

## 課題4レポート

情報学科 計算機コース 1029275871 芦田聖太

提出日 18/1/26

## 課題 4

### 手書き数字画像識別器の構築

#### 仕様

- パラメータの読み込み
- 入力したインデックスのデータを評価
- テスト画像全体の正答率を表示

## 1 設計方針

以下の 5 つの項目を実装することで仕様を満たすことにした。

- 画像読み込み
- 評価する画像のインデックス入力
- パラメータ読み込み
- テストデータ全体の評価
- 入力したインデックスの評価結果と解答を表示
- テストデータの正答率の計算と表示

## 2 実装とプログラムの説明

### 2.1 画像読み込み

この部分の実装は課題 1 とほとんど変わっていない。変数の名前をわかりやすくするために、learn と answer に付け替えた。learn は (784, 10000) の行列で各列に 10000 個の画像データがそれぞれ含まれている。

```
# テストデータ
mndata = MNIST("/Users/omushota/ex4-image/le4nn")
X, Y = mndata.load_testing()
X = np.array(X)
X = X.reshape((X.shape[0], 28, 28))
Y = np.array(Y)

# 定数設定
line = X.shape[0]
row = X.shape[1]

# テスト
learn = np.reshape(np.array(X), (line, row * row)).T
answer = Y
```

## 2.2 評価する画像のインデックス入力

この部分は課題1と全く同じなので説明は省略する。

```
# 入力受け取り
while True:
    strnum = input("input_number:_")
    print("")
    num = int(strnum)
    if (num < 0) or (num > 9999):
        print("Please_type0~9999")
    else:
        break
```

## 2.3 パラメータ読み込み

np.loadで重みを保存したファイルをweightfileに読み込む。w1, b1, w2, b2という名前で各パラメータを保存していたので、weightfile['w1']などで重みをそれぞれ読み込む。

```
# パラメータの読み込み
weightfile = np.load('parameters.npz')

# 重み1
weight1 = weightfile['w1']
b1 = weightfile['b1']

# 重み2
weight2 = weightfile['w2']
b2 = weightfile['b2']
```

## 2.4 テストデータ全体の評価

この部分も以前の課題で説明している部分なので説明は省略する。

```
# 中間層#####
# 定数
middle = 300

# 中間層への入力
midinput = weight1.dot(learn) + b1

# シグモイド
midout = funcs.sigmoid(midinput)

# 出力層#####
# 定数
end = 10

# 出力層への入力
fininput = weight2.dot(midout) + b2

# ソフトマックス
finout = funcs.softmax(fininput)
```

## 2.5 入力したインデックスの評価結果と解答を表示

indexmaxにテストデータ全体の評価結果を格納し、その中から入力したインデックスの答えを取り出す。ansも同様にして、解答を取り出している。

```

indexmax = finout.argmax(axis=0)
Tindexmax = indexmax[num]
ans = Y[num]

print("結果")
print(Tindexmax)
print("答え")
print(ans)
print("")

```

## 2.6 テストデータの正答率の計算と表示

power には 10000 個のデータの評価結果列と解答列を各要素で引き算をしたものを格納する。power のなかで、0 になっている部分の数を数えることで正解数を数える。最後に正答率を計算して表示している。

```

power = indexmax - Y
counter = len(np.where(power == 0)[0])

print("10000 枚の正答率")
print((counter / 10000.0) * 100.0)

```

## 3 実行結果

以下のような実行結果が得られる。入力画像の評価結果と答えを表示し、さらに 97.2% の正答率が得られた。

```

input number : 10

# 結果
0

# 答え
0

10000 枚の正答率
97.2

```

## 4 考察

### 4.1 工夫点

学習がうまく進んでいるかを知るために正答率を測定する機能をつけた。その過程で、同時に入力した index の画像の評価結果も表示することができた。ただ、10000 枚のデータを扱うので処理速度が心配された。そこで、10000 個のデータを一つの行列にすることで計算を早くするようにした。評価結果の返ってくるスピードが for 文を使うと非常に遅くなってしまっていたため、(784, 10000) の行列で全てのテストデータを表すことで、効率化を測った。

### 4.2 問題点

10000 枚のデータを行列にして効率化をはかったが、まだ少し時間が長く感じられる。ニューラルネットの部分で関数を多く使っているからかもしれない。そのあたりをもう少し効率よくできれば、評価が早くなると思われる。