Deep Reinforcement Learning
Applications + **Hacking**

Arjun Chandra
Research Scientist
Telenor Research / Telenor-NTNU AI Lab
arjun.chandra@telenor.com
@boelger

21 November 2017

https://join.slack.com/t/deep-rl-tutorial/signup

The Plan

Few words on applications (not exhaustive…)

Games
Board Games, Card Games, Video Games, VR, AR, TV Shows (IBM Watson)

Robotics
Thermal Soaring, Robots, Self-driving *, Autonomous Braking, etc.

Embedded Systems
Memory Control, HVAC, etc.

Internet/Marketing
Personalised Web Services, Customer Lifetime

Energy
Solar Panel Control, Data Centres

Cloud/Telecommunications
Scaling, Resource Provisioning, Channel Allocation, Self-organisation in Virtual Networks

Health
Treatment Planning (Diabetes, Epilepsy, Parkinson's, etc.)

Maritime
Decision Support

… growing list

Hack!

Backgammon

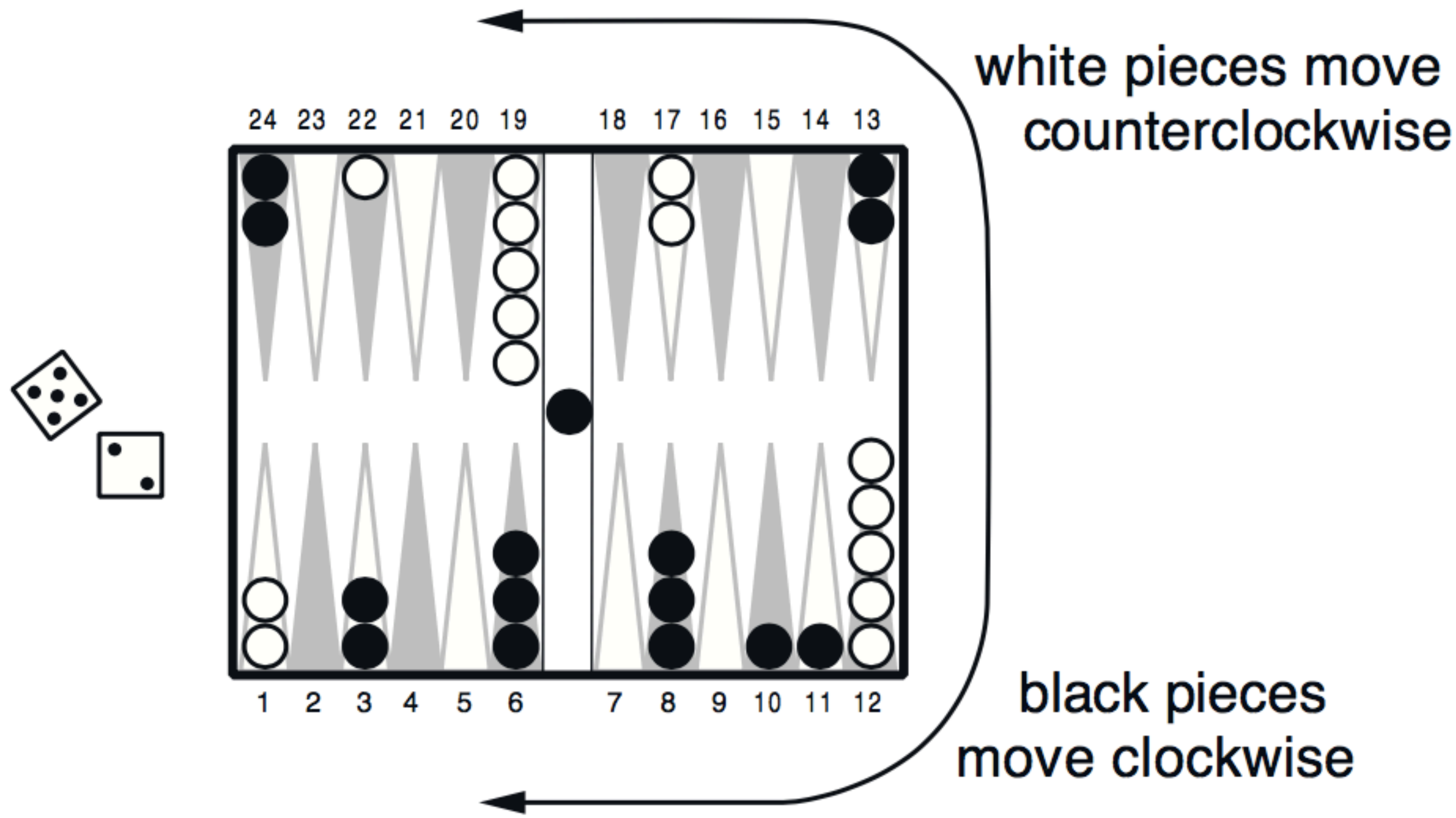
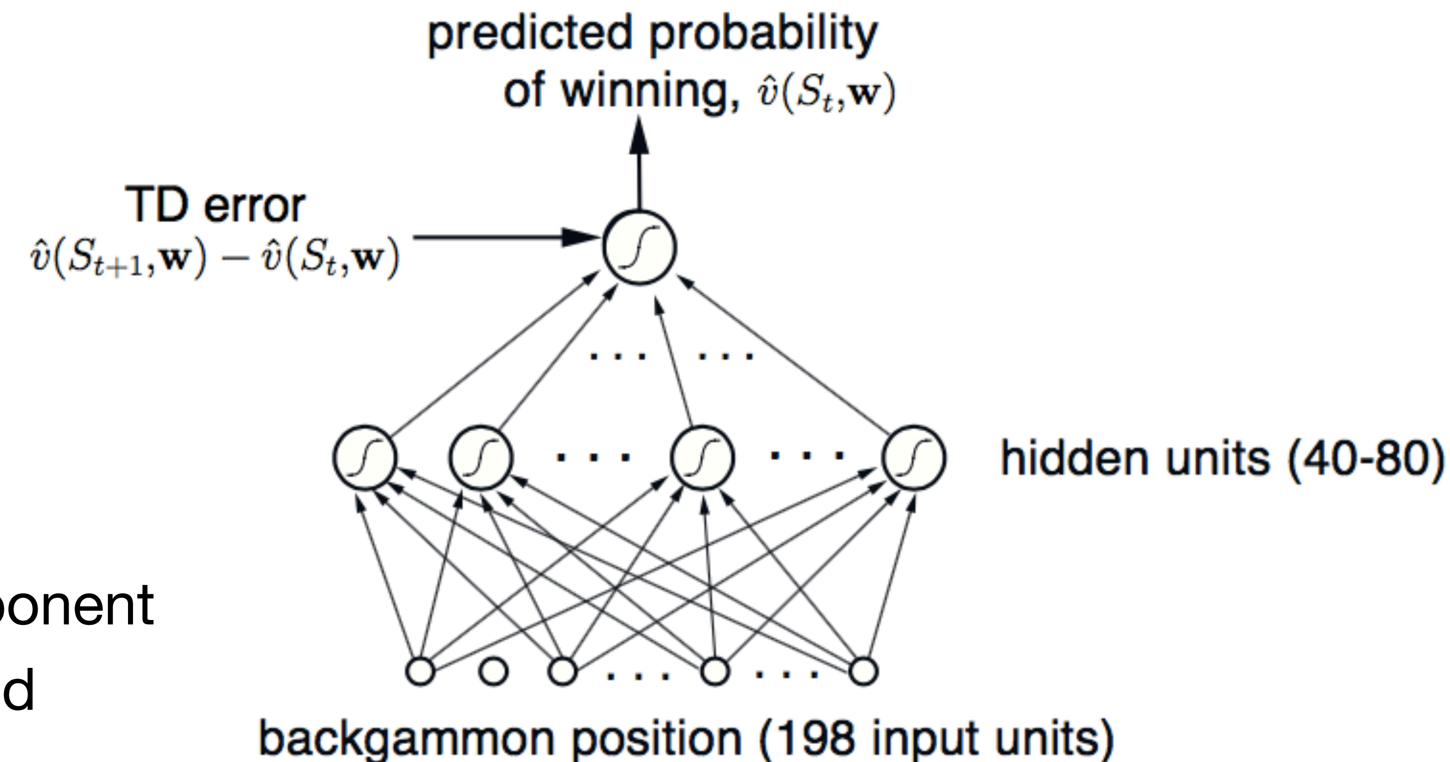


Image credit: http://incompleteideas.net/sutton/book/the-book-2nd.html

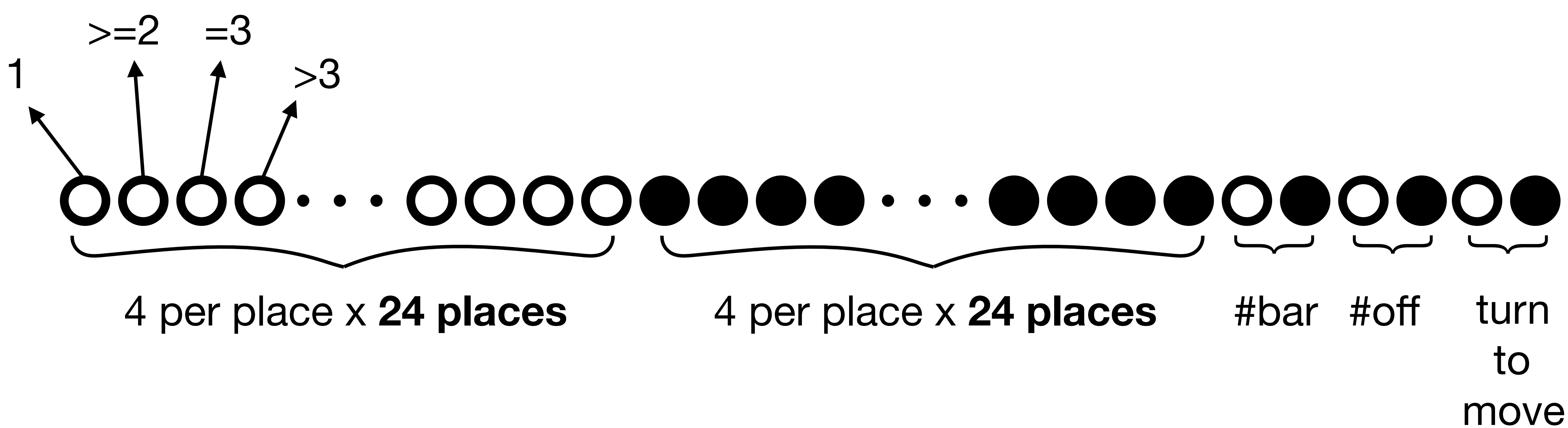


1: piece can be hit by opponent

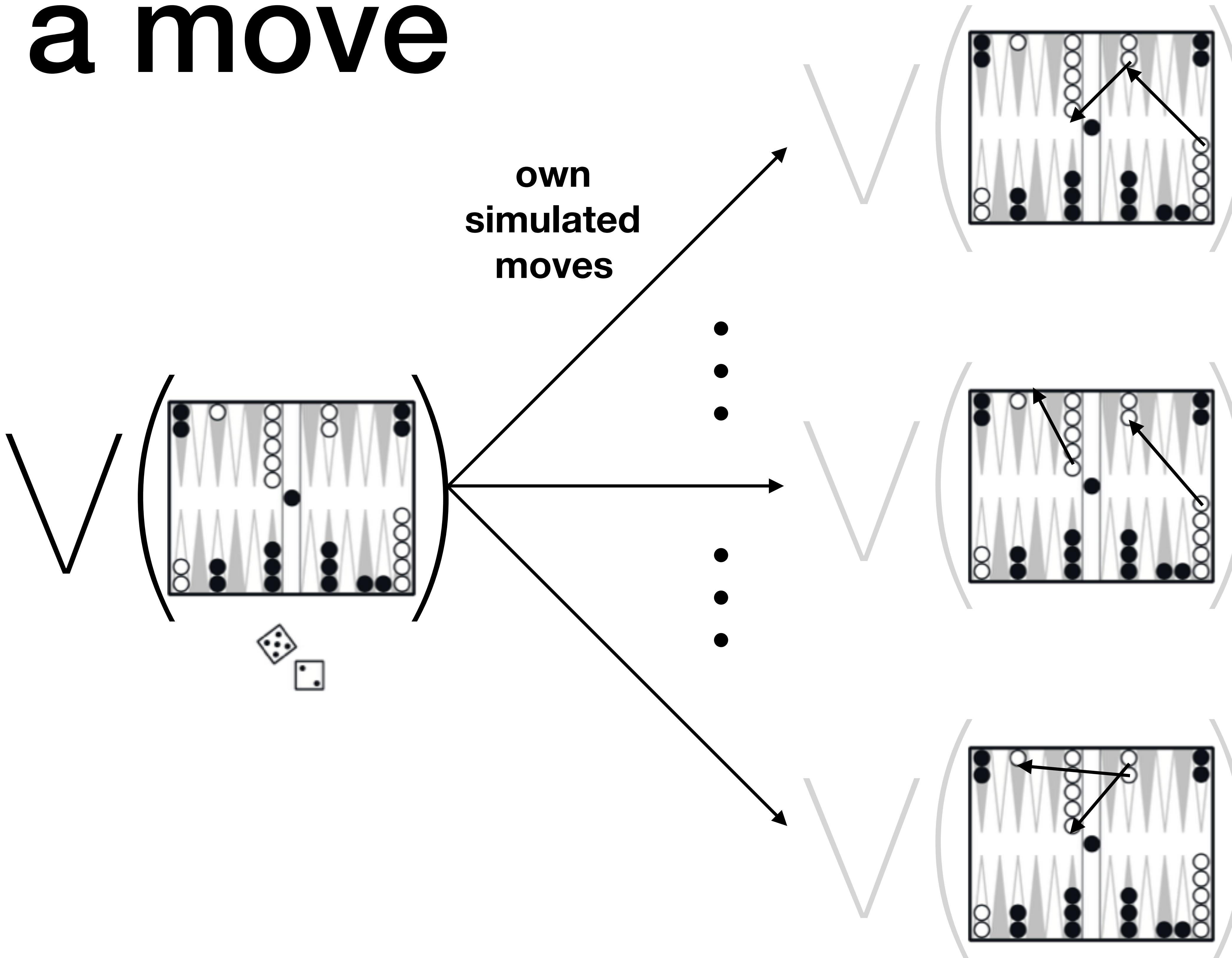
>=2: opponent cannot land

=3: single spare/free to move

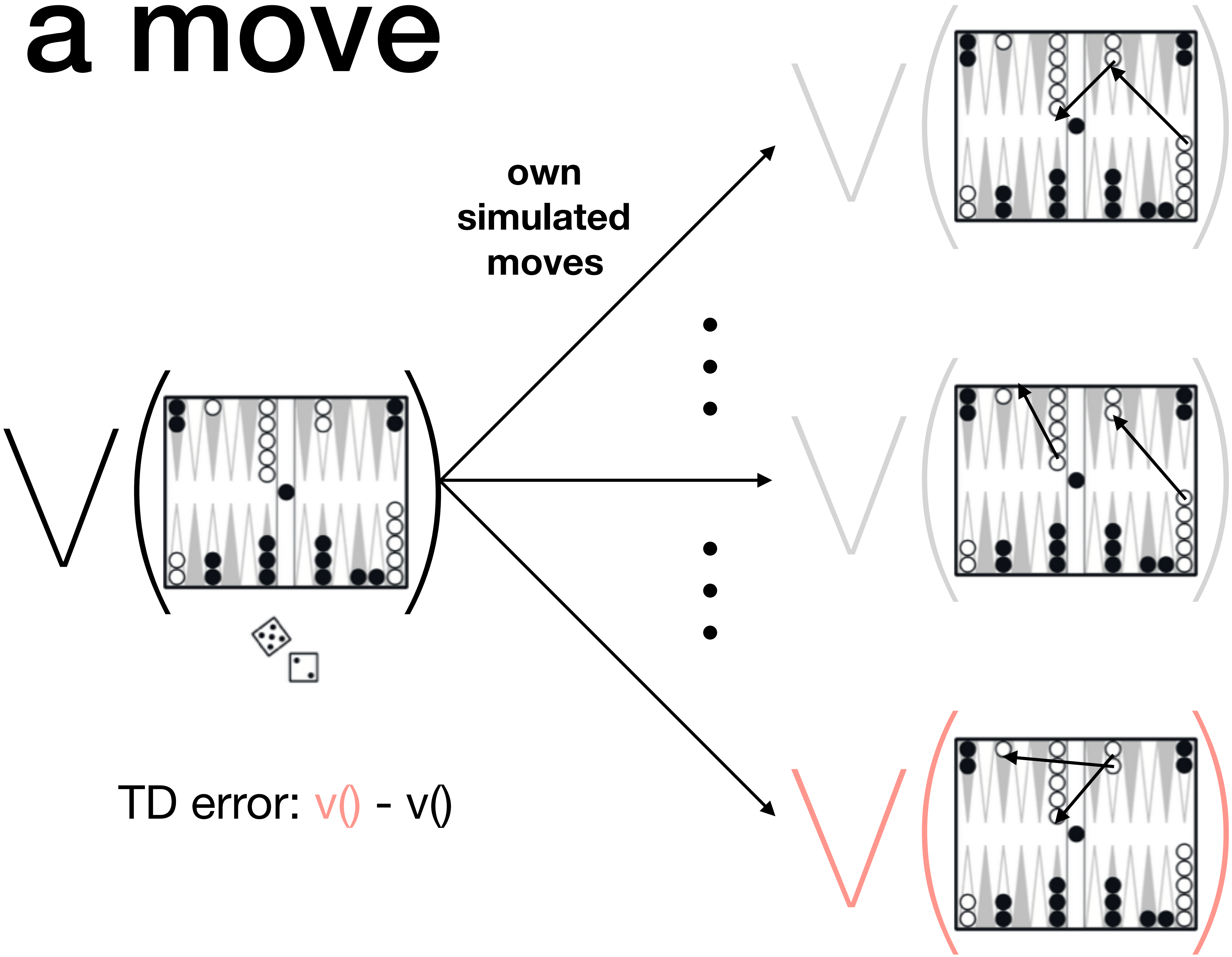
>3: multiple spare pieces!



a move



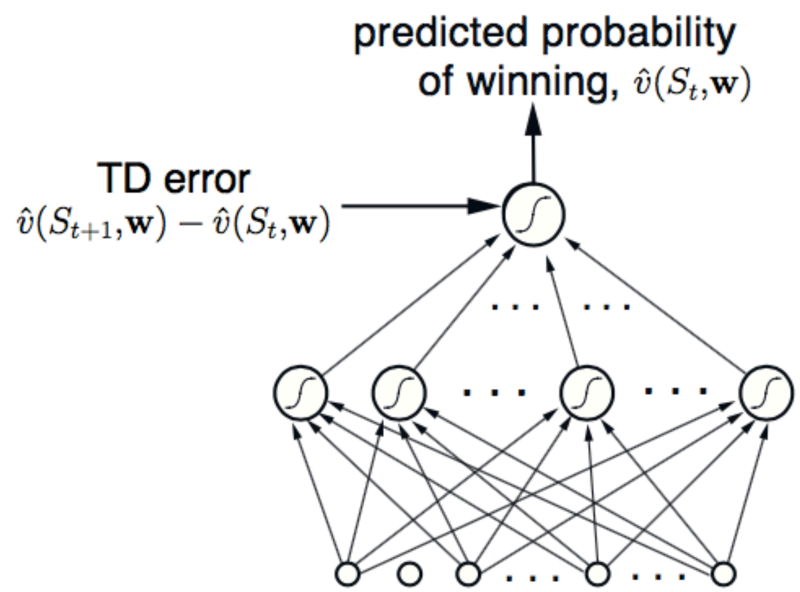
a move

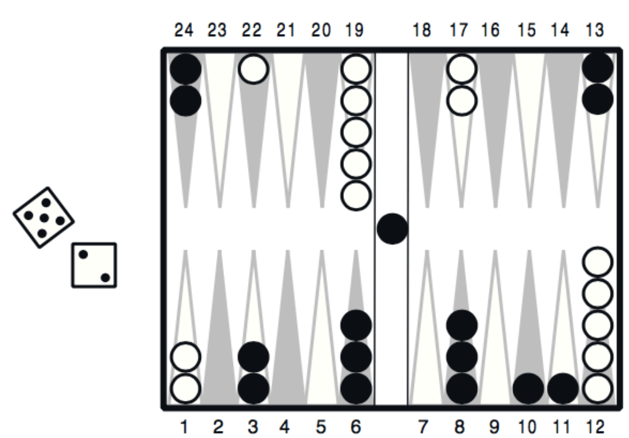


TD error: v() - v()

play to the end…

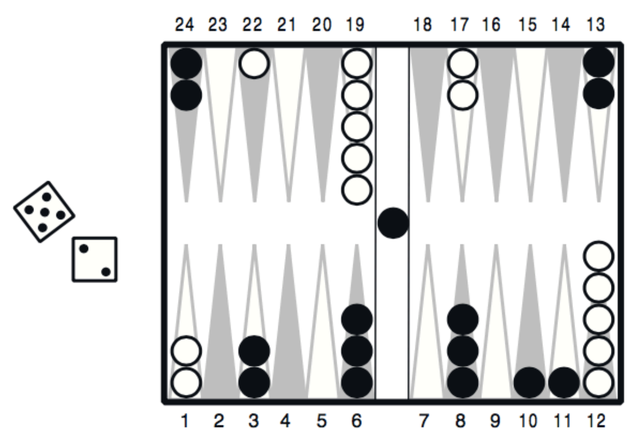
TD-Gammon 0.0

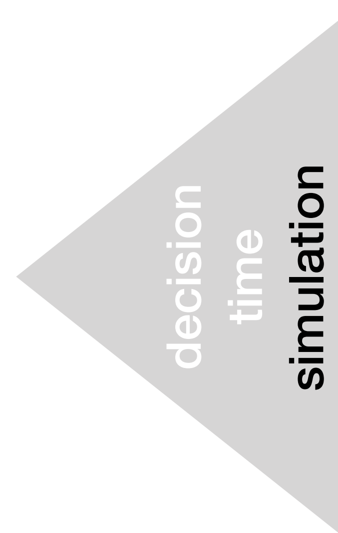




- **No Backgammon knowledge**
- NN, Backprop to represent and learn
- Self-play TD to estimate returns
- Good player beating programs with expert training and hand crafted features

TD-Gammon >1.0++





v() of simulated next moves
inform
v() of move to play

Simulation:

-> own move given dice roll

-> opponent dice roll

-> opponent move

Assume opponent choses best value move.

Best move given opponent's best move is selected.

- **Specialised Backgammon features**
- NN, Backprop to represent and learn
- Self-play TD and decision time search, to estimate returns
- World class — **impacted human play**

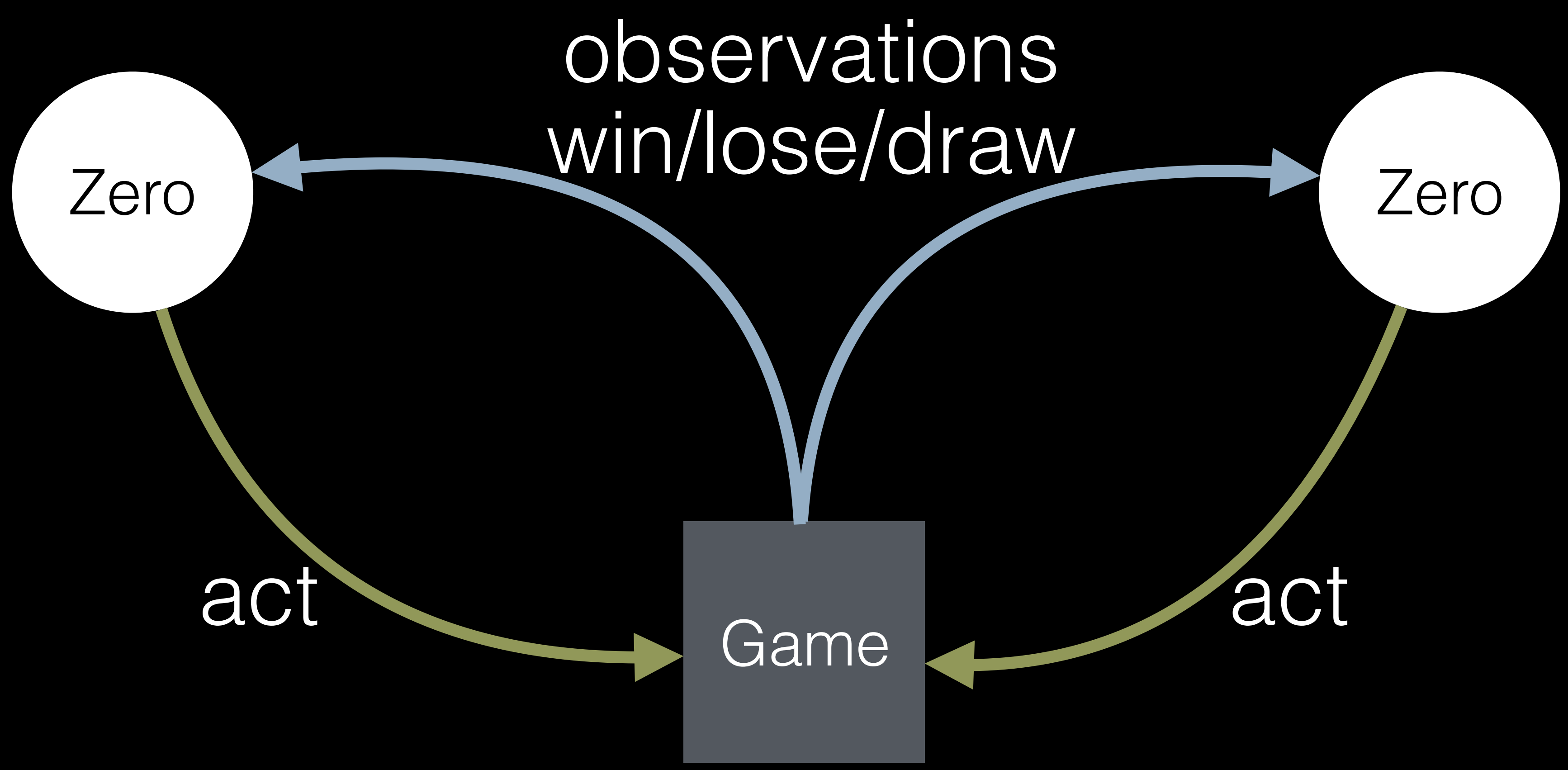
1992, 1994, 1995, 2002…
NB. impacted human play, raised human caliber

Program	Hidden Units	Training Games	Opponents	Results
TD-Gammon 0.0	40	300,000	other programs	tied for best
TD-Gammon 1.0	80	300,000	Robertie, Magriel, ...	−13 pts / 51 games
TD-Gammon 2.0	40	800,000	various Grandmasters	−7 pts / 38 games
TD-Gammon 2.1	80	1,500,000	Robertie	−1 pt / 40 games
TD-Gammon 3.0	80	1,500,000	Kazaros	+6 pts / 20 games

Combination of **learnt value function** and **decision time search** powerful!

Deep RL in **AlphaGo Zero**

Improve
planning (search) and **intuition (evaluation)**
with **feedback from self-play**
[**zero** human game data]



Mastering the game of Go without human knowledge, Silver et.al., Nature, Vol. 550, **October 19, 2017**

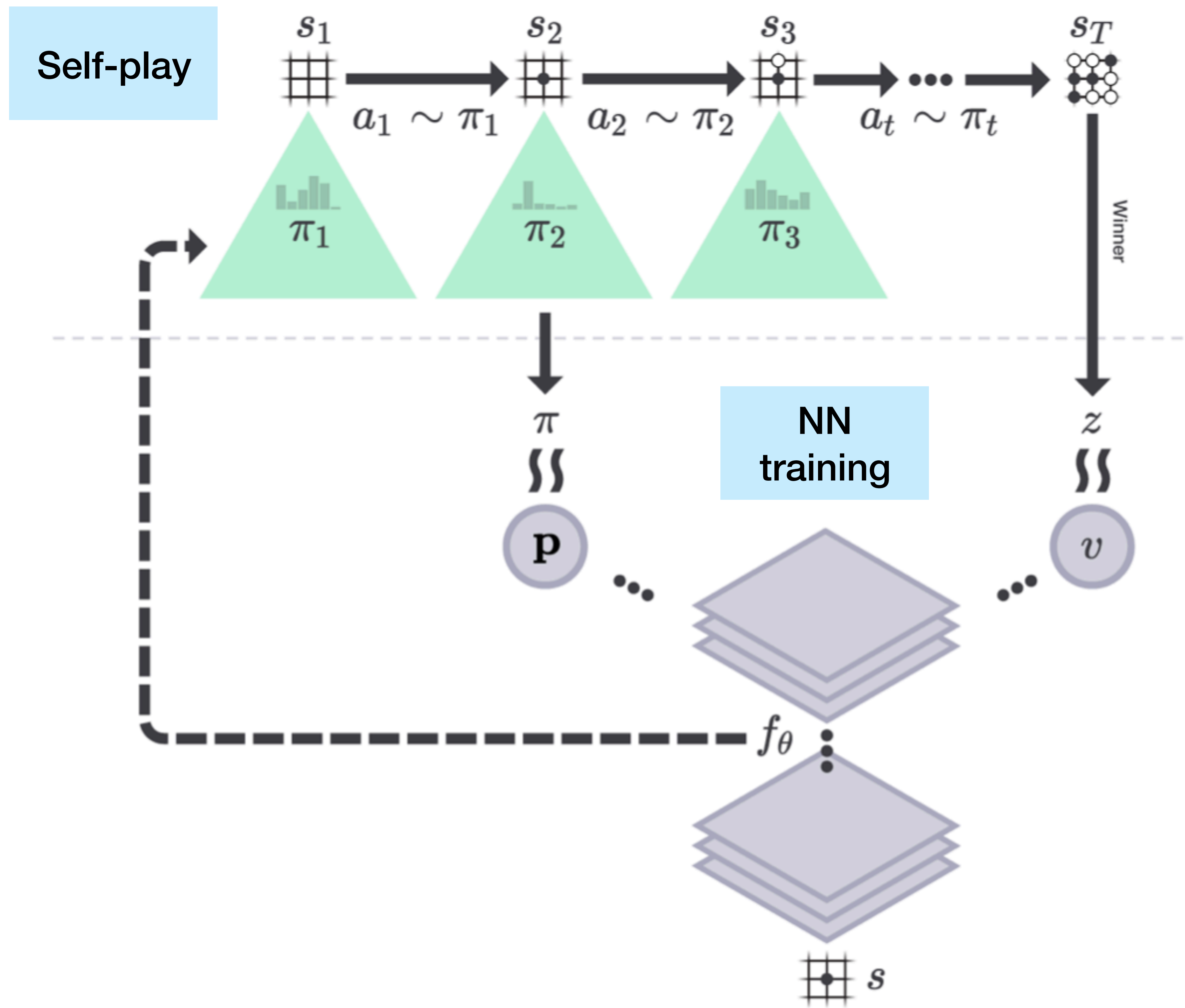
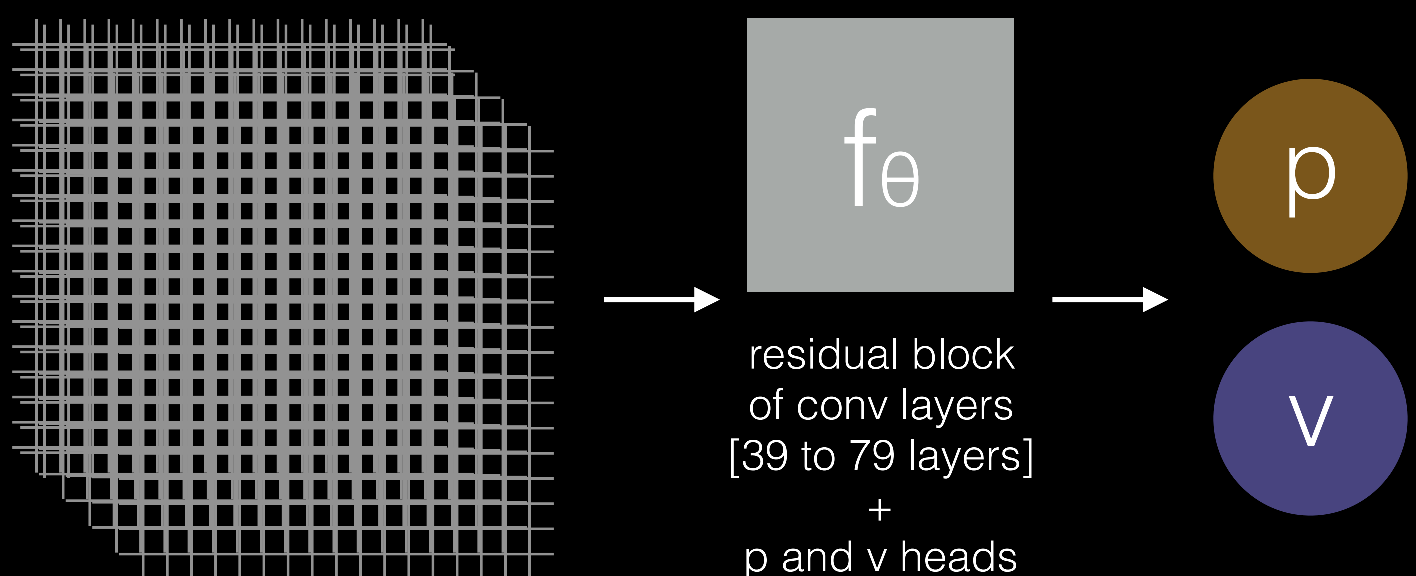


Image credit: http://incompleteideas.net/sutton/book/the-book-2nd.html

Deep Net



probability of taking one of 362 actions

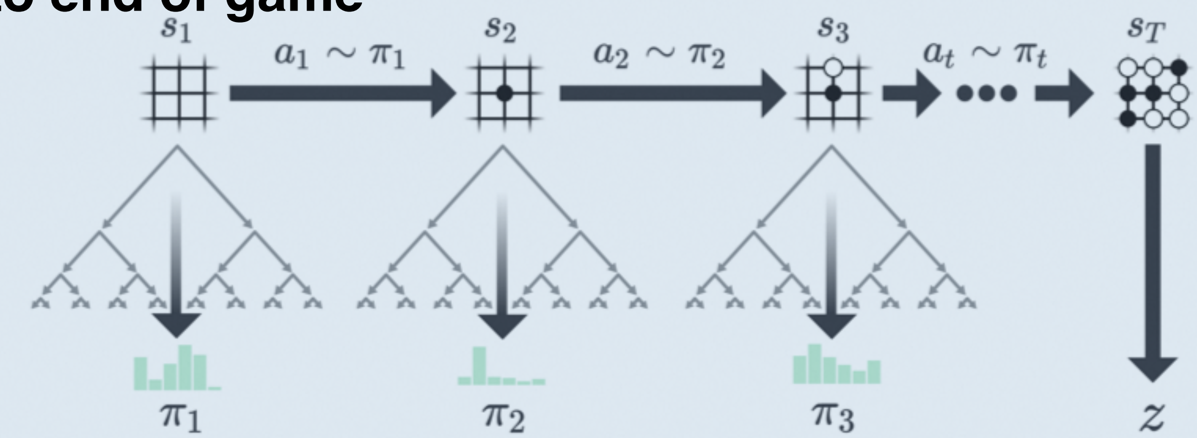
likelihood of win/loss

[Xt, Yt, Xt-1, Yt-1, ..., Xt-7, Yt-7, C]

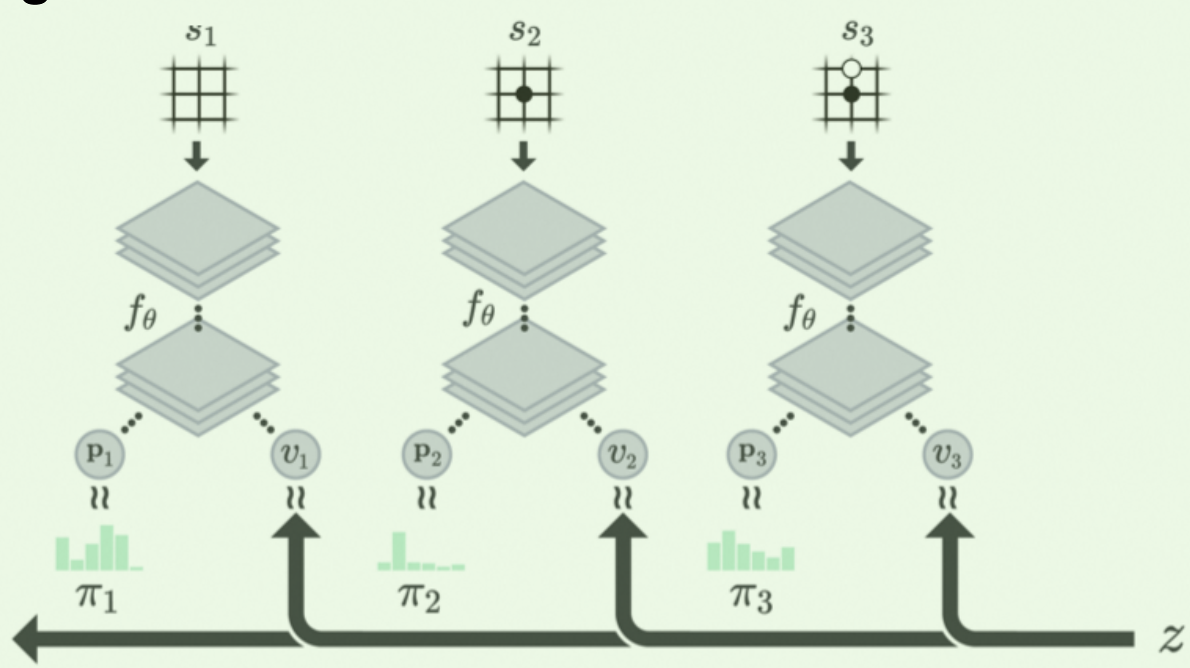
historical map of stones
X: 1/0 player stones
Y: 1/0 opponent stones
C: player, all 1 black, all 0 white

residual block
of conv layers
[39 to 79 layers]
+
p and v heads
[2 layers, 3 layers]

Self-play to end of game

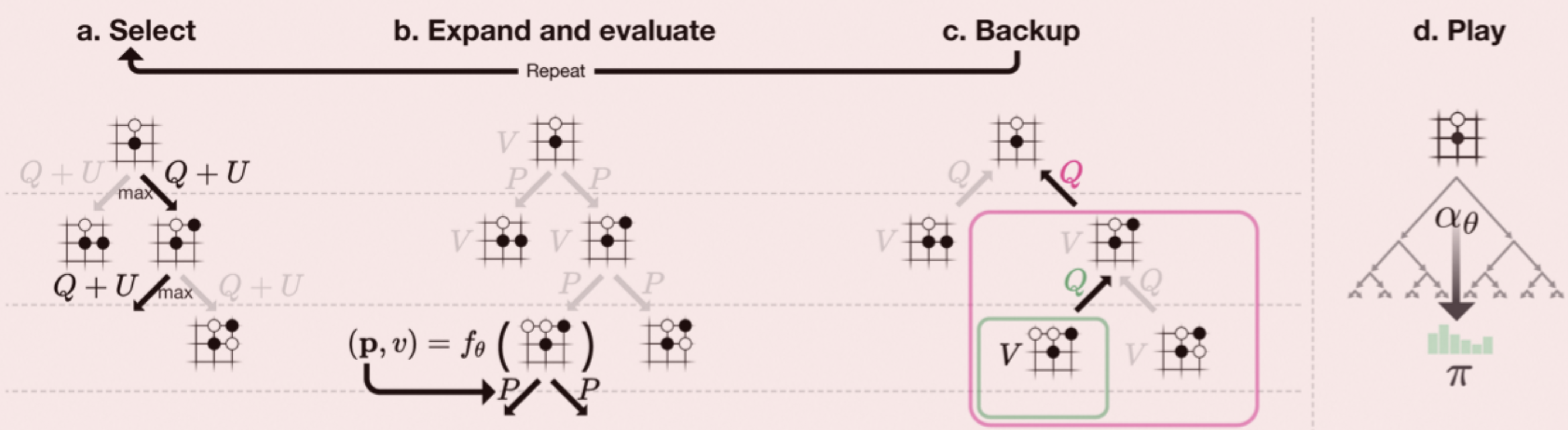


NN training: learn to evaluate



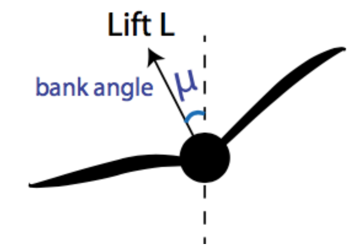
$$l = (z - v)^2 - \boldsymbol{\pi}^{\mathrm{T}} \log \boldsymbol{p} + c\|\theta\|^2$$

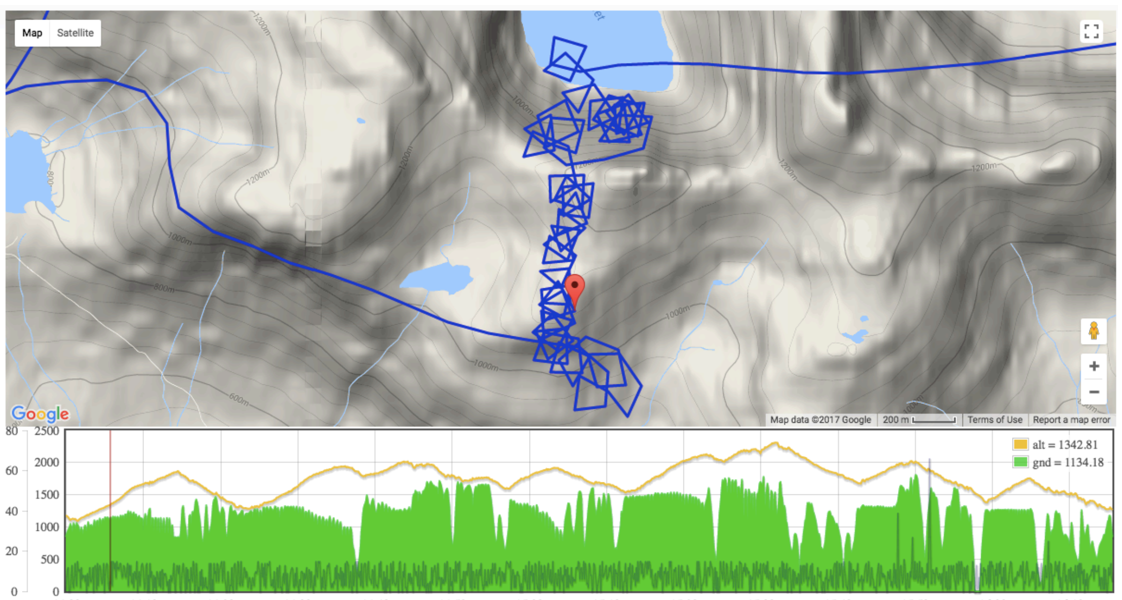
Self-play step: select move by simulation + evaluation

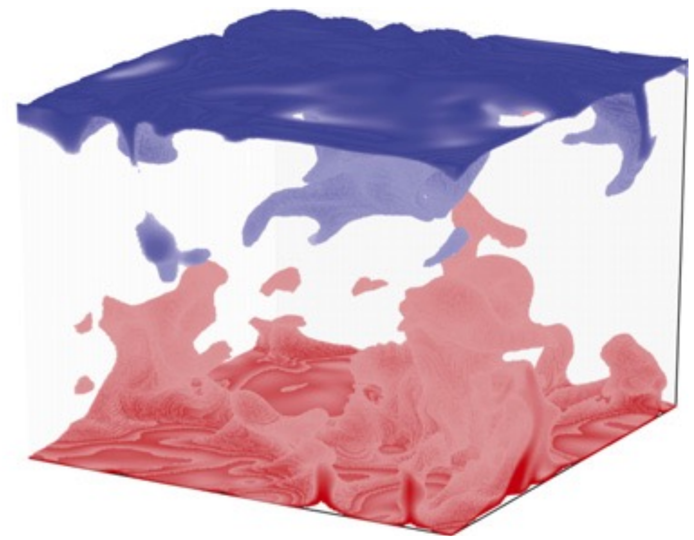


Mastering the game of Go without human knowledge, Silver et.al., Nature, Vol. 550, **October 19, 2017**

Thermal Soaring

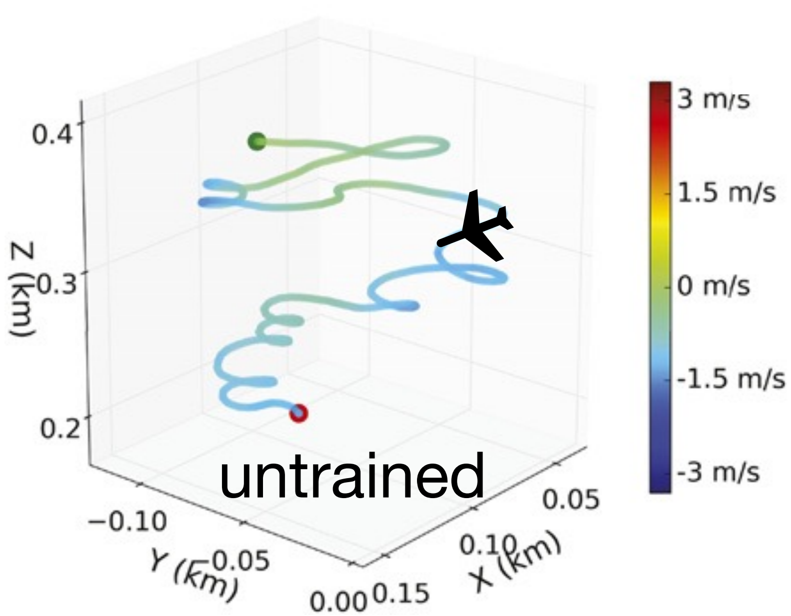


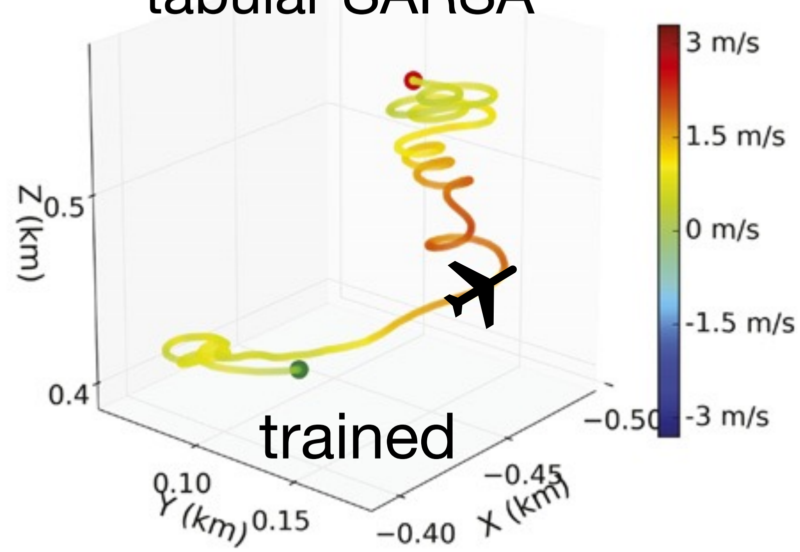

https://www.onlinecontest.org/olc-2.0/gliding/flightinfo.html?flightId=1631541895

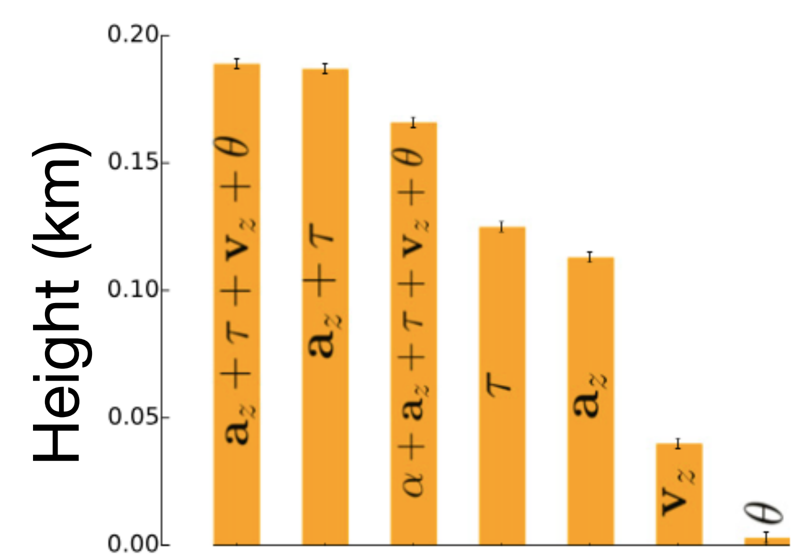

simulation

state: (local, descritised)
acceleration (a_z),
torque,
velocity (v_z),
temperature
action: bank +/-, no-op
reward: after step v_z **+** Ca_z
goal: climb to cloud ceiling

tabular SARSA



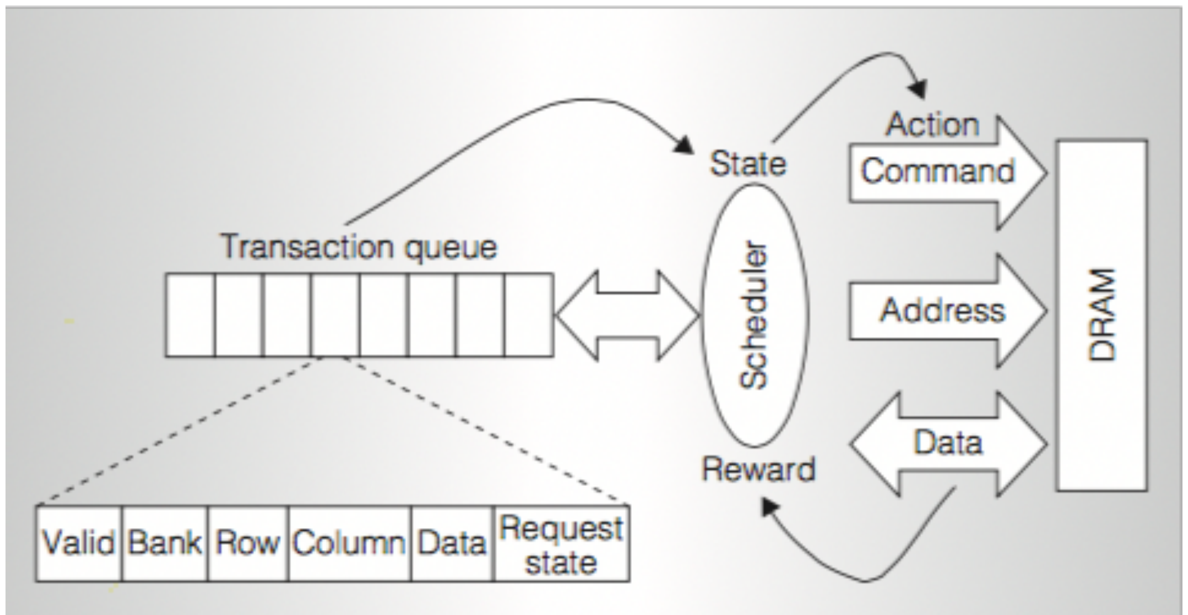




Learning to soar in turbulent environments, Gautam Reddy et. al., PNAS 2016

Memory Control

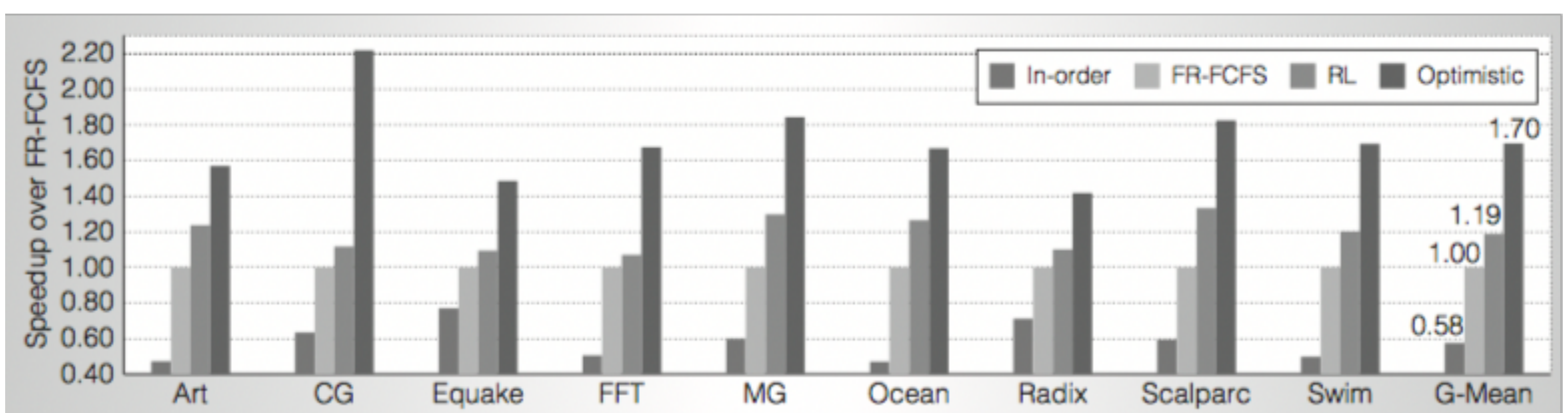
scheduler is the agent



http://incompleteideas.net/sutton/book/the-book-2nd.html

state: based on contents of transaction queue, e.g. #read requests, #write requests, etc.
action: *activate*, *precharge*, *read*, *write*, *no-op*
reward: 1 for read or write, 0 otherwise
goal: (max read/write ~ throughput)
constraints on valid actions/state

H/W implementation of SARSA

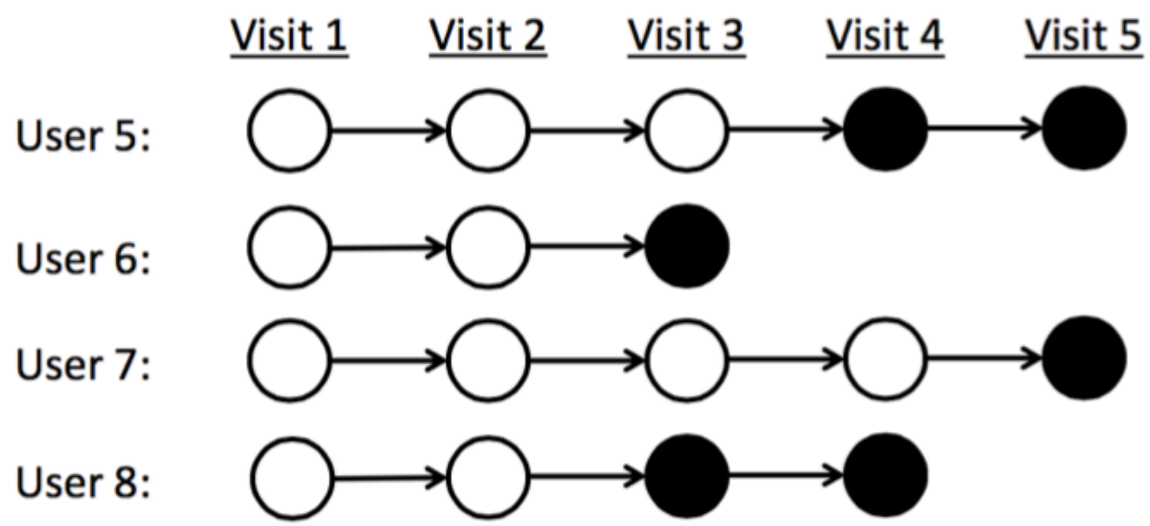


http://incompleteideas.net/sutton/book/the-book-2nd.html

Dynamic multicore resource management: A machine learning approach
Martinez and Ipek, IEEE Micro, 2009

Personalised Services

(content/ads/offers)



http://incompleteideas.net/sutton/book/the-book-2nd.html

$$\frac{\#clicks}{\#visits} \qquad CTR = \frac{6}{17} \approx 0.35$$

$$LTV = \frac{6}{4} = 1.5 \qquad \frac{\#clicks}{\#visitors}$$

goal

policy encouraging users to engage in extended interactions

state: (per customer)
time since last visit,
total visits,
last time clicked,
location,
interests,
demographics
action: offers/ads
reward: 1 click, 0 otherwise

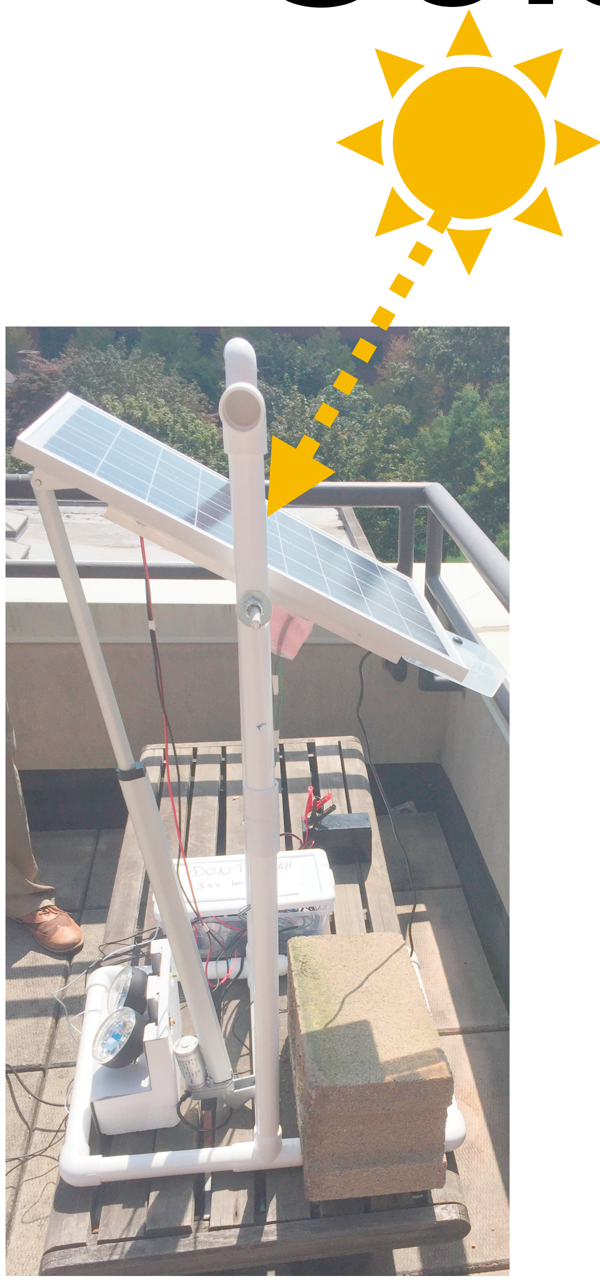
(s,a,r,s')
tuples from the past policies

sampled tuples and train random forest to **predict return** (fitted Q iteration ~ DQN)

Adobe® Marketing Cloud

Personalized Ad Recommendation Systems for Life-Time Value Optimization with Guarantees. Theocharous et. al. IJCAI, 2015

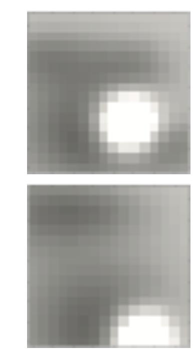
Solar Panel Control



Solar tracking — pointing at sun enough?
Missing:

- diffused radiation
- reflected — ground/snow/surroundings
- power consumed to reorient
- shadows — foliage, clouds etc.

state: panel orientation, relative location of sun
OR downsampled 16X16 image
actions: set of discrete orientations
OR *tilt forward/back/no-op*
reward: energy gathered at time step
goal: maximise energy gathered over time



Bandit-Based Solar Panel Control
David Abel et. al. IAAI 2018

Improving Solar Panel Efficiency using Reinforcement Learning. David Abel et. al. RLDM 2017

Approach	Total Energy Gathered (J)
`lin-ucb`	77103.19
`sarsa`	12219.55
`grena-tracker`	26600.33

https://github.com/david-abel/solar_panels_rl

Hack!

Code

Clone this repo:

https://github.com/traai/drl-tutorial

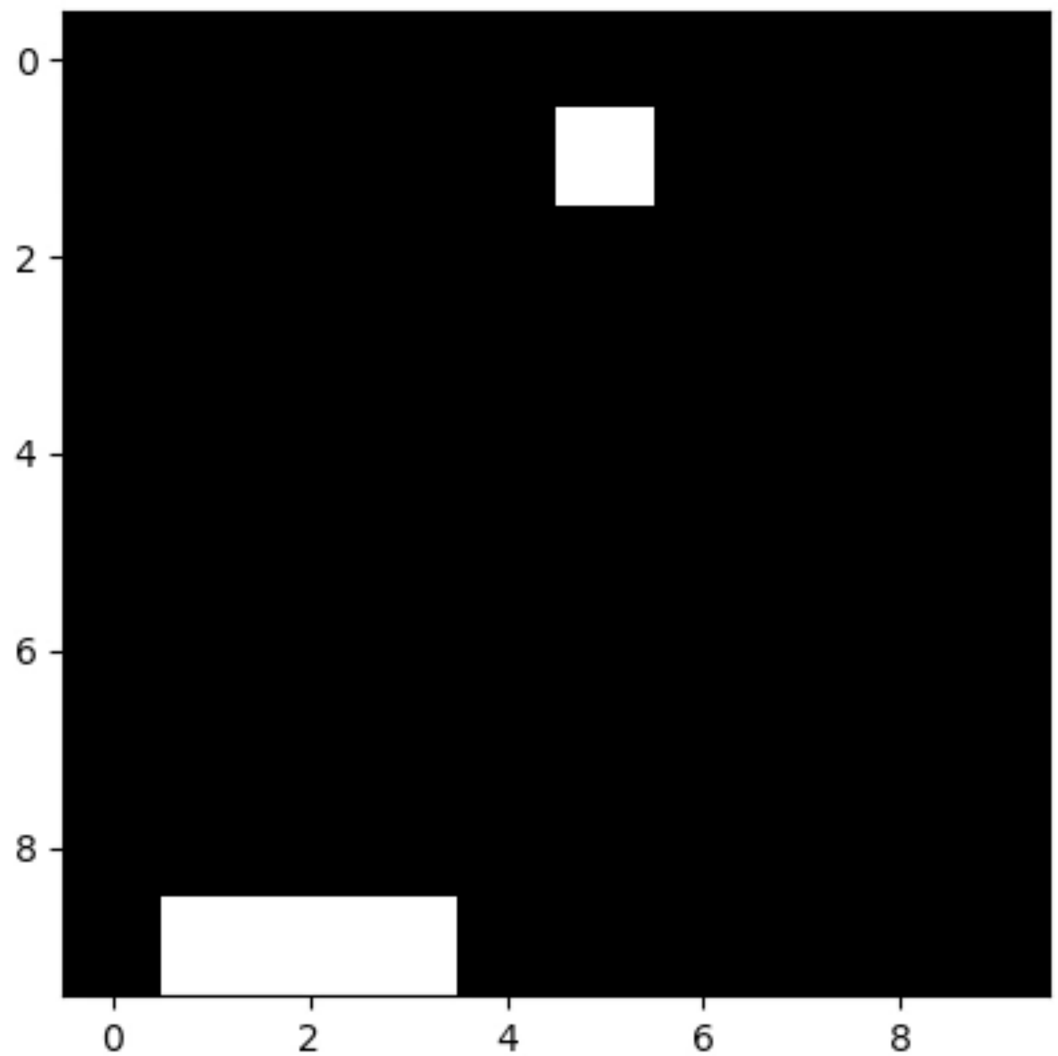
Go through **README** to set up Python environment and read through the tasks. Build on provided code/code from scratch.

Use Slack for questions:

https://join.slack.com/t/deep-rl-tutorial/signup

Value Based (DQN)

Catch fruit in basket!



state: 1 for fruit, 1s for basket

```
array([[ 0.,  0.,  0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  1.,  1.,  1.,  0.,  0.,  0.]])
```

actions: left, right, no-op

rewards

+1: fruit caught

-1: fruit not caught

0: otherwise

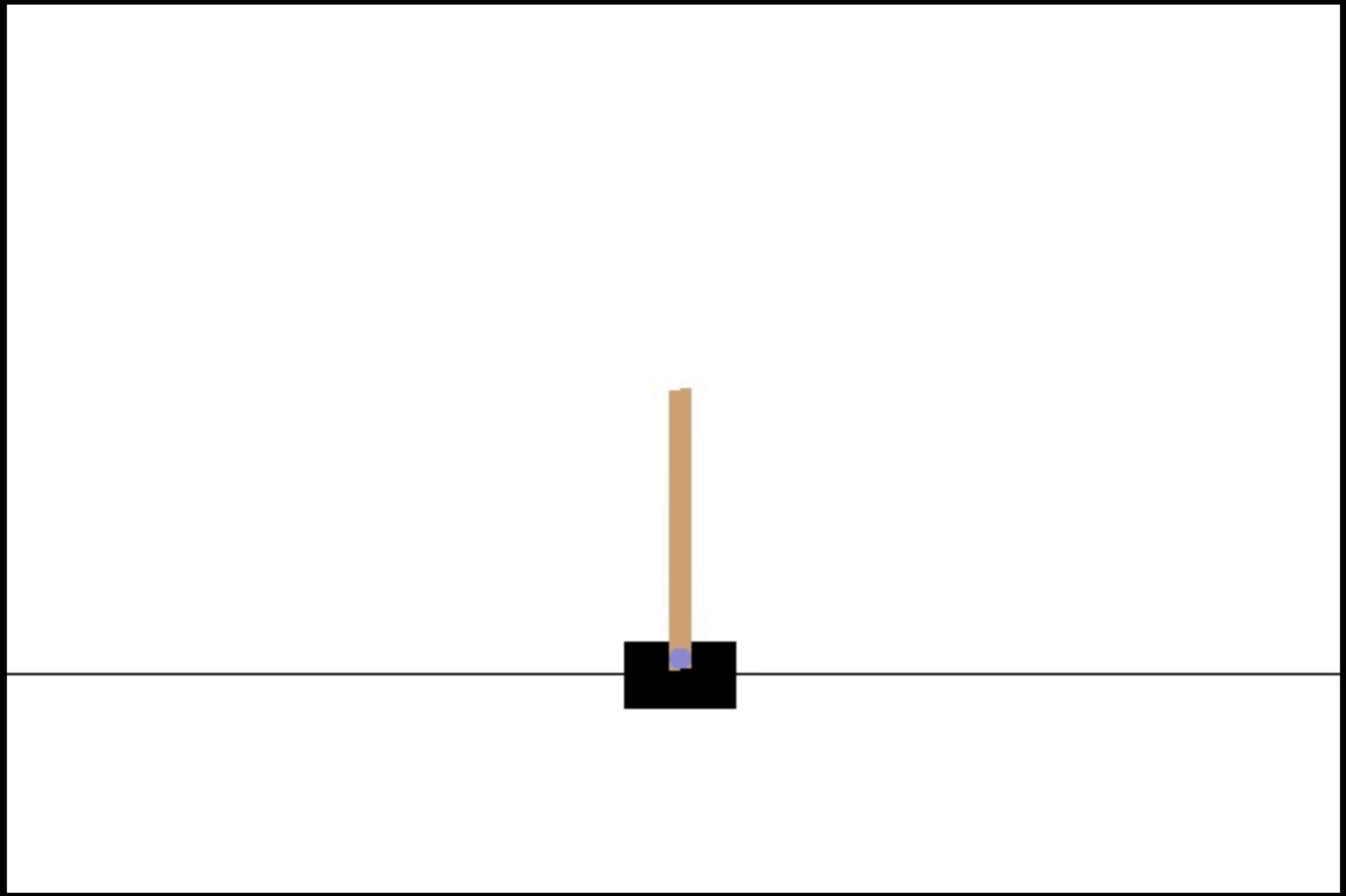
goal: catch fruit (!)

Simple DQN solution:

https://github.com/traai/drl-tutorial/blob/master/value/dqn.py

Policy Based

Balance a pole!



state

Num	Observation	Min	Max
0	Cart Position	-2.4	2.4
1	Cart Velocity	-Inf	Inf
2	Pole Angle	~ -41.8°	~ 41.8°
3	Pole Velocity At Tip	-Inf	Inf

action

Num	Action
0	Push cart to the left
1	Push cart to the right

reward: 1 for each step

goal: maximise cumulative reward

https://github.com/openai/gym/wiki/CartPole-v0

Simple PG solution:
https://github.com/traai/drl-tutorial/blob/master/pg/pg.py