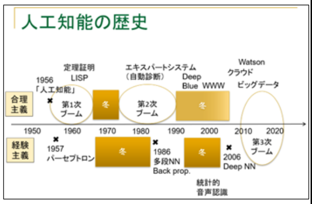
知的情報処理論
（第2回）

2023年4月18日（火）
産業科学研究所
駒谷 和範

第2回第3回の目標

- パーセプトロンの学習を例として，「学習とは何か」を具体的に体感
  - 最も単純なモデル
  - この組み合わせが深層学習 (deep learning)



- 「〇〇学習」のいくつかを整理

機械学習とパターン認識

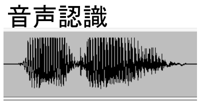
- 以下の4分野の中身はかなり共通
  - 機械学習
  - 統計分析
  - パターン認識
  - データマイニング

機械学習・統計分析：「手段」に注目した名前
パターン認識・データマイニング：「対象」「結果」に注目した名前

- ○「パターン認識をするのに機械学習を使う」

パターン認識の例

- 音声認識



→ 「大阪」

- メールのスパム判定

> Hello. I am writing to you on behalf of a famous executive search agency. I write to you already for the second time. Perhaps, you did not receive our information or our offer did not interest you. ...

→ スパム or スパムでない

- 顔画像認証 → その人のID

機械学習のタスク

- 分類 (classification)
  - 正解がラベル（クラス）で与えられる
    E.g. 入力画像に対して、「犬」「猫」. . .
    入力音声波形に対して「名古屋」「大阪」. . .
- 回帰 (regression)
  - 正解が連続値で与えられる
    E.g., x-y平面上の点の集合に対して、回帰関数を求める

学習データの前提による分類

- 教師あり学習 (supervised learning) ― 正解yの付与（アノテーション）が大変
  - データ x とそれに対する正解 y の組の集合
  - *直感的には* $y = f(x)$ となる写像 $f()$ の推定
- 教師なし学習 (unsupervised learning)
  - データ x の集合のみが与えられる
    - モデル推定：x を生じさせるクラスや関数を推定
    - クラスタリング，異常検知，パターンマイニング
- 強化学習 (reinforcement learning)
  - データは逐次的に入力され、報酬が遅れて与えられる
  - 報酬が最大となるように，システムの行動を選択

教師あり学習（データと正解の組）

https://www.amazon.co.jp/dp/4593594200

教師なし学習

仲間はずれはどれ？

くし

車

肉

発破

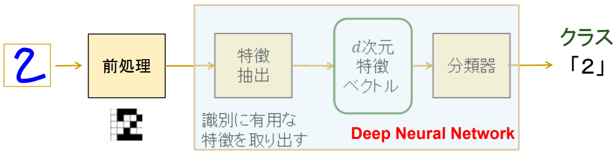
他にもある〇〇学習

学習の「させ方」
- データ $\{x\}$ は比較的簡単に集まるが，それに正解 y を人手で付与するのが大変

- 半教師あり学習 (semi-supervised learning)
  - データの一部に対してのみ正解が付与されている
  - 教師ありデータで学習した分類器を，教師なしデータに適用してその出力ラベルを正解とみなす
- 能動学習 (active learning)
  - どういう「順番」で正解を与えると，効率よく性能が向上するか
- 転移学習 (transfer learning)
  - ある対象で学習した機械学習モデルを、他の対象に対してうまく使う学習のさせ方
- アンサンブル学習（ensemble learning）
  - 複数の学習器を作り，その結果を統合する手法
- 自己教師あり学習 (self-supervised learning)
  - データの一部を「目隠し」し，それを正解とみなして学習
- ・・・

パターン認識の中身

パターン認識を構成するモジュール



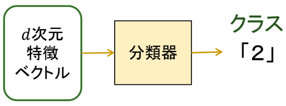
- アナログデータをデジタル情報に変換
- 正規化

手書き数字認識の特徴（例）

縦横比
上半分のピクセルの比率
右半分のピクセルの比率
一筆で書かれた線分の本数
イメージの中心からの平均距離
Y軸を中心とした対称性
X軸を中心とした対称性

- 機械学習アルゴリズムの対象は上記「分類」部
- その前の処理には対象分野の専門知識や経験が必要

パターン認識の定式化



- 入力：d次元ベクトル $\boldsymbol{x} = (x_1, \dots, x_d)$
- 出力：c個のクラスのいずれか $C_i \quad i \in \{1, \dots, c\}$

- 学習データ $\chi = \{(\boldsymbol{x}_1, \widehat{C_1}), \dots, (\boldsymbol{x}_n, \widehat{C_n}), \dots\}$
  - 入力データと正解の組の集合
  - nは学習データの一要素（n番目）を表す添え字
  - $\widehat{C_1}, \dots, \widehat{C_n}, \dots \in \{C_i\}$

Mini Quiz #1

- 前ページ「パターン認識の定式化」で述べているのは
  - 分類問題？回帰問題？
  - 教師あり学習？教師なし学習？強化学習？

  - もしくはどちらでもない？

エクセルデータで言うと

学習データχ のn番目 $n = 854$

1	intell	E1C_1	E1C_2	E1C_3	E1W_1	E1W_2	E1W_3	E0_1	E0_2	E0_3	IC_1	IC_2	IC_
850	NO	0	0	1	0	0	0	0	0	0	0	0	
851	NO	0	0	0	0	0	0	0	1	0	1	0	
852	YES	0	0	0	1	1	0	0	0	1	0	0	
853	NO	1	0	0	0	0	1	0	0	0	0	0	
854	NO	0	0	0	0	1	0	1	0	1	0	0	
855	YES	0	1	0	1	0	0	0	0	0	0	0	
856	YES	1	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	
[illegible]	[illegible]	[illegible]	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	0	0	1	
[illegible]	[illegible]	[illegible]	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	0	0	1	
859	NO	0	0	0	1	0	1	0	0	0	0	0	
860	NO	0	0	0	0	1	0	0	0	1	0	0	
861	NO	0	0	0	0	0	1	0	0	0	0	1	
862	NO	0	0	0	0	0	1	1	0	0	0	1	
863	NO	0	0	0	1	1	0	0	0	0	0	0	
864	YES	1	0	0	0	0	0	0	1	0	0	0	
865	NO	0	0	0	0	1	0	1	0	1	0	0	
866	YES	1	0	1	0	0	0	0	0	0	0	1	
867	YES	0	0	1	0	0	0	1	1	0	0	0	

$\widehat{C_n} \in$ {YES, NO}
$c = 2$

$\boldsymbol{x}_n = (x_1, \dots, x_d)$
d次元ベクトル

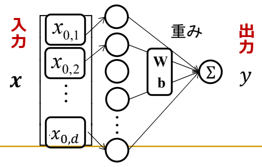
「特徴」ベクトル

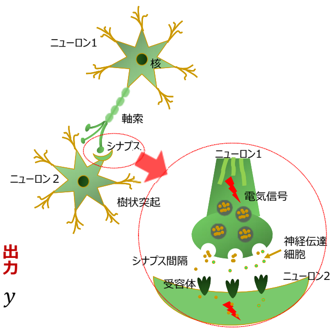
- 由来とする分野により呼び方が異なる
  - 特徴（量），素性，記述子，説明（独立）変数
    ⇒ 全て同じものを指す
- 連続量ではなく離散値（ラベル）でも可
  - 決定木学習など
  - 機械学習パッケージによっては，内部でラベル値と数値を変換して処理している場合がある（変数の型の問題）
    - 例：「1」「-1」をラベルとして処理（大小関係なし）

パーセプトロンによる分類（識別関数）

パーセプトロン

- Neural Network（神経回路網）
- 1段：パーセプトロン [Rosenblatt, 1962]

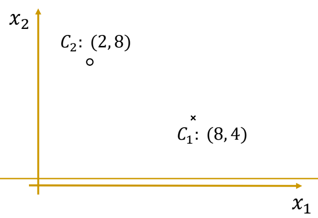




- ニューロンとシナプスの関係を模倣

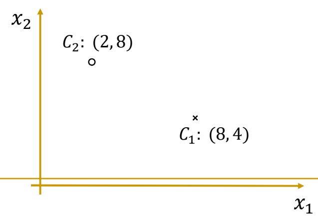
最近傍法（Nearest Neighbor法）による分類

- 各クラス i の代表点 $\boldsymbol{p}_i$ に最も近いクラスに入力 $\boldsymbol{x}$ を分類
  - （$\boldsymbol{x}$に対するクラス）$= argmin_i\ |\boldsymbol{x} - \boldsymbol{p}_i|$



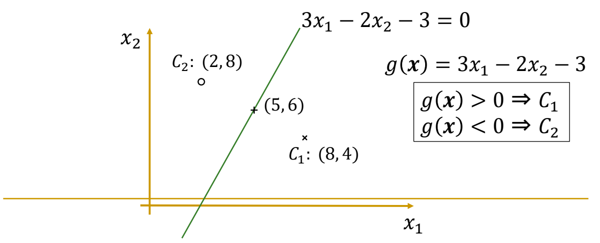
Mini Quiz #2

- 2つのクラスC_1, C_2の代表点が以下のように与えられたとき、最近傍法による境界線の式を求めよ
  - ヒント：「距離が近い方に分類される」
    → 二つの点から距離が等しい点の集合の式



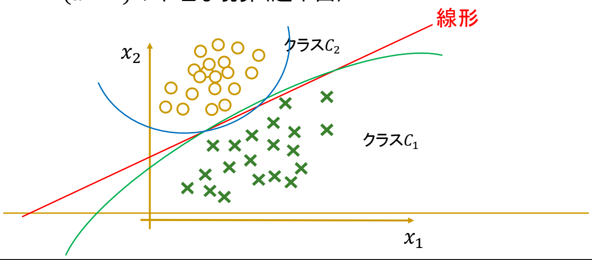
最近傍法（Nearest Neighbor法）による分類

- 各クラスの代表点 $\boldsymbol{p}_i$ に最も近いクラスに入力データを分類
  - （$\boldsymbol{x}$に対するクラス）$= argmin_i\ |\boldsymbol{x}-\boldsymbol{p}_i|$
    つまり $g_i(\boldsymbol{x}) = -|\boldsymbol{x}-\boldsymbol{p}_i|$
  - 2クラス分類問題の場合は $g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$ の正負で分類可能



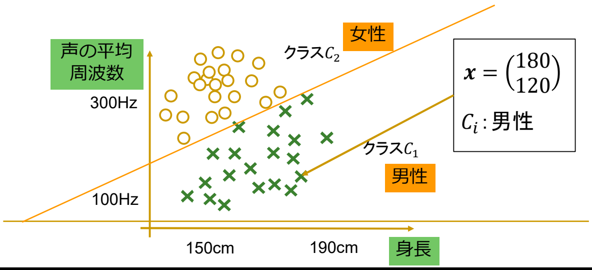
識別超平面

- 2次元の場合，ラベルが C_i となる $\boldsymbol{x}$ の範囲を示すために平面を分割する線
  - d次元の場合，d次元空間を i 個に分割する，次元が$(d-1)$の平坦な境界（超平面）



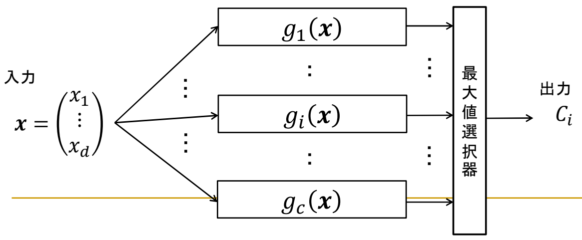
例えば

- 学習データは，データと正解の組の集合
- クラスの境界を求める → 学習
  - 新たなデータ $\boldsymbol{x}$ を分類（＝認識）できる



識別関数 (discriminative function)

- 識別関数（各クラスごと）が最大となるものを出力クラスとする
  - <u>2クラス分類の場合は $g(x) = g_1(x) - g_2(x)$ としてその正負を調べるのと等価</u>



表記の簡略化

- 識別関数が入力ベクトル $\boldsymbol{x}$ に対して線形
  - $g_i(\boldsymbol{x}) = w_{i0} + \sum_{j=1}^{d} w_{ij}x_j$

$$= (w_{i0}\ w_{i1} \ldots\ w_{id})\begin{pmatrix}1\\x_1\\ \vdots \\ x_d\end{pmatrix} \quad \leftarrow x_0 = 1$$

$$= \boldsymbol{w}_i^t\ \boldsymbol{x} \quad （常に x_0 = 1）$$

    - $\square^t$ は転置
  - $\boldsymbol{x}, \boldsymbol{w}$はともに$(d+1)$次元

表記の簡略化

- 識別関数が入力ベクトル $\boldsymbol{x}$ に対して線形
  - $g_i(\boldsymbol{x}) = w_{i0} + \sum_{j=1}^{d} w_{ij}x_j$

$$\begin{pmatrix}g_1(x)\\ \cdots \\ g_i(x) \\ \cdots \\ g_c(x)\end{pmatrix} = \begin{pmatrix}W\\(d+1)\times c\end{pmatrix}\begin{pmatrix}1\\x_1\\ \vdots \\ x_d\end{pmatrix} \quad \leftarrow x_0 = 1$$

$$= \boldsymbol{w}_i^t\ \boldsymbol{x} \quad （常に x_0 = 1）$$

    - $\square^t$ は転置
  - $\boldsymbol{x}, \boldsymbol{w}$はともに$(d+1)$次元

Nearest Neighbor法の識別関数は線形

p_iはクラスC_iの代表点

- 導出
  - （$\boldsymbol{x}$に対するクラス）$= argmin_i\, |\boldsymbol{x} - \boldsymbol{p}_i|$
    - $-|\boldsymbol{x} - \boldsymbol{p}_i|$ を最大とする i を求めたい
  - $argmax_i\, \{-|\boldsymbol{x} - \boldsymbol{p}_i|^2\}$ を考える
    - 二乗しても最大となる i は変わらない
    - $-|\boldsymbol{x}|^2 + 2\boldsymbol{p}_i\boldsymbol{x} - |\boldsymbol{p}_i|^2$
    - i について考えると，$-|\boldsymbol{x}|^2$ は定数
  - つまり，
    - $g_i(x) = \underbrace{\boxed{2\boldsymbol{p}_i}}_{\boldsymbol{w}_i^t}\boldsymbol{x} \underbrace{\boxed{-|\boldsymbol{p}_i|^2}}_{w_{i0}}$ を最大にする i を求めればよい

識別面を決める
$= p_i$を適切に調整する

パーセプトロンの学習

- 「パーセプトロンによる学習」は
  - 教師あり学習
    - 学習データとして、データと正解の組が与えられている
  - 分類問題
    - 出力はラベル C_i のいずれか
    - $\boldsymbol{x}$ の分布の推定ではなく，分類さえできればよい
      ＝ラベルが C_i となる $\boldsymbol{x}$ の範囲がわかればよい

パーセプトロンの学習

- 識別関数の学習

データ点が存在する空間内で，最適なクラス間の境界（識別関数）を求めたい

  - 学習データ $\chi = \{(\boldsymbol{x}_1, \widehat{C_1}), \ldots, (\boldsymbol{x}_p, \widehat{C_p}), \ldots\}$
  - 与えられた学習データを「うまく」分類できるように，重み $\boldsymbol{w}$ をどう適切に調整するか

線形識別関数を求める
＝重みベクトルを適切に決めること

  - $g_i(\boldsymbol{x}) = w_{i0} + \sum_{j=1}^{d} w_{ij}x_j$

$$= \underbrace{\boxed{(w_{i0}\ w_{i1}\ \cdots\ w_{id})}}_{\text{重み}} \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{pmatrix}$$

$$= \boxed{\boldsymbol{w}_i^t}\,\boldsymbol{x} \quad （常に x_0 = 1）$$

2クラスの場合，
$g(x) = g_1(x) - g_2(x)$として
一方のクラスは $g(x)$ が正
他方のクラスは $g(x)$ が負

パーセプトロンの学習規則

1. 重みベクトル $\boldsymbol{w}$ の初期値を適当に設定
2. 学習データ χ の全てについて以下を実行
   - 識別関数 $g(x) = \boldsymbol{w}^t\boldsymbol{x}$ による分類が誤りであった場合，
     $\boldsymbol{w}$ を新しい重みベクトル $\boldsymbol{w}'$ へと更新
     - $\boldsymbol{w}' \leftarrow \boldsymbol{w} + \rho\boldsymbol{x}$（$C_1$のパターンに対して$g(x) \leq 0$となったとき）
     - $\boldsymbol{w}' \leftarrow \boldsymbol{w} - \rho\boldsymbol{x}$（$C_2$のパターンに対して$g(x) > 0$となったとき）

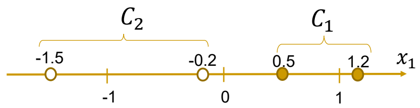
   ただし ρ は学習係数で，正の定数
3. 学習データが全て正しく分類できていたら終了．誤りがあった場合は2に戻る．

パーセプトロンの収束定理

- 学習データが線形分離可能である場合，
  パーセプトロンの学習規則は有限回の
  繰り返しで必ず終了する
  - つまり、データ集合が線形識別関数で分離できる場合であれば、このアルゴリズムで識別平面が必ず求まる

例題：1次元空間での学習

- パーセプトロンの学習規則を用いて，下記の1次元データを分類する識別関数を求めよ．
  - クラスC_1：{0.5, 1.2}
  - クラスC_2：{−1.5, −0.2}



  - この2つのクラスは線形分離可能

回答

- 重みベクトルの初期値 $\boldsymbol{w}^t = (w_0, w_1) = (0.2, 0.3)$とし，学習係数は$\rho = 0.5$とする．
- 初期値に対する識別関数

$$g(\boldsymbol{x}) = \boldsymbol{w}^t\ \boldsymbol{x} = (w_0\ w_1)\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = 0.2 + 0.3x_1$$

∵「表記の簡略化」部分から常に $x_0 = 1$

- 学習規則を適用（1周目）
  - $x_1 = 1.2$：$g(\boldsymbol{x}) = 0.56 > 0$ で判定 C_1 ⇒ 更新なし
  - $x_1 = 0.5$：$g(\boldsymbol{x}) = 0.35 > 0$ で判定 C_1 ⇒ 更新なし
  - $x_1 = -0.2$：$g(\boldsymbol{x}) = 0.14 > 0$ で判定 C_1 **⇒ 要更新**

$$\begin{pmatrix} w_0' \\ w_1' \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0.3 \end{pmatrix} - 0.5\begin{pmatrix} 1 \\ -0.2 \end{pmatrix} = \begin{pmatrix} -0.3 \\ 0.4 \end{pmatrix}$$

この結果，新しい $g(\boldsymbol{x}) = -0.3 + 0.4x_1$

  - $x_1 = -1.5$：$g(\boldsymbol{x}) = -0.9 < 0$ で判定 C_2 ⇒ 更新なし

回答（続き）

- 学習規則を適用（2周目）

$g(\boldsymbol{x}) = -0.3 + 0.4x_1$

  - $x_1 = 1.2$：$g(\boldsymbol{x}) = 0.18 > 0$ で判定 C_1 ⇒ 更新なし
  - $x_1 = 0.5$：$g(\boldsymbol{x}) = -0.1 < 0$ で判定 C_2 **⇒ 要更新**

$$\begin{pmatrix} w_0' \\ w_1' \end{pmatrix} = \begin{pmatrix} -0.3 \\ 0.4 \end{pmatrix} + 0.5\begin{pmatrix} 1 \\ 0.5 \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0.65 \end{pmatrix}$$

この結果，新しい $g(\boldsymbol{x}) = 0.2 + 0.65x_1$

  - $x_1 = -0.2$：$g(\boldsymbol{x}) = 0.07 > 0$ で判定 C_1 **⇒ 要更新**

$$\begin{pmatrix} w_0' \\ w_1' \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0.65 \end{pmatrix} - 0.5\begin{pmatrix} 1 \\ -0.2 \end{pmatrix} = \begin{pmatrix} -0.3 \\ 0.75 \end{pmatrix}$$

この結果，新しい $g(\boldsymbol{x}) = -0.3 + 0.75x_1$

  - $x_1 = -1.5$：$g(\boldsymbol{x}) = -1.425 < 0$ で判定 C_2 ⇒ 更新なし

回答（続き）

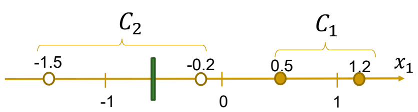
- 学習規則を適用（3周目）

$g(\boldsymbol{x}) = -0.3 + 0.75x_1$

  - $x_1 = 1.2$：$g(\boldsymbol{x}) = 0.6 > 0$ で判定 C_1 ⇒ 更新なし
  - $x_1 = 0.5$：$g(\boldsymbol{x}) = 0.075 > 0$ で判定 C_1 ⇒ 更新なし
  - $x_1 = -0.2$：$g(\boldsymbol{x}) = -0.45 < 0$ で判定 C_2 ⇒ 更新なし
  - $x_1 = -1.5$：$g(\boldsymbol{x}) = -1.425 < 0$ で判定 C_2 ⇒ 更新なし
- 学習データが全て正しく分類できたので終了．

$g(\boldsymbol{x}) = -0.3 + 0.75x_1$

結果の確認

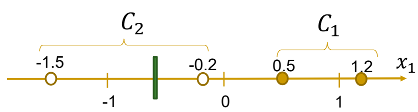


$g(\boldsymbol{x}) = 0$

- 初期値　$(w_0, w_1) = (0.2, 0.3)$
  - $g(\boldsymbol{x}) = 0.2 + 0.3x_1$

$x_1 = -0.67$

結果の確認



$g(\boldsymbol{x}) = 0$

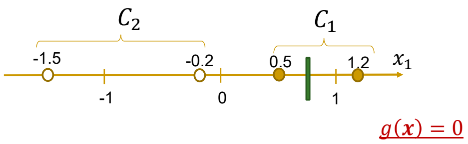
- 初期値　$(w_0, w_1) = (0.2, 0.3)$
  - $g(\boldsymbol{x}) = 0.2 + 0.3x_1$

$x_1 = -0.67$

- 重み更新（1回目）$(w_0, w_1) = (-0.3, 0.4)$
  - $g(\boldsymbol{x}) = -0.3 + 0.4x_1$

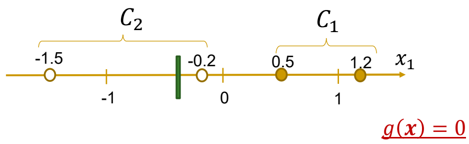
$x_1 = 0.75$

結果の確認



- 初期値 $(w_0, w_1) = (0.2, 0.3)$ $x_1 = -0.67$
  - $g(\boldsymbol{x}) = 0.2 + 0.3x_1$
- 重み更新(1回目)$(w_0, w_1) = (-0.3, 0.4)$ $x_1 = 0.75$
  - $g(\boldsymbol{x}) = -0.3 + 0.4x_1$
- 重み更新(2回目)$(w_0, w_1) = (0.2, 0.65)$ $x_1 = -0.31$
  - $g(\boldsymbol{x}) = 0.2 + 0.65x_1$

結果の確認



- 初期値 $(w_0, w_1) = (0.2, 0.3)$ $x_1 = -0.67$
  - $g(\boldsymbol{x}) = 0.2 + 0.3x_1$
- 重み更新(1回目)$(w_0, w_1) = (-0.3, 0.4)$ $x_1 = 0.75$
  - $g(\boldsymbol{x}) = -0.3 + 0.4x_1$
- 重み更新(2回目)$(w_0, w_1) = (0.2, 0.65)$ $x_1 = -0.31$
  - $g(\boldsymbol{x}) = 0.2 + 0.65x_1$
- 重み更新(3回目)$(w_0, w_1) = (-0.3, 0.75)$ $x_1 = 0.4$
  - $g(\boldsymbol{x}) = -0.3 + 0.75x_1$

この例題の位置づけ

	この例題	AlexNet [Krizhevsky 2012]
特徴量次元	1 $(= d)$	150,528 (= 224 x 224 x 3)
クラス数	2 $(= c)$	1,000
モデル	線形識別関数	8層NeuralNet (5層CNN, 3層全結合)
学習すべきパラメータ数	2 $= (d+1)$	60 millions
学習データ数	4	1.2 millions

- 特徴量次元の増加やモデルの複雑化により，モデルが持つ学習すべきパラメータ数は格段に増える
  - その分学習データが必要（方程式が不定になるイメージ）
- 線形識別関数の場合でも，c クラス分類$(c > 2)$だと $c \times (d+1)$ 個

[Krizhevsky 2012] https://www.cs.toronto.edu/~kriz/imagenet_classification_with_deep_convolutional.pdf

参考書

1. C.M. ビショップ著，元田浩，他訳: “パターン認識と機械学習（上・下）”, 丸善出版, 2012.
2. 石井健一郎，他: “わかりやすいパターン認識”, オーム社, 1998.
3. 荒木雅弘: “フリーソフトでつくる音声認識システム”, 森北出版, 2007.

下ほど平易