International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER SSRN

IJIEMR Transactions, online available on 26th Jul 2019. Link :http://www.ijiemr.org/downloads.php?vol=Volume-08&issue=ISSUE-07

Title **DESIGN AND IMPLEMENTATION OF A CNN ACCELERATOR**

Volume 08, Issue 07, Pages: 366–370.

Paper Authors

G.ANJANI, N. RAMES H BABU

St.Mary's Women's Engineering College, Budampadu; Guntur (Dt); A.P, India.

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

DESIGN AND IMPLEMENTATION OF A CNN ACCELERATOR

[1]G.ANJANI, [2]N. RAMES H BABU

[1]M-tech Student Scholar,Department of VLSI Engineering, ST.Mary's women's Engineering College,Budampadu;Guntur (Dt); A. p, India..

[2]Assistant Professor,Department of Electrical &Communication Engineering, ST.Mary's women's Engineering College, Budampadu;Guntur (Dt); A. P, India

[1]anjanigera2@gmail.com

Abstract: In recent years, Convolutional Neural Networks (CNNs) have revolutionized computer vision tasks. However, inference in current CNN designs is extremely computationally intensive. This has led to an explosion of new accelerator architectures designed to reduce power consumption and latency [20]. In this paper, we design and implement systolic array based architecture we call Convolutional AU to efficiently accelerate dense matrix multiplication operations in CNNs. We also train an 8-bit quantized version of Squeeze net[14] and evaluate our accelerator's power consumption and throughput. Finally, we compare our results to the reported results for the K80 GPU and Google's TPU. We find that Convolutional AU gives a 200x improvement in TOPs/W when compared to a NVIDIA K80 GPU and a 1.9x improvement when compared to the TPU.

1. Introduction

Since the remarkable success of AlexNet[17] on the 2012 ImageNet competition[24], CNNs have become the architecture of choice for many computer vision tasks. Inference on a trained CNN can be highly computationally expensive. A typical model might require billions of multiplyaccumulate operations (MACs), load millions of weights, and draw dozens of watts during inference (see table 1). This computational cost can provide significant barriers to deployment; in [16], Google projected in 2013 that three minutes of voice search per user (implemented using deep neural networks) would require Google to double its datacentre size at the time.In order to decrease execution time and power consumption, researchers and tech companies have investigated and built special purpose hardware for CNN inference. We present and analyse our own CNN accelerator Convolutional AU. Our architecture's main feature is a 256×256 systolic array of multiply-accumulate cells, allowing fast dense matrix-matrix and matrix-vector operations. We benchmark our system's latency, power, and performance using Squeezenet[14] trained on ImageNet[24] with Keras [4], quantized to 8 bits and executed using Tensor flow[1]. Finally, we compare our accelerator's throughput and power consumption to a GPU and Google's TPU.

2. Related Work

In recent years neural network accelerator design has been a topic of enormous interest to the computer architecture

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

community. CNNs were traditionally executed on CPUs, GPUs, and even FPGAs [22], which can easily adapt to new network architectures. However, this flexibility comes at the price of efficiency. As a result, application specific accelerators have been developed to maximize efficiency. Current approaches can mostly be classified into a combination of four categories, described below Dense Linear Algebra Accelerator The most common approach for CNN accelerators is to accelerate dense matrix-matrix and matrix-vector operations. Examples include DianNao[3] and Google's TPU[16]. Since dense linear algebra operations dominate the computational cost of CNN inference, these approaches have dramatically outperformed traditional CPUs and GPUs. However, one drawback is that many activations during inference are sparse, which means that some computation is wasted.Quantization Another approach is to quantize all operations to a small number of bits[8, 28, and 2]. Quantization can be used on CPUs and GPUs, but is most powerful in combination with hardware acceleration. Since small integer operations can be significantly more efficient to implement in hardware than floating point, this can dramatically increase the overall efficiency of the accelerator. However, this increased efficiency comes at the cost of decreased accuracy, although recent research has shown that by retraining the network this decrease can be made acceptably small[27, 6].Sparse Linear Algebra Accelerators Some more recent approaches take advantage of sparsely. The EIE[10] and ESE[9] authors design an architecture that is able to operate directly on a sparse form, which allows them to avoid computation where the graph is zero. In addition, they encode the weight matrix in an efficient compressed format, which allows them to avoid loading zero weights from memory. In order to maintain accuracy, they perform pruning during training by removing weights below a threshold and then retraining, which allows the network to recover and only nominally affects accuracy[11, 12].

Analogy Accelerators Finally, another approach that has gathered significant interest is to use mixed signal or analog computation. In [23], the author's show that mixed signal MAC operations can be performed efficiently using a switched capacitor design. In [19], the authors use a similar design to perform full matrix-matrix operations. Other approaches have included using resistive RAM elements [25] and flash transistors [7].

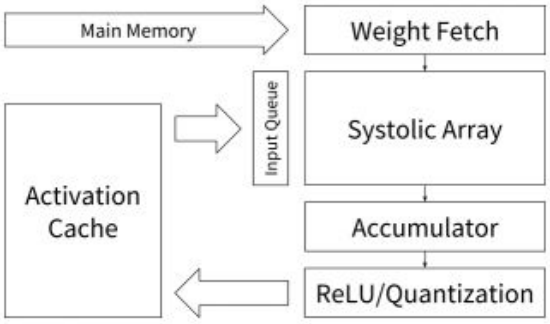

Figure 1: Overview of our entire design

One drawback is that analog and mixed signal accelerators require potentially expensive conversions between digital and analog representations. Additionally, analog designs tends to be larger, more error prone, more difficult to implement, and more costly to manufacture than the equivalent digital designs.

3. Accelerator Design

At it's core, our accelerator is a dense matrix multiplication unit (MMU) that can perform 256 × 256 8-bit integer multiply and 32-bit integer accumulate per cycle. Dense matrix multiplication acceleration

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

has been researched extensively as it has been used in GPUs and DSP algorithms, with most common implementation methods being systolic arrays, FFTs, or the Win grad algorithm. Convolutional AU uses a systolic array loosely based on Google's TPU[16]. A systolic array is a homogeneous grid of processing elements (PEs), each with a small amount of with each element connected only to it'sneighbours. During execution, each PE can optionally read from it'sneighbours, compute a simple function, and store the result in it's local memory. We decided to use simple pipelining concepts to increase the throughput of the design. We have three main pipe stages: input/weight loading, matrix multiplication and activation, and load results to output queue. Since the three operations are independent, we can schedule our operations to perform a matrix multiplication every operation cycle.

We chose to use a systolic array since it's easy to design and allows for an extremely efficient implementation. It also gives us a large amount of flexibility since it can accelerate any network architecture that uses dense matrix multiplication. For CNNs in particular, convolutions, batch normalization, and fully connected layers can all be efficiently implemented using dense matrix multiplications (see section 4). Together, these represent the vast majority of the computation required during inference.

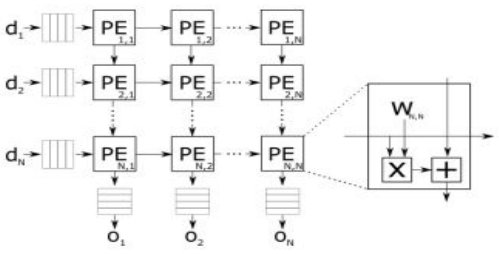

Figure 2: Our systolic array design

4. Results

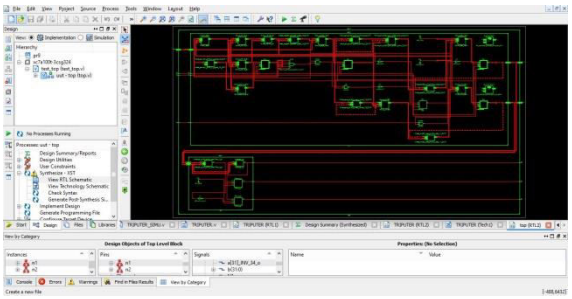

Figure 3: RTL schematic of the proposed system

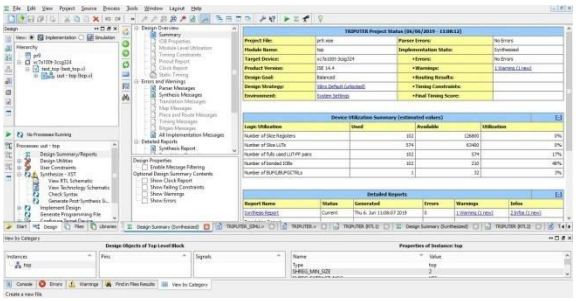

Figure 4: summary report of the proposed system

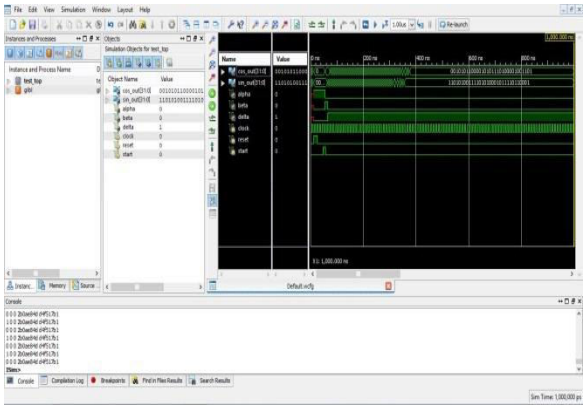

Figure 5: simulation result of the proposed system

5. Conclusion

In this paper we introduce the design of a CNN inference accelerator. We then train an 8-bit quantized version of Squeezenet, and evaluate throughput, latency, and power consumption during inference. When power consumption is adjusted to memory size, we find that the TPU have similar TOPs/W at a slightly higher clock rate. Thus, CNN is able to accelerate typical CNN workloads with comparable efficiency and performance to existing architectures.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane,´ R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. War- ´den, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] R. Andri, L. Cavigelli, D. Rossi, and L. Benini.Yodann: An architecture for ultra-low power binary-weight cnn acceleration. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2017.

[3] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. In ACM Sigplan Notices, volume 49, pages 269–284. ACM, 2014.

[4] F. Chollet et al. Keras.https://github.com/ fchollet/keras, 2015.

[5] A. Claros. Asynchronous fifo.

[6] M. Courbariaux, Y. Bengio, and J.-P. David. Training deep neural networks with low precision multiplications.arXiv preprint arXiv:1412.7024, 2014.

[7] L. F. et al. Floating-gate transistor array for performing weighted sum computation, August 2014. US Patent Application US14459577.

[8] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan.Deep learning with limited numerical precision.In ICML, pages 1737–1746, 2015.

[9] S. Han, J. Kang, H. Mao, Y. Hu, X. Li, Y. Li, D. Xie, H. Luo, S. Yao, Y. Wang, H. Yang, and W. J. Dally. ESE: Efficient Speech Recognition Engine with Sparse LSTM on FPGA. ArXiv e-prints, Dec. 2016.

[10] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally. EIE: efficient inference engine on compressed deep neural network. CoRR, abs/1602.01528, 2016.

[11] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149, 2015.

[12] S. Han, J. Pool, S. Narang, H. Mao, E. Gong, S. Tang, E. Elsen, P. Vajda, M. Paluri, J. Tran, et al. Dsd: Densesparse-dense training for deep neural networks. 2016.

[13] K. Hornik. Approximation capabilities of multilayer feedforwardnetworks.Neural Networks Volume 4, Issue 2, page 251257, 1991.

[14] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360, 2016

Author's Profile

GERA ANJANI. Received B.Tech from ST. Mary's Women's engineering collage. Budampadu, Guntur in ,Electronics and communication Engineering in the year 2017 and now pursuing M.Tech in the stream of Very large scale integration at ST. Mary's Women's

Engineering College Budampadu, Guntur,AndhraPradesh. His areas of interests are Vlsi systems, analog communication and Digital communication.

N.Ramesh babu is currently professor of Electronics and Communication Engineering in ST. Mary's Women's engineering college ,Budampadu, Guntur Dust. He has more than 6 years of experience in teaching and research. He is interesting in embedded systems and vlsi design.