修士論文

マルチ出力サブネットワークを有するディープニューラルネットワークによる声質変換

2015 年 2 月 5 日

指導教員　広瀬 啓吉 教授

東京大学大学院 情報理工学系研究科
電子情報学専攻

48-136429　橋本 哲弥

概要

本研究では，話者性の柔軟な制御に向けての Deep Neural Network(DNN) による声質変換手法を提案する．DNN では，既存手法である Gaussian Mixture Models(GMM) よりも高精度の変換が可能であるという報告もされているが，DNN は各ノード・各レイヤーがどういった情報を扱い，どのような変換を行っているかが不明瞭であるため，GMM のようなパラメータ適応が難しく，柔軟な変換を実現することが難しいという問題がある．この問題を改善する足がかりとして，DNN による声質変換の処理を，浅い層 (入力層に近い層) における話者非依存な特徴量抽出と，深い層 (出力層に近い層) における話者依存の話者性の変換・生成に分離するために，１つの話者非依存サブネットワークと複数の話者依存サブネットワークを持つ DNN を，複数の話者からなるパラレルコーパスによって学習することで，単一の入力層と複数の出力層を持つ DNN を構築した．提案手法によって構築した DNN に対して，学習に使用した話者と学習に使用していない未知話者に対しての変換精度を客観評価指標によって評価した結果，既存手法である GMM と通常の DNN を上回る変換精度が得られた．

目次

図目次

表目次

第1章

序論

1.1　背景・目的

声質変換 (Voice Conversion) は入力されたある話者の発話を，言語情報を損なうことなく他の話者の音声に変換する技術であり，Text-To-Speech (TTS) など多くのアプリケーションに応用されている [1, 2]．声質変換は，入力音声と出力音声の特徴量空間上でのマッピングを構築するというタスクと考えられ, 統計的な変換モデルによる実装がよく用いられている．近年では，統計的手法の中でも Gaussian Mixture Models (GMM)，Artificial Neural Networks (ANN) の 2 手法が広く用いられている [3][4]．これらの手法で変換モデルを学習する際には，パラレルコーパスと呼ばれる，変換元の話者 (ソース話者) と変換先の話者 (ターゲット話者) による同一文の読み上げ音声データが必要となる．

しかし，この方法で構成した変換モデルは，学習に用いていない話者に対しては変換精度が低く，学習に使用した特定の話者間に対してしか用いることができない．そのため，入力・出力話者に対して柔軟な声質変換は現在も多く研究されており，GMM においてはパラメータ適応と呼ばれる手法による一対多および多対一声質変換が提案されている [5, 6, 7]．

一方で，ANN の応用手法であり，近年多分野において研究が行われている Deep Neural Netwotk (DNN) では，GMM よりも高精度の変換が可能であるという報告もされている [8, 9]．しかし，DNN は各ノード・各レイヤーがどういった情報を扱い，どのような変換を行っているかが不明瞭であるため，GMM のようなパラメータ適応が難しく，柔軟な変換を実現することが難しいという問題がある．GMM における固有声変換 [10] やテンソル空間を用いた声質変換 [11] のように，話者性の柔軟な制御を可能とするためには，DNN による声質変換においても変換元・変換先の話者に依存した処理 (speaker-dependent:SD) と各話者に依存しない処理 (speaker-independent:SI) とを分離する必要があると考えられる．

ここで，DNN を用いた自動音声認識において，サブネットワーク構造を持つ DNN によって多言語認識を行う手法が松田らによって提案されている [12]．この手法では，ネットワークの浅い層 (入力に近い層) では言語に非依存な特徴量抽出のような処理が行われており，ネットワークの深い層 (出力に近い層) では言語に依存した識別のような処理が行われているという仮定を置いている．この仮定を基に，ネットワークの浅い層を 1 つの言語非依存サブネットワークによって構成し，深い層を複数の言語依存サブネットワークによって構築することで，浅い層に複数の言語のデータを，深い層に認識する言語に対応したデータをそれぞれ学習させることで，認識率の改善を行っている．

本研究では多言語認識における松田らの手法を参考に，DNN による声質変換の処理を，浅い層における話者非依存な特徴量抽出と，深い層における話者依存の話者性の変換・生成に分離することを試み，それによる変換精度の変化を実験的に検討する．この考えを基に本手法では，DNN に対して変換先の話者毎のサブネットワークを導入した，多対一型の変換手法を提案する．

1.2　本稿の構成

本稿は，全 7 章で構成される．2 章では声質変換の基本的な考え方と統計的声質変換において一般的に用いられている特徴量，データの前処理，そして単純な手法としてコードブックマッピングによる声質変換について示す．3 章では現在最も広く研究されている GMM を用いた声質変換の基本について示し，GMM を用いて柔軟な声質変換を試みた手法としてパラメータ適応，MAP 適応，Eigen-voice conversion (固有声変換) を挙げ，その概要を説明する．4 章では，古くから用いられてきている ANN と，近年研究が進められている ANN の応用手法である DNN，及びその

要素技術について説明し，それらを用いた声質変換手法を挙げる．5 章では，3 章で述べた GMM による声質変換手法と 4 章で述べた DNN による声質変換手法それぞれの利点及び欠点について論じ，提案手法の着想・目的・具体的な手法について説明する．6 章では，提案手法の有効性を示すために行った DNN の pre-training に関する予備的実験，学習データ中の話者に関する提案手法と既存手法 (GMM，DNN) の間での変換精度の比較，及び未知話者のデータに対する提案手法と既存手法の間での精度比較を行い，結果について示す．最後に 7 章では，研究全体のまとめと今後の課題・応用について述べる．

第2章

従来の声質変換

2.1　声質変換の概要

近年，Text-to-Speech システムや自動翻訳システムの発展によって，合成，及び変換音声に対して多様な声質が求められている．医療の分野においても，喉の障害などによってはっきりとした発音ができなくなった人や，人工声帯による発話を行っている人の音声をその人本人の声であり且つ自然な発話となるような音声合成器が必要とされている．このような需要に応える技術として声質変換が存在する．

声質変換は，入力された合成音声や自然音声に対して，言語内容を損なうこと無くその音声の話者性 (個人性，性別など) や感情などの発話様式を他のものに変換する技術のことをいう．音声の声質を決定する要因としては，声道特性と音源特性の 2 つがある．音源特性は声帯の開閉によって発生する振動から生成される音源の特性のことをいい，音声に置ける韻律 (イントネーション) に寄与する．一方で，声道特性は声帯から発せられる音源に対して口腔や鼻腔，舌の動きによって韻律を付加する特性のことをいう．声質変換では，音声波形からこれらの特性を特徴量として抽出し，変換前の話者 (ソース話者) の特徴量から目的とする変換先の話者 (ターゲット話者) の特徴量への変換を行う．すなわち，声質変換は入力音声と出力音声の特徴量空間上でのマッピングを構築するというタスクと考えられる．そのため，声質変換ではソース話者とターゲット話者によって同一の文章を発声してもらい，その特徴量系列の間で変換モデルを学習する統計的手法が用いられている．

2.2　音声特徴量

2.2.1　ソースフィルタモデル

音声の分析は一般的にソースフィルタモデルにしたがって行われる．ソースフィルタモデルは, 音声を声帯の振動 (ソース) による音源特性と声道の形状特性 (フィルタ) による声道特性の 2 つに分けて考えるものであり, 声帯による音源を $s(t)$, 声道の形状特性を $v(t)$ とすると音声 $x(t)$ は以下の式のような関係で表される．

$$|X(\omega)| = |S(\omega)||V(\omega)| \tag{2.1}$$

ここで, $X(\omega)$, $S(\omega)$, $V(\omega)$ は音声, 音源, 声道特性のフーリエ変換である．音声における声質は, 主に声道特性によって表されるものであるため, 声質変換ではこの声道特性をよく表す特徴量を主な特徴量として用いる．

2.2.2　基本周波数 (F0)

音声信号において，母音は周期的な波形を持つ．この周期的な波形の 1 周期を取り出して周期関数と考えることで，以下の式のようなフーリエ級数に展開することができる．

$$x(t) = \sum_{n=0}^{\infty} A_n \cos(2\pi v F_0 t + \theta_n) \tag{2.2}$$

式 (2.2) 中の F_0 を基本周波数という．この基本周波数は音声の高さ (ピッチ) に寄与する値であり，声帯の振動の周期を表す．声質変換において基本周波数の変換は，ソース話者とターゲット話者の声の高さの変換となる．

2.2.3　ケプストラム

一般的に音声波形は初めに得られる時間系列の波形信号から，フーリエ変換によって周波数領域の信号に変換して扱われる．この周波数スペクトルの振幅を絶対値化したものを周波数パワースペクトル (power spectrum) という．

音声信号のパワースペクトルは音声に含まれる韻律情報を担う重要な変量であり，音源特性と音源位置，声道特性，口唇からの放射特性などの総体として出力される特性である．音声信号におけるパワースペクトルの包絡抽出は概ね声道特性の抽出に相当する．また，抽出される包絡は，包絡に対するモデルの違いと元の波形のフーリエ・スペクトルに対する誤差尺度の違いによって僅かに異なるものとなる．

ここにケプストラムという尺度を導入する．ケプストラムは信号波形のパワースペクトルの対数のフーリエ変換として定義される値であり，複数の信号が畳み込みのような形で結合されているような信号の解析に用いられる．ここで，ソースフィルタモデルを考えると，音声 $x(t)$ は声帯による音源 $s(t)$ と声道の $v(t)$ の畳み込みによって以下の式で表現される．

$$x(t) = \int_{-\infty}^{+\infty} s(\tau)v(t-\tau)d\tau \tag{2.3}$$

それぞれのフーリエ変換を考えると，

$$|X(\omega)| = |S(\omega)||V(\omega)| \tag{2.4}$$

ここで対数を取ることで，

$$\log|X(\omega)| = \log|S(\omega)| + \log|V(\omega)| \tag{2.5}$$

フーリエ変換を $\mathcal{F}[\cdot]$ とすると，ケプストラムは以下のように表される．

$$\mathcal{F}[\log|X(\omega)|] = \mathcal{F}[\log|S(\omega)|] + \mathcal{F}[\log|V(\omega)|] \tag{2.6}$$

式 (2.5) のように，対数パワースペクトルでは声道成分を表す $V(\omega)$ に音源特性，すなわち基本周波数成分を表す $S(\omega)$ が足し合わされており，実際のスペクトル上でも声道成分の連続的なスペクトル上に基本周波数による周期的な離散値が重畳しているような形が見られる．この対数パワースペクトルにフーリエ変換を行うことで，連続的なスペクトルを持っていた声道成分は低次のケプストラムとして現れ，離散的なスペクトルを持っていた基本周波数はケプストラム上の対応する周期の位置にピークとして現れる．また，ケプストラムの 0 次元には元の音声波形のパワー成分となる．声質変換ではこのケプストラムが主な変換の対象として扱われ，基本周波数に関してはソース話者とターゲット話者の平均と分散を用いた線形変換で簡易的に行っている研究も多い．

2.2.4　メルケプストラム

式 (2.6) で表されるケプストラムは時間領域で等間隔にサンプリングされた標本値を用いて計算されたものであり，周波数軸尺度は線形である．しかし，人間の聴感覚特性は低周波数の音声に対しては高い分解能，高周波数の音声に対しては低い分解能を持ち，全体的には対数に近い，メル尺度 (mel-scale) と呼ばれる周波数感度となっている．そこで，スペクトルを扱う際にも実際の人間の聴覚特性を表すメル尺度を導入することで，人間の聴覚にとって重要な周波数成分をより

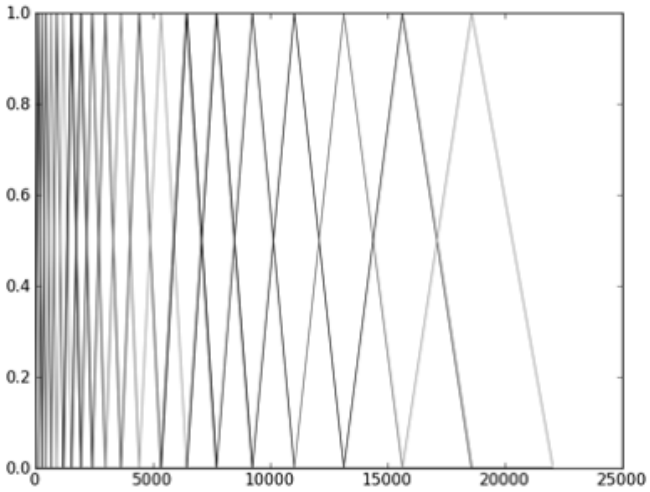

図 2.1: メル尺度に基づく帯域フィルター

重点的に扱うことが考えられる．このメル尺度を導入して計算されたケプストラムのことをメルケプストラムという．

メルケプストラムの計算方法としては，スペクトルを周波数軸上でメル尺度で等間隔になるように再サンプリングし，その標本値を用いてスペクトルを再推定するというものがある．この方法では音声のスペクトルに対し，図 (2.1) のような周波数軸上でメル尺度において等間隔となるような帯域フィルタ群を用いてフィルタリングを行い，その出力に対して離散フーリエ変換を行うことでメルケプストラムを得る．

2.3　統計的声質変換

これまでに示したような，声道特徴量および音源特徴量を変化させることで音声の変換を行うことができる．例として，声道特徴量であるスペクトル包絡を周波数軸上で伸ばすことで音声を子供のような高い音に変化させたり，縮めることで音声を大人の男性のような太い低い声にすることができる．音源特徴量に関しても，基本周波数を高くすれば高い声，低くすれば低い声にでき，非周期成分を大きくすればかすれた声に変換することができる．しかし，これらの変換を話者毎にヒューリスティックに決定するのは，発声に含まれる音素の数及び組み合わせを考えると現実的ではない．そのため，声質変換では一般的に統計的手法を用いた変換処理が用いられている．

統計的声質変換では，ソース話者の特徴量ベクトルを x，ターゲット話者の特徴量ベクトルを y としたとき，x が与えられた時の y の事後確率 $P(y \mid x)$ をモデル化し，生成モデルによって x から y への変換を行う手法と，変換モデルによって x から y への変換を直接行うものとがある．どちらのモデルにおいても，モデルのパラメータの学習には一般的にソース話者とターゲット話者による同一の言語情報を発話した音声データ (パラレルコーパス) を学習データとして使用する．特徴量の変換には Code-book Mapping を用いたもの [13] や Neural Network を用いたもの [4] などが存在するが, 現在は Gaussian Mixture Models を用いたものが主流となっている [11].

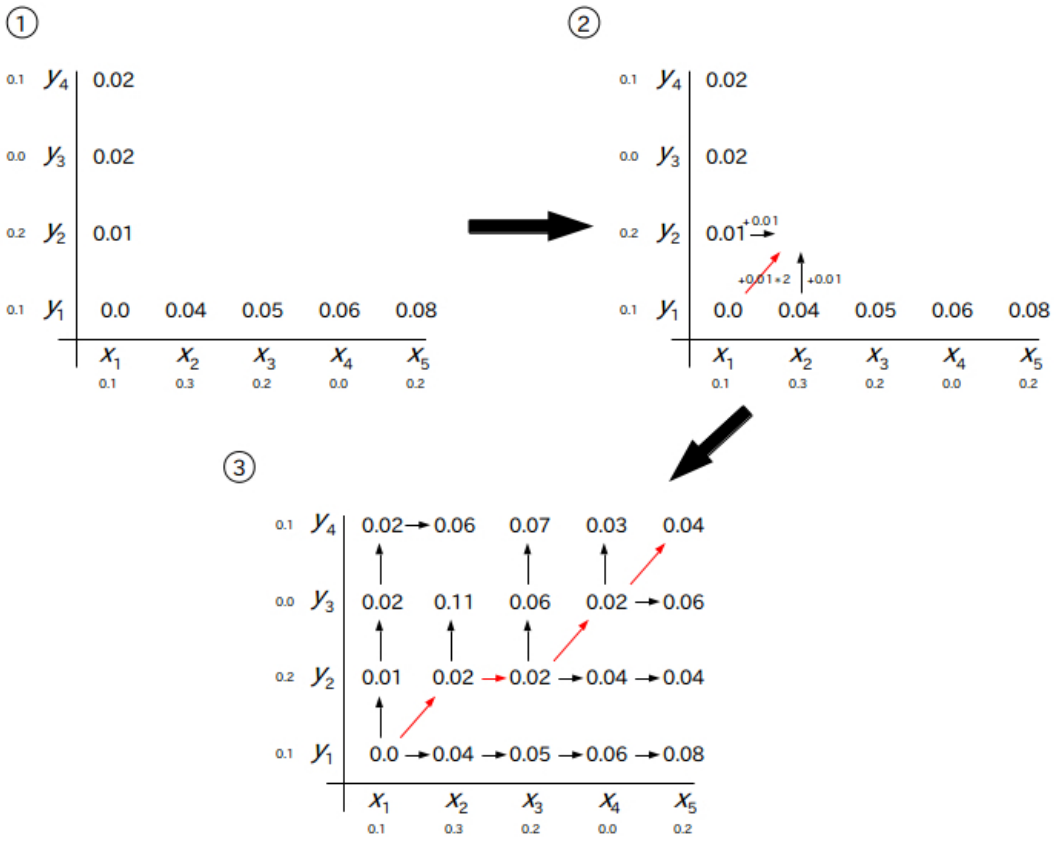

図 2.2: DTW による系列アラインメントの概要

2.4　パラレルコーパス間のアラインメント

一般的な統計的声質変換では，ソース話者とターゲット話者による同一の言語情報を発話したパラレルコーパスによってモデルの学習を行う．しかし，同一の発話内容であっても話者毎に発話速度は異なるため，入力話者の特徴量ベクトル群と出力話者の特徴量ベクトル群は殆どの場合同じデータ数にはならない．そのため，入力話者と出力話者の特徴量間でフレームのアラインメントを考える必要がある．このアラインメントには Dynamic Time Warping (DTW) と呼ばれる動的計画法によるマッチング手法が用いられる．

元々の DTW は長さの異なる 2 つの系列データ間の特徴量を照合し，距離を求めることで系列データの類似度を測るものであり，音声認識などに用いられていた．一方，声質変換を行う際のアラインメントを取るための DTW では，2 つの系列データ (音響特徴量) 間に何らかの距離尺度を導入し，その系列データ間の距離が最小となるように一方の各フレームがもう一方のどのフレームに対応するかを決定する．

図 2.2 に DTW の概要を示す．図 2.2 では簡単のため，特徴量を 1 次元としている．初めに 2 つの系列 (それぞれ長さ m，n とする) をそれぞれ横軸，縦軸とし，$m * n$ 行列の空行列を 2 つ (それぞれ H, C とする) 作る．次に，系列間の全てのフレーム間の距離 D_{ij} を計算し，保持しておく．$(i = 1, 2, ..., m,\ j = 1, 2, ..., n)$ この D_{ij} を基に，$(1, 1)$ から順に，以下の式に従って隣接する座標

から各座標までの最小距離を求め, H_{ij} に保存していく.

$$H_{ij} = \min \begin{cases} H_{i-1,\ j-1} & + & 2D_{ij} \\ H_{i-1,\ j} & + & D_{ij} \\ H_{i,\ j-1} & + & D_{ij} \end{cases}$$

この際，どの式に従って H_{ij} を計算したかを C_ij に保存しておく (図 2.7 の座標中の矢印に相当する)．式 (2.7) 中の $H_{i-1,\ j-1}$ は両系列を伸縮無しに 1 フレームを対応させ，$H_{i-1,\ j}$，$H_{i,\ j-1}$ はどちらかの系列の 1 フレームに複数のフレームを対応させることを意味する．また，$2D_{ij}$ では経路毎に通過する格子点の数が異なってくるため，フレーム間の距離に重みを掛けることで，通過する格子点の数を実効的に等しくしている．全ての座標の H および C を計算した後，座標 (m, n) から C に保持されている移動に従って座標 $(0, 0)$ までの経路を辿ることで，2 系列のアラインメントを得る.

DTW の手順をまとめると以下の様になる.

1. 長さ m，長さ n である 2 つの系列を x 軸・y 軸として置く
2. 2 つの系列の全フレーム間の距離 D_{ij} を計算する $(i = 1, 2, ..., m,\ j = 1, 2, ..., n)$
3. $(x, y) = (1, 1)$ の点を始点とし，$x = 1$ または $y = 1$ である座標の距離とその座標への経路重みから最短経路と累積距離を計算する
4. (3) を x と y を大きくしながら全座標が埋まるまで繰り返す
5. 終点 (座標 (m, n)) から保存しておいた経路を辿ることによって最短経路を求める

2.5　コードブックマッピング

統計的声質変換の初期の 1 手法としてベクトル量子化によるコードブックマッピング法を用いたものがある [13]．この手法は，コードブックのマッピングを生成する学習ステップと，そのマップに従って音声の変換を行う変換ステップという 2 つのステップから構成される.

ベクトル量子化とは，入力ベクトル群にクラスタリングを行った後，各クラスタに含まれるベクトルからそのクラスタのセントロイドを計算し，入力ベクトル群をそのセントロイドで置換するというものであり，元々は情報圧縮などの分野に用いられていた．このときの各クラスタに宛てがわれるセントロイド群をコードブックと言う．コードブックマッピングによる声質変換では，ベクトル量子化における入力ベクトル群を特定話者の音響特徴量とし，そのコードブックが特定話者の個人性を表すと考えている．以下に，その具体的な処理を示す.

初めに，学習ステップについて述べる．学習ステップでは，2 話者の特徴量から張られるベクトル空間を表すコードブックを生成し，その間のマッピング関数を計算する．このマッピング関数をコードブックマッピングという．2 話者の音声特徴量からコードブックマッピングを生成する過程を図 2.4 に示す．詳細な処理は以下のようになる.

1. 2 話者 (ソース話者，ターゲット話者) による同一発話内容からなる単語コーパス (パラレルコーパス) を基に各フレーム毎の特徴量にベクトル量子化を行う
2. コーパス中の各単語に対応する 2 話者の量子化ベクトルに対して DTW によってアラインメントを取る
3. 量子化ベクトルのアラインメントから 2 話者間の対応関係をヒストグラムとして求める

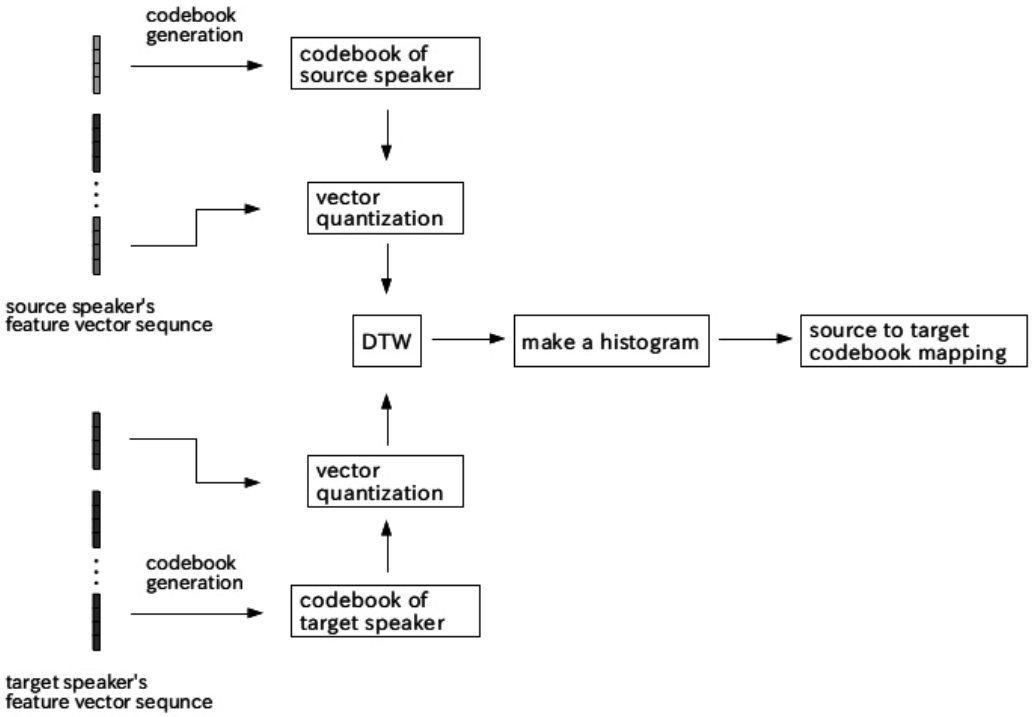

図 2.3: コードブックマッピングにおける学習ステップ

4. 2 話者間の量子化ベクトルのヒストグラムを重み付けと考え 2 話者間の量子化ベクトルのマッピング (コードブックマッピング) を取り，ターゲット話者のコードブックの線形組み合わせでソース話者のコードブックを表現する
5. (2)，(3)，(4) の処理を精度が十分になるまで繰り返す
6. 基本周波数及びパワーに関してはスカラー量子化を行い，同様にコードブックマッピングを学習して求める

次に，変換ステップについて示す．変換ステップでは，生成したコードブックマッピングに従ってソース話者の発話をターゲット話者のものに変換する．概要を図 2.4 に示す．具体的な処理は以下の様になる．

1. 入力として与えられたソース話者の音声特徴量群 (スペクトル，基本周波数，パワー) を線形予測分析によってクラスタリング
2. クラスタリングされた特徴量をコードブックに置換する
3. コードブックに置換されたソース話者の音声特徴量群をコードブックマッピングに従ってターゲット話者のコードブックにデコーディングする

この処理によって得られた変換後の音声特徴量を合成することで，ターゲット話者の音声への変換を実現する．

コードブックマッピングによる声質変換の問題点としては，量子化ベクトルを用いてソース話者，及びターゲット話者の特徴量空間を離散的に表現してしまうために，最終的に合成される変換音声が不連続なものになってしまうというものがある．この点を改善する手法としてファジー

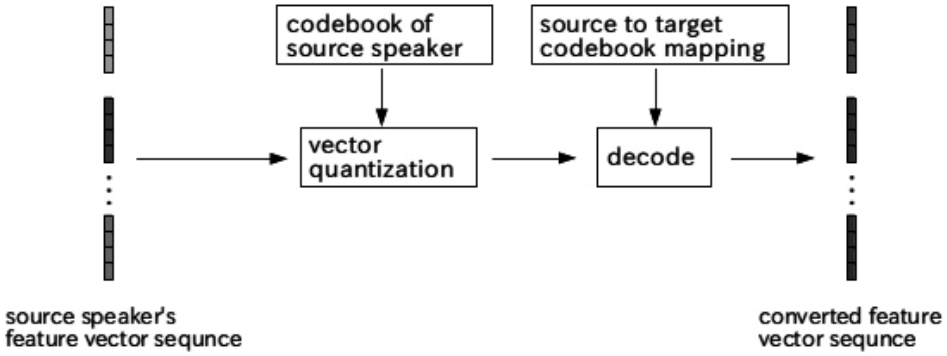

図 2.4: コードブックマッピングにおける変換ステップ

ベクトル量子化と差分ベクトルに基づくコードマッピング法 [14] も提案されている．しかし，現在は連続的かつより高精度な変換手法として Gaussian Mixuture Model (混合正規分布モデル) や Artificial Neural Network を用いた変換手法が主に扱われている．

第3章

Gaussian Mixture Modelsを用いた声質変換手法と応用手法

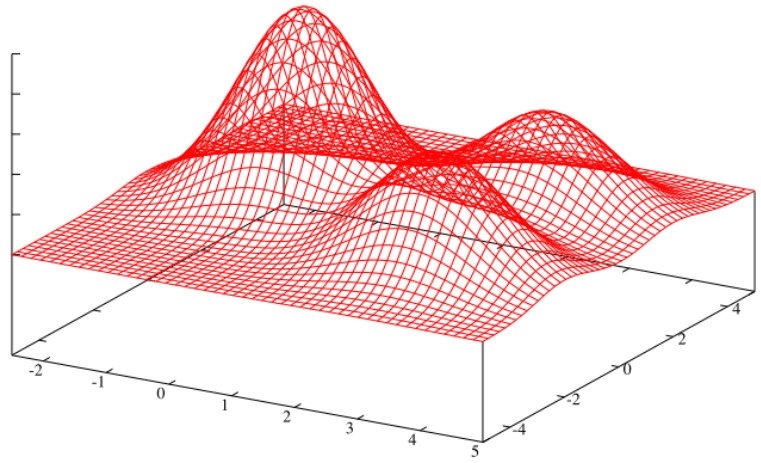

図 3.1: 混合ガウス分布の概略図

3.1　Gaussian Mixture Models

Gaussian Mixture Models (GMM) は, 未知の確率変数ベクトルの確率密度関数を, 混合ガウス分布として推定するモデルである [15]. 図 3.1 に簡略化した混合ガウス分布を示す.

D 次元の確率変数ベクトル $\mathbf{x}$ が与えられたとき, GMM ではその確率密度関数 $p(\mathbf{x})$ を (3.1) 式として表す.

$$p(\mathbf{x}) = \sum_{i=1}^{M} P(w_i)\mathcal{N}(\mathbf{x}; \mu_i^x, \mathbf{\Sigma}_i^{xx}) \tag{3.1}$$

ここで, M は混合数, $P(w_i)$ は各ガウス分布の重み, μ, Σ はそれぞれ i 番目のガウス分布における平均ベクトルと分散共分散行列を表し, $\mathcal{N}(\mathbf{x}; \mu, \mathbf{\Sigma})$ は以下で表されるガウス分布である.

$$\mathcal{N}(\mathbf{x}; \mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}} \Sigma^{\frac{1}{2}}} \exp\left(-\frac{(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}{2}\right)$$

このモデルに対して, 訓練データとして入力話者の特徴量の時間系列 $X = [x_1, x_2, ..., x_n]$ と出力話者の特徴量の時間系列 $Y = [y_1, y_2, ..., y_n]$ のパラレルデータを用い, X と Y の各要素の結合ベクトル $z_i = [x_i^T, y_i^T]^T$ の時間系列 $Z = [z_1, z_2, ..., z_n]$ の確率密度関数を考え, EM アルゴリズムによって二乗誤差平均が最小となる適切なパラメータを推定する. このとき, 入力話者の特徴量 x_k を出力話者の特徴量 y に変換する関数は (3.2) 式のように表される.

$$\begin{aligned} F(x_k) &= E(y \mid x_k) \\ &= \sum_{i=1}^{M} P(w_i \mid \mathbf{x}_k) \left[\mu_i^y + \mathbf{\Sigma}_i^{yx} \mathbf{\Sigma}_i^{xx^{-1}} (\mathbf{x}_k - \mu_i^x)\right] \end{aligned} \tag{3.2}$$

ここで, $E(.)$ は期待値を表し, 条件付確率 $P(w_i \mid \mathbf{x}_k)$ は式 (3.3) で与えられる.

$$P(w_i \mid \mathbf{x}_k) = \frac{P(w_i)\mathcal{N}(\mathbf{x}_k; \mu_i^x, \mathbf{\Sigma}_i^{xx})}{\sum_{j=1}^{M} P(w_j)\mathcal{N}(\mathbf{x}_k; \mu_j^x, \mathbf{\Sigma}_j^{xx})} \tag{3.3}$$

EM アルゴリズムによる学習においては, 時間対応の取れた x_k と y_k の組が必要となるため, パラレルコーパスが必要となる.

3.2　EM アルゴリズム

EM アルゴリズムは Expectation Step(E-step) と Maximization Step(M-step) の 2 ステップを繰り返すことで最尤推定を行うアルゴリズムをいう．X と Y の結合ベクトルの時間系列 $Z = [z_1, z_2, ..., z_n]$ の確率密度関数を計算するのに必要なパラメータは各ガウス分布 w_i に対してのガウス分布の重み $P(w_i)$, 平均ベクトル μ_i^z, 共分散行列 $\mathbf{\Sigma}_i^{zz}$ であり, k-means などによって定めた初期値に従ってこれらのパラメータを推定する．

E-step では, 現在のパラメータの値から式 (3.4) の条件付確率確率を計算する. ここで t は反復回数を示す．

$$P^{(t)}(w_i \mid \mathbf{z}_k) = \frac{P^{(t)}(w_i)\mathcal{N}(\mathbf{z}_k; \mu_i^{(t)z}, \mathbf{\Sigma}_i^{(t)zz})}{\sum_{j=1}^{M} P^{(t)}(w_j)\mathcal{N}(\mathbf{z}_k; \mu_j^{(t)z}, \mathbf{\Sigma}_j^{(t)zz})} \tag{3.4}$$

M-step では, E-step によって得られた条件付確率 $P^{(t)}(w_i \mid \mathbf{z}_k)$ によって GMM のパラメータを再推定する．各パラメータは式 (3.5)(3.6)(3.7) によって計算される．

$$P^{(t+1)}(w_i) = \frac{1}{n}\sum_{k=1}^{n} P^{(t)}(w_i \mid z_k) \tag{3.5}$$

$$\mu_i^{(t+1)z} = \frac{\sum_{k=1}^{n} P^{(t)}(w_i \mid z_k) z_k}{\sum_{k=1}^{n} P^{(t)}(w_i \mid z_k)} \tag{3.6}$$

$$\Sigma_i^{(t+1)zz}\Gamma = \frac{\sum_{k=1}^{n} P^{(t)}(w_i \mid z_k)(z_k - \mu_i^{(t+1)z})(z_k - \mu_i^{(t+1)z})^T}{\sum_{k=1}^{n} P^{(t)}(w_i \mid z_k)} \tag{3.7}$$

この E-step と M-step を各パラメータおよび条件付確率が収束するまで繰り返す．最終的な結果から, 入力特徴量 x_k を特徴量 y に変換する (3.2) 式,(3.3) 式の計算において必要となる共分散行列 $\mathbf{\Sigma}_i^{xx}$, $\mathbf{\Sigma}_i^{yx}$, 平均ベクトル μ_i^x, μ_i^y を推定されたパラメータから以下の式によって得る．

$$\Sigma_i^{zz} = \begin{bmatrix} \Sigma_i^{xx} & \Sigma_i^{xy} \\ \Sigma_i^{yx} & \Sigma_i^{yy} \end{bmatrix} \tag{3.8}$$

$$\mu_i^z = \begin{bmatrix} \mu_i^x \\ \mu_i^y \end{bmatrix} \tag{3.9}$$

3.3　Gaussian Mixture Models における話者適応手法

GMM による声質変換では学習の際に入力話者と出力話者による時間対応の取れたパラレルコーパスが必要となり, 他の話者を入出力話者とするモデルを作ろうとする度に大規模なコーパスが必要となってしまうという問題がある．このような問題を改善することを目的とした, すなわち声質変換を話者適応的なシステムに改善することを目的とした手法が提案されている [6]．ここではその中の代表的な手法を挙げ, その概要を示す．

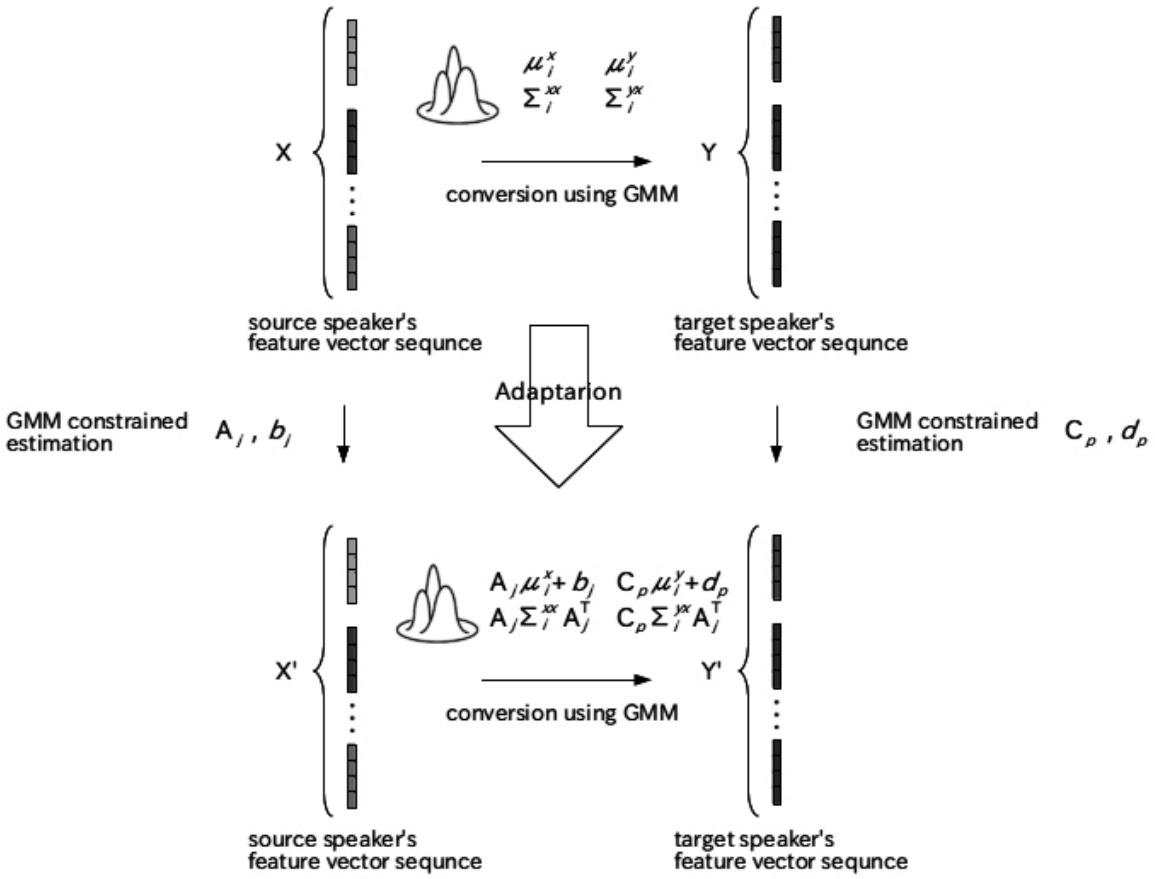

図 3.2: パラメータ適応の概要

3.3.1　parameter adaptation

Parameter Adaptation は, 新たな入力話者から出力話者への変換モデルを学習する際に, 新しくモデルを作り直すのではなく, 他の話者間のモデルに対して新たな入出力話者のデータでモデルの適応を行うという手法である. 手法の概要を図 3.2 に示す.

特定話者 X, Y 間の GMM から新たな話者 X', Y' の GMM のパラメータを推定する場合を考える. 基の GMM の入力話者の確率変数ベクトルを x としたとき, 対象話者の確率変数ベクトル x' を x の確率的線形変換として (3.10) 式のように定義する.

$$\mathbf{x}' = \begin{cases} A_1\mathbf{x} + \mathbf{b}_1 & \text{with probability } p(\lambda_1|w_i) \\ A_2\mathbf{x} + \mathbf{b}_2 & \text{with probability } p(\lambda_2|w_i) \\ \vdots & \vdots \\ A_N\mathbf{x} + \mathbf{b}_N & \text{with probability } p(\lambda_N|w_i) \end{cases} \tag{3.10}$$

λ_j は x の各ガウス分布 w_i に対応した変換を表し, $p(\lambda_j \mid w_i)$ は (3.11) 式を満たす.

$$\sum_{j=1}^{N} p(\lambda_j \mid w_i) = 1, \quad i = 1, ..., M. \tag{3.11}$$

また, M は適応元の GMM の混合数, A_j は $K * K$ 行列 (K は x の次元) である. 出力話者 y, y' に関しても同様に (3.12) 式のように定義する.

$$\mathbf{y}' = \begin{cases} C_1\mathbf{y} + \mathbf{d}_1 & \text{with probability } p(\kappa_1|w_i) \\ C_2\mathbf{y} + \mathbf{d}_2 & \text{with probability } p(\kappa_2|w_i) \\ \vdots & \vdots \\ C_L\mathbf{y} + \mathbf{d}_L & \text{with probability } p(\kappa_L|w_i) \end{cases} \tag{3.12}$$

$$\sum_{\rho=1}^{L} p(\kappa_j \mid w_i) = 1, \quad i = 1, ..., M. \tag{3.13}$$

式 (3.10) から, x' の確率密度関数は w_i, λ_j が与えられたとき,

$$g(\mathbf{x}' \mid w_i, \lambda_j) = \mathcal{N}(\mathbf{x}'; A_j\mu_i^x + \mathbf{b}_j, A_j\mathbf{\Sigma}_i^{xx}A_j^T) \tag{3.14}$$

$$g(\mathbf{x}') = \sum_{i=1}^{M}\sum_{j=1}^{N} p(w_i)p(\lambda_j \mid w_i)\mathcal{N}(\mathbf{x}'; A_j\mu_i^x + \mathbf{b}_j, A_j\mathbf{\Sigma}_i^{xx}A_j^T) \tag{3.15}$$

となり, 混合数 $M * N$ の GMM と考えることができる. そのため, 式 (3.10), 式 (3.11), 式 (3.12), 式 (3.13) における未知パラメータ A_j, C_ρ, $\mathbf{b}_j$, $\mathbf{d}_\rho$ は x と y のパラレルコーパスによる GMM を基に, x' と y' の非パラレルコーパスから, EM アルゴリズムによって推定することができる.

t 回目の試行における E-step では, 以下の式でパラメータを計算する.

$$n_{ij}^{(t)} = \sum_{k=1}^{n} p^{(t)}(w_i \mid \mathbf{x}'_\mathbf{k})p^{(t)}(\lambda_j \mid \mathbf{x_k}', w_i) \tag{3.16}$$

$$\mu_{ij}^{(t)x'} = \frac{1}{n_{ij}^{(t)}}\sum_{k=1}^{n} p^{(t)}(w_i \mid \mathbf{x}'_\mathbf{k})p^{(t)}(\lambda_j \mid \mathbf{x_k}', w_i)\mathbf{x}'_\mathbf{k} \tag{3.17}$$

$$\Sigma_{ij}^{(t)x'x'} = \frac{1}{n_{ij}^{(t)}}\sum_{k=1}^{n} p^{(t)}(w_i \mid \mathbf{x}'_\mathbf{k})p^{(t)}(\lambda_j \mid \mathbf{x_k}', w_i) \Gamma(\mathbf{x}'_\mathbf{k} - \mu_\mathbf{ij}^{(\mathbf{t})\mathbf{x}'})(\mathbf{x}'_\mathbf{k} - \mu_\mathbf{ij}^{(\mathbf{t})\mathbf{x}'})^T \tag{3.18}$$

このとき, $p^{(t)}(w_i \mid \mathbf{x}'_\mathbf{k})$, $p^{(t)}(\lambda_j \mid \mathbf{x_k}', w_i)$ は以下のようになる.

$$p^{(t)}(w_i \mid \mathbf{x}'_\mathbf{k}) = \frac{p(w_i)\sum_{j=1}^{N} p^{(t)}(\lambda_j|w_i)g^{(t)}(\mathbf{x}'_\mathbf{k}|w_i,\lambda_j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} p(w_i)p^{(t)}(\lambda_j|w_i)g^{(t)}(\mathbf{x}'_\mathbf{k}|w_i,\lambda_j)}\Gamma \tag{3.19}$$

$$p^{(t)}(\lambda_j \mid \mathbf{x_k}', w_i) = \frac{p^{(t)}(\lambda_j|w_i)g^{(t)}(\mathbf{x}'_\mathbf{k}|w_i,\lambda_j)}{\sum_{j=1}^{N} p^{(t)}(\lambda_j|w_i)g^{(t)}(\mathbf{x}'_\mathbf{k}|w_i,\lambda_j)} \tag{3.20}$$

同様に, M-step では以下の式でパラメータを計算する.

$$p^{(t+1)}(\lambda_j \mid w_i) = \frac{n_{ij}^{(t)}}{\sum_{j=1}^{N} n_{ij}^{(t)}} \tag{3.21}$$

$$\Gamma \sum_{i=1}^{M} n_{ij}^{(t)} \{ \mathbf{A_j^{(t+1)}} - {\Sigma_i^{xx}}^{-1} [{\mathbf{A_j^{(t+1)}}}^{-1} (\mu_{\mathbf{ij}}^{(\mathbf{t})\mathbf{x}'} - \mathbf{b_j^{(t+1)}}) - \mu_{\mathbf{i}}^{\mathbf{x}}] (\mu_{\mathbf{ij}}^{(\mathbf{t})\mathbf{x}'} - \mathbf{b_j^{(t+1)}})^T - {\Sigma_i^{xx}}^{-1} {\mathbf{A_j^{(t+1)}}}^{-1} \Sigma_{ij}^{(t)x'x'} \} = 0\Gamma \quad (3.22)$$

$$\mathbf{b_j^{(t+1)}} = \left[\sum_{i=1}^{M} n_{ij} {\mathbf{A_j^{(t+1)}}}^{-\mathbf{T}} {\Sigma_i^{xx}}^{-1} \mathbf{A_j^{(t+1)}}^{-1} \right]^{-1} \Gamma \left[\sum_{i=1}^{M} n_{ij} {\mathbf{A_j^{(t+1)}}}^{-\mathbf{T}} {\Sigma_i^{xx}}^{-1} \mathbf{A_j^{(t+1)}}^{-1} \left(\mu_{\mathbf{ij}}^{(\mathbf{t})\mathbf{x}'} - \mathbf{A_j^{(t+1)}} \mu_{\mathbf{i}}^{\mathbf{x}} \right) \right] \quad (3.23)$$

C_ρ, d_ρ に関しても, x と x' の代わりに y と y' を用いることで同様に計算することができる．最終的に, 新たな入力話者特徴量 x'_k から y' への変換関数は以下の式から計算される．

$$\begin{aligned} F(\mathbf{x'_k}) &= E(\mathbf{y}' \mid \mathbf{x'_k}) \\ &= \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{L} p(w_i \mid \mathbf{x'_k}, w_i) p(\kappa_\rho \mid w_i) \\ &\quad \Gamma[\mathbf{C}_\rho \mu_{\mathbf{i}}^{\mathbf{y}} + \mathbf{d}_\rho + \mathbf{C}_\rho \mathbf{\Sigma_i^{yx}} {\mathbf{\Sigma_i^{xx}}}^{-1} \mathbf{A_j^{-1}} \\ &\quad (\mathbf{x'_k} - \mathbf{A_j} \mu_{\mathbf{i}}^{\mathbf{x}} - \mathbf{b_j})] \end{aligned}$$

$$p(w_i \mid \mathbf{x'_k}) = \frac{p(w_i) \sum_{j=1}^{N} p(\lambda_j \mid w_i) g(\mathbf{x'_k} \mid w_i, \lambda_j)}{\sum_{i=1}^{M} \sum_{j=1}^{N} p(w_i) p(\lambda_j \mid w_i) g(\mathbf{x'_k} \mid w_i, \lambda_j)}$$

$$p(\lambda_j \mid \mathbf{x_k}', w_i) = \frac{p(\lambda_j \mid w_i) g(\mathbf{x'_k} \mid w_i, \lambda_j)}{\sum_{j=1}^{N} p(\lambda_j \mid w_i) g(\mathbf{x'_k} \mid w_i, \lambda_j)}$$

このような，既に学習済みのモデルを初期値とし少量のデータによって新たな話者への変換モデルへ適用を行う parameter adaptation を主軸とした手法が GMM ではよく用いられている．

3.3.2　MAP-based parameter adaptation

GMM における parameter adaptation を用いた手法の 1 つとして MAP-Based parameter adaptation がある (MAP: maximum a posteriori probability) [6]．この手法では上述したような線形変換を元にした parameter adaptation と同様に，パラレルコーパスの存在する特定話者間の同時確率をあらかじめ GMM により学習し，そこに新しい話者の非パラレルデータを少量用いてパラメータの適応を行う．手法の概要を図 3.3 に示す．図 3.3 のように，入力話者と出力話者の同時確率を学習した GMM のパラメータに対して MAP 推定を行うことによって，入力話者と新たな出力話者の同時確率を表す GMM へとパラメータの適応を行う．[6] では，分散共分散行列に関しては入出力話者による変化が少ないと仮定し，適応前のものをそのまま用いる．また，話者間の変換は非線形であるとし式で表される GMM による変換式を以下の式に変更する．

$$y = x + \sum_{i=1}^{M} P(i \mid x)(\mu_{\mathbf{i}}^{\mathbf{Y}} - \mu_{\mathbf{i}}^{\mathbf{X}}) \quad (3.24)$$

MAP 推定は入力データ列 X が与えられたとき，モデルパラメータ θ を確率変数とし，その事後確率が最大となるような θ を推定する．これを式で表すと以下のようになる．

$$\hat{\theta} =_\theta P(\theta \mid X)$$

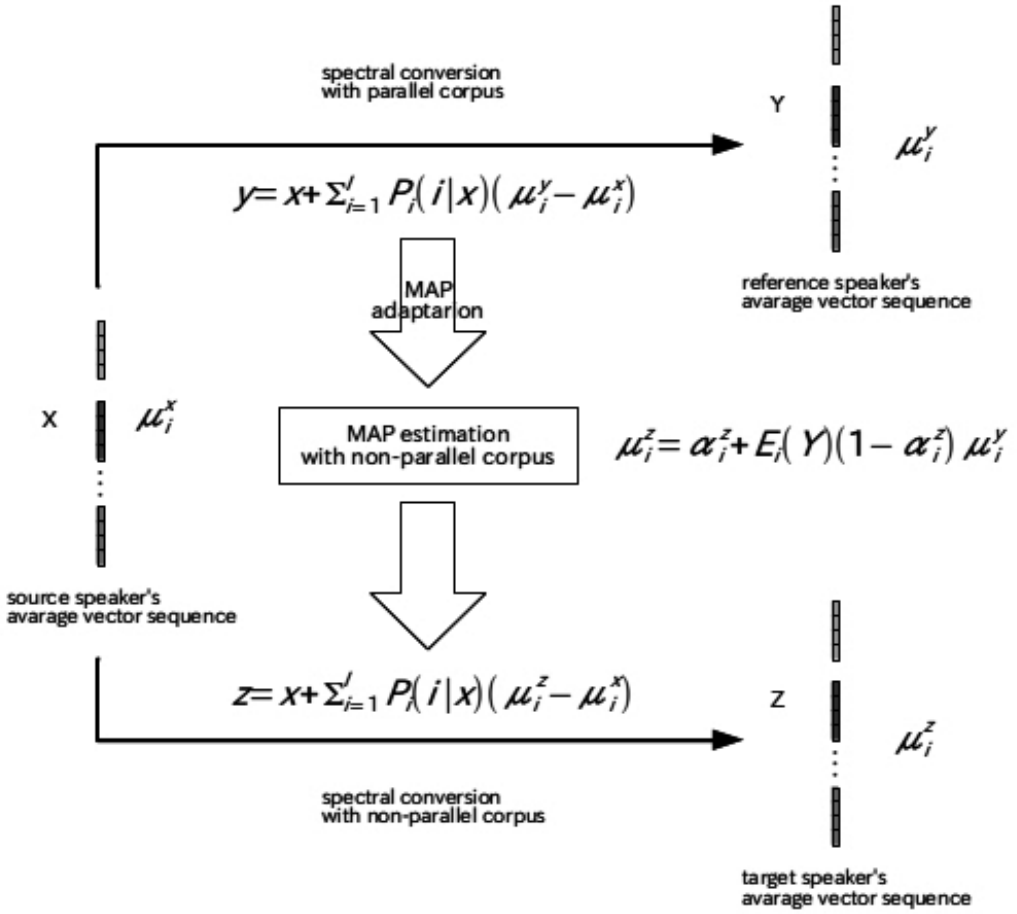

図 3.3: MAP 適応による声質変換の概要

この式は，ベイズの定理より以下のように変形できる．

$$\hat{\theta} =_\theta P(X \mid \theta)P(\theta) \tag{3.25}$$

ここで，$P(\theta)$ は入力データ列 X が与えられていないときの θ の事前分布となり，$P(X \mid \theta)$ は X に関する最尤推定値となる．即ち，MAP 推定は入力データ列 X のサイズが十分大きいときは X に関する最尤推定値に近くなり，十分でないときは事前情報として得られている $P(\theta)$ による値に近くなる．これは声質変換のモデルに当てはめると，事前分布 $P(\theta)$ がパラレルコーパスの存在している特定話者間の GMM のパラメータを表し，それを初期値として非パラレルデータによって適応を行う．

声質変換における GMM の出力話者に関する平均ベクトルの適応を考える．ソース話者の特徴量の時間系列 $X = [x_1, x_2, ..., x_n]$ と学習済みのターゲット話者 (リファレンス話者とする) の特徴量の時間系列 $Y = [y_1, y_2, ..., y_n]$ のパラレルデータを用い, X と Y の各要素の結合ベクトル $z_i = [x_i^T, y_i^T]^T$ の時間系列を $Z = [z_1, z_2, ..., z_n]$ としたとき，Z の GMM を以下の式で表す．

$$P(z) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(z; \mu_m^{(Z)}, \Sigma_m^{(Z)})$$

$$\mu_m^{(Z)} = \begin{bmatrix} \mu_m^{(X)} \\ \mu_m^{(Y)} \end{bmatrix} \tag{3.26}$$

$$\Sigma_m^{(Z)} = \begin{bmatrix} \Sigma_m^{(XX)} & \Sigma_m^{(XY)} \\ \Sigma_m^{(YX)} & \Sigma_m^{(YY)} \end{bmatrix} \tag{3.27}$$

このとき，X および Y の GMM も同様に以下の式で表される．

$$P(x) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(x; \mu_m^{(X)}, \Sigma_m^{(XX)})$$

$$P(y) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(y; \mu_m^{(Y)}, \Sigma_m^{(YY)})$$

このときソース話者とリファレンス話者の変換式は式 (3.24) で表される．この特徴量ベクトル x から特徴量ベクトル y への変換モデルを，特徴量ベクトル x から新しいターゲット話者 (W とする) の特徴量ベクトル z への変換に適応させる．

変換式 (3.24) を考えると，適応するパラメータは平均ベクトルのみ，すなわち μ_m^Y を μ_m^W に適応すればよい．新しいターゲット話者の非パラレルコーパスからデータ w_i が与えられたとき，この w_i がリファレンス話者の GMM における k 番目のガウス分布から生成されている確率は，

$$P(k \mid z_i)_{k,i} = \frac{\alpha_k \mathcal{N}(w_i; \mu_k^{(Y)}, \Sigma_k^{(YY)})}{\sum_{m=1}^{M} \alpha_m \mathcal{N}(w_i; \mu_m^{(Y)}, \Sigma_m^{(YY)})}$$

となる．このとき，非パラレルコーパス中のデータ $w_1, w_2, ..., w_n$ について，k 番目のガウス分布から生成されるデータに関する確率的サンプル数 N_k と確率的平均ベクトル e_k は以下の式で表される．

$$N_k = \sum_{i=1}^{n} P(k \mid z_i)_{k,i} \tag{3.28}$$

$$e_k = \frac{1}{N_k} \sum_{i=1}^{n} P(k \mid z_i)_{k,i} w_i \tag{3.29}$$

これらの式を用いて，μ_m^Y を以下の様に更新する．

$$\hat{\mu}_m^Y = \frac{N_k}{N_k + \gamma} e_k + \frac{\gamma}{N_k + \gamma} \mu_k^Y \tag{3.30}$$

γ は事前分布，すなわちリファレンス話者の特徴量系列における k 番目のガウス分布から生成される確率的サンプル数を表す．

3.3.3 Eigen-Voice Conversion

Eigen-Voice Conversion [7] の概要を図 3.4 に示す．Eigen-Voice Conversion では, 初めに変換元の話者と多数の事前収録話者からなるパラレルコーパスを用いて, 話者非依存の GMM の学習を行う. 次に, この話者非依存の GMM を初期モデルとして, 各事前収録話者の話者依存 GMM を

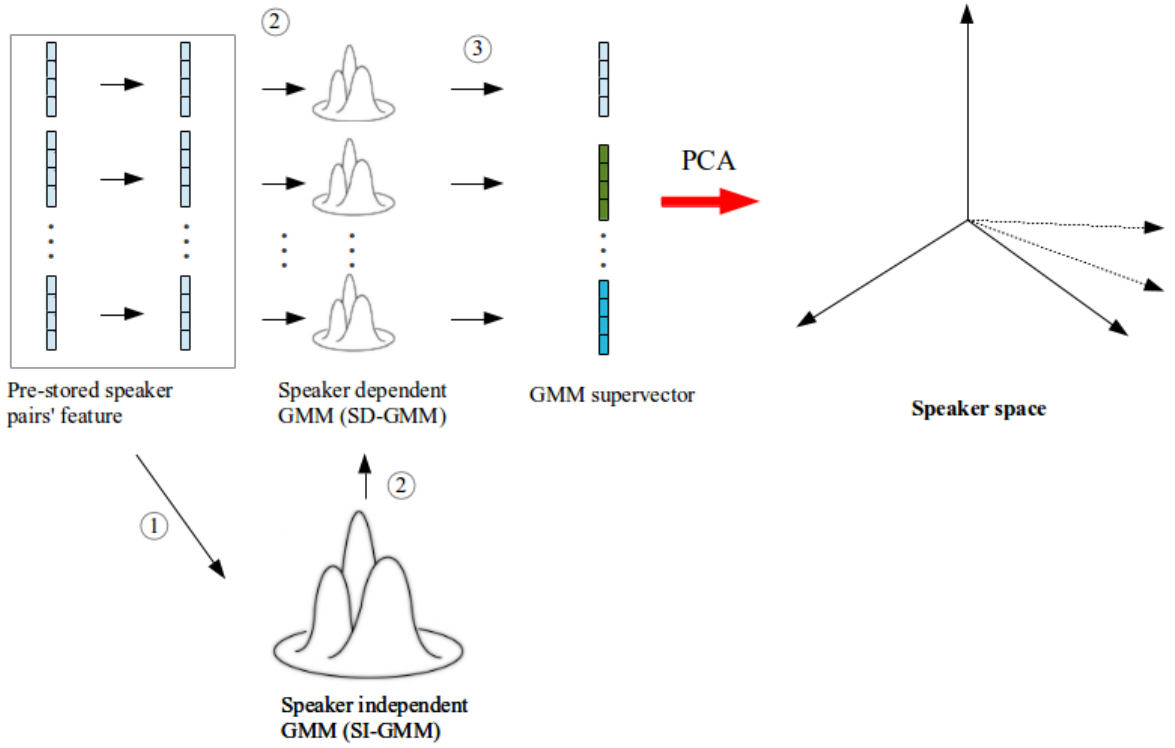

図 3.4: Eigen voice coversion の概要

学習する．ここで, 特徴量ベクトルの次元を D, ガウス分布の混合数を $M(m = 1, 2, ..., M)$, 添字 s を各話者 $(s = 0, 1, 2, ..., S$ で, $p = 0$ は不特定話者) とすると, 変換元の特徴量 $\mathbf{X_t}$ と s 番目の事前収録話者の特徴量 $\mathbf{Y_t^{(s)}}$ は Eigen-Voice GMM として以下の様にモデル化される．

$$
\begin{aligned}
&P(\mathbf{X_t}, \mathbf{Y_t^{(s)}} \mid \lambda^{(\mathbf{EV})}, \omega^{(\mathbf{s})}) \\
&= \sum_{m=1}^{M} \alpha_m \mathcal{N}([\mathbf{X_t^T}, \mathbf{Y_t^{(s)^T}}]^{\mathbf{T}}; \mu_{\mathbf{m}}^{(\mathbf{Z})}(\omega^{(\mathbf{s})}), \mathbf{\Sigma_m^{(Z)}})
\end{aligned}
\tag{3.31}
$$

$$
\mu_m^{(Z)}(\omega^{(s)}) = \begin{bmatrix} \mu_m^{(X)} \\ \mathbf{B_m}\omega^{(\mathbf{s})} + \mathbf{b_m^{(0)}} \end{bmatrix} \tag{3.32}
$$

$$
\Sigma_m^{(Z)} = \begin{bmatrix} \Sigma_m^{(XX)} & \Sigma_m^{(XY)} \\ \Sigma_m^{(YX)} & \Sigma_m^{(YY)} \end{bmatrix} \tag{3.33}
$$

ここで α_m は m 番目のガウス分布の重みを表す．EV-GMM では, S 人の事前収録話者を用いて出力話者の平均ベクトル $\mu_m^{(Y)}$ をバイアスベクトル $b_m^{(0)}$ と K 個の表現ベクトルの線型結合で表す．すなわち, 話者空間が K 個の基底スーパーベクトルとバイアススーパーベクトルによって張られる．

主成分分析に基づいた EV-GMM では, 初めに変換元の話者に対して全ての事前収録話者とのパラレルデータによる学習を行い, 話者非依存の GMM を構築する．それを初期モデルとして, 各

事前収録話者の話者依存 GMM を学習する．ここから, 話者空間の特徴量ベクトルとして各事前登録話者の GMM の平均ベクトルを連結し, 得られたスーパーベクトルに主成分分析を行うことで基底ベクトルとし, バイアスベクトル $\mathbf{b}$ と表現ベクトル $\mathbf{B}$ を計算する．

任意の話者に対しての EVGMM の適応は, 出力話者のデータを用いた重みベクトル ω の最尤推定によって行う．出力話者の特徴量系列を $\mathbf{Y^{(tar)}}$ としたとき, ω は以下の様に推定される．

$$\hat{\omega} = argmax_{\omega} P(\mathbf{Y^{(tar)}} \mid \lambda^{(\mathbf{EV})}, \omega) \tag{3.34}$$

出力の確率密度関数は GMM で表されるため, 補助関数として (3.35) 式を導入し, EM アルゴリズムを用いることで重みベクトル ω を最適化する．

$$Q(\omega,\hat{\omega})=\sum_m P(m|\mathbf{Y^{(tar)}},\lambda^{(\mathbf{EV})},\omega) \log P(\mathbf{Y^{(tar)}},\mathbf{m}|\lambda^{(\mathbf{EV})},\omega)$$

この補助関数により, $\hat{\omega}$ に関する以下の更新式が得られる．

$$\hat{\omega} = \left\{ \sum_{m=1}^{M} \bar{\gamma}_m^{(tar)} \mathbf{B_m^T \Sigma_m^{(YY)^{-1}} B_m} \right\}^{-1} \sum_{m=1}^{M} \mathbf{B_m^T \Sigma_m^{(YY)^{-1}} \bar{Y}_m^{(tar)}} \tag{3.35}$$

$$\bar{\gamma}_m^{(tar)} = \sum_{t=1}^{T} \gamma_{m,t} \tag{3.36}$$

$$\bar{\mathbf{Y}}_m^{(tar)} = \sum_{t=1}^{T} (\mathbf{Y}_t^{(tar)} - b_m^{(0)}) \tag{3.37}$$

$$\gamma_{m,t} = P(m \mid \mathbf{Y}_t^{(tar)}, \lambda^{(EV)}, \omega) \tag{3.38}$$

式 (3.35) は話者空間の基底ベクトルへの射影重みを推定していることに相当する．

これにより, 推定する必要のあるパラメータが少量かつ出力話者の発話内容を知る必要が無いため, 通常の GMM の学習に比べて極小量のデータによる, 教師無し適応が可能となる．

第4章

Artificial Neural Networkを用いた声質変換手法と応用手法

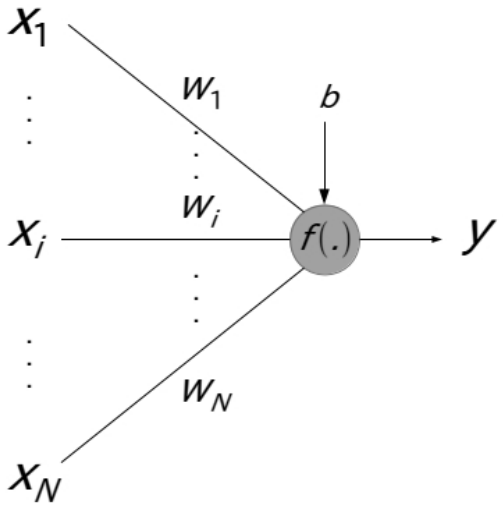

図 4.1: Artificial neuron

4.1 Artificial Neural Network

GMM 以外の入力特徴量を出力特徴量に直接変換するモデルの 1 つとして，Artificial Neural Network (ANN) がある．ANN は，人間のニューロン (神経細胞) を数式的に模した Artificial Neuron を用いてネットワーク構造を構築したモデルのことをいう．Artificial Neuron の概要を図 4.1 に示す． Artificial Neuron は数式化すると，以下の式 (1) で表される．

$$y = f(\sum_{i=1}^{N}(w_i x_i + b_i)) \tag{4.1}$$

ここで，x_i は入力信号を表し，w_i は各入力信号にかけられる結合重み，y は出力信号をそれぞれ表す．$f(\cdot)$ は活性化関数であり，シグモイド関数などが用いられる．また，b はバイアス項，N は入力の次元数をそれぞれ表す．これは他のニューロンからの入力信号がシナプスを通して現在のニューロンに伝達し，それが閾値を越えたとき現在のニューロンが発火し，結合している他のニューロンへ信号を伝達するという仕組みをモデル化している．この Artificial Neuron を複数接続し，ネットワーク構造を成したものを ANN と呼ぶ．ANN の 1 つであり，後述する Deep Neural Network と同じ構造を持つモデルである多層パーセプトロンの例を図 4.2 に示す．多層パーセプトロンは，複数の Artificial Neuron を層状に配置し，隣接する層との間で結合したものである．h_i は隠れ層，v_1 と v_2 は可視層と呼ばれ，v_1 と v_2 がそれぞれ入力層と出力層となる．このモデルに対して，入力特徴量と正解データ (とする特徴量) の組からなるパラレルコーパスを用いて，入力に対する出力と正解データの間の誤差を計算し，その誤差の値によって出力層側から順に各層の重み w_{ij}，バイアス b_j を更新することで学習を行う．この学習を誤差逆伝搬法という．

ANN は，十分な層数・ノード数を用いて構成することで，任意の関数を再現できるという非常に高い表現力を持つ．一方で ANN は，層を増やす程に，誤差逆伝搬法によるパラメータの更新が入力側に近い層まで伝達しづらくなり，また，その高い表現力のために過学習が起きやすくなるという問題点がある．この問題点を改善するために後述する Deep Learning という ANN の学

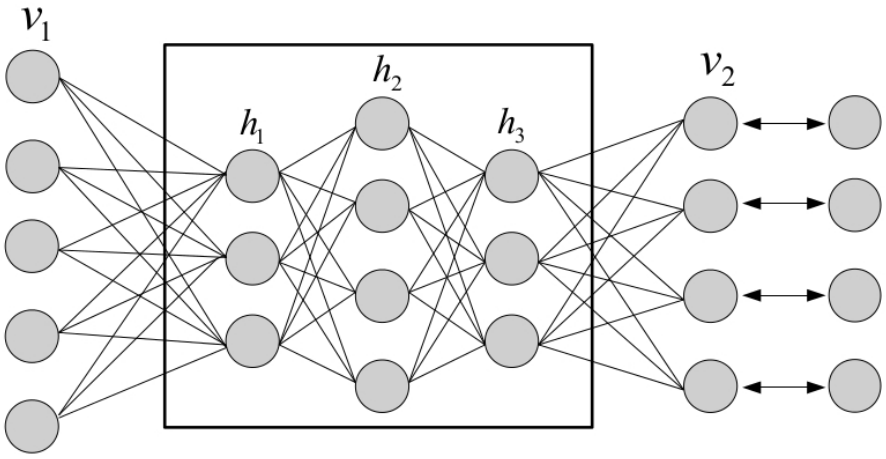

図 4.2: Multi-layer perceptron

習手法が提案された.

4.1.1 Artificial Neural Network による声質変換

ANN による実際の学習の例として，M 層の ANN を用いて声質変換を行う場合を考える ($m = 1, ..., M$ とする) [4]．この DP マッチングを行ったソース話者とターゲット話者の D 次元特徴量系列を $X = x_1, x_2, ..., x_n$，$Y = y_1, y_2, ..., y_n$ とし，m 層目の隠れ層での値を h_m とすると，ANN によって t 番目の入力データ x_t に対して各層で行われる処理は以下の式で表される.

$$\mathrm{h}_t^{(1)} = \mathrm{sigm}(\mathrm{W}^{(1)}\mathrm{x}_t + \mathrm{b}^{(1)}) \quad : \text{input layer} \tag{4.2}$$

$$\mathrm{h}_t^{(m+1)} = \mathrm{sigm}(\mathrm{W}^{(m+1)}\mathrm{h}_t^{(m)} + \mathrm{b}^{(m+1)}) \quad : \text{hidden layer}\Gamma(m < M) \tag{4.3}$$

$$\hat{\mathrm{y}}_t = \mathrm{W}^{(M)}\mathrm{h}_t^{(M-1)} + \mathrm{b}^{(M)} \quad : \text{output layer} \tag{4.4}$$

$\hat{\mathrm{y}}_t$ はソース話者からターゲット話者に変換された特徴量，$\mathrm{W}^{(m)}$，$\mathrm{b}^{(m)}$ は m 層目の結合重み行列とバイアスベクトルをそれぞれ表す．また，sigm は式 (4.5) で定義されるシグモイド関数を表し，緩やかな閾値関数のように働く.

$$\mathrm{sigm}(\mathrm{x}) = \frac{1}{1 + \exp^{-ax}} \tag{4.5}$$

上述した出力層の式で得られた変換特徴量 $\hat{\mathrm{y}}_t$ と正解データ y_t の間で誤差を計算する．ANN による連続値変換の誤差関数としては主に以下の式で表される二乗平均誤差 (MSE) が用いられる.

$$\mathrm{MSE} = \sum_{k=1}^{n} \mid \hat{\mathrm{y}}_k - \mathrm{y}_k \mid^2 \tag{4.6}$$

ANN ではこの誤差関数の値が最小となる様にパラメータ W，b を誤差逆伝搬法によって更新する．初めに 1 フレームの特徴量に対して，出力層において $\hat{y}_j$ を値として持つ j 番目のノードに関する誤差逆伝搬法を考える．($j = 1, 2, ..., D$) このとき，出力層の j 番目のノードにおける出力の

誤差は $E_j = \mid \hat{y}_j - y_j \mid^2$ で求められるとする．誤差逆伝搬法におけるパラメータの更新は勾配法によるものであり，m 層における入力側 i 番目のノードと出力側 j 番目のノードの結合重みの更新式は以下の様に定義される．

$$\mathrm{W}_{ij}^{m} = \mathrm{W}_{ij}^{m} - \epsilon \frac{\partial E}{\partial \mathrm{W}_{ij}^{m}} \tag{4.7}$$

ϵ は学習率である．出力層の前の層 ($M-1$ 層) の i 番目のノードの出力値を $h_i^{(M-1)}$ とすると，$\hat{y}_j$ の変化量に対する $\mathrm{W}_{ij}^{(M)}$ の変化量の関係は以下の式で示される．

$$\Delta \hat{y}_j = \Delta \mathrm{W}_{ij}^{(M)} h_i^{(M-1)} \tag{4.8}$$

$E_j = \mid \hat{y}_j - y_j \mid^2$ なので，

$$\Delta E = 2(\hat{y}_j - y_j)\Delta \hat{y}_j \tag{4.9}$$

よって出力層における更新式は式 (4.10) の様になる．

$$\mathrm{W}_{ij}^{(M)} = \mathrm{W}_{ij}^{(M)} - 2\epsilon(\hat{y}_j - y_j)h_i^{(M-1)} \tag{4.10}$$

出力層以外の層，すなわち活性化関数としてシグモイド関数を用いている層でも同様の手順によって以下の更新式が得られる．

$$\begin{aligned} z_j^{(m)} &= a(1 - h_j^{(m)})h_j^{(m)} \sum_{k=1}^{L} \mathrm{W}_{jk}^{(m+1)} z_k^{(m+1)} \\ \mathrm{W}_{ij}^{(m)} &= \mathrm{W}_{ij}^{(m)} - \epsilon z_j^{(m)} h_i^{(m-1)} \end{aligned} \tag{4.11}$$

ここで，L は $m+1$ 層におけるノード数である．また，$x = \mathrm{sigm}(s)$ のとき，$x' = ax(1-x)$ を利用している．

バイアス項に関しては結合重みの更新の際に求めた $z^{(m)}$ を用いて以下の式で更新する．

$$\mathrm{b}^{(M)} = \mathrm{b}^{(M)} - 2\epsilon(\hat{y} - y) \quad : \mathrm{outputlayer} \mathrm{b}^{(m)} = \mathrm{b}^{(m)} - \epsilon z_{(m)} \quad _{:\mathrm{other}} \tag{4.12}$$

4.1.2 Deep Learning

Deep Learning (深層学習) は，ANN における収束が遅いという問題点と過学習に陥りやすいという問題点を改善するために提案された手法である．初めに，Deep Learning の要素技術である Restricted Boltzmann Machine，Denoising Auto Encoder について述べる．

i) Restricted Boltzmann Machine

Restricted Boltzmann Machine (RBM) はニューラルネットの特殊形であり, 可視層と隠れ層の間にのみ結合が存在し, 可視層間・隠れ層間での結合が存在しない無向グラフィカルモデルで表される [16]. RBM では通常の ANN と同様に，可視層への入力に対して重み付け和をとりバイアス項を足し，シグモイド関数を掛けたものを隠れ層の値とする．図 4.3 に RBM のグラフィカルモデルを示す.

図 4.3 中の v は可視層, h は隠れ層, w は結合の重みをそれぞれ表す.

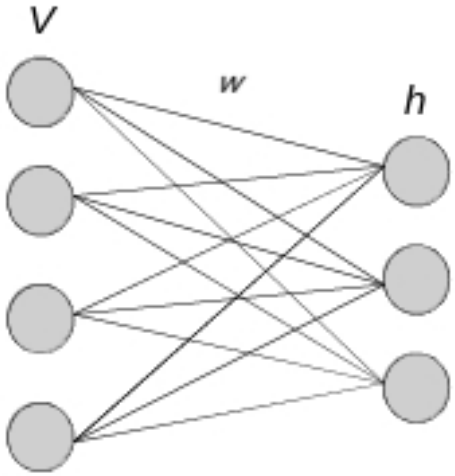

図 4.3: Restricted Boltzmann Machine

数式で示すと, RBM では, 可視素子の集合 $\mathrm{v} = \{v_1, v_2, ..., v_N\} \in \{0,1\}$ と隠れ素子の集合 $\mathrm{h} = \{h_1, h_2, ..., h_N\} \in \{0,1\}$ からなる結合確率 $p(\mathrm{v},\mathrm{h})$ を, 以下の式 (6)(7)(8) で定義する.

$$p(\mathrm{v},\mathrm{h}) = \frac{1}{Z}\exp(-E(\mathrm{v},\mathrm{h})) \tag{4.13}$$

$$E(\mathrm{v},\mathrm{h}) = -\mathrm{b}^{\mathrm{T}}\mathrm{v} - \mathrm{c}^{\mathrm{T}}\mathrm{h} - \mathrm{v}^{\mathrm{T}}\mathrm{W}\mathrm{h} \tag{4.14}$$

$$Z = \Sigma_{\mathrm{v},\mathrm{h}}\exp(-E(\mathrm{v},\mathrm{h})) \tag{4.15}$$

W は結合重み, b は可視素子のバイアスパラメータ, c は隠れ素子のバイアスパラメータをそれぞれ表す.

RBM には可視層間・隠れ層間での結合が存在しないという制約があるため, 条件付確率 $p(v_j = 1 \mid \mathrm{h})$, $p(h_i = 1 \mid \mathrm{v})$ は以下の式のようになる.

$$p(v_j = 1 \mid \mathrm{h}) = \mathrm{sigm}(b_j + \mathrm{W}_j\mathrm{h}) \tag{4.16}$$

$$p(h_i = 1 \mid \mathrm{v}) = \mathrm{sigm}(c_i + \mathrm{W}_i^{\mathrm{T}}\mathrm{v}) \tag{4.17}$$

ここで, sigm はシグモイド関数を示す.

パラメータの推定には, $p(\mathrm{v})$ に対する最尤推定を行う, つまり, 対数尤度の, 任意のパラメータ θ に関する最大化を行う. $p(\mathrm{v})$ の対数尤度を J とすると 3, その任意のパラメータ θ により微分は, 以下の様に表される.

$$\frac{\partial J}{\partial \theta} = -\left\langle \frac{\partial E}{\partial \theta} \right\rangle_{\mathrm{data}} + \left\langle \frac{\partial E}{\partial \theta} \right\rangle_{\mathrm{model}} \tag{4.18}$$

ここで $\langle . \rangle_{\mathrm{data}}$, および $\langle . \rangle_{\mathrm{model}}$ はそれぞれ観測データの対しての期待値, 内部モデルに対しての期待値を表す. 第一項は, 式 (9)(10) から比較的簡単に求めることができるが, 第二項は, 全ての v, h の組み合わせを考えなければならないので, ノード数によっては困難となる. そのため, 条件付確率 $p(\mathrm{v} \mid \mathrm{h})$ と $p(\mathrm{h} \mid \mathrm{v})$ によって可視素子の状態集合 v を再構成した v' を用いて, 最急降下法によって近似的にパラメータの更新を行っていく方法がよく取られる (Contrastive Divergence 法) [18].

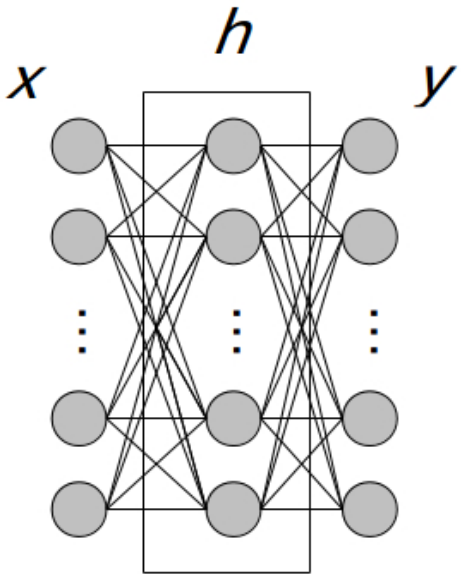

図 4.4: Denoising Auto Encoder

ii)　Denoising Auto Encoder

RBM 以外の Deep Learning の pre-training に用いられる特徴量抽出器として Denoising Auto Encoder (以下 dAE) がある．初めに通常の Auto Encoder (AE) について示す．AE は RBM と同様にニューラルネットの特殊形であり，入力層・隠れ層・復元層の 3 層からなる各層内に結合が存在しない無向グラフィカルモデルで表される．図 4.4 に dAE のグラフィカルモデルを示す．

dAE では入力層に与えられた値を隠れ層を通した上で復元層で再構成し，入力データそのものを教師データとして誤差が最小になる様にパラメータの学習を行う．このとき，隠れ層のノード数を入力データの次元数より少なくした場合，再構成の際により少ない情報から元のデータを復元する必要がある．そのため，隠れ層には元のデータよりも情報が圧縮された特徴量が生成されると考えられる．また，AE では通常の ANN と同様に，可視層への入力に対して重み付け和をとりバイアス項を足し，シグモイド関数を掛けたものを隠れ層の値とする．そして隠れ層の値に対して，同様に重み付け和をとりバイアス項を足し，シグモイド関数を掛けることで復元層の値を出力する．式で表すと以下のようになる．

$$\begin{aligned} \mathrm{h} &= \mathrm{sigm}(\mathrm{W}\mathrm{x} + \mathrm{b}) \\ \mathrm{y} &= \mathrm{sigm}(\mathrm{W}^T\mathrm{h} + \mathrm{b}') \end{aligned}$$

ここで，W，W^T はそれぞれ可視層から隠れ層への結合重みと隠れ層から復元層への結合重み (可視層から隠れ層への結合重みの転置), b，b' は可視層から隠れ層へのバイアスパラメータと隠れ層から復元層へのバイアスパラメータをそれぞれ表す．x，h，y はそれぞれ入力層と隠れ層と復元層の値を表す．

パラメータの更新は，ANN の時と同様に誤差関数 E が最小となるように勾配法によって得ら

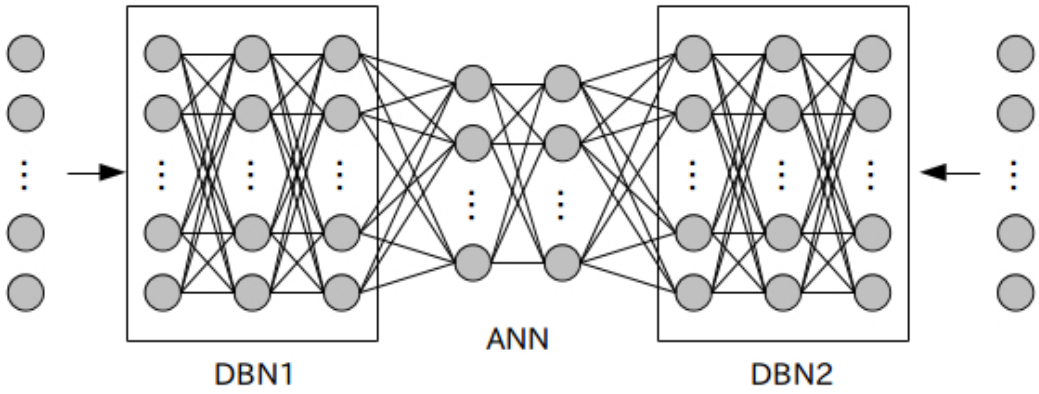

図 4.5: Deep Belief Nets による低次元空間表現を用いた声質変換

れる以下の式で行う．

$$
\begin{aligned}
\mathrm{W} &= \mathrm{W} - \epsilon \frac{\partial E}{\partial \mathrm{W}} \\
\mathrm{b} &= \mathrm{b} - \epsilon \frac{\partial E}{\partial \mathrm{b}} \\
\mathrm{b}' &= \mathrm{b}' - \epsilon \frac{\partial E}{\partial \mathrm{b}'}
\end{aligned}
$$

dAE では，AE での入力層の値 x に対してノイズを加えた $\hat{x}$ を入力値として用い，$\hat{x}$ から x を復元するようなパラメータ W，b，b’ をそれぞれ求める．これにより，隠れ層の次元数が入力の次元数よりも大きい場合であっても，恒等関数が最適にならないような問題に変形することができ，より頑健かつ汎化性能の高い特徴量の抽出を行うことができる．

iii)　Deep Neural Network

DNN では ANN における収束が遅いという問題点と過学習に陥りやすいという問題点を改善するために，誤差逆伝搬法を行う前に pre-training と呼ばれる各パラメータの初期値を計算する処理を行う [8].

pre-training では，dAE または RBM を用いて学習データに対して教師なし学習を行ったものを 1 層目とする．そして，学習の終わった 1 層目の隠れ層の値を学習データとして，同じように 2 層目を dAE，RBM によって学習する．この処理を繰り返すことによって全ての層の初期値を決定する．pre-training によって初期値が決定した後は，通常の ANN と同様に誤差逆伝搬法によって教師あり学習を行う．DNN における，この誤差逆伝搬法による教師あり学習を fine-tuning と呼ぶ.

4.1.3　Deep Belief Nets による声質変換手法

Deep Learning を用いた声質変換手法の 1 つとして Deep Belief Nets による低次元空間表現を用いた声質変換という手法が存在する [9]．Deep Belief Nets(DBN) は通常の 2 層からなる RBM を学習し，その隠れ層を次の RBM の入力と考えることで RBM を多層化した特徴量抽出器のことをいう．図 4.5 に手法の概要を示す.

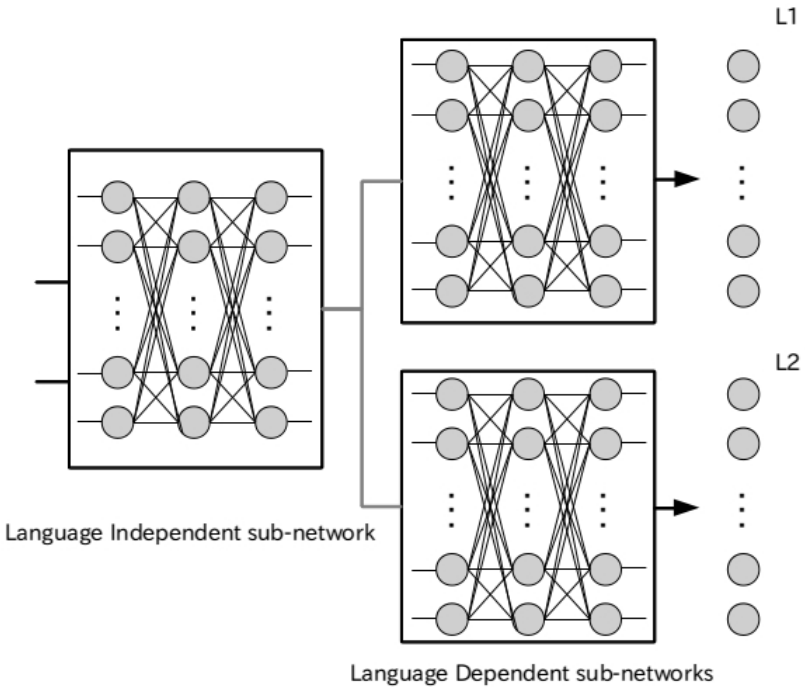

図 4.6: sub-networks on Deep Neural Networks

この研究では，深い階層を持つ DBN では各層のノード数で入力特徴量を表現するため，層の数が増えるほど入力特徴量が基底集合に近くなると仮定している．その考えを基に，ソース話者の特徴量と出力話者の特徴量をそれぞれ別の DBN によって抽出し，それぞれの最上位を ANN で接続する．このモデルによって fine-tuning を行うことで，より話者性の薄れた低次元空間において変換を行うことができるため，最適な非線形変換が可能となると考えている．実験として GMM による変換音声との主観評価と客観評価による比較を行っており，両尺度において GMM を上回る結果が得られていた．

4.1.4　多言語音声を学習した Deep Neural Network における言語非依存サブネットワークの自動適応

DNN のパラメータの適応を試みた手法として，松田らによる多言語音声を学習した Deep Neural Network における言語非依存サブネットワークの自動適応がある [12]．この手法は, 多言語音素認識というタスクを対象としており, 入力として複数の国の言語による音声が与えられ, その音声中の音素を識別するという一種の classifier の実装を目的としている．

手法の概略を図 4.6 に示す．この手法では, 大枠としては 5 章で説明した Deep Belief Nets を使用しており, pre-training においては, 複数言語の音声からなる訓練データによって RBM を入力層から順に学習していく．次に pre-training の終わったネットワークを一定の層で区切り, 前半はそのまま言語非依存のサブネットワークとして使用し, 後半は言語の数だけネットワークを複製し, 言語依存のネットワークとして使用する．具体的には, 教師あり学習 (fine-tuning) の際に, 前半の層は学習データの言語に関わらず 1 つのものを使用し, 後半の層は, 学習データの言語によって使用するネットワークを選択し, 学習を行う．実際の classtering の際には，全てのサブネットワークの出力値 (入力特徴量が与えられた時の各言語，各音素である事後確率を表す) から最大値

となるものを最終的な認識結果とする.

この手法では, まず初めに仮定として, Deep Learning では前半の浅い層において, 殆どの音響的事象に共通する時間変動や周波数などの特徴量の識別を行っており, 逆に深い層では音素や言語依存の特徴量などの複雑な情報を扱っていると考えている. そこで, 前半の層を入力言語に対して共通にする一方で, 後半で使用する層を言語毎に変更することで, 前半の言語非依存のサブネットワークでは言語に依存しない処理が集中し, 後半の言語依存のサブネットワークでは言語に依存する処理が集中するということが, 前述した仮定がより顕著になることで実現できるのではないかと考えており, 実験結果においても, 一対一で学習を行った DBN に対して高い, もしくは同程度の精度を出している.

第5章

提案手法

5.1　目的

5.1.1　GMM による声質変換と DNN による声質変換

4 章で示したように，GMM においては MAP 適応や eigen-voice conversion を用いたパラメータ適応による多対一および一対多声質変換手法が多く提案されている．一方で，ANN と DNN に関してはそういった多対一や一対多の声質変換を目的とした手法が GMM に比べて非常に少ない．GMM においてパラメータ適応を扱う手法が多く存在しているのは，適応するべきパラメータが各話者における平均ベクトルや話者間の分散共分散行列というように，それぞれ何を表しているかが分かりやすく，明示的なためである．しかし，ANN と DNN では，各層および各ノードの持つ結合重みやバイアス項が声質変換においてどういった情報を持つのかが明示的でなく，GMM のように柔軟なパラメータ適応を行うことができない．

一方で，単純な一対一の声質変換においては，ANN を用いた声質変換や DNN を用いた声質変換による変換精度が GMM の変換精度を上回っているという研究が報告されている [4][9]．これは，入力話者および出力話者の音声が発せられる声道の形状は非線形的であることから，非線形的な変換を行う ANN が音声情報を扱うのに適しているからであると考えられる．このことから，DNN を用いた声質変換において話者適応を行うことができれば，GMM によるものよりもより高い精度の変換が可能であると考える．

そこで，本研究では音声を扱うのにより適していると考えられる ANN，DNN の枠組みを用いて一対多や多対一のような柔軟な声質変換を可能とする変換モデルの構築を目的とする．

5.1.2　着想・理論

GMM におけるパラメータ適応手法として挙げた eigen-voive conversion の枠組みを考える．eigen-voive conversion では，あらかじめ用意してある多数の話者からなるパラレルコーパスによって話者に非依存な GMM(Universal Background Model :UBM) を学習する．そこから分散共分散行列を固定したまま，特定の話者ペアの GMM を平均ベクトルの更新のみで求め，この平均ベクトルに対して主成分分析を行うことで話者毎の平均ベクトルのばらつきの基底を計算する．この基底を話者毎に適応することで，元々の GMM のパラメータ数に比べて少ないパラメータの更新で新しい話者を含む変換に適応することができる．この eigen-voice conversion の処理では，分散共分散行列で表される入出力特徴量の各次元間の関係はソース話者とターゲット話者に依存しない情報であると考え，平均ベクトルのみがソース話者とターゲット話者に大きく依存すると考えている．

MAP 適応による一対多声質変換においても，初めにパラレルコーパスの存在している話者間で GMM による同時確率分布を計算し，分散共分散行列の値は適応前の話者間のものをそのまま用い，MAP 適応によるパラメータの適応は平均ベクトルのみとなっている．

これらの手法の間では，特定の話者に依存しないと考えられるパラメータに関しては，理想的なデータ (パラレルコーパス) が揃っている話者を用いて予め事前知識として推定しておき，新しい話者に関する変換を行う際には，その新しい話者に依存するパラメータのみを更新するという考え方が共通している．これを GMM のような生成モデルではなく ANN や DNN のような識別モデルに当てはめると，声質変換においては話者非依存な変換処理と話者依存な変換処理を分けて考えることが有効であり，話者非依存な変換処理と話者依存な変換処理を分離することができれば，そこから話者依存な変換処理のみを更新するような柔軟な声質変換が実現できると考えられる．

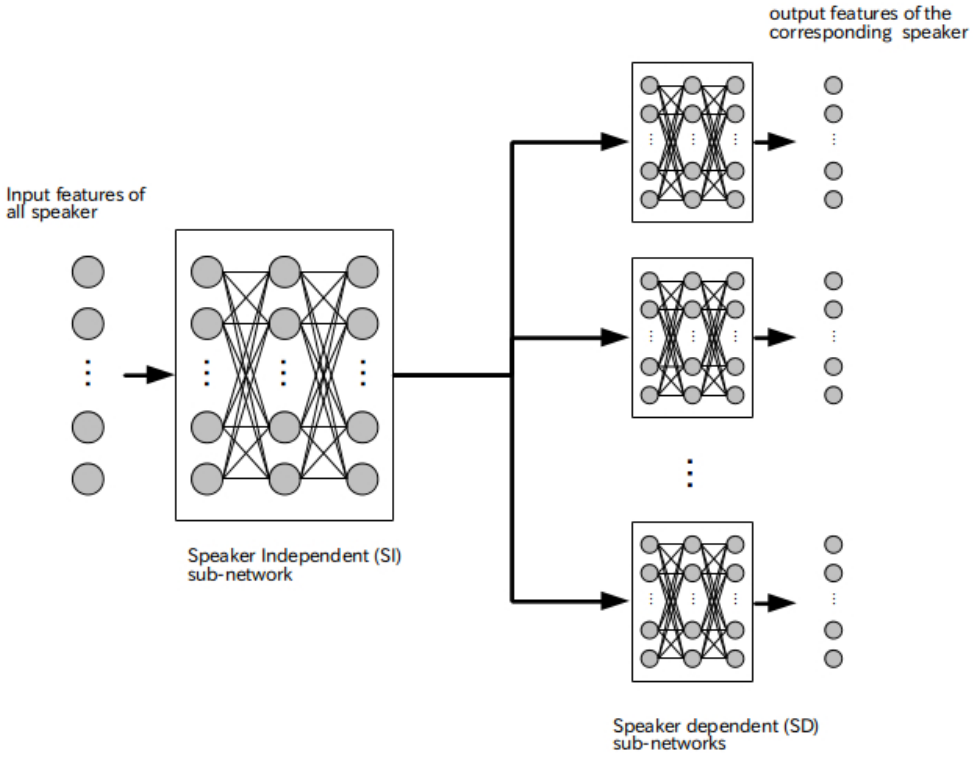

図 5.1: Configuration of the proposed method: Deep neural network with speaker independent and dependent sub-networks.

DNN においてこのような処理を可能とするための枠組みとして，多言語音素認識という複数の言語による音声中の音素を識別するという一種の clustering タスクに提案されている言語非依存サブネットワークの手法を参考にした [12].

5.2　マルチ出力サブネットワークを用いた DNN による声質変換

5.2.1　提案手法の概要

本節では，松田らによる多言語音声を学習した Deep Neural Network における言語非依存サブネットワークの自動適応の手法を参考とした，提案手法であるマルチ出力サブネットワークを持つ DNN の構造について説明する．提案手法では，DNN に対して変換先のターゲット話者毎に異なる話者依存サブネットワークを導入し，Eigen-voice conversion のように複数の話者からなるコーパスを用いて声質変換モデルの学習を行う．手法の概要を図 5.1 に示す.

提案手法では松田らの手法と同様の仮定を置く，すなわち，Deep Learning では前半の浅い層において, 殆どの音響的事象に共通する時間変動や周波数などの特徴量の識別を行っており, 逆に深い層では音素や言語依存の特徴量などの複雑な情報を扱っていると考えているこの過程を基に，提案手法における DNN は，1) ソース話者，ターゲット話者に依存しない特徴量抽出器のような処理を行うと考えられる入力層近傍のサブネットワーク (SI サブネットワーク)，そして 2) 話者性の再構成のような処理を行うと考えられる出力用の複数のサブネットワーク (SD サブネットワー

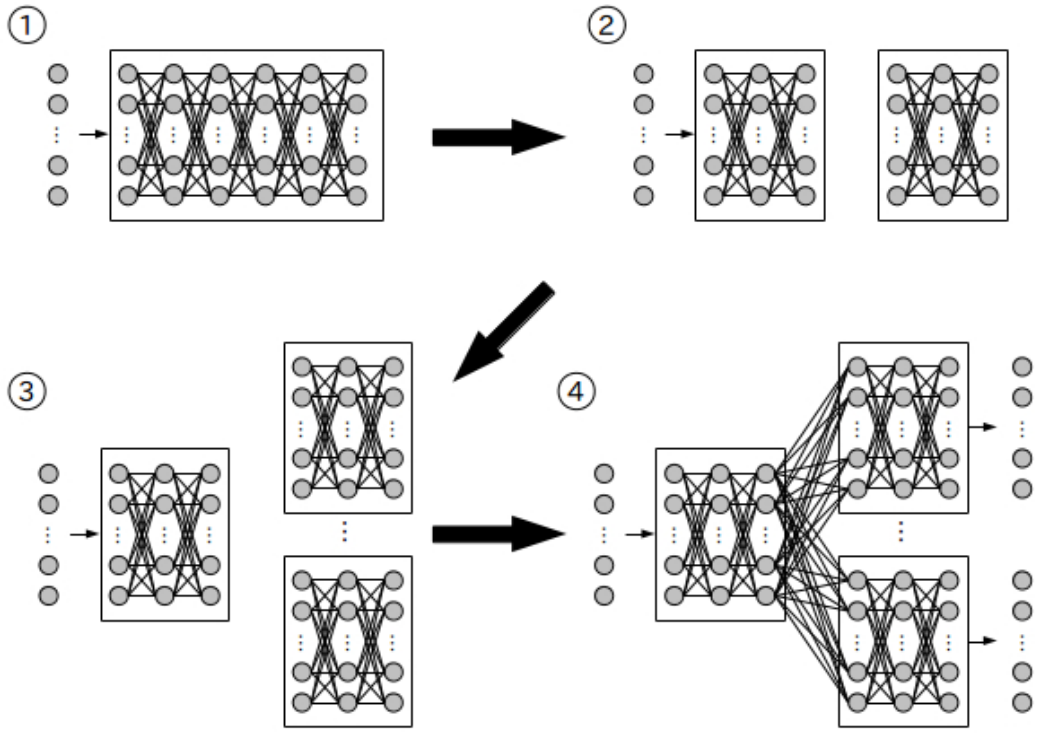

図 5.2: 提案手法における pre-training

ク)，という 2 つのサブネットワークによって構成される．このネットワークを，複数の話者からなるパラレルコーパスによって学習する．ここに，あるソース話者の特徴量ベクトルを入力することで，複数の SD サブネットワークによってソース話者から各ターゲット話者に対して変換を行った複数の出力特徴量ベクトルが得られる．

i)　提案手法における pre-training

学習では，初めに pre-training の処理を行う．提案手法における pre-training の概要を図 5.2 に示す．pre-training では，通常の Deep Belief Netwotk (DBN) の学習方法と同様に，RBM によってネットワークの層を 1 層ずつ積み上げることで, ANN の各パラメータの初期値を計算する (図中の 1)．このとき，学習データとしてはコーパス中に含まれる全ての話者の特徴量を使用する．次に，出力層付近の何層かのネットワークを学習データ中の話者の数だけ複製し，話者依存マルチ出力サブネットワークとする (図中の 2，3)．この複数の話者依存サブネットワークを元のネットワークに接続し，途中で枝分かれしているような構造とする (図中の 4)．これにより，1 つの根ネットワークとマルチ出力を為す複数の枝サブネットワークによって構成される DNN が得られる．

ii)　提案手法における fine-tuning

次に，fine-tuning を行う．fine-tuning の際には，ある話者をソース話者とした場合，その話者を含むコーパス中の全ての話者をターゲット話者と考え，モデルの学習に用いる (コーパスに N 人の話者データが含まれているとすると，一人の話者から N 人の話者への変換を同時に学習する)．これは，特徴量の次元を D とすると，入力 D 次元，出力 ND 次元の変換を行っているとも考えられる．複数の話者の特徴量間でアラインメントを取るため，本手法では，DP マッチングによる特徴量のアラインメントを，各発話毎にデータの系列長が最も長いデータの長さを基準とし，データの系列長が短いデータを引き伸ばすことでマッチングを行う．具体的には，2.4 節で示したアラインメントの経路重みを計算する配列 H を以下の様に変形する．

$$H_{ij} = \min \begin{cases} H_{i-1,\ j-1} & + & 2D_{ij} \\ H_{i,\ j-1} & + & D_{ij} \end{cases} \quad (5.1)$$

このとき，x 軸 (変数 i) に相当する特徴量系列には各発話毎に最も発話長の長いデータを用いるとする．これにより，全ての話者の特徴量系列の長さを統一し，入力 D 次元，出力 ND 次元の変換として誤差逆伝搬法による学習を行うことができるようになる．

iii)　提案手法の利点

提案手法の pre-training について考える．提案手法の pre-training では 4.4 節で挙げた DBN による声質変換手法と同様に RBM を複数積み上げる手法を用いている．[9] の手法では，深い階層を持つ DBN では各層のノード数で入力特徴量を表現するため，層の数が増えるほど入力特徴量が話者性の薄れた基底集合に近くなると仮定している．これは，一対一の変換では入力側・出力側それぞれの RBM の学習に用いられる特徴量の話者が一定であるため，話者性という学習データに共通の情報が層間の変換によって表され，言語情報 (音素情報) という特徴量のフレーム毎に異なる情報が深い層 (入力から遠い層) に集約されるという仮定である．提案手法の場合を考えると，RBM によって学習する特徴量は複数の話者からなる音声コーパスであるため，学習に用いられる特徴量の話者が一定ではなく，[9] らと同様の仮定を置くことはできない．一方で，入力されるデータが複数の話者からなる音声コーパスであるために，特徴量のフレーム毎に異なる情報，すなわち話者性に関しては RBM の深い層に集約されると考えられる．これにより，通常の 1 話者のデータによる pre-training よりもより話者性をよく表現する特徴量の抽出が可能になっていると考えられる．

提案手法による fine-tuning について考える．提案手法による変換では，複数話者からなる学習データは常に根ネットワークである SI サブネットワークを経て，SD サブネットワークとの結合部分でコーパス中の話者の数に分岐する．一方で，枝ネットワークである SD サブネットワークでは変換先の話者は常に固定である．誤差逆伝搬法によって出力層での誤差が伝搬していくことを考えると，SI サブネットワークと SD サブネットワークとの結合部分では，1 人のソース話者とコーパス中の全ての話者をターゲット話者と考えたときの誤差が伝搬することとなり，この学習を全ての話者をソース話者として行うことになる．これにより，SI サブネットワークにおける SD サブネットワークとの結合部分付近では，1 人の話者と複数の話者との誤差を最小化するような学習が行われる．そのため，SI サブネットワークの結合部分付近では話者性を正規化，または除去するような変換が学習され，入力された特徴量から話者非依存な特徴量を抽出する一種の特徴量抽出器の働きを持つことが期待される．一方で，各 SD サブネットワークはそのサブネットワークに対応する話者をターゲットとしたデータによってのみ学習されるため，この SD サブネッ

トワークが話者非依存な特徴量からターゲット話者依存な特徴量の再構成器としての働きをすることが期待される．

また，提案手法は話者変換以外のタスクへの利用も考えられる．例として，声質変換におけるもう 1 つの課題である感情変換・感情付与を考えると，話者変換において複数話者のデータを入力し各話者への変換をそれぞれのサブネットワークで学習していたものを，複数の感情データを入力とし，出力のサブネットワークを個別の感情への変換用として学習を行えばよい．その他にも，出力に新しい話者を加える際には，前半の SI サブネットワークの値は学習済みのものをそのまま使用し，SD サブネットワークのみを pre-training 済みの状態から fine-tuning によって更新することで，計算量を削減することができるなどの応用が考えられる．

第6章

実験・評価

6.1　実験の概要

提案手法による声質変換の有効性を示すために評価実験を行った．具体的な実験としては，以下のものを行った．

1. pre-training に複数話者のデータを用いることによる変換精度への影響の確認
2. 学習データ中の話者に関する従来手法と提案手法の間の変換精度の比較
3. 提案手法の変換精度が学習に用いる話者数によってどのように変化するかの確認
4. 学習データ外の未知話者に関する従来手法と提案手法の間の変換精度の比較

6.1.1　pre-training に複数話者のデータを用いることによる変換精度への影響の確認

i)　目的・実験条件

提案手法と通常の DNN の違いとして，pre-traning と fine-tuning の両方に複数話者のデータを用いるという点がある．そのため，提案手法全体の精度を比較しただけでは，その精度の差が pre-training によるものなのか fine-tuning によるものなのか，または両方が組み合わせられたことによるものなのかが判別できない．そこで初めに予備実験として，pre-training に用いる話者数及びデータ数を変化させた際の通常の DNN による一対一声質変換の精度比較を行った．

実験においてデータセットとしては，ATR 日本語音声データベースの B セット [19] を使用した．ATR 日本語音声データベースから男性話者 3 名 (以下 M1，M2，M3 とする) を選択し，各話者について学習に 50 文 (subset A)，テストに 50 文 (subset B) の計 100 文を用いた．特徴量としては，STRAIGHT 分析 [20] によって得られたメルケプストラムの 0 次元目 (パワーに相当する) を除く 24 次元を使用した．非周期性指標に関しては今回の実験では変換を行わず，全ての帯域に関して −30dB で固定とした．パワーと基本周波数に関しては，学習データから得られる平均と分散による線形変換によって変換を行った．

実験に使用する DNN の構成としては，層数を 6 層，隠れ層のノード数を 1024 ユニットとした (入力・出力は 24 次元)．変換精度は変換された特徴量と正解データの特徴量の間で，式 (6.1) で表されるメルケプストラム歪み (Mel cepstral distortion:Mel-CD) をとり，客観評価を行った．

$$\mathrm{MelCD[dB]} = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{24}(mc_d - \bar{mc}_d)^2} \tag{6.1}$$

ここで，c_d と $\tilde{c}_d$ はターゲット話者の正解データの特徴量ベクトルと，ソース話者の特徴量ベクトルから変換された特徴量ベクトルをそれぞれ表す．

pre-training に複数話者のデータを用いることによる変換精度への影響を確認するため，学習に用いる話者 M1，M2，M3 のデータ量を 5 通りに変化させ，各条件において pre-training を行った通常の DNN によって声質変換を行った．実際に変換を行う話者はソース話者を M1，ターゲット話者を M3 とした．

6.1.2　結果・評価

各条件における変換の結果を表 6.1 に示す．Condition A から D の結果から，より多くの話者を pre-training に用いた場合の方が変換精度が良くなっていることが分かる．また Condition D

表 6.1: 学習に使用した各話者のデータ数毎の客観評価結果 ((M1/M2/M3) : 話者 M1，M2，M3 から使用した文の数)

Condition	#utterances	Mel-CD[dB]
A	(50/0/0)	4.293
B	(50/50/0)	4.281
C	(50/0/50)	4.270
D	(50/50/50)	4.270
E	(150/0/0)	4.264

と E の結果から，複数の話者のデータを学習に用いた変換精度とソース話者のデータをより多く用いた変換精度がほぼ同程度となっており，pre-training に複数の話者のデータを学習に使用することは，変換に直接関与していない話者であっても有用であると考えられる．

6.2　学習データ中の話者に関する従来手法と提案手法の間の変換精度の比較

6.2.1　目的・実験条件

次に，学習データ中の話者に関する提案手法の変換精度を，既存手法である GMM，DNN と比較することで提案手法の有効性を確認した．

6.2 節の実験条件で述べたように，提案手法と通常の DNN の違いとして，pre-traning と fine-tuning の両方に複数話者のデータを用いるという点がある．pre-training に用いる話者数の増加による変換精度の向上が確認されたため，実験条件を平等にするため，既存手法である DNN の pre-training においても提案手法と同じようにコーパス中の話者を全て用いて学習を行った．

実験では実験 (1) と同様に ATR 日本語音声データベースから男性話者 3 名 (以下 M1，M2，M3 とする) を選択し，各話者について学習に 50 文 (subset A)，テストに 50 文 (subset B) の計 100 文を用いた．具体的な実験としては，提案手法と GMM および DNN の 3 手法に対して，話者 M1 から M2 への変換と，話者 M1 から M3 への変換の精度比較を行った．また，pre-training と同様に，提案手法ではソース話者として複数の話者を用いて fine-tuning を行っており，学習データの条件が他の 2 手法と異なる．そこで，より正確な比較を行うために，GMM と DNN でもソース話者に関して以下の 2 種類の学習を行ったものを用意した．

- 実際に変換を行う話者間の変換のみを学習 (pair-specific)
- 変換先の話者を固定して残りの 2 話者をソース話者として学習 (target-specific)

例として，話者 M1 から M2 への変換を学習する場合，前者では話者 M1 をソース話者として話者 M2 をターゲット話者として学習を行い，後者では話者 M1 をソース話者として話者 M2 をターゲット話者とした学習に加えて話者 M3 をソース話者，話者 M2 をターゲット話者とした変換の学習も行う．

提案手法では，SD サブネットワークの複製を行った後に，話者 M1，M2，M3 をそれぞれソー

表 6.2: 3 手法による声質変換の客観評価結果．(Pair-specific：実際に変換を行う話者間の変換のみを学習したモデル，Target-specific：変換先の話者を固定して残りの 2 話者をソース話者として学習したモデル)

Methods	Mel-CD [dB] (M1 to M2)	Mel-CD [dB] (M1 to M3)
GMM (pair-specific)	4.324	4.313
GMM (target-specific)	4.417	4.571
DNN (pair-specific)	4.290	4.270
DNN (target-specific)	4.290	4.241
Proposed	**4.256**	**4.200**

ス話者とターゲット話者の両方に使用し，fine-tuning を行った．

実験に使用する DNN の構成としては実験 (1) と同様に層数を 6 層，隠れ層のノード数を 1024 ユニットとし (入力・出力は 24 次元)，SI サブネットワークと SD サブネットワークをそれぞれ 3 層とした．

6.2.2　結果・評価

表 6.2 に各手法の客観評価の結果を示す．表 2 から，従来手法である GMM と DNN とで異なる傾向が見られる．GMM では，target-specific な学習を行ったモデルが pair-specific な学習を行ったモデルに対して変換精度が下回っているのに対して，DNN では 2 種類のモデルがほぼ同等の変換精度か，target-specific な学習を行ったものの方が僅かに上回っている．この結果は，複数の話者を用いることで DNN の浅い層が効果的に学習されているためと考えられる．また，提案手法と他の 2 手法の変換精度を比較すると，話者 M1 から M2 への変換と話者 M1 から M3 への変換の両方において提案手法が従来手法を上回っていることが分かる．

6.3　提案手法に用いる学習話者数による変換精度の変化

6.3.1　目的・実験条件

提案手法において，学習する話者の数が変換精度にどのような影響を与えるかを確認するため，pre-training と fine-tuning に用いる話者数を 3 話者，6 話者，9 話者とした場合 (それぞれ spk3, spk6，spk9 とする) の変換精度の比較を行った．コーパスとしては多数話者データベース [1] から音素バランス文 50 文 (APP-BLA) を使用し，男性話者 9 名の各話者 50 文のデータを，学習 40 文とテスト 10 文に分けて実験を行った．

また，メルケプストラム歪みによる客観評価はソース話者とターゲット話者によって値の範囲にばらつきがあるため，各条件の比較を行う場合には，比較を行う条件間で共通して用いられている話者ペアに関して，メルケプストラム歪みの平均を取ったものによって比較を行った．例として，spk6 と spk9 の比較を行う際には，spk9 の変換精度は spk6 で使用している 6 話者に関する変換結果を平均したものとし，残りの 3 話者の結果は含まないものとする．

[1] http://www.atr-p.com/products/sdb.html#MS

表 6.3: 学習に用いる話者数の変化に対する提案手法の客観評価結果 (テストデータ 3 話者)

#speakers	Mel-CD [dB]
spk3	4.637
spk6	4.459
spk9	4.460

表 6.4: 学習に用いる話者数の変化に対する提案手法の客観評価結果 (テストデータ 6 話者)

#speakers	Mel-CD [dB]
spk3	-
spk6	4.547
spk9	4.520

実験に使用する DNN と提案手法の構成としては実験 (1) と同様に層数を 6 層，隠れ層のノード数を 1024 ユニットとし (入力・出力は 24 次元)，SI サブネットワークと SD サブネットワークをそれぞれ 3 層とした．

6.3.2　結果・評価

表 6.3 に 3 話者 (6 通り) の変換に関するメルケプストラム歪みの平均値を，表 6.4 に 6 話者 (30 通り) の変換に関するメルケプストラム歪みの平均値をそれぞれ示す．表 6.3 および表 6.4 から，多くの話者を学習に用いたモデルの方がより精度の高い変換を行っており，提案手法の学習に複数の話者を用いることの有効性が示されている．

6.4　学習データ外の未知話者に関する従来手法と提案手法の間の変換精度の比較

6.4.1　目的・実験条件

次に，学習データ外の話者 (未知話者) に関する提案手法の変換精度を，既存手法である GMM, DNN と比較することで提案手法の有効性を確認した．

学習データとしては実験 (3) と同様に，多数話者データベースから音素バランス文 50 文 (APP-BLA) を使用し，男性話者 10 名 (M1, M2, ..., M10 とする) の各話者 50 文のデータを，学習 40 文とテスト 10 文に分けて実験を行った．この話者 10 名のうち 9 名を学習に使用し，残りの 1 名 (M10) を未知話者とした．具体的な実験としては，提案手法と DNN の 2 手法に対して学習に用いていない未知話者 M10 から話者 M1，M2，M3 それぞれへの変換の精度比較を行った．

DNN の pre-training には話者 M1 から話者 M9 の 9 話者のデータを使用した．DNN の fine-tuning に関しては実験 (2) と同様に DNN におけるソース話者に関して以下の 2 種類の学習を行ったものを用意した．

- 特定の話者からの変換のみを学習 (pair-specific)
- 変換先の話者を固定して残りの 8 話者をソース話者として学習 (target-specific)

表 6.5: 提案手法と DNN における未知話者入力に対する声質変換の客観評価結果．(Pair-specific : 実際に変換を行う話者間の変換のみを学習したモデル，Target-specific : 変換先の話者を固定して残りの 2 話者をソース話者として学習したモデル)

Methods	Mel-CD [dB] (M10 to M1)	Mel-CD [dB] (M10 to M2)	Mel-CD [dB] (M10 to M3)
DNN (pair-specific)	4.869	5.457	4.946
DNN (target-specific)	4.794	4.968	4.685
Proposed(spk3)	4.832	5.057	5.267
Proposed(spk6)	**4.701**	4.935	4.582
Proposed(spk9)	4.715	**4.908**	**4.547**

ただし，pair-specific に関してはターゲット話者が M1 のときは M2 をソース話者とし，それ以外の話者がターゲット話者のときは M1 をソース話者とした．例として，話者 M10 から M1 への変換を考えた場合，前者では話者 M2 をソース話者として話者 M1 をターゲット話者として学習を行い，後者では話者 M2 から話者 M9 までをソース話者，話者 M1 をターゲット話者とした学習を行う．

提案手法の pre-training にも同様に話者 M1 から話者 M9 の 9 話者のデータを使用し，fine-tuning では，学習に使用する話者数を 3 話者，6 話者，9 話者の 3 条件 (それぞれ spk3，spk6，spk9 とする) に変化させ，それぞれソース話者とターゲット話者の両方に使用して学習を行った．

実験に使用する DNN と提案手法の構成としては実験 (1) と同様に層数を 6 層，隠れ層のノード数を 1024 ユニットとし (入力・出力は 24 次元)，SI サブネットワークと SD サブネットワークをそれぞれ 3 層とした．

6.4.2　結果・評価

表 6.5 に DNN と提案手法における未知話者を入力とした際の話者 M1，M2，M3 への変換精度を示す．表 6.5 から，話者 M1，M2，M3 の全てに対する変換において，6 話者・9 話者を学習に用いた提案手法が従来手法である DNN の変換精度を上回っている．

DNN においては，実験 (2) での傾向と同様に，pair-specific な fine-tuning よりも target-specific な fine-tuning を行った場合の方が変換精度が高くなっている．この結果からも，DNN の学習に複数の話者を用いることの有効性が示されているといえる．

一方で，提案手法において学習に 3 話者を用いたものは，target-specific な DNN よりも精度が悪くなっている．これは，学習に用いる話者数が十分でないときには局所解に陥りやすく，未知話者と話者性が近い話者を学習に用いているかが変換結果に大きな影響を与えるためだと考えられる．

また，話者 M1 への変換では 6 話者を用いて学習を行ったときよりも，9 話者で学習を行ったときの方が変換精度が悪くなっている．これは，実験 (3) でも同様の傾向が見られていることから，コーパス中の話者 M7 から M9 の中に話者 M1 への変換が難しく，他の話者からの変換に悪影響を与えている話者がいるためだと考えられる．

第7章

結論

7.1　本研究のまとめ

本研究では，話者性の柔軟な制御を目的とした DNN による声質変換手法を提案した．話者性の柔軟な制御を目的とした手法としては，既存のモデルからパラメータを適応するというものが一般的であり，声質変換においては GMM による実装が多く取られている．この GMM による Eigen-voice conversion や MAP 適応などの話者適応型声質変換手法を例に挙げ，複数話者コーパスを用いた学習を行うことで話者に依存していない，音声に共通な特徴量を抽出することが有用であると考えた．

また，DNN を用いた声質変換手法として，Deep Belief Nets による低次元空間表現を用いた声質変換という手法を挙げ，そこで用いられている,「深い階層を持つ DBN では各層のノード数で入力特徴量を表現するため，層の数が増えるほど入力特徴量が基底集合に近くなる」という考えを参考とした．この考えを仮定すると，DNN の pre-training において入力されるデータを複数の話者からなる音声コーパスとすることで，特徴量のフレーム毎に異なる情報，すなわち話者性が RBM の深い層に集約されると考えた．

そこで，多言語音素認識タスクに用いられている手法を参考に，1 つの話者非依存サブネットワークと複数の話者依存サブネットワークからなる DNN による声質変換の枠組みを提案した．この手法では pre-training を複数の話者によって行うことで，前述したように話者性を深い層に集約し，その上で fine-tuning を出力話者毎の話者依存サブネットワークと話者非依存サブネットワークを導入して行うことで，話者非依存サブネットワークと話者依存サブネットワークの分岐点に集約した話者性を正規化するように学習を行う．これにより，学習に用いていない未知話者入力に対しても話者性が正規化されるために，柔軟な変換が可能となると考えられ，また，ANN において問題であった入力層に近い浅い層の学習をより効率的に行うことが可能となると考えられる．

実験の結果，pre-training に複数話者を用いることによる変換精度の向上と，提案手法による学習データ中の話者と未知話者の両方の入力に対する変換精度の向上を確認し，提案手法が入力話者に対して柔軟な変換が可能であることを示した．

今後の課題としては，応用としては固有声変換のような手法との組み合わせを考えている．固有声変換やテンソル表現を用いた声質変換では，あるターゲット話者のモデルは複数の話者モデルの足し合わせによって表現できるとされている．本手法では，1 話者の入力から複数の出力を得ることができるため，これらの手法のように複数の話者モデルへの分解や再合成といった応用が考えられる．

謝辞

本研究を進めるにあたって, 指導教員である広瀬啓吉教授，峯松信明教授，斎藤大輔助教授には研究についての助言から, 発表の場の斡旋, 論文の添削など様々な面で大変お世話になりました.

柏木陽佑さん, 橋本浩弥さんには週次ミーティングやグループ勉強会において研究についての様々な意見を頂いたり, 発表原稿や論文の添削などもしていただきました.

研究室の先輩方, 並びに同期の方々にもいつもお世話になり, 大変感謝しております. この場を借りて研究室の皆様方に心からお礼申し上げます.

参考文献

[1] A. Kain, and M. W. Macon, “Spectral voice conversion for text-to-speech synthesis,” ICASSP, vol. 1, pp. 285–288, 1998.

[2] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. Huang, “High-performance robust speech recognition using stereo training data,” ICASSP, pp. 301–304, 2001.

[3] D. Saito, H. Doi, N. Minematsu, and K. Hirose, “Application of matrix variate Gaussian mixuture model to statistical voice conversion,” Proc. Interspeech, pp. 2504–2508, 2014.

[4] S. Desai, A.W. Black, B. Yegnanarayana, and K. Prahallad, “Spectral mapping using artificial neural networks for voice conversion,” IEEE Trans. on Audio, Speech, and Language Processing, VOL. 18, no.5, pp. 954–964 , 2010.

[5] A. Mouchtaris, J.V. der Spiegel, and P. Mueller, “Nonparallel training for voice conversion based on a parameter adaptation approach,” IEEE Trans. on Audio, Speech, and Language Processing, vol. 14, no. 3, pp. 952–963, 2006.

[6] C.H. Lee, and C.H. Wu, “Map-Based Adaptation for Speech Conversion Using Adaptation Data Selection and Non-Parallel Training,” Proc. Interspeech, pp. 2254–2257, 2006.

[7] T. Toda, Y. Ohtani, and K. Shikano, “A Voice Conversion Algorithm Based on EigenVoice,” IEICE Technical Report SP2006–39, 2006.

[8] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition,” IEEE Signal Processing Magazine, pp. 82–97, 2012.

[9] 中鹿亘, 他, “Deep Belief Nets による低次元空間表現を用いた声質変換の検討”, 日本音響学会春季研究発表会講演論文集, 3–P–46b, pp. 517–520, 2013.

[10] Y. Ohtani *et al,* “Adaptive training for voice conversion based on eigenvoices,” IEICE TRANS. INF. &SYST., VOL.E93-D, no.6, pp. 1589–1598, 2010.

[11] D. Saito, K. Yamamoto, N. Minematsu, and K. Hirose, “One-to-Many Voice Conversion Based on Tensor Representation of Speaker Space,” Proc. Interspeech, pp. 653–656, 2011.

[12] S. Matsuda, X. Lu, and H. Kashioka, “Automatic localization of a language-independent sub-network on deep neural networks trained by multi-lingual speech,” Proc. ICASSP, pp. 7359–7362, 2013.

[13] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, “Voice conversion through vector quantization,” Proc. ICASSP, vol. 1, pp. 655–658, 1988.

[14] 中村哲, 鹿野清宏, “ファジィベクトル量子化を用いたスペクトログラムの正規化,” 日本音響学会学会誌, pp. 107–114, 1989.

[15] Y. Stylianou, O. Cppe, and E. Moulines, “Continuous probabilistic transform for voice conversion,” IEEE Trans. on Speech, and Audio Processing, vol. 6, no. 2, pp. 131–142, 1998.

[16] Y. Freund and D. Haussler, “Unsupervised learning of distributions on binary vectors using two layer networks,” Computer Research Laboratory, 1994.

[17] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, “ Extracting and composing robust features with denoising autoencoders,” Proc. ICML’08, pp. 1096–1103, 2008.

[18] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief net,” Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006.

[19] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, “ATR Japanese speechdatabase as a tool of speech recognition and synthesis,” Speech Communication, vol. 9, pp. 357–363, 1990.

[20] H. Kawahara, I. Masuda-Katsuse, and A.de Cheveigne, “Restructuring speech representations using a pitch-adaptive time frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds,” Speech Communication, vol. 27, pp. 187–207, 1999.

発表文献

国内研究会・全国大会

[1] 橋本 哲弥，柏木 陽佑，齋藤 大輔，広瀬 啓吉，峯松 信明，“話者依存サブネットワークを用いた深層学習による多対一声質変換，”日本音響学会秋季講演論文集, 2-Q-38, pp. 353–356, 2014.

[2] 橋本 哲弥，柏木 陽佑，齋藤 大輔，広瀬 啓吉，峯松 信明，“複数出力サブネットワークを有するディープニューラルネットに基づく声質変換，”信学技報, vol. 114, no. 365, SP2014–117, pp. 99–104, 2014.