

|              |                                                                                 |
|--------------|---------------------------------------------------------------------------------|
| Title        | 最小分類誤り基準識別学習法の高精度化                                                              |
| Author(s)    | 六井, 淳                                                                           |
| Citation     |                                                                                 |
| Issue Date   | 1998-03                                                                         |
| Type         | Thesis or Dissertation                                                          |
| Text version | author                                                                          |
| URL          | <a href="http://hdl.handle.net/10119/1157">http://hdl.handle.net/10119/1157</a> |
| Rights       |                                                                                 |
| Description  | Supervisor: 下平 博, 情報科学研究科, 修士                                                   |

修 士 論 文

**最小分類誤り基準識別学習法の高精度化**

指導教官 下平 博 助教授

北陸先端科学技術大学院大学  
情報科学研究科情報処理学専攻知識情報処理学講座

六井 淳

1998 年 2 月 13 日

## 要 旨

本稿では近年、高性能な識別学習法として注目されている最小分類誤り識別学習法を取り上げ、その問題点を調査する。それとともに、より優れた汎化能力を有する新たな最小分類誤り基準を提案する。

# 目次

|          |                        |           |
|----------|------------------------|-----------|
| <b>1</b> | <b>まえがき</b>            | <b>1</b>  |
| 1.1      | 研究背景                   | 1         |
| 1.2      | 研究目的                   | 2         |
| 1.3      | 本論文の構成                 | 2         |
| <b>2</b> | <b>最小分類誤り学習の改良</b>     | <b>3</b>  |
| 2.1      | はじめに                   | 3         |
| 2.2      | MCE の特徴                | 3         |
| 2.3      | MCE の構成                | 6         |
| 2.3.1    | 識別関数の定義                | 6         |
| 2.3.2    | 誤分類尺度の定義               | 6         |
| 2.3.3    | 損失関数の定義                | 7         |
| 2.3.4    | 評価関数の定義                | 8         |
| 2.3.5    | 調整規則                   | 9         |
| 2.4      | MCE の問題点               | 12        |
| 2.5      | 汎化能力の指標                | 14        |
| <b>3</b> | <b>階層型ニューラルネットへの適用</b> | <b>16</b> |
| 3.1      | はじめに                   | 16        |
| 3.2      | 階層型ニューラルネットでの汎化能力      | 16        |
| 3.3      | 提案学習法                  | 18        |
| 3.3.1    | 正則化項                   | 18        |
| 3.3.2    | 学習アルゴリズム               | 18        |

|          |                             |           |
|----------|-----------------------------|-----------|
| 3.3.3    | 修正規則 . . . . .              | 21        |
| 3.3.4    | $m = 3$ の場合の修正規則 . . . . .  | 21        |
| 3.3.5    | $m = 2$ の場合の修正規則 . . . . .  | 22        |
| 3.4      | 計算短縮化修正則 . . . . .          | 22        |
| <b>4</b> | <b>認識実験</b>                 | <b>26</b> |
| 4.1      | はじめに . . . . .              | 26        |
| 4.2      | 実験 1：人工データによる認識実験 . . . . . | 26        |
| 4.3      | 実験 2：実データによる実験 . . . . .    | 28        |
| 4.4      | 正則化項の影響 . . . . .           | 30        |
| 4.4.1    | 評価関数への影響 . . . . .          | 30        |
| 4.4.2    | 入出力への影響 . . . . .           | 31        |
| 4.5      | 損失関数の影響 . . . . .           | 32        |
| 4.6      | 時間短縮化正則化項の認識結果 . . . . .    | 33        |
| <b>5</b> | <b>むすび</b>                  | <b>40</b> |
| 5.1      | 研究結果 . . . . .              | 40        |
| 5.2      | 課題 . . . . .                | 41        |

# 目 次

|     |                   |    |
|-----|-------------------|----|
| 2.1 | ML と MCE          | 4  |
| 2.2 | シグモイド関数           | 8  |
| 2.3 | ハイパータンジェント関数      | 9  |
| 2.4 | MCE の構成           | 11 |
| 2.5 | 汎化能力低下問題          | 12 |
| 3.1 | 三層構造ニューラルネットの概形   | 24 |
| 3.2 | 提案学習法             | 25 |
| 4.1 | 人工データ分布           | 28 |
| 4.2 | 評価関数値、正則化項値の推移    | 30 |
| 4.3 | 学習データ分布形状         | 35 |
| 4.4 | 評価用データ分布形状        | 35 |
| 4.5 | 従来法で学習終了          | 36 |
| 4.6 | 提案評価関数で学習終了       | 37 |
| 4.7 | $\xi$ の変化         | 38 |
| 4.8 | $\xi$ の変化による認識率変化 | 39 |

# 表 目 次

|     |       |    |
|-----|-------|----|
| 2.1 | 基準    | 5  |
| 4.1 | 比較    | 27 |
| 4.2 | UCI   | 29 |
| 4.3 | 比較    | 31 |
| 4.4 | 比較 2  | 33 |
| 4.5 | UCI2  | 34 |
| 5.1 | 重みの影響 | 42 |

# 第 1 章

## まえがき

### 1.1 研究背景

古典的パターン認識理論では、認識対象カテゴリの確率統計量が既知である場合、Bayes 識別が誤り損失最小の意味で最良であることが知られている。しかし、学習データから確率分布を推定するには、最尤推定法が用いられるが、有限個の学習データから確率分布を精度良く推定することは困難である。一方、分布の推定を行う代わりに、認識誤り最小化基準のもとで識別関数のパラメータを決定する手法として識別学習法がある。識別学習法は、実世界の問題においては最尤法に基づく Bayes 識別より優れた性能をしばしば示している。しかし、識別誤りの評価に用いられる損失関数は、識別関数のパラメータ集合に関して不連続であるため、効率の良い解法が存在しないという問題があった。

この問題に対して、Juang と片桐は1992年に最小分類誤り学習 (Minimum Classification Error Learning、以下 MCE) 法を提案した。MCE 学習では、微分可能な滑らかな損失関数を定義することによって、問題を連続関数の最適化問題の枠組で定式化することに成功した。その結果、最急降下法等の最適化手法の適用が可能となった。

MCE 法はこれまでに、階層型ニューラルネットワーク [1]、音声認識で利用される動的時間伸縮 (DTW) [4, 12, 5] や隠れマルコフモデル (HMM) [2] への適用が試みられ、従来の最小二乗誤差基準や最尤推定基準に基づく学習法と比較して、高い認識性能が得られることが示されている。しかし、MCE には問題もあり、実問題に適用された事例は少ないのが現状である。



## 1.2 研究目的

実問題に MCE 法を適用する場合に最も問題となるのが学習資料に対して高い認識精度を求めようとする、従来法と同様、識別関数のパラメータ集合が学習資料に特化してしまう傾向 (過学習) があることである。そのため、学習資料とは分布の異なった評価用資料に対しては識別性能 (汎化能力) が極端に低下するという問題がある。

過学習問題に対して、片桐や MacDermott は MCE 学習で用いる損失関数の形状 (主に傾き) を調整することによって過学習を抑え、汎化能力を高めることが可能であることを示している。しかし、損失関数の形状と汎化能力の関係は必ずしも明確ではないという問題がある。

本研究では、過学習問題を識別関数の入出力特性の側面から検討し、不良設定問題の解法で用いられる正則化の手法を用いることによって、MCE 法の汎化能力を改善する新しい手法を提案する。

MCE の持つ本質的問題点に対して解決法を与えることで、MCE を用いた様々な認識モデルに対する認識性能向上の効果も期待できる。

## 1.3 本論文の構成

次の第 2 章では MCE の特徴と構成について述べ、MCE の持つ問題点を示し、その解決法を与える。第 3 章では第 2 章で与えた解決法を階層型ニューラルネットに適用した場合の新たな MCE の構成法について述べる。第 4 章では第 3 章で与えた改良 MCE による実験結果を示し、提案手法の有効性について検討する。第 5 章では本研究のまとめと課題について述べる。

## 第 2 章

# 最小分類誤り学習の改良

### 2.1 はじめに

最小分類誤り (Minimum Classification Error) 基準識別学習法は学習データに対して高い認識性能を示す識別学習法である。この章では本研究で MCE を取り上げ、MCE の持つ特徴と学習規則の流れを説明する。

さらに本研究では MCE の高精度化を計ることを目標としている。そのため MCE の持つ問題点を明確にし、その解決法を与えることが重要である。この章では MCE の持つ問題点とその問題点として挙げられる汎化能力問題について触れる。また、汎化能力の指標を定義し、MCE の学習規則へ導入するための手法について述べる。

### 2.2 MCE の特徴

MCE は従来、広く用いられていた ML 基準などの従来法よりも学習データに関しては優れた認識結果を与える学習法であり、ML 基準などと根本的に違う点は学習基準である。図 2.1 は正規分布を考えた場合の ML 基準と MCE の学習基準の違いを示したものである。

モデルパラメータを平均と分散と考えた場合、ML はデータについて尤度が最大になるようパラメータを調節し、クラスの形状を推定する学習法である。これに対して MCE は各クラスごとの認識誤り数が最小になるようモデルパラメータを調節する。これは図 2.1 で見た場合、オーバーラップ部分ができるだけ小さくなるように各クラスの平均と分散を決めることになる。この違いを分かりやすくまとめたものが表 2.1 である。

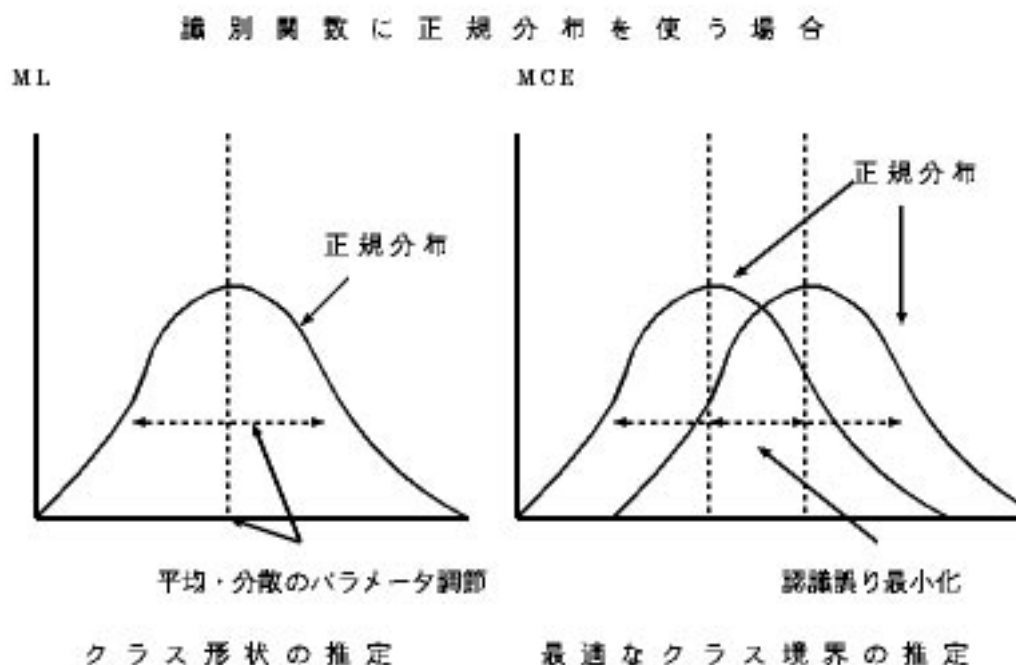


図 2.1: ML と MCE

ML などの従来法と学習基準の点で根本的に違うというだけでなく、学習データに関して極めて高い認識率を与えることが可能となった。なぜなら学習データに対しては誤認識率が最小になるように厳密にクラス境界を探索するため、単に各クラス形状を推定するよりもクラス認識に則した学習法といえるからである。現在では大量データを用いた認識モデルの学習において MCE で学習したモデルの方が ML で学習した認識モデルよりも高性能な認識モデルが作成できることが確認されている [1] [3] [4]。

MCE の特徴として挙げられるもう一点は MCE で用いる評価基準である。一般に学習はデータを識別関数に与え、ML ならば、尤度が最大になるような基準に基づいてパラメータが最適化される。しかし、MCE は認識誤り率を最小にすることを目標としているため認識誤り率を考える必要がある。認識誤り率は有限なデータ集合に関してパラメータ集合の区分的な定数関数であるため、従来の最適化法では容易に最小化することができない。MCE はこの問題に対し、パラメータに対して滑らかな認識誤り率の関数を定義でき

| 学習基準の比較 |                                   |                    |
|---------|-----------------------------------|--------------------|
|         | ML                                | MCE                |
| 学習基準    | 尤度を最大                             | 認識誤り数を最小           |
| 特徴      | カテゴリ形状を最も良く表現する<br>モデルパラメータを求める学習 | カテゴリの認識率を<br>上げる学習 |
| 学習データ   | 対象カテゴリのデータ                        | 全カテゴリのデータ          |

表 2.1: MCE・ML 各学習法の学習基準比較

る評価基準を有している。このため、最急降下法のような比較的簡単な最適化法によって最小化することが可能である。

この2点がMCEの大きな特徴である。次の節からはこの評価基準を与えるMCEの構成について説明する。

## 2.3 MCE の構成

パターン認識器や認識器の構成は、基本的に二つの段階に基づいている。クラス  $i, (i = 1, 2, \dots, M)$  に対する任意の識別関数  $g_i(\chi; \Lambda)$  の定義と、誤分類尺度決定における識別関数の利用である。

パーセプトロン学習に代表される従来の識別学習法では、誤認識率がパラメータ集合  $\Lambda$  に関して不連続な関数となるため、学習問題を最適化問題として定式化することに困難があった。

これに対して MCE ではこれから示すような誤認識率を表す評価関数がパラメータ集合に関する滑らかな関数として与えられるため、評価関数の最小化問題として容易に定式化される点に特徴がある。

### 2.3.1 識別関数の定義

識別学習の目的は最適な認識結果を与えるパラメータ集合  $\Lambda$  を得ることである。例えば  $K$  次のパターンベクトル  $x, (x \in \chi)$  についての線形識別関数を用いた場合にはパラメータ集合  $\Lambda = w_1, w_{01}, w_2, w_{02}, \dots, w_M, w_{0M}$  とする時、次のように書ける

$$g_i(\chi; \Lambda) = (w_i)^* x + w_{0i}, (i = 1, 2, \dots, M), * \text{は転置行列}$$

MCE ではこのような識別関数を学習規則に従い学習する。

### 2.3.2 誤分類尺度の定義

識別関数が与えられた後、誤分類尺度を定義する。誤分類尺度の最もシンプルな形は 2 カテゴリーに対して決まる次のようなベイズ識別である。

$$h(\chi) = P(C_2|\chi) - P(C_1|\chi)$$

$P(C_i|\chi)$  はポステリオリ確率。これはどのようにクラス 1 がクラス 2 として誤認識されているかを示してくれる。MCE の場合、誤分類尺度を決める手順は様々あり、甘利らによる提案では、誤分類尺度は次のようになる。

$$h_k(\chi; \Lambda) = \sum_{i \in S_k} \frac{1}{m_k} [g_i(\chi; \Lambda) - g_k(\chi; \Lambda)] \quad (2.1)$$

$S_k = \{i | g_i(\chi; \Lambda) > g_k(\chi; \Lambda)\}$ 、 $m_k$ はクラス数である。これだと  $S_k$ は固定した集合でないため、 $\chi$ とパラメータ集合 $\Lambda$ によって変化してしまう。つまり、 $\Lambda$ において不連続であり、最急降下法を用いる場合には MCE にとってあまり好ましくないものである。

そこでパラメータと関連した連続な誤分類尺度を次に示す。

$$h_i(\chi; \Lambda) = -g_i(\chi; \Lambda) + \left\{ \frac{1}{M-1} \sum_{j, j \neq i} g_j(\chi; \Lambda)^\eta \right\}^{1/\eta} \quad (2.2)$$

$\eta$ は平滑化パラメータ、 $M$ はクラス数である。この誤分類尺度は(2.1) と似ているが、(2.2) は連続であり、順応性がある。 $\eta$ が $\infty$ の時には誤分類尺度は次のように定義される。

$$h_i(\chi; \Lambda) = -g_i(\chi; \Lambda) + g_j(\chi; \Lambda)$$

この誤分類尺度から  $h_i(x; \Lambda) > 0$  は認識誤りを含んでいることを意味し、 $h_i(x; \Lambda) \leq 0$  ならば正しい決定を意味することになる。

### 2.3.3 損失関数の定義

誤分類尺度の決定を受けて、誤認識に伴う損失を表現するため、誤分類尺度を損失関数の形に変換する。損失関数は次のような形になる。

$$\ell_i(\chi; \Lambda) = \ell_i(h_i(\chi; \Lambda))$$

MCE の特徴として損失関数は微分可能な  $[0 - 1]$  関数である。損失関数として適用できるものとしては次の2つが挙げられる。1. シグモイド関数

$$\ell(h) = \frac{1}{1 + e^{-\xi(h+\alpha)}}, \quad \xi > 0 \quad (2.3)$$

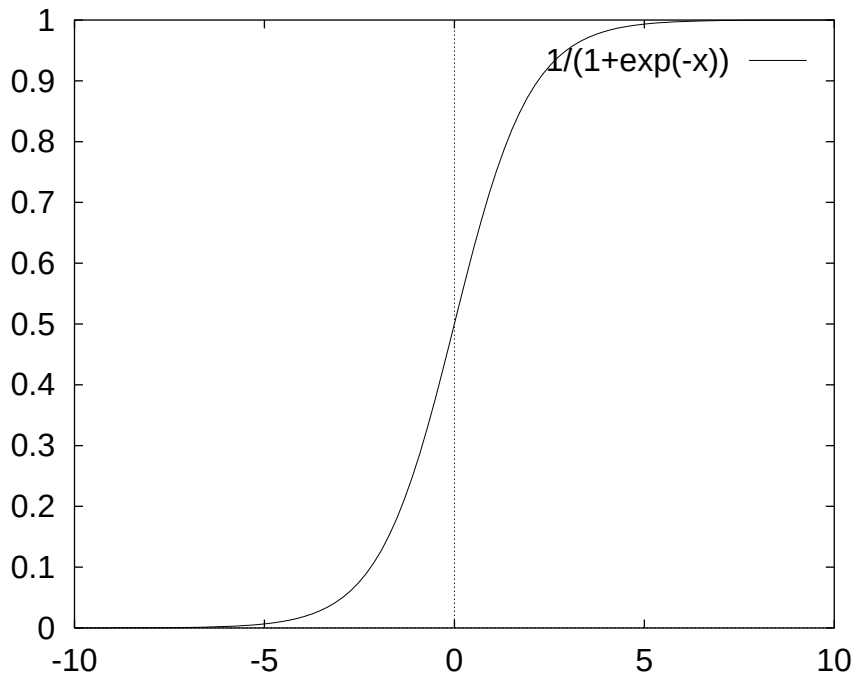


図 2.2: シグモイド関数

## 2. ハイパータンジェント関数

$$\ell(h) = \tanh(h)$$

本研究ではシグモイド関数を使用した。

### 2.3.4 評価関数の定義

損失関数で平滑化されると、 $h_i(\chi; \lambda) < 0$  であればコストはかからない。すなわち、 $h_i(\chi; \lambda)$  は認識誤りの計算のペナルティーを導くことになる。 $\chi$ に対して、次のような平滑化誤り量を定義する。

$$\ell(x; \Lambda) = \sum_{i=1}^M \ell_i(x; \Lambda) 1(x \in C^i)$$

$$1(\varepsilon) = \begin{cases} 1 & \text{事象 } \varepsilon \text{ が真} \\ 0 & \text{それ以外} \end{cases}$$

ここで、 $C^i$ はクラス  $i$  のパターン集合である。 $1(\cdot)$  は指標関数である。

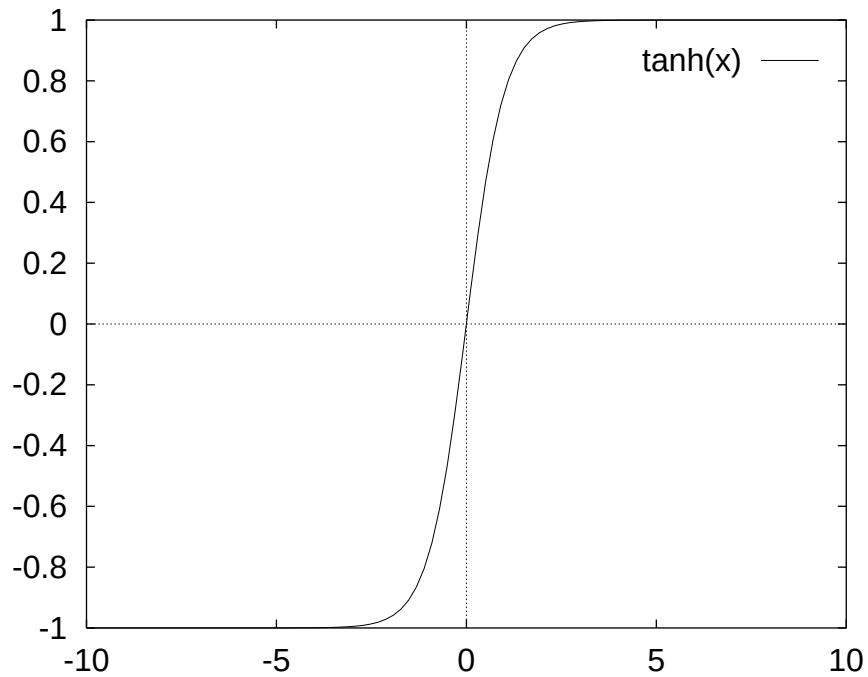


図 2.3: ハイパータンジェント関数

MCE の評価関数にはこの  $\ell(\chi; \Lambda)$  の期待値を使用する。

$$\begin{aligned}
 L_0(\Lambda) &= E[\ell(\chi; \Lambda)] \\
 &= \frac{1}{N} \sum_{k=1}^N \sum_{i=1}^M \ell_i(\chi_k; \Lambda) 1(\chi_k \in C^i) \\
 1(\varepsilon) &= \begin{cases} 1 & \text{事象 } \varepsilon \text{ が真} \\ 0 & \text{それ以外} \end{cases}
 \end{aligned} \tag{2.4}$$

評価関数 (2.4) はパラメータ集合  $\Lambda$  をラベル付けされた学習データを評価する度に少しずつ調整していく最急降下法によって最適化することが可能である。

### 2.3.5 調整規則

MCE の基準 (2.4) はパラメータ集合  $\Lambda$  を評価する度に、少しずつ調整していく最急降下法を用いて連続的に最小化することが可能である。今、 $\Lambda_{j+1}$  を  $\chi_j$  の適用後のパラメータ集合であるとする、 $\Lambda_{j+1}$  を得るための調整規則は次のようになる。



$$\Lambda_{j+1} = \Lambda_j + \Delta\Lambda_j$$

$$\Delta\Lambda_j = -\epsilon_j U \nabla \ell(x_j; \Lambda_j)$$

$\epsilon_j$ は確率的収束性に関する制約、 $U$ は正定値行列、 $\nabla$  はパラメータ集合 $\Lambda$ の勾配オペレーターである。この調整規則によって次の結果に到達する。

1.  $E[L_0(\Lambda_{j+1}) - L_0(\Lambda_j)] \leq 0$
2.  $\Lambda_j$ は  $L(\Lambda)$  の最小値 $\Lambda^*$ に収束

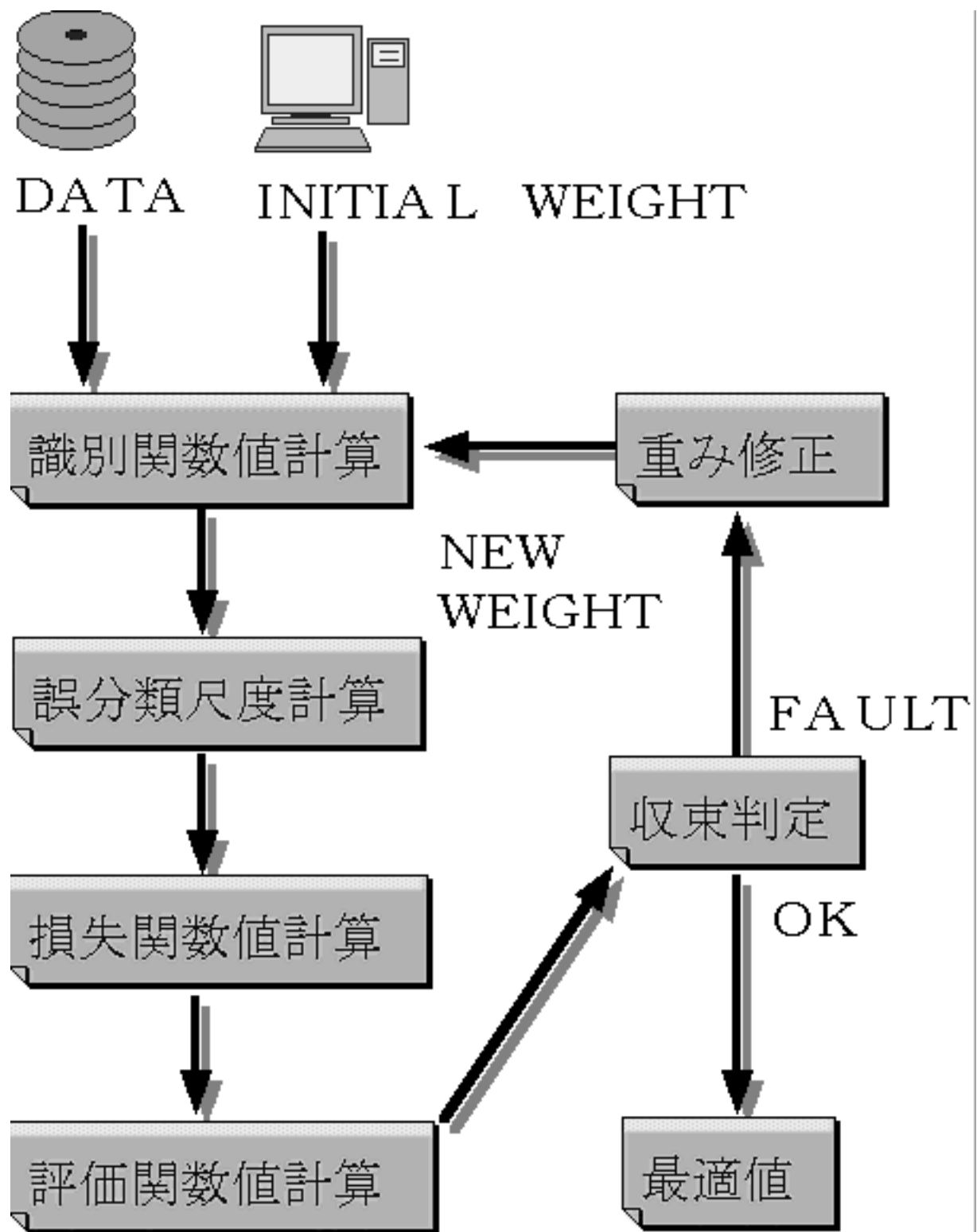


図 2.4: MCE の構成

## 2.4 MCE の問題点

MCE は学習サンプルが属するモデルが学習されるだけでなく、誤認識されやすいクラスについても認識率が向上する極めて高性能な学習法であるが、問題点もある。

MCE の抱える問題点として次のようなことが挙げられる。

- 最急降下法を使った学習アルゴリズムの効率問題
- 学習外資料を与えた際の認識率低下の問題

本研究では汎化能力低下問題を取り上げ、この解決を行う。先に述べたように実問題に MCE を適用する場合、学習の高速化や構造決定問題以上に優れた汎化能力を獲得することが重要な課題となる。しかし、汎化能力改善の研究は重要な課題であるにも関わらず、この問題に対する研究はほとんど行われていない。

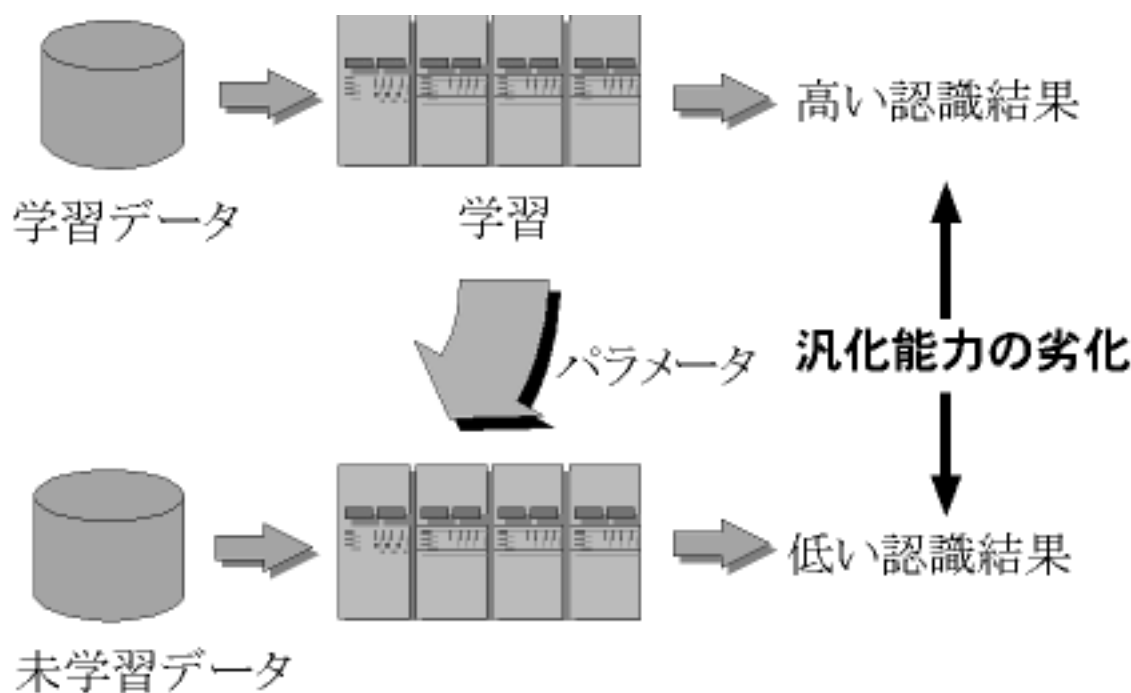


図 2.5: 汎化能力低下問題

式 (2.4) を用いた MCE 学習は、有限個の学習資料に対する識別誤りの最小化を行う。そのため、他の学習法と同様に、学習の結果得られる識別関数のパラメータ集合  $\Lambda$  が学

習資料に特化し (過学習)、学習外資料 (評価用資料) に対する識別性能 (汎化能力) が低下する問題が生じる。学習資料に関して厳密に境界を求める MCE 本来の目的が致命的問題点でもある。

MCE の汎化能力を向上する手法として、式 (2.3) に代表される損失関数の活性領域の傾きを調整する手法が提案されている [12]。すなわち、式 (2.3) のシグモイド型の場合は、 $\xi$  が汎化能力を制御するパラメータとなる。この手法の問題点は、汎化能力が特徴空間上の識別境界の形状や位置に関わる問題であるのに対して、パラメータ  $\xi$  はそれらとの関係が不明瞭である点である。

## 2.5 汎化能力の指標

汎化能力問題を解決するために本論文では、汎化能力 [6] [7] の観点から、過学習現象を、学習によって推定された識別関数と真の識別関数の間の誤差の分散として捉える。この誤差分散を低く抑えるためには、識別関数にはある種の滑らかさが要求される。すなわち、識別関数の入力変動に対する出力変動の比が小さいことが汎化能力の向上に繋がると考える。一般に学習が進むにつれ、識別関数は学習資料の細かな変動まで表現するようになるが、識別関数の滑らかさをペナルティとすることによって過学習を抑えることができる。したがって、MCE の学習に用いる評価関数を前記の評価関数 (2.4) に代えて次式のように定義する。

$$L(\Lambda) = L_0(\Lambda) + \gamma F(\Lambda) \quad (2.5)$$

この評価関数  $L(\Lambda)$  に与える  $F(\Lambda)$  は識別関数の滑らかさを表すペナルティ (正則化項)、 $\gamma$  は重みである。

識別関数の滑らかさを表す指標として次のような入力  $I$  と出力  $O$  についての指標 [9] [10] [11] を与える。

$$F = \frac{1}{2} \sum_{r=0}^R \int_a^b h_r(I) \left( \frac{\partial^r O}{\partial I^r} \right)^2 \partial I \quad (2.6)$$

ここで  $h_r \geq 0, r = 0, \dots, R-1, h_R > 0$ 。

この式 (2.6) は Tikhonov 型 [11] の正則化項である。これを有限個の学習データによる正則化項と考えると、 $R = 2$  の時は次のような和の形に変形することができる。

$$\begin{aligned} & \frac{1}{2} \int \left\{ \left( \frac{\partial O}{\partial I} \right)^2 h_1(I) + \left( \frac{\partial^2 O}{\partial I^2} \right)^2 h_2(I) \right\} \partial I \\ &= \frac{1}{2} \sum_{n=1}^N \sum_{i=1}^D \int \left\{ \left( \frac{\partial y_n}{\partial x_i} \right)^2 h_1(x_i) + \left( \frac{\partial^2 y_n}{\partial x_i^2} \right)^2 h_2(x_i) \right\} \partial x_i \end{aligned}$$

ここで、 $x_i$  は  $D$  次元空間の入力、 $y_n$  は出力を示している。

$$= \frac{1}{2C} \sum_{k=1}^C \sum_{n=1}^N \sum_{i=1}^D \left\{ \left( \frac{\partial g_n(x_k)}{\partial x_{ki}} \right)^2 1(x_k \in C^n) + \left( \frac{\partial^2 g_n(x_k)}{\partial x_{ki}^2} \right)^2 1(x_k \in C^n) \right\}$$

この式の後半部分を取り出すと、次のようになる。

$$F_2 = \frac{1}{2C} \sum_{k=1}^C \sum_{n=1}^N \sum_{i=1}^D \left( \frac{\partial^2 g_n(x_k)}{\partial x_{ki}^2} \right)^2 1(x_k \in C^m) \quad (2.7)$$

$x_k = (x_{k1}, \dots, x_{kD})$  は  $D$ 次元空間の  $n$  番目学習データである。この (2.7) を (2.5) で示した評価関数の正則化項  $F(\Lambda)$  として定義する。

正則化項については学習の計算時間短縮のため、(2.6) の  $R = 1$  とした場合の式は次のように与えることが可能である。

$$F_1 = \frac{1}{2C} \sum_{k=1}^C \sum_{n=1}^N \sum_{i=1}^D \left( \frac{\partial g_n(x_k)}{\partial x_{ki}} \right)^2 1(x_k \in C^m) \quad (2.8)$$

二乗和をとる代わりに (2.8) は次のようにも定義できる。

$$F_1 = \frac{1}{C} \sum_{k=1}^C \sum_{n=1}^N \sum_{i=1}^D \left| \frac{\partial g_n(x_k)}{\partial x_{ki}} \right| 1(x_k \in C^m) \quad (2.9)$$

次の章では階層型ニューラルネットを使って識別関数を学習した場合の (2.7) の正則化項としての適用方法について説明する。

## 第 3 章

# 階層型ニューラルネットへの適用

### 3.1 はじめに

(2.5) で与えた新たな評価関数  $L(\Lambda)$  はパラメータの異なる任意の識別関数に適用させることができる。この章では学習規則として階層型ニューラルネットを取り上げる。

### 3.2 階層型ニューラルネットでの汎化能力

提案した MCE 学習には一般に線形識別関数、学習ベクトル量子化そして階層型ニューラルネット、隠れマルコフモデルのような様々な識別関数が用いられる。本研究では階層型ニューラルネット取り上げ、改良型 MCE の実装法を示す。なぜなら階層型ニューラルネットの分野では様々なケースに応じた汎化能力向上研究が数多く行われているためである。本研究のような正則化項を扱うことで汎化能力向上を目指す先行研究も数多くあることを確認している。

階層型ニューラルネットにおいては広い意味での汎化能力を次のように分類している。

- 追加学習性能
- 耐ノイズ性
- 補間性能
- 外延的類推性能

汎化能力低下の事実上の問題点として荷重係数初期値、中間層ユニット数の影響が指摘されている。このような内部表現の重要性を考慮し、次のような手法による汎化能力改善学習法が提案されている。

- 雑音付加学習データの利用
- 荷重係数の再修正
- 過剰学習の回避

渡辺 [8] らは荷重係数値および中間層ユニット出力変化の抑制に着目し、誤差逆伝搬法に入出力変化量を適用することで汎化能力向上の効果を確認している。本研究では (2.7) を汎化能力を考慮した正則化項として評価関数へ加え、(2.5) のような新たな評価基準を与える。階層型ニューラルネットを識別関数として用いる場合の改良型 MCE の実装法を次に示す。

三層型ニューラルネットを対象とする第  $m$  層第  $i$  ユニット入力  $I_{pi}^{(m)}$  と第  $m$  層第  $i$  ユニット出力  $O_{pi}^{(m)}$  の関係を示すと次のようになる。

$$I_{pi}^{(m)} = \sum_{j=1}^{n_{m-1}} w_{ij}^{m,m-1} O_{pj}^{(m-1)} + \theta_i^{(m)}$$

$$O_{pi}^{(m)} = f(I_{pi}^{(m)})$$

ここで、 $p$  は学習パターン番号、 $f$  はシグモイド関数、 $\theta_i^{(m)}$  は第  $m$  層  $i$  ユニットの閾値、 $n_m$  は第  $m$  層内のユニット数を表し、 $w_{ij}^{m,m-1}$  は第  $m$  層第  $i$  ユニットと第  $m-1$  層第  $j$  ユニット間の荷重係数を示す。

三層型ニューラルネットの場合は入力層ユニット入力  $I_i^{(1)}$ 、出力層ユニット出力  $O_k^{(3)}$ 、正則化項  $F$  は (2.7) に従い次式のように表される。

$$F = \frac{1}{2P} \sum_{p=1}^P \sum_{k=1}^{n_3} \sum_{i=1}^{n_1} \left( \sum_{j=1}^{n_2} w_{kj}^{32} (w_{ji}^{21})^2 f''(I_j^{(2)}) \right)^2$$

この正則化項  $F$  は、入力層ユニット入力にする出力層ユニット出力の変化の情報を与え、汎化能力に対する適切な内部表現となる。



### 3.3 提案学習法

提案学習法は汎化能力向上のため新たな評価関数  $\mathcal{L}$  を MCE の学習規則へ適用する。評価関数  $L$  に正則化項  $F$  を加えた提案評価関数  $\mathcal{L}$  を (2.5) より次のように定義する。

$$\mathcal{L} = L + \gamma F$$

$\gamma$  は第 2 項に対する重みで経験的に決定する。提案学習法は正則化項  $F$  の値を最小にするもとで評価関数  $L$  を最小にする制限付き学習法である。

#### 3.3.1 正則化項

すでに正則化項  $F$  は次式のように定義した。

$$F = \frac{1}{2P} \sum_{p=1}^P \sum_{k=1}^{n_3} \sum_{i=1}^{n_1} \left( \sum_{j=1}^{n_2} w_{kj}^{32} (w_{ji}^{21})^2 f''(I_j^{(2)}) \right)^2$$

ただし、 $P$  はデータ数である。また、(2.4) より評価関数  $L$  は次のように定義できる。

$$L = \frac{1}{P} \sum_{p=1}^P \sum_{i=1}^{n_3} \ell_i(x; \Lambda) 1(x \in C^i)$$

#### 3.3.2 学習アルゴリズム

具体的に提案手法のパラメータ修正規則を述べる前に、背景となる理論について説明を行います。

$$z_i = f(a_i) \quad a_i = \sum_j w_{ij} z_j \quad (3.1)$$

$z_j$  は  $j$  番目ユニットの出力、 $w_{ij}$  はユニット  $j$  から  $i$  への重みである。関数  $f$  は次の通り

$$f(a) \equiv \frac{1}{1 + e^{-a}} \quad (3.2)$$

$$\frac{\partial f}{\partial a} = f(1 - f) \quad (3.3)$$

ネットワークは入力  $x_l, l = 1, \dots, L$ 、出力  $y_n, n = 1, \dots, N$ を持っているとする。

$$y_n = a_n = \sum_i w_{ni} z_i \quad (3.4)$$

(2.7) で与えた正則化項を次のように考える

$$F = \frac{1}{2P} \sum_{p=1}^P \sum_{l=1}^L \sum_{n=1}^N \left( \frac{\partial^2 y_{np}}{\partial x_{lp}^2} \right)^2$$

正則化項の重みによる導関数を示すと次のようになる。

$$\frac{\partial F}{\partial w_{ij}} = \frac{1}{P} \sum_{p=1}^P \sum_{l=1}^L \frac{\partial F_{pl}}{\partial w_{ij}} \quad (3.5)$$

$$F_{pl} = \frac{1}{2} \sum_{n=1}^N \left( \frac{\partial^2 y_{np}}{\partial x_{lp}^2} \right)^2 \quad (3.6)$$

分りやすくするために、データ番号  $p$  を省き、重みによる導関数を示すと次のような形になる。

$$\frac{\partial F_l}{\partial w_{ij}} = \sum_{n=1}^N \frac{\partial^2 y_n}{\partial x_l^2} \frac{\partial}{\partial w_{ij}} \frac{\partial^2 y_n}{\partial x_l^2} \quad (3.7)$$

$$= \sum_{n=1}^N \frac{\partial^2 y_n}{\partial x_l^2} \frac{\partial^2}{\partial x_l^2} \frac{\partial y_n}{\partial w_{ij}} \quad (3.8)$$

ここで (3.4) より次のことがいえる。

$$\frac{\partial y_n}{\partial w_{ij}} = \frac{\partial y_n}{\partial a_i} z_j \quad (3.9)$$

(3.9) を (3.8) に適用すると、与えられた導関数は次のようになる

$$\frac{\partial F_l}{\partial w_{ij}} = \sigma_{li} \frac{\partial^2 z_j}{\partial x_l^2} + 2\hat{\sigma}_{li} \frac{\partial z_j}{\partial x_l} + \hat{\sigma}_{li} z_j \quad (3.10)$$

$$\sigma_{li} \equiv \sum_{n=1}^N \frac{\partial^2 y_n}{\partial x_l^2} \left( \frac{\partial y_n}{\partial a_i} \right) \quad (3.11)$$

$$\hat{\sigma}_{li} \equiv \sum_{n=1}^N \frac{\partial^2 y_n}{\partial x_l^2} \frac{\partial}{\partial x_l} \left( \frac{\partial y_n}{\partial a_i} \right) \quad (3.12)$$

$$\hat{\hat{\sigma}}_{li} \equiv \sum_{n=1}^N \frac{\partial^2 y_n}{\partial x_l^2} \frac{\partial^2}{\partial x_l^2} \left( \frac{\partial y_n}{\partial a_i} \right) \quad (3.13)$$

(3.1) と (3.3) より

$$\frac{\partial y_n}{\partial a_i} = z_i(1 - z_i) \sum_k \frac{\partial y_n}{\partial a_k} \quad (3.14)$$

(3.14) より (3.11)(3.12)(3.13) は次のように与えられる

$$\sigma_{li} \equiv z_i(1 - z_i) \sum_k w_{ki} \sigma_{lk} \quad (3.15)$$

$$\hat{\sigma}_{li} \equiv z_i(1 - z_i) \sum_k w_{ki} \hat{\sigma}_{lk} + (1 - 2z_i) \frac{\partial z_i}{\partial x_l} \sum_k w_{ki} \sigma_{lk} \quad (3.16)$$

$$\hat{\hat{\sigma}}_{li} \equiv z_i(1 - z_i) \sum_k w_{ki} \hat{\hat{\sigma}}_{lk} + 2(1 - 2z_i) \frac{\partial z_i}{\partial x_l} \sum_k w_{ki} \hat{\sigma}_{lk} + \left\{ (1 - 2z_i) \frac{\partial^2 z_i}{\partial x_l^2} - 2 \left( \frac{\partial z_i}{\partial x_l} \right)^2 \right\} \sum_k w_{ki} \sigma_{lk} \quad (3.17)$$

また、同様に (3.1) と (3.3) から

$$\frac{\partial z_i}{\partial x_l} = z_i(1 - z_i) \sum_j w_{ij} \frac{\partial z_j}{\partial x_l} \quad (3.18)$$

$$\frac{\partial^2 z_i}{\partial x_l^2} = z_i(1 - z_i) \sum_j w_{ij} \frac{\partial^2 z_j}{\partial x_l^2} + (1 - 2z_i) \frac{\partial z_i}{\partial x_l} \sum_j w_{ij} \frac{\partial z_j}{\partial x_l} \quad (3.19)$$

以上より (3.10) が求まる。

### 3.3.3 修正規則

先の学習アルゴリズムに従い、提案学習法を行う際のパラメータ更新に三層型ニューラルネットを使用する場合の修正規則について述べる。

まず、改めて提案学習法の評価関数を下に示す。

$$\begin{aligned}\mathcal{L} &= L + \gamma F \\ &= \frac{1}{P} \sum_{p=1}^P \sum_{i=1}^M \ell_i(x; \Lambda) 1(x \in C^i) + \gamma \frac{1}{2P} \sum_{p=1}^P \sum_{k=1}^{n_3} \sum_{l=1}^{n_1} \left( \sum_{j=1}^{n_2} w_{kj}^{32} (w_{ji}^{21})^2 f''(I_j^{(2)}) \right)^2\end{aligned}$$

$P$ はデータ数、 $M$ はクラス数、 $n_m$ は第  $m$  層内のユニット数、 $\ell_i(x; \Lambda) 1(x \in C^i)$  は先に述べたようにクラス  $i$  の損失関数値である。このように与えた評価関数を次のように変形する。

$$\begin{aligned}\mathcal{L} &= \frac{1}{P} \sum_{p=1}^P L_p + \gamma \frac{1}{P} \sum_{p=1}^P \sum_{i=1}^{n_1} F_{pi} \\ L_p &= \sum_{i=1}^{n_3} \ell_i(x; \Lambda) 1(x \in C^i) \\ F_{pi} &= \frac{1}{2} \sum_{k=1}^{n_3} \left( \sum_{j=1}^{n_2} w_{kj}^{32} (w_{ji}^{21})^2 f''(I_j^{(2)}) \right)^2\end{aligned}$$

提案評価関数  $\mathcal{L}$  を最小化するためのパラメータ修正を次に示す。

$$w_{ij}^{mm-1} \leftarrow w_{ij}^{mm-1} - \alpha \frac{1}{P} \sum_{p=1}^P \Delta w_{pij}^m$$

ここで  $\Delta w_{pij}^m$  が以下のように与えられる。

$$\Delta w_{pij}^m = \frac{\partial L_p}{\partial w_{ij}^{mm-1}} + \gamma \frac{\partial F_{pi}}{\partial w_{ij}^{mm-1}}$$

これより修正規則は学習アルゴリズムに従い次のようになる。

### 3.3.4 $m = 3$ の場合の修正規則

$$\frac{\partial L_p}{\partial w_{kj}^{mm-1}} = \ell'_k(I_{pk}^{(m)}; \Lambda) 1(I_{pk}^{(m)} \in C^k) \frac{\partial h_k(I_{pk}^{(m)}; \Lambda)}{\partial I_{pk}^{(m)}} O_{pj}^{(m-1)}$$

$$\begin{aligned}
\frac{\partial F_{pi}}{\partial w_{kj}^{(mm-1)}} &= \frac{1}{2} \frac{\partial^2 O_{pk}^{(m)}}{\partial I_{pi}^{(m-2)^2}} \frac{\partial^2 O_{pj}^{(m-1)}}{\partial I_{pi}^{(m-2)^2}} \\
&= \frac{1}{2} \left( \sum_{t=1}^{n_2} w_{kt}^{mm-1} (w_{ti}^{m-1m-2})^2 f''(I_{pt}^{(m-1)}) \right) w_{ji}^{m-1m-2} f''(I_{pj}^{(m-1)})
\end{aligned}$$

### 3.3.5 $m = 2$ の場合の修正規則

$$\begin{aligned}
\frac{\partial L_p}{\partial w_{ji}^{mm-1}} &= \sum_{k=1}^{n_{m+1}} \left( \frac{\partial L_p}{\partial I_{pk}^{(m+1)}} w_{kj}^{m+1m} \right) \frac{\partial h_j(I_{pj}^{(m)}; \Lambda)}{\partial I_{pj}^{(m)}} O_{pi}^{(m-1)} \\
\frac{\partial F_{pi}}{\partial w_{ji}^{(mm-1)}} &= \frac{1}{2} \left\{ 2f''(I_{pj}^{(m)}) w_{ij}^{mm-1} + I_{pi}^{(1)} (w_{ij}^{mm-1})^2 [f'''(I_{pj}^{(m)}) - 2(f'(I_{pj}^{(m)}))^2] \right\} \\
&\quad \cdot \left( \sum_{k=1}^{n_{m+1}} w_{kj}^{m+1m} \left( \sum_{j=1}^{n_m} w_{kj}^{m+1m} (w_{ji}^{mm-1})^2 f''(I_{pj}^{(m)}) \right) \right) \\
&= \frac{1}{2} \left\{ 2f''(I_{pj}^{(m)}) w_{ij}^{mm-1} + I_{pi}^{(1)} (w_{ij}^{mm-1})^2 [f'''(I_{pj}^{(m)}) - 2(f'(I_{pj}^{(m)}))^2] \right\} \sum_{k=1}^{n_{m+1}} w_{kj}^{m+1m} \left( \frac{\partial F_{pi}}{\partial w_{kj}^{(m+1m)}} \right)
\end{aligned}$$

## 3.4 計算短縮化修正則

(2.9) の計算短縮化の正則化項を使用する場合は、修正規則も変える必要がある。

$$F_1 = \frac{1}{P} \sum_{p=1}^P \sum_{k=1}^{n_3} \sum_{i=1}^{n_1} \left| \frac{\partial g_k(x_n)}{\partial x_{ni}} \right|$$

ここで修正規則を考える。

$$w_{ij}^{mm-1} \leftarrow w_{ij}^{mm-1} - \alpha \frac{1}{P} \sum_{p=1}^P \Delta w_{pij}^m$$

$$\Delta w_{pij}^m = \frac{\partial L_p}{\partial I_{pi}^{(m)}} + \gamma \frac{\partial F_{1pl}}{\partial I_{pi}^{(m)}}$$

このように定義した場合、次のように与えられる。

$$L_p = \sum_{i=1}^{n_3} \ell(h_i(x_p; \Lambda_i) 1(x \in C^i))$$

$$F_{1pi} = \sum_{k=1}^{n_3} \frac{\partial f(I_{pk}^{(3)})}{\partial I_{pk}^{(3)}} \sum_{i=1}^{n_1} \left| \sum_{j=1}^{n_2} w_{kj}^{32} w_{ji}^{21} \frac{\partial f(I_{pj}^{(2)})}{\partial I_{pj}^{(2)}} \right|$$

この場合、 $m = 3$  の時の修正規則を次のように変化させれば良い。

$$\frac{\partial L_p}{\partial I_{pi}^{(m)}} = \ell'_i(I_{pi}^{(m)}; \Lambda) 1(I_{pi}^{(m)} \in C^k) \frac{\partial h_i(I_{pi}^{(m)}; \Lambda)}{\partial I_{pi}^{(m)}}$$

$$\frac{\partial F_{1pi}}{\partial I_{pi}^{(m)}} = f''(I_{pi}^{(m)}) \sum_{k=1}^{n_1} \left| \sum_{j=1}^{n_2} w_{ij}^{32} w_{jk}^{21} f'(I_{pj}^{(m-1)}) \right|$$

また、中間層ユニット ( $m = 2$ ) に関しては、次のようになる。

$$\frac{\partial L_p}{\partial I_{pi}^{(m)}} = \sum_{j=1}^{n_{m+1}} \left( \frac{\partial L_p}{\partial I_{pj}^{(m+1)}} w_{ji}^{m+1m} \right) \frac{\partial h_i(I_{pi}^{(m)}; \Lambda)}{\partial I_{pi}^{(m)}}$$

$$\frac{\partial F_{1pi}}{\partial I_{pi}^{(m)}} = \sum_{k=1}^{n_{(m+1)}} \left( \frac{\partial F_p}{\partial I_{pi}^{(m+1)}} w_{ki}^{m+1m} \right) f'(I_{pi}^{(m)})$$

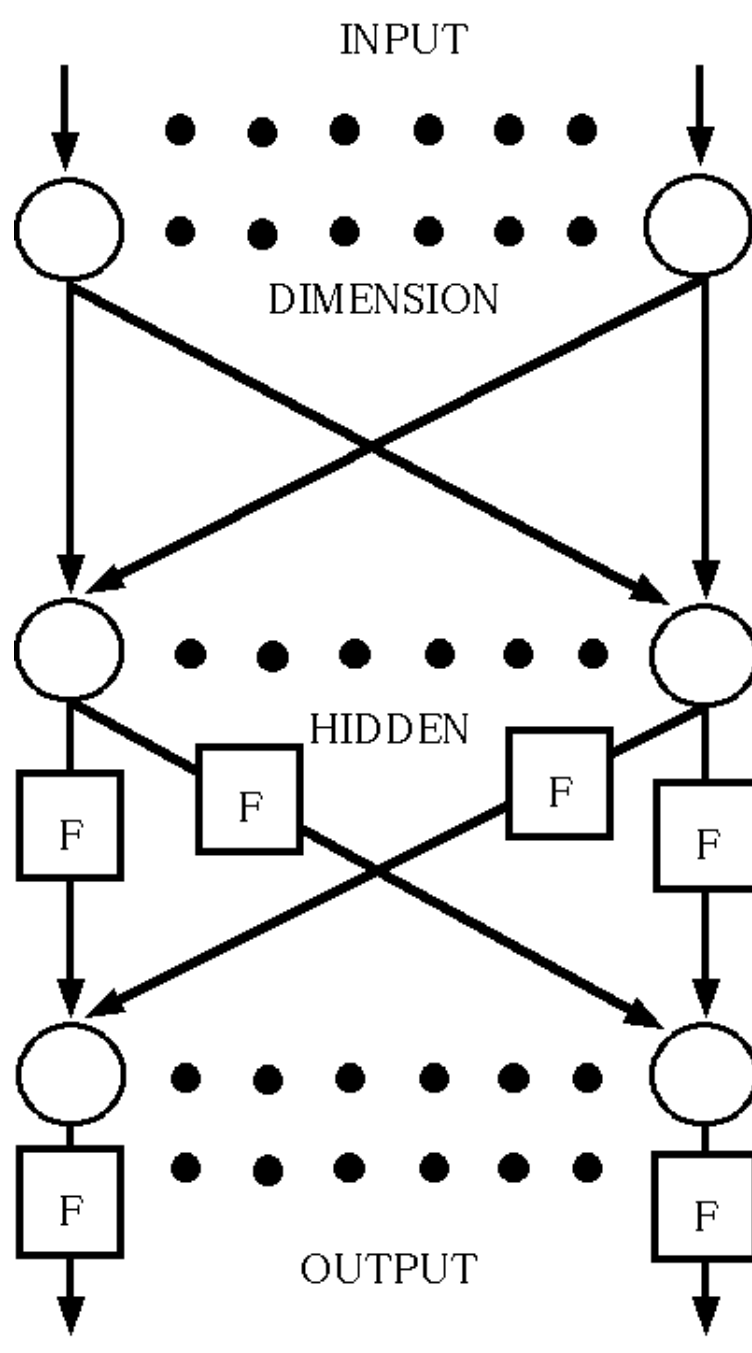


図 3.1: 三層構造ニューラルネットの概形

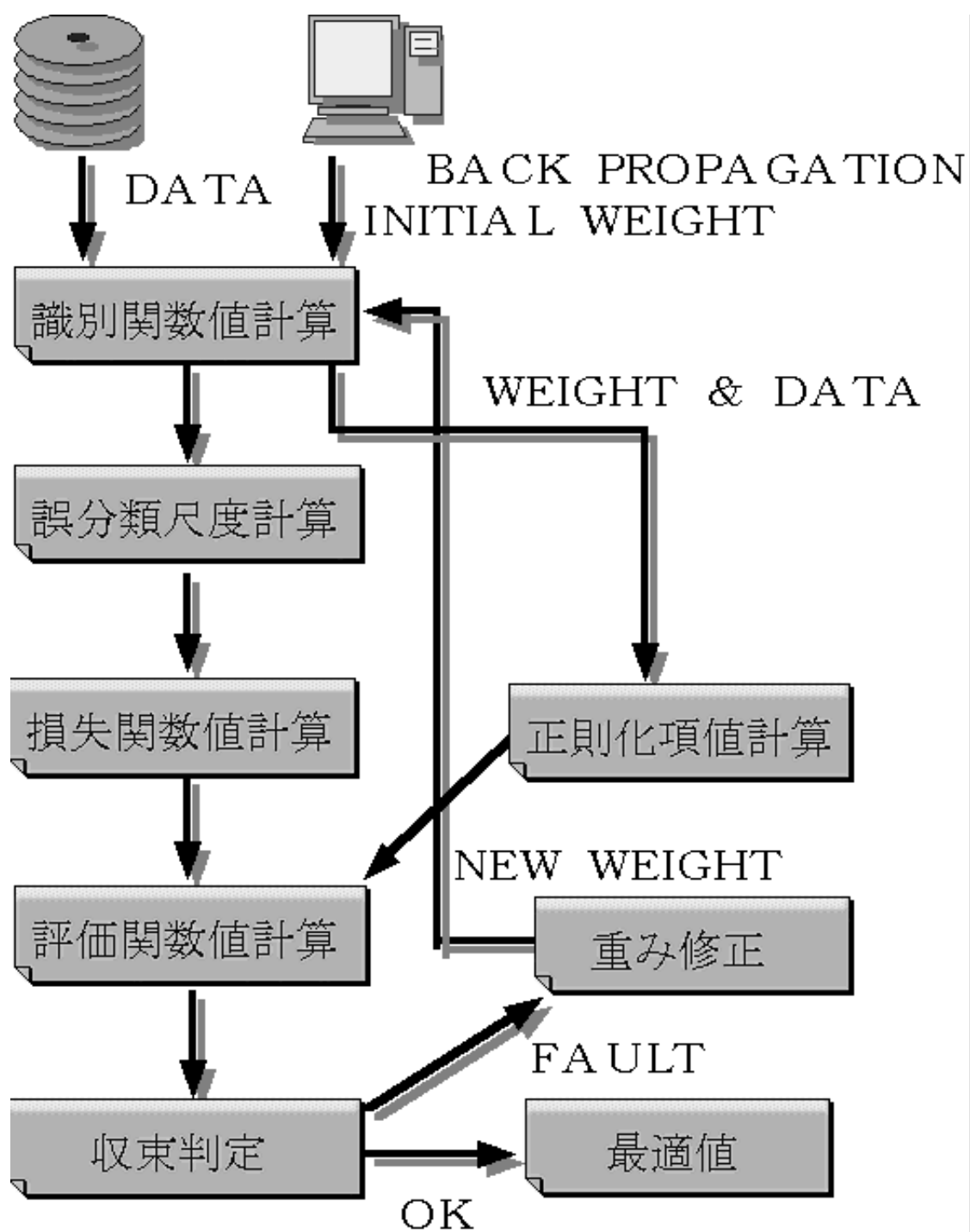


図 3.2: 提案学習法



## 第 4 章

# 認識実験

### 4.1 はじめに

実験に際し、学習規則には三層構造のニューラルネットワークを用いた。また、MCE はクラス同士の最適な境界を探索し、認識誤り率が最小になるよう学習を行う学習法である。このため、MCE は学習効率が悪い。MCE による学習を行う前に誤差逆伝搬学習法などによって初期パラメータを決定しておくことが望ましい。つまり、学習の精度と高速化の面から大体のクラス形状を予め与えておくということである。

実験で用いられるデータには人工データと実データの 2 種類がある。実データには UCI の機械学習用データベース<sup>1</sup>を使用している。

この章では実験結果を示すと共に、本研究で提案した改善法の有効性について説明する。

### 4.2 実験 1：人工データによる認識実験

人工的に発生させたデータについての実験結果を示す。2 次元 3 クラス問題を考えてみた。与えられた各々のクラスデータは正規分布から発生させたものである。学習データ 600 個、未学習データ 400 個を使用している。データの分布形状は図 5.1 に示してある。クラス 1 は平均  $(0.0, 2.0)$ 、分散  $\begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}$ 、クラス 2 は平均  $(-2.0, -2.0)$ 、分散

---

<sup>1</sup>UCI machine learning repository from University of California

| 実験結果            |           |              |
|-----------------|-----------|--------------|
| Method          | data type | correct rate |
| <i>Bayes+ML</i> | Training  | 97.83        |
|                 | Test      | 98.00        |
| <i>NN(MSE)</i>  | Training  | 97.67        |
|                 | Test      | 97.50        |
| <i>MCE</i>      | Training  | 98.83        |
|                 | Test      | 98.50        |
| 改 <i>MCE</i>    | Training  | 98.67        |
|                 | Test      | 98.75        |

表 4.1: 比較実験結果

$\begin{pmatrix} 1.1 & 0.0 \\ 0.0 & 1.1 \end{pmatrix}$ 、クラス 3 は平均  $(2.0, -2.0)$ 、分散  $\begin{pmatrix} 0.8 & 0.0 \\ 0.0 & 0.8 \end{pmatrix}$  である。クラス同士にはある程度のオーバーラップを持たせてある。

認識実験結果を次に示す。比較実験として Bayes+ML、NN(Minimizing Squared Error)、MCE、本研究で提案する改良 MCE の四種類の学習法による比較を行った。結果は表 4.1 に示す。

この実験結果から分るように改良 MCE は他の三つの学習法と比べて、評価用データに対して最も良い認識結果を得ている。注目すべき点はオリジナル MCE では学習データに対して 98.83% という結果を得ているが、改良 MCE では 98.67% と悪くなっている。個数にして 1 個の認識結果の差を生じている。ところが、評価用データを与えた場合、認識結果はオリジナル MCE では 98.50% であるが、改良 MCE では 98.75% となり個数にして 1 個を正しく認識している。これは厳密に境界を探索するという MCE が持つ本来の目的が汎化能力の低下を招いた例である。改良 MCE では正則化項の働きによって汎化能力を考慮した最小化が行われたため、オリジナル MCE よりも評価用データに関して良い結果が得られたものと考えられる。

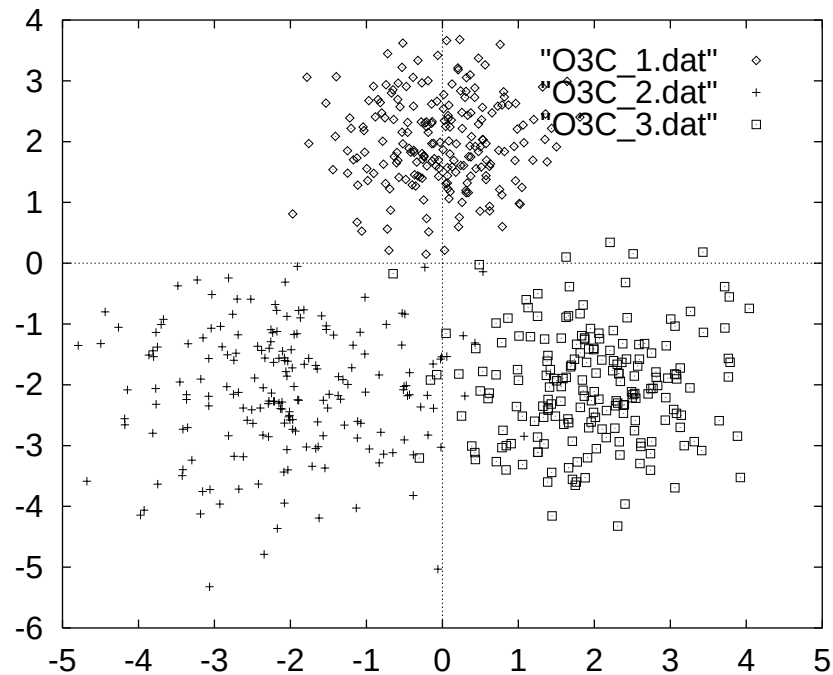


図 4.1: 与えた人工データ分布形状

### 4.3 実験 2：実データによる実験

様々な形状のデータに対しても人工データを用いた場合と同様に改良 MCE が結果を出せるかどうかについて考える。実験では実際に一般公開されているデータを使用し、三種類の学習法と比較した。実験結果は表 4.2 に示す。

UCE データベースのデータを使用した場合でも、改良 MCE が評価用データに対して最も良い認識結果を示している。MCE と改良 MCE を比較した場合、評価用データに関しての向上が大きいことが分る。特に、データベース Sonar による認識結果には各学習法に大きな差を生じている。これはデータベース Sonar が学習データと評価用データの分布形状に他のデータベースと比べて差があるためと考えられる。Sonar のような複雑なデータ分布形状のデータに対しては従来法を含め、これまでの手法では評価用データに関して著しく認識結果が悪くなる傾向がある。しかし、他の二手法に比べて、汎化能力の指標としての正則化項を含んでいた改良 MCE に良い結果が出た。このことから、本研究で提案した学習規則への正則化項導入が優れた汎化能力の獲得にうまく機能していることが分る。

| 実験結果      |             |        |       |       |
|-----------|-------------|--------|-------|-------|
|           | database    | Cancer | House | Sonar |
|           | Class       | 2      | 2     | 2     |
|           | Training    | 420    | 265   | 141   |
|           | Test        | 279    | 170   | 67    |
|           | Dimension   | 9      | 15    | 60    |
|           | Hidden Unit | 12     | 12    | 12    |
| $NN(MSE)$ | Training    | 91.90  | 96.25 | 95.04 |
|           | Test        | 90.32  | 96.47 | 82.09 |
| $MCE$     | Training    | 93.57  | 97.35 | 92.91 |
|           | Test        | 94.27  | 95.29 | 85.08 |
| 改 $MCE$   | Training    | 95.00  | 94.34 | 91.49 |
|           | Test        | 95.70  | 97.65 | 89.55 |

表 4.2: UCI データによる比較実験

## 4.4 正則化項の影響

### 4.4.1 評価関数への影響

正則化項がどのように機能しているのかを示す。図 4.2は評価関数値の推移を示したものである。学習回数 100 回までをプロットしてある。提案学習法で評価関数として使用するのは正則化項値と評価関数値の和の値である。図では”total” の値である。この実験では UCI データベース House を使用した。

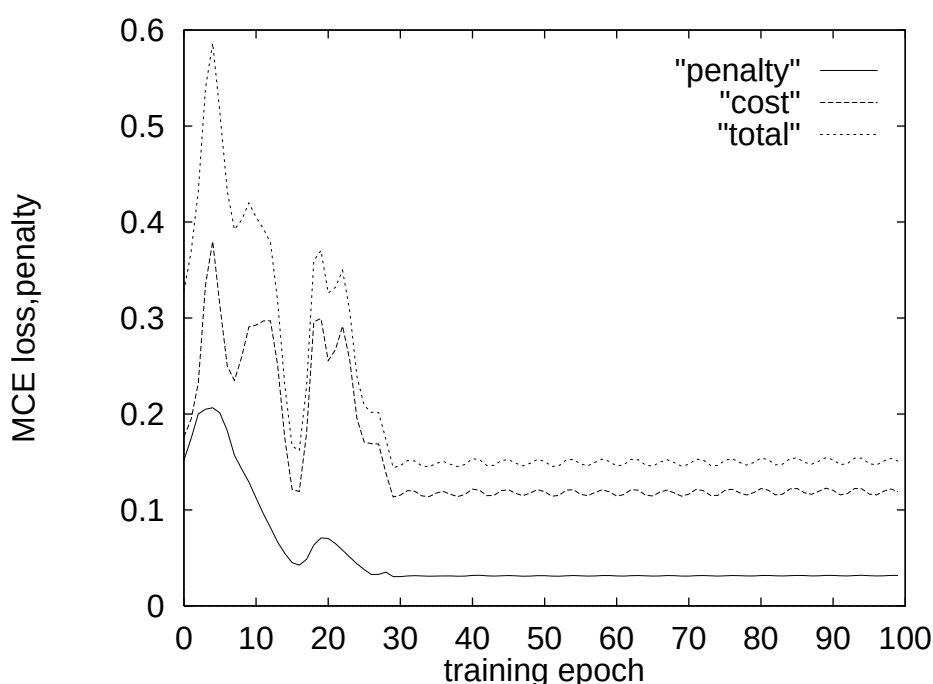


図 4.2: 評価関数値、正則化項値の推移

この実験は正則化項の動きとそれに伴う評価関数値の動きが焦点である。提案した評価関数を使用した場合、学習は 31 回で終了する。これに対し、従来法の場合、学習は完全に収束するまで行われる。表 4.3は学習終了基準を評価関数のみと正則化項を加えた提案評価関数を評価基準とした場合に分け、認識結果を比較したものである。学習回数は収束状況が分るよう意図的に 100 回まで表示した。

従来の MCE の場合、評価関数は学習の反復回数に関して単調減少となるが、改良 MCE ではある回数以上学習を重ねると正則化項の影響で評価値が悪化する。最急降下法による

| 実験結果                          |           |              |
|-------------------------------|-----------|--------------|
| estimation                    | data type | correct rate |
| <i>objective function</i>     | Training  | 97.35        |
|                               | Test      | 95.29        |
| <i>new objective function</i> | Training  | 94.34        |
|                               | Test      | 97.65        |

表 4.3: 評価基準の違いによる認識結果

収束判定は評価関数の改善率が所定の閾値以下になったところで行われるので、実際問題として改良 MCE では極小点付近で学習が停止することになる。これにより、新たな評価関数は過学習を抑えることで汎化能力を向上させていることが分る。

#### 4.4.2 入出力への影響

正則化項の影響によって学習終了時に入力と出力の差分の関係はどのようなになっているかを示す。図 4.5、図 4.6 は MCE と改良 MCE それぞれで学習した後の評価用データの入力と出力の差分の状況を示した図である。

図 4.5、図 4.6 を比較すると図 4.6 が図 4.5 よりも滑らかであることが分る。これはオリジナル MCE の場合、評価関数値が収束しきるまで境界を探索することになる。このため入力に対する出力が考慮されていないため出力の差分の変動が大きくなっている。これに対して提案評価関数を用いて学習を終了させた場合、過学習とならず学習が終了するため、出力の差分の変動はオリジナル MCE の場合よりも小さく抑えられている。

このように実際に入力と出力の変動状況を見れば、本提案手法が汎化能力を改善させていることが分る。

## 4.5 損失関数の影響

これまで MCE の汎化能力向上を考える上で重要なのが損失関数のパラメータである。本研究で損失関数として扱うシグモイド関数 (2.3) を使用している。

パラメータ $\xi$ を変化させることで汎化能力向上を計るのが主に行われている汎化能力改善法 [12] である。

次の実験ではオリジナル MCE と改良 MCE で用いられている損失関数の重み $\xi$ を調節することによって変化する認識率を調査した。

実験には UCI 機械学習用データベース House を使用した。図 4.8 上の認識率は評価用データについての認識率である。sig rate は MCE での認識率、sig rate new は本研究で提案した MCE での認識率である。

図 4.8 から $\xi$ の変化によって認識結果に大きな影響が出ていることが分る。この結果からどのように $\xi$ を変化させても改良 MCE の方が同等、もしくは、良い認識結果を得ている。これは、本研究で提案した手法が $\xi$ に関係なく正則化項の影響によって MCE の学習超過を抑え、MCE よりも優れた汎化能力を与えていると考えられる。

| 実験結果               |           |              |
|--------------------|-----------|--------------|
| Method             | data type | correct rate |
| <i>Bayes+ML</i>    | Training  | 97.83        |
|                    | Test      | 98.00        |
| <i>NN(MSE)</i>     | Training  | 97.67        |
|                    | Test      | 97.50        |
| <i>MCE</i>         | Training  | 98.83        |
|                    | Test      | 98.50        |
| <i>short-改 MCE</i> | Training  | 98.67        |
|                    | Test      | 99.25        |

表 4.4: 比較実験結果 (時間短縮化正則化項使用)

## 4.6 時間短縮化正則化項の認識結果

時間短縮化のために提案した手法 (2.9) についても、本研究で考えた提案手法と同様に汎化能力の向上が見込めることを示す。

まず最初に、人工データについての認識結果を表 4.4 に示す。人工データは提案手法と同じデータを使用した。

次は実データを用いた場合である。ただし、ここで用いている UCI データベースのデータは先の実験で示したデータとは違い、各データベースを 10 個のグループに分割し、9 グループのデータで学習、残り 1 グループについて認識実験を行い、この実験を 10 通り全ての組合せについて行って、認識率の平均値を示したものである (10-fold cross validation)。

この実験から分るように時間短縮化の正則化項を用いた場合でも、提案手法と同じように汎化能力向上が見込めることをが分る。



| 実験結果               |             |        |       |        |
|--------------------|-------------|--------|-------|--------|
|                    | database    | Cancer | House | Sonar  |
|                    | Class       | 2      | 2     | 2      |
|                    | Training    | 629    | 391   | 187    |
|                    | Test        | 70     | 44    | 20     |
|                    | Dimension   | 9      | 15    | 60     |
|                    | Hidden Unit | 12     | 12    | 12     |
| <i>Bayes+ML</i>    | Training    | 95.45  | 96.30 | 100.00 |
|                    | Test        | 95.27  | 94.72 | 71.12  |
| <i>NN(MSE)</i>     | Training    | 96.50  | 97.95 | 92.14  |
|                    | Test        | 95.14  | 90.91 | 71.43  |
| <i>MCE</i>         | Training    | 96.14  | 94.71 | 79.88  |
|                    | Test        | 95.59  | 93.64 | 75.00  |
| <i>short-改 MCE</i> | Training    | 95.38  | 95.89 | 89.11  |
|                    | Test        | 97.14  | 95.45 | 85.23  |

表 4.5: UCI データによる比較実験 (時間短縮化正則化項使用)

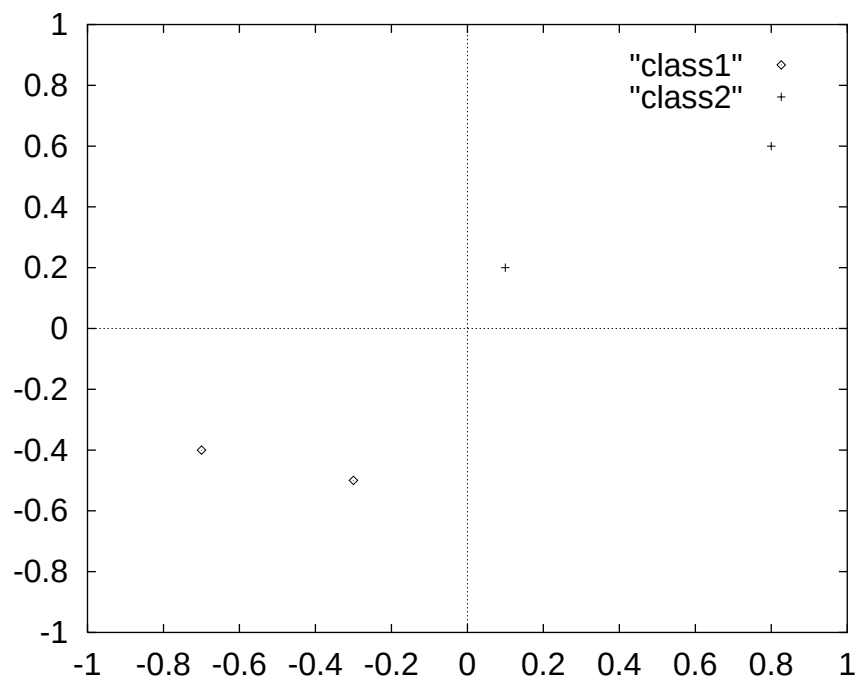


図 4.3: 学習データ分布形状

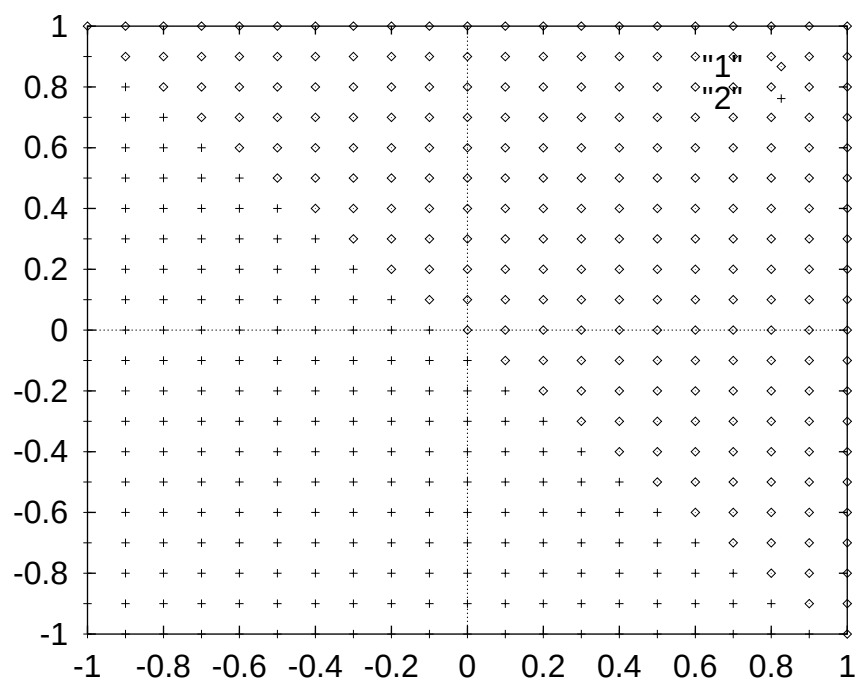


図 4.4: 評価用データ分布形状

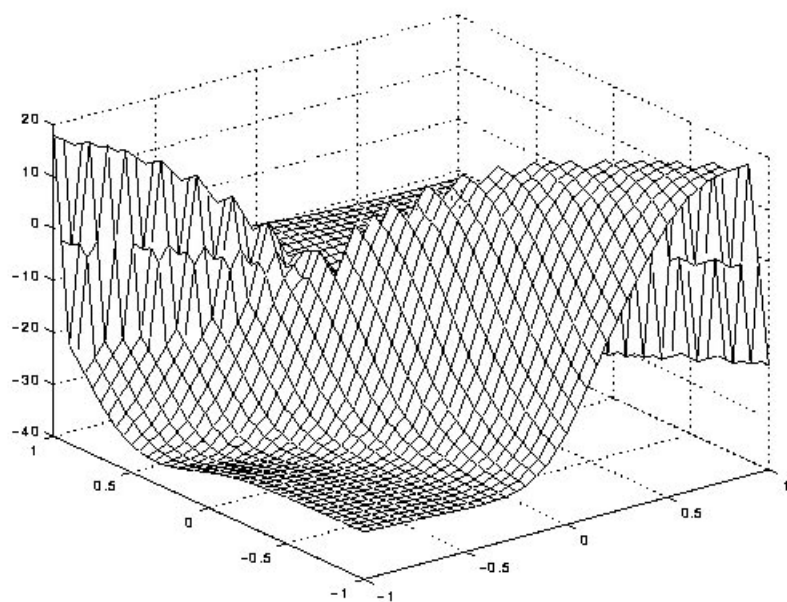


図 4.5: 従来法で学習終了

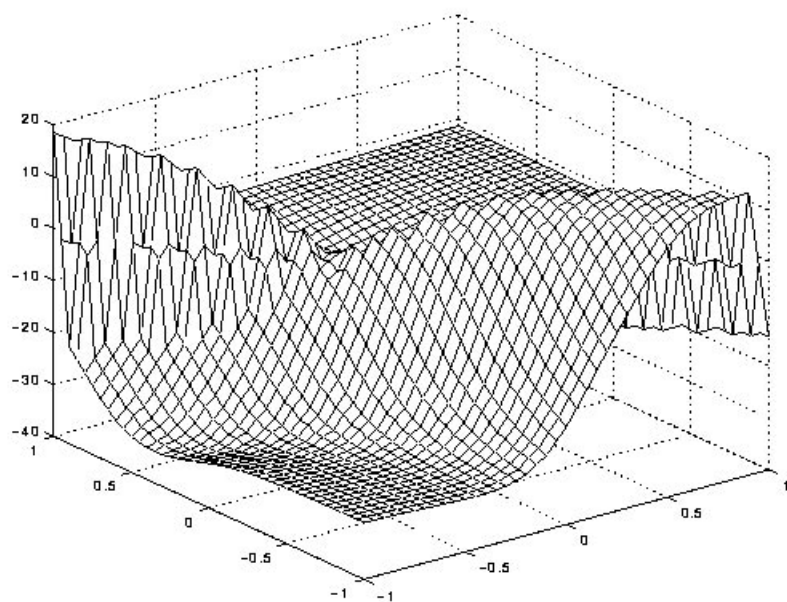


図 4.6: 提案評価関数で学習終了

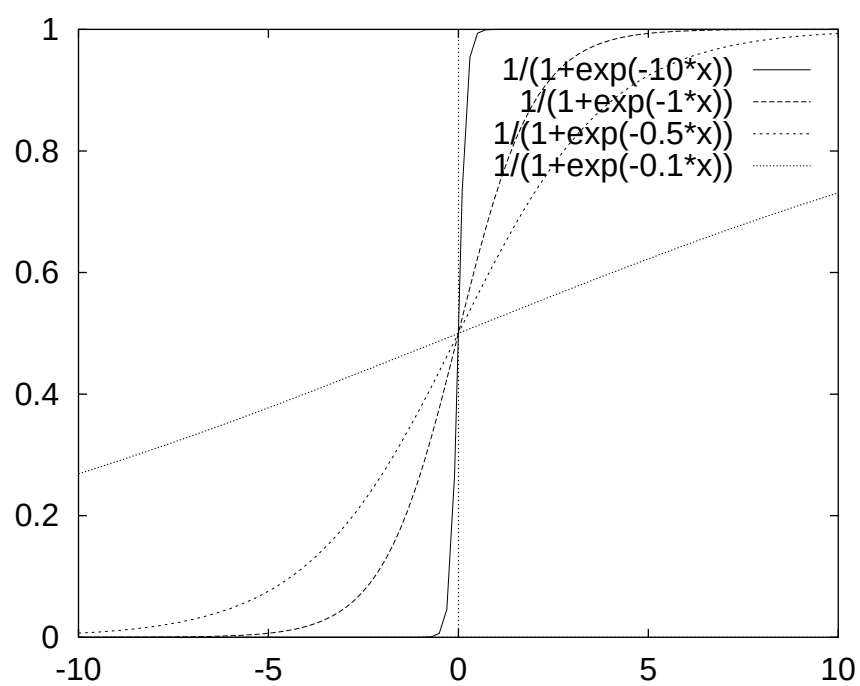


図 4.7:  $\xi$  の変化

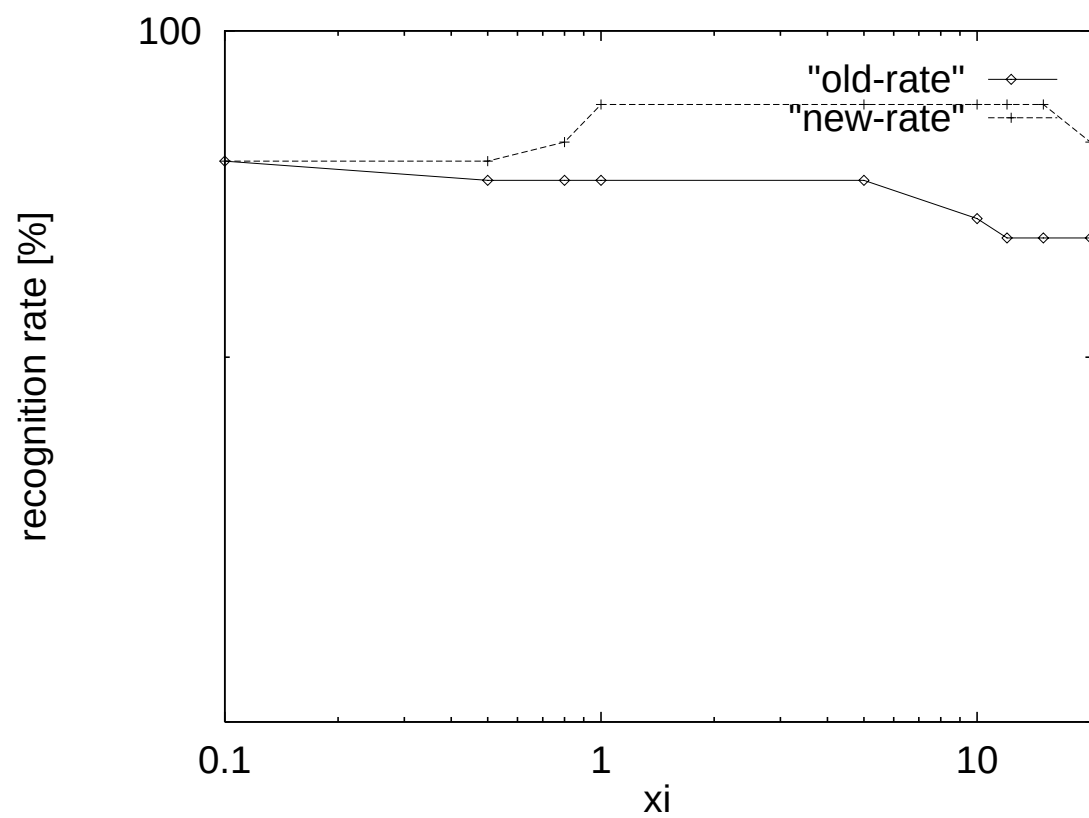


図 4.8:  $\xi$ の変化による認識率変化

## 第 5 章

# むすび

### 5.1 研究結果

本研究では高性能な識別学習法として近年注目されている MCE を取り上げ、その高精度化を行った。実問題に適用する際に MCE の抱える根本的問題である汎化能力問題解決のため、汎化能力の定量的指標となる一般形を与え、さらに MCE へ導入するという形で研究を進めた。過学習問題に対して、これまでは損失関数の形状を調節することで過学習を抑え、汎化能力向上を図っていた。しかし、損失関数の形状と汎化能力の関係は必ずしも明確でないという問題があった。

本研究では過学習問題に対して、識別関数の入出力特性の側面から検討を図った。提案手法は識別関数の入力と出力の変化量を汎化能力の指標に定め、これを正則化項として利用することで MCE 学習に伴う不良設定問題の回避を図った。識別関数として汎化能力改善の研究が多くなされている三層構造のニューラルネットを使用し、改良 MCE 構成の理論的枠組を示すと共に、実験を通して本研究の有効性を示した。

その結果、様々な形状のデータを使った実験では NN(MSE)、MCE との比較を行い、学習データでの認識結果よりも評価用データを与えた場合の認識結果が大きく向上したことを確認した。これにより様々なデータ形状に対応した汎化能力向上が見込めることが分った。また、提案した正則化項の影響についても実験を行い、従来法では評価関数が学習の反復回数に関して単調減少するのに対し、提案手法ではある回数以上学習を重ねると評価関数値が悪化することが確認された。これは提案学習法の評価関数に極小点が存在することを意味しており、収束判定を所定の閾値以下で行う手法において非常に有効であること

が分った。

さらに入力と出力の差分による比較から正則化項と評価関数を同時に最小化するという新たな評価基準を与えたことで、無駄な学習を抑えるだけでなく、出力の変動が抑えられた状態で学習を終えることができていることを確認した。

先に述べたように一般に汎化能力改善法として行われている損失関数のパラメータ調整による影響についても実験を行った。その結果、MCE との比較においてどのようにパラメータを変化させても提案 MCE の方が MCE と同等もしくは良い結果を得た。これにより、任意にパラメータを設定した場合でも MCE より本手法の方が優れた汎化能力を得ることができることを確認した。

## 5.2 課題

本研究で提案した評価関数は次のものである。

$$\mathcal{L} = L + \gamma F$$

評価関数  $L$ 、正則化項  $F$  を加えた提案評価関数  $\mathcal{L}$ 、 $\gamma$  は第 2 項に対する重みである。提案学習法は正則化項  $F$  の値を最小にするもとで評価関数  $L$  を最小にする制限付き学習法である。

この正則化項に与える  $\gamma$  の値は経験的に決定する必要があるため、データによって変えているのが現状である。時間短縮、認識率向上の面から、この  $\gamma$  を適切に決定する基準を設けることが大きな課題である。

本研究で提案した新たな評価関数 ( 2.5 ) において、正則化項の重みである  $\gamma$  は経験的に決定している。この  $\gamma$  の値によって、学習の収束時間、認識精度は大きな影響を受ける。現在は  $\gamma$  の値をデータによって経験的に変えているのが現状である。

表 5.1 は  $\gamma$  の値を変化させた時の認識率の状態を示したものである。データは UCI データベース House を使用した。

これから分るように  $\gamma$  の影響は認識性能に大きな影響があることが分る。現在はデータによって  $\gamma$  の値を決定しているが、この  $\gamma$  を決定する適切な基準を設けることができれば、さらなる認識性能が期待できる。今後はこの  $\gamma$  の値を与える基準を検討することが必要である。



| 実験結果  |           |              |
|-------|-----------|--------------|
| Gamma | data type | correct rate |
| 10.0  | Training  | 78.87        |
|       | Test      | 80.00        |
| 1.0   | Training  | 90.94        |
|       | Test      | 91.76        |
| 0.1   | Training  | 94.72        |
|       | Test      | 95.29        |
| 0.01  | Training  | 94.34        |
|       | Test      | 97.65        |
| 0.001 | Training  | 96.25        |
|       | Test      | 96.47        |

表 5.1:  $\gamma$ の影響

また、今回提案している正則化項以外にも様々な正則化項の定義があり、より有効なものの可能性についても検討する必要がある。

# 謝辞

本研究を行うにあたり、全般的御指導、御助言を頂いた下平博助教授に心から深く感謝致します。

また、木村正行教授、中井満助手には研究内容や問題点に関する御助言を頂き心から深く感謝致します。

下平研究室の Kanad Keeni 氏には研究を行う上で不可欠な階層型ニューラルネットに関する様々な御指導、御助言を頂き心から深く感謝致します。

木村・下平研究室の諸兄には日頃よりゼミなどで御討論頂き、心から感謝致します。

合同ゼミにおいては赤木研究室、小谷研究室、阿部研究室の方々にも御討論頂き、心から感謝致します。

最後に、本研究を進める上で非常にすばらしい研究環境を提供して頂いた両親と多くの大学関係者の方々に敬意を表し、本論文の結びとします。

## 参考文献

- [1] Biing-Hwang Juang, Shigeru Katagiri : “Discriminative Learning for Minimum Error Classification” , IEEE TRANSACTION ON SIGNAL PROCESSING , VOL.40 , NO.12 , DECEMBER 1992.
- [2] 高橋淳一、嵯峨山茂樹：「識別誤り最小化を用いた少量データのための効果的なHMM学習法」 信学技報 SP95-103.
- [3] 植本、松岡、松井、古井：「連続数字音声における音響モデル学習法の検討」、音学議論 1-Q-13、pp.121-122、1995.
- [4] W.Chou, C.-H.Lee, B.-H.Juang : “Minimum Error Rate Training of Inter-word Context Dependent Acoustic Model Units in Speech Recognition”, Proc, ICASSP94, pp.II-652-655.
- [5] 米沢祐司、赤木正人：「最小分類誤り学習による文脈効果モデルの定式化」、信学技報 SP94-114.
- [6] 喜多一：「ニューラルネットワークの汎化能力」、システム制御情報学会誌、Vol.36 No.10 pp.625-633 1992.
- [7] 小川英光：「ニューラルネットワークと汎化能力」、信学技報 NC95-8.
- [8] 渡辺栄治：「パターン認識問題に対する階層型ニューラルネットワークの汎化能力改善学習法」、信学論 D-II Vol.J79-D-II NO.5 pp.917-923 1996 5.
- [9] Christopher M. Bishop : “Curvature-Driven Smoothing: A Learning Algorithm for Feed-forward Networks”, IEEE Transactions on Neural Networks 4 NO.5 (1993) 882-884

- [10] Christopher M.Bishop: Neural Networks for Pattern Recognition,Oxford University Press,1995.
- [11] A.N.Tikhonov and V.Y.Arsenin: Solutions of Ill-Posed Problems,V.H.Winston,1977.
- [12] Eric MacDermott and Shigeru Katagiri : “Prototype-based minimum classification error / generalized probabilistic descent training for various speech units “ , Computer Speech and Language ,351-368 ,August 1994.
- [13] Lawrence Rabiner, Biing-Hwang Jung : 「音声認識の基礎 (上)(下)」 , NTT アドバンステクノロジー株式会社 1995.
- [14] 木村 義政 : 「誤差および出力変動を最小化する学習アルゴリズムとその低品質手書き漢字認識への応用」、信学論 D-II Vol.J73-D-II No.6 pp.840-847 1990 6.
- [15] 渡辺 栄治 : 「関数近似問題に対する階層型ニューラルネットワークの汎化能力改善学習法」、信学論 D-II Vol.J79-D-II NO.5 pp.924-931 1996 5.
- [16] 石川 眞澄 : 「コネクショニストモデルの忘却を用いた構造化学習」、信学技報、MBE88-144、1988.
- [17] 芳我 尚秀、石川 眞澄 : 「各種構造化学習法の構造化及び汎化能力の比較」、信学技報、NC93-20、1993.
- [18] A.N.Tikhonov, A.v.Goncharsky, V.V.Stepanov and A.G.Yagola : “Numerical Methods for the Solution of ill-posed Problems”, KLUWER ACADEMIC PUBLISHERS, 1990.

## 本論文に関する発表

- 六井 淳、下平 博：「最小分類誤り識別学習法の高精度化」、電気関係学会北陸支部連合大会、金沢市、11月2・3日1997.
- 六井 淳、下平 博、中井 満：「最小分類誤り学習の汎化能力改善とニューラルネットへの応用」、MIRU98 画像の認識、理解シンポジウム、岐阜市、7月29～31日1998投稿中.
- Hiroshi Shimodaira, Jun Rokui And Mitsuru Nakai : "Modified Minimum Classification Error Learning and Its Application to Neural Networks", 2nd International Workshop on Statistical Techniques in Pattern Recognition (SPR'98), Sydney, Australia, (August, 1998). Submitted
- Hiroshi Shimodaira, Jun Rokui and Mitsuru Nakai : "Improving the Generalization Performance of the Minimum Classification Error Learning", 13th biennial European Conference on Artificial Intelligence (ECAI-98), Brighton, UK, (August 1998). Submitted