Reykjavík University Project Report, Thesis, and Dissertation Template

by

Sigurður Helgason

Thesis of 60 ECTS credits submitted to the School of Computer Science
at Reykjavík University in partial fulfillment
of the requirements for the degree of
Master of Science (M.Sc.) in Computer Science

December 2019

Examining Committee:

Yngvi Björnsson, Supervisor
Professor, Reykjavík University, Iceland

Tough E. Questions, Examiner
Associate Professor, Massachusetts Institute of Technology, USA

Reykjavík University Project Report, Thesis, and Dissertation Template

Sigurður Helgason

December 2019

Abstract

this is my abstract written in the language of English I am testing thought

alas I knew

ok so now I'm trying something new

asdf

test

Herkænsku mat á djúptauganetum

Sigurður Helgason

desember 2019

Útdráttur

this is my abstract written in the language of íslensku þæöasdf

Important!!! Read the Instructions!!!

If you have not already done so, LaTeX the `instructions.tex` to learn how to setup your document and use some of the features. You can see a (somewhat recent) rendered PDF of the instructions included in this folder at `instructions-publish.pdf`. There is also more information on working with LaTeX at `http://samvinna.ru.is/project/htgaru/how-to-get-around-projects-publish.pdf`. This includes common problems and fixes.

This page will disappear in anything other than draft mode.

I dedicate this thesis to my family who while not understanding most of what I do always support me. The friends that still want to talk to me even though my schedule has rendered me unmeetupwithable. Lastly but certainly not least, Hulda Lilja who always easead my mind during my spurts of imposter syndrome, and fear of failure. Don't Panic.

Acknowledgements

So long, and thanks for all the fish.

Acknowledgements are optional; comment this chapter out if they are absent Note that it is important to acknowledge any funding that helped in the workThis work was funded by 2020 RANNIS grant "Survey of man-eating Minke whales" 1415550. Additional equipment was generously donated by the Icelandic Tourism Board.

Preface

This dissertation is original work by the author, Sigurður Helgason.

The preface is an optional element explaining a little who performed what work. See `https://www.grad.ubc.ca/sites/default/files/materials/thesis_sample_prefaces.pdf` for suggestions.

List of publications as part of the preface is optional unless elements of the work have already been published. It should be a comprehensive list of all publications in which material in the thesis has appeared, preferably with references to sections as appropriate. This is also a good place to state contribution of student and contribution of others to the work represented in the thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

MSc	Masters of Science
ML	Machine Learning
AI	Artificial Intelligence

This part of the dissertation introduces the concepts and the field.

Chapter 1

Introduction

State the objectives of the exercise. Ask yourself: Why did I design/create the item? What did I aim to achieve? What is the problem I am trying to solve? How is my solution interesting or novel?

In the world of deep learning is exciting, we have models that are able to examine images and very reliably be able to identify it's content. Deep learning models have beaten Opthalmologists in identifing diabetic retinopathy, they've identified cancer cells where others have not. Deep learning models have even predicted the likelyhood a person will die in the following year with good accuracy (CITE). These models have done these things extremely accurately, cheap, and fast.

The field of examining deep learning models to gain a richer understanding in how they work, and what makes them so much better than humans at various tasks is still new. This is the field of Explainable Artificial Intelligence (XAI). If we wish to continue using artificial intelligence (AI) and machine learning (ML), to improve our lives we must examine how they work, this is not only to improve ourselves but these explanations are a legal requirement now in many areas, namely, legal, and medical. Recently, XAI has generally been focused on Model Interpretability, in particular focus on gaining insights on the concepts learned by Deep Neural Networks.

In this we thesis takes a look at how we can examine an artificial neural network (ANN) to understand which higher level concepts (HLC) it deems important for a given state within games. These games can be simple like Tic-Tac-Toe or Breakthrough and

extremely complicated like Chess or Go. Some of the research questions are what if an ANN values a HLC greatly which we as humans don't find particularily important, and yet the ANN is able to defeat us what does that mean?

The method of examining HLC's within games has generally been done by examining the current state of the game by evaluating them with respect to a heuristic. A heuristic within games are a evaluation of the end cost for a state given just the state, for example, the amount of pieces left within a game of chess. Intuitively, the amount of pieces left is a good estimate for a state in chess, this is a higher level concept we use to evaluate a chess position. The piece amount can be considered as a lower-level concept than other concepts we use. For instance, grand master chess players evaluate a position w.r.t. states where the king is safe from attack, or the structure of the pawn positions. The idea is to examine a neural networks evaluation of a state regarding those higher-level concepts, if human players evaluate the king safety of a state low but the neural network highly values it, and the neural network plays better than the player. There could be an avenue for us to improve our game by considering king safety more thoroughly.

In this thesis we take an example game of Breakthrough, train a neural network to play the game using only self-play. That is, the neural network is trained only by playing against itself with no outside input other than the rules of the game itself. And we examine the neural networks against popular Higher-level concepts (heuristics) that we use to play the game.

The neural network architecture is based of the popular AlphaZero neural network used to play Chess, Shogi, and Go on a superhuman level. (REMOVE ME) Unfortunately due to lack of thousands of TPU's this research can't go as far as to examine those more complicated games. (REMOVE ME)

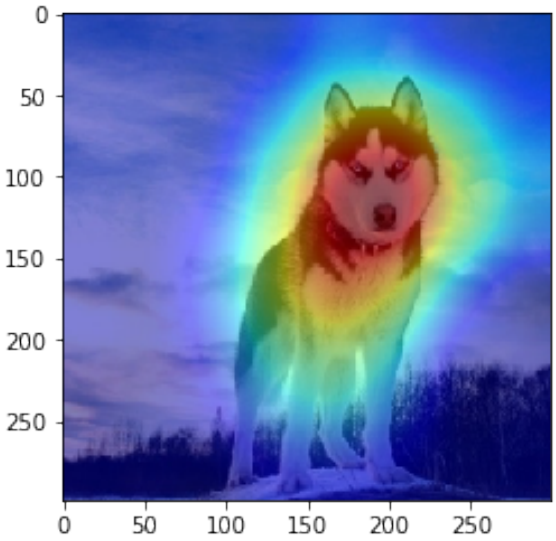

Figure 1.1: Example of a models saliency map for an image of a dog

1.1 Background

1.1.1 Explainable Artificial Intelligence (XAI)

The field of XAI research is still very far behind it's counterpart AI research, within XAI two fields are concidered the largest that is Model Interpretability and Model Explainability.

1.1.1.1 Model Interpretability

Within XAI many methods have been developed to try to evaluate ANN's, these methods are often referred as model interpretability methods. In the field image recognition there has been a lot of work examining which pixels of an image the model deems important. Arguably the most popular method for this is Saliency Maps (CITE ME). There the pixel values the model deems important are colored in s.t. a human can examine the image and get a sense of what portions of the image are important to the model, an example of a classification of a dog can be seen in Figure 1.1.

While this method is understandable in the context of image recognition it lacks severely when your input is not an image.

Methods for explaining models that aren't image recognition models include shapley values. There the input is examined against it's output, then iteratively input

values are selected to be fixed. Then the other input values are varied and an average change in prediction is calculated. With this the shapley value can be estimated for the fixed input value. This is done to examine which input values have the strongest link to the output value. Shapley values on a dataset can give insights on which input values the model deems important.

1.1.1.2 Model Explainability

Model explainability within the context of neural networks isn't possible today. Model explainability referrs to firstly considering some input and output from a model. Then afterwards the model is examined to determine exactly what led to the predicted output. This concept is simple when we're working with Decision Trees. A decision tree is a tree whose nodes are representative of an input value and at every node a branch is selected based on the value of the input value. It is therefore easy to see how to examine the tree to explain the output. We just follow the branches in the tree. That being said the branches are created by algorithms like ID3 which construct branches based on the initial dataset used to construct the tree, but again ID3 follows simple statistics and can be explained properly.

When we talk about neural networks this process is much more difficult, the underlying nodes are generally in the thousands, the different layers of the neural network varies in the operations it applies to the input value and such while travelling through the neural network the modified value becomes far removed from the initial input value to the eyes of the reader. That being said, while the possibility of completely monitoring the training process and completely monitoring the evaluation process is truly possible it is not feasible. And secondly the process of seeing an input and it's corresponding output will not be of any value if one were to consider the process of prediction.

1.1.2 Machine Learning

1.1.2.1 Reinforcement Learning

Reinforcement Learning is a subfield of Machine Learning where a model learns from experience. That is, the notion of input/output values changes, the model uses itself to generate input values by acting in an environment. The environment then returns some result, that could be losing in a game, making a correct prediction, or any number of outcomes. The result is then propagated through the model moving it's parameters in the direction of the recieved output from the environment.

Many algorithms are popular in reinforcement learning, for instance Q-learning(CITE ME), CARLA (CITE ME), Monte Carlo Tree Search, and many others. The work in this paper relies heavily on Monte Carlo Tree Search, and will be the only one thoroughly discussed.

1.1.2.2 Monte Carlo Tree Search

In the algorithm Monte Carlo Tree Search there is an agent exists within some environment. Where each node in the environment represents a state-action pair of the environment, by state-action pair what is meant it is some state and the action that brought the agent to that state. This pair should be unique within the environment. We will discuss these concepts within game-environments and therefore we might say player instead of agent and game instead of environment.

Monte Carlo Tree Search is a method of exploring an enviroment in a randomized manner (Monte Carlo is the term implying randomness). In MCTS there are four stages. Selection, Expansion, Simulation, and Backpropagation. They happen sequentially and repeatedly. MCTS is initialized with a tree consisting of the unexpanded initial state of the environment.

In MCTS there is a tree representing the game-environment. This tree consists of nodes n_i where i represents the point in time of the node, for example N_0 in chess is the initial position and N_x is some position in the middle of the game and N_e is one of the states representing a position where there is either a draw or one player has won the

match. Each of the nodes has 4 values, s, a, Q, and N. These values represent these items, s is state of the environment, a is the action that brought the previous node $n_(i-1)$ to node n_i, Q is the average value of the node (value meaning the outcome of the game), and N the amount of times the MCTS algorithm has visited this node. The values s and a uniquely identify a position in the environment and are often called state-action pairs.

The MCTS algorithms four phases

1. Selection
2. Expansion
3. Simulation/Rollout
4. Backpropagation

$$\text{Child UCT value} = \frac{Q_{(s',a')}}{N_{(s',a')}} + c_{uct} * \frac{\sqrt{log(N_{(s,a)})}}{N_{(s',a')}} \tag{1.1}$$

Selection

During the selection phase a node (s, a) within the tree which has not yet been expanded is found. This process uses Upper Confidence Bound on Trees (UCT) to find that node (s, a), the formula is described in Equation 1.1. For a parent node (s, a) (initially the root of the tree) we select the child with the highest UCT value. Repeatedly until an unexpanded node is found. This process is done to balance the amount of exploration vs exploitation of nodes in the tree.

Expansion

Then the expansion phase expands the node generating all of (s, a)'s children (s', a') are generated and connected to (s, a). This is done by applying all actions a' to (s, a). These children are the product of applying each action a' to s in the parent node.

Rollout

Next during rollout actions from (s, a) are randomly selected to move to (s', a'), then repeated to go to (s'', a''), until a terminal node within the environment is reached. By terminal we mean a state in which the game is finished. A terminal node in MCTS can generally return any value, but in the context of this paper we only return (+1 white wins, -1 black wins, 0 draw).

Backpropagation

The result from the terminal node is then propagated up through the path taken by selection (s, a) up to the root of the tree, updating the $Q_{(s,a)}$ values of each node (s, a).

When training a neural network the UCT formula is modified slightly to prefer selecting nodes that the neural network values highly by introducing a second scalar to the formula $f(s, a) = (p, v)$, the resulting formula is described in Equation 1.2, and is called PUCT. Secondly, the backpropagation process is modified to instead of doing rollout/simulation to recieve a reward the predicted value v from the neural network is used instead.

$$\text{Child PUCT value} = \frac{Q_{(s',a')}}{N_{(s',a')}} + c_{uct} * P((s, a)) * \frac{\sqrt{log(N_{(s,a)})}}{N_{(s',a')}} \tag{1.2}$$

This part of the dissertation describes the work done.

Chapter 2

Methods

2.1 Monte Carlo Tree Search

To train the models

This section discusses the various machine learning methods utilized throughout this project and discusses their applicability.

I'm a little teapot

I'm a smaller teapot

Chapter 3

Results

In this section you discuss any issues that came up while developing the system. If you found something particularly interesting, difficult, or an important learning experience, put it here. This is also a good place to put additional figures and data.ello world

This part of the dissertation talks about future work discussion concludes the work and

Chapter 4

Discussion

Here I will discuss the myriad of fields research like this can help for instance for health organizations, patients, taxpayers, and pharmaey

4.1 Summary

summarize the workey

4.2 Conclusion

conclude the work, discuss the significance of the workey