Introduction

With the increasing popularity of machine learning algorithms, many modern applications aim to utilize them in order to gain insight into their user base. However, these algorithms tend to be computationally complex, which yields negative implications on energy efficiency and overall runtime. By recognizing patterns and categorizing data sets, we can pinpoint bottlenecks within the algorithms to consider for speedup and therefore improve overall efficiency.

Problem Statement

We aim to profile a key machine learning algorithm for acceleration which will aid increasing throughput of similar algorithms used in mobile technology.

Case Study Algorithm

Multilayer perceptron is a neural network algorithm that allows machines to make decisions or conclusions from intricate data by training them with predetermined outputs.

Our Approach

The algorithm was profiled using KCachegrind to find the most expensive and time-intensive functions. We analyzed these functions and theorized possible ways to speed them up.

Profiling Results

matrix_multiplication: multiplication of first and second matrix is stored in third matrix Runtime: O(N^3)

add_matrix: performs summation of the coordinate values from the first and second matrix and stores at a coordinate in the third matrix Runtime: O(N^2)

scalar_multiplication: multiplies each element in the given matrix with a given scalar value Runtime: O(N^2)

Solution: Parallelization

One solution to speedup bottlenecks is to make use of parallelization, or running pieces of code concurrently. Ideally, sequential parts could be run on the CPU and the parallel parts on a parallelizable device. Amdahl’s Law is a model to theoretically measure the magnitude of speedup through parallelization.

Parallel part is divided up by N workers

Application: OpenCL

OpenCL is an API that we can use to implement parallelization. It allows users to take parallelizable code and run it on the system’s GPU through API calls and specialized overhead code.

Future Works

Though our work was for theoretical speedups, this study may expanded by implementing parallelization of the code. The following specialized devices can be used for speedup through parallelization:

- **GPUs**: Hardware based graphical processing units that can perform arithmetic intensive operations in parallel. Significant improvement from CPUs, which only run in sequential.
- **FPGAs**: Hardware based parallel processing device that increases the number of instructions per clock cycle, the time it takes to run a program. Like GPUs, FPGAs execute and distribute data across all the units in parallel.

Conclusion

After profiling the multilayer perceptron to locate the bottlenecks, these functions can be sped up through parallelization using GPUs or FPGAs. Such optimization of machine learning algorithms such as multilayer perceptron can aid in meeting demands of current needs in fields such as mobile technology.

References


Acknowledgments

We would like to acknowledge Christine Alvarado, Shelby Thomas, Michael Barrow, and Michael Taylor for assisting and guiding us throughout the duration of our research. This material is based upon work supported by the National Science Foundation under Grant No. CNS-1339335.