 | Batchnorm2d | - | 16 | 16 | - |
| 19 | ReLU | - | 16 | 16 | - |
| 20 | Maxpool2d | 2 | 16 | 8 | - |
| 21 | Linear | - | 32768 | 1024 | - |
| 22 | ReLU | - | 1024 | 1024 | - |
| 23 | Batchnorm1d | - | 1024 | 1024 | - |
| 24 | Linear | - | 1024 | 512 | - |
| 25 | ReLU | - | 512 | 512 | - |
| 26 | Batchnorm1d | - | 512 | 512 | - |
| 27 | Linear | - | 512 | 100 | - |

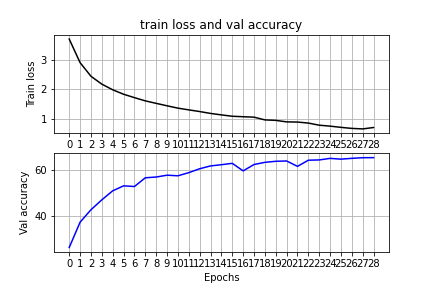
**Description of the network:**

* Data Normalization is done on the mode. The mean set to zero and the standard deviation was set to one to normalize the data. Moreover, data was transformed with the techniques like **RandomHorizontalFlip()** and **RandomCrop(32, padding =4).** Hence Data Normalization and Data augmentation was done on the model.
* After the Data transformation with the above-mentioned techniques, convolution layers were added to make the network deeper and more accurate. Thereafter, many fully connected layers were added to make the accuracy better.
* To reduce overfitting and improve training of the model, normalization layers like **BatchNorm2d** and **BatchNorm1d** were added before and after **ReLU** and **Linear layers**. Different arrangement of layers was tested to get better training accuracy.
* Though the model was tested with different values of **Epochs**, but the above final network has Epochs value set to **29**, which gives the final validation accuracy of **65%**.
* The rest of the things were set to default values.

**Plot Illustrating the training loss and validation accuracy:**

The final model is trained with Epochs value = 29 which gives the validation accuracy of 65%.

Below is the plot for the network:



**Ablation Study:**

* Initially, the default **BaseNet** gives the validation accuracy of about 26%.

* After Data Normalization and Data Augmentation, Epochs value was set to 25 and 5 conv layers along with **Maxpool** , **Linear**, and **Normalization** layers were added that has 6 as the least input channel value and 512 as the maximum. The accuracy noted was about **61%**. The model was tested by changing the distribution and numbering of the layers by keeping range of the input channel values similar. The accuracy noted were ranging from **61% - 62%.**
* After many testing by setting the Epochs value in range of 25 -35, accuracy was ranging from **61% - 63%**.
* Later, model testing was fixed with 6 conv layers, 3 Linear layers, and 2-3 Maxpool layers. ReLU layers were added after every Linear Layer. Model was tested with different positioning of Maxpool layers but later the position is fixed to one after 3rd conv layer and the other after 6th. The best accuracy achieved here with Epochs = 33 was **66%** with my 2nd best Kaggle score of **66.6%.** Note: The range of input channels here was **32 – 512**.
* **Final Network:** After testing with every possible combination in the reach, model is finalized with 6 conv layers with the above-mentioned layer combinations. The range of the input channels was changed to 64 - 384 and the Epochs value was set to 29. The maximum output channel value was set to 512 which is getting by giving 384 as input channel in the last conv layer. The **validation accuracy** achieved was **65%** with the best Kaggle score of **67.7%.**

**Base Performance:**

Below is the result of training the network with final validation accuracy of **65%**:

|  |
| --- |
| [1] loss: 3.702  Accuracy of the network on the val images: 26 %  [2] loss: 2.899  Accuracy of the network on the val images: 37 %  [3] loss: 2.437  Accuracy of the network on the val images: 42 %  [4] loss: 2.168  Accuracy of the network on the val images: 46 %  [5] loss: 1.977  Accuracy of the network on the val images: 50 %  [6] loss: 1.825  Accuracy of the network on the val images: 53 %  [7] loss: 1.710  Accuracy of the network on the val images: 52 %  [8] loss: 1.600  Accuracy of the network on the val images: 56 %  [9] loss: 1.515  Accuracy of the network on the val images: 57 %  [10] loss: 1.432  Accuracy of the network on the val images: 57 %  [11] loss: 1.353  Accuracy of the network on the val images: 57 %  [12] loss: 1.294  Accuracy of the network on the val images: 58 %  [13] loss: 1.237  Accuracy of the network on the val images: 60 %  [14] loss: 1.173  Accuracy of the network on the val images: 61 %  [15] loss: 1.123  Accuracy of the network on the val images: 62 %  [16] loss: 1.077  Accuracy of the network on the val images: 63 %  [17] loss: 1.061  Accuracy of the network on the val images: 59 %  [18] loss: 1.046  Accuracy of the network on the val images: 62 %  [19] loss: 0.953  Accuracy of the network on the val images: 63 %  [20] loss: 0.937  Accuracy of the network on the val images: 63 %  [21] loss: 0.886  Accuracy of the network on the val images: 64 %  [22] loss: 0.880  Accuracy of the network on the val images: 61 %  [23] loss: 0.845  Accuracy of the network on the val images: 64 %  [24] loss: 0.773  Accuracy of the network on the val images: 64 %  [25] loss: 0.741  Accuracy of the network on the val images: 65 %  [26] loss: 0.699  Accuracy of the network on the val images: 64 %  [27] loss: 0.664  Accuracy of the network on the val images: 65 %  [28] loss: 0.647  Accuracy of the network on the val images: 65 %  [29] loss: 0.694  Accuracy of the network on the val images: 65 %  Finished Training |

**Relative Performance:**

The best performing model is submitted with .csv file on Kaggle under Id **Spag01**. The network has best score of **67.7%.**

**Part 2: Transfer Learning**

In the second part of the project, it is asked to **fine-tune** a **ResNet Model** for classifying the

[Caltech-UCSD Birds dataset](http://www.vision.caltech.edu/datasets/cub_200_2011/). Hyperparameter tunning is done in this part to get the train achieve the train accuracy above 80% and test accuracy above 55%. The model will be trained and get tested for **ResNet as a fixed Feature Extractor** which is used to train the final layer and **Fine-tuning the Resnet** which trains the whole ResNet model and generate test results for the same.

**Hyperparameter Tuning**

Initially, the model with default hyperparameter values would give the training accuracy about 15.5%. So, for tuning the model, data is normalized and augmented. The mean and standard deviation was to zero and one respectively. Data was transformed using architectures like **RandomHorizontalFlip(), RandomRotation(35),** and **RandomResizedCrop(256).** Moreover, **Epochs** were set to **40**, **learning rate** was set to **0.001**, and **Batch size** was set to **32**. By tuning the model, required accuracies were achieved.

Below are the results for **ResNet as a fixed Feature Extractor** and **Fine-tuning the Resnet:**

**ResNet as a fixed Feature Extractor:**

To get the training and test accuracy of the model, variable RESNET\_LAST\_ONLY was set to True. The model achieved **55.8% training accuracy** and **58.1% test accuracy as shown below-**

**Training accuracy**

|  |
| --- |
| TRAINING Epoch 1/40 Loss 0.3382 Accuracy 0.0070  TRAINING Epoch 2/40 Loss 0.3213 Accuracy 0.0273  TRAINING Epoch 3/40 Loss 0.3075 Accuracy 0.0533  TRAINING Epoch 4/40 Loss 0.2949 Accuracy 0.0890  TRAINING Epoch 5/40 Loss 0.2817 Accuracy 0.1400  TRAINING Epoch 6/40 Loss 0.2713 Accuracy 0.1727  TRAINING Epoch 7/40 Loss 0.2617 Accuracy 0.2053  TRAINING Epoch 8/40 Loss 0.2530 Accuracy 0.2523  TRAINING Epoch 9/40 Loss 0.2450 Accuracy 0.2713  TRAINING Epoch 10/40 Loss 0.2359 Accuracy 0.3017  TRAINING Epoch 11/40 Loss 0.2285 Accuracy 0.3367  TRAINING Epoch 12/40 Loss 0.2212 Accuracy 0.3447  TRAINING Epoch 13/40 Loss 0.2151 Accuracy 0.3677  TRAINING Epoch 14/40 Loss 0.2108 Accuracy 0.3753  TRAINING Epoch 15/40 Loss 0.2048 Accuracy 0.3957  TRAINING Epoch 16/40 Loss 0.2019 Accuracy 0.4063  TRAINING Epoch 17/40 Loss 0.1949 Accuracy 0.4217  TRAINING Epoch 18/40 Loss 0.1904 Accuracy 0.4267  TRAINING Epoch 19/40 Loss 0.1872 Accuracy 0.4397  TRAINING Epoch 20/40 Loss 0.1840 Accuracy 0.4507  TRAINING Epoch 21/40 Loss 0.1840 Accuracy 0.4373  TRAINING Epoch 22/40 Loss 0.1756 Accuracy 0.4750  TRAINING Epoch 23/40 Loss 0.1731 Accuracy 0.4740  TRAINING Epoch 24/40 Loss 0.1700 Accuracy 0.4847  TRAINING Epoch 25/40 Loss 0.1684 Accuracy 0.4790  TRAINING Epoch 26/40 Loss 0.1689 Accuracy 0.4773  TRAINING Epoch 27/40 Loss 0.1655 Accuracy 0.4940  TRAINING Epoch 28/40 Loss 0.1606 Accuracy 0.5007  TRAINING Epoch 29/40 Loss 0.1578 Accuracy 0.5143  TRAINING Epoch 30/40 Loss 0.1560 Accuracy 0.5217  TRAINING Epoch 31/40 Loss 0.1550 Accuracy 0.5163  TRAINING Epoch 32/40 Loss 0.1540 Accuracy 0.5157  TRAINING Epoch 33/40 Loss 0.1500 Accuracy 0.5367  TRAINING Epoch 34/40 Loss 0.1475 Accuracy 0.5393  TRAINING Epoch 35/40 Loss 0.1478 Accuracy 0.5353  TRAINING Epoch 36/40 Loss 0.1477 Accuracy 0.5147  TRAINING Epoch 37/40 Loss 0.1456 Accuracy 0.5380  TRAINING Epoch 38/40 Loss 0.1442 Accuracy 0.5297  TRAINING Epoch 39/40 Loss 0.1449 Accuracy 0.5353  TRAINING Epoch 40/40 Loss 0.1397 Accuracy 0.5583  Finished Training |

**Test Accuracy**

|  |
| --- |
| Test Loss: 0.1028 Test Accuracy 0.5813 |

**ResNet as a fixed Feature Extractor:**

To get the training and test accuracy of the full ResNet model, variable RESNET\_LAST\_ONLY was set to False. The model achieved **81.27% training accuracy** and **56.02% test accuracy as shown below-**

**Training Accuracy**

|  |
| --- |
| TRAINING Epoch 1/40 Loss 0.1680 Accuracy 0.0073  TRAINING Epoch 2/40 Loss 0.1578 Accuracy 0.0357  TRAINING Epoch 3/40 Loss 0.1455 Accuracy 0.1027  TRAINING Epoch 4/40 Loss 0.1332 Accuracy 0.1837  TRAINING Epoch 5/40 Loss 0.1224 Accuracy 0.2460  TRAINING Epoch 6/40 Loss 0.1134 Accuracy 0.3000  TRAINING Epoch 7/40 Loss 0.1051 Accuracy 0.3567  TRAINING Epoch 8/40 Loss 0.0979 Accuracy 0.4143  TRAINING Epoch 9/40 Loss 0.0920 Accuracy 0.4380  TRAINING Epoch 10/40 Loss 0.0875 Accuracy 0.4673  TRAINING Epoch 11/40 Loss 0.0821 Accuracy 0.5037  TRAINING Epoch 12/40 Loss 0.0782 Accuracy 0.5170  TRAINING Epoch 13/40 Loss 0.0738 Accuracy 0.5603  TRAINING Epoch 14/40 Loss 0.0704 Accuracy 0.5753  TRAINING Epoch 15/40 Loss 0.0673 Accuracy 0.5933  TRAINING Epoch 16/40 Loss 0.0639 Accuracy 0.6233  TRAINING Epoch 17/40 Loss 0.0614 Accuracy 0.6197  TRAINING Epoch 18/40 Loss 0.0588 Accuracy 0.6467  TRAINING Epoch 19/40 Loss 0.0566 Accuracy 0.6553  TRAINING Epoch 20/40 Loss 0.0554 Accuracy 0.6640  TRAINING Epoch 21/40 Loss 0.0531 Accuracy 0.6753  TRAINING Epoch 22/40 Loss 0.0497 Accuracy 0.7053  TRAINING Epoch 23/40 Loss 0.0494 Accuracy 0.6963  TRAINING Epoch 24/40 Loss 0.0483 Accuracy 0.6957  TRAINING Epoch 25/40 Loss 0.0456 Accuracy 0.7277  TRAINING Epoch 26/40 Loss 0.0435 Accuracy 0.7423  TRAINING Epoch 27/40 Loss 0.0430 Accuracy 0.7363  TRAINING Epoch 28/40 Loss 0.0415 Accuracy 0.7477  TRAINING Epoch 29/40 Loss 0.0405 Accuracy 0.7550  TRAINING Epoch 30/40 Loss 0.0386 Accuracy 0.7763  TRAINING Epoch 31/40 Loss 0.0380 Accuracy 0.7720  TRAINING Epoch 32/40 Loss 0.0377 Accuracy 0.7687  TRAINING Epoch 33/40 Loss 0.0359 Accuracy 0.7763  TRAINING Epoch 34/40 Loss 0.0349 Accuracy 0.7803  TRAINING Epoch 35/40 Loss 0.0338 Accuracy 0.7997  TRAINING Epoch 36/40 Loss 0.0326 Accuracy 0.8147  TRAINING Epoch 37/40 Loss 0.0320 Accuracy 0.8060  TRAINING Epoch 38/40 Loss 0.0316 Accuracy 0.8090  TRAINING Epoch 39/40 Loss 0.0315 Accuracy 0.8140  TRAINING Epoch 40/40 Loss 0.0302 Accuracy 0.8127  Finished Training  ---------- |

**Test Accuracy**

|  |
| --- |
| Test Loss: 0.0529 Test Accuracy 0.5608 |

The **lab2.ipynb** file is submitted for the reported model.