

# 計算機科学実験4 画像処理

1029311501 坂井優斗

提出日：2021年12月3日

## 1 課題内容

MNIST の画像 1 枚を入力として、3 層のニューラルネットワークを用いて、0 ~ 9 から値を 1 つ出力するプログラムを作成する。プログラムの仕様は以下の通りである。

- 入力として、0 ~ 9999 の整数  $i$  を 1 つ受け取り、ニューラルネットワークに通したのち 0 ~ 9 の整数 1 つを出力する。
- 使用する MNIST の画像は 10000 枚のテストデータのうち、 $i$  番目を使用する。
- 中間層のノードの数  $M$  は自由  $\rightarrow$  100 とした。
- 重み  $W^{(1)}, W^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}$  は、手前の層のノードの数  $N$  の逆数  $1/N$  を分散とする正規分布によって定められる乱数で決定する。

## 2 プログラムの説明

関数ごとにソースコードに対する説明をまとめる。

### 2.1 main 関数

Code 1: main 関数

```
1 def main():
2     test_images = mnist.download_and_parse_mnist_file("t10k-images-idx3-ubyte.gz")
3     test_labels = mnist.download_and_parse_mnist_file("t10k-labels-idx1-ubyte.gz")
4
5     i = int(input())
6     image = test_images[i]
7     processed_image = output_layer(inner_layer(input_layer(image)))
8     prediction = np.argmax(processed_image)
9     print(prediction)
```

まず、test\_images, test\_labels という名前の変数に MNIST のテスト画像データと正解ラベルを読み込み、保存する。次に、input() 関数を用いて、標準入力で整数を受け取り、それを  $i$  とする。テスト画像が格納されている配列 test\_images から  $i$  番目の画像をとってきて、その画像を 3 層からなるニューラルネットワークに入力する。入力層、中間層、出力層はそれぞれ input\_layer, inner\_layer, output\_layer 関数で定義されているので、これらの関数に入力することでニューラルネットワークによる処理が行える。処理を行なったデータを processed\_image として、この配列の要素は各ラベルである尤度を示しているのので、numpy の argmax 関数で尤度が最大であるインデックスを取得し、その値を標準出力に出力している。

### 2.2 sigmoid\_function

Code 2: sigmoid\_function

```
1 def sigmoid_function(t):
2     return 1/(1 + np.exp(-t))
```

この関数は, numpy の exp 関数 を使って,

$$f(\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{x})}$$

で表されるシグモイド関数を実装している。

## 2.3 softmax\_funciton

## 3 実行結果

## 4 工夫点

- numpy が提供している行列演算の関数を利用することで, for 文を回すことなく行列の内積等の計算を行った。これによって計算時間の短縮が可能となった。

## 5 問題点