Kaggle Competition - GTSRB

Shubham Chandel - sc7238@nyu.edu

October 26, 2018

1 Introduction

The German Traffic Sign Benchmark (GTSRB) [1] is a multi-class, single-image classification challenge hosted on Kaggle. As a part of the assignment, we are required to train our image classifier to perform well on the hidden test dataset on Kaggle.

Our model is a parameterized Deep Neural Network which is able to achieve a test accuracy of 99.683% on the hidden test dataset on the public leaderboard.

The results are presented in this report and the code along with the models are available here.
https://github.com/sksq96/pytorch-gtsrb

2 Dataset and Augmentations

The provided dataset presents us with a number of challenges, including but not limited to, class imbalance, deformation, and the dataset itself being small in size.

To get a better understanding of the dataset, as an exercise, try to understand the content of the Figure 1.

(a) (b) (c) (d)

Figure 1: Sample of images from the GTSRB dataset.

A number of train-time data augmentations techniques are used to tackle these issues. All these aim to make the training data difficult to memorize by the network, thereby the nature of these augmentations are to distort the training data.

However, we only augment the image with a probability of 50%. That is, for every image in the train dataset, a fair coin flip is used to determine if the image will be augmented or not.

We used the imgaug python library [3] to perform all the data augmentations. From the following list of data augmentations, at random, k are selected and performed in random order if the train image is available for augmentations.

1. Brightness Reduction
2. Sharpen
3. Contrast Normalization
4. Shear Affine
5. Rotation Affine

One important observation is, a lot of images are take in bad lightning conditions. To solve this, we use only one channel Y from YCbCr color space as proposed in [4].

3 Models and Techniques

3.1 Models

We implemented the following architectures, all of them belonging to the class of parameterized Deep Neural Networks, described in brief below:

1. Two Layer Convolutional Neural Network

   The original CNN provided as starter code [2], was trained to 50 epochs. This small model was able to achieve 92% accuracy on the test dataset.

2. Resent-50

   The deep residual network [5], with 50 layers, balanced the right trade-off between number of parameters and the achieved final accuracy. This model alone, was able to push the test dataset accuracy to 98%.

3. DenseNet-169

   The densenet architecture [6], with 169 layers performed better than the previous resnet, with the test dataset accuracy on the public leaderboard to 99%.

4. Ensemble

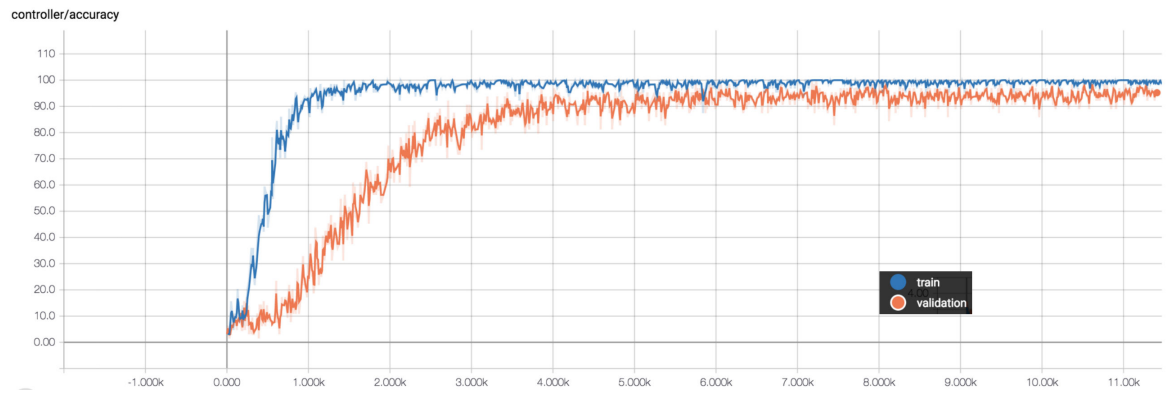
   The final model included model ensemble from 5 different models, resent-34, resent-50, densenet-121, densenet-169 and resent-34 (with RGB input), with 3 checkpoints from each of these models.

   Finally, these 15 checkpoints were used to get a diverse opinion on each test image, and a majority voting scheme was used to classify each test image.

   This ensemble from the group of models achieved an accuracy of 99.683% on the test dataset on the public leaderboard.

2 Layer CNN	ResNet-50	DenseNet-169	Ensemble
92.89%	98.48%	99.07%	99.68%

Table 1: Test accuracy for different models on public leaderboard.



(a)

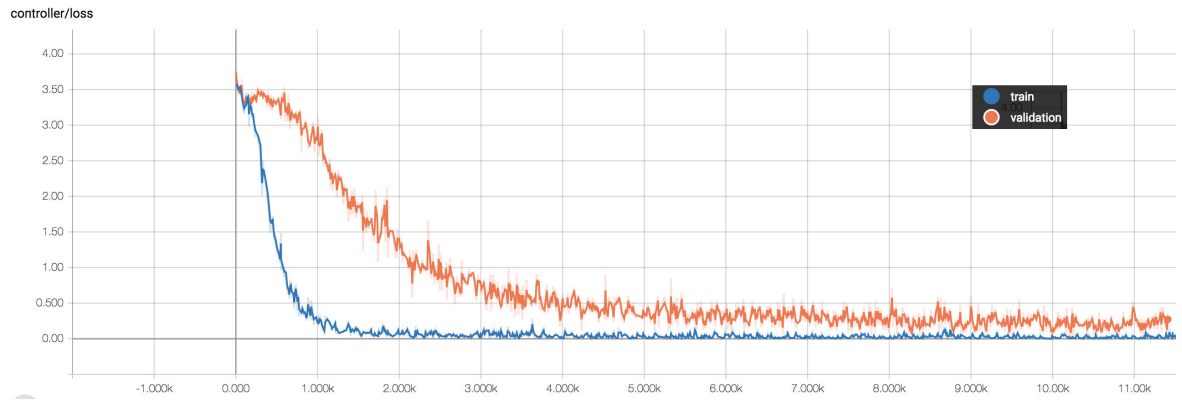
Figure 2: Training and Validation accuracy as a function of timestep.

3.2 Hyperparameters

The models were trained using Adam optimizer [7], with learning rate of 1e-3, cross entropy loss and dropout of 0.5. Validation cross entropy loss was minimized to find the best model, as we want our model to be confident of its prediction so that it can do well on generalized test set.

4 Conclusion

We used ensemble of diverse Deep Neural Networks with the majority voting approach to take into account each prediction. The kind of data imbalace present by GTSRB dataset was tacked by using a variety of data augmentation. We achieve the final accuracy of 99.68% on the test dataset.



(a)

Figure 3: Training and Validation loss as a function of timestep.

References

[1] German Traffic Sign Recognition Benchmark (GTSRB) http://benchmark.ini.rub.de

[2] Assignment 2: Traffic sign competition https://github.com/soumith/traffic-sign-detection-homework

[3] Image augmentation for machine learning experiments. https://github.com/aleju/imgaug

[4] Traffic sign recognition with multi-scale convolutional networks http://yann.lecun.com/exdb/publis/pdf/sermanet-ijcnn-11.pdf

[5] Deep Residual Learning for Image Recognition https://arxiv.org/abs/1512.03385

[6] Densely Connected Convolutional Networks https://arxiv.org/abs/1608.06993

[7] Adam: A Method for Stochastic Optimization https://arxiv.org/abs/1412.6980