

現場で使える機械学習・データ分析  
基礎講座 (DAY4)  
SkillUP AI

# 本講座の構成

DAY1	DAY2
<ul style="list-style-type: none"><li>• 機械学習概論<ul style="list-style-type: none"><li>• 人工知能とは</li><li>• 機械学習とは</li><li>• 機械学習アルゴリズムの実装とワークフロー</li><li>• 機械学習アルゴリズム概観</li></ul></li><li>• 教師あり学習の基礎<ul style="list-style-type: none"><li>• 線形回帰</li><li>• ロジスティック回帰</li><li>• 多変量モデルへの拡張</li></ul></li><li>• モデルの評価指標<ul style="list-style-type: none"><li>• 回帰問題 (MAE/MSE/RMSE)</li><li>• 分類問題 (精度/適合率/再現率/F1-score)</li></ul></li></ul>	<ul style="list-style-type: none"><li>• モデルの検証・正則化<ul style="list-style-type: none"><li>• 訓練誤差と汎化誤差</li><li>• 過学習</li><li>• 正則化 (L2/L1)</li><li>• ホールドアウト法・交差検証法</li></ul></li><li>• 前処理<ul style="list-style-type: none"><li>• 正規化 / 標準化</li><li>• 無相関化 / 白色化</li></ul></li><li>• 教師あり学習の発展的トピック<ul style="list-style-type: none"><li>• サポートベクターマシン</li></ul></li></ul>

# 本講座の構成

DAY3	DAY4
<ul style="list-style-type: none"><li>前処理<ul style="list-style-type: none"><li>特徴選択</li></ul></li><li>教師あり学習の発展的トピック<ul style="list-style-type: none"><li>木モデル (決定木・ランダムフォレスト)</li><li>ニューラルネットワーク</li></ul></li></ul>	<ul style="list-style-type: none"><li>教師あり学習の発展的トピック<ul style="list-style-type: none"><li>深層学習</li><li>k-最近傍法</li></ul></li><li>教師なし学習<ul style="list-style-type: none"><li>クラスタリング</li><li>特徴抽出・次元削減</li></ul></li><li>モデルの改善<ul style="list-style-type: none"><li>ハイパーパラメータ最適化</li></ul></li></ul>

# DAY4の目次

---

- グループワーク
  - 通し課題の最終発表 (30分)
- 深層学習
  - 置み込みニューラルネットワーク (10分)
  - 置み込みとプーリング(15分)
  - Notebook演習 (20分)
  - 再帰ニューラルネットワーク (15分)
  - Notebook演習 (10分)
  - 生成モデル (5分)
- k-最近傍法 (5分)
- 教師なし学習
  - k-means法 (15分)
  - エルボー法による分割数の設定 (5分)
  - 主成分分析 (15分)
  - 自己符号化器 (15分)
  - Notebook演習 (20分)
- ハイパーパラメータ最適化
  - ランダムサーチ (5分)
  - ベイズ的最適化 (5分)
- グループワーク (30分)
- 質疑応答 (15分)

## 通し課題の最終発表

---

- ・発表2分、質疑応答1分とします
- ・データの可視化を行った結果、わかったことはありますか？
- ・モデルや前処理、パラメータチューニングなど、今回の通し課題に対して工夫した点はどこですか？
- ・積極的に質問をしましょう！

# 深層学習

1. 置み込みニューラルネットワーク
2. 置み込みとプーリング
3. 再帰型ニューラルネットワーク
4. 生成モデル

## 畳み込みニューラルネットワーク / 畳み込みとプーリング

# 深層学習の様々なモデル

確定的

階層型

全結合型ニューラルネットワーク  
Fully connected neural network

畳み込みニューラルネットワーク  
Convolutional neural network

再帰型ニューラルネットワーク  
Recurrent neural network

自己符号化器型

自己符号化器  
Autoencoder

雑音除去自己符号化器  
Denoising Autoencoder

変分自己符号化器  
Variational Autoencoder

スパース自己符号化器  
Sparse Autoencoder

確率的

ボルツマンマシン型

ボルツマンマシン  
Boltzmann machine

制約ボルツマンマシン  
Restricted Boltzmann machine

深層信念ネットワーク  
Deep belief network

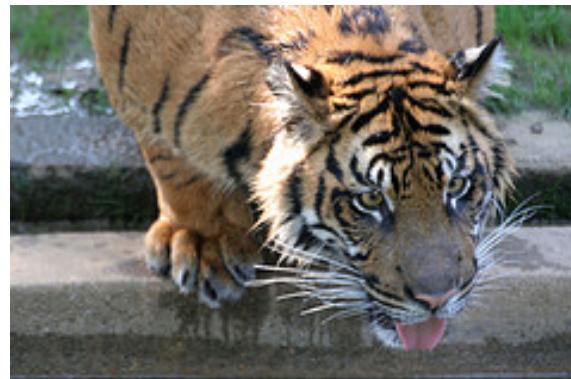
深層ボルツマンマシン  
Deep Boltzmann machine

参考：『深層学習』(神鳶ら、近代科学社)

# ニューラルネットワークの成功の理由は何だろう？

---

- ・深層学習に関するニュースでは、「深層学習は画像認識の分野で、大きな成功をした」という記述をよく見かける
- ・ニューラルネットワークの成功には、どのような工夫があったのか？
- ・画像を例に考えてみよう
  - ・画像に特有であり、かつ、有効そうな特徴量とは何だろう？
  - ・それはどのように処理されて得られるだろうか？



この画像から「トラ」であることを認識するためにはどのような特徴が必要？

# 畳み込みをニューラルネットワークに組み入れる

---

- 画像認識で大きな成功を収めたニューラルネットワークは、畳み込みの計算方法を取り入れることで画像に効果的な特徴を抽出している
- 畳み込みに基づいたデータ処理を取り入れたニューラルネットワークを  
**畳み込みニューラルネットワーク (Convolutional Neural Network; CNN)**  
と呼ぶ
- 畳み込みに使うフィルター（カーネルと呼ばれることがある）の値が  
学習すべきパラメータ

# フィルタ処理と畳み込み：画像におけるデータ処理方法

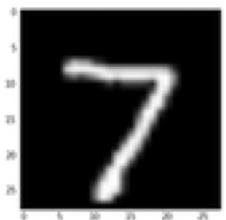
- たとえば、画像のエッジ（輪郭）は有効な特徴な1つであると考えられる
- エッジの抽出は、畳み込みと呼ばれる演算で実現される
- 畳み込みは画像処理でよく用いられる計算方法

$$\begin{matrix} 1 & 2 & 3 & 0 \\ 0 & 1 & 2 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 3 & 0 & 1 \end{matrix} \times \begin{matrix} \text{フィルター} \\ \begin{matrix} 2 & 0 & 1 \\ 0 & 1 & 2 \\ 1 & 0 & 2 \end{matrix} \end{matrix} \rightarrow \begin{matrix} 15 & 16 \\ 6 & 15 \end{matrix}$$

✖ : 畳み込み演算



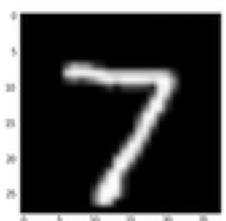
# 画像認識における従来手法とCNNの違い



→ 人の考えたアルゴリズム → 予測値



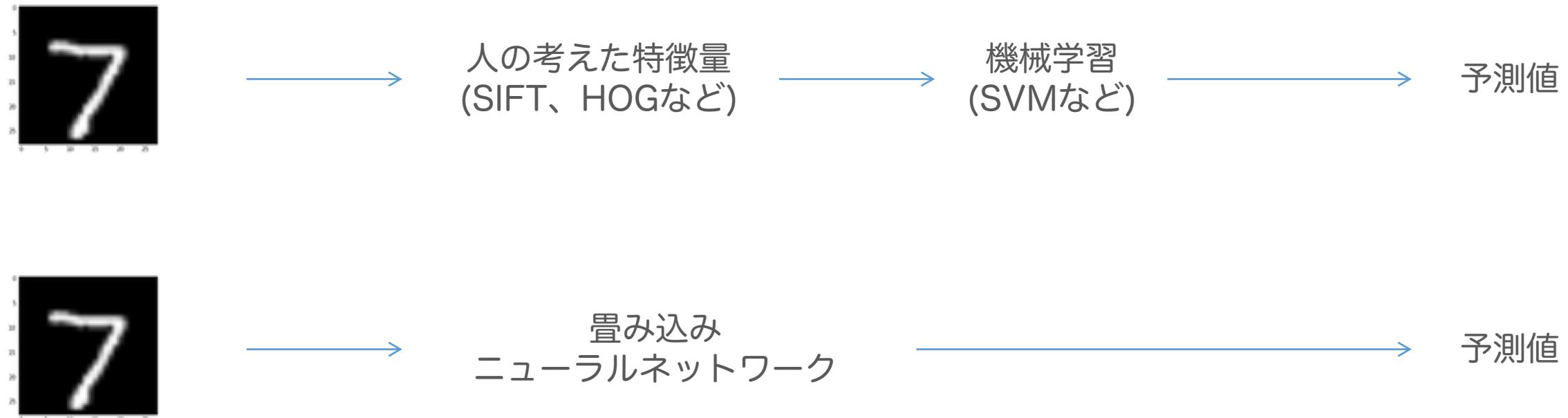
→ 人の考えた特徴量  
(SIFT、HOGなど) → 機械学習  
(SVMなど) → 予測値



→ 置み込み  
ニューラルネットワーク → 予測値

# 画像認識における従来手法とCNNの違い

- ・畠み込みニューラルネットワークは、他の機械学習手法と異なり、特徴量の設計を自動で行っているものと見ることができる



# 深層学習の様々なモデル

確定的

階層型

全結合型ニューラルネットワーク  
Fully connected neural network

畳み込みニューラルネットワーク  
Convolutional neural network

再帰型ニューラルネットワーク  
Recurrent neural network

自己符号化器型

自己符号化器  
Autoencoder

雑音除去自己符号化器  
Denoising Autoencoder

変分自己符号化器  
Variational Autoencoder

スパース自己符号化器  
Sparse Autoencoder

確率的

ボルツマンマシン型

ボルツマンマシン  
Boltzmann machine

制約ボルツマンマシン  
Restricted Boltzmann machine

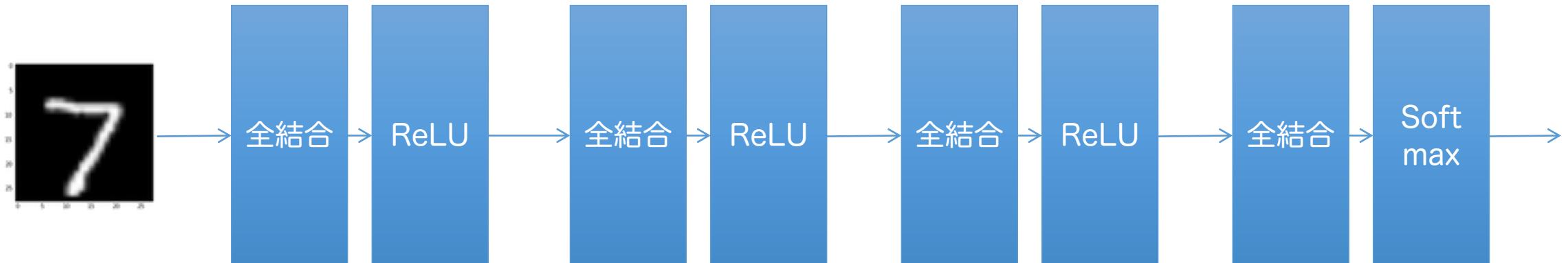
深層信念ネットワーク  
Deep belief network

深層ボルツマンマシン  
Deep Boltzmann machine

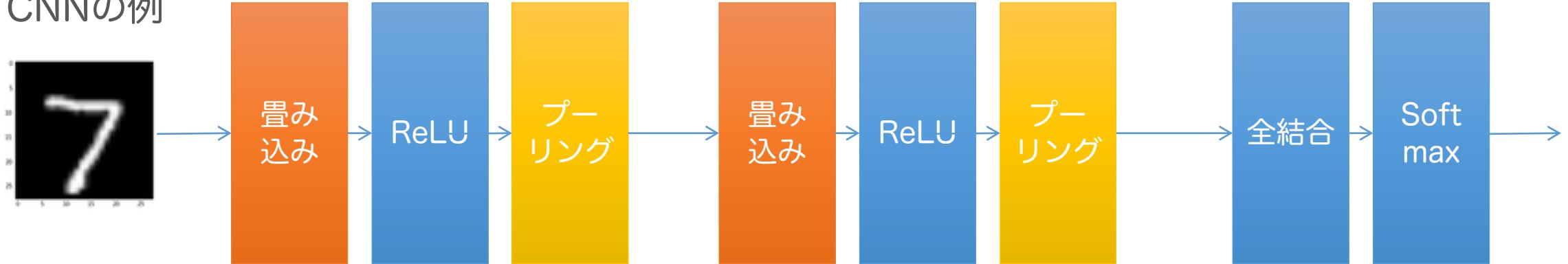
参考：『深層学習』(神鳶ら、近代科学社)

# 全結合NN（今までのニューラルネットワーク）とCNNの違い

全結合NNの例

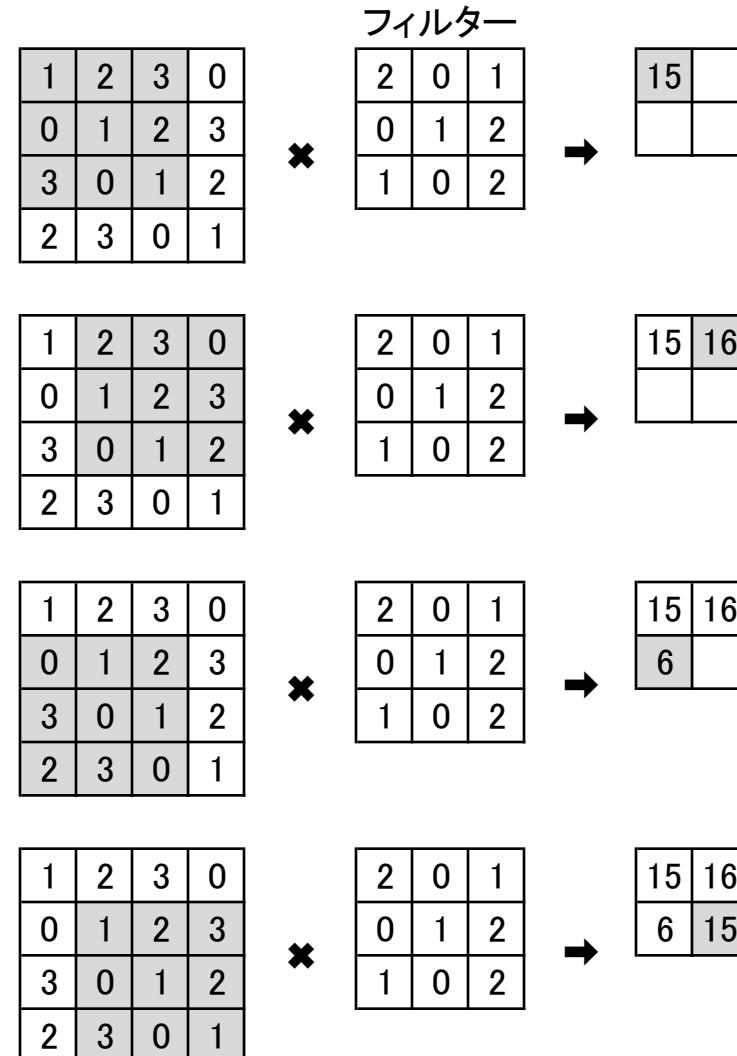


CNNの例



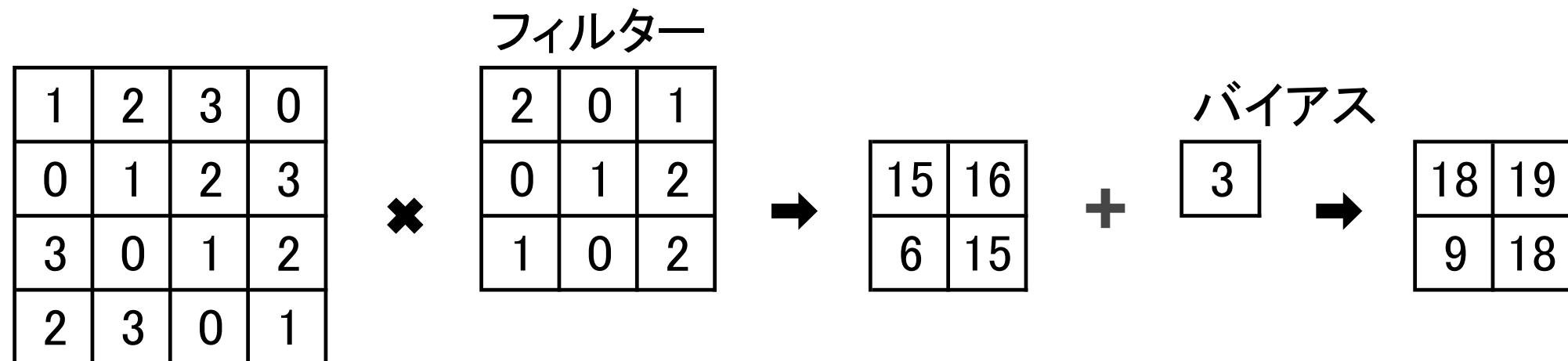
# 畳み込み層

- ・ 畳み込み層は画像の一部に対して注目して処理を行う
  - ・ エッジ抽出フィルターなどの画像処理と同じ計算方法
- ・ フィルターの値を誤差逆伝播法で学習



## 畳み込み層におけるバイアス

- ・畠み込み層にも、バイアス項を導入することができる
- ・処理された画像に決まった値を足し込むことでバイアスとする



## 畳み込み層の弱点？

---

- 以下の2つの画像について、値が大きい画素が重要な部分であるものとする
- しかし、位置がズれてしまっているため、このままでは畳み込みの結果が位置によって左右されてしまう
- 位置に依らない、つまり、認識したい物体の移動に対して剛健にするにはどうしたら良いだろうか？

1	2	0	7	1	0
0	9	2	3	2	3
3	0	1	2	1	2
2	4	0	1	0	1
6	0	1	2	1	2
2	4	0	1	8	1

1	1	2	0	7	1
3	0	9	2	3	2
2	3	0	1	2	1
3	2	4	0	1	0
2	6	0	1	2	1
1	2	4	0	1	8

# プーリング演算

- 画像の一部分の最大値や平均値をとる操作をプーリングと呼ぶ
- CNNではプーリングを用いることで、位置に対する不变性を獲得している
- 以下はサイズ $2\times 2$ の最大値プーリング (Max Pooling) の例

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1

A 2x2 grid with the top-left cell containing the number 2 and the other three cells empty.

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1

A 2x2 grid with the top-left cell containing the number 2, the cell to its right containing 3, and the bottom two cells empty.

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1

A 2x2 grid with the top-left cell containing the number 2, the cell to its right containing 3, the bottom-left cell containing 4, and the bottom-right cell empty.

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1

A 2x2 grid with the top-left cell containing the number 2, the cell to its right containing 3, the bottom-left cell containing 4, the bottom-right cell containing 2, and the bottom cell empty.

# プーリング演算の意義

- 先程の例について、プーリングを適用することで位置不变となるかを確認してみよう
- たとえばサイズ3x3の最大値プーリングを使ってみると、プーリングの結果は同じものが得られる
  - つまり、位置によって出力が変化しないことがわかる

1	2	0	7	1	0
0	9	2	3	2	3
3	0	1	2	1	2
2	4	0	1	0	1
6	0	1	2	1	2
2	4	0	1	8	1



9	7
6	8

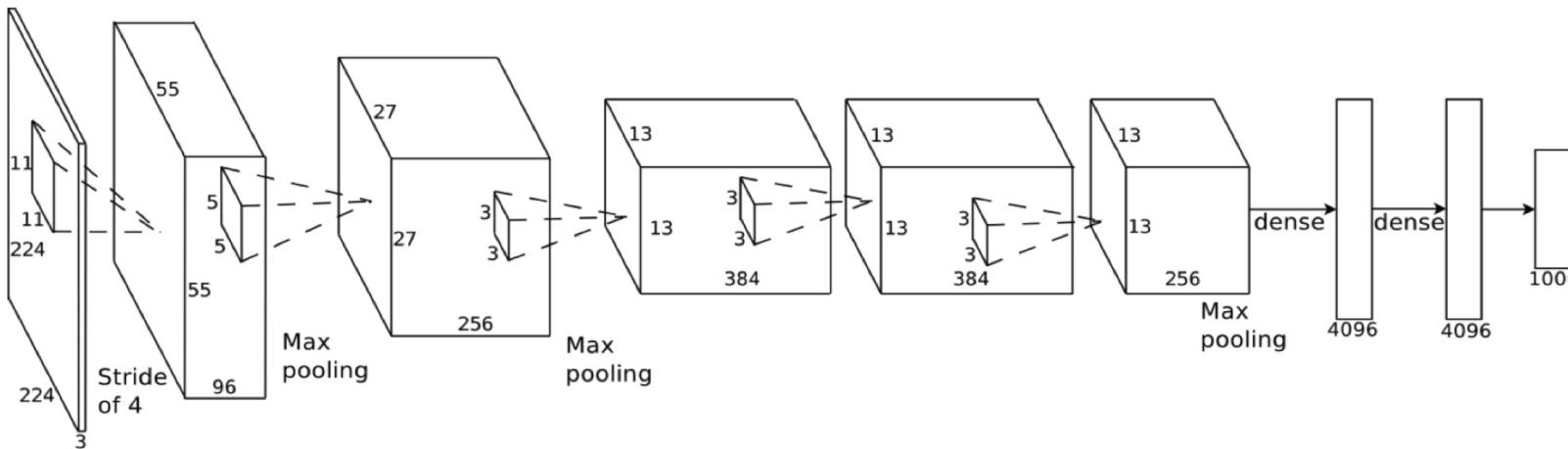
1	1	2	0	7	1
3	0	9	2	3	2
2	3	0	1	2	1
3	2	4	0	1	0
2	6	0	1	2	1
1	2	4	0	1	8



9	7
6	8

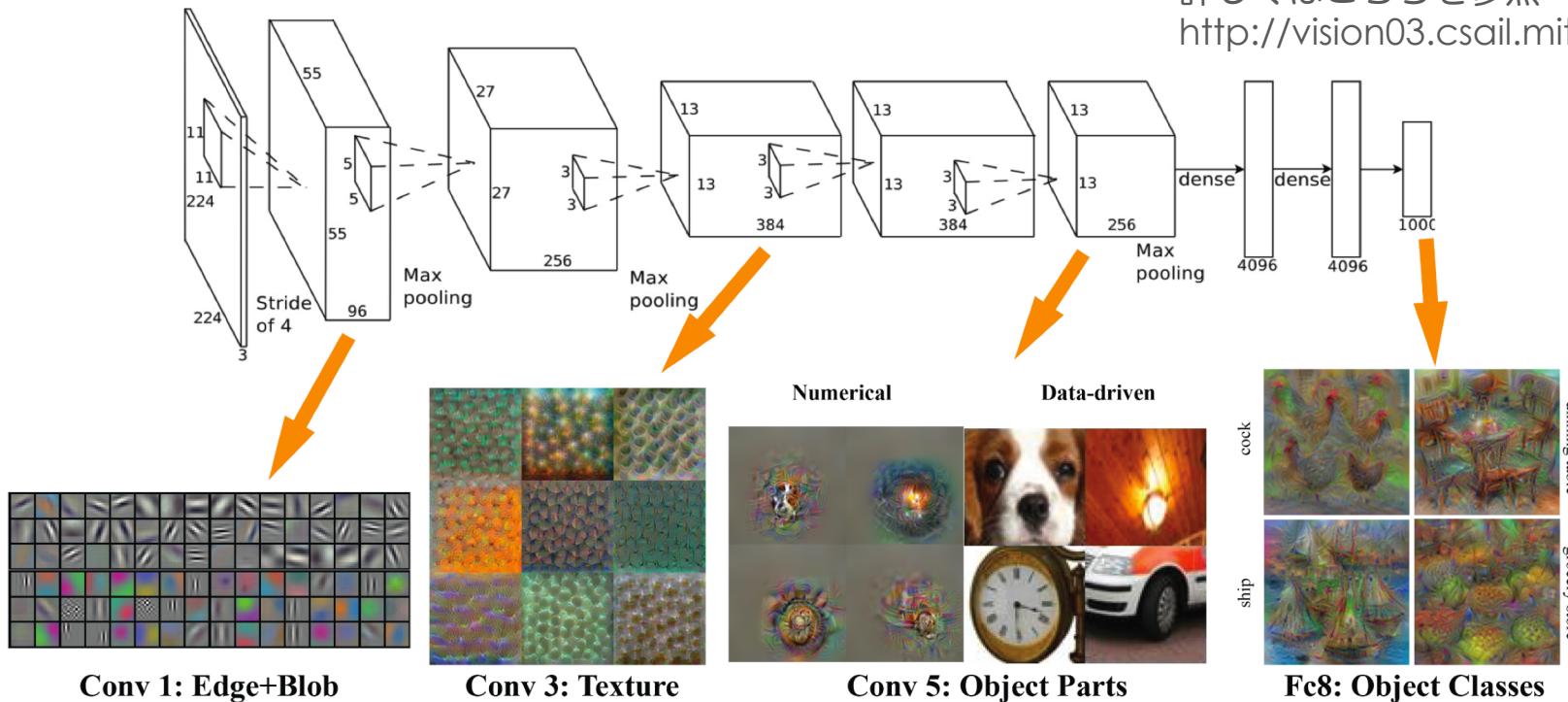
# CNNで抽出された特徴とは？

- CNNでは、どのような特徴が抽出されているのだろうか？
- CNNでは、複数の畳み込み層を積み重ねているが、それぞれの畳み込み層で抽出されている特徴はどのようにになっているのだろうか？
- AlexNetと呼ばれる8層のCNNで、抽出された特徴を覗いてみよう



# CNNで抽出された特徴とは？

- ・入力側の1層目 (Conv1) は、輪郭 (edge) や塊 (blob) に関する特徴
- ・3層目 (Conv3) は、質感 (texture) に関する特徴
- ・5層目 (Conv5) は、物体の部品に関する特徴
- ・最終層 (Fc8) では、物体のクラスを出力

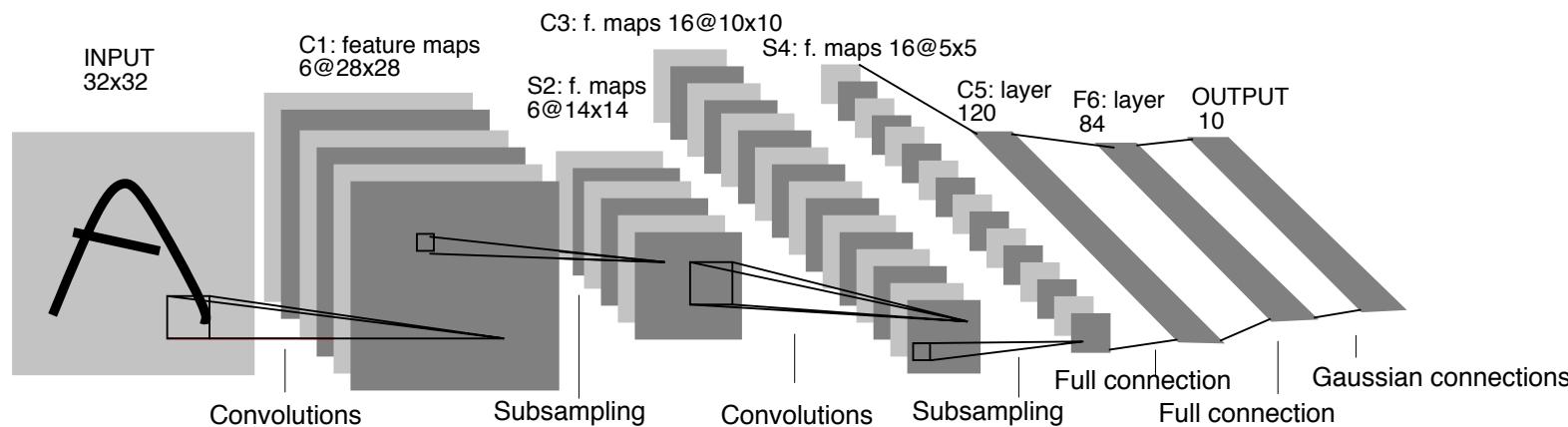


詳しくはこちらを参照：

[http://vision03.csail.mit.edu/cnn\\_art/index.html](http://vision03.csail.mit.edu/cnn_art/index.html)

# LeNet-5 | 5層の畳み込み層を持つCNN

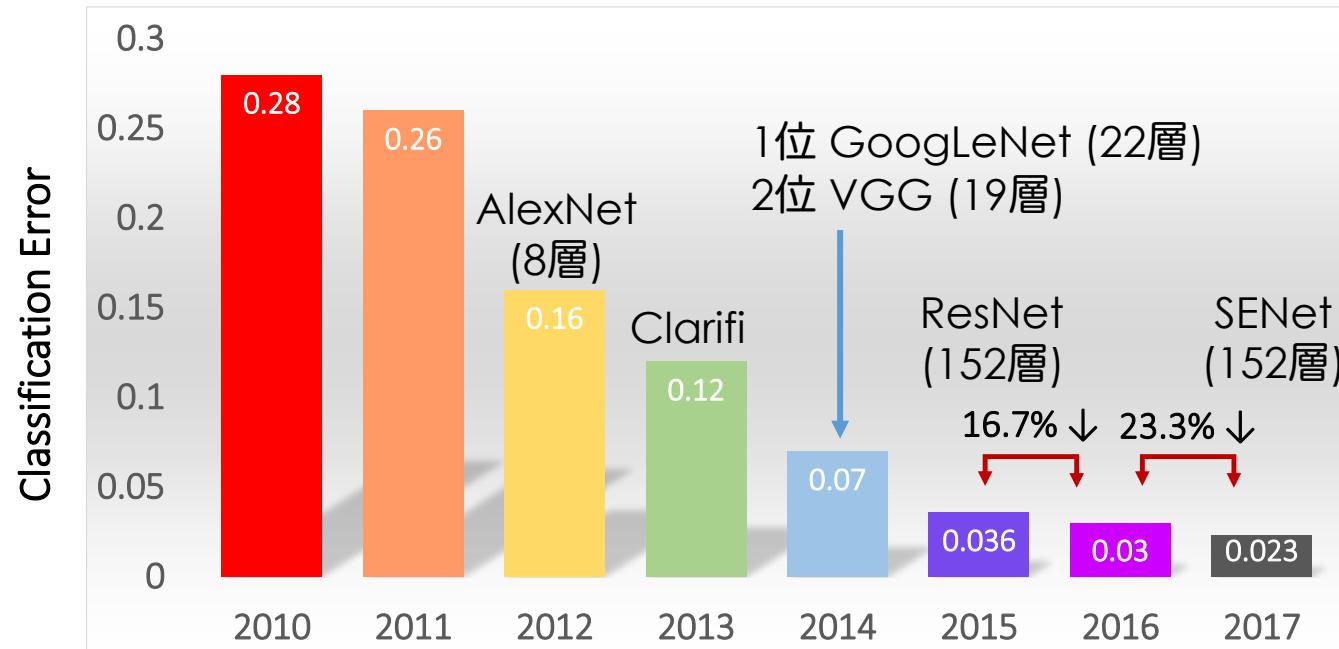
- 1998年ごろヤン・ルカンらによって提案されたCNN
  - 1層目：サイズ $5 \times 5$ のフィルタを、6個持つ畳み込み層
  - 2層目： $2 \times 2$ 画素領域にサブサンプリングを行うプーリング層
  - 3層目：サイズ $6 \times 6$ のフィルタを16個持つ畳み込み層
  - 4層目： $2 \times 2$ の画素領域にサブサンプリングを行うプーリング層
  - 5層目：サイズ $6 \times 6$ のフィルタを120個持つ畳み込み層
  - 最終層（6層目）：全結合層



引用：  
[http://vision.stanford.edu/cs598\\_spring07/papers/Lecun98.pdf](http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf)

# ILSVRC | Image net Large Scale Visual Recognition Challenge

- 2010年から始まった大規模なデータセットを用いた画像認識のコンテスト
- 2012年にはじめてCNNを用いた手法であるAlexNetが優勝し、深層学習ブームのきっかけとなった
  - ILSVRCの動向はこちらの解説論文を参照：  
<http://www.nlab.ci.i.u-tokyo.ac.jp/pdf/ieice201705cvcomp.pdf>

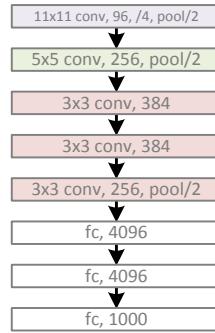


引用：[http://image-net.org/challenges/talks\\_2017/ILSVRC2017\\_overview.pdf](http://image-net.org/challenges/talks_2017/ILSVRC2017_overview.pdf)

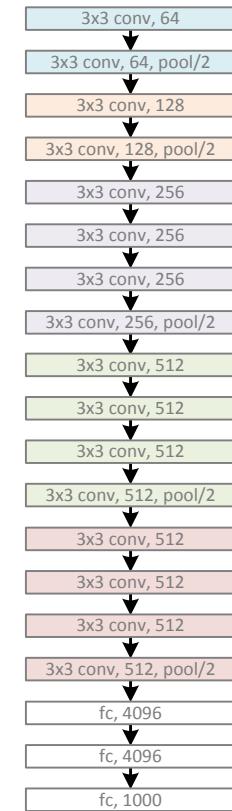
# 著名なCNNの比較 (2012年～2014年)

- ILSVRC優勝モデルは2012年では8層、2014年では22層まで深くなっている

AlexNet, 8 layers  
(ILSVRC 2012)



VGG, 19 layers  
(ILSVRC 2014)



GoogleNet, 22 layers  
(ILSVRC 2014)



引用 : [http://image-net.org/challenges/talks/ilsvrc2015\\_deep\\_residual\\_learning\\_kaiminghe.pdf](http://image-net.org/challenges/talks/ilsvrc2015_deep_residual_learning_kaiminghe.pdf)

# 著名なCNNの比較 (2012年~2015年)

- なんと2015年のILSVRC優勝モデルは152層！
- 今後は更に層数が多く増えていく可能性も…

AlexNet, 8 layers  
(ILSVRC 2012)



VGG, 19 layers  
(ILSVRC 2014)



ResNet, **152 layers**  
(ILSVRC 2015)

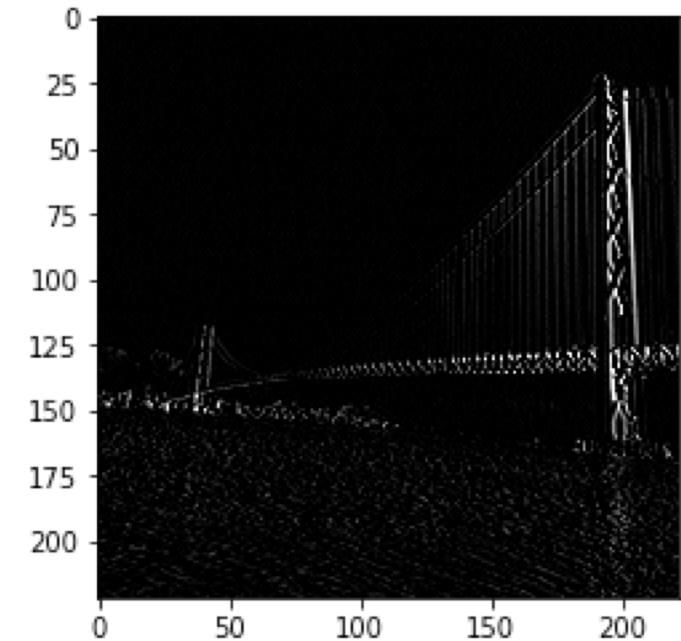
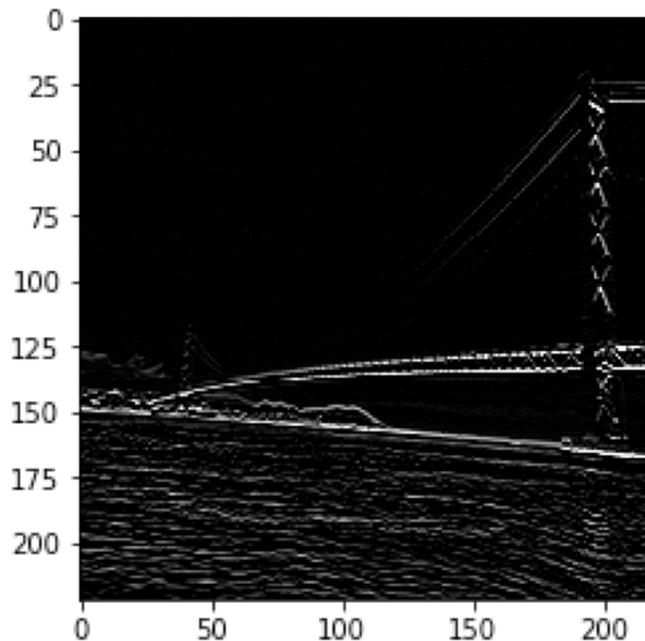


引用 : [http://image-net.org/challenges/talks/ilsvrc2015\\_deep\\_residual\\_learning\\_kaiminghe.pdf](http://image-net.org/challenges/talks/ilsvrc2015_deep_residual_learning_kaiminghe.pdf)

## [演習] 1\_convolution.ipynb

---

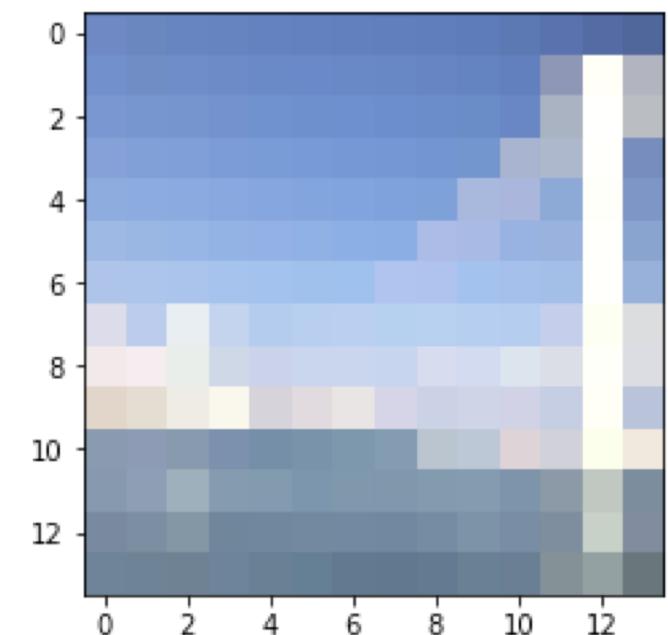
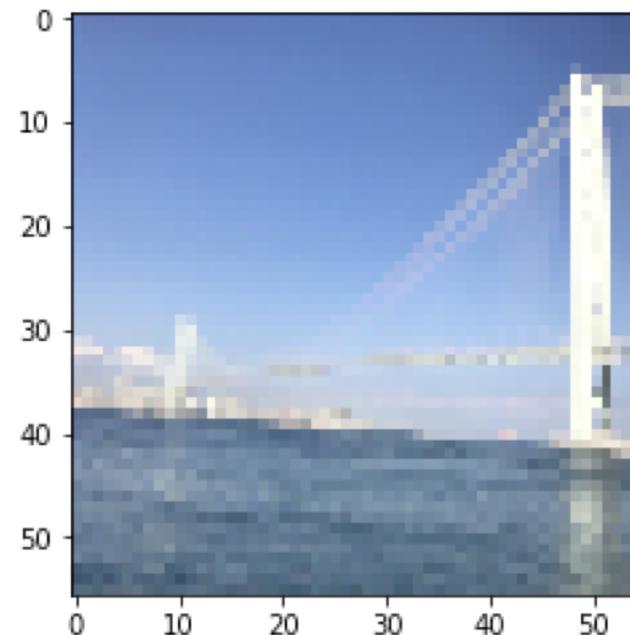
- 畠み込み演算の振る舞いを確認してみましょう



## [演習] 2\_pooling.ipynb

---

- ・プーリングの振る舞いを確認してみましょう



## [演習] 3\_CNN.ipynb

---

- CNNは学習済みモデルが多く配布されており、比較的簡単に利用することができます
- Inception-v3の学習済みモデルを用いて、画像の識別を行ってみましょう
  - Inception : GoogLeNetをベースに改良を重ねたモデル

## 再帰型ニューラルネットワーク

## 「時系列」の関係をニューラルネットワークに組み込もう

---

- CNNは画像に適した処理方法である、畳み込みを組み込むことで効果的な特徴を抽出していた
- 次は自然言語や音声などの時系列データを考えてみよう
- ある時点におけるデータは、以前のデータと何らかの関係を持つはず
  - 自然言語であれば、ある時点の単語はその前に出た単語と関係があるはず
- 以前のデータとの関係も使って、予測値を決めるにはどのようにすればよいだろうか？

- ・ 時間方向に状態を引き継ぎながら計算を進めることができるニューラルネットワーク
- ・ 自然言語の単語など順番に並んでいるデータを自然に扱うことができるため、自然言語処理や音声認識に利用される
- ・ 時間方向に計算するネットワークであるため、「時系列数値データをディープラーニングする」 = 「RNN」とよく言われるが、自己回帰モデル自体は、通常の全結合型ニューラルネットワークでもモデル化できる
- ・ 自然言語のように入力長が毎回異なるデータを扱う場合は、RNNが向いている

# 深層学習の様々なモデル

確定的

階層型

全結合型ニューラルネットワーク  
Fully connected neural network

畳み込みニューラルネットワーク  
Convolutional neural network

再帰型ニューラルネットワーク  
Recurrent neural network

自己符号化器型

自己符号化器  
Autoencoder

雑音除去自己符号化器  
Denoising Autoencoder

変分自己符号化器  
Variational Autoencoder

スパース自己符号化器  
Sparse Autoencoder

確率的

ボルツマンマシン型

ボルツマンマシン  
Boltzmann machine

制約ボルツマンマシン  
Restricted Boltzmann machine

深層信念ネットワーク  
Deep belief network

深層ボルツマンマシン  
Deep Boltzmann machine

参考：『深層学習』(神鳶ら、近代科学社)

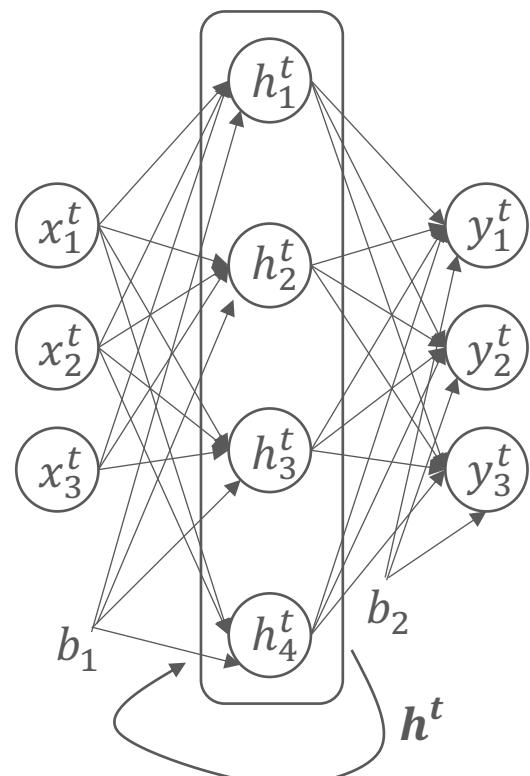
# 再帰型ニューラルネットワークの種類

---

- シンプルなRNN
  - 単純な全結合層を用いて状態を更新
  - 長い系列を学習しても、直近の情報を重視するため、過去の情報があまり反映されない
- LSTM (Long Short Term Memory unit)
  - シンプルなRNNの欠点を解決したRNN
  - ゲートと呼ばれる仕組みを複数導入することで、過去の情報を「忘れるか忘れないか」を判断しながら、必要な情報だけ次の状態に引き継ぐ
- GRU (Gated Recurrent Unit)
  - LSTMをシンプルにしたRNNで、LSTMよりもパラメータの数が少ない
  - リセットゲート、更新ゲートでのみ構成

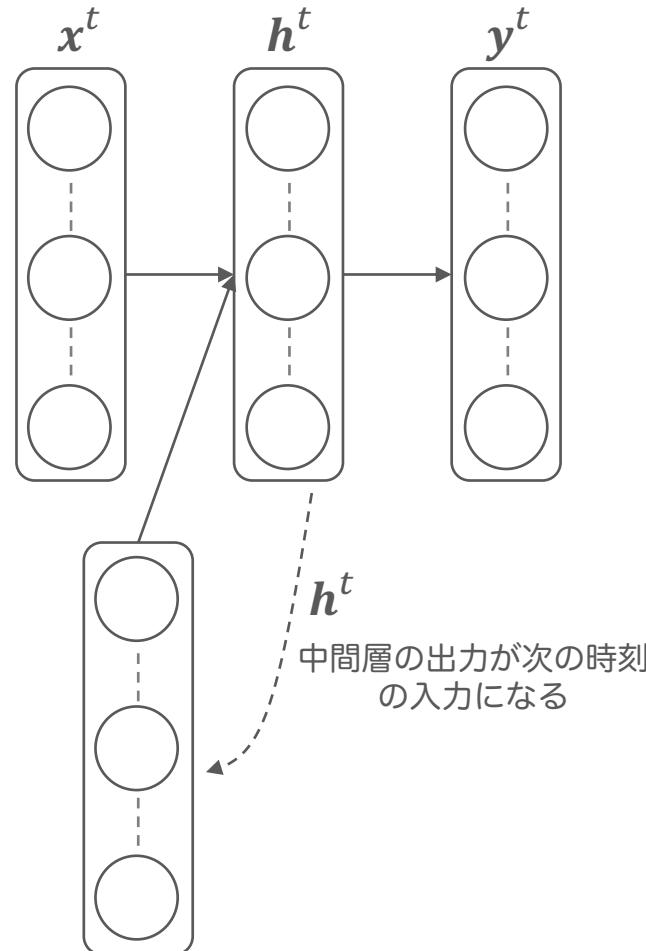
# シンプルなRNNのネットワーク

- RNNは中間層の出力が次の時刻の入力となる再帰的な接続を持つ



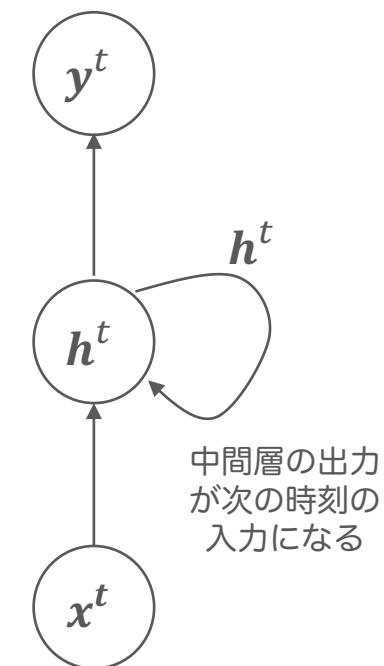
中間層の出力が次の時刻  
の中間層への入力になる

簡略化する  
→



中間層の出力が次の時刻  
の入力になる

もっと  
簡略化する  
→



中間層の出力  
が次の時刻  
の入力になる

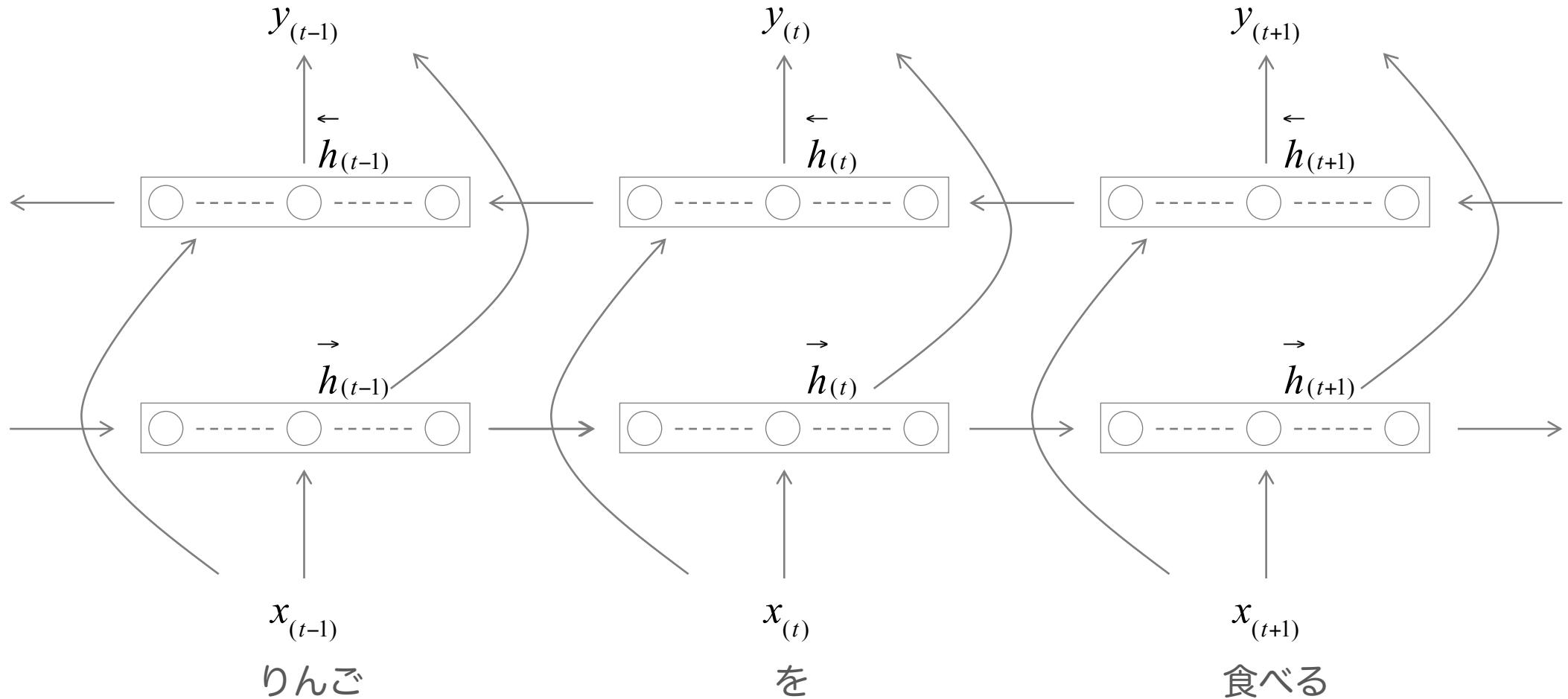
# RNNの応用

---

- Bi-directional RNN
  - 過去から未来だけでなく、未来から過去への方向も考慮したRNN
  - 未来の予測ではなく、すでに手元にあるデータに対して、何らかの予測を行う場合は、過去から未来だけでなく、未来から過去への方向も考慮した方が良い精度が期待できるはずというのがコンセプト
- RNN Encoder-Decoder
  - 入力も出力もシーケンス(sequence)になっているモデル
  - Sequence-to-sequence modelとも呼ばれる

# Bi-directional RNNの構造

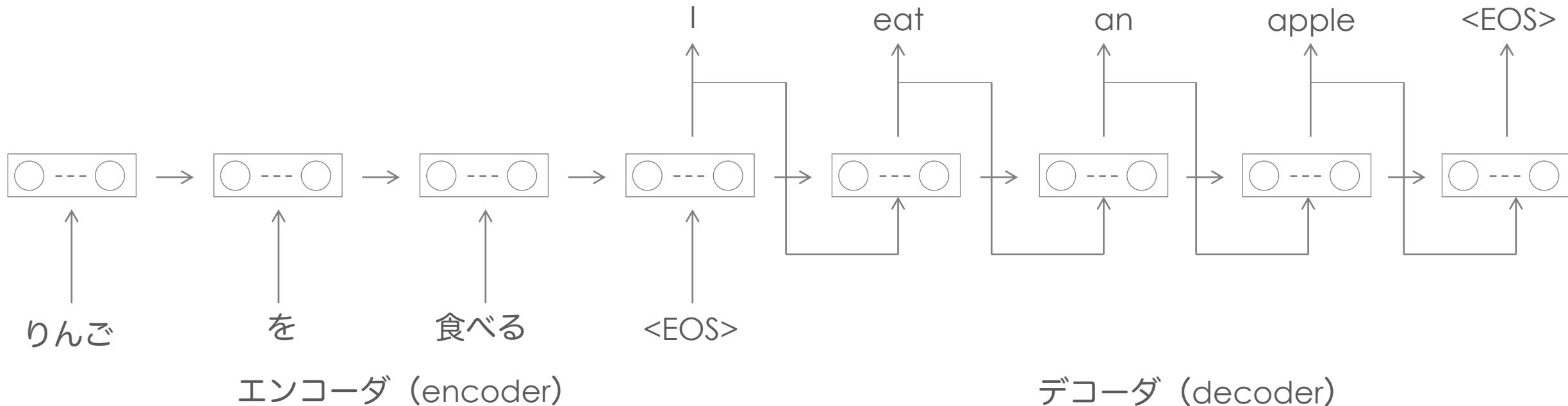
Bi-directional RNNの例



# RNN Encoder-Decoderの構造 | 日一英 翻訳モデルを例に

- 入力の系列長（例：単語数）と出力の系列長は一致していなくてよい
- 系列長は、データごとに異なっていてもよい

RNN Encoder-Decoderの例



## 生成モデルとは

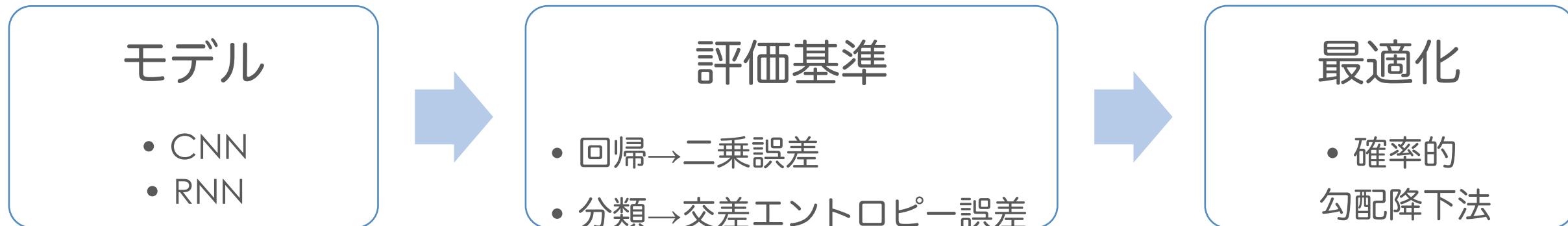
- ・「観測されたデータは、ある確率分布から生成されたサンプルである」と仮定し、データを生成する確率分布をモデリングする方法
  - ・ニューラルネットワークを用いた例：敵対的生成ネットワーク（GAN）
  - ・他にもガウス混合モデル、ナイーブベイズモデルなども生成モデルのひとつ
- ・入力の分布がわかると、データのシミュレートができるようになる
  - ・応用例：欠けた画像の自動補完、超解像処理など



引用：<http://hi.cs.waseda.ac.jp/~iizuka/projects/completion/ja/>

# 深層学習まとめ（モデル、評価基準、最適化の観点から）

- CNN：畳み込み層とプーリング層を取り入れた、主に入力データが画像であるときに効果的なニューラルネットワーク
- RNN：再帰的な接続を取り入れた、入力データが時系列データであるときに効果的なニューラルネットワーク

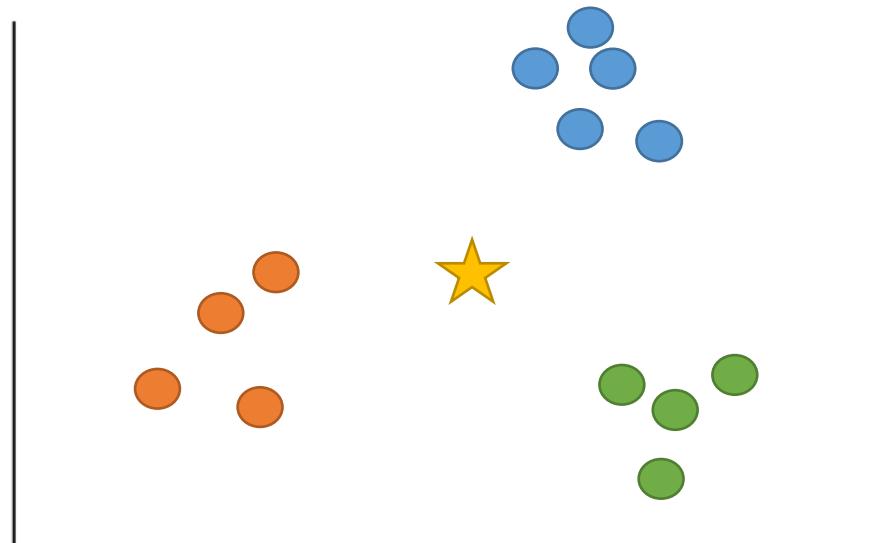


# k-最近傍法

# 最もシンプルな分類アルゴリズムを考えてみよう

---

- 既にクラスがわかっているデータが以下のように分布している
- 新しいデータ★に対して、どのクラスであるかを予測したい
- できるだけ簡単な方法で予測したいが、どのような方法が良いだろうか？



# 予測したいデータに近いものに合わせよう

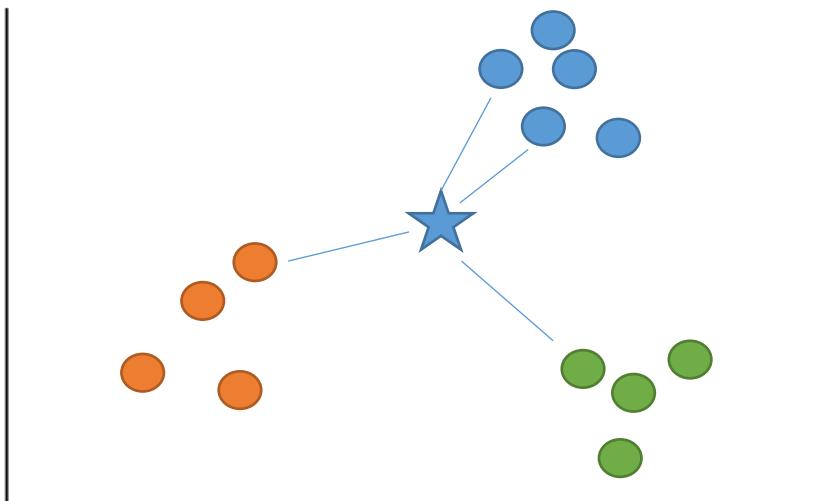
---

- 最もシンプルな方法は、★に最も近い点と同じクラスに割り当てる方法
- この方法は最近傍法と呼ばれる
- 結果をもう少し剛健にするために、見るデータの個数を増やしてみよう



# k-最近傍法 | k-Nearest Neighbor algorithm, k-NN

- 新たなデータ点から、近い順にk個のデータを考える
- このk個のデータのうち、最も多数派のクラスに割り当てるのがk-最近傍法
  - シンプルで解釈がしやすいが、全てのデータに対する距離の計算が必要なためデータ数が増えていくと、計算時間も増えていくのが難点



$k=4$ のとき、★から近い順に4つのデータを見る  
このとき多数派はクラス青  
よって、★はクラス青とする

# k-最近傍法まとめ（モデル、評価基準、最適化の観点から）

- あるデータが与えられたとき、距離が最も近いk個のデータを見て、その多数決によって分類を行う手法をk-最近傍法と呼ぶ
- シンプルであるため結果が解釈しやすいが、データの数が増えると距離の計算に時間がかかるという欠点もある

## モデル

- 近傍のk個のデータによる多数決

## 評価基準

- なし  
k-最近傍法には誤差を測る仕組みはない

## 最適化

- なし  
k-最近傍法には最適化の必要がない

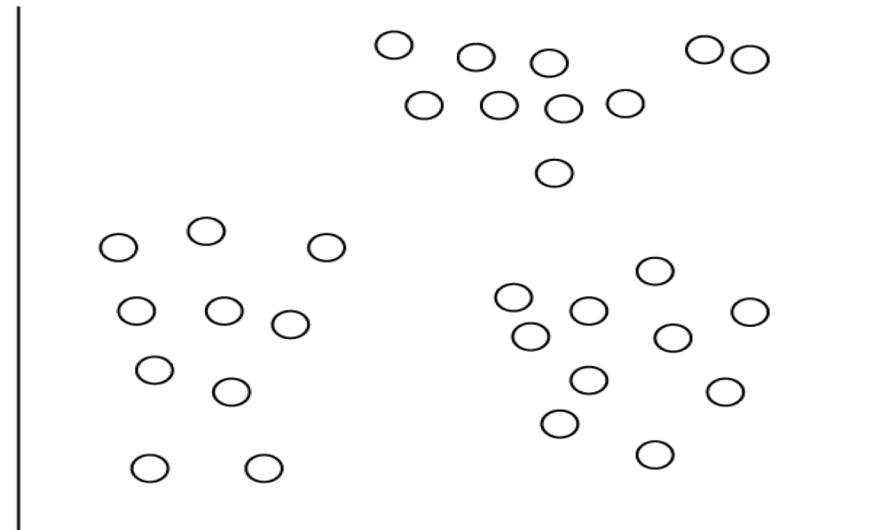
# 教師なし学習

1. クラスタリング
2. k-means法
3. 主成分分析
4. 自己符号化器

## データのグループ分け

---

- グループの数はいくつ？
- それぞれのグループの、どこに中心がある？



# クラスタリングとは

---

- ・ クラスタリングとは、似たものを集めて群（クラスタ）を作り、対象を分類しようという方法
- ・ クラスタという言葉は、マーケティング分野でよく使われる
- ・ 分類結果の正解は、無いことの方が多い

# クラス分類・クラスタリング問題で解決する課題例

---

## [クラス分類、教師あり]

- 事前に収集・識別済みのデータを基に、新たに集められたデータを識別する
  - 手書き文字認識
  - スパムメールの検知

## [クラスタリング、教師なし]

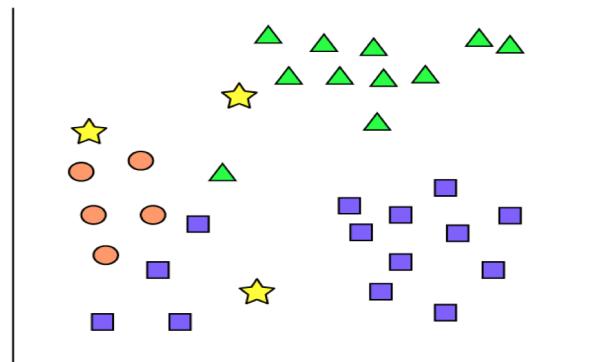
- 構造が明らかではないデータをグループ分けし、次の分析に備える
  - 顧客のセグメンテーション
  - 化学物質の構成に応じた分類

# クラスタリング手法の一覧

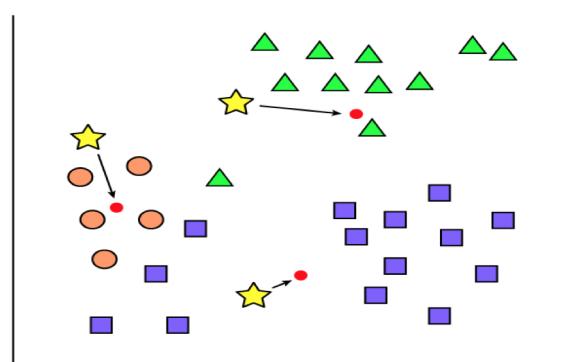
日本語名	英語名	クラスタ数の 決め方	分散	手法の概要
k-平均法	k-means	分析者が決める	等分散なデータで ないとうまく分類 できない	各データとクラスタ中心の距離の総和が最小に なるように分類する方法
X-平均法	X-means	BICが決める		BICでk数を決めるようにしたk-means
混合ガウス分布 モデル	Gaussian Mixture Model	分析者が決める		ガウス分布の重ね合わせによる確率分布を求め、 属するクラスタを求める方法
ディリクレ過程 混合ガウス モデル	Dirichlet Process Gaussian Mixture Model	クラスタリング と同時に推定 される	等分散でない データもうまく 分類できる	クラスタに分けられるデータ要素の確率だけ でなく、クラスタ数も推定する方法。  小さなクラスタとして判定され得るような微妙 なデータがあると、それをクラスタリングして しまい、真のクラスタ数よりも多いクラスタ数 を推定してしまうことがある。

# k-means の基本原理

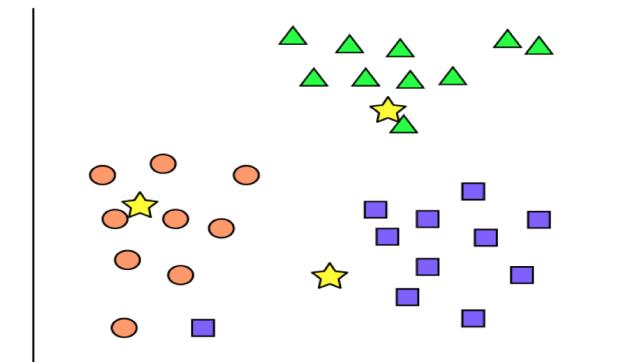
- 事前にk個のグループがあることを想定
- グループの中心点の配置、データのグループへの割り当て、グループ内の中心点の更新、という一連の処理を繰り返す
- 中心点が動かなくなるまで、処理を継続



中心点ランダム配置、  
初期グループ割り当て



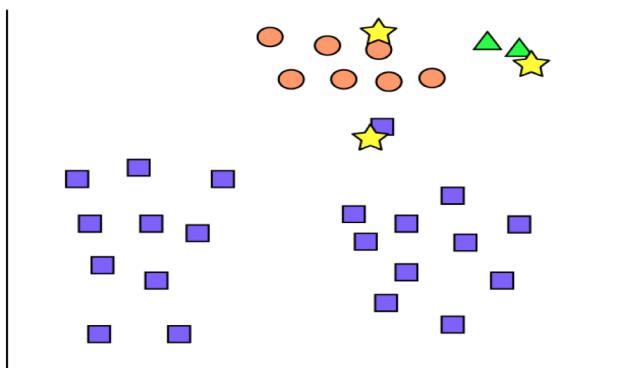
グループ中心を更新



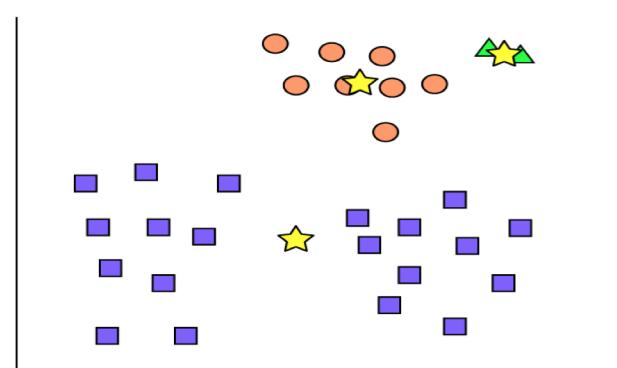
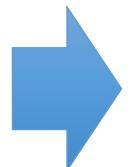
グループを再割り当て

# ランダム初期化と局所最適

- 中心の初期配置によっては、満足な結果が得られない
- ランダム性を保ちつつも、できるだけ最初の中心点間の位置を離しておく手法は、k-means++法と呼ばれる。この手法で、ある程度まで良い結果を得やすくなる
- scikit-learn の KMeans は、デフォルトでk-means++法を使用
- 初期値違いを複数回実行し、誤差平方和を最小にするものを選択するという方法も有効



偶然偏った初期配置



偏ったクラスタリング結果例

1. 入力データの中から初期の中心点をランダムに1個選択
2. 中心点以外の入力データを対象に、各中心点との距離の2乗を求める
3. 中心点以外の入力データを対象に、最近傍中心点からの距離の2乗に比例する重みを割り当てる
4. その重みを考慮しながら、次の中心点をランダムに選択
  - これにより、現在の中心点から遠い点ほど次の中心点に選ばれやすくなる
5. ステップ2~4を中心点がk個になるまで繰り返す
6. ステップ5の結果を初期値にk-means法を実行

## クラスタリング結果の評価

---

- 初期化状況により結果が変わりうるため、クラスタリングを複数回実行し妥当な結果を選ぶようにするのがよい
- 選択時、クラスタリング結果を評価するための指標が必要と
- 評価指標は、クラスタ内のデータと重心の誤差平方和 (SSE)
  - 分散に相当する指標
- 誤差平方和が小さいとまとまりが良いと考えられる。
- ただし、ばらつきが少ないほど良い、というケースばかりではない点に注意
- kの数が大きければ、ばらつきは少なくなる。また、分布に規則性が見られるなら、規則への適合度合いが重要。

# クラスタ内誤差平方和

---

- ・クラスタ内誤差平方和とは、クラスタリング後のバラツキを表す指標
- ・データの中心点はクラスタ内誤差平方和が最小となる位置に更新される
  - ・中心点の具体的な更新方法は『はじめてのパターン認識』10.2節を参照

## クラスタ内誤差平方和(SSE)

$$SSE = \sum_{i=1}^n \sum_{j=1}^k w^{(i,j)} \|x^{(i)} - \mu^{(j)}\|_2^2$$

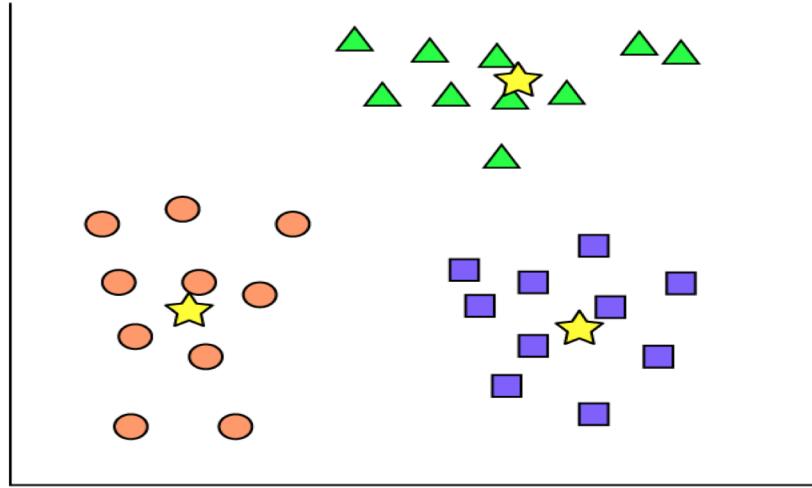
$n$  : データ数

$k$  : クラスタ数

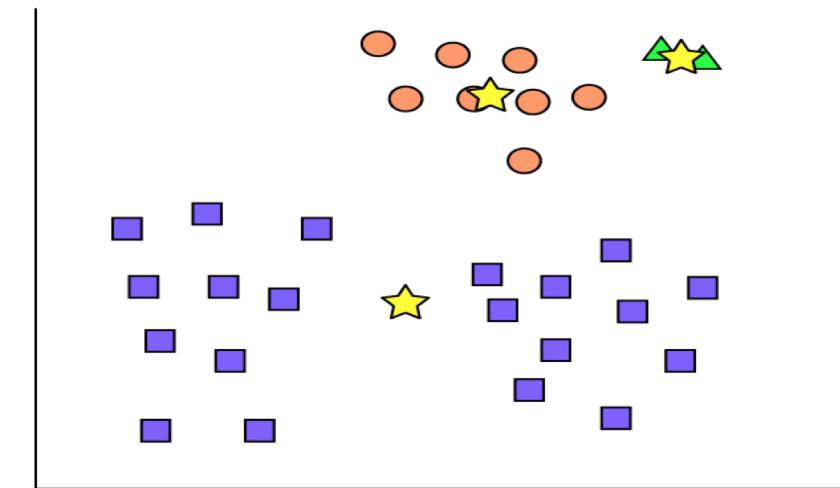
$w$  : サンプル点  $x^{(i)}$  がクラスタ  $j$  内に存在する場合は  $w^{(i,j)} = 1$ 、存在しない場合は、 $w^{(i,j)} = 0$

# クラスタリング結果の評価

---



理想的なケース



分散が大きいまま収束

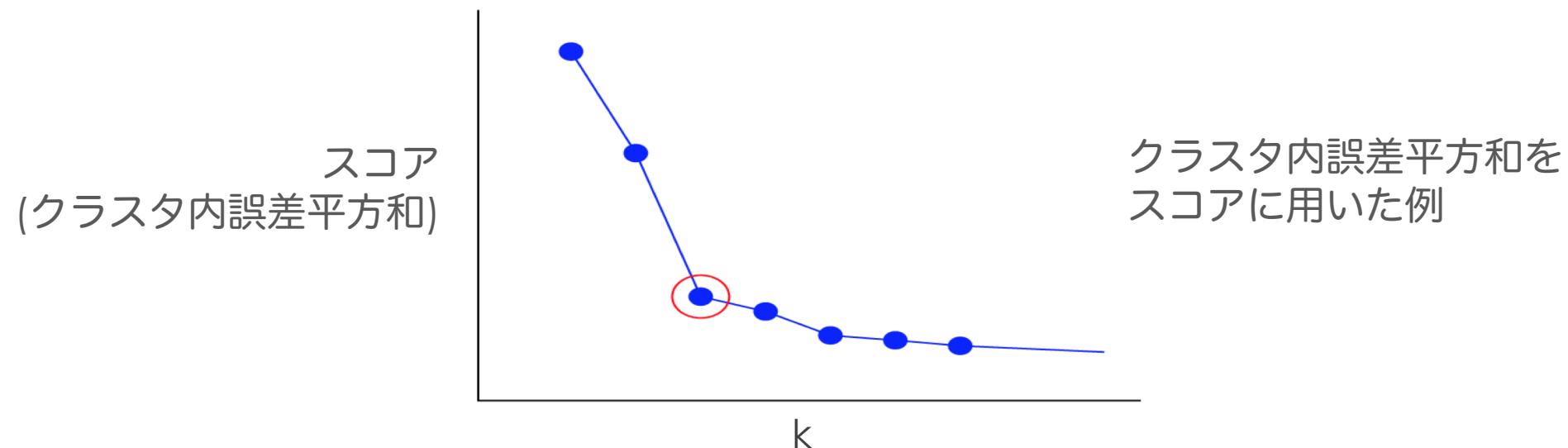
## 最適なkを決定する

---

- 所定のビジネスニーズがある場合は、その内容に従う。
  - 例えばA・B・Cの3つの広告プランがある時、何らかの尺度に基づき、顧客セグメントを3分割する
- クラスタリングが次の分析への前工程である場合、事前にkを決められないこともある
- kを決められない場合は、**小さな値から試していき**、その結果を指標(スコア)に基づき評価
  - 一定数のkまでクラスタリングをまとめておこない、後でスコアを一覧して良いものを選択
  - kごとに逐次スコアを確認し、スコア改善が見られなくなるまでkを増加させ続ける

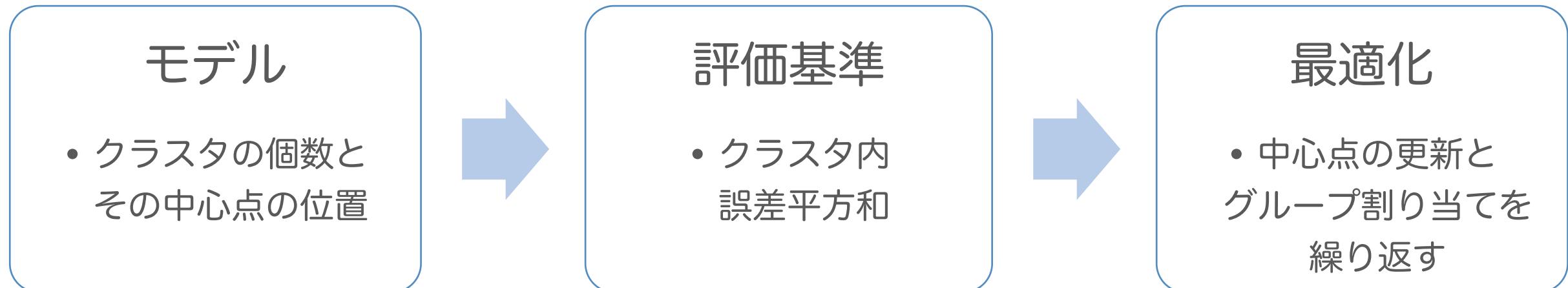
## エルボー法によるkの評価

- ・ 「妥当な」  $k$  の値まで実際に学習をおこない、スコアを後から評価するタイプの手法
- ・ スコアが急激に減少する部分をターゲットの  $k$  とする
- ・ 実際はきれいに「ひじ」が見えるケースは稀で、参考程度



# k-means法まとめ（モデル、評価基準、最適化の観点から）

- ・ グループの中心点の配置、データのグループへの割り当て、グループ内の中心点の更新、という一連の処理を繰り返すことで、データをグループ分けする方法をk-means法と呼ぶ
- ・ グループの個数はエルボー法によって決定



## [演習] 4\_k-means\_artificial.ipynb

---

- k-means法の実装を確認してみましょう
- 人工的なデータを用いて、実際にk-means法を試してみましょう

## [演習] 5\_k-means\_real.ipynb

---

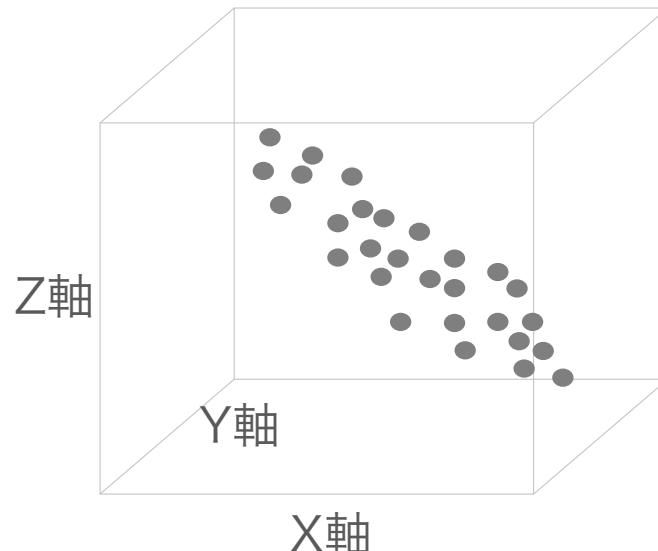
- ・ 実際のデータを用いて、k-means法でクラスタリングしてみましょう
- ・ 分割数が既知、未知のそれぞれのケースでk-means法による  
クラスタリングとデータの分類を行ってみましょう

# 主成分分析

# そのデータ、もっとわかりやすくならない？

---

- 以下のように、X軸・Y軸・Z軸で表現される3次元データがある
- データの次元を減らして、低次元でも意味のある表現を獲得したい
- もし、自由に軸を設定できるとすればどのように軸を置くべき？
  - 1次元（=軸1本）だったらどこに置く？2次元（=軸2本）だったら？

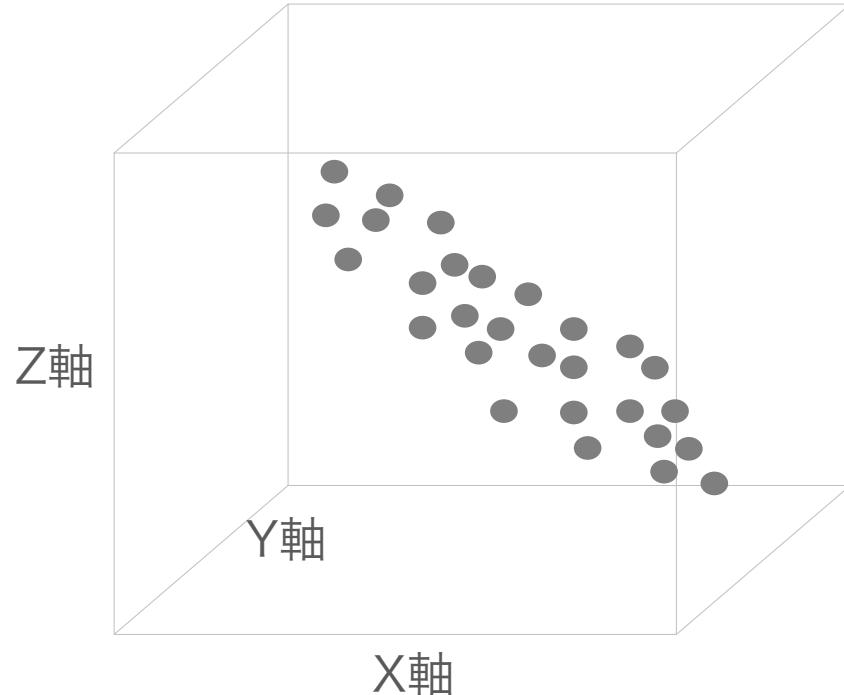


# 主成分分析とは？

---

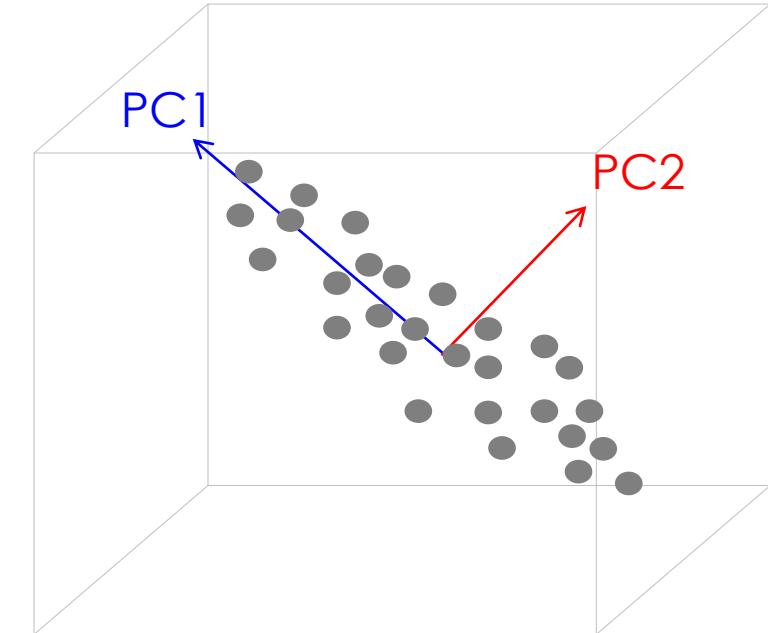
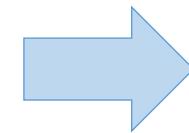
- ・ 主成分分析とは、元の変数に重みをかけた新しい合成変数（主成分）によって、次元軸を減らす手法
- ・ データの次元削減や、データの特性を抽出するのに役立つ
- ・ 多次元データを低次元（2次元のグラフなど）で可視化できる

# 主成分分析のイメージ



X軸、Y軸、Z軸  
の3次元データ

主成分分析



PC1軸とPC2軸の  
2次元空間(平面)に変換

## 主成分軸の性質

---

- PC1は第一主成分軸、PC2は第二主成分軸と呼ばれる
- 第一主成分軸は最も分散が大きい方向を表現しており、  
第二主成分軸は2番目に分散が大きい方向を表現している
- 各主成分軸は必ず直交するという性質がある
- これらの主成分軸は「固有値分解」という線形代数分野の手法によって  
求められる
- 元データが3次元の場合、第三主成分まで求められ、第三主成分を採用せ  
ず第二主成分まで採用すると、2次元のデータになり、次元が減る

# 参考：顔写真データの次元削減の例

- 顔写真データを主成分分析で次元削減している例

- [http://nbviewer.jupyter.org/github/contaconta/PCA\\_lecture/blob/master/PCA.ipynb](http://nbviewer.jupyter.org/github/contaconta/PCA_lecture/blob/master/PCA.ipynb)

## 顔データをPCAしてみる

画像データは2次元のデータ列->PCAをするとどうなるのか？

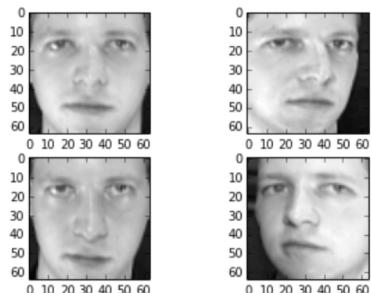
### 今回の実験

画像のサイズ64\*64px, 400人の顔データをPCAしてみるそれぞれの軸がどのようなデータの意味を表しているのかを見てみる

```
In [34]: # load data
import matplotlib as mpl
from sklearn.datasets import fetch_olivetti_faces
data = fetch_olivetti_faces()
```

```
In [35]: # データの中身確認
plt.subplot(221)
plt.imshow(data.images[0], cmap=mpl.cm.gray)
plt.subplot(222)
plt.imshow(data.images[1], cmap=mpl.cm.gray)
plt.subplot(223)
plt.imshow(data.images[2], cmap=mpl.cm.gray)
plt.subplot(224)
plt.imshow(data.images[3], cmap=mpl.cm.gray)
```

```
Out[35]: <matplotlib.image.AxesImage at 0x11bfff9550>
```



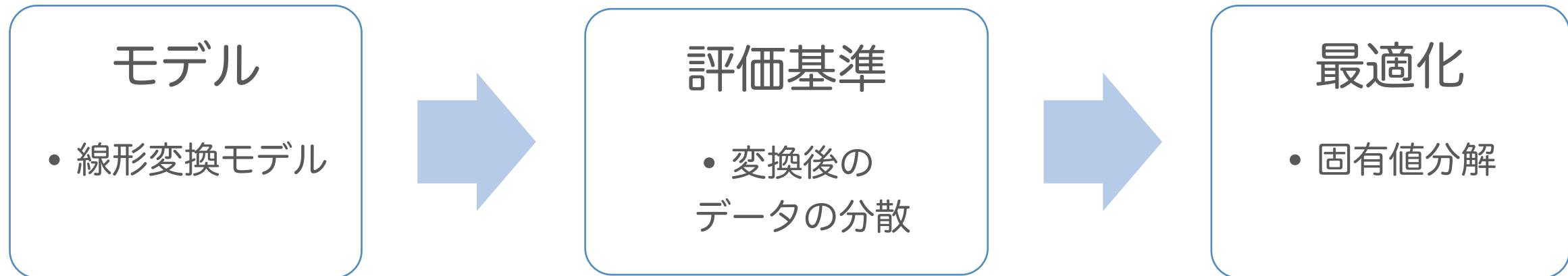
## [演習] 6\_PCA.ipynb

---

- ・主成分分析によって、データの次元削減を行ってみましょう
- ・寄与率を確認し、重要な特徴とはなにかを考えてみましょう

# 主成分分析まとめ（モデル、評価基準、最適化の観点から）

- ・主成分分析とは、元の変数に重みをかけた新しい合成変数（主成分）によって、次元軸を減らす手法
- ・データの次元削減や、データの特性を抽出するのに役立つ
- ・多次元データを低次元(2次元のグラフなど)で可視化できるようになる



## 自己符号化器

# 深層学習の様々なモデル

確定的

階層型

全結合型ニューラルネットワーク  
Fully connected neural network

畳み込みニューラルネットワーク  
Convolutional neural network

再帰型ニューラルネットワーク  
Recurrent neural network

自己符号化器型

自己符号化器  
Autoencoder

雑音除去自己符号化器  
Denoising Autoencoder

変分自己符号化器  
Variational Autoencoder

スパース自己符号化器  
Sparse Autoencoder

確率的

ボルツマンマシン型

ボルツマンマシン  
Boltzmann machine

制約ボルツマンマシン  
Restricted Boltzmann machine

深層信念ネットワーク  
Deep belief network

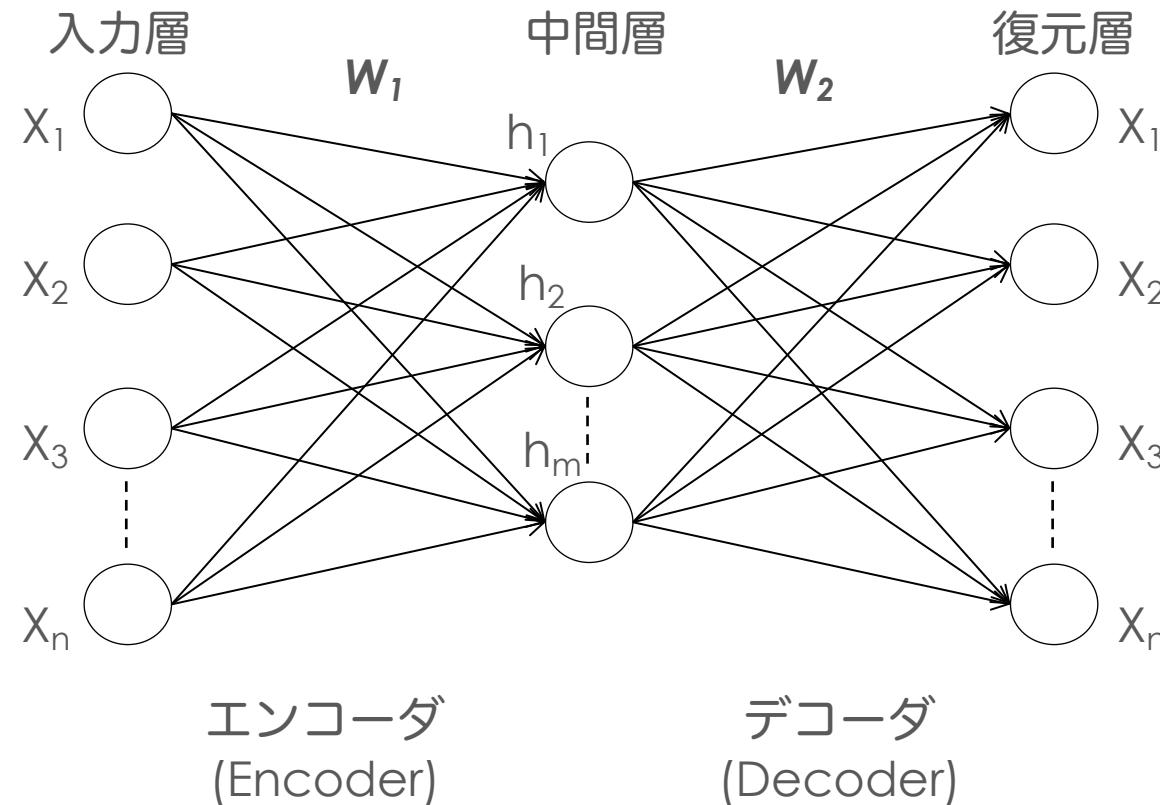
深層ボルツマンマシン  
Deep Boltzmann machine

参考：『深層学習』(神鳶ら、近代科学社)

# 自己符号化器型のネットワーク

- 学習時は、入力層と復元層に同じデータを入れる。
- 学習結果によって求められた重みWや中間層 $h_m$ が次の計算に利用されたりする。

自己符号化器の例



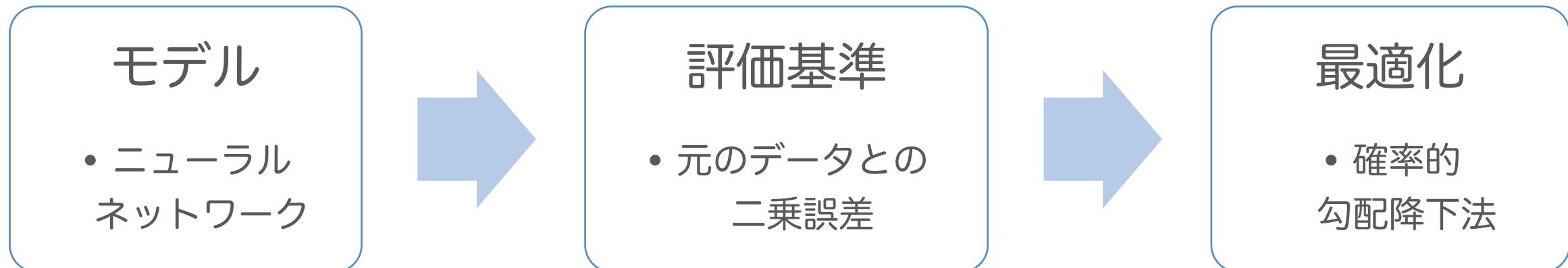
# 自己符号化器 (Autoencoder)とは

---

- ・自己符号化器とは、エンコーダとデコーダを組み合わせたネットワークを組み、ノード間の重みを学習する方法
- ・特徴抽出器(次元圧縮)、ノイズ除去、階層型NNの初期パラメータの取得、復元誤差を利用した異常検知などに利用される
- ・スパース自己符号化器は、自己符号化器に正則化項を加えて学習させる方法
  - ・効率的にユニット数を少なくできることが特徴
- ・雑音除去自己符号化器とは、ノイズを付加したデータを入力層に入れる方法
  - ・復元層から出てくるデータがノイズを付加する前のデータに近くなるように学習させる

# 自己符号化器まとめ（モデル、評価基準、最適化の観点から）

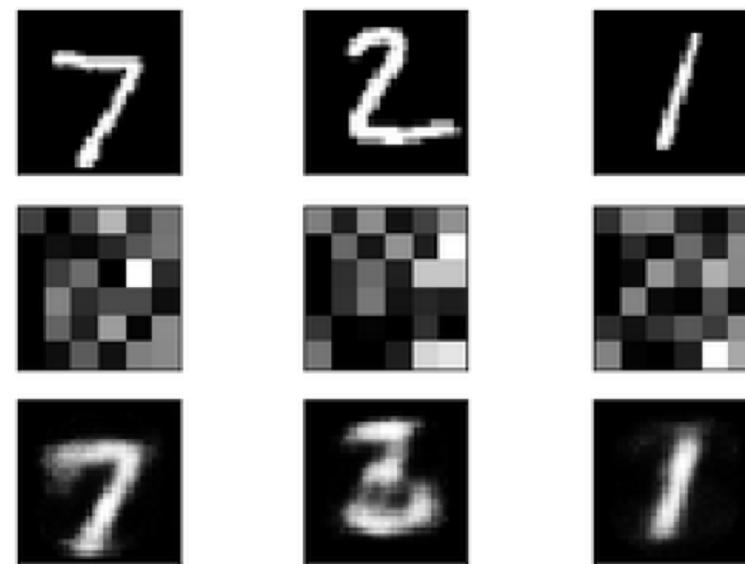
- ・自己符号化器とは、エンコーダとデコーダを組み合わせたネットワークを組み、ノード間の重みを学習する方法
- ・特徴抽出器(次元圧縮)、ノイズ除去、階層型NNの初期パラメータの取得、異常検知などに使用される



## [演習] 7\_AutoEncoder.ipynb

---

- ・自己符号化器を用いて、手書き数字画像のデータを圧縮してみましょう
- ・パラメータを変更すると、デコーダから再構成される画像がどのように変化するか試してみましょう



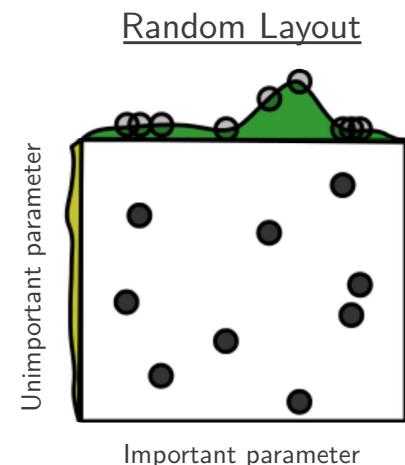
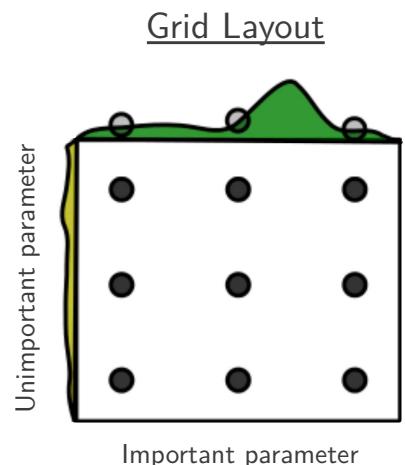
# ハイパーパラメータ最適化

1. ランダムサーチ
2. ベイズ的最適化

# ランダムサーチ | Random Search

---

- ・ハイパーパラメータの最適化手法として、DAY2ではグリッドサーチを紹介
- ・しかし、グリッドの間隔が広すぎると最適なパラメータを見つけられない
  - ・例) グリッドの間に最適なパラメータがある場合
- ・そこで、乱数を用いて探索するランダムサーチが提案されている



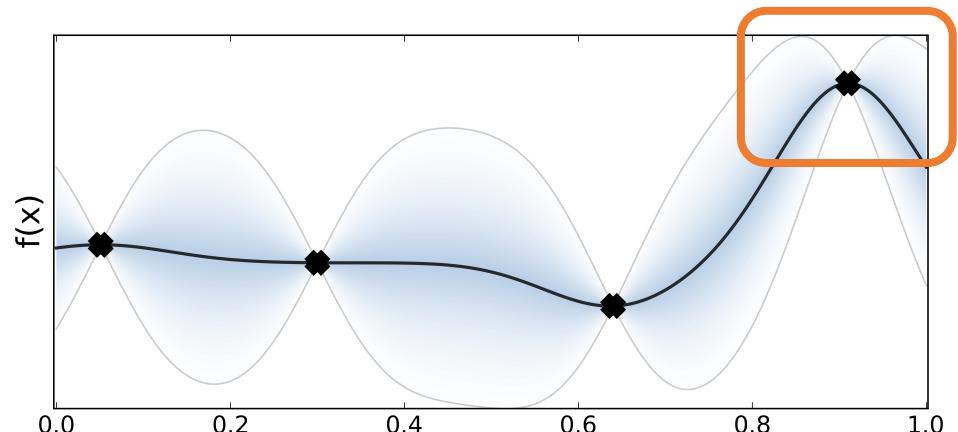
Scikit-learnでは、`RandomizedSearchCV`として  
実装されている

詳しくはこちらを参照：

<http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf>

# ベイズ的最適化 | Bayesian Optimization

- ・ パラメータの値を特徴とし、パラメータに対する性能を予測する回帰モデルを学習
- ・ その後、その回帰モデルが示す最も性能が良さそうなパラメータを使って、パラメータをチューニングしたいモデルを学習。その結果を用いて、性能予測モデルを再学習する
- ・ これを何度も行い、良いパラメータを見つける方法がベイズ的最適化である
- ・ 少ない探索回数でも、良いパラメータを見つけやすい手法であるため、学習に時間のかかるモデルのパラメータ探索に適している。しかし、点の数が増えると計算時間がかかるのが難点



このあたりが、性能が最大になりそうなところと予測されている  
この結果を使って、ハイパーパラメータを選択

詳しくはこちらを参照： <https://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf>

## グループワーク

---

- DAY1~4にかけて、様々な機械学習手法について触れていきました
- 総仕上げとして自分の専門分野に機械学習を活用しようとするを考えてみましょう
- 自分の専門分野に対して機械学習をどのように活用できるか、できるだけ多く挙げてみましょう（5分）
- 最も独創的なアイデアをピックアップし、どのようなモデルを使うべきか、どのようなデータを集めるべきか検討してみましょう（10分）
- 最後に以上2つで議論した結果を全体に向けて発表しましょう（各グループ2分）

# Any Questions ?

# Appendix

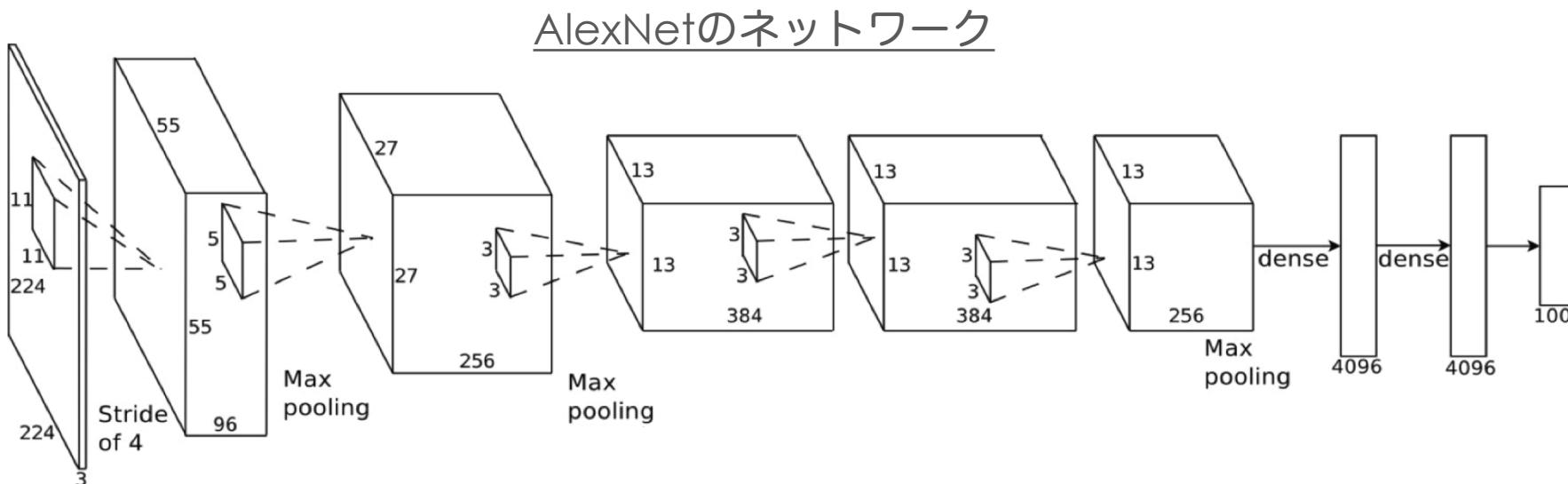
1. 著名なCNN
2. LSTM / GRU
3. 機械学習・深層学習に関する論文調査の方法

## 著名なCNN

---

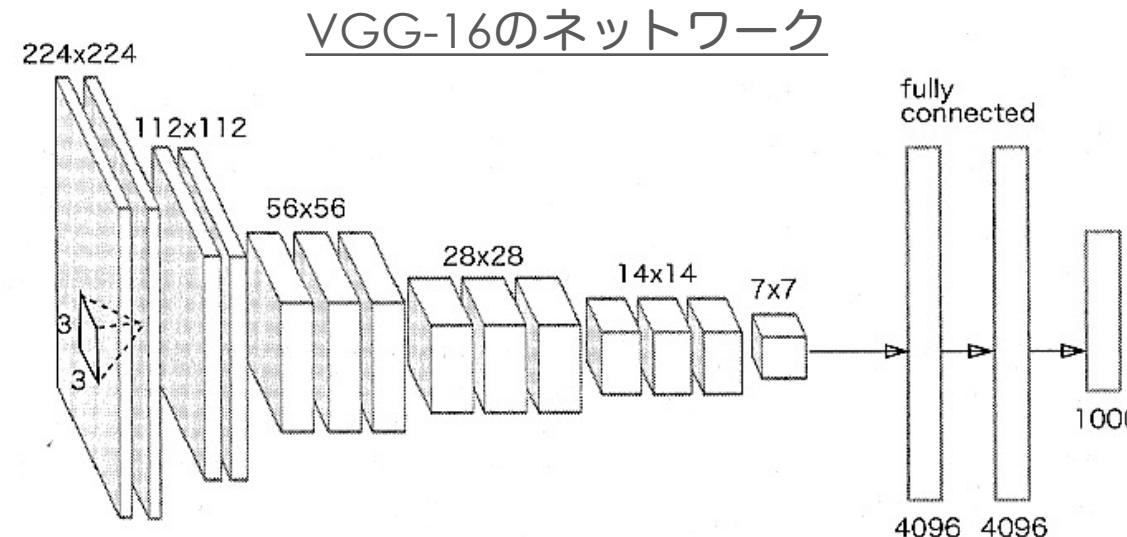
# AlexNet

- 2012年のILSVRCで優勝したモデル
- 5層の畳み込み層と3層の全結合層で構成
- 論文：<https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>



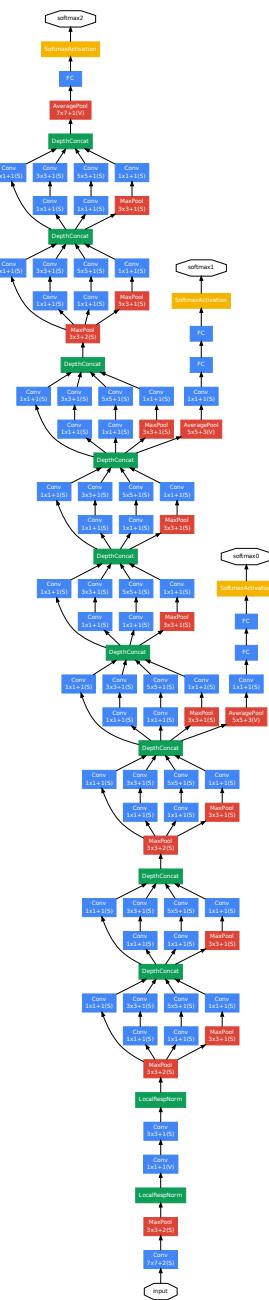
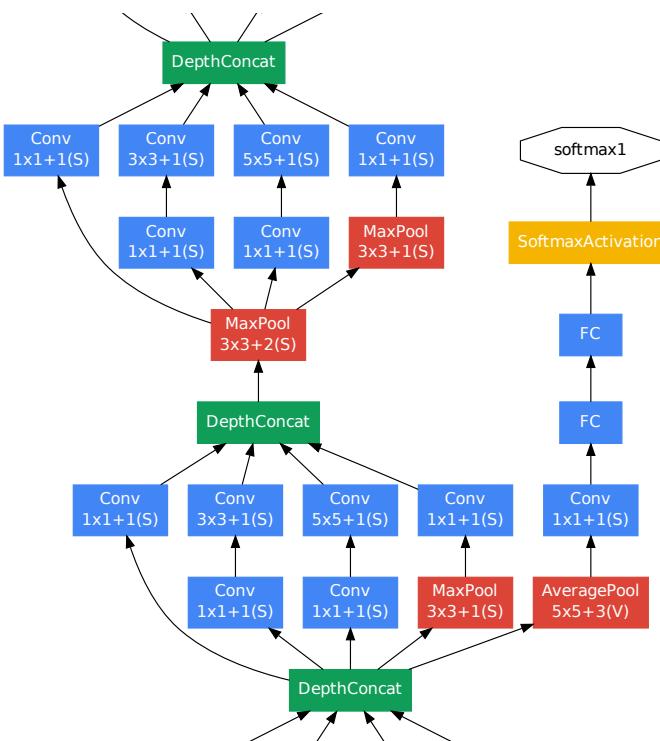
# VGG

- 2014年のILSVRCで2位になったモデル
- 19層のモデルはVGG-19、16層のモデルはVGG-19と呼ばれる
- VGGとは、Visual Geometry Group (Oxford大学の研究室) の略称
- 構造がシンプルであるため、実務でよく使用される
- 論文：<https://arxiv.org/pdf/1409.1556.pdf>



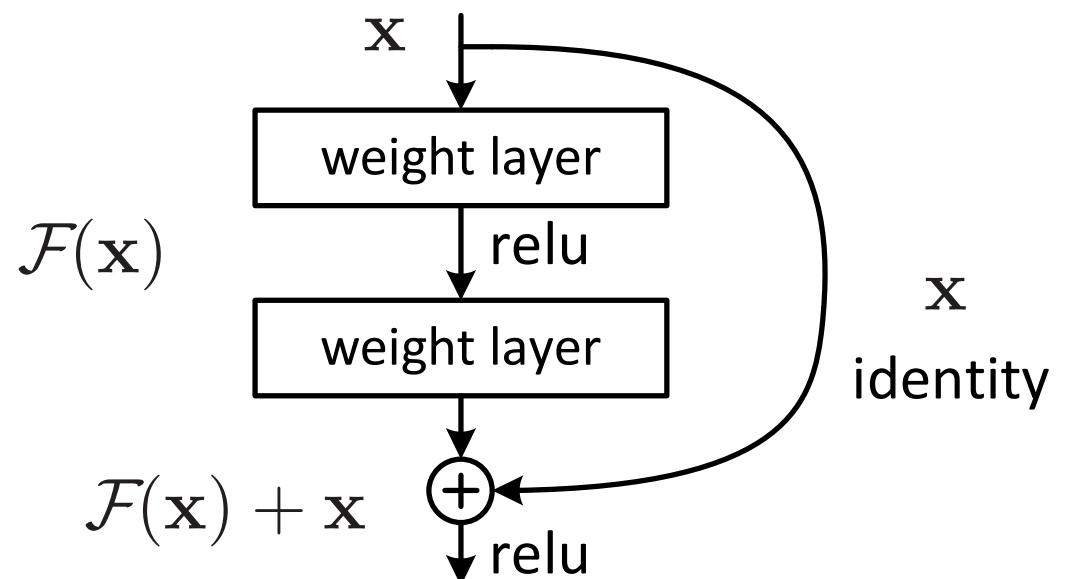
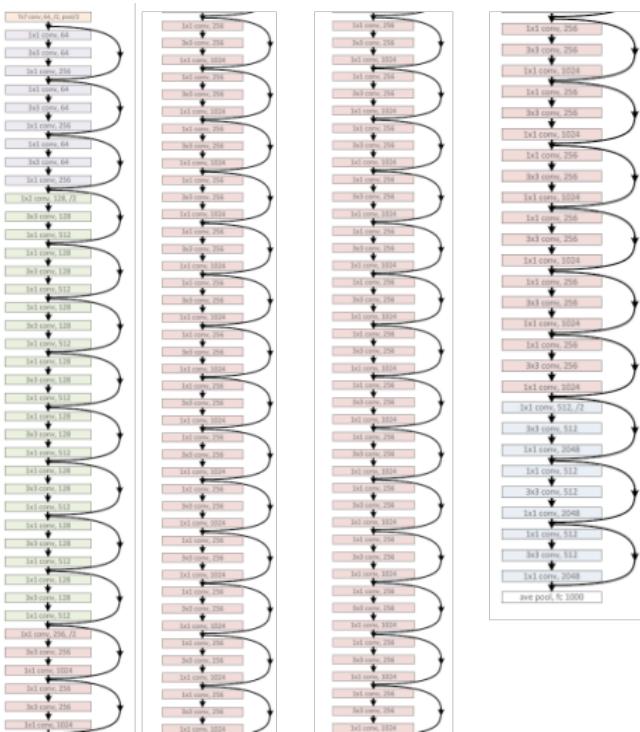
# GoogLeNet

- 2014年のILSVRCで1位になったモデル
- 論文：[https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2015/papers/Szegedy\\_Going\\_Deeper\\_With\\_2015\\_CVPR\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf)



# ResNet

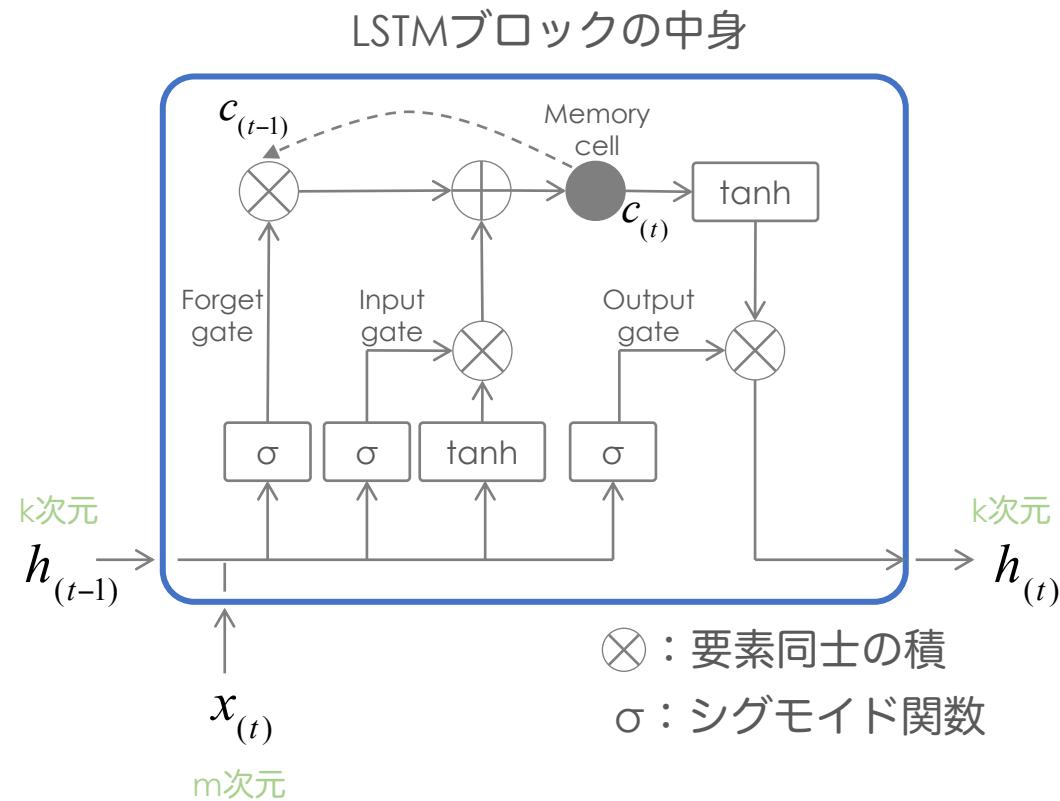
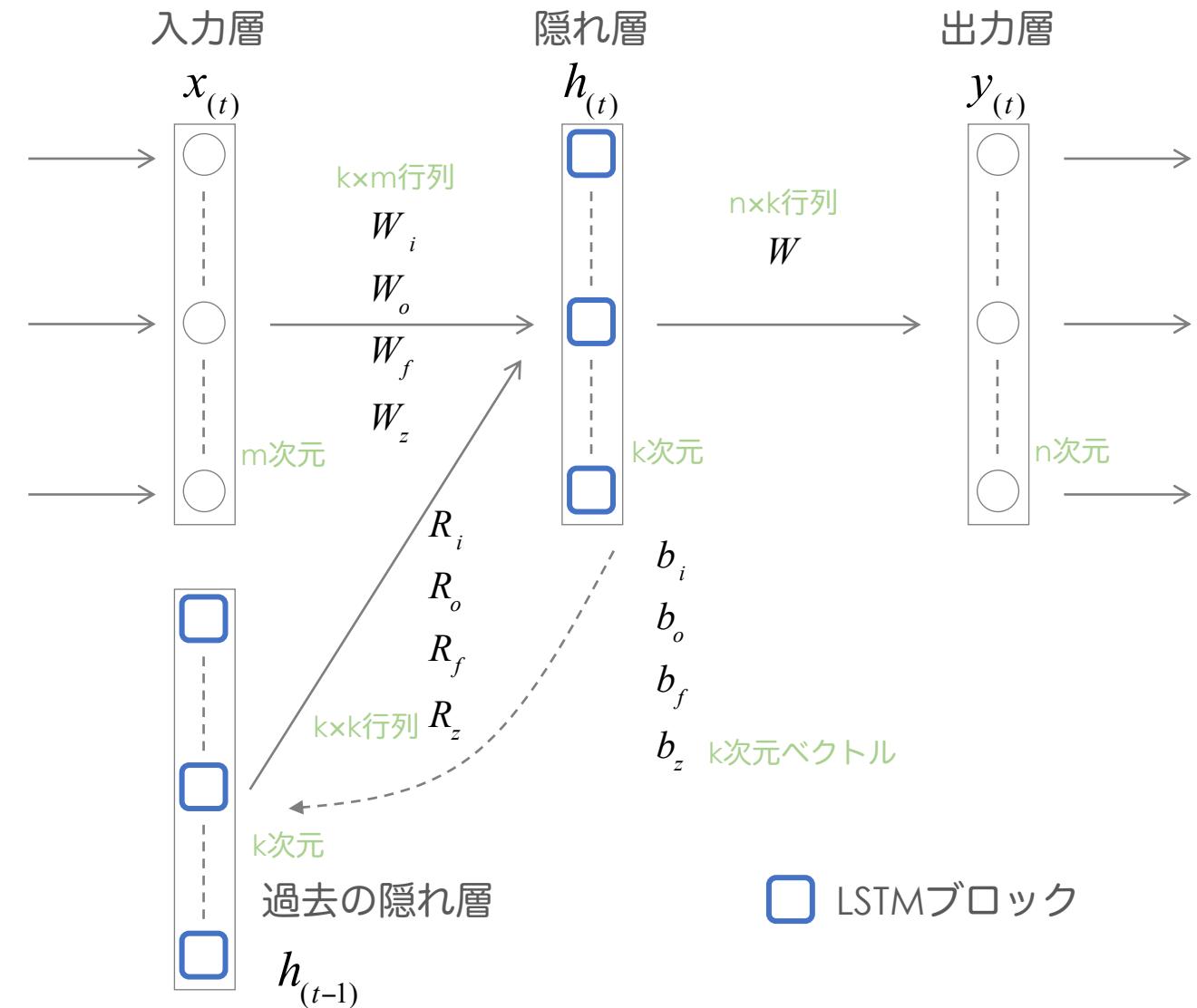
- 2015年のILSVRCで1位になったモデル
- 論文：[https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/He\\_Deep\\_Residual\\_Learning\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf)



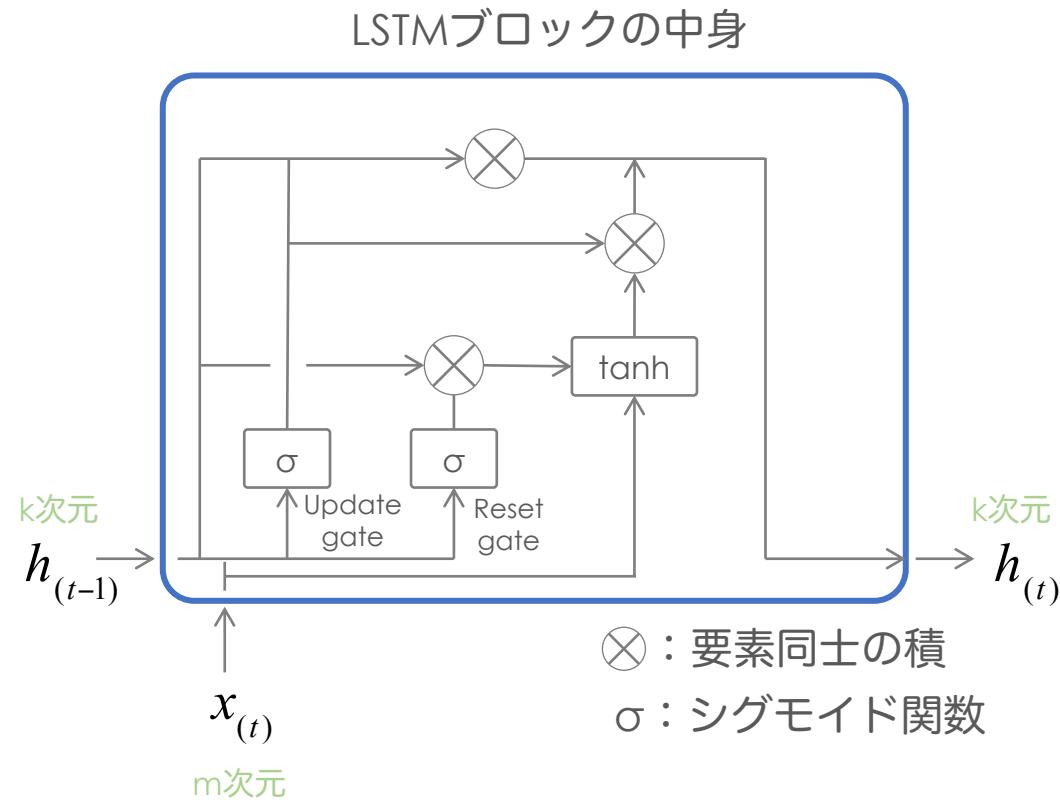
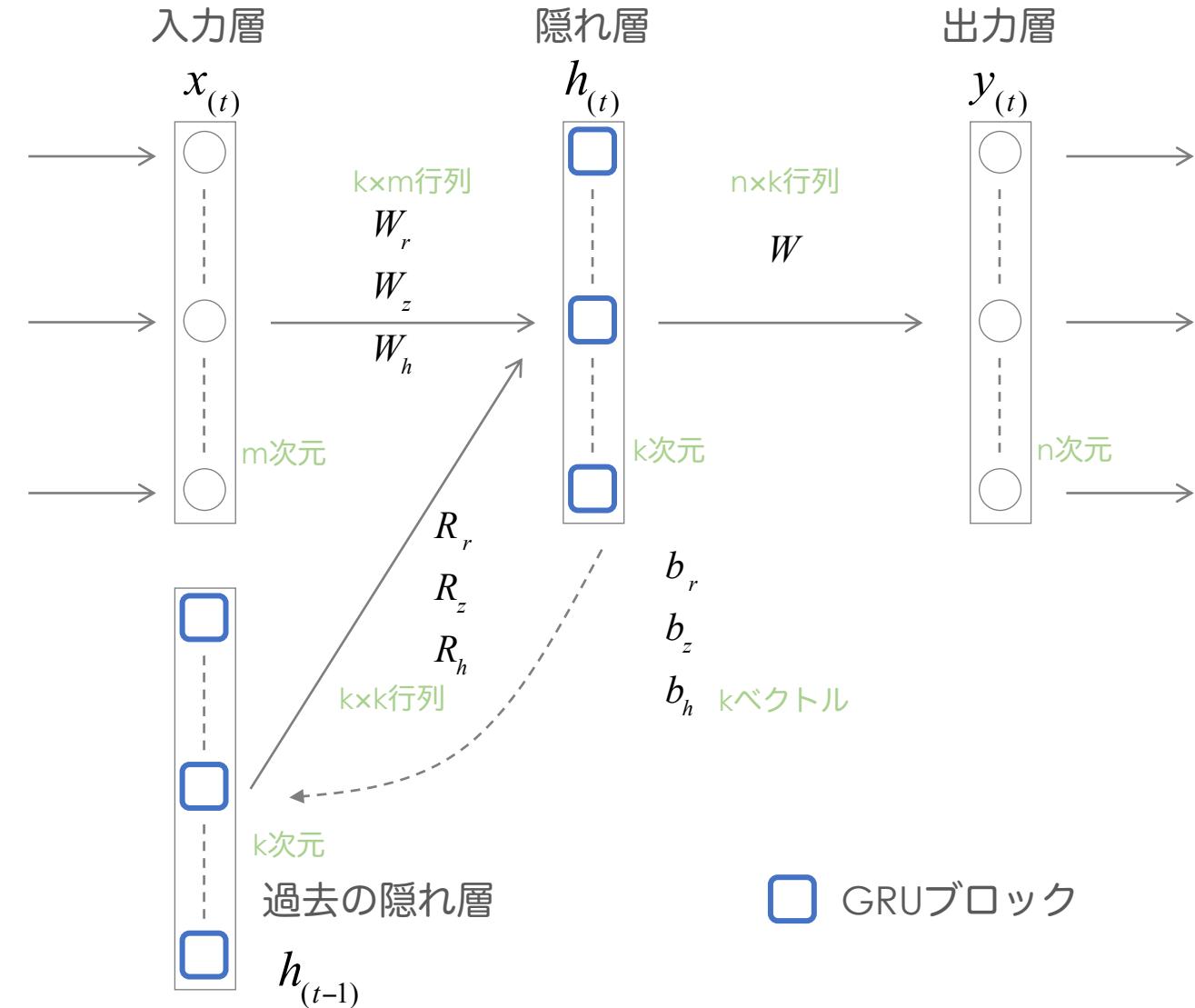
## **LSTM / GRU**

---

# LSTMのネットワーク



# GRUのネットワーク



## 機械学習・深層学習に関する論文調査の方法

---

# 機械学習・深層学習に関する論文調査の方法

---

- ・機械学習や深層学習は、日々新しい手法が開発されている分野
- ・最新の手法は書籍に掲載されていないことが多いため、動向を掴むには論文を読む必要がある
- ・国際会議論文が重要視される分野であるため、主要な国際会議を把握しておき、そこに採択された論文を読むのがおすすめ
  - ・査読（複数人の研究者によるチェック）が入っているため、質の良い論文が多い
- ・機械学習系の会議リストはこちらを参考：  
<http://ibisforest.org/index.php?Meeting>

# 機械学習・深層学習に関する主要な国際会議

---

- 下記3つの国際会議は、機械学習・深層学習に関する主要な国際会議

## 1. ICML (International Conference on Machine Learning)

- ICML2018の採択論文リスト：<http://proceedings.mlr.press/v80/>

## 2. NIPS (Neural Information Processing Systems)

- NIPS1987~2017の採択論文リスト：<https://papers.nips.cc/>

## 3. ICLR (International Conference on Learning Representations)

- ICLR2018の採択論文リスト：

<https://openreview.net/group?id=ICLR.cc/2018/Conference>

# 動画像処理・自然言語処理に関する主要な国際会議

---

- 応用分野ごとの国際会議を見ても、機械学習・深層学習絡みの論文が多く発表されているため、応用を知りたい場合は以下を覗くのがおすすめ
- 画像認識などの動画像処理に関する主要な国際会議
  - CVPR (Computer Vision and Pattern Recognition)
  - ICCV (International Conference on Computer Vision)
  - 上記2つの採択論文リスト：<http://openaccess.thecvf.com/menu.py>
- 自然言語処理に関する主要な国際会議
  - ACL (Association of Computational Linguistics)
  - ACL2017の採択論文リスト：<http://aclweb.org/anthology/P/P17/>