CS4400
DEEP REINFORCEMENT LEARNING

Lecture 11: Advanced MARL

Wendelin Böhmer

<j.w.bohmer@tudelft.nl>

TU Delft

11th of January 2024

Content of this lecture

11.1

Advanced MARL
Value factorization

core concept: Value factorization

- Cooperative tasks allow centralized objectives
  - with decentralized decision policies

- Decentralizable value factorization $Q_\theta(\boldsymbol{\tau}_t, \boldsymbol{a}) \;:=\; \sum\limits_{i=1}^{N} q_\theta^i(\tau_t^i, a^i)$
  - called *value decomposition networks* (VDN)
  - *utilities* q_θ^i do not fulfill Bellman equation

$$\boldsymbol{a}_t^* \;=\; \arg\max_{\boldsymbol{a}\in\mathcal{A}} Q_\theta(\boldsymbol{\tau}_t, \boldsymbol{a}) \;=\; \Big\{ \arg\max_{a^i\in\mathcal{A}^i} q_\theta^i(\tau_t^i, a^i) \Big\}_{i=1}^{N}$$

- Optimized with centralized (double) DQN loss

$$\mathcal{L}_{[\theta]}^{\text{VDN}} \;:=\; \mathbb{E}_{\mathcal{D}}\Big[\Big(r_t + \gamma Q_{\theta'}\big(\boldsymbol{\tau}_{t+1}, \arg\max_{\boldsymbol{a}\in\mathcal{A}} Q_\theta(\boldsymbol{\tau}_{t+1}, \boldsymbol{a})\big) - Q_\theta(\boldsymbol{\tau}_t, \boldsymbol{a}_t)\Big)^2\Big]$$

VDN by Sunehag et al. (2018)

11.1 Monotonic value factorization

- VDN factorization restricts function class of Q_θ
- Any monotonic value mixture is decentralizable: $\frac{\partial Q_\theta(\boldsymbol{\tau}_t, \boldsymbol{a})}{\partial q^i_\theta(\tau^i_t, a^i)} \geq 0$
  - e.g. VDN is monotonic $\frac{\partial \sum_{j=1}^N q^j_\theta(\tau^j_t, a^j)}{\partial q^i_\theta(\tau^i_t, a^i)} = 1$

QMIX by Rashid et al. (2018, 2020b)

11.1 Monotonic value factorization

- VDN factorization restricts function class of Q_θ
- Any monotonic value mixture is decentralizable: $\frac{\partial Q_\theta(\boldsymbol{\tau}_t, \boldsymbol{a})}{\partial q^i_\theta(\tau^i_t, a^i)} \geq 0$
  - e.g. VDN is monotonic $\frac{\partial \sum_{j=1}^N q^j_\theta(\tau^j_t, a^j)}{\partial q^i_\theta(\tau^i_t, a^i)} = 1$
- QMIX: learn centralized monotonic mixture network f_ϕ
  - f_ϕ not required during execution
  - f_ϕ yields different mixing for different states s_t

$$Q_{\theta\phi}(s_t, \boldsymbol{\tau}_t, \boldsymbol{a}) \;:=\; f_\phi\Big(s_t, q^1_\theta(\tau^1_t, a^1), \ldots, q^N_\theta(\tau^N_t, a^N)\Big) \;=:\; f_\phi\Big(s_t, \boldsymbol{q}_\theta(\boldsymbol{\tau}_t, \boldsymbol{a})\Big)$$

- *Hyper-network* f_ϕ with non-negative weights, e.g.:

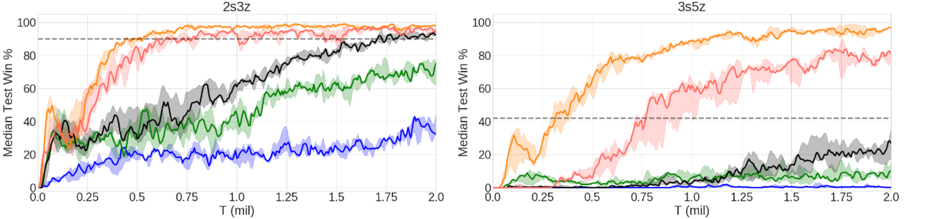
$$f_\phi(\boldsymbol{s}, \boldsymbol{q}) = \boldsymbol{w}^{1\top}_{(\boldsymbol{s})} \sigma(\mathbf{W}^2_{(\boldsymbol{s})} \boldsymbol{q} + \boldsymbol{w}^3_{(\boldsymbol{s})}) + w^4_{(\boldsymbol{s})}\,, \qquad \boldsymbol{w}^k_{(\boldsymbol{s})} = \Big| \underbrace{\mathbf{A}^k_2}\, \bar\sigma(\underbrace{\mathbf{A}^k_1} \boldsymbol{s} + \underbrace{\boldsymbol{b}^k_1}) + \underbrace{\boldsymbol{b}^k_2} \Big|$$

hyper-net parameters ϕ

QMIX by Rashid et al. (2018, 2020b)

Comparison: StarCraft II micro-management

- Play against build-in AI
- Partial observability
- Varying units available
    - here: stalkers and zealots



QMIX VDN QTRAN IQL COMA

StarCraft II multi-agent challenge (SMAC, Samvelyan et al., 2019); QTRAN (Son et al., 2019); results from (Rashid et al., 2020b)

11.1 Summary

- Centralized value functions can be *factorized*
- Factorization must be monotonic in per-agent utilities
- Maximizing utilities independently maximizes joint value
- QMIX extends VDN by using non-negative hyper-networks
- QTRAN defines decentrability as constraints, but is instable

Learning Objectives

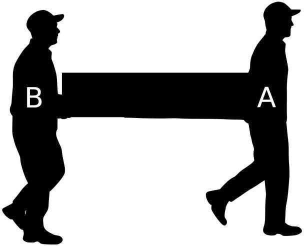
LO11.1: Explain monotonic value factorization in VDN and QMIX

11.2

Advanced MARL
Relative overgeneralization

core concept: Relative overgeneralization

- Why is independent learning or value factorization problematic?
  - e.g. single-state IQL $q^i(a^i; \boldsymbol{\pi}^{-i}) := \mathbb{E}\big[Q^{\boldsymbol{\pi}}(\boldsymbol{a}) \,|\, \boldsymbol{a}^{-i} \sim \boldsymbol{\pi}^{-i}(\cdot)\big]$
- What are the IQL values if all agents explore randomly?
  - team needs to transport a fragile box (reward $+1$)
  - the box falls (punishment -1) if A is too far from B
  - will a team of IQL agents learn an optimal policy?

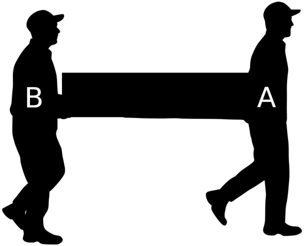


		A				
		$\leftarrow$	$\varnothing$	$\rightarrow$		
B	$\leftarrow$	0	-1	-1		
	$\varnothing$	0	0	-1		
	$\rightarrow$	0	0	$+1$		
q^A_{explore}			?			

image source: pxhere.com

core concept: Relative overgeneralization

- Why is independent learning or value factorization problematic?
  - e.g. single-state IQL $q^i(a^i; \boldsymbol{\pi}^{-i}) := \mathbb{E}\big[Q^{\boldsymbol{\pi}}(\boldsymbol{a}) \,|\, \boldsymbol{a}^{-i} \sim \boldsymbol{\pi}^{-i}(\cdot)\big]$
- What are the IQL values if all agents explore randomly?
  - agent A learns incorrectly to go *backwards*

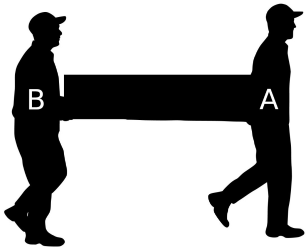


		A $\leftarrow$	A $\varnothing$	A $\rightarrow$	q^B_{exp}	q^B_{gre}
B	$\leftarrow$	0	-1	-1		
	$\varnothing$	0	0	-1	?	
	$\rightarrow$	0	0	$+1$		
q^A_{explore}		0	$-\frac{1}{3}$	$-\frac{1}{3}$		

this exploration effect on independent values is called *relative overgeneralization* (Panait et al., 2006); image source: pxhere.com

core concept: Relative overgeneralization

- Why is independent learning or value factorization problematic?
  - e.g. single-state IQL $q^i(a^i; \boldsymbol{\pi}^{-i}) := \mathbb{E}\big[Q^{\boldsymbol{\pi}}(\boldsymbol{a}) \,|\, \boldsymbol{a}^{-i} \sim \boldsymbol{\pi}^{-i}(\cdot)\big]$
- What are the IQL values if all agents explore randomly?
  - agent A learns incorrectly to go *backwards*
  - agent B learns correctly to go *forwards*
- Will the team learn the task when they act more greedy?

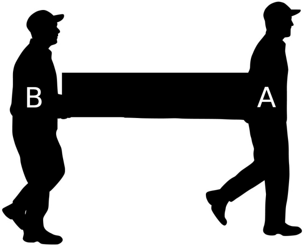


		A $\leftarrow$	A $\varnothing$	A $\rightarrow$	q^B_{exp}	q^B_{gre}
B	$\leftarrow$	0	-1	-1	$-\frac{2}{3}$	
B	$\varnothing$	0	0	-1	$-\frac{1}{3}$	?
B	$\rightarrow$	0	0	$+1$	$\frac{1}{3}$	
q^A_{explore}		0	$-\frac{1}{3}$	$-\frac{1}{3}$		
q^A_{greedy}			?			

this exploration effect on independent values is called *relative overgeneralization* (Panait et al., 2006); image source: `pxhere.com`

core concept: Relative overgeneralization

- Why is independent learning or value factorization problematic?
  - e.g. single-state IQL $q^i(a^i; \boldsymbol{\pi}^{-i}) := \mathbb{E}\big[Q^{\boldsymbol{\pi}}(\boldsymbol{a}) \,|\, \boldsymbol{a}^{-i} \sim \boldsymbol{\pi}^{-i}(\cdot)\big]$
- What are the IQL values if all agents explore randomly?
  - agent A learns incorrectly to go *backwards*
  - agent B learns correctly to go *forwards*
- Will the team learn the task when they act more greedy?
  - $q^A(a^A_{\rightarrow}) = \pi^B(a^B_{\rightarrow}) - (1 - \pi^B(a^B_{\rightarrow}))\,, \quad \pi^B(a^B_{\rightarrow}) > \frac{1}{2} \;\Rightarrow\; q^A(a^B_{\rightarrow}) > 0$



		A			q^B_{exp}	q^B_{gre}
		$\leftarrow$	$\varnothing$	$\rightarrow$		
B	$\leftarrow$	0	-1	-1	$-\frac{2}{3}$	0
	$\varnothing$	0	0	-1	$-\frac{1}{3}$	0
	$\rightarrow$	0	0	$+1$	$\frac{1}{3}$	$+1$
q^A_{explore}		0	$-\frac{1}{3}$	$-\frac{1}{3}$		
q^A_{greedy}		0	0	$+1$		

this exploration effect on independent values is called *relative overgeneralization* (Panait et al., 2006); image source: pxhere.com

Question: unlearnable games

- Create a reward matrix for a one-state 2 player cooperative game, which **cannot** be learned by IQL agents

assignment sheet 4

Question: unlearnable games

- Create a reward matrix for a one-state 2 player cooperative game, which **cannot** be learned by IQL agents
- for example the following table
  - note that an unfactored value $Q^\pi(\boldsymbol{a})$ could solve the task

		A		
	a_1^A	a_2^A	a_3^A	q^B_{exp}
B a_1^B	0	0	-2	$-\frac{2}{3}$
a_2^B	0	0	-2	$-\frac{2}{3}$
a_3^B	-2	-2	$+1$	-1
q^A_{explore}	$-\frac{2}{3}$	$-\frac{2}{3}$	-1	

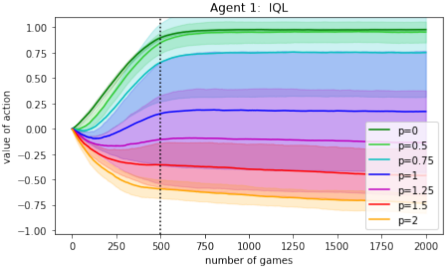
assignment sheet 4

Question: unlearnable games

- Create a reward matrix for a one-state 2 player cooperative game, which **cannot** be learned by IQL agents

- for example the following table
  - note that an unfactored value $Q^\pi(\boldsymbol{a})$ could solve the task
  - punishment p only increases *probability* of failure

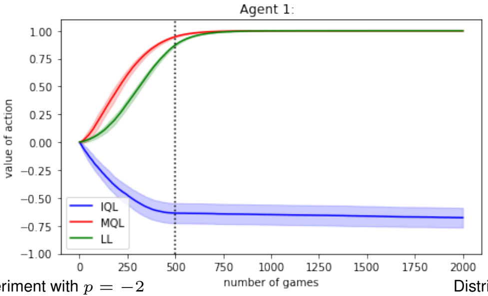
		A			
		a_1^A	a_2^A	a_3^A	q^B_{exp}
B	a_1^B	0	0	$-p$	$-\frac{p}{3}$
	a_2^B	0	0	$-p$	$-\frac{p}{3}$
	a_3^B	$-p$	$-p$	$+1$	$\frac{1-2p}{3}$
q^A_{explore}		$-\frac{p}{3}$	$-\frac{p}{3}$	$\frac{1-2p}{3}$	



assignment sheet 4

11.2 Optimistic return methods

- IQL always learn when partners behave optimal
  $$\mathrm{TD}(\tau^i_{t+1}, r^i_t) = r^i_t + \gamma \max_{a'^i} q^i_{\theta^i}(\tau^i_{t+1}, a'^i) - q^i_{\theta^i}(\tau^i_t, a^i_t)$$
- Distributed Q-learning pretends others play best
  - use best past reward: $\mathrm{TD}(\tau^i_{t+1}, \max\{r^i_{t'} \mid {}^{s_{t'}=s_t}_{a_{t'}=a_t}\})$
  - or ignore negative TD errors: $\max(0, \mathrm{TD}(\tau^i_{t+1}, r^i_t))$
  - both fail in stochastic environments



ϵ-greedy experiment with $p = -2$

Distributed Q-learning (Lauer and Riedmiller, 2000)

Optimistic return methods

- IQL always learn when partners behave optimal
  $\mathrm{TD}(\tau^i_{t+1}, r^i_t) = r^i_t + \gamma \max_{a'^i} q^i_{\theta^i}(\tau^i_{t+1}, a'^i) - q^i_{\theta^i}(\tau^i_t, a^i_t)$
- Distributed Q-learning pretends others play best
  - use best past reward: $\mathrm{TD}(\tau^i_{t+1}, \max\{r^i_{t'} \mid {}^{s_{t'}=s_t}_{a_{t'}=a_t}\})$
  - or ignore negative TD errors: $\max(0, \mathrm{TD}(\tau^i_{t+1}, r^i_t))$
  - both fail in stochastic environments
- Lenient learning reduces slowly to IQL
  - ignore negative TD errors only *sometimes*
  - decreases ignoring probability $\eta(s,a)$ with visitations
  - deep version must remember $\eta(s,a)$ in hash table

Lenient learning by Panait et al. (2008), deep lenient learning by Wei and Luke (2016)

11.2 Optimistic return methods

- IQL always learn when partners behave optimal
  $\mathrm{TD}(\tau_{t+1}^i, r_t^i) = r_t^i + \gamma \max_{a'^i} q_{\theta^i}^i(\tau_{t+1}^i, a'^i) - q_{\theta^i}^i(\tau_t^i, a_t^i)$

- Distributed Q-learning pretends others play best
  - use best past reward: $\mathrm{TD}(\tau_{t+1}^i, \max\{r_{t'}^i \mid {}^{s_{t'}=s_t}_{a_{t'}=a_t}\})$
  - or ignore negative TD errors: $\max(0, \mathrm{TD}(\tau_{t+1}^i, r_t^i))$
  - both fail in stochastic environments

- Lenient learning reduces slowly to IQL
  - ignore negative TD errors only *sometimes*
  - decreases ignoring probability $\eta(s,a)$ with visitations
  - deep version must remember $\eta(s,a)$ in hash table

- Weighted-QMIX weights some outcomes with $\alpha \in [0,1]$
  - QMIX with additional centralized Q-value function
  - weights TD-errors with α if greedy policies disagree

wighted QMIX by Rashid et al. (2020b), other approaches sample *similar tasks* optimistically (e.g. Gupta et al., 2021)

11.2 Higher order factorizations

- Unfactored value functions have huge action space
- Use higher order functions can coordinate more agents

$$Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}_t, \boldsymbol{a}) \;:=\; \underbrace{\overbrace{\sum_{i \in \mathcal{E}^1} q^i(\tau^i_t, a^i)}^{\text{VDN}} + \sum_{(i,j) \in \mathcal{E}^2} q^{ij}(\boldsymbol{\tau}^{ij}_t, \boldsymbol{a}^{ij})}_{\text{COORDINATION GRAPH}} + \sum_{(i,j,k) \in \mathcal{E}^3} \ldots$$

max-plus algorithm see Pearl (1988), CG introduced by Guestrin et al. (2002), tabular CG used in Kok and Vlassis (2006)

Higher order factorizations

- Unfactored value functions have huge action space
- Use higher order functions can coordinate more agents

$$Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}_t, \boldsymbol{a}) \;:=\; \underbrace{\overbrace{\sum_{i \in \mathcal{E}^1} q^i(\tau_t^i, a^i)}^{\text{VDN}} + \sum_{(i,j) \in \mathcal{E}^2} q^{ij}(\boldsymbol{\tau}_t^{ij}, \boldsymbol{a}^{ij})}_{\text{COORDINATION GRAPH}} + \sum_{(i,j,k) \in \mathcal{E}^3} \dots$$

- Maximum no longer decentralizable
  - e.g. *coordination graph* (CG) $\mathcal{G} := \langle \mathcal{E}^1, \mathcal{E}^2 \rangle$
  - all agents are nodes: $\mathcal{E}^1 := \{i\}_{i=1}^N$
  - pairwise coordination: $\mathcal{E}^2 \subseteq \{(i,j) | 1 \leq i \leq N, 1 \leq j \leq N\}$
  - maximum can be computed with *max-plus algorithm*
  - computed using multiple message passes between agents

max-plus algorithm see Pearl (1988), CG introduced by Guestrin et al. (2002), tabular CG used in Kok and Vlassis (2006)

Deep coordination graphs

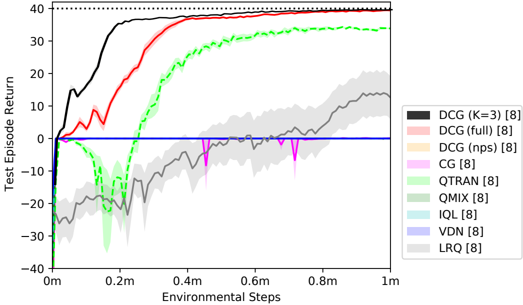
- Deep coordination graphs (DCG) implements pairwise edges

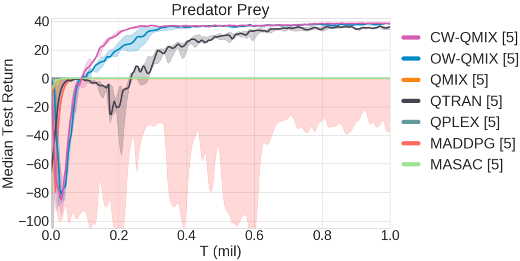
$$Q^{\text{DCG}}_{\theta\phi\psi\varphi}(s_t, \boldsymbol{\tau}_t, \boldsymbol{a}) := q^0_\varphi(s_t) + \sum_{i \in \mathcal{E}^1} q^i(\tau^i_t, a^i) + \sum_{(i,j) \in \mathcal{E}^2} q^{ij}(\boldsymbol{\tau}^{ij}_t, \boldsymbol{a}^{ij})$$

- Required extensive engineering to work with neural nets
  - shared history encoding: $\boldsymbol{h}^i_t := h_\psi(\tau^i_t) = h_\psi(\boldsymbol{h}^i_{t-1}, o^i_t, a^i_{t-1})$
  - shared utility functions in $\mathcal{E}^1$: $q^i(\tau^i_t, a^i) := q^v_\theta(\boldsymbol{h}^i_t, a^i)$
  - symmetrized shared payoff functions in $\mathcal{E}^2$:
    $q^{ij}(\tau^{ij}_t, \boldsymbol{a}^{ij}) := \frac{1}{2}\Big(q^e_\phi(\boldsymbol{h}^i_t, \boldsymbol{h}^j_t, a^i, a^j) + q^e_\phi(\boldsymbol{h}^j_t, \boldsymbol{h}^i_t, a^j, a^i)\Big)$
  - K-rank approximation of payoff matrices:
    $q^e_\phi(\boldsymbol{h}^i_t, \boldsymbol{h}^j_t, a^i, a^j) := \sum_{k=1}^{K} g^k_\phi(\boldsymbol{h}^i_t, \boldsymbol{h}^j_t, a^i)\, \bar{g}^k_\phi(\boldsymbol{h}^i_t, \boldsymbol{h}^j_t, a^j)$
  - state-dependent bias $q^0_\varphi(s_t)$

Böhmer et al. (2020) introduce DCG, and Castellini et al. (2019) explore higher order factors

Comparison: predator-prey





- 10x10 gridworld
- 8 agents, 8 prey
- reward $+10$/prey
- punishment -2/attempt
- catch removes agents+prey

image and upper results from Böhmer et al. (2020), lower results from (Rashid et al., 2020a)

11.2 Summary

- Random exploration can cause RO in independent learners
- When expected return is low only because others explore
- Can be counteracted by down-weighting errors with low returns
- Higher-order value factorization can represent joint-Q-values
- DCG requires message passing and many stabilization tricks

Learning Objectives

LO11.2: Derive when a game exhibits relative overgeneralization
LO11.3: Explain what can be done against relative overgeneralization

11.3

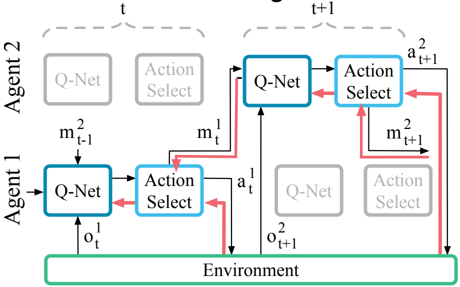
Advanced MARL
Communication

Communication between agents

- Coordination graphs use communication for message passing
  - communication protocol fixed by max-plus algorithm
  - can we learn a useful protocol between agents?
- Communication almost always possible
  - *implicitly* by manipulating the environment
  - *explicitly* by using “cheap-talk” channels
- Not every game benefits from communication
  - no incentive to help opponent in zero-sum games
  - messages in general-sum games can be antagonistic
  - fully observable cooperation games do not need communication
  - e.g. communicate all observations for centralized agent

- Learn communication implicitly as part of architecture
  - requires additional *cheap-talk* channels
  - messages m_t^i at time t are inputs of time $t+1$

- Continuous messages allow gradient flow between agents



e.g. Wang et al. (2018) use graph neural networks for message passing, figure modified from Förster et al. (2016)

core concept: Differentiable communication

- Learn communication implicitly as part of architecture
  - requires additional *cheap-talk* channels
  - messages m_t^i at time t are inputs of time $t+1$
- Continuous messages allow gradient flow between agents
- No gradient flow through sampled messages!
  - $m_t^i \in \{1, \ldots, M\}$
  - $m_t^i \sim \mu_\theta^i(\cdot|\tau_t^i)$

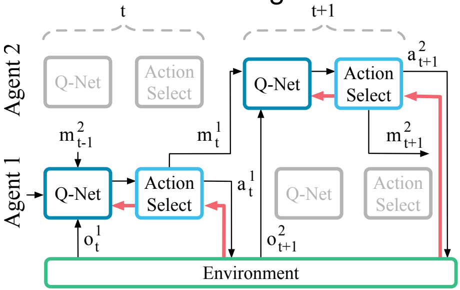


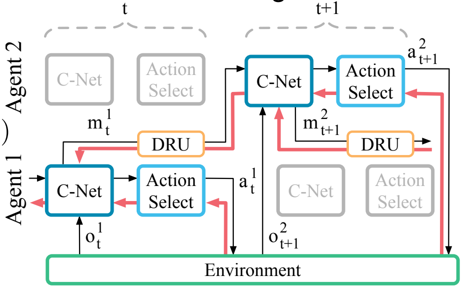
figure from Förster et al. (2016), sending (non-differentiable) messages between RNN-agents

core concept: Differentiable communication

- Learn communication implicitly as part of architecture
  - requires additional *cheap-talk* channels
  - messages m_t^i at time t are inputs of time $t+1$

- Continuous messages allow gradient flow between agents

- No gradient flow through sampled messages!
  - $m_t^i \in \{1, \ldots, M\},\ \epsilon \sim \mathcal{N}(\cdot|0,1)$
  - $m_t^i = f_\theta(\tau_t^i, \epsilon) \sim \mu_\theta^i(\cdot|\tau_t^i)$



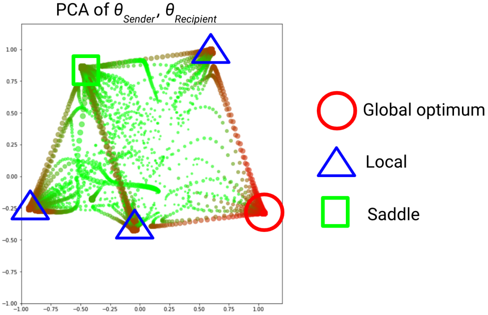
- Use reparametrization trick
  - gradient flows through f_θ
  - $\nabla_\theta \mathbb{E}\Big[g(m_t^i)\,\Big|\, m_t^i \sim \mu_\theta^i(\cdot|\tau_t^i)\Big] = \mathbb{E}\Big[\nabla_\theta\, g\big(f_\theta(\tau_t^i, \epsilon)\big)\,\Big|\, \epsilon \sim \mathcal{N}(\cdot|0,1)\Big]$

figure with reparametrizaytion modules (DRU) from Förster et al. (2016), see also Jang et al. (2017) for differentiable sampling

11.3 Learning communication protocols

- Can we learn to communicate in cooperative games?
  - provide some communication actions, e.g. $m_t^i \in \{0, 1\}$
  - negotiate meaning of m_t^i during training
- Established conventions are hard to change, e.g.
  - A sees $\triangle \Rightarrow m^A = 1$
  - $m^A = 1 \Rightarrow$ B does $\triangle$
- Many Nash-equilibria!

		B does		
		◯	△	▽
A sees	◯	$+1$	-1	-1
	△	-1	$+\frac{1}{2}$	-1
	▽	-1	-1	$+\frac{1}{2}$



example and PCA plot by Jakob Förster

Summary

- Communication can help to coordinate agents
- Not all games benefit from communication
- Differentiable communication in centralized training
- Discrete channels must use the reparameterization trick
- Learned communication protocols have many Nash equilibria

Learning Objectives

LO11.4: Explain in which games communication is beneficial
LO11.5: Explain differentiable communication and how it can fail

11.3 Next lecture

- Next lecture: **applied RL**!
- Questions? Ask them here: answers.ewi.tudelft.nl

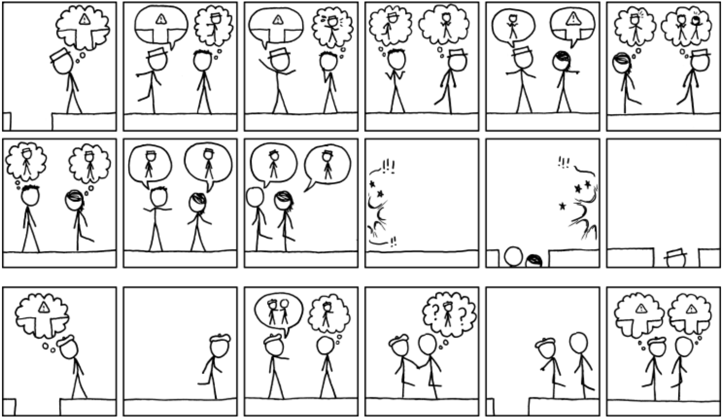


image source: xkcd.com

References I

Wendelin Böhmer, Vitaly Kurin, and Shimon Whiteson. Deep coordination graphs. In *Proceedings of Machine Learning and Systems (ICML)*, pages 2611–2622, 2020. URL https://arxiv.org/abs/1910.00091.

Jacopo Castellini, Frans A. Oliehoek, Rahul Savani, and Shimon Whiteson. The representational capacity of action-value networks for multi-agent reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, pages 1862–1864, 2019. URL http://www.ifaamas.org/Proceedings/aamas2019/pdfs/p1862.pdf.

Jakob Förster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems 29*, pages 2137–2145. 2016. URL http://papers.nips.cc/paper/6042-learning-to-communicate-with-deep-multi-agent-reinforcement-learning.pdf.

Carlos Guestrin, Michail Lagoudakis, and Ronald Parr. Coordinated reinforcement learning. In *ICML*, volume 2, pages 227–234, 2002.

Tarun Gupta, Anuj Mahajan, Bei Peng, Wendelin Böhmer, and Shimon Whiteson. UneVEn: Universal value exploration for multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*. 2021. URL https://arxiv.org/abs/2010.02974.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumbel-softmax. In *Proceedings International Conference on Learning Representations 2017*, 2017. URL https://openreview.net/pdf?id=rkE3y85ee.

Jelle R Kok and Nikos Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research*, 7(Sep):1789–1828, 2006.

Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *In Proceedings of the Seventeenth International Conference on Machine Learning*, pages 535–542. Morgan Kaufmann, 2000.

Liviu Panait, Sean Luke, and R. Paul Wiegand. Biasing coevolutionary search for optimal multiagent behaviors. *IEEE Transactions on Evolutionary Computation*, 10(6):629–645, 2006.

References II

Liviu Panait, Karl Tuyls, and Sean Luke. Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *The Journal of Machine Learning Research*, 9:423–457, 2008.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. ISBN 0-934613-73-7.

Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Förster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 4292–4301, 2018.

Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation. In *Advances in Neural Information Processing Systems*, 2020a. URL https://arxiv.org/abs/2006.10800.

Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Förster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 2020b. URL https://arxiv.org/abs/2003.08839.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Förster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5887–5896, 2019. URL http://proceedings.mlr.press/v97/son19a.html.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 2085–2087, 2018.

References III

Tingwu Wang, Renjie Liao, Jimmy Ba, and Sanja Fidler. Nervenet: Learning structured policy with graph neural networks. In *ICLR*, 2018. URL https://openreview.net/forum?id=S1sqHMZCb.

Ermo Wei and Sean Luke. Lenient learning in independent-learner stochastic cooperative games. *The Journal of Machine Learning Research*, 17(1):2914–2955, 2016.