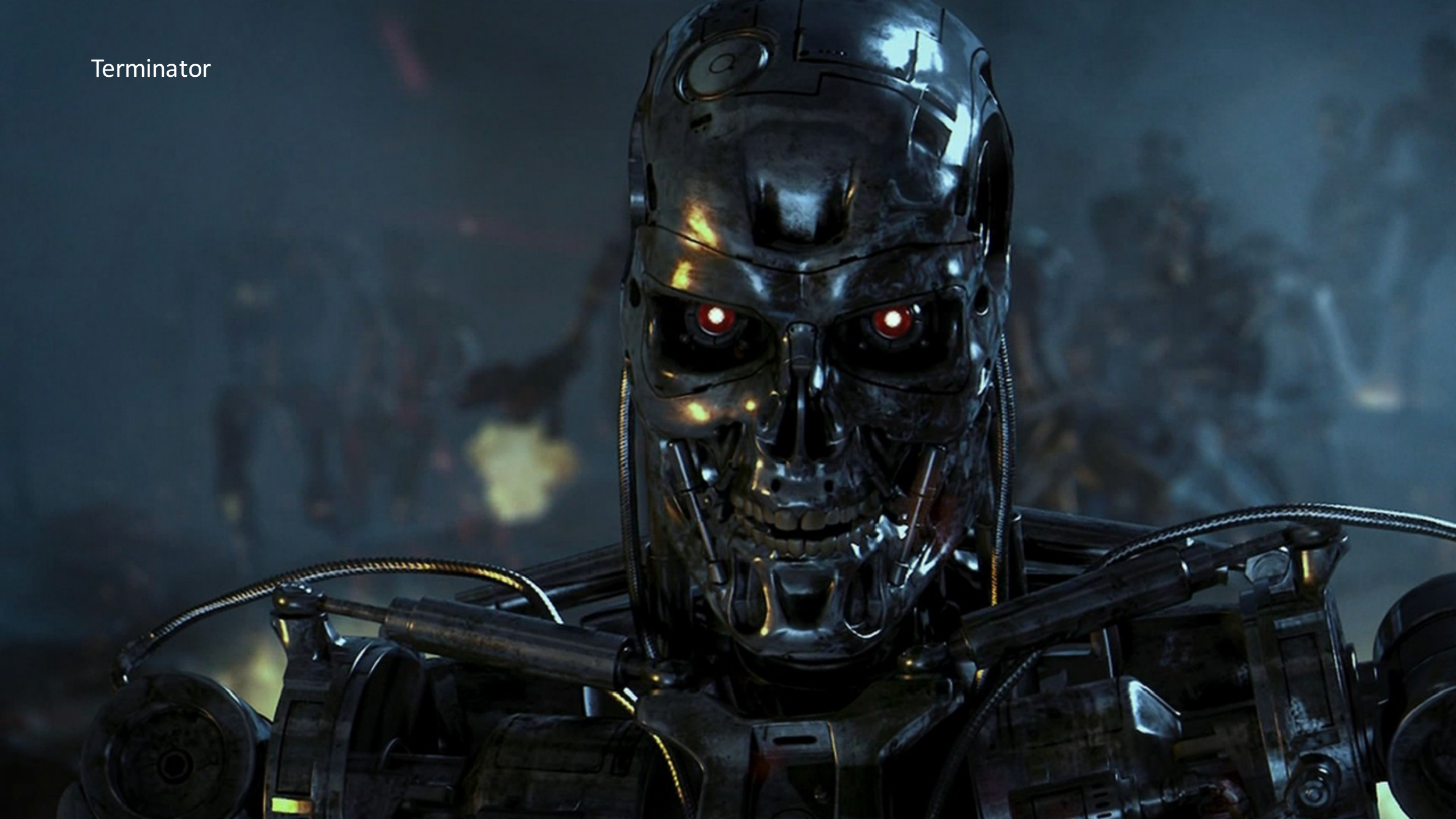
游戏高手

通用人工智能之梦

张江

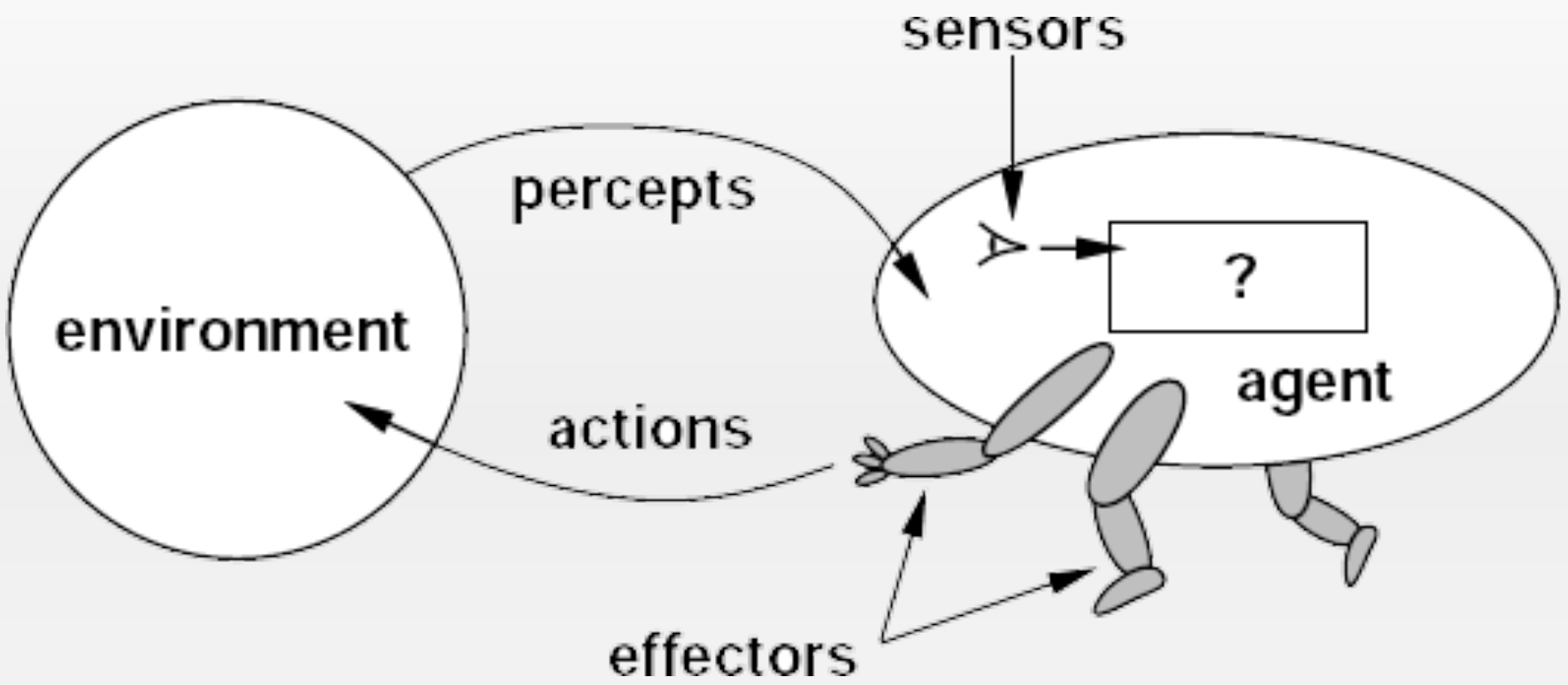
集智AI学园
campus.swarma.org


Terminator

WallE

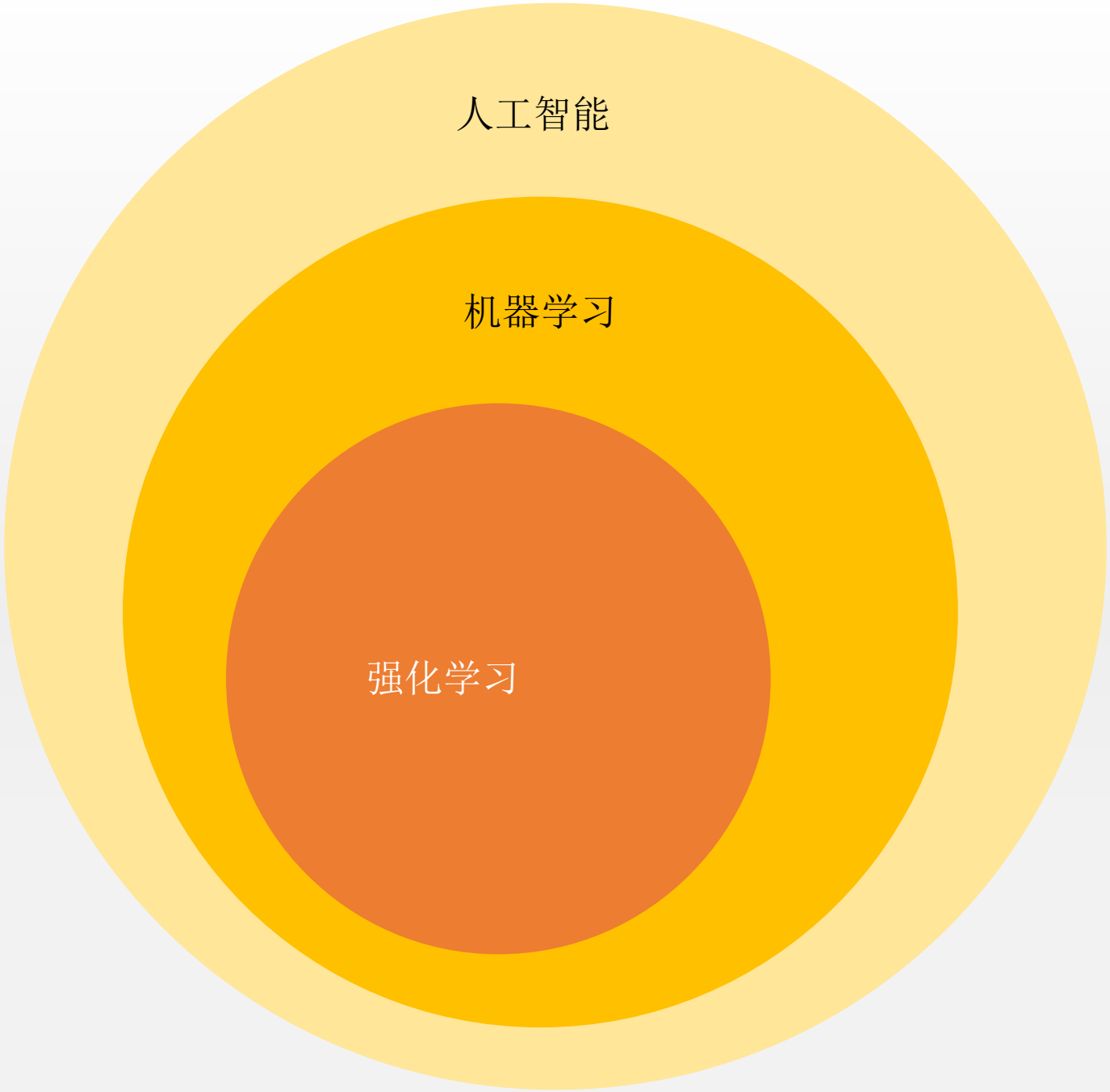

A.I.

人工智能 v.s. 机器学习



智能体（Intelligent agent）应该是一个完整的可以感知、决策、行动、学习的整体，而不是分开的一系列任务

强化学习



Reinforcement
Learning

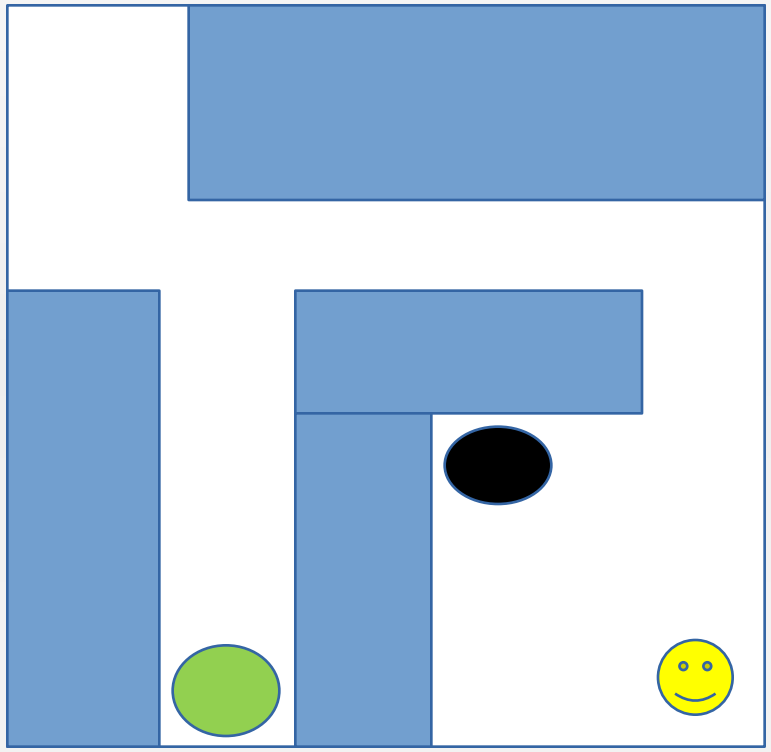
什么是强化学习？

Reinforcement learning (**RL**) is an area of machine learning inspired by behaviourist psychology, concerned with how software agents ought to take *actions* in an *environment* so as to maximize some notion of cumulative *reward*.

- 没有即时反馈
- 边做边学
  - 平衡探索与利用

Arthur Samuel

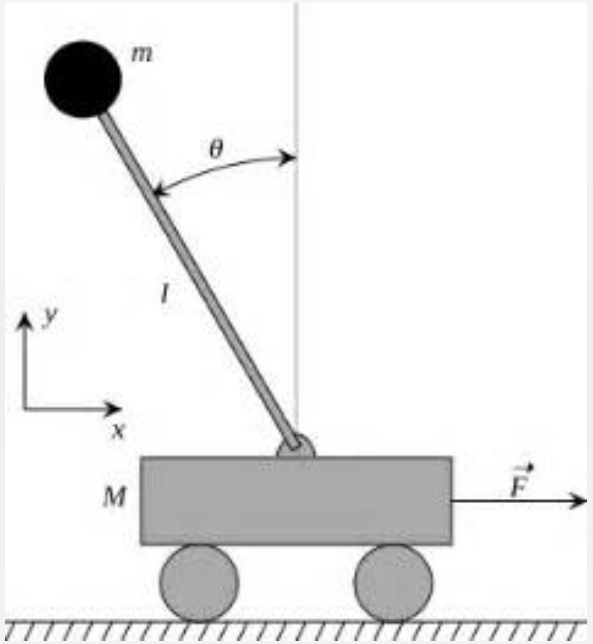
典型的强化学习框架

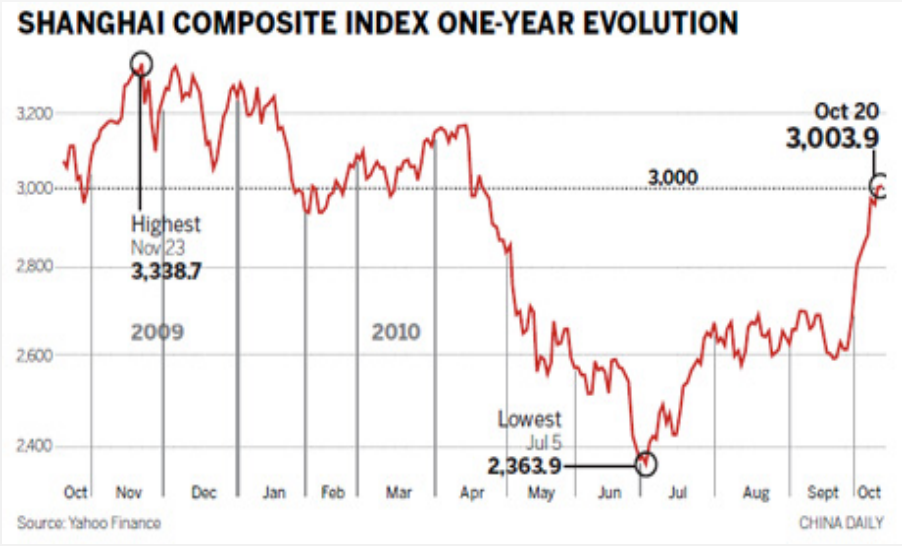


- 每一时刻
- 环境状态S，纪录Agent可能的位置(x,y)
- 有一个Agent，它可以采取行动：A={0,1,2,…)
- 状态转移：
  - σ(S，α)=S'
- 环境会给Agent反馈r:
  - 踩到炸弹－10
  - 踩到蘑菇＋20
  - 一般情况0

强化学习应用领域

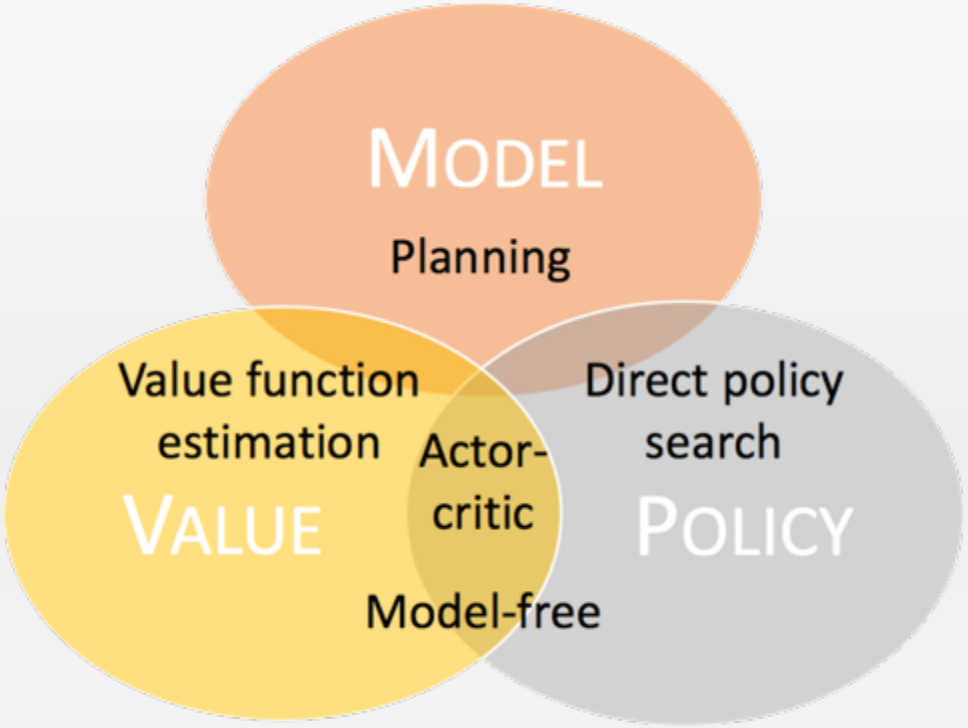

柯洁 KE JIE
00:20:36
ALPHAGO
01:57:08


m
θ
l
y
x
M
F


SHANGHAI COMPOSITE INDEX ONE-YEAR EVOLUTION
3,200
3,000
2,800
2,600
2,400
Highest
Nov 23
3,338.7
Oct 20
3,003.9
3,000
2009
2010
Lowest
Jul 5
2,363.9
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sept
Oct
Source: Yahoo Finance
CHINA DAILY

强化学习

- 按照Agent对环境的理解程度
  - Model-free: Q-learning, DQN,...
  - Model-based: AlphaGo
- 按照学习对象的划分
  - Policy based
  - Value based: AI打游戏
  - Policy-value: AlphaGo
  - Actor-critic



游戏的人工智能

LETTER

doi:10.1038/nature14236

Human-level control through deep reinforcement learning

Volodymyr Mnih[1]*, Koray Kavukcuoglu[1]*, David Silver[1]*, Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

The theory of reinforcement learning provides a normative account[1], deeply rooted in psychological[2] and neuroscientific[3] perspectives on animal behaviour, of how agents may optimize their control of an environment. To use reinforcement learning successfully in situations approaching real-world complexity, however, agents are confronted with a difficult task: they must derive efficient representations of the environment from high-dimensional sensory inputs, and use these to generalize past experience to new situations. Remarkably, humans and other animals seem to solve this problem through a harmonious combination of reinforcement learning and hierarchical sensory processing systems[4,5], the former evidenced by a wealth of neural data revealing notable parallels between the phasic signals emitted by dopaminergic neurons and temporal difference reinforcement learning

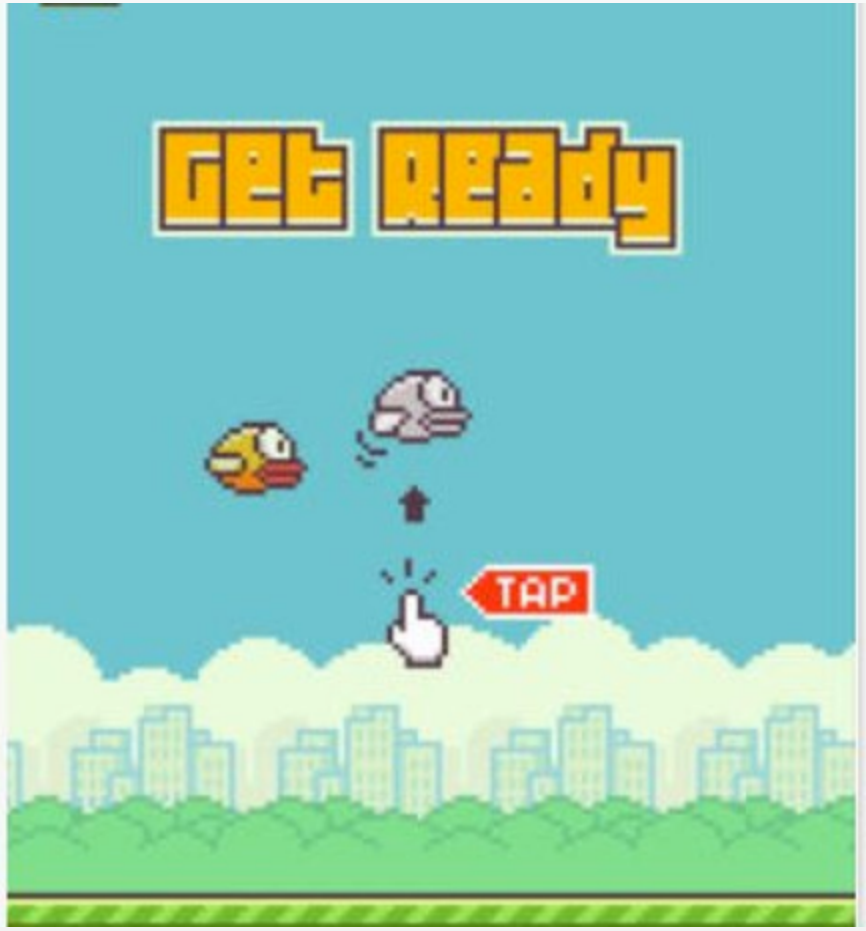
agent is to select actions in a fashion that maximizes cumulative future reward. More formally, we use a deep convolutional neural network to approximate the optimal action-value function

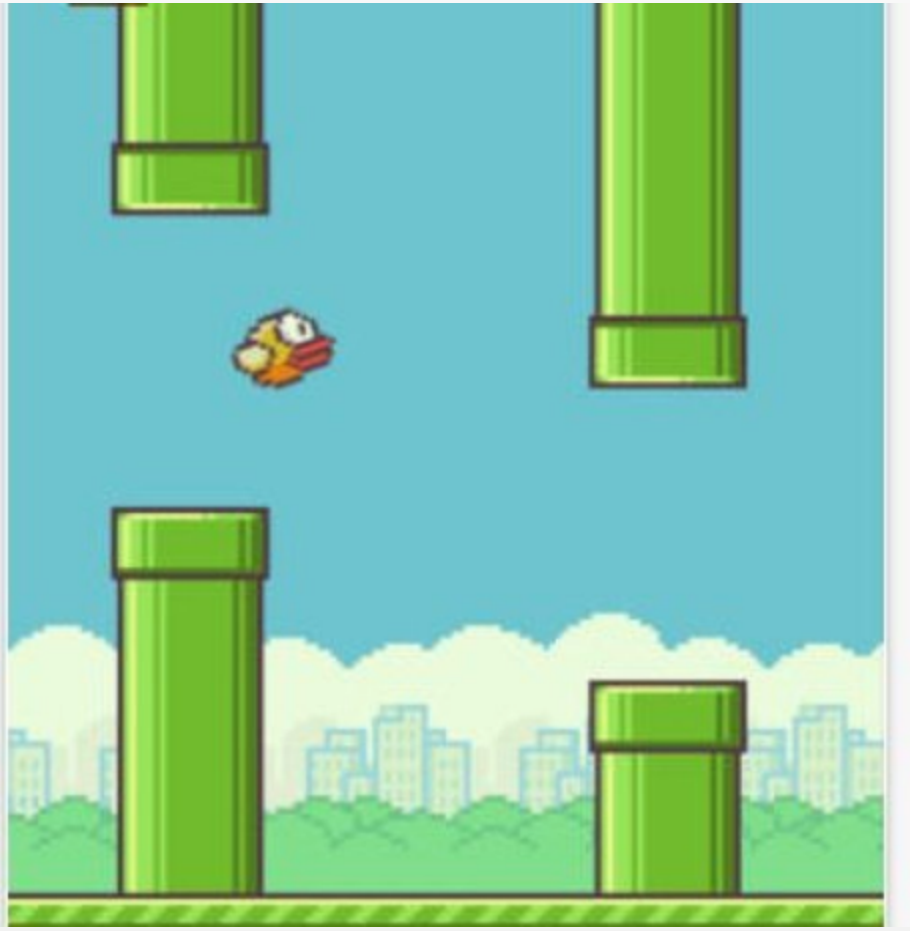
$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots | s_t = s,\, a_t = a,\, \pi\right],$$

which is the maximum sum of rewards r_t discounted by γ at each time-step t, achievable by a behaviour policy $\pi = P(a|s)$, after making an observation (s) and taking an action (a) (see Methods)[19].

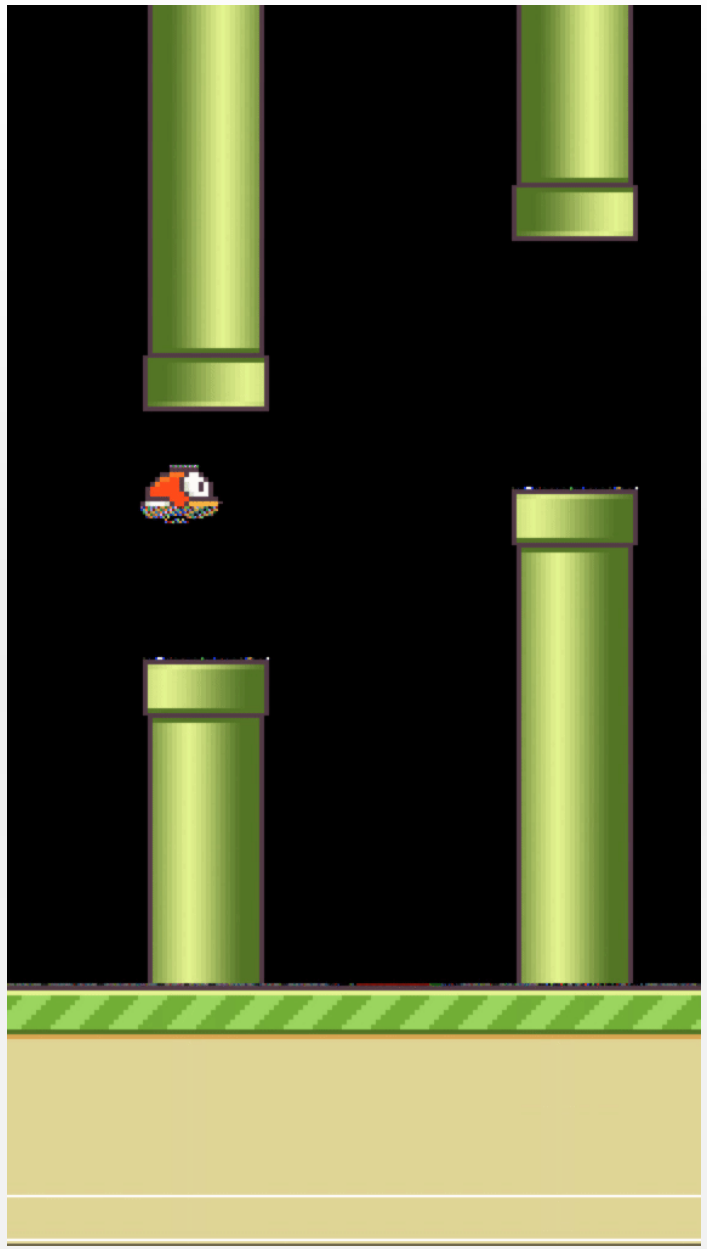
Reinforcement learning is known to be unstable or even to diverge when a nonlinear function approximator such as a neural network is used to represent the action-value (also known as Q) function[20]. This instability has several causes: the correlations present in the sequence of observations, the fact that small updates to Q may significantly change

Flappy bird

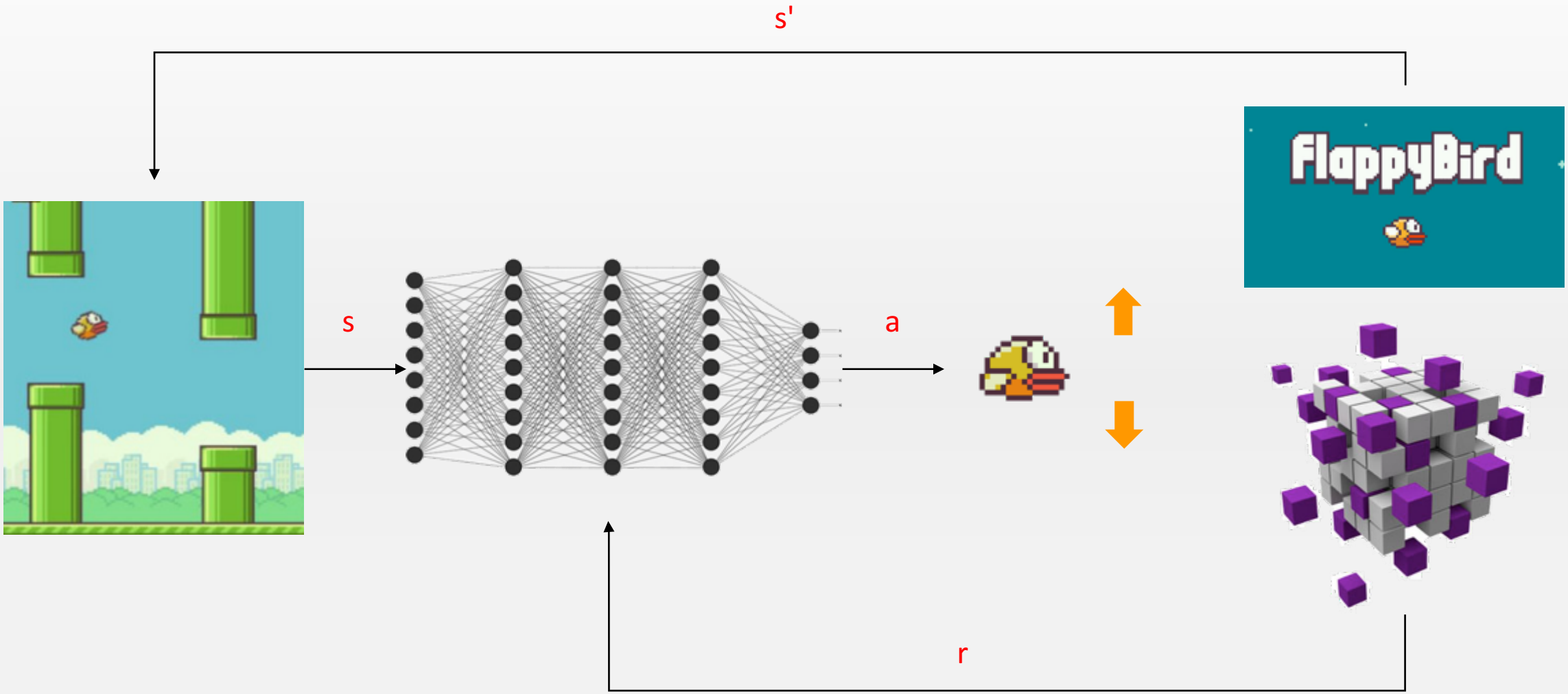




Flappy bird



游戏高手



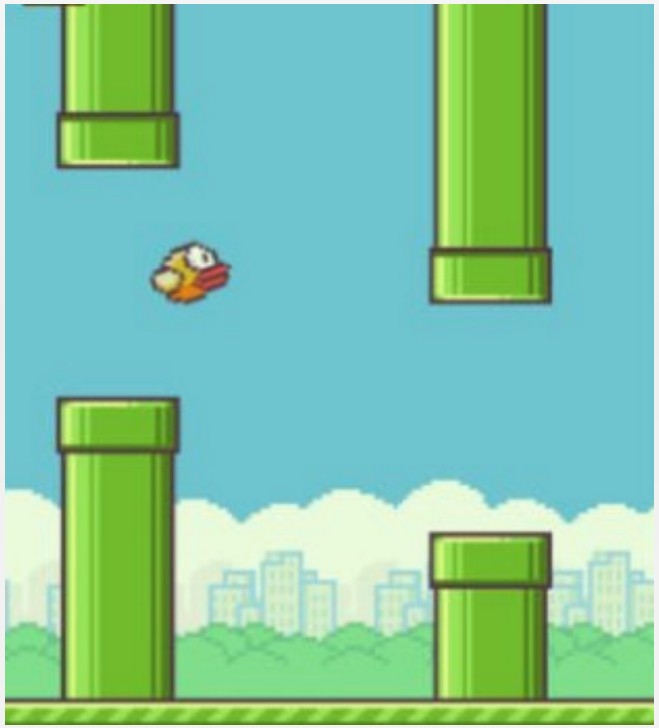
Flappy bird

+0.1

每一帧都＋0.1分

Flappy bird

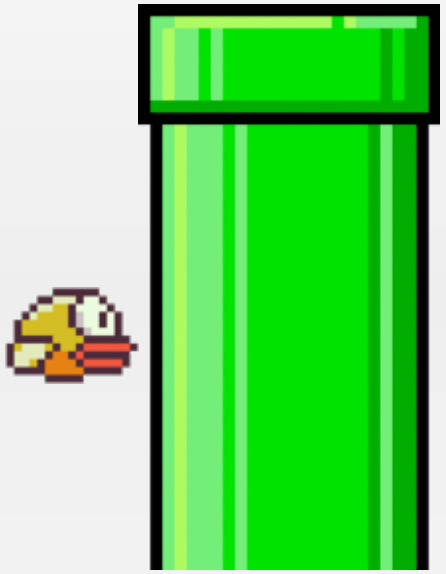
+1



每过一个管道＋1分

Flappy bird

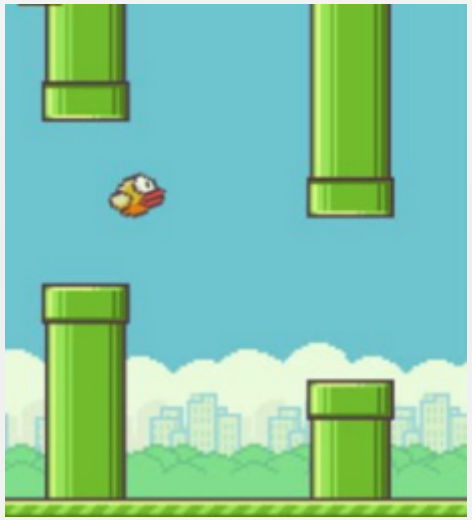
-1



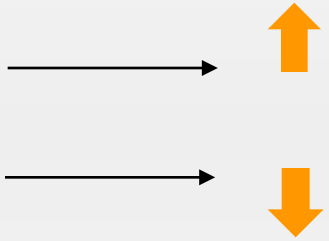
每撞一个管道-1分

游戏重新开始

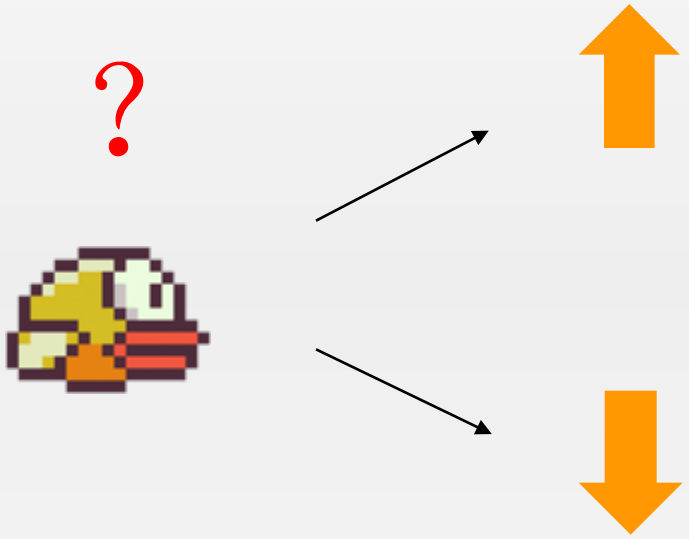
游戏高手：神经网络



S

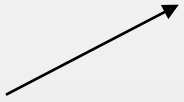


游戏高手：评价系统

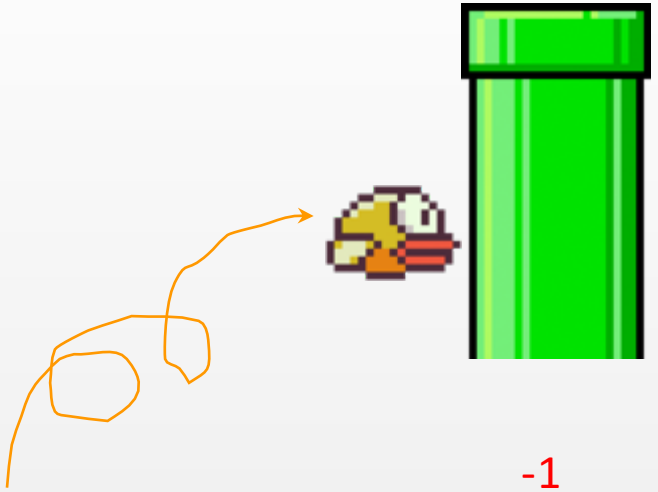


我们应该如何评价这两个行动的好坏呢？

游戏高手：评价系统



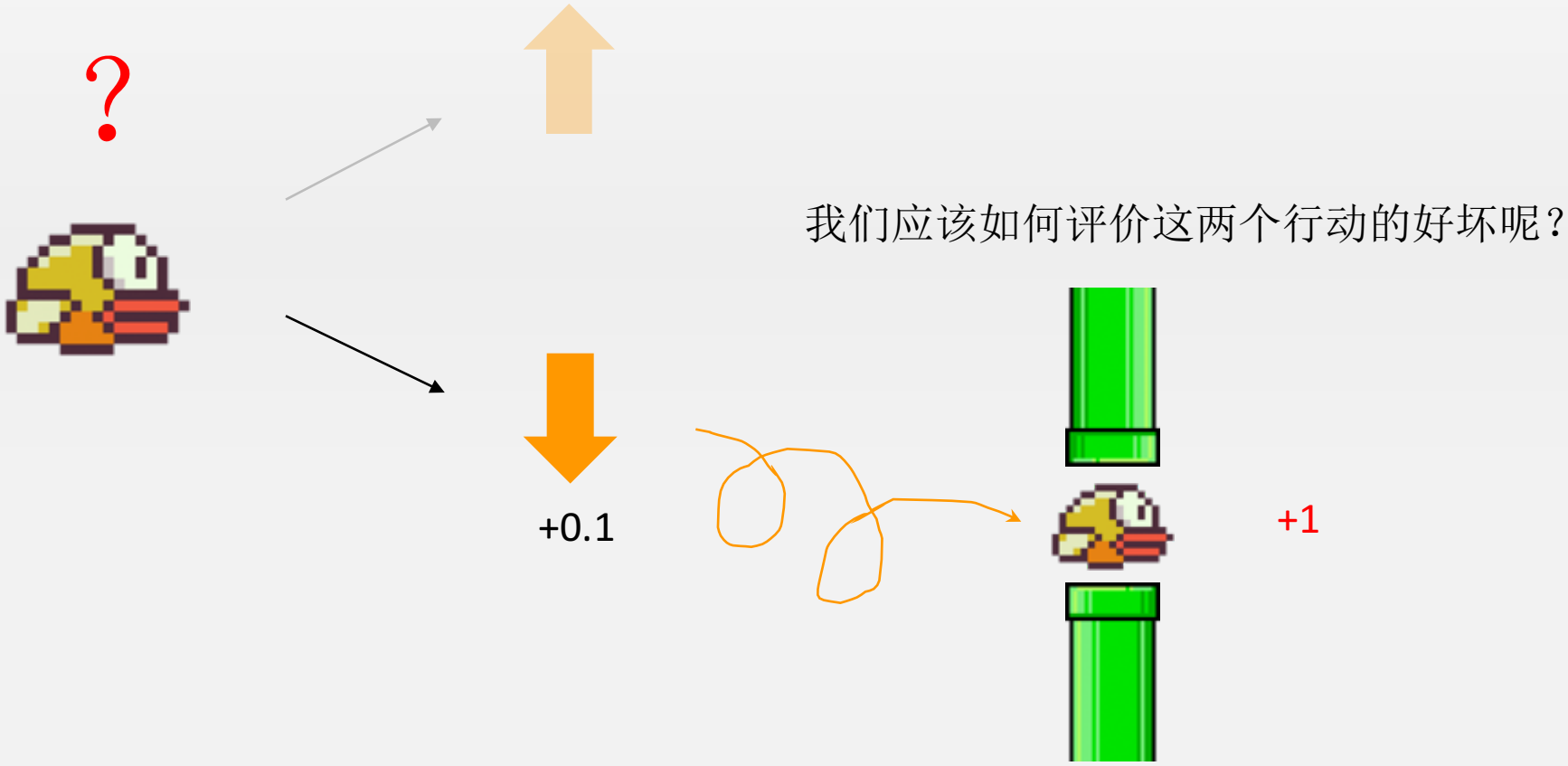
+0.1



-1

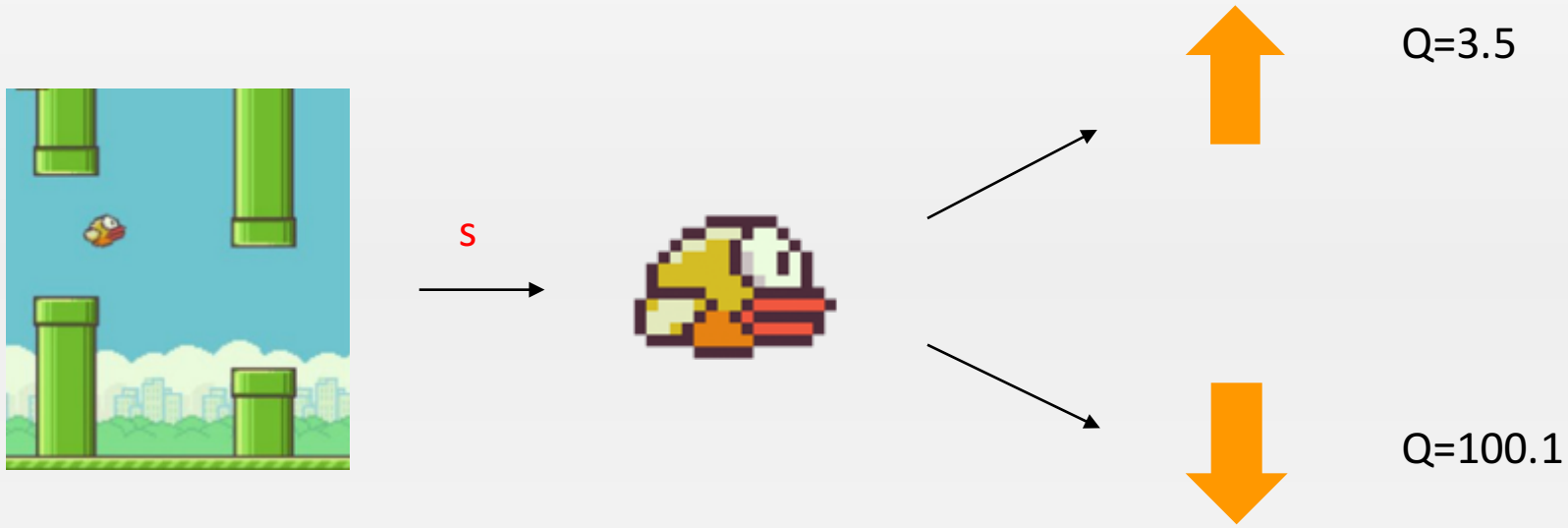
我们应该如何评价这两个行动的好坏呢？

游戏高手：评价系统



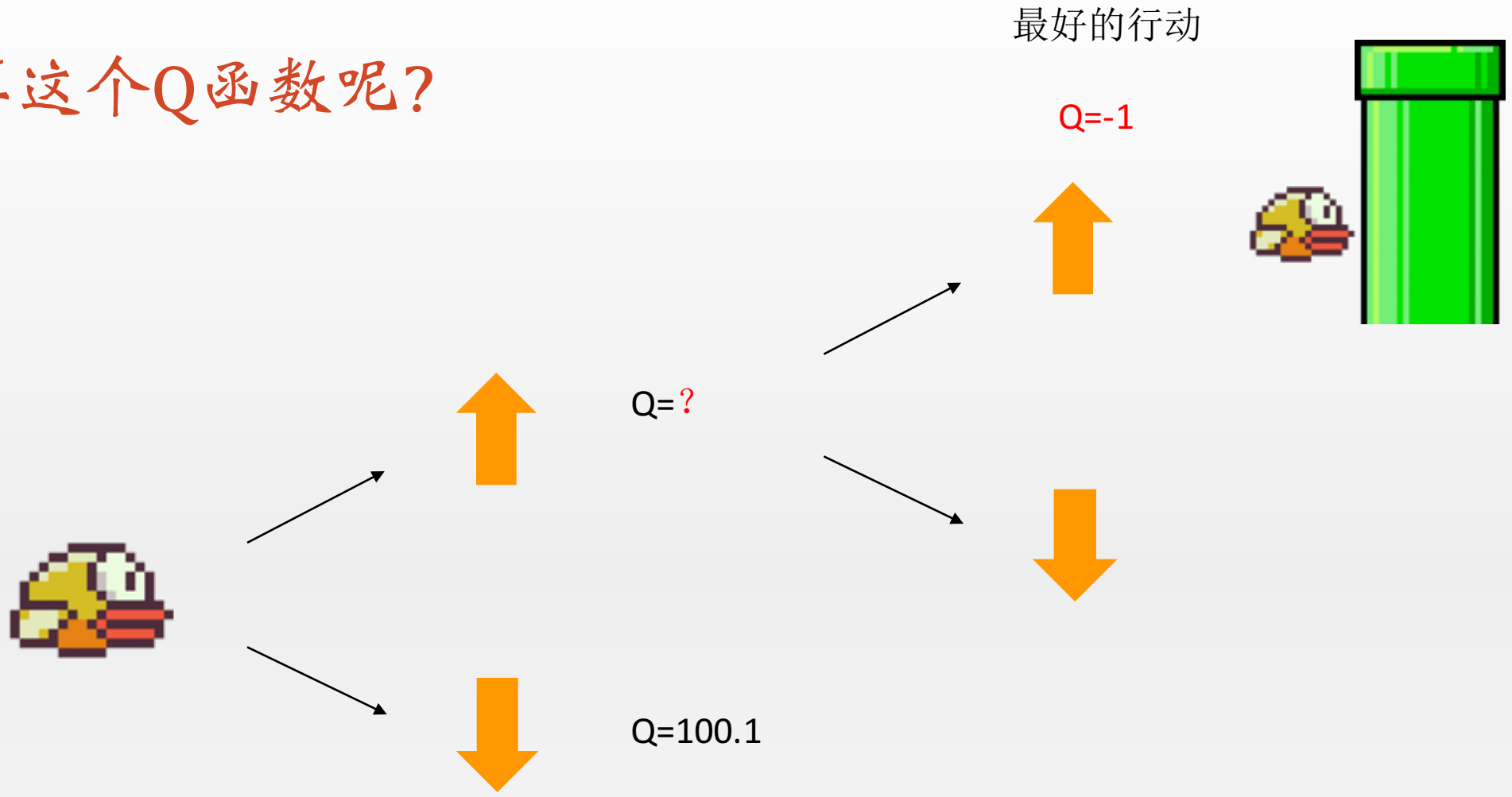
我们应该如何评价这两个行动的好坏呢？

游戏高手：评估函数Q



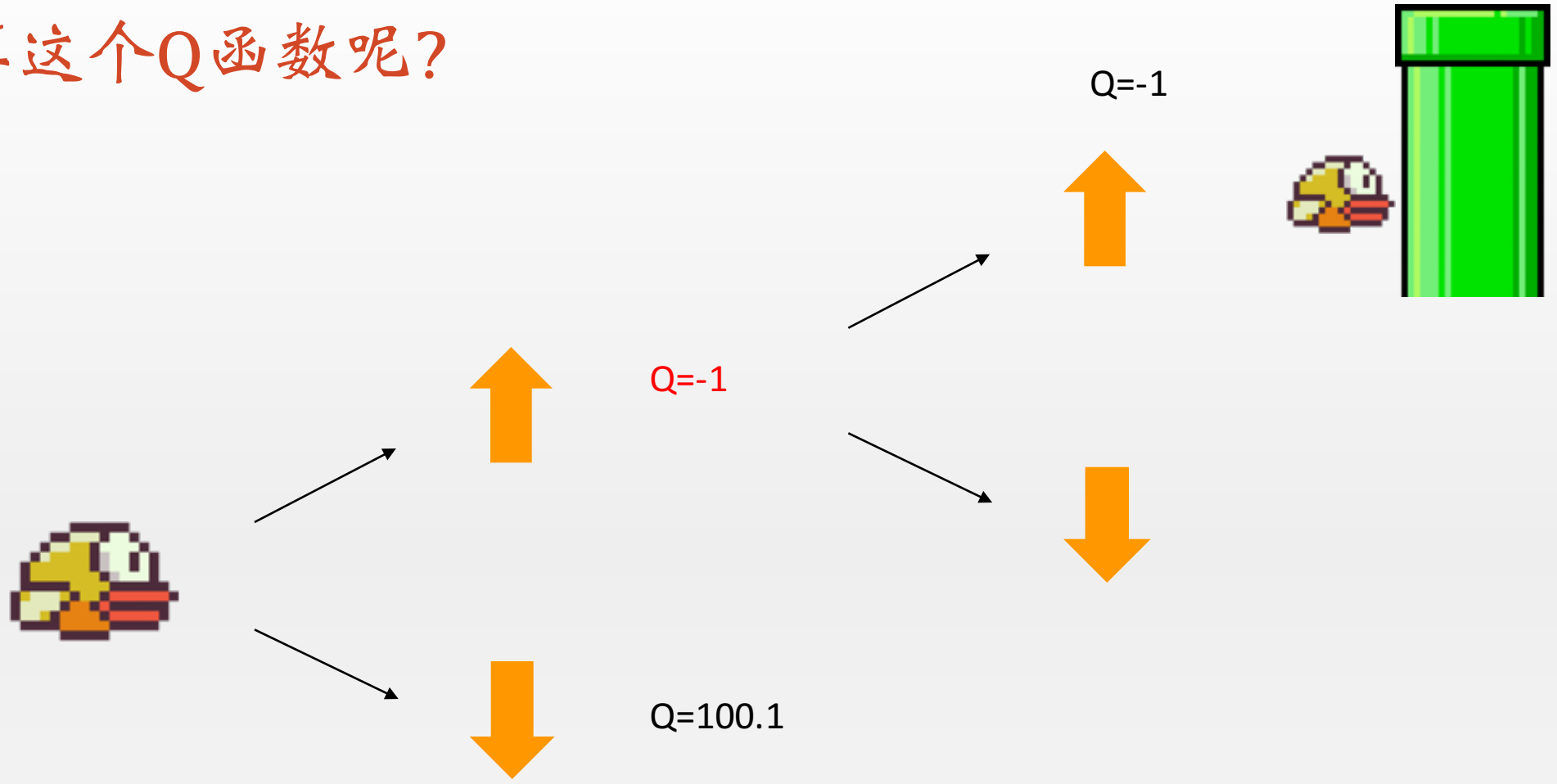
Q是一个s和a的函数，Q(s,a)指导了小鸟的行动

如何计算这个Q函数呢?



假如根据Q在t步的指导，导致了t+1步的未来r=-1的回报

如何计算这个Q函数呢?



那么，t步的Q也应该等于-1，也就是说Q(t)~ r(t)+Q(t+1)

如何计算这个Q函数呢?

$$Q(t) \sim r(t) + Q(t+1)$$

如何计算这个Q函数呢?

$$Q(t) \sim r(t) + Q(t+1)$$

$$Q(s,a) \sim r(t) + Q(s',a')$$

根据游戏画面s,s'以及玩家的行动来决定Q

如何计算这个Q函数呢?

$$Q(t) \sim r(t) + Q(t+1)$$

$$Q(s,a) \sim r(t) + Q(s',a')$$

$$Q(s,a) \sim r(t) + \gamma Q(s',a')$$

遥远的未来对现在的作用必须衰减

如何计算这个Q函数呢?

$$Q(t) \sim r(t) + Q(t+1)$$

$$Q(s,a) \sim r(t) + Q(s',a')$$

$$Q(s,a) \sim r(t) + \gamma Q(s',a')$$

$$Q(s,a) \sim r(t) + \gamma \max_{a'} Q(s',a')$$

现在的Q需要考虑未来最好的情况

Q要尽可能地与未来一致，并且客观地反映出环境的反馈

Q函数呢

$$Q(s,a) \sim r(t) + \gamma \max_{a'} Q(s',a')$$

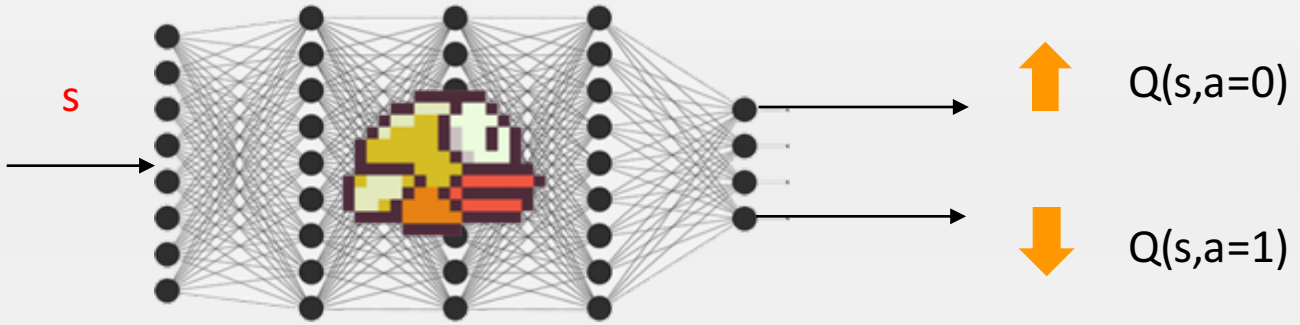
这就是著名的Q-learning算法的思想

Q函数呢

$$Q(s,a) \sim r(t) + \gamma \max_{a'} Q(s',a')$$

如何实现上面的过程呢？

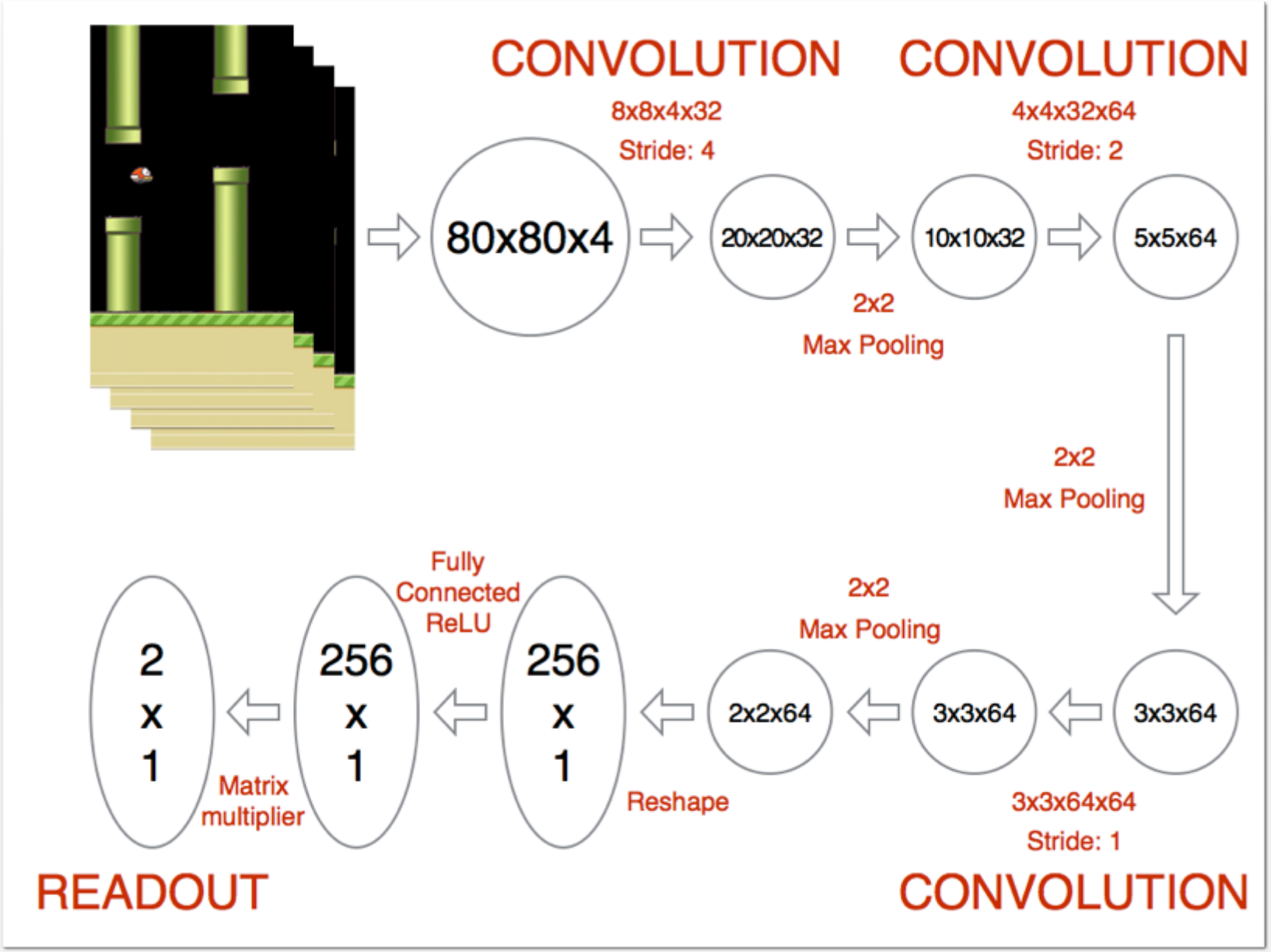
游戏高手：评估函数Q



实际上，我们就是由神经网络实现了一个Q函数

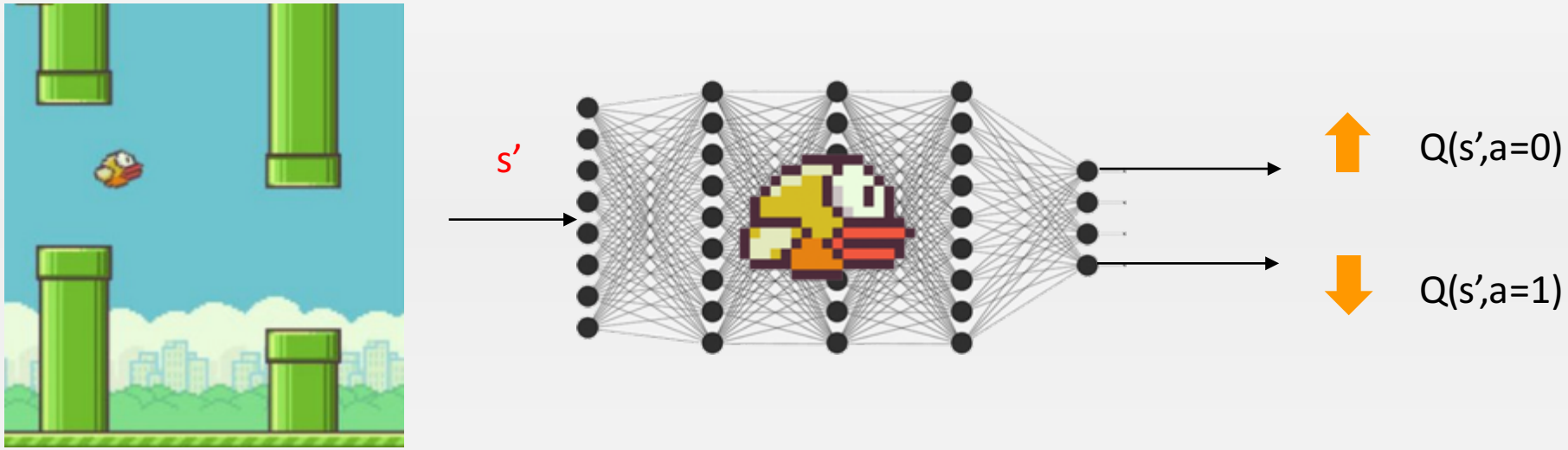
游戏高手：神经网络

yenchenlin



https://github.com/yenchenlin/DeepLearningFlappyBird

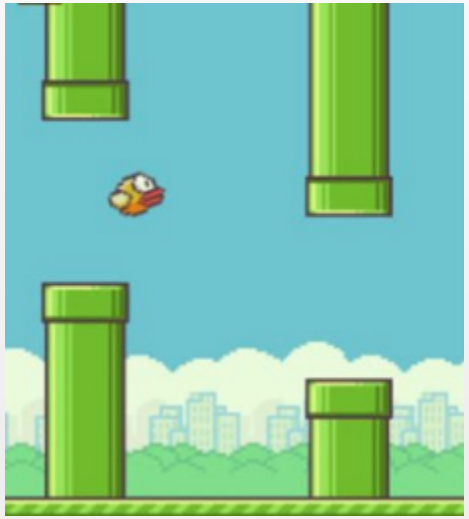
游戏高手：如何计算损失函数？



为了计算Q(t+1)，我们只需要有t+1时刻的s’

游戏高手：如何计算损失函数？

s, Q(s,a)

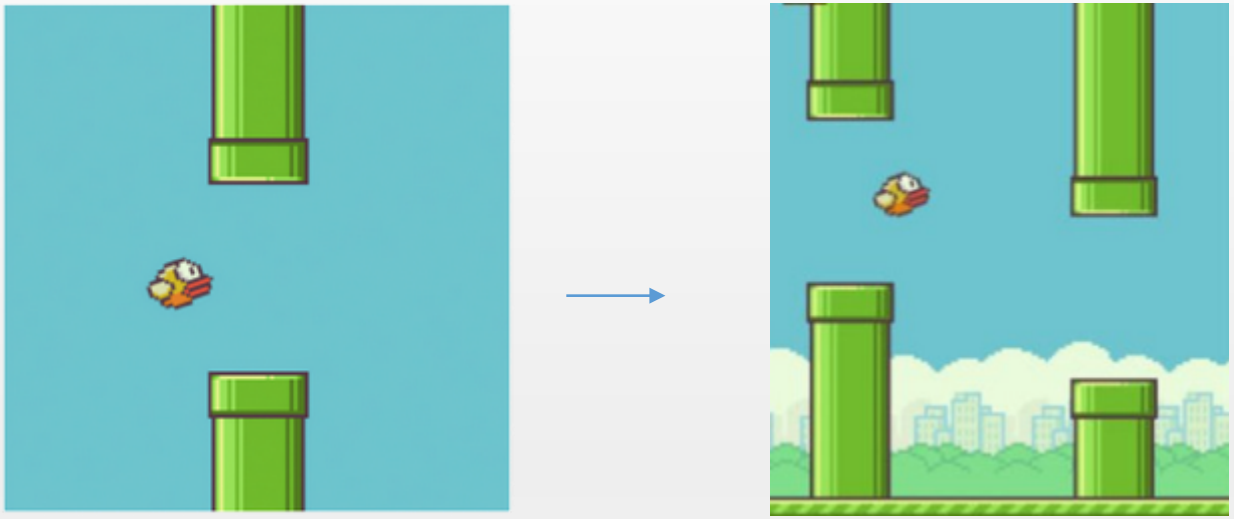


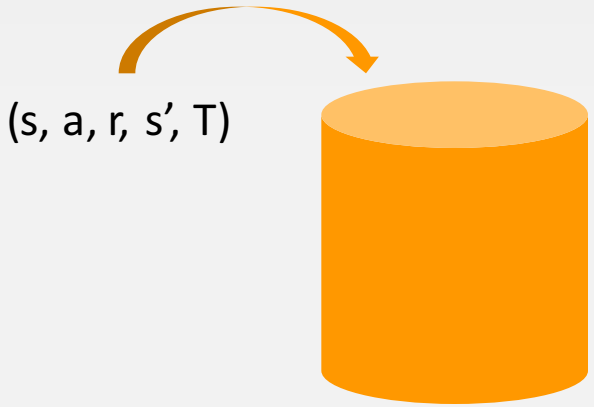
s‘, Q(s’,a’)

$$L(s,a) = \left(\max_{a'} Q(s',a') - Q(s,a)\right)^2$$

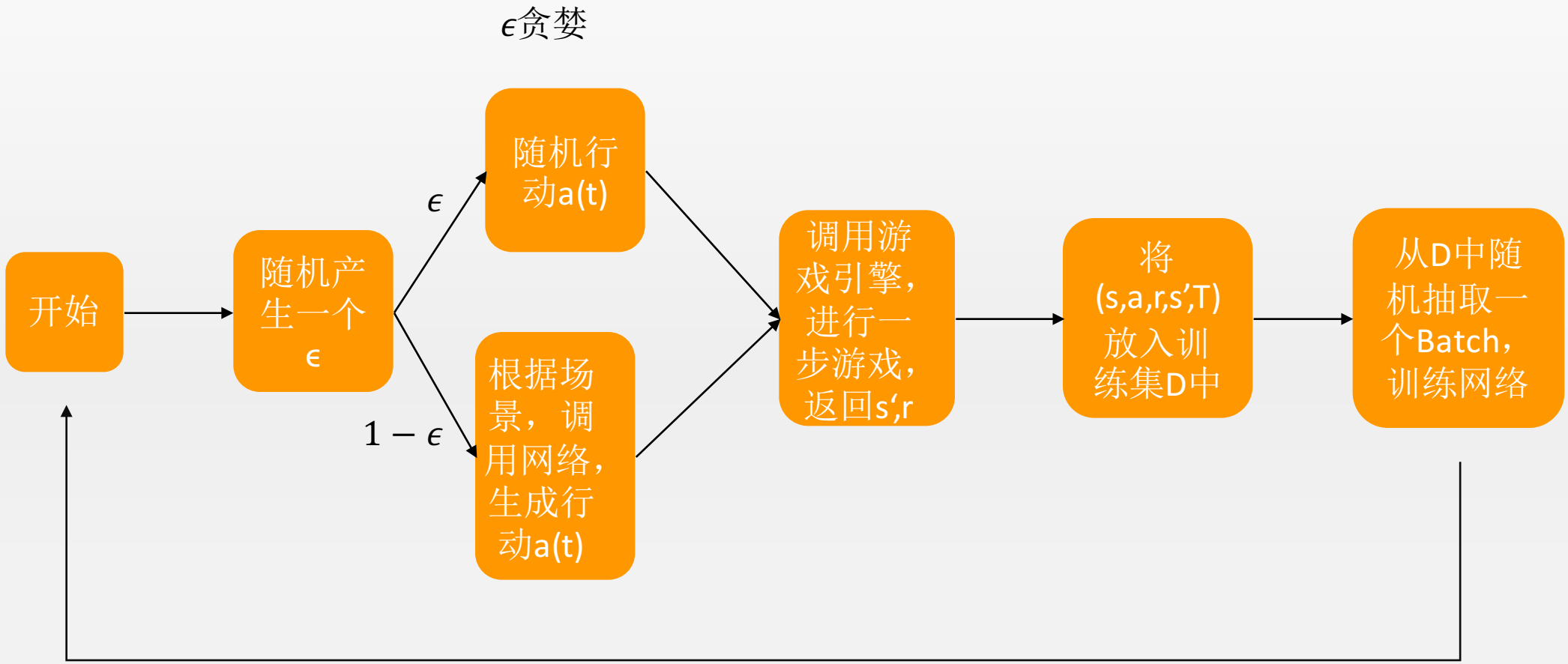
训练：L.backward()

如何形成训练集（miniBatch）？

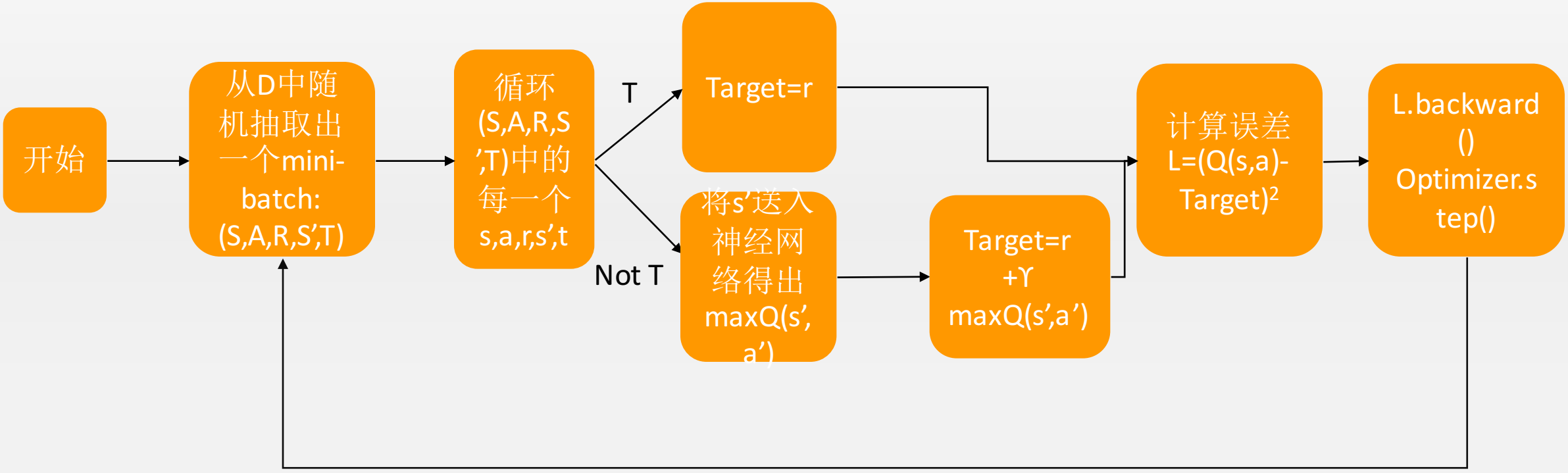




运行流程


ϵ贪婪
开始
随机产生一个ϵ
ϵ
随机行动a(t)
1 − ϵ
根据场景，调用网络，生成行动a(t)
调用游戏引擎，进行一步游戏，返回s‘,r
将(s,a,r,s’,T)放入训练集D中
从D中随机抽取一个Batch，训练网络

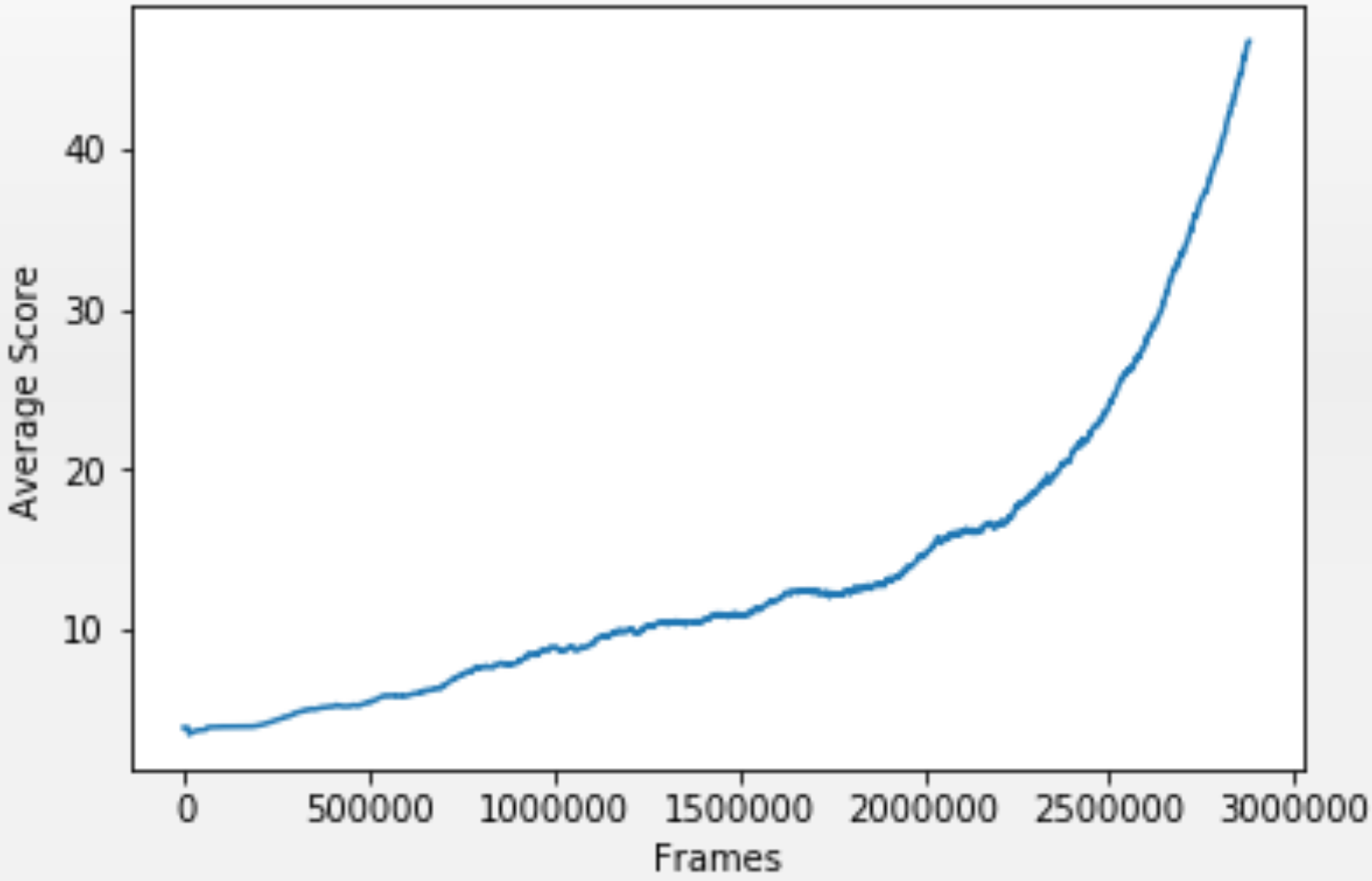
训练流程


开始
从D中随机抽取出一个mini-batch:(S,A,R,S',T)
循环(S,A,R,S',T)中的每一个s,a,r,s',t
T
Not T
Target=r
将s'送入神经网络得出maxQ(s',a')
Target=r+ϒmaxQ(s',a')
计算误差L=(Q(s,a)-Target)²
L.backward()
Optimizer.step()

PyGame

- 一个基于Python的游戏制作环境
- 让多媒体操作（图像、声音）等变得简单
- 完全开源、免费

PyGame


Average Score
40
30
20
10
0
500000
1000000
1500000
2000000
2500000
3000000
Frames

游戏的人工智能

LETTER

doi:10.1038/nature14236

Human-level control through deep reinforcement learning

Volodymyr Mnih[1]*, Koray Kavukcuoglu[1]*, David Silver[1]*, Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

The theory of reinforcement learning provides a normative account[1], deeply rooted in psychological[2] and neuroscientific[3] perspectives on animal behaviour, of how agents may optimize their control of an environment. To use reinforcement learning successfully in situations approaching real-world complexity, however, agents are confronted with a difficult task: they must derive efficient representations of the environment from high-dimensional sensory inputs, and use these to generalize past experience to new situations. Remarkably, humans and other animals seem to solve this problem through a harmonious combination of reinforcement learning and hierarchical sensory processing systems[4,5], the former evidenced by a wealth of neural data revealing notable parallels between the phasic signals emitted by dopaminergic neurons and temporal difference reinforcement learning

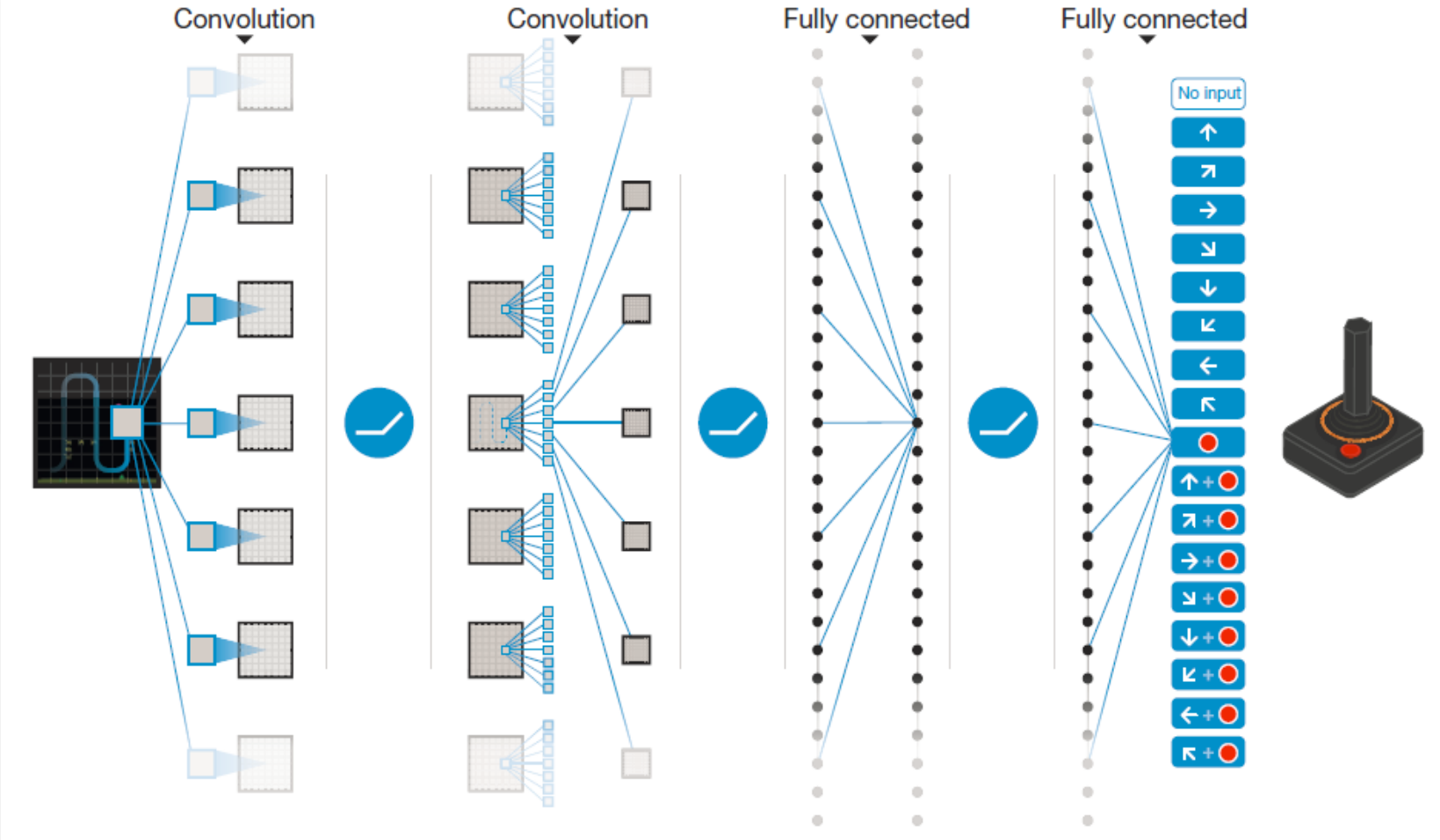
agent is to select actions in a fashion that maximizes cumulative future reward. More formally, we use a deep convolutional neural network to approximate the optimal action-value function

$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots | s_t = s,\, a_t = a,\, \pi\right],$$

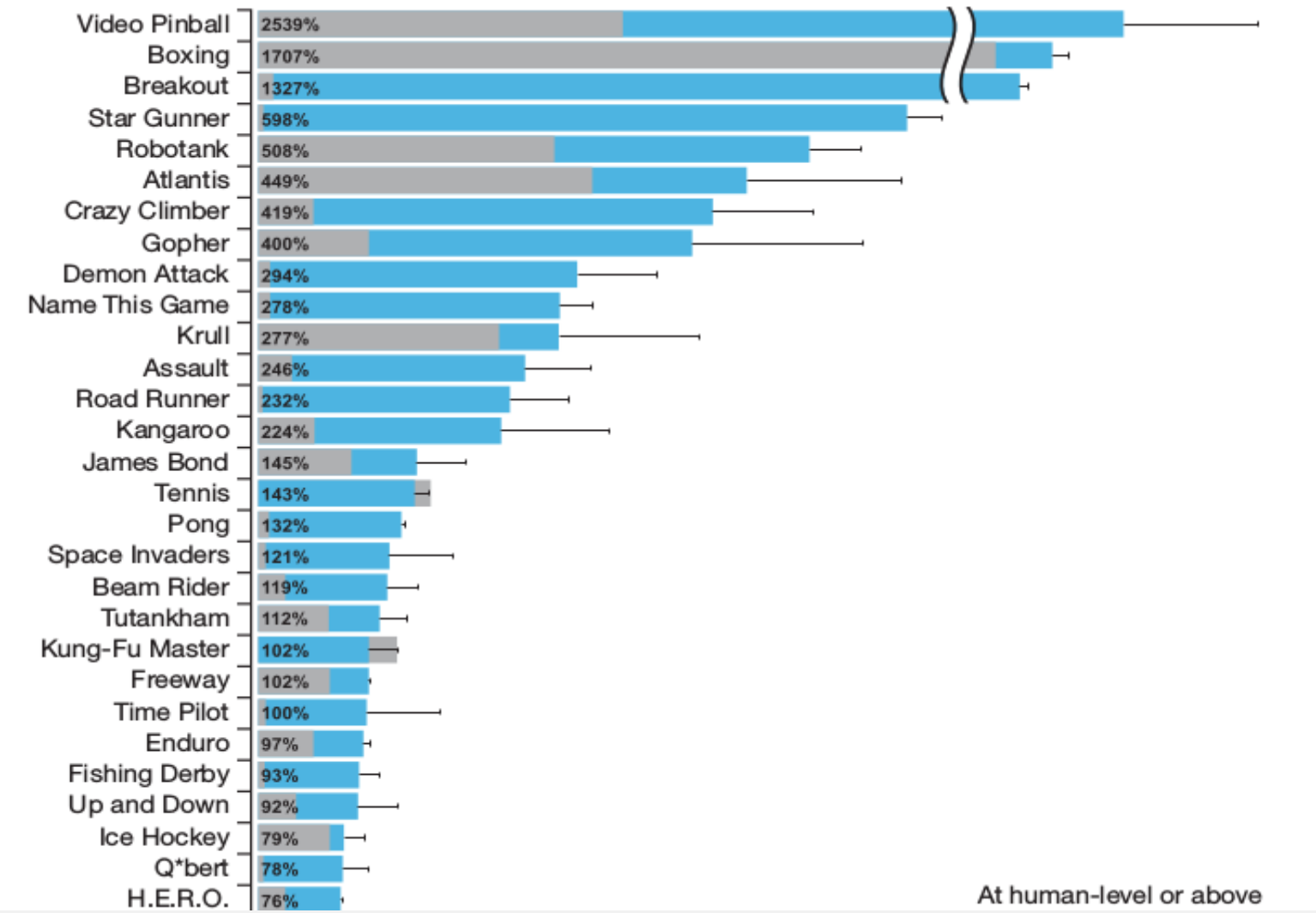
which is the maximum sum of rewards r_t discounted by γ at each time-step t, achievable by a behaviour policy $\pi = P(a|s)$, after making an observation (s) and taking an action (a) (see Methods)[19].

Reinforcement learning is known to be unstable or even to diverge when a nonlinear function approximator such as a neural network is used to represent the action-value (also known as Q) function[20]. This instability has several causes: the correlations present in the sequence of observations, the fact that small updates to Q may significantly change

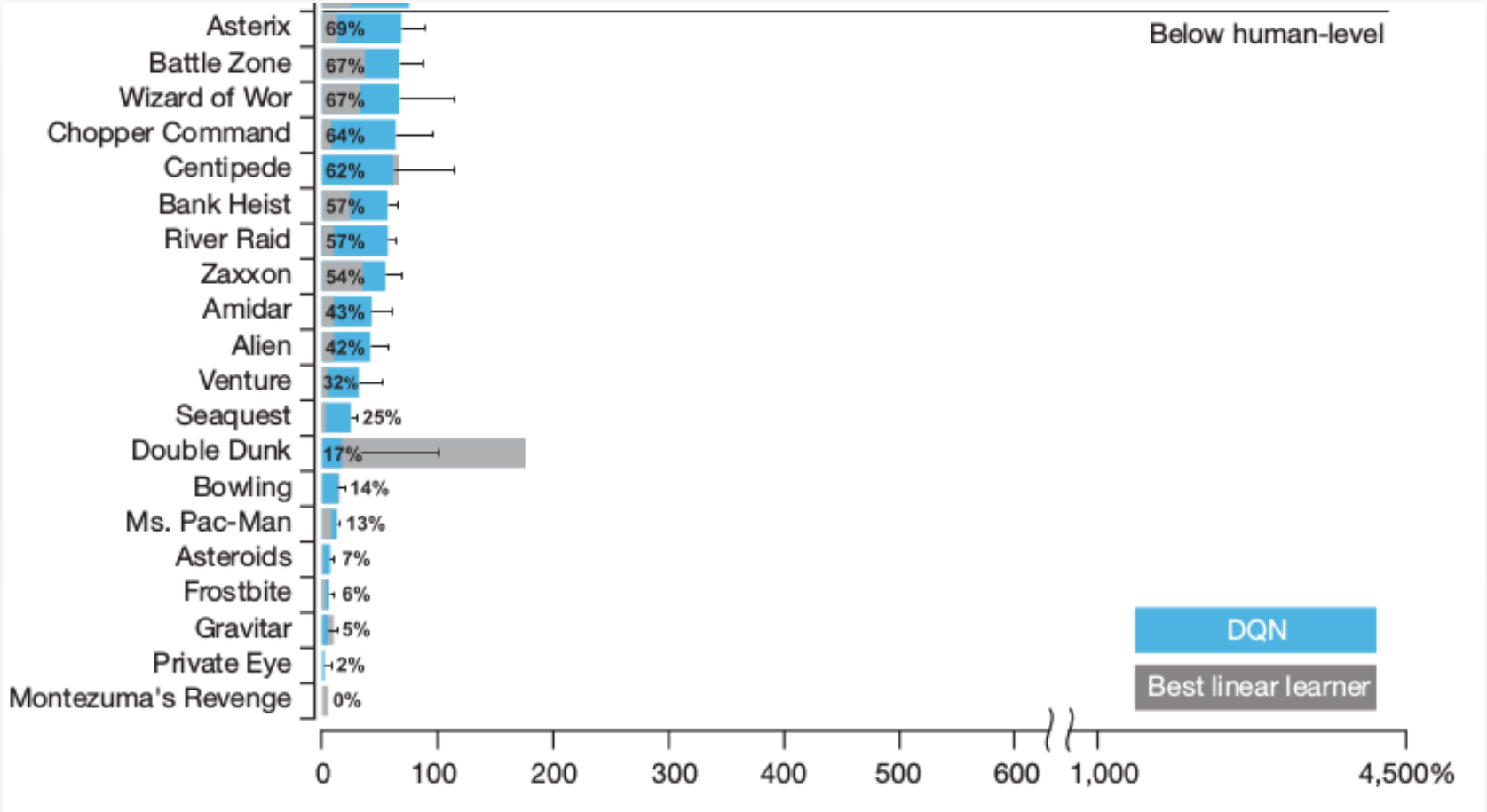
游戏的人工智能


Convolution
Convolution
Fully connected
Fully connected
No input

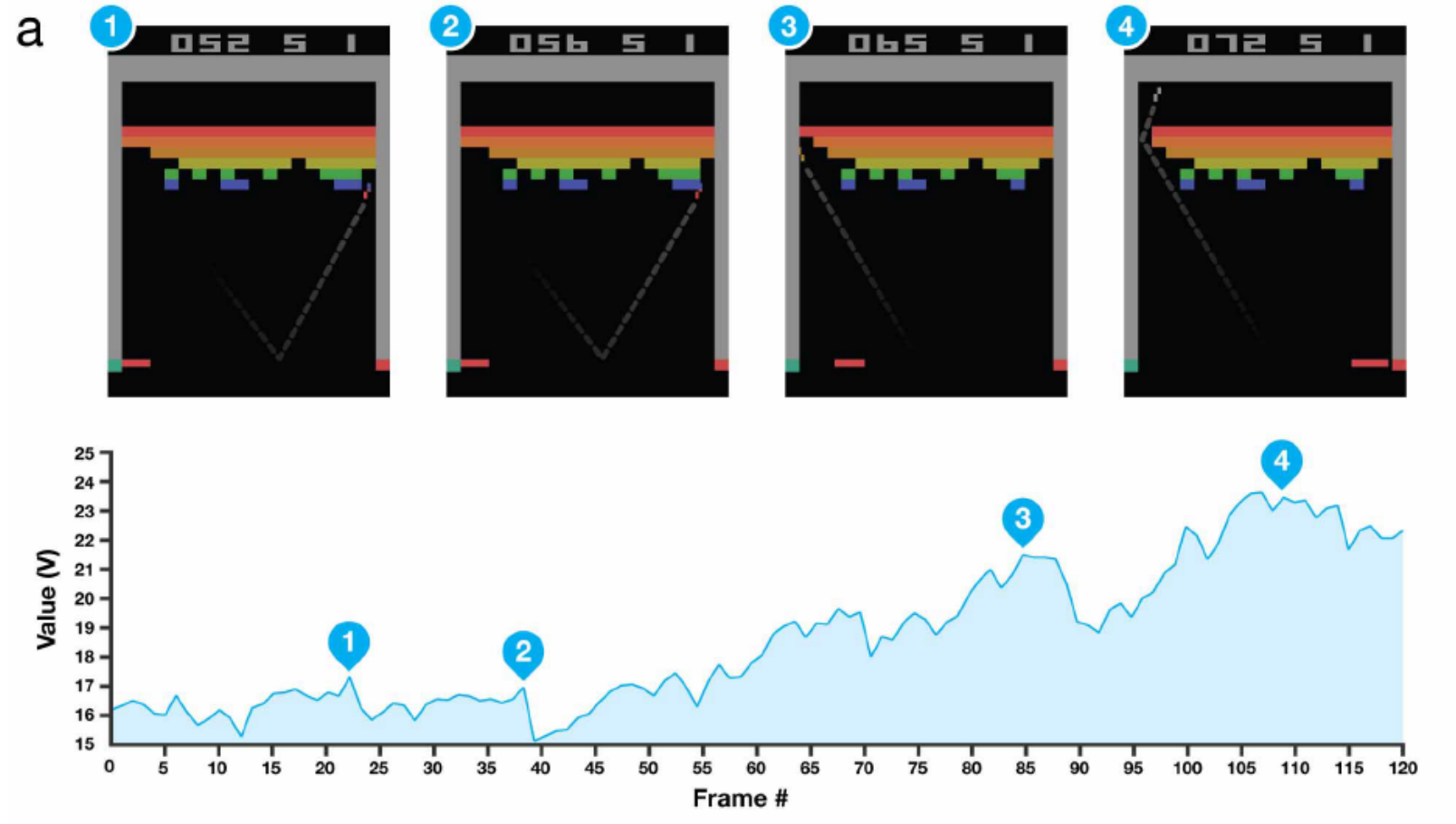
完成效果


Video Pinball
Boxing
Breakout
Star Gunner
Robotank
Atlantis
Crazy Climber
Gopher
Demon Attack
Name This Game
Krull
Assault
Road Runner
Kangaroo
James Bond
Tennis
Pong
Space Invaders
Beam Rider
Tutankham
Kung-Fu Master
Freeway
Time Pilot
Enduro
Fishing Derby
Up and Down
Ice Hockey
Q*bert
H.E.R.O.
2539%
1707%
1327%
598%
508%
449%
419%
400%
294%
278%
277%
246%
232%
224%
145%
143%
132%
121%
119%
112%
102%
102%
100%
97%
93%
92%
79%
78%
76%
At human-level or above

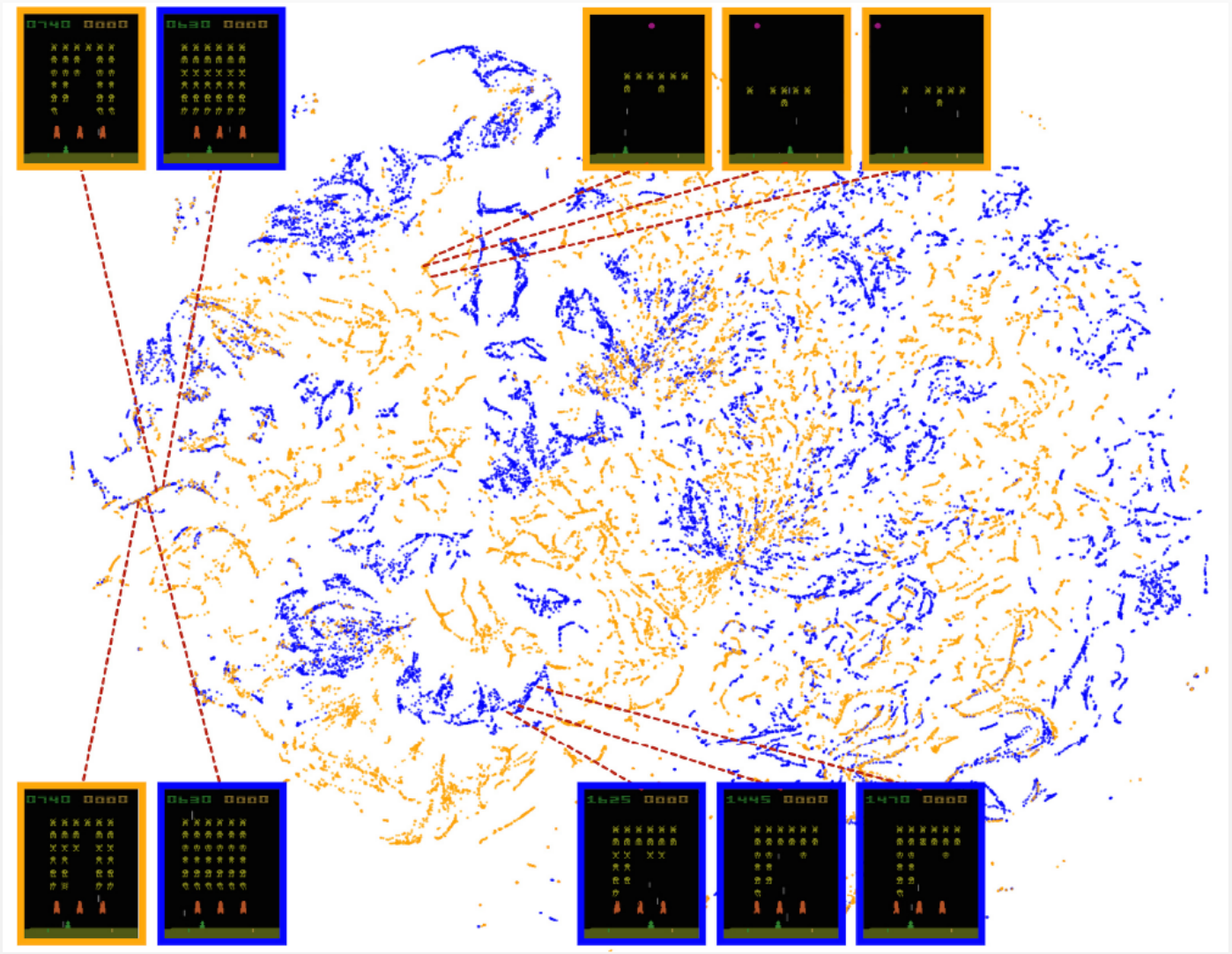
完成效果


Asterix
Battle Zone
Wizard of Wor
Chopper Command
Centipede
Bank Heist
River Raid
Zaxxon
Amidar
Alien
Venture
Seaquest
Double Dunk
Bowling
Ms. Pac-Man
Asteroids
Frostbite
Gravitar
Private Eye
Montezuma's Revenge
69%
67%
67%
64%
62%
57%
57%
54%
43%
42%
32%
25%
17%
14%
13%
7%
6%
5%
2%
0%
Below human-level
DQN
Best linear learner
0
100
200
300
400
500
600
1,000
4,500%

完成效果


a
1
052 5 1
2
056 5 1
3
065 5 1
4
072 5 1
Value (V)
25
24
23
22
21
20
19
18
17
16
15
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
Frame #

完成效果



- 蓝色：人类
- 橙色：机器

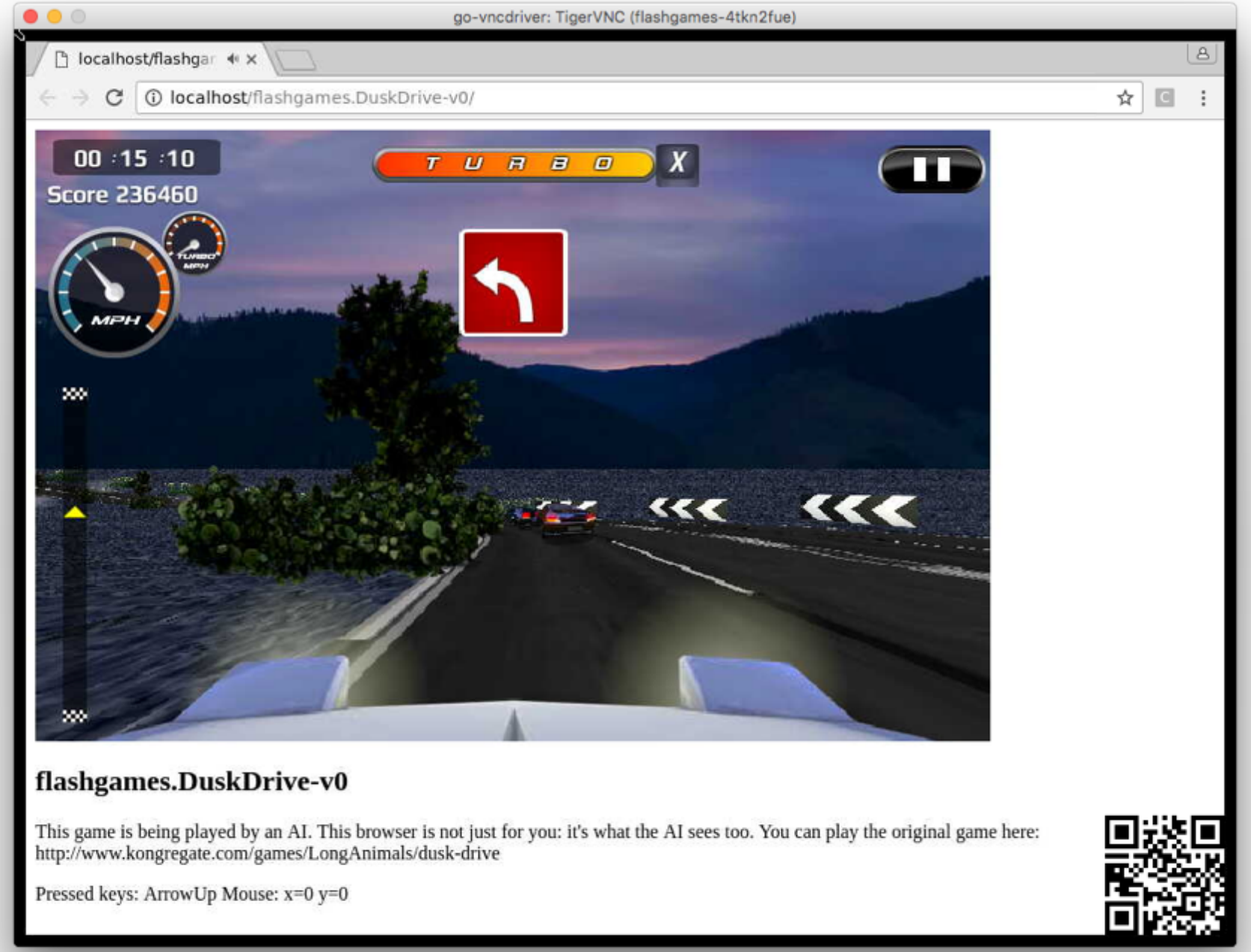
更多游戏AI

https://deepmind.com/blog/deep-reinforcement-learning/

更多游戏AI

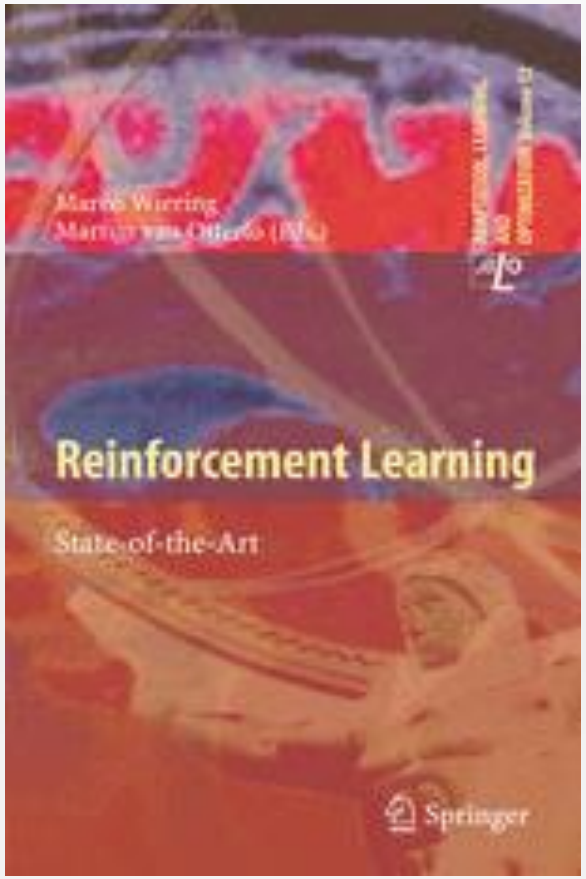

Developing & Comparing RL Algorithms
OpenAI Gym
BETA
A toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Go.

更多游戏AI



OpenAI：
Universe

更多游戏AI



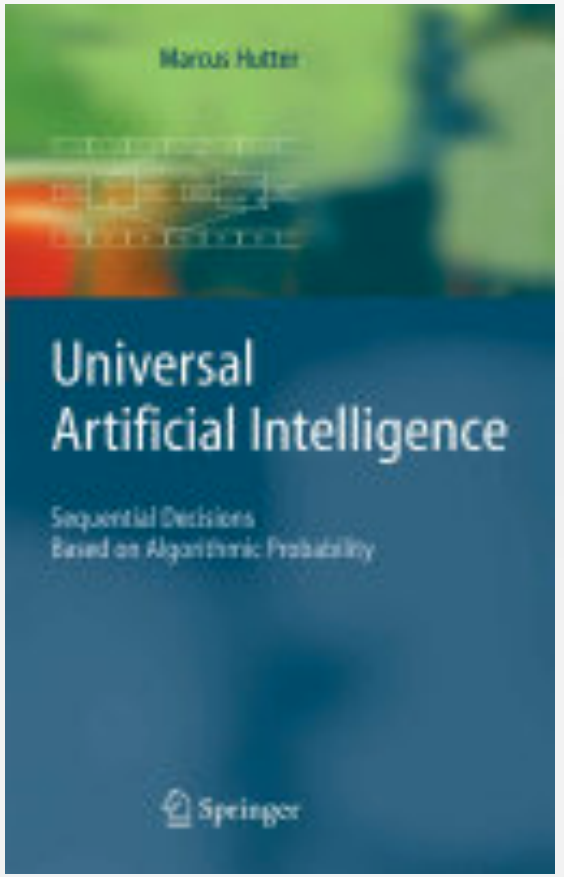
https://link.springer.com/content/pdf/10.1007%2F978-3-642-27645-3.pdf

有关AGI（Artificial General Intelligence)

- **Artificial general intelligence** (**AGI**) is the intelligence of a machine that could successfully perform any intellectual task that a human being can.
- It is a primary goal of some artificial intelligence research and a common topic in science fiction and future studies.
- Artificial general intelligence is also referred to as "**strong AI**",[1] "**full AI**"[2] or as the ability of a machine to perform "general intelligent action".[3] Academic sources reserve "strong AI" to refer to machines capable of experiencing consciousness.
- DQN可以看作游戏世界中的AGI

Universal Artificial Intelligence

Marcus Hutter



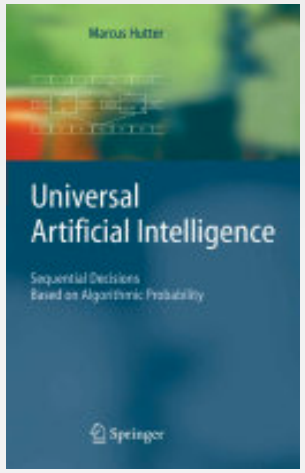
AIXI

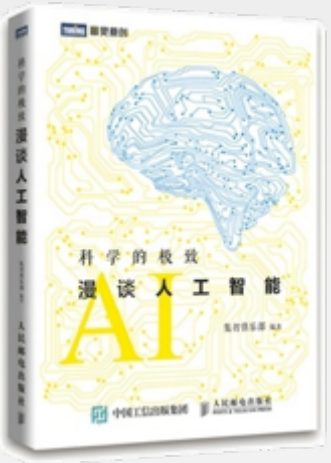
一个公式定义人工智能：

$$\arg\max_{a_t}\sum_{o_t r_t}\ldots\max_{a_m}\sum_{o_m r_m}[r_t+\ldots+r_m]\sum_{q:\,U(q,a_1\ldots a_m)=o_1 r_1\ldots o_m r_m}2^{-\mathrm{length}(q)},$$

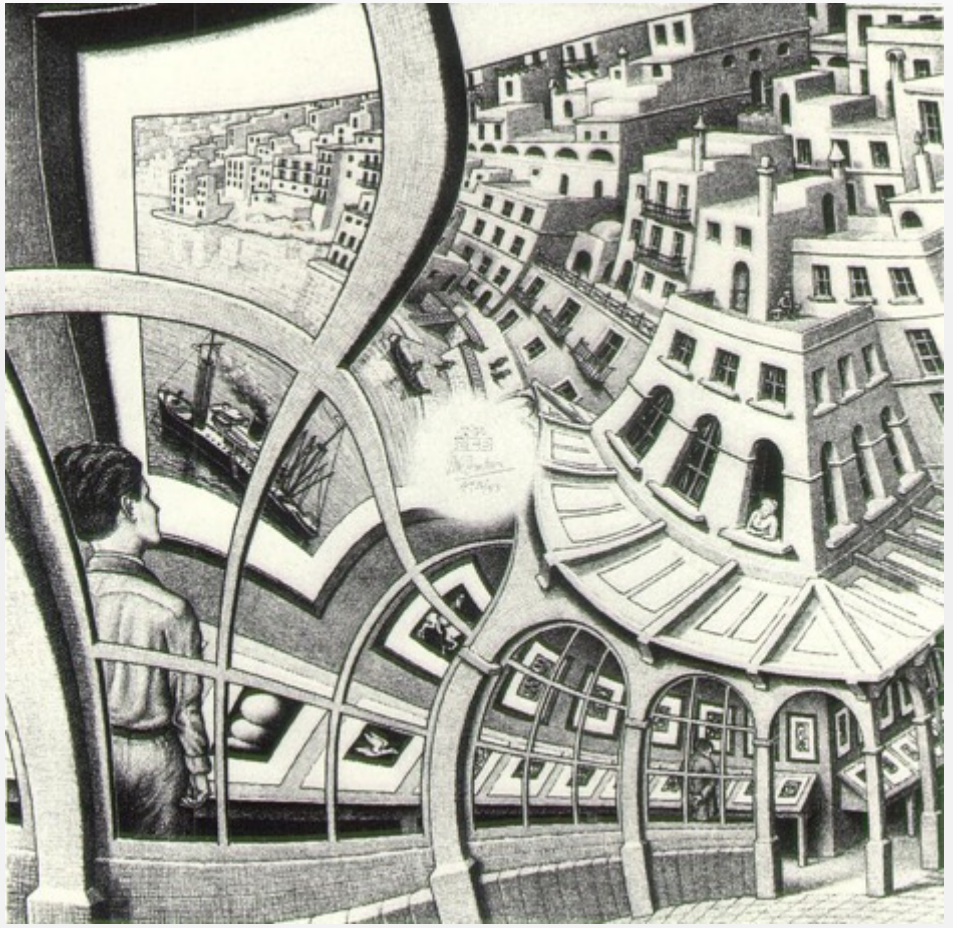
通用归纳
Universal
Induction

最优决策理论
Optimal
Decision
Theory





有关自我意识 (Self-awareness)



递归定理 (Recursion Theorem)

- Kleene第二递归定理

- 对于任意的程序F，总存在一段程序代码c，使得我们执行代码c的结果完全等价于把源代码c作为数据输入给程序F执行的结果。

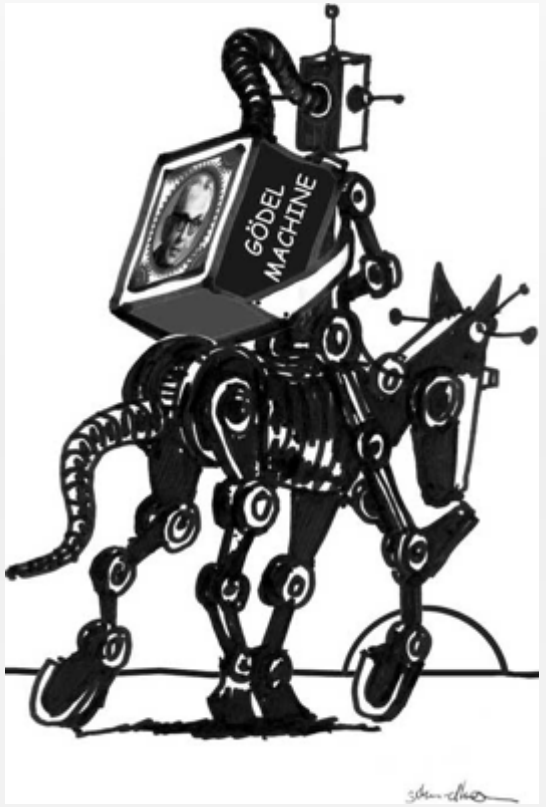
自我反省的程序 (self-introspection)

- 自我意识的核心：自我反省
- 存在着这样一种计算机程序$F_t(x)$，它的作用就是计算任意的源代码为x的程序在经过t时间步的运算后的结果。
- 于是根据递归定理，我们便知道，存在着一个源程序O，它所做的就是：把自己的源代码拿出来，然后在自己的虚拟机上模拟自己运算t时间步后的结果。

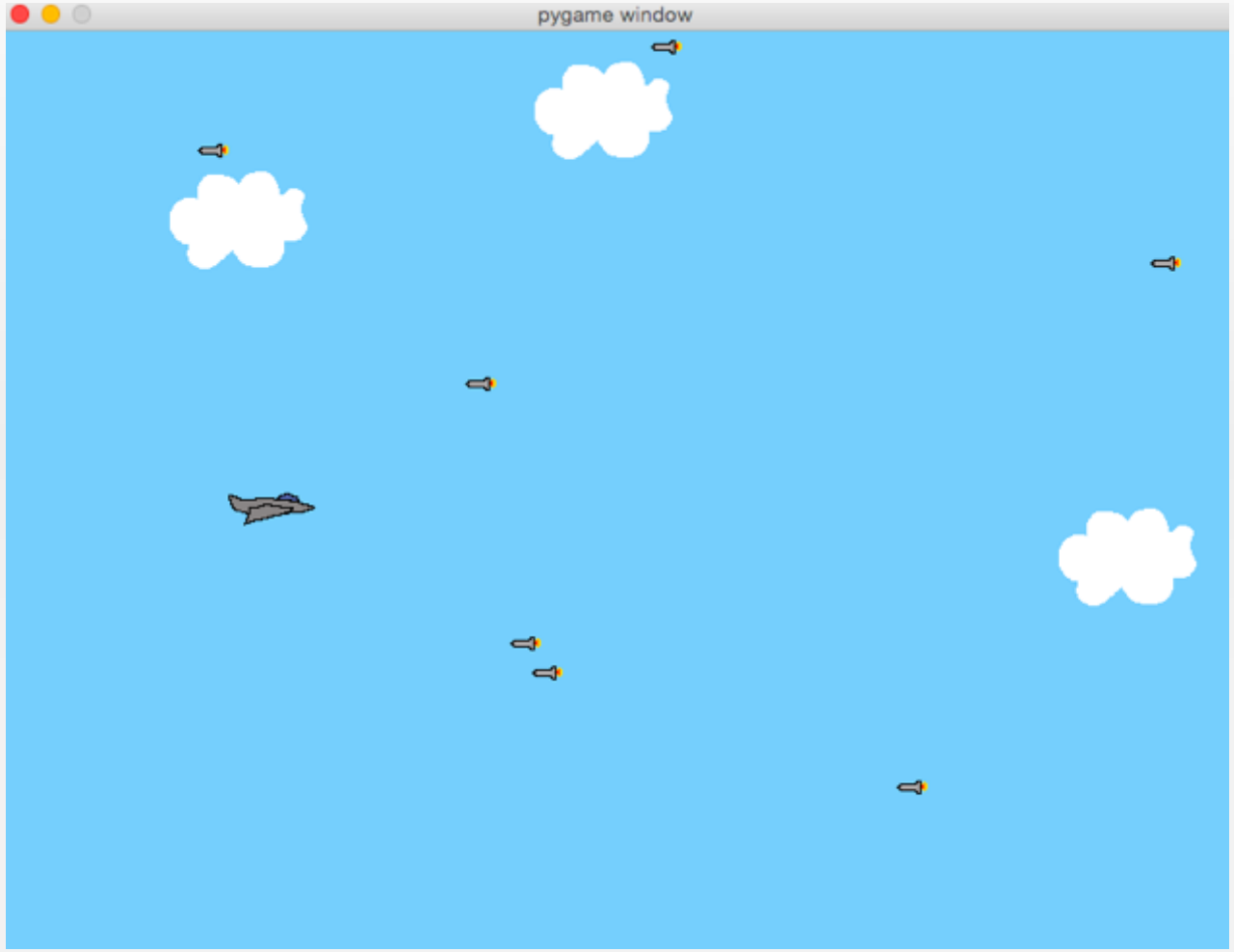
哥德尔机

A **Gödel machine** is an approach to Artificial General Intelligence that uses a recursive self-improvement architecture proposed by Jürgen Schmidhuber. It was inspired by the mathematical theories of Kurt Gödel, where one could always find a mathematical truth or axiom that if attached to a formal system would make it stronger. A Gödel Machine is a universal problem solver that will make provably optimal self-improvements – self-improvements which can be proved to better maximize its utility

Jürgen Schmidhuber



项目：飞机躲炸弹



- 制作一个飞机躲炸弹的游戏：https://realpython.com/blog/python/pygame-a-primer/
- 训练一个AI程序自动玩这个游戏

敬请期待

Swarmer  舞台

集智AI学园
campus.swarma.org