RITSUMEIKAN

機械学習 第４回 識別（２）

立命館大学 情報理工学部

村上 陽平

Beyond Borders

講義スケジュール

- 担当教員１：村上、福森（第1回～第15回）

1	機械学習とは、機械学習の分類		9	深層学習
2	機械学習の基本的な手順		10	アンサンブル学習
3	識別（１）		11	モデル推定
4	**識別（２）**		12	パターンマイニング
5	識別（３）		13	系列データの識別
6	回帰		14	強化学習
7	サポートベクトルマシン		15	半教師あり学習
8	ニューラルネットワーク			

- 担当教員２：叶昕辰先生（第16回の講義を担当）

今回の講義内容

- 取り扱う問題の定義
- 最小二乗法
- 最急降下法（さいきゅうこうかほう）
- Widrow-Hoffの学習規則
- 確率的（かくりつてき）最急降下法
- 演習問題

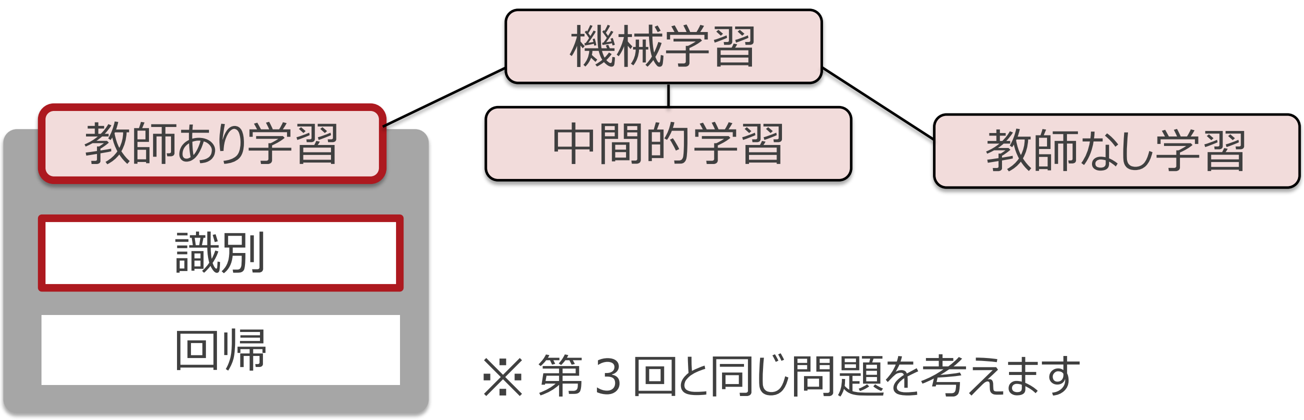
取り扱う問題の定義 ： 教師あり・識別問題

- カテゴリデータ、または数値データからなる特徴ベクトルを入力して、それをクラス分けする識別器を作る

※ 教師あり学習の識別問題での学習データは、以下のペアで構成される

入力データの特徴ベクトル（カテゴリデータ/数値データ） ← $\{x_i, y_i\}, \quad i = 1,2,\dots,N$ → 学習データの総数

y_i：**カテゴリ形式の正解情報 →「クラス」と呼ぶ**

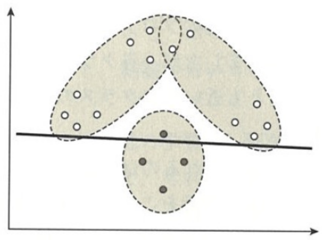


※ 第３回と同じ問題を考えます

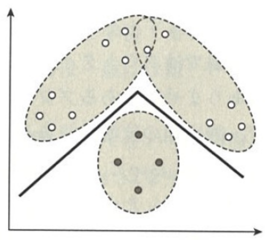
区分的線形識別

◘ パーセプトロンの学習規則

- 特徴空間上の学習データが線形分離可能ならば識別面を発見できる
  - 線形分離不可能な場合（1つの超平面で区切れない場合）は？
    - 超平面：1次元低い部分空間
    - 識別面を折り曲げれば学習データを分離できる
- 区分的線形
  - 折れ曲がっている部分だけが **非線形** で、それ以外の区間は **線形**



線形分離不可能なデータ

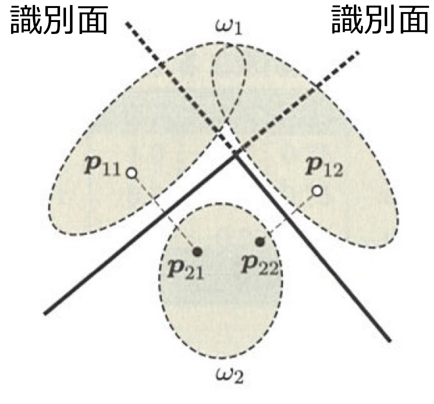


区分的線形識別を用いた場合

区分的線形識別面の定式化

◘ 区分的線形識別面の実現

- 2つの超平面（2次元の場合は2本の直線）を繋ぐ
  - 1つのクラスに対して、2個のプロトタイプを用意すれば2つの線形識別面ができる
    - 右図の場合
      - » クラスω_1のプロトタイプ：$\boldsymbol{p}_{11}$、$\boldsymbol{p}_{12}$
      - » クラスω_2のプロトタイプ：$\boldsymbol{p}_{21}$、$\boldsymbol{p}_{22}$
      - » 1つ目の識別面：$\boldsymbol{p}_{11}$、$\boldsymbol{p}_{21}$から求める
      - » 2つ目の識別面：$\boldsymbol{p}_{12}$、$\boldsymbol{p}_{22}$から求める
    - 2つの超平面（識別面）の内、クラスの識別に関係する面を繋いで区分的線形識別面（右図の実線）を実現



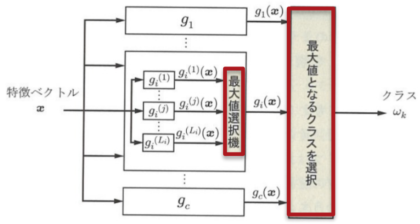
区分的線形識別面

区分的線形識別面の定式化

区分的線形識別面の定式化

- クラスω_iについて、L_i個の線形識別面を繋ぎ合わせれば、他のクラスと分離できると仮定
  - クラスω_iには L_i個のプロトタイプが必要

- クラスω_iにおけるL_i個のプロトタイプに対応する**副次(ふくじ)識別関数**を $g_i^{(l)}(\boldsymbol{x})\ (l = 1, \ldots, L_i)$ とする
  - クラスω_iの識別関数 $g_i(\boldsymbol{x})$を L_i個の**副次識別関数の最大値**として表現
  - 各クラスの識別関数 $g_1(\boldsymbol{x}), \ldots, g_c(\boldsymbol{x})$の内(うち)、最大値をとる識別関数が$g_k(\boldsymbol{x})$なら、入力$\boldsymbol{x}$はクラス$\omega_k$に識別される



区分的線形識別関数を用いた識別器

区分的線形識別関数の識別能力と学習

区分的線形識別関数の識別能力

- プロトタイプの数（副次識別関数の数）を増やせば、理論上は、非線形な曲面(きょくめん)を任意(にんい)の精度で近似(きんじ)できる
  - 学習データがどんな複雑な分布でも識別面を決められるが…

区分的線形識別関数の学習は難(むずか)しい

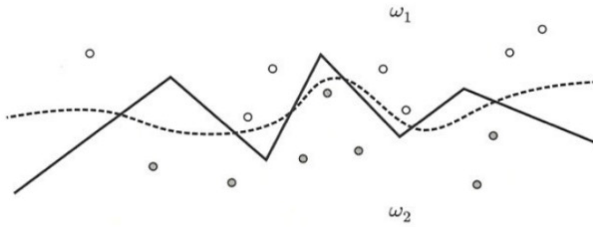
- 副次識別関数の**個数**L_i（何回(なんかい)曲げればクラスを分離できるか？）とそれらの**重み**の両方(りょうほう)を学習しなければならない
  - これらを同時に学習することはできない
  - 副次識別関数の個数を変えると、重みの学習をやり直(なお)す
    - ただし、十分な個数の副次識別関数を用意しないと学習が終了しない

区分的線形識別関数の識別能力と学習

□ 区分線形識別面の学習は難しい（つづき）

- ■ L_iをクラスω_iの学習データの個数とした場合、**最も複雑な識別面**が得られる（下図の実線）
  - 「十分に多い学習データからプロトタイプを選ぶという前提で全ての学習データをプロトタイプした場合」と考えてよい

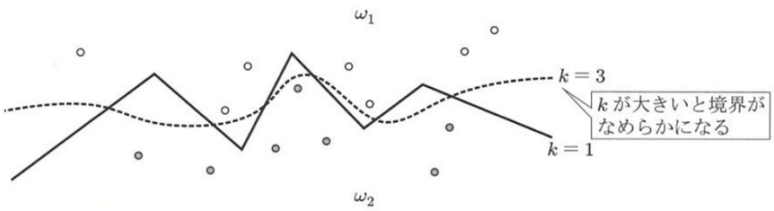
- ■ **クラスの識別面は滑らかな形の方が良さそう**（下図の点線）
  - それを実現するのが**k-NN法**



k-NN法

□ k-NN法（※ NN法：最近傍決定則）

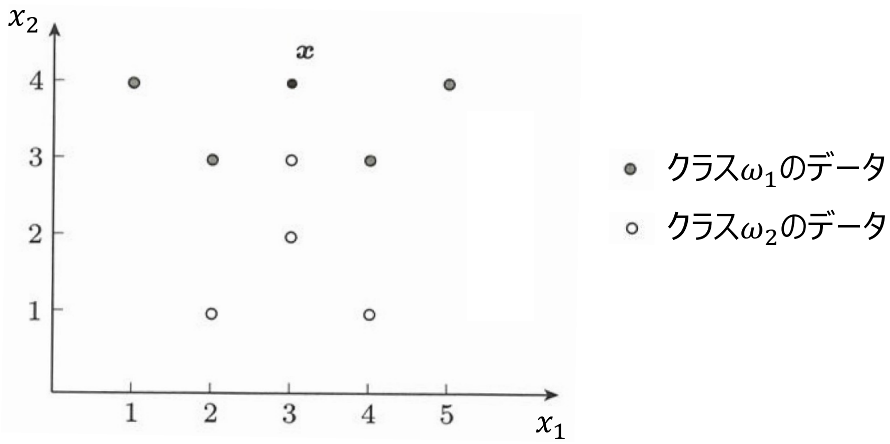
- ■ 入力xに近いk個のデータからクラスを識別する方法
  - 一般にkが大きいほど、識別面は滑らかになる傾向
    - – 1-NN法：入力に一番近いプロトタイプが識別結果
    - – 3-NN法：入力に近い3個のプロトタイプの多数決から識別結果が決まる



- ■ 識別結果の決め方は様々
  - 単純な多数決、順位による重み付き多数決
  - 副次識別関数の値を基準にする方法 など

演習問題4-1（10分間）

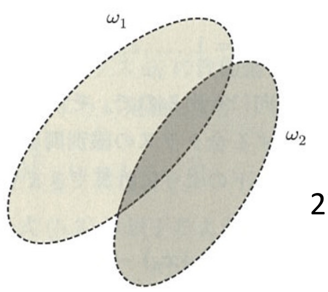
- 下図に示す学習データを用いて、
  3-NN法によって入力$\boldsymbol{x}$ = (3,4)を識別せよ
  - 識別結果の決め方は多数決とする



学習データを分離できない場合

- 先ほどまで、線形分離 or 区分的線形分離可能な学習データを取り扱ってきた
  - 下図のように特徴空間上で複数のクラスの学習データが重(かさ)なり合(あ)って分布している場合
    - 区分的線形関数でも誤識別を無くすことができない
    - 学習データに対(たい)する識別関数の誤差(ごさ)を定義して、
      その誤差を最小にする識別面を見つける方法が必要

→ その代表的な手法が**最小二乗法**



2つのクラスの学習データの分布が重なっている
（学習データを分離できない）

誤差評価に基づく学習：最小二乗法

□ 最小二乗法

- **誤差の2乗和**を**最小**にすることで識別関数を求める方法
- 次のスライドで最小二乗法を説明するために使用する記号
  - $\chi \stackrel{\text{def}}{=} \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$：学習データの集合（※ $A \stackrel{\text{def}}{=} B$： AをBと定義する）
  - $\boldsymbol{x}_p$：集合χから取り出したp番目のデータ
  - c：クラス数
  - $g_i(\boldsymbol{x}_p)$：学習データ$\boldsymbol{x}_p$に対するクラスω_iの識別関数の値
    - $g_i(\boldsymbol{x}_p) = \boldsymbol{w}_i^T \boldsymbol{x}_p$（前回の講義資料より）
  - b_{ip}：望ましい出力値（**教師信号**）
    - $\boldsymbol{x}_p$がクラスω_iに属する場合：$b_{1p}, \dots, b_{cp}$ $\left(b_{ip} = 1, b_{jp} = 0 \ (j \neq i)\right)$
  - ε_{ip}：識別関数の値 $g_i(\boldsymbol{x}_p)$と教師信号 b_{ip}の誤差
    - $\varepsilon_{ip} \stackrel{\text{def}}{=} g_i(\boldsymbol{x}_p) - b_{ip}$ $\quad (i = 1, \dots, c)$

誤差評価に基づく学習：最小二乗法

□ 最小二乗法

- J_p：$\boldsymbol{x}_p$に対する全クラスの識別関数の誤差の二乗和

$$J_p \stackrel{\text{def}}{=} \frac{1}{2}\sum_{i=1}^{c} \varepsilon_{ip}^2 = \frac{1}{2}\sum_{i=1}^{c}\left\{g_i(\boldsymbol{x}_p) - b_{ip}\right\}^2 = \frac{1}{2}\sum_{i=1}^{c}\left(\boldsymbol{w}_i^T \boldsymbol{x}_p - b_{ip}\right)^2$$

  - 上式は特定の学習データ$\boldsymbol{x}_p$に関する識別関数の誤差を評価

- J：最終的に評価すべき誤差
  - 全学習データとの誤差の和

$$J \stackrel{\text{def}}{=} \sum_{p=1}^{n} J_p = \frac{1}{2}\sum_{p=1}^{n}\sum_{i=1}^{c} \varepsilon_{ip}^2 = \frac{1}{2}\sum_{p=1}^{n}\sum_{i=1}^{c}\left(\boldsymbol{w}_i^T \boldsymbol{x}_p - b_{ip}\right)^2$$

最小二乗法は Jを**最小**にするようにクラスiの識別関数の重み$\boldsymbol{w}_i$ $(i = 1, \dots, c)$を**調整**

最小二乗法の解析的な解法

- 誤差Jを最小にする重み$\boldsymbol{w}_i$の解析的(かいせきてき)な解法
  1. 最小値を計算したい関数 J を重み$\boldsymbol{w}_i$で偏微分(へんびぶん)
  2. 1.から極小値(きょくしょうち)を計算し、そのときの$\boldsymbol{w}_i$が誤差を最小とする$\boldsymbol{w}_i$

- 計算しやすくするために、以下の記号を定義
  - $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)^T$ ：パターン行列
  - $\boldsymbol{b}_i$ ：クラスω_iの全ての教師信号を並べたn次元ベクトル
    - $\boldsymbol{b}_i \stackrel{\text{def}}{=} (b_{i1}, \dots, b_{in})^T \ \ (i = 1, \dots, c)$
  - 上記の記号を使って誤差Jを書(か)き換(か)えると…

$$J = \frac{1}{2}\sum_{p=1}^{n}\sum_{i=1}^{c}\left(\boldsymbol{w}_i^T\boldsymbol{x}_p - b_{ip}\right)^2 = \frac{1}{2}\sum_{i=1}^{c}\|\boldsymbol{X}\boldsymbol{w}_i - \boldsymbol{b}_i\|^2$$

最小二乗法の解析的な解法

- 誤差Jを最小にする重み$\boldsymbol{w}_i$の解析的な解法（つづき）
  1. 誤差Jを重み$\boldsymbol{w}_i$で偏微分

$$\frac{\partial J}{\partial \boldsymbol{w}_i} = \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w}_i - \boldsymbol{b}_i) \ \ (i = 1, \dots c)$$

  2. 1.から極小値を計算（上式が0となる$\boldsymbol{w}_i$を計算）

$$\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w}_i - \boldsymbol{b}_i) = 0 \ \ (i = 1, \dots c)$$

$$\Rightarrow \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}_i = \boldsymbol{X}^T\boldsymbol{b}_i$$

$$\Rightarrow \boldsymbol{w}_i = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{b}_i$$

この計算によって誤差Jを最小にする重み$\boldsymbol{w}_i$を解析に求めることができた

ただし、あくまでも全学習データに対して誤差を最小にするだけであり、
線形分離可能な学習データでも適切な識別面を発見できない可能性があることに注意する

最急降下法

□ 最小二乗法の解析的な解法

- ■ データ数が多いと、逆行列演算(ぎゃくぎょうれつえんざん)に多くの時間が必要
- ■ この問題を解消(かいしょう)するのが、**最急降下法**

□ **最急降下法**（パラメータ最適化(さいてきか)手法の1つ）

- ■ ある関数の値が最小値をとるように、その**パラメータを関数の値が減少する方向へ徐々(じょじょ)に変化させる方法**
  - 今回の場合、誤差Jが小さくなる方向へ重み$\boldsymbol{w}$を変化させる

$$\underline{\boldsymbol{w}'} = \underline{\boldsymbol{w}} - \rho \underline{\frac{\partial J}{\partial \boldsymbol{w}}}$$

更新(こうしん)後の重み　　更新前の重み　　誤差Jが小さくなる方向

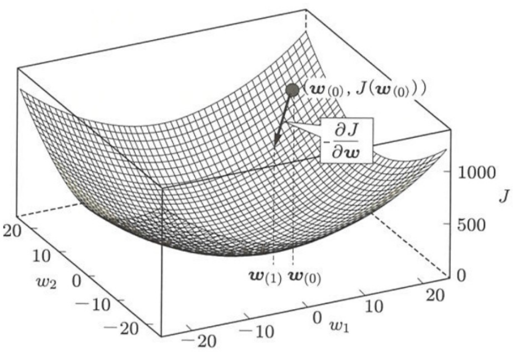
$\boldsymbol{w}$：識別関数の重み
$\boldsymbol{w}'$：更新後の重み
ρ：学習係数

最急降下法

□ 最急降下法のイメージ

- ■ 例：重みを2次元 $\boldsymbol{w} = (w_1, w_2)$ として考える
  - 誤差Jは$\boldsymbol{w}$の**2次式**となるので、図のような**2次曲面**が得られる
  - $\boldsymbol{w}_{(0)}$：重みの初期値
  - $\dfrac{\partial J}{\partial \boldsymbol{w}} = \left(\dfrac{\partial J}{\partial w_1}, \dfrac{\partial J}{\partial w_2}\right)$：**勾配ベクトル**
    - 勾配ベクトルの方向は点$\left(\boldsymbol{w}_{(0)}, J(\boldsymbol{w}_{(0)})\right)$にボールを置(お)いたときに転(ころ)がる方向と**真逆**(まぎゃく)

【重み$\boldsymbol{w}$の更新】
- 点$(\boldsymbol{w}, J(\boldsymbol{w}))$から $-\rho \frac{\partial J}{\partial \boldsymbol{w}}$ だけ $\boldsymbol{w}$を移動させる
- 移動を繰り返すと、修正幅 $\rho \frac{\partial J}{\partial \boldsymbol{w}}$ が小さくなり、やがて$\boldsymbol{w}$は谷底(たにそこ)付近に落(お)ち着(つ)く



$\boldsymbol{w}$を2次元として考えたときの最急降下法のイメージ

Widrow-Hoffの学習規則

- Widrow-Hoffの学習規則
  - 下記の更新式を使った学習アルゴリズム
    - 重みの修正量
      - **全データに対する「学習係数・誤差・学習データの乗算(じょうざん)結果」の合計**

$$\boldsymbol{w}_i' = \boldsymbol{w}_i - \rho \sum_{p=1}^{n} (\boldsymbol{w}_i^T \boldsymbol{x}_p - b_{ip}) \boldsymbol{x}_p$$

左の更新式は①～③を組み合わせて導出(どうしゅつ)可能

【① $\boldsymbol{w} \to \boldsymbol{w}_i$】重み$\boldsymbol{w}$をクラス$\omega_i$の識別関数の重み$\boldsymbol{w}_i$に置換(ちかん)

【② 誤差Jを$\boldsymbol{w}_i$で偏微分】

$$J = \frac{1}{2} \sum_{p=1}^{n} \sum_{i=1}^{c} (\boldsymbol{w}_i^T \boldsymbol{x}_p - b_{ip})^2 \longrightarrow \frac{\partial J}{\partial \boldsymbol{w}_i} = \sum_{p=1}^{n} (\boldsymbol{w}_i^T \boldsymbol{x}_p - b_{ip}) \boldsymbol{x}_p$$

【③ 最急降下法の修正式】$\boldsymbol{w}' = \boldsymbol{w} - \rho \frac{\partial J}{\partial \boldsymbol{w}}$

確率的最急降下法

- **バッチ法**（※ 最急降下法はバッチ法の1つ）
  - 全学習データに対して誤差を求め、一括(いっかつ)で重みを更新
  - 学習データが多いと、1回の重み更新に長(なが)い時間が必要
  - **ミニバッチ法**
    - ある程度、まとまったデータで最急降下法を実行(じっこう)する方法
- 確率的最急降下法
  - 個々のデータ$\boldsymbol{x}_p$に対して、下式で重みを修正する手法
  - データ$\boldsymbol{x}_p$は学習データからランダム（確率的）に選択

最急降下法

$$\boldsymbol{w}_i' = \boldsymbol{w}_i - \rho \sum_{p=1}^{n} (\boldsymbol{w}_i^T \boldsymbol{x}_p - b_{ip}) \boldsymbol{x}_p$$

確率的最急降下法

$$\boldsymbol{w}_i' = \boldsymbol{w}_i - \rho (\boldsymbol{w}_i^T \boldsymbol{x}_p - b_{ip}) \boldsymbol{x}_p$$

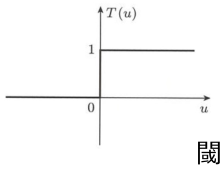
パーセプトロンの学習規則との比較

パーセプトロンの学習規則

Widrow-Hoffの学習規則の特殊（とくしゅ）なケース

- 閾値関数を使って識別関数の出力を0または1に限定（げんてい）

$$\begin{cases} g_i(\boldsymbol{x}_p) = T(\boldsymbol{w}_i^T \boldsymbol{x}_p) = 1 \\ g_j(\boldsymbol{x}_p) = T(\boldsymbol{w}_i^T \boldsymbol{x}_p) = 0 \ \ (j \neq i) \end{cases}$$



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

閾値関数

- 正解のクラスは$\boldsymbol{w}_i^T \boldsymbol{x}_p$が正、
  それ以外のクラスは$\boldsymbol{w}_i^T \boldsymbol{x}_p$が負になるように$\boldsymbol{w}_i$を学習すれば良い
- このときの誤識別のパターン（重みを更新する条件）は2通りのみ

$$\begin{cases} g_i(\boldsymbol{x}_p) = 0,\ b_{ip} = 1 \\ g_j(\boldsymbol{x}_p) = 1,\ b_{jp} = 0 \ (j \neq i) \end{cases}$$

パーセプトロンの学習規則との比較

パーセプトロンの学習規則

Widrow-Hoffの学習規則の特殊なケース（つづき）

- この設定では確率的最急降下法による
  Widrow-Hoffの学習規則は以下のように表現できる

$$\boldsymbol{w}_i' = \boldsymbol{w}_i - \rho\{g_i(\boldsymbol{x}_p) - b_{ip}\}\boldsymbol{x}_p$$

$$\Leftrightarrow \underline{\begin{cases} \boldsymbol{w}_i' = \boldsymbol{w}_i + \rho \boldsymbol{x}_p \ \ (g_i(\boldsymbol{x}_p) = 0,\ b_{ip} = 1\text{のとき}) \\ \boldsymbol{w}_j' = \boldsymbol{w}_j - \rho \boldsymbol{x}_p \ \ (g_j(\boldsymbol{x}_p) = 1,\ b_{jp} = 0\text{のとき}) \end{cases}}$$

パーセプトロンの学習規則の
重みの更新式と同じ

演習問題4-2（10分間）

- 次の文は、それぞれ「パーセプトロンの学習規則」と「Widrow-Hoffの学習規則」のどちらについて説明しているか？
  1. 識別関数の値と教師信号の二乗誤差の総和（そうわ）を最小化する
  2. 線形分離可能の場合でも、全ての学習パターンが正しく識別される重みが得られるとは限らない
  3. 線形分離不可能の場合は、学習は収束しない