Files Submitted

CRITERIA

MEETS SPECIFICATIONS

Submission Files

[Download Jupyter Notebook from GitHub]

Note: traffic sign images were too large to upload to GitHub

Dataset Exploration

CRITERIA

MEETS SPECIFICATIONS

Dataset Summary

The submission includes a basic summary of the data set.

Number of training examples = 34799
Number of testing examples = 12630
Number of validation examples = 4410
Image data shape = (32, 32, 3)
Number of classes = 43

Exploratory Visualization

The submission includes an exploratory visualization on the dataset.

Design and Test a Model Architecture

CRITERIA

MEETS SPECIFICATIONS

Preprocessing

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

Preprocessing including normalizing data to range 0-1.

I did not convert to grayscale as I wanted to see results in RGB.

CRITERIA

MEETS SPECIFICATIONS

Model Architecture

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.

Input Layer 1: Convolutional. Input = 32x32x1. Output = 28x28x6
Activation Layer 1: Relu
Pooling Layer 1: 2x2 kernel size, 2x2 stride, valid padding. Output = 14x14x6

Layer2: Convolutional. Output = 10x10x16
Activation Layer 2: Relu
Pooling Layer 2: 2x2 kernel size, 2x2 stride

Flatten output

Fully Connected Layer 1: y = xW + b
Activation: Relu
Dropout Layer: 80% keep probability

Fully Connected Layer 2: y = xW + b
Activation: Relu
Dropout Layer 80% keep probability

Output Layer: Fully Connected Input = 84. Output = 43
Fully Connected Layer Output: y = xW + b

Model Training

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

Submission #1
I chose the following model parameters:

- Epochs = 15 (due to limited computer resources)
- Batch Size = 128
- AdamOptimizer was used to minimize loss with learning rate of 0.001

Solution Approach

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

Submission#3
Please note that in order to meet this rubric a little more information why you chose these techniques is needed. (please update the writeup for this section)

Parameter	Why Chosen
Epochs = 20	Increase the training the model of the model
Batch Size = 128	I chose this batch size to better train the model and cycle through Epochs faster.

CRITERIA MEETS SPECIFICATIONS

AdamOptimzer	I selected this to optimize the model in fewest number of steps
Learning Rate = 0.001	I chose this to minimize accuracy fluctuations; smaller steps in the gradient descent. This worked best after trial/error
Dropout rate = 20%	I chose this because the model was over-optimizing for the training set and accuracy was not getting better despite tuning the parameters
Preprocessing: Adaptive Histogram Equalization	After inspecting the images, I noticed that some information was lost due to brightness or image focus. I researched the OpenCV docs and found this technique to maximize local contrast so both the background and foreground can be seen.
Grayscale	I chose this to reduce the channels down to 1 for better processing

Submission#2
I improved the accuracy of the model by changing images (converted to single channel grayscale, maximized contrast, and normalized)
I chose the following model parameters:

- Epochs = 20 (increased to better train model)
- Batch Size = 128
- AdamOptimizer was used to minimize loss with learning rate of 0.001

EPOCH 17 ...
Validation Accuracy = 0.926

EPOCH 18 ...
Validation Accuracy = 0.928

EPOCH 19 ...
Validation Accuracy = 0.932

EPOCH 20 ...
Validation Accuracy = 0.936

Test Accuracy = 0.925

Submission#3

Please note that in order to meet this rubric a description on your approach for achieving this model is required. If the model was achieved through trial and error, please explain in the README what your initial approach was and which parameters were tuned in order to achieve the final model

Answer

CRITERIA

MEETS SPECIFICATIONS

Initial approach was to run the images ‘‘as is’’ with minimum normalization. This didn’t produce very high accuracy.

I achieved better accuracy by preprocessing the images and trying different learning rates (.01, .005, .002, .001).

I also played with the batch sizes. I noticed that smaller batches didn’t work as well for the model accuracy.

Additionally, I increased the number of epochs, starting from 5, 10, 15, 20, 25. I settled on 20 when I saw accuracy wasn’t improving.

Submission #1
To improve the accuracy of the model, I would convert the images to grayscale, augment the dataset with more imagesl Increased the number of epochs. Model had Test Accuracy = 0.864 Training Accuracy = 0.965 in gray scale.

Test a Model on New Images

CRITERIA	MEETS SPECIFICATIONS
Acquiring New Images	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. Collected 5 images. The images are downloaded from the internet. The model seems to work fine when the sign is central and focused. It doesn't seem to perform well when the sign does not occupy the vast majority of the image.
Performance on New Images	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. Accuracy of new images: 60.0 %(3 out of 5) Test Accuracy = 87.3%

CRITERIA	MEETS SPECIFICATIONS

Model Certainty - Softmax Probabilities	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.
	`Top 5 values of 13: [ 1.  0.  0.  0.  0.] / indices [13  0  1  2  3]`
	`Top 5 values of 17: [ 1.  0.  0.  0.  0.]/ indices [17  0  1  2  3]`
	`Top 5 values of 25: [ 1.  0.  0.  0.  0.] / indices [25  0  1  2  3]`
	`Top 5 values of 29: [ 1.  0.  0.  0.  0.] / indices [17  0  1  2  3]`
	`Top 5 values of 33: [ 1.  0.  0.  0.  0.] / indices [35  0  1  2  3]`

Awesome job using local contrast maximization, grayscale, and normalization.

Please note that in order to meet this rubric a description on your approach for achieving this model is required. If the model was achieved through trial and error, please explain in the README what your initial approach was and which parameters were tuned in order to achieve the final model