SIG Proceedings Paper in LaTeX Format

ABSTRACT

Training convolutional neural network (CNN) usually requires large amount of computation resources, time and power. Researchers and cloud service providers in this region needs fast and efficient training system. GPU is currently the best candidate for DNN training. But FPGAs have already shown good performance and energy efficiency as CNN inference accelerators. In this work, we design an FPGA-based fast and energy efficient CNN training accelerator. We adopt two of the widely used model compression methods, quantization and pruning, to accelerate CNN training process.

Although quantization and pruning are proved to be efficient for CNN inference, we are still faced with challenges when adopting them in training.

But accelerating CNN training is still hard. On one hand, current work focuses on training an efficient network rather than efficiently training a network. The training phase is still not hardware friendly enough. On the other hand, the difference between inference phase and back propagation phase in training brings more challenges to hardware design. Existing designs are hard to support training efficiently and utilize sparse property in training.

In this paper, we propose an efficient CNN training method with both software optimization and hardware architecture design. A hardware friendly network training method is proposed with all-fixed-point-data computation and sparse network parameters. An FPGA based architecture is designed to accelerate this training process. Proposed hardware achieves 641GOP/s equivalent performance and 3x better energy efficiency compared with GPU.

KEYWORDS

FPGA, Convolutional Neural Network, Training

ACM Reference Format:

. 2018. SIG Proceedings Paper in LaTeX Format. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Convolutional neural networks (CNN) has made significant performance improvement in computer vision tasks [3, 7]. However, the accuracy improvement comes at significant cost of training computation.

Model pruning and quantization have proved to be effective to reduce the requirements of computation, bandwidth and memory footprint, and have almost no effect on the performance (accuracy metric) of neural networks [1, 2, 8]. A series of the hardware accelerator architecture designs are focused on the forward inference phase of neural networks, which takes full advantage of the benefits of sparseness and quantization and shows outstanding performance and energy efficiency. To deploy a sparse network, first the neural network model well-trained in dense needs to be pruned, and the loss of accuracy can be recovered by fine-tuning. Sparse neural networks can be deployed on different platforms and can achieve higher speed and energy efficiency than dense neural networks, as shown in Figure1.

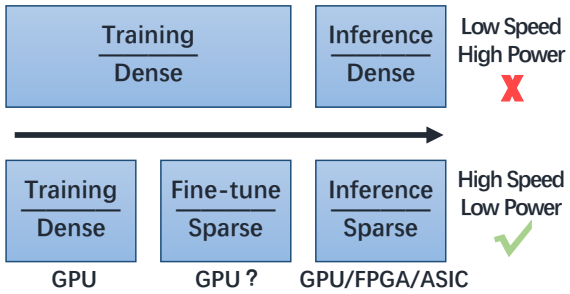

Figure 1: Dense training, Sparse fine-tune, Sparse inference

Although the GPU is suitable for training of dense neural networks, it can not be a proper user of the sparsity in the sparse neural network to further improve the performance (speed metric). Customized design hardware can make better use of the sparsity of the network to achieve the purpose of improving training efficiency. As a hardware programmable devices, FPGA has been widely deployed in the server cluster [4–6], but there is still no training architecture can take advantage of sparsity. On the other hand, the current fixed-point neural network training algorithm is still not hardware-friendly, for example, need floating point number to calculate the gradients[9].

In this paper, we propose an architecture design for sparse neural network training accelerators on the FPGA platform, which reached a high computing capability and energy efficiency. At the same time, we provide a sparse convolution neural network transformation technology, making the sparse neural network more suitable for training and inference on customized hardware platform while maintaining the accuracy at almost the same level. The main contributions of this paper are summarized as follows:

- A hardware friendly training process is proposed with all-fixed-point operations.
- Dedicated processing element (PE) on FPGA is designed to utilize the sparsity in both feed forward and back propagation phases of training.
- We analyze the limitation on unroll parameters brought by sparsity and loop dimension variety between feed forward and back propagation phases. A corresponding flexible PE

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

array structure is proposed to improve hardware utilization ratio.
- Data arrangement and schedule strategies are proposed to improve bandwidth utilization .

Experimental results show that Proposed hardware achieves 641GOP/s equivalent performance and 3x better energy efficiency compared with GPU.

The rest of this paper is organized as follows. Section **??** introduces the background of training of a CNN. Section **??** reviews previous work on software and hardware level CNN optimization. Section **??** and section **??** introduces the proposed training process and hardware platform respectively. Experimental results are shown in section **??**. Section **??** concludes this paper.

REFERENCES

[1] Song Han, Huizi Mao, and William J Dally. 2016. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *International Conference on Learning Representations* (2016).

[2] Song Han, Jeff Pool, John Tran, and William Dally. [n. d.]. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*. 1135–1143.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Computer Vision and Pattern Recognition*. 770–778.

[4] Andrew Putnam. [n. d.]. Large-scale reconfigurable computing in a Microsoft datacenter. In *Hot Chips 26 Symposium (HCS), 2014 IEEE*. IEEE, 1–38.

[5] Andrew Putnam, Adrian M Caulfield, Eric S Chung, Derek Chiou, Kypros Constantinides, John Demme, Hadi Esmaeilzadeh, Jeremy Fowers, Gopi Prashanth Gopal, and Jan Gray. [n. d.]. A reconfigurable fabric for accelerating large-scale datacenter services. In *Computer Architecture (ISCA), 2014 ACM/IEEE 41st International Symposium on*. IEEE, 13–24.

[6] Yi Shan, Bo Wang, Jing Yan, Yu Wang, Ningyi Xu, and Huazhong Yang. [n. d.]. FPMR: MapReduce framework on FPGA. In *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*. ACM, 93–102.

[7] E Shelhamer, J. Long, and T Darrell. 2017. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 4 (2017), 640.

[8] Yi Sun, Xiaogang Wang, and Xiaoou Tang. [n. d.]. Sparsifying neural network connections for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4856–4864.

[9] Shuchang Zhou, Zekun Ni, Xinyu Zhou, He Wen, Yuxin Wu, and Yuheng Zou. 2016. DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients. *CoRR* abs/1606.06160 (2016). http://arxiv.org/abs/1606.06160