

Classic Snake Game played by a Reinforcement Learning Agent

**Svetlana Gabdullina
Denis Rollov**

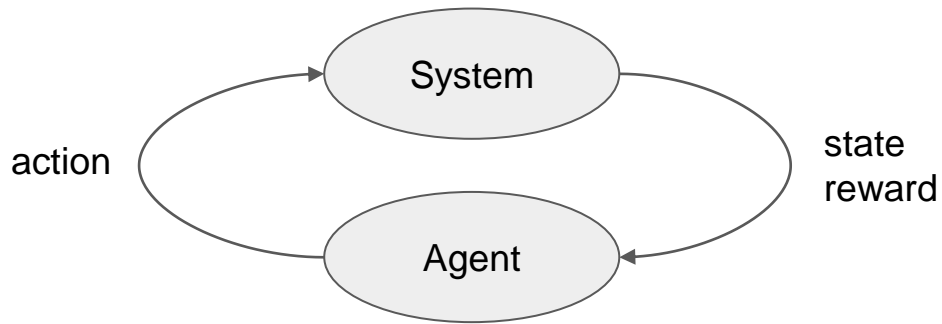
Approaches

1. Value Iteration
2. Policy Iteration
3. Using NN



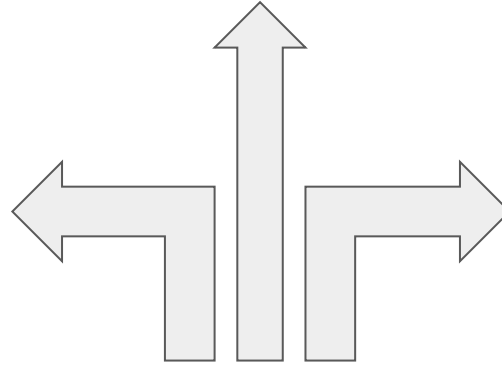
System

1. Action
2. State
3. Reward



Action

- 1. Left (-1)
- 2. Straight (0)
- 3. Right (1)



State

1. Snake's location
2. Food's location
3. Borders' location

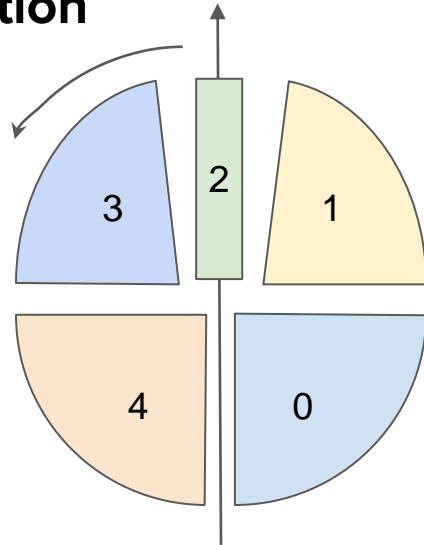


$O(n^2)$



State

1. If there is an obstacle further (True / False)³
2. Food's direction



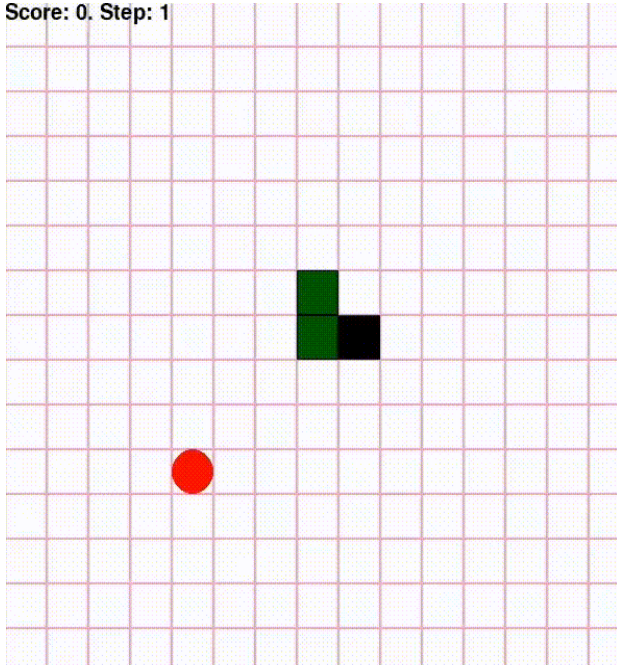
Reward

1. **+500**, if food is eaten
2. **-100**, if snake is dead
3. **+0**, if got closer to food
4. **-10**, otherwise

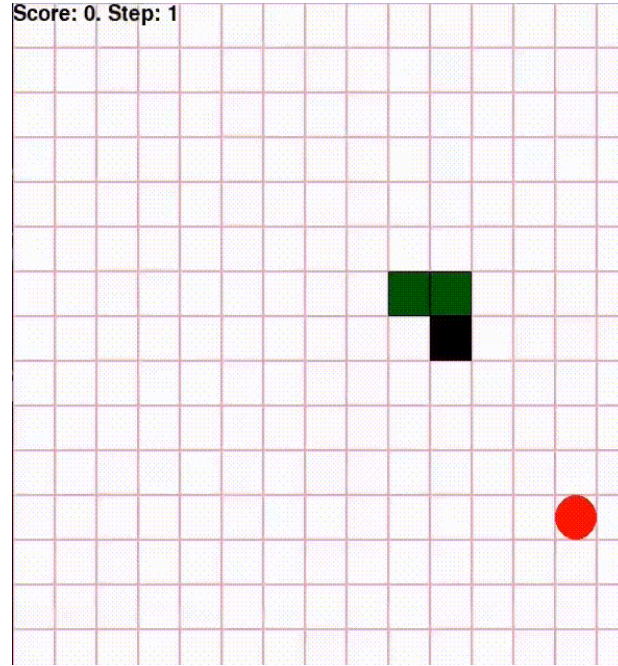


Value iteration / Policy iteration

Score: 0. Step: 1



Score: 0. Step: 1

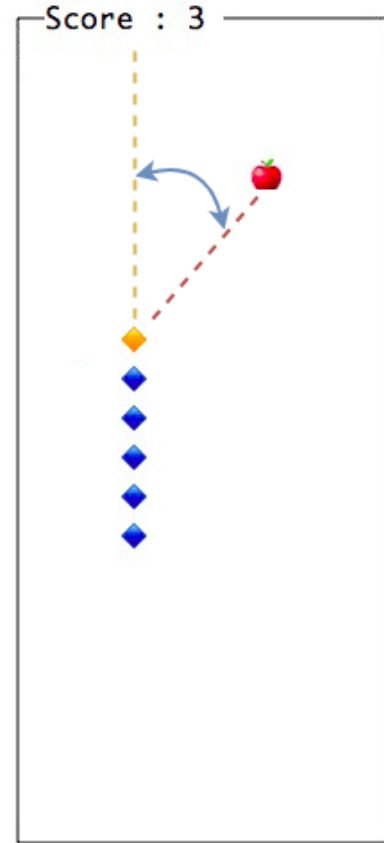


30.000 random games with up to 500 steps



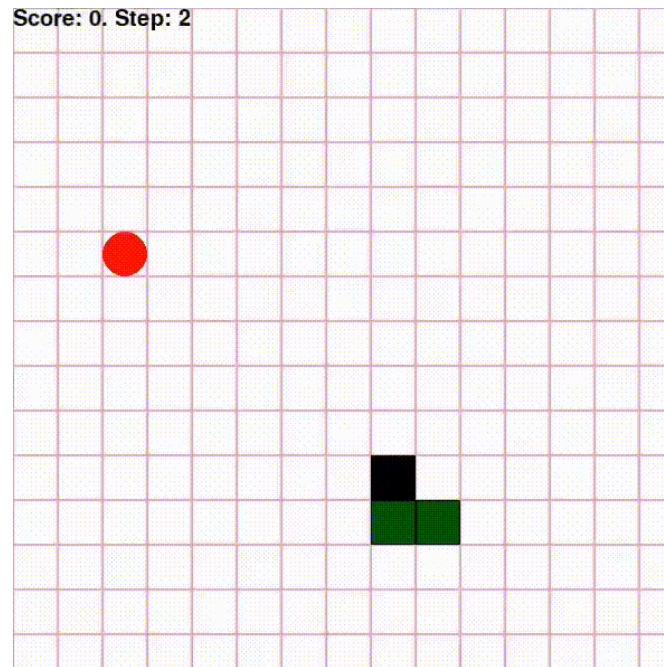
NN-approach

1. Dataset of (X, y)
2. X : $(l, s, r, \text{angle}, \text{action})$
3. y :
 - 1, if food is eaten
 - 1, if snake is dead
 - 0, otherwise



NN-approach

1. Model: MLP(hidden=15)
2. Problem: regression
3. Best action maximizes the prediction



Experiments

Field size: 15x15

Number of training games: 30000

	Value Iteration	Policy Iteration	MLP
Average number of steps	424	198	123
Average maximal length	22	23	12



Github repo



Github repo

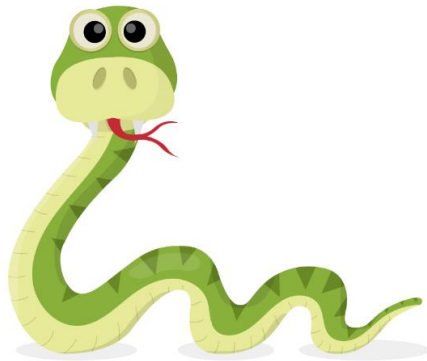


Scan or click the qr-code

Conclusions

1. It was interesting
2. Policy iteration method showed the best result
3. We definitely should try additional data sampling





Thank you for your attention!

**Svetlana Gabdullina
Denis Rollov**