When Deep Learning Meets Edge Computing

Yutao Huang*, Xiaoqiang Ma† *, Xiaoyi Fan*, Jiangchuan Liu‡, Wei Gong*,
*School of Computing Science, Simon Fraser University, Canada
† School of Electronic Information and Communications, Huazhong University of Science and Technology, China
‡ College of Natural Resources and Environment, South China Agricultural University, China

***Abstract*—The state-of-the-art cloud computing platforms are facing challenges, such as the high volume of crowdsourced data traffic and highly computational demands, involved in typical deep learning applications. More recently, *Edge Computing* has been recently proposed as an effective way to reduce the resource consumption. In this paper, we propose an *edge learning* framework by introducing the concept of *edge computing* and demonstrate the superiority of our framework on reducing the network traffic and running time.**

I. Introduction

With the advances in personal computing devices and the deep penetration of high-speed mobile networking, today's crowdsourced applications are geo-distributed globally, and the crowdsourced data are highly heterogeneous. Crowdsourced systems obtain resources by collecting numerous raw data from a large crowd of contributors, and thus are necessarily deployed on cloud service platforms to benefit from on-demand self-service, unlimited resource pooling, and dynamic scalability. Modern deep learning technique has quickly risen to become a key component in various crowdsourced applications, including speech recognition [1], recommendation systems [2], and video classification [3]. Yet, the high volume of crowdsourced data traffic and highly computational demands involved in typical deep learning applications, e.g., face recognition and human tracking in camera networks, put significant pressure on the infrastructure of state-of-the-art cloud computing paradigm.

The concept of *Edge Computing* has been recently proposed to complement cloud computing by performing certain data processing tasks at the edge of the network. This new generation of paradigm has shown a significant reduction in the system running time, memory cost, and energy consumption for a broad spectrum of big data applications [4], [5], as compared to conventional cloud computing.

To cope with the huge network traffic and high computational demands, as well as to improve the system response time, we propose *Edge Learning*, a complementary service to existing cloud computing platforms, seeking to combat the challenges in crowdsourced deep learning applications. *Edge learning* performs data pre-processing and preliminary learning at the edge of the network; the raw data in the local regions are processed on edge servers to reduce the network traffic, so as to speed up the computation in data centers. We have implemented an *edge learning* prototype with one edge server and one cloud cluster consisting 2 GPU-accelerated computing servers. Experiments demonstrate that the edge learning design reduces the network traffic by 80% and the running time by 69% over state-of-the-art cloud-based solutions.

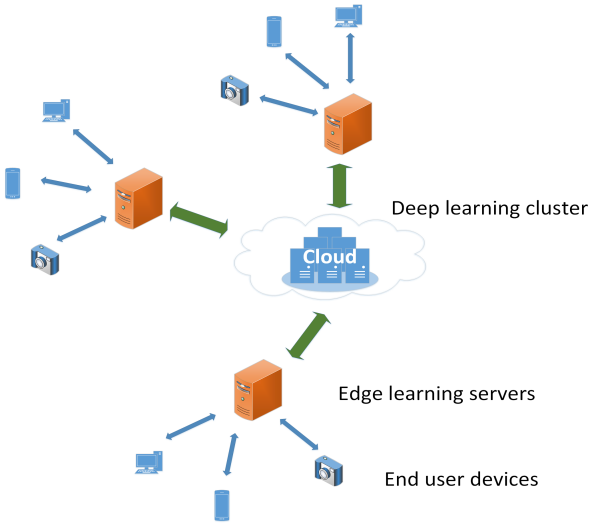


Fig. 1. The edge learning framework

II. The Edge Learning Framework Design and Evaluation

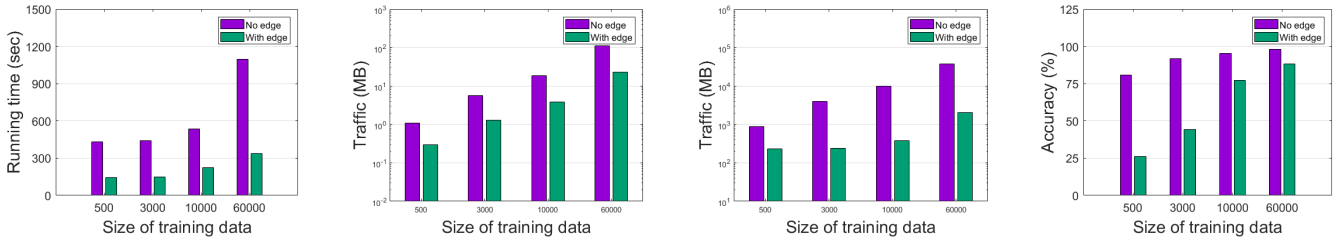
A. Framework Design

As illustrated in Fig. 1, we propose the *edge learning* framework, which consists of three major components: end user devices, edge learning servers, and deep learning clusters on remote cloud. In the edge learning framework, end users devices, e.g., mobile phones, cameras, and Internet-of-Things devices, crowdsourced data, which can be noisy and highly redundant. The edge learning servers gather the massive raw data from end users and perform pre-processing and preliminary learning techniques, so as to filter out the noises and extract key features of the raw data. The deep learning cluster, equipped with powerful and scalable GPU resources, executes the deep learning tasks, e.g., Convolutional Neural Networks (CNN) and Long Short-term Memory (LSTM) Networks, based on the outputs from edge servers.

The design of edge learning has two major advantages.

1) The edge learning servers alleviate the workload of the network infrastructure, comparing to state-of-the-art cloud computing architecture. The raw data can be preprocessed

978-1-5090-6501-1/17/$31.00 ©2017 IEEE



(a) Running time vs. the size of training data (b) Data transmitted to the cloud vs. the size of training data (c) Traffic produced among the cluster vs. the size of training data (d) Accuracy vs. the size of training data

Fig. 2. The learning performance under different size of training data

by dimensionality reduction methods such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to eliminate the noise and redundancy yet preserve the important information. The edge learning servers can also further apply the advanced learning-based technique, e.g., autoencoder, to extract common features from the collected regional data.

2) The network latency between end user devices and the edge servers is significantly shorter than that between the end user device and the cloud server, since such local edge servers are close to the end user device. The model trained on the cloud can be deployed on edge servers to provide timely services to end users, and the new data can be continuously transferred to the cloud to further update the model.

B. Edge MNIST training: A Case Study

We next present the real-world experiment results in a typical virtualized environment. Our deep learning cluster consists of three customized desktops, each equipped with Intel Core i7-6850K CPU and dual NVIDIA GeForce GTX 1080 Ti GPUs. Deep learning works on the cluster with CNN and LSTM classifiers, which is implemented in Apache Spark framework and Keras with Tensorflow backend.

Our edge server works on a Dell server (OPTIPLEX 7010), equipped with an Intel Core i7-3770 3.4 GHz quad core CPU, 16 GB 1333 MHz DDR3 RAM. The edge server is responsible for performing Principal Component Analysis (PCA) [7] on the public MNIST dataset [6]. PCA is known to efficiently reduce the dimensionality of a data set to some principle components. In the experiments, each picture with 28×28 pixels will be performed PCA, and then uploaded to the deep learning cluster. The client application runs on a Google Nexus 9 Android Tablet, which uploads a sequence of handwriting images from the MNIST dataset.

remote traffic and running time. In the following we present

In the experiments, we compare our edge learning framework with the state-of-the-art cloud distributed computing scheme. The default Apache Spark Distributed Deep learning system serves as the baseline. Apache Spark is one of the most popular frameworks for cloud distributed computing. For large-scale computation, a single device on the remote cloud is usually not powerful enough to support the computing process, which makes the necessity of distributed computing. In general, our design outperforms with significantly reduced our detailed analysis with respect to each system parameter, with the size of training data from 500 to 60,000. Fig. 2 (a) shows that the edge server can significantly reduce the job running time. Fig. 2 (b) demonstrates that the edge server performing PCA algorithms is capable to significantly reduce network traffic. Fig. 2 (c) illustrates that the inner traffic load in the cloud cluster. These improvements come from the dimensionality reduction of input data. Fig. 2 (d) shows the impact of the training data size. We can see that with larger dataset size, the learning accuracy gets improved. Although the PCA pre-processing at the edge servers lowers the accuracy, the accuracy can be improved with more training data, and the running time has a significant reduction. Moreover, with more training data, the accuracy with the edge computing will get close to the cloud-based deep learning. When there are 60,000 examples in the training data, the accuracy can reach 90%.

III. Future Work

In the future work, we plan to deploy more advanced learning techniques like autoencoder on edge servers, and compare the performance of different techniques. Furthermore, we will implement our edge learning framework for a large scale distributed camera network, which consists of hundreds of smart cameras as end devices.

References

[1] Hinton, Geoffrey, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." IEEE Signal Processing Magazine 29.6 (2012): 82-97.

[2] Elkahky, Ali Mamdouh, Yang Song, and Xiaodong He. "A multi-view deep learning approach for cross domain user modeling in recommendation systems." Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2015.

[3] MLA Karpathy, Andrej, et al. "Large-scale video classification with convolutional neural networks." Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2014.

[4] Pang, HweeHwa, and K-L. Tan. "Authenticating query results in edge computing." Data Engineering, 2004. Proceedings. 20th International Conference on. IEEE, 2004.

[5] Li, Dawei, et al. "DeepCham: Collaborative Edge-Mediated Adaptive Deep Learning for Mobile Object Recognition." Edge Computing (SEC), IEEE/ACM Symposium on. IEEE, 2016.

[6] L. Deng, "The MNIST database of handwritten digit images for machine learning research", IEEE Signal Process. Mag., vol. 29, no. 6, 2012.

[7] Hfig:frameworkfig:performancefig:performanceolland, Steven M. "Principal components analysis (PCA)." Department of Geology, University of Georgia, Athens, GA (2008): 30602-2501.