Project Proposal - Traffic Pattern Analysis and Comparison of Distributed Deep Learning Models

Li He V00972954

1 Problem Statement

There is huge increase of using distributed deep learning models, such as those used for training neural networks, for handle the large datasets and complete the complex tasks. However, those models produce unique traffic patterns that will affect the overall network performance. Analyzing and comparing the traffic patterns of different DNN models in a distributed training environment is essential for optimizing the models itself and improving the network infrastructure. Additionally, in the distributed deep learning models always have the high computational and communication costs, it can help minimize the costs and improve the utilities through understanding the traffic patterns.

2 Previous Work and Limitation

Recent studies use machine learning techniques like clustering, classification, and regression to analyze traffic patterns[1]. Deep learning models such as CNNs[2] help with tasks such as traffic prediction, anomaly detection, and pattern recognition. Most traffic analysis primarily examines general network traffic or traditional distributed systems, rather than the specific traffic patterns generated by distributed deep learning models. Deep learning workloads have unique communication patterns, leaving many areas open for further research and study.

3 Methodology

- **Data Collection and Preparation:**
  - Analyze the public datasets: Doing the literature survey and look for some public available dateset online. As for now, I found one in the paper, they offer their dataset [3]on the github.
  - Collecting data from distributed deep learning workloads:
    1, Find the typical deep learning frameworks such as TensorFlow, PyTorch, and Horovod. 2, Deploy a distributed cluster using cloud platforms (e.g., AWS, Compute Canada) 3, Choose some popular deep learning models, for example, Convolutional Neural Networks (CNNs) for image classification, Recurrent Neural Networks (RNNs) for sequence modeling and Transformers for natural language processing. 4, Generate workloads and collect the data: using the selected frameworks and models to run distributed training jobs. We can capture the network traffic through network monitoring tools such as Wireshark, NetFlow during the training.
- **Data Cleaning and Analysis:** 1, Clean the datasets to remove noise, irrelevant information and aggregate the data into suitable time intervals. Handling missing values and normalizing timestamp formats; 2, Use visualization tools (e.g., Matplotlib, Seaborn) to plot traffic trends and anomalies; 3, Apply clustering algorithms (e.g., Knn, k-means) to group similar traffic patterns.
- **Compare the traffic patterns of different distributed deep learning models:** We can conduct a framework comparison by running the same deep learning model (e.g., ResNet-50) on different frameworks (e.g., TensorFlow, PyTorch) and analyzing the resulting traffic patterns in each case. Additionally, we can run different deep learning models (e.g., CNN, RNN, Transformer) on the same framework to examine how model architecture influences traffic patterns.

4 Timeline

Course Project Website:
`https://studentweb.uvic.ca/~lihe0628/`

Deadline	Tasks	Deliverables
Feb 7	literature review; define tools and datasets.	Project proposal document
Feb 21	install frameworks; prepare datasets;	progress report
Mar 7	collect and preprocess data; analyze traffic patterns.	midterm progress report and presentation
Mar 21	Complete analysis; compare frameworks/models.	progress report
Apr 4	Finalize results; prepare slides.	Final presentation slides
Apr 11	Write final report.	Final project report

Table 1: Project Timeline

References

[1] A. Ashfaq et al., "A novel k-means clustering-based approach for network intrusion detection," Computers & Security, 2017.

[2] Z. Wang et al., "A CNN-based approach for network traffic prediction," IEEE Transactions on Network and Service Management, 2019.

[3] J. Cao, Y. Guan, K. Qian, J. Gao, W. Xiao, J. Dong, B. Fu, D. Cai, and E. Zhai, "Crux: GPU-Efficient Communication Scheduling for Deep Learning Training," in Proceedings of the ACM SIGCOMM 2024 Conference, pp. 1–15, Aug. 2024.