Win or Learn Fast Proximal Policy Optimisation

Dino Stephen Ratcliffe
EECS
Queen Mary University of London
London, England
D.Ratcliffe@QMUL.ac.uk

Katja Hofmann
Microsoft Research Cambridge
Microsoft
Cambridge, England
Katja.Hofmann@Microsoft.com

Sam Devlin
Microsoft Research Cambridge
Microsoft
Cambridge, England
Sam.Devlin@Microsoft.com

***Abstract*—AI agents within video games are often required to compete within an environment shared by many other agents. This problem can be tackled by multi-agent reinforcement learning (MARL). One solution to MARL is to learn a Nash Equilibrium Strategy (NES) that guarantees a known minimum payoff when playing against other rational agents. We focus on one approach for learning a NES, Win or Learn Fast (WoLF), WoLF has been shown to converge towards a NES in a variety of matrix-games and grid based games. Research into Deep MARL has focused on performance against opponent agents and with limited quantitative results regarding learning a NES. We present a systematic empirical investigation into the ability of Proximal Policy Optimisation (PPO) to learn a NES, showing instability in certain matrix games. We then present an extension, WoLF-PPO, that is able to learn a policy that is closer to the NES.**

***Index Terms*—Artificial intelligence, Artificial neural networks, Multi-agent systems**

I. Introduction

In multi-agent environments the learning problem is non-stationary due to the fact that the other agents' policies are changing over time. Guarantees provided by single agent reinforcement learning rely on the problem being framed as a Markov Decision Process (MDP), but in the case of playing against learning opponents the problem becomes non-stationary and these guarantees are lost. Having a guarantee of convergence to a Nash Equilibrium Strategy (NES) would be very desirable as it allows us to learn a stationary policy that has a known lower bound payoff when dealing with rational agents. Win or Learn Fast (WoLF) attempts to provide guarantees in these situations [1] by varying the learning rate of the agent based on the performance in comparison to an estimated NES. It provides guarantees of converging to a NES in theory, and has been shown empirically to get closer to a NES than a fixed learning rate agent. In this paper we present an initial study into the ability of the popular Deep-RL approach Proximal Policy Optimisation (PPO), to learn the NES in a set of traditional game theory matrix-games. We then present an extension to PPO called WoLF-PPO, that is designed to learn policies closer to the NES. We observe that PPO can learn policies close to the NES when it happens to be the max-entropy policy. We also show that WoLF-PPO is able to learn policies closer to the NES than PPO irrespective of whether the NES is the max-entropy policy, as well as being more robust to high learning rates.

978-1-7281-1884-0/19/$31.00 ©2019 IEEE

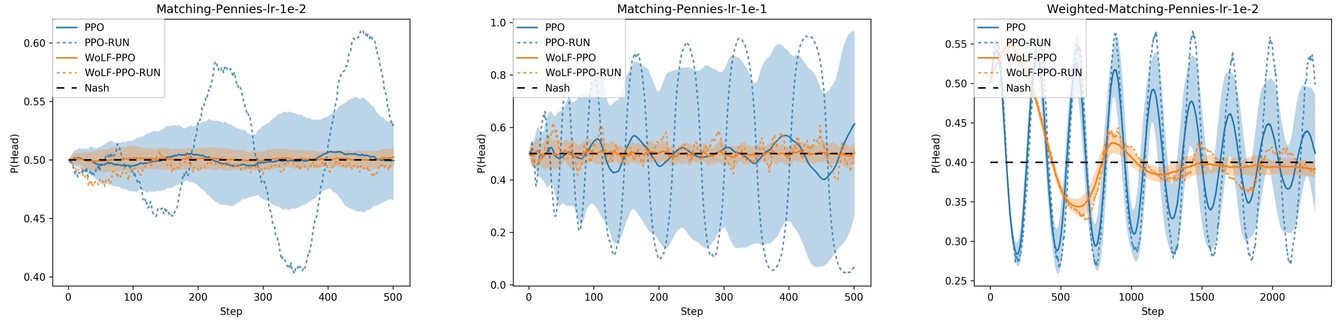
II. Background

A. Matrix Games

In this work we focus on Matrix games as they are the simplest form of game that allow us to compare the convergence properties of learning approaches to the NES. Matrix games are defined by a payoff matrix where the players of the game simultaneously pick actions and their respective payoffs are dependent on the actions of all players [2]. There are two forms of matrix games, zero-sum games and general-sum games. We focus on Zero-sum games that are strictly competitive, where if one of the agents receives a positive reward then the opposing agent receives an equal negative reward. When considering solution concepts for this framework, two are most common: the *best response* strategy and the *NES* [3]. The best response strategy is the optimal strategy against the joint actions of all the other agents. In the case of playing against a set of stationary opponents there will exist a deterministic best response strategy. The NES states that all agents should be playing a best response against the joint actions of all its respective opponents. This results in a stable equilibrium point where no player can gain an advantage by changing strategy. This means that the NES is not an optimal strategy against all agents, however it does provide stability by preventing the agent from being exploited by another strategy. NES have been proven to exist in all zero-sum games (contain a unique Nash equilibrium) and all general sum games, making it a very desirable strategy to be able to learn.

B. Properties of Multi-Agent systems

Two main properties have been identified as desirable for any multi-agent learning systems. The first of these properties is rationality, "*If the other players' policies converge to stationary policies then the learning algorithm will converge to a policy that is a best-response to the other players' policies*" [1]. The second property is convergence defined as, "*The learner will necessarily converge to a stationary policy. This property will usually be conditioned on the other agents using an algorithm from some class of learning algorithms*" [1]. Most literature focuses on obtaining these properties in self play, however some work has been empirically shown to converge with a small subset of learning agents beyond self play [1]. If two agents are both rational and convergent then



(a) Standard MP. Small learning rate.

(b) Standard MP. Large learning rate.

(c) Weighted MP. Small learning rate.

Fig. 1: WoLF-PPO and PPO matching pennies results, $P(Head)$ is the probability of picking head throughout training. Averaged over 50 runs by the solid lines. Dashed lines show single representative runs.

they will both converge to a stationary best response strategy against each other resulting in having converged to a NES.

C. *Win or Learn Fast (WoLF)*

Win or Learn Fast (WoLF) is an extension to Infinitesimal Gradient Ascent (IGA), where the step size of the learning agent is reaching an infinitesimal step size ($lim_{\eta \to 0}$) [4]. WoLF gives a stronger notion of convergence to a NES by introducing separate learning rates for when the agent is winning and when it is losing, referred to as α_{WIN} and α_{LOSE} where $\alpha_{WIN} < \alpha_{LOSE}$. The agent is judged to be winning if their current expected payoff is better than playing the Nash equilibrium. This change has the effect of the winning agent learning slower and being more "cautious" about updating its strategy until the other agent has learned to counter the new strategy. Although the proofs for this method do require knowing detailed information about the environment and opponent, a practical algorithm has been presented [1]. This method is based on Policy Hill Climbing (PHC) and is referred to as WoLF-PHC. WoLF-PHC uses an estimation of the NES. It achieves this by tracking the average policy over training. By learning Q values during training the agent can then compare the performance of the current policy to the current average policy. This gives the ability to select a larger or smaller learning rate based on the performance of the current policy compared to the estimated NES, where α_{WIN} is the learning rate used when the current policy is doing better than the current estimated NES strategy and α_{LOSE} is used when the current policy is doing worse than the estimated NES. WoLF-PHC is shown to have good results in a variety of matrix games and a gridworld based soccer game, achieving convergence to policies much closer to the NES than PHC without varying the learning rate.

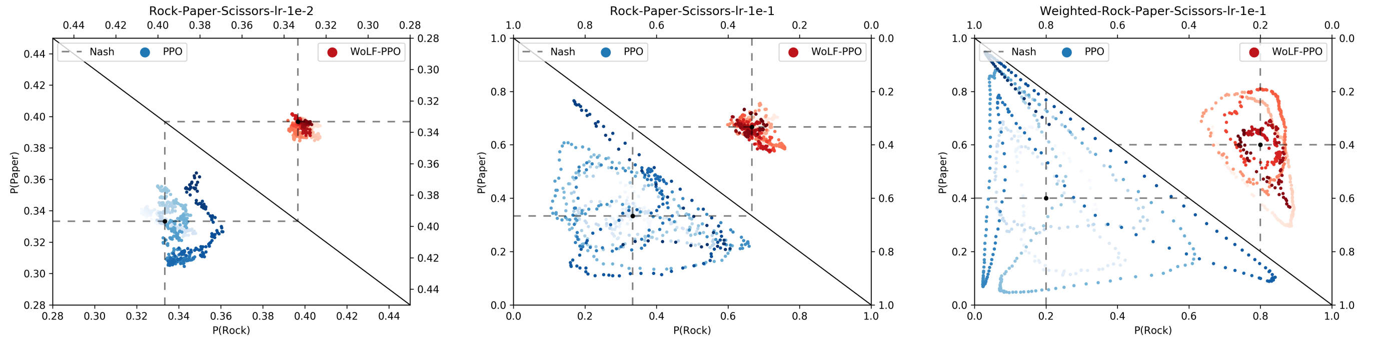
D. *Proximal Policy Optimisation (PPO)*

Proximal Policy Optimisation (PPO) is a policy based gradient method for deep reinforcement learning. It is a current state of the art deep reinforcement learning algorithm that has shown good results on Atari games, MuJoCo control tasks [5] and DoTA 2 [6]. PPO is based on the ideas introduced with Trust Region Policy Optimisation (TRPO) [7], that limit the update of the policy to a "Trust Region" defined by the distance from the old policy by a given KL divergence between the two. PPO introduces clipped probability ratios that can be used as a lower bound estimate of the policy performance. This means that PPO only requires first-order optimisation, making PPO easier to implement and support noisy architectures and parameter sharing. Critically for this paper, we emphasise that PPO has a term in its objective function that rewards higher entropy strategies. This is to prevent the agent from learning an overly deterministic policy early in training promoting exploration. PPO also introduces multiple update steps for each iteration of experience, this is made possible because of the clipped probability ratios.

E. *Multiagent Policy Gradient Methods*

There has been work attempting to use deep policy gradient methods in a multi-agent setting. Little work has been done however to evaluate the ability of these systems to learn a NES, instead focusing on performance against other approaches. The work by Lowe et al [8] focuses on performance of actor-critic methods in a range of competitive and cooperative tasks. They present a method that uses extra information at training time with a centralised critic. They also present results for their model when trained with policy ensembles, showing that it improves training stability and agent robustness. However the robustness of the agent is measured by testing the agent against other pre-trained policies. This means that the lower bound for performance is not known for these agents.

Bansal et al [9] evaluate PPO in a variety of complex fully competitive multi-agent control tasks. They observe that during training one agent would often learn to dominate the opponent to the point where the opponent is unable to recover. They introduce opponent sampling in order to give the agents a form of curriculum learning. They also observe that the agents' final policies vary greatly from run to run with different random seeds. No attempt is made in this work to evaluate the distance from the NES.



(a) Standard RPS. Small learning rate. Note the smaller scale of the axes.

(b) Standard RPS. Large learning rate.

(c) Weighted RPS. Large learning rate.

Fig. 2: WoLF-PPO and PPO rock paper scissors results, $P(Rock)$ is the probability of picking rock throughout training, $P(Paper)$ is the probability of picking paper throughout training. Colour transitions from Light to Dark over training steps.

III. Win or Learn Fast Proximal Policy Optimisation (WoLF-PPO)

When extending an approach to use WoLF, two properties are required. Due to the fact that the NES can be stochastic, the ability to learn stochastic policies is required. The other property required is that the learning rate of the agent can be varied over training. We chose to use PPO as the base agent as it meets both requirements, along with achieving good results in both single player environments and when training on large sample sizes in complex multi-agent environments [6].

The objective function that is maximised during training for PPO and our extension WoLF-PPO is shown in equation 1.

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\left[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\right] \quad (1)$$

c_1 and c_2 are coefficients. S is the max entropy bonus, L_t^{CLIP} is the clipped surrogate objective and $L_t^{VF} = (V_\theta(s_t) - V_t^{targ})^2$ the squared error loss for learning a simple state value mapping. The clipped surrogate objective L_t^{CLIP} is a clipped version of the TRPO surrogate objective, that was introduced because in TRPO this surrogate objective can result in very large policy updates. By clipping the surrogate objective PPO prevents the large policy updates by preventing the surrogate loss from being too large in the beginning.

In order to extend PPO to WoLF-PPO, we make two main changes inspired by the WoLF approach [1]. The first is to keep track of the average payoff over training, this gives us an estimation of the payoff that would be achieved when playing the NES. This works well for the games presented but would need to be extended for extensive form games.

The second change made is that we have two separate learning rates α_{WIN} and α_{LOSE}, these learning rates function in the same way as the learning rates in WoLF-PHC, where α_{WIN} is used when the current payoff is larger than the estimated payoff from the estimated NES. For all of our experiments we used a ratio of 4 between our winning and losing learning rates where $\alpha_{WIN} = \frac{\alpha_{LOSE}}{4}$.

Finally, the network used for all experiments is a fully connected feed forward neural network, with two hidden layers of 20 neurons. In our experiments we used Stochastic Gradient Descent (SGD) as we wanted to avoid interference from the adapting of learning rates introduce by ADAM [10]. We have however observed similar results when using ADAM as the optimiser.

IV. Experiments

For our empirical results we aimed to investigate the difference between PPO and WoLF-PPO. We have therefore used the same experimental setup and environments from the original WoLF paper, as these were also designed to investigate the advantages of WoLF [1]. We also introduce several weighted variants of the environments to investigate the effect of the entropy term present in PPO's objective function. In table I we present quantitative results for the distance from the NES that PPO and WoLF-PPO achieve in the various environments. As the policies will be circling around and through the NES we take the policy that had the furthest distance from the NES in the last 10 policy updates for each run and then average over all 50 runs.

A. *Matching Pennies (MP)*

Matching pennies is a two-player zero-sum game were both players pick a side of a coin (heads or tails) these choices are then revealed simultaneously with one player receiving a point if they match and the other receiving a point if they differ. This results in the NES being uniform random, if you pick the side of the coin at random then you will win 50% of the games irrespective of your opponent's strategy. The payoff matrix for the weighted variant is shown in Table IIa. This variant shifts the NES away from uniform random to $P(H) = 0.4$.

Starting with PPO and WoLF-PPO on the standard weighting of matching pennies we can see that both agents stay close to the NES as shown in Fig 1a and Table I. In this setting we see that WoLF-PPO does get closer to the NES than PPO but both agents learn strategies close to the NES with PPO within 0.056 and WoLF-PPO within 0.012.

When the learning rate is increased by an order of magnitude, as shown in Fig 1b, it is apparent that the distance from

TABLE I: Comparison of euclidean distance from NES across approaches, learning rates and games. Mean and Standard Deviation over 50 runs, Max distance taken over last 10 policy updates.

	$\alpha_{LOSE} = 0.1$		$\alpha_{LOSE} = 0.01$	
Game	PPO	WoLF-PPO	PPO	WoLF-PPO
MP	0.558 ± 0.093	$\mathbf{0.078 \pm 0.033}$	0.056 ± 0.035	$\mathbf{0.012 \pm 0.007}$
Weighted MP	0.543 ± 0.150	$\mathbf{0.066 \pm 0.029}$	0.113 ± 0.051	$\mathbf{0.085 \pm 0.041}$
RPS	0.476 ± 0.114	$\mathbf{0.078 \pm 0.032}$	0.042 ± 0.021	$\mathbf{0.013 \pm 0.007}$
Weighted RPS	0.436 ± 0.120	$\mathbf{0.080 \pm 0.040}$	0.124 ± 0.055	$\mathbf{0.077 \pm 0.010}$

the NES increases for both PPO and WoLF-PPO. However, WoLF-PPO stays much closer to the NES with PPO being within 0.558 and WoLF-PPO being 0.078. We believe that the relatively good performance of PPO in this environment is due to the maximising of entropy in PPO's objective function. In games such as matching pennies the NES is to play uniform random and thus max entropy. This results in this version of matching pennies having its NES directly optimised by the max entropy term in the objective function of both PPO and WoLF-PPO.

TABLE II: Payoff matrices for the weighted matrix games.

(a) Matching Pennies

	H	T
H	$(2, -2)$	$(-1, 1)$
T	$(-1, 1)$	$(1, -1)$

(b) Rock Paper Scissors.

	R	P	S
R	$(0, 0)$	$(-1, 2)$	$(1, -2)$
P	$(2, -1)$	$(0, 0)$	$(-1, 1)$
S	$(-2, 1)$	$(1, -1)$	$(0, 0)$

We demonstrate this phenomenon by using the weighted variant of Matching Pennies to push the NES away from the max entropy policy. In Fig 1c we can see that PPO now diverges away from the NES, increasing from a distance of 0.056 to 0.113, with the distance being larger than when dealing with non weighted matching pennies. We also see that WoLF-PPO continues to outperform PPO by on average learning a strategy within 0.085 of the NES.

B. Rock Paper Scissors (RPS)

Rock Paper Scissors (RPS) consists of three possible actions that form a cyclic winning pattern. This environment is of interest because in many commercial video games (e.g. StarCraft [11]) relative skill is non-transitive similar to this cyclic dynamic demonstrated by RPS. We again use the standard version of RPS and a weighted variant to move the NES away from uniform random. The payoff matrix for the weighted version can be found in Table IIb, giving a NES of $P(ROCK) = 0.2$ and $P(PAPER) = 0.4$.

In RPS we see very similar results to what we observed in matching pennies. In Fig 2a we show a sample run of PPO and WoLF-PPO. As shown, they both stay close to the NES. However, as with standard matching pennies, the max entropy strategy is the NES resulting in relatively good performance from PPO. In Table I we can see that PPO was within 0.042 and WoLF-PPO was within 0.013 of the NES on average. When increasing the learning rate by an order of magnitude we end up with WoLF-PPO showing a greater advantage over PPO as was the case with matching pennies, shown in Fig 2b.

We then ran PPO and WoLF-PPO on a weighted version of RPS. In Fig 2c and Table I we show that in this environment WoLF-PPO stays closer to the NES than PPO. This is consistent with the matching pennies results.We also observe that WoLF-PPO does have a reduction in performance when moving from standard to weighted matching pennies going from a distance of 0.012 to 0.085, and rock paper scissors going from 0.013 to 0.077. This is likely due to WoLF-PPO still benefiting from the max entropy term when playing the standard versions of these games, but being more robust to the influence of this term than PPO when dealing with the NES not on the max-entropy policy.

V. Conclusion

In this paper we present an initial study into the ability of PPO to learn the NES in a set of traditional game theory matrix-games. We then present an extension to PPO, WoLF-PPO, that is designed to learn policies closer to the NES. We demonstrate that PPO is able to learn policies close to the NES when it is the max-entropy policy. We also shown that WoLF-PPO is able to learn a strategy closer to the NES than PPO in a set of matrix games including games where the NES is not the max-entropy policy.

We also observed that WoLF-PPO is more robust than PPO when dealing with large learning rates. In the future we would like to expand on this work and move to extensive form games. We would then like to expand beyond what is possible with tabular RL in order to demonstrate both the advantages of WoLF and Deep RL by training on games with raw pixel state representations.

References

[1] M. Bowling and M. Veloso, “Multiagent learning using a variable learning rate,” *Artificial Intelligence*, vol. 136, no. 2, pp. 215–250, 2002.

[2] T. Basar and G. J. Olsder, *Dynamic noncooperative game theory*, vol. 23. 1999.

[3] J. Nash, “Non-cooperative games,” *Annals of mathematics*, pp. 286–295, 1951.

[4] S. Singh, M. Kearns, and Y. Mansour, “Nash convergence of gradient dynamics in general-sum games,” in *UAI*, pp. 541–548, 2000.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv:1707.06347*, 2017.

[6] OpenAI, “Openai five.” https://blog.openai.com/openai-five/, 2018.

[7] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in *ICML*, pp. 1889–1897, 2015.

[8] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive environments,” in *NIPS*, pp. 6379–6390, 2017.

[9] T. Bansal, J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch, “Emergent complexity via multi-agent competition,” *arXiv:1710.03748*, 2017.

[10] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *ICLR*, 2015.

[11] D. Balduzzi, M. Garnelo, Y. Bachrach, W. M. Czarnecki, J. Perolat, M. Jaderberg, and T. Graepel, “Open-ended learning in symmetric zero-sum games,” *arXiv:1901.08106*, 2019.