

Deep Learning Seminar

~4~

渡邊研究室 M1 二見悠樹

今日の目標

畳み込みニューラルネットワーク(Convolutional Neural Networks; CNN)の構造について理解する

目次

1. Deep Learningブームのはじまり
2. Deep Learning以前と以後で何が変わったか
3. 画像処理におけるフィルタ
4. 畳み込みニューラルネットワーク

前回のおさらい

- 学習を行うまでのデータ整形を行いました

課題の答え

MLPで256クラス分類は無理です

→ 画像の画素データを生で入力しているのは非常に冗長

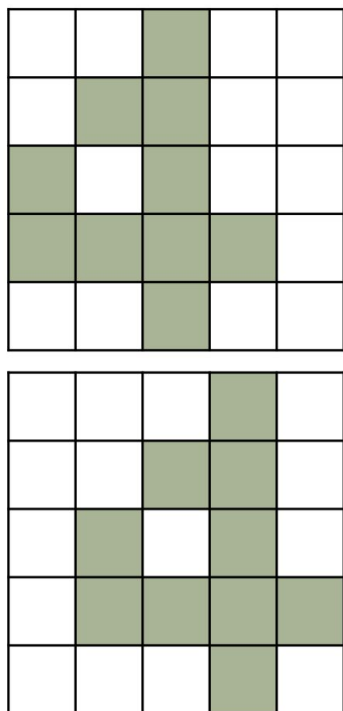
→ そもそもメモリに乗り切らなかったんじゃない？

→ そういうわけでニューラルネットのブームは80年代以降下火になるわけ

何故学習が上手くいなかったのか？

- 全結合層は2次元のデータを1次元のベクトルとして扱う

→ ちょっとでも位置がずれていると全く別のデータなのでこれをどげんとせんといかん



[0, 0, 1, 0, 0, 0, 1, 1, 0, 0, ...]

[0, 0, 0, 1, 0, 0, 0, 1, 1, 0, ...]

25の要素を持つベクトル

NNの歴史

第1期

冬の時代

第2期

冬の時代

第3期

1960

1970

1980

1990

2000

2010

パーセプトロン
[Rosenblatt, 55]

単純型・複雑型細胞
[Hubel-Wiesel, 59]

形式ニューロン
[McCulloch, 47]

ネオコグニトロン
[Fukushima, 80]

誤差逆伝播法
[Rumelhart+, 86]

非線形問題
[Minsky-Papert, 69]

積層自己符号化器
[Hinton+, 06]

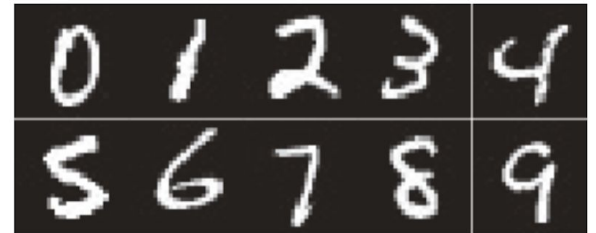
深層畳み込みNN
[Krizhevsky+, 12]

畳み込みNN
[LeCun+, 99]

2010年頃まで...

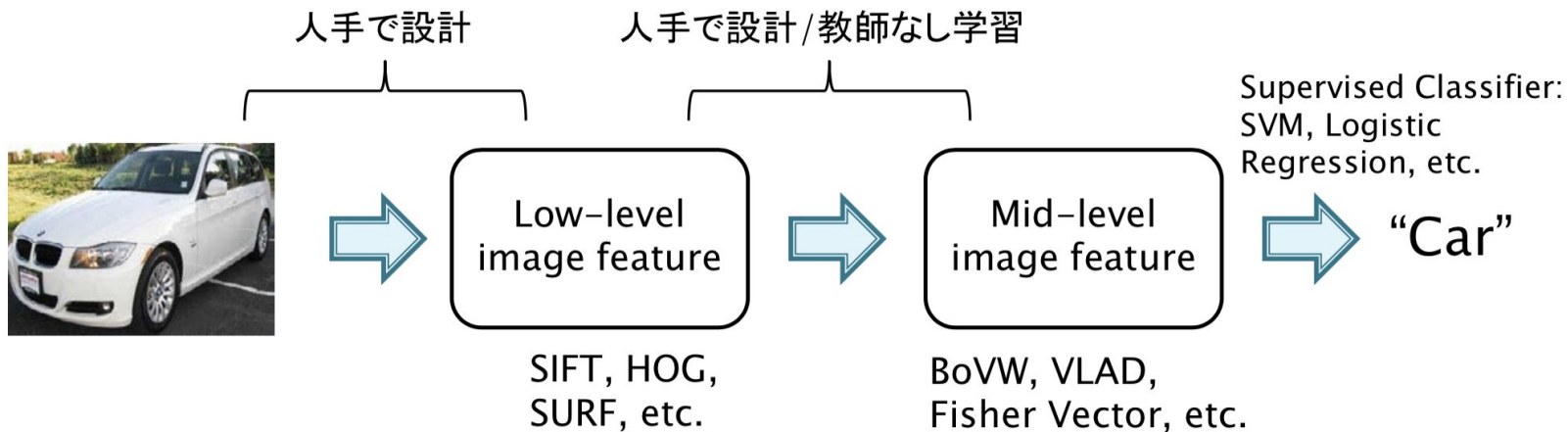
▶ 小さな画像を用いた基礎研究が主流

- MNISTデータセット [LeCun]
 - 文字認識、 28×28 ピクセル、6万枚
- CIFAR-10/100 データセット [Krizhevsky]
 - 物体認識、 32×32 ピクセル、5万枚

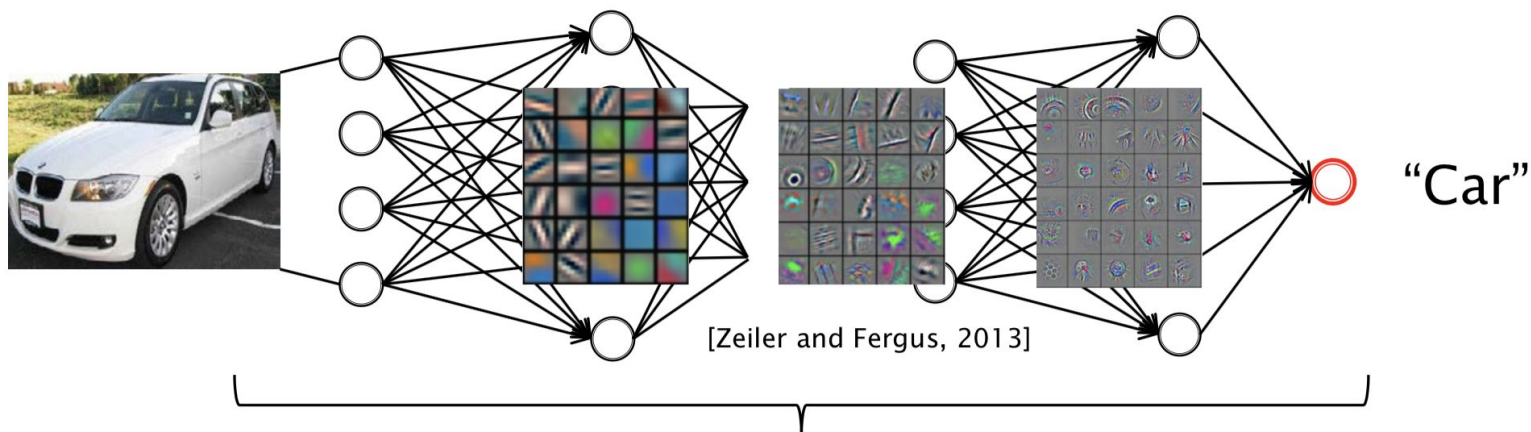


Deep以前と以後

伝統的
方法論
("Shallow"
learning)



Deep
learning



すべて学習で行う！つまりは人での設計が不要になった

一般的な画像処理 フィルタについて

フィルタ: 重みを持った小さな窓 (2x2, 3x3 ...)

→ 入力画像の左上から順にこのフィルタを通して各値を重み付けして足し合わせると画像の平滑化やエッジ検出ができたりする

77	80	82	78	70	82	82
83	78	80	83	82	77	94
87	82	81	80	74	75	112
87	87	85	77	66	99	151
84	79	77	78	76	107	162
86	72	70	72	81	151	166
78	72	73	73	107	166	170

入力画像

×

0.01	0.08	0.01
0.08	0.62	0.08
0.01	0.08	0.01

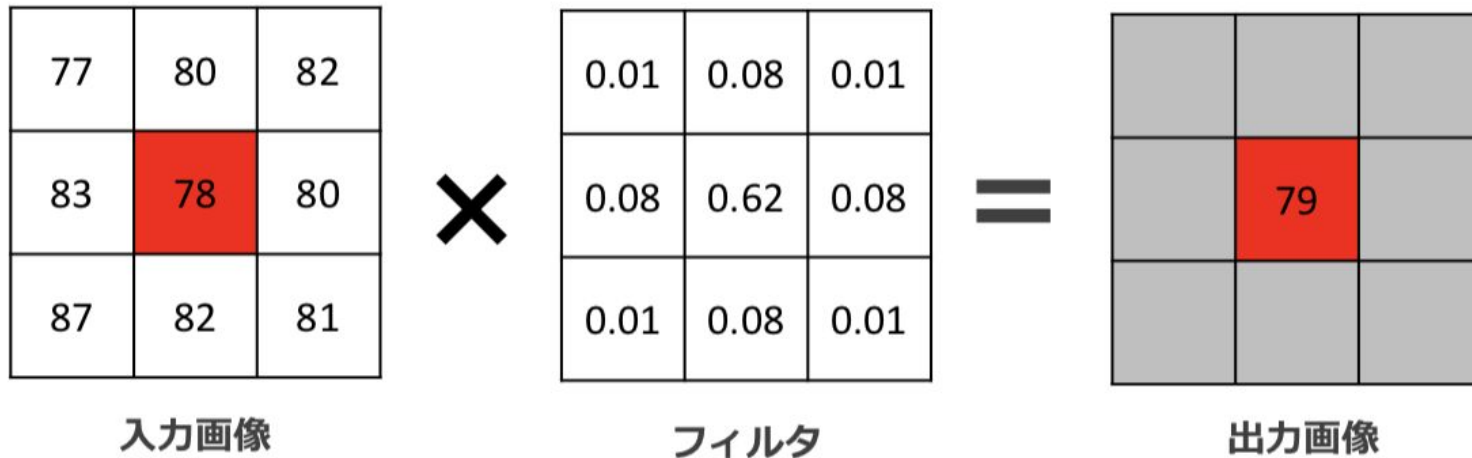
フィルタ

=

	79	80	81	79	79	
	82	81	79	75	81	
	85	83	77	72	99	
	79	77	77	79	112	
	73	71	73	89	142	

出力画像 (特徴マップ)

フィルタの計算



$$77 \times 0.01 + 80 \times 0.08 + 82 \times 0.01 + 83 \times 0.08 + 78 \times 0.62 + 80 \times 0.08 + 87 \times 0.01 + 82 \times 0.08 + 81 \times 0.01 = 79$$

フィルタ例1



入力画像

×

フィルタ

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

=

平滑化（ぼかし、ノイズ除去）をする



×

-1/8	-1/8	-1/8
-1/8	2	-1/8
-1/8	-1/8	-1/8

=

鮮鋭化（シャープ）をする



フィルタ例2



入力画像

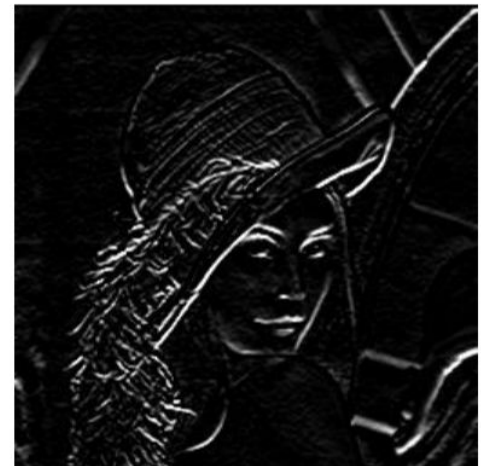
$$\times \begin{array}{|c|c|c|} \hline & \text{フィルタ} & \\ \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array} =$$

縦方向のエッジを検出する

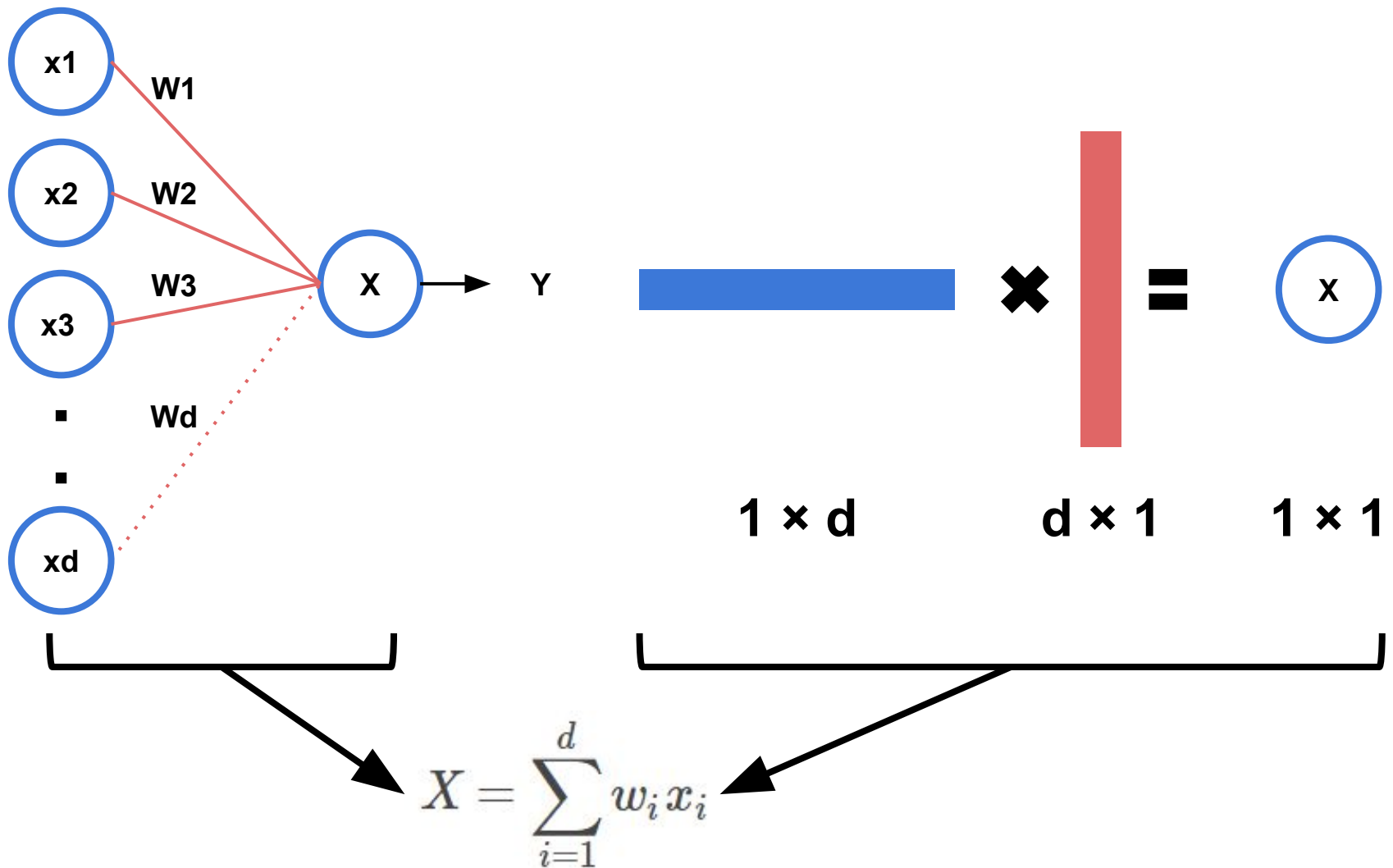


$$\times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array} =$$

横方向のエッジを検出する



おわかりいただけただろうか...



フィルタの計算

77	80	82
83	78	80
87	82	81

入力画像

×

0.01	0.08	0.01
0.08	0.62	0.08
0.01	0.08	0.01

フィルタ

=

	79	

非線形活性化関数(とても重要)

$$77 \times 0.01 + 80 \times 0.08 + 82 \times 0.01 + 83 \times 0.08 + 87 \times 0.01 + 82 \times 0.08 + 81 \times 0.01 = 79$$

$$r = \phi(\mathbf{w} * \mathbf{h} - \theta)$$

フィルタの係数

例えば、5x5の畳み込み、
10チャンネルの入力の場合、
5x5x10=250個

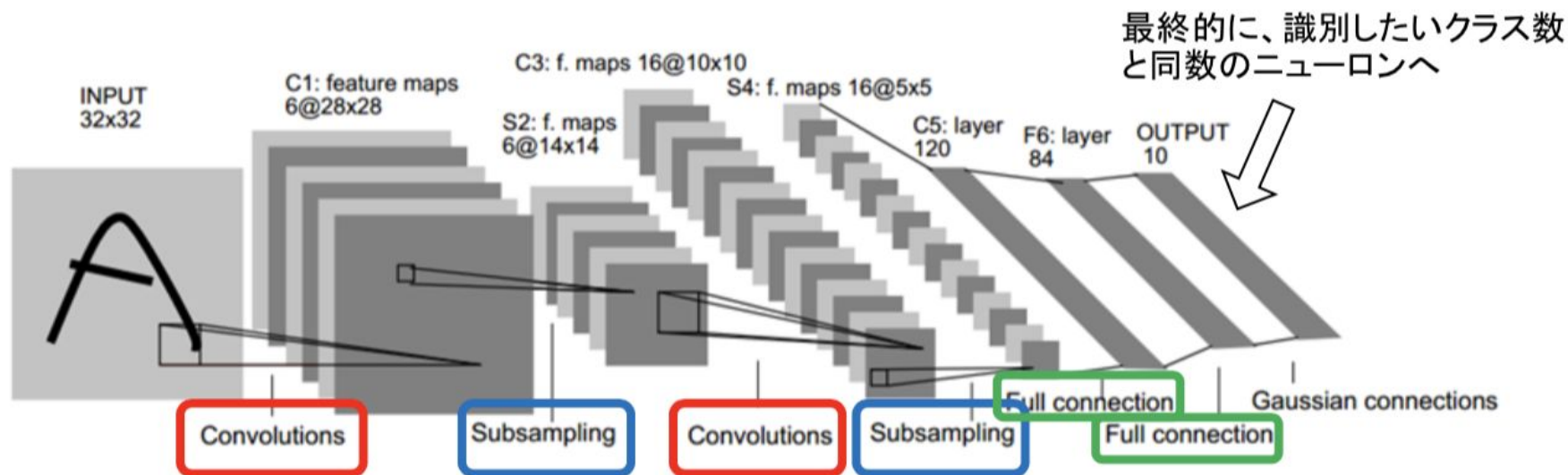
入力

バイアス

Convolutional Neural Network(CNN)

局所領域(受容野)の畳み込みとプーリングを繰り返す多層ネットワーク

ニューロン間の結合を局所に限定することでパラメータ数の大幅な削減

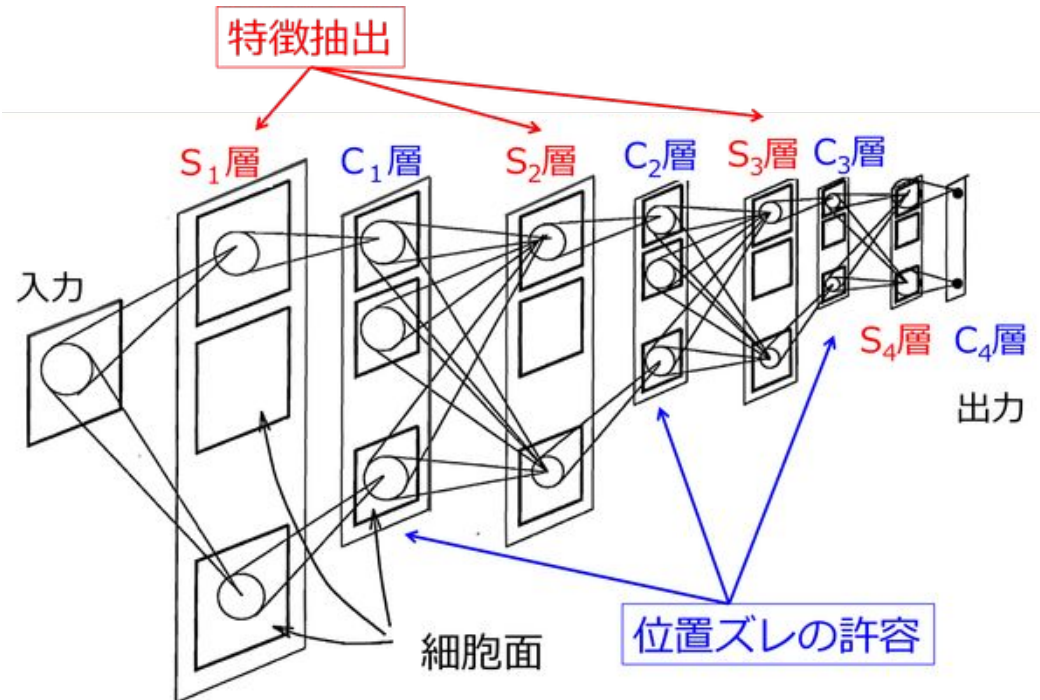
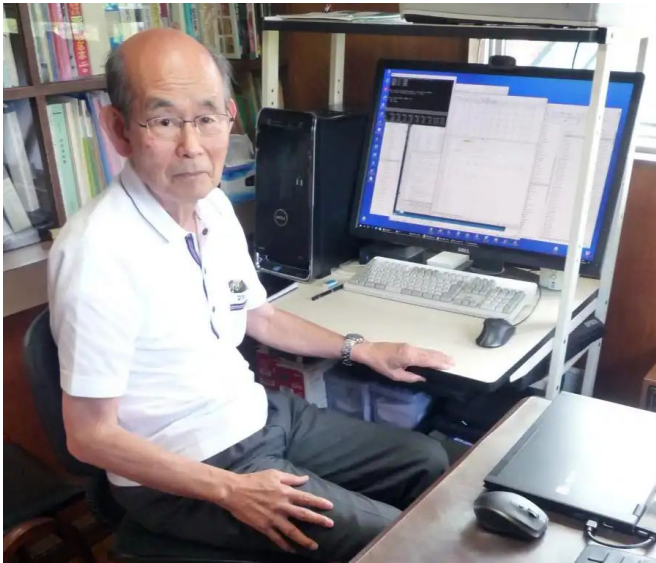


ネオコグニトロン

実は最初に提唱したのは日本人！

福島邦彦氏が1982年に発表したネオコグニトロンが元ネタ

脳の視覚野にある2つの細胞：単純型細胞と複雑型細胞を数理モデル化

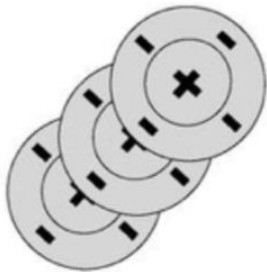


単純型細胞

特徴抽出を行ってる

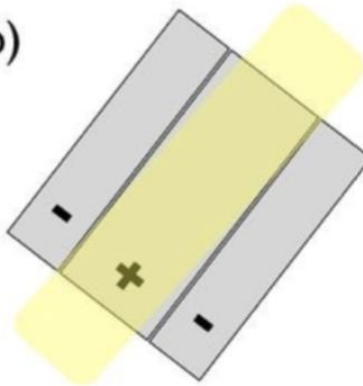
単純型細胞

c)



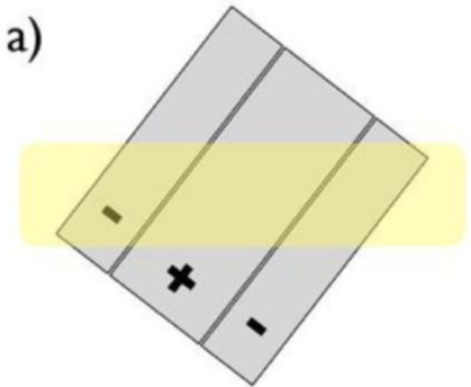
受容野を並べて単純細胞を形成

b)



+に光、-には当てないと興奮反応

a)

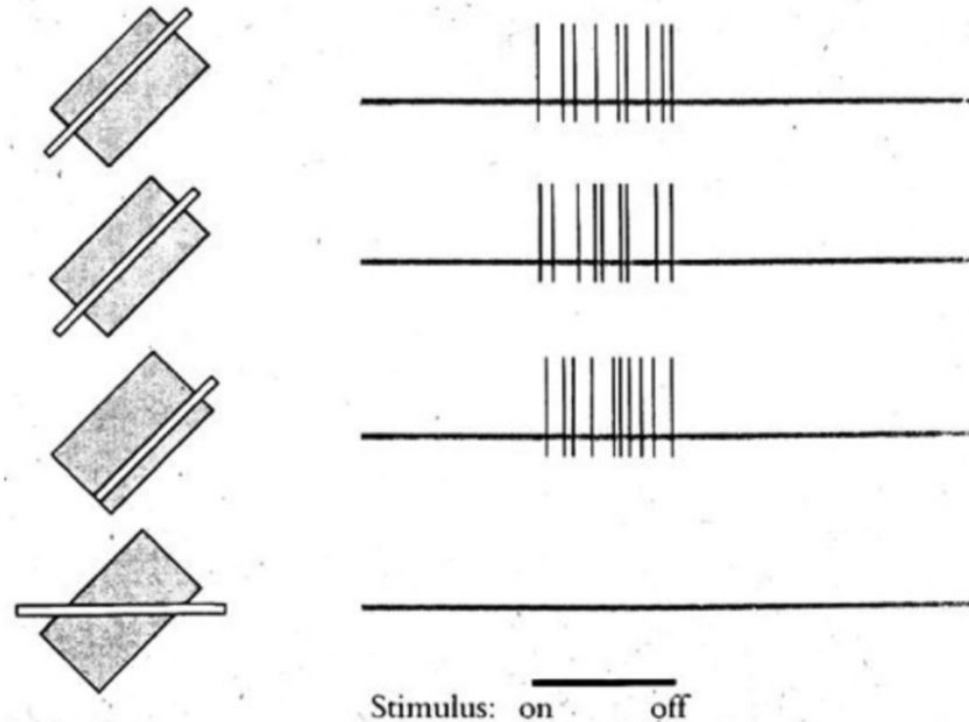


+に光、-にも光を当てると
相殺されて興奮反応なし

複雑型細胞

位置ずれを許容する働きを持つ

複雑型細胞



平行移動しても反応が起こるが、回転すると反応しない。

畳み込み

畳み込み処理の注意点

1 回目

81	80	74	75	112
85	77	66	99	151
77	78	76	107	162
70	72	81	151	166
73	73	107	166	170

入力画像

×

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

フィルタ

=

	77	81	102	
	75	89	117	
	78	101	131	

出力画像

3x3のフィルタで処理した場合、周囲1画素ずつ画像が小さくなる

2 回目

	77	81	102	
	75	89	117	
	78	101	131	

入力画像

×

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

フィルタ

=

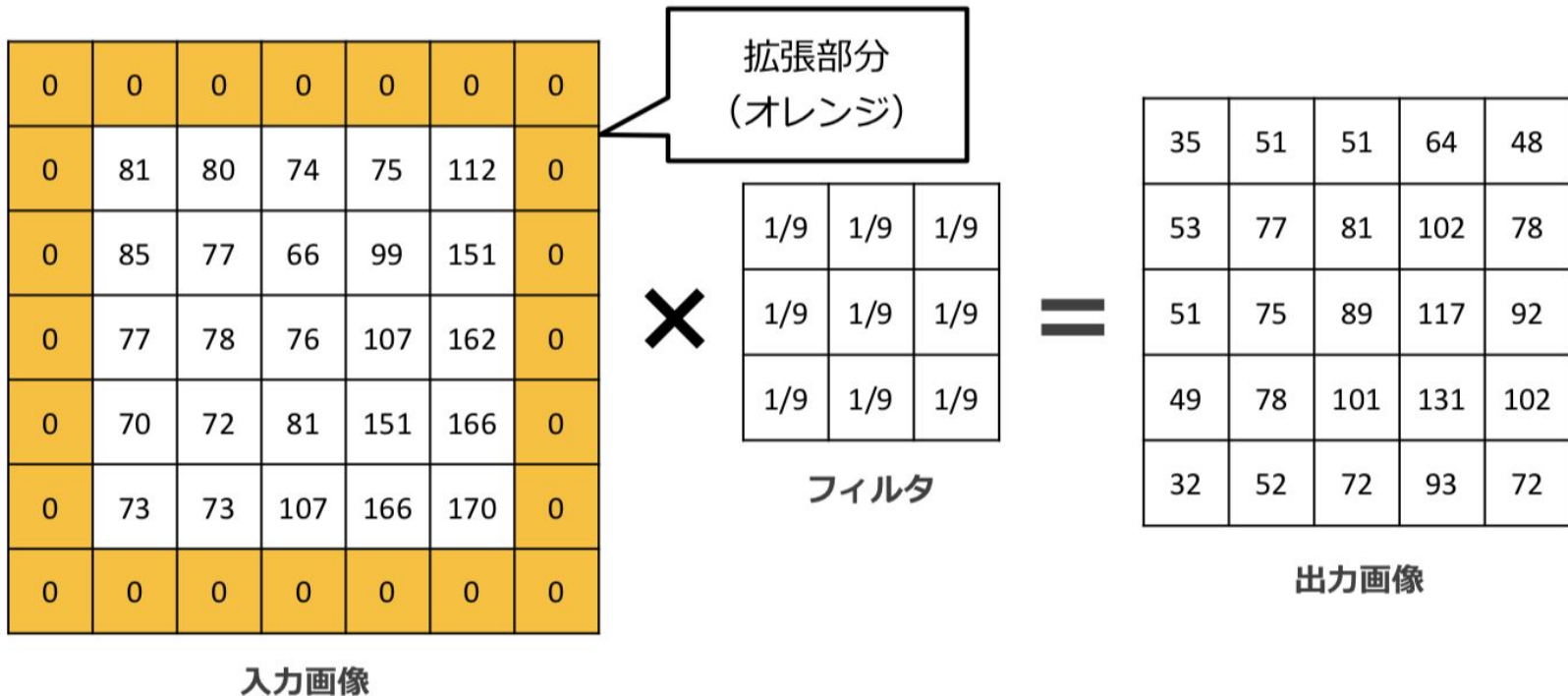
		94		

出力画像

ゼロパディング

画像サイズを拡張すればいいじゃないか

→ とりまゼロでいいしょ



プーリング層

- プーリングとは、特徴抽出から認識に余分な情報を捨て、認識に必要な情報を保ったままの表現に変換する処理である
- サブサンプリングとも呼ばれる
- プーリング層は通常、畳み込み層の直後に設置される。

→ ユニット数を減らすため、結果として調整するパラメータを削減できる

→ 対象となる特徴量の画像内での位置が若干ずれている場合でも、プーリング層の力が不変となるようにする(位置不変性の獲得)

プーリング層

プーリングサイズを2x2で実行した場合、縦横のサイズが1/2となり、データが1/4に圧縮される

12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12



20	30
112	37

最大プーリング
(max pooling)

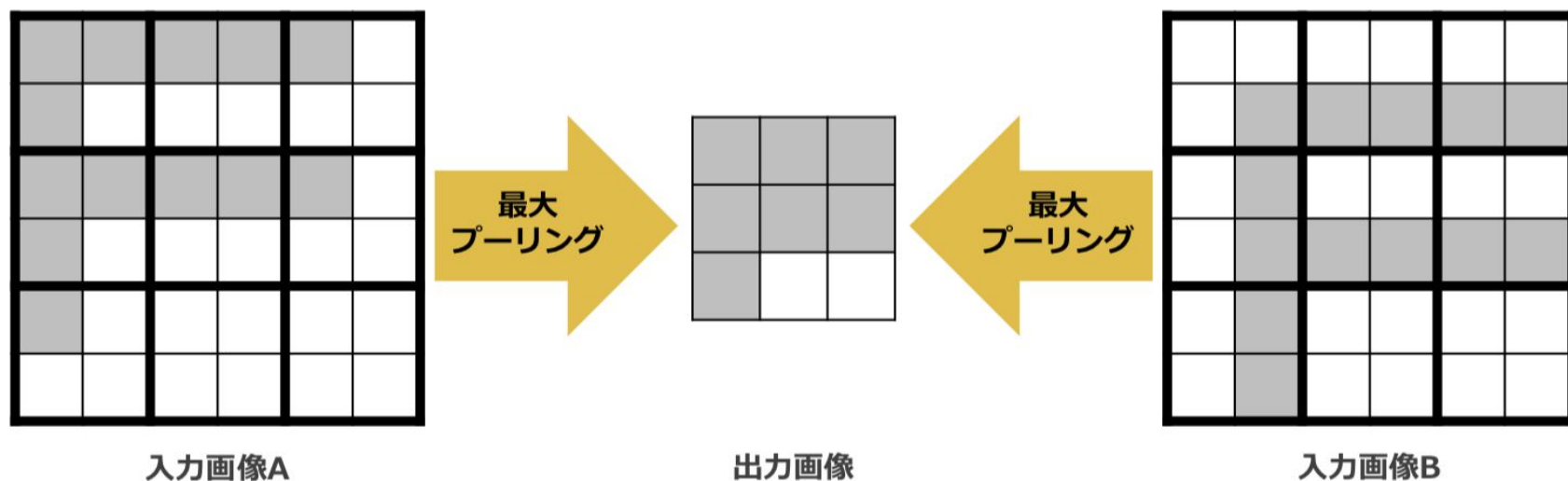


13	8
79	20

平均プーリング
(average pooling)

位置変動への不変性

不変な特徴量が抽出されたと言える(よね?)



それでは...

<https://qiita.com/yu4u/items/7e93c454c9410c4b5427>

CNNを実装してみよう！

次回予定

❖ みんなでバトルしよう！

- 最も良い画像認識モデルを作った人が**優勝**
- データセットはCaltech256
- MLP,CNN **何でもアリ(この際DLじゃなくてもおk)**

❖ 学習環境は統一させます

- Batch_size: 32
- Epoch: 30

❖ PCスペックが足りない人は僕のマシンで回すのでモデルが出来たらソースコードください.

❖ 結果発表:2週間後

学習環境

caltech256

epoch :30

batch_size =32

steps_per_epoch 200

validation_steps 100

訓練データとテストデータの割合も自由

(データの増しは有効→新地が詳しい)

たくさんぐぐっていいアーキテクチャを利用しよう！