Data Augmentation

- Bhagyesh Gaikwad.

Used learning rate of 0.001, batch size of 32 and ran it for 100 epochs(32,000 iterations in code) 200 epochs(64,000 iterations in code) .

Report of Final Test Accuracy with 100 and 200 epochs respectively:

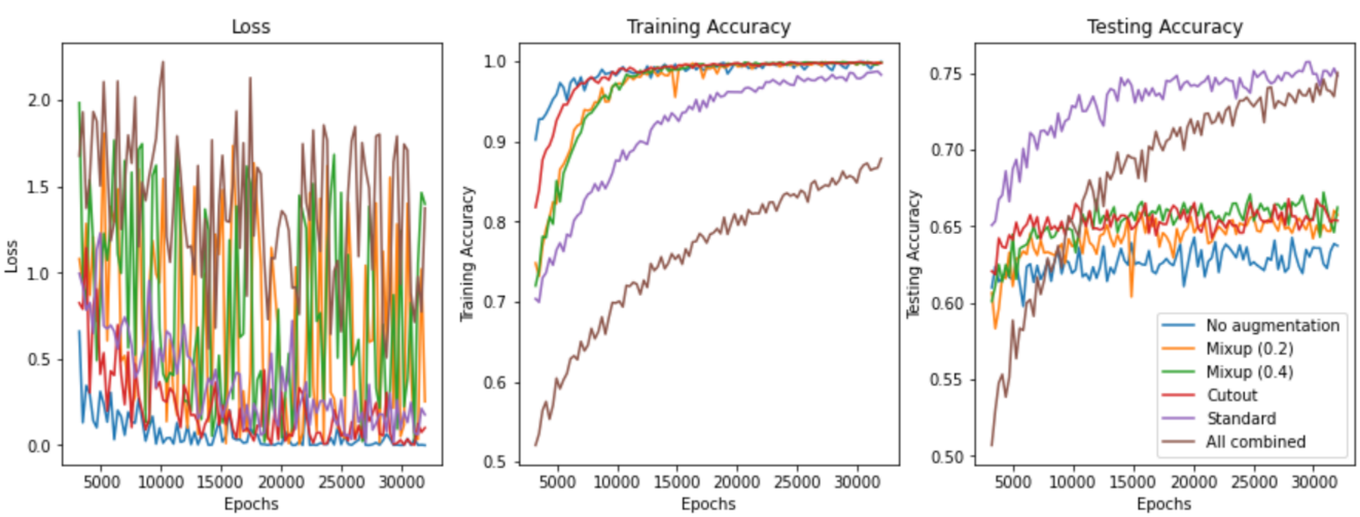
1) Resnet model without augmentation: 64.27, 64.84
2) Mixup with alpha = 0.2: 66.03, 67.12
3) Mixup with alpha = 0.4: 66.22, 67.73
4) Cutout with K = 16: 66.79, 66.74
5) Standard with K = 4: 75.73, 76.35
6) All Combined (as stated in assignment): 75.02, 76.91

Applying standard and cutout augmentations on the training images and then apply mixup to blend them:

- Chose alpha = 0.4 is for are combined augmentation as it performed better than 0.2.
- Combining all methods certainly gave better results than mixup and cutout, but performance is similar to standard for 100 epochs, and a little better for 200 epochs.

Role of data augmentation:

For 100 epochs:



For 200 epochs:

- We can see from the above graph that training loss converges slowly for implementations with augmentations than with no augmentation.
    - Adam optimizer also plays a part in spikes that we see in the loss graph
- Eventually every implementation reaches 99% train accuracy (except for combined, but the trend in both graphs suggests with more epochs we can get it to 99%).
    - Additionally as seen in the graph, implementation with augmentation needs more epochs to reach better accuracy.
    - This intuitively makes sense since augmentation makes it difficult to overfit and would eventually give better results.
- Test accuracy is lower than training by a good margin. But here we can see that implementations with augmentations have performed better than no augmentation.
    - ‘Combined’ augmentation has performed better than others with max test accuracy in 200 epochs.