How AlphaGo Works

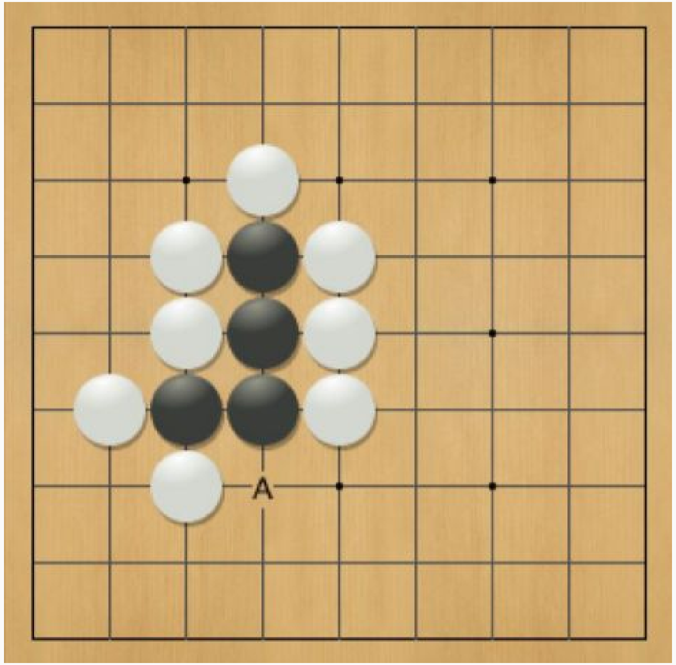
Yuu Sakaguchi

How to Play Go

Played on a 19 x 19 square grid board.

Black and white stones.

Points awarded for surrounding empty space.



Why is Go Hard to Compute?

Why is Go Hard to Compute?

Search space is huge

After the first two moves of a Chess game, there are 400 possible next moves. In Go, there are close to 130,000.

Complexity : 250^{150} possible sequenses

Match against Lee Sedol

AlphaGo played professional Go player Lee Sedol, ranked 9-dan, one of the best players at Go in March 2016.

AlphaGo won by 4 - 1.

How did AlphaGo solve it?

How did AlphaGo solve it?

Ideas

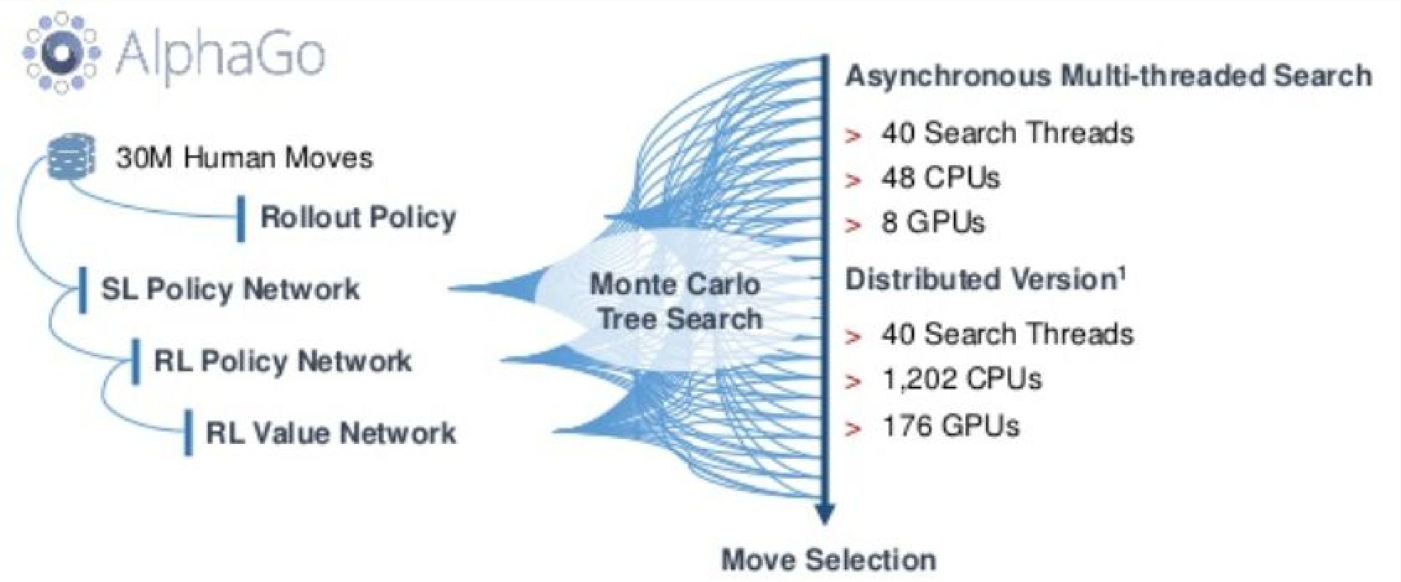
- Deep Learning
- Convolutional Neural Network
- Supervised Learning
- Reinforcement Learning
- Monte-Carlo Tree Search

How did AlphaGo solve it?

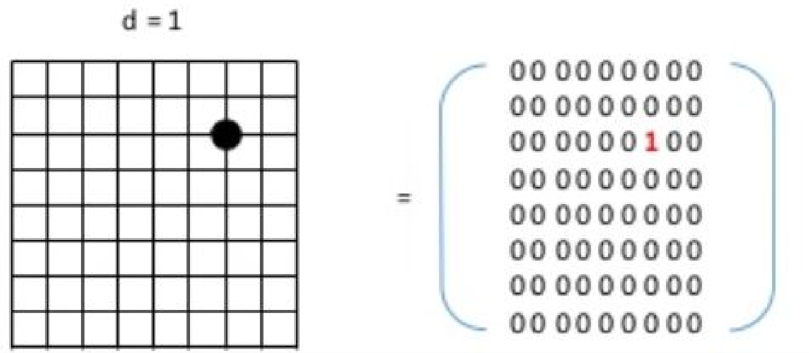
Strategies

Knowledge learned from human expert games and self-play.

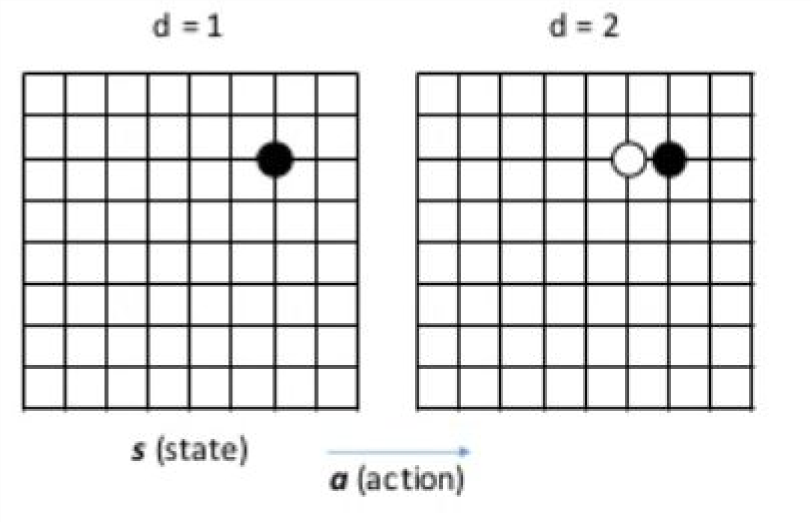
Monte-Carlo search guided by **policy and value networks**.



Computing Go

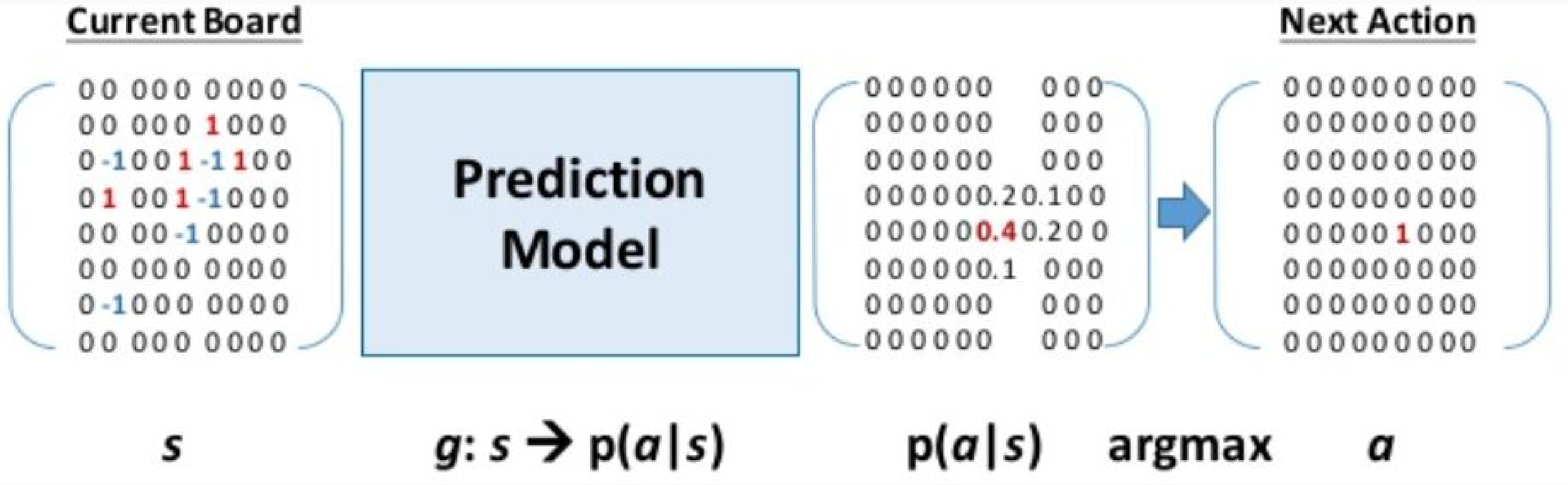


AlphaGo sees the board as One-hot matrix.

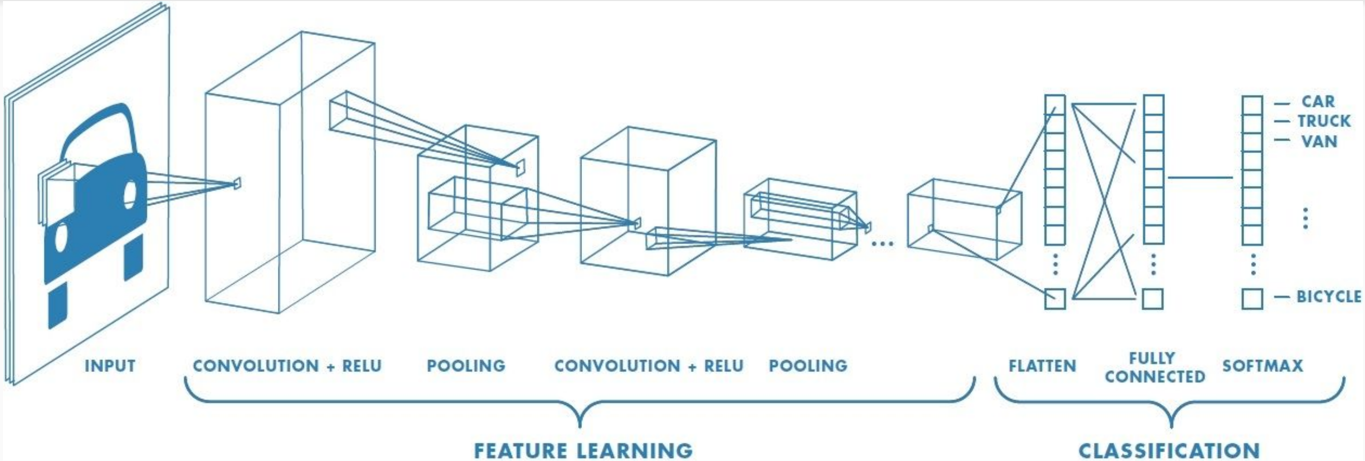


Give a state **s**, pick the best action **a**.

Computing Go



Convolutional Neural Network (CNN)



The hidden layers of a CNN consist of convolutional layers, pooling layers, fully connected layers and normalization layers. There are many applications such as image and video recognition, recommender systems and natural language processing.

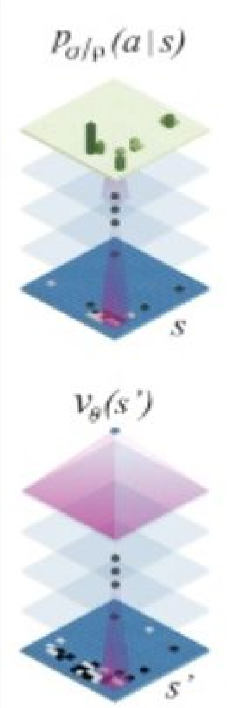
Types of Neural Networks

1.Policy Network

Breath Reduction. Finds the probability of the next move, and reduces the action candidates.

2. Value Network

Depth Reduction. Evaluates the value of the board at each state.



Types of Neural Networks

Policy Network
$P(a|s)$
$\Sigma_a P(a|s) = 1$

Name	Network	Data Set	Speed
P_π P_ζ	Linear Softmax	8M from expert players	CPU 2µs
P_σ P_ρ	Deep Network	28M from expert players	GPU 2ms

Value Network
$V\theta(S)$
[-1,1]

V_θ	Deep Network	30M random states from P_σ + 160M probabilities from P_ρ	GPU 2ms

Types of Neural Networks

Policy Network

- Input layer : 19 x 19 x 48
- Hidden layers : 19 x 19 x k x (12 layers)
- Output layer : 19 x 19 P(a|s)

Value Network

- Input layer : 19 x 19 x 49
- Hidden layer : 19 x 19 x 192 x (12 layers) + 19 x 19 x (1 layer) + 256 x (1 layer)
- Output layer : 1 output V(S)

Types of Networks

Extended Data Table 2 | Input features for neural networks

Feature	# of planes	Description	
Stone colour	3	Player stone / opponent stone / empty	
Ones	1	A constant plane filled with 1	
Turns since	8	How many turns since a move was played	
Liberties	8	Number of liberties (empty adjacent points)	
Capture size	8	How many opponent stones would be captured	
Self-atari size	8	How many of own stones would be captured	1
Liberties after move	8	Number of liberties after this move is played	
Ladder capture	1	Whether a move at this point is a successful ladder capture	
Ladder escape	1	Whether a move at this point is a successful ladder escape	
Sensibleness	1	Whether a move is legal and does not fill its own eyes	
Zeros	1	A constant plane filled with 0	
Player color	1	Whether current player is black	2

Feature planes used by the policy network (all but last feature) and value network (all features).

Types of Networks

Policy Network

Input - First hidden layer :

- 2x2 padding
- 5x5 convolutional by 5 filters
- ReLU function

n - n+1 hidden layer

- 21x21 padding
- 3x3 convolutional by 3 filters
- ReLU function

12th hidden layer - Output

- 1 output
- Different biases on each place on board
- Softmax function

Types of Networks

Value Network

Input - 12th hidden layer :

Same as policy network.

12th - 13th hidden layer

- 1x1 filter
- ReLU function

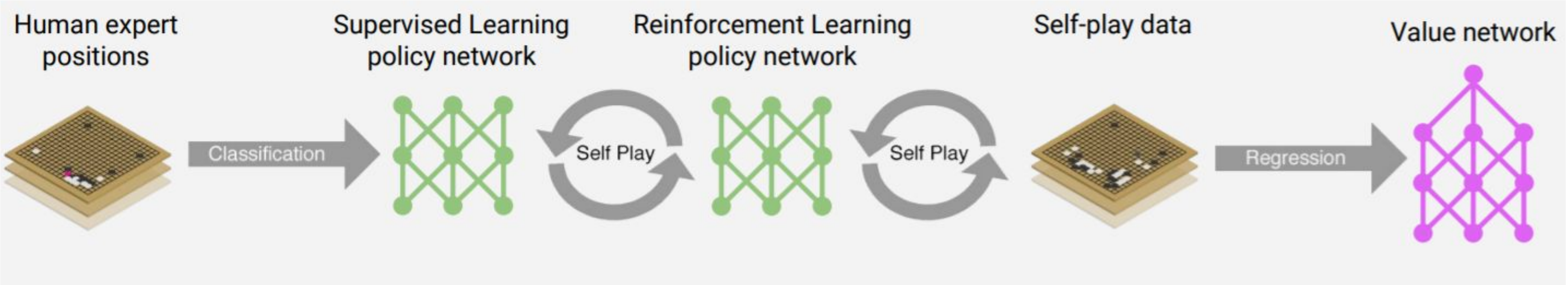
13th - 14th hidden layer

- Fully connected
- ReLU function

14th - output

- Fully connected
- tanh function

Training

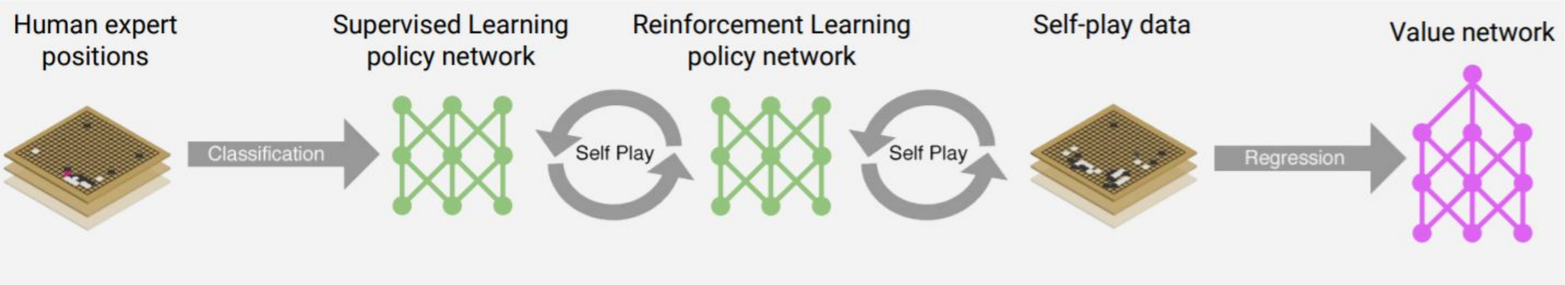


Supervised learning of policy network

4 weeks on 50 GPUs using Google Cloud.

57% accuracy on test data.

Training

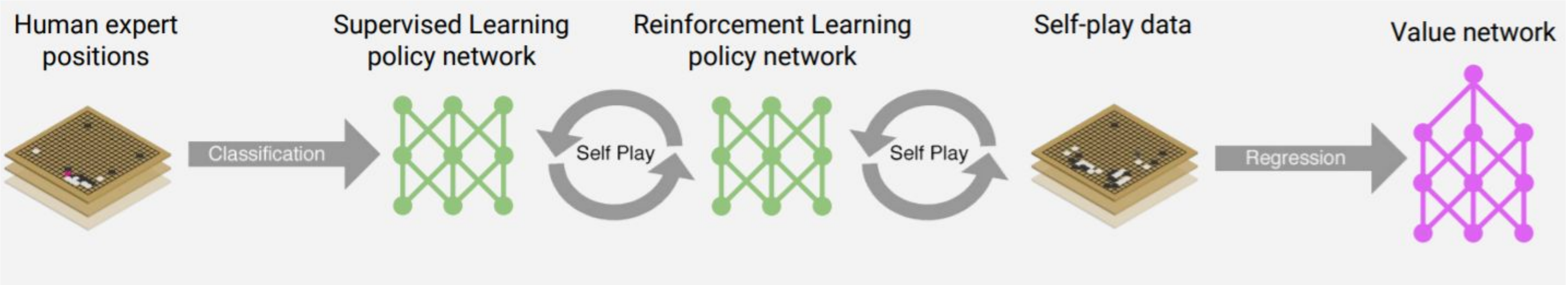


Reinforcement learning of policy network

1 week on 50 GPUs using Google Cloud.

80% against supervised learning.

Training

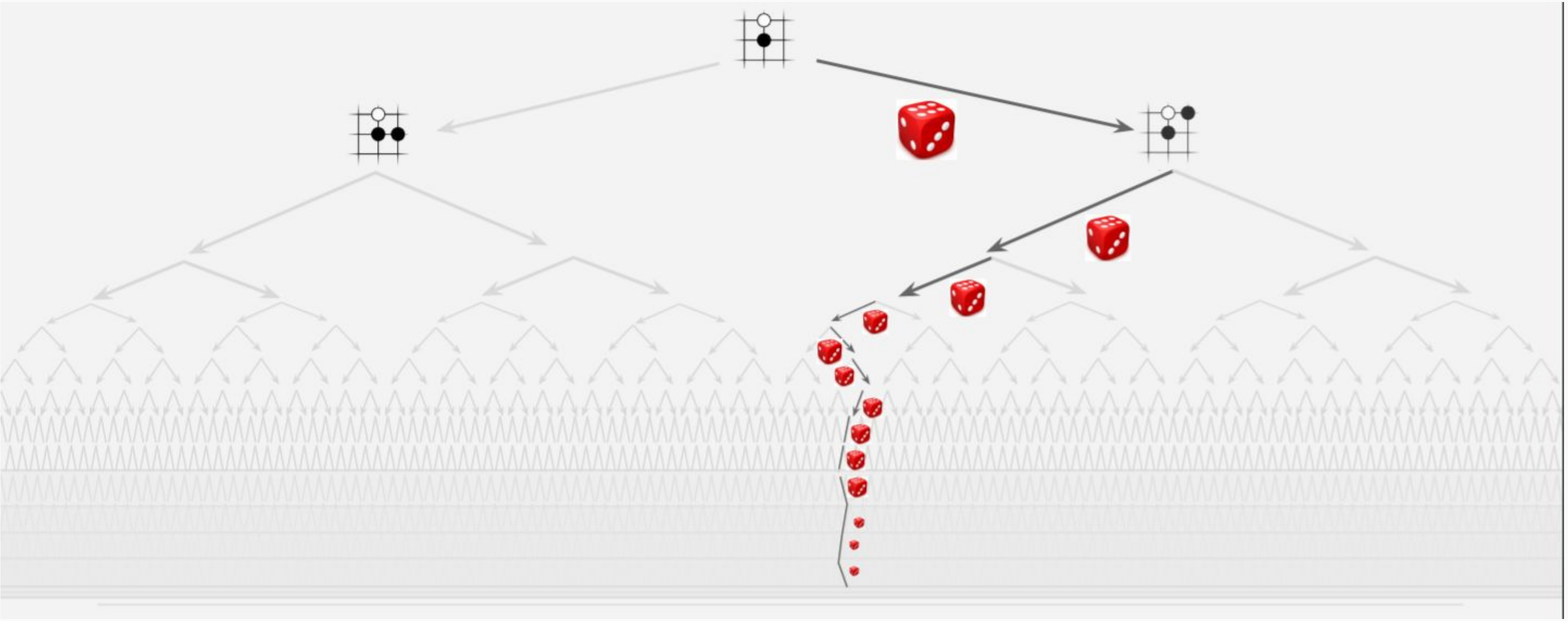


Supervised learning of value network

1 week on 50 GPUs using Google Cloud.

Monte-Carlo Tree Search

Monte-Carlo Tree Search

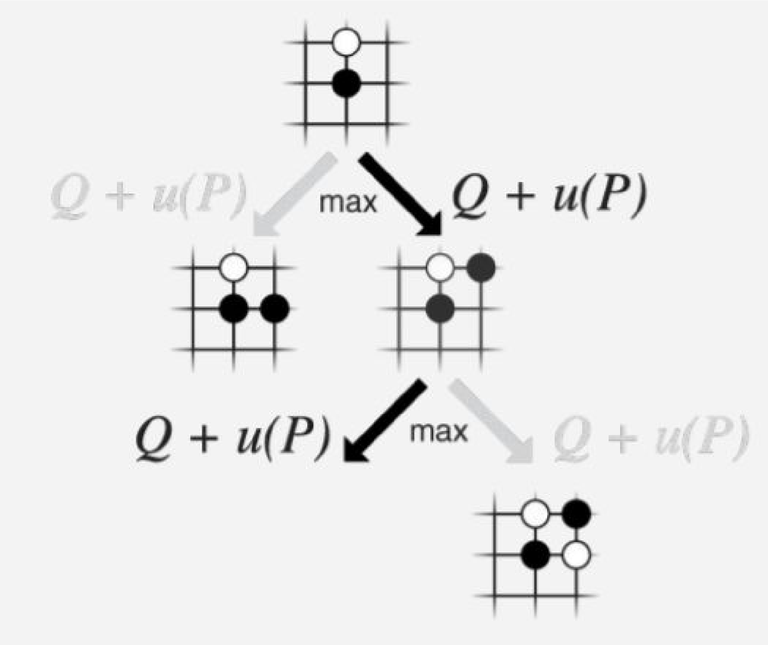


Monte-Carlo Tree Search : selection

P : prior probability

Q : action value

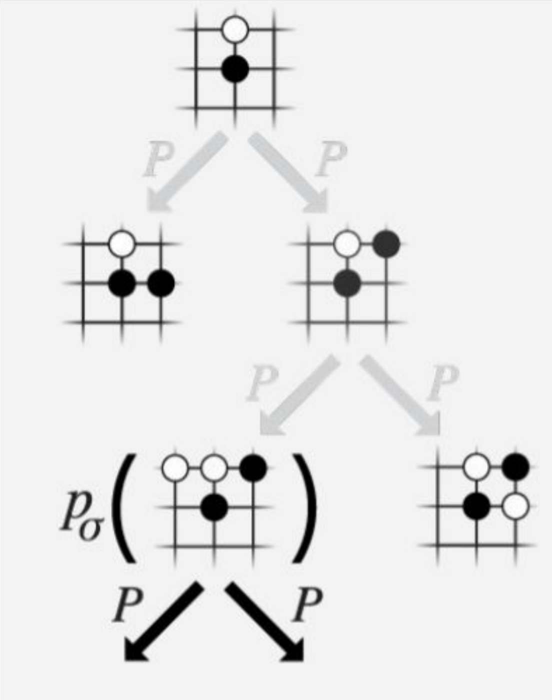
u(P) = P/N



Monte-Carlo Tree Search : expansion

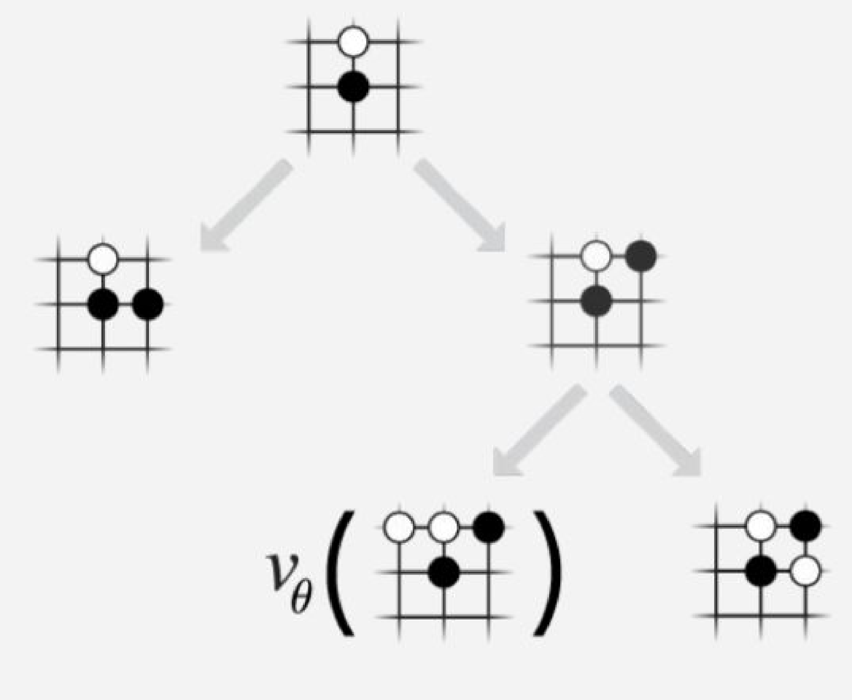
P_σ = policy network

P = prior probability



Monte-Carlo Tree Search : evaluation

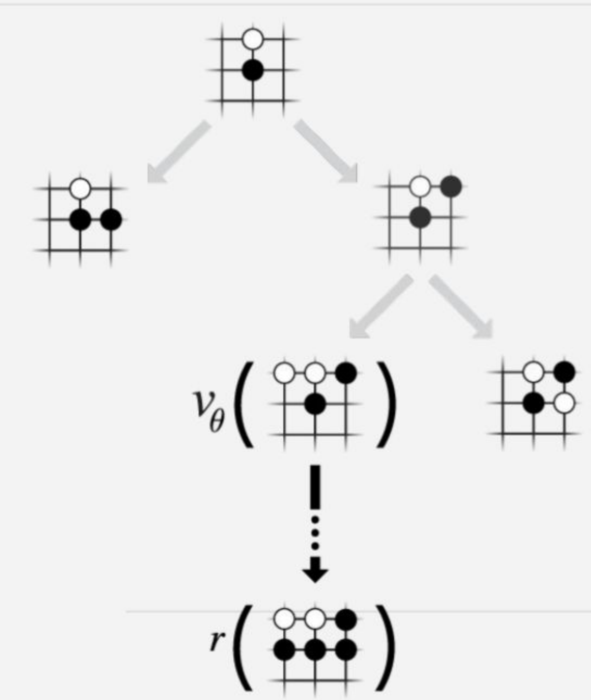
V_θ = value network



Monte-Carlo Tree Search : rollout

V_θ = value network

r = game score

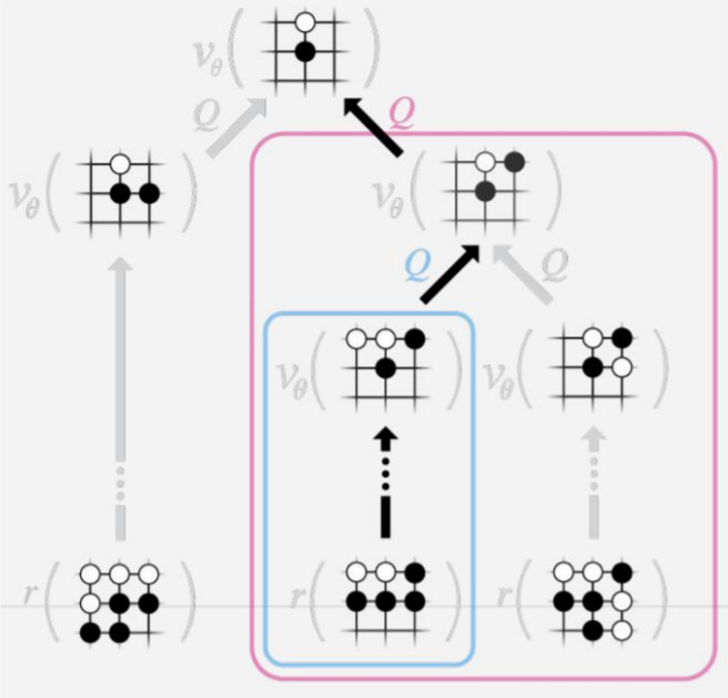


Monte-Carlo Tree Search : backup

Q = action value

V_θ = value network

r = game score



DeepMind - Beyond AlphaGo

Questions?