

March 2016



1
HD

Mainstream Media

tagesthemen 1

1997





2000 “應氏杯”世界電腦圍棋錦標賽
2000 ING CUP WORLD COMPUTER GOE CHAMPIONSHIP

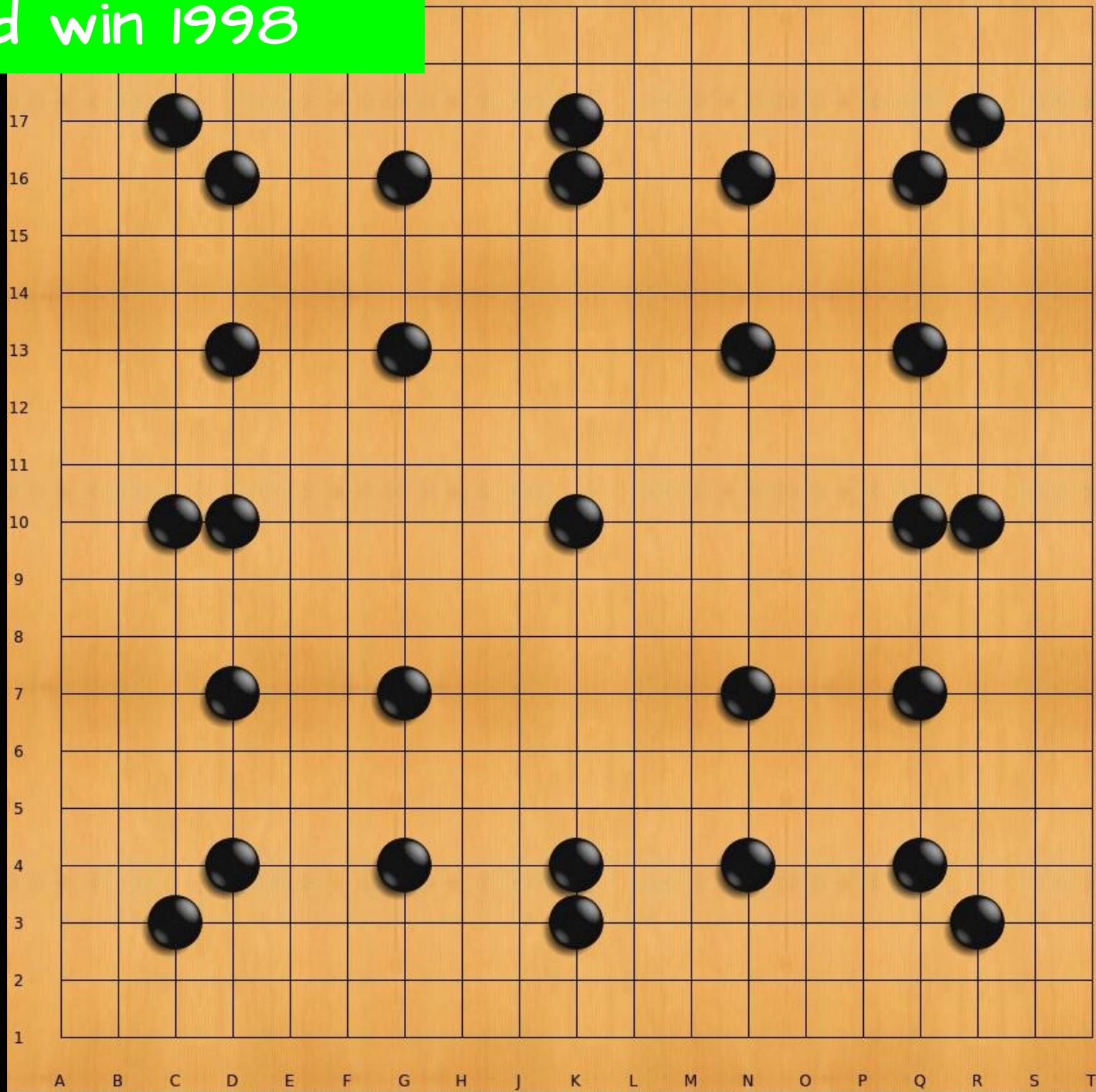
主辦：應昌期圍棋教育基金會
承辦：貴陽市棋類協會
貴陽應氏圍棋活動中心

(1985-2000)



A B C D E F G H J K L M N O P Q R S T

5d win 1998



A photograph of a man with glasses and a grey hoodie sitting at a white desk in a laboratory setting. He is leaning forward, resting his head on his hand. On the desk in front of him is a computer keyboard and a small screen displaying a blue image. To his right is a large computer monitor showing a dark image. Behind him are several pieces of scientific equipment, including a microscope on the left and a circular device with a blue center. The wall behind him is orange and features two black panels with blue and white circular designs. A green rectangular overlay in the top right corner contains the text "October 2015".

October 2015

January 2016

"This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away."

Silver, D. et al., 2016. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), p.484-489.

What did AlphaGo do to beat the strongest human Go player?

Tobias Pfeiffer

@PragTob

pragtob.info



LIEFERY

Go

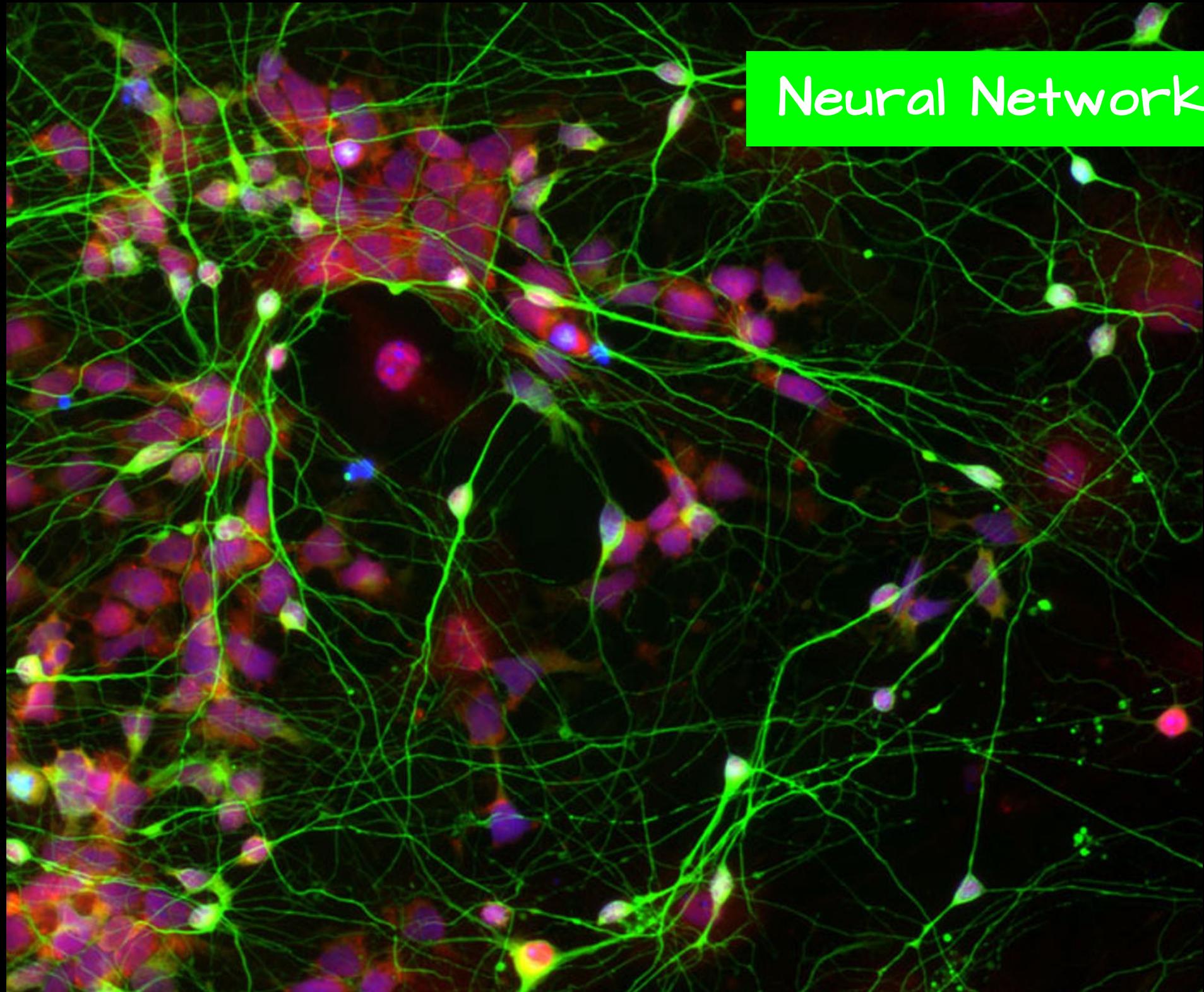




Computational Challenge

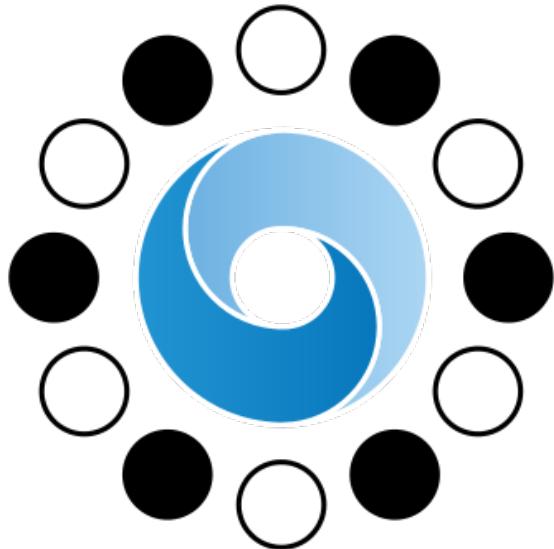
A black and white photograph capturing the iconic harbor of Monte Carlo, Monaco. In the foreground, the dense urban sprawl of the city is visible, with numerous multi-story apartment buildings and office structures. A large marina occupies the middle ground, filled with a multitude of luxury yachts and sailboats. The harbor is framed by a dramatic, rugged coastline featuring several prominent hills and mountains. The sky above is filled with heavy, textured clouds, creating a moody atmosphere.

Monte Carlo Method

A fluorescence microscopy image showing a dense network of neurons. The neurons are stained with several different colors: green, purple, red, and yellow. The green staining appears as thin, branching processes. The purple staining is concentrated in the cell bodies of many neurons. Red and yellow staining are also visible in some neurons, often appearing near the cell bodies or at the ends of processes.

Neural Networks

Revolution with Neural Networks



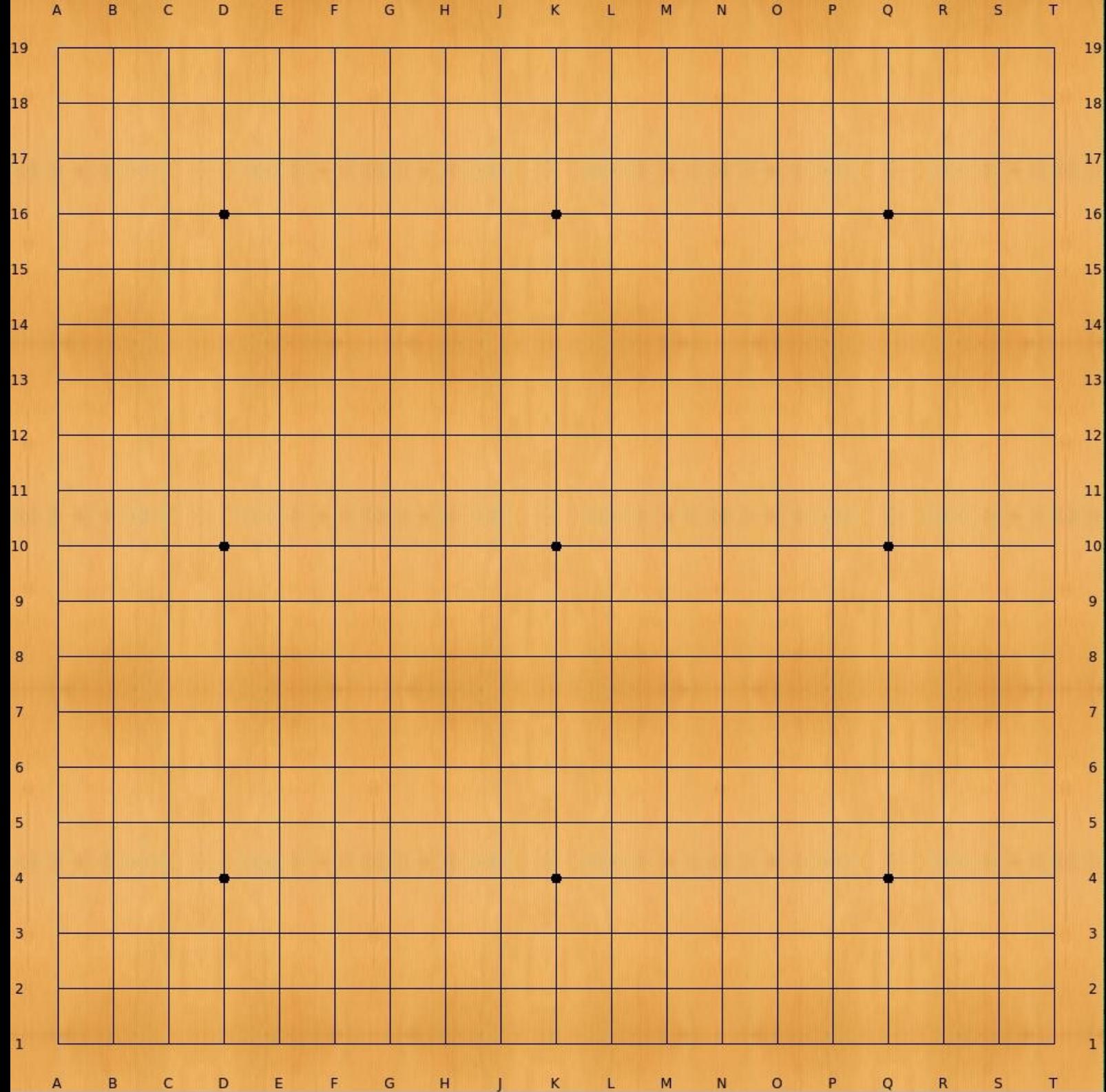
AlphaGo

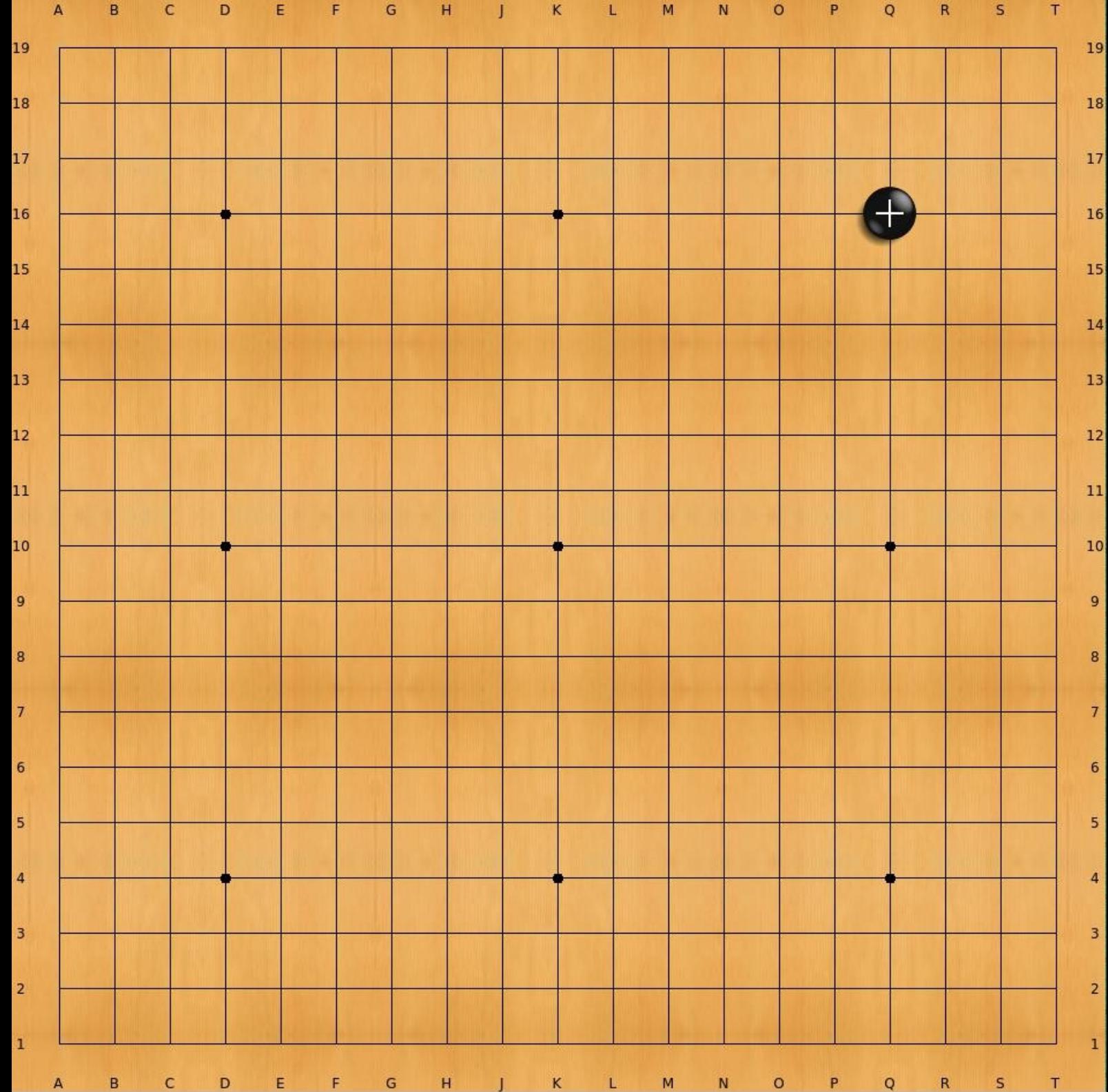
What did we learn?

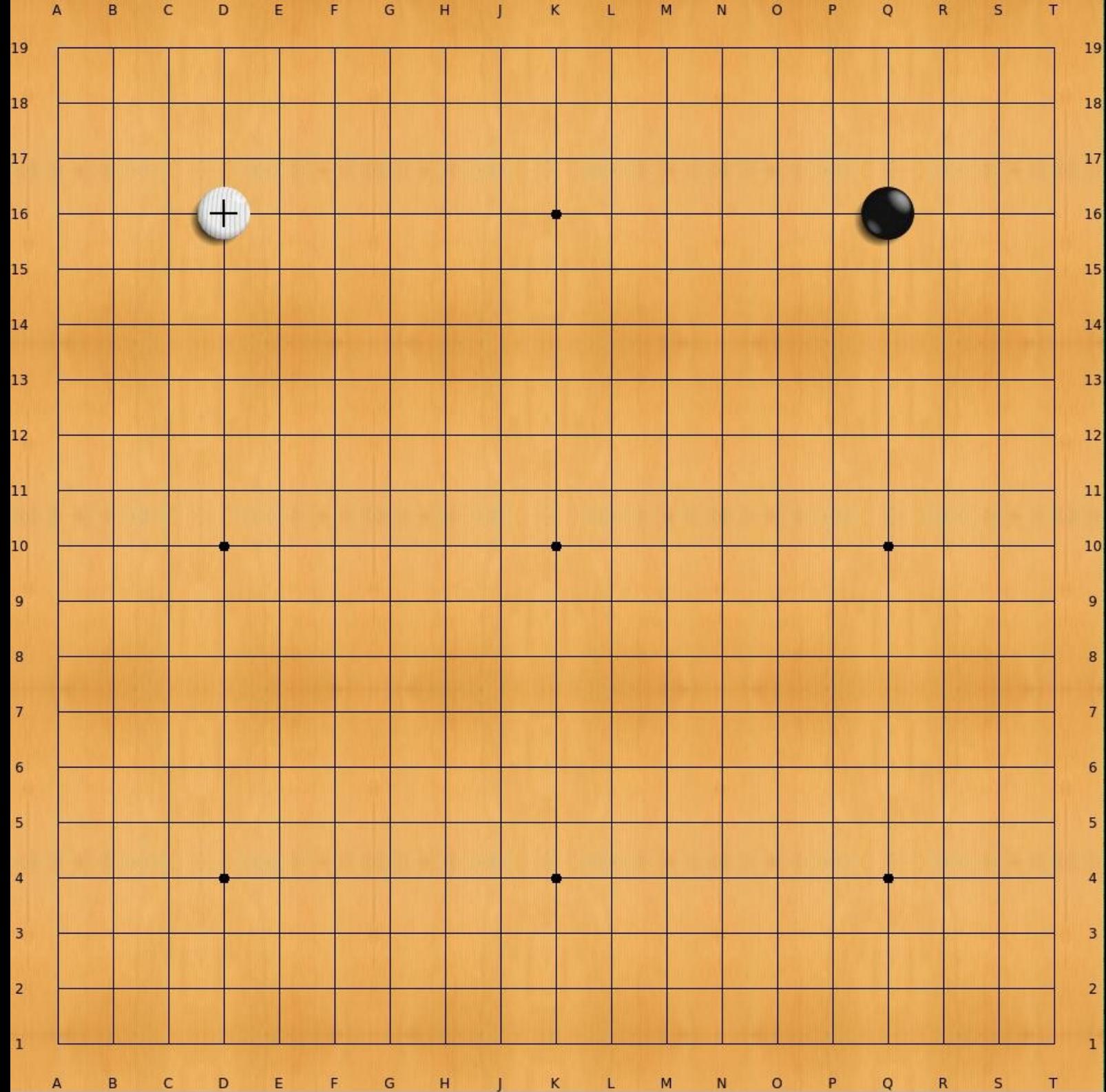


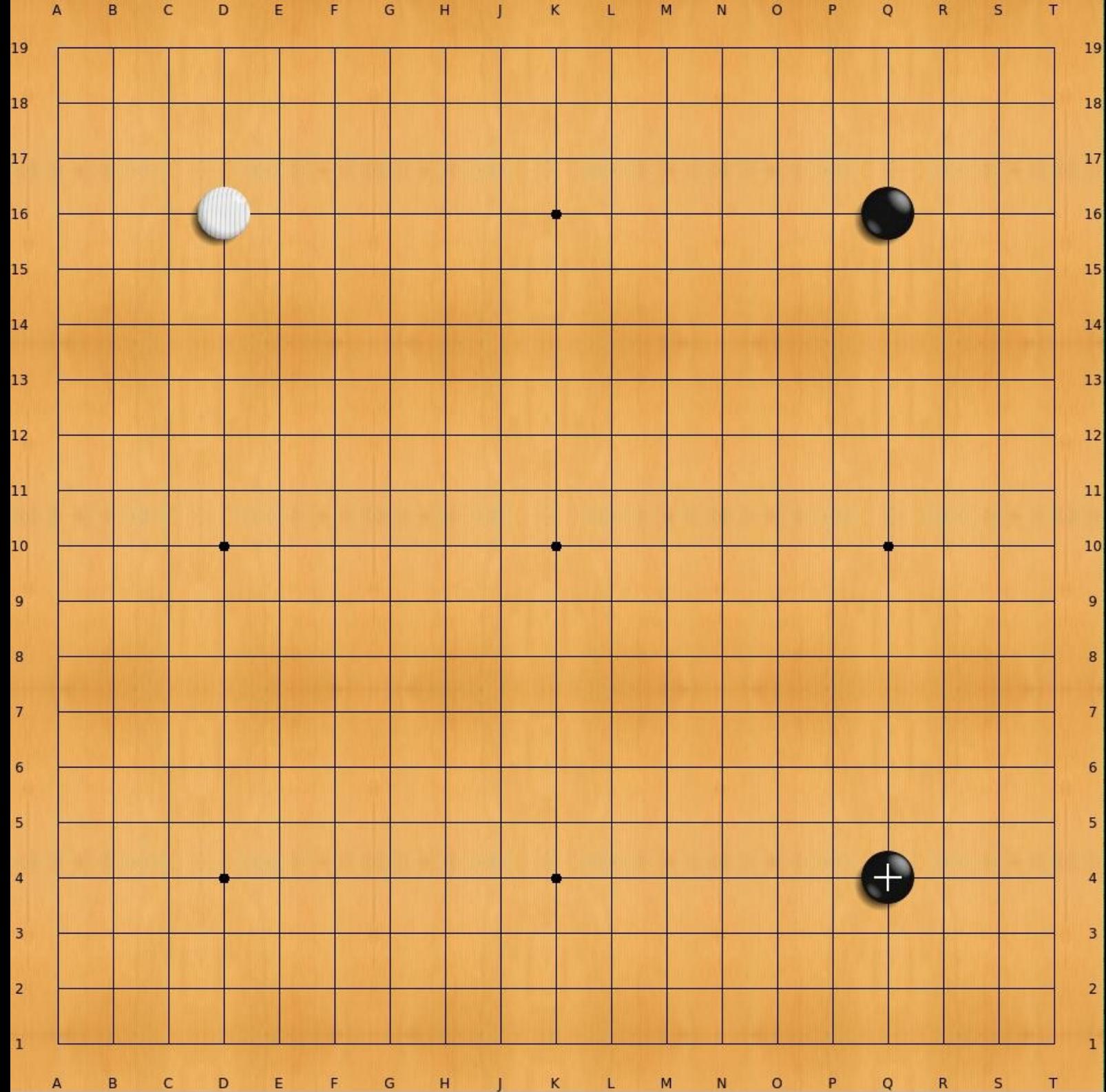
Go

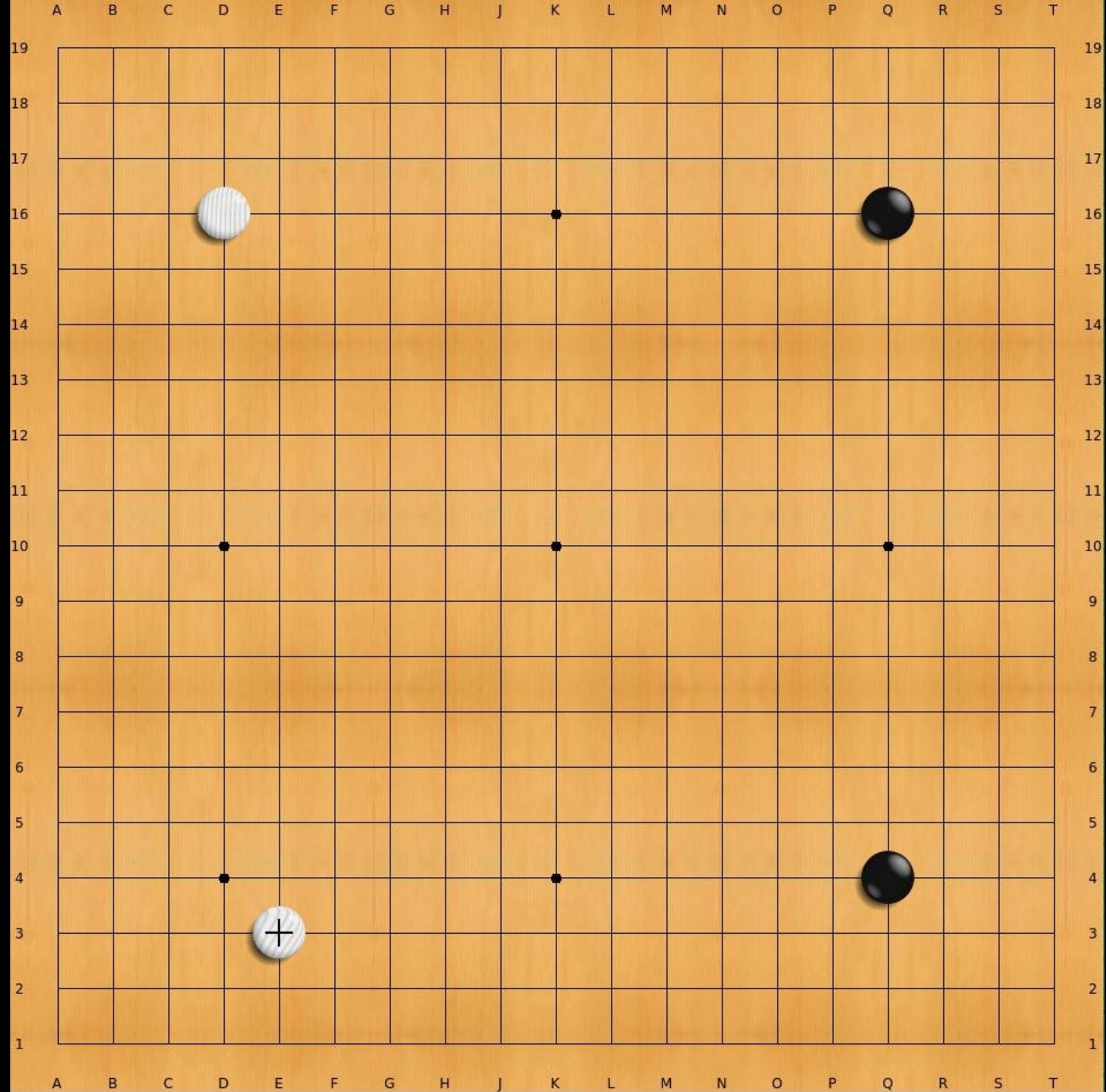


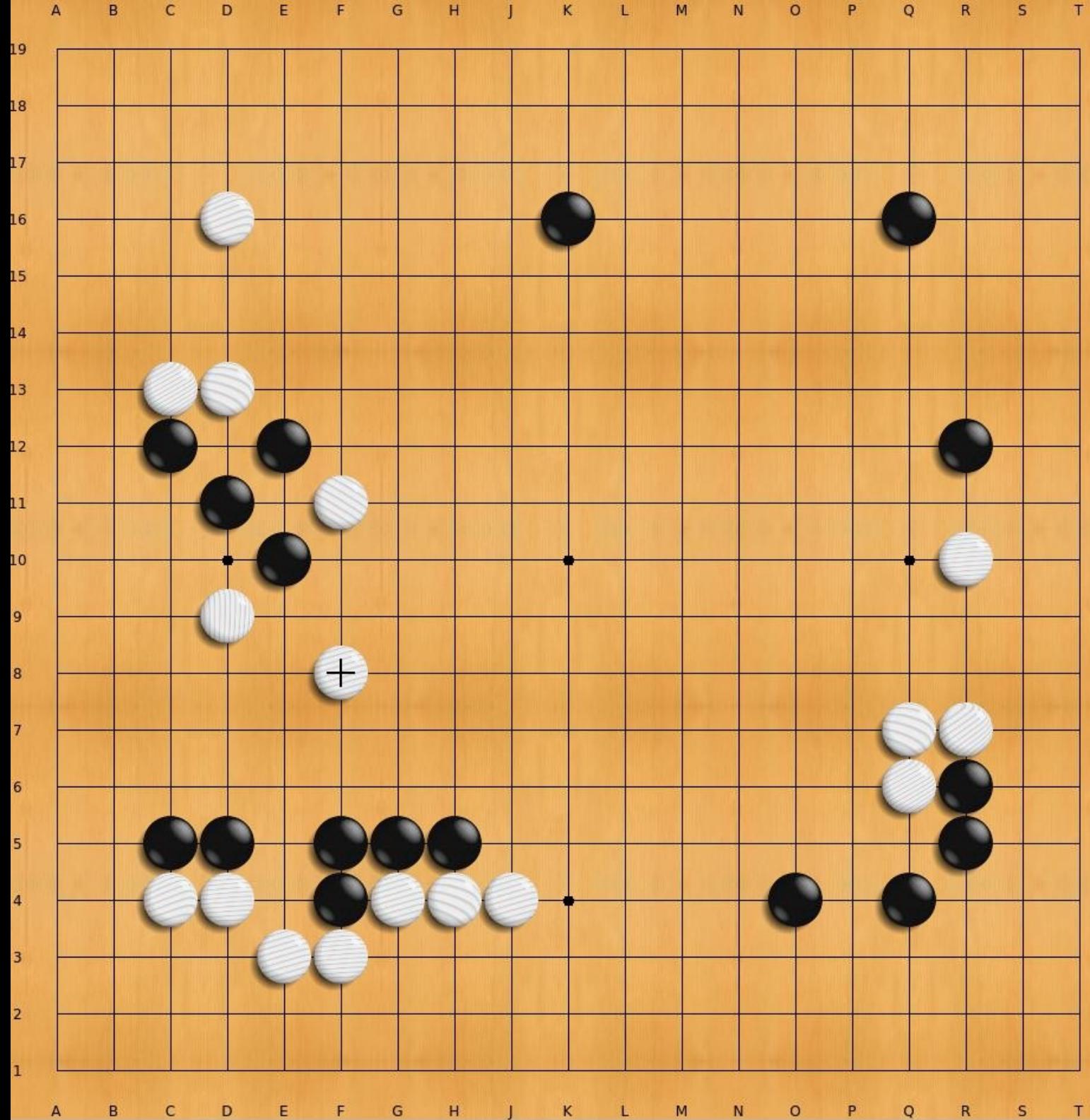


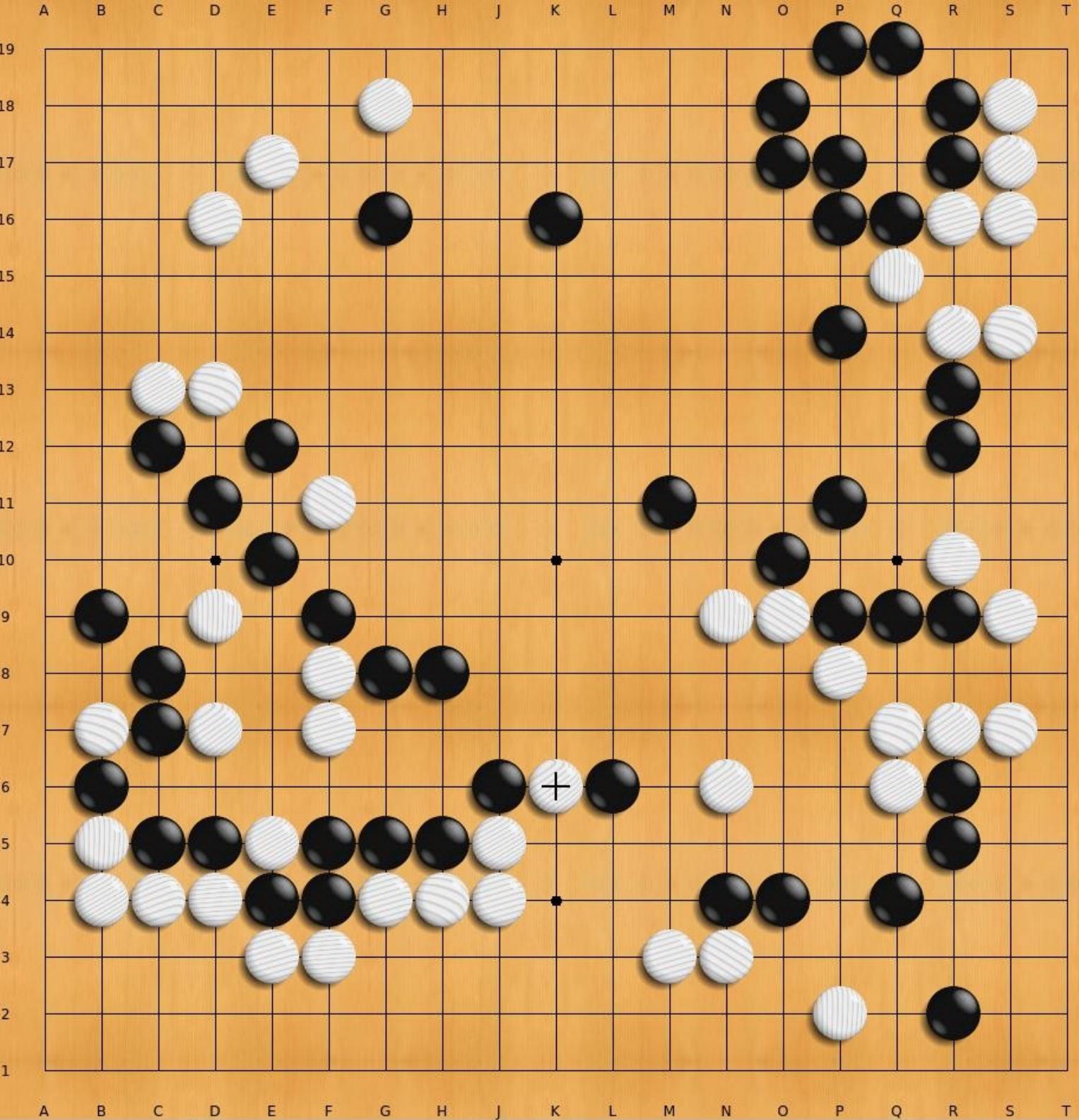




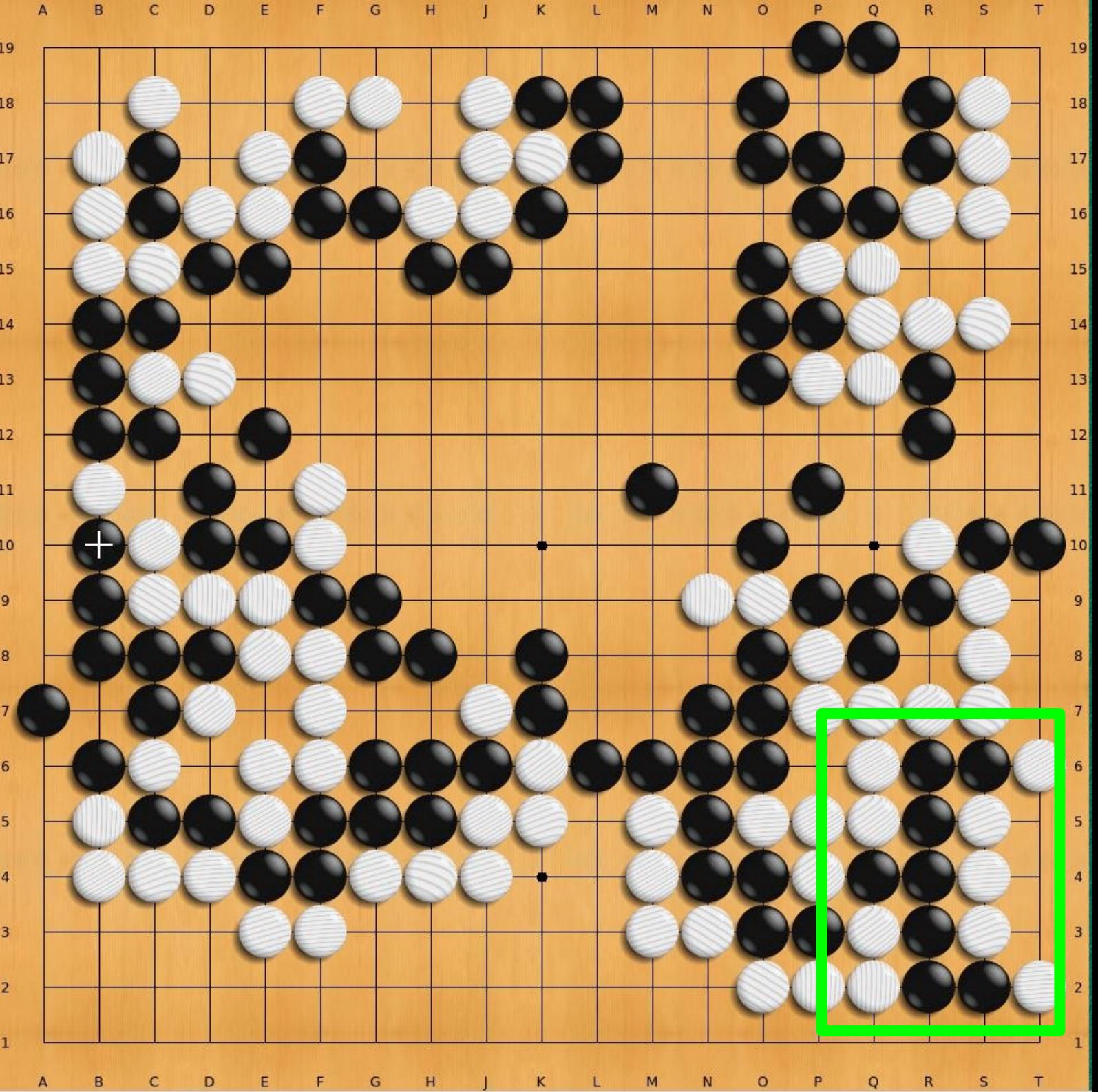


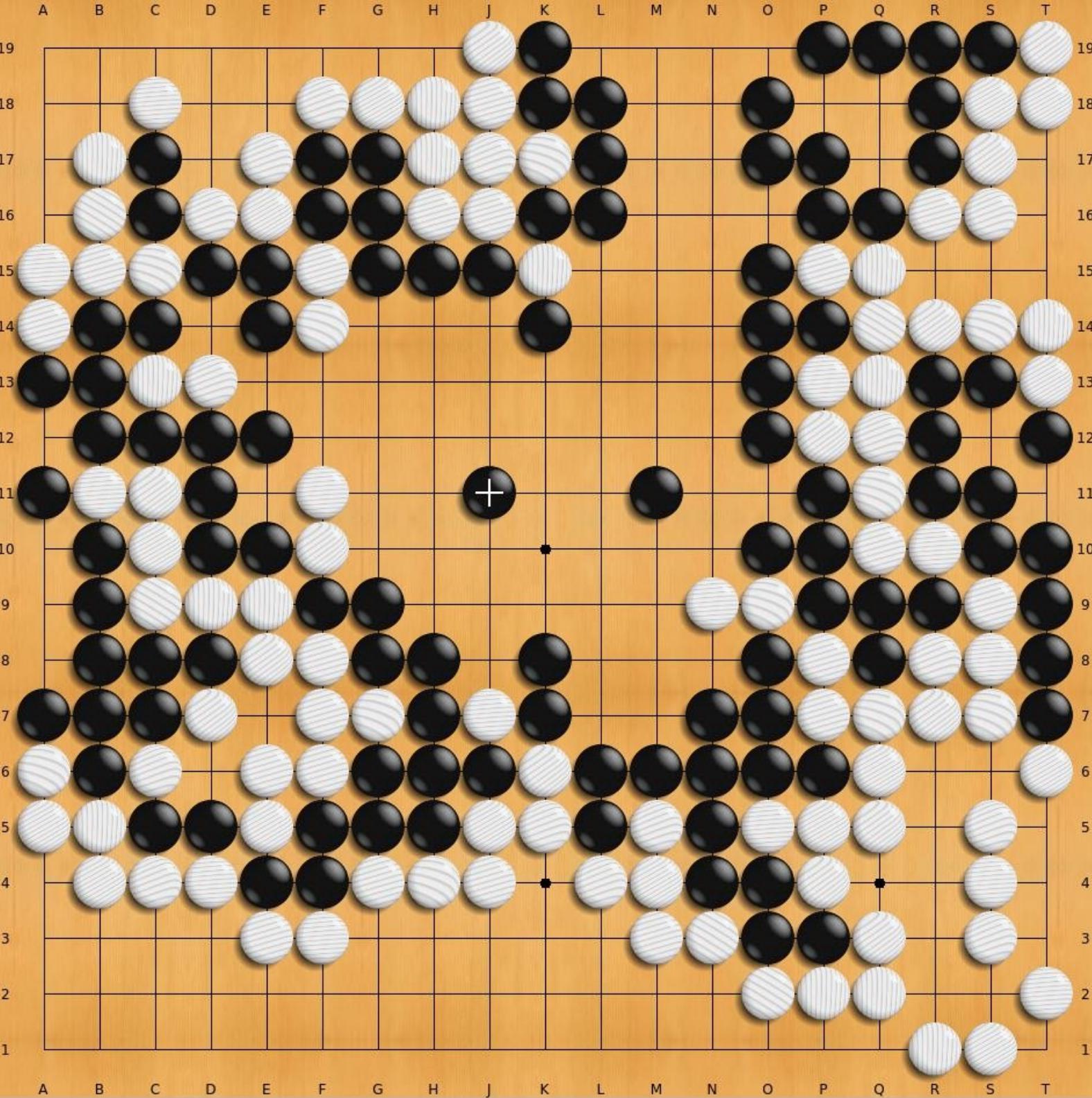


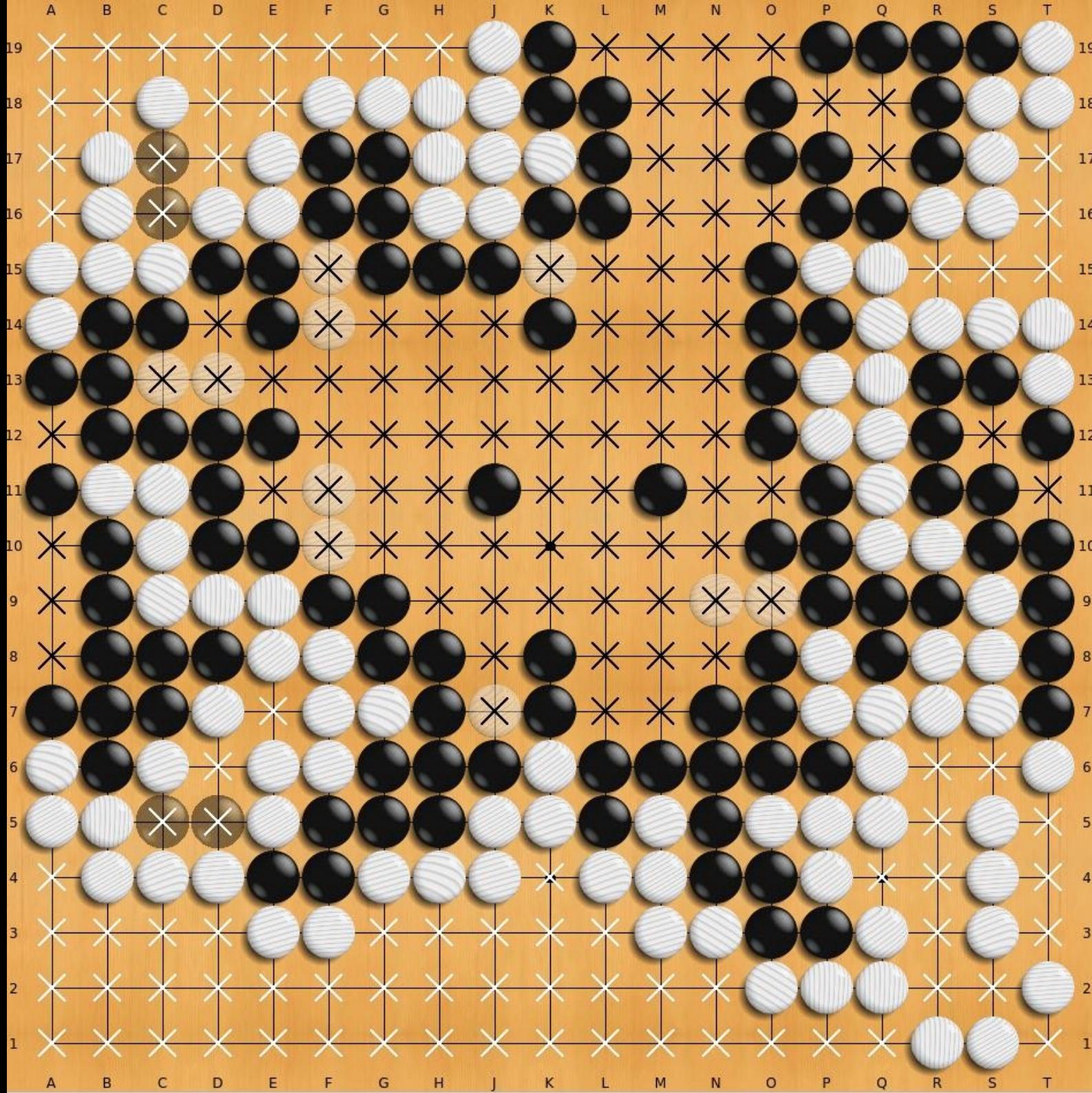












Computational Challenge

Go vs. Chess

Complex vs. **Complicated**

“While the Baroque rules of *chess* could only have been created by humans, the rules of *go* are so elegant, organic, and rigorously logical that if intelligent life forms exist elsewhere in the universe, they almost certainly play *go*.¹¹

Edward Lasker (chess grandmaster)

Larger board
19x19 vs. **8x8**

Almost every move is **legal**

Average branching factor:
250 vs 35

State Space Complexity:

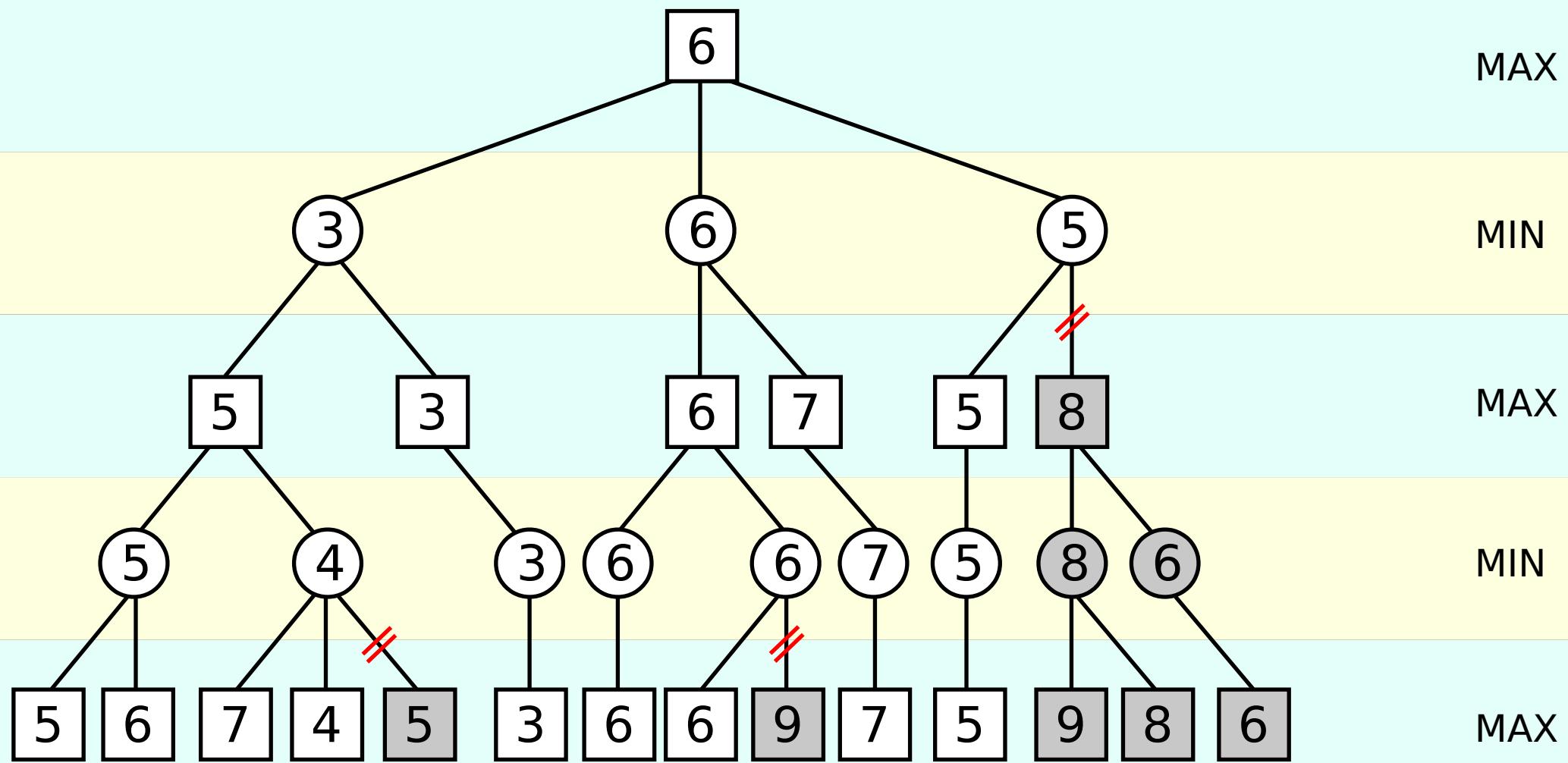
10^{171} vs 10^{47}

10^{80}

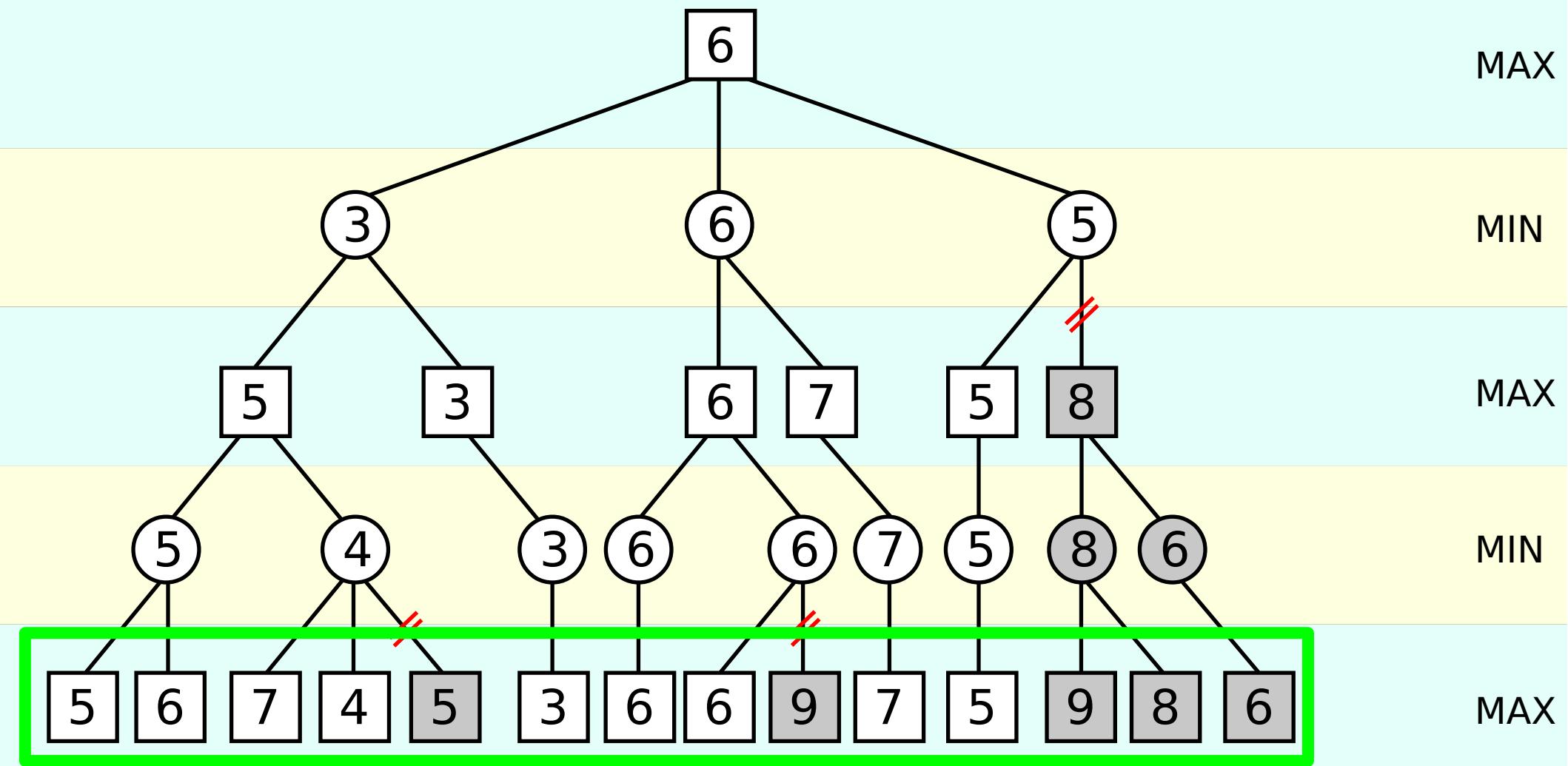


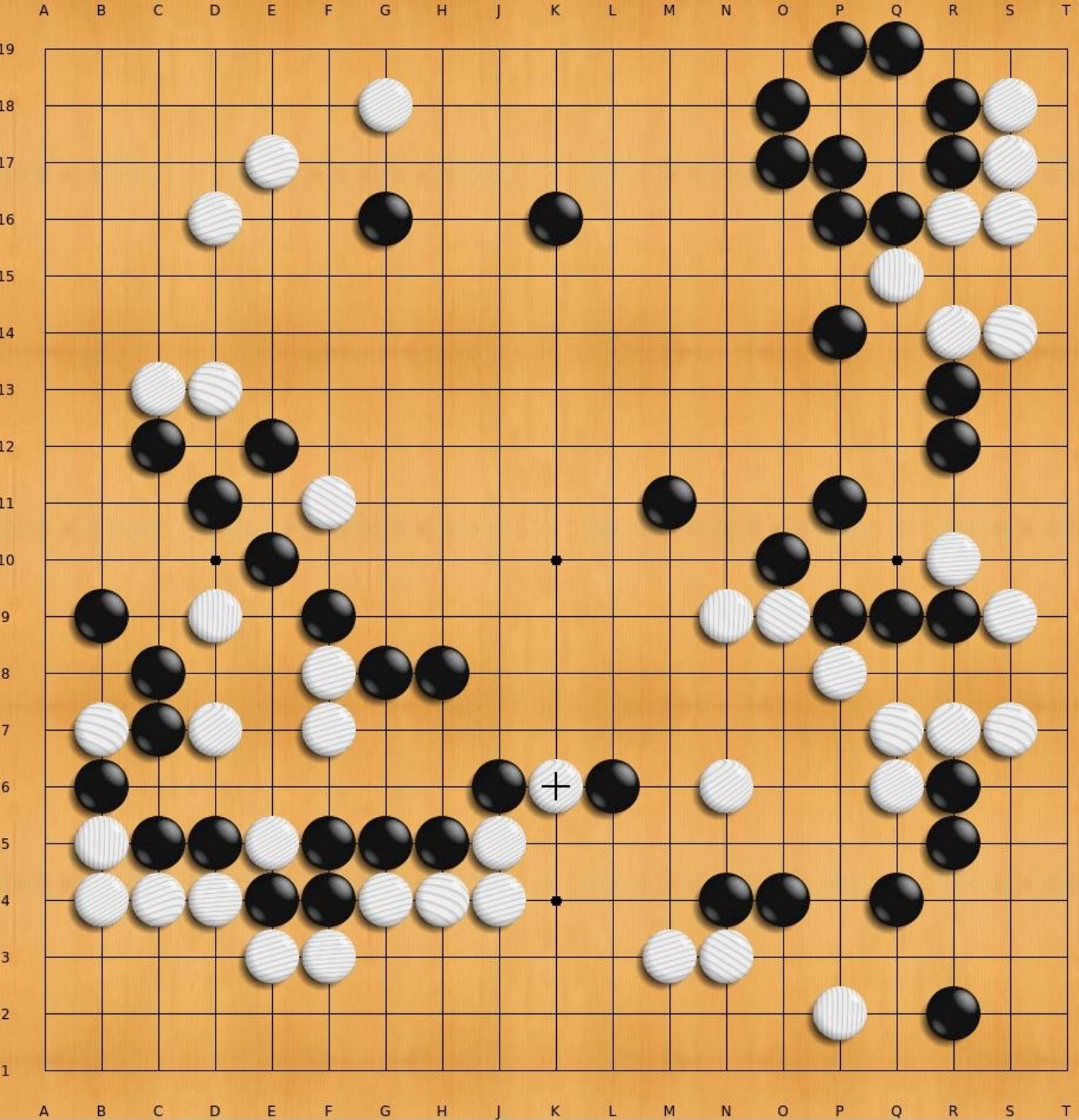
Global impact of moves

Traditional Search



Evaluation Function





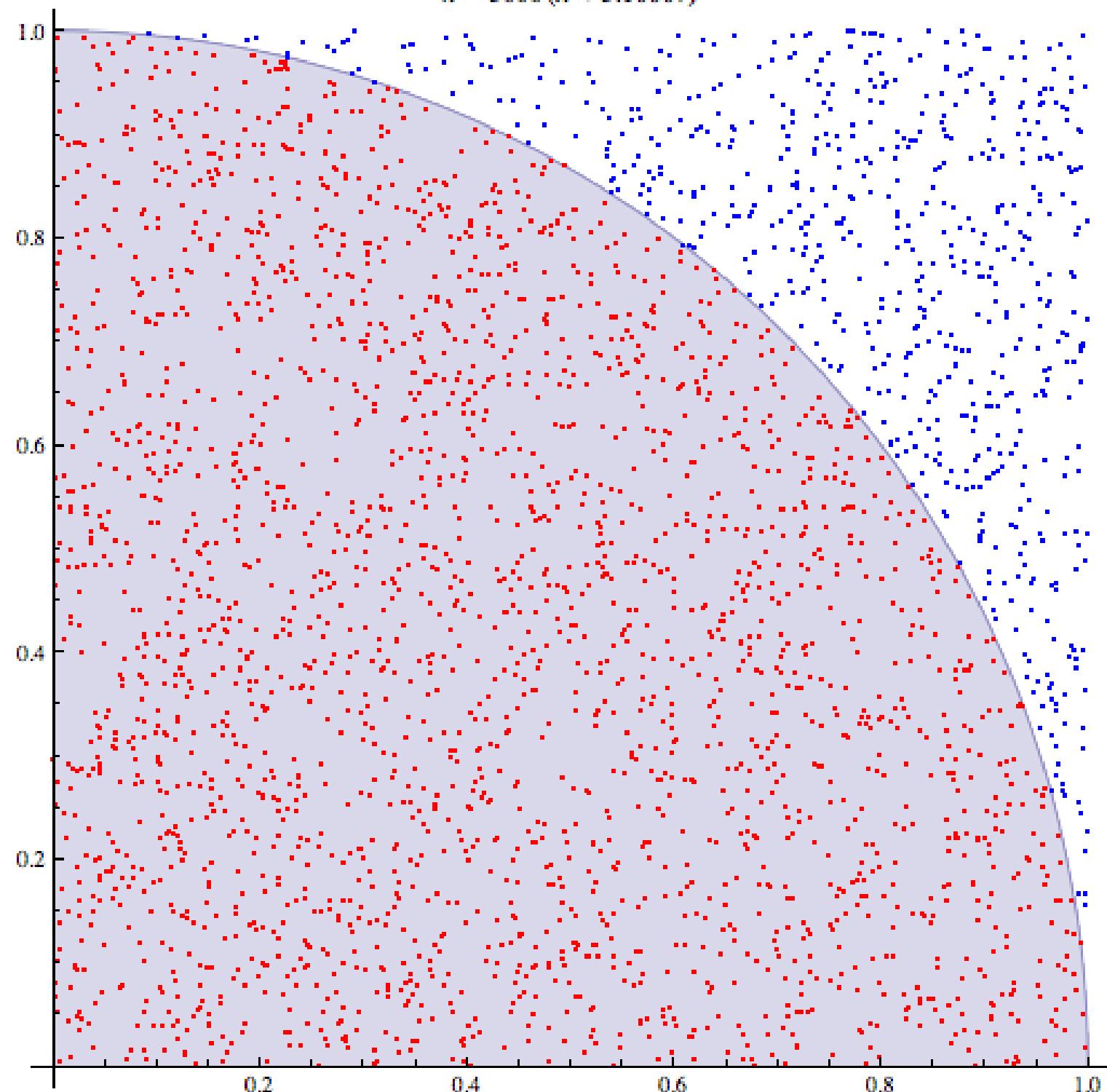
A black and white photograph capturing the iconic harbor of Monte Carlo, Monaco. In the foreground, numerous luxury yachts and sailboats are docked in the harbor. Beyond the water, the dense urban sprawl of Monaco is visible, characterized by its dense collection of modern and classical architecture. A prominent feature is a tall, multi-story residential or office building situated on a hillside. The background is dominated by the rugged, rocky mountains that rise behind the city, their peaks partially obscured by a dramatic, cloudy sky.

Monte Carlo Method

What is Pi?

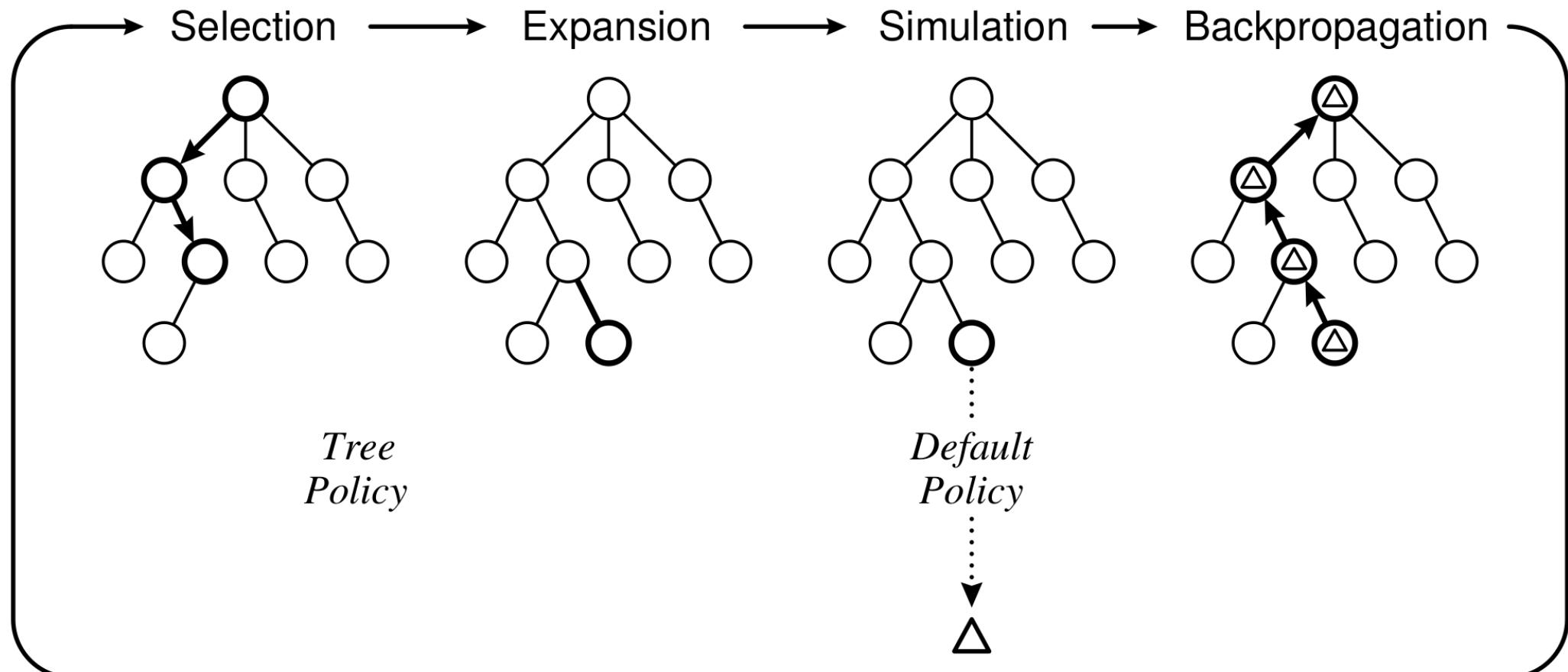
How do you determine Pi?

$n = 3000 (\pi \approx 3.16667)$

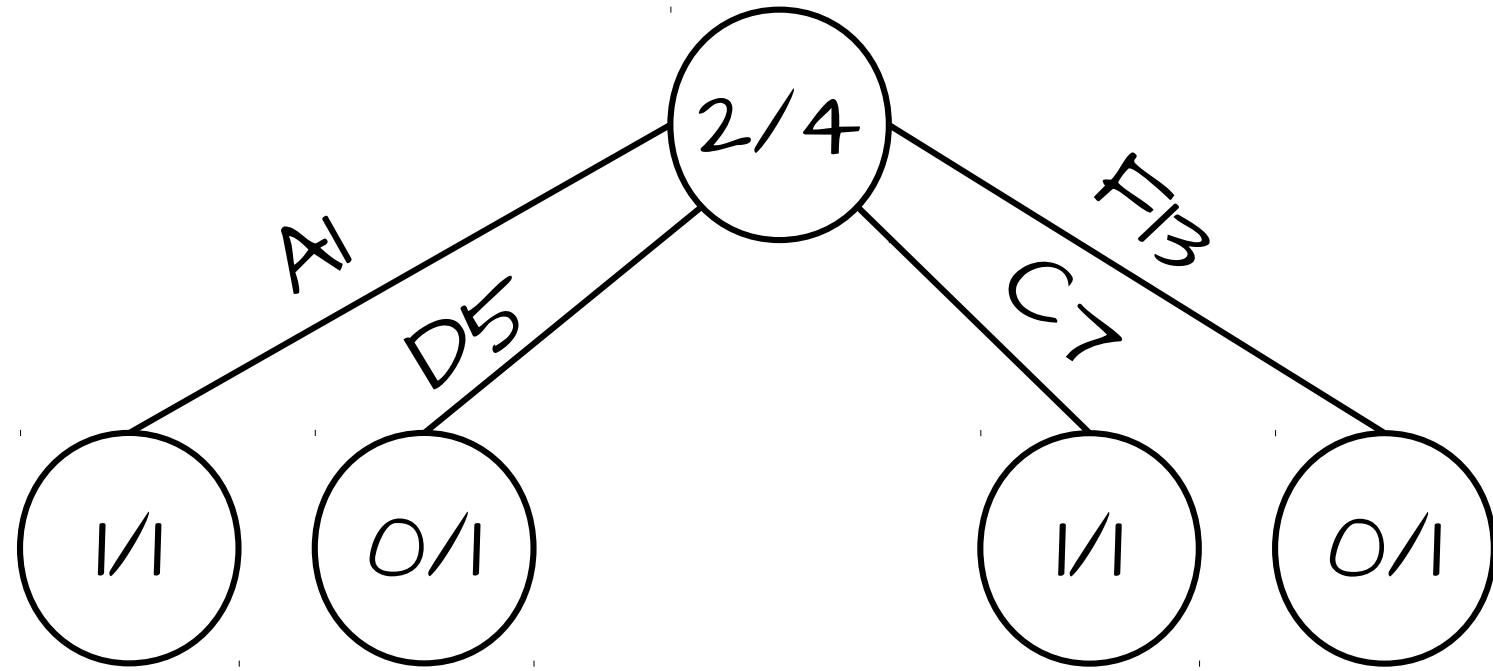


2006

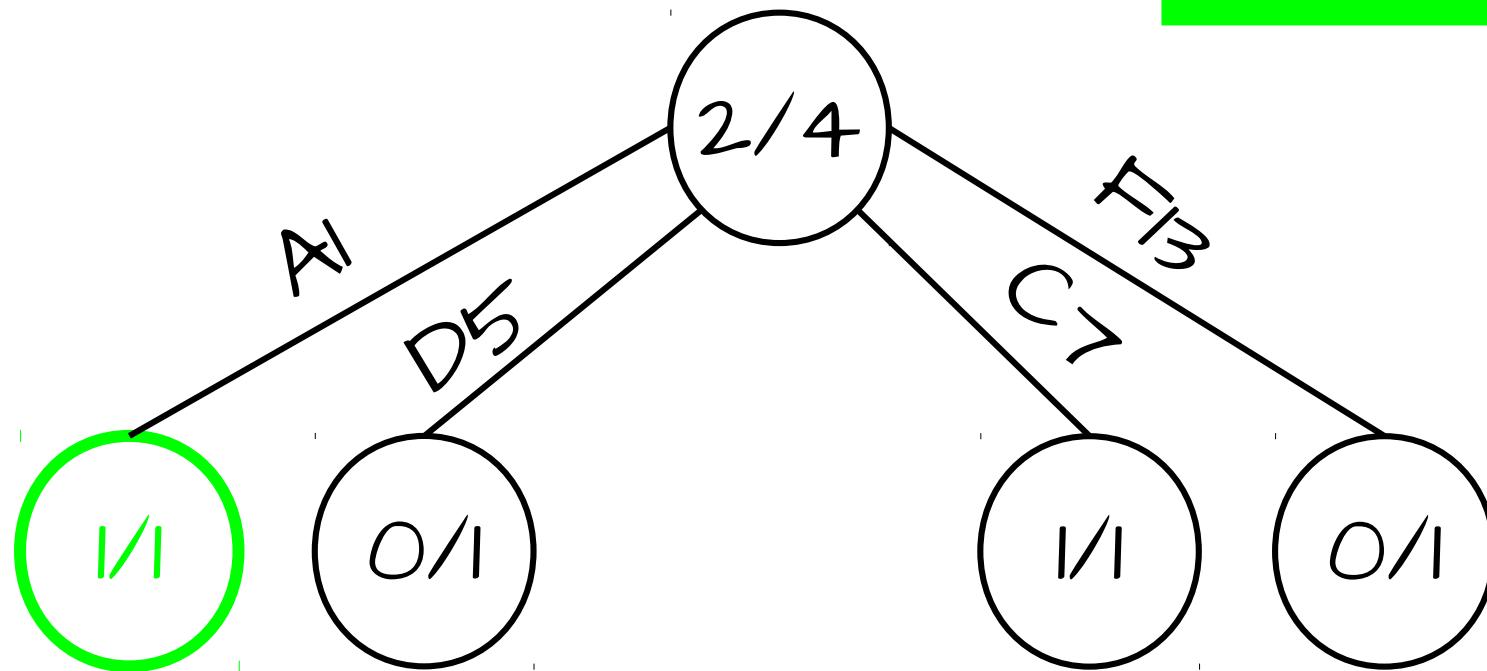




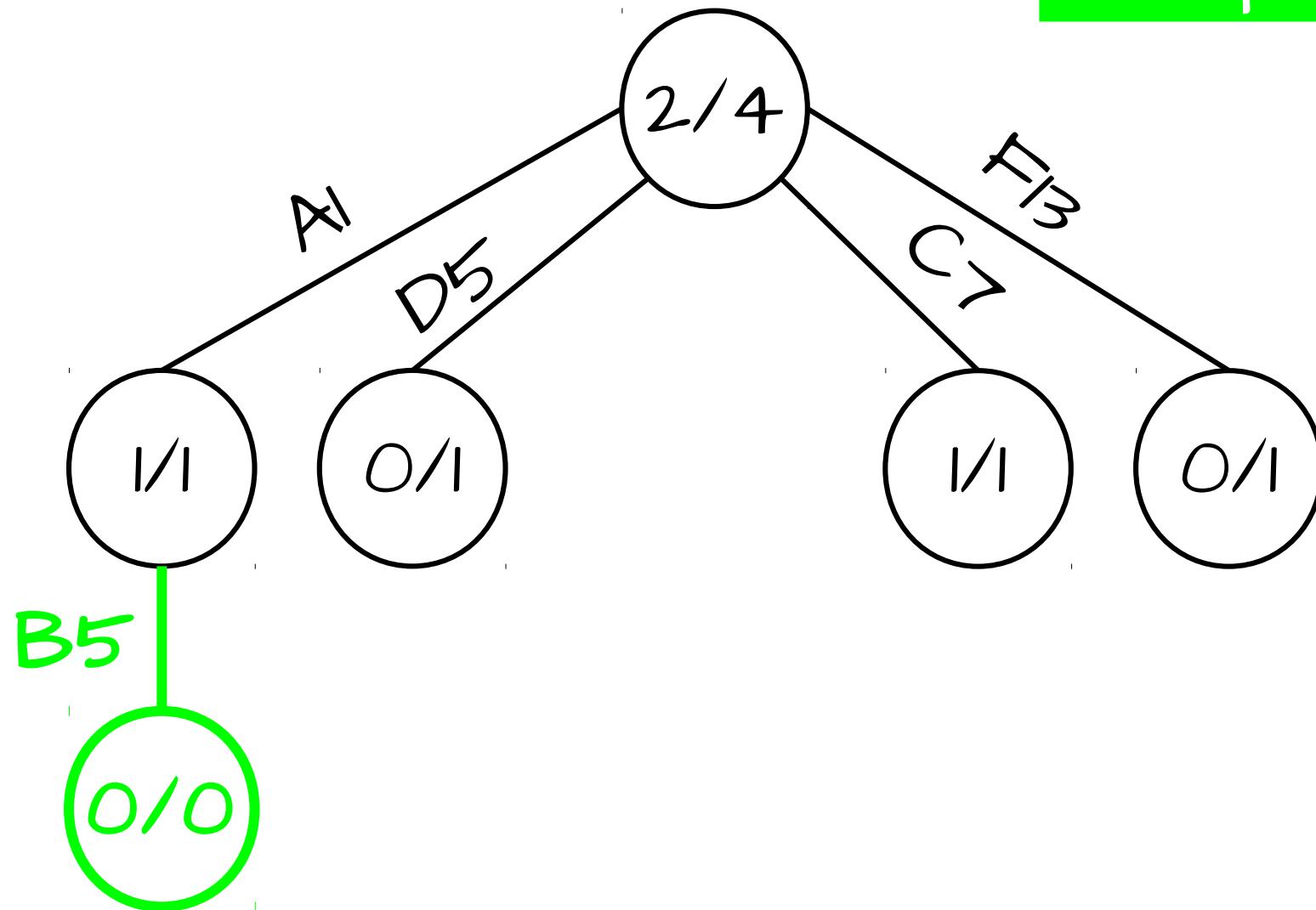
Browne, Cb, and Edward Powley. 2012. A survey of monte carlo tree search methods. *Intelligence and AI* 4, no. 1: 1-49



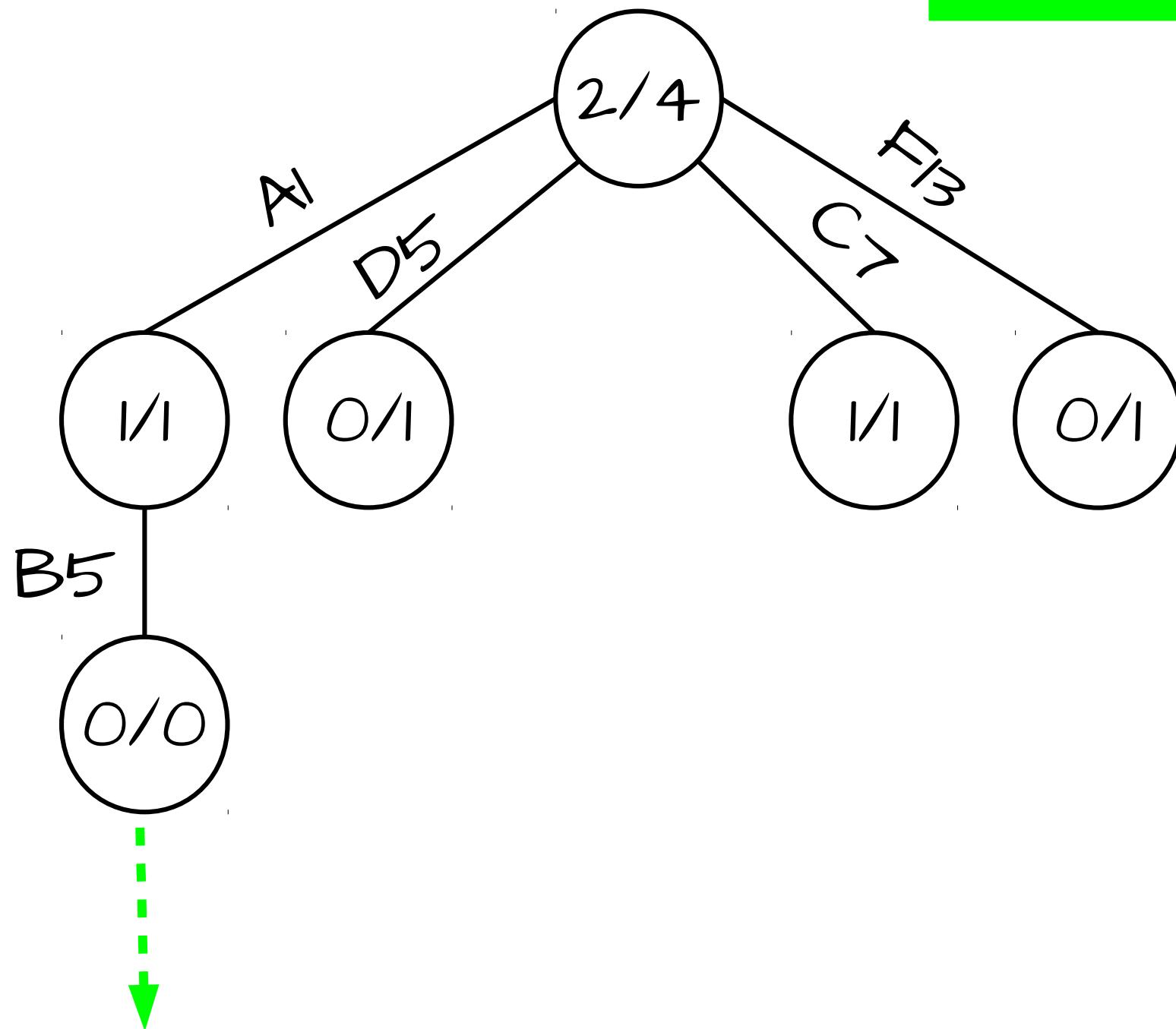
Selection



Expansion



Simulation



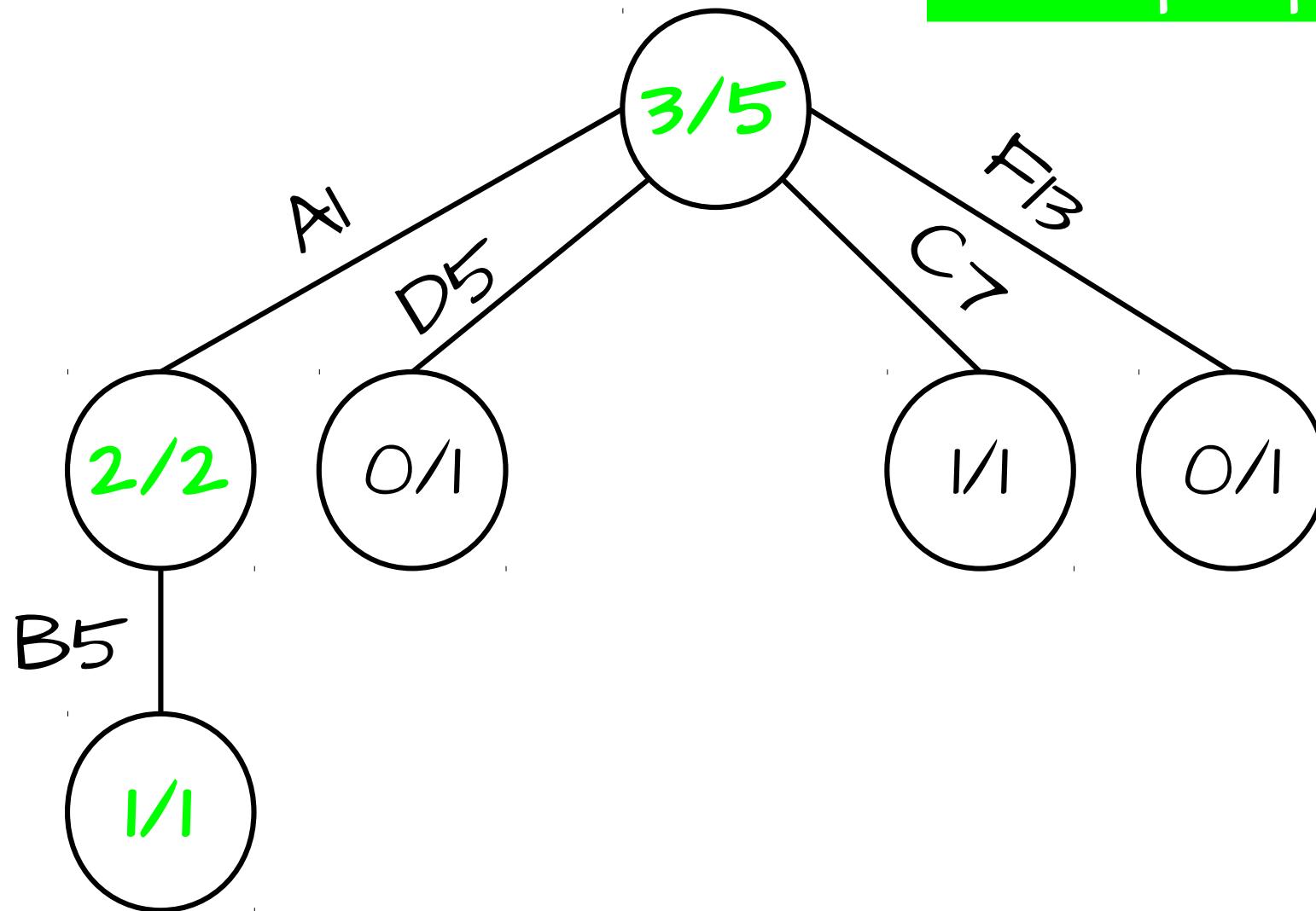
Random



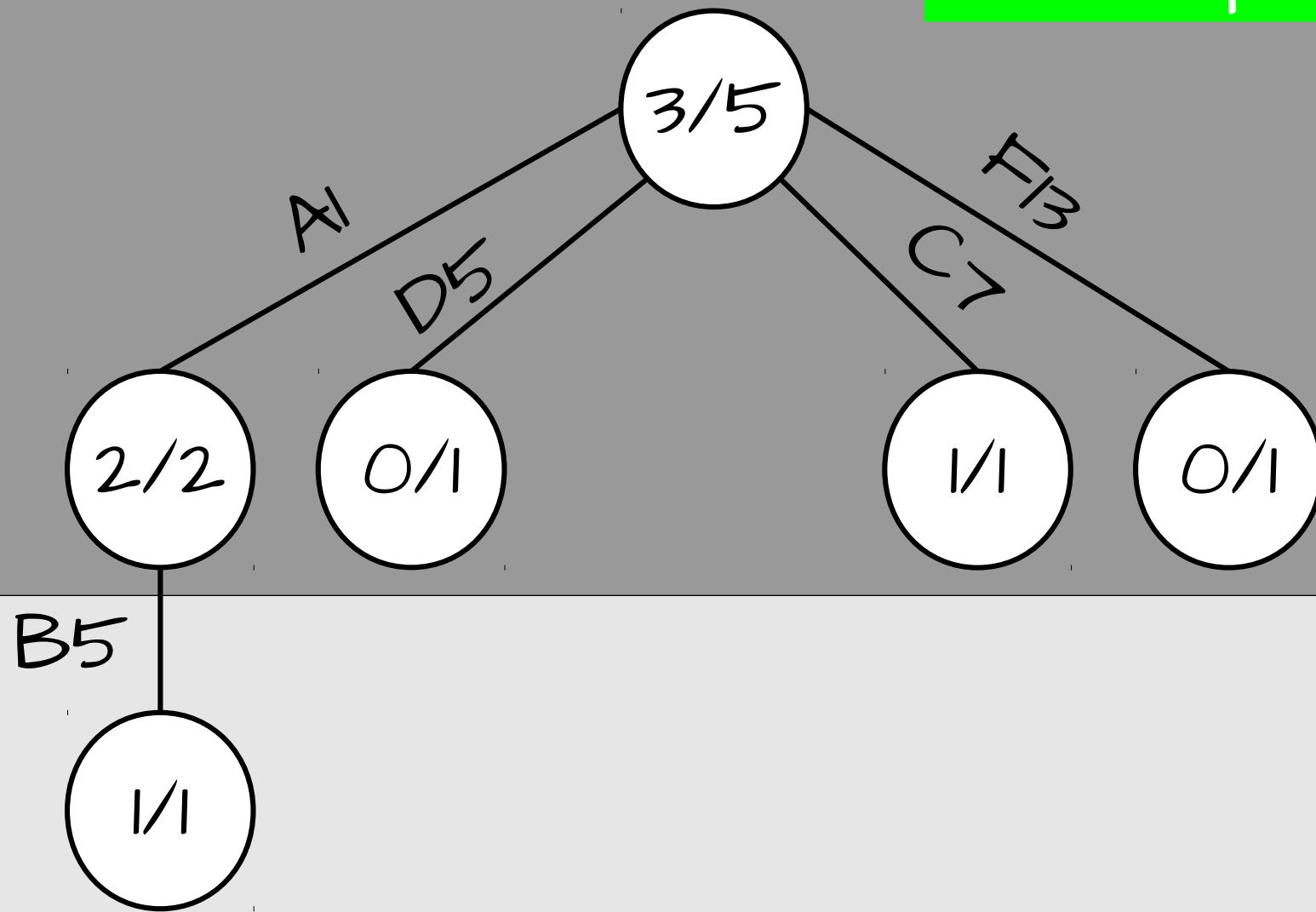
Not Human like?



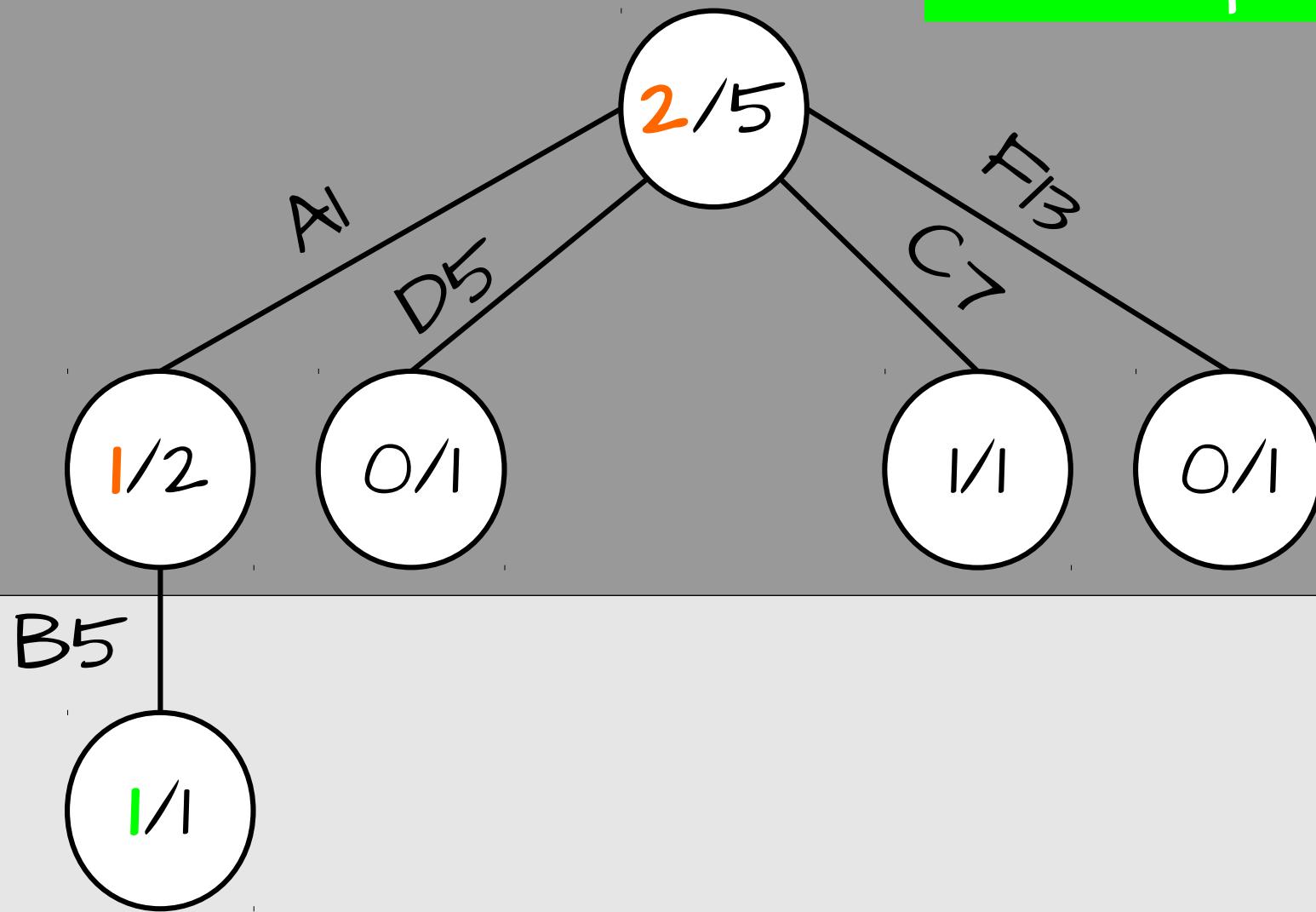
Backpropagation



Perspective



Perspective



Multi Armed Bandit



Multi Armed Bandit

Exploitation vs Exploration



$$\frac{wins}{visits} + explorationFactor \sqrt{\frac{\ln(totalVisits)}{visits}}$$

15042

36/116

86/193

58/151

0/1

2/2

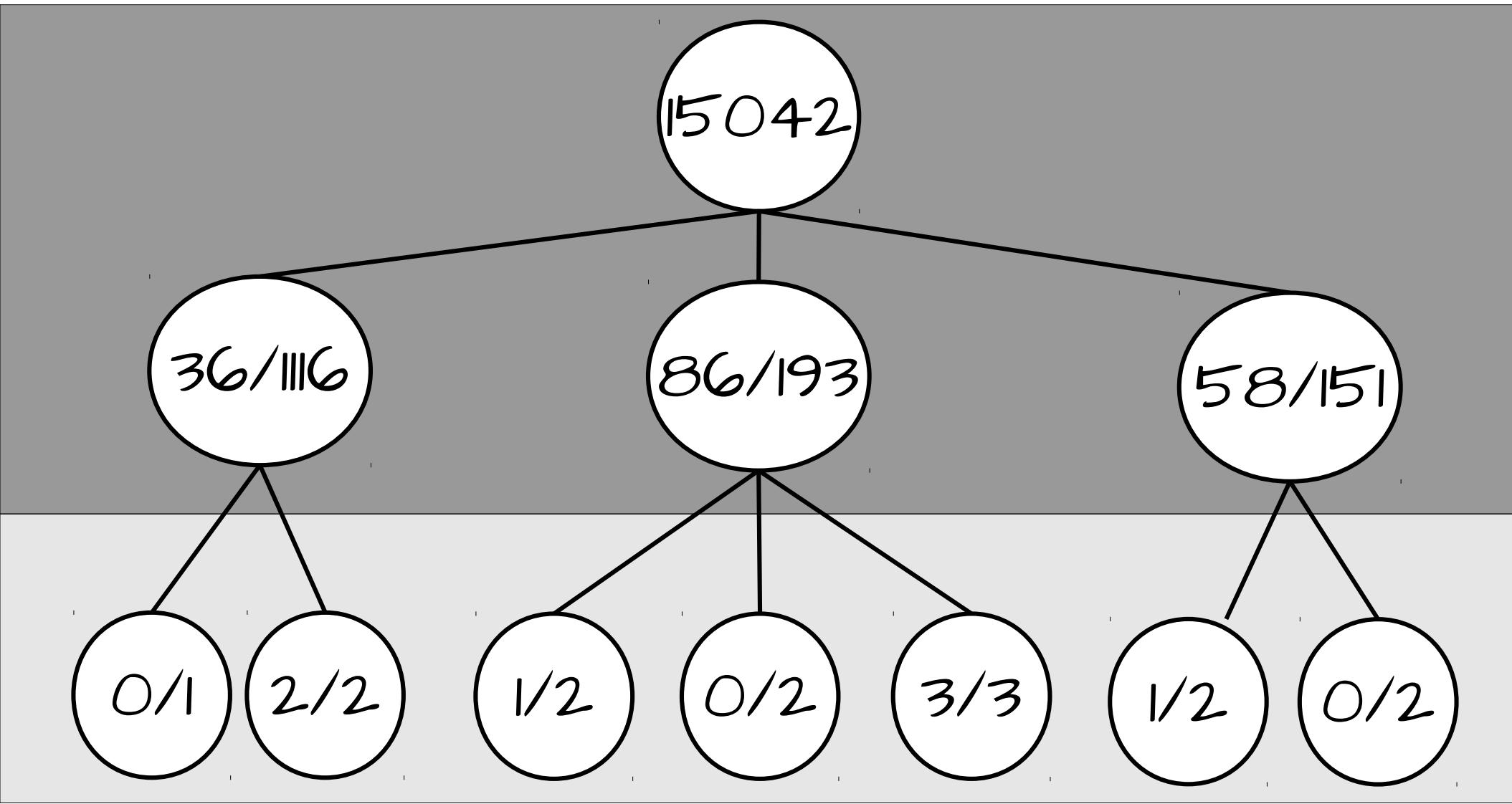
1/2

0/2

3/3

1/2

0/2



15042

86/193

36/116

58/151

0/1

2/2

1/2

0/2

3/3

1/2

0/2

15042

36/116

86/193

58/151

0/1

2/2

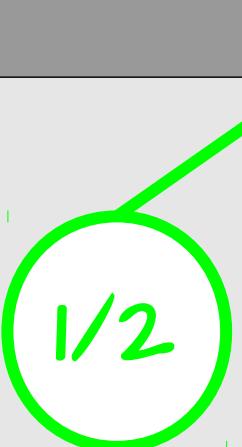
1/2

0/2

3/3

1/2

0/2



Generate a valid random move

Who has won?



General Game Playing

Anytime

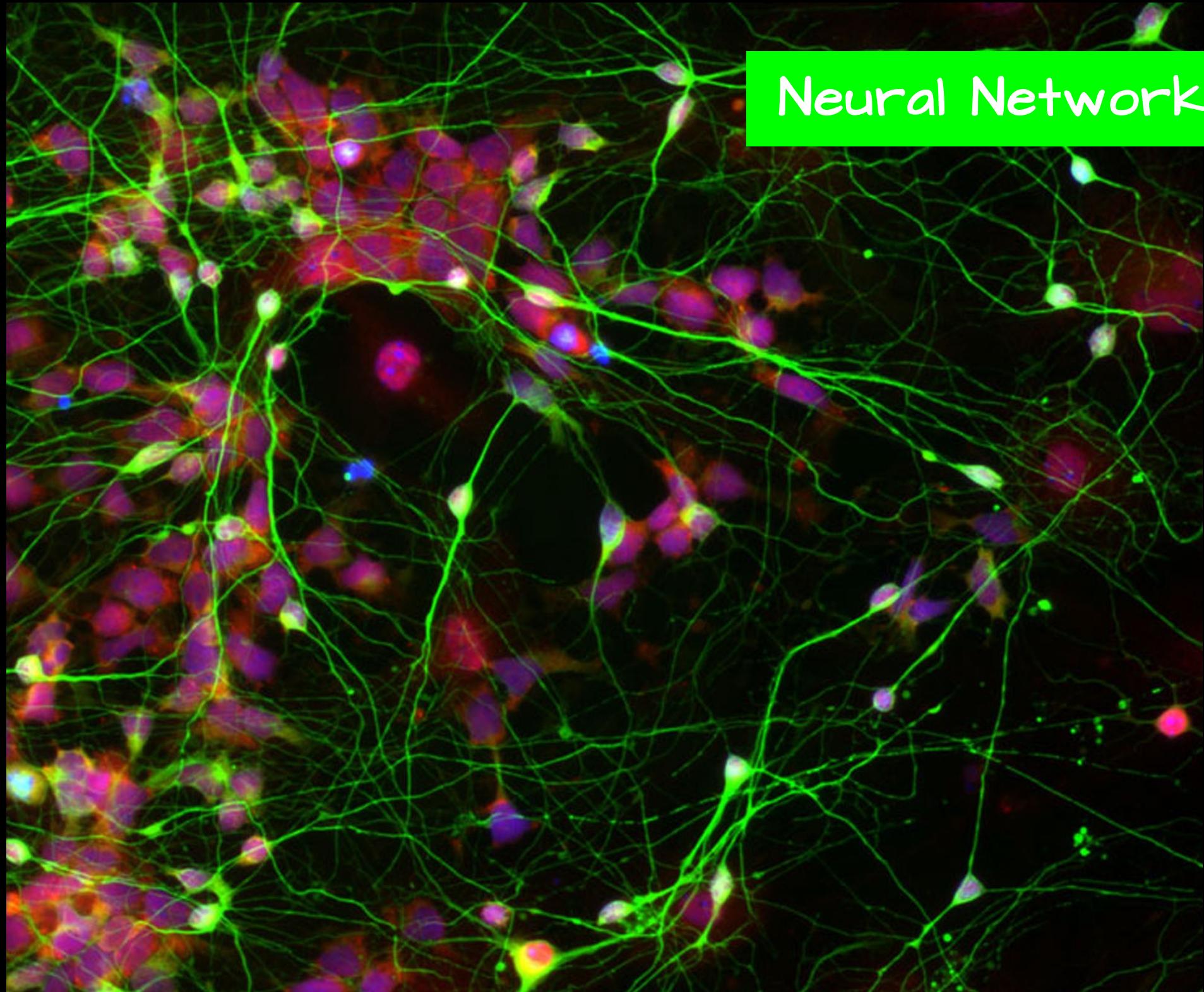
Lazy







Expert Knowledge

A microscopic image showing a dense network of neurons. The neurons are stained with two colors: green and purple. The green staining highlights the neuron membranes, creating a complex web-like structure. The purple staining is concentrated in the cell bodies (soma) and some of the shorter processes. The overall image has a dark background, making the bright green and purple signals stand out.

Neural Networks

MOVE EVALUATION IN GO USING DEEP CONVOLUTIONAL NEURAL NETWORKS

Chris J. Maddison

University of Toronto

cmaddis@cs.toronto.edu

2014

Aja Huang¹, Ilya Sutskever², David Silver¹

Google DeepMind¹, Google Brain²

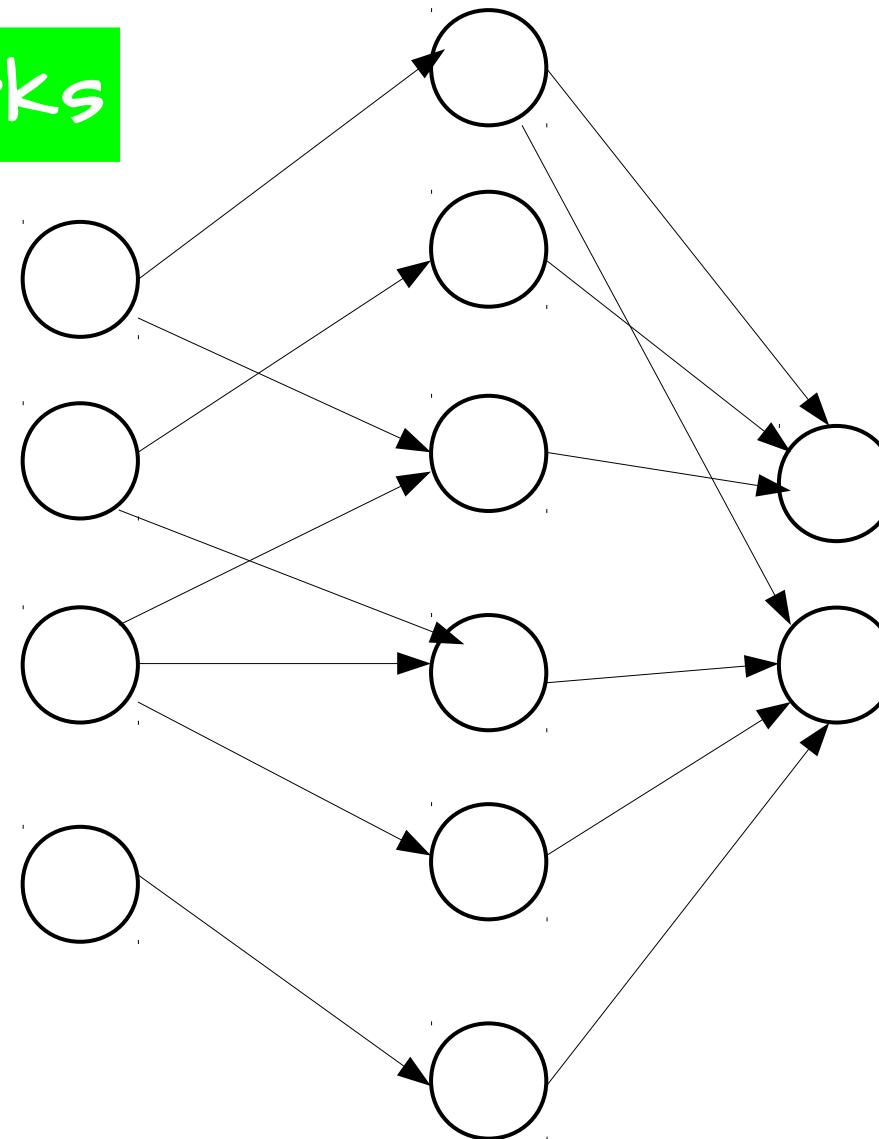
{ajahuang, ilyasu, davidsilver}@google.com

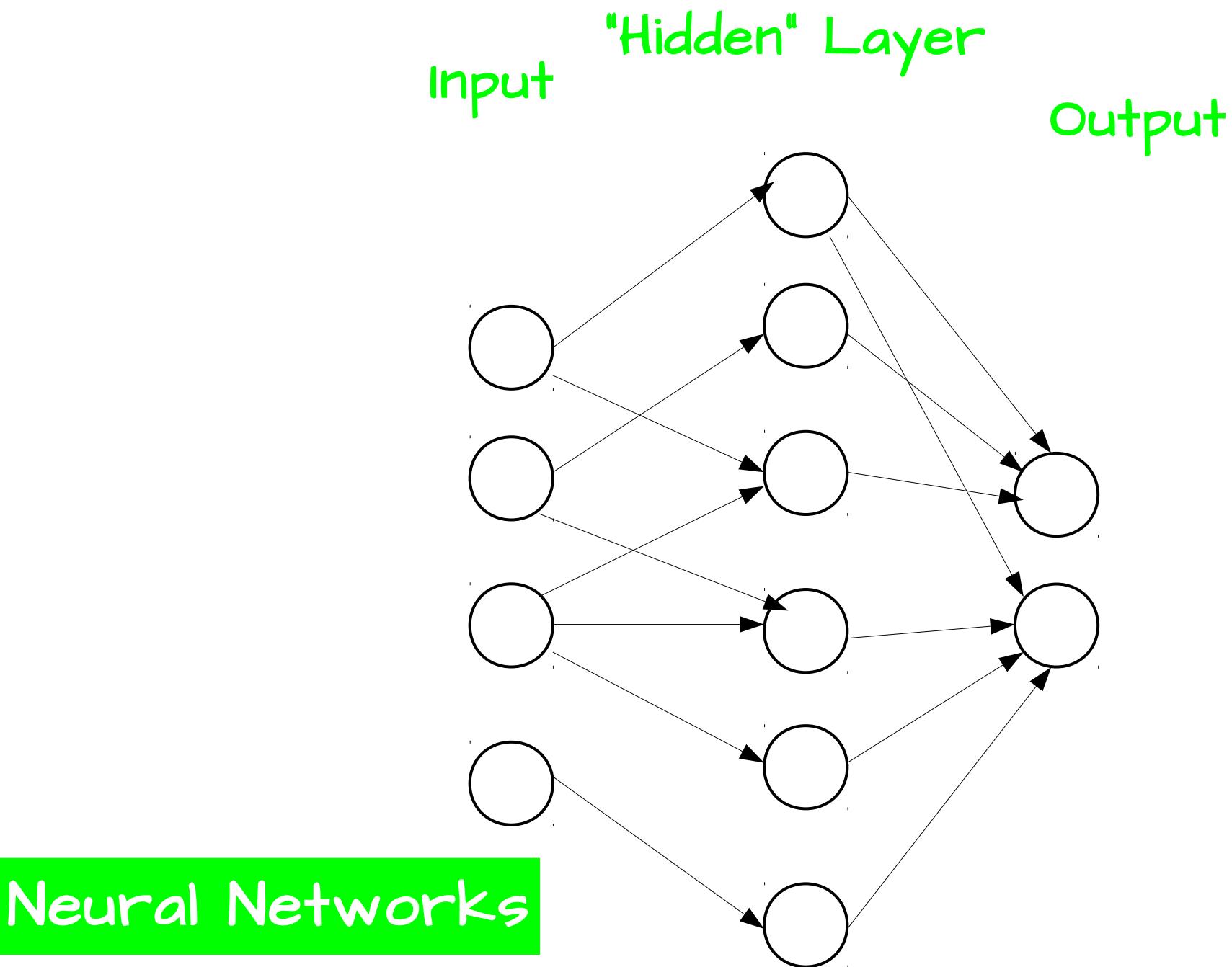
ABSTRACT

The game of Go is more challenging than other board games, due to the difficulty of constructing a position or move evaluation function. In this paper we investigate whether deep convolutional networks can be used to directly represent and learn this knowledge. We train a large 12-layer convolutional neural network by supervised learning from a database of human professional games. The network correctly predicts the expert move in 55% of positions, equalling the accuracy of a 6 dan human player. When the trained convolutional network was used di-

What does this even mean?

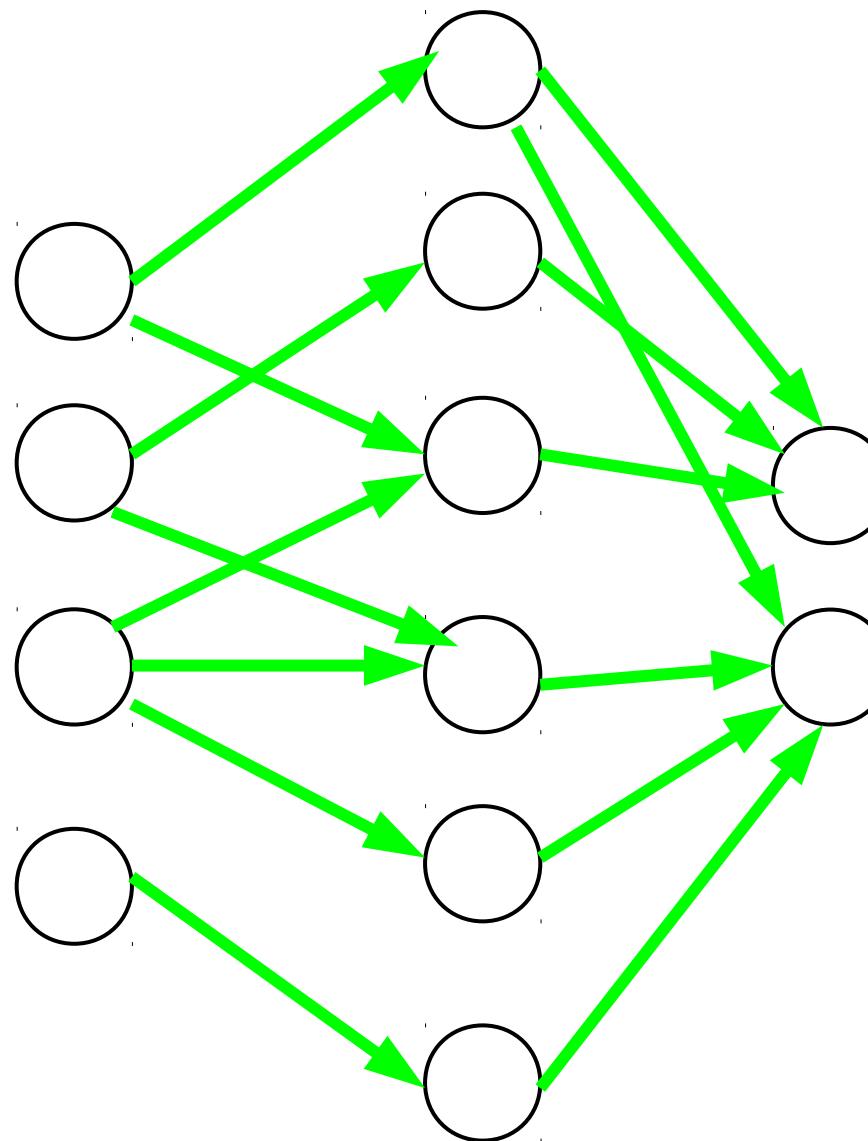
Neural Networks



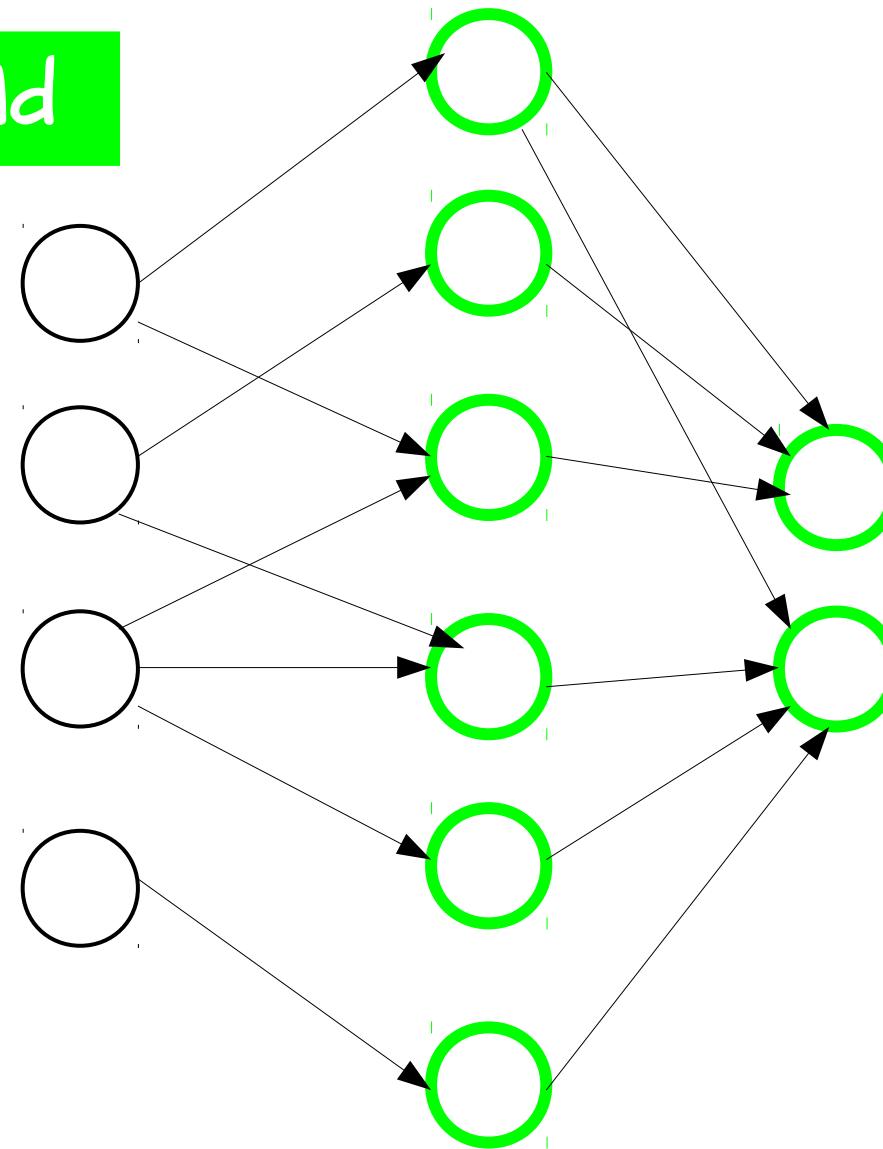


Neural Networks

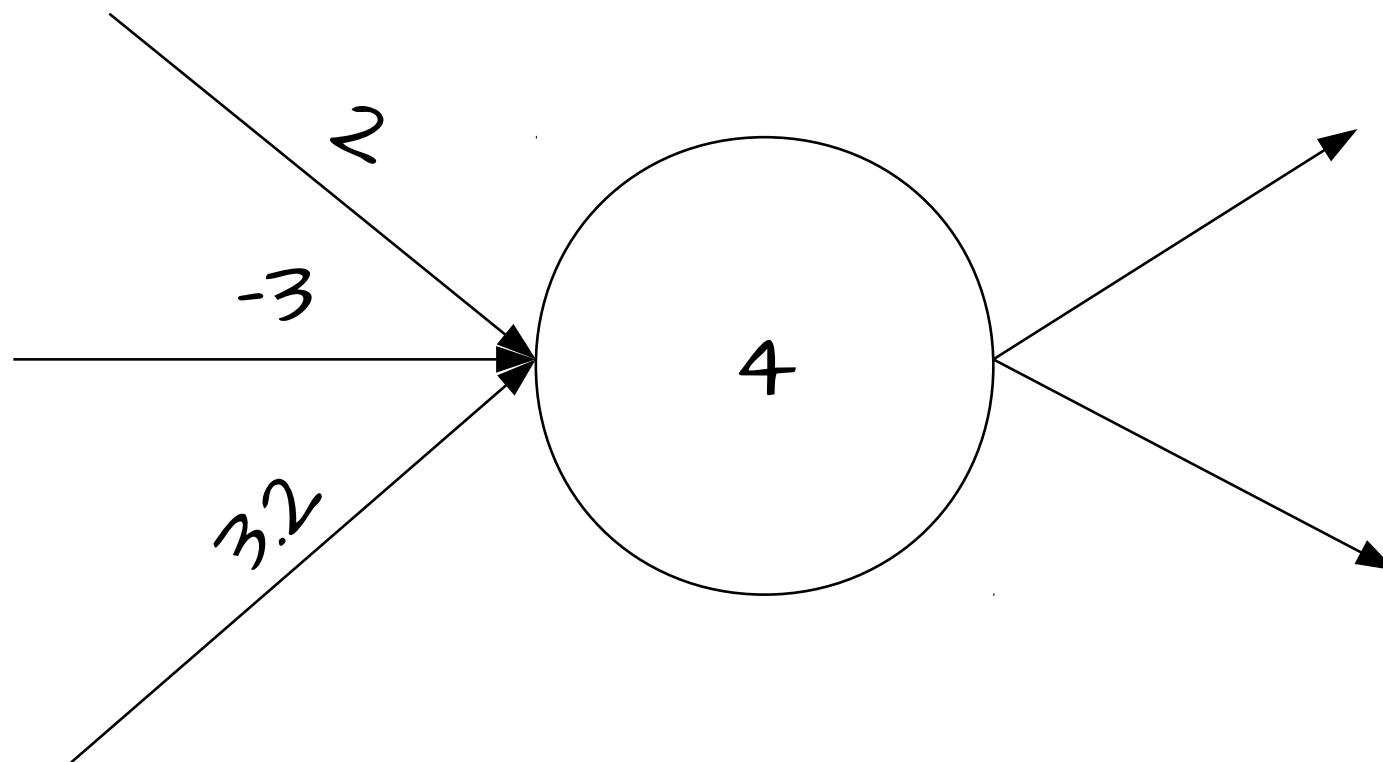
Weights



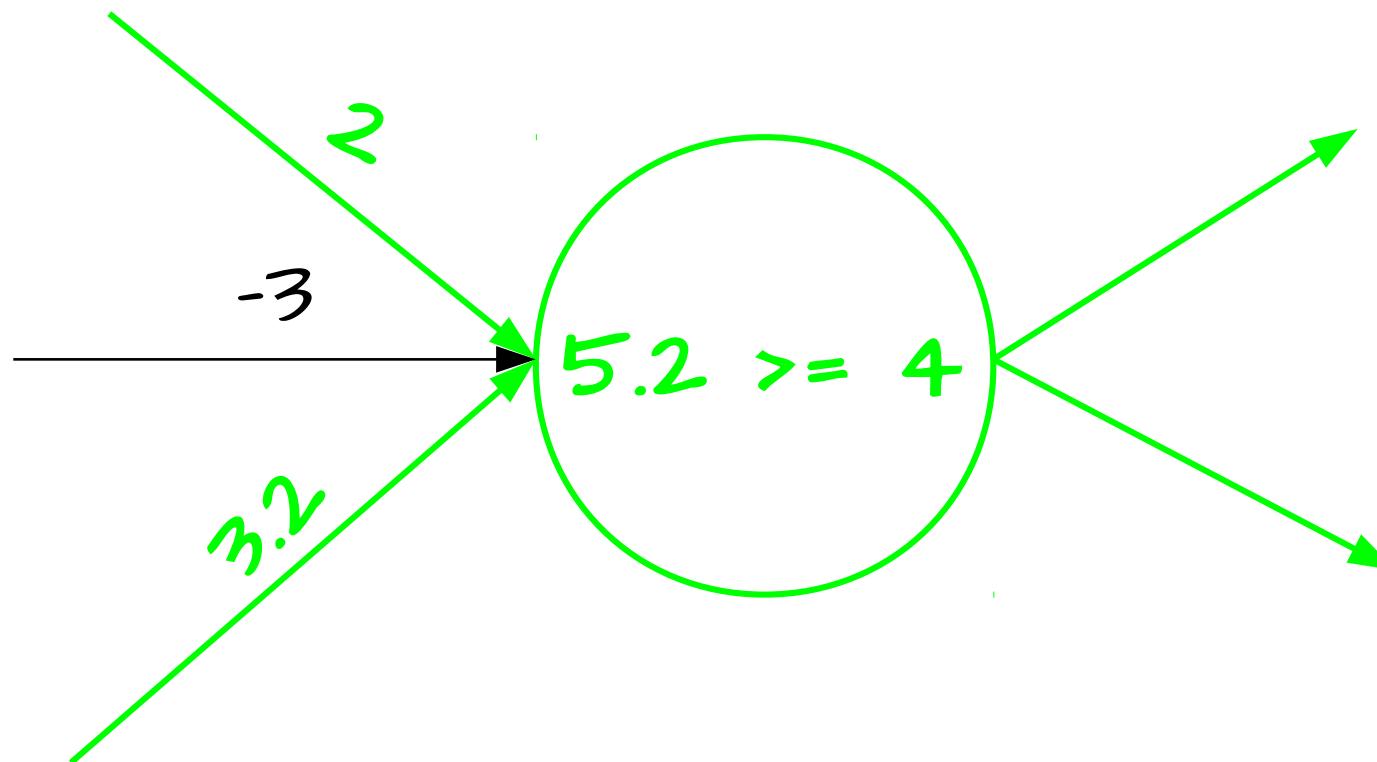
Bias/Threshold



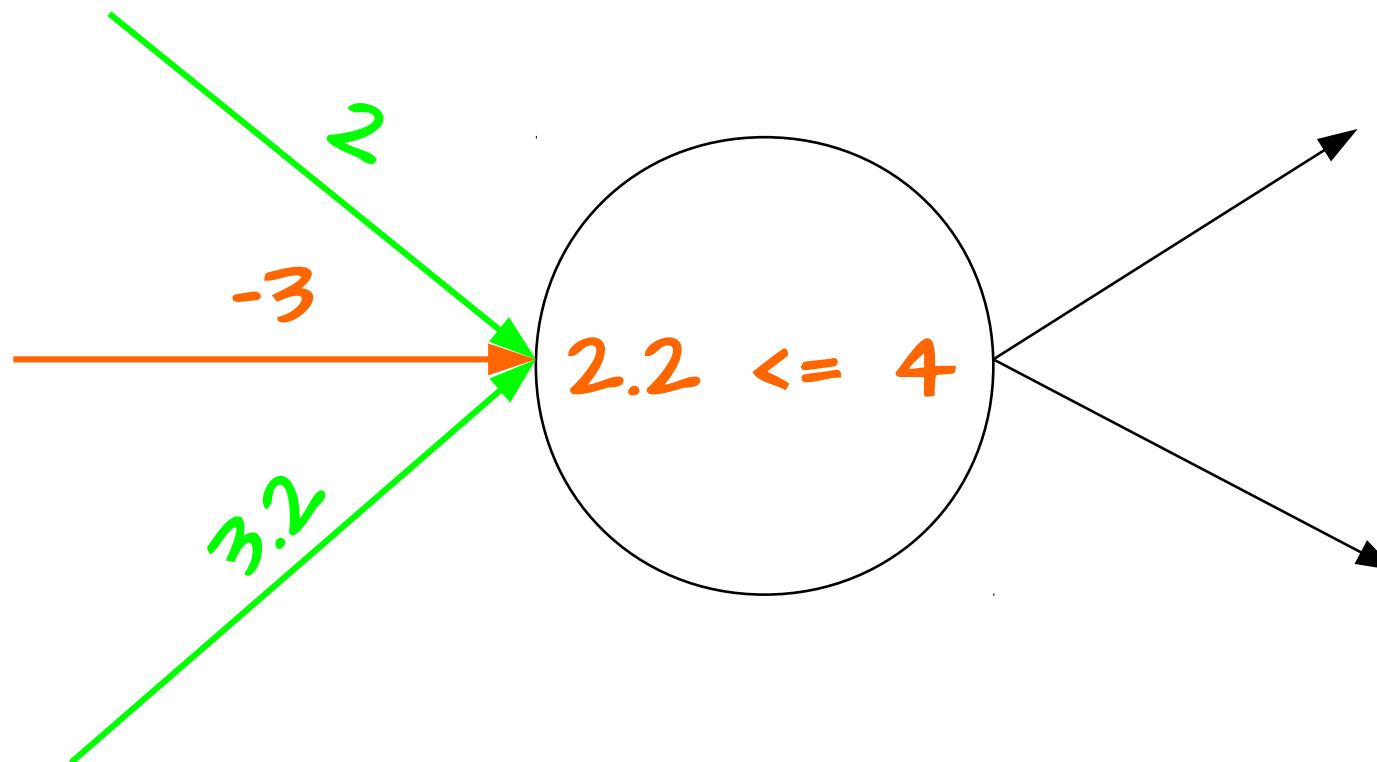
Activation



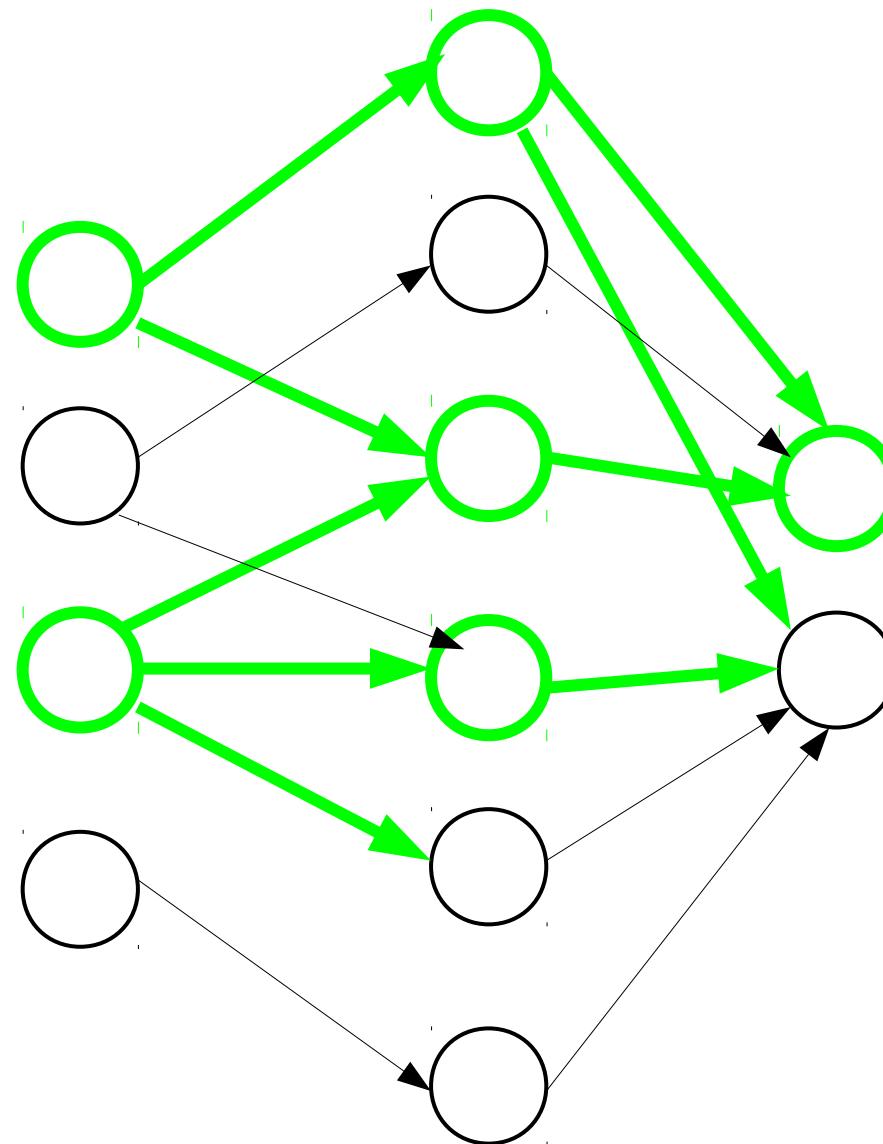
Activation



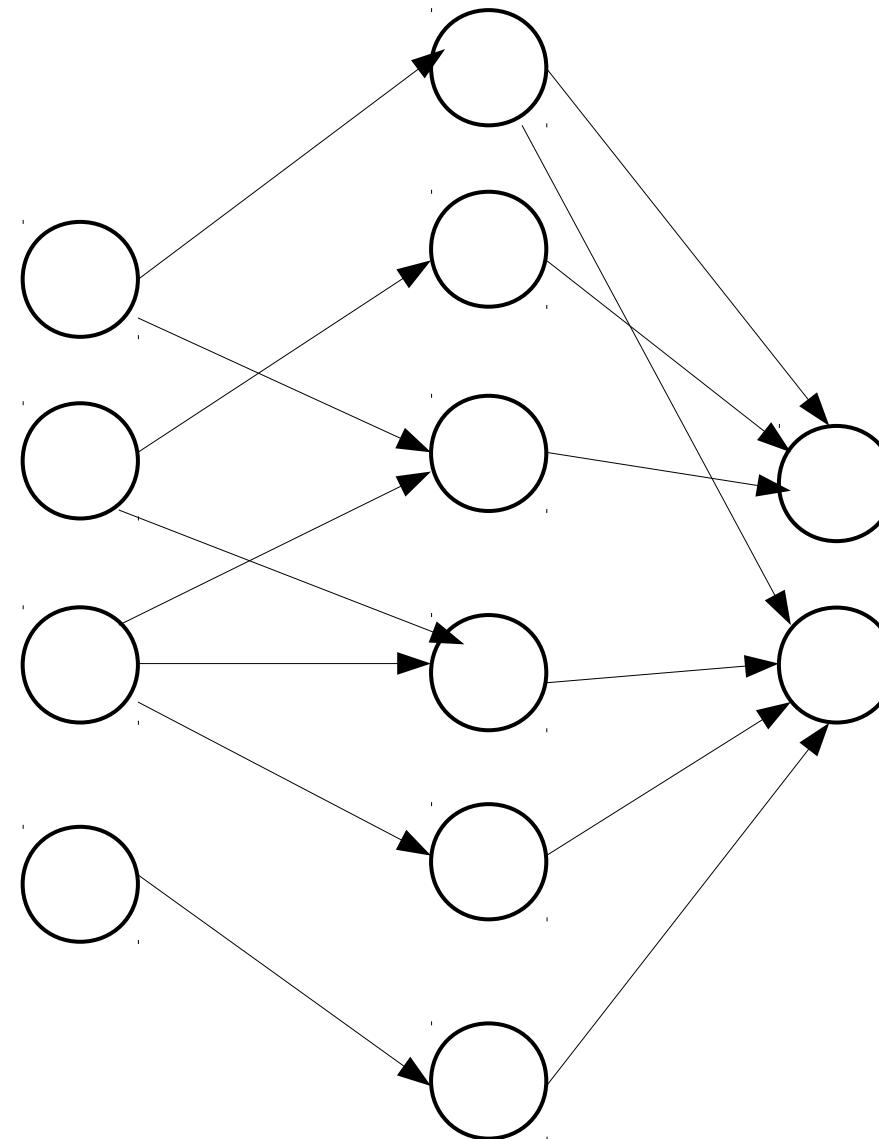
Activation



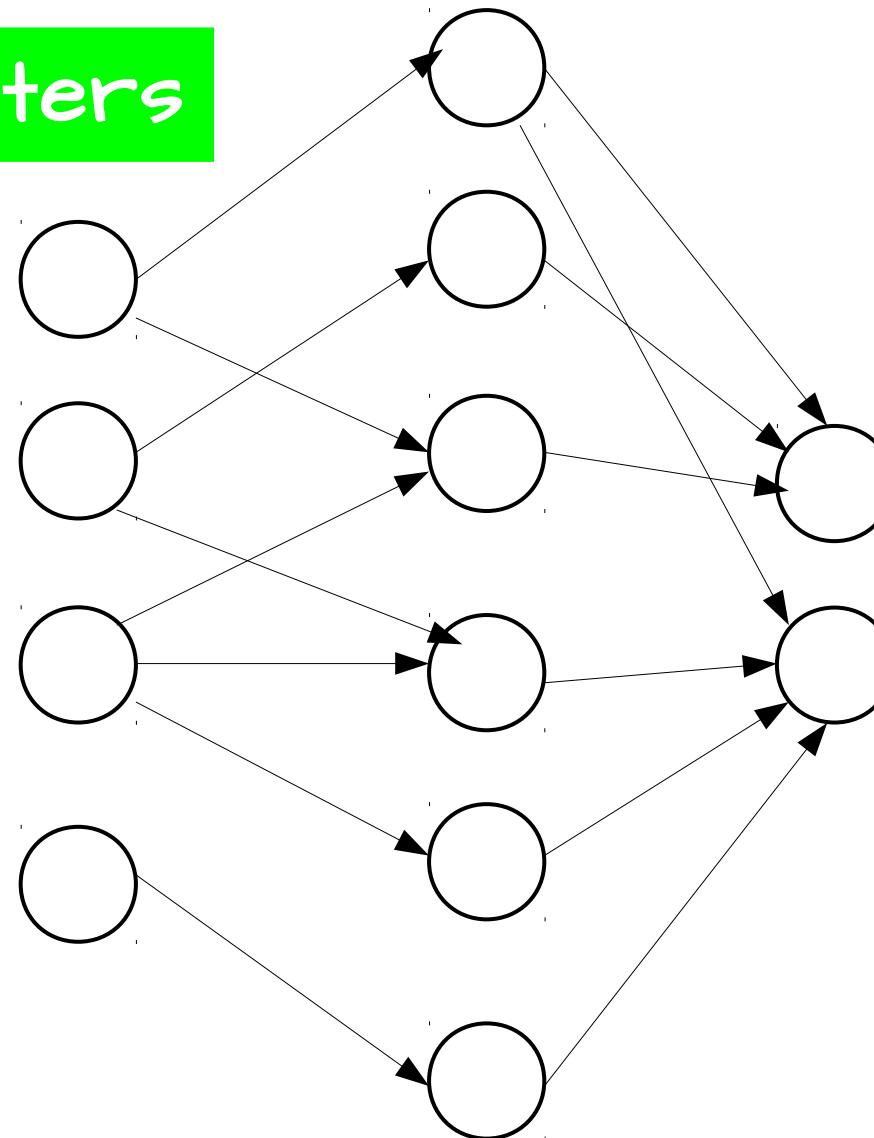
Activation



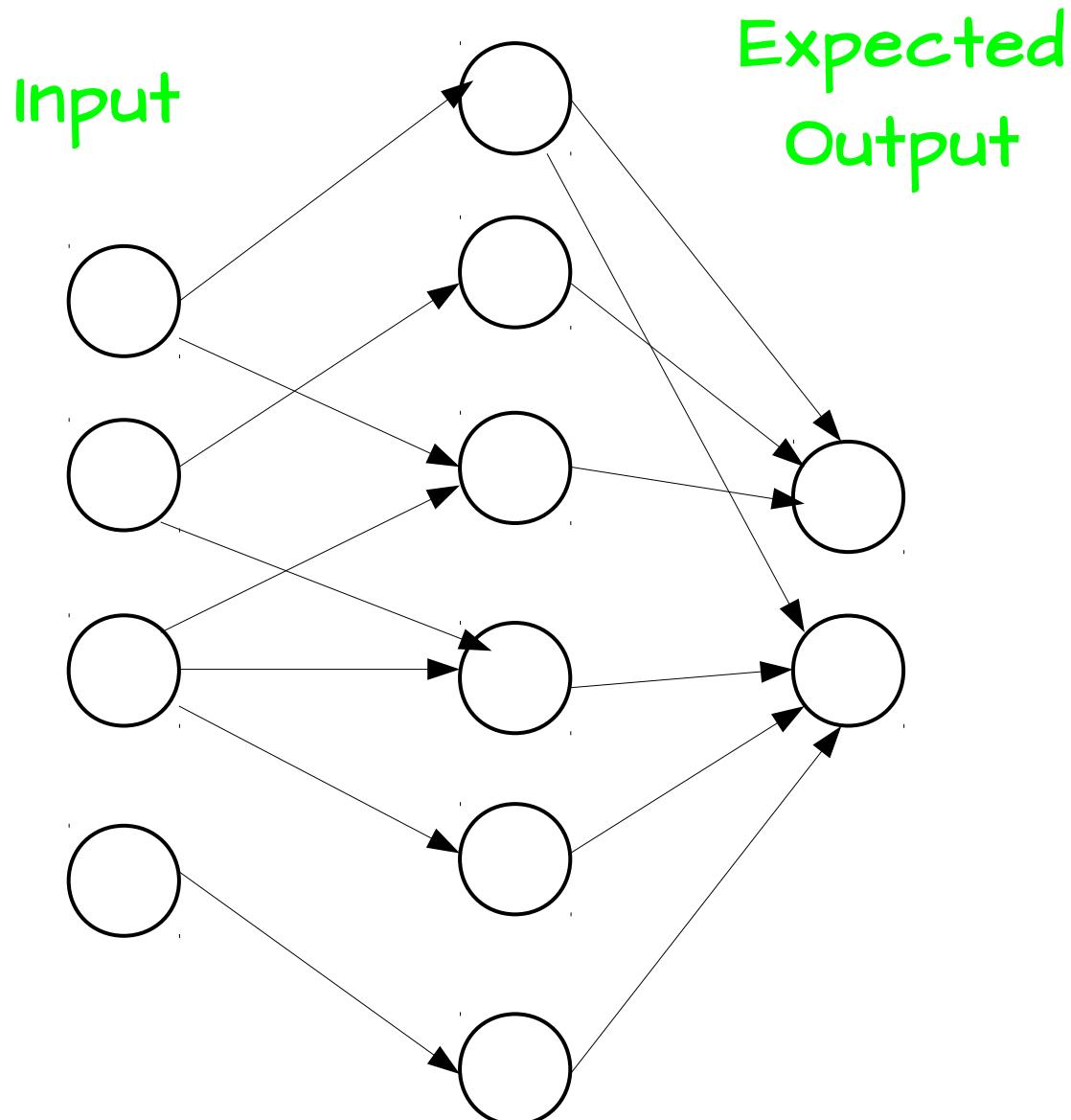
Training



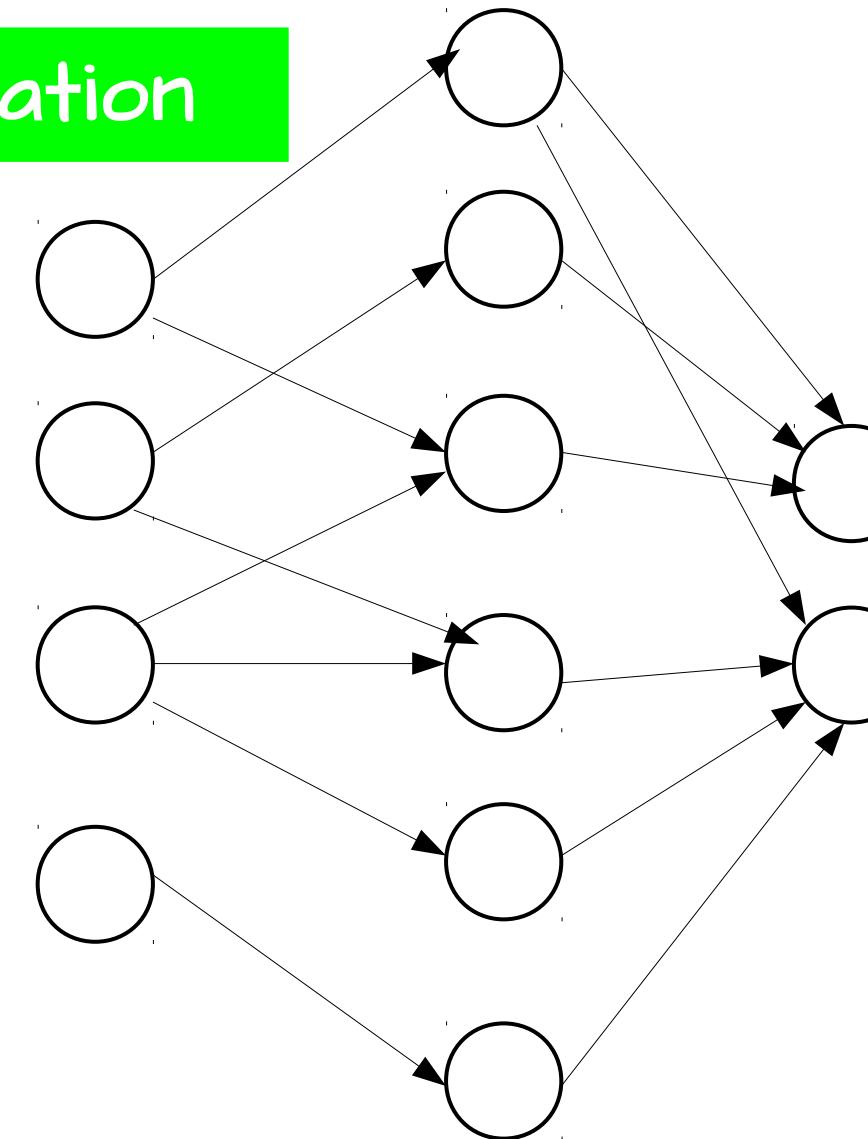
Adjust parameters



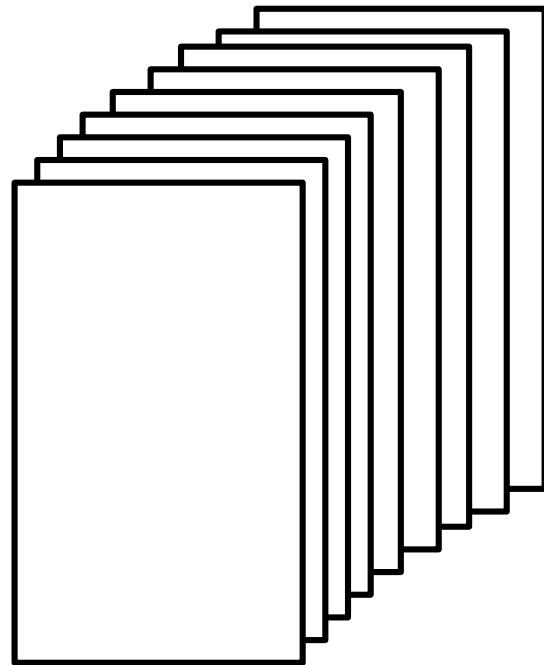
Supervised Learning



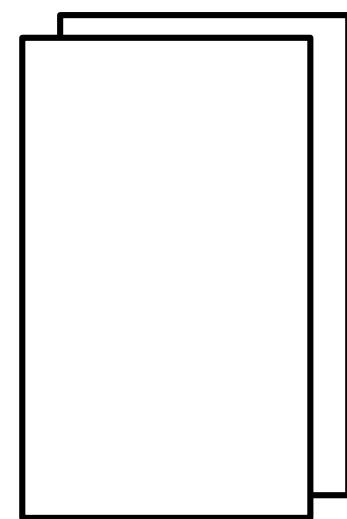
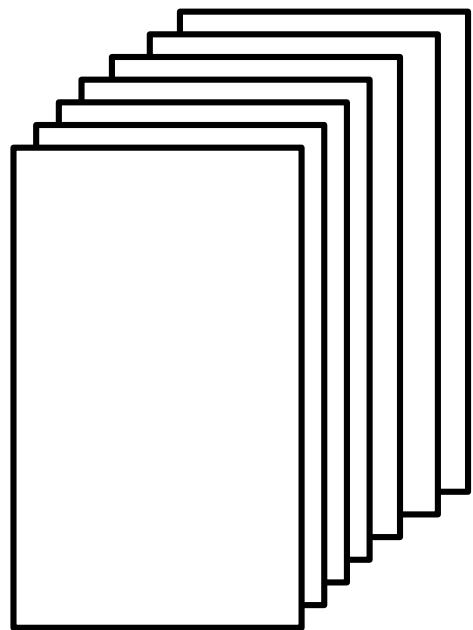
Backpropagation



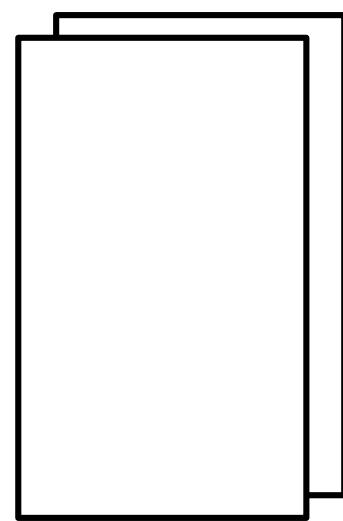
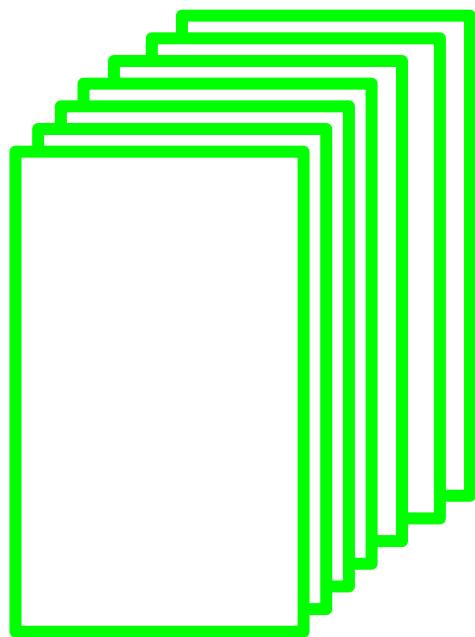
Data set



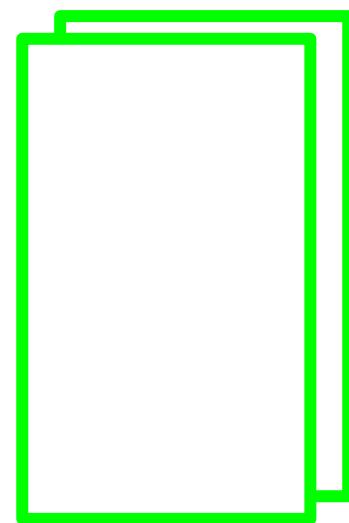
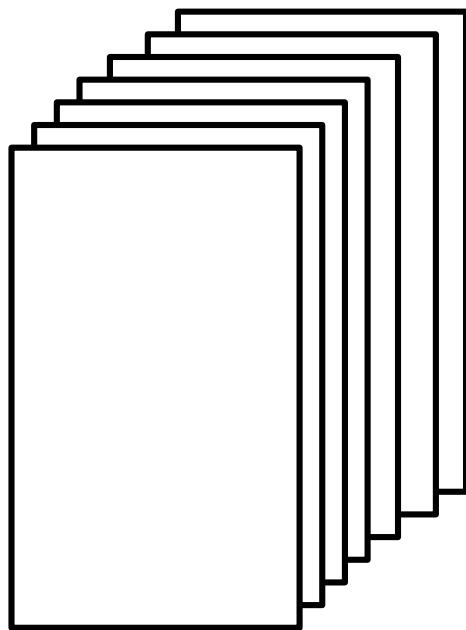
Training data + test data



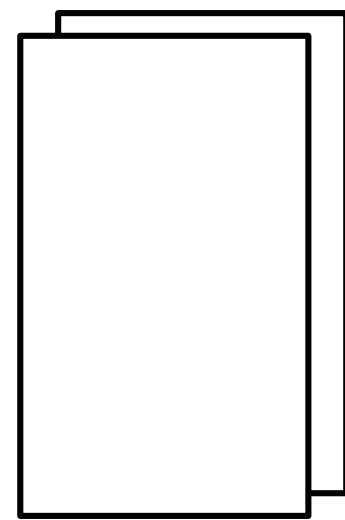
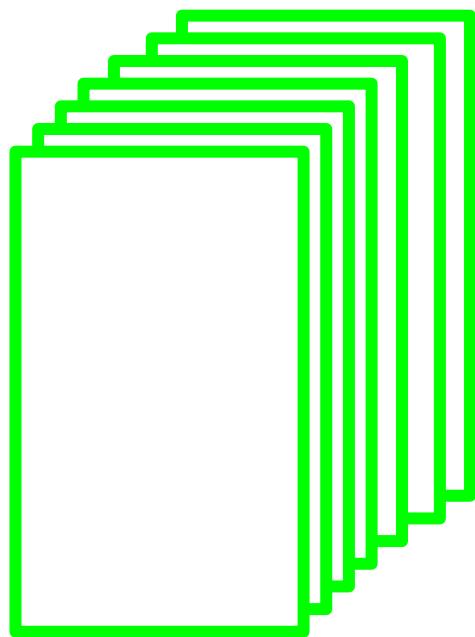
Training



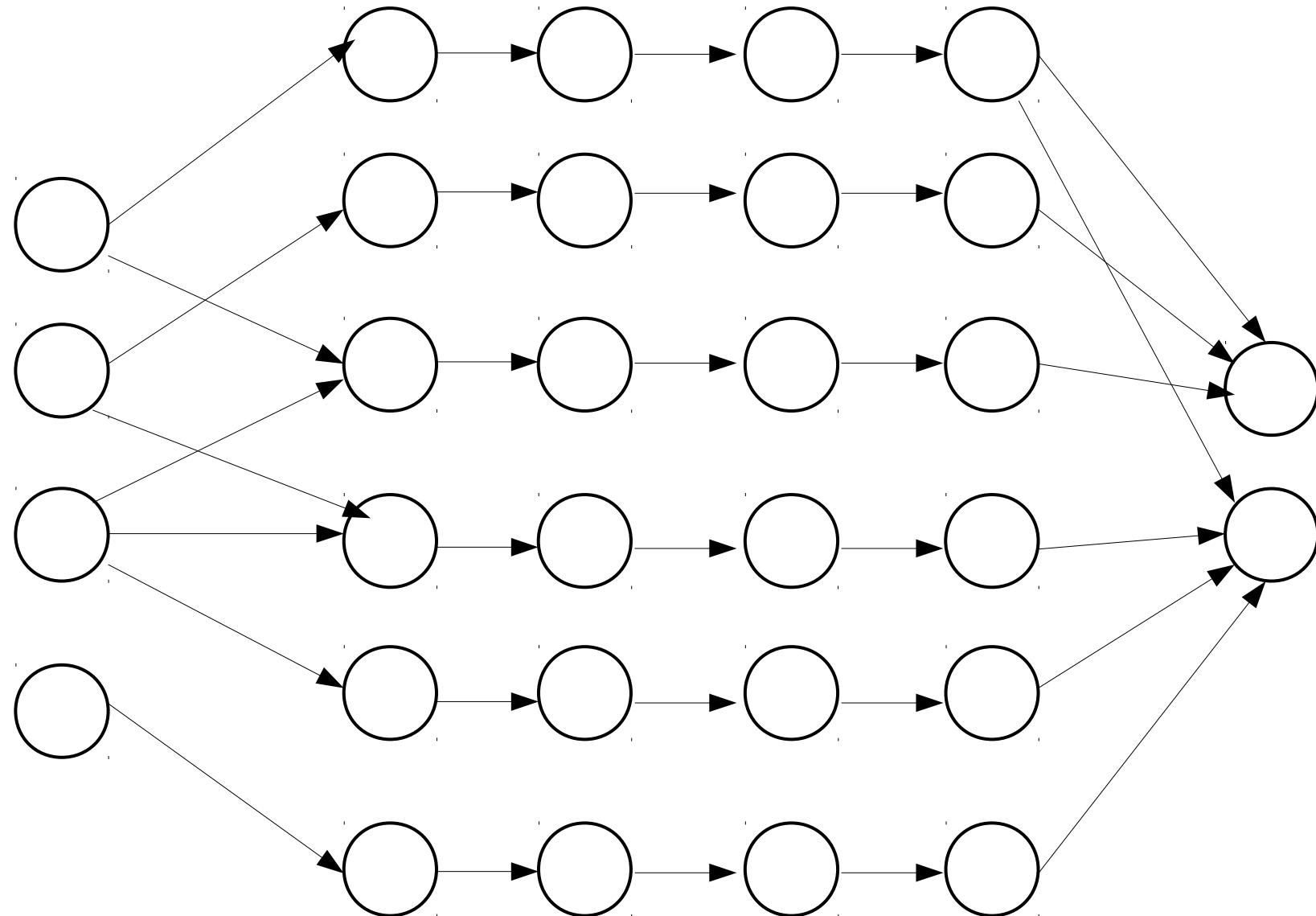
Verify



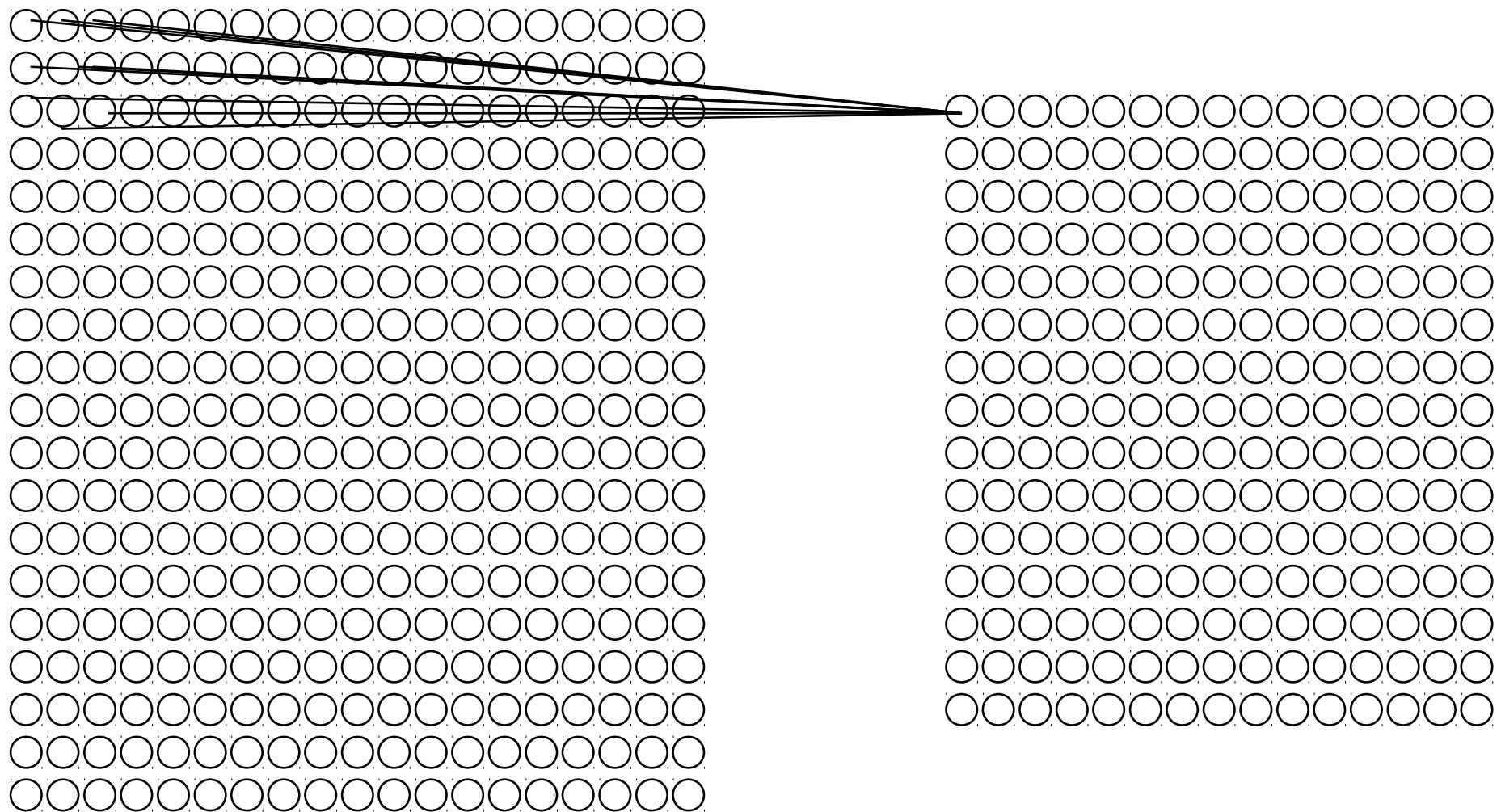
Overfitting



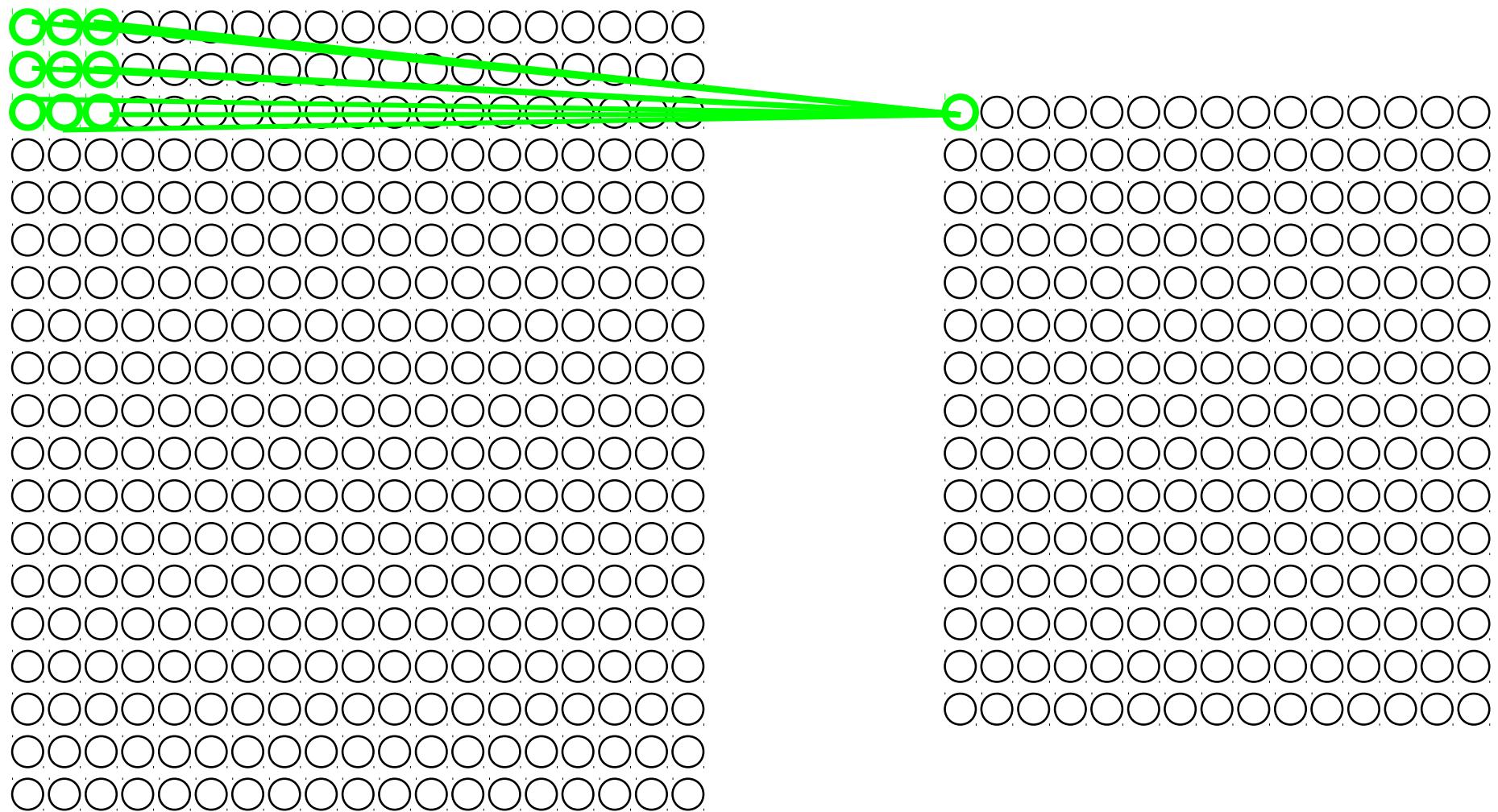
Deep Neural Networks



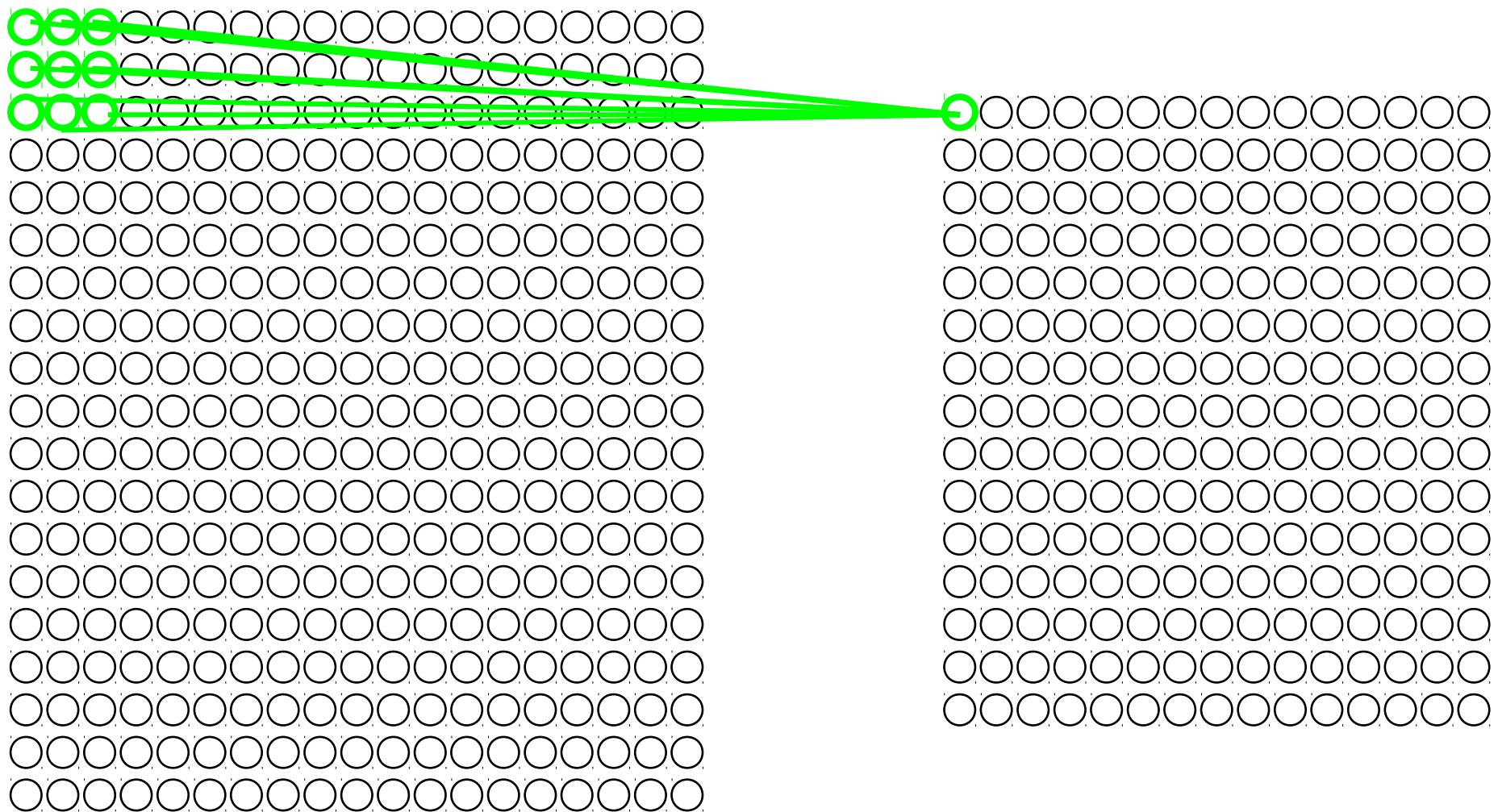
Convolutional Neural Networks



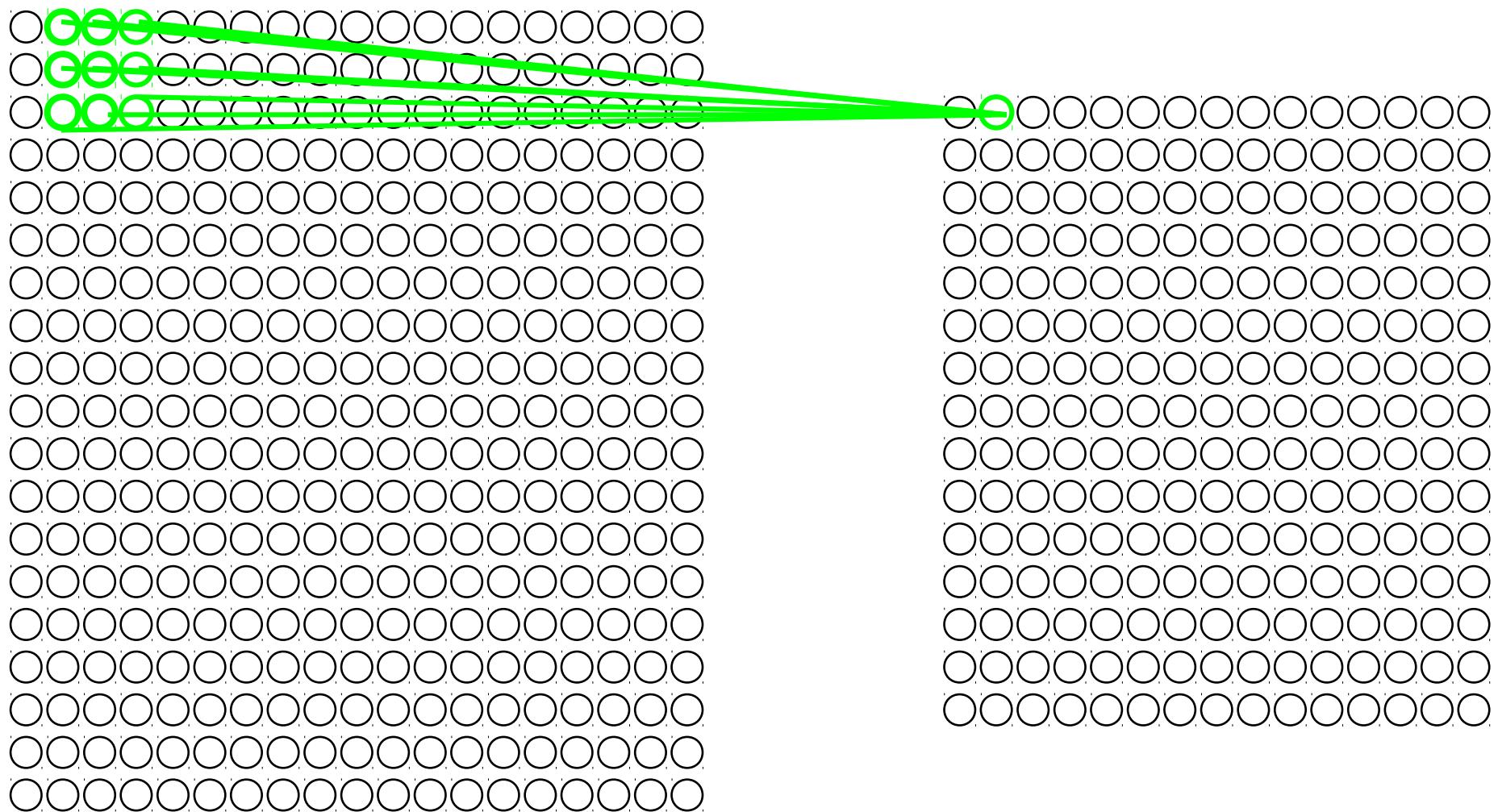
Local Receptive Field



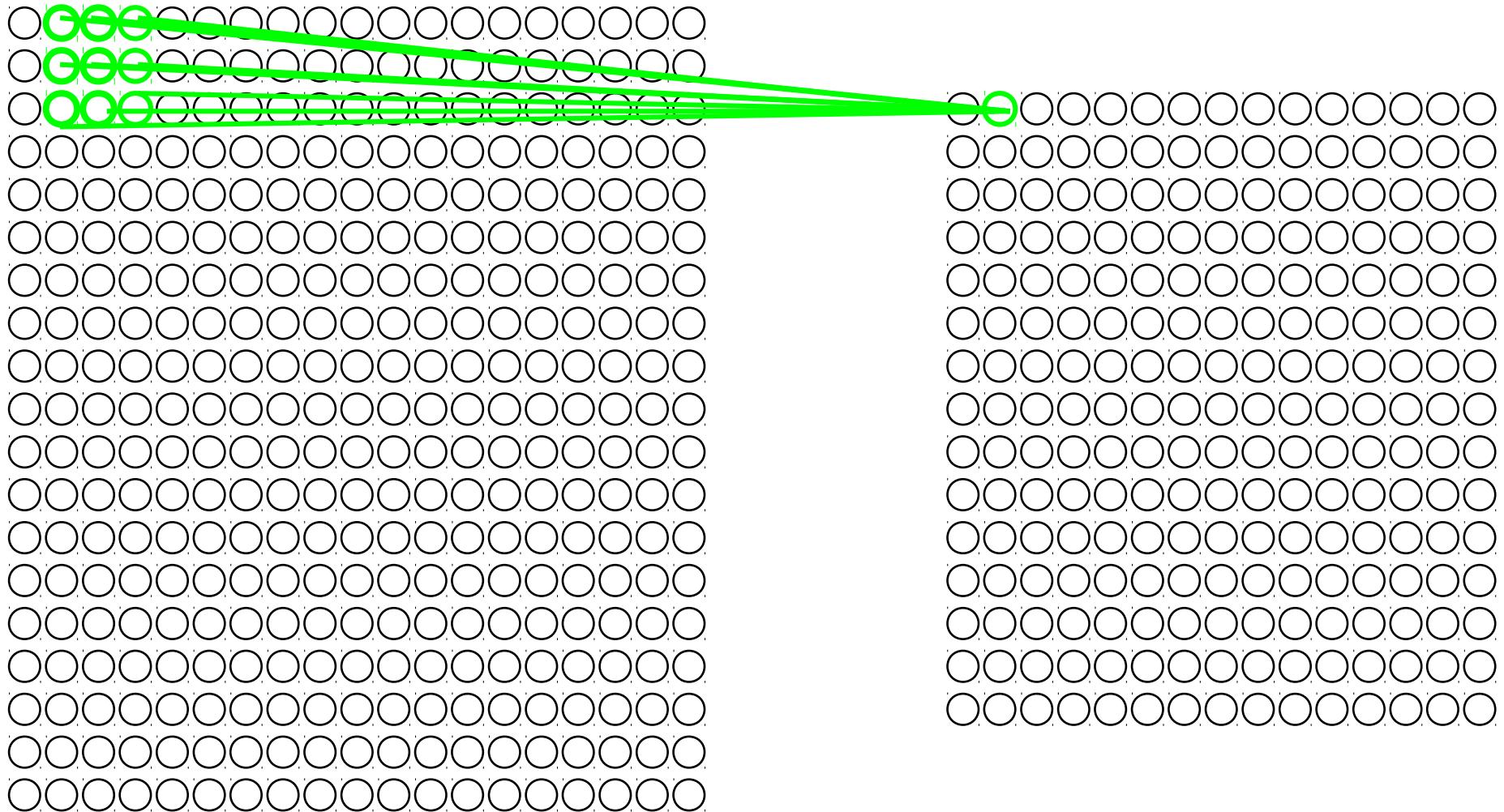
Feature Map



Stride



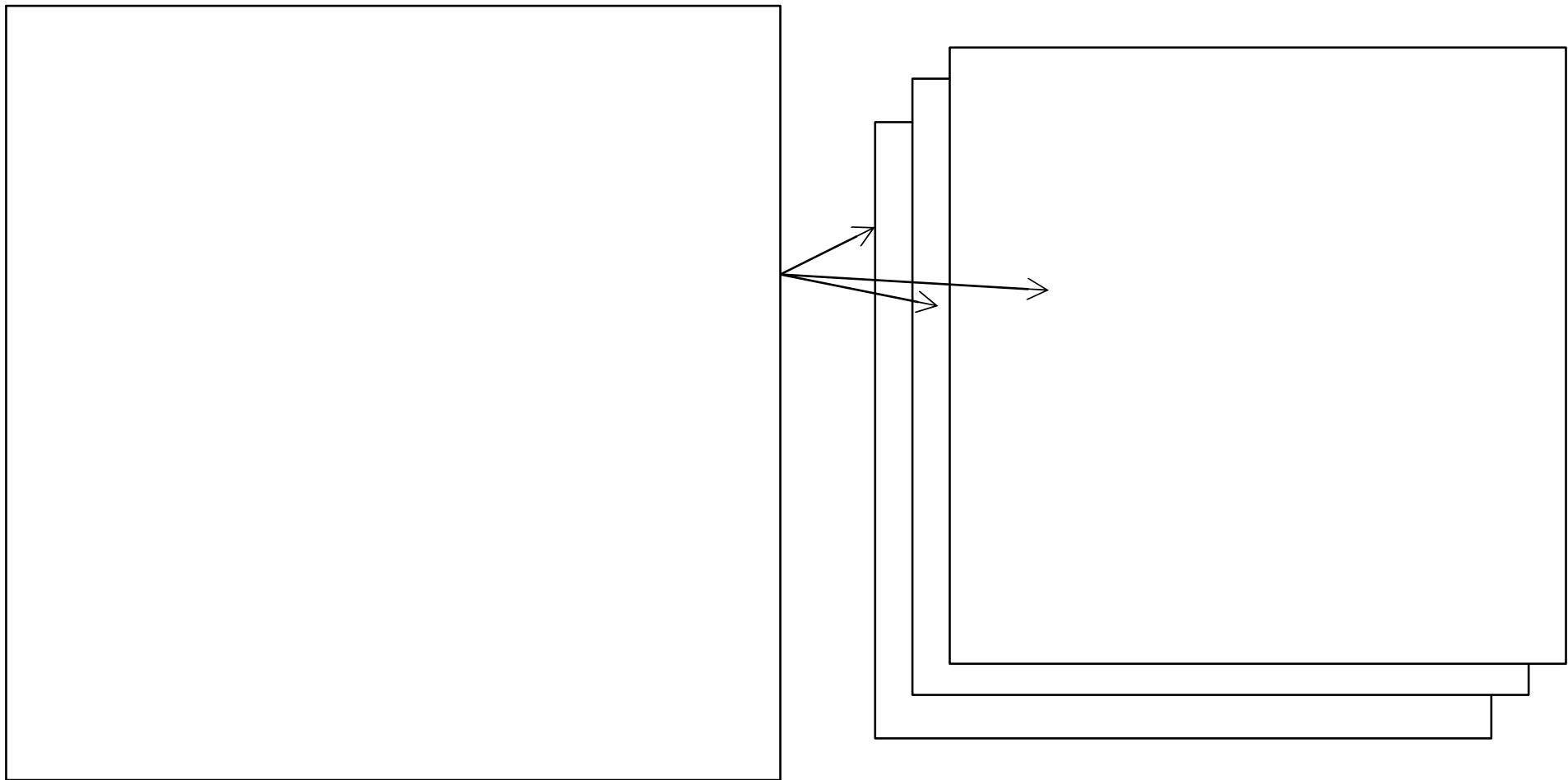
Shared weights and biases



Multiple Feature maps/filters

19×19

$3 \times 17 \times 17$

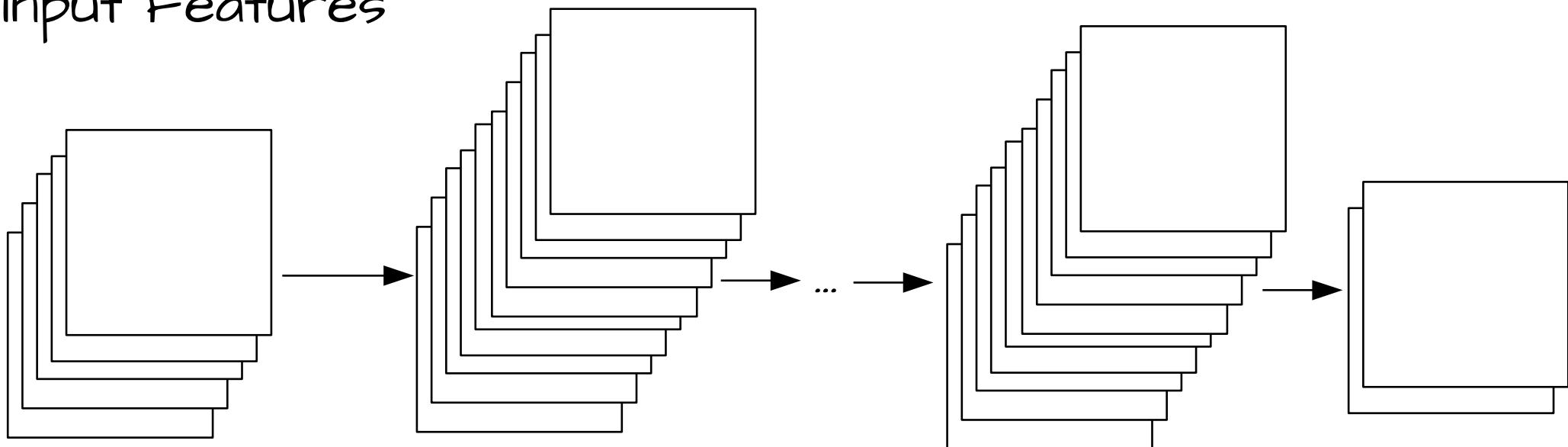


Architecture

12 layers with 64 - 192 filters

Input Features

Output

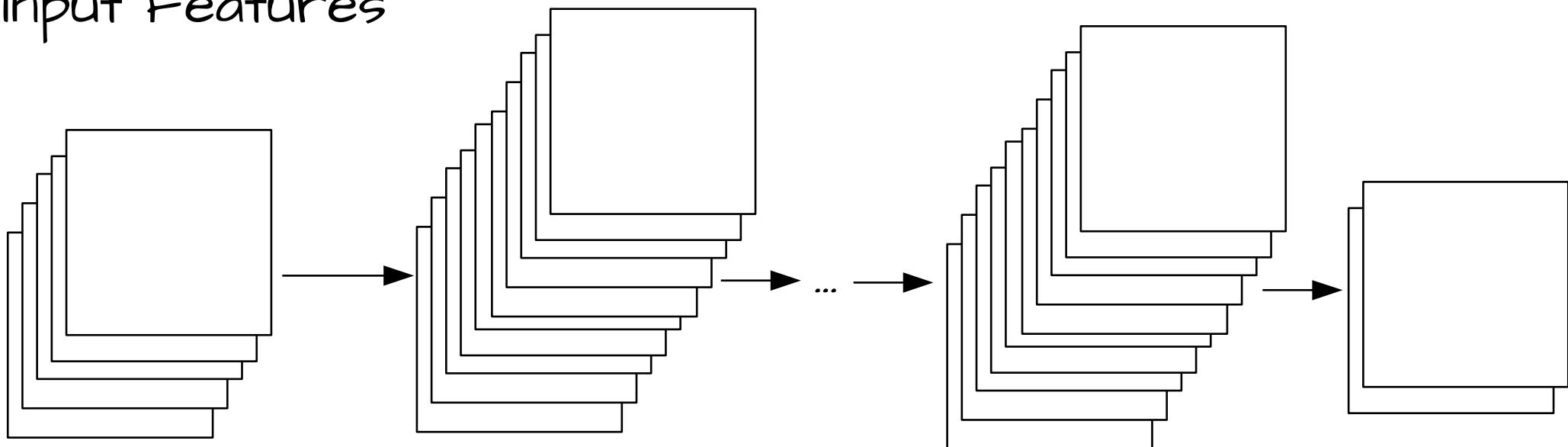


Architecture

12 layers with 64 - 192 filters

Output

Input Features

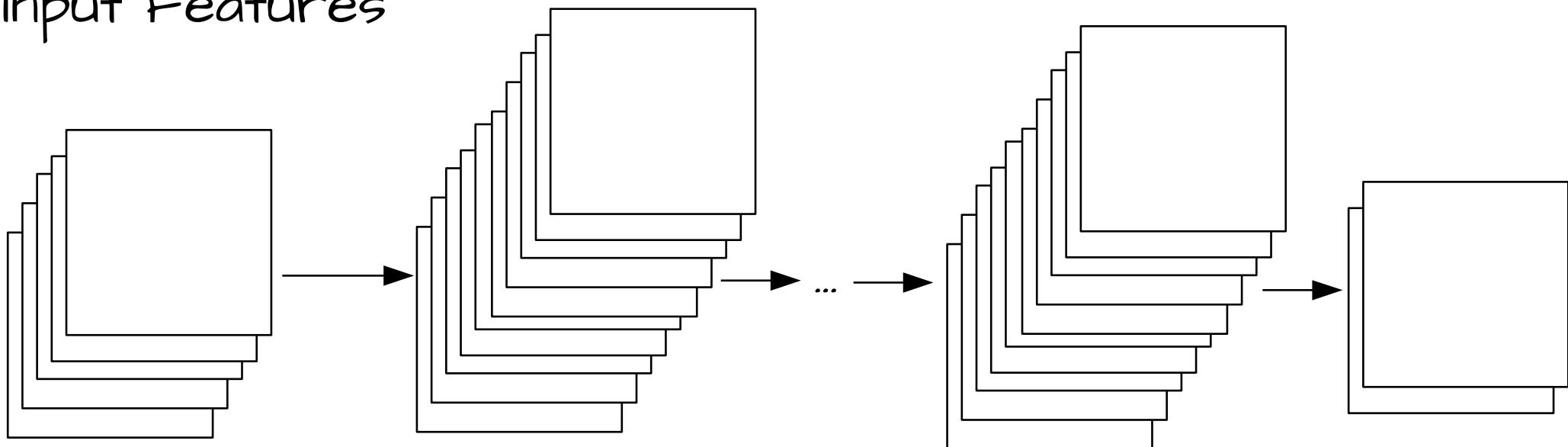


Architecture

12 layers with 64 - 192 filters

Output

Input Features



2.3 million parameters
630 million connections

Input Features

- Stone Colour x 3
- Liberties x 4
- Liberties after move played x 6
- Legal Move x 1
- Turns since x 5
- Capture Size x 7
- Ladder Move x 1
- KGS Rank x 9

Training on game data predicting
the next move

55% Accuracy

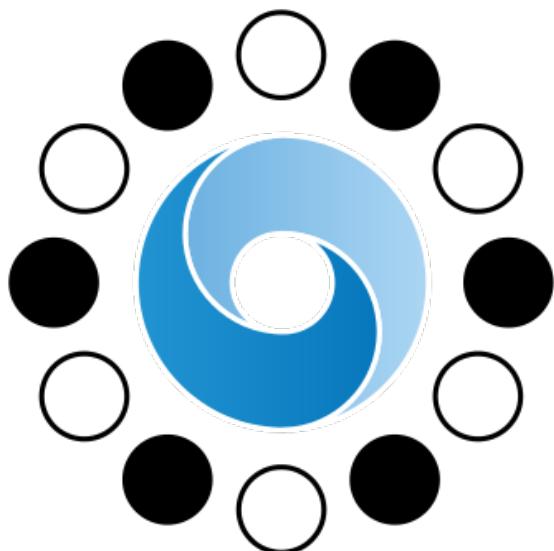
Mostly beats GnuGo

Combined with MCTS in the
Selection

Asynchronous GPU Power



Revolution



AlphaGo

Networks in Training

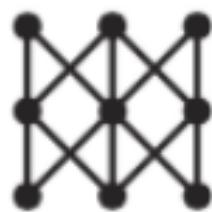
Rollout policy

$$P_\pi$$



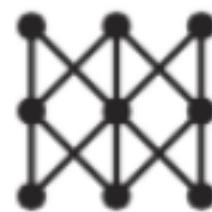
SL policy network

$$P_\sigma$$



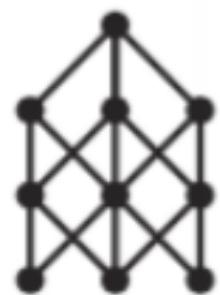
RL policy network

$$P_\rho$$

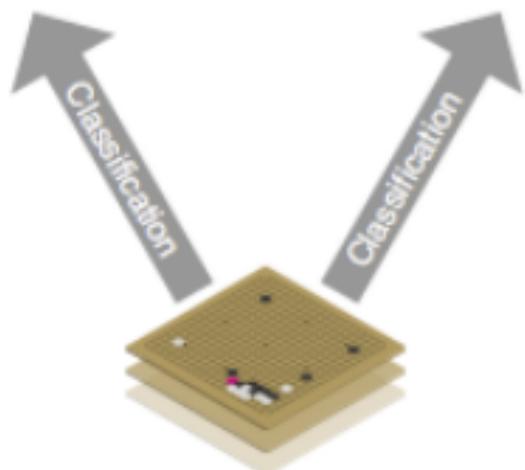


Value network

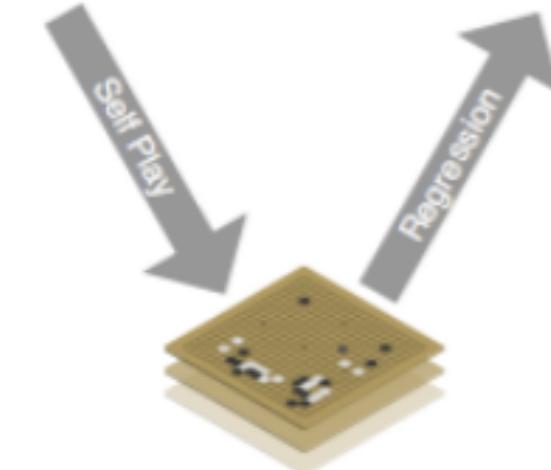
$$V_\theta$$



Policy Gradient



Human expert positions



Self-play Positions

Silver, D. et al., 2016. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), p.484-489.

Networks in Training

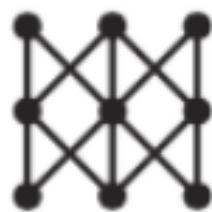
Rollout policy

$$P_\pi$$



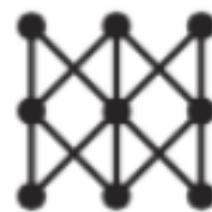
SL policy network

$$P_\sigma$$



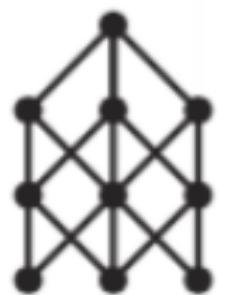
RL policy network

$$P_\rho$$

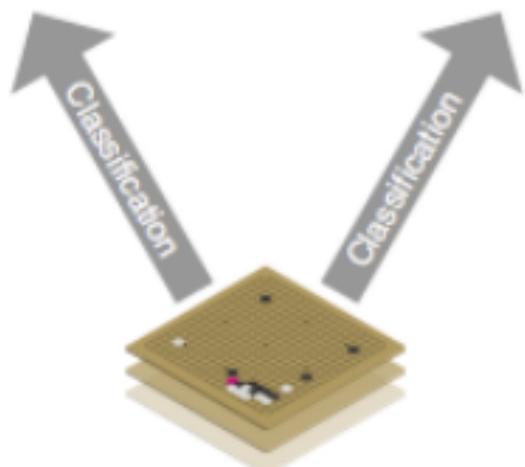


Value network

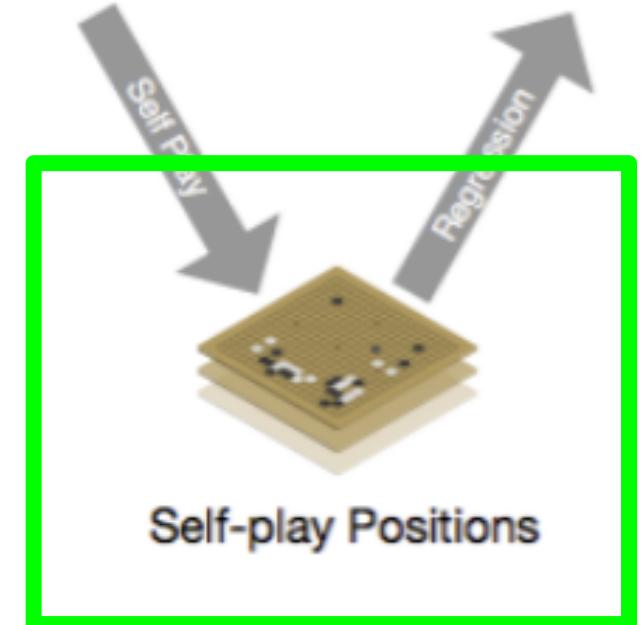
$$V_\theta$$



Policy Gradient



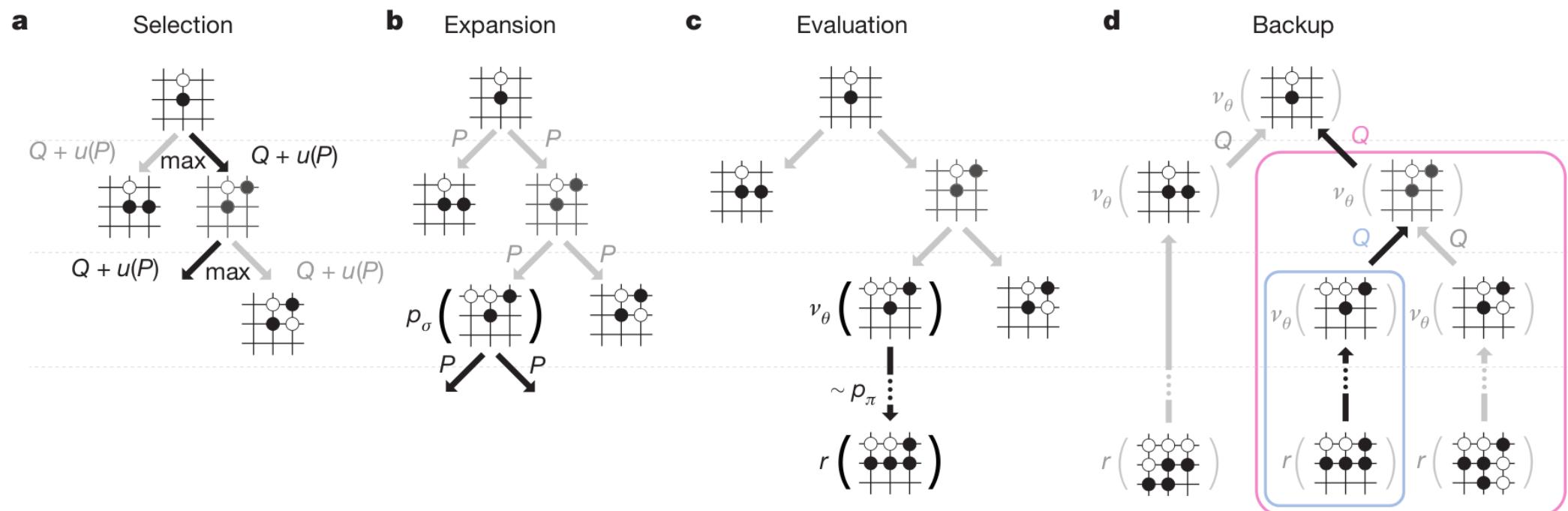
Human expert positions



Self-play Positions

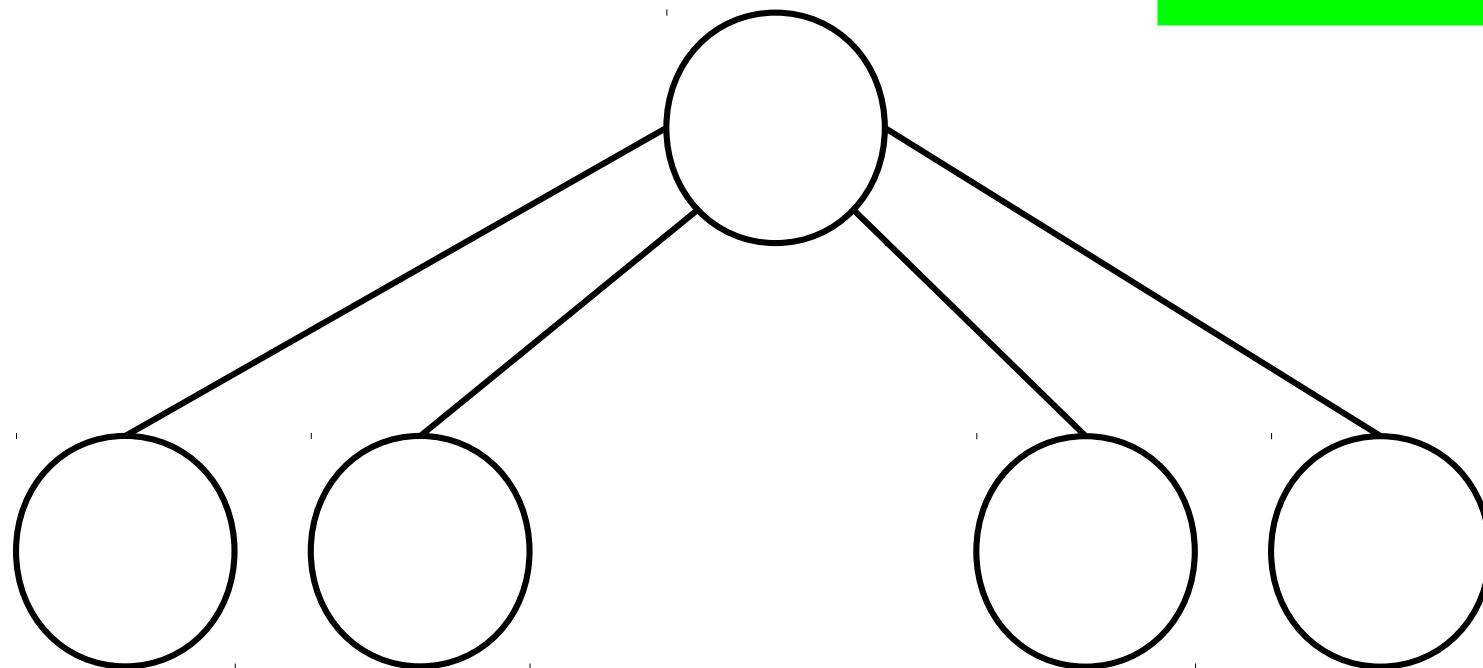
Silver, D. et al., 2016. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), p.484-489.

AlphaGo Search

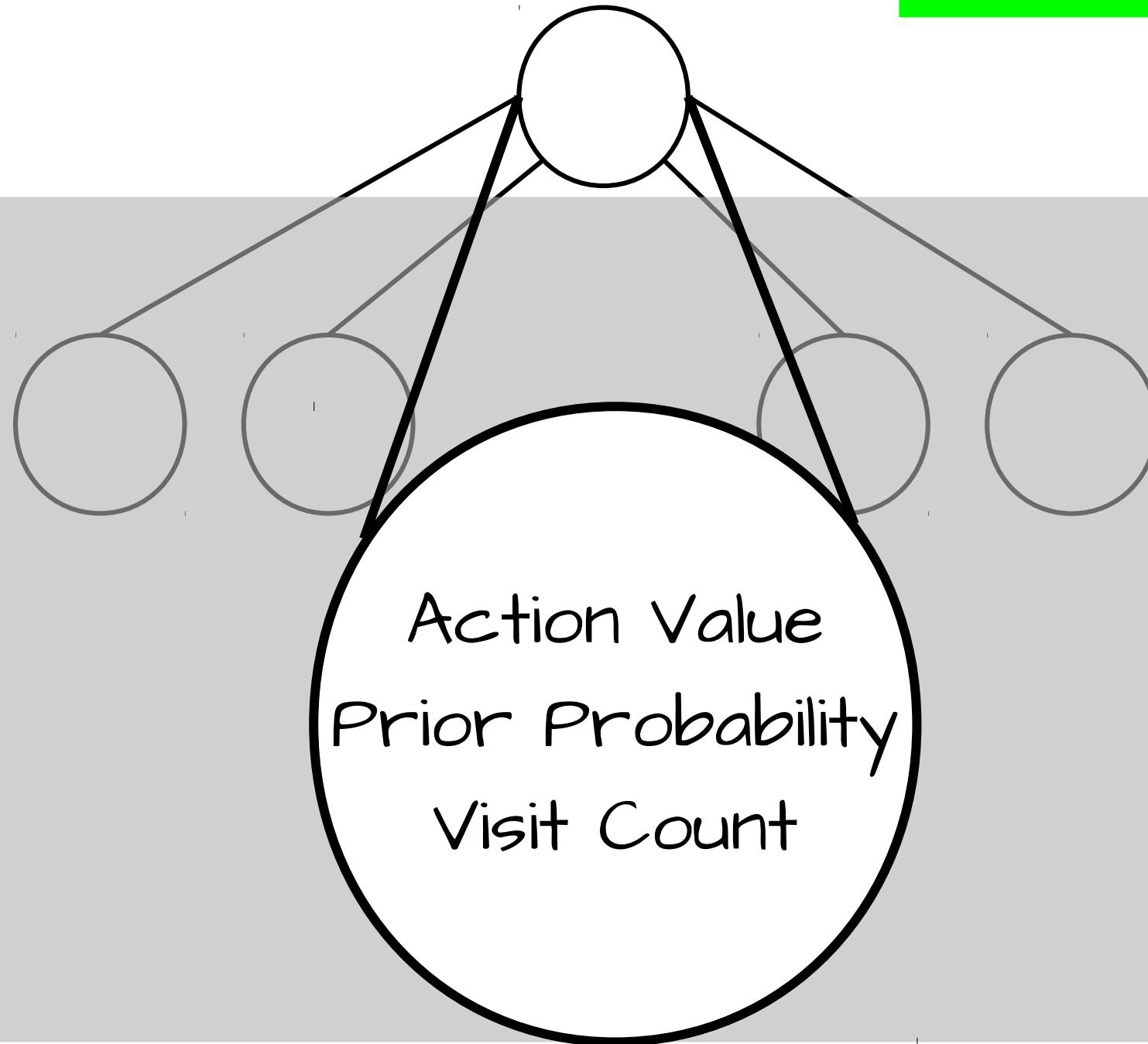


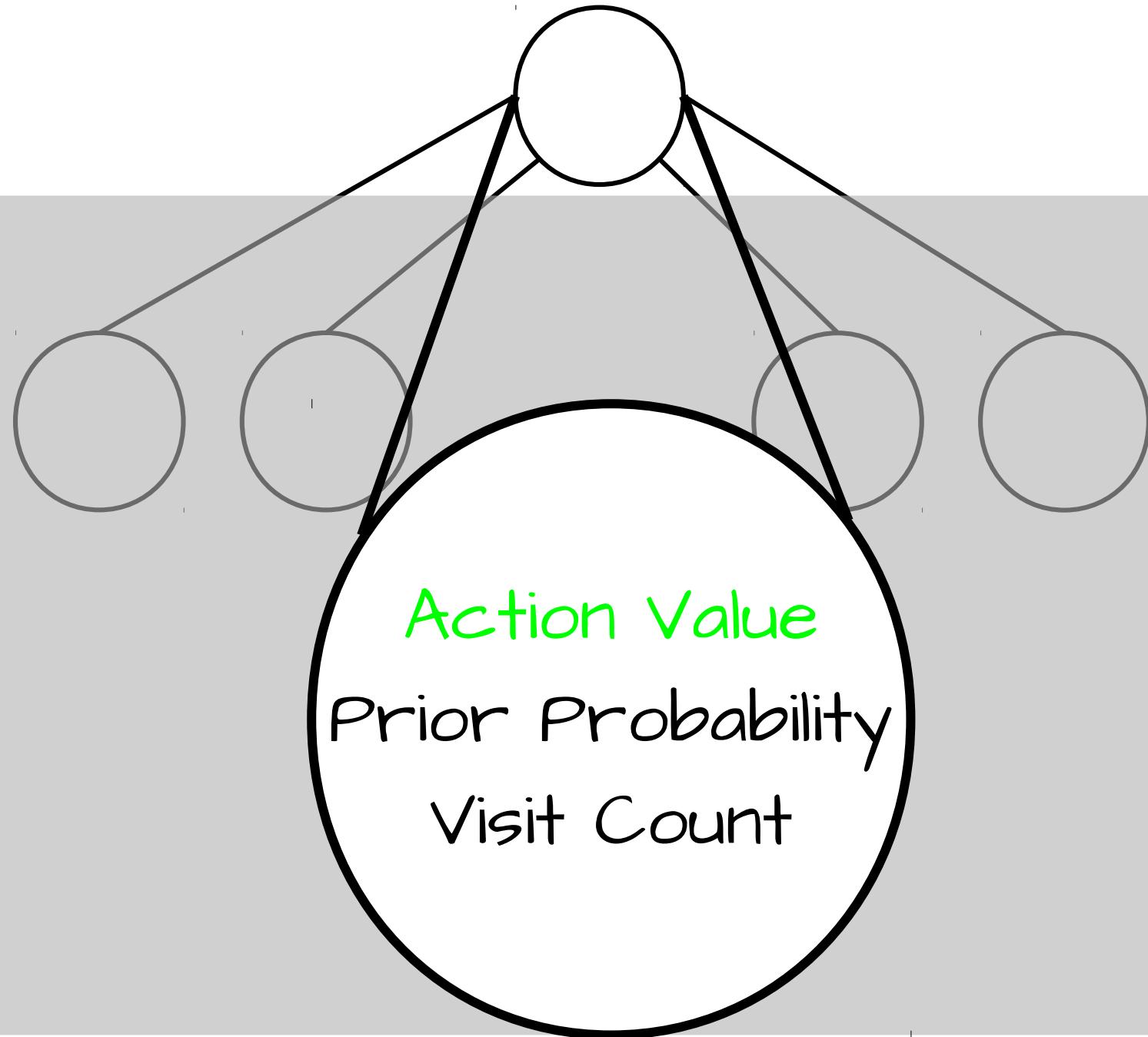
Silver, D. et al., 2016. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), p.484-489.

Selection

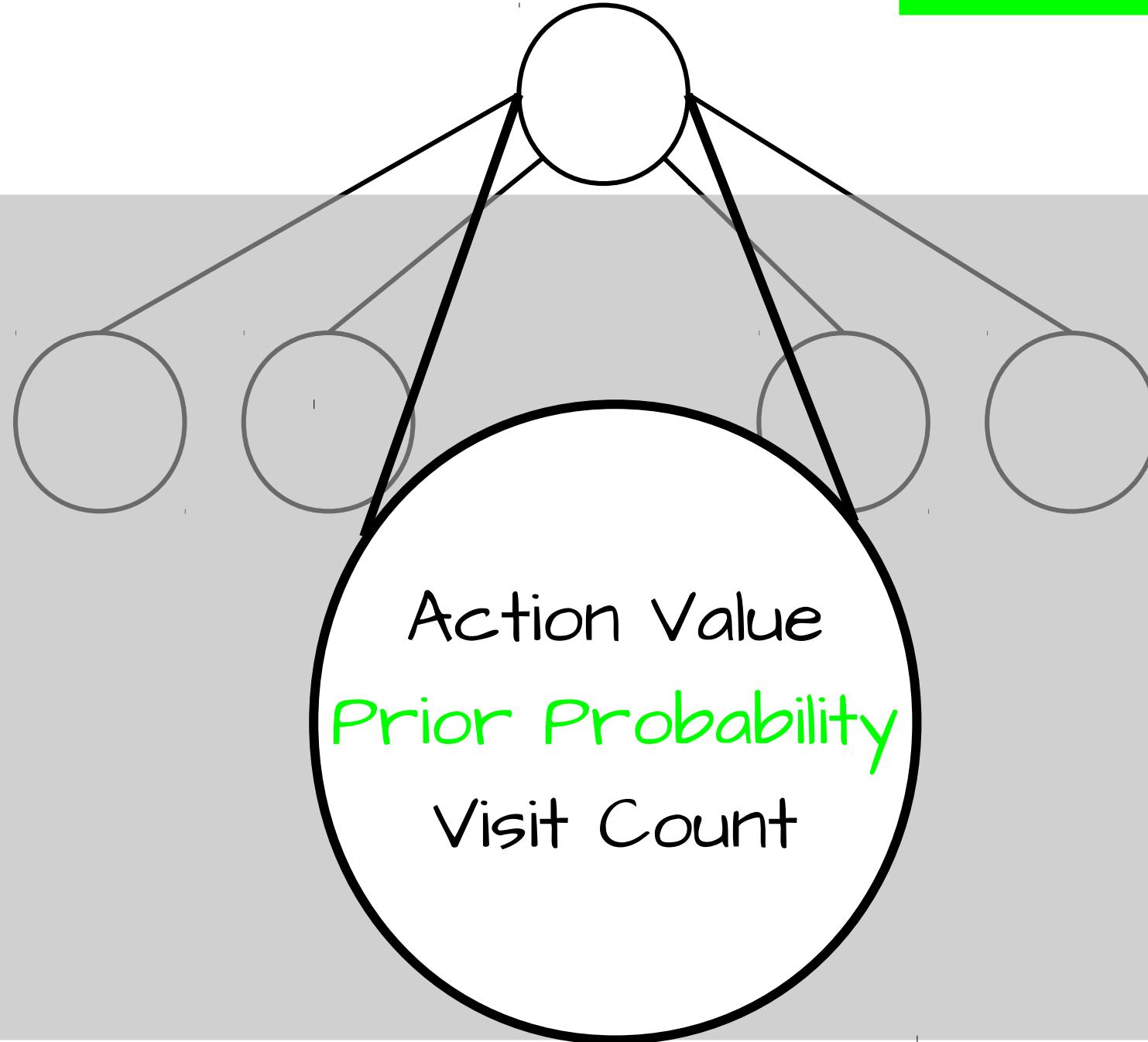


Selection

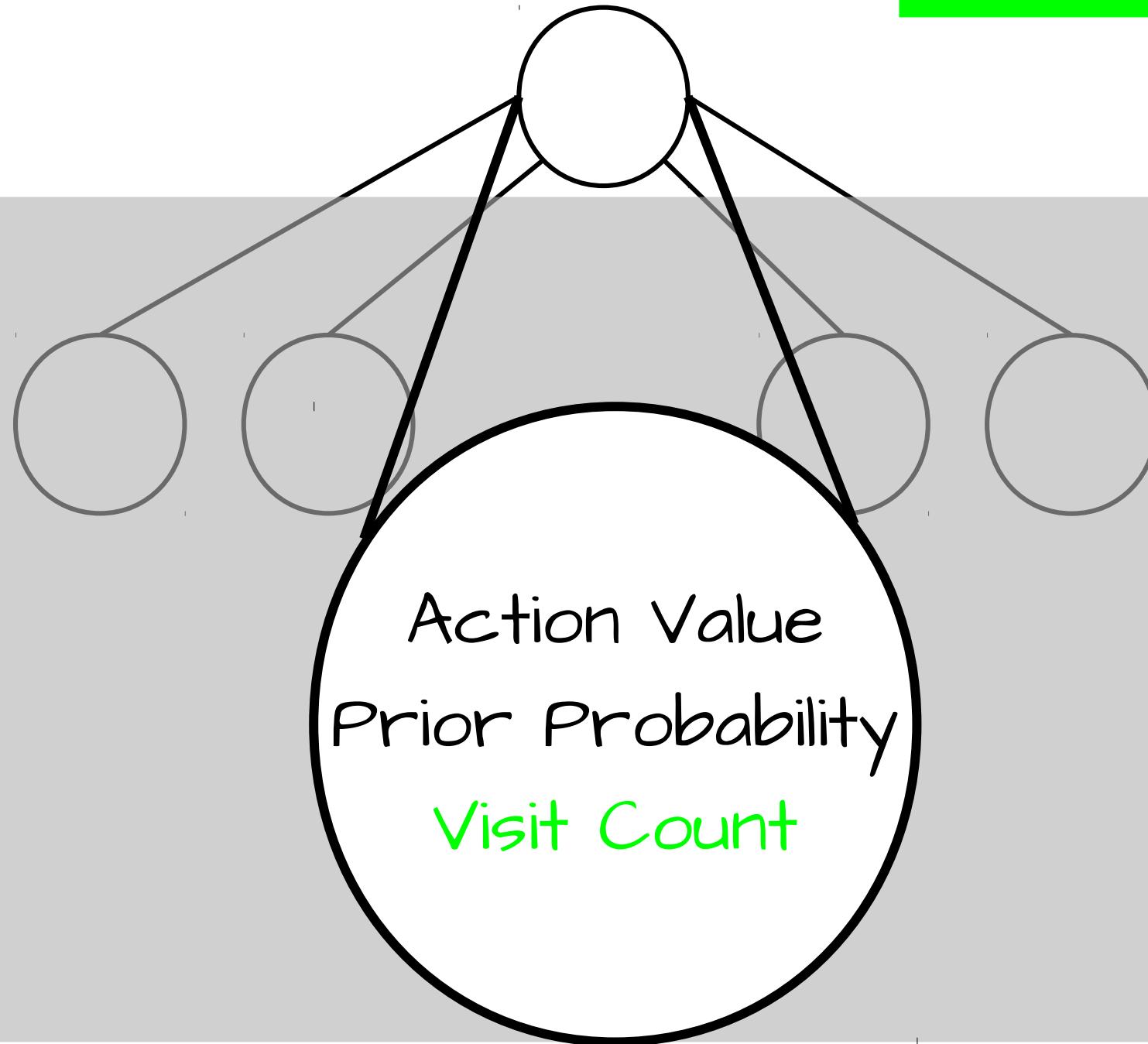




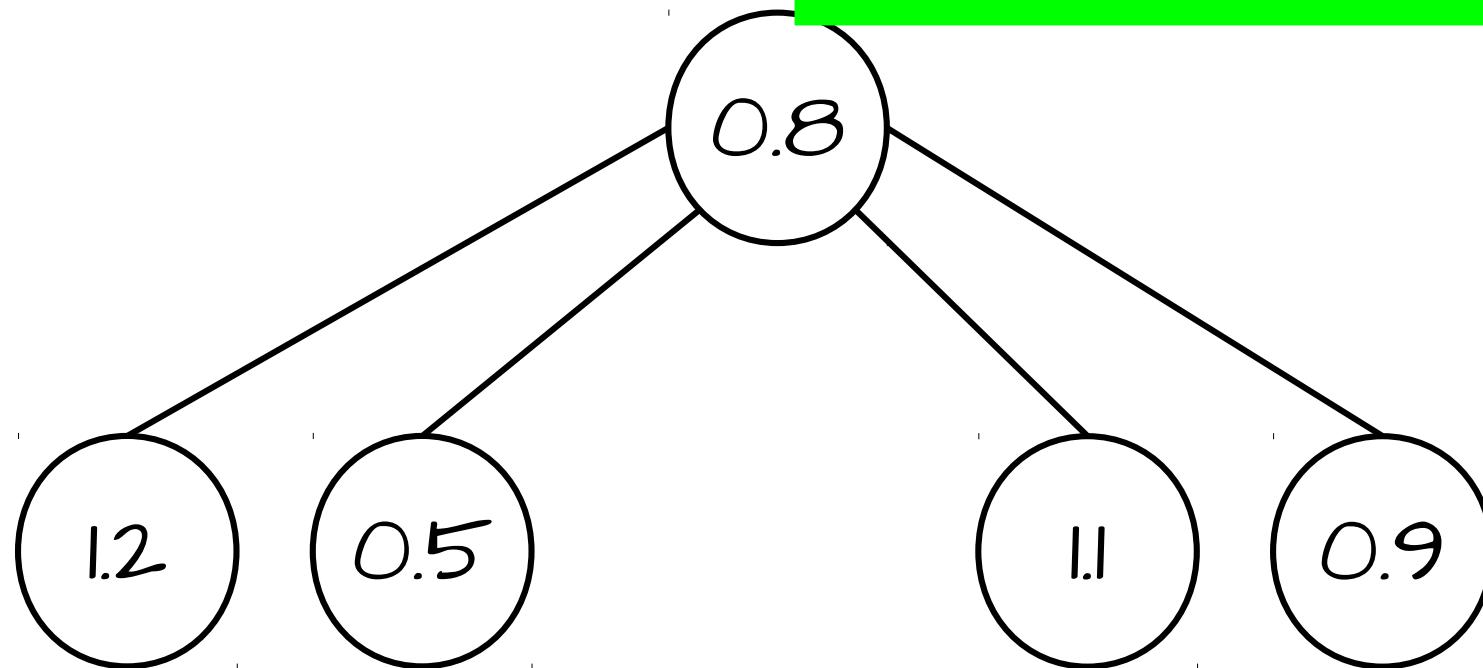
Selection



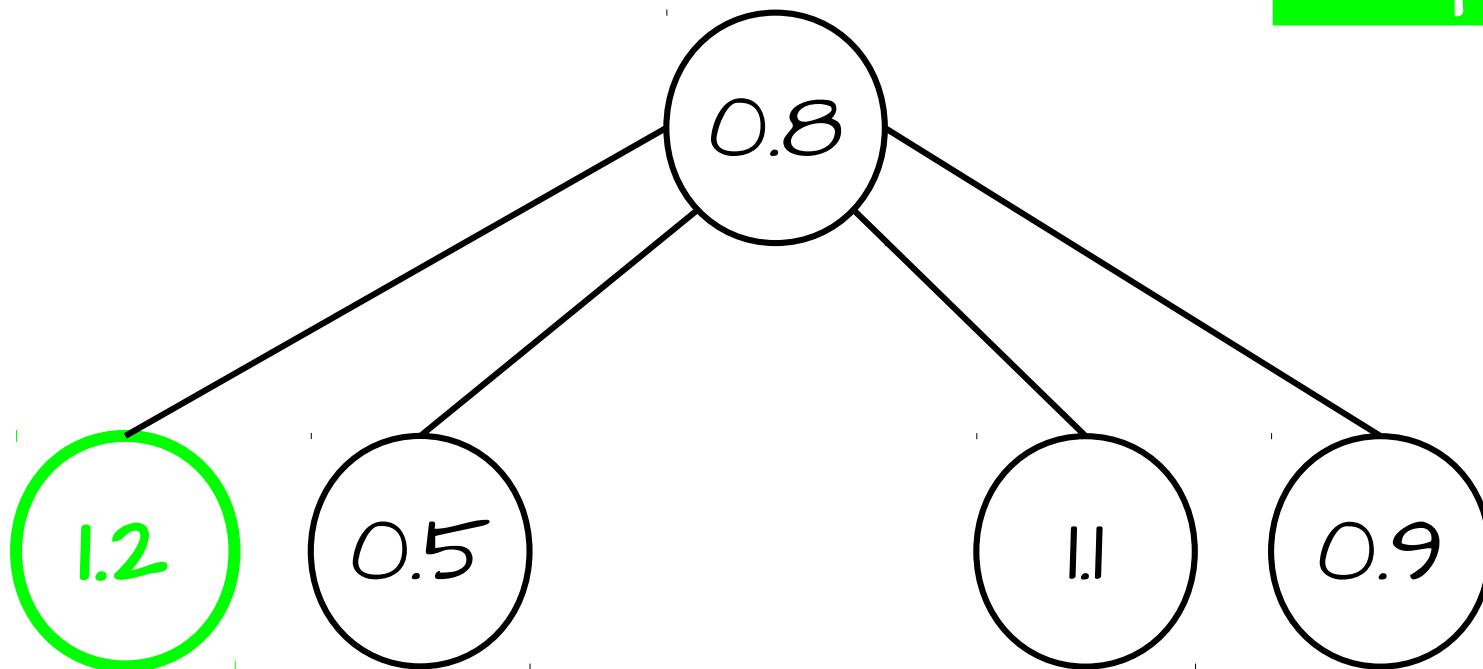
Selection



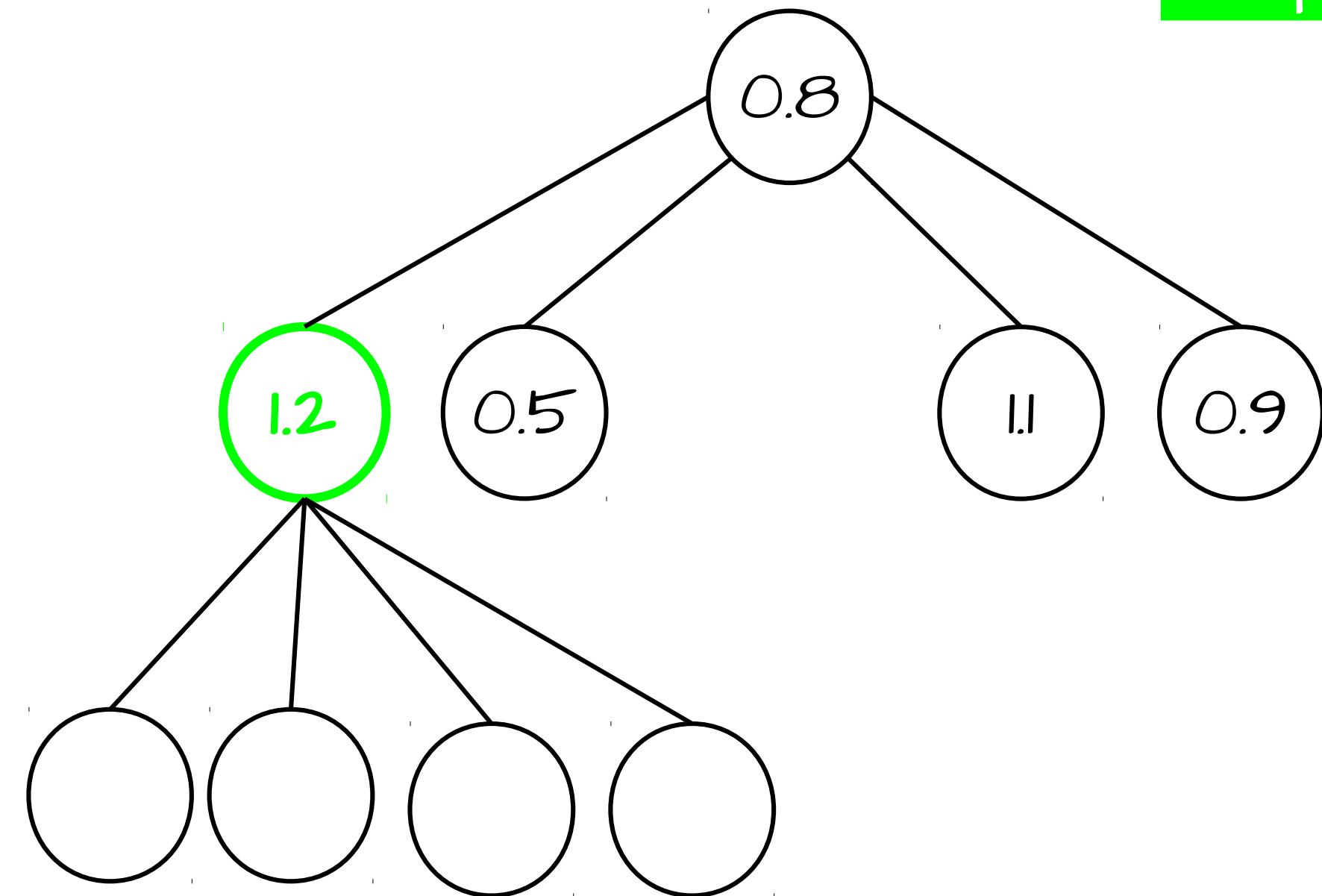
Action Value + Bonuses



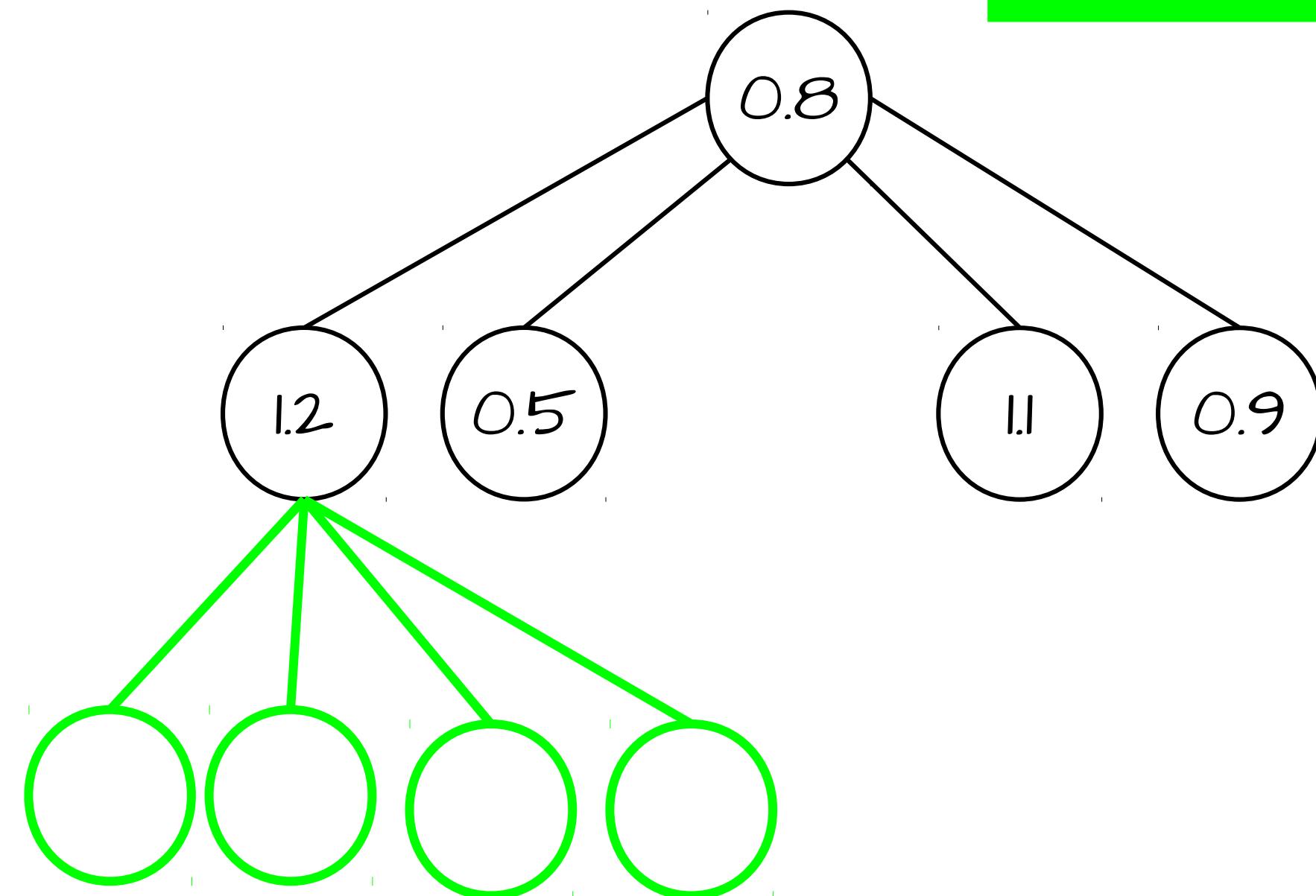
Expansion



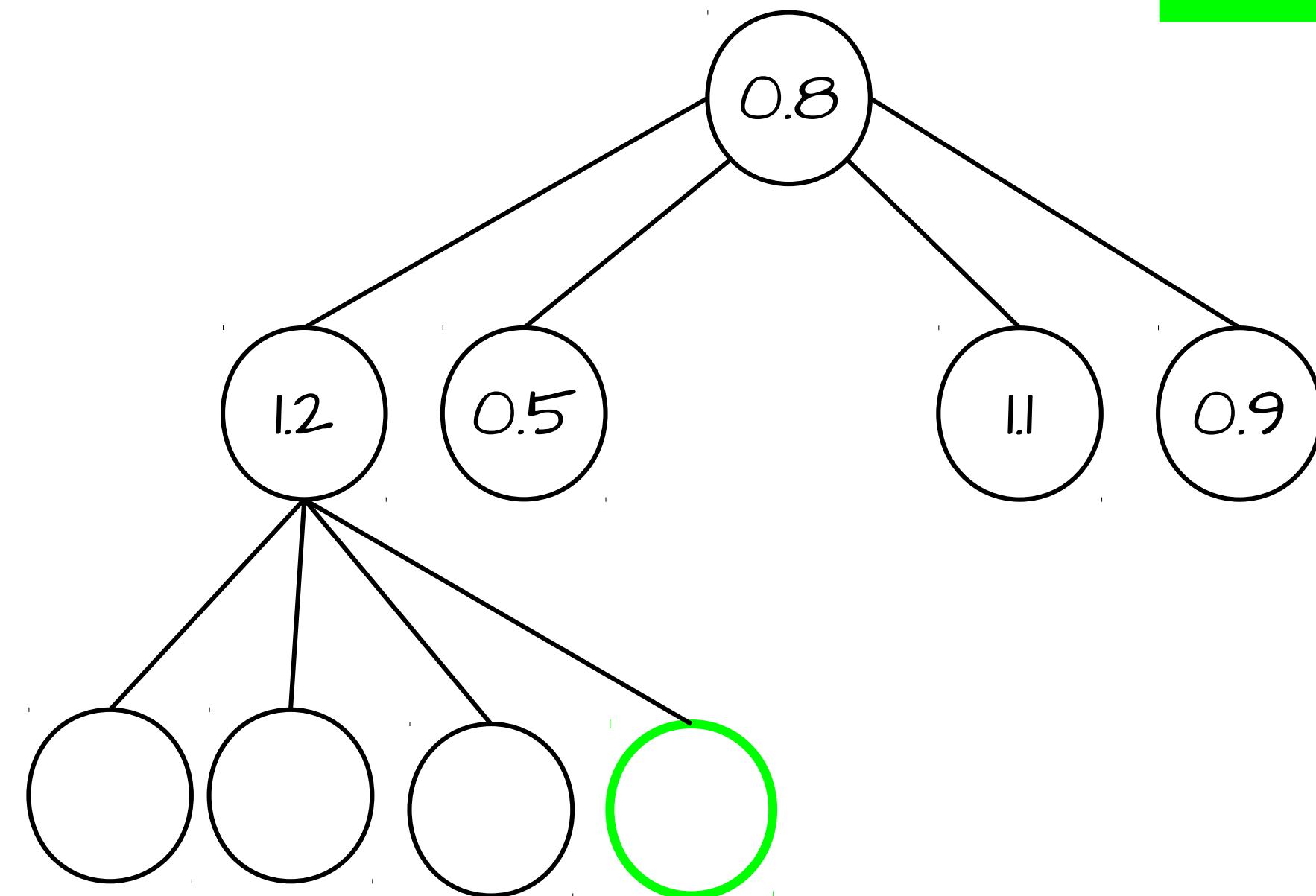
Expansion



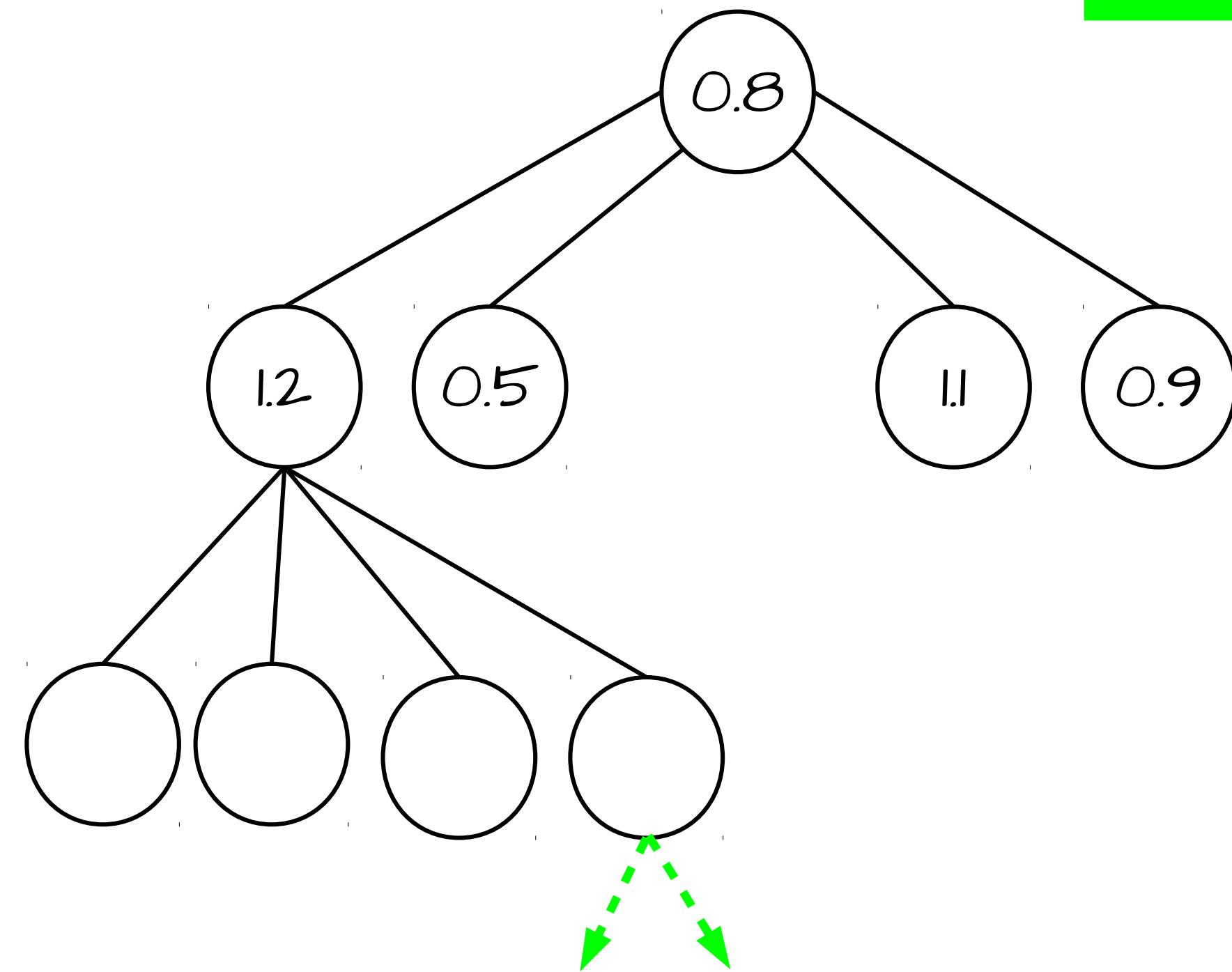
Prior Probability



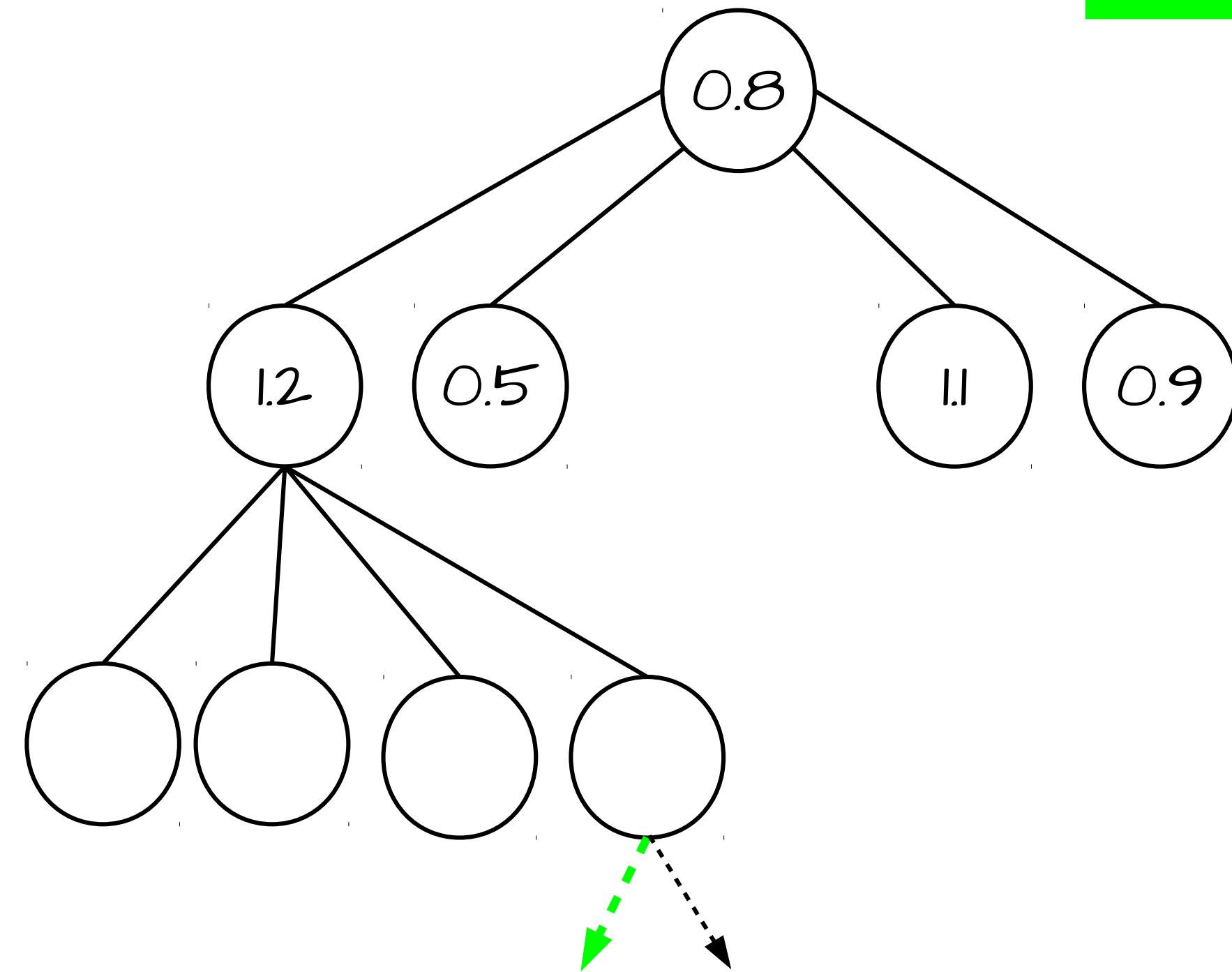
Evalution



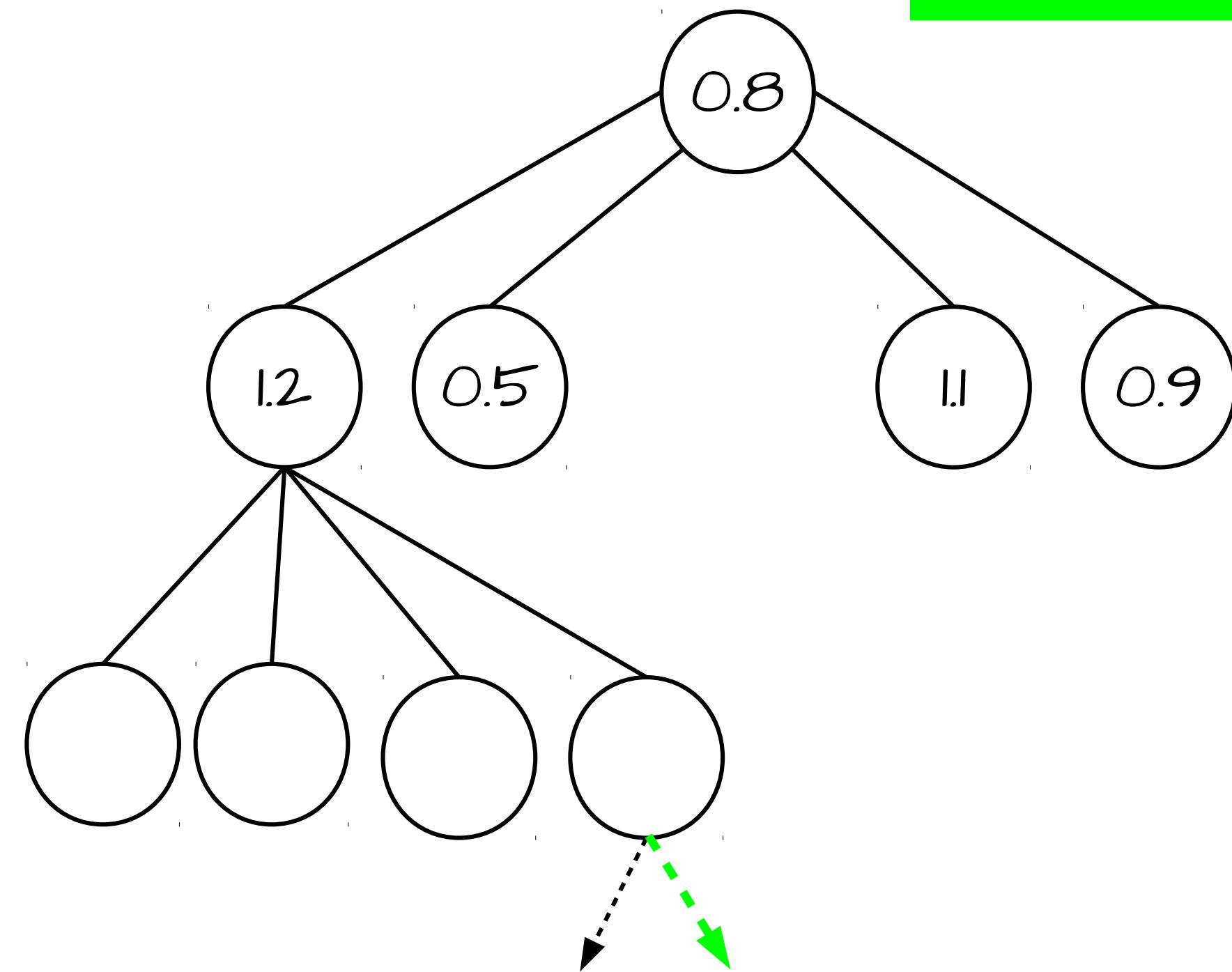
Evalution



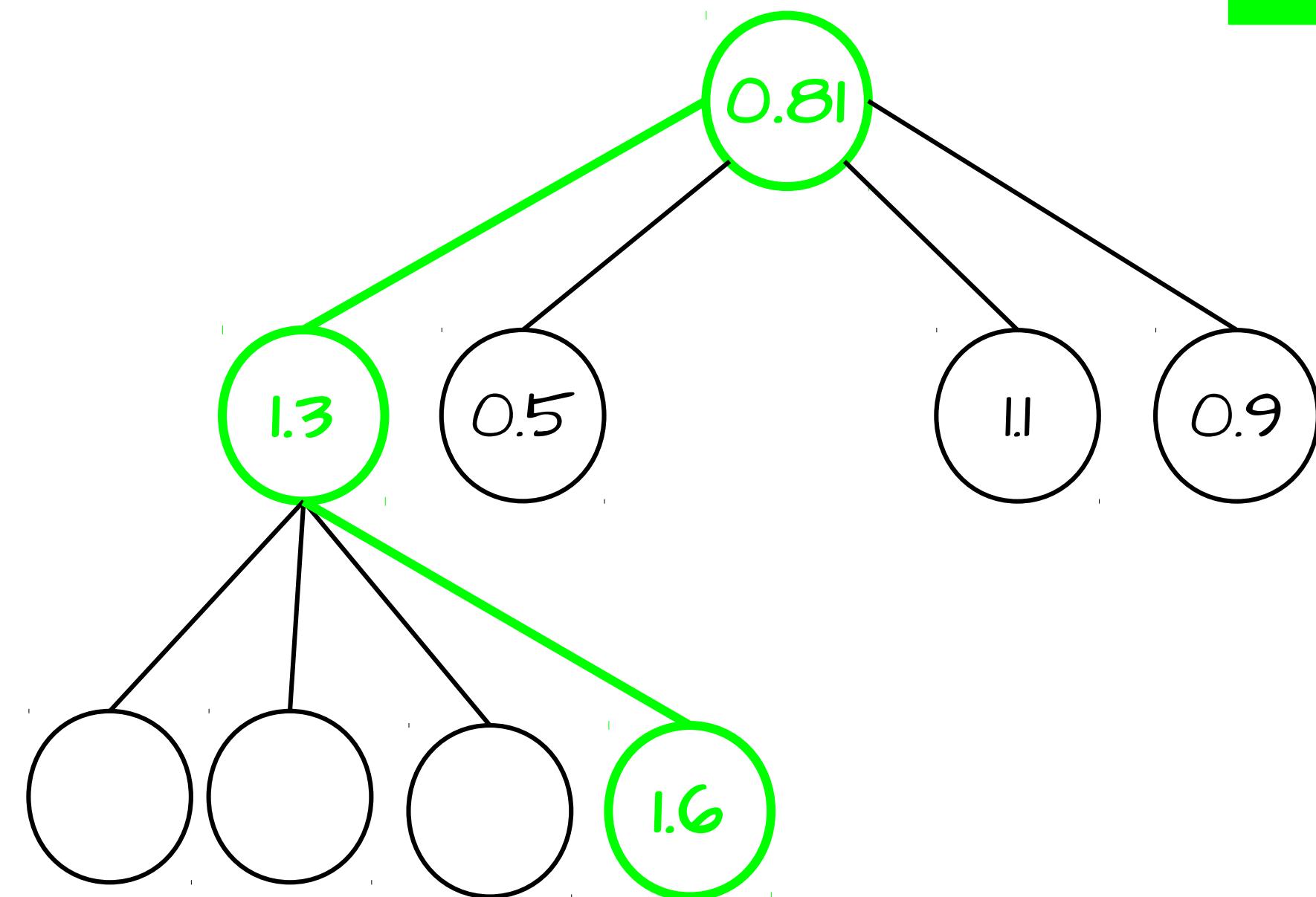
Rollout



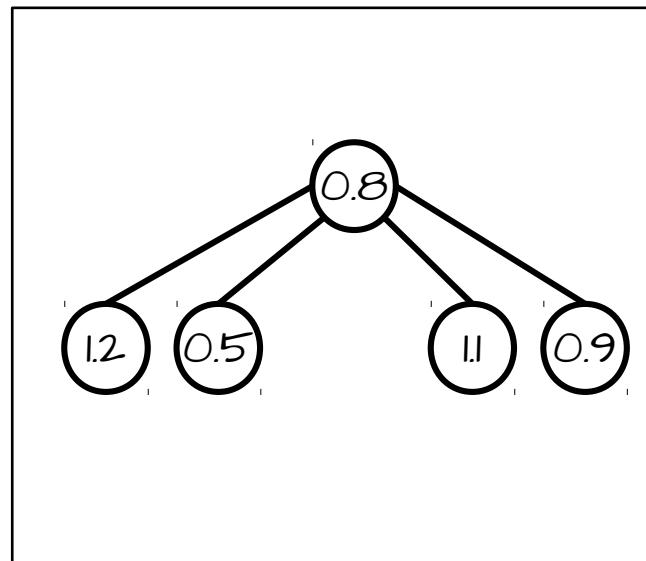
Value Network



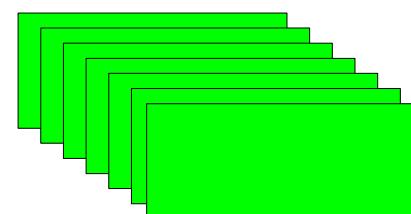
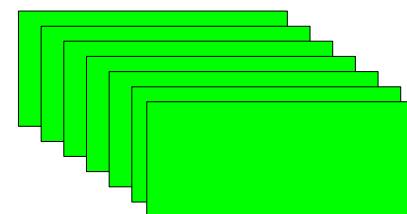
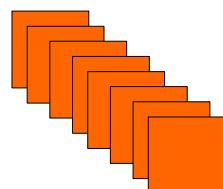
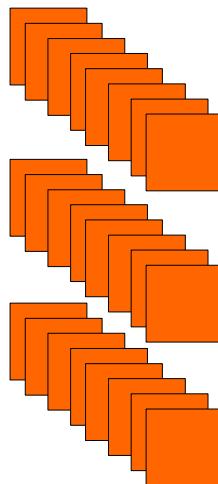
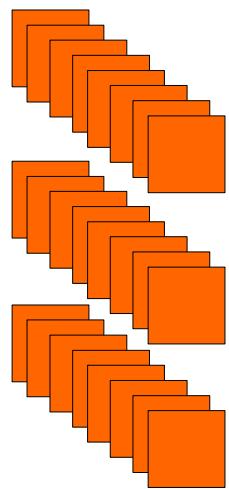
Backup



1202 CPUs



176 GPUs



Tensor



3 Strengths of AlphaGo

Human Instinct

Policy Network

Reading Capability

Search

Positional Judgement

Value Network

Most Important Strength

Human Instinct

Policy Network

Reading Capability

Search

Positional Judgement Value Network

More Natural



Lee Sedol match

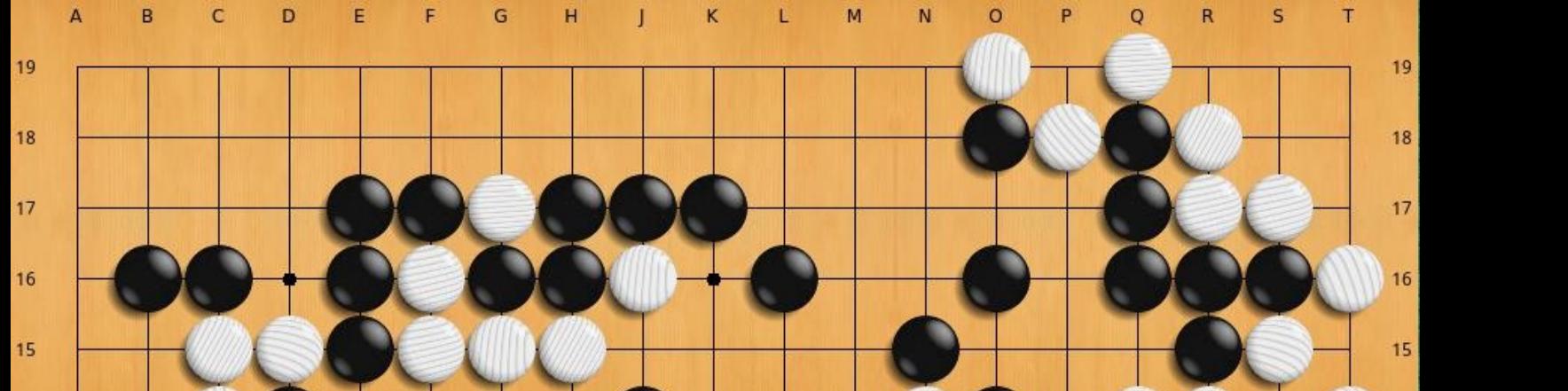


Style



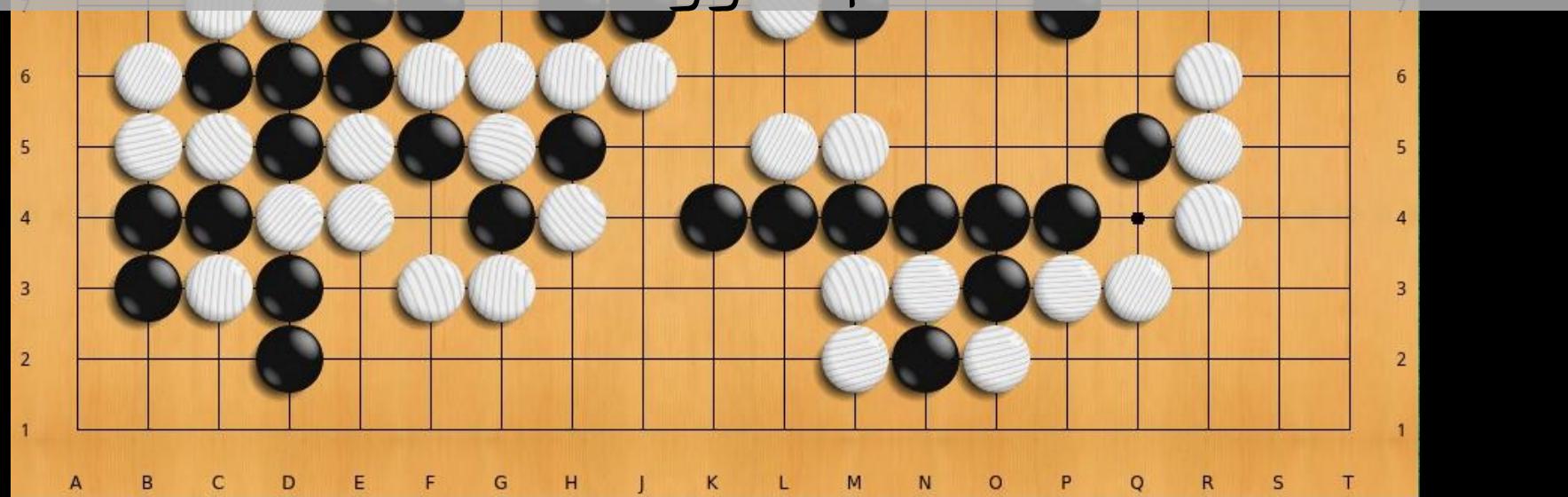
Game 2





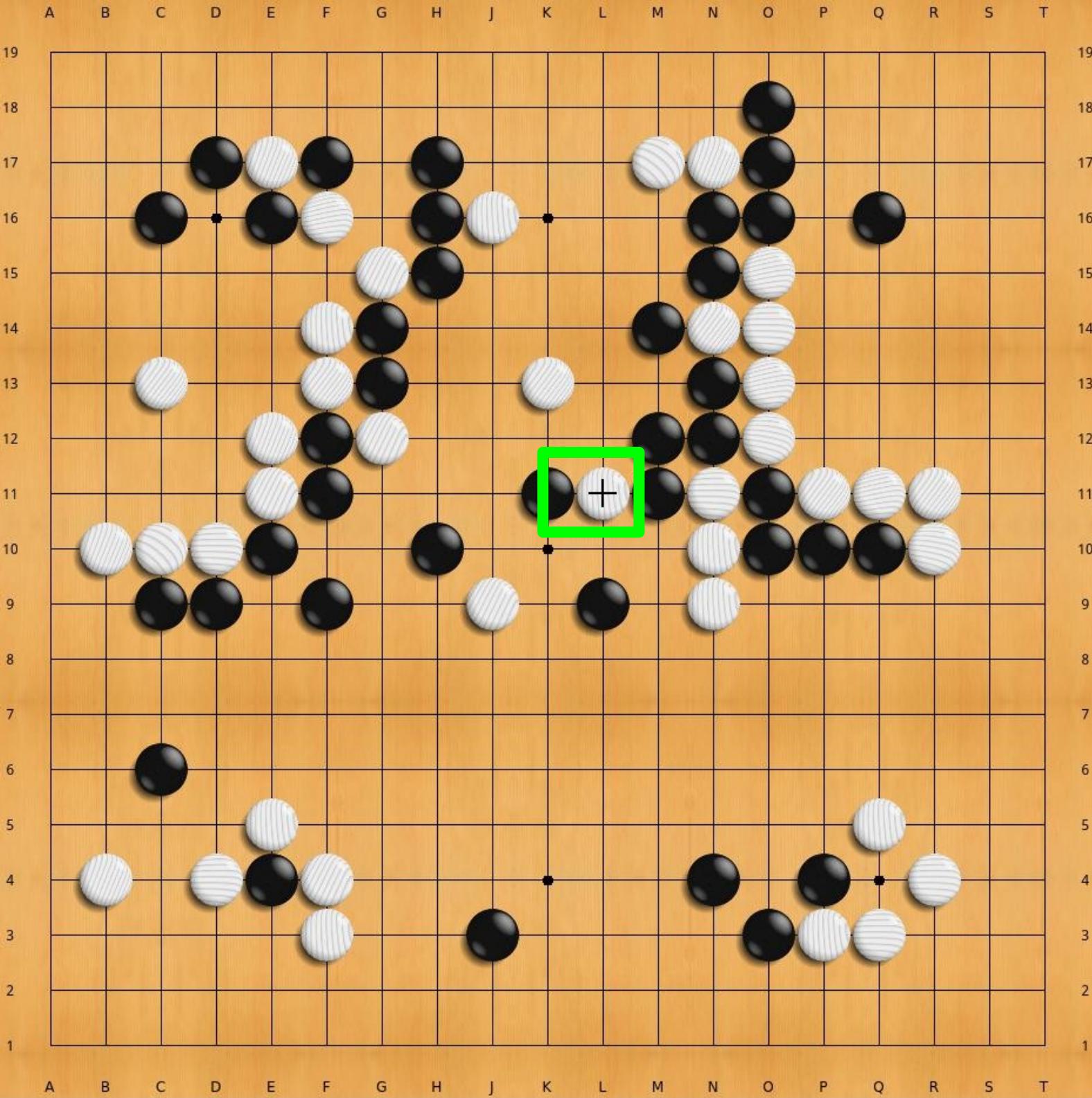
So when AlphaGo plays a *slack looking move*,
we may regard it as a mistake,
but perhaps it should more accurately be viewed
as a *declaration of victory*?

An Younggil 8p



Game 4





Game 4



Game 4



What can we learn?



Making X faster

vs

Doing less of X

Benchmark everything

Solving problems the **human** way

vs

Solving problems the **computer**
way

Don't **blindly dismiss** approaches
as infeasible

One Approach

vs

Combination of Approaches

Joy of Creation

```

white's turn to move!
Make a move in the form XY, e.g. A19, D7 as the labels indicate!
> F8
A B C D E F G H J
9 . . . . . . . . 9
8 . X . . . 0 X . . 8
7 . 0 0 . . . 0 . . 7
6 . . . 0 . . . . 6
5 . 0 0 X . . . 0 . 5
4 . X X . . . . . 4
3 . . X . . . X . . 3
2 . . . . . X . . 2
1 . . . . . . . . 1
A B C D E F G H J
white played at F8
black's turn to move!
Rubykon is thinking...
D2 => 0.4881516587677725
C8 => 0.4827586206896552
F4 => 0.4827586206896552
F5 => 0.47761194029850745
G4 => 0.47474747474747475
E8 => 0.4744897959183674
G6 => 0.47150259067357514
D6 => 0.46875
D4 => 0.4627659574468085
F7 => 0.46236559139784944
A B C D E F G H J
9 . . . . . . . . 9
8 . X . . . 0 X . . 8
7 . 0 0 Rubykon.rb 0 . . 7
6 . . . 0 . . . . 6
5 . 0 0 X . . . 0 . 5
4 . X X . . . . . 4
3 . . X . . . X . . 3
2 . . X . . . X . 2
1 . . . . . . . . 1
A B C D E F G H J
black played at D2
white's turn to move!
Make a move in the form XY, e.g. A19, D7 as the labels indicate!
> F5
M CHANGELOG.md
A B C D E F G H J
9 . . . . . . . . 9
8 . X . . . 0 X . . 8
7 . 0 0 . . . 0 . . 7
6 . Guard! 0 . . . . 6
5 . 0 0 X . 0 . 0 . 5
4 . X X . . . . . 4
3 . . X . . . X . . 3
2 . . X . . . X . 2
1 . . package! . . . . 1
A B C D E F G H J
white played at F5
black's turn to move!
Rubykon is thinking...

```

```

1 # coding: utf-8
2 $LOAD_PATH.unshift(File.join(File.dirname(__FILE__), '..', 'lib'))
3 $LOAD_PATH.unshift(lib) unless $LOAD_PATH.include?(lib)
4 require 'rubykon/version'
5
6 Gem::Specification.new do |spec|
7   spec.name      = "rubykon"
8   spec.version   = Rubykon::VERSION
9   spec.authors   = ["Tobias Pfeiffer"]
10  spec.email     = ["pragtob@gmail.com"]
11
12  spec.summary   = %q{An AI to play Go using Monte Carlo Tree Search.}
13  spec.description = %q{An AI to play Go using Monte Carlo Tree Search. Currently includes the mcts gem and benchmark/avg. Works on all major ruby versions.}
14  spec.homepage  = "https://github.com/PragTob/rubykon"
15  spec.license    = "MIT"
16
17  spec.files     = `git ls-files -z`.split("\x0").reject { |f| f.match(%r{^(test|spec|features)/}) }
18  spec.bindir     = "exe"
19  spec.executables = spec.filesgrep(%r{^exe/}) { |f| File.basename(f) }
20  spec.require_paths = ["lib"]
21
22

```

PragTob/Rubykon

Outer ignore rules:
.idea

Minimalistic Go MCTS Engine

 117 commits 2 branches 0 releases 5 contributorsBranch: **master** ▾**michi** / +

Fork this project and create a new file

**pasky** Merge pull request #5 from traveller42/master

Latest commit 2f22d84 18 days



.gitignore Keep only repository specific entries in repository .gitignore

a month



README.md Update links to michi-c and michi-c2

a month



michi.py Test for Pass in the GTP code was too strict

18 days

pasky/michi

Michi --- Minimalistic Go MCTS Engine

Michi aims to be a minimalistic but full-fledged Computer Go program based on state-of-art methods (Monte Carlo Tree Search) and written in Python. Our goal is to make it easier for new people to enter the domain of Computer Go, peek under the hood of a "real" playing engine and be able to learn by

What did AlphaGo do to beat the strongest human Go player?

Tobias Pfeiffer

@PragTob

pragtob.info



LIEFERY

Sources

- Maddison, C.J. et al., 2014. Move Evaluation in Go Using Deep Convolutional Neural Networks.
- Silver, D. et al., 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), p.484-489.
- Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015
<http://neuralnetworksanddeeplearning.com>
- Gelly, S. & Silver, D., 2011. Monte-Carlo tree search and rapid action value estimation in computer Go. *Artificial Intelligence*, 175(11), p.1856-1876.
- I. Althöfer, “On the Laziness of Monte-Carlo Game Tree Search In Non-tight Situations,” Friedrich-Schiller Univ., Jena, Tech. Rep., 2008.
- Browne, C. & Powley, E., 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Intelligence and AI in Games*, 4(1), p.1-49.
- Gelly, S. & Silver, D., 2007. Combining online and offline knowledge in UCT. *Machine Learning*, p.273-280.
- <https://www.youtube.com/watch?v=LX8Knl0g0LE&index=9&list=WL>

Photo Credit

- <http://www.computer-go.info/events/ing/2000/images/bigcup.jpg>
- <https://en.wikipedia.org/wiki/File:Kasparov-29.jpg>
- <http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-black/product-images>
- <http://giphy.com/gifs/dark-thread-after-ICP95tGSbMmWI>
- <https://cloudplatform.googleblog.com/2016/05/Google-supercharges-machine-learning-tasks-with-custom-chip.html>
- <https://gogameguru.com/i/2016/01/Fan-Hui-vs-AlphaGo-550x364.jpg>
- <http://makeitstranger.com/>
- CC BY 2.0
 - https://en.wikipedia.org/wiki/File:Deep_Blue.jpg
 - <https://www.flickr.com/photos/luisbg/2094497611/>
- CC BY-SA 3.0
 - https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning#/media/File:AB_pruning.svg
- CC BY-SA 2.0
 - <https://flic.kr/p/cPUtny>
 - <https://flic.kr/p/dLSKTQ>
 - <https://www.flickr.com/photos/83633410@N07/7658272558/>

Photo Credit

- CC BY-NC-ND 2.0
 - <https://flic.kr/p/q15pzb>
 - <https://flic.kr/p/bHSj7D>
 - <https://flic.kr/p/ixSsfM>
 - <https://www.flickr.com/photos/waxorian/4228645447/>
 - <https://www.flickr.com/photos/pennstatelive/8972110324/>
 - <https://www.flickr.com/photos/dylanstraub/6428496139/>
- https://en.wikipedia.org/wiki/Alphabet_Inc._Logo_2015.svg
- CC BY 3.0
 - https://en.wikipedia.org/wiki/File:Pi_30K.gif