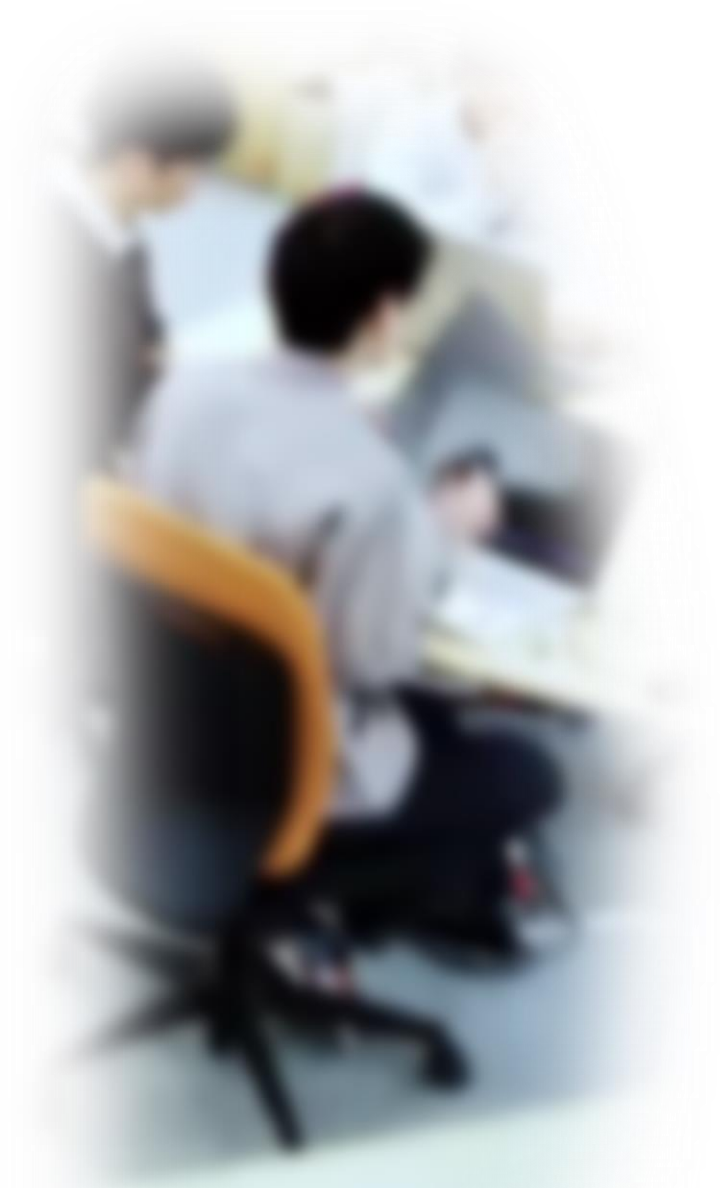
aa-4. 機械学習、ニューラルネットワーク

（人工知能）

URL: https://www.kkaneko.jp/ai/mi/index.html

金子邦彦



アウトライン

1. 機械学習
2. 機械学習の現状
3. ニューラルネットワークの概要
4. ニューラルネットワークの種類，応用分野
5. 線形近似（学習の例）
6. 最適化

今日の授業では，**Office 365 のインストール**を済ませておくことが便利です．

4-1 機械学習

機械学習の特徴

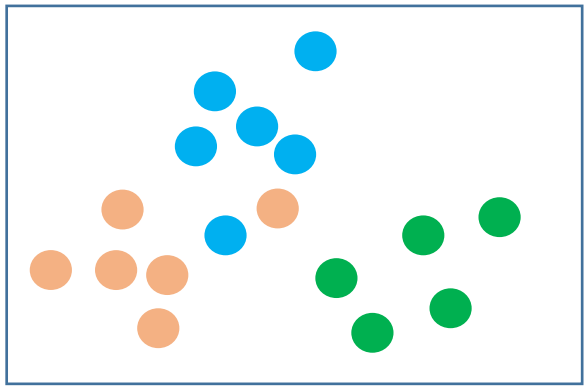
機械学習は，**コンピュータ**が**データ**を使用して**学習**することにより**知的能力を向上**させる技術

- **情報の抽出**：データの中からパターンや関係性を自動で見つけ出す能力
- **簡潔さ**：人間が設定しなければならなかったルールを、自動で生成できるようになる
- **限界の超越**：他の方法では難しかった課題でも、機械学習を用いることで解決策や視点を得られる可能性がある

機械学習と訓練データ

機械学習は，**コンピュータ**が**データ**を使用して**学習**することより**知的能力を向上**させる技術

訓練データ



3 種類に分類済み

学習

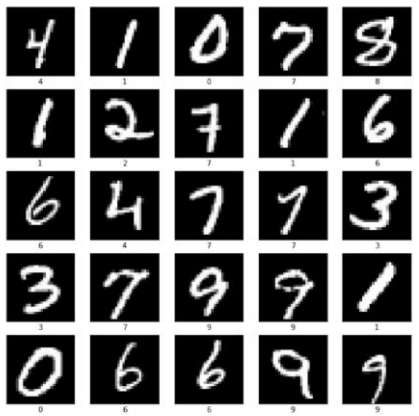
学習者

大量の訓練データを用いて
学習を行う

機械学習と訓練データとプログラム

機械学習のプログラム

学習に使用する訓練データ
（抜粋）



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

画像 60000枚
（うち一部）

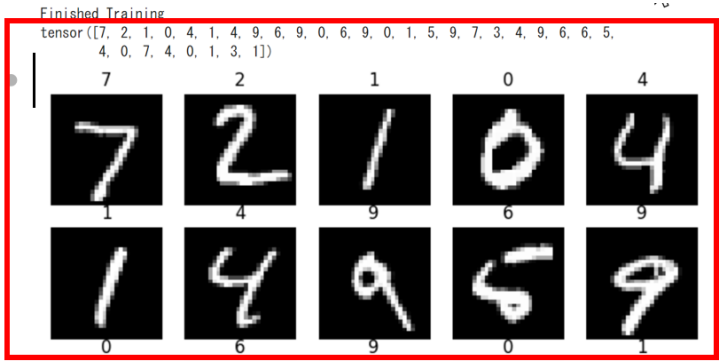
正解 60000個
（うち一部）

プログラム

データを用いて学習を行う
学習ののち，画像分類を行う

```
[4] !pip install -U scikit-learn matplotlib

    import torch
    import torch.nn as nn
    import torch.optim as optim
    from sklearn import datasets
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler
    import matplotlib.pyplot as plt

    # データの取得と前処理
    iris = datasets.load_iris()
    X = iris.data
    y = iris.target

    # データの標準化
    scaler = StandardScaler()
    X = scaler.fit_transform(X)

    # 訓練データとテストデータの分割
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    X_train = torch.tensor(X_train, dtype=torch.float32)
    y_train = torch.tensor(y_train, dtype=torch.long)
    X_test = torch.tensor(X_test, dtype=torch.float32)
    y_test = torch.tensor(y_test, dtype=torch.long)

    # ニューラルネットワークの定義
    class Net(nn.Module):
        def __init__(self):
            super(Net, self).__init__()
            self.fc1 = nn.Linear(4, 10)  # 入力は4次元（Irisの特徴量）
            self.fc2 = nn.Linear(10, 3)  # 出力は3クラス

        def forward(self, x):
            x = torch.relu(self.fc1(x))
            x = self.fc2(x)
            return x

    net = Net()
    criterion = nn.CrossEntropyLoss()
    optimizer = optim.SGD(net.parameters(), lr=0.01)
```

学習の結果、文字認識の能力を獲得



```
Finished Training
tensor([7, 2, 1, 0, 4, 1, 4, 9, 6, 9, 0, 6, 9, 0, 1, 5, 9, 7, 3, 4, 9, 6, 6, 5,
        4, 0, 7, 4, 0, 1, 3, 1])
```

機械学習まとめ

機械学習の特徴

- データを用いて知的能力を向上
- 自動でデータのパターンを抽出
- さまざまなタスクを自動実行

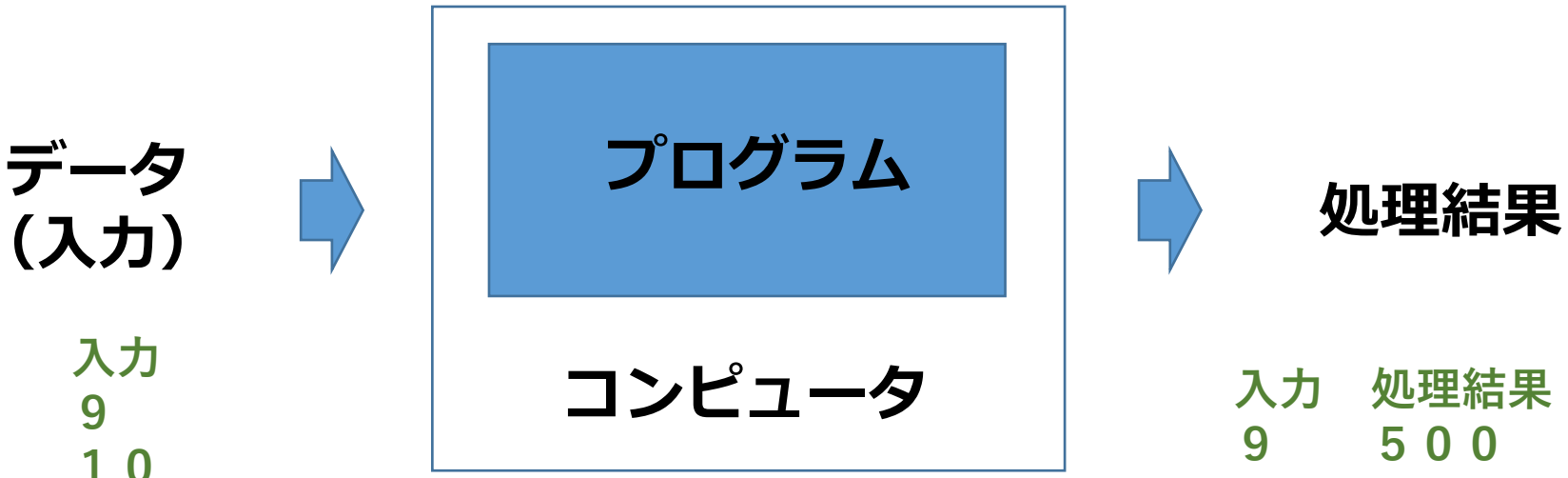
応用事例

画像理解、自然言語処理、予測など多数

機械学習の定義：

- **訓練データ**を用いて**学習**し、その結果として知的能力が向上
- 訓練データの追加により、さらに知的能力が向上する可能性

一般のプログラミング

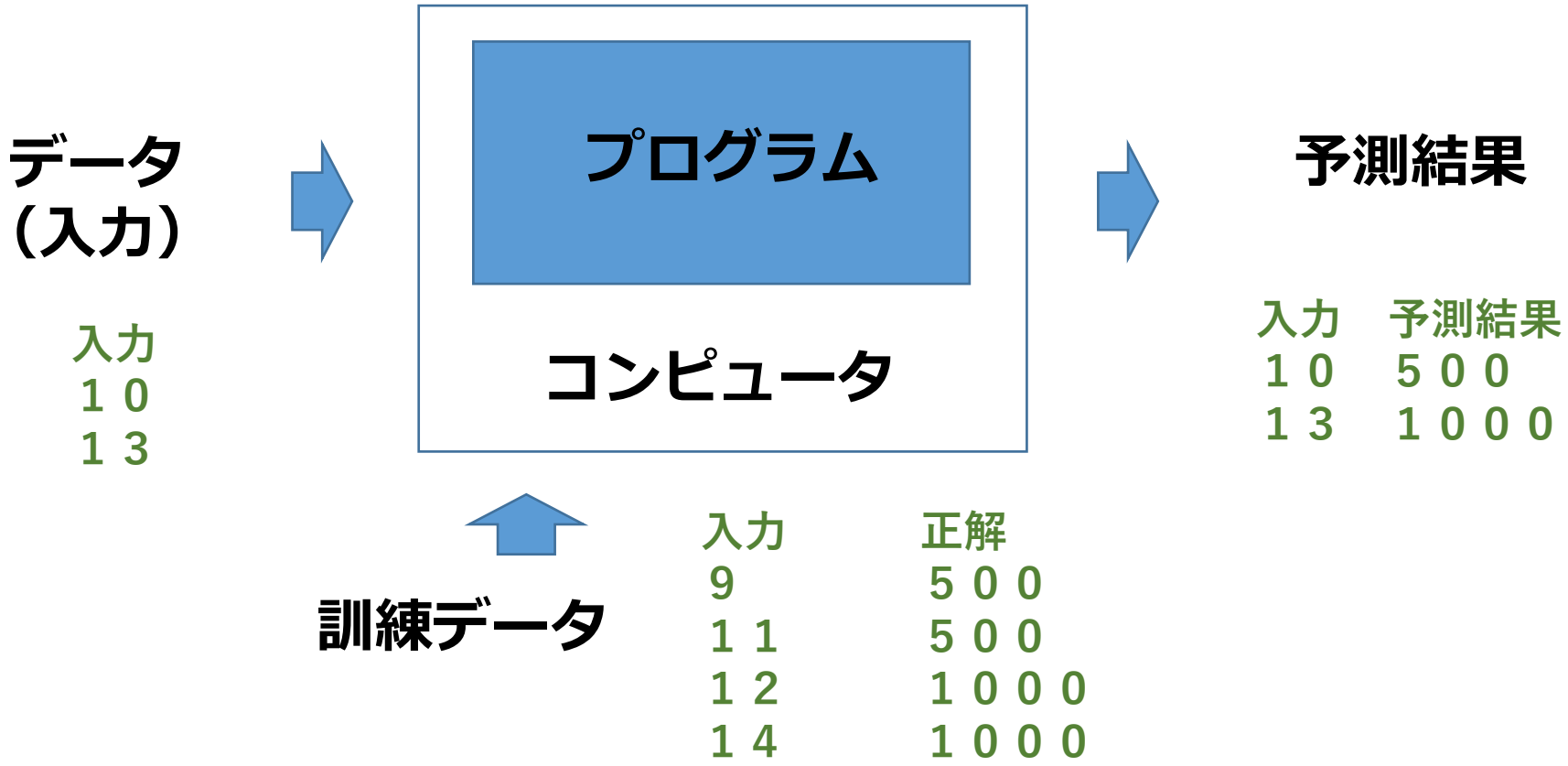


入力
9
10
11
12
13
14

入力	処理結果
9	500
10	500
11	500
12	1000
13	1000
14	1000

あらゆる入力について
正しい処理結果が得られるように，
プログラムを作成し，テストする

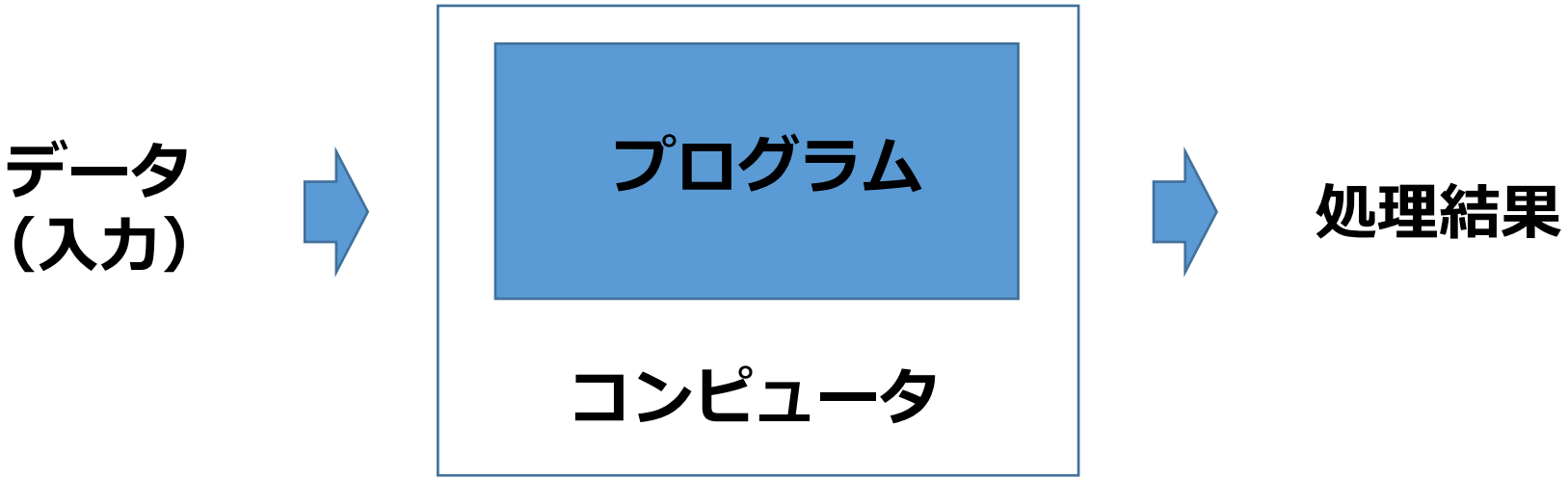
機械学習での予測



訓練データにより，学習が行われる。
新しいデータに対しては、自動で処理が行われる。

① 一般のプログラミング

・プログラムは人間が作成し，テストし，調整する．



② 機械学習

・データを利用して知的能力を向上させる

4-2 機械学習の現状

機械学習の特徴と応用

- **能力の多様性**:

分類、検出、識別、認識（文字認識、画像の認識、音声認識）、予測、合成、翻訳、特徴抽出、ランク付け、情報推薦など、**さまざまな能力を持つ**

- **応用範囲の広さ**:

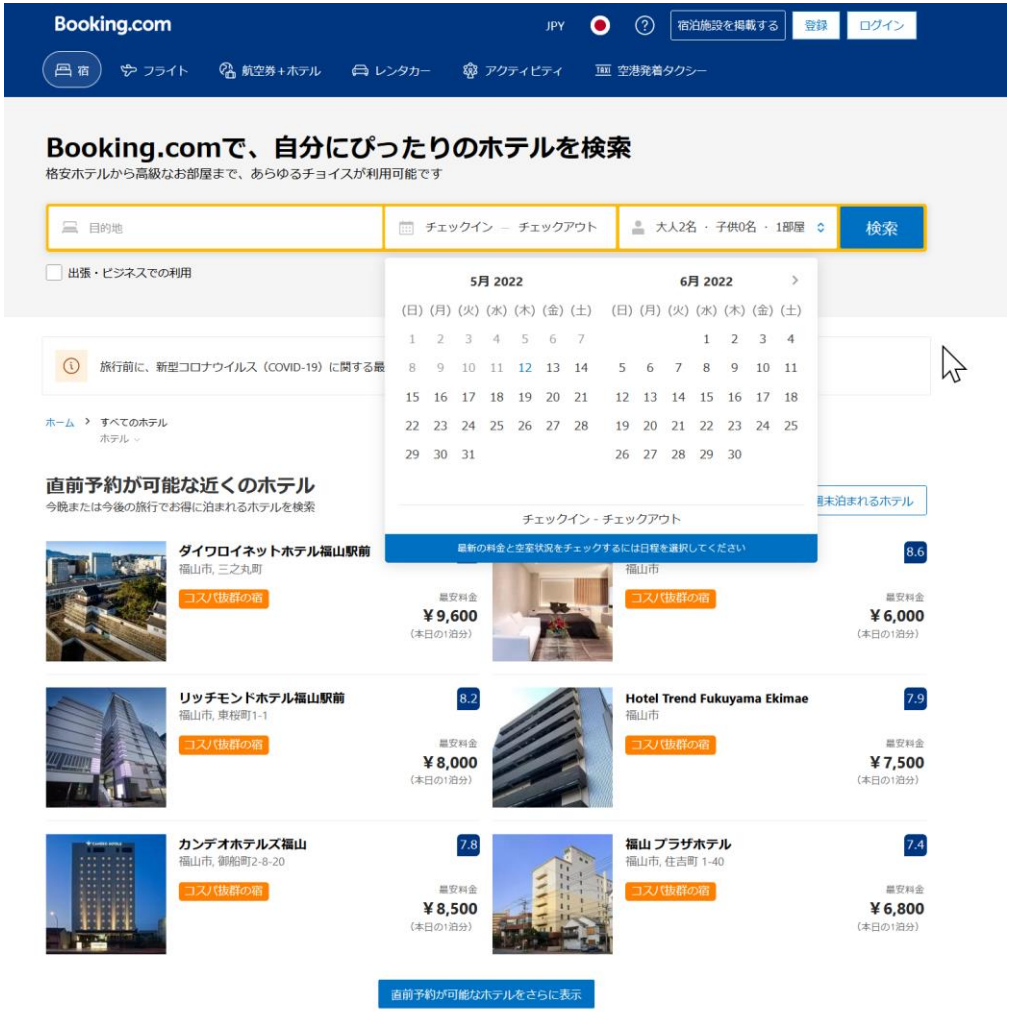
経済、金融、ロボット、医療、医学研究（遺伝子解析）など、様々な分野で問題解決に活用

機械学習は多様な能力を持ち、経済や金融から医療やロボットまで広範な応用が可能

人工知能の宿泊サイトでの応用

- 宿泊予約サイト Booking.com

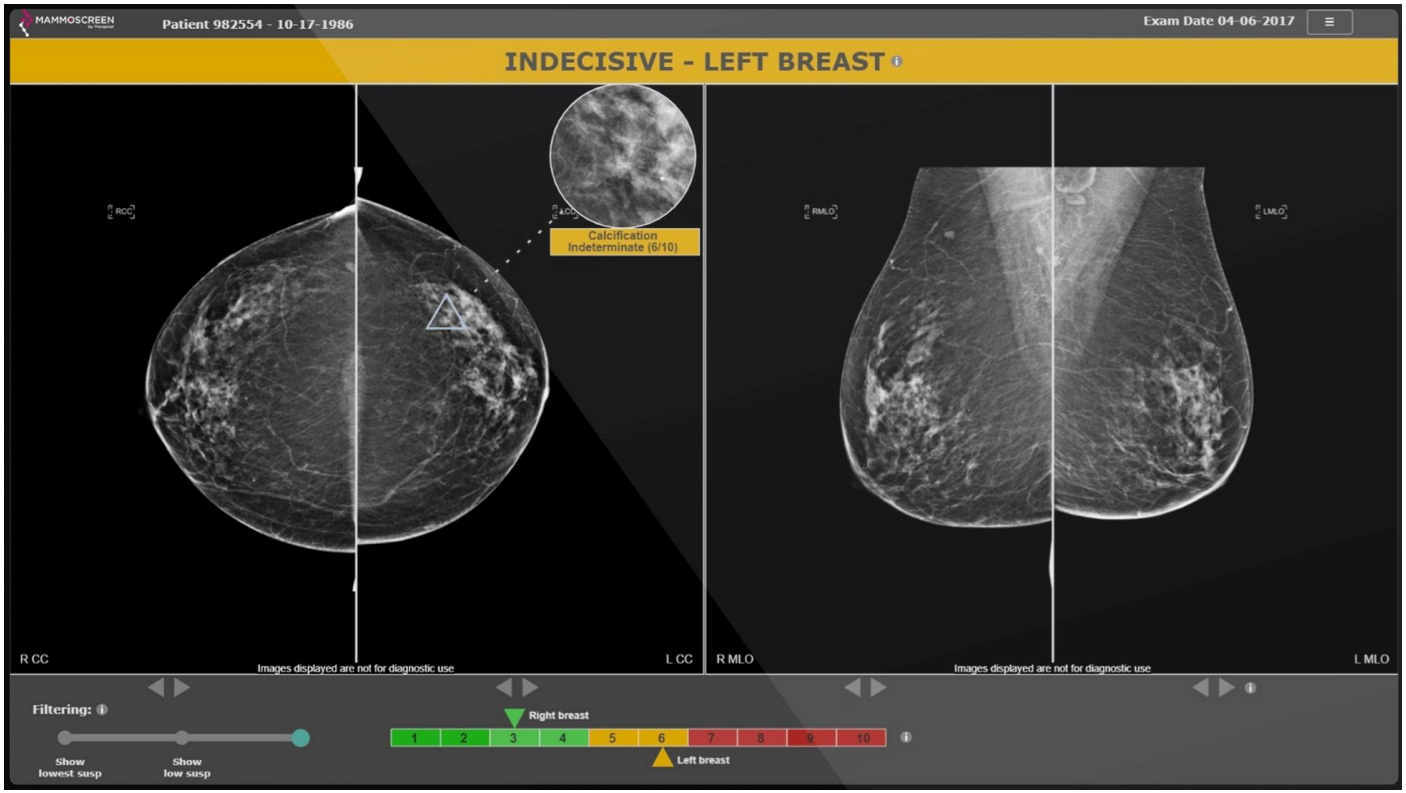
情報推薦，表示の改良，好みの予測



医療での人工知能の応用のニュース

- 画像診断について，さまざまなニュースがインターネットで公開されている.

胸部のX線画像について，自分自身で，人工知能アプリを動かし，ヒントを得る



MAMMOSCREEN社のニュース

https://www.mammoscreen.com/how-it-works

医学での人工知能の応用のニュース

血液検査による癌の検査（人間の免疫機能によって破壊されたがん細胞の検出）に関するニュース

https://www.theguardian.com/science/2020/mar/31/new-blood-test-can-detect-50-types-of-cancer

人工知能とゲーム

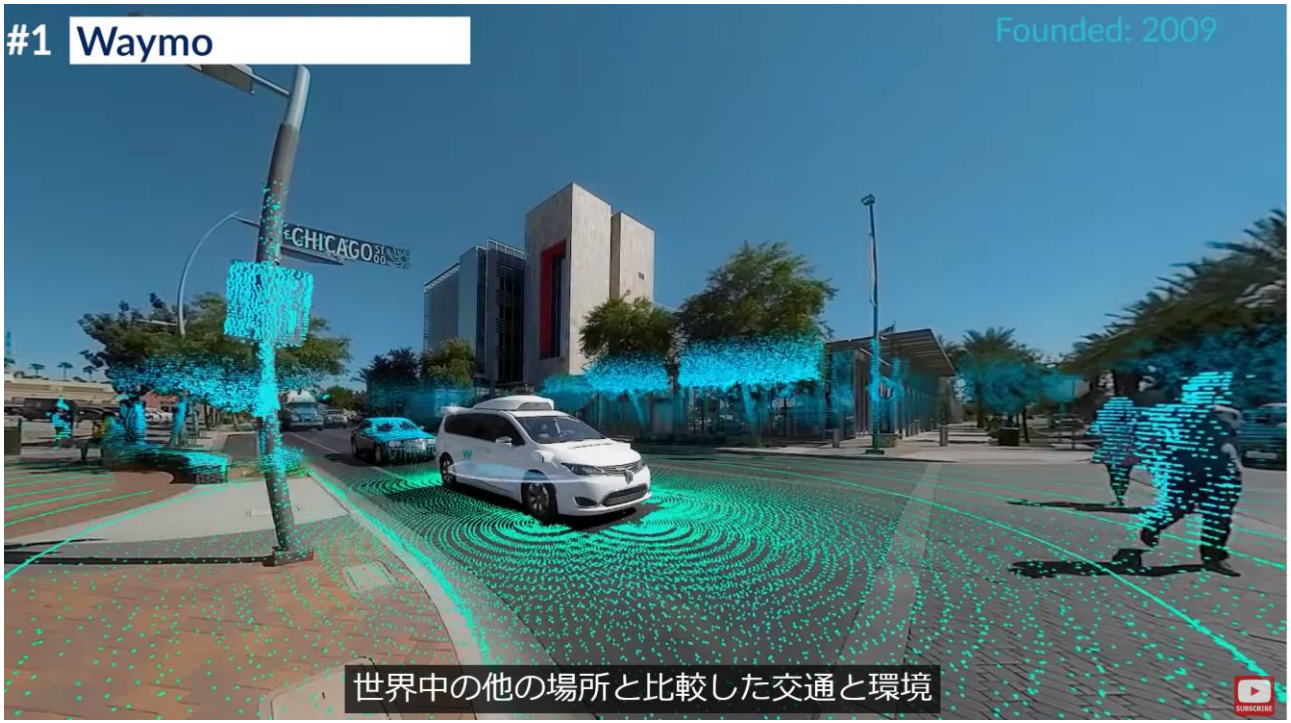
能力の向上が続いている.

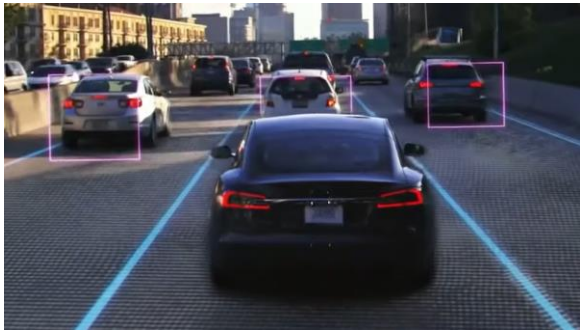
- 1997年：ディープ・ブルーがチェスの世界チャンピオンに勝利
- 2017年：自己対戦による学習能力を持つAlphaZeroが登場
- 2017年5月：AlphaZeroが囲碁で世界レベルの選手に対して3戦全勝

機械学習が、複雑なゲームでも、知的な戦略や判断を行う能力を持っている

自動運転のニュース

- 自動運転について，さまざまなニュースがインターネットで公開されている．





6 Autonomous Vehicles & Companies to watch in 2021-2022 | Self Driving Cars

https://www.youtube.com/watch?v=bdFi3RToOBk

人工知能による画像の合成

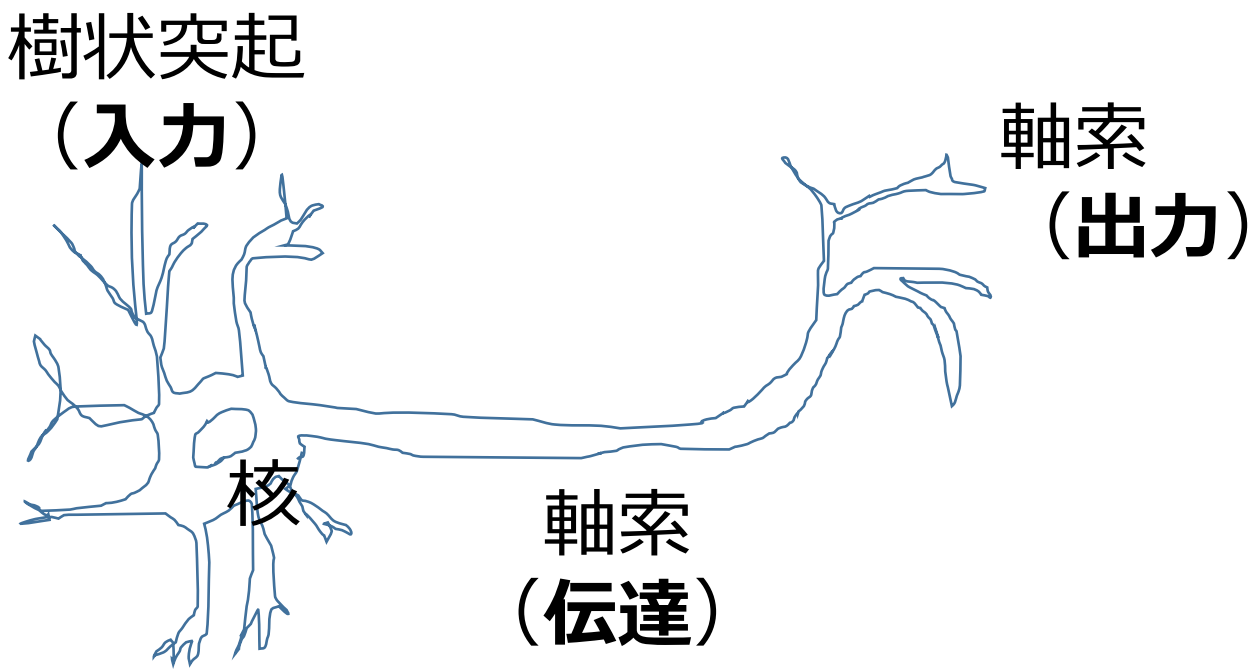
さまざまな種類の画像を，安定して，高精細に生成することが可能に

Large Scale GAN Training for High Fidelity Natural Image Synthesis
Andrew Brock, Jeff Donahue, Karen Simonyan, https://arxiv.org/abs/1809.11096v2

4-3. ニューロンと脳の構造

ニューロン

ニューロンは，**脳や神経系**において，**情報伝達，記憶，情報処理**を行う基本要素.



脳のニューロン

脳のニューロンの数（推定値）

- カタツムリ　11,000
- ロブスター　100,000
- アリ　250,000
- カエル　16,000,000
- ハツカネズミ 71,000,000
- タコ　500,000,000
- ネコ　760,000,000
- **ヒト　86,000,000,000**
  ～ 100,000,000,000
- アフリカゾウ 257,000,000,000

- 生物の種によって数が異なる
- 人間の脳には**約860億～1000億個**のニューロンが存在
- 複雑なネットワークを形成

Wikipeida の記事:
https://ja.wikipedia.org/wiki/%E5%8B%95%E7%89%A9%E3%81%AE%E3%83%8B%E3%83%A5%E3%83%BC%E3%83%AD%E3%83%B3%E3%81%AE%E6%95%B0%E3%81%AE%E4%B8%80%E8%A6%A7 より

脳や神経系の研究の進展

1900年頃: **脳や神経系**は，**ニューロンの集まり**と考えられるようになった．ニューロンが情報伝達を行うことも分かってきた.

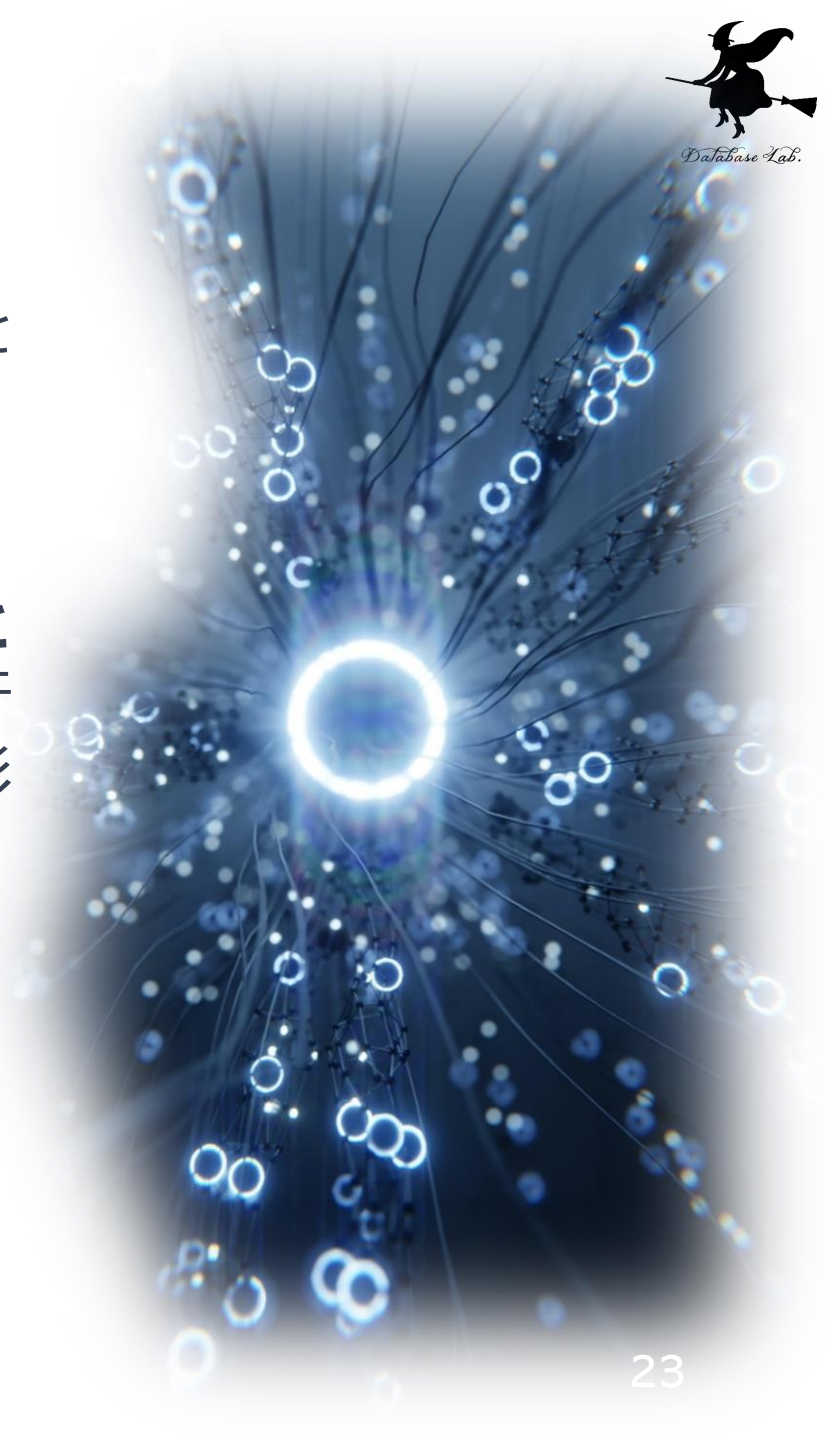
1980年頃: **脳機能マッピングによる脳の解明（どの部分がどの機能を持つか）**が進んだ．生きた脳を対象とした測定が行われるようになってきた.

2010年頃：**異なる脳領域の連携**が観察されるようになった.

脳の仕組みと機能について**深い洞察**が得られるように

ニューラルネットワーク

- **コンピュータを使用**
- **生物の脳の働きを模倣**することを目指した，**人工知能の一種**
- すでに広く普及している
- 仕組み：**入力の重みづけ，合計とバイアス，活性化関数の適用**を行う**ニューロン**が**ネットワーク**を形成



ニューラルネットワークの歴史

1940年代： **ニューラルネットワークの誕生**

1960年代： **バックプロパゲーションの誕生**（ニューラルネットワークの**学習の基本的な仕組み**）（1986年にRumelhartらによって再発見）

2010年代： 多層化，正規化，ReLU，ドロップアウト
など，ニューラルネットワークの**新技術**

ニューラルネットワークの進展傾向

年	コンピュータで扱えるニューロン数の規模
2010年	100,000個
2020年	2,000,000個
2030年	50,000,000個
2040年	1,000,000,000個
2050年	20,000,000,000個

所説あります

2055年ごろには1000憶を超えるかも？

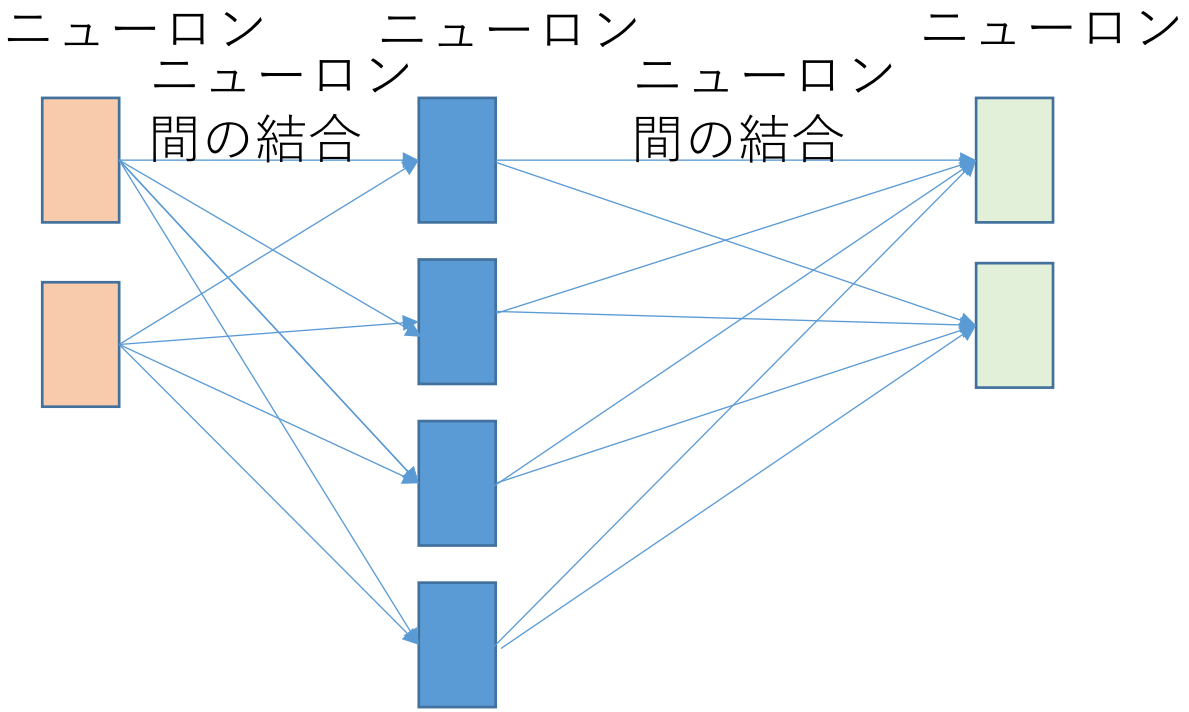
【将来傾向】

- **規模は増加傾向**
- GPU（プロセッサの一種），クラウドコンピューティング（大規模計算の基盤）の**発展**
- 人間の脳の**約860～1000億個**のニューロンを模倣することも、あながち不可能ではないかも

4-3. ニューラルネットワークの基本構造

ニューラルネットワークの仕組み

入力の重みづけ，合計とバイアス，活性化関数の適用を行う**ニューロン**が**ネットワーク**を形成



入力の重みづけ，合計とバイアス，活性化関数の適用

1. **入力の重みづけ**: **各入力値に対して**，それぞれの**対応する重み**を掛ける．例： **0.3**×**0.1**　**-0.5**×**0.8**　**0.2**×**-0.5**
2. **合計とバイアス**: **合計を計算**し，**バイアス**（数値）を加える．例 **0.03　-0.4　-0.1 の合計は -0.47．これに 0.2 を加える**
3. **活性化関数の適用**: 結果に，**活性化関数**を適用．**ニューロンの活性度**を得る（→次のニューロンの入力になる）

バイアス 0.2

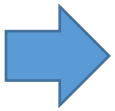
0.3　× 0.1　⇒ 0.03

-0.5　× 0.8　⇒ -0.4

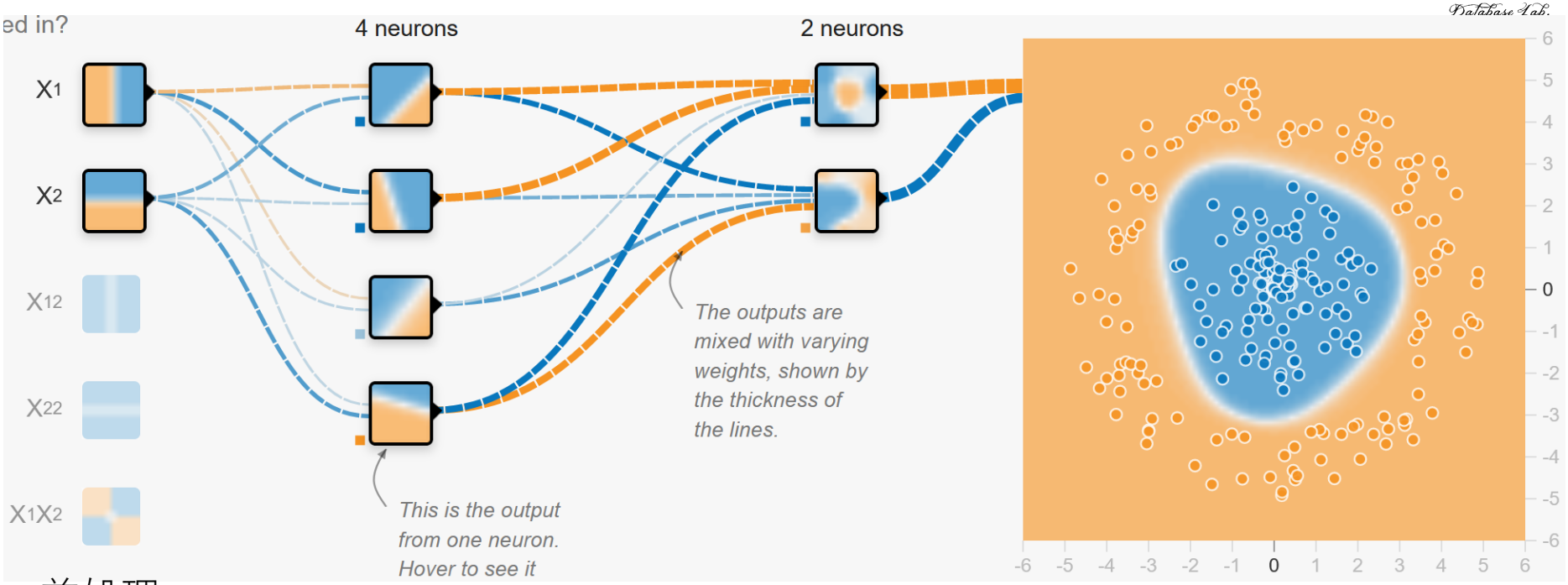
0.2　× -0.5　⇒ -0.1

入力　重み

合計 -0.47



-0.27 に活性化関数を適用



前処理
（データが青い部分にあれば活性化）

データが中央にあれば活性化

→結合→ 1層目 →結合→ 2層目

ニューラルネットワーク

URL: https://playground.tensorflow.org/

ニューロンの活性化と非活性化

ニューロンの出力が，プラスの大きな値
→　**活性化**している

ニューロンの出力が，０に近い値
→　**活性化**していない

4-4. ニューラルネットワークの種類、応用分野

ニューラルネットワークの種類

① オートエンコーダ（オートエンコーダ）

データを低次元の符号にマッピング

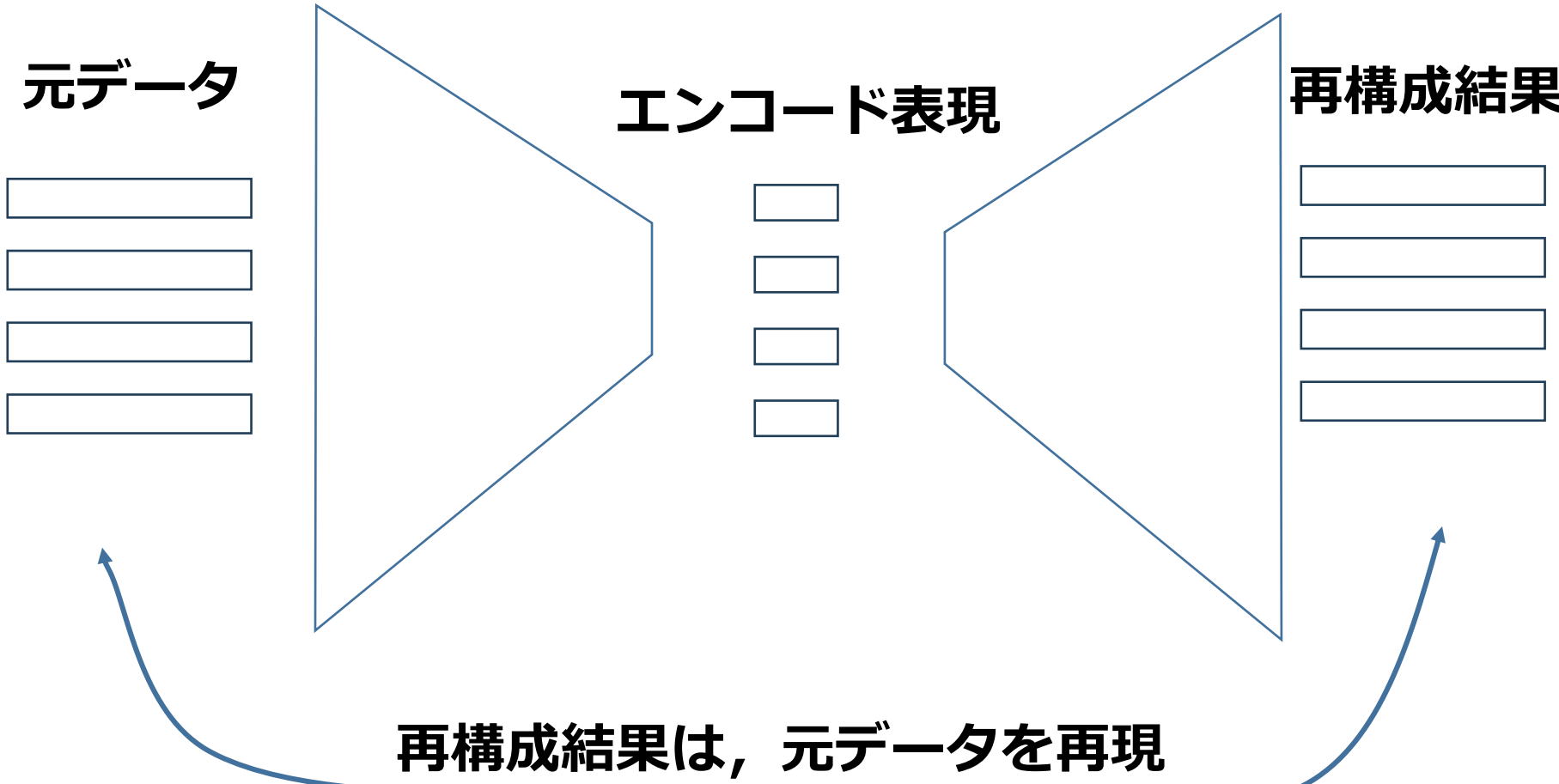
② 分類や予測

データから分類結果，予測結果を導く

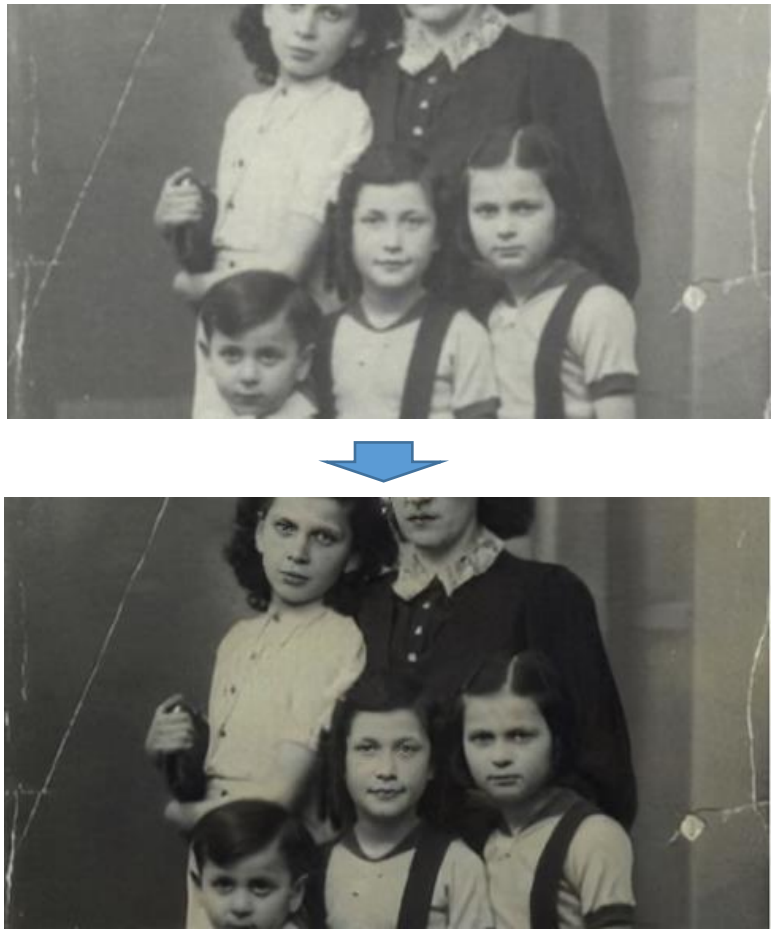
データを用いて学習を行うことは共通している

オートエンコーダでの符号化と再構築

- **元データ**を**圧縮**して**エンコード表現**を得る
- エンコード表現から再構成し，元データを再現
- **元のデータの最も重要な特徴を自動的に学習**



オートエンコーダの応用例



画像復元

写真からの顔の３次元化

オートエンコーダの応用分野

- **特徴抽出**

オートエンコーダは、データを低次元の符号にマッピングするので，特徴抽出に活用できる

- **データ圧縮**

低次元の符号化表現は元のデータよりも効率的な表現となる

- **ノイズ除去**

ノイズのあるデータを与え、ノイズのないデータを復元できる

- **異常検出**

正常なデータのパターンを学習し、異常なデータを検出するために使用できる

オートエンコーダの例

元のデータ

２５画素の白黒画像

11111

10001

10001

10001

11111

オートエンコーダの圧縮と再構成

元のデータ

８個の数値

１０１０１０１０

エンコード表現

元のデータを**圧縮**

0.993 0.838 0.995

再構成データ

８個の数値を再構成可能（完ぺきではない）

0.104 0.997 0.999,
0.968 0.997 0.945,
0.975 0.000

オートエンコーダの圧縮と再構成

１．データの圧縮と再構成

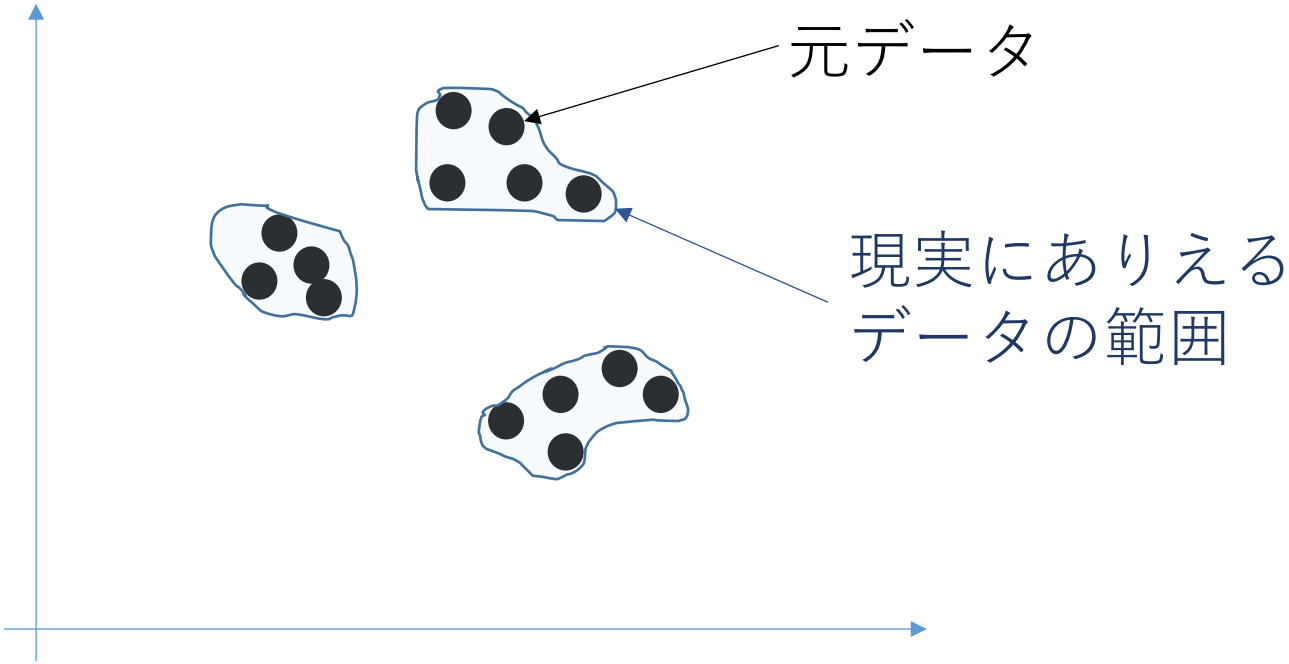
- 入力: [1, 0, 1, 0, 1, 0, 1, 0]
- 再構成: [0.998, 0.028, 0.979, 0.016, 0.985, 0.017, 0.975, 0.004]
- 高精度な再現: 1 → 0.9以上, 0 → 0.03未満

２．エンコード表現によるサイズ削減

- 元データは８つの数値 → エンコード表現は３つの数値
- エンコード例: [1, 0, 1, 0, 1, 0, 1, 0] → [0.993, 0.838, 0.995]
- 本質的特徴を保持しつつデータ圧縮

オートエンコーダによるデータ生成

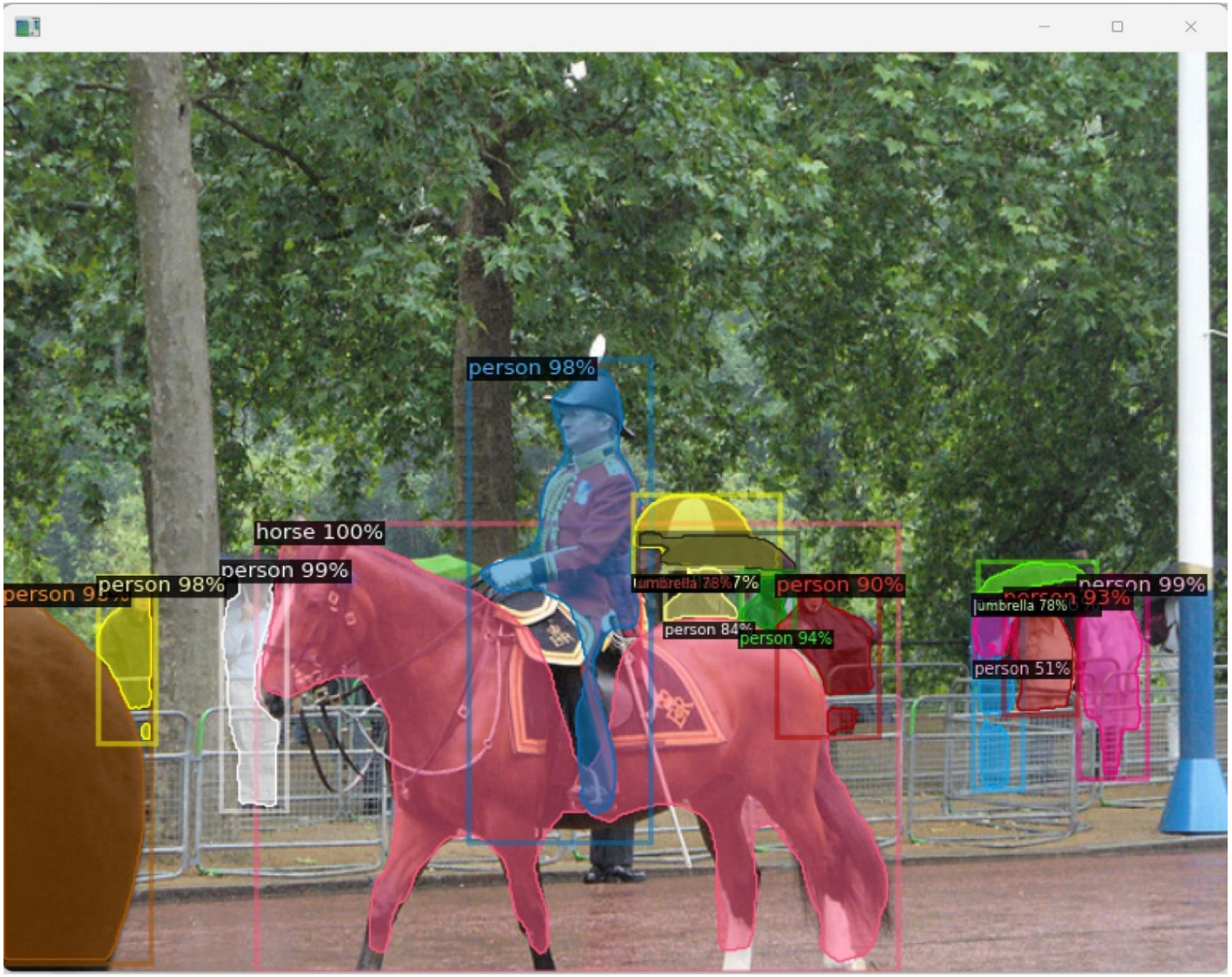
オートエンコーダの結果として，**再構成（エンコード表現からの元データの再構成）ができる**ことから，「**現実にあり得るデータを生成できる能力を獲得**」と考えることもできる



ここまでのまとめ

- **オートエンコーダの基本**
  - データ圧縮と再構成*:* データの重要な特徴を保持し、効率的に再構成。
  - **教師なし学習の一種*: ラベルなし*で特徴抽出**を行う。
- **オートエンコーダの機能**
  - ***ノイズ除去***: データからノイズを削減し、元のパターンに近づける。
  - **データ生成**: ランダムな潜在表現から新しいデータを作成可能。
- **応用例**: ノイズ除去、異常検知、情報検索、画像生成。
- **オートエンコーダの種類**: 通常型、変分(VAE)、積層型、畳み込み型。

分類の応用例



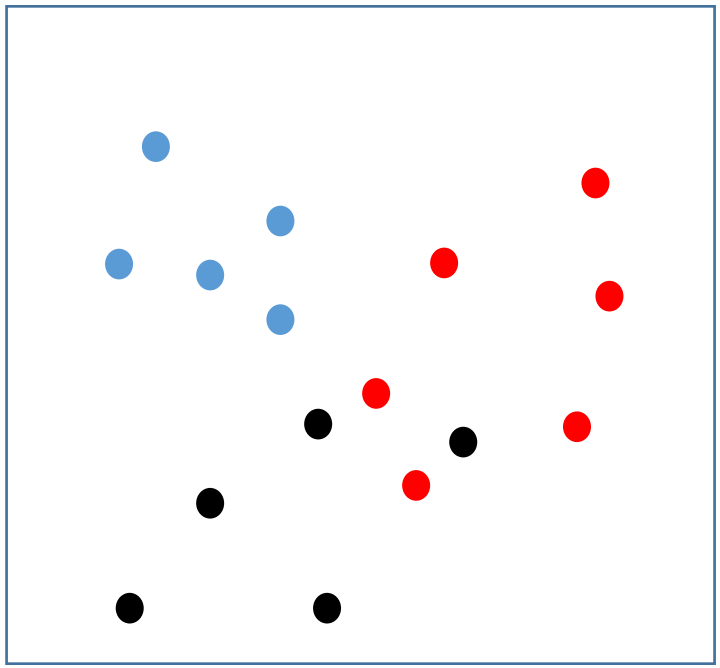
物体検知、セグメンテーションなどの画像認識

機械学習における分類

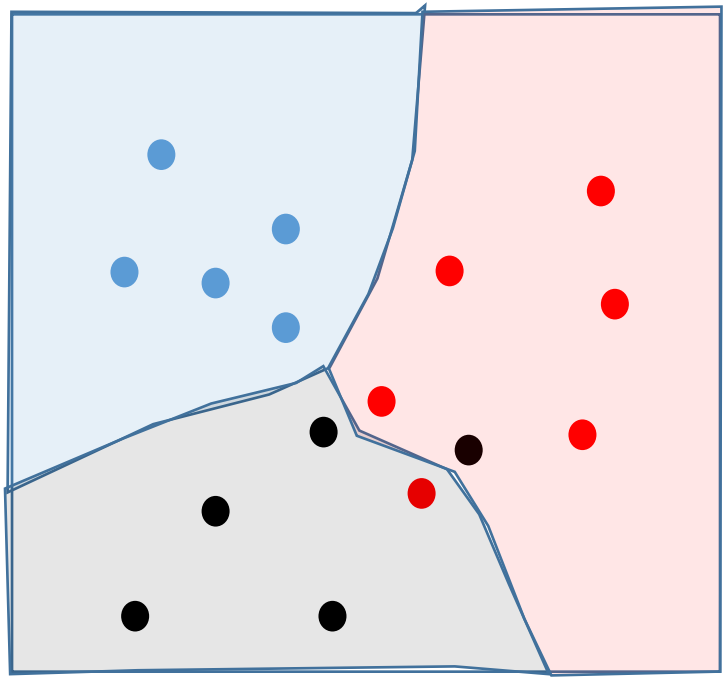
分類：データを，あらかじめ分かっている種類（例：自動車，人間，自転車）に分類すること

主要プロセス①　訓練データからのモデル構築

訓練データ



構築されたモデル



分類や予測の応用分野

人工知能による**分類や予測の応用分野**は**多岐にわたる**

- **画像認識と画像理解**

道路標識の認識や顔認識、物体検出、画像セグメンテーションなど。自動運転技術や監視カメラの映像解析などに応用。

- **自然言語処理**

要約、校正、質問応答、翻訳、音声認識、音声合成、感情分析、文書分類、アイデア出しなど。

- **バイオインフォマティクス**

遺伝子解析やタンパク質構造の予測、薬剤の設計など。

- **金融と経済予測**

市場のトレンド分析やリスク評価、予測モデルの構築、意思決定のサポートなど。

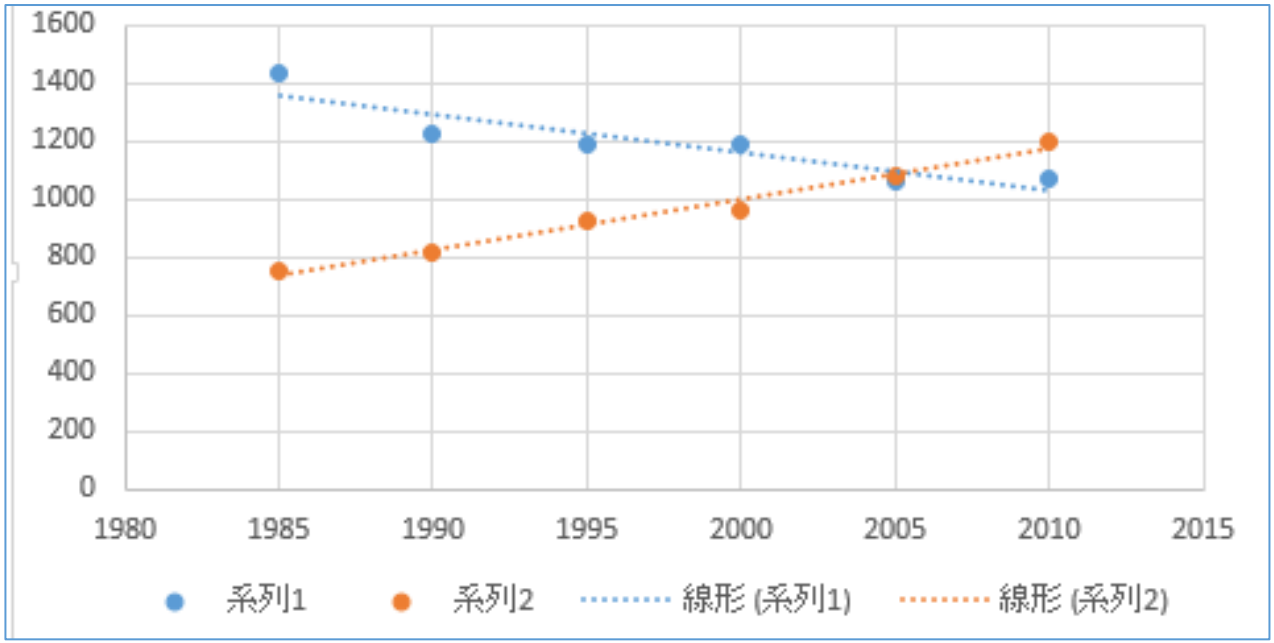
- **医療診断と予測**

疾患の診断や予測、画像診断（MRIやCTスキャン）、病気の予後予測、薬物副作用の予測など。

4-5 線形近似（学習の例）

線形近似

- **線形近似**は、**データを直線で近似**する手法
- **データ点が多い**ほど、**より正確な近似結果**
- 線形近似はデータの特徴を捉える一種の学習手法としても考えることができる

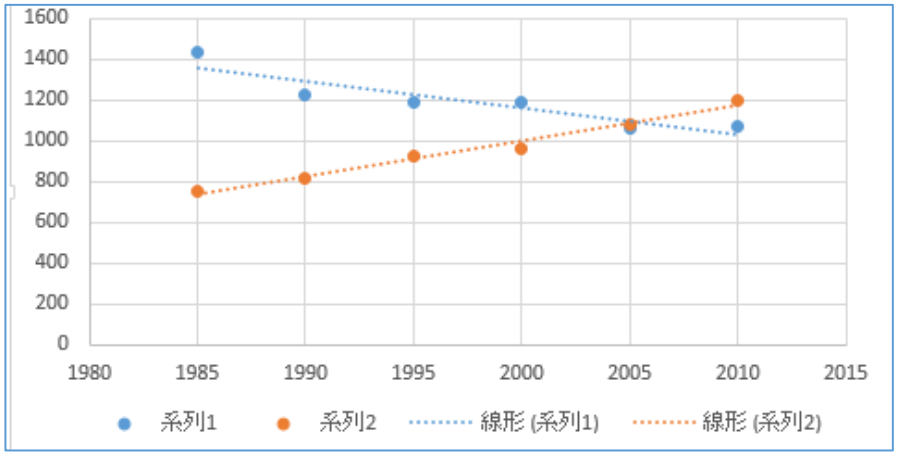


散布図＋線形近似

Excel での線形近似

年次	出生数（千人）	死亡数（千人）
1985	1432	752
1990	1222	820
1995	1187	922
2000	1191	962
2005	1063	1084
2010	1071	1197

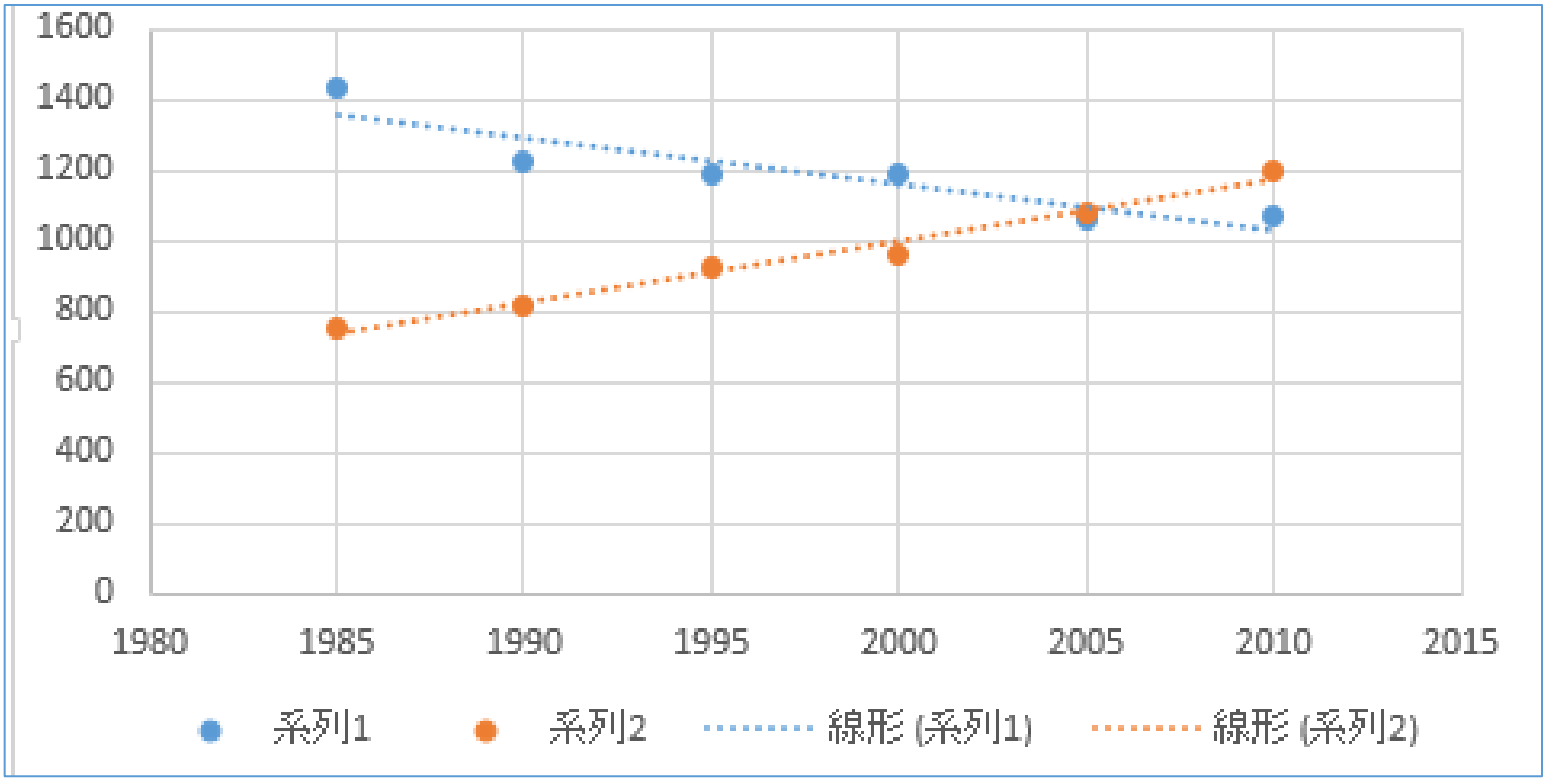
入力　　出力



散布図＋線形近似

出生数，死亡数の推移
出典：総務省「第63回 日本統計年鑑 平成26年」

線形近似の例



線形近似とは，データの分布を近似した補助線

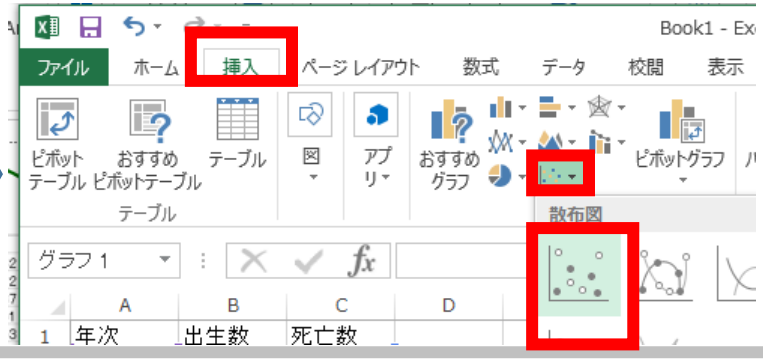
Excel での散布図＋線形近似の作成手順（１／３）

	A	B	C
1	年次	出生数	死亡数
2	1985	1432	752
3	1990	1222	820
4	1995	1187	922
5	2000	1191	962
6	2005	1063	1084
7	2010	1071	1197

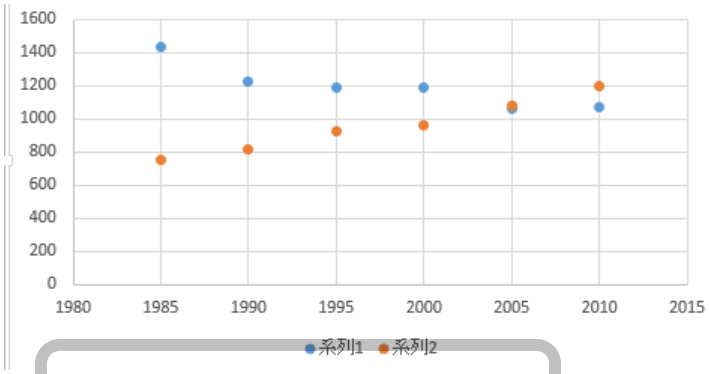
元データ

	A	B	C
1	年次	出生数	死亡数
2	1985	1432	752
3	1990	1222	820
4	1995	1187	922
5	2000	1191	962
6	2005	1063	1084
7	2010	1071	1197

① **データ部分**を**範囲選択**

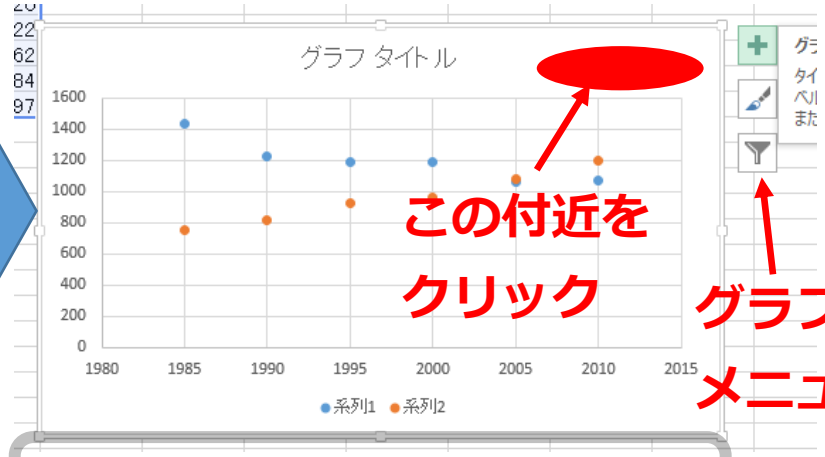


② リボンで「**挿入**」→散布図の選択

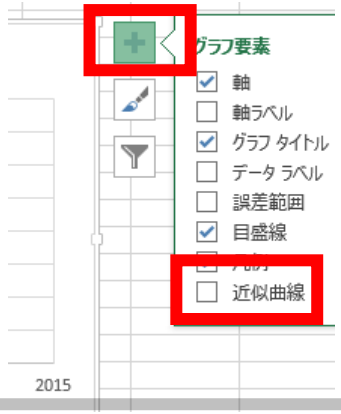


③ 散布図を確認

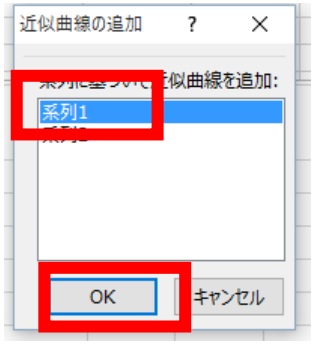
Excel での散布図＋線形近似の作成手順（２／３）



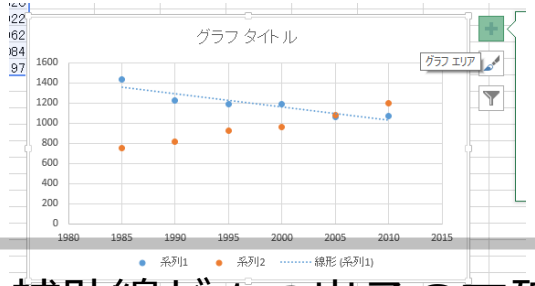
④　グラフメニューを出す



⑤　「＋」のボタンをクリックしたのちに，「近似曲線」をクリック

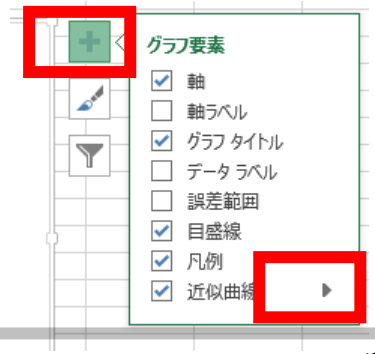


⑥「**系列１**」をクリックして「**ＯＫ**」

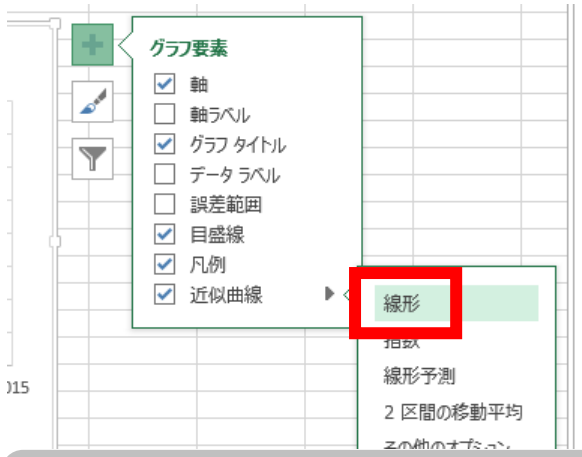


⑦　補助線が１つ出るので確認

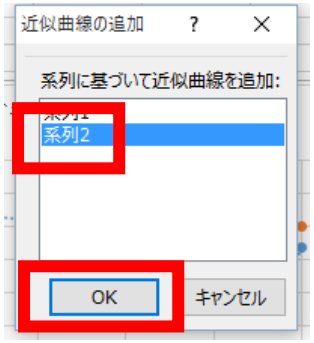
Excel での散布図＋線形近似の作成手順（３／３）



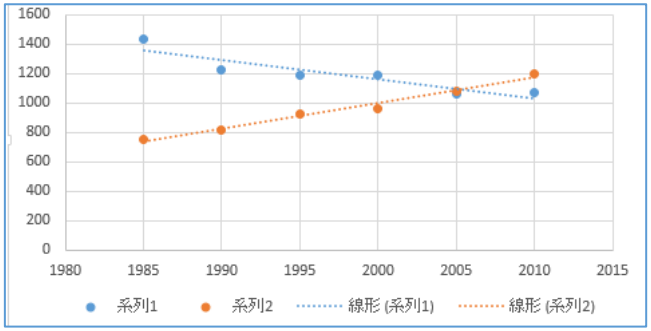
⑧　もう１度，**グラフメニュー**を出し，**近似曲線の右横の「▶」を展開**



⑨　「**線形**」を選ぶ



⑩「**系列２**」をクリックして「**ＯＫ**」



完成

4-6 最適化

最適化

- **最適化**は、与えられた問題に対して**最適な解を求めるプロセス**
- 最適化は、**パラメータや変数の値などを調整**して、目的関数（ゴールともいう）と呼ばれるものを**最小化または最大化**する

ゴールの例：　誤差の最小化

パラメータの例：　直線の上下の位置と，直線の傾き

最適化により方程式を解く

- **次の式の値が最小**になるように，x **の値を定めたい**．但し，$N = 5$ とする．

$$f(\mathbf{x}) = \sum_{i=2}^{N} 100\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2.$$

- https://docs.scipy.org/doc/scipy/reference/tutorial/optimize.html

① 最適化により方程式を解く Python プログラム


```
import numpy as np
from scipy.optimize import minimize
def rosen(x):
    """The Rosenbrock function"""
    return sum(100.0*(x[1:]-x[:-1]**2.0)**2.0 + (1-x[:-1])**2.0)

x0 = np.array([1.3, 0.7, 0.8, 1.9, 1.2])
res = minimize(rosen, x0, method='nelder-mead',
    options={'xtol': 1e-8, 'disp': True})
print(res.x)
```

① 最適化により方程式を解く Python プログラム


```
import numpy as np
from scipy.optimize import minimize
def rosen(x):
    """The Rosenbrock function"""
    return sum(100.0*(x[1:]-x[:-1]**2.0)**2.0 + (1-x[:-1])**2.0)

x0 = np.array([1.3, 0.7, 0.8, 1.9, 1.2])
res = minimize(rosen, x0, method='nelder-mead',
    options={'xtol': 1e-8, 'disp': True})
print(res.x)
```

```
Optimization terminated successfully.
         Current function value: 0.000000
         Iterations: 339
         Function evaluations: 571
[1. 1. 1. 1. 1.]
```

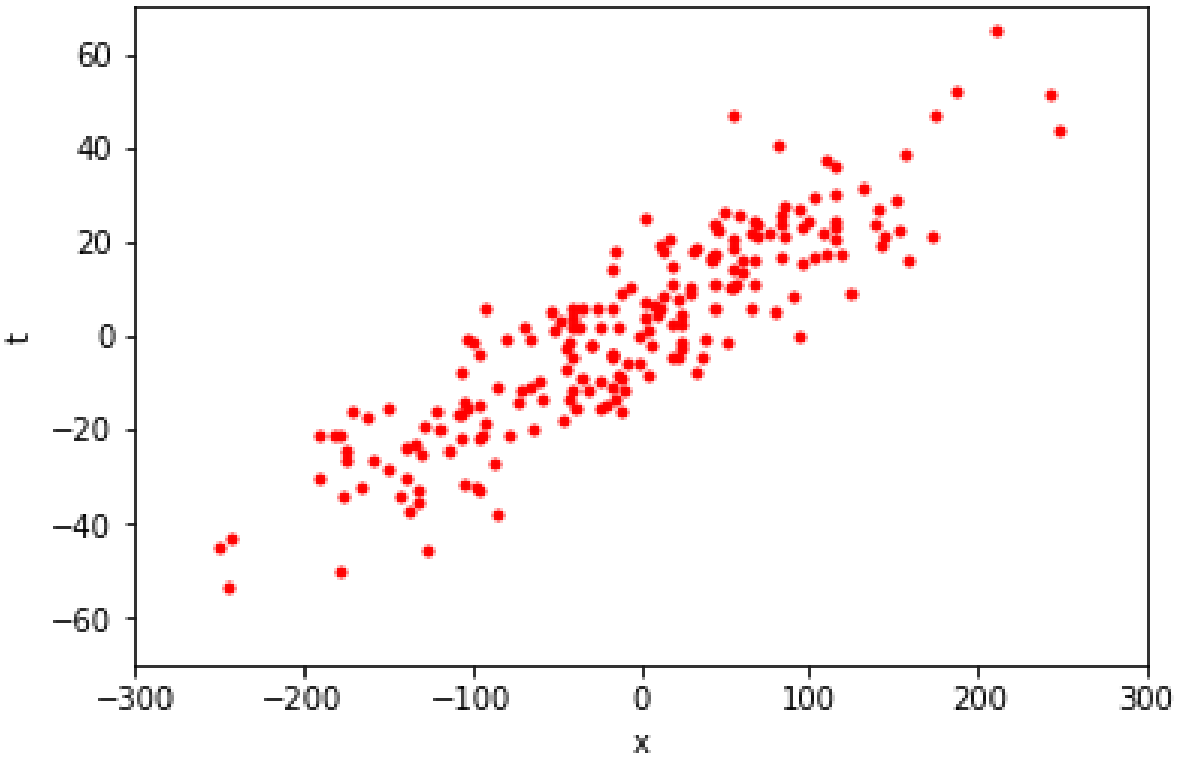
x = [1 1 1 1 1] のとき（すべての値が 1のとき）
最適であると求まった.

線形近似での最適化

線形近似では**最適化**が行われ、**最適な直線の探索**が行われる

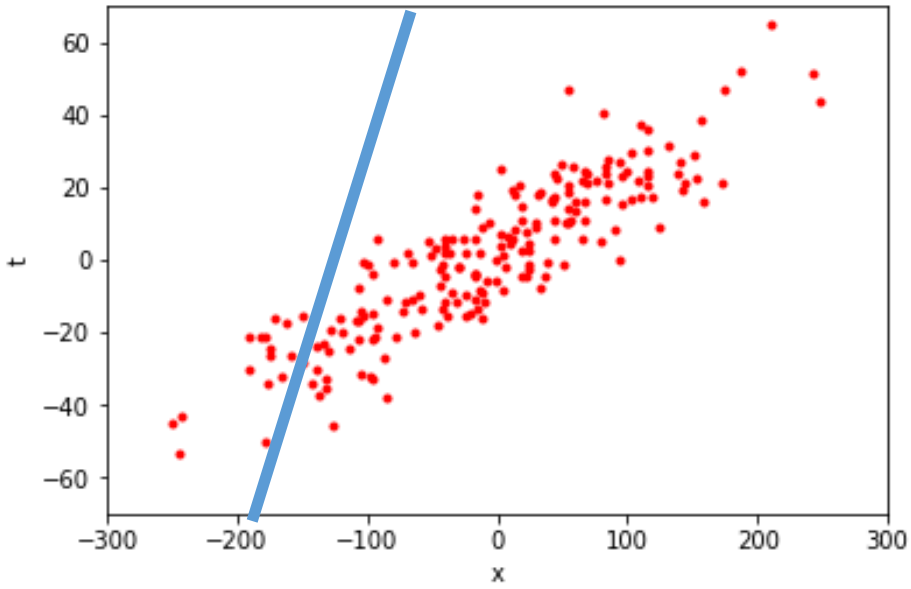
- 線形近似は、**データを直線で近似**することを目指す
- **最適な直線**を探索するために、**最適化**の手法を使用
- 直線を適切に調整することで、**データに最もよくフィットする直線を見つける**

データの例

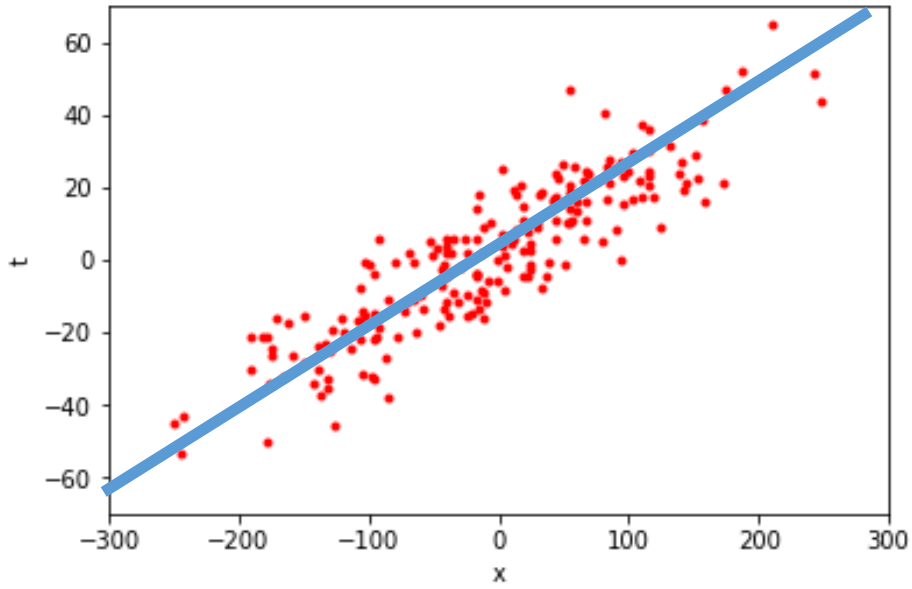


- **多数のデータの集まり**
- 上の図では，点１つで，１つのデータ

直線による近似の例

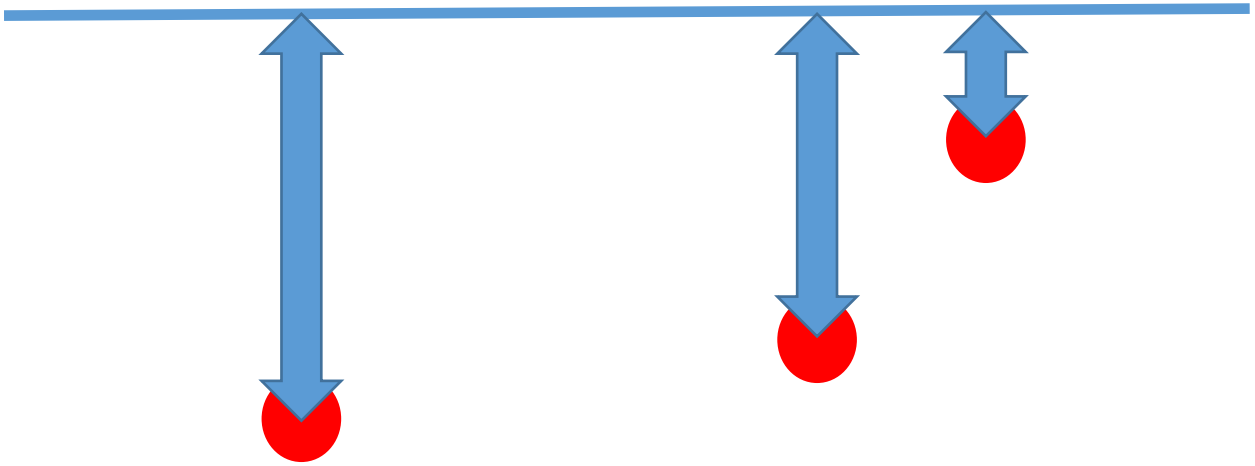


最適でない近似
（誤差大）



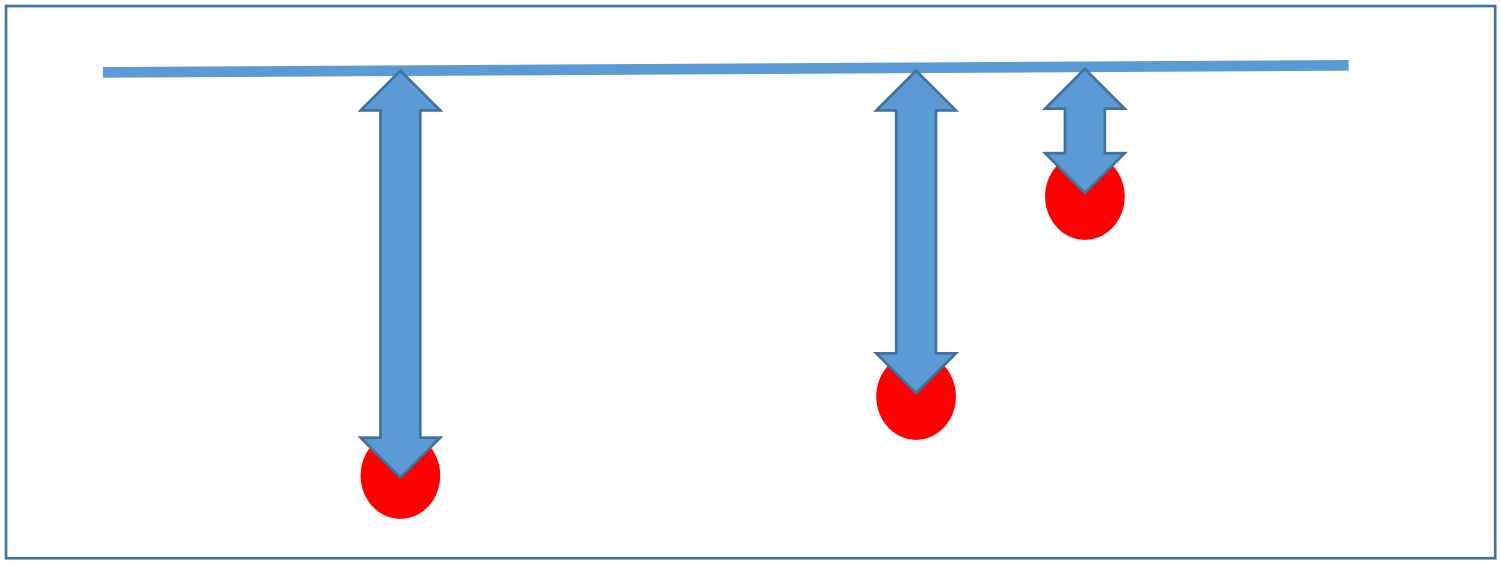
最適な近似
（誤差小）

誤差

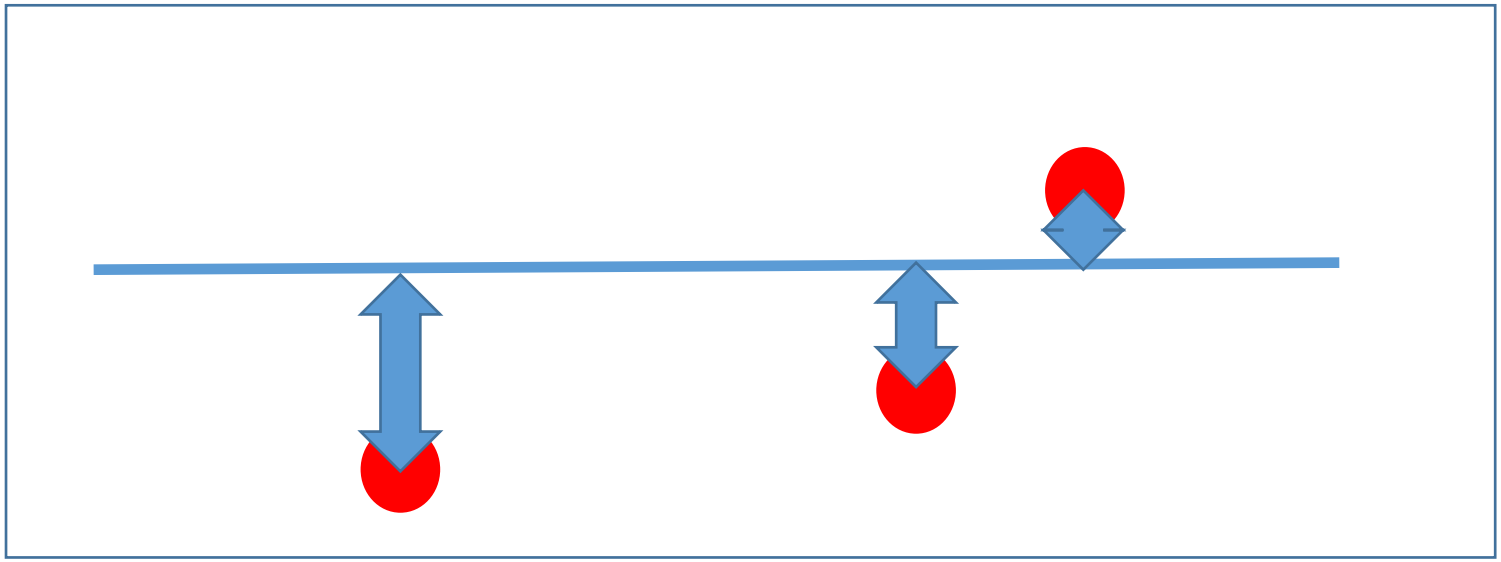


赤点：元データ

直線の上下移動による誤差の変化



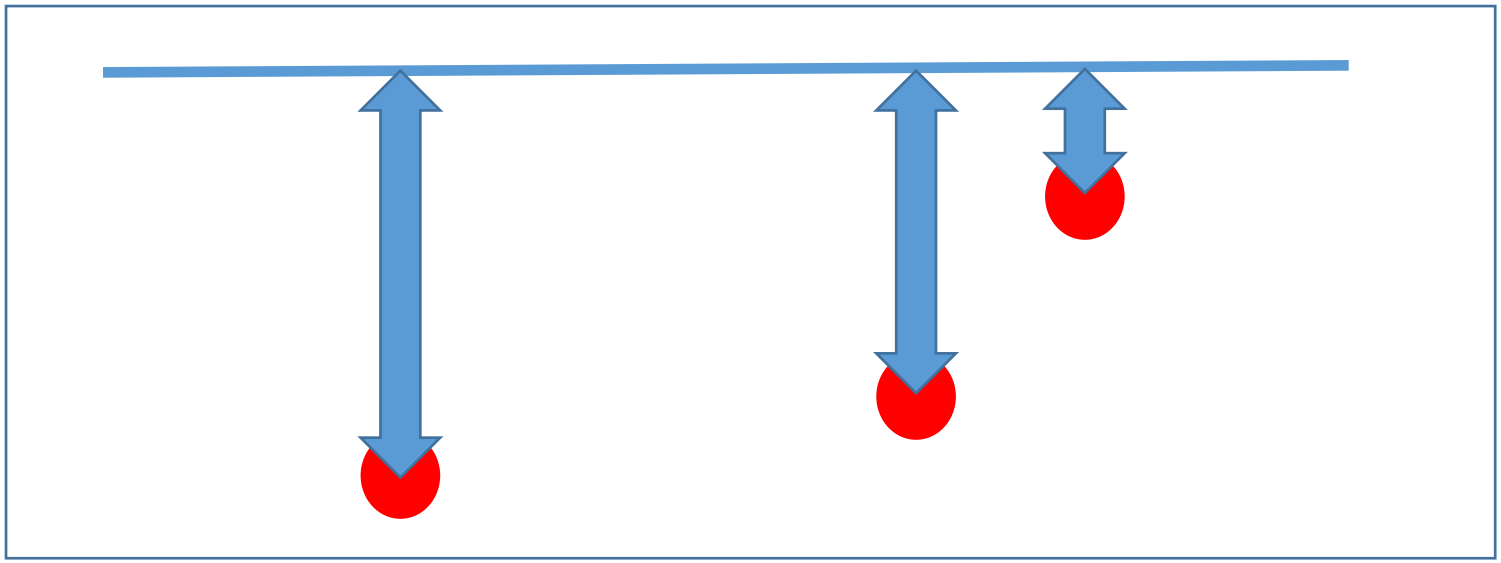
誤差大



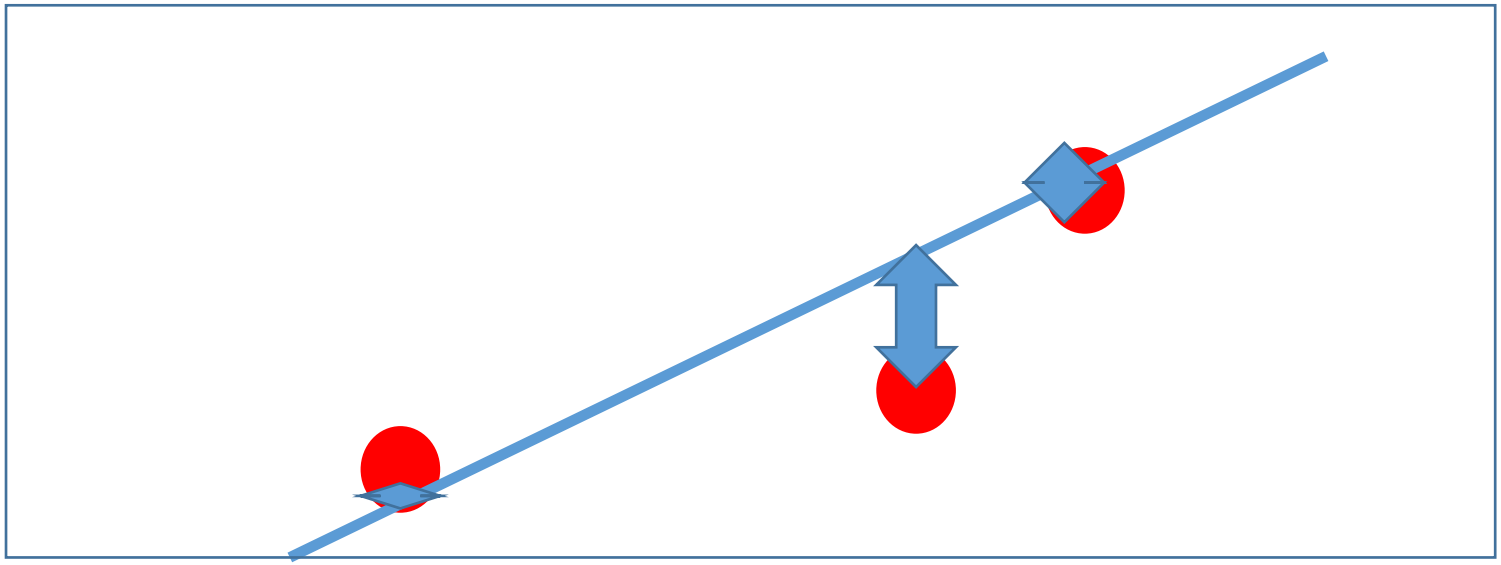
誤差小

赤点：元データ

直線の傾きの変化による誤差の変化



誤差大

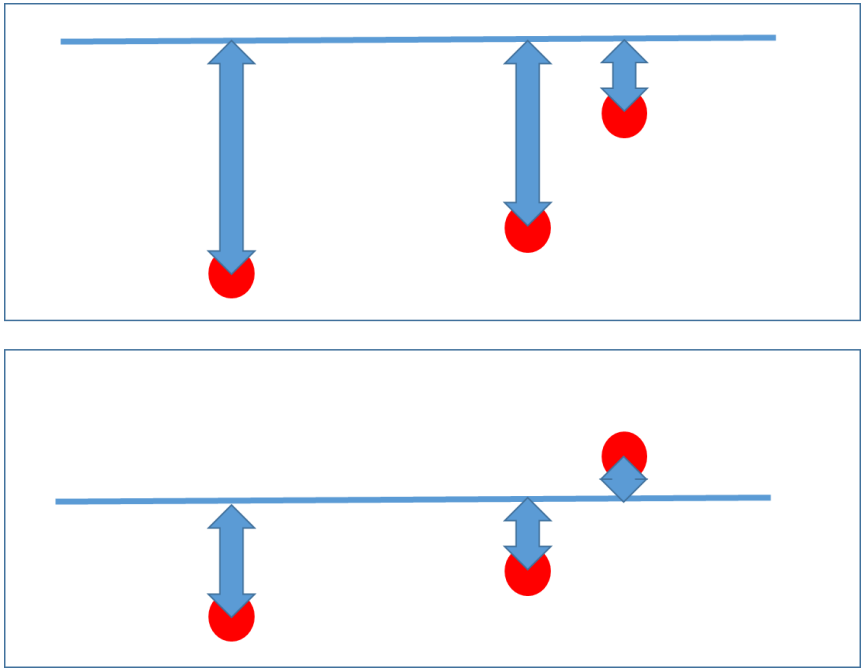


誤差小

赤点：元データ

誤差の変化

- 直線の上下移動や，傾きの変化により，**誤差**が変化



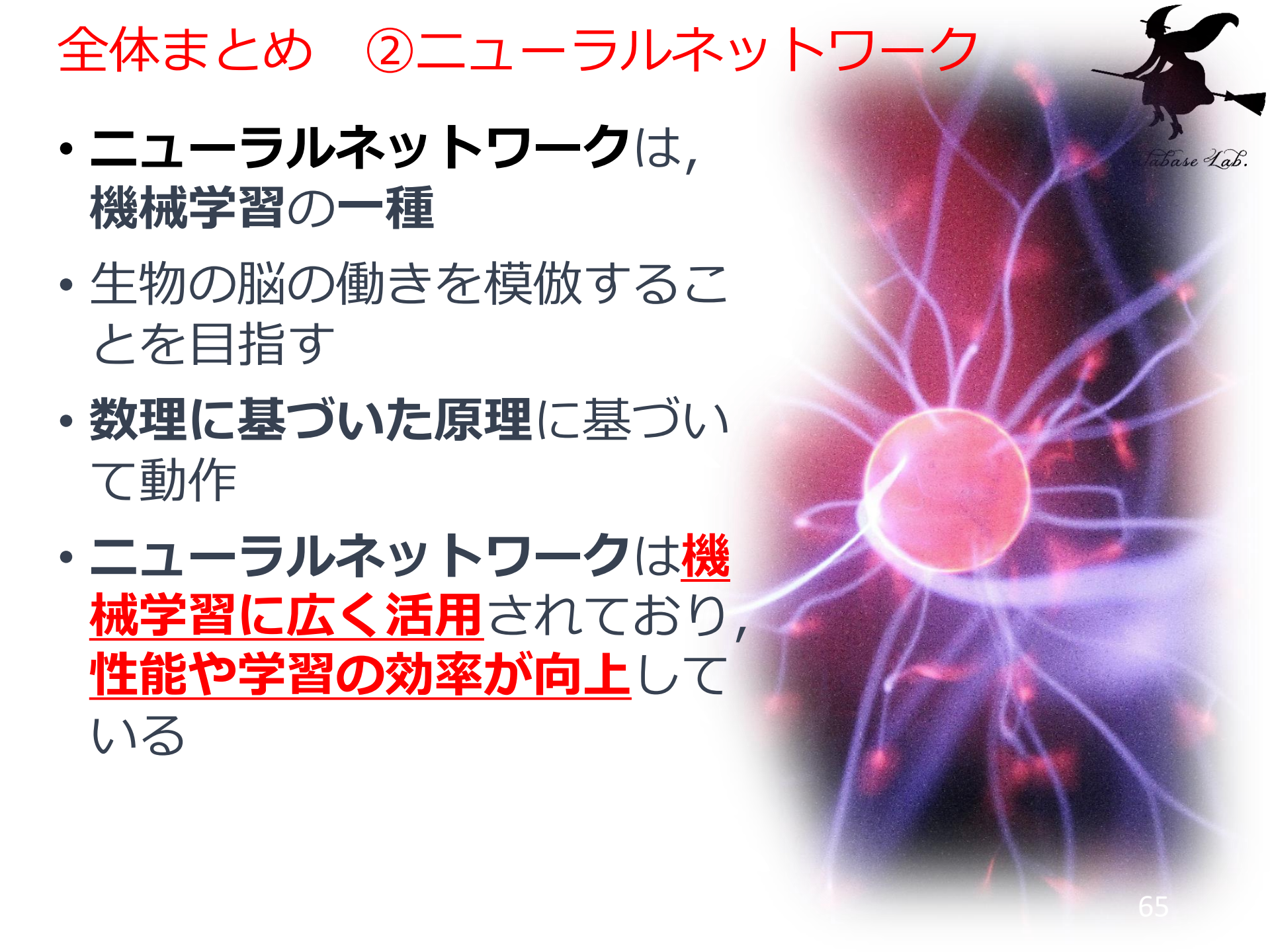
最適化は機械学習でも重要

- **ニューラルネットワーク**は，**学習**の過程で，**最適な状態になるように調整**される
- **最適化**は，**ニューラルネットワーク**の**学習**で，重要な役割を果たす
- **ニューラルネットワーク**のパラメータ（重みやバイアス）を**最適化**することで、**性能を向上**させる

全体まとめ　①機械学習

- **機械学習**は，**コンピュータ**が**データ**を使用して**学習**することより**知的能力を向上**させる技術
- 情報の抽出能力，ルールや知識のプログラム化不要などの**特徴**を持つ
- 他の方法では**解決が難しかった問題に取り組むことが可能**に

全体まとめ　②ニューラルネットワーク



- **ニューラルネットワーク**は，**機械学習の一種**
- 生物の脳の働きを模倣することを目指す
- **数理に基づいた原理**に基づいて動作
- **ニューラルネットワーク**は**機械学習に広く活用**されており，**性能や学習の効率が向上**している