OPPONENT MODELING IN DEEP REINFORCEMENT LEARNING

HE H., BOYD-GRABER J., KWOK K., DAUME H. (2016)

11/8/2021

Presented By: Muhammad Hassan

CS 885: Reinforcement Learning (Fall 2021)

MMath (thesis) Computer Science | **David R. Cheriton School of Computer Science**

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

OUTLINE

- *Terminology and Notation*
- Introduction
- Motivation
- Literature Review
- Approach
- Evaluation
- Conclusion
- *References*

TERMINOLOGY AND NOTATION

- **Primary Agent / Agent:** *The agent that is being trained / optimized*
- **Secondary Agents / Other Agents:** *Agents that the primary agent is competing against/collaborating with. (ally/opponent)*
- **Ø(x):** *denotes the features of x*
- **h(x):** *denotes hidden representation of x*
- **A(x):** *denotes output layer of network*
- **o:** *denotes opponent actions/behavior*

INTRODUCTION

- Opponent Modeling is important in all multi-agent environments (collaborative/competitive). Examples: multi-player games, negotiations, self-driving cars
- Every secondary agent's actions affect the state of the environment, and preclude/advance opportunities for the primary agent
- ISSUES: What variables to consider in opponent modeling? How to use the opponent model in evaluating actions for primary agent?
- **SOLUTION:** A general opponent modeling framework that models uncertainty of opponent policy, and learns its own policy *jointly*

MOTIVATION

- A multi-agent environment is one that has multiple agents interacting with each other either collaborating on a task or competing
- Learning optimal policies for such environments is challenging, because every secondary agent's actions change the environment, and as a result, the reward distribution is nonstationary, and so the policy must be too

$$Q^{\pi|\pi^o}(s_t, a_t) = \sum_{o_t} \pi_t^o(o_t|s_t) \sum_{s_{t+1}} \mathcal{T}(s_t, a_t, o_t, s_{t+1}) \\ \left[\mathcal{R}(s_t, a_t, o_t, s_{t+1}) + \gamma \mathbb{E}_{a_{t+1}} \left[Q^{\pi|\pi^o}(s_{t+1}, a_{t+1})\right]\right]. \quad (1)$$

- Two main categories of past work: Explicit vs Implicit Opponent Modeling
- Our approach is based on past works in implicit opponent modelling

LITERATURE REVIEW

Explicit Opponent Modeling		Implicit Opponent Modeling	
• Uther & Veloso (2003) • Ganzfried & Sandholm (2011) • Billings et al. (1998) • Richards & Amir (2007) • Schadd et al. (2007) • Southey et al. (2005)	• Build separate models (decision trees, Bayesian models etc) to learn opponent policy characteristics and use them in decision-making • Domain-specific (e.g Poker, Scrabble) - *Need lots of data* - *Difficult to integrate with primary policy learning*	• Rubin & Watson (2011) • Bard et al. (2013)	• Create an array of strategies offline based on domain knowledge, then use a multi-arm bandit online to select a strategy - *Separate training phases, one to learn strategies, second to learn strategy selector*
• Davidson (1999) • Lockett et al. (2007) • Foerster et al. (2016) • Tampu et al. (2015)	• Use neural networks to learn opponent policy characteristics in supervised fashion • Foerster et al. (2016) trained collaborating DRQN agents with shared parameters • Tampu et al. (2015) applied two DQN agents in a multi-agent setting but the agents were fully observable to each other - *For competing agents, the opponent policy space is unknown* - *Supervised learning alone does not work well in complex environments*		

APPROACH

- **Deep Reinforcement Opponent Network (DRON) :**
- Q(s,a) = *A(h(h*(state) + *h*(opponent actions)))
- 2 ways considered to + the hidden representations: *Concatenation* and *Mixture-of-Experts*
- Additional supervised guidance can be added via *multitasking*

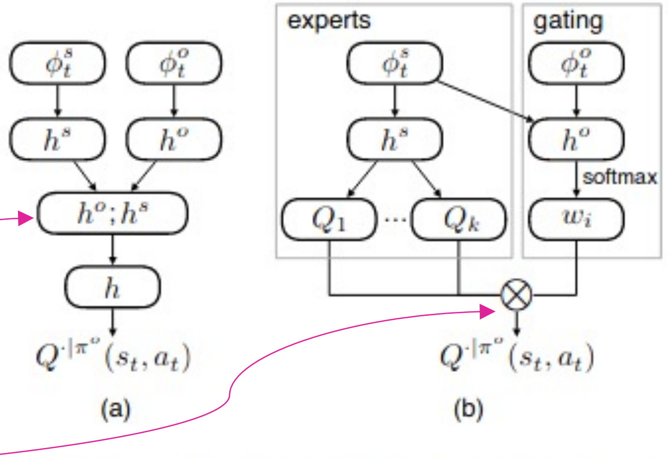


Figure 1. Diagram of the DRON architecture. (a) DRON-concat: opponent representation is concatenated with the state representation. (b) DRON-MoE: Q-values predicted by K experts are combined linearly by weights from the gating network.

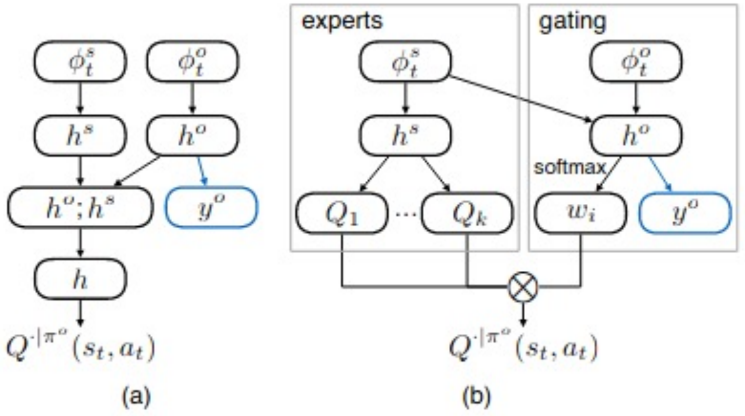


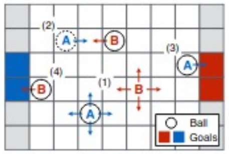
Figure 2. Diagram of the DRON with multitasking. The blue part shows that the supervision signal from the opponent affects the Q-learning network by changing the opponent features.

APPROACH DETAILS

	DRON-Concat	DRON-MoE	DRON w/ supervision
Function	• Learns the distribution of Q-values conditional upon opponent behavior $Q(s,a\mid o)$	• Learns the distribution of Q-values conditional upon opponent behavior $\Sigma\ (\Pi(o\mid s)*Q(s,a\mid o))\forall o$	• Combines explicit modeling into the approach, through a supervision signal that guides opponent model training;
Advantages	• Simple and efficient	• Represents a stronger prior	• Additional signals for opponent model.
Disadvantages	• Ignores environment-opponent interaction • Opponent representation needs to be more distinct and discriminative; stronger prior needed	• Necessarily complex and costly	• Signals may conflict with indirect signal coming from Q-value (more so in case of DRON-MoE)
	• Both prone to errors due to insufficient data and Q-value estimation since opponent model updated through Q-values		
Comparison with past work	• Incorporation of h(opponent) into Q-value model, removes the need to learn opponent separately • Policy and opponent model learnt jointly, so no integration issues • Opponent model updated indirectly through Q-values, so no need for separate opponent training or large amounts of data • Allows for incorporation of explicit opponent modeling techniques through supervision signals		

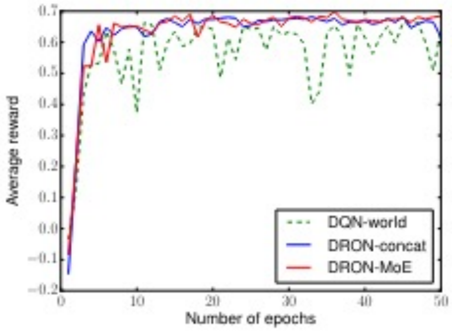
EVALUATION # 1

	Experiment # 1: Soccer
Description	2-player 6x9 grid soccer
Opponent	Stochastic opponent-2 modes(offensive/defensive). Features: frequencies of observed opponent moves, most recent move and action, frequency of losing the ball to opponent
Supervision	Current opponent action, opponent mode
Baseline	DQN-world (treats opponents as environment)
Results	• Performs much better than baseline • More stable learning (low variance) • DRON adjusts well against both modes of opponents • SURPRISE: No significant difference by varying number of experts in DRON-MoE • SURPRISE: Adding supervision makes DRON-MoE results poorer (reason: conflict between Q-value and supervision signal)

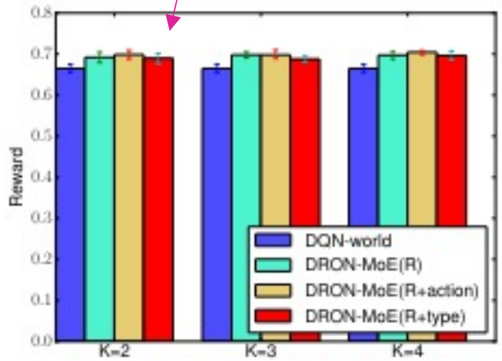


	Defensive	Offensive
w/ ball	Avoid opponent	Advance to goal
w/o ball	Defend goal	Intercept the ball

Figure 3. Left: Illustration of the soccer game. *Right:* Strategies of the hand-crafted rule-based agent.



Model	Basic	Multitask	
		+action	+type
	Max R		
DRON-concat	0.682	0.695*	0.690*
DRON-MOE	**0.699***	0.697*	0.686*
DQN-world	0.664	-	-
	Mean R		
DRON-concat	0.660	0.672	0.669
DRON-MOE	0.675	0.664	0.672
DQN-world	0.616	-	-

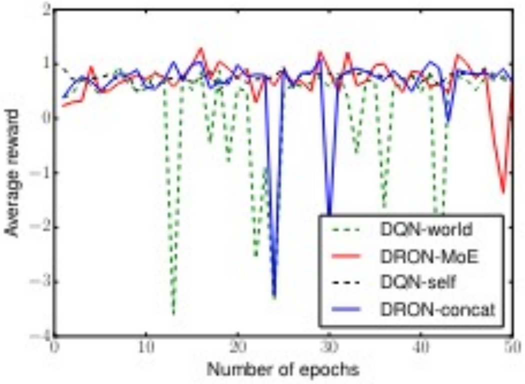


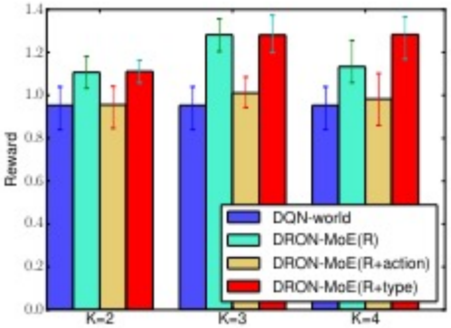
	DQN O only	DQN D only	DQN -world	DRON -concat	DRON -MOE
O	**0.897**	-0.272	0.811	0.875	0.870
D	0.480	**0.504**	0.498	0.493	0.486

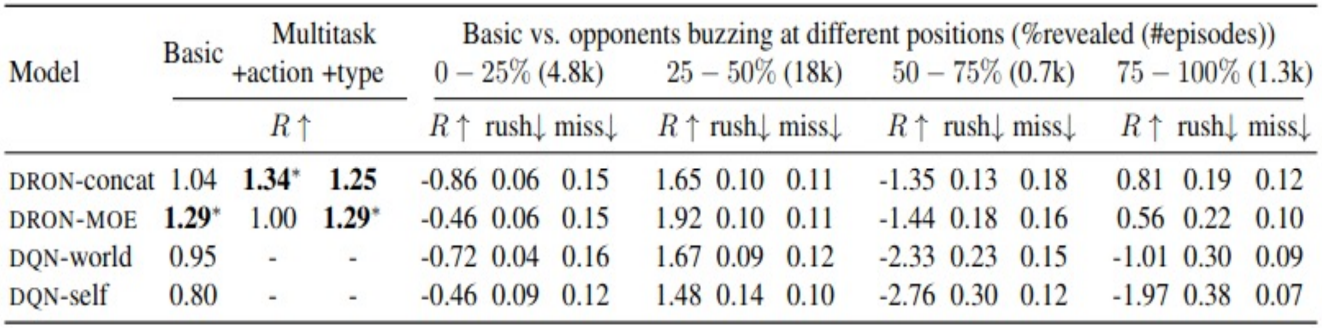
UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

EVALUATION # 2

	Experiment # 2: Quiz Bowl
Description	2-player buzz-and-answer game
Player	Content model (RNN to read questions and give distribution over answers), Buzzing model (when to buzz)
Opponent	Stochastic opponent-4 modes(<= 25, 50, 75, 100 % question heard). Features: # of questions answered, average buzz position, error rate
Supervision	Opponent buzz pattern, opponent type
Baseline	DQN-world (treats opponents as environment), DQN-self (answer only when sure)
Results	- Performance much better than baselines - More stable learning (low variance) - DRON adjusts well against all 4 modes of opponents - Adding supervision does not improve DRON-MoE but improves DRON-Concat significantly - <u>SURPRISE:</u> action supervision is useless, but type supervision yields competent results (especially with K=4)







Model	Basic	Multitask		Basic vs. opponents buzzing at different positions (%revealed (#episodes))											
		+action	+type	0 – 25% (4.8k)			25 – 50% (18k)			50 – 75% (0.7k)			75 – 100% (1.3k)		
	$R\uparrow$			$R\uparrow$	rush↓	miss↓	$R\uparrow$	rush↓	miss↓	$R\uparrow$	rush↓	miss↓	$R\uparrow$	rush↓	miss↓
DRON-concat	1.04	**1.34***	**1.25**	-0.86	0.06	0.15	1.65	0.10	0.11	-1.35	0.13	0.18	0.81	0.19	0.12
DRON-MOE	**1.29***	1.00	**1.29***	-0.46	0.06	0.15	1.92	0.10	0.11	-1.44	0.18	0.16	0.56	0.22	0.10
DQN-world	0.95	-	-	-0.72	0.04	0.16	1.67	0.09	0.12	-2.33	0.23	0.15	-1.01	0.30	0.09
DQN-self	0.80	-	-	-0.46	0.09	0.12	1.48	0.14	0.10	-2.76	0.30	0.12	-1.97	0.38	0.07

CONCLUSION

- This work presents DRON, DQN-based approach that helps model the uncertainty of opponent behavior (implicitly) and learn a non-stationary policy jointly
- Joint modeling removes the need for lots of data and domain-specific opponent modeling, avoids integration issues, while allowing supervision to be incorporated as well if desired
- Extends the power of DQN to multi-agent competitive environments with unknown secondary agents
- Evaluation of DRON in two experiments show superior results over DQN baseline(s).
- The broader implications of the work are that generalized opponent modeling is tractable and can deliver excellent results online. This can be extended to other domains.
- Potential Future Work: learning opponent features automatically, exploration v/s exploitation in multi-agent environments, hierarchical reinforcement learning with deep MoEs

REFERENCES

- He, He, Boyd-Graber, Jordan, Kwok, Kevin and Daume III, Hal. Opponent Modeling in Deep Reinforcement Learning. In *International Conference on Machine Learning*, 2016.
- Bard, Nolan, Johanson, Michael, Burch, Neil, and Bowling, Michael. Online Implicit Agent Modelling. In *International Conference on Autonomous Agents and Multiagent Systems*, 2013
- Foerster, Jakob, Assael, Yannis, Freitas, Nando, and Whiteson, Shimon. Learning to Communicate to Solve Riddles with Deep Distributed Recurrent Q-Networks. In *arXiv*, 2016.
- Ganzfried, Sam and Sandholm, Tuomas. Game Theory-Based Opponent Modeling in Large Imperfect-Information Games. In *International Conference on Autonomous Agents and Multiagent Systems,* 2011.

UNIVERSITY OF WATERLOO

FACULTY OF MATHEMATICS

Thank You ☺ Any Questions?