TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

BÀI TẬP LỚN

**HỌC MÁY**

Đề tài: Áp dụng Reinforcement Learning cho game Pacman

|  |  |  |
| --- | --- | --- |
| **Nhóm số 5** | Ngô Quang Hòa | 20141840 |
|  | Nguyễn Văn Thành | 20144102 |
|  |  |  |
|  |  |  |

**Mã lớp**: 91620

**Giảng viên**: TS. Nguyễn Nhật Quang

**Hà Nội, 5/2017**

Mục lục

[I. Giới thiệu về bài toán điều khiển pacman trong game . 2](#_Toc482593681)

[**1. Luật chơi game Pacman** 2](#_Toc482593682)

[**2. Bản đồ** 3](#_Toc482593683)

[**3. Yếu tố môi trường** 4](#_Toc482593684)

[**4. Nhận xét** 4](#_Toc482593685)

[**5. Mục tiêu và phương pháp áp dụng** 4](#_Toc482593686)

[**II.Thuật toán Reinforcement learning** 4](#_Toc482593687)

[**1. Lý thuyết chung** 4](#_Toc482593688)

[**2. Q-learning áp dụng với Pacman** 6](#_Toc482593689)

[III. Các hướng cải thiện: 7](#_Toc482593690)

[**1. Tăng hiệu quả của hệ thống với việc lưu các trạng thái đã có** 7](#_Toc482593691)

[IV. Kết quả huấn luyện hệ thống 8](#_Toc482593692)

[**1, Thay đổi hệ số học α** 8](#_Toc482593693)

[**2, Thay đổi hệ số phụ thuộc** 11](#_Toc482593694)

[V.Các chức năng chính của hệ thống và Hướng dẫn sử dụng. 12](#_Toc482593695)

[VIII. Các gói phần mềm và tài liệu tham khảo 14](#_Toc482593696)

[VIII.Khó khăn và cách giải quyết 14](#_Toc482593697)

[VII. Kết quả đạt được 14](#_Toc482593698)

# I. Giới thiệu về bài toán điều khiển pacman trong game .

## **1. Luật chơi game Pacman**

* Pacman và Ghost không thể di chuyển qua tường.
* Pacman phải ăn hết các viên đậu đề giành chiến thắng.
* Mỗi lần di chuyển Pacman bị trừ 1 điểm.
* Pacman thắng nếu ăn hết các viên đậu và được thưởng 500 điểm.
* Pacman bị Ghost bắt sẽ bị thua và trừ 500 điểm.
* Mỗi lần ăn đậu Pacman được thưởng 10 điểm.
* Pacman có thể bắt Ghost nếu như ăn được viên năng lượng, điểm thưởng là 200.
* Ghost không được dừng và chỉ có thể quay đầu nếu gặp ngõ cụt.
* Ghost sẽ chuyển sang trạng thái sợ nếu như Pacman ăn viên năng lượng, thời gian duy trì trạng thái là 40 bước đi.

## **2. Bản đồ**

**Hình 1:** Bản đồ dạng lưới 11 x 17

Bản đồ dạng lưới 11 x 17 ô. Pacman được đánh chỉ số là 0, Ghost đánh chỉ số bắt đầu từ 1.

Có tất cả 96 vị cho Pacman và Ghost di chuyển, có 90 ô có thể thay đổi trạng thái (có hoặc không có đậu, viên năng lượng trên đó).

Pacman luôn nằm trong một ô trong lưới . Ghost luôn nằm trong một ô trong lưới nếu như ở trạng thái bình thường và có thể nằm ở giữa hai ô trong lưới nếu như đang sợ (đi chậm), lúc đó Ghost chỉ có thể di chuyển tiếp theo hướng cũ cho tới khi nằm trọn vẹn trong một ô.

Chi phí dịch chuyển giữa các ô trong lưới luôn là 1.

## **3. Yếu tố môi trường**

* Môi trường quan sát được toàn bộ, Pacman được quyền truy cập đến trạng thái đầy đủ của môi trường hiện tại.
* Trạng thái tiếp theo của môi trường được xác định hoàn toàn dựa trên trạng thái hiện tại và hành động của các tác tử.
* Việc lựa chọn hành động của tác tử phụ thuộc vào các hành động trước đó.
* Tĩnh, khi các tác tử cân nhắc đưa ra hành động thì môi trường không thay đổi.
* Rời rạc, các nhận thức và hành động là hữu hạn và được phân biệt rõ ràng.
* Đa tác tử cạnh tranh (2 Ghost và Pacman) trong đó, Ghost sử dụng chiến lược Cắt tỉa α – β để tìm đường đi.

### **4. Nhận xét**

* Tìm đường đi cho Pacman thuộc dạng toán NP – hard [[1]](#footnote-1).
* Số trạng thái cỡ rất lớn và trạng thái kế tiếp chỉ được biết khi đã thực xong hiện hành động do 2 ghost di chuyển cùng lúc với di chuyển của Pacman.

## **5. Mục tiêu và phương pháp áp dụng**

* Cài đặt các thuật toán thông minh Pacman.
* Thử nghiệm một số cải tiến.
  + Lý do chọn phương pháp Học tăng cường:

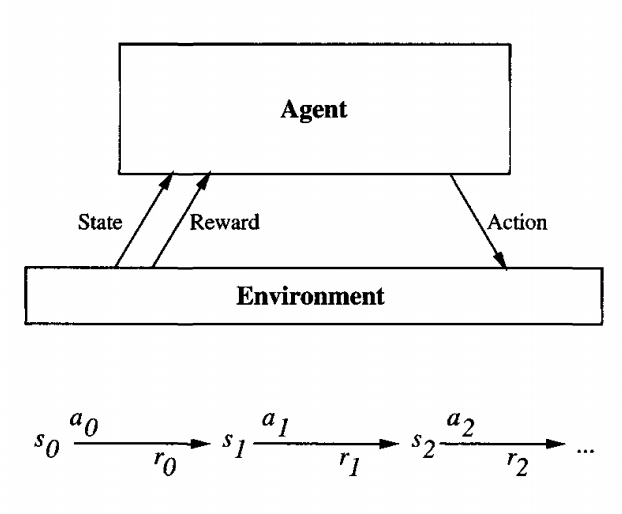
Reinforcement learning phù hợp cho bài toán mà tác tử có thể tương tác, thu được phản hồi từ môi trường. tri thức được thu thập thêm trong quá trình chơi. Pacman có thế lấy phản hồi, chính bằng chênh lệch điểm số giữa trước, và sau khi thực hiện nước đi. Còn việc lý giải vì sao cần giải thuật Q-leaning được trình bày kĩ ở phần II.

## **II.Thuật toán Reinforcement learning**

## **1. Lý thuyết chung**

Học tăng cường là một thuật toán học máy được áp dụng để học cho một tác tử ở trong một môi trường nhất định. Tác tử sẽ học cách tương tác với môi trường để tối đa hàm giải thưởng. Mỗi tác tử sẽ thu nhận thông tin trạng thái từ môi trường, thực hiện hành động, và thu được phản hồi từ môi trường thông qua hàm giải thưởng

Mô hình chung của Reinforcement learning



Mục tiêu bài toán học máy trong trường hợp này là học được chiến thuật chơi  để chọn hành động tiếp theo at dựa trên trạng thái hiện tại st, tức là π(st) =at .Thông thường, ta chọn hàm π để tối đa hàm giá trị chồng chất V(st), với hàm V(st) được định nghĩa như sau :



Trong đó :

Rt là phần thưởng tại thời điểm thứ t

ᵞ là giá trị phụ thuộc thể hiện mối quan hệ giữa phần thưởng hiện tại và và phần thưởng sau đó.

Vậy bài toán đặt ra là tìm chiến thuật chơi tối ưu π\* thỏa mãn



Nhìn chung, trong thực tế thì chiến thuật tối ưu này rất khó tìm trực tiếp cho một môi trường bất kì. Bởi vì, ta không có tập dữ liệu phục vụ cho việc training theo dạng <s,a> với s là trạng thái của tác tử và a là hành động tương ứng với trạng thái đó.

Vậy thì ta nên chọn hàm giá trị V như thế nào để đánh giá tác tử?. Một lựa chọn dễ thấy nhất là chọn hành động a1 để dẫn tới trạng thái s1 có hàm V\*(s1) > V\*(s2) . Vậy lựa chọn tối ưu là chọn hành động a trong trạng thái s để tối đa tổng của phần thưởng r(s,a) và giá trị hàm V\* theo công thức sau;



Trong đó

δ(s,a): là hàm trạng thái thu được sau khi thực hiện hành động a trong trạng thái s

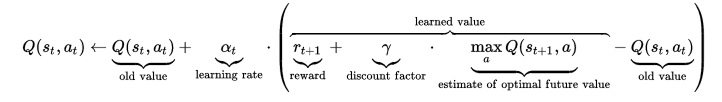
ᵞ: hệ số phụ thuộc

## **2. Q-learning áp dụng với Pacman**

Trong thực tế,tác tử không biết chính xác hàm r(s,a) và δ(s,a),mà phải chạy thực tế để lấy được từng giá trị r. từ đó không thể chọn được chiến thuật đi tôi ưu dựa vào công thức trên. Ta xét hàm Q(s,a) như sau



Ở đây, ta chỉ cần biết giá trị hàm Q(s,a), nhưng không cần biết giá trị từng thành phần cấu tạo nên Q(s,a). Tất nhiên, ta cũng không biết chính xác giá trị hàm Q, nhưng giá trị hàm này ta có thể tính toán xấp xỉ thông qua thuật toán truy hổi.



α : Hệ số học (learning rate)

* ᵞ : Hệ số phụ thuộc
* r : Hàm phần thưởng

Hàm phần thưởng r chính là điểm ở trạng thái s

* R = là điểm mà Pacman đang có ở trạng thái S.
* Phần thưởng, để khách quan, tính bằng sự chênh lệch điểm số trước, và sau khi thực hiện hành động.

Như đã nêu ở phần lý thuyết chung, quá trình học của Pacman không dựa vào tập huấn luyên thông thường có dạng <s,a> với s là trạng thái, a là hành động tương ứng với trạng thái đó. Vì vậy, Pacman được huấn luyện dựa vào việc đi thử, tạo 1 trạng thái “Pac-state” tính Q, cho nó và bổ sung vào vị trí phù hợp trong “chuỗi các trạng thái Pac-state” của pacman

# III. Các hướng cải thiện:

## **1. Tăng hiệu quả của hệ thống với việc lưu các trạng thái đã có**

Trong quá trình huấn luyện hệ thống, chúng em nhận thấy, hệ thống huấn luyện Pacman rất tốn thời gian. Lý do là trong quá trình học, có nhiều trạng thái bị lặp lại. Nhưng Pacman không nhân được ra nên tại các trạng thái bị lặp pacman vẫn tính toán như trạng thái mới. Chính vì vậy, chúng em sẽ lấy đặc trưng môi trường của trạng thái hiện tạiCác đặc trưng của môi trường gồm có:

a.vị trí hiện tại của pacman

b.Số lượng hạt đậu còn lại

c.Vị trí hạt đậu gần nhất

d.Trong vòng 5 nước đi có ghost nào không.

e.Vị trí 2 ghost đang ở trạng thái truy đuổi

Nếu một trạng thái mới và trạng thái đã có trong danh sách, Có:

(d là false và a,b,c trùng nhau), hoặc (d là true a,b,c,e trùng nhau)

(khi không gặp nguy hiểm, thì không cần kiểm tra điều kiện 4)

Thì sẽ xác định được trạng thái mới đã được trải qua, lặp lại.

**2.**Chọn nước đi khi gặp một trạng thái chưa được khám phá trước đây.

Chúng em đề xuất 2 hướng giải quyết:

* Dùng một hàm đánh giá , do lập trình viên thiết lập, để gợi ý xem nước đi nào sẽ dẫn được trạng thái có Q cao. Tuy cách làm này đưa về kết quả tốt. nhưng một sự can thiệp từ con người sẽ làm cho quá trình học của máy trở nên thiếu tự nhiên, không thể theo dõi được.
* Dùng một modul học máy hồi quy tuyến tính để tính gần đúng Q cho trạng thái kế tiếp . sử dụng tập dữ liệu huấn luyện từ các trạng thái cũ đã trải qua.

Q=w1\*x1+w2\*x2+….

Đầu vào: x1, x2 là các thuộc tính của môi trường đã lưu được, ví dụ như: số đậu, viên năng lượng còn lại, khoảng cách đến các ghost, vv….

Đầu ra: Q là giá trị ứng với trạng thái đó.   
w là các hệ số cần xác định   
ta xác đinh được w, và tính ra Q gần đúng cho một trạng thái tiếp theo.  
Tuy vậy , việc chọn ra các thuộc tính x là một công việc khó khăn. Do dữ liệu của trạng thái là kiểu có cấu trúc, lấy mẫu các x như thế nào để có được một tính toán hồi quy hợp lý chúng em chưa thực hiện được.

#Ý tưởng về phương pháp này Approximate Q-Learning là từ CS188 Pacman Project của đại học Berkeley**,**. <http://ai.berkeley.edu/reinforcement.html>

# IV. Kết quả huấn luyện hệ thống

Hệ thống học với Pacman phụ thuộc vào 2 tham số chủ yếu

* Hệ số học α – leaning rate. Xác định mức độ tiếp thu, tức là mức độ thay thế thông tin mới nhận được cho thông tin cũ thu được lúc trước.
* Hệ sô phụ thuộc ᵞ- discount vector. Xác định tầm quan trọng của các phần thưởng trong tương lai, tức mức độ mong đợi phần thưởng về sau của tác tử.

Chính vì vậy, ta cần tìm bộ tham số <α,ᵞ> sao cho Pacman có thể đạt kết quả cao nhất

## **1, Thay đổi hệ số học α**

Ta lần lượt thay đổi hệ số học α , giữ nguyên hệ số phụ thuộc y

Chúng em nhận thấy, hiệu quả học của Pacman tỉ lệ thuận với hệ số học α.   
Điều này theo chúng em suy đoán nguyên nhân do: 2 ghost chạy với alpha-belta cắt tỉa.Đối với mỗi bước của pacman, 2 ghost luôn luôn phản ứng: chạy lại gần pacman 1 nước, vậy nên môi trường có thể xem là xác định - fully [deterministic](https://en.wikipedia.org/wiki/Deterministic_system) environments. Mà theo lý thuyết, môi trường càng xác định thì α được gợi ý là cho càng gần 1 cảng tốt.

## **2, Thay đổi hệ số phụ thuộc**

Giữ nguyên α =0.9 , thay đổi hệ số phụ thuộc

Nhận xét: Có thể thấy discount vector càng lớn thì xu hướng đồ thị có độ dốc càng cao. Tức là: sự thỏa mãn với điểm số hiện tại thấp.pacman cố dành đc các phần thưởng to hơn về sau.

# V.Các chức năng chính của hệ thống và Hướng dẫn sử dụng.

**Phần mềm hỗ trợ cách thức tạo lập 1 thí nghiệm, in kết quả, như sau:**

**Hướng dẫn sử dụng**

* Trong thư mục chứa chương trình thực thi hoàn thiện (thư mục dist) . Mở theo đường dẫn"dist\src\Experiment" ta thấy thư mục thí nghiệm. có 2 file config.txt và result.txt

Bố trí thí nghiệm bằng cách thay đổi hệ các số trong file config.txt :

Ví dụ:

Learning rate: 0.9 Discount factor: 0.5 Number of run: 1000 LessState: 1

Sẽ tạo ra thí nghiệm 1000 lần chạy với Learning rate: 0.9 Discount factor: 0.5. LessState =1: bật chế độ lưu các trạng thái đã có (trong chương III đã giải thích)\_LessState =0 sẽ tắt chế độ này.

* Chạy game bằng cách nhấn trực tiếp vào file ProjectAI.jar trong thư mục dist
* Kết quả sẽ in trong file result.txt

**VI. Cấu trúc của mã nguồn chương trình- vai trò của các lớp:**

Chương trình gồm các packages:

Images: chứa các file ảnh của game.

Projectai: chứa các class.

Experiment: chứa các file congfix cấu hình thuộc tính cho pacman, và result in kết quả chạy

Vai trò của các class:

GraphicDisplay, Layout, main, Game, Rules, Util. cấu hình nên môi trường chạy game.

AstarSearch: sử dụng giống như một hàm tiện ích, dùng để đo chính xác số bước đi giữa hai ô trong bản đồ

GhostAlphaBetaPrunningAgent: cài đặt thuật toán đi của Ghost- alpha beta cắt tỉa

PacmanClassicQleaning và StateThinking: cài đặt thuật toán đi của Pacman khi chưa cải tiến (theo mục III). Trong đó StateThinking mô tả trạng thái hiện tại mà theo cách nhìn của pacman.

PacmanClassicQleaningLessState, và StateThinkingLess : cài đặt thuật toán đi của Pacman khi đã cải tiến (theo mục III). Việc cải tiến làm loại bỏ bớt các trạng thái lặp lại nên đặt tên là “lessState”

# VIII. Các gói phần mềm và tài liệu tham khảo

* Artificial Intelligence: A Modern Approach, Stuart J. Russell and Peter Norvig, 3rd edition.
* UC Berkeley**,** CS188 Pacman Project.
* Slide Học máy.
* Neural-Dynamic Programming

# VIII.Khó khăn và cách giải quyết

* Source code Java cho bài toán Q-leaning không có nhiều, việc cài đặt lại phụ thuộc lớn vào mỗi bài toán riêng nên cần phải cài đặt theo lý thuyết chứ không thể dùng các thư viện, hay các mô hình học máy cho các trò chơi khác (Cờ backgammon).
* Không thể áp dụng trực tiếp lý thuyết toán học (chuỗi Markov) nên khó lập trình
* Các source code Python tham khảo của Pacman Project của UC Berkeley**,** CS188 không tương thích với các bản Python hiện tại nên không thể kế thừa

# VII. Kết quả đạt được

* Cài đặt được pacman có khả năng tự học, rút kinh nghiệm qua mỗi lượt đi
* Hiểu được mô hình học tăng cường
* Thu được nhiều kinh nghiệm thiết kế, lập trình với Java
* Rèn luyện được tinh thần và kỹ năng làm việc tập thể.
* Rút được bài học về sự quan trọng của phần thưởng và phạt cho quá trình học tự nhiên của con người.

1. Giovanni Viglietta, Gaming is a hard job, but someone has to do it! [↑](#footnote-ref-1)