MP4

Part1.

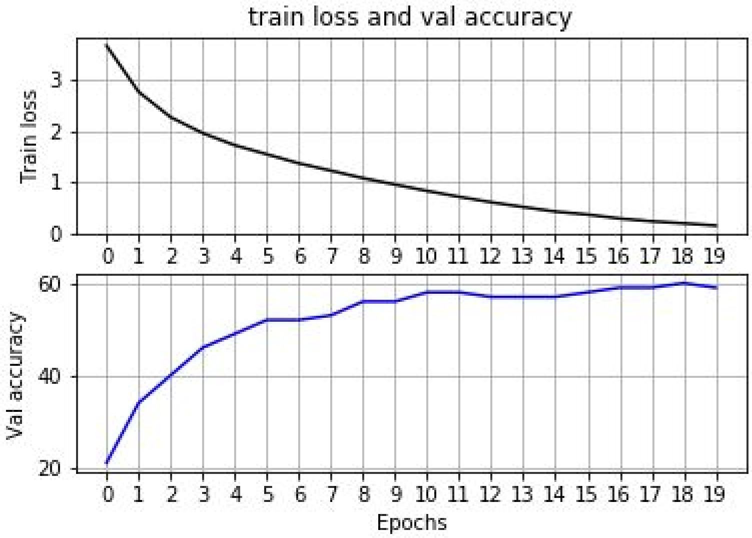
a. My name I used on Kaggle is: Yichen Zhou
b. My best accuracy is 58.700%
c.

Layer NO.	Layer Type	Kernel Size (for conv layers)	Input | Output Dimension	Input | Output Channels (for conv layers)
1	conv2d	5	32 | 32	3 | 32
2	batchnorm2d		32 | 32	32 | 32
3	relu		32 | 32	
4	conv2d	5	32 | 32	32 | 64
5	batchnorm2d		32 | 32	64 | 64
6	relu		32 | 32	
7	maxpool2d	2	16 | 16	
8	conv2d	5	16 | 16	64 | 128
9	batchnorm2d		16 | 16	128 | 128
10	relu		16 | 16	
11	conv2d	5	16 | 16	128 | 256
12	batchnorm2d		16 | 16	256 | 256
13	relu		16 | 16	
14	maxpool2d	2	8 | 8	
15	conv2d	5	8 | 8	256 | 512
16	batchnorm2d		8 | 8	512 | 512
17	relu		8 | 8	
18	conv2d	5	8 | 8	512 | 1024
19	batchnorm2d		8 | 8	1024 | 1024
20	relu		8 | 8	
21	maxpool2d	2	4 | 4	

22	linear		16384 | 1600	
23	relu		1600 | 1600	
24	linear		1600 | 400	
25	relu		400 | 400	
26	linear		400 | 100	

d. One of the factors that helped is the number of layers I put. I used 6 convolutional layers, each is followed by a normalization layer and a RELU layer. After every two conv+norm+relu, I would then put a max pooling layer to reduce spatial size and prevent overfitting.
   Also, I changed the training batch size to 64 and used learning rate of 0.006.

e. The plot:



f. When I dropped the learning rate to 0.0045. The validation accuracy went to 57%. However, the testing accuracy is only 55.3%.
   When I changed the epoch to 20 in this case. The validation accuracy dropped to 56%. Which has caused overfitting.
   But when I chose learning rate at 0.006. Increasing the epoch to 20 get better accuracy.

Part2:

1. For fixed feature extractor:
   Training accuracy: 80.50%
   Testing accuracy: 47.58%

   For fine tuning:
   Training accuracy: 92.47%
   Testing accuracy: 49.72%

2. batch_size: 32
   learning_rate: 0.0045
   RESNET_LAST_ONLY: True for fixed feature extractor, False for fine tuning
   Num_epochs: 50