Human-assisted reinforcement learning demonstrated on the Flappy Bird Game

Jana Ristovska
89191025@student.upr.si
Faculty of Mathematics, Natural
Sciences and Information Technologies
University of Primorska
Glagoljaška ulica 8
SI-6000 Koper, Slovenia

Domen Šoberl
domen.soberl@famnit.upr.si
Faculty of Mathematics, Natural
Sciences and Information Technologies
University of Primorska
Glagoljaška ulica 8
SI-6000 Koper, Slovenia

ABSTRACT

In model-free reinforcement learning, the agent usually starts learning by blind exploration, which can take a significant amount of time before starting to experience positive reinforcement that drives further progress. In this paper, we address the following question: Can the learning time be reduced if the agent first observes successful behaviors demonstrated by humans, before commencing its independent learning? We propose an adaptation of the traditional Q-learning algorithm, so that it can gradually integrate recorded demonstrations into the learning process, and demonstrate this method on the well-known Flappy Bird game. We recorded 1496 gameplays of Flappy Bird played by 22 volunteers, and selected 941 successful recordings to assist the Q-learning algorithm. The results show that such human assistance speeds up the learning process in a logarithmic manner, which means that the biggest gain is made in the initial stages of learning and becomes almost negligible later on. We show experimentally that in the initial stages of learning, human demonstrations contain more useful information than the agent can acquire independently.

KEYWORDS

Reinforcement learning, Q-learning, Learning by demonstration, Imitation learning

1 INTRODUCTION

Using reinforcement learning [9], computers can learn, by trial and error, how to play simple video games. The learning process typically takes several hours, after which the computer can reach, or even surpass the human level of playing [6]. However, an interesting question arises: Can the learning time be reduced if the computer is given the opportunity to observe a number of successfully played games by humans, before commencing its independent learning? Incorporating player gameplay data has already been considered as a potential way to enhance the learning process in the domain of video games. In this paper, we aim to research possible improvements in the training time of the traditional Q-learning algorithm when integrating into the learning process a successfully played game by humans. We demonstrate our approach on the well-known Flappy Bird Game. The Flappy Bird Game serves as a suitable platform to investigate this research question due to its simplicity and well-defined gameplay mechanics. We study if and how the training improves with the quality of the recorded playthroughs i.e. the number of scores reached by the human. Throughout the paper, we will use the term *human-assisted reinforcement learning* to label any approach in the field of machine learning, specifically reinforcement learning, where human demonstrations are incorporated into the learning process to aid the reinforcement learning agent's decision-making, although similar ideas have been tried under various names.

2 RELATED WORK

Reinforcement learning has garnered significant interest in the domain of playing video games. Traditional approaches like Q-learning and Deep Q-Networks (DQNs) have demonstrated the ability to train agents to achieve superhuman performance in various games. More recent advancements, such as Proximal Policy Optimization (PPO) [8] and Asynchronous Advantage Actor-Critic (A3C) [2], have shown improved stability and sample efficiency. These algorithms have been employed to train agents in a range of games, including Atari 2600 games, Gymnasium environments, and custom game simulations [6, 9].

The concept of human-assisted reinforcement learning, where agents learn from demonstrations provided by humans, has gained traction in recent research. One approach in this research area is called "Imitation Learning" or "Behavioral Cloning", where the agent tries to mimic the expert's behavior directly [11]. However, this approach can suffer from issues like compounding errors, resulting in sub-optimal performance.

Another prominent approach is "Inverse Reinforcement Learning" (IRL), where the agent attempts to infer the underlying reward function from expert demonstrations [1]. This allows the agent to learn the task's structure without requiring exact expert actions. However, Inverse Reinforcement Learning can be challenging due to the need to model the reward function accurately.

Researchers have applied human-assisted reinforcement learning to various tasks, such as robotic manipulation [12], autonomous driving [4, 13], and game playing [8]. Studies have shown that leveraging human demonstrations can lead to faster convergence and better performance compared to training from scratch [3, 8]. In the context of video games, human demonstrations have been used to teach agents to play games like Super Mario Bros, Dota 2, and Montezuma's Revenge, showing promising results in reducing learning time and improving performance [5]. Similar research employing learning from demonstrations in combination with transfer learning has been showcased in a soccer simulator called Keepaway [10], investigating the effects of both the quality of recorded

DOI: https://doi.org/10.26493/scores23.13

demonstrations and the integration of demonstrations from multiple instructors.

3 IMPLEMENTATION

As the training environment, we adapted an existing Flappy Bird implementation, which was written in Python and released under the MIT license [7]. This open-source implementation offers a simplified version of the classic Flappy Bird game, to which we added a state discretization feature and a reward function. The state is composed of the horizontal distance from the bird to the right side of the closest lower pipe and the vertical distance from the bird to the top side of the closest lower pipe. We discretize the state by splitting the horizontal distance into 10 and the vertical distance into 25 equidistant intervals. This way we obtain 250 discrete states, which we uniquely map into their integer representations $\{0, \ldots, 249\}$. The rewarding system returns the negative reward of -1000 when the bird dies (which also terminates the episode), and the positive reward of +10 points at every step when the bird stays alive. This reward system was influenced by similar research in the field.

We implemented two types of Q-learning algorithms: the traditional Q-learning algorithm with discrete states and actions to serve as the performance baseline, and our proposed human-assisted Q-learning algorithm that can integrate recorded gameplay data into the learning process.

Figure 1 shows the data flow with the traditional (baseline) Q-learning approach. Based on the current state, as received from the environment, the agent selects and executes the most prominent action according to the current Q-table policy. When the action is executed, a new state (state') is received from the environment, together with the obtained reward. The previous state, the executed action, the reward, and the new state are then used to update the Q-table using the Bellman equation:

$$Q(s,a) \leftarrow Q(s,a) \quad \alpha \cdot \left(r \quad \gamma \cdot \max_{a'} Q(s',a') \right), \tag{1}$$

where s is the current state, a the executed action in state s, r the received reward, s' the next state, a' a future action, α the learning rate and γ the discount factor. We set the learning rate α to 0.45, and the discount factor γ to 0.9. We selected these values through meticulous fine-tuning, aiming to achieve the best training results we could.

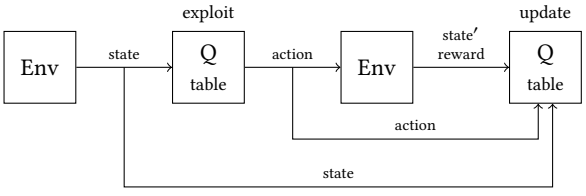

Figure 1: The learning process of the baseline Q-learning algorithm.

The human-assisted Q-learning algorithm that we propose in this paper works as shown in Figure 2. We use two separate Q-tables to implement the policy that is used by the intelligent agent during the Q-learning process: the (trainable) Q-table Q, which is the same as the Q-table used by the traditional Q-learning algorithm, and the user Q-table Q_U, which is populated with the recorded human gameplays.

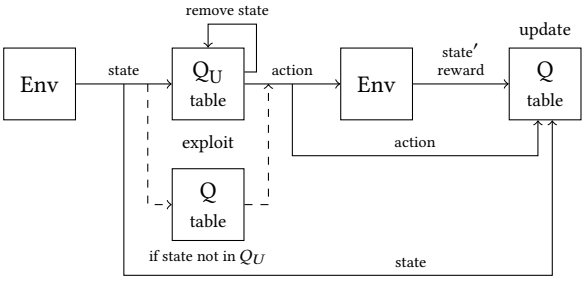

Figure 2: The learning process of our human-assisted Q-learning algorithm.

The process of populating the Q_U table is as follows: For every step of the recorded game, we discretize the game state as described above, read the action that the user made in that state, the obtained reward and the consequent state, which we also discretize. We then compute the $Q(s,a)$ value similarly as in the online training using Equation 1, however, there is no interaction with the environment in the process. The computed $Q(s,a)$ value is inserted into the Q_U table at the $[s,a]$ position. Because the computation relies on the existing Q_U entries for s' and a', the process is repeated two times for the same recording. Our experiments showed that doing more than two passes does not improve the training performance.

During the online training, the Q_U table is utilized as follows: If an entry for the current state s exists in the Q_U table, the action is decided based on this entry, after which the entry for state s is removed from Q_U. If there is no entry for s in Q_U, table Q is used as with the traditional Q-learning algorithm. We may say that the knowledge from the recorded human gameplay is gradually being integrated into the learned policy. When the executed action is based on the values from the user Q-table Q_U, the agent has followed the human's strategy. Presuming that the human player played a feasible strategy, such a choice should indicate a certain advantage over blind exploration.

4 OBTAINING HUMAN GAMEPLAY DATA

In order to obtain a pool of recorded gameplays, we got 22 volunteers to play the game of Flappy Bird. They were instructed to play the game as many times as they want and to aim to score as many points as they can. They played 1496 games in total, 555 of which scored 0. Since our methodology is based on providing successful human demonstrations, we only used the recordings with at least 1 point reached, which was 941 gameplays. The maximum score reached was 53 and the second best was 33. The distribution of all the recorded gameplays is shown in Figure 3.

A gameplay recording is composed of a tuple (h, v, a, r, s) for each consecutive game step. Variables h and v represent the horizontal and the vertical distances in pixels from the next pipe. Values a, r and s respectively represent the executed action, the obtained reward and the current score. The resulting state can be obtained from the subsequent tuple in the recording.

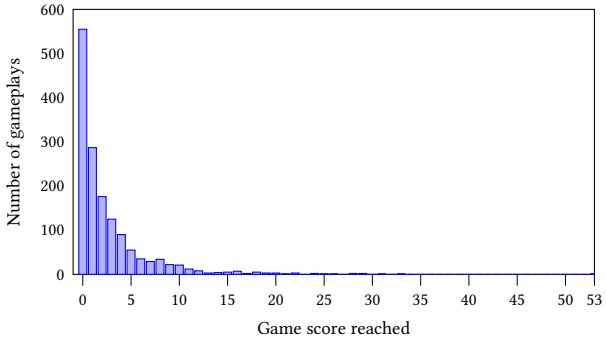

Figure 3: Distribution of the recorded human gameplays according to the reached game score.

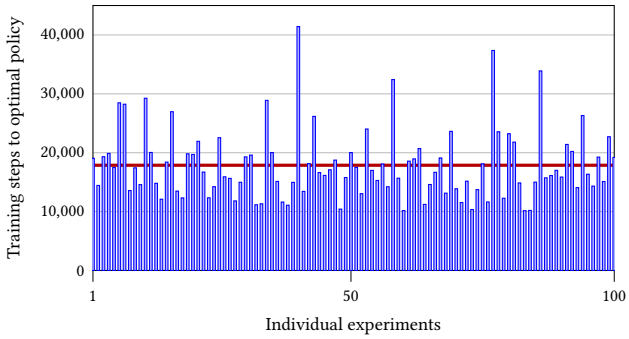

Figure 4: The results of 100 experiments using the traditional Q-learning algorithm.

5 EXPERIMENTAL RESULTS

In reinforcement learning, the usual approach to measuring the performance is to count the number of episodes over which the learning curve converges. In our case, an episode ends only with the bird's death, so the episodes vary in length considerably. Moreover, after the agent learns how to play optimally, an infinitely long episode follows until terminated manually. Therefore, as a measure of performance, we count the number of training steps needed for the agent to reach a specific score, after which the training is concluded. The initial experiments showed that the agent always learned an optimal policy before reaching 60 points, therefore we set the training goal to reaching 100 points.

We conducted a single experiment as follows. We ran the learning algorithm and let it play for as many training episodes as needed, until the bird scored 100 points. Afterward, the training was stopped and we recorded the total number of training steps taken during all the episodes together.

5.1 Traditional Q-learning

We conducted 100 independent experiments using our implementation of the traditional Q-learning algorithm, which we took as the baseline method. The average number of training steps to learn an optimal policy was 17872.18, with the standard deviation of 5912.22. The maximum number of training steps was 41426 and the minimum was 10158. To put this into perspective, with the frequency of 60 frames (actions) per second, the average training time took 296.4 seconds, which is 4 minutes and 56.4 seconds. Individual results are shown in Figure 4, where the red line denotes the average number of training steps. Different outcomes are due to the fact that the pipes are generated randomly by the environment.

5.2 Human-assisted Q-learning

When assisting the learning algorithm with a single recorded human gameplay, the duration of that gameplay is of crucial importance. Not only more data is provided by a longer gameplay, the playing skills and therefore the employed strategy might also be better, since the player managed to 'survive' longer. In our case, the duration of the game is strongly correlated with the final score achieved in that game, since the speed of the bird is constant and the pipes are spaced equally. We therefore conducted the experiments as follows: We grouped the recorded gameplays according to their duration — gameplays with the total score of 1 were put into the first group, gameplays with the total score of 2 into the second group, etc. This way we formed 30 groups of recorded gameplays with the same distribution as shown in Figure 3 (note that for some scores in the range of 1 to 53, no gameplay was recorded). We then ran the learning algorithm independently 100 times for each group, each time selecting randomly a gameplay from the current group to assist in training. We measured how much the training time decreased on average for each group in relation to the average training time of the baseline learning algorithm. We were interested in whether and in which cases the decrease in the training time would be greater than the duration of the assisted human gameplay.

The left plot in Figure 5 shows the average decrease in training time for all 30 gameplay groups, which are denoted with red dots. Gameplay durations are averages for each group, although variations within a group are negligible (up to the amount of time needed to travel between two adjacent pipes). The improvement increases with the duration of the used gameplay recording in a near-logarithmic manner. We can notice that the improvement rate slows down significantly when the duration of the gameplay exceeds 30 seconds. This indicates that the majority of the human skill is encoded into the first 30 seconds of gameplay, after which the same strategy is very likely repeated, with less probability of novelty in the recorded data.

The right plot in Figure 5 shows the same data from a different perspective. Instead of the absolute reduction in the training time, the difference between the reduced time and the average duration of the used recordings is shown along the y-axis. A positive value indicates that the training time was reduced more than the duration of the recording. This means that there was more useful information in the recorded gameplay than could be obtained by independent training in the same amount of time. Shorter recordings show a higher ratio, the highest being with the first group of recordings, where the human player managed to pass only one obstacle and lasted for about 3 seconds. The ratio is dropping with longer games and reaches 0 at about 15 seconds mark.

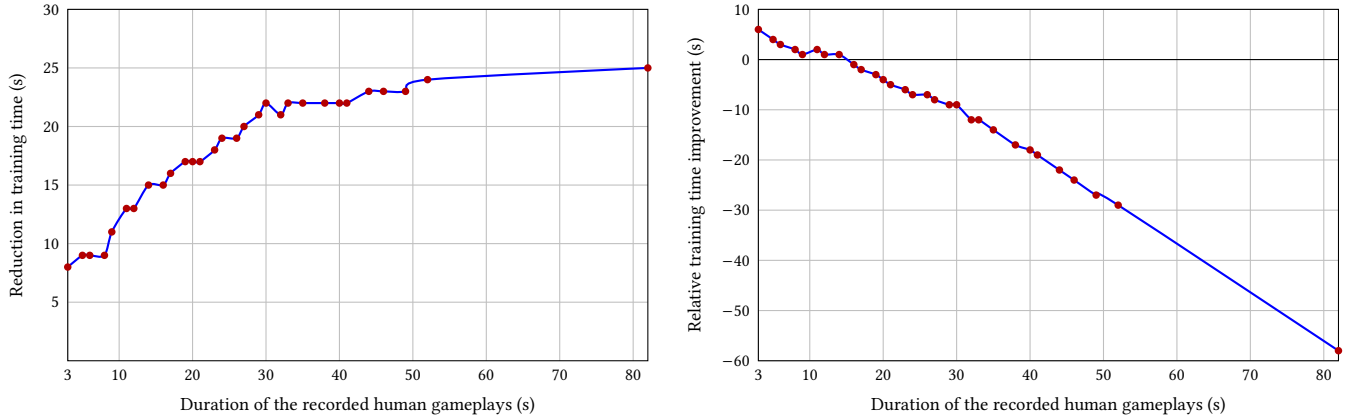

Figure 5: Improvement in the training time when using human gameplay recordings of different durations.

6 CONCLUSION

The results of our study demonstrate the potential of human-assisted reinforcement learning in improving the performance of training an agent. The main premise of this paper is that a human demonstration of a successful playing strategy contains more useful information than an agent can in the same amount of time obtain independently. We verified this idea on the well-known Flappy Bird game by first implementing a traditional Q-learning algorithm, which was able to learn an optimal strategy in about 5 minutes on average. We then proposed an adaptation to the traditional Q-learning algorithm that can integrate gameplay recordings into the learning process. We obtained 941 useful gameplay recordings from 22 volunteers that we used with our adapted learning algorithm and compared the reduction of the training time in comparison to the traditional Q-learning approach.

The results confirmed our hypothesis and showed that the most useful information regarding a successful Flappy Bird playing strategy is encoded in the first 30 seconds of human gameplay, after which the improvement still somewhat increases, but at an insignificant rate. We may conclude that in those first 30 seconds, a human player encounters most of the possible situations, after which it mostly repeats already seen playing maneuvers.

Such absolute training time improvement alone may suffice in cases where offline data is abundant and online training is expensive. However, to assess whether a human demonstration actually contains more information than blind exploration, we analyzed relative improvement in the training time, which showed that on average, the first 15 seconds of human demonstration provided more useful information than could independently be obtained by the agent.

The main limitation of our research is the simplicity of the chosen training environment. After discretization, we obtained a state space spanning 250 possible states, which can fairly quickly be explored by the traditional Q-learning algorithm, as we have also demonstrated. This leaves little room for improvement through human assistance. We believe that in more complex domains, where it takes an agent hours or even days to train, human assistance could prove much more beneficial. Moreover so when the reward is sparse and the agent must rely on pure luck to obtain the reward for the first time. Human demonstrations of obtaining a reward could significantly speed up this initial part of the training.

Our approach is currently limited to discrete states and actions, which is the limitation of the traditional Q-learning algorithm with tabulated Q-values. As a part of our future work, we aim to move towards continuous environments and explore the possibilities to integrate human assistance into modern deep Q-learning architectures. This way we may be able to tackle more complex environments and even real-world control problems.

REFERENCES

[1] Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship Learning by Inverse Reinforcement Learning. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML-2004)* (Banff, Alberta, Canada). ACM, New York, NY, USA, 1–8.

[2] M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz. 2016. Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU.

[3] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. 2018. Deep reinforcement learning from human preferences. arXiv:1706.03741 [stat.ML]

[4] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy. 2018. End-to-end Driving via Conditional Imitation Learning.

[5] J. Leike, M. Martic, and S. Legg. 2017. Learning through human feedback. DeepMind Blog. https://www.deepmind.com/blog/learning-through-human-feedback

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning.

[7] Russs123. 2020. Flappy Bird. https://github.com/russs123/flappy_bird.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG]

[9] R. S. Sutton and A. G. Barto. 2018. *Reinforcement learning: An introduction* (2 ed.). The MIT Press, Cambridge, MA, USA.

[10] M. E. Taylor, H. B. Suay, and S. Chernova. 2011. Integrating reinforcement learning with human demonstrations of varying ability. In *10th International Conference on Autonomous Agents and Multiagent Systems 2011, AAMAS 2011*, Vol. 1. FAAMAS, Richland, SC, 617–624.

[11] F. Torabi, G. Warnell, and P. Stone. 2019. Recent Advances in Imitation Learning from Observation. arXiv:1905.13566 [cs.RO]

[12] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller. 2017. Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards. arXiv:1707.08817 [cs.RO]

[13] J. Wu, Z. Huang, C. Huang, Z. Hu, P. Hang, Y. Xing, and C. Lv. 2021. Human-in-the-Loop Deep Reinforcement Learning with Application to Autonomous Driving. arXiv:2104.07246 [cs.RO]