

平成 14 年度

学士学位論文

ニューラルネットワークによる話者識別システムの提案

Proposal of Speaker Recognition System with Neural Network

1030286 中島弘喜

指導教員：竹田史章教授

2003 年 2 月 12 日

高知工科大学工学部情報システム工学科

要旨

高知工科大学工学部情報システム工学科 ニューラルネットワークによる話者識別システムの提案

中島 弘喜

本研究では、音声認識を用いた個人識別に着目し、指定単語を用いた話者識別を検討する。現状、音声認識には、HMM (Hidden Markov Model 隠れマルコフモデル) やコーパスを用いた手法で、単語、文章を読み、発話者自身の音響モデルを作成し音声認識を行う言語認識に主力が置かれている。そのため、音声認識を経て個人識別を行うことは、処理時間、識別性能の観点から適さない。

そこで、著者は音声を信号として扱い、その信号特徴を用いて、非線形識別能力を持つニューラルネットワーク (以下 NN と略記) による話者識別システムを提案し、高精度な本人識別率を得ることを、本研究の目的とする。

本研究では、NN による識別手法と NN テンプレートマッチング識別手法を融合させたニューロテンプレートマッチング識別手法を、話者識別に適用することにより、本人識別率の性能向上に期待する。以上により音声データを用いて、提案システムの有効性をオフラインシミュレーションで定量的に示す。

キーワード: 個人認証, ニューラルネットワーク, 話者識別, HMM, パターン認識, テンプレートマッチング

Abstract

Proposal of Speaker Recognition System with Neural Network

Nakazima Hiroki

In this paper, I discuss individual recognition using speech with word. In the current speech recognition, the main component is put on the language recognition which reads a word and a text, creates an utterance person's own sound model, and performs speech recognition by the technique using HMM(Hidden Markov Model) or the Corpus. Therefore, speech recognition is not suitable from a processing time and a recognition performance to perform individual recognition.

I propose neuro-template matching method for speaker recognition system which an author treats a voice wave as a signal and has nonlinear recognition capability using the signal feature. I expect to improve recognition ability of individual recognition.

I show effectiveness of the propose method by off-line simulation with voice data.

Keywords: Individual Recognition, Neural Network, Speaker Recognition, HMM, Pattern recognition, Template matching method

目次

1	はじめに	1
2	個人識別としての音声認識	2
3	ニューラルネットワークを用いた話者識別	4
3.1	入力部	5
3.2	前処理部	8
3.3	特徴抽出部	8
3.4	登録、識別部	14
4	実験	17
5	まとめ	27
6	謝辞	28
7	参考文献	29
8	付録	30

図目次

2.1	音素 HMM の構造	3
3.1	システム構成	5
3.2	マイク位置	6
3.3	ノイズ波形	7
3.4	「あ」と発声した FFT 波形	7
3.5	発話ピーク FFT 波形	10
3.6	データファイルの構成	14
3.7	NN パラメータ設定ファイルの内容	15
3.8	関数定数と振動定数の範囲	16
4.1	NNT の構成	18
4.2	NNC の構成	18
5.1	各被験者のスラブ値の分散値グラフ	25

表目次

3.1	NN データ 1 サンプルのフォーマット	11
3.2	ヘッダ情報のフォーマット	12
4.1	実験条件	20
4.2	実験 1 実験結果	20
4.3	実験 2 実験結果	22
5.1	各被験者のスラブ値の分散値グラフ	25

1 はじめに

近年、インターネットの普及により、インターネット銀行やメールの送受信において本人確認を行う個人認証の機会が増している。現在、その場面において文字による ID, Password 方式が用いられている。しかしながら、様々なサービス毎に異なる ID, Password の管理等の複雑化により、利便性の高い個人認証の確立が求められている。本研究では、人が普段のコミュニケーションでもっとも用いられる「音声」に着目し、バイオメトリクス^[1]個人認証技術として話者識別システムの提案を行う。

有声音などの音声波は、声帯振動波に人それぞれ固有の口腔や鼻腔などの声道の共振特性が加わって固有な特徴が生じる。音声認識の分野において、音響的な特徴を統計的にモデル化する隠れマルコフモデル(以下 HMM と略記)手法が一般的に用いられる。音声認識は多くのデータを分析し、そこから得られた特徴空間内における特徴の分布をもとにいくつかの正規分布を認識に用いる。多くの音声認識システムでは、音素の定常部に着目している^[2]。さらに、定常部以外の前後の音素も考慮に入れてモデル化をする場合もある^[3]。前後両方の音素の組み合わせを、考慮することから組み合わせ総数は、膨大なものとなる。また話者識別においては、発声者自身の正解データを作成する際の作業が長時間かかるという問題点が生じる^[4]。

本研究では話者識別において、人の音声波固有の特徴で識別を行う必要性から筆者は非線形識別能力を持つニューラルネットワーク(以下 NN と略記)による識別手法に用いることを提案し、高性能な識別性能を得ることを本研究の目的とする。ここでは、NN による選別手法とテンプレートマッチング識別手法を融合させたニューロテンプレートマッチング識別手法を適用することにより、被験者と被験者以外を識別するシステムを提案する^[5]。

2 個人識別としての音声認識

本章では、一般的に用いられている HMM の性能や構成について述べ、問題点を明確にする

音声は時間軸上に振幅値が並ぶ二次元情報である。音声を認識するためには、言語情報に対応する物理的な特徴量のみを音声波形から特徴抽出することが必要である。音声の音源としては声帯の振動(周期信号)、声道の狭めに伴う乱流(雑音信号)、声道の閉鎖と急速な開放に伴う破裂(インパルス信号)等から構成される。また摩擦音、破裂音、破摩擦音、半母音からなる有声音の音声波は声帯振動波に、人それぞれ固有の口腔や鼻腔などの声道の共振特性が加わって固有な特徴が生じる^[6]。音声の特徴量は微小な時間区間の音素と呼ばれるフレーム時間毎に計算される局所的な特徴で表される。しかしながら、音声中の音素は連続する複数のフレームからなるために、音素の音響的な性質を扱うためには、本来の特徴量を時系列として扱わなければならない。音素の継続時間は、発声者、環境等に応じて一定ではなく、多様に変化するため、1960 年代から様々な長さの時系列を確率的に生成することができる HMM が一般的に用いられるようになった^[7]。

HMM は時系列信号の確率モデルであり、複数の定常信号の音素間を遷移することで、非定常な時系列信号をモデル化することができる。図 2.1 に音素 HMM の構成例を示す。

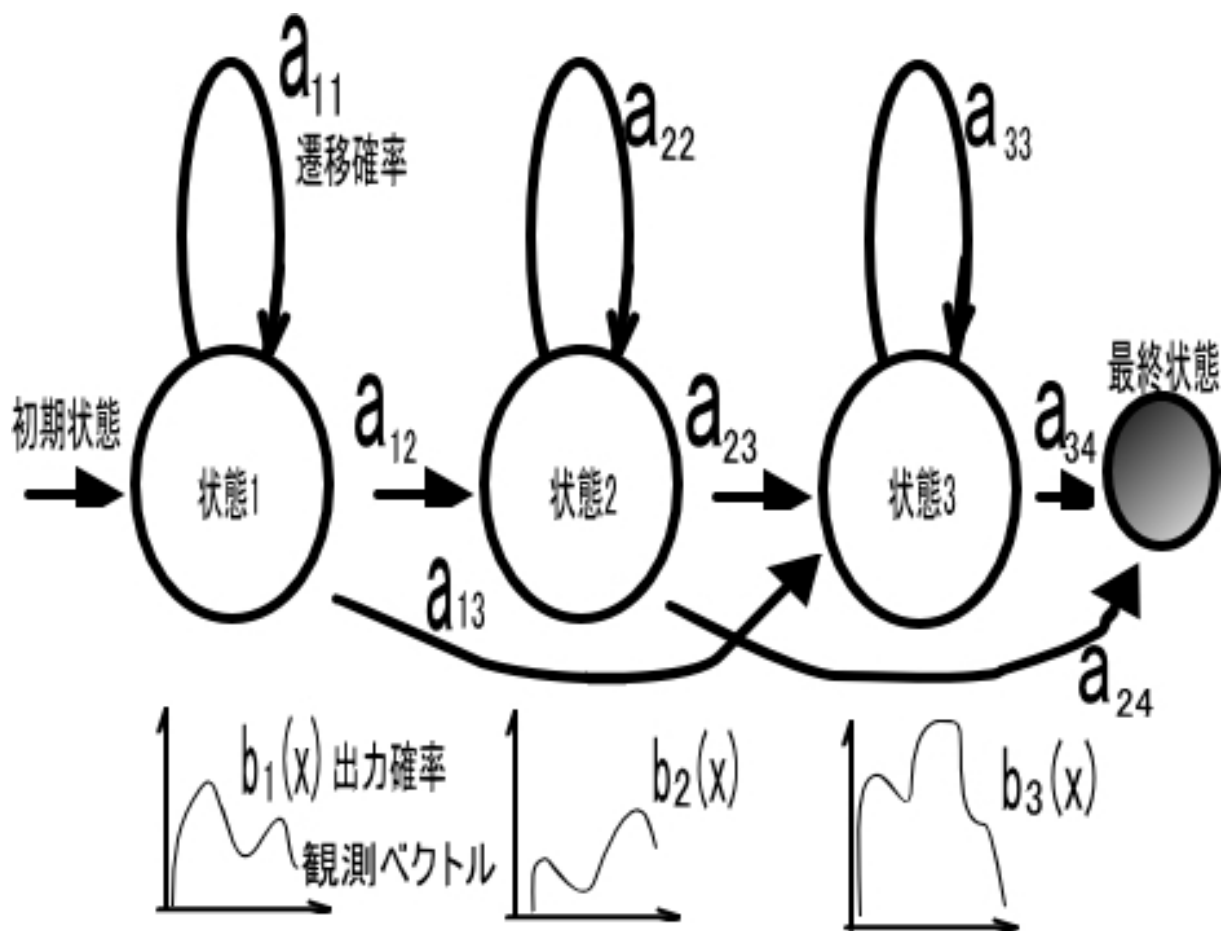


図 2.1 音素 HMM の構造

音素 HMM は、状態間の遷移の確率をあらわす状態遷移確率と、状態が遷移する際にシンボルを出力する出力確率からなる。発話者毎に、音素データを集合させた音響モデルを生成し音声認識を行う^[8]。したがって、認識を行うためには発話者毎、発声単語毎に大量な音響モデルデータベースの作成が必要のため、膨大な計算時間と音声を登録する作業時間を要する問題点が生じる。

3 ニューラルネットワークを用いた話者識別

現在主流の HMM を用いた話者識別システムは, 単語や文による音声認識を経ることを前提とするため, 作業効率および, 処理効率を考慮すると必ずしも有効な手段ではないと考えられる. そこで, 本研究では音声を用いた話者識別に NN を用いた識別手法を提案し, 識別性能の高性能化, 高効率化を図る. また, 特徴抽出において音素毎の特徴を識別に用いるのではなく, 単語毎の特徴を用いた識別を行う.

本章では, 提案システムの構成について 5 つの機能の概要を記述し, 各節にて詳細を示す.

- 入力部 音声を録音する
- 前処理部 NN へ入力するために入力データを高速フーリエ変換(以下 FFT と略記)により, リニア値化する
- 特徴抽出部 さらに, リニア値化したデータを NN への入力スペクトルを作成するために, NN の入力細胞数に合わせるために 7 加算平均し 80 次元の入力ベクトルを作成する
- 登録部 特徴データの学習を行う
- 識別部 学習結果から, 識別結果を出力する

で, 構成される. システム構成を図 3.1 に示す.

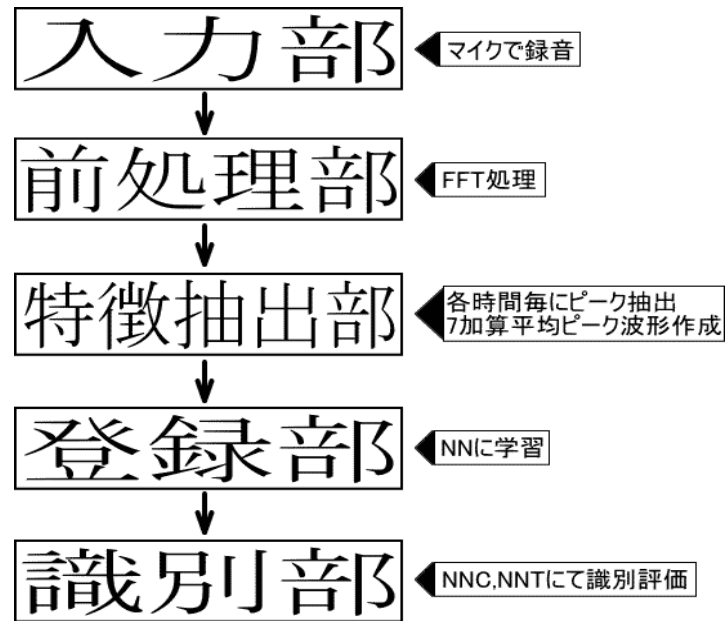


図 3.1 システム構成

3.1 入力部

本節では，入力部の音声録音について記述する．音声録音はパソコンにつないだモノラル 1Ch マイクから，音声を PCM，サンプリング周期を 44.1kHz とし，16bit で量子化し採取する．マイク位置は息によるノイズが入らないようにするために，口から水平線を引いたラインよりも下に位置するように配置する．また口からの距離は約 3.0cm 離れた位置とする．図 3.2 にマイク位置を示す．

録音環境は，室内でエアコンを停止状態の静音空間で採取する．図 3.3 に，発話せず録音した無音（室内ノイズ）波形を示す．図 3.4 に「あ」と発声した波形と比較したところ，室音ノイズはほぼ無視できる音量であると言える．



図 3.2 マイク位置



図 3.3 ノイズ FFT 波形



図 3.4 「あ」と発声した FFT 波形

提案システムは、各被験者が特定の同単語を発話することにより、話者識別を行う。本研究では、指定単語として「こうちこうかだい」を用いる。

3.2 前処理部

本節では、前処理部の録音音声の数値化について記述する。前節にて、録音した音声データ(アナログデータ)を NN へ入力するために、フーリエ係数のパワースペクトルを高速フーリエ変換(以下FFTと略記)により算出した、リニア値化する。本研究では、コンピュータや電子機器から 70Hz 付近にノイズが発生していることと、図 3.4 に示すように 6000Hz 付近まで音声波の特徴があることを考慮し、周波数 200Hz から 6000Hz の範囲を時間窓 46.4ms (分解能 21.5Hz, データ数 2048 点)で FFT 処理を行う。また、時間窓の両端が不連続になり、波形に歪みが生じることを解消するための重みがけとして、サイドローブの最高値が小さい、式(3.1)に示す Hamming 窓を用いる。音声波は、主に低周波帯域において個人毎の特徴を、多く含有しており高周波帯域においては雑音が混入しやすいことが報告されている^[10] 両端にゼロの信号を付加し、データ数を 2 倍するゼロ補間を行う。

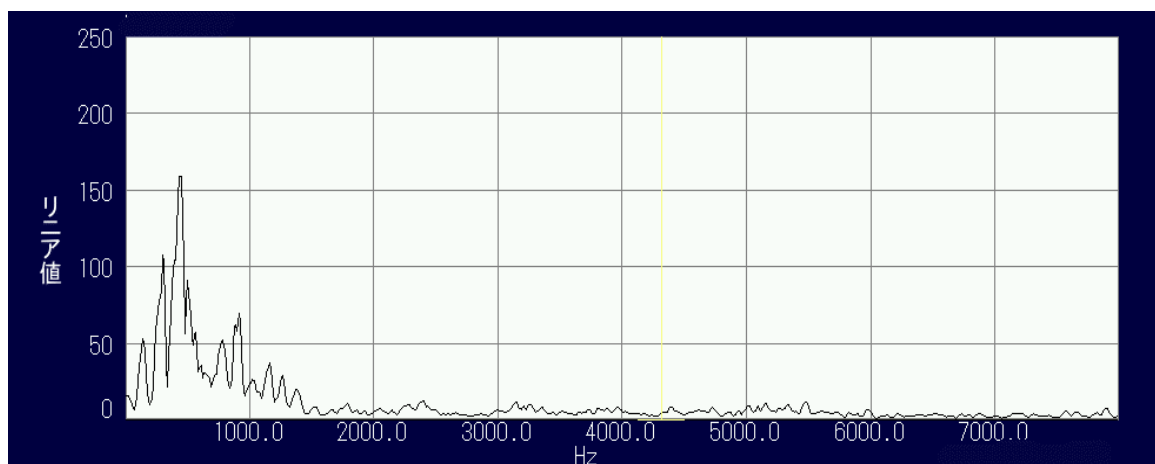
$$W_H = 0.54 - 0.46 \cos\left(\frac{2n\pi}{N-1}\right) \quad (3.1)$$

3.3 特徴抽出部

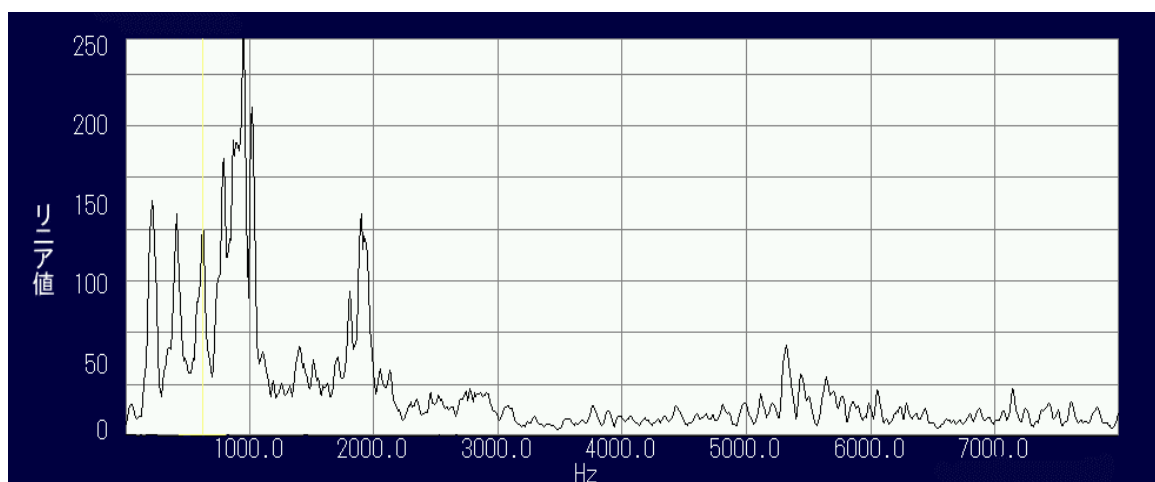
本節では、前処理部によってリニア値化された音声データから、NN へ入力するための特徴抽出方法について記述する。

音声波形を時間領域から周波数領域へ変換すると、通常 5kHz 以下に 4～5 個のピークが観測される。このピークはホルマントと呼ばれ、母音の識別に大きな役割を占める。HMM では、音素毎に音響モデルを生成し識別することが一般的である。しかしながら、前章で述べたように音素毎に音響データベースを作成するために膨大な計算量を必要とすることに

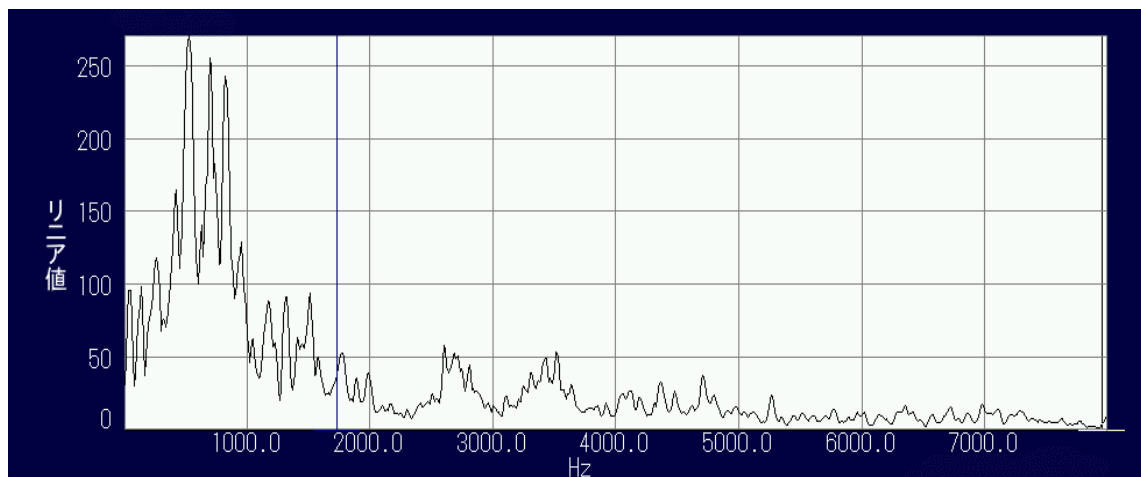
加え、正確にホルマントを抽出することが難しいことが報告されている^[9]。そこで本研究では指定単語「こうちこうかい」の音声をFFT処理したピーク波形に、個人毎の特徴があることに着目する。したがって、46.4ms毎にFFT処理した各周波数におけるピークを、NNへ入力データとする。図3.5に各被験者の発話ピーク波形を示す。



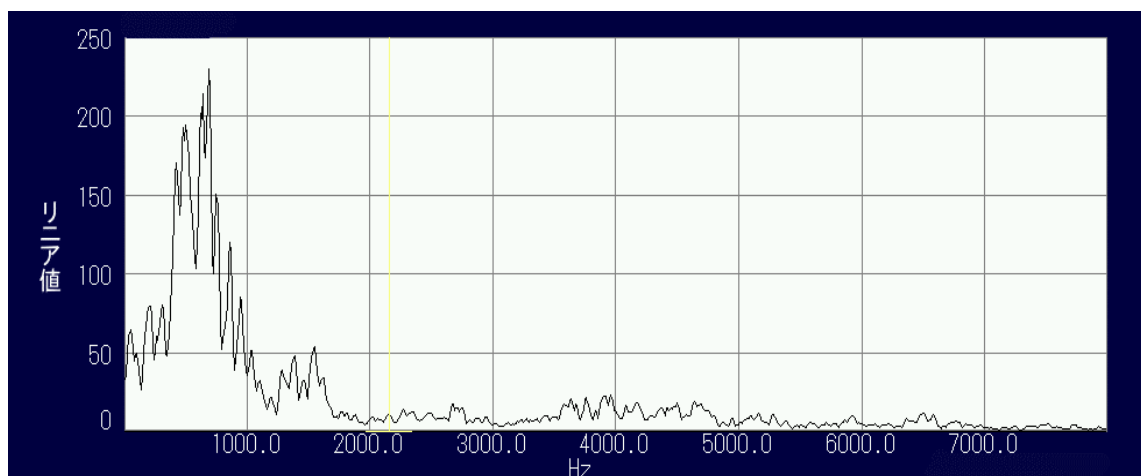
(a) 被験者A (21歳日本人男性)



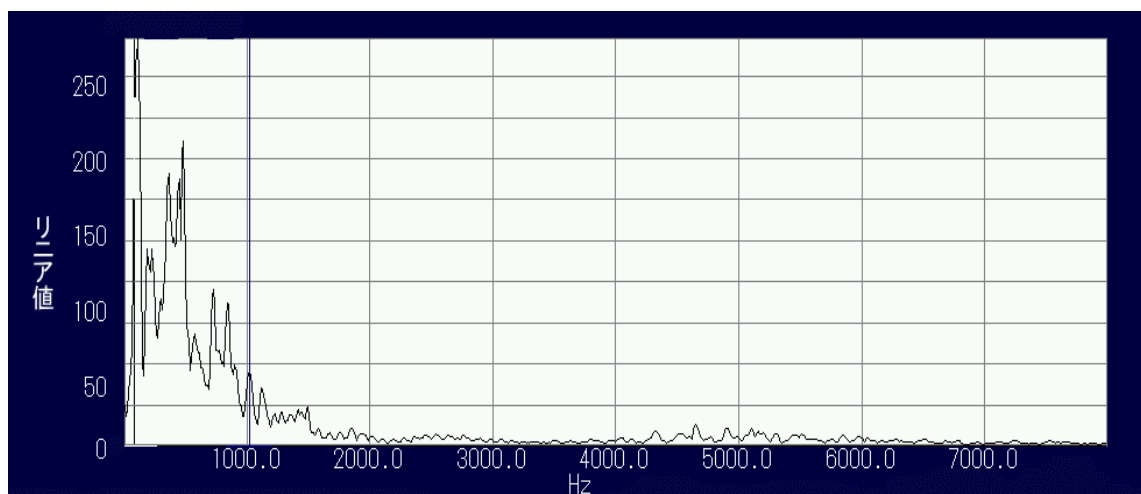
(b) 被験者B (21歳日本人男性)



(c) 被験者C (23歳日本人男性)



(d) 被験者D (22歳タイ人女性)



(e) 被験者E (21歳日本人男性)

図3.5 発話ピーク波形

発話時間における各周波数のリニア値のピーク値から、NNの入力細胞数分データ抽出する手法について記述する。本研究では、NNの入力細胞数を80とする。入力細胞数を80とする理由は、16ビットCPUの計算機でのメモリ空間上で用意できる最大のNNの構成が80×80×80であるためである。本研究では、200Hzから6000Hzの範囲でFFT処理を行うために、データ数が539になる。入力細胞数80に合わせるためにFFT値を7加算平均した値を一つの入力値とする。末尾の余った入力細胞には、ダミー入力値として1.0を入力し80次元入力ベクトルを作成し、入力細胞に対応させる。ここで、NNへの入力値をスラブ値と記述する。

本研究で使用するNNは元来、紙幣識別用に開発された学習・識別システムである。このため、NNは入力データとして前項により算出したスラブ値、金種、パターン番号などを記述したヘッダ情報、紙幣画像データを要求する。しかしながら、音声データには画像データは不必要である。そのため、学習・識別システムの汎用性を保つためにダミーデータとして領域の確保のみを行う。表3.1に音声データ1サンプルのフォーマットを示す。

表3.1 NNデータ1サンプルのフォーマット

ヘッダ情報	64 byte	
ダミーデータ	400 byte	
スラブ値データ	入力層細胞数 80	320 byte

1 データの構成はヘッダ情報に 64 byte, ダミーデータに 400 byte を確保する。スラブ値はプログラム中、float 型 (4 byte) で宣言されている。スラブ値データには入力層細胞数が 80 であるため 80 個×4 byte=320 byte を確保している。表 3.2 には、ヘッダ情報のフォーマットを示す。図 3.6 にデータ 1 サンプルのヘッダ部分の構成を示す。

表3.2 ヘッダ情報のフォーマット

予約領域 1

(4 byte)

データ 1枚分の総サイズ

(4 byte)

パターン番号

(1 byte)

金種

(1 byte)

方向

(1 byte)

国情報

(1 byte)

通し番号

(2 byte)

機種

(1 byte)

号機

(1 byte)

種別

(1 byte)

センサ番号

(1 byte)

画像のXサイズ(画素数)

(2 byte)

画像のYサイズ(画素数)

(2 byte)

センサ有効域チャンネル番号(L)

(2 byte)

センサ有効域チャンネル番号(H)

(2 byte)

予約領域 2

(2 byte)

センサ分解能(X方向)

(1 byte)

センサ分解能(Y方向)

(1 byte)

予約領域 3

(6 byte)

紙幣の中心画素のX座標

(2 byte)

紙幣の中心画素のY座標

(2 byte)

画像データのオフセット

(4 byte)

予約領域 4

(4 byte)

スラブ値の種類数

(1 byte)

1種類目のスラブ値のマスクID

(1 byte)

1種類目のスラブ値の個数

(1 byte)

予約領域 5

(5 byte)

スラブ値のオフセット

(4 byte)

予約領域 6

(4 byte)

- ・ 予約領域

学習システムを拡張するために用意する.

- ・ データ 1 枚分の総サイズ

NN データの総サイズを示す. 本研究の NN のデータサイズは, 上記で示したヘッダ情報 64 byte, 画像データ 400 byte, スラブ値データ 320 byte で計 784 byte である.

- ・ 画像フレームの X サイズ, Y サイズ

画像データの大きさ.

- ・ 画像データのオフセット

ヘッダの先頭を 0 バイト目とする際, 画像データの先頭からのバイト数.

- ・ 1 パターン目のスラブ値の数

ニューラルネットへの入力層数.

- ・ スラブ値データのオフセット

ヘッダの先頭を 0 バイト目とする際、スラブ値データの先頭からのバイト数.

作成した発声パターンごとのデータの集合をデータファイルとし、同被験者発声パターンを 1 ファイルに管理する. データファイルの構成を図 3.6 に示す.

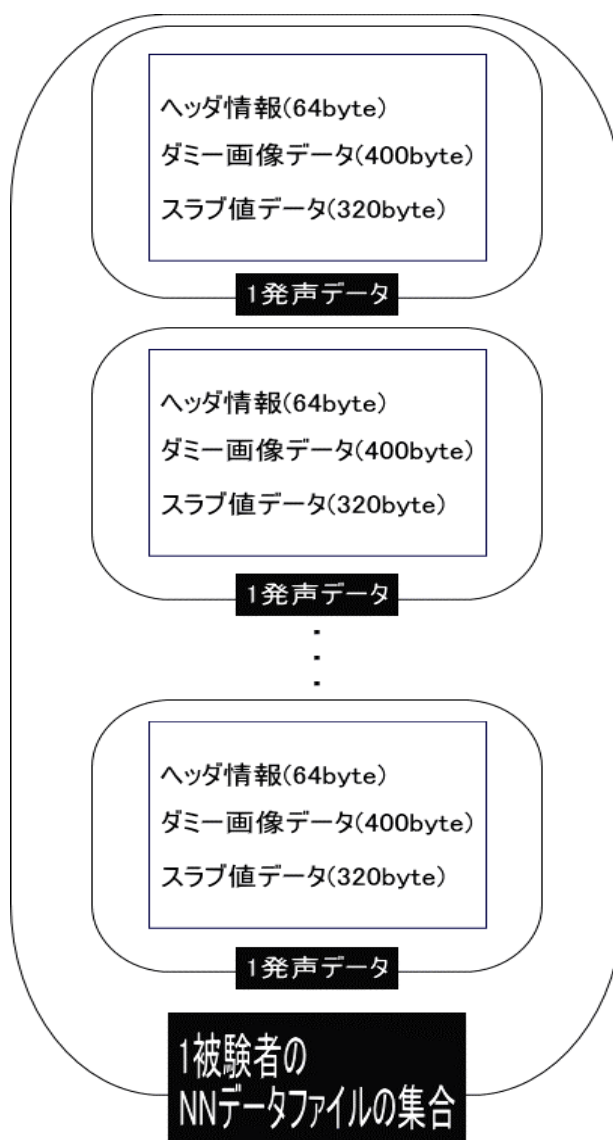


図 3.6 データファイルの構成

3.4 登録, 識別部

本節では、取得した各パターンの音声データを認識する上で使用するニューロ学習・評価システムおよび NN の構成について記述する。

ニューロ学習システム(以下、学習システムと略記)は、後述の改良型逆誤差伝搬法を学習アルゴリズムとするバッチファイルである。学習システム全体の動作項目を図 3.7 に示す NN パラメータ設定ファイルの内容から学習データの抽出を行う。次に学習ファイルを作成するスラブファイルを作成するために、学習データファイルのスラブ値を抽出し学習収束を安定させ、かつ収束条件に早く到達するためにスラブファイルの並び替えを行う。そして、NN パラメータ設定ファイルで指定されたパターン数に従って教師ファイルを作成し、学習を行う。

入力層細胞数	学習定数
中間層数	慣性定数
中間層細胞数	振動定数
出力層細胞数	最終誤差判定値
学習パターン数	最大学習回数
1 パターンの提示(学習)枚数	

図 3.7 NN パラメータ設定ファイルの内容

最急降下法の収束を早めるために上記に BP 法に現在の重みの変化を考慮し、慣性項と振動項を付加する。本論文では式(3.2)を NN の学習アルゴリズムとして採用する。以降、改良型 BP 法と記述する。

$$\Delta W_{jk}(t) = -\eta \delta_k O_j + \alpha \Delta W_{jk}(t-1) + \beta \Delta W_{jk}(t-2) \quad (3.2)$$

ここで t は学習回数、 α は慣性項の比例定数、 β は振動項の比例定数を示す。慣性項により学習初期には誤差が局所的に増加するような方向へも変化が起こり、その後全体的な

最小値へ向かうことが期待できるからである. すなわち, 学習初期でネットワークがローカルミニマムに与えられる機会を減少させることができる. また, 振動項は極小値から脱出させる働きをする. これは Asakawa らがバネ-ダッシュポット系として振動項を導出し, 図 3.8 に示す慣性項・振動項の境界条件を設定した^[11]. したがって, 慣性項・振動項について, その領域内で線形計画法と同じような形で任意の組み合わせをその範囲内から試行錯誤的に選択すればよい. この手法を用いて, 竹田らは紙幣識別において, 最も早く安定した状態で学習収束状態に到達した組み合わせ ($\alpha = 0.95$, $\beta = -0.1$) を算出した^[12]. 本研究ではこの組み合わせを採用し, 学習を行う. 上記の改良型誤差伝搬法を用いて, NN ウェイトを作成する. 学習プログラム, は NN パラメータ設定ファイルに記述されている最終誤差判定値, または学習回数のどちらかの条件を満たした時点で終了する. そして, 学習により作成した NN ウェイトを用いて, 未学習データを評価し, 識別を行う.

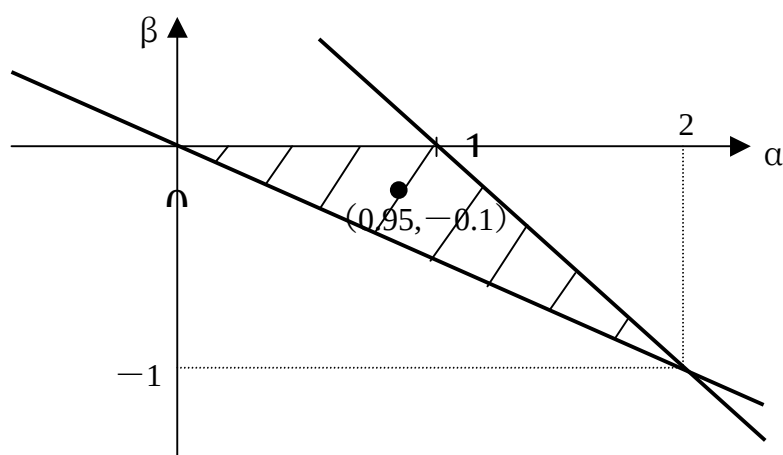


図 3.8 慣性定数と振動定数の範囲

4 実験

本章では, NN 出力層に 2 つのユニットを設定するニューロテンプレートマッチング方式を検討する. 出力層ユニットの一つ目は, 目的パターンの教師データ提示に, 反応するものであり, もう一つは目的外のパターンに反応するものである(以下 NNT と略記). 図 4.1 に NNT の構成を示す. NNT を用いて提案手法による音声識別の学習データと未学習データによる識別実験をシミュレーションで実行する. NNT と従来の階層型 NN (以下 NNC と略記) による, 識別性能を比較検討する. ここで, NNC の出力細胞には被験者毎の音声データを割り当てる. 図 4.2 に NNC の構成を示す.

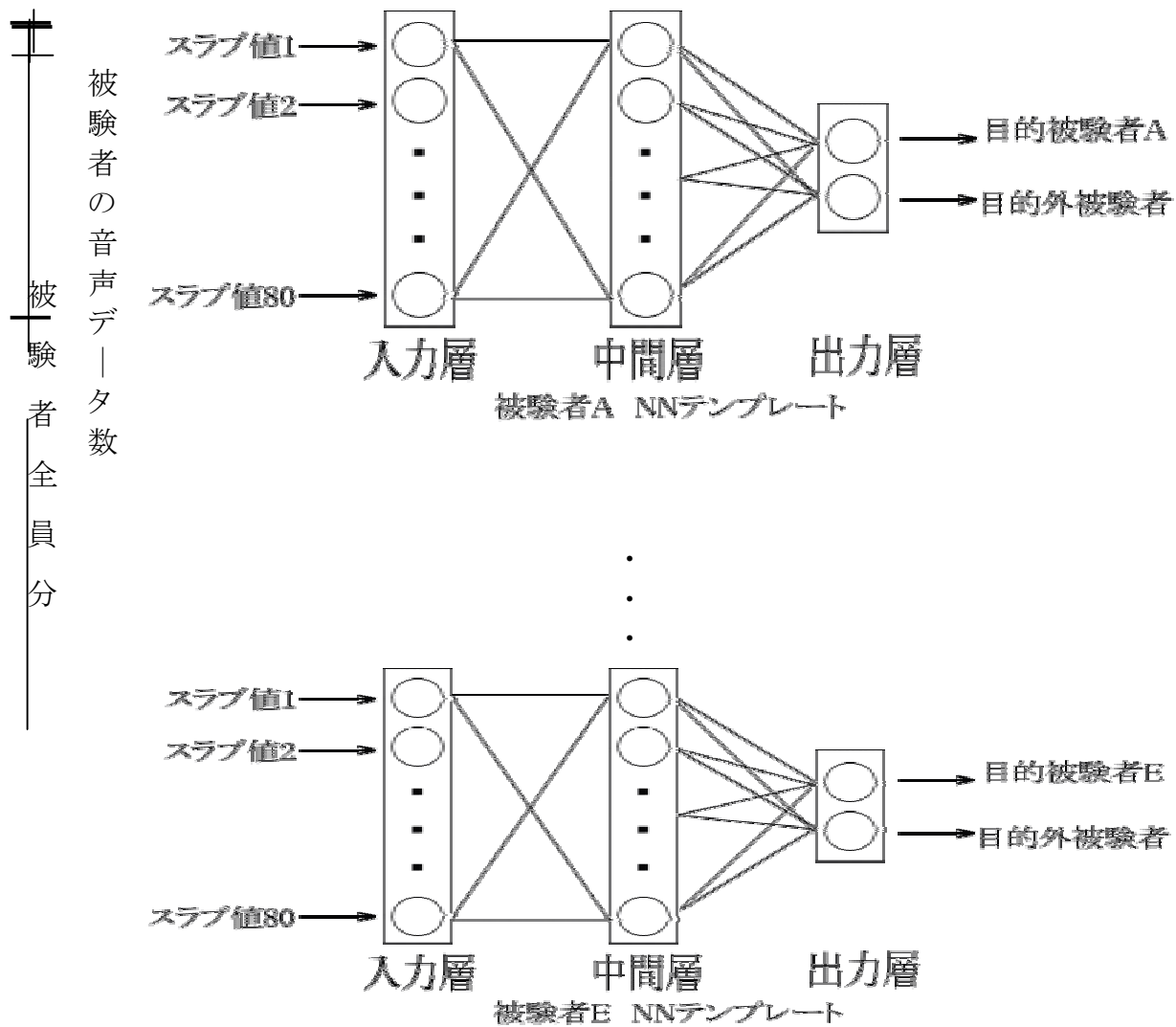


図 4.1 NNT の構成

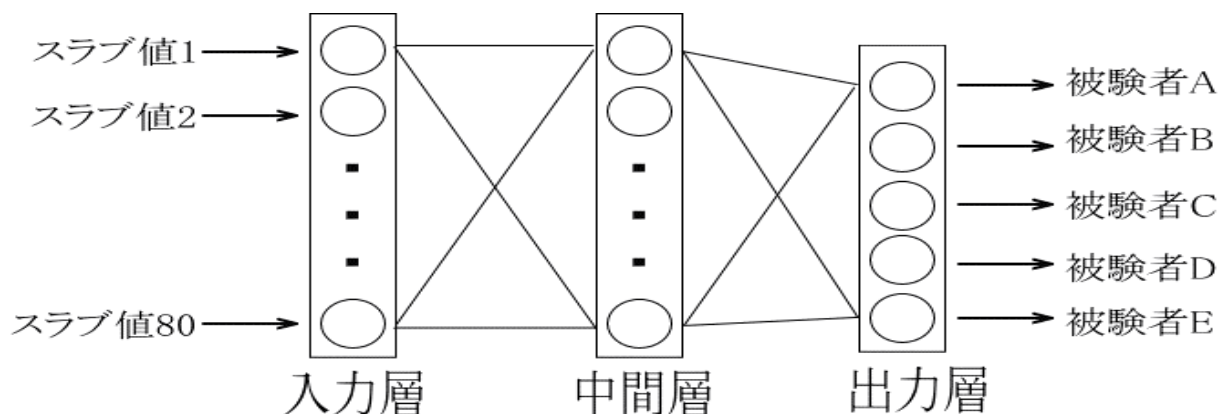


図 4.2 NNC の構成

(実験 1) NNT を用いて、被験者 A の音声データを学習データとし、被験者 A のみ学習データおよび、被験者 A 以外の被験者 4 人の識別対象データ(評価データ)との識別実験を行う。また、同様に被験者 B~E を学習データとし、他被験者との識別実験を行う。実験条件を表 4.1, 実験結果を表 4.2 に示す。

(実験 2) NNC を用いて、実験 1 と同実験条件で識別実験を行う。実験条件を表 4.1, 実験結果を表 4.3 に示す。

識別率は、本人識別率を式(4.1)に示す通り、入力パターンが目的被験者パターンに最大に反応を示したものを正解とする。理想値は 100%となる。本人以外誤識別率は、式(4.2)に示す通り、目的テンプレートに目的外被験者データを提示した際に、間違って目的被験者パターンとして、出力されてしまったパターンを誤りとする。理想値は 0%となる。

$$\text{本人識別率} = \frac{\text{目的被験者パターンが最大発火した個数}}{\text{評価データ数}} \times 100 \quad (4.1)$$

$$\text{本人以外誤識別率} = \frac{\text{目的被験者パターンに誤って最大発火した個数}}{\text{評価データ数}} \times 100 \quad (4.2)$$

表 4.1 実験条件

学習データ		被験者 A～E 各 5 サンプル
評価データ		被験者 A～E 各 10 サンプル (学習データ以外のデータ)
NN 構成	NNT	入力細胞数 80 中間細胞数 80 出力 2
	NNC	入力細胞数 80 中間細胞数 80 出力 5
学習アルゴリズム		改良型誤差伝搬法 式(3.2)
学習定数 η		0.05
慣性定数 α		0.95
振動定数 β		-0.1
最大学習回数		20000
最終誤差判定値		0.0001

表 4.2 実験 1 の実験結果 (NNT)

(a) 目的データ:被験者A

出力	評価データ	被験者 A	被験者 B	被験者 C	被験者 D	被験者 E
正解数 (10)		10				
誤識別数 (各 10)			8	4	0	1
本人識別率 (*1)		100.0%				
平均本人以外誤識別率 (*2)		32.5 %				

(*1) 理想値 100 % (*2) 理想値 0 %

(b) 目的データ:被験者 B

評価データ	A	B	C	D	E
出力					
正解数 (10)		10			
誤識別数 (各 10)	0		0	0	0
本人識別率	100.0 %				
平均本人以外誤識別率	0.0 %				

(c) 目的データ:被験者 C

評価データ	A	B	C	D	E
出力					
正解数 (10)			10		
誤識別数 (各 10)	1	0		0	1
本人識別率	100.0 %				
平均本人以外誤識別率	5.0 %				

(d) 目的データ:被験者 D

評価データ	A	B	C	D	E
出力					
正解数 (10)				10	
誤識別数 (各 10)	2	0	0		2
本人識別率	100.0 %				
平均本人以外誤識別率	10.0 %				

(e) 目的データ:被験者 E

出力 評価データ	A	B	C	D	E
正解数 (10)					9
誤識別数 (各 10)	2	5	0	0	
本人識別率	90.0 %				
平均本人以外誤識別率	17.5 %				

(f) 実験 1 実験結果平均

平均本人識別率	98.0 %
平均本人以外後識別率	11.0 %

表 4.3 実験 2 実験結果 (NNC)

評価データ数各 10 のうちの発火数

評価データ 学習データ	A (10)	B (10)	C (10)	D (10)	E (10)	個人識別率 (理想値 100%)
被験者 A	10	0	0	0	0	100.0 %
被験者 B	2	8	0	0	0	80.0 %
被験者 C	0	0	10	0	0	100.0 %
被験者 D	0	0	0	10	0	100.0 %
被験者 E	0	0	1	0	9	90.0 %
平均識別率 (*3)						94.0 %
平均誤識別率 (*4)						6.0 %

(*3) 理想値 100 % (*4) 理想値 0 %

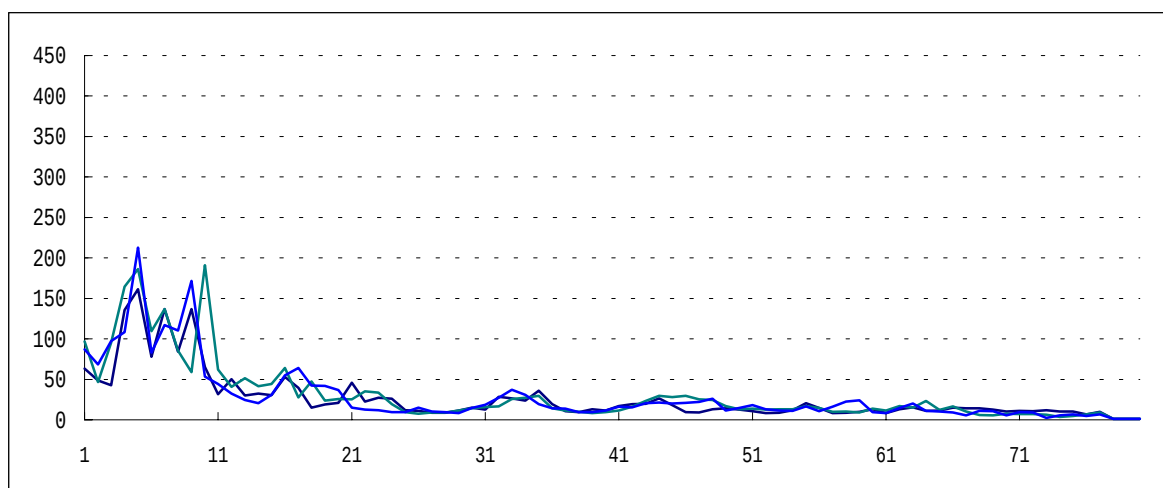
実験 1, 2 において, 提案手法により個人毎の特徴抽出が行えた. これにより, 実用性能

(90%程度)の良好な結果が得られることが判明した。しかしながら、本人外誤識別率に関しては平均 10%程度、学習データによっては 37.5%の誤識別結果となった。この問題の発生要因の一つとして、学習データ数が 5 サンプルと少ないことから、入力値データの幅が広く他被験者に誤識別したと考えられる。

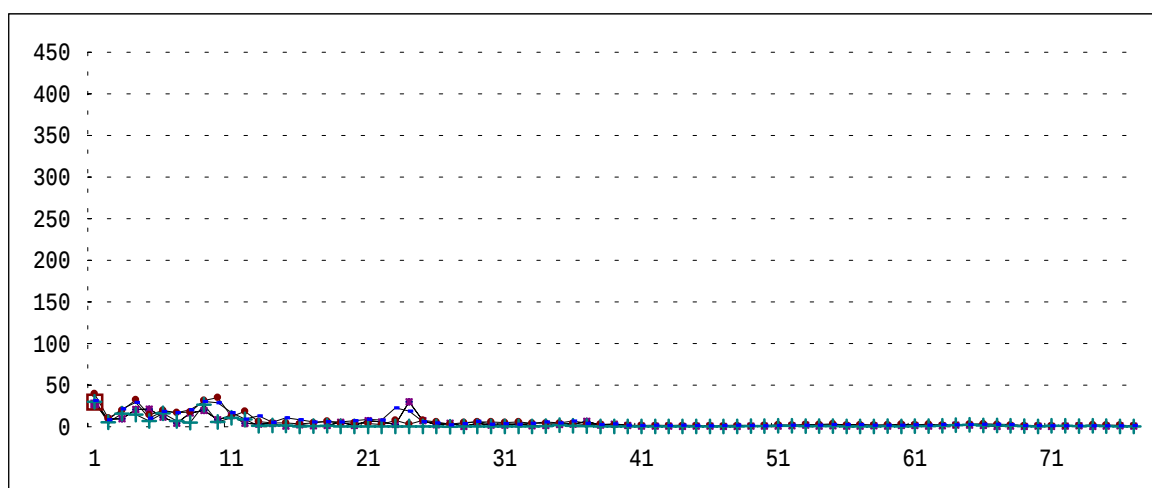
そのため検証用に各被験者スラブ値の平均値からどの程度、偏在しているかを調べるために、評価データの分散値 式(5.1)を求める。図 5.1 に各被験者のスラブ値 5 パターン分の分散値グラフを示す。グラフは、縦軸に分散値、横軸に NN の入力細胞数をとる。各被験者の分散値の平均を表 5.1 に示す。

$$\text{分散値} = \frac{n \sum x^2 - (\sum x)^2}{n^2} \quad (5.1)$$

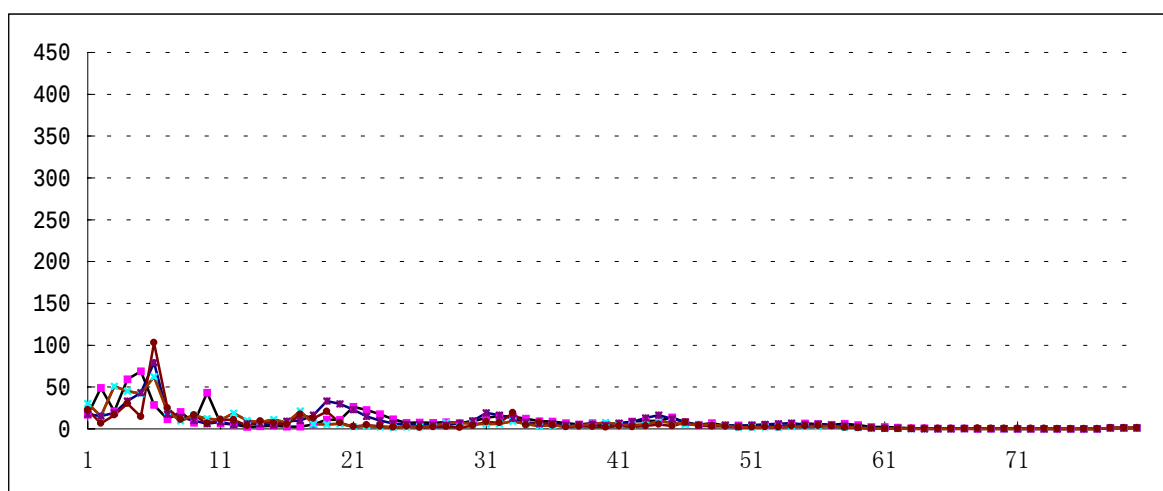
n は平均値, x は 1 スラブ値を示す。



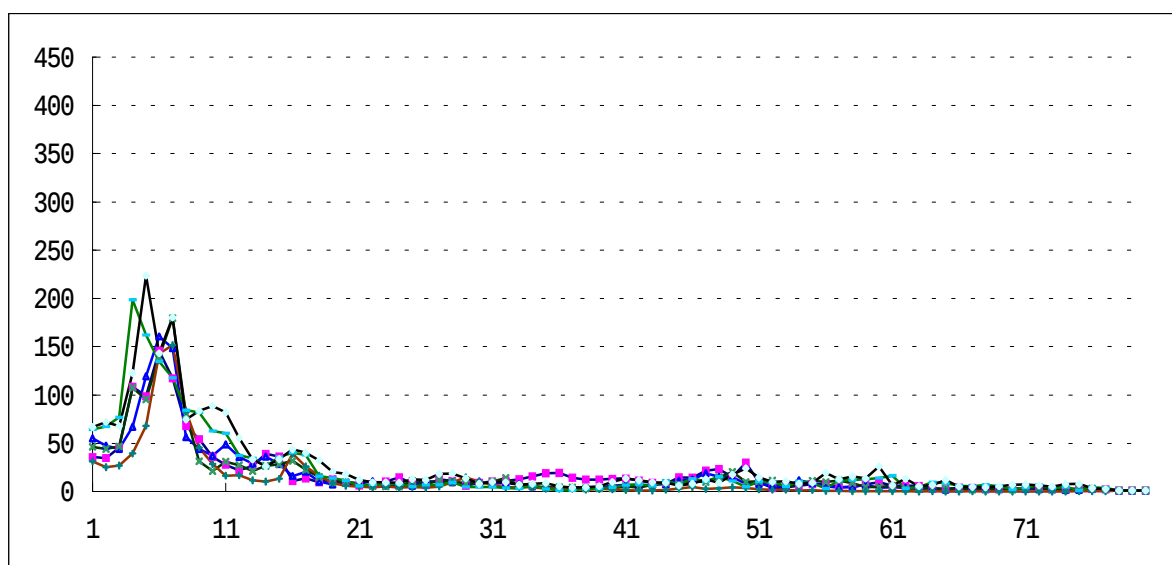
(a) 被験者 A



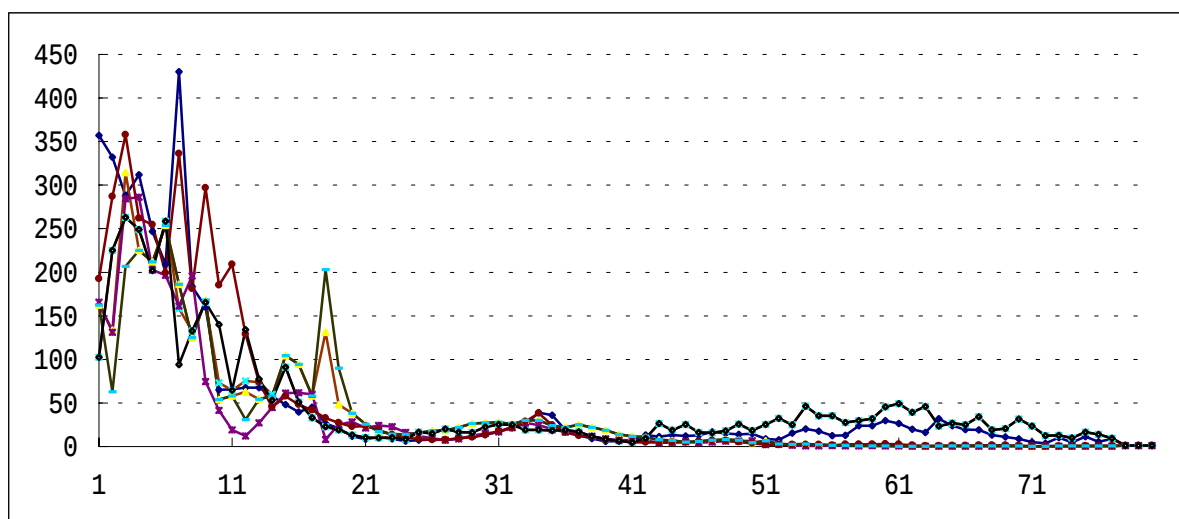
(b) 被験者 B



(c) 被験者 C



(d) 被験者 D



(e) 被験者 E

図 5.1 各被験者のスラブ値の分散値グラフ

表 5.1 各被験者の分散値平均

被験者	A	B	C	D	E
分散値平均	382.0	23.1	91.4	261.5	414.4

表 5.1 より, 被験者 A と被験者 E においてスラブ値が他被験者に誤識別した要因として, スラブ値の分散値が大きく, 入力ベクトルの範囲が広いために, 特徴領域が他被験者と重複したためと推察される. また, 学習データ数が少ないことから, 同一被験者において平均音声波とは異なる音声波が誤差となり, 十分な特徴抽出, および学習データの構築ができなかったためと考えられる.

5 まとめ

本研究では、文字による ID, Password 方式に変わる利便性の高い個人認証技術として、人が日常のコミュニケーションに使用する「音声」に着目し、指定単語を発話することによる話者識別システムを検討した。その中で、音声認識に一般的に用いられる HMM を用いた手法ではなく、識別性能の向上、処理時間短縮のために、音声波の登録、評価に NNC, NNT を用いた手法の検証を行った。その結果、本人認証に有効であることを示した。

今後の課題として、NN の学習データのサンプル数を増やし、被験者毎に十分な特徴抽出を行う事に加え、本人以外誤識別率低下を図る必要がある。また、実用化においてもっとも有効な使用場面として、携帯電話による個人識別が考えられる。しかしながら、携帯電話を通した音声は、低ビットレートであるため、提案システムでは被験者毎に特徴抽出を行うのが難しいと考えられる。低ビットレートにおいても十分な識別結果が得られるような特徴抽出法を検討することが今後の課題である。

6 謝辞

本研究の遂行ならびに、本論文の執筆にあたり的確なる、ご指導と校閲を賜りました高知工科大学情報システム工学科 竹田史章教授に深甚なる謝意を表します。また、音声録音に協力していただいた 34 人の学部生に心から感謝します。さらに、本研究に関して、豊富な知識の元にニューラルネットワーク、アルゴリズム、数学的な知識の指導していただいた修士 2 年中原昌樹氏、平山純一郎氏にお世話になりました。最後に、本研究を行う機会を与えていただいた高知工科大学に心から御礼申し上げます。

7 参考文献

- [1] 中島弘喜, 竹田史章, 中原昌樹: “ニューラルネットワーク応用による筋活動電位パターン識別”, 高速信号処理応用技術学会春季研究会, P.67-68, 2002
- [2] 堂下修司, 白井克彦, 溝口理一郎, 新美康永, 田中穂積: “音声による人間と機械の対話”, オーム社, P.39-92, 1998
- [3] 堂下修司, 白井克彦, 溝口理一郎, 新美康永, 田中穂積: “音声による人間と機械の対話”, オーム社, P.39-92, 1998
- [4] 中川聖一, 鹿野清博, 東倉洋一: “ニューロサイエンス&テクノロジーシリーズ 音声・聴覚と神経回路網モデル”, オーム社, P.45-48, 1990
- [5] 竹田史章, 西蔭紀洋: “紙幣用ニューロテンプレートマッチング識別手法の開発”, 電気学会論文誌, Vol.112-C, No.1, P196-205, 1992
- [6] 中川聖一, 鹿野清博, 東倉洋一: “ニューロサイエンス&テクノロジーシリーズ 音声・聴覚と神経回路網モデル”, オーム社, P.12-17, 1990
- [7] 鹿野清博, 伊藤克亘, 河原達也, 武田一哉, 山本幹雄: “IT Text 音声認識システム”, オーム社, P.17-20, 2001
- [8] 鹿野清博, 伊藤克亘, 河原達也, 武田一哉, 山本幹雄: “IT Text 音声認識システム”, オーム社, P.1-15, 2001
- [9] 高木一幸, 板橋秀一: “スペクトルモーメントを利用した音声のホルマント周波数推定法”, 信学技報, SP88-80 (1988)
- [10] 入野俊夫, 河原英樹: “神経回路網による母音認識—特徴抽出能力の検討—”, 日本音響学会聴覚研究会資料, H87-52, 1987
- [11] 竹田史章, 大松繁, 井上卓, 尾波宰三: “フーリエ変換を前処理とするニューラルネットワークによる紙幣識別”, システム制御情報学会論文誌, Vol.5, No.7, pp.265-273, 1992

[12] 竹田史章, 大松繁, 井上卓, 尾波宰三, 小西健一:“ニューラルネットワークを用いた高速搬送紙幣の識別”, 電気学会論文誌, Vol.112-C, No.4, pp.249-258, 1992

<付録> ニューロヘッダ付加プログラム

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <conio.h>
```

```
#include <string.h>
```

```
#define LOWPASSFREQ 100
```

```
#define HIGHPASSFREQ 1060
```

```
#define CELLNO 80
```

```
#define MAXCHNO 1 //CH 数
```

```
#define DATASIZE 80
```

```
/*ワークディレクトリの指定ファイル*/
```

```
#define NNWORKDIR_FILE "..¥¥workdir.cfg"
```

```
#define MAKECFG_FILE "makeldb.cfg"
```

```
char nnworkpath[128]; //ワークディレクトリのパスを記憶
```

```
// 出力ファイルサイズ確認のための変数設定
```

```
##define BFSIZE 10000
```

```
// char buffer[BFSIZE];
```

```
unsigned long id1 = 0; // 予約領域1
```

```
unsigned long id2 = 784; // このデータ1枚分のサイズ
```

```
unsigned char id3 = 1; // パターン番号
```

```
unsigned char id4 = 1; // 機種
```

```
unsigned char id5 = 1; // 方向
```

```
unsigned char id6 = 1; // 国情報
```

```
unsigned short id7 = 1; // 通し番号
```

```
unsigned char id8 = 1; // 機種
```

```
unsigned char id9 = 1; // 号機
```

```
unsigned char id10 = 0; // 種別(0:元画像 1:補正後)
```

```
unsigned char id11 = 0; // センサ番号
```

```
unsigned short id12 = 20; // 画像フレームのXサイズ
```

```
unsigned short id13 = 20; // 画像フレームのYサイズ
```

```
unsigned short id14 = 4; // センサ有効域チャンネル番号(L)
```

```
unsigned short id15 = 201; // センサ有効域チャンネル番号(H)
```

```
unsigned short id16 = 0; // 予約領域2
```

```
unsigned char id17 = 10; // センサ分解能X
```

```
unsigned char id18 = 40; // センサ分解能Y
```

```
unsigned char id19 = 0; // 予約領域3
```

```
unsigned short id20 = 0; // 中心画素のX座標
```

```
unsigned short id21 = 0; // 中心画素のY座標
```

```
unsigned long id22 = 64; // ダミーデータのオフセット
```

```
unsigned long id23 = 0; // 予約領域4
```

```
unsigned char id24 = 1; // 格納されているスラブ種類数
```

```
unsigned char id25 = 1; // 1種類目のスラブ値のマスクid
```

```
unsigned char id26 = 80; // 1種類目のスラブ値の数
```

```
unsigned char id27 = 0; // 予約領域5
```

```
unsigned long id28 = 464; // スラブ値データのオフセット
```

```
unsigned long id29 = 0; // 予約領域6(id1~id29=ヘッダ)
```

```

float d30 = 1;    // ダミーデータ4byte

int read_workdir()
{
    char *nnworkdir = NNWORKDIR_FILE;
    FILE *fp_nnwork;

    if ((fp_nnwork = fopen( nnworkdir, "rt" )) == 0){
        printf (" workdir.cfg FILE open error !! ¥n");
        return 1;
    }

    if ( fscanf( fp_nnwork , "%s", nnworkpath ) == -1){
        printf (" workdir.cfg READ error !! ¥n");
        return 1;
    }
    fclose(fp_nnwork);
    return 0;
}

int fusion_path(char *infile, char outfile[])
{
    sprintf (outfile, "%s¥¥¥s", nnworkpath, infile);
    return 0;
}

```

```

int datatrans(char infilename[], char outfile[], int patNo, int fileNo)
{
    int      ChNo, dataNo, SlabNo;
    int      sumfreqno = 0, minfreqno = 0, maxfreqno = 0;
    float     fdata[DATASIZE], slab[CELLNO];
    FILE      *infilept, *outfilept;

    // 必要なヘッダ情報の書き込み
    id3 = (unsigned char) patNo; //パターン番号
    id4 = (unsigned char) patNo; // 金種
    id5 = 1;                      // 方向
    id7 = (unsigned short)  fileNo; // 通し番号
    id11 = (unsigned char) fileNo;  // センサ番号

    // 入力ファイルのオープン
    if ((infilept = fopen(infilename, "rb")) == NULL){
        printf("%s: can't open file ! ¥n", infilename);
        return(-1);
    }

    // FFT ファイル(ASCII)からデータ読み出し ---> fdata[5, DATASIZE]
    for(dataNo = 0; dataNo <= (DATASIZE - 1); dataNo++){

        fscanf(infilept, "%f", &fdata[dataNo]);
        //printf("fdata[%d]--

```

```

-%f¥n",dataNo,fdata[ChNo][dataNo]);
    }
    //printf("fdata[0]---%f¥n", fdata[0]);
    //printf("fdata[1]---%f¥n", fdata[1]);
// 入力ファイルのクローズ
    fclose(infilept);

// 出力ファイルオープン
    outfilept = fopen(outfilename,"ab");

// ヘッダ情報のファイル出力
    fwrite(&id1,4,1,outfilept); //4byte //予約領域1
    fwrite(&id2,4,1,outfilept); //4byte //このデータ(1枚分の総サイ
ズ)

    fwrite(&id3,1,1,outfilept); //1byte //パターン番号
    fwrite(&id4,1,1,outfilept); //1byte //金種
        ---- 10
    fwrite(&id5,1,1,outfilept); //1byte //方向
    fwrite(&id6,1,1,outfilept); //1byte //国情報
    fwrite(&id7,2,1,outfilept); //2byte //通し番号
    fwrite(&id8,1,1,outfilept); //1byte //機種
    fwrite(&id9,1,1,outfilept); //1byte //号機
    fwrite(&id10,1,1,outfilept); //1byte //種別
    fwrite(&id11,1,1,outfilept); //1byte //センサ番号

```

```

fwrite(&id12,2,1,outfilept); //2byte //画像のXサイズ(画素数) ---
- 20
fwrite(&id13,2,1,outfilept); //2byte //画像のYサイズ(画素数)
fwrite(&id14,2,1,outfilept); //2byte //センサ有効域チャンネル番号
(L)
fwrite(&id15,2,1,outfilept); //2byte //センサ有効域チャンネル番号
(H)
fwrite(&id16,2,1,outfilept); //2byte //予約領域2
fwrite(&id17,1,1,outfilept); //1byte //センサ分解能(X方向)
fwrite(&id18,1,1,outfilept); //1byte //センサ分解能(Y方向) --
-- 30
fwrite(&id19,1,6,outfilept); //6byte //予約領域3
fwrite(&id20,2,1,outfilept); //2byte //紙幣の中心画素のX座標
fwrite(&id21,2,1,outfilept); //2byte //紙幣の中心画素のY座標
---- 40
fwrite(&id22,4,1,outfilept); //4byte //ダミーデータのオフセット
fwrite(&id23,4,1,outfilept); //4byte //予約領域4
fwrite(&id24,1,1,outfilept); //1byte //スラブ値の種類数
fwrite(&id25,1,1,outfilept); //1byte //1種類目のスラブ値のマス
クID---- 50
fwrite(&id26,1,1,outfilept); //1byte //1種類目のスラブ値の個数
fwrite(&id27,1,5,outfilept); //5byte //予約領域5
fwrite(&id28,4,1,outfilept); //4byte //スラブ値のオフセット
---- 60
fwrite(&id29,4,1,outfilept); //4byte ---- 64 (ヘッダの終了)

```



```

        //予約領域6
        fwrite(&id30,4,100,outfilept);//4byte- 464 (ダミーデータの終了)
        //ダミーデータ
        // スラブ値のファイル出力    //320byte 1 セットで 784byte

        for (SlabNo = 0; SlabNo < 80; SlabNo++){
            slab[SlabNo] = fdata[SlabNo] / 100;
            fwrite(&slab[SlabNo], 4, 1, outfilept);
        //
            printf("minfreqno = %d maxfreqno = %d ¥n", minfreqno,
maxfreqno);
        }
        //printf("slab[0] = %f ¥n", slab[0]);
        //printf("slab[1] = %f ¥n", slab[1]);

        // 出力ファイルクローズ
        fclose(outfilept);
        return 0 ;
    }

    main(){
        FILE      *CFGfilept, *outfilep;
        char      *infile, filename[128];
        int        patnum, fnum; //, dammy;
        int        patNo, fileNo;
        char      infilename[100], outfilename[100];

```

定

```

if (read_workdir() == 1) exit(1); //ワークスペースパスのロード

infile = MAKECFG_FILE;
fusion_path(infile,filename); //ワークスペース CFG ファイルを指

CFGfilept = fopen (filename, "r"); //CFG ファイルのオープン

//認識パターン数、結合ファイル数の読み出し --> patnum, fnum
fscanf(CFGfilept, "%d %d¥n", &patnum, &fnum);
printf("結合ファイル数 = %d × %d¥n",patnum, fnum);

for(patNo = 1; patNo <= patnum; patNo++){
    // 出力ファイル名の読み出し --> outfilename[]
    fscanf(CFGfilept, "%s¥n", outfilename);
    printf("出力ファイル名 = %s¥n", outfilename);
    //出力ファイルのオープン(0byte ファイルを作成)
    infile = outfilename;
    fusion_path(infile, filename);
    outfilep = fopen(filename, "wb");
    fclose(outfilep);

    for(fileNo = 1; fileNo <= fnum; fileNo++){
        // 入力ファイル名の読み出し --> infilename[]
        fscanf(CFGfilept, "%s¥n", infilename);

```

```

        // 入力ファイル内容の読み出し-->書き出し
        if(datatrans(infile, filename, patNo, fileNo) == -1){
            printf("file trans error!!");
            exit(1);
        }
        // 入力ファイルのクローズ
        fclose(infile);
    }
    // 出力(結合)ファイルのクローズ
    fclose(outfile);
}

// prm.cfg ファイルのクローズ
fclose(CFGfilept);

printf("End of Job %d\n", jobNo);

// 書き込みファイルサイズのチェックルーチン
//outfile = fopen(outfilename, "rb");
//dammy = fread(buffer, sizeof(char), BFSIZE, outfile);
//printf ("FILESIZE = %d\n", dammy);
//fclose(outfile);
return 0;
}

```