直交化最小二乗法による階層型ニューラルネットワークの 中間層ニューロン数の削減法

楊 子 江*

Hidden-Layer Size Reducing for Multilayer Neural Networks Using the Orthogonal Least-Squares Method

Zi-Jiang YANG*

This paper proposes a new and computationally efficient approach to hidden-layer size reducing for multilayer neural networks. The author's attention is focused on minimizing hidden-layer redundancy using the orthogonal least-squares (OLS) method with the aid of Gram-Schmidt orthogonal transformation. The neural network with a large hidden-layer size is first trained via a standard training rule. Then the OLS method is introduced to identify and eliminate redundant neurons such that a simpler neural network is obtained. The OLS method is employed as a forward regression procedure to select a suitable set of neurons from a large set of preliminarily trained hidden neurons, such that the input to the output-layer is reconstructed with less hidden neurons. At each step of regression, the increment to the energy of the summation of the weighted outputs of the hidden neurons (the input to the output-layer) is maximized. And the weights of the links between the selected hidden neurons and the output layer is automatically determined through the regression procedure. Therefore the neurons which contribute trivially to the input to the output-layer can be eliminated without much distortion of the neural network output. Simulation results are included to show the efficiency of the proposed method.

Key Words: neural network, hidden neuron, orthogonal least-squares method, redundancy elimination

1. はじめに

バックプロパゲーション学習に基づく階層型ニューラルネットワークが、パターン認識、システム同定と制御などに数多く応用されている^{1)~5)}。この階層型ニューラルネットワークを個々の実際問題に適用する場合、問題に適したサイズのネットワークを用いることが非常に重要である。ネットワークのサイズが小さすぎると十分な学習ができないし、逆に大きすぎると過剰自由度のためオーバーフィッティングをおこして、ネットワークの汎化能力の低下を招く。理論的に、3層ニューラルネットワークの中間層ニューロン数が無限にとれば、任意の連続関数が3層ニューラルネットワークで任意の精度で近似できることはFunahashi⁶⁾およびHornik⁷⁾などによって証明されている。しかし、中間層のニューロン数に関しては、入出力のマッピングや、汎化能力に十分な個数を、試行錯誤的なシミュレーションで決定する場合が大半であり、多大の労力を必要としているのが現状である。

近年、適切な中間層ニューロン数を決定する手法は多くの研究者たちによって研究されている^{3), 8)}。学習の段階で、ニューロン数を逐次に加減しながら、最終的に最適なニューロン数を得ようとする方法は盛んに研究されている。大きく分けて、小さめのネットワーク構造から出発して、必要なニューロンを逐次追加していく生成的学習法と、逆に大きめのネットワーク構造から出発して、不要なニューロンを逐次削除していく削除的学習法という二つのタイプに分類できる⁸⁾。この二つのアプローチに関して、数多くの手法が提案されているが、それぞれの長所と短所があり、簡便で広く応用される手法はまだ多くない。多数の中間層ニューロン数の候補に対し、それぞれのネットワークを学習してから、AICに基づいて最良のネットワークを選択する手法もあるが⁹⁾、全体としての計算量が莫大であるほか、AICの使用に当たっては限らないといった問題点が指摘されている¹⁰⁾。なお、各手法の詳細については、文献3), 8)の解説を参照されたい。

本論文は、ネットワークの中間層ニューロンの出力の線形従属性を検出することによって、冗長と思われるニューロンを削除し、よりコンパクトなネットワーク構造を得る手法にスポットを当てる。この考えに基づいた手法は近年多数提案

* 九州工業大学情報工学部 飯塚市川津 680-4

* Faculty of Computer Engineering and Systems Science,
Kyushu Institute of Technology, Iizuka
(Received May 27, 1996)
(Revised October 28, 1996)

され、注目を集めている。増田ら¹¹⁾は学習の途中、出力の相関が強い二つのニューロンを合成することによって、ネットワークの単純化をはかったが、3個以上のニューロンの間に存在する線形従属性を検出することができない。中間層ニューロンの出力相関行列の微小特異値を検出することによって、適切なニューロン数を決める手法もあるが、具体的に何番目のニューロンが冗長であるかが分からず、微小特異値を判定するしきい値の設定とネットワークの近似精度との関係も明白ではない¹²⁾。中間層ニューロンに対する線形重回帰分析によって、線形従属の部分を取り除き、最適な中間層ニューロン数を求める方法も提案されており、良好な結果を得ているが¹³⁾、中間層ニューロンの処理順番の影響、非線形成分を単純に加え合わせて良いとの理論的根拠などをさらに論じる必要があると思われる。村田ら¹⁴⁾も線形従属性判定による中間層ニューロン数の削減法を提案したが、文献11),12),13)の手法と比べて、冗長と思われるニューロンを取り除くことがネットワークの近似精度にどのように影響を与えるかを近似的に数値で検討できること、縮約されたネットワークの中間層と出力層間の結合重みも再学習することなしにある程度決めることができるなどの利点があげられる。ただし、削除判定基準値は事前に決めておかなければならず、その適切な設定法を検討する必要がある。ほかに、以上にあげた手法と類似して、中間層ニューロンの主成分分析を用いた手法も提案されている¹⁵⁾。これらの手法は、中間層ニューロンの出力の線形従属性を中心に検討しているが、実際に、中間層の出力は、各ニューロンの出力の線形結合として、出力層の入力となるので、ネットワークにおける各中間層ニューロンの役割はその出力が出力層ニューロンの入力に対する寄与で評価することはもっと適切だと思われる。すなわち、中間層ニューロンの出力の線形従属性だけでなく、出力層との結合重みの影響も考慮して、総合的に判断する必要がある。

本研究は、システム同定分野で、モデル選択とパラメータ推定において、非常に有力な手法として、広く使われるGram-Schmidtの直交化に基づいた直交化最小二乗法^{16)~18)}による階層型ニューラルネットワークの中間層ニューロン数の削減法を提案する。提案される手法は、ネットワークの学習を一度行った後、直交化最小二乗法を用いて、出力層ニューロンの入力の自乗和への寄与が大きい順番で、中間層ニューロンを採用して行き、出力層ニューロンの入力を再構成する。ほかのニューロンの線形結合で表現できるニューロンまたは出力層との結合重みが小さいニューロンは、自然に後の方で選択される。また、選択の過程で、選択された各ニューロンの出力層との結合重みの新しい値、及び既に選択されたいくつかのニューロンによる出力層ニューロン入力の再構成における誤差も自動的に計算されるので、設計者が要求する表現力を満たす最小限のニューロン数を簡単に決められる。提案した手法の有効性はシミュレーションを通して明らかにする。

Fig. 1 A three layer neural network.

2. 問題の設定

本論文は、簡単のため、一般性を失わず、Fig. 1に示すような多入力1出力の3層ニューラルネットワークを考える。入力層、中間層、出力層のニューロンの数はそれぞれ $n_i, n_h, 1$ であり、入力層ニューロンは入力信号をそのまま出力するだけの作用をする。中間層の j 番目のニューロンの入力は次式で表わされる。

$$y_j(k) = \sum_{i=1}^{n_i} w_{ij}^h I_i(k) + b_j^h \quad (1)$$

ただし、 $I_i(k)$ は i 番目の入力層ニューロンに対する $k(k = 1, \dots, N)$ 番目の入力データである。 w_{ij}^h は i 番目の入力層ニューロンと j 番目の中間層ニューロンとの結合重みである。また、 b_j^h は j 番目の中間層ニューロンのバイアスである。中間層の出力は次のようなシグモイド関数の出力である。

$$h_j(k) = \frac{1}{1 + \exp(-y_j(k))} \quad (2)$$

出力層ニューロンへの入力は各中間層ニューロンの出力の線形結合

$$d(k) = \sum_{j=1}^{n_h} w_j^o h_j(k) + b^o \quad (3)$$

である。ただし、 w_j^o は j 番目の中間層ニューロンと出力層ニューロンとの結合重みであり、 b^o は出力層ニューロンのバイアスである。したがって、ネットワークの出力は

$$O(k) = f(d(k)) \quad (4)$$

となる。ここで、 $f(d(k))$ は問題に応じて、シグモイド関数 $f(d(k)) = 1/[1 + \exp(-d(k))]$ または線形関数 $f(d(k)) = d(k)$ とする。

一般に、中間層ニューロンが多すぎると、各ニューロンの間に強い線形従属関係が存在し、いくつかのニューロンの出力がほかのニューロンの出力の線形結合で表現できるので、結果的にこれらのニューロンは冗長であり、役割がほかのニューロンに肩代わりされることが可能であることは既に報

告されている^{11)~15)}。したがって、これらの冗長ニューロンを削除することによって、ネットワークの表現能力を大きく損なわなくても中間層ニューロン数を削減することができる。

そこで、本研究は、直交化最小二乗法^{16)~18)}による中間層ニューロン数の削減法を提案する。収束に十分な数の中間層ニューロンをもって、ネットワークの学習を一度行った後、教師入力を再入力した場合の出力層ニューロンの入力 $d(k) (k=1, \dots, N)$ の近似問題を検討する。すなわち、(3)式において、冗長な $h_j(k)$ を削減し、必要最小限の項で、 $d(k)$ を精度良く再構成する。具体的にいうと、モデル選択とパラメータ推定において、非常に有力な手法として広く使われてきた直交化最小二乗法を用いて、出力層入力の自乗和への寄与が大きい順番で、中間層ニューロンを採用していく。直交化最小二乗法の実行において、出力層ニューロンの入力に対する寄与が小さいニューロン、すなわちほかのニューロンの線形結合で表現できるニューロン及び出力層との結合重みが小さいニューロンは、自然に後の方で選択される。したがって、文献11),14)のように線形従属性判定のしきい値を事前に設定する必要がなく、 $d(k)$ またはネットワークの出力 $O(k)$ を再構成するときの近似誤差を評価することによって、設計者が要求する表現力を満たす最小限のニューロン数を簡単に決められるので、本手法は極めて柔軟性に富むと思われる。

3. ニューラルネットワークの中間層ニューロン数の削減法

Gram-Schmidt の直交化に基づいた直交化最小二乗法は、モデル選択とパラメータ推定において、非常に有力な手法として、近年注目されている。この手法が最低限の複雑さを持つモデルを構築しながら、非線形システムを精度よく同定できることはシミュレーションおよび実システムへの応用で確認されている^{16),17)}。ラジアル基底関数ネットワークの基底関数の選定においても、良好な結果が得られたことが示されている¹⁸⁾。本研究は、直交化最小二乗法によるシグモイドタイプの階層型ニューラルネットワークの中間層ニューロン数の削減法を提案する。

まず、(3)式を次のように書き直す。

$$\begin{aligned} d(k) &= z^T(k) w^o \\ z^T(k) &= [h_1(k), h_2(k), \dots, h_{n_h+1}(k)] \\ w^o &= [w_1^o, w_2^o, \dots, w_{n_h+1}^o]^T \end{aligned} \quad (5)$$

ここで、出力層ニューロンのバイアス b^o は、出力が常に $h_{n_h+1}(k) = 1$ という $n_h + 1$ 番目の仮想中間層ニューロンと出力ニューロンとの結合重みと見なされている。すなわち、 $w_{n_h+1}^o = b^o$ 。本研究では、バイアスは削除の対象とならないとする。さらに、(5)式を $k = 1, \dots, N$ に対して、次式のようにまとめる。

$$\begin{aligned} d &= P w^o \\ d &= [d(1), d(2), \dots, d(N)]^T \\ P &= [p_1, p_2, \dots, p_{n_h+1}] \\ p_j &= [h_j(1), h_j(2), \dots, h_j(N)]^T \\ &\quad (j = 1, 2, \dots, n_h + 1) \end{aligned} \quad (6)$$

そこで、直交化最小二乗法でより少ないニューロンで $d(k)$ を再構成することを考える。いま、何らかの方法で、各中間層ニューロンの出力ベクトル $p_j (j = 1, 2, \dots, n_h + 1)$ を、それぞれが採用された場合の $d(k)$ の自乗和に対する寄与が大きい順番で並べ替えて、上式を次のように書き直す。

$$\begin{aligned} d &= P' c_o \\ c_o &= [c_1, c_2, \dots, c_{n_h+1}]^T \\ P' &= [p'_1, p'_2, \dots, p'_{n_h+1}] \\ p'_j &= [p'_j(1), p'_j(2), \dots, p'_j(N)]^T \\ &\quad (j = 1, 2, \dots, n_h + 1) \end{aligned} \quad (7)$$

直交化最小二乗法は、回帰行列 P' の列ベクトル $p'_j (j = 1, \dots, n_h + 1)$ を直交基底ベクトル $r_j (j = 1, \dots, n_h + 1)$ に変換している。ここで、 P' は次のように分解される。

$$P' = R A_o \quad (8)$$

ただし、

$$A_o = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1(n_h+1)} \\ 0 & 1 & \alpha_{23} & \cdots & \alpha_{2(n_h+1)} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & & \ddots & 1 & \alpha_{n_h(n_h+1)} \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

R は $N \times (n_h + 1)$ 行列で、その列ベクトル

$$r_j = [r_j(1), r_j(2), \dots, r_j(N)]^T \quad (j = 1, 2, \dots, n_h + 1) \quad (10)$$

は互いに直交である。

$$R^T R = B \quad (11)$$

ただし、 B は $(n_h + 1) \times (n_h + 1)$ 対角行列で、対角要素 b_j は次のようである。

$$\begin{aligned} b_j &= r_j^T r_j \\ &= \sum_{k=1}^N r_j(k) r_j(k) \\ &\quad (j = 1, 2, \dots, n_h + 1) \end{aligned} \quad (12)$$

以上より、(7)式は次のように書ける。

$$d = R g_o \quad (13)$$

ただし、

$$\begin{aligned} g_o &= A_o c_o \\ &= [g_1, g_2, \dots, g_{n_h+1}]^T \end{aligned} \quad (14)$$

は直交変換後の新しいパラメータベクトルであり、その推定値 $\hat{g}_o$ は次式で与えられる。

$$\hat{g}_0 = B^{-1} R^T d \quad (15)$$

すなわち,

$$\hat{g}_j = \frac{r_j^T d}{r_j^T r_j} \quad (j=1, 2, \dots, n_h+1) \quad (16)$$

基底ベクトル r_j は互いに直交であるので、いくつかの項を削除したり、新たな項を加えたりしても、すでに計算された $\hat{g}_0$ のほかの要素に影響を与えない。また、各基底ベクトルによる $d(k)$ の自乗和への寄与の計算が可能となる。これらの性質は、ネットワークの構造決定において、非常に重要である。ここで、 n_s 個の中間層ニューロンだけを採用した場合、すなわち、バイアス項を含む n_s+1 個の項だけを残して、ほかの項を削除した場合を考える。採用された中間層ニューロンの出力層との結合重みベクトル c の推定値は次の関係で簡単に求められる。

$$A \hat{c} = \hat{g} \quad (17)$$

ただし,

$$\begin{aligned} \hat{c} &= [\hat{c}_1, \hat{c}_2, \dots, \hat{c}_{n_s+1}]^T \\ \hat{g} &= [\hat{g}_1, \hat{g}_2, \dots, \hat{g}_{n_s+1}]^T \end{aligned} \quad (18)$$

$$A = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1(n_s+1)} \\ 0 & 1 & \alpha_{23} & \cdots & \alpha_{2(n_s+1)} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & & \ddots & 1 & \alpha_{n_s(n_s+1)} \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

なお、 P' は Gram-Schmidt 法によって、次のように直交化される。

$$\left. \begin{aligned} r_1 &= p'_1 \\ \alpha_{ij} &= \frac{r_i^T p'_j}{r_i^T r_i} \quad (1 \leq i < j) \\ r_j &= p'_j - \sum_{i=1}^{j-1} \alpha_{ij} r_i \end{aligned} \right\} \quad (20)$$

($j=2, \dots, n_h+1$)

実際に、列ベクトルに関する直交変換を行う場合、(8) 式のような大きいサイズのマトリクスに関する計算を直接行う必要がなく、ニューロンの出力マトリクス P の n_h+1 個の列ベクトルから、 $d(k)$ の自乗和に対する寄与がもっとも大きいものを選び、直交変換を行いながら、マトリクス P' と R の列ベクトルを順次構築していくことが可能である^{16)~18)}。

(7) または (13) 式において、 n_s+1 個の項だけを残して、ほかの項を削除した場合、 $d(k)$ の自乗和は

$$d^T d = \sum_{j=1}^{n_s+1} g_j^2 r_j^T r_j + E^T E \quad (21)$$

となる。ここで、 E は $d(k)$ を再構成するときの近似誤差ベクトルである。

$$E = [e(1), e(2), \dots, e(N)]^T \quad (22)$$

ただし、(21) 式より

$$\begin{aligned} e(k) &= \sum_{j=n_s+2}^{n_h+1} r_j(k) g_j \\ &= d(k) - \sum_{j=1}^{n_s+1} r_j(k) g_j \\ &= d(k) - \sum_{j=1}^{n_s+1} p'_j(k) c_j \end{aligned} \quad (23)$$

(21) 式の右辺の第1項は採用されたベクトル r_j による $d(k)$ の自乗和への寄与の総和を表す部分であり、第2項は中間層ニューロンの削除による $d(k)$ を再構成する場合の自乗誤差である。したがって、 $g_j^2 r_j^T r_j$ はベクトル r_j を取り入れることによる $d(k)$ の自乗和の増加分を意味する。そこで、ベクトル r_j の導入による誤差減少率 (error reduction ratio) を

$$[err]_j = \frac{g_j^2 r_j^T r_j}{d^T d} \quad (24)$$

と定義する。これより、マトリクス P の列ベクトルの内で、 $[err]_j$ を最も大きくするベクトルを逐次選んで、直交変換を行っていくという Feedforward regression アルゴリズムを用いて、必要最小限の中間層ニューロンで $d(k)$ を再構成すればよい。具体的アルゴリズムは次のようである。

ステップ A

バイアスによる n_h+1 番目の仮想中間層ニューロンは削除されないで、一番最初に採用される。

$$r_1 = p_{n_h+1}$$

$$\hat{g}_1 = \frac{r_1^T d}{r_1^T r_1} \quad (25)$$

$$[err]_1^{(i_1)} = \frac{g_1^2 r_1^T r_1}{d^T d}$$

ただし、 $i_1 = n_h+1$ 。

ステップ B

中間層ニューロンの出力ベクトル p_i から $[err]_j$ を最大にするベクトルを逐次を選び、直交ベクトル r_j とパラメータ推定値 $\hat{g}_j$ を計算する。ただし、 $j=2, \dots, n_s+1$ ($n_s \leq n_h$)、 $1 \leq i \leq n_h, i \neq i_1, \dots, i_{j-1}$ 。ここで、 $i_1, \dots, i_{j-1}$ はすでに選ばれた p_i の番号である。

まず j 番目に採用されるベクトルの各候補を取り入れた場合の誤差減少率を計算する。

$$\left. \begin{aligned} \alpha_{mj}^{(i)} &= \frac{r_m^T p_i}{r_m^T r_m} \quad (1 \leq m < j) \\ r_j^{(i)} &= p_i - \sum_{m=1}^{j-1} \alpha_{mj}^{(i)} r_m \\ \hat{g}_j^{(i)} &= \frac{(r_j^{(i)})^T d}{(r_j^{(i)})^T r_j^{(i)}} \\ [err]_j^{(i)} &= \frac{(g_j^{(i)})^2 (r_j^{(i)})^T r_j^{(i)}}{d^T d} \end{aligned} \right\} \quad (26)$$

($1 \leq i \leq n_h, i \neq i_1, \dots, i_{j-1}$)

次に、誤差減少率を最大にする列ベクトル（ j 番目に採用された中間層ニューロン）の番号 i_j を見つける。すなわち

$$[\text{err}]_j^{(i_j)} = \max\{[\text{err}]_j^{(i)}, 1 \leq i \leq n_h, i \neq i_1, \dots, i_{j-1}\}$$

を見つける。この場合、 r_j は次のようになる。

$$\begin{aligned} r_j &= r_j^{(i_j)} \\ &= p_{i_j} - \sum_{m=1}^{j-1} \alpha_{mj} r_m \end{aligned} \quad (27)$$

ただし、

$$\alpha_{mj} = \alpha_{mj}^{(i_j)} \quad (m=1, \dots, j) \quad (28)$$

なお、パラメータ推定値は $\hat{g}_j = \hat{g}_j^{(i_j)}$ として求まる。また、 j 番目に選択されたニューロンの出力ベクトルは $p'_j = p_{i_j}$ として得られる。(26)~(28) 式の計算は、 $d(k)$ を再構成する場合の相対自乗誤差がしきい値 ρ より小さくなり、すなわち

$$1 - \sum_{j=1}^{n_s+1} [\text{err}]_j^{(i_j)} < \rho \quad (29)$$

が成立するまで、 $j=2, \dots, n_s+1$ に対して、繰り返される。この場合、 n_s 個の中間層ニューロンが採用され、ほかの冗長と思われるニューロンは切り捨てられる。

ステップ C

選択された中間層ニューロンの出力層ニューロンとの結合重み及び出力層ニューロンのバイアスを (17) 式に基づいて計算し直す。

(26)~(28) 式の繰り返し計算は、各繰り返しのステップで、まだ選ばれていないニューロンの候補の中から、 $d(k)$ の自乗和に対する寄与を最大にするものを選び出し、すでに選択されたニューロンの出力ベクトル $p'_1, \dots, p'_{j-1}$ によって張られる空間の次元を $j-1$ から j へ拡張している。ゆえに、ほかのニューロンとの線形従属性が強いニューロンまたは出力層との結合重みが小さいニューロンは、貢献度が小さいので、後の方で選択される。特に、(26) 式において、 $(r_j^{(i)})^T r_j^{(i)} \approx 0$ となる場合、 p_i はすでに選択されたニューロンの出力ベクトル $p'_1, \dots, p'_{j-1}$ の線形結合ではほぼ完全に表現できるので、 $d(k)$ の自乗和に対する寄与などを調べなくても、すぐ削除すればよい。したがって、直交化最小二乗法は、冗長な中間層ニューロン数の削減において、非常に簡便でしかも有力な手法であると思われる。

必要なニューロン個数を決めるしきい値 ρ を適切に決めることも重要である。 ρ が小さすぎると、冗長性を許すことになるが、 ρ が大きすぎると、より多くのニューロンを削除してしまい、ネットワークの表現能力を損なう恐れがある。著者の経験では、 ρ の値が 0.001 ~ 0.01 前後、すなわち、 $d(k)$ を再構成する時の相対自乗誤差が 0.1 ~ 1% 前後であれば、無難である。文献 [11], [14] では、冗長性を判定するしきい値は事前に決めなければならないので、試行錯誤的にやり直すとき、最初から計算し直さなければならない。一方、本手法では、各ニューロンをその重要度に応じて並べ替えた後

で ρ を決めることができる。また、 ρ の値は並べ替えられたニューロンを第何個までとればよいかとの結論だけに影響を与えるので、労力のいる試行錯誤的やり直しは必要としない。したがって、本手法は実用上非常に柔軟性に富むと思われる。

実用上、このように $d(k)$ の近似誤差を評価すれば、必要最小限のニューロンの選定は効率よくできるが、中間層ニューロン数の削減はネットワークの近似精度にどれほど影響を与えるのかを調べたい場合、再構成した出力層ニューロンの入力 $\hat{d}(k)$ を用いて、(4) 式より縮約されたネットワークの出力を計算することができる。すなわち、ネットワークの出力誤差で評価することによって、ニューロンを何個まで採用すればよいのかを決めることもできる。

本手法では、削減直後のネットワークの学習誤差は多少大きくなるので、さらに学習を行った方が無難である。しかし、縮約されたネットワークの結合重みをそのまま初期値とすれば、さらなる学習が必要とする学習回数はそれほど多くない。

以上の議論より、提案した手法の全体のアルゴリズムは次のようである。

ステップ 1

十分大きな中間層ニューロン数、結合重みなどの初期設定を行ってから、収束条件または最大学習回数を満たすまでバックプロパゲーションや、ニュートン法などの学習則でネットワークの学習を行う。

ステップ 2

直交化最小二乗法を用いて、必要最小限の中間層ニューロンを選定する。ニューロン数の削減が必要でない場合、終了する。ニューロン数の削減があった場合、ステップ 3 へ進む。

ステップ 3

縮約されたネットワークの結合重みをそのまま初期値として、収束条件または最大学習回数を満たすまでネットワークの学習をさらに行ってから終了する。

提案する手法の実行に当たって、ネットワークの学習アルゴリズムが局所的最適解に陥る場合や、初期設定の中間層ニューロン数が非常に多く、過学習の効果が非常に大きい場合、1 回の学習と削減だけでは十分でなく、上記のアルゴリズムを 2 ~ 3 回繰り返して実行した方が無難である。

4. シミュレーション結果

4.1 関数近似問題

提案したアルゴリズムの有効性を示すために、まず、次の連続関数の近似問題を考える。

$$\begin{aligned} g(x) &= 0.5 + 0.2 \tanh(10x - 2) \\ &\quad - 0.3 \tanh(15x - 6) + 0.2 \tanh(20x - 12) \\ &\quad - 0.2 \tanh(5x - 4) \end{aligned} \quad (30)$$

Table 1 Determined number of hidden neurons

Initial	30	20	10	8	6	5	4	3
Reduced	5	5	4	4	4	4	4	3

Table 2 Training error of the neural network.

Initial number	30	20	10	8	6	5	4	3
Preliminary training	4.5×10^{-4}	4.8×10^{-4}	4.8×10^{-4}	4.2×10^{-4}	4.8×10^{-4}	4.9×10^{-4}	5.0×10^{-4}	1.4×10^{-2}
After pruning	1.6×10^{-2}	1.2×10^{-2}	3.6×10^{-3}	2.3×10^{-2}	1.3×10^{-2}	4.2×10^{-3}	—	—
Further training	5.0×10^{-4}	4.8×10^{-4}	4.6×10^{-4}	3.7×10^{-4}	4.7×10^{-4}	4.6×10^{-4}	—	—

Table 3 Epochs of training.

Initial number	30	20	10	8	6	5	4	3
Preliminary training	33	46	71	71	74	143	821	1000
Further training	19	35	10	5	68	4	—	—

Fig. 2 Training data of $g(x)$.

教師データは Fig. 2 に示されるような、 $x \in [0, 1]$ の範囲内、刻み $\delta x = 0.02$ でサンプルした $N = 51$ 組の関数の入出力データを使った。学習則は Matlab のニューラルネットワークツールボックス¹⁹⁾にある `trainlm` という関数を使った。Levenberg-Marquardt と呼ばれる Newton 法の近似を使用することによって、勾配降下に基づくバックプロパゲーション法より収束がはるかに速いことが関数 `trainlm` を使う主な理由である。中間層ニューロン数の削減という目的では、Newton 法とバックプロパゲーション法はほぼ同じ結果を出すことをすでに確認している。`trainlm` の使用に当たって、終了条件は最大学習回数 1000 回、または学習誤差 $e_l = \sum_{k=1}^{51} [O(k) - g(k)]^2 \leq 0.0005$ とした。また、ほかのパラメータはすべてデフォルト値を使った。用意した 3 層ネットワークの出力層ニューロンをシグモイド関数と選んだ。中間層ニューロン数の初期値を 3, 4, 5, 6, 8, 10, 20, 30 とし、それぞれに対して学習を行ってから、1 回の削減を行った。ただし、必要なニューロン個数を決めるしきい値を $\rho = 0.01$ とした。ニューロン数の判定結果を Table 1 に示した。Table 1 の結果から、中間層ニューロン数の初期値を 4, 5, 6, 8, 10 とした場合、いずれも適切な中間層ニューロン数 4 個という結果を得た。しかし、中間層ニューロン数の初期値を 20 と 30 とした場合、中間層ニューロン数は 5 個であるという結果になった。前述のように、中間層ニューロン数の初期設定値が非常に大きい場合、1 回の削減だけでは十分でなく、学

習と削減を 2~3 回繰り返せば、すなわち、中間層ニューロン数を 5 と設定してもう一度学習してから削減すれば、適切な中間層ニューロン数 4 個という結果が得られるのは明らかである。30 個のニューロンから一気に 5 個まで削減できたことは提案した手法が非常に強力であることを反映している。本手法は、中間層ニューロンの出力ベクトルの直交基底ベクトルによって張られる部分空間の次元数に基づいて、中間層ニューロンの数を決めている。しかし、この部分空間の次元数を数えるとき、どうしても人間の主観的（あいまいな）判断に頼らざるを得ない。本論文は、(29) 式のように相対誤差のしきい値 ρ を使って判断している。前述のように、必要なニューロン個数を決めるしきい値 ρ を適切に決めることも重要である。 ρ が小さすぎると、冗長性を許すことになるが、 ρ が大きすぎると、より多くのニューロンを削除してしまい、ネットワークの表現能力を損なう恐れがある。一般に初期設定の中間層ニューロン数が非常に大きい場合、もっと複雑な曲線を近似することができ、過学習の効果も非常に大きいと予想されるので、中間層ニューロンの出力が張る部分空間の次元数も多少（あいまいであるが）大きくなる傾向がある。したがって、初期設定の中間層ニューロン数が非常に大きい場合、しきい値 ρ を大きめに、初期設定の中間層ニューロン数がそれほど大きくない場合、しきい値 ρ を小さめにした方がよいが、このように具体的値を設定することは面倒で難しい。したがって、削除するのを多少保守的にして、 ρ を小さめに設定し、2~3 回にわたって学習と削除を行った方が賢明である。

削減直前、削減直後、および削減した後さらに学習を行った場合のネットワークの誤差を Table 2 に示した。中間層ニューロン数の初期値を 4, 5, 6, 8, 10, 20, 30 とした場合、学習誤差はいずれも収束条件 $e_l \leq 0.0005$ を満足している。ニューロン数を削減した後では、誤差は多少上昇しているが、縮約されたネットワークの結合重みをそのまま初期値として、さらにネットワークの学習を行うと、学習誤差はまた収束条件 $e_l \leq 0.0005$ を満足しており、提案した手法は必要最小限のニューロン数を正確に判定していることが分かる。一方、中間層ニューロン数 3 個の場合では、学習誤差が収束条件を満足していないので、ニューロン数が不十分であることが分かる。

削減前と削減後の必要とした学習回数を Table 3 に示した。削減後のさらなる学習は縮約されたネットワークの結合重みをそのまま初期値としているので、必要とした学習回数は削減前の学習の回数より少ないことが確認できる。

Table 5 Training error of the neural network.

Initial number	30	20	10	9	8	7	6	5	4
Preliminary training	1.0×10^{-2}	9.9×10^{-3}	1.0×10^{-2}	1.0×10^{-2}	1.0×10^{-2}	1.0×10^{-2}	1.1×10^{-2}	1.1×10^{-2}	4.2×10^{-2}
After pruning	4.8×10^{-1}	2.4×10^{-1}	2.0×10^{-1}	6.0×10^{-1}	5.0×10^{-2}	1.2×10^{-1}	1.1×10^{-1}	—	—
Further training	1.0×10^{-2}	1.0×10^{-2}	1.0×10^{-2}	1.1×10^{-2}	1.1×10^{-2}	1.1×10^{-2}	1.1×10^{-2}	—	—

Table 4 Determined number of hidden neurons

Initial	30	20	10	9	8	7	6	5	4
Reduced	10	8	8	7	5	5	5	5	4

4.2 システム同定問題

ここで、次式で表わされる非線形システムの同定を考える²⁰⁾。

$$y(k+1) = \frac{y(k)}{1+y^2(k)} + u^3(k) \quad (31)$$

入力信号を次のように選んだ。

$$u(k) = \sin\left(\frac{2\pi k}{25}\right) + \sin\left(\frac{2\pi k}{10}\right) \quad (32)$$

出力の観測値に低レベルで、ゼロ平均の白色雑音 $n(k)$ を加えた。

$$z(k) = y(k) + n(k) \quad (33)$$

用意したニューラルネットワークは、2入力 ($u(k), y(k)$) 1出力 ($y(k+1)$) 3層で、出力層ニューロンを線形関数とした。ただし、実際に同定を行うとき、出力の真値 $y(k)$ のかわりに観測値 $z(k)$ を使った。同定に使われるデータは $k = 1, \dots, 101$ の範囲でサンプルした $N = 100$ 組のシステムの入出力データを使った。ただし、観測雑音の標準偏差を 0.01 とした。学習則は前の例題と同じく関数 trainlm を使った。終了条件は最大学習回数 1000 回、または学習誤差 $e_l = \sum_{k=2}^{101} [O(k) - y(k)]^2 \leq 0.01$ とした。中間層ニューロン数の初期値を 4, 5, 6, 7, 8, 9, 10, 20, 30 とし、それぞれに対して学習を行ってから、1回の削減を行った。ただし、必要なニューロン個数を決めるしきい値を $\rho = 0.001$ とした。結果を Tables 4, 5 に示した。Table 4 の結果から、中間層ニューロン数の初期値を 5, 6, 7, 8 とした場合、いずれも適切な中間層ニューロン数 5 個という結果を得た。中間層ニューロン数の初期値を 9, 10, 20, 30 とした場合、1回の削減だけでは十分でなかったが、中間層ニューロン数の初期値をそれぞれ 7, 8, 8, 10 と設定し直してもう 1 回学習してから削減すれば、最終的に中間層ニューロン数 5 個という結果が得られることがわかる。Table 5 に示されているネットワークの学習誤差からも、適切な中間層ニューロン数は 5 個であることが確認できる。

ネットワークの近似能力を維持しながら、中間層ニューロン数を削減してネットワークのスリム化を実現することで、オーバーフィッティングの防止や汎化能力の向上といったことが期待できる。この効果を示すため、新たな入力として、 $[-1, 1]$ の範囲で一様分布する乱数を (31) 式のシステムのモデル、削減前のネットワークおよび削減後のネットワークにそれぞれ入力し、出力の真値 $y(k)$ とその推定値 $O(k)$ を計算した。ま

た、ネットワークの汎化誤差を $e_g = \sum_{k=2}^{101} [O(k) - y(k)]^2$ と定義し、その計算結果を Table 6 に示した。Table 6 より、中間層ニューロン数の初期設定値が大きい場合、ネットワーク汎化誤差も大きいですが、削減を行った後、汎化誤差も大幅に減少したことが確認できる。すなわち、提案した手法によってネットワークの汎化能力が向上したことがいえる。なお、ネットワークの汎化能力をもっと厳密に論じるには、ネットワークのスリム化だけでなく、学習アルゴリズムにも工夫しなければならないと思われる²¹⁾。

よく知られているように、一般にニューラルネットワークの学習において、重みの初期値によって、学習結果が異なることがある。中間層ニューロン数の選定においても、重みの初期値によっては、選定されたニューロン数が異なる（大体の場合、1, 2 個の差）ことがあるが、前述のように学習と削減を 2~3 回繰り返したり、または重みの初期値を 2~3 試したりすると、ほとんどの場合、必要最小限のニューロン数が得られることを確認した。

5. 終わりに

本研究は、システム同定の分野で、モデル選択とパラメータ推定において、非常に有力な手法として、広く使われる直交化最小二乗法による階層型ニューラルネットワークの中間層ニューロン数の削減法を提案した。提案した手法は、ネットワークの学習を一度行った後、直交化最小二乗法を用いて、出力層ニューロンの入力の自乗和への寄与が大きい順番で、中間層ニューロンを採用して行き、出力層ニューロンの入力を再構成する。ほかのニューロンの線形結合で表現できるニューロンまたは出力層との結合重みが小さいニューロンは選ばれないので、設計者が要求する表現力を満たす最小限のニューロン数を簡単に決められる。提案した手法の有効性はシミュレーションを通して示した。本論文は、簡単のため、多入力 1 出力のネットワークについて検討したが、提案されたアルゴリズムを拡張すれば、多入出力のネットワークの場合にも対応できる¹⁶⁾。

参考文献

- 1) D. R. Hush and B. G. Horne: Progress in supervised neural networks, IEEE Signal Processing Magazine, 6-1 8/39 (1993)
- 2) K. J. Hunt, D. Sbarbaro, R. Zbikowski and P. J. Gawthrop: Neural networks for control systems—a survey, Automatica, 28-6, 1083/1112 (1992).
- 3) 阿部重夫: ニューラルネットとファジィシステム [理論と応用], 近代科学社 (1995)
- 4) 馬場, 小島, 小澤: ニューラルネットの基礎と応用, 共立出版 (1994)
- 5) 西川, 北村: ニューラルネットと計測制御, 朝倉書店 (1995)

Table 6 Generalization error of the neural network.

Initial number	30	20	10	9	8	7	6	5	4
Preliminary network	4.5×10^{-1}	3.8×10^{-2}	1.1×10^{-2}	2.1×10^{-2}	5.0×10^{-2}	3.1×10^{-2}	1.1×10^{-2}	3.2×10^{-3}	1.2×10^{-1}
Reduced network	1.5×10^{-2}	7.7×10^{-3}	9.0×10^{-3}	6.9×10^{-3}	3.2×10^{-3}	3.2×10^{-3}	3.2×10^{-3}	—	—

- 6) K. Funahashi: On the approximate realization of continuous mappings by neural networks, Neural Networks, **2**-3, 183/192 (1989)
- 7) K. Hornik, M. Stinchcombe and H. White: Multilayer Feedforward Networks are Universal Approximators, Neural Networks, **2**-5, 359/366 (1989)
- 8) 石川真澄: ネットワーク学習アルゴリズムの最近の話題, 計測と制御, **30**-4, 285/290 (1991)
- 9) 栗田多喜夫: 情報量基準による3層ニューラルネットワークの隠れ層のユニット数の決定法, 電子情報通信学会論文誌, **J73-DII**-11, 1872/1878 (1990)
- 10) 喜多一: ニューラルネットの汎化能力, システム/制御/情報, **36**-10, 625/633 (1992)
- 11) 増田, 藤井, 池谷, 相部: 隠れユニットの合成による階層型ニューラルネットワークのコンパクト構造化, 計測自動制御学会論文集, **28**-4, 519/527 (1992)
- 12) Psychogios D. C. and L. H. Ungar: SVD-net: an algorithm that automatically selects network structure, IEEE Trans. on Neural Network, **5**-3, 513/515 (1994)
- 13) 鹿山, 阿部, 武長, 諸岡: 多層ニューラルネットの最適中間層ニューロン数決定法, 電気学会論文誌, **112-D**-11, 1064/1070 (1992)
- 14) 村田, 藤井, 池田, 平沢, 相良: 冗長性判定に基づく階層型ニューラルネットワークの構造決定法, 計測自動制御学会論文集, **31**-2, 236/243 (1995)
- 15) 金指正和: 主成分分析を用いた階層型ニューラルネットの縮約法, 電気学会論文誌, **115-C**-9, 1059/1065 (1995)
- 16) S. Chen, S. A. Billings and M. J. Korenberg: Identification of MIMO non-linear systems using a forward-regression orthogonal estimator, International Journal of Control, **49**-6, 2157/2189 (1989)
- 17) S. Chen, S. A. Billings and W. Luo: Orthogonal least squares methods and their application to non-linear system identification, International Journal of Control, **50**-5, 1873/1896 (1989)
- 18) S. Chen, S. A. Billings, C. F. N. Cowan, and P. M. Grant: Practical identification of NARMAX models using radial basis functions, International Journal of Control, **52**-6, 1327/1350 (1990)
- 19) H. Demuth and M. Beale: Neural Network Toolbox ユーザーズガイド, サイバネットシステム (1996)
- 20) K. S. Narendra and K. Parthasarathy: Identification and control of dynamical systems using neural networks, IEEE Trans. on Neural Network, **1**-1, 4/27 (1990)
- 21) 渡辺英治: 関数近似問題に対する階層型ニューラルネットワークの汎化能力改善学習法, 電子情報通信学会論文誌, **J79-D-II**-5, 924/931 (1996)

[著者紹介]

楊 子 江 (正会員)

1992年3月九州大学大学院工学研究科電気工学専攻博士課程修了。同年4月九州工業大学助手。1996年4月同大学情報工学部助教授、現在に至る。システム同定およびそれに関連する信号処理、遺伝的アルゴリズム、ニューラルネットワーク、ウェーブレット解析、モーションコントロールなどの研究に従事(工学博士)。システム制御情報学会、電子情報通信学会、電気学会、ロボット学会、日本機械学会会員。