Multi-agent System and Application

Learning to Communication

Xiangfeng Wang

Multi-Agent Artificial Intelligence Laboratory,
School of Computer Science and Technology,
East China Normal University

2021 Spring

Joint work with Junjie Sheng, Wenhao Li and Yun Hua

Table of Contents

Table of Contents

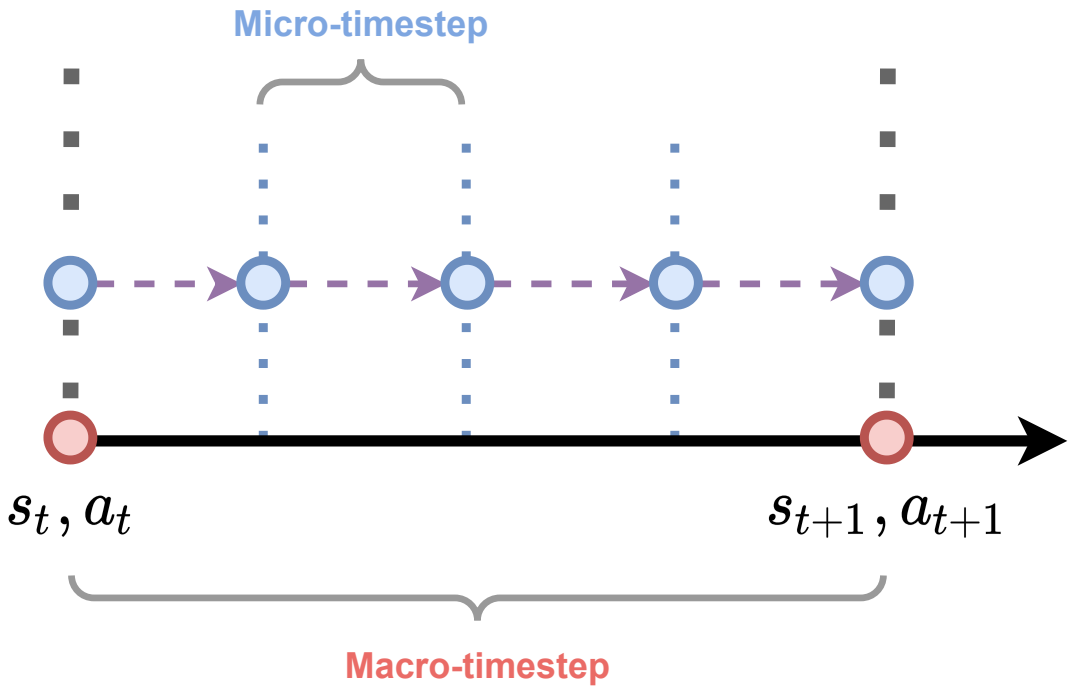
Introduction: Key Issues of "Learning to Communication"

To design a learning to communication algorithm, for each agent in a multi-agent system, the following three key issues need to be solved.

- **When**: Determine whether message needs to be sent to the other agents based on the current environmental state or the current local observation.
- **Who**: Determine the recipient of the message, but the recipient may not necessarily receive the message.
- **What**: Determine the content of the message to be sent, the sender who needs to receive the message, and the post-processing method of the received message.

Introduction: The Key Concepts

- Macro- and micro-timestep.



Introduction: Typical Algorithms

Each algorithm needs to solve the three key issues mentioned above, but different algorithms have different concerns. We will introduce three typical algorithms for each issue separately, including

- **What:** Reinforced & Differentiable Inter-Agent Learning (Foerster et al., **NeurIPS, 2016**), Learning Communication with Backpropagation (Sainbayar et al., **NeurIPS, 2016**)
- **When:** Learning Attentional Communication (Jiang et al., **NeurIPS, 2018**), Learning to Schedule Communication (Daewoo et al., **ICLR, 2019**)
- **Who:** Learning Individually Inferred Communication (Ding et al., **NeurIPS, 2020**)

Table of Contents

Learning Communication with Backpropagation

- Focuses on the generation of effective communication messages by end-to-end learning.
- Follows the *centralized training* but *decentralized execution* framework. The agent needs to rely on local observations and received messages to make decisions in the execution phase.
- All agents only have partial observability and share the goal of maximizing the same discounted sum of rewards R_t.

Learning Communication with Backpropagation

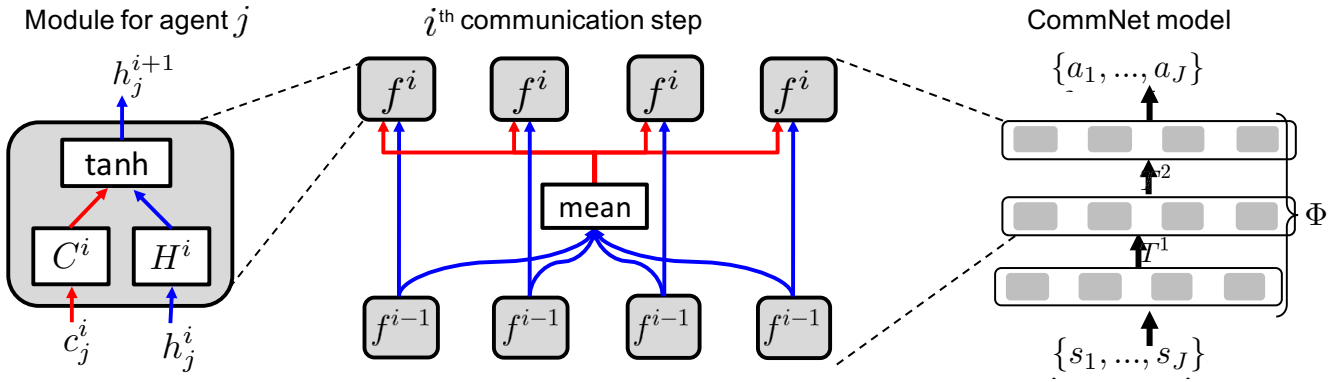


Figure 4: An overview of CommNet model. Left: view of module f^i for a single agent j. Note that the parameters are shared across all agents. Middle: a single communication step, where each agents modules propagate their internal state h, as well as broadcasting a communication vector c on a common channel (shown in red). Right: full model, showing input states s for each agent, two communication steps and the output actions for each agent.

Learning Communication with Backpropagation

- Each agent sends a message to all other agents at two consecutive micro-timesteps per macro-timestep.
- Each message is a continuous vector output by a feed-forward neural network.
- CommNet use policy gradient with a state specific baseline for delivering a gradient to the model. The baseline is a scalar function of the states $b(s, \theta)$, computed via an extra head on the model producing the action probabilities.
- Beside maximizing the expected reward with policy gradient, the models are also trained to minimize the distance between the baseline value and actual reward. Thus after finishing an episode, we update the model parameters θ by

$$\Delta\theta = \sum_{t=1}^{T} \left[\frac{\partial \log p(a(t)|s(t),\theta)}{\partial\theta} \left(\sum_{i=t}^{T} r(i) - b(s(t),\theta) \right) - \alpha \frac{\partial}{\partial\theta} \left(\sum_{i=t}^{T} r(i) - b(s(t),\theta) \right)^2 \right]$$

Learning Communication with Backpropagation

- **Local Connectivity:** An alternative to the broadcast framework described above is to allow agents to communicate to others within a certain range. As the agents move, enter and exit the environment, CommNet has a natural interpretation as a dynamic graph. In this setting, the message exchanging being equivalent to belief propagation.
- **Skip Connections:** For some tasks, it is useful to have the input encoding present as an input for communication steps beyond the first layer.
- **Temporal Recurrence:** CommNet also can replace the feed-forward network with recurrent neural network to alleviate the partial observability.

Learning Communication with Backpropagation

Lever Pulling Task

- We start with a very simple game that requires the agents to communicate in order to win.
- This consists of m levers and a pool of N agents.
- At each round, m agents are drawn at random from the total pool of N agents and they must each choose a lever to pull, simultaneously with the other $m-1$ agents, after which the round ends.
- The goal is for each of them to pull a different lever.
- Correspondingly, all agents receive reward proportional to the number of distinct levers pulled.
- Each agent can see its own identity, and nothing else, thus $s_j = j$.