

COL780:A4: Transfer Learning

Sidharth Agarwal 2019CS50661

May 11, 2023

Libraries Used:

numpy, torch, PIL, torchvision, matplotlib

Run:

To run the python code use the command `python3 main.py {dataDir}`

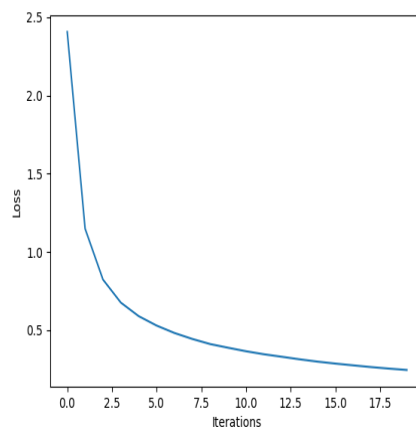
About code:

- For normalization I divided the image tensor by 255.
- The best final hyper-parameters that I chose were `BATCH_SIZE = 64`, `EPOCHS = 5`, `EPSILON = 1e-3`, `REGULARIZATION = True`, `WEIGHT_DECAY = 0`, `DEBUG = True`, `LR = 0.001`, `MOMENTUM = 0.9` and `WEIGHT_DECAY = 1e-5`.
- I changed the number of output features from 4096 to 10, in the last layer.
- I defined my own `ourDataset` class to load the dataset as pytorch dataset.
- For regularization, I implemented L2 regularization with the above weight decay. And as well as early stopping with the difference between consecutive loss as the terminating criteria with threshold as EPSILON.

Results

The results before regularization were as follows.

Final Training Accuracy == 94.88
Final Validation Accuracy == 93.60
Final Testing Accuracy == 92.80



The above graph is training loss convergence without regularization.
The results after regularization obtained were as follows.

Final Training Accuracy == 94.04
Final Validation Accuracy == 94.57
Final Testing Accuracy == 93.64