

Apache Spark によるディープラーニング の並列分散処理

安藤 祐斗

富山県立大学 情報基盤工学講座
t815008@st.pu-toyama.ac.jp

June 4, 2021

はじめに

並列分散処理

ディープラーニング

使用するライブラリ

サンプルプログラムの実行

機械学習による超解像

超解像プログラムの並列化

まとめ

背景

機械学習の手法の一つであるディープラーニングは、近年の進歩により、画像認識などにおける認識精度の向上、自動運転、医療研究などの幅広い分野での活用がされている。

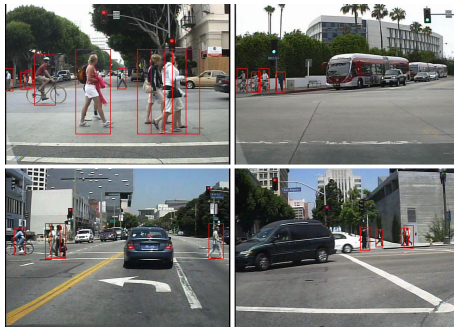


図 1: ディープラーニングの例（歩行者検知）

はじめに

並列分散処理

ディープラーニング

使用するライブラリ

サンプルプログラムの実行

機械学習による超解像

超解像プログラムの並列化

まとめ

目的

本研究では,Apache Spark の並列分散処理機能を使いディープラーニングを実行する.

次に, この二つの組み合わせによって得られる優位性や, 既存のプログラムにはない新規性を確認する.

はじめに

並列分散処理

ディープラーニング

使用するライブラリ

サンプルプログラムの実行

機械学習による超解像

超解像プログラムの並列化

まとめ

Apache Spark とは

大量のデータを複数のコンピュータで処理を行う、並列分散処理を可能としたソフトウェア。

複数のサーバーでデータを格納するファイルシステムである HDFS (Hadoop Distributed File System) と、格納されたデータを繰り返し加工し処理する RDD という分散データセットによって構成されている。



処理モデル



図 2: Spark の構成

ニューラルネットワークとディープラーニング

ニューラルネットワークとは、神経細胞（ニューロン）と神経回路網（シナプス）で構成された、人間の脳神経を模倣した数理モデルである。ニューラルネットワークは入力層、中間層、出力層の3つの層に分けられ、この中のさまざまな計算を行う中間層が、3層以上のニューラルネットワークを用いた手法をディープラーニングと呼ぶ。中間層を多く用いることによってより複雑な分析ができ、データの特徴を抽出することができる。

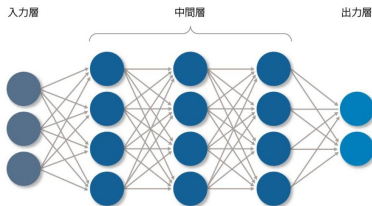


図 3: ニューラルネットワーク

はじめに

並列分散処理

ディープラーニング

使用するライブラリ

サンプルプログラムの実行

機械学習による超解像

超解像プログラムの並列化

まとめ

BIGDL とは

Spark によるディープラーニングの分散処理を容易にするライブラリである.

現在,BIGDL の公式サイトに則り, 使い方を勉強中です.

はじめに

並列分散処理

ディープラーニ
ング

使用するライブ
ラリ

サンプルプログラ
ムの実行

機械学習による超
解像

超解像プログラ
ムの並列化

まとめ

サンプルプログラムの概要

7/14

はじめに

並列分散処理

ディープラーニング

使用するライブラリ

サンプルプログラムの実行

機械学習による超解像

超解像プログラムの並列化

まとめ

最初に, 画像からパターンや物体の認識に最も利用されている, 畳み込みニューラルネットワークの一つである LeNet5 をベースに構築し, MNIST と呼ばれる手書き画像のデータセットを用いて学習をさせる. 次に, 学習で作成したモデルのテストを行い, 正確性を確認する. Spark を使い, これらを分散処理させる.

4 台の PC でサンプルプログラムの実行を行った。

合計のコア数は 8, メモリ数は 9.4GB

スレイブにそれぞれ 2 コア, メモリを 2GB ずつ与えている。

```
2021-05-10 17:15:04 INFO DistriOptimizer$:180 - [Epoch 15 60000/60000][Iteration
28125][Wall Clock 1898.373815905s] Loss is (Loss: 442.4443, count: 10000, Averag
e Loss: 0.04424443)
```

図 4: かかった時間 (4 台)

```
Top1Accuracy is Accuracy(correct: 9857, count: 10000, accuracy: 0.9857)
```

図 5: テスト結果 (4 台)

下は PC2 台で実行したときで, スレイブそれぞれ 2 コア, メモリを 2GB ずつ与えている。

```
2021-05-07 03:18:53 INFO DistriOptimizer$:180 - [Epoch 15 60000/60000][Iteratio
n 112500][Wall Clock 13824.818030905s] Loss is (Loss: 408.97772, count: 10000, A
verage Loss: 0.04089777)
```

図 6: かかった時間 (2 台)

```
Top1Accuracy is Accuracy(correct: 9871, count: 10000, accuracy: 0.9871)
```

図 7: かかった時間 (2 台)

超解像とは

- ・ 静止画や動画を入力とし、高解像度化して出力する技術
- ・ ディープラーニングを応用した SRCNN などの手法がある。



pic01_L.png



pic02_L.png



pic03_L.png



pic01_S.png



pic02_S.png



pic03_S.png

図 8: 高解像度化の例

はじめに

並列分散処理

ディープラーニング

使用するライブラリ

サンプルプログラムの実行

機械学習による超解像

超解像プログラムの並列化

まとめ

超解像の実装 (SRCNN)

10/14

実際に Web サイトに載っていた超解像のプログラムを PC1 台で実行してみました。

超解像手法

SRCNN (Super-Resolution Convolutional Neural Network) :3 層の畳み込みニューラルネットワーク (CNN) を用いて超解像を行う。

評価指標

PSNR(Peak signal-to-noise ratio):ピーク信号対雑音比と言われ、画像の劣化を表す評価指標。標準的な値は 30~50dB で、高い方が画質が良い。

データセット

General-100:動物、植物、建造物などの 100 個の画像からなるデータセット

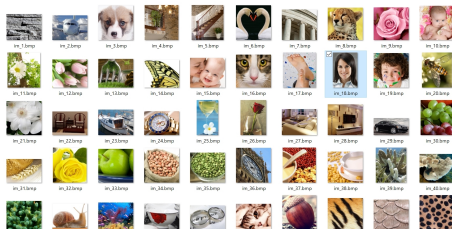


図 9: データセット (General-100)

はじめに

並列分散処理

ディープラーニング

使用するライブラリ

サンプルプログラムの実行

機械学習による超解像

超解像プログラムの並列化

まとめ

SRCNN の手順

- データセットの高解像度の画像をぼやかして、低解像度化した画像を生成する.
- 低解像度画像を CNN に適用する
- 低解像度画像を学習データ, 高解像度画像を正解データとして学習する.

学習パラメータ

- エポック数:3000
- バッチサイズ:16

はじめに

並列分散処理

ディープラーニング

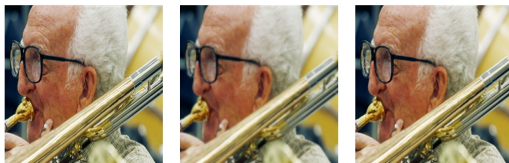
使用するライブラリ

サンプルプログラムの実行

機械学習による超解像

超解像プログラムの並列化

まとめ



オリジナル画像 → 低解像度 → 超解像

図 10: 実行結果

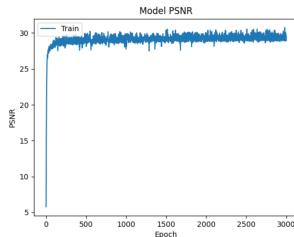


図 11: PSNR

はじめに

並列分散処理

ディープラーニング

使用するライブラリ

サンプルプログラムの実行

機械学習による超解像

超解像プログラムの並列化

まとめ

やってみたいこと

機械学習による超解像プログラムの並列化を行う。
それによって学習時間の行う短縮と精度 (PSNR) の向上を図る。

はじめに

並列分散処理

ディープラーニング

使用するライブラリ

サンプルプログラムの実行

機械学習による超解像

超解像プログラムの並列化

まとめ

進捗

- BIGDL のサンプルプログラムを 4 台で実行した.
- 超解像プログラムを 1 台で実行した.

今後の課題

- BIGDL ライブラリを使った Python の記述方法を学ぶ.
- 超解像プログラムの並列化をする.
- できるだけ最新の超解像技術を色々調べる.

はじめに

並列分散処理

ディープラーニング

使用するライブラリ

サンプルプログラムの実行

機械学習による超解像

超解像プログラムの並列化

まとめ