530 - Principles of AI
Snake Game using Reinforcement Learning

Yashshree Patil
Guided by Dr. Casimir Kulikowski

Motivation

There is no fun and engagement when the agents in our games outsmart human players.

(Uncertainty makes the game interesting)

A perfect AI will either win a game or the match would end in a draw.

Alpha Go, The First Reinforcement Learning computer program to defeat a professional human Go player.

Goal and Expected behavior

Goal: Build a snake game (to grab as many apples as possible while not walking into a wall or the snake’s body) such that the goal of the agent is to maximize the sum of the rewards during an episode/iteration.

Expected Behavior:
Initially, the agent explores a lot
Gathers Information
When the learning continues, exploration decreases and Agent chooses the action to perform.

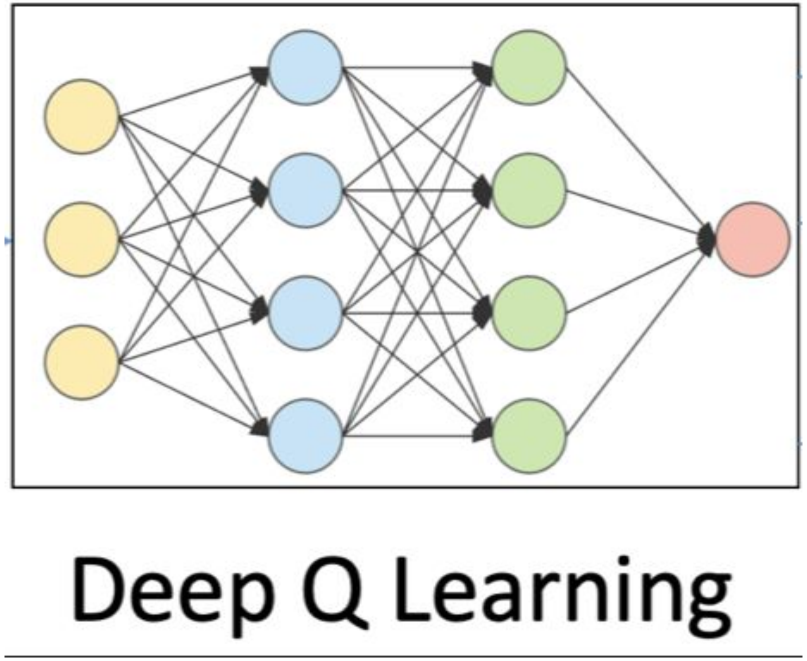
Model used: Deep Reinforcement Learning

Deep Reinforcement Learning also known as Deep Q Learning, a Combination of Deep Learning and Reinforcement Learning.

Input: State of the snake

Output: Action to perform

Reward: Reward for performing certain action (example Eating the apple will have a positive reward while dying will have a negative reward)



Algorithm:

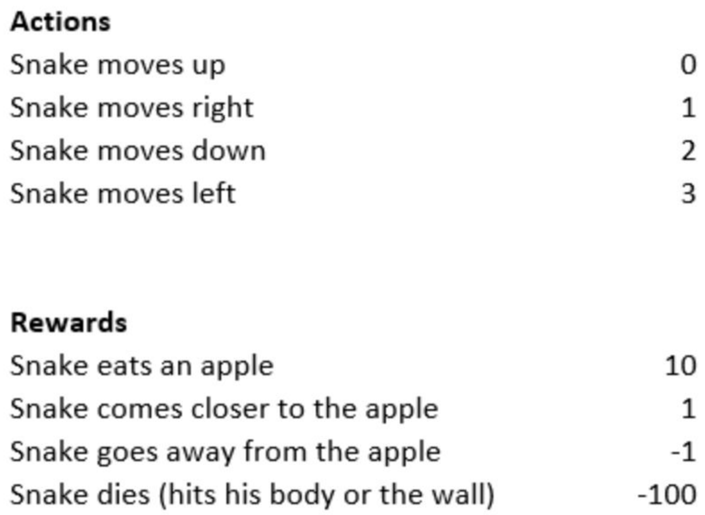
Step 1: Game starts with a random state value

Step 2: Initially system chooses random action, later exploration rate decays and relies more on Neural Network

Step 3: System is updated depending upon the actions performed (states are stored in replay previous experiences also known as memory)

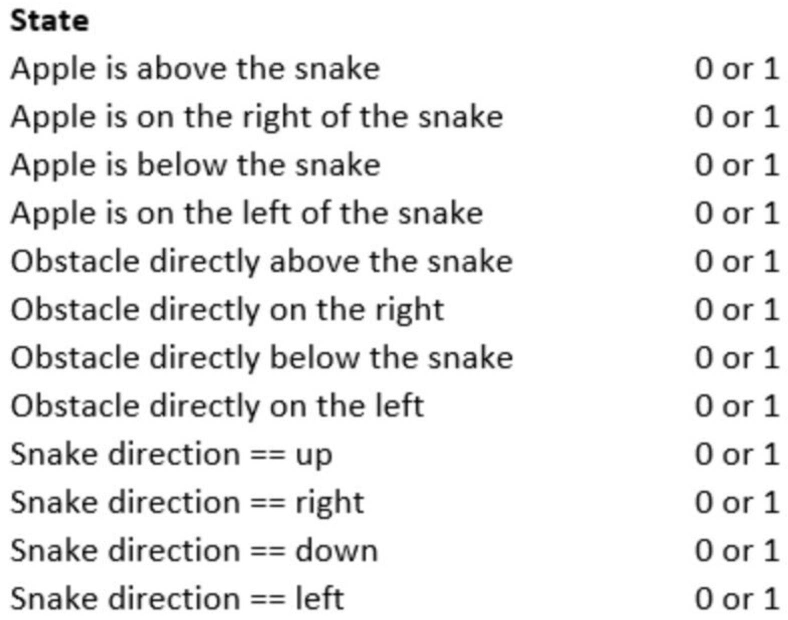
Step 4: Last 2 operations are repeated until the snake dies

Actions, Rewards and states



Actions	
Snake moves up	0
Snake moves right	1
Snake moves down	2
Snake moves left	3

Rewards	
Snake eats an apple	10
Snake comes closer to the apple	1
Snake goes away from the apple	-1
Snake dies (hits his body or the wall)	-100



State	
Apple is above the snake	0 or 1
Apple is on the right of the snake	0 or 1
Apple is below the snake	0 or 1
Apple is on the left of the snake	0 or 1
Obstacle directly above the snake	0 or 1
Obstacle directly on the right	0 or 1
Obstacle directly below the snake	0 or 1
Obstacle directly on the left	0 or 1
Snake direction == up	0 or 1
Snake direction == right	0 or 1
Snake direction == down	0 or 1
Snake direction == left	0 or 1

Epsilon sets the level of exploration and decreases over time :
param['epsilon'] = 1
param['epsilon_min'] = .01
param['epsilon_decay'] = .995

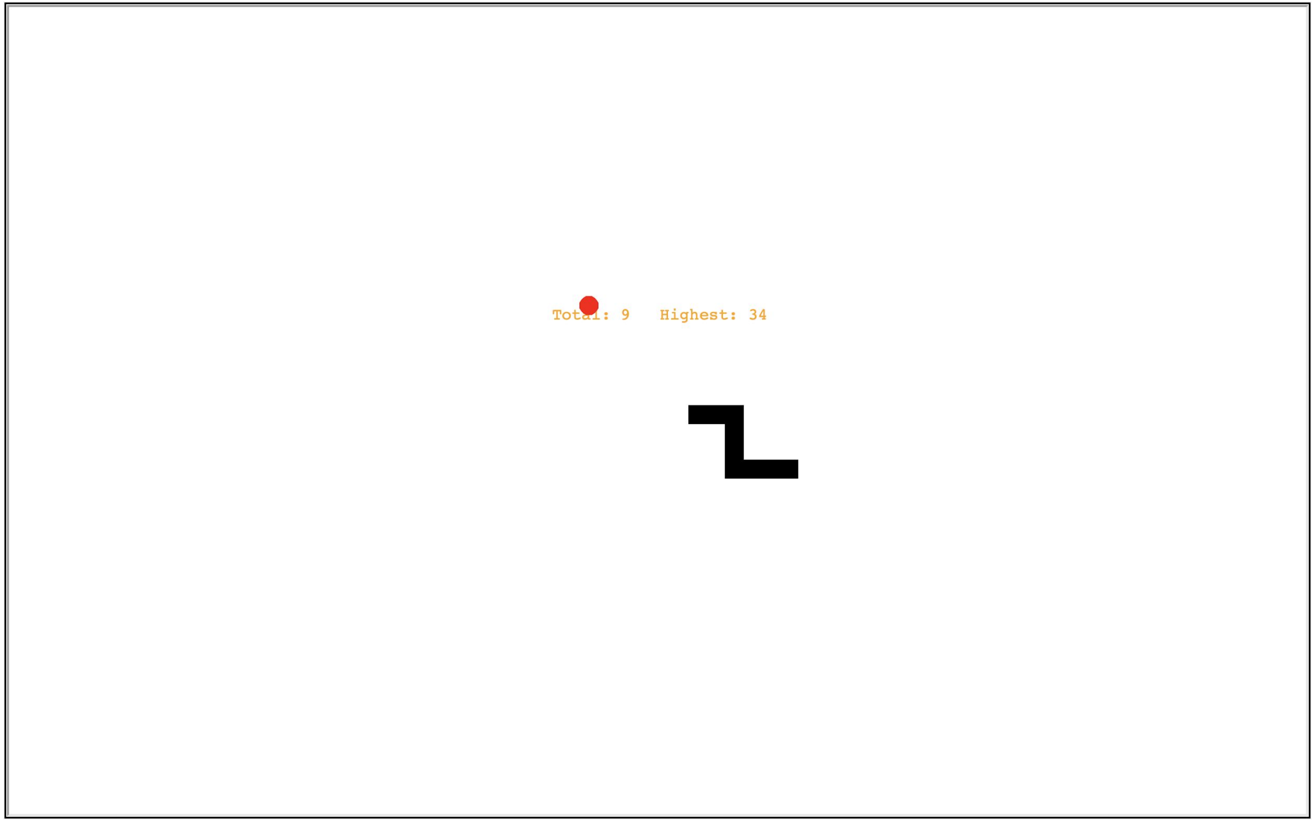
the batch size is needed for replaying previous experiences :
param['batch_size'] = 500
neural network parameters :
param['learning_rate'] = 0.00025

Problems Faced:

Reward system : At first I did not set reward for getting close to the apple. Therefore, it took time for the snake to eat the apple.

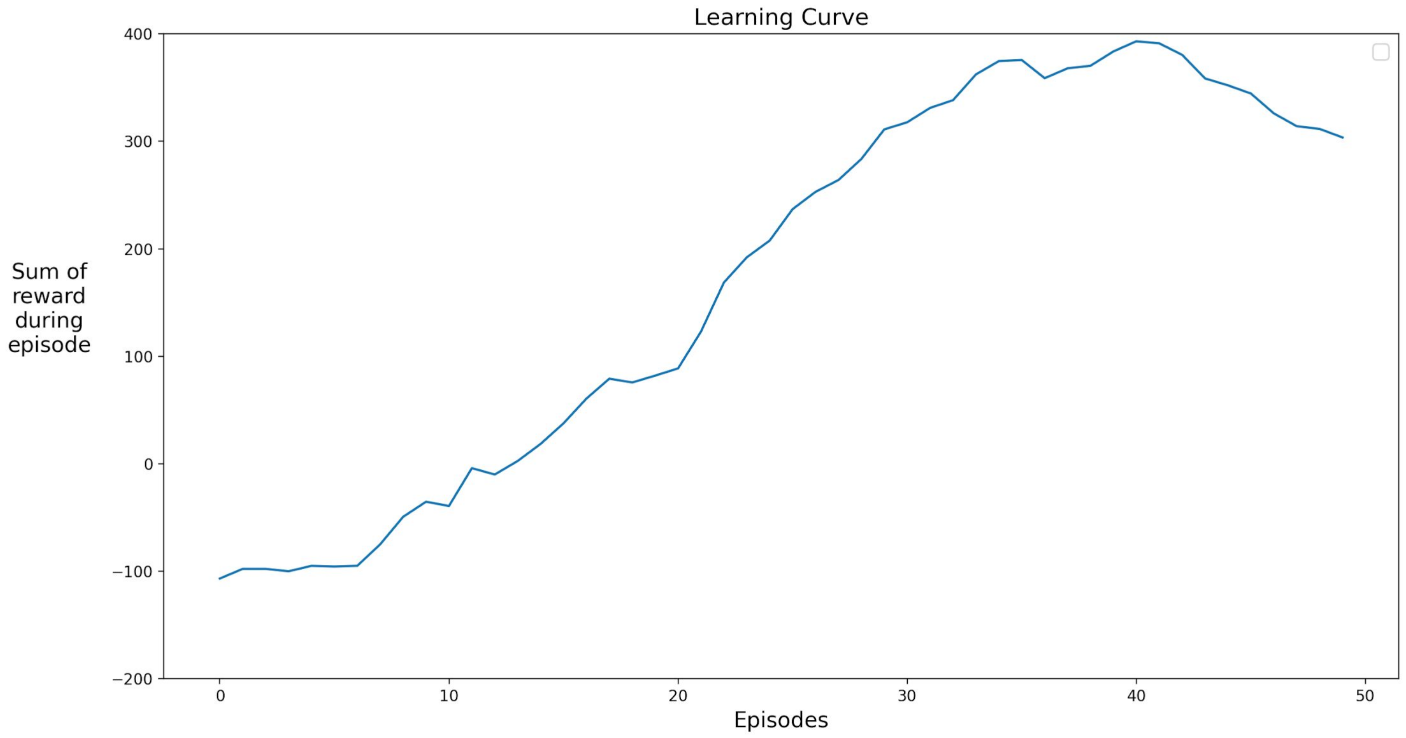
Solution to the problem: Changed the reward system.

Used Turtle feature in python for Graphics



A small video of the game

Graph of Sum of reward vs Episodes/number of games played



More validations can be performed by considering different state (trying to normalize the distance, state space only walls)

Conclusion :

Results and Conclusion : Max Number of apples eaten : 34+ apples eaten in 50 iterations with a score of 300-500 points.

THANK YOU