Speedup Critical Stage of Machine Learning with Batch Scheduling in GPU

Yuan Gao, Rui Wang, Ning An, Yanjiang Wei, Depei Qian

Speedup Critical Stage of Machine Learning with Batch Scheduling in GPU

Yuan Gao, Rui Wang, Ning An, Yanjiang Wei, and Depei Qian

BeiHang University, XueYuan Road No.37 HaiDian District Beijing, China,
rui.wang@jsi.buaa.edu.cn

Abstract. As a superior data analysis method, Machine Learning suffers the bottleneck from limited computing capability for many years. With the advent of numerous parallel computing hardwares, modern GPU is becoming a promising carrier for the tasks of Machine Learning. In this paper, we propose an efficient GPU execution framework to speedup the forward propagation process of convolution neural network. By extending the convolution unrolling method to fit this batch mode, we get a significant increase of throughput but very little overhead.

Keywords: convolution neural network, framework, GPU, batch process

1 Introduction

With the fast-development of computer hardware, GPU has become a type of important computation carriers with dramatic speedup and better energy efficiency. More and more computing-intensive applications with natural data parallelism have been migrated to GPU environment. However, although the NVIDIA CUDA can help programmers to develop faster applications, programming on GPU is much more difficult than that on CPUs. Programmers need to determine the timing of copying the data from the host to the GPU, the number of blocks or threads that should be divided reasonably to take full advantage of computing resources, and the size of data slice that should be allocated to each thread. It is quite difficult to develop a framework for a beginner to write efficient GPU programs in any purpose. However, research on high productivity method both in coding and in execution for special domains is promising.

2 CNN GPU Execution Framework

In this section, we take the image recognition application as an example to illustrate the operating mechanism of our framework.

2.1 Batch Process

Improving the GPU computing resource utilization is a good way to speed up the forward propagation since the computing hardware resource of GPU is quite

rich and forward propagation cannot take full advantage of GPU's ability if only handle one small size image at a time. What we can do, however, is to try to increase the throughput in a single task, which means the CNN processes several images in one forward propagation. The figure 1 presents the proposed framework which shows two of the most important modules: the input organization module and the convolution module. The input organization module automatically splices tiny input images into a larger one before CNN executes on GPU.

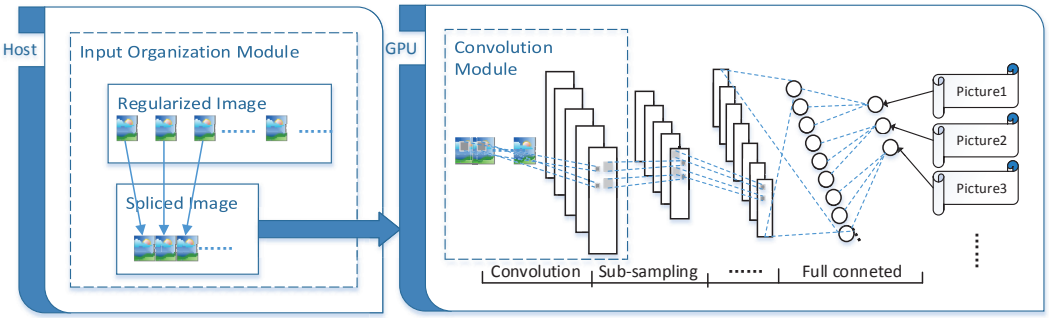

Fig. 1. Proposed GPU execution framework schematic.

The convolution layer requires the most amount of the calculation and is also the optimization emphasis of our framework while the traditional convolution layer speeding up method involves unrolling convolution

In our framework, we extend the simple unfolding of a convolution technique to a batch mode which enables the convolution layer to process multiple images (input features) at the same time.

When the spliced image (input features) arrives at the convolution layer which came from the host or last layer a split operation is necessary at first. As we know, each layer of the neural network has its fixed input size that is obviously the processing layer's input size is $\mathbf{3 \times 3}$ in the figure 2. According to this input size, the framework will split the spliced images (input features) into the standard size before unrolling convolution. The convolution operation works in a similar way except the unrolled input features from different images need to be added to the bottom of input feature matrix. As shown in the figure 2, the spliced image (input features) will be divided into nine $\mathbf{3 \times 3}$ input features and be unrolled to a $\mathbf{12 \times 12}$ input feature matrix and the kernel matrix is exactly same as the traditional unrolling convolution. After multiplying input features matrix by the kernel matrix, we get a $\mathbf{12 \times 2}$ output feature matrix that is still able to find the boundary of different images. Instead of convoluting the three images in a row, the batch mode unrolls three images to one matrix and does only one multiplication to get the output features.

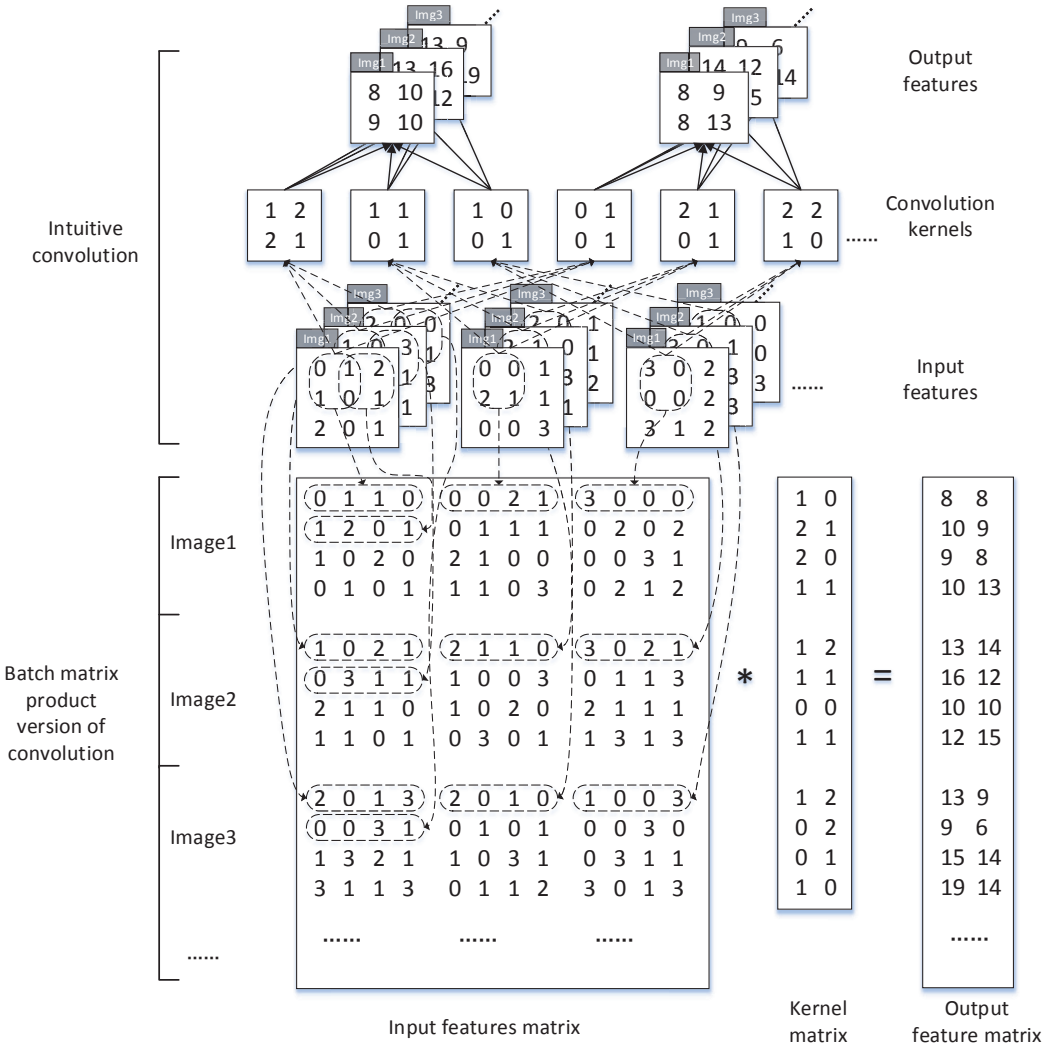

Fig. 2. Example batch mode unrolling convolution.

3 Evaluation

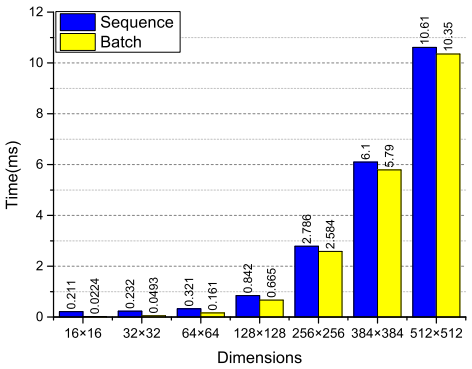

Fig. 3. Execution time for executing image convolution operations with varying image dimensions on the GPU and the splicing number is 10.

Figure 3 and compared the execution time between the two methods, and in the experiment, the filter size is 5 and the number of output features is fixed by 10. We, however, still take the image dimension and the splicing number as variables.

4 Conclusion

In this paper, we propose an efficient neural network framework on GPU platforms. Our framework addresses the gap in abstraction between domain experts and current GPU programming frameworks, and accelerates the process of CNN forward propagation. The framework can organize the input data automatically to make use of GPU resources as much as possible for better performance. We demonstrate the advantage of our framework in each CNN phase with five experiments. According to the power of GPU, framework can extremely increasing the throughput of CNN. The optimization method that we adopt in the framework doesn't modify the structure of CNN or require training extra neurons, which can combine with other optimization method to achieve better performance.

Acknowledgment

This research is supported by 863 Program of China under grant 2012AA010902, and by the NSFC under grant 61133004, 61073011, 61202425, and Huawei company under grant No.YB2012120105.