

Traffic Sign Detection

1. As the original dataset was small for deep neural network. Hence I augmented the data by adding more images which were randomly translated, rotated and added shear to them. Also the classes that had less images, augmentation was done more number of times for their each image. The code for the same is in Augmentation.ipynb. So total images in training were 79835
2. The parameters that I changed in main.py were

No of Epochs – 25
Learning rate – 0.01
Momentum – 0.7
Batch Size -128

Also added a function to store train accuracy after every epoch. The values of train accuracy are stored in a pickle file, which is used later for visualization

3. Convolution Training Network

Layer 1 CNN

Input 32 x 32 x 3
Kernel 5 x 5
Padding = 2
Output 32 x 32 x 32

Layer 2 Max Pool (2,2)

Output 16 x 16 x 32

Layer 3 ReLu

Layer 4 CNN

Input 16 x 16 x 32
Kernel 5 x 5
Padding = 2
Output 16 x 16 x 64
Dropout with p =0.1

Layer 5 Max Pool (2,2)

Output 8 x 8 x 64

Layer 6 ReLu

Layer 7 CNNInput $8 \times 8 \times 64$ Kernel 5×5

Padding = 2

Output $8 \times 8 \times 128$ Dropout with $p = 0.3$ **Layer 8 Max Pool (2,2)**Output $4 \times 4 \times 124$ **Layer 9 ReLu****Flatten to $128 \times 4 \times 4$** **Layer 10 Fully connected**

Input = 2048

Output = 128

Layer 11 ReLu

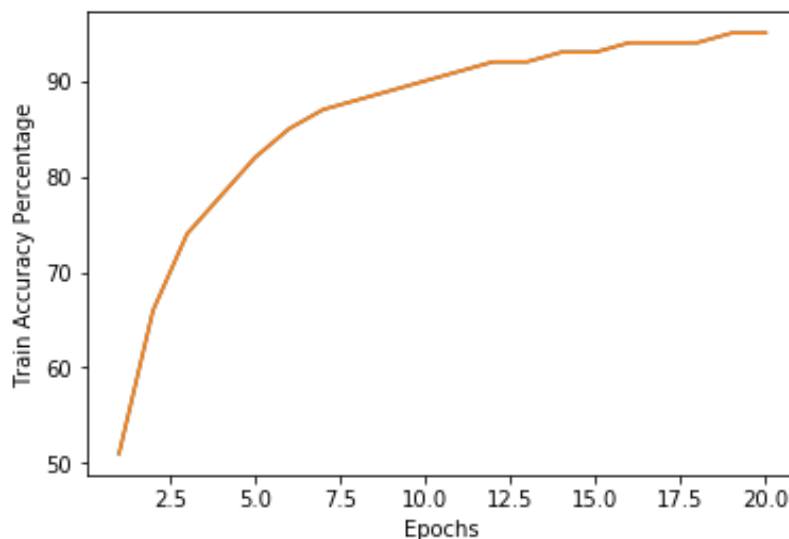
Dropout

Layer 12 Fully connected

Input = 128

Output = 43

I calculated accuracy (percentage of correct in training set) for 20 epochs and the plot is as follows-



The accuracy of validation came to be 99% and Train to be 95% after 20 epochs