Toward Understanding Catastrophic Interference in Value-based Reinforcement Learning

Vincent Liu[1], Hengshuai Yao[2], Martha White[1]
[1]Department of Computing Science, University of Alberta
{vliu1,whitem}@ualberta.ca
[2] Huawei Technologies
hengshuai.yao@huawei.com

Abstract

We study catastrophic interference in reinforcement learning. Catastrophic interference is typically considered for multi-task learning. However, in reinforcement learning, it could occur even in a single-task setting. To better understand catastrophic interference, we aim to quantify interference in reinforcement learning. Finally, we empirically evaluate the proposed measure of interference in a classic reinforcement learning environment.

1 Introduction

Generalization is a key issue in function approximation. It is important for an agent to generalize from previous encountered samples to a larger subset of samples which have not been seen. Generalization has been extensively studied in supervised learning, where we normally assume that we can sample i.i.d. inputs $\{x_i\}_{i=1}^N$ from a fixed input distribution and the targets $\{y_i\}_{i=1}^N$ are sampled from a fixed conditional distribution. Therefore, we can use the Empirical Risk Minimization (ERM) to find a solution $\theta^* = \arg\min_\theta \sum_{i=1}^N J(\theta; x_i, y_i)$ where J is an objective function and θ is the parameter.

The assumption of i.i.d. inputs, however, does no hold in general. For example, in multi-task learning, the agent continually faces new tasks. While learning on a new task, the learner can forget previously learned information. This issue is called *catastrophic interference*. Catastrophic interference is typically considered for multi-task learning [Kirkpatrick et al., 2017, Riemer et al., 2018]. In reinforcement learning (with function approximation), it could occur even in a single-task setting [Goodrich, 2015, Ghiassian et al., 2018] since (a) when an agent explores an environment, it receives a sequence of observations, which are likely to be temporally correlated; (b) the agent is changing its policy while learning, which makes the sequence of observations non-stationary; and (c) the agent uses its own estimates as targets, which makes the target outputs non-stationary. If estimates change incorrectly due to interference, there could be a cascading effect.

There are several works on quantifying and evaluating interference in reinforcement learning. However, most of the works study interference between multiple tasks (or multiple objectives) [Riemer et al., 2018, Schaul et al., 2019]. In this paper, we aim to quantify interference in a single-task setting, and investigate how it affects the stability and the control performance.

2 Background

In reinforcement learning (RL), an agent interacts with its environment, receiving observations and selecting actions to maximize a reward signal. We assume the environment can be formalized as a Markov decision process (MDP). An MDP is a tuple $(\mathcal{S}, \mathcal{A}, \Pr, R, \gamma)$ where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is

Optimization Foundations for Reinforcement Learning Workshop at NeurIPS 2019, Vancouver, Canada.

an set of actions, Pr : $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, and γ is the discount factor $\in [0, 1]$ which define the relative value of future rewards.

Given a policy π, the value function is defined as

$$V^{\pi}(s) := \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right],\ \forall s \in \mathcal{S},$$

and the optimal value function is defined as

$$V^{*}(s) := \max_{\pi} V^{\pi}(s),\ \forall s \in \mathcal{S}.$$

We define the Bellman optimality operator $T : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ as:

$$(TV)(s) := \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \Pr(s, a, s')[R(s, a, s') + \gamma V(s')].$$

Note that V^* is the unique solution of the bellman equation $TV = V$. Therefore, to obtain the optimal value function, we aim to find the fixed point of the Bellman optimality operator. If the environment is deterministic, the optimality bellman operator T can be simply written as

$$(TV)(s) = \max_{a' \in \mathcal{A}} R(s, a, s') + \gamma \max_{a'} V(s').$$

3 Defining Pairwise Interference

We begin by discussing interference in supervised learning, where we consider the pairwise interference between two samples. When we perform an update based on one sample, the update can generalize to another sample positively (positive generalization), negatively (interference), or no effect (no generalization). Then we extend the concept to reinforcement learning.

3.1 Quantifying Interference in Supervised Learning

In supervised learning, we consider a well-defined pointwise loss function J with parameter θ, and we aim to find a parameter which minimize the expected loss

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim p}[J(\theta; x, y)]$$

where p is a (fixed) data distribution.

We consider pairwise interference between two samples. Suppose we perform a SGD update with the sample (x_t, y_t) at time t:

$$\theta_{t+1} = \theta_t - \alpha \nabla_{\theta_t} J(\theta_t; x_t, y_t)$$

where α is a step size. Given the update based on (x_t, y_t), we define the *pairwise interference* as the change in the loss function for another sample (x_i, y_i) by

$$PI(\theta_t; (x_t, y_t), (x_i, y_i)) := J(\theta_{t+1}; x_i, y_i) - J(\theta_t; x_i, y_i). \tag{1}$$

and define the *expected interference* by

$$EI(\theta_t; (x_t, y_t)) := \mathbb{E}_{(x_i,y_i)\sim p}[PI(\theta_t; (x_t, y_t), (x_i, y_i))]. \tag{2}$$

Intuitively, EI quantifies how much the loss changes in expectation given one update on the parameter. If EI is negative, we can expect the loss function decreases in expectation, so it generalizes positively. If EI is positive, the loss function increases in expectation, so interference occurs. The update on (x_t, y_t) results in unlearning of other samples in expectation.

We can approximate equation (1) by a Taylor expansion:

$$\begin{aligned} PI(\theta_t; (x_t, y_t), (x_i, y_i)) &\approx \nabla_{\theta_t} J(\theta_t; x_i, y_i)^\top (\theta_{t+1} - \theta_t) \\ &= -\alpha \nabla_{\theta_t} J(\theta_t; x_i, y_i)^\top \nabla_{\theta_t} J(\theta_t; x_t, y_t). \end{aligned} \tag{3}$$

This analysis provides some insight into previous definition of interference and generalization. Riemer et al. [2018] define pairwise measures of transfer and interference between two examples (x_t, y_t) and (x_i, y_i). If

$$\nabla_{\theta_t} J(\theta_t; x_t, y_t)^\top \nabla_{\theta_t} J(\theta_t; x_i, y_i) > 0$$

then transfer occurs. If

$$\nabla_{\theta_t} J(\theta_t; x_t, y_t)^\top \nabla_{\theta_t} J(\theta_t; x_i, y_i) < 0$$

then interference occurs. In parallel with our work, Fort et al. [2019] define stiffness as

$$\mathbb{E}[\text{sign}(\nabla_{\theta_t} J(\theta_t; x_t, y_t), \nabla_{\theta_t} J(\theta_t; x_i, y_i))] \text{ or } \mathbb{E}[\cos(\nabla_{\theta_t} J(\theta_t; x_t, y_t), \nabla_{\theta_t} J(\theta_t; x_i, y_i))].$$

3.2 Quantifying Interference in Policy Evaluation

Given a policy π, we denote Q^π as the true action-value function and Q_θ as the estimated action-value function, parameterized by the parameter θ. Similar to supervised learning, we have an loss function for policy evaluation. We want to minimize the mean square value error (MSVE), defined as:

$$\mathbb{E}_{s \sim d^\pi(\cdot), a \sim \pi(\cdot|s)}[J(\theta; s, a)]$$

where

$$J(\theta; s, a) := (Q^\pi(s, a) - Q_\theta(s, a))^2$$

and d^π is the stationary distribution under π.

In reinforcement learning, we usually use some approximation to the true value as our target. We denote the target of the t-th example (s_t, a_t) by $U_t \in \mathbb{R}$. When we perform an update on (s_t, a_t), the update is computed by

$$\theta_{t+1} = \theta_t + \alpha \delta(s_t, a_t) \nabla_{\theta_t} Q_{\theta_t}(s_t, a_t)$$

where $\delta(s_t, a_t) = U_t - Q_{\theta_t}(s_t, a_t)$. Similar to equation (3), the pairwise interference for between two samples can be approximated by

$$\begin{aligned} PI_t(\theta_t; (s_t, a_t), (s_i, a_i)) &:= J(\theta_{t+1}; s_i, a_i) - J(\theta_t; s_i, a_i) \\ &\approx \alpha \delta(s_t, a_t) J(\theta_t; s_i, a_i) \nabla_{\theta_t} Q_{\theta_t}(s_t, a_t)^\top \nabla_{\theta_t} Q_{\theta_t}(s_i, a_i). \end{aligned} \tag{4}$$

and

$$EI(\theta_t; (s_t, a_t)) := \mathbb{E}_{s \sim d^\pi(\cdot), a \sim \pi(\cdot|s)}[PI(\theta_t; (s_t, a_t), (s, a))].$$

The term $\nabla_{\boldsymbol{\theta}_t} Q_{\boldsymbol{\theta}_t}(s_t, a_t)^\top \nabla_{\boldsymbol{\theta}_t} Q_{\boldsymbol{\theta}_t}(s_i, a_i)$ is the neural tangent kernel (NTK) [Jacot et al., 2018] of the Q function, which has been used to analyze generalization in the Q function across state-action pairs [Achiam et al., 2019]. However, to determine whether positive generalization or interference occurs, we also need to know $\delta(s_t, a_t)$ and $J(\theta_t; s_i, a_i)$.

The PI provides a way to think about interference in reinforcement learning. When we update the value of a state which has high PI with many other states, the update might cause instability in training. In linear function approximation, to make PI small, we want orthogonal feature vectors. If we force the feature vectors to be non-negative, then they are likely to be sparse. The insight matches the finding in Liu et al. [2019] that sparse representation can mitigate interference in reinforcement learning.

3.3 Quantifying Interference in Control

For control tasks, we aim to minimize the performance loss of the greedy policy π_θ with respect to the current estimation Q_θ:

$$\arg\min_\theta \mathbb{E}_{s \sim \nu}[J(\theta; s)]$$

where

$$\begin{aligned} J(\theta; s) &:= (Q^*(s, \pi^*(s)) - Q^{\pi_\theta}(s, \pi_\theta(s)))^2 \\ &= (V^*(s) - V^{\pi_\theta}(s))^2 \end{aligned}$$

and ν is a performance-measuring distribution on $\mathcal{S}$ [Farahmand, 2011], which is typically chosen as the initial state distribution.

Given a update $\Delta\theta_t = \theta_{t+1} - \theta_t$ on the parameter, we can define expected interference, similar to equation (2)[1], as

$$EI(\theta_t; \Delta\theta_t) := \mathbb{E}_{s\sim v}[J(\theta_t + \Delta\theta_t; s) - J(\theta_t; s)].$$

This performance loss, however, is difficult to compute because both V^* and $V^{\pi_{\theta_t}}$ are unknown[2]. Fortunately, there are some upper bounds on the performance loss [Williams, 1993, Farahmand et al., 2010, Munos, 2007]. Here, we adapt the bound derived in Munos [2007].

Definition 1 (State Distribution Concentration Coefficient) *Let $\pi_1, ..., \pi_m$ be a sequence of policies. For any integer $m \geq 1$, define $c(m) \in \mathbb{R}^+ \cup \{\infty\}$ by*

$$c(m) = \max_{\pi_1,...\pi_m, y\in\mathcal{S}} \frac{(\nu P^{\pi_1}...P^{\pi_m})(y)}{\mu(y)}.$$

(let $c(m) = \infty$ if $\nu P^{\pi_1}...P^{\pi_m}$ is not absolutely continuous w.r.t μ). We define $c(0) = 1$. Moreover, we define $C_1(\nu, \mu) \in \mathbb{R}^+ \cup \{\infty\}$, the discounted future state distribution concentration coefficients, by

$$C(\nu, \mu) := (1-\gamma)\sum_{m=0}^{\infty}\gamma^m c(m).$$

Theorem 1 (Performance Bound on the Bellman residual) *Let $Q \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, π be a greedy policy with respect to Q, i.e. $\pi(s) = \operatorname{argmax}_a Q(s, a)$, and $V(s) = Q(s, \pi(s))$ for all $s \in \mathcal{S}$. Let ν and μ be two probability measures on $\mathcal{S}$. Then*

$$||V^* - V^\pi||_\nu \leq \frac{2}{1-\gamma}[C(\nu,\mu)]^{1/2}||TV - V||_\mu,$$

where $||V||_d$ is a d-weighted l_2-norm of the vector V. Moreover, for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$,

$$(TV)(s) - V(s) = (TQ)(s, \pi(s)) - Q(s, \pi(s)).$$

By choosing ν as the initial distribution and μ as the stationary distribution induced by π_{θ_t}, we can bound the performance loss on the Bellman residual. Therefore, we define the expected interference in term of the Bellman Residual by

$$EI(\theta_t; \Delta\theta_t) := \mathbb{E}_{s\sim u, a\sim\pi(\cdot|s)}[\hat{J}(\theta_t + \Delta\theta_t; s, a) - \hat{J}(\theta_t; s, a)] \tag{5}$$

where

$$\hat{J}(\theta; s, a) = ((TQ_\theta)(s, a) - Q_\theta(s, a))^2.$$

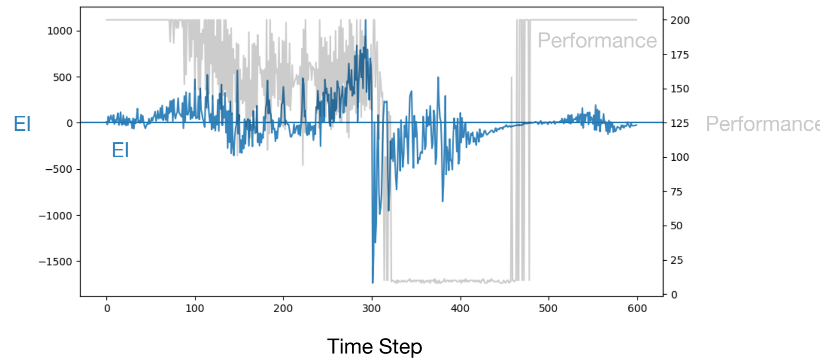
When EI is negative, the performance bound decreases and generalization occurs. When it is positive, then interference might occur. Note that the bellman residual is zero for all state-action pairs if and only if $Q_\theta = Q^*$.

Approximate the Bellman operator In general, it would require multiple samples to compute the bellman residual [Baird, 1995]. However, in our experiment, we only test on deterministic environments where $(TQ_{\theta_t})(s, a)$ can be computed with one sample transition. In stochastic environments, we can use a model of the environment to compute the value.

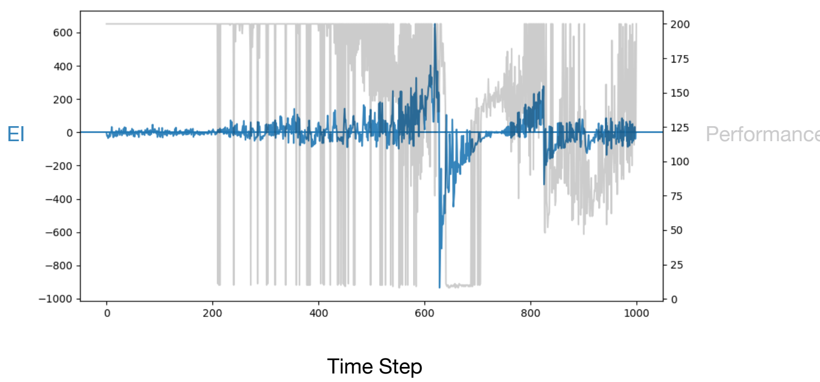
Approximate the expectation Another question is how to compute the expectation in Equation 5. If we have a model of the environment, we could run the policy π_{θ_t} to collect transitions to approximate the expectation. A more practical method is to keep a buffer of recent transitions and hope the policy does not change much.

[1] Instead of taking a sample (x_t, y_y) as an input, we abuse the notation to take a change on the parameter as an input here.

[2] In fact, we can estimate $V^{\pi_{\theta_t}}$ by Monte Carlo rollouts



(a)



(b)

Figure 1: Examples of catastrophic interference in cart-pole. The performance is evaluated offline. We report the sum of discounted rewards per episode.

4 Experimental Results

In this section, we aim to answer the following questions: can we quantify interference? We run a simple experiment in cart-pole, a classic reinforcement learning environment. We use a Q-learning agent with a two-layer neural network with hidden size 256 and a replay buffer of size 100. We use a small buffer to enable learning while allowing interference to occur, since a sufficiently large buffer might prevent catastrophic interference[3] [Lin, 1993]. At each time step, we evaluate equation (5) on a buffer containing recent transitions of size 1000. We hope the state distribution in the buffer is approximately close to u.

Figure 1 shows the learning curve of the Q-learning agent, we can see that the performance starts to oscillate when EI started to increase (e.g. t=100 in figure 1.a and t=500 in figure 1.b), and the perform drops when EI increases significantly (e.g. t=300 in figure 1.a and t=600 in figure 1.b). That is, catastrophic interference occurs when EI increases significantly. This result provides some evidences that EI can be used to quantify interference, and it is correlated with the stability and control performance.

5 Discussion

In this paper, we propose a measure to quantify positive generalization and interference in reinforcement learning, and we empirically evaluate the measure in a classic reinforcement learning

[3]Note that the goal of the paper is to study this phenomenon, not to propose a new algorithm to solve it.

environment. This work is a first-step investigation toward understanding catastrophic interference in reinforcement learning. There are still several open questions we could not answer in the paper. For example, the bound we use in section 3.2 might be loose in deterministic environments, so we would need to analyze how tight the bound is. In section 5.4, we only provide qualitative evidence that the measure is correlated with the control performance. However, quantitative evidences are needed.

References

Joshua Achiam, Ethan Knight, and Pieter Abbeel. Towards characterizing divergence in deep q-learning. *arXiv:1903.08894*, 2019.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

Amir-massoud Farahmand. Regularization in reinforcement learning. 2011.

Amir-massoud Farahmand, Csaba Szepesvári, and Rémi Munos. Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems*, 2010.

Stanislav Fort, Paweł Krzysztof Nowak, and Srini Narayanan. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*, 2019.

Sina Ghiassian, Huizhen Yu, Banafsheh Rafiee, and Richard S Sutton. Two geometric input transformation methods for fast online reinforcement learning with neural nets. *arXiv:1805.07476*, 2018.

Benjamin Frederick Goodrich. Neuron clustering for mitigating catastrophic forgetting in supervised and reinforcement learning. 2015.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, 2018.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

Long-Ji Lin. Reinforcement learning for robots using neural networks. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1993.

Vincent Liu, Raksha Kumaraswamy, Lei Le, and Martha White. The utility of sparse representations for control in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4384–4391, 2019.

Rémi Munos. Performance bounds in l_p-norm for approximate value iteration. *SIAM journal on control and optimization*, 2007.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv:1810.11910*, 2018.

Tom Schaul, Diana Borsa, Joseph Modayil, and Razvan Pascanu. Ray interference: a source of plateaus in deep reinforcement learning. *arXiv:1904.11455*, 2019.

Ronald J Williams. Tight performance bounds on greedy policies based on imperfect value functions. Technical report, Citeseer, 1993.