

機械学習入門

経済学部 BX584

第1回 イントロダクション

自己紹介

- 1995年 理学修士(東大理・情報)
 - 計算幾何に関する研究
- 1999年 学術修士(東大総合文化・科哲)
 - ブレンターノの判断論に関する研究
- 1999~2001年 光学メーカー勤務(富士写真光機)
 - コンパクトカメラのレンズ系の光学設計
- 2004年 情報理工学博士(東大情報理工・電子情報学)
 - Web検索
- 現在、ベイズ理論を使ったデータ・マイニングを研究

Pythonコーディング環境

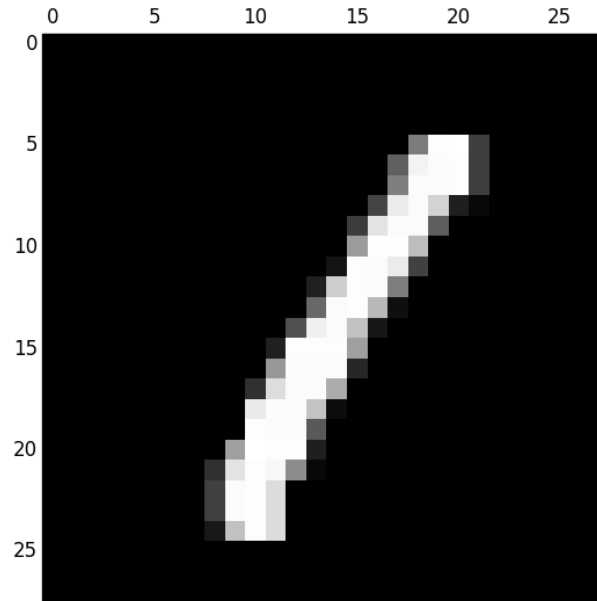
- Google Colaboratory

<https://colab.research.google.com/>

- Gmailアカウントがあれば誰でも使えます。

この授業の目標

- 画像に0～9のどの数字が書いてあるかを判定するコードを書く。
 - 右の画像は「1」
- 機械学習を用いる



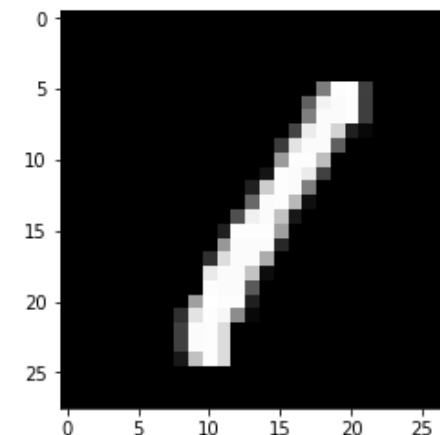
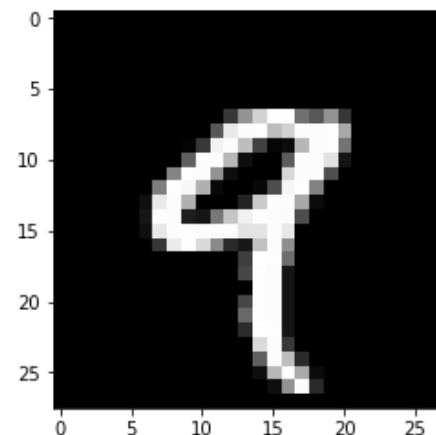
全体の講義内容

- 今日は機械学習のイントロダクション
 - 線形回帰の話(後の回でまた戻ってきます)
- 機械学習の道具に関する講義
 - Python入門
 - NumPy, matplotlibなどのライブラリ
- 機械学習の解説(scikit-learnを利用)

機械学習とは？

例：手書き数字画像の分類 (MNISTデータセット)

- 画像に0～9のどの数字が書いてあるかを計算機に判定させる
 - 右下の画像はそれぞれ「9」「1」が正解。
- 機械学習＝計算機に学習させる
 - どうやって？



label = 5



label = 0



label = 4



label = 1



label = 9



label = 2



label = 1



label = 3



label = 1



label = 4



label = 3



label = 5



label = 3



label = 6



label = 1



label = 7



label = 2



label = 8



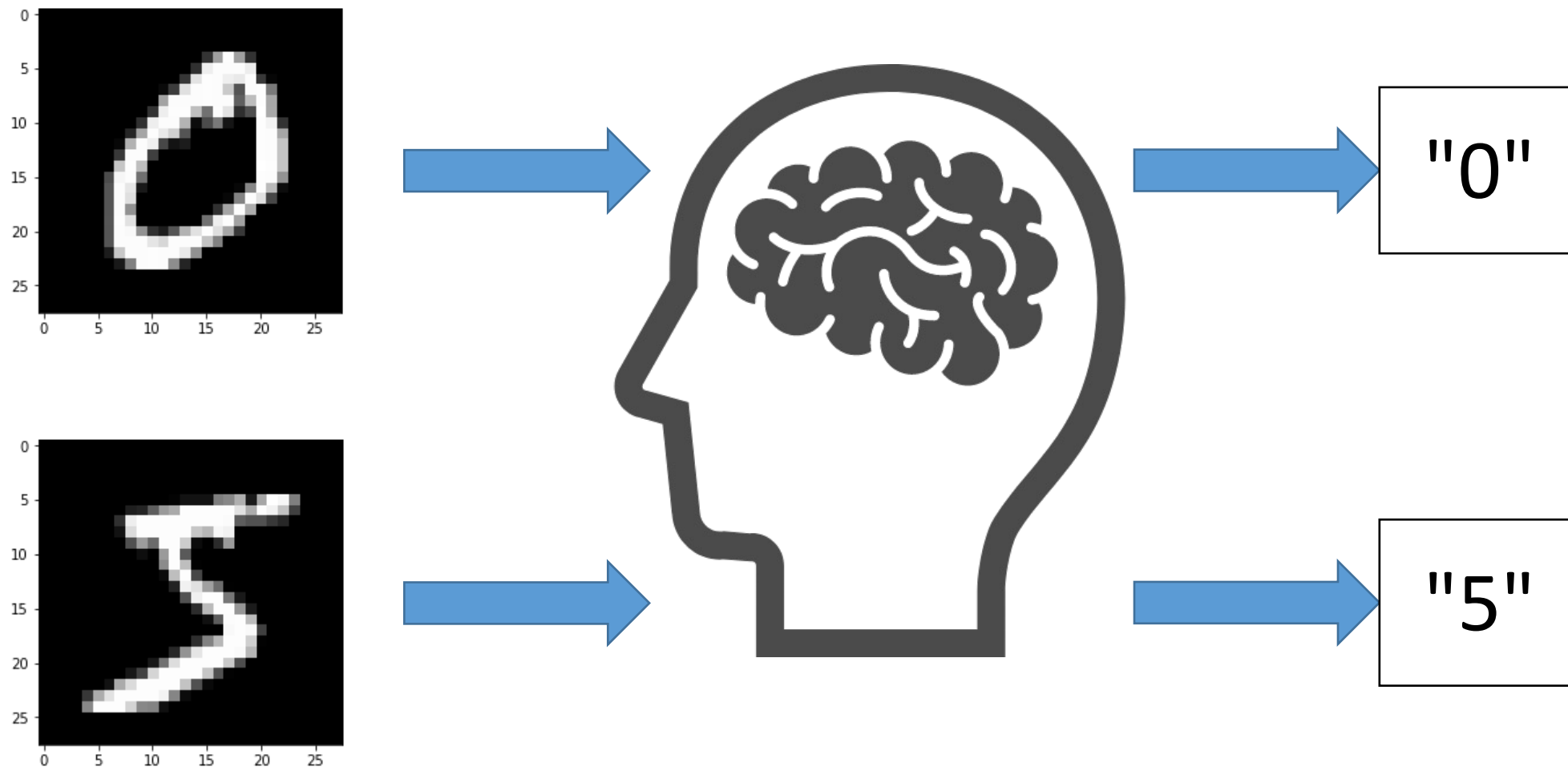
label = 6



label = 9



人間はすぐ答えが分かる (すでに学習済みなので)



計算機に答えを当てさせるにはどうすればいいか？



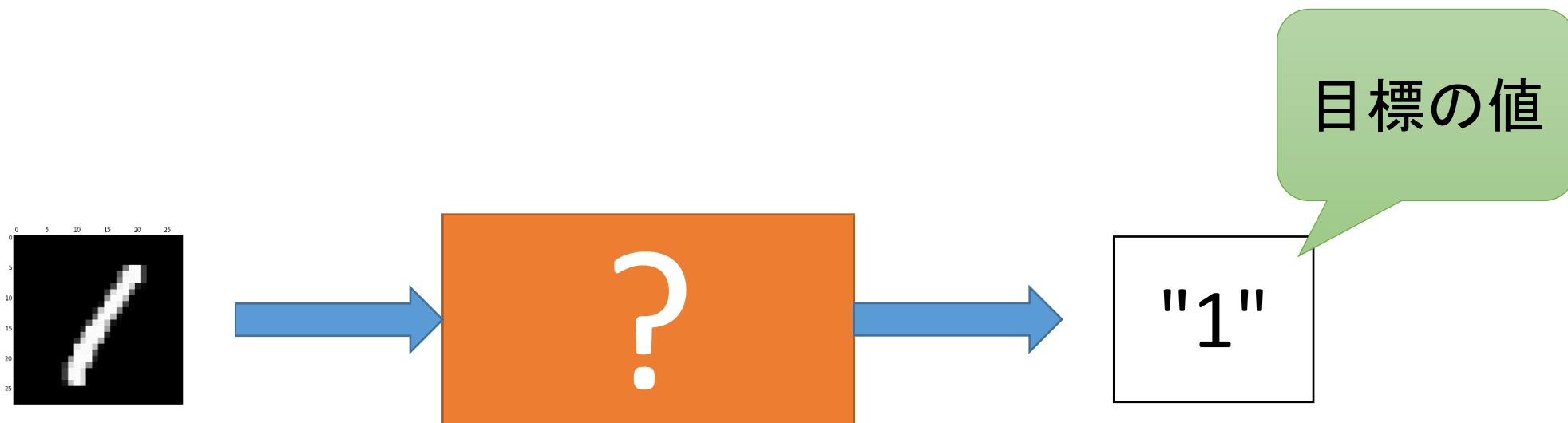
分類(classification)問題

- データを複数のグループへと自動的に分ける問題
 - 入力: 分類したいデータ
 - 出力: どのグループへ分類すべきかを示す値 = 「ラベル」
- どんな入力データにも正解のラベルを出して欲しい
 - 欲しいものは図の「？」の部分。



機械学習とは何か

- 学習: 出てくる値が目標の値になる箱の中身(「?」の部分)を見つける
- 箱の中身: 入力から出力を得るための何らかの計算方法
- 機械学習: 箱の中身を計算機に見つけさせる



機械学習とは

良い関数を計算機に見つけてもらうこと。

関数＝適当な値を入れると、その値に応じて何かの値が出てくる箱
良い関数＝どんな値を入れても、出てくる値が目標の値

計算機に見つけてもらうのであって、
人間が試行錯誤して見つけるのではない。

もう少しテクニカルに言うと・・・

$$y=ax^2 + bx + c$$

関数の、無数にあるパラメータ設定の中から
良い設定を計算機で見つけること。

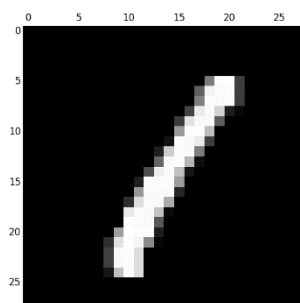
良い設定＝どんな入力に対しても（見たことがない入力に対しても）
望みどおりの出力が得られる

関数を選ぶ範囲

- 関数を選ぶ範囲は、あらかじめ決めておく
- どう決める？
 - 入力データのフォーマットを決める
 - 出力データのフォーマットを決める
 - 計算式の「かたち」を決める
 - 計算式はパラメータを含む(このパラメータを計算機で決めるのが機械学習)

- ベクトルにする

⇒ $28 \times 28 = 784$ 次元ベクトル



画像を画像のまま扱う方法は
今日は割愛します。

16

例) 出力記号をどうやって数値化する？

- "0", "1", "2", ... はラベルであって数値ではない
- ベクトルにする

例) 10種類のラベルがある場合 ("0", "1", "2", ..., "9" の10種類)

⇒ 10次元ベクトルとして数値化

- "3" というラベルの場合:

[0 0 0 1 0 0 0 0 0 0]

こういうものを
one-hot vectorと呼ぶ。

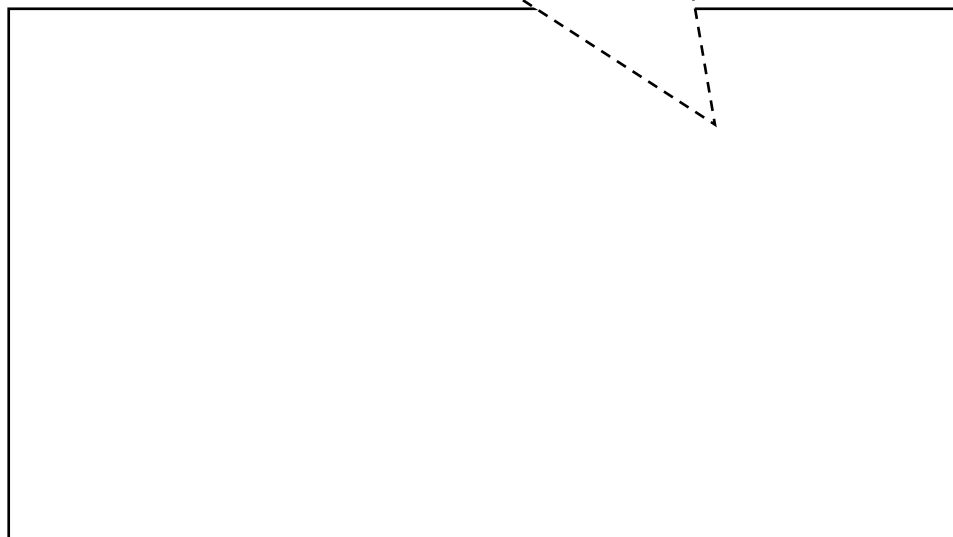
手書き数字画像の分類

入出力データをベクトル化した後の状態



(鋭い方ならこう思いつかれるかも・・・)

「こういう行列を見つければいいのでは？」



■

0
0
...
0
124
253
...
64
251
...
0
0

\equiv

0
1
0
...
0

「で・・・どうやって見つけるの？」

ちょっと複雑すぎるので・・・

- 入力も出力も1個の数値である場合を考える
 - 1個の数値の入力を
 - 1個の数値の出力に
- こういう変換のなかだけから選ぶ

単回帰 (入力も出力も1次元)

問題1



- ある箱に、
 - 1という数値を入れたら3という数値が出てきてほしい。
 - 2という数値を入れたら4という数値が出てきてほしい。
 - 3という数値を入れたら5という数値が出てきてほしい。
 - 4という数値を入れたら6という数値が出てきてほしい。
- 箱の中でどういう計算をすればいいのでしょうか？

問題2



- ある箱に、
 - 1という数値を入れたら3という数値が出てきてほしい。
 - 2という数値を入れたら8という数値が出てきてほしい。
 - 3という数値を入れたら13という数値が出てきてほしい。
 - 4という数値を入れたら18という数値が出てきてほしい。
- 箱の中でどういう計算をすればいいのでしょうか？

問題3



- ある箱に、
 - 1という数値を入れたら2という数値が出てきてほしい。
 - 2という数値を入れたら-1という数値が出てきてほしい。
 - 3という数値を入れたら-4という数値が出てきてほしい。
 - 4という数値を入れたら-7という数値が出てきてほしい。
- 箱の中でどういう計算をすればいいのでしょうか？

問題4(出力が0か1かの二値)

- ある箱に、
 - 1という数値を入れたら0という数値が出てきてほしい。
 - 2という数値を入れたら0という数値が出てきてほしい。
 - 7という数値を入れたら1という数値が出てきてほしい。
 - 8という数値を入れたら1という数値が出てきてほしい。
- 箱の中でどういう計算をすればいいのでしょうか？
 - こういうタイプの問題は、またいずれ。

箱を関数だと思う

- 「 x を入れたら y が出てきてほしい」
 - $y = f(x)$ と書ける
 - $f(x)$ は x の関数(関数: 行き先がひとつに決まる)
- 問題を解くことで何をしていたか？
 - 関数 $f(x)$ の式を求めている

問題5(ちょっとデータ数が多い)

- ある箱に、
 - 2.0という数値を入れたら-4.0という数値が出てきてほしい。
 - 1.0という数値を入れたら-2.0という数値が出てきてほしい。
 - -3.0という数値を入れたら5.0という数値が出てきてほしい。
 - 0.5という数値を入れたら-0.9という数値が出てきてほしい。
 - -4.1という数値を入れたら8.3という数値が出てきてほしい。
 - -1.5という数値を入れたら2.9という数値が出てきてほしい。
 - -2.5という数値を入れたら4.9という数値が出てきてほしい。
 - 6.2という数値を入れたら-12.2という数値が出てきてほしい。
- 箱の中でどういう計算をすればいいのでしょうか？

モデルを設定する

- モデル＝箱の中身を数式で表したもの
- ここでは関数のかたちを一次式に設定（一番簡単なので）

$$f(x) = ax + b$$

- そして「 a と b をいくらにすればいいか？」という問題を解く

問題5の続き

$$2.0a + b = -4.0$$

$$1.0a + b = -2.0$$

$$-3.0a + b = 5.0$$

$$0.5a + b = -0.9$$

$$-4.1a + b = 8.3$$

$$-1.5a + b = 2.9$$

$$-2.5a + b = 4.9$$

$$6.2a + b = -12.2$$

解けない方程式

- 未知数は a と b の二つだけ。
- なのに等式がたくさんある。
 - 式が2つだったら解ける。
- つまり・・・解はない。
- 困った！

問題5で式が二つだけだったら・・・

$$2.0a + b = -4.0$$

$$1.0a + b = -2.0$$

- これは普通の連立一次方程式。

解決法：問題を変える

- 未知数は a と b の二つだけ
 - なのに等式がたくさんある(式が2つだったら解ける)
 - つまり・・・解はない
- そこで・・・値がズレてもいいことにする
 - 誤差(残差)を許す

別の解決法：関数の次数を上げればいいのか？

- 確かにそのとおり
- ただ、この解決法がいいとは限らない
- この点についてはまた後日
 - いわゆる「過学習 overfitting」の問題

問題5の続き (簡単のために式を3つにした。)

$$2.0a + b \approx -4.0$$

$$1.0a + b \approx -2.0$$

$$-3.0a + b \approx 5.0$$

完全に一致しなくてもいい、
という意味。

誤差 (残差ともいう)

- $2.0a + b \approx -4.0$
 - 誤差は $-4.0 - (2.0a + b)$
- $1.0a + b \approx -2.0$
 - 誤差は $-2.0 - (1.0a + b)$
- $-3.0a + b \approx 5.0$
 - 誤差は $5.0 - (-3.0a + b)$

誤差の2乗

- $2.0a + b \approx -4.0$
 - 誤差の2乗は $\{-4.0 - (2.0a + b)\}^2$
- $1.0a + b \approx -2.0$
 - 誤差の2乗は $\{-2.0 - (1.0a + b)\}^2$
- $-3.0a + b \approx 5.0$
 - 誤差の2乗は $\{5.0 - (-3.0a + b)\}^2$

問題を解く方針

- 誤差の2乗の和を最小にすることで

$$f(x) = ax + b$$

の a と b (モデルのパラメータ)を求める

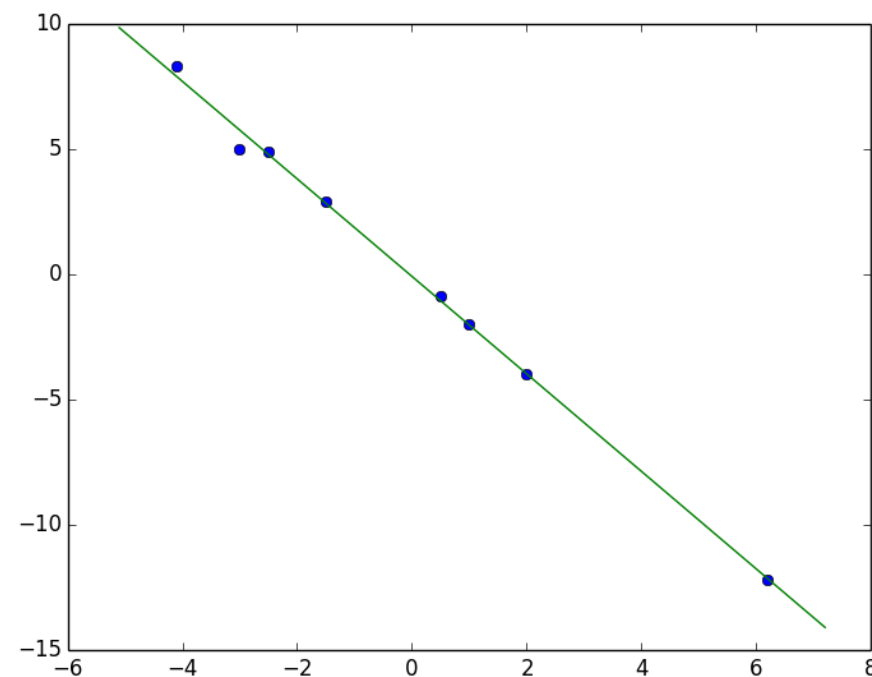
- いまの問題の場合、次の式の値を最小にする a と b を求める

$$\{-4 - (2a + b)\}^2 + \{-2 - (a + b)\}^2 + \{5 - (-3a + b)\}^2$$

- この誤差を「2乗和誤差」と呼ぶ

問題の解き方のイメージ

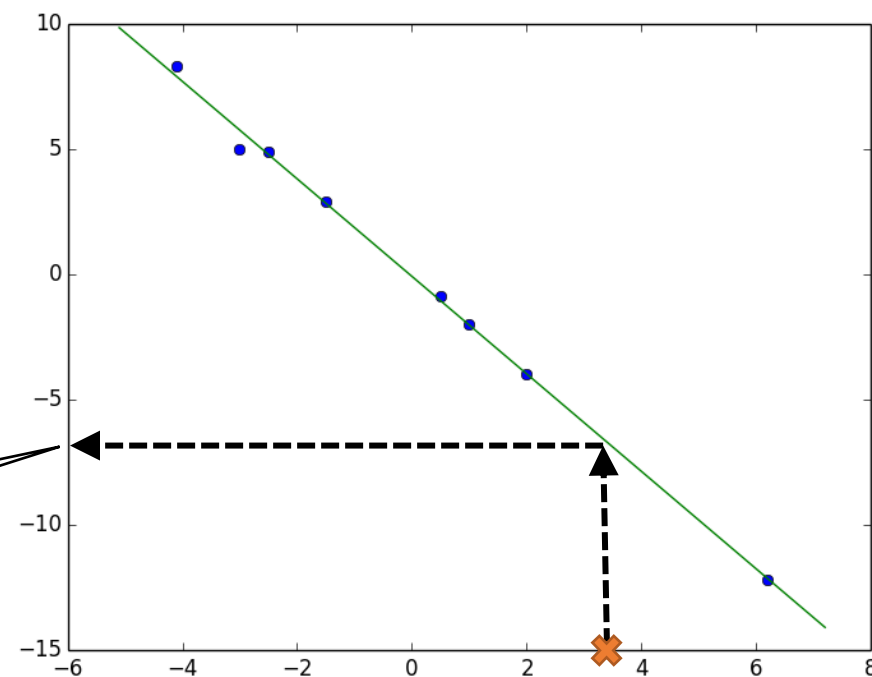
- 入力値と出力値のペアを表す点がたくさんある
 - 入力値が x 座標、出力値が y 座標。
- それらの点にぴったり合う直線を引く
- こういう問題を「線形回帰」という



線形回帰による予測

- 直線が求まれば...
- 任意の入力値について出力値を予測可
- 機械学習は予測に使える

未知の入力値について
予測された出力値



機械学習とは・・・

- 入力値と出力値のペアが大量に与えられているとき・・・
- 入力値から出力値を計算する方法(関数)を計算機に学習させる
 - 関数は特定の形式をしていると仮定(例: 一次式)
 - 出力値からのズレ具合の測り方を決める(例: 誤差の二乗の和)
 - そのズレを機械に小さくさせる＝機械に関数のパラメータを推定させる
(例: 一次式の係数を、誤差の二乗和ができるだけ小さくなるように決める)
- 正確に言えばこれは「教師あり」学習(「教師なし」学習はまた別の話。)

逆に言えば・・・

「”良い関数を見つける”という問題へ落としこめない問題は
いくら頑張っても機械学習では解けません」

- 解きたい問題を、「良い関数を見つける」問題として、どうにかして、
言い換えてみてください

課題1(義務ではない)

$$\{-4 - (2a + b)\}^2 + \{-2 - (a + b)\}^2 + \{5 - (-3a + b)\}^2$$

上の2乗和誤差を最小にする a と b の値を求めよ

- 電子ファイルでCanvas LMSに提出してください。
 - 手書き計算をスマホで撮った画像、Word、LaTeXなど、OK。