

# 機械学習（11回目）

創域理工学部 情報計算科学科

桂田 浩一

1

10/19/2023

## 前回の復習

2

- 深層学習（1）
  - 層を多くした時の問題点
  - 問題の解決方法（3種類）

10/19/2023

3

## 本日の内容

- 深層学習（2）
  - 画像処理とCNN

10/19/2023

4

## 画像処理とフィルタ

- フィルタとは？
  - （画像等の）局所領域に適用する畳み込み演算のこと

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

×

|               |               |               |
|---------------|---------------|---------------|
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |

=

|   |               |               |   |
|---|---------------|---------------|---|
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |

フィルタ  
(平均値フィルタ：平滑化)

畳み込み演算： $1 \times \frac{1}{9} + 1 \times \frac{1}{9} + 1 \times \frac{1}{9} + \dots = 1$

10/19/2023

5

## 画像処理とフィルタ

### ■ フィルタとは？

- (画像等の) 局所領域に適用する畳み込み演算のこと

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

 $\times$ 

|               |               |               |
|---------------|---------------|---------------|
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |

 $=$ 

|   |               |               |   |
|---|---------------|---------------|---|
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |

画像                      フィルタ  
 (平均値フィルタ：平滑化)

畳み込み演算:  $1 \times \frac{1}{9} + 1 \times \frac{1}{9} + 0 \times \frac{1}{9} + \dots = \frac{2}{3}$

10/19/2023

6

## 画像処理とフィルタ

### ■ フィルタとは？

- (画像等の) 局所領域に適用する畳み込み演算のこと

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

 $\times$ 

|               |               |               |
|---------------|---------------|---------------|
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |

 $=$ 

|   |               |               |   |
|---|---------------|---------------|---|
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |

画像                      フィルタ  
 (平均値フィルタ：平滑化)

畳み込み演算:  $1 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + \dots = \frac{1}{3}$

10/19/2023

7

## 画像処理とフィルタ

### ■ フィルタとは？

- (画像等の) 局所領域に適用する畳み込み演算のこと

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

 $\times$ 

|               |               |               |
|---------------|---------------|---------------|
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |
| $\frac{1}{9}$ | $\frac{1}{9}$ | $\frac{1}{9}$ |

 $=$ 

|   |               |               |   |
|---|---------------|---------------|---|
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |
| 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 |

画像

フィルタ  
(平均値フィルタ：平滑化)

$$\text{畳み込み演算} : 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + 0 \times \frac{1}{9} + \dots = 0$$

10/19/2023

8

## 画像処理とフィルタ

### ■ フィルタとは？

- (画像等の) 局所領域に適用する畳み込み演算のこと

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

 $\times$ 

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

 $=$ 

|   |   |   |   |
|---|---|---|---|
| 0 | 4 | 4 | 0 |
| 0 | 4 | 4 | 0 |
| 0 | 4 | 4 | 0 |
| 0 | 4 | 4 | 0 |

画像

フィルタ  
(ソーベルフィルタ  
：縦のエッジ検出)

$$\text{畳み込み演算} : 1 \times 1 + 1 \times 0 + 1 \times (-1) + \dots = 0$$

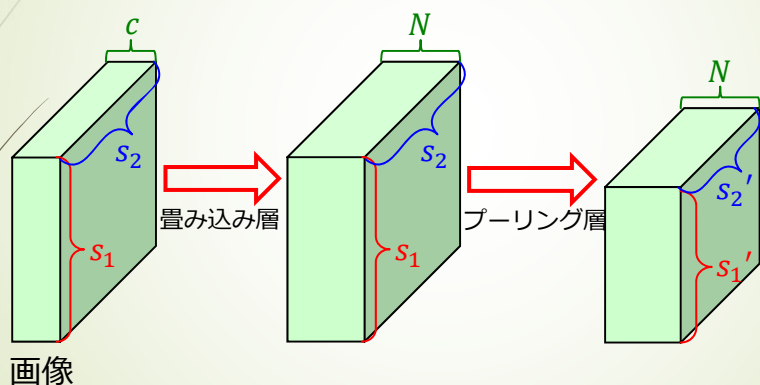
↑ ニューラルネットワークの  $\sum x_i w_{j,i}$  の式と似てる

10/19/2023

9

## CNN(Convolutional Neural Network : 畳み込みニューラルネット)

■ 主に画像処理で用いられるネットワーク



画像

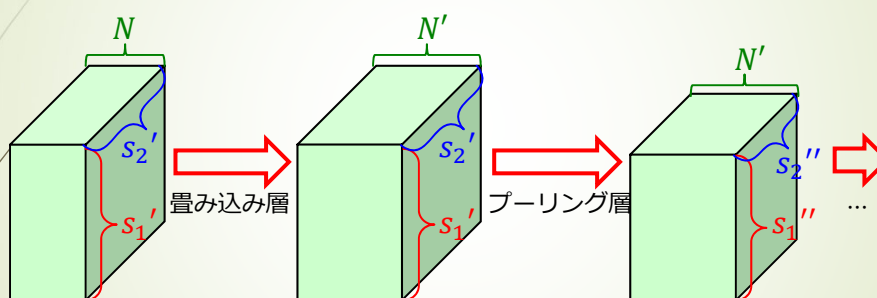
$s_1 \times s_2$  : 画素  $c$  : チャンネル (カラーなら3, モノクロなら1)

10/19/2023

10

## CNN(Convolutional Neural Network : 畳み込みニューラルネット)

■ 主に画像処理で用いられるネットワーク



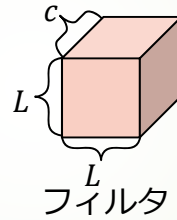
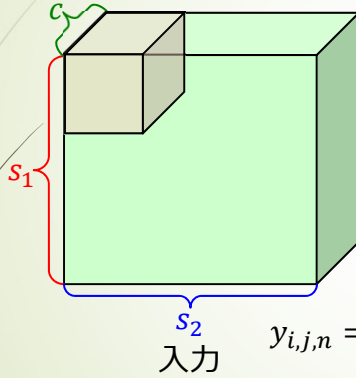
10/19/2023

11

## 畳み込み層(Convolution layer)

左上から  $l$  画素ずつずらしながらフィルタをかける (通常  $l = 1$ )

$l$  をストライドという



$$y_{i,j,n} = f(u_{i,j,n}) \quad (p, q, r) : \text{フィルタ内の座標}$$

$$u_{i,j,n} = \left( \sum_{p=0}^{L-1} \sum_{q=0}^{L-1} \sum_{r=0}^{C-1} \underbrace{x_{i+p,j+q,r}}_{\text{入力}} \underbrace{w_{p,q,r,n}}_{\text{フィルタ}} \right) + b_{i,j,n}$$

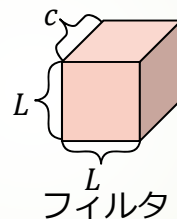
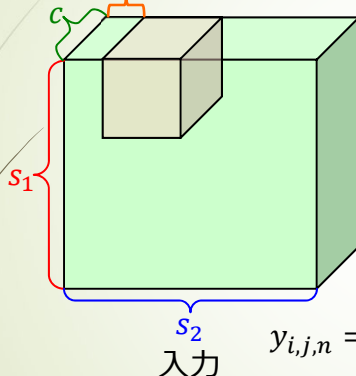
10/19/2023

12

## 畳み込み層(Convolution layer)

左上から  $l$  画素ずつずらしながらフィルタをかける (通常  $l = 1$ )

$l$  をストライドという



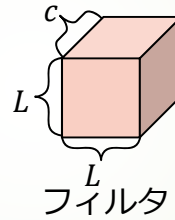
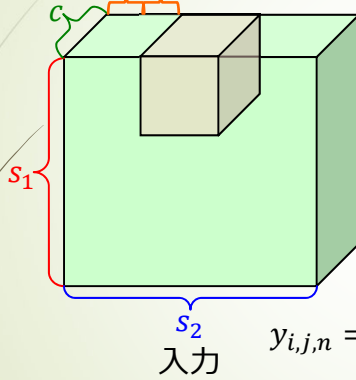
$$y_{i,j,n} = f(u_{i,j,n}) \quad (p, q, r) : \text{フィルタ内の座標}$$

$$u_{i,j,n} = \left( \sum_{p=0}^{L-1} \sum_{q=0}^{L-1} \sum_{r=0}^{C-1} \underbrace{x_{i+p,j+q,r}}_{\text{入力}} \underbrace{w_{p,q,r,n}}_{\text{フィルタ}} \right) + b_{i,j,n}$$

10/19/2023

13

## 畳み込み層(Convolution layer)

左上から  $l$  画素ずつずらしながらフィルタをかける (通常  $l = 1$ ) $l$  をストライドという

入力

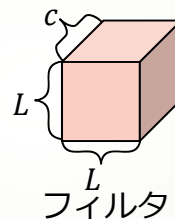
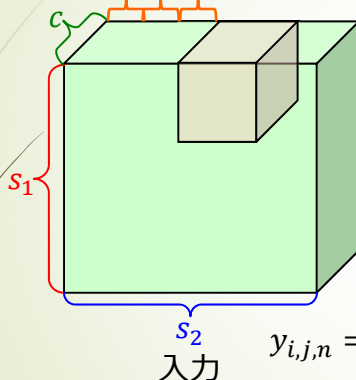
$$y_{i,j,n} = f(u_{i,j,n}) \quad (p, q, r) : \text{フィルタ内の座標}$$

$$u_{i,j,n} = \left( \sum_{p=0}^{L_1-1} \sum_{q=0}^{L_2-1} \sum_{r=0}^{C-1} \underbrace{x_{i+p,j+q,r}}_{\text{入力}} \underbrace{w_{p,q,r,n}}_{\text{フィルタ}} \right) + b_{i,j,n}$$

10/19/2023

14

## 畳み込み層(Convolution layer)

左上から  $l$  画素ずつずらしながらフィルタをかける (通常  $l = 1$ ) $l$  をストライドという

入力

$$y_{i,j,n} = f(u_{i,j,n}) \quad (p, q, r) : \text{フィルタ内の座標}$$

$$u_{i,j,n} = \left( \sum_{p=0}^{L_1-1} \sum_{q=0}^{L_2-1} \sum_{r=0}^{C-1} \underbrace{x_{i+p,j+q,r}}_{\text{入力}} \underbrace{w_{p,q,r,n}}_{\text{フィルタ}} \right) + b_{i,j,n}$$

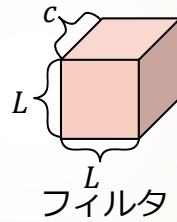
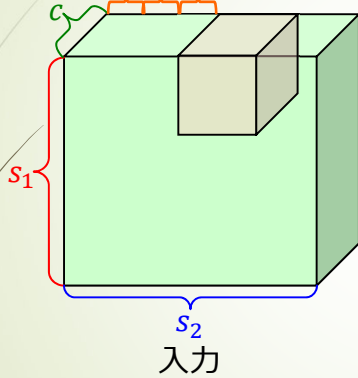
10/19/2023

15

## 畳み込み層(Convolution layer)

左上から  $l$  画素ずつずらしながらフィルタをかける (通常  $l = 1$ )

$l$  をストライドという



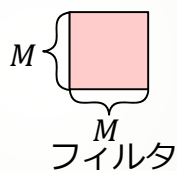
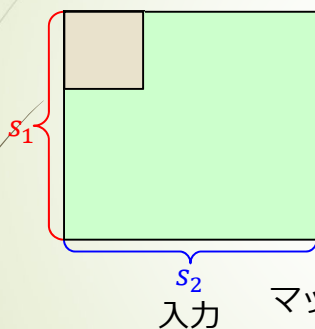
$l = 1$  のとき, 出力の大きさは  $(s_1 - L + 1) \times (s_2 - L + 1) \times N$

元画像の周囲を0で埋める (0-padding) と出力を  $s_1 \times s_2 \times N$  にできる

16

## プーリング層(Pooling layer)

左上から  $m$  画素ずつずらしながらフィルタをかける (通常  $m = M \geq 2$ )



マックスプーリング  $y_{i,j,n} = \max_{p,q} (x_{i+p,j+q,n})$

平均値プーリング  $y_{i,j,n} = \frac{\sum_{p,q} x_{i+p,j+q,n}}{M^2}$

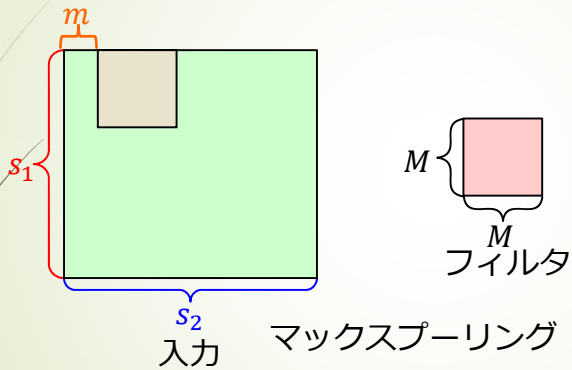
10/19/2023



17

## プーリング層(Pooling layer)

左上から  $m$  画素ずつずらしながらフィルタをかける (通常  $m \geq 2$ )



マックスプーリング  $y_{i,j,n} = \max_{p,q} (x_{i+p,j+q,n})$

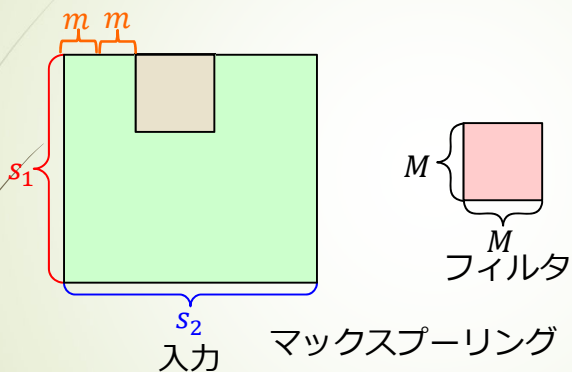
平均値プーリング  $y_{i,j,n} = \frac{\sum_{p,q} x_{i+p,j+q,n}}{M^2}$

10/19/2023

18

## プーリング層(Pooling layer)

左上から  $m$  画素ずつずらしながらフィルタをかける (通常  $m \geq 2$ )



マックスプーリング  $y_{i,j,n} = \max_{p,q} (x_{i+p,j+q,n})$

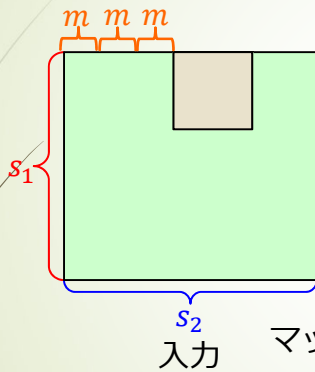
平均値プーリング  $y_{i,j,n} = \frac{\sum_{p,q} x_{i+p,j+q,n}}{M^2}$

10/19/2023

19

## プーリング層(Pooling layer)

左上から  $m$  画素ずつずらしながらフィルタをかける (通常  $m \geq 2$ )



$M$   
 $M$   
フィルタ

マックスプーリング  $y_{i,j,n} = \max_{p,q} (x_{i+p,j+q,n})$

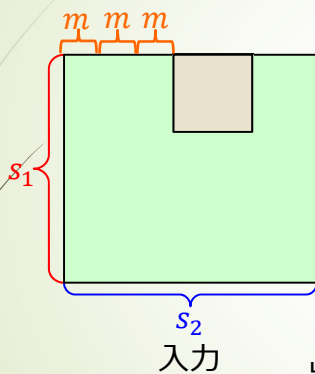
平均値プーリング  $y_{i,j,n} = \frac{\sum_{p,q} x_{i+p,j+q,n}}{M^2}$

10/19/2023

20

## プーリング層(Pooling layer)

左上から  $m$  画素ずつずらしながらフィルタをかける (通常  $m \geq 2$ )



$M$   
 $M$   
フィルタ

出力の大きさは約  $\frac{s_1}{m} \times \frac{s_2}{m} \times N$  となる

10/19/2023

21

## CNNの学習

- 基本的には通常の誤差逆伝播法を用いる.
- プーリング層では, マックスプーリングの場合「最大値」となった画素にのみ逆伝播する. 平均値プーリングの場合, 誤差  $\delta$  の  $1/M^2$  の値を逆伝播する.
- 畳み込み層では通常の誤差逆伝播法をベースに, 次のように異なった逆伝播を行う.
  - フィルタ外への誤差の伝搬は行わない
  - フィルタ内の伝播は荷重が共有されているとみなす

10/19/2023

22

## DNNで用いられる様々なテクニック

1. ミニバッチ学習
  - 学習安定化のための手法
  - 学習データが  $N$  個与えられたとき
    - 一つのデータごとに重みを更新 ... オンライン法
    - $N$  個のデータで誤差を求め, 誤差の平均値で重みの更新値を求める ... バッチ法
    - $N/t$  個のデータで誤差を求め, 誤差の平均値で重みの更新値を求める ... ミニバッチ法
2. Dropout
  - 学習の高速化 + 過学習の防止
    - 学習時にノード (+それに接続されている結合) を半分程度間引く

10/19/2023

23

## DNNで用いられる様々なテクニック

### 3. データの正規化

- 学習を効率的にする
  - 入力の正規化
  - ミニバッチ毎に（活性化関数をかける直前の）出力の正規化 ... バッチ正規化
  - 層毎にノード間で値を正規化 ... Layer normalization

### 4. 様々な最適化法

- 勾配降下法を工夫して学習効率を上げる
  - モーメンタム法, AdaGrad, Adamなど

10/19/2023

24

## DNNで用いられる様々なテクニック

### 5. 転移学習 (Transfer learning)

- 別の（大規模）データで学習したネットワークを流用してファインチューニングする


### 6. 知識蒸留 (Knowledge distillation)

- 別の（大規模）データで学習した大規模ネットワークの出力を教師データとして、小規模ネットワークを学習する

### 7. 敵対的学習 (Adversarial training)

- 学習したいモデルと、それを妨げるモデルを同時に学習することにより学習の精度を上げる

10/19/2023



## 出題予定の演習課題

- CNNについて