Evaluating Data Resilience in CNNs from an Approximate Memory Perspective

Yuanchang Chen[1], Yizhe Zhu[4], Fei Qiao[1]*, Jie Han[2], Yuansheng Liu[3] and Huazhong Yang[1]
[1]Dept. of Electronic Engineering, Tsinghua University, China
[2]Dept. of Electrical and Computer Engineering, University of Alberta, Canada
[3]College of Robotics, Beijing Union University, China
[4]Beijing University of Posts and Telecommunications, China
qiaofei@tsinghua.edu.cn

ABSTRACT

Due to the large volumes of data that need to be processed, efficient memory access and data transmission are crucial for high-performance implementations of convolutional neural networks (CNNs). Approximate memory is a promising technique to achieve efficient memory access and data transmission in CNN hardware implementations. To assess the feasibility of applying approximate memory techniques, we propose a framework for the data resilience evaluation (DRE) of CNNs and verify its effectiveness on a suite of prevalent CNNs. Simulation results show that a high degree of data resilience exists in these networks. By scaling the bit-width of the first five dominant data subsets, the data volume can be reduced by 80.38% on average with a 2.69% loss in relative prediction accuracy. For approximate memory with random errors, all the synaptic weights can be stored in the approximate part when the error rate is less than 10^{-4}, while 3 MSBs must be protected if the error rate is fixed at 10^{-3}. These results indicate a great potential for exploiting approximate memory techniques in CNN hardware design.

Keywords

Data Resilience Evaluation; Convolutional Neural Network; Approximate Memory

1. INTRODUCTION

For many machine-learning tasks, a convolutional neural network (CNN) is a state-of-the-art technique with the processing capacity of huge data volumes and high computational demands. A number of designs [1, 2] focus on the computational part of the algorithm and aim to achieve fast efficient networks. However, efficient memory and data access are crucial for high-performance implementations of a CNN. ShiDianNao [3] focuses on minimizing memory transfers to achieve high efficiency, but it is only available for small-scale neural networks rather than large-scale ones. It is imperative to explore data storage and transmission optimization for both small and large neural networks.

Approximate memory is one of the most promising techniques to achieve efficient memory and data access in CNN hardware implementations. The main idea of approximate memory is leveraging the inherent data resilience of applications to trade off output quality for improved performance, such as energy efficiency and processing capacity. Prevalent approximate memory techniques can be divided into four categories: on-chip memory design [4, 5], off-chip memory design [6, 7], approximation in off-chip memory access [8, 9], and approximate storage in emerging devices [10, 11].

In order to apply suitable approximate memory techniques, it is important to understand data resilience in greater detail. It is required to quantitatively evaluate the data resilience of a given CNN, and identify which parts of the data are amenable to be stored in approximate memories. Motivated by the above, we propose a data resilience evaluation (DRE) framework to aid designers in adopting approximate memory techniques in CNN hardware design. The proposed DRE framework differs from previous paradigms [12, 13, 14] in one major point. All the above frameworks are established from the perspective of computation, mainly focused on the approximation in arithmetic units. Due to the fact that approximations in memory access and data transmission are distinct from that in arithmetic units, these paradigms are not applicable in approximate memory architecture design. This is the main reason for proposing the DRE framework.

The key contributions of this paper are summarized as:

- The DRE framework: Our primary contribution is a systematic framework (i.e., the DRE framework) established from the perspective of data storage in memory. It is used to quantitatively evaluate the data resilience of a given CNN and help designers to quickly estimate the potential of applying specific approximate memory techniques.

- A benchmark analysis: To verify the effectiveness of the proposed DRE framework, we characterize and analyze the inherent data resilience of several prevalent CNNs. We demonstrate the high degree of data resilience in these neural networks and emphasize the potential of approximate memory techniques.

- A design guide: Based on the analysis, we present several strategies for designers in the adoption of various

GLSVLSI'2017, May 10-12, 2017, Banff, AB, Canada.

DOI: http://dx.doi.org/10.1145/3060403.3060435

approximate memory techniques in CNN hardware implementations.

The rest of the paper is organized as follows. Section 2 introduces the approximation models for prevalent approximate memory techniques and analyze the data resilience of CNN algorithms. The proposed DRE framework is presented in Section 3. Simulation results are provided in Section 4. Finally, Section 5 concludes the paper.

2. CNN DATA RESILIENCE ANALYSIS

2.1 Modeling Approximate Memory

Inherent data resilience is defined as the property of an application to produce acceptable outputs despite some of its (input and intermediate) data being approximate. Approximate memory techniques exploit the inherent data resilience of applications to trade off output quality for improved performance.

The data of a given algorithm can be partitioned into two parts: the resilient part and the sensitive part. Approximate memory techniques should be targeted towards resilient data while avoiding the sensitive ones. For the resilient part of data, it is imperative to evaluate the applicability of various approximate memory techniques. Generally, we use approximation models to abstract a wide range of approximate memory techniques so that we can make quick evaluations. The approximation models of different approximate memories greatly depend on how the techniques are utilized. Most approximate memory techniques can be modeled as random errors that are uniformly injected into resilient bits [4, 5, 6, 7, 10, 11]. For off-chip memory reduction, major approximate techniques focus on data bit-width scaling [9]. The other approximate memory techniques are usually related to the characteristics of applications [8]. Table 1 summarizes the prevalent approximation techniques and their models. By using these approximation models for resilient data subsets, we can quickly evaluate the applicability of various approximate memory techniques.

Table 1: Approximate Memory Model

Approximate Memory Technique	Approximation Model
On-chip Memory [4, 5]	Uniformly Injected Random Errors
Off-chip Memory [6, 7]	
Emerging Device [10, 11]	
Off-chip Memory Access [9]	Bit-Truncation

2.2 CNN Data Resilience

CNNs play important roles in the machine-learning domain. Many CNN hardware designs focus on fast and efficient implementations of feed-forward networks because off-line learning is sufficient for many applications. It consists of a number of layers, including convolutional layers, pooling (sub-sampling) layers and a multi-layer perceptron (MLP) which consists of several full connection layers at the top of the network. These layers are executed in sequence so that they can be considered independently. The operands of each layer includes a number of neurons (output of the previous layer) and synaptic weights obtained from early off-line training. Figure 1 shows the major data flow of a typical CNN feed-forward network example. The inherent

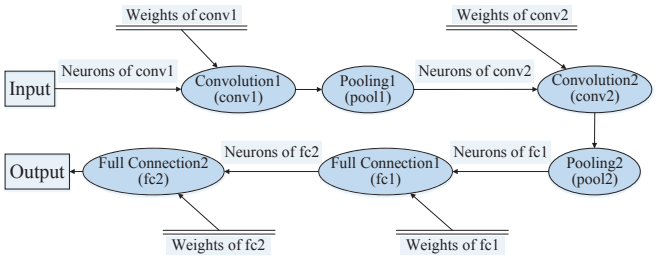

Figure 1: Data Flow of a CNN Example.

data resilience of a CNN can be reflected in the following two aspects.

2.2.1 *Numerical Representation Requirement*

Firstly, the data resilience of CNNs is reflected in their numerical representation requirements. Many implementations use the worst-case numerical precision for all values. Most software implementations use 32-bit floating-point numbers, while hardware implementations use 16-bit/8-bit fixed-point numbers. However, the numerical representation requirement of each layer is different. The numerical representation requirement of CNNs significantly varies not only across networks but also across the layers of the same network [15]. Their operand bit-width can be decreased according to their minimum numerical requirements so that memory access and data transmission can be significantly reduced. Therefore, we use numerical representation requirement to characterize the resilience of CNN data, including neurons and weights of each layer.

2.2.2 *Random Error Tolerance*

Secondly, the data resilience of CNNs is reflected in the random error tolerance. It is described as an maximum tolerable random error rate (MTRER) injected into the resilient bits with a negligible decrease in the prediction accuracy. For most approximate memories with random errors, the difference between different accuracy levels is beyond orders of magnitude [4, 5, 6, 7, 10, 11]. Therefore, the order of magnitude of MTRER is sufficient to characterize the tolerance of random errors. From a different perspective, the random error tolerance can also be described as the number of MSBs (Most Significant Bits) that must be stored in the precise part. Given approximate cells with a specific error rate, many LSBs (Less Significant Bits) can be stored in these cells, while some MSBs must be precise (protected). The number of protected MSBs can be used to characterize the random error tolerance of a given CNN.

3. PROPOSED DRE FRAMEWORK

The proposed data resilience evaluation (DRE) framework is shown in Figure 2. It consists of three analysis modules and a characterization module. The three analysis modules are for: (1) single layer analysis; (2) comprehensive multi-layer analysis; and (3) random error tolerance evaluation. Module 1 and Module 2 implement the two steps for numerical representation requirement analysis, while Module 3 evaluates the tolerance of random errors. The whole DRE framework is realized on the Caffe [16] platform. The inputs are the network description (.prototxt), synaptic weights (.caffemodel), a representative validation data set and the

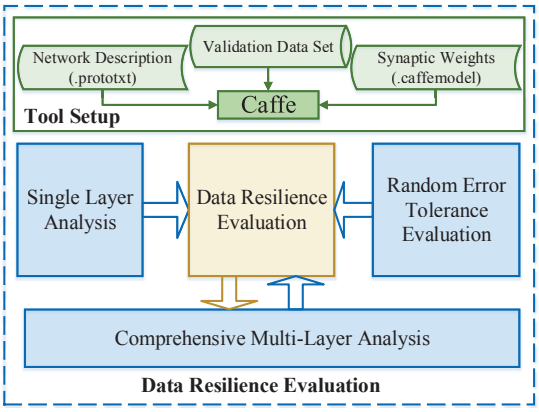

Figure 2: Overview of the DRE framework.

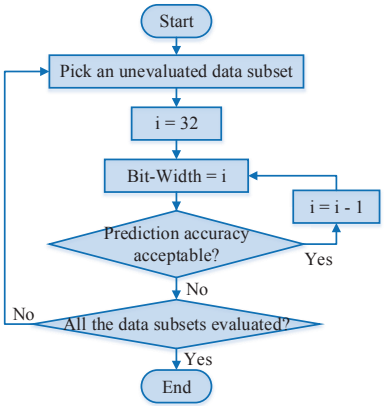

Figure 3: The Flow of Single Layer Analysis.

user-defined classification accuracy requirement. The outputs include a list of resilient data subsets, the results of evaluating the two approximation models mentioned in Section 3.2 on the resilient data subsets and their impacts on the prediction accuracy. In the sequel, we will describe the three analysis modules and the characterization module of the DRE framework in detail.

3.1 Single Layer Analysis

The first procedure of the framework is the single layer analysis. For a CNN algorithm, the control logic is highly sensitive so that it must avoid approximation. Neurons and weights of each layer are potentially resilient data subsets. The objective of the single layer analysis is to analyze and evaluate the numerical representation requirement of neurons and weights of each layer.

To explore the numerical representation requirement of a single layer, bit truncation is a commonly used technique, where approximations are introduced by reducing the data bit-width. Considering that Caffe is a 32-bit floating-point computing platform, we use 32-bit floating-point number as a baseline and the initial numerical representation in this module. The flow of this module is shown in Figure 3. We first pick a resilient data subset and transform it into the chosen numerical representation to observe the impact on the prediction accuracy of the network. If the drop in prediction accuracy is acceptable, reduce the bit-width of the chosen numerical representation by bit truncation and repeat the last step. Otherwise, the numerical representation requirement of this data subset is the chosen bit-width. Through these steps, we can derive the numerical representation requirement of the neurons and weights at each layer of CNNs. By truncating a resilient data subset into its minimum bit-width requirement, we can achieve some performance improvements in spite of a slight decrease in prediction accuracy.

3.2 Comprehensive Multi-Layer Analysis

The second procedure of the framework is a comprehensive analysis of multi-layers, which aims to derive a complete numerical representation scheme for all data subsets. This is an optimization problem with a search space of exponential scale. However, the number of neurons and synapses of each layer varies significantly. Usually, there are several dominant data subsets contributing to the total data volume. Approximate memory techniques should be targeted towards these data subsets rather than all the data in the algorithm. Therefore, the intent of this procedure is to explore a numerical representation scheme for several dominant data subsets that mainly contribute to the total data volume. Other data subsets remain using their numerical representation in 32-bit floating-point numbers.

The comprehensive analysis of multi-layers uses the same strategy as the single layer analysis. However, there are two differences. First, the judging conditions of this procedure are different from that of the single layer analysis. Not only the prediction accuracy but also the ratio of gain and loss should be considered in this procedure, while the single layer analysis only considers the former. Second, the working way of truncated operations are different from that in the single layer analysis. Truncations of different data subsets work together to influence the prediction accuracy of CNNs, while they are in isolation in the single layer analysis. The desired numerical representation scheme of a given CNN can be obtained through the procedure of Module 1 and 2.

3.3 Random Error Tolerance Evaluation

The last analysis module of the framework is for random error tolerance evaluation. Considering the fact that the approximation of many approximate memory techniques is modeled as random errors, we need to explore the random error tolerance of the neurons and weights of a given CNN. In this module, we also use the same strategy as in the single layer analysis, while the operations are random error injections other than bit truncations. Random errors with a given probability are introduced into specific resilient data subsets.

There are two parameters to evaluate the random error tolerance of the data in a CNN. The first one is the MTRER. Given a resilient data subset, the output quality is acceptable when the probability of random error is within a certain range while it is unacceptable when the probability is out of the range. The maximum probability within this range is the MTRER, which can be used to characterize the random error tolerance of this resilient data subset. On the other hand, when the probability of the random error is given, we need to determine which parts of data can be stored in such approximate cells while others must be precise. In most cases, LSBs can be stored in approximate cells while MSBs must be protected. Therefore, the amount of protect-

Table 2: Data Volume Statistics of Dominant Data Subsets

Network	Validation Data Set	Dominant Data Subsets (Major contribution to the total data volume)					
		subset1	subset2	subset3	subset4	subset5	sum
LeNet-5	MNIST	weight_fc1 (65.59%)	weight_fc2 (21.52%)	weight_conv2 (5.12%)	neuron_conv2 (1.84%)	weight_fc3 (1.79%)	95.86%
Cifar10-full	CIFAR-10	weight_conv3 (49.27%)	weight_conv2 (24.64%)	weight_fc1 (9.85%)	neuron_conv2 (7.88%)	neuron_conv1 (2.96%)	94.60%
AlexNet	ImageNet	weight_fc1 (61.50%)	weight_fc2 (27.33%)	weight_fc3 (6.67%)	weight_conv3 (1.44%)	weight_conv4 (1.08%)	98.02%
VGG-16	ImageNet	weight_fc1 (69.68%)	weight_fc2 (11.38%)	weight_fc3 (2.78%)	neuron_conv1-2 (2.18%)	weight_conv5-3 (1.60%)	87.62%

Notes: Subset weight_layer means the weights of this layer, while subset neuron_layer means the neurons of this layer. For example, weight_fc1 means the weights of layer fc1 (the first full-connection layer), while neuron_conv1 means the neurons of layer conv1 (the first convolution layer). The percentages are their contributions to total data volume.

ed MSBs can be used as a metric to evaluate the random error tolerance of these resilient bits.

3.4 Data Resilience Characterization

Data resilience evaluation goes through the whole process of the proposed DRE framework. It gives periodical evaluations as indicated in the three analysis modules and draws a concluding evaluation of data resilience for a given CNN. A periodical evaluation for single layer analysis mainly contains the bit-width requirement of each single layer and the benefits of corresponding bit-truncated operations. In a comprehensive analysis of multi-layers, we derive a complete numerical representation scheme based on the periodical evaluation in the single layer analysis. An MTRER and the amount of protected MSBs with a specific error probability are given in the random error tolerance evaluation. Additionally, the impact on prediction accuracy is also estimated during every evaluation in the above three modules.

4. SIMULATION RESULTS

The DRE framework is applied on a benchmark suite including CNNs with different scales and validation data sets with different kinds of features. The benchmark suite is shown in Table 2. We characterize the data resilience using the DRE framework and present the results in this section. Simulations are implemented on the Caffe platform [16]. The inputs of the framework include a network description (.prototxt), synaptic weights (.caffemodel), a representative validation data set and a user-defined prediction accuracy requirement. The network definitions (.prototxt) and pre-trained synaptic weights (.caffemodel) are taken from BVLC Caffe GitHub [17]. We assume that the acceptable relative reduction of prediction accuracy is 3% and use it as an example to conduct the following simulations.

4.1 Single Layer Evaluation

The intent of the proposed DRE framework is to evaluate the data resilience of dominant resilient subsets rather than all the data in the algorithm. Dominant subsets in our framework are defined as those mainly contributing to the total data volume. The dominant subsets of the benchmark suite and their contributions are shown in Table 2. It can be seen that top-5 subsets occupy a major contribution to the total data volume. The sum of the contribution of these subsets ranges from 87.62% to 98.02%. Therefore, it is sufficient to focus approximate memory techniques on these subsets.

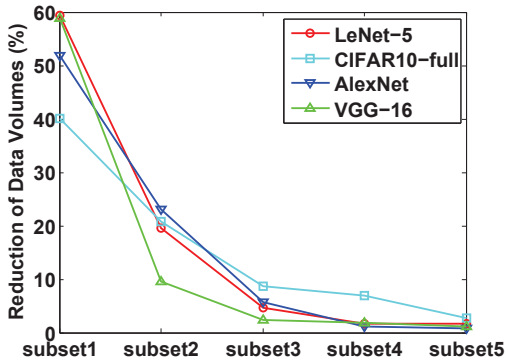

Figure 4: Benefits of single subset optimizations.

Table 3: Single Layer Bit-Width Requirements

Network	Bit-Width Requirement				
	subset1	subset2	subset3	subset4	subset5
LeNet-5	3	3	3	3	3
Cifar10-full	6	5	4	4	3
AlexNet	5	5	5	7	7
VGG-16	5	5	4	5	8

In this section, a bit-truncated operation is conducted on a single data subset to explore its numerical representation requirement, while other subsets remain as 32-bit floating-point numbers. The format of the target subset is first transformed from 32-bit floating-point to 32-bit fixed-point numbers. The 32-bit fixed-point numbers are then truncated into fewer bits. Table 3 shows the numerical representation requirement of the dominant data subsets in the single layer analysis. The bit-width requirements of these subsets range from 3 to 8 bits, which are far less than the original numerical representation (32-bit floating-point number) of the Caffe platform. This demonstrates a high degree of inherent resilience in these subsets. Benefits of those truncated operations (by single subset optimization) are presented in Figure 4. It can be seen that a larger reduction of data volume is obtained by the optimization of a more dominant subset. According to this observation, certain subsets can be targeted and processed (bit-truncated) with priority in the comprehensive multi-layer analysis, thereby reducing the search

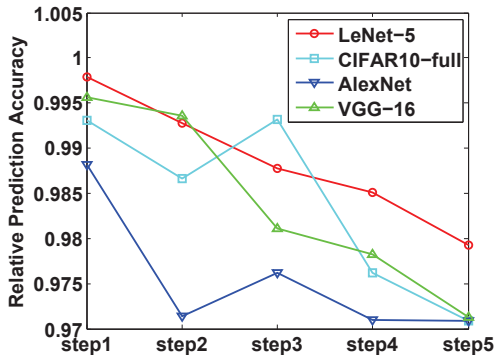

Figure 5: Trend of Prediction Accuracy.

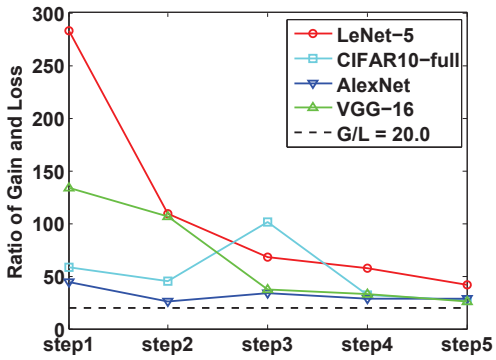

Figure 6: Ratio of Gain and Loss.

space of the optimization problem. The processing priority is determined by the size of each data subset.

4.2 Numerical Representation Scheme

The objective of the comprehensive multi-layer analysis is to derive a numerical representation scheme for dominant resilient data subsets. Truncations of each subset are conducted step by step to explore the connected bit-width requirements of these subsets. The first step in this section is to set the first dominant subset (subset1) as its bit-width requirement in a single layer evaluation. In Step i (i ranges from 2 to 5), the bit-width of subset i is scaled to reach its bit-width requirement based on the previous truncated operations. Bit-truncated operations work together to affect the output quality of a given CNN in this process. The complete numerical representation scheme is obtained through the above 5 steps, as presented in Table 4. Compared with their single layer bit-width requirements in Table 3, the combined bit-width requirements of some subsets are the same as those in the single layer analysis, while others are several (from 1 to 3) bits more. This implies that a combined operation can make a better use of the inherent data resilience. With this numerical representation scheme, the total data volume decreases on average by 80.38% with 2.69% decrease in relative prediction accuracy.

Table 4: Numerical Representation Scheme

Network	Complete Scheme					DVR[1]	RPA[2]
	S1	S2	S3	S4	S5		
LeNet-5	3	3	4	4	4	86.60%	97.93%
Cifar10-full	6	5	5	4	3	78.71%	97.09%
AlexNet	5	5	5	8	10	82.40%	97.09%
VGG-16	5	5	4	6	8	73.80%	97.13%

4.2.1 *Impact on Prediction Accuracy*

Prediction accuracy is the most important metric to measure a CNN's performance. The impact on prediction accuracy of the above steps is explored in this section, as shown in Figure 5. Basically, the prediction accuracy decreases step by step. Occasionally, it increases in a specific step, which implies that the bit-width requirement of different data subsets are interconnected with each other. Sometimes combined operations of serval subsets can achieve a better effect than single subset optimization.

[1]DVR: Data Volume Reduction.
[2]RPA: Relative Prediction Accuracy.

4.2.2 *Gain and Loss*

Gain and loss must be considered in the tradeoff of data volume reduction (DVR) benefits and accuracy loss. The gain in this section is defined as the reduction of data volume, while the loss is defined as the decline in relative prediction accuracy. The data volume reduction after step n can be calculated by (1). In (1), DVR_n represents the percentage of data volume reduction after step n, while DVC_i and BW_i represent the contribution (as shown in Table 2) and the bit-width of subset i. Figure 6 shows the trend of the ratio of gain and loss (G/L). For all the CNNs in the benchmark suite, this ratio follows a decreasing trend. However, the ratios are all larger than 20.0 after the above 5 steps. That means a small loss in relative prediction accuracy can be used to trade for larger benefits (DVR) that is over 20 times more than the loss in such context. Such tradeoff between data volume reduction and relative prediction accuracy is cost effective.

$$DVR_n = \sum_{i=1}^{n} DVC_i \times \frac{32 - BW_i}{32} \quad (1)$$

4.3 Random Error Tolerance

For most approximate memories, bit flipping is considered as the error model and the error rate varis from 10^{-8} to 10^{-1} [4, 5, 11]. The difference between different accuracy levels is in several orders of magnitude. Therefore, using the order of magnitude of MTRER is sufficient to characterize the tolerance of random errors. In the first part of this section, we assume that random errors with certain probability (from 10^{-8} to 10^{-1}) are uniformly introduced into approximate memory bit cells and all the synaptic weights of CNNs in the benchmark suite are stored in the approximate memory. Their impacts on prediction accuracy are shown in Figure 7. The relative accuracy is acceptable (i.e., greater than 97%) when the probability of random error is less than 10^{-4}. It indicates that it is feasible to use approximate memories with error rates less than 10^{-4} for the storage of synaptic weights in these CNNs. It is realizable for many approximate memory techniques to attain an error rate of bit cells below 10^{-4} [4, 5, 11].

When the injected error probability is larger than MTRER (10^{-4}), several MSBs must be protected to restrict the

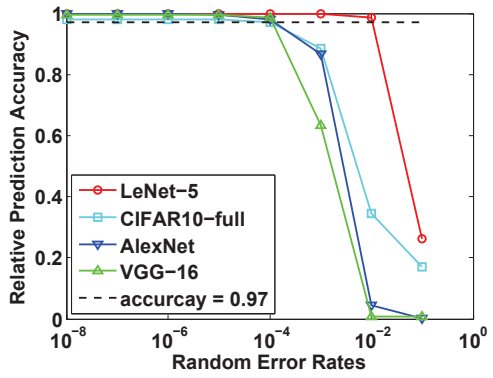

Figure 7: Accuracy with Injected Errors.

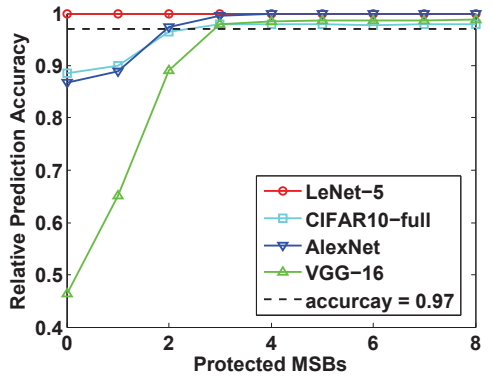

Figure 8: Accuracy with Protected MSBs [3].

magnitude of errors and to alleviate their impacts on prediction accuracy. In the second part of this section, we use fixed error probability (i.e., 10^{-3}) to explore how many MSBs should be protected while others can be stored in approximate bit cells for this error probability. Figure 8 shows the trend of prediction accuracy with various numbers of protected MSBs. It can be seen that the prediction accuracy is acceptable (i.e., greater than 97%) when the protected MSBs are more than 3 bits. Designers need to take measures to protect at least 3 MSBs to produce an acceptable output quality when using an approximate memory with a fixed error rate as 10^{-3}.

5. CONCLUSIONS

In this paper, an analytical framework is proposed for the data resilience evaluation (DRE) of CNNs. We apply the proposed DRE framework to four prevalent CNNs and demonstrate that a high degree of data resilience exists in these networks. For off-chip memory access, bit-width scaling can be used to reduce the amount of data from off-chip memory to on-chip memory. On average, the data volume can be reduced by 80.38% with a 2.69% loss in relative prediction accuracy. For approximate memory with random errors, all the synaptic weights can be stored in approximate part when the error rate is less than 10^{-4}, which is attainable in many approximate memories. When the error rate is fixed at 10^{-3}, 3 MSBs must be protected while other LSBs can be stored in the approximate part. Extensive simulations will be conducted in our subsequent work to explore appropriate approximate memory architectures for implementations in CNN hardware.

[3]The injected random error rate is 10^{-3} in this simulation.

6. ACKNOWLEDGMENTS

The authors would like to acknowledge support from National Natural Science Foundation of China under grant No. 91648116, and the Research Fund from Beijing Innovation Center for Future Chip under grant No. KYJJ2016009.

7. REFERENCES

[1] Srimat Chakradhar, Murugan Sankaradas, Venkata Jakkula, and Srihari Cadambi. A dynamically configurable coprocessor for convolutional neural networks. In *International Symposium on Computer Architecture (ISCA)*. ACM, 2010.

[2] Clément Farabet, Berin Martini, Benoit Corda, Polina Akselrod, Eugenio Culurciello, and Yann LeCun. Neuflow: A runtime reconfigurable dataflow processor for vision. In *CVPR 2011 Workshops*. IEEE, 2011.

[3] Zidong Du, Robert Fasthuber, Tianshi Chen, Paolo Ienne, Ling Li, Tao Luo, Xiaobing Feng, Yunji Chen, and Olivier Temam. Shidiannao: shifting vision processing closer to the sensor. In *International Symposium on Computer Architecture (ISCA)*. ACM, 2015.

[4] Ik Joon Chang, Debabrata Mohapatra, and Kaushik Roy. A voltage-scalable & process variation resilient hybrid sram architecture for mpeg-4 video processors. In *Design Automation Conference (DAC)*. ACM, 2009.

[5] Yuanchang Chen, Xinghua Yang, Fei Qiao, Jie Han, Qi Wei, and Huazhong Yang. A multi-accuracy-level approximate memory architecture based on data significance analysis. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2016.

[6] Song Liu, Karthik Pattabiraman, Thomas Moscibroda, and Benjamin G Zorn. Flikker: saving dram refresh-power through critical data partitioning. *ACM SIGPLAN Notices*, 2012.

[7] Jan Lucas, Mauricio Alvarez-Mesa, Michael Andersch, and Ben Juurlink. Sparkk: Quality-scalable approximate storage in dram. In *The Memory Forum*, 2014.

[8] Fei Qiao, Ni Zhou, Yuanchang Chen, and Huazhong Yang. Approximate computing in chrominance cache for image/video processing. In *International Conference on Multimedia Big Data (BigMM)*. IEEE, 2015.

[9] Ye Tian, Qian Zhang, Ting Wang, Feng Yuan, and Qiang Xu. Approxma: Approximate memory access for dynamic precision scaling. In *Great Lakes Symposium on VLSI (GLSVLSI)*. ACM, 2015.

[10] Jacob Nelson, Adrian Sampson, and Luis Ceze. Dense approximate storage in phase-change memory. ACM, 2011.

[11] Ashish Ranjan, Swagath Venkataramani, Xuanyao Fong, Kaushik Roy, and Anand Raghunathan. Approximate storage for energy efficient spintronic memories. In *Design Automation Conference (DAC)*. IEEE, 2015.

[12] Vinay Chippa, Anand Raghunathan, Kaushik Roy, and Srimat Chakradhar. Dynamic effort scaling: Managing the quality-efficiency tradeoff. In *Design Automation Conference (DAC)*. ACM, 2011.

[13] Woongki Baek and Trishul M Chilimbi. Green: a framework for supporting energy-conscious programming using controlled approximation. In *ACM Sigplan Notices*, volume 45, pages 198–209. ACM, 2010.

[14] Vinay K Chippa, Srimat T Chakradhar, Kaushik Roy, and Anand Raghunathan. Analysis and characterization of inherent application resilience for approximate computing. In *Design Automation Conference (DAC)*. ACM, 2013.

[15] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Fixed point optimization of deep convolutional neural networks for object recognition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015.

[16] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of ACM international conference on Multimedia*. ACM, 2014.

[17] https://github.com/BVLC/caffe.