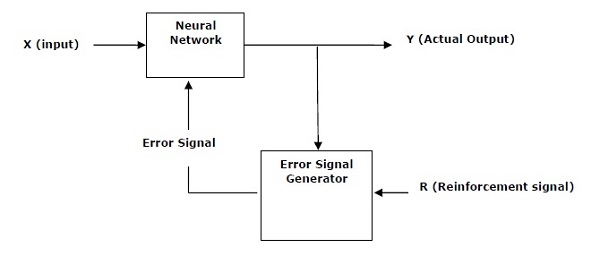
**Bài 21: Reinforcement Learning - Lập trình AI bằng Python**

Trong bài này, chúng ta cùng tìm hiểu chi tiết về các khái niệm học củng cố trong AI với Python.

**1. Kiến thức cơ bản về học củng cố**

Loại hình học tập này được sử dụng để củng cố hoặc tăng cường mạng lưới dựa trên thông tin phê bình. Đó là, một mạng lưới đang được đào tạo theo phương pháp học tăng cường, nhận được một số phản hồi từ môi trường. Tuy nhiên, phản hồi chỉ mang tính đánh giá chứ không mang tính hướng dẫn như trong trường hợp học có giám sát. Dựa trên phản hồi này, mạng thực hiện các điều chỉnh trọng số để thu được thông tin phê bình tốt hơn trong tương lai.

Quá trình học tập này tương tự như học tập có giám sát nhưng chúng ta có thể có rất ít thông tin. Hình sau cho thấy sơ đồ khối của học củng cố



**2. Xây dựng blocks: Môi trường và Tác nhân**

Môi trường và Tác nhân là các khối xây dựng chính của việc học tăng cường trong AI. Phần này thảo luận chi tiết về chúng -

**a. Tác nhân :**

Tác nhân là bất cứ thứ gì có thể nhận biết môi trường của nó thông qua các cảm biến và tác động lên môi trường đó thông qua các tác nhân.

* Tác nhân của con người có các cơ quan cảm giác như mắt, tai, mũi, lưỡi và da song song với các cảm biến, và các cơ quan khác như tay, chân, miệng đối với các tác nhân.
* Một tác nhân robot thay thế máy ảnh và công cụ tìm phạm vi hồng ngoại cho các cảm biến cũng như các động cơ và bộ truyền động khác nhau cho các hiệu ứng.
* Tác nhân phần mềm có các chuỗi bit được mã hóa làm chương trình và hành động của nó

**Các thuật ngữ về tác nhân :**

Các thuật ngữ sau được sử dụng thường xuyên hơn trong học tập củng cố trong AI

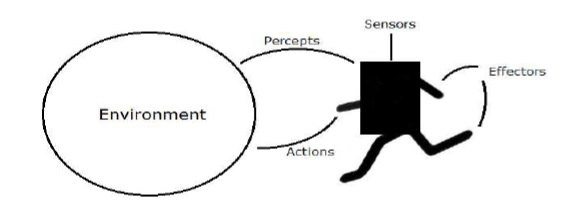
* **Performance Measure of Agent** : Đây là tiêu chí quyết định mức độ thành công của một agent
* **Hành vi của Tác nhân** - Là hành động mà tác nhân thực hiện sau bất kỳ chuỗi khái niệm nhất định nào
* **Percept** - Đây là đầu vào tri giác của tác nhân tại một trường hợp cụ thể.
* **Trình tự nhận thức**: Đó là lịch sử của tất cả những gì một tác nhân đã nhận thức được cho đến nay.
* **Agent Function**: Nó là một bản đồ từ trình tự giới hạn đến một hành động.

**b. Môi trường :**

Một số chương trình hoạt động trong một môi trường hoàn toàn nhân tạo chỉ giới hạn ở đầu vào bàn phím, cơ sở dữ liệu, hệ thống tệp máy tính và đầu ra ký tự trên màn hình.

Ngược lại, một số tác nhân phần mềm, chẳng hạn như rô bốt phần mềm hoặc rô bốt phần mềm, tồn tại trong các miền softbot phong phú và không giới hạn. Trình mô phỏng có một môi trường rất chi tiết và phức tạp. Tác nhân phần mềm cần chọn từ một loạt các hành động trong thời gian thực.

Ví dụ: một softbot được thiết kế để quét các sở thích trực tuyến của khách hàng và hiển thị các mặt hàng thú vị cho khách hàng hoạt động trong môi trường thực cũng như nhân tạo.



**3. Xây dựng môi trường với Python**

Đối với tác nhân học tập củng cố xây dựng, chúng tôi sẽ sử dụng gói OpenAI Gym có thể được cài đặt như sau :

**pip** install gym

Có nhiều môi trường khác nhau trong phòng tập thể dục OpenAI có thể được sử dụng cho các mục đích khác nhau. Một số ít trong số đó là Cartpole-v0, Hopper-v1 và MsPacman-v0. Họ yêu cầu các động cơ khác nhau

Đoạn code sau đây cho thấy một ví dụ về mã Python cho môi trường cartpole-v0

**import** gym

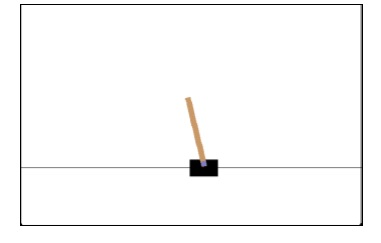
env = gym.make('CartPole-v0')

env.reset()

**for** \_ in **range**(1000):

env.render()

env.step(env.action\_space.sample())



Bạn có thể xây dựng các môi trường khác theo cách tương tự.

**4. Xây dựng tác nhân học tập với Python :**

Đối với việc xây dựng một agent học củng cố, mình sẽ sử dụng thư viện OpenAi Gym như sau :

**import** gym

env = gym.make('CartPole-v0')

**for** \_ in **range**(20):

observation = env.reset()

**for** i in **range**(100):

env.render()

print(observation)

action = env.action\_space.sample()

observation, reward, done, info = env.step(action)

**if** done:

print("Episode finished after {} timesteps".format(i+1))

**break**

