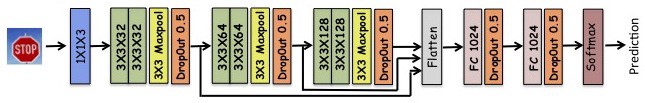
**Assignment 2 –GTSRB**

Overall, I achieved 98.115% on the test dataset and 99.76% on training dataset using the below model and with some data transformation tricks while training the model.

**The Model:**

Inspired by the architecture from a blog [1], I Implemented and trained the model from scratch. After training and implementing various models, this particular architecture worked best on the test data set.



**Source** [1] : <https://medium.com/@vivek.yadav/improved-performance-of-deep-learning-neural-network-models-on-traffic-sign-classification-using-6355346da2dc>

In most applications, changing the color map can result is significant improvement in the performance. We let the model decide which color works best by using 3 1x1 filters.

The architecture uses three different convolutional modules of 32, 64, 128 3x3 filters respectively followed by max pooling and dropouts whose output are fed into fully connected layer.

The idea is that fully connected layer has access to outputs from lower-level and higher-level filters and has the ability to choose the features that works the best.

Further the two fully connected layers have 1024 neurons in each, followed by a softmax layer to compute the probability of each class.

**Preliminary Results:**

Training the model for 20 epochs resulted in 99.76% accuracy on validation dataset but not very surprisingly 94% on the test dataset.

Reason being the imbalance in dataset and some distortions in the original images.

**Data Augmentation:**

Data augmentation helped to boost the accuracy on the test dataset. On exploratory analysis, I saw there is imbalance in dataset, some of the classes has 2000 samples where as others have only 200 samples. Hence to balance the classes, I augmented 10000 samples for each image.

**Data Transformation:**

This trick is supposed to improve the accuracy a lot as referenced on various blogs. The brightness of the image plays a huge role. Hence, I used Affine transformations and colo\_jitter while training the model to get a feel of the test dataset.

After running through 100 epochs the model’s accuracy improved over the test dataset and I achieved 98.115%.

**Training Accuracy vs epochs:**

**Link to the Model:**

[**https://drive.google.com/open?id=1hUbrgt\_vhJno4MA0w8l9ctv4Bw23hviF**](https://drive.google.com/open?id=1hUbrgt_vhJno4MA0w8l9ctv4Bw23hviF)

**Other models considered:**

A simple 3 layered convolutional model, which I ran without transforming the dataset yielded 92% accuracy on test data.

VGG16 model which resulted in ~93% accuracy.

Resnet18, for some weird reason the model kept on diverging. On retrospect, I believe it may be due to the fact that the data wasn’t in the range [0, 1] as desired by the model.