

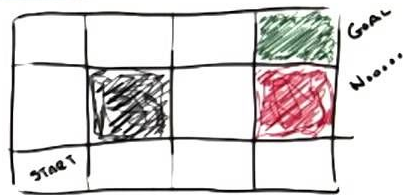
AlphaGo Zero

Starting from scratch



- 1) MDP
- 2) MCTS
- 3) Q-Learning
- 4) Residual neural networks

Markov Decision Processes



STATES : S

MODEL : $T(s, a, s') \sim \Pr(s' | s, a)$ UP, DOWN, LEFT, RIGHT

ACTIONS : $A(s), A$

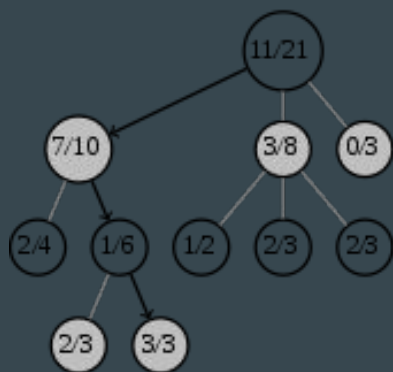
REWARD : $R(s), R(s, a), R(s, a, s')$

Policy : $\pi(s) \rightarrow a$

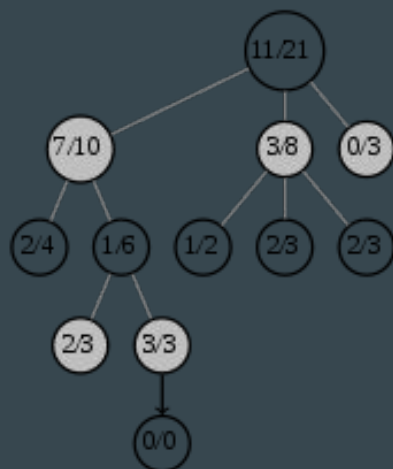
π^*

Monte Carlo Tree Search

Selection



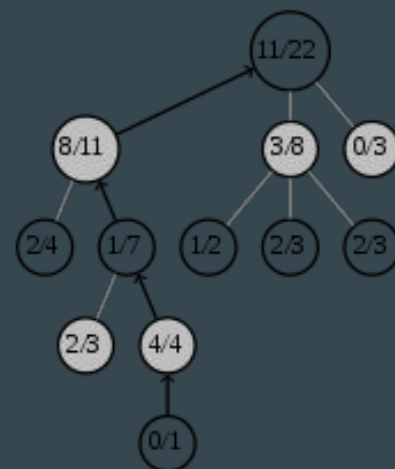
Expansion



Simulation



Backpropagation



Monte Carlo Tree Search

Monte Carlo Tree Search

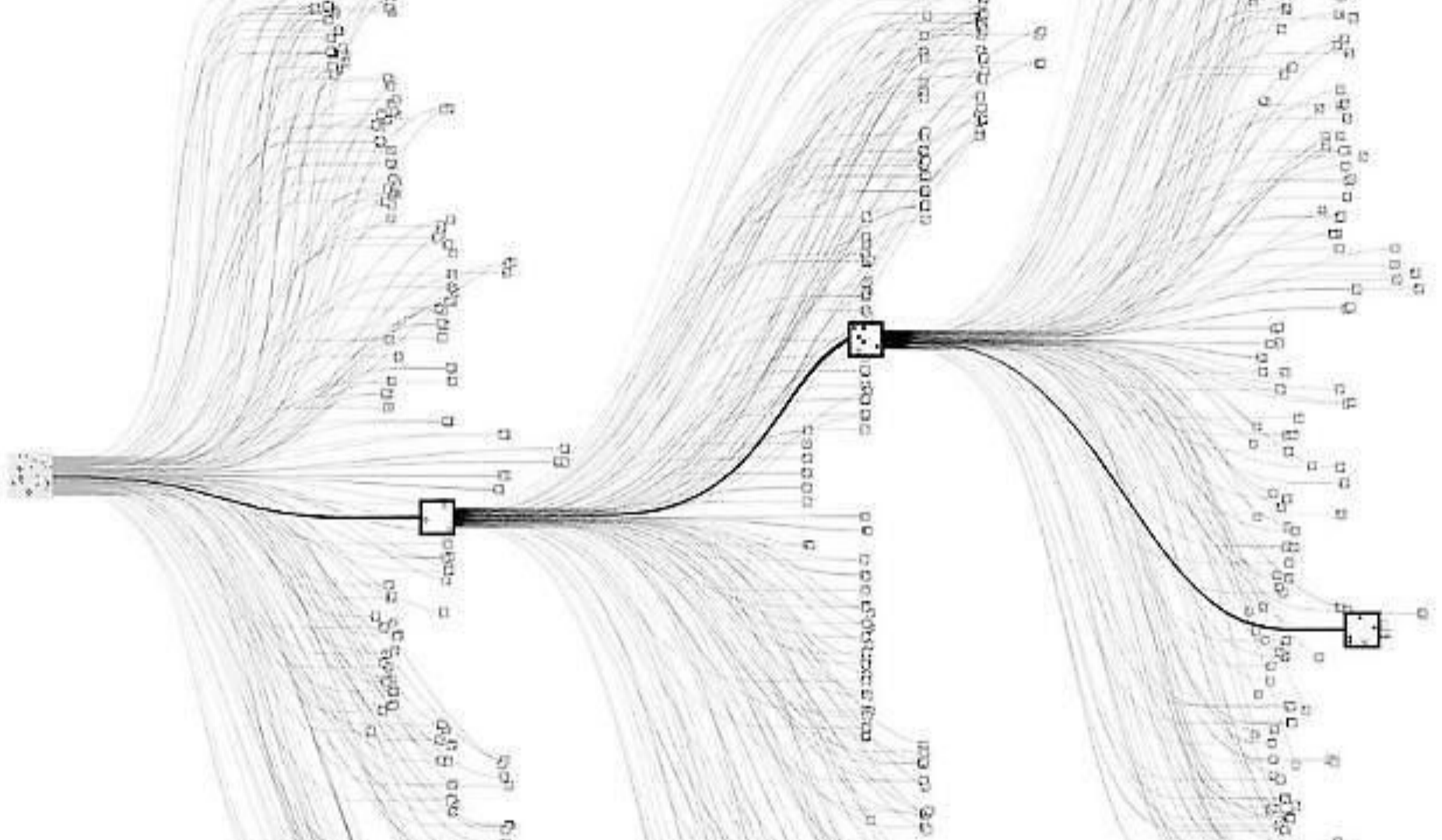
Worked example:

$$UCB1(S_i) = \bar{V}_i + 2 \sqrt{\frac{\ln N}{n_i}}$$

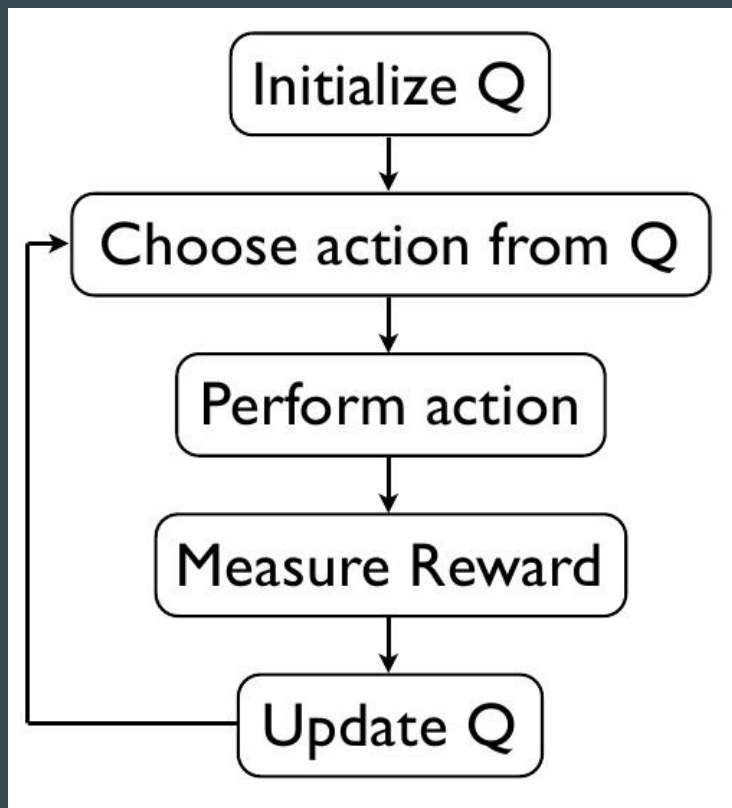
The diagram shows a tree structure with nodes S_0, S_1, S_2, S_3, S_4 and a terminal state $V=0$. The nodes are labeled with t and n values. The edges are labeled a_1, a_2, a_3, a_4 . A wavy line connects S_3 to $V=0$.

12:04 / 15:49

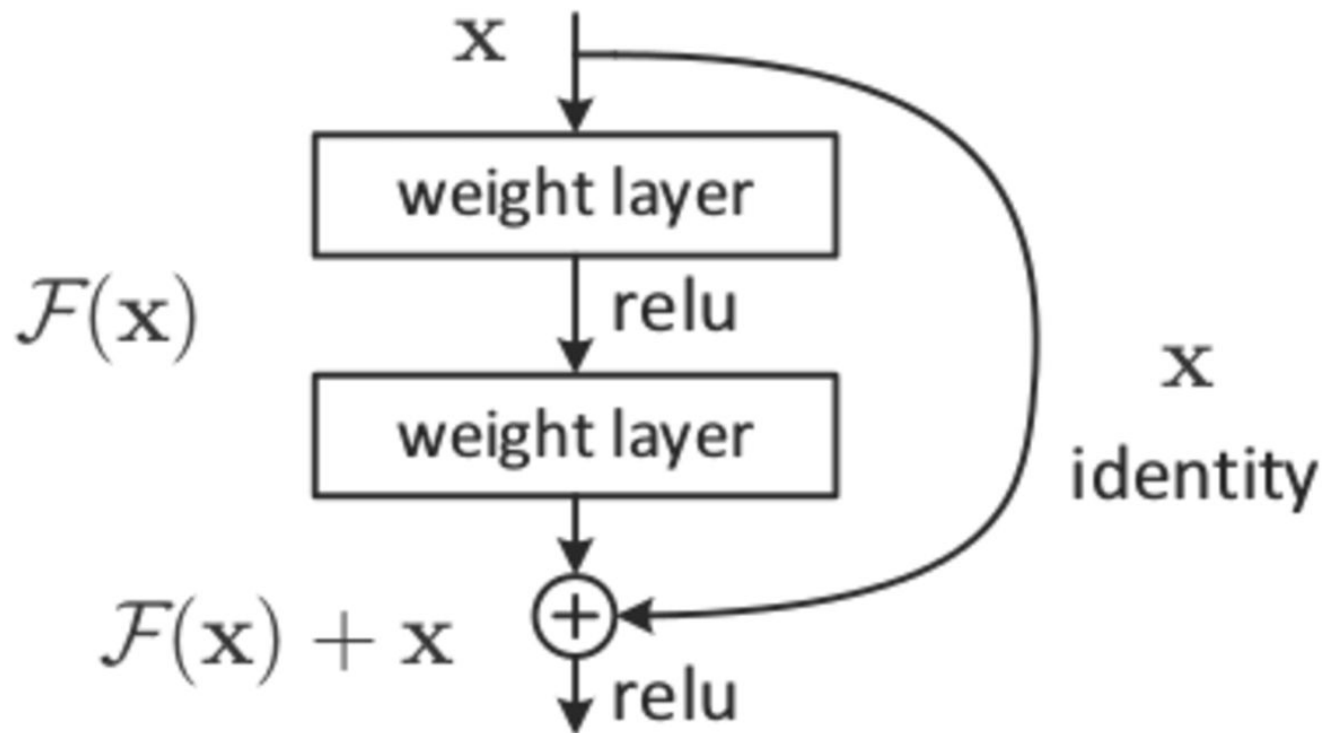
source : <https://www.youtube.com/watch?v=UXW2yZndl7U&t=127s>



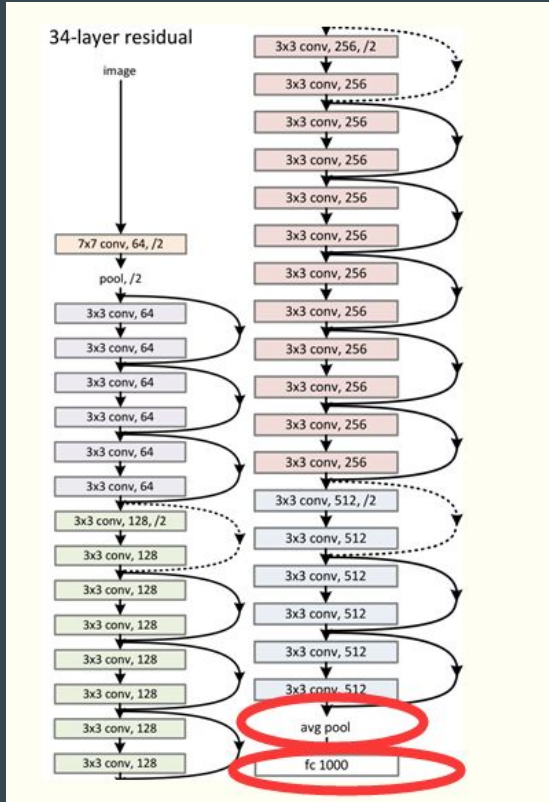
Q-learning



Residual Neural Networks



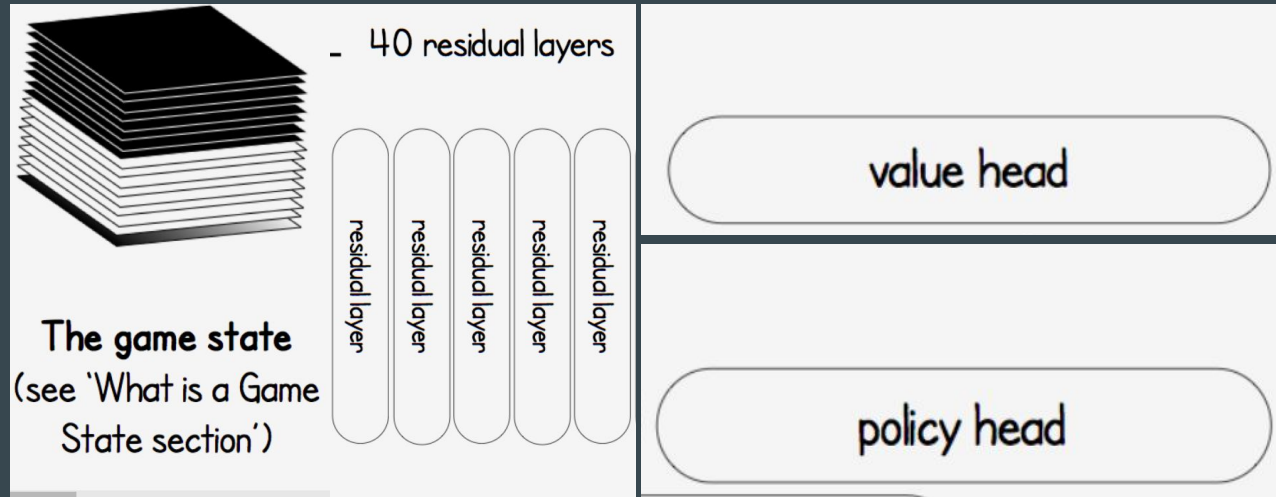
Residual Neural Networks



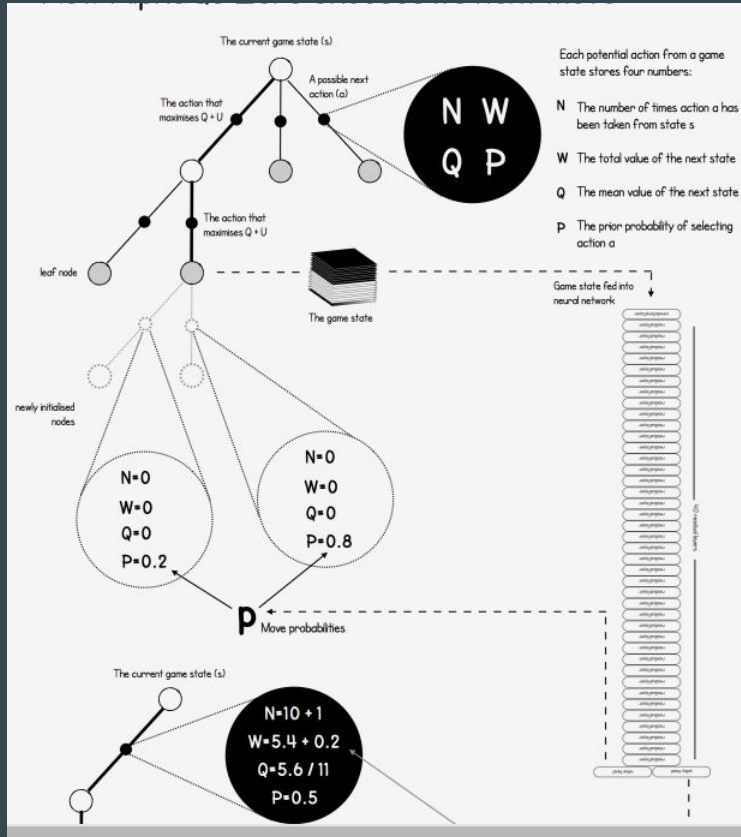
- 1) To accelerate the speed of training of the deep networks
- 2) Instead of widen the network, increasing depth of the network results in less extra parameters
- 3) Reducing the effect of Vanishing Gradient Problem
- 4) Obtaining higher accuracy in network performance especially in Image Classification

Residual Neural Networks

2 output :



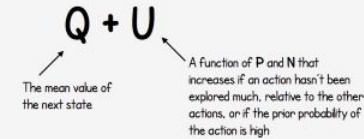
Neural network in MCTS



First, run the following simulation
1,600 times...

Start at the root node of the tree (the current game state)

1. Choose the action that maximises...



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

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

3. Backup previous edges

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

How Residual neural networks Learn ?

1600 simulations for a move

25000 games against itself => Training site

1000 training loops : 2048 positions taken randomly in 500 000 games => Train

400 games against the latest best neural network => Evaluate

if the new neural network is better than the previous one it becomes the best

How many GPU ?

Learning :

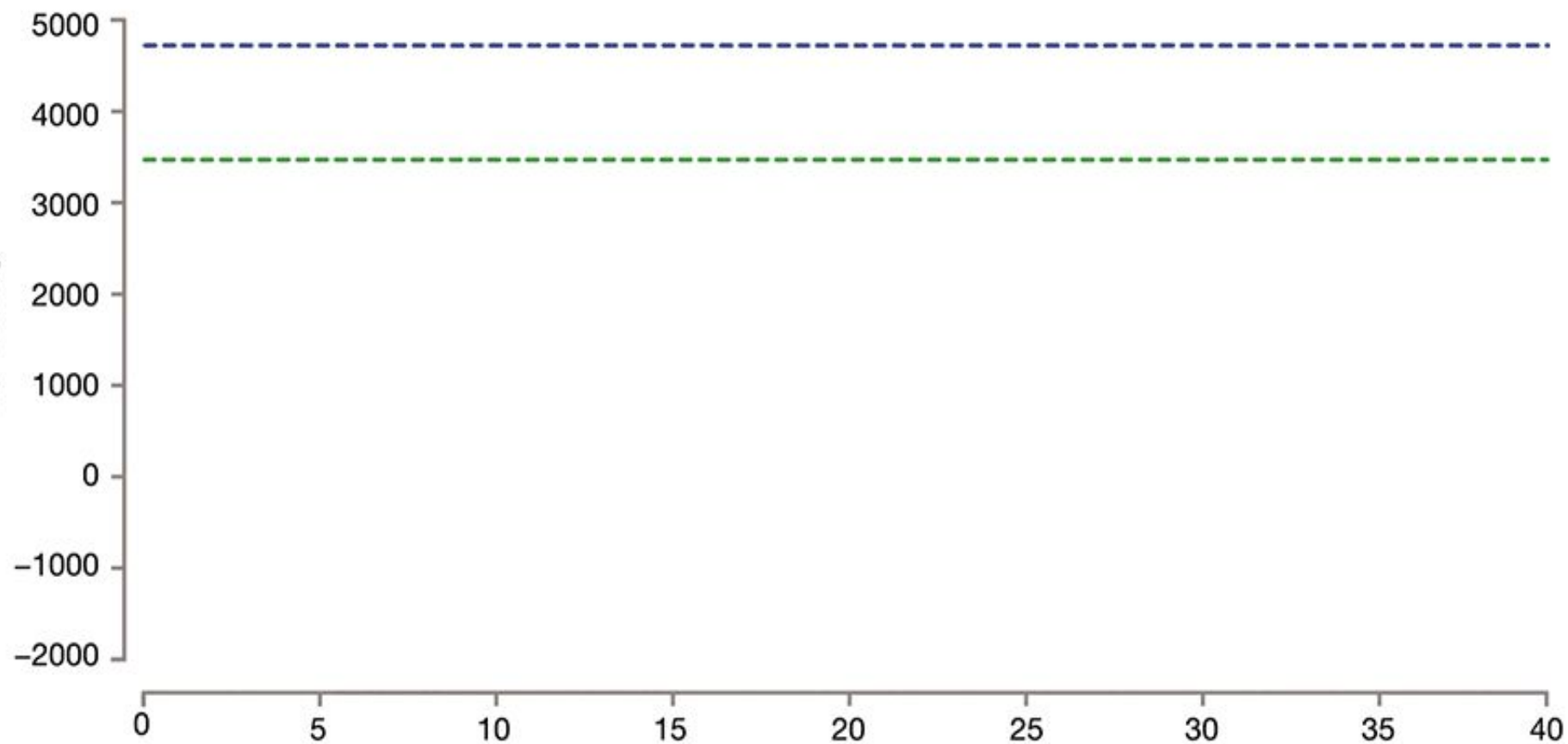
19 CPU

64 GPU

Inference :

4 TPU

Elo Rating



AlphaGo Zero 40 blocks

AlphaGo Lee

AlphaGo Master

