2020 International Conference on Computing and Information Technology, University of Tabuk, Kingdom of Saudi Arabia.
Volume: 01, Issue: ICCIT- 1441, Page No.: 368 – 373, 9th & 10th Sep. 2020.

Analyzing the Reliability of Convolutional Neural Networks on GPUs: GoogLeNet as a Case Study

Younis Ibrahim
College of IoT Engineering
Hohai University
Changzhou, China
Younis@hhu.edu.cn

Haibin Wang
College of IoT Engineering
Hohai University
Changzhou, China
wanghaibin@hhuc.edu.cn

Khalid Adam
Faculty of Electrical & Electronic Eng.
University Malaysia Pahang
Pahang, Malaysia
khalidwsn15@gmail.com

***Abstract*— Convolutional Neural Networks (CNNs) are used for tasks such as object recognition. Once a CNN model is used in a radiative environment, reliability of the system against soft errors is a crucial issue, especially in safety-critical and high-performance applications that bound with real-time response. Selectively-hardening techniques do improve the reliability of these systems. However, the hard question in selective techniques is "how to exclusively select code portions to harden, to safe the performance from being degraded". In this paper, we propose a comprehensive analysis methodology for CNN-based classification models to confidently determine the only vulnerable parts of the source code. To achieve this, we propose a technique, Layer Vulnerability Factor (LVF) and adopt another technique, Kernel Vulnerability Factor (KVF). We apply these techniques to GoogLeNet, which is a famous image classification model, to validate our methodology. We precisely identify the parts of the GoogLeNet model that need to be hardened instead of using expensive duplication solutions.**

Keywords*— *convolutional neural networks, GoogLeNet, reliability, soft errors, GPUs

I. INTRODUCTION

Convolutional Neural Networks (CNNs are the new revolution that dominating the Artificial Intelligence (AI), and they are the most efficient technique to perform computer vison tasks, such as image classification [1] and object detection [2]. Due to the computational requirements, CNNs are often run on heterogeneous systems that composed of CPUs and accelerators, such as Graphics Processing Unit (GPUs). In fact, GPUs nowadays are the dominated hardware to accelerate CNN models [3].

CNN models are widely used in safety-critical systems, such as self-driving cars [4] and space applications [5]. Therefore, analyzing the reliability of such systems is critical. One way to address reliability issues is utilizing software redundancy. A bunch of techniques have been proposed for soft errors mitigation in GPUs based on software solutions, including Double Modular Redundancy (DMR), Triple Modular Redundancy (TMR), and Algorithm-Based Fault Tolerance (ABFT). However, the major challenging of using these techniques is the runtime overheads that associated with these solutions.

In this study, "which portion of the source code to harden" is the question we try to answer, and implement it on GoogLeNet algorithm as a case study. To answer this question, we propose a systematic analysis methodology for classification models to identify code portions that worth protecting. As a first objective, we propose a technique, Layer Vulnerability Factor (LVF) and adopt an exist technique, Kernel Vulnerability Factor (KVF). We implement these techniques on a CNN model that is used in image classification tasks, GoogLeNet. Using our methodology, we are able identify different vulnerable parts of the GoogLeNet model that need to be hardened.

The main contributions of this work are: (1) a methodology to evaluate the likelihood of faults in specific parts of the source code that likely to cause errors at the output; (2) the LVF concept and implementing it to a realistic case-study; and (3) an extensive analysis of GoogLeNet characteristics under SASSIFI fault injection.

The remainder of the paper is organized as follows. Section II shows the background and reviews related work. Section III presents the proposed methodology. Section IV analyzes and discusses the results. Section V concludes the paper.

II. BACKGROUD AND RELATED WORK

In this section, we present a brief background on Convolutional neural networks and especially GoogLeNet. We then, summarize previous findings on CNNs reliability.

a. Convolutional Neural Networks

Due to their outstanding performance that bypassed even the human ability in object recognition benchmarks (i.e., classification), CNNs are arguably the most popular type of the Deep Learning architectures [6]. The convolutional operation is the key component in CNNs. They use filters to extract features of the image, by sliding a filter over the input image, multiplying and accumulating products at every position of the input (i.e., receptive field) with this filter [7]. The well-known CNN architectures include AlexNet, VGGNet, GoogLeNet, ResNet, and DenseNet.

b. GoogLeNet

GoogLeNet developed by Google, is the winner of the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) in 2014. It is the first CNN architecture replaced the expensive Fully-connected layers at the end of the model with a simple global average-pooling layer, which averages out the given values of each feature map. This change has dramatically reduced the number of parameters used in the model, which made it a faster in the training phase, lighter in size, and higher in performance, compared to its predecessor architectures, such VGGNet and AlexNet [8]. For these reasons, GoogLeNet has been widely adopted in many applications, including self-driving cars [9].
GoogLeNet handles our input data (i.e., images) in a fixed stack of operations to give the desired predictions. The image begins from the first layer (input layer) passes across

978-1-7281-2680-7/20/$31.00 ©2020 IEEE