Abstract

An immense number of IoT devices has led to an exponential growth of data moving across the network. Traditionally, Artificial Intelligence (especially Deep Learning, DL) tasks on the IoT data are usually run on the cloud or other centralized systems which leads to issues like network congestion and significantly higher latencies for real-time applications such as human activity recognition (HAR) or facial recognition. Thus, unleashing DL services using resources at the network edge near the data sources has emerged as a desirable solution. The latest research in this field is inclined towards performing Distributed Deep Learning on the edge nodes. One of the major obstacles encountered is that edge nodes have considerably smaller computational power compared to their cloud counterparts which act as a hindrance to perform heavy DL tasks.

This study aims to enhance the latencies of real-time applications by experimenting with distributed DL on the edge infrastructure. Data reduction is performed on the edge along with the DL tasks. This avoids the network congestion caused by traditional techniques like running heavy DL models on the cloud and brings the computation closer to the end devices. In this study, we aim to use preprocessing techniques like sliding window, data reduction techniques such as Autoencoders(AE), and inference techniques on edge nodes.

We are proposing to move the Deep Learning tasks from remote cloud servers to the distributed edge nodes and along with that use the data reduction techniques. We plan to achieve lower latencies for real-time applications like HAR as well as lower the memory footprint for running the Deep Learning tasks on edge nodes.

Introduction

The number of IoT devices increased 31% year-over-year to 8.4 billion in the year 2017 and it is estimated that there will be 30 billion devices by 2020. The global market value of IoT is projected to reach $7.1 trillion by 2020. The exponential increase in data generated by IoT devices has made a positive impact by introducing novel and convenient use cases for the end-users. IoT devices have an extensive set of applications in consumer, commercial, industrial, and infrastructure spaces. IoT supports smart systems such as smart cities, smart health care, smart transportation, and smart energy; however, the realization of these smart systems relies on the ability to analyze the massive amount of generated data. The increase of IoT devices at the edge of the network is producing a massive amount of data to be computed at data centers, pushing network bandwidth requirements to the limit.

Traditionally IoT data is sent to the cloud for performing any type of analysis. Cloud Computing provides reliability, computation power, and scalability to fulfill simple as well as complex use cases. Despite the improvements in network technology, data centers cannot guarantee acceptable transfer rates and response times, which could be a critical requirement for many applications. This is due to the remote cloud servers which could be located thousands of miles away from the end devices leading to higher latencies, network congestion, and higher running costs of remote servers. Real-time applications like self-driving cars, voice recognition, speech to text, or human activity recognition (HAR) require prompt responses to have a functional user experience.

Edge computing is a distributed computing paradigm that brings computation and data storage closer to the location where it is needed, to improve response times and save bandwidth. Edge Computing aims to move the computation away from data centers towards the edge of the network, exploiting smart objects, mobile phones, or network gateways to perform tasks and provide services on behalf of the cloud. By moving services to the edge, it is possible to provide content caching, service delivery, storage, and IoT management resulting in better response times and transfer rates. At the same time, distributing the logic in different network nodes introduces new issues and challenges.

DL lies at the heart of most AI applications and involves teaching a computer to learn data representations with different levels of abstraction. DL has been successful in many domains including various vision tasks, natural language processing, and speech recognition. Most well-known applications of AI rely on Cloud Computing because of the superior and scalable compute power. But owing to the limitations of the cloud with real-time applications, we can start to move tasks to the edge and get away with computing the least expensive tasks. Here, we can try to slowly build up to more complex tasks by optimizing over redundant computations. With devices being more computationally better over the years we have reached a point where we can offload some DL/AI compassion logic distributed over multiple edges to simulate a powerful cloud-like behavior.

This study investigates using edge nodes for performing DL tasks by using data reduction techniques, model slicing, and efficient sharing of gradients among edge nodes. The evaluation of the data is done on the HAR dataset which comprises the data from sensors such as gyroscopes, accelerometers mounted on different parts of the human body.

Literature review

Edge computing improves the performance of real-time applications by eradicating the back and forth transmission of data across large network distances because it performs computation close to the data sources. This review explores the training and inferencing of DL models across resource-constrained devices.

Wang et al. [1] presented a survey on mobile edge networks focusing on computing-related issues, edge offloading, and communication techniques for edge-based computing. Wang et al. [1] identified real-time analytics as one of the future directions of edge computing. Recent work in this area has been focused in that direction. For example, A.M. Ghosh et al. [2] proposed embedding of intelligence in the edge with DL. They investigate the latencies of the real-time application by performing data reduction at the edge nodes and transferring reduced data to the remote cloud servers to perform intensive DL tasks. This article explored merging edge and cloud computing for ML with IoT data to reduce network traffic and latencies. Our study aims to put edge computing as the primary infrastructure for driving DL tasks specifically in the case of HAR [3].

X. Wang et al. [4] presented a comprehensive survey on the convergence of edge computing and deep learning. The survey focuses on a broad set of topics and techniques such as optimization of DL models on edge, distributing the training of these models by the use of gradient sharing, and DL inferencing at the edge. It also focuses on the methods used for virtualizing the edge via VMs, containers, and network slicing. K. Bhardwaj [5] et al., in their research on “Toward Lighter containers for the edge” focus on efficient containerization by splitting containerized applications into two parts:- application container and bloat-causing execution environment. Through this, they can achieve significant reductions of the resource pressure at the edge, thus presenting a path toward greater efficiency and scalability for edge computing. We can take insights from this study to offload DL tasks from the cloud to the edge and further optimize it by applying data reduction techniques. Significant research has also been done on efficiently training DL models on large datasets at the edge. C. Hardy [6] et al., in their research on “Feasibility of distributed deep learning via adaptive compression” has proposed a novel solution to reduce the ingress traffic at the parameter server using compressed updates via AdaComp. The techniques mentioned in this research can be used to train DL models at the edge.

Efficient inference techniques for large DNN models have been the recent direction of research in edge computing. Z.Zhou et al. [7], have proposed a framework named Edgent, a framework that leverages edge computing for DNN collaborative inference through device-edge synergy. They exploit two major design ideas:- 1) DNN partitioning that adaptively partitions computation between device and edge for purpose of

coordinating the powerful cloud resource and the proximal edge resource for real-time DNN inference (2) DNN right-sizing that further reduces computing latency via early exiting inference at an appropriate intermediate DNN layer. We can utilize techniques and research specified in this study for efficient inference of DL models on the edge nodes.

Another complementary study on DNN inference is done by S. Han et al.[8], where they proposed a technique called Deep Compression which can reduce the storage requirements by 35x and increase energy efficiency by 7x for DNN models. They follow a three-step approach: pruning of low information nodes; quantization of similar weights by weight sharing and finally applying Huffman coding. This gives significant insights on techniques for deploying DNN on the edge node where resources are constrained.

K. Bhardwaj et al. [9] research on DNN slicing answers an open problem, given a model and a set of edge resources, how to quickly determine the best way to partition the model and create deployment-ready model partitions needed to seamlessly run the model pipeline across the edge and the cloud?. This research targets some of the techniques which could be utilized to successfully partition a DNN model among edge nodes.

Current Approach

Our main motive here is to enhance the performance of the HAR task by using edge nodes as the primary infrastructure for performing DL. We are going to use the MHEALTH dataset and classify human activities using DL inference on edge nodes. Current research has been focused on using edge computing as the inference layer for real-time applications. We are going to utilize some of the techniques mentioned in the literature review as well as apply data reduction using non-linear transformation models like Autoencoders to reduce the data and the latency at the inference layer where the resources are constrained. Our initial test-bed thoughts are to use personal laptops as edge nodes for DL computations and to use the same network for any parameter transfers.

References

[1] Wang et al., “A survey on mobile edge networks: Convergence of computing, caching and communications” IEEE Access, vol. 5, pp. 6757–6779, 2017.

[2] A. M. Ghosh and K. Grolinger, "Edge-Cloud Computing for the Internet of Things Data Analytics: Embedding Intelligence in the Edge With Deep Learning," in IEEE Transactions on Industrial Informatics, vol. 17, no. 3, pp. 2191-2200, March 2021, DOI: 10.1109/TII.2020.3008711.

[3] H. F. Nweke, Y. W. Teh, M. A. A.-G., and U. R. Alo, “Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges,” Expert Syst. Appl., vol. 105, pp. 233–261, 2018.

[4] Wang, X., Han, Y., Leung, V. C. M., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. IEEE Communications Surveys & Tutorials, 22(2), 869–904.

[5] Park, M., Bhardwaj, K., & Gavrilovska, A. (n.d.). Toward lighter containers for the edge. Usenix.Org. Retrieved March 2, 2021, from https://www.usenix.org/system/files/hotedge20_paper_park.pdf

[6] Hardy, C., Le Merrer, E., & Sericola, B. (2017). Distributed deep learning on edge-devices: Feasibility via adaptive compression. 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA).

[7] E. Li, L. Zeng, Z. Zhou and X. Chen, "Edge AI: On-Demand Accelerating Deep Neural Network Inference via Edge Computing," in IEEE Transactions on Wireless Communications, vol. 19, no. 1, pp. 447-457, Jan. 2020, DOI: 10.1109/TWC.2019.2946140.

[8] Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. Retried from https://arxiv.org/pdf/1510.00149.pdf.

[9] Hsu, K.-J., Bhardwaj, K., & Gavrilovska, A. (2019). Couper: DNN model slicing for visual analytics containers at the edge. Proceedings of the 4th ACM/IEEE Symposium on Edge Computing.

[10] http://archive.ics.uci.edu/ml/datasets/mhealth+dataset