論　文

階層型ニューラルネットの2変数関数近似能力の比較

野中　和明[†a)]　　田中　文英[†b)]　　森田　昌彦[†c)]

Empirical Comparison of Feedforward Neural Networks on Two-Variable Function Approximation

Kazuaki NONAKA[†a)], Fumihide TANAKA[†b)], and Masahiko MORITA[†c)]

あらまし　階層型ニューラルネットは，与えられたサンプルから入出力関係を推定する関数近似器としてよく用いられる．なかでも多層パーセプトロン（MLP）は，理論上任意の連続関数を精度よく表現可能であることが知られているが，実際にサンプルを学習して多変数関数を近似する能力には大きな疑問がある．本研究では，MLP および同様に万能な関数表現能力をもつ放射状基底関数ネットワーク（RBFN），並列パーセプトロン（PP），選択的不感化ニューラルネット（SDNN）について，関数近似器としての実際的能力の違いを明らかにするために，やや複雑な2変数関数を用いて評価実験を行った．その結果，学習能力や汎化能力を含めた近似能力は SDNN が最も高く，計算コストなど実用性の点でも SDNN が優れていた．また，このような高性能には，アナログの入力値を多数の2値素子によって表現するパターンコーディングと，複数のパターン表現を統合する選択的不感化が共に大きく貢献していることがわかった．この結果は SDNN が MLP などにはない高い有用性をもつことを示しており，これによってニューラルネットの応用範囲が大きく広がる可能性がある．

キーワード　神経回路　機械学習　汎化能力　近似誤差　分散表現

1. まえがき

多層パーセプトロン（Multi-Layer Perceptron, 以下 MLP）に代表される階層型ニューラルネットの重要な用途として，関数近似問題への応用がある．これは与えられたサンプル（入力データと出力データの組）から入出力関係を推定し，任意の入力に対する関数値を求めるというものであり，工学のさまざまな領域で用いられる基本的技術である．もう一つの主要な用途であるパターン識別も，カテゴリーを離散的な関数値と考えれば，関数近似問題の一種とみなすことができる．

MLP については,「任意の可積分な連続関数を任意の精度で近似する MLP が存在する」という，いわゆる万能性の定理 [1], [2] がある．この定理は，関数を表現する能力に関する理論上の可能性を述べているだけであって，関数近似器としての性能を保証しているわけでは全くない．実際的な能力は，関数の表現能力だけでなく，サンプルを学習する能力や未知入力に対する汎化の能力にも大きく依存するからである．

しかしながら，上記定理のことを「十分多数の中間（隠れ）素子があれば，任意の関数を近似できる」，あるいは「万能な関数近似能力をもつ」と言い表すことがよくある．このように言う場合，学習のことは考慮に入っていないのであるが，これを「MLP はどんな関数近似問題も解ける」という意味に誤解する例がしばしば見受けられる．このような誤解を避けるために，本論文では,「近似能力」と「表現能力」とを区別して用い，前者は後者だけでなく学習能力や汎化能力も含んだ実際的な能力を指すものとする．同様に「近似できる」とは,「サンプルの学習によって精度よく近似することができる」という意味である．

上記の誤解を助長している一つの要因として，関数近似に関する解説や論文において数値実験例として扱われているのが，ほとんどの場合連続な1変数関数であることが挙げられる．これには，1変数関数だと近似の様子を理解しやすいという理由もあるだろうが，

† 筑波大学　大学院システム情報工学研究科，茨城県
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba-shi, 305-8573 Japan
a) E-mail: nona@bcl.esys.tsukuba.ac.jp
b) E-mail: fumihide@iit.tsukuba.ac.jp
c) E-mail: mor@bcl.esys.tsukuba.ac.jp

実験上の都合という面も大きい．実は，1 変数関数，あるいは実質的に 1 変数（独立変数は 1 つだけで他は従属変数または冗長変数）の関数は，真の（独立変数が複数ある）多変数関数に比べて学習するのがずっと容易であり，少々複雑であってもうまく近似できるのである．

しかし，そもそも連続な 1 変数関数であれば，MLP やその他の関数近似器を用いるまでもない．例えばすべてのサンプルを記憶しておいて，隣り合うサンプル間を直線または曲線で補間するといった方法でも実用上十分であろう．これに対して多変数関数の場合，入力空間が広いので単純な補間は困難であるし，問題が難しくなるため手法による差が出やすい．

また，関数が連続であるというのもかなり強い制約である．もちろん，いたるところ不連続であれば近似のしようがないが，大部分の領域では連続だが一部分で不連続という関数は実際の問題でよく現れる．そのような関数を事前知識なしにうまく近似できれば，非常に有益であろう．したがって，関数近似器の性能は，部分的に不連続な多変数関数を用いて評価すべきと考える．

以前我々は，MLP のような従来型のニューラルネットでは，複数の分散表現を統合することが困難であり，期待されるような汎化が生じないことを示した [3]．また，選択的不感化という手法を導入することによってこの問題が解決できることを明らかにし，選択的不感化ニューラルネット（Selective Desensitization Neural Network，以下 SDNN）を提案した．この SDNN を 2 変数関数の近似問題に適用したところ，高い汎化能力をはじめとする数々の優れた性質を示した．ただし，このとき用いたのはごく単純な連続関数であり，より複雑な関数や不連続関数の場合にどうであるかは不明であった．

その後 SDNN を 4 次元の連続状態空間における強化学習の価値関数近似に用いたところ，従来にない高い性能が得られた [4]．このことは SDNN の優れた関数近似能力を示しているが，近似の対象である価値関数の真の値が不明なため，近似精度や汎化能力について十分に解析することができない．また，4 変数関数の視覚的な表示は困難であるため，具体的な近似の様子を詳細に知ることができない．

そこで本研究では，不連続な部分を含む比較的複雑な既知の 2 変数関数を標的関数とした数値実験を行い，MLP と SDNN に加えて放射状基底関数ネットワーク（Radial Basis Function Network，以下 RBFN）と並列パーセプトロン（Parallel Perceptron，以下 PP）の関数近似能力を比較検討する．これを通じてニューラルネットによる関数近似に対する理解を深めると共に，SDNN の高い性能が何に由来するのか追究する．

以下では，まず研究の背景となる知見および対象とするニューラルネットについて簡単に説明する．次いで実験の方法および結果について述べた上で，主に SDNN に関して詳しい解析と考察を行う．

2. 研究の背景

2.1 多層パーセプトロン（MLP）

まず，2 個の入力素子と 1 個の出力素子からなる単純パーセプトロンによって 2 変数関数 $f(x, y)$ を近似することを考える．各入力素子は x と y の値をそのままアナログ表現しており，出力素子は $z = g(w_1x + w_2y + h)$ を計算することによって $f(x, y)$ を近似するものとする．ここで w_1, w_2 は結合荷重，h は定数である．$g(u)$ は活性化関数であり，パターン識別の場合には $u > 0$ のとき 1，$u \leq 0$ のとき 0 をとるヘビサイド関数を用いることが多いが，ここでは任意の単調増加関数とする．

よく知られているように，単純パーセプトロンは XOR 課題を解けない [5]．つまり，$f(0, 0) = f(1, 1) = 0$, $f(0, 1) = f(1, 0) = 1$ を満たす関数を表現できない．それどころか，x-y 平面上のある直線およびそれと平行な直線上では一定の関数値をとり，それ以外の直線上では関数値が単調に増加または減少するもの以外は表現できない．このように，一つの単純パーセプトロンが近似できる関数というのは極めて限られる．

次に，2 個の入力素子と 1 個の出力素子の間に n 個の中間素子がある 3 層の MLP を考えよう．中間素子の活性化関数はシグモイド関数とし，出力素子の活性化関数を線形関数とする．このとき，中間層から出力層の部分は線形近似器にすぎないから，各中間素子が入力 (x, y) に応じて出力する関数値 $f_i(x, y)$ の線形和によって $f(x, y)$ を近似していることになる．ところが，入力層から中間層の部分だけを見れば，これは 2 入力 1 出力で活性化関数 $g(u)$ をシグモイド関数に固定した単純パーセプトロンが n 個並んだものである．MLP が万能の表現能力をもつといっても，任意の複雑な関数 $f(x, y)$ をいくつかの関数 $f_i(x, y)$ の線形和に展開したとき，それらすべてを単純パーセプトロンでうまく表現できるとは考えがたい．

実は，理論上の万能性の証明では，少しだけ平行移動した二つのシグモイド関数 $g(u)$ と $g(u-\delta)$ の差をとると，$u=0$ 付近を除いてほぼ 0 となることを用いている．直観的に言うならば，x-y 平面上のごく細い帯状の領域を担当する素子が無数あって，それらが平面を埋め尽くすことによって任意の連続関数を表現することを可能しているわけである．しかし，そのような方法では汎化能力は期待できないし，有限個のサンプルを学習したとき過剰適合が生じやすい．表現の効率も悪く，入力の次元が低くても非現実的な数の素子が必要となる．理論的に必要な中間素子数に関して，ある条件下において入力次元に依存しない上界がある [6] ことが知られており，それゆえ「次元の呪いを回避できる」としばしば言われるが，次元が低くても無数に必要というのでは実用上意味がない．

MLP の実用上の問題点として，学習が遅いことも挙げられる．最も一般的な誤差逆伝搬（BP）法を用いた場合，ローカルミニマムに陥って学習が収束しない（学習誤差が十分小さくならない）ことがよくある．そのため，学習アルゴリズムのさまざまな改良が提案されてきたが，問題の本質的な解決には至っていない．我々の考えでは，学習の収束性が悪い根本的な原因は MLP の関数表現効率の悪さにあり，学習アルゴリズムを変えるだけでそれが解消されることはない．

2.2 放射状基底関数ネットワーク（RBFN）

実用的な関数近似器としてよく用いられるのが RBFN である．これは，中心の異なる多数の放射状基底関数（ここではガウス関数を用いる）の線形和によって関数を近似する手法であり，基底関数を計算する素子を中間層に配置した一種の 3 層ニューラルネットとみなすことができる．数式で表すと，関数 $f(x,y)$ を

$$z=\sum_i w_i \exp\left\{-\frac{(x-x_i)^2+(y-y_i)^2}{\sigma_i^2}\right\} \quad (1)$$

によって近似する．ここで，x_i と y_i は i 番目の基底関数の中心の x 座標と y 座標，σ_i はガウス関数の広がりを表す定数である．w_i は i 番目の中間素子から出力層への結合荷重であり，通常は学習によってこの値を設定する．

RBFN は，十分な数の中間素子（基底関数）を適切に配置すれば，任意の連続関数を任意の精度で表現できる [7]．つまり，MLP と同様の万能性を有する．また，基底関数の形や配置が適切ならば，学習の収束性も良い．

2.3 並列パーセプトロン（PP）

一つの閾素子は，ヘビサイド関数を活性化関数とした単純パーセプトロンとみなすことができ，決定境界と呼ばれるある超平面（2 変数関数の場合は直線）によって入力空間を二つに分割する機能をもつ．このような閾素子を m 個並列に並べ，それらの出力値の総計（1 を出した素子の数）に応じて最終的な出力値を決定するのが PP である．PP は，3 層の MLP において，中間層の活性化関数をヘビサイド関数にし，中間層から出力層の結合荷重を固定したものとみなすこともできる．

最近，PP もまた万能性をもつことが示されている [8]．つまり，閾素子の数 m が十分大きければ，任意の有界な連続関数を任意の精度で表現可能な PP が存在する．但し，MLP の場合と同様に，この万能性自体に実用上の意味はない．また，単純パーセプトロンの学習は，解が存在すれば必ず有限ステップで収束するのに対し，PP の場合，必ず収束するとは限らない．

PP の学習には，基本的には単純パーセプトロンと同じ誤り訂正学習を用いるが，誤っているとみなす素子の決め方にはいくつかの方法がある．ここでは，次のような方法を用いる．

ある入力に対して，m 個の閾素子のうち l 個が 1 を出力すべきなのに，実際に 1 を出力したのは $k(<l)$ 個だけだったとしよう．このとき，0 を出力した $m-k$ 個の素子の中で，内部電位が最も閾値に近いものから順に $l-k$ 個を選び，これらについて誤り訂正学習を行う．$k>l$ の場合には，1 を出力した素子の中で内部電位が閾値に近いもの $k-l$ 個について学習を行う．

PP は，MLP や RBFN と違って，理屈の上では部分的に不連続な関数も表現可能である．但し，そのためには不連続な部分において，多数の閾素子の出力値が同時に変化する必要がある．これは多数の決定境界が不連続点上でちょうど重なることを意味するから，実現するのはそう容易でない．

2.4 選択的不感化ニューラルネット（SDNN）

選択的不感化とは，分散表現された 2 つのパターンがあるとき，一方のパターンに応じて他方の一部の要素を中立値に変えることによって，両者が表す情報を統合する手法である．具体的に 2 種類の n 次元の 2 値（±1）パターン $S=(s_1,\ldots,s_n)$ および $C=(c_1,\ldots,c_n)$ があるとき，S の要素の約半数をパターン C に応じて選んで 0 にする．これにより，約半数が 0 で残りが ±1

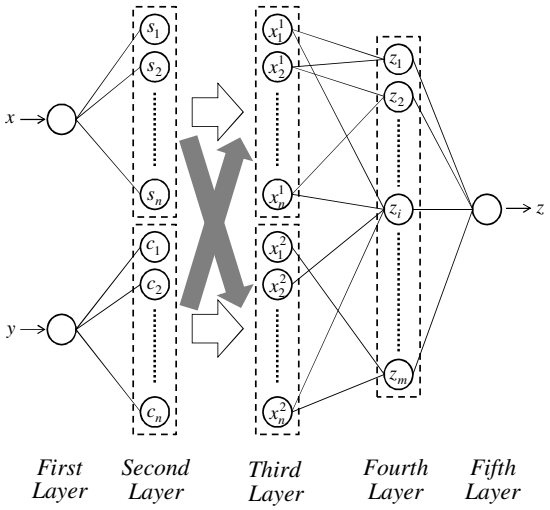


図 1　選択的不感化ニューラルネットを用いた 2 変数関数の近似器
Fig. 1　Two-variable function approximator using a selective desensitization neural network.

の 3 値パターンが得られるが，このパターンのことを「C によって修飾された S」といい，$S(C)$ で表す．同様に，S によって修飾された C を考えることもできる．

この手法を階層型ニューラルネットに適用したものが SDNN である．SDNN の構成にはかなりの自由度があるが，2 変数関数の近似器の基本的な構成を図 1 に示す（3 変数以上の場合については文献 [4] を参照されたい）．図では 5 層構造をしているが，これは他のニューラルネットと入力および出力層をそろえるためであり，文献 [3] や [4] では第 2 層と第 4 層の部分をそれぞれ入力層および出力層と呼んでいる．

第 2 層は，n 個ずつの素子からなる二つの素子群で構成される．素子群 1 の各素子は，入力変数 x の値が素子ごとに決められたある区間（1 つとは限らない）にあるとき 1，そうでないとき -1 を出力する．同様に素子群 2 の各素子は，y の値に応じて ± 1 を出力する．これによって，x および y の値は，それぞれ成分の約半数が 1 をとる n 次元 2 値パターン S および C として分散表現される（具体的な方法は **2.5** で述べる）．

第 3 層は，選択的不感化による修飾を相互に行ったパターン $S(C)$ および $C(S)$ を表す二つの素子群からなる．具体的には，素子群 1 の i 番目の素子は，信号 s_i および $c_{\sigma(i)}$ を受けとり，

$$x_i^1 = \frac{1 + c_{\sigma(i)}}{2} s_i \tag{2}$$

を出力する．ここで，$\sigma(i)$ は順列 $(1, 2, \ldots, n)$ をランダムな順序に置換したときの i 番目の要素を表す．同様に，素子群 2 の i 番目の素子の出力は

$$x_i^2 = \frac{1 + s_{\sigma'(i)}}{2} c_i \tag{3}$$

で与えられる（σ' は σ とは別の置換を表す）．

第 3 層から第 5 層は PP を構成する．すなわち，第 4 層には，第 3 層を入力とし 0 または 1 を出力とする閾素子が m 個並んでおり，第 5 層は第 4 層の出力値の合計 k に基づいて z を出力する 1 個の素子からなる．第 4 層から第 5 層への結合荷重は一様な定数であり，学習は第 3 層から第 4 層の結合荷重を PP と同様の方法で修正することによって行う．

第 3〜5 層が万能性をもつ PP であるため，SDNN もまた万能性をもつ．すなわち，理論上 n と m が十分に大きいとき，定義域と値域が有界な任意の連続関数を任意の精度で表現できる．また，PP と同様に，部分的に不連続な関数を表現できる場合がある．

2.5　パターンコーディング

SDNN の第 2 層のように，連続な変数の値を，対応するパターン（コードパターン）によって表現する方法を「パターンコーディング」と呼ぶことにする．これに対して，第 1 層のようにアナログ値を出力する 1 個の素子で表現する方法を「アナログコーディング」と呼ぶ．

パターンコーディングの具体的な方法はさまざまであるが，以下の（1），(2）を満たすことが必要であり，また（3），(4）を満たすことが望ましい．

（1）　変数値に関する情報がパターン全体に広く分散し，かつ全体として十分な冗長性をもつこと．これは分散表現の定義のようなものであるが，わかりやすく言うならば，パターンの成分のうち，どの 1 つまたは数個を見ても変数値が特定できないが，全体の半分程度を見れば必ず特定できるということである．

（2）　変数値が連続的に変化するにつれてコードパターンが徐々に変化すること．また，変数値が近いほどコードパターン間の相関が高く，遠いほど低くなること．これによって，非局所的な汎化が生じることが期待できる．

（3）　1 と -1 の成分が常にほぼ同数であること．これにより，第 3 層において常に約半数の素子が不感化されて 0 を出力することになる．

（4）　十分に離れた変数値を表すコードパターン間の相関が 0 であること．あるサンプルを学習したとき，

無関係な点への影響（干渉）が小さくなる．

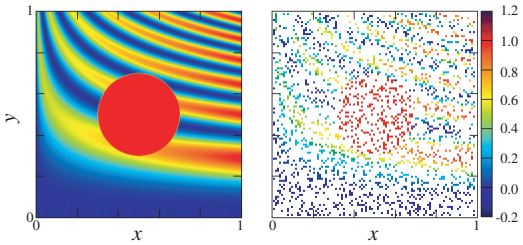
ここで，(2) と (4) に述べたコードパターン間の相関については任意性があり，どのように減少してどの程度離れると 0 になるのが最適であるかというのは近似する関数の形や空間周波数に依存する．しかし，実際試してみると，パターンの次元 n が十分に大きければ，相関の大きさや細かいコーディングの違いによる差はあまり見られなかった．したがって，コーディングを完全に最適化する必要はなく，せいぜい数通りを試す程度で十分である．

本研究で用いたコーディング，すなわち変数値 x に対するコードパターン $P(x)$ は以下のようなものである（なお，これは n がある程度大きい場合の簡便な方法であり，n が小さい場合には指定したいくつかのパターンを補間して作成する [4] のがよい）．

まず，$0 < x \leqq 1$ の区間を q 個に分割し，各区間に対応するコードパターンを $P_1, \ldots, P_q$ とする．すなわち，$(k-1)/q < x \leqq k/q$ のとき $P(x) = P_k$ である $(k = 1, \ldots, q)$．また，パターンの次元 n は偶数とし，1 と -1 の成分が同数となるパターンの 1 つを $P(0)$ とする．なお，分割数 q が大きいほど量子化による誤差が減少するが，必要な素子数は多くなる．

次に，$P(0)$ の成分の中から 1 および -1 をとるものをそれぞれ r 個ずつランダムに選び，それらの符号を反転して得られるパターンを P_1 とする．同様に，P_1 の成分の中から 1 および -1 をとるものを r 個ずつランダムに選んで反転し，P_2 を得る．以下同様に，P_{k-1} の成分の一部を反転することによって順次 P_k を作成する．

ところで，パターンコーディングは SDNN にとって必須であるが，SDNN に特有というわけではない．実施例はあまりないが，他のニューラルネットにも適用することができる．RBFN の場合は局所的な基底関数を用いるので意味がないが，MLP や PP の場合にはパターンコーディングによって汎化能力が高まる可能性がある（理論上の万能性は変わらない）．そこで，以下の実験では MLP や PP においてパターンコーディングを用いた場合も扱い，MLP-A や PP-P のように末尾に-A および-P を付けることによってアナログコーディングとパターンコーディングを区別することにする．



(a) Target function $f(x, y)$　(b) Training samples for Experiment 1 ($N = 2000$)

図 2　実験方法
Fig. 2　Experimental method.

3. 数値実験

3.1 標的関数

近似の対象とする既知の関数（標的関数）として，$\{(x, y) : x, y \in [0, 1]\}$ を定義域とし，0 から 1 の値をとる 2 変数関数

$$f(x,y) = \begin{cases} 1 & \left((x-0.5)^2+(y-0.5)^2 \leqq 0.04\right) \\ \dfrac{1+x}{2}\sin^2(6\pi\sqrt{x}y^2) & (\text{otherwise}) \end{cases} \quad (4)$$

を用いる．図 2(a) はこの関数を視覚的に表示したものであり，x-y 平面上の各点の関数値を色で表している．

このような関数を選んだのは，これが次の条件を満たすからである．

（1） ほとんどの点で連続であるが，不連続な部分もある（不連続関数の近似能力を見ることができる）．

（2） 空間周波数，および勾配の大きさと方向が領域によって異なる（これらが一定だと一般性に乏しく，特定の手法に有利となる可能性がある）．

（3） 同じ関数値をとる点が連続的に広がっている（非局所的な汎化能力を見ることができる）．

なお，他にもさまざまな関数を試したが，上記の条件を満たしていれば結果はほとんど同じであった．また，条件の一部を満たしていない場合でも，基本的に同じ傾向であった．したがって，以下で示す実験結果にはかなり強い一般性がある．

3.2 方　　法

実験 1 では，6 種類の関数近似器 MLP-A，MLP-P，PP-A，PP-P，RBFN，SDNN について，それぞれ以下の手順で性能を評価した．

（1） 標的関数の定義域中に設定した 0.01 間隔 ($101 \times 101 = 10201$ 個) の格子点の中から N 個のサ

ンプル点をランダムに（但し重複することなく）選ぶ．$N = 2000$ の場合のサンプル点の例をを図 2(b) に示す．

（2） 各サンプル（サンプル点 (x, y) と関数値 $f(x, y)$ の対）について，近似器の学習を十分な回数だけ繰り返し行う．具体的な学習回数は，MLP が 2000 回，それ以外は 300 回である．この数は近似誤差の変化を見ながら相当大きめに設定したものであり，それだけの回数が必要というわけではない．

（3） 10201 個の格子点すべてについて，近似器の出力 z と真の関数値との差の絶対値 $|z - f(x, y)|$ を求め，その平均値を算出して近似誤差とする．

（4） $N = 100, 200, 500, 1000, 1500, 2000, 4000$ の 7 通りについて（1）〜（3）を行う．

（5）（1）〜（4）を 1 実験試行とし，乱数系列を変えてこれを 10 回繰り返す．

各近似器のパラメータ等は以下の通りである．

MLP 学習アルゴリズムは標準的な BP 法とし，中間素子数は 50 に設定した．学習のローカルミニマム問題などがあるため，パラーメータや結合荷重の初期値を変えた試行を 10 回程度繰り返し，最も結果がよかったものを採用した．なお，中間素子数をさまざまに変えた実験も行ったが，結果に大きな差はなかった．

RBFN 基底関数はすべて $\sigma_i = 0.1$ のガウス関数とし，441 個を各中心が 0.05 間隔の格子点上になるよう配置した．一般に，基底関数の数 n を大きくするほど表現能力が高くなるが，汎化能力は低下する．この点も考慮しつつ，適切と思われる n および σ_i の値を実験的に求めた．

PP 単純パーセプトロン（閾素子）の数 m は 280 とし，そのうち k 個が 1 を出したときの近似器の出力値を $z = 0.005k - 0.2$ とした．なお，理論上 m が大きいほど表現能力が高いが，これ以上 m を増やしても結果に大きな違いは見られなかった．

SDNN 過剰適合が生じることはないので n は大きいほどよい [3] が，他の近似器とのバランスを考慮して $n = 200$（第 3 層の素子数は 400）とした．第 4〜5 層については PP と全く同じである（$m = 280$, $z = 0.005k - 0.2$）．パターンコーディングに関するパラメータは $q = 100$, $r = 5$ とした．なお，同一のコードパターン（$n = 200$）を MLP-P および PP-P でも用いた．

実験 1 で求めた近似誤差には，サンプル点における誤差（学習誤差）とそれ以外の点における誤差（汎化誤差）の両方が含まれている．これは，学習後に実際に近似器として利用する場合，入力データがサンプル点と一致しようがしまいが，正しい関数値にどれだけ近い値を出力するかが重要だと考えるからである．また，実験 1 の場合，サンプル数を増やしていくとサンプル点の密度が高まっていき，サンプル点とそうでない点を区別する意義が薄くなる．実際，N が大きい場合の学習誤差は，評価値とあまり差がなかった．

しかし，現実にはサンプル数を増やしても，サンプル点の密度が一様に高まるとは限らない．ある入力領域ではサンプルが全く得られないこともありえる．そのような場合にも，他の領域のデータから関数値を推定できれば有用であろう．また，MLP のような非局所的近似手法には，サンプル点の近傍に限らない，広域的な汎化の能力が期待されていると思われる．

このような観点から，実験 2 ではサンプルを全く与えない領域を設定し，その領域における汎化誤差を評価した．具体的には，図 6(a) の黒く塗りつぶされた部分（全体の約 4 分の 1）をそのような領域とし，これに含まれる 2675 個の格子点に関して誤差を求めた．対象とする近似器は，実験 1 の結果が非常に悪かった MLP-A と PP-A を除く 4 種類（MLP-P，PP-P，RBFN，SDNN）とした．近似器のパラメータやその他の実験手順は実験 1 と同じである．

3.3 結　　果

実験 1 の結果を図 3 および図 4 に示す．

図 3 は，近似誤差の 10 回の実験試行に関する平均値を，サンプル数 N に対してプロットしたものである．エラーバーは標準偏差を表すが，一部の場合を除いてごく小さな値であった．

この図からわかるように，SDNN は全体として近似精度が高い．MLP-P は，$N > 1000$ のとき SDNN をやや上回る精度を示したが，N が小さいときの誤差が大きい．RBFN は全体にやや精度が劣り，特に N が小さいときの誤差が大きい．PP-P および MLP-A は $N < 1000$ の範囲ではある程度誤差が減少するが，それ以上はほとんど減少しない．PP-A は，N に関係なく誤差が大きい．

図 4 は，ある平均的な実験試行において，図 2(b) に示したサンプル（$N = 2000$）を学習した後の各関数近似器の出力値を，図 2(a) と同様の方法で表示したものである．一見してわかるように，MLP-A と PP-A は，標的関数の近似が全くできていない．PP-P は大まかには近似できているが，細部は標的関数とかなり

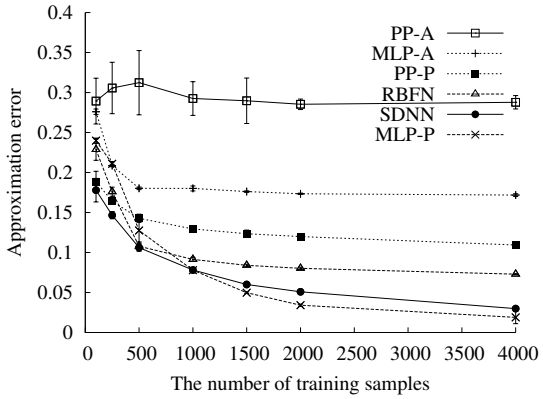


図 3　実験 1 の結果
Fig. 3　Results of Experiment 1.

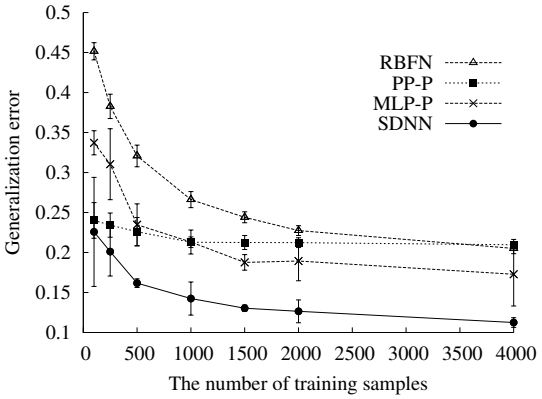
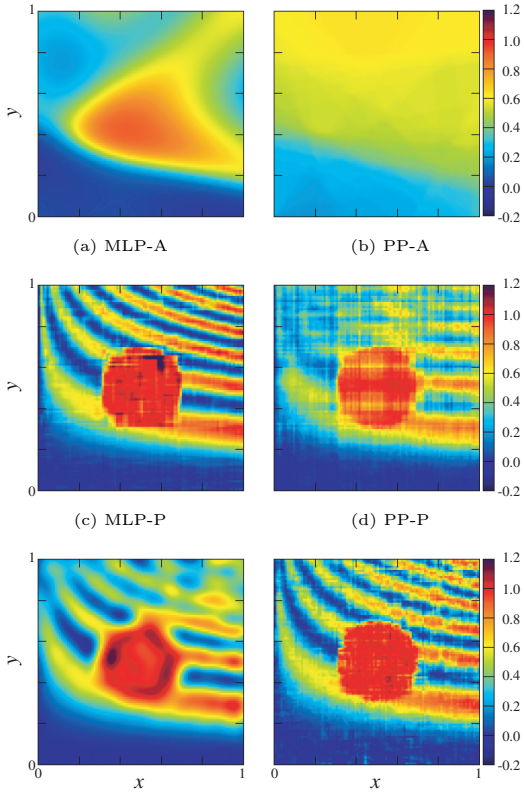


図 5　実験 2 の結果
Fig. 5　Results of Experiment 2.

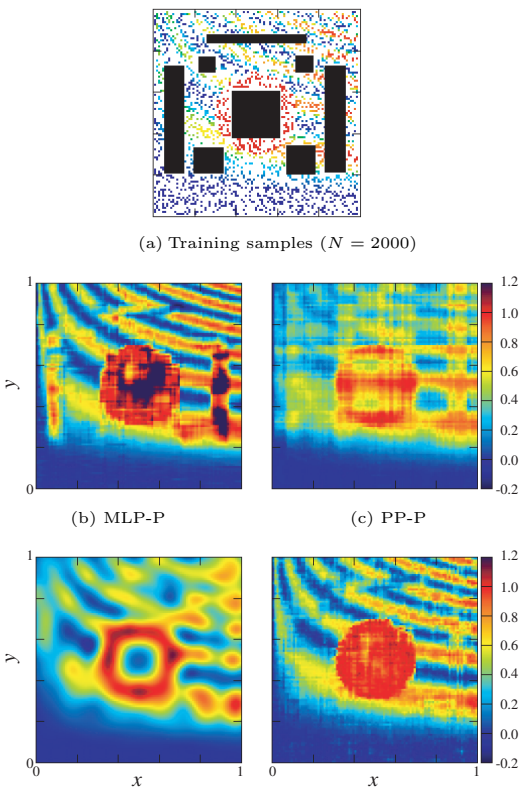


(a) MLP-A　(b) PP-A

(c) MLP-P　(d) PP-P

(e) RBFN　(f) SDNN

図 4　学習後の各近似器が表現する関数 （$N = 2000$）
Fig. 4　Approximated functions after training.



(a) Training samples ($N = 2000$)

(b) MLP-P　(c) PP-P

(d) RBFN　(e) SDNN

図 6　汎化能力の比較
Fig. 6　Comparison on generalization ability.

違っている．RBFN は比較的よく近似しているが，中央付近と右上の領域における近似精度はあまりよくない．MLP-P と SDNN は，どちらも標的関数をかなりよく近似しているが，近似の仕方に若干の違いがあり，前者は不連続点 (中央の円周) の付近，後者はなだらかに変化する領域の近似がやや苦手である．

同様に，実験 2 の結果を図 5 および図 6(b)〜(e) に示す．SDNN の汎化誤差は他よりも圧倒的に小さく，実験 1 の RBFN の近似誤差と比べてもさほど遜色がないことがわかる．PP-P は，$N < 500$ のときの誤差は SDNN に次いで小さいが，N を増やしても誤差がほとんど減っていない．MLP-P および RBFN は，実験 1 に比べて誤差の増加が著しく，PP-P と同程度の性能であった．

4. 考　　察

これまでの議論と数値実験の結果を基に，各関数近似器の性能や特徴などについて考察する．

4.1 RBFN

実験 1 の結果からわかるように，局所基底関数を適切に配置し，十分なサンプルを与えたとき，RBFN の近似精度は比較的よい．また，構造が単純で学習の収束も早いという点も，実用上有利である．

但し，RBFN にはいくつかの弱点がある．まず，関数が不連続な点や関数値の変化が激しい（空間周波数が高い）領域での誤差が大きい．一方で広い範囲にわたって関数値が一定値（0 を除く）をとる場合にも，うまく近似することができない．例えば，図 4(e) 中央の $f(x, y) = 1$ となる領域において，出力値は 0.9 以下から 1.1 以上まで波を打っている．

また，図 6(d) から明らかなように，サンプルが与えられない空白領域が大きいと，そこでの近似は全く不正確になる．このような領域での誤差を減らすためには，基底関数の広がりを表すパラメータ σ_i を大きくする必要があるが，そうすると全体の近似誤差が増大してしまう．このように，RBFN では一般に近似精度と汎化能力が両立しない．

こうした弱点は，基底関数をうまく配置することによってある程度補うことができると考えられる．しかし，基底配置の最適化には，標的関数の性質（空間周波数など）や問題の条件（サンプルの分布など）に関する事前知識が必要である．基底の配置などを適応的に修正する方法も提案されているが [9]，学習の収束性の低下や計算コストの増大を考えるとあまり実用的ではない．

さらに，本研究の対象外ではあるが，一般に RBFN をはじめとする局所的近似手法には，入力変数の数が増えると計算コストが爆発的に増大してしまう [4] という本質的な問題がある．関数値に無関係な冗長変数がある場合の悪影響も大きい．そのため，入力変数が多くなると事前の変数選択や次元圧縮が不可欠となるが，次元圧縮はそれ自体が難しい問題である．したがって，実際に RBFN を適用できる範囲はそれほど広くない．

4.2 MLP

通常のアナログコーディングを用いた MLP，すなわち MLP-A は，他の近似器の 20 倍以上の時間をかけて学習を繰り返したにもかかわらず，サンプル数や中間素子数，その他のパラメータをどう調整しても図 4(a) のようなぼんやりとした関数しか表現できなかった．

これは，たまたま今回の標的関数が MLP-A と相性が悪かったからではない．経験上，また 2.1 の議論からも，MLP-A がうまく学習できるのは次のいずれかの場合に限られると考えられる．

（1） 近似する関数の入力空間が 1 次元，または実質的に 1 次元（独立変数が 1 つのみ）である．

（2） 近似する関数が少数のシグモイドニューロンの線形和で表すことができる．

（3） 関数の定義域または実際に入力が与えられる領域が多次元空間の一部分に限られており，そこでは局所的に（1）または（2）が成り立つ．

これに加えて，計算量の多さ，中間素子数の選択や学習パラメータの調整の手間などを考慮すると，MLP-A に実用上の利点は全く見当たらない．

実験 1 の結果が示すように，パターンコーディングを導入することによって近似精度が大きく向上する場合がある．この近似精度の向上は，学習誤差が非常に小さくなった（$N = 2000$ の場合で 0.004）こと，それに伴ってサンプル点近傍の汎化誤差も大きく低下したことによる．

しかし，実験 2 の結果は，その場合でも広域的な汎化能力は向上しないことを示している．これは，1 対多対応による荷重平均化の問題 [3] を避けるために，学習によって中間層の表現（入力信号に対する活動パターン）が次のように変化するからだと考えられる．

まず，学習前の初期状態において，中間層への結合荷重はランダムな値に設定されている．このとき，ある入力パターン $\boldsymbol{x}_1$ が与えられたときの中間層の表現 $\boldsymbol{y}_1$ と，$\boldsymbol{x}_1$ に非常に近い入力パターン $\boldsymbol{x}_2$ に対する表現 $\boldsymbol{y}_2$ とは，非常に似たものとなる．$\boldsymbol{x}_1$ と $\boldsymbol{x}_2$ の相関が 1 から 0 に低下するにつれて $\boldsymbol{y}_1$ と $\boldsymbol{y}_2$ の相関も徐々に低下するが，両者はほぼ同じ減少曲線を描く．

中間層の学習は，一般に出力すべき値が異なる入力

信号を分離する（相関を減らす）働きがある．そのため学習後の中間層表現間の相関は，入力パターン間の相関よりも急速に減少しやすい．実際，実験 2 において学習の前後を比較すると，関数値が異なる二つのサンプル点の表現間の相関は大きく低下していた．また，サンプル点とその周辺の表現間でも相関が減少する傾向が見られたが，特にサンプルを与えない領域の内部においてそれが顕著であった．このような相関の低下は，サンプル点から離れた領域に汎化が及ぶことを妨げる．

一方で，サンプル点の中間層表現が互いに分離すれば，中間層から出力層の学習が容易になって学習誤差が減少する．但し，そうなるように中間層を学習するのは，入力空間においてサンプル点同士がある程度離れていれば容易であるが，近すぎると困難な場合が多い．したがって，パターンコーディングによって 2 次元の入力空間中のサンプル点が高次元のパターン空間に分散されたことが，学習誤差の大幅な減少をもたらしたと思われる．

以上のように，MLP-P は RBFN と同様に近似精度と広域的な汎化能力を両立させることができない．また，MLP-A よりも学習の収束性は高いものの素子数が多いので，計算コストはやはり他の近似器よりかなり大きい（本実験では 20 倍以上）．入力変数が増えたときの近似能力については今のところ不明であるが，計算コストがさらに増大することは間違いない．これらを考慮すると，MLP-P が有用な場合というのはかなり限定的であろう．

4.3 PP と SDNN

SDNN は近似精度や汎化能力が共に高いだけでなく，計算量や学習の収束性の点でも優れている．また，関数の形や実験条件の変化に対するロバスト性が高いため，パラメータの設定が容易である．さらに，先行研究 [4] において，入力変数の増加に対して計算コストの増加が緩やかなこと，冗長変数による悪影響をほとんど受けないこと，追加学習が容易であることが示されている．これらを考慮すると，汎用の関数近似器としての有用性は，階層型ニューラルネットだけでなく，既存のどの近似器と比べても高いと思われる．

一方，PP の関数近似能力はあまり高くなく，MLP よりも劣っていた．計算量の少なさを考慮しても，決して実用的ではない．このように性能が低く，あまり一般に知られてもいない PP を本研究で敢えて取り上げたのは，SDNN を解析し，その優れた能力が何に由来するのかを理解する上で有益だからである．

本研究で扱っている SDNN は，PP-P に選択的不感化法を導入したものに相当する．従って，両者の差が選択的不感化の効果を反映している．また，PP-P と PP-A の差は，SDNN の性能に対するパターンコーディングの寄与を表していると考えられる．

そのような観点で実験結果を眺めると，パターンコーディングに近似精度と広域的な汎化能力をある程度高める効果があり，選択的不感化はそれをさらに強化するものだと言える．MLP-P における中間層の学習が，いわば汎化能力を犠牲にして近似精度を高めるのに対して，選択的不感化は両者を共に向上させる点が重要である．

4.4 選択的不感化の効果

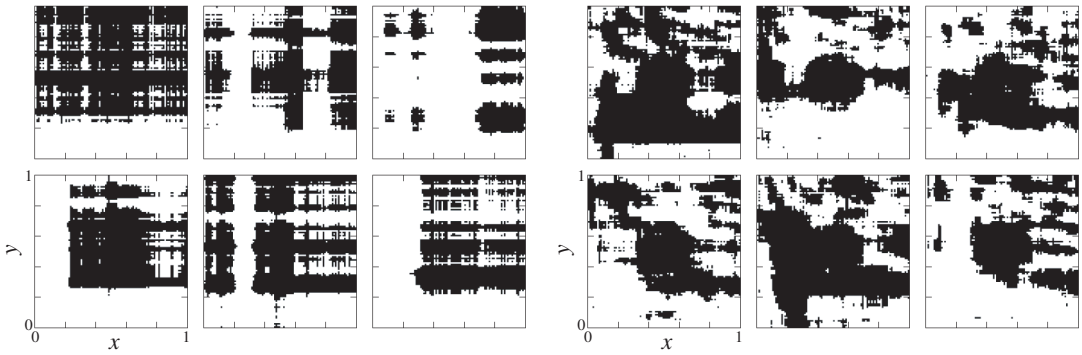
選択的不感化の効果をより具体的に解明するために，SDNN と PP-P の近似の様子の違いに着目しよう．

図 4 の各グラフの中央付近を見ると，PP-P では赤色（出力値約 1.0）の中に黄〜橙色（0.6〜0.8）の部分が筋状に混じっているのに対し，SDNN ではほぼ一様に赤色である．図 6 ではその差が一層顕著である．このことは，特に関数値が一定となる領域において，選択的不感化の効果が強く現れることを示してる．

そのメカニズムを考察するために，まず関数値が変数 x だけで決まり，変数 y は関数値に全く影響しない冗長変数であると仮定する．SDNN は冗長な入力変数に対して非常にロバストであることが知られているが [4]，これは，冗長変数 y の値だけが異なるいくつかのサンプルを学習した場合，x を表現する第 3 層の素子のどれが不感化されても同じ関数値が出力されるように学習がなされるからである．これによって y の値に基づく選択的不感化が無効化され，y の出力値への影響が非常に小さくなる．つまり，y 方向に関数値が同じサンプル点がいくつか並んでいると，y の全域にわたって強力な汎化が生じるのである．

y が完全な冗長変数でなくても，「x や y の値がある範囲にあれば関数値は y に依存しない」という場合，定義域をその範囲に限ってしまえば y は冗長変数である（このような y を「部分的に冗長」な変数と呼ぶことにする）．上記のメカニズムはこの場合にも機能し，y が冗長な範囲にわたって強い汎化が生じるはずである．

ただし，その範囲外のサンプルも学習するため，干渉によって範囲内の出力値が y に依存して変動する可能性がある．しかし，2 種類の領域におけるコードパ



(a) PP-P　　(b) SDNN

図 7　個々の閾素子の決定境界
Fig. 7　Decision boundaries of individual threshold elements.

ターンの相関が非常に大きくない限り，汎化の効果が打ち消されるほど強い干渉は生じないと考えられる．これに関する解析は十分にできていないが，上記の実験結果はこの考えが正しいことを示唆している．

次に注目すべきは，PP-P では同じ色（一定の出力値をとる領域）が縦横の筋状に伸びているのに対し，SDNN ではそのような筋が少ない点である．これが両者の全体的な近似能力の差につながっているように思われる．

この現象を追究するために，PP-P と SDNN を構成する個々の単純パーセプトロンが入力変数 (x, y) の空間をどのように切り分けるかを調べた．図 7 は，図 4 に示した実験試行に関する結果である．280 個の閾素子の中から典型的なものを 6 個ずつ選んで表示した．各グラフの白い部分が 0 を出力する領域，黒い部分が 1 を出力する領域を表す．このようなグラフを 280 個の素子すべてについて重ね，各点の平均濃度に応じて色をつけたものが図 4 のグラフに相当する．

この図からわかるように，2 種類の領域が共に入力空間の広い範囲に分布しており，領域の境目すなわち決定境界はかなり複雑に入り組んでいる．パターンコーディングを用いない PP-A の場合の決定境界は 1 本の直線であるから，このような複雑な領域分布自体はパターンコーディングの効果だと言える．

しかしながら，(a) の PP-P の場合，細かい凹凸を無視すれば，境界の大部分が縦または横方向の直線である．そのため，特に標的関数の等高線が曲線や斜め方向の直線を描く部分においてうまく近似ができないと考えられるが，このような境界になる理由は以下のように説明できる．

まず，x-y 平面中に辺が軸と平行な長方形 ABCD を任意にとり，その座標を A(x_1, y_1)，B(x_1, y_2)，C(x_2, y_2)，D(x_2, y_1) とおく．このとき，対角線上の点 A と C が一方の領域に，B と D がもう一つの領域に属するよう切り分けるという課題（一般化 XOR 課題と呼ぶ）は，PP-P の各単純パーセプトロンには解くことができない．x と y のコードパターンを連結した $2n$ 次元パターンの空間において，この 4 点の配置は線形分離不可能だからである．

したがって，実現可能な決定境界は，x_1, x_2, y_1, y_2 をどう選んでもこの課題の解にならないようなものに限られる．実際，図ではすべてそうなっている．この制約は相当に厳しく，斜め方向に走る境界はごくわずかしか許されない．必然的に境界が複雑になればなるほど，縦横の直線が大部分を占めることになるのである．

一方，(b) の SDNN では，境界が曲線を描く部分や斜め方向に延びる部分も多く見られる．このことは，SDNN の各単純パーセプトロンが一般化 XOR 課題を解けることを意味する．直観的に言うならば，これは二つのコードパターンを選択的不感化によって統合することによって，一方のコードパターンの違いがパターン全体に及ぶからである．そのため，両者を単に連結した場合と比べて，一方のコードパターンのみが異なるときに異なる値を出力するよう学習するのが容易である．

以上の考察から，選択的不感化には，ある変数が部分的にでも冗長な場合に強い汎化を生む効果と，個々

の閾素子の決定境界の自由度を高めることによって近似誤差を減らす効果の二つがあると考えられる．

5. む す び

多層パーセプトロン（MLP），放射状基底関数ネットワーク（RBFN），並列パーセプトロン（PP），および選択的不感化ニューラルネット（SDNN）という 4 種類の階層型ニューラルネットを，2 変数関数の近似問題に関して実験的に比較し，学習能力や汎化能力を含めた関数近似能力は SDNN が最も優れていることを示した．SDNN は，計算コストや学習の収束性，パラメータ調整の容易さなど，実用的な面でも優れていた．

また，実験結果の解析から，SDNN の高い関数近似能力には，アナログ値を多次元の 2 値パターンによって分散表現するパターンコーディングと，複数のパターンを統合する選択的不感化の両方が貢献していることがわかった．前者にはサンプルの学習を容易にすると共に広域的な汎化を可能にする働きがあり，後者には関数表現能力を高めて近似誤差を減らすとともに，入力に冗長性がある場合に汎化能力を高める効果があると考えられる．

本研究では，解析の容易さを優先して近似対象を 2 変数関数に限定したが，関数近似器の特性は変数が多いほど顕著に表れると考えられる．3 変数以上の関数を対象とした実験と解析は今後の課題であるが，先行研究 [4] および本研究の結果から見て，SDNN の優位性は一層拡大すると思われる．

ところで，パターン識別は関数近似の一種とみなすことができるが，サポートベクターマシンや各種の統計的手法もよく用いられるため，関数近似の場合とはやや事情が異なる．しかし，**4. 4** の考察から，SDNN はパターン識別器としても有用だと考えられる．特に，入力変数（特徴量）が多く冗長性が高い場合や，識別境界面が複雑なのに十分な数のサンプルが得られない場合，従来手法を凌駕する可能性がある．実際，表面筋電位信号から複数の動作を識別する問題に SDNN を適用することによって，これまでになく高い実用性が得られている [10]．

本論文でも指摘したように，MLP の理論的万能性と現実的性能との間には大きなギャップがあり，このことがニューラルネット全般の評価を押し下げているように感じられる．本研究の結果が，階層型ニューラルネットの正当な評価と幅広い活用につながることを期待したい．

謝辞　数値実験の追試を行い有益なコメントを下さった篠塚正成，堀江和正，桑原昭之の各氏に感謝する．本研究の一部は，科学研究費補助金特定領域研究「情報爆発 IT 基盤」（課題番号 21013007）および基盤研究 (B)（課題番号 22300079）の支援を受けて行われた．

文　献

[1] B. Irie and S. Miyake, “Capabilities of three-layered Perceptrons, ” Proc. IEEE International Conference on Neural Nerworks, vol.1, pp.641–648, 1988.

[2] K. Funahashi, “On the approximate realization of continuous mappings by neural networks, ” Neural Networks, vol.2, pp.183–192, 1989.

[3] 森田昌彦，村田和彦，諸上茂光，末光厚夫, “選択的不感化法を適用した層状ニューラルネットの情報統合能力, ” 信学論 (D-II), vol.J87-D-II, no.12, pp.2242–2252, Dec. 2004.

[4] 新保智之，山根　健，田中文英，森田昌彦, “選択的不感化ニューラルネットを用いた強化学習の価値関数近似, ” 信学論 (D), vol.J93-D, no.6, pp.837–847, June 2010.

[5] M. Minsky and S. Papert, Perceptrons: An Introduction to Computational Geometry, MIT Press, 1969.

[6] A. R. Barron, “Universal approximation bounds for superpositions of a sigmoidal function,” IEEE Transactions on Information Theory, vol.39, no.3, pp.930–945, 1993.

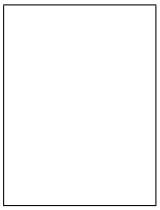
[7] J. Park and I.W. Sandberg, “Universal approximation using radial-basis-function networks, ” Neural Computation, vol.3, no.2, pp.246–257, 1991.

[8] P. Auer, H. Burgsteiner and W. Maass, “A learning rule for very simple universal approximators consisting of a single layer of perceptrons, ” Neural Networks, vol.21, no.5, pp.786–795, June 2008.

[9] S. Chen, C.F.N. Cowan, and P.M. Grant, “Orthogonal least squares learning algorithm for radial basis functions networks, ” IEEE Transactions on Neural Networks, vol.2, pp.302–309, 1991.

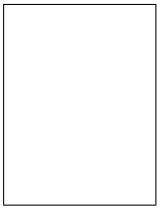
[10] H. Kawata, F. Tanaka, A. Suemitsu and M. Morita, “Practical surface EMG pattern classification by using a selective desensitization neural network, ” Neural Information Processing (Part II), Lecture Notes in Computer Science, vol. 6444, pp.42–49 (2010).

（平成年月日受付，月日再受付）



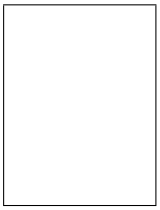
野中　和明

平 22 筑波大・工学システム学類卒．現在，同大大学院システム情報工学研究科在学中．神経回路モデルの研究に従事．



田中　文英

平 15 東工大大学院総合理工学研究科博士課程修了．博士 (工学)．ソニー (株) 及びソニー・インテリジェンス・ダイナミクス研究所 (株) リサーチャー，Universiy of California，San Diego 客員研究員を経て，現在，筑波大学大学院システム情報工学研究科准教授，JST さきがけ「情報環境と人」研究員．人間‐ロボット間インタラクション，発達学習，社会的相互作用の研究に従事．平 13 人工知能学会全国大会優秀論文賞，平成 17IEEE 国際会議 RO-MAN Best Paper Award など．



森田　昌彦　（正員）

昭 61 東大・工・計数卒．平 3 同大大学院博士課程修了．日本学術振興会特別研究員，東京大学工学部助手を経て，平 4 筑波大学電子・情報工学系講師．同大機能工学系助教授などを経て，平 19 より同大大学院システム情報工学研究科教授．脳の情報処理機構及び神経回路網による情報処理の研究に従事．平 5 日本神経回路学会研究費，平 6 同学会論文賞，平 11 日本心理学会研究奨励賞受賞．

Abstract Some feedforward neural networks can theoretically approximate any continuous function, but their practical approximation performance is not guaranteed. We compared four function approximators, multi-layer perceptron (MLP), radial basis function network (RBFN), parallel perceptron (PP), and selective desensitization neural network (SDNN) in numerical experiments of approximating a two-variable function. The results showed that SDNN offered the best performance in the approximation ability including learning ability and generalization ability, and also in other practical aspects such as low computational cost. Introducing pattern coding and selective desensitization are considered to contribute to the high performance of SDNN. These findings will greatly expand the application of neural networks.

Key words Neural Network, Machine Learning, Generalization Ability, Approximation Error, Distributed Representation