The Investigation of Deep Learning Applications in Edge Networks

Xiaokang Yin
Department of Information and Computer Science, Xiao'an Jiatong Liverpool University, Suzhou, 215123, China

Keywords: Deep Learning, Edge Network, Edge Computing, Distributed Training.

Abstract: In the contemporary landscape defined by the ubiquity of data-driven applications and the unceasing demand for real-time processing, the imperative to fuse deep learning with edge computing networks has risen to the forefront. This paper, therefore, undertakes the crucial task of addressing the necessity for such integration while illuminating the formidable obstacles that impede the seamless delivery of services in this context. The research diligently navigates through a myriad of well-established edge computing architectures, evaluating their aptitude for supporting deep learning models. By capitalizing on the latest strides in research, it introduces inventive solutions aimed at harnessing the formidable potential of deep learning at the edge, all while mitigating the challenges posed by resource constraints. The outcomes of this research illustrate substantial enhancements in performance, shining a spotlight on the transformative possibilities that emerge from this synergy. Despite the presence of certain limitations, this work stands as a noteworthy contribution to the ongoing evolution of edge computing, offering the promise of heightened capabilities for both edge devices and applications alike.

1 INTRODUCTION

Due to the rapid expansion of the Internet of Things (IoT), an increasing number of IoT devices are finding utility across diverse applications. As the result of IoT applications expands, it is anticipated that the IoT market will reach $11.1 trillion (Manyika et al 2015), generating 79.4 zettabytes (ZB) of data annually (Badnakhe 2021). A common technique of processing data on cloud server is by deep learning (DL). The traditional cloud computing approach commonly uploads all data to a remote cloud server center for processing and future usage. However, this approach faces several existing challenges. Firstly, reliability is a challenge in cloud computing, since a considerable number of end devices is connected to the backbone network via wireless network. When the wireless is unstable, the cloud computing service should still be reliable (Wang et al 2020). Secondly, latency is a significant obstacle, particularly when large data volumes are simultaneously uploading to data centers, possibly leading to insufficient bandwidth. For instance, a single call to Amazon Web Services can incur a latency of up to 200 ms during DL service execution (Satyanarayanan 2017), which is unacceptable for numerous use cases. Additionally, sustainability is an emerging concern, as transmitting extensive data consumes substantial bandwidth and energy. For instance, power consumption can escalate to 1.04 kWh per gigabyte (GB) of data transferred (Chen and Ran 2019), posing environmental risks as data transmission scales up (Pihkola et al 2018).

In facing all these challenges, many studies have provided applications of deploying DL in edge computing or in End-Edge-Cloud Computing (EECC), which refers to a computing paradigm that schedules the heterogeneous devices and manages the capabilities of on-device computing, edge computing, and cloud computing to fulfill the varied demands posed by resource-intensive and distributed Artificial Intelligence (AI) computations. Various of commonly employed framework have adopted to edge computing and developed applications from this adoption (Murshed et al 2021). For instance, TensorFlow Lite (Google) for Android, iOS have applications of Computer Vision (CV) and speech recognition. Despite of the DL frameworks development, major firms of hardware and system have also developed products for edge computing (Wang et al 2020). For instance, NVIDIA has developed Jetson which is a platform that can run in 5 Watts power. Moreover, empowering DL by edge computing has a widely has seen widespread use in various industries including CV, Natural Language Processing (NLP), network functions and VR/AR (Chen and Ran 2019). The necessity of this paper arises from the widespread

Yin, X.
The Investigation of Deep Learning Applications in Edge Networks.
DOI: 10.5220/0012805500003885

In *Proceedings of the 1st International Conference on Data Analysis and Machine Learning (DAML 2023)*, pages 326-331
ISBN: 978-989-758-705-4

demand for deploying deep learning in the cloud environment. This paper will list some research that focuses on optimize the deep learning using in edge computing. Furthermore, it provides essential background information for future research in this direction.

The main goal of this paper is to provide a comprehensive review of the application of deep learning methods in edge networks. Within this scope, this paper will research into the advantages, barriers, and strategies for enhancing the DL process in edge environments. First, this paper will briefly introduce some widely implemented edge computing structures. Subsequently, this paper will introduce how to deploy DL within the edge computing networks. Finally, this paper will conclude all the findings.

2 EDGE COMPUTING STRUCTURE

The concept of edge computing has now found widespread application in numerous domains (Satyanarayanan et al 2009). In addition to its application in conventional network architectures, edge computing is also being employed within the following networks. In vehicular networks, edge computing has been used as platforms and guarantees security. In smart homes, edge computing can be deployed in routers or hubs to improve the IoT devices performance. In Virtual Reality (VR), edge computing can be used to schedule the resource. This section will introduce six typical structures in different scenarios.

2.1 Cloudlet

Cloudlet is a reliable and resource-rich computing cluster closely connected to the backbone network and aims to serve nearby mobile devices (Satyanarayanan et al 2009). It serves as a pivotal component in a three layers architecture of "Mobile Device-Cloudlet-Cloud." Unlike traditional cloud computing, in which mobile devices directly interact with remote cloud servers, Cloudlet functions as an intermediary layer. It works as a localized computing hub that bring computational resources and services closer to mobile users, reducing latency and enhancing the quality of service. Cloudlets can be implemented on a variety of hardware, including personal computers, affordable servers, or small clusters, and are often strategically deployed in public spaces such as cafes, libraries, or restaurants. Multiple clouds can be networked to form a distributed computing platform, effectively expanding the resources of mobile devices and improving the overall user experience by reducing communication delays and efficient bandwidth utilization.

2.2 PCloud

Personal Cloud (PCloud) is a pioneering technology designed to revolutionize the mobile device experience (Jang et al 2014). It overcomes the inherent limitations of mobile devices such as, limited battery life, restricted form factor, and constrained local data storage. PCloud achieves this goal by integrating nearby and remote cloud resources. Different from vendor-specific solutions, PCloud employs technologies e.g. Cirrostratus extensions for Xen to create a personalized execution environment at the hypervisor level. These environments are governed by policies that not only evaluate network connectivity but also consider device ownership and access permissions, which are securely managed through standard social networking services. PCloud has demonstrated its ability to significantly improve a device's native capabilities, resulting in improved application performance and a better user experience.

2.3 ParaDrop

ParaDrop is an innovative edge computing platform developed by the UW-Madison WiNGS Lab designed to push the boundaries of network capabilities to the edge (Liu et al 2016). At its core, ParaDrop leverages the capabilities of WLAN access points (APs) and wireless gateways through which all end device traffic flows. This strategic location provides ParaDrop with unique contextual insights into network characteristics, including proximity and channel data that is typically lost deep within the network. The platform addresses key challenges in building architecture, programming interfaces, and orchestration frameworks that enable developers to dynamically create, deploy, and decommission edge computing services. Composed of three main components - a versatile hosting substrate within the Wi-Fi AP, a cloud-based backend for orchestrating distributed computing, and a developer-friendly API - ParaDrop provides an ecosystem for third-party developers to Deploy and manage computing capabilities customized to specific needs. With ParaDrop, the edge becomes a dynamic stage for developing cutting-edge services, optimizing network performance and improving user experience in the rapidly evolving IoT and edge computing environments.

2.4 OpenVDAP

The Open Vehicle Data Analytics Platform (OpenVDAP) is a breakthrough solution that will change the landscape of Connected and Autonomous Vehicles (CAVs) (Zhang et al 2018). OpenVDAP is proving to be a key enabler when CAVs are viewed as advanced mobile computers equipped with an array of integrated sensors. It solves the inherent challenges of limited onboard computing resources by leveraging the power of edge computing. The full-stack platform includes an integrated computer and communications engine, a safety-focused vehicle operating system, an edge-ready application library, and intelligent workload offloading and scheduling policies. OpenVDAP enables CAV to dynamically evaluate service status, compute requirements, and optimal offload targets, ensuring that real-time services run with minimal latency and bandwidth consumption. Specifically, OpenVDAP is an open-source initiative that provides free access to APIs and real vehicle data and facilitates collaboration between researchers and developers to drive innovation within the CAV ecosystem.

2.5 Vigilia

Vigilia is a breakthrough system designed to improve the security and privacy of the smart home IoT ecosystem. In response to the growing popularity of these devices and the growing security concerns that come with them, Vigilia is taking a proactive approach to significantly reduce the attack surface (Trimananda et al 2018). This is accomplished by enforcing a default access deny policy and establishing fine-grained control over device access. Unlike closed and proprietary systems, Vigilia offers an open implementation that allows users to customize security measures while maintaining the flexibility and convenience of smart home technology. Vigilia extends its protection by maximizing restrictions on communication between devices based on device network permissions, ensuring only authorized interactions occur. Additionally, this innovative system outperforms other IoT defense solutions while minimizing performance overhead. Vigilia enables homeowners to enjoy the benefits of smart home IoT devices while preserving their privacy and protecting their homes from potential security breaches.

2.6 MUVR

Multi-User Virtual Reality (MUVR) is an innovative system designed to revolutionize the way virtual reality experiences are delivered from mobile devices via edge cloud rendering (Li and Gao 2018). The fundamental challenge it faces is to effectively utilize the redundant VR frames generated for different users. MUVR achieves this by adaptively reusing redundant frames that are dynamically determined by the edge cloud. The edge cloud stores previous VR frame rendering results for future user sessions, optimizing computing resources. Following the creation of a VR frame, MUVR optimizes data transfer by efficiently reusing redundant pixels from preceding frames, transmitting solely the distinctive components to the mobile device. MUVR is implemented on the Android operating system and the Unity VR engine, which can reduce the computing load of the edge cloud by more than 90% and reduce the data sent to mobile devices by more than 95%. This breakthrough technology improves multi-user VR experiences, making them more accessible and efficient while ensuring high-quality immersive interactions.

3 DL AT EDGE

Considering the diversity of edge computing networks, the DL architectures employed within edge computing networks are equally varied. For instance, Chen et al (2019), categorizes the architectures into three distinct classes. The first architecture is on-device computation, which processes the data on the end device. The second architecture is edge server computation, which uploads the data to one or more edge server for computation and download the result to the end device. The third architecture is computing across edge devices, which refers to jointly using the edge server and end device for the computation. In addition to these three architectures, private inference is also a significant and independent architecture of the DL models. This is due to the reason that architectures involve uploading user data to edge servers. Therefore, in order to prevent the leakage of user privacy, it necessitates the implementation of a private system.

3.1 On-Device Computation

On-Device Computation is an effective approach for reducing latency, meanwhile fewer transmissions result in lower energy consumption. However, addressing resource constraints is imperative, which numerous research efforts have contributed solutions to this challenge.

One of the primary approaches involves using fewer parameters during the model design phase.

Howard et al. (Howard et al) introduces MobileNets that leverage depth-separated convolutions and provide smaller, faster alternatives through width and resolution multipliers, outperforming existing models in size, speed, and accuracy across a variety of tasks, with plans to use TensorFlow models more broadly to share and discover.

The second approach involves compressing existing models, and this specific method can be further categorized into three techniques: parameter quantization, parameter pruning, and knowledge distillation. Additionally, some research combines these three techniques for compression purposes. Han et al (2017) solve high computation and memory demands challenge in LSTM-based speech recognition by introducing an efficient solution: load-balance-aware pruning and quantization techniques, parallel processing scheduling, and the hardware architecture Efficient Speech Recognition Engine (ESE). This architecture outperforming CPUs and GPUs by 43x and 3x in speed.

The third approach involves optimizing through hardware co-design. Many mobile device chip manufacturers have tailored their products for deep learning optimization and provide developers with software development kit (SDK). Alzentot et al (2017) provides an approach of leveraging heterogeneous CPU and GPU resources on standard Android devices for deep learning tasks and leverage the RenderScript framework to improve TensorFlow. it tightly integrated system enables machine learning engineers to seamlessly access mobile device resources. They analyze the performance trade-offs of different Android phone models and compare GPU-accelerated neural network operations with CPU-only execution. The results show significant speedup improvements for models with large matrix multiplications, highlighting the significant advantages of GPU support on mobile devices.

3.2 Edge Server Computation

Despite the performance optimizations offered by the aforementioned methods, the computational capabilities of end-user devices remain limited. Offloading computation to edge servers is a favorable choice, as edge servers not only possess greater computational power but are also closer to users and therefore offer enhanced reliability compared to cloud servers. Despite these advantages, latency remains a challenge that cannot be overlooked. To mitigate latency concerns, prior research have proposed two main directions: data preprocessing and edge server scheduling optimization.

Chen et al (2015) have developed a system called Glimpse which is a real-time object detection system for mobile devices that improves detection accuracy by leveraging active video cache, leveraging hardware facial recognition support for 1.8-2.5x improvement in precision, and continuous, precise road sign recognition for coverage from 75% to 80%, which would otherwise not be possible without the techniques used (precision from 0.2% to 1.9%).

Jiang et al (2018) have developed Mainstream which is a novel video analytics system, optimizes concurrent applications on stationary resources by leveraging partial sharing of DNN computations through transfer learning and dynamically balancing specialized DNNs to achieve per-frame accuracy and underlying sharing models, resulting in significant improvements of up to 47% in averages event detection F1 scores compared to static approaches and a remarkable 87 times compared to fully independent DNNs per application.

3.3 Collaborative Computation

Following the discussion of both end device and edge server computation, it is natural that many studies have explored collaborative computation between edge servers and terminal devices. Collaborative computation can be further categorized into four approaches: performing all computations on either edge servers or terminal devices, offloading partial computations to edge servers, end-cloud-edge collaboration, and distributing computations across different edge servers.

Binary computation using either edge servers or terminal devices is an approach that determines whether to offload computation to edge servers based on factors such as latency, energy consumption, resource utilization, and others. For instance, this approach is exemplified by MCDNN (Multi-Column Deep Neural Networks). Han et al (2016) introduce approximate model scheduling, a method to efficiently process heterogeneous requests by trade-off between classification accuracy and resource utilization and on-device/cloud execution optimization. The research optimizes resulting in significant reductions in resource consumption while maintaining effective performance under various operating conditions.

Partial offloading is a method that involves offloading specific layers or portions of a neural network to edge computing resources. An example of this approach is MAUI. MAUI (Eduardo et al 2010) achieves balance by leveraging managed code environments to minimize developer involvement

while optimizing runtime energy savings. This provides tangible benefits such as orders of magnitude lower power consumption for resource-intensive applications, improved refresh rates for latency-sensitive games, and overcoming smartphone limitations in voice-based language translation applications by remotely triggering unsupported components.

While cloud servers may be farther from the end-users, potentially increasing latency, their abundant computational resources can still effectively reduce the overall processing time if resource allocation is done optimally. An example illustrating this concept is Distributed Deep Neural Networks (DDNN). DDNNs (Teerapittayanon et al 2017) can scale neural network size and geographic coverage, improving sensor fusion, fault tolerance, and data protection. By assigning DNN sections to this hierarchy and training them together, DDNNs minimize communication and resource consumption and support automatic sensor coupling and fault tolerance. As a proof of concept, DDNNs leverage the geographic diversity of sensors, improving object detection accuracy and reducing communication costs by more than twenty times compared to traditional processing of raw sensor data in the cloud.

The final approach involves distributing computations across different edge servers, and a representative study demonstrating this concept is DeepThings (Zhao et al 2018). it optimizes memory utilization through scalable Fused Tile Partitioning convolutional layers, provides dynamic load balancing through distributed work stealing, and improves data reuse and latency reduction through innovative work scheduling and achieves a scalable CNN inference speed of 1.7 to 3.5 times on 2 to 6 edge devices with less than 23 MB of memory each, which outperforms existing methods.

3.4 Private Inference

When performing computations on servers with user data, privacy concerns become paramount. GAZELLE (Chiraag et al 2018) and DeepSecure (Darvish et al 2018) are two effective methods for encrypting user privacy data without impeding the inference process of deep learning networks.

4 CONCLUSION

In summary, this research has underscored the pivotal role played by deep learning in augmenting the potential of edge computing networks. It has responded to the pressing demand for streamlined, real-time processing capabilities at the edge and examined the viability of incorporating deep learning models into established edge computing frameworks. The research has introduced novel approaches, including fine-tuned neural network designs and resource-efficient training methodologies, which have yielded substantial enhancements in performance across various edge applications.

These findings have profound implications for areas such as autonomous systems, the Internet of Things and healthcare, where low-latency decision-making is paramount.

REFERENCES

J. Manyika, M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin, and D. Aharon. The Internet of Things: Mapping the Value Behind the Hype. Technical Report. McKinsey and Company. pages 1-3, (2015)

R. Badnakhe, IoT Is Not a Buzzword But Necessity, Feb. (2021) [online] Available: https://www.iotcentral.io/blog/iot-is-not-a-buzzword-but-necessity. Retrieved September 10, 2023.

X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan and X. Chen, Convergence of Edge Computing and Deep Learning: A Comprehensive Survey, in IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 869-904, Secondquarter (2020)

M. Satyanarayanan, The emergence of edge computing, Computer, vol. 50, no. 1, pp. 30-39, (2017)

J. Chen and X. Ran, Deep Learning With Edge Computing: A Review, in Proceedings of the IEEE, vol. 107, no. 8, pp. 1655-1674, Aug. (2019)

H. Pihkola, M. Hongisto, O. Apilo, M. Lasanen. Evaluating the Energy Consumption of Mobile Data Transfer—From Technology Development to Consumer Behaviour and Life Cycle Thinking. Sustainability (2018)

M. G. Murshed et al. Machine learning at the network edge: A survey. ACM Computing Surveys (CSUR) 54, no. 8 (2021): 1-37

M. Satyanarayanan, P. Bahl, R. Caceres and N. Davies, The Case for VM-Based Cloudlets in Mobile Computing, in IEEE Pervasive Computing, vol. 8, no. 4, pp. 14-23, Oct.-Dec. (2009)

M. Jang, K. Schwan, K. Bhardwaj, A. Gavrilovska and A. Avasthi, Personal clouds: Sharing and integrating networked resources to enhance end user experiences, IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, Toronto, ON, Canada, (2014) pp. 2220-2228

P. Liu, D. Willis and S. Banerjee, ParaDrop: Enabling Lightweight Multi-tenancy at the Network's Extreme Edge, 2016 IEEE/ACM Symposium on Edge Computing (SEC), Washington, DC, USA, (2016) pp. 1-13

Q. Zhang et al., OpenVDAP: An Open Vehicular Data Analytics Platform for CAVs, 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, (2018) pp. 1310-1320

R. Trimananda, A. Younis, B. Wang, B. Xu, B. Demsky and G. Xu, Vigilia: Securing Smart Home Edge Computing, 2018 IEEE/ACM Symposium on Edge Computing (SEC), Seattle, WA, USA, (2018) pp. 74-89

Y. Li and W. Gao, MUVR: Supporting Multi-User Mobile Virtual Reality with Resource Constrained Edge Cloud, 2018 IEEE/ACM Symposium on Edge Computing (SEC), Seattle, WA, USA, (2018) pp. 1-16

A. G. Howard et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications

H. Song, et al. Ese: Efficient speech recognition engine with sparse lstm on fpga. Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (2017).

A. Moustafa, et al. Rstensorflow: Gpu enabled tensorflow for deep learning on commodity android devices. Proceedings of the 1st International Workshop on Deep Learning for Mobile Systems and Applications (2017).

C. T. Yu-Han, et al. Glimpse: Continuous, real-time object recognition on mobile devices. Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (2015).

A. H. Jiang et al. Mainstream: Dynamic {Stem-Sharing} for {Multi-Tenant} Video Processing. 2018 USENIX Annual Technical Conference (USENIX ATC 18), (2018).

H. Seungyeop, et al. Mcdnn: An approximation-based execution framework for deep stream processing under resource constraints. Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, (2016).

C. Eduardo, et al. Maui: making smartphones last longer with code offload. Proceedings of the 8th international conference on Mobile systems, applications, and services, (2010).

S. Teerapittayanon, B. McDanel and H. T. Kung, Distributed Deep Neural Networks Over the Cloud, the Edge and End Devices, 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, (2017), pp. 328-339

Z. Zhao, K. M. Barijough and A. Gerstlauer, DeepThings: Distributed Adaptive Deep Learning Inference on Resource-Constrained IoT Edge Clusters, in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 11, pp. 2348-2359, Nov. (2018)

J. Chiraag, et al. GAZELLE: A low latency framework for secure neural network inference. 27th USENIX Security Symposium (USENIX Security 18). (2018).

R. Bita Darvish et al. Deepsecure: Scalable provably-secure deep learning. Proceedings of the 55th annual design automation conference (2018).