Proceedings of the VIII International Conference "Distributed Computing and Grid-technologies in Science and Education" (GRID 2018), Dubna, Moscow region, Russia, September 10 - 14, 2018

CONVOLUTIONAL NEURAL NETWORKS FOR SELF-DRIVING CARS ON GPU

B.V. Tiulkin [a], N.V. Kulabukhova [b]

Faculty of Applied Mathematics and Control Processes, Saint Petersburg State University, 13B Universitetskaya Emb., St Petersburg 199034, Russia

E-mail: [a] st050135@student.spbu.ru, [b] n.kulabukhova@spbu.ru

Self-driving vehicles are considered to be safer than those driven by humans. Since they are always aware of what is happening around them and focuses on all the details. But to be really safe and respond to all the events happening around the drones need to process information and make decisions in the shortest possible time. The challenge is to teach how to drive a vehicle without human with the help of deep learning power using visual data from the cameras installed on the machine. The problem is to process the amount of data in the real time. Convolutional neural networks (CNNs) are used for training data. And the idea of how to use CNNs on graphical processing units is described.

Keywords: Self-Driving Cars, High-Performance Computing, Convolutional Neural Networks, GPU

© 2018 Boris V. Tiulkin, Nataliia V. Kulabukhova

CEUR-WS.org/Vol-2267/611-614-paper-117.pdf

1. Introduction

Currently, there are a large number of projects for the development of unmanned vehicles. The most famous such as Mobileye by Intel [1], WayMo by Google [2], or even self-driving taxi from Yandex Research [3]. The basic principle of such systems is the processing of data collected from cameras and lidars. In the case of Intel, any sensors are absent, since only visual data obtained from cameras on the machine is analyzed. However, processing of a large number of images in real time is required.

Main problems for practically every self-driving project:

1. speed of data processing, because it is necessary to obtain them in real time,
2. driving in the absence of road markings,
3. influence of weather conditions and lighting,
4. motion obscure visual landmarks,
5. actions in unforeseen situations.

Convolutional Neural Networks has been used for commercial purposes for more than twenty years. [4]. But two important events have served as a great impetus to the popularity of these neural networks in recent years. First, large, labeled datasets such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [5] are now widely available for training and validation. Another is that CNNs learning algorithms can now be implemented on massively parallel graphics processing units (GPUs), greatly speeding up learning and output.

2. Initial data

A person drives a car. It has cameras for data collection. The result is a set of frames, for each there is a value of the angle of rotation at that time.

Data is collected under different weather conditions, lighting and time of day. They are sorted on the basis of the above conditions.

To train a network to get out of a bad situation, we expand dataset by adding artificial perturbations.

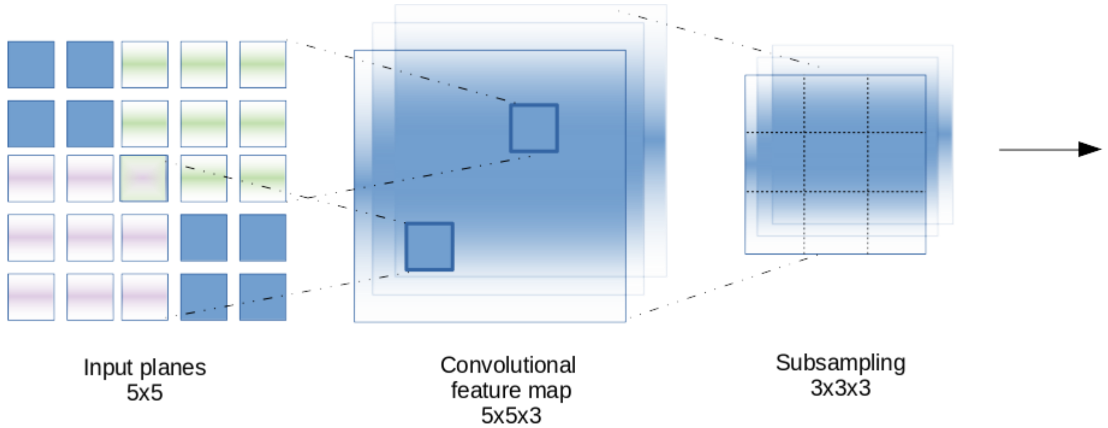

Figure 1. Diagram of a one layer convolutional network: convolution followed by downsampling

3. Selection of neural networks

We use convolutional neural networks because they, by reason of their architecture, are well suited for solving complex graphic image recognition and classification tasks [6].

More specifically, the choice was due to the following advantages:

- Algorithms using convolutional neural networks are invariant to various distortions, such as camera rotation, uneven the distribution of light in the image, horizontal or vertical shifts and other.

– CNN does not require the allocation of large amounts of memory for storing the extracted features in a work process.

– Fast learning speed, which is achieved by reducing the number of parameters used.

The performance of a CNN is several times greater than the performance of other neural networks used in recognition tasks.

The idea of convolutional neural networks is the alternation of convolutional layers and layers of pooling (subsample, subsampling) (see figure 1). The meaning of the convolution operation is that each fragment of the image is multiplied by the matrix (core) of the convolution element by element, and the result is summed and recorded in the similar position of the output image [7].

Convolutional neural networks are particularly effective for image recognition tasks, because the convolution operation captures the 2D nature of images.

4. Neural networks on GPU

To solve the problem requires processing a large amount of data in real time. In order to speed up the process, the calculations will be performed on the GPU.

Comparing with CPUs, GPUs are better suited for machine learning, because technical features help them perform at the same time a large amount of data, which is used to train the neural network presented in a matrix form. But there are some difficulties associated with a limited volume of the GPU memory. Therefore, it is better to use heterogeneous environment [8] for such kind of problems, where the graphical accelerators are used for calculations during training and testing data, and the analysis is made on CPU. The schematic view of the working process is shown on figure 2.

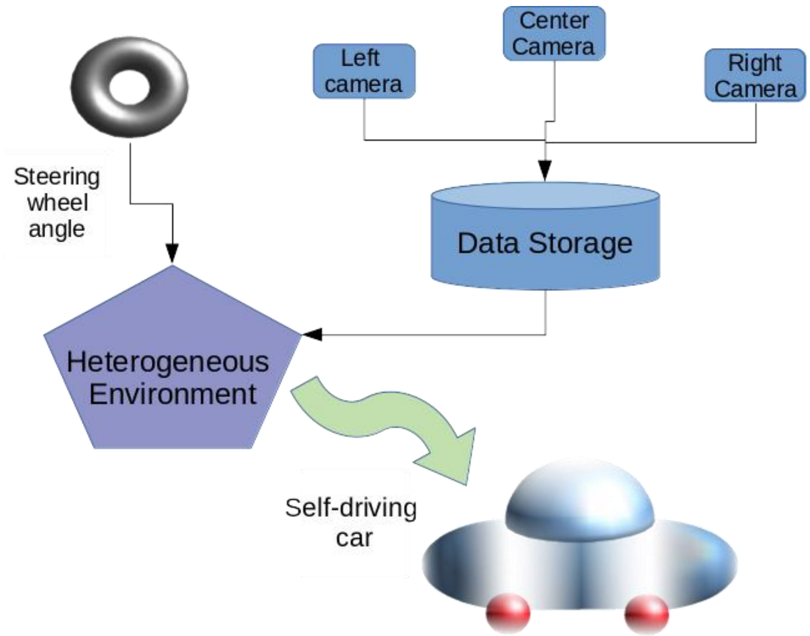

Figure 2. Schematic view of the system

5. Simulation

Lidars, cameras and radars scan everything around the car. The quality of information-pictures and parameters from sensors - allows the car to analyze road markings, road signs, vehicles, people, animals and all other elements of the visual environment. Thus, the ability to predict the further course of moving objects, the vehicle determines what to do next. But to do this, the data must be worked out. Since, for example, when we use a training set of data obtained during the control of the car by the driver, there is a problem that a person is naturally not able to constantly drive the car at the same distance from the roadside. There are a number of other things to consider before using the data.

Since the classification can take countless hours, it is usually used a larger image of the objects. Thus, the database already contains two-dimensional preprocessed contours of the required objects. Such as road signs, people, bicycles, other vehicles, trains and traffic lights.

In addition, generally processed images already contain notes for vehicles in cases where driving decisions are most difficult, such as at busy intersections, cluttered road systems or in the presence of multiple lanes.

For the described architecture of convolutional neural network was chosen Python 3.5 language. This is due to the fact that the language is well developed in the machine learning and has a large number of libraries. Was selected the library TensorFlow [9], as it allows computation as on the GPU.

The challenge is to select the weights while minimizing the root-mean-square error between the steering output over the network and the driver data. For training, the method of back propagation of the error is used.

6. Brief conclusions and development prospects

The paper deals with the topic of learning to drive unmanned vehicles, the method of its realization by means of convolutional networks is chosen. Hereinafter to test the effectiveness of this method, a practical implementation.

We test our neural network for speed and accuracy. To improve the work of the CNN, increase its stability and prevent overfitting, try the dropout method (subnet training method with throwing out random single neurons).

Next, we try various variable network parameters, such as the number of layers, the core dimension of the bundle for each of the layers, the number of cores for each of the layers, the step of the core shift when processing the layer, the need for sub-sampling layers, the degree of dimension reduction, the transfer function neurons, the presence and parameters of the output fully connected neural network at the output of the convolutional. These parameters greatly affect the operation of the neural network and they are selected only empirically.

We test on a different amount of data with artificial distortions.

Testing with different parameters and changing the neural network, we will achieve the best result.

References

[1] Amnon Shashua, If You Can Drive in Jerusalem You Can Drive (Almost), Anywhere, https://newsroom.intel.com/editorials/if-you-can-drive-jerusalem-you-can-drive-almost-anywhere/

[2] WayMo Technology, https://waymo.com/tech/

[3] Yandex.Taxi Solution, https://yandex.ru/promo/taxi/sdc

[4] L. D. Jackel, D. Sharman, Stenard C. E., Strom B. I., , and D Zuckert. Optical character recognition for self-service banking. AT&T Technical Journal, 74(1):16–24, 1995

[5] Large scale visual recognition challenge (ILSVRC). URL: http://www.image-net.org/challenges/LSVRC/

[6] Bojarski M. et al. End to end learning for self-driving cars //arXiv preprint arXiv:1604.07316. 2016

[7] Haykin S. Neural networks: a full course, 2nd edition. - Williams Publishing House, 2008 (in Russian)

[8] John Levesque, Aaron Vose, Programming for Hybrid Multi/Manycore MPP Systems, CRC Press Taylor & Francis Group, p.374, 2018

[9] TensorFlow API Documentation https://www.tensorflow.org/api_docs/