Remove Watermark Wondershare PDFelement

Snake Ai

What is snake game ?

Sanke is a popular game from the old days where the player moves the snake inside a rectangular board where the snake grows in length when eating food which randomly spawn . The game concept relies on 2 obstacles the snake itself and the walls . Game ends if hitting the snake body , hitting a board wall , or the snake fills the whole board and there isn't a new place for food then the snake **WINS** ! 🎉

Wondershare PDFelement

Project Main Scope:

The project is about the popular snake game known worldwide and the goal is to get the highest possible score and the highest score i could get with this Ai agent was 83 after 2 hours of training loops and after 1000+ Games . so the purpose of the game is the snake to learn to eat food and trying not to die by colliding with the wall or it body . i conquered this problem by using deep Q-Learning approch while this game has alot of different ai solvers approches . I Started by doing the normal snake game and make sure that all the logic is good and u can test it in a seperate python file as i will mention it below . then i started thinking about the deep Q-Learning algorithm so i altered the game in play step to add the reward then creating the model with pytorch with 3 layers that will return an action (right , left , straight) that will move the snake upon it we will have a new game state that will enter the neural network and like that untill it dies.

Remove Watermark Wondershare PDFelement

Overview

Our goal is develop an ai agent to play snake game and acheive the highest score we can get . To solve snake game with artificial intellingence there are various ways mainly divided into 2 categories : **Domain Specific** and **General Purpose** .

- **Domain Specific :** domain specific solvers aim to maximize the output in a very specific task [Weak AI].

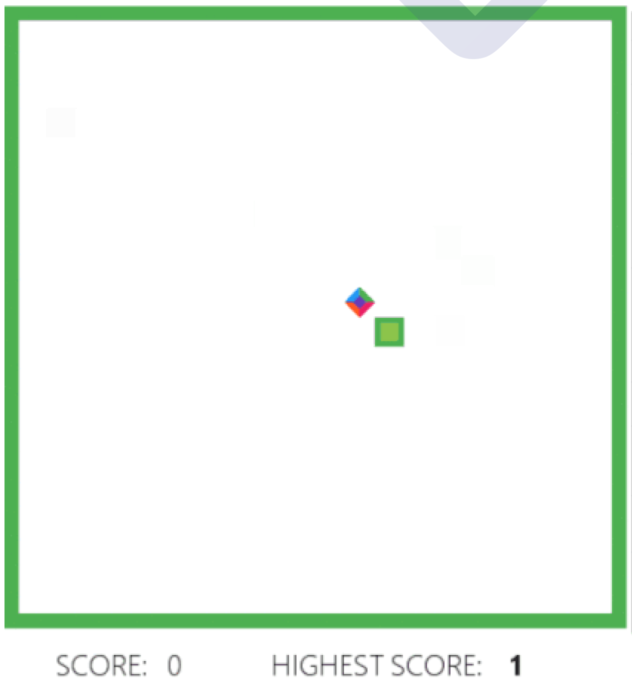
    - Shortest Path Algorithms
        - BFS (Breadth First Search)
        - DFS (Depth First Search)
        - A*
    - Longest Path Algorithms
    - Hamilton or Hamiltonian Cycle Algorithm
    - Genetic Evolution

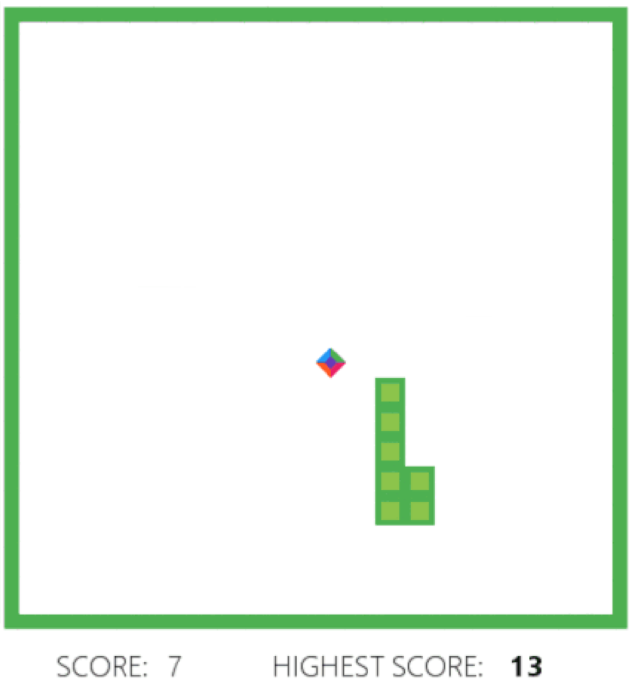
- **General Purpose :** general purpose ones don't aim for solving narrow tasks. Their goal is to be universal and solve a broader spectrum of problems .

    - Reinforcement Learning
        - Deep Q Learning

Wondershare PDFelement

In the following project we will use deep Q learning to solve snake game .





Remove Watermark Wondershare PDFelement

On the left, the AI does not know anything about the game. On the right, the AI is trained and learnt how to play . THIS IS NOT AN EXAMPLE FROM OUR CODE THIS GENERAL EXAMPLE FOR DEEP Q LEARNING ON SNAKE GAME

How Deep Q Learning Works ?

In Reinforcement Learning, we have two main components: the ***environment*** (our game) and the ***agent*** (our Snake). Every time the agent performs an action, the environment gives a **reward** to the agent, which can be positive or negative depending on *how good the action was from that specific state*. The goal of the agent is to learn what actions maximize the reward.

The game pipeline or the game algorithm works as follows:

- Q value is randomly initialized
- we get the current state of the game
- Action is taken from neural network
- a new state is acheived after performing the action
- update the Q-value with bellman formula
- saving game states in memory to re-train the neural network
- we repeat last 2 until game ends

Wondershare PDFelement

Project Components:

- **State:** A state is an input array to the neural network that represents information of the current state of the game upon it the neural network will perform an action that will lead to a new state . the state array consists of 11 boolean variables :
    - if there is danger around the snake (left , right , straight) [3]
    - snake's direction (up , down , left , right) [4]
    - food's position relative to snake's head (above , below , left , right) [4]
- **Loss:** The job of a neural network is to minimize the loss, to reduce the difference between the real target and the predicted one. the loss is expressed as:

Remove Watermark Wondershare PDFelement

$$loss = \left(r + \gamma \max_{a`} \hat{Q}(s,a`) - Q(s,a)\right)^2$$

- **Reward:** a **positive** reward is only given to the agent when it eats the food target **(+10)**. If the snake hits a wall or hits itself, the reward is **negative (-10)** .

- **Deep Neural Network (PyTorch):** Our Neural Netowrk Consists of 3 layers each layer with 512 (2^9^) nodes. The network receives as input the state, and returns as output three values related to the three actions: move left, move right, move straight.

Requirements and Dependcies :

- Python
- Pygame Module
- PyTorch

To Play the game go to the folder directory open cmd :

```
python snake_game_human.py
```

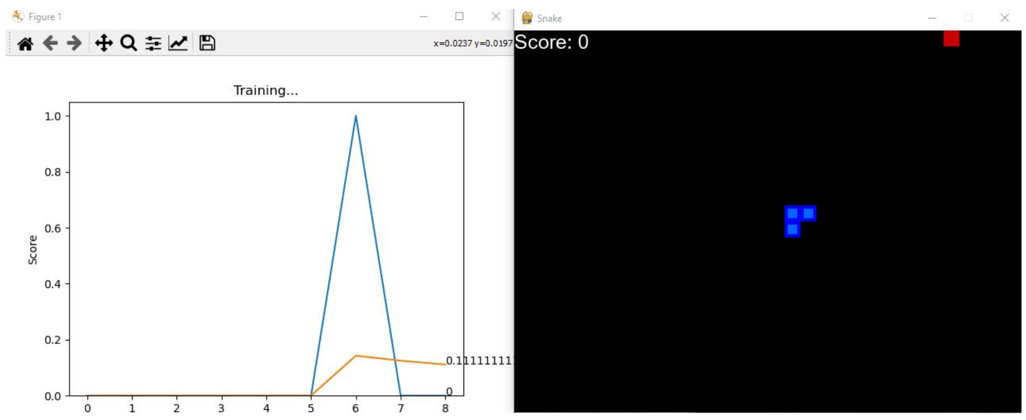
Wondershare PDFelement

To Run the agent :

```
python snake_agent.py
```

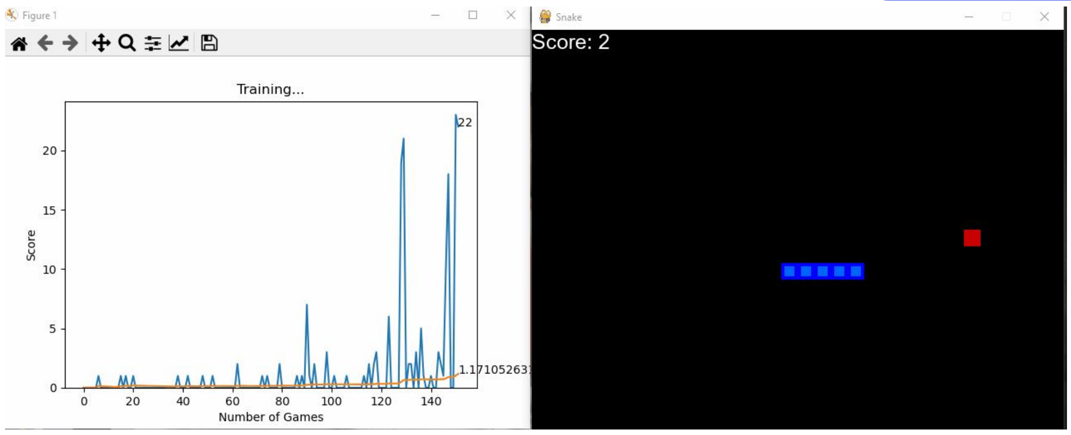
Game Output Samples:

Baby Snake:

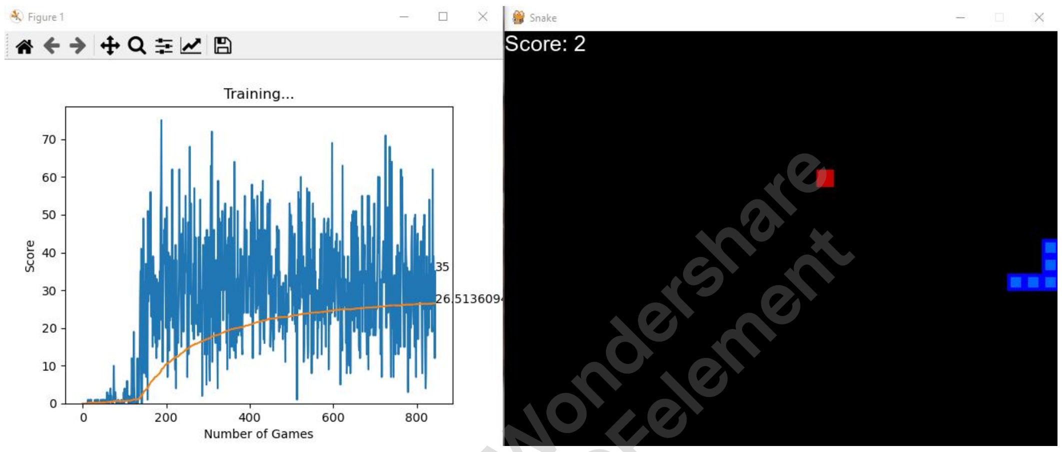


Remove Watermark Wondershare PDFelement

Slightly Learned Snake:



Advanced Snake:



Wondershare PDFelement