COINRG   C. Li, H.Yang, Z. Sun
Internet-Draft   Beijing University of Posts and Telecommunications
Intended status: Standards Track   S. Liu
Expires: 11 January 2023   China Mobile Research Istitute
                                                                 H. Zheng
                                                      Huawei Technologies
                                                             11 July 2022

Distributed Learning Architecture based on Edge-cloud Collaboration
draft-li-coinrg-distributed-learning-architecture-00

Abstract

This document describes the distributed learning architecture based on edge-cloud collaboration.

Table of Contents

1. Introduction 2
   1.1. Requirements Language 3
2. Scenarios 3
   2.1. Federated Learning 3
   2.2. Model Parallelism-Based Distributed Training 4
3. Problem Statement 4
4. Distributed Learning Architecture based on Edge-cloud Collaboration. 5
   4.1. Model Splitting 5
   4.2. Distributed Learning Architecture based on Edge-cloud Collaboration 5
5. Manageability Considerations 7
6. Security Considerations 7
7. IANA Considerations 7
8. References 7
Acknowledgments 8
Authors' Addresses 8

1. Introduction

The rapid growth of Internet of Things (IoT) and social networking applications has led to exponential growth in the data generated at the edge of the network. The ability of a single edge node to process data cannot meet the needs of IoT services. Edge-cloud collaboration technology emerged as the times require, offloading some computing tasks at the edge to the cloud. Service latency includes edge-side computing latency and service transmission latency, which is crucial to model quality in distributed training based on edge-cloud collaboration, because it affects the synchronization of training. How to ensure these two delays has become a key factor in improving the quality of the model.

The distributed learning architecture based on edge-cloud collaboration has become a solution to the above problems. The training tasks are flexibly deployed to edge devices and cloud devices through model parallelism, and deterministic network technology is used to ensure uniform edge training delay and model transmission delay, and then distributed training technology is used to generate a unified model.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Scenarios

With the proliferation of mobile and IoT devices, the data required forartificial intelligence model training is increasingly generated at the edge of the network. Distributed edge model training has become the main means to achieve edge intelligence.

2.1. Federated Learning

Federated learning is a special case of data parallelism in distributed training, which is dedicated to solving the privacy problem in distributed training. Federated learning is an emerging but promising approach to preserve privacy when training AI models based on data produced by multiple clients. Federated learning does not require aggregating raw data into a centralized data center for training. Instead, the raw data is collected through network edge devices (such as base stations), then trained locally, and aggregated models from edge devices on the server Train a shared model.

Federated learning is affected by a variety of factors, which can affect the accuracy of federated learning. These factors are related to network attributes, including computing power, bandwidth, and delay of edge devices. When the above conditions can meet the requirements, federated learning can achieve higher training accuracy.

2.2. Model Parallelism-Based Distributed Training

Model parallelism means that the data on each device is complete and consistent, and the model is divided into various devices. Each device only has a part of the model, and each device is responsible for training a part of the model. Models put together are the complete model.

Model parallelism is affected by several factors, which can affect the efficiency of training. These factors are related to network attributes, including computing power, bandwidth, and delay of edge devices. When the above conditions can meet the requirements, model parallelism will be like pipeline production, with high training efficiency and model accuracy.

3. Problem Statement

The computing power of edge nodes is small and cannot meet the model training in the case of a large amount of data. Therefore, distributed training based on edge-cloud computing power coordination has become an important means to realize edge intelligence.

In order to obtain good training results, distributed training based on edge-cloud collaboration requires the deterministic performance of the underlying optical network. The synchronization of distributed training is achieved through deterministic performance. At this time, it is necessary to synchronize the edge training delay and model transmission delay. These require the support of various quality factors, such as computing resources, end-to-end delay, delay jitter, bandwidth. The above factors can be achieved by deterministic optical networks.

4. Distributed Learning Architecture based on Edge-cloud Collaboration

At present, the common method is to realize the training of distributed models by combining model splitting and distributed training.

4.1. Model Splitting

Since each layer of an artificial intelligence model has independent inputs and outputs, a model can be split into multiple sub-models for independent training, where the training layer that links the sub-models is called a segmentation layer. This method provides the realization basis for edge-cloud collaborative training.

In order to maintain the synchronization in the data parallel process, the training time of all edge nodes in this paper needs to be consistent. Before the model is divided, the computing resources required by each layer are first calculated, and the model is splited according to the remaining situation of the current computing resources. The model splitting in this document is dynamic, that is, the splitting scheme of the model may be different for each round of training.

4.2.Distributed Learning Architecture based on Edge-cloud Collaboration

Edge devices provide services to nearby users, and collect data generated in the process of providing services in real time to form edge data sets. After the edge collects enough edge data, it sends a model training request to the cloud node. After the cloud node receives all training requests from the edge device, it prepares for model training, which is divided into data standardization and model determination. Model determination: The cloud node determines the model architecture according to the training task and sends it to all edge devices. In order to reduce the amount of computation in the training process, the dataset needs to be standardized before training. Common methods include normalization, log transformation, and regularization. The data standardization method is determined by the cloud node, and the standardized algorithm is sent to the edge device, and the edge device processes the edge data set according to the standardized algorithm.

After the preparations are completed, enter the model training phase. In order to ensure the quality of model training, it is necessary to ensure the consistency of training delay in all edge devices and the consistency of model transmission delay. Training delay and transmission delay are set by cloud nodes based on historical experience. At present,the computing power network calculation can calculate the training time of the training task. Therefore, in terms of the training delay, combining the model splitting and the computing power network can calculate the training time of each layer of the model, and then calculate the edge device according to the training delay. The number of layers to train. At the same time, it is also possible to determine the size of the data volume of the segmentation layer, and then reserve bandwidth for model transmission in advance based on the determined network technology. The edge device finishes training the pre-training model, and after the training is completed, sends the segmentation layer of the pre-training model to the cloud node. After receiving the segmentation layer of the model, the cloud node completes the subsequent training of the model, and then updates the model weights according to the back-propagation algorithm. So far, the edge device and the cloud node have completed a round of model training. After every 5 rounds of training, all edge devices generate a global model through distributed learning, and edge devices continue to train according to the local model according to the global model until the model accuracy meets the requirements.

5. Manageability Considerations

TBD

6. Security Considerations

TBD

7. IANA Considerations

This document requires no IANA actions.

8. References

TBD

Acknowledgments

TBD

Authors' Addresses

Chao Li
Beijing University of Posts and Telecommunications

Email: lc96@bupt.edu.cn

Hui Yang
Beijing University of Posts and Telecommunications

Email: yanghui@bupt.edu.cn

Zhengjie Sun
Beijing University of Posts and Telecommunications

Email: sunzhengjie@bupt.edu.cn

Sheng Liu
China Mobile

Email: liushengwl@chinamobile.com

Haomian Zheng
Huawei Technologies

Email: zhenghaomian@huawei.com