



TRIAM-1M におけるフィードバック制御に用いる ニューラルネットワークの構造最適化

彌 政 敦 洋, 竹 田 辰 興¹⁾, 中 村 一 男²⁾, 伊 藤 智 之²⁾
(九州大学大学院総合理工学研究科, ¹⁾電気通信大学電気通信学部, ²⁾九州大学応用力学研究所)

Optimization of Neural Network Structure for Feedback Control of TRIAM-1M

IYOMASA Atsuhiko, TAKEDA Tatsuoki¹⁾, NAKAMURA Kazuo²⁾ and ITOH Satoshi²⁾
Interdisciplinary Graduate School of Engineering Sciences, Kyushu University, Kasuga 816-8580, Japan
¹⁾*Faculty of Electro-Communications, The University of Electro-communications, Chofu 182-8585, Japan*
²⁾*Research Institute for Applied Mechanics, Kyushu University, Kasuga 816-8580, Japan*

(Received 9 February 2000/ Accepted 21 July 2000)

Abstract

A neural network for feedback control must be compact for insuring fast executing with high accuracy. A procedure of determining structures of input signals and 2 hidden layers is developed in order to construct the network for TRIAM-1M feedback control with high accuracy and high speed. From relation between degradation of neural network performance and effect to output by removing unit, two expressions of redundancy is necessary. The structures depend on an expression of redundancy and a process of the determination. The most useful procedure is iteration of removing the most redundant unit until the redundancy becomes under a criterion of removal. The number of units in the network developed can be reduced to about 60% of units used in the original network without the degradation of the accuracy. The validity of the procedure is verified by the neural networks, which is constructed to calculate plasma position and learned with the supervisor data is calculated by equilibrium code on TRIAM-1M.

Keywords:

neural network, real time, structure design, redundancy test, feedback control

1. はじめに

ニューラルネットワークは人の脳神経を模した情報処理技術という一面を持っており、様々な分野で応用がなされている。これらの応用の一つとして学習・汎化の能力を活用した高速高精度実験データ処理がある。トカマクプラズマの研究に関しては Lister *et al.* [1] がプラズマ平衡量の高速計算に用いたのをはじめとして様々な応用がなされてきた。これらの応用においては教師データを

author's e-mail: iyomasa@triam.kyushu-u.ac.jp

用いた学習を前提とする階層型ニューラルネットワークにより数値計算を行っている。これにより、学習データの作成およびニューラルネットワークの学習に時間はかかるものの、一般に他の方法では時間のかかる計測データの解析を短時間で実行することが可能となる。これはニューラルネットワークによるデータ処理を行う上での最大の利点であり、この特長を生かしてリアルタイムデータ処理や制御およびそれを前提とした研究がなされ

ている。ニューラルネットワークをリアルタイムで使用するプラズマ核融合研究分野の応用対象としては逆変換による実験データ処理[2]，実時間制御[3-6]，およびディスラプション等の予測[7-9]がある。

ニューラルネットワークをリアルタイムで使用する際には高速性と高精度という相反する要求を満たす必要がある。出力への寄与の大きい入力のみを選択することで実用的な精度を維持しつつ，ニューラルネットワークへ信号を入力するまでの処理に要する時間を短縮できる。それに加えてニューラルネットワークの入力決定は計測系の設計[10]や物理的解釈[5]への貢献もする。因果関係が数式で与えられていない種々の組み合わせの計測データ（磁気計測，分光計測他）とプラズマパラメータ（出力データ）の間に写像関係が構築できることはニューラルネットワークの特長の一つであり，それぞれの入力の出力に対する寄与を調べることでそれぞれの重要性がわかる。また隠れ層の各ユニットにおける計算時間がニューラルネットワークの実行時間のうちで大きな比率を占めることから，隠れ層のユニット数を必要最小限にすることが高速化のために重要となる。隠れ層の構造決定の方法として冗長性判定に基づく構造決定法[11]があるが，これは過学習の緩和を目的とするものであって，ここでの目的とは異なっている。

出力に対する重要度を用いた入力決定法では選択する際の基準の設定が困難であるが，冗長性判定に基づき入力決定を行う場合には各入力の重要度と独立性をネットワーク出力への影響という形で評価するので削除基準の設定が比較的容易である。過学習の緩和を目的とした構造決定の際にはネットワーク性能が劣化しない範囲でユニットを削除することで目的を達成でき，再学習の必要はない。またネットワークの変化が小さいので得られる構造は決定の手順に影響されにくいと考えられる。しかし，高速化を目的として構造決定を行う場合には，多数のユニットを削除する必要があるため構造決定の途中におけるネットワーク性能の劣化を許容し再学習を行う。このため冗長性判定の妥当性は構造決定の手順に依存し，構造決定の順序を変えることで得られる構造は異なってしまう。

そこで，「冗長性判定に基づく隠れ層の構造決定法」を入力決定および高速化を目的とした隠れ層のユニット削除に用いる場合の手法を確立し，ユニットの削除基準の設定に関する知見を得ることを目的として本研究を行った。対象とするニューラルネットワークは入力を磁気コイル信号，出力をプラズマ垂直位置(Z_p)あるいはプラ

ズマ水平位置(R_p)とする多入力1出力で複数の隠れ層を持つもので，TRIAM-1M トカマクの単一Xポイント配位プラズマの平衡解析データから学習用データおよび検証用データを作成した。

以下，2章にニューラルネットワークの概要，3章に構造決定方法を記し，4章にトカマク実験に用いるニューラルネットワーク構造の決定の数値実験を記して5章はまとめにあてる。

2. ニューラルネットワーク

使用したニューラルネットワークはFig. 1に示すような入力層，2つの隠れ層および出力層で構成されている階層型ニューラルネットワークである。隠れ層ユニットの σ は活性化関数がシグモイド関数であることを示している。シグモイド関数は $\sigma(x) = (1 + \exp(-x)) / (1 - \exp(-x))$ で表される関数である。出力層ユニットの Σ は入出力関係が線形和のみで非線形な活性化関数は使われていないことを表す。出力層の写像は線形で行っているが，これはネットワーク出力が ± 1 をこえた場合，すなわち学習範囲を外れた場合に対応する可能性を残すためである。また，ニューラルネットワークは誤差逆伝播法[12]により10万回の反復計算で学習を行った。

学習用データセットは平衡解析コードを用いて作成し

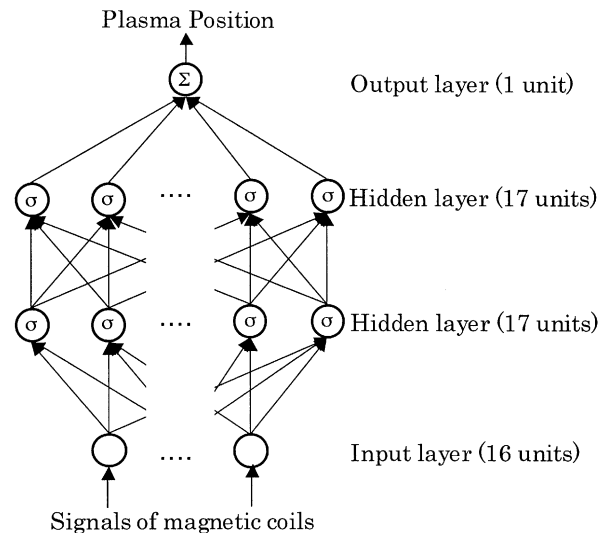


Fig. 1 Structure of a neural network with 2 hidden layers. An activation function of a unit in the hidden layers is a sigmoidal function. On the other hand, an activation function of a unit in the output layer is not nonlinear activation function. This architecture can deal with output data, the values of which are not in a range of a learning dataset.

た. このデータの中には単一 X ポイント配位, リミタ配位の両方の MHD 平衡が含まれており, 楕円度, プラズマ位置を変化させてデータを作成している. 検証用データセットも平衡解析コードを用いて学習用データとは異なる楕円度, プラズマ位置を持つように作成した. Fig. 2 に TRIAM-1M トカマクのポロイダル断面図を示す. 真空容器の周りに40ヶ所, 各箇所に法線方向成分測定用と接線方向成分測定用の磁気コイルがあり, 合計80個の磁気コイルが配置されている.

3. 構造決定方法

目的にあったニューラルネットワークの最適な規模や構造を決定する一般的な方法は現在のところ存在しない. ネットワーク規模や構造の決定にあたっては, 多くの場合, 入力層ユニットの決定法と隠れ層ユニットの決定法は別々に考えられてきた. 問題によっては, 入力層ユニットすなわち入力変数があらかじめ明確に決まっている場合もあるが, 我々の対象とする問題では, 入力変数としてどの磁気コイル信号をとるかは必要な計算結果の精度を設定することによって決まるもので, 入力層ユニット数と隠れ層ユニット数は, 計算時間と計算精度のトレードオフで最適のものが決定されるべきものである.

計算時間はシグモイド関数を用いている隠れ層のユニット数に強く依存する. Fig. 3に隠れ層のユニット数と学習精度, 計算時間の関係の例を示す. これはプラズマ水平位置を計算するニューラルネットワークにおいて, 隠れ層の構造を変化させその時の精度および計算時間を示したものである. ユニット数と計算時間とはほぼ比例関係にあるが, ユニット数と精度との関係はユニット数が少なすぎる場合には急激に精度が悪くなっている. 精度を維持しつつ計算時間が最小となるようなユニット数を見つけるのが構造最適化であるが, このような関係を明らかにするためにはユニット数の異なる多数のネットワークの学習が必要なため長時間の計算が必要になる. 必要となる構造は使用する入力によって異なると考えられるので, 高速・高精度のニューラルネットワークを効率よく作成するためには構造最適化が重要な役割を果たす.

ここではまず, 入力層, 隠れ層ともに十分な数のユニットを用意して学習を行い, いずれの層についても冗長性判定に基づいて冗長なユニットを順次削除していった最適構造を確定する. 以下, この方法について詳しく記す.

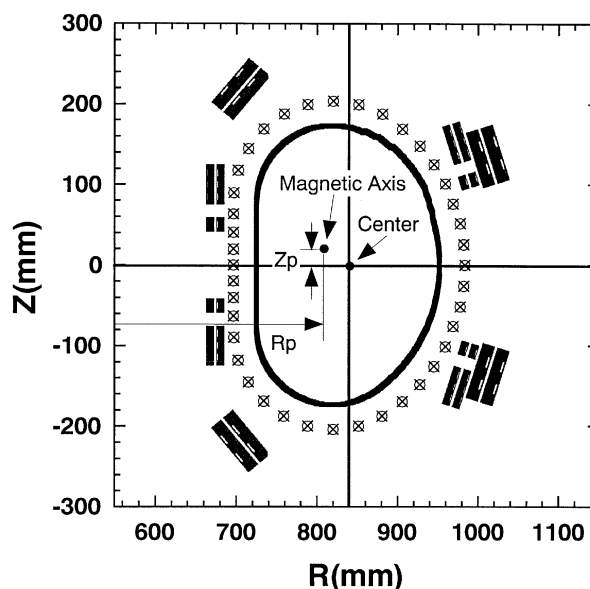


Fig. 2 Locations of magnetic coils around the vacuum vessel of TRIAM-1M. Symbols ○ and × indicate positions of magnetic coils for measuring normal component and tangential component, respectively. A plasma horizontal position (R_p) is major radius of the magnetic axis. A vertical position (Z_p) is defined as displacement of the magnetic axis from equatorial plane.

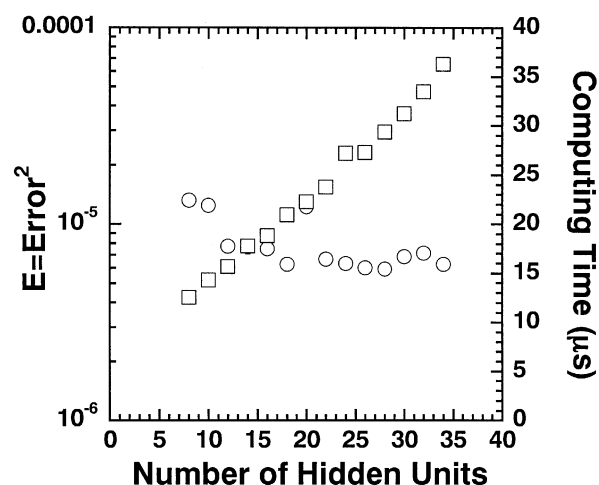


Fig. 3 Number of hidden units versus root mean square error (RMS error) and computing time for a neural network execution. A symbol ○ indicates RMS error. A symbol □ indicates computing time. Computing time is proportional to number of units. On the other hand, RMS error is constant. However when number of units is 10, RMS error becomes large. A long learning time is necessary for obtaining a relationship between number of units and RMS error.

3.1 従来の入力ユニット決定法

トカマクプラズマ計測、制御にニューラルネットワークを応用する際には今まで2つの入力変数決定法が用いられてきた。一つはある入力を0にしてその入力により与えられる情報が失われた場合の自乗誤差の劣化を比較し、入力変数間の重要性をランキングする[10, 13]。もう一つはネットワーク出力 Y をネットワーク入力 x_i で偏微分することで、ある入力の出力への寄与 $\partial Y / \partial x_i$ を求める方法である[9]。

これらの方法は非常に簡単で、ニューラルネットワークの性質をうまく利用したものであるが、入力の冗長性に関してはまったく考慮されていない。また選択基準の設定も難しい。したがって効率の良い入力ユニットの決定の際にはこれらの点を考慮する必要があると考えられる。

3.2 冗長性判定に基づく隠れ層ユニット決定法

隠れ層の構造決定には以下に示すような冗長性判定に基づく方法[11]を用いる。第 l 層の i 番目のユニット出力を $y_{l,i}^n$ 、第 $l+1$ 層の j 番目のユニット出力を $y_{l+1,j}^n$ 、重みを $w_{l,i,j}$ としたとき、それらの関係は

$$y_{l+1,j}^n = \sigma \left(\sum_{i=0} w_{l,i,j} y_{l,i}^n \right) \text{ となる。ここで } n \text{ は学習データの番号である。以後変数の右肩の } n \text{ は } n \text{ 番目の学習データに対する量を表すこととする。この時、全学習データにわたって (すべての } n \text{ に対して) } y_{l,i'}^n = \sum_{\substack{i=1 \\ i \neq i'}} \alpha_i^{i'} y_{l,i}^n \text{ なる関係が}$$

成り立てば、 $\sum_{i=0} w_{l,i,j} y_{l,i}^n = \sum_{\substack{i=0 \\ i \neq i'}} (w_{l,i,j} + w_{l,i',j} \alpha_i^{i'}) y_{l,i}^n$ となるので、第 l 層の i' 番目のユニットがなくても他のユニットにより表現できる。したがって、このユニットは冗長なユニットであるといえるので、削除できる。実際にはこのような関係が厳密に成り立つことはないため、係数 α は全学習データに対して最小自乗法で求め、ユニットを削除することで失われる情報 $\Delta_{l,i'}^n = y_{l,i'}^n - \sum_{\substack{i=1 \\ i \neq i'}} \alpha_i^{i'} y_{l,i}^n$ がネットワーク出力 Y に及ぼす影響 $\Delta Y_{l,i'}^n = (\partial Y / \partial y_{l,i'})^n \Delta_{l,i'}^n$ の絶対値の平均値によりそのユニットの冗長度を判定し、この数値が小さくなるようなユニットを冗長なユニットとして削除する。係数 α はユニット削除ごとに計算される。ここで $(\partial Y / \partial x_i)^n$ は n 番目の学習データに対する偏微分の値を表す。

この方法では、ある層において相互に一次従属の関係にあるユニットのうち最も冗長なものが削除される。全ネットワーク中で最も冗長なユニットを削除しているの

ではないので、複数の層に対して構造決定を行う際には注意する必要がある。文献[11]では多入力1出力の3層ネットワークの隠れ層のユニット数決定にこの方法を用いてその有効性を確認している。この方法は各ユニットの冗長性をネットワーク出力の変化で表現できるので、評価基準の設定が容易であるという特長を持つ。

3.3 冗長性判定による全ネットワーク構造決定方法

本研究では隠れ層の構造決定および入力の決定の双方について冗長性判定に基づく構造決定法[11]を用いる。入力層ユニットの決定のためには前述の冗長性判定において $l=1$ とすればよい。ネットワーク入力を x_i 、ネットワーク出力を Y とすると n 番目の学習データにおいて評価に用いる量は $\Delta Y_{1,i}^n = (\partial Y / \partial x_i)^n \Delta_{1,i}^n$ となる。この評価に用いる量はある入力の出力への寄与 $(\partial Y / \partial x_i)^n$ とその入力の独自性を示す量 $(\Delta_{1,i}^n)$ によって構成されており、前述した2つの入力決定法を包括するものである。特に複数のユニット間にまたがるような冗長性を考慮できるので、独立性の高い入力を選択できる入力決定法である。したがって磁気コイル信号のように本質的に各コイルの信号の間に相関があるような場合には有効であると考えられる。

高速化を目的として冗長性判定に基づく構造決定を隠れ層を複数持つネットワークの入力層と隠れ層の構造決定に用いる場合には、その手順（構造決定を行う層の順序）により得られる構造が異なると考えられる。このため、入力→隠れ層の順に構造決定を行うか、隠れ層→入力の順に行うか、あるいは全ネットワーク中で最も冗長なユニットを削除するのか、といった構造決定の手順をどう選ぶかが問題となる。そこで、本研究では構造決定の手順と得られたニューラルネットワークの構造とネットワーク性能（ネットワークによる写像の精度）とを比較することで、有効な適用方法を明らかにする。

3.4 ユニット削除判定基準

冗長性を $\Delta Y_{l,i}^n = (\partial Y / \partial y_{l,i'})^n \Delta_{l,i'}^n$ で表される“ネットワーク出力への影響”により評価し構造決定を行うが、ネットワーク性能の評価は自乗誤差によって行う。したがって“ネットワーク出力への影響”とそのユニットを削除することによる自乗誤差の劣化との関係は削除基準の設定について重要な情報を与えると考えられる。

ここで、“ネットワーク出力への影響”と自乗誤差の劣化との関係について考察する。 Y_l^n を教師信号、 Y_0^n をユニット削除前のネットワーク出力、 Y^n を削除後のネットワーク出力とする。この時それぞれの自乗誤差は次式で表される。

$$E_0 = \sum (Y_0^n - Y_t^n)^2 / N, \quad E = \sum (Y_0^n - Y_t^n)^2 / N \quad (1)$$

ここで $\sum$ はすべての学習データに関して和をとることを意味し, N は学習データの総数を表す. 各学習データに対する実際のネットワーク出力の変化 ΔY^n を以下のように定義するとユニット削除後の自乗誤差は次式で表せる.

$$\Delta Y^n = Y^n - Y_0^n \quad (2)$$

$$\begin{aligned} E &= \sum (Y_0^n - Y_t^n + \Delta Y^n)^2 / N \\ &= \sum (Y_0^n - Y_t^n)^2 / N + 2 \sum \Delta Y^n (Y_0^n - Y_t^n) / N \\ &\quad + \sum (\Delta Y^n)^2 / N \\ &= E_0 + \sum (\Delta Y^n)^2 / N + 2 \sum \Delta Y^n (Y_0^n - Y_t^n) / N \quad (3) \end{aligned}$$

ユニットを削除したことによるネットワーク出力への影響 $\Delta Y_{l,i}^n = (\partial Y^n / \partial y_{l,i}) \Delta y_{l,i}^n$ をその絶対値の平均値 $\langle |\Delta Y| \rangle \equiv \sum \Delta Y_{l,i}^n / N$ で規格化したものを $\Delta \tilde{Y}_{l,i}^n$ とすれば, 上式より, ネットワーク性能の劣化量 ΔE は次のように表される.

$$\begin{aligned} \Delta E &= \langle |\Delta Y| \rangle^2 \frac{\sum (\Delta \tilde{Y}_{l,i}^n)^2}{N} + \frac{2 \sum \Delta Y_{l,i}^n (Y_0^n - Y_t^n)}{N} \\ &= a_{l,i} \langle |\Delta Y| \rangle^2 + \frac{2 \sum \Delta Y_{l,i}^n (Y_0^n - Y_t^n)}{N} \quad (4) \end{aligned}$$

ここで $a_{l,i} = \sum (\Delta \tilde{Y}_{l,i}^n)^2 / N \geq 1$ ですべてのデータに対して ΔY が等しいとき等号が成り立つ.

十分学習された精度の良いネットワークにおいてすべての学習データについて冗長性が認められたユニットを削除する場合は次式が満足される.

$$\left| \sum \Delta Y_{l,i}^n (Y_0^n - Y_t^n) \right| < a_{l,i} \langle |\Delta Y| \rangle^2 \quad (5)$$

$$a_{l,i} \sim 1 \quad (6)$$

したがって

$$\Delta E \sim \langle |\Delta Y| \rangle^2 \quad (7)$$

ネットワーク性能を十分高く保ったままユニット削除を行う限り基本的には(7)式が成り立つ. このような状況に置いて $\langle |\Delta Y| \rangle$ の許容最大値 (削除基準) を b とすれば ΔE の許容最大値は b^2 となり, 下記の2つの条件のいずれによってユニット削除を行っても同じ結果が期待できる.

$$\langle |\Delta Y| \rangle < b \quad (8)$$

$$\Delta E < b^2 \quad (9)$$

しかし, 現実的なデータ処理においては(5), (6)式からはずれた条件の下での解析が必要となる. 代表的な場合として次の2例が考えられる.

例1

$$\left| \sum \Delta Y_{l,i}^n (Y_0^n - Y_t^n) \right| \ll a_{l,i} \langle |\Delta Y| \rangle^2 \quad (10)$$

$$a_{l,i} > 1 \quad (11)$$

この場合, ΔE と $\langle |\Delta Y| \rangle$ との関係は次式で表される.

$$\Delta E > \langle |\Delta Y| \rangle^2 \quad (12)$$

このような状況では ΔY 基準((8)式)でユニットの削除を行うと, $\Delta E > b^2$ となり過剰なユニット削除が行われるので, ΔE 基準によるユニット削除を行う必要がある.

例2

$$\left| \sum \Delta Y_{l,i}^n (Y_0^n - Y_t^n) \right| \sim a_{l,i} \langle |\Delta Y| \rangle^2 \quad (13)$$

$$\sum \Delta Y_{l,i}^n (Y_0^n - Y_t^n) < 0 \quad (14)$$

$$a_{l,i} \sim 1 \quad (15)$$

この場合, ΔE と $\langle |\Delta Y| \rangle$ との関係は次のようになる.

$$\Delta E < \langle |\Delta Y| \rangle^2 \quad (16)$$

このような状況では ΔY 基準((8)式)でユニット削除を行うと, $\Delta E < b^2$ となるのでネットワーク性能を落とさずにさらにユニットを削除する余地が残されている. このようにユニット削除基準値を同一にしても状況によって $\langle |\Delta Y| \rangle$ と ΔE のいずれを評価量とすべきかは断定できないので両方の削除基準を用いるべきである.

上記議論の結論について数値実験的に確かめるためにユニット削除中のネットワークの各ユニットに対する“ネットワーク出力への影響の絶対値の平均値” $\langle |\Delta Y| \rangle$ と“ユニット削除による自乗誤差の劣化” ΔE との関係を Fig. 4 に示す. この結果から $\langle |\Delta Y| \rangle$ と ΔE との関係は $\Delta E = \langle |\Delta Y| \rangle^2$ 程度であることがわかる. この関係は $\langle |\Delta Y| \rangle$ が小さい場合には良く成り立っているが, $\langle |\Delta Y| \rangle$ が大きくなるとずれが大きくなっている. 高速化を目的とする大規模なユニット削除を行う場合には $\langle |\Delta Y| \rangle$ に対

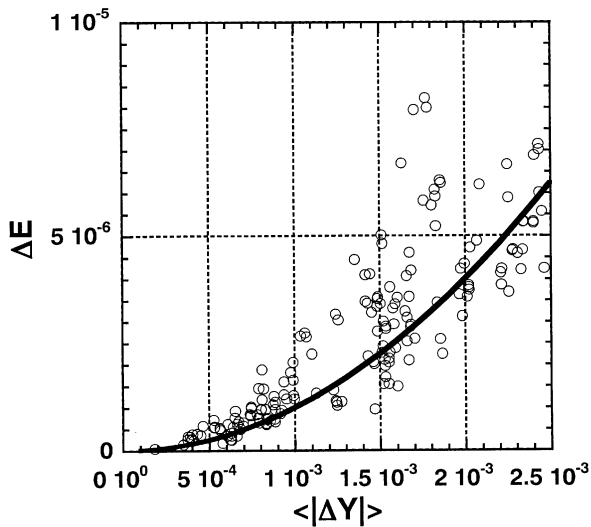


Fig. 4 Degradation of neural network performance (ΔE) by removing a unit versus effect of a unit removal to output ($\langle |\Delta Y| \rangle$). A solid line indicates $\Delta E = \langle |\Delta Y| \rangle^2$. Calculated relationship between ΔE and $\langle |\Delta Y| \rangle$ is indicated by symbol $\circ$. If redundancy test based on evaluation of $\langle |\Delta Y| \rangle$ is employed, it may occur that removed units bring large degradation of network performance.

する削除基準を大きく設定する必要がある。そのためネットワークの性能に、より大きな影響を持つユニットを削除する可能性がある。入力決定の際にはネットワーク性能に対して寄与の大きいものを削除することとなるので問題となる。また、隠れ層の構造決定でも構造決定中のネットワークの変化が大きければ有効な冗長性判定ができない可能性がある。したがって従来使用されてきた $\langle |\Delta Y| \rangle$ を評価量とする構造決定だけでは不十分であるので、前述の議論の結論に従って ΔE を評価量とする構造決定も同時に行うことを提案する。

4. トカマク実験に用いるニューラルネットワークの構造決定

4.1 入力ユニットの決定

冗長性判定により入力ユニットの決定を行う場合には前章において述べたように冗長性の評価に“ネットワーク出力への影響の絶対値の平均値” $\langle |\Delta Y| \rangle$ を用いる手法と“ユニット削除による自乗誤差の劣化” ΔE を用いる手法の2つがある。Fig. 5に入力ユニット削除中のネットワーク性能の劣化、Fig. 6に再学習の際の収束状況、Table 1に1万回の再学習後のネットワーク性能を示す。Table 1に記してある4つの数字のつながり(16-17-17-1等)はニューラルネットワークの構造であり、それぞれ入力層—第1隠れ層—第2隠れ層—出力層の構成

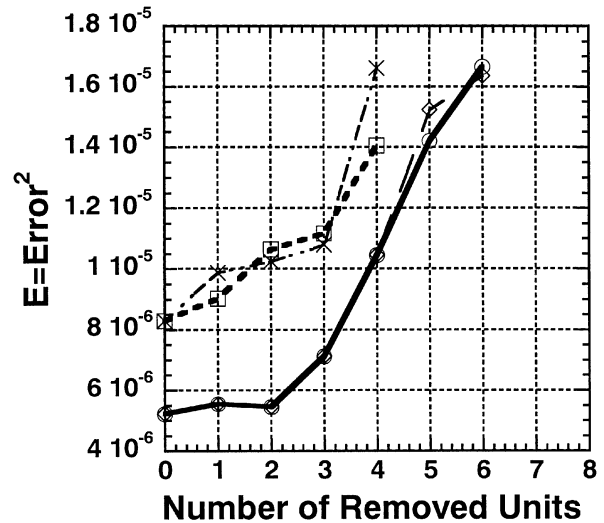


Fig. 5 Degradations of the network performance during removing units in input layer. A symbol $\circ$ indicates a case of structure optimization of a neural network for R_p calculation by unit removing based on ΔE . A symbol $\square$ indicates a case of Z_p and ΔE . A symbol $\diamond$ indicates a case of R_p and $\langle |\Delta Y| \rangle$. A symbol $\times$ indicates a case of Z_p and $\langle |\Delta Y| \rangle$. Degradations in case of ΔE are a little more slowly than those in case of $\langle |\Delta Y| \rangle$.

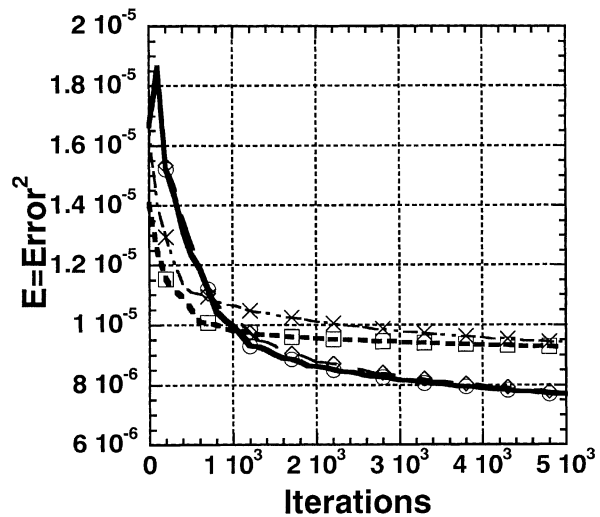


Fig. 6 Convergence curves in relearning process after determination of units in input layer. A symbol $\circ$ indicates a case of structure optimization of a neural network for R_p calculation by unit removing based on ΔE . A symbol $\square$ indicates a case of Z_p and ΔE . A symbol $\diamond$ indicates a case of R_p and $\langle |\Delta Y| \rangle$. A symbol $\times$ indicates a case of Z_p and $\langle |\Delta Y| \rangle$. Convergences in case of ΔE is a little faster than those in case of $\langle |\Delta Y| \rangle$. It is suggested that the unit removing based on ΔE give good weights at start of relearning.

Table 1 Neural network performance after determination of units in input layer and relearning.

Rp		Root Mean Square Error (mm)	
Procedure	Structure	Learning	Test
Original	16-17-17-1	0.4296034	0.4766308
Input : ΔE	10-17-17-1	0.5002083	0.5017930
Input : $\langle \Delta Y \rangle$	10-17-17-1	0.5021127	0.5065425

Zp		Root Mean Square Error (mm)	
Procedure	Structure	Learning	Test
Original	16-17-17-1	0.5376626	1.8969560
Input : ΔE	12-17-17-1	0.5605577	1.7151140
Input : $\langle \Delta Y \rangle$	12-17-17-1	0.5608377	1.7098925

ユニット数を表す。Root Mean Square Error はネットワーク出力の平均自乗誤差を表しており、Learning は学習用データに対する出力誤差、Test は検証用データに対する出力誤差で汎化能力を表している。この際、削除判定基準は $\langle |\Delta Y| \rangle$ を評価に用いる場合には 2×10^{-3} 、 ΔE を評価に用いる場合には 4×10^{-6} とし、 $\Delta E = \langle |\Delta Y| \rangle^2$ を満たすように設定している。構造決定前のネットワークの自乗誤差は $5 \sim 8 \times 10^{-6}$ 程度である。削除判定基準は1つのユニットを削減した場合に最大でどれほどのニューラルネットワークの劣化を許容するかを表す量である。1つのユニットを削除した場合にオリジナルのネットワークの誤差が倍になってしまうような場合にはそのユニットは冗長であるとは言えない。そこで削除判定基

準を構造決定前のネットワークの誤差程度の値に設定した。

Fig. 5 から $\langle |\Delta Y| \rangle$ を評価に用いた場合の方が削除中の性能劣化がわずかに大きいことが、また、Fig. 6 から ΔE を評価に用いた場合の方が再学習の際の収束がわずかに速いことが判明した。しかし、選択された入力がほぼ同一のため再学習後の性能は同等のものになっている。学習の際の収束が速いということは学習開始時の重みの初期値が良いことを示しており、構造決定の有効性を示している。

4.2 隠れ層の構造決定

冗長性判定に基づく2層の隠れ層の構造決定を行う。この場合も冗長性の評価に“ネットワーク出力への影響の絶対値の平均値” $\langle |\Delta Y| \rangle$ を用いる手法と“ユニット削除による自乗誤差の劣化” ΔE を用いる手法の2通りがある。また、隠れ層が2層あるので第1隠れ層 (h1 と略記)、第2隠れ層 (h2 と略記) の順に決定するか、h2→h1 の順に決定するか、隠れ層の構成ユニットの中で最も冗長なユニットを削除していく (all と略記) か、という3通りの手順がある。Fig. 7に隠れ層のユニット削減中のネットワーク性能の劣化、Fig. 8に再学習の際の収束状況、Table 2に1万回の再学習後のネットワーク性能を示す。

Fig. 7から、隠れ層の構成ユニットの中で最も冗長なユニットを削除していく場合には各層毎に削除する場合と比べ、ユニット削除中のネットワーク性能の劣化が小さいことがわかる。また、Fig. 8の再学習の状況、Table 2

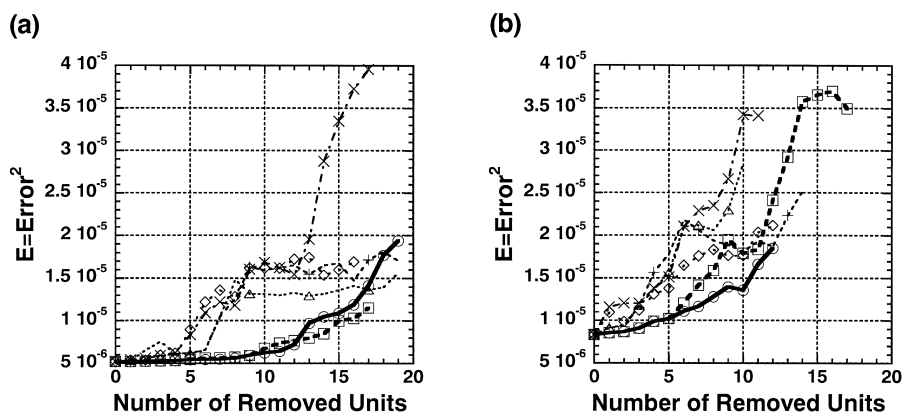


Fig. 7 Degradations of the network performance during removing units in hidden layers. (a) is a case of a neural network for R_p calculation. (b) is a case of Z_p . Symbols $\circ$ and $\square$ are cases of iteration of most redundancy unit removing based on ΔE and $\langle |\Delta Y| \rangle$, respectively. A symbol $\diamond$ indicates a case that unit removing based on ΔE is executed in the order of the first hidden layer to the second hidden layer (h1→h2). A symbol $\times$ indicates a case of $\langle |\Delta Y| \rangle$ and h1→h2. A symbol $+$ indicates a case of ΔE and h2→h1. A symbol $\triangle$ indicates a case of $\langle |\Delta Y| \rangle$ and h2→h1. Degradations in the case of removing most redundancy unit are slower than the other cases. It is interpreted that the redundancy test during the iteration is successful.

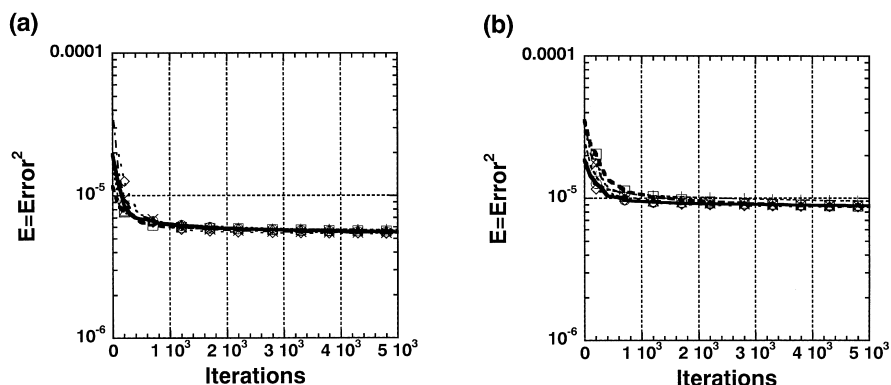


Fig. 8 Convergence curves in relearning process after determination of structure of hidden layers. (a) is a case of a neural network for R_p calculation. (b) is a case of Z_p . symbols $\circ$ and $\square$ are cases of iteration of most redundancy unit removing based on ΔE and $\langle |\Delta Y| \rangle$, respectively. A symbol $\diamond$ indicates a case that unit removing based on ΔE is executed in the order of the first hidden layer to the second hidden layer ($h1 \rightarrow h2$). A symbol $\times$ indicates a case of $\langle |\Delta Y| \rangle$ and $h1 \rightarrow h2$. A symbol $+$ indicates a case of ΔE and $h2 \rightarrow h1$. A symbol $\triangle$ indicates a case of $\langle |\Delta Y| \rangle$ and $h2 \rightarrow h1$. Convergence speed is the same order in all procedures.

Table 2 Neural network performance after determination of structure of hidden layers and relearning.

R_p		Root Mean Square Error (mm)	
Procedure	Structure	Learning	Test
Original	16-17-17-1	0.4296034	0.4766308
all : ΔE	16-7-8-1	0.4365858	0.4556863
all : $\langle \Delta Y \rangle$	16-7-10-1	0.4370852	0.4547285
$h1 \rightarrow h2$: ΔE	16-8-10-1	0.4319556	0.4482607
$h1 \rightarrow h2$: $\langle \Delta Y \rangle$	16-8-9-1	0.4322337	0.4547600
$h2 \rightarrow h1$: ΔE	16-7-8-1	0.4462971	0.4571013
$h2 \rightarrow h1$: $\langle \Delta Y \rangle$	16-7-8-1	0.4418572	0.4517136

Z_p		Root Mean Square Error (mm)	
Procedure	Structure	Learning	Test
Original	16-17-17-1	0.5376626	1.8969560
all : ΔE	16-11-11-1	0.5395267	1.6076135
all : $\langle \Delta Y \rangle$	16-8-9-1	0.5397043	1.6282543
$h1 \rightarrow h2$: ΔE	16-12-10-1	0.5450730	1.6700130
$h1 \rightarrow h2$: $\langle \Delta Y \rangle$	16-10-13-1	0.5620318	1.6093027
$h2 \rightarrow h1$: ΔE	16-9-11-1	0.5439164	1.6485168
$h2 \rightarrow h1$: $\langle \Delta Y \rangle$	16-12-12-1	0.5464311	1.6664073

の再学習結果には有意な差は見られない。しかし用いる手法により削除できるユニット数に違いが見られる。その中で、すべてのユニットの中で最も冗長なユニットを削除する方法により、構造決定前のネットワークと同程度の性能を有しながら最も構成ユニット数の少ないネットワークが構築されている。

冗長性を評価する量として ΔE と $\langle |\Delta Y| \rangle$ どちらを選択するかについて、有効性の確認された“すべてのユニッ

トの中で最も冗長なユニットを削除する方法”において検討を行う。 R_p の場合は ΔE により冗長性を評価した方が、 Z_p の場合には $\langle |\Delta Y| \rangle$ により評価した方が少ない構成ユニット数のネットワークを構築することができた。通常は ΔE により評価する方法でのユニット削除中のネットワークの性能劣化は $\langle |\Delta Y| \rangle$ による方法より小さいか同程度である。しかし $\langle |\Delta Y| \rangle$ により評価した方が少ない構成ユニット数のネットワークを構築できた Z_p の場合では、ユニット削除の過程でネットワーク性能が劣化しない領域(削除したユニット数が15程度のところ)があり、この影響でより多くのユニットを削除できたのではないかと考えられる。そこで Z_p において削除基準を大きくとりさらに多くのユニットを削除したときに、 ΔE により評価する方法により $\langle |\Delta Y| \rangle$ による方法と同様のネットワークが構築できれば冗長性の評価には ΔE を用いればよいと結論できる。

Fig. 9 に削除基準を大きくとった場合の構造決定中のネットワーク性能の劣化を、Table 3 に1万回の再学習後のネットワーク性能を示す。 R_p の場合には構造決定中の性能劣化にほとんど差がないことがわかる。しかし削除したユニットが22個になるあたりから $\langle |\Delta Y| \rangle$ により評価する方法の劣化が大きくなっていることがわかる。最終的には ΔE により評価する方法の方が性能劣化は大きくなったが、 $\langle |\Delta Y| \rangle$ により評価する方法に比べ構成ユニット数が少なく精度は高いという結果が得られた。一方 Z_p に関しては両者の間に明らかな差が見られる。ユニット削除の初期段階(～15)までは ΔE により評価する方法の方が明らかに劣化が小さい。しかし $\langle |\Delta Y| \rangle$ による評価では削除したユニットが15個程度のところでユニット

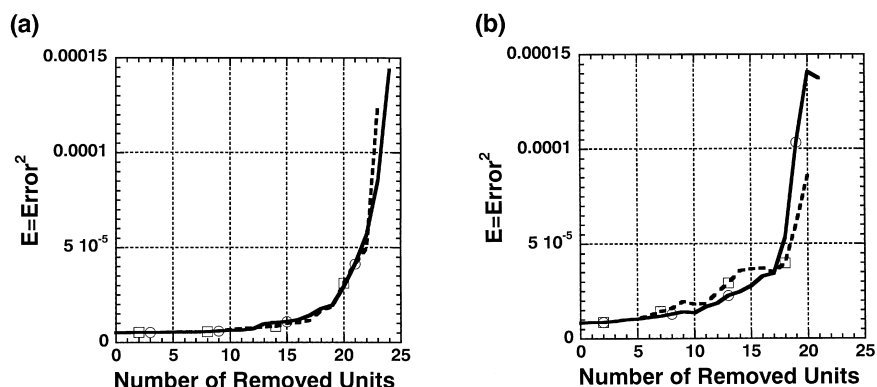


Fig. 9 Degradations of the network performance during strong removal of units of hidden layers. (a) is a case of a neural network for R_p calculation. (b) is a case of Z_p . A symbol ○ indicates a case of unit removing based on ΔE . A symbol □ indicates a case of $\langle \Delta Y \rangle$. Generally degradations in case of ΔE are a little more slowly than those in case of $\langle \Delta Y \rangle$, for example, in all case in (a) and in the case that the number of reduced units is less than 15 in (b). However, it may occur that the degradation in case of $\langle \Delta Y \rangle$ is smaller than the one in case of ΔE , for example the number of reduced unit is about 15 in (b).

Table 3 Neural network performance after strong removal of units in hidden layers and relearning.

		Criterion of removal	Number of units			Error after 10000 relearning
			1st hidden layer	2nd hidden layer	Hidden layers	
Rp	ΔE	4.0000e-06	7	8	15	5.3866e-06
		3.6000e-05	5	6	11	6.1748e-06
		6.4000e-05	5	5	10	6.6900e-06
	$\langle \Delta Y \rangle$	0.0020000	7	10	17	5.3953e-06
		0.0060000	6	6	12	6.6894e-06
		0.0080000	6	5	11	7.2056e-06
Zp	ΔE	4.0000e-06	11	11	22	8.5693e-06
		3.6000e-05	8	8	16	9.5068e-06
		6.4000e-05	6	7	13	1.0425e-05
	$\langle \Delta Y \rangle$	0.0020000	8	9	17	8.3571e-06
		0.0060000	6	8	14	8.4295e-06
		0.0080000	6	8	14	8.4295e-06

削除による性能劣化が見られない。その後 ΔE による構造決定では急激に性能劣化が起こったのに比べ、 $\langle \Delta Y \rangle$ による構造決定ではこのような現象は見られなかった。再学習後得られたネットワーク性能も $\langle \Delta Y \rangle$ によるものの方が少ない構成ユニット数であるにもかかわらず良いことがわかった。

通常は各学習データに対する出力誤差（自乗誤差ではない）は平均的に 0 に近い値をとるため 3.4 節の議論における例 1 のように ΔE による評価が有効である。学習により自乗誤差を小さくしていく過程ではこの状況になるが、構造最適化を施しユニットを削除した場合には、ネットワークへの影響を最小限にするようにユニットの削除を行っているにもかかわらず、1つのユニットの削除による影響により出力誤差が平均的に 0 でなくなるという 3.4 節の議論における例 2 のような状況もあり得

る。 $\langle \Delta Y \rangle$ による評価の方が有効であるという場合にはこのような状況になっていると考えられる。

以上のことから学習が十分に行われその状態が維持されている場合には ΔE による評価が有効であるが、ユニット削除によりその状態が崩れてしまった場合には $\langle \Delta Y \rangle$ による評価が有効な場合もあることがわかった。したがってより少ない構成ユニット数で高精度なネットワークを得るためには $\langle \Delta Y \rangle$ を評価することによる構造決定と ΔE を評価することによる構造決定との両方を行う必要がある。

4.3 入力・隠れ層の構造の同時決定

冗長性判定に基づく構造決定を用いて入力層と隠れ層の構造決定を同時に行う。一般に必要な隠れ層の構造は入出力関係に依存するため、ニューラルネットワークの構造決定は入力を決定、その入力を用いたニューラルネットワークの構築、隠れ層の構造決定・再学習という手順で行うのが妥当であると考えられる。しかし、冗長性判定による構造決定ではネットワーク出力への影響が小さいユニットを削除していくので、構造の変更によるネットワークへの影響を最小限にすることが可能であるため入力層と隠れ層の構造決定を同時に行うことが可能である。これにより大規模なネットワークから高速なニューラルネットワークを速やかに構築することができる。

入力層と隠れ層の構造の同時決定をする際の手順として、i) すべてのユニットの中で最も冗長なユニットを削除する、ii) 入力層ユニット決定後、隠れ層構造決定を行う、iii) 隠れ層構造決定後、入力層ユニットを決定するという 3 通りが考えられる。またそのほかに入力層ユニッ

Table 4 Neural network performance after determination of units in input layer and structure of hidden layers, and relearning.

Rp		Root Mean Square Error (mm)	
Procedure	Structure	Learning	Test
Original	16-17-17-1	0.4296034	0.4766308
all: ΔE	10-8-7-1	0.4799113	0.5587404
all: $\langle \Delta Y \rangle$	10-7-10-1	0.4963079	0.4841195
in $\rightarrow$ h1 $\rightarrow$ h2: ΔE	10-7-10-1	0.5031978	0.4876058
in $\rightarrow$ h1 $\rightarrow$ h2: $\langle \Delta Y \rangle$	10-8-9-1	0.4922093	0.4867793
in $\rightarrow$ h2 $\rightarrow$ h1: ΔE	10-9-11-1	0.5118521	0.5024603
in $\rightarrow$ h2 $\rightarrow$ h1: $\langle \Delta Y \rangle$	10-9-10-1	0.5042712	0.4887039
h1 $\rightarrow$ h2 $\rightarrow$ in: ΔE	9-8-10-1	0.5358885	0.4817883
h1 $\rightarrow$ h2 $\rightarrow$ in: $\langle \Delta Y \rangle$	10-8-9-1	0.4846339	0.7745502
h2 $\rightarrow$ h1 $\rightarrow$ in: ΔE	9-7-8-1	0.5678079	0.4795553
h2 $\rightarrow$ h1 $\rightarrow$ in: $\langle \Delta Y \rangle$	10-7-8-1	0.5503250	0.5208498
in $\rightarrow$ relearn $\rightarrow$ h1 $\rightarrow$ h2: ΔE	10-9-10-1	0.4855596	0.4909198
in $\rightarrow$ relearn $\rightarrow$ h1 $\rightarrow$ h2: $\langle \Delta Y \rangle$	10-8-10-1	0.5057433	0.4962565
in $\rightarrow$ relearn $\rightarrow$ h2 $\rightarrow$ h1: ΔE	10-8-10-1	0.5079587	0.5076864
in $\rightarrow$ relearn $\rightarrow$ h2 $\rightarrow$ h1: $\langle \Delta Y \rangle$	10-8-10-1	0.5342128	0.5242596
h1 $\rightarrow$ h2 $\rightarrow$ relearn $\rightarrow$ in: ΔE	11-8-10-1	0.5444189	0.4840986
h1 $\rightarrow$ h2 $\rightarrow$ relearn $\rightarrow$ in: $\langle \Delta Y \rangle$	11-8-9-1	0.4815561	0.4788421
h2 $\rightarrow$ h1 $\rightarrow$ relearn $\rightarrow$ in: ΔE	10-7-8-1	0.4918585	0.4953406
h2 $\rightarrow$ h1 $\rightarrow$ relearn $\rightarrow$ in: $\langle \Delta Y \rangle$	10-7-8-1	0.5089823	0.4676608

Zp		Root Mean Square Error (mm)	
Procedure	Structure	Learning	Test
Original	16-17-17-1	0.5376626	1.8969560
all: ΔE	12-9-9-1	0.5655978	1.6613067
all: $\langle \Delta Y \rangle$	12-9-8-1	0.5815097	1.7145441
in $\rightarrow$ h1 $\rightarrow$ h2: ΔE	12-9-10-1	0.5734202	1.5522112
in $\rightarrow$ h1 $\rightarrow$ h2: $\langle \Delta Y \rangle$	12-11-10-1	0.5865166	1.7241554
in $\rightarrow$ h2 $\rightarrow$ h1: ΔE	12-9-11-1	0.5568700	1.6211042
in $\rightarrow$ h2 $\rightarrow$ h1: $\langle \Delta Y \rangle$	12-11-12-1	0.5555324	1.7108795
h1 $\rightarrow$ h2 $\rightarrow$ in: ΔE	11-12-10-1	0.5806588	1.6332292
h1 $\rightarrow$ h2 $\rightarrow$ in: $\langle \Delta Y \rangle$	12-10-13-1	0.5814444	1.8041478
h2 $\rightarrow$ h1 $\rightarrow$ in: ΔE	12-9-11-1	0.5930476	1.6167956
h2 $\rightarrow$ h1 $\rightarrow$ in: $\langle \Delta Y \rangle$	11-12-12-1	0.5655442	1.6590712
in $\rightarrow$ relearn $\rightarrow$ h1 $\rightarrow$ h2: ΔE	12-11-9-1	0.5461450	1.6849992
in $\rightarrow$ relearn $\rightarrow$ h1 $\rightarrow$ h2: $\langle \Delta Y \rangle$	12-9-10-1	0.5671693	1.7094979
in $\rightarrow$ relearn $\rightarrow$ h2 $\rightarrow$ h1: ΔE	12-10-11-1	0.5706948	1.7040214
in $\rightarrow$ relearn $\rightarrow$ h2 $\rightarrow$ h1: $\langle \Delta Y \rangle$	12-10-11-1	0.5393515	1.7050831
h1 $\rightarrow$ h2 $\rightarrow$ relearn $\rightarrow$ in: ΔE	12-12-10-1	0.5423187	1.6947922
h1 $\rightarrow$ h2 $\rightarrow$ relearn $\rightarrow$ in: $\langle \Delta Y \rangle$	12-10-13-1	0.5458276	1.7740074
h2 $\rightarrow$ h1 $\rightarrow$ relearn $\rightarrow$ in: ΔE	12-9-11-1	0.5541536	1.7375961
h2 $\rightarrow$ h1 $\rightarrow$ relearn $\rightarrow$ in: $\langle \Delta Y \rangle$	11-12-12-1	0.6259738	1.8544857

トあるいは構造を決定しニューラルネットワークを構築した後、構造あるいは入力層ユニットを決めるというiv) 入力層ユニット決定後再学習を行い、隠れ層構造決定を行う、v) 隠れ層構造決定後再学習を行い、入力層ユニットを決定する、という2通りの方法も考えられる。これはii), iii)の場合において構造決定中にネットワークの変化が大きくなり、効果的な冗長性判定を行うことができなくなるという可能性を考慮したものである。また、ii)～v)においてはさらに第1隠れ層から決定する場合と第2隠れ層から決定する場合の2通りがある。

Table 4にそれぞれの手法で構造決定を行い1万回の再学習を行ったニューラルネットワークの性能を示す。relearnは再学習のタイミングを表す。得られたネットワーク性能に有意な差はないが、得られた構造は手法により異なり、すべてのユニットの中で最も冗長なユニットを削除する方法が比較的多くのユニット削除を実現し

た。これはFig. 10に示すように、構造決定中のネットワーク性能の劣化が比較的小さく有効な冗長性判定ができたものであると考えられる。ここでも Z_p の場合は $\langle |\Delta Y| \rangle$ を用いる手法、 R_p の場合は ΔE を用いる手法の方が少ない構成ユニット数のネットワークを構築しており、隠れ層の構造決定の場合と同様の傾向が見られた。またここで得られた入力および隠れ層の構造は、入力のみを決定した場合の入力、隠れ層の構造のみを決定した場合の構成ユニット数と同様の結果が得られており、入力層と隠れ層の同時決定の有効性が明らかになった。

5. まとめ

入力層と隠れ層の構造決定を冗長性判定による構造決定法により行う際に有効な手法について研究した。ユニット削除判定基準の設定は“失われる情報量の絶対値の平均値”を $\langle |\Delta Y| \rangle$ 、“ユニット削除による自乗誤差の劣

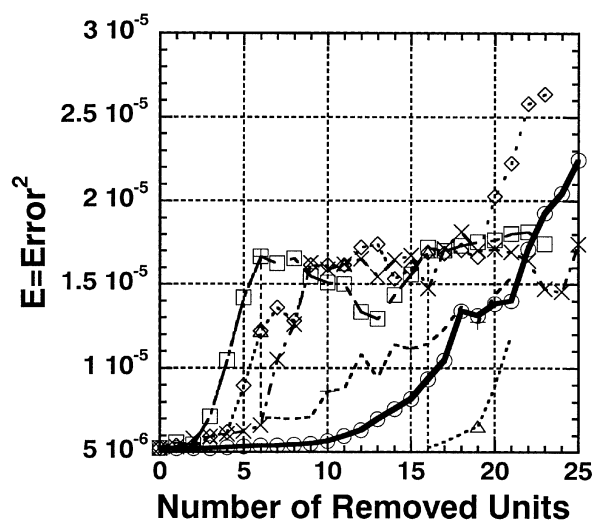


Fig. 10 Degradations of the neural network performance during removing units in input layer and hidden layers. This figure shows the case of structure optimization of a neural network for R_p calculation by unit removing based on ΔE , for example. A symbol $\circ$ indicates cases of iteration of most redundancy unit removing based on ΔE . A symbol $\square$ indicates a case that unit removing based on ΔE is executed in the order of the input layer, the first hidden layer to the second hidden layer. (in $\rightarrow$ h1 $\rightarrow$ h2). A symbol $\diamond$ indicates a case of h1 $\rightarrow$ h2 $\rightarrow$ in. A symbol $\times$ indicates a case of h2 $\rightarrow$ h1 $\rightarrow$ in. A symbol $\times$ indicates a case of in $\rightarrow$ h1 $\rightarrow$ h2 with relearning after determination of input layer. A symbol $\times$ indicates a case of h1 $\rightarrow$ h2 $\rightarrow$ in with relearning after determination of hidden layers. The degradation in case of removing most redundancy unit is more slowly than the other cases.

化”を ΔE とすると, $\Delta E = \langle |\Delta Y| \rangle^2$ を目安にして設定できる. 削除判定基準はオリジナルのネットワークの誤差と同程度に設定した. それよりも大きな値に設定した場合には構造決定後の精度は明らかに悪くなっていることが Table 3 よりわかる. このことから今回行った削除判定基準の設定は妥当であったことがわかる. 高速化を目的とする大規模なユニット削除を行う際には削除基準を大きくとる必要がある. この場合には $\langle |\Delta Y| \rangle$ を評価することによる構造決定では不十分であるため, ΔE を評価する

ことによる構造決定も必要となる. この両者は3.4節の議論にあったようにどちらかが優れているということはない. したがって高速・高精度のネットワークを得るためには両者を併用する必要がある. また構造決定を行う手順により得られた精度に大きな差は見られなかったが, 得られた構造には違いが見られた. すべてのユニットの中で最も冗長なユニットを削除する方法により, より多くのユニットの削除に成功した. これは構造決定中のネットワーク性能の劣化が他の方法に比べ小さいためである. この方法により入力層および隠れ層の構造を同時決定した場合には, 入力のみを決定する場合に得られた入力層および隠れ層の構造のみを決定する場合に得られた構造と類似しており, 入力層と隠れ層の構造の同時決定は有効であることが判明した. これにより大規模なネットワークからリアルタイムで使用可能な規模のネットワークの作成に要する時間を大幅に短縮できる.

しかしこの方法では構造決定・再学習後のネットワーク性能を予測することは原理的にはできない. したがって要求する性能を有する最小構成のネットワークを得るためには削除基準を変えていくつかのニューラルネットワークを構築する必要がある.

参考文献

- [1] J.B. Lister *et al.*, Nucl. Fusion **31**, 1291 (1991).
- [2] T. Wijnands *et al.*, Nucl. Fusion **36**, 1405 (1996).
- [3] R. Yoshino *et al.*, Fusion Technol. **30**, 237 (1996).
- [4] C.G. Windsor *et al.*, Fusion Technol. **32**, 416 (1997).
- [5] D. Wroblewski, Rev. Sci. Instrum. **68**, 1281 (1997).
- [6] J. Santos *et al.*, Rev. Sci. Instrum. **70**, 521 (1999).
- [7] J.V. Hernandez *et al.*, Nucl. Fusion **36**, 1009 (1996).
- [8] A. Vannucci *et al.*, Nucl. Fusion **39**, 255 (1999).
- [9] D. Wroblewski *et al.*, Nucl. Fusion **37**, 725 (1997).
- [10] R. Albanese *et al.*, Fusion Technol. **30**, 219 (1996).
- [11] 村田純一, 藤井洋司, 池田謙介, 平澤宏太郎, 相良節夫: 計測自動制御学会論文 **31**, 236 (1995).
- [12] D.E. Rumelhart *et al.*, Nature **323**, 533 (1986).
- [13] E. Coccorese *et al.*, Nucl. Fusion **34**, 1349 (1994).