Kiểm tra và cải thiện các thuật toán tăng cương

Hành động liên tục và rời rạc

# Section 1: Reinforcement Learning Basics

1. Học tăng cường là một loại máy học trong đó tác nhân học cách đưa ra quyết định bằng cách tương tác với một môi trường.

1. Một agent là một thực thể tương tác với môi trường nhằm học cách ra quyết định để tối đa hóa một mục tiêu hay mục đích cụ thể. Agent có thể được xem như một đơn vị ra quyết định tự động nhận vào các thông tin từ môi trường, thực hiện các hành động và nhận được phần thưởng hoặc phạt dựa trên hành động của nó.
2. Môi trường, theo cách nhìn khác, là hệ thống hoặc bối cảnh bên ngoài mà agent hoạt động. Môi trường có thể là một mô phỏng, một hệ thống vật lý hoặc bất kỳ hệ thống nào mà agent tương tác với. Môi trường cung cấp phản hồi cho agent dưới dạng phần thưởng hoặc phạt, mà agent sử dụng để học cách ra quyết định tốt hơn.
3. Agent nhận được phản hồi dưới dạng phần thưởng hoặc phạt và mục tiêu của nó là tối đa hóa tổng phần thưởng nó nhận được theo thời gian.
4. Agent theo quá trình thử và sai của việc thực hiện hành động, quan sát các hậu quả và điều chỉnh hành vi của mình dựa trên các phần thưởng mà nó nhận được. Theo thời gian, agent học cách xác định các hành động dẫn đến phần thưởng cao nhất và tránh những hành động dẫn đến phạt.
5. Các đối tượng học tăng cường có khả năng thích ứng với những tình huống và môi trường mới. Họ có thể điều chỉnh hành vi của mình dựa trên thông tin phản hồi mà họ nhận được và có thể liên tục cải thiện hiệu suất của mình theo thời gian.
6. Các đối tượng học tăng cường có thể học cách tối ưu hóa hành vi của mình theo thời gian, dẫn đến quyết định hiệu quả hơn và hiệu suất tốt hơn.

Các ứng dụng thực tế của học tăng cường: Robot: Học tăng cường có thể được sử dụng để huấn luyện robot thực hiện các nhiệm vụ như nắm lấy đối tượng, điều hướng trong môi trường phức tạp và tương tác với con người. Ví dụ, RL đã được sử dụng để huấn luyện robot chơi bóng bàn, trong đó robot học cách điều chỉnh các chuyển động của mình dựa trên vị trí của quả bóng.

# Section 2: What is Gymnasium?

1. Gymnasium cung cấp một bộ sưu tập các môi trường hoặc "nhiệm vụ" có thể được sử dụng để thử nghiệm và phát triển thuật toán học tăng cường. Các môi trường này thường giống như trò chơi, có các quy tắc được xác định rõ ràng và một cấu trúc thưởng, làm cho chúng hữu ích để đánh giá và so sánh các thuật toán học tăng cường khác nhau.
2. Bộ công cụ Gymnasium bao gồm một tập hợp các giao diện và công cụ để tương tác với các môi trường, chẳng hạn như không gian quan sát, không gian hành động và phần thưởng. Điều này làm cho việc xây dựng và thử nghiệm thuật toán học tăng cường trở nên dễ dàng và được chuẩn hóa.
3. Các môi trường này đa dạng, từ các trò chơi đơn giản như Pong hoặc Breakout đến các mô phỏng phức tạp hơn như robot hoặc lái xe tự động. Các môi trường Gymnasium được thiết kế để sử dụng dễ dàng và đi kèm với một giao diện tiêu chuẩn để tương tác với môi trường.

Section 3: Getting Started with Gymnasium

Để sử dụng Gymnasium để huấn luyện một mô hình Học tăng cường, bạn cần thực hiện các bước sau:

1. Xác định môi trường mà bạn muốn làm việc.
   * Môi trường là các trò chơi, bài toán hay tác vụ được định nghĩa trước với các luật chơi và cấu trúc thưởng (reward) được định nghĩa rõ ràng.
2. Tạo một phiên bản của môi trường đó.
3. Xác định chính sách của đại lý (agent) - tức là làm thế nào để đại lý quyết định hành động nào sẽ được thực hiện.
4. Tương tác với môi trường bằng cách thực hiện các hành động và nhận thưởng.
5. Cập nhật chính sách của đại lý dựa trên các thưởng (rewards) mà nó nhận được.
6. Lặp lại các bước 4 và 5 cho đến khi hiệu suất của đại lý đạt được yêu cầu.

Tóm lại, các bước này cho phép bạn tạo ra và huấn luyện một đại lý Học tăng cường để tự động ra quyết định trong một môi trường với các luật chơi và cấu trúc thưởng được xác định trước đó. Bạn có thể lặp lại quá trình này để cải thiện hiệu suất của đại lý.

# Section 4: Main concepts of Gymnasium

1. Khái niệm về Observation và Action Spaces: Observation space là tập hợp các trạng thái có thể có mà một agent có thể quan sát được trong môi trường. Action space là tập hợp các hành động có thể thực hiện bởi agent trong môi trường.

**Ví dụ** về observation space và action space:

Giả sử bạn muốn huấn luyện một agent để chơi game Flappy Bird, một game đơn giản nhưng phổ biến. Trong game, observation space có thể là một mảng 2D gồm các giá trị pixel trong khung hình hiện tại. Action space có thể chỉ có 2 hành động, đó là nhảy lên hoặc không nhảy.

Trong quá trình huấn luyện, agent sẽ nhận thông tin từ observation space, chọn hành động trong action space và nhận được phần thưởng tương ứng. Nhiệm vụ của agent là học cách chọn hành động để tránh va chạm với các chướng ngại vật và đạt được điểm số cao nhất.

**Ví dụ:** Trong trò chơi Flappy Bird, một episode được bắt đầu từ trạng thái ban đầu khi con chim đang ở vị trí ban đầu của nó và chưa va chạm vào bất kỳ ống nào. Trong mỗi episode, người chơi cố gắng điều khiển con chim bay qua các ống bằng cách nhấn phím để con chim nhảy lên và tránh va chạm với các ống. Episode kết thúc khi con chim va chạm vào một ống hoặc chạm đáy màn hình. Trong quá trình chơi, các trạng thái của con chim được cập nhật liên tục và các điểm số được tính dựa trên số lượng ống đã vượt qua.

1. Khái niệm về Episode: Một episode là một quá trình chạy đầy đủ của một môi trường, bắt đầu từ trạng thái ban đầu và tiếp tục cho đến khi đạt được trạng thái kết thúc. Mỗi episode bao gồm một chuỗi các trạng thái, hành động và phần thưởng.

**Vidu:**Khái niệm về Episode: Một episode là một quá trình chạy đầy đủ của một môi trường, bắt đầu từ trạng thái ban đầu và tiếp tục cho đến khi đạt được trạng thái kết thúc. Mỗi episode bao gồm một chuỗi các trạng thái, hành động và phần thưởng.

1. Khái niệm về Wrapper: Wrapper là một công cụ trong Gymnasium cho phép bạn thay đổi hành vi của một môi trường mà không thay đổi mã nguồn của nó. Wrappers có thể được sử dụng để thêm các tính năng như giới hạn thời gian, định hình phần thưởng và hạn chế hành động.

**vidu**Trong trò chơi Flappy Bird, bạn có thể sử dụng wrapper để thêm một tính năng thời gian giới hạn vào game. Bạn có thể tạo ra một wrapper để giới hạn thời gian mà một episode của game có thể chạy, chẳng hạn như chỉ cho phép chơi trong vòng 1 phút. Điều này sẽ thêm một yếu tố áp lực và khó khăn vào trò chơi, khiến cho việc phải đạt được điểm số cao trở nên khó khăn hơn.

1. Khái niệm về Benchmark: Gymnasium cung cấp một tập hợp các môi trường Benchmark, đó là các nhiệm vụ tiêu chuẩn được sử dụng để đánh giá và so sánh các thuật toán học tăng cường. Những nhiệm vụ Benchmark này bao gồm các nhiệm vụ điều khiển cổ điển, trò chơi Atari và các nhiệm vụ robot.

**Vidu:**Trong trò chơi Flappy Bird, một trong những benchmark environment có thể là đạt được số điểm cao nhất trong một số lần chơi liên tiếp. Cụ thể, ta có thể đặt mục tiêu là đạt được số điểm trung bình cao nhất sau 10 lần chơi liên tiếp với một thuật toán học tăng cường nào đó. Sau đó, ta có thể so sánh hiệu suất của các thuật toán khác nhau trên cùng một benchmark environment và đưa ra kết luận về thuật toán nào có hiệu suất tốt nhất để chơi Flappy Bird.

**Ứng dụng thuật toán Epsilon-greedy để giải quyết Blackjack**

1. Chiến lược epsilon-greedy là một phương pháp để giải quyết bài toán đưa ra hành động tốt nhất cho một tình huống dựa trên chính sách hiện tại. Theo chiến lược này, đôi khi đại lý sẽ chọn một hành động tốt nhất (với xác suất 1-epsilon) hoặc một hành động ngẫu nhiên (với xác suất epsilon) để tìm kiếm các chính sách mới, có thể đem lại phần thưởng tốt hơn trong tương lai.

**VD:**

Giả sử ta đang chơi Blackjack và ta đang ở trạng thái có tổng giá trị của hai lá bài là 12. Chính sách hiện tại (policy) của ta là ở trạng thái này, nếu tổng giá trị của lá bài của dealer là 4, 5 hoặc 6, ta nên "stand" (không lấy thêm lá bài) và nếu tổng giá trị của lá bài của dealer là 2, 3, 7 hoặc cao hơn, ta nên "hit" (lấy thêm lá bài).

1. Trong trò chơi Blackjack, chiến lược epsilon-greedy có thể được áp dụng để xác định liệu người chơi có nên rút thêm một lá bài hay dừng lại. Tại mỗi bước của trò chơi, đại lý (tức là người chơi) có thể chọn hành động được khuyến nghị bởi chính sách hiện tại hoặc một hành động ngẫu nhiên. Chính sách sẽ được học qua thời gian bằng cách cập nhật ước lượng giá trị hành động cho mỗi cặp trạng thái-hành động dựa trên các phần thưởng nhận được trong trò chơi.

**VD:**

Tuy nhiên, ta muốn giải quyết vấn đề exploration-exploitation trade-off bằng cách sử dụng chiến lược epsilon-greedy. Ta đặt giá trị epsilon = 0.1, có nghĩa là với xác suất 0.9 ta sẽ lấy action được đề xuất bởi policy hiện tại (ở đây là "stand") và với xác suất 0.1 ta sẽ lấy action ngẫu nhiên ("hit" hoặc "stand").

1. Khi trò chơi được chơi nhiều lần, đại lý sẽ học được chính sách tối ưu nhằm tối đa hóa phần thưởng trung bình. Ban đầu, đại lý có thể khám phá bằng cách thực hiện các hành động ngẫu nhiên để tìm kiếm các chiến lược mới. Tuy nhiên, khi trò chơi diễn ra, đại lý sẽ bắt đầu tận dụng chính sách tốt nhất đã biết, nhằm tối đa hóa phần thưởng trung bình trong thời gian dài.

**VD:**

Nếu ta quyết định lấy action ngẫu nhiên, và kết quả là lấy "hit", ta sẽ cộng thêm lá bài mới vào tổng giá trị của hai lá bài, vì vậy tổng giá trị của hai lá bài của ta sẽ tăng lên. Sau đó, ta sẽ tiếp tục thực hiện các bước tiếp theo của game (ví dụ: dealer lấy thêm lá bài, kiểm tra kết quả) và nhận được các reward tương ứng. Nếu sau một số lượt chơi, ta thấy rằng việc lấy "hit" trong trường hợp này thường dẫn đến kết quả không tốt, ta có thể cập nhật policy của mình để luôn lấy action "stand" trong trường hợp này để tối đa hóa expected reward.