Report for AI3007: Group 27

1st Bùi Văn Khải
22022574

Linkgithub: https://github.com/vankhaipro/Magent2-pixel

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Học Tăng Cường là một kỹ thuật máy học (ML) đào tạo phần mềm đưa ra quyết định nhằm thu về kết quả tối ưu nhất. Kỹ thuật này bắt chước quy trình học thử và sai mà con người sử dụng để đạt được mục tiêu đã đặt ra. RL giúp phần mềm tăng cường các hành động hướng tới mục tiêu, đồng thời bỏ qua các hành động làm xao lãng mục tiêu. Thuật toán RL sử dụng mô hình khen thưởng và trừng phạt trong quy trình xử lý dữ liệu. Các thuật toán này tiếp thu ý kiến phản hồi của từng hành động và tự khám phá ra con đường xử lý tốt nhất để thu về kết quả cuối cùng. Thuật toán RL còn có khả năng trì hoãn khen thưởng. Chiến lược tổng thể tốt nhất có thể đòi hỏi phải đánh đổi một vài lợi ích trước mắt, vì vậy cách tiếp cận tốt nhất mà RL khám phá ra có thể bao gồm một số trừng phạt hoặc giai đoạn quay lui. RL là phương thức hiệu quả giúp hệ thống trí tuệ nhân tạo (AI) đạt kết quả tối ưu trong môi trường chưa biết.

Trong bài tập cuối kì này em đã sử dụng MAgent2 một thư viện dùng để tạo ra các môi trường nơi số lượng lớn các tác nhân pixel trong một thế giới dạng lưới tương tác với nhau trong các trận chiến hoặc các tình huống cạnh tranh khác. Áp dụng các thuật toán trong học tăng cường và model học máy như QNetwork để train các tác nhân pixel tương tác với nhau.

Bảng I
MAIN RESULTS.

Model	Win	Draw	Lose
Random	1.0	0.0	0.0
Pretrain-0	0.0	0.0	1.0
New Pretrain	0.73	0.27	0.0

Average score for each models, using 1 point for Win, 0.5 point for Draw and 0 point for Lose:

- Random: 1
- Pretrain-0: 0
- New Pretrain: 1

II. Methods

- Đầu tiên ta cần tìm hiểu về thư viện MAgent2. Nó là một công cụ mô phỏng môi trường đa tác nhân (multi-agent environment) hỗ trợ các bài toán tương tác chiến lược phức tạp. Cấu trúc môi trường của nó gồm trạng thái (state),hành động (action),phần thưởng(reward),kịch bản (Scenarios)
- Phương pháp học tăng cường Deep-Q-learning là sự kết hợp giữa học tăng cường và mạng nơron sâu. Chúng được sử dụng để giải quyết vấn đề Q-learning một kỹ thuật học tăng cường truyền thống. DQN sử dụng một mạng nơron để ước lượng giá trị Q (Q-value), cho phép tác nhân chọn hành động tối ưu trong mỗi trạng thái.Gồm các thành phần chính :Mạng nơron sâu nhận vào trạng thái của môi trường và trả về giá trị Q cho từng hành động khả thi, Bộ nhớ Lưu trữ các trải nghiệm của tác nhân dưới dạng bộ (state, action, reward, next state), Sử dụng thuật toán tối ưu hóa để điều chỉnh trọng số của mạng dựa trên các trải nghiệm trong bộ nhớ. Deep-Q-learning sẽ học theo quy trình tác nhân khám phá các hành động khác nhau trong môi trường để thu thập dữ liệu trước sau đó với mỗi trải nghiệm sẽ được lưu vào bộ nhớ, mỗi khi có đủ dữ liệu mạng nơron được cập nhật để cải thiện khả năng dự đoán giá trị Q, cuối cùng dựa trên giá trị Q ước lượng, tác nhân chọn hành động tốt nhất.

III. Implementation

Tạo một file train bằng model deep-Q-learning gồm :

- Class QNetworkpre
  - Mạng CNN (self.cnn) xử lý dữ liệu quan sát (observation) đầu vào,Hai tầng tích chập (Conv2d) với kernel kích thước 3×3 và hàm kích hoạt ReLU,Input và output giữ nguyên số kênh, phù hợp với dữ liệu hình ảnh.
  - Tầng Fully Connected (self.network) sau khi qua CNN, đầu ra được làm phẳng và đưa qua một mạng MLP (Multi-Layer Perceptron). Các tầng : tầng 1 Flattened dim→120 ,tầng 2 120→84 , tầng 3 84→action-shape (tương ứng số lượng hành động có thể chọn).
  - dummyinput :Một tensor đầu vào giả được sử dụng để xác định kích thước đầu ra của tầng CNN, từ đó tính số chiều làm phẳng.
  - Hàm forward : Nhận đầu vào x (dữ liệu quan sát từ môi trường),chạy qua CNN,làm phẳng đầu ra (reshape) để phù hợp với tầng Fully Connected,tính toán giá trị Q cho tất cả các hành động từ trạng thái đầu vào.
- Class DQNAgent :
  - Mạng Q và Target Q :q-network mạng Q chính, được huấn luyện để xấp xỉ hàm Q, target-network một bản sao của q-network, được dùng để tính toán giá trị Q mục tiêu, Ban đầu target-network sao chép tham số của q-network.
  - Replay Memory : Bộ nhớ dạng hàng đợi (deque) với kích thước tối đa 10,000 để lưu trữ các trải nghiệm

- Hyperparameters : batch-size kích thước batch dùng để huấn luyện (32 mẫu/lần), gamma hệ số chiết khấu, giúp cân bằng giữa phần thưởng hiện tại và tương lai, epsilon tham số của chiến lược -greedy (khám phá hành động mới), learning-rate tốc độ học của bộ tối ưu.
- Optimizer : Sử dụng Adam để cập nhật tham số mạng Q.
- Hàm act - Chọn hành động dựa trên trạng thái hiện tại Với xác suất chọn hành động ngẫu nhiên (để khám phá môi trường), ngược lại chọn hành động tốt nhất (hành động có giá trị Q lớn nhất).
- Hàm remember - Lưu trải nghiệm
- Hàm replay - Huấn luyện mạng Q :Chọn một batch từ replay memory, chuẩn hóa dữ liệu, tính giá trị Q mục tiêu, Tính giá trị Q hiện tại, tính loss và cập nhật mạng Q , Cập nhật target-network đồng bộ hóa tham số của target-network với q-network sau mỗi bước

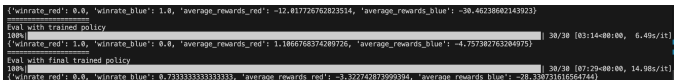
- Hàm train :

  - Khởi tạo môi trường battle-v4 được thiết lập với kích thước bản đồ 45×45, cung cấp quan sát dưới dạng ảnh RGB.
  - Tác nhân và mạng Q : Tác nhân "blue"sử dụng mạng Q để ước tính giá trị Q cho từng hành động và học qua replay memory, tác nhân "red"sử dụng mạng Q được huấn luyện sẵn hoặc chọn hành động ngẫu nhiên
  - Vòng lặp qua các tập huấn luyện :Tại mỗi bước, tác nhân "blue"chọn hành động dựa trên chiến lược -greedy, lưu trải nghiệm vào bộ nhớ và huấn luyện bằng phương pháp batch training, Tác nhân "red"tương tác bằng cách thực hiện hành động ngẫu nhiên.
  - Theo dõi và lưu kết quả : Trọng số của mạng Q được lưu khi đạt phần thưởng tốt nhất, phần thưởng từng tập được theo dõi.

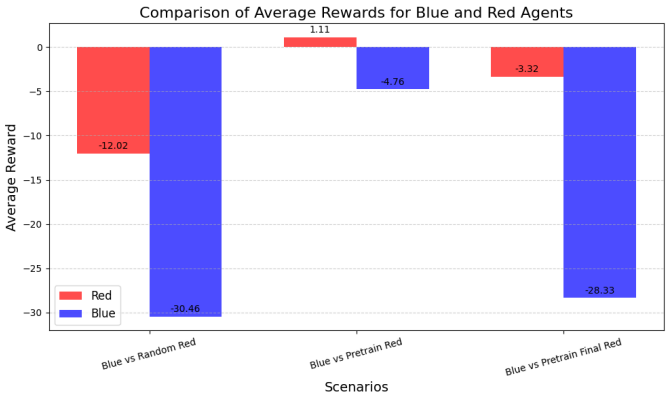
- Sau khi train xong mình lưu được một file là blue.pt sẽ mang đi thử nghiệm cho tương tác với các pixel khác qua file eval.py để biết kết quả hoặc qua file main.py xem lại video mà các tác nhân pixel tương tác với nhau.

IV. Experiment Results

- Biểu đồ hiện thị average-reward của blue với 3 tác nhân red.



Hình 1. Thông số chạy code



Hình 2. Biểu đồ hiển thị average-reward

V. Conclusion

- Sau khi thực hành làm việc về đề tài project sử dụng MAgent2 và các thuật toán RL để train các tác nhân tương tác với nhau em đã học thêm được nhiều thứ thú ví về AI tuy rằng em chưa chiến thắng được hết tác nhân của thầy và model còn bị overfitt nhưng em sẽ cố cải thiện nâng cấp lại model để hoàn hảo hơn.

Tài liệu

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] Github :https://github.com/Farama-Foundation/MAgent2?tab=readme-ov-file https://github.com/giangbang/RL-final-project-AIT-3007?tab=readme-ov-file