# Compressed CNN Training with FPGA-based Accelerator


## 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 a compressed training process together with an FPGA-based accelerator. We adopt two of the widely used model compression methods, quantization and pruning, to accelerate CNN training process.

The difference between inference and training brought challenges to apply the two methods in training. First, training requires higher data precision. We use the gradient accumulation buffer to achieve low operation complexity while keeping gradient descent precision. Second, sparse network results in different types of functions in forward and back-propagation phases. We design a novel architecture to utilize both inference and back-propagation sparsity. Experimental results show that the proposed training process achieves similar accuracy compared with traditional training process with floating point data. The proposed accelerator achieves 641GOP/s equivalent performance and 3x better energy efficiency compared with GPU.



## KEYWORDS

FPGA, Convolutional Neural Network, Training

**ACM Reference Format:**

. 2018. Compressed CNN Training with FPGA-based Accelerator. 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, 5]. However, the accuracy improvement comes at significant cost of both inference and training computation. A wide range of work have bee proposed [1, 4] to achieve fast and energy efficient inference, showing outstanding performance over GPU. But the training phase of neural networks was not fully focused on.

Training neural networks requires large amount of computation resources and can take days or weeks. Fast and energy efficient training is important for researchers and cloud service providers. GPU is currently the most suitable platform for CNN training. Recently, a number of cloud service providers including Microsoft, Amazon, Alibaba, and Huawei have deployed large FPGA clusters in their data centers. This provides the environment for FPGA-based neural network training acceleration.

The training of neural network requires similar operations as inference. Previous work on inference accelerator design gives hint on training acceleration. 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 network inference [2]. A series of the accelerator designs [1, 4] follow these ideas with specific hardware. But applying pruning and quantization to training are faced with challenges.

The first challenge lies in the optimization of training. Existing work [2] applies pruning and quantization to training but only in the fine-tune phase after the model converges. The benefit of accelerating the fine-tune phase is quite limited. Using quantization and pruning in early stage of training is risky. Different from inference, training process fine steps of gradient descent to improve the model accuracy. Using low precision data can lead to convergence failure or accuracy decline. Pruning on the network before convergence can also limit the parameter space of the model and hurt the final training accuracy.

The second challenge lies in hardware design for processing sparse network training. Existing accelerator designs implements sparse matrix-vector multiplication kernels to skip the zero multipliers in sparse models. We refer to this kind of sparsity as operator-sparse pattern. In the back-propagation phase of training, utilizing sparsity means to avoid computing gradients for those already pruned parameters, which introduce the result-sparse pattern. Error propagation also requires the transposition of a sparse parameter matrix, bringing difficulty in accessing data efficiently from external memory. No existing designs support result-sparse pattern and sparse matrix transposition.

In this work, we address the above two challenges with the following contributions:

- We propose a hardware friendly training process with advanced quantization and pruning. Specially, we explore the effect of when to apply quantization and pruning with real tasks.
- We design dedicated processing elements (PEs) on FPGA to support both operator-sparse and result-sparse patterns. The sparse matrix transposition function is supported by specific scheduling method with a novel data organization in external memory.
- We evaluate a prototype system on FPGA to show the effectiveness of the design. Experimental results show that the proposed accelerator 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.




*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

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, Junlong Kang, Huizi Mao, Yiming Hu, Xin Li, Yubin Li, Dongliang Xie, Hong Luo, Song Yao, Yu Wang, et al. 2017. ESE: Efficient Speech Recognition Engine with Sparse LSTM on FPGA.. In *FPGA*. 75–84.

[2] Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* (2015).

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[4] Jiantao Qiu, Jie Wang, Song Yao, Kaiyuan Guo, Boxun Li, Erjin Zhou, Jincheng Yu, Tianqi Tang, Ningyi Xu, Sen Song, et al. 2016. Going deeper with embedded fpga platform for convolutional neural network. In *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 26–35.

[5] 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.