Annotated Bibliography of Federated Learning Papers

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 Horizontally Federated Deep Reinforcement Learning (HFDRL), 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 HFDRL against non-IID data.

Index Terms—Federated Learning, Machine Learning, Reinforcement Learning

1 Introduction

Federated Learning (FL) [**?**] 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 [**?**], DotA II [**?**], and Go [**?**]. Further, RL approaches have been applied in computer vision [**?**] and natural language processing [**?**], 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, Horizontally Federated Deep Reinforcement Learning (HFDRL), 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 HFDRL 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.

2 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 [**?**]. 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 [**?**], or strategies which improve privacy by injecting a small amount of noise in the training data or weight updates [**?**].

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 HFDRL in terms of starting state spaces.

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.