DQN

- Q Learning 的局限性：
  (1) discrete action space.
  大小为 $|S| \times |A|$
  (2) discrete → continuous.
  想法：对其进行参数化.
  $Q(s,a) \rightarrow Q_\theta$

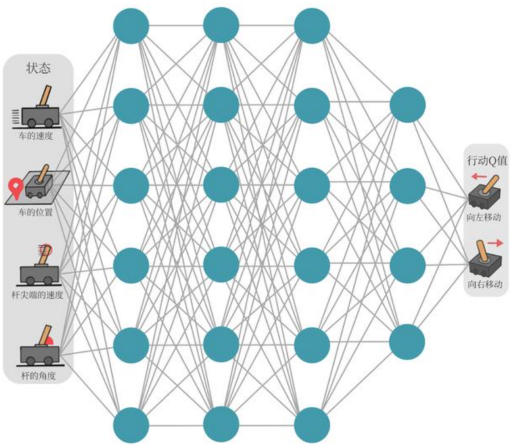
- CartPole 环境 (gym)
  (1) 状态空间：$[x, v_c, \theta, v_p]$ (四维连续向量)
  车的位置 $x \in [-2.4, 2.4]$
  车的速度 $v_c \in (-\infty, +\infty)$
  杆的角度 $\theta \in [-41.8°, 41.8°]$
  杆尖端的速度 $v_p \in (-\infty, +\infty)$
  (2) 动作空间：
  0：左移. 1：右移.
  (3) Reward Function：
  1 帧 → 奖励分数 1.

- DQN：
  (1) 函数拟合的思想：

$Q_w(s,a)$

用神经网络去拟合

给定 s. $\forall a$. / 给定 a. $\forall s$. $\Rightarrow Q_w(s,a)$



(2) 如何给出DQN的损失函数？ A B

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max Q(s',a') - Q(s,a)]$$

头：希望A与B尽量接近.

Loss Function:

$$L(w) = \frac{1}{2N}\sum_{i=1}^{N}[Q_w(s_i,a_i) - (r_i + \gamma \max Q_w(s_i',a_i'))]^2$$

$*$: $\frac{1}{2}$ 只是方便计算.

◦ DQN Tricks:

(1) 经验回放. (experience replay)

① 使样本相互独立

② 提高使用效率

(2) 目标网络. (target network)

更新参数时，目标不能改变. $\Rightarrow$ 两套Q-network

① 训练网络: $Q_w(s,a)$ 计算 $Q_w(s,a)$

② 目标网络: $Q_{w^-}(s,a)$ 计算 $r+\gamma \max Q_{w^-}(s',a')$

并非每一步都更新, 而是C步后 $w^- \leftarrow w$

- 以图像作为输入自身DQN: (e.g. videogame)

利用卷积层提取图像特征.