

TR-H-087

0043

多価正則化ネットワーク
—多対 h 写像を学習する多層ネットワークの理論—

志沢 雅彦

1994. 7. 29

ATR 人間情報通信研究所

〒619-02 京都府相楽郡精華町光台 2-2 ☎07749-5-1011

ATR Human Information Processing Research Laboratories

2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

Telephone: +81-7749-5-1011

Facsimile: +81-7749-5-1008

多価正則化ネットワーク — 多対 h 写像を学習する多層ネットワークの理論 —

正員 志沢 雅彦 †

Multi-Valued Regularization Network —A Theory of Multi-Layer Networks
for Learning Many-to- h Mappings—

Masahiko SHIZAWA †,Member

†ATR 人間情報通信研究所 京都府
ATR Human Information Processing Research Laboratories, Kyoto, 619-02
Japan

Multi-Valued Regularization Network —A Theory of Multi-Layer Networks for Learning Many- to- h Mappings—

Abstract The regularization network (RN) is extended to approximate *multi-valued functions* so that many-to- h mapping, where h denotes multiplicity of the mapping, can be represented and learned from a finite number of input-output examples *without hard clustering operations* on the training data set. Multi-valued function approximations are useful for learning ambiguous input-output relations from examples. This extension, which we call the Multi-Valued Regularization Network (MVRN), is derived from the Multi-Valued Standard Regularization Theory (MVSRT), which is an extension of standard regularization theory to multi-valued functions. MVSRT is based on a direct algebraic representation of multi-valued functions by using tensor product (Kronecker's product). By simple transformation of the unknown functions, we can obtain *linear* Euler-Lagrange equations. Therefore, the learning algorithm for MVRN is reduced to solving a linear system. It's rather surprising that the dimension of the linear system is invariant to the multiplicity h . The proposed theory can be specialized and extended into Radial Basis Function (RBF) Methods, Generalized RBF (GRBF), spline approximation, and HyperBF networks of multi-valued functions. We also describe how the vector-valued function approximations can be extended into the multi- and vector-valued function approximations.

Keywords: Regularization network, Function approximation, Regularization theory, Multi-valued mapping, Computational learning theory, Multi-layer network, Neural computation, Inverse problem.

目次

あらかし	3
1 はじめに	5
2 スカラー値の多価正則化ネットワーク	10
2.1 スカラー値多価関数の一般表現	11
2.2 h 価スカラー値関数の標準正則化理論	12
2.3 スカラー値多価正則化ネットワークの表現理論	13
2.4 スカラー値多価正則化ネットワークの学習アルゴリズム	16
2.5 一般化多価正則化ネットワーク	17
2.6 分解ネットワーク	18
2.6.1 代数方程式の解析解の利用	19
2.6.2 一つの超曲面をなぞるためのリカレントネットワーク	19
2.6.3 すべての解を出力するリカレントネットワーク	21
2.6.4 フィードフォワードネットワークの利用	22
3 ベクトル値の多価正則化ネットワーク	22
3.1 ベクトル値多価関数の一般表現	22
3.2 2次元ベクトル値2価関数の標準正則化理論	23
3.3 2次元ベクトル値2価正則化ネットワークの表現	26
3.4 2次元ベクトル値2価正則化ネットワークの学習アルゴリズム	28
4 シミュレーションによる理論の検証	29
4.1 シミュレーション(1)	29
4.2 シミュレーション(2)	30
4.3 シミュレーション(3)	30
5 まとめ	31
謝辞	35

あらまし

正則化ネットワーク (Regularization Network, RN) は、「学習 = 関数近似」という立場から、Poggio & Girosi によって提案された学習ネットワークである。RN は、標準正則化理論に基づいて導出される。本論文では、非線形システムの逆モデル学習を可能にするため、RN を拡張し、多価写像を例から直接学習する多価正則化ネットワーク (Multi-Valued Regularization Network (MVRN)) を提案する。MVRN は、多価標準正則化理論 (Multi-Valued Standard Regularization Theory (MVSRT)) に基づいて導出される。MVSRT は、テンソル積 (Kronecker 積) を用いた多価関数の直接表現法に基づく。MVRN では、教師データからのネットワーク結合重みパラメータの学習が、従来の正則化ネットワークと同様、連立 1 次方程式に帰着される。本理論では、教師データを 1 価の要素関数に分離するためのクラスタリングは必要ない。従来の RN と同様、MVRN を特殊化したり、教師データ数よりも少ない基底関数を用いて、円形基底関数法 (Radial Basis Function (RBF))、一般化円形基底関数法 (Generalized RBF)、スプライン近似、HyperBF ネットワークなどが多価関数に拡張される。本理論で用いた多価関数の直接表現法は、RN に限らず、一般のネットワークに多価写像を学習させるた

めの基礎方程式として広く利用可能である。

キーワード：

正則化ネットワーク、関数近似、写像学習、正則化理論
、多価写像、計算論的学習理論、多層ネットワーク、ニュー
ーロコンピューティング、逆問題

1 はじめに

正則化ネットワーク (Regularization Network, RN) は、標準正則化理論 (Standard Regularization Theory) にもとづいて導出され、「入出力写像の関数近似 = 例からの学習」を行う 3 層構造のフィードフォワードネットワークである (16) (17)。この正則化ネットワークを特殊化することにより、円形基底関数ネットワーク (Radial Basis Function Network, RBF Network) (1), スプライン近似 (22) などの古典的な関数近似法の多くが導かれる。隠れ層ユニットの数を教師データ数よりも少なくした一般化正則化ネットワーク (Generalized Regularization Network, GRN) を特殊化すると、一般化円形基底関数ネットワーク (Generalized RBF Network) などが導かれる。最近では、sigmoid 関数を基本ユニットとしたニューラルネットワーク (NN) との関係が明らかにされつつあり、その学習アルゴリズムおよび理論的解析方法としても注目されている (6) (13) (25)。

NN や関数近似の重要な応用に、非線形システムの逆モデルの学習がある。人間の感覚情報処理も、広い意味でこの逆モデルを実現していると考えられる (15)。ビジョン、物体認識、パターン認識、運動制御における問題の多くが不良設定の逆問題である。例えば、2次元の網膜

上への投影像から外界の3次元情報を推定するビジョンは、典型的な逆問題である。これを実現する逆モデルを例からの学習によってネットワーク上に実現できれば、解析解を求めることが非常に困難であったり、生物の脳における情報処理のように記憶や経験に基づいて適応的にパラメータを調節したり、学習によって成長・進化する能力を与えたい場合に都合がよい (cf. 文献 (15) (19) (18) (26))。

ところが、画像生成過程は非線形写像である。したがって、その逆問題は、正則化が必要な不良設定問題であると同時に、たとえ、正則化を施しても、一般には解が複数 (有限個) 存在しうるため、多価写像を求める問題である。実際、コンピュータビジョンでは、陰影画像からの表面復元⁽⁷⁾や、運動からの3次元構造復元⁽¹⁰⁾をはじめとした問題で多義解の存在が知られている。人間も多義的な視知覚をする能力があることは、有名な Necker の立方体をはじめとしてよく知られた事実である。近年、複数運動や、複数物体の表面が重なり合った透明視 (Perceptual Transparency)⁽²⁰⁾において、複数の属性値を推定する逆問題の解析解が筆者によって導かれた^{(27) (28)}。そこでは、従来不可欠と見られていた緩和計算を含むデータに対するセグメンテーションやクラスタリング処理が必要ない。もし、これらの課題を、例からの学習によってネットワーク上に

逆モデルとして実現しようとするれば、多価写像を学習できるネットワークの理論が不可欠である。

本論文では、順モデルを表す写像が、連続かつある滑らかさを持つと仮定し、その逆写像を学習する問題を想定する。順システムを表す写像を $g : \mathcal{R}^m \mapsto \mathcal{R}^n$ とする。有界領域 $\mathcal{V} \subset \mathcal{R}^m$, $\mathcal{D} \subset \mathcal{R}^n$ を、それぞれ写像 g の定義域と値域とする。写像 g は連続で、 C^1 級微分可能写像と仮定する。逆写像 $g^{-1} : \mathcal{D} \mapsto \mathcal{V}$ は、入力に対して出力が一意に定まるとは限らない。出力値に無限非加算個の可能性がある場合も考えられる。この場合にも、適当な付加条件を課すことによって、加算個の解を持つ場合に帰着させることができる。無限加算個の解を持つ場合も、数学的には考えうる(例えば、 $g : x \mapsto x \sin(1/x)$ の逆写像は、 $x = 0$ の近くで無限個の零点を持つ)。本論文では、この場合は扱わず、実用上重要な有限加算個の解を持つ場合を想定する。最も簡単な $n = m = 1$ の場合の説明を図1に示す。この場合、順モデルの写像 $y \mapsto x$ は一価であるが、逆モデル $x \mapsto y$ は、0 価(値なし)、1 価、2 価、3 価写像をつなぎ合わせた写像である。この逆モデルを誤差を含む例から学習するプロセスを想定する(図2)。本論文では、この問題設定において、逆モデルの学習を実現するための基本機能である h 価写像 ($h = 1, 2, \dots$)

の学習を行う多層ネットワークの基本理論を構築する。

以下、任意の入力値 $x \in D$ に対して出力値 $y \in V$ が定義域全体で h 価である場合に議論を限定する。多価度の変化への対応に関しては、まとめにおいて、複数の実現可能なアイデアを述べる。1 価写像の逆写像には超曲面同士の交差は存在しない。したがって本論文では、交差における計算上の困難は扱わない。

RN は、NN と同様、隠れ層ユニット (= 基底関数) を増やせば、有界閉領域において任意の連続関数をいくらでも精度よく近似できる⁽⁵⁾。しかし、RN を含め、いままで提案されてきた関数近似を行う NN とその学習アルゴリズムは、不連続をもつ関数や多価関数を学習することを想定していない。不連続を持つ関数は、不連続の近傍において複数の超曲面が重なり合った多価関数として表現できる。

多価写像を学習する問題に応用可能な従来方法を次に示す。

(1) NN を用いた方法：expert network と競合学習を用いて、教師データのクラスタリングと写像学習を modular network に実現するアプローチ^{(8) (9)}。

(2) 統計的パターン認識を用いた方法：入力空間と出力空間を結合した空間で、Parzen window, k-means clustering などの混合密度推定のための古典的統計手法を用いて

教師データの分布密度関数の推定を行い、入力変数に関する周辺確率分布 (多峰関数になる) を評価する方法⁽³⁾ ⁽⁴⁾。

(1)の方法では、最急勾配法 や EM (Expectation and Maximization) 法⁽²⁾ など、一般の非線形関数に対する最適化が用いられる。局所最適解をさけるためには、よい初期値をあらかじめ必要としたり、模擬焼きなまし法⁽¹²⁾ などの確率緩和手法を用いなければならない。これらは計算量が著しく多く、パラメータの調節が難しい。(2)の方法では、上記問題点に加えて、入出力写像そのものを表現するネットワークとして実現できないという欠点がある。したがって、リアルタイム応答が必要な用途や、生体における学習のモデルとしては不適當である。複数の異なる出力値の可能性を同一入力に対応させるネットワークの、適切な学習アルゴリズムを伴う一般的な構成理論は知られていない。

本論文では、筆者が、ビジョンにおける透明視の表面復元に関して提案した多価標準正則化理論 (Multi-Valued Standard Regularization Theory, MVSRT)⁽³⁰⁾ にもとづいて、多価写像を近似する 多価正則化ネットワーク (Multi-Valued Regularization Network, MVRN) を導出する。MVSRT では、 n 次元入力空間から、 m 次元出力空間への h 価写像を直接代数的に表現する一個の基

本方程式を用いる。関数系の変換によって、この基本方程式を未知関数に関して線形形式に変換する。すると、2次元関数のエネルギー最小化問題である標準正則化問題として定式化できる。したがって、ネットワークの重み係数の学習が、従来の1価RNと同様に連立1次方程式の解法に帰着される。

以下、第2章で、MVSRTの一般論を述べる。次に、それを用いてスカラー値写像のMVRNと学習アルゴリズムを導出する。さらに、基底関数の数を学習サンプル数よりも少なくした一般化多価正則化ネットワーク(MVGRN)について述べる。第3章では、出力空間の次元数が、 $m = 2$ で多価度が $h = 2$ の場合についてベクトル値写像のMVRNを導出する。第4章では、スカラー値のMVRNについて、基底関数がGaussianの場合(Gaussian RBF)と、3次スプラインの場合のシミュレーション結果を示す。第5章で本論文をまとめる。

2 スカラー値の多価正則化ネットワーク

RNは、標準正則化理論⁽¹⁴⁾⁽²¹⁾に基づいている。RNでは、 n 次元実数空間 R^n から m 次元実数空間 R^m への滑らかな写像の学習を、教師データを近似する超曲面の生成問題と考える「学習=関数近似」の立場をとる⁽¹⁶⁾。

筆者は、文献⁽³⁰⁾において、テンソル代数を用いた多価

関数の直接表現法を用いて、多価関数の正則化問題を、未知関数に関して2次の汎関数のエネルギー最小化に帰着する標準正則化問題として定式化した。この多価標準正則化理論 (MVSRT) から、ビジョンにおける透明視問題の基本アルゴリズムである、ノイズを含んだ疎なデータからの多重表面復元の超並列アルゴリズムを導いた。

本論文で提案する MVRN もこの定式化から導かれる。原理は多重表面復元と同一である。しかし、導かれる結果は全く異なる。表面復元の場合には、正則化された関数の各点における値そのものを計算するアルゴリズムであった。RN の場合には、入出力写像を表すネットワークとその重み係数を求める。

本章では、 $m = 1$ の場合の MVRN、つまり、スカラー値の h 価関数を学習する正則化ネットワークの理論を展開する。本章の結果は、次章で述べるベクトル値の MVRN においてもその基本構成要素として用いられる。

2.1 スカラー値多価関数の一般表現

$\mathcal{R}^n \times \mathcal{R}$ 上の h 枚の超曲面 $y = f_i(\mathbf{x})$ ($i = 1, 2, \dots, h; \mathbf{x} \in \mathcal{R}^n, y \in \mathcal{R}$) を同時に表現する基本方程式は次式である。

$$\begin{aligned} \Lambda^{(h)}(\mathbf{x}, y) &\stackrel{\text{def}}{=} \prod_{k=1}^h (y - f_k(\mathbf{x})) \\ &= F_1^{(h)}(\mathbf{x}) + yF_2^{(h)}(\mathbf{x}) + \dots + y^{h-1}F_h^{(h)}(\mathbf{x}) + y^h \end{aligned}$$

$$= 0 \quad (1)$$

h 個の関数 $F_k(\mathbf{x})$ ($k = 1, 2, \dots, h$) は、問題本来の h 個の関数 $(-f_j(\mathbf{x}))$ ($j = 1, 2, \dots, h$) の基本対称式によって定義される次の関数系である。

$$\begin{aligned} F_1^{(h)}(\mathbf{x}) &\stackrel{\text{def}}{=} (-1)^h f_1(\mathbf{x}) f_2(\mathbf{x}) \cdots f_h(\mathbf{x}) \\ &\cdots, \\ F_{h-1}^{(h)}(\mathbf{x}) &\stackrel{\text{def}}{=} \sum_{i_1=1}^h \sum_{i_2=i_1+1}^h f_{i_1}(\mathbf{x}) f_{i_2}(\mathbf{x}), \\ F_h^{(h)}(\mathbf{x}) &\stackrel{\text{def}}{=} -\sum_{i=1}^h f_i(\mathbf{x}) \end{aligned} \quad (2)$$

この関数系の変換によって、未知関数に関して線形の方程式 (1) が得られる。変換 (2) の逆変換は、 y に関する h 次の代数方程式

$$\Lambda^{(h)}(\mathbf{x}, y) = 0 \quad (3)$$

の h 個の解を \mathbf{x} の関数と見なしたものである。

2.2 h 価スカラー値関数の標準正則化理論

基本方程式 (1) を用いて、 h 価関数の MVSRT を次のエネルギー汎関数の最小化問題として定式化する。

$$E^{(h)}[F_1^{(h)}, F_2^{(h)}, \dots, F_h^{(h)}] = \sum_{i=1}^N \left\{ \Lambda^{(h)}(\mathbf{x}_{(i)}, y_{(i)}) \right\}^2 + \sum_{k=1}^h \lambda_k \left\| S_k^{(h)} F_k^{(h)} \right\|^2. \quad (4)$$

教師データを $\{(\mathbf{x}_{(i)}, y_{(i)}) | i = 1, 2, \dots, N\}$ とした。 S_k および λ_k は、関数 $F_k^{(h)}$ の正則化作用素、および正則化パラ

メータを表す。正則化作用素には、微分作用素、または、擬微分作用素を用いる。 $\| \cdot \|$ は、関数空間の 2 乗ノルムで、関数 $g(\mathbf{x})$ に対して、

$$\|g\|^2 = \int_{\mathcal{R}^n} |g(\mathbf{x})|^2 d\mathbf{x} \quad (5)$$

と定義する。簡単のため、以下では、 $S_k = S$, $\lambda_k = \lambda$ ($k = 1, 2, \dots, h$) の場合を考察する。

2.3 スカラー値多価正則化ネットワークの表現理論

以下、 \mathcal{R}^n 上の h 価スカラー値関数を近似する MVRN を MVSRT から導出する。

エネルギー汎関数 (4) の最小化問題における、平衡条件 (Euler-Lagrange 方程式) は、各未知関数 $F_k^{(h)}$ ($k = 1, 2, \dots, h$) に関する変分を零とおいて求められる。

$$\frac{\delta E^{(h)}[F_1^{(h)}, F_2^{(h)}, \dots, F_h^{(h)}]}{\delta F_k^{(h)}} = 0 \quad (6)$$

実際に計算すると、次の h 個の方程式が得られる ($k = 1, 2, \dots, h$)。

$$\sum_{i=1}^N (y_{(i)})^{k-1} \Lambda^{(h)}(\mathbf{x}, y_{(i)}) \delta(\mathbf{x} - \mathbf{x}_{(i)}) + \lambda \hat{S} S F_k^{(h)}(\mathbf{x}) = 0 \quad (7)$$

$\delta(\cdot)$ は、Dirac のデルタ関数で、ベクトル $\mathbf{x} = (x_1, x_2, \dots, x_n)$ に対して、

$$\delta(\mathbf{x}) = \delta(x_1) \delta(x_2) \cdots \delta(x_n) \quad (8)$$

で定義される。作用素 \hat{S} は、 S の随伴 (adjoint) 作用素である。実定数係数の線形偏微分作用素

$$T = \sum_i a_i(\mathbf{x}) \frac{\partial^{i_1+\dots+i_n}}{\partial x_1^{i_1} \dots \partial x_n^{i_n}} \quad (9)$$

の随伴作用素 \hat{T} は、

$$\hat{T}u(\mathbf{x}) = \sum_i (-1)^{i_1+\dots+i_n} \frac{\partial^{i_1+\dots+i_n}}{\partial x_1^{i_1} \dots \partial x_n^{i_n}} (\bar{a}_i(\mathbf{x})u(\mathbf{x})) \quad (10)$$

で定義される ($i = (i_1, i_2, \dots, i_n)$)。 $a_i(\mathbf{x})$ は係数で、 $\bar{a}_i(\mathbf{x})$ は、その複素共役である。方程式 (7) は、

$$\hat{S}S F_k^{(h)}(\mathbf{x}) = -\frac{1}{\lambda} \sum_{i=1}^N (y_{(i)})^{k-1} \Lambda^{(h)}(\mathbf{x}, y_{(i)}) \delta(\mathbf{x} - \mathbf{x}_{(i)}) \quad (11)$$

と変形できる。この (偏) 微分方程式は、作用素 $\hat{S}S$ の Green 関数を用いて解ける。Green 関数 $K(\mathbf{x}, \mathbf{x}')$ は、次の関数方程式で定義される。

$$\hat{S}SK(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}') \quad (12)$$

この場合、 $\hat{S}S$ が自己随伴作用素なので、Green 関数は、対称である ($K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$)。この Green 関数を用いて偏微分方程式 (11) を解くと次式が得られる。

$$F_k^{(h)}(\mathbf{x}) = -\frac{1}{\lambda} \sum_{i=1}^N (y_{(i)})^{k-1} \Lambda^{(h)}(\mathbf{x}_{(i)}, y_{(i)}) K(\mathbf{x}, \mathbf{x}_{(i)}) \quad (13)$$

これは、各関数 $F_k^{(h)}(\mathbf{x})$ ($k = 1, 2, \dots, h$) が、 $K(\mathbf{x}, \mathbf{x}_{(i)})$ の線形結合で表されることを示している。

$$r_i^{(h)} = -\frac{1}{\lambda} \Lambda^{(h)}(\mathbf{x}_{(i)}, y_{(i)}) \quad (14)$$

とおくと、式(13)は、次の形に表現される。

$$F_k^{(h)}(\mathbf{x}) = \sum_{i=1}^N r_i^{(h)} (y_{(i)})^{k-1} K(\mathbf{x}, \mathbf{x}_{(i)}) \quad (15)$$

各 k に関して、係数 $r_i^{(h)}$ が共通である。つまり、 N 個の重み係数 $r_i^{(h)}$ によって、 h 個の未知関数すべてが決定される。こうして、入力 \mathbf{x} から、中間表現 $F_k^{(h)}(\mathbf{x})$ ($k = 1, 2, \dots, h$) への写像が得られる。

基底関数は、正則化作用素 S によって決まる。例えば、基底関数が Gaussian 円形基底関数 (Radial Basis Function, Gaussian RBF) になるのは、 S が次式で定義される擬微分作用素の場合である⁽¹⁶⁾。

$$\|Sf\|^2 = \sum_{p=0}^{\infty} \frac{\sigma^{2p}}{p!2^p} \sum_{\substack{1 \leq i_1, \dots, i_p \leq n, \\ i_1 + \dots + i_p = p}} \int_{R^n} \left(\frac{\partial^p f(\mathbf{x})}{\partial x_{i_1} \dots \partial x_{i_p}} \right)^2 d\mathbf{x} \quad (16)$$

このとき、Green 関数は、次の等方的 Gauss 関数である (A は定数)。

$$K(\mathbf{x}, \mathbf{x}') = G(\|\mathbf{x} - \mathbf{x}'\|) = A \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \quad (17)$$

スプライン近似は、次の正則化作用素を用いた場合である。

$$\|Sf\|^2 = \sum_{\substack{1 \leq i_1, \dots, i_p \leq n, \\ i_1 + \dots + i_p = p}} \int_{R^n} \left(\frac{\partial^p f(\mathbf{x})}{\partial x_{i_1} \dots \partial x_{i_p}} \right)^2 d\mathbf{x} \quad (18)$$

このとき、グリーン関数 (基底関数) は、次式である⁽¹⁶⁾。

$$K(\mathbf{x}, \mathbf{x}') = \begin{cases} \|\mathbf{x} - \mathbf{x}'\|^{2p-n} \ln \|\mathbf{x} - \mathbf{x}'\|, & 2p > n \text{ かつ } n \text{ が偶数} \\ \|\mathbf{x} - \mathbf{x}'\|^{2p-n}, & \text{その他} \end{cases} \quad (19)$$

導かれた h 価 MVRN を、図 4 に図示した。このネットワークは、2つのモジュールで構成される。入力から見て初めのモジュールは、入力 \mathbf{x} から中間表現 $F_k^{(h)}(\mathbf{x})$ ($k = 1, 2, \dots, h$) への写像を行う。それに続くモジュールは中間表現 $F_k^{(h)}(\mathbf{x})$ ($k = 1, 2, \dots, h$) から出力 $f_i(\mathbf{x})$ ($i = 1, 2, \dots, h$) への逆変換を行う。後者は、基本対称式の形に撚り合わされた情報を分解して取り出すので「分解ネットワーク」と名付ける。分解ネットワークの構成法は、第 2.6 節で述べる。

2.4 スカラー値多価正則化ネットワークの学習アルゴリズム

重みパラメータ $r_i^{(h)}$ を決める学習アルゴリズムは、式 (15) において、 \mathbf{x} を $\mathbf{x}_{(i)}$ に、添字 i を j にそれぞれ置き換えた

$$F_k^{(h)}(\mathbf{x}_{(i)}) = \sum_{j=1}^N r_j^{(h)} (y_{(j)})^{k-1} K(\mathbf{x}_{(i)}, \mathbf{x}_{(j)}) \quad (20)$$

を式 (14) に代入して得られる N 元連立 1 次方程式である

。

$$\mathbf{K}^{(h)} \mathbf{r}^{(h)} + \mathbf{z}^{(h)} = 0 \quad (21)$$

各行列とベクトルは次式で定義される。

$$(\mathbf{K}^{(h)})_{ij} = \left\{ \sum_{k=1}^h (y_{(i)} y_{(j)})^{k-1} \right\} K(\mathbf{x}_{(i)}, \mathbf{x}_{(j)}) + \lambda \delta_{ij},$$

$$\begin{aligned} \mathbf{r}^{(h)} &= (r_1^{(h)}, r_2^{(h)}, \dots, r_N^{(h)})^T, \\ \mathbf{z}^{(h)} &= (\{y_{(1)}\}^h, \{y_{(2)}\}^h, \dots, \{y_{(N)}\}^h)^T \end{aligned} \quad (22)$$

連立1次方程式(21)は、次元数が多価度 h に依存しない。これは、学習のための計算量が、写像の多価度に依存しないことを示している。

2.5 一般化多価正則化ネットワーク

連立1次方程式は、その次元数が大きいほど数値計算が悪条件になりやすい。そこで、中間層 = 基底関数の数を N より少ない M 個とし、基底関数の中心を $t_{(j)}$ ($j = 1, 2, \dots, M < N$) とおいた一般化正則化ネットワーク (Generalized Regularization Network, GRN) が考えられる⁽¹⁶⁾。GRN と同様に、MVRN において中間層 = 基底関数の数を N より少なくした多価一般化正則化ネットワーク (Multi-Valued Generalized Regularization Network, MVGRN) が考えられる。MVGRN は、MVRN と同じ基本構造を持つ。ネットワークの重みの決め方は異なる。

基底関数の中心を t_j ($j = 1, 2, \dots, M < N$) とおき、 $F_k(\mathbf{x})$ に対応する中間表現を

$$\tilde{F}_k(\mathbf{x}) = \sum_{j=1}^M \tilde{r}_{k,j} K(\mathbf{x}, t_{(j)}) \quad (23)$$

とおく。MVGRN の場合は、関数間で重みパラメータの共有はできない。 (hM) 個の未知パラメータを決定する必

要がある。パラメータの学習アルゴリズムを求めるために、この式をエネルギー汎関数 (4) に代入する (ただし、 $S_k = S$, $\lambda_k = \lambda$ とした。)。次に、 $\tilde{r}_{k,j}$ ($k = 1, 2, \dots, h; j = 1, 2, \dots, M$) に関して偏微分し、零とおく。すると、次の (hM) 元連立 1 次方程式が導かれる。

$$\tilde{K}^{(h)} \tilde{r}^{(h)} + \tilde{z}^{(h)} = 0, \quad (24)$$

各行列とベクトルは、次式で定義される。

$$\begin{aligned} \tilde{K}^{(h)} &= \begin{bmatrix} D_1^T D_1 + \lambda J & D_1^T D_2 & \dots & D_1^T D_h \\ D_2^T D_1 & D_2^T D_2 + \lambda J & \dots & D_2^T D_h \\ \dots & \dots & \dots & \dots \\ D_h^T D_1 & D_h^T D_2 & \dots & D_h^T D_h + \lambda J \end{bmatrix}, \\ \tilde{r}^{(h)} &= (\tilde{r}_{1,1}, \dots, \tilde{r}_{1,M}, \tilde{r}_{2,1}, \dots, \tilde{r}_{2,M}, \dots, \tilde{r}_{h,1}, \dots, \tilde{r}_{h,M})^T, \\ \tilde{z}^{(h)} &= \left((z^{(h)})^T D_1, (z^{(h)})^T D_2, \dots, (z^{(h)})^T D_h \right)^T, \\ (D_k)_{ij} &= (y^{(i)})^{k-1} K(\mathbf{x}^{(i)}, t^{(j)}), \\ (J)_{ij} &= K(t^{(i)}, t^{(j)}), \\ z^{(h)} &= \left(\{y^{(1)}\}^h, \{y^{(2)}\}^h, \dots, \{y^{(N)}\}^h \right)^T \end{aligned} \quad (25)$$

2.6 分解ネットワーク

$F_k^{(h)}(\mathbf{x})$ ($k = 1, 2, \dots, h$) から、 $f_i(\mathbf{x})$ ($i = 1, 2, \dots, h$) への逆変換には、複数の実現方法が考えられる。式 (3) によれば、関数系の逆変換を行うネットワークには、 h 次代数方程式を解く機能が与えられればよい。

ネットワークで実現することを想定して、次の 4 通りの方法を述べる。

2.6.1 代数方程式の解析解の利用

4次以下の代数方程式には解析解が存在する。多価度 h が4以下の場合、解の公式をハードウェアイメージで実現すれば、原理的には、フィードフォワードネットワークが構成できる。ただし、3次、4次の代数方程式は解の公式が複雑すぎて現実的でない。2次代数方程式の解の公式は簡単なので、 $h=2$ の場合は、この方法が最も適している。具体的には、次式が逆変換である。

$$\begin{aligned} f_1(\mathbf{x}) &= \frac{1}{2} \left(-F_2^{(2)}(\mathbf{x}) + \sqrt{\left(F_2^{(2)}(\mathbf{x})\right)^2 - 4F_1^{(2)}(\mathbf{x})} \right) \\ f_2(\mathbf{x}) &= \frac{1}{2} \left(-F_2^{(2)}(\mathbf{x}) - \sqrt{\left(F_2^{(2)}(\mathbf{x})\right)^2 - 4F_1^{(2)}(\mathbf{x})} \right) \end{aligned} \quad (26)$$

学習の結果によっては、解が虚数になる可能性もある。この事実を用いると、写像の多価度の変化を実数解の個数で表現できる。その判定は、判別式

$$\Delta \stackrel{\text{def}}{=} \left(F_2^{(2)}(\mathbf{x})\right)^2 - 4F_1^{(2)}(\mathbf{x}) \geq 0 \quad (27)$$

の検査によって実現できる。

2.6.2 一つの超曲面をなぞるためのリカレントネットワーク

すべての解が同時に必要でなく、一つの超曲面をなぞる様に1個の出力を得られればよい場合には、代数方程式(3)に対する反復法や Newton-Raphson 法など種々の方法をネ

ネットワーク表現したものが利用できる。これは、一般にリカレント結合のネットワークになる。たとえば、Newton-Raphson 法では、

$$\frac{df(\mathbf{x})}{dt} = -\frac{\Lambda^{(h)}(\mathbf{x}, f(\mathbf{x}))}{\Lambda_y^{(h)}(\mathbf{x}, f(\mathbf{x}))} \quad (28)$$

というダイナミカルシステムを実現するか、あるいは、

$$f^{[k+1]}(\mathbf{x}) = f^{[k]}(\mathbf{x}) - \frac{\Lambda^{(h)}(\mathbf{x}, f^{[k]}(\mathbf{x}))}{\Lambda_y^{(h)}(\mathbf{x}, f^{[k]}(\mathbf{x}))} \quad (29)$$

で定義される繰り返し計算アルゴリズムを用いる。右肩の $^{[k]}$ は繰り返し回数を表す。 $\Lambda_y^{(h)}$ は、次式で定義される。

$$\Lambda_y^{(h)}(\mathbf{x}, y) \stackrel{\text{def}}{=} \frac{\partial \Lambda^{(h)}(\mathbf{x}, y)}{\partial y} = F_2^{(h)}(\mathbf{x}) + 2yF_3^{(h)}(\mathbf{x}) + \dots + (h-1)y^{h-2}F_h^{(h)}(\mathbf{x}) + hy^{(h-1)} \quad (30)$$

このネットワークを用いると、入力連続変化する場合、 h 枚のなめらかな超曲面のうちの1枚をなぞる様に出力を生成できる。どの超曲面をなぞるかは、ネットワークの初期値に依存する。この方法では、重根や虚根を持つ場合にネットワークが不安定になる可能性がある。しかし、ネットワークの計算を複素数に拡張すれば虚根の場合も破綻せずに計算できる。虚根を実際に使うか否かは別問題と考える。

2.6.3 すべての解を出力するリカレントネットワーク

入力に対して h 個すべての出力を得ることが必要な場合は、次のダイナミクスをもつ連続時間の Durand-Kerner 法⁽¹¹⁾ を実現するリカレントネットワークが適している。

$$\frac{df_j(\mathbf{x})}{dt} = -\Lambda^{(h)}(\mathbf{x}, f_j(\mathbf{x})) \left[\prod_{i=1, i \neq j}^h \{f_j(\mathbf{x}) - f_i(\mathbf{x})\} \right]^{-1}, \quad (31)$$

これは、もともとは次式で定義される繰り返しアルゴリズムである。

$$f_j^{[k+1]}(\mathbf{x}) = f_j^{[k]}(\mathbf{x}) - \Lambda^{(h)}(\mathbf{x}, f_j^{[k]}(\mathbf{x})) \left[\prod_{i=1, i \neq j}^h \{f_j^{[k]}(\mathbf{x}) - f_i^{[k]}(\mathbf{x})\} \right]^{-1}, \quad (32)$$

Durand-Kerner 法は、 $f_i(\mathbf{x})$ ($i = 1, 2, \dots, h$) に関する連立代数方程式 (2) に Newton-Raphson 法を適用したものであることが証明されている⁽¹¹⁾。入力が連続変化する場合、一度ある入力に対して解が求めれば、その後、 h 枚すべての超曲面を同時になぞる様に h 個の出力を生成できる。この方法も重根や虚根を持つ場合には適用できない。しかし、ネットワークを複素数に拡張することによって、虚根の場合も破綻せずに計算できる¹。

¹Durand-Kerner 法は、もともと、代数方程式の全複素数根を求める手法として提案された。

2.6.4 フィードフォワードネットワークの利用

別の RN を用いて逆変換の写像を近似的に実現することも考えられる。この場合、MVRN は全体が 6 層からなるフィードフォワード型ネットワークである。この方法は、入力が必ずしも連続変化しない場合に特に適している。分解ネットワークの重み係数の学習は、原問題とは独立に行うことができる。また、重根や虚根の判定条件なども学習させておくことが可能である。ただし、分解ネットワークの構造は、必要とされる近似精度から決定される。

3 ベクトル値の多価正則化ネットワーク

3.1 ベクトル値多価関数の一般表現

m 次元ベクトル値関数 $f(x)$ による入出力写像 $y = f(x)$ ($(x, y) \in R^n \times R^m$) を考える。ベクトル値関数を多価関数へ拡張する場合は、ベクトルの各成分ごとに多価に拡張するだけでは十分でない。ベクトルの成分間で、複数候補の対応関係を表せなければならない。このことを数学的に表現するために、テンソル代数を用いた多価関数の表現法を用いる。一般の h 価 m 次元ベクトル値関数の場合には、次式である⁽³⁰⁾。

$$(y - f_1(x)) \otimes (y - f_2(x)) \otimes \cdots \otimes (y - f_h(x)) = 0, \quad (33)$$

\otimes は、テンソル積 (Kronecker 積) を表す。この多価写像の基本方程式は、各教師データが、関数 $y = f_k(x)$ ($k = 1, 2, \dots, h$) のいずれか少なくとも一つを満たすという関係を数学的に厳密に表現している。一般論は複雑になるので、以下では、 $m = 2$ 、 $h = 2$ の場合について具体的に述べる。

3.2 2次元ベクトル値2価関数の標準正則化理論

$m = 2$ 、 $h = 2$ の場合のベクトル値関数の基本方程式は次式である⁽³⁰⁾。

$$(y - f_1(x)) \otimes (y - f_2(x)) = 0, \quad (34)$$

対称化により、 $f_1(x)$ と $f_2(x)$ の役割を同等にした形式では、

$$\frac{1}{2} \{ (y - f_1(x)) \otimes (y - f_2(x)) + (y - f_2(x)) \otimes (y - f_1(x)) \} = 0, \quad (35)$$

となる。各関数 $f_1(x)$ 、 $f_2(x)$ の成分関数を次式で定義する。

$$\begin{aligned} y &= f_1(x) = (f_1(x), g_1(x))^T, \\ y &= f_2(x) = (f_2(x), g_2(x))^T. \end{aligned} \quad (36)$$

すると、式 (35) は、次の3個の方程式と等価である。

$$y_1^2 - (f_1(x) + f_2(x))y_1 + f_1(x)f_2(x) = 0,$$

$$\begin{aligned}
 & y_2^2 - (g_1(\mathbf{x}) + g_2(\mathbf{x}))y_2 + g_1(\mathbf{x})g_2(\mathbf{x}) = 0, \\
 & 2y_1y_2 - (g_1(\mathbf{x}) + g_2(\mathbf{x}))y_1 - (f_1(\mathbf{x}) + f_2(\mathbf{x}))y_2 \\
 & + (f_1(\mathbf{x})g_2(\mathbf{x}) + g_1(\mathbf{x})f_2(\mathbf{x})) = 0.
 \end{aligned} \tag{37}$$

式 (37) を未知関数に関して線形の形式に変換すると、次の方程式が得られる。

$$\begin{aligned}
 & y_1^2 + F_2(\mathbf{x})y_1 + F_1(\mathbf{x}) = 0, \\
 & y_2^2 + G_2(\mathbf{x})y_2 + G_1(\mathbf{x}) = 0, \\
 & 2y_1y_2 + G_2(\mathbf{x})y_1 + F_2(\mathbf{x})y_2 + H(\mathbf{x}) = 0.
 \end{aligned} \tag{38}$$

この場合には、本来の関数の自由度 4 に比較して、1 多い 5 個の未知関数が必要である。これら 5 個の未知関数は次式で定義される。

$$\begin{aligned}
 F_1(\mathbf{x}) &= f_1(\mathbf{x})f_2(\mathbf{x}), \\
 F_2(\mathbf{x}) &= -(f_1(\mathbf{x}) + f_2(\mathbf{x})), \\
 G_1(\mathbf{x}) &= g_1(\mathbf{x})g_2(\mathbf{x}), \\
 G_2(\mathbf{x}) &= -(g_1(\mathbf{x}) + g_2(\mathbf{x})), \\
 H(\mathbf{x}) &= f_1(\mathbf{x})g_2(\mathbf{x}) + g_1(\mathbf{x})f_2(\mathbf{x}).
 \end{aligned} \tag{39}$$

未知関数の数が本来の関数の自由度よりも多いので、複数の定式化が考えられる。以下では、 $F_1(\mathbf{x})$ と $F_2(\mathbf{x})$ に関する正則化問題、 $G_1(\mathbf{x})$ と $G_2(\mathbf{x})$ に関する正則化問題、 $H(\mathbf{x})$ に関する正則化問題の合計 3 個の標準正則化問題に

帰着する方法について述べる。簡単のため、正則化作用素と正則化パラメータは、すべての関数に共通とする。

まず、次の2個の標準正則化問題より、 $H(\mathbf{x})$ 以外の関数が求められる。

$$E_F^{(2,2)}[F_1, F_2] = \sum_{i=1}^N \left\{ \left(y_{1(i)} \right)^2 + F_2(\mathbf{x}_{(i)})y_{1(i)} + F_1(\mathbf{x}_{(i)}) \right\}^2 + \lambda \|SF_1\|^2 + \lambda \|SF_2\|^2, \quad (40)$$

$$E_G^{(2,2)}[G_1, G_2] = \sum_{i=1}^N \left\{ \left(y_{2(i)} \right)^2 + G_2(\mathbf{x}_{(i)})y_{2(i)} + G_1(\mathbf{x}_{(i)}) \right\}^2 + \lambda \|SG_1\|^2 + \lambda \|SG_2\|^2. \quad (41)$$

これらは、スカラー値の2価 MVSRT を入力空間の各座標成分に関して独立に定式化したものにほかならない。

これらの正則化問題に基づいて得られる関数 $F_1(\mathbf{x})$, $F_2(\mathbf{x})$, $G_1(\mathbf{x})$, $G_2(\mathbf{x})$ をそれぞれ $\bar{F}_1(\mathbf{x})$, $\bar{F}_2(\mathbf{x})$, $\bar{G}_1(\mathbf{x})$, $\bar{G}_2(\mathbf{x})$ とする。基本関係式 (38) の第3式を用いて、関数 $H(\mathbf{x})$ に関する標準正則化問題を次式で定式化する。

$$E_H^{(2,2)}[H] = \sum_{i=1}^N \left\{ 2y_{1(i)}y_{2(i)} + \bar{G}_2(\mathbf{x}_{(i)})y_{1(i)} + \bar{F}_2(\mathbf{x}_{(i)})y_{2(i)} + H(\mathbf{x}_{(i)}) \right\}^2 + \lambda \|SH\|^2 \quad (42)$$

これは、 $\tilde{y}_{(i)} = 2y_{1(i)}y_{2(i)} + \bar{G}_2(\mathbf{x}_{(i)})y_{1(i)} + \bar{F}_2(\mathbf{x}_{(i)})y_{2(i)}$, $\tilde{f}(\mathbf{x}) = -H(\mathbf{x})$ と置き換えれば、スカラー値1価関数 $\tilde{f}(\mathbf{x})$ による関数近似のための標準正則化問題である。

3.3 2次元ベクトル値2価正則化ネットワークの表現

式(40), (41)は、2価のMVSRTであり、式(42)は、1価の標準正則化問題であるから、その解は、式(12)によって定義される作用素 \hat{S} のグリーン関数 $K(\mathbf{x}, \mathbf{x}')$ を用いた次式の形に書くことができる。

$$\begin{aligned}
 F_1(\mathbf{x}) &= \sum_{i=1}^N u_i K(\mathbf{x}, \mathbf{x}_{(i)}), \\
 F_2(\mathbf{x}) &= \sum_{i=1}^N u_i y_{1(i)} K(\mathbf{x}, \mathbf{x}_{(i)}), \\
 G_1(\mathbf{x}) &= \sum_{i=1}^N v_i K(\mathbf{x}, \mathbf{x}_{(i)}), \\
 G_2(\mathbf{x}) &= \sum_{i=1}^N v_i y_{2(i)} K(\mathbf{x}, \mathbf{x}_{(i)}), \\
 H(\mathbf{x}) &= \sum_{i=1}^N w_i K(\mathbf{x}, \mathbf{x}_{(i)}) \quad (43)
 \end{aligned}$$

u_i, v_i, w_i は、重みパラメータである。これらの式は、5個の関数 F_1, G_1, F_2, G_2, H が、これらにすべて共通の N 個の基底関数の線形結合で表されることを示している。したがって、ここまでの計算は、5次元ベクトル値を出力とするRNで表される。

次に、代数方程式の解析解に基づいた分解ネットワークを求める。まず、関数 F_1, F_2 および G_1, G_2 から出力ベクトル \mathbf{y} の成分 y_1, y_2 それぞれの候補を求める(複号同順

)。

$$f_{\pm}(\mathbf{x}) = \frac{1}{2} \left[-F_2(\mathbf{x}) \pm \sqrt{\{F_2(\mathbf{x})\}^2 - 4F_1(\mathbf{x})} \right] \quad (44)$$

$$g_{\pm}(\mathbf{x}) = \frac{1}{2} \left[-G_2(\mathbf{x}) \pm \sqrt{\{G_2(\mathbf{x})\}^2 - 4G_1(\mathbf{x})} \right] \quad (45)$$

すると、2価出力ベクトル値として、

$$\begin{aligned} \mathbf{f}_1(\mathbf{x}) &= (f_+(\mathbf{x}), g_+(\mathbf{x}))^T \\ \mathbf{f}_2(\mathbf{x}) &= (f_-(\mathbf{x}), g_-(\mathbf{x}))^T \end{aligned} \quad (46)$$

および、

$$\begin{aligned} \mathbf{f}_1(\mathbf{x}) &= (f_+(\mathbf{x}), g_-(\mathbf{x}))^T \\ \mathbf{f}_2(\mathbf{x}) &= (f_-(\mathbf{x}), g_+(\mathbf{x}))^T \end{aligned} \quad (47)$$

の2組の可能性がある(ただし、便宜上、 $f_1(\mathbf{x}) > f_2(\mathbf{x})$ と仮定した)。各点 \mathbf{x} で、

$$|H(\mathbf{x}) - f_1(\mathbf{x})g_2(\mathbf{x}) - g_1(\mathbf{x})f_2(\mathbf{x})| \quad (48)$$

が小さくなるほうを選ぶスイッチの働きをするゲートネットワークを設けて、これら2組の可能性から最終的な2価ベクトル値を求めることができる。

図5は導かれたネットワークを図示したものである。ネットワークの重みの決め方は次節で述べる。

3.4 2次元ベクトル値2価正則化ネットワークの学習アルゴリズム

スカラー値のRN, MVRNの学習アルゴリズムを応用すれば、2次元ベクトル値2価MVRNの学習は、次の3個の N 元連立1次方程式に帰着される。

$$\mathbf{K}_F \mathbf{u} + \mathbf{z}_F = 0, \quad (49)$$

$$\mathbf{K}_G \mathbf{v} + \mathbf{z}_G = 0, \quad (50)$$

$$\mathbf{K}_H \mathbf{w} + \mathbf{z}_H = 0 \quad (51)$$

各行列とベクトルは次式で定義される。

$$\begin{aligned} (\mathbf{K}_F)_{ij} &= \{1 + y_{1(i)}y_{1(j)}\} K(\mathbf{x}_{(i)}, \mathbf{x}_{(j)}) + \lambda\delta_{ij}, \\ \mathbf{u} &= (u_1, u_2, \dots, u_N)^T, \\ \mathbf{z}_F &= \left(\{y_{1(1)}\}^2, \{y_{1(2)}\}^2, \dots, \{y_{1(N)}\}^2 \right)^T \end{aligned} \quad (52)$$

$$\begin{aligned} (\mathbf{K}_G)_{ij} &= \{1 + y_{2(i)}y_{2(j)}\} K(\mathbf{x}_{(i)}, \mathbf{x}_{(j)}) + \lambda\delta_{ij}, \\ \mathbf{v} &= (v_1, v_2, \dots, v_N)^T, \\ \mathbf{z}_G &= \left(\{y_{2(1)}\}^2, \{y_{2(2)}\}^2, \dots, \{y_{2(N)}\}^2 \right)^T \end{aligned} \quad (53)$$

$$\begin{aligned} (\mathbf{K}_H)_{ij} &= K(\mathbf{x}_{(i)}, \mathbf{x}_{(j)}) + \lambda\delta_{ij}, \\ \mathbf{w} &= (w_1, w_2, \dots, w_N)^T, \\ (\mathbf{z}_H)_i &= -\left(2y_{1(i)}y_{2(i)} + \bar{G}_2(\mathbf{x}_{(i)})y_{1(i)} + \bar{F}_2(\mathbf{x}_{(i)})y_{2(i)} \right) \end{aligned} \quad (54)$$

学習手順は、次の通り。まず、方程式 (49) と (50) を解いて、関数 F_1, F_2, G_1, G_2 を学習する。次に、その結果を用いて、式 (43) の $H(x)$ 以外の RN を構成する。この RN を用いて、式 (54) の z_H を計算する。最後に、方程式 (51) を解いて、 w を求め、関数 H を学習する。

4 シミュレーションによる理論の検証

MVRN は、正則化作用素つまり基底関数の選択で極めて広い範囲の近似手法が導かれる。その選択には、問題固有の性質を反映しなければならない。本論文は、特定の応用は想定せず、基礎理論の展開が主目的である。本章では、よく知られた関数近似法である、Gaussian RBF と 3 次スプラインの MVRN について、理論検証のために行ったシミュレーション結果を示す。

4.1 シミュレーション (1)

Gaussian RBF の MVRN について、2 入力、1 出力、2 価写像を学習させる実験を行った (図 6)。これは、次の sigmoid 型の関数を 2 つ重ねて得られる 2 価写像を $(x_1, x_2) \in [0, 1] \times [0, 1]$ なる領域において近似したものである。

$$\begin{aligned} f_1: & \quad (x_1, x_2) \mapsto \frac{0.6}{1 + \exp(-15(x_1 - 0.5))} + 0.35 \\ f_2: & \quad (x_1, x_2) \mapsto \frac{0.6}{1 + \exp(-15(x_1 - 0.5))} + 0.10 \end{aligned} \quad (55)$$

なお、分解ネットワークは、解析解 (26) を用いた。

4.2 シミュレーション (2)

Gaussian RBF を用いた n 入力スカラー値出力の h 価 MVRN について、CM-5 上にアルゴリズムをインプリメントした。分解ネットワークには、Durand-Kerner 法のリカレントネットワークを想定した繰り返し公式 (32) を用いた。

図7は、多価度 $h = 3$ の場合について、教師データと学習結果を示したものである。同図 (a) は、3本の直線上にそれぞれ 500 個のデータ点を平均 0, 標準偏差 0.02 の Gaussian ノイズを加えてランダムに発生させたものである。同図 (b) は、3 価 MVRN による近似写像のプロットである。

4.3 シミュレーション (3)

1 次元入力、スカラー値出力の 2 価 MVRN において、正則化作用素を式 (18) の $p = 2$ とした場合である 3 次スプラインを用いた。このとき、基底関数は、 $K(x, x') = |x - x'|^3$ である。分解ネットワークは、解析解 (26) を用いた。

図8(a) は、シミュレーションに用いた合計 50 点の教師データである。教師データは、次の 2 個の関数上にノイズ

を重畳してランダムに発生させた。

$$\begin{aligned} f_1: & \quad x \mapsto 0.5 \sin(2\pi x) \\ f_2: & \quad x \mapsto 0.5 \sin(2\pi x) + x \end{aligned} \quad (56)$$

同図 (b) は、学習された写像である。少数のまばらなサンプルデータからでも 2 価関数が正しく学習されている。同図 (c) は、比較のために作成したものである。同じデータに対して、1 価の 3 次スプラインを用いた。

5 まとめ

標準正則化理論に基づいた「学習 = 関数近似」の理論である正則化ネットワーク (RN) を拡張し、多価写像を近似する多価正則化ネットワーク (MVRN) を提案した。MVRN は、多価標準正則化理論 (MVSRT) に基づいて構築された。MVSRT は、多価関数の直接代数表現に基づいて標準正則化理論を拡張して得られた。MVRN においては、任意の多価度 h について、教師データ数と同一次元の連立 1 次方程式を解くだけで、 h 価写像の近似を行う多層ネットワークが構成できる。また、中間層 = 基底関数の数を学習サンプル数よりも少なくした一般化正則化ネットワーク (GRN) の多価関数への拡張である多価一般化正則化ネットワーク (MVGRN) の理論も導いた。これらにより、ビジョンをはじめ、逆モデルが多価写像になる様な、

生体あるいはそれを工学的に模した情報処理課題における学習ネットワークの構成が可能になる。ただし、本論文で述べたのは、あくまでも基本理論であり、実際の応用のためには、種々の付帯処理が必要である。

以下に、今後の課題を列挙する。

(1) 逆モデルの学習では、写像の多価性の他に、写像のヤコビアンが零になる特異点の問題がある。一般の多価写像では、超曲面同士が交差する場合も起こりうる。これらの特異点を正しく扱う方法を確立する必要がある。

(2) 教師データがもつ誤差に対する学習の最適性をより精密に定式化する必要がある。本論文では、誤差評価は、多価関数の基本方程式の2乗ノルムを用いた。統計的に最適なノルムを学習する重み付き最小2乗法に発展させる必要がある。

(3) 多価一般化正則化ネットワーク (MVGRN) においては、基底関数の中心と形状が固定された場合の理論を展開した。これらも学習すると、より高精度な写像が、より少ない基底関数で実現できる。入力空間の次元が大きい場合の「次元の呪い (curse of dimensionality)」の問題もこれによって緩めることができる。この拡張は、HyperBF ネットワーク⁽¹⁶⁾の学習アルゴリズムやEMアルゴリズム⁽²⁾を流用できる。この問題は、加法スプラインや ridge 近似に

よっても解決できる⁽⁶⁾。

(4) MVRN では、関数系の置き換えによって、一つのネットワークで多価関数を近似した。しかし、目的によっては、複数の一価 RN(または NN) を用意し、各 RN(または NN) にそれぞれ 1 枚の超曲面を学習させたい場合もある。この場合にも基本関係式 (33) を学習アルゴリズムの導出に用いることができる。例えば、この基本関係式の 2 乗誤差を誤差関数として、学習アルゴリズムが導びかれる。この場合、式 (33) と数学的に等価な次式を用いることもできる。

$$\|y - f_1(x)\| \|y - f_2(x)\| \cdots \|y - f_h(x)\| = 0 \quad (57)$$

この方程式は、方程式 (33) において両辺の 2 乗ノルムをとったものに等価である。本式の方が、計算の簡単さの点で多少有利である。RN を用いた場合、この定式化から導かれる学習アルゴリズムは、係数が相互に依存しあった h 組の連立 1 次方程式となる。この場合、単純な数値計算によって最適解が得られる保証はない。しかし、従来のモジュラーネットワークの学習方法よりも見通しがよい。

(5) 多価関数の基本方程式 (33) と (57) は、本来、特定のネットワーク構造や学習アルゴリズムとは独立である。その応用は、RN やフィードフォワード型 NN に限らない。相互結合型 NN の学習原理として用いることも考えられ

る。

(6) 本論文では、定義域全体で多価度が一定である場合を扱った。多価度が変化する場合には、これを表現する手段が必要である。次の2通りの方法が考えられる。(i) 分解ネットワークの実数解の個数で表現する方法。(ii) すべての出力値から、実際に意味のある出力値を選択するゲートネットワークを置く方法。これらは、ネットワークの表現として可能な事は明白である。しかし、そのための学習アルゴリズムは今後の研究課題とする。

(7) MVRN の学習には、教師データのクラスタリングは含まれていない。しかし、学習された写像を用いれば、教師データを各超曲面に分離するためのクラスタリングは極めて容易である。出力空間において、教師データを最も近い値を出力する要素関数へ「一撃的に」クラスタリングすることができる。この過程は、ネットワーク構造の面から見ると基底関数のグループ化、あるいは、ネットワークの分裂と見ることもできる。これは、成長、増殖するネットワークのモデルとしても興味深い。

本論文の主眼は写像学習ネットワークにあった。しかし、本論文の理論は、多価関数によるデータ近似を必要としていた諸分野に新手法を提供する。

謝辞

本研究を始めるにあたり有益な討論をしていただいた、MIT Center for Biological and Computational Learning の Tomaso Poggio 教授、Federico Girosi 博士に感謝します。また、ご討論頂いた 佐藤雅昭博士、安藤広志博士、水谷博之氏、早川秀樹氏、五十嵐治一博士に感謝します。シミュレーションプログラム作成にご協力頂いた、株式会社 CSK 吉川徹氏、豊橋技科大 Edgardo S. Curog 氏に感謝します。最後に、本研究の機会を与えていただいた ATR 人間情報通信研究所 東倉洋一社長、赤松茂室長に感謝します。

文 献

- (1) Broomhead DS, Lowe D (1988) Multivariable functional interpolation and adaptive networks. *Complex Systems* 2: 321-355
- (2) Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *J Roy Statist Soc* 39(1): 1-38
- (3) Duda RO, Hart PE (1973) *Pattern classification and scene analysis*. John Wiley & Sons
- (4) Ghahramani Z, Jordan MI (1993) Function approximation via density estimation using an EM approach. MIT Computational Cognitive Science technical report No.9304.
- (5) Girosi F, Poggio T (1990) Networks and the best approximation property. *Biol Cybern* 63: 169-176
- (6) Girosi F, Jones M, Poggio T (June,1993) Priors, stabilizers and basis functions: from regularization to radial, tensor and additive splines. A.I. Memo No.1430, Massachusetts Institute Technology.
- (7) Horn BKP, Brooks MJ(eds.) (1989) *Shape from shading*. MIT Press, Cambridge MA

- (8) Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE (1991) Adaptive mixtures of local experts. *Neural Computation* 3: 79-87
- (9) Jacobs RA, Jordan MI (1993) Learning piecewise control strategies in a modular neural network architecture. *IEEE Trans Sys Man and Cybern*, SMC-23(2): 337-345
- (10) Kanatani K (1993) Geometric computation for machine vision. Oxford Univ Press, Oxford UK
- (11) Kerner IO (1966) Ein Gesamtschrittverfahren zur Berechnung der Nullstellen von Polynomen. *Numer Math* 8: 290-294
- (12) Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220: 219-227
- (13) Maruyama M, Girosi F, Poggio T (1992) A connection between HBF and MLP. A.I.Memo No.1291, M.I.T. Artificial Intelligence Laboratory.
- (14) Poggio T, Torre V, Koch C (1985) Computational vision and regularization theory. *Nature* 317: 314-319
- (15) Poggio T (1990) A theory of how the brain might work. *Proc Cold Harbor Symp on Quantitative Biology IV*: 899-

910

- (16) Poggio T, Girosi F (1990) Networks for approximation and learning. Proc IEEE 78(9): 1481-1497
- (17) Poggio T, Girosi F (1990) Regularization algorithms that are equivalent to multilayer networks. Science 247: 978-982
- (18) Poggio T, Edelman S (1990) A network that learns to recognize 3D Objects. Nature 343: 263-266
- (19) Poggio T, Edelman S, Fahle M (1992) Learning of visual modules from examples: A framework for understanding adaptive visual performance. CVGIP: Image Understanding 56(1): 22-30
- (20) Rosenfeld A, Jasinschi R (eds.) (1993) Special Issue on Perceptual Transparency, Spatial Vision 7(2)
- (21) Tikhonov AN, Arsenin VY (1977) Solutions of ill-posed problems. W.H.Winston, Washington DC
- (22) Wahba G (1990) Spline Models for observational data. Society for Industrial and Applied Mathematics, Philadelphia, Penn.

- (23) 伊理正夫、韓太舜 (1973) テンソル解析入門、教育出版、東京
- (24) 並木美喜雄 (1982) デルタ関数と微分方程式、岩波、東京
- (25) 宮崎明雄、山田毅 (1993-03) GRBF ネットワークとニューラルネットワークとの関係について、信学論 (A) J76-A 3: 381-389
- (26) 丸山稔、寺岡照彦、阿部茂 (1993-11) 学習ネットワーク GRBF の変形する 3 次元物体の認識能力、信学論 (D-II) J76-D-II 11: 2357-2364
- (27) 志沢雅彦、間瀬健二 (1993-05) 多重オプティカルフロー：基本拘束方程式と運動透明視・運動境界検出の統一計算理論、信学論 (D-II) J76-D-II 5:987-1005
- (28) 志沢雅彦 (1994-02) 運動立体視におけるトランスペアレンシー — 2 透視投影像における点对応からの準最適線形アルゴリズム — 信学論 (D-II) J77-D-II 2: 286-300
- (29) 志沢雅彦 (1994-07) 両眼立体視におけるトランスペアレンシーの計算理論と 2 重視差の一撃計算モデル、信学論 (D-II) J77-D-II 7: (印刷中)
- (30) 志沢雅彦 (1994-06) 標準正則化理論の多価関数への拡張 — 滑らかな多重表面の復元 —、信学論 (D-II) J77-D-II 6:

1146-1156

- (31) 志沢雅彦 (1993-12) 多価関数を近似する正則化ネットワークとその学習アルゴリズム、信学会ニューロコンピューティング研究会、信学技報 NC93-66
- (32) Shizawa M (June-July 1994) Regularization networks for approximating multi-valued functions: Learning ambiguous input-output mappings from examples. Proc. IEEE International Conference on Neural Networks, Orlando, FL, (in press).
- (33) Shizawa M (July 1994) Networks for learning inverse models of nonlinear systems —Direct approximation of many-to- h input-output mappings—. The 2nd International Symposium on Inverse Problems in Engineering Sciences (IPES-94), Osaka, Japan pp.B38-B39.

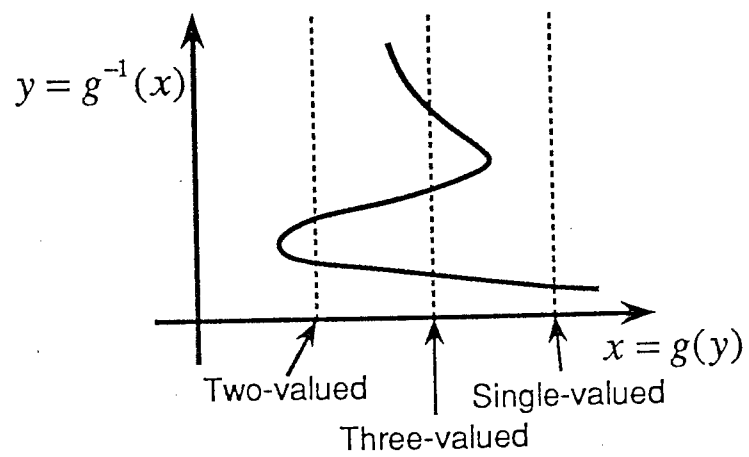


図 1 非線形写像の逆写像は、一般に多価写像である
 Fig. 1 Inverse of a nonlinear mapping is generally a multi-valued mapping.

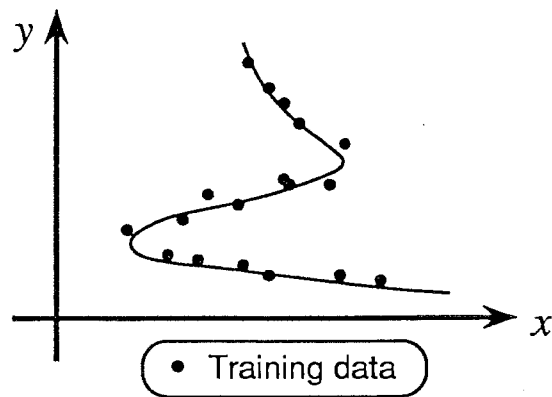


図 2 多価写像の学習問題

Fig. 2 Learning problem of multi-valued mapping.

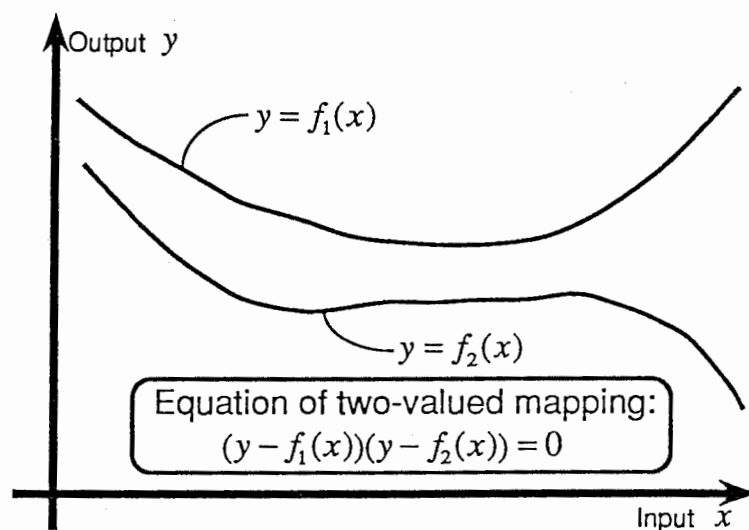


図 3 代数式による多価写像の直接表現

Fig. 3 Direct representation of a multi-valued mapping using an algebraic equation.

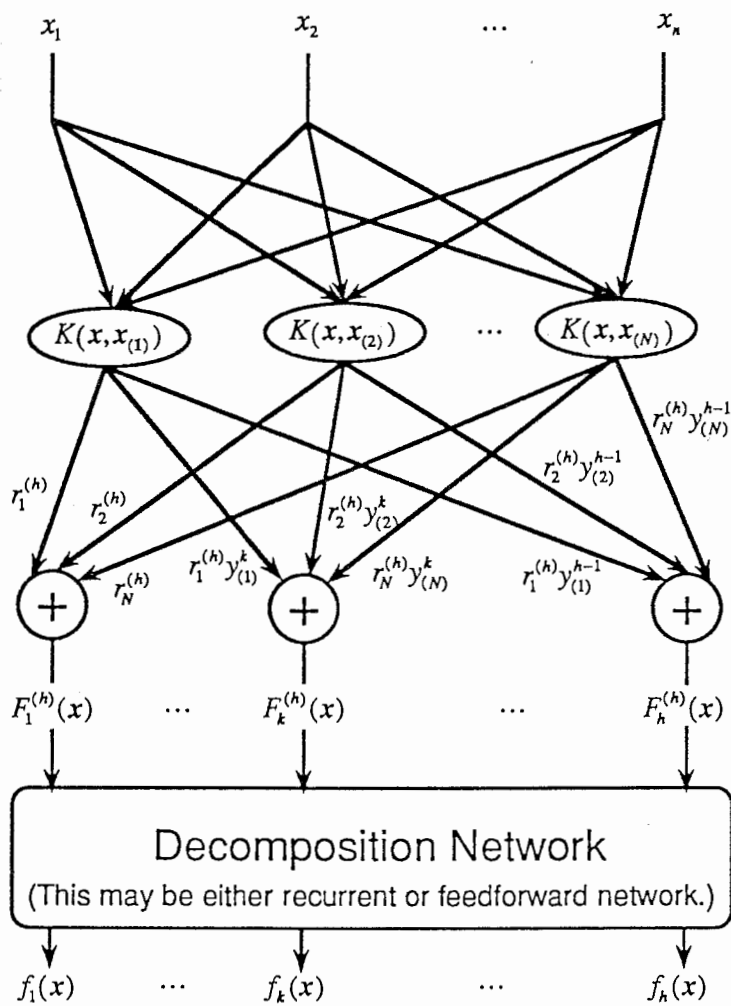


図 4 スカラー値 h 価正則化ネットワーク
 Fig. 4 Scalar-, h -valued regularization network.

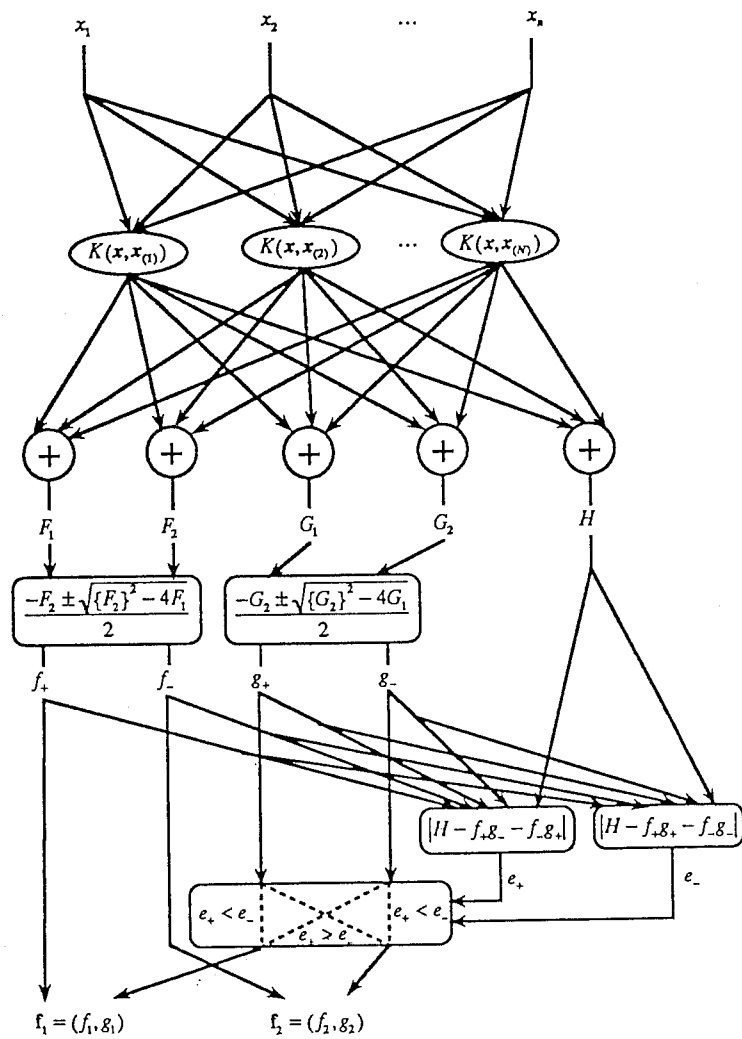
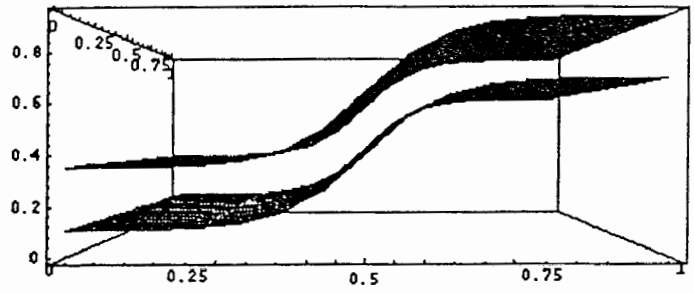
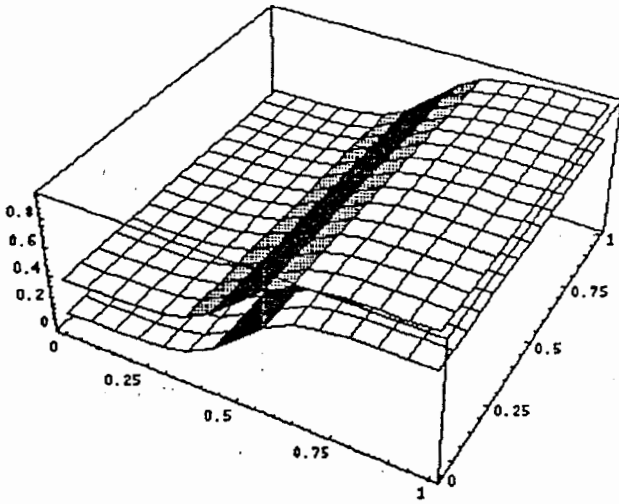
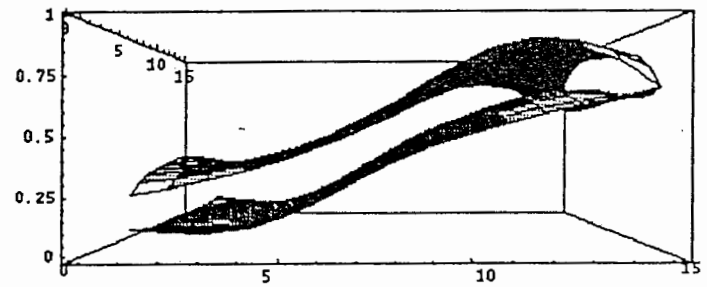
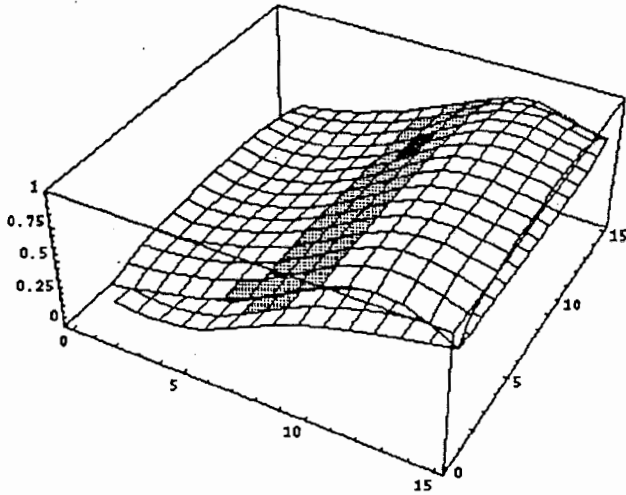


図 5 2次元ベクトル値2価正則化ネットワークの1実現例
 Fig. 5 Example of 2-D vector-, 2-valued regularization network.



(a)



(b)

図 6 シミュレーション結果 (1): Gaussian RBF による 2 次元入力
1 次元出力 2 価写像学習

Fig. 6 Simulation results (1): Learning of 2-input, 1-output, 2-valued
mapping by using Gaussian RBF version of MVRN.

(a) Model for simulation: Two-valued sigmoidal mapping were used. 200 training data were randomly generated with additive Gaussian noise of standard deviation = 0.02. (b) Result of learning by Gaussian RBF version of two-valued MVRN. The parameters for Gaussian RBF were $\sigma = 0.4$ and $\lambda = 0.008$.

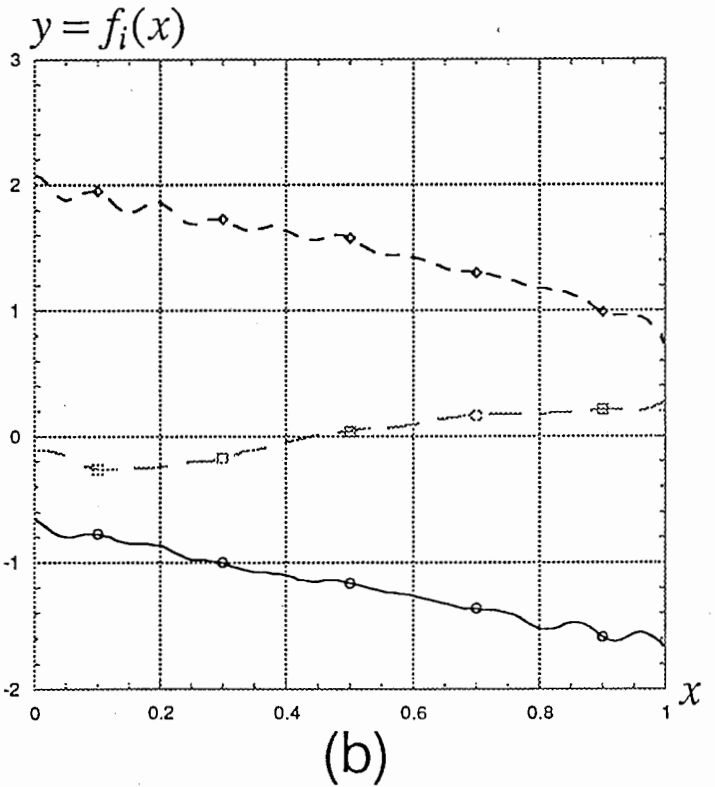
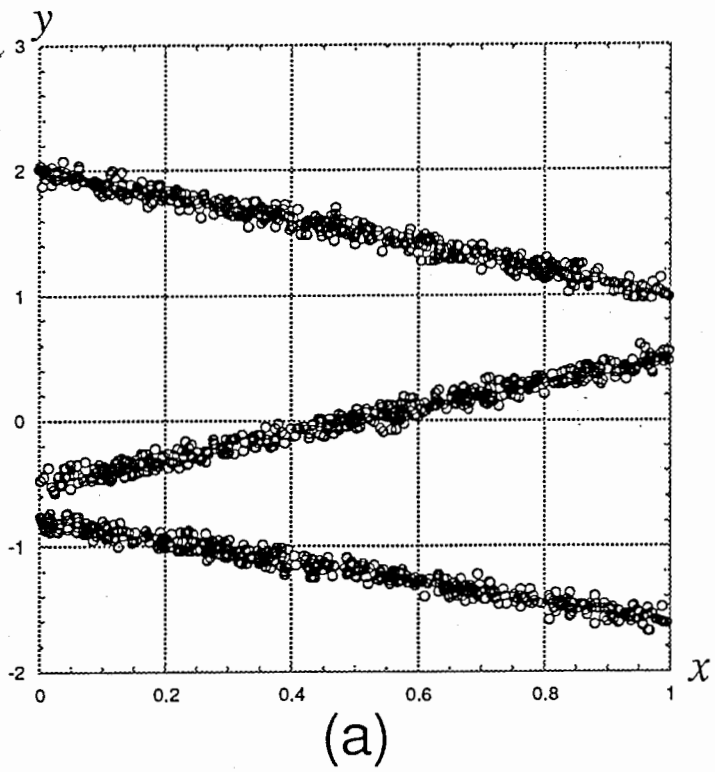
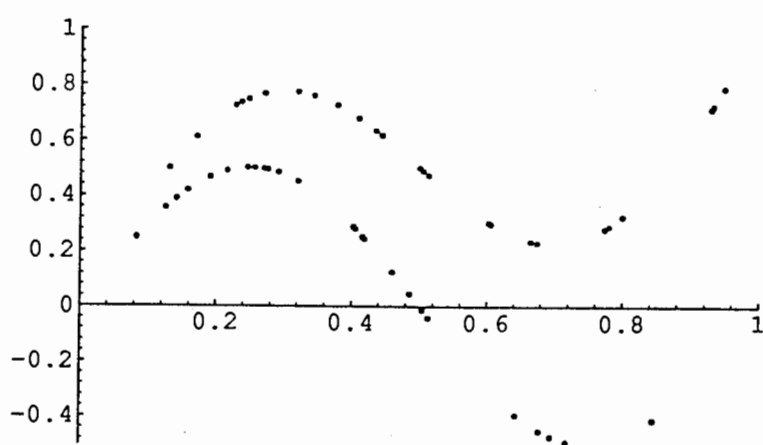


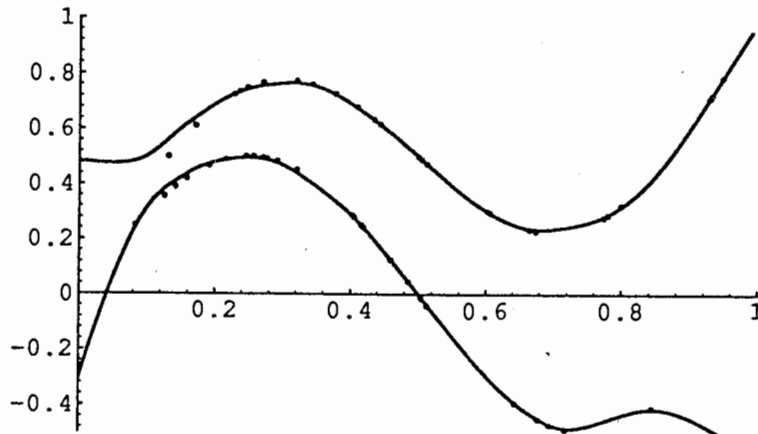
図 7 シミュレーション結果 (2): Gaussian RBF による 1 次元入力
1 次元出力 3 値写像学習

Fig. 7 Simulation results (2): Learning of 1-input, 1-output, 3-valued
mapping by using Gaussian RBF version of MVRN.

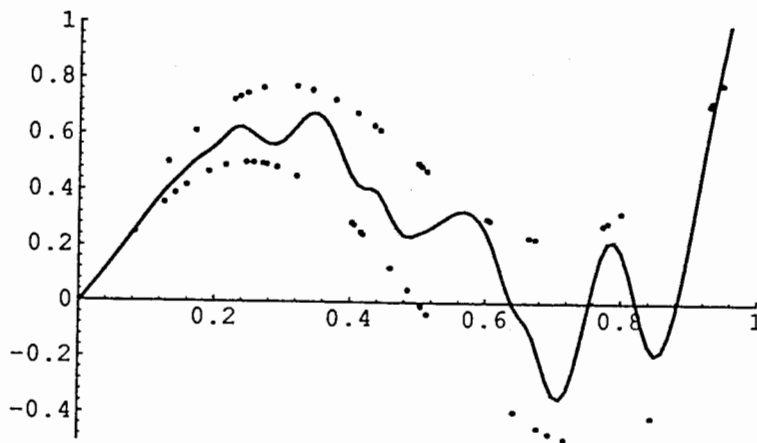
(a) Training data set with additive Gaussian noise (standard deviation = 0.02). 500 training data were generated on each line. Number of total training data was 1500. (b) Mapping learned by Gaussian RBF version of 3-valued MVRN. The parameters for Gaussian RBF were $\sigma = 0.1$ and $\lambda = 10.0$.



(a)



(b)



(c)

図 8 シミュレーション結果 (3): 3 次スプライン近似による 1 次元入力 1 次元出力 2 価写像学習

Fig. 8 Simulation results (3): Learning of 1-input, 1-output, 2-valued mapping by using cubic spline version of MVRN.

(a) Training data set with additive Gaussian noise (standard deviation = 0.01). Totally, 50 points were randomly generated. (b) Mapping learned by using cubic spline version of 2-valued MVRN. The regularization parameter was $\lambda = 0.0001$. (c) Results for comparison: Mapping learned by using cubic spline version of single-valued RN for the same training data set.