

27<sup>th</sup> Apr 2022

スタートアップゼミ

# 機械学習と連続量最適化

---

機械学習モデルと学習の基礎

望月陽介

# 今回の目標

- 機械学習

- 初学者・聞いたことだけある人

→「機械学習とは？」のイメージ、「モデルと学習」の理解

- 学んだことがある人

→機械学習の表現について整理

- 連続量最適化

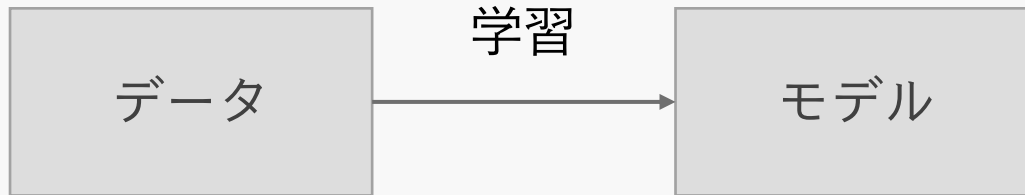
- 微分を用いた最適化アルゴリズム

- 機械学習の中での仕事

# 機械学習とは？

- 機械学習

- モデルが**データ**や**経験**を通してルールを自動で獲得する手法



パターン認識

意思決定

- ルールベースAI

- 記述されたパターンをルールとする手法



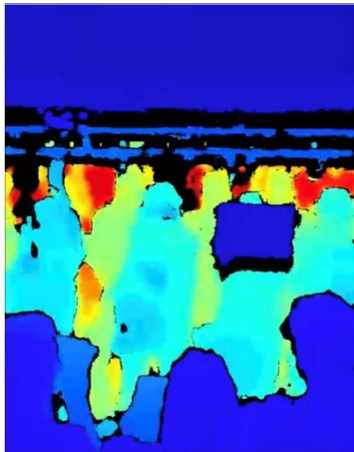
パターン認識

意思決定

# 機械学習とは？

- 利用の一例

## 画像認識



デプスカメラ撮影画像イメージ  
(撮影画像は上記イメージであり、プライバシーは確保されています。)

東京メトロの列車混雑計測システム  
<https://www.tokyometro.jp/news/2021/209731.html>

## 自然言語処理

機械翻訳 DeepL  
<https://www.deepl.com/ja/translator>

# 機械学習の分類

- 教師あり学習

答えのある  $(x, y)$  のペアから、対応関係  $y = f(x)$  を学習する

例) 交通手段選択予測、物体認識、株価予測. . .

- 教師なし学習

データ  $X$  の構造を解析する

例) 購買者のクラスタリング、行列処理. . .

- 強化学習

得られる報酬に基づき行動選択を行うモデルの構築

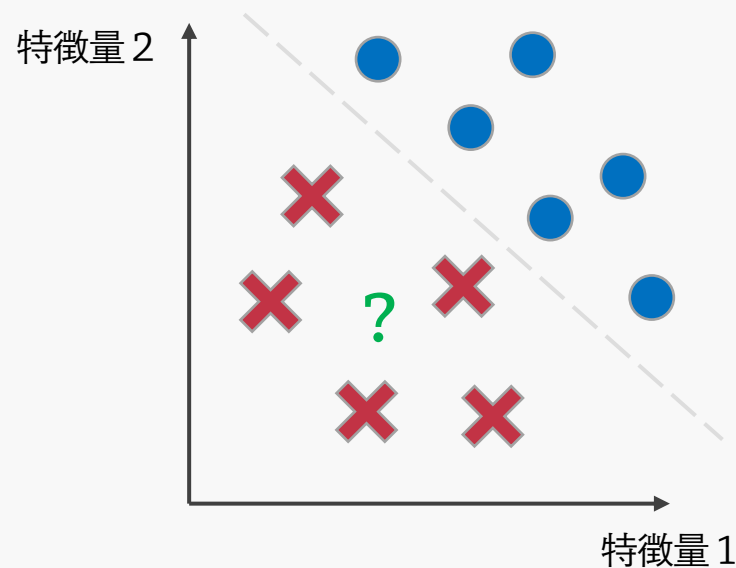
例) Alpha Go、ロボット、コロナ禍の移動選択学習. . .

# 教師あり学習

- 機械学習モデルの処理

- データセット  $D$  に含まれる訓練セット (入力, 出力) =  $(\mathbf{x}_t, \mathbf{y}_t)$  から  $\mathbf{y} = f(\mathbf{x})$  なる関数を学習し、未知の入力  $\mathbf{x}$  に対応する出力  $\mathbf{y}$  を推定する

- 分類の例



- データ訓練セット  $(\mathbf{x}_t, \mathbf{y}_t)$  と、未知のセット  $(\mathbf{x}, \mathbf{y})$  が同質なデータであることが前提

→  $P(\mathbf{x}), P(\mathbf{y}|\mathbf{x})$  が同じ

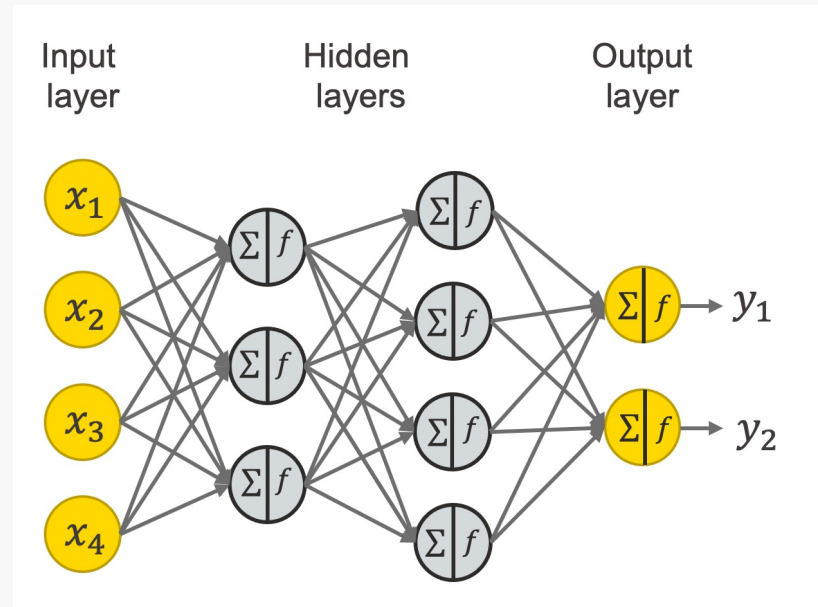
# 教師あり学習

- 関数中のパラメータ

- モデルは、関数 $f$ を用いて出力の予測 $\hat{y}$ を計算する
- 関数 $f$ はパラメータ( $\theta$ )を用いて記述されることが多い (パラメトリックモデル)
- 関数 $f$ の形はモデルの構築時点で決まっている一方、**パラメータはデータに応じて決定する** (時間価値とか、切片とか)

- 関数の例 :

$$f = \operatorname{argmax}_i \left( \frac{\exp(\mu V_i)}{\sum_{j \in C} \exp(\mu V_j)} \right),$$
$$V = \theta \cdot x$$



# 教師あり学習

- パラメータの決定

- 関数 $f$ を用いて出力の予測 $\hat{\mathbf{y}}$ を計算するため、**パラメータを決定**する必要がある
- 予測 $\hat{\mathbf{y}}$ と実際 $\mathbf{y}$ の出力が近い時、モデルの精度は高いと言える
- 予測と実値が最も近くなるように、パラメータ( $\boldsymbol{\theta}$ )を調整

- 「近さ」の尺度

- 損失関数、誤差関数、目的関数

$$f\left(\begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}\right) = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \vdots \\ \hat{\mathbf{y}}_n \end{bmatrix} \leftrightarrow \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_n \end{bmatrix}$$

予測                      実際

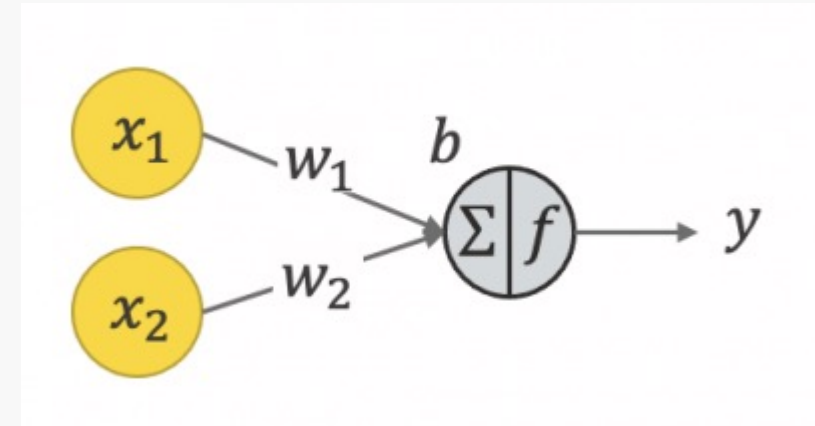
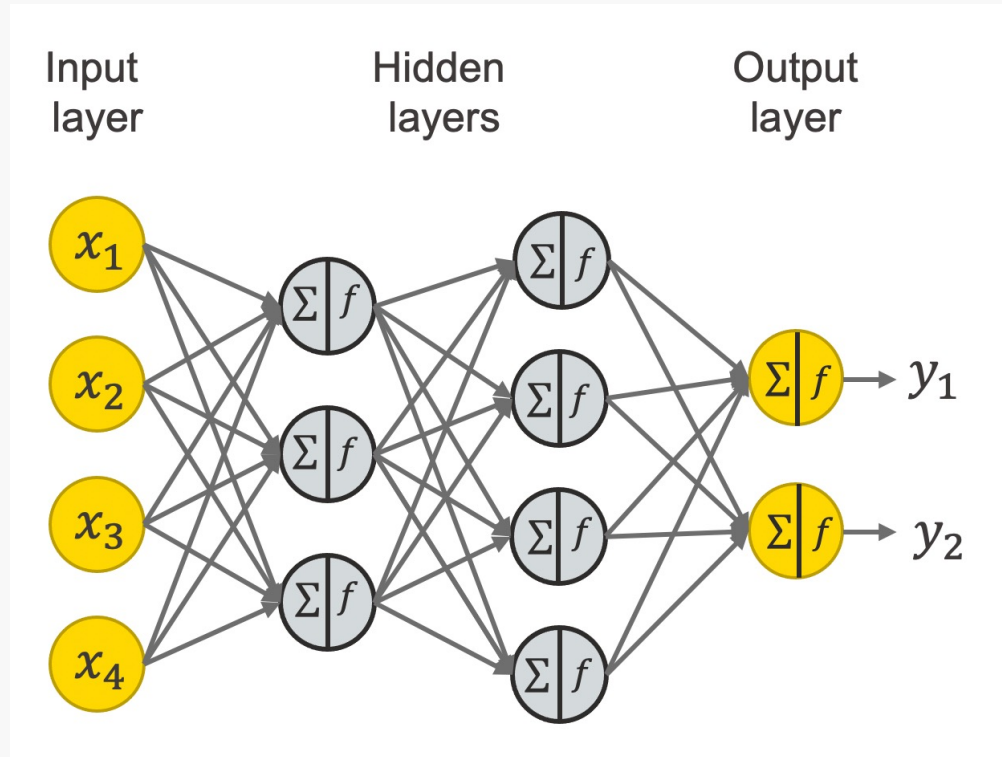
$$Loss(\boldsymbol{\theta}) = - \sum_{n=1}^N \sum_i y_{in} \cdot \log P_n(i|\boldsymbol{\theta})$$

- これを最小(大)化することで、最適なパラメータを得る



# 閑話休題 深層学習

- 深層学習(Deep Neural Net)



パラメータ  $w$ ,  $b$ 、活性化関数  $f$  による計算を多層に繰り返す

<https://www.knime.com/blog/a-friendly-introduction-to-deep-neural-networks>

# 閑話休題 深層学習

- 画像処理←CNN
- 自然言語処理←RNN
- ネットワーク構造←GNN
- 生成モデル←GAN
- 画像・言語←Transformer

# 閑話休題 Tips

- 使ってみたい時

- モデルを使いたい時はとりあえずネットで検索して、ライブラリ等を探す

例) scikit-learn, tensorflow, PyTorch, Keras...

PyTorchのドキュメント : <https://pytorch.org/docs/stable/index.html>

- ライブラリの使い方をまとめてくれてるページを探す

例) Qiita, 英語のページ...

Pytorchで「月とすっぽん」の画像認識を試みた :

<https://qiita.com/ryryry/items/b1da4855504dcd3f9d98>

- 新しいモデルや、学習済みモデルはgithubに載っていることが多い

自分でコードを変えられるメリット

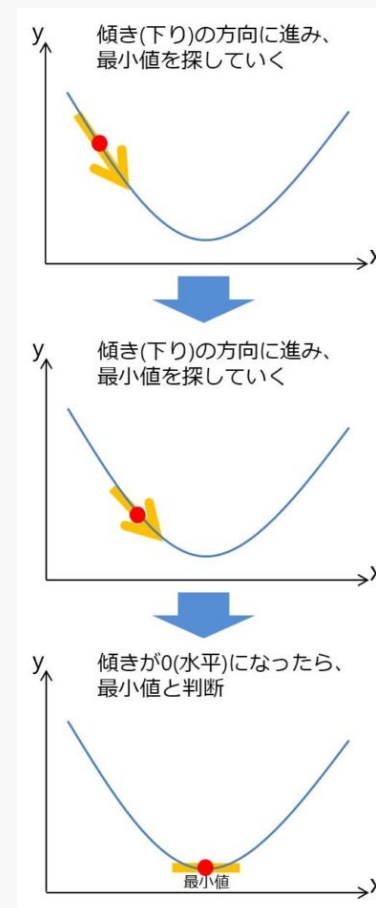
# モデルの学習

- モデルの学習=パラメータ決定=最適化

- $\frac{\partial Loss}{\partial \theta_k} = 0 \quad \forall k$  を解いて、損失関数を最小化する最適な  $\theta$  を求める ← 解析解難

- 勾配降下法

$\theta = \theta - \eta \frac{\partial Loss}{\partial \theta}$  を繰り返すことで解に至る



<https://nisshingeppo.com/ai/gradient-descent/>

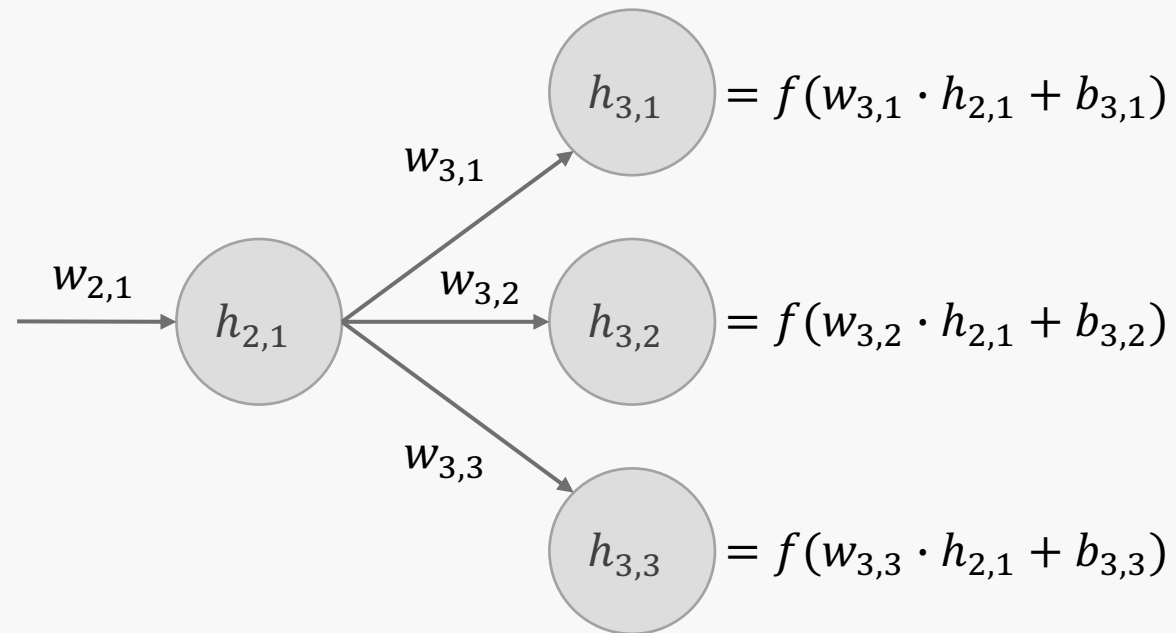
# モデルの学習

- 微分の計算方法

- 誤差逆伝播
- 自動微分を用いることで、順計算一回のみで全てのパラメータに対する微分が求まる
- 連鎖率によって、後ろ向きに誤差を伝達

$$\frac{\partial Loss}{\partial w_{2,1}} = \frac{\partial Loss}{\partial h_{2,1}} \cdot \frac{\partial h_{2,1}}{\partial w_{2,1}}$$

$$\frac{\partial Loss}{\partial h_{2,1}} = \sum \frac{\partial Loss}{\partial h_{3,i}} \cdot \frac{\partial h_{3,i}}{\partial h_{2,1}}$$



# まとめ

- 機械学習モデル(教師あり)はデータから関数を推定することで、未知のデータに対する予測を行う
- 予測のため行う学習では、訓練データで計算される損失関数を最小化するパラメータを推定し(=最適化)、予測精度を保証する
- 深層学習モデルの学習では、誤差逆伝播を用いて微分を求め、勾配降下法が用いられる

# モデルの学習（補足）

- 補足：ステップサイズ計算の有無

- step-sizeをWolfe条件等の制約することで、短調減少や収束性を保証

一方で、

- 誤差逆伝播を用いて微分値の計算コストを小さくできる場合は、何度も損失関数の評価を行うFrank-Wolfe法のようなstep-size決定手順を設けると計算コストがより大きくなる

↓

- パラメータ更新の運動量を考慮した収束速度向上、確率的なサンプリングによる大局解への収束性を上げる工夫がされる

# 参考

- urlと書籍まとめて後日加えます
- 松尾研 「深層学習」 第二回抗議資料