Optimizing Neural Network Hardware Latency and Energy Use with Genetic Algorithms

Regina Deri
rderi25@amherst.edu
Amherst College
Amherst, Massachusetts, USA

ACM Reference Format:
Regina Deri. 2024. Optimizing Neural Network Hardware Latency and Energy Use with Genetic Algorithms. In . ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION & MOTIVATION

Genetic algorithms can be great tools in finding and optimizing solutions to complex problems, such as those involving design. Previously, they have been successfully used to evolve circuits, antennas, or even hardware modules for quantum computing (for some examples, see Koza et al., 2005).

Neural network design space exploration also seems to be an area where genetic programming could be fruitfully applied. When developing neural networks, it's often difficult to consider the tradeoffs between latency, energy use, and accuracy. In this project, I will create a genetic algorithm framework that will optimize the hardware latency and energy use of neural networks. In addition, I aim to analyze the potentials of GA based neural network optimizations by experimenting with both completely new and prior models.

2 BACKGROUND & RELATED WORK

2.1 Key definitions

Neural networks: Neural networks is one of the most commonly used machine learning models, inspired by the functioning of the human brain. They consist of layers of interconnected neurons that process and transform data. Neural networks have many applications, such as image recognition, natural language processing, and autonomous vehicles.

Genetic algorithms (GAs): Genetic algorithms are a type of optimization algorithm inspired by the process of natural selection. They work by evolving a population of steadily improving solutions over multiple generations, using an array of operators such as mutation and crossover. Genetic algorithms have been successfully used in a variety of domains, such as circuit design, antenna design, and robotics.

Design Space Exploration (DSE) Design space exploration is the process of exploring the space of possible designs for a given problem, in order to find the best design according to some metric. Since the process is automatized, it often has the power to reveal insights into design that otherwise would not have been found.

2.2 Related work

Domashova et al. 2021: Neural networks is one of the most commonly used machine learning models, inspired by the functioning of the human brain. The paper attempts to find optimal architecture for neural networks using genetic algorithms. While the paper doesn't focus on hardware latency and energy use, it provides a good case study for this project.

Mei et al. 2022: This paper introduces ZigZag, a tool that will be used as an essential part of this project to measure the hardware latency and energy use of candidate solutions.

Chiroma et al. 2017: This paper is a review of similar attempts at optimization neural networks with genetic algorithms. It gives a thorough overview of the field, and the referenced papers could prove very useful for my work.

Eiben et al. 2015: This book is essentialy a textbook on genetic algorithms. It will be useful when I get to the point of fine-tuning aspects of my genetic algorithm framework, such as survivor selection functions.

3 METHODS & TOOLS

The project is implementation-heavy, and will involve the creation of a genetic algorithm system that optimizes PyTorch sequential models for hardware latency and energy use. Subsequently, the system will be evaluated on Amherst College's High Performance Cluster (HPC), on a diverse selection of existing models and neural networks generated from scratch by the genetic algorithm framework itself.

One limitation of my methods is that the measured fitness scores of models will be specific to the hardware/accelerator designs used. I am still looking for a way to generalize these results, possibly by using multiple accelerator designs.

Since much of the project involves implementation, there's only a small number of tools that I will use:

- *Python*: The majority of the project's code will be implemented in Python.
- *PyTorch*: PyTorch is a popular open-source machine learning library for Python. It was created by Meta's AI Research lab

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

and is now governed by the PyTorch foundation. It's one of the most important tools for this project, and will be the tool of choice in creating, training, and evaluating neural networks.
- *ZigZag*: ZigZag is a tool that helps estimate the hardware latency and energy use of neural networks running on a given accelrator design. It also provides a heuristic mapper to map neural networks to accelerator designs, however, I will mainly use it for the former purpose. Zigzag was first proposed in a paper by Mei et al. (2021), and has since been a subject of nine other papers. It requires an AI model and an accelerator design as input, and is able to output the estimated latency and energy use of the model running on the accelerator. It will be called from the genetic algorithm framework to evaluate the fitness score of the neural networks.

4 PRELIMINARY RESULTS OR ANALYSIS

So far, I have implemented most of my genetic algorithm framework, including genomes, fitness measurement, crossover, mutation, and selection. There is still some work left to connect these modules, after which the framework will be ready for testing.

5 OUTLINE OF FINAL PAPER

(1) Introduction
(2) Background and Related Work
(3) Description of the genetic algorithm framework
(4) Case study 1: optimizing model created from scratch
(5) Case study 2: optimizing existing model
(6) Discussion
(7) Conclusion

6 UP-TO-DATE TIMELINE FOR COMPLETION

(1) **Week of March 24**:
   - Finish GA framework
   - Begin comprehensive code testing
(2) **Week of March 31**:
   - Plug in to ZigZag
   - Prepare HPC scripts, run preliminary test
   - Find a way to organize the massive amount of generated results
(3) **Week of April 7**:
   - Optimize if needed
   - Begin the measurements that will be in the final paper
   - Get started with writing the paper
(4) **Week of April 14**:
   - Start analyzing results
   - The time to redo measurements if something goes wrong
(5) **Week of April 21**:
   - Potentially begin experimenting with different GA methods (e.g., in term of survivor selection)
   - Continue writing paper
(6) **Week of April 28**:
   - Most of the code and results should be ready by this week
   - The week will be spent finalizing the first draft
(7) **Week of May 5**:
   - Finish first draft