Files Submitted

Submission Files

The following files are included in this submission:

1. Traffic_Sign_Classifier.ipynb
2. report.html
3. Internet Image Files (5 total)
4. Writeup Document

Dataset Exploration

Dataset Summary

The data set included in this deep learning model includes over 30,000 German traffic sign images. These traffic signs vary in many ways including shape and color. Signs include ‘Roundabout Mandatory’, varying Speed Limit signs, and Hazard Signs such as ‘Road Work’. Each of these images can be classified in one of 43 different classes defined within the CSV file.

Exploratory Visualization

Please see project file for visuals. Visuals include a display on one of each sign classification in the dataset and a breakdown of the number of samples provided in each sign classification.

Design and Test a Model Architecture

Preprocessing

Preprocessing includes converting the images from RGB to Grayscale. Although the traffic signs can differ in color, the main identifying qualities of the sign are based on shapes and numbers. In real life, signs have to be unique enough for even color-blind drivers to read properly. As such, to avoid introducing unnecessary complexity to the deep learning architecture, the images are converted to grayscale prior to being processed.

Model Architecture

I decided to base my architecture off of LeNet. Therefore, the model starts with four convolutional layers that increase the channels up to 160. In between these convolutions include MaxPool activations to decrease the size. Each of the convolutional layers utilize a filter size of 2x2 with the exception of the last one at 3x3. These convolution layers are followed by three fully-connected layers which decrease the number of neurons from 5760 to 1024, then 1024 to 256, and finally 256 to 43, which is the number of traffic sign classes. Prior to reaching the logits, I added a Dropout function to help prevent overfitting of the data with a keep probability of 0.5.

Model Training

Used in this architecture was an AdamOptimizer. The batch size used was 128 with 10 epochs. Using more Epochs I felt would overfit the data too much and also resulted in no change to the overall accuracy. 0.001 was used as the learning rate and, from my experience, diverting from that value resulted in negative effects on the accuracy. Mu and Sigma for the truncated_normal function was defaulted to 0 and 0.1 respectively.

Solution Approach

I based the architecture off of LeNet and started off with some larger filter sizes for the convolution layers. The convolutional layers are important as they help to share weights as filters across the image. These layers then highlight various features such as shapes, which is important in identifying traffic signs. MaxPooling is added to help simplify the data and reduce the dimensions of the network. This is again to help highlight key features of the data, which is helpful as traffic signs typically have a few key distinct features to identify them. This resulted in a validation accuracy less than 90%. I then lowered the filter sizes and increased the channels which helped to break 90% accuracy. I actually made a mistake and evaluated my validation set with the Dropout layer at a keep probability of 0.5. Changing this value to 1 during validation operations resulted in around 95% accuracy. The Dropout function is to prevent overfitting of the model as we want to be able to identify traffic signs in various conditions (important for self-driving cars in different weather conditions or night vs day). Messing around with the Epochs resulted in no change in accuracy beyond 10 and less accuracy at around 5. Therefore, I kept the Epoch at 10. The final result by testing my “test” dataset resulted in over 93% accuracy.

Test a Model on New Images

Acquiring New Images

Doing a few searches online netted me five different traffic signs that were not a part of the dataset. I chose signs that were tilted or even rotated to really push my architecture's capabilities. For instance, the Stop sign is rotated slightly, the speed limit sign has additional information beneath it, and the roundabout sign is tilted and has a shadow on it.

Performance on New Images

The model performed well on these images and got an accuracy of 80% (i.e. 4 out of 5 correct). Listed in the project is both the correct label and model prediction on the internet images. The accuracy of the Traffic Sign Dataset was around 95% which might suggest that the model is overfitting. However, the mislabeled traffic sign correctly identified the sign type (as a Speed Limit sign) but mistook it for 50km/h instead of 30km/h. I believe adding more samples of speed limit signs to be trained might assist the model in correctly identifying the difference in numbers on the signs.

Model Certainty - Softmax Probabilities

For the four correct predictions, the certainty of the model was pretty high as the prediction identifier was very close to being 1. The one prediction that was incorrect (Image #2) actually demonstrated uncertainty as two different labels were within equal probabilities.