IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

CHALLENGE IN PYRAT

Introduction to Artificial Intelligence

NGUYEN Binh Minh

SUMMARY

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1. Introduction

Introduction

- Target: Create an AI to help the agent win against the greedy algorithm.
- Methods: Reinforcement Learning (Q learning), Epsilon greedy strategy and Game theory combination.
- Winning rate achieved: 82%

2. Reinforcement Learning

Motivation

- In this challenge, the maze size is 21x15 and the number of cheeses is 40, therefore the amount of information for making a good movement dataset used for supervised methods is tremendously huge, which is hard to get (1000 games seems not to be enough because the distribution of 40 pieces of cheeses on the maze 21x15 can lead to many different situations in a game). Some supervised methods like SVM or Random Forests can not provide a good result (only 40% to 50% winning chance).
- About reinforcement learning, it can generate its own dataset through the interaction with environment and “learn” from the experiences in order the maximize the reward for “a game” (which is different from supervised learning because in supervised learning, each action is labeled and the machine try to get that right label, which means the algorithm only concentrates on each action rather than a game). Therefore, in this challenge, reinforcement learning is the suitable method to implement.

Q learning and training network

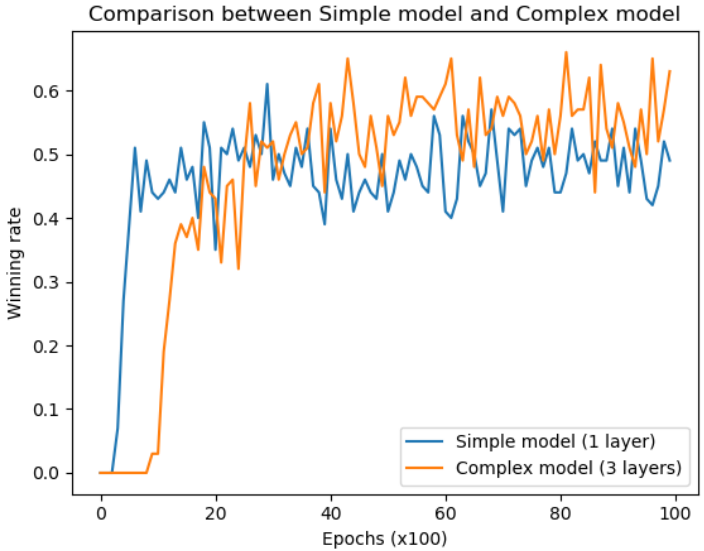
- In Q learning, we seek for the actions that maximize the rewards for a game.
- We create Q table and calculate Q values within it related to the states and actions.
- We update the Q values using the formula:

$$Q^*(s_i, a) = reward + \gamma * maxQ(s_{i+1}, a)$$

  where s_i and s_{i+1} is the current and next state, γ is discount factor.
- The actions were chosen as the arguments of maximum Q values.
- For training the data , I used a neural network with 3 layers, which achieved a higher winning rate than a simple network with 1 layer.
- A trade-off had to be made between the number of layers, neurons and the calculation time because obviously a complex model would require more time to train and calculate than a simple one (In this case, 3 layers can work well).

Q learning and training network

- As the result shown, a simple network will help us have a higher rate of learning (start learning at about 300 epochs) but produce a lower winning rate comparing with the complex one.
- Having more layers helps the model process more features of information, which leads to better final results. However, it requires more time to compute as well as making decisions.



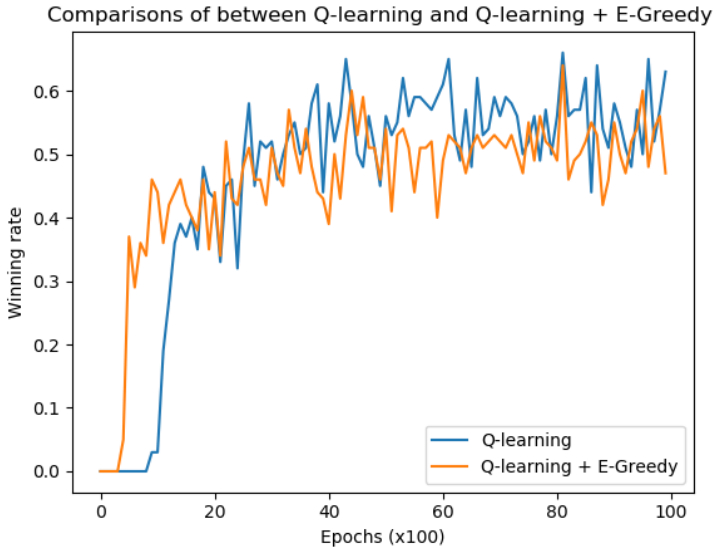
Epsilon Greedy Strategy

- In some cases, we want to explore the new choices of actions rather than always exploit a known strategy.
- In this challenge, I implemented the Epsilon Greedy Strategy in order to explore the potential of the other choices of actions.
- The explore probability was introduced to decide how to choose the next action.
- The explore probability was used as a threshold for **choosing a random action (explore)** and **choosing the action from Q-network (exploit)**.
- The motivation for applying this strategy is the problem of overestimations of action value in Q learning, which means that the actions chosen by selecting the positions of biggest values of Q might not be the optimal one, thus slower the learning process. Therefore we introduce the Epsilon Greedy Strategy in order to explore other actions which might be better and accelerate the learning speed.
- The decay step was also introduced and increased by the training time in order to reduce the explore probability in the later period of the training process (because at the time, the action selected by Q networks was better than a random one).

$$explore\ probability = \min probability + (max\ probability\ - \min probability) * e^{-decay\ rate * decay\ step}$$

Epsilon Greedy Strategy

- We can see that using the Epsilon Greedy Strategy helped us accelerate the learning time by exploring the potential choices of actions at the earlier time in the training process.
- In this challenge, this strategy helped to accelerate the learning time by 400 epochs.

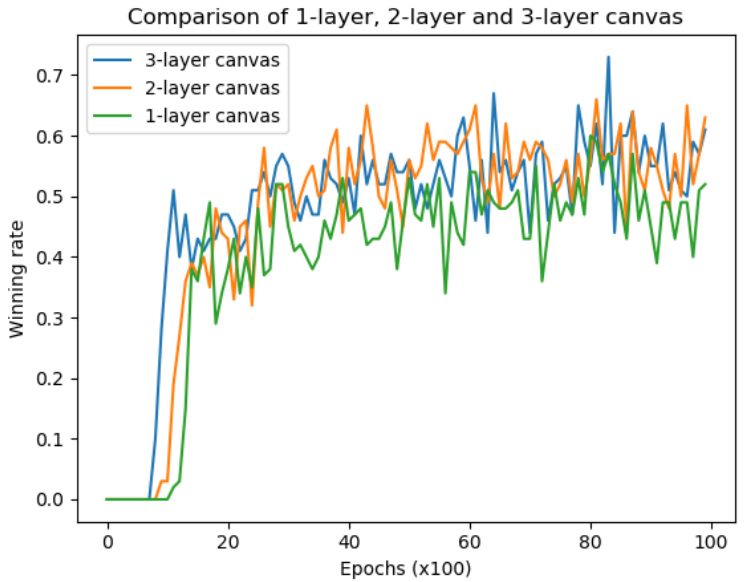


Canvas’ layers

- One problem that we should concern is the amount of information that we put in the canvas.
- Each layer contains different information which sometime does help to increase the winning rate and sometime does not.
- In this challenge, I set up 3 layers in the canvas. The first layer contains the positions of cheeses, the second one contains the opponent’s position and the final one contains player’s position (which is always in the center).

Canvas' layers

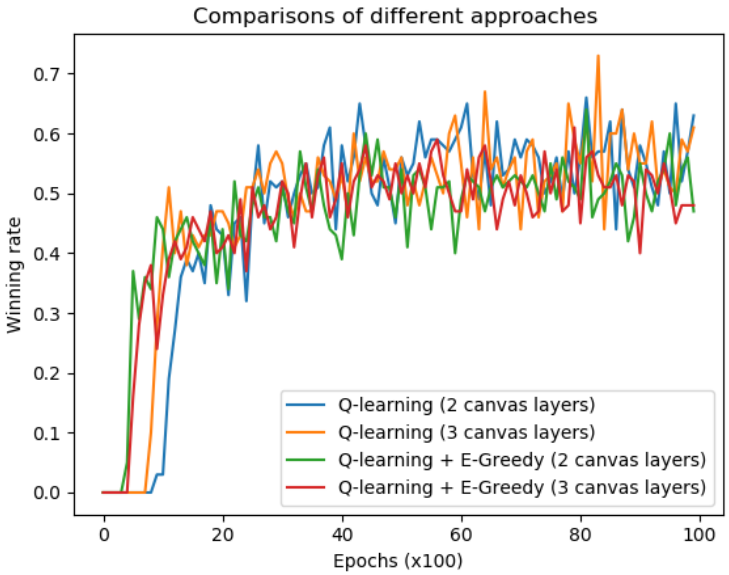
- We can see that using 2 or 3 layers would help the model achieve a better winning rate comparing to use only 1 layer because they give the model more features to process.
- The results of using 2 and 3 layers are not too different because the third layer only contains the position of player, and that position is always at the center of the canvas, which mean the information in this layer does not change (while the information in the other 2 layers changes regularly).



("1-layer" means the layer of cheeses' positions, "2-layer" means the cheeses' positions and the opponent's position)

Comparison of different approaches using Q learning.

- Final results obtained in the approaches were almost the same (0.55 to 0.6 winning rate). However, the approaches using epsilon greedy algorithm started the learning process faster than the non-using one due to the potential choices made by this algorithm at the early time of the training process.
- The results of approaches using E-greedy and the one which do not use are almost the same because the exploring random actions works regularly at the beginning of the training process. After some epochs, the explore probability becomes smaller due to the decay step, which means the training process choose to exploit the actions from Q-network more regularly.



3. Game Theory Combination

Motivation

- In Q learning, the maximum approximated action values when updating Q values can sometimes overestimate the actions values, which leads to the false choices of optimal actions. In this challenge, I have introduced the Epsilon Greedy Strategy, which will sometimes explore the new other choices (depend on the exploration probability), which have potential to be the optimal and it does help to accelerate the learning process.
- However, Epsilon Greedy Strategy does not help to increase the winning chance for the Q learning (in the later period of training process, almost all the actions were taken by the Q-network because at that time, after the network started learning, its decisions were better than random ones). As a result, I decided to implement the game theory into this challenge in order to find as many as possible the most rational actions and increase the winning rate.

Implementation

- Since the opponent’s strategy is deterministic, implementing game theory can help us calculate the best targets which will lead to the best outcome.
- To find the best targets, it is necessary to simulate the game to each of the pieces of cheeses, which will be done by testing recurrently all possible situations of the game.
- The constraints of the approach is the computation time because it needs to do the simulation of all possible situations, especially with a large number of cheeses. Therefore, I set a threshold of the number of cheeses in order to reduce the computation time. Particularly, the game theory strategy can only be played if there are fewer than 12 pieces of cheeses left.

Table of comparisons

Approach	Winning rate of player	Winning rate of opponent	Miss
Q learning (2- layer canvas)	0.567	0.312	0.669
Q learning (3- layer canvas)	0.574	0.314	0.765
Q learning + E-greedy (2- layer canvas)	0.552	0.327	0.599
Q learning + E-greedy (3- layer canvas)	0.57	0.315	0.701
Q learning + E-greedy + Game theory (2- layer canvas)	**0.822**	**0.107**	**0.0125**

Table of comparisons

- As expected, the combination of game theory and Q learning model gave a pretty high winning chance (82 %) because it always pointed to the best actions to play.
- The results of the other 4 approaches correspond exactly to the training results (shown in slide 12), which were not too much different. The miss indexes of the approaches using 2-layer canvas were always lower than using 3-layer one because of the number of input features of 2-layer canvas were fewer, which lead to less calculation time.
- When combing with game theory, I chose to use 2-layer canvas instead of 3 because the results of 2-layer canvas and 3-layer one were almost the same, and 2-layer canvas required less time to compute than the 3-layer canvas due to fewer features.

4. Conclusion

Conclusion

- In this challenge, I have implemented Q learning, Epsilon Greedy Strategy and Game theory to design an agent to win against the greedy algorithm. The winning chance obtained was 82%, which is satisfied the constrains of the challenge.

- Some improvements have been made such as modifying the network for learning and adding more information to the canvas.

- Future works: Try to implement some advanced AIs such as the one used for Alpha Go. Try to explore other reinforcement learning methods like Double Deep Q learning.

THANK YOU FOR YOUR ATTENTION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom