Bài tập 1: Cài đặt Cross-Entropy method để huấn luyện agent hạ cánh trong LunarLander-v2 environment.

Yêu cầu: Sử dụng các reward thresholds (reward bounds) và network architectures khác nhau và so sánh kết quả. Dùng Monitor wrapper class để quay lại quá trình chạy thử.

Gợi ý: Có thể sử dụng incomplete code được cung cấp (file “[INCOMPLETE] week13\_cross\_entropy\_CartPole.py”) và bổ sung missing parts.

Trả lời:

So sánh kết quả khi sử dụng các reward thresholds (reward bounds) và network architectures khác nhau

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
| Chạy chậm hơn các model còn lại vì số lượng neural lớn | Chạy chậm hơn model 3 mặc dù neural thấp hơn, nhưng BATCH\_SIZE và % lấy cái tốt thấp | Model 3 quá trình training khá nhanh, vì số lượng BATCH\_SIZE và neural lớn và lấy 25% cái tốt nhất (khá cao). |
|  |  |  |
|  |  |  |
|  |  |  |

Kết luận: sau khi training xong thì agent chạy từ các model khác nhau có kết quả tương đối ổn và đạt được reward cao. Model 3 training nhanh hơn (khoảng 12 phút) nhưng kết quả thu được khá ấn tượng, 2 model còn lại trên 15 phút. Nhìn chung cả 3 model trên đều đáp ứng được reward điểm thưởng cao

Bài tập 2: Nếu cài đặt Cross-Entropy method để huấn luyện agent leo dốc trong MountainCar-v0 environment, thì thời gian huấn luyện đến khi agent thực hiện được nhiệm vụ sẽ có thể mất nhiều giờ. Thời gian này dài hơn rất nhiều so với Bài tập 1 (khoảng dưới vài chục phút) dù sử dụng cùng reward bounds và network architectures.

Hãy giải thích lý do.

Trả lời:

LunarLander-v2 MountainCar-v0

|  |  |
| --- | --- |
|  | Đây là các thông số observation của 2 môi trường trên |

Một tính chất tiêu biểu của Reinforcement là agent không cần hiểu các giá trị của observation thu vào ở mỗi step là gì mà nó vẫn chạy được. Chính vì sự không hiểu các con số vật lý từ các observation thu được nên ở những môi trường khác nhau thì thời gian training sẽ khác nhau. Có thể ở môi trường LunarLander-v2 các thông số đầu vào nhiều nên vô tình các hướng đi đúng đắn của agent nó được thúc đẩy. Ở MountainCar-v0 ít thông số nên hướng đúng ít được thúc đẩy hơn.

Mặc dù là nó không hiểu ý nghĩa của các con số vật lý, nhưng môi trường của nó khi thực hiện nhiệm vụ được ưu ái hơn (theo em là khi từ trên cao xuống thì nó có thể nhìn được nhiều thứ hơn là ở dưới dốc sâu nhình lên của MountainCar). Vì thế nó thu được nhiều các episole ưu tú có reward cao -> nó luôn tự tin và kỳ vọng rằng sẽ đi đúng hướng. Còn ở MountainerCar thì nó thu được các episole ưu tú ít hơn -> nó chưa có tự tin và kì vọng các hướng đi của nó là đúng đắn.

Vì vậy dù có sử dụng cùng reward bounds và network architectures thì MountainCar vẫn chậm hơn nhiều.