DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Data Engineering and Analytics

Analyzing Bottlenecks in Hivemind

Adrian David Castro Tenemaya

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Data Engineering and Analytics

Analyzing Bottlenecks in Hivemind

Analysieren von Engpässen in Hivemind

Author:	Adrian David Castro Tenemaya
Supervisor:	Prof. Dr. Ruben Mayer
Advisor:	M. Sc. Alexander Isenko
Submission Date:	August 31, 2022

I confirm that this master's thesis in data engineering and analytics is my own work and I have documented all sources and material used.

München, August 31, 2022 Adrian David Castro Tenemaya

Acknowledgments

I was 4 years old and a half when I first arrived in Italy. Now, 20 years later (wow), I am completing my journey. Thanks to everyone who tagged along on my academic journey. Special thanks to my wonderful and supportive parents Laura and José; my "sore" Valeria, and my girlfriend Iris. Also, thanks to everyone who has believed in me and who didn't make me quit (you know who you are).

Abstract

The amount of computing resources required to train state-of-the-art deep neural networks is steadily growing. Most institutions cannot afford the latest technologies, which are sometimes needed to keep up with today's demanding deep neural network research. Access to powerful devices is therefore limited to few research institutions, slowing down research.

In [RG20] they propose a novel concept for decentralizing deep neural network training using large amounts of consumer-grade hardware. The training algorithm described in the paper is called "Decentralized Mixture-of-Experts" (DMoE) and employs a combination of decentralized and Mixture-of-Experts [Sha+17] techniques. This allows thousands of computing devices to join forces and train a single neural network model together. DMoE achieves this by splitting the target neural network model into different parts called partitions, similarly to model parallelism. Each partition is then replicated across a subset of workers participating in the training. Next, a gating function is used to select which workers can perform the next operation on the given input. After the workers have been selected and located using a Distributed Hash Table (DHT), the input data is sent to the workers, and a forward pass is performed. A similar algorithm is applied during the backward pass. DMoE proved that scaling model training to thousands of heterogeneous compute nodes is possible, thus enabling large-scale community research projects.

Although DMoE is robust against training latency, it also requires large amounts of data to be exchanged between every participant worker. We may assume, however, that most participants in the network will not have datacenter-grade network connections and bandwidth. Therefore, the communications needed to perform training may saturate a worker's network. An approach suggested by [RG20] to reduce the network load is to compress and convert tensors to a lower precision before transfer.

From the combination of features and findings of [RG20; Rya+21], Hivemind was created. Hivemind [tea20] is an open-source framework that enables collaborative model training using a large number of heterogeneous devices from universities, companies, and volunteers. Every device participating in the computation may differ in its characteristics, featuring different architectures and network speeds.

In Hivemind interactive demonstration [tea20], 40 devices jointly trained a modified DALL-E [Ram+21] neural network model over 2.5 months. The reported results,

however, do not include the participant's device information and metrics. Without this type of information, it's not possible to perform an independent analysis of the effects of different configurations on training. In this paper, we intend to reproduce the results of [tea20] on our cluster using different device configurations to identify the impact of key system metrics on Hivemind.

Over the last years, research and software libraries like Hivemind have been focused on reducing and optimizing deep neural network model training times with techniques such as data and model parallelism. In [Xin+21] however, the authors show that as much as 45% of total training time may be spent on preprocessing tasks alone. Despite this, the impact of preprocessing pipelines is often ignored in current research. Therefore with this paper, we propose to explore the impact of preprocessing pipelines in [tea20].

As noted by [Ise+22], it is crucial to find and analyze bottlenecks during computation to maximize performance. In their work, they also detail several possible improvements that can be applied in preprocessing pipelines, increasing throughput under certain circumstances. Intuitively, given the high amount of communications and data loading that DMoE needs, the preprocessing pipeline may be subject to inefficiencies. Using the techniques and findings showcased in [Ise+22], this paper further aims to find bottlenecks in the Hivemind preprocessing pipeline.

In this paper, we will analyze the impact of Hivemind's different possible scenarios on preprocessing pipelines. As we test the software and its limitations, we might find possible areas of improvement in Hivemind. Whenever possible, we will further contribute using the knowledge gathered through our experiments by improving the Hivemind [tea20] source-code. Our contributions can be summarized as follows:

- We analyze the challenges of optimizing preprocessing pipelines in decentralized distributed training and provide insights on possible improvements
- We verify the effectiveness of Hivemind for different peer hardware configurations concerning preprocessing pipelines
- We use the gained knowledge and insights to contribute to the Hivemind open-source library.

Contents

1 Introduction

Neural networks (NNs) have been one of the main focuses of research in the past decade. They were first conceptualized back in

1.1 Section

1.1.1 Subsection

List of Figures

List of Tables

Bibliography

[Ise+22] A. Isenko, R. Mayer, J. Jedele, and H.-A. Jacobsen. "Where Is My Training Bottleneck? Hidden Trade-Offs in Deep Learning Preprocessing Pipelines." In: *Proceedings of the 2022 International Conference on Management of Data*. SIGMOD '22. Philadelphia, PA, USA: Association for Computing Machinery, 2022, pp. 1825–1839. ISBN: 9781450392495. DOI: 10.1145/3514221.3517848.

[Ram+21] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. "Zero-Shot Text-to-Image Generation." In: *CoRR* abs/2102.12092 (2021). arXiv: 2102.12092.

[RG20] M. Riabinin and A. Gusev. "Learning@home: Crowdsourced Training of Large Neural Networks using Decentralized Mixture-of-Experts." In: *CoRR* abs/2002.04013 (2020). arXiv: 2002.04013.

[Rya+21] M. Ryabinin, E. Gorbunov, V. Plokhotnyuk, and G. Pekhimenko. "Moshpit SGD: Communication-Efficient Decentralized Training on Heterogeneous Unreliable Devices." In: *CoRR* abs/2103.03239 (2021). arXiv: 2103.03239.

[Sha+17] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, G. E. Hinton, and J. Dean. "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer." In: *CoRR* abs/1701.06538 (2017). arXiv: 1701.06538.

[tea20] L. team. *Hivemind: a Library for Decentralized Deep Learning*. https://github.com/learning-at-home/hivemind. 2020.

[Xin+21] D. Xin, H. Miao, A. G. Parameswaran, and N. Polyzotis. "Production Machine Learning Pipelines: Empirical Analysis and Optimization Opportunities." In: *CoRR* abs/2103.16007 (2021). arXiv: 2103.16007.