

情報・システム工学概論 画像・映像認識のモデル化

知能機械情報学専攻
機械情報工学科（機械B）
原田達也

Results (2012)

<http://www.isi.imi.i.u-tokyo.ac.jp/pattern/ilsvrc2012/index.html>



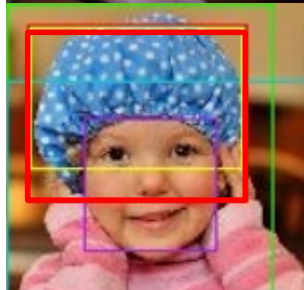
1. brown bear
2. Tibetan mastiff
3. sloth bear
4. American black bear
5. bison



1. baseball player
2. unicycle
3. racket
4. rugby ball
5. basketball



1. digital watch
2. Band Aid
3. syringe
4. slide rule
5. rubber eraser



1. shower cap
2. bonnet
3. bath towel
4. bathing cap
5. ping-pong ball



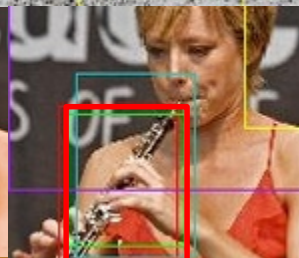
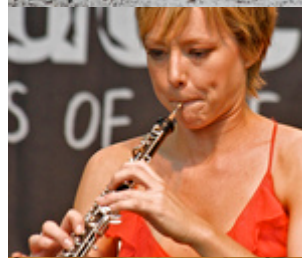
1. diaper
2. swimming trunks
3. bikini
4. miniskirt
5. cello



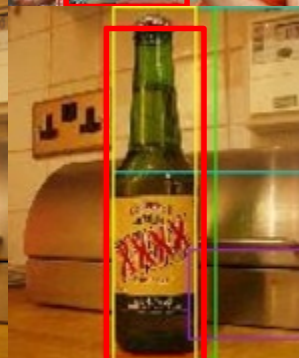
1. Siamese cat
2. Egyptian cat
3. Ibizan hound
4. balance beam
5. basenji



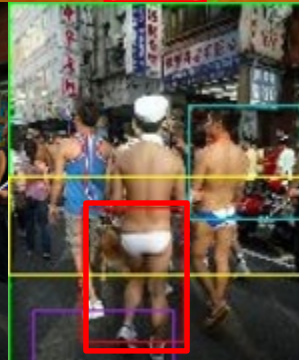
1. king penguin
2. sea lion
3. drake
4. magpie
5. oystercatcher



1. oboe
2. flute
3. ice lolly
4. bassoon
5. cello



1. beer bottle
2. pop bottle
3. wine bottle
4. Polaroid camera
5. microwave



1. butcher shop
2. swimming trunks
3. miniskirt
4. barbell
5. feather boa

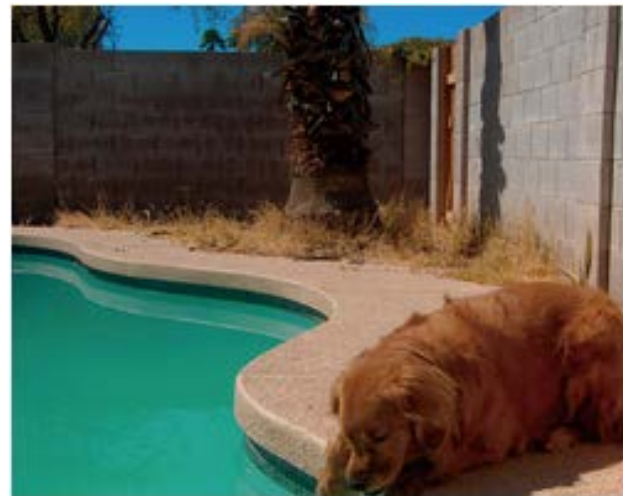
Sentence Generation Results



A giraffe standing next to a tree in a fence.



A yellow train on the tracks near a train station.



A dog laying on the side of a zoo enclosure.



With a cat laying on top of a laptop computer.

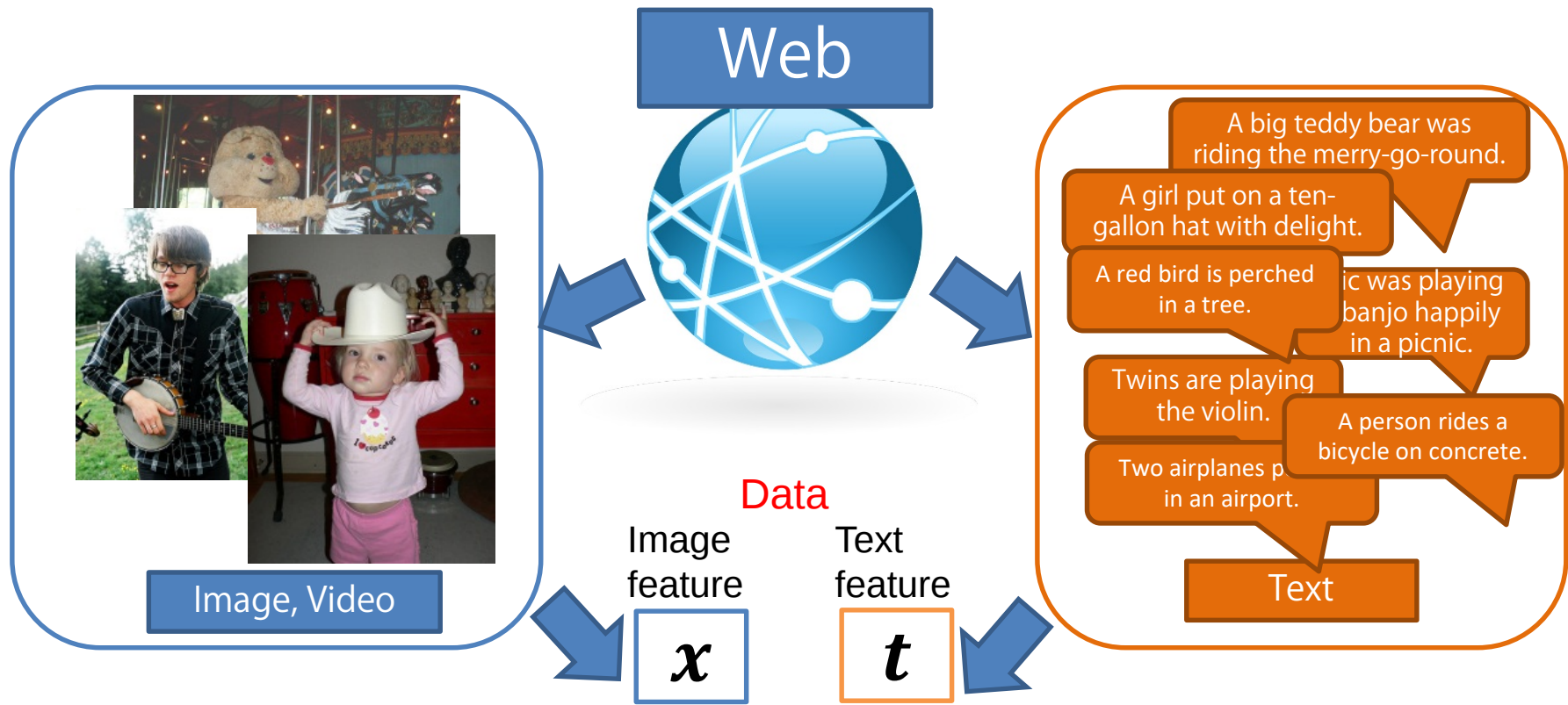


A man in the beach with a surfboard.



Black and white dog on the grass in a frisbee.

Overview: Machine Learning for Visual Recognition



Learning the relationships between images and text

Mapping function

Risk

$$R(\mathbf{w}) = \iint \underbrace{r(\Psi(\mathbf{x}, \mathbf{w})|\mathbf{t})}_{\text{Loss}} \underbrace{p(\mathbf{x}, \mathbf{t})}_{\text{Joint probability}} d\mathbf{x} d\mathbf{t}$$

Loss

Joint probability

Minimization



Ex) SGD

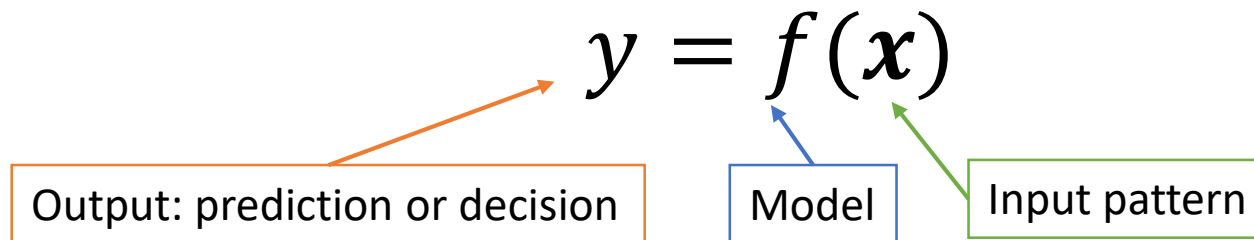
$$\mathbf{w}_{t+1} = \mathbf{w}_t - \epsilon \nabla_{\mathbf{w}} r(\Psi(\mathbf{x}, \mathbf{w}_t)|\mathbf{t})$$

機械学習とは

https://en.wikipedia.org/wiki/Machine_learning

- Machine learning is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence.
- In 1959, Arthur Samuel defined machine learning as a "Field of study that gives computers the ability to learn without being explicitly programmed".
- Machine learning explores the study and construction of algorithms that can learn from and make predictions on data.
- Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions expressed as outputs, rather than following strictly static program instructions.

分類と回帰

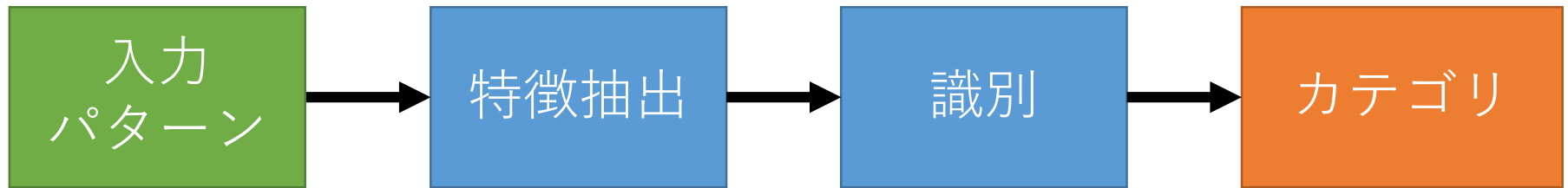


- 回帰 (regression)
 - 従属変数 y が連続変数である場合.
 - 独立変数 x が1次元の場合を単回帰, 2次元以上を重回帰という.
 - 例) 国民の所得から経済全体の消費を推定する.
- 分類, 識別 (classification)
 - 従属変数 y が有限個の離散カテゴリの場合.
 - 例) 画像を入力して, そこに写る動物の種類 (犬, 猫, 馬など) を推定する.

学習の分類

- 教師付き学習 (supervised learning)
 - パターンと正解ラベルの組からなるデータセットが与えられたときに、このデータセットから入出力の対応関係モデルを構成すること.
- 教師なし学習 (unsupervised learning)
 - ラベルデータのないパターンのデータセットから、パターンに内在する構造をとらえてモデル化すること.
- 半教師付き学習 (semi-supervised learning)
 - 正解ラベルありとなしの混在したデータセットから、入出力の対応関係モデルを構成すること.
- 強化学習 (reinforcement learning)
 - ある環境下で報酬が最も高くなる環境と行動との関係 (方策, policy) を学習する. 強化学習は、学習時に明示的な入出力のペアが与えられない点で教師付き学習とは異なる.

分類問題のパイプライン



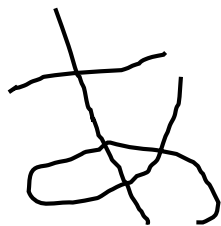
Input image

“Tiger”

- 分類問題の過程を特徴抽出と識別の2段階に大きく分けて考える.

特徴抽出と特徴ベクトル

- 特徴抽出 (feature extraction)
 - 多くの情報を持ったパターンから、認識に本質的な情報を抽出すること.
 - パターンを圧縮した効率のよい空間の点として表現する.
 - ノイズ除去などの前処理も含む.
- 特徴ベクトル (feature vector)
 - 例えば文字認識の場合、特徴量として、線の傾き、長さ、曲率などが考えられる.
 - それぞれの特徴を数値で表現し、それらを組としたベクトルが利用される.



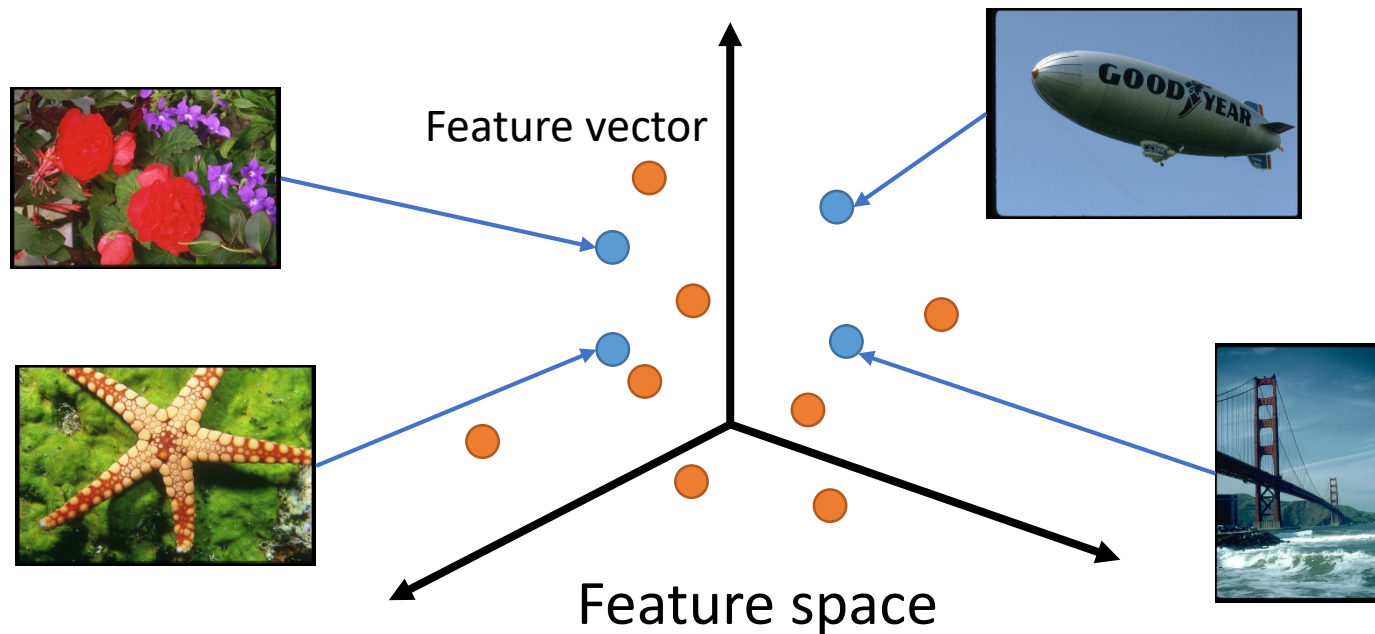
$$\mathbf{x} = (x_1, x_2, \dots, x_d)^T$$

線の傾き

長さ

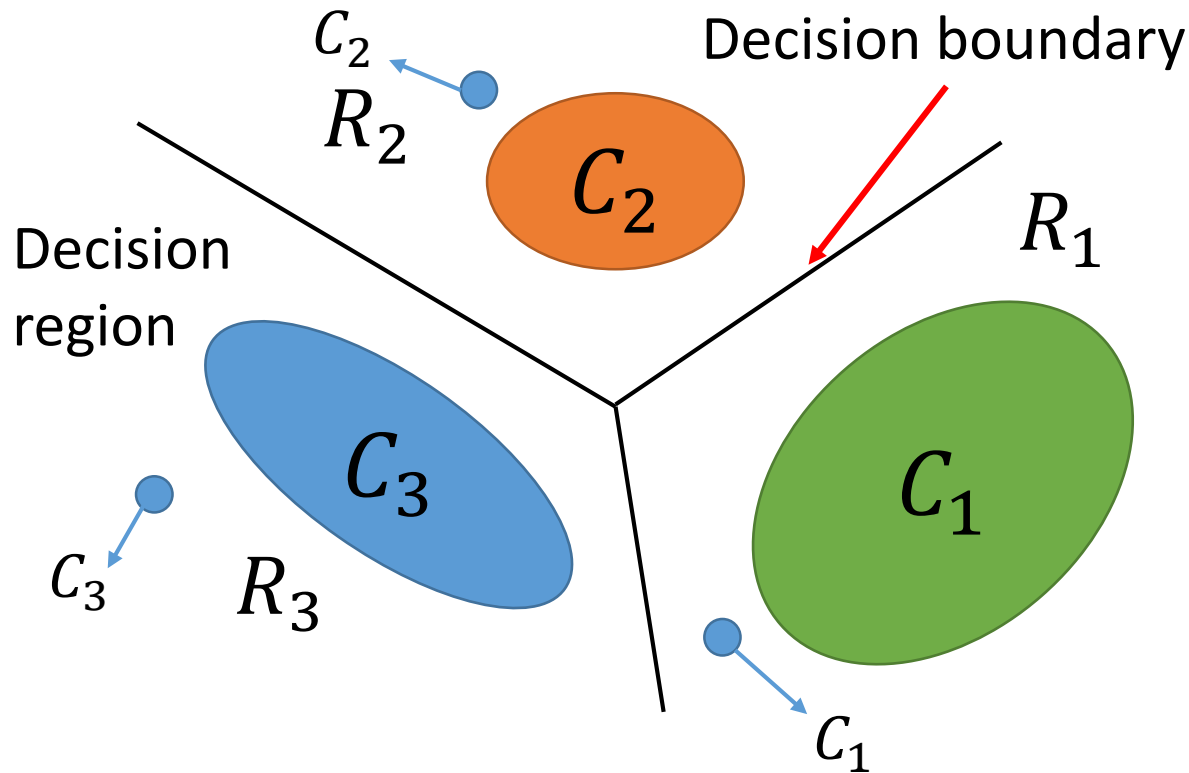
曲率

特徴空間



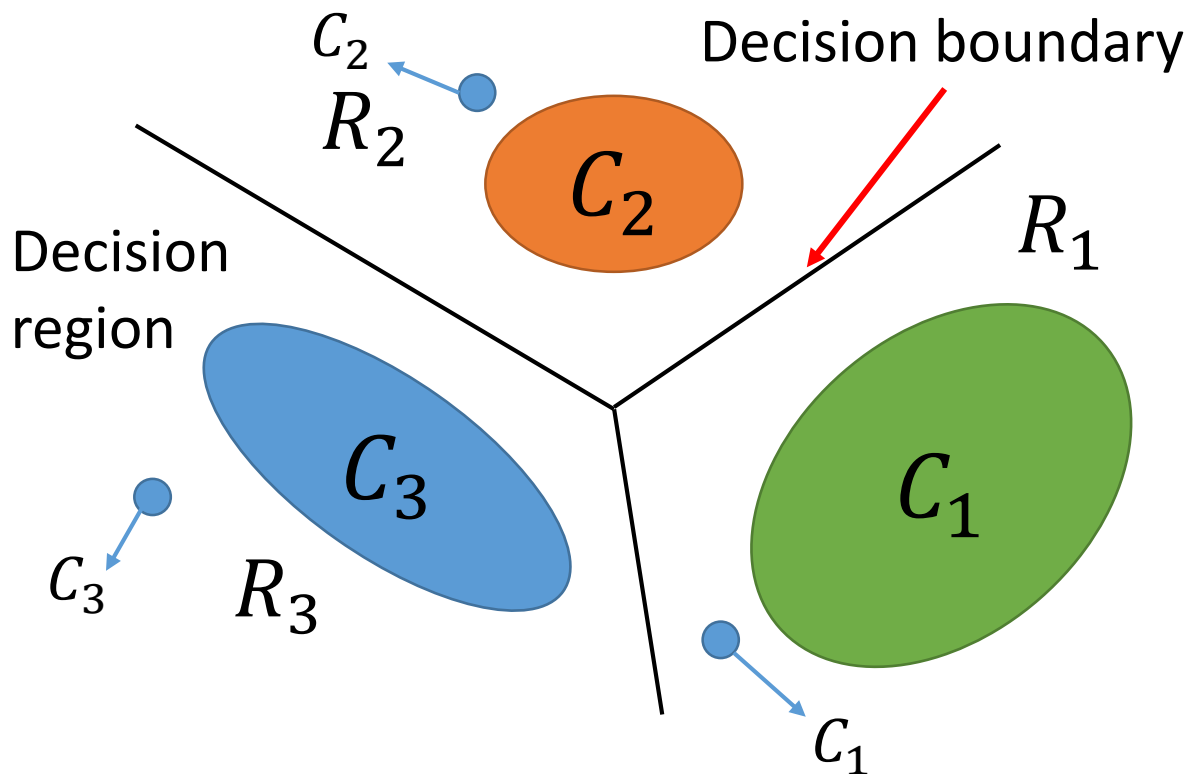
- 特徴ベクトルによって張られる空間を特徴空間 (feature space) という.
- 1つのパターンは d 次元の特徴空間の1点として表現される.

決定境界・決定領域



- 決定境界 (decision boundary)
 - クラス間を分離する境界
- 決定領域 (decision region)
 - クラストラベルの付与された分割された領域

識別器の設計

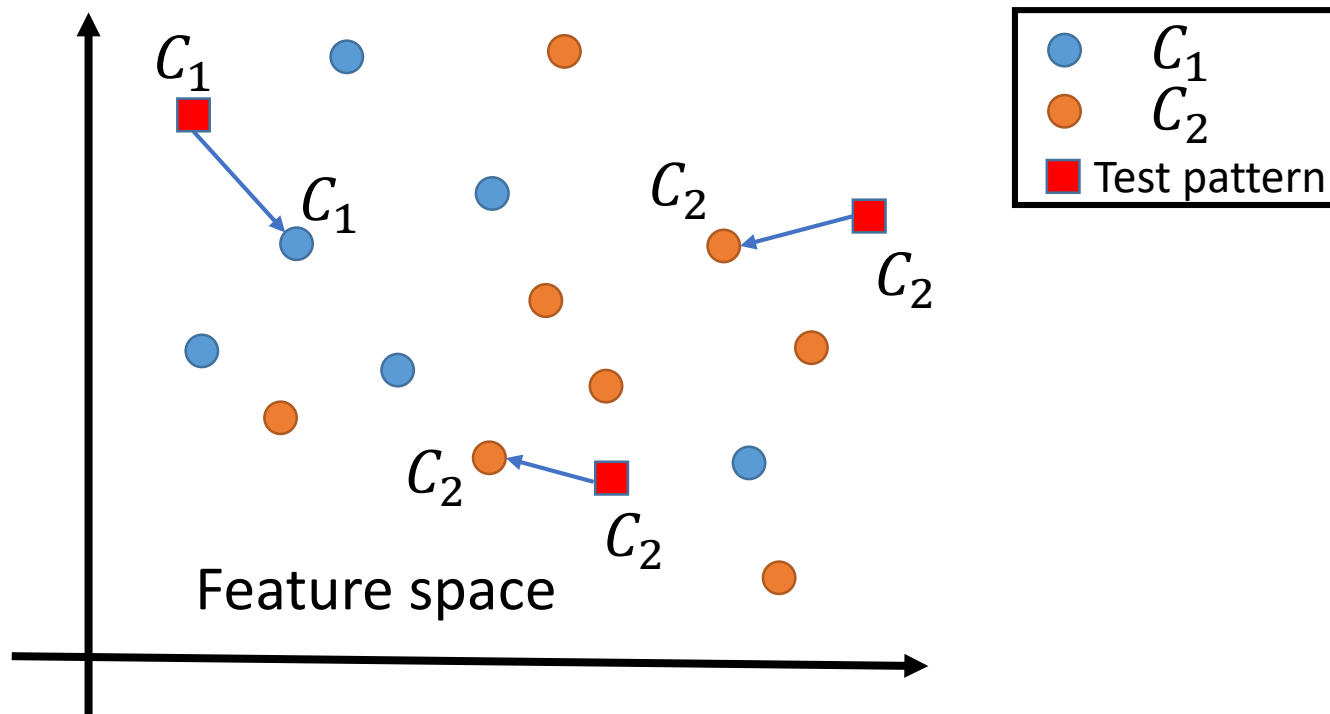


- 特徴空間の適切な領域分割の決定と領域に対するカテゴリ付与

学習データ，テストデータ，汎化

- 学習データ（training data）
 - 対応するクラスが与えられている少数のパターン集合から識別器を構成することを考える．このパターン集合を学習データ集合（training data）という．
 - 学習データから識別器を求める過程を学習（learning）という．
- テストデータ（test data）
 - 予測対象となるクラスが分かっていないパターン集合をテストデータ（test data）という．
- 汎化（generalization）
 - 学習データに含まれないパターンのクラスを予測する能力を汎化（generalization）という．

識別器の例：最近傍法



- 最近傍法 (nearest neighbor method)
 - 入力パターンと学習パターンとの距離を計算し、最も近いパターンが属するカテゴリを入力パターンのカテゴリと判断する手法.

最近傍法のアルゴリズム

$$\mathcal{X} = \{(\mathbf{x}_i, t_i)\}_{i=1}^N$$

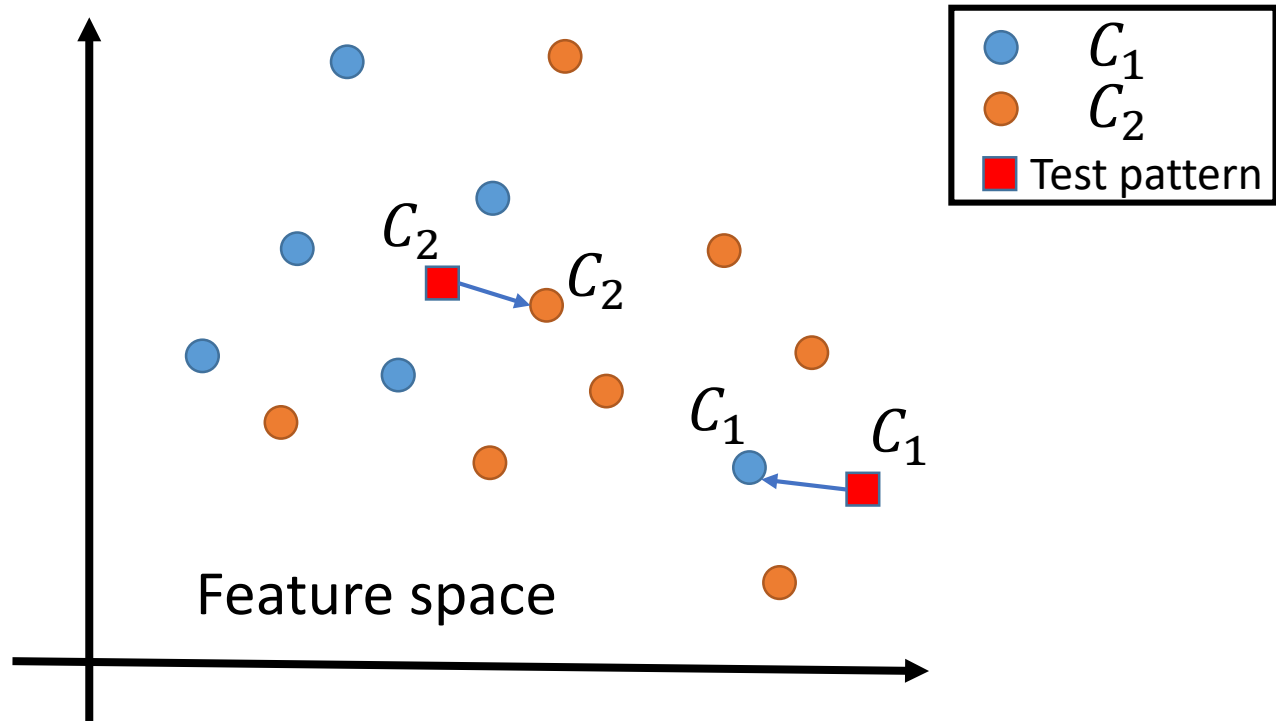
$$\mathbf{x}_j = \arg \min_{\mathbf{x}_i \in \mathcal{X}} \underline{d(\mathbf{z}, \mathbf{x}_i)} \Rightarrow \mathbf{z} \in t_j$$

距離関数

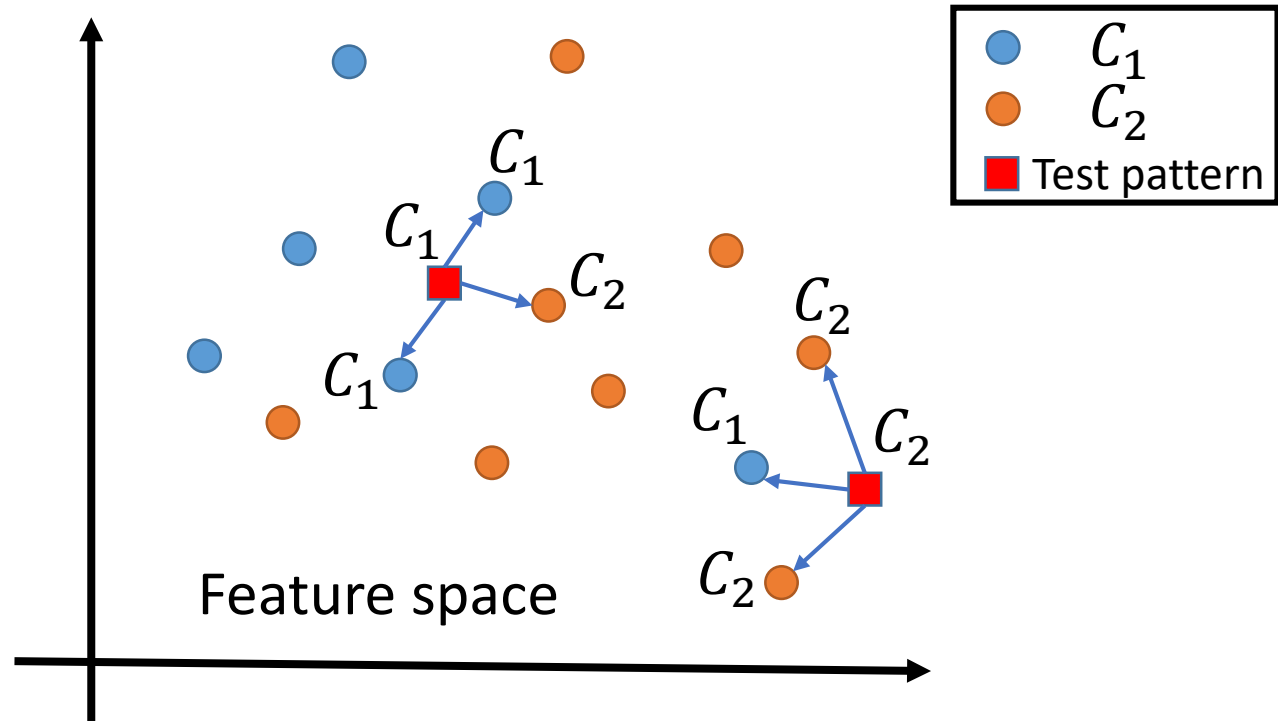
- 距離

- ユークリッド距離 : $d(\mathbf{p}, \mathbf{q}) = \sqrt{(\mathbf{q} - \mathbf{p})^T (\mathbf{q} - \mathbf{p})}$
- マハラノビス距離 : $d(\mathbf{p}, \mathbf{q}) = \sqrt{(\mathbf{q} - \mathbf{p})^T \Sigma^{-1} (\mathbf{q} - \mathbf{p})}$
- マンハッタン距離 : $d(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^d |q_i - p_i|$
- など

最近傍法の問題点



K近傍法



- k近傍法 (k nearest neighbor method)
 - 入力パターンに近いk個の学習パターンを取り上げて、その中で最も多数を占めたカテゴリを入力パターンのカテゴリとする手法.

K近傍法のアルゴリズム

1. 入力パターンと全ての学習パターンとの距離を計算する.
2. 距離の昇順に学習パターンをソートする.
3. ソートした学習パターンの上位k個を取り上げ、最も出現回数の多いカテゴリを出力する.

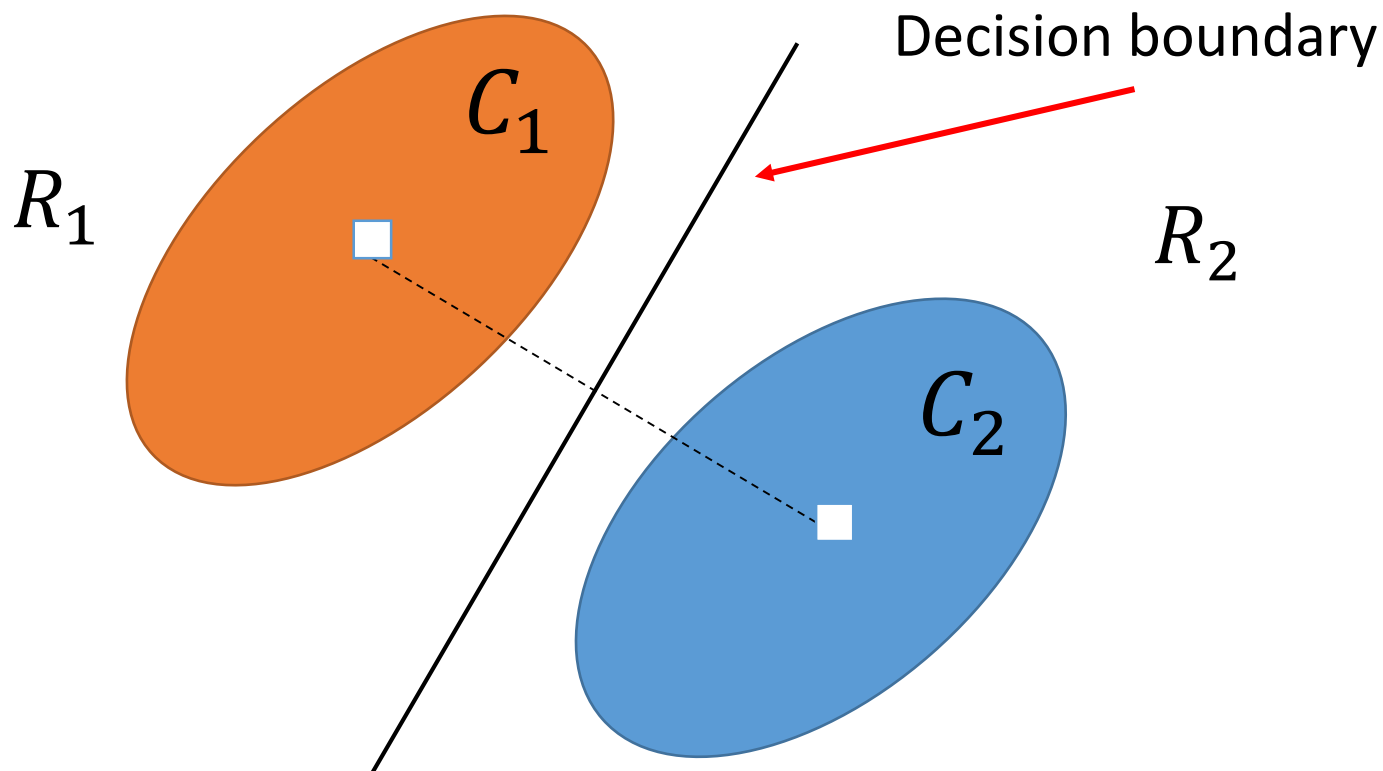
- 長所

- 単純であり実装が容易.
- 学習パターンが少なくても安定して動作する.

- 短所

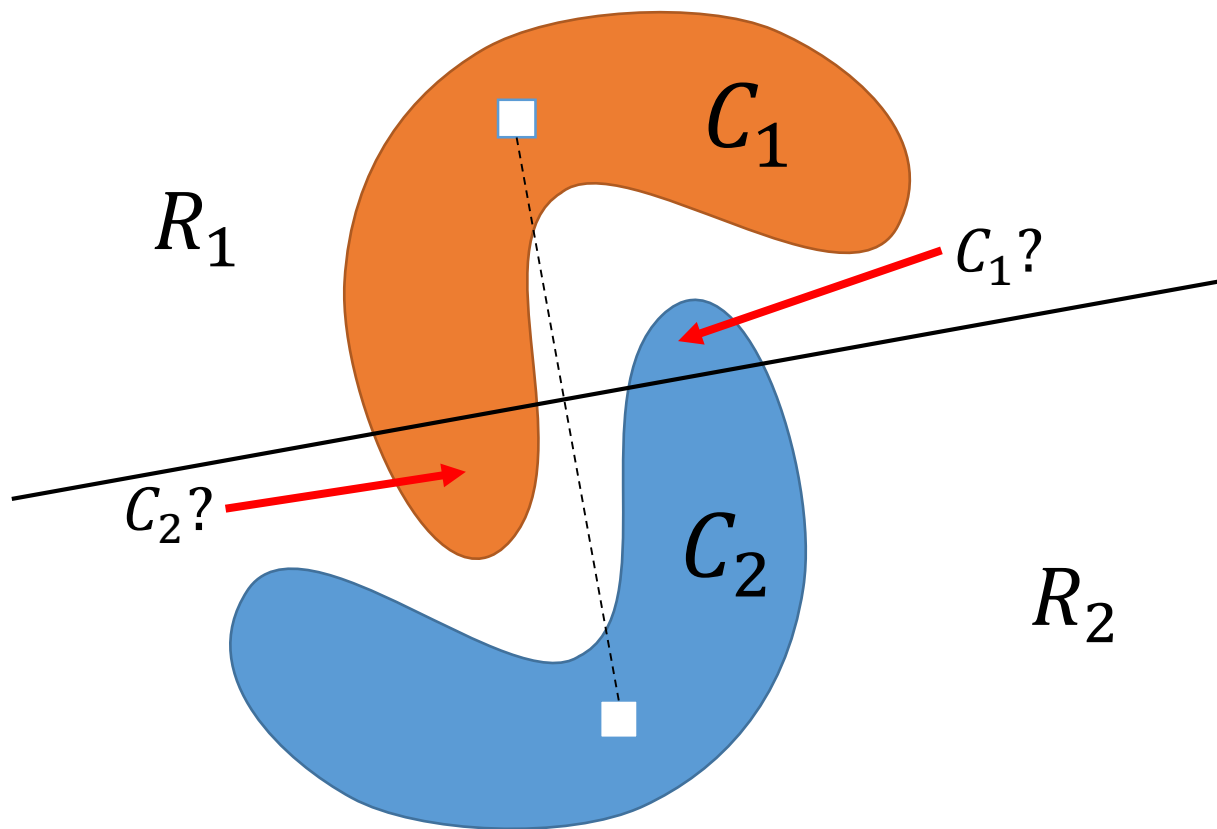
- 距離関数により結果が異なる.
- 学習データが増えると計算コストが増大する.

最近傍法と決定境界



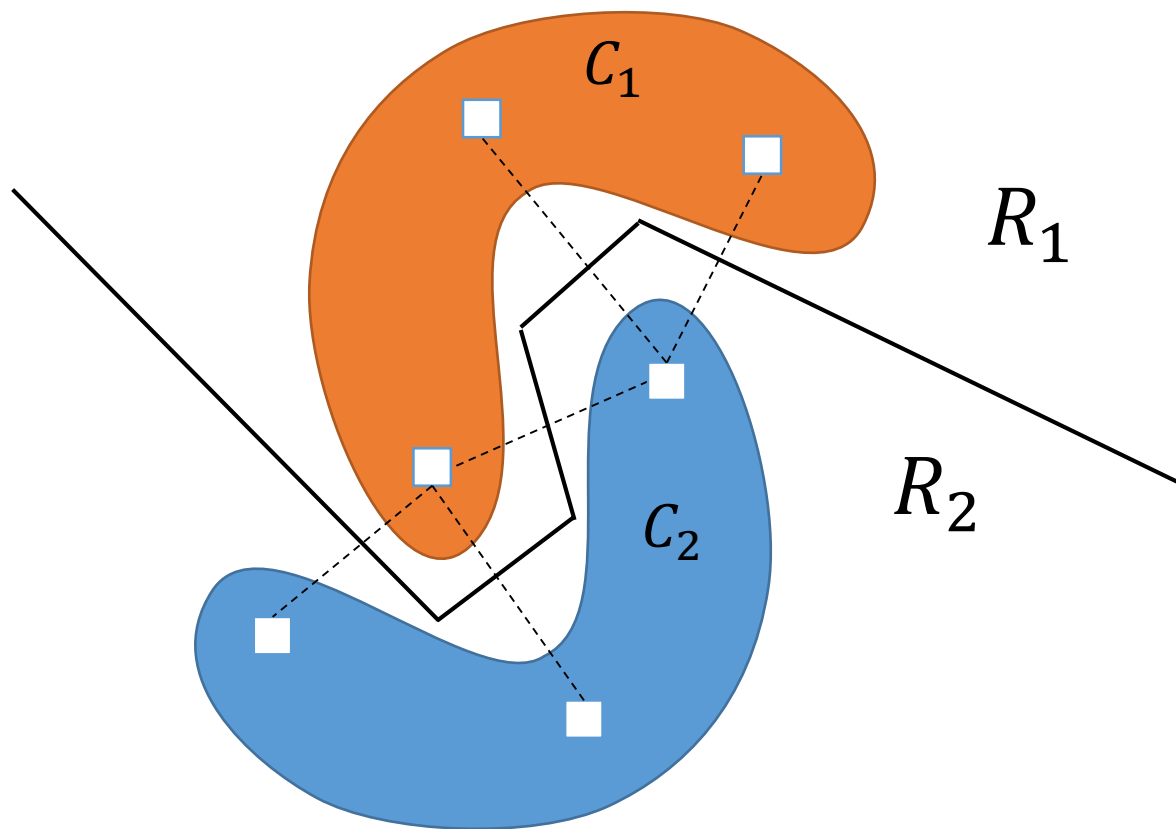
- 簡単のため，各クラス一つの代表点のみを考える．
- クラスあたり**1**つの代表点で最近傍法を適用すると，決定境界は代表点を結ぶ線の垂直二等分線となる

最近傍法と決定境界



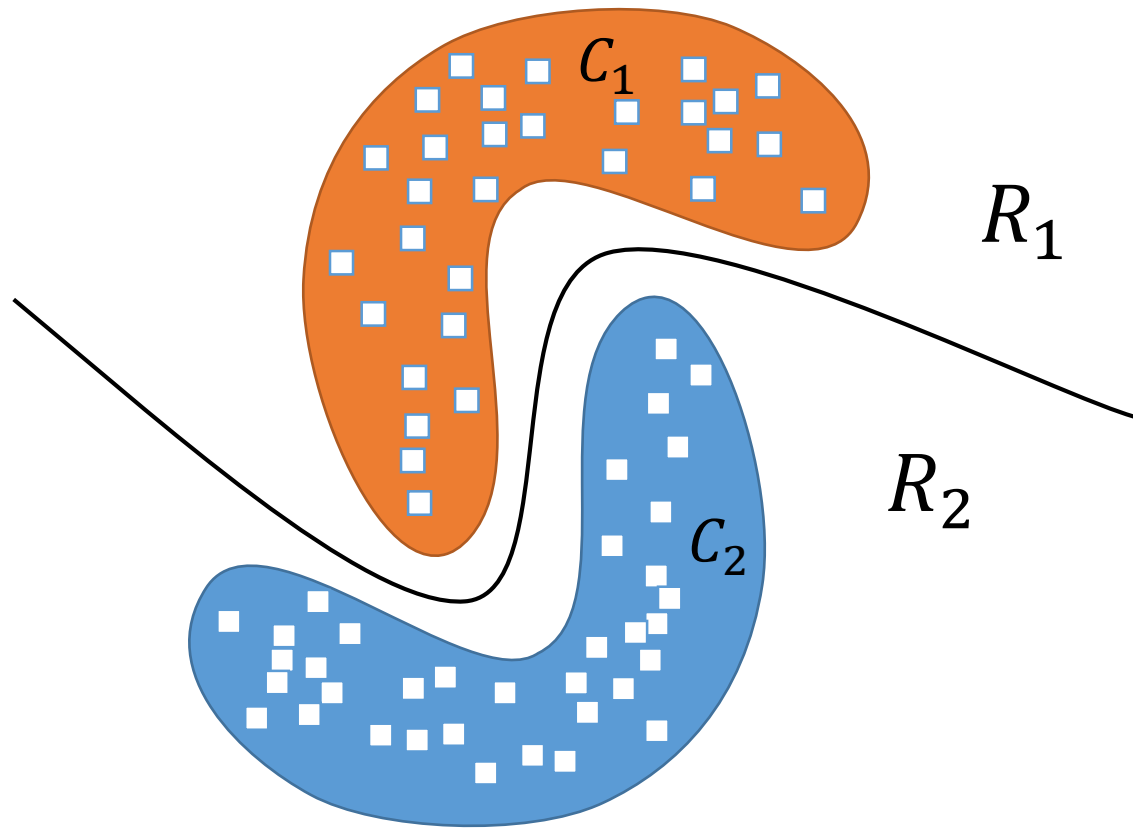
- 線形分離不可能な、複雑なパターン分布の場合、クラス当たり1つの代表点ではうまくいかない。

最近傍法と決定境界



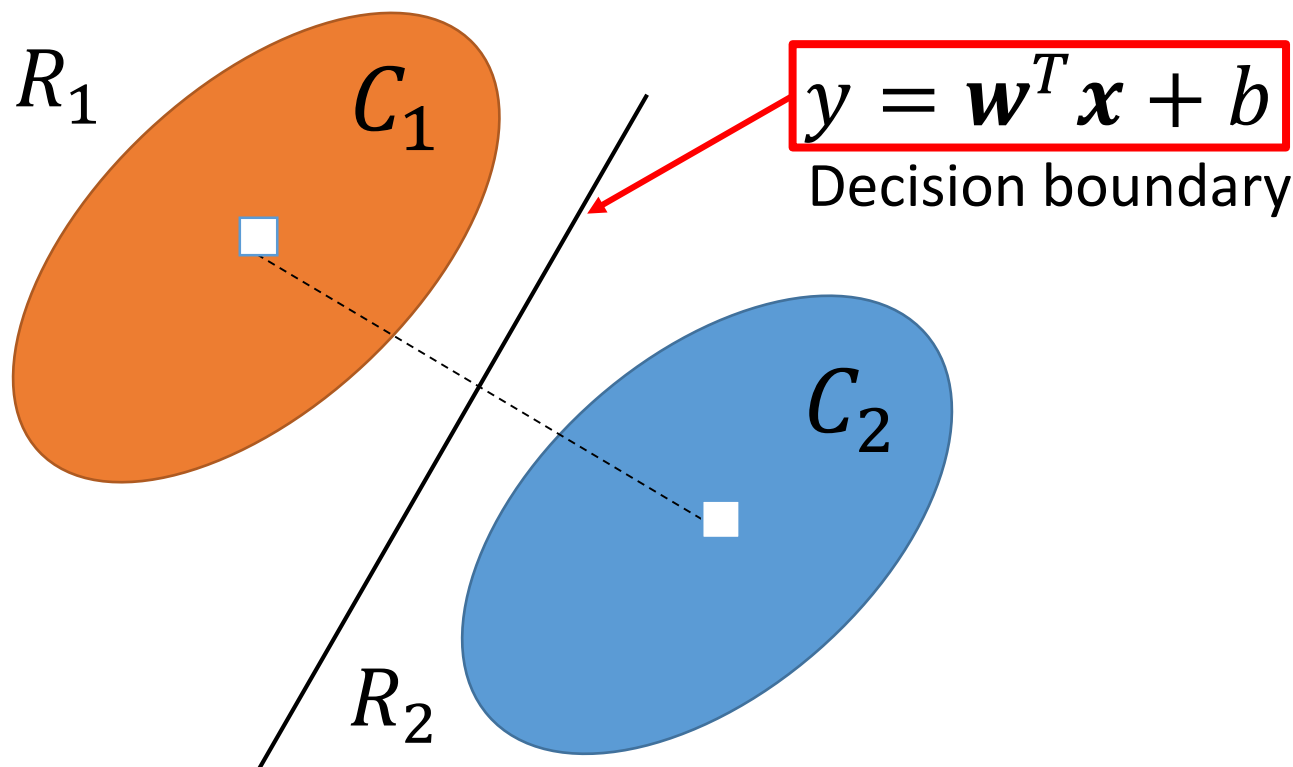
- クラス当たり3つの代表点とすると，決定境界は折れ線となり，適切に決定領域を設定可能.

最近傍法と決定境界



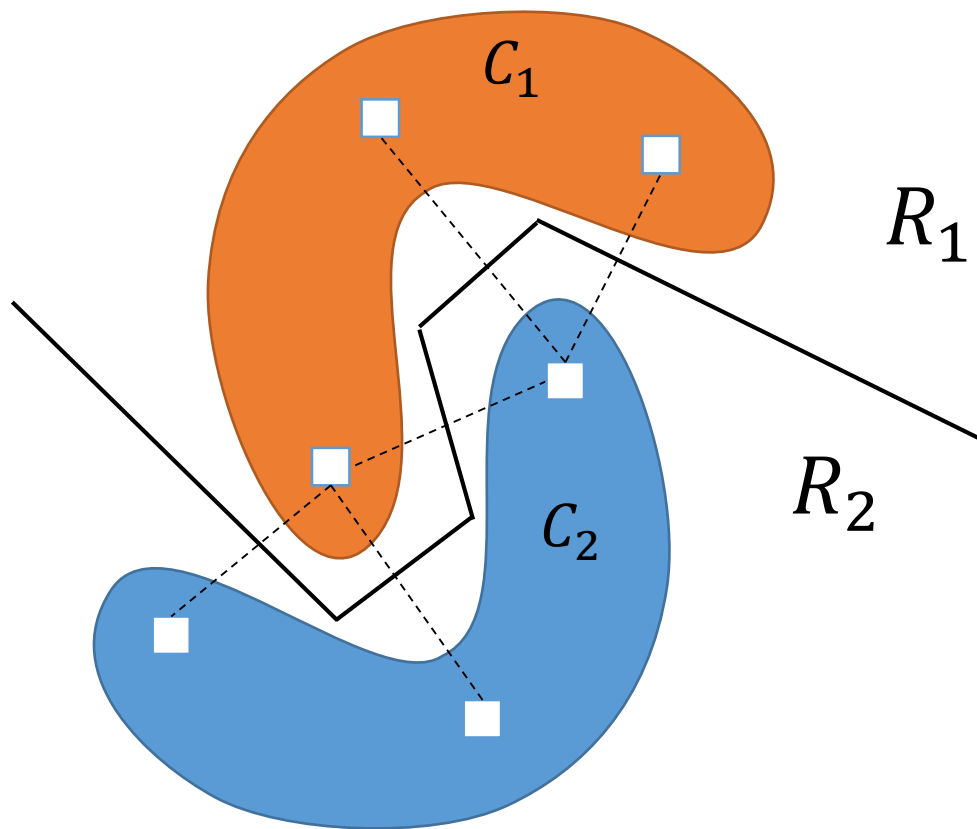
- クラス当たりの代表点が増えると，決定境界はなめらかになる．
- すべてのパターンを代表点と考えると最近傍法となる．

線形識別器



- 入力パターンに対して線形な識別器を線形識別器 (linear classifier) と呼ぶ.

区分的線形識別器

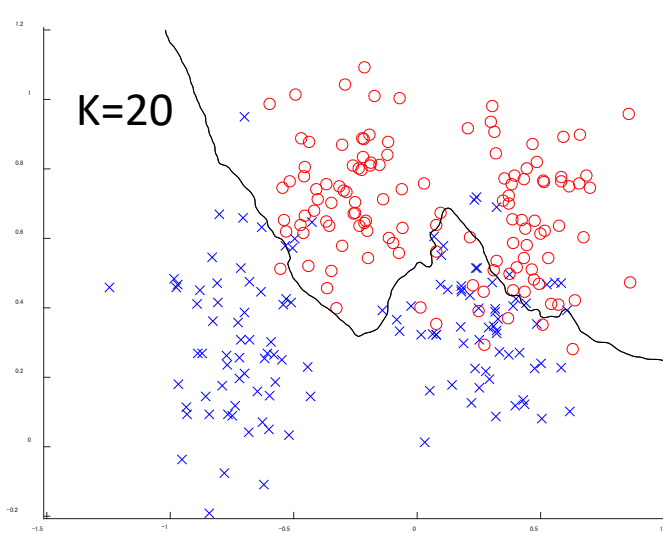
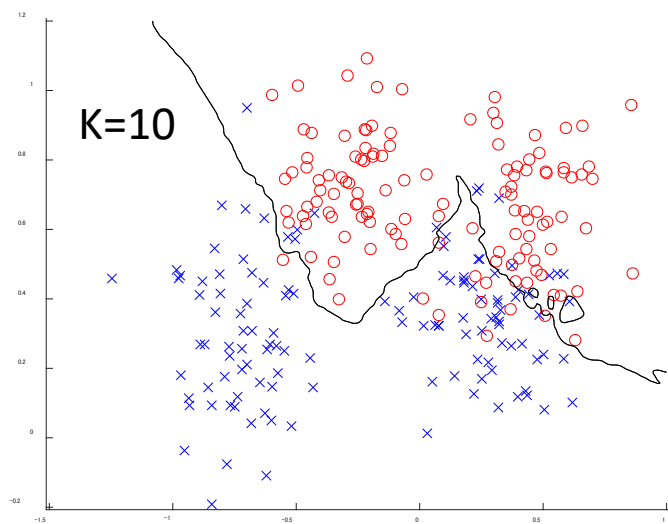
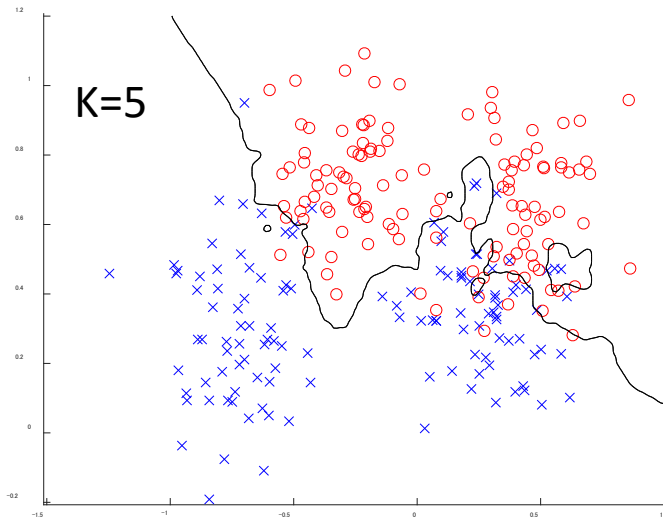
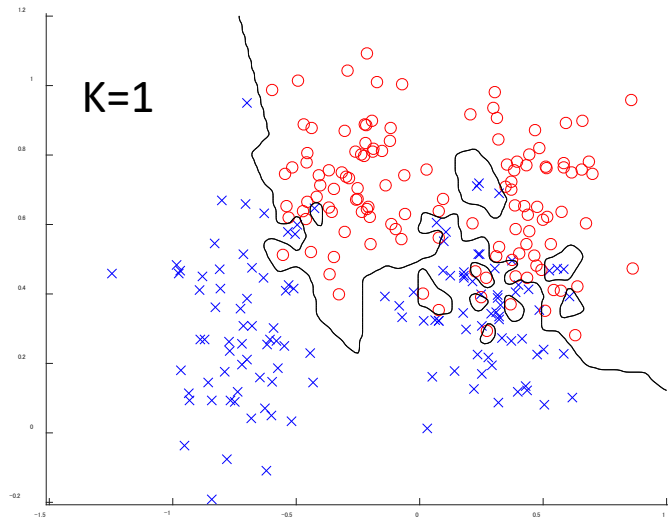


クラス C_k の識別関数

$$f_k(\mathbf{x}) = \max_{l=1, \dots, L_k} \{f_k^l(\mathbf{x})\}$$
$$f_k^l(\mathbf{x}) = b_k^l + \langle \mathbf{w}_k^l, \mathbf{x} \rangle$$

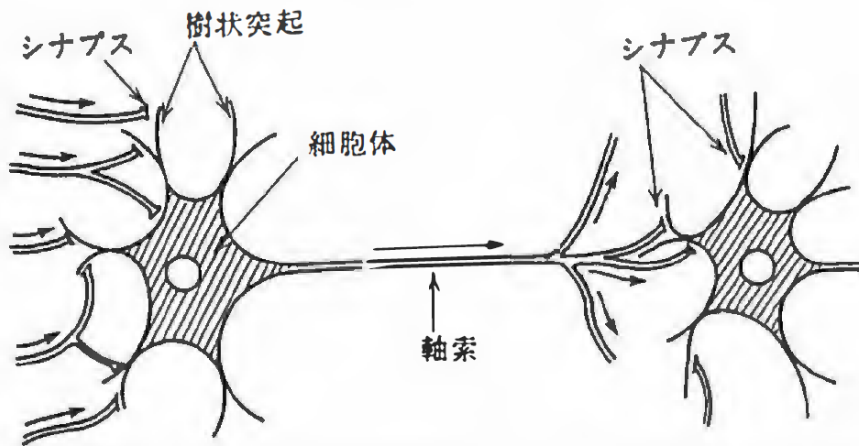
- 複数の線形識別器で構成される識別器を区分的線形識別器 (piecewise linear classifier) という。
- 最近傍法は区分的線形識別器となる。
- フィードフォワード型多層ニューラルネットワークは区分的線形識別器と関連が深い。中間層のユニット数 $\leftrightarrow$ 副次識別関数の数。

K近傍法と決定境界

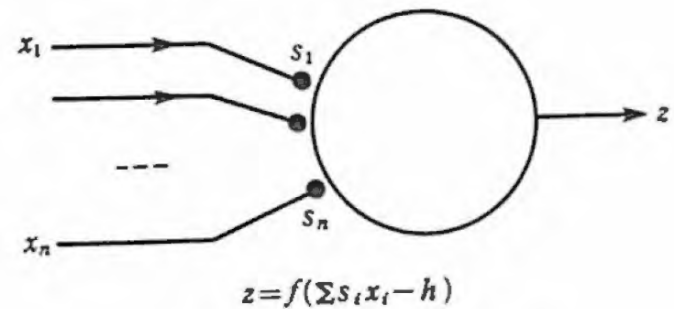


- 近傍数 k が増えるにつれて決定境界が滑らかになる。

ニューロン



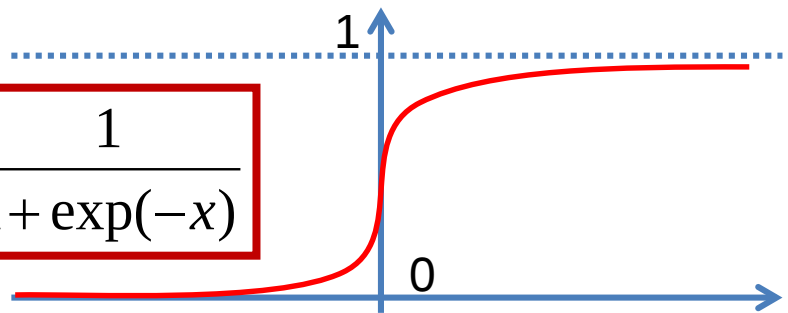
ニューロン



ニューロンの数理モデル

$$z = f\left(\sum_{i=1}^n a_i x_i + h\right)$$

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

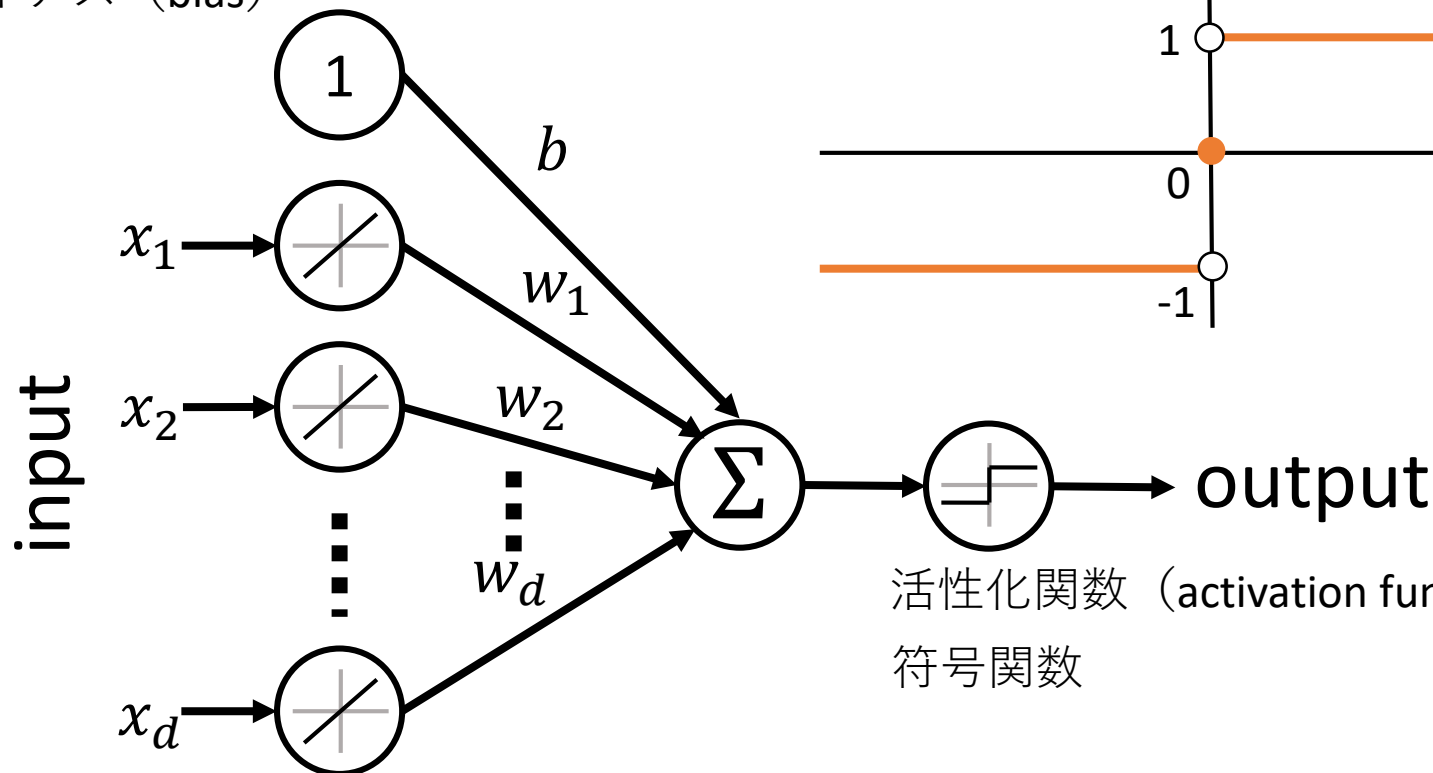


単一ニューロンモデル

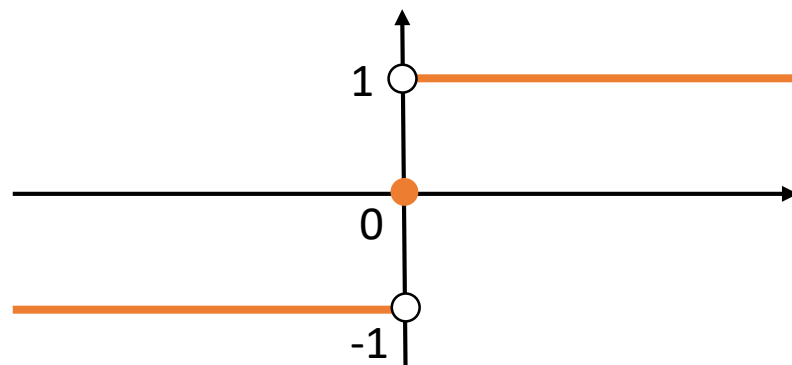
- パーセプトロン (perceptron)
 - 線形識別器の基本
 - 活性化関数は符号関数 (sign function)

$$y = \text{sgn}(\mathbf{w}^T \mathbf{x} + b)$$

バイアス (bias)



符号関数 (sign function)

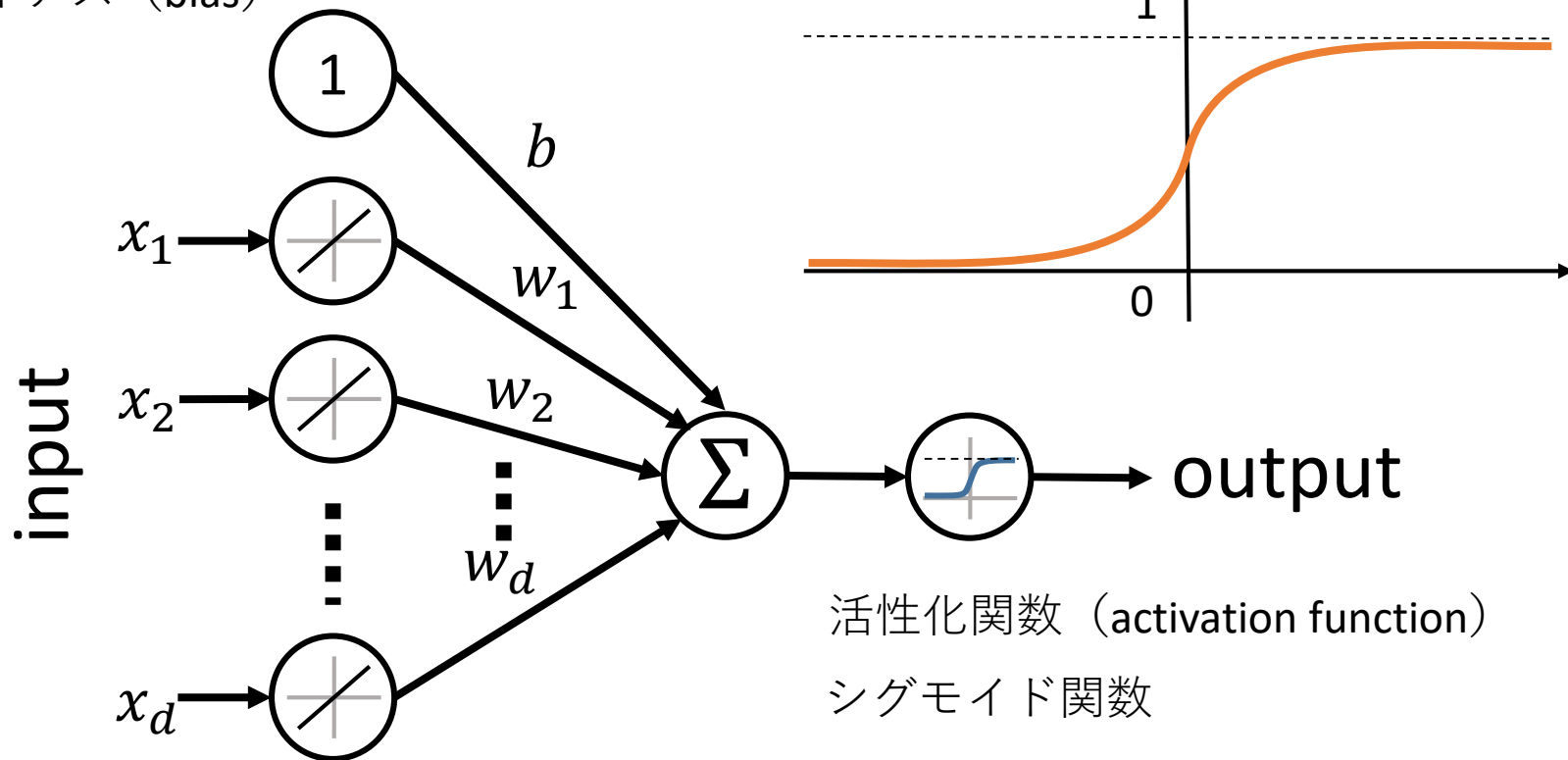


単一ニューロンモデル

- ロジスティック回帰
 - パーセプトロンの活性化関数をシグモイド関数に置き換えた手法

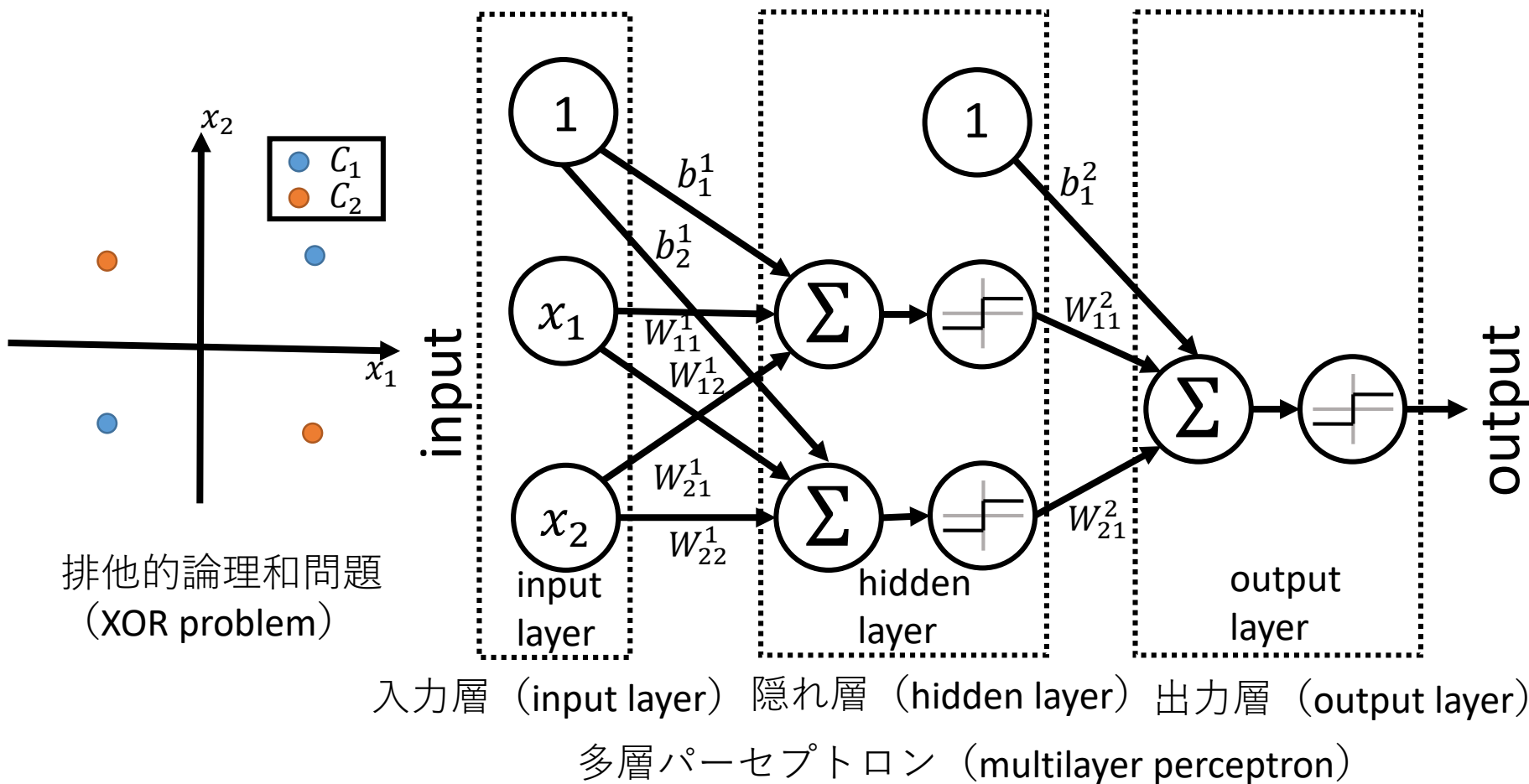
$$y = \sigma(\mathbf{w}^T \mathbf{x} + b)$$

バイアス (bias)

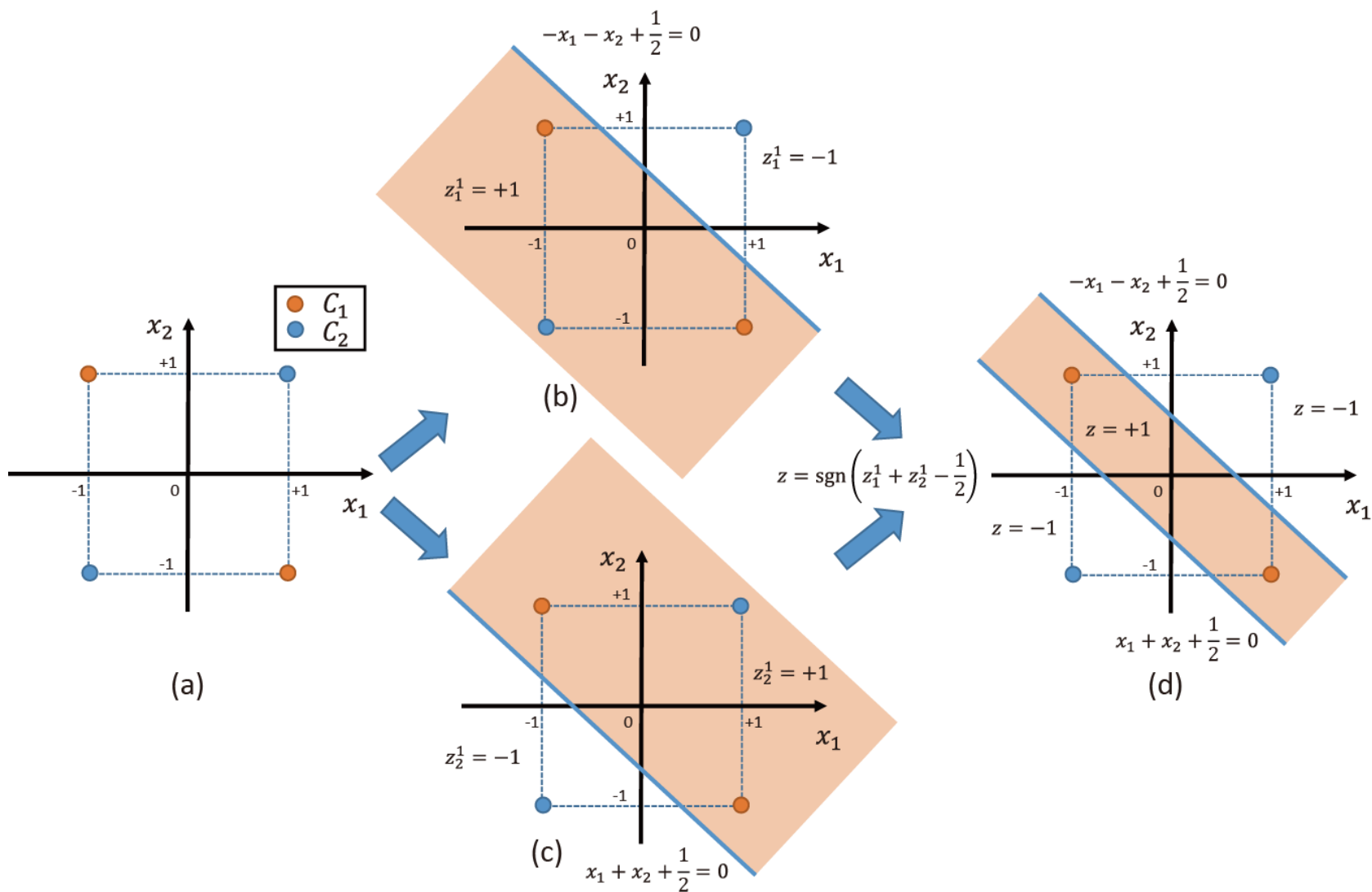


線形分離不可能なデータの識別

- 線形識別器は、例えば2次元の特徴空間では、直線1本で分離できるような問題しか対応できない。
 - 例) 排他的論理和問題では一つの線形識別器では解けない。
- 線形識別器を2つ組み合わせた識別器を考えると、適切な重みを設定することで排他的論理和問題を解くことが可能

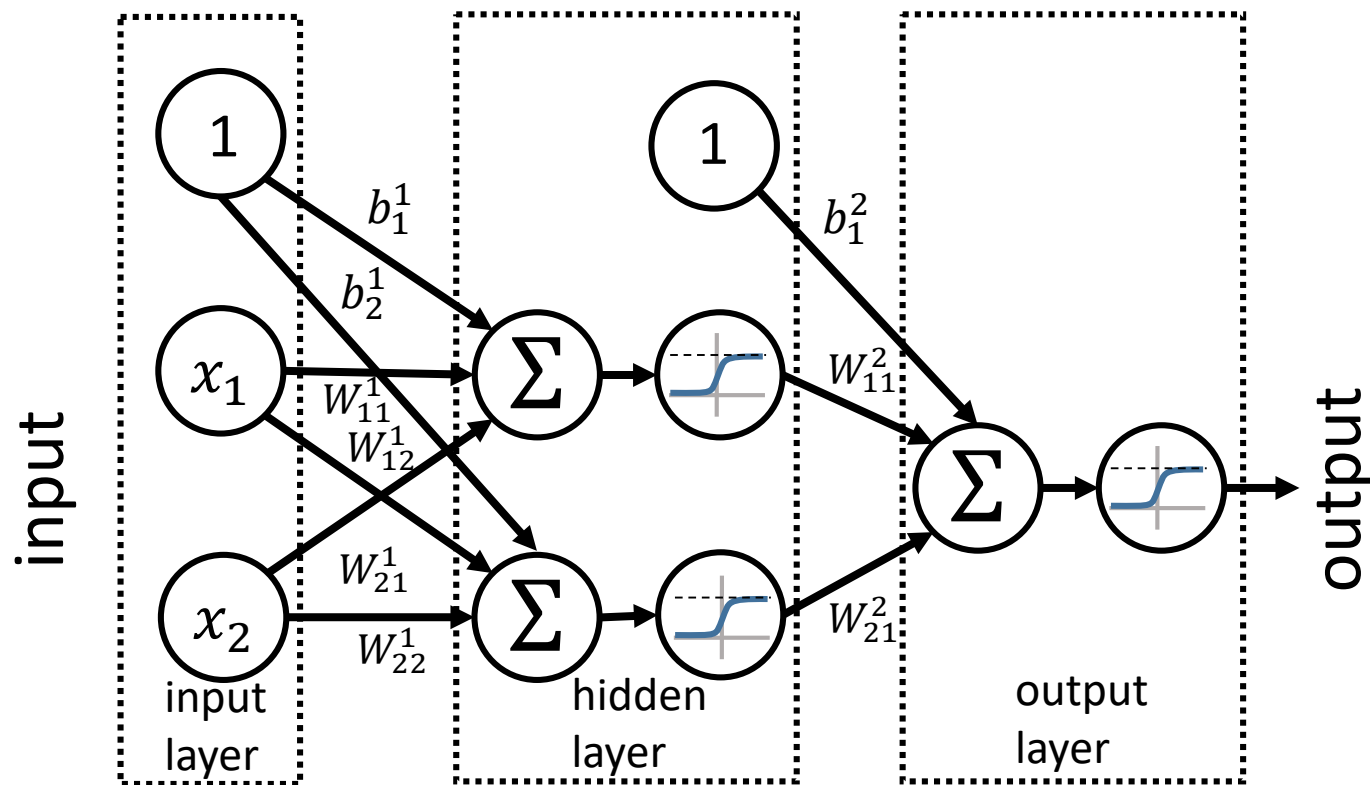


線形分離不可能なデータの識別



フィードフォワードニューラルネットワーク

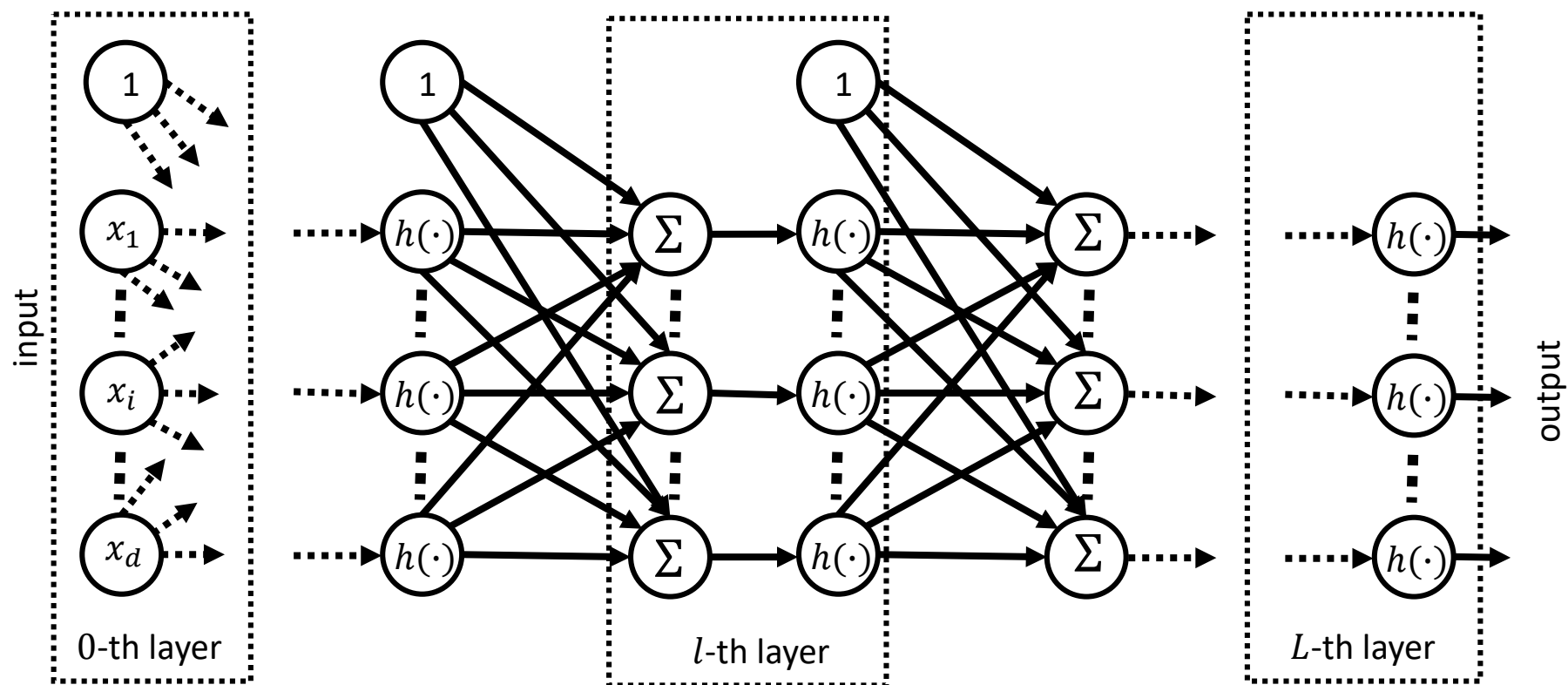
- フィードフォワードニューラルネットワーク
(feedforward neural network)
 - 活性化関数としてシグモイド関数のような微分可能な関数を利用される。



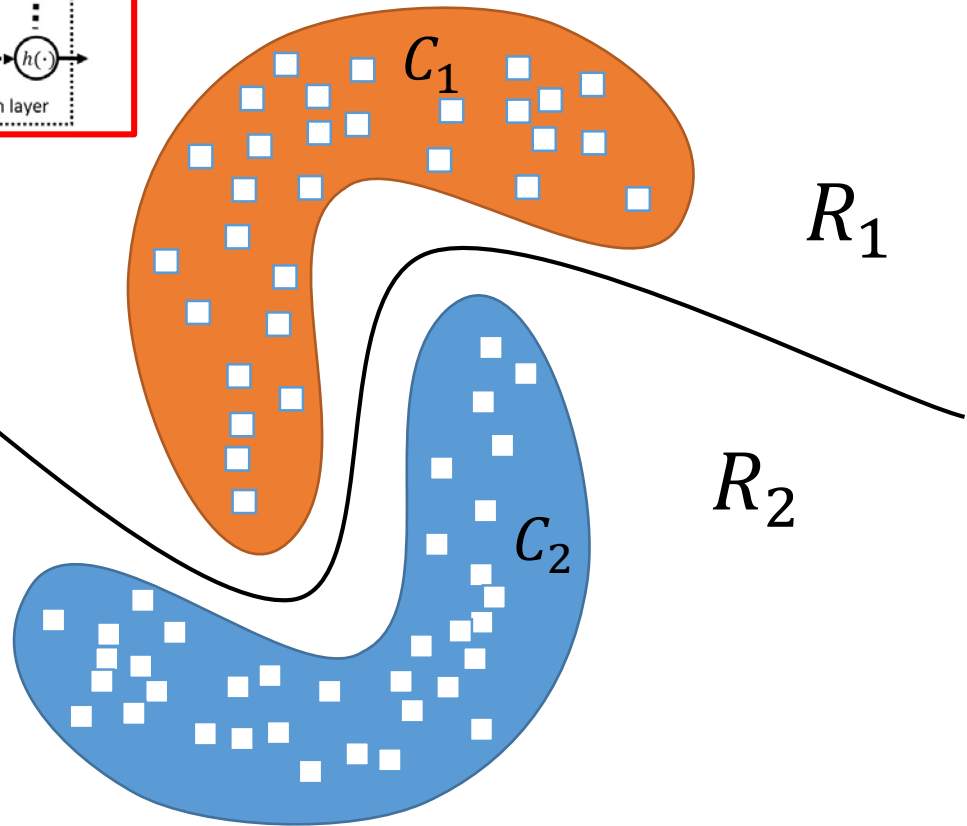
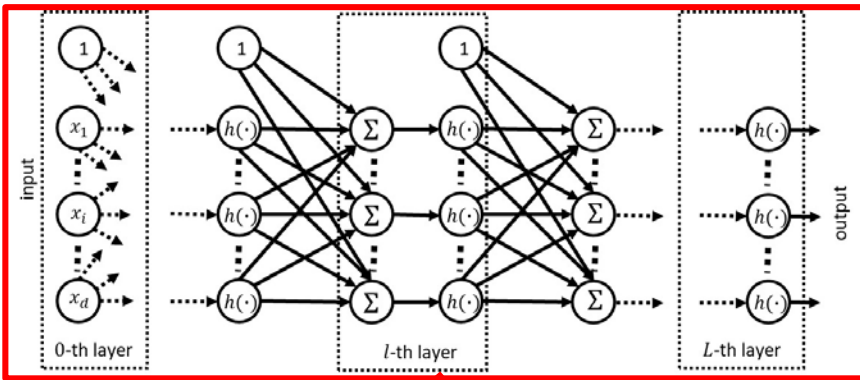
入力層 (input layer) 隠れ層 (hidden layer) 出力層 (output layer)

フィードフォワードニューラルネットワーク

- 隠れ層のユニット数や層の数を増やせば、より高い表現能力を持つ識別器を構成可能
- フィードフォワードニューラルネットワークは入力層，隠れ層，出力層の3種類の層で構成される。
- 隠れ層は複数の層を持つことが可能
- 本講義では，入力層は0番目の層，出力層を L 番目の層とする。
- 活性化関数を $y = h(z)$ としている。



多層ニューラルネットワークと決定境界

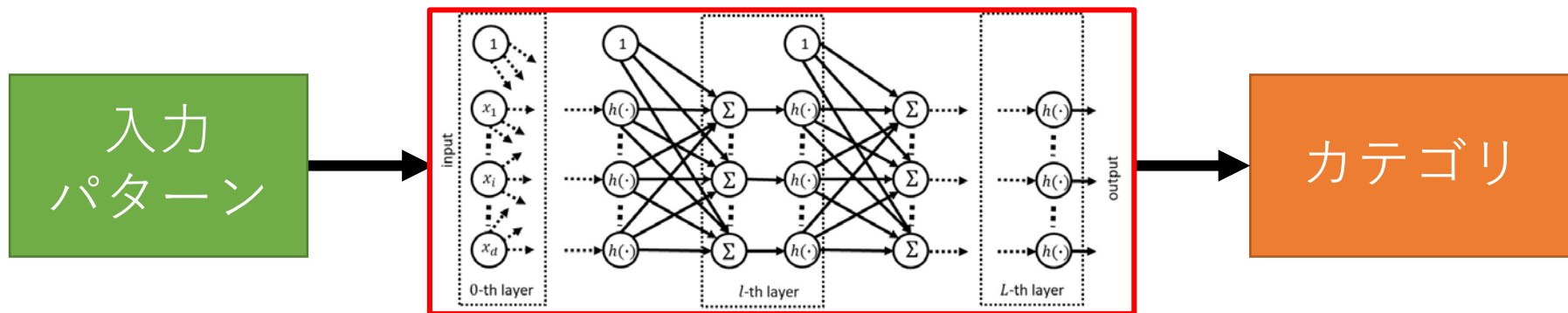
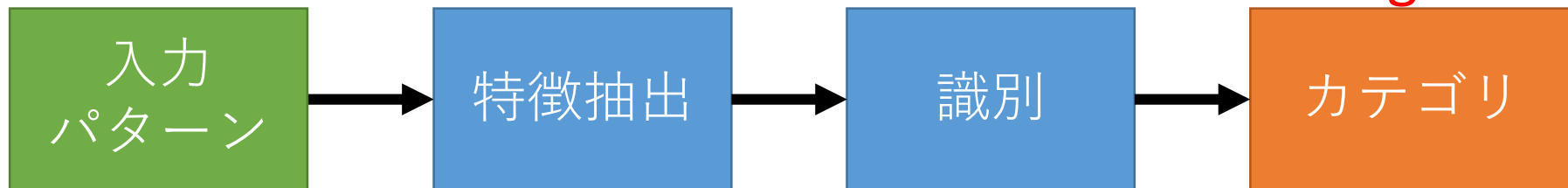
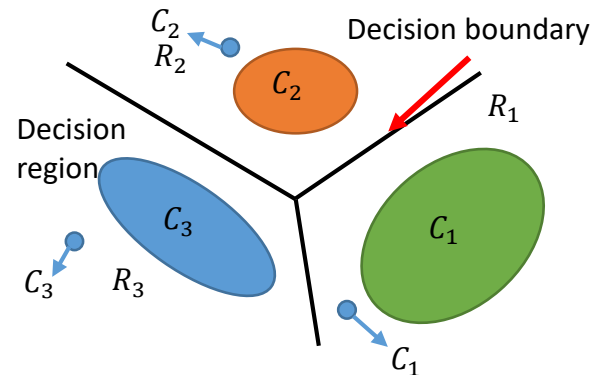
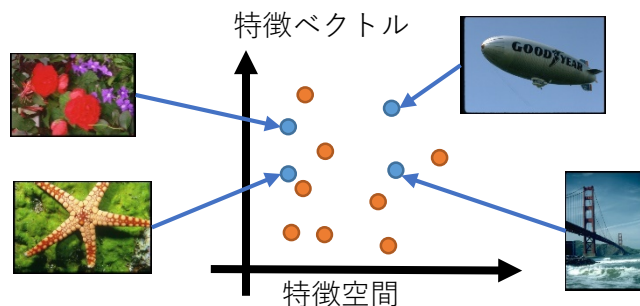


- 多層ニューラルネットワークを利用することで、任意の決定境界を引くことができるようになる。

分類問題

色 形

$$\mathbf{x} = (x_1, x_2)^T$$



なぜ今できるようになったのか？

- 基本的な考え方は**20年前のものと同じ**.
- 膨大なパラメータを学習させるための工夫.
 - バッチ正規化
 - ドロップアウト
 - 残差ネットワーク
 - ネットワーク構造の制約
- 良質かつ膨大なデータ
- 強力な計算機能力（GPGPU等）

フィードフォワードニューラルネットワークの学習

- 訓練データ集合 $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^N$ が与えられたときに、損失関数 $J(W, B)$ を（以後 J と書く）最小化する重みパラメータ $W = \{W^1 \dots W^L\}$ とバイアスパラメータ $B = \{\mathbf{b}^1 \dots \mathbf{b}^L\}$ を求める.
- ここでは勾配降下法によりパラメータを求める.
 - 損失関数が非凸であるために局所解に陥る可能性があるが、実用上うまくいく場合が多い.
 - $W^l = W^l - \varepsilon \frac{\partial J}{\partial W^l}, \quad \mathbf{b}^l = \mathbf{b}^l - \varepsilon \frac{\partial J}{\partial \mathbf{b}^l}$
- パラメータの初期化
 - パラメータの初期値を0に近いランダムな値を設定.
 - $W_{ij}^l \sim \mathcal{N}(0, \epsilon^2), \quad b_j^l \sim \mathcal{N}(0, \epsilon^2)$

---- Algorithm 3: backward propagation (online learning)---

input $X = (\mathbf{x}_1 \cdots \mathbf{x}_N)^T$, $W = \{W^1 \cdots W^L\}$, $B = \{\mathbf{b}^1 \cdots \mathbf{b}^L\}$

output $W = \{W^1 \cdots W^L\}$, $B = \{\mathbf{b}^1 \cdots \mathbf{b}^L\}$

initialize $W = \{W^1 \cdots W^L\}$, $B = \{\mathbf{b}^1 \cdots \mathbf{b}^L\}$

for $m = 1, \dots, M$ do **% M: maximum iterations**

 for $\mathbf{z}^0 = \mathbf{x}_i, i = 1, \dots, N$ do

% forward propagation

 for $l = 1, \dots, L$ do $\mathbf{u}^l = (W^l)^T \mathbf{z}^{l-1} + \mathbf{b}^l$, $\mathbf{z}^l = h^l(\mathbf{u}^l)$ end for

% error of output layer

$$\boldsymbol{\delta}^L = \frac{\partial J}{\partial \mathbf{u}^L}$$

$$\frac{\partial J}{\partial W^L} = \mathbf{z}^{L-1} (\boldsymbol{\delta}^L)^T, \quad \frac{\partial J}{\partial \mathbf{b}^L} = \boldsymbol{\delta}^L$$

% back propagation

 for $l = L - 1, \dots, 1$ do

$$\boldsymbol{\delta}^l = h'^l(\mathbf{u}^l) \odot (W^{l+1} \boldsymbol{\delta}^{l+1}) \quad \text{\textbf{\% error}}$$

$$\frac{\partial J}{\partial W^l} = \mathbf{z}^{l-1} (\boldsymbol{\delta}^l)^T, \quad \frac{\partial J}{\partial \mathbf{b}^l} = \boldsymbol{\delta}^l \quad \text{\textbf{\% gradient}}$$

 end for

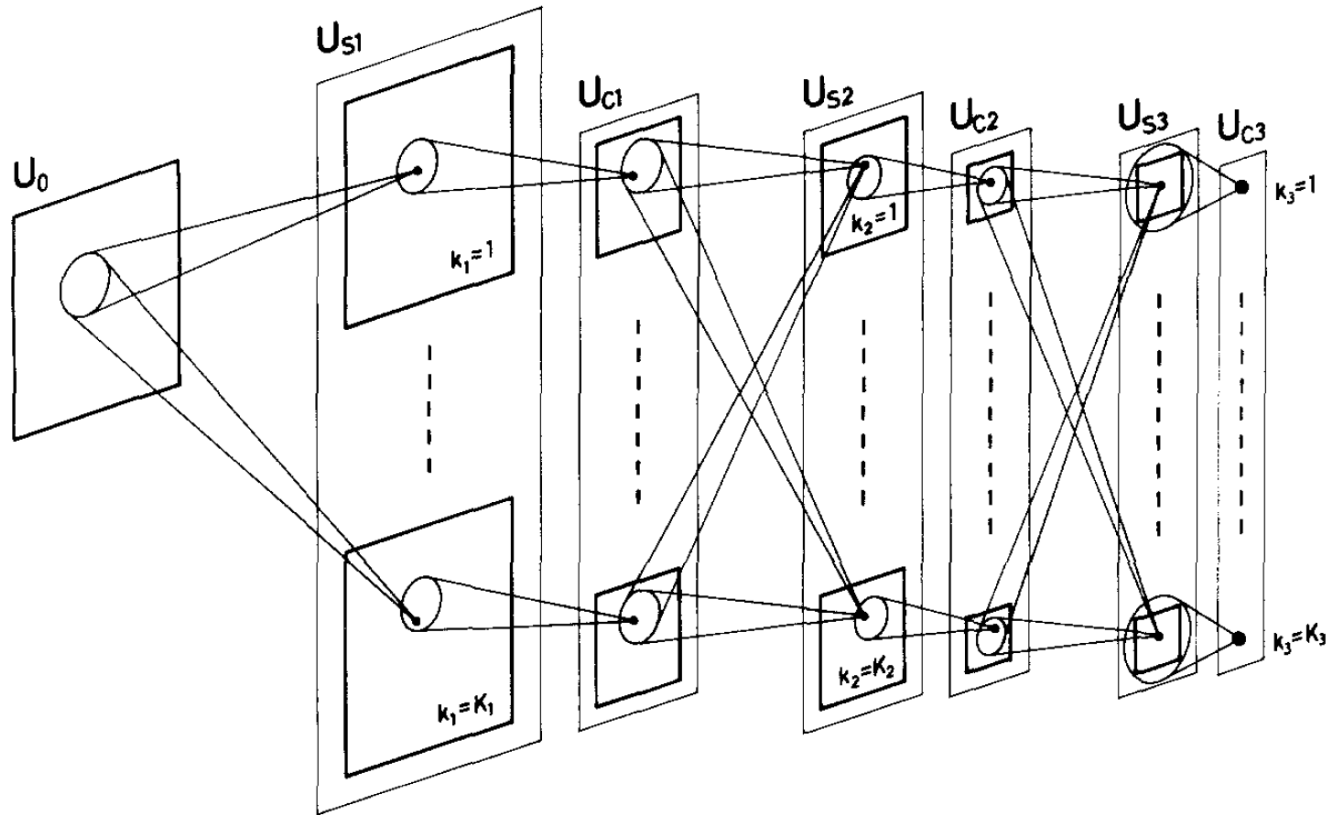
% online learning

 for $l = 1, \dots, L$ do $W^l = W^l - \varepsilon \frac{\partial J}{\partial W^l}$, $\mathbf{b}^l = \mathbf{b}^l - \varepsilon \frac{\partial J}{\partial \mathbf{b}^l}$ end for

 end for

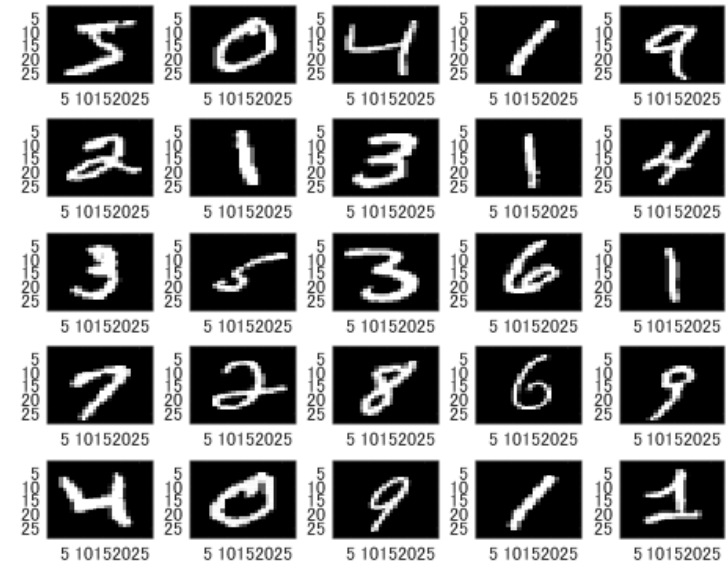
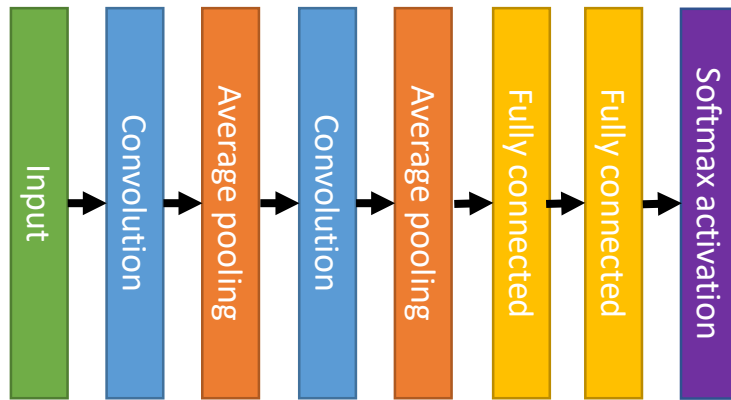
end for

Neocognitron



Kunihiko Fukushima. Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. Biol. Cybernetics 36, 193 202 (1980)

Convolutional Neural Networks



(a) LeNet-5, 1989

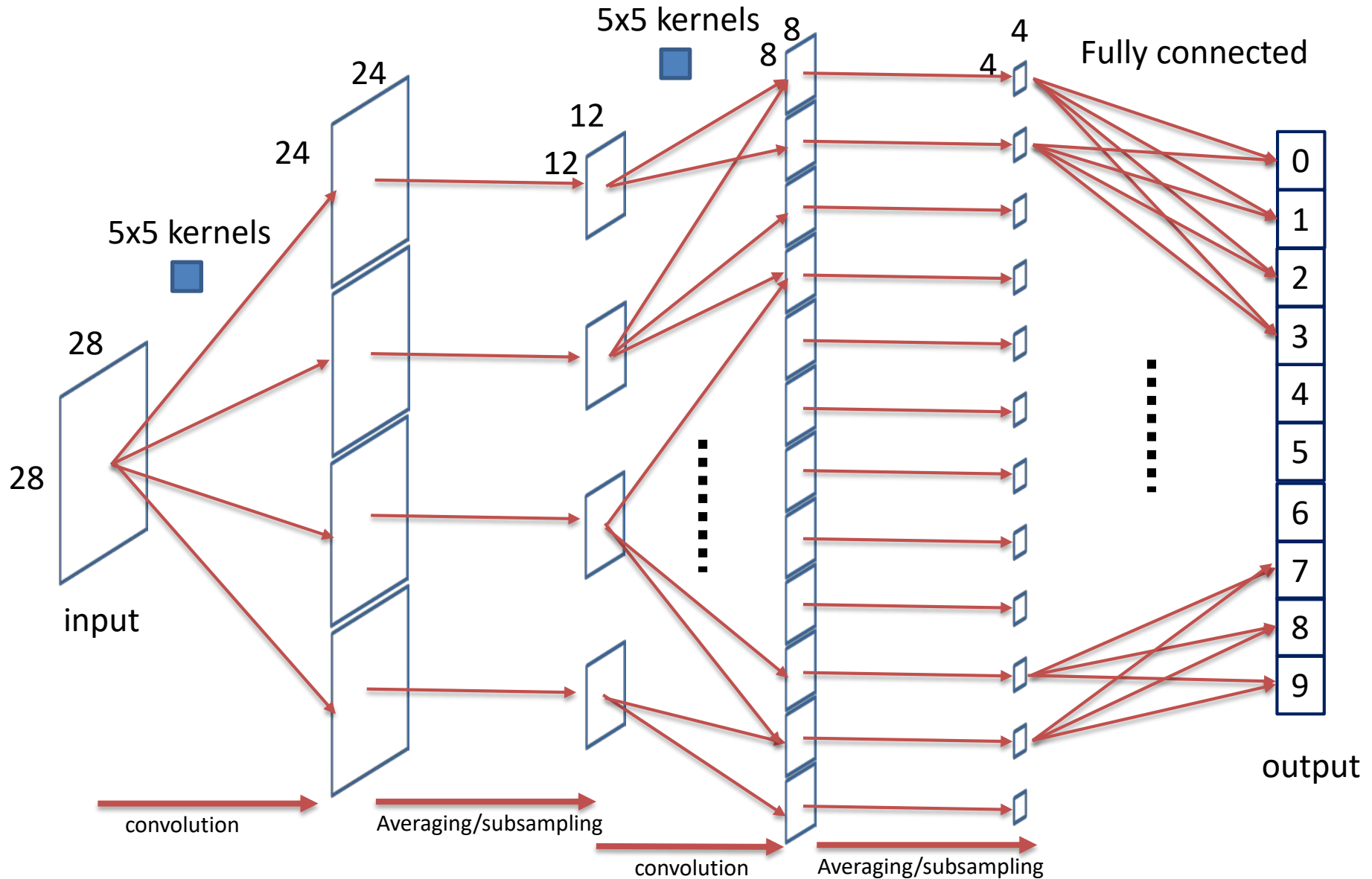
Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. NIPS 1989.

Modules of Convolutional Neural Networks

- Convolution layer
- fully-connected layer
- Pooling layer

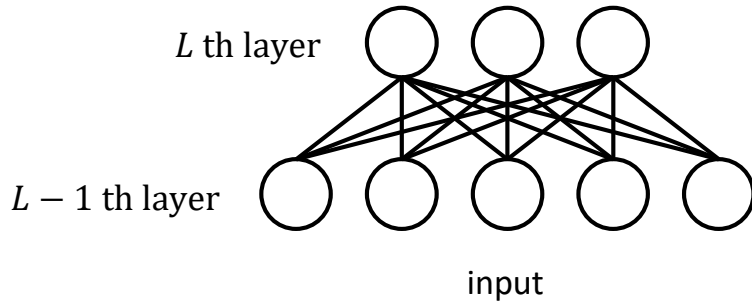
Example of CNNs Architecture

- Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In David Touretzky, editor, Advances in Neural Information Processing Systems 2 (NIPS*89), Denver, CO, 1990. Morgan Kaufman.

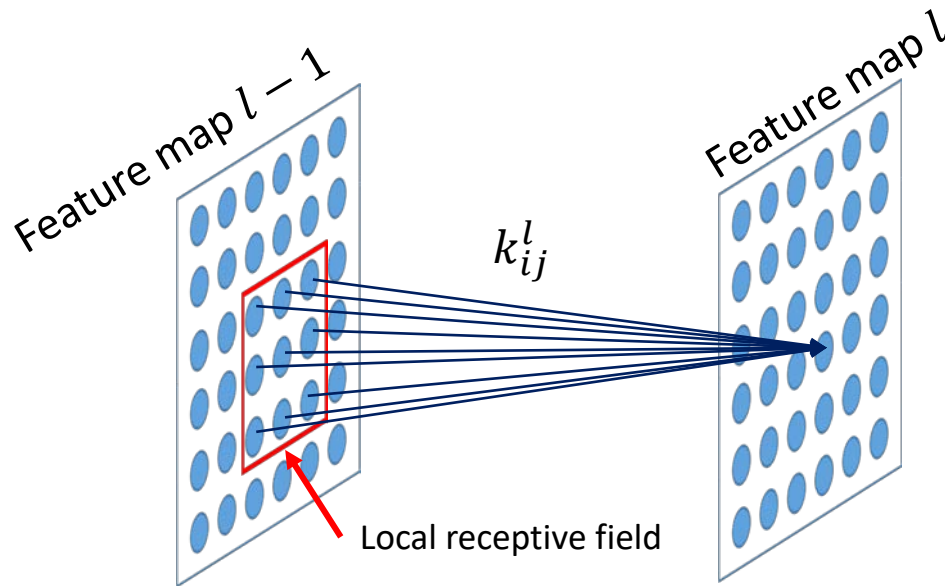
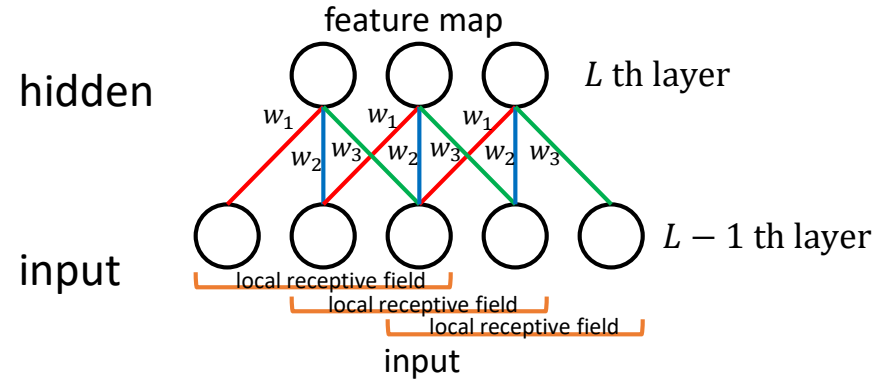


Convolutional and Fully-Connected Layer

Fully-Connected Layer



Convolutional Layer



Logistic sigmoid function

$$f(x) = (1 + e^{-\beta x})^{-1}$$

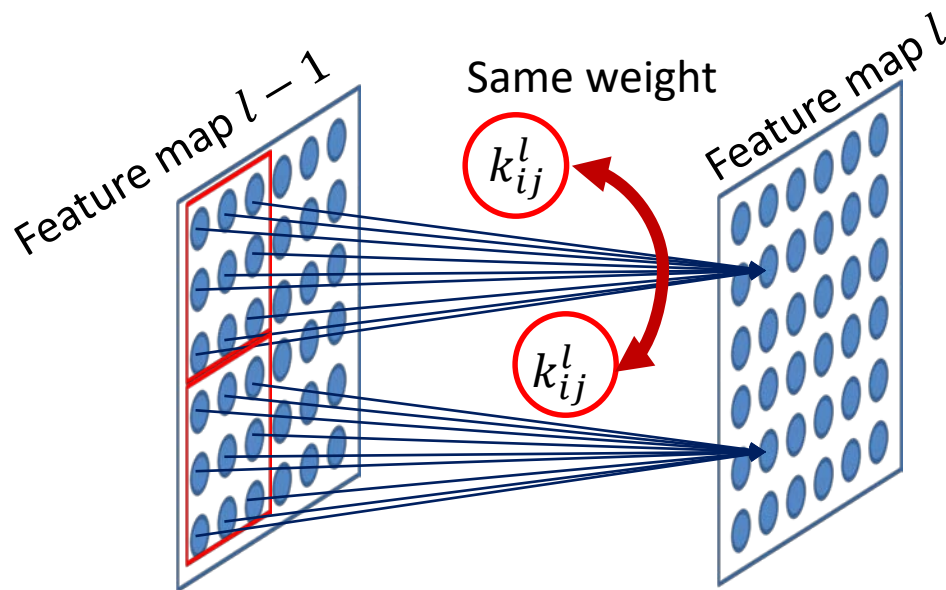
Hyperbolic tangent function

$$f(x) = \text{atanh}(bx)$$

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j\right)$$

Weight Sharing

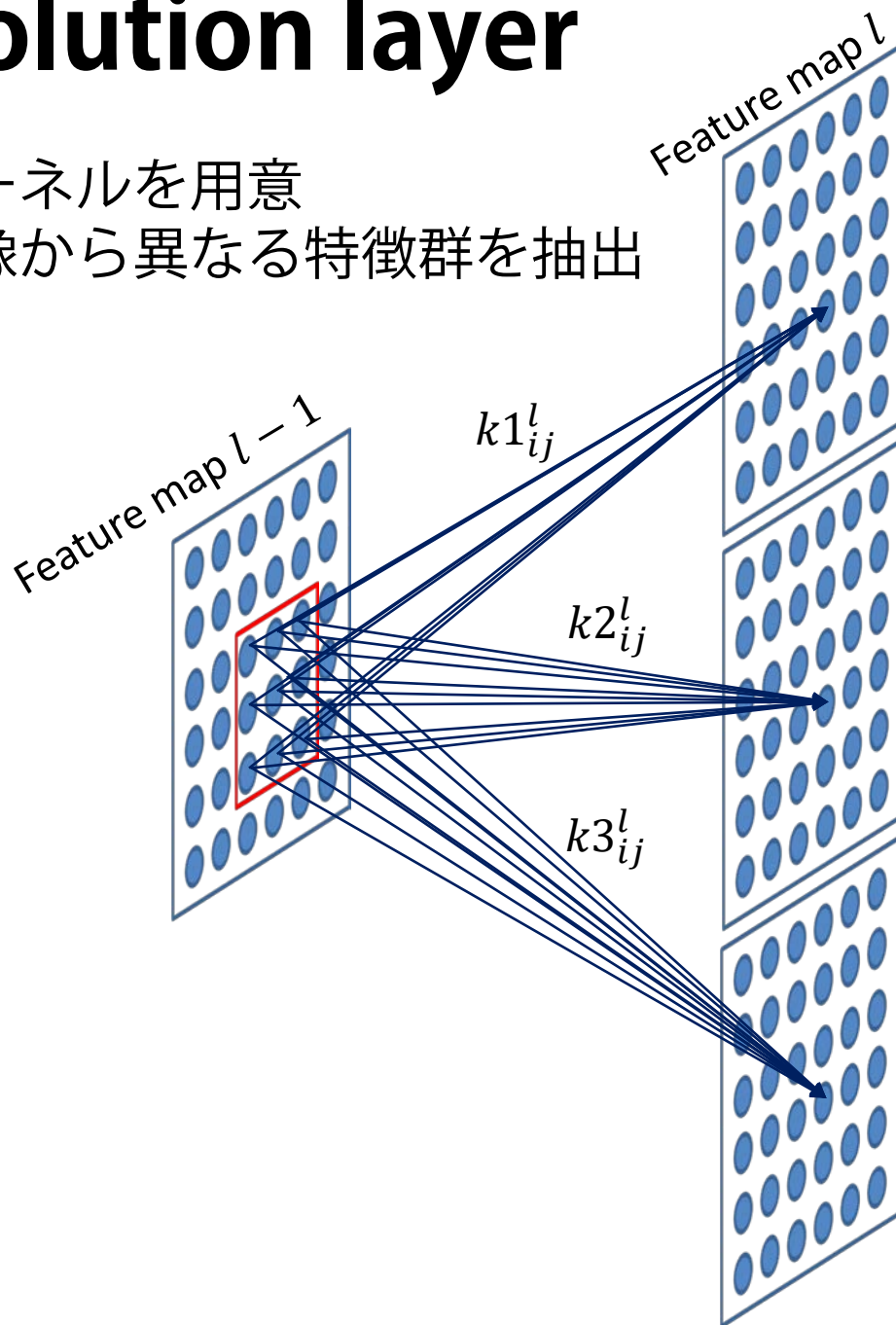
- 画像の一部で有効な特徴抽出であれば、画像の他の部分でも有効な特徴と考える。



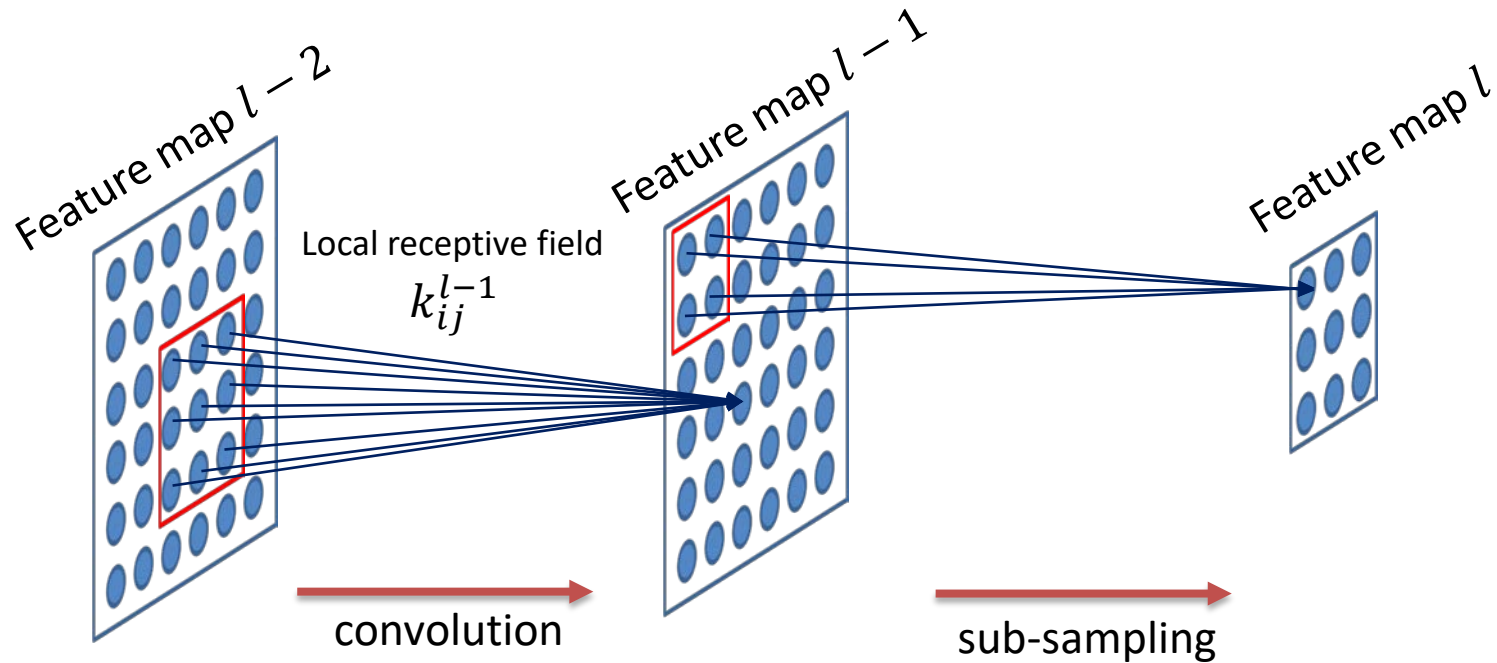
パラメータ数を大幅に減らすことができる.

Convolution layer

- 複数のカーネルを用意
- 同一の画像から異なる特徴群を抽出



Pooling Layer



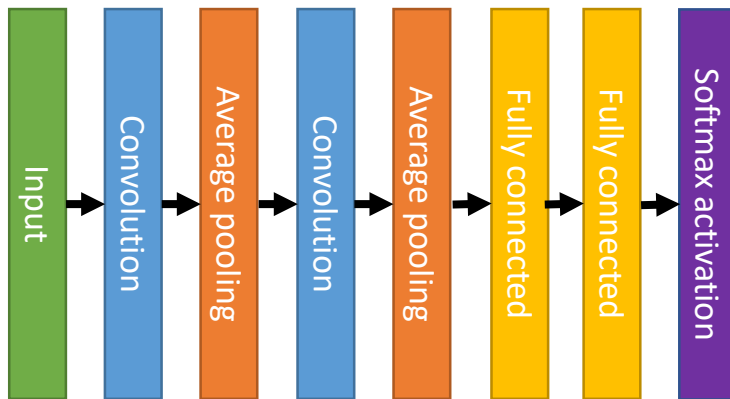
subsampling function

$$x_j^l = f(\beta_j^l \text{down}(x_i^{l-1}) + b_j^l)$$

Average pooling

Max pooling

LeNet and Alex Net

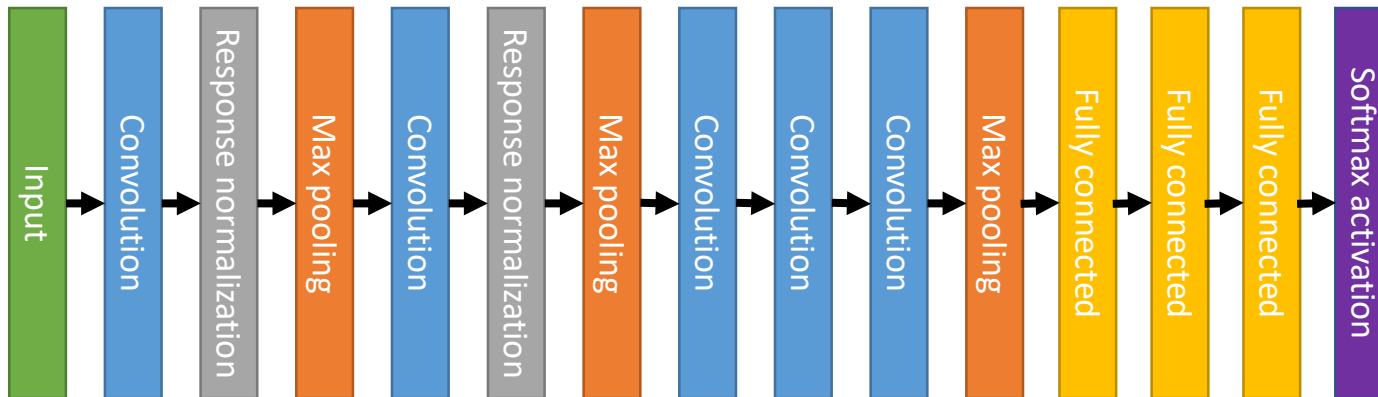


Features of Alex Net

- 5 conv. layers + 3 fully-connected layers
- ReLU nonlinearity
- Multiple GPGPUs
- Local response normalization
- Overlapping pooling
- Dropout
- Data augmentation

(a) LeNet-5, 1989

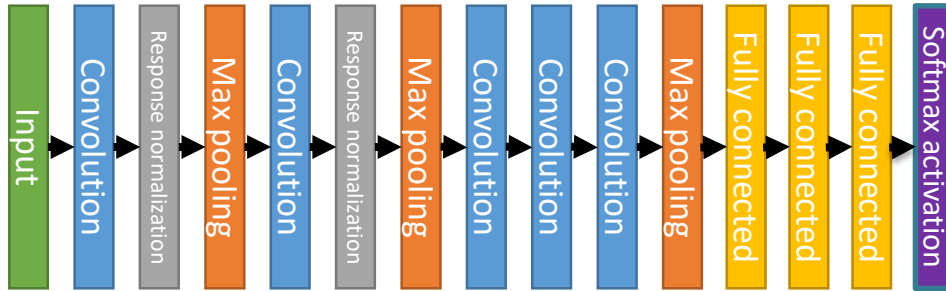
Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. NIPS 1989.



(b) AlexNet, 2012

Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. NIPS, 2012.

VGG Net

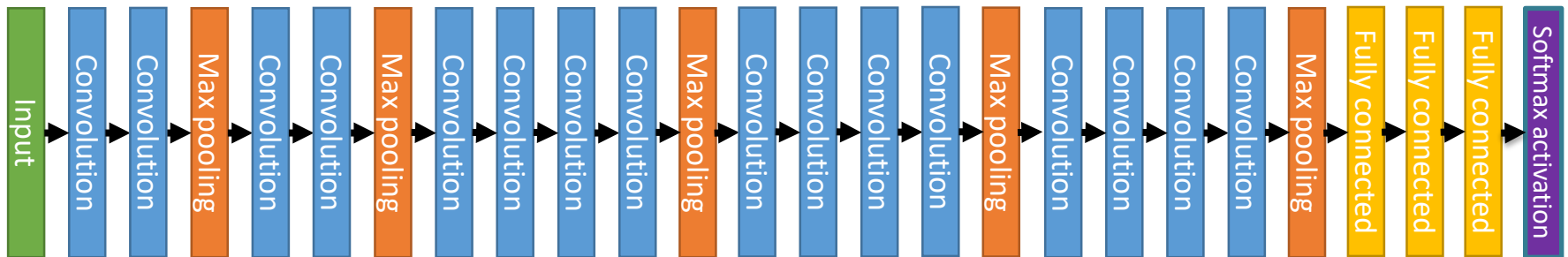


(b) AlexNet, 2012

Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. NIPS, 2012.

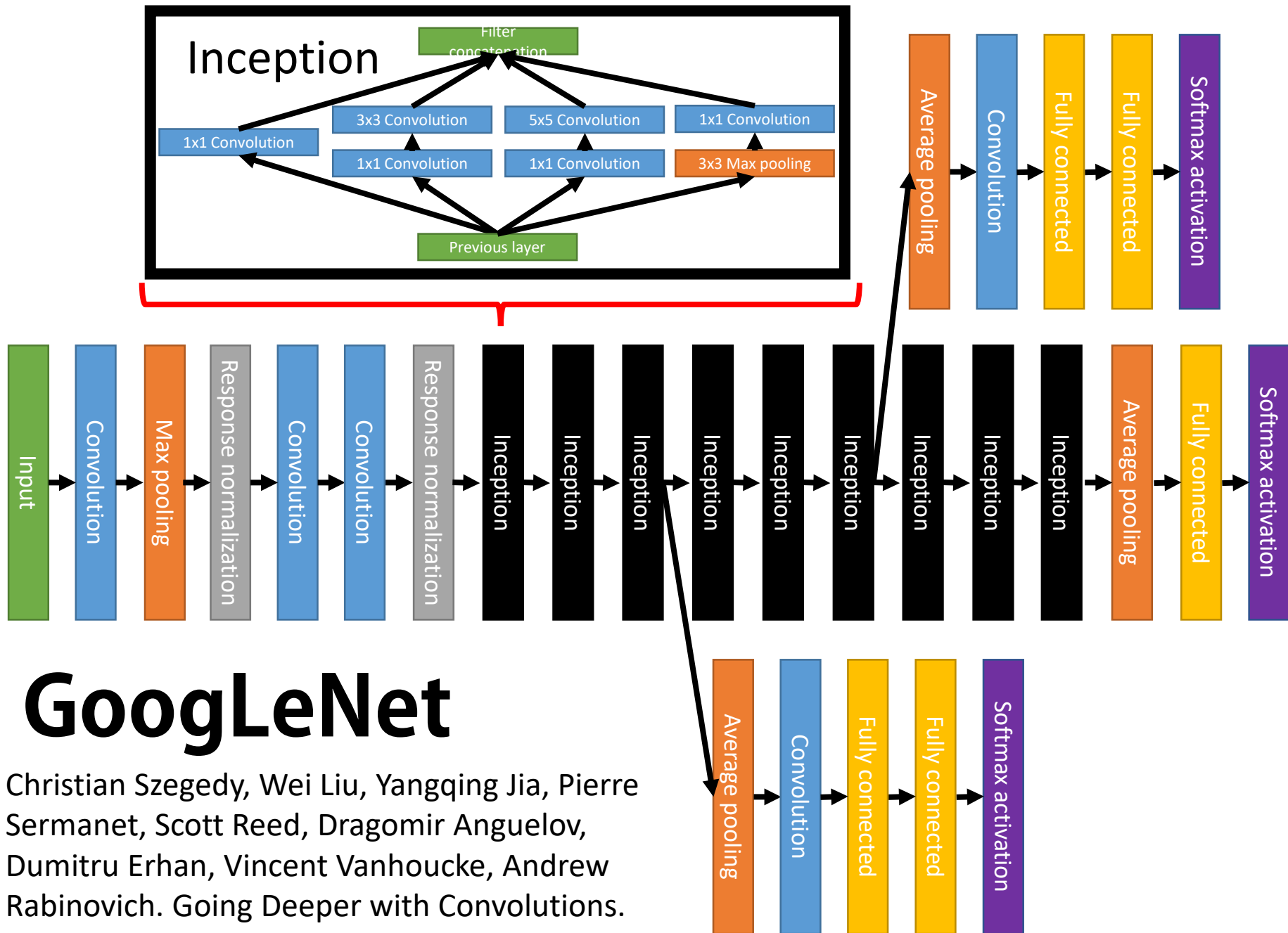
Features of VGG Net

- 16 conv. layers + 3 fully-connected layers
- Very small receptive fields
- No Local response normalization
- Data parallelism



(c) VGG Net, 2014

Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556, 2014.

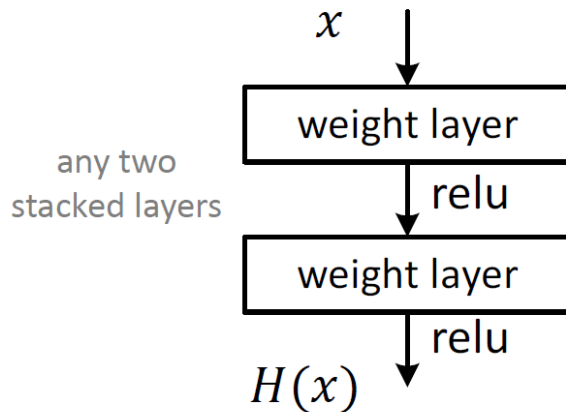


GoogLeNet

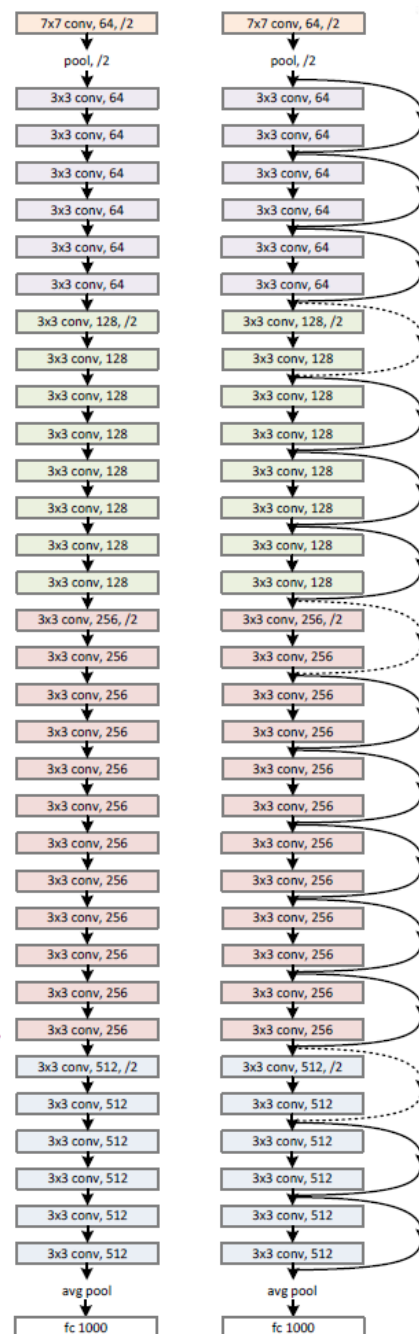
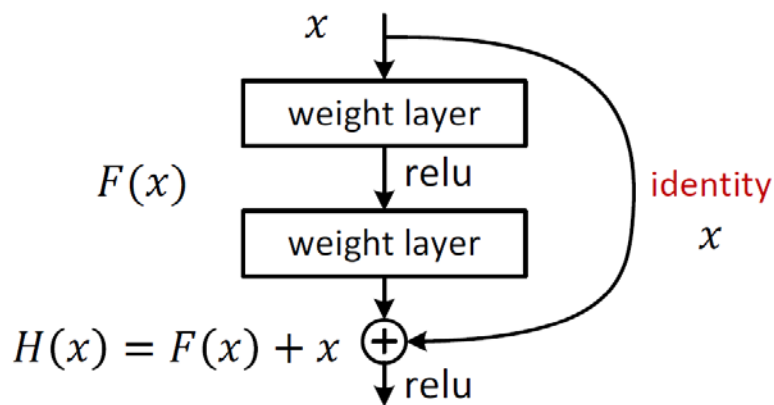
Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Going Deeper with Convolutions. arXiv:1409.4842, 2014.

Residual Network

- Plain net



- Residual net

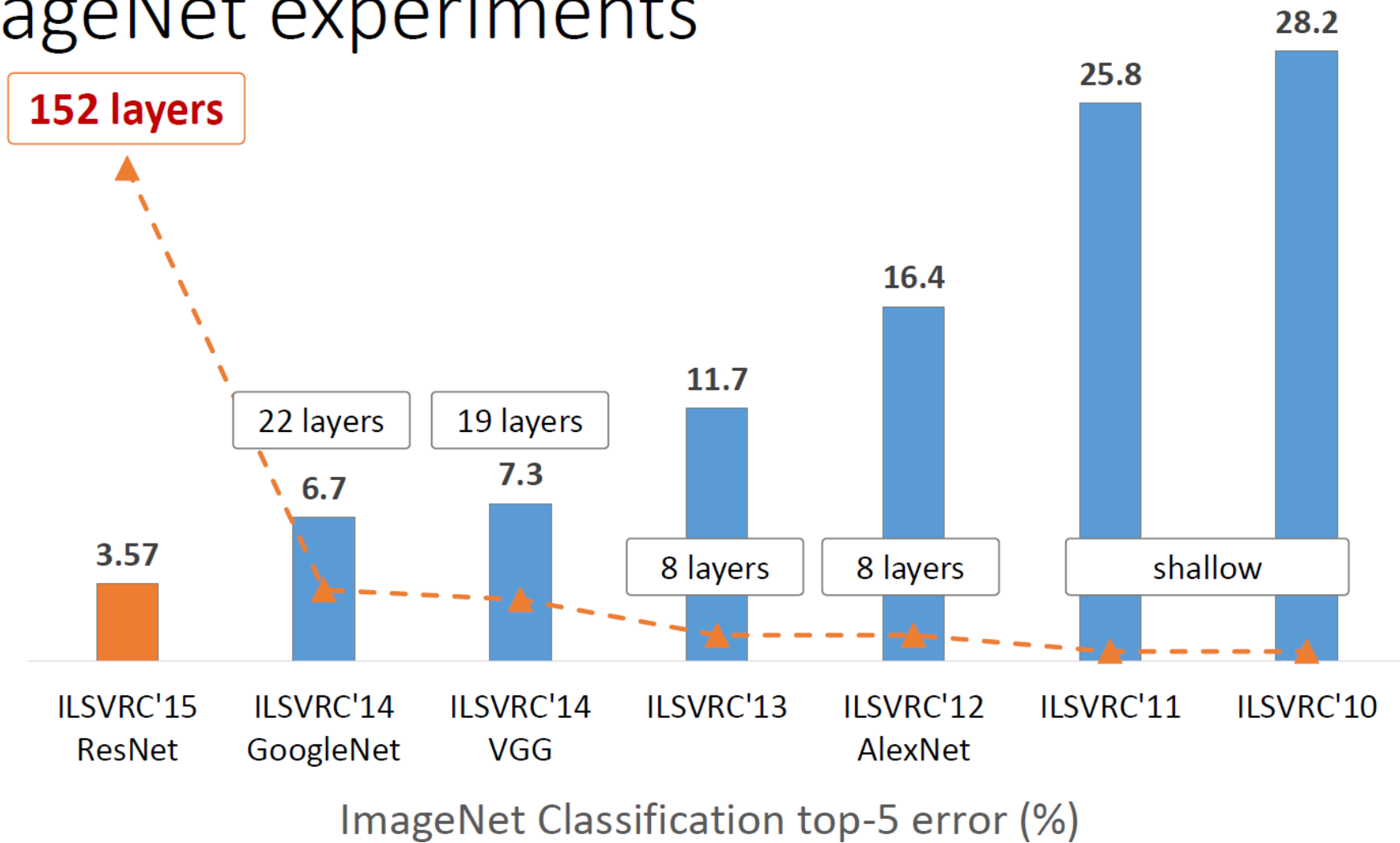


ResNet, 152 layers
(ILSVRC 2015)

AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

ImageNet experiments



Deep Learning Frameworks

- TensorFlow
 - <https://www.tensorflow.org/>
- Theano
 - <http://deeplearning.net/software/theano/>
- Torch
 - <http://torch.ch/>
- Caffe
 - <http://caffe.berkeleyvision.org/>
- CNTK
 - <https://www.cntk.ai/>
- Chainer
 - <http://chainer.org/>

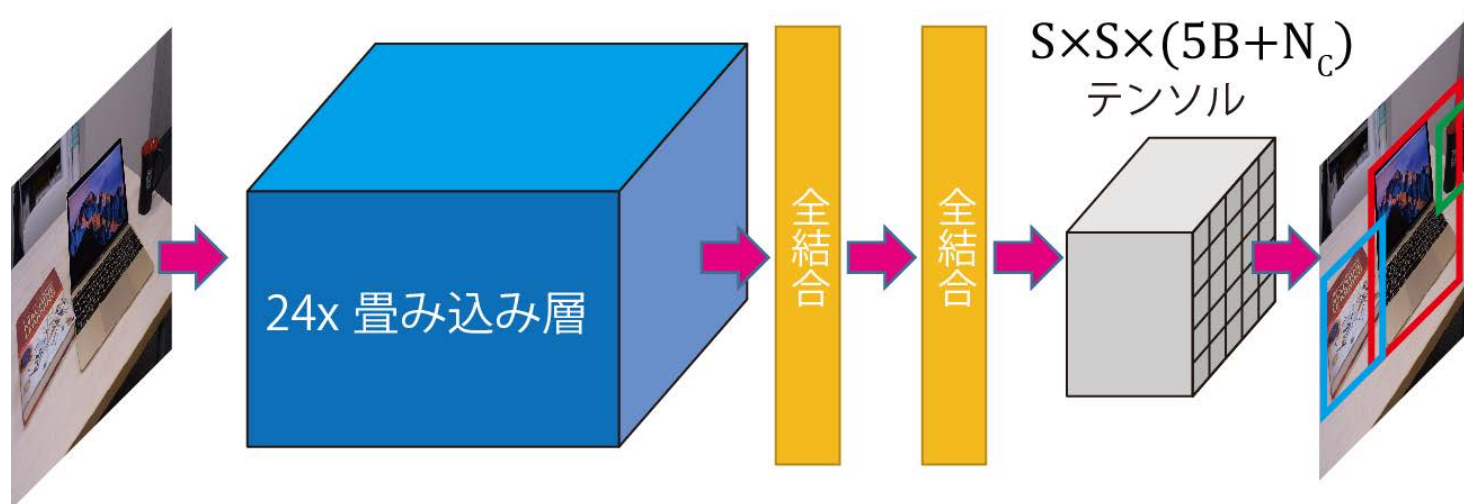
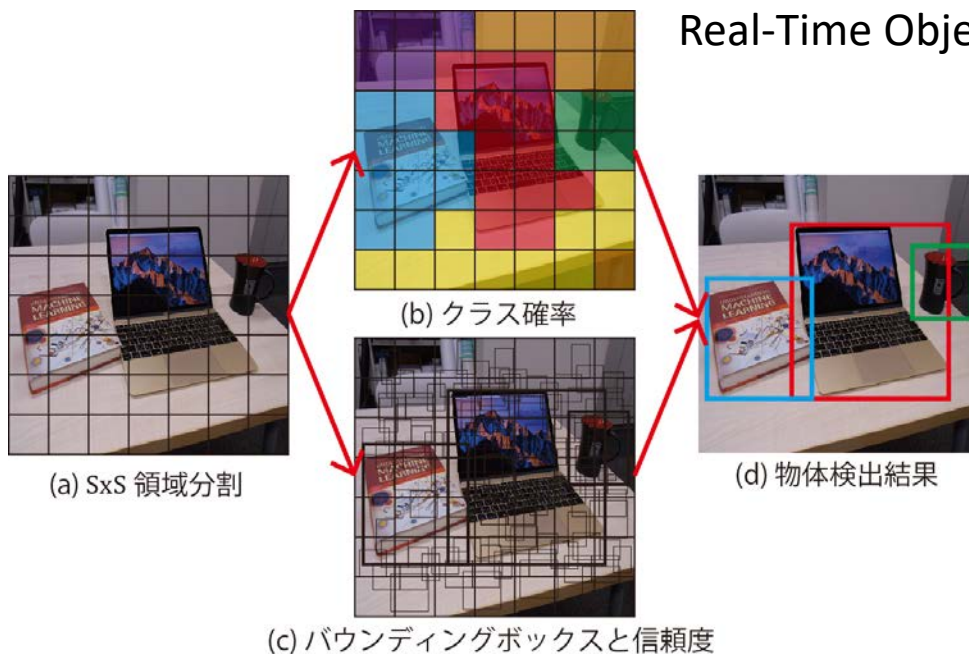
Object Detection

Object Detection



YOLO

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. CVPR, 2016.



YOLO v2



<http://pureddie.com/yolo>

<https://pjreddie.com/darknet/yolo/>

Semantic Segmentation

Semantic Segmentation

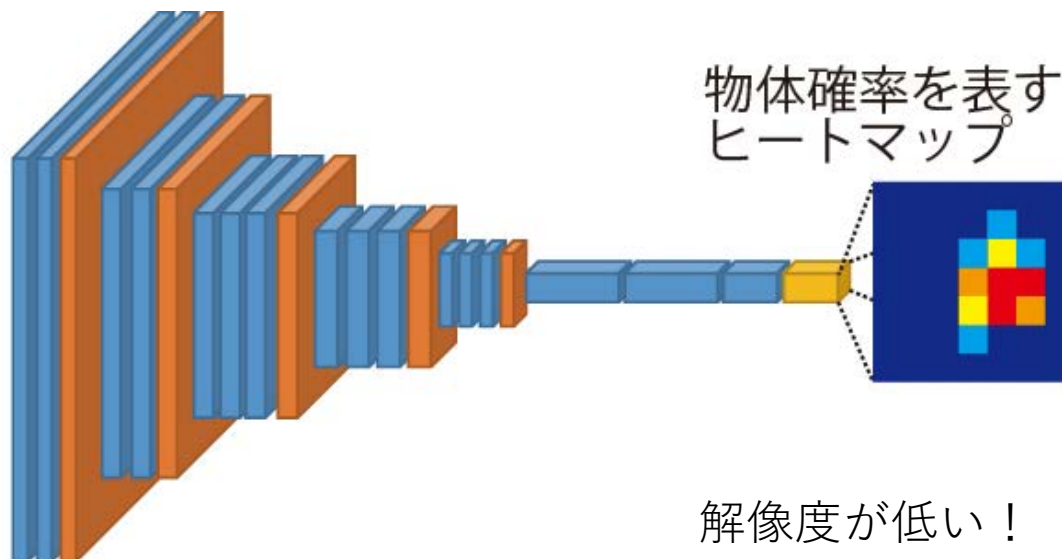
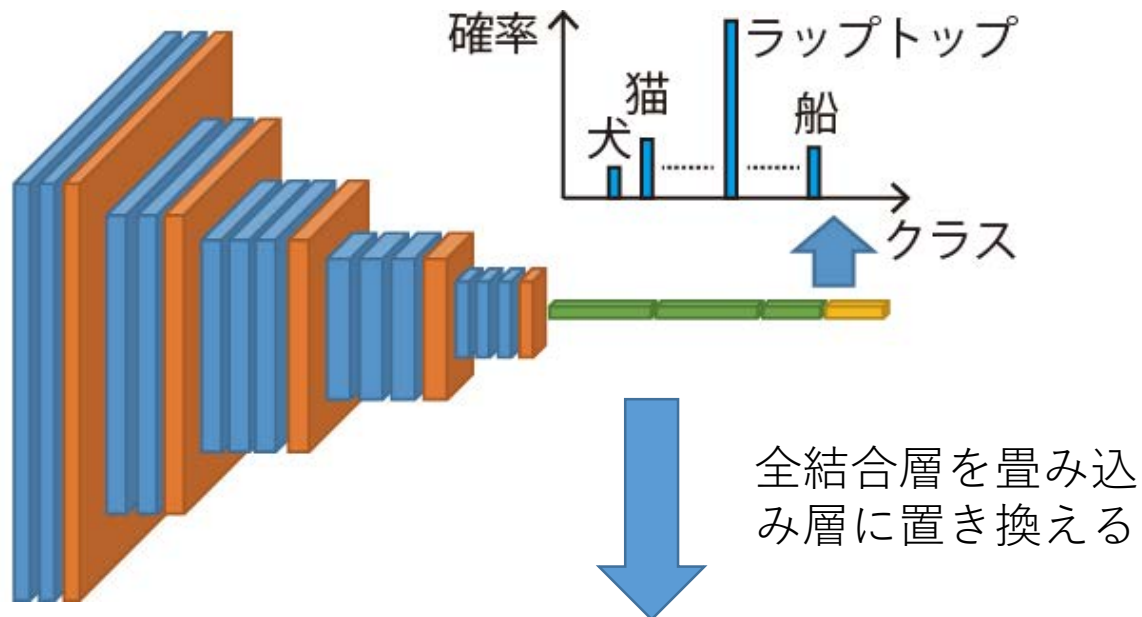


入力画像



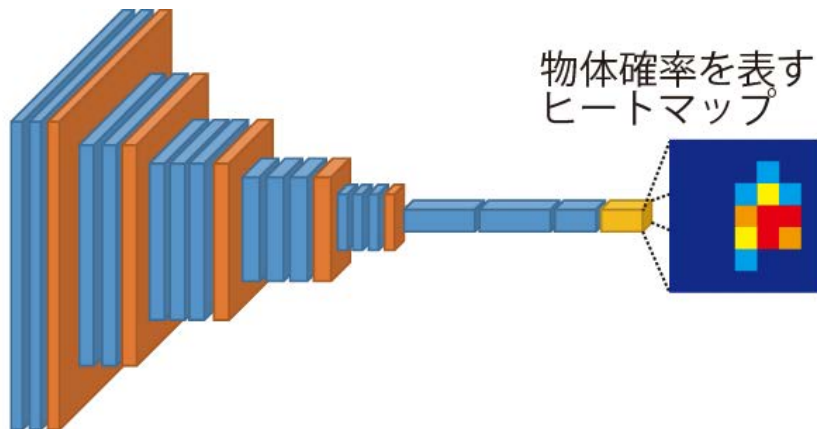
出力結果

Fully Convolutional Network



SegNet

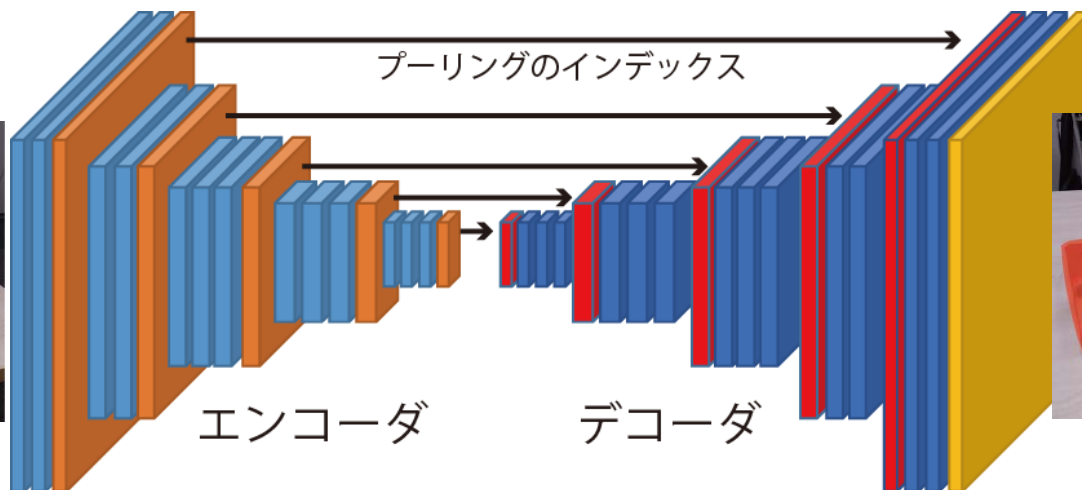
Vijay Badrinarayanan, Ankur Handa and Roberto Cipolla "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling." arXiv preprint arXiv:1505.07293, 2015.



物体確率を表す
ヒートマップ

セマンティクスから形状を
復元するネットワーク

Upsampling



プーリングのインデックス

エンコーダ

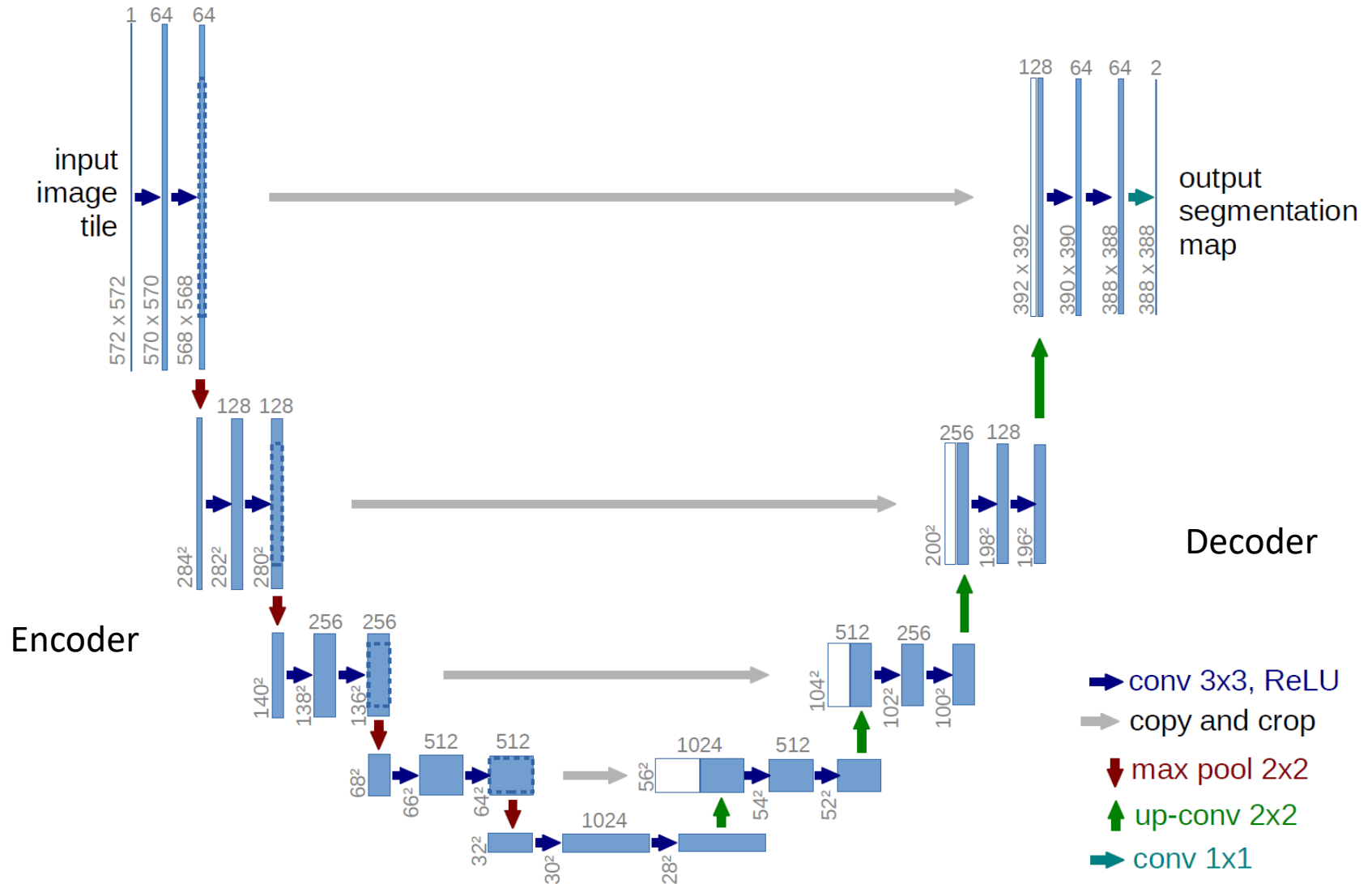
デコーダ



畳み込み層 プーリング層 全結合層 ソフトマックス層 逆プーリング層 転置畳み込み層

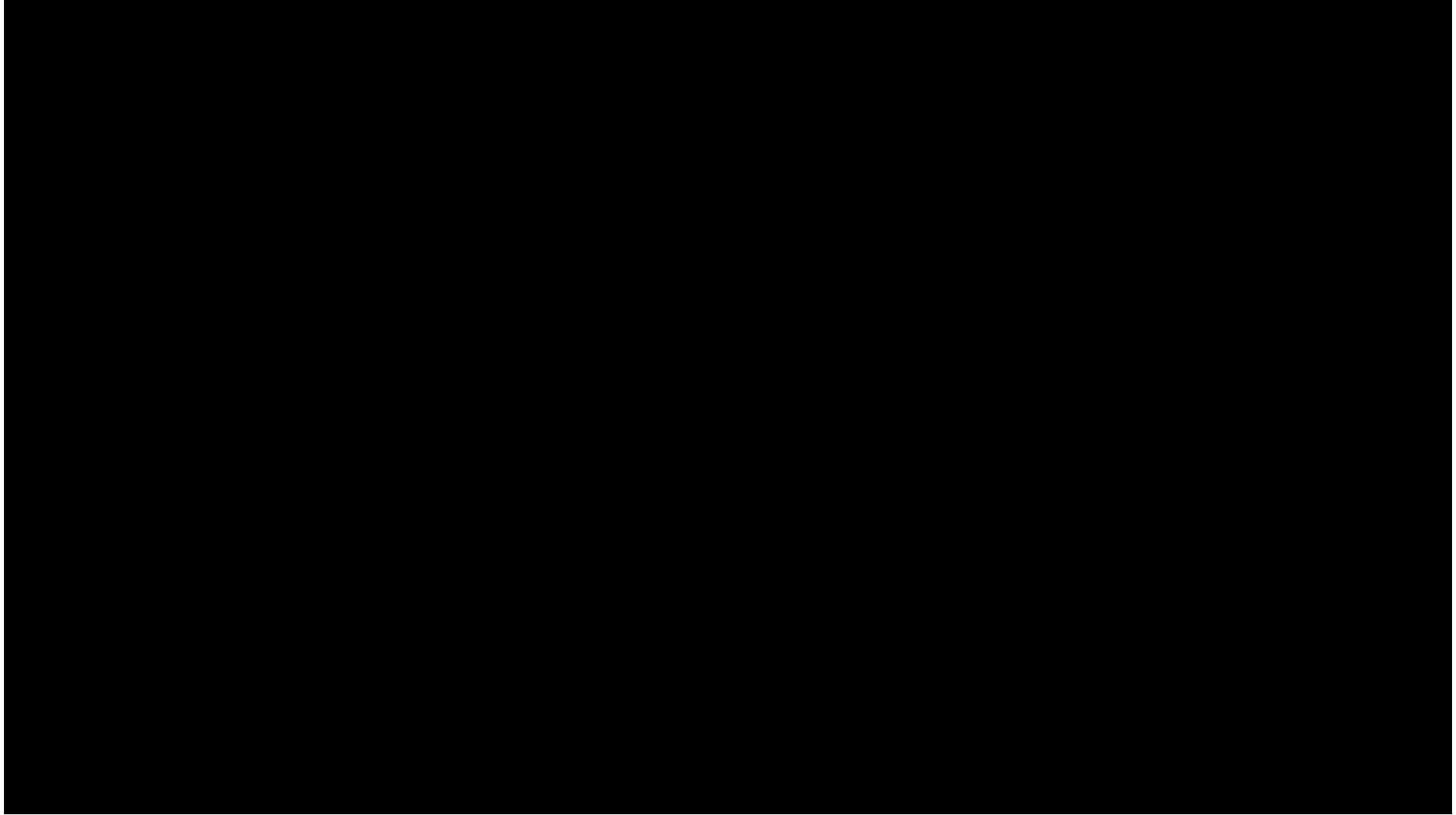
U-Net

Olaf Ronneberger, Philipp Fischer, Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI, Springer, LNCS, Vol.9351: 234--241, 2015.



SegNet demo

<http://mi.eng.cam.ac.uk/projects/segnet/>



Robotic Grasping with Deep Learning

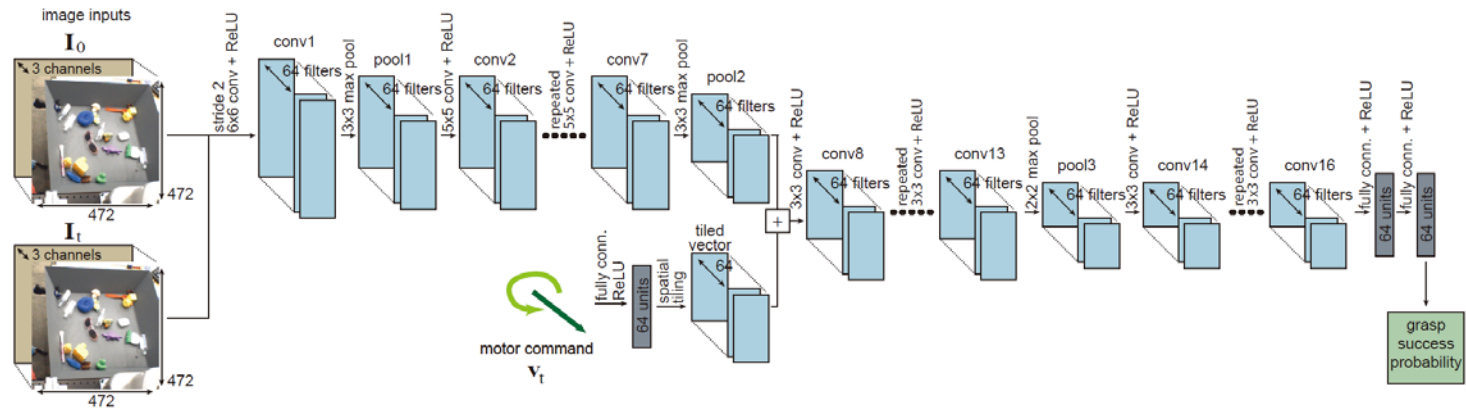


Figure 1. Our large-scale data collection setup, consisting of 14 robotic manipulators. We collected over 800,000 grasp attempts to train the CNN grasp prediction model.

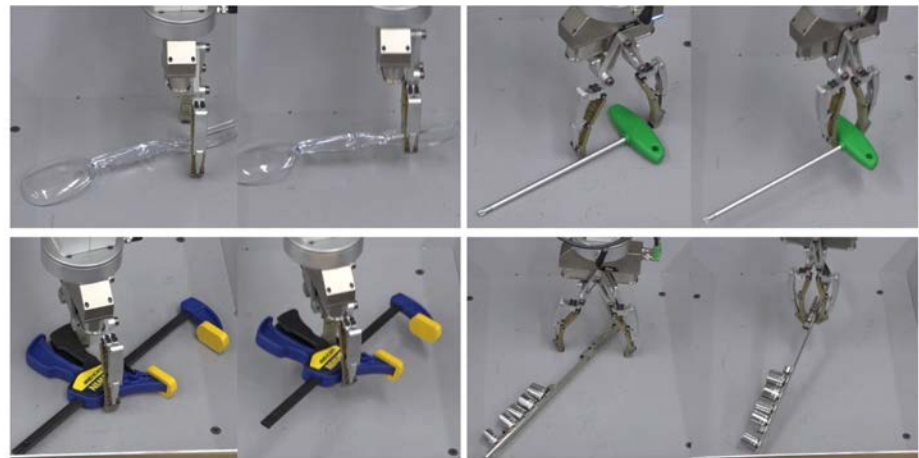


Figure 10. Examples of difficult objects grasped by our algorithm, including objects that are translucent, awkwardly shaped, and heavy.

Sergey Levine, Peter Pastor, Alex Krizhevsky, Deirdre Quillen. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection. arXiv:1603.02199v1 [cs.LG] 7 Mar 2016

Robotic Grasping with Deep Learning

Sergey Levine, Peter Pastor, Alex Krizhevsky, Deirdre Quillen. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection.
arXiv:1603.02199v1 [cs.LG] 7 Mar 2016

