A3_Q2_Haider

August 11, 2023

0.1 Question 2: Animal classification (15 marks)

For this question, we will use the Animal (https://cloudstor.aarnet.edu.au/plus/s/cZYtNAeVhWD6uBX) dataset. This dataset contains images of 151 different animals.

The dataset contains a total of 6270 images corresponding to the name of animal types.

All images are RGB images of 224 pixels wide by 224 pixels high in .jpg format. The images are separated in 151 folders according to their respective class.

The task is to categorize each animal into one of 151 categories.

We provide baseline code that includes the following features:

- Loading and Analysing the dataset using torchvision.
- Defining a simple convolutional neural network.
- How to use existing loss function for the model learning.
- Train the network on the training data.
- Test the trained network on the testing data.

The following changes could be considered:

1. "Transfer" Learning (ie use a model pre-trained another dataset)
2. Change of advanced training parameters: Learning Rate, Optimizer, Batch-size, Number of Max Epochs, and Drop-out.
3. Use of a new loss function.
4. Data augmentation
5. Architectural Changes: Batch Normalization, Residual layers, etc.
6. Others - please ask us on the Discussion Forums if you're not sure about an idea!

Your code should be modified from the provided baseline. A pdf report of a maximum of two pages is required to explain the changes you made from the baseline, why you chose those changes, and the improvements they achieved.

0.1.1 Marking Rules:

We will mark this question based on the final test accuracy on testing images and your report.

Final mark (out of 50) = acc_mark + efficiency mark + report mark

Acc_mark 10:

We will rank all the submission results based on their test accuracy. Zero improvement over the baseline yields 0 marks. Maximum improvement over the baseline will yield 10 marks. There will

be a sliding scale applied in between.

Efficiency mark 10:

Efficiency considers not only the accuracy, but the computational cost of running the model (flops: https://en.wikipedia.org/wiki/FLOPS). Efficiency for our purposes is defined to be the ratio of accuracy (in %) to Gflops. Please report the computational cost for your final model and include the efficiency calculation in your report. Maximum improvement over the baseline will yield 10 marks. Zero improvement over the baseline yields zero marks, with a sliding scale in between.

Report mark 30:

Your report should comprise: 1. An introduction showing your understanding of the task and of the baseline model: [10 marks]

2. A description of how you have modified aspects of the system to improve performance. [10 marks]

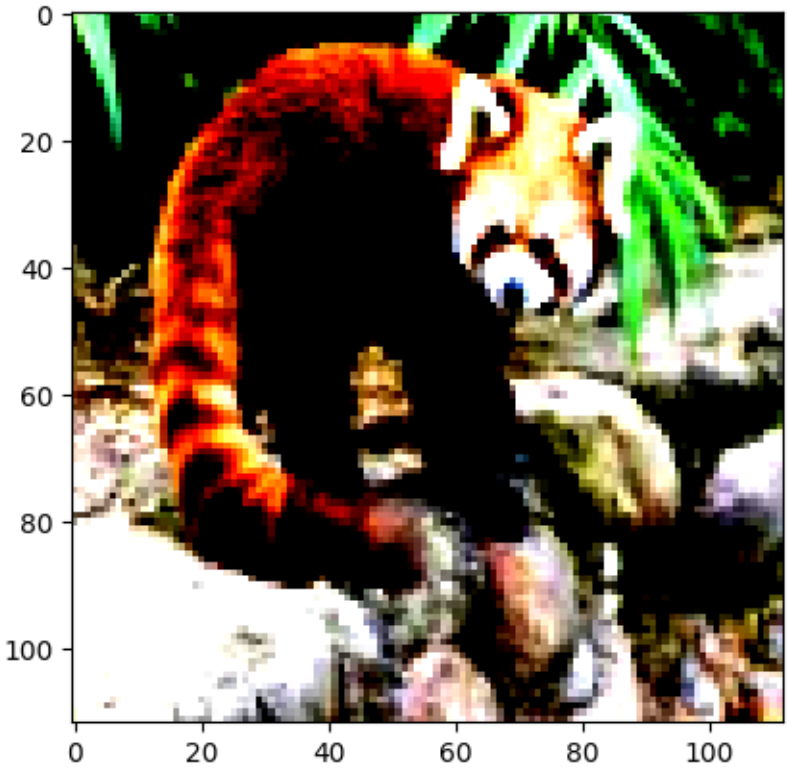
A recommended way to present a summary of this is via an "ablation study" table, eg:

Method1	Method2	Method3	Accuracy
N	N	N	60%
Y	N	N	65%
Y	Y	N	77%
Y	Y	Y	82%

3. Explanation of the methods for reducing the computational cost and/or improve the trade-off between accuracy and cost: [5 marks]

4. Limitations/Conclusions: [5 marks]

```
Mounted at /content/drive
/content/drive/My Drive/cv_assignment3

Size of training dataset : 6270

torch.Size([3, 112, 112])

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Label:  ailurus-fulgens (5)
```




```
(5330, 313, 627)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=3200, out_features=151, bias=True)
)
```

```
images.shape: torch.Size([16, 3, 112, 112])
out.shape: torch.Size([16, 151])
out[0]: tensor([-5.0253, -5.0075, -4.9627, -5.0596, -5.0815, -4.9866, -5.0008,
-4.9870,
```

```
        -5.0678, -5.0146, -5.0684, -4.9802, -4.9810, -5.0258, -5.0535, -5.0307,
        -5.0418, -5.0117, -5.0337, -5.0770, -5.0106, -5.0112, -5.0253, -4.9587,
        -4.9998, -4.9620, -4.9563, -5.0806, -5.0266, -4.9681, -4.9688, -5.0341,
        -5.0871, -5.0054, -5.0148, -4.9946, -5.0253, -4.9978, -4.9086, -5.0187,
        -5.0609, -5.0372, -5.0035, -5.0826, -4.9326, -5.0396, -5.0156, -5.0929,
        -5.0151, -5.0075, -5.0279, -5.0491, -4.9886, -5.0747, -5.0234, -5.0762,
        -5.0536, -5.0433, -5.0373, -4.9688, -5.0567, -5.0227, -5.0481, -5.0432,
        -5.0580, -4.9815, -5.0384, -5.0471, -5.0285, -5.0213, -5.0451, -5.0055,
        -5.0277, -5.0750, -5.0670, -5.0840, -5.0346, -4.9949, -5.0220, -4.9649,
        -5.0837, -5.0014, -5.0467, -4.9690, -4.9925, -4.9640, -5.0435, -5.0278,
        -5.0357, -5.0496, -5.0517, -5.0088, -5.0177, -4.9542, -5.0016, -5.0400,
        -5.0196, -5.0168, -5.0874, -5.0769, -5.0624, -5.0368, -4.9610, -5.0215,
        -4.9595, -5.0215, -5.0327, -5.0223, -4.9540, -5.0473, -5.0826, -5.0351,
        -5.1066, -5.0157, -5.0138, -5.0146, -4.9556, -4.9345, -4.9744, -5.0310,
        -5.0230, -5.0704, -4.9948, -4.9817, -5.0034, -4.9679, -4.9537, -5.0187,
        -4.9682, -5.0083, -4.9759, -4.9625, -5.0273, -4.9851, -5.0317, -5.0196,
        -4.9901, -5.0643, -5.0197, -5.0291, -5.0257, -5.0060, -5.0191, -4.9599,
        -4.9919, -4.9667, -4.9805, -5.0451, -4.9737, -5.0820, -4.9577],
       device='cuda:0', grad_fn=<SelectBackward0>)
```

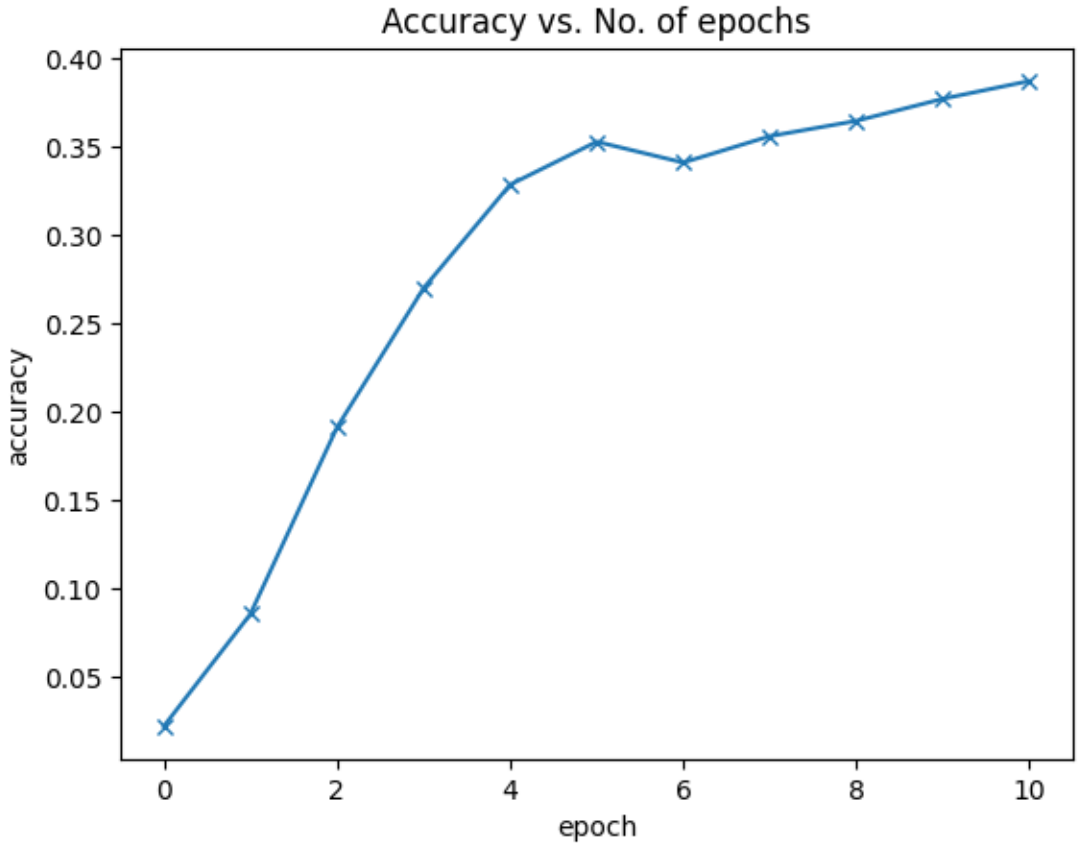
```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=3200, out_features=151, bias=True)
)
```

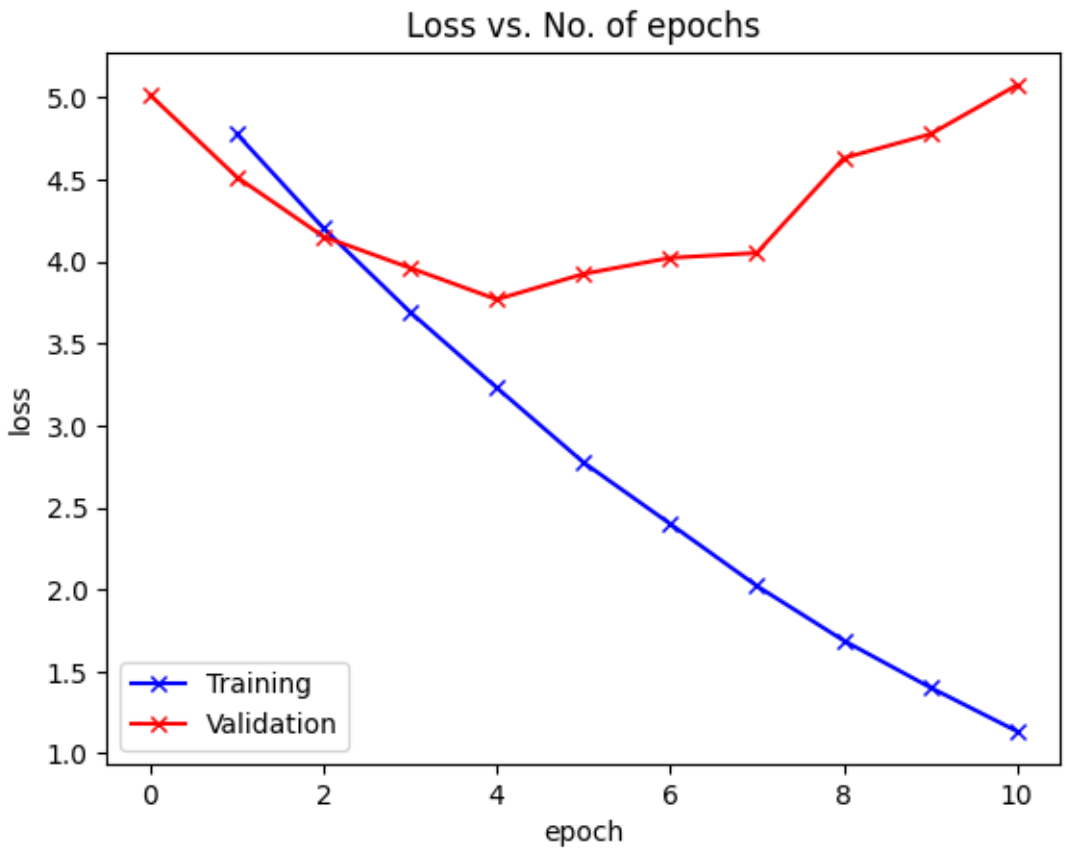
```
[{'val_loss': 5.015639781951904, 'val_acc': 0.02187499962747097}]
```

```
627
```

```
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [0], train_loss: 4.7789, val_loss: 4.5158, val_acc: 0.0854
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [1], train_loss: 4.2031, val_loss: 4.1520, val_acc: 0.1910
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [2], train_loss: 3.6918, val_loss: 3.9611, val_acc: 0.2698
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [3], train_loss: 3.2300, val_loss: 3.7690, val_acc: 0.3285
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [4], train_loss: 2.7744, val_loss: 3.9248, val_acc: 0.3528
  0%|          | 0/334 [00:00<?, ?it/s]
```

```
Epoch [5], train_loss: 2.3999, val_loss: 4.0226, val_acc: 0.3410
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [6], train_loss: 2.0212, val_loss: 4.0528, val_acc: 0.3559
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [7], train_loss: 1.6884, val_loss: 4.6306, val_acc: 0.3646
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [8], train_loss: 1.4022, val_loss: 4.7770, val_acc: 0.3771
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [9], train_loss: 1.1351, val_loss: 5.0764, val_acc: 0.3872
```






```
{'val_loss': 5.2631025314331055, 'val_acc': 0.3567708432674408}
```

##FLOPs

```
--2023-08-10 17:43:55--
https://cloudstor.aarnet.edu.au/plus/s/hXo1dK9SZqiEVn9/download
Resolving cloudstor.aarnet.edu.au (cloudstor.aarnet.edu.au)… 202.158.207.20
Connecting to cloudstor.aarnet.edu.au
(cloudstor.aarnet.edu.au)|202.158.207.20|:443… connected.
HTTP request sent, awaiting response… 200 OK
Syntax error in Set-Cookie: 5230042dc1897=ir1lf3vqh41vhcq8bbkoa44van;
path=/plus; domain=.aarnet.edu.au;; Secure; SameSite=Lax at position 76.
Syntax error in Set-Cookie: oc_sessionPassphrase=Hk0QlLwZahCiZZojDt0M%2FLEj5Fmha
QEgtbUCMJmc64566zZZuTdNt7T7uGeWsxSsouaM89vp74%2B5vPQZidzA8zZxbcAsg%2BpurDSiEE1S2
HXdr%2FiFeAkGHTKms8HUSaY%2B; expires=Fri, 11-Aug-2023 08:13:55 GMT; Max-
Age=86400; path=/plus;; Secure; SameSite=Lax at position 226.

    The file is already fully retrieved; nothing to do.

'mv' is not recognized as an internal or external command,
operable program or batch file.
```

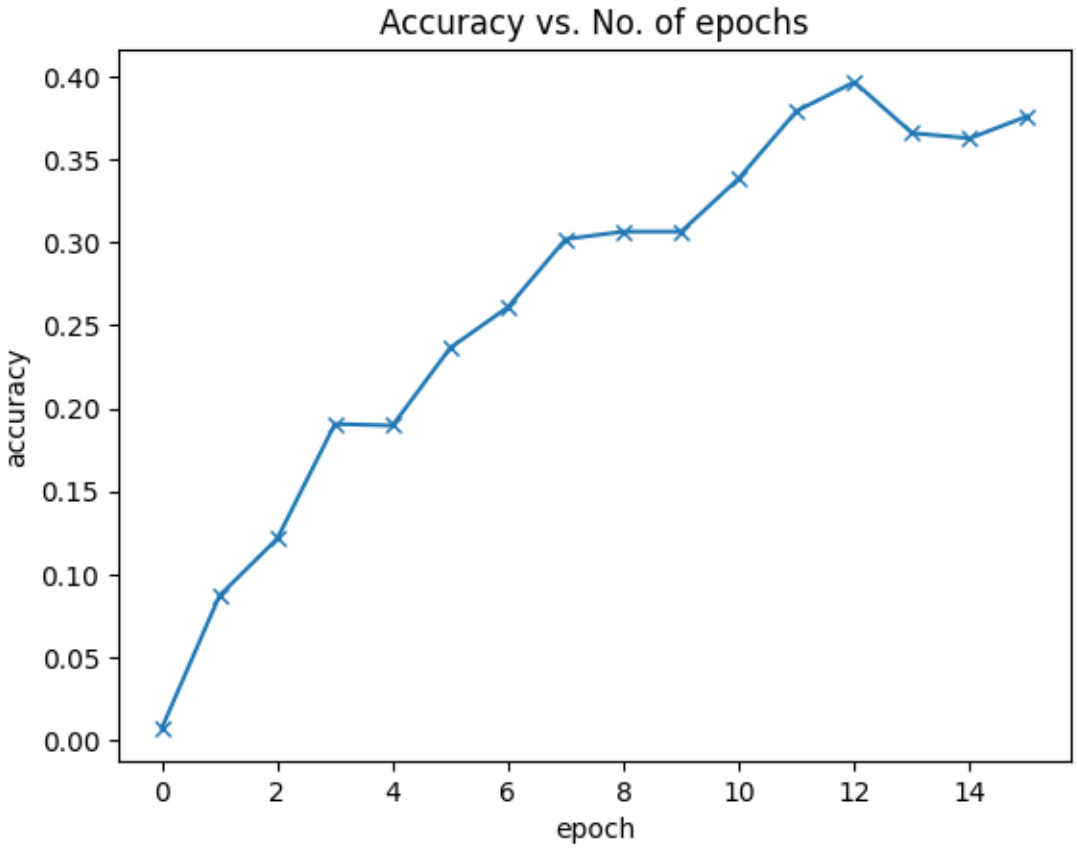
```
 + Number of FLOPs: 0.69G
```

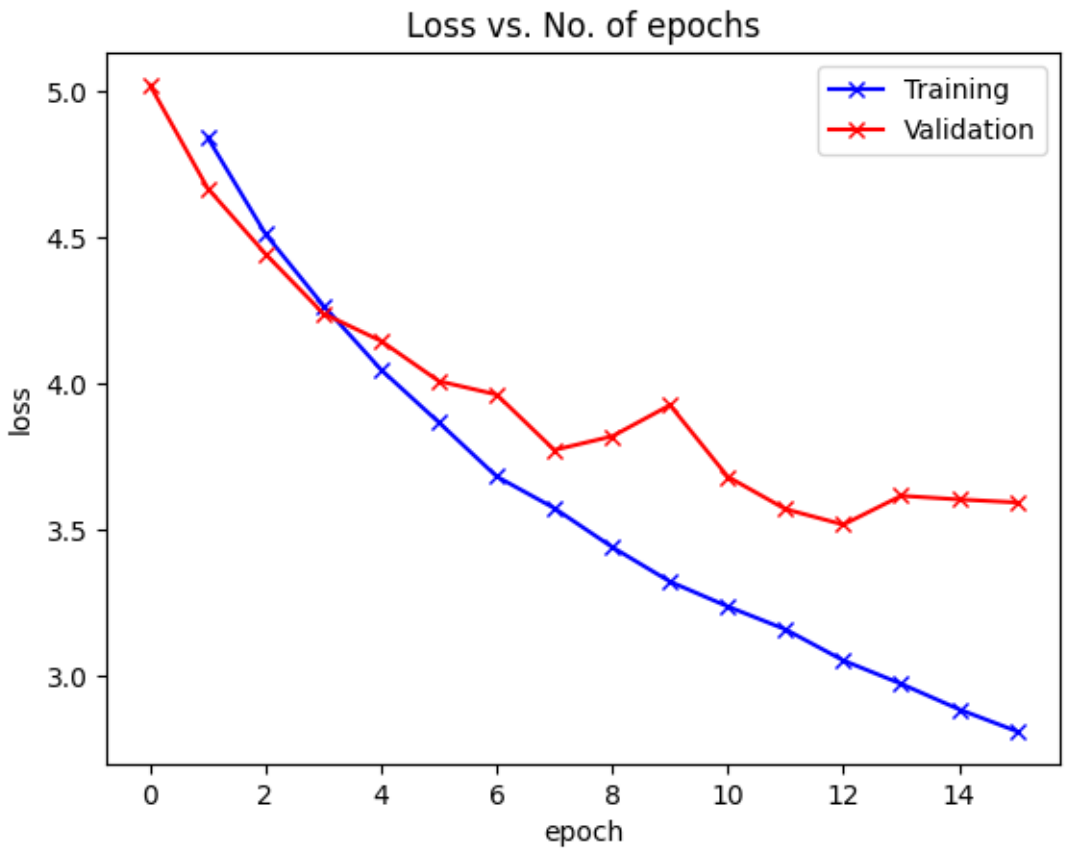
1 Test Model 1

Data Augmentation, Dropout (0.2)(Due to potential overfitting of curve) and learning rate=0.001 and number of epochs = 15 batch size = 32

```
Size of training dataset : 6270

627

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.8413, val_loss: 4.6678, val_acc: 0.0870

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [1], train_loss: 4.5114, val_loss: 4.4433, val_acc: 0.1214

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [2], train_loss: 4.2669, val_loss: 4.2393, val_acc: 0.1905

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [3], train_loss: 4.0471, val_loss: 4.1454, val_acc: 0.1896

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [4], train_loss: 3.8667, val_loss: 4.0079, val_acc: 0.2365

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [5], train_loss: 3.6818, val_loss: 3.9620, val_acc: 0.2610

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [6], train_loss: 3.5730, val_loss: 3.7723, val_acc: 0.3020

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [7], train_loss: 3.4401, val_loss: 3.8191, val_acc: 0.3065

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [8], train_loss: 3.3218, val_loss: 3.9258, val_acc: 0.3065

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [9], train_loss: 3.2368, val_loss: 3.6809, val_acc: 0.3386

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [10], train_loss: 3.1582, val_loss: 3.5697, val_acc: 0.3793

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [11], train_loss: 3.0521, val_loss: 3.5169, val_acc: 0.3966

  0%|          | 0/167 [00:00<?, ?it/s]
```

```
Epoch [12], train_loss: 2.9722, val_loss: 3.6156, val_acc: 0.3659
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [13], train_loss: 2.8850, val_loss: 3.6030, val_acc: 0.3628
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [14], train_loss: 2.8099, val_loss: 3.5928, val_acc: 0.3761
```





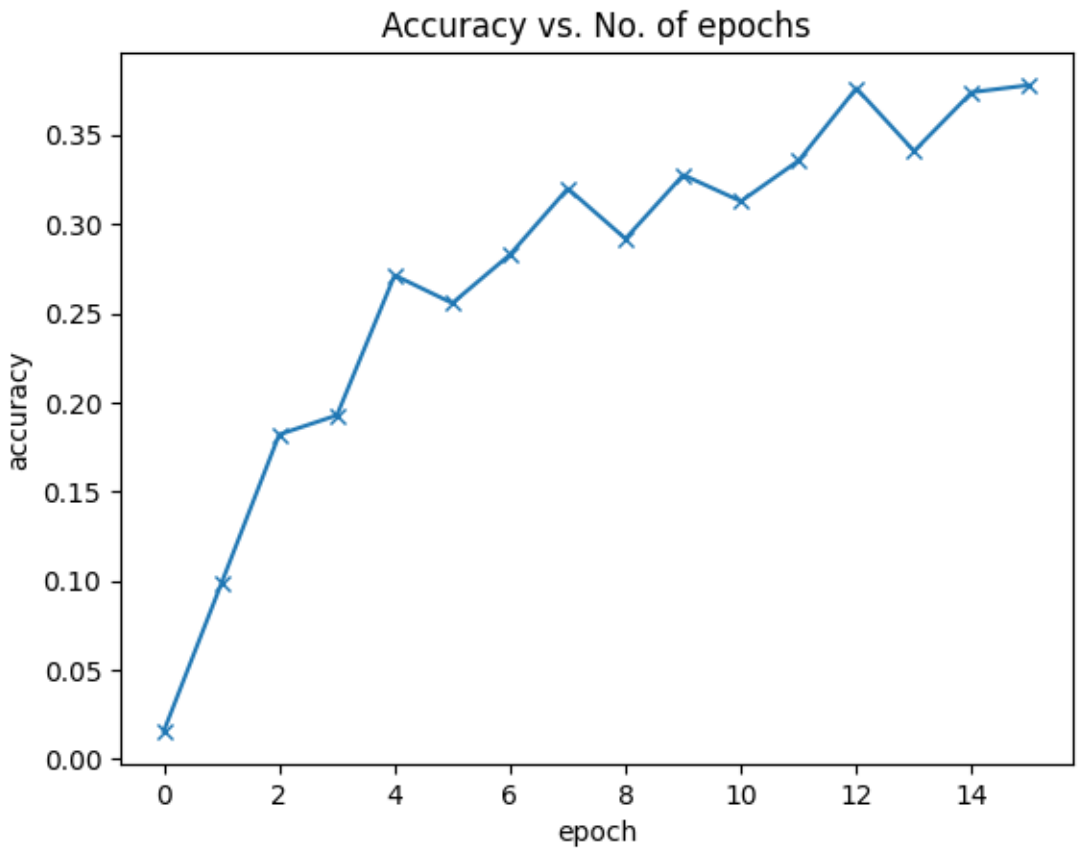

```
{'val_loss': 3.2880661487579346, 'val_acc': 0.43881580233573914}

 + Number of FLOPs: 0.69G
```

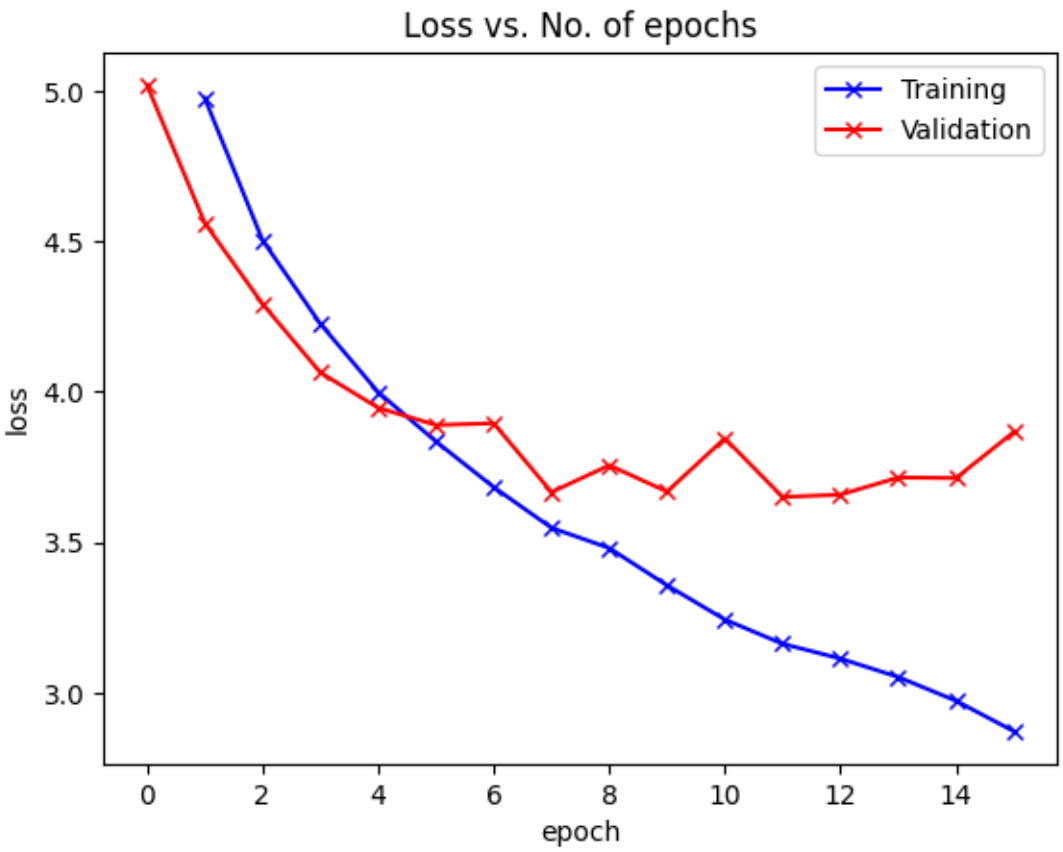
2 Test Model 2

Implemented batch normalization and dropout at layers 3 and 4, and built a wider and deeper model with one additional hidden layer.

```
Size of training dataset : 6270
627

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.9752, val_loss: 4.5606, val_acc: 0.0986

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [1], train_loss: 4.5035, val_loss: 4.2920, val_acc: 0.1820

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [2], train_loss: 4.2261, val_loss: 4.0640, val_acc: 0.1928
```

```
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [3], train_loss: 3.9990, val_loss: 3.9469, val_acc: 0.2713
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [4], train_loss: 3.8351, val_loss: 3.8895, val_acc: 0.2556
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [5], train_loss: 3.6834, val_loss: 3.8953, val_acc: 0.2829
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [6], train_loss: 3.5483, val_loss: 3.6669, val_acc: 0.3199
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [7], train_loss: 3.4810, val_loss: 3.7558, val_acc: 0.2918
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [8], train_loss: 3.3572, val_loss: 3.6684, val_acc: 0.3275
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [9], train_loss: 3.2437, val_loss: 3.8453, val_acc: 0.3128
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [10], train_loss: 3.1629, val_loss: 3.6505, val_acc: 0.3355
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [11], train_loss: 3.1139, val_loss: 3.6583, val_acc: 0.3761
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [12], train_loss: 3.0528, val_loss: 3.7153, val_acc: 0.3409
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [13], train_loss: 2.9738, val_loss: 3.7135, val_acc: 0.3739
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [14], train_loss: 2.8727, val_loss: 3.8664, val_acc: 0.3779
```


Accuracy vs. No. of epochs
accuracy
epoch
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
2
4
6
8
10
12
14

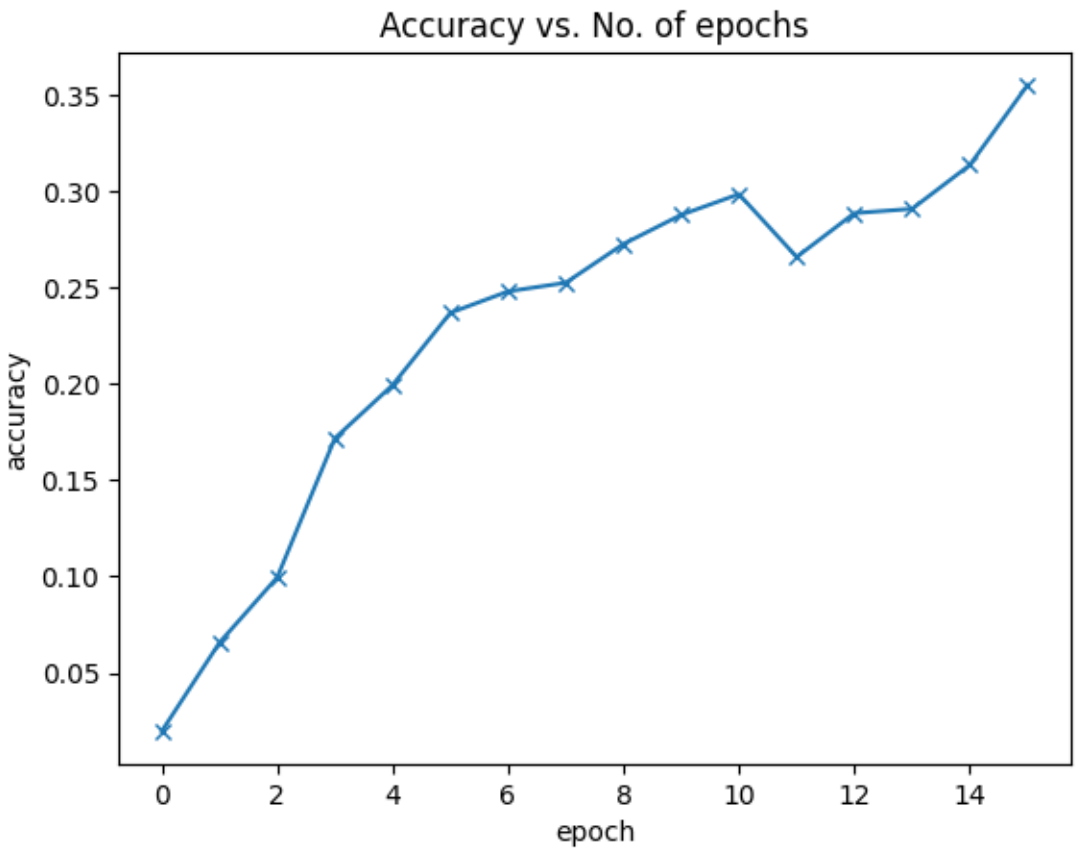


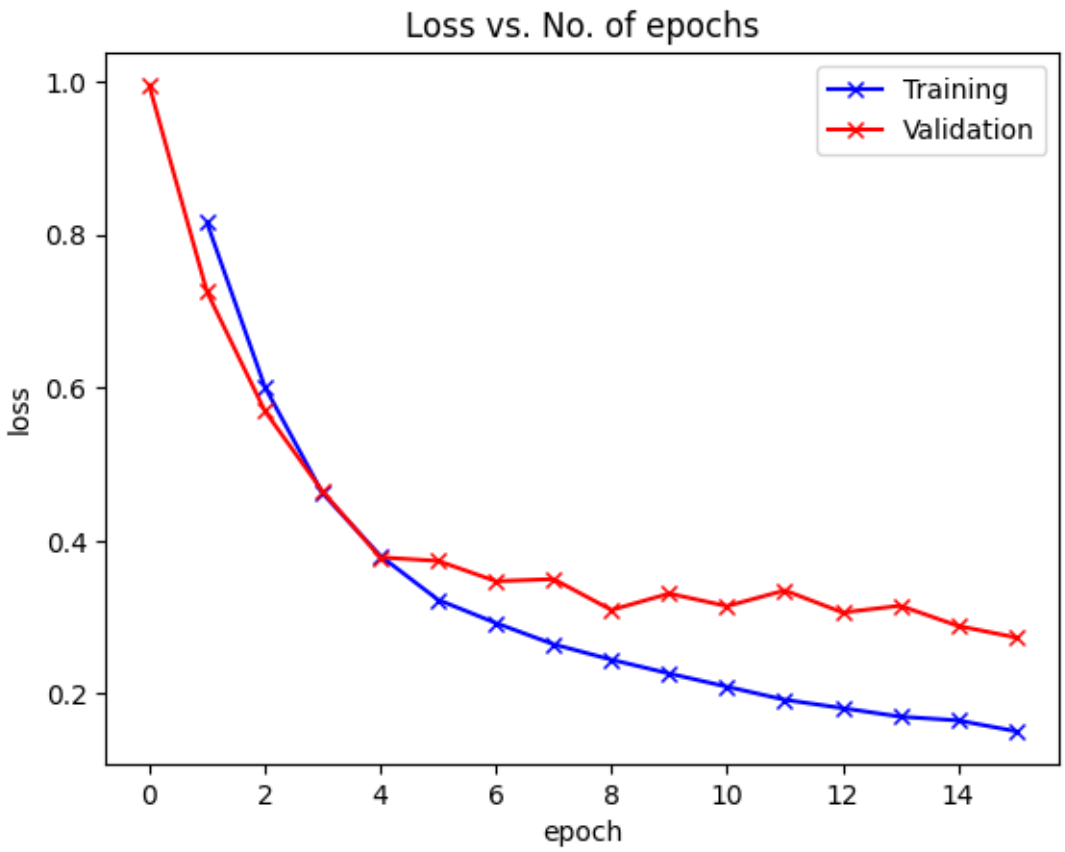

```
{'val_loss': 3.4997520446777344, 'val_acc': 0.43363484740257263}

 + Number of FLOPs: 0.96G
```

2.1 Test Model 3

```
Size of training dataset : 6270

627

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [0], train_loss: 0.8055, val_loss: 0.6791, val_acc: 0.0861

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [1], train_loss: 0.5713, val_loss: 0.5060, val_acc: 0.1178

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [2], train_loss: 0.4559, val_loss: 0.4312, val_acc: 0.1717

  0%|          | 0/167 [00:00<?, ?it/s]

Epoch [3], train_loss: 0.3738, val_loss: 0.3873, val_acc: 0.2093
```

```
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [4], train_loss: 0.3146, val_loss: 0.3533, val_acc: 0.1922
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [5], train_loss: 0.2738, val_loss: 0.3237, val_acc: 0.2735
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [6], train_loss: 0.2500, val_loss: 0.3173, val_acc: 0.2627
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [7], train_loss: 0.2239, val_loss: 0.3119, val_acc: 0.2636
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [8], train_loss: 0.2010, val_loss: 0.3257, val_acc: 0.2940
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [9], train_loss: 0.1927, val_loss: 0.2797, val_acc: 0.3444
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [10], train_loss: 0.1746, val_loss: 0.2858, val_acc: 0.3230
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [11], train_loss: 0.1610, val_loss: 0.2975, val_acc: 0.3011
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [12], train_loss: 0.1473, val_loss: 0.2686, val_acc: 0.3498
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [13], train_loss: 0.1407, val_loss: 0.2773, val_acc: 0.3333
  0%|          | 0/167 [00:00<?, ?it/s]
Epoch [14], train_loss: 0.1321, val_loss: 0.2861, val_acc: 0.3779
```


Accuracy vs. No. of epochs
accuracy
epoch
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
2
4
6
8
10
12
14

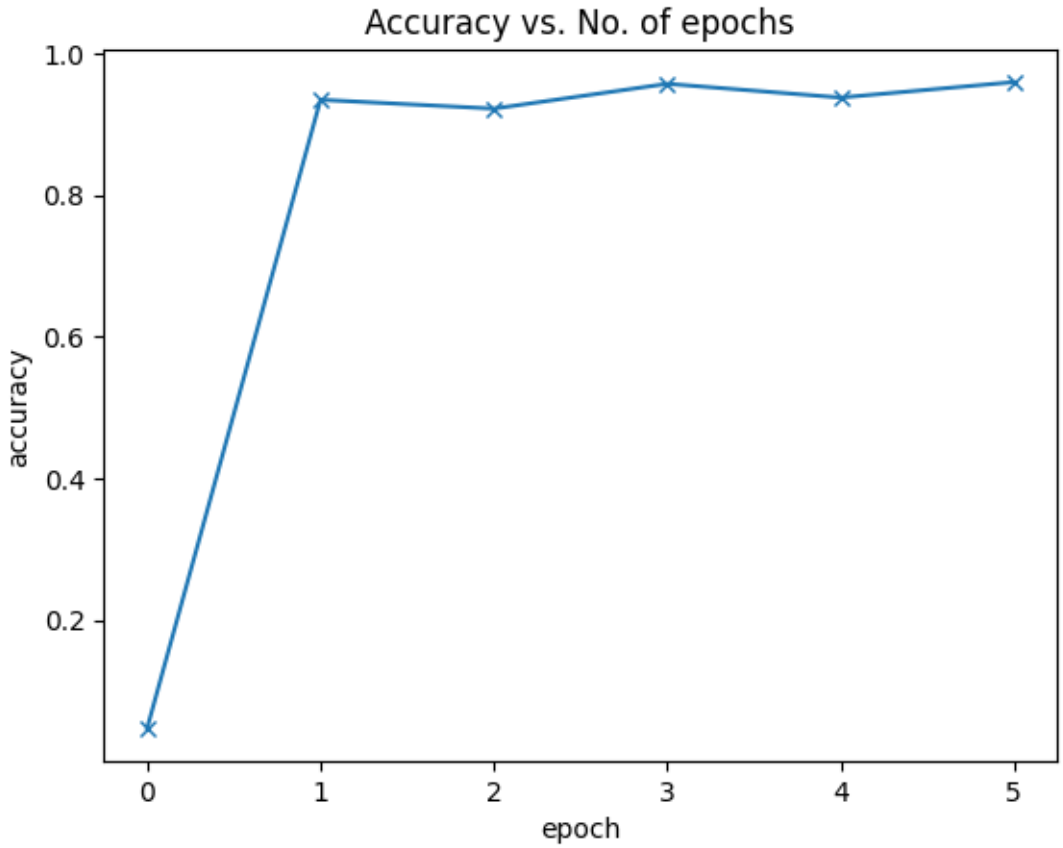


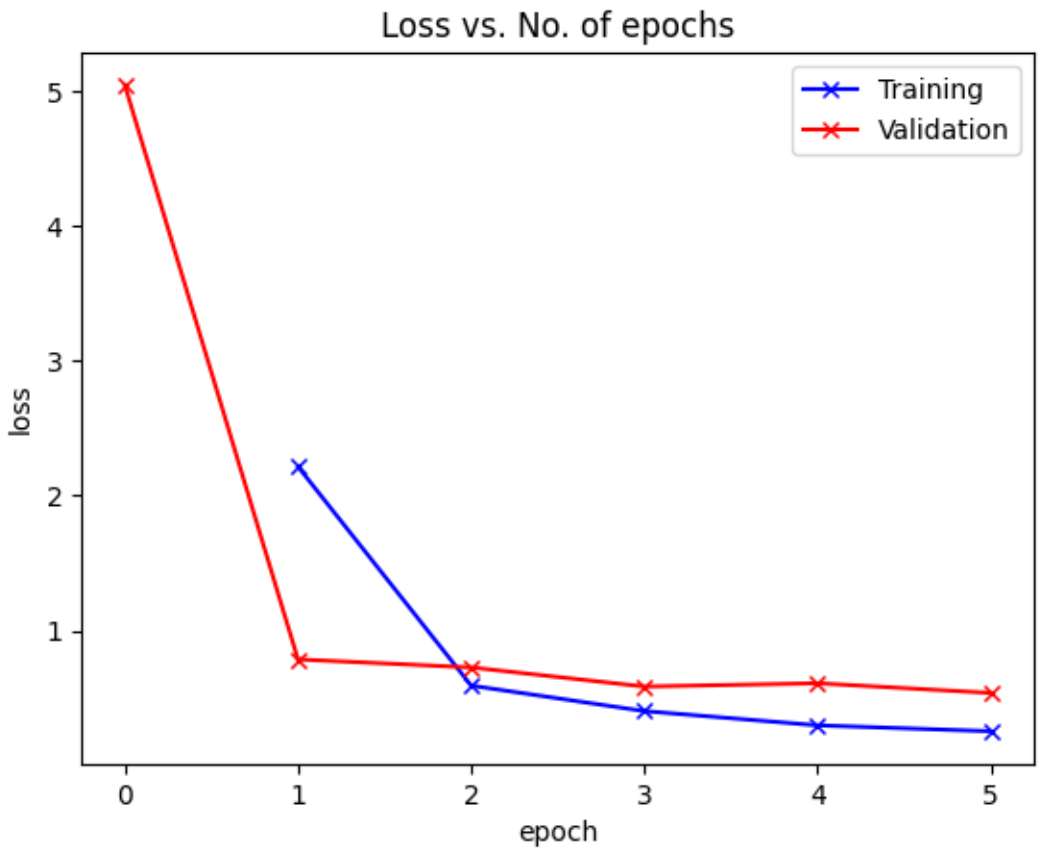

```
{'val_loss': 0.2702194154262543, 'val_acc': 0.3805098533630371}

 + Number of FLOPs: 0.69G
```

3 Transfer Learning 1 through Resnet50

```
Size of training dataset: 6270
627

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [0], train_loss: 2.2147, val_loss: 0.7857, val_acc: 0.9344

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [1], train_loss: 0.5917, val_loss: 0.7262, val_acc: 0.9219

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [2], train_loss: 0.4026, val_loss: 0.5846, val_acc: 0.9569

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [3], train_loss: 0.2971, val_loss: 0.6090, val_acc: 0.9375
```

```
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [4], train_loss: 0.2534, val_loss: 0.5371, val_acc: 0.9594
```



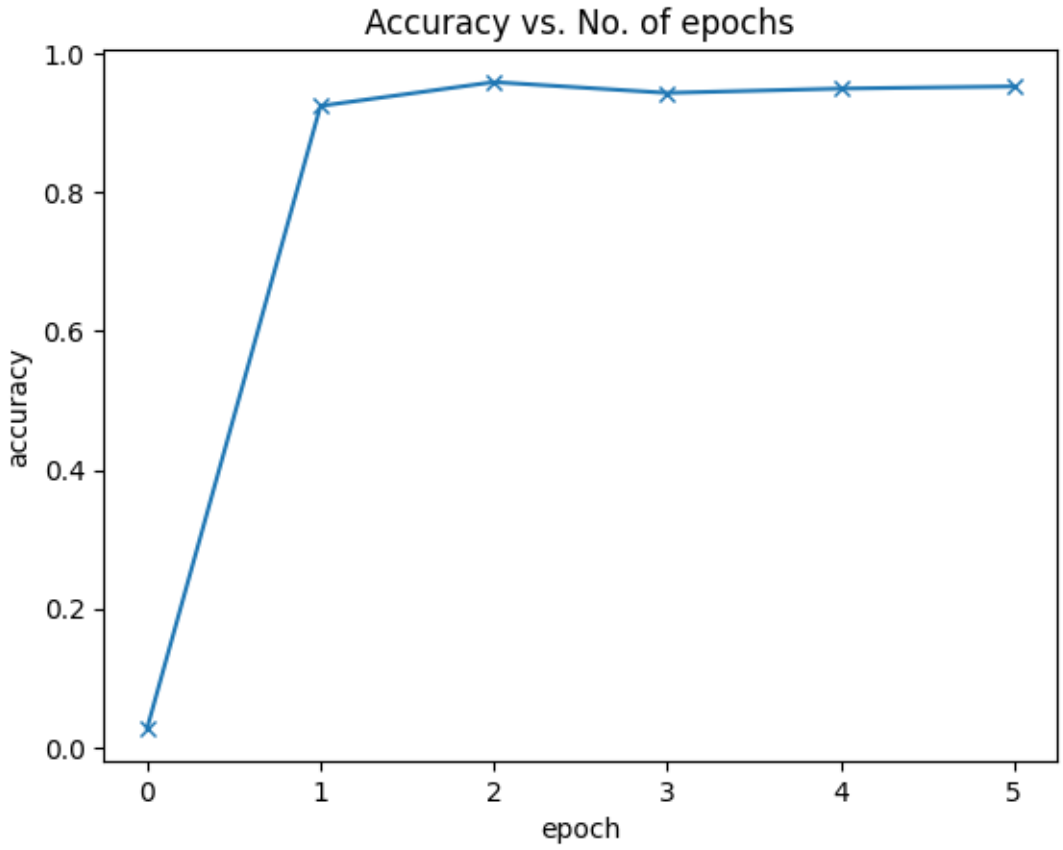


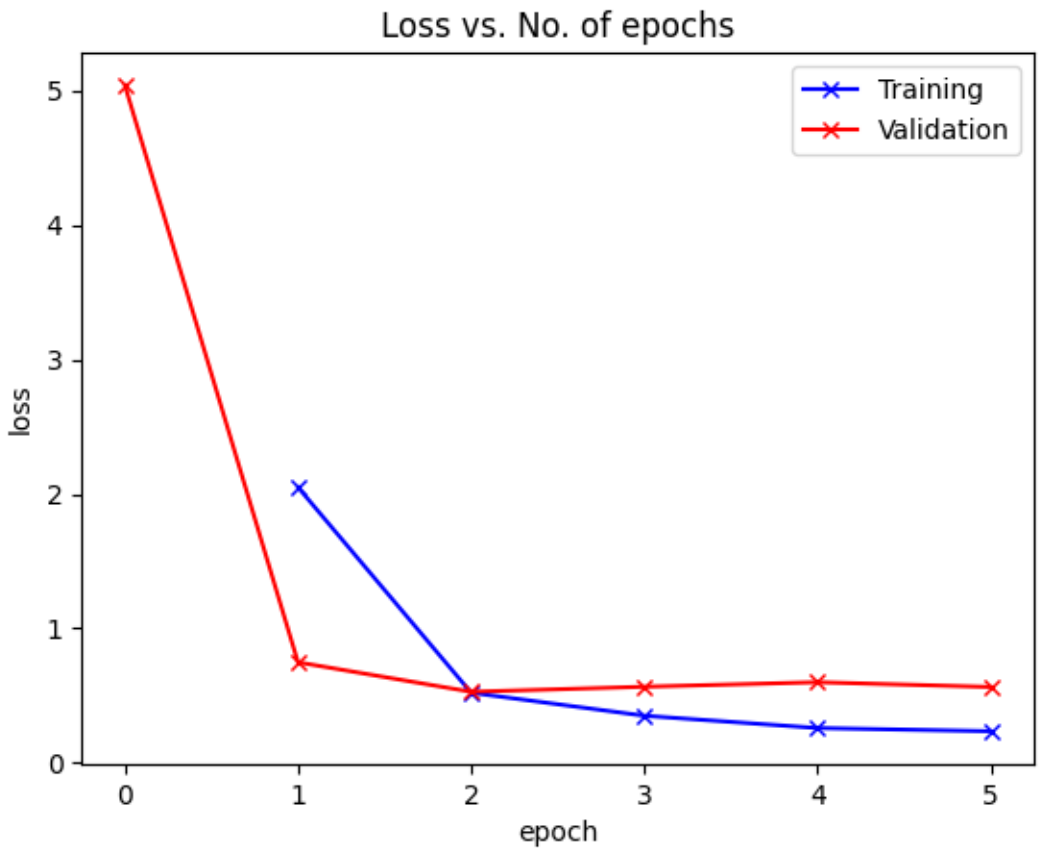

```
{'val_loss': 0.40933844447135925, 'val_acc': 0.964062511920929}

 + Number of FLOPs: 8.17G
```

4 Transfer Learning 2 through resnet101

```
Size of training dataset: 6270

Downloading: "https://download.pytorch.org/models/resnet101-63fe2227.pth" to
/root/.cache/torch/hub/checkpoints/resnet101-63fe2227.pth
100%|      | 171M/171M [00:03<00:00, 53.3MB/s]

627

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [0], train_loss: 2.0500, val_loss: 0.7487, val_acc: 0.9250

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [1], train_loss: 0.5224, val_loss: 0.5294, val_acc: 0.9594

  0%|          | 0/334 [00:00<?, ?it/s]
```

```
Epoch [2], train_loss: 0.3514, val_loss: 0.5670, val_acc: 0.9438
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [3], train_loss: 0.2600, val_loss: 0.6006, val_acc: 0.9500
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [4], train_loss: 0.2363, val_loss: 0.5652, val_acc: 0.9531
```






```
{'val_loss': 0.4656005799770355, 'val_acc': 0.9625000357627869}
```

```
 + Number of FLOPs: 15.61G
```

5 Transfer Learning 3 through resnet18

```
Size of training dataset: 6270
```

```
Downloading: "https://download.pytorch.org/models/resnet18-f37072fd.pth" to
/root/.cache/torch/hub/checkpoints/resnet18-f37072fd.pth
100%|      | 44.7M/44.7M [00:00<00:00, 255MB/s]
```

```
627
```

```
  0%|          | 0/334 [00:00<?, ?it/s]
```

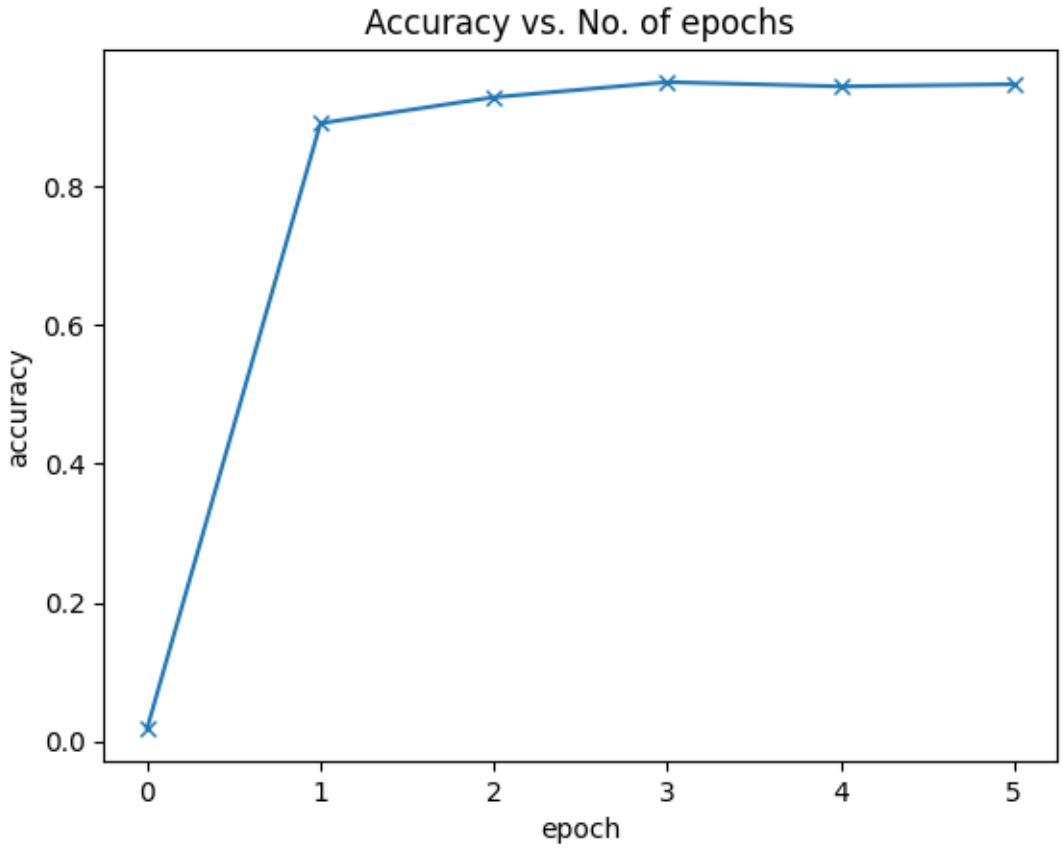
```
Epoch [0], train_loss: 2.7009, val_loss: 1.1856, val_acc: 0.8906
```

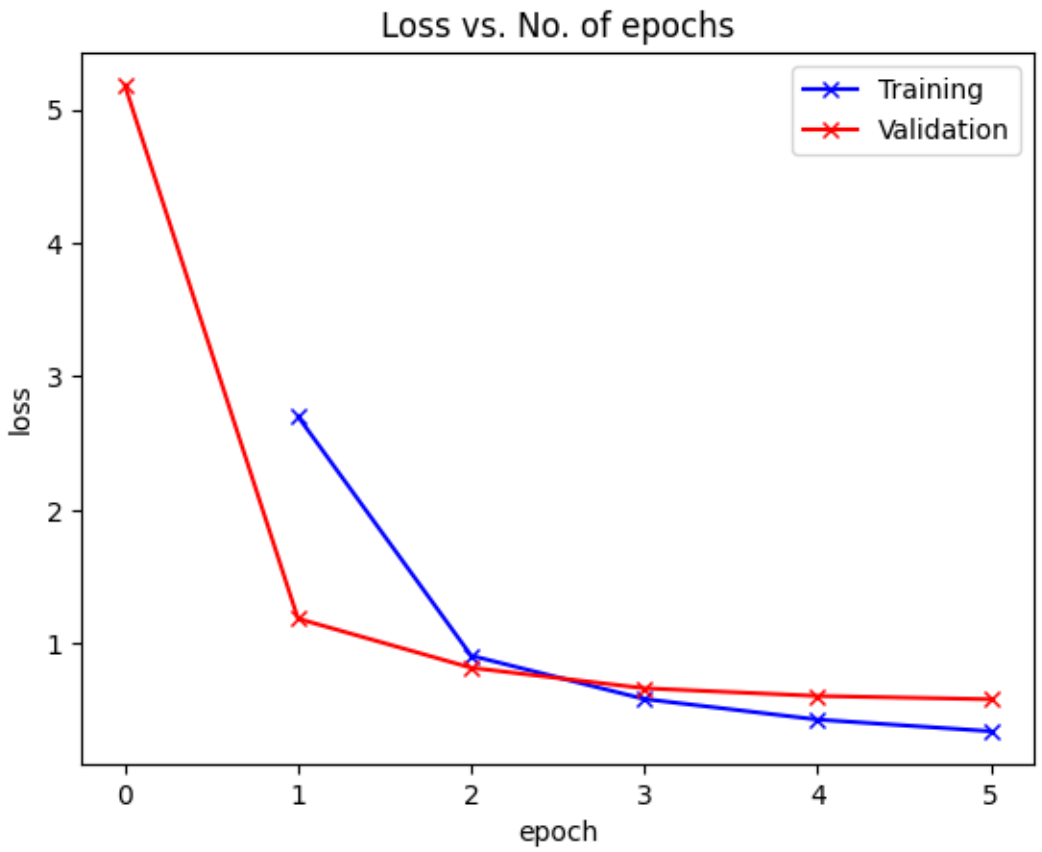
```
  0%|          | 0/334 [00:00<?, ?it/s]
```

```
Epoch [1], train_loss: 0.9085, val_loss: 0.8176, val_acc: 0.9281
```

```
  0%|          | 0/334 [00:00<?, ?it/s]
```

```
Epoch [2], train_loss: 0.5833, val_loss: 0.6642, val_acc: 0.9500
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [3], train_loss: 0.4295, val_loss: 0.6063, val_acc: 0.9438
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [4], train_loss: 0.3422, val_loss: 0.5817, val_acc: 0.9469
```






```
{'val_loss': 0.49581700563430786, 'val_acc': 0.949999988079071}
```

```
 + Number of FLOPs: 3.59G
```

6 Transfer Learning 4 Efficient Net

```
Size of training dataset: 6270
```

```
Downloading:
"https://download.pytorch.org/models/efficientnet_b0_rwightman-3dd342df.pth" to
/root/.cache/torch/hub/checkpoints/efficientnet_b0_rwightman-3dd342df.pth
100%|      | 20.5M/20.5M [00:00<00:00, 220MB/s]
```

```
627
```

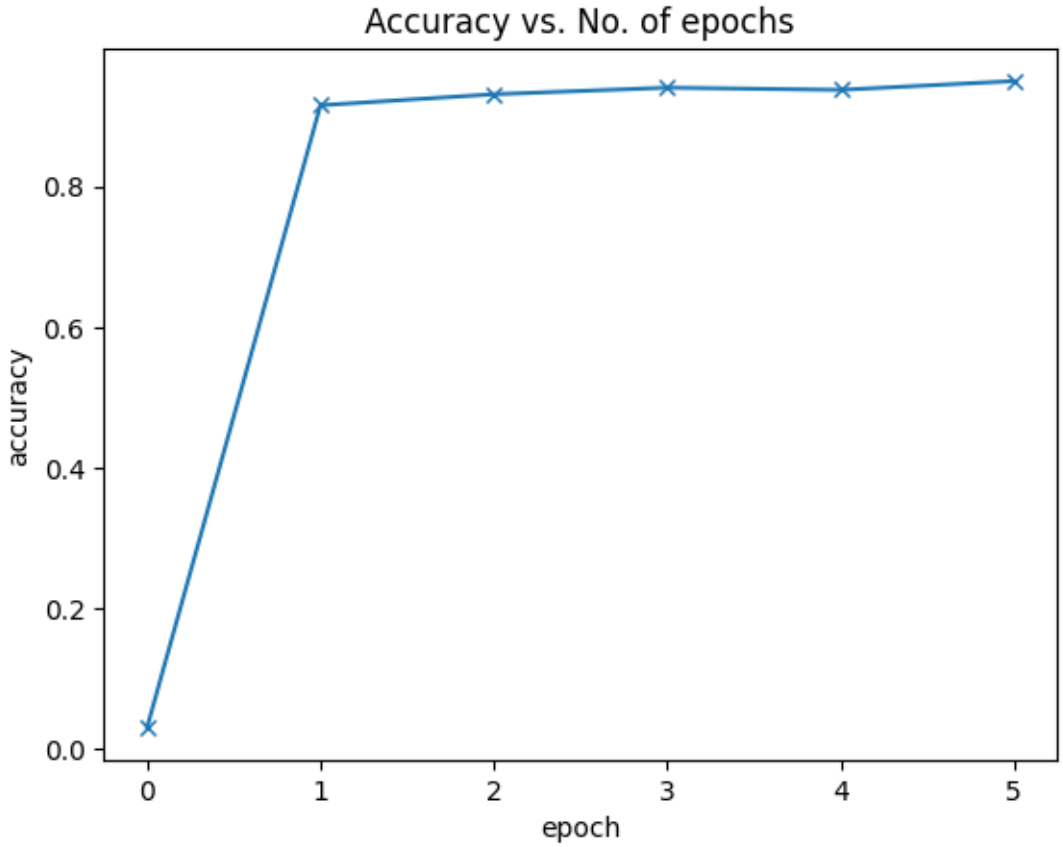
```
  0%|          | 0/334 [00:00<?, ?it/s]
```

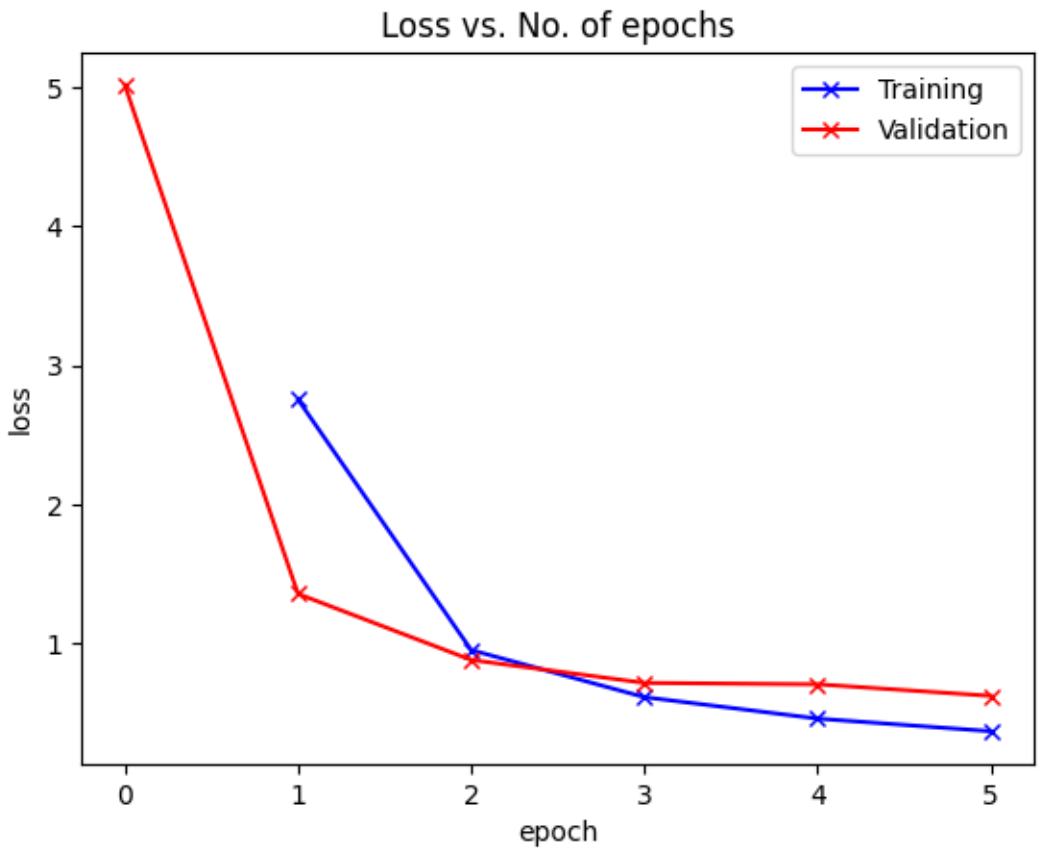
```
Epoch [0], train_loss: 2.7533, val_loss: 1.3541, val_acc: 0.9156
```

```
  0%|          | 0/334 [00:00<?, ?it/s]
```

```
Epoch [1], train_loss: 0.9499, val_loss: 0.8785, val_acc: 0.9313
```

```
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [2], train_loss: 0.6109, val_loss: 0.7134, val_acc: 0.9406
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [3], train_loss: 0.4565, val_loss: 0.7036, val_acc: 0.9375
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [4], train_loss: 0.3657, val_loss: 0.6201, val_acc: 0.9500
```






```
{'val_loss': 0.6366989016532898, 'val_acc': 0.932812511920929}

+ Number of FLOPs: 0.76G
```