Article Reviewed*: S. Li, T. Ben-Nun, S. D. Girolamo, D. Alistarh, and T. Hoefler, “Taming unbalanced training workloads in Deep Learning with partial collective operations,” Proceedings of the 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, 2020.*

This article “Taming unbalanced training workloads in Deep Learning with partial collective operations” by Shigang Li proposes to eager-Stochastic Gradient Descent which relaxes the global synchronization for decentralized accumulation. They use two partial collections solo and majority to handle load imbalances in training workloads. Traditional synchronous Stochastic Gradient Descent (SGD) achieves good accuracy for a wide variety of tasks, but relies on global synchronization while eager-SGD results in a 2.64 x speedup(ResNet-50), achieves 1.29 x speedup (ResNet-50) and 1.27 x (LSTM) speedup without losing accuracy.

Their main motivation is that deep learning models are on a step trajectory to becoming the most important workload on parallel and distributed computer systems, since there is a predicted growth rate in the demand for optimizing the training procedure. Due to this the researchers can see that load imbalance is an additional barrier to scalability which will apply to further applications of neural networks. The authors identify four main contributions made by their research: a detailed analysis of workload imbalance in deep learning training, define and implement partial collectives (majority and solo allreduce), eager-SGD for asynchronous decentralized distributed training of neural networks, and an experimental study of converge and training speedup. Specifically they were able to achieve a 1.27 x speedup over synchronous SGD on a video classification task without losing accuracy.

They continue the paper by breaking down the components of eager-SGD such as how they used Parameter Server architecture to maintain a global view, implementing SGD by using partial collectives and the schedule activation needed for the collectives. A solo collection is forces the slow processes to execute the collective as soon as the fastest process executes it while majority collective waits until at least half of the processes join the collective by randomly selecting an ‘initiator’ process. They proceed to prove that under reasonable set of modeling assumption the algorithm will still converge. They use a partial all-reduce operation to update the collective which guarantees that each ‘ADS’ or asynchronous distributed sum eventually returns an output, is consistent, is the size of Q >1 (where Q is the lower bound parameter), and there exists a bounded parameter that any ADS update can be rejected for at most the bounded parameter.

In the article they show extensive knowledge of the converge proofs as well as the load imbalances. More specifically they recognize repeatedly that load imbalance can be caused by the data itself or by the system, the distinction is important because I have realized after taking higher level courses that the system is just as important as the data. The correct algorithms and set up can result in a potential gain for the researchers, whether that be in accuracy of the algorithms or the potential cost of a system. In the study they utilize a range of data sizes and calculate the speedup, further research could be led into optimizing system performance and evaluating it on the data sizes. Not to mention that the partial all-reduce operation can be used for other algorithms in parallel computing.

Overall, the article gave in-depth analysis of the performance speedup achieved using eager-SGD and its possible applications in language models and object detection. Throughout the paper the sections and figures failed to line up in a coherent order. At most times they were on spoken about pages after they were presented, aside from it was very well-written. With the rising demand for further development of deep learning models their contributions are important as they were the first to implement asynchronous and stale-synchronous decentralized SGD where messages propagate to all at once.