

# Traffic Sign Classifier

---

**Author: Matt Kontz**

**Date: March 2, 2019**

---

## Build a Traffic Sign Recognition Project

The goals / steps of this project are the following:

- Load the data set (see below for links to the project data set)
- Explore, summarize and visualize the data set
- Design, train and test a model architecture
- Use the model to make predictions on new images
- Analyze the softmax probabilities of the new images
- Summarize the results with a written report

output\_images/randomSigns.png

Learning\_Curve\_0p80

---

## Rubric Points

---

Here I will consider the [rubric points](#) individually and describe how I addressed each point in my implementation.

## Dataset Exploration

### Dataset Summary

**Criteria:** The submission includes a basic summary of the data set.

- Number of training examples = 34799
- Number of testing examples = 12630
- Image data shape = (32, 32, 3)
- Number of classes = 43, (0-42)

**Criteria:** The submission includes an exploratory visualization on the dataset.



## Design and Test a Model Architecture

### Preprocessing

**Criteria:** The submission describes the preprocessing techniques used and why these techniques were chosen.

For preprocessing this only thing done was shuffling the data. The data was also shuffled between each EPOCH. This help to ensure that the same type of data is not concentrated in a certain batch.

Unlike for character recognition, the input image was not converted to gray scale, but the color information is an important part of recognizing the signs.

## Model Architecture

**Criteria:** The submission provides details of the characteristics and qualities of the architecture, including the type of model used, the number of layers, and the size of each layer. Visualizations emphasizing particular qualities of the architecture are encouraged.

The LeNet CNN for the class exercise was used as a starting point.

Changes to the LeNet model used in the class exercise:

- input size had to change to account for the 3 color layers
- output size had to change to account for 43 classes
- Drop out was added through the LeNet model for regularization.

## Model Training

**Criteria:** The submission describes how the model was trained by discussing what optimizer was used, batch size, number of epochs and values for hyperparameters.

- The batch size was set to 128 and not changed.
- The Adam optimizer was used.
- Regularization was done via drop-out: final keep probability as 0.8 (80%).
- Learn rate started at 0.001, ended at 0.00025
- Final number of Epoch was set to 200.

## Solution Approach

**Criteria:** The submission describes the approach to finding a solution. Accuracy on the validation set is 0.93 or greater.

Initially the learn rate was 0.001, starting out with the basic LeNet model, the data set fit pretty well, but the model was over fitting as seen in the learning curve below.

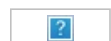
This implies

\* the model is inadequately representing the data, the model is over fitting the data's features.



This implies the model is inadequately representing the data, the model is over fitting the data's features.

The next step was to add regularization to minimize over fitting. Dropout was selected as the method for regularization and the training data was used with dropout with a keep probability ranging between 50% and 100% in increments of 5% for 100 Epochs.



At 50% dropout the data was having a hard time converging, but the overfitting was reduced since the accuracy/error of the validation data tracked the training pretty closely. A keep probability of 80% seemed to be the best trade-off between regularization and speed of training.

To improve the final training, the number of Epoch was increased to 200 and learning rate was reduced to 0.00025. After about 100 Epochs the validation accuracy reaches 93% and continues

increase reaching about 95% after 200 Epochs. The training was still had some over fitting of data, but the over fitting was improved with the 80% keep probably.

The accuracy with the test data set was a little less than the validation set.

**Validation Accuracy: 95.1%, error: 4.9%.** (after last Epoch)

**Test Accuracy: 93.7%, error: 6.3%.** (after training completed)



## Test a Model on New Images

### Acquiring New Images

**Criteria:** The submission includes five new German Traffic signs found on the web, and the images are visualized. Discussion is made as to particular qualities of the images or traffic signs in the images that are of interest, such as whether they would be difficult for the model to classify.



Once the images are resize to 32x32x3, the images look a bit grainy which is similar to the training data. One thing to notice is the main different between the 30kph and 70kph sign is the numbers in the middle of the red circle, the shapes and color on the other signs are more unique and should be easier to classify.

### Performance on New Images

**Criteria:** The submission documents the performance of the model when tested on the captured images. The performance on the new images is compared to the accuracy results of the test set.

One of the six images classified incorrectly which is a lower accuracy than the training and validation sets in general

Predicted: [18 0 25 1 38 33]

Expected: [18 4 25 1 38 33]

6 Random signs from internet: accuracy: 83.3%, error: 16.7%

### Model Certainty - Softmax Probabilities

**Criteria:** The top five softmax probabilities of the predictions on the captured images are outputted. The submission discusses how certain or uncertain the model is of its predictions.

Image #1, sign #18 - General caution

- Probability: 1.000, sign #18 - General caution
- Probability: 0.000, sign #26 - Traffic signals
- Probability: 0.000, sign #28 - Children crossing
- Probability: 0.000, sign #11 - Right-of-way at the next intersection
- Probability: 0.000, sign #27 - Pedestrians

Image #2, sign #4 - Speed limit (70km/h)

- Probability: **0.687**, sign #0 - Speed limit (20km/h)
- Probability: **0.286**, sign #4 - Speed limit (70km/h)
- Probability: 0.013, sign #1 - Speed limit (30km/h)

- Probability: 0.011, sign #8 - Speed limit (120km/h)
- Probability: 0.002, sign #2 - Speed limit (50km/h)

Image #3, sign #25 - Road work

- Probability: 1.000, sign #25 - Road work
- Probability: 0.000, sign #22 - Bumpy road
- Probability: 0.000, sign #26 - Traffic signals
- Probability: 0.000, sign #29 - Bicycles crossing
- Probability: 0.000, sign #18 - General caution

Image #4, sign #1 - Speed limit (30km/h)

- Probability: **0.502**, sign #1 - Speed limit (30km/h)
- Probability: **0.498**, sign #2 - Speed limit (50km/h)
- Probability: 0.000, sign #0 - Speed limit (20km/h)
- Probability: 0.000, sign #3 - Speed limit (60km/h)
- Probability: 0.000, sign #5 - Speed limit (80km/h)

Image #5, sign #38 - Keep right

- Probability: 1.000, sign #38 - Keep right
- Probability: 0.000, sign #34 - Turn left ahead
- Probability: 0.000, sign #36 - Go straight or right
- Probability: 0.000, sign #0 - Speed limit (20km/h)
- Probability: 0.000, sign #1 - Speed limit (30km/h)

Image #6, sign #33 - Turn right ahead

- Probability: 1.000, sign #38 - Keep right
- Probability: 0.000, sign #34 - Turn left ahead
- Probability: 0.000, sign #36 - Go straight or right
- Probability: 0.000, sign #0 - Speed limit (20km/h)
- Probability: 0.000, sign #1 - Speed limit (30km/h)

Notice that the two sign related speed limits where the one with the least confidence. The 70km/h sign misclassified and the 30km/h sign had almost the same probability as 50km/h. This makes sense because, the small blurry images look very similar.