Reinforcement Learning of Intelligent Characters in Fighting Action Games

Byeong Heon Cho[1], Sung Hoon Jung[2], Kwang-Hyun Shim[1], Yeong Rak Seong[3], Ha Ryoung Oh[3]

[1] Digital Content Research Division, ETRI, Daejeon, 305-700 Korea
[2] Dept. of Information and Comm. Eng., Hansung Univ., Seoul 136-792 Korea
[3] School of Electrical Engineering, Kookmin Univ., Seoul 136-702 Korea

Abstract. In this paper, we investigate reinforcement learning (RL) of intelligent characters, based on neural network technology, for fighting action games. RL can be either on-policy or off-policy. We apply both schemes to *tabula rasa* learning and adaptation. The experimental results show that (1) in *tabula rasa* leaning, off-policy RL outperforms on-policy RL, but (2) in adaptation, on-policy RL outperforms off-policy RL.

1 Overview

Reinforcement Learning (RL) is one of the learning algorithms for Neural Networks(NNs) [1]. The RL NN learns how to achieve the goal by repeating trial-and-error interactions with the environment. Generally, in RL, a NN can be either on-policy or off-policy [2]. With on-policy RL, the NN's decision is reflected to output. Off-policy RL produces no output but observe the decision produced by another static algorithm.

In this paper, we present the RL of NN-based intelligent characters (IC) for fighting action games. We categorize IC's learning into two classes. In *tabula rasa* learning, the IC has no initial knowledge about the game. Thus, it must learn everything. On the other hand, in adaptation, the IC has previously learned about its environment, including game rules and the opponent's action pattern. However, since the environment is abruptly changed now, the IC must re-learn it. For each case, both on-policy and off-policy RL methods are applied, and their performance is compared.

2 *Tabular Rasa* Learning

In this section, we address *tabula rasa* learning of an IC. That is, the NN is initially unaware of its environment, e.g. game rules and OC's action pattern, and must learn everything. Both on-policy and off-policy RL methods are applied. Fig. 1 illustrates them.

In Fig. 1(a), the NN senses the state of the environment and produces actions. Then, the IC associated with the NN executes the actions, which affect the environment. The IC and its opponent score according to the fitness of the actions. Finally, the NN receives a scalar reward value based on the score difference

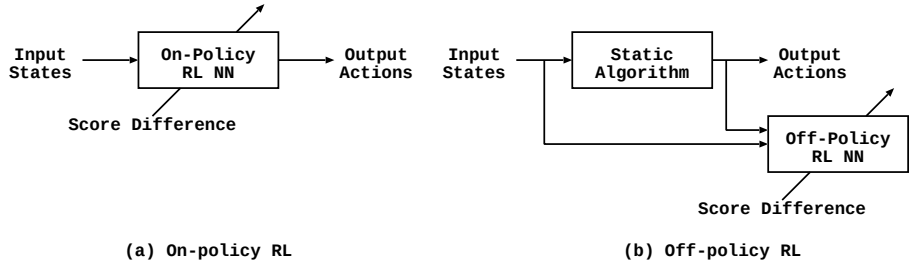

Fig. 1. On- and off-policy reinforcement *tabula rasa* learning

between the two characters. On the other hand, in Fig. 1(b), the IC executes the action produced by a static algorithm, not by the NN. The action is also used to teach the NN. That is, the intermediate learning results do not affect the environment until the learning finishes. In this paper, the static algorithm takes random actions so that the NN explores its environment in an unbiased manner.

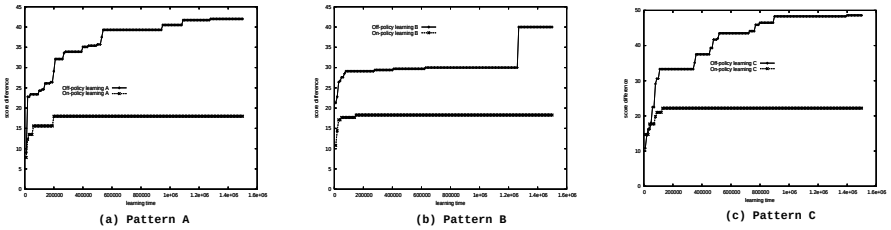

Fig. 2. Result of *tabula rasa* learning

We apply the above two methods to the IC of [3], and compare their performance. We use three action patterns. For each case, we use 10 random initial seeds and measure the average of the score ratio between two characters. Fig. 2 shows the results. For all patterns, off-policy RL outperforms on-policy RL. While the score difference quickly converges in on-policy RL, the final score difference in off-policy RL is 2-2.5 times larger than that in on-policy RL. This is because since the NN should learn everything, it would be better to practice as large a variety cases as possible. In off-policy RL, the static algorithm randomly produces output actions. However, in on-policy RL, although the NN is randomly initialized, its output is not sufficiently random.

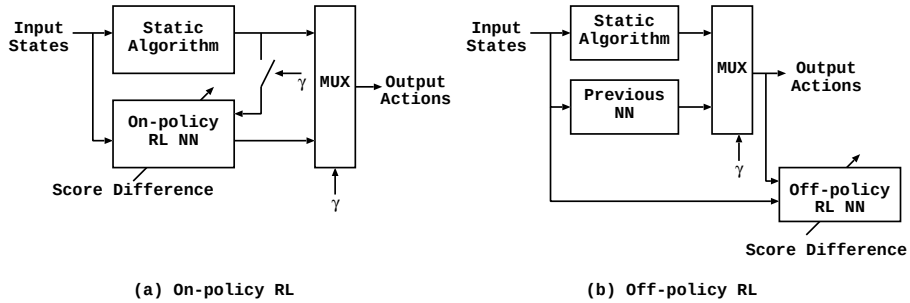

Fig. 3. On- and off-policy adaptation

3 Adaptation

With adaptation, an IC can adjust itself to environmental change. Unlike the previous section, the NN has initial knowledge although it is partially unsuitable for the present environment. Fig. 3 illustrates on-policy and off-policy adaptation.

Fig. 3(a) shows on-policy adaptation. The NN produces output depending on input states. However, as opposed to Fig. 1(a), the output does not always determine which actions the IC will execute in a given input state. The NN controls the IC just at a rate of γ. At other times, a static algorithm decides output actions. However, in both cases, the NN learns the result of output actions by using generated score difference. Thus, the NN can have new experiences with the aid of the static algorithm. Like *tabula rasa* learning, we use a random generator as the static algorithm. On the contrary, as shown in Fig 5(b), the off-policy RL NN does not produce output. At the beginning of adaptation, it is copied from the previous NN which was trained within the previous environment. Then, it only learns by using the output actions, determined by either a static algorithm or the previous NN relying on γ, and the resulting score difference until the end of adaptation. However, after finishing adaptation, it produces output to decide the IC's behavior.

These two methods are applied to the adaptation algorithm of [4]. At first, the OC, which is used for this experiment, acts with one of the three action patterns for building IC's initial knowledge. Then, its action pattern is changed to another pattern. Fig. 4 shows the results. In on-policy RL, although the score ratio is very low during the early stage of adaptation, it strongly increases after all. Thus, as opposed to Section II, on-policy RL outperforms off-policy RL in most cases. This is because the on-policy RL NN immediately learns wrong decisions. That is, when its decision produces bad results, the NN updates the link weights to prevent the same wrong decision. Thus, it has more chances than its rival to try new actions. This allows the exploration of a larger search space. Thus, the on-policy RL NN eventually makes better decision.

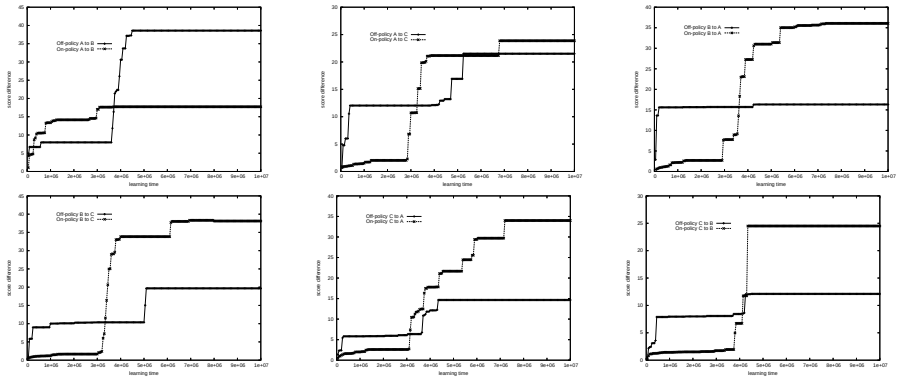

Fig. 4. Result of adaptation

4 Conclusion

In this paper, on-policy and off-policy RL for an IC in fighting action games based on NN technology are investigated. Both learning schemes are applied to *tabula rasa* learning and adaptation for the IC and their performance is compared. The experimental result tells that (1) when the NN has no initial knowledge (*tabula rasa* leaning), off-policy RL produces better performance, but (2) when the NN has some knowledge, which is partly valid (adaptation), on-policy RL outperforms the rival.

Acknowledgements

This work was supported in part by research program 2006 of Kookmin University in Korea.

References

1. R. P. Lippmann. An introduction to computing with neural nets. IEEE ASSP Magazine, vol. 4, no. 2, pp.4–22, 1987.
2. K. R. Dixon, R. J. Malak, and P. K. Khosla. Incorporating prior knowledge and previously learned information into reinforcement learning agents. Tech. Rep., Institute for Complex Engineered Systems, Carnegie Mellon University, 2000.
3. B. H. Cho, S. H. Jung, Y. R. Seong, and H. R. Oh. Exploiting intelligence in fighting action games using neural networks. To appear in IEICE Trans. on Information and Systems.
4. B. H. Cho, S. H. Jung, K. H. Shim, Y. R. Seong, and H. R. Oh. Adaptation of intelligent characters to changes of game environments. CIS 2005, Part 1, LNAI 3801, pp.1064–1073, 2005.