Deep Reinforcement Learning for Economics: Progress and Challenges

ETAN A. GREEN
University of Pennsylvania, Arena–AI
and
E. BARRY PLUNKETT
Skip Protocol

We discuss the application of deep reinforcement learning to economic domains in general, and to bargaining on eBay in particular.

Categories and Subject Descriptors: J.4 [**Social and Behavioral Science**]: Economics

General Terms: Economics, Algorithms

Additional Key Words and Phrases: Deep reinforcement learning, Bargaining, Offline RL

1. INTRODUCTION

If, in 2018, you had asked about the most promising advance in artificial intelligence, the answer almost certainly would have been deep reinforcement learning. AlphaZero, trained using deep RL, had just been crowned the world's best player of chess, Go, and Shogi [Silver et al. 2018], and the application to real-world domains seemed imminent. "Artificial intelligence," said David Silver, winner of the 2019 ACM Prize in Computing, "is deep reinforcement learning." [Silver 2016]

Today, the promise of deep RL has not been realized. The fundamental challenge is that reinforcement learning agents require an environment in which to train, and creating an environment that reliably simulates the real world has proven difficult. The stories of RL successes are almost universally stories of pre-existing, reliable training grounds. This letter discusses an exception—an RL agent that bargains on eBay [Green and Plunkett 2022]—and the promise, as well as the challenges, it portends for applications of RL in economic domains and in the real world more generally. Our view is optimistic, if somewhat dystopian: in the near future, many economic decisions will be made by reinforcement learning agents.

2. BACKGROUND

Reinforcement learning agents learn by trial and error. They observe the state of the world (e.g., the board position in chess), take an action (a move) in a given state (board position), receive a reward (based on the outcome of the game), and reinforce actions that lead to higher rewards. To learn, RL agents need an environment that communicates the consequence of an action: the state in which it will take an action next, and the reward it receives for arriving at that state. By traversing many—often millions, sometimes billions—of states, the agent can learn a policy:

Authors' addresses: etangreen@gmail.com, bpiv400@gmail.com

an action to take in every state that maximizes a potentially distant reward, e.g., a move to make in every board position that maximizes the probability of winning. In deep RL, the mapping from states to actions is learned by a neural network.

The most ready domains for RL are those in which a reliable environment already exists. Among the most popular settings for testing RL algorithms are Atari games, for which the environment is the game itself [Hafner et al. 2019]. The state is defined by the pixels on the screen, an action is a move of the joystick, and the reward is the score. When the agent takes an action, the game responds by updating the pixels on the screen and the player's score.

Recent RL-driven advances in algorithms for matrix multiplication [Fawzi et al. 2022] and sorting [Mankowitz et al. 2023] also exploit pre-existing environments. For sorting, the state is the current order of elements in the array, an action may swap two elements, for instance, and the reward is a penalty for each action taken (so that the optimal policy is one that sorts an array in the fewest number of actions). When the agent takes an action, the next state is simply the new ordering of the array.

Adversarial games like chess pose an added complication: after the agent acts, the state in which it acts next depends on how the opponent responds. Hence, the training environment must incorporate the opponent's response. This problem neatly disappears in two-player, zero-sum games, such as chess and Go. To act optimally in these games, an agent need not learn to best respond to any opponent. Rather, it is sufficient to learn a best response to a best-responding opponent, which an RL agent can learn by playing against itself. By virtue of the minimax theorem, an equilibrium strategy learned in this manner will be optimal against any opponent, regardless of their intelligence. In a matter of days, AlphaZero went from knowing nothing about chess save the rules to the best chess player in the world simply by playing against itself millions of times.

3. BARGAINING ON EBAY

Many real-world games, and particularly those of economic interest, are neither two-player nor zero-sum. Bargaining, for instance, is multi-player: a seller may bargain with more than one buyer. It is also not zero-sum: the buyer and seller share a surplus only if they reach an agreement; otherwise, no surplus is generated. In multi-player or non-zero sum games, the goodness of an action depends on the opponent. Policies that perform well against one opponent may perform poorly against another.

One way around this problem is to train agents that perform well against a particular type of opponent: humans. An agent that exploits humans is useful in two ways: first, to exploit humans in the real world; and second, to help humans make better decisions.

We trained a deep RL agent to exploit humans when bargaining on eBay (in Best Offer listings, in which a seller sets a list price, and buyers and sellers may negotiate a lower price by making offers sequentially) [Green and Plunkett 2022]. The strategy that the agent learned, as either the buyer or the seller, meaningfully outperforms those that humans play. As the seller, the agent sells items more often and for higher prices. As the buyer, the agent purchases items more often and for

lower prices.

We show that most of these gains can be attained through simple tactics. For instance, the seller exploits human buyers by rejecting most first offers, particularly generous first offers, or those that request only a small discount on the list price. Generous first offers signal the buyer's willingness to pay more. By rejecting such offers, the seller communicates that the list price is firm. Human buyers often respond by paying the full list price.

The primary methodological contribution of our work is a template for training RL agents to exploit humans in real-world economic games. In a perfect world, we would have trained the agent on eBay—i.e., by making offers, observing counteroffers, and reinforcing offers that lead to higher payoffs. However, deep RL algorithms require an impossibly large number of actions to learn intelligent policies. We could neither list millions of items on eBay nor make millions of offers.

Our solution was to train a model of the real world from a massive dataset of negotiations on eBay [Backus et al. 2020], and then to train an RL agent in that model. This environment model is a neural network that simulates human behavior on eBay. The model predicts when buyers arrive, what offers they make, and how sellers respond—conditional on the features of the listing and the sequence of prior offers. When the agent acts as a buyer, we sample seller counteroffers from the environment. When the agent acts as a seller, we sample buyer arrivals, first offers, and counteroffers. In this manner, the agent bargains against millions of (simulated) humans in a couple days, and for only the cost of that compute time.

This approach is not without challenges, the most fundamental of which is that the environment model may not perfectly correspond to the real world. This difficulty has impeded applications of RL in robotics, in which agents are often trained in a model of the physical environment. Some aspects of the physical world, such as friction and wear on robotic arms, are difficult to model. As a result, agents trained to perform tasks in the model often cannot perform those tasks in the world [Kormushev et al. 2013].

4. CHALLENGES

Economic domains pose an added difficulty: confoundedness, or missing data. For instance, a buyer or seller on eBay may attach a text message to their offer, and while the eBay dataset we use contains an indicator for whether a text message accompanied the offer, it does not contain the content of the message. What someone says in their message probably affects how the other party responds to their offer [Backus et al. 2021]. If, say, nice messages induce acceptances and mean ones induce rejections, our model will sometimes respond with an acceptance and other times with a rejection—not because the true distribution is bimodal but because messages are sometimes nice and sometimes mean.

A second challenge concerns exploration. Often during training, an RL agent will try an unusual action, such as offering $0. In the real world, a seller will learn that an offer of $0 is unprofitable, and a buyer will learn that it is a waste of time. However, offers of $0 do not exist in our training data because humans never make them; hence, there is no guarantee that the model we train from those data will learns these truths.

We circumvent this issue by restricting the offers that the RL agent can make to those that are common in the data. Because the game tree is shallow—eBay limits the buyer and seller to no more than three offers each—this constraint mostly keeps the agent within the distribution of the training data. In settings with deeper game trees, however, exploration will lead to novel states, even if the training data are large and varied.

A more sophisticated approach is to penalize the agent for exploring outside the confines of the training data. This can be done by penalizing actions that the environment model deems unlikely, or by training an ensemble of environment models, each on a different partition of the training data, and penalizing actions that induce disagreement among the models. Neither approach seems to work well in practice. Rather than smoothly converging to a policy that balances rewards and penalties, standard RL algorithms like PPO oscillate between maximizing reward and minimizing exploration penalty without ever converging [Moskovitz et al. 2023].

To this point, we have considered RL approaches that learn *online*, by training either in the environment of interest or by training in a model of that environment. A newer, more promising alternative may be offline RL, in which a policy is learned directly from data [Kostrikov et al. 2021]. Offline RL proceeds in two steps. First, the data are used to train a critic, or a neural network that estimates the sum of discounted future rewards for taking a given action in the current state, and then taking the best sequence of actions observed in the data. Second, a policy is extracted from the critic by finding the best action in each state. One way to do this is to first train a *clone*, or a model that predicts the distribution of actions taken in the data, conditional on the state. Sampling actions from the clone yields a set of candidate actions. Evaluating those actions using the critic identifies which is best. This approach more naturally constrains exploration to actions that are in the distribution of the training data.

By its very name, offline RL offers an alternative to online RL. However, we view these approaches as complementary. Since the offline critic and the environment model both process state-action pairs, they can share a neural architecture. Hence, they can be trained jointly, by adding their losses before backpropagation. This conjoined approach may yield a better critic—by forcing the model to predict state transitions explicitly, rather than simply their rewards. A second advantage of training an environment model alongside an offline critic is that the environment model can be used to evaluate the policy extracted from the critic.

5. CONCLUSION

The challenges of applying deep reinforcement learning to real-world economic problems are significant, but so are the rewards. Methodological advancements offer hope that this promise will soon be realized.

REFERENCES

Backus, M., Blake, T., Larsen, B., and Tadelis, S. 2020. Sequential bargaining in the field: Evidence from millions of online bargaining interactions. *The Quarterly Journal of Economics 135,* 3, 1319–1361.

Backus, M., Blake, T., Pettus, J., and Tadelis, S. 2021. Communication and bargaining breakdown: An empirical analysis. In *Proceedings of the 22nd ACM Conference on Economics and Computation*. 129–129.

Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatain, M., Novikov, A., R. Ruiz, F. J., Schrittwieser, J., Swirszcz, G., Silver, D., Hassabis, D., and Kohli, P. 2022. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature 610,* 7930, 47–53.

Green, E. A. and Plunkett, E. B. 2022. The science of the deal: Optimal bargaining on ebay using deep reinforcement learning. In *Proceedings of the 23rd ACM Conference on Economics and Computation*. 1–27.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. 2019. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*.

Kormushev, P., Calinon, S., and Caldwell, D. G. 2013. Reinforcement learning in robotics: Applications and real-world challenges. *Robotics 2,* 3, 122–148.

Kostrikov, I., Nair, A., and Levine, S. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*.

Mankowitz, D. J., Michi, A., Zhernov, A., Gelmi, M., Selvi, M., Paduraru, C., Leurent, E., Iqbal, S., Lespiau, J.-B., Ahern, A., Köppe, T., Millikin, K., Gaffney, S., Elster, S., Broshear, J., Gamble, C., Milan, K., Tung, R., Hwang, M., Cemgil, T., Barekatain, M., Li, Y., Mandhane, A., Hubert, T., Schrittwieser, J., Hassabis, D., Kohli, P., Riedmiller, M., Vinyals, O., and Silver, D. 2023. Faster sorting algorithms discovered using deep reinforcement learning. *Nature 618,* 7964, 257–263.

Moskovitz, T., O'Donoghue, B., Veeriah, V., Flennerhag, S., Singh, S., and Zahavy, T. 2023. Reload: Reinforcement learning with optimistic ascent-descent for last-iterate convergence in constrained mdps. *arXiv preprint arXiv:2302.01275*.

Silver, D. 2016. Tutorial: Deep reinforcement learning. `https://icml.cc/2016/tutorials/deep_rl_tutorial.pdf`. Accessed: 2023–06-09.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science 362,* 6419, 1140–1144.