State - Action value function

Q function

Q(s,a) hay state-action value function là **giá trị kỳ vọng return khi:**

1. Bạn bắt đầu từ trạng thái s,
2. Thực hiện hành động a một lần duy nhất,
3. **Sau đó hành động tối ưu** từ đó trở đi (tức là theo policy tốt nhất).

Q(s,a) không đánh giá hành động đó tốt hay xấu, mà **chỉ báo cho biết tổng return** bạn sẽ nhận được nếu đi theo lộ trình đó.

Từ mỗi trạng thái s, chọn hành động a sao cho:

$$\pi^*(s) = \arg\max_a Q(s,a)$$

- Đây là **chính sách tối ưu** vì nó chọn hành động mang lại tổng phần thưởng kỳ vọng lớn nhất.

Bellman equation

$$Q(s,a) = R(s) + \gamma \cdot maxQ(s\prime, a\prime)$$

Trong đó:

- Q(s, a): giá trị kỳ vọng khi bắt đầu ở trạng thái `s`, thực hiện hành động `a`
- R(s): phần thưởng nhận ngay tại `s`
- γ: hệ số chiết khấu (*discount factor*, 0 < γ ≤ 1)
- s′: trạng thái kế tiếp sau khi thực hiện hành động `a` tại `s`
- a′: hành động kế tiếp có thể chọn ở `s'`
- $\max_{a'} Q(s', a')$: giá trị tốt nhất nếu hành động tối ưu từ `s'`

Reinforcement Learning với Môi trường Ngẫu nhiên (Stochastic MDP)

Trong thực tế, khi bạn ra lệnh cho robot thực hiện hành động, **kết quả không phải lúc nào cũng chính xác.** Điều này dẫn đến một **môi trường không xác định (stochastic)**, nơi hành động có thể dẫn đến **nhiều kết quả khác nhau với xác suất khác nhau**.

Kỳ vọng phần thưởng (Expected Return)

- Do phần thưởng trở thành **ngẫu nhiên**, ta không thể tối ưu một giá trị cố định.
- Mục tiêu RL trở thành tối đa hóa phần thưởng kỳ vọng (expected return)

$$\mathbb{E}[R_1 + \gamma R_2 + \gamma^2 R_3 + \dots]$$

- Trong đó $\mathbb{E}$ là **trung bình qua nhiều lần chạy (expected value)**.

Phương trình Bellman trong môi trường ngẫu nhiên:

$$Q(s, a) = R(s) + \gamma \cdot \mathbb{E}_{s'} \left[\max_{a'} Q(s', a') \right]$$

Trong đó:

- $\mathbb{E}_{s'}$ là **kỳ vọng theo các trạng thái kế tiếp khả dĩ** (vì `s'` giờ là ngẫu nhiên)