

Keras + TensorFlow 実習

2019/11/7 岩崎 昌子

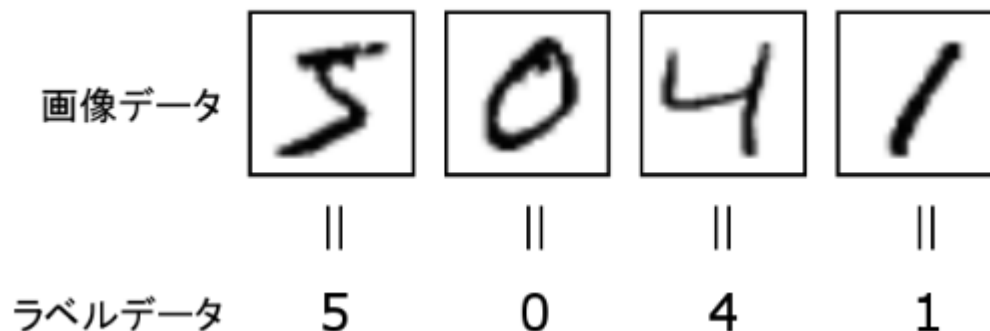
- この実習では、Keras+TensorFlowの
簡単なサンプルプログラムを動かしてみます
- 手書き数字の認識プログラムです
- <https://qiita.com/ash8h/items/29e24fc617b832fba136> と
<https://qiita.com/fukuit/items/b3fa460577a0ea139c88>
のサンプルファイルをもとに、改良したプログラムを使用します
- MNISTデータ(手書き文字)と、
手書き数字データを加工したものを使用します
- プログラムの作成とデータの加工は
大阪市立大学 久野 彰浩さん(4年生)が作成しました

プログラムと、加工データは一般公開用ではないので、
プログラムと加工データは再配布しないでください

Do not distribute the sample programs and sample data, we use today

MNIST 手書き数字データとは？

手書きで書かれた数字を画像にした画像データ(image)と、その画像に書かれた数字を表すラベルデータ(label)から構成される。

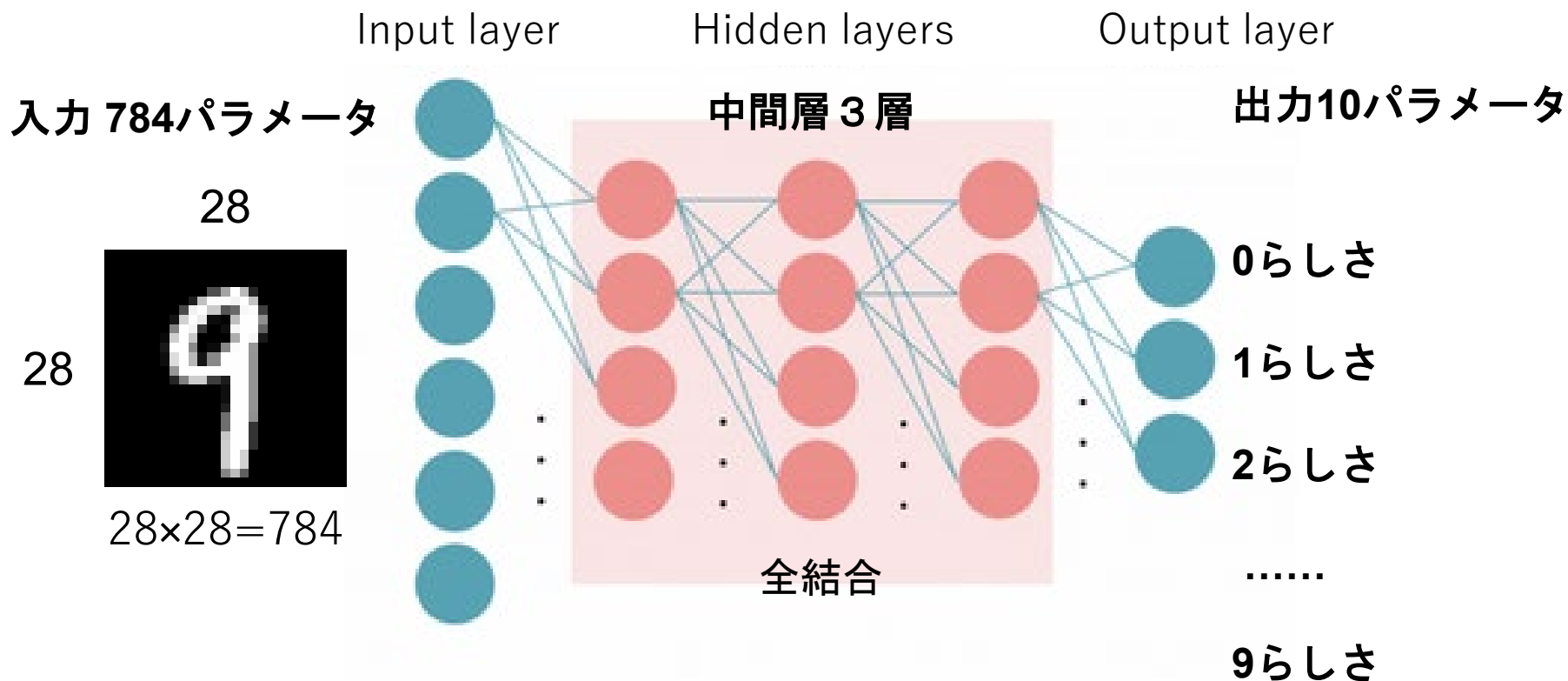


以下の4つのファイルで構成される

- train-images-idx3-ubyte: 学習用の画像セット (60,000個)
- train-labels-idx1-ubyte: 学習用のラベルセット
- t10k-images-idx3-ubyte: 検証用の画像セット (10,000個)
- t10k-labels-idx1-ubyte: 検証用のラベルセット

画像データ: ピクセルサイズ $28 \times 28 = 784$ → 784個の入力データ

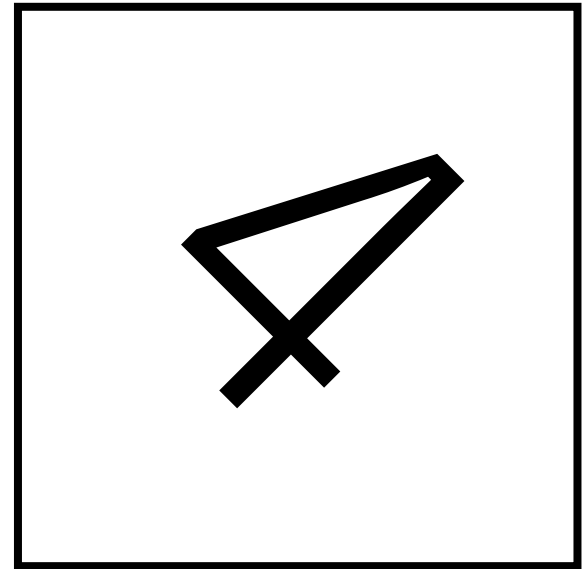
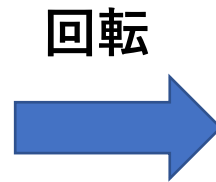
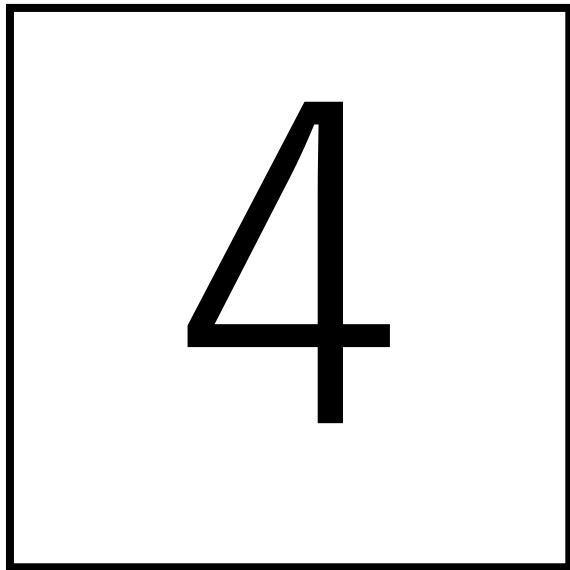
サンプルプログラムで用いるネットワーク



最適化アルゴリズム Optimization : RMSProp
活性化関数 Activation function : ReLU
損失関数 Loss function : Categorical Crossentropy

サンプルで使用する手書き数字データ

MNIST 手書き数字画像データを回転させたものも用意しました



サンプルプログラム

- mlp_mnist_1.py
program from
https://github.com/fchollet/keras/blob/master/examples/mnist_mlp.py
- mlp_mnist_2.py
上のプログラムに正答率(acc),損失関数の値(loss)のグラフを加えたもの
- mlp_mnist_3.py 回転なしデータで学習、回転ありデータで評価
- mlp_mnist_4.py 回転ありデータで学習、回転ありデータで評価
- mlp_mnist_5.py 回転角を示す値を入力に加えたもの
- cnn_mnist.py CNNを用いたプログラム

サンプルプログラムは、以下にあります。

http://www.rcnp.osaka-u.ac.jp/~masako/KEK_DNN_20191107/source.tar.gz

回転ありデータは

http://www.rcnp.osaka-u.ac.jp/~masako/KEK_DNN_20191107/mnistdata.h5
からダウンロードできます。

プログラムとデータは後日削除する可能性があります
プログラムとデータは、再配布しないでください

プログラムの実行

プログラムは `anaconda3` の環境で実行可能です。

KEKCCを使用する場合(ILC)

最初に、環境変数を設定する `/group/ilc/soft/env.sh` を実行
`$ python mlp_mnist_1.py` と入力して実行

KEKCC以外の場合

環境によっては、以下を変更する必要があります

`import keras` または
`import tensorflow.keras` 、どちらか適切な法を選ぶ必要があります

例：`mlp_mnst_1.py`

11-20行目か22-31行目、どちらか適切な方を選んでください。

例：`mlp_mnst_2.py`

11-20行目か22-31行目、85-86行目 または88-89行目、
どちらか適切な方を選んでください。

大阪市大のCPUで Keras + TensorFlowを実行

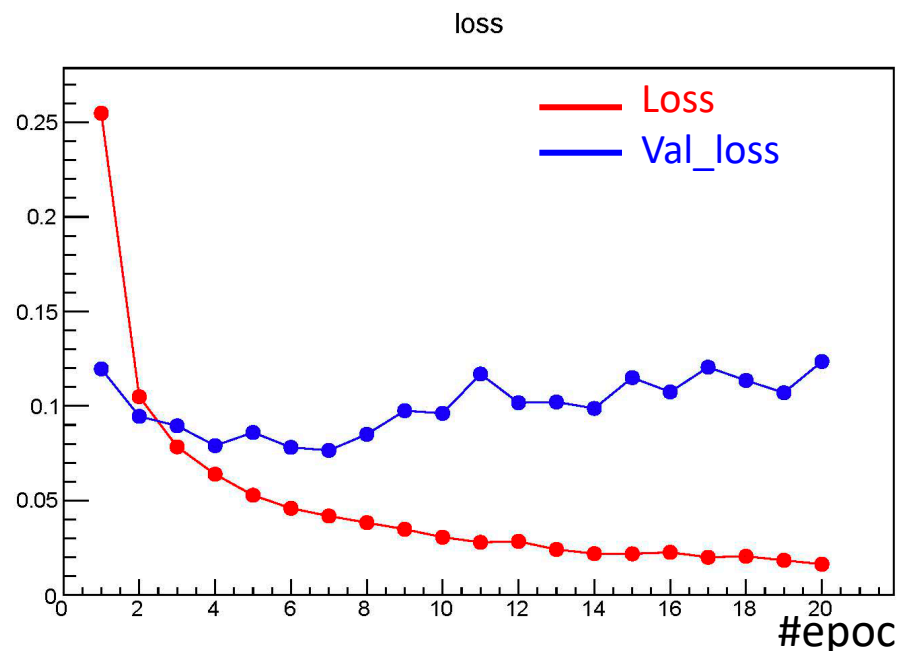
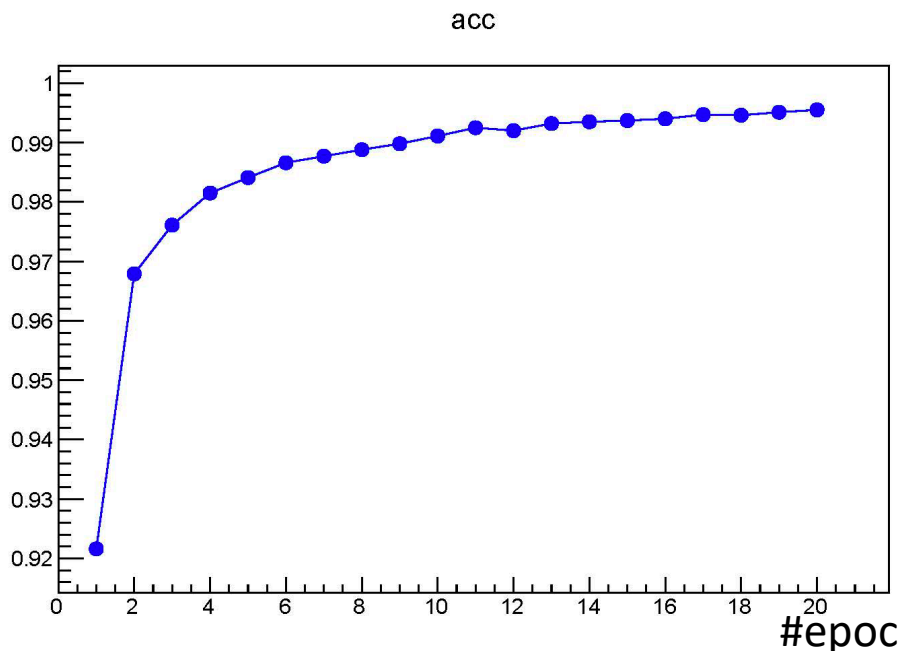
<https://qiita.com/ash8h/items/29e24fc617b832fba136> のサンプルファイル

(手書き数字認識)を実行してみた

Linux CentOS6.9 Intel Core 2 Duo E4500 2.2GHz 3.2GB メモリ

epoc 20回 計算時間は約10分

正答率とlossのグラフは以下の通り



ダウンロードされたMNISTファイルは home directory の下に .keras/datasets/mnist.npz として、コピーされる

RCNPのCPU(miho)で Keras + TensorFlowを実行

Mihoで実行してみた

Linux RH6.5 Intel Xenon CPU E5-2667 3.3GHz 32GB メモリ
(市大 Linux CentOS6.9 Intel Core 2 Duo E4500 2.2GHz 3.2GB メモリ)

市大と同様に <https://qiita.com/ash8h/items/29e24fc617b832fba136> の
サンプルファイル(手書き数字認識)を実行

epoch 20回 計算時間は約22分 ← 遅い?!
正答率とロス市大で計算した結果が再現された
→ バッチシステムでないと、計算が大変です。

参考: ノートPC VMware

Linux CentOS 6.9 Intel Core i7-3667U 2.00GHz, 1GBメモリ
同じ計算を行ったところ、正しい正答率が得られなかった。。。 (acc ~0.65, loss ~5.6)
スペックが足りないと、ちゃんと計算しない? できない? ← 要注意?