Hardware Acceleration of CNN with One-Hot Quantization of Weights and Activations

Gang Li[1,2], Peisong Wang[1], Zejian Liu[1,2], Cong Leng[1], Jian Cheng[*1,2,3]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Center for Excellence in Brain Science and Intelligence Technology, CAS
Email: {gang.li, peisong.wang, jcheng}@nlpr.ia.ac.cn

***Abstract*—In this paper, we propose a novel one-hot representation for weights and activations in CNN model and demonstrate its benefits on hardware accelerator design. Specifically, rather than merely reducing the bitwidth, we quantize both weights and activations into n-bit integers that containing only one non-zero bit per value. In this way, the massive multiply and accumulates (MACs) are equivalent to additions of powers of two that can be efficiently calculated with histogram based computaitons. Experiments on the ImageNet classification task show that comparable accuracy can be obtained on our proposed One-Hot Networks (OHN) compared to conventional fixed-point networks. As case studies, we evaluate the efficacy of the one-hot data representation on two state-of-the-art CNN accelerators on FPGA, our preliminary results show that 50% and 68.5% resource saving can be achieved on *DaDianNao* and *Laconic* respectively. Besides, the one-hot optimized *Laconic* can further achieve an average speedup of $4.94\times$ on AlexNet.**

I. Introduction

Nowadays, Convolutional Neural Networks (CNN) have shown impressive performance on a variety of artificial intelligence applications, including image recognition, natural language processing, and robots. However, the vast computational complexity of deep networks poses a significant challenge to real-world deployment, especially on edge devices such as drones, smartphones, etc. Therefore, dedicated hardware accelerators are gaining popularity in both industry and academia.

Recent advances in deep learning have demonstrated that a variety of methods can be utilized to effectively reduce network complexity, such as pruning, tensor decomposition, fixed-point quantization, etc. Among these, fixed-point quantization that converts high-precision floating-point weights/activations to low-precision integers has become the *de facto* technique in hardware accelerator design due to its simplicity, resource and power efficiency. Some early accelerators use 16-bit or 8-bit quantization to maintain accuracy [1, 2], but at the cost of relatively sizeable on-chip resource usage and memory footprint. YodaNN [3] integrates binary weight (+1/-1) into CNN accelerator, FINN [4] further quantizes activations to binary values and efficiently implements Binarized Neural Networks (BNN) with *xnor* and *popcount* on resource-constraint FPGA. However, due to the limited ability of feature representation, BNN is currently not applicable to complex tasks such as ImageNet classification.

*Corresponding author.

Rather than merely reducing the bitwidth, recent advances in algorithm-hardware co-design have shown a promising way to leverage quantization to balance accuracy and hardware efficiency. Many accelerators adopt mixed-precision quantization due to the fact that the sensitivity of network accuracy with respect to layer precision is uncertain [5, 6]. Most recently, hardware-oriented logarithmic quantization that representing weights as powers of two is gaining focus since bit-wise shift operation is much cheaper than multiplication in terms of area and power consumption [7, 8].

In this paper, we propose One-Hot Networks to push the logarithmic representation to the extreme. Unlike previous work that only focus on weights, we quantize both weights and activations into powers of two. *The insight is to use the least effectual bits*[1] *to cover the broadest range of values, pursuing the maximum gains on hardware accelerator without severe loss of accuracy.* Extensive experiments reveal that our proposed OHN works surprisingly well on both sides of classification accuracy and hardware efficiency.

The rest of the paper is organized as follows. In Section 2, we briefly introduce our motivation for this work. In Section 3, we describe one-hot based quantization in detail. Then, in Section 4, we describe our evaluation of one-hot data representation on two state-of-the-art hardware accelerators, and Section 5 concludes this paper.

II. Motivation

The basic idea of hardware-oriented quantization is to minimize the consumption of on-chip computing and storage resources while maintaining accuracy. In our view, what determines resource consumption is the implicit effective bitwidth of weights/activations, not the numerical one, especially for high-precision values.

For example, when a 16-bit weight is a power of two, it can be equivalently represented as the bitwidth of its exponent plus a 1-bit sign bit [6]. That is, the 16-bit weight can be compressed to 5 bits. Moreover, when weights and activations are both powers of two, the calculation of their products can be converted to additions of the corresponding exponents followed by decoding, which will further simplify

[1]In [6], the effectual bits are equivalent to non-zero bits. We follow this definition throughout this paper.

978-3-9819263-4-7/DATE20/©2020 EDAA

the computing logic with reduced storage usage. In a word, OHN has the potential to find a balance between accuracy and hardware efficiency.

III. One-Hot Quantization

A convolutional layer maps the input tensor $\mathcal{X} \in \mathbb{R}^{c\times H\times W}$ to the output tensor $\mathcal{Y} \in \mathbb{R}^{n\times H\times W}$ through a weight tensor $\mathcal{W} \in \mathbb{R}^{n\times c\times h\times w}$, where H and W represent the height and width of the input/output feature maps, n, c, h, and w refer to the output channels, input channels, height, and width of the filters, respectively. Convolution can be transformed into matrix multiplication as follows:

$$\mathbf{Y} = \psi(\mathbf{W}^T\mathbf{X}), \tag{1}$$

where $\mathbf{W}^T \in \mathbb{R}^{n\times[c\cdot h\cdot w]}$, $\mathbf{X} \in \mathbb{R}^{[c\cdot h\cdot w]\times[H\cdot W]}$, and ψ represents the non-linear activation function (optionally with Batch Normalization (BN) layer).

The goal of network quantization is to turn the floating-point values of $\mathbf{W}^T$ and $\mathbf{X}$ into fixed-point representations. Unlike existing power-of-two quantization that only focuses on weights, for N-bit OHN, all values including weights and activations are quantized to N-bit, with only one bit equalling to 1. As an exception, we introduce zero into the network by allowing all bits to be 0.

A. Quantization of Activations

For the quantization of activations $\mathbf{X}$, we mainly discuss the unsigned quantization, as the ReLU non-linear activation function is commonly used in most of convolutional networks. For other activations like PReLU, we could use N+1-bit quantization, where the additional 1 indicates the sign bit. Thus for N-bit one-hot quantization of $\mathbf{X}$, values are constrained to $\{0, 1, 2, 4, \cdots, 2^{N-1}\} * \beta$, where β is introduced as a scaling factor to rescale the quantized values $\hat{\mathbf{X}}$ into the appropriate scale by the following optimization problem:

$$\min_{\beta} \parallel \mathbf{X} - \beta\hat{\mathbf{X}} \parallel_F^2 . \tag{2}$$

To solve this optimization, we sample the input features $\mathbf{X}$ by feeding a batch of input images into the network. Then an iterative optimization procedure can be conducted, i.e., optimizing β given $\hat{\mathbf{X}}$ and optimizing $\hat{\mathbf{X}}$ given β.

One-hot quantization has the problem of reduced resolution for large values. Thus minimizing the quantization error of equation 2 may not result in the minimal accuracy loss. To solve this problem, we utilize a linear quantization induced scaling factor selection method. Specifically, we use linear quantization (with the same maximum value as one-hot quantization, i.e., 2^{N-1}) when optimizing the scaling factor β, as follows:

$$\min_{\beta} \parallel \mathbf{X} - \beta\hat{\mathbf{X}}_l \parallel_F^2, \tag{3}$$

where $\hat{\mathbf{X}}_l$ represents the linearly quantized tensor whose values are from $\{0, 1, 2, 3, \cdots, 2^{N-1}\}$. In other words, we utilize the scaling factor of a linear quantizer for our one-hot quantizer.

TABLE I
THE EFFECT OF SCALING FACTOR SELECTION ON ACCURACY. RESULTS ARE THE TOP-1 ACCURACY OF ALEXNET WITHOUT FINE-TUNING. 'UOHT' INDICATES THE UNSIGNED ONE-HOT QUANTIZER. 'SELF' MEANS USING SCALING FACTORS OF EQUATION 2, 'UINT-#' MEANS USING SCALING FACTORS OF EQUATION 3

Quantizer	self	uint-3	uint-4	uint-5	uint-6
uoht-4	53.57	**54.27**	46.48	24.30	6.26
uoht-5	56.15	55.58	**56.36**	52.88	34.67
uoht-6	56.37	55.48	56.79	**56.84**	55.26
uoht-7	56.31	55.69	56.33	56.85	**56.93**

TABLE II
THE ACCURACY OF ONE-HOT ACTIVATION QUANTIZATION OF ALEXNET ON IMAGENET.

Model		Baselines	4-bit	5-bit	6-bit	7-bit
AlexNet	Top-1	60.5	58.5	58.7	58.7	58.6
	Top-5	80.9	79.5	79.6	79.7	79.8

B. Quantization of Weights

For the quantization of weights $\mathbf{W}^T$, there are two main differences with activations. (1) Signed quantization is needed. Thus we adopt N+1-bit quantization by constraining all quantized weights to $\{0, \pm1, \pm2, \pm4, \cdots, \pm2^{N-1}\}$. (2) We use separate scaling factors for different 2-D kernels, that is, each row w_i of $\mathbf{W}^T$ has a separate scaling factor λ_i. Thus the optimization problem of weight quantization is:

$$\min_{\Lambda,\mathbf{W}^T} \parallel \mathbf{W} - \Lambda\hat{\mathbf{W}} \parallel_F^2, \tag{4}$$

where Λ is a diagonal matrix, of which the i-th diagonal element λ_i denotes the scaling factor for the i-th kernel.

By the optimization of equation 2 and equation 4, we can get the quantized tensor as well as the scaling factors. Note that the introduced scaling factors for a convolutional layer can be merged into the following batch normalization layer, thus no extra storage and computations are needed.

C. Experiments

We conduct experiments of OHN on the ImageNet classification benchmark. We first evaluate the effect of scaling factors for activation quantization. The results are shown in Table I. It can be seen that using the scaling factors of equation 2 can achieve good accuracy, but not the optimal. The proposed linear quantization induced scaling factors corresponding to equation 3 result in the highest accuracy. Thus in the following experiments, we adopt the linear quantization induced scaling factors for activation quantization.

TABLE III
THE ACCURACY OF ONE-HOT WEIGHT QUANTIZATION OF ALEXNET ON IMAGENET. 'A4' INDICATES THE RESULTS ARE WITH 4-BIT ONE-HOT ACTIVATION QUANTIZATION.

Model		Baselines	4+1-bit	5+1-bit	6+1-bit
AlexNet-A4	Top-1	58.7	58.2	58.3	58.2
	Top-5	79.6	79.4	79.3	79.3

TABLE IV
COMPARISON OF ACCURACY OF ONE-HOT QUANTIZATION AGAINST LINEAR QUANTIZATION OF ALEXNET AND VGG-16 ON IMAGENET.

Model		Baselines	Linear-A3-W4	OH-A4-W5
AlexNet	Top-1	60.5	58.4	58.2
	Top-5	80.9	79.7	79.4
VGG-16	Top-1	73.3	72.1	71.6
	Top-5	91.5	90.7	90.5

The results of activation quantization on AlexNet [9] are shown in Table II. To further illustrate the effect of weight quantization, we also quantize weights based on a 4-bit activation model, as shown in Table III. From these results, it can be concluded that one-hot quantization of activations could result in about 1% drop on Top-5 accuracy, and the subsequent weight quantization could result in additional 0.2%∼0.3% accuracy loss. Another finding is that after fine-tuning, networks with different bit width achieve similar accuracy. The reason is that one-hot representation has reduced resolution with respect to large values so that the benefit of adding too large values is limited.

To compare one-hot quantization against linear quantization, we conduct experiments based on AlexNet and VGG-16 [10], as shown in Table IV. We compare one-hot quantization using 4-bit activations and 5-bit weights, with the linear quantization using 3-bit activations and 4-bit weights. Our method achieves similar results compared with linear quantization. As for hardware acceleration, we will show that one-hot quantization is a promising alternative of linear quantization in the next few sections.

IV. EVALUATION ON HARDWARE ACCELERATORS

A. Setup

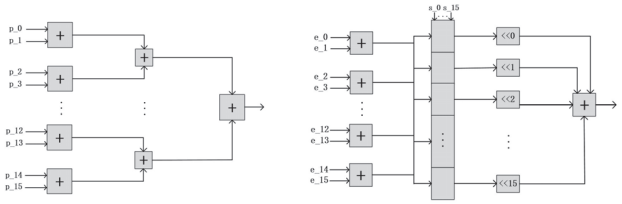
In this section, we demonstrate the efficacy of one-hot data representation on both bit-parallel and bit-serial state-of-the-art accelerators, i.e., *DaDianNao* [1] and *Laconic* [6]. We use the Xilinx Zynq ZC706 SoC (a ZC7045 FPGA with embedded CPUs) running at 150MHz for fast verification. We implement and synthesize all the accelerators with Vivado HLS 2018.2, the latencies and resource consumptions are from the Vivado reports.

B. One-Hot Data Format

In this paper, we represent both weights and activations as powers of two. In this scenario, the exponent bits and an additional sign bit are enough to encode the original value. For a fair comparison, we adopted the format in [6], in which the MSB of the encoded value is used to identify the sign, and the rest bits are used to represent the exponent.

C. Evaluation on DaDiaoNao

DadianNao is a state-of-the-art bit-parallel inference accelerator designed for CNN [1]. It consists of 16 tiles with 32MB distributed weight buffers and 4MB activation buffers. In each tile, 256 pairs of 16-bit input activations and weights can be computed in one cycle in an input-reuse manner to obtain 16 partial output results along the output channels.



(a) (b)

Fig. 1. Examples of (a) Adder tree. (b) Histogram-based reduction unit.

TABLE V
COMPARISON OF RESOURCE CONSUMPTIONS BETWEEN *DaDiaoNao* TILE AND ITS POWER-OF-TWO VARIANTS.

	DSP48E	FF	LUT
DaDN	64	18795	85538
DaDN_S	0	**7643**	50674
DaDN_OH	**0**	9691	**17506**

The function of a *DaDianNao* tile is to compute the inner products of 16-bit weight and activation pairs, so the core computing units are parallel multipliers followed by reduction trees. With one-hot encoding, the resource-consuming multiplications between weights and activations can be converted to pair-wise additions of low-bit exponents. Moreover, the subsequent adder tree can be altered by a histogram-base reduction unit [6], as shown in Fig. 1.

To demonstrate the efficiency of one-hot representation, we conduct an experiment based on a *DaDianNao* tile. We use 16-bit weights and activations in baseline (DaDN) and the corresponding one-hot encoding in DaDN_OH. We also compare our DaDN_OH with shift-based DaDN_S that with power-of-two weights. Results are shown in Table V. As expected, DaDN_OH is the most resource-efficient one among these three, it only consumes 20.5% LUTs and 51.6% FFs of the original DaDN and doesn't use DSP blocks at all. This is reasonable because in DaDN_OH, the pair-wise 4-bit exponent addition is extremely resource-saving than 16-bit multiplication. Although DaDN_S is a multiplier-free architecture as in [7], DaDN_OH outperforms DaDN_S by a large margin, demonstrating that quantizing both weights and activations to one-hot format is much more hardware-efficient than quantizing either of the two.

D. Evaluation on Laconic

Laconic is another state-of-the-art 8-bit CNN accelerator [6]. Unlike *DaDianNao*, it computes the inner product in a bit-serial manner. Moreover, *Laconic* adopts radix-4 booth encoding to obtain the effectual bits of both weights and activations further. Suppose a simple multiplication between an N_w-bit weight and an N_a-bit activation, a naive bit-serial PE will compute this with $N\_w \times N\_a$ cycles without considering the ineffectual zero bits. However, with booth encoding, *Laconic* can aggressively compress both weight and

TABLE VI
COMPARISON OF RESOURCE CONSUMPTIONS BETWEEN *Laconic* TILE AND ITS ONE-HOT COUNTERPART.

	DSP48E	FF	LUT
Lac	0	74798	99823
Lac_OH	0	**5970**	**31457**

TABLE VII
LAYER-WISE SPEEDUP OF ONE-HOT OPTIMIZED *Laconic* ON CONVOLUTIONAL LAYERS OF ALEXNET.

Layer	1	2	3	4	5	geomean
Speedup	6.32×	4.6×	5.28×	4.81×	3.98×	4.94×

activation to $N^{'}\_w$ and $N^{'}\_a$ bits, where $N^{'}\_w \leq N\_w$, $N^{'}\_a \leq N\_a$. Ideally, the acceleration rate of *Laconic* is

$$\frac{N\_w \times N\_a}{N'\_w \times N'\_a} \tag{5}$$

From the equation above, it is clear that if we can further constrain both $N^{'}\_w$ and $N^{'}\_a$ to 1, the acceleration rate will be pushed towards the maximum. That is, with one-hot representation, *Laconic* can be much faster.

A *Laconic* tile consists of an $N \times M$ PE array, a WSpad (Weight Scratchpad), a Aspad (Activation Scratchpad), and a PSpad (Partial sum Scratchpad). In this paper, we take a tile with 4×4 PEs as an example. The activations are shared along 4 columns to obtain the partial/final results of 4 adjacent output neurons. Likewise, the weights are shared along 4 rows to generate 4 outputs corresponding to 4 separate output channels. Each PE computes 16 pair-wise additions of exponents and histogram-based reduction.

It is worth noticing that in the middle of PE array and weight/activation scratchpads, there exist encoding units to execute booth encoding. That is, before sending to PEs, the weights and activations fetched from scratchpads are encoded into lists of effectual terms on the fly. Since each PE calculates 16 pairs of weight/activation in parallel, a total of $16 \times (4 + 4) = 128$ encoding units are required in a single tile. We observe that these booth encoding units account for nearly half of the LUT consumption of a tile, which leads to a significant overhead.

However, with one-hot representation, the booth encoding units are no longer needed, thus the resource usage on encoding can be eliminated. Moreover, the on-chip storage can be directly halved due to the reduced bit width of encoded values. We conduct an experiment on *Laconic* with 4×4 PEs to demonstrate resource efficiency. From the result shown in Table VI, we can see that Lac_OH only consumes 31.5% LUTs and 8% FFs of Lac.

Finally, we verify the inference speedup of convolutional layers of Lac_OH against Lac on AlexNet [9]. Results are shown in Table VII. With the proposed one-hot representation, Lac_OH is able to achieve an average of $4.94\times$ speedup using less than one-third on-chip computing resources of Lac. Note that the original *Laconic* suffers from synchronization problem, i.e., the PEs in a tile finish computing at different cycles due to the variable length of encoded terms. However, in one-hot representation, all the weights and activations are encoded to only one term, so the synchronization problem is avoided, which also contributes a lot to the performance gain.

V. CONCLUSION

In this paper, we propose a hardware-oriented One-Hot Networks that quantizing both weights and activations to powers of two. Extensive experiments on the algorithm side show that comparable accuracy on the ImageNet classification task can be obtained with one-hot quantization. We further integrate one-hot representation on two state-of-the-art CNN accelerators, *DaDianNao* and *Laconic*. Experiments on the bit-parallel *DaDianNao* show that up to 80% LUT saving can be achieved under the same inference speed. As for the one-hot optimized bit-serial *Laconic*, it can achieve $4.94\times$ speedup on AlexNet's convolutional layers with less than one-third LUT consumption. We leave the evaluations of area and energy efficiency on ASIC for future work.

VI. ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China (No.61972396, 61876182, 61906193), the Strategic Priority Research Program of Chinese Academy of Science (No.XDB32050200), and the Advance Research Program (31511130301).

REFERENCES

[1] Y. Chen et al., "DaDianNao: A Machine-Learning Supercomputer," *MICRO*, 2014
[2] N. P. Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit," *ISCA*, 2017
[3] R. Andri, L. Cavigelli, D. Rossi, L. Benini, "YodaNN: An Ultra-Low Power Convolutional Neural Network Accelerator Based on Binary Weights," *CoRR*, 2016
[4] Y. Umuroglu, "FINN: A Framework for Fast, Scalable Binarized Neural Network Inference," *FPGA*, 2017
[5] P. Judd, J. Albericio, T. H. Hetherington, T. Aamodt, A. Moshovos, "Stripes: Bit-Serial Deep Neural Network Computing," *MICRO*, 2016
[6] S. Sharify et al., "Laconic Deep Learning Inference Acceleration," *ISCA*, 2019
[7] H. Tann, S. Hashemi, R. I. Bahar, S. Reda, "Hardware-Software Codesign of Highly Accurate, Multiplier-free Deep Neural Networks," *DAC*, 2017
[8] S. Vogel, J. Springer, A. Guntoro, G. Aschied, "Self-Supervised Quantization of Pre-Trained Neural Networks for Multiplierless Acceleration," *DATE*, 2019
[9] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *NeurIPS*, 2012
[10] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition," *ICLR*, 2015