Biểu diễn các Đặc trưng Phụ thuộc Hành động

- **Sự cần thiết của Hàm Giá trị Hành động:** Để chuyển từ học Temporal Difference (TD) sang Sarsa, chúng ta cần các **hàm giá trị hành động**, có nghĩa là biểu diễn đặc trưng phải bao gồm cả hành động bên cạnh trạng thái.
- **Xếp chồng Đặc trưng (Stacking Features):** Một phương pháp phổ biến là **xếp chồng đặc trưng**. Phương pháp này bao gồm việc lặp lại các đặc trưng trạng thái giống nhau cho mỗi hành động có thể có.
    - **Ví dụ:** Nếu có bốn đặc trưng trạng thái và ba hành động, vector đặc trưng sẽ trở thành một vector 12 thành phần (4 đặc trưng/hành động x 3 hành động).
    - **Kích hoạt:** Chỉ các đặc trưng tương ứng với hành động đang được xem xét mới "hoạt động" (khác không), trong khi các đặc trưng cho các hành động khác được đặt bằng không.
- **Tính toán Giá trị Hành động:** Để có được giá trị hành động cho một trạng thái và hành động cụ thể, bạn trích xuất phân đoạn vector trọng số tương ứng với hành động đó và tính **tích vô hướng** với phân đoạn vector đặc trưng liên quan (đã kích hoạt).
- **Tính tổng quát vượt ra ngoài xấp xỉ tuyến tính:** Khái niệm xếp chồng đặc trưng cho các giá trị hành động không chỉ giới hạn ở xấp xỉ hàm tuyến tính.
    - **Mạng Nơ-ron:** Đối với mạng nơ-ron, điều này tương đương với việc có **nhiều đầu ra**, mỗi đầu ra cho một giá trị hành động. Mạng nơ-ron nhập trạng thái, và lớp ẩn cuối cùng tạo ra các đặc trưng trạng thái. Mỗi giá trị hành động sau đó được tính toán từ một tập hợp trọng số độc lập bằng cách sử dụng các đặc trưng trạng thái đó, phản ánh hiệu quả quy trình xếp chồng.
- **Khái quát hóa qua các Hành động:** Để cho phép khái quát hóa qua các hành động, bạn có thể nhập **cả trạng thái và hành động** vào mạng, dẫn đến một đầu ra duy nhất đại diện cho giá trị hành động xấp xỉ cho cặp trạng thái-hành động cụ thể đó. Điều này cũng có thể được áp dụng cho các kỹ thuật như mã hóa lát (tile coding).

Sarsa theo từng tập (Episodic Sarsa) với Hàm Xấp xỉ

- **Thuật toán Sarsa theo từng tập** để điều khiển với hàm xấp xỉ khá giống với phiên bản dạng bảng của nó.
- **Điểm khác biệt chính:**
    - Nó sử dụng **các hàm giá trị hành động được tham số hóa** để ước tính giá trị.
    - Quy tắc cập nhật kết hợp **gradient** để điều chỉnh trọng số, tương tự như các phương pháp TD dựa trên gradient.

Expected Sarsa với Hàm Xấp xỉ

- **Khái niệm cơ bản:** Expected Sarsa là một biến thể của Sarsa, trong đó mục tiêu cập nhật sử dụng **kỳ vọng** trên chính sách mục tiêu (target policy), thay vì chỉ lấy giá trị hành động của hành động tiếp theo được chọn.
- **Chuyển đổi từ Sarsa:**

    - Nhắc lại, mục tiêu cập nhật của Sarsa bao gồm giá trị hành động cho trạng thái và hành động tiếp theo.
    - Với Expected Sarsa, thay vào đó, chúng ta tính tổng các giá trị hành động, được nhân với xác suất của chúng theo chính sách mục tiêu.
- **Công thức cập nhật với hàm xấp xỉ:**
    - Tương tự như Sarsa với hàm xấp xỉ, các ước tính giá trị hành động được tham số hóa bởi vector trọng số W.
    - Chúng ta cũng có một thuật ngữ gradient để phân phối lỗi đến các trọng số một cách thích hợp.
    - Điểm khác biệt duy nhất là thay vì sử dụng Q(S′,A′), chúng ta tính toán Q(S′,a) cho *mọi hành động* a có thể có trong trạng thái tiếp theo S′, sau đó tính **kỳ vọng** của các giá trị này dưới chính sách mục tiêu.

Q-learning với Hàm Xấp xỉ

- **Q-learning là trường hợp đặc biệt của Expected Sarsa:** May mắn thay, Q-learning là một trường hợp đặc biệt của Expected Sarsa.
- **Chính sách mục tiêu:** Trong Q-learning, chính sách mục tiêu là **tham lam (greedy)** đối với các giá trị hành động xấp xỉ.
- **Tính toán kỳ vọng dưới chính sách tham lam:** Việc tính toán kỳ vọng dưới chính sách tham lam tương đương với việc tính toán **giá trị hành động tối đa** (maxaQ(S′,a)).
- **Công thức cập nhật:** Do đó, cập nhật Q-learning với hàm xấp xỉ khá đơn giản: chúng ta chỉ cần thay thế thuật ngữ kỳ vọng bằng thuật ngữ **max** (giá trị tối đa).