RL: Snake AI

1. Workflows

☑ ~~Snake game resource~~

☑ ~~Reinforcement learning algorithm for training~~

☑ ~~Visualize training~~

□ Optimal algorithm

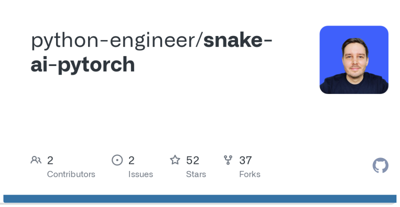
2. Project flow details

1. Snake game resource

snake-ai-pytorch/game.py at main · python-engineer/snake-ai-pytorch

Contribute to python-engineer/snake-ai-pytorch development by creating an account on GitHub.

https://github.com/python-engineer/snake-ai-pytorch/blob/main/game.py



python-engineer/**snake-ai-pytorch**

2 Contributors · 2 Issues · 52 Stars · 37 Forks

2. Algorithm

a. The reward

As the default the reward is calculated by the number of apples that the snake eaten. Using this default reward can make the model converges after a very long time since it's not define that the snake should eat the apple instead of going around on the map.

The solution is to create a term for forcing the snake eat the apple. Let's call s is the current state and s' is the next state after taking action a. If the action a gives the snake an apple, the reward to add to 20. Else the reward minus to an α, so that indirectly makes the snake focus to eat the snake than going around and keep alive. If the state s' doesn't exist (means the snake has reached to collision or ate itself) the reward minus to 10.

b. The state

I used a vector with 11 values. These are:

- The directions (LEFT, RIGHT, UP, DOWN)
- The danger zone (STRAIGHT, LEFT, RIGHT)
- The food location (LEFT, RIGHT, UP, DOWN)

c. The network

Using **Dueling Deep Q-Network.** As mentioned in this paper

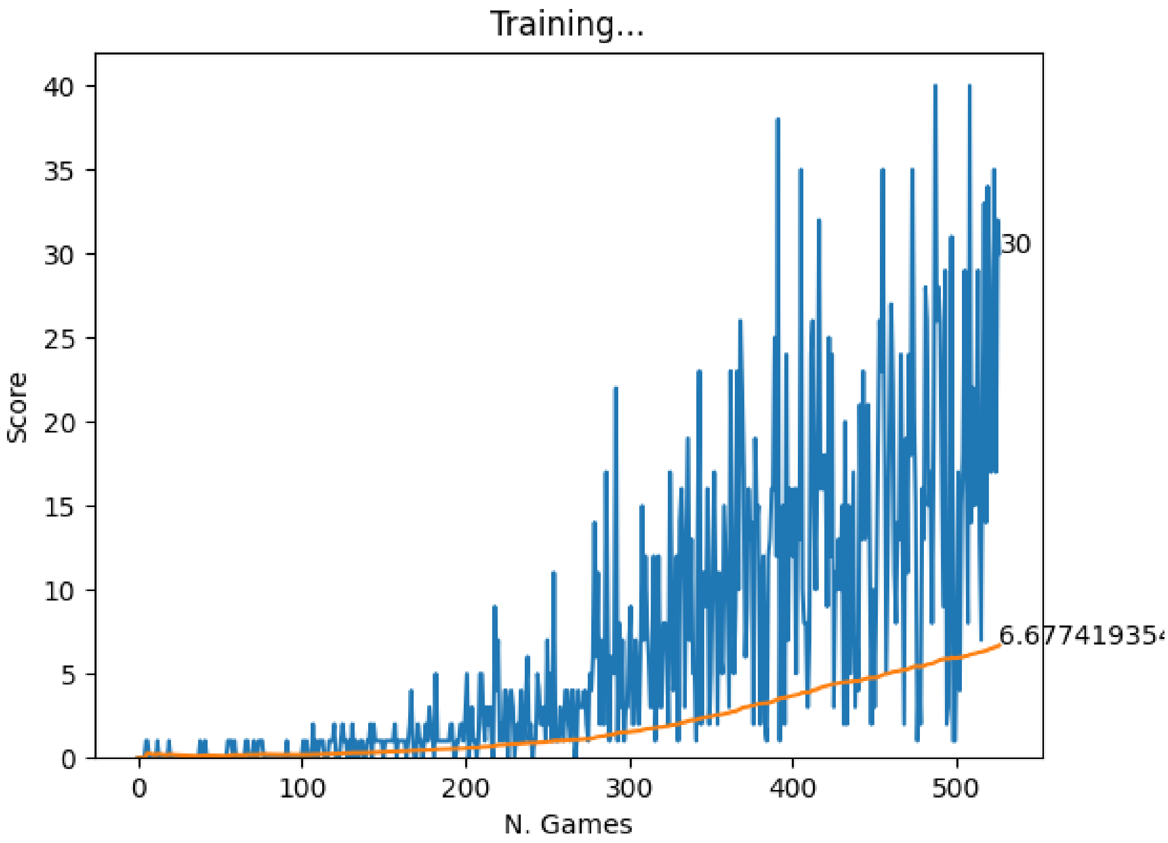
https://arxiv.org/pdf/1511.06581.pdf

d. Training algorithm

Using **target network** trick and **Experience Replay**.

3. Framework and training details

Pytorch is using on this project.



After more than 500 games, my snake has achieved up to 40 points.