Mastering Fighting Game Using Deep Reinforcement Learning With Self-play

Dae-Wook Kim: dooroomie@etri.re.kr
Sungyun Park: tjddbs@etri.re.kr
Seong-il Yang: siyang@etri.re.kr
Published by IEEE Xplore: 20 October 2020

Outline

- Introduction
- Fihgting
- Reward
- Method
- Experiment
- Result
- Conclusion

Introduction

- In this paper, they propose a method to create fighting game AI agent using deep reinforcement learning with self-play and Monte Carlo Tree Search (MCTS).
- Also analyze various reinforcement learning configuration novel performance metric. Agent trained by the proposed method was evaluated against other AIs.
- FightingICE was used for this experiment.

Fighting ICE

- One-on-one fighting game
- There is such a thing as a delay frame in “Fighting ICE”

FightingICE

- The information that can be obtained within this game is shown in this table

	Feature Name	Value	Size	Feature Name	Value	Size	Feature Name	Value	Size
Character Attribute (for each player)	HP	0∼1	1	Energy	0∼1	1	EnergyT5	0∼1	1
	EnergyT30	0∼1	1	EnergyT50	0∼1	1	EnergyT150	0∼1	1
	X position	0∼1	1	Y position	0∼1	1	X movement	0 or 1	1
	Y movement	0 or 1	1	X velocity	0∼1	1	Y velocity	0∼1	1
	Action	0 or 1	56	Character state	0 or 1	4	Remaining frame	0∼1	1
	Controllable	0 or 1	1						
Projectile Attribute (for each player)	X position	0∼1	2	Y position	0∼1	2	Damage	0∼1	2
Distance Attribute	Distance	0∼1	1	DistanceT	0 or 1	3			
Time Attribute	Remaining time	0∼1	1						

Reward

- The first is the HP difference.
- Secondly, the rewards obtained vary depending on the round win or loss.
- Finally, if you do not defeat the enemy within the time of the round, AI will receive a -10 reward.

Method

- AI uses a combination of MCTS and self-play learning methods.
- Self-play learning is a method in which AIs repeatedly play against each other and take turns learning.
- The algorithm used for deep reinforcement learning is Proximal Policy Optimization. Simulation frames are generated to predict behavior.

Method

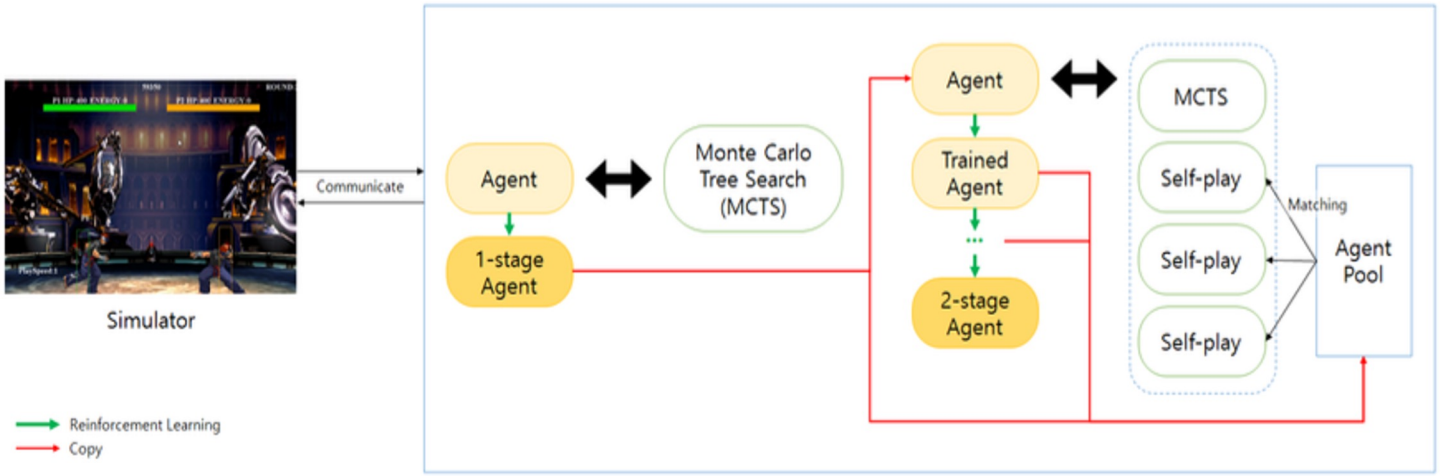


Fig. 1. An overview of proposed two stage reinforcement learning. In the first stage, agent is trained against MCTS AI alone. After performance saturation in the first stage, agent is trained against itself as well as MCTS AI. During training in second stage, agent is duplicated into agent pool to find its weakness.

Experiment

- Perform three experiments
- The experiment used the top-ranked MCTS AI from the 2017, 2019 FightingICE competition as the opponent.
- The characters used by both AIs are the same

Experiment

- In the first experiment, they will conduct a match between the AIs we have created so far and obtain the win rate, average remaining HP, and average remaining time, respectively.
- In this matchup, they will obtain the win percentage, average remaining HP, and average remaining time, respectively.
- In the last experiment, they calculate the SDR scores of the trained AI for three different rewards, two different rewards, and one different reward, respectively.

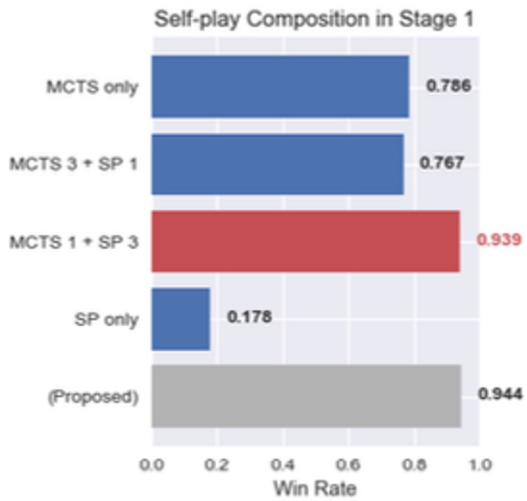
Experiment

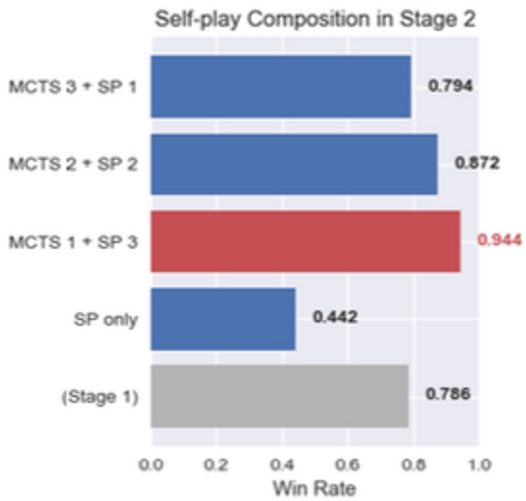
- It also gives an overall rating called the SDR score.
- Using this formula, they evaluated the AI.

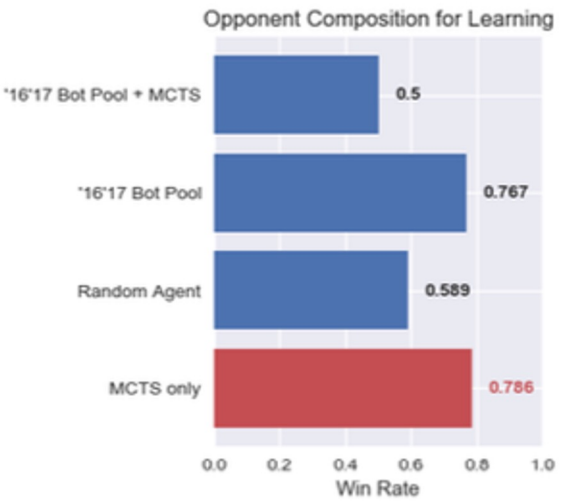
$$\text{SDR Score} = \frac{(\text{Win Rate}) \times (\text{HP Difference})}{(\text{Elapsed Time})}$$

Result(experiment 1)

- The AI with the highest win rate was the MCTS AI 1 : Self-play 3 ratio.







Result(Experiment 2)

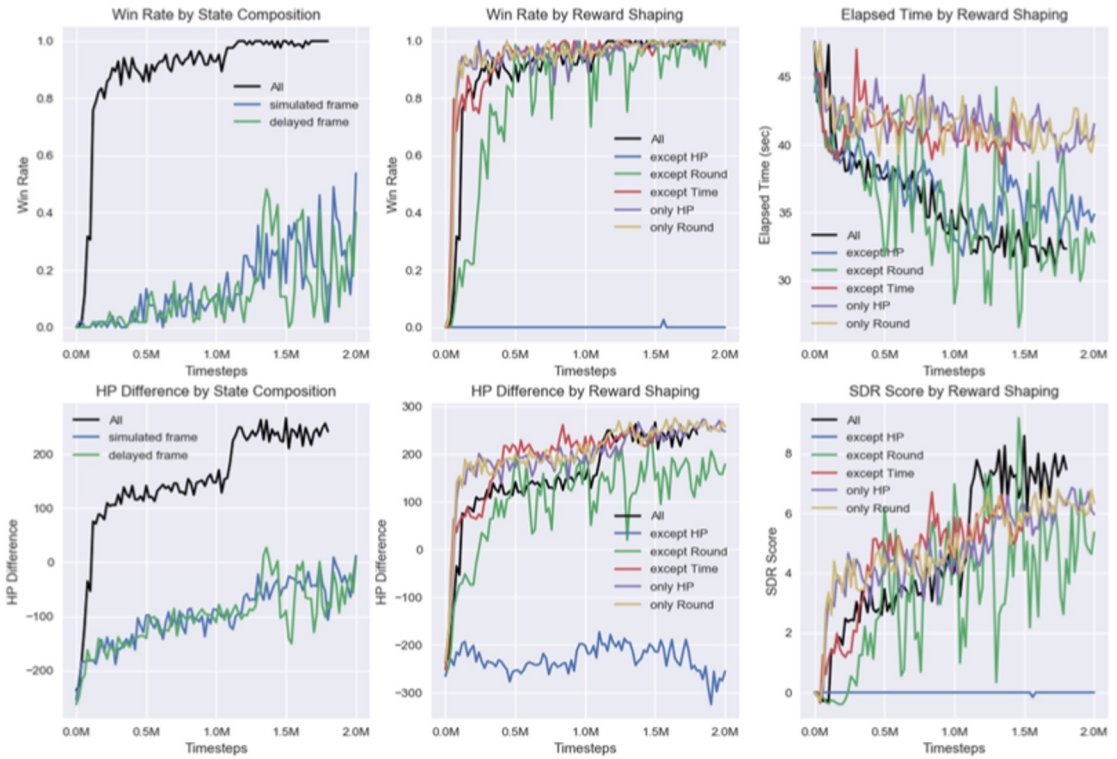
- The results of Experiment 2 are as follows

EVALUATION RESULT OF THE BEST AGENT TRAINED BY PROPOSED METHOD

Submitted Year	Rank	AI Name	Remaining HP (Proposed Agent)	Remaining HP (Competitor)	Elapsed Time (sec)	Win Rate
2017	1	GigaThunder	395.56	0.0	19.96	1.0
	2	FooAI	350.56	0.0	21.12	1.0
	3	JayBot_2017	239.22	0.0	28.99	1.0
	4	Mutagen	60.28	53.44	45.77	0.556
2019	3	Toothless	337.44	25.67	24.65	0.889
	4	FalzAI	185.61	0.0	33.86	1.0
	5	LGIST_Bot	140.50	0.0	37.33	1.0
	7	HaibuAI	254.50	0.0	35.42	1.0
	8	DiceAI	310.44	0.0	31.65	1.0
-	-	MctsAi	198.89	0.0	32.41	1.0
		Average	**247.30**	**7.91**	**31.12**	**0.944**

Result(Experiment 3)

- The results of Experiment 3 are shown here.
- The simulated and delayed frames are considered to complement each other
- Rewards related to HP are very important.



Conclusion

- In this paper, two AI designs were proposed: deep reinforcement learning using MCTS and Self-play.
- The AI created with a ratio of MCTS 1 : Self-play 3, in addition to rewards from delayed and simulated frames, outperformed the AIs that participated in previous competitions.

Thank you for attention