

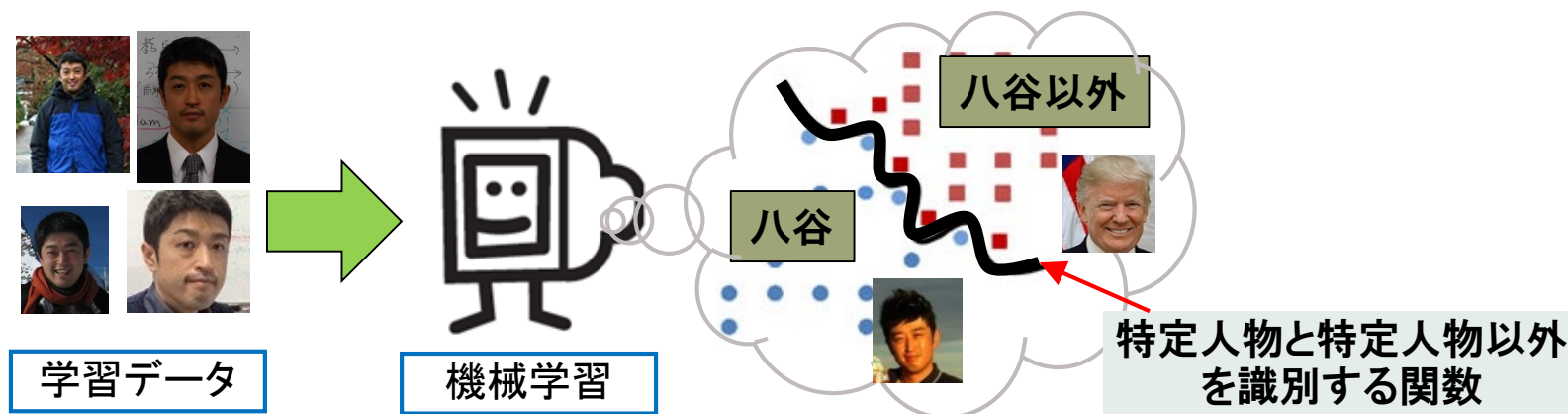
# 機械学習の概要・応用例

2019年5月29日 八谷 大岳

# 機械学習とは

2

- データから知識・ルールを獲得する方法の総称
  - ▣ 人が知識やルールを明示的に与える方法の限界から生まれた

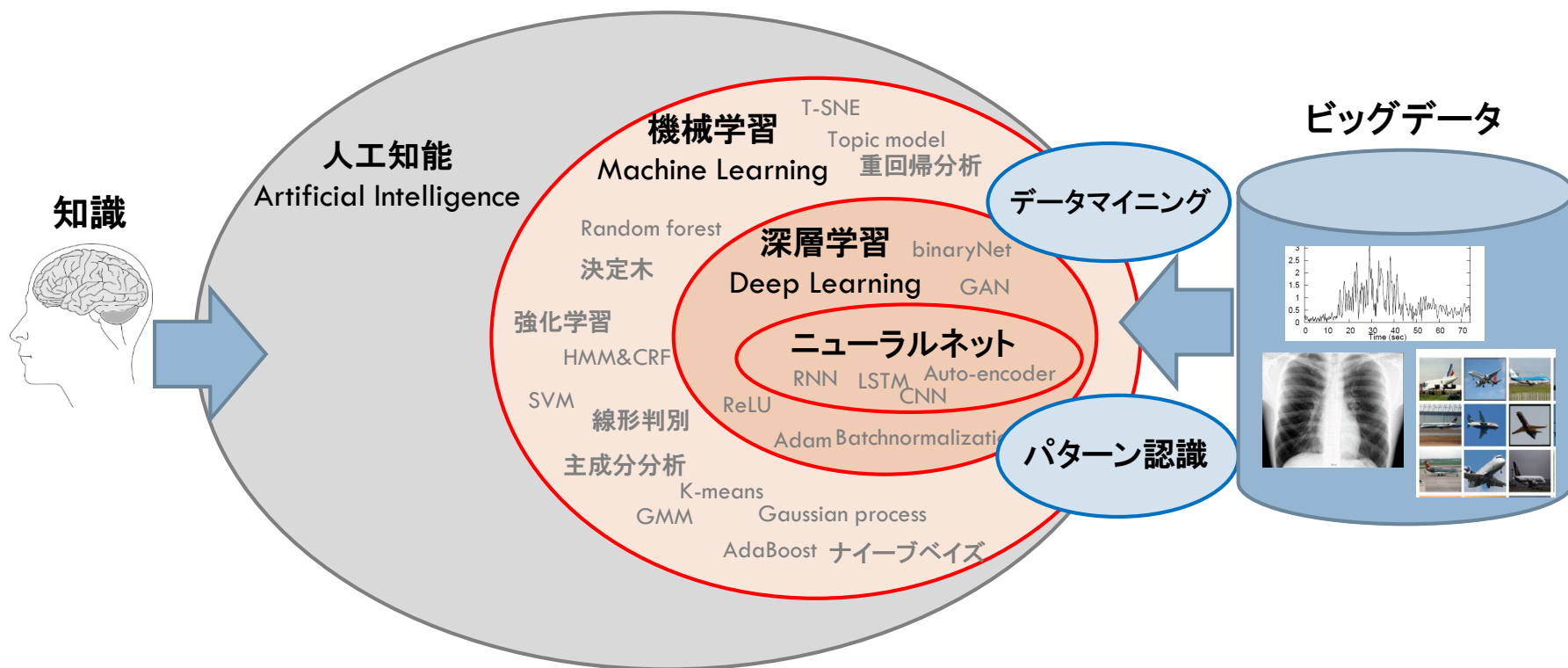


- ▣ 2000年頃から統計的学習技術の研究が盛んに行われてきた
- ▣ 2012年頃からビッグデータブームと、Deep learningの登場で、機械学習の分野を超えて一躍注目を集め始める

# 機械学習と人工知能の関係

3

- **機械学習：人工知能の研究分野の一つ**
  - 人間の知識の活用よりも、データからの知識・ルールの獲得を重視
  - 「データマイニング」、「パターン認識」とアプローチ・目的が類似



# 内容

4

- 機械学習とは
- 機械学習が急速に広まった理由
- 教師あり学習の概要と応用例
- 深層学習(Deep Learning)を用いた特徴抽出
- より発展的な機械学習
  - 異常検知の概要と応用例
  - 強化学習の概要と応用例

# 機械学習が急速に広まった理由1

5

- あらゆる分野で大量のデータが生成されている

テキスト

行動履歴

画像

音声

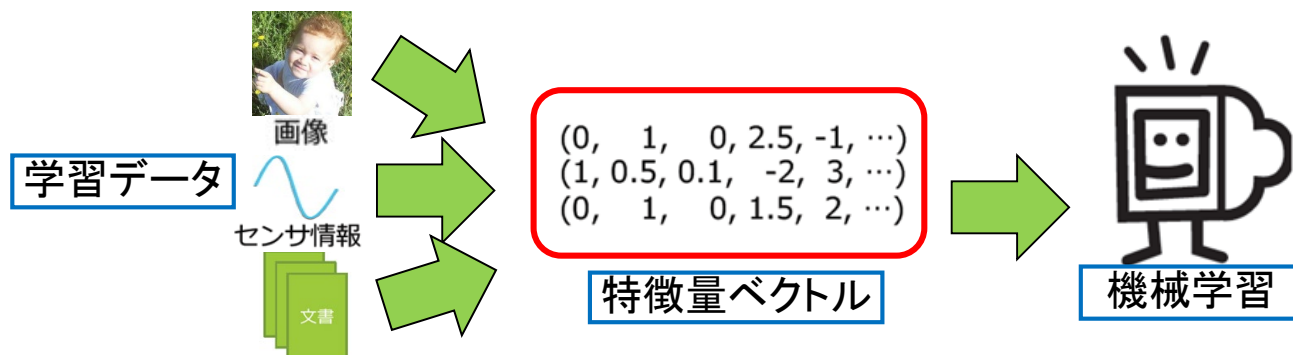
映像

信号

金融

ゲノム

- 特徴量表現と分析処理の分離



- 問題や分野の特徴を捉えた特徴量抽出により、  
どの分野にも機械学習を適用できる

# 機械学習が急速に広まった理由2

6

## □ 無料のライブラリが拡充

### □ scikit-learn, Tensorflow, Caffe, chainer, libsvm, SVMlight,...

```
# 540/2 = 270
convW0 = weight_variable("convW0", [4, 1, 32])
convB0 = bias_variable("convB0", [32])
conv0 = conv1d_relu(x, convW0, convB0, stride=2)

# 270/2 = 135
convW1 = weight_variable("convW1", [4, 32, 32])
convB1 = bias_variable("convB1", [32])
conv1 = conv1d_relu(conv0, convW1, convB1, stride=2)

# 135/3 = 45
convW2 = weight_variable("convW2", [6, 32, 32])
convB2 = bias_variable("convB2", [32])
conv2 = conv1d_relu(conv1, convW2, convB2, stride=3)

# 45/5 = 9
convW3 = weight_variable("convW3", [10, 32, 32])
convB3 = bias_variable("convB3", [32])
conv3 = conv1d_relu(conv2, convW3, convB3, stride=5)

# convert to vector
conv3 = tf.reshape(conv3, [-1, np.prod(conv3.get_shape().as_list()[1:])])

# 9x32 = 288 -> 128
fcW1 = weight_variable("fcW1", [9*32, 128])
fcB1 = bias_variable("fcB1", [128])
fc1 = fc_relu(conv3, fcW1, fcB1)

# 128 -> 20
fcW2 = weight_variable("fcW2", [128, z_dim])
fcB2 = bias_variable("fcB2", [z_dim])
fc2 = fc_relu(fc1, fcW2, fcB2)
```

5階層のConvolutional neural networkが  
数十行のPythonコードで実装できるようになった

### □ 特徴量の設計さえできれば誰でも一般的な機械学習を適用できる時代

## □ しかし、使いこなすためには機械学習の知識が必要

### □ 機械学習は数学の異種格闘技

■ 確率統計、線形代数、微積分、情報理論および最適化

# 機械学習が急速に広まった理由3

7

- **Deep Learning**の登場により、数多くのタスク(画像認識、顔認識、戦略獲得など)で従来の技術および人間を凌駕
- 大規模画像分類Imagenetで、**従来の画像認識技術に10%の大差**で優勝



【2012年の結果】

	Team name	Error
1	SuperVision	0.153
2	ISI (SHIFT+FV, LBP+FVなど)	0.262
3	OXFORD_VGG	0.270
4	XRCE/INRIA	0.271
5	Univ. of Amsterdam	0.296

黒色:従来の画像認識  
赤色:Deep Learning

【2014年の結果】

	Team name	Error
1	VGG	0.074
2	GoogLeNet	0.148
3	SYSU_Vision	0.319
4	MIL	0.337
5	MSRA Visual Computing	0.355

分類誤差:7%まで低減!

上位は全部深層学習

# 内容

8

- 機械学習とは
- 機械学習が急速に広まった理由
- 教師あり学習の概要
- 深層学習(Deep Learning)を用いた特徴抽出
- より発展的な機械学習
  - ▣ 異常検知の概要と応用例
  - ▣ 強化学習の概要と応用例



# 機械学習の種類

9

問題	定義	代表的な方法	応用例
教師あり学習	入力と出力のデータに基づき、入力を出力に変換する関数を学習	SVM, 最小二乗法、決定木、ランダムフォレストなど	顔認識、物体検出、手書き文字分類、株価予測、降水量予測など
教師なし学習	入力のみデータに基づき、入力の特性（パターン、構造）を学習	PCA, LDA, HMMなど	データの可視化（クラスタリング、次元圧縮）
半教師学習	入力のうち部分的に付与された出力の事例に基づき入力を出力に変換する関数を学習	transductive SVM, Laplacian SVMなど	画像、音声、Webログなどの大量データで、コストの問題で一部のデータのみしか出力（答え）が付与されていない場合
強化学習	入力と、出力に対する報酬（評価）のデータに基づき、入力を出力に変換する関数を学習	Q-learning、policy iteration, policy gradient	ロボット制御、Web広告選択、マーケティング

- 準備可能なデータと目的に合わせて適切な機械学習方法を選択する必要

# 教師あり学習：回帰

10

- 入力 $x$ と出力 $y$ (数値)の訓練データを準備
  - **訓練**: 誤差を最小化するように、数値を予測する関数を学習

【訓練フェーズ】



入出力データを準備

CM総時間 $x$	販売台数 $y$
23	76
24	77
29	82
27	84
23	74
24	81
27	86



入力 $x$  (CM総時間)から  
出力 $y$  (販売台数)を予測する  
関数を学習

$$y = f(x)$$

学習する関数

出力: 数値データ    入力: 任意のデータ

$$\text{誤差} = \sum_{i=1}^N (y^i - f(x^i))^2$$

- **運用**: 学習した関数を用いて、入力に対するラベルを予測

【運用フェーズ】

入力 $x$   
CM総時間: 30時間



出力 $y = f(x)$   
販売台数: 90台

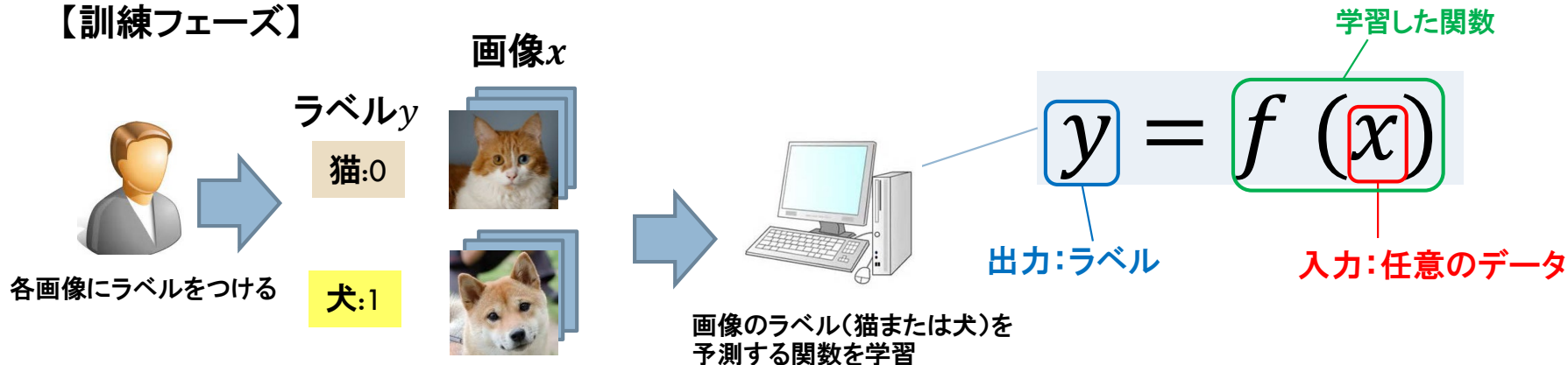
学習した関数を用いて、数値を予測

# 教師あり学習：分類

11

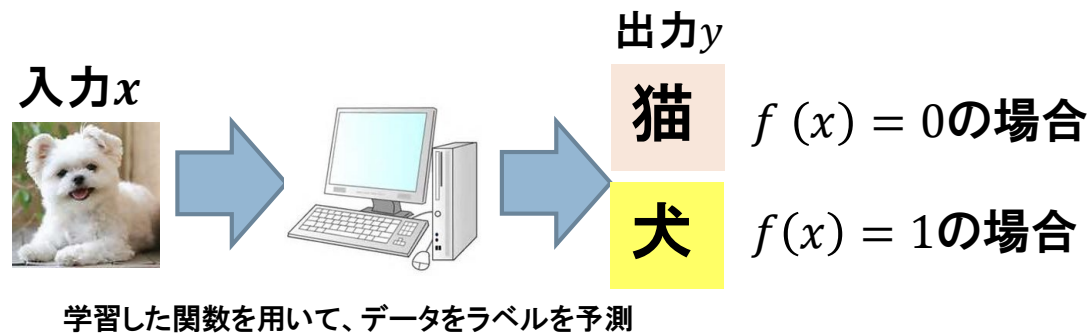
- 人間が予めラベル付けしたデータ(猫、犬など)を準備
  - **訓練**: 分類誤差を最小化するように、ラベルを出力する関数を学習

## 【訓練フェーズ】



- **運用**: 学習した関数を用いて、入力に対するラベルを予測

## 【運用フェーズ】



# 関数 $f(x)$ の基本形

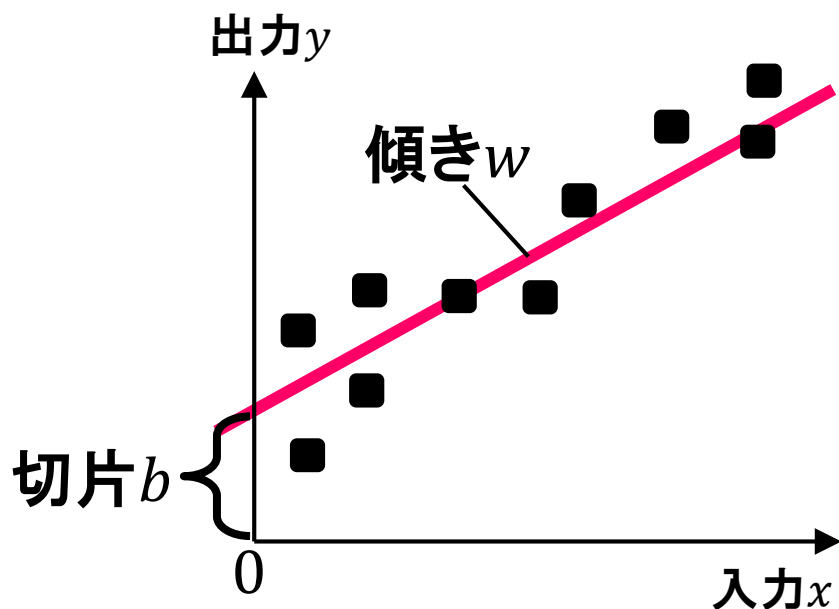
12

- 回帰: 直線関数

$$y = f(x) = wx + b$$

- 学習するパラメータ:

$w$ : 傾き、 $b$ :  $y$ の切片

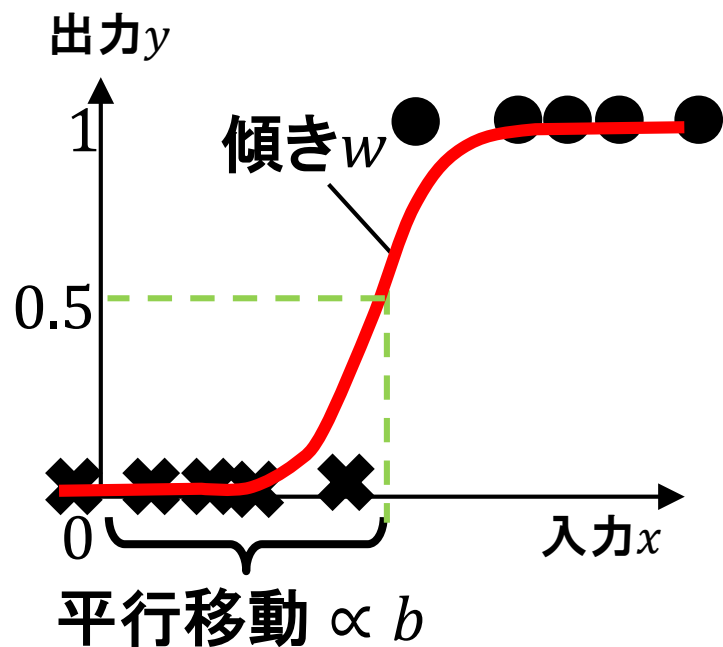


- 分類: シグモイド関数

$$y = f(x) = \frac{1}{1 + e^{-(wx+b)}}$$

- 学習するパラメータ:

$w$ : 傾き、 $b$ :  $x$ の平行移動



# 教師あり学習：分類の例1

13

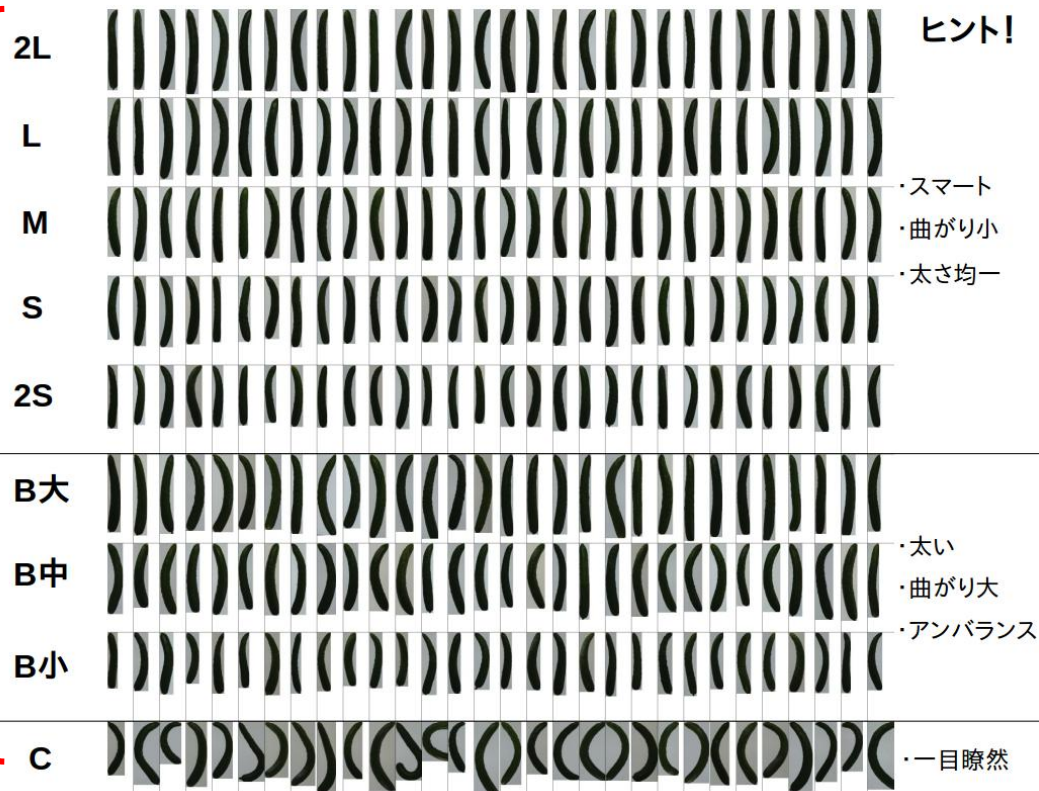


<https://www.youtube.com/watch?v=XkKxSAb4EAw>

# きゅうりの仕分け

14

9種類の仕分けカテゴリ



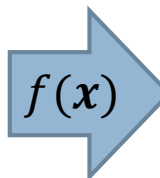
大きさ、形、太さ  
の組み合わせに  
よって決まる

<http://workpiles.com/2017/08/ccb9-proto3-description2/>

入力 $x$ : 画像



...



C L S B小 ...

出力 $y$ : 等級  $\in \{2L, L, M, \dots, C\}$

# 教師あり学習：分類の例2

15

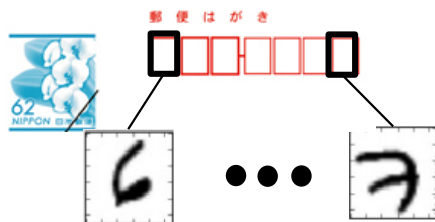
## □ 郵便番号認識

- 1時間に4万通の郵便番号を認識し、地域ごとの仕分けが可能

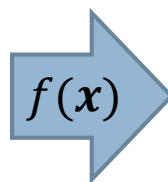


<https://jpn.nec.com/kids/himitsu/04.html>

## □ 手書き数字の分類



入力 $x$ :各文字部分の画像



$\boxed{6} \dots \boxed{7}$

出力 $y$ :数字  $\in \{0, 1, \dots, 9\}$

# 郵便番号自動認識の例





# 内容

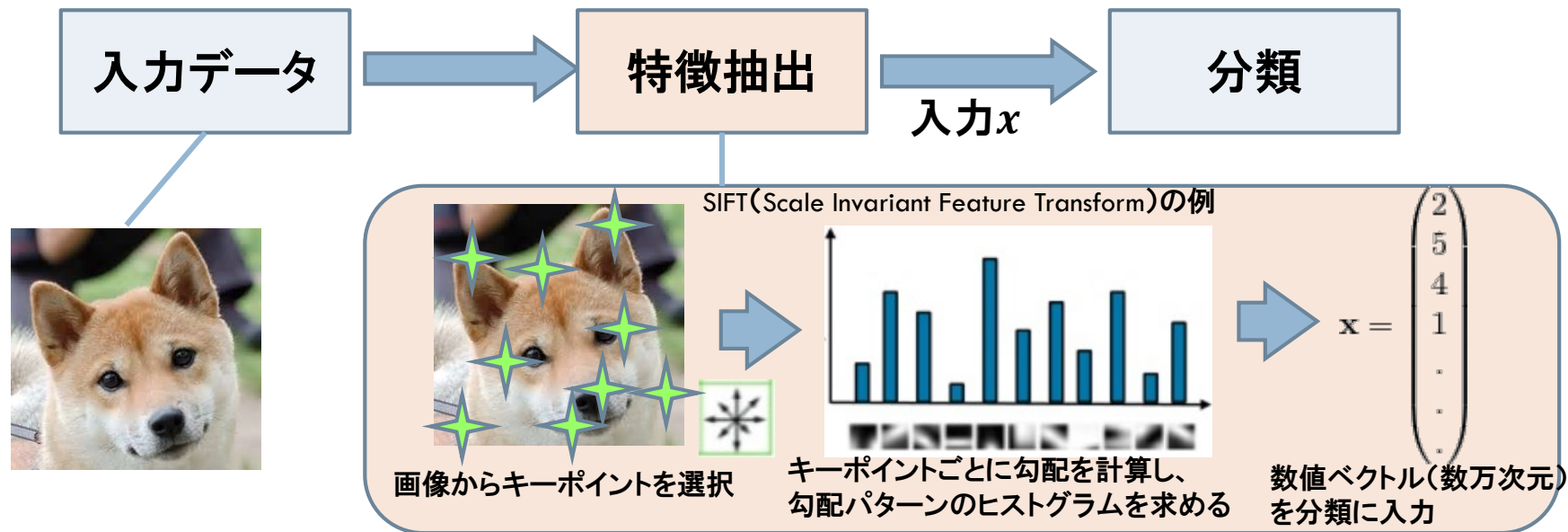
19

- 機械学習とは
- 機械学習が急速に広まった理由
- 教師あり学習の概要
- 深層学習(Deep Learning)を用いた特徴抽出
- より発展的な機械学習
  - ▣ 異常検知の概要と応用例
  - ▣ 強化学習の概要と応用例

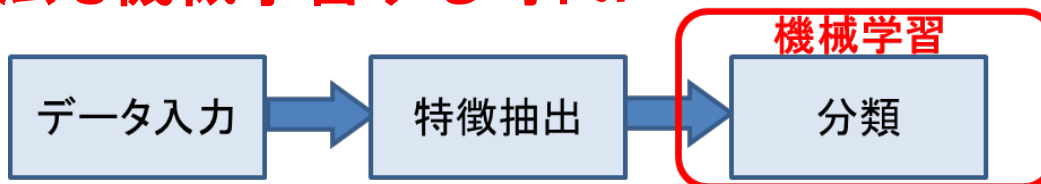
# 教師あり分類問題と特徴抽出

20

- **特徴量**: データから抽出した分類に必要な数値化された情報
  - **特徴抽出方法**: 専門知識に基づき人間が設計



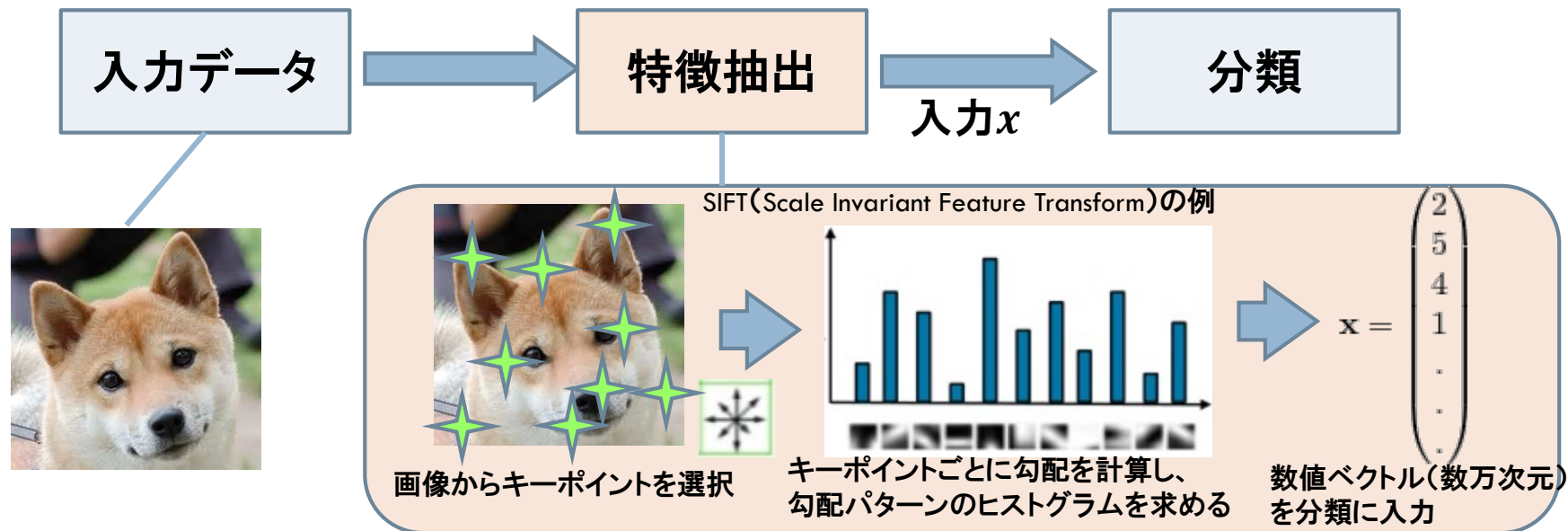
- Deep Learningの登場により、分類精度を最大化するように**特徴抽出方法も機械学習する時代に**



# 教師あり分類問題と特徴抽出

21

- **特徴量**: データから抽出した分類に必要な数値化された情報
  - **特徴抽出方法**: 専門知識に基づき人間が設計



- Deep Learningの登場により、分類精度を最大化するように特徴抽出方法も機械学習する時代に

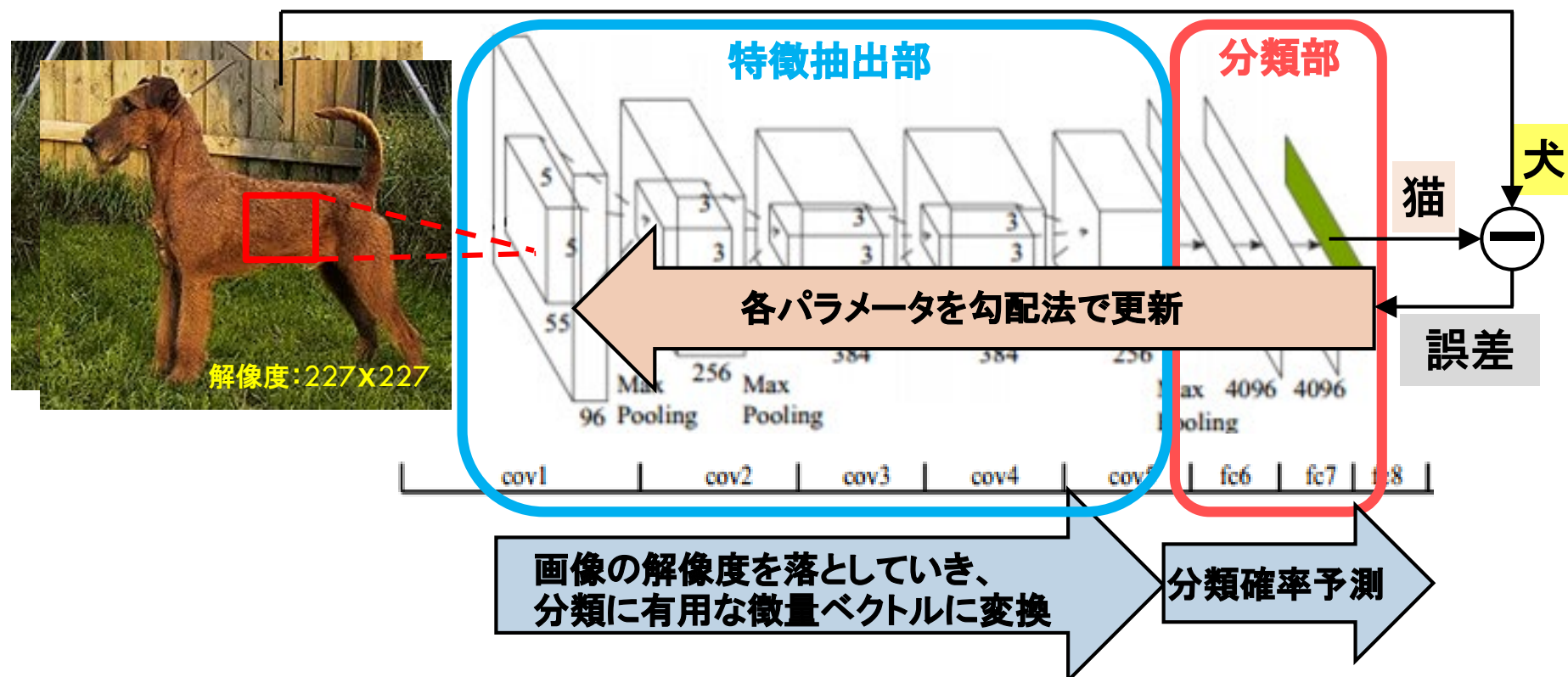


# 深層ネットワークの構成

22

- 特徴抽出部と分類部から構成される
- 画像に特化した特徴抽出部: convolutional neural network

## 【大規模画像分類ImageNet2012で優勝したAlexNet(8層)のモデル】

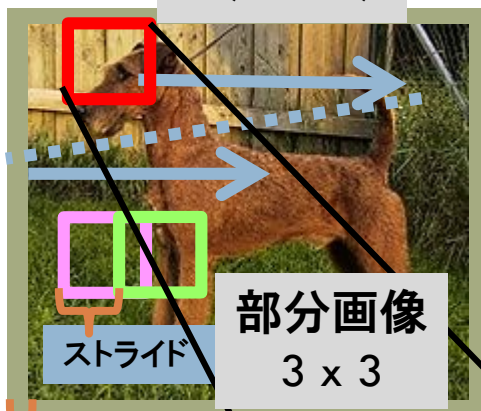


# Convolution(畳み込み)の処理

23

- 入力画像をカーネル(フィルタ)で畳み込み低解像度化

入力画像  
227 x 227



部分画像  
3 x 3

1	0	0
1	1	1
0	1	2

$X$

これらの値を分類精度が  
改善するように決定

カーネル(フィルタ)  
3 x 3

要素ごとの積の和  
(内積)



1	0	1
0	0	1
1	0	1
0	0	0

= 2

重み係数  $W_C$

低解像度画像  
(特徴マップ)

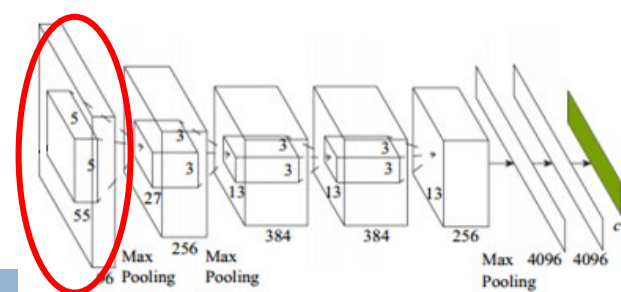
	2	6	5	3	2	0
61	2	5	-6	-8	-5	
1	3	8	-8	-9	-7	
4	1	3	-5	-8	-6	

- Convolutionは、単純な線形変換( $XW_C$ )

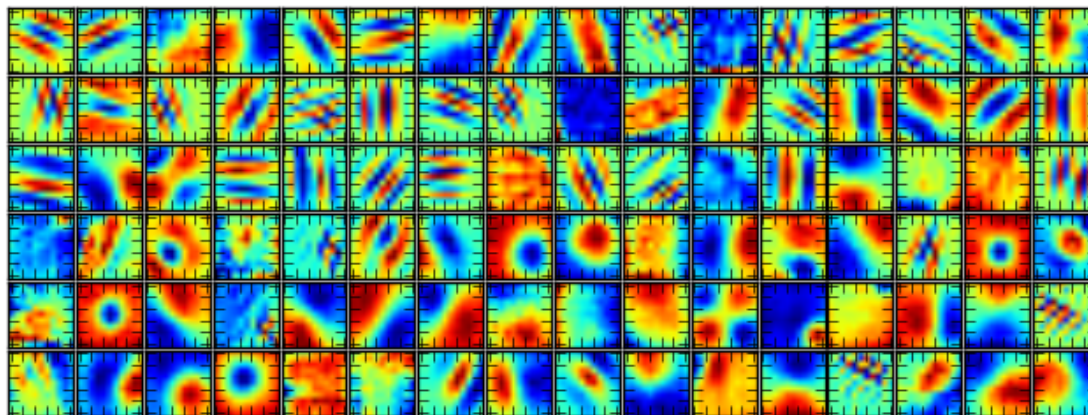


# カーネルと特徴マップの例

24

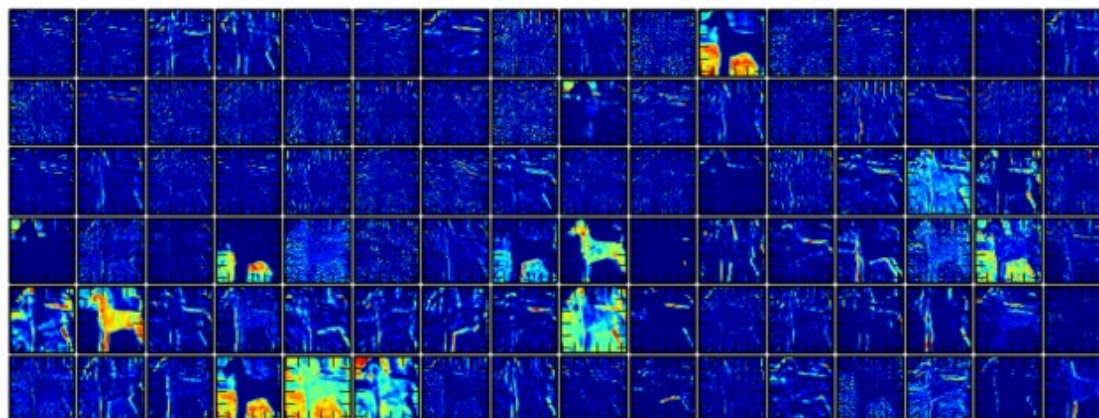
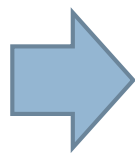


【AlexNetのconv1層のRチャンネルに対する96種類のカーネル(11 x 11)の例】



【AlexNetのconv1層】  
カーネルサイズ: 11x11  
ストライド: 4  
パディング: 0

【AlexNetのconv1層の96種類の特徴マップ(55 x 55)の例】

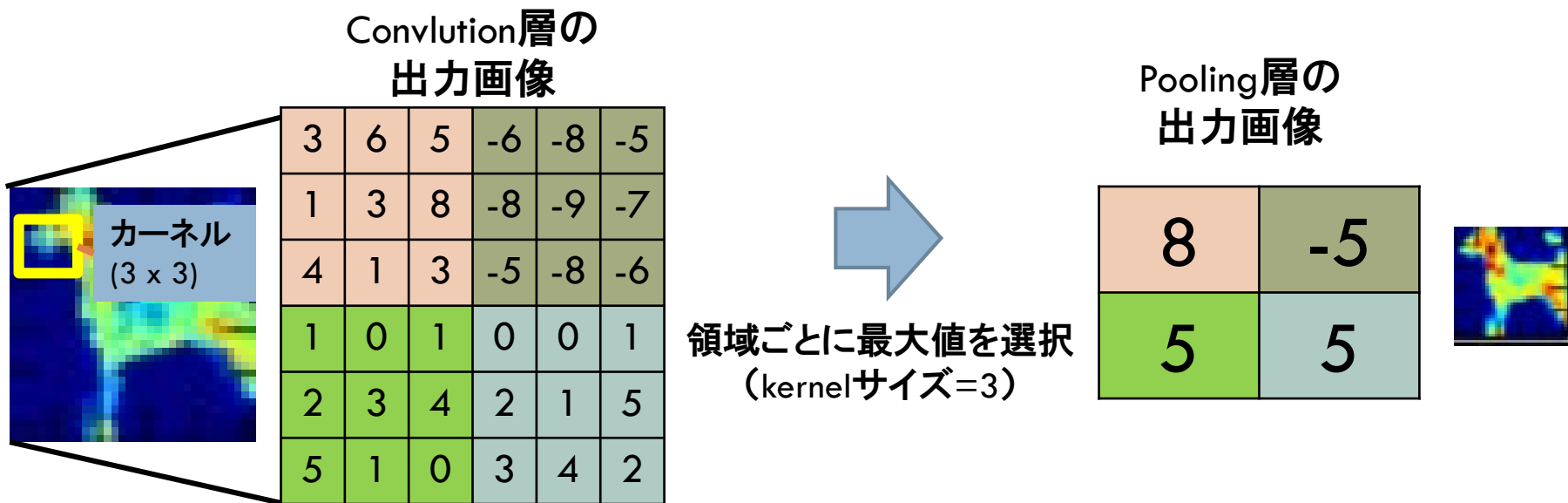


- 1層目のconvolutionでは、多様なエッジフィルタが学習

# Poolingの処理

25

- さらに、poolingの処理により出力画像の解像度を下げる



※Conv層と同様に、pooling層もカーネル、ストライド、パディングのサイズを設定する。

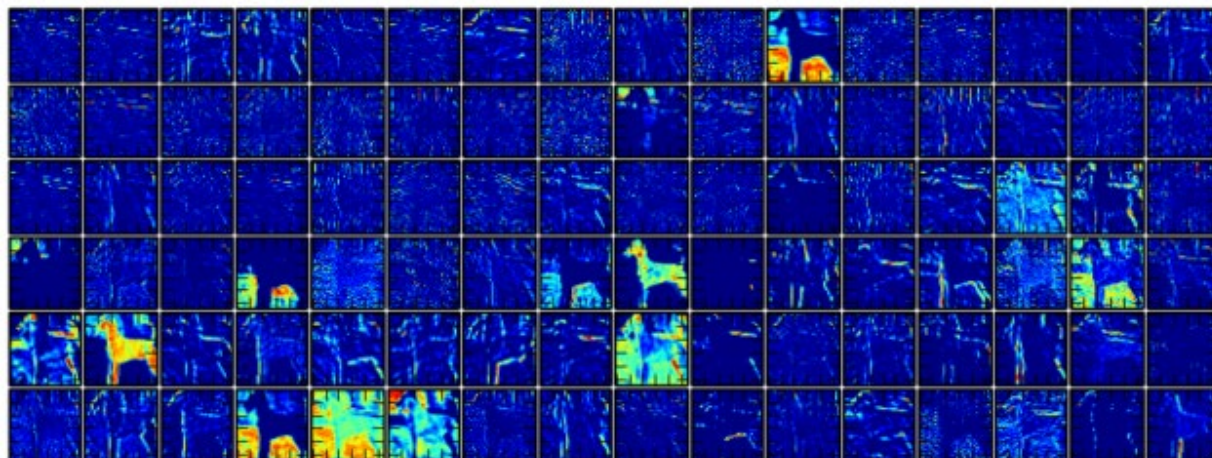
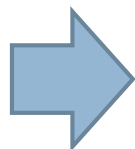
# Max poolingの例

26

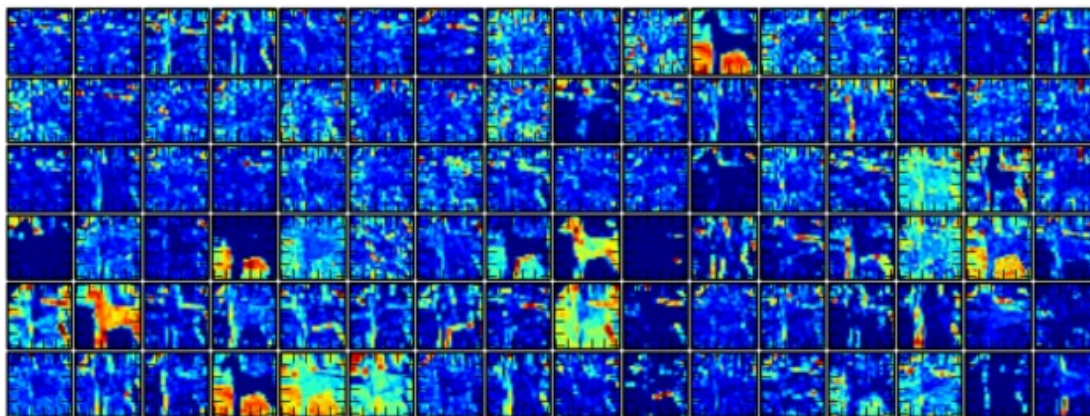
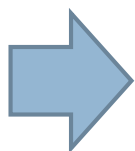
【AlexNetのconv1層の96種類の特徴マップ(55 x 55)の例】



(227 x 227)



【AlexNetのpool1層の特徴マップ(27 x 27)の例】



【AlexNetのpool1層】  
カーネルサイズ: 3x3  
ストライド: 2  
パディング: 0

□ フィルタの特性を維持したまま解像度を圧縮

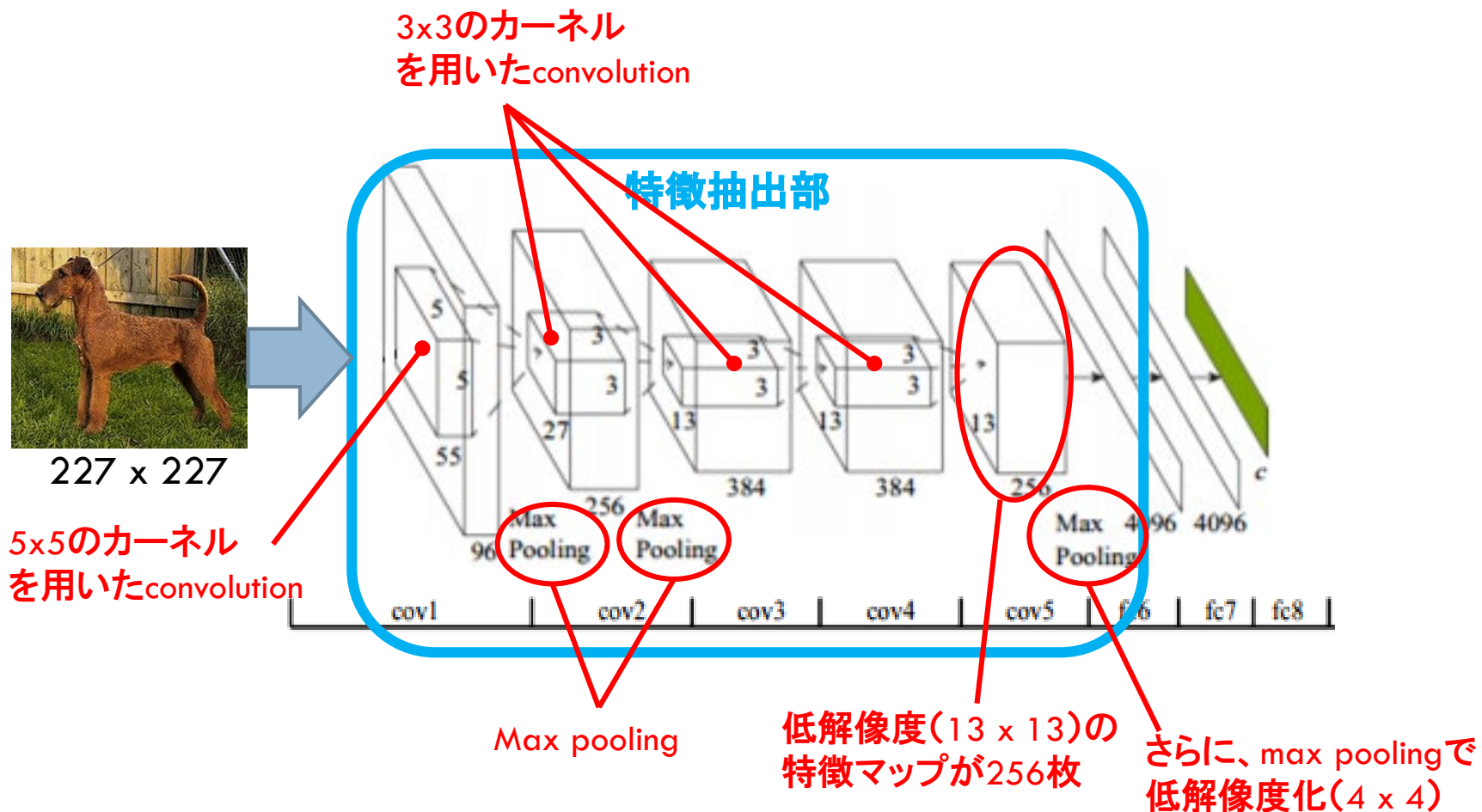
(55x55⇒27x27)



# Deep Neural Netモデルの構成

27

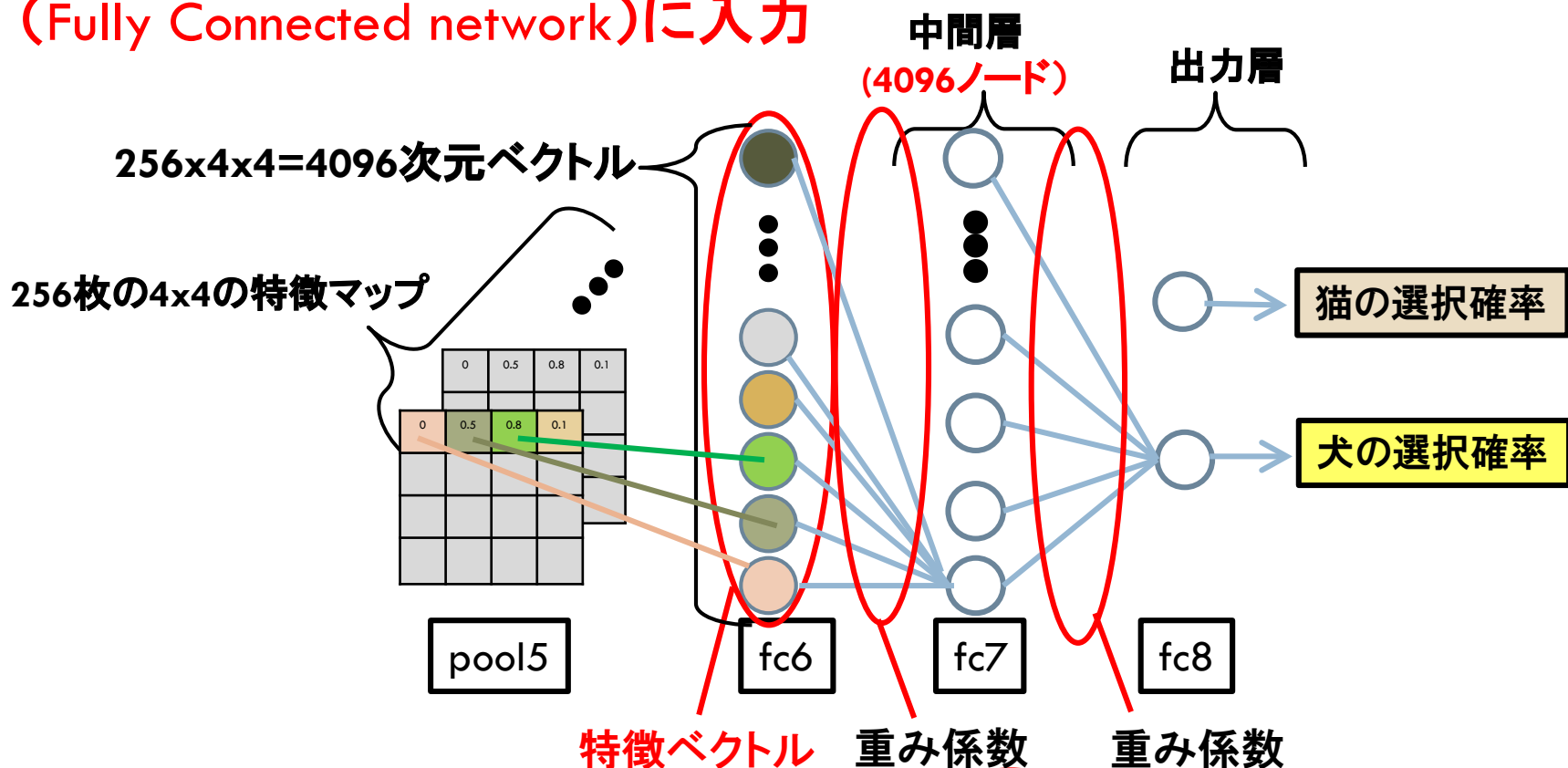
【大規模画像分類ImageNet2012で優勝したAlexNet(8層)のモデル】



# 分類部：全結合ネットワーク

28

- 特徴マップをベクトル化し、全結合ネットワーク (Fully Connected network) に入力



- 単純な線形変換の繰り返し

$X$

●  
内積

重み係数

$W_{f1}$

重み係数

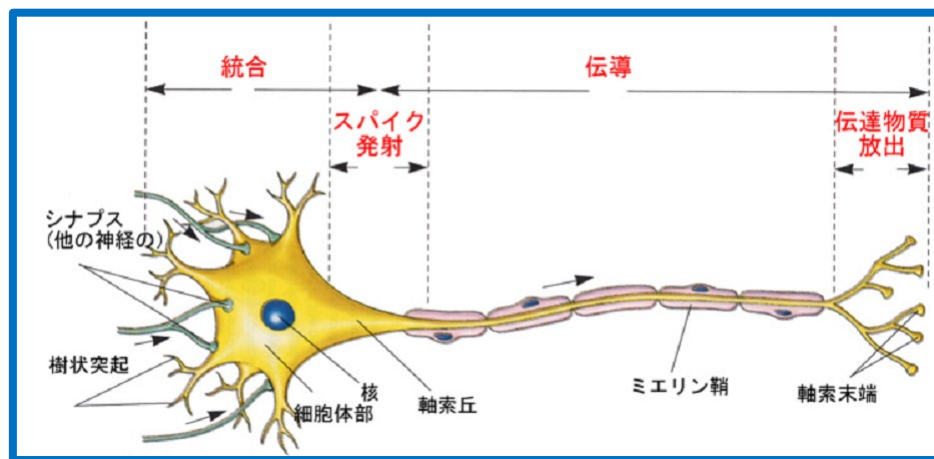
これらの値を分類精度が改善するように決定

# 活性化関数を用いた情報選択

29

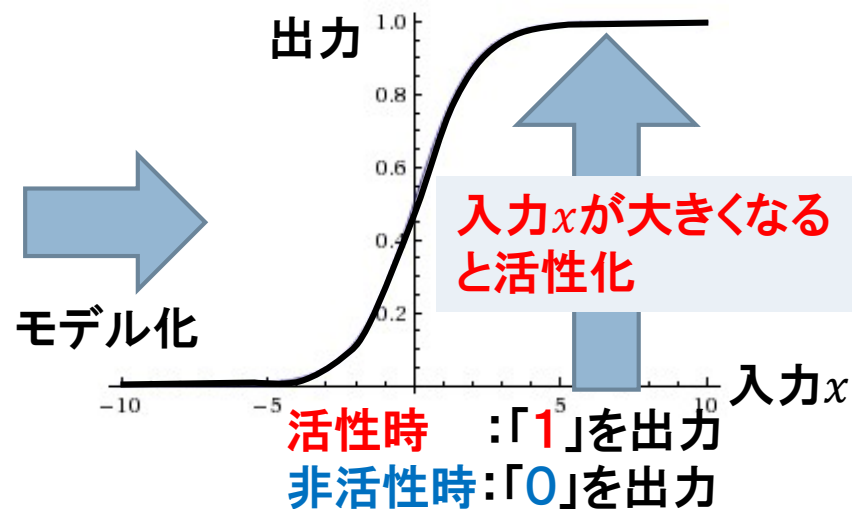
## □ 脳の神経伝達: シナプス結合

- 別の神経細胞からシナプスを通して受け取った神経伝達物質の総和が、閾値を超えた時に活性化し、スパイクという電気を流す



<http://kitsuo-n-kaizen.en.que.jp/hori/108.htm>

【各ノードのシグモイド活性化関数】



## □ 各階層にて、特徴選択のスイッチとして用いられる

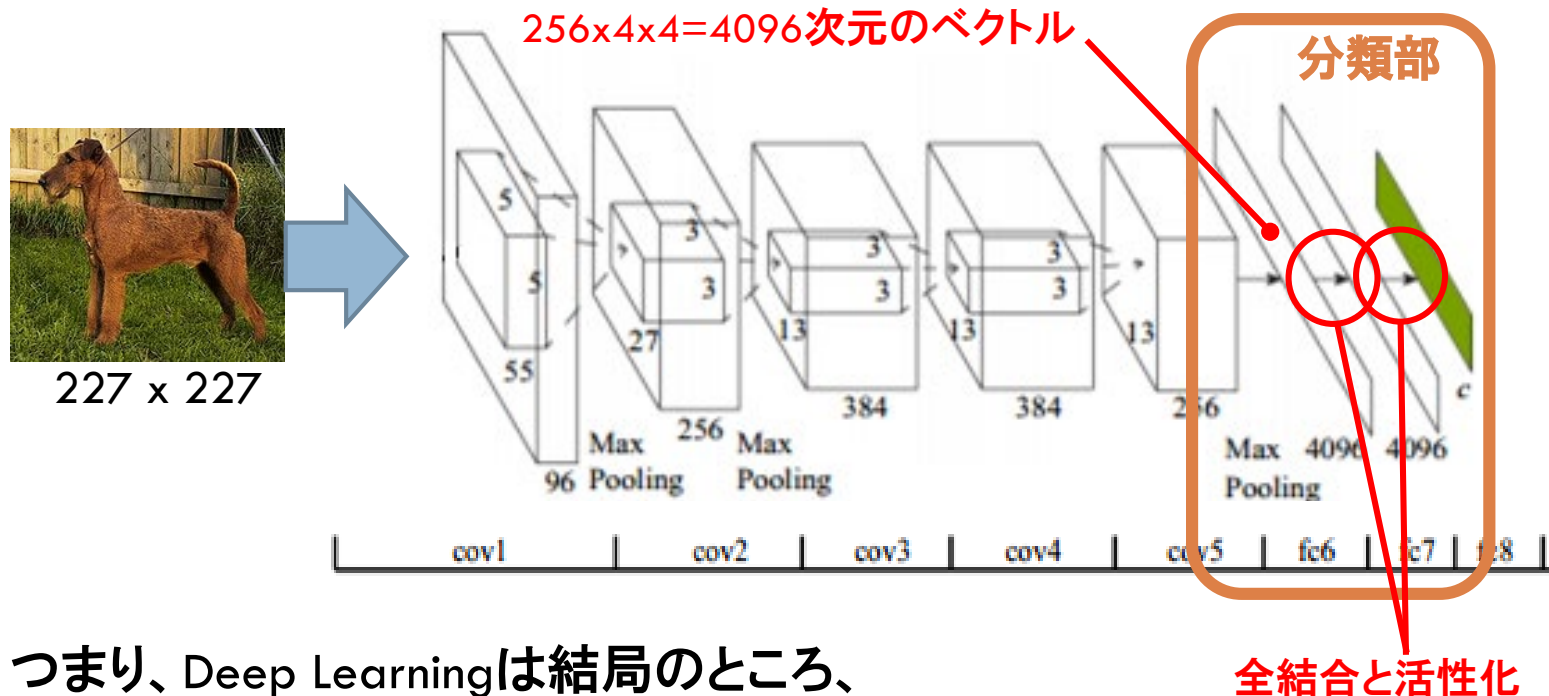
- 重み係数 $W$ と共起(ベクトルの向きが類似)する情報のみが伝搬



# Deep Neural Netモデルの構成

30

【大規模画像分類ImageNet2012で優勝したAlexNet(8層)のモデル】



- つまり、Deep Learningは結局のところ、
  - 線形変換(共起性の判定)
  - 活性化(情報選択)を繰り返しているだけ

# AlexNetの実行結果の例

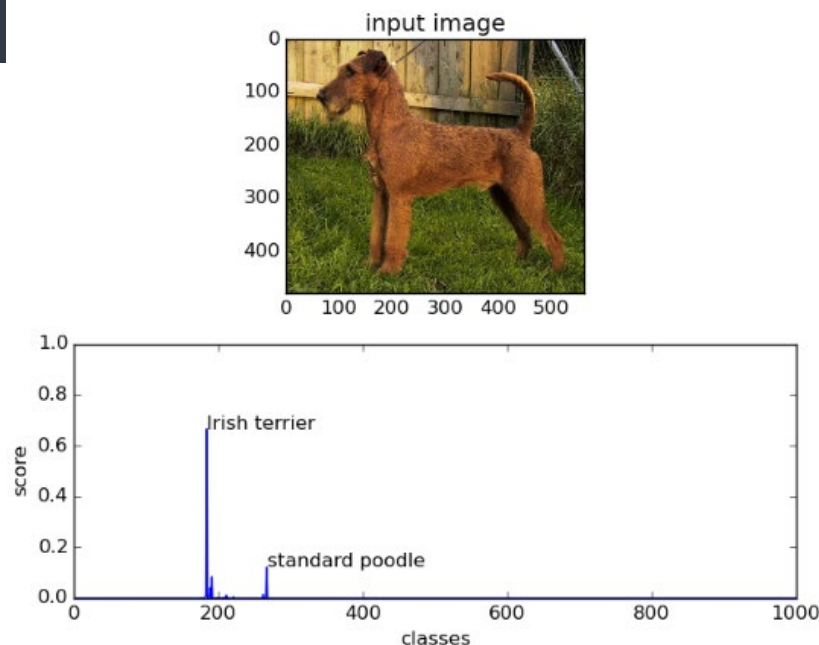
31

```
> python classify_caffenet.py images/irish_terror.jpg
```

	label	score	synset_id
0	Irish terrier	6.657821e-01	n02093991
1	standard poodle	1.216341e-01	n02113799
2	Airedale	8.359165e-02	n02096051
3	Lakeland terrier	4.101080e-02	n02095570
4	wire-haired fox terrier	1.805066e-02	n02095314
5	Brabancon griffon	1.581621e-02	n02112706
...			

67%の確信度で  
Irish terrierを正解

上位4位までがテリア



試してみたい方は、以下を参照：

<http://hirotaka-hachiya.hatenablog.com/entry/2015/03/20/212247>

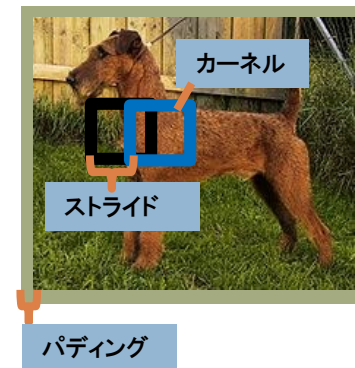
# CNNの階層数とパラメータ数

32

- Convとpool層が出力する特徴マップの大きさ

$$Osize = \frac{\lfloor Isize - Ksize + 2Psize \rfloor}{Ssize} + 1$$

Lname	レイヤー名
Isize	入力サイズ
Csize	チャンネル数
Ksize	カーネルサイズ
Ssize	ストライドサイズ
Psize	パディングサイズ
Osize	出力サイズ
Nparam	パラメータ数

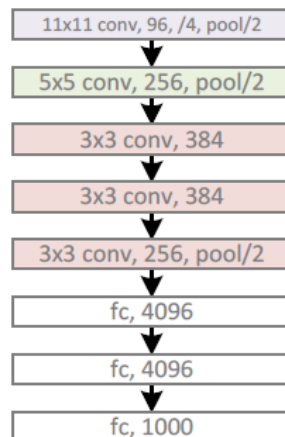


- 学習パラメータ数: conv層とfc層のみ

- 各Conv層:  $(Ksize^2 + 1) \times Csize_{before} \times Csize_{current}$

- 各FC層:  $Isize \times Osize$

【AlexNetの場合】



alex								
Lname	Isize	Csize	Ksize	Ssize	Psize	Osize	Nparam	
input		3				227		
conv 1	227	96	11	4	0	55	35136	
pool 1	55	96	3	2	0	27		
conv 2						27	638976	
pool 2						13		
conv 3						13	983040	
conv 4						13	1474560	
conv 5						13	983040	
pool 5	13	256	3	2	0	6		
fc 6	9216					4096	37748736	
fc 7	4096					4096	16777216	
fc 8	4096					1000	4096000	
total							62736704	

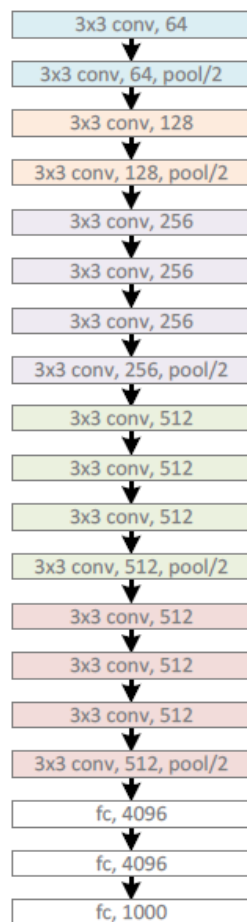
ただし、パラメータ数は、  
FC層が支配的

約6千万

# CNNの階層数とパラメータ数 2

33

## □ ILSVRC2014で優勝したVGG(19層)の場合



vgg-19							
Lname	Isize	Osize	Ksize	Ssize	Psize	Osize	Nparam
input		3				224	
conv1_1	224	64	3	1	1	224	1920
conv1_2	224	64	3	1	1	224	40960
pool1	224	64	2	2	0	112	
conv2_1	112	128	3	1	1	112	81920
conv2_2	112	128	3	1	1	112	163840
pool2	112	128	2	2	0	56	
conv3_1	56	256	3	1	1	56	327680
conv3_2	56	256	3	1	1	56	655360
conv3_3	56	256	3	1	1	56	655360
conv3_4	56	256	3	1	1	56	655360
pool3	56	256	2	2	0	28	
conv4_1	28	512	3	1	1	28	1310720
conv4_2	28	512	3	1	1	28	2621440
conv4_3	28	512	3	1	1	28	2621440
conv4_4	28	512	3	1	1	28	2621440
pool4	28	512	2	2	0	14	
conv5_1	14	512	3	1	1	14	2621440
conv5_2	14	512	3	1	1	14	2621440
conv5_3	14	512	3	1	1	14	2621440
conv5_4	14	512	3	1	1	14	2621440
pool5	14	512	2	2	0	7	
fc6	25088					4096	102760448
fc7	4096					4096	16777216
fc8	4096					1000	4096000
total							143294464

ただし、パラメータ数は、  
FC層が支配的

約1.4億

# CNNの多階層化と性能の推移

34

## Revolution of Depth

AlexNet, 8 layers  
(ILSVRC 2012)



VGG, 19 layers  
(ILSVRC 2014)



ResNet, 152 layers  
(ILSVRC 2015)



- 単に深くしただけ？ ⇒ もちろん違う！
- 深くしても学習ができるように様々な拡張



# 内容

35

- 機械学習とは
- 機械学習が急速に広まった理由
- 教師あり学習の概要
- 深層学習(Deep Learning)を用いた特徴抽出
- より発展的な機械学習
  - ▣ 異常検知の概要と応用例
  - ▣ 強化学習の概要と応用例

# 教師データの準備が困難な事例

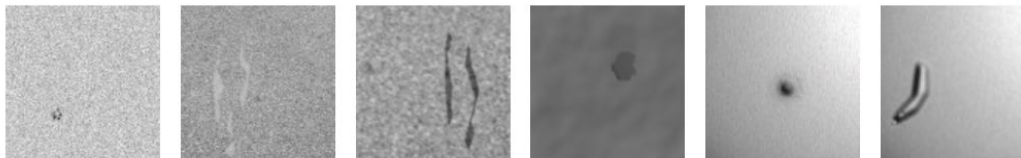
36

- 監視カメラ画像からの異常行動検知：
  - ▣ 正常行動か、異常行動(転倒、ひったくり、万引き)かの分類
  - ▣ 日本のような安全な国では、異常行動時の画像は入手困難



UCSD pedestrian dataset

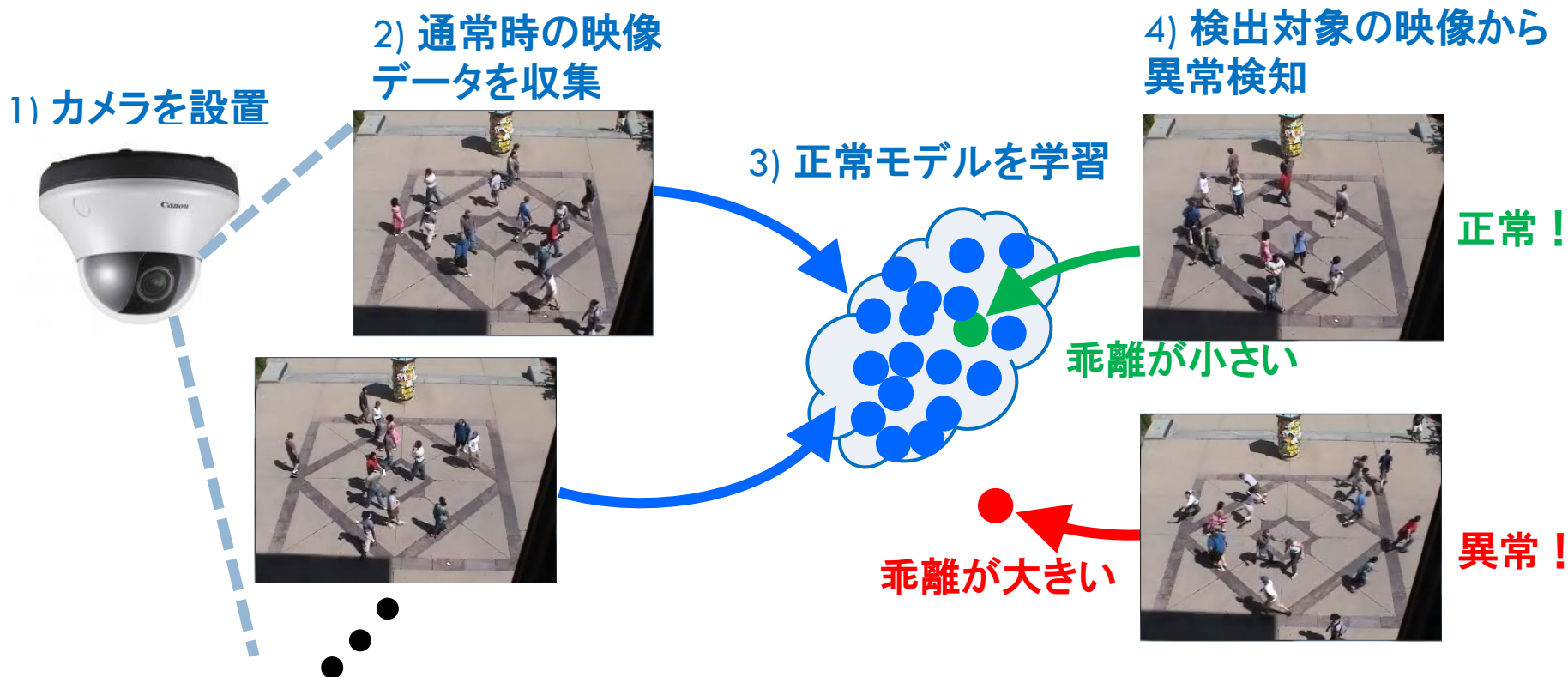
- 製造ラインでの画像・センサを用いた検査
  - ▣ OK品か、NG品(キズ、汚れ、異物混入、破損)の分類
  - ▣ NGの種類は多いが、NGの発生頻度はOKと比べると圧倒的に低い



# 正常モデルを用いた異常検知

37

## □ 正常データを表すモデルからの乖離度合いに基づき異常検知

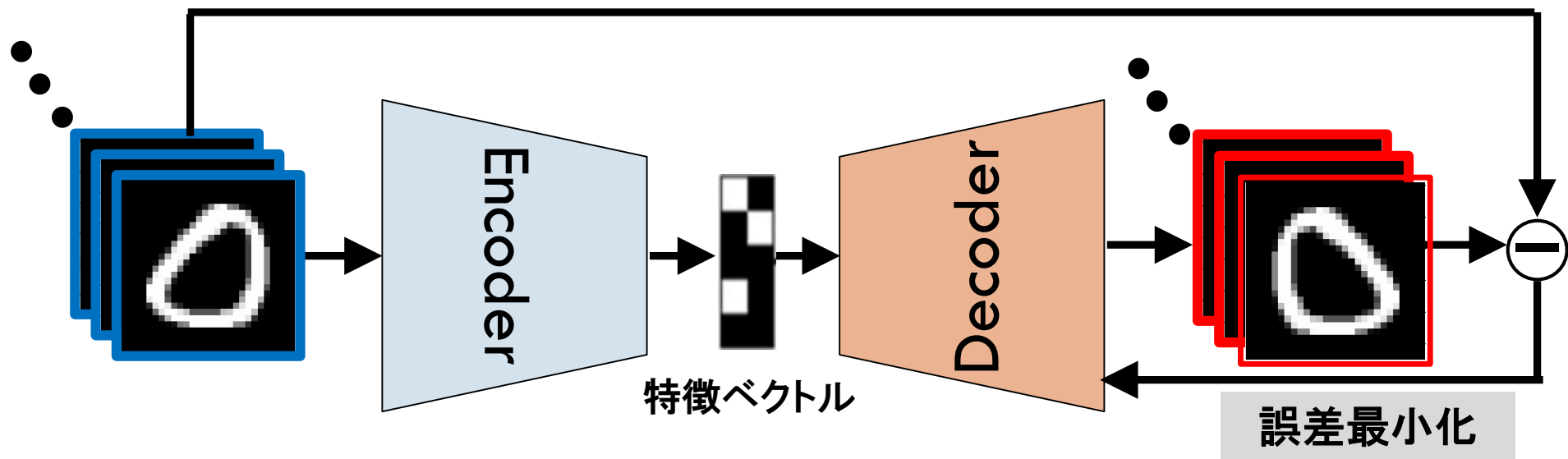


## □ 利点：容易に収集できる正常データのみから異常を検知できる

# Encoder-Decoderによる正常モデル

38

※ここからは、手書き数字「0」を正常とする

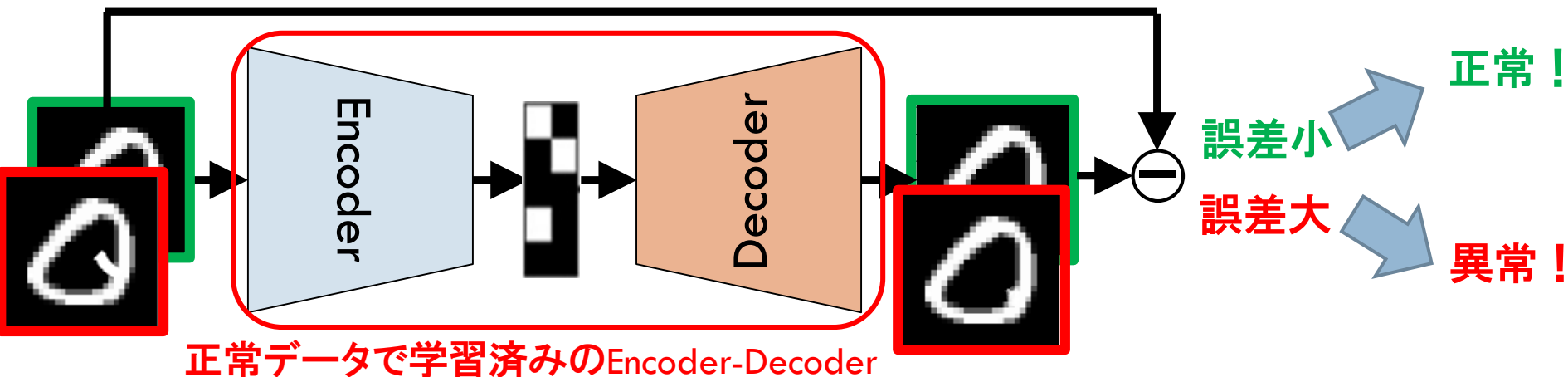


- Encoder: convolutionを繰り返し画像を特徴ベクトルに圧縮
- Decoder: 逆convolutionを繰り返し特徴ベクトルを画像に復元
- 容易に収集可能な大量の正常データを用いて、正常データを精度よく復元できるように学習

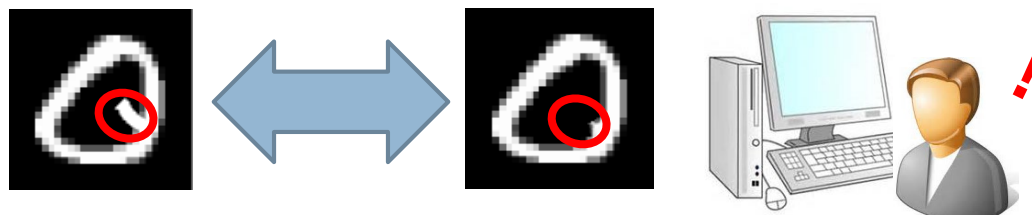
# Encoder-decoderによる異常検知

39

- 乖離度合いとして、再現誤差を用いた異常検知



- 入出力の差分に基づき、異常発生力所を特定しユーザに提示



# X線の手荷物検査

40

- 人手で行っている空港のX線荷物検査を半自動化したい
  - コンピュータでスクリーニングし警報があがったものを人が確認



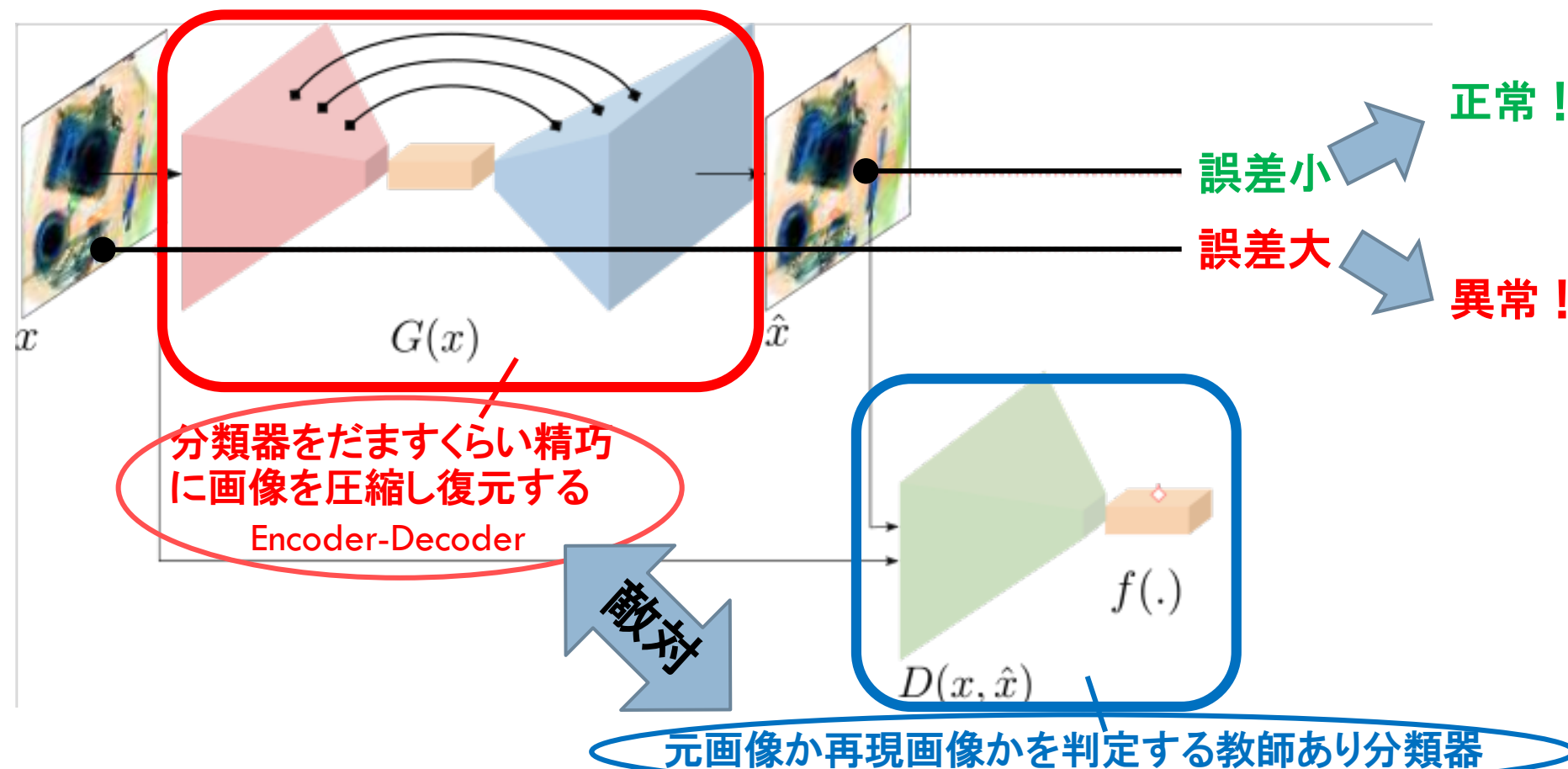
- 正常のX線画像データは大量にあるが、異常データは少ない
  - 異常のパターンが多様(銃やナイフなどの形・サイズは多様)
- イギリス政府が提供するFFOB(Full Firearm vs Operational Benign)データでは、67,672枚の正常データがある

# Encoder-Decoderを用いた異常物体検知

S. Akcay, et al., groundAI, 2019

41

- encoder-decoderと、教師あり分類を組み合わせたネットワーク
  - ▣ Generative Adversarial Network(GAN)と呼ばれている





# 異常物体検知の結果

UBA: University Baggage Anomaly dataset  
大学生の鞆のX線画像

42

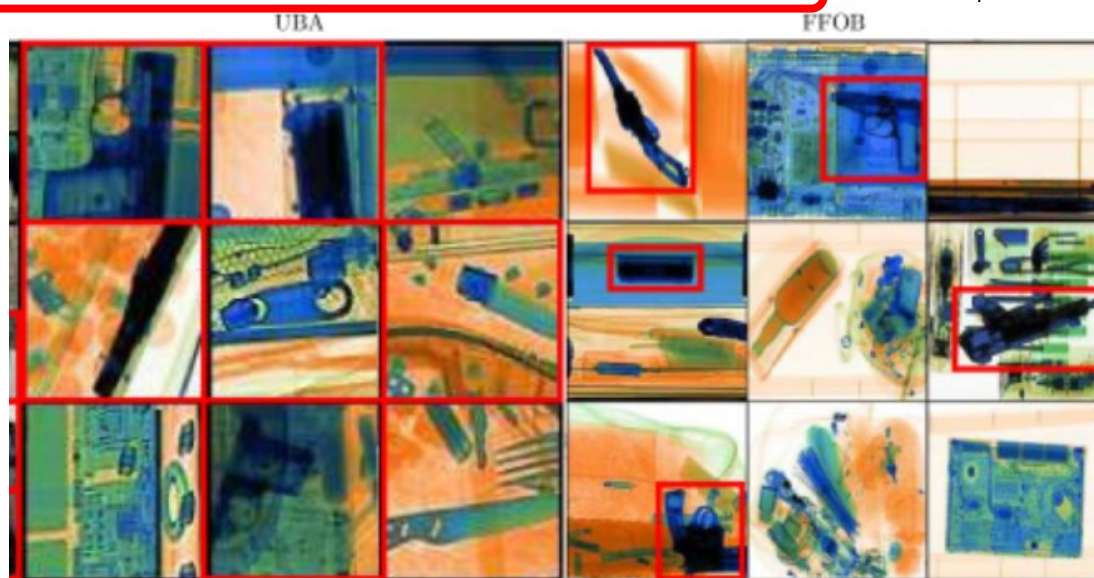
- 銃、銃の部品、ナイフなどの凶器を高い精度で検出

Method	UBA				FFOB
	gun	gun-parts	knife	overall	full-weapon
AnoGAN [9]	0.598	0.511	0.599	0.569	0.703
EGBAD [12]	0.614	0.591	0.587	0.597	0.712
GANomaly [13]	0.747	0.662	0.520	0.643	0.882
<b>Proposed</b>	<b>0.972</b>	<b>0.945</b>	<b>0.904</b>	<b>0.940</b>	<b>0.903</b>

評価指標:

AUC(Area Under Curve)

- 検出結果の例:





# 内容

43

- 機械学習とは
- 機械学習が急速に広まった理由
- 教師あり学習の概要
- 深層学習(Deep Learning)を用いた特徴抽出
- より発展的な機械学習
  - ▣ 異常検知の概要と応用例
  - ▣ 強化学習の概要と応用例

# 機械学習の種類

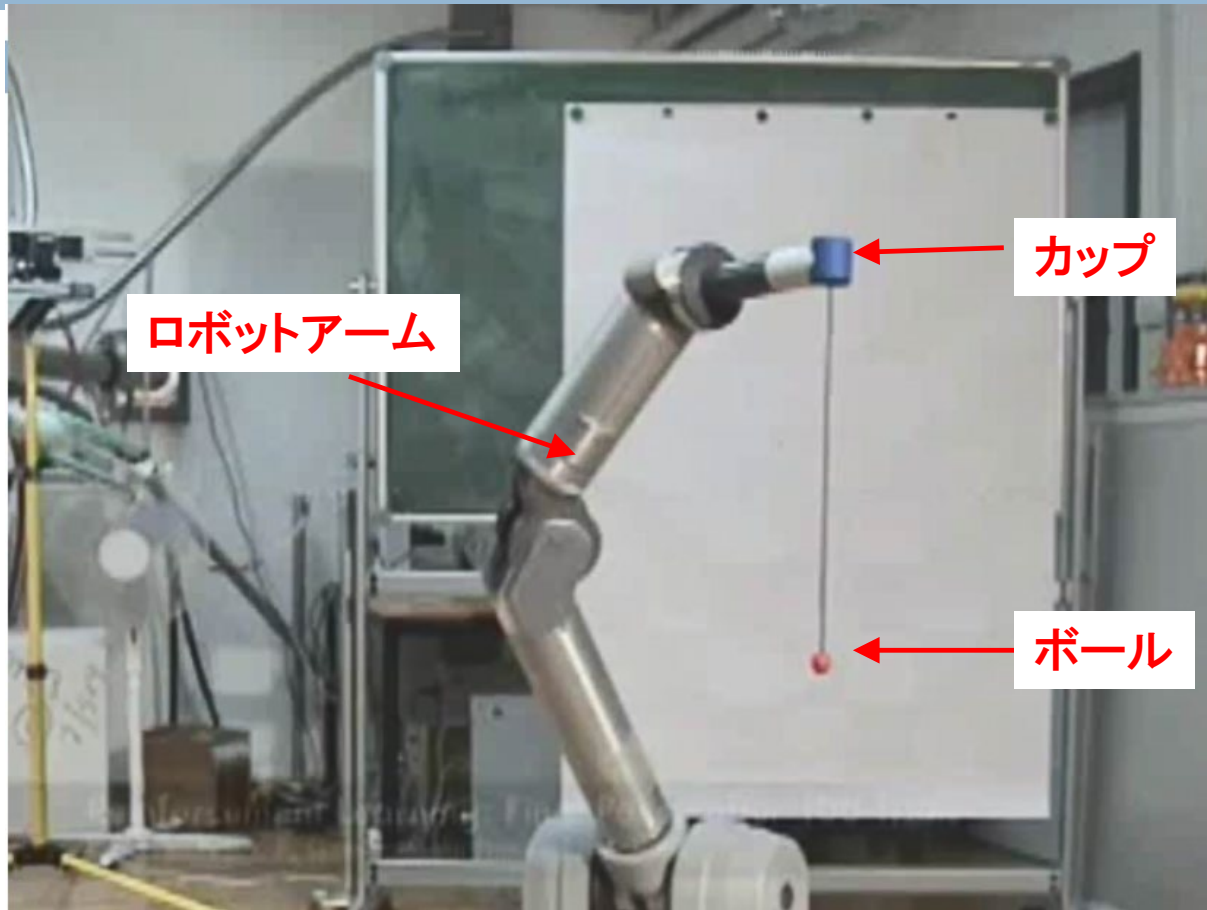
44

問題	定義	代表的な方法	応用例
教師あり学習	入力と出力のデータに基づき、入力を出力に変換する関数を学習	SVM, 最小二乗法、決定木、ランダムフォレストなど	顔認識、物体検出、手書き文字分類、株価予測、降水量予測など
教師なし学習	入力のみデータに基づき、入力の特性（パターン、構造）を学習	PCA, LDA, HMMなど	データの可視化（クラスタリング、次元圧縮）
半教師学習	入力のうち部分的に付与された出力の事例に基づき入力を出力に変換する関数を学習	transductive SVM, Laplacian SVMなど	画像、音声、Webログなどの大量データで、コストの問題で一部のデータのみしか出力（答え）が付与されていない場合
強化学習	入力と、出力に対する報酬（評価）のデータに基づき、入力を出力に変換する関数を学習	Q-learning、policy iteration, policy gradient	ロボット制御、Web広告選択、マーケティング

- 準備可能なデータと目的に合わせて適切な機械学習方法を選択する必要

# 制御の例: Ball in a cup(けん玉)

45



- **目標**: ロボットがボールをカップの中に入れる制御を獲得

# 教師データの収集

ドイツ・マックスプランク研究所

46



- 入力(状態)と出力(行動)の組のデータ:  $\{(x^t, y^t)\}_{t=0}^N$  を収集
- 状態  $x$  : 各関節の角度・角速度、 行動  $y$  : 各関節に加えるトルク量

# 教師あり学習の問題点

47

- 学習に十分な教師データを収集するのは困難
  - ▣ 教師データ数は最低でも学習パラメータの十数倍は必要
  - ▣ 教師データを用意するのは人なので、時間および人件費が大
- 教師データが不正確(特にロボットでは)
  - ▣ 人間とロボットでは、骨格、筋肉配置が異なるため、ロボットにとって最適なダイナミックな行動(関節のトルク)を教えるのが困難
- 人間の知識の限界
  - ▣ 人間が必ずしも最適な方法を知っているとは限らない

学習した関数:

良くて人間の模倣、コンピュータにとって最適とは限らない

# 教師あり学習の結果の例

48



- 動作は似ているが、ボールをカップに入れることができない

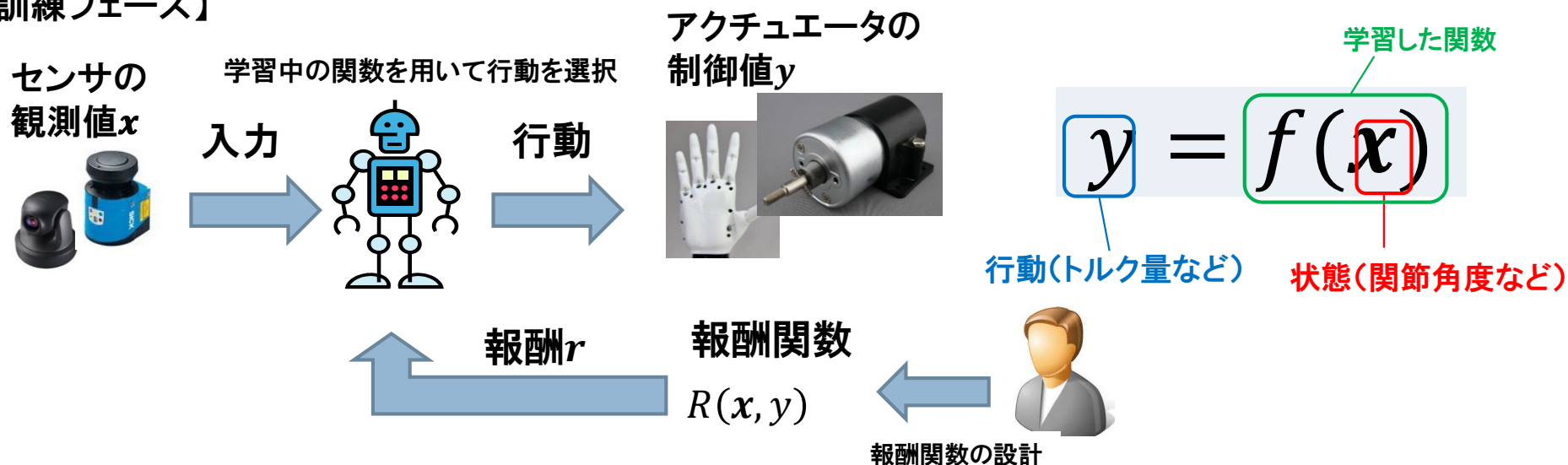
# 強化学習

$x$ : 入力(状態)、 $y$ : 出力(行動)

49

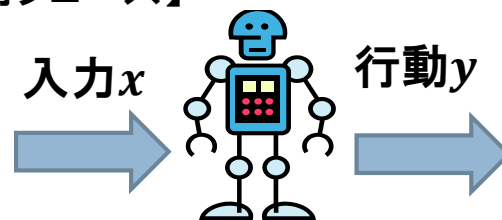
- 人間が予め設計した報酬関数を用意  $r = R(x, y)$
- **訓練**: 報酬和を最大化するように、行動を出力する関数を学習

## 【訓練フェーズ】



- **運用**: 入力された状態に行動を予測

## 【運用フェーズ】 学習した関数を用いて行動を選択



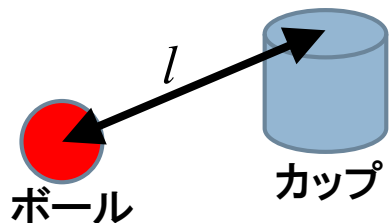


# 強化学習の結果の例

50

報酬の例:

$$r = \exp(-l)$$



$l$ : ボールとカップの距離



- 関数は報酬に基づき試行錯誤的に改善され、ボールをカップに入れることができるようになる。

# 強化学習の成功例

51

- 将棋: 2013年の電王戦(現役プロとコンピュータとの対局)にてコンピュータが3勝1敗1分けて勝ち越し



- 囲碁: 2016年3月、Google DeepMind社の「Alpha Go」が世界No.2の韓国の李セドル氏に4勝1敗の大差で勝利



# 機械学習同士での対戦で進化

52

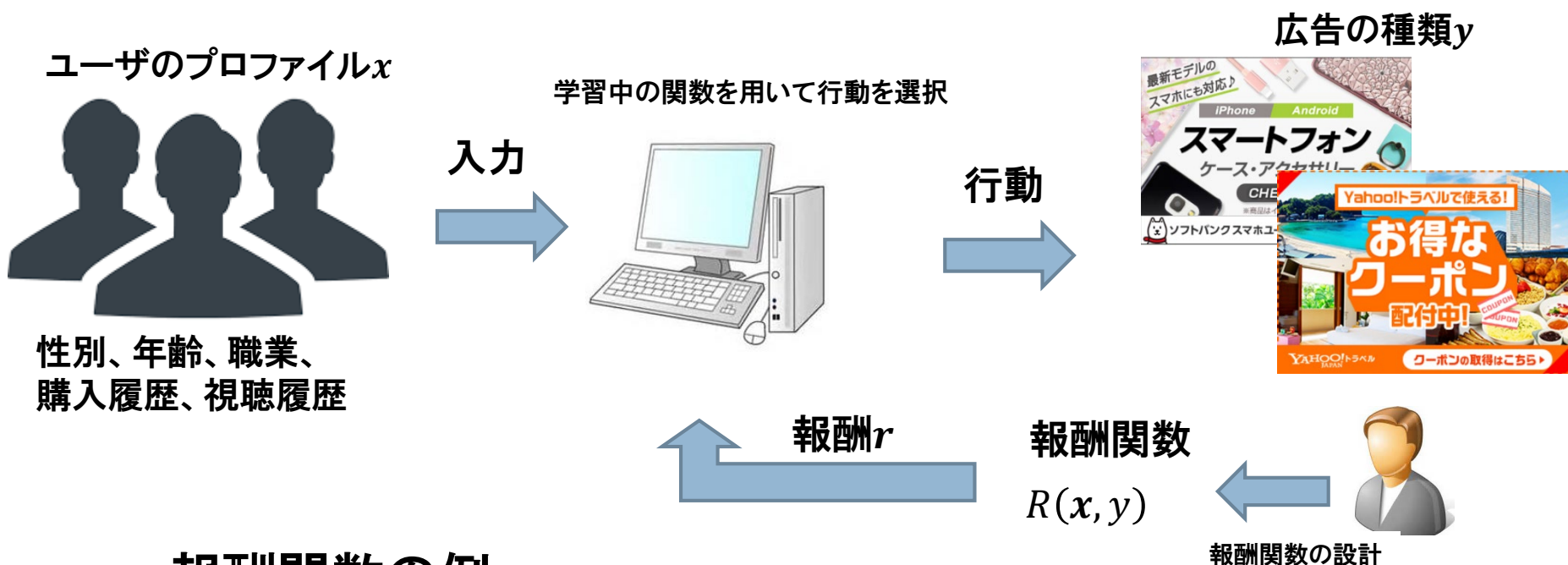
- 強化学習アルゴリズム同士の対戦を繰り返すことにより、人間が数千年かけても見つけられなかった「定石」を数日で発見



# 強化学習の例：広告選択

53

## □ 広告選択における状態(入力) $x$ 、行動(出力) $y$ の例



## □ 報酬関数の例

$$R(x, y) = \begin{cases} 1 & \text{もしユーザ } x \text{ が広告 } y \text{ をクリックした場合} \\ 0 & \text{クリックしない場合} \end{cases}$$

# まとめ

54

- 機械学習とは
- 機械学習が急速に広まった理由
- 教師あり学習の概要 演習にて、Pythonコードを用いて実装
- 深層学習(Deep Learning)を用いた特徴抽出
- より発展的な機械学習
  - ▣ 異常検知の概要と応用例
  - ▣ 強化学習の概要と応用例