Introduction

Reinforcement Learning is a paradigm of Machine Learning where an agent learns from experience on making decisions in an environment. Q learning is a simple algorithm where the agent learns a value for every action in every state, or a function which approximates these values. In a small discrete environment, and tabular Q values, the exact values can be computed using the Bellman update, and the learned values can be compared against these optimal values. The policy is how the agent makes the decision which for Q learning is greedily choosing an action by its value. State of the art methods such as TRPO (Trust Region Policy Optimization) and PPO (Proximal Policy Optimization) are based on a Policy Gradient Approach which uses a function (such as a Neural Network) as the policy and optimizes it with feedback from the environment.

Federated Learning is a method of training where multiple clients collaborate to train a global model using separate data. This approach has several advantages. There is a Speedup of training since each client is computing in parallel. Provides greater access to machine learning for smaller clients who don't have the resources to train a model by themselves. Two important considerations are stability of convergence which is if the global model will still find a local optimum, and if the speedup is linear, which means for N clients working in parallel the model will converge N times faster. Most of the work in Federated Learning is on the supervised learning paradigm. Federated Reinforcement Learning is the combination of the two, and there is little existing work in this area.

The question for this research is does Federated Reinforcement Learning converge to a good solution and does it have linear speedup. This will give general knowledge on how Federated Reinforcement Learning behaves and will aid in designing large scale reinforcement learning experiments to forecast required resources and time, make the most efficient use of federated training, and ensure the model will converge to a solution.

The methodology is to first test Tabular Q Learning on a small-scale Markov Decision Process and analyze the learned Q values convergence to the optimal Q values computed with the Bellman Update. These results will be compared between a single agent and Federated Learning with multiple agents. Then Federated Reinforcement Learning will be tested with more complicated environments in simulations with Open AI gym, different function approximations such as linear and neural networks instead of a q table, and with the state of the art Reinforcement Learning algorithms TRPO and PPO.

The ideal result would be a way to get both stability and linear speedup with Federated Reinforcement Learning that works with existing state of the art methods and a variety of complex environments. Though a negative result would also be useful in informing the design of future research with any potential problems.

Literature Review

This project will be testing multiple reinforcement learning algorithms with different function approximations starting with the simpler Q learning, and then moving to state of the art methods to see how they perform when federated by testing if they converge or are unstable, and if there is linear speedup of convergence. Federated Reinforcement learning is a relatively new development, so there is not that much work studying it.

Federated Learning

Federated learning trains one large model from distributed data. Clients train their model, and the server aggregates the models and sends updates back to clients. FedAvg is a method where this aggregation is averaging weights by clients to construct the global model (Konecnˇ y´ et al., 2017). The main issues are unbalanced data between clients and bandwidth limits. Bandwidth needed can be reduced with structured and sketched updates. Structured updates limit learning to a subset of features each round to reduce communication, and sketched updates learn with the whole model but compress the update before communicating (Konecnˇ y´ et al., 2017). These considerations of federated supervised learning seem like they will also apply to federated reinforcement learning and are a metric that should be included in the tests.

FedAvg is the most used federated learning algorithm. It is both theoretically proven and empirically tested that FedAvg convergence is sped up linearly by the number of active participants (Qu et al., 2020). The advantages of federated learning are preserving data privacy, linear speedup, and makes deep learning more accessible for edge devices and individuals. It does not appear there is a similar study on linear speedup for federated reinforcement learning, so this may be a gap in the literature for this project to help fill.

Reinforcement Learning

Reinforcement Learning is where an agent learns a policy of what action to take in a state of the environment. A simple RL algorithm is Q learning where the Q function is a value for every state action pair. To learn the agent either takes the action with maximum q value in its state, or takes a random action, and then updates its Q function using the reward it received and the next state. For very small-scale tabular examples, the Q function can be computed exactly with the bellman update and compared to the Q function learned from experience. The Q function can either be a table, or a function approximation such as linear or a neural network (Sutton & Barto, 2018). The advantages of Q learning are it has been studied for a while, is easy to implement and explain, though there are newer methods that perform better.

Another method is policy gradient. Instead of acting greedily based on Q value, this method uses a neural network to approximate the policy in addition to approximating the value function. The policy network outputs action selection probabilities for an input state, which lets it model stochastic policies and avoids discontinuity. A gradient of the policy can be estimated through experience, and then used to optimize the policy (Sutton et al., 2000). This is the basis for future state of the art algorithms.

Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO) are 2 state of the art reinforcement learning methods based on policy gradient. These methods alternate between acting in the environment to collect data and optimizing a new policy on the data collected by the old policy, with constraints on how different the policies can be. PPO accomplishes this in a simpler and more flexible way than TRPO. (Schulman, Levine, et al., 2017; Schulman, Wolski, et al., 2017).

Federated Reinforcement Learning

Federated Reinforcement Learning is a combination of these two ideas. One approach is FedRL, which maintains data privacy where sharing datasets or individual models is not an option (Zhuo et al., 2020). There are multiple agents which can’t share their state, and all using their own model. Agents receive gaussian differentials of the Q network outputs from other agents. This allows a global Q network to be constructed without any agent learning about the individual Q network of another agent. Each agent views the other agents' Q network outputs as a constant to update its own. After communication, there is a federated Q network for each agent (Zhuo et al., 2020). It does not appear there is a central server maintaining a global aggregated network unlike FedAvg, so it has stronger privacy.

Two Categories of Federated Reinforcement Learning are Horizontally Federated Reinforcement Learning (HFRL) and Vertically Federated Reinforcement Learning (VFRL). HFRL is where each agent has its own independent environment to act in, and information is shared between agents to explore more possible states. Each agent has similar actions and environment, and shares gradients to aggregate into a global model. VFRL is where there is a global environment that each agent partially observes, and may have different actions in. Actions of agents affect each other, and the areas they are able to observe may partially intersect. While each agent acts differently and pursues different rewards, sharing information gives them a more general model of the environment. (Qi et al., 2021). It seems FedRL method used above is a VFRL approach, though Zhuo et al. do not use this term in the paper. HFRL sounds more similar to the FedAvg method used for supervised learning. This paper does describe that there is a speedup, but not if it is linear (Qi et al., 2021).

Open AI gym is a standardized collection of simulated environments, which has more variety and is easier to use than existing alternatives. OpenAI Gym was created to act as benchmarks for reinforcement learning similar to how computer vision uses ImageNet. Standardization of problem definitions makes it much easier to reproduce, compare and further develop existing work (Open AI, 2016). This is the simulated environment we will be testing with.

References

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An Introduction to Deep Reinforcement Learning. *Foundations and Trends® in Machine Learning*, *11*(3–4), 219–354. https://doi.org/10.1561/2200000071

Konecnˇ Y, J., McMahan, H. B., Yu, F. X., Suresh, A. T., & Bacon, D. (2017). Federated Learning: Strategies for Improving Communication Efficiency. *ArXiv*. https://arxiv.org/pdf/1610.05492.pdf

Open AI. (2016). *Gym: A toolkit for developing and comparing reinforcement learning algorithms*. Open AI Gym. Retrieved February 27, 2022, from https://gym.openai.com/docs/

Qi, J., Zhou, Q., Lei, L., & Zheng, K. (2021). Federated reinforcement learning: techniques, applications, and open challenges. *Intelligence & Robotics*. https://doi.org/10.20517/ir.2021.02

Qu, Z., Lin, K., Kalagnanam, J., Li, Z., Zhou, J., & Zhou, Z. (2020). Federated Learning's Blessing: FedAvg has Linear Speedup. *ArXiv*. https://arxiv.org/pdf/2007.05690.pdf

Schulman, J., Levine, S., Moritz, P., Jordan, M., & Abbeel, P. (2017). Trust Region Policy Optimization. *ArXiv*. https://arxiv.org/pdf/1502.05477.pdf

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. *ArXiv*. https://arxiv.org/pdf/1707.06347.pdf

Sun, T., Shen, H., Chen, T., & Li, D. (2021). Adaptive Temporal Difference Learning with Linear Function Approximation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1. https://doi.org/10.1109/tpami.2021.3119645

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning series)* (second edition). A Bradford Book.

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (2000). *Policy Gradient Methods for Reinforcement Learning with Function Approximation*. Washington.Edu. Retrieved February 27, 2022, from https://homes.cs.washington.edu/~todorov/courses/amath579/reading/PolicyGradient.pdf

Wise, M., & Foote, T. (2017). *About - TurtleBot*. Turtle Bot. Retrieved February 27, 2022, from https://www.turtlebot.com/about/

Zhuo, H. H., Feng, W., Lin, Y., Xu, Q., & Yang, Q. (2020). Federated Deep Reinforcement Learning. *ArXiv*. https://arxiv.org/pdf/1901.08277.pdf