ニューラルネットワークの音声情報処理への応用

関西支部講演会

鹿野清宏*、片桐滋**、中村雅己*、田村震一*、アレックス ワイベル*、
沢井秀文*、パトリック ハフナー*、東倉洋一**、エリック マクダーモット**、船橋賢一**

(*ATR自動翻訳電話研究所、**ATR視聴覚機構研究所)

1 まえがき

ニューラルネットワークのアプローチが応用の面で大きな展開を見せるきっかけとなったのは、Back-Propagation学習法の登場によるところが大きい[Rumelhart 86a][Rumelhart 86b]。この学習法により、従来困難とされていた多層ネットワークの学習が可能となり、ネットワーク中に隠れユニットを想定することで、2層パーセプトロンでは不可能であったXORやパリティ関数の問題が学習可能であることが示された。このBack-Propagation学習のアルゴリズムは計算機上で実現し易いものであるため、各方面の研究者がそれぞれの応用モデルを構成し始め、ニューラルネットワークのアプローチが再び活況を見せ始めたのである。とくに、高速の並列処理計算機が利用可能になり、大規模並列(Massively Parallel)処理も身近な話題になってくるとともに、ニューラルネットワークの研究が再びブームになってきた。

音声の分野でも、音声の規則合成での文字から音韻への変換規則の発見に3層のニューラルネットワークとBack-Propagation アルゴリズムが用いられ、成功をおさめ、注目をあびた[Sejnowski 86]。音声認識の分野でも、ニューラルネットワークを用いた少数カテゴリの認識が行われ、従来技術に匹敵する認識率が報告され始めた[Huang 87a][Lubensky 88][Leung 88]。このように、現在では、単なる脳の構造との類似性だけでなく、工学的にも有効である可能性がでてきたため非常に注目されている。とくに、多層のパーセプトロンの重みをBack-Propagation アルゴリズムで学習させる手法が広く使われている。多層パーセプトロンの能力も実験的に調べられている[Lippmann 87]。

2章ではこの多層パーセプトロンのBack-Propagation学習のアルゴリズムについて解説する。3章では、音声認識への適用例として、Time Delay Neural Network (TDNN) [Waibel 87] [Waibel 88] [Sawai 88]を簡単に紹介する。これは、カテゴリ分類の枠組みでの適用例であるが、その他の適用例として、4章では、空間の写像による雑音抑圧への適用例[Tamura 88a]を、5章では、規則の発見への適用例として、音声の規則合成での文字から音韻への変換規則の発見[Sejnowski 86]と、単語カテゴリの予測問題[Nakamura 88]を取上げ、簡単に紹介する。さらに6章では、Back-Propagation以外のニューラルネットワークへの1アプローチとしてKohonen の Learning Vector Quantization に基づいた音韻認識について紹介する[McDermott 88b]。

2. 多層パーセプトロン

2.1. Back-Propagation学習アルゴリズム

Back-Propagation学習アルゴリズムが対象とするネットワークは入力層、出力層およびHidden-Layerと呼ばれる隠れ層で構成される多層ネットワークである(図2-1)。Back-Propagation学習アルゴリズムの特徴は、教師付き学習において出力層から入力層に誤差を伝搬させることにあり、これにより各ユニットにおいて最急降下法を適用することが可能となった。もう一つの特徴は、各ユニットに非線形関数を導入したことであり、これにより入力から出力への写像に、より一般性を持たせることができた。ここではBack-Propagation学習のアルゴリズムについて数式を追って説明する。

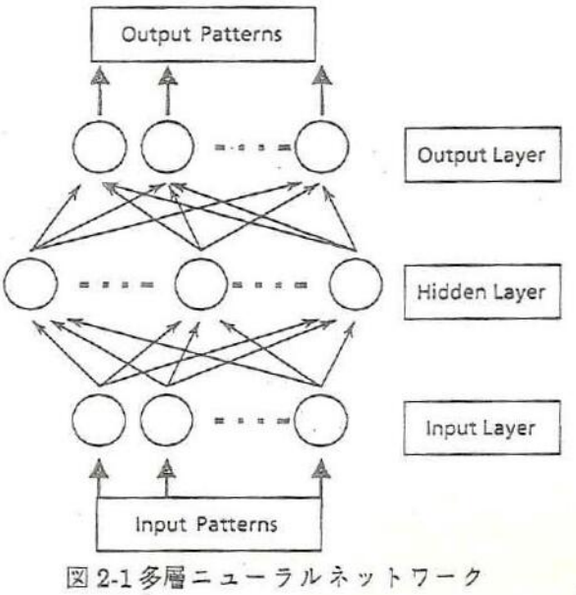

図2-1 多層ニューラルネットワーク

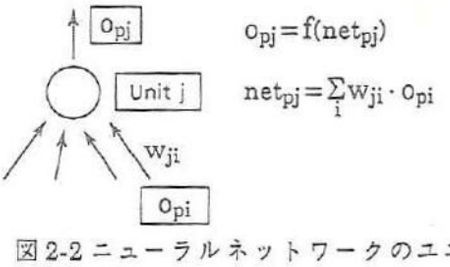

図2-2 ニューラルネットワークのユニット

ネットワークのユニット間は入力層から出力層に向かって結合されており、入力層にある入力パターンpを入力したとき、(図2-2)に示すようにそれぞれのユ

ニットjでは他のユニットiからの入力、すなわちユニットiの出力と、ユニットi,j間の結合重み係数w_{ji}の積の総和$net_{pj}=\Sigma w_{ji}\cdot o_{pi}$をとり、さらに入出力関数f(x)を通して、出力信号$o_{pj}=f(net_{pj})$を出す。すなわち、ある入力信号のパターンをネットワークの入力層にいれたときに、上記のような計算を全てのユニットで行い(但し、入力層では入出力関数を通さないことが多い)、最終的に出力層から出た信号パターンが望ましいパターンになるように、ユニット間の結合重み係数を決定すればよい。

まず、この問題を最小化問題として定式化するために式(2-1)の評価関数を決める。

$$E_p=\frac{1}{2}\sum_j(t_{pj}-o_{pj})^2 \qquad (2\text{-}1)$$

ここでは教師信号とニューラルネットの出力信号の誤差の2乗和E_pを用いている。ここでt_{pj}は入力パターンpに対する出力ユニットjの教師信号であり、o_{pj}は出力ユニットjの出力信号である。この誤差関数E_pを全ての入力パターンに対して最小にする必要がある。よって問題は$E=\Sigma E_p$を最小にするような結合重み係数を決定するという、最小化問題となる。

この問題を解くために、Back-Propagation学習アルゴリズムでは最急降下法を用いている。すなわち、入力パターンp毎の結合重み係数w_{ji}の更新量$\Delta_p w_{ji}$を次式のようにエラー空間の勾配に比例した形で与える。

$$\Delta_p w_{ji}\propto-\frac{\partial E_p}{\partial w_{ji}} \qquad (2\text{-}2)$$

まず、$\partial E_p/\partial w_{ji}$を求める。

$$\frac{\partial E_p}{\partial w_{ji}}=\frac{\partial E_p}{\partial net_{pj}}\cdot\frac{\partial net_{pj}}{\partial w_{ji}} \qquad (2\text{-}3)$$

$net_{pj}=\Sigma w_{jk}\cdot o_{pk}$であるから右辺の右側は

$$\frac{\partial net_{pj}}{\partial w_{ji}}=\frac{\partial}{\partial w_{ji}}\sum_k w_{jk}\cdot o_{pk}=o_{pi} \qquad (2\text{-}4)$$

である。ここで添字kはユニットjに入力結合するすべてのユニットの番号である。次にユニットjに対して

$$\delta_{pj}=-\frac{\partial E_p}{\partial net_{pj}} \qquad (2\text{-}5)$$

とおくことにより、

$$-\frac{\partial E_p}{\partial w_{ji}}=\delta_{pj}\cdot o_{pi} \qquad (2\text{-}6)$$

よって、(2-2)式の学習規則は

$$\Delta_p w_{ji}=\eta\cdot\delta_{pj}\cdot o_{pi} \qquad (2\text{-}7)$$

となる。ここでηはステップサイズを決定する正の定数である。

次にδ_{pj}を求める必要がある。

$$\delta_{pj}=-\frac{\partial E_p}{\partial net_{pj}}=-\frac{\partial E_p}{\partial o_{pj}}\cdot\frac{\partial o_{pj}}{\partial net_{pj}} \qquad (2\text{-}8)$$

ここで、$o_{pj}=f(net_{pj})$であるから右辺の右側は

$$\frac{\partial o_{pj}}{\partial net_{pj}}=f'(net_{pj}) \qquad (2\text{-}9)$$

である。(2-8)式の右辺の左側はユニットjが出力ユニットか、そうでないかによって式は異なる。ユニットjが出力ユニットの場合、

$$E_p=\frac{1}{2}\sum_j(t_{pj}-o_{pj})^2 \qquad (2\text{-}1)$$

であるから、

$$\frac{\partial E_p}{\partial o_{pj}}=-(t_{pj}-o_{pj}) \qquad (2\text{-}10)$$

となり、直接δ_{pj}が次式のように求まる。

$$\delta_{pj}=(t_{pj}-o_{pj})\cdot f'(net_{pj})\ (\text{ユニットj;出力ユニット}) \qquad (2\text{-}11)$$

一方、ユニットjが出力ユニットでない場合、E_pがo_{pj}の直接の関数とならない。従って、次式のように変形してδ_{pj}の再帰関数として求めるという工夫を行う。

$$\begin{aligned}\frac{\partial E_p}{\partial o_{pj}}&=\sum_k\frac{\partial E_p}{\partial net_{pk}}\cdot\frac{\partial net_{pk}}{\partial o_{pj}}\\&=\sum_k\frac{\partial E_p}{\partial net_{pk}}\cdot\frac{\partial}{\partial o_{pj}}\left(\sum_i w_{ki}\cdot o_{pi}\right)\\&=\sum_k\frac{\partial E_p}{\partial net_{pk}}\,w_{kj}=-\sum_k\delta_{pk}w_{kj}\end{aligned} \qquad (2\text{-}12)$$

$$\delta_{pj}=f'(net_{pj})\cdot\sum_k\delta_{pk}w_{kj}\ (\text{ユニットj;隠れユニット}) \qquad (2\text{-}13)$$

このように、$\Delta_p w_{ji}$を計算するのに必要な誤差情報δ_{pj}を出力層から入力層へ逆に伝搬して行くのでBack-Propagationという。

Back-propagation学習アルゴリズムではユニットの入出力関数f(x)として、式(2-11)、(2-13)から明らかなように微分可能な関数が必要とされる。[Rumelhart 86a]では次のような非線形単調増加のSigmoid関数を用いるのが良いとしている。

$$f(x)=\frac{1}{1+e^{-x}} \qquad (2\text{-}14)$$

すなわち、ユニットjの出力は次式のようになる。

$$f(net_{pj})=o_{pj}=\frac{1}{1+e^{-net_{pj}}} \qquad (2\text{-}15)$$

ここで入力の総和net_{pj}は$net_{pj}=\Sigma w_{ji}\cdot o_{pi}+\theta_j$としてバイアス成分$\theta_j$を加える。実際のネットワークでは入力ユニット以外のすべてのユニットと結合する、出力が常に1のバイアスユニットを考え、θ_jをその結合重み係数とみなして学習する。

$f(net_{pj})$の導関数を求めると

$$f'(net_{pj})=\frac{\partial o_{pj}}{\partial net_{pj}}=o_{pj}\cdot(1-o_{pj}) \qquad (2\text{-}16)$$

よって、結合重み係数w_{ji}の更新量$\Delta_p w_{ji}$は次の式で得られる。

$$\Delta_p w_{ji}=\eta\cdot\delta_{pj}\cdot o_{pi} \qquad (2\text{-}17)$$

ただし、ユニットjが出力ユニットの場合は

$$\delta_{pj}=o_{pj}\cdot(1-o_{pj})\cdot(t_{pj}-o_{pj}) \qquad (2\text{-}18)$$

であり、ユニットjが隠れユニットの場合は

$$o_{pj} = o_{pj}\cdot(1 - o_{pj})\cdot\sum_k o_{pk} w_{kj} \qquad (2\text{-}19)$$

である。

結合重み係数w_{ji}の更新は、入力パターンが複数個あるのが一般的であるので、1つの入力パターン提示毎に実行するか、次式のように全入力パターン提示後に

$$\Delta w_{ji} = \eta\cdot\sum_p(\delta_{pj}\cdot o_{pi}) \qquad (2\text{-}20)$$

として実行するか、2つの方法がある。また、結合重み係数w_{ji}の初期値をすべて同じ値にすると、隠れユニットの出力値がすべて同じになり、学習が進まないので、乱数等によりランダムな小さな初期値に設定する必要がある。

.3 Back-Propagation学習アルゴリズムの効率化

Back-Propagation学習アルゴリズムの基本原理は最急降下法であるため、最短距離で最小値に到達するためには、更新幅を無限小にする必要があるが、実際問題として、計算繰り返し回数が増加するため、収束速度は遅くなる。そこで、なるべく大きな更新幅(Δw_{ji})を得るために(2-7)式のηの値を大きくとりたいのであるが、更新方向が振動し易くなる。[Rumelhart 86a]では、前回の更新幅をモーメンタム量として次式のように加算することにより振動を抑制することを提案している。

$$\Delta_p w_{ji(n+1)} = \eta\cdot\delta_{pj}\cdot o_{pi} + \alpha\cdot\Delta_p w_{ji(n)} \qquad (2\text{-}21)$$

ここで、αはモーメンタム量を調整するパラメータである。このη,αは定数であるが、これらの最適な値(収束が早くなる値)はエラー空間の形状、すなわちタスクの種類やサンプルデータの量によって異なるはずであり、さらに学習進行の程度によっても変化すると考えられる。

この節では、この他のBack-Propagation学習アルゴリズムの効率化の試みとして、ステップ幅η、モーメンタム量αの値の数個のセットを試行して、最もエラーを小さくするセットを用いるDynamically Control training Parameter法(D.C.P.法)をまず紹介する[Nakamura 88]。続いて、Sigmoid関数のシフト、最急降下法の方向の利用、重みの更新周期の最適化の試みについても紹介する[Haffner 88]。

.1 D.C.P.法によるBack-Propagationアルゴリズムの効率化[Nakamura 88]

ステップ幅η、モーメンタム量αは、前にも述べたように、エラー空間の形状、すなわちタスクの種類やサンプルデータの量によって異なるはずであり、さらに学習進行の程度によっても変化すると考えられる。そこで筆者らは学習計算繰り返し毎に、次式によりエラーEpが最小となるようにη,αをダイナミックに変更する方法について比較実験を行った。

$$Ep(w_{ji(k)} + \Delta_p w_{ji(k)(\eta(k),\alpha(k))}) \qquad (2\text{-}22)$$

$$1 \quad m$$

ここでは、計算時間の関係からη,αの値を有限個用意して、その中からEpが最も小さくなるη,αを選択するようにした(Dynamically Control training Parameter, D.C.P.法)。実験は5章で述べる単語列予測モデル(Bigram)で行った。実験結果を表2-1に示す。

パラメーター定に比較して、D.C.P.法すなわちη,αをダイナミックに変更する方法は約4.3倍以上収束が早くなり(CASE1,CASE2)、パラメーター定の場合、$\eta=0.4$のときはエラーの振動が生じ易く(CASE3,CASE4)、$\eta=0.1$であれば収束が遅くなる(CASE2,CASE5)。αについては$\alpha=0.9$の場合、ηの値も大きければ不安定な状態が持続し(CASE3)、$\alpha=0$の場合、学習が進んでも収束速度が加速しないことがわかった(CASE5)。

また、エラーの収束判定で学習サンプル数を増加させる実験を行ったところ、αについては学習初期及びサンプル数が増加した場合、αは小さな値をとり、それ以外はほとんど$\alpha=0.9$を選択した。これは、学習の初期はリンクウェイトの修正方向が不安定であるため、αによる加速はオーバーシュートを起こし易いからであり、サンプル数が増加した場合も、エラー空間が変形するため過去のリンクウェイトの修正方向を引きずらない方がよいからであると考えられる。ηについてはサンプル数が増加するにしたがって減少していった。すなわち、D.C.P.法ではサンプル数の大きさに対して自動的にηの値を正規化する。D.C.P.法を採用することにより、タスクの種類や学習サンプルの量に対して最適に近いパラメータ値を自動選択し、学習の進行状況に応じてパラメータを自動調整しており、結果として学習の効率化が図れる。今回は1回の繰り返し計算毎に全てのパラメータの組合せに対し計算を行ったため、η 3種類、α 2種類の場合、1回の繰り返し計算に6倍の時間を必要とした。しかし、ある程度学習が進めば、η、αは同じ値を取ることが多くなり、毎回パラメータを変更する必要はないので、変更のインターバルを大きくとることによりこの問題は解消される。

2.3.2 最急降下法の改良[Haffner 88]

この節では、Back-Propagation学習アルゴリズムの効率化のもう一つの例として、ステップ幅の調整および重みの最適化の試みを紹介する。第3章で紹介する有声破裂音/b/,/d/,/g/を識別するTime-Delay Neural Networkは、783個の音韻パターンを訓練に用いて、Alliantミニスーパーコンピュータで4日かかった学習を、現在では、数分で終了するまでにアルゴリズムが高速化されている。

以下、[Haffner 88]のBack-Propagation学習アルゴリズムの高速化を簡単に紹介する。そこでは、高速化のために次の3つのテクニックを導入している。

(1) Sigmoid関数のバイアス

Back-Propagation のステップ幅は、Sigmoid 関数の値およびその微分値に依存している。Sigmoid 関数の値は、0 を遠ざかるにつれて急速に平坦になる。このため、Sigmoid 関数の微係数が 0 になり、重みの学習が遅くなる。これを防ぐために、Sigmoid 関数の微分値に一定の定数(例えば+0.1) を加えることが有効である。また、Sigmoid 関数自体を原点に対して対称にするために、-0.5だけ引き、Back-Propagation の学習アルゴリズムを変形することも若干有効である。

(2) ステップ幅のスケーリング

ステップ幅の最適な値は、学習の進行とともに誤差関数のローカルな形状によって大きく変わってゆく。この尺度として、繰り返し回数ごとの誤差関数のグラディエントの方向の cos 関数が有効であると提案されている [Franzini 87]。すなわち、

$$\theta = \mathrm{angle}(\nabla E_p(n-1), \nabla Ep(n)) \qquad (2\text{-}24)$$

となる。モーメンタムの項を用いている場合には、発振を防ぐために、

$$\theta = \mathrm{angle}(\Delta_p w_{ji}(n-1), -\nabla Ep(n)) \qquad (2\text{-}25)$$

を用いることが有効である。ここで、$\Delta_p w_{ji}$ は、重みの変化のベクトルを表す。ここで、上記のスケーリングをユニット u へ接続している重みのみに対して計算することを考える。このようにしても、(2-24)(2-25)式は同じように計算できる。すると、ユニット u への重みのステップ幅は、

$$\Delta_p w_{ji}(n) = \Delta_p w_{ji}(n-1)\, e^{(\rho \cos(\theta))} \qquad (2\text{-}26)$$

とくに、TDNNのように、それぞれのユニットが異なる役割を持っているときには、この手法は大きな改善効果を示す。ρの値として、ここでは、0.2 を用いた。しかしながら、この手法は、オーバーシューティングを起こし易い。このオーバーシューティングを防ぐために、$\Delta_p w_{ji}(n)\ \nabla Ep(n)$ のノルムに、例えば、1.0 のような一定値の上限を設定する。

(3) 更新周期の最適化

通常、訓練用データをすべて入力したのちに重みの更新を行っていることが多い。しかし、訓練データの量が増加するに従って、更新周期の設定が問題となってくる。多量の訓練データが与えられた場合には、これをいくつかの統計的に同じ性質をもつサブセットに分割して、サブセットの入力ごとに重みを更新することが、効率を大幅に向上させることになる。さらに、サブセット間の統計的なゆらぎが、アルゴリズムがローカルミニマムに収束することを防いでいる可能性も考えられる。

以上、述べてきた手法のうち (1) (2) の評価実験を、次の3つのタスクで行った。以下の実験では、訓練用データをすべて入力したのちに重みの更新が行われる。また、モーメンタムαの値を、0.9 に設定した。

(a) XOR: 排他的論理和(Exclusive-Or)の学習、

(b) 838: 8ユニット、3ユニット、8ユニットのコーディングの問題(8-3-8 Encoding),

(c) BDG: TDNNによる/b,d,g/の識別(250学習サンプル利用)。

実験結果を表2-2に示す。(2)のスケーリングの効果、(1)のSigmoid 関数への変形の効果がわかる。

上記の高速化手法をもちい、BDG タスクのフルスケール(783学習サンプル利用)を用いて、(3)の更新周期の最適化を試みた。24サンプル(音韻ごとに8サンプルずつ)ごとに重みを更新するときが最も効率がよく、Alliant ミニスーパーコンピュータ(8 CPU)を用いて5分以下の計算時間で学習が終了した。学習サンプル以外の評価用サンプルを用いた認識率での劣化はなかった。よって、[Waibel 87]での Alliant ミニスーパーコンピュータ(4 CPU)を用いた実験での処理速度(4 CPUで4日、すなわち、8 CPUで2日)と比較すると、約 600 倍の高速化が達成できたことになる。

3. 音声認識への適用

この章では、ニューラルネットワークによる音声認識への1アプローチとして、Time-Delay Neural Network (TDNN) による音韻認識の試みを紹介する。

3.1 少数カテゴリ(BDG)の音韻認識[Waibel 87]

ATR では、音声現象を考慮して、工学的立場から図 3-1 に示すような Time Delay Neural Network (TDNN)のアーキテクチャーを考案し、音韻認識の /b, d, g/ の識別で、表3-1 に示すようにHMM より大幅に高い認識率を達成した [Waibel 87]。ニューラルネットワークを用いてTime Shift Invariant な音韻認識を行う試みは、1984年に東北大学で行われているが [Kawabata 84]、特徴抽出層は固定されていた。これに対して、TDNNでは Back-Propagation 法によって特徴抽出層をも学習形成することで認識系の精度を飛躍的に高めて

表2-1 学習効率化策(D.C.P.法)の効果確認実験結果

(Bigram ネットワークモデル、1センテンス学習)

	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5
η (step size)	(1/2, 1, 2)	0.1(fix)	0.4(fix)	0.4(fix)	0.1(fix)
α(momentum)	(0.0 , 0.9)	0.9(fix)	0.9(fix)	0(fix)	0(fix)
Iteration	35	153	more than [illegible]	178	more than [illegible]

る。TDNNは、次の特徴を持っている。(1) 4層のニューラルネットワークの構成であるので、任意の判別境界をBack-Propagation アルゴリズムによる学習により構成することが原理的には可能である。(2) TDNNは、学習により、音韻のスペクトログラムから、音韻特徴を発見して、それに基づいて音韻を認識することを意図した構成になっている。(3) TDNNは、入力音韻の位置ずれに影響されにくい構成をとっている。また、このTDNNの隠れ層で用いられている特徴を調べた結果、音韻弁別特徴として知られている第2フォルマントの動きや母音の立ち上がり位置の検出を学習によって自動的に発見して、音韻識別の特徴の一部として用いられていることも確かめられている。

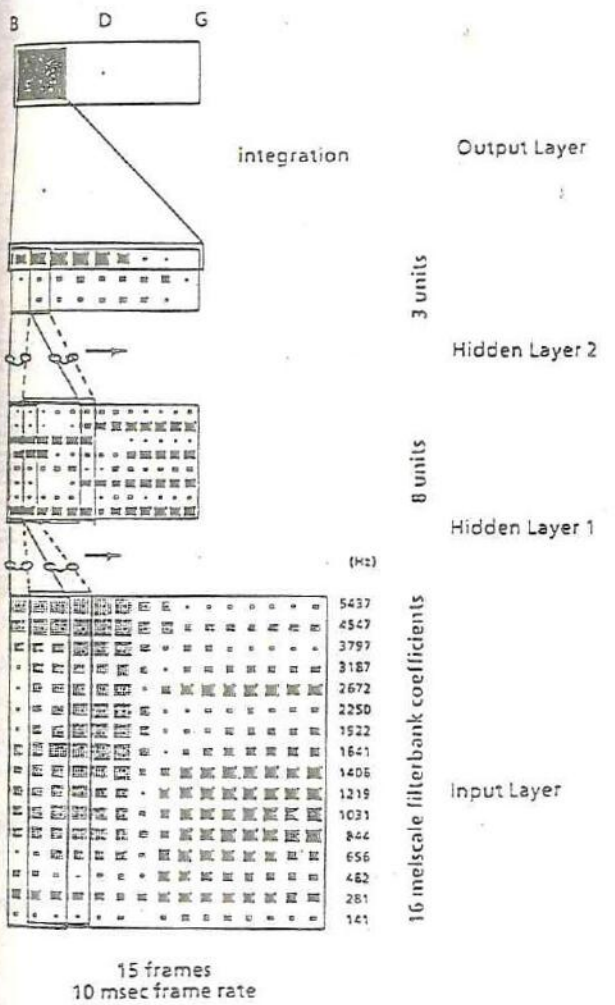

図 3-1 Time Delay Neural Network の構成

表 3-1. TDNN と HMM との音韻認識率の比較
(単語音声中の /b, d, g/の識別)

発声者	TDNN	HMM
MAU	98.8%	92.9%
MHT	99.1%	97.2%
MNM	97.5%	90.9%

3.2 全音韻の認識(ニューラルネットワークのスケールアップ)[Waibel 88]

前節では、有声破裂音 /b, d, g/ のようなカテゴリ数の少ない場合には、Back-Propagation による学習が高精度で達成できることをTDNNによる音韻認識実験で示した。このBDGタスクでのニューラルネットワークのアーキテクチャーをそのまま全音韻の識別に拡張するには、学習サンプル数の増加とともに、Back-Propagation の学習に要する時間がカテゴリ数に指数関数的に増大するので、現実には不可能となる。この節では、このような大規模ニューラルネットワークを構成するスケーリングアップの1アプローチを紹介する。

ここでは、音韻クラスターごとに学習によってえられた小規模の数個のニューラルネットワークを利用して、大規模ニューラルネットワークをBack-Propagation アルゴリズムを用いて効率よく構成することを試みる。

その準備として、有声破裂音のBDGネットワークと無声破裂音のPTKネットワークを利用して、破裂音のBDGPTKネットワークを構成することを考える。そのために、表3-2に示す次の試みを行った。

(a) BDG ネットと PTK ネットを並列に並べて、入力に対して最大の出力値をとる音韻を正解とする。60.5% の低い音韻認識率となってしまう。

(b) BDGPTKのTDNN を、Back-Propagation 学習アルゴリズムで学習させる。学習に要する時間は、BDG ネットの数倍かかったが、98.3% の高い音韻認識率が得られた。この音韻認識率が、以下のスケーリングアップの手法の目標値となる。

(c) BDGPTKのTDNNにおいて、入力と第一層の隠れ層の間の重みは、BDGネットとPTKネットからコピーして固定し、上位レベルのみBack-

表2-2 学習アルゴリズムの速さ (rated in epochs*)

		XOR Task			838 Task				BDG Task		
Sigmoid shift	Derivative add	Initial ε	Standard Algorith	ε scal. Algorith	Initial ε	Minimal ε	Standard Algorith	ε scal. Algorith	Initial ε	Standard Algorith	ε scal. Algorith
0	0	0.5	1010	70	0.1	0.01	1830	never	0.01	560	100
0	0	10	30	35	0.1	0.1	1830	130	0.02	360	80
-0.5	0	10	25	25	0.1	0.1	1525	130	0.02	315	80
-0.5	+0.1	10	215	35	0.1	0.1	640	80	0.02	190	80

表 3-2 BDG ネットからBDGPTK ネットへの
スケーリングアップの手法の比較

スケーリングアップの手法	bdg	ptk	bdgptk
BDG ネット、PTK ネット	98.3%	98.7%	
(a) BDGネットとPTK ネットの最大値			60.5%
(b) BDGPTKネット学習			98.3%
(c) BDGとPTKの下位の重みを固定			98.1%
(d) BDG,PTK,UV/Vの下位の重みを固定			98.4%
(e) Connectionist Glue			98.4%
(f) (e)のネットの全ての重みを再学習			98.6%

Propagation 学習アルゴリズムで再学習させる。少ない学習時間で、98.1% の音韻認識率が得られた。

(d) BDG ネットとPTK ネットの他に、有性/無声ネットを学習させ、(c)のネットに加えて、同様に、入力と第一層の隠れ層の間の重みは、それぞれのネットの重みをコピーして固定し、上位レベルのみBack-Propagation 学習アルゴリズムで再学習させる。98.4% の音韻認識率が得られた。

(e) BDG ネットとPTK ネットの他に、ランダムな初期値を持った隠れユニットを、図3-2のように付け加える。これを"Connectionist Glue"と呼ぶ。BDG ネットとPTK ネットの入力と第一層の隠れ層の間の重みは固定である。残りの重みは自由にして、Back-Propagation 学習アルゴリズムで再学習させる。98.4% の音韻認識率が得られた。

(f) (e)で得られたすべての重みを自由にして、Back-Propagation 学習アルゴリズムでもう一度再学習させる(Fine Tuning)。この再調整で98.6% の音韻認識率が得られた。

次に、全ての子音/(b, d, g), (p, t, k), (m, n, N), (s, sh, h, z), (ch, ts), (r, w, y)/の識別ネットワークの構成を行う。それぞれの子音クラス内で、表3-3に示すような認識率が得られている。また、これらの子音クラス間の識別ネットも学習させ、96.7% の子音クラス識別率が得られている。上記の手法(c)をさらに簡単化し、入力から第2層の隠れ層までを図3-3のように固定して、第2層の隠れ層と出力層の間の重みのみ再学習させた。これにより、95.0% の音韻認識率が達成された。さらに、すべての重みを自由にして、再学習をおこなった。これにより95.9% まで音韻認識率が向上した。この結果は、同じデータに対する Hidden Markov モデルの音韻認識率、92.7% よりも優れている。

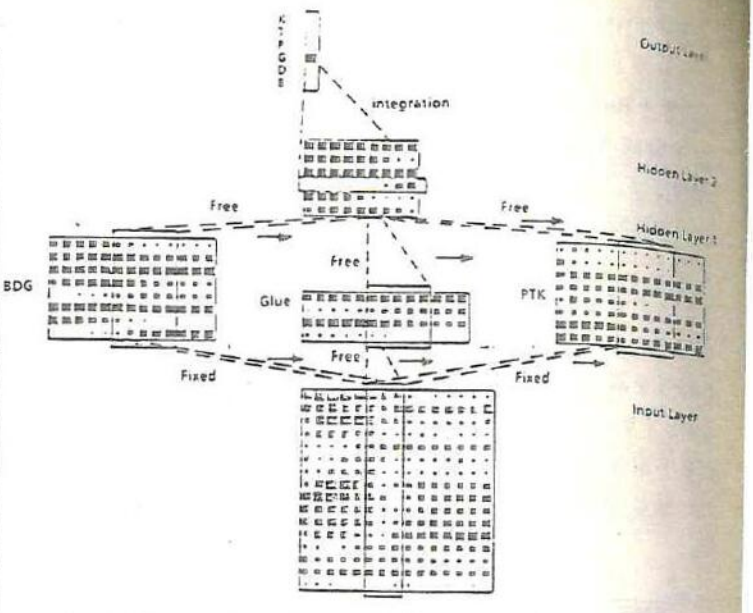

図 3-2 Connectionist Glue を持ったTime Delay Neural Network の構成

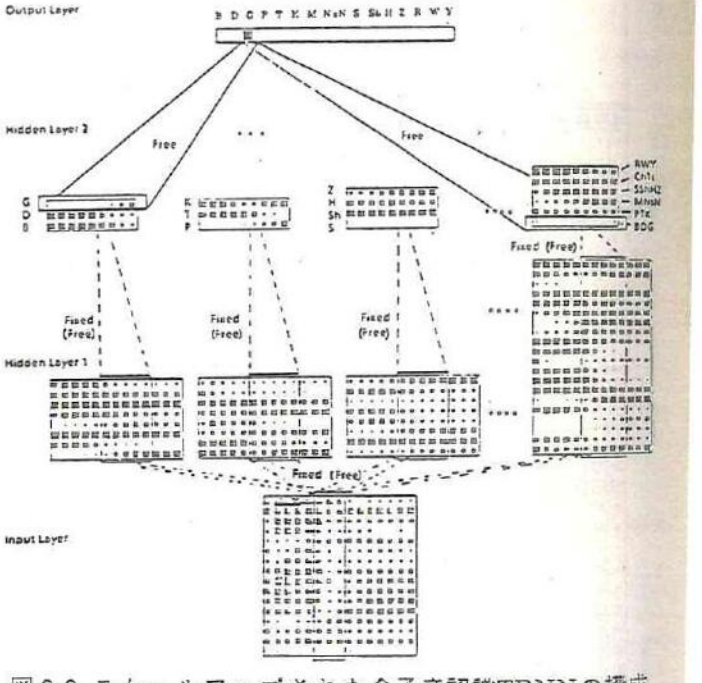

図 3-3 スケールアップされた全子音認識TDNNの構成

3.3 TDNN による音節スポッティングの検討[Sawai 88]

Back-Propagation アルゴリズムは、10カテゴリ以下程度の少数カテゴリの識別には、Lateral Inhibition の能力により、他カテゴリの出力値を考慮することができ、非常に強力である。多カテゴリの場合のLateral Inhibition を利用したインクリメンタルな学習の可能性を前節で示した。その他の日本語の単語や連続音声認識を実現することのできる有望な手法としてスポッティングが考えられる。ここでは、TDNNを音節スポッティングに適用した検討結果を紹介する。

従来、音節(Demi-syllable)のスポッティングをニューラルネットを用いて行った実験とし

"ド"、"レ"、"ミ"、"ファ"、"ソ"、"ラ"、"シ"の7節の実験[Kamm 88]があるが、カテゴリー数が限られている。日本語の約100音節のスポッティングに適用するためには、多くのカテゴリーを扱えるニューラルットを如何に構成するかが問題点となる。

我々は、日本語音声中のCV音節をスポッティングするためのニューラルネットワークとして、ある音節その音節以外とを識別するものを構成する。もし、のようなネットワークが学習でき、精度良く音節をポッティングすることが可能となれば、音節数だけネットワークを用意することにより、原理的に全て音節をスポッティングすることが可能となる。今回、日本語の単音節の一例として"BA"を取り上げ。"BA"以外の音節としては約100音節存在するが、てを使用することは学習上の効率から好ましくないで、"BA"とコンフュージョンを起こし易いと考えれる"DA","GA","PA","TA","KA"の5音節を使用す。

ここでの音節スポッティング用のTDNNは、出力層ユニットが/BA/と/non-BA/(BA以外)の2つであ。学習用のサンプルとしては、重要語5240単語中の数から"BA"を含む単語53語を抽出し、"BA"の部分15フレーム(10ms周期)を切り出した。"BA"以外の音節として、"DA","GA","PA","TA","KA"を含む単語の該当音節部分15フレームを同様に切り出した。学習サンプル数は全部で1014音節である。学習は、Back-Propagation学習アルゴリズムによって行った。未知入力音声中を、"BA"スポッティング用のTDNNを3フレームずつシフトしながらスキャンした。教師データの与え方としては、"B"と"A"の境界と、TDNNの中心とのずれが一定時間内(30msないし50ms)のときに"BA"とした。但し、境界の曖昧な部分は評価対象から除外した。未知入力音声の部分が"BA"であるか"non-BA"であるかの判定は、出力ユニットの値 $0 \leq o(BA), o(non\text{-}BA) \leq 1$ を用い、次の判定条件に従って決定した。

<判定条件>

1. $o(BA) > o(non\text{-}BA)$ なら"BA"と判定
2. $o(non\text{-}BA) > o(BA)$ なら"non-BA"と判定

表3-4と表3-5にスポッティングの実験結果を示す。評価用の単語としては、学習用の単語とは異なる"BA"を含む61単語を用いた。"BA"以外の音節は138音節ある。3フレームずつシフトした時のサンプル数は、"BA"が156個、"non-BA"が1018個である。音節サンプルの同定率は"BA"が83.3%、"non-BA"が98.7%である。また音節単位では、"BA"は95.1%の確率で同定でき、"non-BA"は99.3%の確率で抑圧できた。音節単位での誤りは語頭の/a/(ashiba)、語中の/ba/(keibatsu, shibaraku, jibaN)で生じた。

表3-4. CV スポッティング結果(音節単位)

音節名	BA	non-BA	合計
音節数	61	138	199
音節同定率	58/61 (95.1%)	137/138 (99.3%)	195/199 (98.0%)

表3-5. CV スポッティング結果(サンプル単位)

音節名	BA	non-BA	合計
サンプル数	156	1018	1174
音節サンプル同定率	130/156 (83.3%)	1005/1018 (98.7%)	1135/1174 (96.7%)

日本語の音節スポッティングの検討をTDNNを用いて行った。ある音節とそれ以外の音節を識別できるように学習したニューラルネットを用いてある音節(例:"BA")を95%識別でき、他の音節を99.3%で抑圧できることを示した。これにより、他の音節検出用ニューラルネットを用意しておけば、任意の音節のスポッティングができる可能性を示した。

4. 信号処理への適用

多層パーセプトロンの信号処理への応用は、音声認識などのカテゴリ分類への応用に比べて遥かに少ない。しかし、多層パーセプトロンが原理的に任意の連続写像を実現できること、さらに、Back-Propagation アルゴリズムという簡単で強力な学習アルゴリズムが得られていること等、を考えると、信号処理への応用もまた非常に有望なものであるように思える。

ATRでは、雑音抑圧を雑音に汚染された信号の空間から雑音のない信号の空間への写像と考え、その写像の実現に多層パーセプトロンを適用する研究を行っている[Tamura 88a]。雑音抑圧に用いた多層パーセプトロンは、図4-1に示すようなもので、線形出力関数を持つユニットで構成される入出力層を持ち、波形そのものを入出力信号とする。この四層パーセプトロンを、男性話者一名が発声した音声データとコンピュータルーム雑音を使って、Back-Propagation でトレーニングし、四層パーセプトロンが正しく雑音抑圧の写像を学習できる事を確認した。さらに、女性音声にホワイト雑音を加えた雑音重畳音声入力に対しても、男性話者音声とコンピュータルーム雑音で学習した四層パーセプト

表3-3 子音認識結果

b,d,g	p,t,k	m,n,N	s,sh,h,z	ch,ts	r,w,y	broad class	all cons	fine tuning	HMM
98.6%	98.7%	96.6%	99.3%	100%	99.9%	96.7%	95.0%	95.9%	92.7%

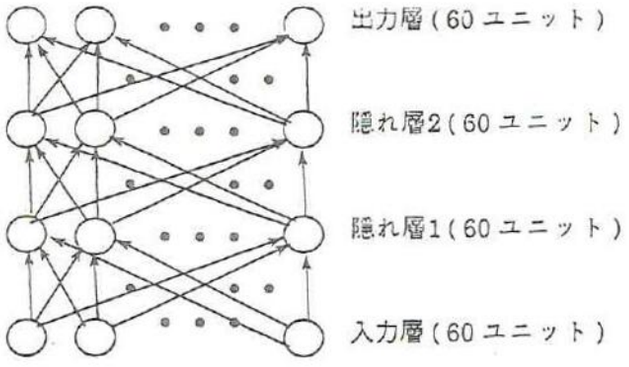

図4-1 雑音抑圧四層パーセプトロン

ロンは、学習データとほぼ同等の雑音抑圧特性を示す事を確認した。さらに、このニューラルネットワークの内部動作の解析を行った[Tamura 88b]。その結果、以下の事柄が明らかになった。

(1) 入力層から第一層の隠れ層までの変換で、線形雑音抑圧、および非線形な音声/雑音の特徴量測定が行われている。

(2) 第一層の隠れ層から第二層の隠れ層への変換で、状態空間における音声成分の領域の大きさを保ったままで、雑音成分の領域が圧縮されている。

(3) 第二層の隠れ層から出力層への変換により、音声成分の領域から雑音の抑圧された音声波形が生成される。

今後、さらにネットワークの解析を進めるとともに、解析で得られた知識をネットワークにあらかじめ入れこみ、雑音抑圧ニューラルネットワーク性能を向上させることができる。

その他、図4-2に示すような、入出力層のユニット数を同じにして、隠れ層のユニット数を入出力層のユニット数より少なくした3層パーセプトロンを情報圧縮に応用する研究もある。この3層パーセプトロンを、出力信号が入力信号と同じになるようにトレーニングすると、入力信号は絞りこまれた隠れ層を通らねばならず、隠れ層に情報圧縮された信号が現れることになる。10kHzでサンプリングされた音声データを使い、線形のユニット五十個から成る入出力層と、sigmoidを持つ二十個のユニットから成る隠れ層一層で構成される三層パーセプトロンを波形入出力で、Back-Propagationにより学習させた結果、良好な情報圧縮特性が学習外データに対しても確認された[Elman 87]。

また、多層パーセプトロンを非線形予測に適用する研究も行われている。予測すべきデータとして、人工的に作成したカオスの数列を使い、コンピュータ実験を行ったところ、Back-Propagationでトレーニングされた四層のパーセプトロンは、従来の予測手法(iterated polynonial, Widrow-hoff, and non iterated polynonial)よりも良い予測特性を示した[Lapedes 88]。

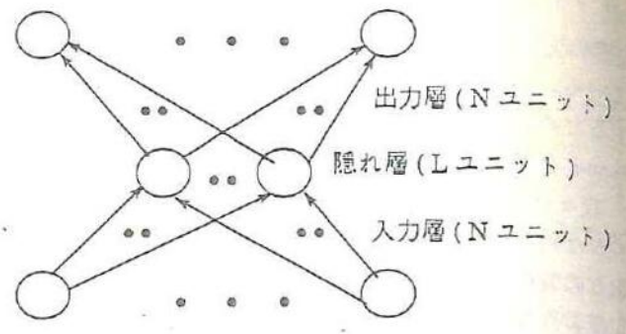

図4-2 情報圧縮三層パーセプトロン(L<N)

5. 音声合成、単語予測への適用

音声合成への適用ではNETtalk [Sejnowski 86] が有名である。これは、英文テキストから音素記号を生成するニューラルネットワークで、図5-1に示すように3層のネットワーク構造を持っている。入力層は7つのグループに分けられ、それぞれ英文テキストの連続した7つの文字が入力される。1つのグループは29個のユニットを持ち、26個のアルファベットと3個の句読点、単語の句切りに対応している。出力層は26個のユニットを持ち、26個の発音記号(正確には21個の音素記号、5個のアクセント、音節の切れ目)に対応している。隠れユニットは80個である。学習は英文テキストをニューラルネットの入力に右から左へ1文字ずつずらしながら与え、中央の入力文字に相当する発音記号を出力に教えることにより行う。1024単語の学習において、50回の学習繰り返し回数で95%の正解率を得たという。

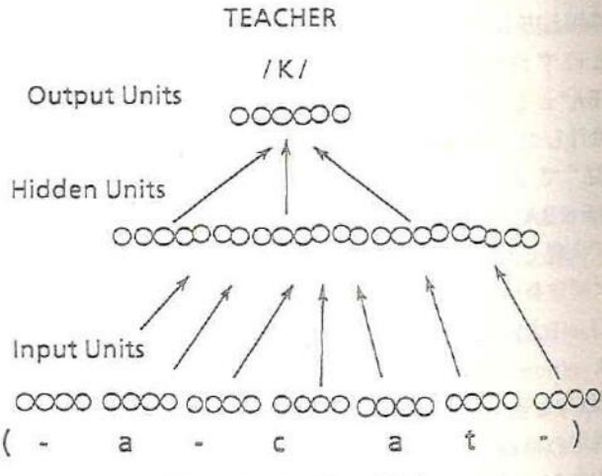

図5-1 NETtalkの構成

同じ様な記号処理の問題として、筆者らは英文テキストの単語列予測をニューラルネットワークに適用することを試みている[Nakamura 88]。目的は音声認識結果の誤りを訂正するためであるが、従来の確率モデルでは情報容量の関係でTrigram予測(3つ組予測)までが実用限度であったのに対し、ニューラルネットでは情報圧縮効果によりN-gram予測(N組予測)まで拡張可能なモデルを構成することが可能である。現在、いくつかのモデルを検討中であるが、そのうちの1つを図5-2に

示す。Bigram予測(2つ組予測)は4層のニューラルネットで、入力層、出力層それぞれ89個のユニットを持ち、英単語の品詞88個とブランク1個に相当している。隠れ層は2層あり、それぞれ16個のユニットを持ち、入力層と出力層のマイクロフィーチャを学習させる。Bigram予測モデルの学習が終われば入力層を新たに左側に追加し、下位の隠れ層に結合して、その間の結合重み係数の初期値を0にしてTrigram予測の学習をする。この方法により、Bigram予測モデルで学習した出力と同じ値から学習を始めることができる。以降、gram数を1つ増やす毎に左側に入力層を追加し、同様の手順で学習を行う。Bigram予測モデルの学習後、Trigram予測モデルで約12,500語を23回学習した段階で、オープンデータに対し、従来の確率によるTrigram予測とほぼ同じ予測的中率を得ている。フリーパラメータ(情報容量)の数は確率モデルの場合、組合せの数8^3=704,969であるのに対し、ニューラルネットワークの場合、それは結合重み係数の数4,649に対応するため、約1/151の情報圧縮効果がある。また、本タスクは例外の多い多対多の写像問題であるため、学習の困難さが予想されたが、約100語のBigram予測学習でニューラルネットワークの出力値がほぼ確率値と同じ値に収束することを確認した。

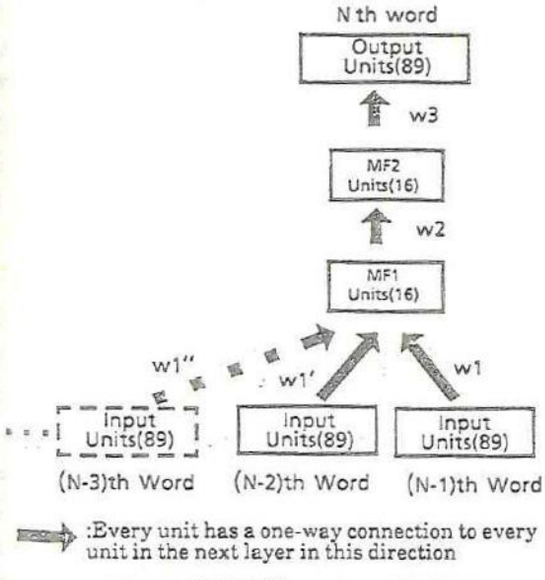

図 5-2 単語予測モデルの構成例

6. Kohonen のネットワーク

ニューラルネットワークの研究は、多層パーセプトロンにおける誤差のBack-Propagationに基づく学習アルゴリズムの提案によって一躍脚光を浴びるようになった。勿論、それまでにも、様々なアプローチによるニューラルネットワクの研究が地道に行われてきた。Kohonen のネットワークもその一つである。本章では、Kohonen のネットワークの基本的手法と音声認識への応用を概説する。Kohonen のネットワークは、ベクトルが属するクラスのベクトル空間内判別境界を、統計的に学習するベクトル量子化の一種と見ることができる。その入力ベクトルの判別関数は、量子化によって得られる参照用ベクトルと入力ベクトルとの距離に基づいて決定される。本方法と、入力ベクトルの判別関数が多数の結合係数を用いて実現される多層パーセプトロンやボルツマンマシンとの差異は、この判別関数の実現方法の差異によって特徴づけられる。しかし、いずれの方法も、求める判別関数を統計的学習法に基づいて実現するという共通点も持っている。この様に異なる学習方法相互の関係を明らかにすることは、ニューラルネットワークの判別能力や判別結果の分析を進めるために必要な有益な知見をもたらすものと考えられる。

以下では、始めに、Kohonen のネットワークの基本的アルゴリズムを、自己組織化特徴写像と学習ベクトル量子化、学習ベクトル量子化2に分けて紹介する。続いて、Kohonen らによる音声認識への応用、筆者らによる音韻認識への応用を紹介する。

6.1 アルゴリズムの概説

6.1.1 準備

まず、アルゴリズムの説明に必要な準備を行なう。ここで取り扱う標本は、各標本の性質を表現するM次元ベクトル$\mathbf{x}$として定義される。これらのベクトルは、いずれもクラスC_j ($j=1,2,3,\ldots,N$)のいずれかに属するものとする。解決すべき課題は、各標本が属するクラスを正確に判別することである。言い替えれば、ベクトル$\mathbf{x}$が属するクラスのベクトル空間内判別境界を正確に学習することである。

Kohonen のネットワークでは、この判別は複数の参照用ベクトル$\mathbf{m}_i$ ($i=1,2,3,\ldots,K$)が実現するベクトル空間の分割に基づいて行われる。この分割とは、各参照用ベクトル$\mathbf{m}_i$を最も近い参照用ベクトルとする全てのベクトル$\mathbf{x}$を含むベクトル部分空間にベクトル空間全体を分割することである。参照用ベクトル$\mathbf{m}_i$の周囲にできるベクトル部分空間は参照用ベクトル$\mathbf{m}_i$の部分空間と呼ばれる。各参照用ベクトル$\mathbf{m}_i$は、あるクラスC_jに属していることを意味するクラスラベルC_jを持つ。参照用ベクトル$\mathbf{m}_i$の部分空間に含まれるベクトル$\mathbf{x}$のクラスは参照用ベクトル$\mathbf{m}_i$が持つクラスラベルとされる。

この参照用ベクトルの学習法によって、Kohonen のネットワークは、ベクトル空間内の判別境界の求め方によって、自己組織化特徴写像(Self-Organizing Feature Maps) [Kohonen 87] と学習ベクトル量子化(Learning Vector Quantization) [Kohonen 86] とに大別される。さらに学習ベクトル量子化は、判別境界付近に存在するベクトルの取り扱い方によって、学習ベクトル量子化と学習ベクトル量子化2 [Kohonen 88a] とに分け

られる。以下、この3種の学習法について順次紹介する。

6.1.2 自己組織化特徴写像[Kohonen 87]

本方法の特徴は、ベクトル空間の判別境界を定義する参照用ベクトルと2次元格子上の節点とが結び付けられている点にある。図6-1は、2次元の格子上の節点とこれと結び付けられている(破線)参照用ベクトルとを模式的に示している。学習が行われる前は、参照用ベクトルはベクトル空間の原点付近に置かれた乱数ベクトルとされる。

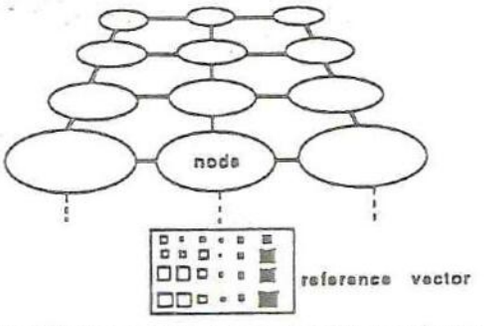

図6-1 2次元格子上の節点とこれに対応する参照用ベクトルの模式図。

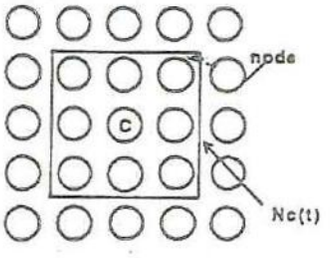

図6-2 近傍節点の模式図。節点Cを中心とする窓$N_c(t)$の内部にある節点が近傍節点とされる。

学習アルゴリズムは、式(6-1)と(6-2)とに示される。時刻t_kにおいて、学習用ベクトルの集合から任意の1個のベクトルxが選択される。このxに最も近い参照用ベクトルm_cが選択され、これに対応する節点Cとこの節点近傍の近傍節点がさらに選択される。ここで、近傍節点とは、節点Cを中心とする窓$N_c(t_k)$に含まれる節点を指す。図6-2は、窓$N_c(t_k)$の例を示している。こうして選択された近傍節点に対応する参照用ベクトルm_cは、式(6-2)に従って学習用ベクトルxに近づけられる。この時、$N_c(t_k)$に含まれない参照用ベクトルm_cは動かされない。

(最近傍参照用ベクトルの選択)

$$\| x(t_k) - m_c(t_k) \| = \min\{ \| x(t_k) - m_i(t_k) \| \} \quad (6\text{-}1)$$

(参照用ベクトルの変更)

$$m_i(t_k+1) = m_i(t_k) + \alpha(t_k)[x(t_k) - m_i(t_k)] \quad \text{for } i \in N_c \quad (6\text{-}2)$$

$$m_i(t_k+1) = m_i(t_k) \quad \text{otherwise.}$$

学習用ベクトルxが提示された時の上記の参照用ベクトルの変更は、予め準備された全ての学習用ベクトルに対して繰り返し行われる。この繰り返しにおいて、窓$N_c(t_k)$の大きさと参照用ベクトル変更の大きさを決定する重み$\alpha(t_k)$は、それぞれ単調減少する様に設定する。

以上の学習が繰り返される結果、格子上で近い位置にある節点に対応する参照用ベクトルはベクトル空間においても近い位置に位置するようになる。即ち、格子上の節点の配置は、多次元ベクトル空間における参照用ベクトルの配置を近似的に表現している。また、十分多くの学習ベクトルを用いて得られる参照用ベクトルは、各参照用ベクトルの部分空間内に在る学習用ベクトル集合の重心に近づくことが統計的に期待できる。

自己組織化特徴写像の過程では、学習ベクトルが属するクラスに関する情報は全く用いられなかった。このため、本手法はベクトル空間の統計的性質を学習する教師無し学習の一種と見なされる。得られた参照用ベクトルを用いて学習用ベクトルのクラスに関する情報を表現するためには、参照用ベクトルの部分空間に含まれる学習用ベクトルのクラスが調べられ、最も多いクラスがその参照用ベクトルのクラスとしてラベル付けされる。この処理の後、クラスラベルを付けられたこの参照用ベクトルを用いた未知ベクトルの判別が可能となる。

6.1.3 学習ベクトル量子化(LVQ)[Kohonen 86]

学習ベクトル量子化のアルゴリズムが式(6-3)と(6-4)とに示されている。式から明らかなように、本方法は自己組織化特徴写像とかなり類似している。両者の大きな違いは、参照用ベクトルに対応する2次元格子上の節点の有無と学習用ベクトルのクラスに関する情報の取り扱いの差異にある。自己組織化特徴写像における参照用ベクトルは2次元格子上の節点と対応付けられていたが、学習ベクトル量子化ではこの節点が対応付けられていない。さらに、ここでは、学習用ベクトルのクラスに基づいて参照用ベクトルの変更方法が決定される、いわゆる教師有り学習が行われる。

まず始めに、M個の参照用ベクトルが準備される。この参照用ベクトルには、自己組織化特徴写像によって得られた参照用ベクトルや、k-平均クラスタリングによるベクトル、あるいは学習用ベクトルの集合から任意に選択されたベクトル等が用いられる。これらは、いわゆる参照用ベクトルの初期値である。

次に、学習ベクトルの集合を用いて学習が行われる。ある学習用ベクトルxに対する最近傍参照用ベクトルm_cが選択される。このm_cのクラスラベルS_sがベクトルxのクラスS_rと等しい場合、m_cはxの方向に近づけられる。一方、m_cのクラスラベルS_sがベクトルxのクラスS_rと異なる場合、m_cはxから遠ざけられる。m_c以外の参照用ベクトルは動かされない。この手続きにおいて、$\alpha(t_k)$は時間とともに単調減少する。

以上の学習後、クラスラベルを持つ参照用ベクトルと未知ベクトル x との距離が計算され、x の判別が行われる。

(最近傍参照用ベクトルの選択)

$$\| x(t_k) - m_c(t_k) \| = \min\{ \| x(t_k) - m_i(t_k) \| \} \quad (6\text{-}3)$$

(参照用ベクトルの変更)

$$m_c(t_k+1) = m_c(t_k) + \alpha(t_k)[x(t_k) - m_c(t_k)] \quad \text{if } S_s = S_r$$

$$m_c(t_k+1) = m_c(t_k) - \alpha(t_k)[x(t_k) - m_c(t_k)] \quad \text{if } S_s \neq S_r \quad (6\text{-}4)$$

$$m_i(t_k+1) = m_i(t_k) \quad \text{for } i \neq c.$$

6.1.4 学習ベクトル量子化 2 (LVQ2) [Kohonen 88a]

それぞれクラスラベル C_j と C_k とを持つ参照用ベクトル m_i と m_h とがある。この2つの参照用ベクトルから同一の距離にある超平面がクラス C_j と C_k との判別境界である。この時点で、参照用ベクトルの位置が不適切なため、この判別境界は正確にベクトル x のクラスを判別できないものとする。この判別境界の回りに幅wの"窓"を設ける。図6-3は、1次元空間における m_i と m_h、及びその中間に設けられた"窓"を示している。また、図中には、m_i のクラスラベルが示すクラス C_j に属するベクトルの確率分布と m_h のクラスラベルが示すクラス C_k に属するベクトルの確率分布とも模式的に図示されている。ベイズの判別理論に基づく最適な判別境界は、図中の2つの確率分布が交差する位置にある。

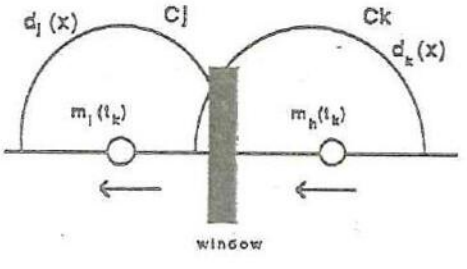

図6-3 1次元空間におけるLVQ 2 の動作原理。矢印は、参照用ベクトル $m_i(t_k)$ と $m_h(t_k)$ とが学習によって動かされる方向を示している。

LVQ2における参照用ベクトルの更新は、最終的な判別境界がLVQの場合よりもベイズの判別境界に近づくように行われる。クラス C_k に属する学習ベクトル x が与えられたとする。この時、この x に最も近い参照用ベクトルが m_i で、これに次いで近い参照用ベクトルが m_h となった。m_i のクラスラベルは C_j で、m_h のクラスラベルは C_k である。さらに、こうした条件下にある x が m_i と m_h との中間にある"窓"の中に在る場合、即ち、x がこれらの2つの参照用ベクトルの中間より m_i に近い位置に在る場合、m_i は x から遠ざけられ、m_h は x に近づけられる。式(6-5)と(6-6)とはこの手続きを示している。この参照用ベクトルの更新が繰り返されると、図6-3 中の m_i と m_h との中間にある判別境界が確率分布の交差する位置に近づいていくことが容易に予想される。

以上の学習の後、未知のベクトルの判別が行われる。

(最近傍参照用ベクトルの選択)

$$\| x(t_k) - m_c(t_k) \| = \min\{ \| x(t_k) - m_i(t_k) \| \} \quad (6\text{-}5)$$

(参照用ベクトルの変更)

$$m_i(t_k+1) = m_i(t_k) - \alpha(t_k)[x(t_k) - m_i(t_k)]$$

$$m_h(t_k+1) = m_h(t_k) + \alpha(t_k)[x(t_k) - m_h(t_k)]$$

; if C_j is the nearest class, but x belongs to $C_k \neq C_j$ where C_k is the next-to-nearest class. Furthermore x must fall into the "window".

$$m_l(t_k+1) = m_l(t_k) \text{ in all the other cases.} \quad (6\text{-}6)$$

6.2 Kohonen のネットワークを用いた音声認識

Kohonen らは、自己組織化特徴写像を用いた音声認識システムを構築した[Kohonen 88b] [Kohonen 88c]。ここでは、その概要を簡単に紹介する。音声信号は、帯域フィルターバンクに通され、9.83ms 毎のスペクトルベクトルとして表現される。また、これらのスペクトルベクトルは、属する音韻クラスのラベルが付けられている。

学習過程では、このスペクトルベクトルに対する自己組織化特徴写像が行われ、図6-4に示す節点とこれに対応する参照用ベクトルとが得られる。図6-4の各節点に付けられた音韻クラスラベルは、6.1.2で述べたラベル付けによって付けられた。複数のクラス名が併記されている節点は、この節点に対応する参照用ベクトルの部分空間に含まれる学習ベクトルの音韻クラスが、複数に分散したことを示している。言い替えれば、これらの参照用ベクトルの音韻クラス判別能力はやや劣っていることになる。特に、スペクトルベクトルの遷移状態が音韻固有の特徴を表現する破裂音等の判別の正確さを向上させるため、2個の連続したスペクトルベクトルを共に用いた参照用ベクトルも補助的に用いられている。

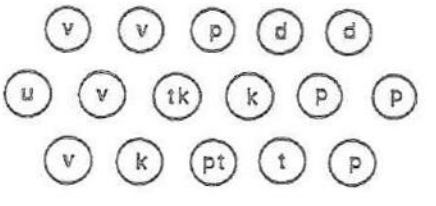

図6-4 音韻ラベルクラスのラベル付けが行われた節点の例 [Kohonen 88a]。節点 tk は、/t/ または /k/ のクラスを意味しており、この節点にクラス判別の信頼性は低い。

認識過程では、学習によって得られた参照用ベクトルと未知音声のスペクトルベクトルとの距離計算が行われた。その結果、未知音声スペクトルベクトルの系

列は、音韻クラスラベル(ここでは擬音韻と呼ばれた)の系列に変換された。自己組織化特徴写像による参照用ベクトルの学習が統計的に優れたものであっても、調音結合などの様々な変動要因を持つスペクトルベクトルから完全な音韻系列を得ることは困難であった。そこで、この音声認識システムでは、得られた擬音韻系列における誤り系列は、Dynamically Expanding Context と呼ばれる手法によって修正が行われた。

以上の手続きによって得られた音韻候補系列を用いた単語音声認識システムと連続音声認識システムとが構築されている。単語音声認識システムでは、得られた音韻候補系列と辞書中の音韻の bigram と trigram との比較がハッシュコード法によって高速に行われた。また、連続音声認識システムでは、N-gram モデルの利用が検討されている。

特に、単語音声認識システムの大部分は、マイクロプロセッサに実現されており、ほぼ実時間認識を可能としている。システムは、特定話者用であり、認識に先立って100単語程度の学習用音声の発声が必要とされた。これらの単語は、自己組織化特徴写像によって予め準備されていた参照用ベクトルを対象話者用に適応させるために用いられた。この適応、即ち、学習は、LVQによって実行された。得られた認識性能は、音韻認識率が70%から90%、1000単語を対象とした単語認識率が96%から98%であった。認識率の変動は、発声者及び対象単語間の類似性の差異等に基づくものであった。

また、これらの他に、自己組織化特徴写像によって作成された節点間に格子平面上の地形図的な距離を導入し、この距離に動的計画法を適用した単語音声認識システムも検討されている。

6.3 Kohonen のネットワークを用いたシフトインバリアントな音韻認識

以上概観したように、Kohonen のネットワークを用いた音声認識手法は、古典的な複数参照用ベクトルを用いたパタンマッチングの手法の一種である。Kohonen のネットワークの核心は、統計的な学習によってベクトル空間内のクラス間判別境界を適切に設定する点にある。そこで、用いるベクトルやベクトル間距離の選択が判別能力を決定する重要な要因となる。本節では、筆者らによって行われているKohonen のネットワークを用いた音韻認識の試みを紹介する[McDermott 88a]。本研究では、音声特徴は時間遅れを持つ複数のベクトル系列で表現されている。

図6-5は、本音韻認識システムの構成を示している。ここで、音韻は、15個の連続した16次元メルスペクトルの系列、即ち、240次元のベクトルとして定義した。認識対象音韻クラスは、/b/、/d/、/g/の3種である。しかし、システムは、240 次元のベクトルに図6-5に示すような窓をかけ、7 個の連続した16 次元のメルスペクトル(10msec毎に求められた)から成る112次元ベクトルを入力とした。この窓は、メルスペクトル系列方向に動かされ、音韻毎に9個の112次元ベクトルを抽出した。参照用ベクトルも112次元ベクトルである。各参照用ベクトルは、音韻の240次元ベクトルの情報の一部によって学習された。システムの学習は、LVQに基づいて行われ、/b/、/d/、/g/の各クラス毎にそれぞれ40、50、60個の参照用ベクトルが準備された。学習前の参照用ベクトルの初期値としては、学習標本中から任意に抽出されたものを用いた。ベクトル間の距離には、ユークリッド距離を用いた。

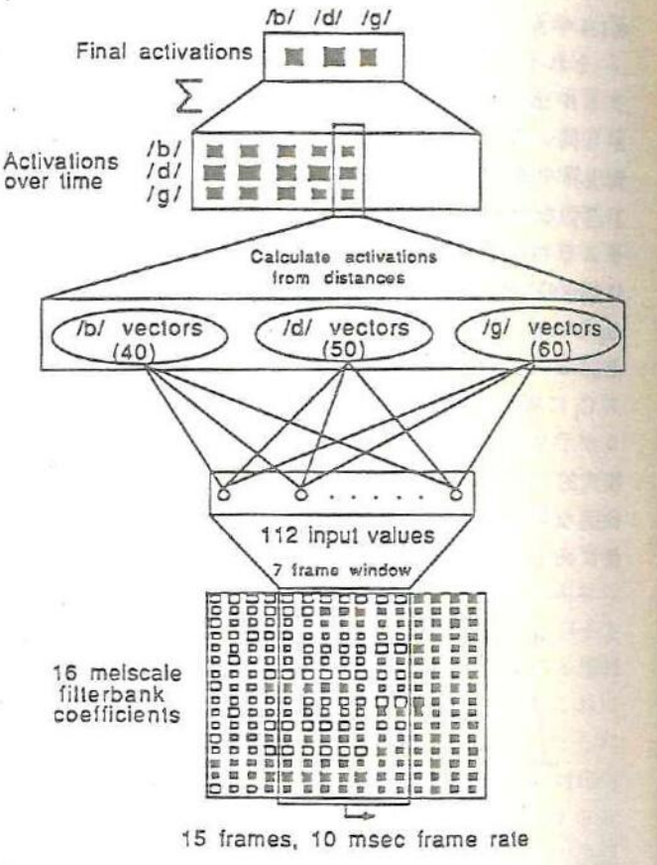

図6-5 Kohonen のネットワークを用いたシフトインバリアントな音韻認識システムの構成図。中間層の( )内の数字は、各音韻クラスごとの参照用ベクトルの数を示している。

認識処理(図6-5 上半分)は以下の手続きによって行われた。入力された240 次元の未知音韻ベクトルに対し、学習時と同様に窓がかけられた。各窓位置(t)において、窓がかけられた112 次元ベクトルと全ての参照用ベクトルとの距離が計算され、各音韻クラス毎に最も小さい距離の値が蓄えられた。/b/、/d/、/g/ のクラスの最小距離をそれぞれ $d_b(t)$、$d_d(t)$、$d_g(t)$ とし、各時点tにおける各音韻らしさ、即ち、各音韻クラス c(c=b, d, g) の活性度 A(c,t) は次式によって求められた。

$$A(c,t)=1-d_c(t)/(d_b(t)+d_d(t)+d_g(t)) \qquad (6\text{-}7)$$

この活性度は次の様な値を示すものである。入力ベクトルがいずれの音韻クラスの最近傍参照用ベクトルと

も等距離にある場合、Aは約2/3となる。また、入力ベクトルがある一つの音韻クラスcの最近傍参照用ベクトルに近く、かつ他の音韻クラスの最近傍参照用ベクトルから離れている場合、Aは1に近づく。一方、入力ベクトルがある一つの音韻クラスcの最近傍参照用ベクトルから遠く、かつ他の音韻クラスの参照用ベクトルに近い場合、Aは0に近付く。Aは0以上である。窓位置(t)の移動に伴って、1つの未知音韻ベクトル全体から得られる各音韻クラスcに対する活性度は次式で求められた。最も大きな活性度が得られた音韻クラスがシステムの出力とされた。

$$A(c)=A(c,0)+A(c,1)+\cdots+A(c,9) \qquad (6\text{-}8)$$

表6-1は、3名の男性話者に関する音韻認識率を示している。この実験に用いられた音声標本は、日本語重要語約5200単語中の様々な音韻環境下にある/b/、/d/、/g/であり、話者は男性3名であった。学習標本集合と評価用標本集合との内訳は、表6-2に示されている。多層パーセプトロンを用いた認識結果(表3-1)と同程度の高い認識率を示している。

表6-1 Kohonenのネットワークを用いたシフトインバリアントな音韻認識システムの音韻認識率

speaker	tokens	# errors	% correct
MAU	b d g	4 4 3	98.3
MHT	b d g	5 0 3	98.7
MNM	b d g	8 1 5	97.8

(結果は評価用標本に対するものである。誤りは個数で、正解は百分率で表されている。)

表6-2 学習用標本と評価用標本の内訳

speaker class	MAU	MHT	MNM	FSU
b	218 (227)	207 (208)	216 (216)	204 (204)
d	203 (179)	185 (170)	196 (177)	184 (170)
g	260 (252)	259 (254)	264 (254)	255 (246)

(Mで始まる話者が男性、Fで始まる話者が女性である。話者FSUは図6-6に関する実験で用いられている。各欄、学習用標本数(評価用標本数)が示されている。音韻位置の抽出が極めて困難な音声標本が実験に用いられなかったため、標本数にやや変動がある。これらの音声は、各話者とも約5200語の重要語単語音声データベースから選択された。)

以上説明した手法では、240次元の音韻ベクトルが時間遅れを伴って分割されて用いられた。こうした方法は、240次元のベクトルが抽出される音声波形の区間の厳密な選択を不用なものとする。このため、本手法

一方、以上の様な音韻ベクトルの分割を行わない場合の認識率も確かめられている[Yokota 88]。図6-6は、240次元ベクトルをそのままLVQにおいて扱った音韻認識システムの結果を示している(話者MAU)。ここでは活性度は用いられずに、240次元のユークリッド距離のみが用いられた。ここでは、ほぼ同数の参照用ベクトルを用いているにもかかわらず上記のシステムより認識率が低い。また、全学習用ベクトル(681個)を参照用ベクトルとして評価用ベクトルの認識実験を行った結果、認識率は98.6%であった。これが、240次元ベクトルをそのまま用いた場合にこの学習標本集合から得られる認識率の上限と考えられる。150個の参照用ベクトルを用いたシフトインバリアントなシステムの認識率がこの上限にかなり接近した高い認識率を実現している原因としては、音韻ベクトルを分割したことと活性度を用いたこととが考えられる。詳しい分析を行う必要はあるが、こうした結果は、入力ベクトルの取扱方が重要であることを示しているものと考えられる。

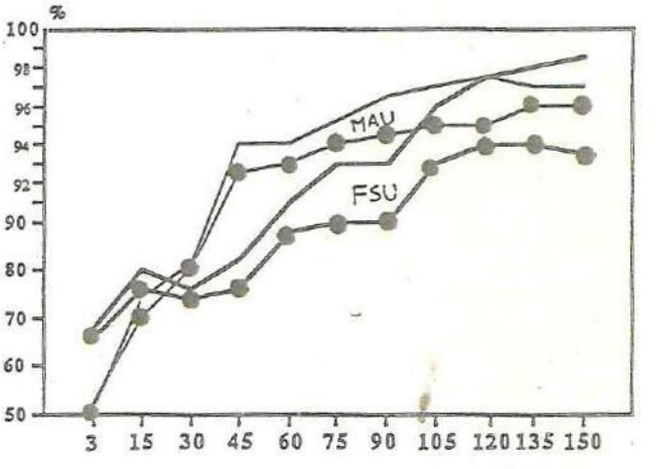

図6-6 240次元音韻ベクトルをそのまま用いた場合の音韻認識率。細線が話者MAUの結果を、太線が話者FSUの結果を示している。折れ線は学習用標本を用いた結果を、黒丸は評価用標本を用いた結果を示している。

さらに、上記のシフトインバリアントな音韻認識システムは、/b/、/d/、/g/の3音韻クラスと/p/、/t/、/k/の3音韻クラスに関する学習を行った後、全体の6音韻クラスを対象にした再学習によって、6音韻クラスに対する99.1%の認識率を実現できる様に拡張された。そこでは、LVQの代わりにLVQ2が用いられた[McDermott 88b]。

7. むすび

以上の例は、カテゴリ分類の枠組みでの問題、空間の写像、規則の発見の問題でのニューラルネットワークの適用例を中心に解説した。このように、多層パーセプトロンをBack-Propagationアルゴリズムで学習させるという枠組のアプローチは、適用範囲を急速に広げつつある。さらに、KohonenのLearning Vector

び、実際の音韻認識実験でのBack-Propagation アルゴリズムとの比較を行った。このようなアプローチ以外にも、Hopfield のモデル[Hopfield 84] によるBoltzmann Machine [Hinton 84] があり、研究が進められている。

(謝辞)

この解説を書くにあたって有益なコメントを頂いた当ATR自動翻訳電話研究所社長の樽松博士、主任研究員の川端博士、当ATR視聴覚機構研究所社長の淀川博士に感謝します。

(参考文献)

[Baum] E.B.Baum,"On the Capabilities of Multilayer Perceptrons", preprint

[Elman 87] J.L.Elman and D.Zipser,"Learning the Hidden Structure of Speech", Tech.Rep. ICS Report 8071, University of Calfornia, (1987-11)

[Franzini 87] M.A.Franzini,"Speech Recognition with Back-Propagation",Proceedings, 9th Annual Conference of IEEE Engineering in Medecine and Biology Society,(1987)

[Fukushima 75] K.Fukushima, "Cognitron : A Self-Organizing Multilayered Neural Network", Biol.Cybernetics, 20,121-136, (1975)

[Fukushima 77] 福島,三宅,"連想記憶能力を持つ自己組織機械", 電子通信学会論文誌(D), 60-D, pp143-150, (1977)

[Funahashi 88] 船橋賢一,"ニューラル・ネットワークによる連続写像の近似的実現について", 電子情報通信学会技報,MBE88-9,(1988-04)

[Haffner 88] P.Haffner, A.Waibel, K.Shikano,"Fast Back-Propagation Learning Methods for Neural Networks in Speech", 音響学会講演論文集,(1988-10)

[Hinton 84] G.E.Hinton, T.J.Sejnowski, D.H.Ackley, "Boltzmann Machine : Constraint Satisfaction Networks that Learn", Tech.Rep.CMU-CS-84-119, (1984)

[Hopfield 84] J.J.Hopfield, "Neurons with Graded Response Have Cllective Computation Properties like Those of Two-State Neurons", Proceedings of National Academy of Science, USA, 81, pp3088-3092, (1984)

[Huang 87a] W.Y.Huang, R.P.Lippmann, "Comparisons between Conventional and Neural Network Classifiers", IEEE First International Conference on Neural Networks, San Diego, (1987-06)

[Huang 87b] W.Y.Huang, R.P.Lippmann, "Neural Networks and Traditional Classifiers", Conference on Neural Information Processing Systems, (1987-11)

[Huang 88] W. Huang, R. Lippmann and B. Gold, "A Neural Net Approach to Speech Recognition", IEEE, ICASSP88, S3.1, pp.99-102, 1988.

[Kawabata 84] 川端,牧野,城戸,"音韻論的特徴量を用いた4層形子音認識モデル", 電子通信学会論文誌(D),Vol.J67D,No.1, pp141-148, (1984-01)

[Kamm 88] C.Kamm, T.Landauer, S.Singhal, "Using an Adaptive Networks to Recognize Demisyllables in Continuous Speech", ASA Meeting, X7, (1988-05)

[Kohonen 77] T.Kohonen, "Associative Memory : A System Theoretical Approach", NewYork, Springer, (1977)

[Kohonen 86] T.Kohonen, "Learning Vector Quantization for Pattern Recognition", Helsinki University of Technology, Report TKK-F-A601, (1986-10)

[Kohonen 87] T.Kohonen, "Self-Organization and Associative Memory (2nd Edition)", pp.119-157, Springer, (1987)

[Kohonen 88a] T.Kohonen, "The Neural Phonetic Typewriter", IEEE Computer, pp 11-22, (1988-03)

[Kohonen 88b] T.Kohonen, G.Barna, R.Chrisley; "Statistical Pattern Recognition with Neural Networks: Benchmarking Studies", IEEE, Proc. of ICNN, pp.I-61 - I-68, San Diego, (1988-07)

[Kohonen 88c] T.Kohonen, K.Torkkola, M.Shozakai, J.Kangas and O.Venta; "Phonetic Typewriter for Finnish and Japanese", IEEE, ICASSP88, S13.10, pp.607-610, (1988)

[Lang 83] S.Lang, "Real Analysis", p3,Urysohn's lemma,

[Lapedes 88] A.Lapedes,Robert Farber, "How Neural Networks Work", preprint, (1988-03)

[Leung 88] H.Leung, V.Zue,"Some Phonetic Recognition Experiments Using Artificial Neural Networks", ICASSP'88, pp422-425, (1988-04)

[Lippmann 87] R.P.Lippmann,"An Introduction to Computing with Neural Networks", IEEE ASSP Magazine, pp 4-22, (1987-04)

[Lubensky 88] D.Lubensky,"Learning Spectral -Temporal Dependencies Using Connectionist Networks", ICASSP'88, pp 418-421, (1988-04)

[McDermott 88a] E.McDermott and S.Katagiri, "Phoneme Recognition Using Kohonen Networks", ATR Workshop on Neural Networks and Parallel Distributed Processing, pp. 13-14, (1988-07)

[McDermott 88b] E.マクダーモット,片桐,"Kohonenのネットワークに基づいたシフトインバリアントな音韻認識",音響学会,秋季研究発表会,(1988-10)

[Minsky 69] M.Minsky, S.Papert, "Perceptrons", Cambridge, MA, MIT Press, (1969)

[Nakamura 88] 中村,鹿野,"英文テキストデータからのニューラルネットによる単語列予測モデルの検討", 信学技報,SP88-, (1988-06)

[Nielsen 87] R.Hecht-Nielsen, "Kolmogorov's Mapping Neural Network Existence Theorem", ICNN '87 Proceedings, pp Ⅲ-11 - Ⅲ-14 (1987)

[Rumelhart 86a] D.E.Rumelhart, J.L.McClelland, "Parallel Distributed Processing : Explorations in the Micro Structure of Cognition", Vol. I, II, (1986)

[Rumelhart 86b] D.E.Rumelhart, G.E.Hinton, R.J.Williams, "Learning Internal Representations by Back-Propagation Errors", Nature, 323, pp 533-536, (1986-10)

[Sawai 88] 沢井,ワイベル,鹿野,"時間遅れ神経回路網による音節スポッティングの検討",音響学会講演論文集,(1988-10)

[Sejnowski 86] T.J.Sejnowski, C.R.Rosenberg, "NETtalk : A Parallel Network that Learns to Read Aloud", Technical Rep. JHU / EECS - 86 / 01, John Hopkins Univ. (1986-06)

[Sprecher 65] D.A.Sprecher, "On the Structure of Continuous Functions of Several Variables", Trans. Amer. Math. Soc., 115, pp 340-355, (1965-03)

[Tamura 88a] S.Tamura, A.Waibel, "Noise Reduction Using Connectionist Models", ICASSP'88, pp 553-556, (1988-04)

[Tamura 88b] 田村,"波形入出力による雑音抑圧ニューラルネットワークの解析",音響学会講演論文集,(1988-10)

[Waibel 87] A.Waibel, T.Hanazawa, G.Hinton, K.Shikano, K.Lang, "Phoneme Recognition Using Time-Delay Neural Networks", ATR Technical Rep. TR-I-0006, (1987-10), あるいは, "Phoneme Recognition: Neural Networks vs. Hidden Markov Models", ICASSP'88, pp 107-110, (1988-03)

[Waibel 88] A.Waibel, H.Sawai, K.Shikano, "Phoneme Recognition by Modular Construction of Time-Delay Neural Networks",音響学会講演論文集,(1988-10)

[Yokota 88] 横田,片桐,マクダーモット,"LVQに基づくスペクトル空間の音響的性質",音響学会,秋季研究発表会,(1988-10)