大規模データ処理フレームワークの ストリーミング機能を利用した機械学習処理の検討

一瀬 絢衣 † 竹房 あつ子 ‡ 中田 秀基 § 小口 正人 †
† お茶の水女子大学 ‡ 国立情報学研究所 § 産業総合技術研究所

1. はじめに

近年各種センサの普及やクラウドコンピューティング技術の習熟に伴い，お年寄りや子供のための安全サービスなどを目的としたライフログの利用が普及してきている．しかし，動画像解析のようなデータ量，計算量の多い処理をクラウドでリアルタイムに行うことは困難である．また，近年ディープラーニング技術の発達で，その高い精度から画像や音声の認識などに広く用いられているが，計算負荷が高いことが問題の一つとなっている．

本研究では，大規模データ処理のための高速かつ汎用性の高いエンジン Apache Spark(以降，Spark と呼ぶ) [1] のストリーミング機能を利用して，ディープラーニングフレームワークの一つである Chainer [2] の機械学習処理を行い，動画像データ解析処理の高速化を図る．本稿ではクライアント側，クラウド側として 2 つの端末間で Apache Kafka [3] を用いてストリーミング処理を行い，クラウド側で機械学習の識別処理を行った．2 つの端末間のネットワーク帯域を変化させ，1 秒あたりに識別処理を行える画像数を計測し，性能を調査した．実験から，低帯域環境においてリアルタイムに識別を行うことが可能であることが示された．

2. 関連技術

2.1 Apache Spark

Spark は，大規模データの格納，処理を目的とした分散処理フレームワークである．Spark は複数のコンポーネントで構成されており，その一つにストリームデータを処理する Spark Streaming がある．Spark Streaming は，数秒から数分ほどの短い間隔で繰り返しバッチ処理を行うマイクロバッチ方式によりストリームデータ処理機能を提供する．

2.2 Chainer

Chainer は，Preferred Networks が開発したディープラーニングフレームワークである．柔軟性，直感的，高速という 3 つの特徴を掲げている．また，「Define-by-Run」方式と呼ばれる，ネットワーク構築と学習を同時に行う方式を採用しているのも大きな特徴の一つである．動的にネットワークを定義するため，手法の自由度が高い．また，インストールも容易にできることから，広く使われている．

A Study of Machine Learning Processing Using a Streaming Function of a Large-scale Data Processing Framework
† Ayae Ichinose, Masato Oguchi
‡ Atsuko Takefusa
§ Hidemoto Nakada
Ochanomizu University (†)
National Institute of Informatcs (‡)
National Institute of Advanced Industrial Science and Technology (AIST) (§)

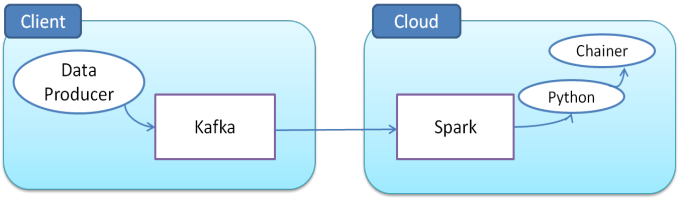

図 1: ストリーミング機械学習

3. ストリーミング機械学習

本研究では図 1 に示す構成でストリーム処理を行う．クライアント側は，分散メッセージングシステムである Apache Kafka を用いてデータの送信を行う．クラウド側では Spark で Kafka からデータを受け取り，Python プログラムを実行して Chainer を呼び出す．ここで，Spark ではデータは RDD (Resilient Distributed Dataset) という，分散処理を前提としたオブジェクトのコレクションとして扱われるため，データは RDD として読み込み，RDD から Chainer の要求する型に変換する．指定されたマイクロバッチサイズに従ってデータを分割し，1 マイクロバッチごとに Chainer で識別処理を行う．

4. スループット計測

本研究では，ストリーミング機能を用いた識別処理の性能を調査するため，クライアント，クラウド間のネットワーク帯域を変化させた環境下で 1 秒あたりに処理できる画像数を調査した．

4.1 実験概要

クラウド側，クライアント側として 2 つの端末を用いてデータのストリーミング処理を行い，クライアント側から流されたデータの識別処理をクラウド側で行う．クラウド側ではマイクロバッチサイズを 1 秒に指定し，1 秒ごとに区切られたデータに対し Chainer を呼び出して識別処理を行う．クラウド側で 2 つの端末間のネットワーク帯域を変化させ，1 秒で処理できる画像の枚数を調査した．Spark Streaming のマイクロバッチサイズを 1 秒に設定しているが厳密には 1 秒とならない可能性があるため，タイムスタンプを用いて時間を計測し，平均値を計算して結果に用いた．実験には 0 から 9 の手書き数字の 28 × 28 画素の画像データに正解ラベルが与えられているデータセットである MNIST [4] を用い，60000 枚のデータを一度に Kafka に流した．実験環境を表 1 に示す．クライアント側及びクラウド側で同質のノードを用い，その間のネットワーク帯域は 1Gbps となっている．ネットワーク帯域の制御には PSPacer [5] を用いた．

表 1: 実験で用いた計算機の性能

OS	Ubuntu 16.04LTS
CPU	Intel(R) Xeon(R) CPU W5590 @3.33GHz (8 コア) × 2 ソケット
Memory	8Gbyte

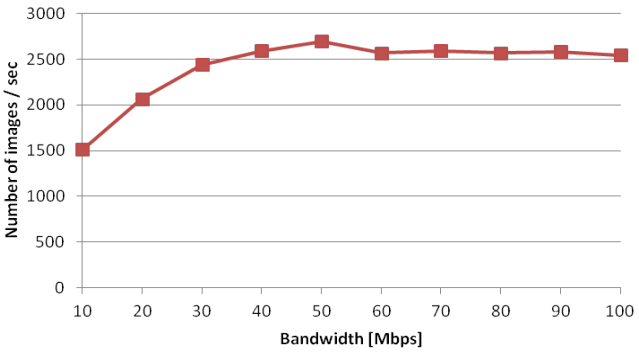

図 2: 1 秒あたりの処理画像数

4.2 実験結果

実験結果を図 2 に示す．横軸が端末間のネットワーク帯域を示し，縦軸が 1 秒あたりに処理できた画像の枚数を示す．ネットワーク帯域が 50Mbps の場合に処理できる画像の枚数が最大であり，60Mbps 以上の場合には減少していることが確認できる．また，タイムスタンプから計測した 1 マイクロバッチサイズを図 3 に示す．20Mbps 以上の場合にマイクロバッチサイズの 1 秒が守られていないことが確認できた．これは，データの到着量が多くなるとクラウド側で Chainer ををを用いた識別処理が間に合わなくなるためだと考えられる．つまり，20Mbps 以上の場合には Spark にデータが溜まってしまい，連続的にデータを流し続けるとドロップされてしまう．しかし，1 秒当たりに 2500 枚の画像の識別が可能であることが確認できた．

5. まとめと今後の課題

本研究では大規模データ処理フレームワーク Spark のストリーミング機能を利用して Chainer を用いた機械学習処理を行う事により，リアルタイムセンサデータ解析処理における性能要件を検討した．実験から，低帯域環境でリアルタイムでの画像識別が可能であることが確認できた．高帯域環境ではクラウド側の処理がボトルネックとなってしまい，マイクロバッチのインターバルが守れなくなってしまうことが確認できた．

今後の課題としては，クラスタを用いた分散実行によりクラウド側の高速化を図り，データの到着量が多い場合におけるリアルタイムを可能にする．また，機械学習の識別処理に加え，学習処理や動画像解析処理を行う．

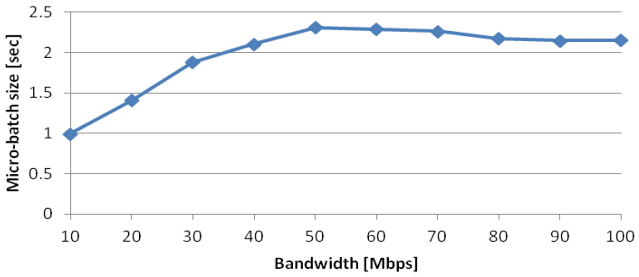

図 3: マイクロバッチサイズ

謝辞

この成果の一部は，JSPS 科研費 JP16K00177 および国立研究開発法人新エネルギー・産業技術総合開発機構（ＮＥＤＯ）の委託業務の結果得られたものです．

参考文献

[1] Apache Spark, https://spark.apache.org/.

[2] Tokui, S., Oono, K., Hido, S. and Clayton, J.: Chainer: a Next-Generation Open Source Framework for Deep Learning, *In Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)* (2015). 6 pages.

[3] Apache Kafka, https://kafka.apache.org/.

[4] Lecun, Y., Cortes, C. and Burges, C. J. The MNIST Database of handwritten digits, http://yann.lecun.com/exdb/mnist/.

[5] Takano, R., Kudoh, T., Kodama, Y. and Okazaki, F.: High-resolution Timer-based Packet Pacing Mechanism on Linux Operating System, *IEICE Transactions on Communications*, Vol. E94.B, No. 8, pp. 2199–2207 (2011).