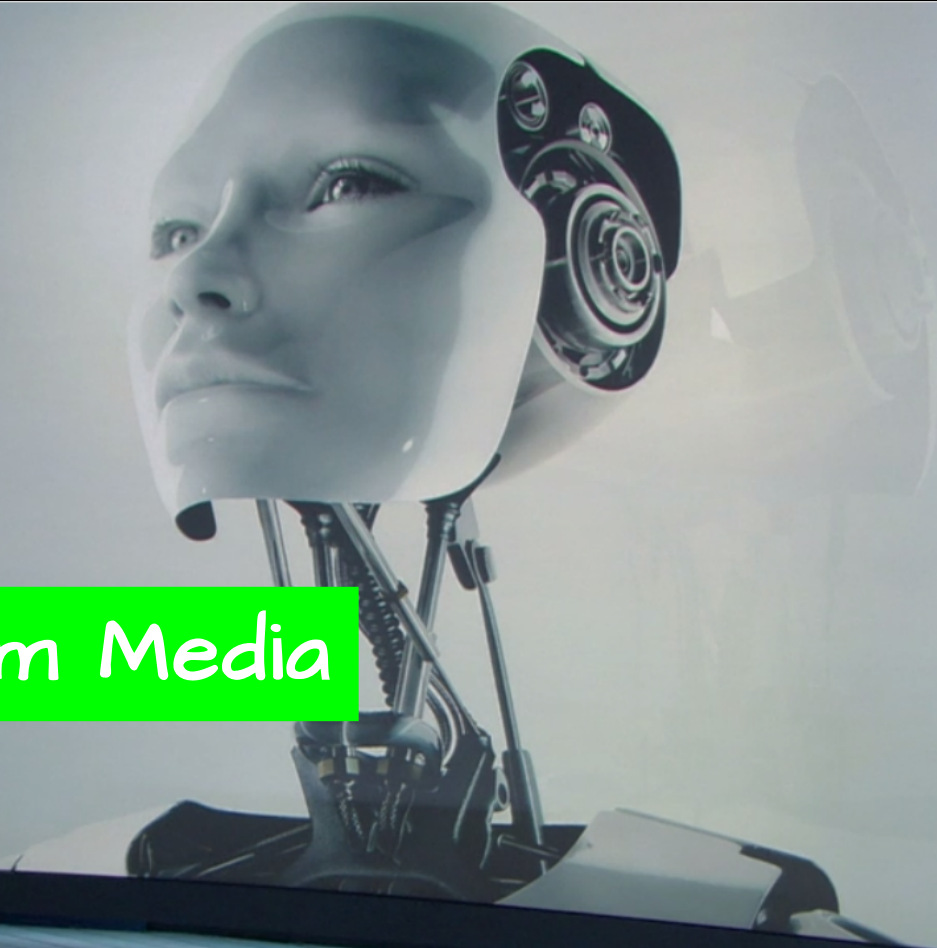


March 2016





# Mainstream Media





1997

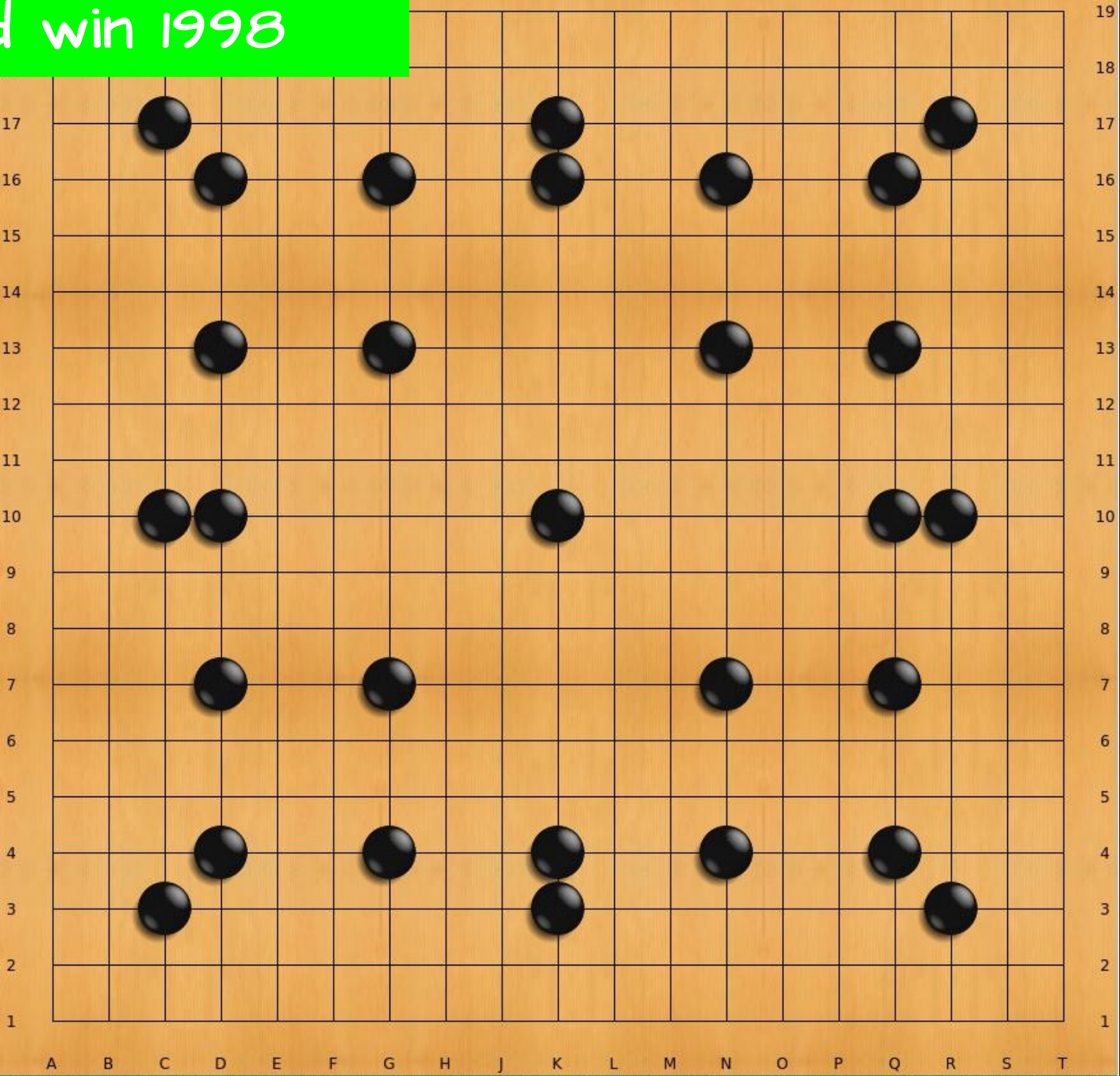




(1985-2000)



5d win 1998



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



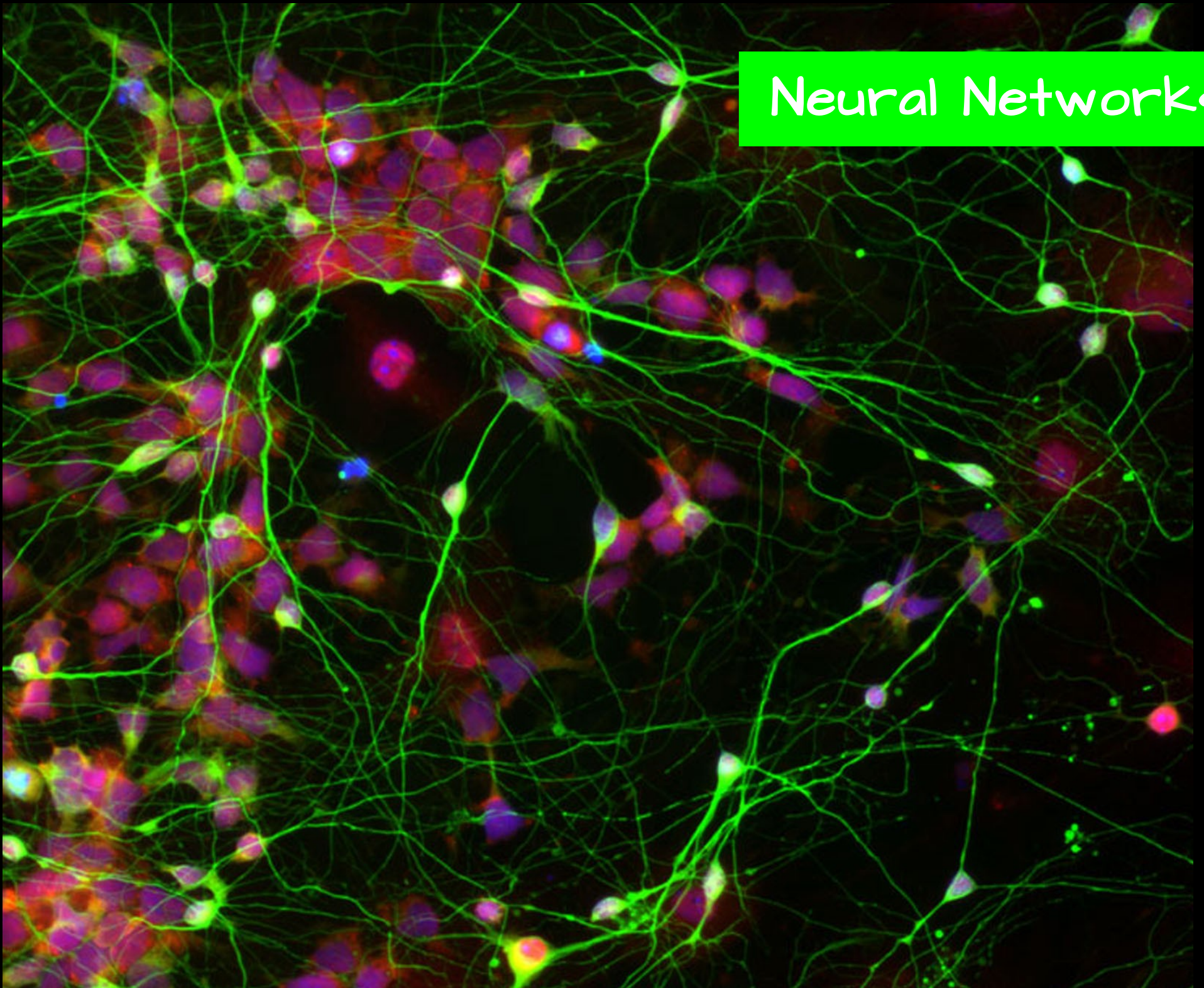


Computational Challenge

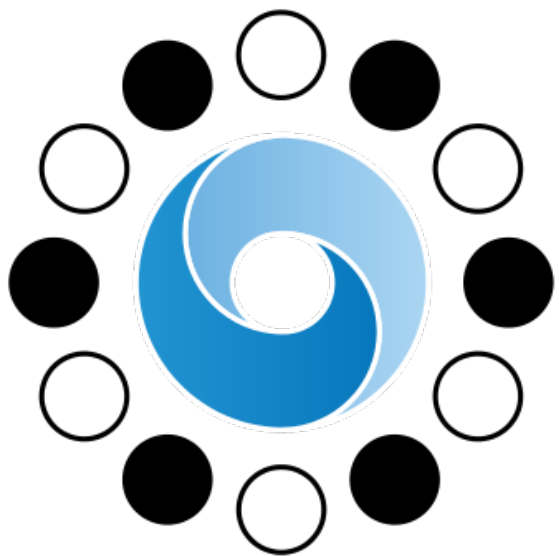
# Monte Carlo Method



# Neural Networks



Revolution with Neural Networks



AlphaGo

What did we learn?



Go











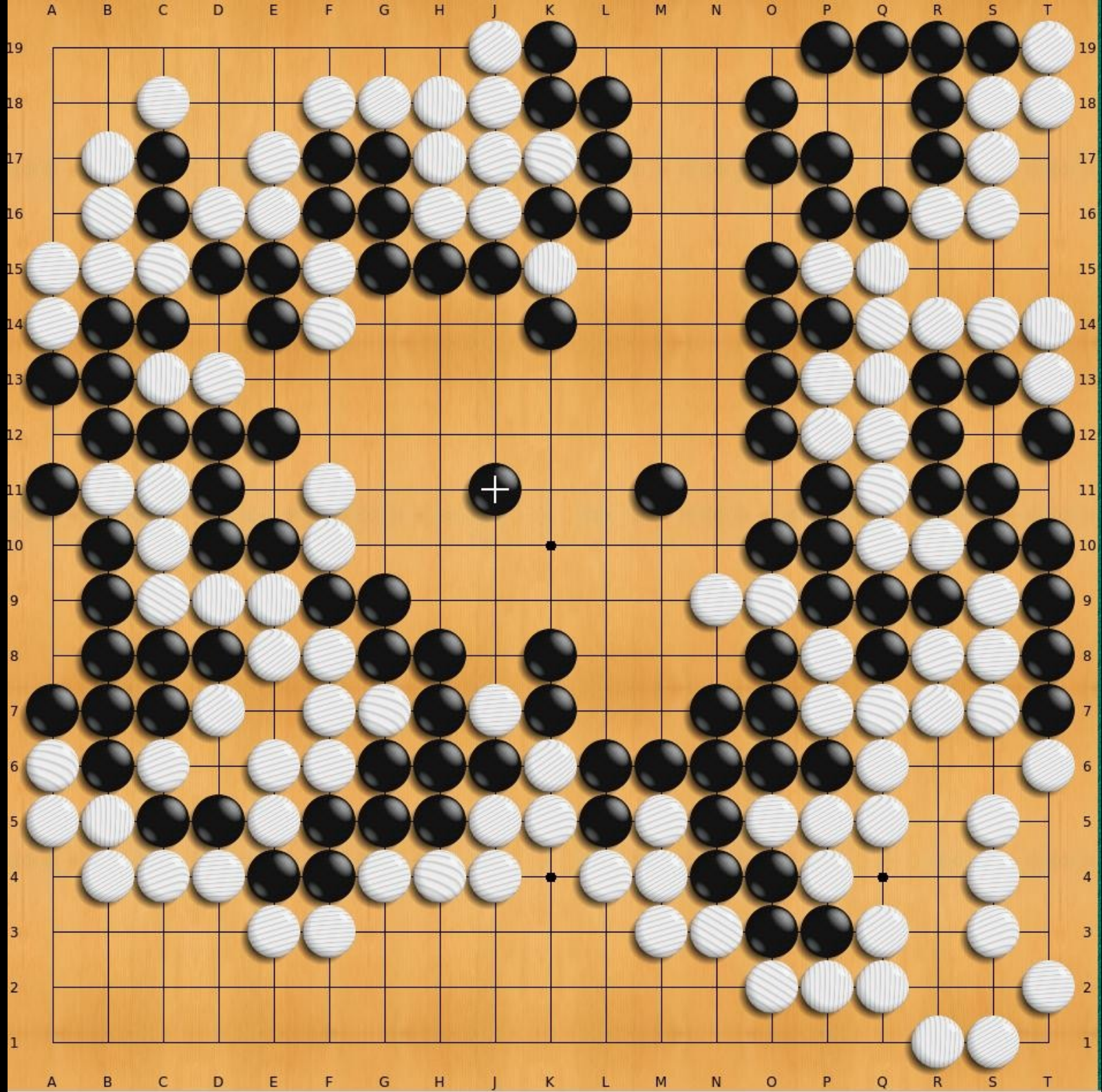


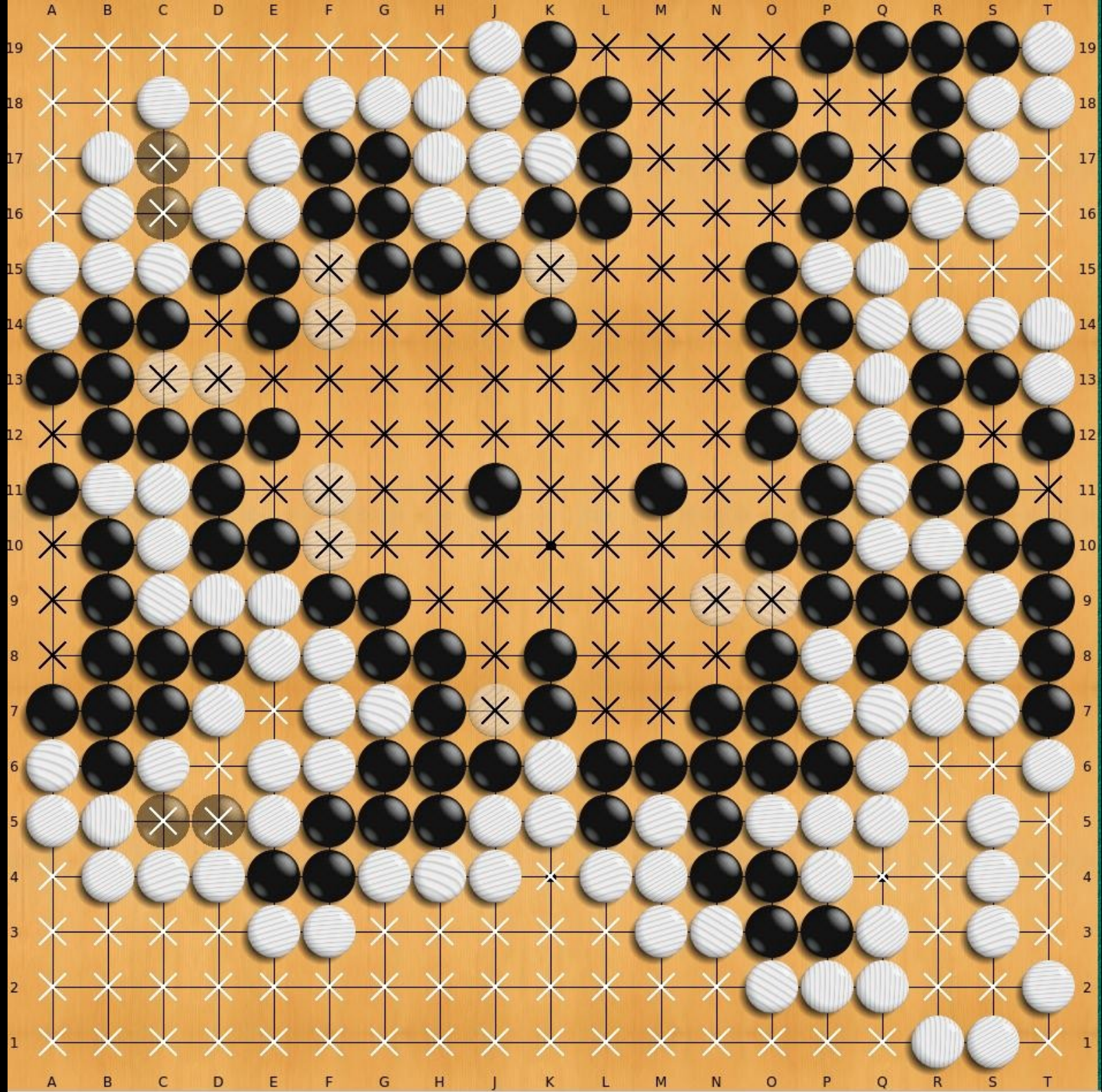














Computational Challenge

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**.“

Edward Lasker (chess grandmaster)

Larger board

19x19 vs. 8x8

Almost every move is legal

Average branching factor:

250 vs 35

State Space Complexity:

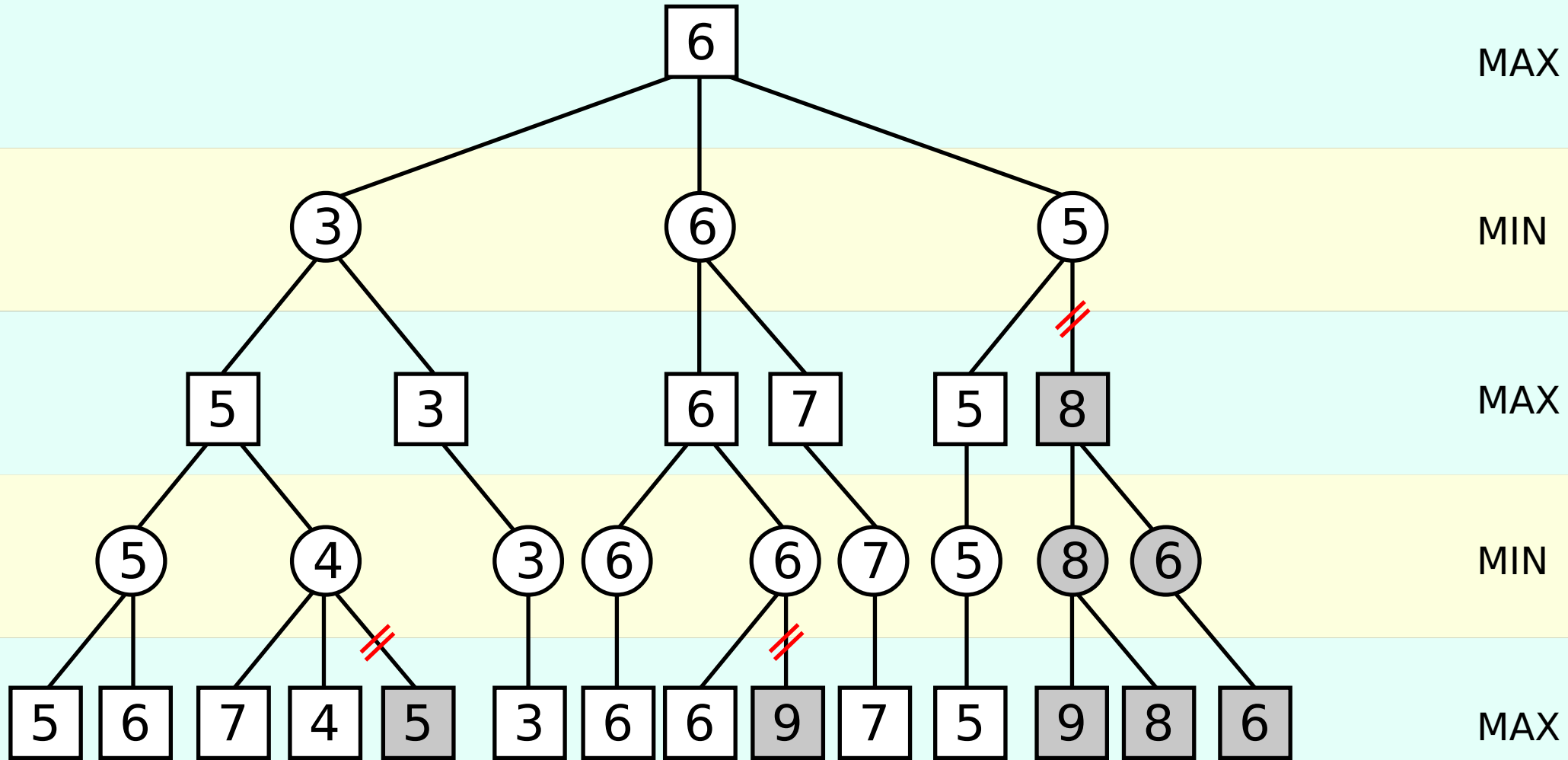
$10^{171}$  vs  $10^{47}$

$10^{80}$

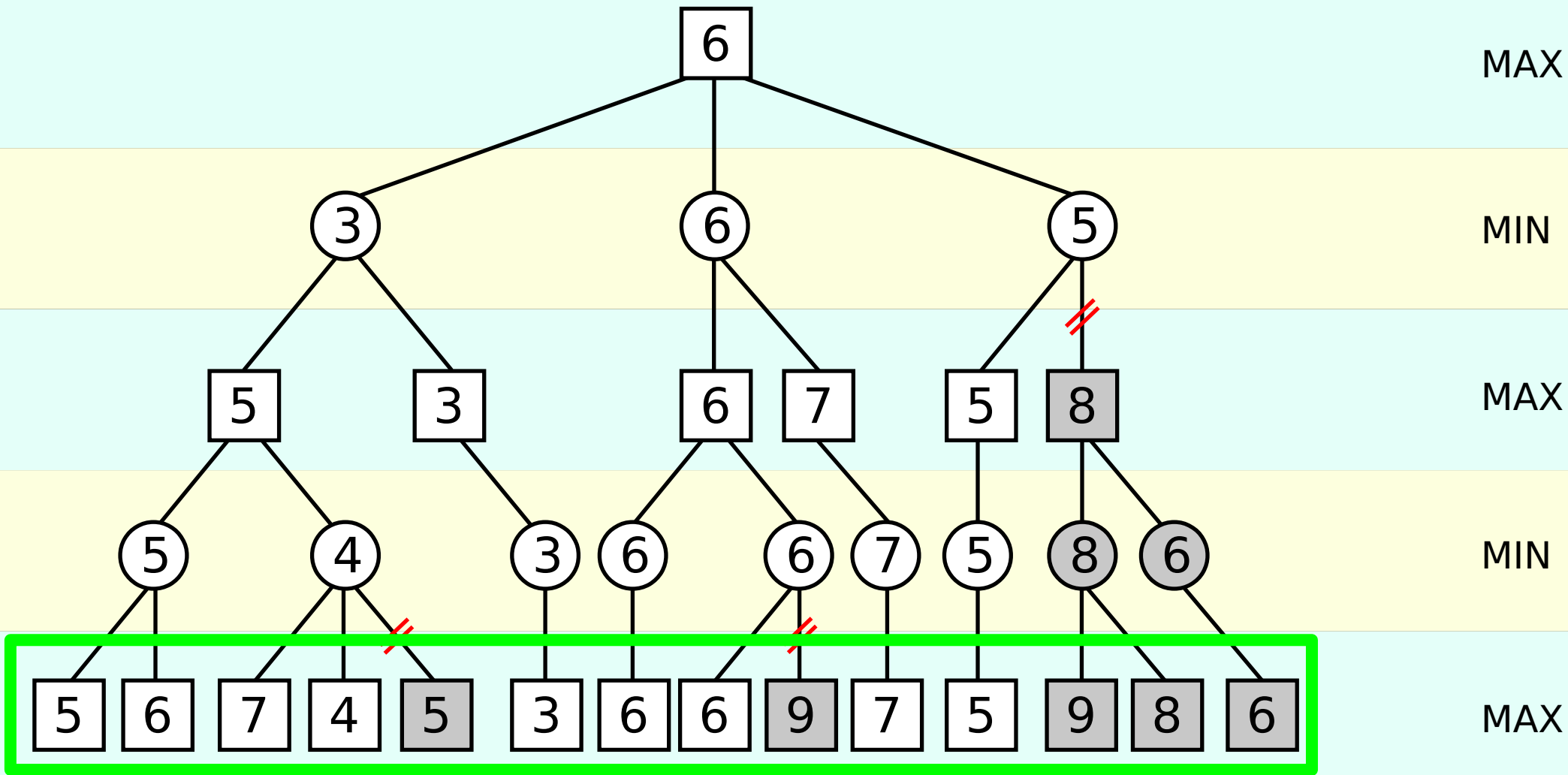


Global impact of moves

# Traditional Search



# Evaluation Function





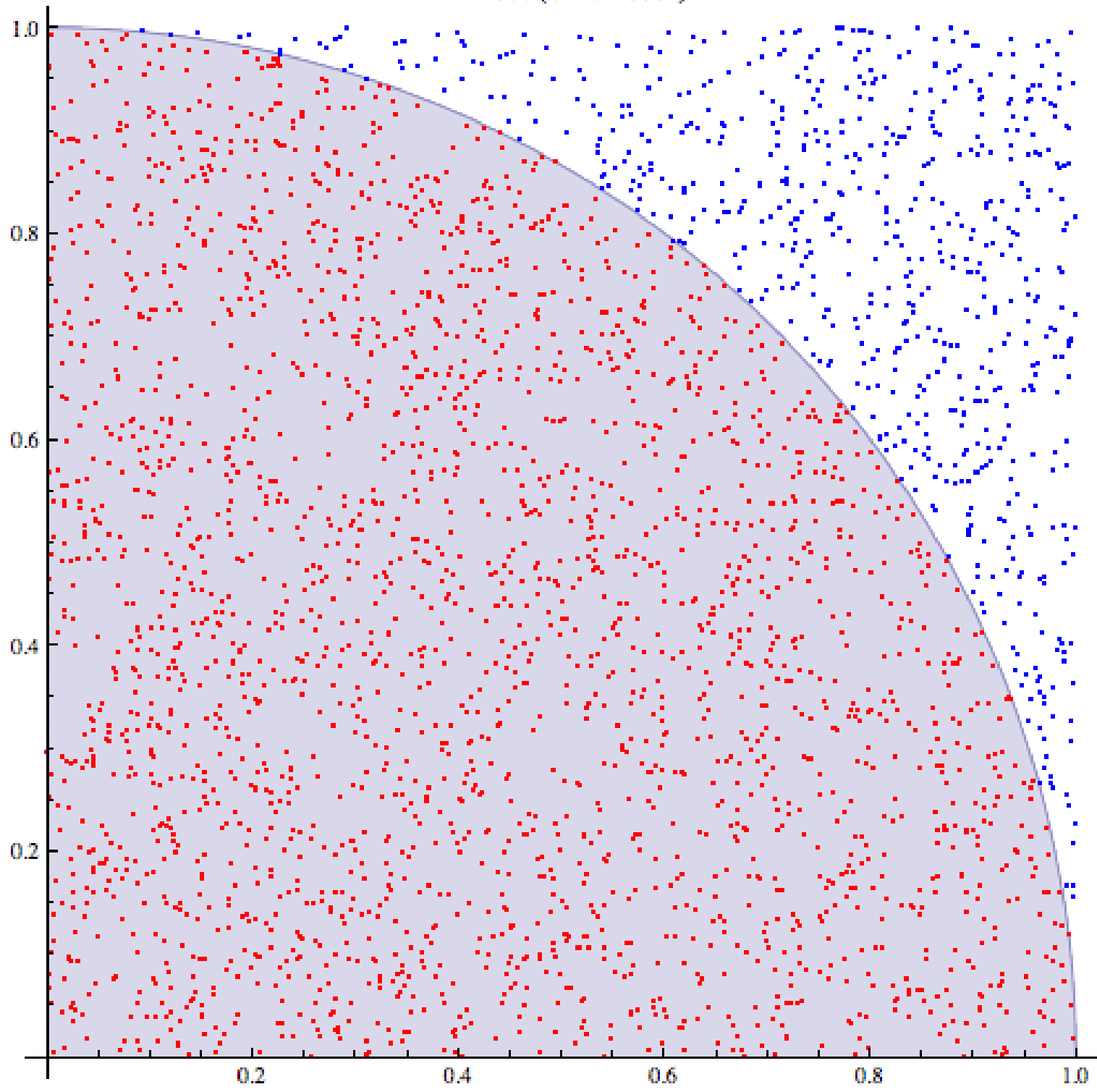
# Monte Carlo Method



What is  $\pi$ ?

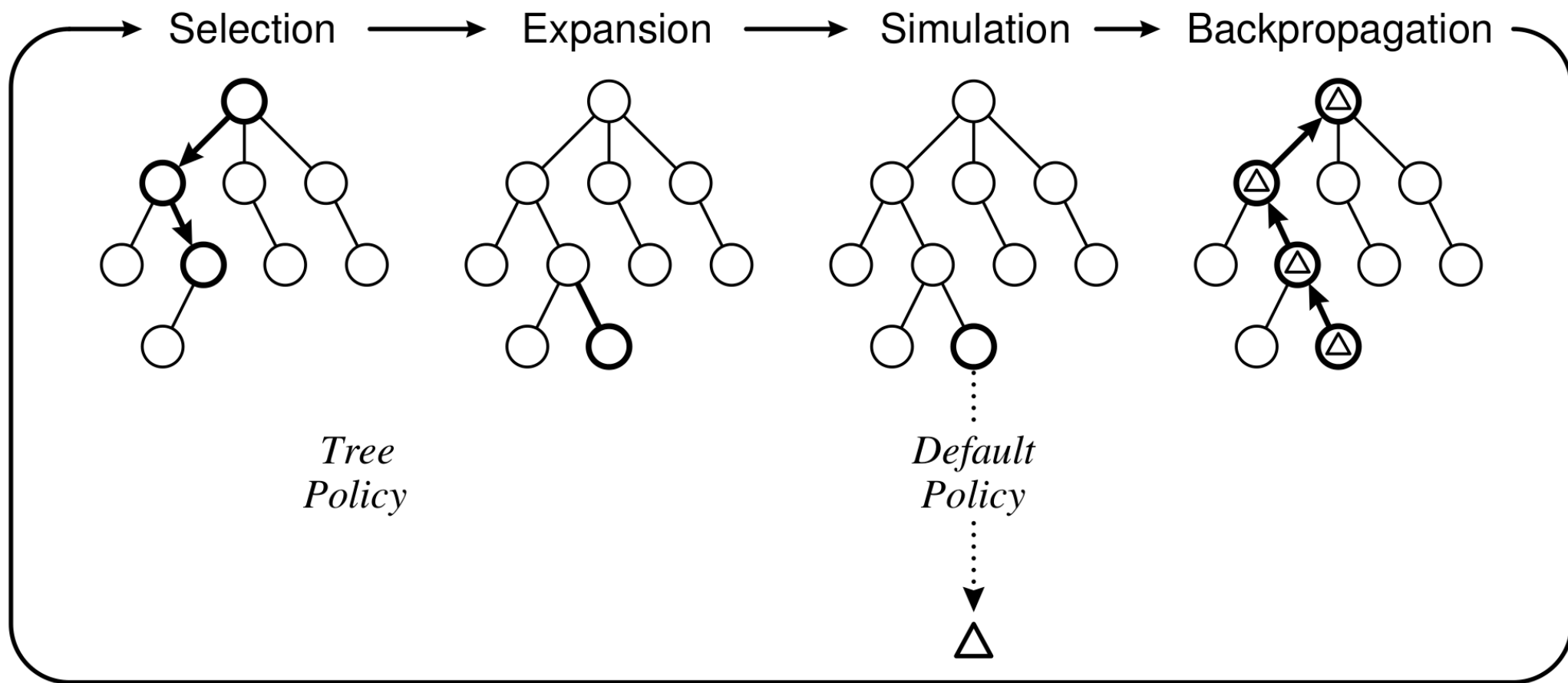
How do you determine  $\pi$ ?

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

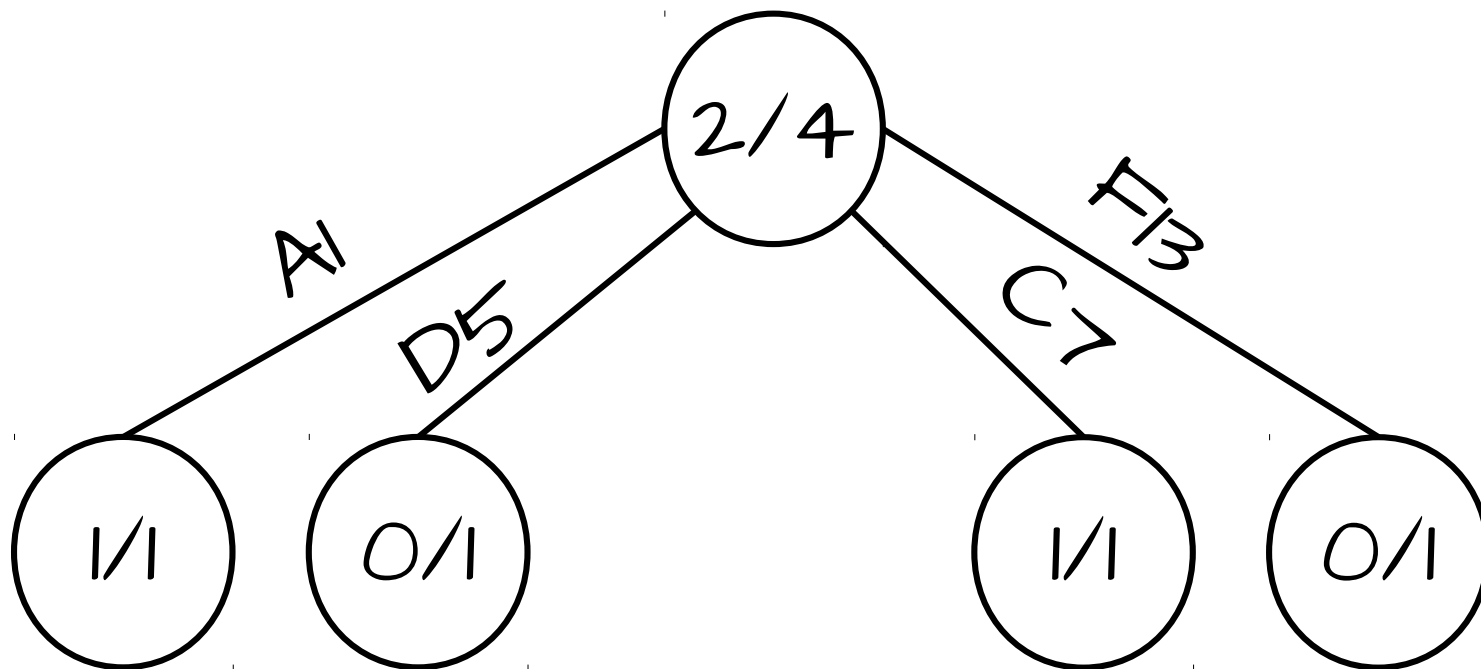


2006

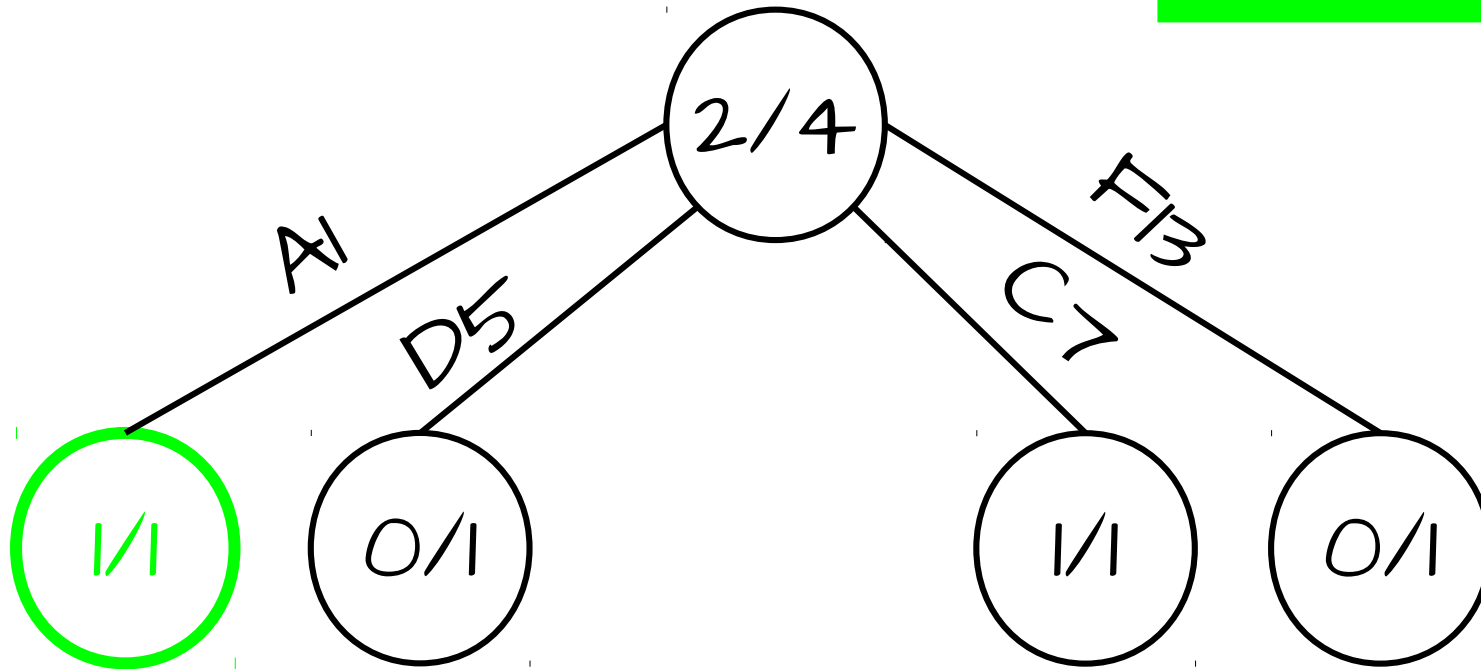




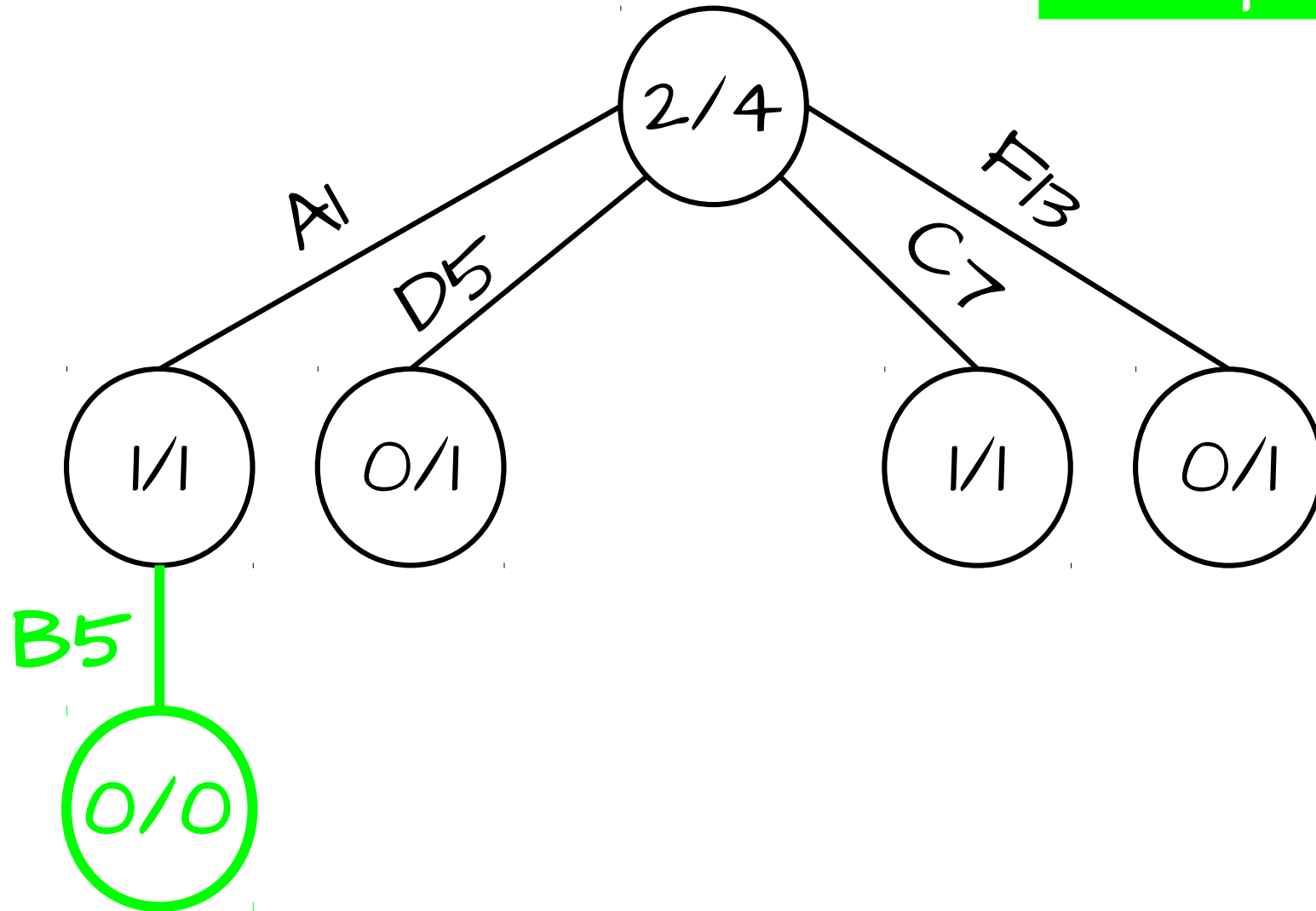
Browne, Cb, and Edward Powley. 2012. A survey of monte carlo tree search methods. *Intelligence and AI* 4, no. 1: 1-49



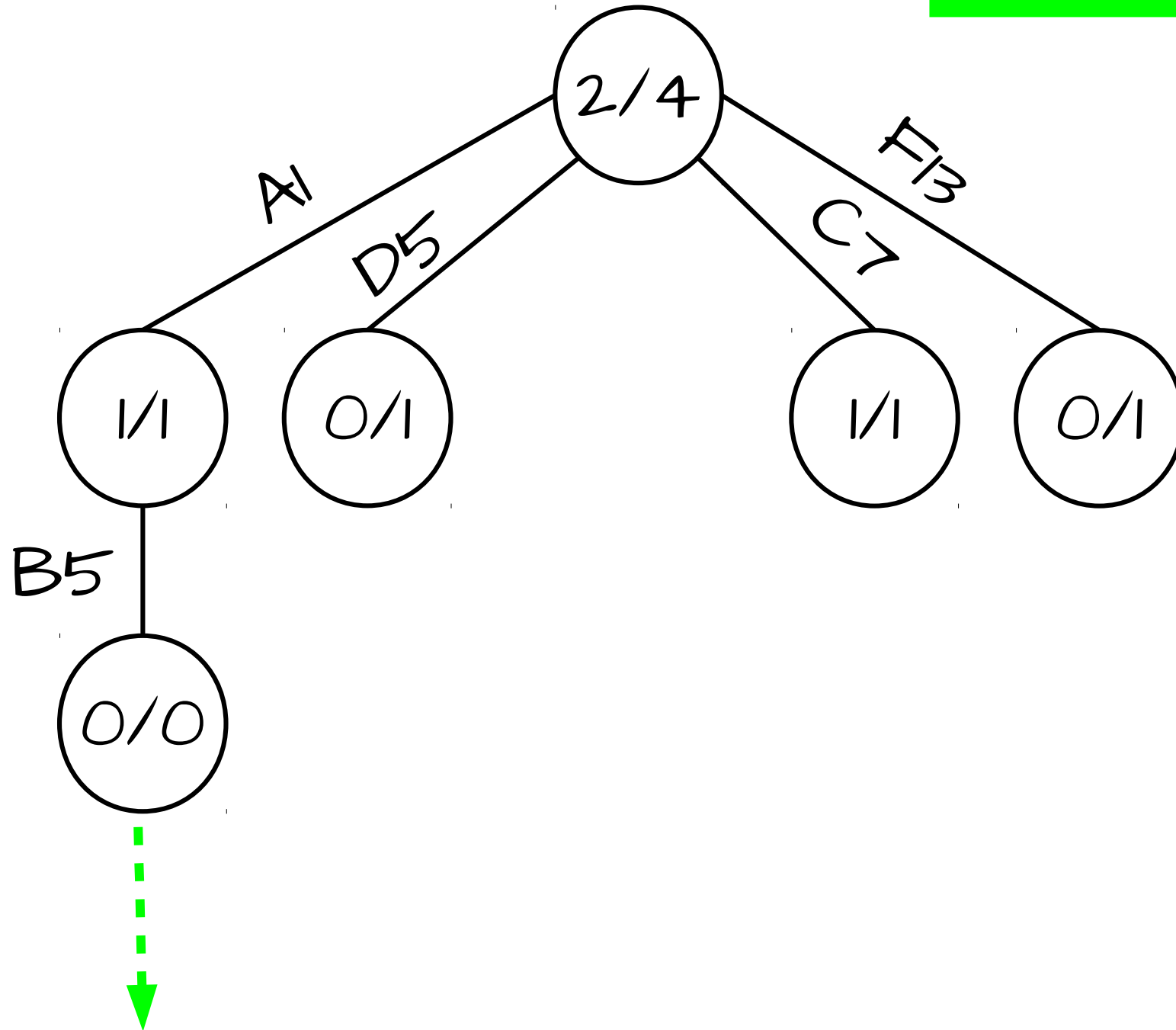
# Selection



# Expansion



# Simulation



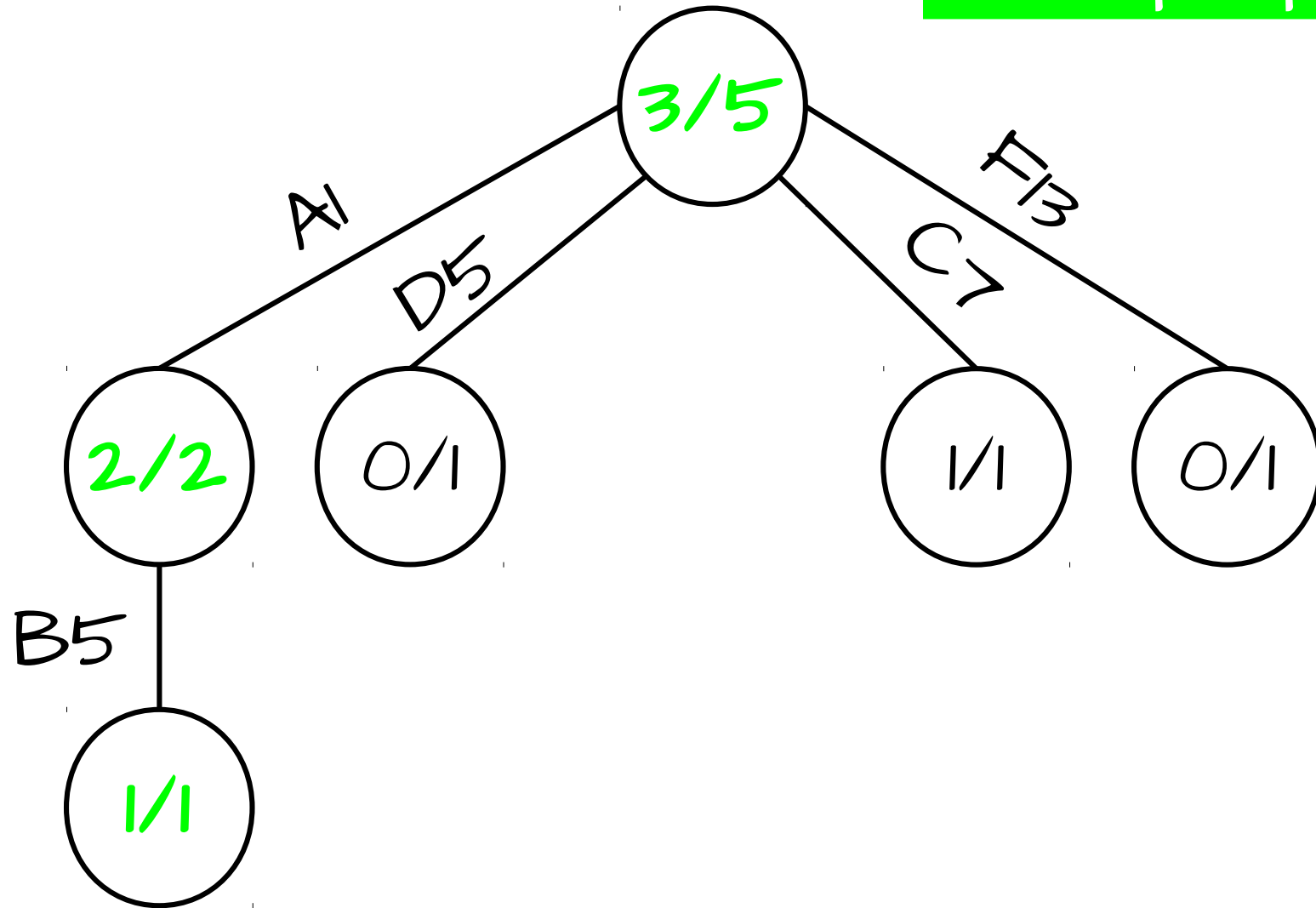
Random



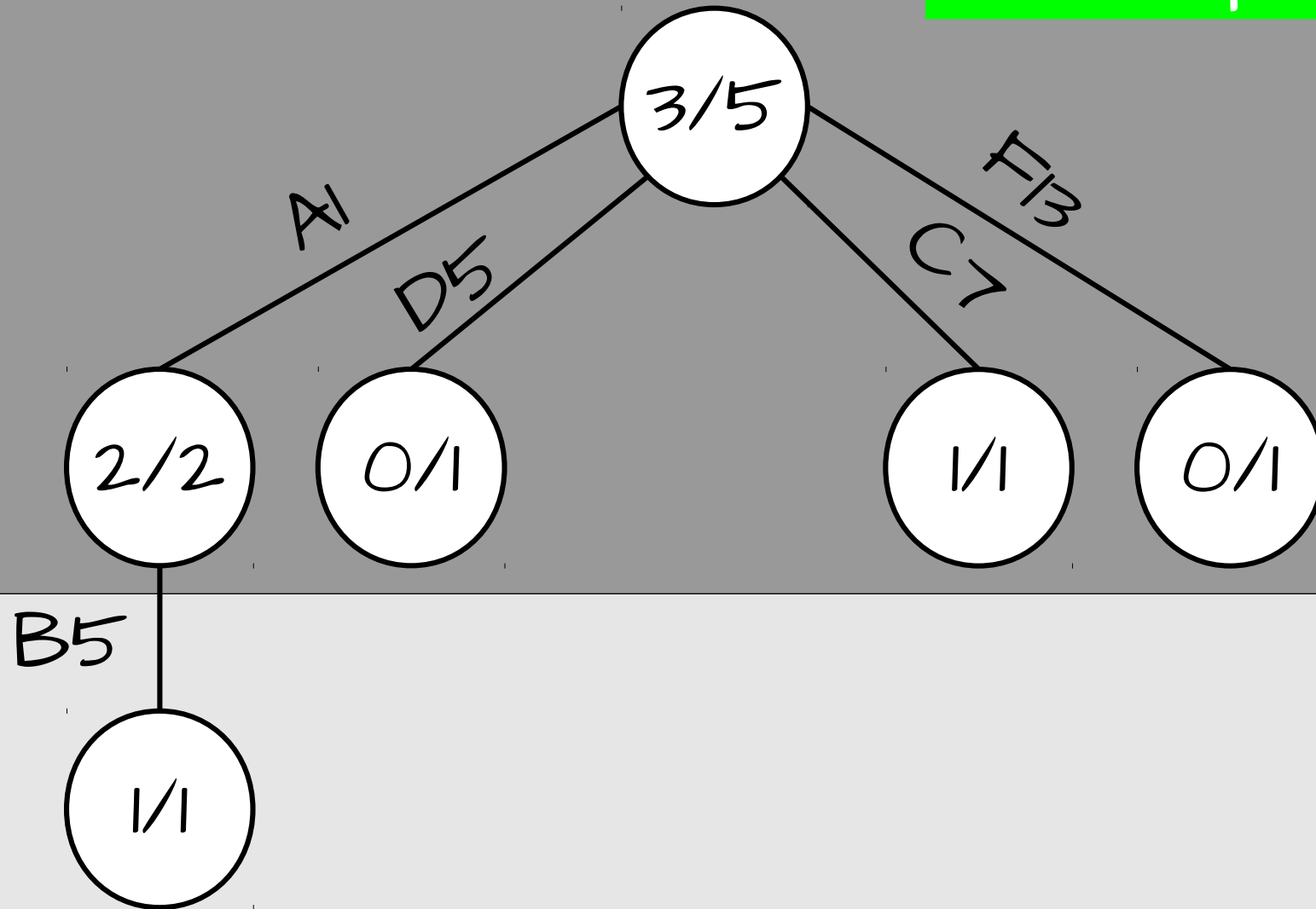
Not Human like?



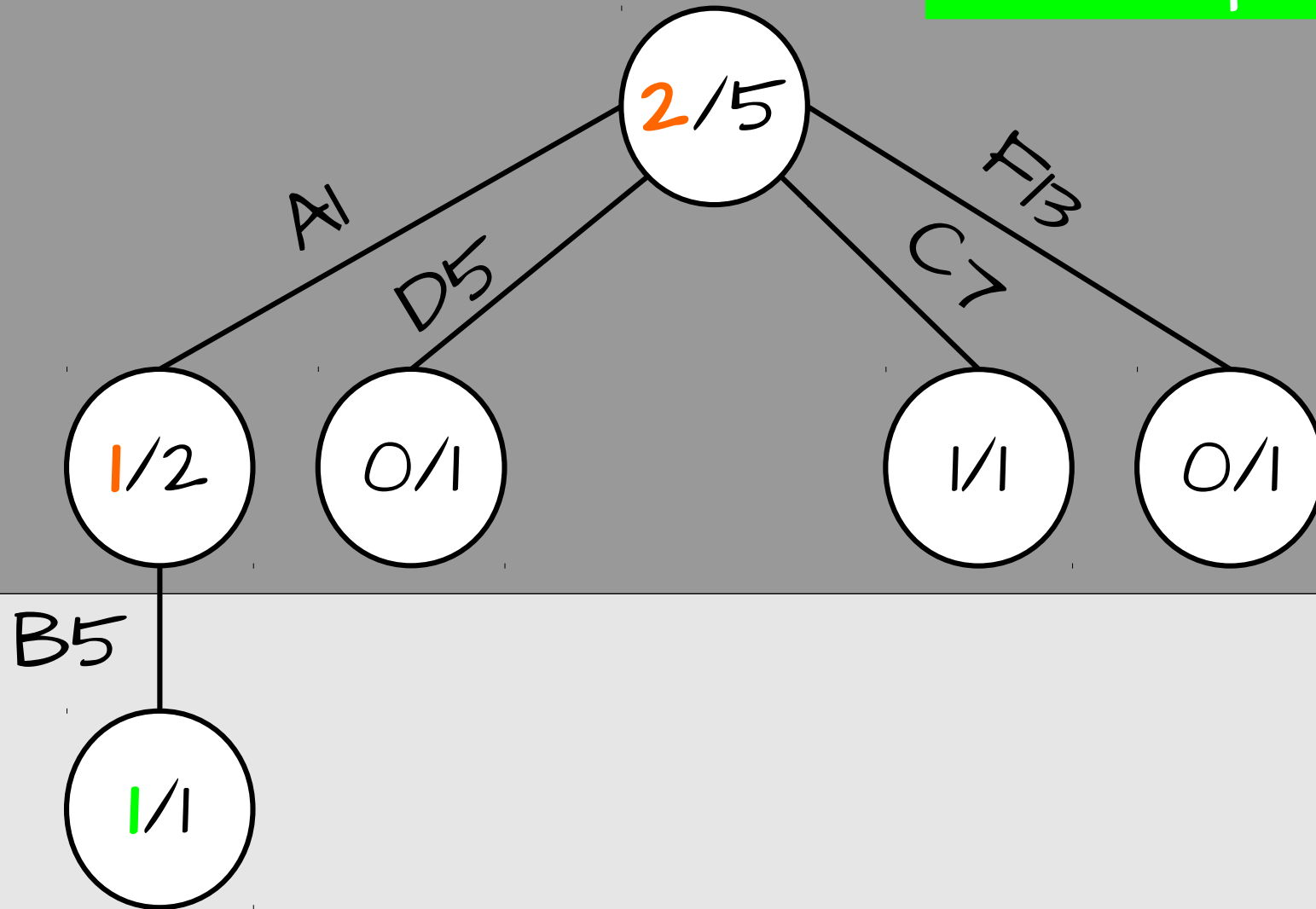
# Backpropagation



# Perspective



# Perspective





# Multi Armed Bandit

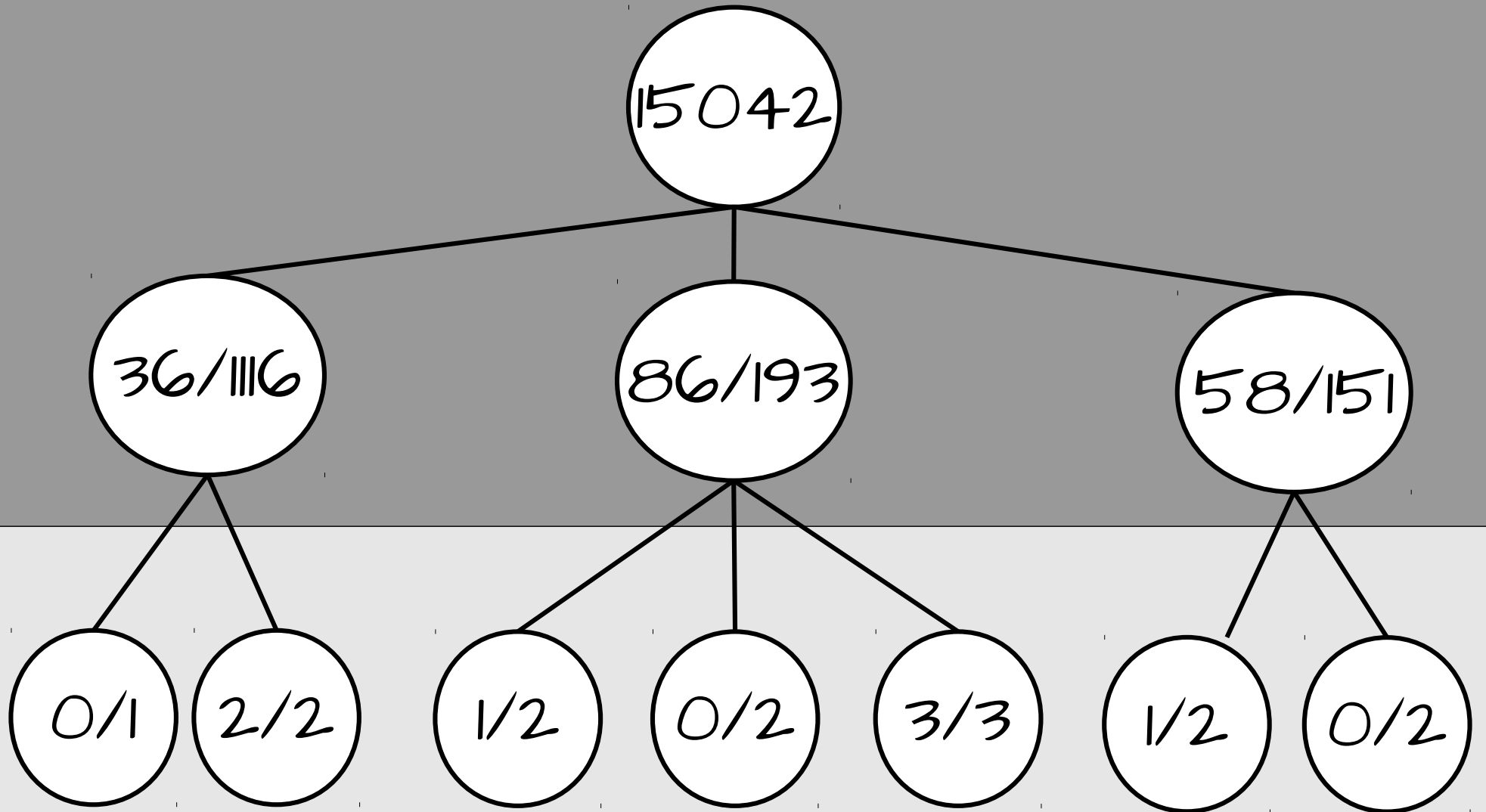


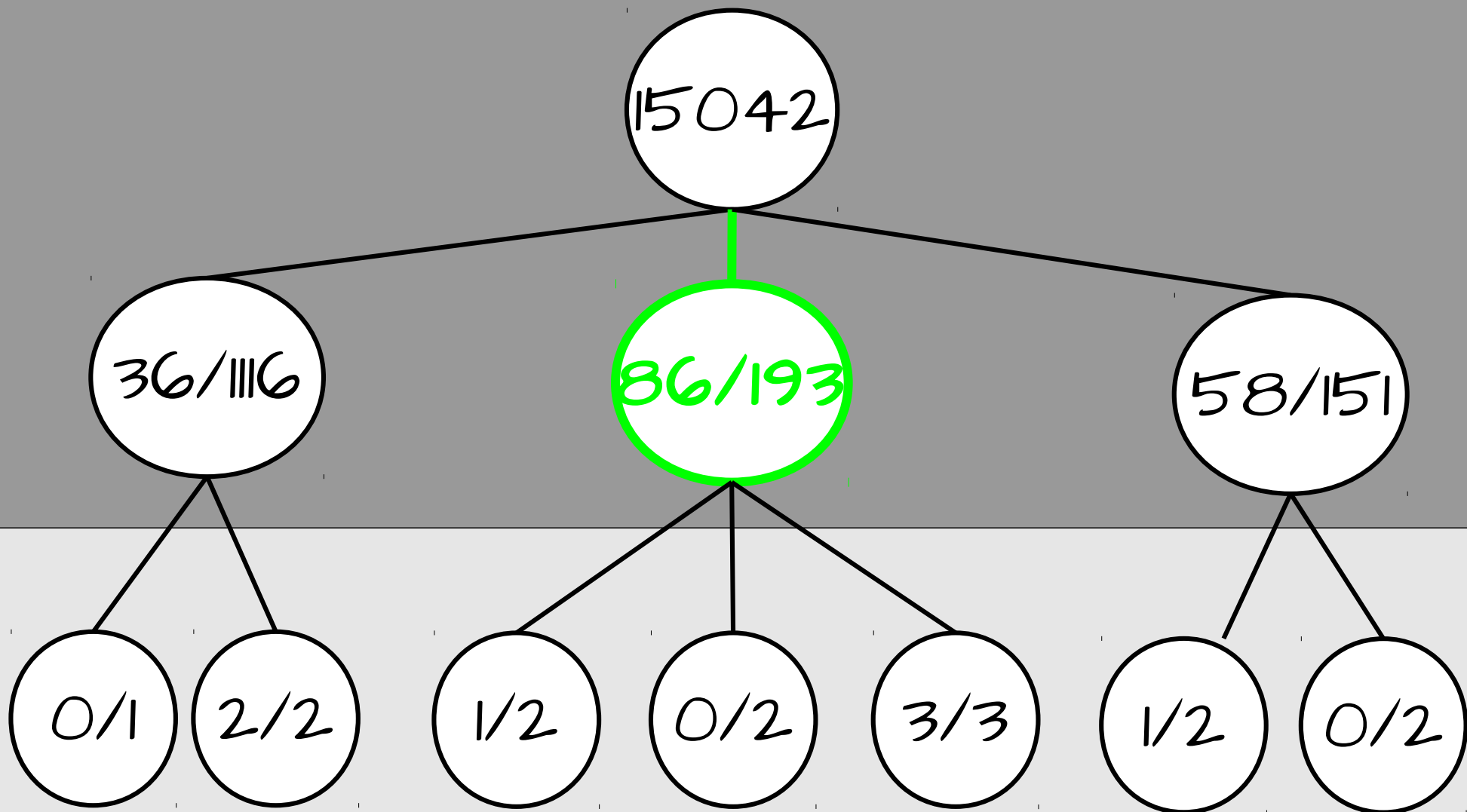
# Multi Armed Bandit

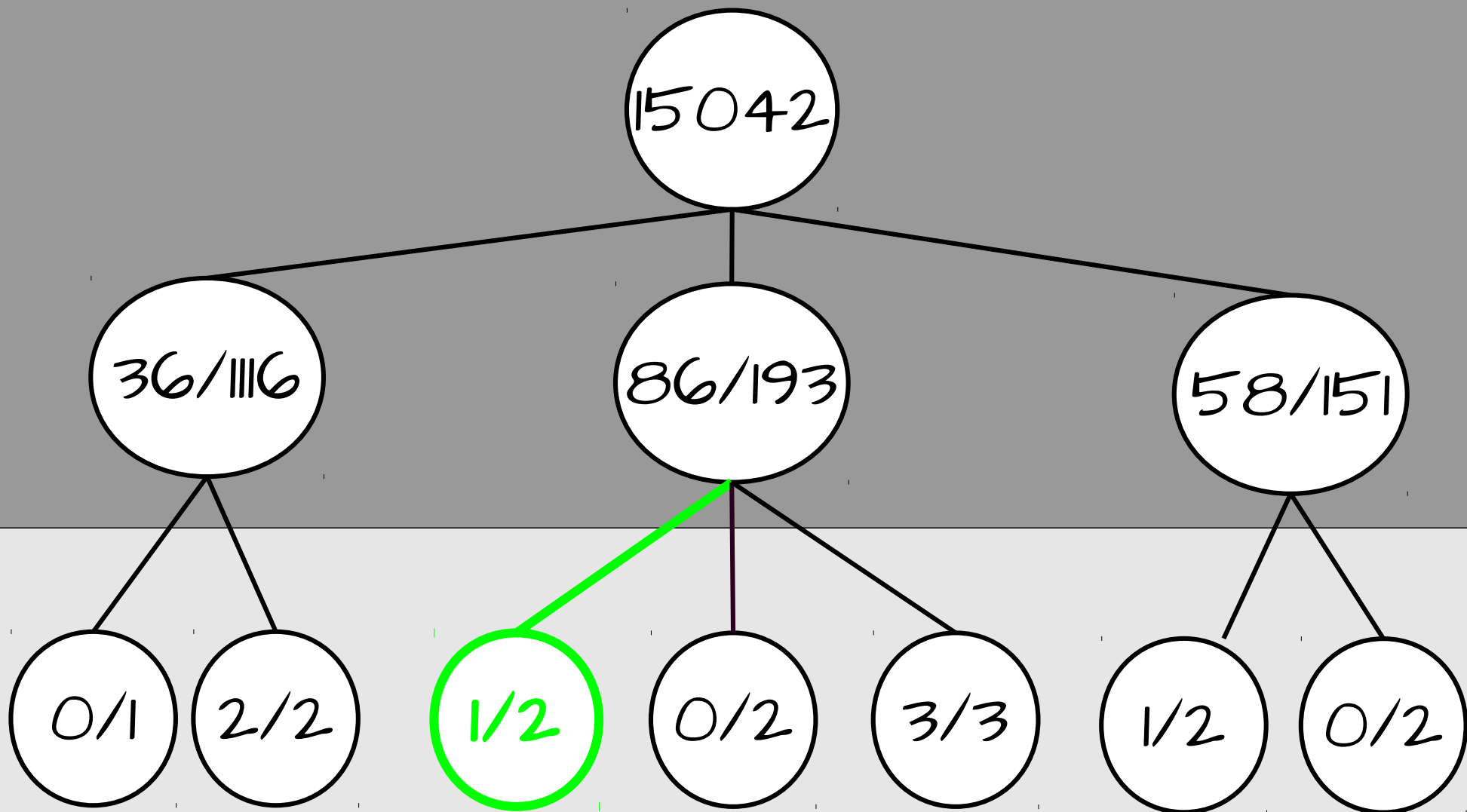
Exploitation vs Exploration



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







Generate a valid random move

Who has won?

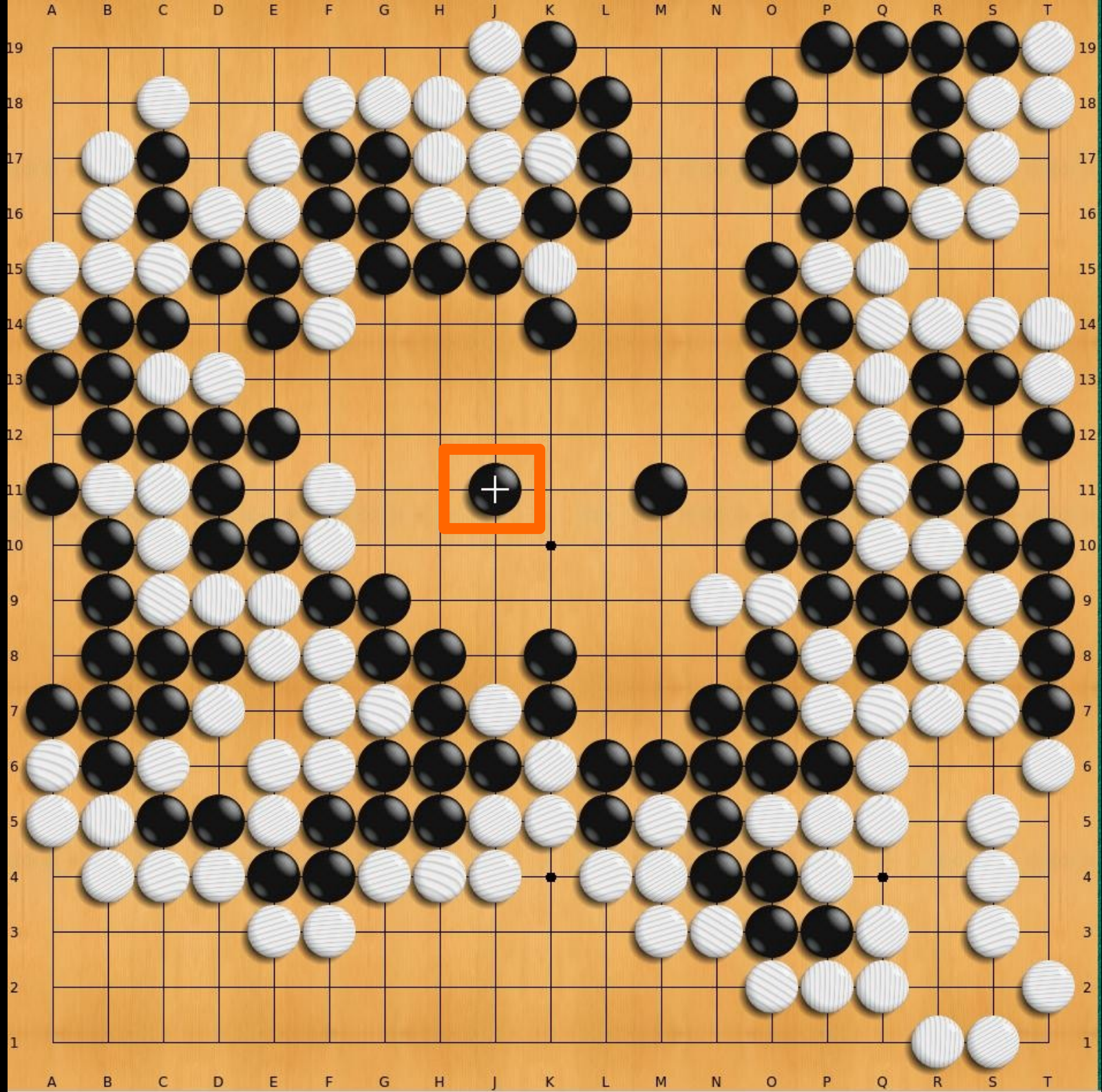


# General Game Playing

Anytime

Lazy

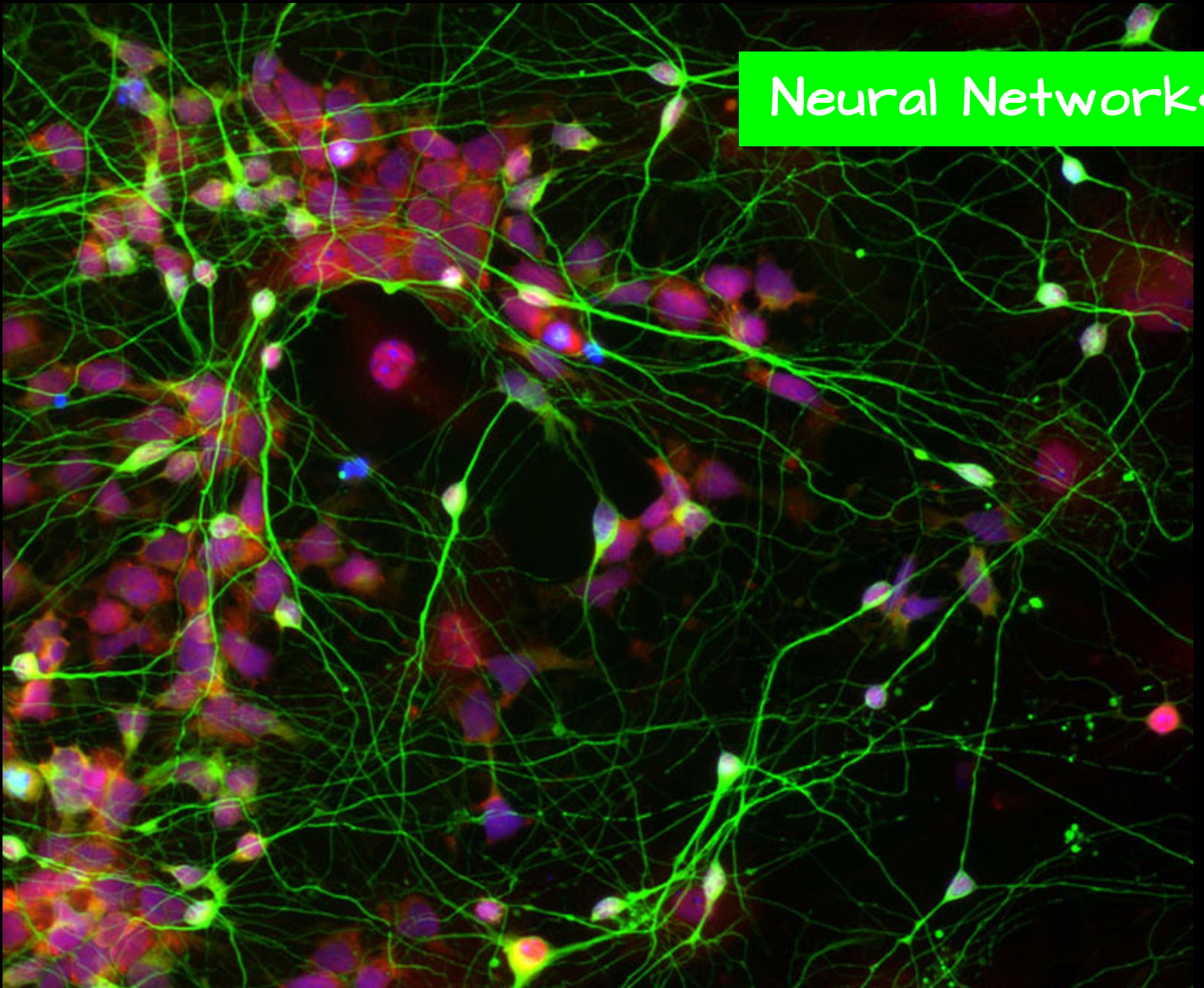




Expert Knowledge



# Neural Networks



# MOVE EVALUATION IN GO USING DEEP CONVOLUTIONAL NEURAL NETWORKS

**Chris J. Maddison**

University of Toronto

cmaddis@cs.toronto.edu

2014

**Aja Huang<sup>1</sup>, Ilya Sutskever<sup>2</sup>, David Silver<sup>1</sup>**

Google DeepMind<sup>1</sup>, Google Brain<sup>2</sup>

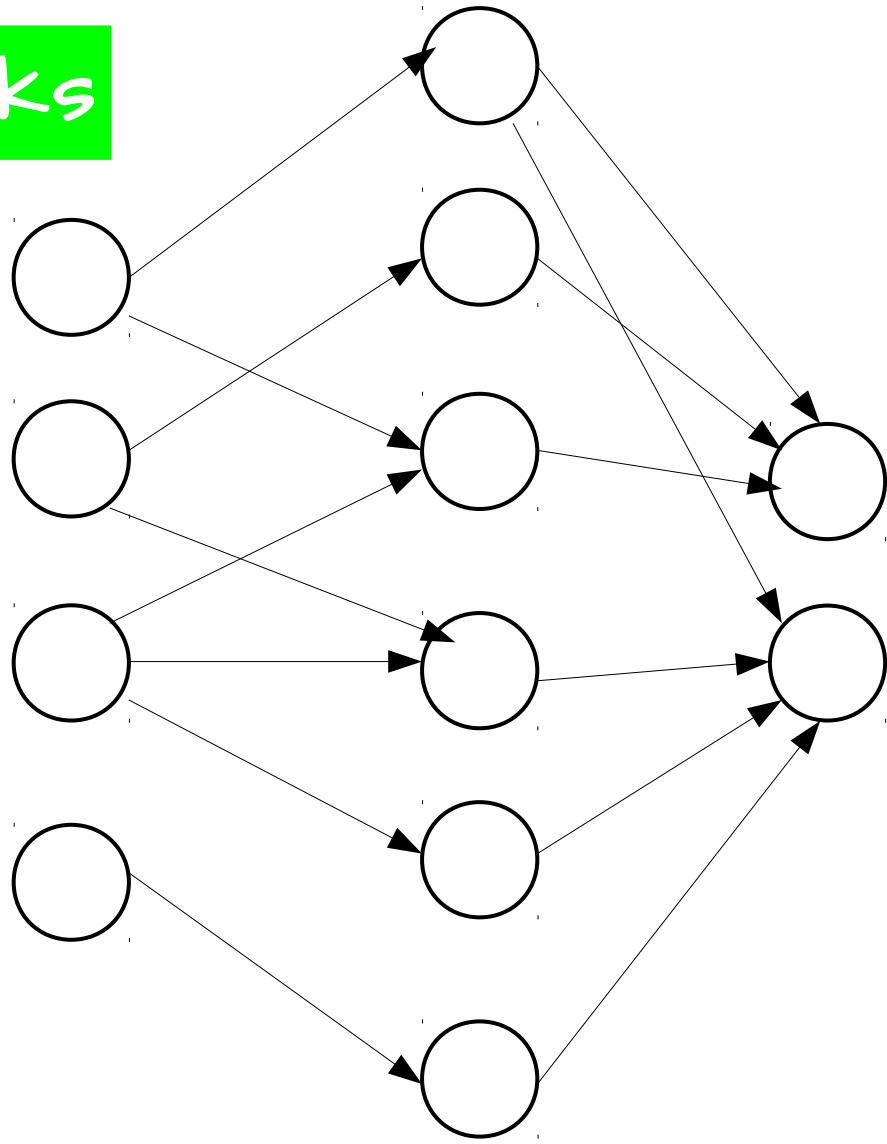
{ajahuang, ilyasu, davidsilver}@google.com

## 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?

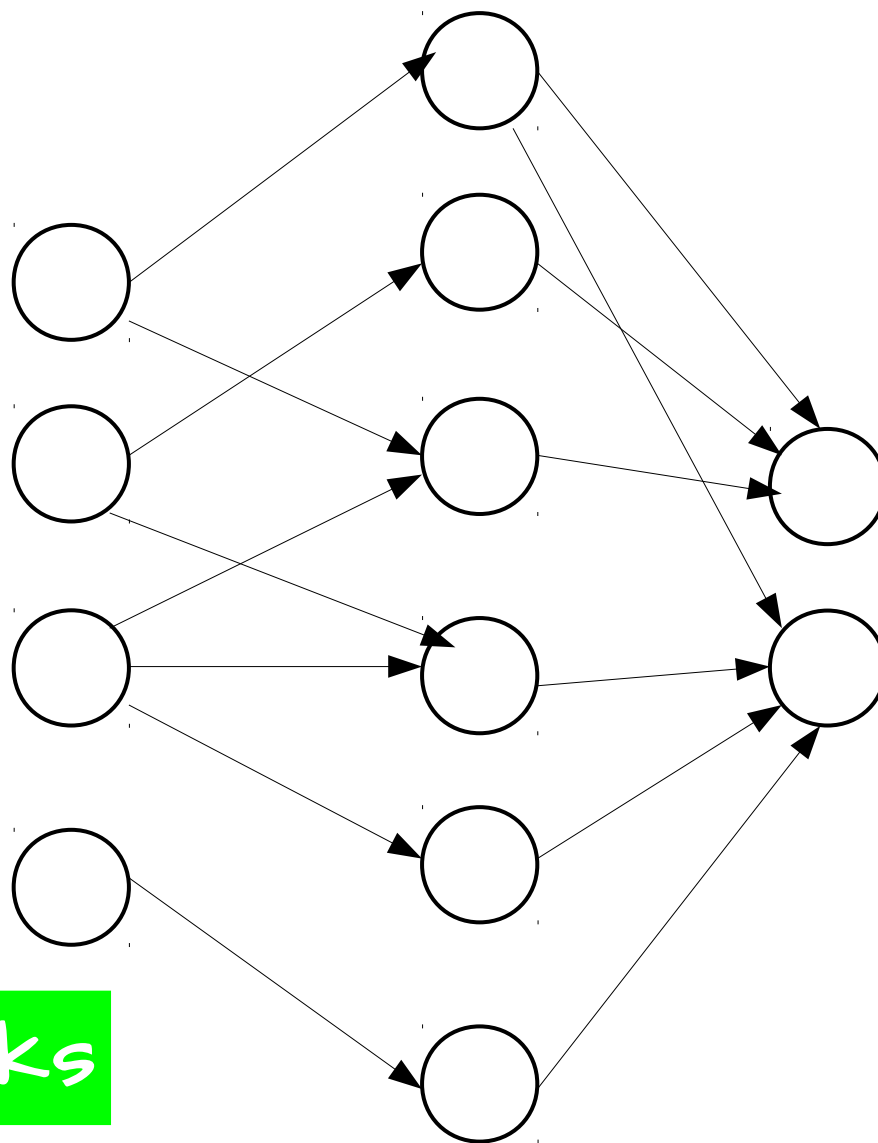
# Neural Networks



Input

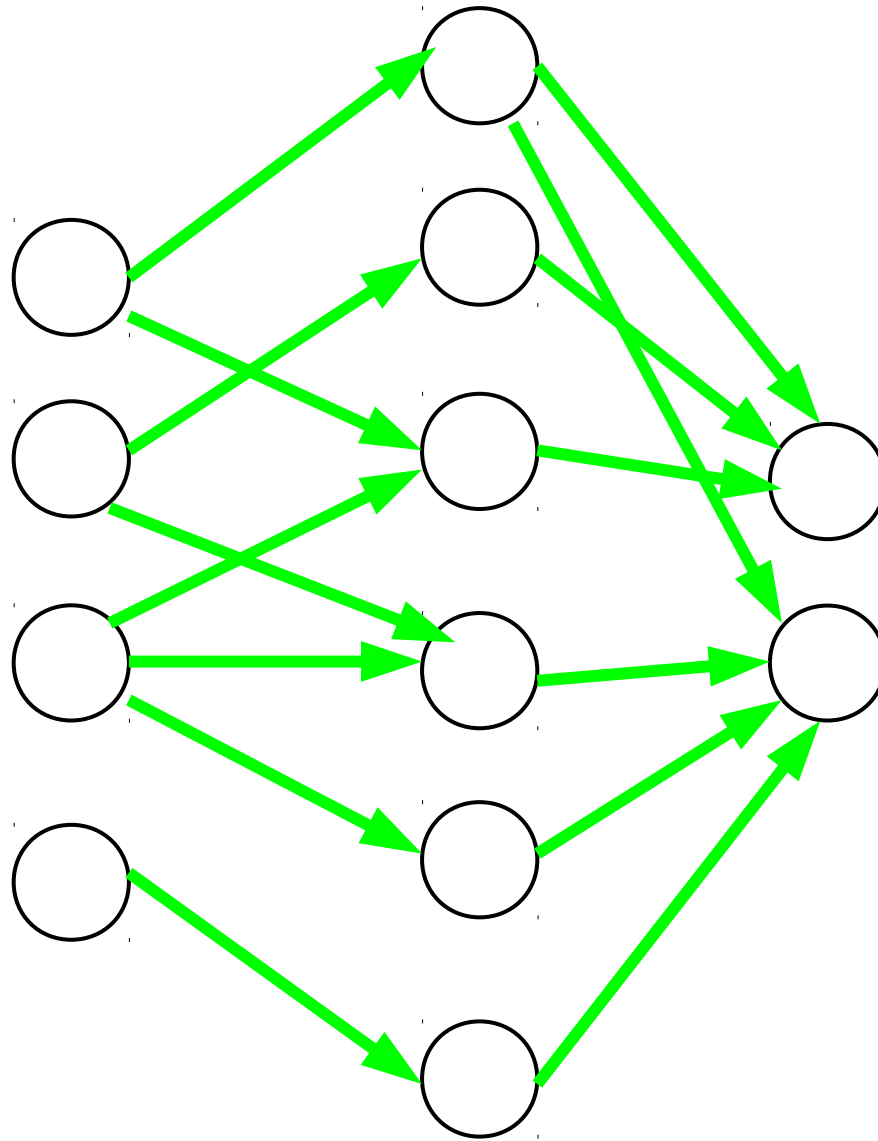
"Hidden" Layer

Output

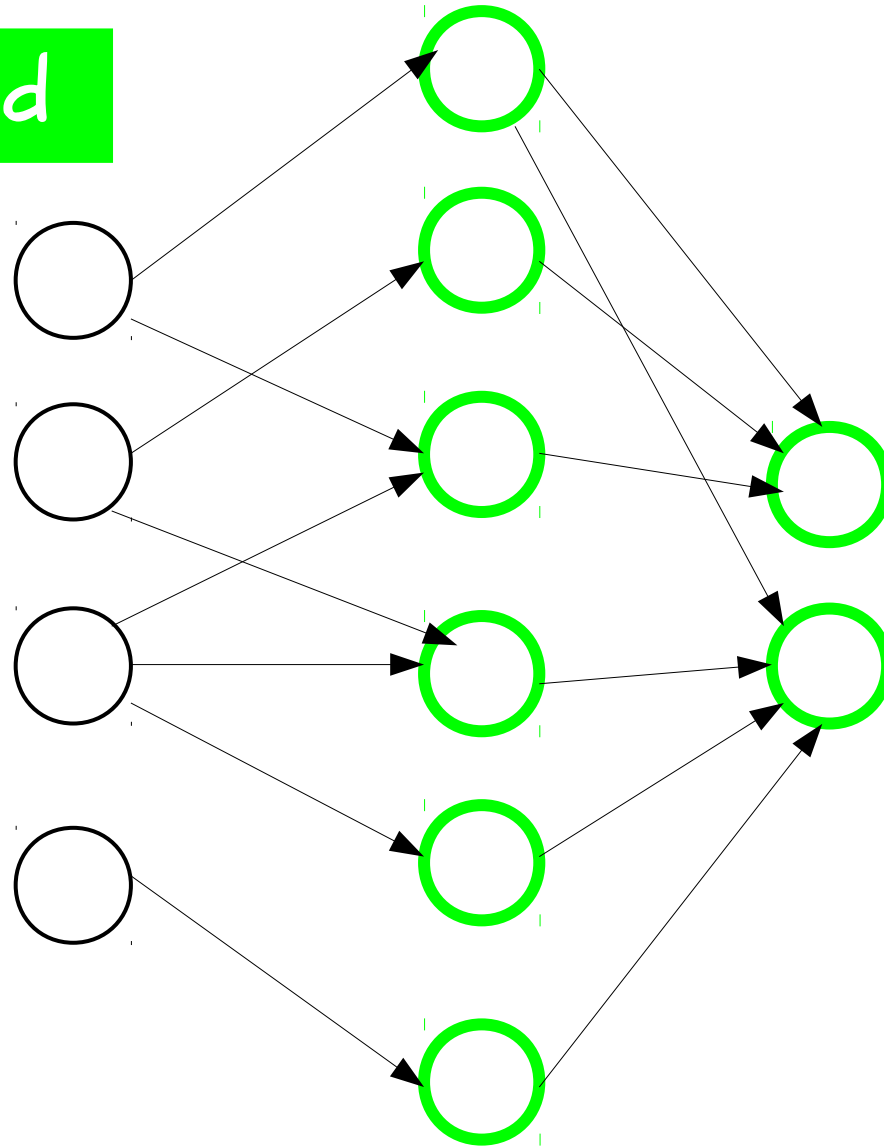


Neural Networks

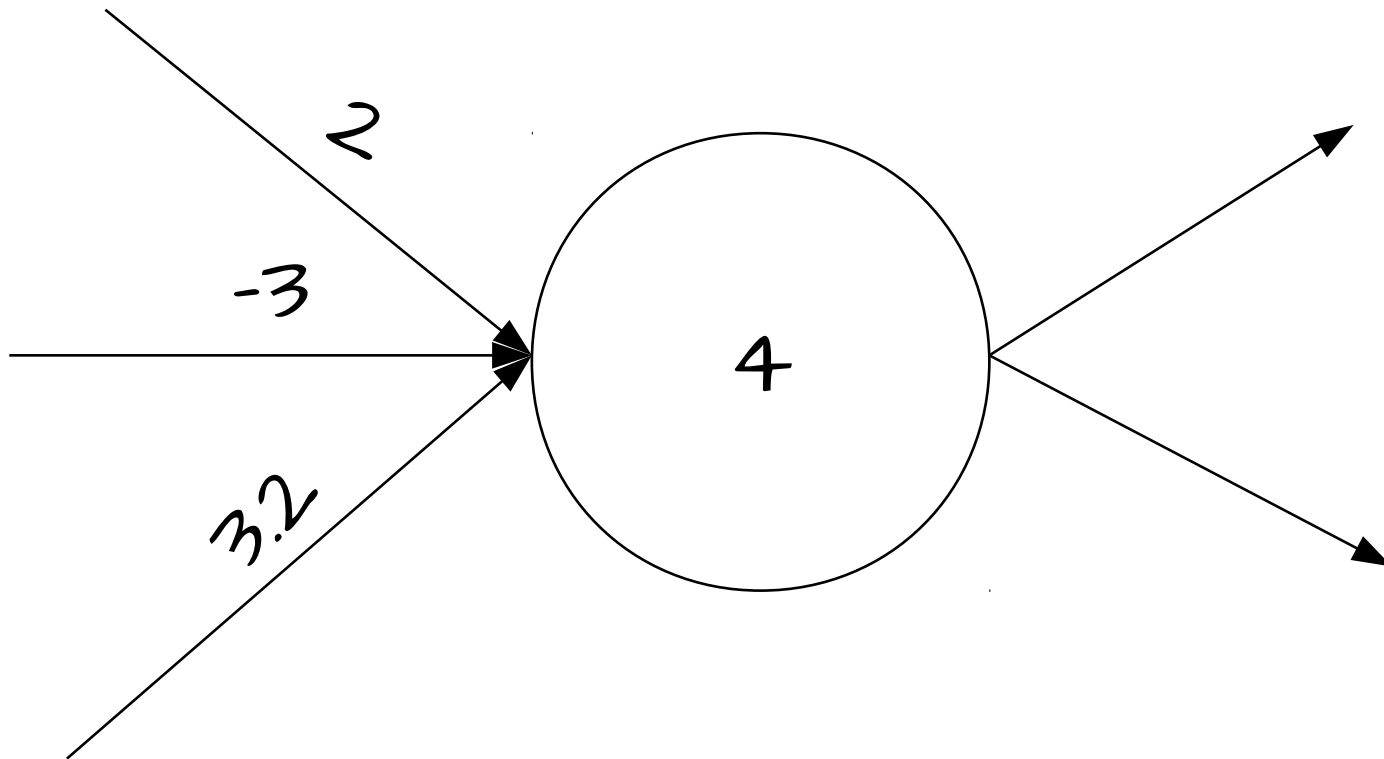
# Weights



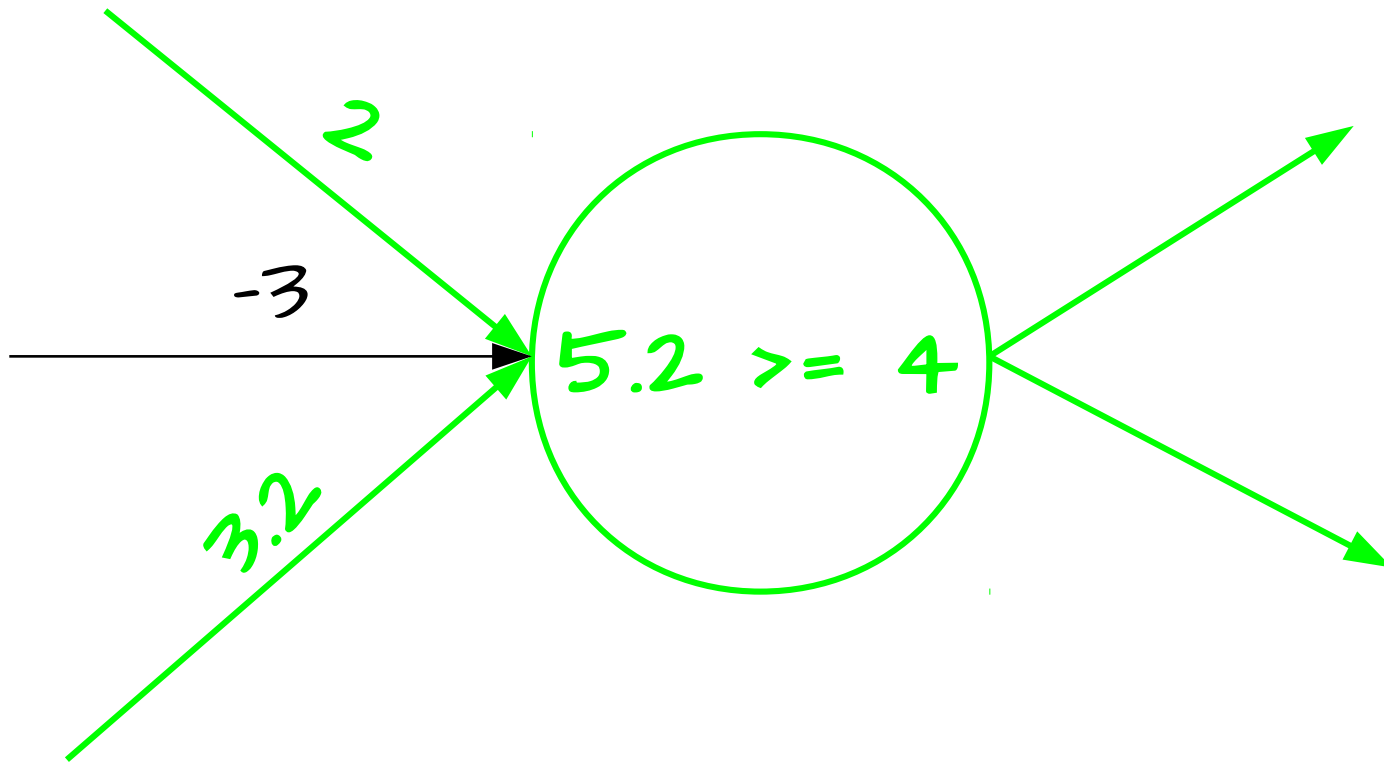
# Bias/Threshold



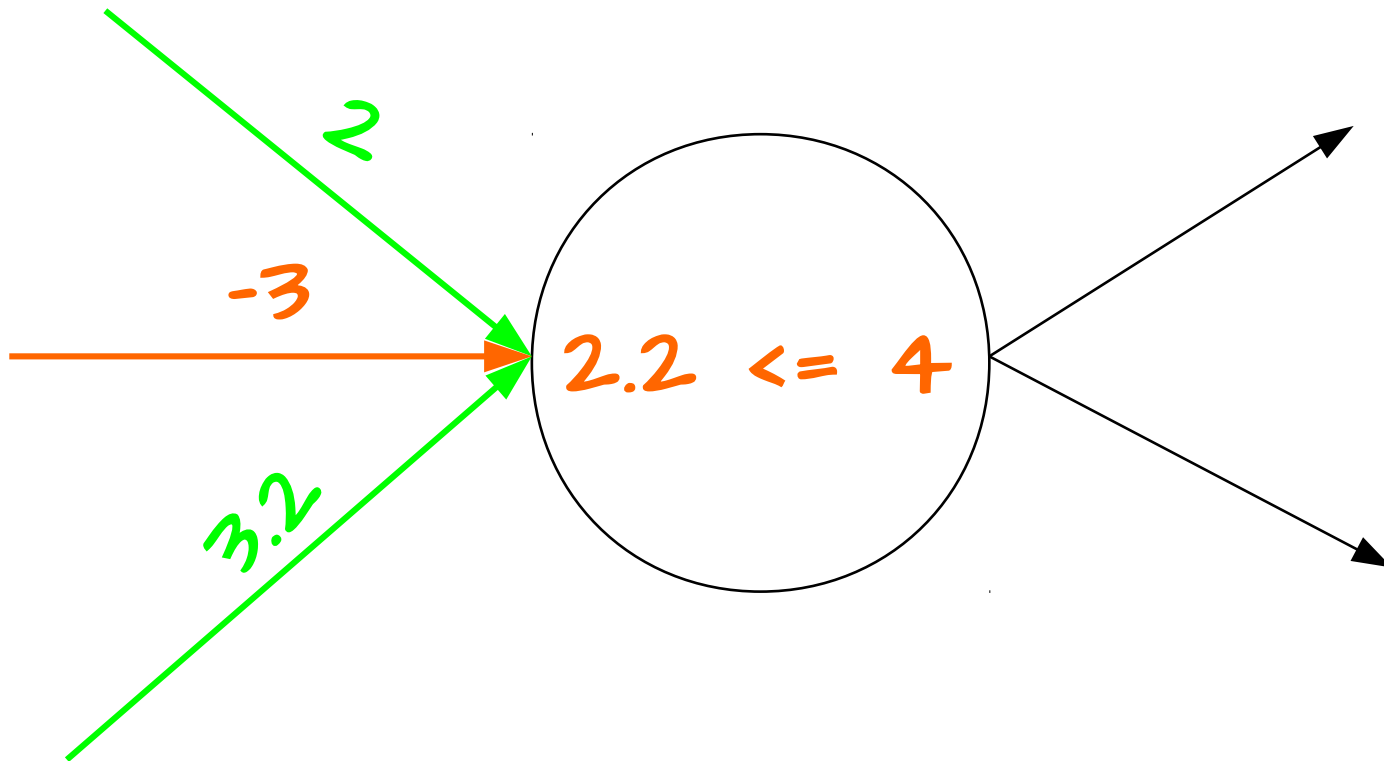
# Activation



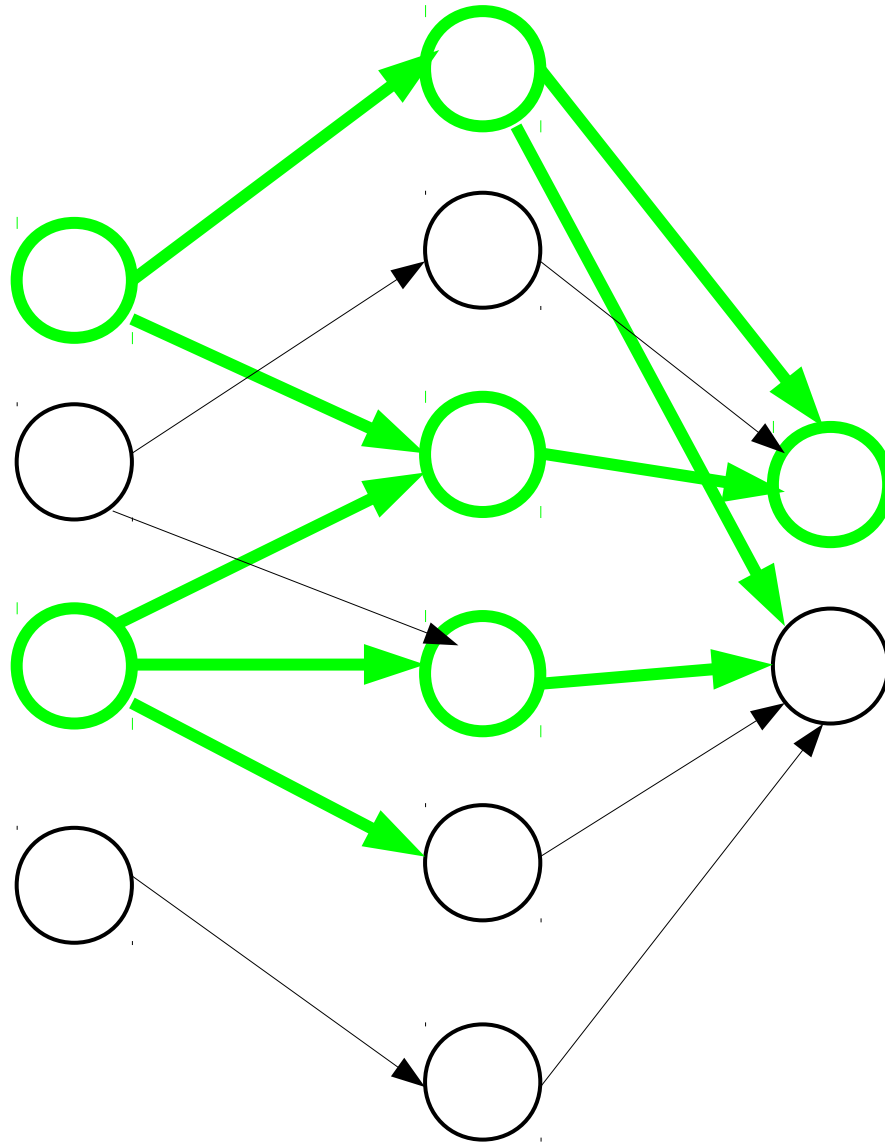
# Activation



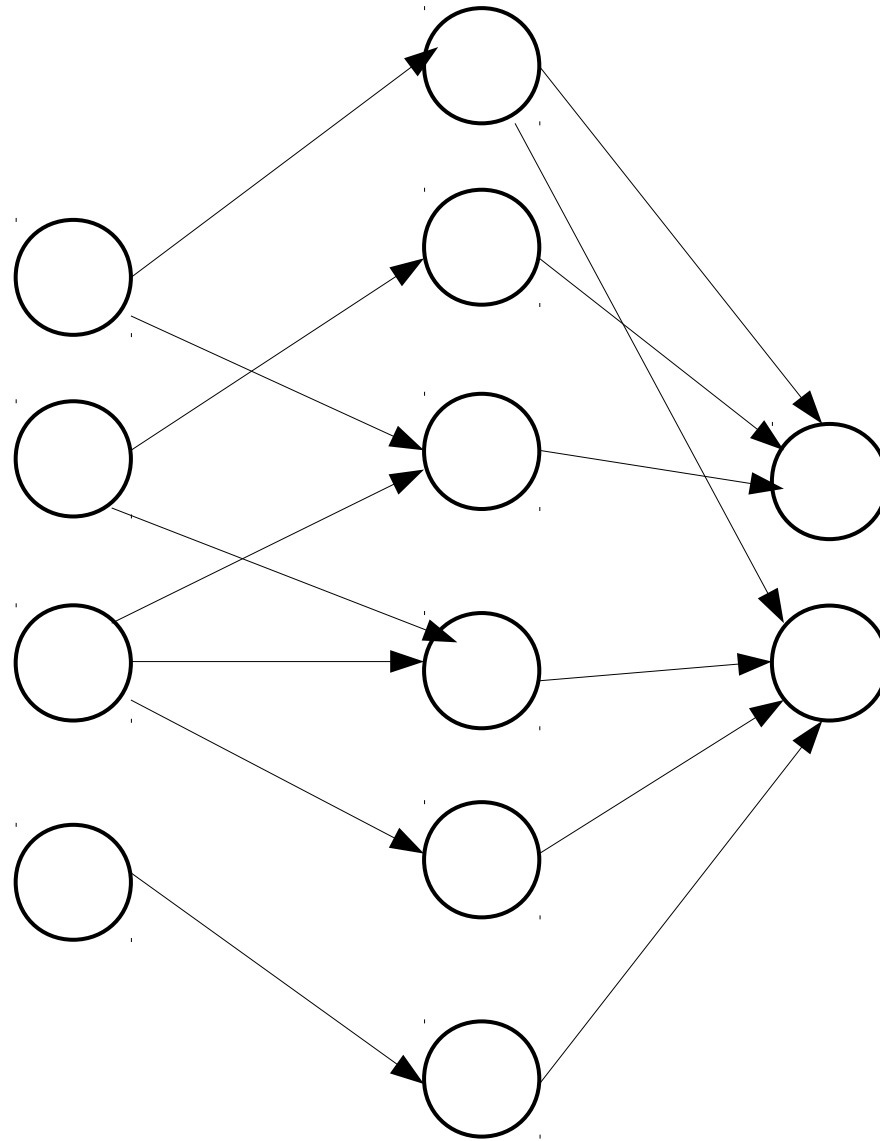
# Activation



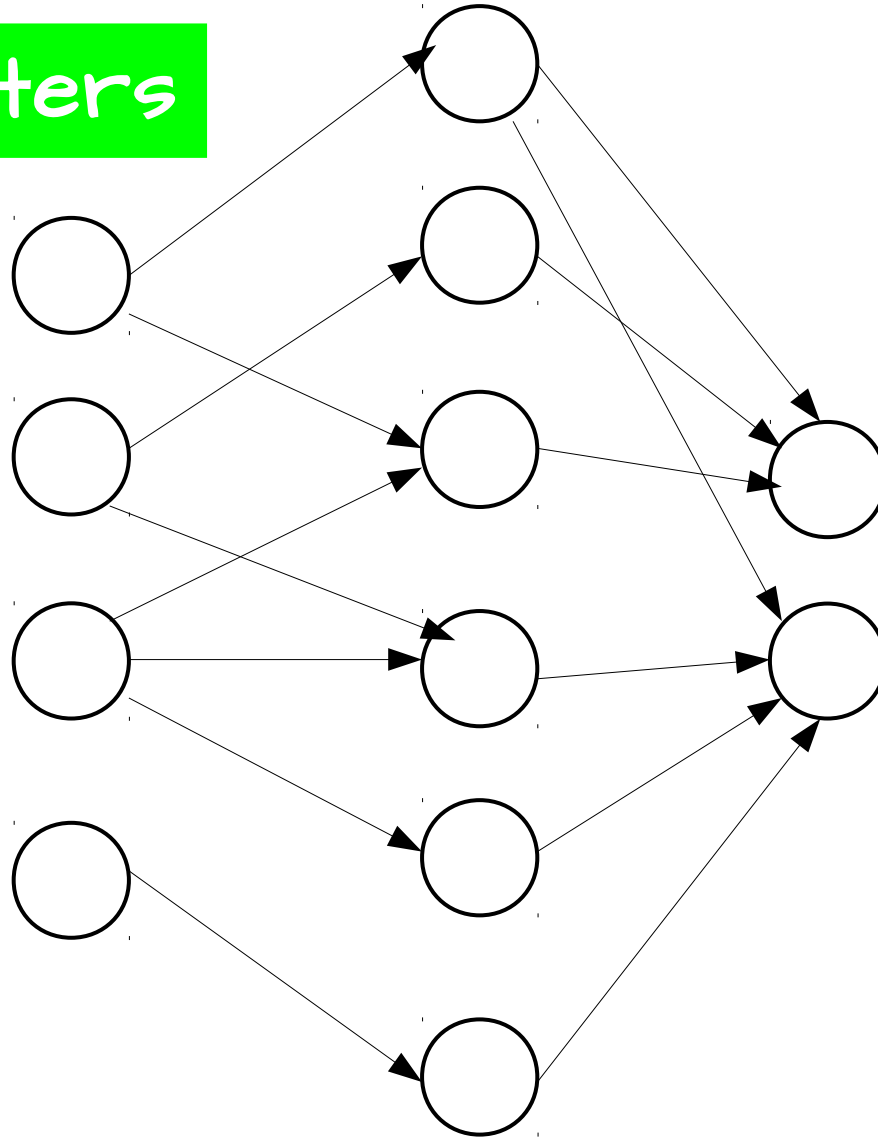
# Activation



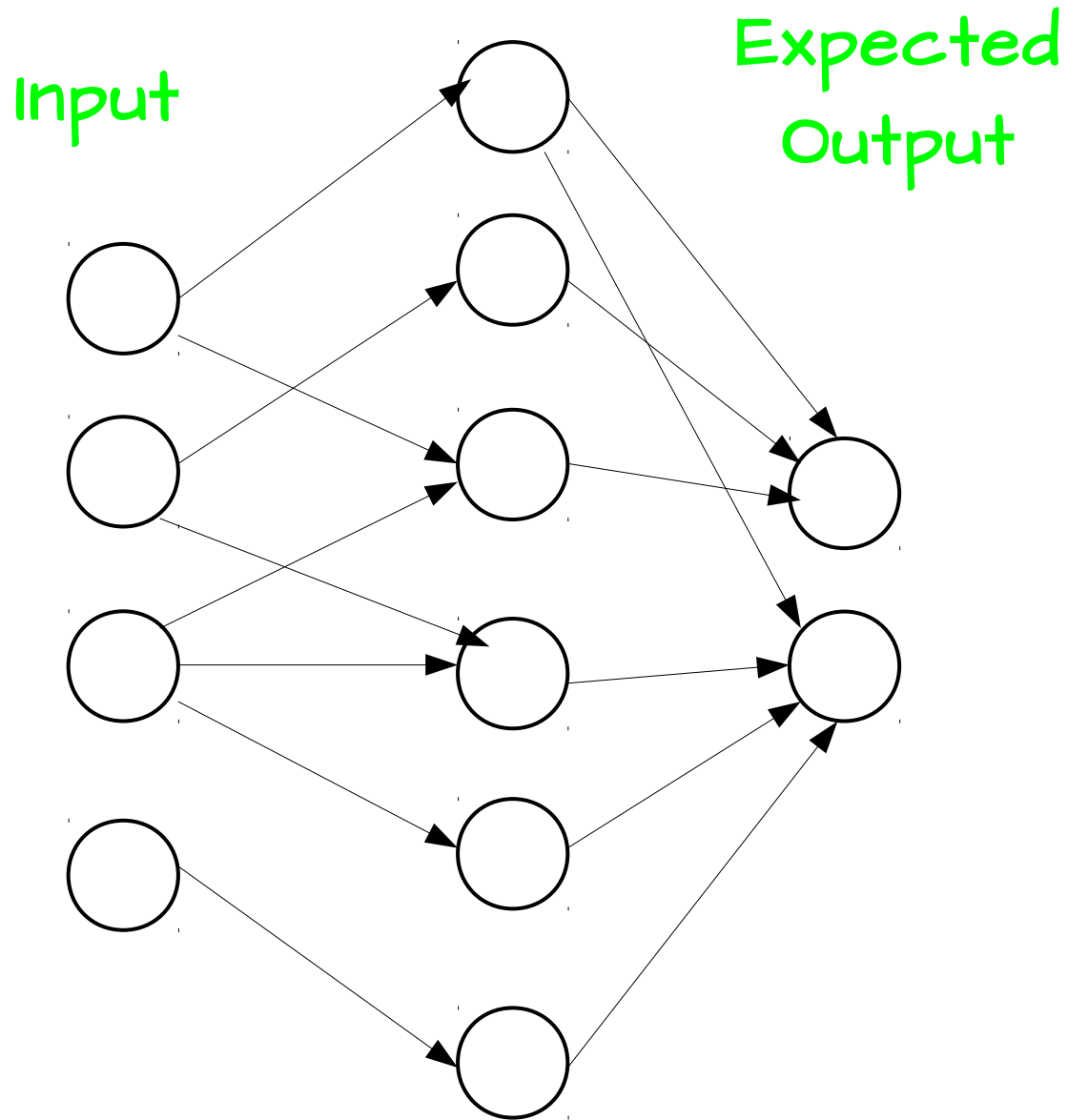
# Training



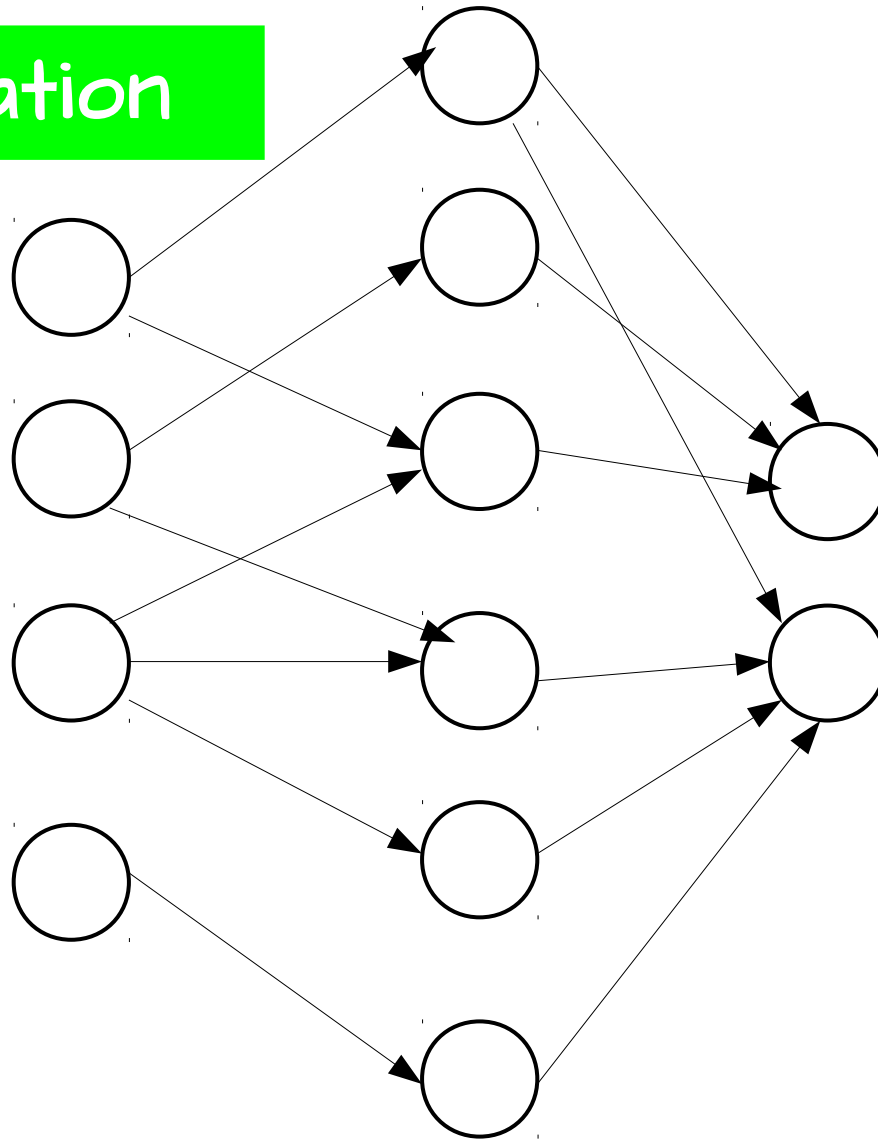
Adjust parameters



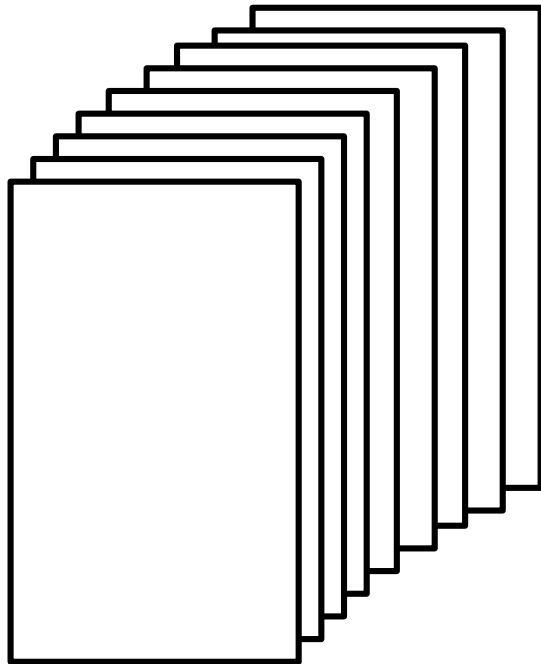
# Supervised Learning



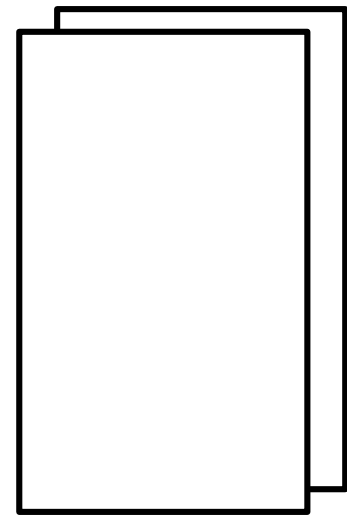
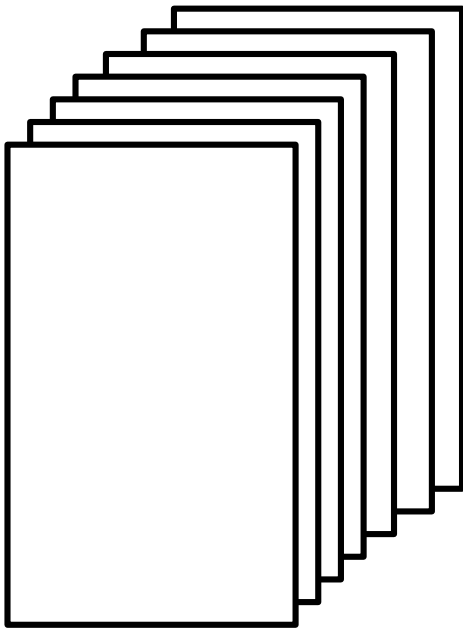
# Backpropagation



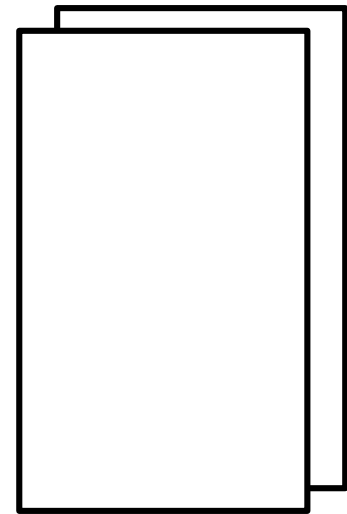
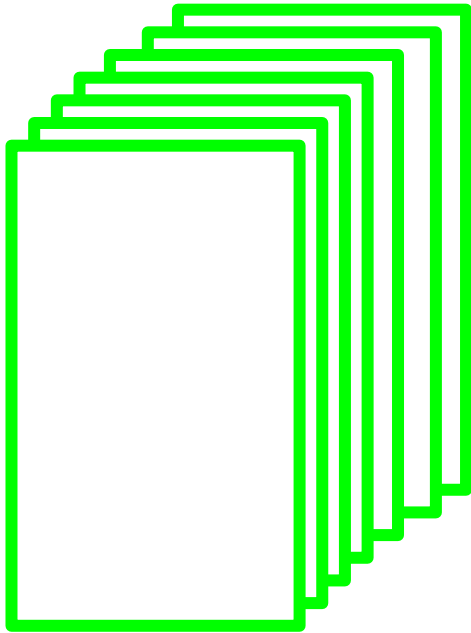
Data set



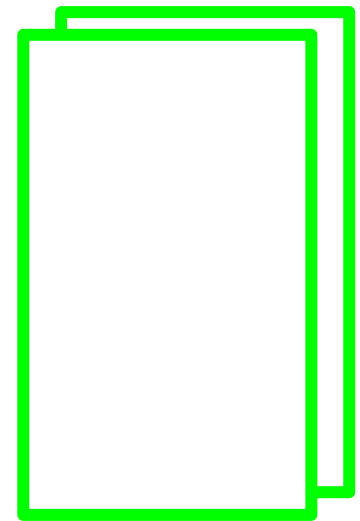
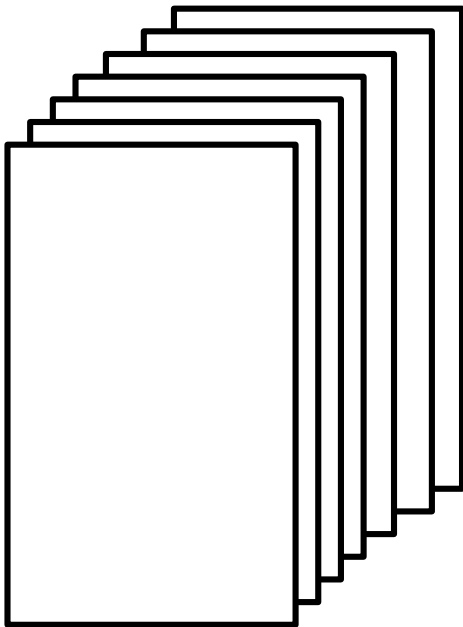
# Training data + test data



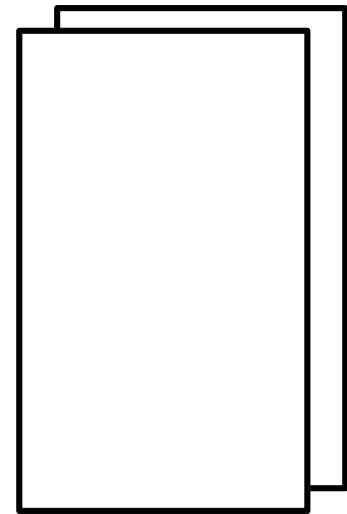
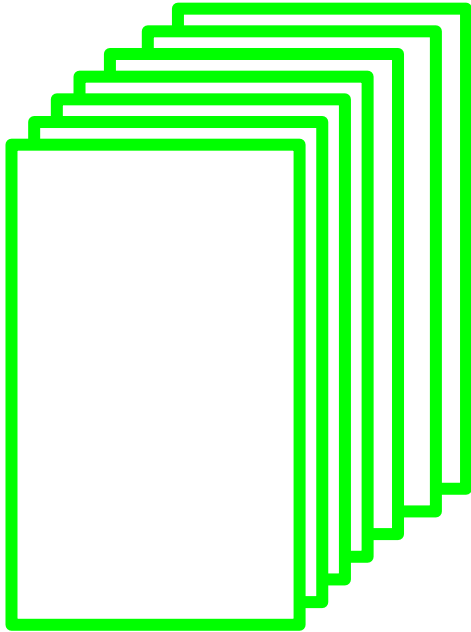
# Training



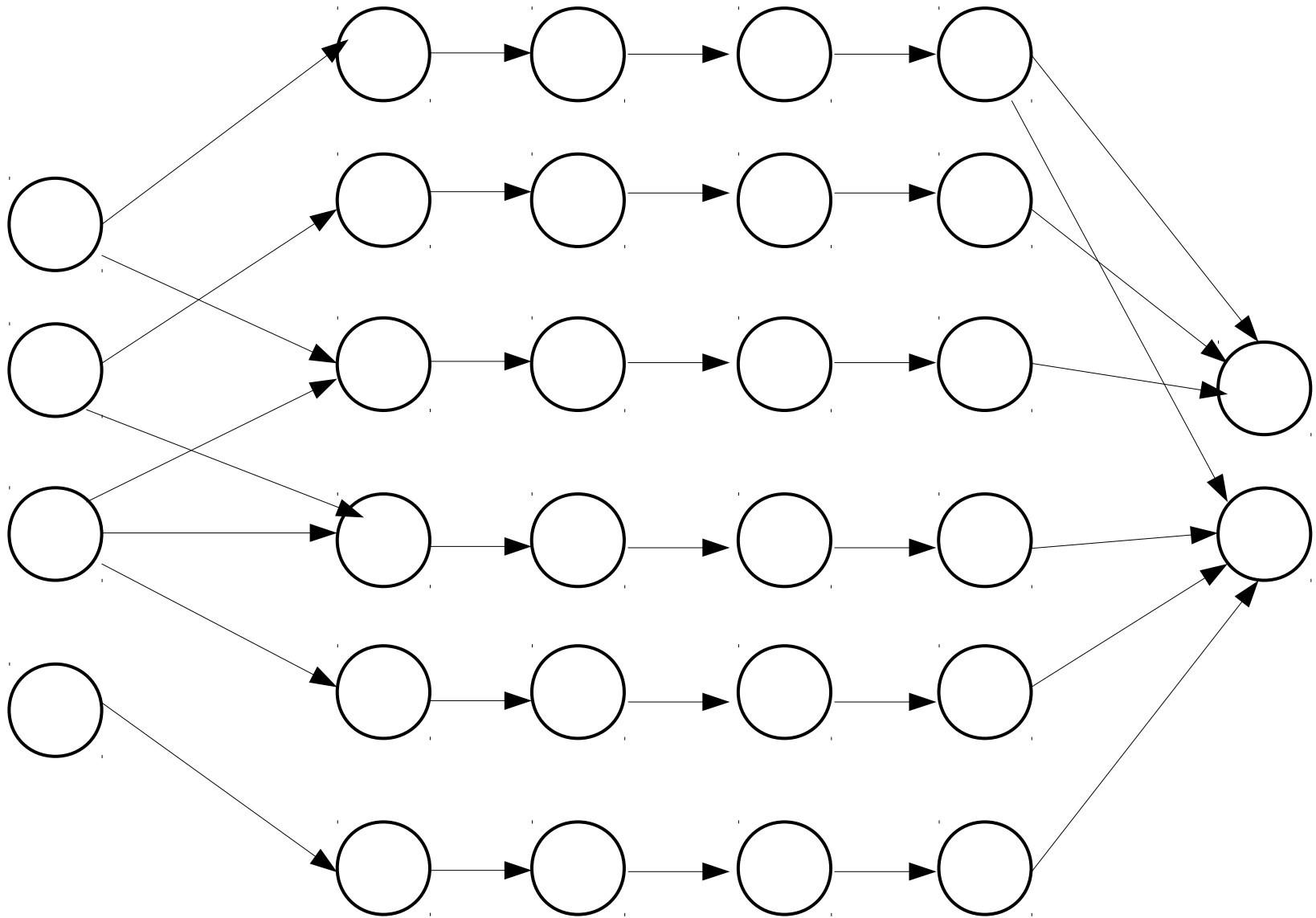
Verify



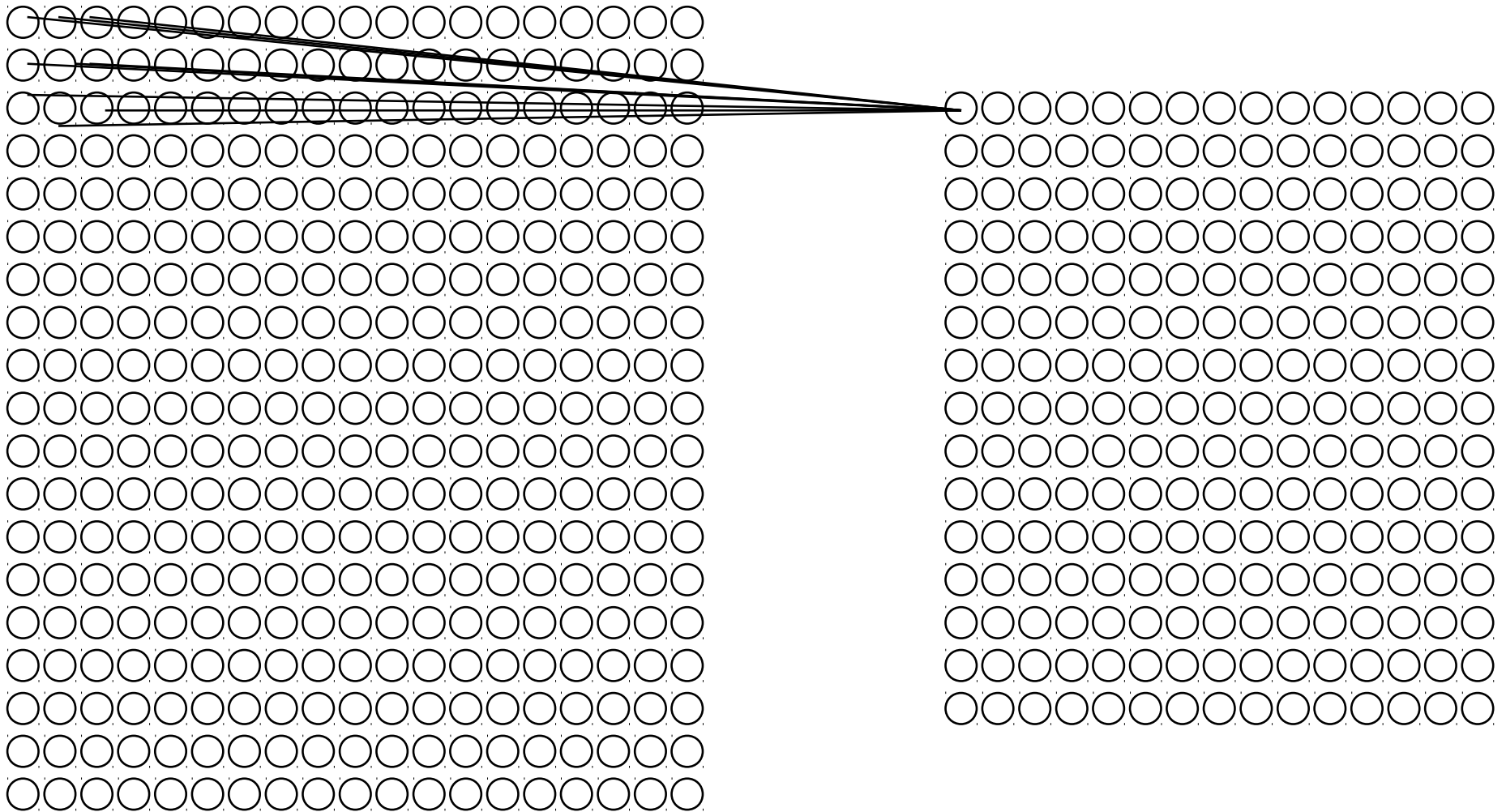
# Overfitting



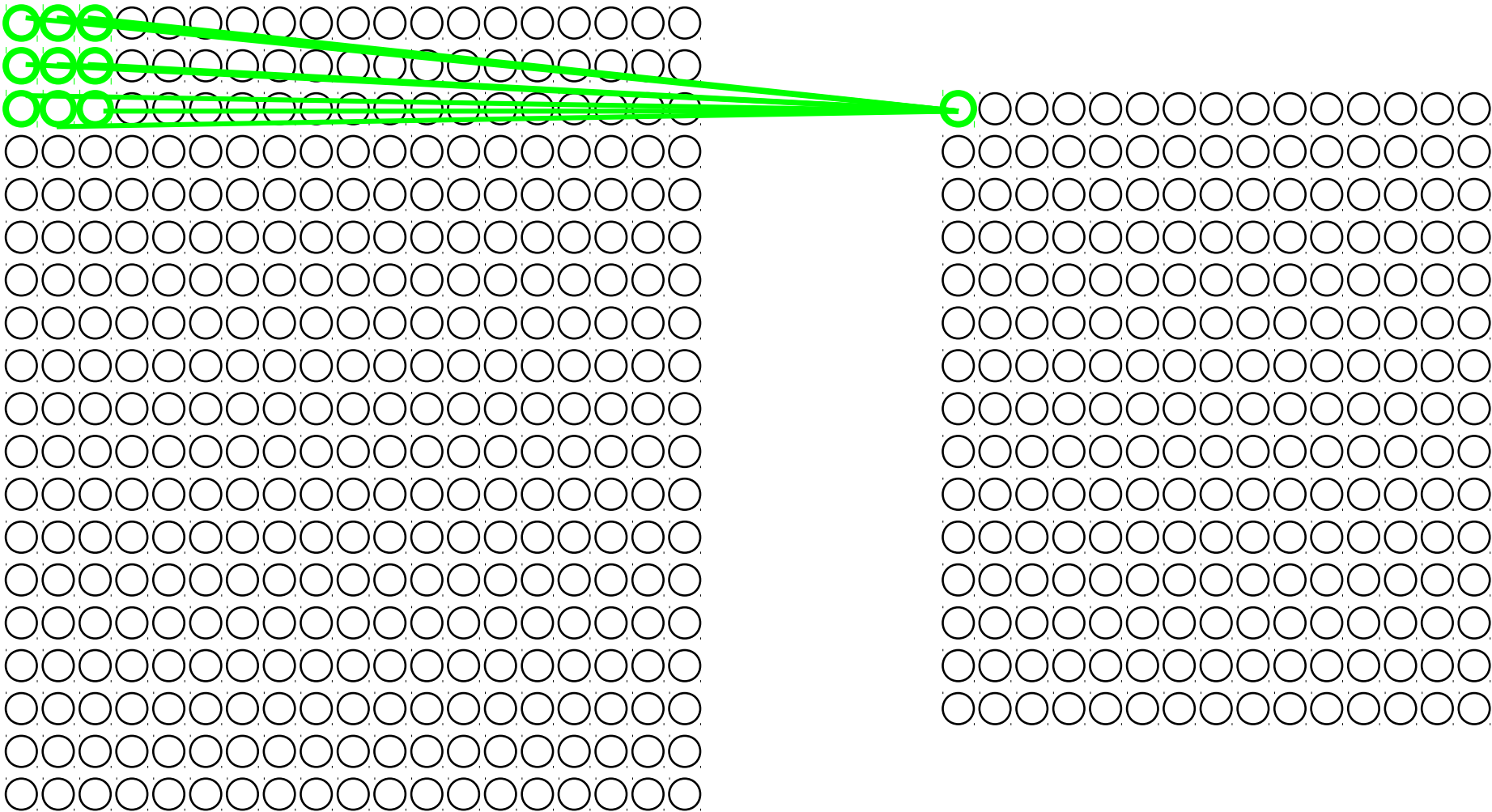
# Deep Neural Networks



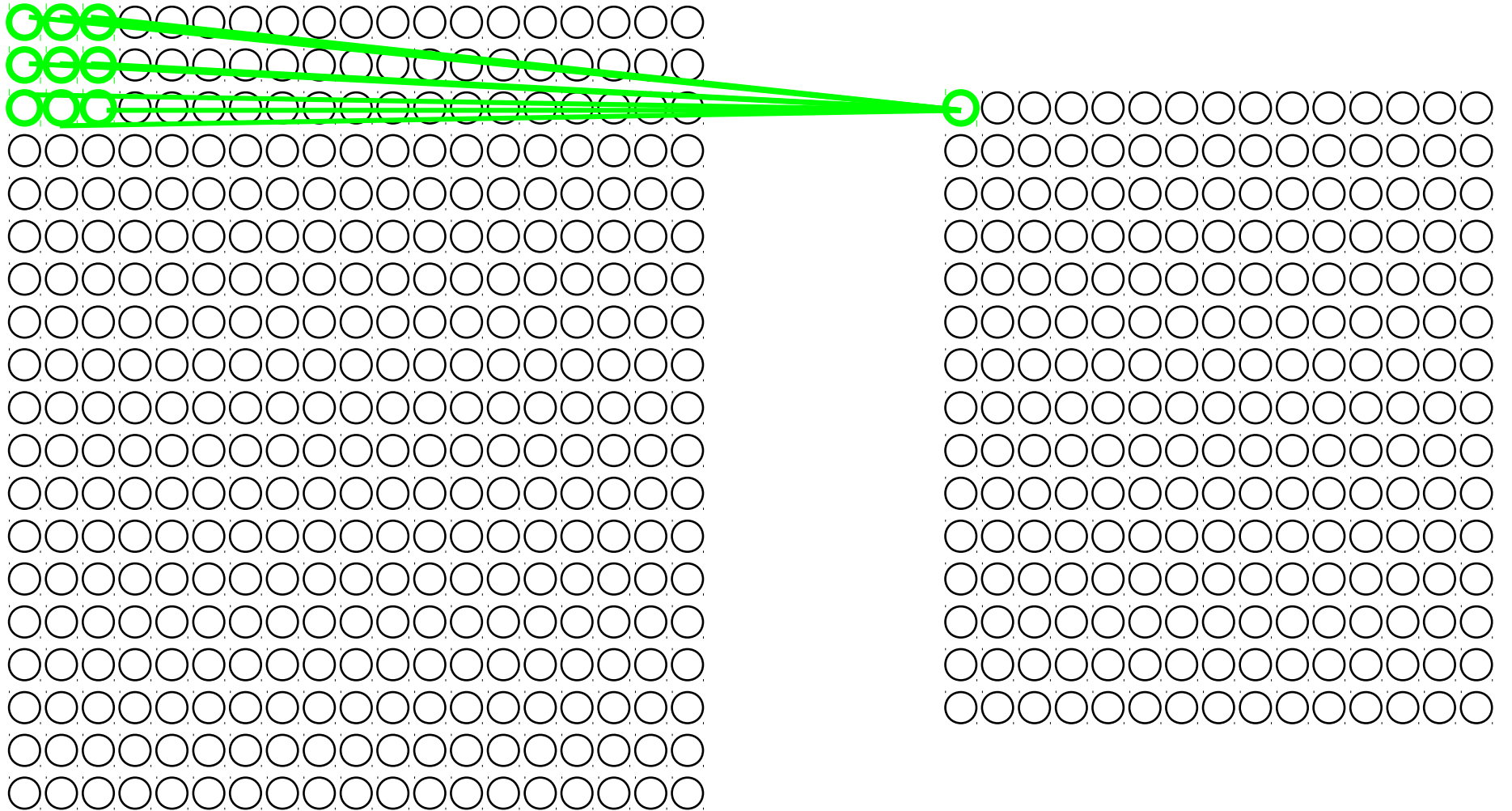
# Convolutional Neural Networks



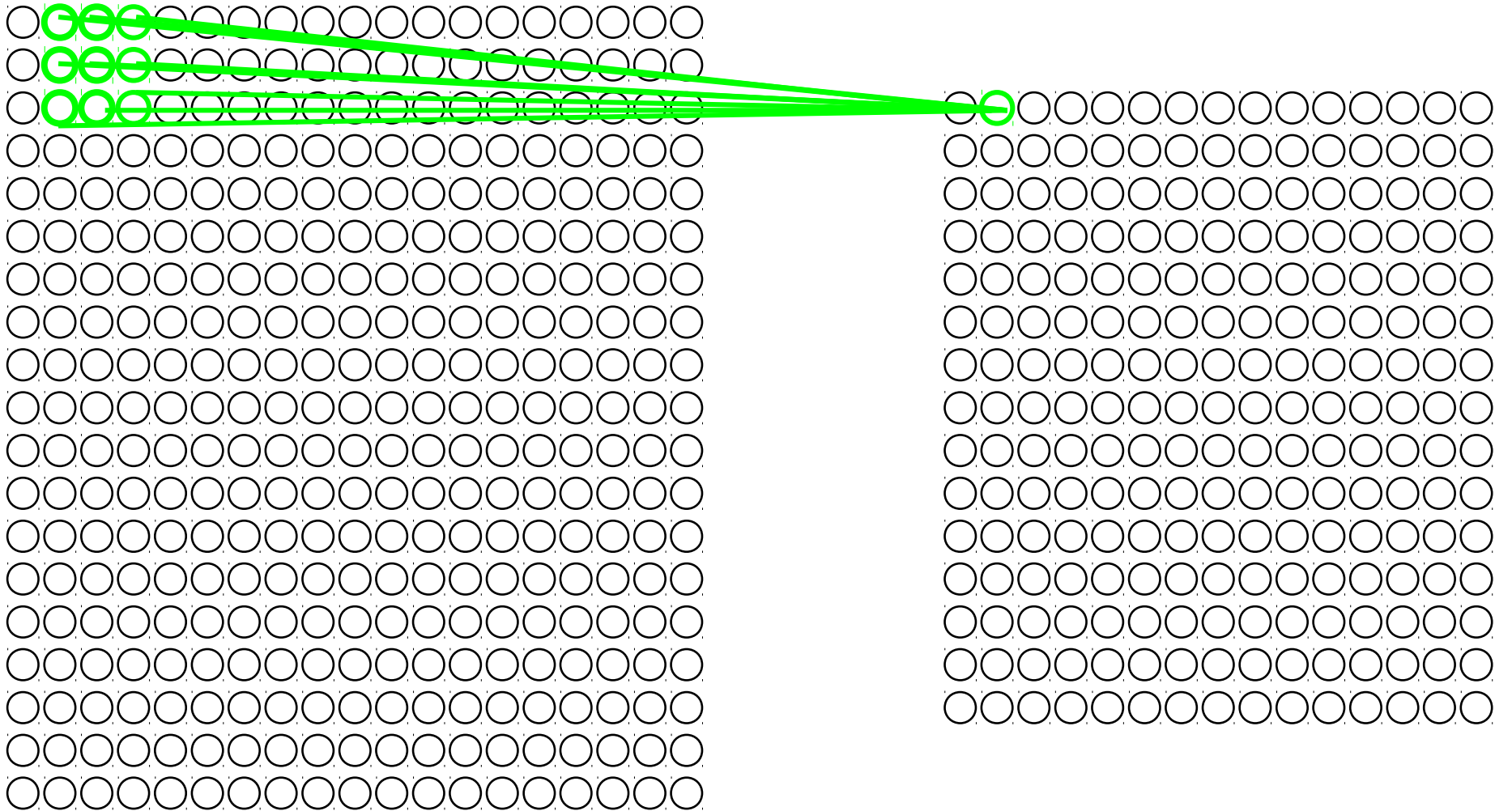
# Local Receptive Field



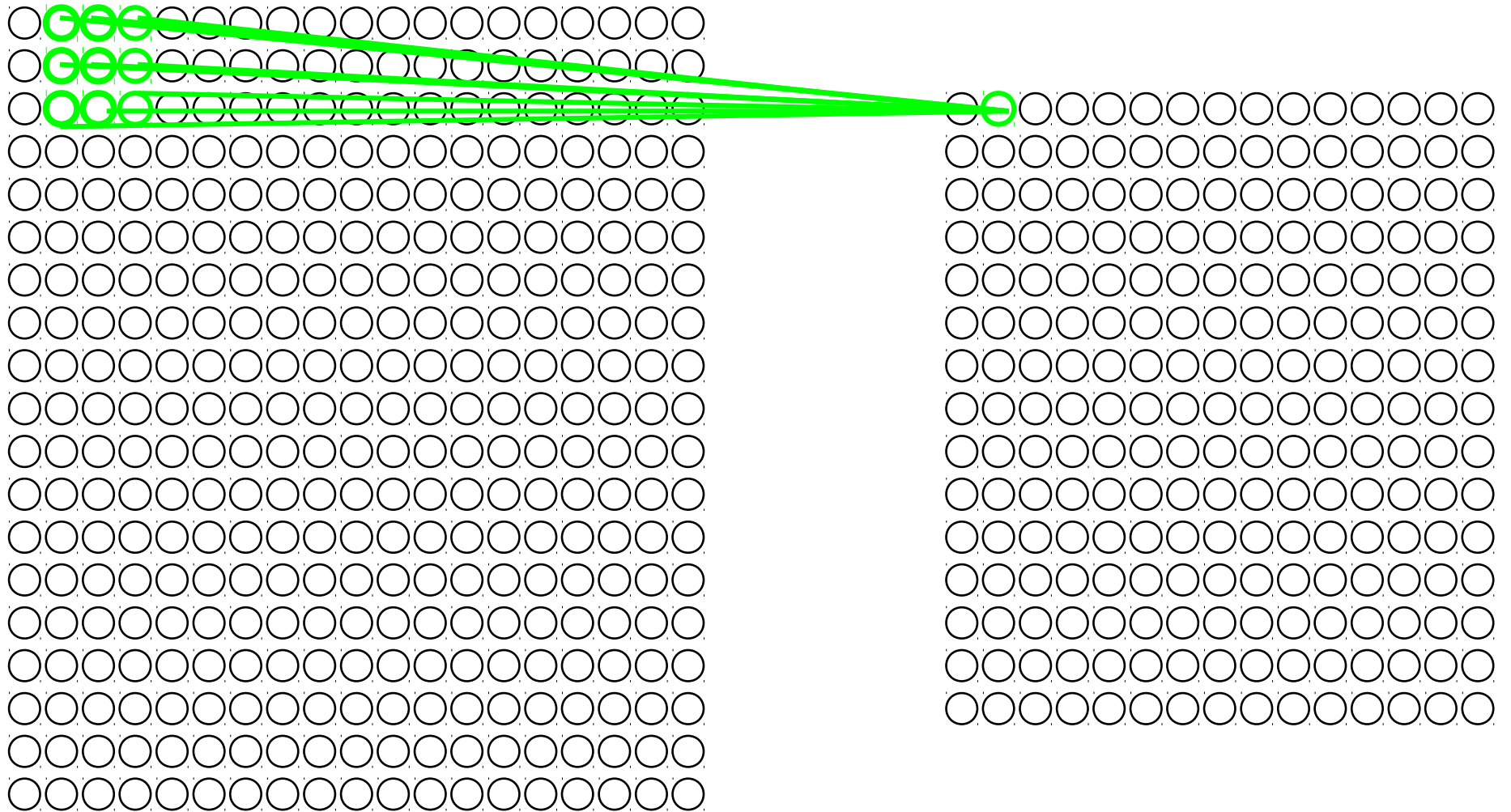
# Feature Map



# Stride



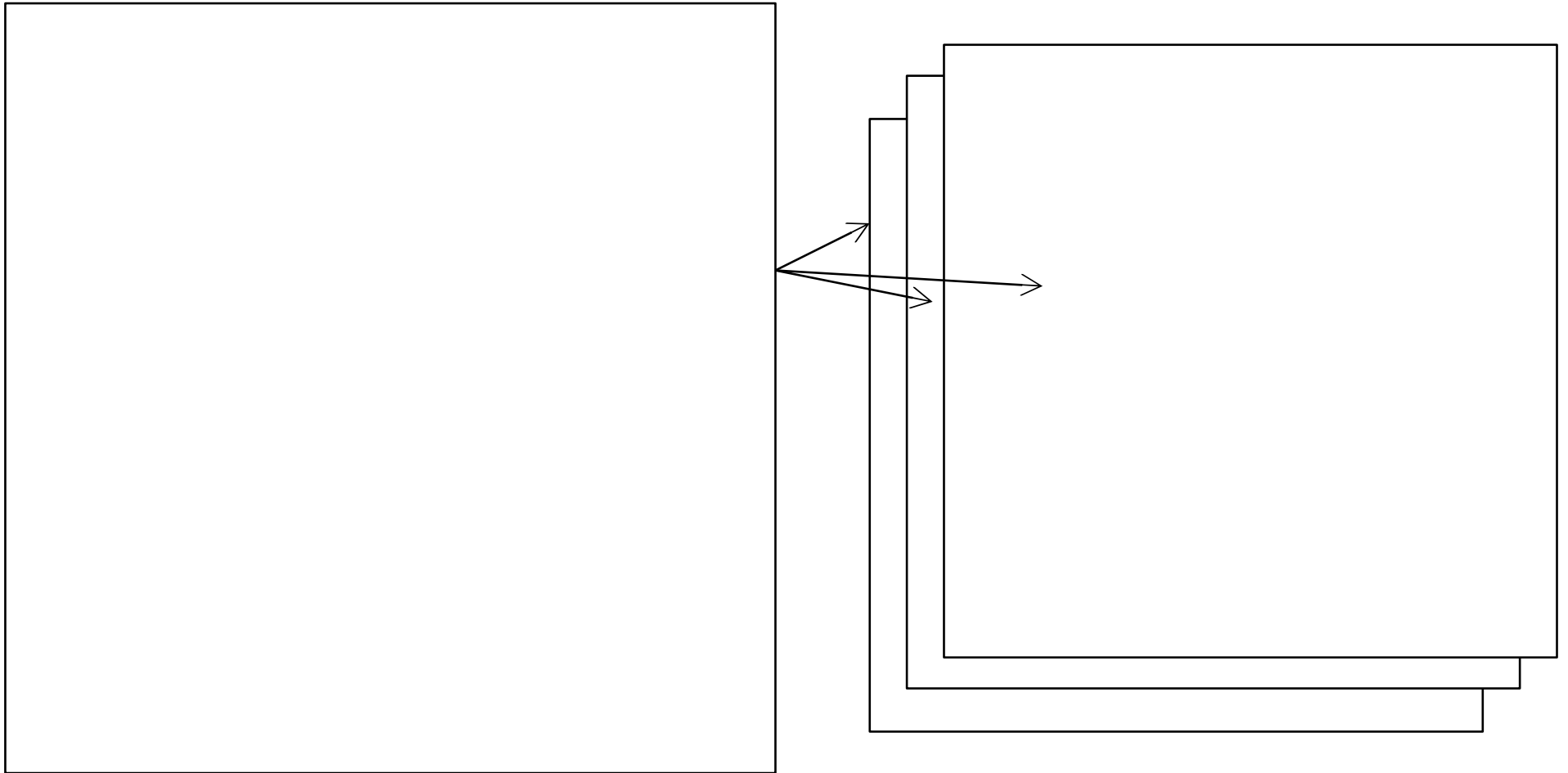
# Shared weights and biases



# Multiple Feature maps/filters

19 x 19

3 x 17 x 17

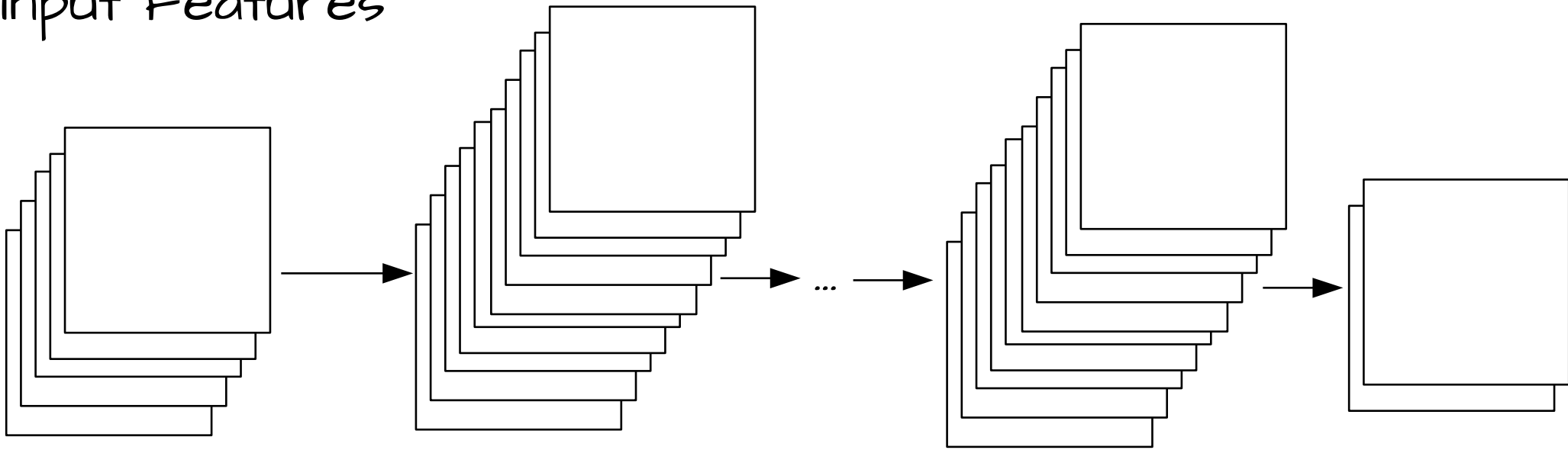


# Architecture

12 layers with 64 - 192  
filters

Output

Input Features

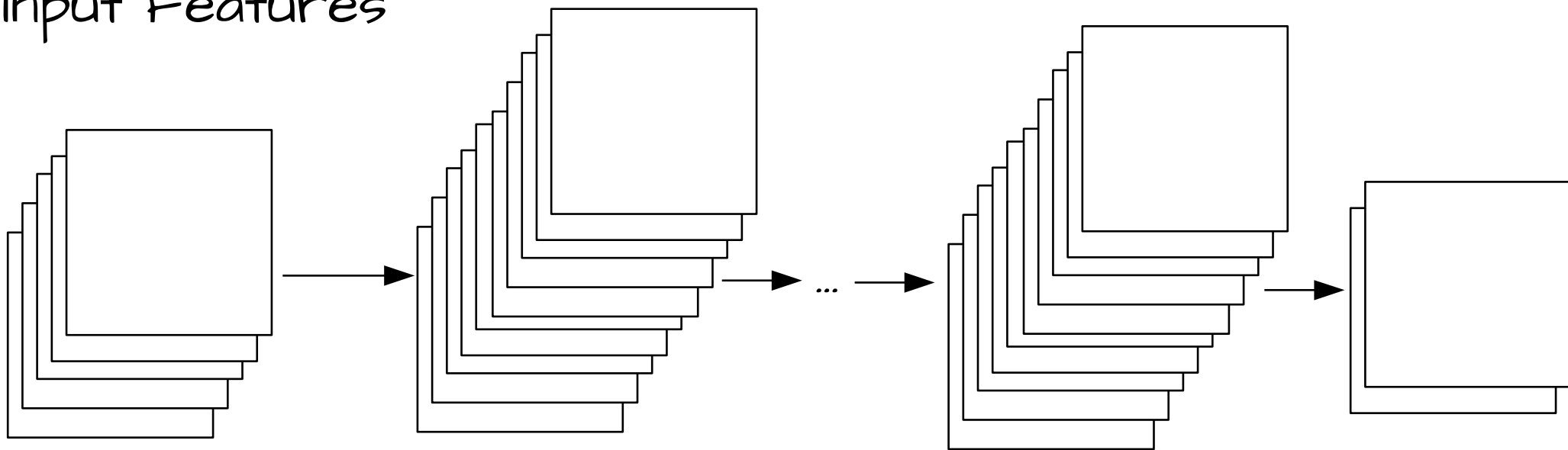


# Architecture

12 layers with 64 - 192  
filters

Output

Input Features

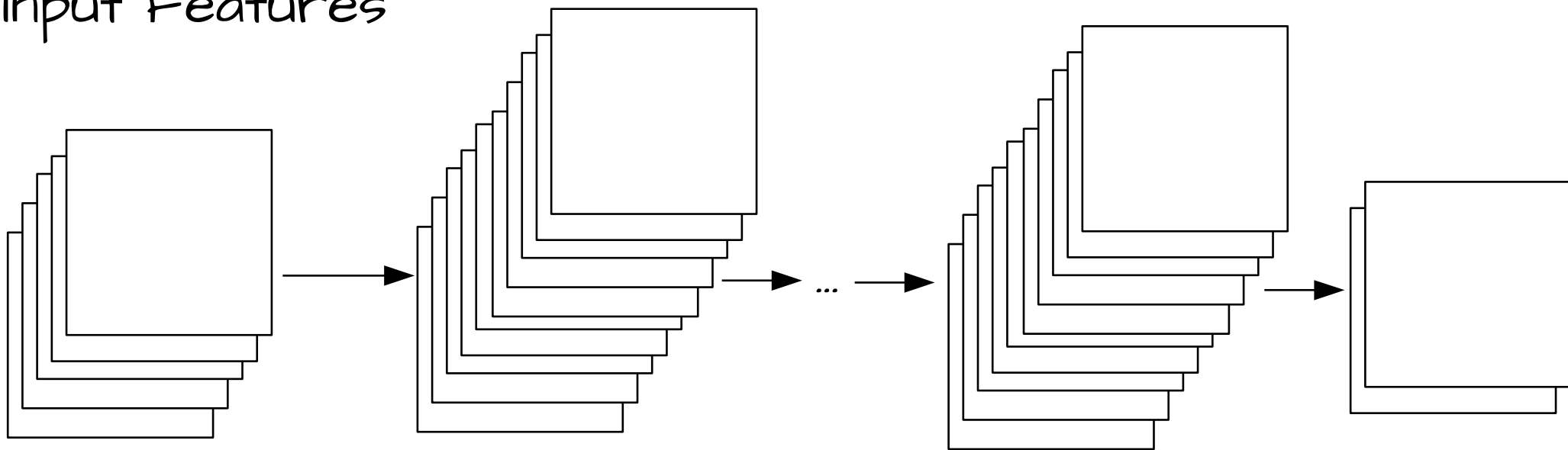


# Architecture

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

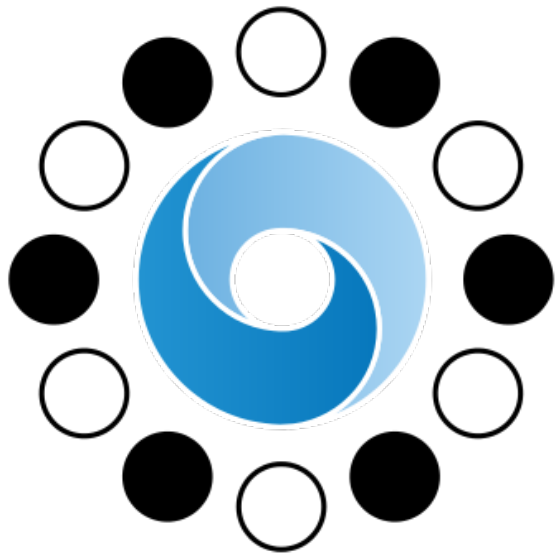
Mostly beats GnuGo

Combined with MCTS in the  
**Selection**

# Asynchronous GPU Power



Revolution



AlphaGo

# Networks in Training

Rollout policy

SL policy network

RL policy network

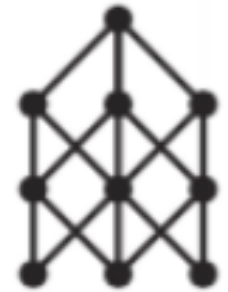
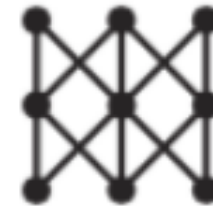
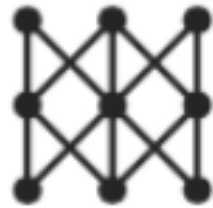
Value network

$P_{\pi}$

$P_{\sigma}$

$P_{\rho}$

$V_{\theta}$



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

SL policy network

RL policy network

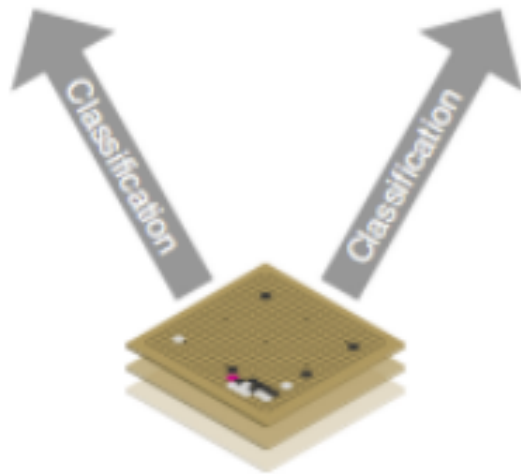
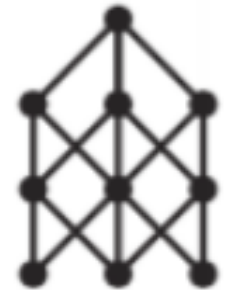
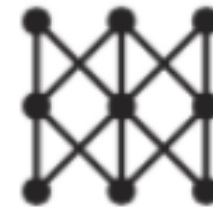
Value network

$P_{\pi}$

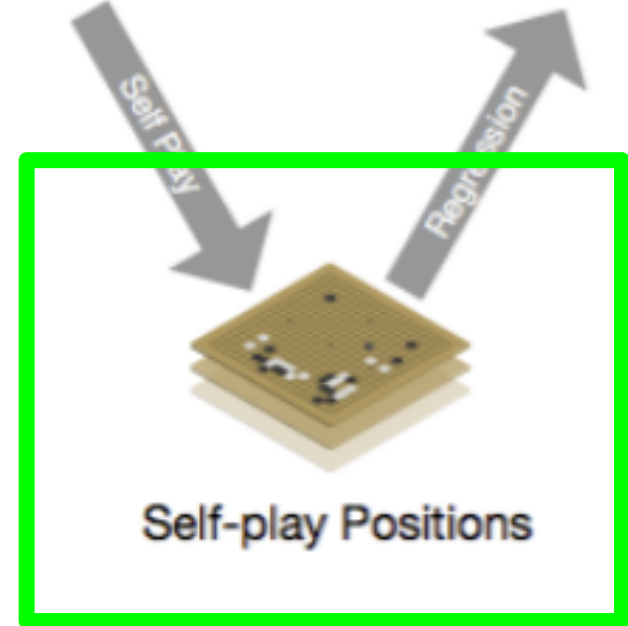
$P_{\sigma}$

$P_{\rho}$

$V_{\theta}$



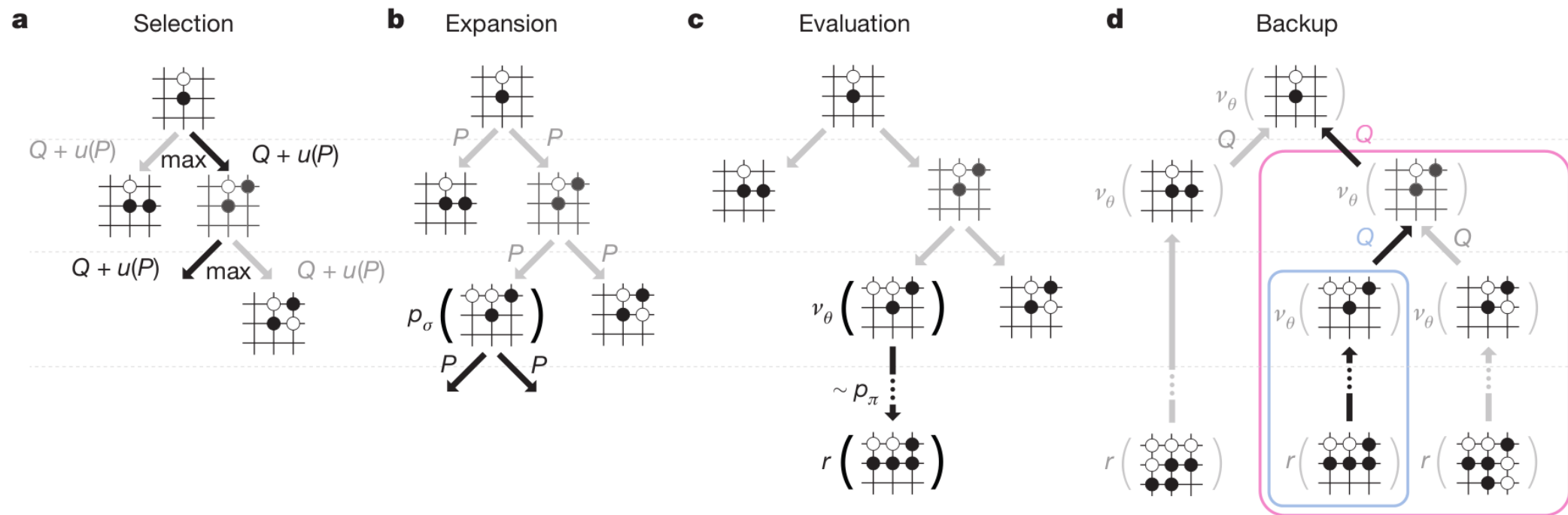
Human expert positions



Self-play Positions

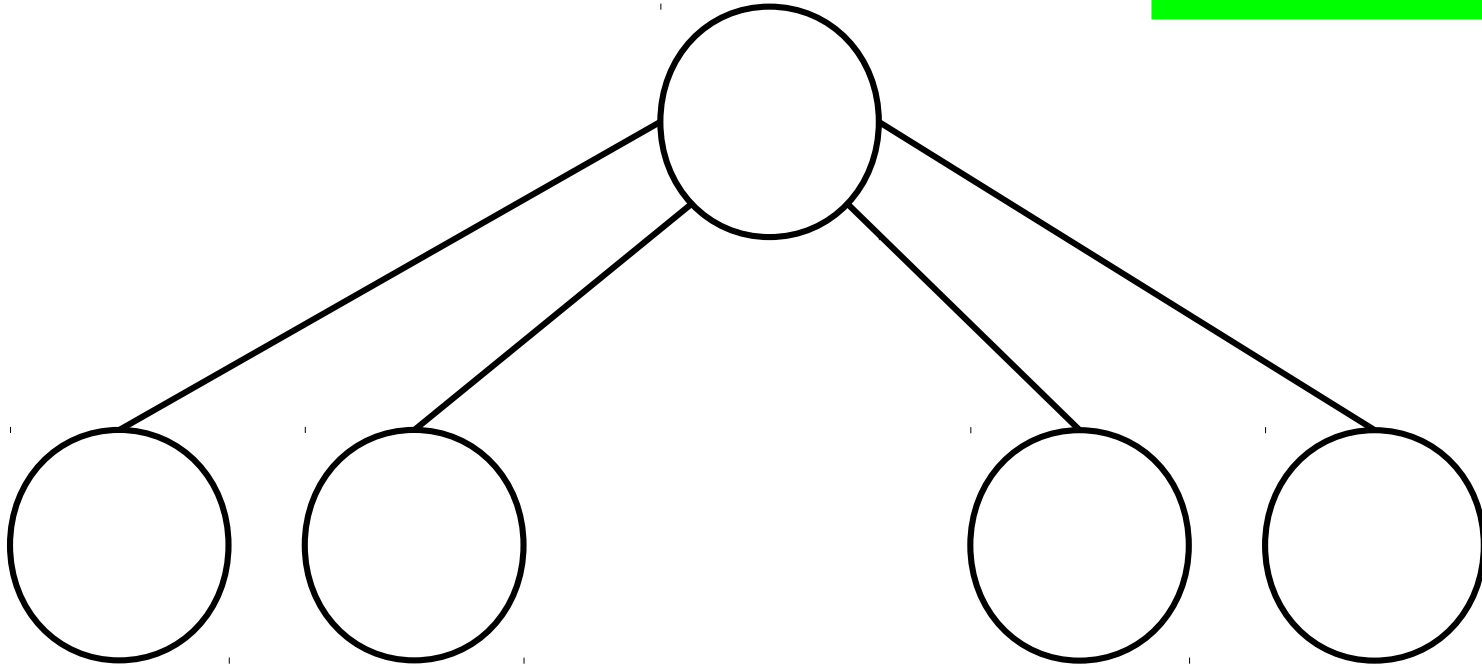
Silver, D. et al., 2016. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), p.484-489.

# AlphaGo Search

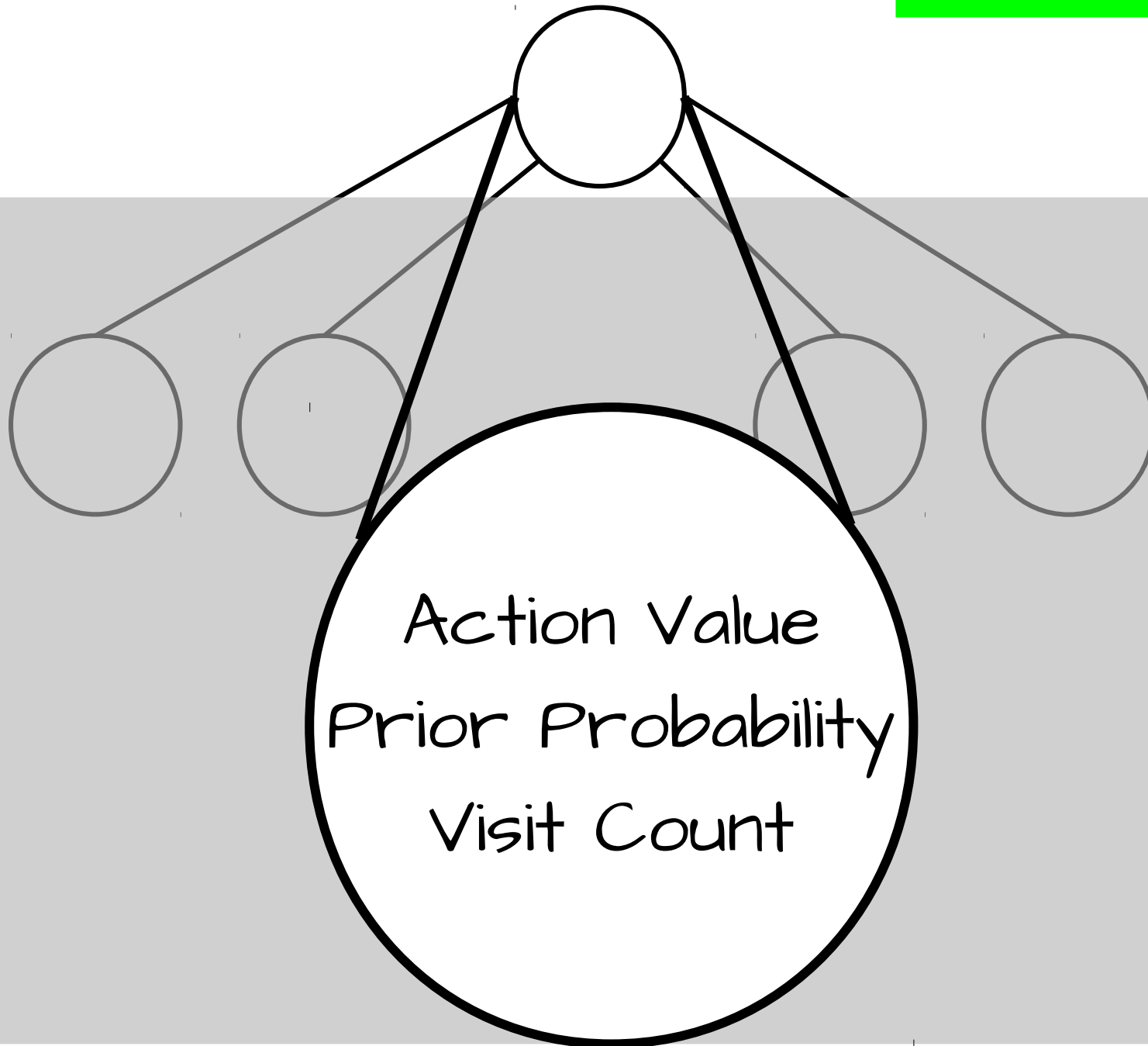


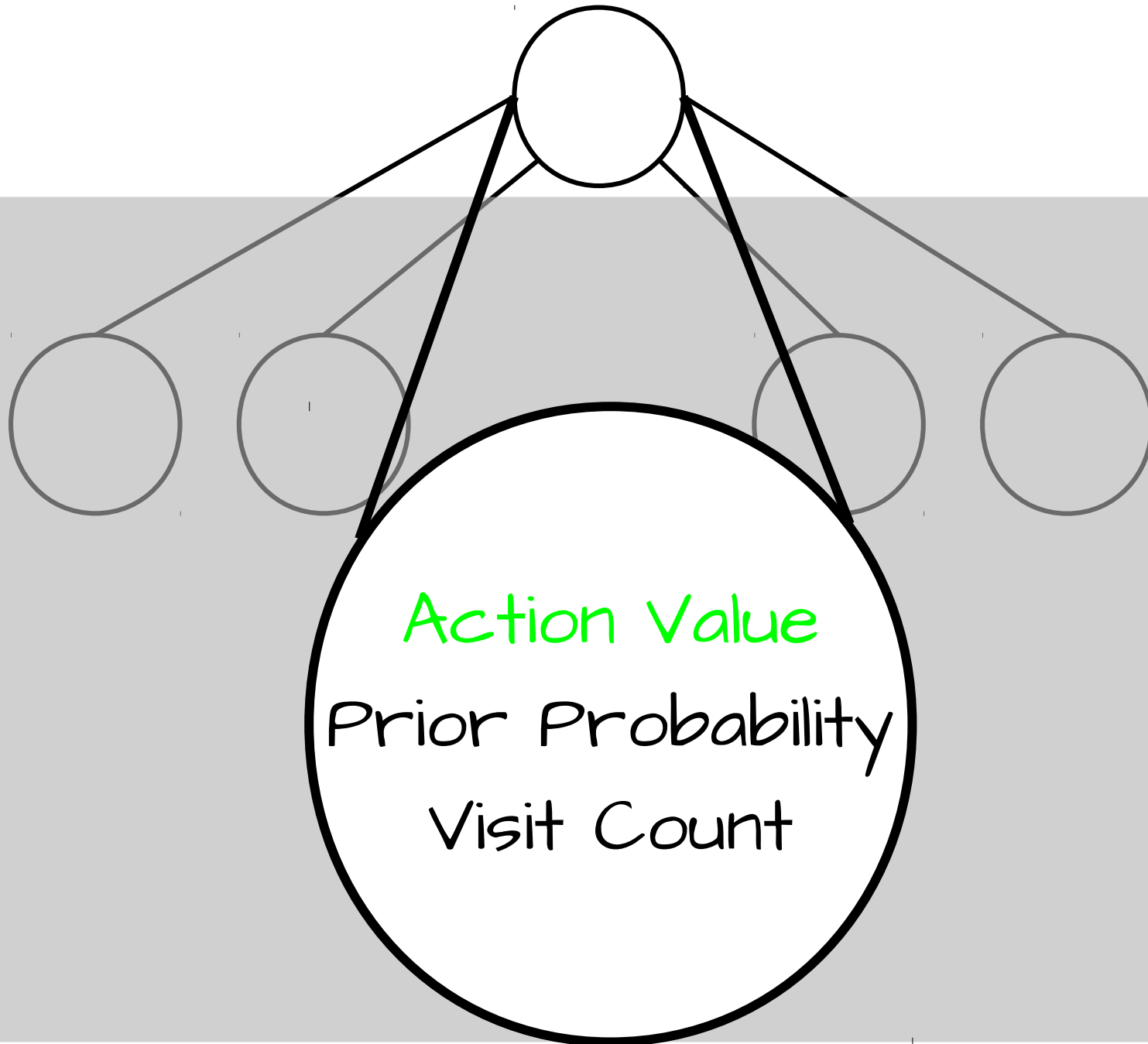
Silver, D. et al., 2016. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), p.484-489.

# Selection

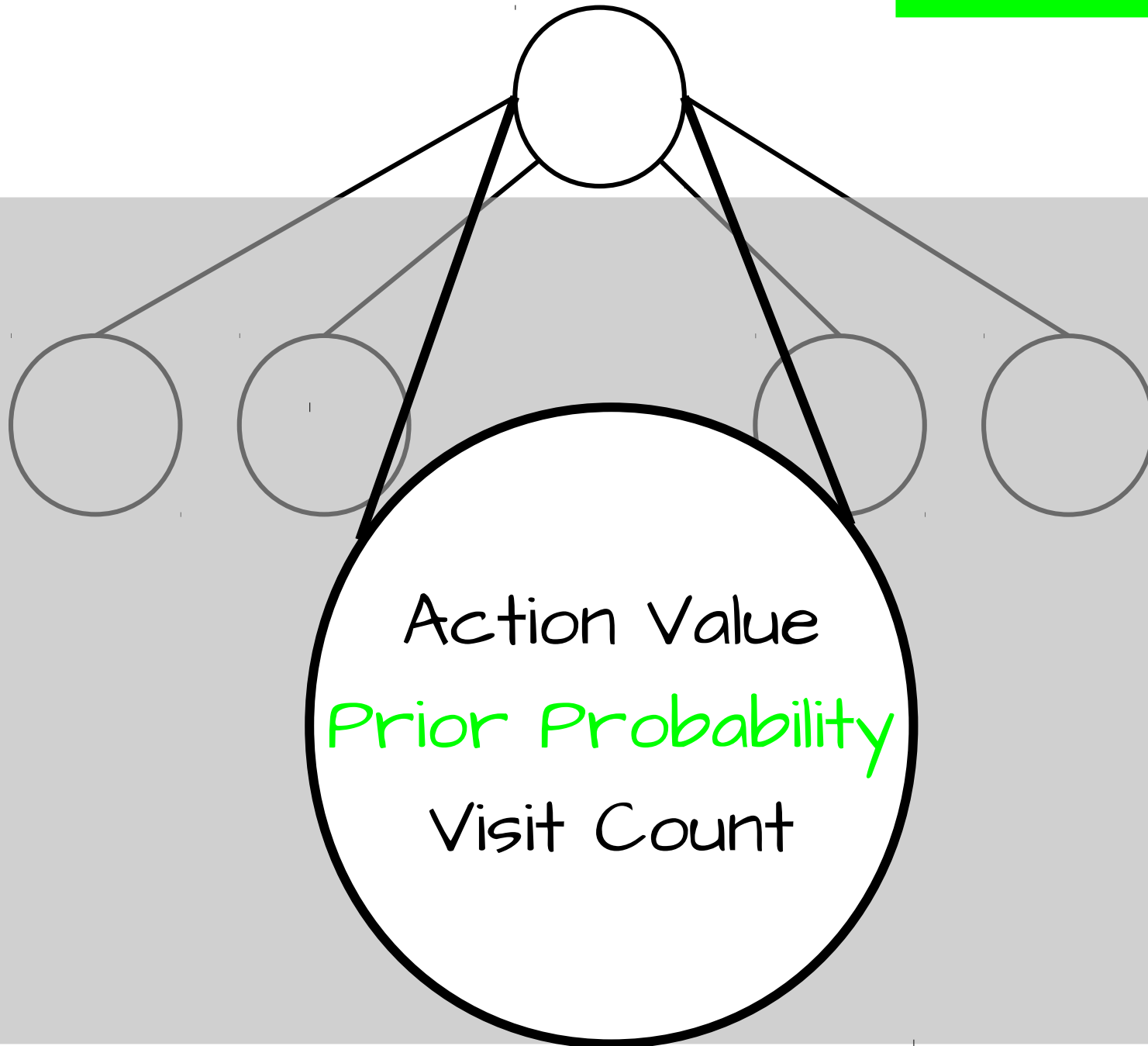


# Selection

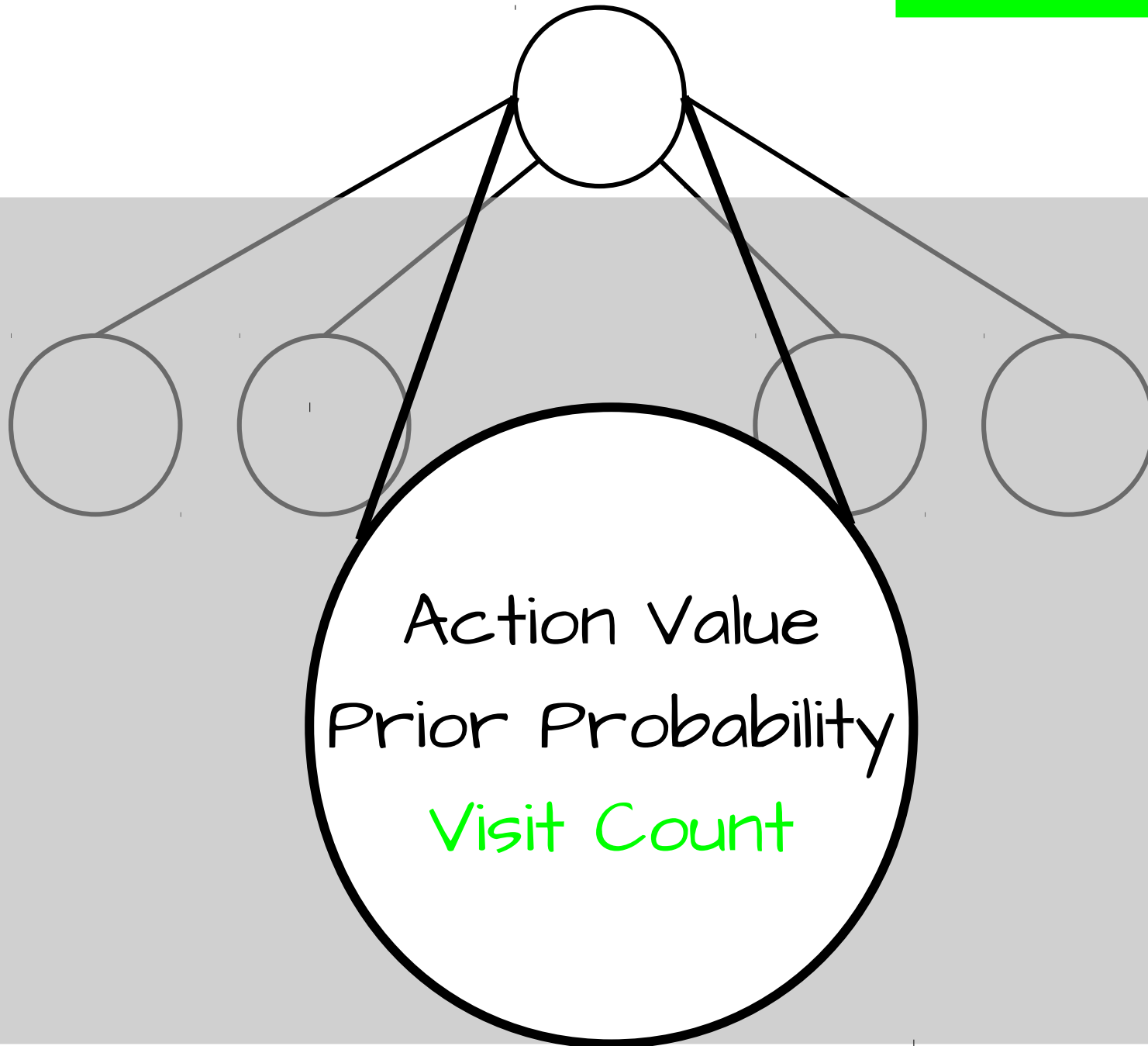




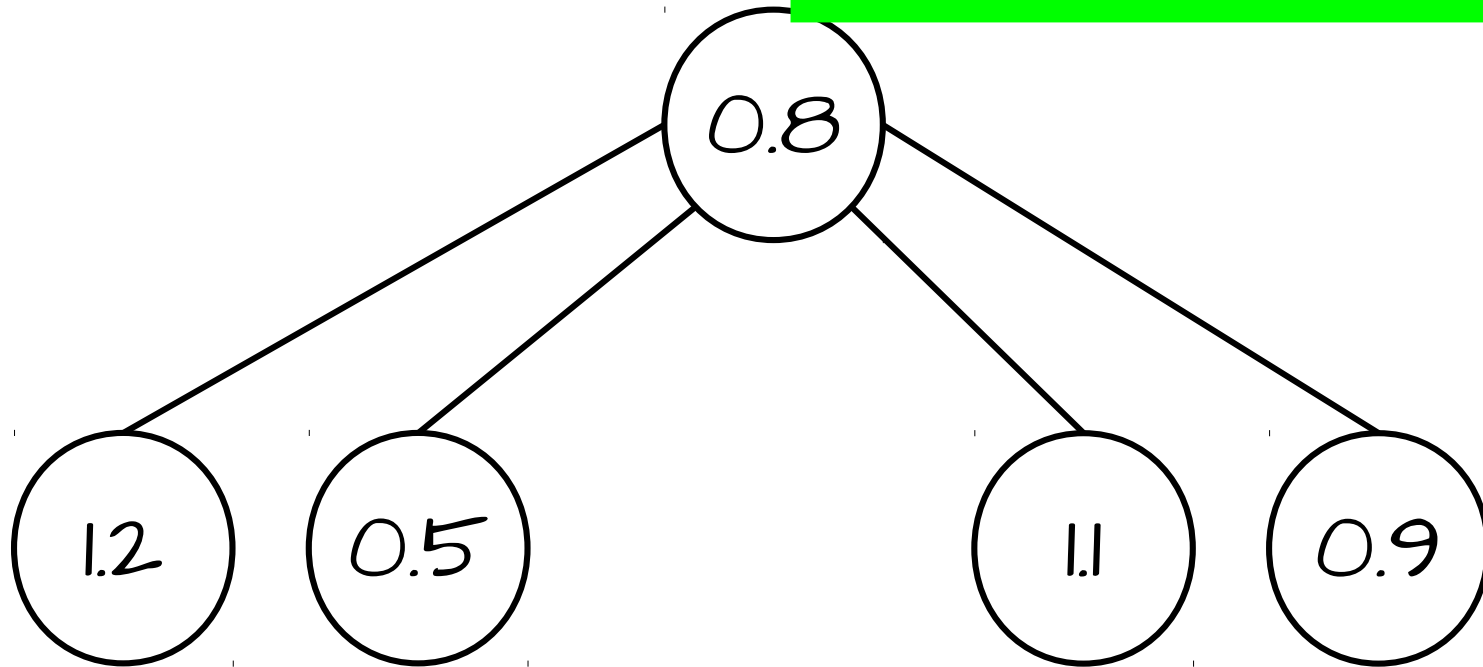
# Selection



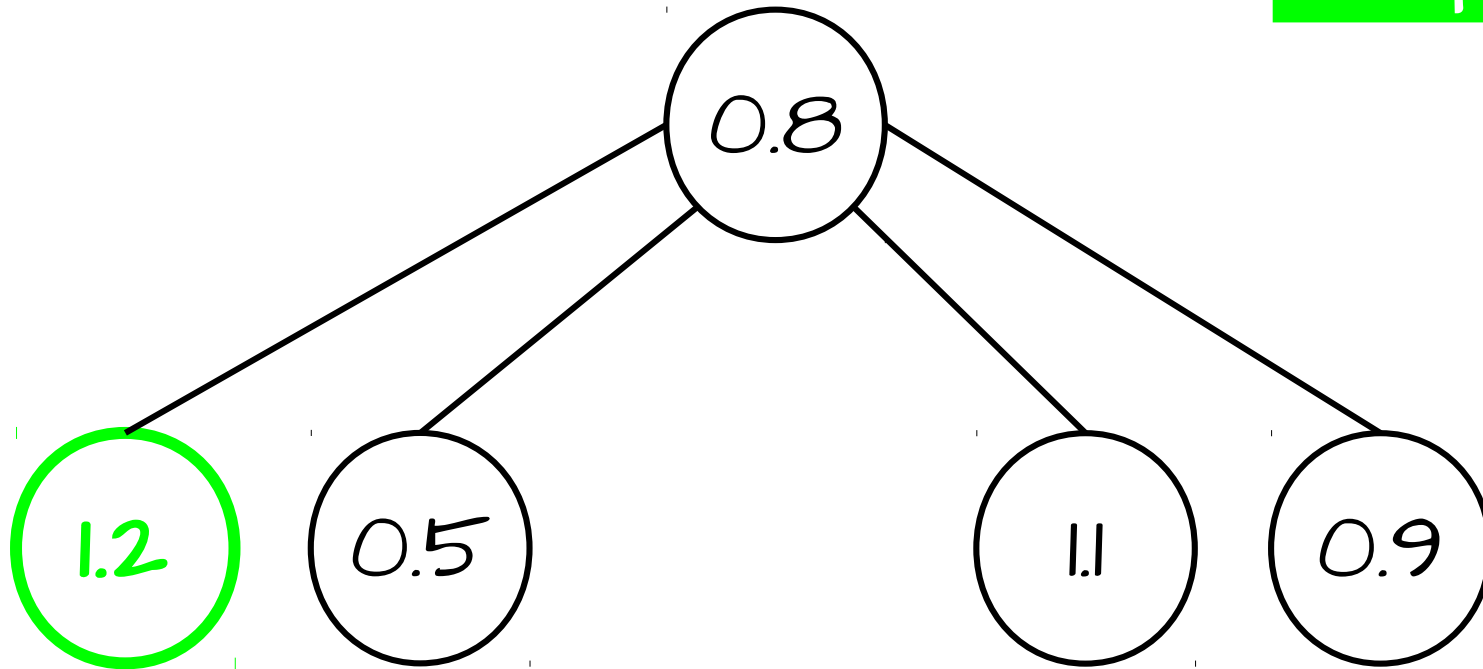
# Selection



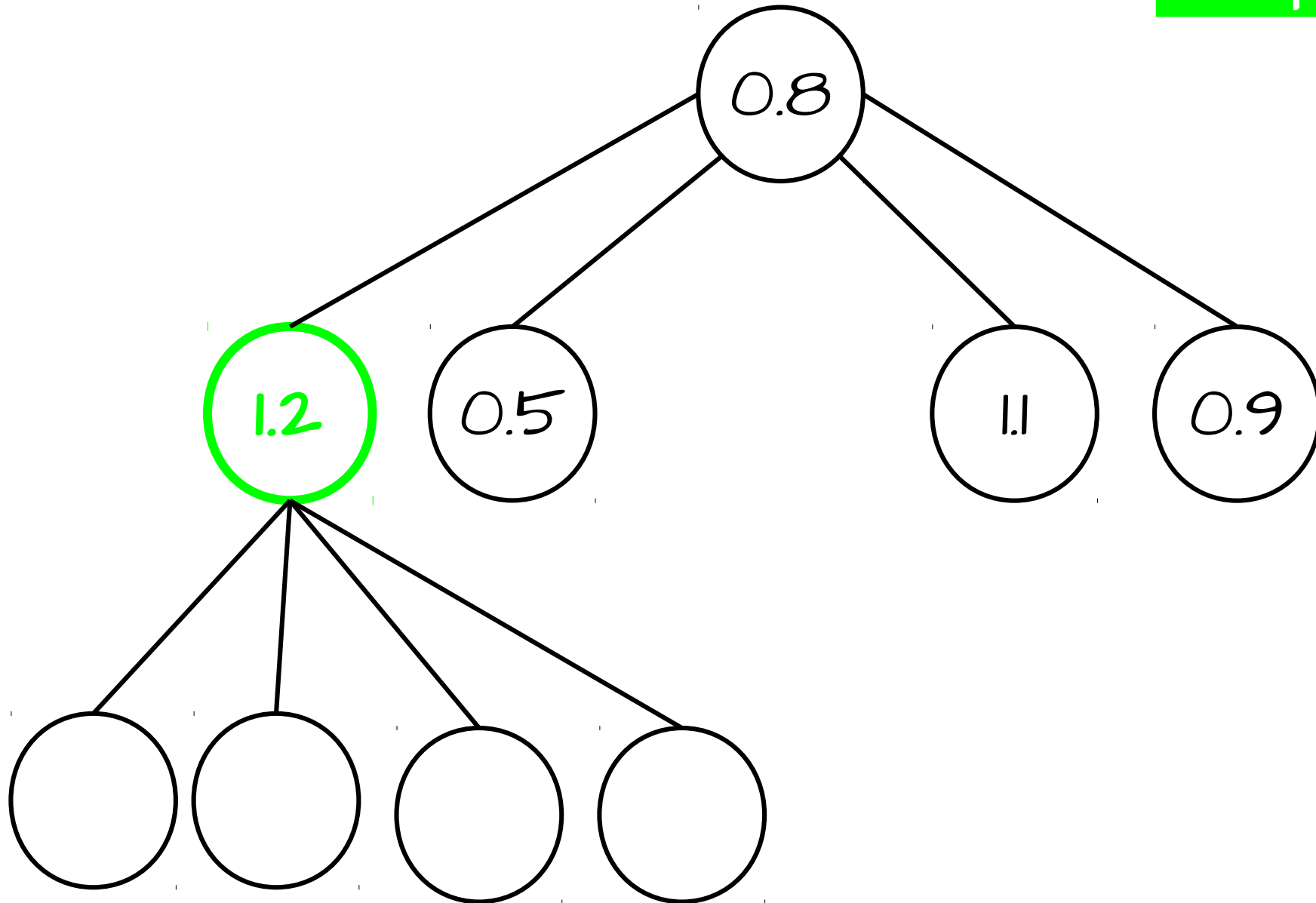
# Action Value + Bonuses



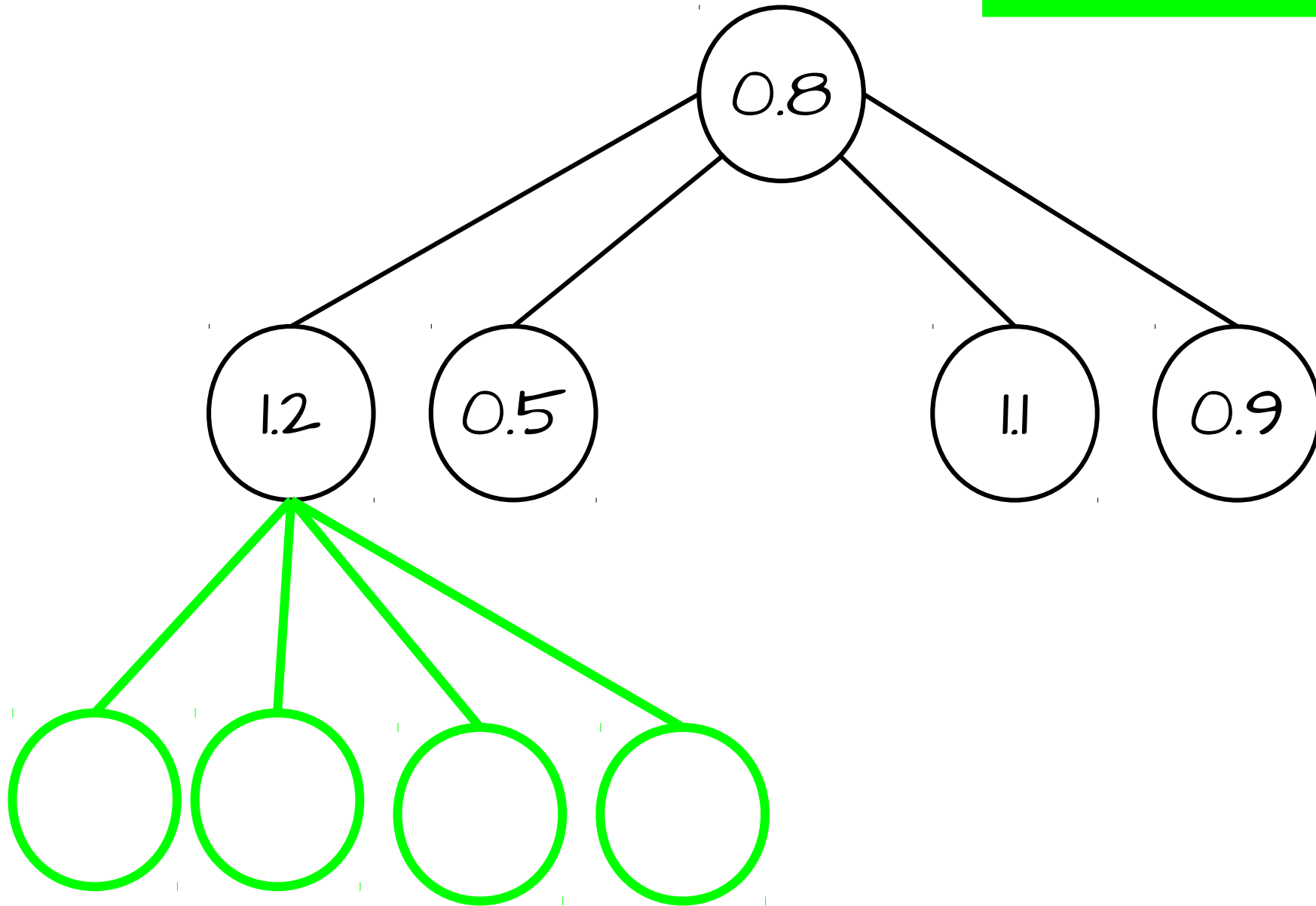
# Expansion



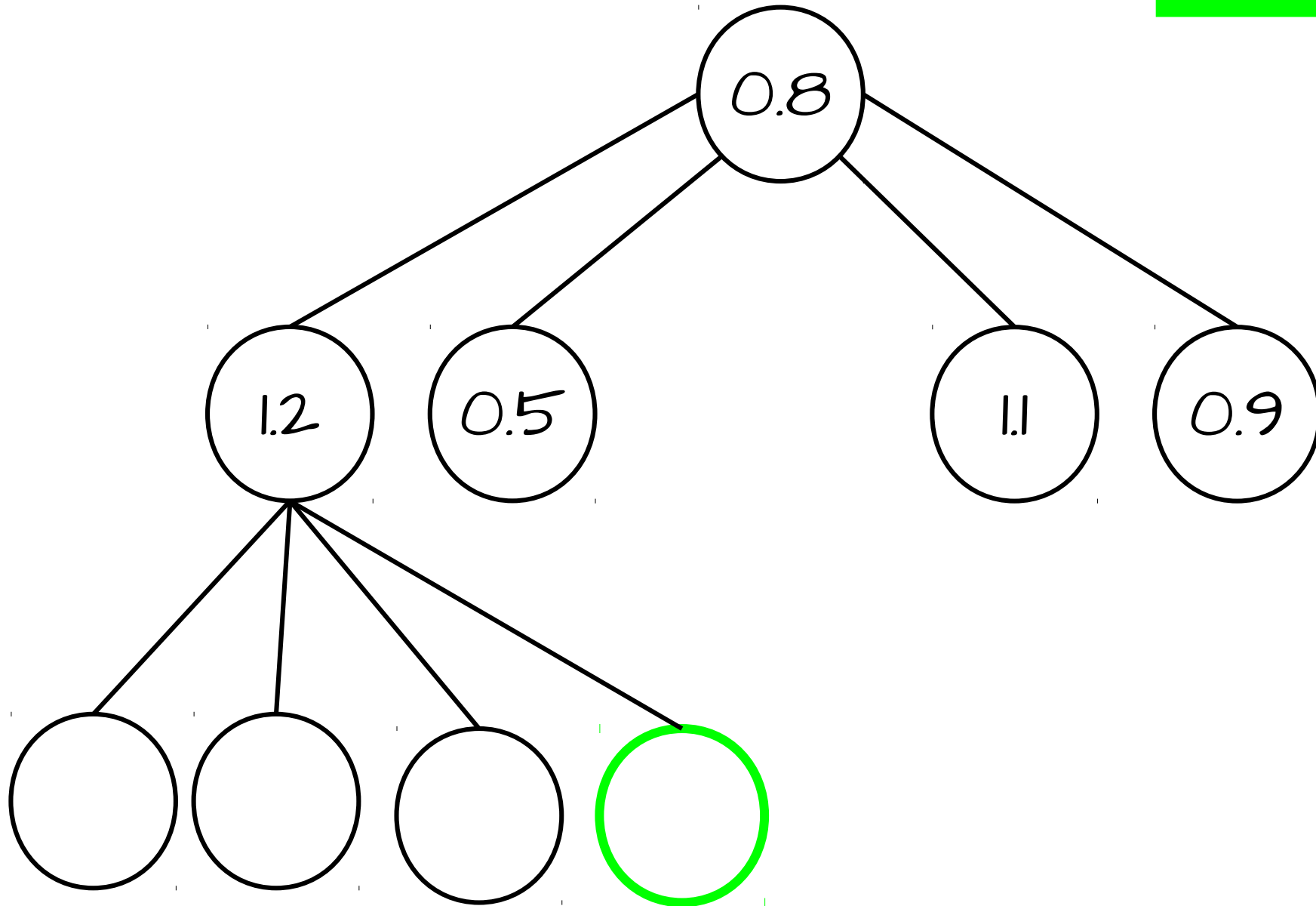
# Expansion



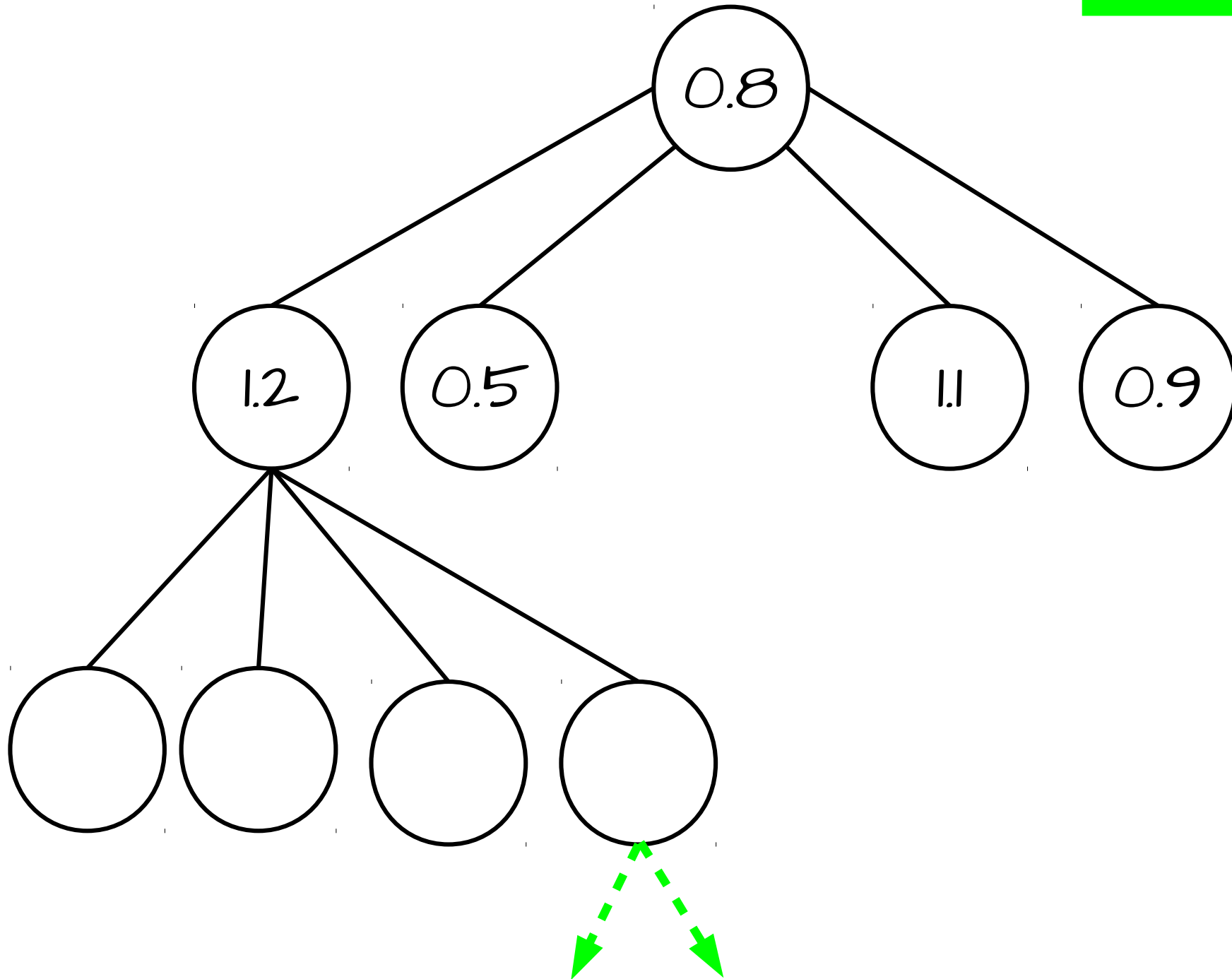
# Prior Probability



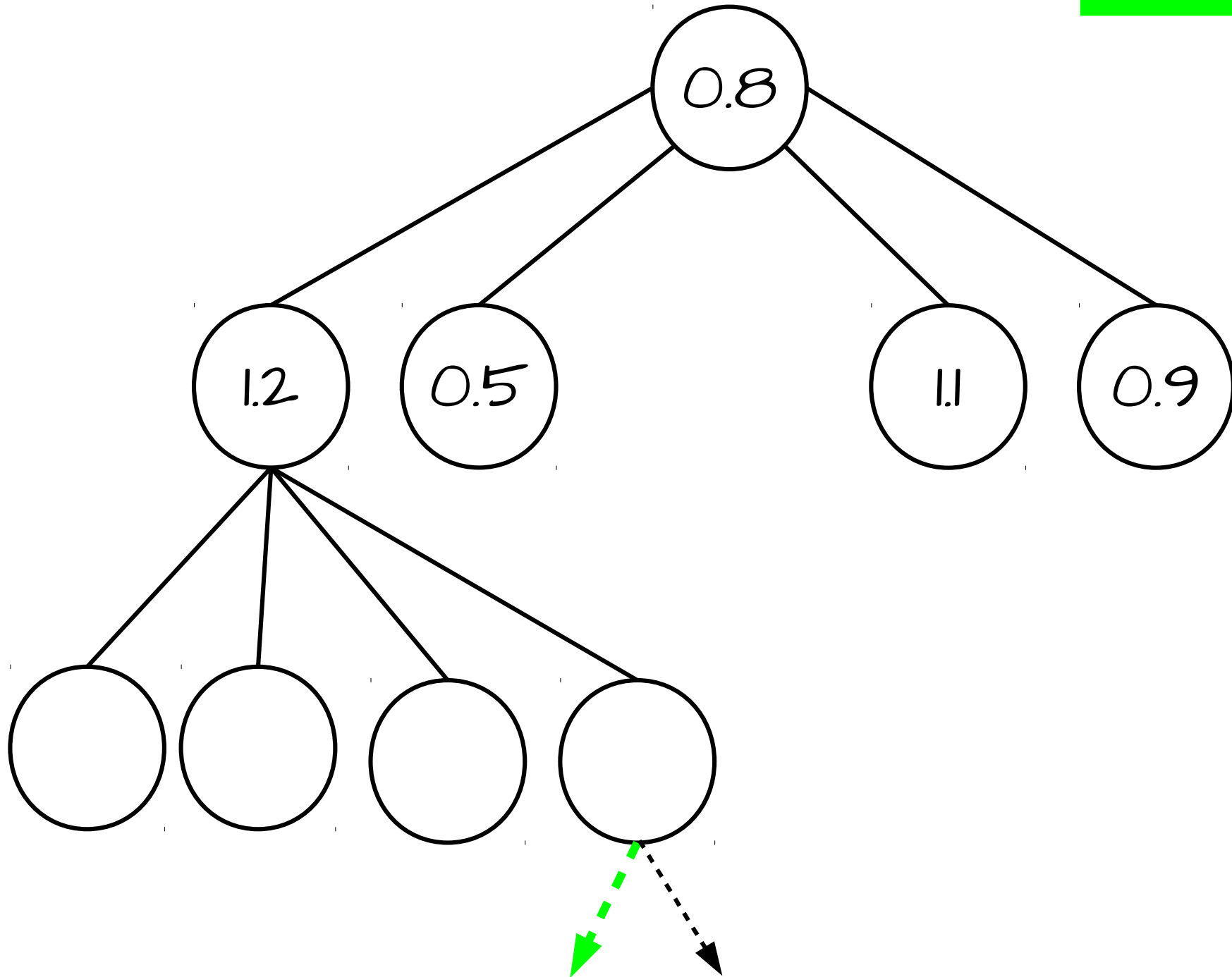
# Evaluation



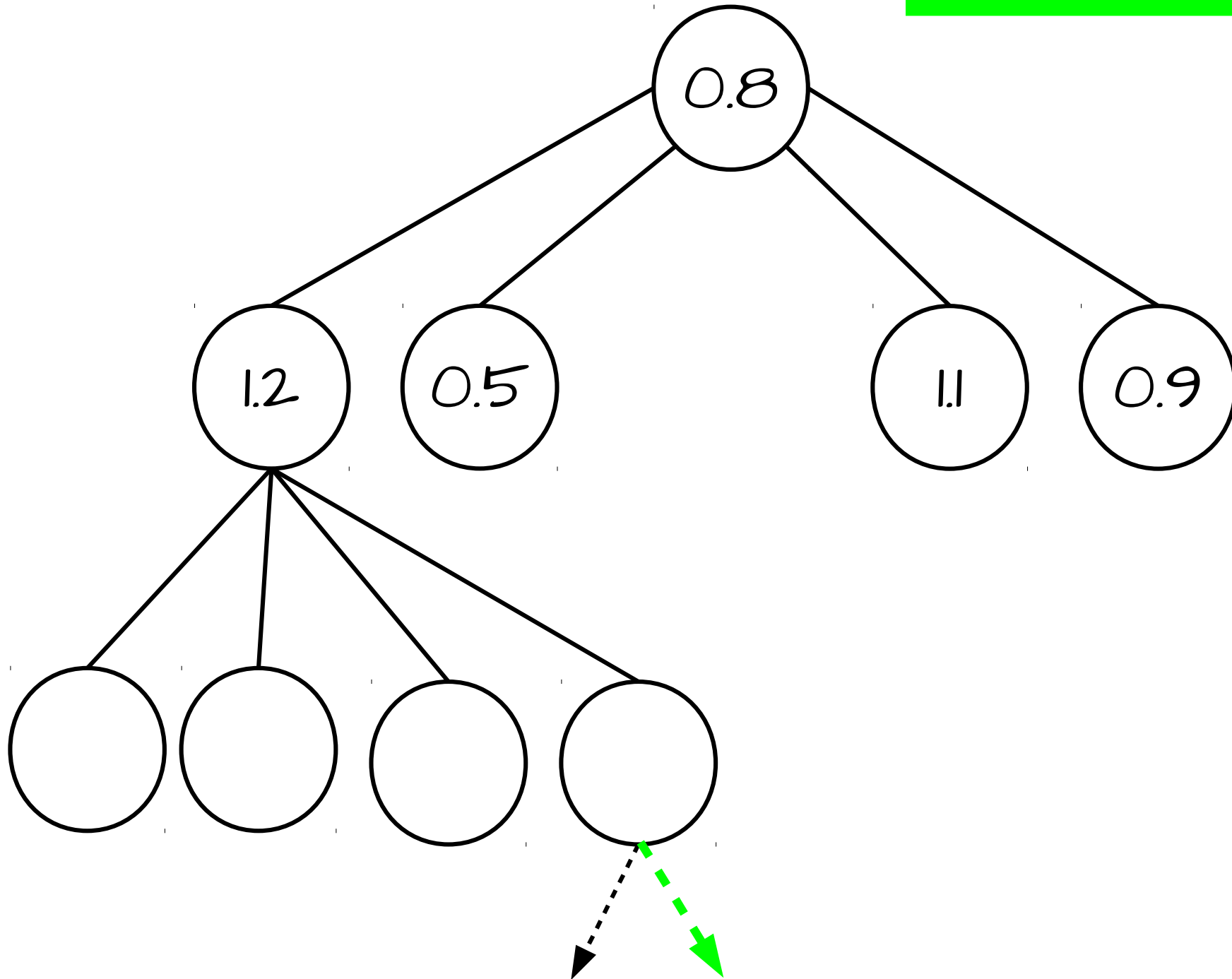
# Evaluation



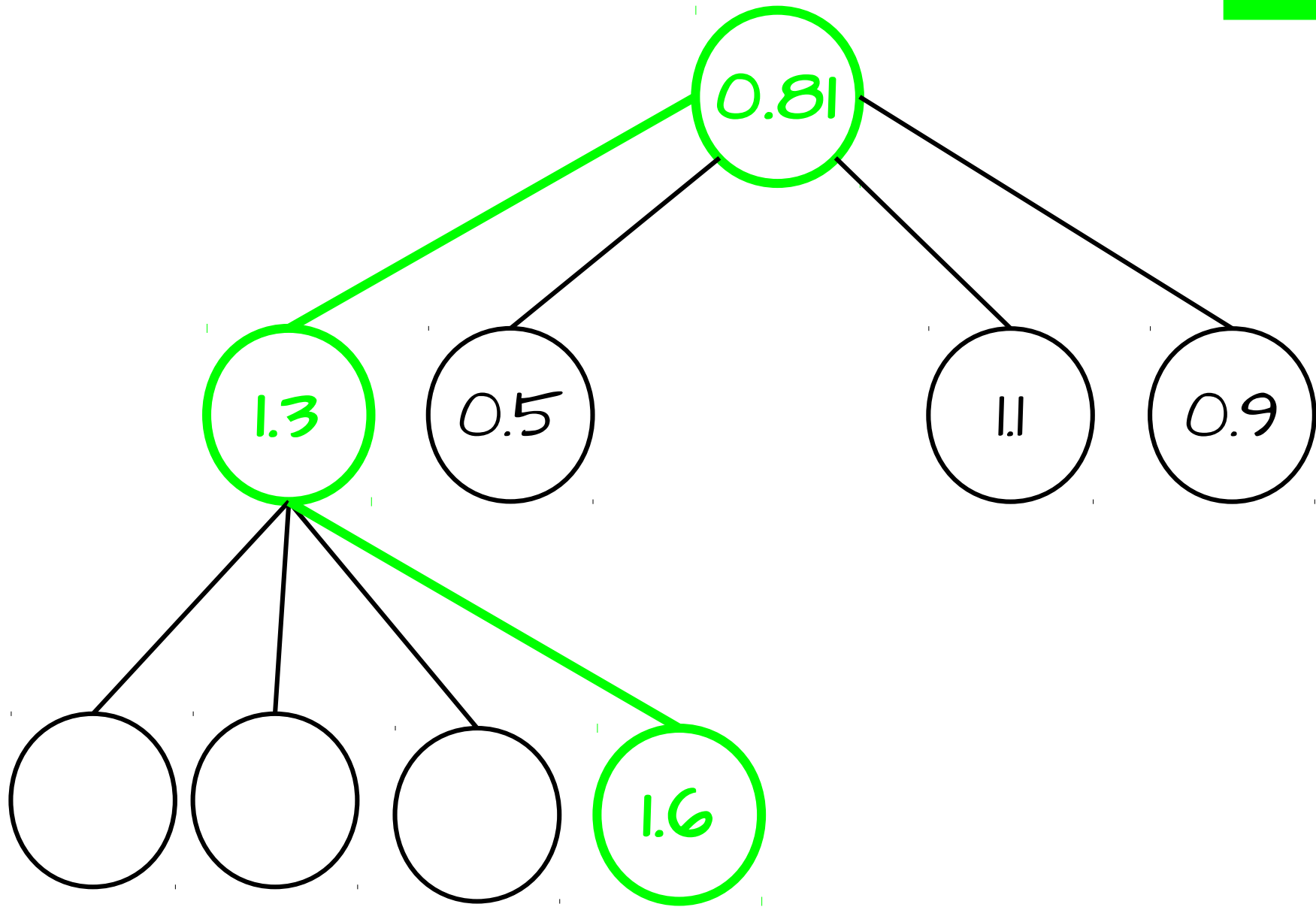
Rollout



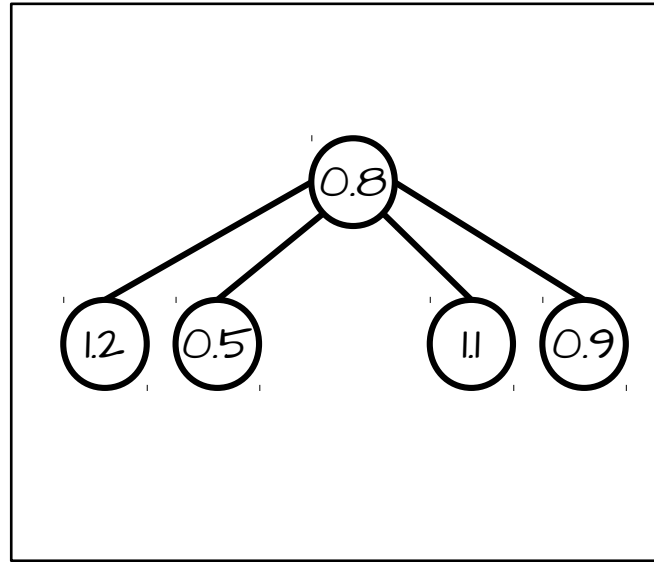
# Value Network



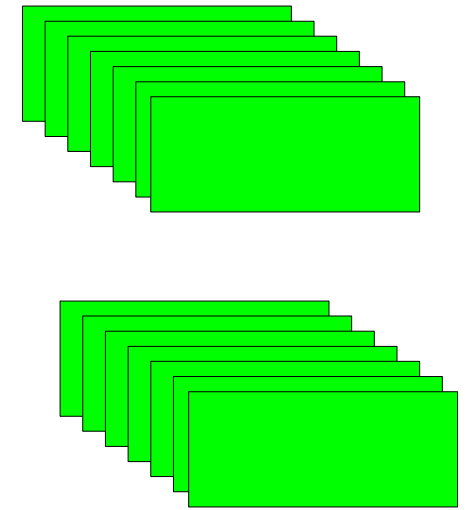
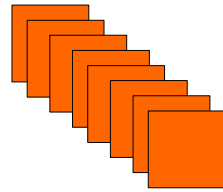
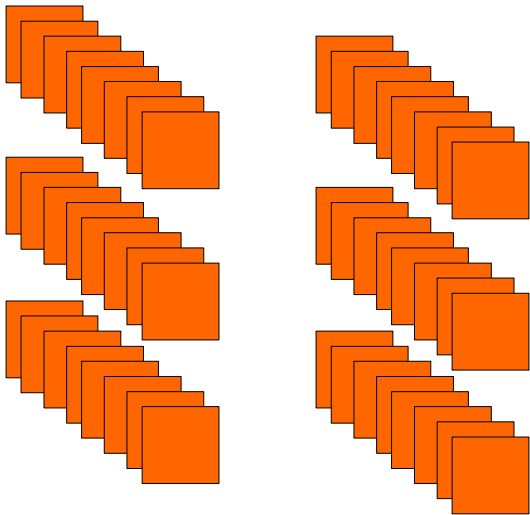
# Backup



1202 CPUs



176 GPUS



Tensor



## 3 Strengths of AlphaGo

Human Instinct

Reading Capability

Positional Judgement

Policy Network

Search

Value Network

# Most Important Strength

Human Instinct

Policy Network

Reading Capability

Search

Positional Judgement Value Network

More Natural



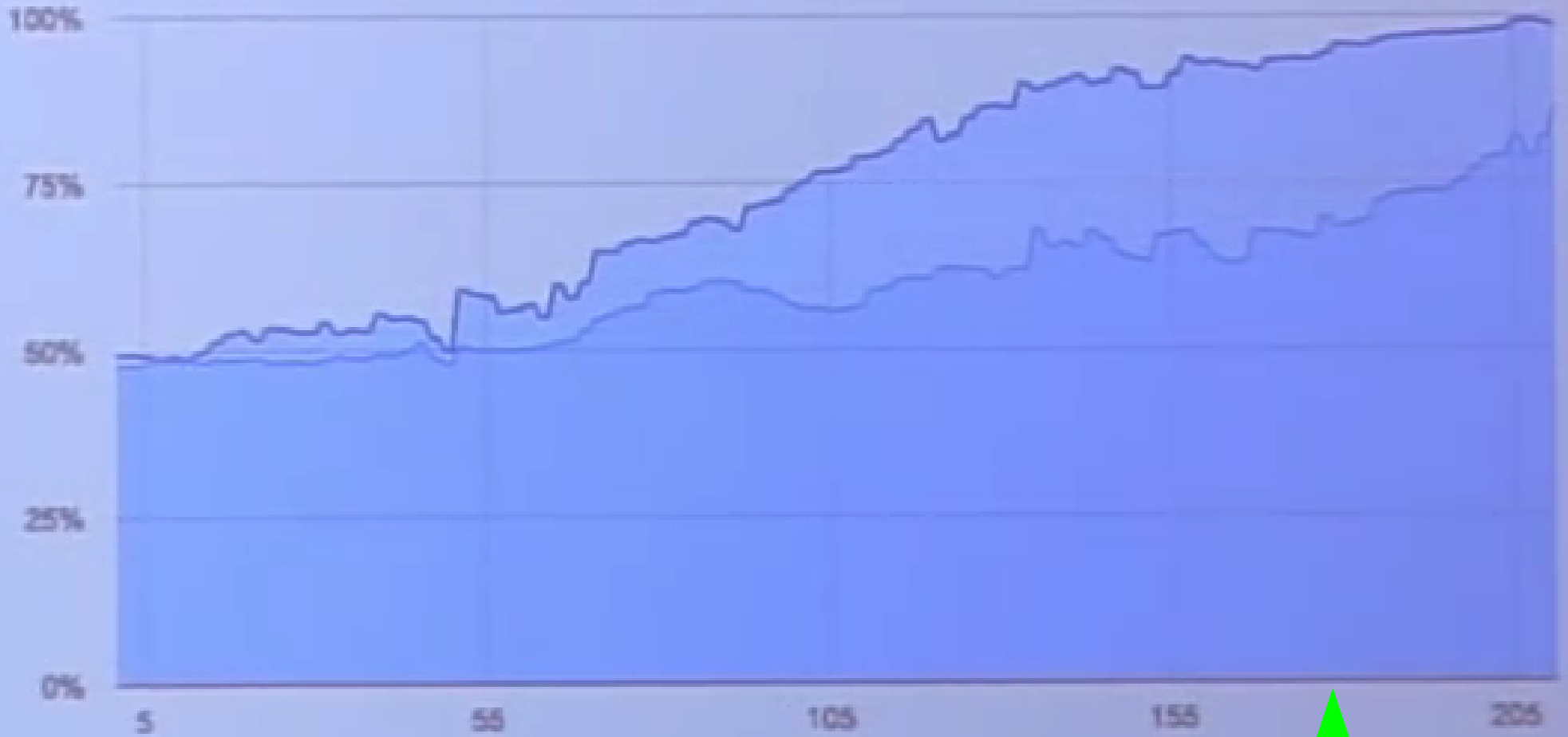
# Lee Sedol match



Style



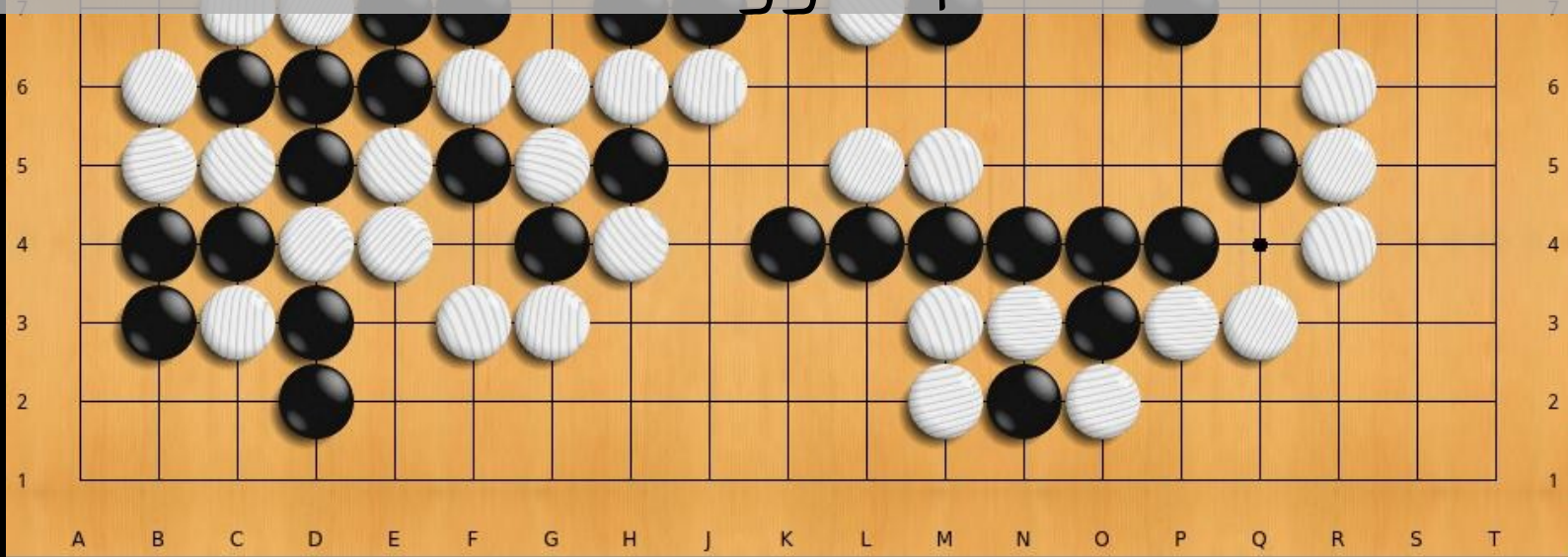
# Game 2





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

```

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 . . . 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

```

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

```

  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 . 0 . 5
4 . X X . . . . . . 4
3 . . X . . . X . . 3
2 . . . X . . . X . 2
1 . . . . . . . . . 1

```

white played at F5

black's turn to move!

Rubykon is thinking...

```

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           = ['pragtobg@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
14  gem and benchmark/avg. Works on all major ruby versions.}
15  spec.homepage        = 'https://github.com/PragTob/rubykon'
16  spec.license         = 'MIT'
17
18  spec.files           = `git ls-files -z`.split("\n0").reject { |f| f.match(%r{^(test|spec|features)/}) }
19  spec.bindir          = 'exe'
20  spec.executables     = spec.files.grep(%r{^exe/}) { |f| File.basename(f) }
21  spec.require_paths   = ['lib']
22 end

```

# PragTob/Rubykon

Outer ignore rules  
idea

## Minimalistic Go MCTS Engine

117 commits

2 branches

0 releases

5 contributors



Branch: 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://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](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.#/media/File:Alphabet\\_Inc\\_Logo\\_2015.svg](https://en.wikipedia.org/wiki/Alphabet_Inc.#/media/File:Alphabet_Inc_Logo_2015.svg)
- CC BY 3.0
  - [https://en.wikipedia.org/wiki/File:Pi\\_30K.gif](https://en.wikipedia.org/wiki/File:Pi_30K.gif)