Reinforcement Learning for Game-Playing: Experiments with AlphaZero

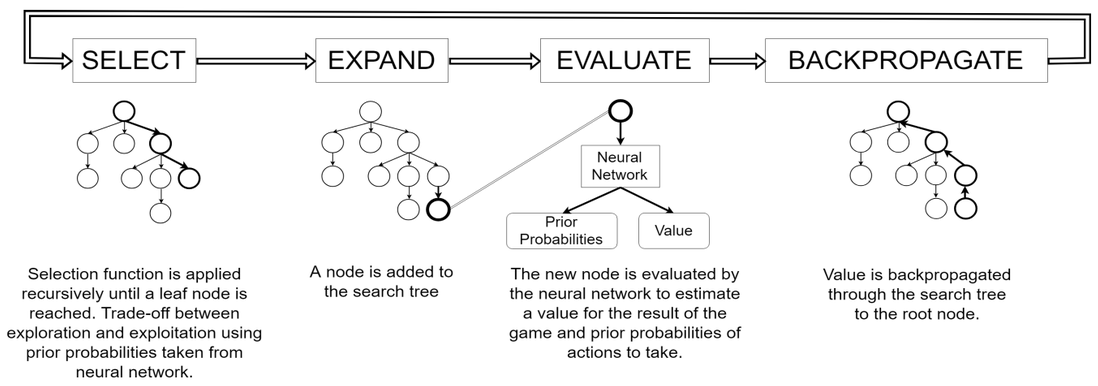
Stephen Church, School of Engineering and Computer Science

INTRODUCTION

Reinforcement learning is a branch of machine learning that focuses on agents learning from interactions in an environment in order to maximise some external reward without the use of labelled expert data. AlphaGo combined reinforcement learning with advancements in deep neural networks to achieve its 4-1 victory over world Go champion, Lee Sedol, in March 2016. Its successor, AlphaZero, improved on these results, but did so with enormous computing resources and little explanation of parameters used. This project focuses on implementing AlphaZero for two games to verify its usefulness on a smaller scale, and further looks to test the feasibility of pre-training networks on small instances of a board game and then fine-tuning them on bigger instances.

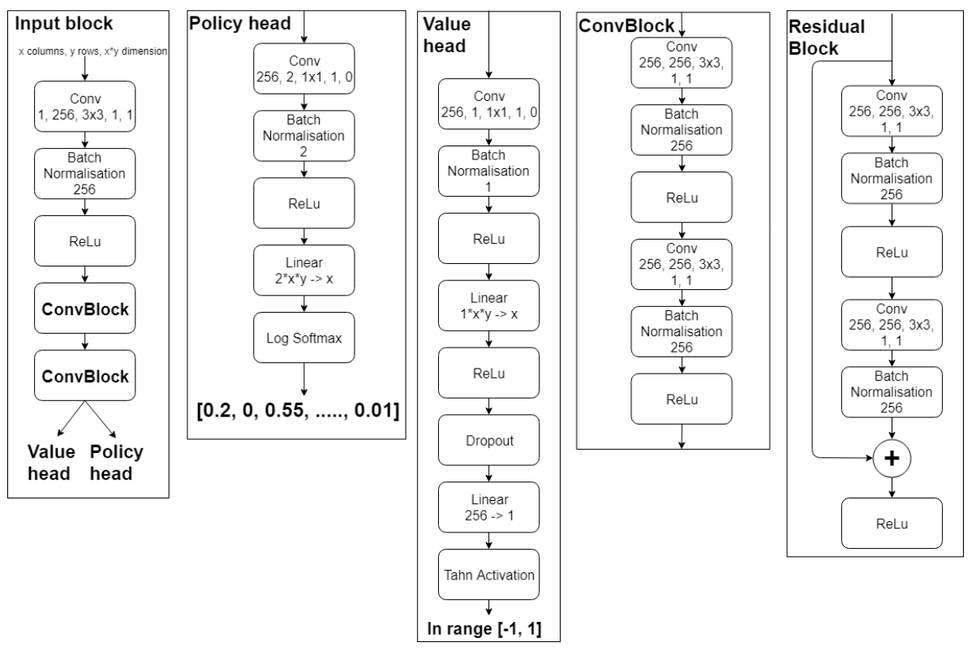
ALPHAZERO AND MONTE CARLO TREE SEARCH (MCTS)

Monte Carlo Tree Search (MCTS) forms the basis of searching game states in AlphaZero and is augmented with a neural network. The basic MCTS algorithm is also implemented: this differs as there are no prior probabilities and the "evaluate" stage is replaced with a "simulate" stage in which random moves are played until the end of the game to determine the value.



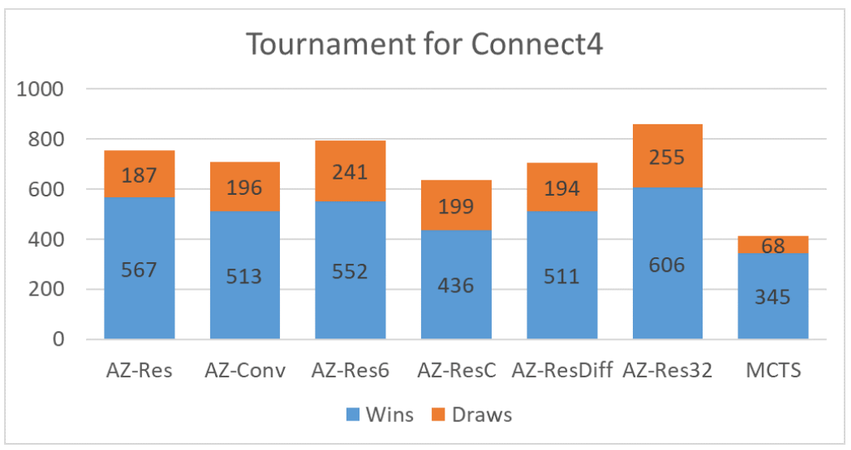
NEURAL NETWORK

All implementations use a split-head architecture, first passing the inputs through an input block before splitting into a value head and policy head. The value head returns an estimate of the reward for the game (between -1 for a loss and +1 for a win). The policy head returns a list of prior probabilities indicating the probability that a player should take each of the next possible moves from the input position. Two network architectures were tried, one using ConvBlocks and the other using Residual Blocks instead.



RESULTS: CONNECT4

Different variations of players were trained for 50 iterations on Connect4 using the convolutional architecture and the residual architecture. These players played in a tournament against each other, and against a basic MCTS player, with each pair of players playing 200 games and each player having the same number of MCTS iterations. Each player was awarded +1 for a win, and -1 for a loss.



TRAINING

Training is done via games of self-play, with a cyclical loop of generating training examples using the current neural network, refining the examples using MCTS, training the neural network on those examples and repeating.

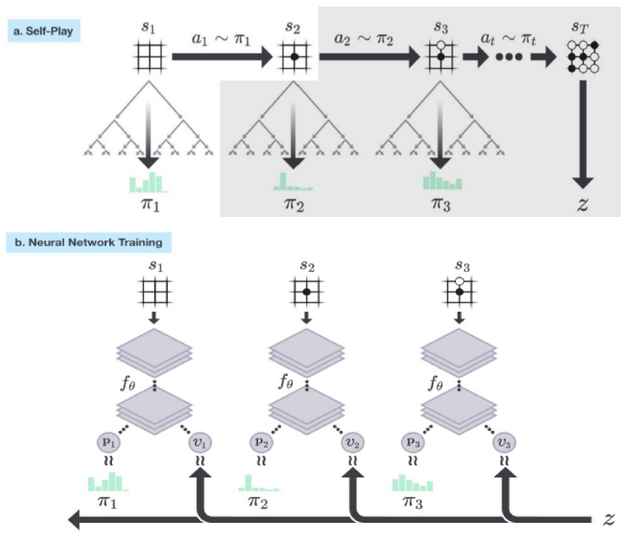
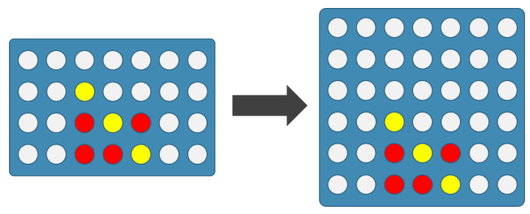


Figure: Taken from https://medium.com/@jonathan_hui/alphago-zero-a-game-changer-14ef6e45eba5.

RESULTS: SCALABILITY



To understand if traditional fine-tuning methods could be applied to scale up AlphaZero to bigger boards, an AlphaZero player was first trained on a 6x7 Connect4 board. This network was then re-used to train an AlphaZero player on a 8x9 board, with the value and policy heads replaced to fit the new board dimensions and all other layers frozen. This player was compared to another AlphaZero player that was trained from scratch on an 8x9 board for the same amount of time.