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

May 7, 2021

本研究の背景

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, 画像認識などにおいての認識精度の向上, 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.

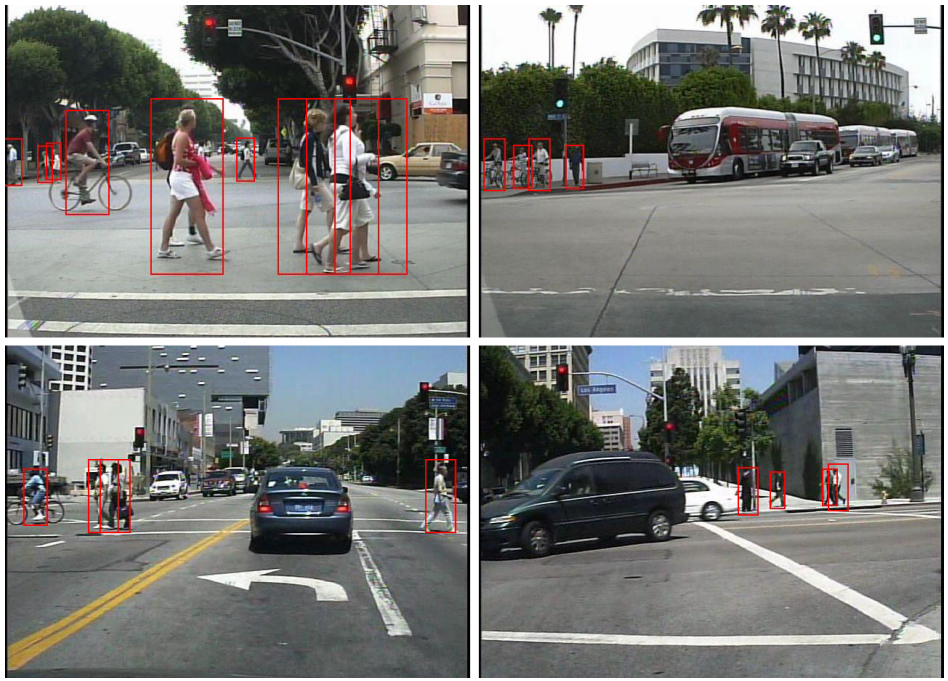

図 1: ディープラーニングの例（歩行者検知）

研究の目的

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.

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.

図 2: Spark の構成

ディープラーニング

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.

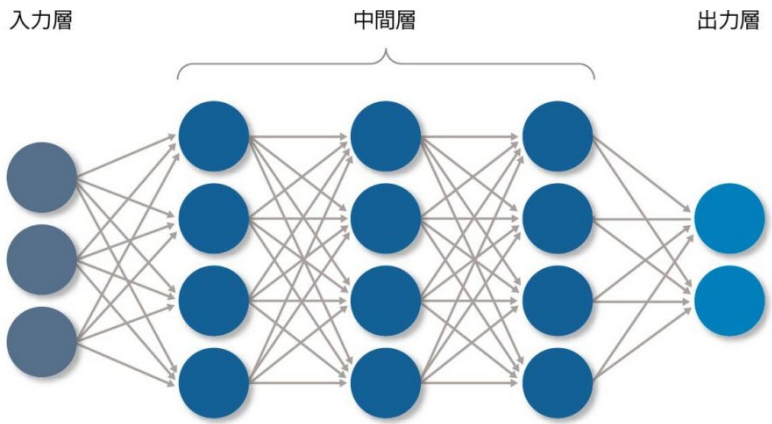

図 3: ニューラルネットワーク

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.

サンプルプログラムの概要

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.

結果

マスター 1 台、スレイブ 2 台でサンプルプログラムの実行を行った。
合計のコア数は 4、メモリは 5.4GB。スレイブにそれぞれ 2 コア、メモリを 2GB ずつ与えている。

```
2021-05-07 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 4: かかった時間（2 台）

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

図 5: テスト結果（2 台）

下はスレイブ一台でコア数が 1，メモリは 4 GB のとき

```
2021-04-05 21:41:38 INFO  DistriOptimizer$:180 - [Epoch 15 60000/60000][Iteration 900000]
[Wall Clock 18621.790747666s] Loss is (Loss: 4808.387, count: 10000, Average Loss: 0.4808
3872)
```

図 6: かかった時間（1 台）

```
Top1Accuracy is Accuracy(correct: 9054, count: 10000, accuracy: 0.9054)
```

図 7: かかった時間（1 台）

GUI アプリ化

BIGDL を動かして思ったこと

・コンソールでの操作は初心者には分かりづらく親しみが持てない
・GUI アプリにしたら視覚的に見やすく分かりやすいのではないか？
・調べた限りでは、ディープラーニングの分散処理に関する GUI アプリはまだない

```
[tpu@slave8 ~]$ $SPARK_HOME/bin/spark-submit --master spark://slave8:7077 \--exe
cutor-cores 1 \--total-executor-cores 2 \--driver-class-path $BIGDL_HOME/lib/big
dl-SPARK_3.0-0.12.1-jar-with-dependencies.jar \--class com.intel.analytics.bigdl
.models.lenet.Train \./dist-spark-3.0.0-scala-2.12.10-all-0.12.1-dist/lib/bigdl-
SPARK_3.0-0.12.1-jar-with-dependencies.jar \-f ./mnist \-b 4 \--checkpoint ./mod
el
```

図 8: コンソール画面

GUI アプリに実装すること

・ディープラーニングのモデルの種類とデータセットの設定
・マスターとスレイブの起動
・spark と学習に必要な各パラメータの設定
・学習とテスト

GUI アプリ化イメージ図

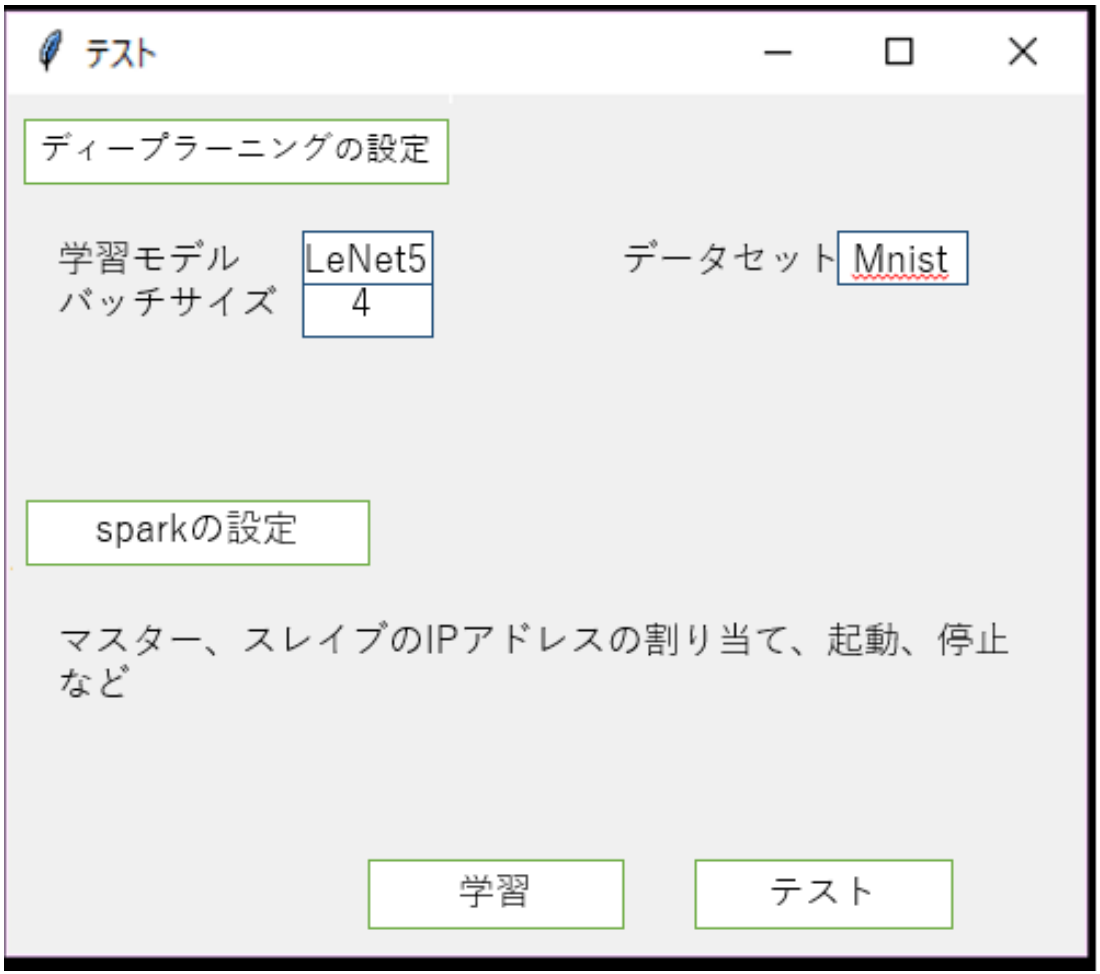

図 9: イメージ図

まとめ

進捗

- BIGDL と Spark の環境構築を 4 台の PC で行った.
- BIGDL の例を 2 台以上で実行したときに発生するエラーの解決
- テーマの提案

今後の課題

- コンソールと GUI をどう連携させるかといったシステム面を考える.
- GUI アプリの作成ツールを選ぶ