Maybe a few considerations in Reinforcement Learning Research?

Anonymous Authors

Abstract
Recent advances in computation power accessible to machine learning researchers sparked the flurry of research interest in large scale Reinforcement Learning (RL). However, any theoretical study as well as empirical investigation of any innovative RL algorithm requires an extensive amount of experts’ time and imposes intolerably massive computation cost. These challenges become even more vital in the percent of the significantly limited resources dedicated to RL research, including human-time and computation resources. These hardnesses and enormous costs impel an immediate consideration and dedication in guiding our research development direction and efficiently allocate our resources to advance RL in practice. We devote this paper to discuss a set of essential issues that are necessary to be addressed in order to provide greater supervisions in conducting future RL studies.

1. Introduction
RL as the study of sequential decision making under uncertainty is the core of data-driven approaches in real-world applications. RL applications abounds in real-world problems ranges from healthcare (Thompson, 1933) and recommendation systems (Li et al., 2010) to autonomous vehicles (Lange et al., 2012), dialogue systems (Singh et al., 2000), and robotics (Argall et al., 2009). This wide range of application has made RL an important topic of interest for many decades. Researcher have made many principled and heuristic contribution in this field (Lai & Robbins, 1985; Lattimore & Szepesvári, 2018; Brafman & Tennenholtz, 2002; Auer et al., 2002; Jaksch et al., 2010; Kocsis & Szepesvári, 2006; Sutton, 1990) and have shed lights on the essence of RL.

Recent advances in deep neural networks and computation allowed the machine learning community to develop deep learning methods for RL problems and propose an extensive set of deep RL (DRL) algorithms. Among the first best-publicized methods, deep Q-network (DQN) (Mnih et al., 2015) tackles video games in Atari Learning Environment (Bellemare et al., 2013), AlphaGo addresses board games (Silver et al., 2016), and Deep Visuomotor Policies to approach robotic problems (Levine et al., 2016).

These all are promising for further improvements in RL.

Despite all the principled and empirical advances in RL, this problem is shockingly barely understood and deserves much more attention from theoreticians and practitioners.

Maybe a sad news: The study of RL indicates the hardness of this problem from information theoretic point of view (Jaksch et al., 2010; Li et al., 2015; Ortner & Ryabko, 2012; Jiang et al., 2017). These studies indicate that we might not be able to design RL methods which achieve our desired performance. Another bad news among all the good news: any innovation in both theory and practice requires a massive amount of effort.

In this short paper, we argue how the lack of theoretical understanding could contribute to the lack of broad applicability of RL in real-world applications. We argue that while we do not clearly understand how to approach an RL problem from the very elementary point of views in order to design efficient RL agents, we may not satisfyingly claim principled and significant progress in RL.

Let us think a little bit about the following topics and issues:

2. Policy Gradient
Policy gradient (PG) is a prominent approach in RL where we directly optimize our policies without explicitly constructing a model or learn a value function (Aleksandrov et al., 1968; Rubinstein, 1969). The preliminary PG methods, despite sounding promising, mainly suffer from high variance estimation of improvement direction and result
Maybe a few considerations in Reinforcement Learning Research?

While competing against state of the art methods always been a driving factor of developments in machine learning, it should not be the ultimate goal of our research.

3. Cherry Picking And Cherry Planting

Randomization plays an important role in the overall performance of RL algorithms both in theory and practice. For example, environments are inherently stochastic, or in function approximation methods, e.g., deep neural networks, there is randomness in initialization. In the simulated environment, the choice of random seed mainly determines these randomnesses, resulting in different agent’s behavior.

Recently there have been discussions on why random-seed optimization might be a wrong thing to do (Henderson et al., 2018). There is a line of research that the authors run their algorithms for many times and report the result of those runs that their algorithms perform well. While this way of evaluating the performance of an algorithm is useful when we care about planning, this is not suitable for even slightly more general goals. In general, one can refer to this way of reporting the results as cherry-picking.

Cherry planting: Besides the cherry picking, there are many scientific works which go beyond and do cherry planting. They not only optimize their algorithm but also they adapt the environment to their algorithms. For instance, there are many simple simulation-based control environments the experimenter can adjust the environment parameters desirable. If we also tune the environment of interested, then one could easily outperform baselines which are optimized for another set of environments parameters.

This approach and this way of doing empirical investigation is a natural outcome when the goal is to outperform the existing methods instead of providing a scientifically meaningful practical development. It is worth stating that both cherry picking and cherry planting are scientifically valid as long as the goal of the research is providing further insight in the understanding of RL problems, rather than climbing the leader-board of DRL scores.

4. Benchmarks

If we aim to outperform baselines on a set of toy control problems, simple adaptive control does it all.

The existing user-friendly simulated environments and wrappers, (Todorov et al., 2012; Bellemare et al., 2013; Machado et al., 2018; Brockman et al., 2016) for RL study made it possible to have numerous developments in RL study. Without these emulators, RL could not get the attention it deserves or even the current excitement from the empiricists, and we probably could not make the current significant progress in both the empirical and theoretical RL.

Among the existing environments, the Atari and the MuJoCo are the most popularized ones. The most used set of MuJoCo environments in the RL community is a set of toy environment that are excellent test beds to study the
behavior of RL algorithms. However, they are not practically sophisticated enough to deserve a significant body of research dedicated to performing well on them. It makes much more sense to use them as useful test beds and study the behavior of RL methods. It makes much more sense to make these environments harder to see where our algorithm breaks, or make them easy to see where our algorithms are the most suitable. While competitions are necessary components of research development, the final goal should not be forgotten unless we are designing a real-world task-specific RL agent. For example, in computer vision, researchers develop algorithms and compete to perform well on a given task, while the task is similar to the desired real-world one that we aim to solve. But competitions makes less sense if the tasks were toy vision tasks and far from practice.

Unfortunately, a great chunk of our researchers’ time has been dedicated to designing algorithm for toy MuJoCo environments with the goal to outperform the existing methods rather than principled studies, the studies which bring a useful amount of insight in developing RL methods for high-dimensional real-world problems. A similar phenomenon is also the case for Atari environment. There are quite many works and papers with extensively expensive empirical studies which clearly overfit to those environments and fail to outperform a random-uniform policy in simple two-state two-action Markov decision processes. While the RL community suffers from a lack of resources, e.g., experts’ time and computation power, limiting such researches could help to allocate our focus more efficiently.

What does it even mean to outperform an algorithm? A wise man/woman once said, "saying my number is bigger than yours is the lowest kind of science". In empirical RL study, mainly, we evaluate our methods on the given set of environments. We call an algorithm more significant if it has a better overall performance on these environments. However, this notion of overall better performance is a vague notion and the used notions in the literature are neither metrics not transitive. Therefore not a useful tool to compare methods, putting aside why even we want to compare algorithm in the first place.

If we are given a single practical task which worth spending our time to design an efficient algorithm for, it might be fine to overfit the task. However, when we deal with toy setting where we urgently need to understand the basics of RL, then competing there has less meaning, and barely informative when we compare a bunch of numbers.

5. What We Are Even Optimizing?

In many machine learning tasks, e.g., classification and regression problems, we are given an objective function, e.g., empirical risk or a surrogate of it, that we optimize over to come up with the desired model. In contrast, in RL, we do not have a clean objective function to optimize. In the online setting, we mostly need to collect our training data resulting in exploration and exploitation challenges. In the batch setting, where we access to a set of offline data, mainly there exists no such concept as test data. In this setting, generally, we are not able to test our agent against the real world.

In the cases that we have a good statistical understanding of our objective function, e.g., classification, climbing the leader board of state of the art corresponds to the designs of statistical optimization methods which perform well in practice. In RL, when we do not have a clear understanding of our objective function yet, we better do not spend the majority of our researchers time in the design of optimizer than study for statistically efficient RL algorithms.

6. Empirical And Theoretical Study

As we argued before, RL problems, by their nature are hard. Besieged this information theoretic hardness, we also argued that the research in both theory and practice are heavy to lift. Consider the process of an empirical study. For such studies, mainly, there is an initial hypothesis suggested by experienced researchers and a group of experienced programmers empirically test the hypothesis. This empirical study can also be accomplished by a group of undergraduate students who are supervised by senior researchers. Now consider theoretical studies, for this sort of researches, due to the hardness and complication of RL problems, mainly junior researchers have a hard time to accomplish a significant contribution. Therefore, it mostly requires experienced and senior researchers to spend their time on these problems to make considerable contributions.

Now consider the theoreticians vs practitioners ratio. The number theoreticians in RL compared to the practitioners is almost incomparable, while also there is a considerably week bonding between these two community. These issues further stimulate the slowness of this field. Clearly, there is a higher and urgent need for more theoreticians, as well as practitioners with much stronger bonding between them to boost up the advances in RL.

7. The Goal Of Our Research

RL is a broad topic of research and has many aspects to study. One can study RL from finding a good policy, e.g., optimal, safe. One can consider sample complexity, how many samples an algorithm takes to reach a good behavior, or how much of reward an algorithm loses due to exploration. Undoubtedly, this list can go on and on. In order to have a more clear scientific contribution, our papers should make their statements clear about which problem they tackle
and what are their achievements. They also need to provide sufficient evidence to back their claims, and what are the cases they fail and they perform very well. Furthermore, in their empirical comparison, they are required to compare against the algorithm in those realms.

8. What is the role of reviewers?

In the last few years, we had an era that more computations and empirical studies were tickets for acceptance of papers, even winning best paper awards. That exciting era played a significant role in encouraging more and more empirical developments in RL, resulted in a fantastic set of empirical studies. This movement toward more empirical research stimulates the community toward appealing more empirical development. All these happened at the time that we have not made a reasonably good theoretical development in RL. Consequently, there is an urgent need for such advancement to serve the high interest in the empiricists’ side.

However, there exists a serious problem worth considering. As we argued before, constructing a theoretical work in RL is hard and requires an extensive set of expertise. To support this hand wavy statement, we randomly chose a famous RL theorist, and also a practitioner. We made sure both parties are active researchers now. The theorist was able to write approximately five papers in 2018, but the practitioner wrote about 50. We perturbed these numbers to approximately conceal these parties’ identities. We do not report these numbers as factual evidence, instead, as a piece of non-negative information to update the readers’ belief or possibly narrowing down the confidence set.

This time, consider a non-famous theorist who writes a paper. Since the principled understanding of RL is still preliminary, this work might provide a slightly better-principled understanding of RL problems. Also, consider that this development is about the basics of RL. Asking this class of works to provide extensive empirical studies can be considered as a most damaging act on the development of RL. Moreover, these primary and theoretical works might be hard to implement, e.g., for high dimensional RL, and they might be statistically efficient not computationally efficient. Furthermore, consider that these class of works are mainly not tuned for a specific instance of simulation-based environments; instead, they are more general.

A famous example of such algorithms is the UCRL [Jaksch et al. 2010] that guarantees order optimal regret upper bound on tabular Markov decision processes. But it is obvious that a simple heuristic algorithm can easily outperform UCRL on some predefined domains.

Imagine we have an Markov decision process with 10 states \{1, 2, \ldots, 10\} and 10 actions \{1, 2, \ldots, 10\}. Imagine the optimal decision at any state is its index, e.g., optimal action at state 4 is the 4th actions. In this case, nothing can do better than an algorithm which naively maps states to actions equal to their indices. In this case, UCRL would result in a huge regret for this instance. However, UCRL regret guarantee not only still holds but also it is instance free.

In another word, if we change the environment a bit, the heuristic algorithm breaks while UCRL would still behave as it was promised.

The main point of this section is to convince the readers that 1) we are far from any claim about a reasonably good understanding of RL. Also, 2) do not expect theoretical works beyond what they promise. For instance, if a paper shows how to guarantee a low variance estimation of the gradient in policy gradient methods and monotonic improvement guarantee (Kakade & Langford 2002), we should not expect this algorithm to outperform anything when the criterion is the final performance. The theoretical analysis of this work just did not claim optimality. 3) Theoretical works progress brick after brick, and results are built on the top of one another. Consider that we just recently learned how properly use Q-learning (Jin et al. 2018).

**We just simply can not go to the moon before understanding how to build a simple engine.**

For this goal, let’s pave the road for the principled RL studies, both in theory and practice.

---

The final quote by a wise man/woman:

This is a big lesson. As a field, we still have not thoroughly learned it, as we are continuing to make the same kind of mistakes. To see this, and to effectively resist it, we have to understand the appeal of these mistakes. We have to learn the bitter lesson that building in how we think we think does not work in the long run. The bitter lesson is based on the historical observations that 1) AI researchers have often tried to build knowledge into their agents, 2) this always helps in the short term and is personally satisfying to the researcher, but 3) in the long run it plateaus and even inhibits further progress, and 4) breakthrough progress eventually arrives by an opposing approach based on scaling computation by search and learning. The eventual success is tinged with bitterness, and often incompletely digested because it is a success over a favored, human-centric approach.
References


Li, L., Munos, R., and Szepesvári, C. Toward minimax off-policy value estimation. 2015.


game of go with deep neural networks and tree search.

Singh, S. P., Kearns, M. J., Litman, D. J., and Walker, M. A.
Reinforcement learning for spoken dialogue systems. In

Sutton, R. S. Integrated architectures for learning, plan-
ning, and reacting based on approximating dynamic pro-

Thompson, W. R. On the likelihood that one unknown
probability exceeds another in view of the evidence of

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics
engine for model-based control. In Intelligent Robots
and Systems (IROS), 2012 IEEE/RSJ International Con-