

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

====o0o====

BÁO CÁO

PROJECT I

ĐỀ TÀI: Ứng dụng giải thuật di truyền trong trò chơi Rắn săn mồi

*Giảng viên hướng dẫn* : thầy Nguyễn Tuấn Dũng

*Sinh viên thực hiện*: Phạm Thành Đông

*Mã số sinh viên*: 20173020

Hà Nội, tháng1/2020

LỜI NÓI ĐẦU

Ngày nay, trí tuệ nhân tạo ngày càng có vai trò quan trọng trong cuộc sống con người, trí tuệ nhân tạo từ khi ra đời đến nay đã giúp thay đổi rất nhiều trong cuộc sông của chúng ta, không chỉ giúp đỡ con người giải quyết nhiều vấn đề mà còn cải thiện cuộc sống của chúng ta… Vì vậy, tìm hiểu và học tập về trí tuệ nhân tạo là nhiệm vụ và sự cần thiết của không chỉ những người nghiên cứu trong ngành, các sinh viên mà còn là của mọi người trong xã hội.

Một trong những vấn đề nghiên cứu của trí tuệ nhân tạo chính là giải thuật di truyền. Giải thuật di truyền là một kỹ thuật của khoa học máy tính và Trí tuệ nhân tạo nhằm tìm kiếm giải pháp thích hợp cho các bài toán tối ưu tổ hợp (combinatorial optimization). Giải thuật di truyền là một phân ngành của giải thuật tiến hóa vận dụng các nguyên lý của tiến hóa như di truyền, đột biến, chọn lọc tự nhiên, và trao đổi chéo.

Nhằm mục đích tìm hiểu và ứng dụng về giải thuật di truyền, cũng như phục vụ mục đích học tập. Em xin trân trọng gửi tới thầy nội dung nghiên cứu: “Xây dựng trò chơi Rắn săn mồi có sử dụng giải thuật di truyền.”

Trong quá trình thực hiện còn nhiều thiếu sót, rất mong thầy giúp đỡ và góp ý để bài làm của em được tốt hơn.

Em xin chân thành cảm ơn !!!

MỤC LỤC

[LỜI NÓI ĐẦU 2](#_Toc30154692)

[CHƯƠNG I: TỔNG QUAN VỀ GIẢI THUẬT DI TRUYỀN 4](#_Toc30154693)

[1. Lịch sử giải thuật di truyền. 4](#_Toc30154694)

[2. Các đặc điểm, đặc trưng của giải thuật di truyền. 4](#_Toc30154695)

[3. Các thành phần của giải thuật di truyền. 4](#_Toc30154696)

[3.1 Khởi tạo quần thể ban đầu 4](#_Toc30154697)

[3.2. Đánh giá cá thể 4](#_Toc30154698)

[3.3 Toán tử lai ghép 5](#_Toc30154699)

[3.4 Toán tử đột biến. 5](#_Toc30154700)

[3.5. Điều kiện kết thúc thuật toán. 6](#_Toc30154701)

[4. Cấu trúc thuật toán di truyền tổng quát 6](#_Toc30154702)

[CHƯƠNG II: ỨNG DỤNG THUẬT TOÁN DI TRUYỀN TRONG TRÒ CHƠI RẮN SĂN MỒI 7](#_Toc30154703)

[1.Tổng quan trò chơi 7](#_Toc30154704)

[2.Áp dụng vào thuật toán 7](#_Toc30154705)

[2.1.Mạng lưới thần kinh 7](#_Toc30154706)

[2.2.Thuật toán di truyền 9](#_Toc30154707)

[CHƯƠNG III: CÀI ĐẶT CHƯƠNG TRÌNH VÀ KIỂM THỬ 11](#_Toc30154708)

[1.Cài đặt thuật toán 11](#_Toc30154709)

[1.1 Cài đặt môi trường 11](#_Toc30154710)

[1.2. Cài đặt mạng lưới thần kinh cho rắn ( neural\_network.py ) 11](#_Toc30154711)

[1.3. Cài đặt các điều kiện của môi trường huân luyện ( settings.py ) 13](#_Toc30154712)

[2. Giao diện trò chơi và kết quả 14](#_Toc30154713)

[TÀI LIỆU THAM KHẢO 15](#_Toc30154714)

CHƯƠNG I: TỔNG QUAN VỀ GIẢI THUẬT DI TRUYỀN

1. Lịch sử giải thuật di truyền.

Giải thuật di truyền đã được một số nhà sinh vật học đưa ra từ những năm 50-60, thế kỉ XX. A.S. Fraser là người tiên phong nêu lên sự tương đồng giữa sự tiến hóa của sinh vật và chương trình tin học giả tưởng về Genetic Algorithms(GA).

Tuy nhiên, chính John Henry Holland mới là người triển khai ý tưởng và phương pháp giải quyết vấn đề dựa theo sự tiến hóa. Từ những bài giảng, bài báo của mình, ông đã đúc kết các ý tưởng vào trong cuốn sách đầu tay Adaptation in Natural and Artificial Systems, xuất bản năm 1975. Dựa trên lý thuyết cơ bản về GA của Holand, Keneth De Jong đã triển khai và chứng minh những thành quả do ông thực hiện đã góp phần quan trọng trong việc tạo ra nền tảng toán học cho lý thuyết GA.

2. Các đặc điểm, đặc trưng của giải thuật di truyền.

Giải thuật di truyền đã mô phỏng sự chọn lọc tự nhiên và di truyền. Trong tự nhiên, các cá thể khỏe, có khả năng thích nghi với môi trường tốt sẽ được tồn tại và phát triển ở các thế hệ sau. Mỗi cá thể có cấu trúc gen đặc trưng cho tính chất của cá thể đó.

Trong quá trình sinh sản, các cá thể con có thể thừa hưởng các phẩm chất của cha mẹ, cấu trúc gen của nó mang một phần cấu trúc gen của cha mẹ. Ngoài ra, trong quá trình tiến hóa, có thể xảy ra hiện tượng đột biến, cấu trúc gen của cá thể con có thể chứa các gen mà cả cha mẹ đều không có.

Trong giải thuật di truyền, mỗi cá thể được mã hóa bởi một cấu trúc dữ liệu mô tả cấu trúc gen của cá thể đó, ta gọi nó là nhiễm sắc thể. Mỗi nhiễm sắc thể được tạo thành từ các đơn vị được gọi là gen.

Giải thuật di truyền sẽ làm việc trên các quần thể gồm nhiều cá thể. Một quần thể ứng với một giai đoạn phát triển gọi là một thế hệ.

Từ một thế hệ được tạo ra, giải thuật di truyền bắt chước sự chọn lọc tự nhiên và di truyền để biến đổi các thế hệ.

3. Các thành phần của giải thuật di truyền.

3.1 Khởi tạo quần thể ban đầu

Tạo quần thể đầu tiên trong giải thuật, là nơi xuất phát quá trình tiến hóa, bao gồm tất cả các giá trị thô ban đầu. Tùy theo vấn đề của bài toán mà có cách khởi tạo khác nhau.

3.2. Đánh giá cá thể

Chắc chắn rằng việc chọn cá thể sẽ thông qua kết quả, hay mục đích của vấn đề. Các cá thể tốt được chọn lọc để đưa vào thế hệ sau. Sự lựa chọn này được thực hiện dựa vào độ thích nghi với môi trường của mỗi cá thể.

Có nhiều phương pháp để chọn các nhiễm sắc thể tốt nhất, ví dụ: chọn lọc roulette wheel, chọn lọc xếp hàng, chọn lọc cạnh tranh, v.v…

3.3 Toán tử lai ghép

Toán tử lai ghép có trật tự bao gồm các bước sau:

- 1. Chọn ngẫu nhiên một chuỗi con từ một cá thể cha mẹ (parent).

- 2. Đưa ra một proto-child bằng cách sao chép chuỗi con vào những vị trí tương ứng như trong cá thể cha mẹ.

- 3. Xoá tất cả các ký hiệu từ cá thể cha mẹ thứ hai, lúc này đã có trong chuỗi con. Chuỗi còn lại chứa các ký hiệu mà proto-child cần.

- 4. Đặt các ký hiệu vào những vị trí không cố định của proto-child từ trái sang phải theo trật tự của chuỗi để tạo ra cá thể con.

Ví dụ:

Cá thể cha: 9 3 | 8 5 7 1 | 6 4 2

Cá thể con: 3 5 | 2 6 1 4 | 8 7 9

Đầu tiên, phân đoạn giữa để cắt các điểm được sao chép vào cá thể con.

Proto-child 1: x x | 8 5 7 1| x x x

Proto-child 1: x x | 2 6 1 4| x x x

Chuỗi bắt đầu từ điểm cắt thứ hai của cá thể cha mẹ thứ hai là: 8-7-9-3-5-2-6-1-4

Chuỗi sau khi loại bỏ các phần tử 8, 5, 7 và 1, cũng ở trong cá thể con đầu tiên là:

9-3-2-6-4

Cuối cùng, chuỗi này được đặt vào proto-child 1 đầu tiên để tạo ra cá thể con (bắt đầu từ điểm cắt thứ hai).

Cá thể con thứ nhất: 6 4 | 8 5 7 1 | 9 3 2

Tương tự, chúng ta được cá thể con khác:

Cá thể con thứ hai: 5 7 | 2 6 1 4 | 9 3 8

3.4 Toán tử đột biến.

+Đột biến đảo ngược(Inversion Mutation)

Chọn hai vị trí ngẫu nhiên trong một nhiễm sắc thể và sau đó, nghịch đảo chuỗi giữa hai vị trí này.

Ví dụ:

Nhiễm sắc thể : 9 3 8 5 7 1 6 4 2

Sau khi đột biến : 9 3 1 7 5 8 6 4 2

+Đột biến chèn (Insertion Mutation)

Chọn ngẫu nhiên một gen và sau đó chèn nó vào vị trí ngẫu nhiên.

Ví dụ:

Nhiễm sắc thể : 9 3 8 5 7 1 6 4 2

Sau đột biến: 9 3 5 7 8 1 6 4 2

+Đột biến thay thế (Displacement Mutation)

Chọn ngẫu nhiên một chuỗi con và chèn nó vào một vị trí ngẫu nhiên. Đột biến chèn có thể được xem như trường hợp đặc biệt của đột biến thay, trong đó, chuỗi con chỉ chứa một gen.

Ví dụ:

Nhiễm sắc thể: 9 3 8 5 7 1 6 4 2

Sau đột biến: 9 3 6 8 5 7 1 4 2

+Đột biến tương hỗ (Reciprocal Exchange Mutation)

Chọn ngẫu nhiên hai vị trí và sau đó hoán vị gen trên những vị trí này.

Ví dụ:

Nhiễm sắc thể: 9 3 8 5 7 1 6 4 2

Sau đột biến: 9 3 1 5 7 8 6 4 2

3.5. Điều kiện kết thúc thuật toán.

Thoát ra quá trình tiến hóa quần thể, dựa vào bài toán mà có các cách kết thúc vấn đề khác nhau một khi đạt đến mức yêu cầu. Một vài trường hợp thông thường như sau:

Kết thúc theo kết quả: một khi đạt đến mức giá trị yêu cầu thì chấm dứt ngay quá trình thực hiện.

Kết thúc dựa vào số thế hệ: chọn số thế hệ, quá trình sẽ dừng đúng ngay số thế hệ đã qui định trước, không cần biết kết quả như thế nào.

Tính theo thời gian: không cần biết đã bao nhiêu thế hệ hay kết quả nào, chỉ dựa vào số giờ qui định mà kết thúc.

Tổ hợp: dùng nhiều phương án khác nhau cho vấn đề, chẳng hạn như: chạy theo số thế hệ xong sau đó đánh giá cho chạy theo kết quả, hoặc ngược lại.

## **4. Cấu trúc thuật toán di truyền tổng quát**

Thuật toán di truyền bao gồm các bước sau:

Bước 1: Khởi tạo quần thể các nhiễm sắc thể.

Bước 2: Xác định giá trị thích nghi của từng nhiễm sắc thể.

Bước 3: Sao chép lại các nhiễm sắc thể dựa vào giá trị thích nghi của chúng và tạo ra những nhiễm sắc thể mới bằng các phép toán di truyền.

Bước 4: Loại bỏ những thành viên không thích nghi trong quần thể.

Bước 5: Chèn những nhiễm sắc thể mới vào quần thể để hình thành quần thể mới.

Bước 6: Nếu mục tiêu tìm kiếm đạt được thì dừng lại, nếu không trở lại bước 3

CHƯƠNG II: ỨNG DỤNG THUẬT TOÁN DI TRUYỀN TRONG TRÒ CHƠI RẮN SĂN MỒI

1.Tổng quan trò chơi

Trò chơi Rắn săn mồi (Snake) lần đầu ra mắt trên điện thoại Nokia 6110 (1997), do Teneli Armanto sáng tạo và trở thành cội nguồn cho những trò chơi điện tử hiện đại.

Quy tắc trò chơi: Bạn điều khiển con rắn của mình tránh va chạm và ăn vật phẩm để nâng điểm.

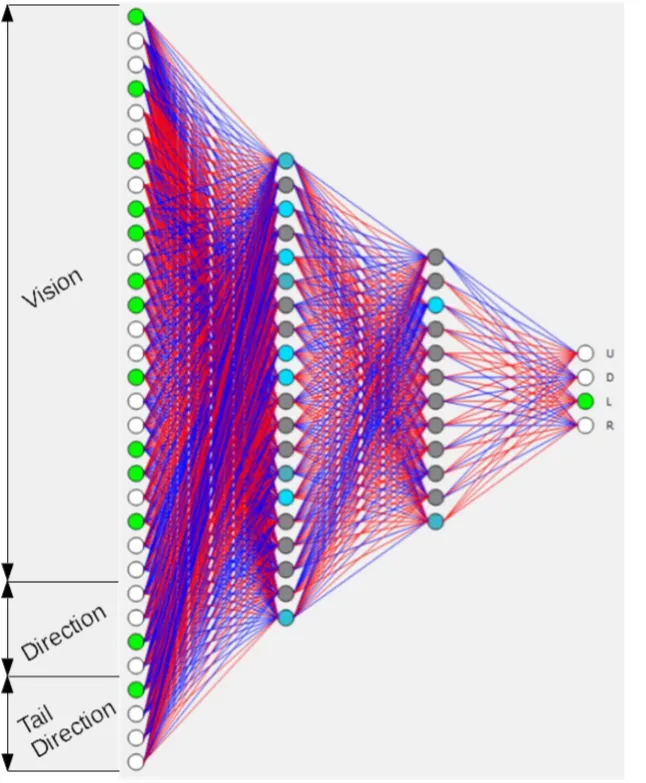
2.Áp dụng vào thuật toán

2.1.Mạng lưới thần kinh

Mạng lưới thần kinh phục vụ mục đích chính là quyết định những gì một con rắn sẽ làm. Bằng cách xử lý các phần dữ liệu của môi trường làm đầu vào, con rắn sẽ được chọn một hướng để di chuyển.

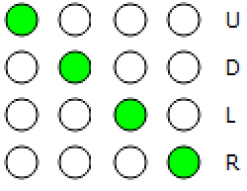
Một mạng lưới thần kinh cho một con rắn. Mỗi con rắn sẽ nhận được một Mạng thần kinh có cùng kiến ​​trúc (cùng các nút trong lớp đầu vào, lớp ẩn và lớp đầu ra). Trong trường hợp này, em xây dựng hai lớp ẩn. Các nút đầu vào là các đầu vào cảm biến của các điểm cảm ứng mà con rắn nhận được từ môi trường xung quanh. Các nút này chuyển qua Mạng đến lớp đầu ra nơi đưa ra một quyết định: hướng di chuyển(trái-L, phải-R, trên-D, dưới-U)

Hình ảnh một dạng của ba loại đầu vào mà mạng lưới thần kinh có thể nhận được: tầm nhìn(vision), hướng(direction) và hướng đuôi(tail direction).



Hình 1. Mạng lưới thần kinh rắn

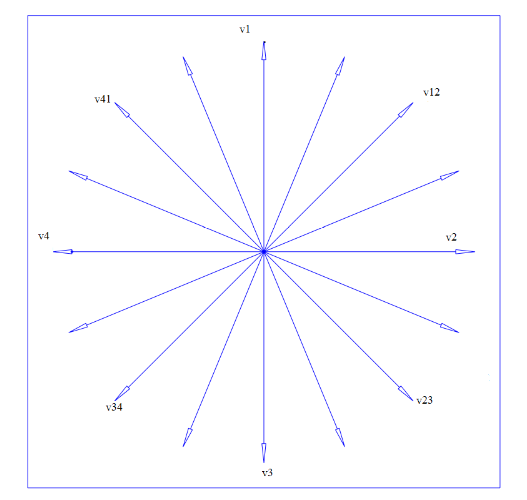
Hướng và hướng đuôi là cả hai biến được mã hóa động. Điều này có nghĩa là mặc dù chúng có thể đảm nhận nhiều giá trị, trong trường hợp này là bốn, chúng chỉ có thể có một nút hoạt động tại một thời điểm. Điều này cho phép lên, xuống, trái và phải để có đường đi riêng. Mã hóa tương tự được sử dụng cho lớp đầu ra với các giá trị có thể và ý nghĩa của chúng có thể được nhìn thấy bên dưới, trong đó mỗi cột sẽ là một thiết lập có thể khác nhau:



Hình 2. Các hướng được mã hóa một lần và ý nghĩa của chúng

*Dựa vào hình 1, có thể xác định rằng hướng bên trái (L) , hướng đuôi lên(chấm thứ nhất của phần đuôi) và con rắn đang quyết định đi sang trái (L) .*

Tầm nhìn của con rắn có thể là 4, 8 hoặc 16 hướng. Một minh họa về điều này có thể được nhìn thấy dưới đây:



Hình 3: Mô tả tầm nhìn 4,8 và 16

Nếu con rắn nhìn thấy là 4 hướng, thì nó chỉ nhìn thấy nơi có các đường lên, xuống, trái, phải (v1, v2, v3, v4). Nếu con rắn nhìn theo 8 hướng, thì nó sẽ thấy đường chéo (v12, v23, v34, v41). Nếu con rắn nhìn theo 16 hướng thì nó chỉ đơn giản nhìn thấy trong tất cả các đường trong hình.

Mỗi đường tầm nhìn có thể cho biết một trong ba điều sau:

1. Khoảng cách đến tường

2. Nó có nhìn thấy một quả táo không ?

3. Nó có nhìn thấy chính nó không ?

Nếu không có bất kỳ lớp ẩn nào, sẽ chỉ có các nút đầu vào được kết nối với các nút đầu ra, điều này sẽ không dẫn đến bất cứ điều gì. Nói chung, càng có nhiều lớp ẩn, có thể học được nhiều thứ phức tạp hơn.

2.2.Thuật toán di truyền

Mạng lưới thần kinh thường được sử dụng trong học tập có giám sát. Chúng ta có một số dữ liệu đào tạo, đưa vào Mạng thần kinh và để nó học các trọng số để giúp với dữ liệu không nhìn thấy. Với Snake AI, tuy nhiên, không có dữ liệu đào tạo. Dữ liệu huấn luyện có thể được sử dụng nhưng sẽ đòi hỏi rất nhiều công việc khi chơi Snake và ghi nhãn quyết định chính xác là gì. Bằng cách sử dụng Thuật toán di truyền, ta có thể coi trọng lượng là nhiễm sắc thể của các cá nhân và từ từ phát triển quần thể.

Đầu tiên, Thuật toán di truyền sẽ bắt đầu với một quần thể rắn được khởi tạo ngẫu nhiên. Những con rắn này lấy môi trường xung quanh làm đầu vào, đưa nó qua Mạng thần kinh của chúng và cuối cùng đưa ra quyết định. Sau khi tất cả các con rắn đã chơi, rắn bố mẹ được chọn ngẫu nhiên để sinh sản (chéo).

Một loại chéo cuối cùng mà em muốn sử dụng là chéo nhị phân một điểm (SPBX). Có thể nghĩ về SPBX như lấy một điểm trong nhiễm sắc thể gốc và trao đổi dữ liệu. Dưới đây là một ví dụ về SPBX trong đó điểm được biểu thị bằng màu đỏ:

Ảnh có chứa màu cam

Mô tả được tạo tự động

Hình 5. Crossover nhị phân một điểm

Lưu ý dữ liệu đầu vào vẫn giữ nguyên giữa parent 1và offspring 1 trái ngược với parent 2 và offspring 2. Trong trường hợp này, đại diện cho mỗi hộp như một gen. Tất cả điều này có thể được thay đổi trong settings.py, có thể chọn xác suất xảy ra SPBX so với SBX.

Khi con cái đã được tạo ra, mỗi gen trong mỗi nhiễm sắc thể của mỗi con có khả năng đột biến nhỏ. Nó chỉ có nghĩa là ngay cả khi một lần trao đổi chéo xảy ra, vẫn có cơ hội cho gen của con cái thay đổi. Quần thể tiếp theo sau đó có thể chỉ bao gồm con cái, được gọi là phương pháp *comma*, hoặc con cái và bố mẹ, được gọi là phương pháp *plus*. Điều này có thể được thay đổi trong settings.py nếu muốn. Phương pháp *comma* có thể tốt nếu có một không gian tìm kiếm cực kỳ lớn và đang cố gắng tập trung vào khám phá. Một ký hiệu khác cho điều này là *(num\_parents, lifespan, num\_offspring).* Điều này có nghĩa là chúng ta có một số cha mẹ được phép ở trong quần thể trong một tuổi thọ nhất định và tạo ra một số con cái. Vào cuối thế hệ tiếp theo, sau đó ta chọn đầu vào *num\_parents* bằng cách sử dụng lai để tạo con cái. Điều này sau đó được lặp lại cho đến khi chúng ta hài lòng với con rắn.

*Bởi vì thuật toán di truyền được mô hình hóa lỏng lẻo quá trình tiến hoá của các sinh vật, chúng có thể giúp phát triển từ từ một quần thể cá thể. Trong trường hợp này dân số bao gồm nhiều con rắn. Lúc đầu, những con rắn sẽ không biết gì - chỉ đơn giản là chạy vào tường, vô tình tự sát, v.v ... Cuối cùng, một con rắn sẽ tìm thấy một quả táo và nhận được phần thưởng. Hay nói rằng nếu hai con rắn tìm thấy táo trong một thế hệ. Nếu một trong những con rắn tìm thấy quả táo bằng cách đi lên, và con rắn khác tìm thấy quả táo bằng cách đi sang trái, có khả năng nếu chúng sinh sản, con cái sẽ biết cách đi lên và sang trái để ăn một quả táo! Điều này sau đó xảy ra trong nhiều thế hệ!*

CHƯƠNG III: CÀI ĐẶT CHƯƠNG TRÌNH VÀ KIỂM THỬ

1.Cài đặt thuật toán

* 1. **Cài đặt môi trường**

Cài đặt PyCharm tại đây: <https://www.jetbrains.com/pycharm/>

Cài đặt Python 3.6 tại đây: <https://www.python.org/downloads/>

1.2. Cài đặt mạng lưới thần kinh cho rắn ( neural\_network.py )

*import numpy as np*

*from typing import List, Callable, NewType, Optional*

*#mạng noron*

*ActivationFunction = NewType('ActivationFunction', Callable[[np.ndarray], np.ndarray])*

*# mạng thần kinh, dưới đây là các hàm sử dụng*

*sigmoid = ActivationFunction(lambda X: 1.0 / (1.0 + np.exp(-X)))*

*tanh = ActivationFunction(lambda X: np.tanh(X))*

*relu = ActivationFunction(lambda X: np.maximum(0, X))*

*leaky\_relu = ActivationFunction(lambda X: np.where(X > 0, X, X \* 0.01))*

*linear = ActivationFunction(lambda X: X)*

*class FeedForwardNetwork(object):*

*def \_\_init\_\_(self,*

*layer\_nodes: List[int],*

*hidden\_activation: ActivationFunction,*

*output\_activation: ActivationFunction,*

*init\_method: Optional[str] = 'uniform',*

*seed: Optional[int] = None):*

*self.params = {}*

*self.layer\_nodes = layer\_nodes*

*self.hidden\_activation = hidden\_activation*

*self.output\_activation = output\_activation*

*self.inputs = None*

*self.out = None*

*self.rand = np.random.RandomState(seed)*

*# Khởi tạo trọng số và độ lệch*

*for l in range(1, len(self.layer\_nodes)):*

*if init\_method == 'uniform':*

*self.params['W' + str(l)] = np.random.uniform(-1, 1, size=(self.layer\_nodes[l], self.layer\_nodes[l-1]))*

*self.params['b' + str(l)] = np.random.uniform(-1, 1, size=(self.layer\_nodes[l], 1))*

*else:*

*raise Exception('Implement more options')*

*self.params['A' + str(l)] = None*

*def feed\_forward(self, X: np.ndarray) -> np.ndarray:*

*A\_prev = X*

*L = len(self.layer\_nodes) - 1 # len(self.params) // 2*

*# Nạp các lớp ẩn*

*for l in range(1, L):*

*W = self.params['W' + str(l)]*

*b = self.params['b' + str(l)]*

*Z = np.dot(W, A\_prev) + b*

*A\_prev = self.hidden\_activation(Z)*

*self.params['A' + str(l)] = A\_prev*

*# Đầu ra thức ăn*

*W = self.params['W' + str(L)]*

*b = self.params['b' + str(L)]*

*Z = np.dot(W, A\_prev) + b*

*out = self.output\_activation(Z)*

*self.params['A' + str(L)] = out*

*self.out = out*

*return out*

*def softmax(self, X: np.ndarray) -> np.ndarray:*

*return np.exp(X) / np.sum(np.exp(X), axis=0)*

*def get\_activation\_by\_name(name: str) -> ActivationFunction:*

*activations = [('relu', relu),*

*('sigmoid', sigmoid),*

*('linear', linear),*

*('leaky\_relu', leaky\_relu),*

*('tanh', tanh),*

*]*

*func = [activation[1] for activation in activations if activation[0].lower() == name.lower()]*

*assert len(func) == 1*

*return func[0]*

1.3. Cài đặt các điều kiện của môi trường huân luyện ( settings.py )

*import numpy as np*

*# cài đặt các thông số*

*settings = {*

*'board\_size': (10, 10), # Khung trò chơi của rắn*

*# Kích hoạt lớp ẩn dành riêng cho các lớp ẩn, tức là tất cả các lớp ngoại trừ đầu vào và đầu ra*

*'hidden\_layer\_activation': 'relu', # chọn [relu, sigmoid, tanh, linear, leaky\_relu]*

*# Kích hoạt lớp đầu ra là dành riêng cho lớp đầu ra*

*'output\_layer\_activation': 'sigmoid', # chọn [relu, sigmoid, tanh, linear, leaky\_relu]*

*# Kiến trúc mạng ẩn mô tả số lượng nút trong mỗi lớp ẩn*

*'hidden\_network\_architecture': [20, 12], # Một danh sách chứa số nút trong mỗi lớp ẩn*

*# Số hướng mà con rắn có thể "nhìn thấy" trong trò chơi*

*'vision\_type': 8, # chọn [4, 8, 16]*

*## Đột biến ##*

*# Tỷ lệ đột biến là xác suất một gen nhất định trong nhiễm sắc thể sẽ đột biến ngẫu nhiên*

*'mutation\_rate': 0.05, # giữa [0.00, 1.00)*

*# Nếu loại tỷ lệ đột biến là tĩnh, thì tỷ lệ đột biến sẽ luôn là 'mutation\_rate',*

*# nếu không, nó sẽ phân rã, nó sẽ giảm khi số lượng thế hệ tăng lên*

*'mutation\_rate\_type': 'static', # chọn [static, decaying]*

*# Xác suất xảy ra nếu đột biến xảy ra, đó là gaussian*

*'probability\_gaussian': 1.0, # giữa [0.00, 1.00]*

*# Xác suất xảy ra nếu đột biến xảy ra, đó là sự ngẫu nhiên*

*'probability\_random\_uniform': 0.0, # giữa [0.00, 1.00]*

*## Crossover ##*

*# eta liên quan đến SBX. Các giá trị lớn hơn tạo ra sự phân phối gần hơn với cha mẹ trong khi các giá trị nhỏ hơn liên quan đến chúng nhiều hơn.*

*'SBX\_eta': 100, # Chỉ được sử dụng nếu xác suất\_SBX> 0,00*

*# Xác suất xảy ra khi sự giao nhau xảy ra, nó được mô phỏng chéo*

*'probability\_SBX': 0.5,*

*# Loại SPBX cần xem xét. Nếu nó là 'r' thì nó làm phẳng một mảng 2 chiều theo thứ tự #chính của hàng.*

*# Nếu SPBX\_type là 'c' thì nó sẽ làm phẳng một mảng 2 chiều theo thứ tự chính của cột.*

*'SPBX\_type': 'r', # chọn r hoăc c*

*# Xác suất xảy ra khi sự giao nhau xảy ra, đó là sự giao nhau nhị phân một điểm*

*'probability\_SPBX': 0.5,*

*# Loại lựa chọn chéo xác định cách chúng ta chọn các cá nhân cho chéo*

*'crossover\_selection\_type': 'roulette\_wheel',*

*## Lựa chọn phối giống ##*

*'num\_parents': 500, # Số lượng cha mẹ sẽ được sử dụng để tái sinh sản*

*'num\_offspring': 500, # Số con sẽ được tạo (num\_offspring> = num\_parents)*

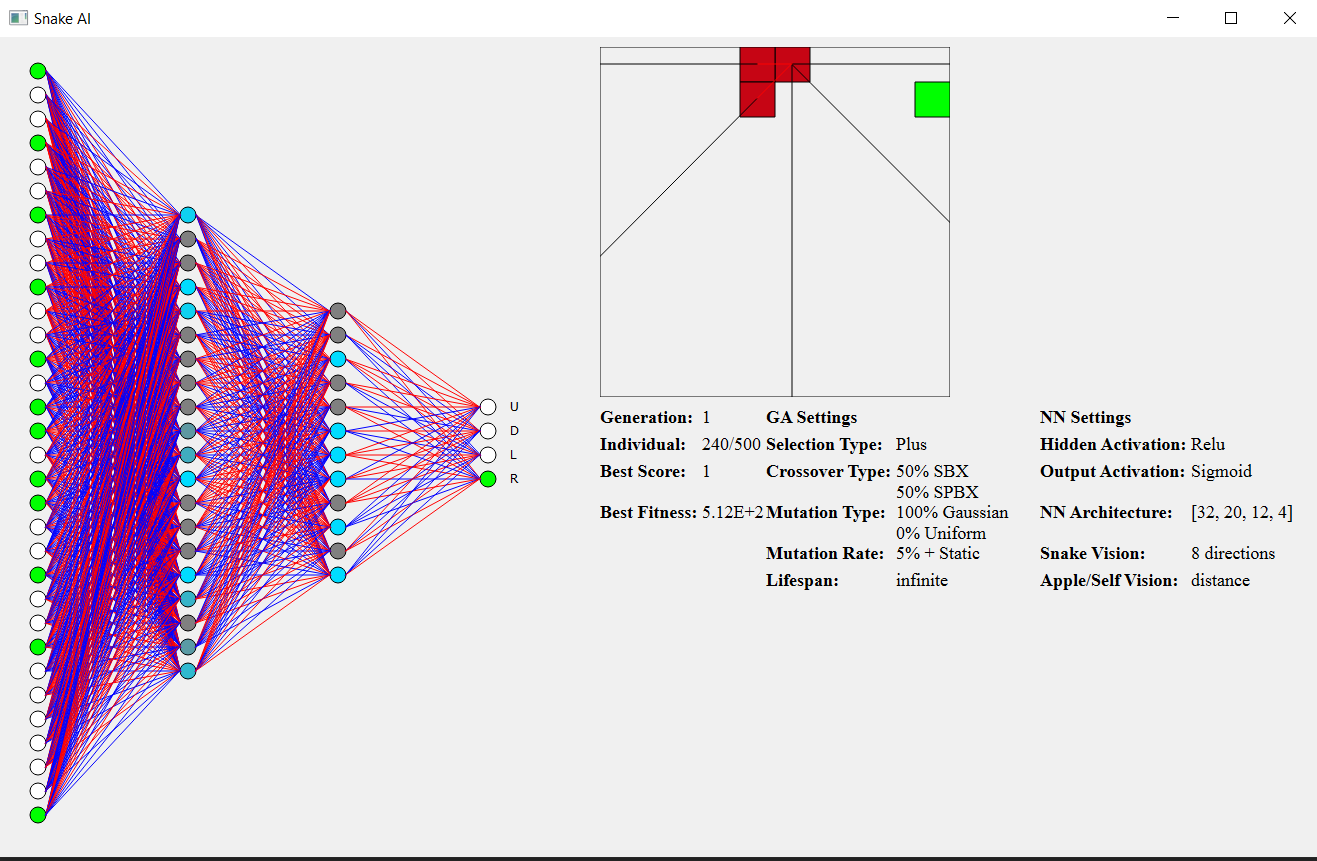
*'selection\_type': 'plus', # có thể là ['plus', 'comma']*

*'lifespan': np.inf,*

*'apple\_and\_self\_vision': 'distance' # có thể là ['binary', 'distance']*

*}*

2. Giao diện trò chơi và kết quả



Hình 6:Mô tả trò chơi

*Chúng ta có thể thấy con rắn đang di chuyển sang phải và hướng đuôi xuống. Các đầu vào từ môi trường xung quanh sau đó được đưa qua Mạng thần kinh và quyết định được đưa ra cho con rắn đi bên trái trong trường hợp này.Các thông số khác có thể thấy như trong hình.*

Kết quả huấn luyện của con rắn sau 10 tiếng với điểm số cao nhất (Best Score) là 47 ứng với thế hệ thứ 435.

TÀI LIỆU THAM KHẢO

1. Lê Thanh Hương (2019). Tài liệu Nhập môn Trí tuệ nhân tạo. Đại học Bách Khoa Hà Nội.
2. Trần Hoài Linh (2019). Mạng noron và ứng dụng trong xử lý tín hiệu. Nhà xuất bản Bách Khoa Hà Nội.
3. Chrispresso (2019). AI Learns To Play Snake

<https://chrispresso.coffee/2019/09/22/ai-learns-to-play-snake/>

1. Wikipedia (2019). Giải thuật di truyền

<https://vi.wikipedia.org/wiki/Gi%E1%BA%A3i_thu%E1%BA%ADt_di_truy%E1%BB%81n>

Và một số tài liệu tham khảo khác.