2019 22nd Euromicro Conference on Digital System Design (DSD)

Bit-shift-based Accelerator for CNNs with Selectable Accuracy and Throughput

Sebastian Vogel*✉, Rajatha B. Raghunath*§, Andre Guntoro*, Kristof Van Laerhoven§, Gerd Ascheid†
*Robert Bosch GmbH, Renningen, Germany §University of Siegen, Siegen, Germany
†RWTH Aachen University, Aachen, Germany
✉sebastian.vogel@de.bosch.com

***Abstract*—Hardware accelerators for compute intensive algorithms such as convolutional neural networks benefit from number representations with reduced precision. In this paper, we evaluate and extend a number representation based on power-of-two quantization enabling bit-shift-based processing of multiplications. We found that weights of a neural network can either be represented by a single 4 bit power-of-two value or with two 4 bit values depending on accuracy requirements. We evaluate the classification accuracy of VGG-16 and ResNet50 on the ImageNet dataset with weights represented in our novel number format. To include a more complex task, we additionally evaluate the format on two networks for semantic segmentation. In addition, we design a novel processing element based on bit-shifts which is configurable in terms of throughput (4 bit mode) and accuracy (8 bit mode). We evaluate this processing element in an FPGA implementation of a dedicated accelerator for neural networks incorporating a 32-by-64 processing array running at 250 MHz with 1 TOp/s peak throughput in 8 bit mode. The accelerator is capable of processing regular convolutional layers and dilated convolutions in combination with pooling and upsampling. For a semantic segmentation network with 108.5 GOp/frame, our FPGA implementation achieves a throughput of 7.0 FPS in the 8 bit accurate mode and upto 11.2 FPS in the 4 bit mode corresponding to 760.1 GOp/s and 1,218 GOp/s effective throughput, respectively. Finally, we compare the novel design to classical multiplier-based approaches in terms of FPGA utilization and power consumption. Our novel multiply-accumulate engines designed for the optimized number representation uses 9 % less logical elements while allowing double throughput compared to a classical implementation. Moreover, a measurement shows 25 % reduction of power consumption at same throughput. Therefore, our flexible design offers a solution to the trade-off between energy efficiency, accuracy, and high throughput.**

***Index Terms*—Deep Neural Networks, Quantization, Efficient Hardware Acceleration**

I. Introduction

In recent years, Deep Neural Networks (DNNs) have reached an unprecedented performance in a variety of challenging computer vision tasks, such as image classification [1] and semantic segmentation [2]. Consequently, these algorithms are promising candidates for future advanced driver assistance systems based on camera, radar, or lidar sensors. However, the computational workload of Deep Neural Networks is large and prevents their extensive deployment in resource constrained systems. Therefore, algorithmic advances to reduce the computational complexity of neural networks have been proposed such as number representations at reduced precision. In contrast to general-purpose processors, dedicated accelerators for neural networks can fully utilize the advantages of reduced precision and furthermore benefit from neural network inherent data reuse patterns.

With this paper we make the following contributions:

- We propose a number representation allowing efficient bit-shift-based multiply-accumulate engines and evaluate it on a variety of networks and datasets.
- We design and implement a multiply-accumulate engine capable of processing weights in our proposed number format which is configurable in terms of accuracy and throughput. The engines use 9 % less logical elements while allowing double throughput.
- We integrate the multiply-accumulate engines in an accelerator for semantic segmentation running at 250 MHz with a peak throughput of 1 TOp/s. We achieve 760.1 GOp/s on a use-case network and reach 1,218 GOp/s exploiting throughput benefits of our novel processing engines.

II. Related Work

Reduced precision for number representation in neural networks has been proposed by a variety of previous works. Most prominently, a special training procedure has been proposed for training neural networks with binary weight representation [3], [4]. However, this training procedure has not yet been extended to complex tasks such as semantic segmentation. Other previous work has been focusing on pre-trained neural networks and quantize weights using a nonuniform quantization procedure to better fit to data distributions in neural networks [5], [6]. To overcome performance degradations [5] retrain the network after quantization. A logarithmic number representation with arbitrary log-base which requires both weights and activations to be in a logarithmic number format is proposed in [6]. Accelerators based on bit-shift-multipliers similar to our work are reported in [7], [8].

In contrast to our work, previous work did not evaluate the benefits of an efficient multiply-accumulate engine based on bit-shifts with selectable accuracy and throughput.

III. Methodology

A. CNNs for Classification and Semantic Segmentation

Convolutional neural networks for image processing typically consist of a number of stacked layers, where convolutional layers are used for extracting features from an input

978-1-7281-2862-7/19/$31.00 ©2019 IEEE
DOI 10.1109/DSD.2019.00106

image. To reduce the spatial dimension so-called pooling layers map neighboring values to a single output value. In classification networks, fully-connected layers usually aggregate the extracted features and output a class label. In the case of semantic segmentation, so-called upsampling layers counteract the pooling operations and restore the spatial dimension of the input such that each input pixel can be classified with an individual class label. Mathematically, convolutional and fully-connected layers can be written as weighted sums, followed by a non-linearity function $\Phi(\cdot)$ using a bias term b:

$$y = \Phi\left(b + \sum wx\right) \tag{1}$$

State-of-the art networks typically feature a piecewise linear function $\Phi(x) = \text{ReLU}(x) = \max(x, 0)$.

B. Number Representation

While a floating-point format with 32 bit (*float32*) offers a large dynamic range and high-precision especially necessary during training, reduced precision representations are beneficial for the deployment of DNNs on embedded devices. In this section, we first discuss state-of-the-art linear and power-of-two quantization before introducing a novel number representation.

1) Linear Quantization: We define function $lin\_quant : x \mapsto x_q$ for mapping a value x from a quasi-continuous representation to a linearly quantized, signed value x_q according to (2), where N denotes the bit-width and Δ the step size:

$$x_q = \Delta \ \text{clip}\left(\text{round}\left(\frac{x}{\Delta}\right), -2^{N-1}, 2^{N-1} - 1\right). \tag{2}$$

Here, $\text{clip}(x, a, b)$ restricts x to the interval $[a, b]$. An example for linear quantization of weights is depicted in Fig. 1, where quantized sampling points are uniformly distributed. Hence the distance between neighboring values x_q is constant Δ.

2) Power-of-Two Quantization: In contrast to linear quantization, a power-of-two scheme incorporates a nonuniform distribution of sampling points. The mapping $pow2\_quant : x \mapsto x_q$ for signed power-of-two quantization with N bit and step size Δ is given by

$$\begin{aligned} x_q &= \text{sign}(x)\ \lfloor 2^{\hat{x}_q} \rfloor \Delta, \quad \text{where} \\ \hat{x}_q &= \text{clip}\left(\text{round}\left(\log_2 |\frac{x}{\Delta}|\right), \ -1, \ 2^{N-1} - 2\right). \end{aligned} \tag{3}$$

To allow $x_q = 0$, we clip $\hat{x}_q$ to the range $[-1, 2^{N-1} - 2]$. Consequently, $\hat{x}_q = -1$ corresponds to a zero-code to indicate

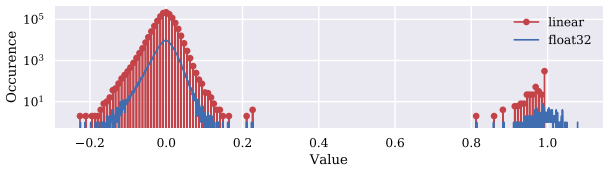

Fig. 1. Linear quantization of weights of a layer with bimodal distribution using 8 bit. Such weight distribution is typically found in layers which were initialized with an identity operation before training.

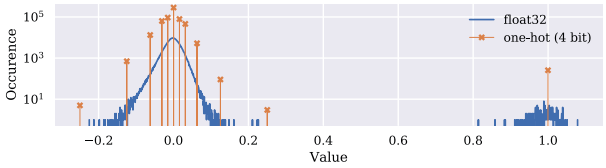

Fig. 2. Power-of-two quantization of weights of a layer with bimodal data distribution using 4 bit. Since a power-of-two quantized value carries at most a single 1 in a binary representation, we use the term *one-hot* synonymously.

$x_q = 0$. Instead of storing x_q in memory, we merely save the exponent value $\hat{x}_q$ using N bit.

Sampling points of a power-of-two quantization are shown in Fig. 2. As can be seen, the quantization is nonuniform with an aggregation of sampling points near zero and sparse sampling towards large absolute values. Especially for multimodal data distributions as examplarily depicted in Fig. 2, power-of-two quantization offers few samples for the second mode near 1. In a power-of-two quantization scheme, Δ denotes the smallest step size between 0 and the first nonzero value.

Using power-of-two values is advantageous because a simple bit-shift ($\ll$) can replace a costly multiplication:

$$w \cdot x_q = w \ll \hat{x}_q. \tag{4}$$

3) Two-Hot Quantization: Since the power-of-two quantization offers merely sparse sampling of large values, we present a number representation with the benefits of bit-shift-based multiplications while providing more sampling points towards large values. Instead of representing a value x_q with a single power-of-two value $\hat{x}_q$, we use two values to represent x_q such that $x_q = x_q^{MSB} + x_q^{LSB}$ and quantize x_q^{MSB} and x_q^{LSB} both as power-of-two values $2^{\hat{x}_q^{MSB}}$ and $2^{\hat{x}_q^{LSB}}$. We use $N/2$ bit for each exponent value $\hat{x}_q^{MSB}$ and $\hat{x}_q^{LSB}$. In total, value x_q is represented by N bit. Since the resulting value x_q carries at most two 1's in a binary representation, we refer to this number format as *two-hot*.

Multiplications with a two-hot quantized value are as cheap as two bit-shifts and an addition:

$$\begin{aligned} w_q \cdot x_q &= w_q \cdot (x_q^{MSB} + x_q^{LSB}) \\ &= \Delta \left(w \ll \hat{x}_q^{MSB} + w \ll \hat{x}_q^{LSB}\right). \end{aligned} \tag{5}$$

To quantize a value x in two-hot format, we first determine x_q^{MSB} according to

$$x_q^{MSB} = pow2\_quant_{N/2}(x, \Delta). \tag{6}$$

The second part x_q^{LSB} is then extracted from the remainder

$$\begin{aligned} x_\delta &= x - x_q^{MSB} : \\ x_q^{LSB} &= pow2\_quant_{N/2}(x_\delta, \Delta). \end{aligned} \tag{7}$$

In Fig. 3, the resulting sampling points of a two-hot quantization with 8 bit are shown. As can be seen, the two-hot scheme resolves the neighborhood of power-of-two sampling points with yet another power-of-two quantization.

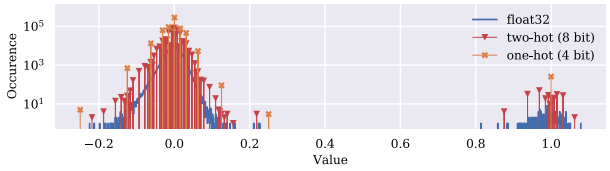

Fig. 3. Two-hot quantization of weights of a layer with bimodal data distribution. Two-hot quantization offers additional sampling points in the neighborhood of power-of-two (one-hot) quantized values.

C. Step Size Optimization

We determine the step size Δ based on the propagation of the quantization error δ through the neural network as we previously introduced in [9]. We write $\tilde{y}$ for the output of a layer with quantized weights w_q (compare (1)):

$$\begin{aligned} w_q &= quant\,(w, \Delta_w) = w + \delta_w, \\ \tilde{y} &= \Phi\left(\sum x w_q + b\right). \end{aligned} \tag{8}$$

To determine the optimal step size Δ_w, we minimize the resulting error at the output of the layer:

$$\Delta_w = \underset{\Delta_w}{\operatorname{argmin}}\left(\left|\tilde{y} - y\right|^2\right). \tag{9}$$

For the step size Δ_x of quantized input activations x follows:

$$\begin{aligned} \tilde{y} &= \Phi\left(\sum x_q w + b\right), \\ \Delta_x &= \underset{\Delta_x}{\operatorname{argmin}}\left(\left|\tilde{y} - y\right|2\right). \end{aligned} \tag{10}$$

This procedure is also used for finding a suitable quantization step size Δ_b for biases.

IV. Evaluation

We evaluate the two-hot quantization procedure on two networks for image classification. For that purpose we use the ILSVRC dataset [10]. This dataset contains more than 1 million RGB-images each categorized into one of 1,000 classes. The results of the following experiments are reported on the ImageNet validation set of 50,000 images. Reported performance figures are top-1 (highest rated label correct) and top-5 accuracy (correct classification among five highest rated labels). The networks for classification are VGG16 [11] and ResNet50 [12].

Furthermore, to include a more complex task, we additionally report results on Convolutional Neural Networks (CNNs) for semantic segmentation. As a benchmark, we use the Cityscapes dataset [13]. It includes 5,000 RGB-images with fine-grained labels for training and its validation set consists of 500 images. The images have a resolution of 2048×1024. On this task, we evaluate two models, Dilated Model [14] and FCN8s [2], by reporting mean Intersection over Union (mIoU) and mean pixel accuracy (pix. acc.) on the validation set after quantizing the networks. For the experiments with the Dilated Model, we downsampled images by a factor of $4\times$ to a resolution of 512×256.

A. Image Classification

Quantization results on the ImageNet dataset of ResNet50 and VGG-16 are listed in Table I. For this evaluation, activations and biases are quantized in an 8 bit linear format, whereas weights are quantized in either linear, power-of-two, or two-hot format. As can be seen, an 8 bit linear quantization of weights does not severely affect the network accuracy in comparison to the original floating-point weights. For the case of VGG16, the top-5 accuracy even increases slightly, which may emerge from regularizing effects. While a 4 bit power-of-two representation for weights reduces the network performance of VGG16 by less than 6 percentage points, the top-1 accuracy of ResNet50 drops by more than 20 percentage points. However, we observe comparably low accuracy degradation with 8 bit two-hot quantization of weights.

Since multiplications with power-of-two and two-hot weights are both based on bit-shifts, we furthermore evaluate the accuracy of a DNN with mixed power-of-two and two-hot quantized layers. For this analysis, we chose ResNet50 since its performance degradation for complete power-of-two quantization is more pronounced. Starting with the last layer, we quantize its weights with a power-of-two scheme and keep the remaining layers in two-hot format. This way, we quantize the network layer by layer until the weights of all layers are in power-of-two format. The corresponding accuracy numbers are plotted in Fig. 4. From these results, we see that particularly layers close to the input of ResNet50 are sensible to quantization noise whereas layers deeper in the network are more resilient to power-of-two quantization.

B. Semantic Segmentation

Semantic segmentation is a rather complex task which is especially promising for scene understanding and environment perception. On this task, we evaluate power-of-two and two-hot quantization on two networks in comparison to a linear fixed-point number representation. The quantization results are summarized in Table II. As can be seen, power-of-two quantization affects the network accuracy the most. A two-hot number representation for weights, results in little to no degradation of network accuracy compared to a linear quantization scheme. Similar to the analysis of ResNet50, we conduct an experiment where some layers of the Dilated Model have power-of-two quantized weights and the remaining weights

TABLE I
Comparison of 8 bit linear fixed-point number representation with 4 bit power-of-two and 8 bit two-hot quantization of weights on image classification networks.

	VGG-16		ResNet50	
accuracy [%]	top-1	top-5	top-1	top-5
float32 baseline	69.58	89.04	72.99	90.93
linear	69.12	89.06	71.67	90.73
power-of-two	63.85	86.76	46.36	72.11
two-hot	68.82	89.51	70.24	90.35
two-hot vs. linear	-0.30	+0.49	-1.43	-0.38

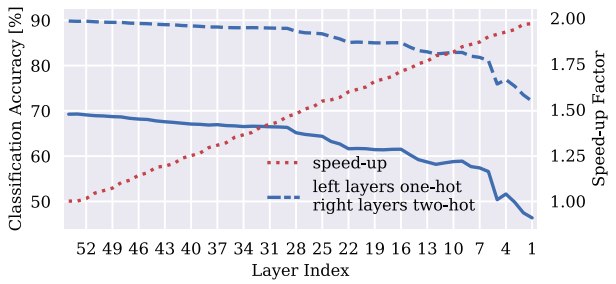

Fig. 4. Top-1 (solid line) and top-5 (dashed line) accuracy are reported when layers to the left of a given layer index are power-of-two quantized whereas layers to the right are in two-hot format. With dedicated processing elements, the power-of-two format results in a higher throughput. The resulting speed-up for processing a frame is reported.

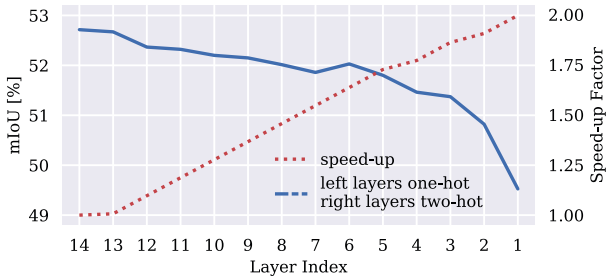

Fig. 5. mIoU of the Dilated Model when layers to the left of a given index are power-of-two quantized whereas layers to the right are in two-hot format. Dedicated processing elements allow a speed-up for the power-of-two format.

TABLE II
LINEAR, POWER-OF-TWO, AND TWO-HOT QUANTIZATION OF WEIGHTS ON SEMANTIC SEGMENTATION NETWORKS.

	Dilated Model		FCN8s	
accuracy [%]	mIoU	pix. acc.	mIoU	pix. acc.
float32 baseline	55.63	92.85	66.48	94.65
linear	55.62	92.78	66.47	94.44
power-of-two	49.52	90.13	60.75	92.10
two-hot	55.23	92.71	66.21	94.40
two-hot vs. linear	-0.39	-0.07	-0.26	-0.04

are in two-hot format. The results are reported in Fig. 5. Remarkably, when quantizing any layer of the Dilated Model in power-of-two format, a performance degradation of at least 3 percentage points is the consequence. We suppose this emerges from the reduced precision for bimodal distributions when quantizing dilated convolutional layers. Moreover, similar to the results on ResNet50, layers close to the input of the network are more susceptible to power-of-two quantization.

V. Neural Network Accelerator

A. Hardware Architecture

We evaluate our proposed two-hot number representation in a dedicated hardware accelerator for neural networks. The accelerator architecture consists of a processing array with 32-by-64 multiply-accumulate engines, weight buffers, and input and output buffers for intermediate data. The accelerator processes each layer of a neural network separately. We use the Dilated Model as a use-case for this accelerator. Since intermediate results of layers are larger than on-chip memory, they need to be transferred to external memory.

The computing block consists of an array of 32-by-64 processing elements (PEs). Each PE implements a multiply-accumulate operation. The array produces 64 output channels in parallel, each of which takes 32 input values that are aggregated to the output through an adder-tree.

B. Processing Elements

The processing elements implement the core multiply-accumulate operation. We propose a novel two-hot-based PE and compare it to a regular PE for multiplication of fixed-point values. The design of our PE which is capable of processing either a single weight in two-hot format or two weights in power-of-two representation is depicted in Fig. 7. Weights consist of two 4 bit power-of-two values each including a sign bit. In case of processing two-hot values, the input X is bit-shifted twice and the result is added and fed to an accumulator. If the engine is configured to process power-of-two quantized weights, the input X is bit-shifted but the results are forwarded to two separate accumulator stages. Consequently, the throughput of the second operating mode is doubled. For this reason, a speed-up factor is reported alongside accuracy numbers in Figs. 4 and 5.

C. FPGA Resource Utilization and Power Measurement

We evaluate our design in a prototypical implementation on a Xilinx Virtex UltraScale+ (XCVU9P) device. To compare our design with a multiplier approach in a fair manner, we switch off automatic DSP-slice instantiation during synthesis. As can be seen from Table III, the multiplier stage of a single twoh-hot-based PE uses 9 % less logical elements. Due to identical input sizes and similar output sizes, the number of

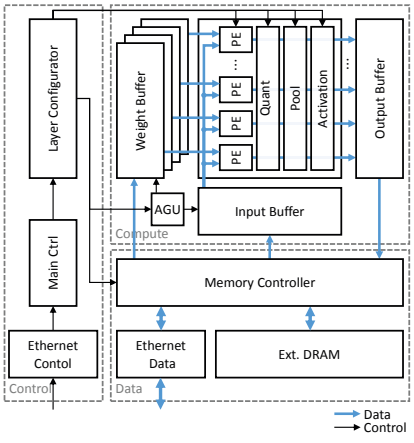

Fig. 6. Accelerator architecture with buffers for weight-, input-, and output-values, a processing array and control logic.

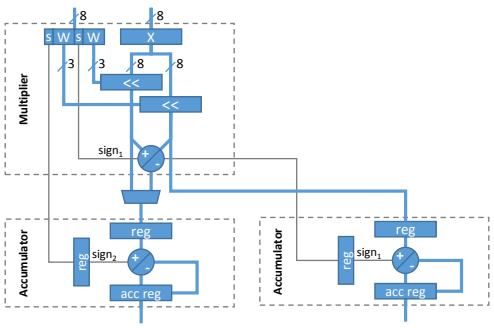

Fig. 7. Our proposed multiply-accumulate engine based on bit-shifts. The engine is capable of processing weights in two-hot format and additionally offers a mode to process two power-of-two quantized weights. In the two-hot-mode only one accumulator is being used whereas for the power-of-two-mode a second accumulator outputs a second accumulation result.

TABLE III
FPGA RESOURCE UTILIZATION OF OUR ACCELERATOR USING FIXED-POINT MULTIPLIERS OR TWO-HOT BIT-SHIFT-BASED PES.

	Multiplier	Two-Hot	rel. diff.
#LUTs (Mult. Stage)	66	60	-9.1 %
#Regs (Mult. Stage)	80	79	-1.2 %
#LUTs (total)	281,698	167,261	-40.6 %
#Regs (total)	268,326	267,317	-0.34 %

register elements is only marginally reduced. Nevertheless, the two-hot PE offers double throughput. Furthermore, we report the total utilization after synthesis of the accelerator. Compared to a regular design, the utilization of logical elements is reduced by 40.6 %. Our accelerator has a peak throughput of $2\,\text{Op} \times 32 \times 64 \times 250\,\text{MHz} = 1\,\text{TOp/s}$ using regular PEs and reaches double peak throughput employing our two-hot PEs. The dilated model has a total workload of 108.6 GOp/frame. The processing time on the accelerator with regular PEs and when all layers are two-hot quantized is 142.6 ms. When switching to power-of-two weights for all layers with more than 64 output channels, the execution time reduces to 89.1 ms, thereby increasing the effective throughput to 1,218 GOp/s.

The Xilinx VCU118 evaluation board allows rudimentary power measurements through the on-board Maxim power regulators. To allow a successful implementation of the more costly mutliplier design without DSP-slice instantiation, we reduced the PE-array size to 4×64 for a power measurement. The dynamic power consumption of the bit-shift-based design showed 2.55 W while a multiplier-based approach uses 3.4 W. Both designs have identical peak throughput. Therefore, our novel PEs allow a reduction of 25 % in power consumption.

VI. CONCLUSION

In this paper we proposed and evaluated two-hot quantization – a novel number representation allowing efficient processing of neural network based on simple bit-shifts. Furthermore, we designed and implemented a processing element with selectable accuracy and throughput. Our PE processes either a single weight in two-hot format or two weights in power-of-two representation and hence reaches double throughput at reduced precision. This PE offers a flexible solution for the accuracy vs. throughput trade-off. We additionally implement the bit-shift-based multiply-accumulate engines in a dedicated accelerator for neural networks for semantic segmentation. On a network with 108.5 GOp/frame, we achieve 781.2 GOp/s in 8 bit accurate mode and 1,218 GOp/s employing the 4 bit reduced precision mode. Furthermore, a power measurement on an FPGA evaluation board showed that our novel bit-shift-based processing elements allow 25 % power reduction compared to a regular fixed-point implementation. Therefore, our flexible design offers a solution to the trade-off between energy efficiency, accuracy, and high throughput. Future work will investigate the benefits of our proposed design on integrated circuit implementations.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, USA, 2012, pp. 1097–1105.

[2] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3431–3440.

[3] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Cambridge, MA, USA, 2015, pp. 3123–3131.

[4] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 29, 2016, pp. 4107–4115.

[5] D. Miyashita, E. H. Lee, and B. Murmann, "Convolutional neural networks using logarithmic data representation," *CoRR*, vol. abs/1603.01025, 2016.

[6] S. Vogel, M. Liang, A. Guntoro, W. Stechele, and G. Ascheid, "Efficient hardware acceleration of cnns using logarithmic data representation with arbitrary log-base," in *International Conference On Computer Aided Design (ICCAD)*, San Diego, USA, 2018.

[7] E. H. Lee, D. Miyashita, E. Chai, B. Murmann, and S. S. Wong, "Lognet: Energy-efficient neural networks using logarithmic computation," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA*, 2017, pp. 5900–5904.

[8] S. S. Sarwar, S. Venkataramani, A. Raghunathan, and K. Roy, "Multiplier-less artificial neurons exploiting error resiliency for energy-efficient neural computing," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2016, pp. 145–150.

[9] S. Vogel, J. Springer, A. Guntoro, and G. Ascheid, "Self-supervised quantization of pre-trained neural networks for multiplierless acceleration," in *Design, Automation and Test in Europe Conference Exhibition (DATE)*, Florence, Italy, 2019.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015.