Machine Learning Engineer Nanodegree

Capstone Proposal

Abdalla Elshikh
March 2nd, 2019

Using Deep learning for classifying traffic signs

Domain Background

One of the fields that has always picked my attention is self-driving cars, which is one of the main reasons I decided to take this program. Even though there's has been a lot of experiments conducted on self-driving cars since the 1950s, but recently this field has seen drastic improvements in the past couple of years.

Doing such task with the traditional image processing approaches would be impossible as we would have to detect very complex patterns within the images to differentiate between different traffic signs. On the other hand, training a convolutional neural network to do this task is the more appropriate approach because we don't need to provide those patterns for the network manually as it is able to pick up on those complex patterns during the training process.

Problem Statement

(approx. 1 paragraph)

Classifying traffic signs has always been an integral task for developing self-driving cars. The idea of autonomous vehicles relies heavily on the vehicle being able to interpret traffic signs and accordingly decide what is the appropriate action to take. This can be approached by utilizing deep learning techniques to train a model to identify traffic signs. Autonomous driving promises more than just saved time, it's enough to make society reassess its concepts of time and space. Self-driving cars could also improve safety as the Transport and Road Research Laboratory estimated autonomy could prevent around 40 percent of accidents. Unlike humans, self-driving cars benefit from the experiences of all the self-driving cars out on the road whereas humans learn to drive from scratch.

Datasets and Inputs

This project is trained on images from the Belgium Traffic signs dataset (**BTSD**). This dataset consists of 4,591 training images, 2,534 testing images and is composed of 62 classes.

The images in this dataset vary in size and orientation of the traffic signs which makes it appropriate to train our model on without needing to augment our data and guarantees that our model will generalize well to the data without overfitting. The training images are then to be fed into a CNN that will give the corresponding traffic sign label as output.

BTSD dataset:

https://btsd.ethz.ch/shareddata/

Solution Statement

(approx. 1 paragraph)

In this section, clearly describe a solution to the problem. The solution should be applicable to the project domain and appropriate for the dataset(s) or input(s) given. Additionally, describe the solution thoroughly such that it is clear that the solution is quantifiable (the solution can be expressed in mathematical or logical terms) , measurable (the solution can be measured by some metric and clearly observed), and replicable (the solution can be reproduced and occurs more than once).

As mentioned, this project trains a deep learning model, namely a CNN, on the Belgium Traffic Signs dataset to predict the traffic sign present on the images. The reason behind choosing such a solution is that CNNs proves to be effective at determining and picking complex patterns within images by using different filters where each one finds specific patterns within different pixel groupings. Another reason for choosing this approach is that CNNs are known to give high accuracy and would be faster and more efficient than normal image processing techniques.

Benchmark Model

The benchmark model that is going to be used for this project CNN model trained by Pierre Sermanet and Yann LeCun. This benchmark model was able to reach an accuracy of 98.97%. I will be trying to train my model to reach a high accuracy score as that of the benchmark model.

More information about the benchmark model can be found here:

http://yann.lecun.com/exdb/publis/pdf/sermanet-ijcnn-11.pdf

Evaluation Metrics

The evaluation metric that is going to be used for this project are the model's accuracy on the testing data of the BTSD dataset. Accuracy is defined as the model being able to correctly predict the label of the traffic sign present in an image. This same evaluation metric was used to evaluate the benchmark model as stated before.

Project Design

(approx. 1 page)

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow of the capstone project.

The first step of the project will be collecting the data. As discussed before, I will be using the Belgium Traffic Signs dataset (BTSD). These images will then be resized to be of the same dimensions so that we can train our model on. After resizing the images, they will be fed into a CNN that will learn to predict the corresponding label of these images.

The network will consist of several convolutional layers which will then be followed by normalization or pooling layers to decrease the dimensionality of our model. Several dropouts will also be placed so that some randomly chosen neurons present in our network will be dropped in order to guarantee that our model will not overfit. The network is then followed by fully connected layers. The output layer will consist of 62 neurons because our dataset consists of 62 classes.

The activation functions used will be the 'relu' activation function, As for the final output layer, a 'softmax' activation function will be used so that our model will output the probabilities that the traffic sign present in an image belongs to each of the 62 classes.

The evaluation of our model will then be determined by calculating the model's accuracy on predicting the labels of the traffics signs present in the testing images. A subset of the training set will be split to form the validation set. This validation set will be used to evaluate the changes in the model's hyperparameters.

During the training process we will compute the model's accuracy on predicting the validation set and consequently save the model's weights if we notice an increase in this accuracy. After training our model, the best weights will then be saved into an 'Hdf5' file so that they are loaded directly without the need train our model again each time we need to predict the corresponding label of a traffic sign.