Multiple Constraints and Non-regular Solution in Deep Declarative Network

Suikei Wang

A thesis submitted for the degree of
Master of Machine Learning and Computer Vision
The Australian National University

August 2020

Draft Copy – 25 August 2020

Except where otherwise indicated, this thesis is my own original work.

Suikei Wang
25 August 2020

to my parents, yyy (yyy is the people you want to dedicated this thesis to.)

Acknowledgments

Who do you want to thank?

Abstract

Put your abstract here.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Deep learning models composed with multiple parametrized processing layers can learn different levels of features and representations of data through the directed graph structure.

Put your introduction here. You could use \fix{ABCDEFG.} to leave your comments, see the box at the left side.

You have to rewrite your thesis!!!

1.2 Thesis Outline

How many chapters you have? You may have Chapter 2, Chapter **??**, Chapter **??**, Chapter 6, and Chapter 7.

1.3 Contribution

Part I

Deep Declarative Network: Multiple Constrained Declarative Nodes

Chapter 2

An Overview of Optimization Methods

At the begging of each chapter, please introduce the motivation and high-level picture of the chapter. You also have to introduce sections in the chapter.

Section 2.1 yyyy.

2.1 Numerical Optimization Methods

2.1.1 Gradient Descent

You may reference other papers. For example: Generational garbage collection [Lieberman and Hewitt, 1983; Moon, 1984; Ungar, 1984] is perhaps the single most important advance in garbage collection since the first collectors were developed in the early 1960s. (doi: "doi" should just be the doi part, not the full URL, and it will be made to link to dx.doi.org and resolve. shortname: gives an optional short name for a conference like PLDI '08.)

2.1.2 Newton and Quasi-Newton Methods

2.1.3 Conjugate Gradients

2.1.4 Lagrange Multipliers

2.2 Bi-level Optimization

2.2.1 Linear Bilevel Programming

2.2.2 Nonlinear Bilevel Programming

2.3 Constrained Optimization

2.3.1 Global Optimization

2.3.2 Local Optimization

2.4 Summary

Summary what you discussed in this chapter, and mention the story in next chapter. Readers should roughly understand what your thesis takes about by only reading words at the beginning and the end (Summary) of each chapter.

Chapter 3

Deep Declarative Network

Same as the last chapter, introduce the motivation and the high-level picture to readers, and introduce the sections in this chapter.

3.1 Notation

3.2 An Overview of Deep Declarative Network

3.2.1 Structure

3.2.2 Declarative Nodes

3.3 Learning

3.4 Back-propagation Through Declarative Nodes

3.4.1 Unconstrained

3.4.2 Equality Constrained

3.4.3 Inequatlity Constrained

3.5 Experiments

3.5.1 Platform

3.5.2 Implementation Details

3.5.3 Experimental Results

3.5.4 Analysis: Is the Gradient Always Converged?

3.6 Summary

Same as the last chapter, summary what you discussed in this chapter and be the bridge to next chapter. s

Chapter 4

The Future of Declarative Nodes

Same as the last chapter, introduce the motivation and the high-level picture to readers, and introduce the sections in this chapter.

4.1 Sub-classes of declarative nodes

4.1.1 Unconstrained

4.1.2 Equality constrained

4.1.3 Inequality constrained

4.2 Back-propagation through declarative nodes

4.3 Summary

Same as the last chapter, summary what you discussed in this chapter and be the bridge to next chapter.

Part II

Deep Declarative Network: Non-regular Solution

Chapter 5

Solution of Deep Declarative Nodes

5.1 Software platform

5.2 Hardware platform

Table 5.1 shows how to include tables and Figure 5.1 shows how to include codes.

Architecture	Pentium 4	Atom D510	i7-2600
Model	P4D 820	Atom D510	Core i7-2600
Technology	90nm	45nm	32nm
Clock	2.8GHz	1.66GHz	3.4GHz
Cores × SMT	2 × 2	2 × 2	4 × 2
L2 Cache	1MB × 2	512KB × 2	256KB × 4
L3 Cache	none	none	8MB
Memory	1GB DDR2-400	2GB DDR2-800	4GB DDR3-1066

Table 5.1: Processors used in our evaluation.

```c
int main(void)
{
  printf("Hello World\n");
  return 0;
}
```

(a)

```java
void main(String[] args)
{
  System.out.println("Hello World");
}
```

(b)

Figure 5.1: Hello world in Java and C.

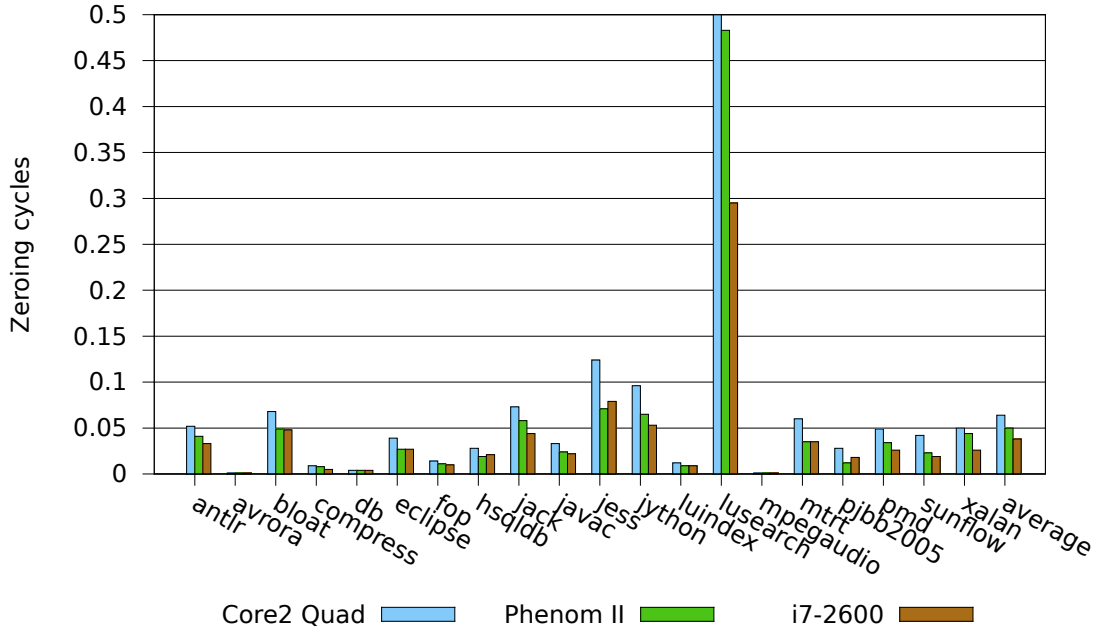
Chapter 6

Results

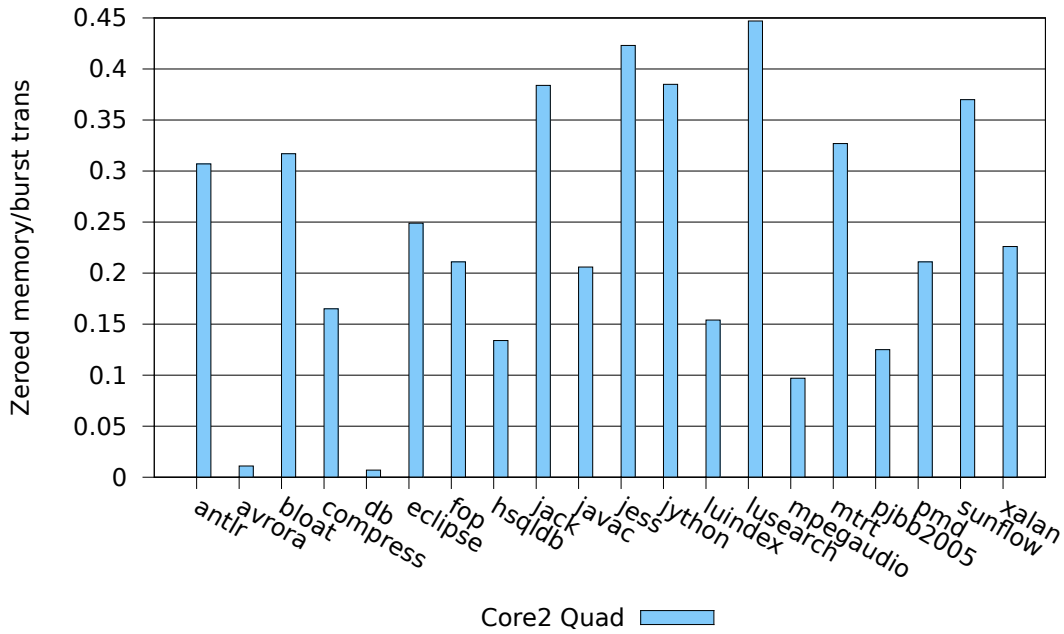
6.1 Direct Cost

Here is the example to show how to include a figure. Figure 6.1 includes two subfigures (Figure 6.1(a), and Figure 6.1(b));

6.2 Summary



(a) Fraction of cycles spent on zeroing



(b) BytesZeroed / BytesBurstTransactionsTransferred

Figure 6.1: The cost of zero initialization

Chapter 7

Conclusion

Summary your thesis and discuss what you are going to do in the future in Section 7.1.

7.1 Future Work

Good luck.

Bibliography

Lieberman, H. and Hewitt, C., 1983. A real-time garbage collector based on the lifetimes of objects. *Communications of the ACM*, 26, 6 (Jun. 1983), 419–429. doi:10.1145/358141.358147. (cited on page 5)

Moon, D. A., 1984. Garbage collection in a large LISP system. In LFP '84: *Proceedings of the 1984 ACM Symposium on LISP and Functional Programming* (Austin, Texas, USA, Aug. 1984), 235–246. ACM, New York, New York, USA. doi:10.1145/800055.802040. (cited on page 5)

Ungar, D., 1984. Generation scavenging: A non-disruptive high performance storage reclamation algorithm. In SDE 1: *Proceedings of the 1st ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments* (Pittsburgh, Pennsylvania, USA, Apr. 1984), 157–167. ACM, New York, New York, USA. doi:10.1145/800020.808261. (cited on page 5)