The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

A Deep Reinforcement Learn-Based FIFA Agent with Naive State Representations and Portable Connection Interfaces

Matheus Prado Prandini Faria,[1] **Rita Maria Silva Julia,**[1] **Lídia Bononi Paiva Tomaz**[2]
[1]Federal University of Uberlândia, [2]Federal Institute of Triângulo Mineiro
matheusprandini.96@gmail.com, ritasilvajulia@gmail.com, ldbononi@gmail.com

Abstract

Video games have proved to be a very defying laboratory to study machine-learning techniques, such as Deep Reinforcement Learning (DRL) algorithms. This paper presents a new approach for a DRL-based agent trained through Deep Q-Network (DQN) technique to perform free kicks in FIFA 18 game. The main motivation for choosing this case study is the fact that, like in many situations of the real life, FIFA represents a stochastic environment. Coping with this task, the main contributions of the present paper consist on: inspired on the OpenAI Gym and on the OpenAI Universe platforms, implementing a new user-friendly interface (in terms of portability and use simplicity) to connect the learning module with the 3D FIFA's game environment; implementing a DRL-based agent for free kicks in FIFA that uses two distinct data representations retrieved from lower cost computational procedures. The results were validated through two evaluative parameters: score of well succeed kicks and training time.

1 Introduction

Reinforcement Learning (RL) is a machine-learning technique in which decision-makings are successively refined through the results of the impact of the actions executed by the agents in the environment in which they operate. Such impact is represented by a numerical reward signal received as a feedback from this environment (Sutton and Barto 1998).

RL has enjoyed a great increase in popularity over the past decade by controlling how agents can take optimal decisions when facing uncertainty (Dedieu and Amar 2017). RL methods have been widely studied in many disciplines, such as operational research, simulation-based optimization, evolutionary computation and multi-agent systems, including games (Torrado, Bontrager, and Togelius 2018).

Recently, the Deep Learning (DL) techniques have taken in the RL methods, producing a new research branch called Deep Reinforcement Learning (DRL). DRL has proved to be very successful in mastering human-level control policies in a wide variety of tasks, including the video games domain. Examples of successful applications include the learning architecture called Deep Q-Learning, which achieved superhuman performance in many Atari games (Mnih et al. 2015), as well as AlphaGo-Zero (Silver et al. 2017b) and AlphaZero (Silver et al. 2017a).

Various platforms for testing and developing DRL algorithms in Video Games have been conceived. These platforms provide interfaces that allow for the connection between the learning module of the agents and the environment of the game. Among these platforms, open-source OpenAI Gym (Brockman et al. 2016) and OpenAI Universe (Mousavi, Schukat, and Howley 2018) can be highlighted.

This work uses the FIFA 19 as a case study. It is a complex 3D football simulation Video Game developed and published by Electronic Arts (EA Games), released worldwide in the second half of 2018. Among the factors that make the FIFA problem an interesting case study, it highlights the fact that it presents a stochastic environment, that is, the same action, executed from the same state, does not always produce the same outcome. In this way, here the authors investigate the use of DRL algorithms - more specifically, the Deep Q-Network (DQN) - , in the free kicks module of FIFA 19. The present work is inspired on the proposal presented in (Trivedi 2018), improving such approach in the following manner: 1) based on the OpenAI Gym and on the OpenAI Universe platforms, this work proposes a new user-friendly interface (in terms of portability and use simplicity) to connect the learning module with the 3D FIFA's game environment; 2) In this work it is implemented a DRL-based agent for free kicks in FIFA that uses two distinct simplified data representation retrieved from lower cost computational procedures. The results were validated through two evaluative parameters: score of well succeed kicks and training time.

This paper is presented according to the following structure: section 2 resumes the Theoretical Foundations; the related works can be found in section 3; section 4 presents the proposed agent; the results are presented and commented in section 5; finally, section 6 show the conclusions and the future works.

2 Theoretical Foundations

In this section, a brief summary of the DQN model and the FIFA's environment is presented.

2.1 Deep Q-Networks (DQN)

Deep Q-Network (Mnih et al. 2015) is a value-function based DRL algorithm which achieved scores across a wide range of classic Atari 2600 video games that were comparable to that of a professional video games tester (Arulkumaran et al. 2017). DQN combines the advantages of deep learning using deep Convolutional Neural Networks (CNN) for abstract representation with the Q-learning method (Sutton and Barto 1998) in order to learn an optimal policy based, exclusively, on the images that represent the game states.

The great success of the DQN lies in the fact that it addressed some stability issues of previous neural network models (Mousavi, Schukat, and Howley 2018) with two solutions: 1) it utilizes the experience replay (Lin 1992) approach to reduce the problem of high correlation between subsequent states, enabling the RL agent to use stochastic mini-batch updates with uniformly random sampling on previously observed data offline; 2) it uses a target network (Mnih et al. 2015) to ensure that the policy does not change frequently, assuring that past experiences are not forgotten. Such network is kept frozen for a period of time and updated periodically.

2.2 FIFA 19 Environment

FIFA 19 is a football simulation video game, developed and published by EA Games, released worldwide in the second half of 2018. In particular, this work explores the free kicks task.

The free kicks training mode explored by this work deals with situations with and without barriers simulating players blocking the goal. Despite the difficulty of scoring goals in the presence of these barriers, it does not include a goalkeeper. So, it is a simpler scenario.

It is necessary to emphasize that there is a random characteristic in this game mode, in the sense that in a certain state, not always the same action results in the same outcome. For example, in a situation with a barrier of players, a kick on the barrier with the same power and the same direction can result in goal in some moments and in others in non goal. So the environment is inherently stochastic, which is a major challenge in the learning process of an agent.

3 Related Works

In 2015, Google DeepMind published a variant of Q-learning (DQN) that learn successful policies directly from high-dimensional sensory inputs (Mnih et al. 2015). They tested the agent on the classic Atari 2600 by means of the Arcade Learning Environment (ALE), using only screen pixels and the game score as inputs, it surpass the performance of all previous algorithms and achieve human level across 49 games.

OpenAI Gym is an open-source tool designed for testing and developing DRL algorithms (Brockman et al. 2016). This platform makes available several interfaces for connecting the learning module to various game environments, such as Atari 2600 using ALE, classic games, MuJoCo physics engine, among others. OpenAI Universe is a platform that supports more complex environments for Flash games, browser applications and also desktop games such as Grand Theft Auto IV (Mousavi, Schukat, and Howley 2018). However, distinctly from OpenAI Gym, OpenAI Universe does not allow to the research community to include new environments to its repository, making impracticable the use of this platform for games that do not belong to its built-in environments.

In (Trivedi 2018), the authors present a short description of the free kicks problem in FIFA's game and how DRL was applied to solve this task. Further, they implement an interface to connect the agent with the FIFA game, since the aforementioned platforms do not support such game. Distinctly from the naive game screenshots adopted in the present work, (Trivedi 2018) processes the game screenshots through a complex pre-trained CNN giving 128-dimensional flattened feature map as data representation. The resulting model achieved about 50% rate of well succeeded goals with 1000 training epochs in a GPU GTX-1070. Further, (Trivedi 2018) did not use a based object-oriented programming to implement the interface between the agent and the environment, which compromises the portability of such interface.

4 The DRL-based Agent for FIFA

4.1 Formulating Free Kicks as a RL Problem

In the free kicks task the agent's goal is to score as many kicks as possible in the simple scenario described in subsection 2.2. The scene is retrieved from approximately 180° front perspective views. The agent scores only if the ball enters the goal. Whenever a kick action is completed, the game resets to one of the following situations: open goal without any barrier; ball on the right side with a barrier also on the right side; or ball on the left side with a barrier also on the left side.

In this way, it is possible to formulate the free kicks task as a reinforcement learning problem as follows:

- **States**: the real game state is not fully represented in its corresponding image, since relevant information concerning it underlies within the game's engine, which is not accessible. Thus, the performance of the learning process strongly depends on the state representation. In this paper, two forms of state representation are explored, as presented in subsection 4.2.
- **Actions**: there are four possible legal actions to be executed: move left, move right, low kick and a high kick at an established height (which usually ranges from the height of the barrier and the height of the goal). It is important to note that the kick power is considered constant in both possibilities of kicking, being empirically defined.
- **Rewards**: after the execution of a kick action, if the agent scores a goal, then the reward is positive. Otherwise, the reward for the agent is negative. For intermediate actions (movements to left and right) the reward is slightly negative. The finality of this negative reward is to avoid sequence of moves that would generate undesirable infinite loops of movements.

4.2 Agent's Model Architecture

As well as at work (Trivedi 2018), the approach chosen to solve the free kicks problem was to use a baseline DQN model. In order to cope with the free kicks task by using DQN, this work proposes two naive representations for the game states that are distinct from that used by (Trivedi 2018). In short, here, the authors propose lower cost computational procedures for retrieving these images, since, instead of using a complex pre-trained CNN for performing the feature extraction, the raw images themselves are used.

Representation 1: Original DQN Representation The first representation used is similar to the one explored by the original DQN's paper to solve Atari games (Mnih et al. 2015). In this sense, the raw inputs of the four most recent frames are preprocessed by downsizing form the original size to a more manageable 84x84, converting to grayscale and then stacked. It's possible to note that this preprocessing step throw away color information that could be helpful in order to allow the agent to be aware of the direction and speed of each sprite. Therefore, each state is represented by a 84x84x4 image produced by this preprocessing step.

Representation 2: RGB Representation The second representation used is the RGB image portraying the current frame. RGB (red, green, and blue) refers to a system for representing the colors to be used on a computer display. Unlike the **Representation 1**, the preprocessing step just downsize from the original size to 84x84 but it considers the color information contained in the images. Therefore, each state is represented by a 84x84x3 (3 channels of colors of the RGB) image produced by the preprocessing step.

Model The Deep Q-Network was implemented from scratch following the experiment outlined in (Mnih et al. 2015) using the Keras library (Chollet and others 2015). In this sense, agents were implemented for each data representation presented. The first agent, called DQN R1, uses the **Representation 1**, while the second agent, namely DQN R2, uses the **Representation 2**.

4.3 The Connection Interface

Another contribution of this study is the implementation of an interface to connect a DRL agent to the complex environment FIFA 19. Distinctly from the interface built by (Trivedi 2018), which is highly specific to the free kicks task in the FIFA games, the platform proposed here has a more generic character, also presenting the advantage of being user-friendly in terms of portability and usability. This interface is implemented in python language and is composed of two modules: View and Controller, as shown in Figure 1 and detailed in the sequence.

The *Connection Interface* performs the connection between the agent and the environment. It is composed of the following elements: *View* module, which has as its purpose getting screenshots of the game and passing them to the agent, allowing it for perceiving the current state of the environment and for making an appropriate decision; and *Controller* module, whose objective is simulating keyboard actions directly on the game environment. In other words, it allows for the agent actuating on the environment. In this way, coupled with the *View module*, the *Controller* module enables the agent to measure the impact of its actions, which

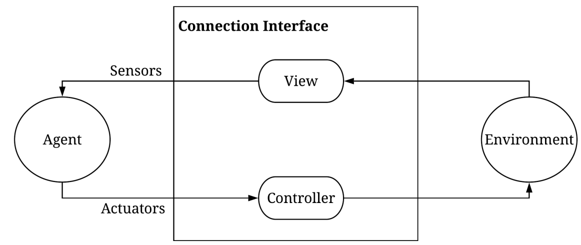

Figure 1: High-level interaction flow between an agent and the environment through the proposed interface

allows it to improve its decision-making policy. It is important to point out that it is necessary to create a reward function for the environment, but this does not interfere in the two aforementioned modules.

As the interface proposed here is highly based on the OpenAI frameworks, widely used in several researches, it becomes more user friendly and understandable both for new users and for experienced users. In addition, the object-oriented programming paradigm was used to implement such an interface, which makes it more flexible and also easy to be enhanced with new functionalities. Details of the implementation of both the connection interface and the DQN model are found at https://github.com/matheusprandini/FifaFreeKickLearning2019.

5 Experiments and Results

The experiments here has as their purpose evaluating, through the connection interface presented in subsection 4.3, the quality of the learning process of the agents taking into account both naive data representations proposed in subsection 4.2. This evaluation will be performed by means of comparative tests based on the following parameters: rate of well succeeded goals in tests involving the naive representations proposed here and the representation used in (Trivedi 2018); and the training time in tests concerning both naive representations proposed here. The experiments ran on a machine with a GPU Nvidia GeForce GTX-745 and 16 GB RAM. The evaluation of the rate of well succeeded goals was performed in the context of two test scenarios: the first one (I) evaluates the quality of the DQN agents trained with the four actions described in subsection 4.1 (move to left and right, low and high kicking); the second scenario (II) evaluates the quality of both agents operating according to the following three actions: move to left, move to right and low kicking. In each test scenario, each agent was trained in the course of 5000 epochs (an epoch ends when a kick action is performed).

Table 1 presents the rates of well succeeded goals of the agents DQN R1 and DQN R2 in both scenarios, showing a slight superiority of the DQN R2 agent in both situations. The lower rates of scenario II compared to scenario I can be explained by the following fact: it is more difficult to score goals without counting on the *high kicking* action, especially in situations involving barriers.

	DQN R1	DQN R2
Scenario I	75%	80%
Scenario II	50%	56%

Table 1: Rates of well succeeded goals with 5000 training epochs for each scenario

Table 2 shows the comparison between the agents implemented here and the agent presented in (Trivedi 2018), named as DQN T (Trivedi). It is important to note that this test takes into account just one scenario (scenario I) involving 1000 training epochs, since only this scenario was performed in the mentioned work. The results show a considerable superiority of the DQN R1 and DQN R2, even using a less powerful GPU than in (Trivedi 2018). The authors believe that the better performance obtained by the naive representation R2 proposed in the present work is due to the adding of color information and the fine-tuning of the hyperparameters of the DQN model.

	DQN R1	DQN R2	DQN T
Scenario I	58%	62%	50%

Table 2: Rates of well succeeded goals with 1000 training epochs for scenario I

Finally, here the experiments for evaluating the training time parameter involve both agents implemented in this work. In (Trivedi 2018), this parameter was not evaluated. In the first experiment, the training time of the DQN R1 agent was 16 hours, whereas the DQN R2 took 14 hours to train. In the second experiment, the training time of the agents DQN R1 and DQN R2 were, respectively, 22 hours and 18 hours.

6 Conclusion and Future Works

The results confirmed that the naive representations proposed, as well as the interface implemented, allowed for a satisfactory performance of the agents implemented according to such approachs, since both agents in general overcame the predecessor (Trivedi 2018) in terms of succeeded goals. With respect to the training time, **Representation 2** proved to be more appropriate than **Representation 1**. As future works, the authors intend to study new DRL-based strategies to cope with the remaining modes of FIFA, such as: inserting the goalkeeper in the scene of the game and detecting appropriate dynamics of passe exchanging among the players.

7 Acknowledgement

To Capes for financial support.

References

Arulkumaran, K.; Deisenroth, M. P.; Brundage, M.; and Bharath, A. A. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34(6):26–38.

Brockman, G.; Cheung, V.; Pettersson, L.; and Schneider, J. 2016. Openai gym. *CoRR*.

Chollet, F., et al. 2015. Keras. https://keras.io.

Dedieu, A., and Amar, J. 2017. Deep reinforcement learning for 2048. http://www.mit.edu/~adedieu/pdf/2048.pdf.

Lin, L.-J. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning* 8(3):293–321.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; and Rusu, A. A. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Mousavi, S. S.; Schukat, M.; and Howley, E. 2018. Deep reinforcement learning: An overview. *CoRR* abs/1806.08894.

Silver, D.; Hubert, T.; Schrittwieser, J.; and Antonoglou, I. 2017a. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR* abs/1712.01815.

Silver, D.; Schrittwieser, J.; Simonyan, K.; and Antonoglou, I. 2017b. Mastering the game of go without human knowledge. *Nature* 550:354–.

Sutton, R. S., and Barto, A. G. 1998. *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st edition.

Torrado, R. R.; Bontrager, P.; and Togelius, J. 2018. Deep reinforcement learning for general video game ai. *CoRR* abs/1806.02448.

Trivedi, C. 2018. Using deep q-learning in fifa 18 to perfect the art of free-kicks. https://github.com/ChintanTrivedi/DeepGamingAI_FIFARL.