



HOKKAIDO
UNIVERSITY

講義「人工知能」

第3回 ニューラルネットワーク1

ニューラルネットワークの仕組み

北海道大学大学院情報科学研究院
情報理工学部門 複合情報工学分野
調和系工学研究室 准教授 山下倫央

<http://harmo-lab.jp>

tomohisa@ist.hokudai.ac.jp

2024年4月16日(火)

本日の内容

2

- 機械学習の分類
- ニューラルネットワークの構成
 - 種類
 - 単純パーセプトロン
 - 多項ロジットモデル
 - 多層ニューラルネットワーク
 - 学習の概要
- ニューラルネットワークを試す
 - A Neural Network Playground

学習とは

知識獲得入門（知識獲得と学習シリーズ1） [1987]

- ・ システムが前と同じタスクまたは同じ個体群から引き出されたタスクを二度目にはもっと効率よく、効果的に行えるようにする.

岩波情報科学辞典 [1990]

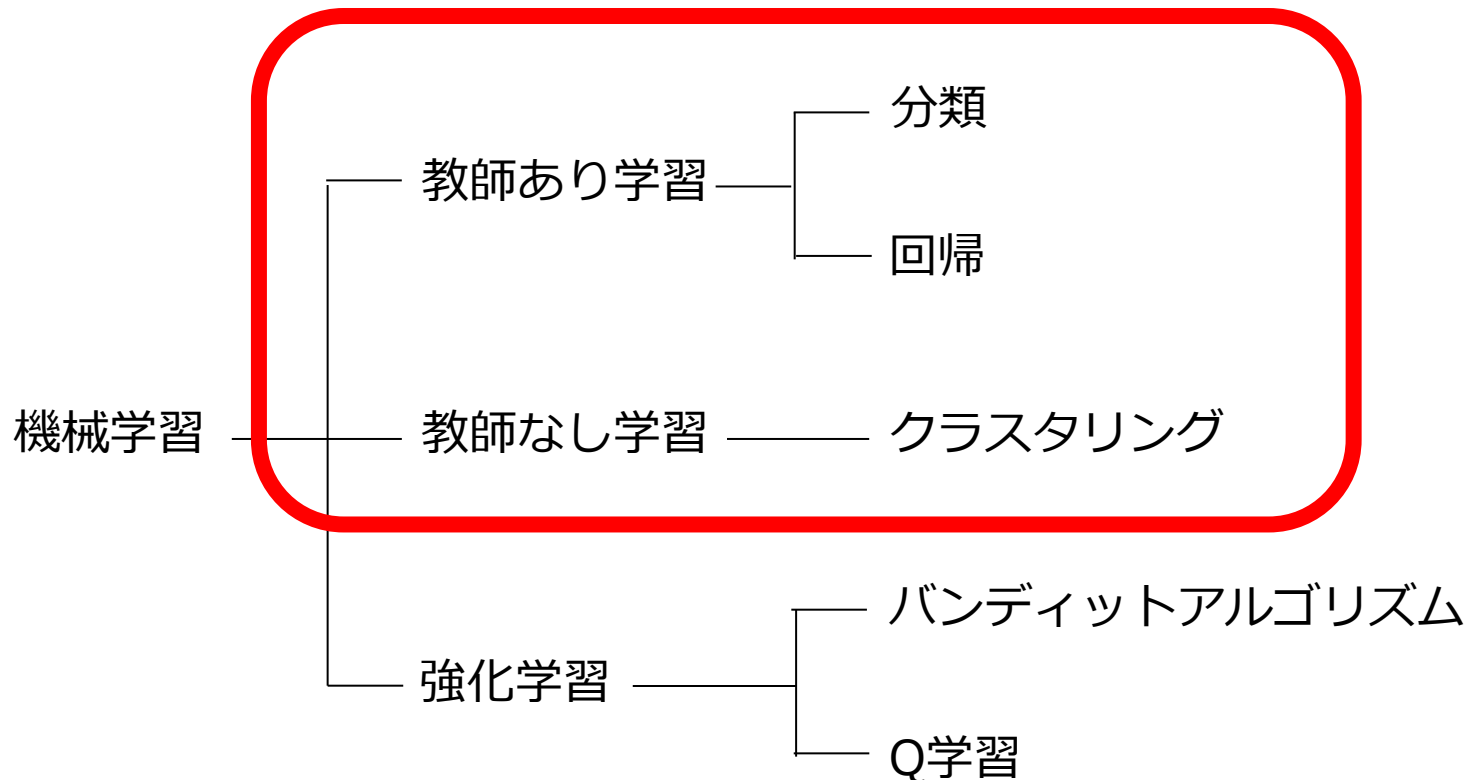
- ・ 経験により動作を改善すること.
- ・ ある程度の長期にわたる行動の、ある目的に向かう不可逆的な変化をいう

エージェントアプローチ 人工知能 [1997, 原著1995]

- ・ 自身の経験についての深い考察を通してその行動を改善することができる.
- ・ 知覚が現在の行為のためだけに使われるのではなく、未来におけるエージェントの行為能力を改善するためにも使われるべきである.

機械学習の分類

- 機械の持つパラメータと呼ばれる数値をより良い推定が行えるように決定・更新すること



機械学習の種類

5

- 教師あり学習

- データのラベルに基づく学習

- 画像データに「猫」「犬」など事前に意味付けを行うことをラベリングという
 - データが入力されたとき対応するラベルを正しく推定できるようにする

- 教師なし学習

- データの特徴を表す数値(特徴量)に基づく学習

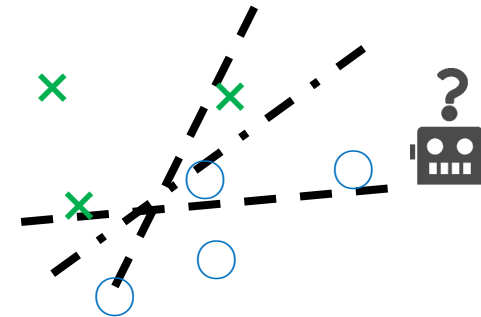
- 特徴を表す数値をもとに各データの近さを推定できるようにする
 - 例) 猫の画像の特徴
 - 色、目の間の距離、耳の長さなど

教師あり学習

- 入力データ（入力）を元に対応する出力を推定
 - 主に分類と回帰の2つ
- 推定した出力の誤差が小さくなるように学習

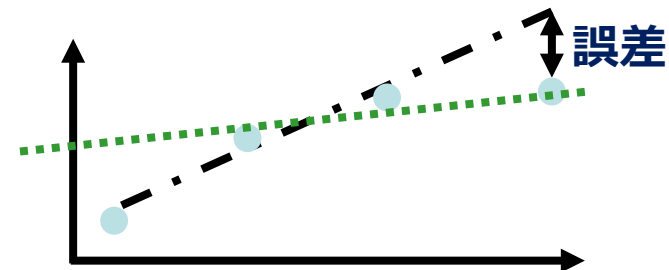
- 分類

- 上手く○と×を分ける
線（式）を見つける



- 回帰

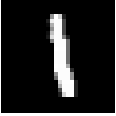


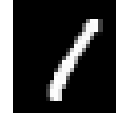


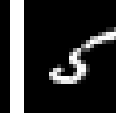

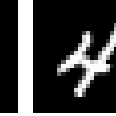
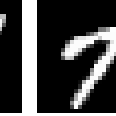
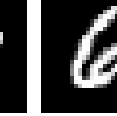
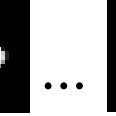
- 全ての点に最も近い線
（式）を見つける



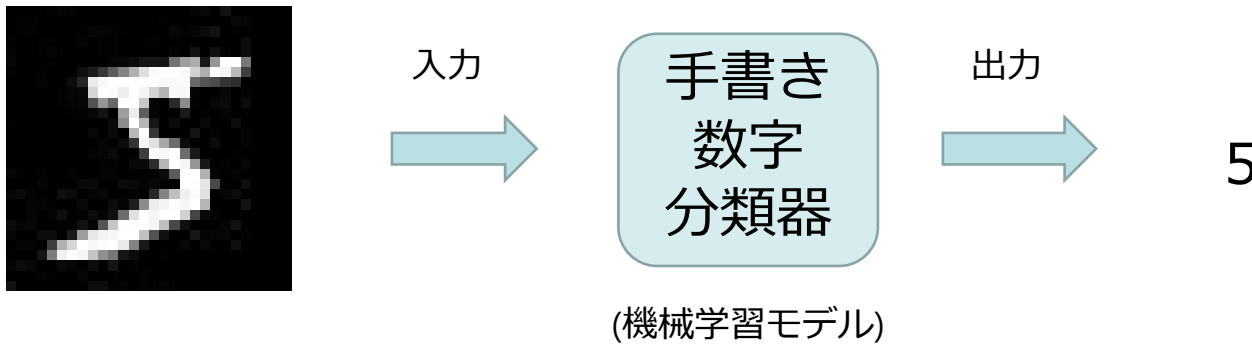
教師あり学習

❖ 例：手書き数字の画像を入力として書かれている数字が何か分類

■ 手書き文字画像

画像												...	
教師ラベル	1	2	9	1	4	0	5	3	4	7	6	...	5

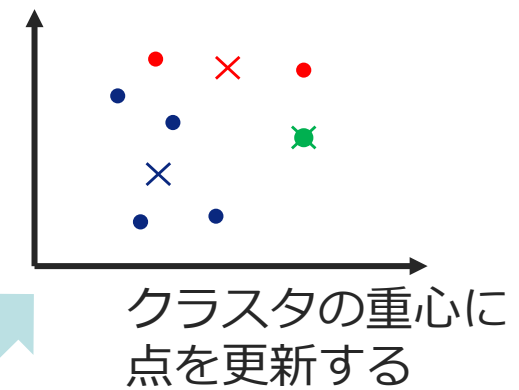
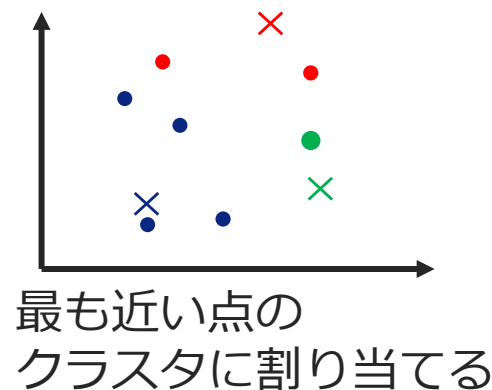
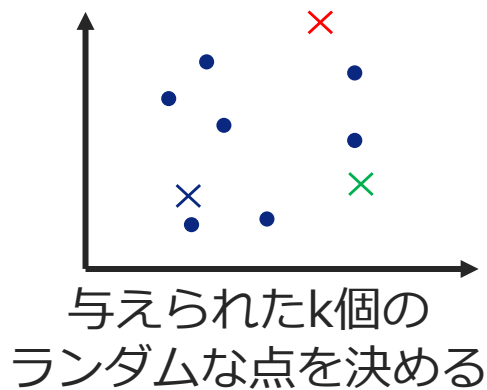
■ 分類器の入出力



教師なし学習の例 k-means法

8

- データを複数のグループに分けるクラスタリング手法
- 「似ているもの同士をグループに分ける」という手法
教師あり学習の分類と違って分けられたグループに意味があるとは限らない
- k-means法



割当が変わらなくなるまで繰り返す

k-means法

9

- デモ
 - テック煮ブログ
 - K-means 法を D3.js でビジュアライズしてみた
 - <http://tech.nitoyon.com/ja/blog/2013/11/07/k-means/>

強化学習

10

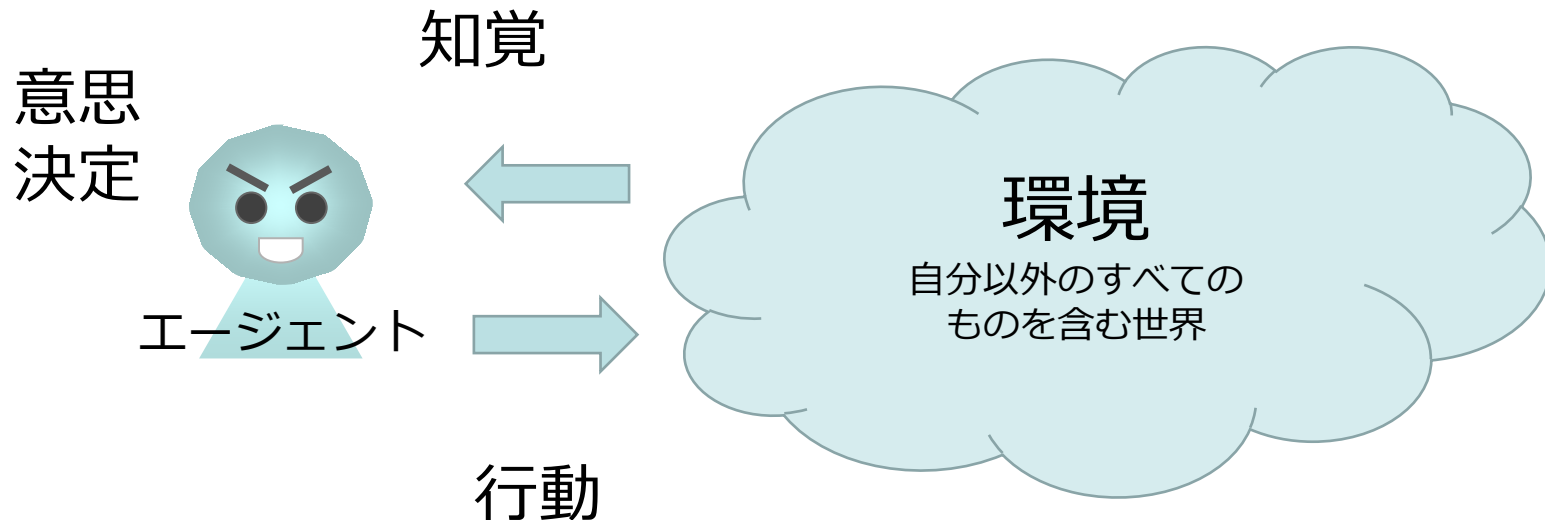
- 強化学習

- 計算主体であるエージェントが周囲の環境を観測することでどう行動すべきかを学習する。環境に影響を及ぼす行動を取った結果、報酬を獲得する
- この報酬に基づいて、エージェントは数多くのエピソードを経験して試行錯誤的に期待報酬を多くするように学習を進める

エージェントと環境

11

- エージェント(Agent)
 - 環境の状態を知覚し, 行動を行うことによって, 環境に対して影響を与えることが出来る自律的主体
- 環境(Environment)
 - エージェントの外部にあって, エージェントの意思によって直接変更できないものすべて



本日の内容

12

- 機械学習の分類
- ニューラルネットワークの構成
 - 種類
 - 単純パーセプトロン
 - 多項ロジットモデル
 - 多層ニューラルネットワーク
 - 学習の概要
- ニューラルネットワークを試す
 - A Neural Network Playground

第一次ブーム パーセプトロンの発見

13

1943年 生物の神経細胞の発火モデル（McCulloch-Pittsモデル）が提案

1949年 シナプス可塑性に関する仮説（ヘッブ則）提唱

- ニューロンAの発火がニューロンBを発火させると
2つのニューロン間の結合が強まる
- 現在のニューラルネットワーク学習の基本的な考え方

1958年 ローゼンブラットが単純パーセプトロンを考案

- 2クラス分類モデル
- 非線形問題が解けないことをミンスキーらが指摘
→ ニューラルブーム自体が下火に

ウォルター・ピッツ
(1923-1969)



数学者。
ニューラルネットワークの元祖とも言える「形式ニューロン」を考案した。

▶カリフォルニア州立大学より

マービン・ミンスキー
(1927-2016)



認知科学者。
MITのAI研究所やダートマス会議の創設者でAIという言葉を生み出した。

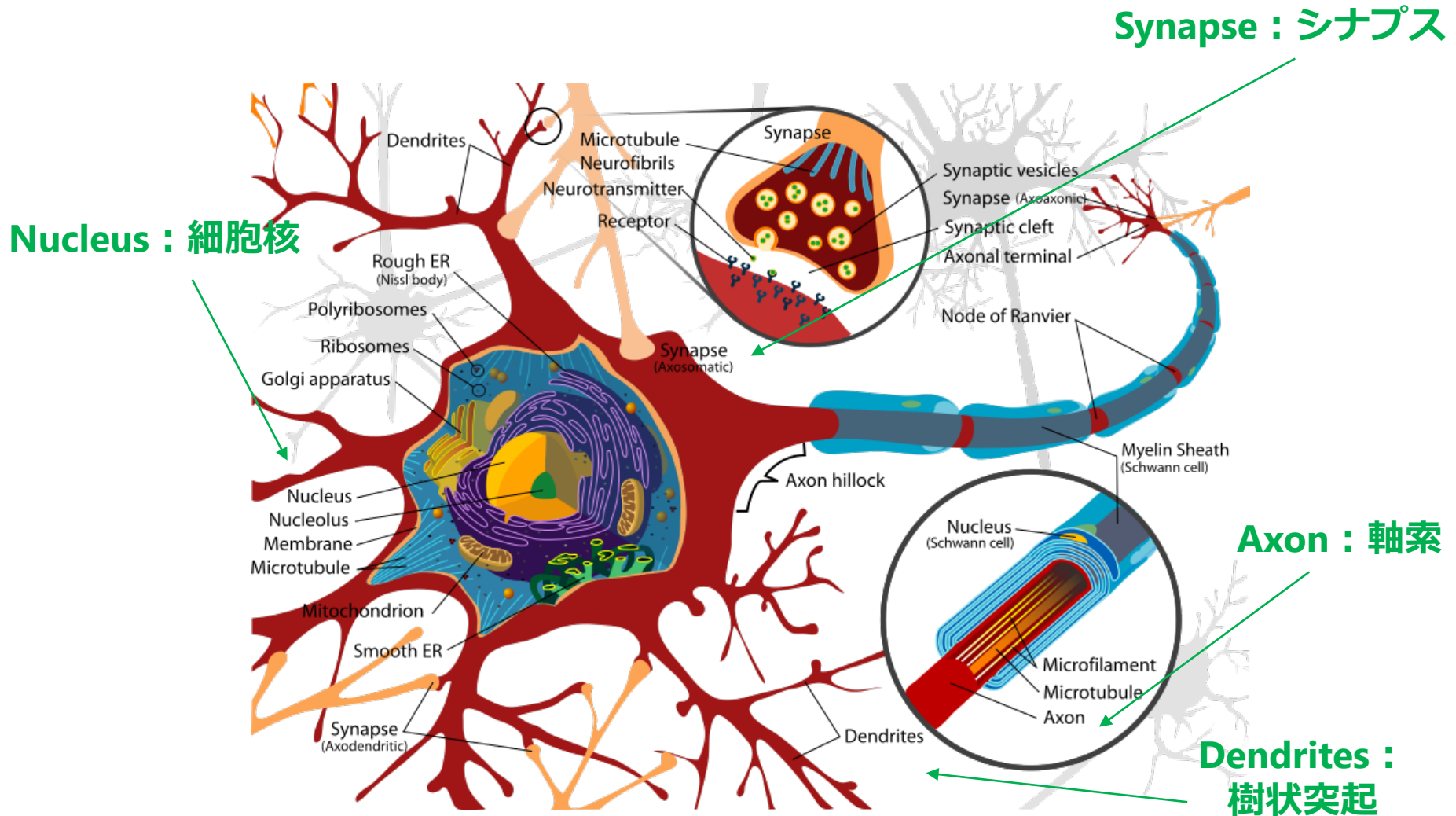
Wikipediaより

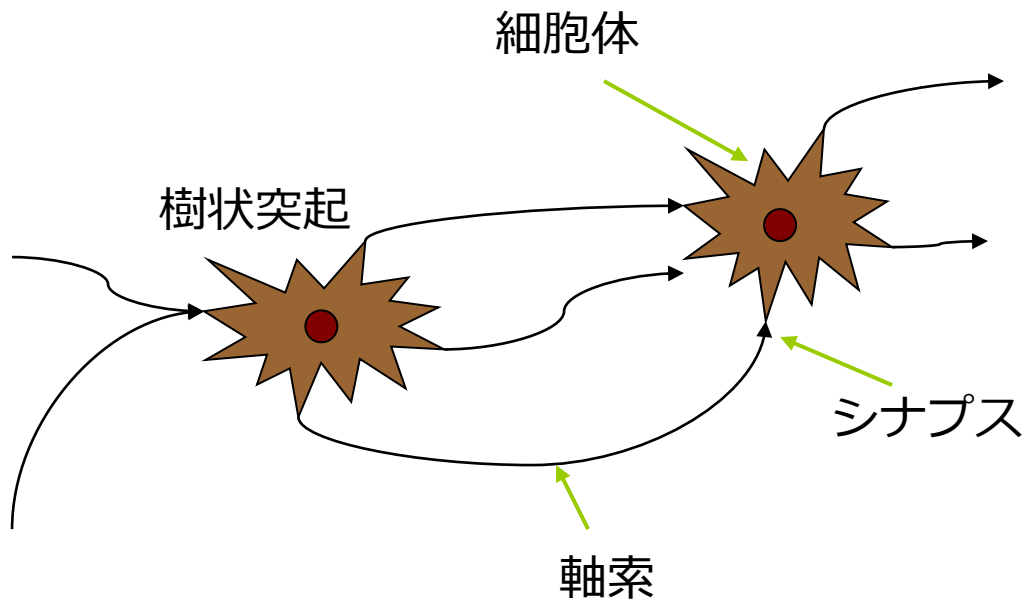
ジョン・マッカーシー
(1927-2011)



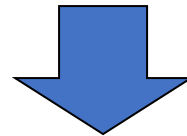
計算機科学者。
ミンスキーと並びAIの研究分野自体の創出に尽力したAI界の巨人。

Wikipediaより





- (1) 軸索からシナプスへ活動電位が伝送される
- (2) 細胞体で活動電位が加算
- (3) 加算された活動電位があるしきい値を越えたときのみ活動電位パルスが発生

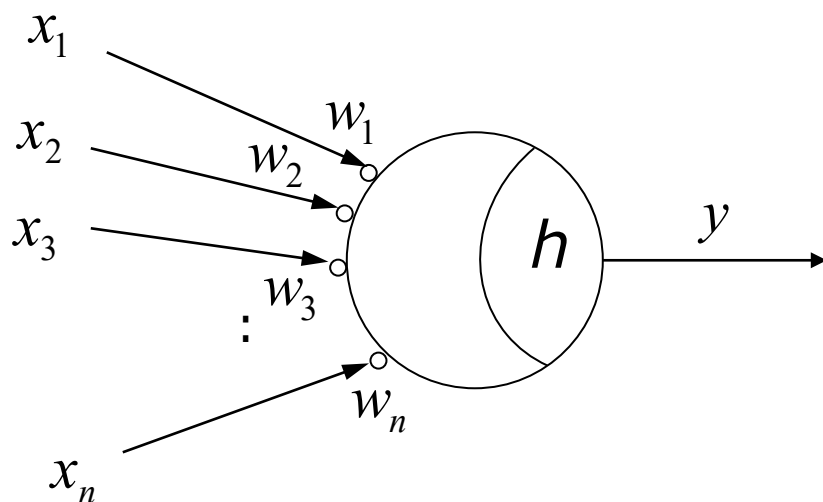


数理モデル化

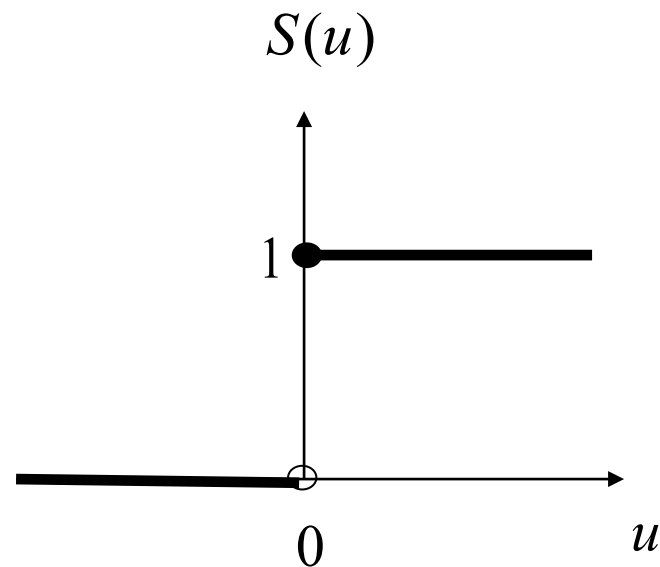
樹状突起 = 入力端末
軸索 = 出力端末

❖ [McCullock, Pitts 1943]

x_i : 入力信号 (0 または 1)
 w_i : 重み (実数)
 h : しきい値 (実数)
 y : 出力信号 (0 または 1)



$$y = S\left(\sum_{i=1}^n w_i x_i - h\right)$$

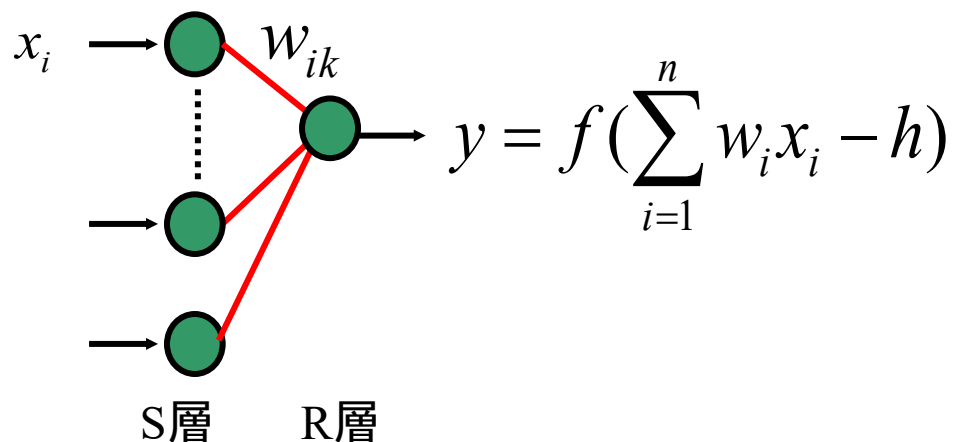


ヘビサイド関数 (ステップ関数, 階段関数)

$$S(u) = \begin{cases} 1 & u \geq 0 \\ 0 & u < 0 \end{cases}$$

$S(u)$: 活性化関数(伝達関数)

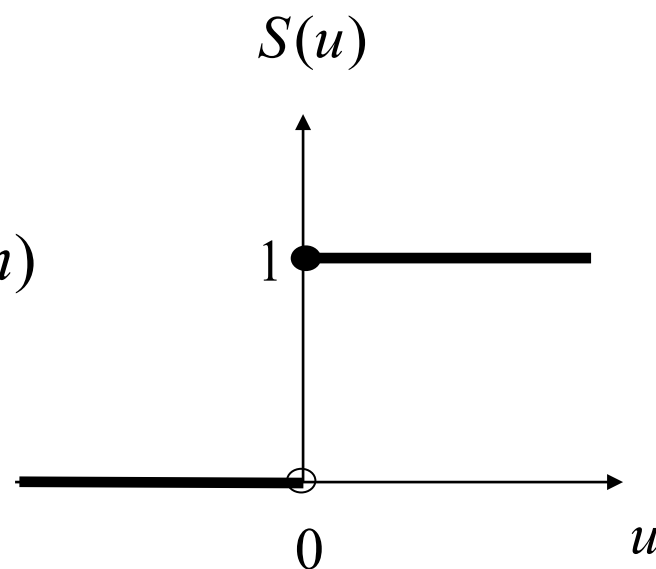
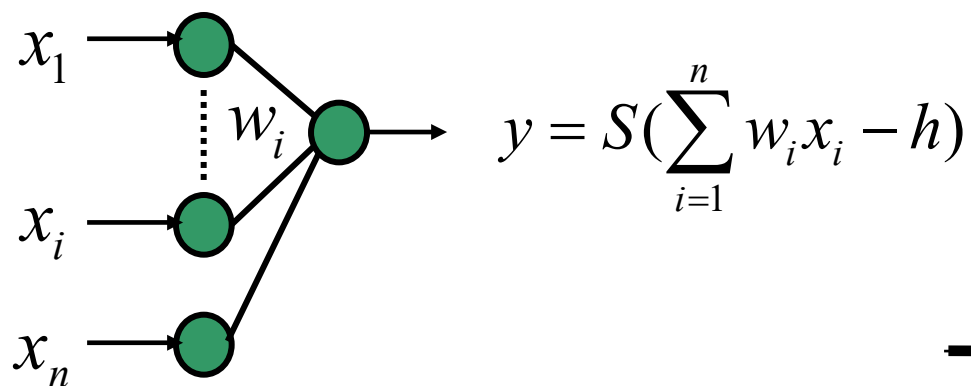
❖ [Rosenblatt 1958, 1961, 1962]



x_i : 入力信号 (0 または 1)
 w_i : 重み (実数)
 h : しきい値 (実数)
 y : 出力信号 (0 または 1)
 f : 伝達関数

S層...感覚層:sensory layer
R層...反応層: response layer

❖ 単純パーセプトロンの出力



任意の入力信号ベクトル (x_1, x_2, \dots, x_n) に対して, 適切な y を出力可能か?

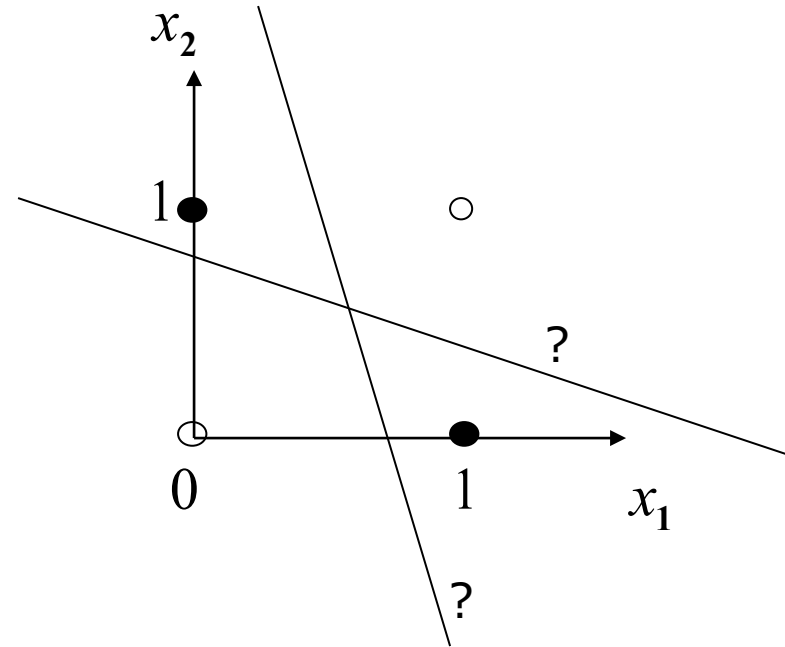


n 次元ユークリッド空間を, $n-1$ 次元の超平面 $\sum_{i=1}^n w_i x_i - h = 0$ で区切り, 一方の入力に対して 1, 反対側の入力に対して 0 を出力する.

❖ 線形分離不可能

排他的論理和 (XOR)

x_1	x_2	$x_1 \oplus x_2$
1	1	0
1	0	1
0	1	1
0	0	0

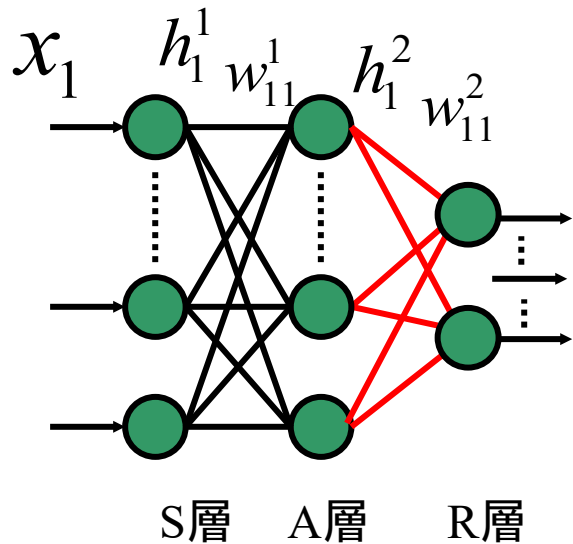


排他的論理和は、単純パーセプトロンで実現できない。

$$y = S(w_1 x_1 + w_2 x_2 - h)$$

どんなに w_1, w_2 を調節しても、正しい y を出力するようにできない。

❖ 単純パーセプトロンに中間層を追加



S層...感覚層:sensory layer
A層...連合層:associative layer
R層...反応層: response layer

$$y_j = f\left(\sum_{i=1}^n w_{ir}^p h_i^p\right)$$

x_i : i 番目の入力信号 (実数値)

h_q^p : p 層目の q 番目ニューロンの出力値

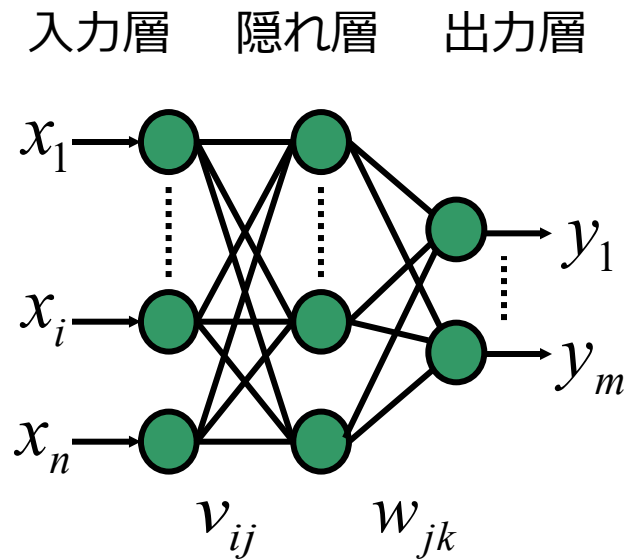
w_{qr}^p : p 層目の q 番目ニューロンで r 番目の
入力にかける重み (実数)

y_j : j 番目の出力信号 (0 または 1)

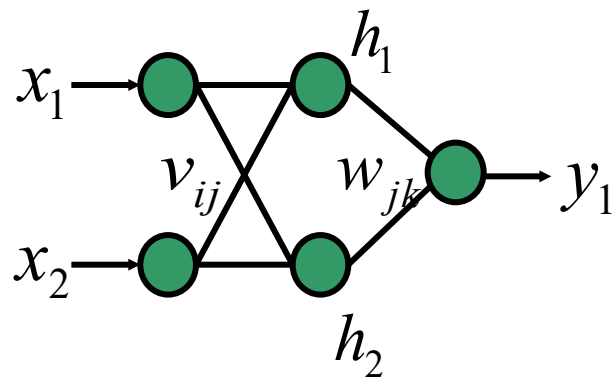
f : 活性化関数

❖ 任意の論理回路の表現

- 3層のネットワークで任意の論理回路の実現可能



NANDとORの組み合わせでXORを実現可能



隠れ層のニューロン数を変化

排他的論理和 (XOR)

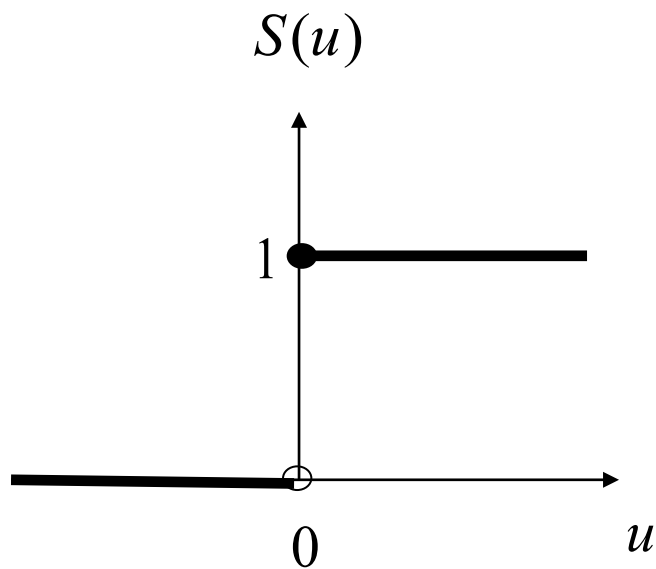
x_1	x_2	$x_1 \oplus x_2$
1	1	0
1	0	1
0	1	1
0	0	0

A blue arrow points from the label d_1 to the first row of the table, specifically to the output value 0.

学習回数と2乗誤差

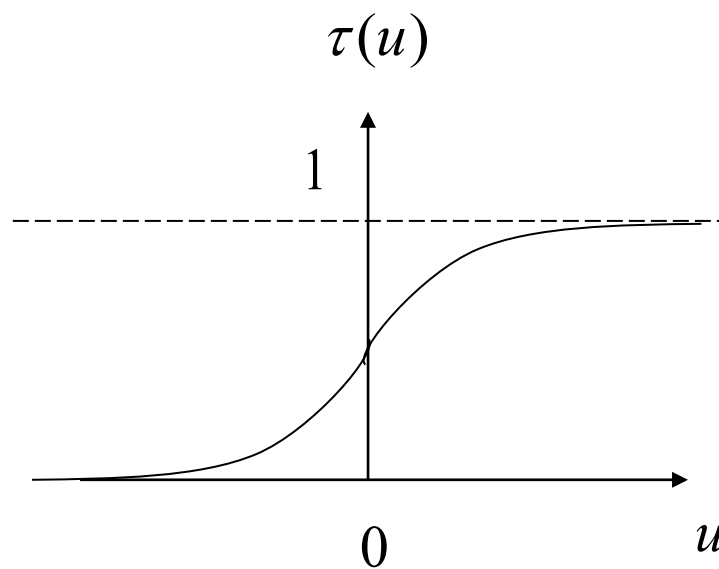
$$E = (d_1 - y_1)^2$$

❖ 微分可能性



ステップ関数

$u = 0$ で微分不可能



シグモイド関数

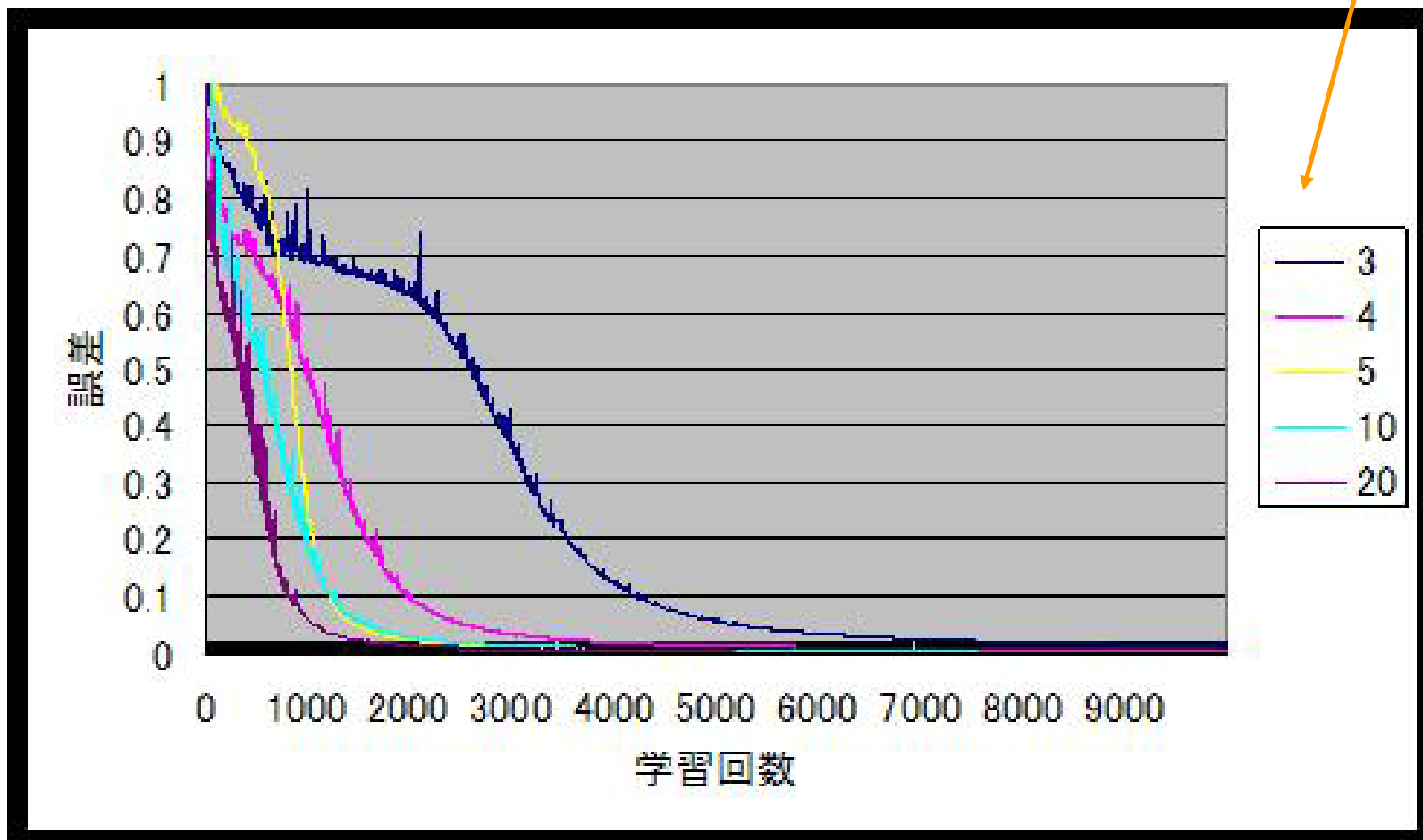
全区間で微分可能

$$\tau(u) = \frac{1}{1 + e^{-u}}$$

$$\tau'(u) = \tau(u)(1 - \tau(u))$$

❖ 誤差の推移

隠れ層のニューロン数



第二次ブーム 多層パーセプトロンへの期待と失望²⁵

1986年 パーセプトロンの多層化により非線形問題(XOR問題)が
解決可能であることが示される

- AND、NAND、ORゲートを組み合わせることで実現

1986年 多層パーセプトロンを効率良く学習できる

誤差逆伝播法(バックプロパゲーション)の考案

- 学習データの量、計算リソースの問題から当時は3層程度が限界
- この規模で学習できることが限定的
→ ニューラルネットワーク研究は再び冬の時代へ

福島邦彦
(1936-)



計算機科学者。
1979年に畳み込み型の多層パーセプトロン「ネオコグニトロン」を考案し、現代のディープラーニングに影響を与える。

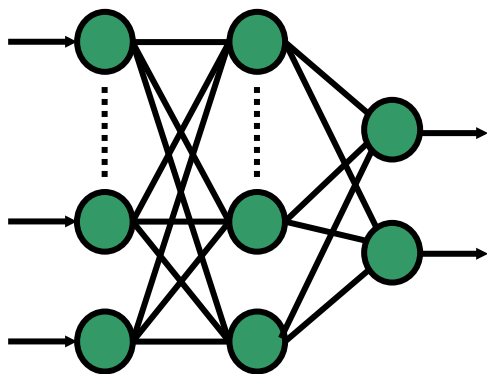
▶福島邦彦より

ジェフリー・ヒントン
(1947-)

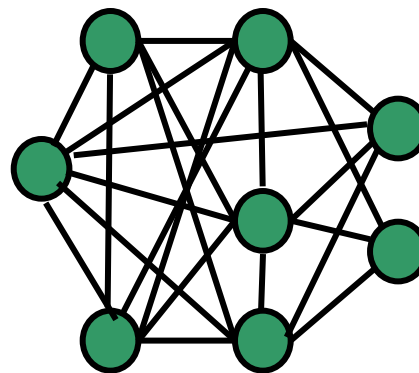


計算機科学者。
ニューラルネットワーク研究で重要な技術である「誤差逆伝播法」「ボルツマンマシン」「自己符号化器」「DBN」などの発明に関わってきたディープラーニング研究の第一人者。
現在はGoogleにも所属している。

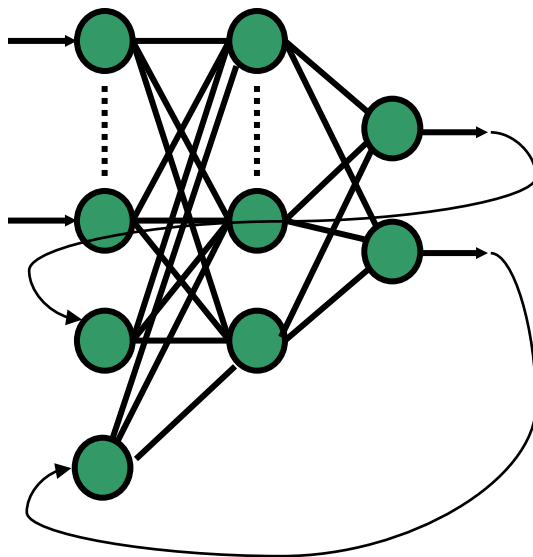
▶Wikipediaより



階層型



相互結合型



階層型リカレント

本日の内容

27

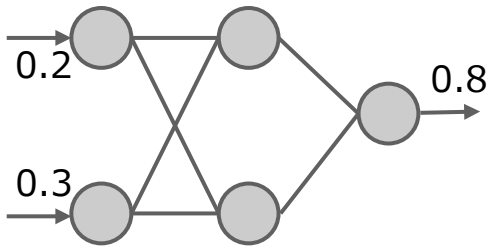
- 機械学習の分類
 - 教師あり学習
 - 教師なし学習
- ニューラルネットワークの構成
 - 種類
 - 計算例
 - 学習
 - 単純パーセプトロン
 - 多項ロジットモデル
 - 多層ニューラルネットワーク
- ニューラルネットワークを試す
 - A Neural Network Playground

ニューラルネットワークの学習の例

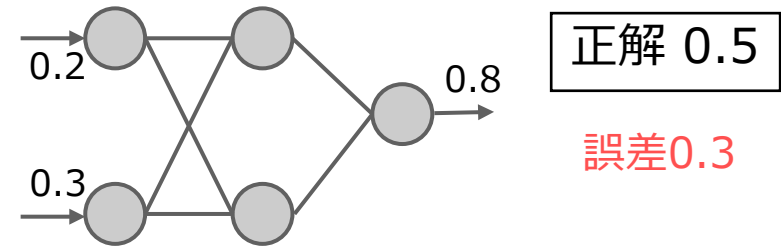
28

• 例) 足し算の学習

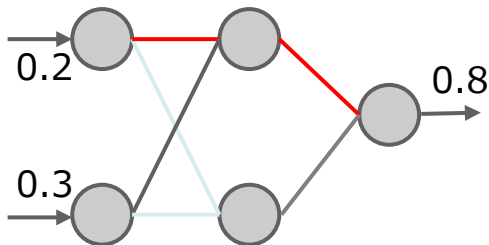
①入力データを元に出力を計算する



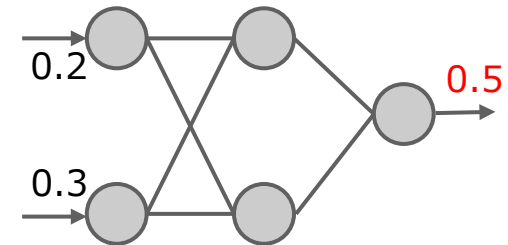
②正解との誤差を計算する



③誤差が小さくなるよう重みを順に更新



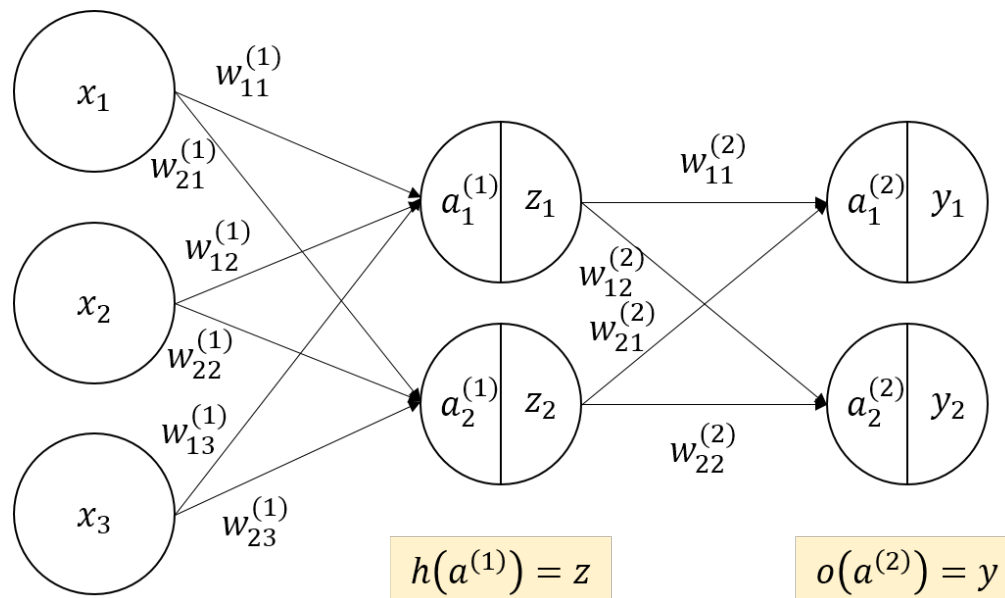
④重みが更新されなくなる（収束する）
まで①～③を繰り返す



ニューラルネットワークの計算例

29

- 入力層ユニット数3
- 隠れ層ユニット数2
- 出力層ユニット数2
- 隠れ層の活性化関数：シグモイド関数 $\frac{1}{1 + e^{-ax}}$
- 出力層の活性化関数は恒等関数 $o(a)=a$



ニューラルネットワークの計算例

30

- 重みの設定

$$W^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.2 & 0.2 & 0.2 \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

- 訓練データ

入力 $\mathbf{x} = (x_1, x_2, x_3) = (1, 2, 3)$

教師信号 $\mathbf{t} = (t_1, t_2) = (0, 1)$

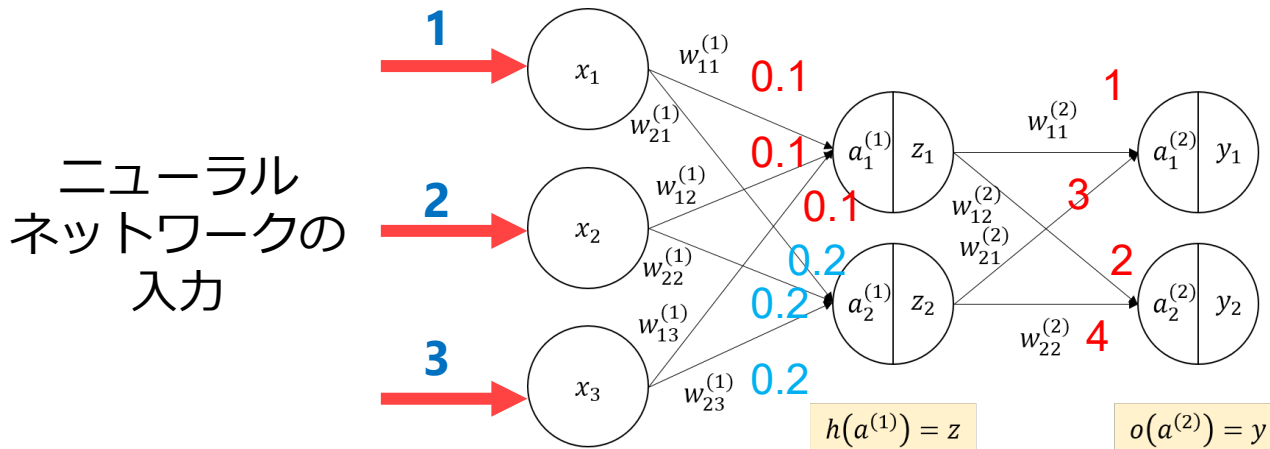
ニューラルネットワークの計算例：順方向

31

- 入力層ユニット数3
- 隠れ層ユニット数2
- 出力層ユニット数2
- 隠れ層の活性化関数：シグモイド関数 $\frac{1}{1 + e^{-ax}}$
- 出力層の活性化関数は恒等関数 $o(a)=a$

$$W^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.2 & 0.2 & 0.2 \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$



$$\mathbf{x} = (x_1, x_2, x_3) = (1, 2, 3)$$

$$\mathbf{t} = (t_1, t_2) = (0, 1)$$

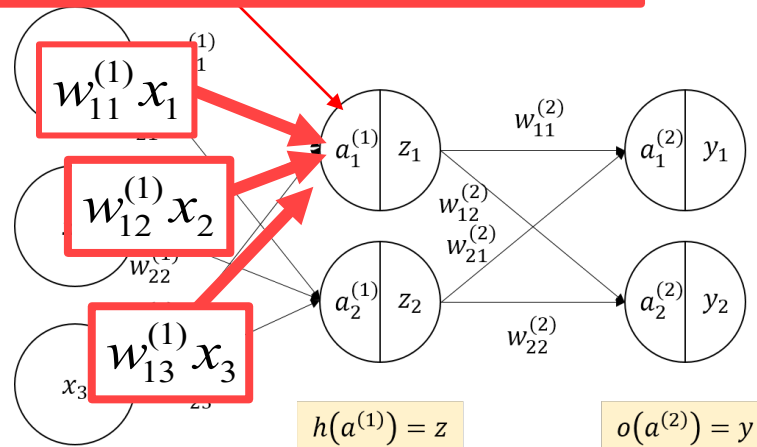
ニューラルネットワークの計算例：順方向

32

- 入力層ユニット数3
- 隠れ層ユニット数2
- 出力層ユニット数2
- 隠れ層の活性化関数
- 出力層の活性化関数

隠れ層のユニット1の入力

$$\begin{aligned}a_1^{(1)} &= w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + w_{13}^{(1)} x_3 \\&= 0.1 \times 1 + 0.1 \times 2 + 0.1 \times 3 \\&= 0.6\end{aligned}$$



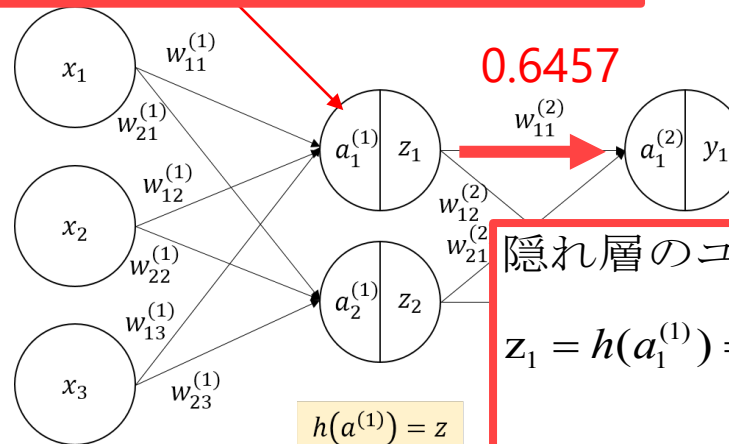
ニューラルネットワークの計算例：順方向

33

- 入力層ユニット数3
- 隠れ層ユニット数2
- 出力層ユニット数2
- 隠れ層の活性化関数
- 出力層の活性化関数

隠れ層のユニット1の入力

$$\begin{aligned} a_1^{(1)} &= w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + w_{13}^{(1)} x_3 \\ &= 0.1 \times 1 + 0.1 \times 2 + 0.1 \times 3 \\ &= 0.6 \end{aligned}$$



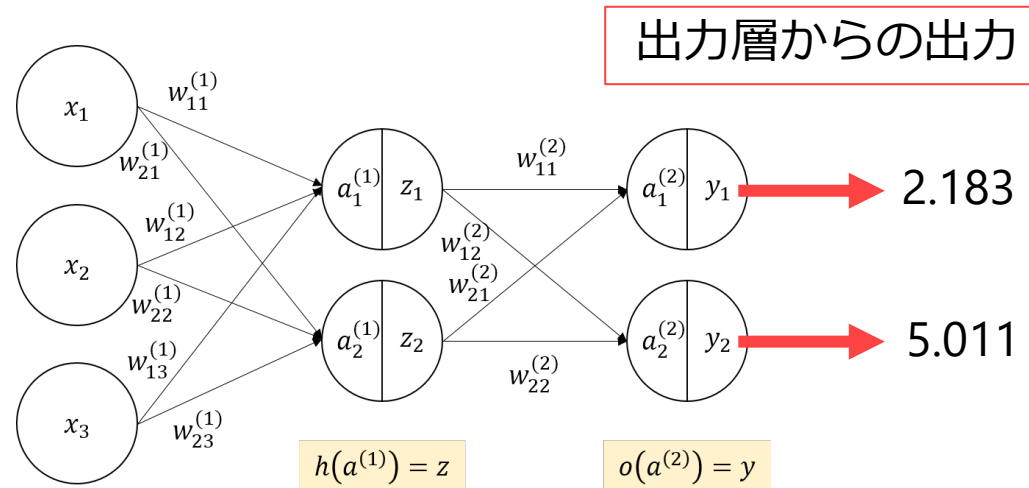
隠れ層のユニット1の出力

$$\begin{aligned} z_1 &= h(a_1^{(1)}) = \frac{1}{1 + \exp(a_1^{(1)})} \\ &= \frac{1}{1 + \exp(0.6)} \\ &\approx 0.6457 \end{aligned}$$

ニューラルネットワークの計算例：順方向

34

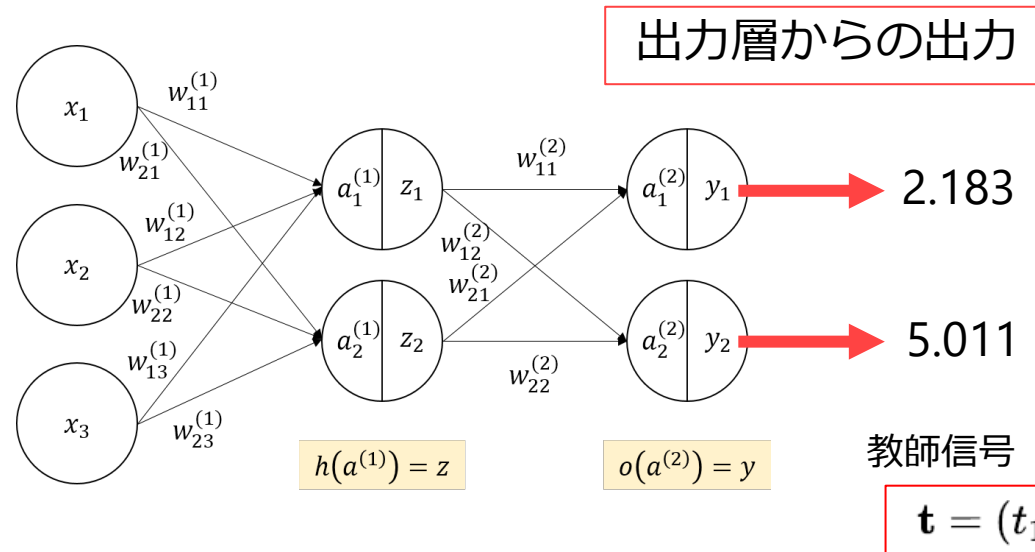
- 入力層ユニット数3
- 隠れ層ユニット数2
- 出力層ユニット数2
- 隠れ層の活性化関数：シグモイド関数 $\frac{1}{1 + e^{-ax}}$
- 出力層の活性化関数は恒等関数 $o(a)=a$



ニューラルネットワークの計算例：順方向

35

- 入力層ユニット数3
- 隠れ層ユニット数2
- 出力層ユニット数2
- 隠れ層の活性化関数：シグモイド関数 $\frac{1}{1 + e^{-ax}}$
- 出力層の活性化関数は恒等関数 $o(a)=a$



ニューラルネットワークに(1, 2, 3)を入力したときに出力してほしい値

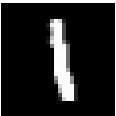


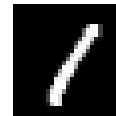








本日の内容

36

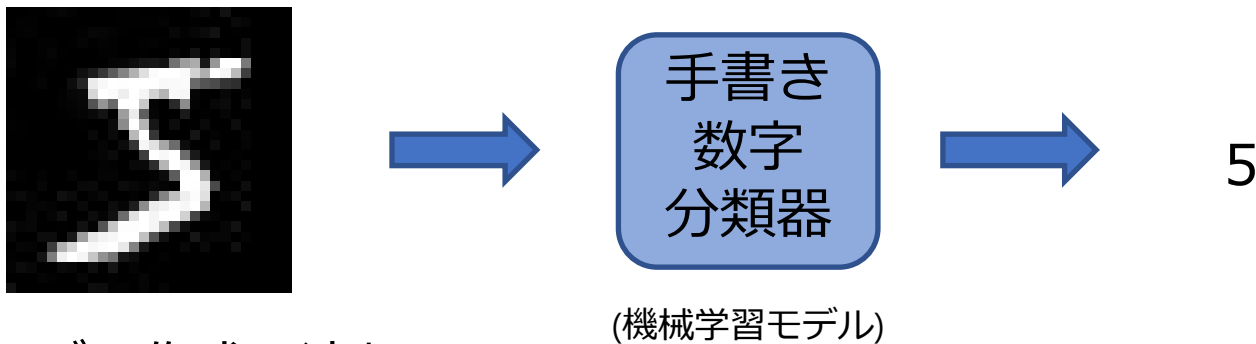
- 機械学習の分類
- ニューラルネットワークの構成
 - 種類
 - 単純パーセプトロン
 - 多項ロジットモデル
 - 多層ニューラルネットワーク
 - 学習の概要
- ニューラルネットワークを試す
 - A Neural Network Playground

❖ 例題：手書き数字の画像を入力として書かれている数字が何か分類

■ 手書き文字画像

画像												...	
教師ラベル	1	2	9	1	4	0	5	3	4	7	6	...	5

■ 作りたい分類器のイメージ



❖ 学習モデル作成の流れ

■ データの分割 (モデルによる推論) → 機械学習モデルの学習 → モデルの評価 (→ モデ

❖ 学習モデル作成の流れ

■ **データの分割** → 機械学習モデルの学習 → モデルの評価 (→ モデルによる推論)

❖ データの分割

■ 手持ちのデータを学習用の**訓練データ**と評価用の**テストデータ**に分割

手持ちデータ

画像

教師
データ

1	2	9	1	4	0	5	3	4	7	6	...	5
---	---	---	---	---	---	---	---	---	---	---	-----	---

分割

- 分割の割合に決まりはない (例...訓練 : テスト = 8 : 2)
 - 基本的に訓練データの数が多の方がモデルの訓練には良い
 - テストデータの数が少ないと評価の妥当性が問題となる
- 訓練データの一部を検証データとして全体で3分割とする場合もある (後述します)

訓練データ

テストデータ

❖ 学習には以下の3種類のデータが必要

- 持っているデータを分割する

■ データの種類

■ 訓練データ

- モデルのパラメータを設定するためのデータ
- 主に全データの6～8割が用いられる

■ 検証データ

- ハイパーパラメータ(人が決定すべき項目)を検証するためのデータ
- 全データの1～2割程度

■ テストデータ

- 学習済みモデルの汎化性能を評価するためのデータ
(実運用で想定される精度がここでわかる)
- 全データの1～2割程度

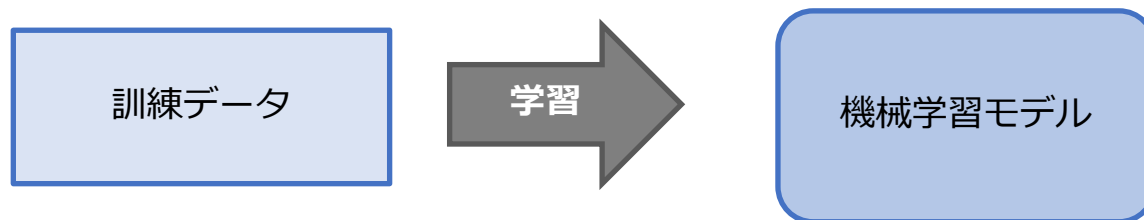


❖ 学習モデル作成の流れ

- データの分割 → **機械学習モデルの学習** → モデルの評価
(→ モデルによる推論)

❖ 機械学習モデルの学習

- 訓練データを使用してモデルを学習させる



~~テストデータ~~

※ここではテストデータを使用しない

❖ 学習モデル作成の流れ

- データの分割 → 機械学習モデルの学習 → **モデルの評価**
(→ モデルによる推論)

ポイント



モデル作成

- テス

- ❖ 近年のモデルは優秀なので、過去に入力されたデータは覚えてしまうことが多い
- ❖ つまり、本当に知りたい“未知”のデータに対する精度が正しく評価できなくなってしまう

→ 訓練データとテストデータに同じデータを含めてはいけない！

訓練データ

※ここでは訓練データを使用しない

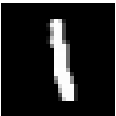











テストデータ

評価

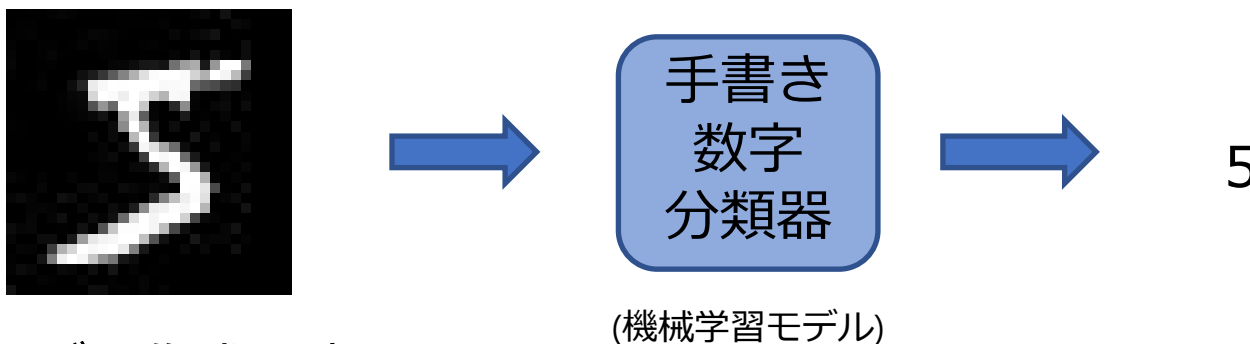
機械学習モデル
(学習済み)

❖ 例題：手書き数字の画像を入力として書かれている数字が何か分類

■ 手書き文字画像

画像												...	
教師ラベル	1	2	9	1	4	0	5	3	4	7	6	...	5

■ 作りたい分類器のイメージ

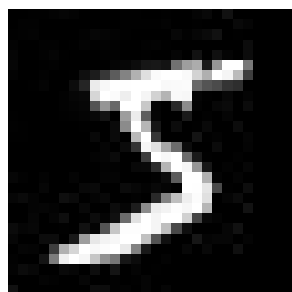


❖ 学習モデル作成の流れ

■ データの分割 (モデルによる推論) → 機械学習モデルの学習 → モデルの評価 (→ モデ

❖ 手書き数字を分類する深層学習モデル

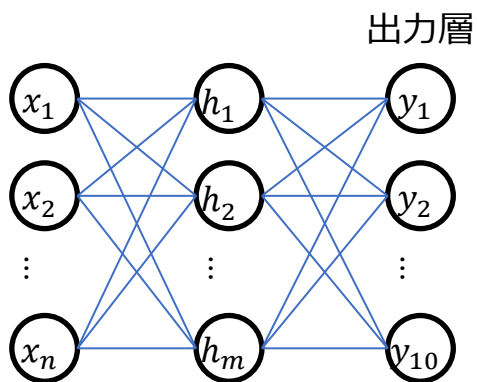
- アルゴリズム：ニューラルネットワーク
- データセット：MNIST



(機械学習モデル)



5

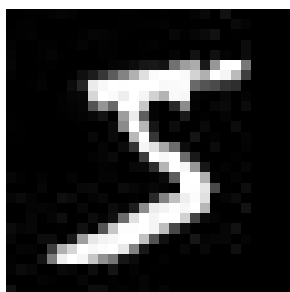


[0,
0,
0.2,
0,
0,
0.8
0,
0,
0,
0]

10個の数値が入った配列

❖ 手書き数字を分類するAI

- アルゴリズム：ニューラルネットワーク
- データセット：MNIST

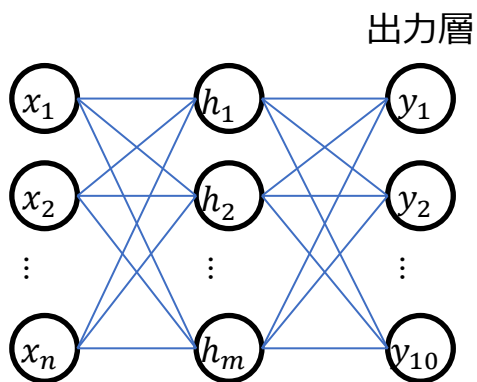


手書き
数字
分類器

(機械学習モデル)

出力層で10個の値を出力させ、それぞれを各数字の確率として扱う

- 1番目：入力画像が「0」である確率
- 2番目：入力画像が「1」である確率
- ...
- 10番目：入力画像が「9」である確率



SoftMax
関数

[0,
0,
0.2,
0,
0,
0.8
0,
0,
0,
0]

10個の数値が入った配列

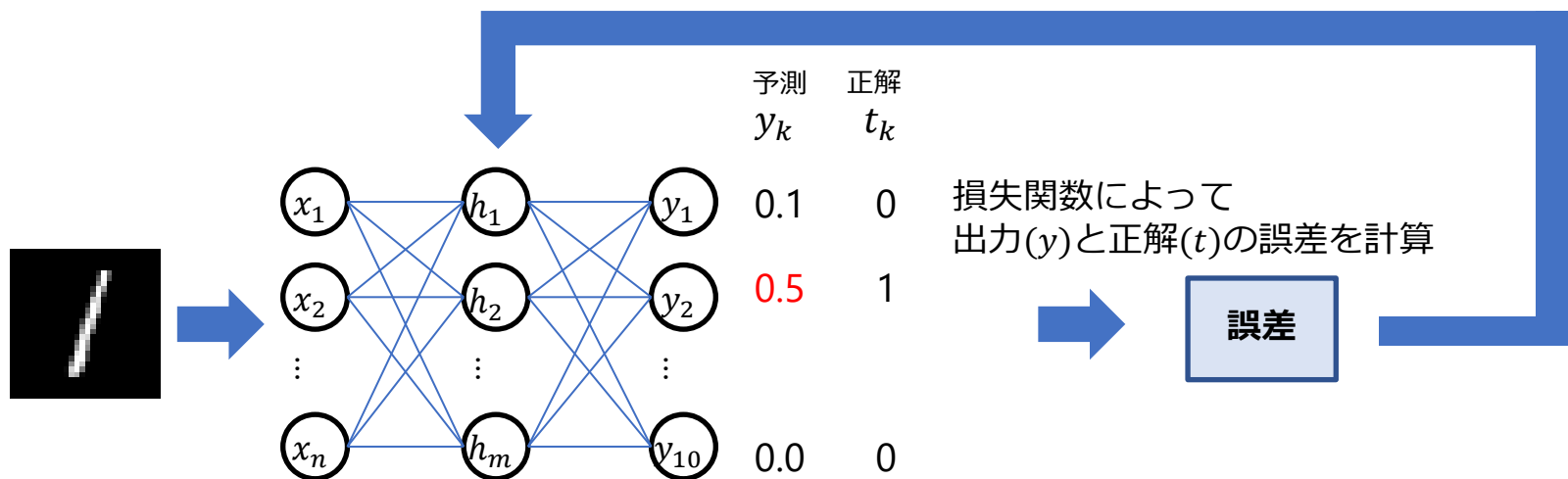
❖ ニューラルネットワークの学習とは

- 損失関数を最小化するパラメータ(テンソル W)を求めること

❖ 損失関数

- 出力の値(y)と正解の値(t)の誤差を表す関数
 - 間違ってる場合には大きい値、正しい場合には小さい値をとる関数を設定

誤差が小さくなるようにニューラルネットワークのパラメータ(テンソル W)を修正



ニューラルネットワークの教師あり学習

46

1. 入力層に入力信号を与えて、順方向に中間層の出力を計算して、出力層からの出力を計算
2. 出力層からの出力と教師信号に損失関数を適用して誤差を計算
3. **誤差逆伝播法**を用いて、出力層から入力層に向かう逆方向に各ユニット間の重みに対する損失関数の勾配を計算
4. この勾配を用いた誤差を最小化するために、**最急降下法**を適用して各ユニット間の重みを更新
5. 1-4 の繰り返し

本日の内容

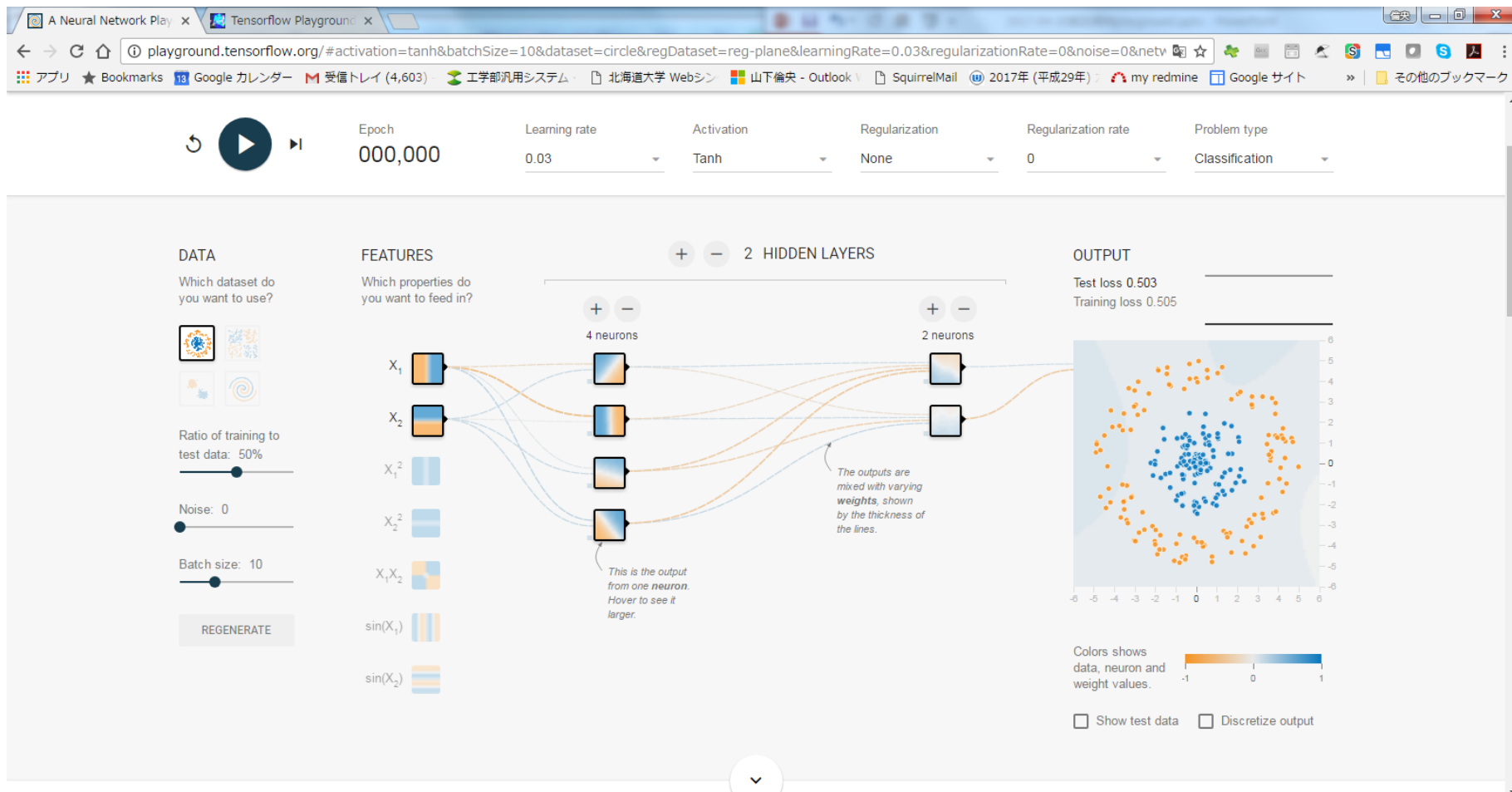
47

- 機械学習の分類
- ニューラルネットワークの構成
 - 種類
 - 単純パーセプトロン
 - 多項ロジットモデル
 - 多層ニューラルネットワーク
 - 学習の概要
- ニューラルネットワークを試す
 - A Neural Network Playground

A Neural Network Playground

48

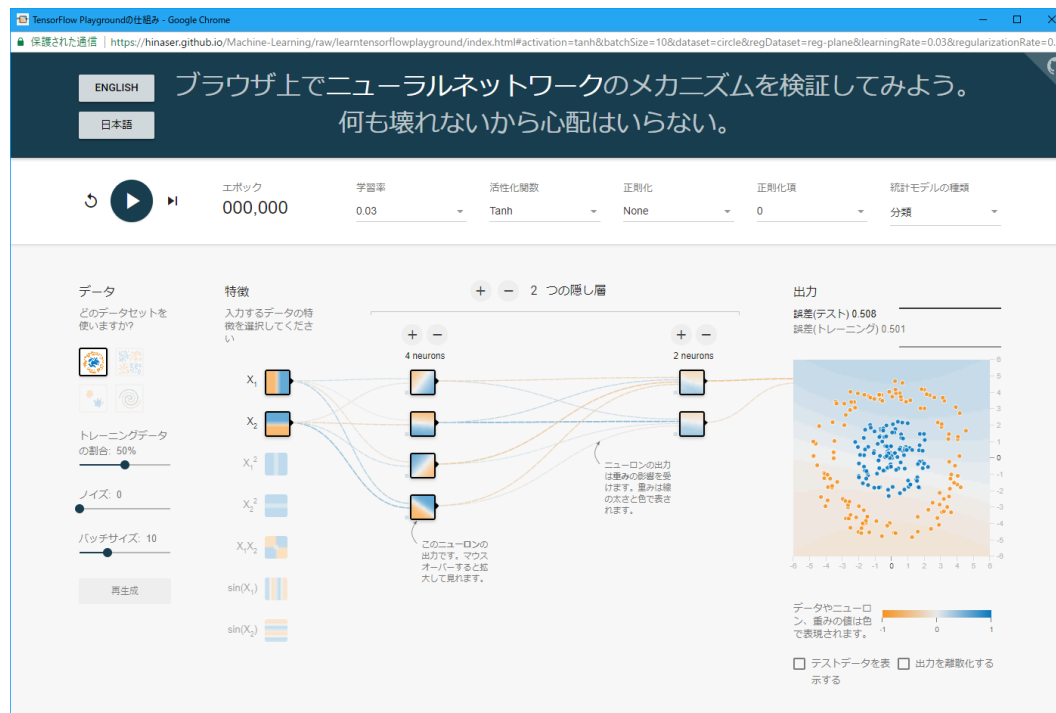
- <http://playground.tensorflow.org/>



A Neural Network Playground日本語版

49

- 日本語版「TensorFlow Playgroundの仕組み」で検索
- <https://hinaser.github.io/Machine-Learning/raw/learntensorflowplayground/index.html#activation=tanh&batchSize=10&dataset=circle®Dataset=reg-plane&learningRate=0.03®ularizationRate=0&noise=0&networkShape=4,2&seed=0.73678&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=classification&initZero=false&hideText=false&lang=jp>



A Neural Network Playground

日本語版解説

50

- <https://hinaser.github.io/Machine-Learning/index.html>

TensorFlow Playgroundの仕組み

はじめに
参考文献
Playgroundの仕組み
ニューラルネットワークについて
ニューラルネットワークの学習に関する設定
学習率
活性化関数
正則化、正則化項
統計モデルの種類
データに関する設定
どのデータセットを使うか
トレーニングデータの割合
ノイズ
バッチサイズ
入力するデータの特徴と隠し層
バックプロパゲーション(誤差逆伝搬法)

TensorFlowでディープラーニング

TensorFlow コードサンプル

機械学習のための数学

その他

機械学習、ディープラーニング

TensorFlow Playgroundの仕組み

2017-02-14 Hinase

TensorFlow Playgroundの公式サイトは[こちら](#)。
本家のTensorFlow Playgroundの日本語訳を作成しましたので、本サイトと合わせてご覧ください。

TensorFlow Playgroundとは何を言葉で説明するより一度自分の目で直接見るほうが早いので、上記のリンクをクリックしてTensorFlow Playgroundを開き、思いつくまゝいろいろ自分で試してみてください。ディープラーニングの仕組みが直感的になんとなく分かると思います。

今回、直感的になんとなく分かる、で満足できなかった人のためにこのサイトを作成しました。
実際にTensorFlow Playgroundは何をしているのか、数ある設定項目はどんな意味を持つのか。
本ページではTensorFlow Playgroundの裏側に詳しく解説します。

私自身は機械学習の研究者でもなく、関連した仕事をしているわけでもありません(そういった仕事に興味はありますが)。ですのでところどころ誤りや間違いが見つかると思います。もし誤りを見つけれましたら[私のツイッター](#)までご連絡ください。

参考文献

はじめにこのサイトを作るにあたって参考にした資料を載せておきます。ここで解説する内容は大人小なりこれらのサイトの解説を自分の言葉でまとめ直したようなものです。私の拙い文章が分かりにくいと感じたらこれらの参考文献を直接読んでいただいてかまいません。

- [Neural Network TensorFlow入門講座](#)
- [TensorFlow Playgroundでわかるニューラルネットワーク](#)
- [Neural Network and Deep Learning](#)
- [Deep Learning An MIT Press book](#)

先頭の丸山氏による「Neural Network TensorFlow入門講座」は初学者でも非常に理解しやすい内容になっています。パワーポイントでニューラルネットワークの基礎について非常にわかりやすくまとまっていますので一度目を通してみることをおすすめします。

上から2番目のGoogleの佐藤氏による「TensorFlow Playgroundでわかるニューラルネットワーク」と本サイトはTensorFlow Playgroundを題材としている点でコンセプトは似ていますが、こちらのサイトはニューラルネットワーク本論よりはPlayground上の用語や仕組みの解説に重きを置いています。

3番目の「Neural Network and Deep Learning」はMichael Nielsen氏による英文のドキュメントです。私のWebサイトで使う数式はこちらのサイトの説明をかなり参考にしています。文章自体もユーモアがふんだんに取り入れられた、技術解説サイトとは思えない素晴らしいWebサイトです。

A Neural Network Playground

51

The image shows the TensorFlow Playground web interface with several components labeled in Japanese:

- 学習率 (Learning rate):** 0.03
- 活性化関数 (Activation):** Tanh
- 正則化 (Regularization):** None
- 正則化率 (Regularization rate):** 0
- 分類: 1 or -1 回帰: [-1,1] (Problem type):** Classification
- 開始・停止 (Start/Stop):** Play button
- Epoch:** 000,000
- 特徴 (Features):** X₁, X₂, X₁², X₂², X₁X₂, sin(X₁)
- データセット (Dataset):** Which dataset do you want to use?
- 出力 (Output):** Test loss 0.503, Training loss 0.505
- 隠れ層 (Hidden layers):** 2 HIDDEN LAYERS, 4 neurons
- 表示 (Display):** Checkboxes for "Show test data" and "Discretize output"
- ノイズ (Noise):** 0
- バッチサイズ (Batch size):** 10
- 分類: 500個のトレーニング/テストの割合 回帰: 1200個のトレーニング/テストの割合 (Ratio of training to test data):** 50%

The interface also includes a visual representation of the neural network with weights shown by line thickness and a scatter plot of the data points.

統計モデルの種類

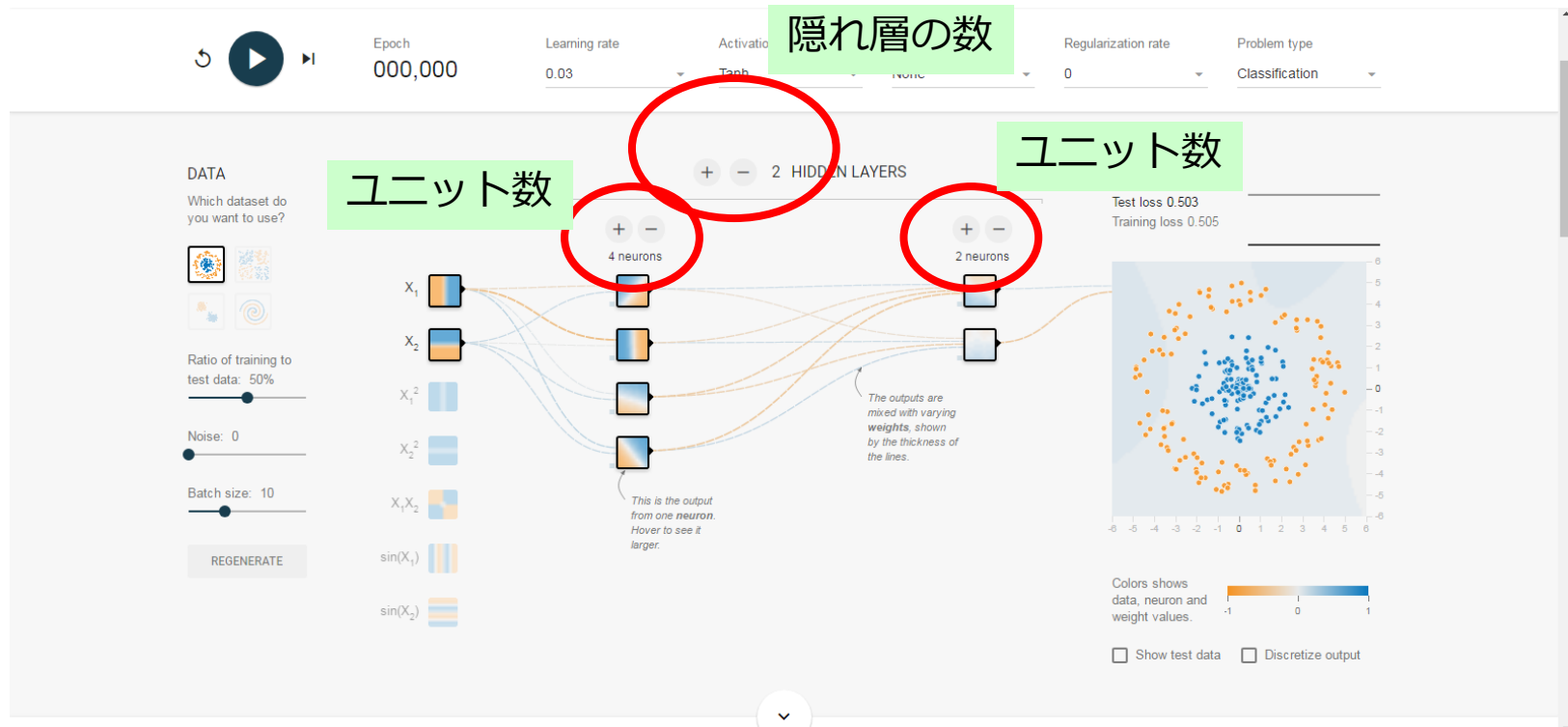
52

- 分類
 - ラベルは離散値
 - 0 or 1
 - ※ 多クラス分類問題もある
 - 例) 入力した画像が猫であるか猫でないかを0/1で判断
- 回帰
 - ラベルは連続値
 - 例) 入力した画像が猫である確率

隠れ層

53

- 隠れ層の設定
 - 層の数
 - 各層のユニット数
- 「+」 「-」 ボタンで調整



隠れ層

- 入力層・隠れ層・出力層の対応

