

ニューラルネットワークの音声情報処理への応用*

鹿野清宏・中村雅己・田村震一・アレックス ワイベル**

(ATR 自動翻訳電話研究所)

1. ま え が き

ニューラルネットワークをパターン認識に応用しようという試みは 1950, 60 年代に一時、盛んに行われたが、研究が進むにつれて原理的な限界が指摘され¹⁹⁾、多層モデルの効率的な学習方法が見つからなかったなどの問題のため、やがて下火になった。しかし、一方でニューラルネットワークモデルの基礎的検討は地道に進められ、代表的なものとして 1970 年代に自己形成可能なコグニトロン^{3), 4)}、連想記憶モデル¹²⁾、1980 年代に統計力学的なアイデアを導入したボルツマンマシン⁶⁾などが提案され、それぞれ独自の発展を見せた。しかし、いずれもアーキテクチャが複雑で専門家以外の者が実際に計算機上で応用モデルを実現し、試行するのが困難であった。

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^{22), 23)}。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 hidden 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。特に、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共に、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、音声の規則合成において、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²⁴⁾。音声認識の分野においても、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^{9), 16), 18)}。このようにニューラルネットワークは、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出てきたため非常に注目されており、特に、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^{8), 17)}。

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 ニューラルネットワーク (TDNN)²⁷⁾ 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²⁶⁾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²⁴⁾と、単語カテゴリの予測問題²⁰⁾を取り上げ、簡単に紹介する。

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、出力層及び 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-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。

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 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 $net_{pj} = \sum_i w_{ji} \cdot o_{pi}$ をとり、更に入出力関数 $f(x)$ 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入れたときに、上記のような計算をすべてのユニットで行い

* Neural network applications to speech processing.

** Kiyohiro Shikano, Masami Nakamura, Shin'ichi Tamura and Alex Waibel (ATR Interpreting Telephony Research Laboratories, Osaka, 540)

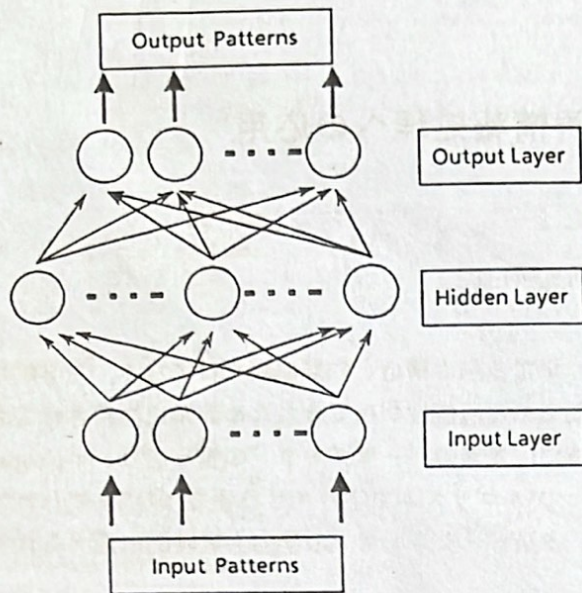


図-1 多層ニューラルネットワーク

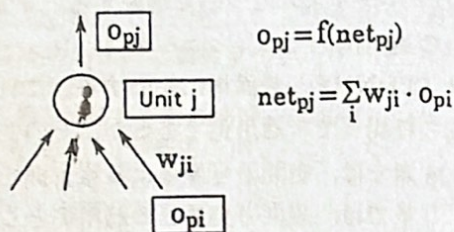


図-2 ニューラルネットワークのユニット

(ただし、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評価関数を決める。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。

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

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 をすべ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 $E = \sum_p E_p$ 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 ごとの結合重み係数 w_{ji} の更新量 Δw_{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。

$$\Delta w_{ji} \propto -\frac{\partial E_p}{\partial w_{ji}} \quad (2)$$

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

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial net_{pj}} \cdot \frac{\partial net_{pj}}{\partial w_{ji}} \quad (3)$$

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

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

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

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} \quad (5)$$

と置くことにより、

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

よって、(2)式の学習規則は

$$\Delta w_{ji} = \eta \cdot \delta_{pj} \cdot o_{pi} \quad (7)$$

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。

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

ここで、 $o_{pj} = f(net_{pj})$ であるから右辺の右側は

$$\frac{\partial o_{pj}}{\partial net_{pj}} = f'(net_{pj}) \quad (9)$$

である。(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、

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

であるから、

$$\frac{\partial E_p}{\partial o_{pj}} = -(t_{pj} - o_{pj}) \quad (10)$$

となり、直接 δ_{pj} が次式のように求まる。

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

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。

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

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

このように、 Δw_{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(11)、(13)から明らかなように微分可能な関数が必要とされる。文献 22) では次のような非線形単調増加のシグモイド関数を用いるのが

よいとしている。

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

すなわち、ユニット j の出力は次式のようになる。

$$f(\text{net}_{pj}) = o_{pj} = \frac{1}{1+e^{-\text{net}_{pj}}} \quad (15)$$

ここで入力総和 net_{pj} は $\text{net}_{pj} = \sum_i w_{ji} \cdot o_{pi} + \theta_j$ としてバイアス成分 θ_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見なして学習する。

$f(\text{net}_{pj})$ の導関数を求めると

$$f'(\text{net}_{pj}) = \frac{\partial o_{pj}}{\partial \text{net}_{pj}} = o_{pj} \cdot (1 - o_{pj}) \quad (16)$$

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。

$$\Delta_p w_{ji} = \eta \cdot \delta_{pj} \cdot o_{pi} \quad (17)$$

ただし、ユニット j が出力ユニットの場合は

$$\delta_{pj} = o_{pj} \cdot (1 - o_{pj}) \cdot (t_{pj} - o_{pj}) \quad (18)$$

であり、ユニット j が隠れユニットの場合は

$$\delta_{pj} = o_{pj} \cdot (1 - o_{pj}) \cdot \sum_k \delta_{pk} w_{kj} \quad (19)$$

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、一つの入力パターン提示ごとに実行するか、次式のように全入力パターン提示後に

$$\Delta w_{ji} = \eta \cdot \sum_p (\delta_{pj} \cdot o_{pi}) \quad (20)$$

として実行するか、二つの方法がある。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、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。

2.2 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(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。文献 22) では、前回の更新

幅をモーメント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。

$$\Delta w_{ji(n+1)} = \eta \cdot \delta_{pj} \cdot o_{pi} + \alpha \cdot \Delta w_{ji(n)} \quad (21)$$

ここで、 α はモーメント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、更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ごとに、次式によりエラー E_p 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。

$$E_p(w_{ji(k)} + \Delta w_{ji(k)}(\eta(k), \alpha(k))) \quad (22)$$

$$= \text{Min}_{l, m} E_p(w_{ji(k)} + \Delta w_{ji(k)}(\eta_l, \alpha_m)) \quad (23)$$

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 E_p が最も小さくなる η, α を選択するようにした (dynamically control training parameter, DCP 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-1 に示す。

パラメータ一定に比較して、DCP 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 (Case 1, Case 2)、パラメータ一定の場合、 $\eta=0.4$ のときはエラーの振動が生じ易く (Case 3, Case 4)、 $\eta=0.1$ であれば収束が遅くなる (Case 2, Case 5)。 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 (Case 3)、 $\alpha=0$ の場合、学習が進んでも収束速度が加速しないことが分かった (Case 5)。

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従って減少していった。すなわち、DCP 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。DCP 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

表-1 学習効率化策 (DCP 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 200	178	more than 200

* 前回の η の 1/2, 1, 2 倍の値から選択

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回の繰り返し計算ごとにすべてのパラメータの組み合わ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 多層パーセプトロンの能力

入出力層の非線形性は本質的でないので、隠れユニットのみが sigmoid のような飽和特性関数を非線形出力関数として持つ多層パーセプトロンを考える。

多層パーセプトロンを、入力空間から出力空間への変換、すなわち、関数と見なした場合、隣接する層間の結合のみを許した4層パーセプトロンのネットワーク構造は、Sprecher の一般化 Kolmogorov の定理に示される関数の表現と同じであることが指摘されている^{17), 21)}。この定理は、任意の多変数連続関数が一変数の連続関数の和で表現できるという驚くべき事実を述べているもので、実数体を R 、閉区間 $[0, 1]$ を I 、写像 F による集合 A のイメージを $FA(= \{F(x): x \in A\})$ 、リプシッツ条件、 $\forall x, y |F(x) - F(y)| \leq c|x - y|^\alpha$ (c : 定数, $0 < \alpha \leq 1$) に属する関数の集合を $Lip[\alpha]$ とすると、以下のように記述できる。

[Sprecher の一般化 Kolmogorov の定理]²⁵⁾

任意の整数 $N(\geq 2)$ について、実数値をとる連続関数 Ψ で、単調増加、 $\Psi I = I$ 、 $\Psi \in Lip[\ln(2)/\ln(2N+2)]$ なる、 N に依存する関数 Ψ が存在し、 Ψ は以下の性質を有する。

任意の実数 $\delta > 0$ に対し、 $0 < \varepsilon \leq \delta$ なる有理数 ε が存在して、任意の $n(2 \leq n \leq N)$ 変数の連続実数値関数 $f(x): I^n \rightarrow R$ は次の表現を持つ。

$$f(x) = \sum_{q=0}^{2n} \Gamma \left[\sum_{p=1}^n \lambda_p \Psi(x_p + \varepsilon q) + q \right] \quad (24)$$

λ は f と独立に定まる定数、 Γ は f に依存して決まる連続実数値関数。

$n=2$ として、この式をネットワークの言葉で表現すると図-3 のような、入出力層のユニットに非線形出力関数がなく、隠れ層1のユニットは出力関数として、 Ψ を持ち、隠れ層2のユニットは、 Γ を持つ、4層ネットワークの形になる。Sprecher の一般化 Kolmogorov の定理から、このネットワークはわずか15個の隠れユニットで、任意の連続関数を実現できる。任意の二つの背反な閉集合 A, B を見分けるような関数、すなわち、 A の上で1をとり、 B で0をとるような特性関数は、連続関数で実現できる¹⁴⁾から、このネットワークは任意のカテゴリ領域も実現できることが分かる。残念ながら、このネットワークは隠れ層のユニットに Ψ, Γ という特

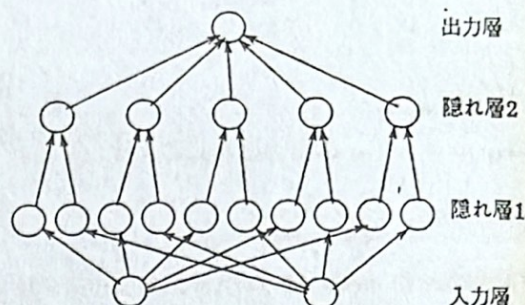


図-3 Sprecher の一般化 Kolmogorov の定理によるネットワーク

殊なユニット出力関数を要するため、この定理から直接に、4層パーセプトロンの任意の連続写像の実現可能性を言うことはできない。しかし、連続関数は、定義域に含まれるコンパクト集合上において、sigmoid のような飽和特性関数の和で、幾らでもよく近似できるという補題を上述の定理に適用すると、隠れユニットの数に制限を設けなければ、4層パーセプトロンは、隠れユニットをどんどん増やすことで、いくらでも精密に任意の連続写像を近似できる、という理論的な結果が得られる⁵⁾。これは、連続関数、 Ψ, Γ を sigmoid のような飽和関数の和で近似して、図-3 のネットワークを、写像としての機能をそこなわずに、4層パーセプトロンに変形するというアイデアである。

以上述べたように、任意の連続関数や任意のカテゴリ領域を多層パーセプトロンで実現する際、理論的には、4層あれば十分である。3層パーセプトロンの能力については、コンピュータ実験により、複雑なカテゴリ領域が形成できることが示されている⁹⁾。

また、入力層のユニットは非線形出力関数を持たず、隠れユニット及び、出力層のユニットがステップ関数を出力関数として持つ多層パーセプトロンの能力についても研究されており、有限個の隠れユニットを持つ3層パーセプトロンで、非常に特殊な位置関係にある入力データを除いて、有限個の入力データを任意の複数カテゴリに分類できることが構成的に証明されている¹⁾。

3. 音声認識への適用

ATR では、音声現象を考慮して、工学的立場から図-4 に示すような time delay neural network (TDNN) のアーキテクチャを考案し、音韻認識の $/b, d, g/$ の識別で、表-2 に示すように HMM より大幅に高い認識率を達成した²⁷⁾。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¹⁰⁾、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いる。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

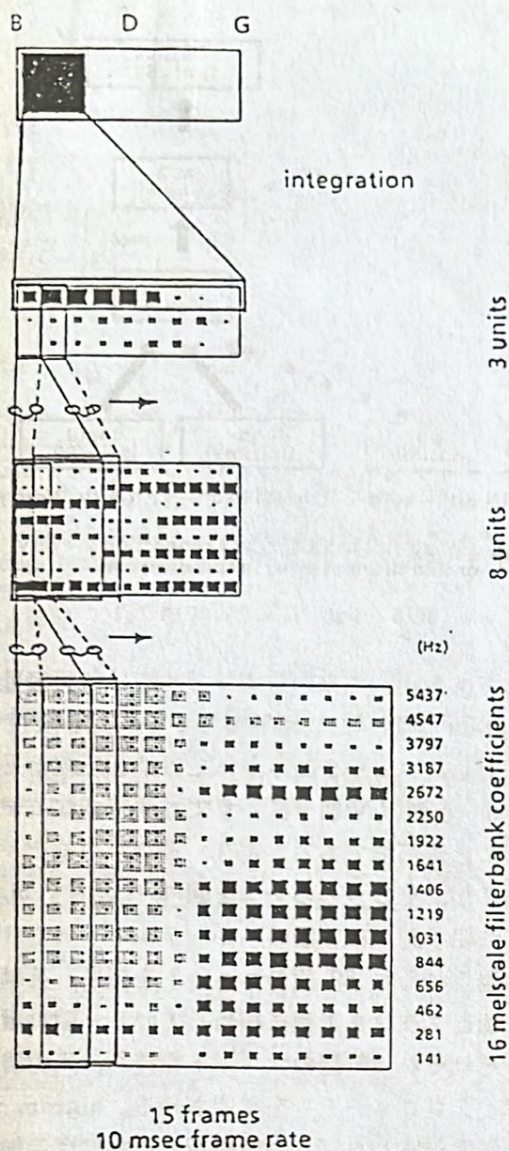


図-4 Time delay neural network の構成

表-2 TDNN と HMM との音韻認識率の比較
(単語音声での /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%

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 hidden layer 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ホ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。

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。しかしながら、多カテゴリの場合の lateral inhibition を利用した学習の実現性、ワードスポッティング¹¹⁾ の場合の verification の能力等については、なお研究する必要がある。

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遙かに少ない。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、更に、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²⁶⁾。雑音抑圧に用いた多層パーセプトロンは、図-5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4層パーセプトロンを、男性話者1名が発声した音声データとコンピューター雑音を使って、back-propagation でトレーニングし、4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ことを確認した。更に、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ー雑音で学習した4層パーセプトロンは、学習データとほぼ同等の雑音抑圧特性を示すことを確認した。

その他、図-6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

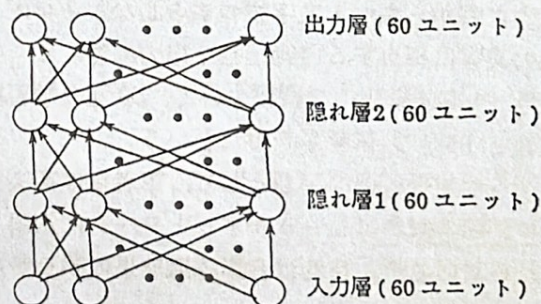


図-5 雑音抑圧4層パーセプトロン

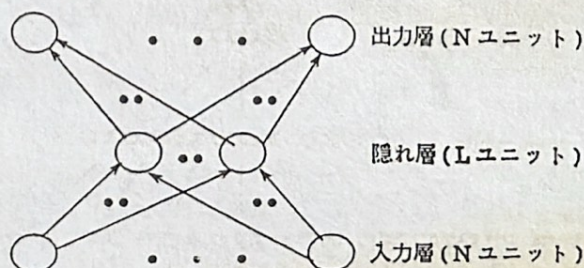


図-6 情報圧縮3層パーセプトロン ($L < N$)

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。12 kHz 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50個から成る入出力層と、sigmoid を持つ20個のユニットから成る隠れ層1層で構成される3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²⁾。

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。予測すべき実験データとして人工的に作成したカオスの数列を使って、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4層のパーセプトロンが、従来の予測手法 (iterated polynomial, Widrow-Hoff, and non-iterated polynomial) よりもよい予測特性を示した¹⁵⁾。

5. 音声合成、単語予測への適用

音声合成への適用では NET talk²⁴⁾ 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-7 に示すように3層のネットワーク構造を持っている。入力層は七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七つの文字が入力される。一つのグループは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。1,024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。

同じよう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²⁰⁾。目的は音声認識結果の誤りを訂正

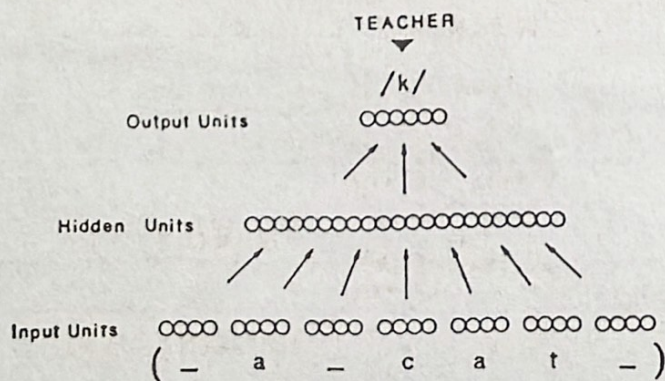


図-7 NETtalk の構成²⁴⁾

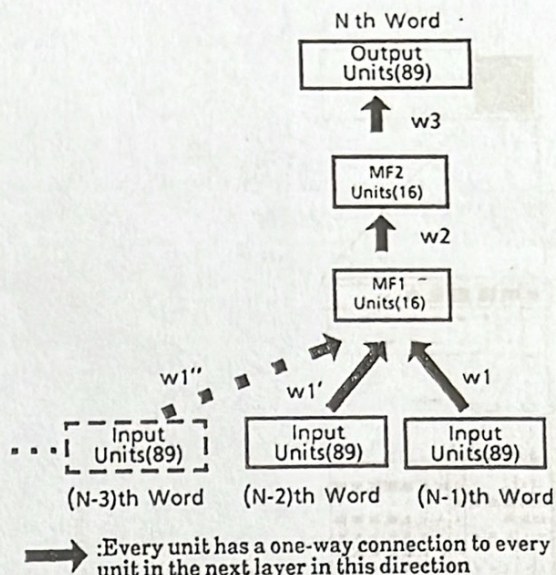


図-8 単語予測モデルの構成例

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 (三つ組予測) 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る 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、幾つかのモデルを検討中であるが、そのうちの一つを図-8 に示す。bigram 予測 (二つ組予測) 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 数を一増やすごと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 (512文章) を23回学習した段階で、未知文章に対し、従来の確率による trigram 予測とはほぼ同じ予測的中率を得ている。フリーパラメータの数は確率モデルの場合、組み合わせの数 $8^3 = 704,969$ 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。

6. 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応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。このようなアプローチ以外にも、Hopfield のモ

デル¹⁾による Boltzmann machine¹⁷⁾ や phonotactic mapping¹³⁾ 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いただいた当 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, 及び, ATR 視聴覚機構研究所の主任研究員の船橋氏に感謝します。

文 献

- 1) E. B. Baum, "On the capabilities of multi-layer perceptrons," preprint (1987).
- 2) J. L. Elman and D. Zipser, "Learning the hidden structure of speech," Tech. Rep. ICS Report 8071, Univ. California (1987.11).
- 3) K. Fukushima, "Cognitron: A self-organizing multi-layered neural network," Biol. Cybern. 20, 121-136 (1975).
- 4) 福島邦彦, 三宅 誠, "連想記憶能力を持つ自己組織機械," 信学論 (D) 60-D, 143-150 (1977).
- 5) 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," 信学技報 MBE 88-9 (1988.04).
- 6) G. E. Hinton, T. J. Sejnowski and D. H. Ackley, "Boltzmann Machine: Constraint satisfaction networks that learn," Tech. Rep. CMU-CS-84-119 (1984).
- 7) J. J. Hopfield, "Neurons with graded response have collective computation properties like those of two-state neurons," Proc. Natl. Acad. Sci. U.S.A. 81, 3088-3092 (1984).
- 8) W. Y. Huang and R. P. Lippmann, "Comparisons between conventional and neural net classifiers," IEEE 1st Int. Conf. Neural Network, San Diego (1987.6).
- 9) W. Y. Huang and R. P. Lippmann, "Neural net and traditional classifiers," Conf. Neural Information Processing Systems (1987.11).
- 10) 川端 豪, 牧野正三, 城戸健一, "音韻論的特徴量を用いた4層形子音認識モデル," 信学論 (D) J 67 D, 141-148 (1984).
- 11) C. Kamm, T. Landauer and S. Singhal, "Using an adaptive networks to recognize demisyllables in continuous speech," ASA Meeting, X7 (1988.5).
- 12) T. Kohonen, *Associative Memory: A System Theoretical Approach* (Springer, New York, 1977).
- 13) T. Kohonen, "The neural phonetic typewriter," IEEE Comput., 11-22 (1988).
- 14) S. Lang, *Real Analysis* (Addison-Wesley, Massachusetts, 1983), p. 3, Urysohn's lemma.
- 15) A. Lapedes and Robert Farber, "How neural nets work," preprint (1988.3).
- 16) H. Leung and V. Zue, "Some phonetic recognition experiments using artificial neural nets," ICASSP 88, 422-425 (1988).
- 17) R. P. Lippmann, "An introduction to computing with neural nets," IEEE ASSP Mag. 4(2), 4-22 (1987.4).
- 18) D. Lubensky, "Learning spectral-temporal dependencies using connectionist networks," ICASSP 88, 418-421 (1988).
- 19) M. Minsky and S. Papert, *Perceptrons* (MIT Press, Cambridge, 1969).
- 20) 中村雅己, 鹿野清宏, "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 SP 88-26 (1988).
- 21) R. Hecht-Nielsen, "Kolmogorov's mapping neural network existence theorem," ICNN '87 Proc., III-11-III-14 (1987).
- 22) D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Micro Structure of Cognition*, Vol. I, II (MIT Press, Cambridge, 1986).
- 23) D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by back-propagation errors," Nature 323, 533-536 (1986).
- 24) T. J. Sejnowski and C. R. Rosenberg, "NETtalk: A parallel network that learns to read aloud," Tech. Rep. JHU/EECS-86/01, Johns Hopkins Univ. (1986).
- 25) D. A. Sprecher, "On the structure of continuous functions of several variables," Trans. Am. Math. Soc. 115, 340-355 (1965).
- 26) S. Tamura and A. Waibel, "Noise reduction using connectionist models," ICASSP 88, 553-556 (1988).
- 27) A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang, "Phoneme recognition using time-delay neural networks," ATR Tech. Rep. TR-I-0006 (1987.10), あるいは, "Phoneme recognition: Neural networks vs. Hidden Markov Models," ICASSP 88, 107-110 (1988).