salmanali@khu.ac.kr, *shbae@khu.ac.kr

Soft Error Adaptable Deep Neural Networks

Muhammad Salman Ali *Sung-Ho Bae
Kyung Hee University *Kyung Hee University

Abstract

The high computational complexity of deep learning algorithms has led to the development of specialized hardware architectures. However, soft errors (bit flip) may occur in these hardware systems due to voltage variation and high energy particles. Many error correction methods have been proposed to counter this problem. In this work, we analyze an error correction mechanism based on repetition codes and an activation function. We test this method by injecting errors into weight filters and define an ideal error rate range in which the proposed method complements the accuracy of the model in the presence of error.

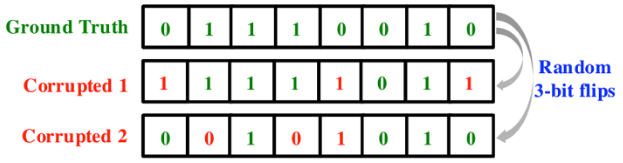
1. Introduction

Deep learning has revolutionized the field of computer vision, with advanced architectures and complex datasets surpassing the human-level performance. However, the computation complexity of these models is very high, which makes them inefficient for real-time scenarios. Therefore, to make them efficient, rigorous efforts have been made at both the hardware and software level.

At the software level, different techniques and architectures have been proposed to tackle this problem. Depth-wise separable convolution was proposed by Howard et al. [1] which decomposes convolutions into spatial and channel convolution, which significantly decreases the computation complexity in a DNN architecture. Deep Compression was proposed by Han et al. [2] which contained Huffman coding, pruning, and quantization to reduce the computational complexity of the model. Although, different methods and techniques have been proposed, it is still considered as an active research problem.

The main bottleneck is at the hardware level, where Complementary Metal Oxide Semiconductor (CMOS) is used as a standard industrial element. The stability shown by CMOS is high. However, it is not power-efficient and has inadequate speed [3]. Researchers have recently started working on specialized hardware accelerators containing thousands of parallel processing engines which can significantly increase deep learning algorithms computational efficiency [4]. However, these specialized hardware may malfunction due to soft errors that may arise in them. Errors are caused by high energy particles striking the electronic devices.

These errors can be catastrophic for deep learning algorithms as they may significantly reduce the accuracy of the model. There have been efforts for error correction of soft errors that may occur in these specialized systems. In [5], an error correction layer (ECL) based on repetition algorithm and a new activation function Piecewise ReLU (PwReLU) was proposed which considerably increases the accuracy of the model in an erroneous environment. They induced the errors in output feature maps after the convolution and observed model performance in the presence of bit errors (soft errors). In this work, we used their approach and injected the errors in filter weights of the convolution and observed the accuracy drop and model performance in the presence of errors with and without ECL.



Ground Truth	0	1	1	1	0	0	1	0
Corrupted 1	1	1	1	1	1	0	1	1
Corrupted 2	0	0	1	0	1	0	1	0

Figure 1: Error injection on 8-bit filter weight data, we randomly flip 3 bits and show different possible output

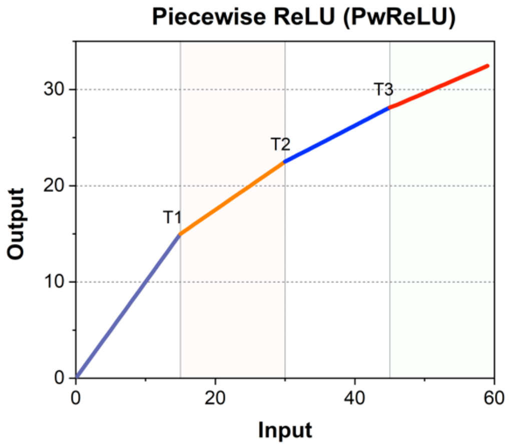


Figure 2: Piecewise ReLU as proposed in [5] with three learnable thresholds. These thresholds are optimized during training.

2. Proposed Method

In this section, we will define the error simulation and error correction method to remove soft errors that occur in specialized hardware systems.

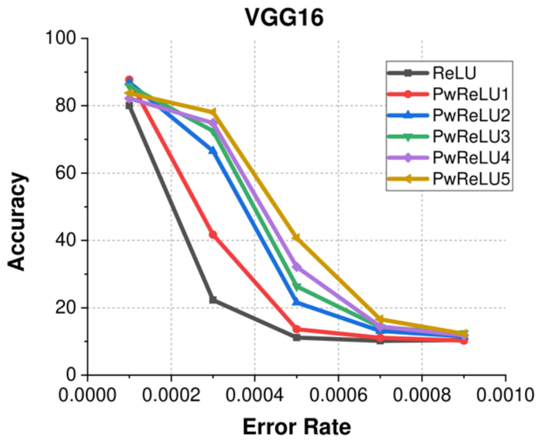


Figure 3: An optimum range of error in which ECL and PwReLU give better accuracy as compared to ReLU

2.1 Error Injection

In this work, we simulate the soft errors that may occur in advanced hardware systems in the form of bit flips. Our error model is in line with the previous literature [6][10]. Error is random and tends to have i.i.d condition [5]. The error can be modelled through Bernoulli distribution as

$$f(k;p) = \begin{cases} q, & \text{if } k = 1 \\ p = 1 - q, & \text{if } k = 0 \end{cases} \qquad (1)$$

where p is the error rate, Figure 1 shows an example of an error occurring in a weight filter value. The weight filter value can be distorted at bit level due to process variation in advanced hardware systems.

2.2 Error Correction Layer (ECL)

ECL is based on repetition codes. Convolution operation at every layer is performed three times, which in turn gives different results due to error. Repetition code algorithm is then applied on these outputs to get the correct result.

Probability of error after applying the repetition codes can be calculated as:

$$P_x(x) = \sum_{i=\lfloor(n/2)\rfloor}^{n} \binom{n}{x} p^x (1-p)^{n-x}, \qquad (2)$$

where $P_x(x)$ represents the total error probability, x represents the total number of errored bits, n denotes the total number of bits and p represents the error rate.

In line with previous work [5], the number of repetitions is set at three as a standard. For the 3-level repetition code, a single input is coded thrice. For example, 1 will be coded as 111. If we receive a code as 110 and the error probability is less than 1/3, then there is a high probability that the correct code was 111 instead of 110. Three-level repetition codes fail if the error occurs in two out of three coded inputs.

	Without ECL			With ECL		
	Error Rate					
	0.1%	0.05%	0.01%	0.1%	0.05%	0.01%
ReLU	9.7	10.4	80.8	10.7	11.1	92.5
PwReLU	11.6	11.6	87.7	11.8	40.71	92.4

Table 1: Performance of ECL and PwReLU on VGG16 with different error rates. Accuracies are given in percentages.

2.3 Piecewise ReLU (PwReLU)

PwReLU was proposed in [5], it is specially designed activation function to suppress errors in error-prone neural networks. Bit errors, when occurred in higher ordered bits, can significantly impact the accuracy of the model. PwReLU was proposed to reduce the impact of high ordered bits. It contains learnable thresholds, after each threshold, the slope of activation function is decreased by predefined hyper-parameter (as shown in Figure 2) leading to a minimized error effect due to bit-flips in high ordered bits. PwReLU is given as:

$$PwReLU_n = \begin{cases} (T_n + x) \times a_n, & x \geq T_n \\ \cdot & \\ \cdot & \\ \cdot & \\ (T_2 + x) \times a_2, & T_2 \leq x \leq T_3 \\ (T_1 + x) \times a_1, & T_1 \leq x \leq T_2 \\ x, & 0 \leq x \leq T_1 \\ 0, & x \leq 0 \end{cases} \qquad (3)$$

where $[T_1, T_2, \ldots\ldots, T_n]$ are thresholds which are optimized during training, n indicates the number of bends, $[a_1, a_2, \ldots\ldots, a_n]$ are predefined hyperparameters which decrease the slope of activation function after each threshold as defined in [5] the value of a is [$a_1 = 0.5,\ a_2 = 0.4,\ a_3 = 0.3,\ a_4 = 0.2,\ a_5 = 0.1$] which remains fixed in all experiments.

3. Experiments and Results

For all the experiments, we used VGG16[8] trained on CIFAR10 [9] dataset. We used the same training details as defined in the original work [5]. The training was performed independent of errors, and errors were only injected during the inference process. Errors were injected into the filter weights similar to [7].

Table 1 shows the comparison of ReLU and PwReLU when an error was injected into the filter weights. At an error rate of 0.1% and 0.5%, without ECL, both ReLU and PwReLU show an accuracy of 10% approximately. However, with ECL, PwReLU gains an accuracy improvement of 30% as compared to ReLU, which is still at 10%. At a very low error rate of 0.01%, without ECL PwReLU shows an accuracy boost of 7.7% over ReLU, with ECL both the models can achieve baseline accuracy.

3.1 Analysis

When the error is injected into filter weights, the error rate may significantly define the behavior of the model. At a very high error rate, the ECL and PwReLU will fail. With significantly low error rate PwReLU may perform significantly better than ReLU without ECL. However, with ECL, both will reach their baseline accuracy as can be seen from results in Table 1. The optimal range between high and low error rates where ECL and PwReLU complement the accuracy of the model in the presence of an error is given in Figure 3. In Figure 3, we also compared different variants of PwReLU with ReLU.

4. Conclusion

In this paper, we have tested the impact of ECL and PwReLU in error correction for advanced hardware systems. We observed an optimal range in which ECL and PwReLU complement the accuracy of the model significantly as compared to ReLU in the presence of error. Generalizability of ECL and PwReLU makes it practical for deployment into various applications of DNN.

Acknowledgment

This research was a result of a study on the "HPC Support" Project, supported by the ‘Ministry of Science and ICT’ and NIPA.

References

[1] François Chollet. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1251- 1258, 2017.

[2] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. arXiv preprint arXiv:1510.00149, 2015.

[3] Miao Hu, Hai Li, Yiran Chen, Qing Wu, Garrett S Rose, and Richard W Linderman. Memristor crossbar-based neuromorphic computing system: A case study. IEEE transactions on neural networks and learning systems, 25(10):1864- 1878, 2014.

[4] Tianshi Chen, Zidong Du, Ninghui Sun, Jia Wang, Chengyong Wu, Yunji Chen, and Olivier Temam. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. ACM SIGARCH Computer Architecture News, 42(1):269- 284, 2014.

[5] M. S. Ali et al., "ERDNN: Error-Resilient Deep Neural Networks with a New Error Correction Layer and Piece-wise Rectified Linear Unit," in IEEE Access, doi: 10.1109/ACCESS.2020.3017211.

[6] Zitao Chen, Guanpeng Li, Karthik Pattabiraman, and Nathan De Bardeleben. Binfi: an efficient fault injector for safety-criticalmachine learning systems. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pages 1– 23, 2019.

[7] Chandramoorthy, Nandhini, et al. "Resilient low voltage accelerators for high energy efficiency." 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2019.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[9] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. online: http://www. cs. toronto. edu/kriz/cifar. html, 55, 2014.

[10] Ali, Muhammad Salman, and Sung-Ho Bae. "A Novel Error-Resilient Memristor-based Neuromorphic Architecture using an Error Correction Layer." 한국정보과학회 학술발표논문집 (2019): 978-980.