Understanding the Role of Perception in the Evolution of Human Language

Isaac Davis (isaacd@andrew.cmu.edu)
Department of Philosophy, Carnegie Mellon University

Abstract

In this paper, we propose a flexible modeling framework for studying the role of perception in language learning and language evolution. This is achieved by augmenting some novel and some existing evolutionary signaling game models with existing techniques in machine learning and cognitive science. The result is a "grounded" signaling game in which agents must extract relevant information from their environment via a cognitive processing mechanism, then learn to communicate that information with each other. The choice of cognitive processing mechanism is left as a free parameter, allowing the model to be tailored to a wide variety of problems and tasks. We present results from simulations using both a Bayesian perception model and a neural network based perception model, which demonstrate how perception can "preprocess" environmental data in a way that is well suited for communication. Lastly, we discuss how the model can be extended to study other roles that perception may play in language learning. **Keywords:** Evolutionary Signaling Games, Perception, Language Evolution, Reinforcement Learning

Introduction

In this paper, we are interested in studying three broadly defined types of interaction between perception and language learning. The first, and perhaps most obvious, is how our perceptions of the world constrain and affect our ability to learn language. The second type of interaction is the reverse of the first- how does learning a language shape our perceptions of the world? The third interaction, with a long history of philosophical inquiry, is how differing cognitive representations *between* agents affects their ability to communicate with each other. We propose a flexible modeling framework that can be used to represent and study all three types of interactions between perception and language learning.

This framework has two core components. The first are evolutionary signaling games which have been used extensively to explain a wide variety of communication phenomena in animals, from mating calls and warning calls in mammals and birds, to pheromone signaling in insect communities. More recent work has applied these models to the evolution of *human* language and human linguistic phenomena, such as compositionality (Franke 2015) and convex perceptual categories (Jager 2007, O'Connor 2014). These models typically represent a situation in which two agents must coordinate their behavior in such a way as to achieve some common goal, but without any pre-defined common language with which to communicate.

We shall see, however, that the standard signaling game model is not well suited for studying interactions between cognition and language learning. In order to represent perception *and* language in the same model, we derive a "grounded" signaling game, in which agents respond to raw sensory inputs, rather than cleanly-defined information states. We

achieve this using unsupervised learning techniques from Machine Learning and Cognitive Science.

In the following section, we present the relevant background in signaling games and reinforcement learning, and briefly discuss why the standard model is not yet equipped for studying interactions between perception and language learning. We then present our grounded signaling game model, and draw on literature in Deep Reinforcement Learning to derive an effective learning rule. In order to avoid introducing too much complexity at once, we first present the model with a "trivial" perception mechanism, consisting of the identity map on sensory inputs. After deriving the learning rule for this model, we then outline how to incorporate a non-trivial perception mechanism, and briefly discuss the two perception mechanisms we will test. We then present results from a battery of experiments in which the agents must learn to communicate structured visual information from a synthetic image environment. These results provide insights into the role of perception as a "pre-processing" step for communication, and reveal some interesting interactions between the type of environment and perception mechanism. Lastly, we discuss how the current model can be extended to study all three types of interaction addressed in the introduction.

Background and Related Material

Signaling Games

In a two-player signaling game, the *sender* observes an information state $d \in \mathcal{D}$, drawn from some probability distribution $P(\mathcal{D})$. The sender then chooses a signal $x \in \mathcal{X}$ according to some decision rule, and transmits the signal to the *receiver*. The receiver then observes the signal, and chooses some act $a \in \mathcal{A}$ according to the receiver's decision rule. Both players then receive a payout $R(d, a)$, which is a function of the state, and act, but not the signal. The game is fully cooperative, in that both players receive the same payout every round.

The sender's decision rule can be represented by a function assigning each state d to a probability distribution $P_s(x|d)$ over signals, and likewise for the receiver's rule $P_r(a|x)$. With this notation, the *expected reward* for both players is given by

$$E[R|P_s, P_r] = \sum_{d \in \mathcal{D}, x \in \mathcal{X}, a \in \mathcal{A}} P(d)P_s(x|d)P_r(a|x)R(d, a) \quad (1)$$

An *evolutionary* signaling game is played for many rounds, and after each round, players update their strategies according to some learning rule. The most well-studied and often used form of reinforcement learning in signaling games is Roth-Erev reinforcement learning. A Roth-Erev learning agent (the sender, for this example) is represented by a set of *decision parameters* W^s, one parameter for each state-signal

pair. Each weight $w_{i,j}^s$ represents the sender's unnormalized probability of choosing signal x_i given state d_j, and the Roth-Erev update rule is very simple. If in one round of play the sender observes state d_j, chooses signal x_i, and receives reward R, then the sender updates weight $w_{i,j}^s$ by some fixed proportion of the reward, $\Delta w_{i,j}^s = \alpha R$, and leaves all other weights unchanged.

In a signaling game with two equiprobable states, Roth-Erev reinforcement learning will always converge to a *signaling system equilibrium*, in which the correct act is always chosen. In games with more than two states, or with non-uniformly distributed states, Roth-Erev learning will sometimes converge to a *partial pooling equilibrium*, in which some states are communicated accurately, and others are "pooled" into a single signal. Such outcomes are Nash Equilibria, meaning that neither player can improve the outcome by unilaterally changing their own strategy. The probability of converging to such partial pooling equilibria increases with the number of states (Huttegger, Skyrms, Smead, & Zollman 2010).

While our general framework is compatible with any signaling game model, the experiments for this paper will be based on the Sim-Max game (Jager 2007), a variation of the standard signaling game. In a Sim-Max game, the receiver's goal is to guess which state the sender observed, and the reward function is a distance metric representing "similarity" between states. Such models are used when the state space is much richer than the signal space, and one-to-one state-signal mappings are no longer possible. An example of this is color- even though colors vary continuously, we employ a relatively small number of words for describing them.

Limitations of the Standard Model

The Roth-Erev reinforcement learning model is favored in signaling games for its simplicity and minimality of cognitive assumptions. However, there are two factors that prevent us from using the standard model for our purposes. The first factor is entirely practical. In particular, Roth-Erev reinforcement learning, while simple and well studied, does not scale well to larger problems, as it requires a single decision parameter for each state-signal pair. This makes Roth-Erev learning prohibitively slow to converge on large problems, and restricts us to relatively small simulations.

The second problem with the standard model is more conceptual. In the standard treatment of signaling games, states are represented as uniform, discrete "labels," or in the usual Sim-Max game as a uniformly distributed compact subset of Euclidean space. The key feature in these models is that there is no internal structure to the states themselves. That is, the players cannot discriminate between two states *except* through the reward function- two different states are either identical or not, but any further delineation between states can only be inferred from their effect on the payout. This, however, is often regarded as an advantage of the standard representation, rather than a limitation. The agents need not be

endowed with an inner mental language (Skyrms 2010), they need not know what the game is "about," or even that they are playing a game at all. It is certainly important, if we wish to study interactions between perception and language-learning, that we not make any restrictive cognitive assumptions which would "screen off" the cognitive details of interest. However, outside of very simple or tightly controlled experimental settings, the standard representation actually imposes some very strict assumptions about the agents' perceptions, albeit implicitly.

To illustrate this, consider the following two very similar, but not identical signaling games: in Game 1, the sender observes one of ten cards $C_1, \ldots, C_{10}$, each of which depicts a digit $0 - 9$, and sends one of ten signals. The receiver must pull on one of ten levers, each of which bears a digit $0 - 9$. If the card and chosen lever show the same number, both players receive a reward of 1, otherwise they receive no payout. This game fits exactly into the signaling game framework described above.

Now consider Game 2: in Game 2, the signals and the receiver's actions are the same as Game 1. However, instead of observing one of ten cards, the sender now observes a card depicting a handwritten digit $0 - 9$. How would we represent the state space for Game 2? A first guess might be to represent it as having the same state space as Game 1, since each observation depicts one of 10 digits. But unlike Game 1, we cannot guarantee uniformity across separate instances of the same digit. Not all instances of 0 will look the same, and so they are distinct information states. To assume that they are not distinct information states is to assume that the sender *perceives* them as the same, or *recognizes* that they are of the same category. But these are acts of cognition. *Seeing* a digit handwritten on a card is not the same an *recognizing* which digit it depicts, or even recognizing that it is a digit at all. In this sense, the standard approach of representing states and acts as finite sets of distinct labels implicitly imposes presupposes a fixed way in which agents perceive, recognize, and process their environments.

This brings us to the two underlying principles of our framework. First, in order to represent all of the information available to agents, without explicitly assuming how they perceive or represent that information, we must represent states as raw sensory inputs, rather than discrete information states. Second, because sensory inputs tend to be high-dimensional, it is no longer the case, as with the standard signaling game, that states cannot be distinguished except through the reward function. In particular, Unsupervised Learning algorithms can infer informational structure from high-dimensional sensory inputs *without* any feedback or supervision. Thus, by integrating an unsupervised perception model into a "grounded" signaling game, we can represent the evolution of both the external signaling language *and* the agents' internal representations in the same framework.

The Model

In this section, we present the model used in the experiments for this paper. We first present the model with a "trivial" perception mechanism, which performs no significant cognitive processing. Once we derive the learning rule for this model, we then outline how a non-trivial perception mechanism can be incorporated. The framework is designed to place no restrictions on what perception model we use, and so we will test two mechanisms, representing two schools of thought on modeling cognitive processing.

The Grounded Signaling Game

In the experiments we present here, the state space $\mathcal{D}$ will be a synthetic image environment, presented to the sender as a vector of raw pixel values, and the signal space $\mathcal{X} = \{0,1\}^k$ will consist of binary vectors of length k. As with the Sim-Max game, the receiver's action will be produce a guess d^{out} as to which state d^{in}, the sender initially observes, based on the sender's signal x. The reward function will be a distance metric representing similarity between images.

Recall that a Roth-Erev learning agent requires one parameter for each state-signal pair. In these experiments, however, we will use 36-pixel binary images, and signals between 4 and 36 bits in length. Even though only a small number of possible images $d \in \{0,1\}^{36}$ will ever appear with non-zero probability, the players do not know ahead of time which images are present, or how many appear with positive probability. This is an important distinction, as the receiver must reconstruct the original image pixel-by-pixel, rather than simply guessing from a list of potential images. Therefore, defining a Roth-Erev reinforcement learning agent for this game would require up to 2^{72} separate decision parameters for each player. Clearly this is intractable even for small images, and we must look elsewhere to derive a tractable learning rule. To this end, we draw on the representational flexibility of Artificial Neural Networks (ANNs).
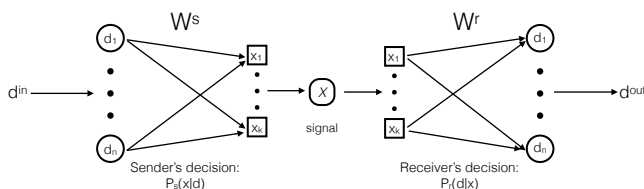

Figure 1: Information flow for one round of the signaling game

Figure 1 depicts a single round of the signaling game. Rather than defining the sender with one decision parameter per state-signal pair, we define a decision rule $P_s(x|d^{in}, W^s)$ with one parameter per state-signal *coordinate* pair. Given a state d, we define, for each signal-coordinate x_i:

$$P_s(x_i = 1|d) = \sigma \left(\sum_{j=1}^{n} W_{i,j}^s d_j \right) \qquad (2)$$

where σ is the standard sigmoid activation function $\sigma(x) = 1/(1 + e^{-x})$. Those familiar with ANNs will note that this is the standard expression for the activation value of a unit with sigmoid activation function. In the stochastic network corresponding to this game, the activation value is treated as a probability, and the output of unit x_i is sampled from the Bernoulli distribution $x_i \sim Bernoulli(P_s(x_i = 1|d))$. Each weight W_{ij}^s determines, in some sense, the "importance" of coordinate j in determining the value of signal component x_i. The sender then generates the signal x by first computing the activation probability for each signal coordinate, then independently sampling each coordinate from the computed Bernoulli distribution. Thus, the sender's probability of sending signal x given object d factors as:

$$P_s(x|d) = \prod_{i=1}^{k} (P_s(x_i = 1|d))^{x_i} (1 - P_s(x_i = 1|d))^{1-x_i} \qquad (3)$$

The receiver's distribution is defined similarly, with the roles of d and x being reversed. With P_s and P_r defined as above, the expected reward function in equation (1) can be interpreted as an objective function $J(W_s, W_r) = E[R|P_s, P_r]$ to a multi-agent optimization problem, where both players wish to maximize $J(W_s, W_r)$, but each player directly controls only one set of parameters. For this experiment in particular, the cooperative objective is to output an image that is most similar to the input image, which is the same objective used in training certain types of auto-encoders (a type of unsupervised ANN, trained to accurately reconstruct its own inputs). Thus, we can efficiently represent a single round of this signaling game as a single forward pass through the three-layer stochastic-sigmoid auto-encoder network shown in figure 1.

The Learning Rule

An auto-encoder, like most feed-forward neural networks, is generally trained using some variation of gradient descent via back-propagation of errors [1]. In each step of the back-propagation algorithm, an input vector is passed through the network, generating a hidden representation (in this case signal) from which the latter half of the network attempts to reconstruct the original input. The error signal (difference between input and output) at each unit is "propagated" backwards, and each weight is adjusted according to its "effect" on the resulting error. Back-propagation algorithms have been extremely successful in training neural networks to perform highly complex tasks, so it is tempting to co-opt the back-propagation algorithm as a "learning rule" for our two agents. However, even though we can represent our signaling game with a three-layer feed-forward network, a key assumption of the model prevents us from using back-propagation directly. In particular, the back-propagation algorithm computes the update to layer l as a function of the parameter and activation

---

[1]Technically, exact back-propagation cannot be applied to a stochastic network, though there are several versions of "approximate" back-propagation for stochastic networks, e.g. (Gu, Levine, Sutskever, & Mnih 2015)

values of layer $l+1$. In this scenario, however, each layer represents a separate human agent, who cannot share parameter information with each other, thus preventing the requisite gradient information from flowing across agents.

Because of this, we instead use a REINFORCE learning rule, first named in Williams (1992). Consider a single round of the signaling game in which sender observes state $d = (d_1, \ldots, d_n)$, sends signal $x = (x_1, \ldots, x_k)$, receiver guesses state $d' = (d'_1, \ldots, d'_n)$, and both players receive reward $R(d, d')$. We define $\Delta W^s_{i,j}$ and $\Delta W^r_{i,j}$, the weight updates for sender and receiver, as

$$\Delta W^s_{i,j} = \varepsilon(R(d,d') - b_{ij})(x_i - P_s(x_i = 1|d))d_j \quad (4)$$

$$\Delta W^r_{i,j} = \varepsilon(R(d,d') - b_{ij})(d'_i - P_r(d'_i = 1|x))x_j \quad (5)$$

where ε is a learning rate and b_{ij} is a *reinforcement baseline*. The main property of REINFORCE rules, as shown in (Williams 1992), is that the weight updates shown in equations (4) and (5) are unbiased estimates of the true gradient of $J(W_s, W_r)$, the expected reward function. That is, $(R(d,d') - b_{ij})(x_i - P_s(x_i = 1|d))d_j$ is an unbiased estimate of $\partial J / \partial W^s_{ij}$, and $(R(d,d') - b_{ij})(d'_i - P_r(d'_i = 1|x))x_j$ is an unbiased estimate of $\partial J / \partial W^r_{ij}$. This allows the two players to cooperatively implement an *approximate* gradient descent algorithm, despite the fact that neither player is explicitly computing any gradients. This rule is both computationally inexpensive and avoids the information-sharing problem of backpropagation, so it is well suited to our task.

Even though equations (4) and (5) are unbiased estimates of the true gradient, they can be very high variance estimates, so outside of very simple tasks, the "pure vanilla" REINFORCE rule (i.e. $b_{ij} = 0$) can be hopelessly slow to converge. We therefore use a minimum variance, unit-specific baseline derived in Bengio (2013), given by the expression.

$$b_{ij} = \frac{E[(h_i - \sigma(a_i))^2 R]}{E[(h_i - \sigma(a_i))^2]} \quad (6)$$

where h_i is the output value and $\sigma(a_i)$ the activation value of unit i. This can be easily computed on the fly by maintaining moving averages of weight updates and rewards over time.

Adding Perception to the Model

The signaling game model we just introduced is "grounded," in the sense that states are represented as sensory inputs, but we have yet to incorporate a perception mechanism. While perception is a broadly defined and widely studied subject, we will adopt a very general stance on what constitutes "perception." We will take perception to be any map $F : \mathcal{D} \to \mathcal{Z}$ from states (represented as sensory inputs) to lower-dimensional internal representations $\mathcal{Z}$. These *internal representations* can be interpreted as the features, categories, concepts, patterns, rules, etc. from which our higher-level decisions are made. For these particular experiments, we shall use perception models that learn both a *recognition* map and a *generative* map. The recognition map $F : \mathcal{D} \to \mathcal{Z}$ infers a latent

representation z from an object d, while the generative map $F^{-1} : \mathcal{Z} \to \mathcal{D}$ generates an object d from an internal representation z (this is a slight abuse of notation, as the generative map will not in general be the inverse of the recognition map).
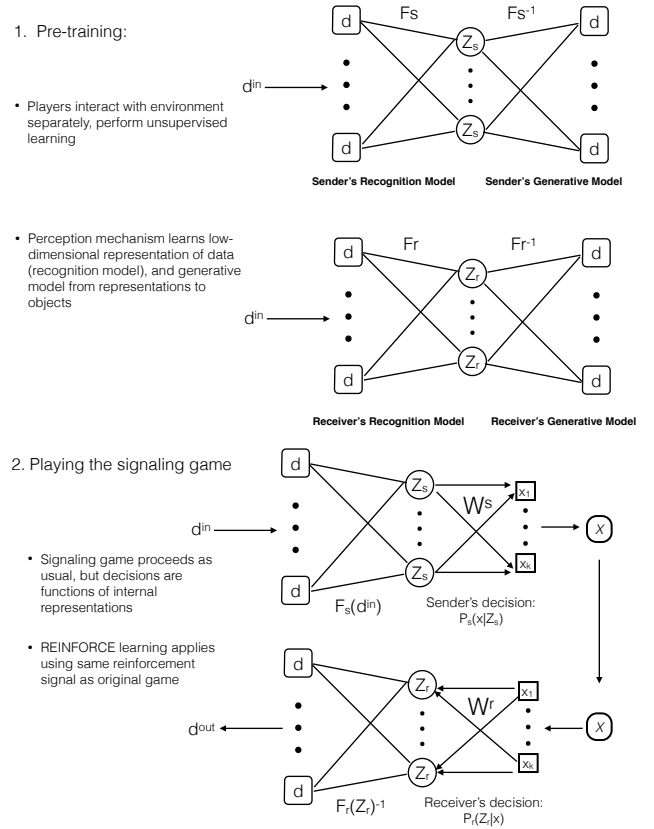

Figure 2: Perception as a pre-processing step prior to signaling game

In the experiments we present here, perception will take the form of a pre-processing step (figure 2). Prior to playing the signaling game, each player will independently sample images from their environment, engaging in unsupervised learning, training their recognition maps F_s, F_r as well as generative maps F_s^{-1}, F_r^{-1}. Once the perception mechanisms have been trained, the signaling game proceeds as usual, except that the sender now makes their signaling decision $P_s(x|z_s^{in}, W^s)$ as a function of the sender's *internal representation* $F_s(d^{in}) = z_s^{in}$ of the state d^{in}. The receiver then observes the signal, and first generates an internal representation z_r^{out}, which is then mapped to output image d_r^{out}. The reward value is then computed as usual, and we apply the same REINFORCE updates in (4)-(5) to the player's *decision parameters*.

Representing Perception

There has been surge of interest in computational models of perception, from both Machine Learning and Cognitive Science. We will test two different models, representing two main approaches that have been taken in studying perception.

The first are Bayesian models, which represent perception as a rational inference problem. Given object d, we infer a latent representation z by maximizing the posterior probability $P(z|d) \propto P(d|z)P(z)$ using Bayes' rule. This requires an object model $P(d|z)$, as well as a prior distribution $P(z)$ over all possible latent representations. We shall use an Infinite Latent Feature Model, which learns a binary feature representation of visual data, without having to pre-define a fixed number of latent features. This is achieved using an Indian Buffet Process (IBP) prior, which defines a probability distribution over binary vectors with an unbounded number of features (Griffiths & Ghahramani 2005). While exact inference over this distribution is intractable, MCMC sampling methods can be used to perform tractable inference.

The second perception mechanism we shall test a Helmholtz Machine (Dayan et al 1995), representing a neuro-computing model of perception. A Helmholtz Machine is a type of variational auto-encoder, which learns a low-dimensional representation of sensory inputs by iteratively inferring latent representations from data, then reconstructing simulated data from internal representations. These two steps are iterated in an alternating "wake-sleep" cycle, with the objective of minimizing the Kullback-Liebler divergence between the true distribution and the generative distribution. The result is a low-dimensional binary representation of the data, encoded on the hidden units of the network. While the Helmholtz Machine in its original form has been rendered largely obsolete by more powerful methods, we use this architecture for its relative simplicity and pedagogical value in the context of our goals.

Experiments and Results

In this section we describe the environments and experimental conditions we tested, present the results of these experiments, and discuss their implications.

Experimental Conditions

We tested three different 36-pixel synthetic image environments, each intended to represent a different kind of informational structure. For the first environment (PICTURE), we defined 8 specific 36-pixel images, each equally likely to appear, and assign 0 probability to all other images. These 8 images were chosen so as to avoid recurring components or features across images. This serves as a baseline evaluation for the perception mechanisms- we can think of the PICTURE environment as representing a "traditional" signaling game setup, in which the "true" state space consists of 8 discrete states that share no common internal structure. In the context of our project, "there are only 8 things here" represents prior knowledge or recognition, and so the players must *learn* that their environment contains only these 8 images directly through their sensory inputs.

The second environment (FEATURE) consists of compositionally distributed images. We define four 3×3 pixel patterns (features), and generate each 6×6 pixel image by randomly selecting any number of the four features and composing them into a single image. This construction is based on an experiment in Griffiths & Ghahramani (2005). The third environment (HIERARCHY) is hierarchically distributed over two categories: there are 12 images with non-zero probability, depicting either horizontal or vertical bars. Images are divided into category A (vertical) and category B (horizontal). Within each category, each image is equally likely to appear, but images from category A are twice as likely to appear as images from category B. The FEATURE and HIERARCHY environment test the agents' abilities to learn non-trivial informational structure from the environment.

For each environment, we test a noiseless version, in which images are presented to the sender with binary pixel values, as well as two levels of corrupting noise, in which each pixel is independently perturbed before being shown to the sender. For a reward function, we test three different distance metrics- Hamming (L_1), Euclidean (L_2), and a patch-specific function that depends only on certain regions of the image.

Results

Figure 3 shows a summary of results across our experiments, using the Hamming metric reward function (we observed no significant differences across reward functions). Convergence rates indicate the number of iterations required to achieve a threshold 90% of optimal performance, averaged over 5 runs for each condition. The *Bayesian* and *Helmholtz* columns correspond to the two perception models, while the *Identify* column indicates the trivial perception mechanism that performs no cognitive processing.

Conditions		Bayesian		Helmholtz		Identity	
Environment	Noise Level[1]	Signal Size[2]	Convergence	Signal Size[2]	Convergence	Signal Size[2]	Convergence
PICTURE	0	8	1.1x10^5	8	1.4x10^5	8	*4.5x10^5
	0.25	8	1.2x10^5	8	1.4x10^5	8	*5.0x10^5
	0.35	8	1.2x10^5	8	1.5x10^5	8	*4.8x10^5
FEATURE	0	4	0.3x10^5	6	0.8x10^5	6	0.8x10^5
	0.25	4	0.4x10^5	6	0.9x10^5	6	0.8x10^5
	0.35	4	0.4x10^5	6	0.9x10^5	6	1.1x10^5
HIERARCHY	0	4	1.0x10^5	4	0.6x10^5	4	1.0x10^5
	0.25	4	1.2x10^5	4	0.8x10^5	4	1.5x10^5
	0.35	4	1.3x10^5	4	0.8x10^5	4	1.3x10^5

[1]Value indicates SD of mean-zero Gaussian noise

[2]Smallest bit-size that converged to optimum (best pooling outcome if optimum not achieved) across all runs

*Indicates partial pooling outcome

Figure 3: Results

In the PICTURE environment, both perception mechanisms drastically improved convergence speed, by effectively reducing the scale of the problem from 2^{36} to 8 states. This pre-processing also smoothed over irregularities that would occur under the trivial perception mechanism, where the receiver would fix the value of certain pixels across all images. Introducing the perception mechanism allows the receiver to reconstruct the image based on their own internal representations of the environment, rather than by individually choosing the value of each output pixel.

In the FEATURE environment, convergence to the optimum was fast and reliable through all levels of noise, even

with the trivial model. The fact that communication is easier to learn in the FEATURE environment than the PICTURE environment, even though the former contains twice as many states as the latter, shows that it is not just the number of distinct states that affects learning, but the content of the states themselves. However, the trivial model was only able to converge to the optimum using a 6-bit signal, which is not minimal. The Bayesian perception mechanism, however, allowed the agents to correctly identify 4 latent features in their environment, which enabled them to learn to communicate using a minimal 4-bit signal. The Helmholtz model did not lead to any reduction in signal size. This is because the Bayesian model learns the number of latent features from the data, while the Helmholtz Machine uses a fixed number of hidden units. Thus the Bayesian model was able to learn a more efficient 4-feature representation than the network-based model, enabling more efficient communication.

In the HIERARCHY environment, convergence was fast and reliable under all 3 models, using a minimal 4-bit signal. The Helmholtz model is able to learn the more efficient representation in this environment, using one hidden unit to code for the category, and 6 more for each image within a category. This reduced convergence time by up to half. The Bayesian model, however, learns a less efficient representation, identifying 1 binary feature for each of the 12 images in the environment, and does not significantly improve convergence speed.

Discussion and Future Work

The results from the previous section demonstrate how a perception mechanism can be incorporated into a signaling game model, and shed some light on the first interaction we raised in the introduction. In particular, we saw that a perception pre-processing step can enable faster and more robust cooperative learning from high-dimensional sensory inputs. We also saw that certain perception models can learn more efficient representations in certain environments. In this section, we discuss how the existing model can be extended to address the other types of interactions we wish to study.

Other Roles of Perception

In the experiments presented here, perception was used strictly as a pre-processing step, but in order to better understand how language-learning can affect perception, we must allow the perception mechanisms to be trained in parallel with the signaling game. While the REINFORCE rule we present here does not scale well to very deep models with multiple hidden layers, recent advances in Deep Reinforcement Learning and Deep Q-learning allow us to scale the basic architecture up to very large tasks. Additionally, we may be interested in fixing the behavior of one agent, so that the non-fixed player learns the language of the fixed player. This would allow us to observe any influence that the fixed player's language has on the non-fixed player's learned representations.

Perceptual Similarity in Communication

The model we present here already has most of what we need to address the third type of interaction, relating to perceptual similarity across agents. That is, we already represent the evolution of both the external language and the internal representations, so all we need is a means of quantifying "perceptual similarity" across multiple agents. To this end, we can use cross-systems analysis techniques like Representational Similarity Analysis (Kriegeskorte, N., Mur, M., & Bandettini, P. A. 2008), a method for quantifying "representational similarity" between two different representation systems, regardless of the underlying topologies of the systems themselves. This would allow us to study inter-agent learning performance as a function of the similarity between their internal representations, and perhaps identify a "communicability threshold" of perceptual similarity below which no communication is possible.

References

Bengio, Y., Lonard, N., & Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Dayan, P., Hinton, G. E., Neal, R. M., & Zemel, R. S. (1995). The helmholtz machine. *Neural computation*, 7(5), 889-904.

Franke, M. (2015). The evolution of compositionality and proto-syntax in signaling games. Journal of Logic, Language and Information, forthcoming.

Griffiths, T. L., & Ghahramani, Z. (2005). Infinite latent feature models and the Indian buffet process. In *NIPS* (Vol. 18, pp. 475-482).

Gu, S., Levine, S., Sutskever, I., & Mnih, A. (2015). MuProp: Unbiased backpropagation for stochastic neural networks. *arXiv preprint arXiv:1511.05176*.

Huttegger, S. M., Skyrms, B., Smead, R., & Zollman, K. J. (2010). Evolutionary dynamics of Lewis signaling games: signaling systems vs. partial pooling. *Synthese*, 172(1), 177-191.

Jager, G. (2007). The evolution of convex categories. *Linguistics and Philosophy*, 30(5), 551-564.

Kriegeskorte, N., Mur, M., & Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2, 4.

O'Connor, C. (2014). Evolving perceptual categories. *Philosophy of Science*, 81(5), 840-851.

Skyrms, B. (2010). *Signals: Evolution, learning, and information*. Oxford University Press.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4), 229-256.