Proposal for a Benchmarking Method for Non-IID Performance in Horizontal Federated Deep Reinforcement Learning

Sam Shippey

***Abstract*—Federated Learning (FL) is an emerging paradigm for training deep learning models in a distributed context on low end devices without requiring each device to submit its data to a global coordinator. Many Reinforcement Learning tasks require huge amounts of data, often gathered from a large number of simultaneous runs through an environment, and are therefore natural fits for FL. However, FL suffers degraded accuracy if the data are not IID. In this paper, I provide code giving a demonstration of an approach called Horizontal Federated Deep Reinforcement Learning (HFRL), provide a definition and intuition for non-IID data in the context of games, and propose a set of experiments to stress-test new proposed modifications of HFRL against non-IID data.**

***Index Terms*—Federated Learning, Machine Learning, Reinforcement Learning**

I. Introduction

Federated Learning (FL) [1] takes advantage of the nature of backpropagation to efficiently make use of large numbers of low end devices as a compute engine for training deep learning models. These devices are often assumed to be somehow constrained, whether in terms of compute, memory, power, storage, and so on. This makes the approach well suited for use with mobile and IoT devices.

Reinforcement Learning (RL) is a method for approximating optimal control of agent in very large state spaces characterized as Markov Reward Processes (MRPs), often with sparse rewards. RL approaches have been used to approximate optimal policies for complex games, such as Starcraft II [2], DotA II [3], and Go [4]. Further, RL approaches have been applied in computer vision [5] and natural language processing [6], proving that the method has potential outside of games. One of the key complaints with RL is that the domains in which it is useful are often involve data which cannot be gathered immediately: Games must be played under a test policy to determine how well that policy works. Since games often have multiple performance bottlenecks (graphics, network, disk, etc), this can be very slow, even if sufficient compute capacity is available to train the deep learning model which serves to approximate the policy is available due to other resource constraints.

However, most games can be tailored to run on low end devices, for instance by turning down graphics settings on a graphically intensive game, or using self play and simulated network connections to avoid costly network latency. This makes it a natural fit for FL, which can take advantage of the large number of devices without sacrificing accuracy. The method I investigate in this paper, Horizontal Federated Deep Reinforcement Learning (HFRL), involves each individual device (or simulated device) considering itself as a single agent.

FL approaches suffer when data at the client devices are not "well-mixed", or IID. If the space of initial states is relatively small, then this will not become a problem. But in many domains of interest (e.g. Go) the first several actions may result in wildly disparate distributions for each of the participating client devices. Testing proposed modifications of the HFRL algorithm against these conditions enables higher confidenence that when these methods are applied to real world problems, they will sufficiently compensate for non-IID data.

II. Federated Learning

Federated Learning is an emerging paradigm for training deep learning models in a distributed context which works by averaging the weight changes reported by many devices which have trained some number of minibatches on their own private data. A central server then propagates these changes out to participating client devices. None of the client devices ever share their data in the process, allowing it to remain private [1]. This algorithm is called Federated Averaging, or **FedAvg**. Many processes can be added on top of **FedAvg**, such as participant selection to estimate the value of a given device's data [7], or strategies which improve privacy by injecting a small amount of noise in the training data or weight updates [8].

Among the key challenges with FL is the presence of non-IID data, or data which are not independent of each other or are from different underlying distributions. Since a typical use-case often involves autonomous gathering of data at the client side, this can have large effects on test accuracy. To my knowledge at the time of writing this, data being IID or not is not a particularly well defined concept for games as it is rarely valuable to know, and DRL literature rarely thinks of data in the same way as the larger deep learning community because of the constraints of the design space. I therefore aim to define IID data for HFRL in terms of starting state spaces.

III. Reinforcement Learning

Reinforcement Learning problems are conceptualized as learning a policy π which defines an action-value function q_π and a state-value function v_π. The policy works over a Markov Reward Process (MRP), which is a Markov Decision Process

(MDP) that also defines rewards. MDPs can be thought of as being defined over graphs where the vertices are game states and the edges are transitions between states weighted by how likely that transition is based on a given action. By defining RL problems in this way, we can immediately gain insight into the likely distribution of states given a set of actions, and most importantly we can reason that future states in an instance of an RL problem are strongly correlated with the initial states.

This is difficult to prove, but easy to see intuitively: The initial conditions of a game greatly impact the future of the game. An easy example can be found in chess, where openings can greatly control the future moves in a game.

IV. Related Work

V. Horizontal Federated Deep Reinforcement Learning

There are two natural approaches to training a Reinforcement Learning agent using FL, referred to as vertical and horizontal. The difference is in their approach to an instance of the environment. In vertical federated deep reinforcement learning (VFRL), all agents share the same environment [4], whereas in horizontal federated deep reinforcement learning (HFRL), all agents generate their own environments independently of each other. In this paper I primarily investigate the impact of non-IID data on HFRL. While non-IID data are still a concern in VFRL, and are easier to generate naively by generating an instance of a game, then restricting each agent to a small and disjoint subsubset of the action space, this approach is impractical for most easily available environments. Namely, environments with relatively small action spaces (such as cartpole or pong) or environments with larger action spaces in which roughly all of the action space is required to play the game competently (for instance, the atari environments in gym [9]) are impractical because each agent will learn almost nothing, and our model will be stuck in a local minimum almost immediately.

VI. Testing non-IID Data in Games for HFRL

I attempted one method of evaluating non-IID performance, and propose another using that experience.

In the first attempt, I built a contrived "guessing game", with the following rules:

1) Generate a "correct" sequence of guesses based on the move number
2) At each move number, insert the correct answer, alongside k random numbers, then shuffle.
3) At each move, the player guesses the number from the list. If this number is correct, increase the move number by one. Otherwise, repeat the move number.
4) The player wins when they finish a predetermined number of moves.

This had several issues, the largest being that the rewards are extremely sparse. Even adding rewards for correct guesses usually just resulted in the agent learning to guess. A tool for learning a distribution (like a neural network) is very different from a tool for building a tool that learns distributions from very few signals.

The proposed method is in some ways more obvious, but requires much more compute, and should have seemed obvious. Unfortunately I did not have time to evaluate this method. In HFRL, the distributions that are actually generated are environments. Therefore, the trick would be to find clustered but distinct sets of environments that could be partitioned in such a way that different agents would learn different subsets of the environment. So in other words, a game with enough initial states that the global model would be forced to confront non-IID data. This game is difficult to come by. I thought of using chess at first, since the initial moves encode a lot of information about the future of the game, but chess is a fairly hard game and so it would not be suitable for quick testing. The game I lean toward right now, and would like to try in the future is Pong. The initial states would encode the initial positions of the ball and paddles, along with the initial velocity vector of the ball. This would result in relatively different distributions of the game without being impossible to learn, and therefore would be suitable for rapidly iterating and testing ways to mitigate non-IID data.

VII. Future Work

This paper briefly investigates and proposes strategies for evaluating non-IID performance of HFRL, but does not show any empirical data. In this regard I was hampered mostly by not understanding how to implement RL algorithms very well. Now that I have some hands on experience, I'd like to try to actually implement them and iterate quicker. It might also be good to evaluate some mitigation strategies.

As mentioned above, VFRL may be significantly more vulnerable to non-IID data. An empirical comparison of the two methods non-IID performance would be difficult because they rely on fundamentally different distributions, but would be of great value in offering choices between different methodologies.

VIII. Conclusion

Federated Learning presents an interesting set of challenges and a wide variety of benefits for the field of machine learning, especially as mobile devices continue to proliferate as a primary computing device. By taking advantage of large numbers of client devices without sacrificing privacy, FL is in a good position to make deep learning faster, easier, more accessible, and maybe even cheaper.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *arXiv:1602.05629 [cs]*, Feb. 2017, arXiv: 1602.05629. [Online]. Available: http://arxiv.org/abs/1602.05629

[2] O. Vinyals, I. Babuschkin, J. Chung, M. Mathieu, M. Jaderberg, W. Czarnecki, A. Dudzik, A. Huang, P. Georgiev, R. Powell, T. Ewalds, D. Horgan, M. Kroiss, I. Danihelka, J. Agapiou, J. Oh, V. Dalibard, D. Choi, L. Sifre, Y. Sulsky, S. Vezhnevets, J. Molloy, T. Cai, D. Budden, T. Paine, C. Gulcehre, Z. Wang, T. Pfaff, T. Pohlen, D. Yogatama, J. Cohen, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, C. Apps, K. Kavukcuoglu, D. Hassabis, and D. Silver, "AlphaStar: Mastering the Real-Time Strategy Game StarCraft II," https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/, 2019.

[3] OpenAI, C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. d. O. Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang, "Dota 2 with Large Scale Deep Reinforcement Learning," arXiv, Tech. Rep. arXiv:1912.06680, Dec. 2019, arXiv:1912.06680 [cs, stat] type: article. [Online]. Available: http://arxiv.org/abs/1912.06680

[4] J. Qi, Q. Zhou, L. Lei, and K. Zheng, "Federated Reinforcement Learning: Techniques, Applications, and Open Challenges," arXiv, Tech. Rep. arXiv:2108.11887, Oct. 2021, arXiv:2108.11887 [cs] type: article. [Online]. Available: http://arxiv.org/abs/2108.11887

[5] M. Bellver, X. Giro-i Nieto, F. Marques, and J. Torres, "Hierarchical Object Detection with Deep Reinforcement Learning," arXiv, Tech. Rep. arXiv:1611.03718, Nov. 2016, arXiv:1611.03718 [cs] type: article. [Online]. Available: http://arxiv.org/abs/1611.03718

[6] A. B. Siddique, S. Oymak, and V. Hristidis, "Unsupervised Paraphrasing via Deep Reinforcement Learning," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Aug. 2020, pp. 1800–1809, arXiv:2007.02244 [cs]. [Online]. Available: http://arxiv.org/abs/2007.02244

[7] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient Federated Learning via Guided Participant Selection," *arXiv:2010.06081 [cs]*, May 2021, arXiv: 2010.06081. [Online]. Available: http://arxiv.org/abs/2010.06081

[8] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farhad, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated Learning with Differential Privacy: Algorithms and Performance Analysis," arXiv, Tech. Rep. arXiv:1911.00222, Nov. 2019, arXiv:1911.00222 [cs] version: 2 type: article. [Online]. Available: http://arxiv.org/abs/1911.00222

[9] "openai/gym," Jun. 2022, original-date: 2016-04-27T14:59:16Z. [Online]. Available: https://github.com/openai/gym