

機械学習プログラムを動かして 実際に「学習」させてみよう

信州大学
岡野浩三

自己紹介

- 岡野浩三
- 大阪生まれ神戸育ち
- 専門はソフトウェア工学 理論より
- 最近は機械学習**X**ソフトウェア工学

<https://www.shinshu-u.ac.jp/faculty/engineering/MDASH/dsaier/>

TA紹介

- みなさんのPCを用意していただきました
- 今日みなさんのお手伝いをします
- こまったことがあればどしどし質問してやってください

今日の予定

- 午前1 数字認識のAIを作って学習、使ってみよう
 - 午前2 AI（機械学習）の基礎
 - 午後1 数字生成のAIを作って学習、使ってみよう
 - 午後2 数独プログラムを改良してみよう
-
- アンケート記入

参考文献

- 斎藤 康毅 (著) ゼロから作る**Deep Learning** —Pythonで学ぶ
ディープラーニングの理論と実装
- 斎藤 康毅 (著) ゼロから作るDeep Learning ⑤ 一生成モデル編
- <https://zenn.dev/lilpacy/articles/cc4dee448568bf>
- <https://qiita.com/gensal/items/613d04b5ff50b6413aa0>
- <https://qiita.com/gensal/items/4be65d9b36fac02f877e>

数字認識のAIを作って学習、使ってみよう

- まずは自分たちで手書きの数字画像を作ろう
- Gimpを使います
- Gimp フリーソフトウェア：一部の機能はAcrobat Illustratorなみ
- 自分で手書き数字を 0 から 9 まで好きなだけ作ってください

画像フィルタ

- 画像フィルタを画像の画素ごとずらしながら適用していく

40	40	60	60	60
100	100	110	120	130
100	110	120	130	130
120	140	150	160	160
140	140	160	160	160

入力画像

— 注目画素
— フィルタがかかる領域

-1	-2	-1
0	0	0
1	2	1

縦方向ソベルフィルタ

		160		

出力画像

積和演算

$$\begin{aligned}
 &100 \times (-1) + 110 \times (-2) + 120 \times (-1) \\
 &+ 100 \times 0 + 120 \times 0 + 130 \times 0 \\
 &+ 140 \times 1 + 150 \times 2 + 160 \times 1 \\
 &= 160
 \end{aligned}$$

画像フィルタ

加重平均化フィルタ [3×3]
(自分自身の画素に近いほど重み大なフィルタ)

$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$
$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$



画像フィルタ

鮮鋭化

-1	-1	-1
-1	9	-1
-1	-1	-1



少しセッティングをします

- 機械学習は実行するのにかなり時間がかかります
- 高性能サーバーで学習を実行しようと思います
- 高性能サーバーにアクセスするためのおまじないを少し唱えます

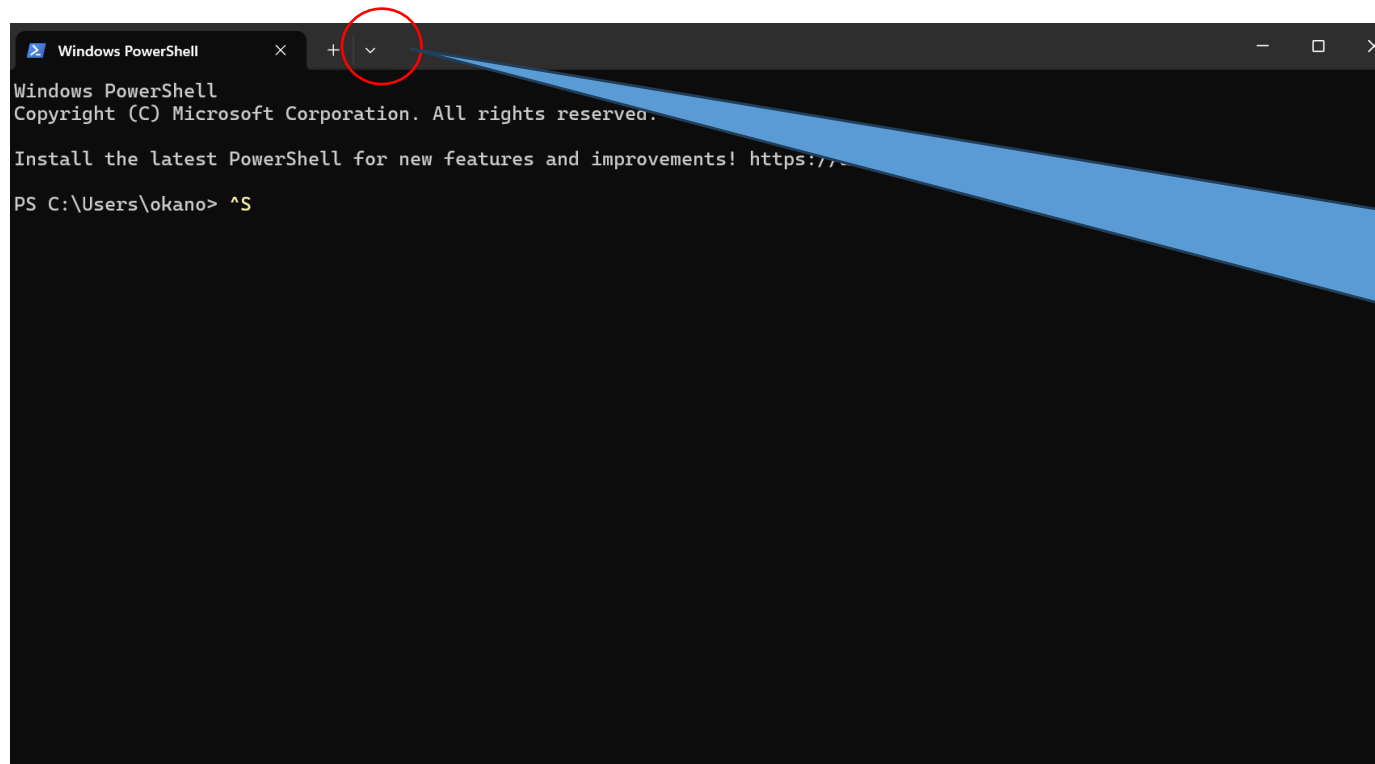
Step 1を実行してください

ssh リモートサーバー → あなたのIDリターン

yリターン

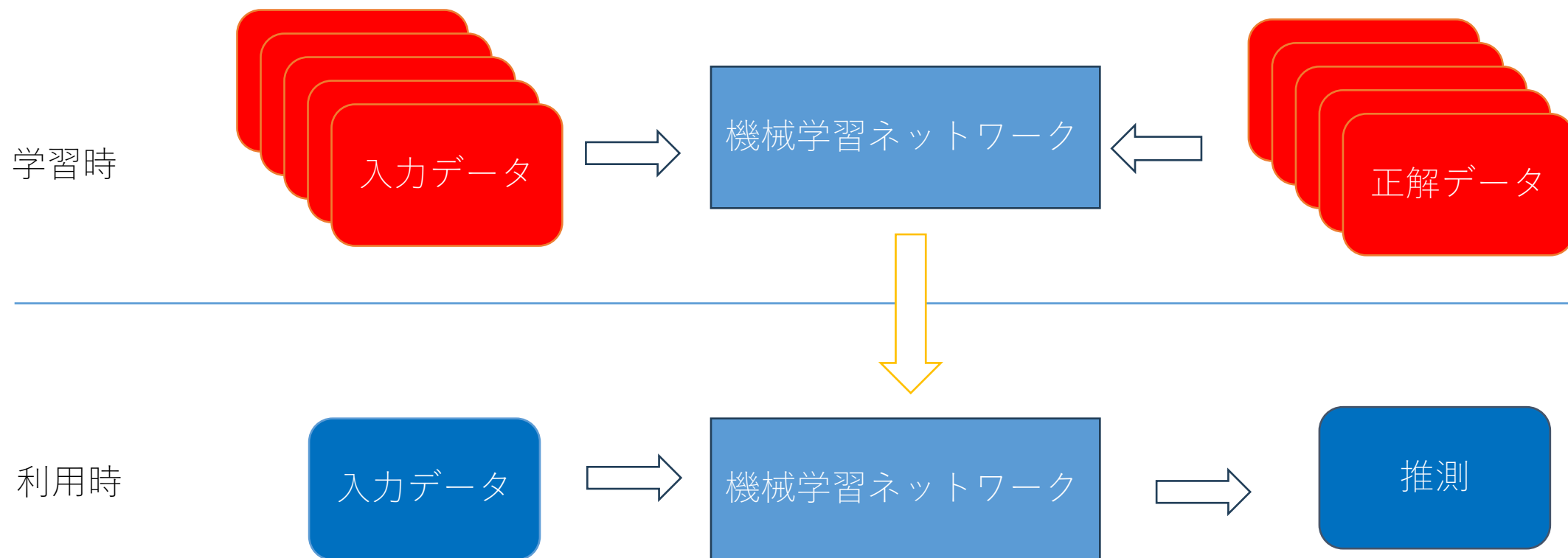
Password リターン

- Windows terminal で別のウィンドウをあけて
- 手元の紙のおまじないStep2 をと覚えてください



ここをクリックして
powerShellをも
う一つ開く

機械学習の基本



mnistの中身を見てみよう

```
STREAMLITPORT=8181
```

```
export STREAMLITPORT
```

```
alias ls='ls -F --color'
```

```
python mnistData2png.py --max 20
```

```
showMnistImages
```

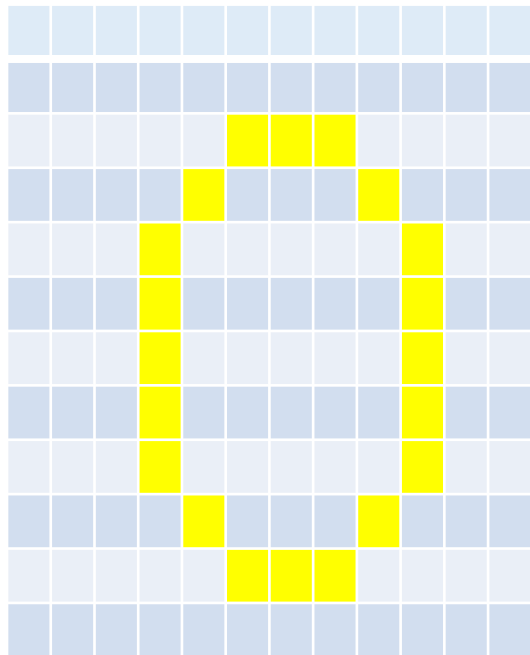
MNIST

- **MNISTデータベース** (Modified National Institute of Standards and Technology database)
- さまざまな画像処理システムの学習に広く使用される手書き数字画像の大規模なデータベース
- 機械学習分野での学習や評価に広く用いられている
 - 60,000枚の訓練用画像
 - 10,000枚の評価用画像

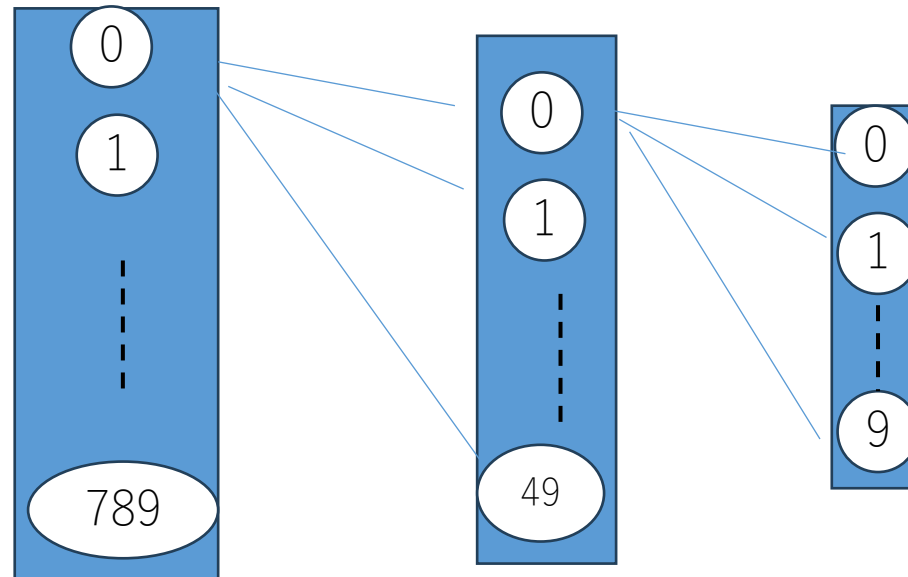


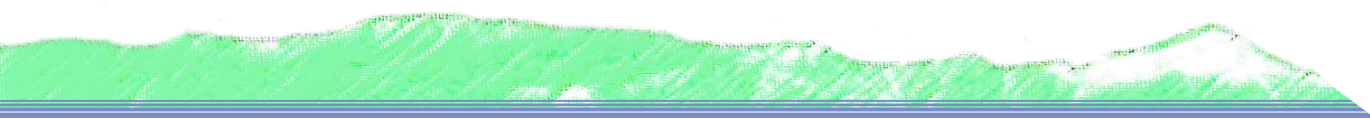
MNISTを判別する簡単なFCNN

- 2層構造
- 1層目 入力 $786 = 28 * 28 + 2$
- 2層目 入力 50
- 出力 10 one-shot



一次元配列にする





プログラムの説明

Mnist (Modified National Institute of Standards and Technology)

```
cd /app/home/Mnist  
python mnistData2png.py --max 20  
./showMnistImages
```

```
python main.py -mdl fcn  
./showResult  
./discriminate
```

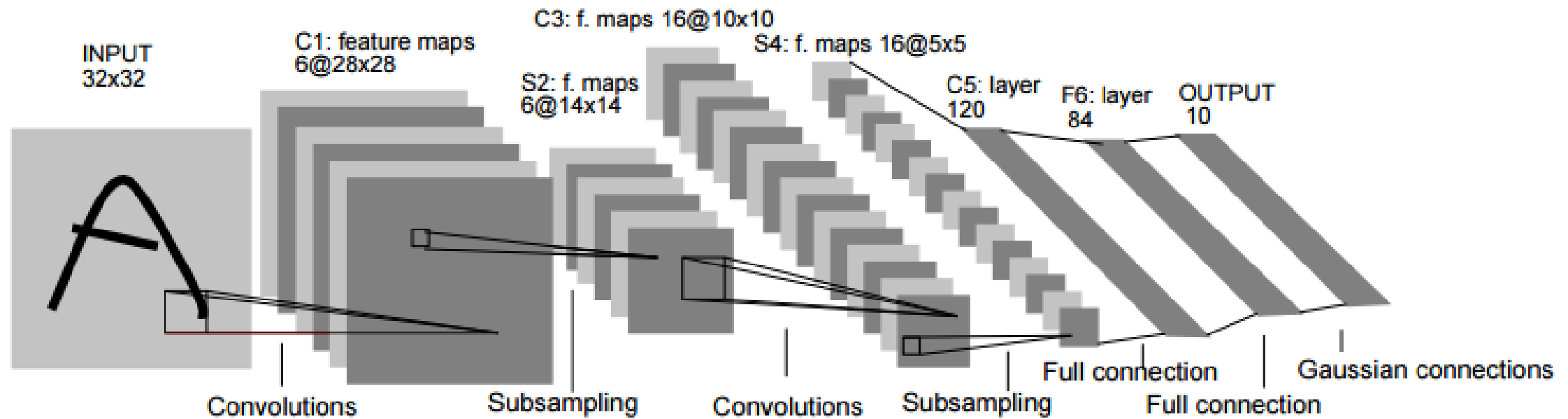
学習結果で実際に認識させてみよう

- 指示通り動かしてみてください
- 認識精度はどれくらいでしょうか

CNN (Convolutional Neural Network)の登場

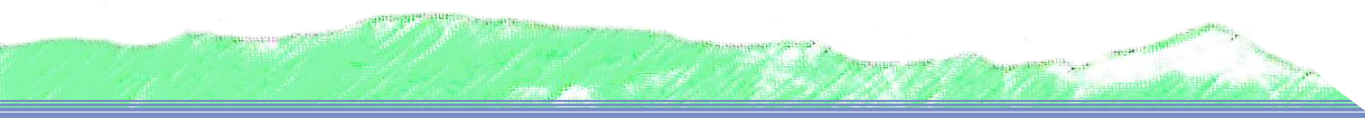
- 画像処理につよい機械学習アーキテクチャとしてCNNが考え出された
- 画像フィルターの概念をニューロンにとりいれた
- 画像フィルターの重みは従前のニューロンの重みと同様に逆伝搬法で学習可能
- 画像フィルターの導入により画像の特徴量の形で出力との関係を学習できる
- 結果として画像認識をより人間に近い感覚でできるようになった。

CNNの具体例 LeNet5



改良版でうごかしてみよう

- 指示どおり動かしてみてください



mnist

```
python main.py
```

```
./showResult
```

```
./discriminate
```

```
python main.py --epoch 20
```

```
./showResult
```

```
./discriminate
```