Tic-Tac-Toe with Reinforcement Learning

Jianhao Zheng, Yujie He
{jianhao.zheng, yujie.he}@epfl.ch

Considering the layout, we apply the following to this report. **For answers whose number of words $<$ 100, we will put them in the caption. Otherwise, we will write a short description of the figure in the caption + a more detailed answer in the main text.**

I. Q Learning

A. Learning from experts (fixed ϵ)

Question 1. Illustrated in Figure. 1

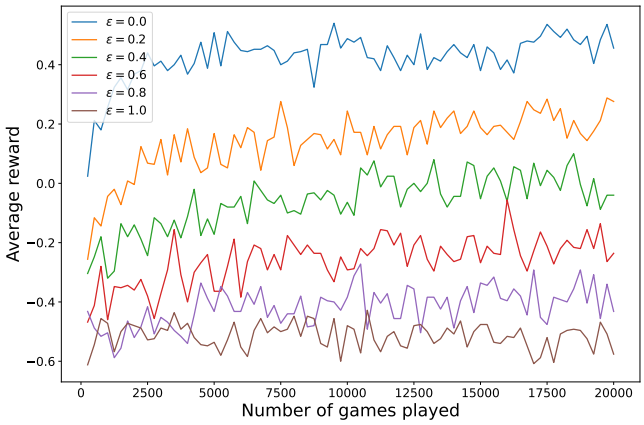

Fig. 1: Average reward for every 250 games during training with different value of fixed exploration rate ϵ. The average reward increases over time for most of the agents. With less exploration rate ϵ., the agent learns how to play Tic-Tac-Toe more efficiently and the reward is higher.

B. Decreasing exploration

Question 2. As illustrated in Figure. 2, we try with different values of n^*. Considering the agent decreases exploration only when $n \leq n^*$, we know that agent with $n^* = 1$ is equivalent to agent with fixed value of ϵ. This agent converges most quickly among all the agents with different n^*, while the best of average reward it can reaches is not significantly different from those of other agents. Especially, the reward of other agents converge slowly when they are decreasing ϵ, namely when $n \leq n^*$, which indicates that decreasing ϵ actually makes the training less efficiently and doesn't help.

With larger value of n^*, the agent spend more times in the mode of decreasing exploration, during which the agent learns less efficiently, and therefore takes more number of games to reach the max average reward. From Figure. 1, we know less exploration rate ϵ results in better learning trend. This phenomenon explains why decreasing exploration doesn't help in this case, as agent with larger n^* spends more time on larger value of ϵ when the learning is less efficient.

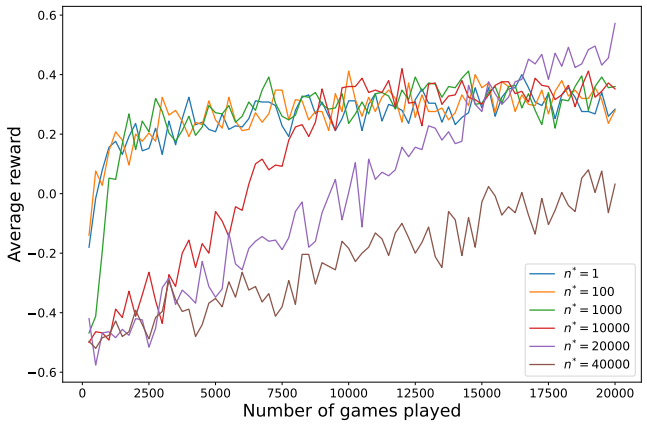

Fig. 2: Average reward for every 250 games during training with decreasing exploration.

Question 3. Illustrated in Figure. 3

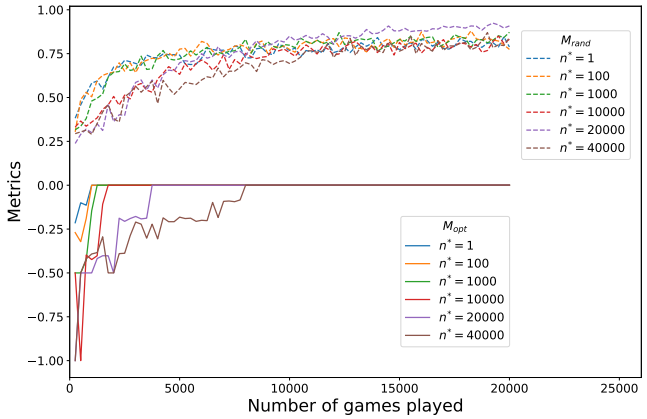

Fig. 3: M_{opt} and M_{rand} for every 250 games during training with decreasing exploration and different values of n^*. Similar to Figure. 2, agents with different n^* reaches almost the same maximal of M_{opt} and M_{rand}, while those with smaller n^* converges more quickly. Surprisingly, agent with $n^* = 40000$ can reaches the same maximal M_{rand} as others while that's not the case for average reward. In addition, M_{opt} of all agents can reach 0, which is the best value one can get. Considering the metrics, different values of n^* all train good agents. The only difference is learning efficiency.

C. Good experts and bad experts

Question 4. The result is showed in Figure. 4 As we see in the previous question, 20000 games are sufficient for agent with $n^* = 10000$ to converge. Considering giving more space for exploration, we decide to take $n^* = 10000$ as the best value. We also tried with $n^* = 1000$ for Question4. But the result doesn't show any difference. With $n^* = 10000$, effect of different values of ϵ_{opt} on the metrics can be clearly observed.

For M_{opt}, agent trained with better experts, i.e. smaller ϵ_{opt}, results in both quicker convergence and larger maximal value. This is reasonable as with smaller ϵ_{opt}, the agent will be more likely to lose if it makes some mistake or takes an aggressive move. Therefore, it will be trained to be good at defense. In the contrast, agent trained with worse experts has better M_{rand} value. As a side effect of training with better expert, the agent will tend to be conservative. It will take less aggressive move as it's more likely for them to be punished by good expert. However, when computing M_{rand}, since the random agent is not that smart, aggressive move can result in larger chance to win.

Another intuitive explanation is that agent trained with better expert is easier to gain higher metric value when computing M_{opt} since they are trained with agent more like to the optimal agent. Same apply for the phenomenon of M_{rand}.

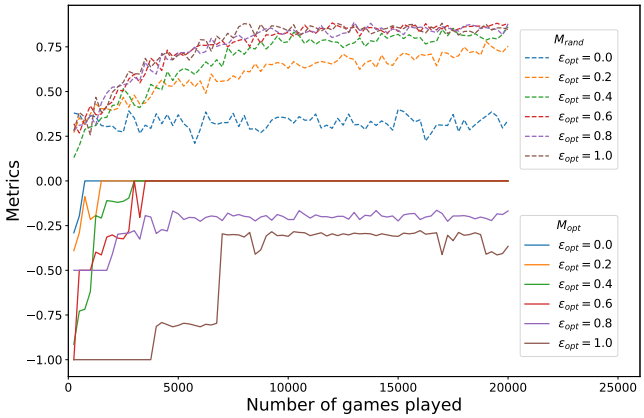

Fig. 4: M_{opt} and M_{rand} for every 250 games during training with $n^* = 10000$ and against optimal player with different ϵ_{opt}.

Question 5.

The highest values of M_{rand} the agent could achieve is 0.882 when playing against expert with $\epsilon_{opt} = 0.6$. Except playing against expert with $\epsilon_{opt} = 0.8$ and expert with $\epsilon_{opt} = 1.0$, all other agents can achieve $M_{opt} = 0$.

Question 6.

$Q_1(s, a)$ and $Q_2(s, a)$ should have different values. When playing against Opt(0), the agent is trained with a master who can never lose. Therefore, it never knows how to win and what it learns is only how to defend. However, agent trained against Opt(1) experiences both lose and win. Moreover, since Opt(0) is deterministic, some state could never happen when playing against it.

For example, when encounter the state showed in Table I, the Q-value in Q_1 is all zero. That's because such state is unlikely to happen as Opt(0) will always first to take the center. In that case, agent trained with Opt(0) doesn't know it's already close to win. In contrast, Q_2 of this state with action (2, 2) is 0.99, far greater than the rest and it knows how to win. More examples are in jupyter-notebook.

-	-	X
O	-	X
-	O	-

TABLE I: state 1

D. Learning by self-practice

Question 7. Illustrated in Figure. 5

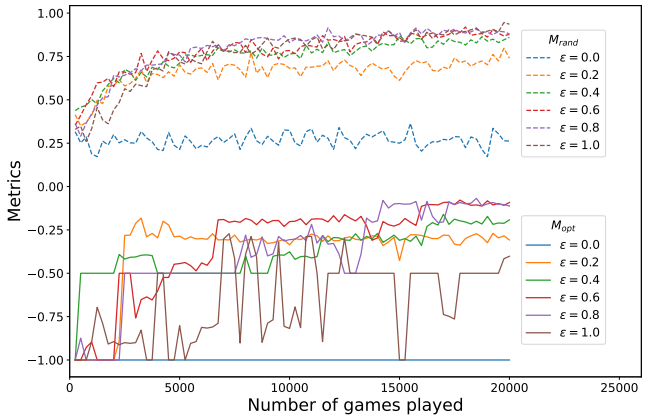

Fig. 5: M_{opt} and M_{rand} for every 250 games during learning by self-practice with different values of ϵ. Except the agent with $\epsilon = 0.0$, M_{rand} of all other agents continuously increases with the number of games. For M_{opt}, despite lots of vibrations exist, the overall trend is that the value increases with the game number. The agents with $\epsilon = 0.0$ may be trapped to a sudden pattern since it has no chance to explore. It can be clearly observed that with larger value of ϵ, M_{rand} performs better. However, the relationship between ϵ and M_{opt} is not so clear.

Question 8. Illustrated in Figure. 6

Question 9.

The highest values of M_{rand} the agent could achieve is 0.81 when $n^* = 40000$. The highest values of M_{opt} the agent could achieve is 0.0 when $n^* = 10000$.

Question 10. Figure. 7 shows the Q-values for three selected different states of an agent trained by self-practice with decreasing exploration and $n^* = 20000$.

The first chosen state is the initial state. For this state, the Q-value prefers (0, 1) instead of (1, 1), which is the optimal choice. The center point has only third greatest Q-value.

The second state is the case that the agent is first to play ("X"), while both the agent and its opponent requires only one point to win the game. Instead of choosing (1, 2) to defend the opponent, the agent has highest Q-value on (0, 2), which can directly win the game. This Q-value makes sense as it's indeed the optimal value.

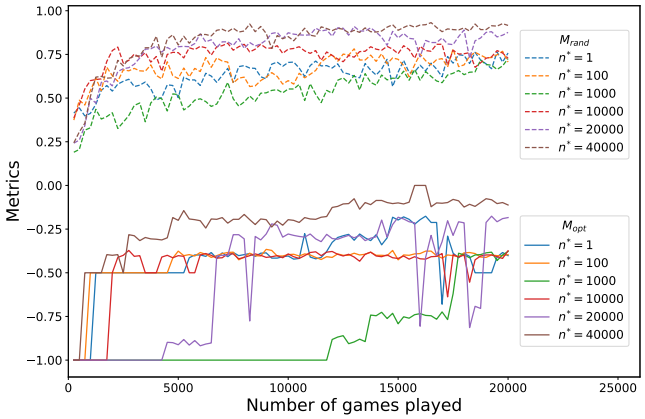

Fig. 6: M_{opt} and M_{rand} for every 250 games during learning by self-practice with decreasing exploration (different values of n^*). Compared to the previous plot, it can be clearly observed that decreasing exploration improves the training. With large value of n^*, M_{rand} quickly reaches the level of 0.75 while M_{opt} can even reach the value of 0.0, which can never be reached with fixed ϵ. As we know from Figure. 5, larger exploration rate results in better metric. It is reasonable to see larger value of n^* generates greater M_{opt} and M_{rand}.

The third state is when the agent is the second to play ("O"), and his opponent is about to win. Under this situation, the highest Q-value lays on (2, 0), which makes sense as it needs to block this position to prevent a direct lose. Meanwhile, the Q-values of all the available action are negative since even it blocks its opponent, the chance to win is still very small.

Generally, the agent learned well.

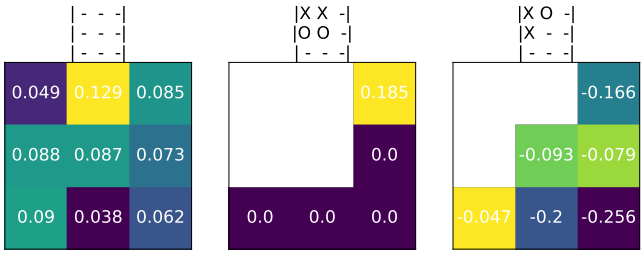

Fig. 7: Heat maps of Q-values for three different states.

II. Deep Q-Learning

A. Learning from experts

Question 11. Illustrated in Figure. 8

Question 12. Illustrated in Figure. 9

Question 13. As showed in Figure. 10, we try with different values of n^*. Compared to Q-Learning, the trend of metrics of DQN agents trained with decreasing exploration doesn't show much difference compared to that of agent trained with a fixed ϵ, i.e. when $n^* = 1$. The evolution of M_{opt} seems vibrate a lot during the training. For fixed ϵ, large vibration never happens after 5000 games, while those with decreasing exploration can still have vibration after 10000 games training.

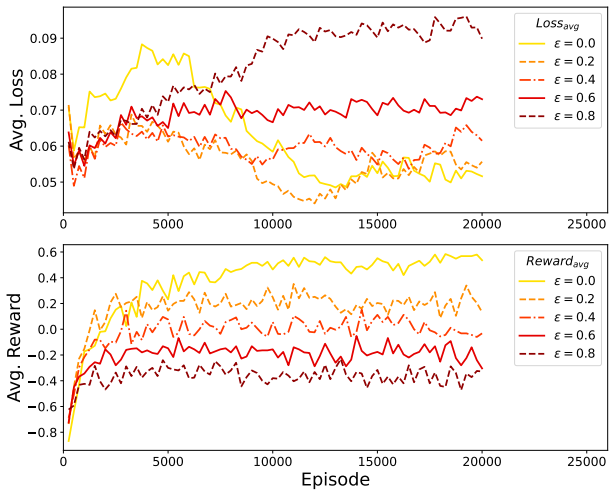

Fig. 8: Average reward and training loss during training with different value of fixed exploration rate ϵ. The average reward increases over time for all ϵ. In contrast, most average loss curves fluctuate and there is a clear downward trend only when $\epsilon = 0.2$. According to the reward evolution, we can see the agent performs better when training.

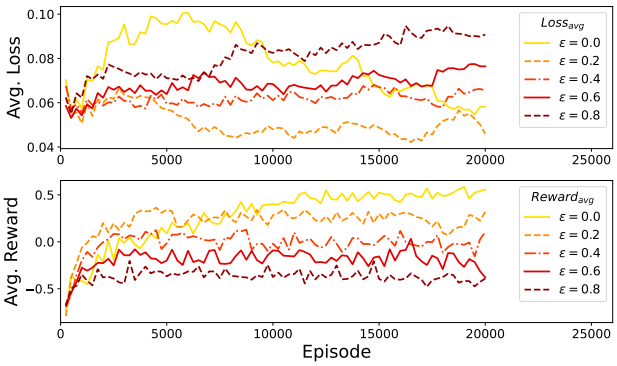

Fig. 9: Average reward and training loss during training without the reply buffer and batch size is 1. The average reward for all ϵ increases over time. In contrast, most average loss curves are more volatile compared to the case of using replay buffer. The agent with smaller ϵ performs better.

However, at least M_{opt} of most of them move backs to 0. In DQN, it seems there are no clear improvement brought by decreasing ϵ.

As mentioned previously, with larger value of n^*, the M_{opt} tends to keep vibrating after 10000 games training, while large vibration stops early for those with small value of n^*. However, the effect of n^* on the trend of M_{rand} is negligible.

Question 14. Figure. 11 shows the metrics of the agent trained with optimal player of different ϵ_{opt}.

Compared to other values, $n^* = 10000$ is chosen because the return on agent and the training loss will show a more stable upward and downward trend, respectively. In contrast, choosing a smaller value may lead to larger oscillations while

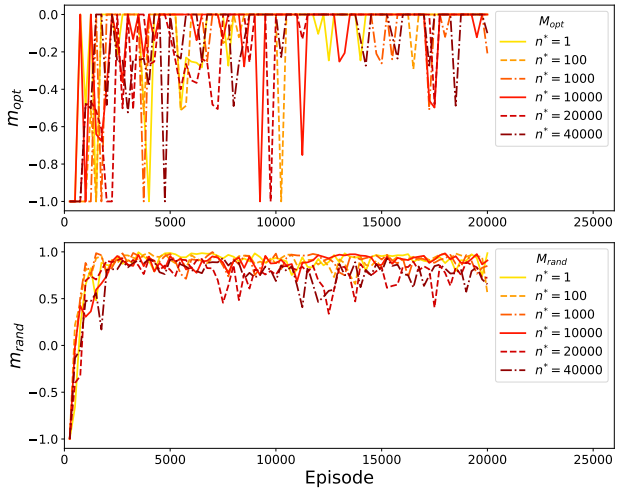

Fig. 10: M_{opt} and M_{rand} for every 250 games during training with decreasing exploration and different values of n^*.

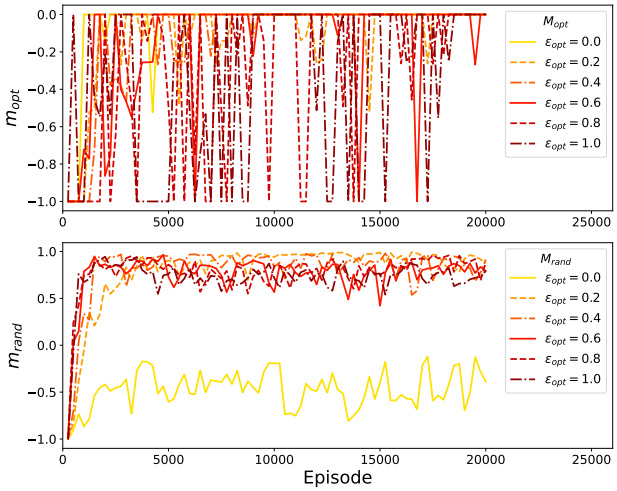

Fig. 11: M_{opt} and M_{rand} for every 250 games during training with $n^* = 10000$ and against optimal player with different ϵ_{opt}.

the agent does not achieve satisfactory rewards and metrics if n^* exceeds 20,000.

For M_{opt}, agents trained with better experts, i.e. smaller ϵ_{opt}, tend to a much stabler curve. However, when trained with worse experts, the evolution of M_{opt} vibrating more during the training. For example, the curve for $\epsilon_{opt} = 0.8$ keeps vibrating all along the whole training time. Moreover, M_{opt} changes frequently from -1 to 0 when trained with $\epsilon_{opt} = 1.0$. This makes sense as with smaller ϵ_{opt}, the test agent for calculating M_{opt} is more like the opponent the agent are trained with. Therefore, it's reasonable for agent trained with smaller ϵ_{opt} performs better in M_{opt}

Except the agent trained with $\epsilon_{opt} = 0.0$, all other agents seem perform quite well on M_{rand}. The reason is same as that in Q-Learning. Since the agent trained with $\epsilon_{opt} = 0.0$ never knows how to win, it always wants to defend. Moreover, it could encounter some states they never see in the training as its master won't have random move. But when testing with a random player, it's possible to see these states and it doesn't know where to move, which results a random move. All these explain its low M_{rand} value. In addition, one can observe with larger ϵ_{opt}, M_{rand} converges to 1.0 more quickly.

Question 15.

The highest values of M_{rand} the agent could achieve is 0.914 when playing against expert with $\epsilon_{opt} = 0.4$. All other agents can achieve $M_{opt} = 0$ after 20000 games.

B. Learning by self-practice

Question 16. Illustrated in Figure. 12

Question 17. Illustrated in Figure. 13

Question 18.

The highest values of M_{rand} the agent could achieve is 0.886 when $n^* = 1000$. The highest values of M_{opt} the agent could achieve is 0.0 when $n^* = [100, 1000, 10000, 40000]$.

Question 19.

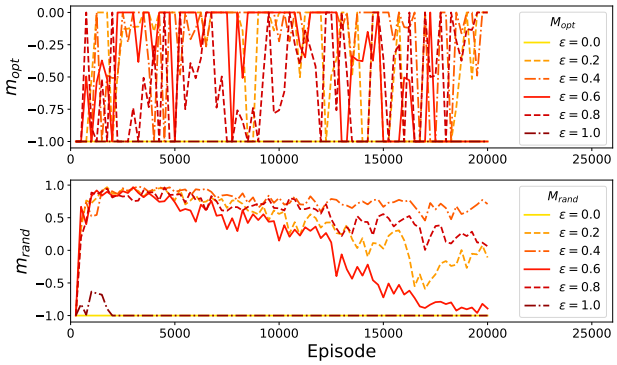

Fig. 12: M_{opt} and M_{rand} for every 250 games during learning by self-practice with different values of ϵ. When setting ϵ to be 0.0 or 1.0, both two metrics never improve during training. In the case that $\epsilon = 0.2$, M_{rand} converges quickly to around 1.0 and keep stable while M_{opt} can converge to 0.0 and have relatively few vibration. For other choice of ϵ, curves of M_{opt} are unstable and those of M_{rand} converges to 1.0 quickly, but the value goes down if keep training. The agent is able to learn by self-practice if the value of ϵ is carefully chosen.

The chosen states follow the choice in Question 10. Figure. 14 shows the Q-values for three selected different states of a DQN agent trained by self-practice with decreasing exploration and $n^* = 10000$. According to Figure. 13, it could achieve best M_{rand} among other agents.

The first chosen state is the initial state. For this state, the Q-value prefers the optimal choice (1, 1) with values equals 1.205, which is ahead of other options by a large margin.

The second state is the case that the agent is first to play ("X"), while both the agent and its opponent requires only one point to win the game. Instead of choosing (1, 2) with values equals 0.47 to defend the opponent, the agent has highest Q-

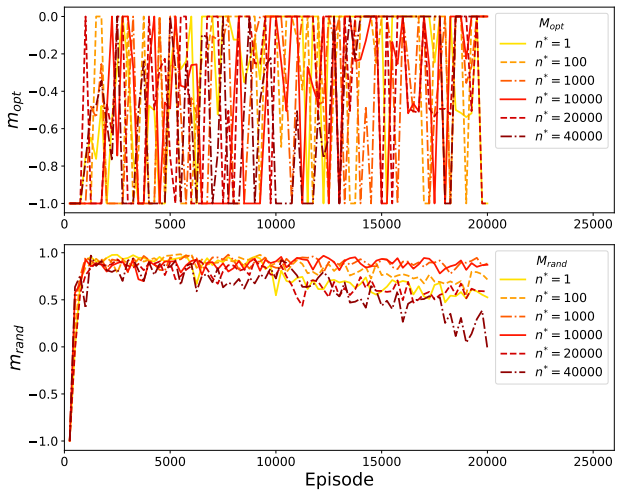

Fig. 13: M_{opt} and M_{rand} for every 250 games during learning by self-practice with decreasing exploration (different values of n^*). Decreasing the ϵ could help increase M_{rand} rapidly and keep stable throughout the training process than having a fixed value. Compared to n^* over 10000, a smaller n^* is more favorable to guarantee the stability of M_{rand} in the case of agent self-practice. In contrast, different choices of n^* have less effect on M_{opt}.

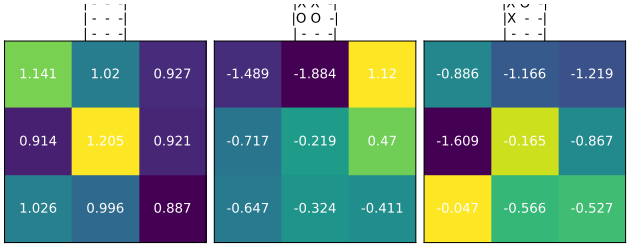

Fig. 14: Heat maps of Q-values for three different states predicted by DQN.

value on (0, 2) with values equals 0.47, which can directly win the game. In addition, The well-trained agents tend not to select the positions that have been placed.

The third state is when the agent is the second to play ("O"), and his opponent is about to win. Under this situation, the highest Q-value lays on (2, 0), which makes sense as it needs to block this position to prevent a direct lose.

III. Comparing Q-Learning with Deep Q-Learning

Question 20. For a fair comparison, when training with experts, we activate decreasing exploration and set $n^* = 10000$, $\epsilon_{opt} = 0.5$ for both Q-Learning and DQN. In addition, decreasing exploration is activated for self-practice of the two algorithms and n^* is set to be 10000. For all training, the metrics are tested for every 250 games.

We present the result in Table. II where $T_{train}(M_{opt})$ is the minimum number of games required to reach $0.8 * (1 + \max(M_{opt})) - 1$, $T_{train}(M_{rand})$ is the minimum number of games required to reach $0.8 * \max(M_{rand})$, $\max(M_{opt})$ and $\max(M_{rand})$ are the best metrics the agent can get during the training with settings stated in last paragraph. That's why the values here are not the same as those in Question 5, 9, 15, 18.

	$T_{train}(M_{opt})$	$T_{train}(M_{rand})$	$\max(M_{opt})$	$\max(M_{rand})$
Q-Learning with $Opt(0.5)$	2500	6500	0.0	0.882
Q-Learning (self-practice)	7000	1750	-0.262	0.834
DQN with $Opt(0.5)$	3500	2000	0.0	0.924
DQN (self-practice)	4000	1250	-0.084	0.844

TABLE II: Comparison between Q-Learning and DQN. Unit of T_{train} is the number of training games.

Question 21. In this project, the famous game of Tic Tac Toe is used as environment to train artificial agents with both Q-learning and Deep Q-learning strategies. Under the 3x3 grid setup, the space of possible board configurations or states the environments is discrete with $3^9 = 19683$ possible states. With Q learning strategy, the numerical representations of state-action pairs are approximated from trials with tabular fashion.

In contrast to learn to fill Q table, DQN is trained to update the parameters of policy network which could have more complex decisions from extensive game plays. Moreover, DQN agent is trained without the limitation of illegal moves so that it could see more illegal moves and corresponding transitions for experience replay.

Both learning from experts and self-practice are evaluated with the two strategies. Because self-practice may lead to non-optimal decision making as the model learns how to beat itself rather than beating the optimal player, we observed that the highest performance of both methods using self-practice will obtain worse metrics after the same training episodes. Given the curve of M_{opt} and M_{rand}, we can find no matter the agents are trained by experts or self-practice, the evolution of the metrics for DQN contains a lot of vibrations, while that for Q-Learning is more stable. In the end, we do a fair comparison between Q-Learning and DQN for training with both experts and self-practice. As listed in Table. II, under same settings, DQN generally performs better than Q-Learning regarding to convergence speed and highest metric value.

In conclusion, DQN can learn faster and better but easy to undergo more noisy fluctuations during the training while Q-learning shows slower convergence curve but more stable trend.