Learning to Communicate with Multi-Agents Reinforcement Learning

Wenhao Yang

School of Mathematical Sciences
Peking University

April 15, 2017

Outline

1. Background
   - Deep Q-Networks
   - Independent DQN
   - Deep recurrent Q-Networks
2. Models
   - Reinforced Inter-Agent Learning(RIAL)
   - Differentiable Inter-Agent Learning(DIAL)
3. Experiments
   - Switch Riddle
   - Multi-Step MNIST
   - Colour-Digit MNIST
4. Reference

Deep Q-Networks

- Aim: Single Agent
- Notation: state: s_t, action: u_t, reward: r_t, discout: γ, cumulative reward: $R_t = \sum_{k=0}^{\infty} r_{t+k}\gamma^k$
- Q-value function: $Q^\pi(s,u) = E[R_t | s_t = s, u_t = u]$
- Bellman equation: $Q^*(s,u) = E_{s'}[r + \gamma \max u' Q(s',u') | s,u]$
- Loss function:
  $L(\theta) = E[(r + \gamma \max_{u'} Q(s',u',\theta^-) - Q(s,u,\theta))^2]$

Independent DQN

- Aim: Multi-Agents
- Settings:
  all observe global state
  Maximize team reward
  Each agents Q-value function: $Q^a(s, u^a, \theta^a)$

Deep recurrent Q-Networks

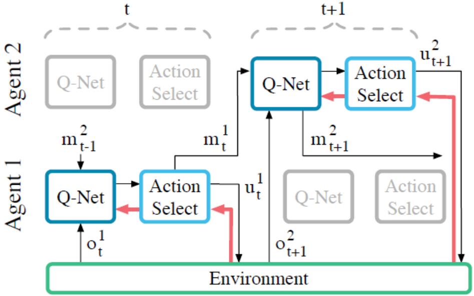
- Only partial state o_t is observed, global state s_t is hidden
- Q(s,u) can't be approximated as s is not known
- Solution: Approximate $Q(o_t, h_{t-1}, u)$ with recurrent network, where h_t represents the hidden state of the network

Setting

- Consider Multi-Agents and partially observation
- Maximize team reward
- Each agent receives o_t^a correlated to s_t
- In each time-step, agents select an environment action $u \in U$ and a communication action $m \in M$
- Communication action is observed by other agents but has no direct impact on the environment or reward
- Parameters sharing

Reinforced Inter-Agent Learning(RIAL)

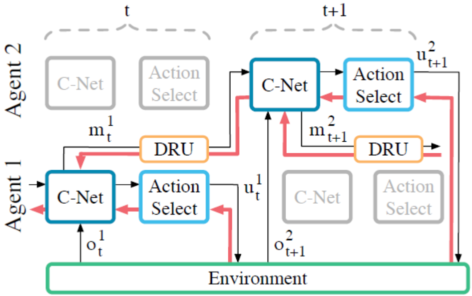
- Deep Recurrent Q-network and independent Q-learning
- Q-value: $Q^a(o^a_t, m^{a'}_{t-1}, h^a_{t-1}, u^a)$
- Structure:



(a) RIAL - RL based communication

Differentiable Inter-Agent Learning(DIAL)

- Feedback about communication actions
- Structure:



(b) DIAL - Differentiable communication

Model Architecture

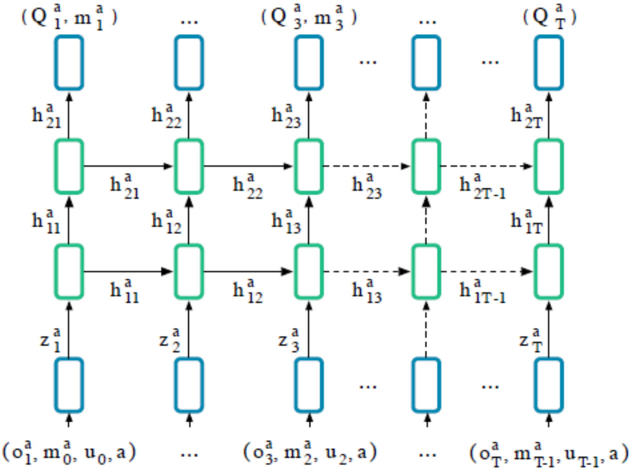


Figure 2: DIAL architecture.

Model Architecture

- Input: $(o_t^a, m_{t-1}^{a'}, u_{t-1}^a, a)$
- Embedding:
  $z_t^a = f_1(o_t^a) + f_2(m_{t-1}) + f_3(u_{t-1}^a) + f_4(a)$
  where f_1 is a task-specific network, f_2 is a 1-layer MLP and f_3 and f_4 are lookup tables
- 2-layer RNN with GRUs
- Output: Q_t^a, m_t^a

Algorithm(DIAL)

Initialise θ_1 and θ_1^-
for each episode e **do**
  s_1 = initial state, $t = 0$, $h_0^a = \mathbf{0}$ for each agent a
  while $s_t \neq$ terminal **and** $t < T$ **do**
    $t = t + 1$
    for each agent a **do**
      Get messages $\hat{m}_{t-1}^{a'}$ of previous time-steps from agents m' and evaluate C-Net:
      $Q(\cdot), m_t^a = \text{C-Net}\left(o_t^a, \hat{m}_{t-1}^{a'}, h_{t-1}^a, u_{t-1}^a, a; \theta_i\right)$
      With probability ϵ pick random u_t^a, else $u_t^a = \max_a Q\left(o_t^a, \hat{m}_{t-1}^{a'}, h_{t-1}^a, u_{t-1}^a, a, u; \theta_i\right)$
      Set message $\hat{m}_t^a = \text{DRU}(m)$, where $\text{DRU}(m) = \begin{cases} \text{Logistic}(\mathcal{N}(m_t^a, \sigma)), \text{ if training, else} \\ \mathbb{1}\{m_t^a > 0\} \end{cases}$
    Get reward r_t and next state s_{t+1}
  Reset gradients $\nabla\theta = 0$
  for $t = T$ **to** $1, -1$ **do**
    for each agent a **do**
      $y_t^a = \begin{cases} r_t, \text{ if } s_t \text{ terminal, else} \\ r_t + \gamma \max_u Q\left(o_{t+1}^a, \hat{m}_t^{a'}, h_t^a, u_t^a, a, u; \theta_i^-\right) \end{cases}$
      Accumulate gradients for action:
      $\Delta Q_t^a = y_t^a - Q\left(o_j^a, h_{t-1}^a, \hat{m}_{t-1}^{a'}, u_{t-1}^a, a, u_t^a; \theta_i\right)$
      $\nabla\theta = \nabla\theta + \frac{\partial}{\partial\theta}(\Delta Q_t^a)^2$
      Update gradient chain for differentiable communication:
      $\mu_j^a = \mathbb{1}\{t < T - 1\} \sum_{m' \neq m} \frac{\partial}{\partial \hat{m}_t^a}\left(\Delta Q_{t+1}^{a'}\right)^2 + \mu_{t+1}^{a'} \frac{\partial \hat{m}_{t+1}^{a'}}{\partial \hat{m}_t^a}$
      Accumulate gradients for differentiable communication:
      $\nabla\theta = \nabla\theta + \mu_t^a \frac{\partial}{\partial m_t^a} \text{DRU}(m_t^a) \frac{\partial m_t^a}{\partial \theta}$

  $\theta_{i+1} = \theta_i + \alpha \nabla\theta$
  Every C steps reset $\theta_i^- = \theta_i$

Experiment: Switch Riddle

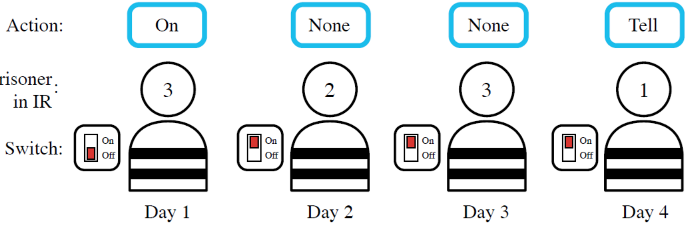
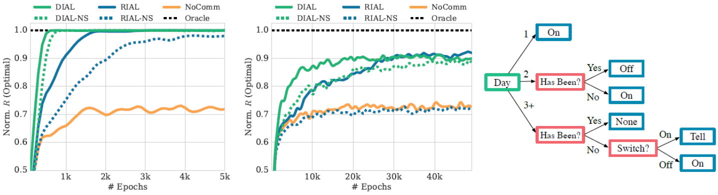


Figure 3: *Switch:* Every day one prisoner gets sent to the interrogation room where he sees the switch and chooses from “On”, “Off”, “Tell” and “None”.

Experiment: Switch Riddle

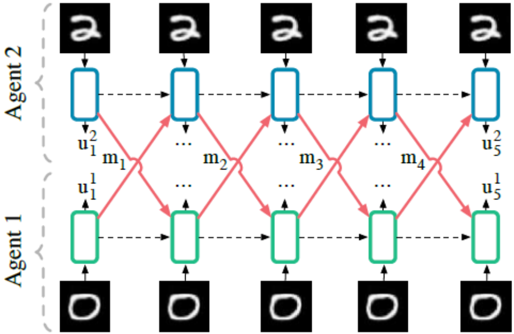
- $m_t^a, o_t^a \in \{0, 1\}$
- $u_t^a \in \{None, Tell\}$
- $r_t \in \{-1, 0, 1\}$
- Results:



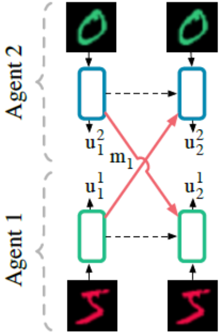
(a) Evaluation of $n = 3$ (b) Evaluation of $n = 4$ (c) Protocol of $n = 3$

Figure 4: *Switch:* (a-b) Performance of DIAL and RIAL, with and without (-NS) parameter sharing, and NoComm-baseline, for $n = 3$ and $n = 4$ agents. (c) The decision tree extracted for $n = 3$ to interpret the communication protocol discovered by DIAL.

Experiment: Multi-Step MNIST

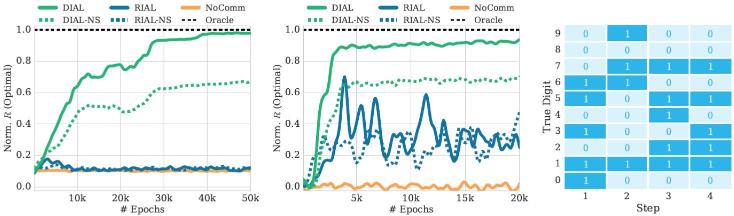


Experiment: Colour-Digit MNIST



Not descriped clearly in paper

MNIST results



(a) Evaluation of Multi-Step (b) Evaluation of Colour-Digit (c) Protocol of Multi-Step

Figure 6: *MNIST Games:* (a,b) Performance of DIAL and RIAL, with and without (-NS) parameter sharing, and NoComm, for both MNIST games. (c) Extracted coding scheme for multi-step MNIST.

Reference

Reference:

[1] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, Shimon Whiteson; Learning to Communicate with Deep Multi-Agent Reinforcement Learning