
AlphaGo Zero

Shahzeb Aamir

Content:

1. What is Go?
2. What is AlphaGo?
3. Monte Carlo Tree Search
4. NN Architecture

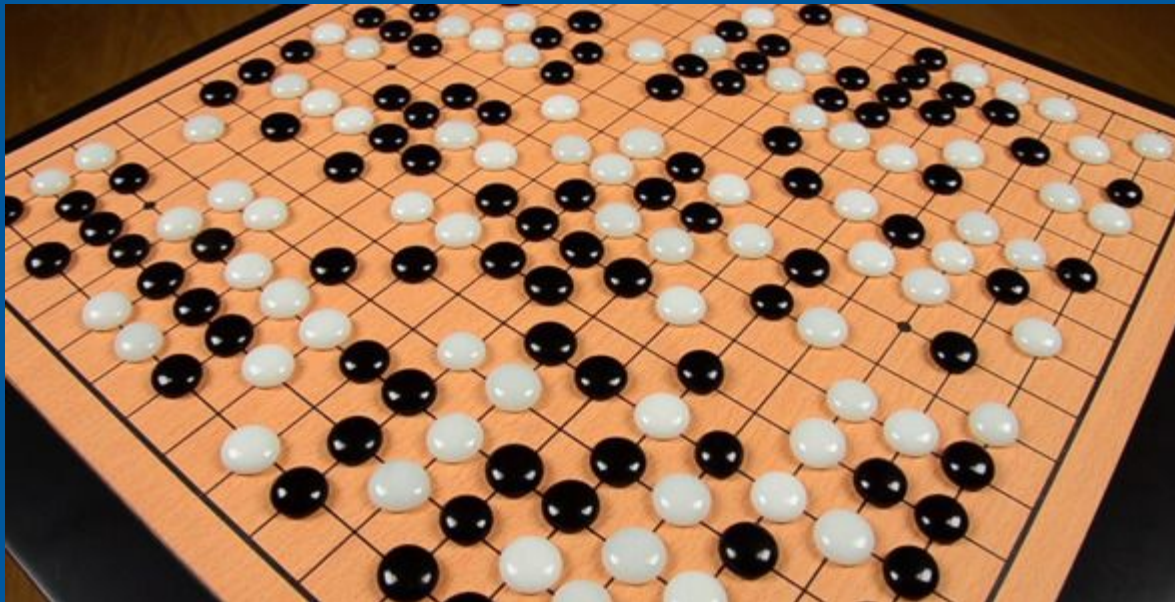
Go

19x19 grid

Turn-Based, Two players game

Goal:

Surround and capture opponents
stones, or strategically create
spaces of territory.



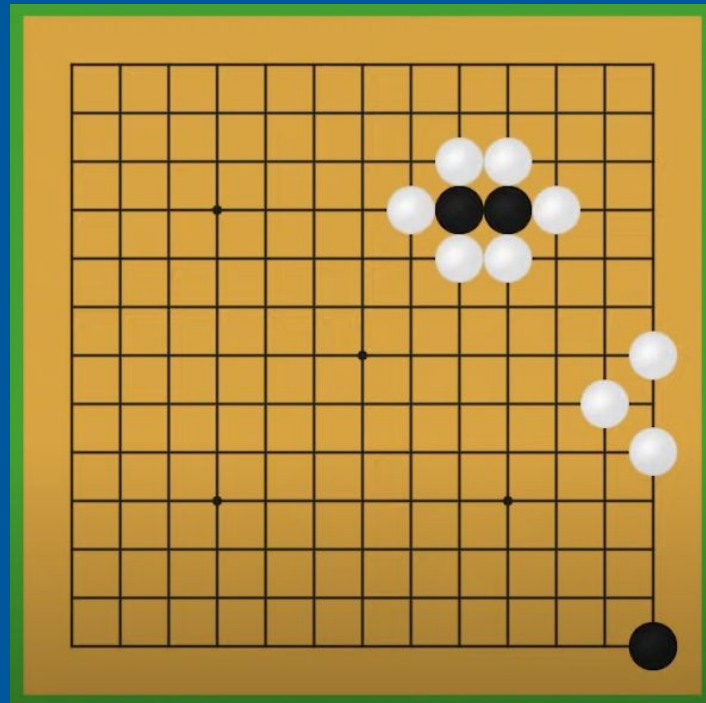
Go

19x19 grid

Turn-Based, Two players game

Goal:

Surround and capture opponents
stones, or strategically create
spaces of territory.



Go

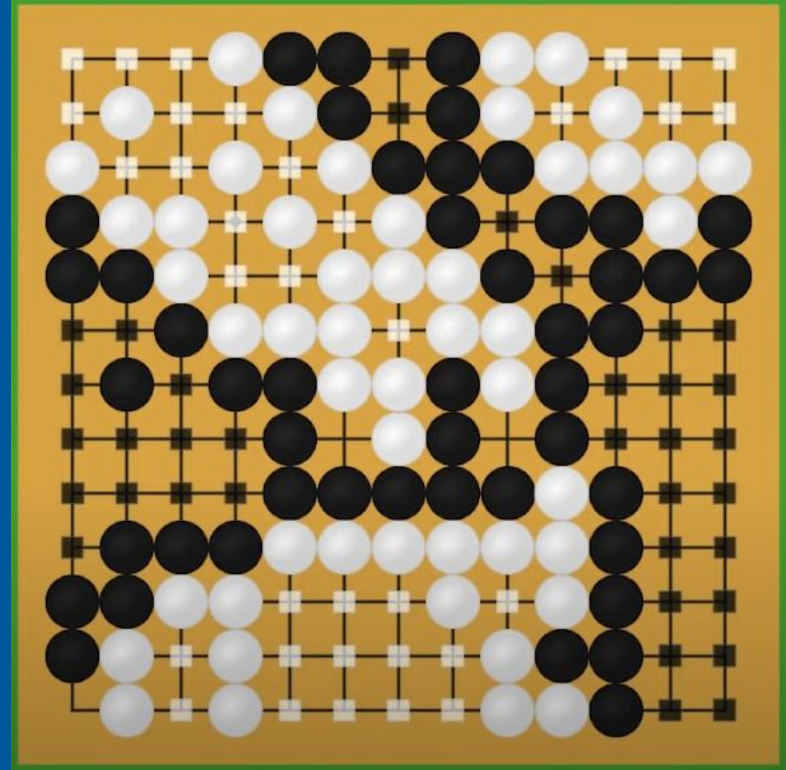
19x19 grid

Turn-Based, Two players game

Goal:

Surround and capture opponents
stones, or strategically create
spaces of territory.

Highest points of "empty spaces" win



Go

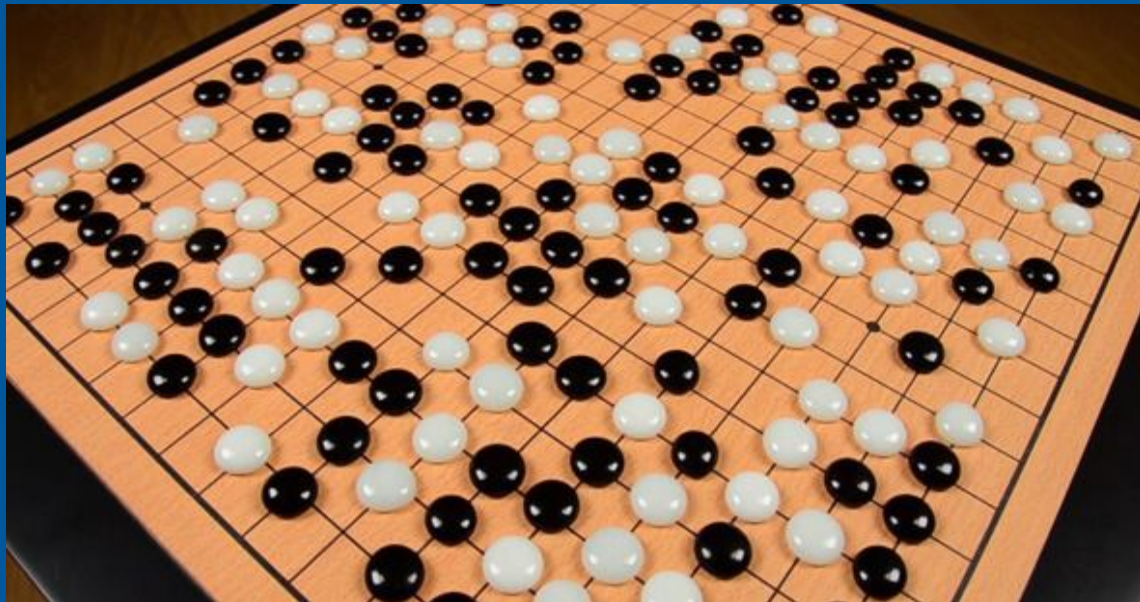
Simple rules, right?

However, it has 10^{170} possible board
Configurations

More than the atoms in this universe

YES, you read right. More than the
atoms in this known universe

It's a googol times more complex than
chess



What is AlphaGo

AlphaZero, a single system that taught itself from scratch how to master the games of chess, shogi, and Go, beating a world-champion computer program in each case.

What Problems does it solve?

Turn-based, fully observable positions with definite sets of rules. The opponent goal is to prevent us from winning

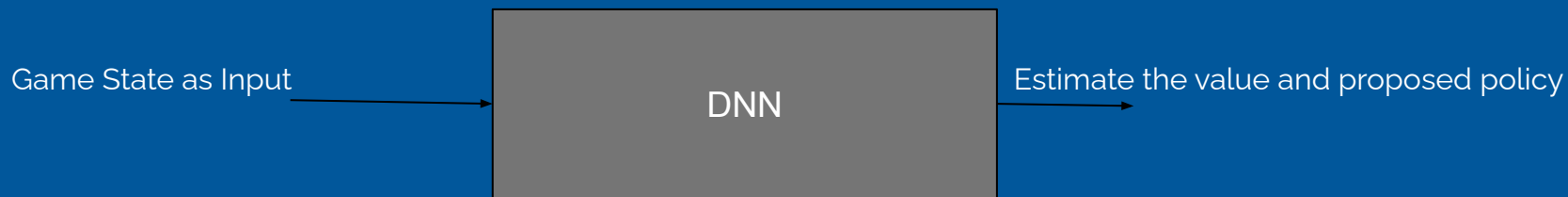
Not so intelligent Approach

Brute force method. Search all possible moves and its subsequent branches to evaluate and select the best move

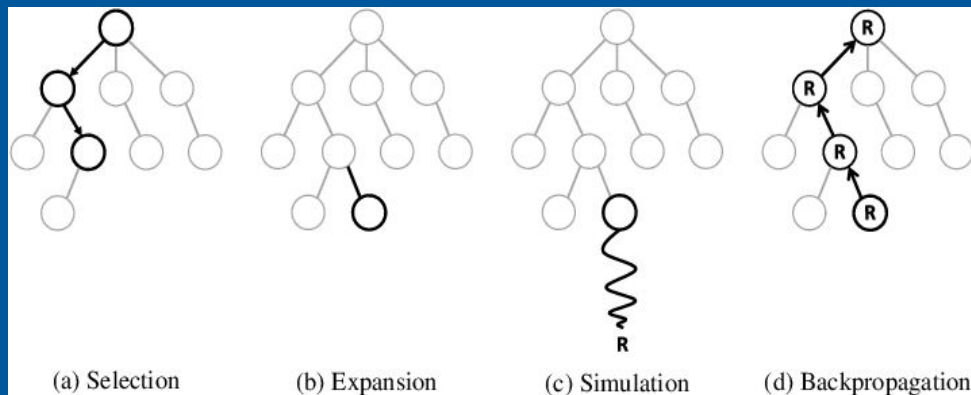
A better Approach, and the idea behind AlphaGo

A deep neural network estimates the most promising set of moves in a search tree

How does it work



Algorithm that performs intelligent search for possible moves based on the suggestion of DNN: Monte Carlo Tree Search



AlphaGo Cheat Sheet

ALPHAGO ZERO CHEAT SHEET

The training pipeline for AlphaGo Zero consists of three stages, executed in parallel

SELF PLAY

Create a 'training set'

The best current player plays 25,000 games against itself.
The MCTS action is selected (see AlphaGo Zero selects best move).
At each move, the following information is stored:



RETRAIN NETWORK

Optimise the network weights

A TRAINING LOOP
Sample a new batch of 500K positions from the last 500,000 games.
Retrain the neural network on these positions.
The game ends on the final move (See AlphaGo Zero selects best move).
Loss Function
Compare predictions from the neural network with the search probabilities and self-play results.
PREDICTIONS: P (Cross entropy), V (Mean squared error), π (Regression).
ACTUAL: π (Regression).

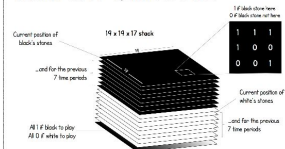
After every 1,000 training loops, evaluate the network

EVALUATE NETWORK

Test to see if the new network is stronger
Play 1000 games between the latest neural network and the current best neural network.
Both players use MCTS to select their moves, with their respective neural networks to evaluate leaf nodes.
Loldest player must win 55% of games to be declared the new best player.



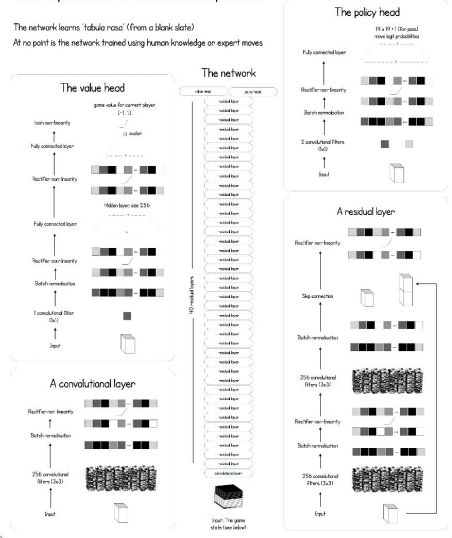
WHAT IS A 'GAME STATE'?



This stack is the input to the deep neural network

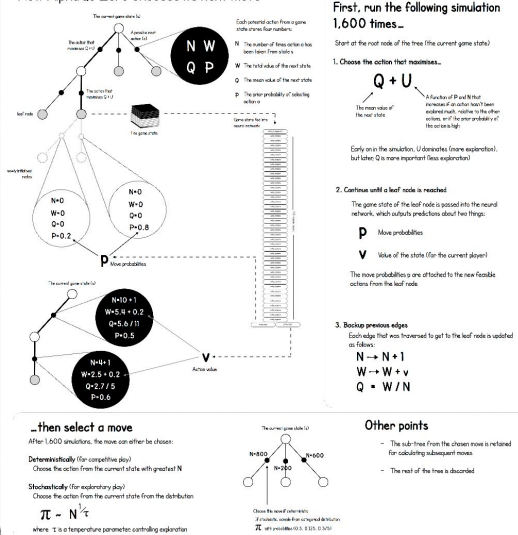
THE DEEP NEURAL NETWORK ARCHITECTURE

How AlphaGo Zero assesses new positions
The network learns 'tabula rasa' (from a blank slate).
At no point is the network trained using human knowledge or expert moves



MONTE CARLO TREE SEARCH (MCTS)

How AlphaGo Zero chooses its next move



How does it work

<https://medium.com/applied-data-science/alphago-zero-explained-in-one-diagram-365f5abf67e0>

Three stages executed in parallel

SELF PLAY

Create a 'training set'

The best current player plays 25,000 games against itself

See MCTS section to understand how AlphaGo Zero selects each move

At each move, the following information is stored



The game state
(see 'What is a Game State' section)

π

The search probabilities
(from the MCTS)



The winner
(+1 if this player won, -1 if this player lost - added once the game has finished)

RETRAIN NETWORK

Optimise the network weights

A TRAINING LOOP

Sample a mini-batch of 2048 positions from the last 500,000 games

Retrain the current neural network on these positions

- The game states are the input (see 'Deep Neural Network Architecture')

Loss function

Compares predictions from the neural network with the search probabilities and actual winner

$$\begin{array}{ccccc} \text{PREDICTIONS} & \mathbf{P} & \text{Cross-entropy} & \mathbf{\pi} & \text{ACTUAL} \\ & \mathbf{V} & + & & \\ & & \text{Mean-squared error} & & \\ & & + & & \\ & & \text{Regularisation} & & \end{array}$$

After every 1,000 training loops, evaluate the network

EVALUATE NETWORK

Test to see if the new network is stronger

Play 400 games between the latest neural network and the current best neural network

Both players use MCTS to select their moves, with their respective neural networks to evaluate leaf nodes

Latest player must win 55% of games to be declared the new best player



Monte Carlo Tree Search:

The Phases of MCTS:

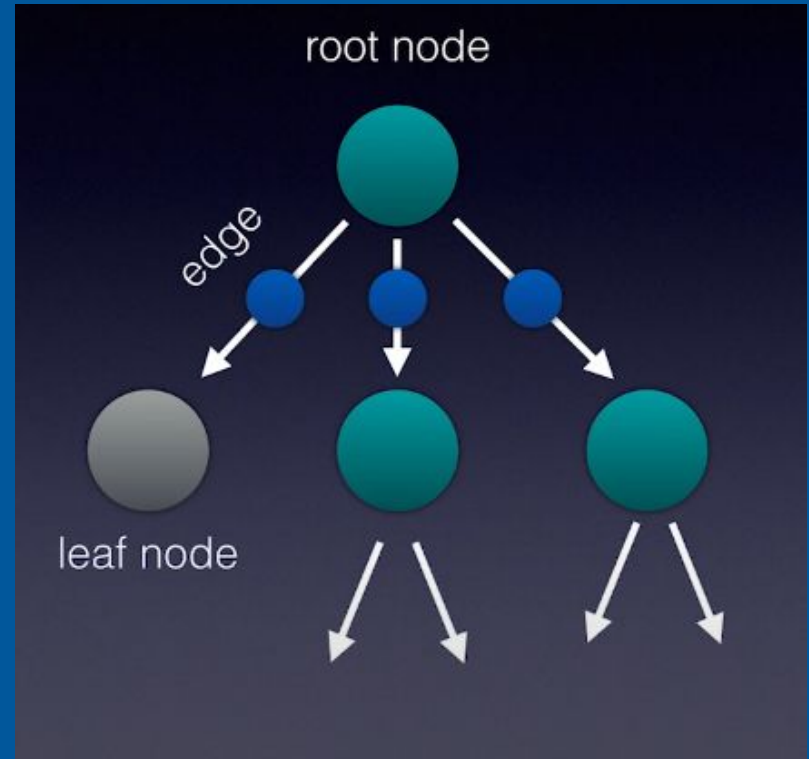
SELECT (Root to Leaf that is most promising)

EXPAND (by using one more move)

BACK UP (and update all edges traversed using statistics)

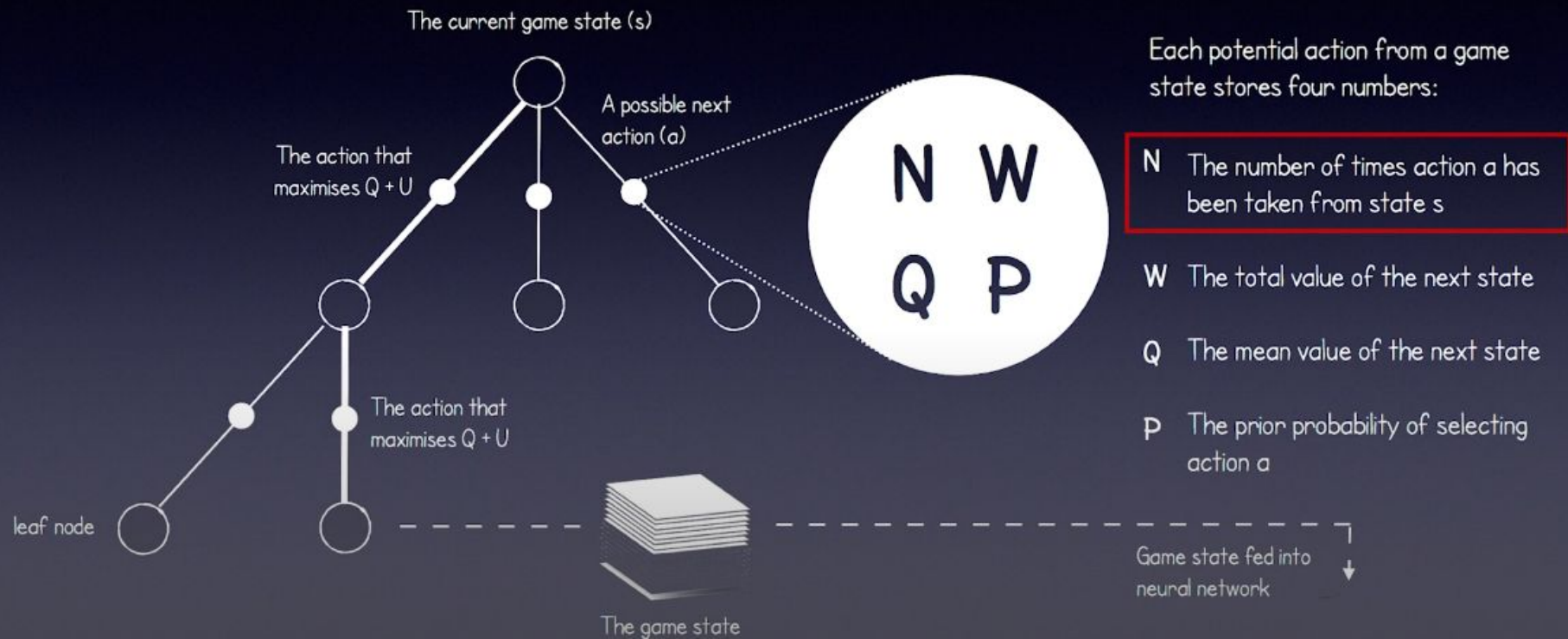
X 1600 times

PLAY



Monte Carlo Tree Search:

Statistics:



Monte Carlo Tree Search:

Details:

1. Choose the action that maximises...

$$Q + U$$

↖
The mean value of
the next state

↗ A function of **P** and **N** that
increases if an action hasn't been
explored much, relative to the other
actions, or if the prior probability of
the action is high

Early on in the simulation, U dominates (more exploration),
but later, Q is more important (less exploration)

$$U(s, a) = c_{puct} \cdot P(s, a) \cdot \frac{\sqrt{\sum_b N(s, b)}}{1 + N(s, a)}$$

Monte Carlo Tree Search:

Details:

2. Continue until a leaf node is reached

The game state of the leaf node is passed into the neural network, which outputs predictions about two things:

p

Move probabilities

v

Value of the state (for the current player)

The move probabilities p are attached to the new feasible actions from the leaf node

Monte Carlo Tree Search:

Details:

3. Backup previous edges

Each edge that was traversed to get to the leaf node is updated as follows:

$$N \rightarrow N + 1$$

$$W \rightarrow W + v$$

$$Q = W / N$$

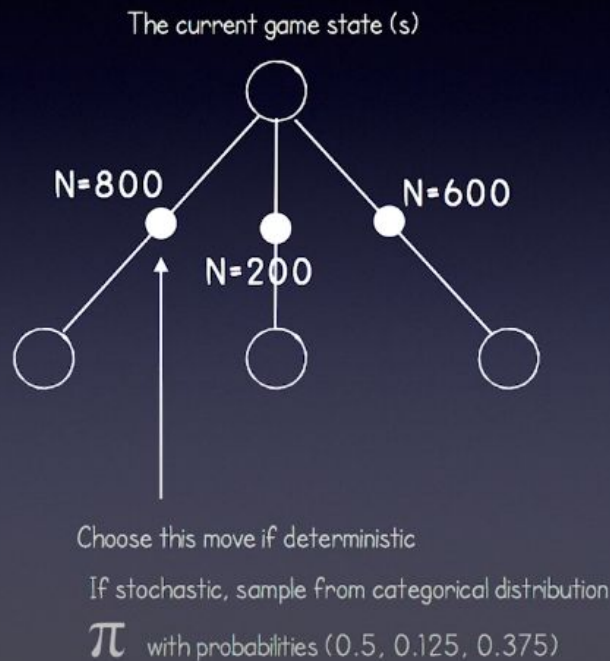
Monte Carlo Tree Search:

Details: For Playing (Selecting the move)

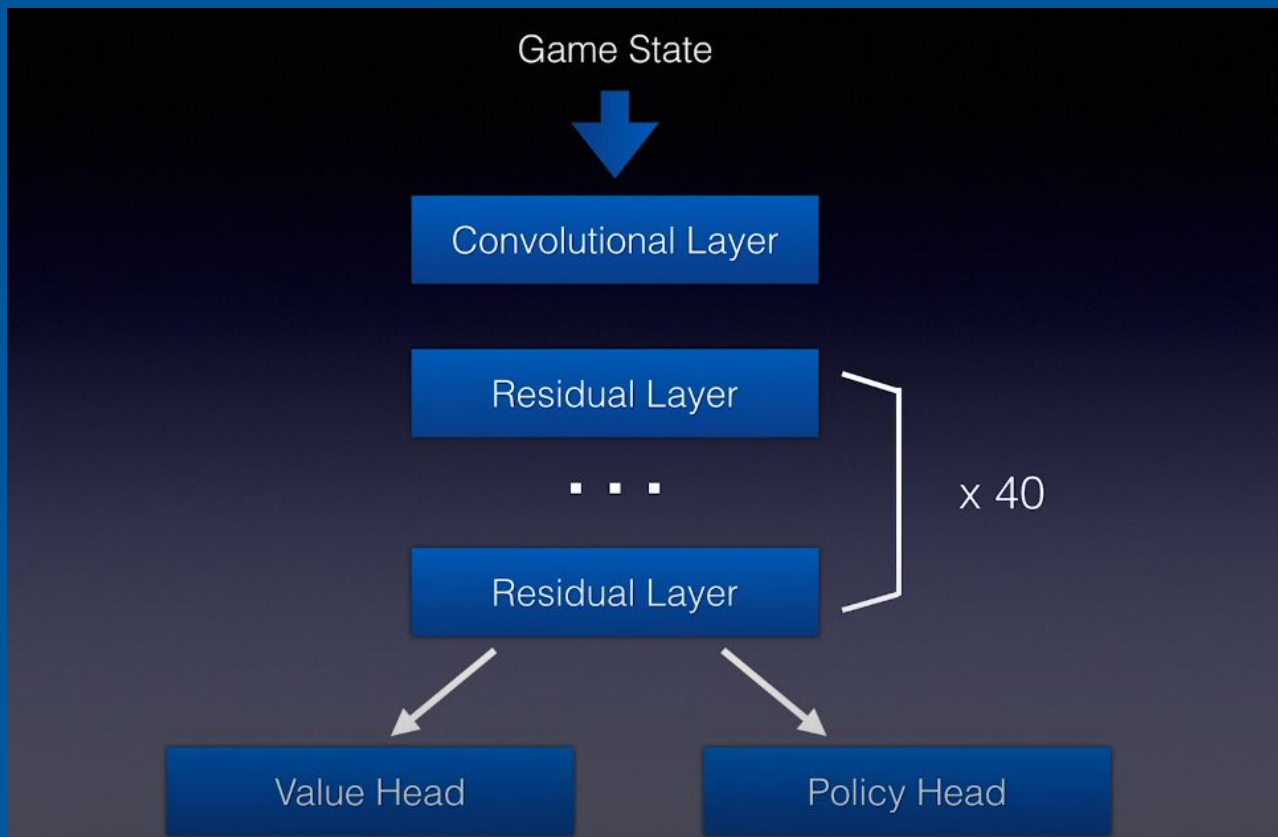
- **Deterministically** (for competitive play)
choose the action with the greatest N
- **Stochastically** (for training)
sample randomly from the probability distribution...

$$\pi(a | s) = \frac{N(s, a)^{1/\tau}}{\sum_b N(s, b)^{1/\tau}}$$

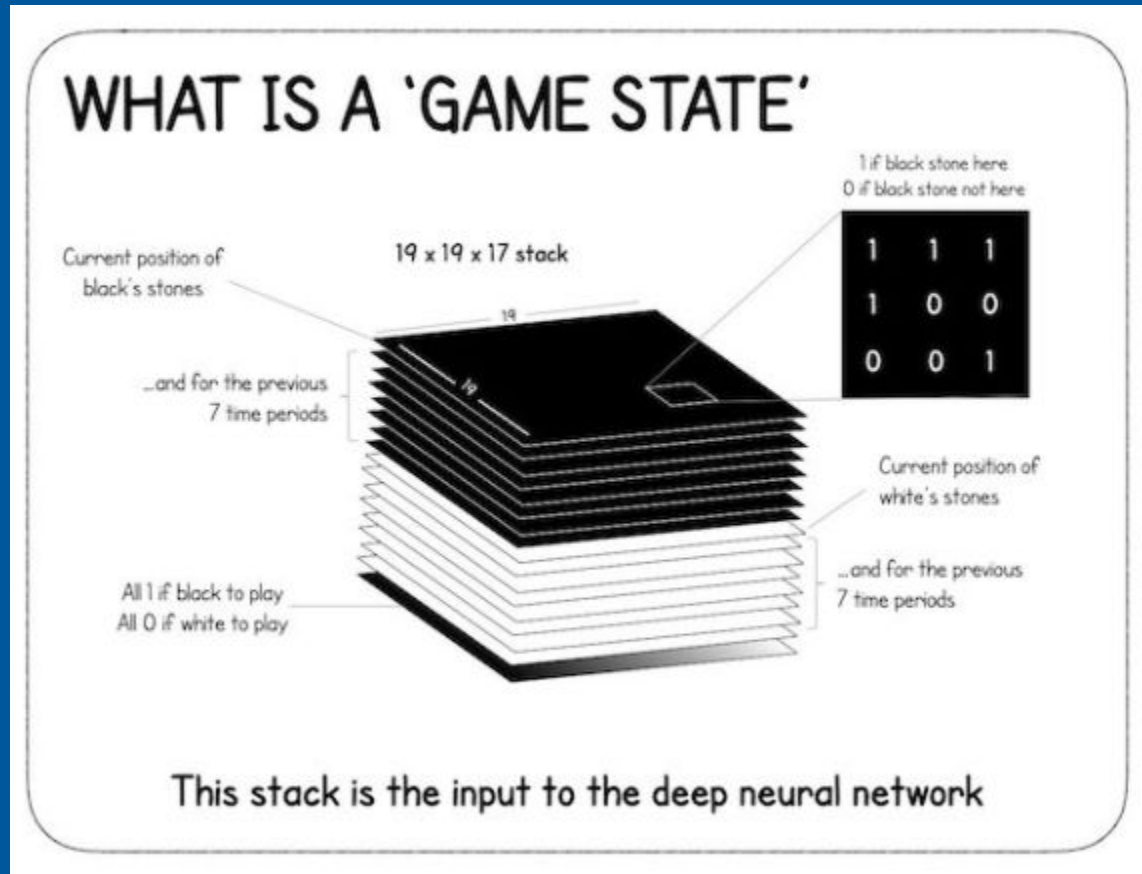
Where τ is a temperature parameter, controlling exploration.



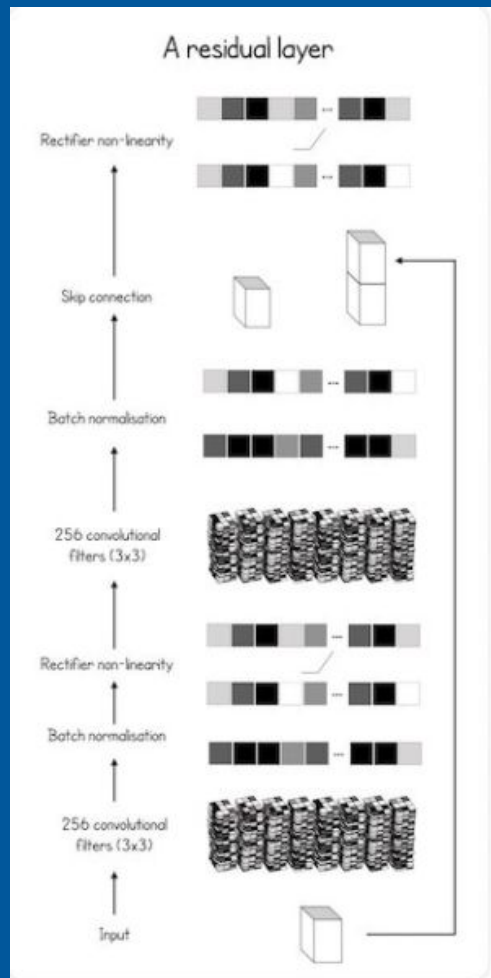
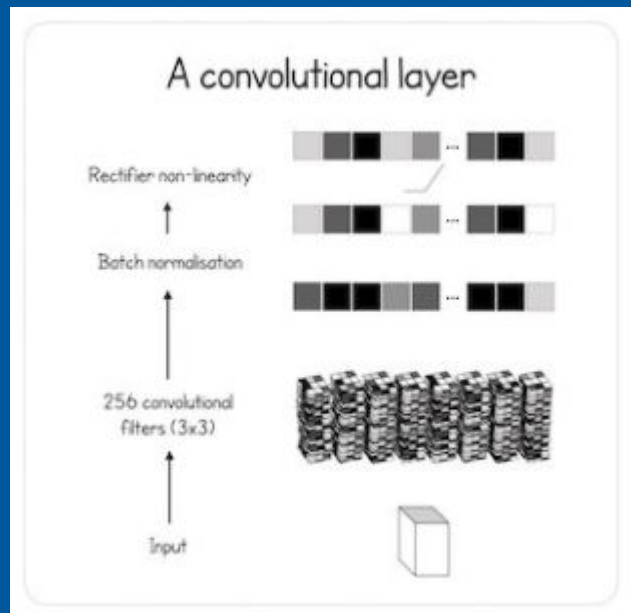
NN Architecture:



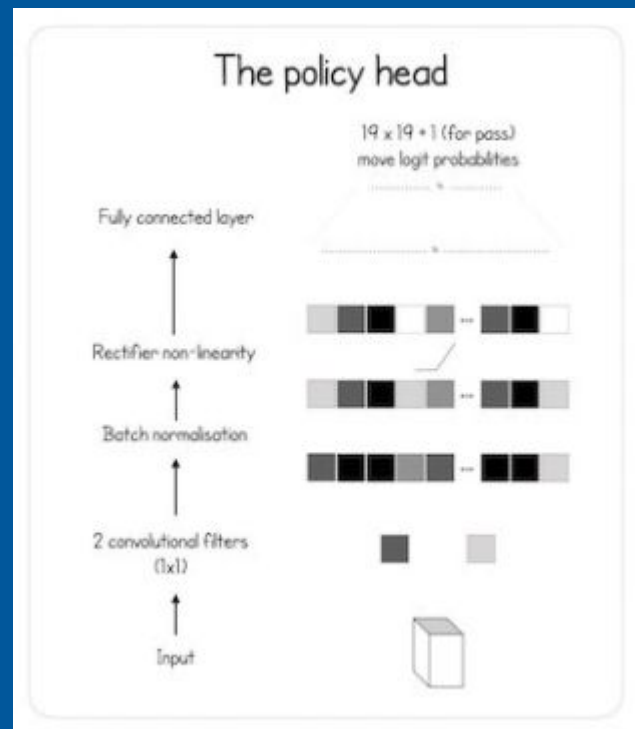
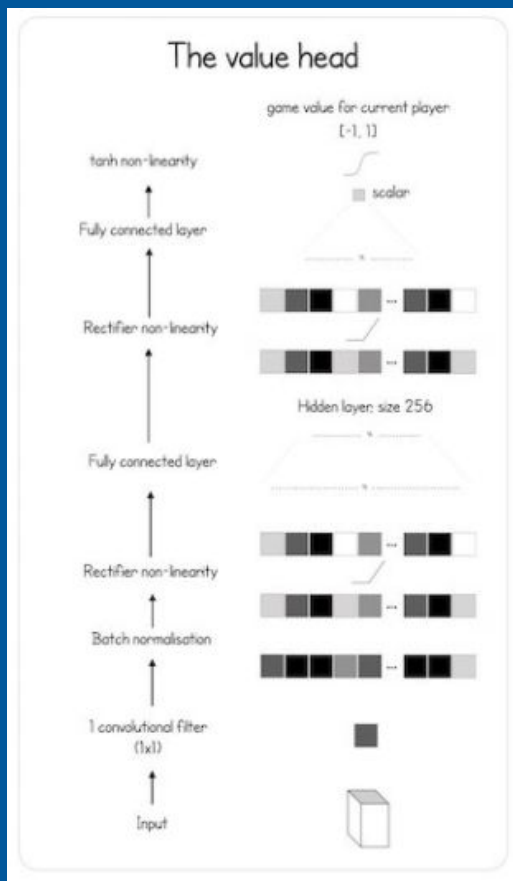
Game State:



NN Architecture:

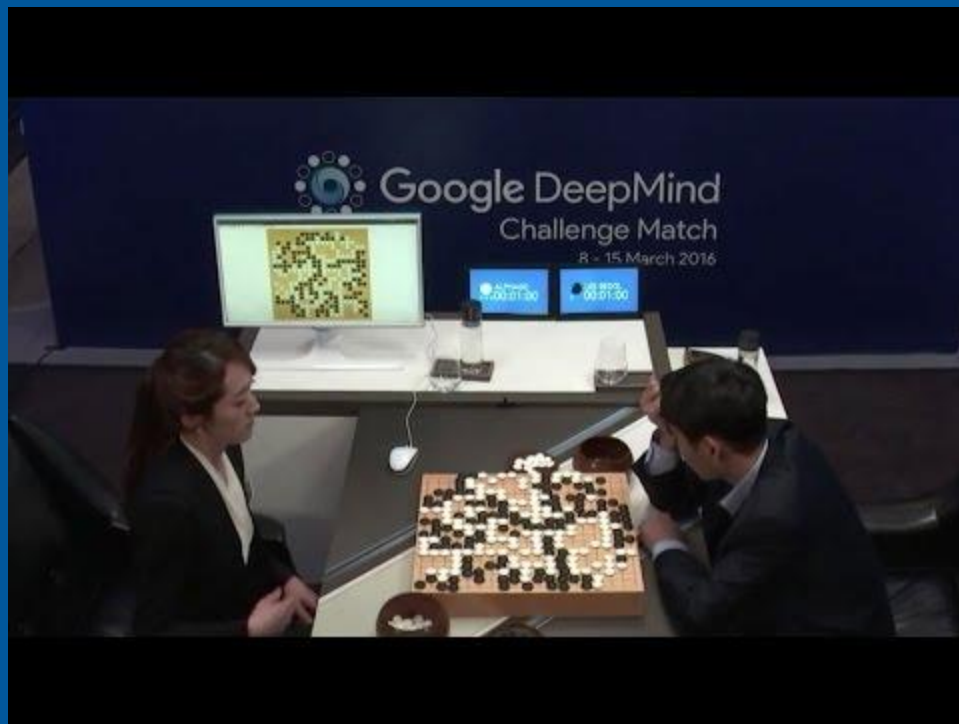


NN Architecture:



$$L = (z - v)^2 - \pi \cdot \log(p) + c \cdot \|\theta\|^2$$

AlphaGo beats Lee Sedol:





Skolkovo Institute of Science and Technology

Thank You!