دانشگاه گیلان

University of Guilan

English Sumary of Final Year Project Thesis

GridSoccer: An experimental study to use deep reinforcement learning in video games

Pouya Pournasir

supervised by
Prof. Seyed Abolghasem Mirroshandel

1 Introduction

Machine learning is a subset of Artificial Intelligence, and it refers to algorithms that discover patterns in data and use them to accomplish specific tasks.(Francois-Lavet et al., 2018) These algorithms provide automated methods for programs to learn and improve their performance without human supervision. (Mitchell and Hill, 1997) Function approximation is a fundamental part of solving machine learning tasks. There are different types of function approximators, such as deep learning(LeCun et al., 2015), which mimics the human brain to process data.

Reinforcement Learning is one of the three paradigms of machine learning tasks, alongside supervised learning and unsupervised learning. An agent, in each step, receives observations in an environment and decides on an action. Based on what decision the agent has taken, it gets a reward, which can be either positive or negative. The agent tries to change its behavior to learn a good one. (Francois-Lavet et al., 2018)(Brockman et al., 2016)

Due to the drastic improvements of deep learning in identifying features from high-dimensional data (e.g., image, text, audio), using its architecture in the field of Reinforcement Learning has provided solutions for previously unsolvable problems, referred to as *Deep Reinforcement Learning*. (Arulkumaran et al., 2017) However, it is difficult to compare deep learning algorithms due to their stochastic nature, and it gets worse when utilizing them in Reinforcement Learning because of the involvement of random variables in the environment(Francois-Lavet et al., 2018). As a result, finding a way for accurate evaluation is invaluable.

A reasonable comparison should challenge algorithms in generality and in an environment similar to its usage in the real world without the *experimenter's bias* (Bellemare et al., 2013). Arcade Learning Environment was among the first platforms which provided emulated Atari 2600 video games as an environment for agents to learn. (Bellemare et al., 2013) Using video games as a benchmark for researchers in this field gained popularity afterward; platforms such as VizDoom (Kempka et al., 2016), DeepMind (Beattie et al., 2016), and Project Malmo (Johnson et al., 2016) have emerged.

These benchmarks, though valuable in particular aspects, do not give a good representation of the real world(Bellemare et al., 2013)(Kempka et al., 2016). For example, most platforms mentioned in this study are from games such as Doom, Minecraft, and Quake III Arena, which do not have the general complexity of the real world. Atari games, for instance, are 2D games that hardly resemble agents' environments in practice.(Juliani1 et al., 2019) Furthermore, due to the ongoing progress in this field, benchmarks such as the ALE have become less informative because of the agent's super-human performance in most environments(Machado et al., 2017).

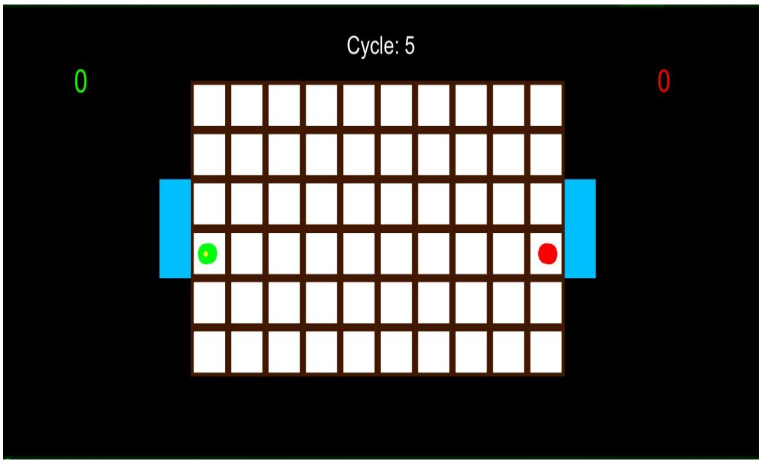
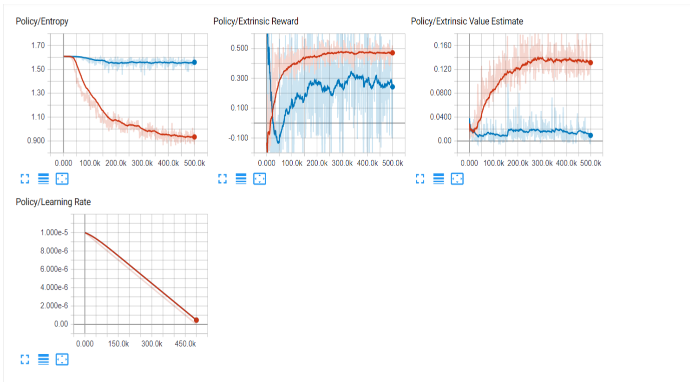


Figure 1: There are two players, red and green. In each set, a player has the ball; therefore, acts as an attacker, and the other one plays as the defender. The aim is to send the ball to the other player's goal while the defender must prevent it. If the attacker and defender reach the same block, the attacker loses its ball and becomes the defender whereas the defender gets the ball and becomes an attacker. Each player can move four directions, and they can only move one block between cycles. For instance, if the cycle is 0.5s, then every 0.5s each player can move one block. In each match, the roles are reversed.

Based on Juliani et al.'s works, now Unity Engine can be used as a platform for not only game development but also for training agents in game environments using ML-Toolkit. That is to say, researchers can create their either 2D or 3D environment with the tools provided for developing games. Also, they have more freedom in simulating the real world due to its physics engine, filling the *reality gap*(Jakobi et al., 1995) as much as possible. Juliani et al. believes that environments should challenge algorithms in four aspects of intelligence simultaneously: sensory, physical, cognitive, and social. In Juliani et al.'s study, it is stated that the Unity Engine can address all four axes.

In this experimental study, the application and implementation of both the environment and agents in the Unity Engine are examined. We created a 2.5D (a 2D game in a 3D environment) video game called "GridSoccer" in which there are two players against each other competing in a discrete environment. The Proximal Policy Optimization (PPO) algorithm is used to train agents.



(a)

(b)

Figure 2: The designed iteration loop puts the agents against each other in each stage. In the first one, An agent competes with an AI with random actions. Later in the next stage, the new agent tries to beat the agent from the previous one. As we look at the result, the training in the first stage (marked in blue) is inconsistent since the agent is competing against an AI with random actions. In the second stage (demonstrated in red), however, there is a 0.2 increase in the rewards obtained, and based on the beta's value, the learning process is stable and correct. As it is apparent, the loop can go on, and each time the new agent is trained better and, consequently, the outcomes improve.

2 Result and Discussion

The project's main motive is to experiment with MLToolkit to discover Unity Engine's potential as a benchmark for AI researchers. We created "GridSoccer", a 2.5D video game, and trained agents with the policy-based Proximal Policy Optimization (PPO) algorithm. The iteration loop has helped the agents to train better in the next stages and can be used in other environments when two players are competing with each other.

Our study has shown that agents can learn how to attack and score the goal as quickly as possible. However, the same is not true for the defender, as its reward function is somehow not as clear as the attacker. For instance, defenders have to defend their goal, block dangerous paths, and get the ball from the attacker whereas the attacker only has to reach the goal and avoid being caught by the defender. The agents have not superseded humans in both tasks. However, The iteration loop has helped the new agents train better in the each new stage, and can be used in other environments when there are two players competing with each other.

A future study might explain the reason why the agents did not have a good performance in this environment, such as by examining the reward function, training variables, or experimenting with other policy-based algorithms. Another interesting study is to tune the reward function based on the player's performance, implementing a dynamic difficulty that changes based on player's skill. That is to say, the reward function can be implemented in a way to maximize the intensity of the game, leading the player to have more fun or reach the flow zone.

There are numerous studies related to using video games as a benchmark for AI research, but there is little in using these new technologies for better AI in video games.(Statt, 2019) The reason is that game designers do not like to have unpredictable variables in the experience they create for players. An interesting study might be about how to set some standards for reward functions to make this study suitable for designers; setting some boundaries for the agents and standardize rewards for a more fun experience with a lot of variety. It can even boost the development process, decrease costs, and can even be used by game designers to prototype their ideas for how Non-Playable-Characters (NPCs) react. Therefore, the MLToolkit may even open new doors for expanding gameplay experiences, or it can be used as a tool for game developers and designers.

References

K. Arulkumaran, M.-P. Deisenroth, M. Brundage, and A.-A. Bharath. A brief survey of deep reinforcement learning. *IEEE Signal Processing Magazine, Special Issue on Deep Learning for Image Understanding*, 2017. arXiv:1708.05866v2.

C. Beattie et al. Deepmind lab. 2016. arXiv:1612.03801v2.

M. G. Bellemare, Y. Naddaf, and J. Veness. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research 47*, 2013. arXiv:1207.4708.

G. Brockman, V. Cheung, L. Petterson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. 2016. arXiv:1606.01540v1.

V. Francois-Lavet, P. Henderson, R. Islam, and J. Bellemare, Marc G.and Pineau. An introduction to deep reinforcement learning. *Foundations and Trends in Machine Learning*, 11(3-4), 2018. arXiv:1811.12560v2.

N. Jakobi, P. Husbands, and I. Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. *European Conference on Artificial Life. Springer.*, 1995.

M. Johnson, K. Hofmann, T. Hutton, and D. Bignell. The malmo platform for artificial intelligence experimentation. *IJCAI*, 2016. preprint.

A. Juliani, V.-P. Berges, E. Vckay, Y. Gao, H. Henry, M. Mattar, and D. Lange. Unity: A general platform for intelligent agents. 1984. arXiv:1809.02627v2.

A. Juliani1, A. Khalifa, V.-P. Berges, J. Harper, E. Teng, H. Henry, A. Crespi, J. Togelius, and D. Lange. Obstacle tower: A generalization challenge in vision, control, and planning. 2019. arXiv:1902.01378.

M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaskowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning. *Proceedings of IEEE Conference of Computational Intelligence in Games 2016*, 2016. arXiv:1605.02097v2.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015.

Machado, M. C., and et al. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. 2017. arXiv:1709.06009v2.

T. Mitchell and M. Hill. *Machine Learning Textbook*. McGraw-Hill Education, 1997. cs.cmu.edu/ tom/mlbook.

N. Statt. How artificial intelligence will revolutionize the way video games are developed and played, March 2019. URL `https://www.theverge.com/2019/3/6/18222203/video-game-ai-future-procedural-generation-deep-learning`.