đồ án III

***Giảng viên hướng dẫn: TS.Nguyễn Thiên Bảo***

Nhóm Thực hiện:

* Nguyễn Bá Lê An – 15110001
* Nguyễn Hữu Khang - 15110062

**Đề Tài**

**hybrid reward architecture for reinforcement learning**

***Bản dịch bài báo***

**Kiến trúc thưởng giống lai cho sự tăng cường học tập**

## Harm van Seijen1

[harm.vanseijen@microsoft.com](mailto:harm.vanseijen@microsoft.com)

## Mehdi Fatemi1

[mehdi.fatemi@microsoft.com](mailto:mehdi.fatemi@microsoft.com)

## Joshua Romoff12

[joshua.romoff@mail.mcgill.ca](mailto:joshua.romoff@mail.mcgill.ca)

## Romain Laroche1

[romain.laroche@microsoft.com](mailto:romain.laroche@microsoft.com)

## Tavian Barnes1

[tavian.barnes@microsoft.com](mailto:tavian.barnes@microsoft.com)

## Jeffrey Tsang1

[tsang.jeffrey@microsoft.com](mailto:tsang.jeffrey@microsoft.com)

1Microsoft Maluuba, Montreal, Canada

2McGill University, Montreal, Canada

# Tóm tắt

*Một trong những thách thức chính trong tăng cường học tập (RL) là tổng quát. Trong phương pháp RL sâu điển hình này được thực hiện bằng xấp xỉ hàm giá trị tối ưu với một đại diện thấp chiều sử dụng một mạng lưới sâu. Trong khi phương pháp này hoạt động tốt trong nhiều lĩnh vực, trong các lĩnh vực nơi mà các chức năng giá trị tối ưu không thể dễ dàng được giảm đến một đại diện thấp chiều, học tập có thể rất chậm và không ổn định. Bài viết này góp phần hướng tới việc giải quyết các lĩnh vực đầy thử thách như vậy, bằng cách đề xuất một phương pháp mới, được gọi là Kiến trúc thưởng lai (HRA). HRA mất như là đầu vào một chức năng phần thưởng bị phân hủy và học một hàm giá trị riêng biệt cho mỗi chức năng thưởng thành phần. Bởi vì mỗi thành phần thường chỉ phụ thuộc vào một tập hợp con của tất cả các tính năng, chức năng giá trị tương ứng có thể xấp xỉ dễ dàng hơn bằng một đại diện thấp chiều, tạo điều kiện cho học tập hiệu quả hơn. Chúng tôi chứng minh HRA trên một món đồ chơi-vấn đề và các trò chơi Atari Ms.Pac-Man, nơi HRA đạt hiệu suất trên con người.*

# **Giới thiệu**

Trong tăng cường học tập (RL) [(Sutton & Barto, năm 1998;](#_bookmark33) [Szepesvári, 2009),](#_bookmark36)mục đích là để tìm một chính sách hành vi nhằm tối đa hóa sự trở lại-tổng chiết khấu thưởng nhận được hơn thời gian một cách hướng dữ liệu. Một trong những thách thức chính của RL là để mở rộng các phương pháp như vậy mà họ có thể được áp dụng cho lớn, các vấn đề thực tế. Bởi vì không gian trạng thái của vấn đề như vậy thường là lớn, tổng quát mạnh mẽ là cần thiết để học hỏi một chính sách tốt hiệu quả.

[Mnih et al. (2015)](#_bookmark23) đạt được một bước đột phá lớn trong lĩnh vực này: bằng cách kết hợp các kỹ thuật RL tiêu chuẩn với các mạng thần kinh sâu, họ đạt được hiệu suất cao hơn con người trên một số lượng lớn các trò chơi Atari 2600, bằng cách học một chính sách từ pixel. Các đặc tính khái quát của Deep Q-Networks họ phương pháp (DQN) được thực hiện bằng xấp xỉ hàm giá trị tối ưu. Một chức năng giá trị đóng một vai trò quan trọng trong việc RL, bởi vì nó dự báo lợi nhuận kỳ vọng, lạnh trên một cặp bang hoặc tiểu bang-action. Khi hàm giá trị tối ưu được biết đến, một chính sách tối ưu có thể được bắt nguồn bằng cách hành động tham lam đối với nó với. Bằng cách mô hình hóa các ước tính hiện tại của hàm giá trị tối ưu với một mạng lưới thần kinh sâu, DQN thực hiện một sự tổng quát mạnh vào chức năng giá trị, và do đó về chính sách.

Các hành vi tổng quát của DQN đạt được bằng cách quy tắc trên mô hình cho hàm giá trị tối ưu. Tuy nhiên, nếu hàm giá trị tối ưu là rất phức tạp, sau đó học một đại diện thấp chiều chính xác có thể là một thách thức hay thậm chí là không thể. Vì vậy, khi các chức năng giá trị tối ưu không thể dễ dàng được giảm đến một đại diện thấp chiều, chúng tôi tranh luận để áp dụng một hình thức bổ sung của quy tắc về phía mục tiêu. Cụ thể, chúng tôi đề xuất để thay thế chức năng giá trị tối ưu là mục tiêu cho đào tạo với một chức năng giá trị thay thế đó là dễ dàng hơn để tìm hiểu, nhưng vẫn mang lại một không tối ưu chính sách hợp lý-nhưng nói chung, khi hành động tham lam đối với nó với.

Các quan sát chính đằng sau quy tắc về chức năng mục tiêu là hai chức năng giá trị rất khác nhau có thể dẫn đến việc chính sách tương tự khi một đại lý đóng vai trò tham lam đối với họ. Cùng lúc đó, một số chức năng giá trị được dễ dàng hơn để tìm hiểu hơn những người khác. Động lực nội tại[(Bia đen](#_bookmark32) [et al., năm 2005;](#_bookmark32) [Schmidhuber, 2010)](#_bookmark30) sử dụng quan sát này để cải thiện việc học trong các lĩnh vực thưa thớt-khen thưởng, bằng cách thêm một tín hiệu thưởng nội tại miền cụ thể để phần thưởng đến từ môi trường. Khi chức năng thưởng nội tại được tiềm năng dựa trên, tối ưu của chính sách kết quả được duy trì[(Ng et al., 1999).](#_bookmark26) Trong trường hợp của chúng tôi, chúng tôi đặt mục tiêu cho các chức năng giá trị đơn giản hơn đó là dễ dàng hơn để đại diện cho cơ quan đại diện thấp chiều.

Chiến lược chính của chúng tôi cho xây dựng một chức năng giá trị dễ dàng để tìm hiểu là để phân hủy các chức năng phần thưởng của môi trường vào các chức năng phần thưởng n khác nhau. Mỗi trong số họ được gán một tác nhân tăng cường học tập riêng biệt. Tương tự như các kiến ​​trúc Horde[(Sutton et al., 2011),](#_bookmark34) tất cả các thuốc này có thể học song song trên dãy cùng một mẫu bằng cách sử dụng off-chính sách học tập. Mỗi đại lý cho hành động giá trị của nó trong tình trạng hiện thời đến một thập dữ liệu, trong đó kết hợp chúng thành một giá trị duy nhất cho mỗi hành động. Các hành động hiện tại được chọn dựa trên các giá trị tổng hợp.

Chúng tôi kiểm tra cách tiếp cận của chúng tôi trên hai lĩnh vực: một món đồ chơi-vấn đề, nơi một đại lý phải ăn 5 loại trái cây nằm một cách ngẫu nhiên, và bà Pac-Man, một trong những trò chơi khó khăn từ tập ALE chuẩn [(Bellemare et al.,](#_bookmark17) [2013).](#_bookmark17)

# **Công việc có liên quan**

Phương pháp HRA của chúng tôi xây dựng dựa trên kiến ​​trúc Horde [(Sutton et al., 2011).](#_bookmark34) Kiến trúc Horde bao gồm một số lượng lớn các 'quỷ' mà học song song thông qua học tập off-chính sách. Mỗi con quỷ huấn luyện một hàm giá trị chung riêng biệt (GVF) dựa trên chính sách và giả thưởng chức năng riêng của mình. Một giả thưởng có thể được bất kỳ tín hiệu dựa trên tính năng mã hóa thông tin hữu ích. Kiến trúc Horde là tập trung vào việc xây dựng kiến ​​thức tổng quát về thế giới, được mã hóa thông qua một số lượng lớn các GVFs. HRA focusses về đào tạo các thành phần riêng biệt của hàm môi trường khen thưởng, để học hiệu quả hơn một chính sách kiểm soát. UVFA[(Schaul et al., 2015)](#_bookmark29) được xây dựng trên Horde là tốt, nhưng mở rộng nó dọc theo một hướng khác. UVFA phép khái quát qua các nhiệm vụ khác nhau / mục tiêu. Nó không giải quyết như thế nào để giải quyết một nhiệm vụ phức tạp duy nhất, đó là trọng tâm của HRA.

Học đối với nhiều chức năng thưởng với cũng là một chủ đề học tập đa mục tiêu [(Roijers](#_bookmark27) [et al., 2013).](#_bookmark27) Vì vậy, cách khác, HRA có thể được xem như là áp dụng học tập đa mục tiêu để học hiệu quả hơn chính sách đối với một chức năng phần thưởng duy nhất.

Chức năng thưởng phân hủy đã được nghiên cứu trong người khác bằng cách [Russell & Zimdar (2003)](#_bookmark28) và [Sprague & Ballard (2003).](#_bookmark31) công việc trước đây focusses về chiến lược mà đạt được hành vi tối ưu. công việc của chúng tôi là nhằm mục đích cải thiện học tập hiệu quả bằng cách sử dụng chức năng giá trị đơn giản và thư giãn yêu cầu tối ưu.

Ngoài ra còn có điểm tương đồng giữa HRA và Unreal [(Jaderberg et al., 2017).](#_bookmark21) Đáng chú ý, cả hai giải quyết nhiều vấn đề nhỏ hơn để giải quyết một vấn đề khó khăn. Tuy nhiên, hai kiến ​​trúc khác nhau về hoạt động của họ, cũng như các loại thách thức họ giải quyết. Unreal là một kỹ thuật làm tăng đại diện học tập trong các tình huống khó khăn. Nó như vậy bằng cách sử dụng các công việc phụ trợ để giúp đào tạo các lớp cấp dưới của một mạng lưới thần kinh sâu. Một ví dụ về một kịch bản học tập representation- thách thức đó được học tập để điều hướng trong lĩnh vực 3D Labyrinth. Về các trò chơi Atari, đạt được hiệu suất báo cáo của Unreal là tối thiểu, cho thấy rằng các tiêu chuẩn kiến ​​trúc RL sâu là đủ mạnh mẽ để trích xuất các đại diện có liên quan. Ngược lại, các kiến ​​trúc HRA phá vỡ một nhiệm vụ thành từng miếng nhỏ. nhiều nhiệm vụ nhỏ hơn của HRA không không có giám sát; họ là những công việc có liên quan trực tiếp đến nhiệm vụ chính. Hơn nữa, trong khi Unreal vốn là một kỹ thuật RL sâu, HRA là thuyết bất khả tri đến các loại xấp xỉ chức năng sử dụng. Nó có thể được kết hợp sâu với mạng thần kinh, nhưng nó cũng làm việc với chính xác, cơ quan đại diện bảng. HRA rất hữu ích cho các tên miền nơi có một đại diện chất lượng cao là không đủ để giải quyết nhiệm vụ một cách hiệu quả.

Cách tiếp cận hướng đối tượng Diuk của [(Diuk et al., 2008)](#_bookmark18) là một trong những phương pháp đầu tiên cho thấy học tập hiệu quả trong các trò chơi video. Cách tiếp cận này khai thác kiến ​​thức miền liên quan đến quá trình chuyển đổi năng động để học hiệu quả một mô hình chuyển tiếp nhỏ gọn, mà sau đó có thể được sử dụng để tìm một giải pháp sử dụng kỹ thuật năng động-lập trình. cách tiếp cận dựa trên mô hình vốn này có nhược điểm là trong khi nó có hiệu quả học một mô hình rất nhỏ gọn của động lực học chuyển tiếp, nó không làm giảm không gian trạng thái của vấn đề. Do đó, nó không giải quyết các thách thức chính của bà Pac-Man: không gian trạng thái khổng lồ của nó, đó là ngay cả đối với các phương pháp DP khó (Diuk áp dụng phương pháp của mình để một trò chơi Atari chỉ với 6 đối tượng, trong khi bà Pac-Man có hơn 150 đối tượng).

Cuối cùng, HRA liên quan đến lựa chọn [(Sutton et al., năm 1999;](#_bookmark35) [Bacon et al., 2017),](#_bookmark15) và tổng quát hơn hierar- học tập theo đẳng [(Barto & Mahadevan, năm 2003;](#_bookmark16) [Kulkarni et al., Năm 2016).](#_bookmark22) Tùy chọn là hành động tạm-mở rộng, giống như người đứng đầu của HRA, có thể được đào tạo song song dựa trên chức năng riêng (nội tại) phần thưởng của mình. Tuy nhiên, khi một tùy chọn đã được đào tạo, vai trò của chức năng thưởng nội tại của nó là kết thúc. Một đại lý cấp cao có sử dụng một tùy chọn nhìn thấy nó như là chỉ hành động khác và đánh giá nó sử dụng chức năng phần thưởng của riêng mình. Điều này có thể mang lại rất lớn tốc độ-up trong học tập và giúp đỡ đáng kể với việc khai thác tốt hơn, nhưng họ không trực tiếp thực hiện chức năng giá trị của các đại lý cấp cao hơn ít phức tạp. Thủ trưởng HRA đại diện cho các giá trị, đào tạo với các thành phần của phần thưởng môi trường. Ngay cả sau khi đào tạo, những giá trị này lại có liên quan, vì aggregator sử dụng chúng để chọn hành động của nó.

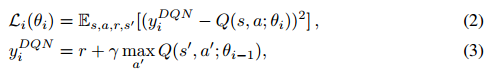
# **mô hình**

Xét một quyết định quá trình (S; A; P; Renv; γi), mà mô hình một tác nhân tương tác với môi trường vào thời điểm rời rạc bước t. Nó có một bộ nhà nước, thiết lập hành động, môi trường thưởng chức năng Renv: S×A×S -> R, và xác suất chuyển chức năng P: S×A×S -> [0; 1]. Tại thời gian *t*, Tác nhân quan sát trạng thái st € *S* và có hành động tại at € *A.* Các đối tượng quan sát trạng thái tiếp theo st + 1, rút ​​ra từ sự phân bố xác suất chuyển P *(st, tại,..)*, và một phần thưởng *rt = Renv (st, at, st + 1).* Các hành vi được xác định bởi một chính sách π: S × A -> [0; 1], đại diện cho xác suất lựa chọn qua hành động. Mục tiêu của một đại lý là để tìm một chính sách nhằm tối đa hóa sự kỳ vọng về sự trở lại, đó là tổng chiết khấu thưởng: Gt :=, Ở nơi mà các nhân tố chiết khấu *γ* €[0*,* 1], điều khiển tầm quan trọng của phần thưởng ngay lập tức so với những phần thưởng trong tương lai. Mỗi π chính sách đều có tương ứng chức năng hành động có giá trị cung cấp cho sự trở lại dự kiến ​​lạnh về tình trạng và hành động, khi hành động theo chính sách:

Tối ưu chính sách π\* có thể được tìm thấy bằng cách lặp đi lặp lại việc cải thiện một ước lượng tối ưu hành động có giá trị hàm Q\*(s, a): = maxπ Qπ (s, a), sử dụng cập nhật mẫu dựa trên. Khi Q\* là đủ chính xác xấp xỉ, hành động tham lam liên quan đến nó mang lại chính sách tối ưu.

## Kiến trúc thưởng lai

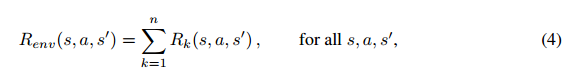
Chức năng Q-giá trị thường được ước tính bằng cách sử dụng chức năng approximator với θ vector trọng lượng: Q (s, a; θ). DQN sử dụng một mạng lưới thần kinh sâu như chức năng approximator và lặp đi lặp lại cải thiện một ước tính của Q \* bằng cách giảm thiểu các chuỗi các chức năng bị mất:



Các cân nặng vector từ các Trước lặp, *θtôi*-1, Là mã hóa sử dụng một tách rời Mục tiêu mạng.

chúng tôi tham khảo các chức năng Q-giá trị giảm thiểu hàm tổn thất (s) như mục tiêu đào tạo. Chúng tôi sẽ gọi một mục tiêu đào tạo phù hợp, nếu hành động tham lam đối với nó với kết quả trong một chính sách đó là tối ưu theo chức năng phần thưởng của môi trường; chúng ta gọi là một mục tiêu đào tạo bán phù hợp, nếu hành động tham lam đối với nó với kết quả trong một chính sách nhưng tốt không phải là một tối ưu một trong các chức năng phần thưởng của môi trường. Đối với (2), mục tiêu đào tạo là Q \* env, chức năng hành động có giá trị tối ưu dưới Renv, đó là mục tiêu đào tạo phù hợp mặc định.

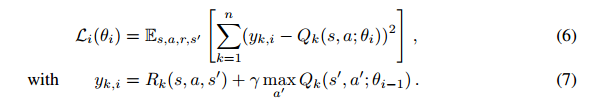
Đó là một mục tiêu đào tạo phù hợp nói gì về cách dễ dàng là để học hỏi mục tiêu đó. Ví dụ, nếu Renv là thưa thớt, mục tiêu học tập mặc định có thể rất khó để tìm hiểu. Trong trường hợp này, thêm một tiềm năng dựa trên tín hiệu phần thưởng bổ sung cho Renv có thể mang lại một cách khách quan học tập phù hợp thay thế đó là dễ dàng hơn để tìm hiểu. Nhưng một phần thưởng môi trường thưa thớt không phải là lý do duy nhất một mục tiêu đào tạo có thể được khó khăn để học tập. Chúng tôi mong muốn tìm một mục tiêu đào tạo thay thế cho các tên miền nơi đào tạo mặc định mục tiêu Q \* env rất khó để tìm hiểu, do chức năng là chiều cao và khó có thể khái quát cho. Cách tiếp cận của chúng tôi được dựa trên một phân hủy của hàm thưởng.



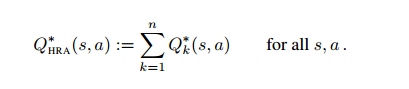
Chúng tôi đề xuất để phân hủy các chức năng phần thưởng Renv vào chức năng thưởng n:

và để đào tạo một tác nhân tăng cường học tập riêng biệt trên mỗi một trong các chức năng phần thưởng. Có vô số phân tách khác nhau của một hàm thưởng có thể, nhưng để đạt được chức năng giá trị được dễ dàng để tìm hiểu, phân hủy nên được như vậy mà mỗi chức năng thưởng bị ảnh hưởng chủ yếu bởi chỉ một số ít các biến trạng thái.

Bởi vì mỗi đại lý k có chức năng phần thưởng của riêng mình, nó cũng có chức năng Q-giá trị riêng của nó, Qk. Nói chung, các đại lý khác nhau có thể chia sẻ nhiều lớp cấp dưới của Q-mạng sâu. Do đó, chúng tôi sẽ sử dụng một θ vector duy nhất để mô tả các trọng kết hợp của các đại lý. Chúng tôi tham khảo các mạng kết hợp đại diện cho tất cả các chức năng Q-giá trị như các Kiến trúc lai thưởng (HRA) (xem Hình[1).](#_bookmark4) lựa chọn hành động cho HRA được dựa trên tổng các chức năng Q-giá trị của đại lý, mà chúng ta gọi QHRA:

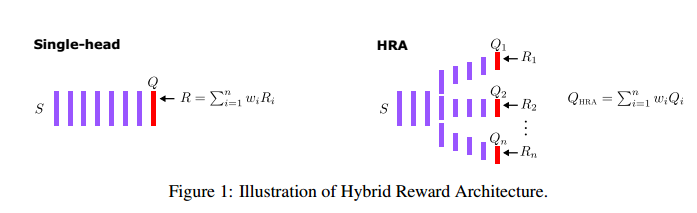
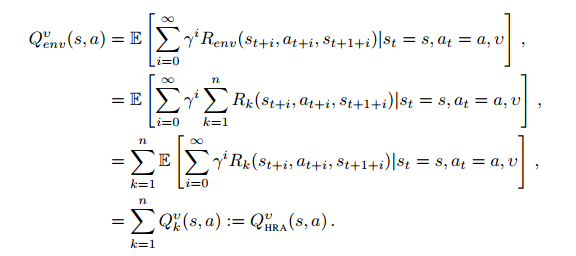


Bằng cách giảm thiểu các chức năng này mất, Thủ trưởng khác nhau của HRA xấp xỉ các chức năng hành động có giá trị tối ưu dưới các chức năng phần thưởng khác nhau: Q \* 1,. . . , Q \* n. Hơn nữa, QHRA xấp xỉ Q\* HRA, định nghĩa là:



chú thích *Q*\* nhân sựHRA Là khác nhau từ *Q*\**env* và nói chung là không phải thích hợp.

Một mục tiêu đào tạo thay thế là một trong đó là kết quả của việc đánh giá υ chính sách thống nhất ngẫu nhiên

dưới mỗi chức năng thưởng thành phần: Qυ (s, a): = Σ*n Qυ*(*S, a*).  bằng với QvHRA,   
*Q*-values ​​của chính sách ngẫu nhiên dưới Renv, như hình dưới đây:

HRA

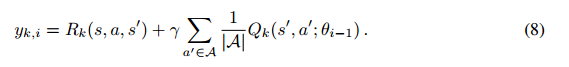
*k*= 1

*k*

HRA

*env*

mục tiêu đào tạo này có thể được học bằng cách sử dụng quy tắc cập nhật Sarsa dự kiến ​​(v[an Seijen et al., 2009),](#_bookmark37) bằng cách thay thế (7), với:



Hành động tham lam đối với Q-giá trị của một chính sách ngẫu nhiên với có thể xuất hiện để mang lại một chính sách mà chỉ là hơi tốt hơn so ngẫu nhiên, nhưng, surpringly, chúng tôi thấy rằng đối với nhiều lĩnh vực chuyển hướng dựa trên đóng vai trò như một mục tiêu đào tạo bán phù hợp.

## Cải thiện hiệu suất hơn nữa bằng cách sử dụng kiến ​​thức miền cấp cao.

Trong bối cảnh cơ bản của nó, những kiến ​​thức miền chỉ áp dụng cho HRA là ở dạng của hàm thưởng bị phân hủy. Tuy nhiên, một trong những thế mạnh của HRA là nó có thể dễ dàng khai thác kiến ​​thức miền hơn, nếu có. kiến thức miền có thể được khai thác trong một trong các cách sau:

* + 1. **Loại bỏ các tính năng không liên quan.** Tính năng mà không ảnh hưởng đến phần thưởng nhận được trong bất kỳ cách nào (trực tiếp hoặc gián tiếp) chỉ làm tăng thêm tiếng ồn để quá trình học tập và có thể được gỡ bỏ.
    2. **Xác định trạng thái thiết bị đầu cuối.** bang hẻm là tiểu bang mà từ đó không có phần thưởng có thể tiếp tục được nhận; họ có theo định nghĩa giá trị 0. Sử dụng kiến ​​thức này, HRA có thể kiềm chế xấp xỉ giá trị này bằng cách mạng giá trị, như vậy mà trọng lượng có thể được sử dụng đầy đủ để đại diện cho các quốc gia không bị đầu cuối.
    3. **Sử dụng chức năng giả thưởng.** Thay vì cập nhật một người đứng đầu của HRA sử dụng một thành phần của phần thưởng môi trường, nó có thể được cập nhật bằng cách sử dụng giả thưởng. Trong tình huống này, một bộ GVFs được đào tạo song song sử dụng giả thưởng.

Trong khi những phương pháp không đặc hiệu với HRA, HRA có thể khai thác kiến ​​thức miền đến một lớn hơn kéo dài, vì nó có thể áp dụng những phương pháp tiếp cận đến từng đầu độc lập. Chúng tôi hiển thị này theo kinh nghiệm tại Mục[4.1.](#_bookmark7)

# Các thí nghiệm

## Fruit Collection nhiệm vụ

Trong phạm vi đầu tiên của chúng tôi, chúng ta xem xét một đại lý có để thu thập các loại trái cây càng nhanh càng tốt trong một mạng lưới 10 10. Có 10 địa điểm ăn quả có thể, trải ra trên lưới điện. Đối với mỗi tập phim, một loại trái cây được đặt ngẫu nhiên trên 5 những 10 địa điểm. Các đại lý bắt đầu từ một vị trí ngẫu nhiên. Phần thưởng là 1

×

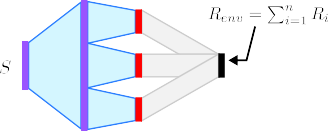
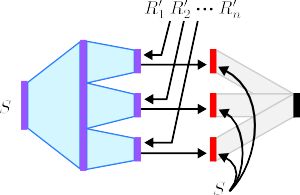
nếu một loại trái cây được ăn và 0 nếu ngược lại. Một tập phim kết thúc sau khi tất cả 5 loại trái cây đã được ăn hoặc sau 300 bước, nào đến trước.

chúng tôi so sánh hiệu suất của DQN với HRA sử dụng cùng một mạng. Đối với HRA, chúng phân hủy các chức năng phần thưởng thành 10 chức năng phần thưởng khác nhau, mỗi vị trí quả càng tốt. Mạng lưới bao gồm một lớp đầu vào nhị phân có độ dài 110, mã hóa vị trí của đại lý và cho dù có là một loại trái cây trên mỗi địa điểm. Tiếp theo là một lớp ẩn kết nối đầy đủ chiều dài 250. Lớp này được kết nối với 10 người đứng đầu gồm 4 nút tuyến tính từng, đại diện cho hành động-giá trị của 4 4 hành động dưới các chức năng phần thưởng khác nhau. Cuối cùng, giá trị trung bình của tất cả các nút trên đầu được tính toán sử dụng một lớp tuyến tính cuối cùng của chiều dài 4 kết nối đầu ra của các nút tương ứng ở mỗi đầu. Lớp này đã cố định trọng lượng với giá trị 1 (tức là, nó thực hiện phương trình[5).](#_bookmark1) Sự khác biệt giữa HRA và DQN là DQN cập nhật mạng từ lớp thứ tư sử dụng chức năng mất [(2),](#_bookmark0) trong khi HRA cập nhật mạng từ lớp thứ ba sử dụng chức năng mất [(6).](#_bookmark2)

**DQN**

**HRA**

**HRA với giả thưởng**



Hình 2: Các kiến ​​trúc mạng khác nhau được sử dụng.

Bên cạnh đó toàn bộ mạng, chúng tôi thử nghiệm sử dụng mức độ khác nhau của kiến ​​thức miền, như được nêu tại Mục [3.2:](#_bookmark6) 1) loại bỏ các tính năng không thích hợp cho mỗi đầu (chỉ cung cấp vị trí của các đại lý + tính năng quả tương ứng); 2) các tiểu bang thiết bị đầu cuối trên cộng với việc xác định; 3) phần thưởng giả trên cộng với sử dụng cho GVFs học tập để đi đến mỗi 10 địa điểm (thay vì học một hàm giá trị liên quan đến hoa quả tại mỗi địa điểm). Ưu điểm là những GVFs có thể được đào tạo ngay cả khi không có trái cây tại một địa điểm. Người đứng đầu cho một bản sao địa điểm cụ thể Q-giá trị của các GVF tương ứng nếu vị trí hiện chứa trái cây, hoặc kết quả đầu ra 0s khác. Chúng tôi đề cập đến điều này như HRA + 1, HRA + 2 và HRA + 3, tương ứng. Đối với DQN, chúng tôi cũng đã thử nghiệm một phiên bản đã được áp dụng cho các mạng tương tự như HRA + 1; chúng tôi đề cập đến phiên bản này như DQN + 1.

mẫu huấn luyện được tạo ra bởi một chính sách ngẫu nhiên; quá trình đào tạo được theo dõi bằng cách đánh giá các chính sách tham lam đối với hàm giá trị học sau mỗi tập phim với. Đối với HRA, chúng tôi thực hiện

experiments với Q \* HRA như mục tiêu đào tạo (sử dụng phương trình [7),](#_bookmark3) như cũng như Qυ

HRA

tương tự, vì DQN chúng tôi đã sử dụng các mặc định đào tạo Mục tiêu, *Q*\**env* , như tốt như Qυ

*env*

(Sử dụng phương trình [số 8).](#_bookmark5)

. Chúng tôi tối ưu hóa

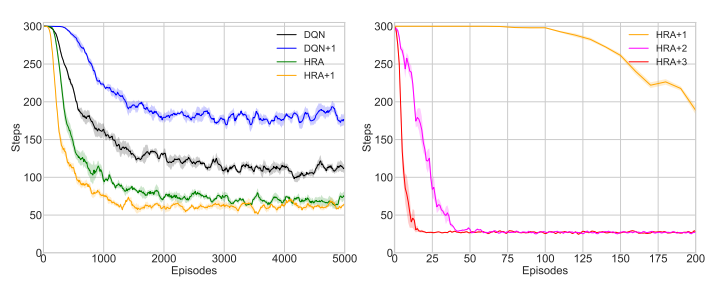
bước kích thước và các yếu tố giảm giá cho từng phương pháp riêng biệt.

Các các kết quả là thể hiện trong Nhân vật [3](#_bookmark8) vì các tốt thiết lập của mỗi phương pháp. Đối với DQN, sử dụng Q \* env

như mục tiêu đào tạo dẫn đến hiệu suất tốt nhất, trong khi đối với HRA, sử dụng Qυdẫn đến việc

HRA

hiệu suất tốt nhất. Nhìn chung, HRA cho thấy một tăng hiệu suất rõ ràng hơn DQN, mặc dù mạng là giống hệt nhau. Bên cạnh đó, bổ sung thêm hình thức khác nhau của kiến ​​thức miền gây cải thiện hơn nữa lớn. Trong khi đó, sử dụng một cấu trúc mạng tăng cường bởi kiến ​​thức miền cải thiện hiệu suất của HRA, sử dụng rằng cùng một mạng cho kết quả DQN đến sự sụt giảm trong hoạt động. Các sự thúc đẩy lớn trong hoạt động xảy ra khi các tiểu bang thiết bị đầu cuối được xác định là do các đại diện trở thành một vector một nóng. Do đó, chúng tôi loại bỏ các lớp ẩn và trực tiếp đưa vector một này.



*Hình 3: Kết quả trên lĩnh vực bộ sưu tập trái cây, trong đó một đại lý phải ăn 5 loại trái cây được đặt một cách ngẫu nhiên. Một tập phim kết thúc sau khi tất cả 5 loại trái cây được ăn hoặc sau 300 bước, nào đến trước.*

thành người đứng đầu khác nhau. Bởi vì người đứng đầu là tuyến tính, đại diện này làm giảm tới một chính xác, trình bày bảng. Đối với các đại diện bảng, chúng ta sử dụng cùng một bước kích thước như là bước kích thước tối ưu cho phiên bản mạng sâu.

## ATARI trò chơi: Ms. Pac-Man

Miền thứ hai của chúng tôi là các trò chơi Atari 2600 Ms. Pac-Man (xem Hình [4).](#_bookmark9) Điểm thu được bằng cách ăn bột viên, trong khi tránh những con ma (tiếp xúc với một nguyên nhân bà Pac-Man để mất một cuộc sống). Ăn một trong những viên sức mạnh đặc biệt biến những bóng ma màu xanh cho một khoảng thời gian nhỏ, cho phép chúng được ăn cho thêm điểm. Bonus trái cây có thể ăn cho điểm hơn nữa, hai lần cho mỗi cấp độ. Khi tất cả các hạt đã được ăn, một cấp độ mới được bắt đầu. Có tổng cộng 4 bản đồ khác nhau và 7 loại trái cây khác nhau, đều có một giá trị điểm khác nhau. Chúng tôi cung cấp đầy đủ chi tiết về các tên miền trong các tài liệu bổ sung.

**Đường cơ sở.** Trong khi phiên bản của chúng ta về Bà Pac-Man là giống như được sử dụng trong văn học, chúng ta sử dụng tiền xử lý khác nhau. Do đó, để kiểm tra ảnh hưởng của tiền xử lý của chúng tôi, chúng tôi thực hiện các phương pháp A3C[(Mnih](#_bookmark24) [et al., 2016)](#_bookmark24) và chạy nó với tiền xử lý của chúng tôi. Chúng tôi tham khảo các phiên bản với tiền xử lý của chúng tôi là 'A3C (kênh), phiên bản với tiền xử lý tiêu chuẩn' A3C (pixel), và điểm A3C của báo cáo trong văn học 'A3C (báo cáo).

**Sơ chế.** Mỗi khung từ ALE là 210 160 pixel. Chúng tôi cắt phần đáy và phần trên của màn hình để kết thúc với 160 160 pixel. Từ đó, chúng tôi trích xuất các vị trí của các đối tượng khác nhau và tạo cho mỗi đối tượng một kênh đầu vào riêng biệt, mã hóa vị trí của nó với độ chính xác 4 pixel. Điều này dẫn đến 11 kênh nhị phân của kích thước 40 40. Cụ thể, có một kênh cho bà Pac-Man, mỗi trong số bốn ma, mỗi trong số bốn con ma xanh (những được coi là đối tượng khác nhau),

×

×

×

trái cây cộng với một kênh với tất cả các bột viên (bao gồm cả viên sức mạnh). Đối với A3C, chúng tôi kết hợp 4 kênh của những bóng ma vào một kênh duy nhất, cho phép nó để khái quát tốt hơn trên những con ma. Chúng tôi cũng làm như vậy với 4 kênh của những bóng ma màu xanh. Thay vì đưa ra lịch sử của 4 khung hình cuối cùng là thực hiện trong văn học, chúng tôi cung cấp cho các định hướng của bà Pac-Man là một vector 1-nóng có độ dài 4 (đại diện cho 4 hướng la bàn).

**HRA kiến ​​trúc.** Các tín hiệu thưởng môi trường tương ứng với các điểm ghi được trong trò chơi. Trước khi phân hủy chức năng khen thưởng, chúng tôi thực hiện phần thưởng định hình bằng cách thêm một phần thưởng tiêu cực của

-1000 cho tiếp xúc với một con ma (gây Bà Pac-Man để mất một cuộc sống). Sau đó, phần thưởng được phân hủy trong một cách mà mỗi đối tượng trong game (viên / trái cây / ma / xanh ma) có chức năng phần thưởng của riêng mình. Do đó, có một đại lý RL khác nhau phù hợp với từng đối tượng trong trò chơi mà ước tính một hàm Q-giá trị của hàm số phần thưởng tương ứng của nó.

Đến ước tính mỗi chức năng thưởng thành phần, chúng ta sử dụng ba hình thức của tri thức miền thảo luận trong Phần [3.2.](#_bookmark6)HRA sử dụng GVFs rằng học giả Q-giá trị (với các giá trị trong khoảng [0, 1]) để nhận đến một vị trí đặc biệt trên bản đồ (GVFs riêng biệt được học kinh nghiệm cho mỗi trong số bốn bản đồ). Ngược lại với nhiệm vụ thu thập hoa quả (Phần[4.1),](#_bookmark7) HRA học phần đại diện của mình trong quá trình đào tạo: nó bắt đầu với 0 GVFs và 0 đứng đầu cho bột viên. Bởi lang thang xung quanh mê cung, nó phát hiện ra địa điểm bản đồ mới có thể đạt được, dẫn đến GVFs mới được tạo ra. Bất cứ khi nào các đại lý tìm thấy một viên tại một địa điểm mới nó tạo ra một cái đầu mới tương ứng với thức ăn viên.

Q-giá trị cho một đối tượng (viên / trái cây / ma / xanh ma) được thiết lập để các pseudo Q-giá trị của các GVF tương ứng với vị trí của đối tượng (ví dụ, di chuyển đối tượng sử dụng khác nhau GVF mỗi lần), nhân với một trọng lượng được thiết lập bằng các phần thưởng nhận được khi đối tượng được ăn. Nếu một đối tượng không xuất hiện trên màn hình, tất cả Q-giá trị của nó là 0.

chúng tôi thử nghiệm hai loại tổng hợp. Người đầu tiên là một trong những tuyến tính rằng tổng Q-giá trị của tất cả người đứng đầu (xem phương trình[5).](#_bookmark1) Đối với một thứ hai, chúng ta lấy tổng của tất cả người đứng đầu sản xuất điểm, và bình thường hóa các kết quả Q-giá trị; Sau đó, chúng tôi thêm tổng của Q-giá trị của người đứng đầu của những bóng ma thường xuyên, nhân với một vector trọng lượng.

Thăm dò, chúng tôi thử nghiệm hai loại bổ sung thăm dò. Mỗi loại thêm một đầu thăm dò bổ sung cho các kiến ​​trúc. Loại thứ nhất, mà chúng ta gọi là đa dạng hóa, tạo ra ngẫu nhiên Q-giá trị, được rút ra từ một phân bố đều trên [0, 20]. Chúng tôi thấy rằng nó chỉ là cần thiết trong suốt 50 bước đầu tiên, để đảm bảo bắt đầu mỗi tập phim một cách ngẫu nhiên. Loại thứ hai, mà chúng ta gọi là đếm dựa trên, bổ sung thêm một tiền thưởng cho cặp trạng thái hành động chưa được khám phá rất nhiều. Nó được lấy cảm hứng từ giới hạn sự tự tin trên[(Auer et al., 2002).](#_bookmark14) Chi tiết đầy đủ có thể được tìm thấy trong các tài liệu bổ sung.

Đối với thử nghiệm cuối cùng của chúng tôi, chúng tôi thực hiện một đầu đặc biệt lấy cảm hứng từ văn học đốc điều hành bộ nhớ [(Fuster,](#_bookmark19) [năm 2003;](#_bookmark19) [Gluck et al., 2013).](#_bookmark20) Khi một cầu thủ trò chơi con người đạt đến tối đa khả năng nhận thức và thể chất của mình, ông bắt đầu tìm kiếm những tình huống thuận lợi hoặc thậm chí ổn định và nhớ kỹ chúng. quá trình nhận thức này thực sự được ghi nhớ một chuỗi các hành động (còn gọi là thói quen), và không nhất thiết phải là tối ưu. người đứng đầu điều hành bộ nhớ của chúng tôi ghi lại tất cả các chuỗi hành động mà dẫn đến vượt qua một mức độ mà không giết. Sau đó, khi phải đối mặt với cùng cấp, người đứng đầu đưa ra một giá trị rất cao để hành động ghi lại, để buộc lựa chọn của aggregator. Lưu ý rằng phiên bản đơn giản của bộ nhớ điều hành của chúng tôi không khái quát.

**số liệu đánh giá.** Có hai phương pháp đánh giá khác nhau được sử dụng trên tài liệu mà dẫn đến điểm số rất khác nhau. Bởi vì ALE là cuối cùng một môi trường xác định (nó thực hiện giả ngẫu nhiên sử dụng một bộ tạo số ngẫu nhiên mà luôn luôn bắt đầu với hạt giống giống nhau), cả hai chỉ số đánh giá nhằm mục đích để tạo ra ngẫu nhiên trong đánh giá để đánh giá phương pháp với hành vi generalising hơn cao hơn. Số liệu đầu tiên giới thiệu một hình thức nhẹ ngẫu nhiên bằng cách lấy một số ngẫu nhiên của những hành động không-op trước khi điều khiển được bàn giao cho các thuật toán học. Tuy nhiên, trong trường hợp của bà Pac-Man, trò chơi bắt đầu với một khoảng thời gian không hoạt động nhất định vượt quá số lượng tối đa các bước không-op, dẫn đến các trò chơi có một sự khởi đầu cố định sau khi tất cả. Số liệu thứ hai lựa chọn điểm xuất phát ngẫu nhiên dọc theo một quỹ đạo của con người và kết quả là ngẫu nhiên mạnh hơn rất nhiều, và không dẫn đến việc đánh giá bắt đầu ngẫu nhiên dự định. Chúng tôi coi những số liệu như 'bắt đầu cố định' và 'bắt đầu ngẫu nhiên'.

**Các kết quả.** Nhân vật [5](#_bookmark13)cho thấy các đường cong đào tạo; Bàn[1](#_bookmark10)cho thấy điểm số cuối cùng sau khi đào tạo. Điểm số bắt đầu cố định được báo cáo tốt nhất đến từ rơm ([Vezhnevets et al.,](#_bookmark38) [Năm 2016);](#_bookmark38) số điểm bắt đầu ngẫu nhiên báo cáo tốt nhất xuất phát từ kiến ​​trúc mạng Dueling [(Wang et al., Năm 2016).](#_bookmark39) Điểm số bắt đầu cố định của con người xuất phát từ [Mnih et al. (2015);](#_bookmark23) số điểm bắt đầu ngẫu nhiên của con người xuất phát từ [Nair et al. (2015).](#_bookmark25)Chúng tôi đào tạo A3C với giá 800 triệu khung. Bởi vì HRA học nhanh, chúng tôi đào tạo nó chỉ cho 5.000 tập, tương ứng với khoảng 150 triệu khung (chú ý các chính sách tốt hơn

|  |  |  |
| --- | --- | --- |
| phương pháp | đã sửa  khởi đầu | ngẫu nhiên  khởi đầu |
| báo cáo tốt nhất | 6673 | 2,251 |
| Nhân loại | 15.693 | 15.375 |
| A3C (báo cáo) | - | 654 |
| A3C (pixel) | 2.168 | 626 |
| A3C (kênh) | 2.423 | 589 |
| HRA | **25.304** | **23.770** |

Bảng 1: Điểm cuối cùng.

dẫn đến nhiều khung hình mỗi tập phim). Chúng tôi đã thử một vài thiết lập khác nhau cho HRA: có / không có tion normalisa- và có / không có từng loại thăm dò. Điểm số hiển thị cho HRA sử dụng sự kết hợp tốt nhất: với bình thường và với cả hai loại thăm dò. Tất cả kết hợp đạt hơn 10.000 điểm trong đào tạo, ngoại trừ sự kết hợp không có thăm dò nào cả, mà-không đáng ngạc nhiên-thực hiện rất kém. Với sự kết hợp tốt nhất, HRA không chỉ nhanh hơn so với các nhà nước-of-the-art trên cả hai số liệu, nó còn vượt trội đáng kể số điểm của con người, thuyết phục chứng minh sức mạnh của HRA.

So sánh A3C (pixel) và A3C (kênh) trong Bảng [1](#_bookmark10) cho thấy một kết quả đáng ngạc nhiên: trong khi chúng ta sử dụng tiền xử lý tiên tiến bằng cách tách hình ảnh màn hình vào các kênh truyền hình đối tượng có liên quan, điều này đã không thay đổi đáng kể hiệu suất của A3C.

Trong thí nghiệm chính thức của chúng tôi, chúng tôi kiểm tra tốt như thế nào HRA không nếu nó khai thác sự yếu kém của việc đánh giá số liệu cố định bắt đầu bằng cách sử dụng một phiên bản đơn giản của bộ nhớ điều hành. Sử dụng phiên bản này, chúng tôi không chỉ vượt qua con người cao điểm 266.330 điểm,[1](#_bookmark11) chúng tôi đạt được số điểm tối đa có thể của 999.990 điểm trong vòng chưa đầy 3.000 tập. Đường cong là chậm trong giai đoạn đầu tiên bởi vì mô hình này phải được đào tạo, nhưng mặc dù các cấp tiếp tục nhận được nhiều hơn và khó khăn hơn, việc thông qua mức độ tăng tốc bằng cách tận dụng đã biết bản đồ. Lấy điểm hơn là không thể, không phải vì các trò chơi kết thúc, nhưng vì tỷ số tràn đến 0 khi đạt một triệu điểm.[2](#_bookmark12)

# Thảo luận

Một trong những thế mạnh của HRA là nó có thể khai thác kiến ​​thức miền đến một mức độ lớn hơn nhiều so với phương pháp đơn đầu. Này được thể hiện rõ ràng bởi các nhiệm vụ thu thập hoa quả: trong khi loại bỏ các tính năng không liên quan cải thiện hiệu suất của HRA, hiệu suất của DQN giảm khi cung cấp cùng với kiến ​​trúc cùng một mạng. Hơn nữa, tách hình ảnh điểm ảnh thành nhiều kênh nhị phân chỉ làm cho một sự cải thiện nhỏ trong việc thực hiện A3C qua học tập trực tiếp từ pixel. Điều đó chứng tỏ lý do mà hiện đại đấu tranh RL sâu sắc với bà Pac-Man không liên quan đến học tập từ pixel; vấn đề cơ bản là hàm giá trị tối ưu cho bà Pac-Man không thể dễ dàng được ánh xạ tới một đại diện thấp chiều.

HRA giải quyết bà Pac-Man bằng cách học gần 1.800 chức năng giá trị chung. Điều này dẫn đến một sự cố mũ kích thước vấn đề: trong khi đầu vào không gian trạng thái tương ứng với các kênh truyền hình nhị phân là theo thứ tự 1077, mỗi GVF có một trạng thái không gian theo thứ tự của 103 quốc gia, đủ nhỏ để có thể đại diện mà không cần bất kỳ chức năng xấp xỉ. Trong khi chúng ta có thể đã sử dụng một mạng sâu sắc đối với đại diện cho mỗi GVF, sử dụng một mạng sâu sắc đối với những vấn đề nhỏ như vậy đau hơn nó

giúp, bằng chứng là các thí nghiệm trên lĩnh vực bộ sưu tập trái cây.

chúng tôi cho rằng nhiều nhiệm vụ thực tế cho phép thưởng phân hủy. Thậm chí nếu các chức năng phần thưởng chỉ có thể bị phân hủy trong hai hoặc ba thành phần, điều này đã có thể giúp đỡ rất nhiều, do sự sụt giảm theo hàm mũ của kích thước vấn đề phân hủy có thể gây ra.