End-to-End Models for Task-Oriented Gameplay with Gated-Attention Networks and Malliavin-Stein Variational Policy Gradients

Ali Zaidi
Stanford University
alikazim@stanford.edu

Abstract

In order to perform tasks and challenges specified by natural language instructions, autonomous agents need to extract semantically meaningful representations of language and map it to the visual elements of their scene and into actions in the environment. This is often referred to as task-oriented language grounding. In this paper, we propose to directly map raw visual observations and text input into actions for instruction execution, using an end-to-end trainable neural architecture. The model synthesizes image and text representations using Gated-Attention mechanisms and learns a policy using Malliavin-Stein Variational gradient descent to execute the natural language instruction. This approach does not require intermediate representations or planning procedures, and can be used to learn hierarchical tasks and challengs. We evaluate our method in Minecraft to the problem of retrieving items in unseen maps and mazes and show improvements over supervised and common reinforcement learning algorithms.

1 Introduction

Reinforcement learning has made incredible progress in the past few years, in large parts due to the effectiveness of neural networks as value function approximators, and their ability to be trained end-to-end using stochastic optimization. Much of the exciting work in this area has occurred in scenarios where the agent can learn directly from the environment exploration. In this work, we examine the ability to define differentiable architectures that can be trained end-to-end using stochastic optimization for solving task-oriented language grounding problems. These are problems where the agent needs to solve specific tasks defined through natural langauge interactions. In order to succeed in such applications, the agent needs to extract semantically meaningful representations of language that they can map to meaningful visual elements and actions in the environment. In particular, to accomplish goals defined through instructions, the agent needs to draw semantic correspondences between the visual and linguistic modalities and learn a policy to perform the task.

To tackle this problem, we propose an architecture that consists of a *state representation module* that provides a joint representation of the natural language instruction and the visual scene observed by the agent's view, and a *policy learning module* to predict the optimal action for the agent in the next timestep t. This state representation module is defined using a Gated-Attention network, and the policy learning module is created using Malliavin-Stein Variational policy gradient. The main contributions of this paper are 1) a novel way to train end-to-end neural architectures that takes raw pixel input of the current scene and natural language instructions and assumes no intermediate representations or prior knowledge 2) a novel Gated-Attention mechanism for multimodal fusion of linguistic and visual modalities for training policy functions, and 3) an open-source implementation of langauge grounding in the popular Minecraft game using the open source framework Project Malmo [JHHB16]. Figure 1 provides an example of task-oriented gameplay in this environment.

2 Network Overview

Let $\mathcal{X}$ denote the set of natural language instructions, and $\mathcal{S}$ the set of states, and $\mathcal{A}$ the set of all actions. An instruction $\boldsymbol{x} \in \mathcal{X}$ is a sequence $\{x_1, \ldots, x_n\}$ where each x_i is a token of the provided sentence. Action execution modifies the state of the world which follows a transition function $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$.

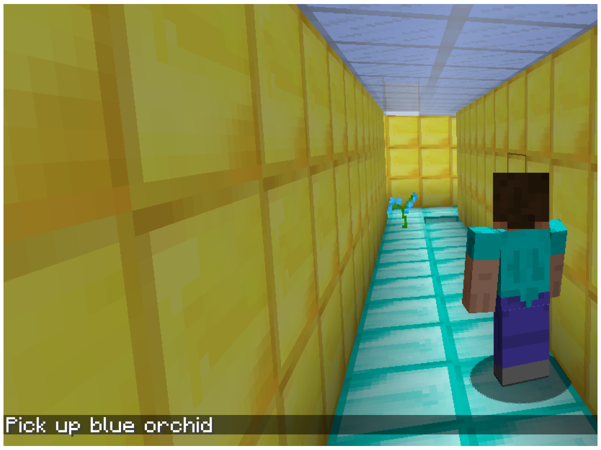

Figure 1: Task-oriented grounding in a maze environment. Here the agent is provided with natural language instructions to pick up a certain item but also needs contextual visual information to avoid the dangerous missing block.

To map instructions to actions, the agent reasons about the context to generate a set of actions. We model the agent with a neural network policy π. At each step j, the network takes as input the current agent context and predicts the next action to execute a_j. Concretely, at time step j, the agent considers a context vector c_j which is a tuple of the instruction, current image and previous K images, and the last action, $(\boldsymbol{x}, I_j, I_{j-1}, \ldots, I_{j-K}, a_{j-1})$. We then generate continuous vector representations of all inputs to jointly reason about text and vision modalities in constructing the next action. To model the visual input stream obtained from the agent's current perspective, we used ideas inspired by the DQN framework [MKS+15], where for each iteration j, the previous K images are concatenated along the channel dimension and embedded with a convolutional neural network to generate a the visual state $\boldsymbol{v}_j$. In addition, the instruction sequence $\boldsymbol{x}$ is modeled using a LSTM to map the sequence to a vector representation $\bar{\boldsymbol{x}}$. Each token x_i is mapped to a fixed dimensional vector with learned embedding function $\phi(x_i)$. In the Gated-Attention unit, the instruction embedding is passed through a fully-connected linear layer with a sigmoid activation function. The instruction representation is then computed by applying the LSTM recurrence to generate a sequence of hidden states $\boldsymbol{l}_i = \mathrm{LSTM}(\phi(x_i), \boldsymbol{l}_{i-1})$. These context vectors, $\boldsymbol{v}_j, \bar{\boldsymbol{x}}, \phi(a_{j-1})$ are all concatenated to create a multimodal agent context vector representation $\boldsymbol{c}_j$.

2.1 Learning Policies with Malliavin-Stein Policy Gradients

Initial breakthroughs in representation learning have mostly been driven by supervised learning methods, yielding state-of-the-art results in a variety of domains where there is access to large labelled datasets. On the other hand, probabilistic approaches for unsupervised generative models [ML16] provide a principled approach towards reasoning and inference, but are inherently more difficult to scale. Generative Adversarial Networks (GANs) [GPAM+14] are an algorithmic approach for generative modeling, providing a direct estimation procedure to compute samples from implicit models. The original formulation of GANs replaced minimizing Jenson-Shannon expectations directly with minimizing samples averages based on an approximation of a variational lower-bound represented by the discriminator. When the function classes being used in this game-formulation are very high-dimensional, uniform laws of large numbers may no longer apply directly, leading to undefined density ratios between the model distribution and the true data generating distribution [Goo16]. This can have a dramatic impact on the performance of GANs in practice, and requires careful configuration of the GAN hyper-parameters, such as the network architecture, optimization procedure and learning schedules, and parameter initialization. In the past few years, a variety methods have been proposed for generating samples using GANs, based on integral probability measures [ACB17], proper scoring rules, and f-divergences [NCT16].

In order to optimize our policy formulation, we utilize the theory of Malliavin calculus and the Stein-Langevin discrepancy measure as a principled approach for generative sampling in implicit models. Our approach combines the key ideas from GANs and kernel-Stein discrepancies [GM15] and provides a discrepancy measure that is differentiable and can be optimized using stochastic gradient descent.

2.2 Distances on Probability Measures

There are a variety of ways of defining similarity between probability measures P_X and P_G . Three common approaches are the method of f-divergences, integral probability metrics, and optimal transport. For f-divergences, the basic goal is similar to importance sampling: take any convex $f : (0, \infty) \mapsto \mathbb{R}$ with $f(1) = 0$, and define

$$D_f(P \| Q) := \int_{\mathcal{X}} f\left(\frac{p(x)}{q(x)}\right) q(x)\,\mathrm{d}x.$$

Within the f-divergences approach, the goal is to minimize $D_f(P_X \| P_G)$ with respect to P_G. The common formulation is the variational dual representation of f-divergences:

$$D_f(P\|Q) = \sup_{T:\mathcal{X}\rightarrow \mathrm{dom}(f^\star)} \mathbb{E}_{X\sim P}[T(X)] - \mathbb{E}_{Y\sim Q}[f^\star(T(Y))],$$

where

$$f^\star(x) := \sup_u x\cdot u - f(u)$$

is a convex conjugate of f.

In contrast, integral probability metrics (IPMs) forms a uniform bound between the two measures in expectation:

$$\gamma_{\mathcal{F}}(P,Q) := \sup_{f\in\mathcal{F}} |\mathbb{E}_P[f(X)] - \mathbb{E}_Q[f(Y)]|.$$

Lastly, in optimal transport we examine the expected cost of the joint distributions between samples from the probability measures:

$$W_c(P,Q) := \inf_{\Gamma\in\mathcal{P}(X\sim P, Y\sim Q)} \mathbb{E}_{(X,Y)\sim\Gamma}[c(X,Y)],$$

where $\mathcal{P}(X\sim P, Y\sim Q)$ is the set of all joint distributions of (X,Y) with marginals P and Q respectively.

3 Malliavin-Stein Operators

3.1 Stein's Method

In an attempt to teach undergraduates the central limit theorem without resorting to characteristic functions and Fourier analysis, Charles Stein [Ste72] introduced a novel new approach for bounding a reference IPM γ_H.

1. First, identify an operator $\mathcal{T}$ that maps input functions $g:\mathbb{R}^d\rightarrow\mathbb{R}^d$ in a domain $\mathcal{G}$ into mean-zero functions under P:

   $$\mathbb{E}_P[(\mathcal{T}g)(Z)] = 0, \quad \text{for all } g\in\mathcal{G}.$$

   The pair $\mathcal{T}$ and $\mathcal{G}$ define the **Stein discrepancy**

   $$\mathcal{S}(Q,\mathcal{T},\mathcal{G}) := \sup_{g\in\mathcal{G}} |\mathbb{E}_Q[(\mathcal{T}g)(X)]| = \gamma_{\mathcal{T}\mathcal{G}}(Q,P),$$

   which is an IPM-measure with no explicit integration under P.

2. Form a lower bound for the Stein-discrepancy by a reference measure $\gamma_{\mathcal{H}}(Q,P)$, where for each test function $h\in\mathcal{H}$ the Stein-equation

   $$h(x) - \mathbb{E}_P[h(Z)] = (\mathcal{T}g_h)(x)$$

   admits a solution $g_h\in\mathcal{G}$.

3. Upper bound the Stein-discrepancy by any means necessary.

3.2 Amortized Stein Variational Gradient Descent with Malliavin Gradient Flow

In [WL16], the authors utilized the dynamics of Stein variational gradient descent to approximate the target data distribution. In their paper, the the generative model for the model distribution is trained to approximately sample from $p(x|\theta)$ and used to calculate the gradient flow for the loss:

$$\hat{\nabla}_\theta L(\theta) = -\hat{\mathbb{E}}_r[\partial_\theta\phi(x;\theta)] + \hat{\mathbb{E}}_\eta[\partial_\theta\phi(x;\theta)].$$

Intuitively, the generator is averaging over the observed data and the expectations under the model distribution.

The main contribution of this paper is to utilize a second-order approximation of the log-likelihood gradient using Hermite polynomials and the Malliavin calculus,[Nua12]. A priori, this approach would be computationally infeasible due to the need to calculate the Hessian matrix of the log-likelihood at every mini-batch. However, using Hermite polynomials we can approximate the second order terms directly:

$$\hat{\nabla}_\theta L(\theta) = -\hat{\mathbb{E}}_r[\partial_\theta\phi(x;\theta)] + \hat{\mathbb{E}}_\eta[\partial_\theta\phi(x;\theta)] + \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N}\theta\varphi_k\phi_i(x_i;\theta)$$

4 Training and Results

During training, we provide our agent with differing levels of supervision and labelled examples, ranging from imitation learning, where the agent is given examples where the location of the item paired with the instructions in known, to semi-supervised demonstrations where annotations are provided for the goal state only. For experiments, we provide our agent with instructions to pick up specific items in a maze, where the maze penalty-states that terminate the agent's life (i.e., falling down a a hole, or picking the wrong item). We examined our architecture at varying distances and with varying number of items which are shown in Table 1. The maximum reward is 1, and the numbers shown in the table are an average of 100 runs for each experiment. The experiments differ from how far the player's initial position is from the item to be retrieved, and from the number of occluding items there are in the player's path. Our initial results show that the model is fairly robust to the distance of the reward, but less robust when occluding items are placed in the agent's path.

	Number of items		
Distance	2	5	10
10	0.92	0.85	0.65
15	0.92	0.84	0.63
30	0.87	0.83	0.53
50	0.86	0.78	0.45
100	0.86	0.73	0.33

Table 1: Mean reward test results

References

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou, *Wasserstein generative adversarial networks*, Proceedings of the 34th International Conference on Machine Learning (International Convention Centre, Sydney, Australia) (Doina Precup and Yee Whye Teh, eds.), Proceedings of Machine Learning Research, vol. 70, PMLR, 06–11 Aug 2017, pp. 214–223.

[GM15] Jackson Gorham and Lester Mackey, *Measuring sample quality with Stein's method*, 226–234.

[Goo16] Ian J. Goodfellow, *Nips 2016 tutorial: Generative adversarial networks*, NIPS **abs/1701.00160** (2016).

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, *Generative adversarial nets*, Advances in Neural Information Processing Systems 27 (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), Curran Associates, Inc., 2014, pp. 2672–2680.

[JHHB16] Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell, *The malmo platform for artificial intelligence experimentation*, Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, 2016, pp. 4246–4247.

[MKS+15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis, *Human-level control through deep reinforcement learning*, Nature **518** (2015), no. 7540, 529–533.

[ML16] Shakir Mohamed and Balaji Lakshminarayanan, *Learning in implicit generative models*, CoRR **abs/1610.03483** (2016).

[NCT16] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka, *f-gan: Training generative neural samplers using variational divergence minimization*, Advances in Neural Information Processing Systems 29 (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), Curran Associates, Inc., 2016, pp. 271–279.

[Nua12] D. Nualart, *The malliavin calculus and related topics*, Jan 2012.

[Ste72] Charles Stein, *A bound for the error in the normal approximation to the distribution of a sum of dependent random variables*, Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory (Berkeley, Calif.), University of California Press, 1972, pp. 583–602.

[WL16] Dilin Wang and Qiang Liu, *Learning to draw samples: With application to amortized MLE for generative adversarial learning*, CoRR **abs/1611.01722** (2016).