FPGA implementation and validation of image classification model protection method

2310054 Naofumi Komata

In recent years, Deep Neural Networks (DNNs) and their applications in society have been increasing. DNN models are generated by training datasets and consist of a “layered architecture” for performing inference from input data and generating output data, as well as weight and bias parameters contained in each layer. In particular, training a high-performance DNN model requires a huge amount of data sets, computational resources, and expertise, and a trained DNN model is considered intellectual property. However, there is a possibility that DNN models can be illegally obtained through unauthorized access such as side-channel attacks targeting edge devices such as embedded computers and FPGAs.

In order to prevent further unauthorized use and distribution of DNN models that have been obtained through such channels, methods have been proposed to protect DNN models on edge devices, and the NN-Lock algorithm[1] proposed by Manaar Alam et al. The AES algorithm encrypts all the coefficient weight parameters of the DNN model, which indicate the strength of the connections between nodes, and temporarily decrypts them by using the correct encryption key by the correct user at the time of inference. On the other hand, if an unauthorized user, such as a thief, uses the wrong encryption key, changes in the values of the encrypted parameters are propagated to all subsequent layers, and the final inference result is different from the model before encryption. This makes it difficult for an attacker without the correct decryption key to steal and use the DNN model because the accuracy of the DNN model inference is greatly reduced due to the encrypted weight parameters, thus preventing unauthorized use and distribution of the DNN model.

In recent years, field-programmable gate arrays (FPGAs) are increasingly being used as processors in advanced driver assistance systems for automobiles to perform inference processing using on-board DNN models. In this study, I apply the NN-Lock algorithm to a hardware DNN model on an FPGA for the first time. Specifically, I propose two methods to construct a DNN model whose weight parameters are already encrypted by the NN-Lock algorithm and a decryption mechanism for the NN-Lock algorithm on FPGA.

The first is “just before writing the weight parameters encrypted by the NN-Lock algorithm to the memory BRAM on FPGA (NN-Lock: Before writing to BRAM)” and the second is “just before executing DNN inference

(NN-Lock: Before DNN execution).” I implemented and evaluated the logic patterns of the two aforementioned application points of NN-Lock modules and the number of parallel processes of the DNN module and the NN-Lock module. In addition, I also propose a method of implementing a stand-alone DNN inference computation module suitable for FPGA implementation of NN-Lock.

Next, based on the circuit size and inference processing time obtained from the implementation and evaluation of the proposed method, I consider the requirements for circuit size and inference processing time with reference to the performance of FPGA boards and camera modules that are actually available in the market. I also discuss the resistance to theft of weight parameters at the point where NN-Lock is applied, by discussing several typical attack methods targeting FPGAs. In light of the above, this paper provides information necessary for applying NN-Lock to DNN models running on FPGAs for a wide range of hardware under the assumption that they will be used in driver assistance systems.

References

[1] Manaar Alam, Sayandeep Saha, Debdeep Mukhopadhyay, and Sandip Kundu. NN-Lock: A Lightweight Authorization to Prevent IP Threats of Deep Learning Models, ACM Journal on Emerging Technologies in Computing Systems, Volume 18, Issue 3, Article 51 (2022)