НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

ИНСТИТУТ ПРОБЛЕМ УПРАВЛЕНИЯ
им. В.А. ТРАПЕЗНИКОВА РАН

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ (ИТММ-2021)

МАТЕРИАЛЫ XX Международной конференции имени А. Ф. Терпугова 1–5 декабря 2021 г.

ТОМСК
Издательство Томского
государственного университета
2022

NEURAL NETWORKS ARCHITECTURE APPLIED TO FPGA

E. Solis Romeu, D. Shashev

Tomsk State University, Tomsk, Russian Federation

Field Programmable Field Arrays distinguish themselves from other integrated circuits, by their reconfigurability. This, along with their parallelism opens possibilities for implementing artificial neural networks with a high efficiency regarding cost and energy use. Research conducted on the implementation of Neural Network Architectures like VGG16 and AlexaNet in FPGAs can result in systems that are faster and more power efficient than GPU implementations.

Keywords: *FPGA, ASIC, Convolutional Neural Networks, AlexaNet, VGG16, Winograd, Matrix multiplication*

Introduction

FPGA stands for Field Programmable Field Arrays. It is a device made out of semiconductors that can be configured after being manufactured. This characteristic gives it the advantage of allowing flexibility in its design, which saves time and costs at the moment of introducing changes to it. FPGAs consist of programmable logic elements that are connected one to another, as well as memory blocks. They stand in contrast to ASICs (Application Specific Integrated Circuits) and ASSPs (Application Specific Standard Products). ASICs as their name implies, have a specific customized particular use. This means that they are designed with a particular functionality in mind. ASICs usually are sold to a specific user and ASSPs, while also having a specific use, are sold to many users. The three are integrated circuits, with the main difference being that FPGAs are reconfigurable while ASICs and ASSPs are not [1].

1. Neural Networks

Neural Networks are algorithms that mimic the functionality of human neural networks to create systems capable of analysing data. They can be used for Artificial Intelligence tasks like image recognition and video analysis. These networks are composed of at least three node layers: the input layer, the hidden layer and the output layer. Each node is a neuron that uses an activation function. An activation function is used to determine

Supported by FRBR, project number 19-29-06078.

the output of the node, given the inputs coming from previous layers and the weights assigned to them. The design of neural networks requires forward and backward processes. The forward process focuses on the amount of layers and the forward flow of the input and output information, while the backward process focuses on defining the loss function to calculate the gap between the prediction value and the actual values of the data samples during the training phase. The result of the backward process is used to alter the weights given to each node to make the final output more accurate. When the system runs a forward and backward process it is known as an epoch. Neural Networks require hundreds of epochs of training to reach high levels of accuracy [2].

2. Why Use a FPGAs for Designing Neural Networks?

The idea of creating a semi custom high performance system for neural networks that is able to outperform conventional processors is the main reason behind the efforts to develop these systems. The results so far have been that FPGAs are outperformed by ASICs. Nevertheless the structure of a FPGA presents an alternative to the software inside a general purpose processor, thus providing the possibility for delivering a superior performance in relation to cost and energy consumption on specific applications. Another apparent alternative would be the use of ASIC systems, nevertheless they have disadvantages over FGPAs.

Two important reasons exist as to why FPGAs are a better alternative than ASICs for developing systems capable of beating general purpose processors in performance. The first one is that when developing ASIC neurocomputers with a significant amount of flexibility, the final result is a system with a structure that closely resembles a FPGA, while never reaching the flexibility that a FPGA can attain. This is important because neural networks require to be easily reconfigured since they are meant to be used for different applications. The second fact is that the design and production of custom neurocomputers has a limited user base, thus the development of software for ASIC neurocomputers will lag behind FPGAs that will be more widely used [3].

3. Parallelism

Other reason for the drive towards the use of FPGA for creating neural networks systems are the perceived abilities that this technology has in the realm of parallelism. Parallelism is the feature that allows one to run two or more computational processes at the same time. FPGAs are more suitable for parallelism than ASICs systems. These capabilities are suitable for working with neural networks. To better understand this, we can take a look at different types of parallelism applicable to this topic. The first one is Training parallelism, this refers to the ability of running different training sessions for the neural network at the same time. Node parallelism refers to the idea of going through the nodes of each layer of the neural network in parallel, and this could be a great advantage for a FPGA system. Nevertheless it is necessary to take into account that neural networks can have up to a million nodes, and a system with that high amount will face limitations that will not allow it to reach those high levels of parallelism. Another promise of FPGA systems is the idea of computing the layers of the neural network in a parallel way [3].

4. Implementation with Winograd Algorithm

In the article Evaluating Fast Algorithms for Convolutional Neural Network, The authors evaluated the performance of two FPGA devices (Xilinx ZC760 and ZCU102) on which they used the Winograd algorithm for matrix multiplication as a way of making more efficient the implementation of two CNN architectures (AlexaNet and VGG16). The Winograd algorithm speeds the convolution process by simplifying the matrix multiplication. This plays an important role in convolutional algorithms since they require the multiplication of matrix filters in each layer of the neural network. During the experiment, other two FPGAs (VX485T and GSD8) were used as a control group without the Winograd algorithm being implemented in them. All models used the AlexaNet Architecture for CNN. [4] The results show that the Winograd implementation improved the average convolution performance from 61.6 Gop/s to 1006.4 Gop/s, while the CNN average performance rose from 72.4 Gop/s to 854.6 Gop/s. The implementation of the modified algorithms resulted in the use of less resources; this can be seen in the energy efficiency rising from 3.79 Gop/s/W to 36.2 Gop/s/w.

5. Conclusion

The advantages of FPGAs over ASICs and ASSPs is their reconfigurability, which opens the possibility for their flexible application by Artificial Intelligence specialists. Processors have the advantage of using their power to deliver systems with a higher analytical capacity as they are more accurate, but the advantage of the FPGA comes from delivering a more efficient use of resources (both computing power and energy). A recurring theme on the implementation and modification of neural network algorithms for FPGA, is that advantages in resource and power efficiency come at the price of less accuracy. How much one is willing to sacrifice for the other will affect the choice of using a FPGA or a GPU for the implementation of a CNN. In the end, the existence of tasks that demand fast systems with low power consumption creates incentives for the use of FPGA. Another of the promises of FPGA systems is parallelism, which would enable them to fulfil several tasks of an algorithm at the same time. These reasons justify the experimentation with CNN architectures and algorithms with the aim of creating competitive configurable systems based on FPGAs.

REFERENCES

1. *More A., Wilson R.* FPGAs for Dummies. City: John Wiley & Sons Inc, 2017. P. 3-6.
2. *Yufeng H.* A General Neural Network Hardware Architecture on FPGA // Dept. of Electronic, Electrical and Systems Engineering University of Birmingham, City: Birminham, 2017.
3. *Omondi A., Rajapakse J., Bajger M.* FPGA Neurocomputers // FPGA Implementations of Neural Networks, Publisher: Springer, 2006.
4. *Lu L., Liang Y., Xiao Q., Yan S.* Evaluating Fast Algorithms for Convolutional Neural Networks on FPGAs // IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), 2017.

Solis Romeu Edgar — PhD Student, Master, Faculty of Innovation Technology. E-mail: *solisromeu@mail.ru*

Shashev Dmitry — Ph.D in Engineering, Deputy Dean for R&D at the Faculty of Innovative Technologies , Faculty of Innovation Technology. E-mail: *dshashev@mail.tsu.ru*