Geo-Distributed Machine Learning for non-IID Data Proposal

Prithvi Gudapati (pgudapat)

November 2018

1 Web Page

https://prithg98.github.io/400Project/

2 Description

2.1 Background and Problem

Increasingly more and more data is being generated by powerful devices such as phones, laptops, and wearable smart devices. These devices generate a lot of rich data that is perfect for machine learning tasks. However, because they are distributed all across the world, transporting all the data to a single data center can be incredibly expensive, slow, and potentially prohibited by security and privacy laws that restrict the movement of raw data. Furthermore, technology companies such as Amazon, Google, and Microsoft have built regional data centers around the world in order to be able to serve their customers as fast possible. These regional data centers along with edge clusters contain valuable data from nearby devices. As a result, new solutions that allow for the training of machine learning models on data distributed across the world are important.

Previous work in the field of geo-distributed such as Hsieh *et al.*'s [2] work focused primarily on the case of independent and identically distributed (IID) data. In this case, Hsieh *et al.* [2] made use of the parameter server architecture because of its ability to work with a wide-variety of machine learning models. Each data center had its own approximately-correct copy of the global machine learning model. Hsieh *et al.* [2] noticed that most communication resulted in insignificant updates to the global model. As a result, they put forward an *Approximate Synchronous Model* in which updates are aggregated until they reach a certain threshold at which point the accumulated updates are significant enough to be communicated. This approach significantly reduced communication across the wide-area networks (WANs), which were used to communicate between data centers, and resulted in similar algorithm convergence and accuracy compared to the case in which all the data was in a single data center.

However, this approach breaks down when dealing with non-IID data, which is the more likely situation in the real world. Thus, it is important to explore solutions for the non-IID case. In order to solve the issue, the problem at hand must first be understood. As a result, it is important to characterize the problem by looking at different applications, machine learning algorithms, and data skewness. Only once the problem at hand is understood should solutions be explored. The challenges of this problem are that there are many possible scenarios and many different solutions to explore.

2.2 Impact

Solving the problem of geo-distributed machine learning in the non-IID case will prove to be incredibly valuable. Machine learning models will then be able to train on data spread across the world without having to transport the data to a single data center, which could prove to be either expensive or impossible. Furthermore, it is important that the non-IID case is handled because it is the most likely scencario to be encountered in the real world.

2.3 Mentors

For this project, I will be working with Professor Phil Gibbons, who will be my faculty mentor, and Kevin Hsieh, a Ph.D. student advised by Professor Gibbons.

3 Goals

I hope that I can help finish this project and develop a solution that can effectively and efficiently allow machine learning models to train on geo-distributed, non-IID data. The success of the project will be evaluated based on the accuracy and convergence of the different algorithms when run on different datasets. It will also be evaluated on the communication costs.

If the project goes slower than expected, we should still be able to characterize the problem and determine whether or not certain solutions are viable. If the project goes expected we will be able to characterize the problem and also develop and implement a viable solution. If we are able to go faster than expected we can then also work on adapting our approach for multi-task learning and even potentially train a hierarchy of models instead of just a single global model.

4 Milestones

4.1 1st Technical Milestone

The first technical milestone would be to help finish the characterization of the problem at hand. In order to do so, I must first become familiar with the relevant literature. I need to also become familiar with the Caffe framework, so I can train different models for the characterization phase.

Once I have the relevant background, I will then move to trying to find computer vision algorithms that can help me partition the data for facial recognition tasks by different features such as race and gender. This will allow me to then skew data based on these features. I will then work on an example with an LSTM network as the current examples have all used CNNs and it is important to understand how things change with different neural network architectures.

4.2 Bi-weekly Milestones

Because of the nature of the project, it is tough to know exactly at what stage of the process I will be at at each of bi-weekly meetings as the results from the problem characterization might lead us to try different approaches. By February 1st, the goal would be to have finished the problem characterization and to have a robust understanding of the problem and how different levels of skewness, different machine learning algorithms, and different applications affect things. For the remainder of the semester milestones would be to come up with and test different potential approaches to solving the problem. One potential solution to explore is communication control. In this approach the communication threshold from the Gaia architecture would adapt to the situation at hand. Ideally, we would have a solution mapped out for this by February 15th and then implement the solution by March 1st. If we find that this approach does not work or a different approach is more suitable we must adapt accordingly. We would then either have to explore different solutions hopefully by March 22nd, and then implement them by April 5th. Once we can successfully train a global model, we need to then be able to adapt to a multi-task setting. We would figure out this approach by April 22nd, and by May 3rd we would implement it.

5 Literature Search

I have read Hsieh *et al.*'s [2] work on Gaia. Gaia proved that the approximately synchronous parallel model works effectively over WANs when the data is approximately IID. I have also read two works dealing with *Federated Learning.* The first of which is McMahan *et al.*'s [3] work that first introduced the notion of *Federated Learning.* Smith *et al.*'s [4] work builds upon this work and introduces the notion of federated multi-task learning in which separate but related models are fit simultaneously in the federated setting. I have also read Tang *et al.*'s [5] work dealing with decentralized parallel stochastic gradient descent in a scenario where there is large data variance among workers.

The final work that I have looked at is Cui *et al.*'s [1] work with GeePS. GeePS deals with deep learning on distributed GPUs using GPU-specialized parameter servers.

In order to gain a further understanding of the relevant topics, I will read more papers on multi-task learning and federated learning. I will also start reading the papers that the papers that I have already read mention.

6 Resources Needed

In order to run experiments I need access to a cluster with server grade CPUs and GPUs. The Parallel Data Lab at CMU has provided me access to the Orca cluster to run these experiments.

References

[1] Henggang Cui, Hao Zhang, Gregory R. Ganger, Phillip B. Gibbons, and Eric P. Xing. Geeps: Scalable deep learning on distributed gpus with a gpu-specialized parameter server. In *Proceedings of the Eleventh European Conference on Computer Systems*, EuroSys '16, pages 4:1–4:16, New York, NY, USA, 2016. ACM.

[2] Kevin Hsieh, Aaron Harlap, Nandita Vijaykumar, Dimitris Konomis, Gregory R. Ganger, Phillip B. Gibbons, and Onur Mutlu. Gaia: Geo-distributed machine learning approaching lan speeds. In *Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation*, NSDI'17, pages 629–647, Berkeley, CA, USA, 2017. USENIX Association.

[3] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016.

[4] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. *CoRR*, abs/1705.10467, 2017.

[5] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D_2: Decentralized training over decentralized data. *CoRR*, abs/1803.07068, 2018.