GRLC: Grid-based Run-length Compression for Energy-efficient CNN Accelerator

Yoonho Park[†], Yesung Kang[‡], Sunghoon Kim[†], Eunji Kwon[†] and Seokhyeong Kang[†]
 {phoenix.yun,shkang}@postech.ac.kr

EE Department[†] and Future IT Innovation Laboratory[‡], POSTECH
 Pohang, South Korea

ABSTRACT

Convolutional neural networks (CNNs) require a huge amount of off-chip DRAM access, which accounts for most of its energy consumption. Compression of feature maps can reduce the energy consumption of DRAM access. However, previous compression methods show poor compression ratio if the feature maps are either extremely sparse or dense. To improve the compression ratio efficiently, we have exploited the spatial correlation and the distribution of non-zero activations in output feature maps. In this work, we propose a *grid-based run-length compression (GRLC)* and have implemented a hardware for the GRLC. Compared with a previous compression method [1], GRLC reduces 11 % of the DRAM access and 5% of the energy consumption on average in VGG-16, ExtractionNet and ResNet-18.

KEYWORDS

CNN Accelerator, Feature Map Compression, Spatial Correlation, Energy Efficiency, DRAM Access

1 INTRODUCTION

Recently, the accuracy of CNNs has improved because of a large network size; consequently, state-of-the-art CNNs require millions of computations and hundreds of megabytes of parameters [2, 3]. According to previous study [4], DRAM access is responsible for most of the energy consumption of CNN accelerators; therefore, to reduce energy consumption of a CNN accelerator, its feature maps are compressed to minimize the number of DRAM access generally. Although the compression requires additional energy consumption, the energy consumption of simple arithmetic operation or accessing internal buffers is much less than that of accessing DRAM [5].

For the effective compression of feature maps, previous researchers [5, 1, 6, 7] have only focused on sparsity resulting from rectified linear units (ReLU) and quantization. Because of the low computational complexity and ease of the “vanishing gradient” problem, state-of-the-art CNNs use a ReLU as an activation function, which is a piece-wise linear function that returns zero if the input is negative; otherwise, it directly returns the input. Consequently, the feature maps of CNNs using ReLU include many zeros and this status provides an opportunity to compress feature maps.

Quantization [6, 7] changes the precision from 32-bit floating-point to N -bit fixed-point, where N is usually smaller than eight. This is an effective method to compress feature maps. Quantization works well in both weights and feature maps of CNNs while it causes few accuracy drops.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISLPED '20, August 10–12, 2020, Boston, MA, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7053-0/20/08...\$15.00

<https://doi.org/10.1145/3370748.3406576>

Figure 1: (a) The feature maps in 12th layer of VGG-16. The 8-bit feature maps are mapped to a grayscale image. ‘0’ and ‘255’ are converted to white and black pixels, respectively. (b) Distribution of activations in maximum bit-width for data representation (x-axis, zero to 8-bit) of VGG-16, where the activations are quantized to 8-bit.

However, state-of-the-art feature map compression methods, i.e. run-length compression (RLC) [5] and zero-value compression (ZVC)[1] have only considered prevalent existence of zero-activations. They do not consider where the non-zero activations are located in the feature map and how the non-zero activations are distributed; they do not consider statistics of feature maps deeply. Intuitively, we identify two important observations of feature maps; (1) non-zero activations have a spatial correlation relationship, and (2) most of non-zero activations in feature maps are enough to be represented as $N/2$ -bit, even in the N -bit fixed-point system.

Figure 1 (a) shows the feature maps in 12th layer of VGG-16. As shown in the figure, non-zero activations, black or gray pixels are clustered, indicating that the activations of CNN are *spatially correlated*. Thus, if an activation has a non-zero value, its neighbor activations are likely to have a non-zero value and vice versa. Moreover, the sparsity of the feature map is also high. A few black pixels are shown owing to the inherent sparsity of the CNN.

Figure 1 (b) shows the distribution of activations of VGG-16, which can be divided into three categories in a fixed-point system; *zeros*, *outliers* and *non-outliers* [8]. Outliers are large data values which cannot be represented with $N/2$ -bit width in N -bit fixed-point. Outliers take a small portions of the feature maps but have a significant effect on the accuracy of CNNs. Moreover, outliers impose a limit on further quantization and its memory storage

overhead is non-negligible. In contrast, non-outliers can be represented sufficiently with $N/2$ -bit width. The bits that are not used by non-outliers are wasted because of the presence of outliers that require to use N bits. The waste of bits is more serious because the majority of non-zero data are non-outliers.

Motivated by these observations, we propose a novel compression method, *grid-based run-length compression (GRLC)*, which achieves a higher compression ratio and a lower DRAM access by adapting statistics of feature maps. To increase the compression ratio, we exploit the spatially correlated property of feature maps, which have non-zero values and are grouped in the spatial domain. Furthermore, the proposed GRLC deals with wasted bits of non-outliers and thereby increases the effectiveness of compression of feature maps.

The primary contributions of our work are as follows:

- We investigate the statistics of output feature maps; the spatial auto-correlation and the distribution of non-zero activations.
- We propose a novel compression method by exploiting the spatial correlation and data distribution of output feature maps. For the evaluation, we have compared the compression ratio of our method with both RLC[5] and ZVC[1].
- We have implemented our compression method into a hardware which compresses/decompresses feature maps in a CNN accelerator. The area and power overhead of the proposed circuits are evaluated.

In Section 2, we discuss the previous compression methods and their limitations. In Section 3, we describe the spatial correlation of feature maps. The GRLC is also proposed in Section 3. In Section 4, we describe the hardware implementation of GRLC. In Section 5, we describe the experimental setup and results. Finally, in Section 6, we provide the conclusion for this work.

2 RELATED WORK

Inspired by the sparsity of feature maps, there are two main streams representing feature maps; (1) focusing computation efficiency and (2) reducing DRAM access. The coordinate list format [9] and compressed sparse row/column formats [10, 11, 12] concentrate on computation efficiency. The coordinate list format can be exploited to store feature maps of CNNs [9]; it just uses a list of tuples (value, row, column) and then saves only the non-zero value, the row/column location of the non-zero value. Compressed sparse column format and compressed sparse row format save only the non-zero values and their indices [10, 11, 12]. The zero-free neuron array format, which is similar to the compressed sparse column format, makes it easy to index non-zero values directly by sacrificing of memory footprint.

The two aforementioned formats store only non-zero activations with the corresponding index; however, the compression ratio is insufficient because of relatively large indices. This weakness occurs because the primary goal of these formats is not to reduce DRAM access but to increase the computation efficiency by skipping zero activations.

In contrast, Sparse Exponential-Golomb (SEG) [13], Run-length Compression (RLC) [5] and Zero Value Compression (ZVC) [1] focus on reducing DRAM access. SEG[13] modifies the code of zero into one-bit code, $'0_{(2)}'$, to exploit the sparsity of feature maps. However, it requires expensive division and modulo operations. RLC[5] encodes consecutive zero values into a single run. After the run, the non-zero activation is appended. Because the compression ratio is highly dependent on the length of the *run*, it should be precisely determined by considering the sparsity of the feature maps. However, depending on the layer, the sparsity of the feature maps considerably varies, and RLC would be unsuitable for compressing the feature maps. ZVC [1] stores non-zero activations with mask bits to indicate whether the corresponding activation is zero or

Figure 2: Spatial correlation of the output feature maps between different distance at each layer (x-axis) in VGG-16. For the investigation, the feature maps of intermediate layers have been generated during inferences of 1k images of the ImageNet 2012 validation set.

not. ZVC achieves a higher compression ratio for floating-point precision. However, the compression ratio is poor for the feature maps that are quantized with 8-bits or less because of the fixed size of mask.

Overall, the previous compression methods are too complex to be implemented in hardware, or the compressed data have large data sizes because of the index. Although both ZVC and RLC are relatively simple to be implemented in hardware, they do not consider the spatial correlation and data distribution, which can be useful characteristics when they adopted to feature map compression method. Thus, in this work, we propose a compression method that utilizes the two important characteristics of feature maps. Furthermore, the proposed methods does not require expensive operations, such as division and modulo and that can be implemented with small area overhead. Regardless of the data sparsity, our proposed method shows excellent performance for quantized CNNs.

3 GRID-BASED RUN-LENGTH COMPRESSION

3.1 Spatial correlation

We have quantitatively investigated the spatial correlation¹ of activations using Moran's I [14]. For the investigation, we have trained VGG-16 to classify ImageNet 2012 [15] and quantized the network to 8-bit. Then, the intermediate layers of VGG-16 are extracted during inference of 1k validation set. Since we are only interested in the existence of non-zero values, we have converted the magnitude of all non-zero activations into '1'. To evaluate the spatial correlation of activations within distance d , we have defined a spatial coefficient, w_{ij} , as follows, where the each i and j are the index of activation in a feature map:

$$w_{ij} = \begin{cases} 1, & \text{if } |i - j| \text{ is } d \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Then, we have calculated Moran's I to evaluate the spatial auto-correlation of the converted feature maps [14]. Moran's I is defined as follows:

$$I = \frac{L}{\sum_i \sum_j w_{ij}} \cdot \frac{\sum_i \sum_j w_{ij} \cdot (x_i - \bar{x}) \cdot (x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2}, \quad (2)$$

where L is the number of activations in a feature map. x and $\bar{x}$ are the activations and the average of the activations, respectively. Moran's I has a value in the closed interval $[-1, 1]$. where $'-1'$ denotes that the non-zero activations and activations are perfectly dispersed, and $'1'$ denotes that the non-zero activations and zero activations are perfectly clustered. In general, if Moran's I of certain

¹Spatial correlation measured by Moran's I (most commonly used metric in geographic information science) helps us understand how much one activation is similar to other nearby activations.

Figure 3: Proposed compression formats. Feature map (top) is encoded in *GRLC without outlier consideration (GRLC w/o OC)* format (left-bottom) and *Grid-based run-length compression (GRLC)* format (right-bottom), where the grid-size is 2×2 for simple representation.

data is greater than $-1/(L-1)$, we can say that the data is spatially correlated positively.

We have calculated Moran's I of the activations within different d of different convolution layers. The correlation of the activations for close distance is higher (Figure 2). The activations that have a distance of one have an average Moran's I of 0.54, and as the distance increases, the average correlation coefficients decrease to 0.35, 0.29 and 0.26. Because the spatial correlation is always greater than $-1/(L-1)$, we can say that the activations in a feature map have a positive spatial correlation.

3.2 Proposed Compression Method

To exploit the spatial correlation of the feature maps, we divide a feature map into grid; we divide a feature map into a $m \times n$ (width $\times$ height) grid and encode data with *run* and *mask*.

Our proposed compression method, *grid-based run-length compression (GRLC)* classifies grids into zero and non-zero grids. In the Figure 3, a *zero grid* means a grid which includes only zero data and a *non-zero grid* includes at least one non-zero data entry. A *run* represents the number of consecutive zero grids. For example, two consecutive zero grids is represented as *run* = 10₍₂₎. The bit-width of *run* can be heuristically determined. The bit-width of *mask* is determined by the grid size m and n . After recording previous consecutive zero grids with *run*, GRLC appends *mask* to denote the position of the zero and non-zero data in each non-zero grid. After the *mask*, only non-zero activations are appended. If the number of the consecutive zero grids is greater than a maximum value of *run*, *run* is encoded to the maximum value and *mask* has all zero values. Our proposed method iterates these procedures and complete the feature map compression with a *end of packet* code which has all zero *run* and *mask*.

Because outliers in the activations are very sparse as shown in Figure 1 (b), we can further improve the compression ratio by using the bit-level sparsity of activations. As shown in Figure 3 (right-bottom), GRLC aggressively compresses the small-value data, non-outliers. Without considering the outliers, GRLC stores the non-zero activations with a fixed bit-width regardless of the magnitude.

Figure 4: The overview of typical CNN accelerator with a compressor and decompressor.

Figure 5: Micro-architecture of the GRLC compressor.

(Figure 3 (left-bottom)) If we use a half bit-width to store the non-outliers, we can improve the compression ratio further. Although *mask* of GRLC needs an additional bit to represent three data cases, zero (00₍₂₎), non-outlier (01₍₂₎) and outlier (10₍₂₎), this compression is much more effective if the ratio of non-outlier activations is high enough. The other operating principle of GRLC is similar to that of GRLC w/o OC.

4 HARDWARE IMPLEMENTATION OF GRLC

Figure 4 shows the overview of a typical CNN accelerator which consists of PE array, on-chip and off-chip memory, controller, compressor and decompressor. For the data computation in PE array, appropriate data in the off-chip DRAM are decompressed and loaded to the on-chip global buffer. The computed data in the PE array are transferred into the on-chip buffer and they are compressed to store in the off-chip DRAM.

Figure 5 shows a hardware implementation of the proposed GRLC compressor. In the first cycle, the incoming k ($k = m \times n$) activations (Din_0 - Din_{k-1}) in a grid are stored to parallel registers, where activations are divided into an upper $N/2$ -bit and a lower $N/2$ -bit to distinguish whether the activation is outlier, non-outlier or zero. In the next cycle, the shift-and-append logic collects only non-zero elements (denoted as p in Figure 5) among k activations and the comparators investigate whether each $N/2$ -bit data is zero. With *mask* table and the comparison results, the compressor generates 2-bit *mask* data (m_0 - m_{k-1}). The *mask* denotes that the data is outlier (10₍₂₎), non-outlier (01₍₂₎), or zero (00₍₂₎). Finally, if the current grid is a zero grid, the control logic increases *run* by one then moves to the next grid; otherwise the control logic records *run*, *mask* and non-zero activations on the buffer. The total latency to compress a grid varies from three to $4+k$ cycles, where the minimum cycle

Figure 6: Micro-architecture of the GRLC decompressor.

results from a zero grid and the maximum cycle is obtained when all k activations in a grid are non-zero.

Figure 6 shows the micro-architecture of the proposed GRLC decompressor. In the first cycle, two leading indicators, *run* and *mask*, are transferred to the decompressor at the same time. In the next cycle, the decompressor stores the lagging p non-zero activations (d_0 - d_{p-1}) into registers. In the next few cycles (less than or equal to the maximum magnitude of *run*), the decompressor transfers ‘0’s ($m \times n$) into the buffer because the *run* means the number of previous zero grids. In the final cycle, the non-zero activations are transferred to the buffer in the proper sequence according to the *mask* data. The decompressor iterates above sequence until a *end of packet* code comes in. The total latency to decompress a grid varies from one to $[3 + \text{the maximum magnitude of } run]$ cycles, where the minimum cycle results from decoding *end of packet* and the maximum cycle is obtained when *run* has its maximum magnitude. Note that the latency to (de)compress data can be reduced by N_c (N_d)-times, when the N_c (N_d) of (de)compressors are used in parallel. In our experiments, we have used four (de)compressors.

To evaluate the circuit overhead, the GRLC compressor and the decompressor are implemented using *Verilog HDL*. The RTL codes are synthesized with *Synopsys Design Compiler* [16] in the TSMC 65 nm worst corner library at a clock frequency of 250 MHz. The area and power consumption of the GRLC compressor/decompressor are summarized in Table 1. The grid size of compressor/decompressor is 2×2 and the process of scrutinizing the best grid size is described in Section 5.2. For the comparison, we also summarize the area and power consumption of a multiply-accumulate unit (MAC), which is a key component of CNN accelerators. The MAC comprises an 8-bit multiplier and a 28-bit accumulator. As shown in Table 1, the area and power consumption of the proposed GRLC compressor/decompressor are compatible with those of a MAC, which accounts for a few part of CNN accelerators. We also compared the circuit area and power consumption with those of 256 MACs that are exploited in a typical CNN accelerator [11]. The area and power overhead of the GRLC compressor/decompressor over the typical accelerator are only 4.62 % and 1.15 %, respectively. Moreover, based

Table 1: The area and power consumption of GRLC compressor/decompressor, multiply-and-accumulate unit (MAC) that consists of an 8-bit multiplier and an 28-bit accumulator and a typical CNN accelerator [11]

Component	Area [μm^2]	Power [mW]
GRLC Compressor	11,124	0.488
GRLC Decompressor	4,060	0.24
One MAC	1,283	0.248
256 MACs[11]	328,448	63.488

Figure 7: Reciprocal of the average of normalized DRAM access (z-axis) exploiting GRLC method with the different row size of grid (m , x-axis) and column size of grid (n , y-axis) in (a) VGG-16, (b) ExtractionNet and (c) ResNet-18, respectively. The feature map activations are quantized from 4-bit to 8-bit. The DRAM access in each quantized bit is normalized by the original data size.

on the RTL simulation, activation compression and decompression require only 7.7%, 2.1% and 7.6% additional latency comparing with a typical CNN accelerator (eyeriss-256, obtained by Timeloop[17]) in VGG-16, ExtractionNet and ResNet-18, respectively.

5 EXPERIMENTAL SETUP AND RESULTS

5.1 Experimental Setup

To evaluate the compression ratio, we implemented RLC, ZVC, and GRLC using C++. The intermediate feature maps were obtained from a deep learning framework, *Darknet*[18]. For the inference of the feature maps, we used three CNNs: VGG16, ExtractionNet, and ResNet-18, which are trained for the ImageNet ILSVRC 2012 image classification challenge [15]. The feature maps of three CNNs were quantized to k -bit using the following equation [6]:

$$Quant(x) = round\left(\frac{x(2^{k-1} - 1)}{\max(|x|)}\right) \cdot \frac{\max(|x|)}{2^{k-1} - 1} \quad (3)$$

The compression ratio of a compression method is defined as follows:

$$Compression\ ratio = \frac{data\ size\ before\ compression}{data\ size\ after\ compression} \quad (4)$$

The compression ratio is a positive rational number, whereas uncompressed data have the compression ratio of ‘1’.

5.2 Evaluation of Proposed Method with Various Grid Sizes

Compression effect was evaluated using the reciprocal of normalized DRAM access of the GRLC algorithm. We plotted the reciprocals of normalized DRAM access with various grid sizes for the

Figure 8: Compression ratio (lines, left y-axis) and sparsity (bars, right y-axis) in the output feature maps for each layer (x-axis) of (a) VGG-16, (b) ExtractionNet and (c) ResNet-18. 7 The feature map activations are quantized to 8-bit.

three different CNN networks. (Figure 7). Each DRAM access is an average access of various quantized feature maps (4-bit to 8-bit), and each average value was normalized by the original data size. VGG-16 has the smallest DRAM access when m and n are both two; DRAM access increases when either m or n is greater than two. Although the large grid size can compress many zeros with the zero-grid, because the opportunity that the all activations in a large grid are all zero is rare, the number of zero-grid decreased rapidly. This trend occurs because the average spatial correlation of VGG-16 decreases largely when the distance is less than three (Figure 2). This means that the number of zeros over 3-distance is relatively low; therefore, the number of zero-grids with size greater than 3 is low. Furthermore, because a large grid requires a large mask, the grid size should be chosen by considering the spatial correlation. As a result, DRAM access is minimum at $m = 2$ and $n = 2$ with 0.37 in VGG-16. Finally, we set the grid size as 2×2 which shows the lowest DRAM access (Section 5.2). Moreover, as shown in Figure 7 (b) and (c), we have exploited the reciprocal of normalized DRAM access of GRLC method with various grid sizes in ExtractionNet and ResNet-18. Normalized DRAM access is minimum at $m = 2$ and $n = 2$ with 0.37 and 0.34 in ExtractionNet and ResNet-18, respectively. Note that average DRAM access through entire layers is minimized with 2×2 grid size, not in each layer. The optimal grid size of each layers are different through the network.

5.3 Evaluation of the Compression Method

We evaluated the compression ratio of RLC8, RLC4, ZVC, GRLC without outlier consideration and GRLC method; RLC4 and RLC8

Figure 9: DRAM access of the compressed feature map for each quantized activations from 8-bit to 4-bit (x-axis) in (a) VGG-16, (b) ExtractionNet and (c) ResNet-18. The access has normalized by the DRAM access of uncompressed data.

denote RLC formats with 4-bit and 8-bit *run*, respectively. Figure 8 (a) shows the sparsity and the compression ratio of the feature maps in each layer of VGG-16. The compression ratio is significantly related to the sparsity of the feature maps. In deeper layers, the sparsity of the feature maps increases; thus, the compression ratios of all algorithms increase. GRLC shows the highest average compression ratio of 2.77, while the average compression ratios of RLC8, RLC4, ZVC are 1.60, 1.86 and 2.21, respectively.

In certain layers, the compression ratios of RLC4 and RLC8 are less than 1.0 owing to low sparsity; however, the compression ratios of ZVC, GRLC and GRLC without outlier consideration are always greater than 1.0. From the eighth layer onward, the difference between the compression ratios of GRLC and that of ZVC gradually increases because GRLC can effectively reduce the size of consecutive zero masks when the sparsity of the feature map is high. Although ZVC achieved a higher compression ratio compared to GRLC without outlier consideration in a few layers (second, third, fourth, and seventh layer), the differences in compression ratios between ZVC and GRLC without outlier consideration in these layers are less than 0.04.

GRLC also shows the best compression ratio in ExtractionNet (Figure 8 (b)). The average compression ratios of RLC8, RLC4, ZVC and GRLC for ExtractionNet are 1.73, 1.97, 2.32 and 2.84, respectively. Figure 8 (c) shows the sparsity and the compression ratios for ResNet-18. The average compression ratios of RLC8, RLC4, ZVC, GRLC without outlier consideration and GRLC are 2.21, 2.36, 2.66, 3.30 and 3.52, respectively. Note that the sparsity of ResNet-18 fluctuates across the shortcut layer.

Furthermore, we have measured the total DRAM access for the feature maps quantized to 8, 7, 6, 5 and 4-bit, which is required for each network and the results are shown in Figure 9. We evaluated DRAM access when the on-chip global buffer loads feature maps tiled in channel dimension for each layer. The feature map which is divided by channel dimension is saved in DRAM with the compressed form. In some CNN accelerators, feature maps can be tiled in not only channel dimension but also width or height dimension. However, the tile size of width and height dimension are not '1' in general, and the tiled feature maps have two-dimensional planes which have channel depth. Therefore, the spatial correlation will be preserved in those two-dimensional planes, and we expect that GRLC can be applied efficiently in that cases.

As shown in Figure 9, in all cases except 4-bit, GRLC achieved the greatest reduction in DRAM access; it achieved the second-greatest reduction (after GRLC without outlier consideration) at 4 bit case. For VGG-16 quantized to 8-bit (Figure 9 (a)), GRLC can save 50 % of DRAM access, while ZVC and RLC4, save 39 % and 19 % of DRAM access, respectively. Especially, because of inappropriate size of run-bit, the DRAM access of RLC8 exceeds the DRAM access of uncompressed data slightly (0.01 %). If we quantize the bit-width to be less than 8-bit, the DRAM access reduces regardless of the compression method, because this access reduction increased the sparsity of the feature maps. This decrease of ZVC is smallest in those five different compression method; because it uses a fixed size of mask. GRLC can successfully reduce the DRAM access for the quantized CNN. Especially, in 4-bit quantization, GRLC and GRLC without outlier consideration reduce DRAM access by 80 % and 82 %, respectively. GRLC without outlier consideration has lower DRAM access than GRLC, because the sparsity is sufficiently high in 4-bit quantization and the GRLC requires a larger mask than GRLC without outlier consideration.

As shown in Figures 9 (b) and 9 (c), the DRAM access of ExtractionNet and ResNet-18 showed similar tendencies to those of VGG-16. For 8-bit quantization, GRLC reduces DRAM access by 54 % and 57 %, respectively and GRLC without outlier consideration reduces DRAM access by 48 % and 50 %, respectively. In 4-bit quantization, GRLC reduces DRAM access by 76 % and 78 %, respectively, and GRLC without outlier consideration reduces DRAM access by 78 % and 80 %, respectively.

We have evaluated how much GRLC can reduce energy consumption. Figure 10 shows the energy consumed to save and load the feature maps of the intermediate layers. The energy consumption is normalized against a CNN accelerator without a compressor. As we described in Section 1, DRAM access is responsible for most of energy consumption. Therefore, even though the energy consumption of GRLC hardware is higher than that of RLC4 and ZVC, GRLC shows the minimum total energy consumption for saving and loading the feature map. GRLC reduces energy consumption for DRAM access by 0.35, 0.38 and 0.40, in VGG-16, ExtractionNet and ResNet-18, respectively. As a results, GRLC can save 37.7 % of energy consumption on average.

6 CONCLUSION

The number of DRAM access is responsible for most of the energy consumption of a CNN accelerator. To reduce the DRAM access, we propose a novel feature map compression method, GRLC, which exploits the spatial correlation and unbalanced value distribution in feature maps. GRLC compresses feature map more effectively saving a half of data bit-width of *non-outliers*. The compression ratio of GRLC is superior to existing compression method, RLC and ZVC. Upon further quantization of the CNNs (especially those below 8-bit), GRLC achieves the higher compression ratio and the lower DRAM access compared to RLC8, RLC4 and ZVC. The proposed GRLC compressor/decompressor can reduce DRAM access at the cost of only 0.728 mW of additional power consumption and 15, 184 μm^2 of circuit area overhead. Finally, the proposed GRLC

Figure 10: Energy breakdown of GRLC and GRLC without outlier in VGG-16, ExtractionNet and ResNet-18. Each energy consumption is normalized by energy consumption of uncompressed data.

method can reduce energy consumption for DRAM access by 37.7 % on average for the feature maps quantized to 8-bit. Our ongoing work seeks to improve the compression ratio with small accuracy degradation. If we damp-down the zero-adjacent pixels to zero, the efficiency of GRLC can be increased with small accuracy degradation.

ACKNOWLEDGEMENT

This work was supported by the Samsung Research Funding and Incubation Center of Samsung Electronics under Project Number SRFC-TB1703-07.

REFERENCES

- [1] N. Chatterjee J. Pool-Y. Kwon M. Rhu M. O'Connor and S. W. Keckler. 2018. Compressing dma engine: leveraging activation sparsity for training deep neural networks. In *Proc. International Symposium on HPCA*, 78–91.
- [2] S. Ren K. He X. Zhang and J. Sun. 2016. Deep residual learning for image recognition. In *IEEE/CVF Proc. CVPR*, 770–778.
- [3] K. Simonyan and Z. Andrew. 2014. Very deep convolutional networks for large-scale image recognition. (2014). arXiv: 1409.1556.
- [4] S. Yin L. Liu F. Tu W. Wu and S. Wei. 2018. Rana: towards efficient neural acceleration with refresh-optimized embedded dram. In *ACM/IEEE Proc. ISCA*, 340–352.
- [5] J. Emer Y. Chen T. Krishna and V. Sze. 2016. Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks. In *IEEE Proc. ISSCC*, 262–263.
- [6] J. Dotzel C. De Sa R. Zhao Y. Hu and Z. Zhang. 2019. Improving neural network quantization without retraining using outlier channel splitting. In *Proc. ICML*, 7543–7552.
- [7] S. Talathi D. Lin and V. Annapureddy. 2016. Fixed point quantization of deep convolutional networks. In *Proc. ICML*, 2849–2858.
- [8] D. Kim E. Park and S. Yoo. 2018. Energy-efficient neural network accelerator based on outlier-aware low-precision computation. In *ACM/IEEE Proc. ISCA*, 688–698.
- [9] R. Huang J. Zhang W. Lin L. Lu J. Xie and Y. Liang. 2019. An efficient hardware accelerator for sparse convolutional neural networks on fpgas. In *IEEE/CVF Proc. CVPR*, 17–25.
- [10] H. Mao J. Pu A. Pedram M.A. Horowitz S. Han X. Liu and W.J. Dally. 2016. Eie: efficient inference engine on compressed deep neural network. In *Proc. ACM/IEEE ISCA*, 243–254.
- [11] J. Emer Y. Chen T. Yang and V. Sze. 2019. Eyeriss v2: a flexible accelerator for emerging deep neural networks on mobile devices. In *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(2), 292–308.
- [12] P. Coussy H. Wouaf C. Chavet and R. Danilo. 2017. Efficient scalable hardware architecture for highly performant encoded neural networks. In *International Workshop on Signal Processing Systems (SiPS)*, 1–6.
- [13] G. Georgiadis. 2019. Accelerating convolutional neural networks via activation map compression. In *IEEE/CVF Proc. CVPR*, 7085–7095.
- [14] P. A. P. Moran, editor. 1950. *Notes on Continuous Stochastic Phenomena*. Number 1-2. Volume 37. Biometrika, 17–23.
- [15] H. Su S. Satheesh S. Ma Z. Huang A. Karpathy A. Khosla M. Bernstein A. C. Berg O. Russakovsky J. Deng and L. Fei-Fei. 2015. Imagenet large scale visual recognition challenge. In *International Journal of Computer Vision (IJCV)*, 115(3), 211–252.
- [16] Synopsys. 2011. *Synopsys Design Compiler User Guide*. <http://www.synopsys.com>. Synopsys.
- [17] Y. S. Shao Y.-H. Chen V. A. Ying A. Mukkara R. Venkatesan B. Khailany S. W. Keckler A. Parashar P. Raina and J. Emer. 2019. Timeloop: a systematic approach to dnn accelerator evaluation. In *IEEE Proc. International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 304–315.
- [18] J. Redmon. 2013–2016. Darknet: open source neural networks in c. <http://pjreddie.com/darknet/>.