

# Contents

|                                                                    |          |
|--------------------------------------------------------------------|----------|
| List of Figures                                                    | ix       |
| List of Tables                                                     | xi       |
| Abbreviations and Acronyms                                         | xiii     |
| Abstract                                                           | xvii     |
| <i>Resumen</i>                                                     | xviii    |
| <i>Resum</i>                                                       | xix      |
| <b>1 Introduction</b>                                              | <b>1</b> |
| 1.0.1 Neural Networks . . . . .                                    | 2        |
| 1.0.2 Faults in Neural Networks . . . . .                          | 3        |
| 1.0.3 Compute Platforms: GPUs and FPGAs . . . . .                  | 3        |
| 1.1 Objectives and Motivation of the Thesis . . . . .              | 5        |
| 1.2 Main Contributions of the Thesis . . . . .                     | 6        |
| 1.3 Thesis Outline . . . . .                                       | 7        |
| <b>2 Background and Related Work</b>                               | <b>9</b> |
| 2.1 Background . . . . .                                           | 10       |
| 2.1.1 History of NNs . . . . .                                     | 10       |
| 2.1.2 Types of NNs . . . . .                                       | 11       |
| 2.1.3 Structure and Functioning . . . . .                          | 12       |
| 2.1.4 Training and Inference . . . . .                             | 14       |
| 2.1.5 Convolutional Networks . . . . .                             | 16       |
| 2.1.6 Hardware Faults Affecting NNs . . . . .                      | 17       |
| 2.1.7 Security Attacks Exploiting Fault-Injection in NNs . . . . . | 18       |
| 2.2 Related Work in Fault-Tolerant NNs . . . . .                   | 19       |
| 2.2.1 Impact of Faults on NNs . . . . .                            | 19       |
| 2.2.2 Fault-Tolerance Mechanisms . . . . .                         | 20       |
| 2.2.2.1 Hardware Mechanisms . . . . .                              | 20       |
| 2.2.2.2 Software Mechanisms . . . . .                              | 22       |
| 2.2.2.3 Algorithmic/Model-Level Solutions . . . . .                | 23       |
| 2.2.2.3.1 Positioning of Our Approach . . . . .                    | 24       |

|           |                                                                   |           |
|-----------|-------------------------------------------------------------------|-----------|
| <b>3</b>  | <b>Bit-level Invariability Analysis</b>                           | <b>27</b> |
| 3.1       | Neural Network Models to be Analysed . . . . .                    | 28        |
| 3.2       | Methodology . . . . .                                             | 31        |
| 3.3       | Bit-Grouping Strategies . . . . .                                 | 33        |
| 3.4       | Bit-Level Invariability . . . . .                                 | 35        |
| 3.4.1     | Adapting Strategies to Neural Network Models . . . . .            | 36        |
| 3.4.2     | Bit Invariability Evolution Layer-by-Layer across NN models . . . | 38        |
| 3.4.3     | General Results for FP32 Data Type . . . . .                      | 40        |
| 3.4.3.1   | ViT Model . . . . .                                               | 43        |
| 3.4.4     | General Results for INT8 Data Type . . . . .                      | 45        |
| 3.4.4.1   | Conclusions . . . . .                                             | 46        |
| <b>4</b>  | <b>Fault Protection Methods</b>                                   | <b>47</b> |
| 4.1       | Fixed Protection (FP) Mechanism . . . . .                         | 49        |
| 4.2       | Variable Protection (VP) Mechanism . . . . .                      | 51        |
| 4.3       | FP and VP Coverage and Overheads . . . . .                        | 52        |
| 4.4       | Evaluation . . . . .                                              | 57        |
| 4.4.1     | Evaluation Methodology . . . . .                                  | 57        |
| 4.4.1.1   | Fault Model . . . . .                                             | 58        |
| 4.4.2     | Experimental Setup . . . . .                                      | 60        |
| 4.4.3     | Results . . . . .                                                 | 60        |
| 4.4.4     | Fault Classification . . . . .                                    | 60        |
| 4.4.4.1   | Bit Flips Effect on Unprotected Models . . . . .                  | 63        |
| 4.4.4.2   | Robustness of VP/FP Mechanisms . . . . .                          | 64        |
| 4.4.4.3   | Silent Data Corruption Effect . . . . .                           | 65        |
| 4.4.4.4   | Faults Evolution per Convolutional Layer . . . . .                | 67        |
| 4.4.4.5   | Bit invariability vs Model Data Precision Type . . . . .          | 71        |
| 4.4.5     | Comparison with Related Works . . . . .                           | 73        |
| 4.4.5.1   | New Methodology and Setup . . . . .                               | 75        |
| 4.4.5.2   | Results . . . . .                                                 | 77        |
| 4.4.6     | Impact of Faults Outside Convolutional Filter Weights in CNNs . . | 81        |
| 4.5       | Summary . . . . .                                                 | 84        |
| <b>5</b>  | <b>FPGA and GPU Implementations</b>                               | <b>89</b> |
| 5.1       | FPGA-based Accelerator . . . . .                                  | 90        |
| 5.1.1     | Background and Implementation Procedure . . . . .                 | 90        |
| 5.2       | Integration and Overheads . . . . .                               | 91        |
| 5.2.1     | Accelerator Architecture Overview . . . . .                       | 92        |
| 5.2.2     | Incorporation of FP and VP Mechanisms . . . . .                   | 93        |
| 5.2.3     | Impact on Execution Time . . . . .                                | 95        |
| 5.2.4     | FPGA Resource Overheads . . . . .                                 | 96        |
| 5.3       | GPU Integration . . . . .                                         | 97        |
| 5.3.1     | Background and Implementation Procedure . . . . .                 | 97        |
| 5.3.2     | CUTLASS Integration . . . . .                                     | 99        |
| 5.3.2.1   | CUTLASS Modifications . . . . .                                   | 100       |
| 5.3.2.1.1 | C++/CUDA Back End . . . . .                                       | 100       |

---

|           |                                                |            |
|-----------|------------------------------------------------|------------|
| 5.3.2.1.2 | Mask Life Cycle and Execution Policy . . . . . | 102        |
| 5.3.2.1.3 | Python Front End . . . . .                     | 102        |
| 5.3.2.2   | PyTorch Modifications . . . . .                | 103        |
| 5.3.2.3   | User View . . . . .                            | 104        |
| 5.3.2.4   | Performance Results . . . . .                  | 104        |
| 5.4       | Summary . . . . .                              | 107        |
| <b>6</b>  | <b>Conclusions</b>                             | <b>109</b> |
| 6.1       | Contributions . . . . .                        | 110        |
| 6.2       | Future Directions . . . . .                    | 112        |
| 6.3       | Publications . . . . .                         | 113        |
|           | <b>References</b>                              | <b>115</b> |