Article, Review

A Generic Framework for Distributed Deep Neural Networks over the Cloud, the Edge, and End Devices

Ahmad Vegah[1],

[1] ahmadvegah@gmail.com

Abstract: This paper proposes deep neural networks (DNNs) over distributed computing hierarchies, consisting of the cloud, the edge and end devices. Although a deep neural network (DNN) can accommodate inference in the cloud, a distributed deep neural network (DDNN) supported by scalable distributed computing hierarchy is advantageous. For a DNNN can scale up in neural network size as well as scale out in geographical span and it allows fast and localized inference using shallow portions of the neural network at the edge and end devices. In implementing a DDNN, we map sections of a DNN onto a distributed computing hierarchy.

Keywords: Edge Computing; Internet of Things; Neural Networks.

1. Introduction

The framework of a large-scale distributed computing hierarchy has assumed new significance in the emerging era of IoT. It is widely expected that most of data generated by the massive number of IoT devices must be processed locally at the devices or at the edge, for otherwise the total amount of sensor data for a centralized cloud would overwhelm the communication network bandwidth. To this end, the goal of this project is to explore the development of heterogeneous mobile edge computing platform (Figure 1) that performs real-time Big Data analytics and deep learning with low-latency computing at the network edge.

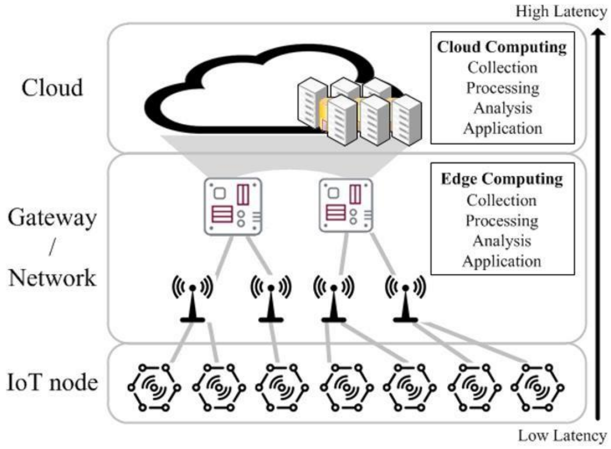


Figure 1. Overview of Mobile Edge Computing Platform.

With the rise in Big Data, Cloud Computing, and the emergence of IoT Services, it is becoming essential to collate, process, manipulate data at the network edge to assist with intelligent re-orchestration. Moreover, key technologies for the creation of autonomous vehicles, smart homes and healthcare will be powered by thousands of IoT sensors and end devices, but questions such as: – What does it take to create and manage a new end point devices? (compute and storage resources) How do you identify an end service canonically across infrastructure and platform services? (identity e.g. scalability) How do you allocate resources for an end device? (resource provisioning e.g. bandwidth) What does it take to operate an end device? (monitoring e.g. dependability) How do you measure resource utilization and cost of operating an end device? (metering and chargeback e.g. energy efficiency) These questions persist regardless of an organization's IoT strategy. Managing autonomous and intelligent re-orchestration end device (i.e., creating them, provisioning resources, deploying, metering, charging, and deprecating) at scale proves to be challenging in addressing these questions.

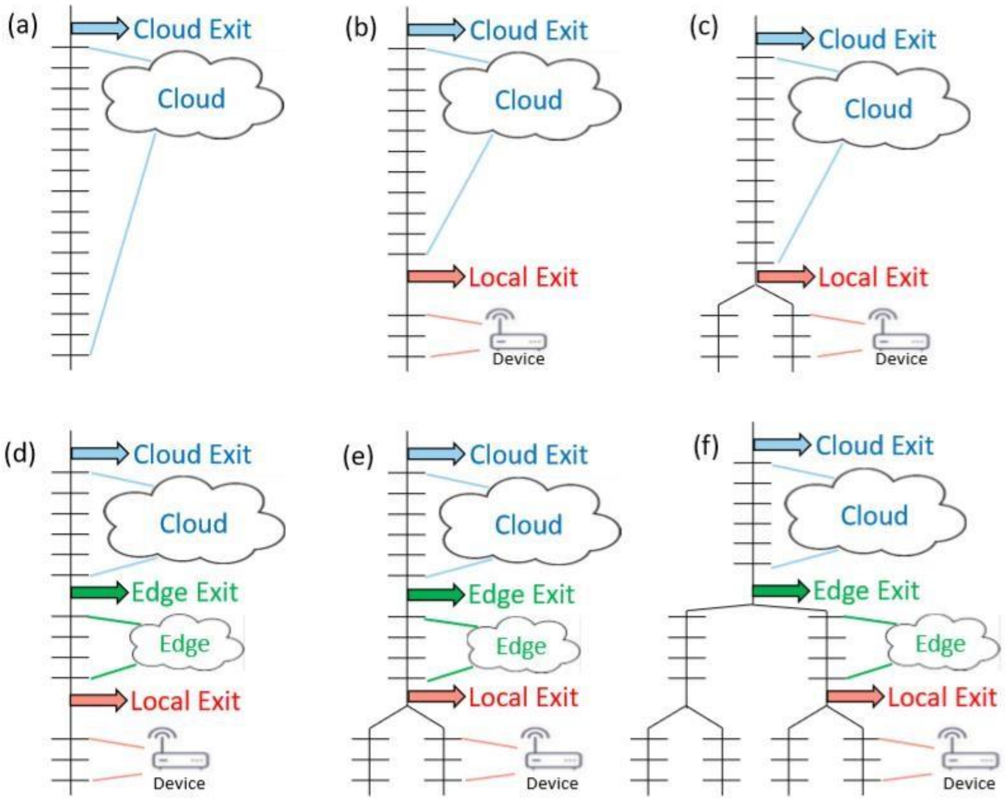
The rise in Internet of Things (IoT) devices as well as a dramatic increase in the number of end devices provides appealing opportunities for machine learning applications at the network edge as they are often directly connected to sensors (e.g., cameras, microphones, gyroscopes) that capture a large quantity of input data. However, the current approaches to machine learning systems on end devices are either to offload input sensory data to DNNs in the cloud, with the associated privacy concerns, latency and communication issues or perform Figure 1. Overview of Mobile Edge Computing Platform A Generic Framework for Distributed Deep Neural Networks over the Cloud, the Edge and End Devices directly on the end device using simple Machine Learning (ML) models leading to reduced system accuracy. The use of hierarchically distributed computing structures consisted of the cloud, the edge and end devices addresses these shortcomings by providing system scalability for large-scale intelligent tasks in distributed IoT and end devices which supports coordinated central and local decisions.

To this end, in implementing a DDNN, we map sections of a single DNN onto a distributed computing hierarchy and by jointly training these sections, the DDNNs can effectively address some of the challenges mentioned before. Moreover, via distributed computing, DDNNs enhance sensor fusion, data privacy and system fault tolerance for DNN applications. As a proof of concept, we show a DDNN can exploit geographical diversity of end point devices. The contributions envisaged are:

1. Literature review of existing methods for data-driven adaptation of end devices and requirements analysis of the needs at scale.
2. A generic DDNN framework and its implementation that maps sections of a DNN onto a distributed computing hierarchy.
3. A joint training method that minimizes communication and resource usage for devices and maximizes usefulness of extracted features which are utilized in the cloud, while allowing low-latency classification via early exit for a high percentage of input samples.
4. A mobile application to extensively test DDNN framework for orchestration and reorchestration of end point devices at scale.
5. A microservices architecture that implements governance at scale

2. Materials and Methods

One approach is the combination of small neural network (NN) model on end devices and a larger NN model in the cloud. The small NN model on the end device performs initial feature extraction and classification if confident or otherwise the end device can draw on the large NN model in the cloud. However, this approach comes with certain challenges such limited memory and battery life one end devices such as sensors as well as multiple models at the cloud, edge and end device need to be learnt jointly which may incur huge communication costs in coordinated decision making. To address these concerns under the same optimization framework, it is desirable that a system could train a single end-to-end model, such as a DNN, and partition it between end devices and the cloud, to provide a simpler and more principled approach. DDNN maps a trained DNN onto heterogeneous physical devices distributed locally, at the edge, and in the cloud. Since DDNN relies on a jointly trained DNN framework at all parts in the neural network, for both training and inference, many of the difficult engineering decisions are greatly simplified. Figure 2 provides an overview of the DDNN architecture. The configurations presented show how DDNN can scale the inference computation across different physical devices. The cloud based DDNN in (a) can be viewed as the standard DNN running in the cloud as described in the introduction. In this case, sensor input captured on end devices is sent to the cloud in original format (raw input format), where all layers of DNN inference is performed.



(a) Cloud-based DDN

(b) DDNN over cloud and device

(c) DDNN over cloud and geographically distributed devices

(d) DDDN over cloud, edge, and device

(e) DDNN over cloud, edge and geographically distributed devices

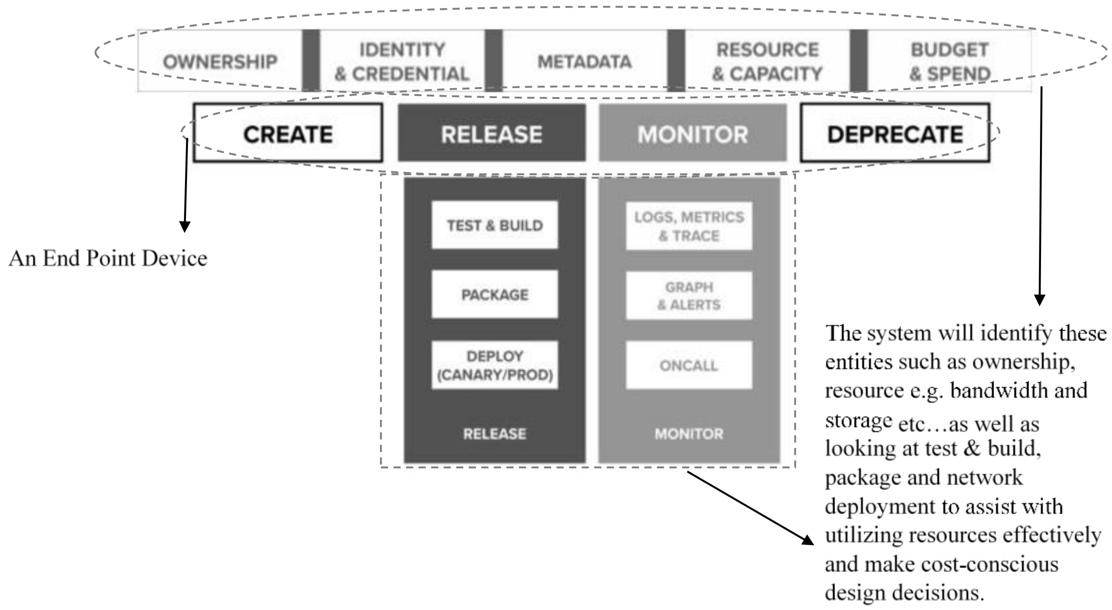
(f) DDDN over cloud and geographically distributed edges and device

Figure 2. Overview of the DDNN architecture. The vertical lines represent the DNN pipeline, which connects the horizontal bars (NN layers). (a) is the standard DNN (processed entirely in the cloud), (b) introduces end devices and a local exit point that may classify samples before the cloud, (c) extends (b) by adding multiple end devices which are aggregated together for classification, (d) and (e) extend (b) and (c) by adding edge layers between the cloud and end devices, and (f) shows how the edge can also be distributed like the end devices.

3. Results

Through DDNNs, we undertook design and development of visual-intelligence and autonomous robotics platform based on the H2020 EoT (Eyes of Things research), Raspberry Pi (x19), Neural Compute Stick (x5), Nvidia Jetson TX1 (x2), Microsoft Kinect V2, Intel Aero Platform, associated machine vision, communications and motor-control libraries and the CUDA and Tensor Flow deep-learning framework. In this work, computer vision and motor-control libraries are used together to design and develop a real-time visually and contextually intelligent live, dense three-dimensional (3D) map of an area.

Governance in Network Management and Resource Allocation for End Point Devices



An End Point Device

The system will identify these entities such as ownership, resource e.g. bandwidth and storage etc…as well as looking at test & build, package and network deployment to assist with utilizing resources effectively and make cost-conscious design decisions.

References

1. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.

2. K. Skala, D. Davidovic, E. Afgan, I. Sovic, and Z. Sojat, "Scalable distributed computing hierarchy: Cloud, fog and dew computing," Open Journal of Cloud Computing (OJCC), vol. 2, no. 1, pp. 16–24, 2015.

3. S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in Hot Topics in Web Systems and Technologies (HotWeb), 2015 Third IEEE Workshop on. IEEE, 2015, pp. 73–78.

4. J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K.

5. Yang, Q. V. Le et al., "Large scale distributed deep networks," in Advances in neural information processing systems, 2012, pp. 1223–1231.

6. J. Dean, "Large scale deep learning," in Keynote GPU Technical Conference, vol. 3, 2015, p. 2015.

7. Benedict, M.B. c2016. How to manage the lifecycle of Micro-Services at scale? [Online]. [29 April 2017]. Available from: http://www.cloudexpoeurope.com/2017programme/micro-services-lifecycle-management-at-twitter-scale.