

# 計算機科学実験及演習4「画像認識」 課題1

工学部 情報学科 計算機科学コース 3回生  
山田瑛平 学籍番号:1029282731

平成30年10月25日

## 課題内容

以下のようなプログラムを作成する.

- キーボードから整数値  $i$  ( $0 \leq i \leq 9999$ ) を入力として受け取り, 0 から 9 までの値を出力する.
- MNIST のテストデータにおける  $i$  番目の画像を入力画像にとる.
- 重みが正規分布に従う乱数に設定されている 3 層ニューラルネットにより, 入力画像を 0 から 9 までのいずれかの値に分類する.

## プログラムの説明

### 定数

当プログラムでは定数を以下のように定義している.

```
SIZEX = 28
SIZEY = 28
PIC = 10000
MID1 = 5
MID2 = 10
```

以下に各変数の説明を記す.

- SIZEX および SIZEY は画像の横軸および縦軸のサイズである.
- PIC はデータセットの画像の枚数である.
- MID1 および MID2 は中間層および出力層のニューロンの数である.

### $\text{sig}(t)$

$\text{sig}(t)$  は以下のように定義される.

```
def sig(t):
    return 1/(1+np.exp((-1)*int(t)))
```

$\text{sig}(t)$  は整数値を入力に取るシグモイド関数

$$\sigma(t) = \frac{1}{1 + \exp(-t)} \quad (1)$$

である.

### $\text{sigArr}(t)$

$\text{sigArr}(t)$  は以下のように定義される.

```
def sigArr(a):
    return (np.apply_along_axis(sig, 1, np.array(a)))
```

$\text{sigArr}(t)$  は  $m \times 1$  行列を入力に取り, 各要素にシグモイド関数を適用した同一サイズのベクトルを返す関数である.

## sof(a)

$sof(a)$  は以下のように定義される.

```
def sof(a):
    m = np.amax(np.array(a))
    row = (np.array(a)).shape[0]
    all_m = np.full((row,1),m)
    comArr1 = np.array(a) - all_m
    comArr2 = (np.apply_along_axis(np.exp, 1, comArr1))
    sum_arr = np.sum(comArr2)
    return comArr2 / sum_arr
```

$sig(t)$  は  $C \times 1$  行列  $a$  を入力に取り, 各要素  $y_i$  が

$$y_i = \frac{\exp(a_i)}{\sum_{j=1}^C a_j} \quad (2)$$

であるベクトルを返す.

## 入力層

入力層は以下のように実装されている.

```
input = X[int(i)].reshape(SIZE_X * SIZE_Y, 1)
```

入力層では  $28*28$  行列を  $784*1$  行列に変換している.

## 中間層

中間層は以下のように実装されている.

```
np.random.seed(0)
w1 = np.random.normal(0.0, np.sqrt(1.0/(SIZE_X * SIZE_Y)), (MID1, SIZE_X*SIZE_Y))
b1 = np.random.normal(0.0, np.sqrt(1.0/(SIZE_X * SIZE_Y)), (MID1, 1))

com11 = w1 @ input + b1
com12 = sigArr(com11)
y1 = np.c_[com12]
```

中間層では, 正規分布に基づいて重み行列  $W_1$  および  $b_1$  を定め, 出力層への入力

$$y_1 = \sigma(W_1 x + b_1) \quad (3)$$

を出力する.

## 出力層

出力層は以下のように実装されている.

```
np.random.seed(1)
w2 = np.random.normal(0.0, np.sqrt(1.0/(MID1)), (MID2, MID1))
b2 = np.random.normal(0.0, np.sqrt(1.0/(MID1)), (MID2, 1))

com21 = w2 @ y1 + b2
y2 = np.c_[com21]
output = np.argmax(sof(y2))
```

出力層では, 正規分布に基づいて重み行列  $W_2$  および  $b_2$  を定め, 中間層から受け取った値を入力として

$$output = \operatorname{argmax} \operatorname{sof}(W_2 y_1 + b_2) \quad (4)$$

を出力する.

## 実行結果

0 から 9999 までの間の値を入力すると, 0 から 9 までのいずれかの数値を出力する. それ以外の数値を入力すると, エラーメッセージと共にプログラムを終了する.

## 工夫点および問題点

当プログラムの実装にあたっては, ソースコードの簡潔さを重視するとともに, 今後の実装において流用しやすいようなプログラムを目指した. 定数の定義などにおいて工夫を施したが, シグモイド関数をプログラム自体から分離するなどの改善が挙げられるなど, 実装の流用という観点から見ればまだ不十分であると言える.

また, 標準入力に整数値以外の値を入力したときの対処をしていない. これは, 課題 1 は 3 層ニューラルネットの枠組みを作る段階であり, 細かいコーナーケースに気を配る必要は無いと考えての判断である. 入力エラーに関しては, 画像を一定水準以上の精度で識別できるようになれば実装する予定である.