Machine Learning Engineer Nanodegree

Capstone Proposal

Luciano Silveira
December, 2019

Proposal

Domain Background

Machine Learning is starting to touch all aspects of our life, in particular it has a lot of potential for automation in robotic applications.

Being part of the donkeycar2 robocar community, an opensource do it yourself self-driving platform for small scale cars; there are many possibilities for exploration in that space.

In this context, the exploration and experimentation has been done mainly with `supervised learning` and the community is starting to play around with `reinforcement learning` (RL) algorithms.

My objective is to implement a classification model for the robocar to detect where it is in a specific race track so the model can be used for other high level models. In particular, it would be useful for `reinforcement learning` and to explore how to autonomously navigate a track withing the specified lane.

The robocar (an agent within the RL enviornment) makes the decision of what action to take depending on the feedback provided by the classification model, trying to always be on the `Center` of the track.

Problem Statement

The objective is to train an inference system to explore possible usage on small scale robots within a known environment. The model will need to predict the location of the car within a known track, that is to say identify 3 possible classes:

- Center of the track
- Left side of the track
- Right side of the track

Datasets and Inputs

The dataset can be gathered from the simulator or using sample driving data from the community.

Once the data adquisition is defined; the training and test data must be manually classified and organized in the following classes: Center, Left, Right.

Some Image Processing may be needed, such as use grayscale, normalization, contrast analysis etc...

Solution Statement

The objective is to correctly classify the images received from the donkeycar with accuracy and high confidence, so it can be used as reliable information from other algorithms.

Part of the problem is to evaluate different algorithms and models to do the task. The robocar community mainly uses the Keras framework to train the autopilot but our exploration is not restricted to it and could explore and evaluate other frameworks and algorithms.

In particular, the keras.py details different supervised learning models with different input and outputs.

Benchmark Model

Initially, get started using some off-the-shelf classification model such as the LeNet model which is implemented in different frameworks.

Then (and depending on the results), evolve and try more specific models, such as the End to End Learning for Self-Driving Cars by NVidia, which details how they did to drive a car directly from Cameras.

This would be a starting point, to use a classifier for another task and we can retrain it for our classification purposes.

Evaluation Metrics

The metrics to be used are those related when using a CNN Convolution Neural Network to classify images; in general a variation of `Softmax` is used; depending on the framework used, the activation functions will be analyzed and tested.

Evaluate the different available loss functions to compare Training vs Test data.

Project Design

The project is expected to be divided in the following steps:

- Data gathering and manual classification
- Framework exploration
- Training and validation
- Summarization of the results

Data Management

Use the provided simulator to collect data or directly download existing images from the community.

Manually classify the images in 3 classes: `Center`, `Left`, `Right` and divide the resultset into two groups: training (70%) and testing (30%).

Framework Exploration

Initially, evaluate the existing code from the Keras framework and train a classifier.

Iterate until the evaluation metrics are satisfied otherwise evaluate other frameworks and start again.

Conclusion

Summarization of the results.

Links

- Donkeycar Github repository
- Train autopilot docs
- Reinforcement Learning