国際研究論叢　19（1）：73～85，2005

階層型ニューラルネットワークにおける学習アルゴリズムの改良

安　高　真一郎[＊1]　岡　本　容　典[＊2]

Improvement of Learning Algorithms in Multi-layer Neural Network

Shinichiro Ataka[＊1]　Yousuke Okamoto[＊2]

Abstract

The number of hidden layer neurons greatly influences the ability of the network, however the appropriate value cannot be known in advance. This research considers how to decide the optimum number of hidden layer neurons and proposes a method for increasing the number of neurons, if necessary. In addition two methods are compared and the results improved.

キーワード

ニューラルネットワーク、中間層、誤差逆伝搬法

1　はじめに[1][2][4]

従来のコンピュータでは、あらかじめ処理手順がプログラムとして与えられ、それに応じた処理を行う。それに対し、ニューラルネットワークは、人間の脳の情報処理機構を模しており、学習というプロセスを持っている。学習データを与えれば、それに応じてパターン認識等の処理を行うことができる。

現在における、最も一般的な学習法は誤差逆伝搬法である。これは学習データと呼ばれる、入力と望ましい出力の組を例示することによって、ニューラルネットワークの結合加重を決定する学習法である。提唱されて以来、多くの分野に応用されてきたが、幾つかの問題を抱えている。その中の一つが、ネットワーク構築時に適切な値を知ることができないため、経験的に設定される数値が存在することである。

本研究では、中間層ニューロンに着目し、誤差逆伝搬法の問題解決を図る。階層型ニューラルネットワークにおいて、中間層ニューロン数が少なすぎては学習が収束せず、また多すぎても汎化能力の低下につながると言われている。そこで、一つの中間層ニューロンから学習を始め、必要な数までニューロンを増やす手法を提案する。また、十分大きな数

＊１あたか　しんいちろう：大阪国際大学経営情報学部博士（後期）課程
＊２おかもと　ようすけ：大阪国際大学経営情報学部助教授〈2005.６.20受理〉

の中間層ニューロンを持つネットワークで学習を行い、その後不要なニューロンを削る逐次削減法[3]との比較を行った。

2　ニューラルネットワーク[1][2][4]

2.1　ニューロンモデル

ニューロンとは、神経細胞のことであり、ニューラルネットワークを構成する1素子である。実際のニューロンの振る舞いは極めて複雑であり、その動作を単純な式で表すことはできない。各神経細胞は他の細胞から入力信号を受け、その入力信号の和がある値を超えた場合に他の細胞へ出力信号を送り出すという働きを持っている。ニューラルネットワークに使われる人工の素子は、生体のニューロンを厳密に模倣したものではなく、ニューロンモデルと呼ばれる、その特定の機能を抽出し、単純化した工学的モデルである。

このモデルにおける出力値の計算手順としては、他のニューロン、または外部からの入力 $\{x_1,x_2,\cdots,x_n\}$ に結合加重 $\{w_1,w_2,\cdots,w_n\}$ を掛け、加え合わせることにより、細胞内電位Sを得る。結合加重 $\{w_1,w_2,\cdots,w_n\}$ は、シナプス結線重みに該当する。細胞膜電位がしきい値 θを越えるとき、出力 y を1とする。また、細胞膜電位がしきい値以下の場合は、0を出力する。これらを数式化すると以下のようになる。

$$s=\sum_{n=1}^{N} w_n x_n \qquad \text{（式2-1）}$$

$$y=\begin{cases}1, & s \geq \theta \\ 0, & s \leq \theta\end{cases} \qquad \text{（式2-2）}$$

ただし、実際の神経細胞は、出力を0と1で単純に表せられるものではない。そこで、シグモイド関数（式2-3）を用い、出力をアナログ的な数値に直す。

$$sigmoid(s)=\frac{1}{1+e^{-s}} \qquad \text{（式2-3）}$$

これにより出力（インパルス頻度）y は、連続した値を取る。つまり細胞膜電位が閾値に到達しない中間的段階においても、ある程度の影響が他のニューロンに及ぶのである。アナログモデルにおける出力値の計算手順は下記の通りである。

$$s=\sum_{n=1}^{N} w_n x_n \qquad \text{（式2-4）}$$

$$y=sigmoid(s-\theta) \qquad \text{（式2-5）}$$

2.2　ニューラルネットワーク

ニューラルネットワークとは、第2-1節で述べたニューロンがシナプスによって結合され、一つのネットワークとなったものである。人間の脳の構造（神経回路網）を模して作った情報処理機構で、人間の得意とするような、パターン認識や、連想記憶などの処理を

効率良く行うことができる。

２.３　階層型ニューラルネットワーク

階層型ニューラルネットワークはいくつかの層からなり、各層は適当な数のニューロンからなる。また、接続されている中間層ニューロン数によって汎化能力が変化する。

入力された値は、入力層から出力層へと一方向に流れる。入力層以外の各層のニューロンは、前の層のニューロンの出力に、結合加重を掛けたものの総和を入力値として受け取り、それにシグモイド関数fを通したものを出力する。

すなわち、i_i^kを第k層の第 i ニューロンの入力の総和とし、$w_j^{k-1}{}_i^k$を第$k-1$層の第jニューロンから第k層の第iニューロンへの結合加重、θ_i^kを第k層第iニューロンのしきい値とすると、出力o_i^kは次式で表される。

$$i_i^k = \sum_j w_j^{k-1}{}_i^k \, o_j^{k-1} - \theta_i^k \qquad \text{（式2-6）}$$

$$o_i^k = f(i_i^k) \qquad \text{（式2-7）}$$

３　階層型ニューラルネットワークの学習法

３.１　バックプロパゲーション法

バックプロパゲーション法（以下BP法）は、1986年にRumelhartらによって提唱された階層型ニューラルネットワークの学習法の一つである。

多くの学習データ、入力と望ましい出力の組を例示することによって、ニューラルネットワークの結合加重を決定する。その手順は、ある入力に対しての望ましい出力を、ネットワークから得られる実際の出力と照らし合わせ、その誤差 r を基に各結合荷重 w を出力層から入力層へ変化させていく。言い換えれば、ニューラルネットワークの結合加重を、ある入力がなされた時に、決められた出力を行うように調整を行う。比較的単純なアルゴリズムでありながら、認識性能に優れており、提唱されて以来、多くの分野に応用されている階層型ネットワークの最も一般的な学習法である。

m層のネットワークを考え、第k層の第iニューロンの入力の総和をi_i^k、出力をo_i^kとし、$w_i^{k-1}{}_j^k$を第$k-1$層の第iニューロンから第k層の第jニューロンへの結合加重とし、各ニューロンの入出力関係を与える関数をfとすると、これらの変数の関係は

$$i_i^k = \sum_j w_j^{k-1}{}_i^k o_j^{k-1} \qquad \text{（式3-1）}$$

$$o_i^k = f(i_i^k) \qquad \text{（式3-2）}$$

となり、損失関数rを誤差の二乗にとると、ある学習データの組（x,y）が与えられたときのrは、

$$r = \sum_{j} \left(o_j^m(w,x) - y_j \right)^2 \qquad \text{(式3-3)}$$

となる。ここで、$o_j^m(w,x)$は結合加重w、入力データ x が与えられたときの出力値を表す。wの修正量を求めるには、このrのwについてのgradientを計算すればよい。結合加重の修正量$\Delta w_{ij}^{k-1\,k}$は、

$$\Delta w_{ij}^{k-1\,k} = -\varepsilon d_j^k o_i^{k-1} \qquad \text{(式3-4)}$$

$$d_j^m = 2(o_j^m - y_j) f'(i_j^m) \qquad \text{(式3-5)}$$

$$d_j^k = \left(\sum_{l} w_{jl}^{k\,k+1} d_l^{k+1} \right) f'(i_j^k) \qquad \text{(式3-6)}$$

という式を満たす。ただしεは、１回の修正の大きさを決めるパラメータで、小さな正の定数をとる。

学習中は、ネットワークに学習パターンを通し、その都度結合加重の修正を繰り返していく。全ての入力値に対し誤差関数rが指定した値を下回った時点で終了する。

３.２　BP法の問題点

提唱されて以来、さまざまな分野への応用が進められてきたBP法であるが、問題も数多く指摘されている。代表的なものとして、初期結合加重、中間層ニューロン数、学習時間及び計算量が挙げられる。

ある学習データに対し最短の時間、及び最小の回数で学習を収束させるような初期結合加重の与え方は分かっていないため、初期結合加重は乱数により設定されるのが一般的である。また、中間層ニューロン数は、経験的に設定するのが現状であり、事前に必要な中間層ニューロン数を知ることはできない。さらには、中間層ニューロン数が少なすぎる場合、学習が収束せず、逆に多すぎる場合は汎化能力の低下につながってしまう。複雑なパターンを学習させるには、ネットワークの規模も大きなものとなり、学習にかかる時間や計算量も大きなものとなってしまう。

これらの問題の克服が、近年のニューラルネットワークの主な研究テーマとなっている。

４　逐次削減法 [3]

４.１　逐次削減法

逐次削減法は、押野らによって提案された中間層ニューロン数の最適化手法である。その手法は、まず十分大きな中間層ニューロン数で学習する。その後、２種類の評価基準を用いて不要な中間層ニューロンを選定、及び削減するというものである。

① 分散基準

まず、どの入力パターンに対しても同じような出力しかしないようなニューロンを分散

基準を用いて選定する。学習パターンpに対する第 i 中間層ニューロンの出力値をy_{ip}、第i中間層ニューロンの出力平均を $\bar{y}_i$としたとき、分散基準s_i^2を以下のように定式化する。

$$s_i^2 = \frac{\sum_p (y_{ip} - \bar{y}_i)^2}{\sum_i \sum_p (y_{ip} - \bar{y}_i)^2} \qquad \text{（式4-1）}$$

② 相関基準

次に、相関基準を用い、全入力パターンに対するニューロンの出力値の相関が強いもののどちらか一方を選定する。

学習パターンpに対する第i,j中間層ニューロンの出力値をそれぞれ, y_{ip},y_{ip} としたとき、相関基準 R_{ij} を以下のように定式化する。

$$R_{ij} = 1 - \left| \frac{\sum_p (y_{ip} - \bar{y}_i)(y_{jp} - \bar{y}_j)}{\sqrt{\sum_p (y_{ip} - \bar{y}_i)(y_{jp} - \bar{y}_j)^2}} \right| \qquad \text{（式4-2）}$$

相関の強い二つのニューロンi,jのうち、どちらのニューロンを削減するかについては、分散基準値を参考に、$s_i^2 < s_j^2$ ならばニューロンiを削減候補とする。

4.2 不要ニューロンの削減

まず、分散基準によって求められた値 s_i^2 を昇順に並べた集合を新たにθ_hとしたとき、ある閾値Tにおいて次式のような関係にある H 個のニューロンを削減する。

$$\max_H \left\{ \sum_{h=1}^{H} \theta_h \right\} < T \qquad \text{（式4-3）}$$

この閾値で削減ニューロンが存在しない場合は、θ_1 すなわち一つのニューロンのみが削減される。

次に相関基準によって選定されたニューロンを削減する。以後、相関基準による選定を繰り返していく。ただし、削減プロセスの各段階で未学習パターンに対する認識率を求め、これが特定の値を下回った場合は、その削減処理を行う前の状態にネットワークを戻して終了させる。

4.3 ニューロンの合成

分散基準により削減候補となったニューロンは、出力層に対し、一定の出力値しか供給しないバイアス的なニューロンである。

削減候補ニューロンを第iニューロンとし、その全学習パターンに対する出力平均を$\bar{y}_i$とする。ここで、出力層ニューロンへの添え字をkとする。そのとき、次式によって削除ニューロンiは、バイアスニューロンに合成される。

$$w_{0k}(t+1) = w_{0k}(t) + \bar{y}_i w_{ik}(t) \quad k = 1,\cdots,K \tag{式4-4}$$

ただし、w_{0k}はバイアスニューロンと出力層第kニューロンの結合加重を表す。

強い相関をもつ中間層ニューロンをi,jとし、分散基準が$S_i^2 < S_j^2$となるニューロン i を削減候補とする。ニューロンi,jの相関係数をρ_{ij}としたとき、次式によって削減ニューロンiはニューロンjおよびバイアスニューロンに合成される。

$\rho_{ij} > 0$ のとき

$$w_{ik}(t+1) = w_{jk}(t) + w_{ik}(t) \quad k = 1,\cdots,K \tag{式4-5}$$

$\rho_{ij} < 0$ のとき

$$\left\{\begin{array}{l} w_{ik}(t+1) = w_{jk}(t) - w_{ik}(t) \quad k = 1,\cdots,K \\ w_{0k}(t+1) = w_{0k}(t) + w_{ik}(t) \quad k = 1,\cdots,K \end{array}\right\} \tag{式4-6}$$

5　今回提案する手法とその検証

5.1　単純増加法

今回提案する増加法は、一つの中間層ニューロンから学習をはじめ、必要な数までニューロンを増加させていく。より具体的には、学習中において、誤差 r の変動がある一定値以下になった場合に、新たな中間層ニューロンを一つ追加する。そして、追加が行われた段階で、全ての結合加重は新たに乱数で設定され、完成したネットワークで再度学習を行う。

以下に、処理の流れを示す。n は中間層ニューロン数を表し、誤差 r が x を下回った時点で、学習を終了させる。r_{t-1} は学習中における前の回の r の値、r_t は現在の回の r の値を表しており、この値が y を下回った場合、ニューロンの追加処理に移る。また、initializeは全ての結合加重を再度乱数にて初期化することを意味している。以後、この方法を単純増加法と呼ぶものとする。

```
learning:
        flag=1
        while(flag=1)
        [
          ・中間層ニューロンn個で、入力に対する出力を求める
          ・実際の出力と望ましい出力を比べ、誤差 r を求める
          if(r ≤ x)
          [
               flag=0
          ]
          if(|r_t − r_{t−1}|≤ y and flag=1)
          [
             go to addition
          ]
          ・誤差逆伝搬法を行う
        ]
addition:
        n=n+1
        initialize
go to learning
```

図5-1　処理の流れ

５.２　逐次削減法と単純増加法の検証

両手法を比較するために、5×5マスのアルファベット文字認識（図5-2）による実験を行った。

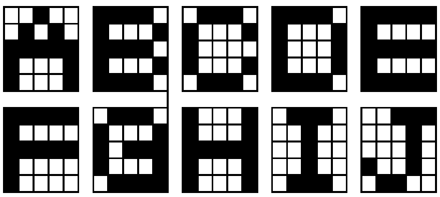


図5-2　文字データ

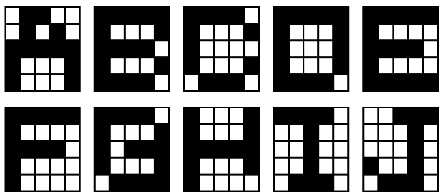


図5-3　未学習データ

５.２.１　逐次削減法による実験

学習終了後、１回目の削減処理は分散基準にて、２回目以降は相関基準によって一つずつ削減を行っていく。その後、未学習データ（図5-3）に対する認識率が、押野らの論文で用いられていた基準89％を下回った場合は、ネットワークをその一つ前の状態に戻して終了させる。

今回の実験では、入力、出力層のニューロン数25のネットワークを用いて、$r \leq 0.25$ で

学習を終了させた。さらにTの値を0.2、0.02、0.002と３段階に変化させ、また各T値においての学習データ数を１（Aのみ）から10（A～J）に変化させながら、学習データ数、T、認識率の３つが、どのように関係しており、またどのような影響を及ぼし合っているか実験を行った。次に、初期の中間層ニューロン数が、分散、相関両基準の削減プロセスに影響を与えている可能性が考えられるため、初期の中間層ニューロン数も15、30、60と変化させ、実験を行った。表5-1は中間層ニューロン数の初期値が15個、$T=0.002$ での実験結果である。表内の数値は、学習データの個数、学習終了時の認識率、分散基準による削減が終了した時点の認識率、相関基準による削減が終了した時点での認識率、最終的に残っている中間層ニューロンの個数を表している。また、各認識率の右側、カッコ内の数値はその削減基準によって削減された中間層ニューロンの個数である。表内の空欄の部分は、その削減基準を適用したものの、認識率が89%を下回ってしまい、ニューロンの削減が行われなかったことを示している。

5.2.2　単純増加法による実験

実験には、入力、出力層のニューロン数25、初期の中間層ニューロン数は１のネットワークを用い、$r \leq 0.25$で学習が終了させた。新たなニューロンを追加する条件を、学習 $t-1$回目と t 回目の誤差 r の変動が0.00001以下（式5-1）になった場合とし、学習データ数を、先の実験と同じように１（Aのみ）から10（A～J）まで変更させてどのような結果が現れるか実験を行った。

表5-2は、学習データ数、学習が完了した時点でネットワークに存在している中間層ニューロン数、そのネットワークの認識率、学習時間を表している。

$$|r_t - r_{t-1}| \leq 0.00001 \qquad \text{（式5-1）}$$

表5-1　逐次削減法による実験

学習データ数（個）	学習終了時（%）	分散基準終了時（%）	相関基準終了時（%）	最終ニューロン数（個）
1	100.0	100.0（4）		1
2	76.0			15
3	100.0	97.3（1）		14
4	100.0			15
5	100.0	100（0）		15
6	98.7	98.7（0）		15
7	98.9	98.9（0）		15
8	98.5	98.5（0）		15
9	98.7	98.7（0）		15
10	98.8	98.8（0）	99.6（4）	11

表5-2　単純増加法による実験結果

学習データ数(個)	中間層ニューロン数(個)	認識率(%)	時間(秒)
1	1	100.0	0.0
2	6	90.0	0.3
3	3	100.0	2.1
4	5	100.0	3.5
5	4	99.2	4.7
6	3	98.0	159.0
7	4	98.3	74.5
8	4	98.5	385.4
9	4	96.8	193.3
10	5	97.6	8.0

逐次削減法の実験結果は一部を抜粋したものであるが、他の実験結果でも同様に片方のみの削減基準しか適用されない場合が多く見られた。押野らは、片方のみの削減基準によって削減が行われたとしても、不必要なニューロンは未だ存在しており、両削減基準の併用が望ましいと述べている。また、削減候補が選ばれた後、押野らの論文では閾値Tによって削減ニューロンを決定していた。しかし、このTや中間層ニューロン数の初期値は経験的に設定する必要がある。したがって、これらの値によって、削減対象の選定、完成するネットワークの構造が大きく変わる可能性もある。他にも、学習が完了するまでの計算量が大きすぎる、という問題点も見受けられた。

それに対し、増加法では逐次削減法よりも少ない数のニューロン数を実現可能で安定した認識率を持つことや、少ない数のニューロンで対応可能なデータに対しての有益さ等が確認できた。しかし、多くの数のニューロンを必要とするデータに対しては、長い学習時間を必要とするという問題を残した。

５.３　単純増加法と改良型増加法

第5-2節で述べたように増加法は、コンパクトなネットワーク形成を得意とする。しかし、多くの中間層ニューロンを必要とするネットワークを形成させようとした場合、時間的な問題を有している。

そこで、この欠点を克服するため、また新たなアルゴリズムの提案を目指すために改良を加えることにした。さらに増加法とこの新たな手法を、15×15マスのアルファベット文字認識に用い比較を行うこととした。改良を行った部分は、以下の２点である。

①　ニューロンの追加が行われた際の、初期結合加重の設定方法
②　ニューロン追加後の学習アルゴリズム

①は、これまでの増加法では、ニューロン追加時に全ての結合加重値を設定し直すよう

にしていた。これを今回は、追加されたニューロンのみに行うものとする。また、それまでに存在していたニューロンの結合加重値は、そのまま引き継がせる。次に②において、ニューロンの追加後、新たなネットワークが完成し、再度学習を開始することになる。その場合、学習による結合加重の修正は、追加されたニューロンの結合加重値のみ行うものとする。それ以外のものについては、追加直前の値のまま固定した状態である。以後、これまでの増加法を単純増加法、改良を加えた手法を改良型増加法と呼ぶ。

まず、誤差 r の変動をグラフにより比較し、その後、学習が完了したネットワークの未学習データに対する認識率についても調べた。

実験には、入力、出力層のニューロン数225、初期の中間層ニューロン数は１のネットワークを用い、$r \leq 0.25$で学習が終了するようにしている。新たなニューロンを追加する条件を、誤差 r の変動が0.00001以下になった場合とした。また、誤差 r の変動を見る実験においては、図5-4に示す学習データを３個（A,B,C）、認識率を比較する実験では、学習データ数を１個(Aのみ)から５個(A〜E)まで変更させて考察を行った。

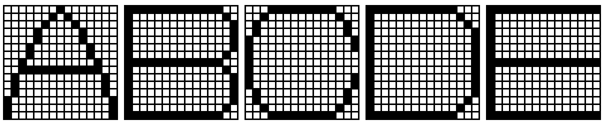


図5-4　学習データ

図5-5、5-6に両手法の学習中における中間層ニューロンの増加と誤差の変動を表す。単純増加法では、ニューロンが追加されるごとに結合加重値を振り直し再度学習を行うため、追加が行われる度に誤差が大きな値へと戻ってしまう。また、学習中に誤差が急激に増大する場合が見られ、このことが学習時間の増大に繋がると考えられる。

それに対し改良型増加法では、必要となった中間層ニューロン数が一つ多いものの、安定した誤差の減り方を見せている。

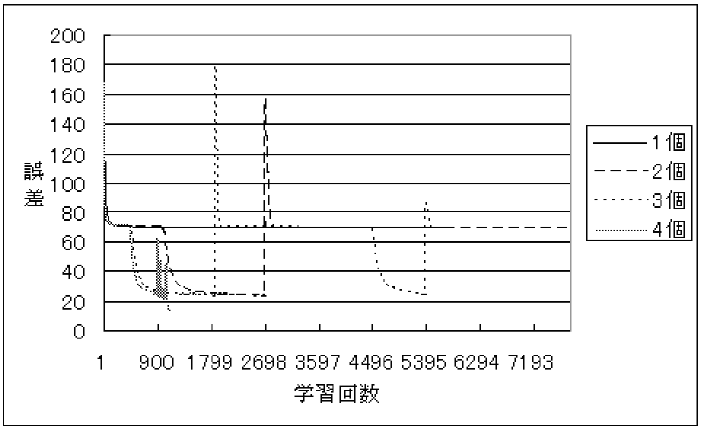


図5-5　単純増加法の学習中における誤差の変動

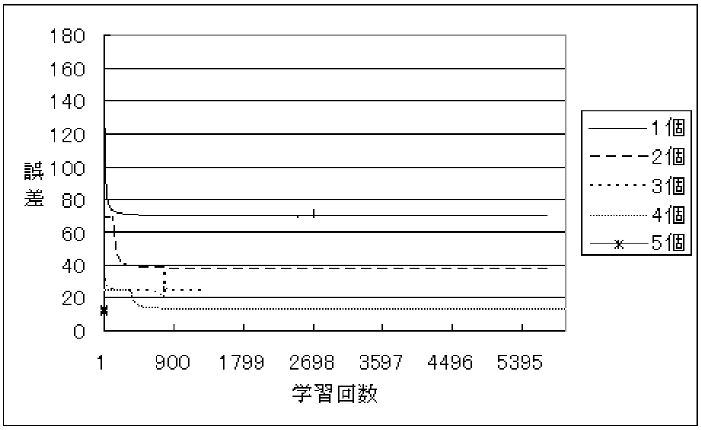


図5-6　改良型増加法の学習中における誤差の変動

次に、学習時間と学習に必要となった中間層ニューロン数、未学習データに対する認識率の比較を行う。認識率は学習が完了したネットワークに３通りの未学習データ（図5-6）を通して計測した。この実験により両者の汎化能力を検証する。

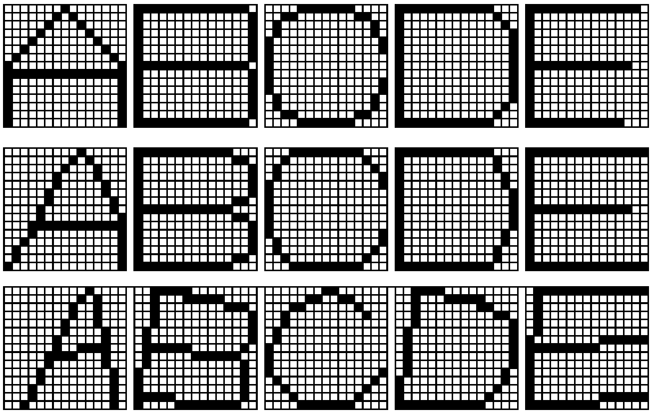
(a)
(b)
(c)


図5-7　未学習データ

表5-3、5-4は、両手法に、図5-1の学習データを１～５個に変化させながら学習させた結果と、図5-7の各未学習データを通した時の認識率を表している。表中の中間層ニューロン数という項目は、学習に必要となった中間層ニューロン数であり、認識率a～認識率cは図5-7の(a)～(c)に対応した認識率を表している。

表5-3　単純増加法による実験結果

学習データ数(個)	中間層ニューロン数(個)	学習時間(秒)	認識率 a(%)	認識率 b(%)	認識率 c(%)
1	2	0.0	100.0	100.0	100.0
2	3	5.2	100.0	80.2	80.2
3	4	12.1	100.0	91.3	94.5
4	12	33.3	94.2	93.4	94.3
5	12	43.8	92.6	94.2	93.9

表5-4　改良型増加法による実験結果

学習データ数(個)	中間層ニューロン数(個)	学習時間(秒)	認識率 a(%)	認識率 b(%)	認識率 c(%)
1	2	0.0	100.0	100.0	100.0
2	2	2.6	100.0	99.6	80.6
3	5	11.2	100.0	91.3	90.4
4	6	14.5	93.8	91.2	86.3
5	8	47.5	91.6	93.6	93.4

これらの結果より、改良型増加法を用いた方が、さらに中間層ニューロン数と学習時間を減らすことが可能であると確認できた。ニューロンの増加を行った時点で、それまでの学習で得た結合加重値は初期化せずに継続して用いた点、また追加されたニューロンのみで残りの学習を行うという点が有効であったと考えられる。

6　おわりに

誤差逆伝搬法の問題点として、経験や乱数により設定される値が存在することが挙げられる。初期結合加重は、ある学習データに対し最短の時間、及び最小の回数で学習を収束させるような値が分からないため、乱数により設定されるのが一般的である。同様に、中間層ニューロン数も事前に必要な中間層ニューロン数を知ることはできず、経験的に設定する現状である。

本研究ではその中の中間層ニューロン数に着目し、より少ない数の中間層ニューロン数で安定した認識率を導くアルゴリズムとして単純増加法を提案した。計算機実験の結果、この提案手法は逐次削減法に比べ、少ない数のニューロンで安定した認識率を出すことを確認できたものの、学習時間がかかるという問題点も見つかった。そこで、この欠点を克服すべく、改良型増加法を提案し、二つの手法の比較も行った。その結果、改良型増加法は中間層ニューロン数、及び学習時間という点で単純増加法に比べ優れていることが明らかとなった。ただし、認識率がやや低下する点が見受けられたため、この問題克服が今後の課題となっている。

参考文献

[1] 麻生英樹："ニューラルネットワーク情報処理"，産業図書
[2] 熊沢逸夫："学習とニューラルネットワーク"，森北出版株式会社
[3] 押野隆弘, 尾島潤, 山本眞司："誤差逆伝搬学習における中間層ユニットの逐次削減法"，電子情報通信学会論文誌（D-Ⅱ）,Vol.J76-D-Ⅱ,No.7,pp.1414-1423,1993
[4] http://www.aso.ecei.tohoku.ac.jp/
[5] http://mars.elcom.nitech.ac.jp/java-cai/neuro/menu.html