

リッチクライアント-エッジサーバ間における 分散機械学習の検討

高野 紗輝† 中尾 彰宏†† 山本 周†† 山口 実靖††† 小口 正人†
 †お茶の水女子大学 ††東京大学 †††工学院大学

1 はじめに

近年 IoT デバイスが普及し、低遅延でネットワークの負荷分散が可能であるエッジコンピューティングに注目が集まっている。従来のエッジコンピューティングではエッジデバイス-エッジサーバ間において生データやそれに少し加工を施した容量の大きなデータの受け渡しが行われている。そのため、プライバシーの保護や通信コストの削減といった観点で問題が生じ、IoT の普及に対応するためにはさらなる工夫が必要である。

そこで本研究では、リッチクライアントの登場により、機械学習等の複雑な処理もエッジデバイス上で行うことが可能になったことと合わせ、リッチクライアントに適した分散機械学習モデルを検討する。

2 関連研究

2.1 エッジ/フォグコンピューティング

現在主流となっているクラウドコンピューティングは遅延が大きく、帯域も多く必要とするため、リアルタイムアプリケーションや大量のデータを送受信するアプリケーションの実装には向いていない。

一方でエッジコンピューティングは、物理的にエッジデバイスに近い位置で処理を行うことにより、低遅延であり、エッジ-クラウド間におけるネットワークの負荷を削減することも可能である [1]。

2.2 Federated learning

近年、高性能な CPU や GPU を搭載したリッチクライアントが登場し、IoT デバイス上でサーバが行っていた機械学習の処理を実行できるようになった。そして、エッジデバイス上で機械学習を行うモデルとして Federated learning という分散型機械学習が提案された [2]。Federated learning はエッジコンピューティングとは異なり、プライバシーに考慮しながらエッジデバイスの情報をクラウドに集約し、クラウドが一括管理するモデルとして実装が進められている [3]。

A Study on Distributed Machine Learning between Rich Client and Edge Server

†Saki Takano ††Akihiro Nakao ††Shu Yamamoto †††Saneyasu Yamaguchi †Masato Oguchi

†Ochanomizu University

††the University of Tokyo

†††Kogakuin University

3 提案モデル

従来のエッジコンピューティングモデルにおいて、リッチクライアントも学習に参加するモデルを図 1 に示すように提案する。

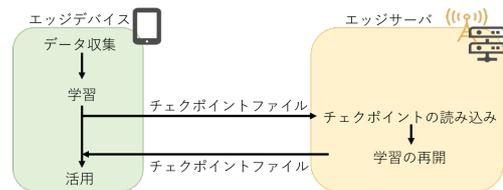


図 1: 提案モデル

従来のエッジコンピューティングではエッジサーバ上で行っていた機械学習等の高度なデータ処理を、エッジデバイスで実行する。その際、必要に応じてエッジサーバへの一部オフロードも可能であり、エッジデバイスで得られた学習結果をエッジサーバと共有することでさらに学習を進め、より良い学習結果をエッジデバイスで利用することができる。

このような構成にすることで、エッジデバイスから外に出せないプライバシーに関わるデータも利用可能になり、エッジサーバとの通信コストも大幅に削減可能となる。

4 実験

4.1 実験環境

使用したエッジサーバの性能を表 1 に、エッジデバイスとして使用した Jetson Nano の性能を表 2 に示す。

表 1: エッジサーバの性能

OS	Ubuntu 18.04 LTS
CPU	Intel Core i7-8700
GPU	GeForce RTX 2080Ti
Memory	32Gbyte

表 2: Jetson Nano の性能

OS	Ubuntu 18.04 LTS
CPU	Quad-core ARM A57 @ 1.43 GHz
GPU	128-core Maxwell
Memory	4 GB 64-bit LPDDR4 25.6 GB/s

Jetson NanoはGPUを搭載した小型AIコンピュータボードであり、近い将来、スマートフォンや様々なIoTデバイスがこのような性能を持つことが期待される。

4.2 予備実験

エッジサーバと Jetson Nano 上で同一の機械学習処理を行った際の実行時間を比較した結果を図2に示す。

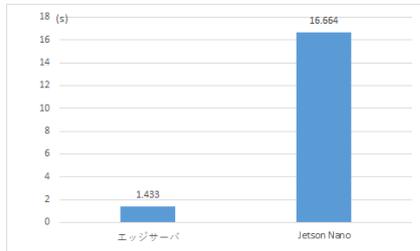


図2: エッジサーバ, Jetson Nano による実行時間

エッジサーバと比較して Jetson Nano はおよそ 11.6 倍の時間を要する。一方で、エッジサーバ, Jetson Nano 共におよそ 96%と同等精度で学習を行うことが可能である。ある程度低速ではあるものの Jetson Nano 内のみでも十分学習できることが分かり、プライバシーが非常に重要なデータもそのような形で学習に用いる事が可能になる。

4.3 実験概要

提案モデルにおいて、MNIST を用い、1 から 100 番目のデータを Jetson Nano に与え、それ以降の 101 から 60000 番目のデータをエッジサーバに与えた。そして、Jetson Nano 上で epoch 数を 2、各 epoch の step 数を 10 で学習を行い、エッジサーバでは、epoch 数を 2、各 epoch の step 数を 1000 で学習を再開させた。

4.4 実験結果

学習精度を図3に示す。

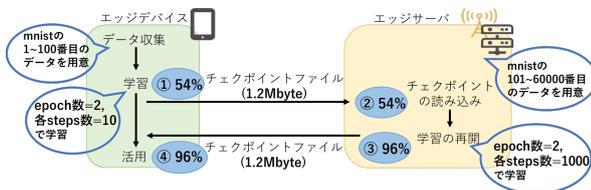


図3: 学習精度

Jetson Nano 上で 54% (①) まで学習できたことが分かり、そこで得られたチェックポイントをエッジサーバへと渡し、エッジサーバ側で計測すると精度は同じく 54% (②) となった。よって、Jetson Nano からエッジサーバへの学習データの受け渡しにより学習精度は落ちないことが読み取れる。ここで得られた結果は十分な精度であるとは言えないが、エッジサーバで学習を

再開させることでエッジサーバ上, Jetson Nano 上共に 96% (③, ④) と高い精度の結果を得ることができた。

また、チェックポイントファイルの容量は共に 1.2Mbyte と学習に使用したデータ数に寄らないため、多くのデータや容量の大きなデータを学習に使用する場合には通信量が削減できるという結果が得られた。

5 提案モデルの検討

エッジデバイス上のみでの学習では精度が不十分な場合にエッジサーバへと学習を引き継ぐことでより精度の高い学習結果が得られ、本提案モデルには以下の利点があると言える。

- 元データの復元が難しい学習の重みのみをエッジデバイスの外へと持ち出すため、プライバシーの点で優れている。
- 必要に応じてエッジサーバへ情報を送るか送らないか判断して選別することができる形を作ることで、機密性の高いデータを扱う場合や取得データを即時的に活用したい場合、エッジデバイスが一時的にネットワークに繋がっていない場合にも対応することが可能となる。
- エッジデバイス-エッジサーバ間はチェックポイントファイルのみを送受信するため、通信コストの削減が可能である。
- 従来エッジサーバが行っていた学習の一部をエッジデバイスにオフロードする形であると言え、エッジサーバの負荷分散が可能である。

6 まとめと今後の課題

従来のエッジコンピューティングで課題となっているプライバシーの保護や通信コストの削減を目的として、リッチクライアントに適した分散機械学習モデルの検討を行った。

今回は MNIST を用いて実験を行なったが、今後はスマートフォン上にあるような画像や SNS のテキスト情報を用いて実験を行い、それらを利用するアプリケーションの構築および検討を行いたいと考えている。

参考文献

- [1] S. Yang. Iot stream processing and analytics in the fog. *IEEE Communications Magazine*, Vol. 55, No. 8, pp. 21-27, 2017.
- [2] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. *Federated machine learning: Concept and applications*, 2019.
- [3] T. Yang, G. Andrew, Hubert Eichner, Haicheng Sun, W. Li, Nicholas Kong, D. Ramage, and F. Beaufays. *Applied federated learning: Improving google keyboard query suggestions*. *ArXiv*, Vol. abs/1812.02903, , 2018.