Deep Reinforcement Learning

Vikas Dhiman

November 8, 2022

BF Skinner's Reinforcement Learning for Pigeons

Video

[1]

[1]Image source:bfskinner.org

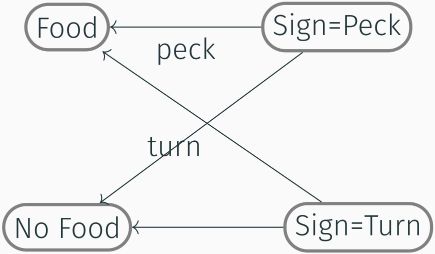
Video

[1]

[1] Image source:bfskinner.org

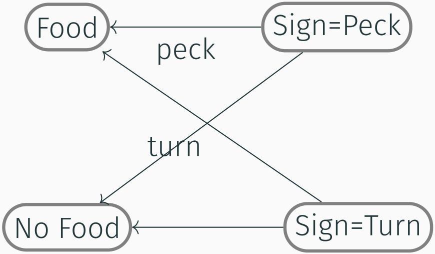
1. BF Skinner demonstrated that pigeons could learn to repeat an action that lead them to a particular reward. [3, p15]

RL terminology



State $(s_t \in \mathcal{S})$

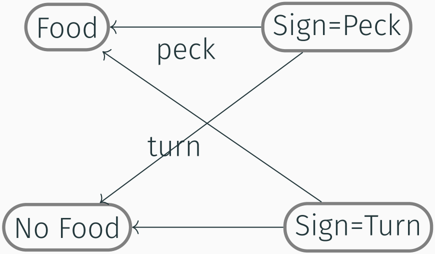
RL terminology



State $(\mathsf{s}_t \in \mathcal{S})$

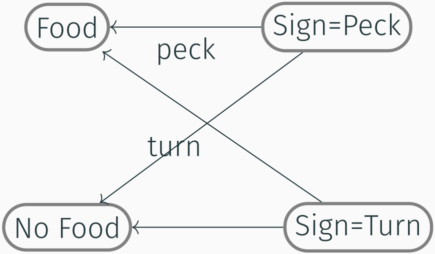
Actions $(\mathsf{a}_t \in \mathcal{A})$

RL terminology



$$\text{State} \quad (s_t \in \mathcal{S})$$
$$\text{Actions} \quad (a_t \in \mathcal{A})$$
$$\text{Transition probabilities} \quad (P(s_{t+1}|s_t, a_t))$$

RL terminology



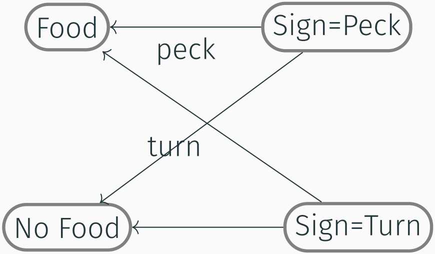
State $(s_t \in \mathcal{S})$

Actions $(a_t \in \mathcal{A})$

Transition probabilities $(P(s_{t+1}|s_t, a_t))$

Rewards $(r_t \sim P(.|s_t, a_t) \in \mathbb{R})$

RL terminology



State $(\mathsf{s}_t \in \mathcal{S})$

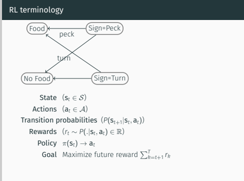
Actions $(\mathsf{a}_t \in \mathcal{A})$

Transition probabilities $(P(\mathsf{s}_{t+1}|\mathsf{s}_t, \mathsf{a}_t))$

Rewards $(r_t \sim P(.|\mathsf{s}_t, \mathsf{a}_t) \in \mathbb{R})$

Policy $\pi(\mathsf{s}_t) \to \mathsf{a}_t$

Goal Maximize future reward $\sum_{k=t+1}^{T} r_k$



State is the full description of the world at time t that captures the entire history. Example: in this example the state can be captured with two bits $\mathbf{s}_t = [f_t; p_t]$, where $f_t \in \{0, 1\}$ describes a food or no food state and $p_t \in \{0, 1\}$ describes the sign showing peck or turn.

Applications of RL

Atari games[1]

Applications of RL

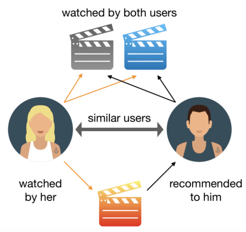
Atari games[1]

Alpha Go[2]

Applications of RL

Atari games[1]

Alpha Go[2]



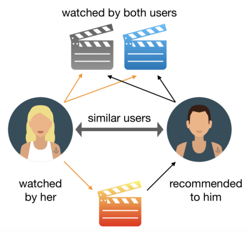
Recommender systems

a

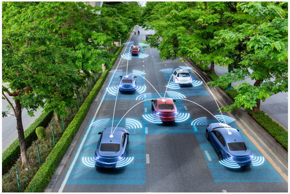
Applications of RL

Atari games[1]

Alpha Go[2]



Recommender systems



Autonomous cars

[a]

[a] Img: towardsdatascience.com

[b] Img:udacity.com

Exercise: Modeling the Breakout game

1. State space?
2. Action space?
3. Rewards?

RL problem

Discount factor $\gamma \in (0,1]$.

$$\pi^*(.) = \arg \max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$$

Value Function

$$V_\pi(\mathsf{s}_k) = \mathbb{E}\left[\sum_{t=k}^{T} \gamma^{t-k} r_t \middle| \mathsf{s}_k\right]$$

Action Value Function

$$V_\pi(\mathsf{s}_k) = \mathbb{E}\left[\sum_{t=k}^{T} \gamma^{t-k} r_t \middle| \mathsf{s}_k\right]$$

$$Q_\pi(\mathsf{s}_k, \mathsf{a}_k) = \mathbb{E}\left[\sum_{t=k+1}^{T} \gamma^{t-k-1} r_t \middle| \mathsf{s}_k, \mathsf{a}_k\right]$$

Advantage Function

$$V_\pi(\mathsf{s}_k) = \mathbb{E}\left[\sum_{t=k}^{T} \gamma^{t-k} r_t \middle| \mathsf{s}_k\right]$$

$$Q_\pi(\mathsf{s}_k, \mathsf{a}_k) = \mathbb{E}\left[\sum_{t=k+1}^{T} \gamma^{t-k-1} r_t \middle| \mathsf{s}_k, \mathsf{a}_k\right]$$

$$A_\pi(\mathsf{s}_k, \mathsf{a}_k) = Q_\pi(\mathsf{s}_k, \mathsf{a}_k) - V_\pi(\mathsf{s}_k)$$

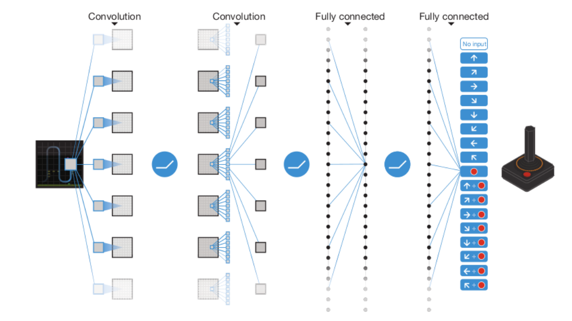
Bellman Equation

$$
\begin{aligned}
V_*(\mathsf{s}_k) &= \mathbb{E}\left[\sum_{t=k}^{T} \gamma^{t-k} r_t \middle| \mathsf{s}_k\right] \\
Q_*(\mathsf{s}_k, \mathsf{a}_k) &= \mathbb{E}\left[\sum_{t=k+1}^{T} \gamma^{t-k-1} r_t \middle| \mathsf{s}_k, \mathsf{a}_k\right] \\
&= \mathbb{E}[r_k + \gamma V_*(\mathsf{s}_{k+1})] \\
&= \mathbb{E}[r_k + \gamma \max_{\mathsf{a}'} Q_*(\mathsf{s}_{k+1}, \mathsf{a}') | \mathsf{s}_k, \mathsf{a}_k]
\end{aligned}
$$

DQN (2013): Bellman equation as a loss function

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim U(D)}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta)\right)^2\right]$$

$$\nabla_\theta L(\theta) = \mathbb{E}_{(s,a,r,s')\sim U(D)}\left[-\nabla_\theta Q(s, a; \theta)\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta)\right)\right]$$



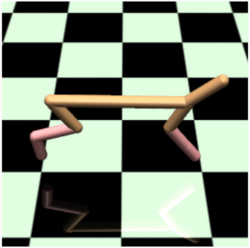
Other design considerations

1. Exploration vs exploitation trade-off
2. Off-policy vs on-policy learning

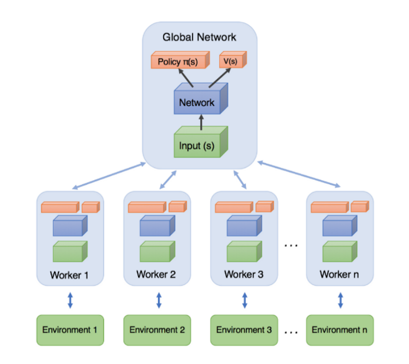
DDPG (2015) : Deep deterministic Policy gradients

$$\nabla_\phi L(\phi) = -\mathbb{E}_{(s,a,r,s')\sim U(D)}[\nabla_a Q(s, a; \theta)\nabla_\phi \pi(s; \phi)]$$

$$\nabla_\theta L(\theta) = \mathbb{E}_{(s,a,r,s')\sim U(D)}\left[-\nabla_\theta Q(s, \pi(s; \theta); \theta)\left(r + \gamma \max_{a'} Q(s', \pi(s'; \phi^-); \theta^-) - Q(s, \pi(s; \phi); \theta)\right)\right]$$



A3C (2016): Async. Advantage Actor Critic



[2]

$$\nabla_\theta L_V(\theta) = \mathbb{E}_{(s,a,r,s')\sim U(D)} \left[-\nabla_\theta V(s;\theta_V)(r + \gamma V(s';\theta_V^-) - V(s;\theta_V)) \right]$$

$$\nabla_\phi L_\pi(\phi) = -\mathbb{E}_{(s,a,r,s')\sim U(D)} \left[\nabla_\phi \log \pi(a|s;\phi) A(s,a;\theta) \right]$$

[2] Img:towardsdatascience.com

My work

References

- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al.
  Human-level control through deep reinforcement learning.
  nature, 518(7540):529–533, 2015.
- David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al.
  Mastering the game of go with deep neural networks and tree search.
  nature, 529(7587):484–489, 2016.
- Richard S Sutton and Andrew G Barto.