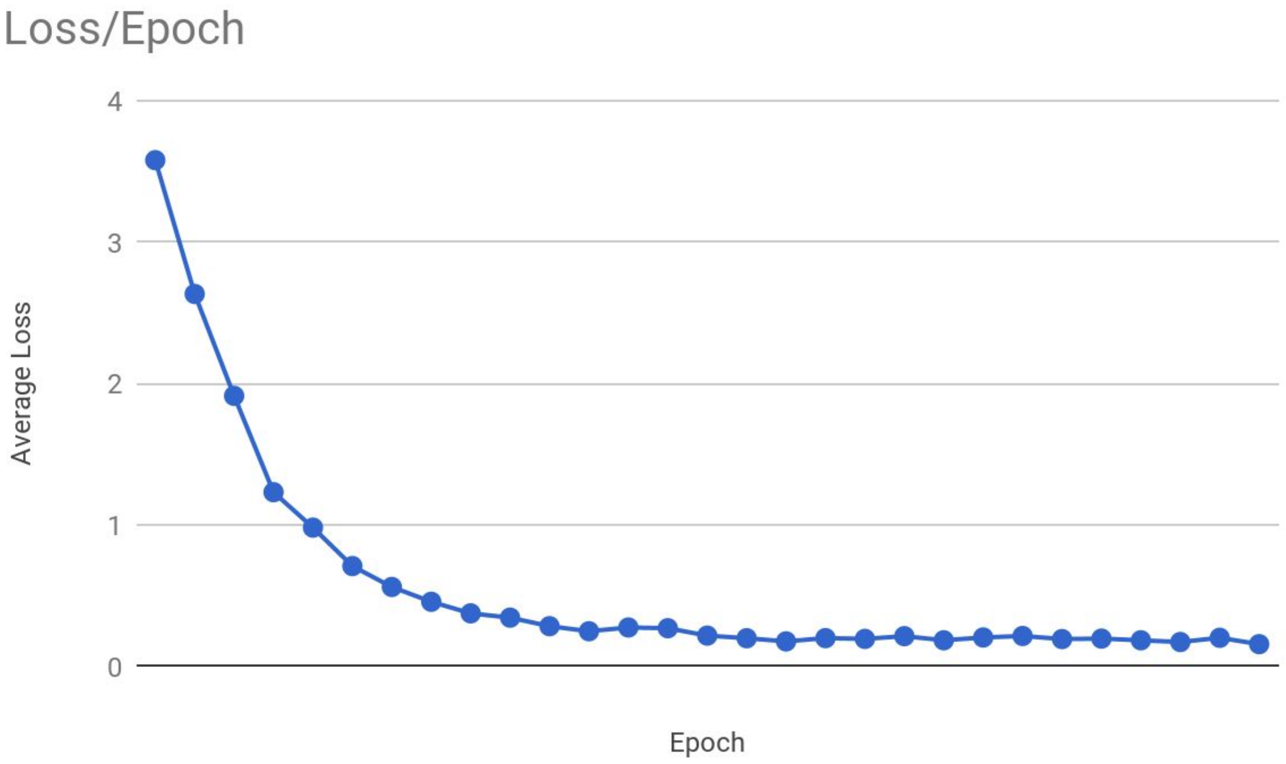
CV HW3

Chc631 Cheng-Hsi Chao

Baseline CNN Model

The first thing I tried was to add more layers and played around with kernel sizes.

A testing of both speed and result found that a kernel size of 5x5 with 3 convolution layers and 2 fully connected layer with the default learning rate can achieve 97% Acc in 30 epochs



Attempted Improvement

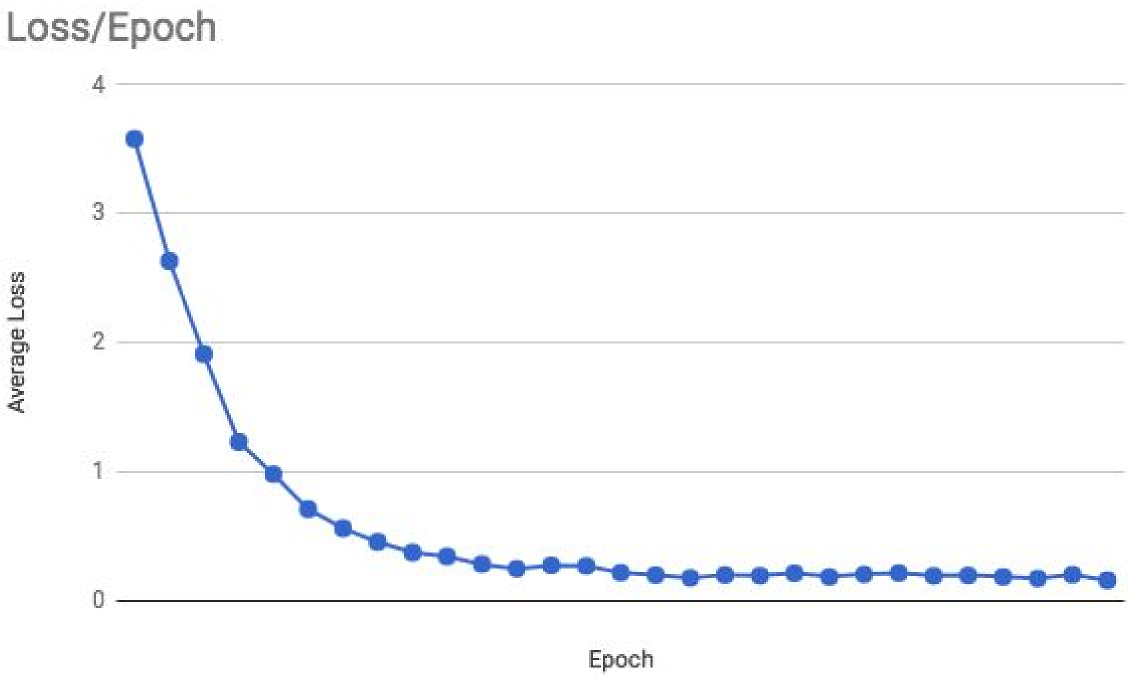
An improvement can be found in a model proposed by Vivek Yadav in the blog post https://chatbotslife.com/german-sign-classification-using-deep-learning-neural-networks-98-8-solution-d05656bf51ad
Which says that by expanding the training with jitterred image through affine transformations, such as rotations, translations and shearing and also brightness augmentations, a improvement of 99% can be achieved.

The steps I have taken to implement this is as follows:

1. Duplicate multiple copies of files through bash script
2. Pipleine through custom Transform function that randomizes affine transformations
    a. The function is manual implemented within the torchvision package transforms.py and functional.py
3. Use the ColorJitter method in the transform class to randomize brightness

When the expanded test data is loaded through the dataloader, the transforms will reshape it in randomized angles as a new data point to learn from. 282712 training files are used, 8 times larger than the original

The resulting accuracy was not optimal, in fact, far from:


Loss/Epoch
Average Loss
Epoch