

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



機械学習とニューラルネットワークの基礎

機械学習とは

データからパターンを自動発見し、ルールを自動生成する技術

教師あり学習

入力 x + 正解 t → 学習 → 未知データの予測 (大量のペア)

応用分野

画像認識
(自動運転・顔認識)

自然言語処理
(翻訳・要約・音声認識)

医療診断
(MRI・CT・がん検出)

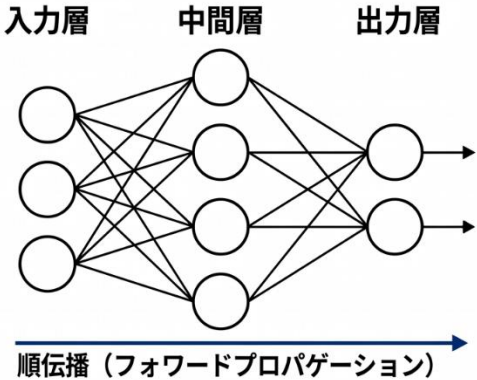
金融予測

ゲーム

情報推薦
(宿泊サイト等)

能力：分類・検出・認識・予測・合成・翻訳

ニューラルネットワークの仕組み

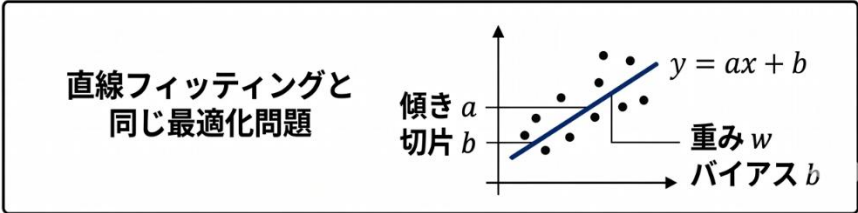


ニューロン

- ① 入力 \times 重み w
- ② 合計 + バイアス：
 $u = \sum(w \cdot x) + b$
- ③ 活性化関数：
 $y = f(u)$

脳のニューロン (約860億~1000億個) を模倣

学習 = 最適化



4-1. 機械学習と教師あり学 習

機械学習の定義と特徴



機械学習は、**コンピュータ**が**データ**を使用して**学習**し**知的能力を向上**させる技術

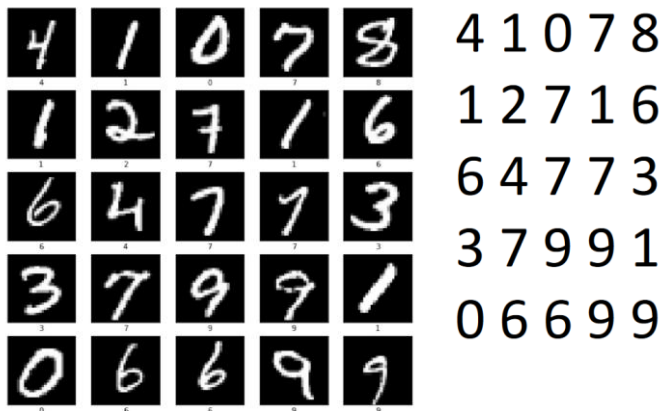
- **情報の抽出**：データからパターンや関係性を自動で発見する能力
- **簡潔さ**：人間が設定していたルールを自動生成できる
- **限界の超越**：他の方法では困難だった課題にも解決策を得られる可能性がある



機械学習プログラムの構成



訓練データ



例：手書き数字の
画像60000枚

それぞれの正解
(4, 1, 0, 7, 8, ...)
60000個

⇒訓練データとして使用する。

学習のプログラム

プログラム

```
[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)

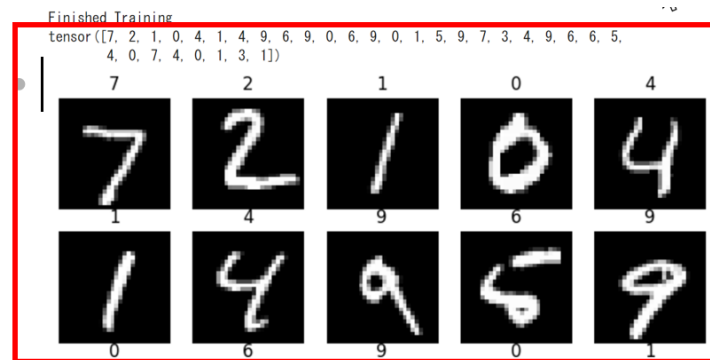
# ニューラルネットワークの定義
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(4, 10) # 入力は4次元 (Irisの特徴量)
        self.fc2 = nn.Linear(10, 3) # 出力は3クラス

    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

学習ののち、新たな手書き数字の画像分類を行うことができる。

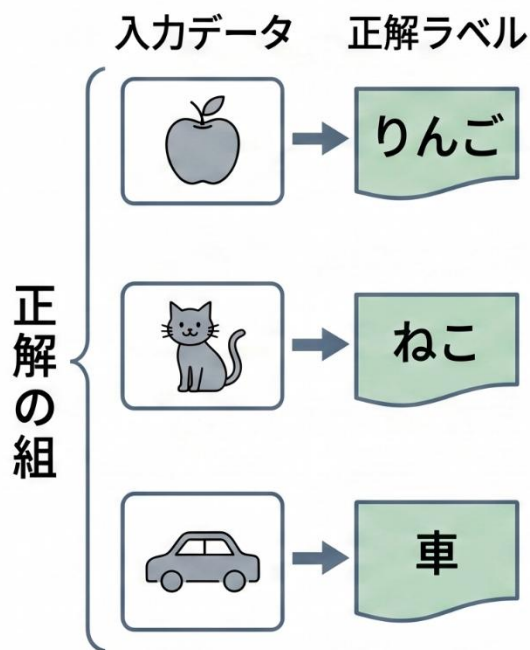


教師あり学習の仕組み



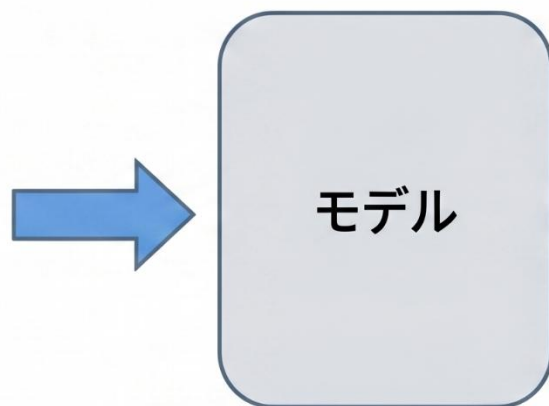
正解の組を用いる方式を 教師あり学習 と呼ぶ。
(このほかに教師なし学習や強化学習といった方式もある。)

1. 大量の訓練データ



2. 学習プロセス

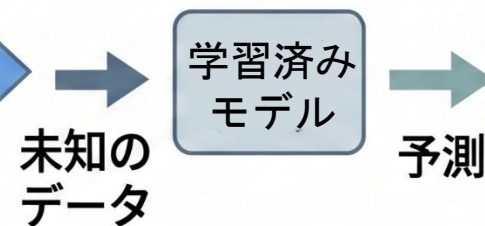
大量の訓練データを用いて学習を行う。



正解の組を用いる方式を教師あり学習と呼ぶ。

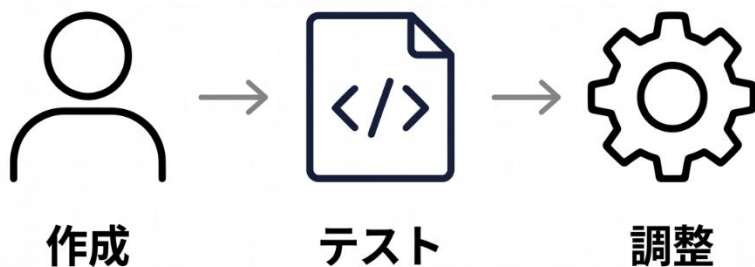
3. 学習済みモデル

未知のデータから予測が可能になる。



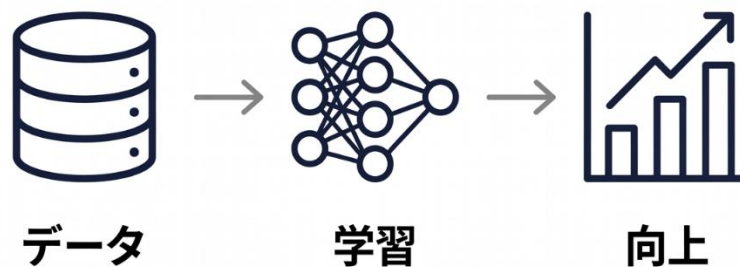
一般のプログラミングと機械学習の違い

一般のプログラミング



人間がプログラムを
作成・テスト・調整する

機械学習



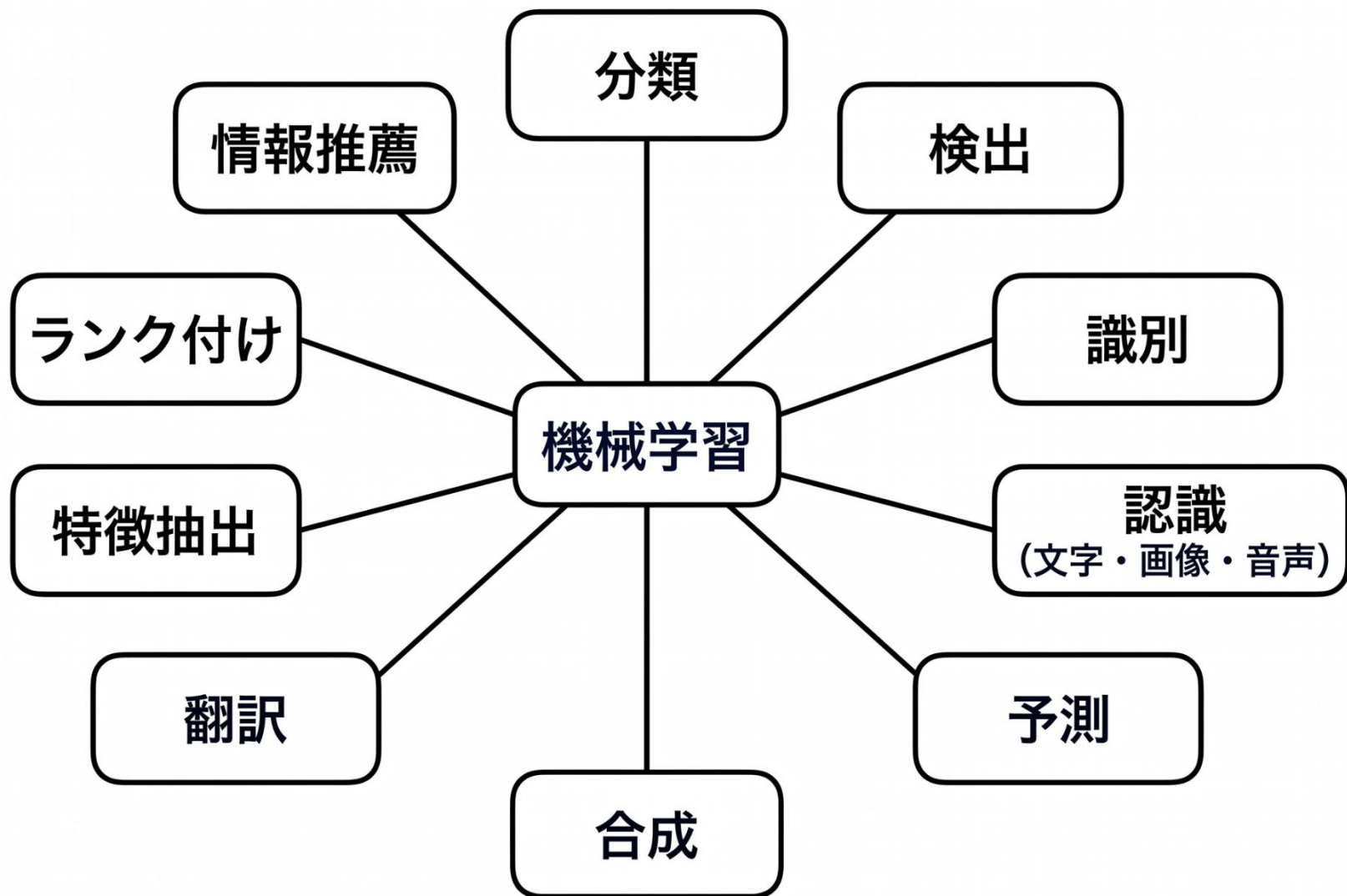
データを利用して
知的能力を向上させる



4-2. 機械学習の現状

機械学習の能力の多様性

機械学習は、データから学習することで、さまざまな能力を獲得する。

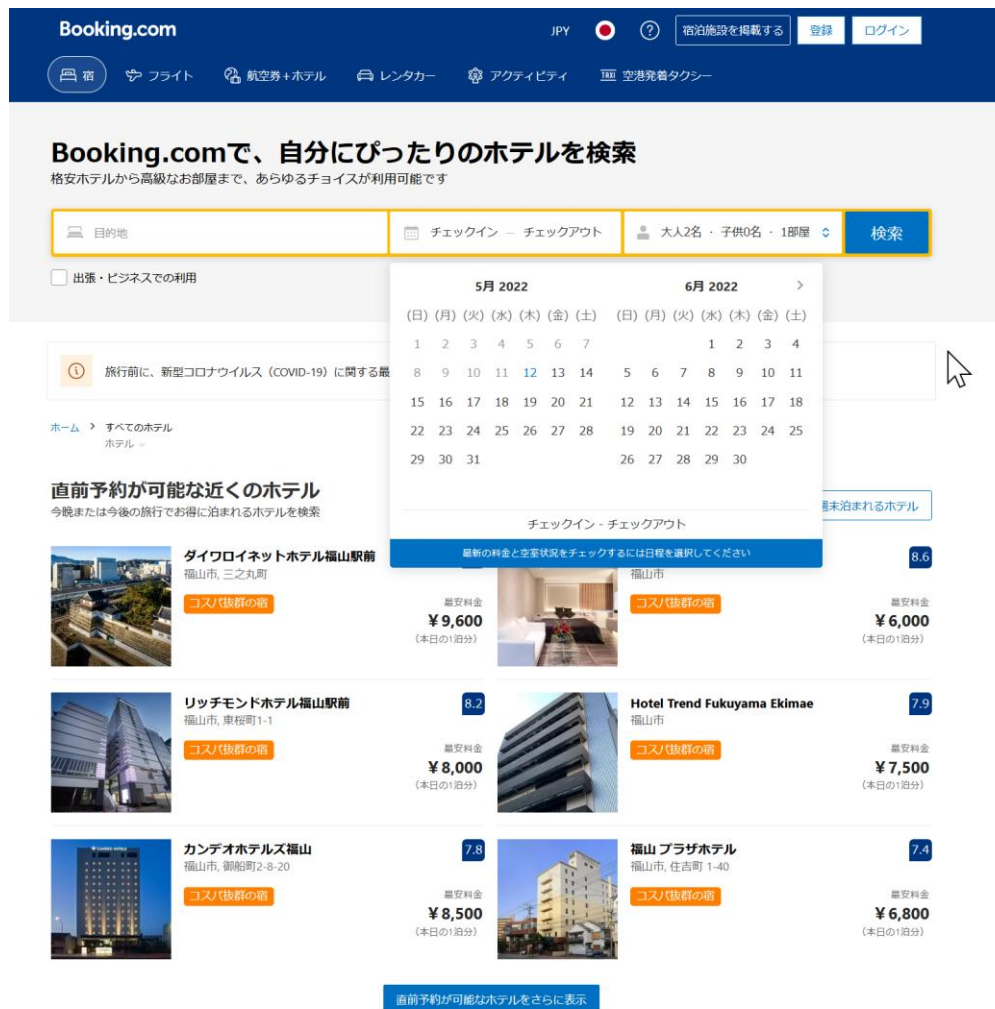


- **画像認識と画像理解**：道路標識の認識や顔認識，物体検出，画像セグメンテーションなど。自動運転技術や監視カメラの映像解析などに応用される。
- **自然言語処理**：要約，校正，質問応答，翻訳，音声認識，音声合成，感情分析，文書分類，アイデア出しなどに活用される。
- **バイオインフォマティクス**：遺伝子解析やタンパク質構造の予測，薬剤の設計などに用いられる。
- **金融と経済予測**：市場のトレンド分析やリスク評価，予測モデルの構築，意思決定のサポートなどに使われる。
- **医療診断と予測**：疾患の診断や予測，画像診断（MRIやCTスキャン），病気の予後予測，薬物副作用の予測などに応用される。

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



The screenshot displays the Booking.com search interface. At the top, there are navigation links for '宿' (Accommodation), 'フライト' (Flights), '航空券+ホテル' (Flights + Hotels), 'レンタカー' (Rent a Car), 'アクティビティ' (Activities), and '空港発着タクシー' (Airport Transfer). The main search area includes a destination input field, check-in and check-out dates, and the number of guests (currently set to 2 adults, 0 children, 1 room). A search button is located to the right of these fields.

Below the search area, there is a calendar for selecting dates. The calendar shows the months of May and June 2022. A tooltip is visible over the calendar, displaying the days of the week and the dates for both months.

The main content area features a section titled '直前予約が可能な近くのホテル' (Hotels available for last-minute booking nearby). This section lists several hotels with their names, addresses, and prices. Each hotel listing includes a small image, a rating, and a price per room per night.

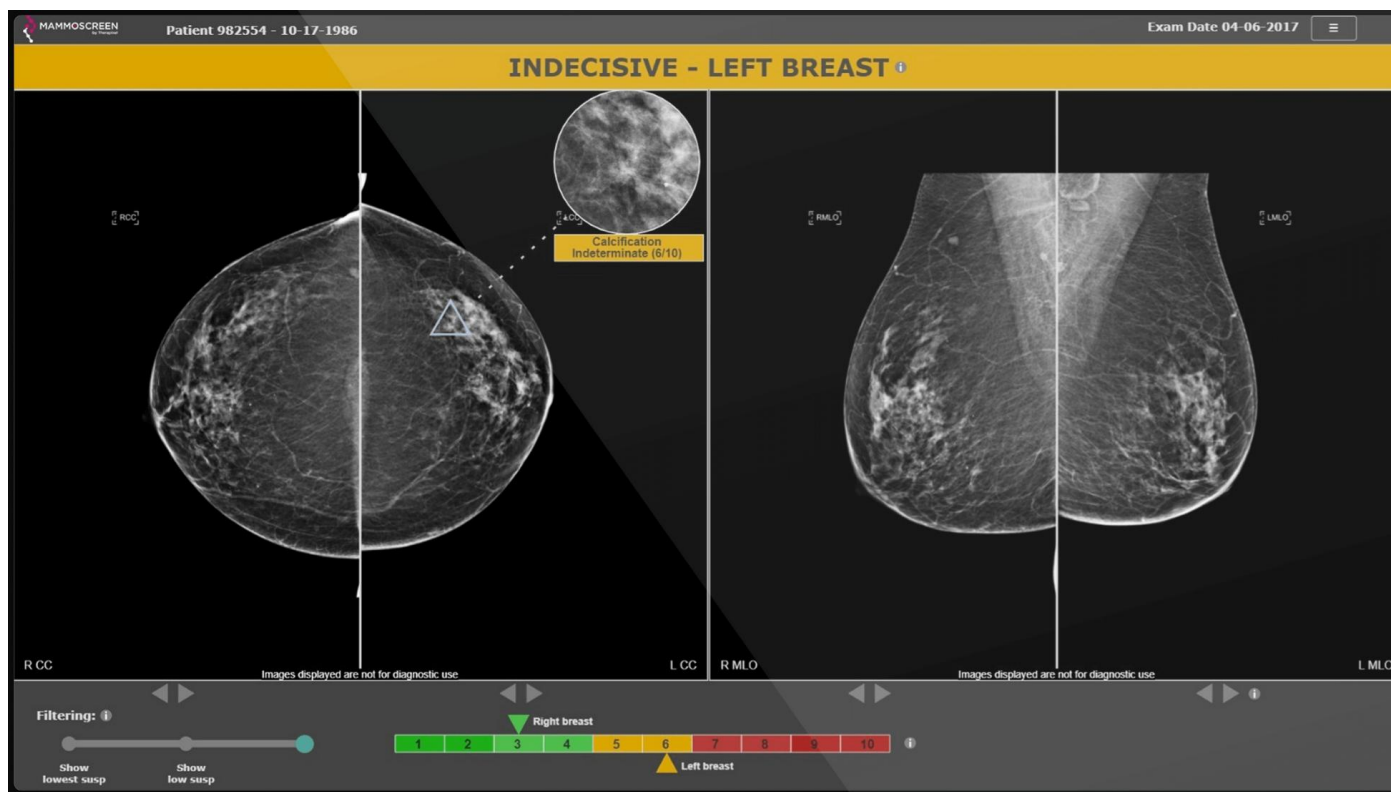
ホテル名	住所	料金 (本日の1泊分)	評価
ダイワロイネットホテル福山駅前	福山市, 三之丸町	¥9,600	8.6
リッチモンドホテル福山駅前	福山市, 東桜町1-1	¥8,000	8.2
Hotel Trend Fukuyama Ekimae	福山市	¥7,500	7.9
カンデオホテルズ福山	福山市, 御船町2-8-20	¥8,500	7.8
福山プラザホテル	福山市, 住吉町1-40	¥6,800	7.4

At the bottom of the page, there is a button that says '直前予約が可能なホテルをさらに表示' (Show more hotels available for last-minute booking).

医療での応用



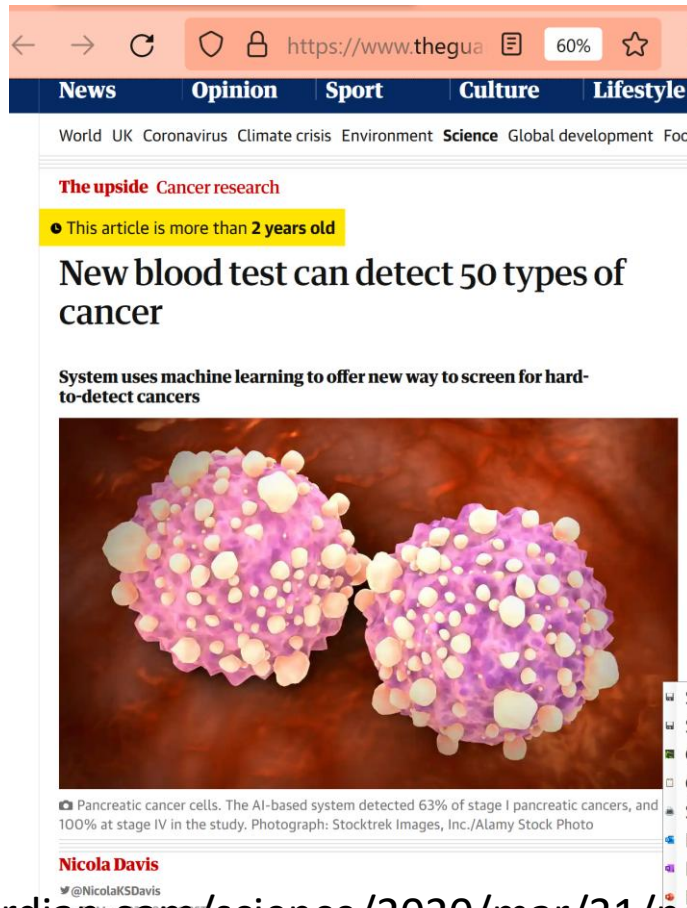
胸部のX線画像について、自分自身で、人工知能アプリを動かし、ヒントを得る



MAMMOSCREEN社のニュース

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

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



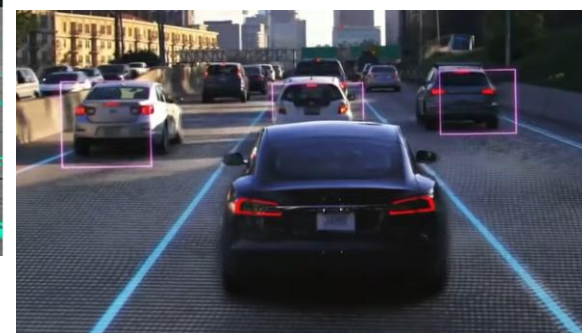
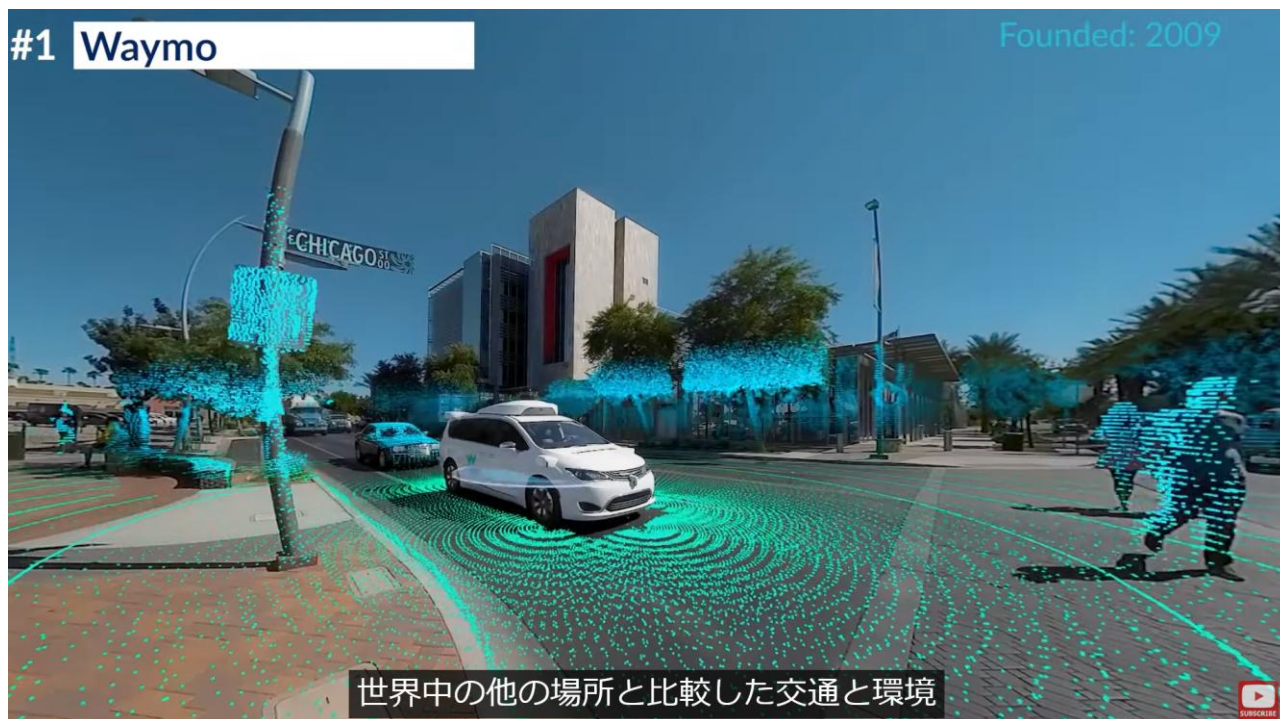
The screenshot shows a web browser displaying a news article from The Guardian. The browser's address bar shows the URL <https://www.theguardian.com/science/2020/mar/31/new-blood-test-can-detect-50-types-of-cancer>. The article is categorized under 'Science' and 'Cancer research'. A yellow banner indicates 'This article is more than 2 years old'. The main headline is 'New blood test can detect 50 types of cancer'. A sub-headline reads 'System uses machine learning to offer new way to screen for hard-to-detect cancers'. Below the text is a photograph of two clusters of pink, spherical pancreatic cancer cells. A caption below the image states: 'Pancreatic cancer cells. The AI-based system detected 63% of stage I pancreatic cancers, and 100% at stage IV in the study. Photograph: Stocktrek Images, Inc./Alamy Stock Photo'. The author's name, Nicola Davis, and their Twitter handle, @NicolaKSDavis, are also visible.

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

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



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

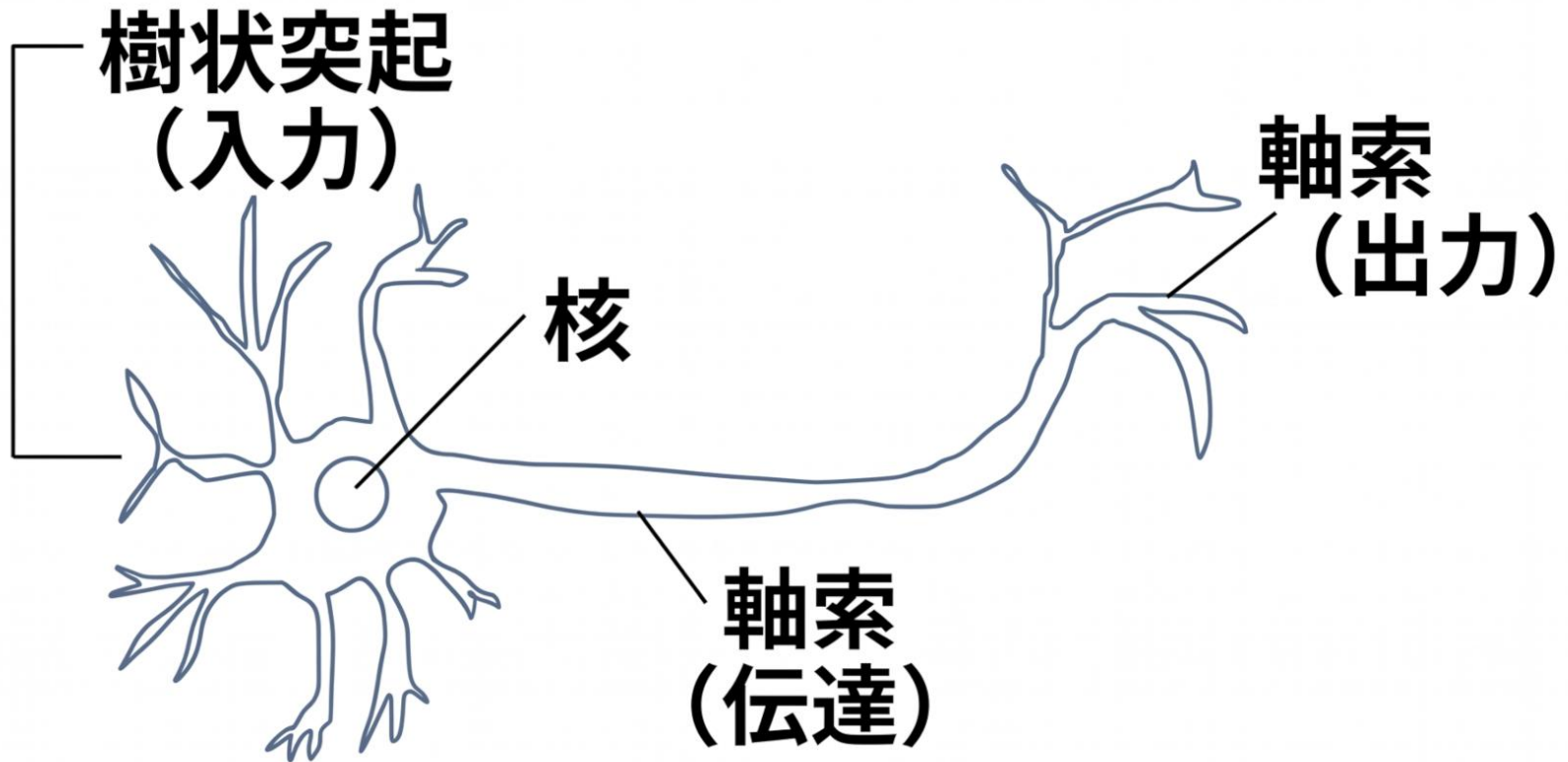
<https://www.youtube.com/watch?v=bdFi3RToOBk>

4-3. ニューロンと脳の構造

ニューロン



ニューロンは、脳や神経系における情報伝達、記憶、情報処理の基本要素。



生物種別のニューロン数（カタツムリからアフリカゾウまで）



脳のニューロンの数（推定値）

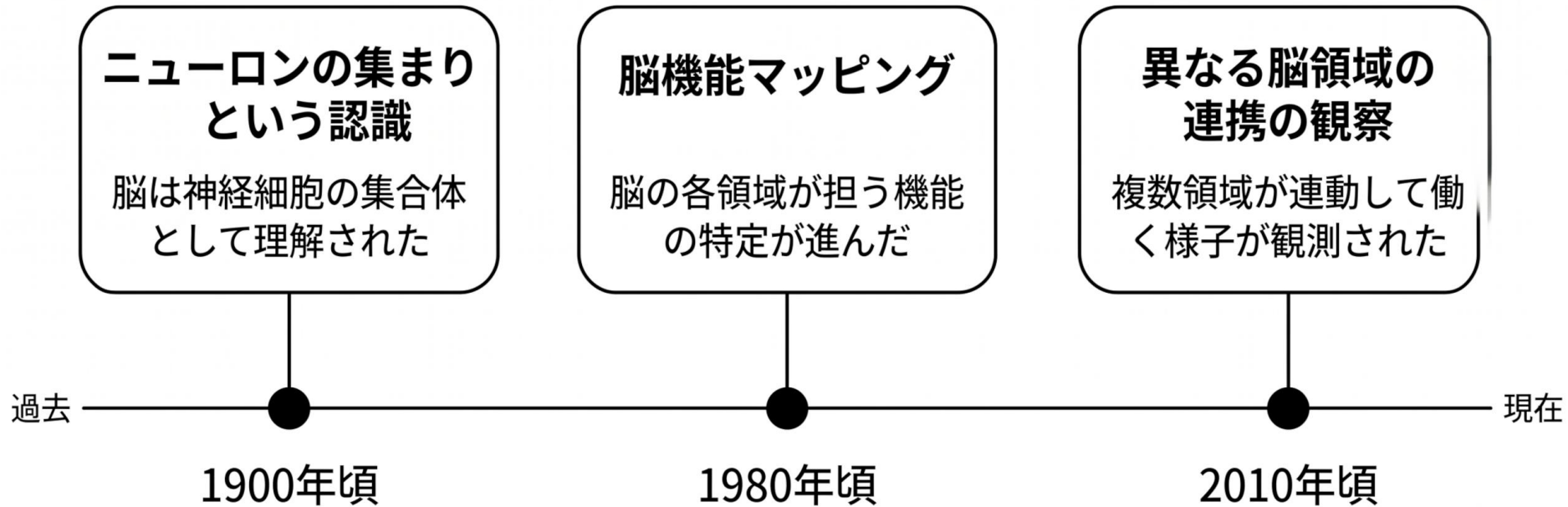
- カタツムリ 11,000
- ロブスター 100,000
- アリ 250,000
- カエル 16,000,000
- ハツカネズミ 71,000,000
- タコ 500,000,000
- ネコ 760,000,000
- ヒト 86,000,000,000
~ 100,000,000,000
- アフリカゾウ 257,000,000,000

- 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

Wikipedia 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

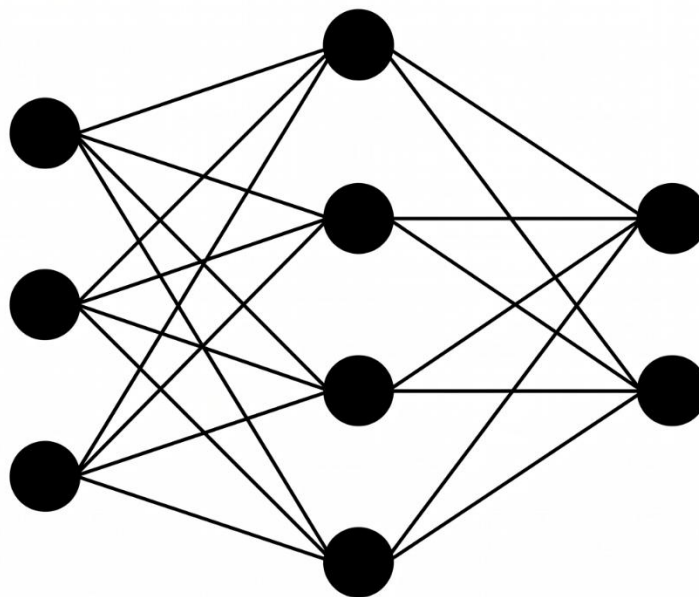
脳や神経系の研究の進展



生物の脳の働きの模倣を目指す人工知能の一種

コンピュータを使用

ソフトウェアとして
実装される



人工知能の一種

生物の脳の働きの
模倣を目指す

コンピュータ上で計算される

ニューラルネットワークの歴史



誕生

ニューラルネットワークの概念が提案された

1940年代

バックプロパゲーションの前史

制御理論や最適化の文脈で関連手法が提案された (Kelley 1960, Bryson 1961, Linnainmaa 1970)。1974年にWerbosが博士論文でニューラルネットワークへの応用を提案

1960~1970年代

バックプロパゲーションの再発見

Rumelhart, Hinton, Williamsが論文で再発見・普及させた

1986年

新技術の登場

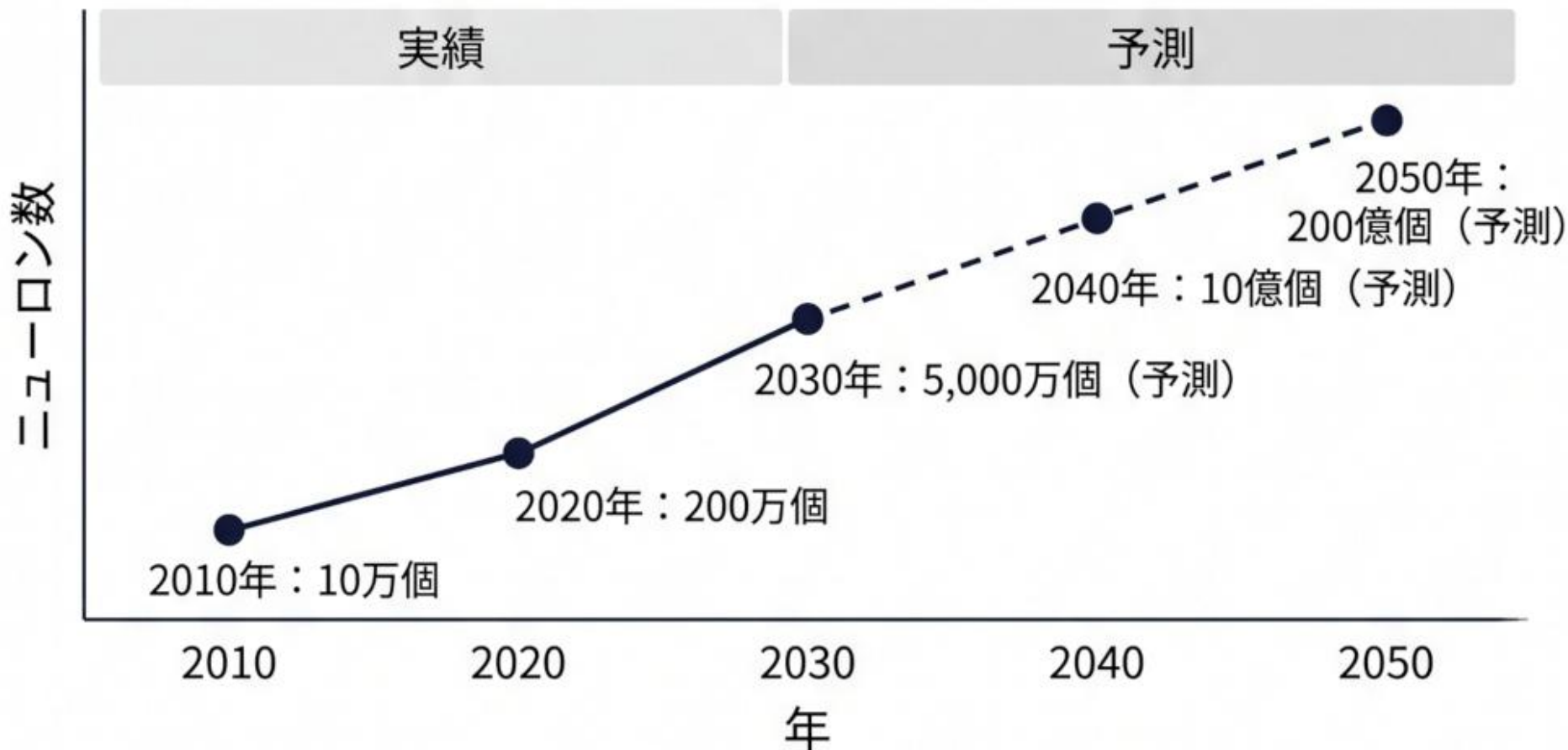
多層化, 正規化, ReLU, ドロップアウト

2010年代

ニューラルネットワークの発展



扱えるニューロン数の進展傾向

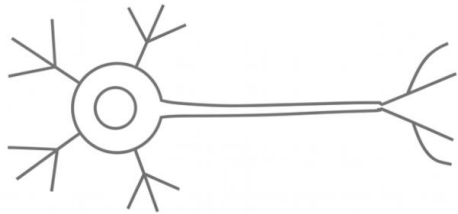


将来の数値は予測の一例であり，技術発展や計算資源の状況により変動する

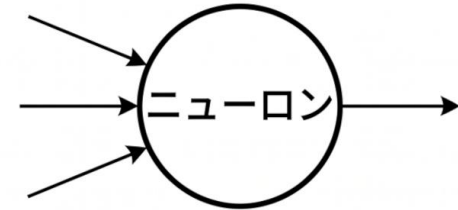
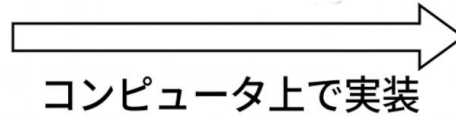
関連技術：GPU、クラウドコンピューティング

4-4. ニューラルネットワーク ワークの仕組み

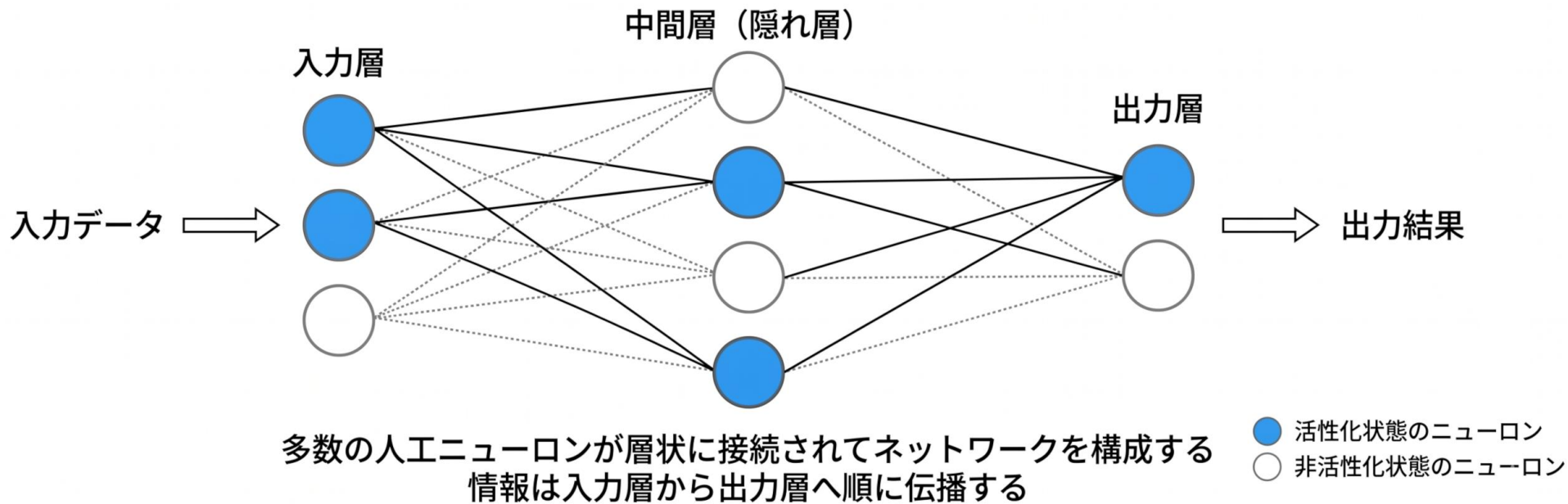
ニューラルネットワークの構造



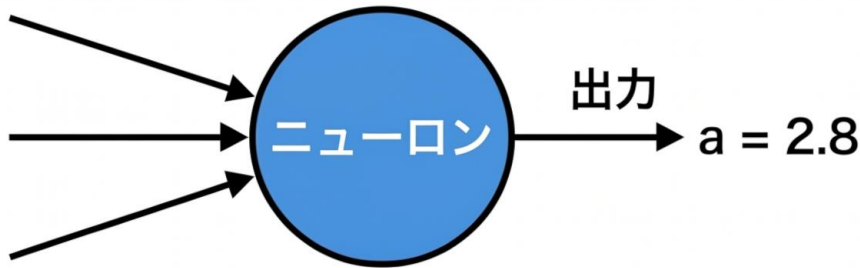
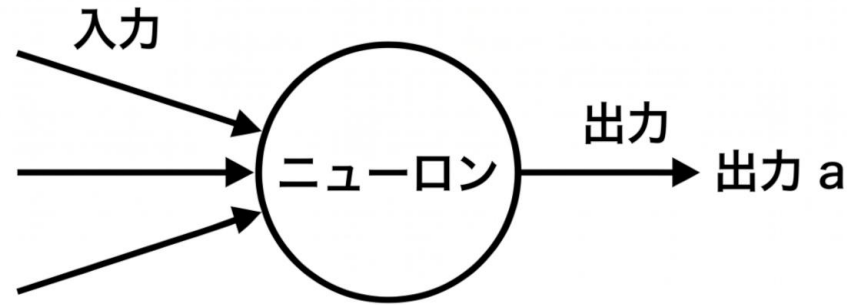
生物の神経細胞



人工ニューロン (数値計算ユニット)

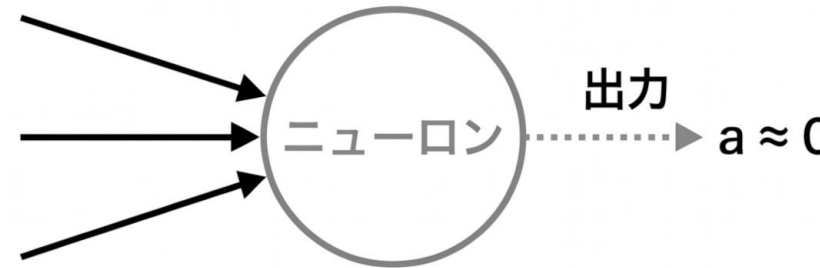


ニューロンの活性化と非活性化



活性化状態

出力がプラスの大きな値である
情報が次のニューロンへ伝わる



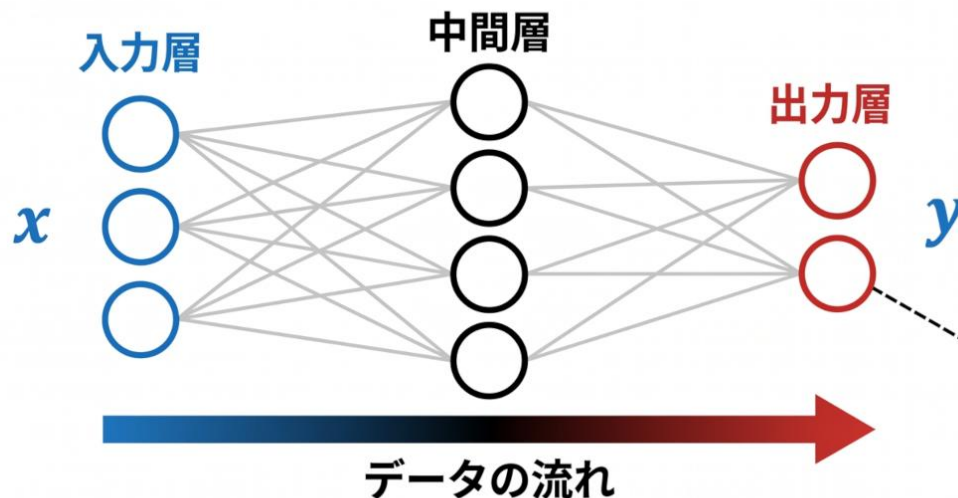
非活性化状態

出力が0に近い値である
情報が次のニューロンへ伝わらない

ニューロンの処理



ニューロンがネットワークを形成



ニューロンの3つの処理

青：入力値 / 赤：重み

① 入力重みづけ

x_1	w_1	\times
x_2	w_2	\times
x_3	w_3	\times

青：入力値 / 赤：重み

各入力値に対応する重みを掛ける

② 合計とバイアス

$$\sum (w \cdot x) + \underset{\text{バイアス}}{b}$$

合計を計算しバイアスを加える

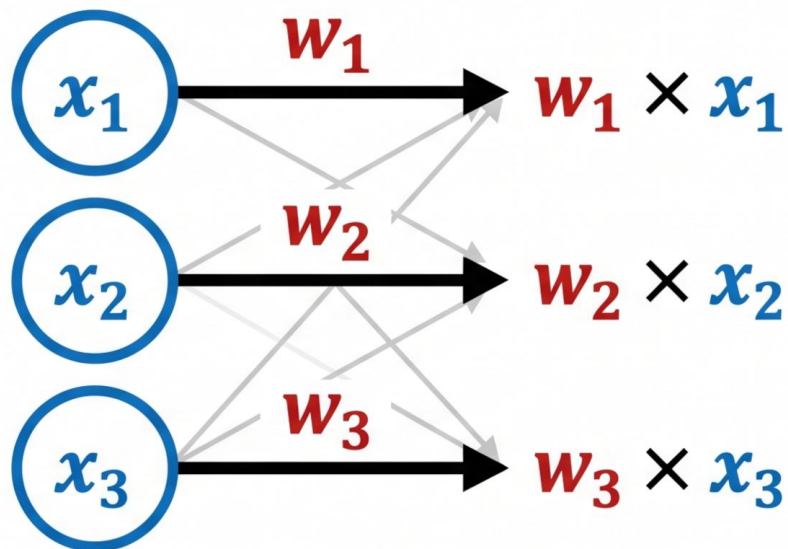
③ 活性化関数の適用

活性化関数

$$f(\quad) \quad y$$

ニューロンの活性度を算出

各入力値に対応する重みを掛ける



具体例

$$\begin{aligned}x_1 &= 0.5, x_2 = 0.8, x_3 = 0.2 \\w_1 &= 0.4, w_2 = 0.3, w_3 = 0.6 \\w_1 \times x_1 &= 0.4 \times 0.5 = 0.20 \\w_2 \times x_2 &= 0.3 \times 0.8 = 0.24 \\w_3 \times x_3 &= 0.6 \times 0.2 = 0.12\end{aligned}$$

入力 (x) : ニューロンに与えられるデータ

青=データ / 赤=パラメータ

重み (w) : 各入力の重要度を示すパラメータ。学習により最適化される

→次の処理 : ②合計とバイアスへ

合計を計算しバイアスを加える

①からの値

$$\begin{aligned} & w_1 \times x_1 \\ & w_2 \times x_2 \\ & w_3 \times x_3 \end{aligned} \quad \left. \begin{array}{l} \text{総和：すべての（重み} \times \text{入力）を足し合わせる} \\ \sum (w \times x) = w_1 \times x_1 + w_2 \times x_2 + w_3 \times x_3 \\ + \\ b \\ u = \sum (w \times x) + b \\ \text{バイアス：発火しやすさを調整する定数} \end{array} \right\}$$

総和 (Σ) : 重みづけ入力の合計

バイアス (b) : 加算する調整用定数。学習により最適化

u : ニューロンへの入力総和 (次の処理へ渡す値)

具体例

① 0.20, 0.24, 0.12
 $b = 0.1$

合計 = 0.20
+ 0.24
+ 0.12
= 0.56

$u = 0.56 + 0.1$
= 0.66

$u = 0.66$

←①入力の重みづけから

→③活性化関数の適用へ

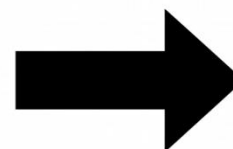
ニューロンの活性度を算出

②からの受け取り

$$u = \sum (w \times x) + b$$

活性化関数：入力総和を出力に変換する関数

$$f(u)$$



$$y$$

出力（活性度）

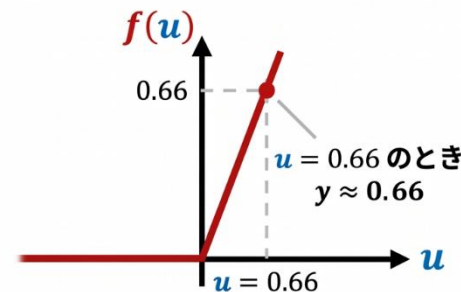
→ 次の層のニューロンへ

具体例（ReLU関数を使用）

②からの値： $u = 0.66$

$$y = f(0.66) = \max(0, 0.66) = 0.66$$

活性化関数の例：ReLU関数



活性化関数 (f)：入力総和を出力に変換する関数。

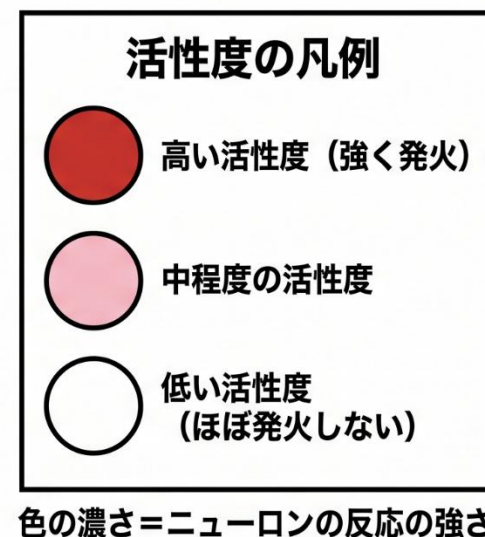
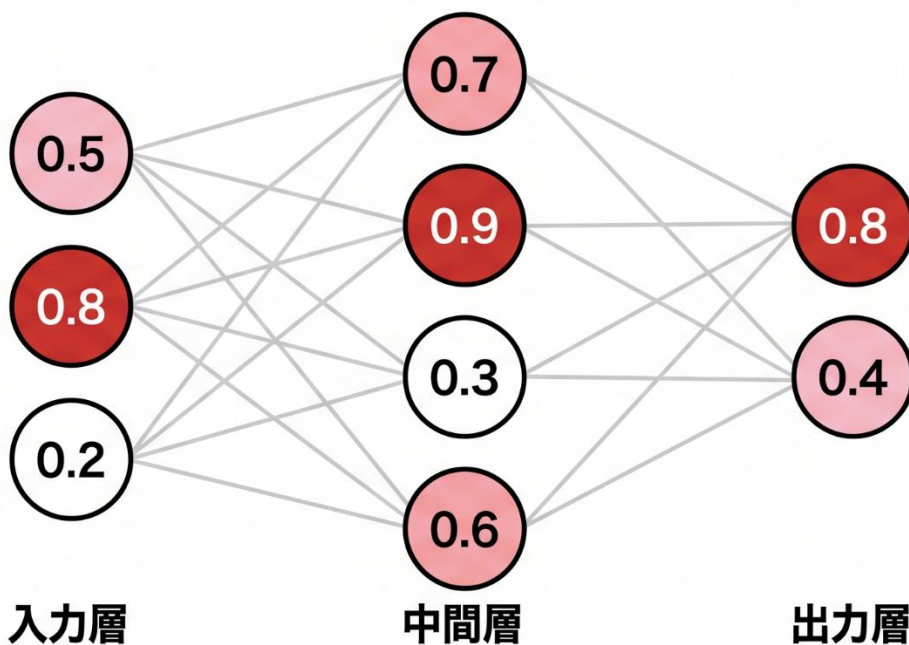
出力 (y)：次の層のニューロンへの入力となる 代表例：ReLU関数, シグモイド関数, tanh関数

②合計とバイアスから

ニューロンの活性度



各ニューロンがどれだけ強く反応しているか



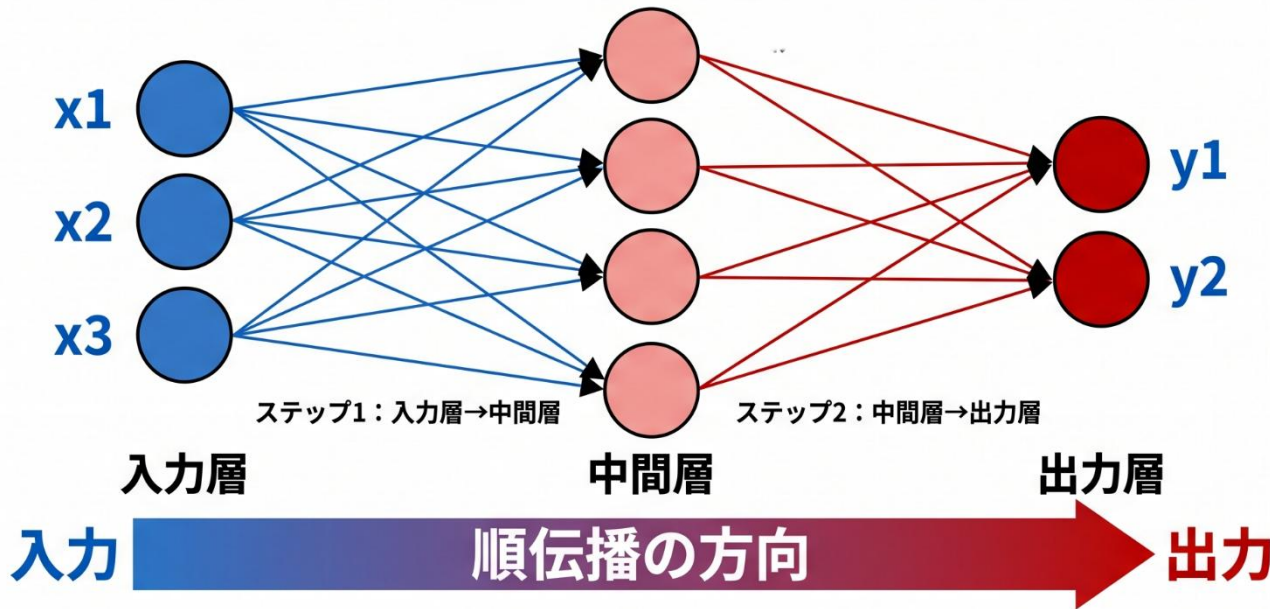
ニューロンの活性度とは

活性化関数 f を通して算出された各ニューロンの出力値である。値が大きいほどそのニューロンは強く反応している (発火している)

フォワードプロパゲーション (順伝播)



入力から出力へ情報が一方方向に伝わる仕組み



順伝播の手順

- ① 入力層に値 x を与える
- ② 各層で重みづけ・合計・活性化関数を適用
- ③ 出力層から結果 y を得る

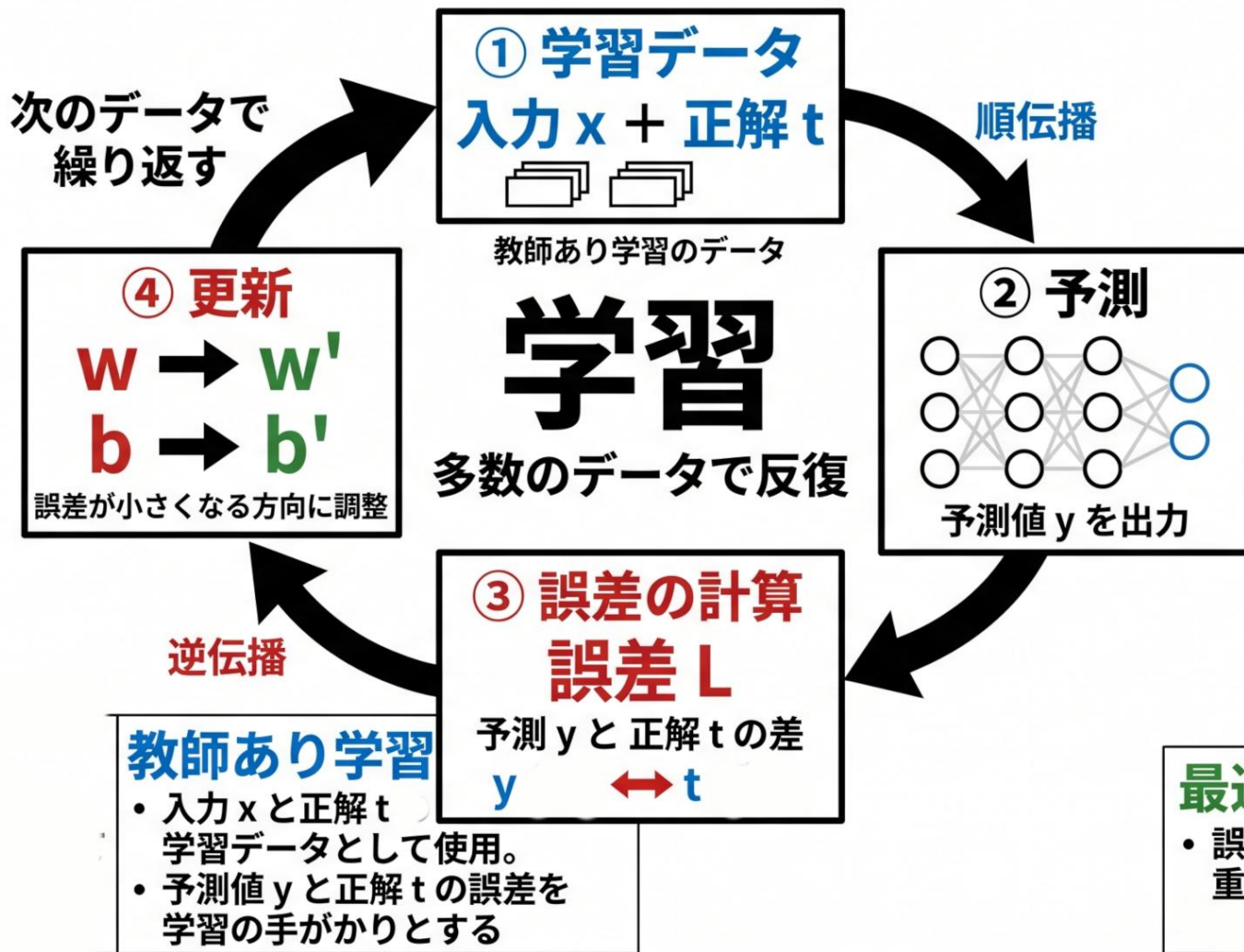
フォワードプロパゲーションとは
入力データがネットワークを
通って出力へ伝わる
一方方向の処理である

各層のニューロンは前の層からの値を受け取り、
活性化関数を通じて次の層へ値を渡す
この順方向の計算によって、ネットワークは入力から
最終的な出力 (予測値) を算出する

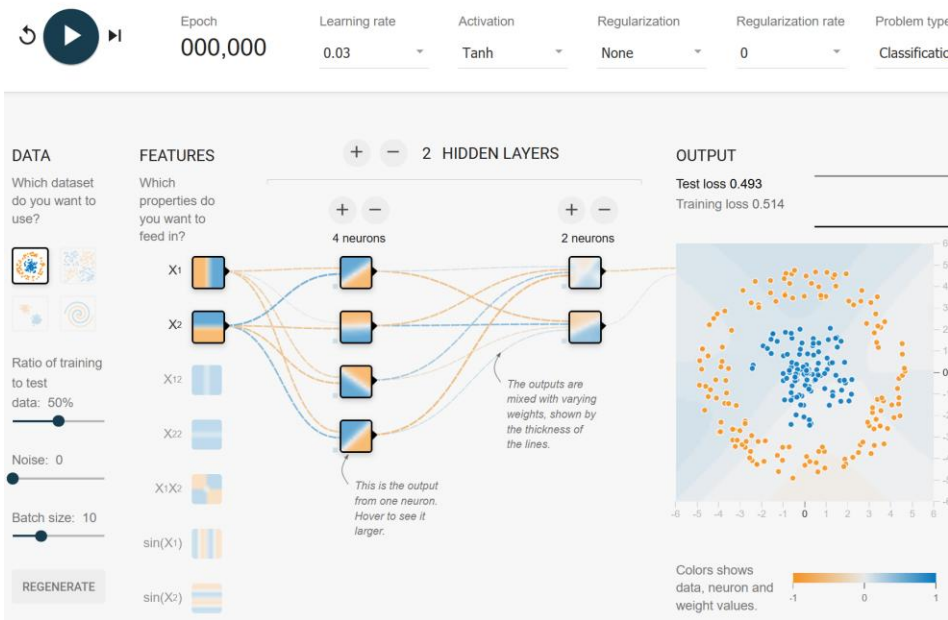
ニューラルネットワークの学習



多数のデータで繰り返し、重みとバイアスを最適化する



サイト TensorFlow Playground

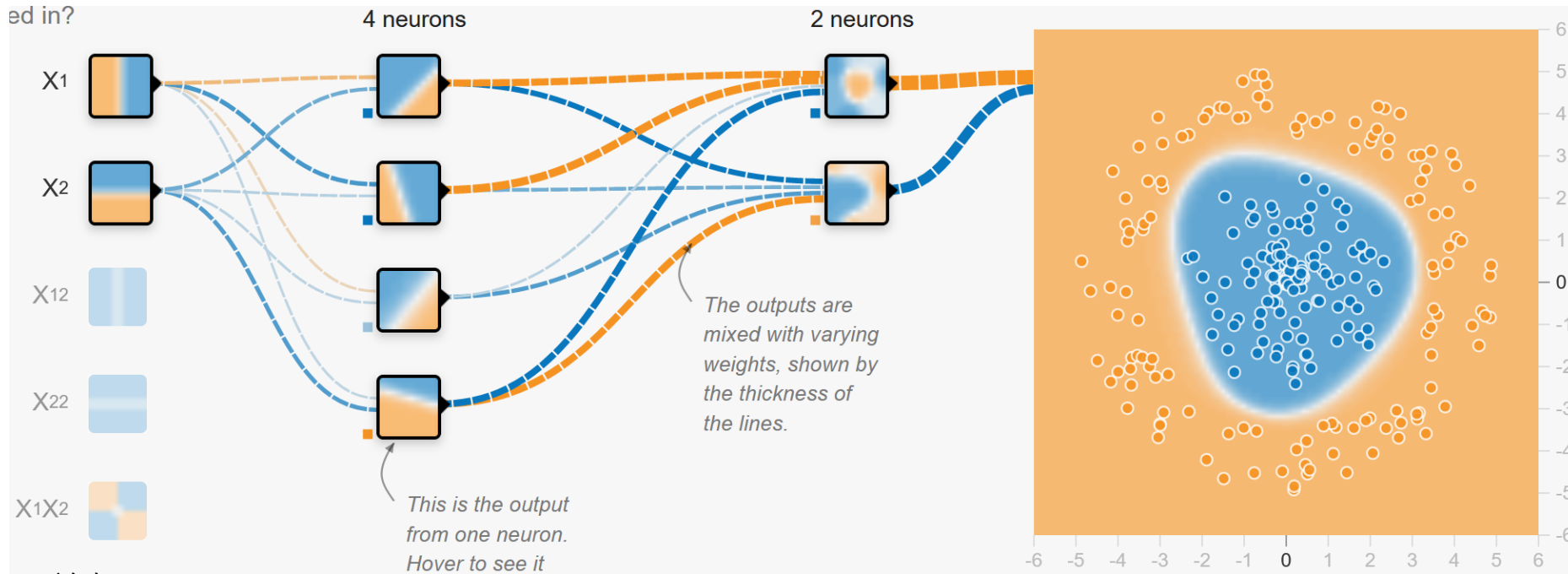


ニューラルネットワークの構造可視化

- DATA (データセットの選択)
 - FEATURES (入力特徴量)
 - HIDDEN LAYERS (隠れ層の数とニューロン数)
 - OUTPUT (分類結果)
- 学習パラメータ
- Learning rate (学習率)
 - Activation (活性化関数)
 - Epoch (学習回数)
- 分類の様子を観察できる。

<https://playground.tensorflow.org>

サイト TensorFlow Playground



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

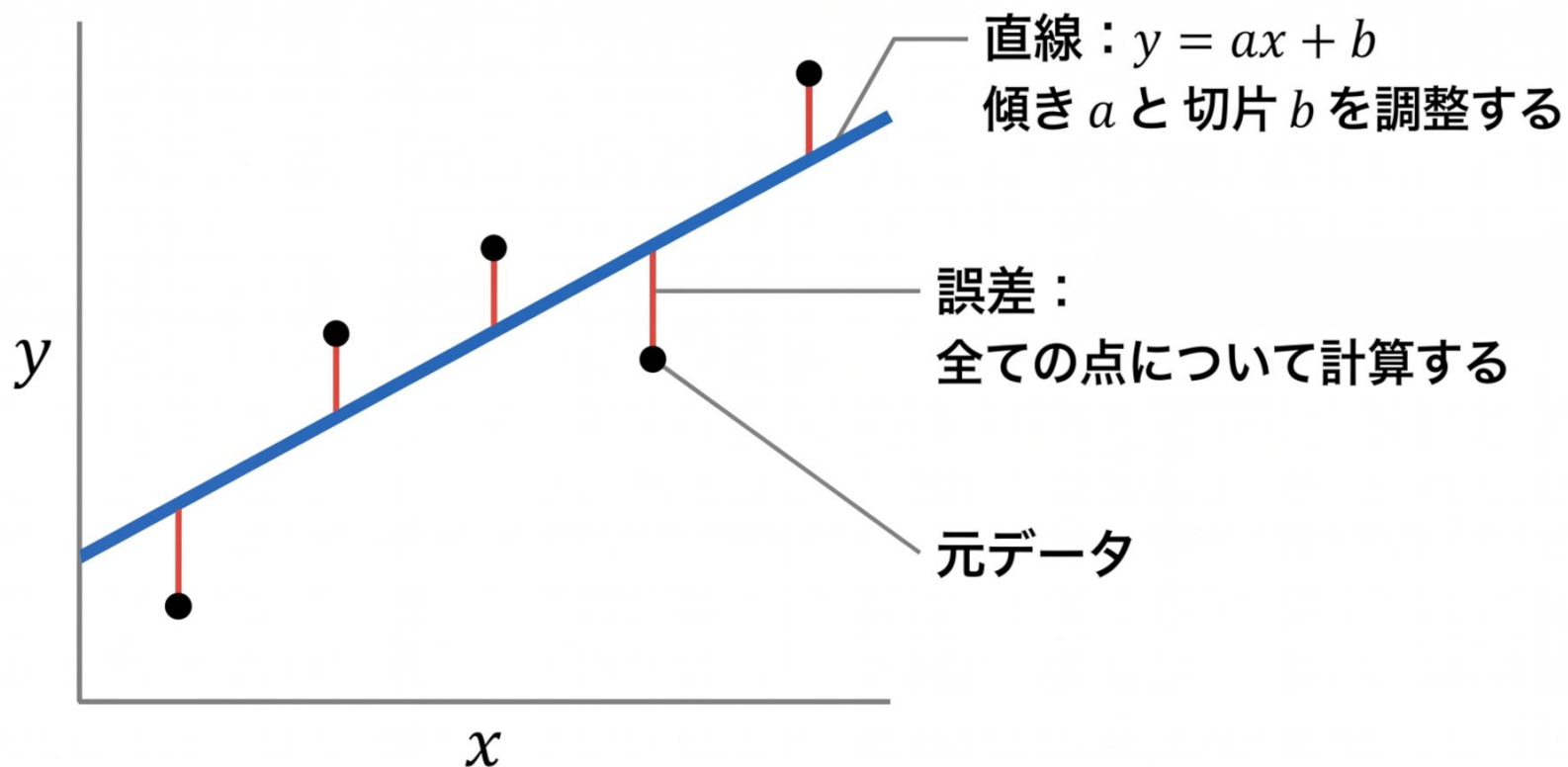
URL: <https://playground.tensorflow.org/>

4-5. 最適化とニューラル ネットワークの学習

直線フィッティング



点群を最もよく通る直線を求める問題

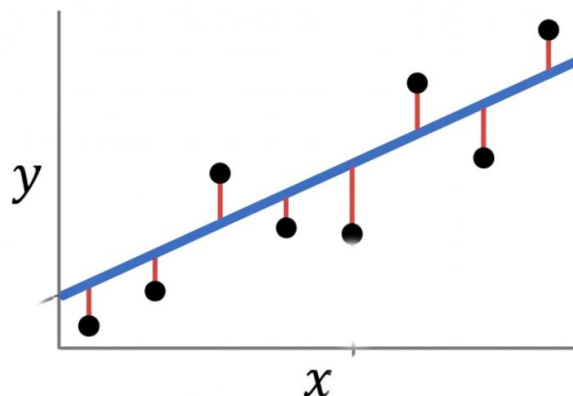


a と b を調整して、誤差の合計が最小になる直線を求める

最適化の3つの主要概念



最適化とは、パラメータを調整して目的関数を最小化または最大化するプロセスである



直線フィッティングの例

①

パラメータ

傾き a と切片 b

②

目的関数

誤差の合計

③

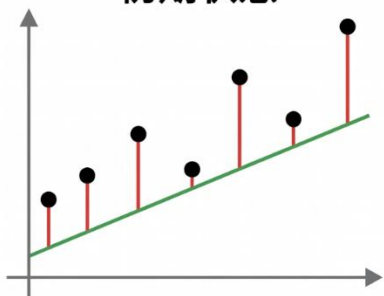
最適化

最適化の様子



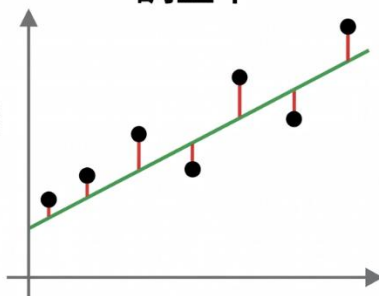
最適化とは、パラメータを調整して目的関数（ゴール）を最小化または最大化するプロセスである

初期状態



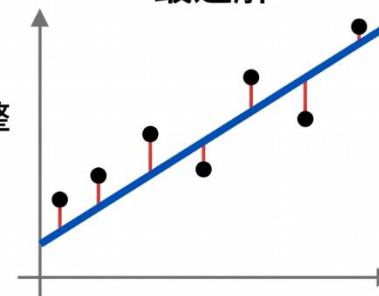
パラメータを調整
→

調整中



パラメータを調整
→

最適解



直線が点から大きく外れている
誤差：大きい

直線を少しずつ調整する
誤差：中くらい

直線が点の近くを通る
誤差：最小

パラメータ

目的関数（ゴール）

最適化

a ：直線の傾き

全ての点と直線との

a と b の値を変えながら

b ：直線の上下位置（切片）

誤差の合計

最小になる組合せを探す

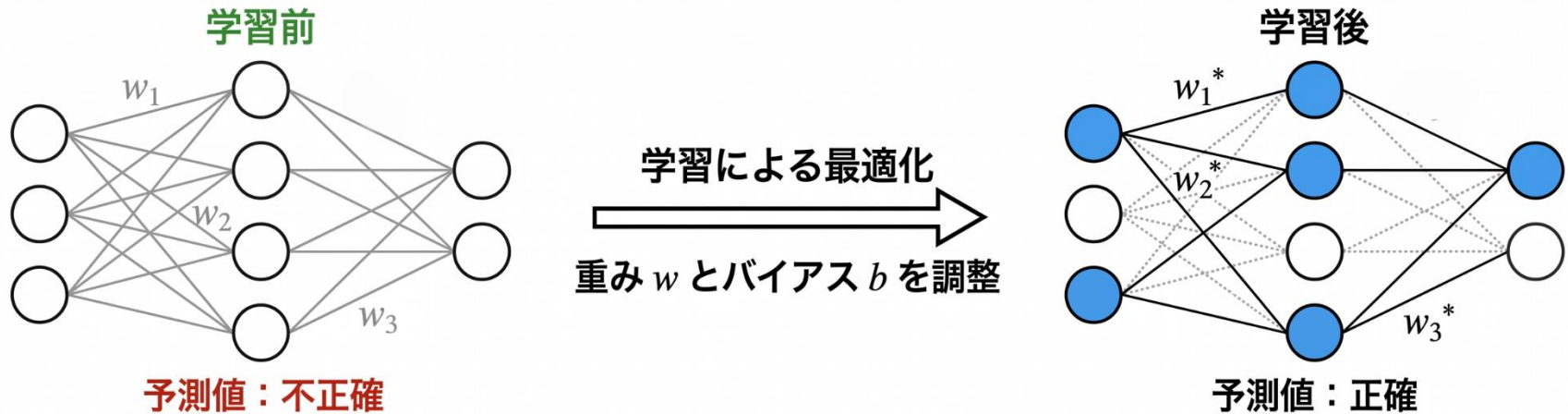
調整可能な数値

最小化したい量

ニューラルネットワークの学習における最適化



ニューラルネットワークの重みとバイアスを調整し、損失を最小化するプロセスである



パラメータ

重み w
バイアス b

目的関数 (損失)

予測値と正解との誤差

$L =$ 誤差の合計

最適化

w と b の値を調整し
 L が最小になる組合せを探す

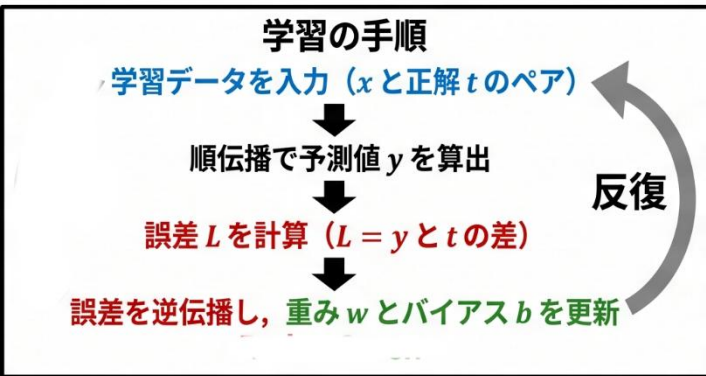
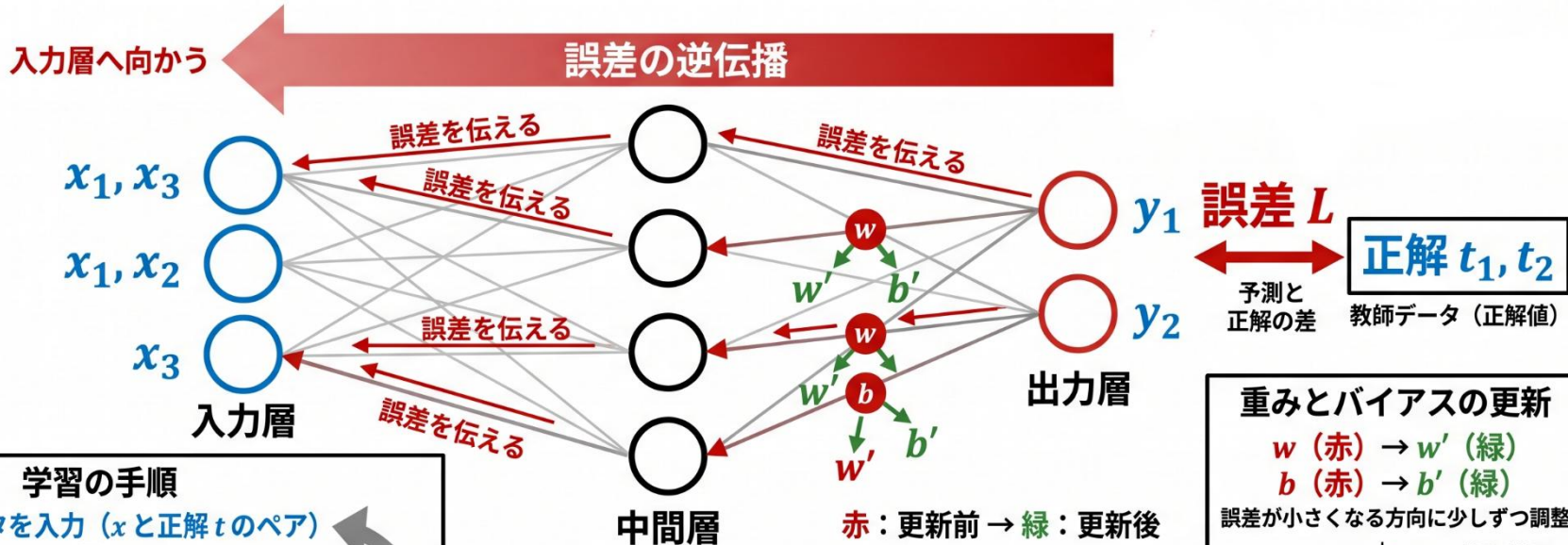
最適な w, b が求まる

最適化は、ニューラルネットワーク学習の中核をなすプロセスである

ニューラルネットワークの学習とバックプロパゲーション



誤差を逆方向に伝え、重みとバイアスを更新する学習の仕組み



バックプロパゲーションとは

学習データ (入力 x と正解 t のペア) を用いて、ネットワークの予測値 y と正解 t の誤差を計算
その誤差を出力層から入力層へ逆方向に伝播させる手法

多数のデータで繰り返すことで w と b が最適化される

