Multi-agent Reinforcement Learning for Networked System Control Report

Ivana Louis
CSCI 49000: Deep Learning
Indiana University Purdue University Indianapolis
Indianapolis, USA
ilouis@iu.edu

This paper considers the new method proposed by a published paper of Multi-agent reinforcement learning in a networked system control. Each must learn a decentralized control policy, whether it be communicative or non-communicative. These learnings are based on local observations and messages from connected neighbors. The scientists used both spatiotemporal MDP and a newly proposed method known as NeurComm. These protocols were tested using two environments, Adaptive Traffic System Control and Cooperative Adaptive Cruise Control. Each environment also had two scenarios five by five grid, and Monaco net for ASTC and Catch up Slow Down for CACC.

Keywords— MARL, NMARL, NeurComm, Multi-Agent Reinforcement Learning

I. Introduction

In this paper the authors wanted to create a new method that could help improve results for Multi-agent Reinforcement Learning (MARL) in a networked system. The problem began with MARL in a game or simulation environment. This version of MARL didn't have spatial structure and it assumed global observation and communication for each agent. They then turned to MARL in a networked system. This version uses a conjoined infrastructure that is widely distributed making it difficult to collect and consume global information. After research, they figured out that using MARL in a networked system, but switching to the use of decentralized rewards rather than observations gave clear, static spatial structure and led to local observations and communication. Their aim was to develop a new approach under those new settings since it could benefit many applications, such as traffic control and mobile network control.

II. Method

A. Goals

The writers new goal was to create a stable and scalable MARL for networked system control. To achieve this goal they leveraged the network structure in both the Markov Decision Process (MDP) Formulation and communication protocol design.

B. Spatiotemporal MDP

Eq. (1) assumes that the Markovian property holds both temporally and spatially. This means that the neighboring states and policies are the only thing that dictates the next local state. This assumption is allowed in many networked control systems such as traffic and wireless networks. The impact of each agent is spread over the entire system through controlled flows, or chained local transitions.

$$p_i\left(s_{i,t+1}|sv_{i,t},a_{i,t}\right)=\sum_{a_{\mathcal{N}_i,t}\in\mathcal{A}_{\mathcal{N}_i}}\prod_{j\in\mathcal{N}_i}\pi_j\left(a_{j,t}|\tilde{s}_{j,t}\right)\cdot p\left(s_{i,t+1}|sv_{i,t},a_{i,t},a_{\mathcal{N}_i,t}\right),$$

Eq. (1)

$$R_{i,t}^{\pi}=\sum_{\tau=t}^{T}\gamma^{\tau-t}\left(\sum_{j\in\mathcal{V}}\alpha^{d_{ij}}r_{j,t}\right)$$

Eq. (2)

Since in a networked system control each agent is connected to a limited number of neighbors, spatiotemporal MDP is decentralized during model execution, and it naturally extends properties of MDP. A spatiotemporally discounted return is introduced in Eq. (2), to reduce the learning difficulty of spatiotemporal MDP, a spatiotemporally discounted return is introduced in Eq. (2).

C. Neural Communication

For efficient and adaptive information sharing, the researchers proposed a new communication protocol called NeurComm. In Eq. (3), each agent is utilizing delayed global information. This allows the agent to learn its belief, and the message that optimizes the control performance of all other agents.

$$h_{i,t}=g_{\nu_i}\left(h_{i,t-1},e_{\lambda_i^s}\left(sv_{i,t}\right),e_{\lambda_i^p}\left(\pi_{\mathcal{N}_i,t-1}\right),e_{\lambda_i^h}\left(h_{\mathcal{N}_i,t-1}\right)\right),$$

Eq. (3)

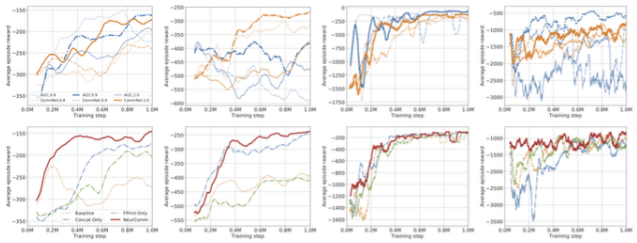
Identify applicable funding agency here. If none, delete this text box.

III. Experiment

The researchers evaluated their aggregates in two networked environments. In the Adaptive Traffic Signal Control (ATSC) they used a five by five traffic grid and Monaco traffic net. The second environment was Cooperative Adaptive Cruise Control (CACC) with Catch-Up and Slow-Down. They tested 6 policies, 3 non-communicative and 3 communicative. IA2c, Fprint, and ConseNet are the non-communicative policies, and Neurcom, commNet, and Dial are the communicative policies.

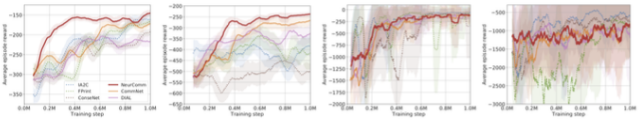
Table 1: Best spatial discount factors α^* across NMARL scenarios.

Scenario Name	NeurComm	CommNet	DIAL	IA2C	FPrint	ConseNet
ATSC Grid	1.0	1.0	1.0	0.9	0.95	0.9
ATSC Monaco	1.0	0.9	0.9	0.9	0.9	0.9
CACC Catch-up	1.0	1.0	1.0	1.0	1.0	1.0
CACC Slow-down	1.0	1.0	1.0	0.8	0.9	0.8



(a) ATSC Grid. (b) ATSC Monaco. (c) CACC Catch-up. (d) CACC Slow-down.

Fig. 1. Results from the researchers study. Note that CACC is giving a large penalty whenever a collision happens.

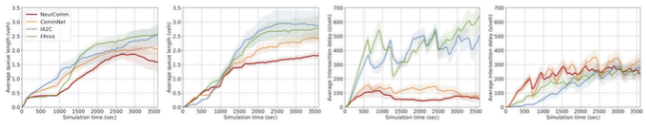


(a) ATSC Grid. (b) ATSC Monaco. (c) CACC Catch-up. (d) CACC Slow-down.

Fig. 2. Training performance comparison after tuned spatial discount factors

Table 2: Execution performance comparison over trained MARL policies. Best values are in bold.

Scenario Name	NeurComm	CommNet	DIAL	IA2C	FPrint	ConseNet
ATSC Grid	**-136.1**	-165.1	-214.4	-160.2	-155.9	-187.5
ATSC Monaco	**-226.3**	-263.0	-339.4	-369.7	-359.4	-528.9
CACC Catch-up	-94.6	-95.6	-246.4	-261.7	**-57.8**	-419.7
CACC Slow-down	-934.7	-950.8	-1112	-2209	**-697.9**	-1038



(a) Average queue length in ATSC Grid. (b) Average queue length in ATSC Monaco. (c) Average intersection delay in ATSC Grid. (d) Average intersection delay in ATSC Monaco.

Fig. 3. Execution performance comparison among top policies in ATSC scenarios, measured as average queue length and average intersection delay over time.

With new updates to SUMO the paper is ran based off of an outdated version of SUMO. The program should be rerun based on the newer version of SUMO, version 1.2.0.

IV. Results

The below figure visualizes spatial discount factors of controllers across the different networked MARL ATSC scenarios. Just like the results given by the paper scenarios like ATSC Monaco, a lower α is preferred by almost all policies (except NeurComm). This demonstrates that α is an effective way to enhance MARL performance in general, especially for challenging tasks like ATSC Monaco. From another view point, α serves as an informative indicator on problem difficulty and algorithm coordination level.

ATSC Grid and Monaco

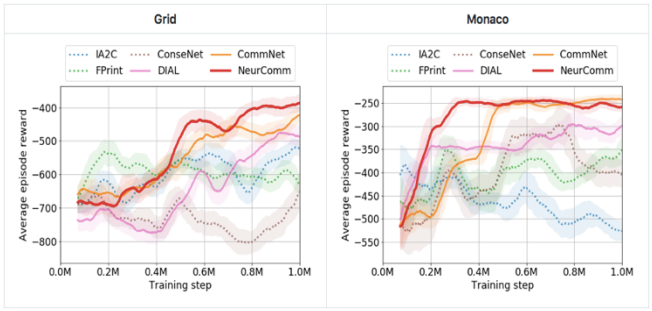


Fig. 4. Visualizes spatial discount factors of controllers across different Networked Multi-agent Reinforement Learning Scenarios

V. Conclusion

The results proved that the code and the given results were reproducible. This allows us to use the NeurComm in future networked MARL scenarios. This could be used in studies of the power grid, wireless mobile networks, and even autonomous cars. However, this method is very difficult and complicated. I wonder to myself if there are easier methods to go about this. Not only that, but how would this benefit the scenarios more than other methods. The NeurComm does not always outperform the other policies, which doesn't make it superior all around.