

# 機械学習講習会

## 第一回：学習とは何か

2025/06/25

@Kobakos32

# 自己紹介

- ハンドル： @Kobakos32
- 所属： 23B 情報工学系
- 好きな分野： 画像系

Kaggle Expert, Signate Master  
自作キーボード・料理も好きです



# 講習会全体の見通し

- この講習会で目指すこと:
  - 機械学習コンペに参加するための基礎知識を身に着ける
  - PyTorchを使って自分でニューラルネットワークが実装できるようになる

## ニューラルネットワークはすべての基礎

いわばもっとも汎用的な部品。chatGPTにも使われている。  
使わない場合もあるが、考え方は共通している。

# 講習会全体の見通し

1. **第一回：学習とは何か (今回)**
  - 機械学習の種類、教師あり学習の概要、線形回帰の導入
2. **第二回：線形回帰と勾配降下法 (06/27)**
  - 線形回帰を微分を使って学習する
3. **第三回：ニューラルネットワークの基礎 (07/01)**
  - 線形回帰よりも複雑な表現を可能にするニューラルネットワークを学ぶ

## 講習会全体の見通し (続き)

### 4. 第四回：誤差逆伝播法 (07/03)

- ニューラルネットワークをどうやって微分するのか

### 5. 第五回：ニューラルネットワークの訓練 (07/08)

- ニューラルネットワークを実装して訓練してみよう

## 講習会全体の見通し (さらに続き)

### 6. 第六回：部内コンペキックオフ (07/10)

- 部内コンペのルール説明、コンペサイトに登録
- スターターノートブックの解説・ファーストサブを出す

### 7. 第七回：部内コンペ振り返り (07/18)

- コンペの結果発表、振り返り、これからについて

# 本題

# もくじ

1. 機械学習の種類
2. 教師あり学習の概要
3. 線形回帰の導入

## 機械学習の主な種類

- 教師あり学習 (Supervised Learning)
- 教師なし学習 (Unsupervised Learning)
- 強化学習 (Reinforcement Learning)

ただし、厳密な区分ではなく、半教師あり学習といった言葉もあります。

## 教師あり学習 (Supervised Learning) とは？

- **定義:** 入力データとそれに対応する「正解」の出力データ（ラベル）のペアを使って、モデルを学習させる方法。
- **目的:** 未知の入力データに対して、正しい出力を予測できるようにする。

本講習会で扱うものはこれ

## 教師あり学習の主なタスク

- **回帰 (Regression):** 連続値を予測する。
  - 例1：[住宅の広さ、築年数] から [住宅価格] を予測する。
  - 例2：[過去の気温、湿度] から [明日の最高気温] を予測する。
  - 例3：[広告費用] から [商品の売上] を予測する。

## 教師あり学習の主なタスク (続き)

- **分類 (Classification):** データがどのカテゴリに属するかを予測する。
  - 例1：[メールの本文] から [スパムメールか否か] を判定する。
  - 例2：[画像] から [写っている動物の種類 (犬、猫など)] を識別する。
  - 例3：[顧客の購買履歴、属性] から [特定のキャンペーンに反応するか否か] を予測する。

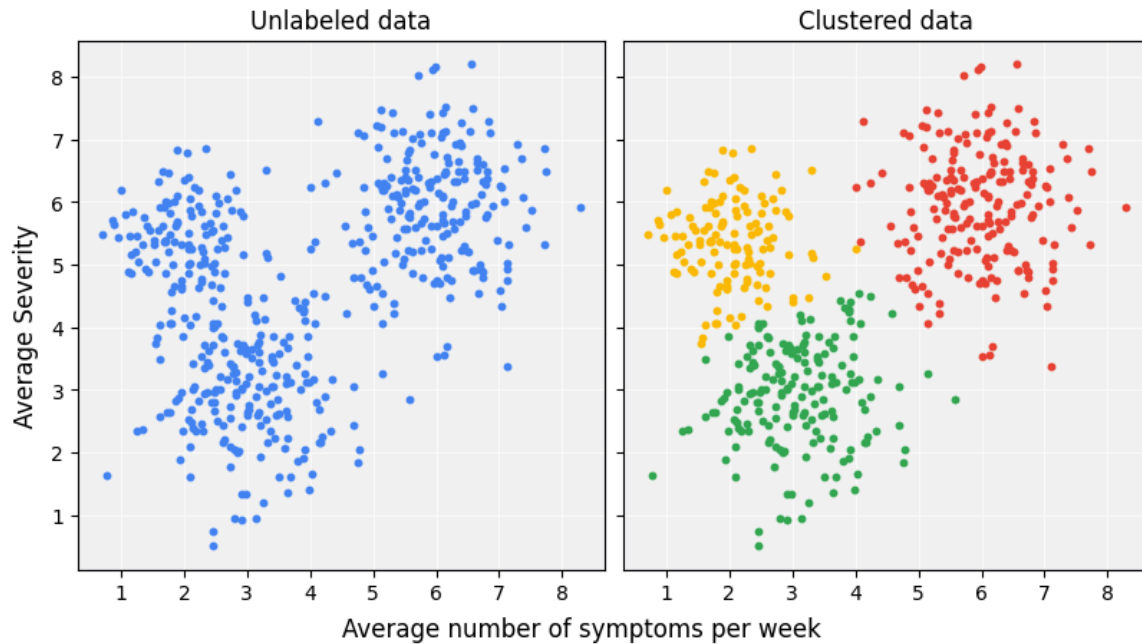
## 教師なし学習 (Unsupervised Learning) とは？

- **定義:** 「正解」の出力データ（ラベル）がない入力データのみを使って、データに潜むパターンや構造を見つけ出す方法。
- **目的:** データの本質的な特徴を捉えたり、データをグループ分けしたりする。

## 教師なし学習の主な例: 異常検知

- **異常検知 (Anomaly Detection):** 他のデータと大きく異なるパターンを持つデータ (異常値) を検出する。
  - 例1: [クレジットカードの取引履歴] から [不正利用の可能性が高い取引] を見つける。
  - 例2: [工場のセンサーデータ] から [機械の故障予兆] を検知する。

# 教師なし学習の主な例：クラスタリング

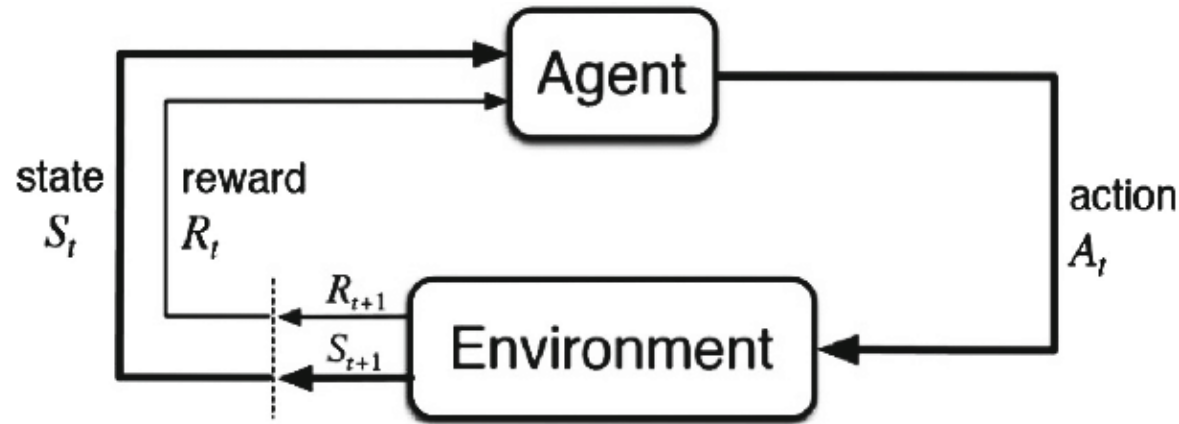


データをいい感じに分ける  
"正解"となるラベルはなしにやる

# 強化学習 (Reinforcement Learning) とは？

- **定義:**「エージェント」が「環境」内で試行錯誤を繰り返す学習方法。
- **学習のサイクル:**
  - i. エージェントが現在の**状態**を観測する。
  - ii. ある**行動**を選択・実行する。
  - iii. 行動の結果、環境が次の**状態**に変化する (**状態遷移**)。
  - iv. エージェントは環境から**報酬** (または罰) を受け取る。
- **目的:** 将来にわたって得られる総報酬を最大化するような行動方針 (方策) を学習すること。

# 強化学習とは



エージェントが環境に対して行動を取り、環境から報酬と新たな状態を受け取る  
(状態遷移)

## 強化学習の具体例: ゲームAI

- ゲームAI:
  - エージェント：ゲームキャラクター
  - 状態：ゲームの盤面やステージ
  - 行動：キャラクターの操作（移動、攻撃など）
  - 報酬：スコアアップ、ステージクリア（プラスの報酬）、ゲームオーバー（マイナスの報酬）

# 強化学習の具体例: LLM (大規模言語モデル)

- LLM (大規模言語モデル):
  - エージェント：モデル
  - 状態：文脈
  - 行動：次に続く単語の選択
  - 報酬：ユーザーの評価、問題に正解したか、報酬モデル

# 本講習会の内容：教師あり学習

## (再掲) 教師あり学習 (Supervised Learning) とは？

- **定義:** 入力データとそれに対応する「正解」の出力データ（ラベル）のペアを使って、モデルを学習させる方法。
- **目的:** 未知の入力データに対して、正しい出力を予測できるようにする。

入力データを  $x$ 、出力データを  $y$  とします。

## 教師あり学習：関数近似という考え方

- 教師あり学習の核心: 入力データ  $x$  から出力データ  $y$  を予測する「未知の関数  $f$ 」を近似的に見つけ出すこと。
  - $y \approx f(x)$
- モデルとは: この関数  $f$  の具体的な形（数式やアルゴリズム）のこと。
  - 意味としては模型に近い(現実世界の関係を模したもの)
  - 例：線形回帰モデル、決定木モデル、ニューラルネットワークモデルなど。

## 関数近似：具体例

- **例1:** LLM (大規模言語モデル)
  - 入力  $x$ : テキスト
  - 出力  $y$ : 入力テキストの次に来る単語のid
  - モデル  $f$ : テキストを入力にとって次の単語を予測する関数

繰り返し適用することで文章生成ができる

- **例2:** 画像認識 (犬・猫)
  - 入力  $x$ : 画像データ (数字の配列)
  - 出力  $y$ : その画像の"犬らしさ"
  - モデル  $f$ : 画像を入力にとって0から1の値を出力する関数

## 教師あり学習：関数近似という考え方

- パラメータとは: モデルの振る舞いを決定する調整可能な値。
  - モデルの出力を調整するためのつまみのようなもの。
  - 関数の形を決めるための値

## 学習：データからパラメータを調整すること

- 「学習する」とは: 訓練データに基づいて、現実世界を模したモデルという関数のパラメータを"適切に"調整すること。
- 目的:  $f(x)$  ができるだけ  $y$  に近づくようにする。
  - つまり、モデルが訓練データをうまく表現できるようにする。  
(これはほぼ保証できる)
  - 訓練データ以外のデータもうまく表現できるようにしたい。  
(これは"うまく"学習しないとできない)

# 損失関数 (Loss Function) とは？

- **定義:** モデルの予測値と実際の正解値の「誤差」を数値化する関数
  - 予測値を  $\hat{y}$  (y hat)・正解値を  $y$  とすると：
  - 損失関数  $L(y, \hat{y}) =$  予測がどれだけ「間違っているか」の指標
- **学習の目標:** 全訓練データにおける損失の合計（または平均）を最小化すること
  - $\min \sum_{i=1}^n L(y_i, \hat{y}_i)$

# 学習フェーズと推論フェーズ

## 1. 学習 (Training / Learning) フェーズ:

- 目的: 訓練データセットを用いて、モデルのパラメータをいい感じに調整する。
- 入力: 訓練データとはじめの関数
- 処理:
  - a. モデルの選択
  - b. 損失関数の定義
  - c. 最適化アルゴリズムによるパラメータの調整
- 出力: 学習済みモデル (最適化されたパラメータを持つモデル)

# 推論 (Inference / Prediction) フェーズ

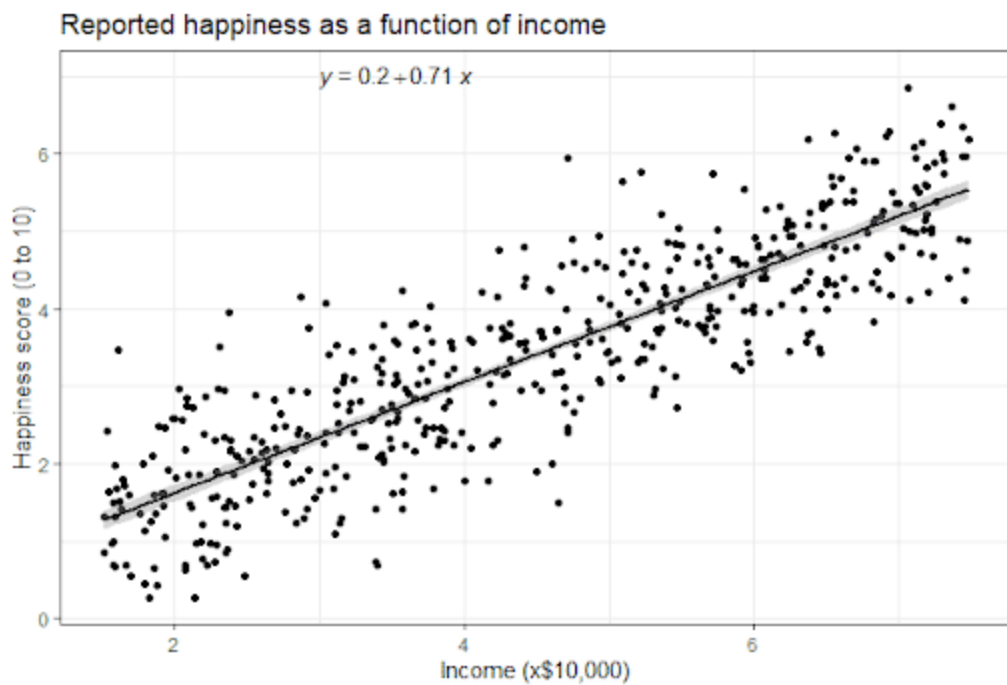
## 2. 推論 (Inference / Prediction) フェーズ:

- **目的:** 学習済みモデルを用いて、新しい未知のデータに対して予測や判断を行う。パラメータの更新は行われぬ。
- **入力:**
  - 学習済みモデル
  - 新しい入力データ (特徴量のみ、ラベルは未知)
- **処理:** 学習済みモデルのロジックとパラメータに従い計算を行う
- **出力:** 調整された関数の出力 (予測値や分類結果など)

実際にやってみましょう

## 例：線形回帰 (Linear Regression) - 概要

- 最も基本的な教師あり学習アルゴリズムの一つ。
- 目的: 入力変数（説明変数）と出力変数（目的変数）の間に「線形の関係」を仮定し、その関係性をモデル化する。
  - つまり、データポイントに最もよく適合する「直線」や「平面（超平面）」を見つける。



データの関係をもっともよく説明する直線を見つける

## (再掲)教師あり学習：関数近似という考え方

- 教師あり学習の核心: 入力データ  $x$  から出力データ  $y$  を予測する「未知の関数  $f$ 」を近似的に見つけ出すこと。
  - $y \approx f(x)$
- モデルとは: この関数  $f$  の具体的な形（数式やアルゴリズム）のこと。
  - 意味としては模型に近い(現実世界の関係を模したもの)
  - 例：線形回帰モデル、決定木モデル、ニューラルネットワークモデルなど。

## 例：線形回帰 - 2次元の場合 (係数・バイアス)

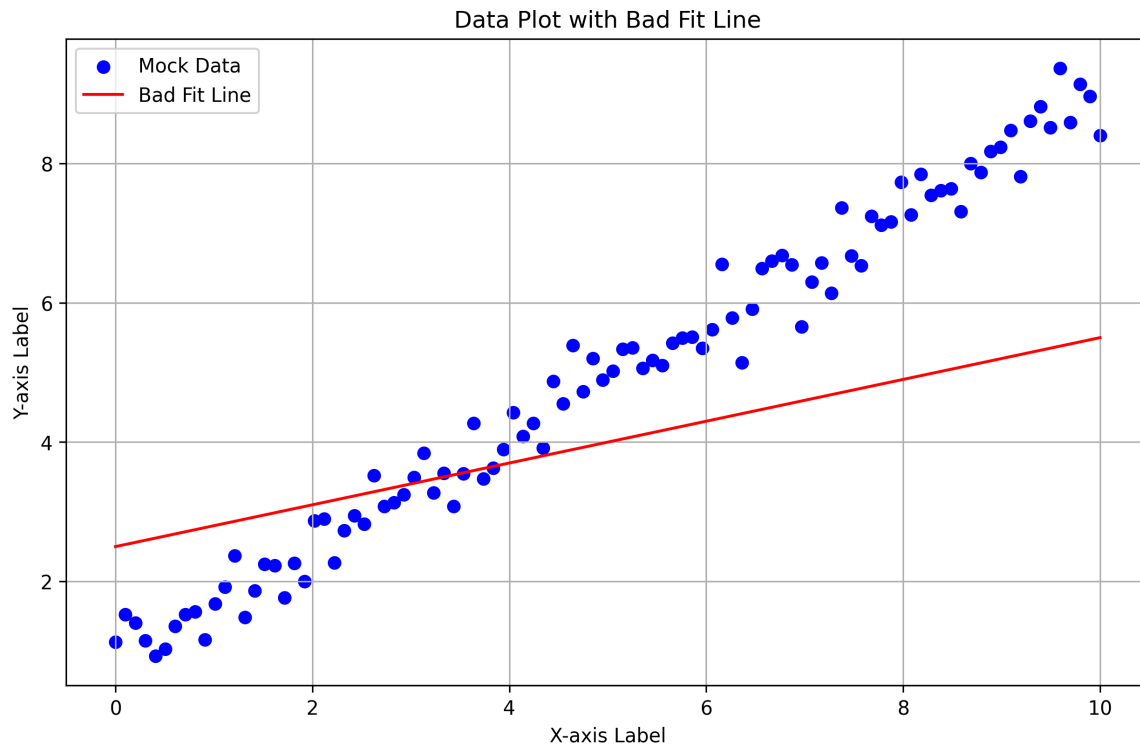
- 最もシンプルなケース: 入力変数が1つ、出力変数が1つの場合。
  - 入力:  $x$  (例: 勉強時間)
  - 出力:  $y$  (例: テストの点数)
- モデルの数式:
  - $y = wx + b$

## 例：線形回帰 - 2次元の場合

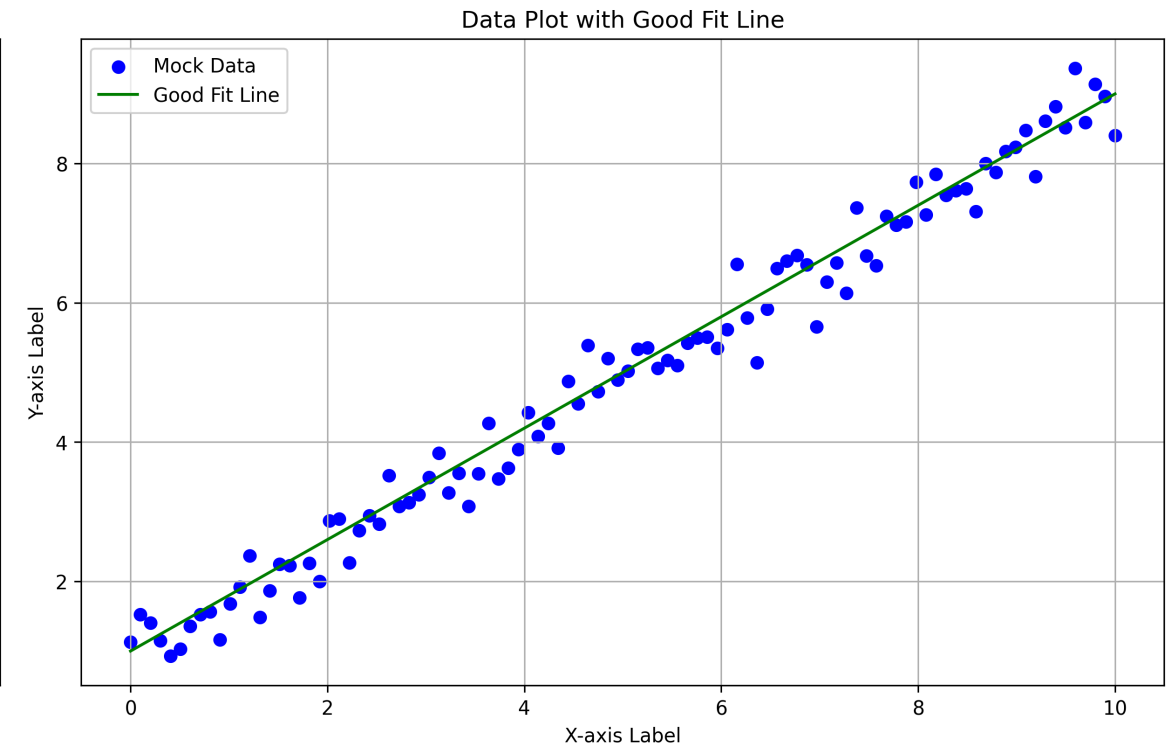
- パラメータ:
  - $w$ : 重み (係数) - 入力  $x$  の影響度を表す
    - 直線の傾き
  - $b$ : バイアス (切片) - 出力  $y$  の基準値
    - 直線が  $y$  軸と交わる点
- 目的: データ  $(x_i, y_i)$  の集まりに最もよくフィットする  $w$  と  $b$  を見つけること

# 例：線形回帰 - 2次元の場合 (学習目標)

- 学習の目標: データ  $(x_i, y_i)$  の集まりに最もよくフィットする  $w$  と  $b$  を見つけること。



これよりも



このほうがよさそう

## 線形回帰の損失関数：平均二乗誤差 (MSE)

- 「最もよくフィット」の定義（の一例）：平均二乗誤差 (Mean Squared Error, MSE) を最小化すること

- $$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^n (y_i - (wx_i + b))^2$$

- 二乗がついてるのは符号を打ち消すため

- 絶対値じゃダメなの？

→ ダメではない (Mean Absolute Error, MAE もある) がそれぞれの特徴がある

# まとめ

- 本日のまとめ:
  - 教師あり学習において、学習はデータから最適なパラメータを見つけるプロセスである。
  - 損失関数は予測値と正解値の「ずれ」を数値化し、学習の目標は損失を最小化することである。
  - 単純な線形回帰では係数と切片がパラメータとなり、MSE（平均二乗誤差）を最小化して最適な直線を見つけることが目標である。

## 次回予告

- 次回: 勾配降下法
  - どうやってパラメータを調整するのか
    - 線形回帰はいろいろな解き方がありますが、(ほぼ) なんにでも使える「勾配降下法」を使います
  - 線形回帰の拡張
    - 複数変数、分類
  - 線形回帰の限界