Grounded Semantic Networks for Learning Shared Communication Protocols

Matthew Hausknecht
Department of Computer Science
University of Texas at Austin
mhauskn@cs.utexas.edu

Peter Stone
Department of Computer Science
University of Texas at Austin
pstone@cs.utexas.edu

Abstract

Cooperative multiagent learning poses the challenge of coordinating independent agents. A powerful method to achieve coordination is allowing agents to communicate. We present the Grounded Semantic Network, an approach for learning a task-dependent communication protocol grounded in the observation space and reward function of the task. We show that the grounded semantic network effectively learns a communication protocol that is useful for achieving cooperation between agents. Analyzing the messages transmitted between agents reveals that the agents' policies are highly influenced by the communication received from teammates. Further analysis highlights the limitations of the grounded semantic network, identifying the characteristics of domains that it can and cannot solve.

1 Introduction

Cooperation is the process where groups of organisms act together for mutual benefit. In human society, communication is a key ingredient of successful cooperation. With communication, humans can identify a problem, collectively discuss strategies for addressing that problem, and solve the problem using a strategy that employs further communication. Through communication, human cooperation can emerge on timescales of minutes rather than years (or millenia) required for cooperation to evolve in nature. However, communication relies on a pre-established syntax and protocol. This paper is concerned with the question of how multiple agents can learn a useful communication protocol that helps them cooperate on an arbitrary task.

A communication protocol should answer two questions: First, the world around us is unfathomably complex. Of all the possible phenomenon that could be described, what concepts *should* be communicated? Second, *how* should the communication protocol transform concepts into messages?

In this paper, we introduce the grounded semantic network (GSN), a neural network that learns a task-dependent communication protocol. We demonstrate that the learned communication protocol allows agents to cooperatively solve tasks that are otherwise inaccessible. Finally, we analyze the learned communication protocol and show how it maps concepts to messages.

2 Background

We consider multiagent reinforcement learning tasks that are fully cooperative and partially observable. Specifically, our experiments examine episodic two-agent tasks: at each timestep t, Agent$^{(1)}$ and Agent$^{(2)}$ receive observations $o_t^{(1)}, o_t^{(2)}$, select actions $a_t^{(1)}, a_t^{(2)}$, sends messages $m_t^{(1)}, m_t^{(2)}$, and receive rewards $r_t^{(1)}, r_t^{(2)}$ respectively. Communication takes places over a continuous channel: each agent is allowed to transmit a message $m \in \mathbb{R}^n$ where n is the number of continuous values that may be transmitted at each timestep (e.g. bandwidth of the communication channel). Transmitted

Workshop on Deep Reinforcement Learning (NIPS 2016), Barcelona, Spain.

messages are received by the teammate in the timestep after they are sent and are concatenated with the receiving agent's state-observation $o_t^{(1)} = o_t^{(1)} \oplus m_{t-1}^{(2)}$.

3 Grounded Semantic Network

The *grounded semantic network* learns a task-dependent communication protocol that simultaneously answers the questions of what concepts should be communicated and how they should be encoded.

The GSN (depicted in Figure 1) is a network that models the teammate's one-step reward, conditioned on the teammate's action and the agent's state-observation. The hidden-layer activations of this model are communicated to the teammate as a message. The communication protocol learned by the GSN is grounded in the reward function of the task and embodies a semantic mapping: a transformation from concept to message.

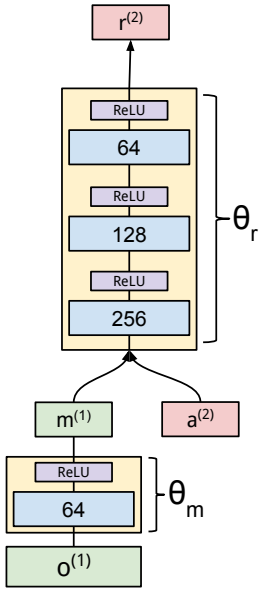


Figure 1: **The Grounded Semantic Network** predicts the teammate's one-step reward $r^{(2)}$ conditioned on the agent's current observation $o^{(1)}$ and teammate's action $a^{(2)}$. The message $m^{(1)}$ is an intermediate layer in this network, and learns a compact representation of the current observation that is useful for predicting teammate reward. The activations of the message layer are transmitted to the teammate as the message. Training a GSN requires direct access to the teammate's actions and rewards. However, at test time, only the agent's observations are required to generate the message.

From the perspective of Agent$^{(1)}$, a GSN learns a mapping from the observation $o^{(1)}$ and teammate action $a^{(2)}$ to teammate reward $r^{(2)}$. The network contains two major parts: a message extractor $m^{(1)} = \mathcal{M}(o^{(1)}; \theta_m)$ which maps the agent's observation into a message, and a one-step reward model $\hat{r}^{(2)} = \mathcal{R}(m^{(1)}, a^{(2)}; \theta_r)$ that predicts the teammate's immediate reward. Composing these components, the the full GSN computes the following function:

$$\hat{r}^{(2)} = \mathcal{R}\Big(\mathcal{M}\big(o^{(1)}; \theta_m\big), a^{(2)}; \theta_r\Big) \tag{1}$$

Training. GSN training follows a supervised learning paradigm. Given experience tuple $(o^{(1)}, a^{(2)}, r^{(2)})$, the GSN is trained to regress its predictions towards the rewards of the teammate, minimizing the following loss function:

$$L(\theta_r, \theta_m) = \mathbb{E}_{(o^{(1)}, a^{(2)}, r^{(2)})}\left[\Big(r^{(2)} - \mathcal{R}\big(\mathcal{M}(o^{(1)}; \theta_m), a^{(2)}; \theta_r\big)\Big)^2\right] \tag{2}$$

GSN training happens in parallel with the updates to the learning agents. Specifically, we perform a single update on the GSN for every update to both learning agents. Joint experience tuples $(o^{(1)}, a^{(2)}, r^{(2)})$ are drawn uniform at random from a replay queue of experience. The GSN does not depend on the choice of policy representation for the learning agents. In this paper, the agents use an actor-critic architecture to handle the continuous action space, but the GSN would be equally applicable to DQN-style architectures in discrete action spaces. GSN's only requirement is the ability to perform updates over joint experience tuples.

GSN follows a centralized training procedure with decentralized execution: during training, GSN requires direct access to the teammate's actions and rewards. Direct access breaks the standard boundaries between independent agents. However, at execution time, only the agent's current observation is required to generate a message. This paradigm is similar to a team sport – at practice, team members experiment with different strategies and compare notes about what each player did and how well the strategy worked. At competition, there is only time to execute the practiced strategies.

Stability. In the context of deep reinforcement learning, experience tuples are generated from interactions with the environment, stored in a replay queue, and sampled randomly for updates. Unlike standard supervised learning from a fixed dataset, there is a dangerous loop in which the policies of the agents affect the messages generated by the GSN, which, in turn, affect the policies of the agents. Without care, such a loop can result in instability or collapse of the communication protocol and the agents' policies. To alleviate this danger, we train the GSN with a learning rate of 10^{-6}, an order of magnitude smaller than the learning rate used to train the agents' policies. This reduced learning rate encourages slow changes to the GSN and allows the agents to smoothly adapt to alterations in the communication protocol[1]. GSN is trained using the Adam optimizer and updated once for each update to the agent.

Relation to Reward Models. The GSN is similar to a one-step teammate reward model. The GSN differs from a standard reward model $\hat{r}^{(2)} = \mathcal{R}(o^{(2)}, a^{(2)})$ by replacing the teammate's observation $o^{(2)}$ with a message $m^{(1)}$ generated through an encoder $\mathcal{M}$ conditioned on the agent's state observation $o^{(1)}$. In this sense, the message encoder can be understood as trying to reconstruct elements of $o^{(2)}$ that are predictive of $r^{(2)}$.

Limitations. Intuitively, the GSN combats partial observability of the multiagent environment by learning a transformation of the agent's observation that is relevant for predicting the teammate's reward. In essence, if there is information available in Agent$^{(1)}$'s observations that could help Agent$^{(2)}$'s performance, the GSN will learn to extract and communicate this information. However, the GSN is not conditioned on Agent$^{(1)}$'s policy and cannot extract the intentions of Agent$^{(1)}$. The abilities and limitations of the GSN are explored further in the experiments below.

4 Related Literature

There is an extensive body of literature on communication between reinforcement learning agents [7, 5, 8].

Similar to the blind soccer experiments in Section 6.1, Stroupe et al. [6] show that a blindfolded robot is able to track a moving soccer ball by using the observation of two sighted teammates. This work reinforces the idea that multiple agents can overcome partial observability by fusing distributed sensor readings. However, these robots rely on a pre-established rather than learned communication protocol.

Learned communication in the context of deep reinforcement learning has been studied by Foerster et al. [1], who presents two approaches: Reinforced Inter-Agent Learning (RIAL) and Differentiable Inter-Agent Learning (DIAL). Due to the importance of these approaches we discuss them in more detail below:

Reinforced Inter-Agent Learning (RIAL) is a decentralized procedure which treats communication the same way as normal action: both agents independently learn communication actions that maximize personal reward. In other words, communication is the same as an extended action space, with communication actions having a direct effect on the teammate's observations rather than the

[1] No systematic stability gains were observed from using target networks or reducing the ratio of GSN updates versus policy updates.

environment. While RIAL has the capability of learning a communication protocol, there is no mechanism to encourage stable or meaningful transmissions. As such, RIAL serves as a baseline.

Differentiable Inter-Agent Learning (DIAL), is a more sophisticated approach that allows each agent to alter the teammate's message by applying a gradient to the teammate's communication actions. In continuous action space there are different possible ways to realize DIAL. The next section presents one method of sharing gradients between agents. Since this implementation is only one way DIAL may be realized in continuous space, we refer to the method as *Teammate Communication Gradients* rather than DIAL.

5 Teammate Communication Gradients

Teammate Communication Gradients (TCG) allows each agent to alter the teammate's message by applying a gradient to the teammate's communication actions. Implemented in the DDPG framework, each agent's communication actions are updated according to the following gradients:

$$\nabla_{\theta^\mu}\mu^{(1)}(o^{(1)}) = \nabla_{m^{(1)}}Q^{(2)}(o^{(2)}, a^{(2)}|\theta^Q)\nabla_{\theta^\mu}\mu^{(1)}(o^{(1)}|\theta^\mu)$$

where $\mu^{(1)}$, parameterized by θ^μ, is Agent-1's actor network, and $Q^{(2)}$ parameterized by θ^Q, is Agent-2's critic network.

Essentially these gradients serve as a way for each agent to shape the messages that are sent by the teammate. TCG is end-to-end trainable across agents, and follows the paradigm of centralized training and decentralized execution. By allowing each agent to alter the messages sent by its teammate, each agent can push the teammate towards a stable communicate protocol.

RIAL and TCG are validated on several partially observed cooperative domains, and results show TCG is able to reach high performance faster than RIAL, indicating that fully-differentiable communication benefits cooperative multiagent learning. The experiments in the next section compare RIAL and TCG to GSN.

6 Experiments

We evaluate RIAL, TCG, and GSN on two domains: the first features rewards generated from the environment, and communication is not directly rewarded. In contrast, the second domain features a reward signal that stems directly from the *content* of the communicated messages. In the first domain, communication is only a means to achieve cooperation, whereas in the second, it is an end in and of itself.

Setup. Both domains are implemented within the Half-Field-Offense (HFO) simulated 2D soccer framework [2]. HFO features a continuous observation space ($o \in \mathbb{R}^{66}$) that consists of angles and distances to salient soccer objects such as the ball, goal, and teammate. Agents use a parameterized-continuous action space ($a \in \mathbb{R}^{10}$) in which an agent must choose between dashing, turning, or kicking, and then select continuous parameters to accompany that action. For example, if the agent decides to turn, it must specify a continuous value corresponding to the desired degrees to turn. For more information on the Half-Field-Offense environment see [2].

The learning agents used in this paper extend prior work on deep reinforcement learning in parameterized action space [3]. Specifically, the actor-critic architecture employed by the DDPG algorithm [4] enables learning in HFO's parameterized-continuous action space. For more information on the network architecture and updates used for learning, see [3].

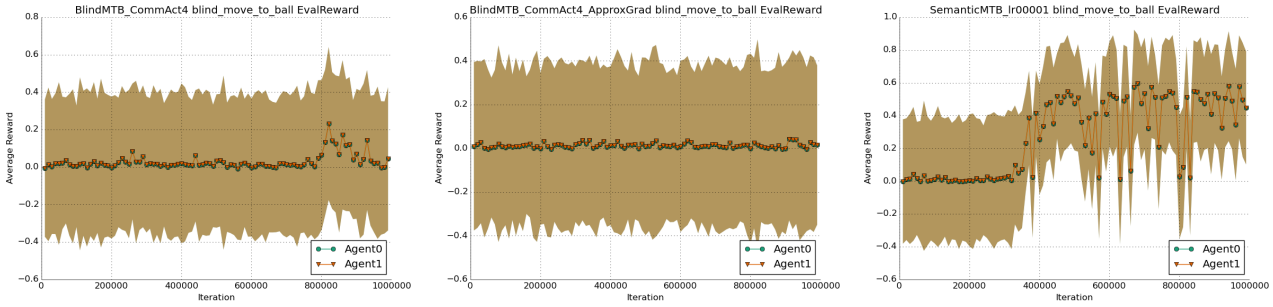
Agents are modified with a variable number of continuous communication actions. Each communication action transmits a single floating point value. In RIAL, communication actions are treated identically to all other continuous actions and are learned by following gradients generated by the critic network. In TCG, gradients for communication actions are generated in the same fashion, but swapped with the teammate. In this way, each agent directly influences the messages communicated by its teammate. Finally, using a GSN, there are no communication actions and messages are generated directly from the GSN.

6.1 Blind Soccer

The objective of the blind soccer task is to steer a blind agent towards the soccer ball using communication. This task features asymmetric information: the blind agent ($\text{Agent}^{\text{blind}}$) cannot see anything on the field: its state features are present but constantly zeroed. However, it can still hear incoming communication messages. The blind agent's teammate ($\text{Agent}^{\text{sight}}$) has normal sight but cannot move[2].

Both agents are rewarded for minimizing the distance between the blind agent and the ball: $r_t = -\Delta d(\text{Agent}^{\text{blind}}, \text{Ball})$, where $d(\cdot, \cdot)$ is Euclidean distance. Episodes start with the blind agent, teammate, and ball initialized randomly on the field and end when the blind agent reaches the ball or 100 timesteps pass. To solve this task, $\text{Agent}^{\text{sight}}$ must learn a stable protocol for directing $\text{Agent}^{\text{blind}}$ to the ball. It is impossible for either agent to solve the task alone or without communication.

Experiments compare RIAL, TCG, and GSN using a communication channel of size four. Results in Figure 2 show that only GSN successfully solves the task[3].



(a) RIAL (b) TCG (c) GSN

Figure 2: **Blind Soccer:** Performance of RIAL, TCG, and GSN. The maximum achievable reward is .6. Note the variance of rewards is high because total reward is proportional to the distance between $\text{Agent}^{\text{blind}}$ and the ball. Random initialization of agents and ball results in high variance of reward even for a perfect agent.

To further understand these results, it is necessary to recognize that TCG's use of communication gradients is not grounded in reality: $\text{Agent}^{\text{blind}}$ uses the shared communication gradients to shape $\text{Agent}^{\text{sight}}$'s communication into messages it wants to hear, but the messages it wants to heard don't necessarily reflect reality. For example, $\text{Agent}^{\text{blind}}$ wants to hear that the ball is directly ahead, because it can easily dash forward and obtain reward. In essence, $\text{Agent}^{\text{blind}}$'s use of shared gradients is detached from reality in the sense that the ball may or may not actually be ahead of $\text{Agent}^{\text{blind}}$. In contrast, GSN learns a model that maps from observation to teammate reward. Thus, communication remains grounded by the actual state of the environment.

7 Analysis

We perform an ablation analysis on policies learned in the Blind Soccer task by disabling communication and running each of the learned policies. Policies learned by RIAL and TCG remain unchanged when communication is disabled, indicating that communicated messages are not actively used by $\text{Agent}^{\text{blind}}$. In contrast, GSN's policy is adversely affected by a lack of communication: without guidance from $\text{Agent}^{\text{sight}}$, $\text{Agent}^{\text{blind}}$ walks directly forward regardless of the location of the ball.[4]

In order to analyze the communication protocol learned by GSN on the blind soccer task, Figure 3 visualizes the space of messages. There is a strong correlation between the content of the message and the action selected by the teammate in the next timestep. This correlation illustrates that messages contain information that is used by the blind agent to decide whether it should dash or turn.

[2] Specifically, $\text{Agent}^{\text{sight}}$ can turn but cannot dash.

[3] `https://www.cs.utexas.edu/~mhauskn/eln/dqn-hfo/blind_move_to_ball/2016-10-17/SemanticMTB_NoTanh.mp4`

[4] `http://www.cs.utexas.edu/~larg/hausknecht_thesis/SemanticMTB_NoTanh_DisabledComm.mp4`

Figure 3: **t-SNE Visualization:** Two dimensional projection of 4-dimensional messages sent by Agent$^{\text{sight}}$ while performing 10 episodes of the blind soccer task. Each message is colored according to the action taken by Agent$^{\text{blind}}$ in the next timestep: black dots correspond to Dash actions and white dots are Turn actions. The content of the Agent$^{\text{sight}}$'s messages influences the actions selected by Agent$^{\text{blind}}$.

7.1 Guess My Number

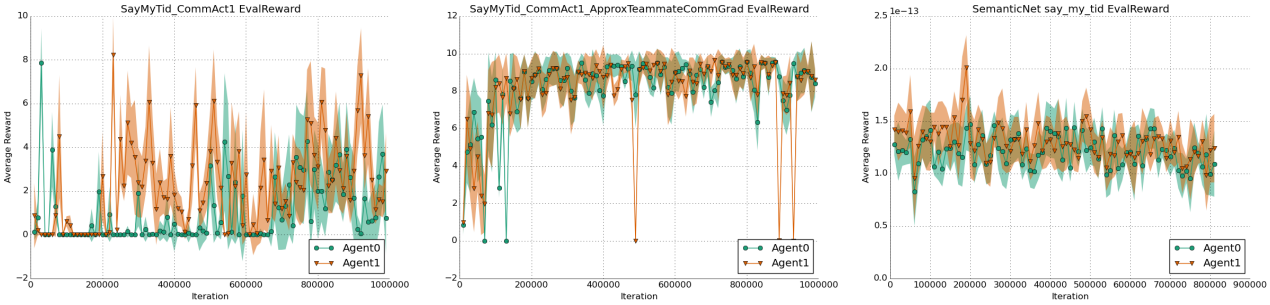
Guess My Number is a two-player game in which each agent is assigned a secret number, represented by a single floating point value. The goal is for each agent to help its teammate correctly guess its hidden number. This domain uses a single communication action, so each agent is allowed to send one floating point value every timestep. Both agents are rewarded for minimizing the distance between the teammate's message $m \in \mathbb{R}^1$ and their own hidden value $h^{(i)}$. Specifically, reward for Agent$^{(1)}$ is $r_t^{(1)} = \alpha / e^{\beta(h^{(1)} - m_{t-1}^{(2)})^2}$, where $\alpha = .1$ and $\beta = 50$ are constants controlling the magnitude and decay of reward. Reward is symmetric for Agent$^{(2)}$.

To solve this game, each agent must convince its teammate to communicate a message resembling its own hidden number. As shown in Figure 4, only TCG successfully solves the task by shaping the teammate's messages in a way that maximizes personal reward. Since RIAL has no control of the teammate's messages, neither agent can find a way to maximize personal reward. Likewise, GSN cannot solve this task because the network is trained to predict rather than maximize teammate reward: the messages learned by GSN only correlate with teammate reward and have no incentive to converge towards the teammate's hidden number.

Guess My Number is an instance of a class of domains in which rewards correspond only to content of communicated messages, rather than interactions with the environment. These domains highlight a limitation of GSN: the inability to directly alter communication following a reward gradient. In such domains, TCG remains the method of choice since it can directly alter the content of messages in the direction of higher rewards.

8 Future Work and Conclusion

This paper presented the Grounded Semantic Network, a trainable model that learns a task-dependent communication protocol for solving cooperative multiagent tasks. We introduced and evaluated the GSN on two domains - the Guess My Number task rewarded optimizing the content of messages, and the Blind Soccer task used communication as a means to solve a guide a blind agent to a soccer ball. GSN outperforms existing algorithms on Blind Soccer task in which communication is used to achieve a goal in the environment. Analyzing the communicated messages shows that the communication protocol is highly correlated to the actions selected by the blind teammate. In



(a) RIAL (b) TCG (c) GSN

Figure 4: **Guess My Number:** Performance of RIAL, TCG, and GSN. The maximum achievable reward is 10.

general, these results highlight the ability of communication to overcome partial observability and help facilitate cooperation between independent agents.

Comparing GSN to communication learning approaches that directly optimize the content of communicated messages, we see that direct optimization of message content (TCG) is most effective on domains in which reward stems directly from the communicated message and the messages don't need to reflect the state of the environment. GSN works better in domains in which communication is used as a means to accomplish some objective in the environment. In such cases, the learned communication protocol remains grounded in reality.

An interesting possibility for extending the GSN: it would be desirable to have a grounded communication learning method that optimized the content of messages towards higher rewards, a method would be able to solve both tasks examined in this paper. One approach would be to perform updates on the GSN that alternated between predicting and maximizing teammates rewards. In such a way, the learned messages would be both grounded and optimized to maximize teammate reward.

References

[1] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *CoRR*, abs/1605.06676, 2016.

[2] Matthew Hausknecht, Prannoy Mupparaju, Sandeep Subramanian, Shivaram Kalyanakrishnan, and Peter Stone. Half field offense: An environment for multiagent learning and ad hoc teamwork. In *AAMAS Adaptive Learning Agents (ALA) Workshop*, May 2016.

[3] Matthew Hausknecht and Peter Stone. Deep reinforcement learning in parameterized action space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, May 2016.

[4] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *ArXiv e-prints*, September 2015.

[5] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

[6] Ashley Stroupe, Martin C. Martin, and Tucker Balch. Distributed sensor fusion for object position estimation by multi-robot systems. In *IEEE International Conference on Robotics and Automation, May, 2001*. IEEE, May 2001.

[7] Ming Tan. Multi-agent reinforcement learning: Independent versus cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning (ICML 1993)*, pages 330–337, San Francisco, CA, USA, 1993. Morgan Kauffman.

[8] Chongjie Zhang and Victor Lesser. Coordinating multi-agent reinforcement learning with limited communication. In *Proceedings of the 2013 International Conference on Autonomous Agents*

and Multi-agent Systems, AAMAS '13, pages 1101–1108, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.