Artificial Overmind through Deep Learning

Game is a good choice for simulation of non-trivial environment

What for

In many tasks of control, humans are not the best choice, for example, when decision should be made very fast.

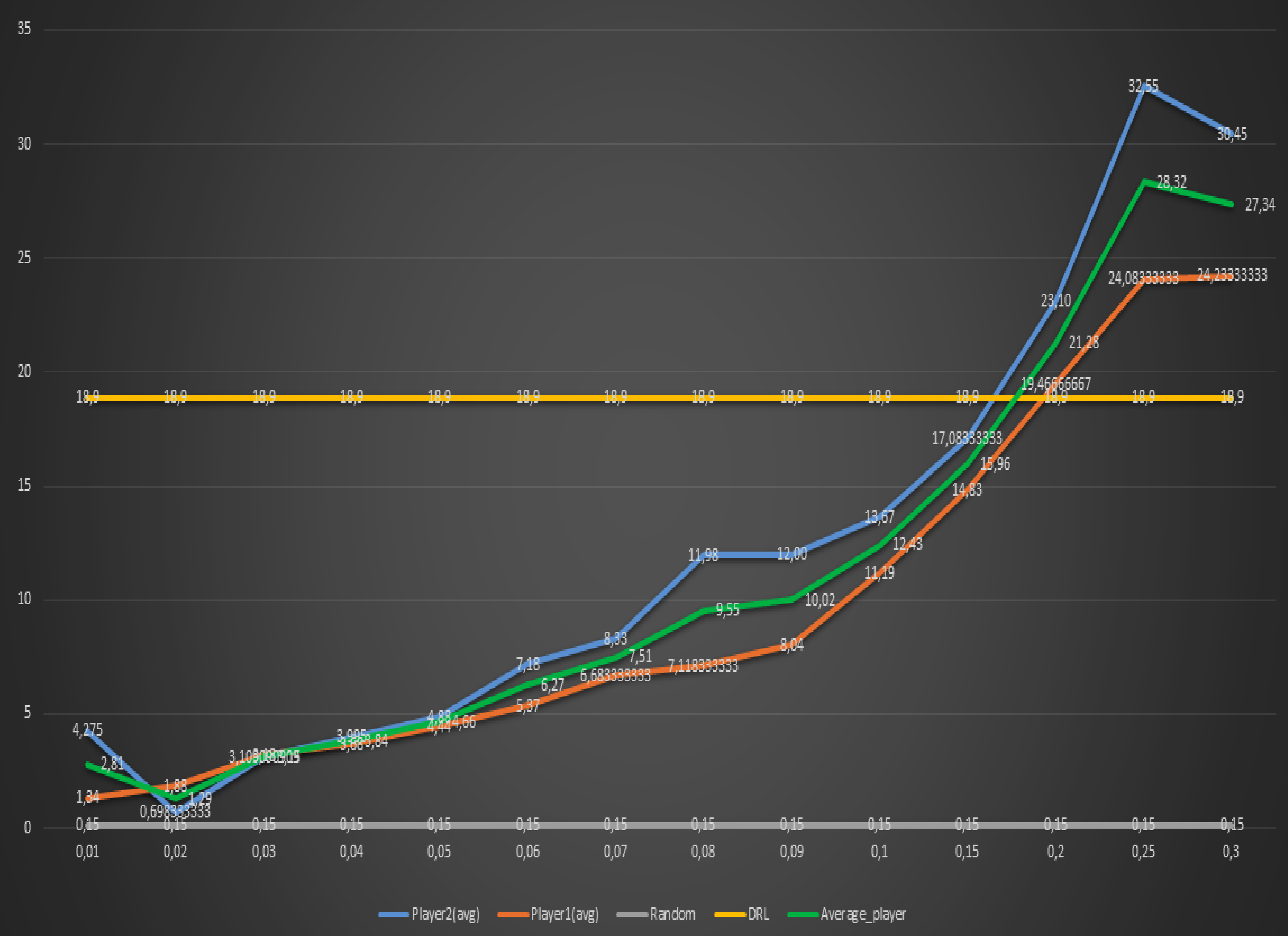
Help in danger situation

Free time

Huge response time

24/7 monitoring


Google
self-driving car
6UEJ089

35
30
25
20
15
10
5
0
0,01
0,02
0,03
0,04
0,05
0,06
0,07
0,08
0,09
0,1
0,15
0,2
0,25
0,3
32,55
30,45
28,32
27,34
24,08333333
24,23333333
23,10
21,28
19,46666667
18,9
17,08333333
15,96
14,83
13,67
12,43
11,19
11,98
12,00
10,02
9,55
8,04
8,33
7,51
7,118333333
6,683333333
7,18
6,27
5,37
4,88
4,275
2,81
1,88
1,29
1,34
0,698333333
0,15
Player2(avg)
Player1(avg)
Random
DRL
Average_player

An intelligent agent perceives its environment via sensors and acts rationally upon that environment with its effectors.

An ideal rational agent should, for each possible percept sequence, do whatever actions will maximize its expected performance measure.

Artificial Intelligence - A Modern Approach | Stuart Russell & Peter Norvig

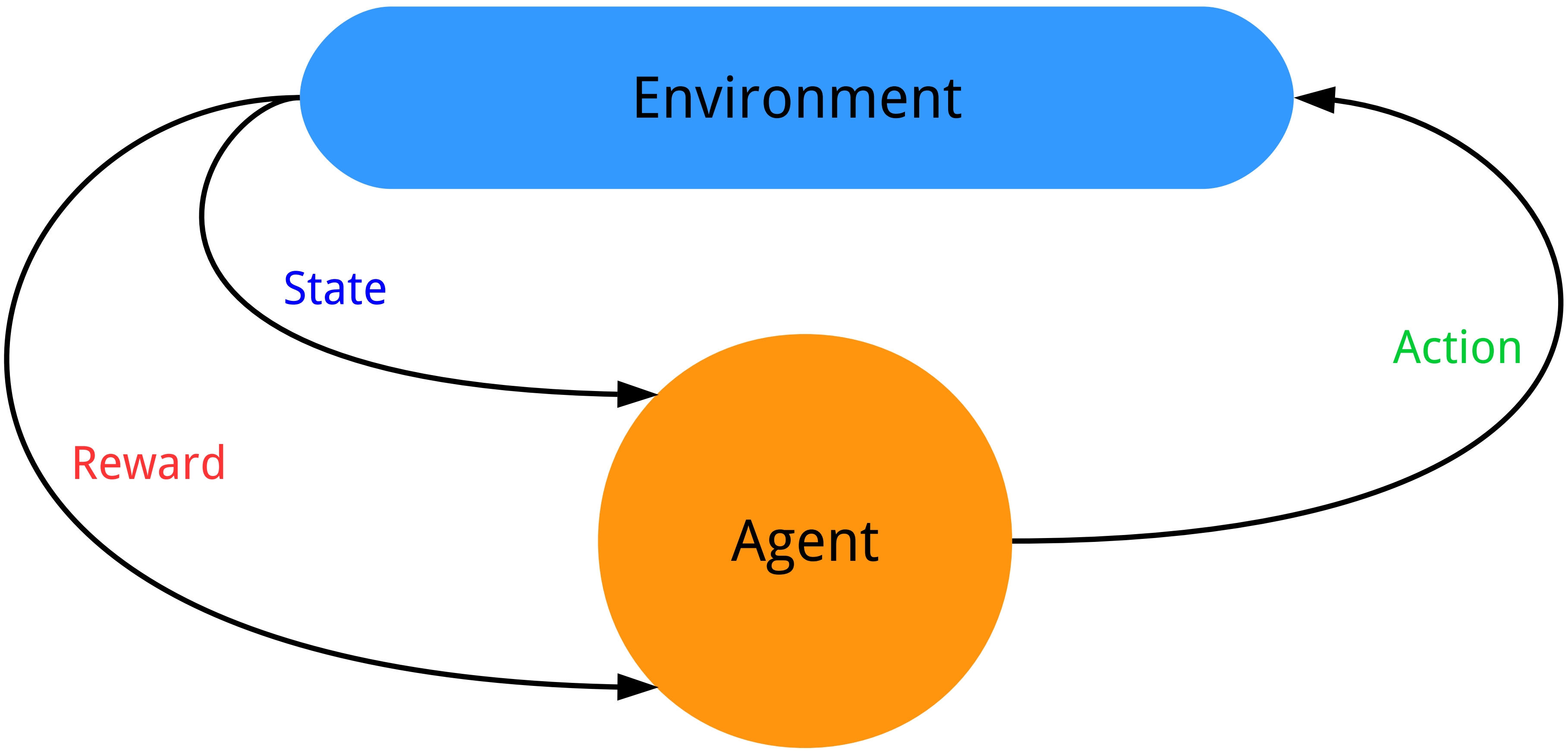
There are many examples of "simple" control which can be solved by if-then-else logic

The world in which automatic door "lives" can be described by a single number

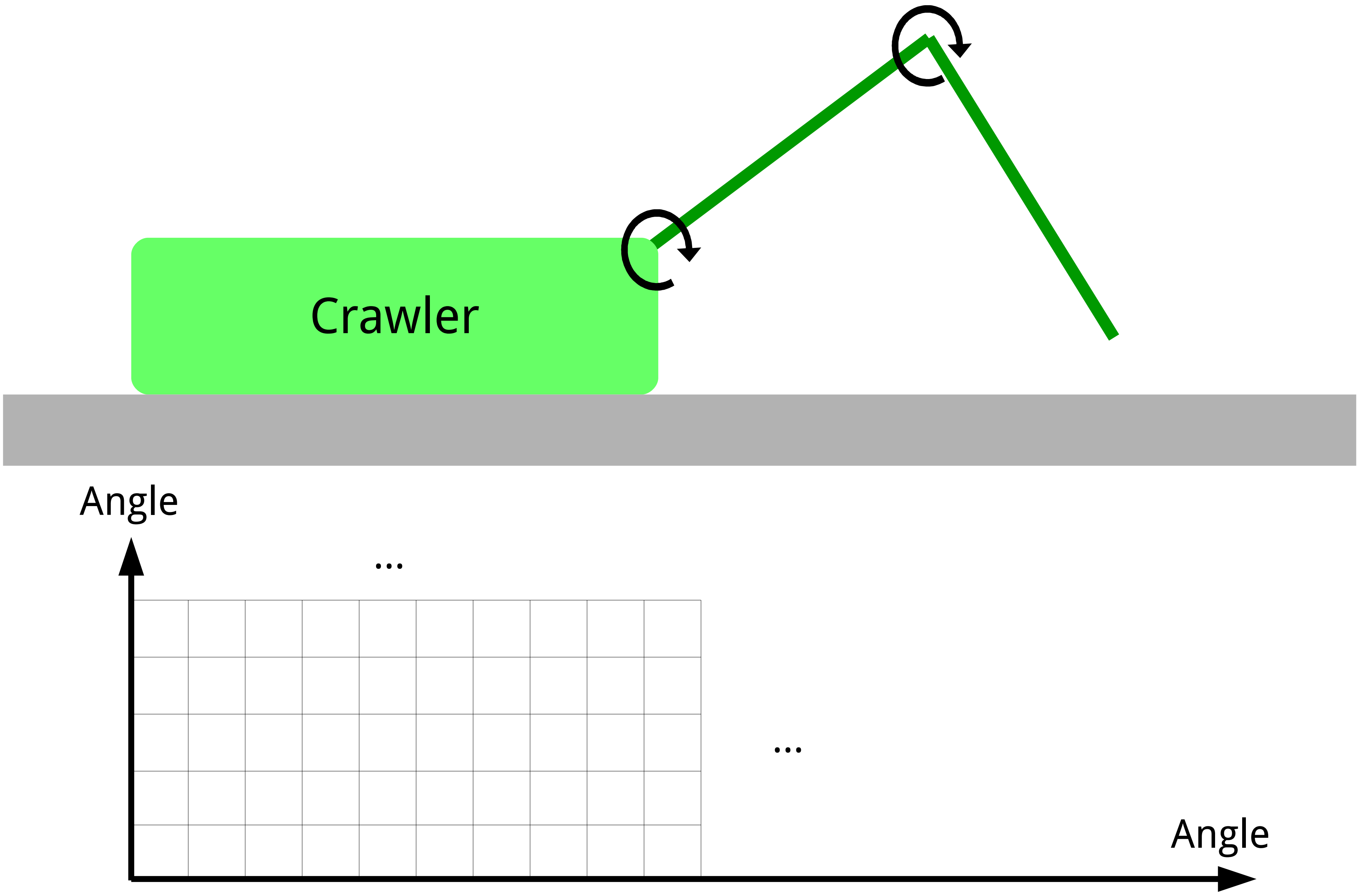
The world around us is stochastic and we can't describe it using few states.

Machine learning is the main instrument that can help us here. Trained system would react even in a situation that it sees for the first time.

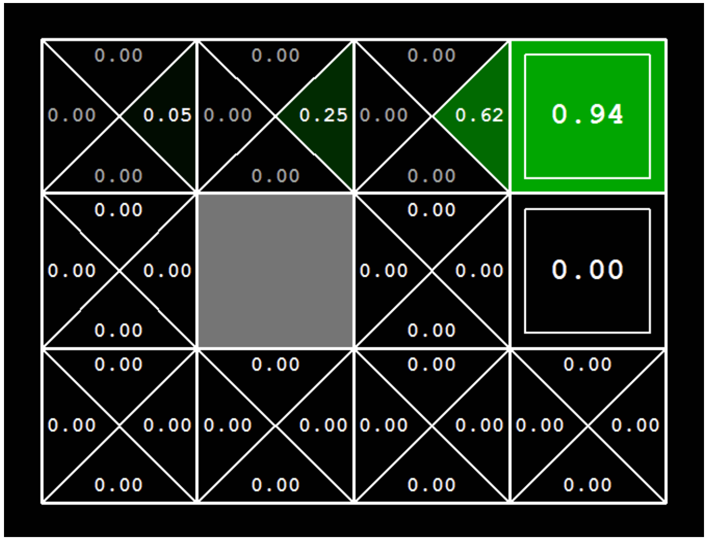
Reinforcement learning



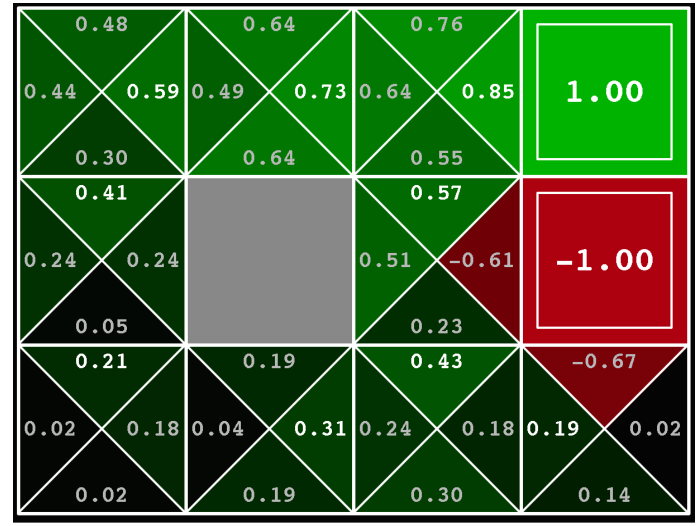
Q learning



Q learning



Early state



State after N iterations

- In Q-learning, the agent learns an action-value function, or Q-function, given the value of taking a given action in a given state.

- Q-learning at its simplest uses tables to store data.

- Each time the agent selects an action, and observes a reward and a new state that may depend on both the previous state and the selected action, "Q" is updated.

$$Q_{t+1}(s_t, a_t) = \underbrace{Q_t(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t, a_t)}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{R_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_a Q_t(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q_t(s_t, a_t)}_{\text{old value}} \right)$$

Q learning

Advantages:

- Q-learning can be used to find an optimal action-selection policy. We should not do it manually.

- Able to compare the expected utility of the available actions without requiring a model of the environment.

Problems:

- Problem of computing the optimal Q-function in environment with infinite state-space.

Q learning with Function approximation

- We take a function approximation approach by representing each state as a small fixed number of features.

This allows us to perform in extremely large state spaces.

- We learn a distinct Q-function for each of actions:

$$Q^a(s,a) = \theta_1^a f_1 + \cdots + \theta_n^a f_n$$

The update rule is:

$$\theta_k^a = \theta_k^a + \alpha[r + \gamma \max_{a'} Q^a(s',a') - Q^a(s,a)] \frac{dQ^a(s,a)}{d\theta_k^a}$$

Q learning with Function approximation

Advantages:

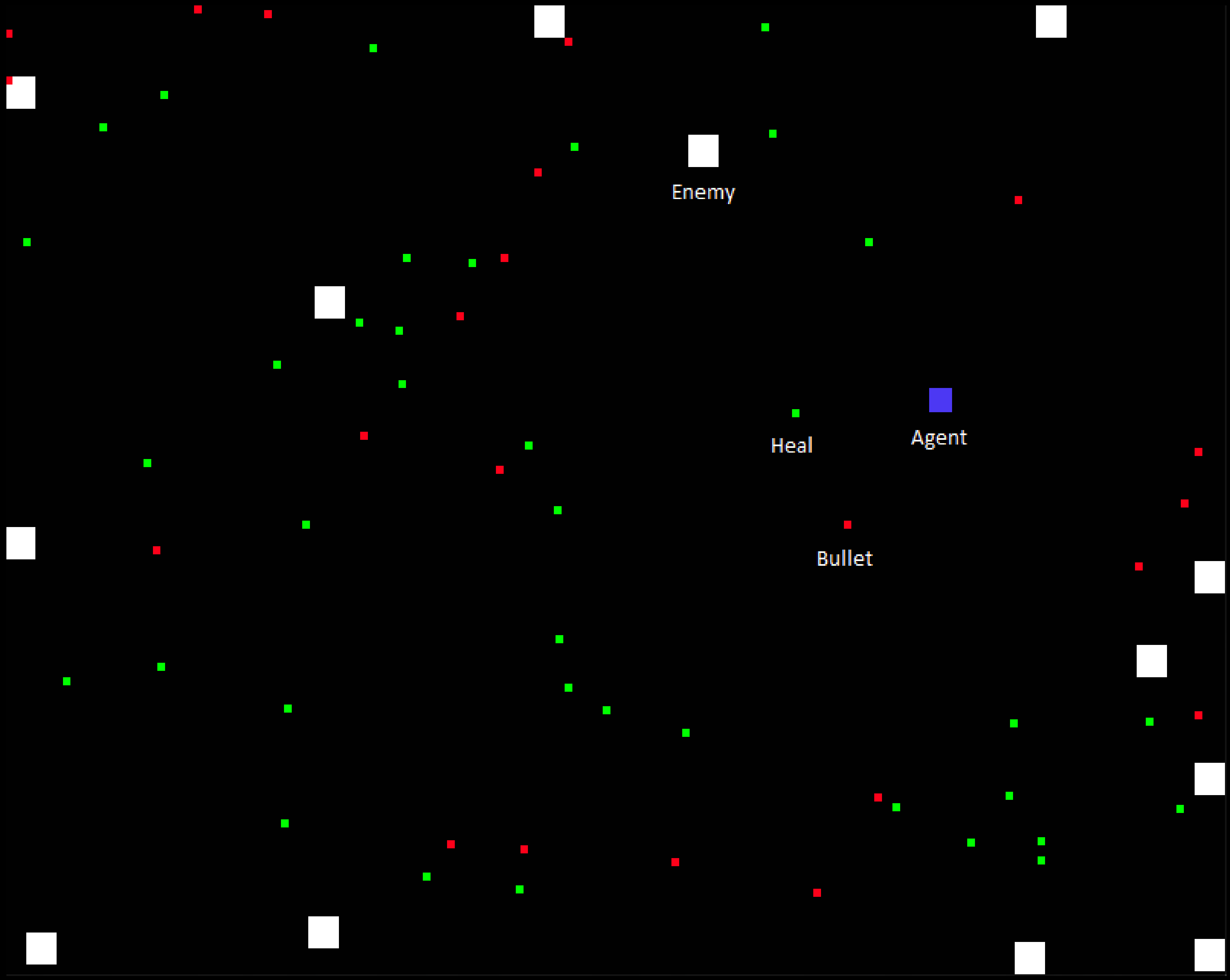
- Possible to apply the algorithm to larger problems, even when the state space is continuous, and therefore infinitely large.

- May speed up learning in finite problems, due to the fact that the algorithm can generalize earlier experiences to previously unseen states.

Problems:

- Hand-crafted features. Performance of such systems heavily relies on the quality of the feature representation.

- Linear value functions.


Enemy
Heal
Agent
Bullet

Deep Reinforcement Learning

Advantages:

- Learning to control agents directly from high-dimensional sensory inputs.

- Extract high-level features from raw sensory data.

Problems:

- Making decisions based only on current state.

Possible improvements:

- Using recurrent architectures of neural nets with memory (LSTM, etc.)

Feature detection

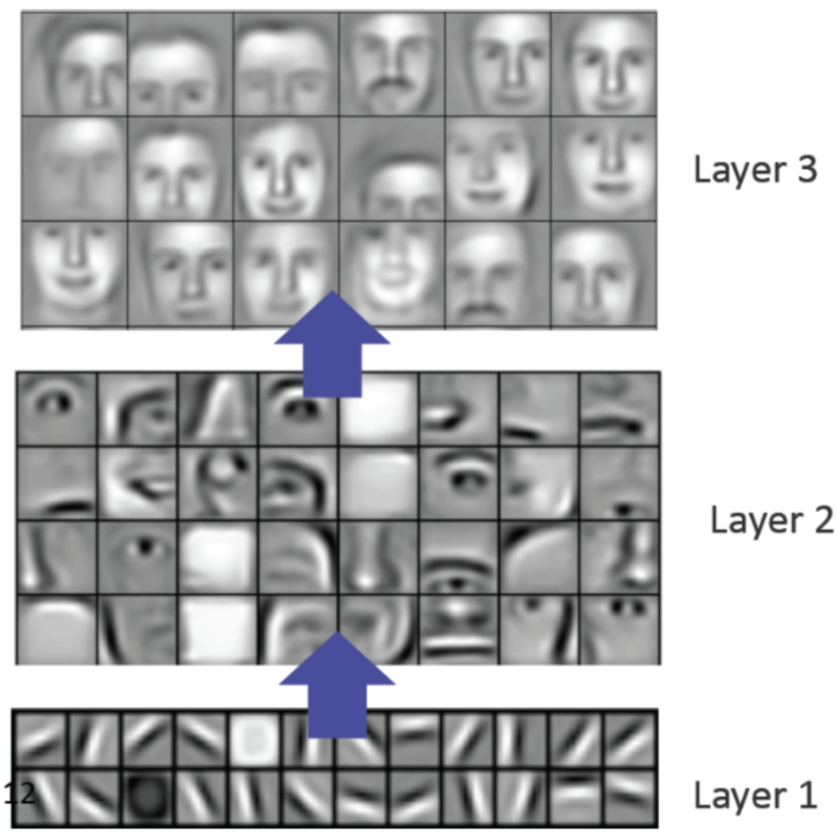
Hand-crafted features disadvantages:

- Subject-matter expert

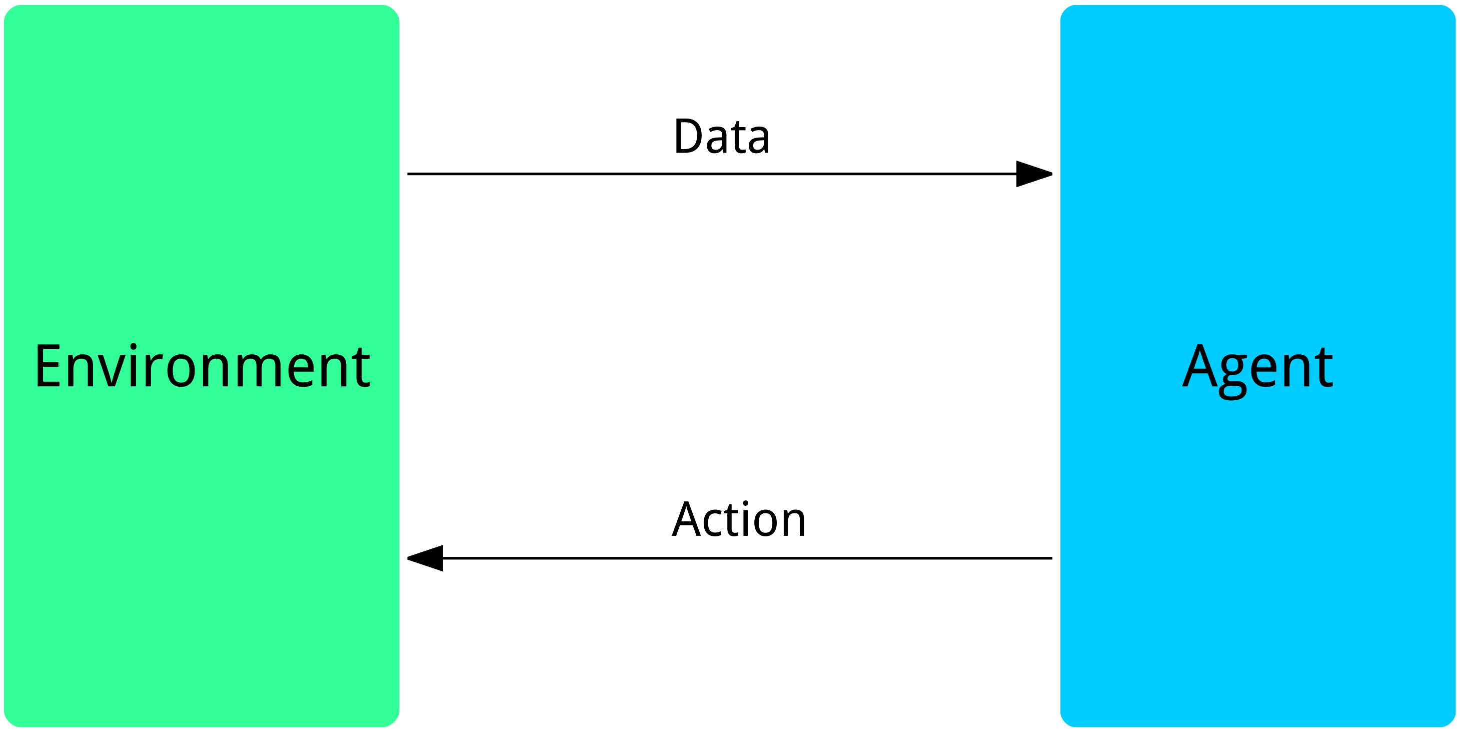
- Time to explore and discover

- Performance of such systems heavily relies on the quality of the feature representation

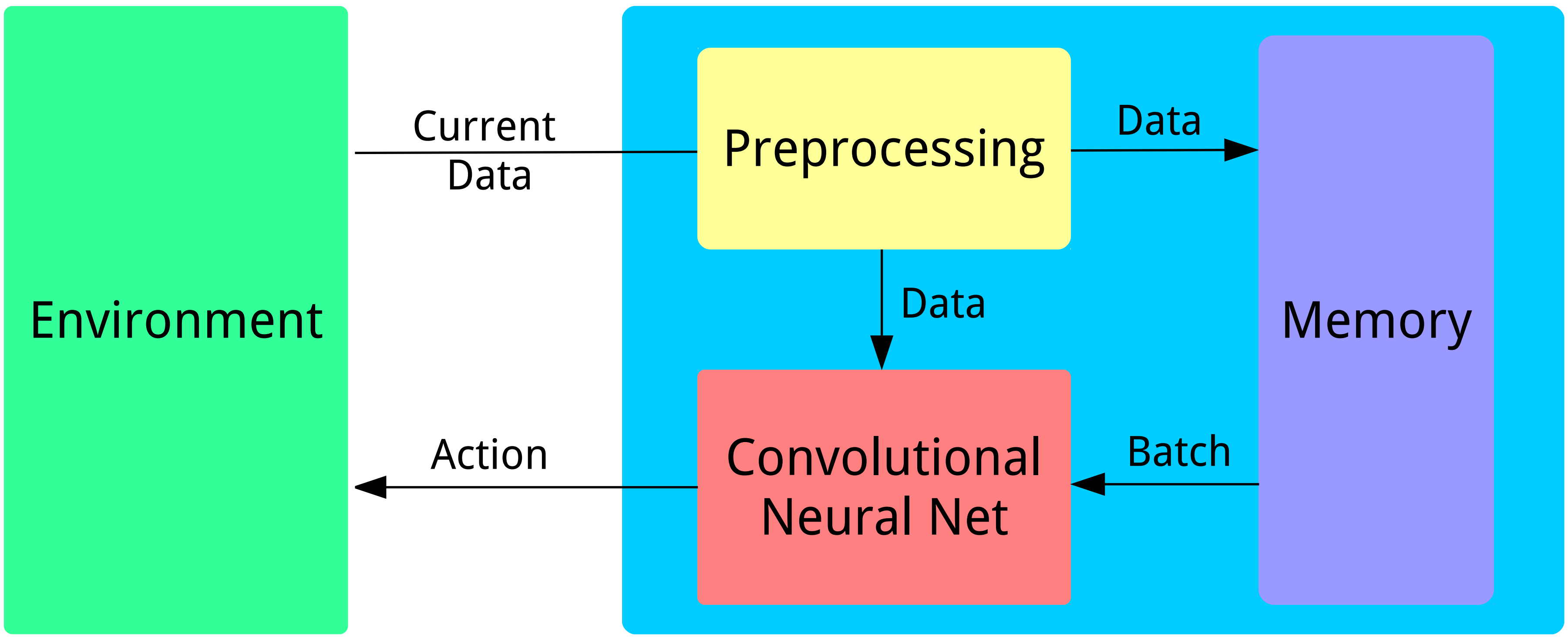
Deep Learning:



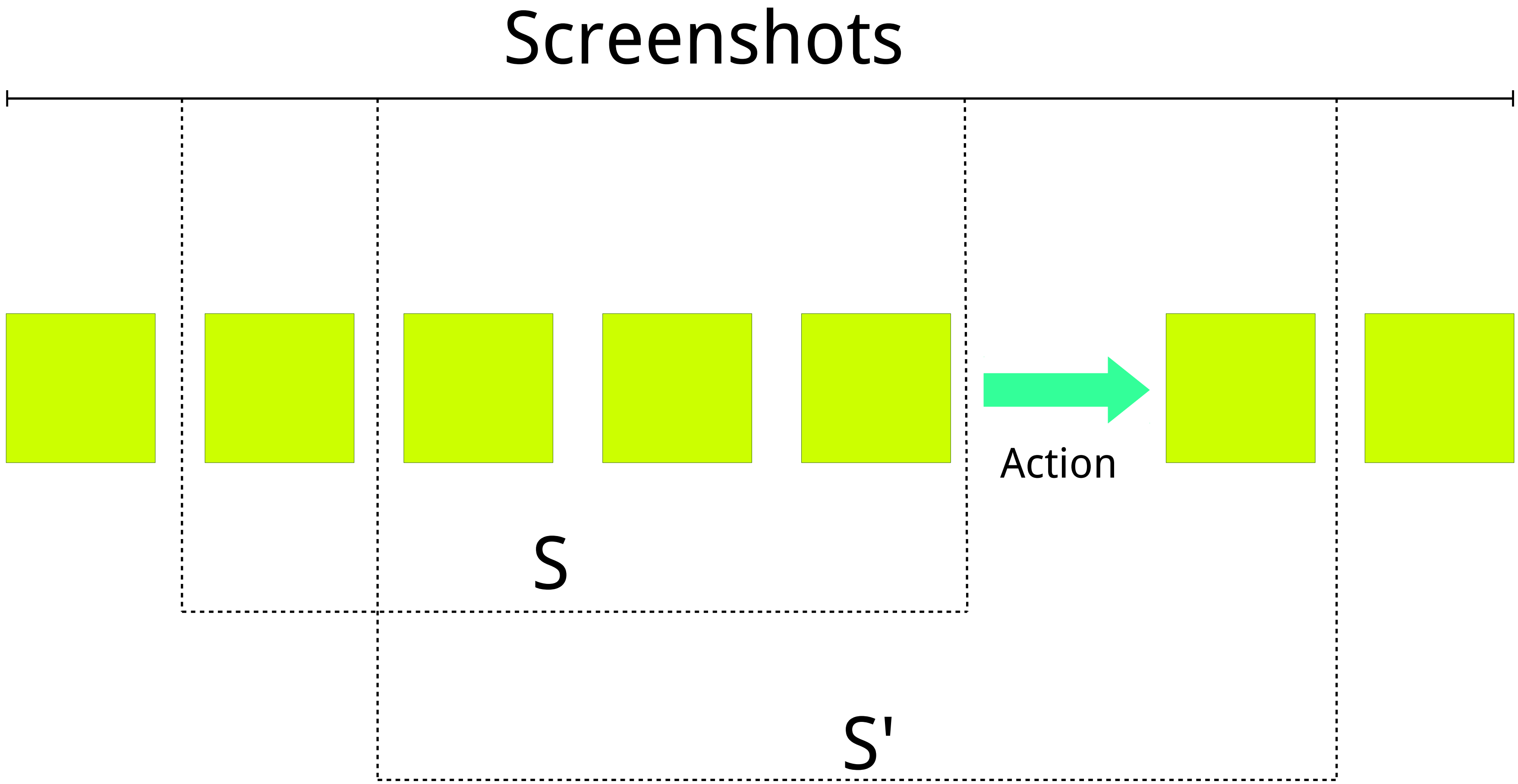
Deep Reinforcement Learning

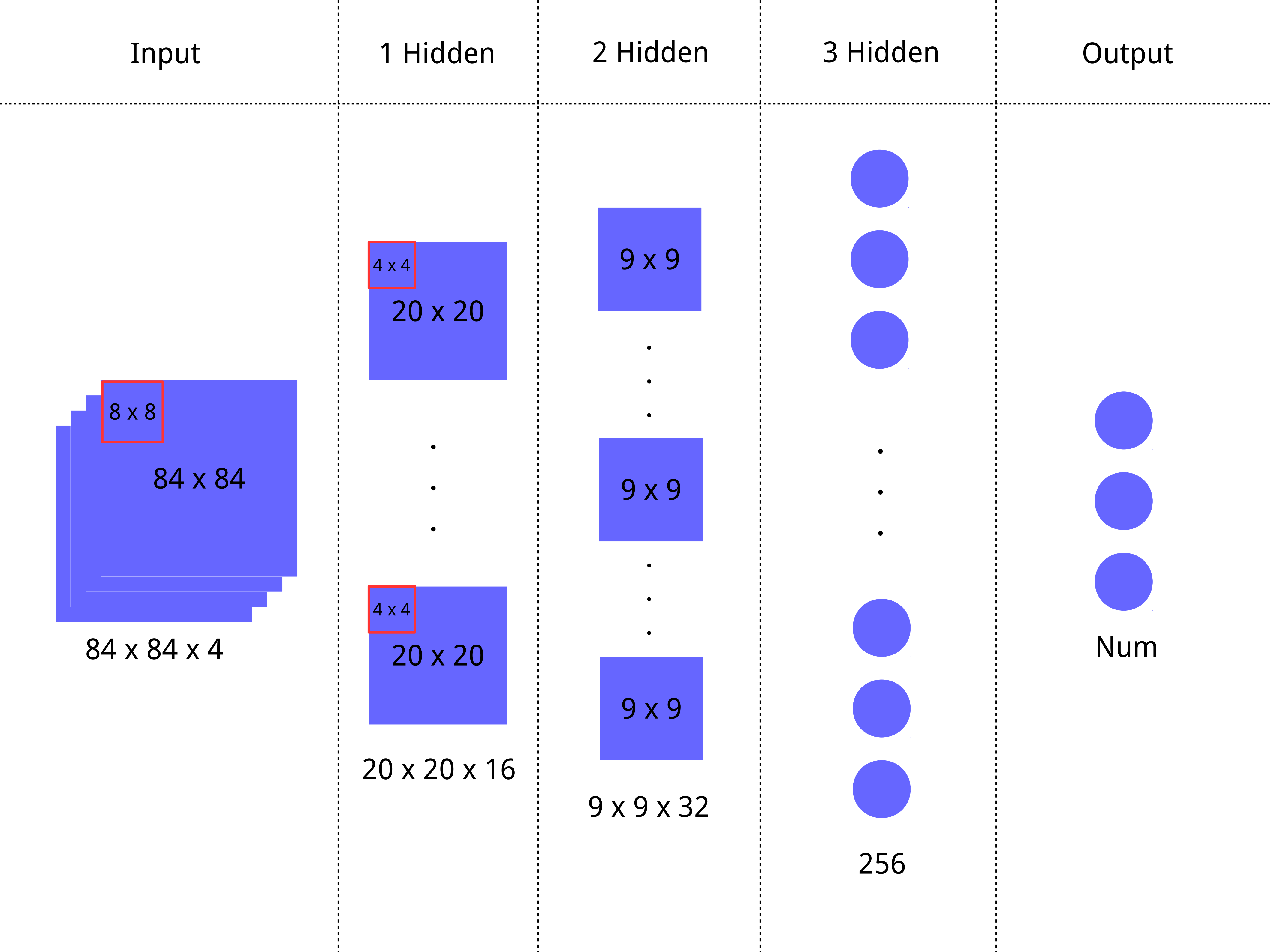


Deep Reinforcement Learning

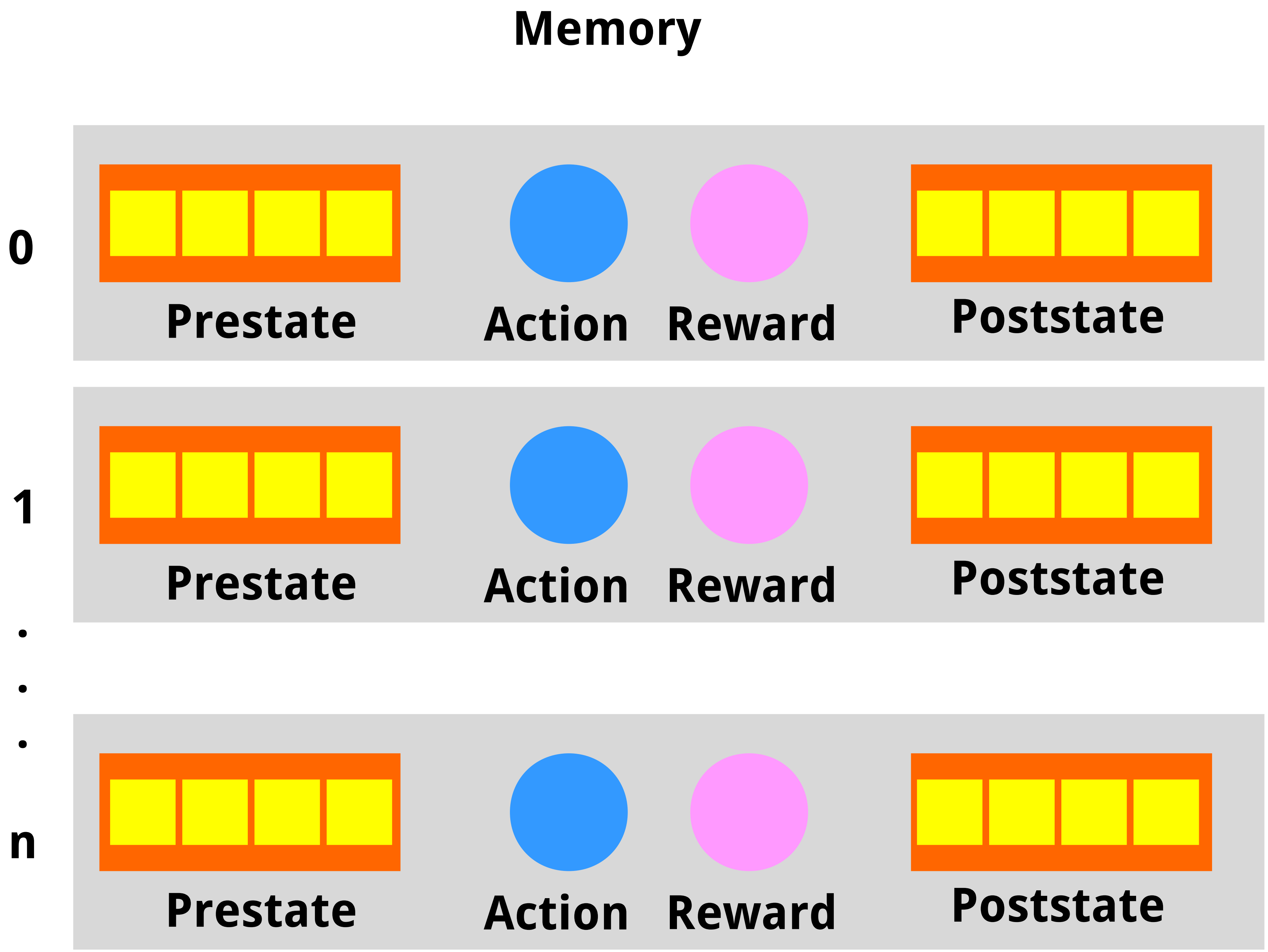


Deep Reinforcement Learning





Input
1 Hidden
2 Hidden
3 Hidden
Output
8 x 8
84 x 84
84 x 84 x 4
4 x 4
20 x 20
4 x 4
20 x 20
20 x 20 x 16
9 x 9
9 x 9
9 x 9
9 x 9 x 32
256
Num


Memory
0
Prestate
Action
Reward
Poststate
1
Prestate
Action
Reward
Poststate
n
Prestate
Action
Reward
Poststate

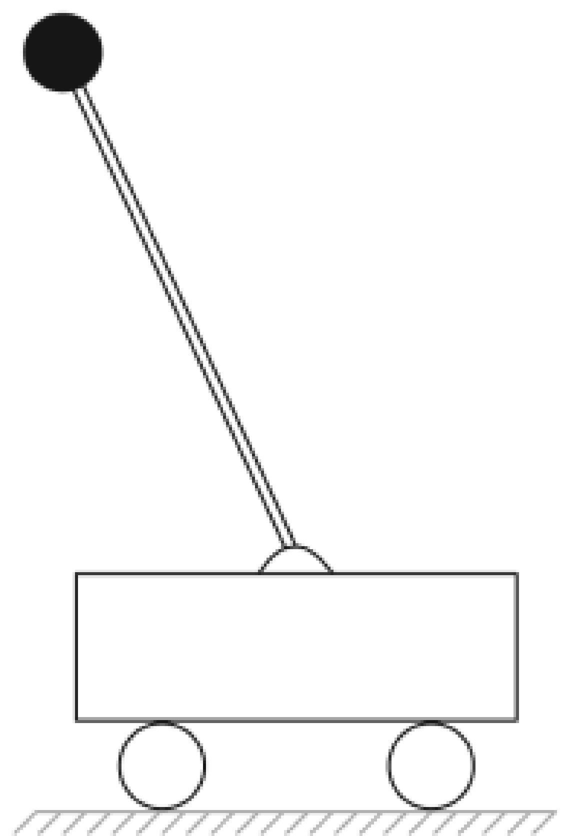

001
2
1

Future

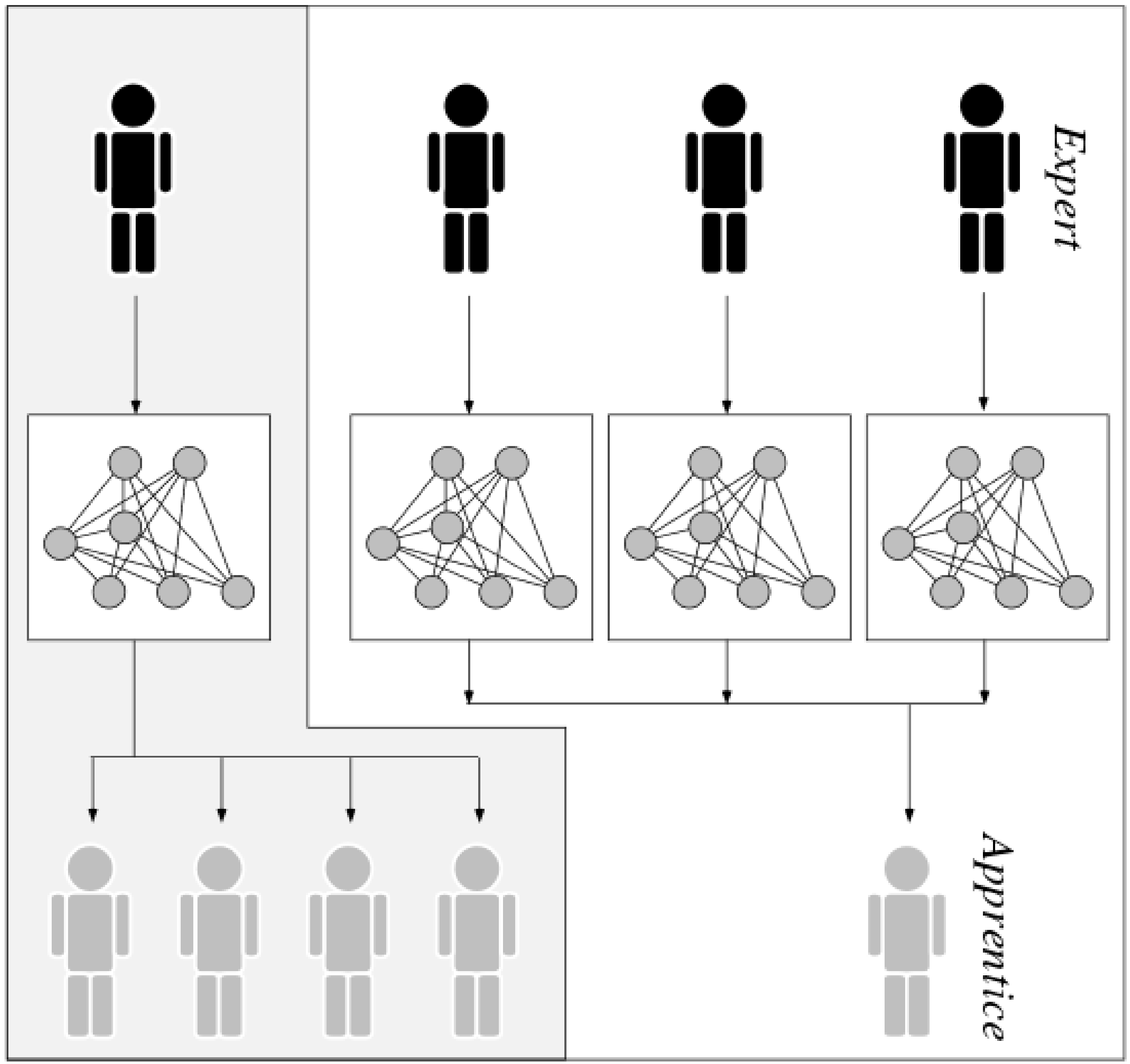
Human Skill Transfer:

- Learning computational models of human skill so that human skill may be successfully transferred to robots and machines.

Inverted pendulum

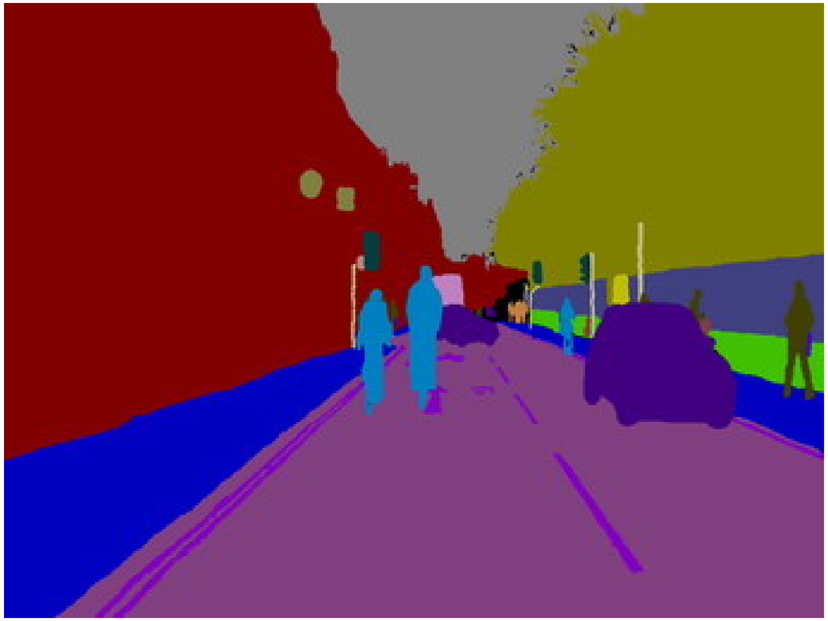


Human Skill Transfer

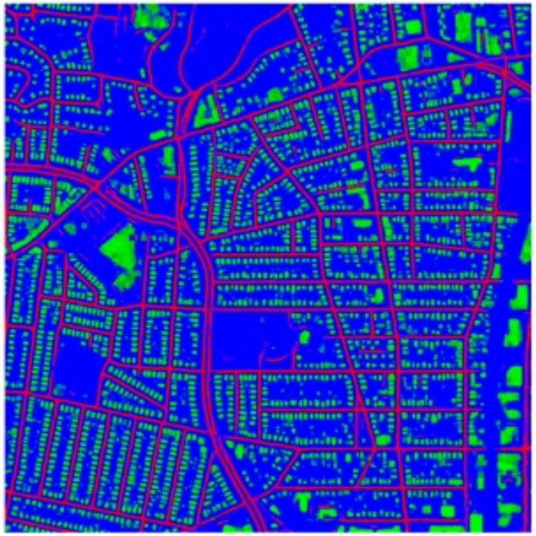


Future

Autonomous cars:



Drone autopilots:



Links

Deep Reinforcement Learning Implementation

http://drlearner.org/

https://github.com/DSG-SoftServe/DRL

Neural Net Framework for Deep Learning

https://github.com/spaceuniverse/TNNF

Q-learning sandbox with Function approximation

https://github.com/spaceuniverse/QLSD

Related papers

http://www.cs.uic.edu/~sloan/my-papers/FLAIRS05-to-appear.pdf

https://www.ri.cmu.edu/pub_files/pub1/nechyba_michael_1995_2/nechyba_michael_1995_2.pd

https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf

http://www.cs.nyu.edu/~fergus/papers/zeilerECCV2014.pdf

Info

My LinkedIn

https://www.linkedin.com/in/awesomengineer

Demo video

https://youtu.be/T58HkwX-OuI

https://youtu.be/IsF4IDfKNgE

Pictures and different stuff:

http://universespace.tumblr.com/

Thanks ^_^