

Model-based Reinforcement Learning in Computer Systems

Sean J. Parker
Clare Hall



*A dissertation submitted to the University of Cambridge
in partial fulfilment of the requirements for the degree of
Master of Philosophy in Advanced Computer Science*

University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
UNITED KINGDOM

Email: sjp240@cam.ac.uk

March 27, 2021

Declaration

I, Sean J. Parker of Clare Hall, being a candidate for the M.Phil in Advanced Computer Science, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Total word count: 0

Signed:

Date:

This dissertation is copyright ©2021 Sean J. Parker.

All trademarks used in this dissertation are hereby acknowledged.

Acknowledgements

Abstract

Write a summary of the whole thing. Make sure it fits in one page.

Contents

1	Introduction	vii
2	Background and Related Work	1
2.1	Introduction to Deep Learning Models	1
2.1.1	Approaches to optimising deep learning models	1
2.2	Reinforcement Learning	2
2.2.1	Model-Free	2
2.2.2	Model-Based	2
2.2.3	Comparison	2
2.3	Graph Neural Networks	2
3	Title	3
4	Title	5
5	Title	7

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 2

Background and Related Work

In this chapter, we detail the background and prior research that underpins the work described in later chapters. Further, we include an explanation of the theoretical concepts, both reinforcement learning and graph neural networks, which we have used extensively in this work.

2.1 Introduction to Deep Learning Models

This section discusses the way in which machine learning models are represented for efficient execution on physical hardware devices. First, we discuss dataflow graphs and the mapping of tensor operations to the graphs and finally recent methods to optimise the graphs prior to execution.

2.1.1 Approaches to optimising deep learning models

Over the past decade, there has been a rapid development of various deep learning architectures that aim to solve a specific task. Common examples include convolutional networks (popularised by AlexNet then ResNets, etc), transformer networks that have seen use in the modelling and generation of

language. Recurrent networks that have shown to excel at learning long and short trends in data.

Importantly, the fundamental building blocks of the networks have largely remained unchanged. As the networks become more complex, it becomes untenable to manually optimise the networks to reduce the execution time on hardware. Therefore, there is extensive work in ways to both automatically optimise the models, or, alternatively apply a set of hand-crafted optimisations.

Tensorflow, a common machine learning framework is designed to greedily apply a set of pre-defined substitutions to an input graph in an attempt to optimise the graph. Tensorflow made use of low-level libraries such as cuBLAS for optimised matrix operations and cuDNN for convolutional kernels. Furthermore, Tensorflow also contains a set of 155 substitutions that are implemented in 53,000 lines of code; to complicate matters, new operators are continuously proposed, such as grouped or transposed convolutions, which leads to a large amount of engineering effort required to maintain the library.

2.2 Reinforcement Learning

2.2.1 Model-Free

2.2.2 Model-Based

2.2.3 Comparison

2.3 Graph Neural Networks

Chapter 3

Title

Chapter 4

Title

Chapter 5

Title