知的情報処理論
（第3回）

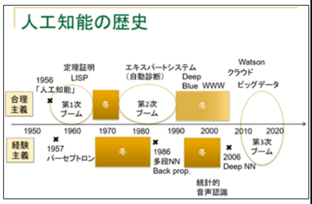
2023年4月25日（火）
産業科学研究所
駒谷 和範

レポート課題（第1回）を出題しました

- CLE上にあります
- 締切：5月15日（月）
  - 考察や説明を，他人が読んでわかる日本語（または英語）で書くこと．
  - プログラムや実行結果のみを送りつけてきた場合は，極めて低く評価する，または受理しない．
  - 本レポートは情報通信工学演習の一部である．知的情報処理論の成績にも加味する．
- レポート
  - 駒谷担当分では2回，武田先生担当分で1〜2回を予定

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)



- 誤差逆伝播 (Back propagation)

線形識別関数

- 各クラス i に対する識別関数 $g_i(\boldsymbol{x})$ が入力ベクトル $\boldsymbol{x}$ に対して線形
  - $g_i(\boldsymbol{x}) = w_{i0} + \sum_{j=1}^{d} w_{ij} x_j$ ← $x_0 = 1$

$$= (w_{i0}\ w_{i1}\ \dots\ w_{id}) \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{pmatrix}$$

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

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

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 $\chi = \{(\boldsymbol{x}_1, \widehat{C_1}), \dots, (\boldsymbol{x}_p, \widehat{C_p}), \dots\}$
  - $\boldsymbol{x}_p$ をクラス C_i に分類 ≡ 識別関数 $g_i(\boldsymbol{x}_p)$ の値が最大
  - 与えられた学習データを「うまく」分類できるように，線形識別関数 $g(\boldsymbol{x}) = \boldsymbol{w}_i^t\,\boldsymbol{x}$ の重み $\boldsymbol{w}$ をどう調整するか
  - $g_i(\boldsymbol{x}) = w_{i0} + \sum_{j=1}^{d} w_{ij} x_j$

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

$$= \boxed{\boldsymbol{w}_i^t}\, \boldsymbol{x}$$

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 $g(\boldsymbol{x}) = \boldsymbol{w}^t\boldsymbol{x}$ 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）

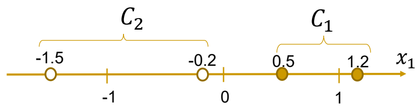
     ただし ρ は学習係数で，正の定数
3. 学習データが全て正しく分類できていたら終了．誤りがあった場合は2に戻る．

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

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．

Mini Quiz #1

- 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．
- クラスC_1とC_2に対する識別関数 $g(\boldsymbol{x})$ の式を示せ．
- 識別境界となる超平面（点）の座標を示せ．
- なぜクラス C_i ごとに $g_i(\boldsymbol{x})$ がない？
  - $argmax_i$ を取るのでは？

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

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

∵「表記の簡単化」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 $C_1 \Rightarrow$ 更新なし
  - $x_1 = 0.5$: $g(\boldsymbol{x}) = 0.35 > 0$ で判定 $C_1 \Rightarrow$ 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 $C_2 \Rightarrow$ 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 $C_1 \Rightarrow$ 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**

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

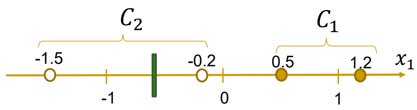
この結果，新しい $g(\boldsymbol{x}) = -0.3 + 0.75x_1$

  - $x_1 = -1.5$: $g(\boldsymbol{x}) = -1.425 < 0$ で判定 $C_2 \Rightarrow$ 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.
  $g(\boldsymbol{x}) = -0.3 + 0.75x_1$

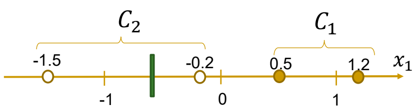
結果の確認



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

- 初期値 $(w_0, w_1) = (0.2, 0.3)$
  - $g(\boldsymbol{x}) = 0.2 + 0.3x_1$ 　 $x_1 = -0.67$

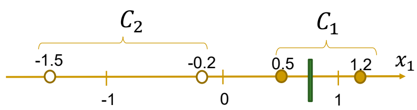
結果の確認



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

- 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）$(w_0, w_1) = (-0.3, 0.4)$
  - $g(\boldsymbol{x}) = -0.3 + 0.4x_1$ 　 $x_1 = 0.75$

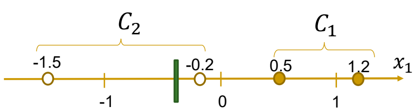
結果の確認



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

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

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

重み空間内での学習の解釈

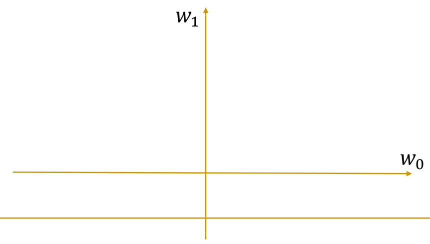
疑問その1

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

- 重みの初期値は適当に設定してよい？
- 学習係数 ρ（正の定数）？
- パーセプトロンの収束定理は本当？

重み空間内での図示

- 先ほどの例題の学習過程を図示
  - 重み空間は $(d+1) = 2$次元
  - ここでは $\boldsymbol{w}$が変数，$\boldsymbol{x}$は定数（学習データは不変）

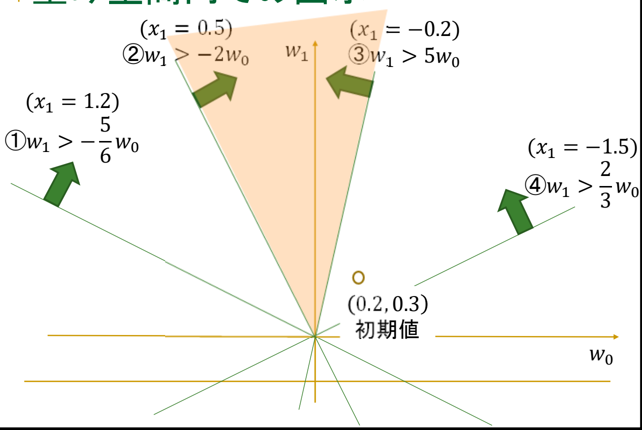


重み空間内での図示

- 学習データ4つから得られる制約
  - クラスC_1: {0.5, 1.2}　　$\Rightarrow g(\boldsymbol{x}) > 0$
  - クラスC_2: {−1.5, −0.2} $\Rightarrow g(\boldsymbol{x}) < 0$
- $g(\boldsymbol{x}) = \boldsymbol{w}^t\boldsymbol{x} = w_0x_0 + w_1x_1$（常に$x_0 = 1$）より，
  - ①　$x_1 = 1.2$　　$\frac{5}{6}w_0 + w_1 > 0$（クラスC_1）
  - ②　$x_1 = 0.5$　　$2w_0 + w_1 > 0$（クラスC_1）
  - ③　$x_1 = -0.2$　　$5w_0 - w_1 < 0$（クラスC_2）
  - ④　$x_1 = -1.5$　　$\frac{2}{3}w_0 - w_1 < 0$（クラスC_2）

これらの領域を重み空間内に図示

重み空間内での図示

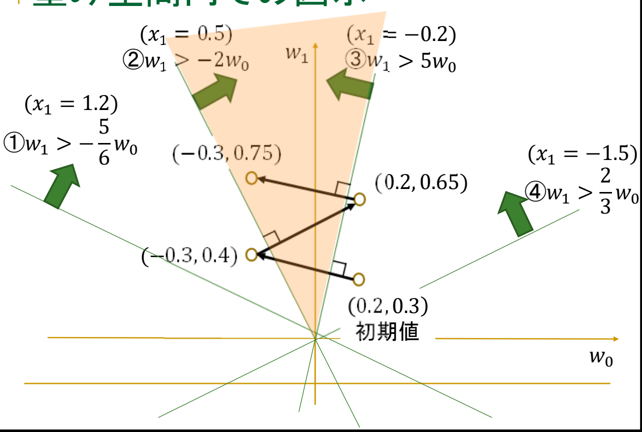


重みの更新式の解釈

$\boldsymbol{w}' = \boldsymbol{w} + \rho\boldsymbol{x}$
（C_1のパターンに対して$g(x) \leq 0$となったとき）

- 幾何学的な解釈：
  - 重みベクトル $\boldsymbol{w}$ を $\boldsymbol{x}$ 方向にその ρ 倍動かす
  - $\boldsymbol{x}$ は，重み空間内の超平面 $\boldsymbol{w}^t\boldsymbol{x} = 0$ に直交する方向（法線ベクトル）
  - つまりこの更新式は，ある学習データ x_n に対して，超平面 $\boldsymbol{w}^t x_n = 0$ を垂直にまたぐ方向に $\boldsymbol{w}$ を更新
    - ただし繰り返し演算なので，必ずしも1回でまたぐ必要はない

重み空間内での図示



Mini Quiz #2

- パーセプトロンの学習において，学習係数 ρ を｛大きく｜小さく｝するとどうなるか考えよ．

ポイント

- 学習係数 ρ の位置づけ
  - 大き過ぎる場合／小さ過ぎる場合
- 重みベクトルの初期値の意味
- パーセプトロンの収束定理のイメージ
- 初期値や学習係数によって，得られる重みベクトルの値は異なる

誤差の最小化（アルゴリズムの拡張）

疑問その２

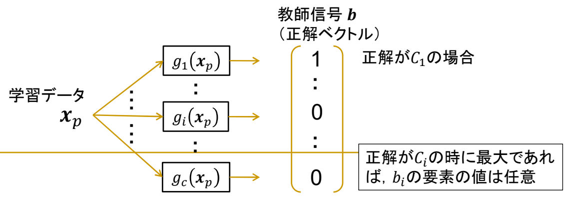
学習データが線形分離可能である場合，
パーセプトロンの学習規則は
有限回の繰り返しで必ず終了する

- 学習データが線形分離可能かどうか事前にわかるの？
  - 線形分離可能な場合しか結果が得られない（アルゴリズムが停止しない）のはとてもマズい
- 「間違っていたら重みを一律更新」
  - 間違いの度合によって、更新量を変えた方がよいのでは？ →「誤差」を考える

誤差の最小化

アルゴリズムの停止条件を変更：
「誤りがなくなるまで」 → 「誤差が（概ね）最小になるまで」

- 誤差 $\varepsilon_{ip} = b_i - g_i(\boldsymbol{x}_p)$
  - b_iは教師信号（正解）
  - $\boldsymbol{x}_p$ の正解がC_iである場合 $b_i = 1$，それ以外 $b_i = 0$　one-hotベクトル
- 学習＝誤差の総和が最小になるように重みを調整



誤差の最小化

- $\boldsymbol{x}_p$ に対する誤差の二乗を使って評価関数 J_p を定義（損失関数）

$$J_p = \frac{1}{2}\sum_{i}^{c} \varepsilon_{ip}^2 = \frac{1}{2}\sum_{i}^{c} (\boldsymbol{w}_i^t \boldsymbol{x}_p - b_i)^2 \quad \cdots 式(*)$$

- 学習データ全体での総和 J

$$J = \sum_{p} J_p = \frac{1}{2}\sum_{p}\sum_{i}^{c} \varepsilon_{ip}^2 = \frac{1}{2}\sum_{p}\sum_{i}^{c} (\boldsymbol{w}_i^t \boldsymbol{x}_p - b_i)^2$$

これが最小となる重みベクトル $\boldsymbol{w}_i$ を求めたい

- 逆行列の計算が必要なため，逐次計算で求める

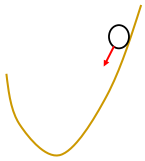
勾配法

- 解析的に求めるのを避け反復計算
  最急降下法 (steepest descent method)
  - p番目の学習データ$\boldsymbol{x}_p$について

$$\boldsymbol{w}_i' = \boldsymbol{w}_i - \rho \boxed{\frac{\partial J_p}{\partial \boldsymbol{w}_i}}$$

J_pが小さくなる方向

ρ は学習係数（正の定数）



- 前ページ式(*)を偏微分

$$\frac{\partial J_p}{\partial \boldsymbol{w}_i} = (\boldsymbol{w}_i^t \boldsymbol{x}_p - b_i)\boldsymbol{x}_p = \varepsilon_{ip}\boldsymbol{x}_p$$

- 代入すると更新式は

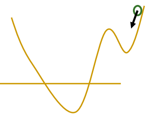
$$\boldsymbol{w}_i' = \boldsymbol{w}_i - \rho\varepsilon_{ip}\boldsymbol{x}_p$$

Widrow-Hoffの学習規則

- 更新式 (Widrow-Hoffの学習規則)

$$\boldsymbol{w}_i' = \boldsymbol{w}_i - \rho\varepsilon_{ip}\boldsymbol{x}_p$$

  - パーセプトロンの学習規則は $\varepsilon_{ip} = 1$ に相当
- 停止条件
  - 誤差が最小となった場合
    - 実際には誤差の減少幅が設定した値を下回った時
      ⇒ 線形分離可能でない場合も停止する
  - 局所最適解に留まっている可能性
    → 初期値や学習係数を変えて何度か実験



大量データの学習時

勾配の計算やパラメータ更新のタイミング

- オンライン学習（確率的最急降下法(SGD)）
  - 1データずつ順次読み込みながら
- バッチ学習
  - 全ての学習データを使用して勾配を計算し更新
- ミニバッチ (mini-batch) 法
  - N個ずつデータを読みこみながら
  - N: バッチサイズ (batch size)

Mini Quiz #3

- 「学習データにおける誤差が最小」となるようにパラメータを定めれば、未知データ（学習データ以外）もうまく分類できるか？

得られる重みの値に関する制約

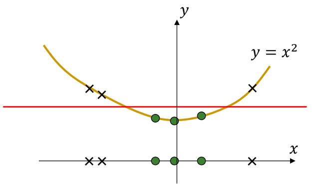
- 得られた重みの値は「最適」なのか
  何をもって「最適」とするか？
  - 識別面が各クラスの点のちょうど間くらいにある方がよい
    →マージン(margin)最大化
    SVM (support vector machine)
  - 重みの値が極端に大きい値を取らない方がよい
    → 正則化項 (regularization term)の導入
    誤差関数に重みの値を含める.
    L1正則化(Lasso), L2正則化(Ridge)
- SVMの特徴
  1. マージン最大化
  2. カーネル(kernel)関数の導入

得られる重みの値に関する制約

- 重みの値が極端に大きい値を取らない方がよい
  → 正則化項 (regularization term)の導入
  損失関数に重みの値を含める → L1正則化(Lasso)
  $\min\{(誤差の総和) + \lambda \sum_{i=1}^{n} |w_i|\}$
- 識別面が各クラスの点のちょうど間くらいにある方がよい
  →マージン(margin)最大化
  - SVM (support vector machine) の特徴
    1. マージン最大化
    2. カーネル(kernel)関数の導入

SVMの特徴

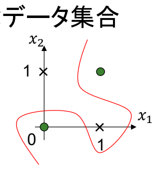
2. カーネル(kernel)関数の導入
   - 非線形な高次元空間に写像することで分類可能とする



多層ニューラルネットワーク（ディープニューラルネットワーク）

パーセプトロンの問題点

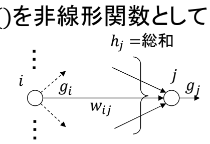
- 線形識別関数なので，線形分離可能なデータ集合しか分類できない
  ＝XORを学習できない (Minsky, 1969)



- 多層ニューラルネットワーク
  - パーセプトロンの出力に非線形の活性化関数を加えて，多層にしたもの
    - 活性化関数：しきい値関数，シグモイド関数，ランプ関数（ReLU），...
  - ここでの「多層」は「中間の隠れ層がある」という意味で、実際には3層程度 (Amari, 1967)

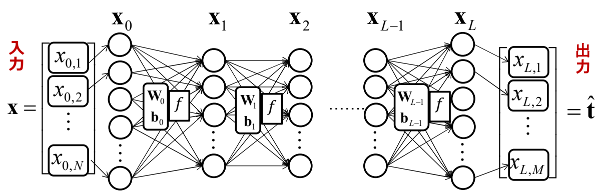
多層ニューラルネットワーク

- 1ユニットの演算
  - 学習パターンx_p
  - 1階層前のi番目のユニットからの信号をg_{ip}，重みをw_{ij}とすると，ある階層のユニットjへの入力h_{jp}は，
    $h_{jp} = \sum_i w_{ij} g_{ip}$
  - ユニットjの出力は，$f()$を非線形関数として
    $g_{jp} = f(h_{jp})$



この図では p は省略

Deep Neural Network (DNN)



Originally written by Dr. R. Takeda

- 層が深いもの：deep neural network (DNN)

誤差逆伝播法（Back Propagation）(1986)

中間の隠れ層にどう教師信号を与える？

- 出力層 l には，教師信号 $\boldsymbol{b}_p$ がある
  - $J_p = \frac{1}{2}\sum_l (\overbrace{g_{lp} - b_{lp}}^{\text{誤差}})^2$

    添字pは，p番目の学習データ

  - $w'_{ij} = w_{ij} - \rho \boxed{\frac{\partial J_p}{\partial w_{ij}}}$ （Widrow-Hoffと同様）
  - $\frac{\partial J_p}{\partial w_{ij}} = \underbrace{\boxed{\frac{\partial J_p}{\partial h_{jp}}}}_{= \varepsilon_{jp}} \overbrace{\boxed{\frac{\partial h_{jp}}{\partial w_{ij}}}}^{= g_{ip}}$

    $= g_{ip}$（前々ページh_{jp}の定義より）

    $= \varepsilon_{jp}$とおく（h_{jp}に関する誤差の変化量）

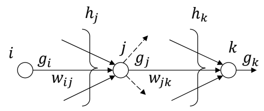
    中間層 j の出力g_{jp}を使って表す

誤差逆伝播法（Back Propagation）(1986)

- $\varepsilon_{jp} = \frac{\partial J_p}{\partial h_{jp}} = \boxed{\frac{\partial J_p}{\partial g_{jp}}}\boxed{\frac{\partial g_{jp}}{\partial h_{jp}}} = f_j'(h_{jp})$（3ページ前$g_{jp} = f(h_{jp})$より）
  具体的には次のページ
  ↑これを層ごとに以下のように置く
- ユニットjが出力層のとき
  - 誤差
    $\frac{\partial J_p}{\partial g_{jp}} = g_{jp} - b_{jp}$
- ユニットjが中間層のとき，一つ先の層kについて
  - $\frac{\partial J_p}{\partial g_{jp}} = \sum_k \frac{\partial J_p}{\partial h_{kp}} \frac{\partial h_{kp}}{\partial g_{jp}}$
    $= \sum_k \varepsilon_{kp} w_{jk}$

ε_{jp}の一部

先の層での誤差ε_{kp}に重みw_{jk}をかけて総和をとり、j層目での誤差ε_{jp}とする



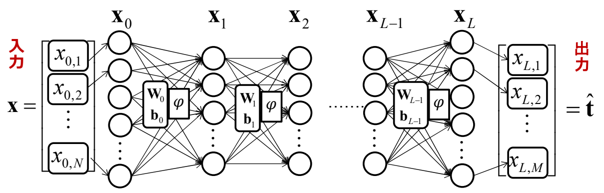
この図では p は省略

誤差逆伝播法（Back Propagation）(1986)

- 前ページ$f_j'(h_{jp})$
  - 微分可能で，しきい値関数と似たふるまいをする関数としてシグモイド関数を導入
    - $S(u) = \frac{1}{1+\exp(-u)}$
    - $S'(u) = S(u)(1 - S(u))$
  - $f_j'(h_{jp}) = g_{jp}(1 - g_{jp})$
- 以上をまとめると，

$$\varepsilon_{jp} = \begin{cases} \boxed{(g_{jp} - b_{jp})}\, g_{jp}(1 - g_{jp}) & \text{（出力層）} \\ \boxed{(\sum_k \varepsilon_{kp} w_{jk})}\, g_{jp}(1 - g_{jp}) & \text{（中間層）} \end{cases}$$

Deep Neural Network (DNN)



Originally by Dr. R. Takeda

- 層が深いもの：deep neural network (DNN)
- 学習すべきパラメータの数が多い
- 事前に設定すべきハイパーパラメータも多い
  - 層の深さL，中間層のノード数，活性化関数 φ
  - 初期値や学習係数にもsensitive

用語の整理

- パラメータ
  学習で求める数値
  - $g(x) = \boldsymbol{w}^t\boldsymbol{x}$ の場合，重みベクトル $\boldsymbol{w}$
- ハイパーパラメータ (hyperparameter)
  学習の前に人手で決めておくもの
  - モデルの「形」を決めるもの
    - 関数 $g(x)$ の形：「線形」，・・・
    - 正則化項の種類，その係数λ
    - DNNの場合
      層の深さ，中間層のノード数，活性化関数，ドロップアウトの有無，・・・
  - 学習のさせ方に関する数値
    - 学習率，重みの最適化方法，バッチサイズ

DNNの問題点

1. 過学習問題
2. 初期値問題（局所最適解問題）
3. ハイパーパラメータの設定
4. 勾配消失問題
   - 時系列モデル（RNN，LSTM）の場合
     → Transformer

機械学習の本来の目的

× 学習データを正しく分類する
○ 学習データにない未知のデータを正しく分類する
汎化 (generalization) 性能

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