基本概念

概率论概念

随机变量

概念：一个取决于未知事件的变量，

- 使用大写 X 来表示随机变量

如在抛硬币之前我是不知道硬币结果是什么，但是我知道事件的概率

- 使用小写 x 来表示观测值，只是表示一个数，没有随机性，如下面观测到三次抛硬币的结果
  a. $x_1 = 0$
  b. $x_2 = 1$
  c. $x_3 = 1$

概率密度函数

probability Density Function，PDF

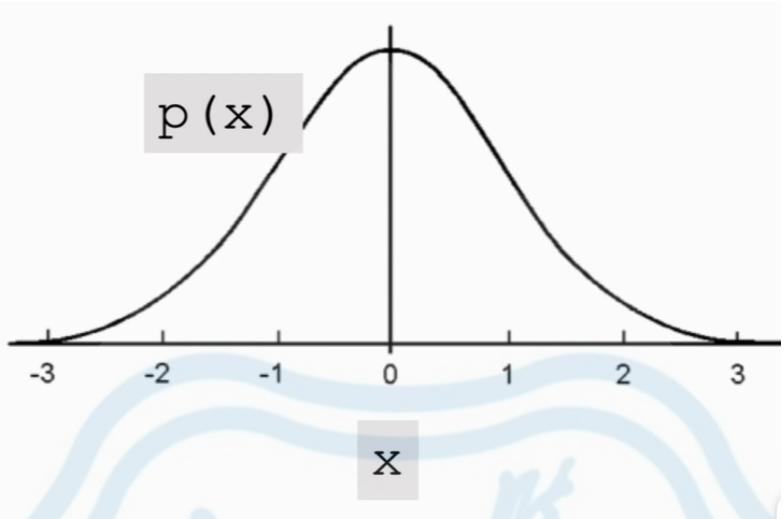
概念：意味着随机变量在某个确定的取值点附近的可能性

连续分布

如高斯分布这个连续分布

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{x-\mu}{2\sigma^2})$$

μ 为均值， σ 为标准差。



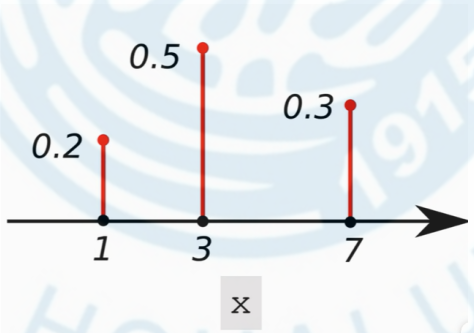
横轴是随机变量 X 取值，纵轴是概率密度，曲线是高斯分布概率密度函数 $P(X)$ ，说明在原点附近概率取值比较大，在原理原点附近概率取值比较小。

离散分布

离散随机变量 $X \in 1, 3, 7$ 。

PDF:

$$p(1) = 0.2, p(3) = 0.5, p(7) = 0.3$$



性质

- 随机变量 X 作用域定义为花体 $\mathcal{X}$
- 如果 X 是连续的变量分布，则可对概率密度函数做定积分，值为1。

$$\int_{\mathcal{X}} p(x)dx = 1$$

- 如果 X 是离散的变量分布，则可对 $p(x)$ 做一个加和，值为1。

$$\sum_{x \in \mathcal{X}} p(x) = 1$$

期望

- 对于作用域 $\mathcal{X}$ 中的随机变量 X
- 对于连续分布，函数 $f(x)$ 的期望为：

$$\mathbb{E}[f(x)] = \int_{\mathcal{X}} p(x) \cdot f(x) dx$$

- 对于离散分布，函数 $f(x)$ 的期望为：

$$\mathbb{E}[f(x)] = \sum_{x \in \mathcal{X}} p(x) \cdot f(x)$$

$p(x)$ 是概率密度函数

随机抽样

- 假设有10个球，2红，5绿，3蓝，随机抽一个球，会抽到哪个球。
- 在抽之前，抽到球的颜色就是个随机变量 X ，有三种可能取值红绿蓝。
- 抽出一个球，是红色，这时候就有了一个观测值。
- 上述过程就叫随机抽样

换一个说法

- 箱子里有很多个球，也不知道有多少个
- 做随机抽样，抽到红色球概率是0.2，绿色球概率是0.5，蓝色球概率是0.3。
- 抽一个球，记录颜色，然后放回去摇匀，重复一百次，这样就有统计意义。

```python
from numpy.random import choice

samples = choice(['R', 'G', 'B'], size=100, p=[0.2, 0.5, 0.3])
print(samples)
```

```
['R' 'G' 'R' 'R' 'R' 'R' 'B' 'B' 'B' 'G' 'G' 'B' 'G' 'B' 'B' 'G' 'B' 'G'
 'B' 'B' 'G' 'B' 'G' 'B' 'B' 'G' 'B' 'B' 'G' 'B' 'G' 'G' 'G' 'G' 'G' 'B'
 'B' 'B' 'B' 'B' 'B' 'G' 'G' 'B' 'R' 'R' 'B' 'R' 'B' 'G' 'R' 'G' 'R' 'G'
 'R' 'R' 'B' 'G' 'G' 'G' 'B' 'R' 'G' 'B' 'G' 'R' 'G' 'G' 'G' 'B' 'B' 'R'
 'G' 'G' 'B' 'B' 'R' 'B' 'B' 'B' 'R' 'B' 'G' 'B' 'R' 'B' 'R' 'G' 'B' 'R'
 'B' 'B' 'G' 'G' 'G' 'R' 'R' 'B' 'R' 'G']
```

强化学习术语

state与action

假设在玩超级玛丽

状态state s 可以表示为当前游戏这一帧的画面

观测到状态后可以做出相应动作action $a \in \{left, right, up\}$

这个例子中马里奥被称为agent，若在自动驾驶中，汽车就被称为agent。动作谁做的就被称为agent

策略policy

$policy\ \pi$ ，指根据观测到的状态，然后做出决策，来控制agent运动。 π 是一个概率密度函数。

- 数学定义： $\pi : (s, a) \mapsto [0, 1]$ ： $\pi(a|s) = \mathbb{P}(A = a|S = s)$
- 意思给定状态 s ，做出动作 a 的概率密度
- 比如给定一个马里奥的运行状态图

  $\pi(left|s) = 0.2$ 向左概率是0.2

  $\pi(right|s) = 0.1$ 向右概率是0.1

  $\pi(up|s) = 0.7$ 向上概率是0.7
- 如果让策略函数自动来操作，它就会做一个随机抽样，0.2的概率向左，0.1的概率向右，0.5的概率向上。
- 强化学习就是学习这个策略函数。
- 给定观测到的状态state $S = s$ ，agent的action A 可以是随机的(最好是随机)

奖励reward

agent做出一个动作，游戏就会给一个奖励，奖励通常需要自己来定义。奖励定义好坏非常影响强化学习结果。

例如在马里奥例子中：

- 马里奥吃到一个金币： $R = +1$ 。
- 赢了这场游戏： $R = +10000$ 。
- 碰到敌人goomba，game over： $R = -10000$ 。
- 啥也没发生： $R = 0$ 。

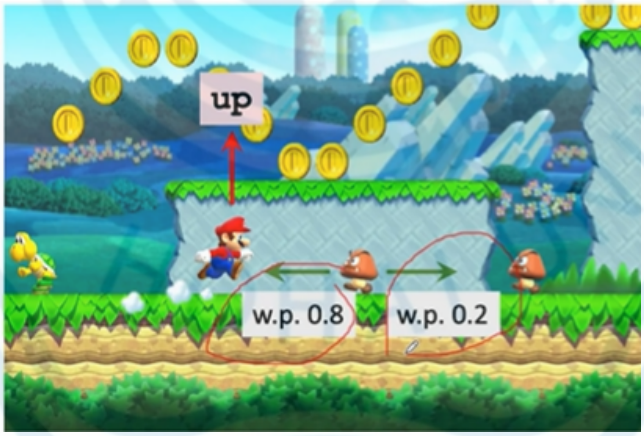
强化学习目标就是奖励获得的总额尽量要高

状态转移 state transition

无法复制加载中的内容

当前状态下，马里奥做一个动作，游戏就会给出一个新的状态。比如马里奥跳一下，屏幕当前帧就不一样了，也就是状态变了。这个过程就叫状态转移。

- 状态转移可以确定的也可以是随机的。
- 状态转移的随机性来自于环境，这里环境就是游戏的程序，程序决定下一个状态是什么。
- **状态转移函数**： $p(s'|s,a) = \mathbb{P}(S' = s'|S = s, A = a)$ 意为观测到当前状态 s 与动作 a ，p 函数输出状态 s' 的概率。



如果马里奥向上跳后，goomba向左和向右的概率分别是0.8和0.2，这个状态转移函数只有环境知道，玩家是不知道的。

交互

agent environment interaction

无法复制加载中的内容

1. 环境告诉Agent一个状态 s_t
2. agent看到状态 s_t 之后，做出一个动作 a_t

无法复制加载中的内容

3. agent做出动作后，环境会更新状态为 s_{t+1} ，同时给出一个奖励 r_t 。

强化学习中的随机性

4. 第一个随机性是从agent动作来的，因为动作是根据 policy 函数随机抽样得来的。
   - $\pi(left|s) = 0.2$
   - $\pi(right|s) = 0.1$
   - $\pi(up|s) = 0.7$

   agent可能做其中任何一个中动作，但动作概率有大有小。
5. 另一个随机性来源是状态转移。
   - 假定agent做出一个动作，那么环境就要生成一个新状态 S' 。
   - 环境用状态转移函数 p 算出概率，然后用概率来随机抽样来得到下一个状态

AI玩游戏

- 观测状态 state s_1
- 做出动作 action a_1
- 观测获取新状态 state s_2 以及奖励 reward r_1
- 做出动作 action a_2
- ...
- 直到打赢游戏或者输掉游戏

这样可以得到一个轨迹 (state, action, reward) trajectory

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_t, a_t, r_t$$

Reward与Return

回报定义

Return 回报，又被称为cumulative future reward，未来的累计奖励

- $U_t = R_t + R_{t+1} + R_{t+2} + \cdots$

把从t时刻开始的奖励全都加起来，一直加到游戏结束的最后一个奖励。

问题： R_t 和 R_{t+1} 同样重要吗

  - 假设右两个选项
    - 立马给你一百块
    - 一年后给你一百块

  一般会选立刻，因为未来的不确定性很大。

  如果改成一年后给你两百块，这时候就会做不同选择。

所以得出以下结论：

- 未来的奖励不如现在的奖励好，应该打一个折扣。
- R_{t+1} 的权重要小于 R_t 。

由于未来奖励不如现在奖励值钱，所以强化学习一般采用 Discounted Return。

Discounted Return 折扣回报，又被称为cumulative discounted future reward

- 折扣率称为 γ ，该值介于0到1之间，是一个超参数。
- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \cdots$

回报中的随机性

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \cdots$

假如游戏已经结束了，所有的奖励都观测到了，那么奖励就都是数值，用小写 r 表示。

如果在 t 时刻游戏还没有结束，这些奖励就还都是随机变量，就用大写字母 R 来表示奖励。

回报 U 依赖于奖励 R ，所以它也是个随机变量，也要用大写字母表示。

随机性有两个来源:

6. 动作是随机的， $\mathbb{P}[A = a|S = s] = \pi(a|s)$
7. 下一个状态是随机的， $\mathbb{P}[S' = s'|S = s, A = a] = p(s'|s, a)$

对于任意时刻的 $i \geq t$ ，奖励 R_i 取决于 S_i 和 A_i ，而回报 U 又是未来奖励的总和。

因此，观测到 t 时刻状态 s_t ，回报 U_t 就依赖于如下随机变量

- $A_t, A_{t+1}, A_{t+2}, \cdots$ 和 $S_{t+1}, S_{t+2}, \cdots$

价值函数

Action-Value Function $Q(s,a)$

Discounted Return 折扣回报，cumulative discounted future reward

- $U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \cdots$

U_t 是个随机变量，在 t 时刻并不知道它的值是什么，那如何评估当前形势？

可以对 U_t 求期望，把里面的随机性都给积掉，得到的就是个实数。打个比方就是抛硬币之前，不知道结果是什么，但知道正反面各有一半的概率，正面记作1，反面记作0，得到的期望就是0.5。

同样对 U_t 求期望，就可以得到一个数，记作 Q_π

这个期望怎么求的?

- 把 U_t 当作未来所有动作 A 和状态 S 的一个函数，未来动作 A 和状态 S 都有一个随机性
- 动作 A 的概率密度函数是策略函数 $\mathbb{P}(A=a|S=s)=\pi(a|s)$
- 状态 S 的概率密度函数是状态转移函数 $\mathbb{P}(S'=s'|S=s,A=a)=p(s'|s,a)$
- 期望就是对这些 A 和 S 求的，把这些随机变量都用积分给积掉，这样除了 S_t 与 A_t ，其余所有的随机变量($A_{t+1},A_{t+2},\cdots$ 和 $S_{t+1},S_{t+2},\cdots$)都被积掉了。
- S_t 与 A_t 被当作被作为观测到的数值来对待，而不是随机变量，所以没有被积分积掉。
- 求期望得到的函数就被称为动作价值函数

动作价值函数

对于策略 π ，动作价值函数定义如下

- $Q_\pi(s_t,a_t)=\mathbb{E}[U_t|S_t=s_t,A_t=a_t]$
  a. Q_π 依赖于当前动作 a_t 与状态 s_t ，还依赖于策略函数 π (积分时会用到它， π 不一样，得到的 Q_π 就不一样)。
  b. 直观意义：如果用策略函数 π ，那么在 s_t 这个状态下做动作 a_t ，是好还是坏。它会给当前状态下每个动作打分，这样就知道哪个动作好那个动作差。

如何去掉 π ？

可以对 Q_π 关于 π 求最大化。意思就是可以有无数种策略函数 π ，但我们要采用最好的那一种策略函数，即让 Q_π 最大化的那个函数。

最优动作价值函数 Optimal action-value Function

- $Q^*(s_t,a_t)=\max_\pi Q_\pi(s_t,a_t)$

状态价值函数

- $V_\pi(s_t) = \mathbb{E}_A[Q_\pi(s_t, A)]$

V_π 是动作价值函数 Q_π 的期望， Q_π 与策略函数 π ,状态 s_t ，动作 a_t 都有关。可以把这里的 A 作为随机变量，关于 A 求期望把 A 消掉，这样就只跟 π 与 s 有关。

其直观意义在于告诉我们当前局势好不好，比如下围棋，当前是快赢了还是快输了。

这里期望是关于随机变量 A 求的，它的概率密度函数是 $\pi(\cdot|s_t)$,根据期望定义可以写成连加或者积分的形式。比如动作是离散的，如上下左右

- $V_\pi(s_t) = \mathbb{E}_A[Q_\pi(s_t, A)] = \sum_a \pi(a|s_t) \cdot Q_\pi(s_t, a)$ 这里动作是离散的。

如果动作是连续的，如方向盘角度，从正90度到负90度。

- $V_\pi(s_t) = \mathbb{E}_A[Q_\pi(s_t, A)] = \int \pi(a|s_t) \cdot Q_\pi(s_t, a) da$ 这里动作是连续的

总结

动作价值函数 $Q_\pi(s_t, a_t) = \mathbb{E}[U_t|S_t = s_t, A_t = a_t]$

它跟策略函数 π ，状态 s_t ，动作 a_t 有关，是 U_t 的条件期望。

能告诉我们处于状态 s 时采用动作 a 是否明智，可以给动作 a 打分。

状态价值函数 $V_\pi(s_t) = \mathbb{E}_A[Q_\pi(s_t, A)]$

它是把 Q_π 中把 A 用积分给去掉，这样变量就就只剩状态 s 。它跟策略函数 π ，状态 s_t 有关，跟动作 a_t 无关。

- 能够评价当前局势是好是坏，
- 也能评价策略函数的好坏，如果 π 越好，则 V_π 期望值 $\mathbb{E}_S[V_\pi(S)]$ 越大。

用强化学习打游戏

假设在马里奥游戏中，目标在于尽可能吃金币，避开敌人，通关。如何做?

8. 一种是学习一个策略函数 $\pi(a|s)$,这叫policy based learning 策略学习，然后基于此来控制agent做动作。
   a. 每观测到一个状态 s_t
   b. 把 s_t 作为 $\pi(\cdot|s)$ 函数输入， π 函数输出每一个动作的概率
   c. 随机采样获取动作 a_t
9. 学习最优动作价值函数 $Q^*(s, a)$ ，这叫value based learning 价值学习，它告诉如果处于状态 s，做动作是好还是坏。

a. 每观测到一个状态 s_t

b. 把 s_t 作为 $Q^*(s,a)$ 函数输入，然 $Q^*(s,a)$ 对每一个动作做一个评价，得到每个动作的 Q 值。

c. 选择输出值最大的动作， $a_t = argmax_a Q^*(s_t,a)$

d. 因为 Q 值是对未来奖励总和的期望，如果向上动作 Q 值比其他动作 Q 值要大就说明向上跳的动作会在未来获得更多的奖励。