

Todo list

8 - 10 pages of introduction	1
4 - 5 pages	1
describe search in general and with a focus on MCTS	1
describe MCTS, selection, expansion, e.t.c.	2
also describes how MCTS and NN work together	2
(5 pages)	2

1 Introduction

8 - 10 pages of introduction

Deep Learning (DL) is a branch of **Artificial Intelligence (AI)** that emphasizes the use of neural networks to fit the inputs and outputs of a dataset. The training of a neural network is done by computing the gradients of the loss function with respect to the weights and biases of the network. A better trained neural network can better approximate the function that maps the inputs to the outputs of the dataset.

Reinforcement Learning (RL) is a branch of AI that emphasizes on solving problems through trials and errors with delayed rewards. RL had most success in the domain of **Game Playing**: making agents that could play boardgames, Atari games, or other types of games. An extension to Game Playing is **General Game Playing (GGP)**, with the goal of designing agents that could play any type of game without having much prior knowledge of the games.

Deep Reinforcement Learning (DRL) is a rising branch that combines DL and RL techniques to solve problems. In a DRL system, RL usually defines the backbone structure of the algorithm especially the Agent-Environment interface. On the other hand, DL is responsible for approximating specific functions by using the generated data.

Planning refers to any computational process that analyzes a sequence of generated actions and their consequences in the environment. In the RL notation, planning specifically means the use of a model to improve a policy.

A **Distributed System** is a computer system that uses multiple processes with various purposes to complete tasks.

In this thesis we will describe a framework for solving the problem of GGP. We will also detail **MooZi**, a system that implements the GGP framework and the **MuZero** algorithm for playing both boardgames and Atari games.

2 Literature Review

4 - 5 pages

Early AI research has been focused on the use of search as a planning method. Algorithms like **A*** were designed to find the optimal path to goals. Although A* works quite well for many problems, it falls short in cases where the assumptions of A* do not hold. For example, A* does not yield optimal solution under stochastic environments and it could be computationally infeasible on problems with high branching

describe search in general and with a focus on MCTS

factors. More sophisticated search algorithms were developed to cater to the growing complexity of use cases.

Real-Time Heuristic Search pioneered the study of search algorithms with bounded computation. Monte-Carlo techniques were adopted to handle environment stochasticity. Tree-based search algorithms such as **MiniMax** and **Alpha-Beta Pruning** were also designed to better play two-player games.

2.1 Monte-Carlo Tree Search (MCTS)

Monte-Carlo Tree Search (MCTS) is a search algorithm for game AI that combines both Monte-Carlo rollouts and tree search. MCTS requires less domain knowledge than other classic approaches to game AI while also being competent in strength.

describe MCTS, selection, expansion, e.t.c.

2.2 AlphaGo, AlphaGo Zero, and Alpha Zero

AlphaGo is the first Go program that beats a human Go champion. AlphaGo first learns the policy

also describes how MCTS and NN work together

AlphaGo Zero is a successor of AlphaGo with the main difference of not learning from human.

- MuZero - other open source implementations

3 Problem Definition

(5 pages)