TR－H－075

多価正則化ネットワーク
ー多重表面復元と２重運動検出への応用ー

エドガルド クレグ（豊橋技術科学大学）
志沢 雅彦

1994. 4. 25

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

— 多重表面復元と2重運動検出への応用 — [1]

エドガルド クレグ[2]　　志沢 雅彦

ATR 人間情報通信研究所

[1]本テクニカルレポートは、著者の一人であるエドガルド クレグ (Edgardo S. Cureg) が、１９９４年1月１５日から１９９４年2月２５日の間に、ATR 人間情報通信研究所で行った学外実習に関する報告である。

[2]現在、豊橋技術科学大学

Final Report: Studies On
MULTI-VALUED REGULARIZATION NETWORKS

Edgardo S. Cureg

February 25, 1994

1 Introduction

Feedforward neural networks are usually employed in the learning of an input-output mapping from a given (finite) set of sample data. When the mapping to be approximated is single-valued, these networks are known to successfully perform the approximation task without altering the composition of the learning data. However, for the case of multi-valued functions, i.e. functions that assign multiple values to each point in the input space, it is necessary to cluster the data into groups that correspond to different hypersurfaces on the output space[3]. Such an additional task is computationally burdensome, especially with noisy learning data.

The approximation of multi-valued functions can be viewed as a mathematical framework that provides a solution to one of the problems encountered in computer vision, namely that of multiple transparencies, where it is necessary to reconstruct occluding surfaces (assumed to possess a certain degree of smoothness) from a given set of data[5][6].

Multi-Valued Regularization Networks (MVRN) have been proposed to solve this problem. MVRNs are derived from the Multi-Valued Standard Regularization theory (MVSRT), which is an extension of the standard regularization theory to multi-valued functions[1][3]. The idea is to represent the mapping by a single algebraic equation which is linear with respect to the coefficient functions. This representation then yields a linear Euler-Lagrange equation for the resulting energy minimization problem, thus facilitating the extension of the techniques used to derive regularization networks to the approximation of multi-valued functions.

A brief discussion of MVSRT and the derivation of MVRNs are given in the next section, while some numerical experiments that suggest the power of MVRNs in solving surface transparency problems in computational vision are described in the last section.

2 Multi-valued Regularization Networks[1][2][3]

2.1 Multi-valued Regularization Theory

We consider only scalar h-valued functions $f : \mathcal{R}^n \to \mathcal{R}$. Here, $\mathcal{R}$ is the set of all real numbers and h stands for the multiplicity of f, thus

$$\mathcal{R}^n \ni \mathrm{x} = (x_1, \cdots, x_n) \mapsto \begin{cases} y_1 \\ \vdots \\ y_h \end{cases} \quad y_i \in \mathcal{R},\ i{=}1, \cdots, h. \tag{1}$$

Let $f_i(\mathrm{x})$ be the single-valued function sending x into y_i. Then $y = f(\mathrm{x})$ can be expressed as follows:

$$(y - f_1(\mathrm{x}))(y - f_2(\mathrm{x})) \cdots (y - f_h(\mathrm{x})) = 0. \tag{2}$$

Eq. (2) asserts that each point (x, y) must lie on at least one of the hypersurfaces

$$y = f_i(\mathrm{x}),\ i{=}1, \cdots, h.$$

Expanding Eq. (2) yields

$$\prod_{i=1}^{h} (y - f_i(\mathrm{x})) = F_1(\mathrm{x}) + yF_2(\mathrm{x}) + \cdots + y^{h-1}F_h(\mathrm{x}) + y^h = 0 \tag{3}$$

where F_k denotes the kth elementary symmetric function on the f_is:

$$\begin{aligned} F_1 &= (-1)^h f_1 f_2 \cdots f_h \\ &\vdots \\ F_{h-1} &= \sum_{i_1=1}^{h} \sum_{i_2=i_1+1}^{h} f_{i_1} f_{i_2} \\ F_h &= -\sum_{i=1}^{h} f_i \end{aligned}$$

Here and below, we shall refer to the f_is as the *component functions* and to the F_ks as the *coefficient functions*.

With this representation, which is linear in the F_ks, the regularization problem for multi-valued functions becomes one of minimization of the following functional with respect to the F_ks:

$$E[F_1, F_2, \cdots, F_h] = \sum_{i=1}^{N} \{\Lambda(\mathrm{x}_{(i)}, y_{(i)})\}^2 + \sum_{k=1}^{h} \lambda_k\ ||S_k F_k(\mathrm{x})||^2 \tag{4}$$

where N denotes the number of sample data, $(\mathrm{x}_{(i)}, y_{(i)})$ the ith input-output data pair, $\Lambda(\mathrm{x}_{(i)}, y_{(i)})$ represents the left-hand side of Eq. (3). $||\cdot||$ is the square-integral norm in the function space $\mathcal{F}$ to which $S_k F_k$ belongs, which for $g \in \mathcal{F}$ is defined as

$$||g||^2 = \int_{\mathcal{R}^n} \{g(\mathrm{x})\}^2\, d\mathrm{x}$$

and S_k, λ_k are the *regularization operator* and *regularization parameter*, respectively, of F_k. S_k is usually a differential operator, with $\hat{S}_k$ as its adjoint. In the following, we will assume that $S = S_k$, $\lambda = \lambda_k$ for $k{=}1, \cdots, h$, so that Eq. (4) simplifies to

$$E[F_1, F_2, \cdots, F_h] = \sum_{i=1}^{N} \{\Lambda(\mathrm{x}_{(i)}, y_{(i)})\}^2 + \lambda \sum_{k=1}^{h} ||SF_k(\mathrm{x})||^2. \tag{5}$$

As can be seen from Eq. (5), the functional to be minimized consists of two terms: the first representing the goodness of approximation, and the second reflecting the degree of smoothness of the required solution. The trade-off between these two terms is controlled by the regularization parameter λ.

2.2 Derivation of the MVRN

Using standard techniques from variational calculus, we set each partial difference w.r.t. the unknown functions F_k (k=1, $\cdots$, h) in Eq. (4) to 0, yielding the following Euler-Lagrange equation:

$$\sum_{i=1}^{N} (y_{(i)})^{k-1}\{\Lambda(\mathbf{x}_{(i)}, y_{(i)})\}\delta(\mathbf{x}-\mathbf{x}_{(i)}) + \lambda\ \hat{S}SF_k(\mathbf{x}) = 0 \tag{6}$$

or, rearranging,

$$\hat{S}SF_k(\mathbf{x}) = -\frac{1}{\lambda}\left(\sum_{i=1}^{N} (y_{(i)})^{k-1}\{\Lambda(\mathbf{x}_{(i)}, y_{(i)})\}\delta(\mathbf{x}-\mathbf{x}_{(i)})\right) \tag{7}$$

Eq. (7) is a partial differential equation, and to solve this equation we again resort to the techniques of variational calculus: compute the integral transformation of the right-hand side of Eq. (7) with a kernel given by the Green's function of the differential operator $\hat{S}S$, which is the function K satisfying the distributional differential equation

$$\hat{S}SK(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x}, \mathbf{x}'). \tag{8}$$

The presence of the delta functions in Eq. (7) converts the integral transformations into discrete sums, and the unknown functions F_k can then be expressed as

$$F_k(\mathbf{x}) = \sum_{i=1}^{N} r_i\ (y_{(i)})^{k-1}K(\mathbf{x}, \mathbf{x}_{(i)}) \tag{9}$$

$$r_i = -\frac{1}{\lambda}\Lambda(\mathbf{x}_{(i)}, y_{(i)}). \tag{10}$$

From these equations we see that each function F_k can be written as a linear combination of the N Green's functions $K(\mathbf{x}, \mathbf{x}_{(i)})$. The properties of the Green's function K defined in Eq. (8), and therefore of the functions F_k, are determined by the differential operator S, also called the *stabilizer*. In this paper, we shall use a rotationally and translationally invariant stabilizer which leads to the following n-dimensional isotropic Gaussian Green's function:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\sigma^2}\right) \quad \forall \mathbf{x}, \mathbf{x}' \in \mathcal{R}^n. \tag{11}$$

Fig. 1 shows the architecture of the Multi-valued Regularization Network derived from Eqs. (9) and (10). The MVRN consists of two network modules, with the first module used to map the input vector $\mathbf{x}$ into $F_k(\mathbf{x})$, and the second used to map $F_k(\mathbf{x})$ into the original component functions $f_i(\mathbf{x})$. In the first module, one can verify that

1. the number of weight parameters r_i is equal to the number of data samples N and is independent of the multiplicity h of the function, and

2. the r_is are shared by the h functions F_k.

These properties are consistent with Eqs. (9) and (10). The second module, on the other hand, works as a decomposition network which solves the inverse problem of finding the value of the component functions $f_i(\mathbf{x})$ from the F_ks. Since this process involves the determination of the h (real) roots of the linear equation (3), any numerical method of solving h-degree algebraic equations can be used. However, because of the low (at most 3) h values considered in this paper, the second module reduces to a set of algebraic formulas that express the roots of (3) in terms of the coefficients F_k.

In the next subsection we outline a learning algorithm for determining the network's weight parameters r_i.

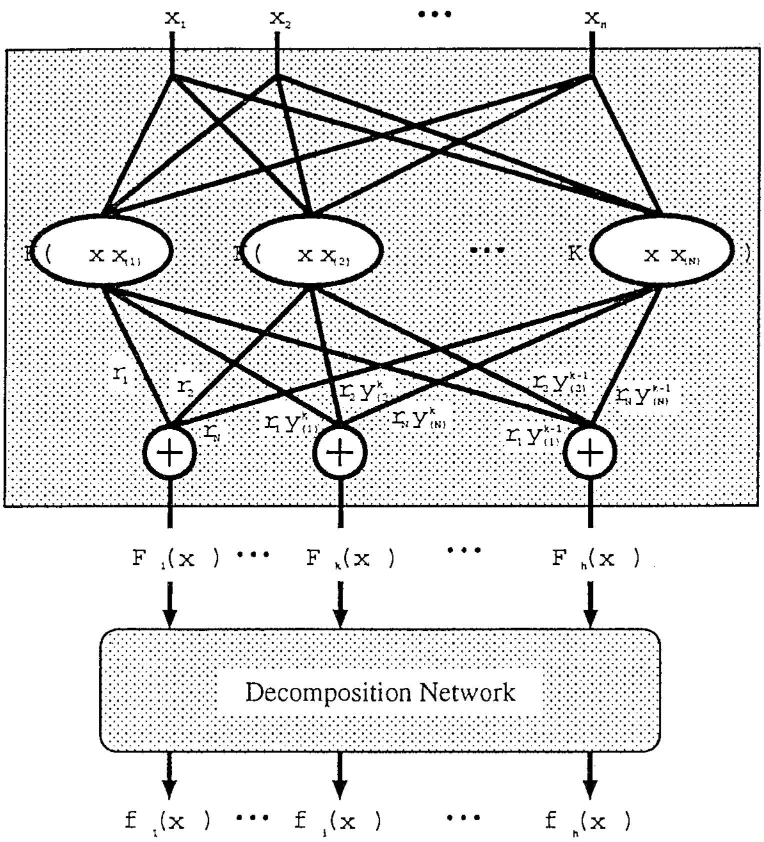


図 1: Multi-valued Regularization Network

2.3 MVRN Learning Algorithm

Making the substitutions $\mathbf{x}\rightarrow\mathbf{x}_i$, $i\rightarrow j$ in Eq. (9) and substituting in Eq. (10) we obtain, after making the necessary simplifications, the following N-dimensional linear system of equations with respect to the r_is:

$$\mathbf{K}\mathbf{r} + \mathbf{z} = 0, \tag{12}$$

where

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

$$\mathbf{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{pmatrix}, \quad \text{and} \quad \mathbf{z} = \begin{pmatrix} \{y_{(1)}\}^h \\ \{y_{(2)}\}^h \\ \vdots \\ \{y_{(N)}\}^h \end{pmatrix}.$$

Here, δ_{ij} is the Kronecker delta function.

The learning algorithm for MVRNs thus amounts to solving a system of linear equations of dimension N for the network weights r_i. This scheme is different from the learning algorithms for most of the current neural networks because all network weights are known and fixed.

3 Numerical Experiments

3.1 Example 1

Here we consider a two-valued scalar function $f : \mathcal{R} \rightarrow \mathcal{R}$ whose single-valued components f_1 and f_2 are given by

$$\begin{aligned} f_1 : \quad & x \mapsto -0.3x + 0.85 \\ f_2 : \quad & x \mapsto 0.3x + 0.15 \end{aligned}$$

Two sets of 50 x values in the interval [0, 1] were randomly generated, with the y values for the first (resp. second) set computed using f_1 (resp. f_2). A noise component $\varepsilon \sim N(0, \sigma_y^2)$, σ_y=0.01 was added to the y values. The N=100 data points used for this experiment are shown in Fig. 2.

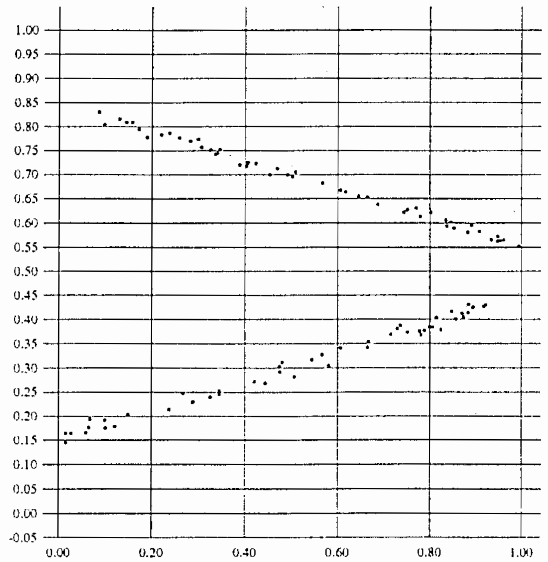


図 2: Learning Data for Example 1

For the learning phase, the regularization parameter λ was set to 0.001, and the fol-

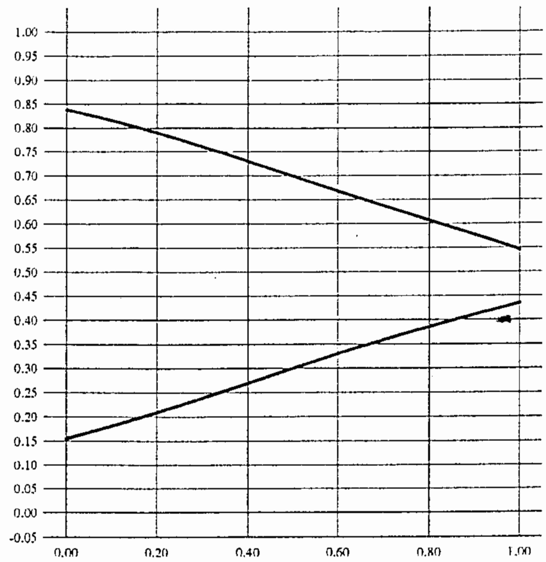


図 3: Network Output

lowing Gaussian function was used as the Green's function.

$$K(x,x') = \exp\left(-\frac{|x-x'|^2}{2\sigma^2}\right) \tag{13}$$

The value of σ for this experiment was 1.0. Fig. 3 shows the output of the network when M=500 test data $x_i = i/(M-1)$, i=0, $\cdots$, $M-1$ are used as input.

3.2 Example 2

Next we consider a two-valued mapping $f : \mathcal{R}^2 \to \mathcal{R}$ with sigmoidal component mappings f_1 and f_2:

$$f_1 : \quad (x_1, x_2) \mapsto \frac{0.6}{1+\exp(-15(x_1-0.5))} + 0.35$$

$$f_2 : \quad (x_1, x_2) \mapsto \frac{0.6}{1+\exp(-15(x_1-0.5))} + 0.10$$

for $(x_1, x_2) \in [0,1] \times [0,1]$. A 3-dimensional graph of this mapping is shown in Fig. 4.

As in the preceeding example, a Gaussian function was employed as the Green's function, with the absolute value appearing in Eq. 13 replaced by the usual norm $||\cdot||$ in $\mathcal{R}^2$. σ for this case was set to 0.4.

The procedure for training the network follows closely that of the preceeding example, with N=200 randomly generated learning input data $\mathrm{x}_{(i)} = (x_{i1}, x_{i2}) \in [0,1] \times [0,1]$ such that the corresponding output y_i is given by

$$y_i = \begin{cases} f_1(\mathrm{x}_{(i)}) + \varepsilon_i, & 1 \le i \le N/2 \\ f_2(\mathrm{x}_{(i)}) + \varepsilon_i, & N/2 < i \le N \end{cases}$$

Here, $\varepsilon \sim N(0, \sigma_y^2)$ with σ_y=0.02, and λ=0.008. Fig. 5 shows the approximated mapping using M^2=225 points

$$\left(\frac{i}{M-1}, \frac{j}{M-1}\right), \quad i,j=0,\cdots,M-1.$$

as test input data.

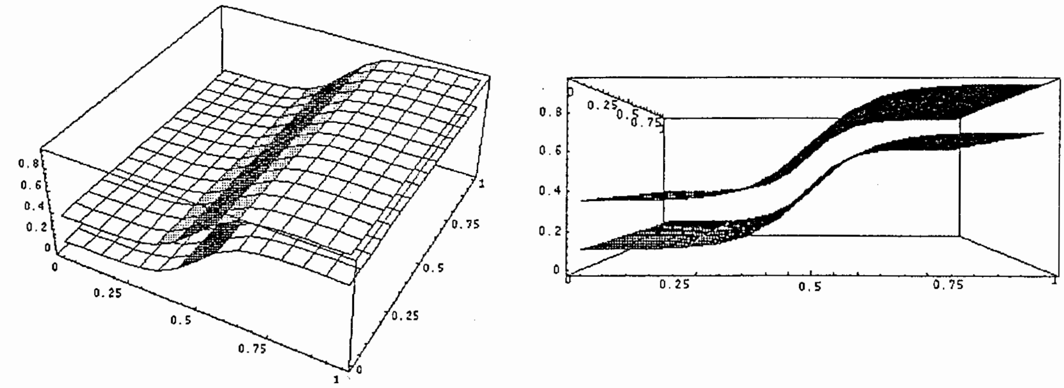


図 4: 2-valued mapping $f : \mathcal{R}^2 \to \mathcal{R}$

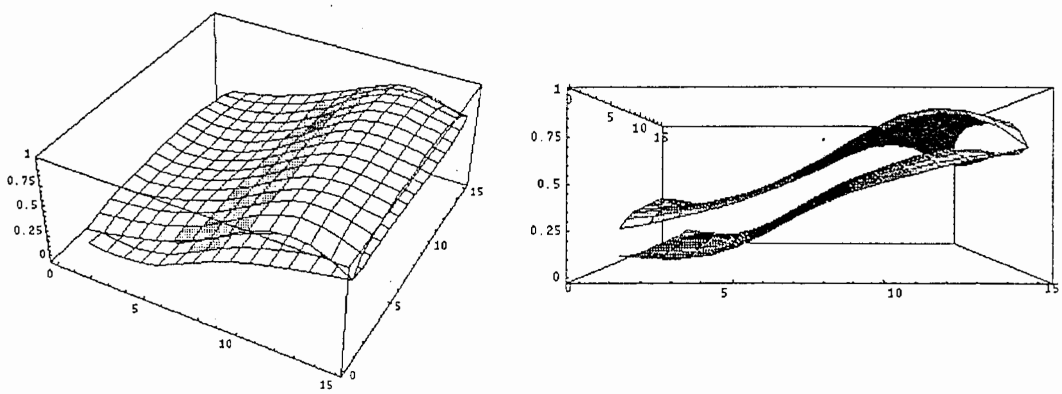


図 5: Learned mapping

3.3 Example 3

We consider the space $\mathcal{H} = \{(w_x, w_t, w'_x, w'_t) | w_x, w_t, w'_x, w'_t \in [-1, 1]\} \subset \mathcal{R}^4$. Let a two-valued mapping $f : \mathcal{H} \to \mathcal{R}$ be defined as follows:

$$(w_x, w_t, w'_x, w'_t) \mapsto \begin{cases} u_1 = -w_t/w_x \\ u_2 = -w'_t/w'_x \end{cases}$$

The variables $w_x, w_t, w'_x, w'_t, u_1$ and u_2 are related by the following system of equations:

$$u_1 u_2 w_x^2 + (u_1 + u_2) w_x w_t + w_t^2 = 0 \quad (14)$$

$$u_1 u_2 w_x'^2 + (u_1 + u_2) w'_x w'_t + w_t'^2 = 0 \quad (15)$$

By fixing the values of w'_x and w'_t, we can see the behavior of f for $(w_x, w_t) \in [-1,1] \times [-1,1]$. Letting w'_x=0.8, w'_t=0 leads to a graph shown in Fig. 6.

In using an MVRN to approximate f, we randomly generate N=576 quadruples $(w_x, w_t, w'_x, w'_t) = \mathbf{x}_{(i)}$ such that both $|w_t/w_x|$ and $|w'_t/w'_x|$ are less than or equal to 3, and assign u_1 (resp. u_2) to $\mathbf{x}_{(i)}$ if i is odd (resp. even).

The parameters for this experiment are as follows: λ=1.0, a Gaussian Green's function with σ=0.08 (as in the previous example, Eq. 13 has to be modified so that the usual norm $||\cdot||$ in $\mathcal{R}^4$ is used instead of the absolute value), and noise components $\varepsilon_u \sim N(0, \sigma_u^2)$ with σ_u=0.02, $\varepsilon_x \sim N(0,\sigma_x^2), \varepsilon_t \sim N(0, \sigma_t^2)$ with $\sigma_x = \sigma_t$=0.01, added to the u, w_x, and w_t values.

Using $M^2 = N$=576 test input data

$$\left(\frac{2i}{M-1}-1, \frac{2j}{M-1}-1, 0.8, 0\right), \quad i, j=0, \cdots, M-1$$

yields the approximated mapping whose graph is shown in Fig. 7.

4 Summary

This report is aimed at presenting the ideas behind the Multi-valued Regularization Theory (MVRT) and its network implementation, the Multi-valued Regularization Network (MVRN) which is used to approximate multi-valued mappings on $\mathcal{R}^n$. The techniques used in deriving MVRNs relied heavily on the theory of calculus of variations, starting with a direct algebraic representation of a multi-valued mapping and proceeding to solve the Euler-Lagrange equation associated with the regularization problem for this case. It was shown that the learning algorithm for MVRNs is equivalent to the solution of a system of linear equations whose dimension is equal to the number N of sample data and is independent of the multiplicity h of the mapping to be approximated. This is a remarkable property since the computational time for solving an N-dimensional system of linear equations is proportional to N^3.

In order to verify the applicability of MVRNs, several numerical simulations on the recovery of multiple surfaces from randomly generated noisy data. The results were convincing, showing that further research on the possible applications of MVRNs to other fields is worth pursuing. The estimation of double optical flows, a major theme in computational vision, can be given as an example of such an application.

Acknowledgements

The author wishes to thank Mr. Masahiko Shizawa of the 2nd Department, ATR Human Information Processing Research Laboratories, for the all the guidance and help that he extended during the author's 7-week research training at ATR. Thanks are also due to all the wonderful people of the ATR Planning Department. Finally, he wishes to thank Toyohashi University of Technology's Mr. Katsuyuki Hagiwara for providing technical support, and Prof. Shiro Usui for the chance to do research in ATR.

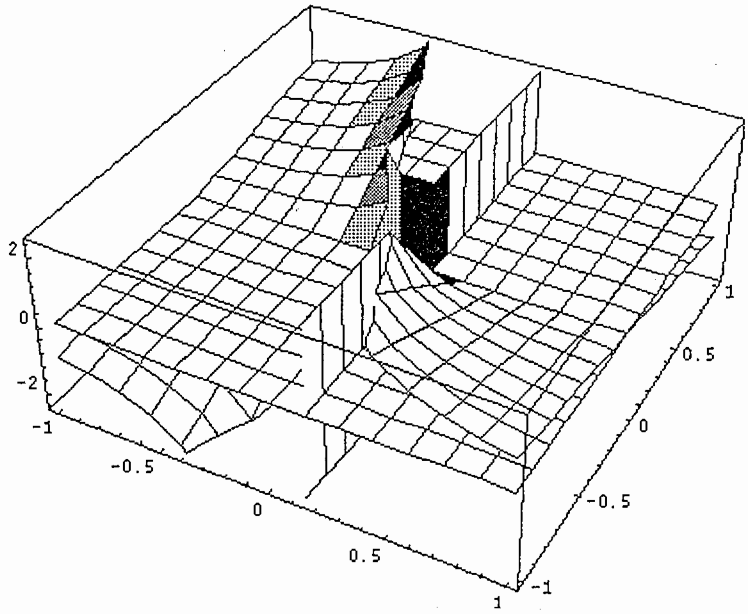


図 6: The restriction of f to $\mathcal{A}=\{(w_x, w_t, 0.8, 0)|w_x, w_t \in [-1, 1]\}$

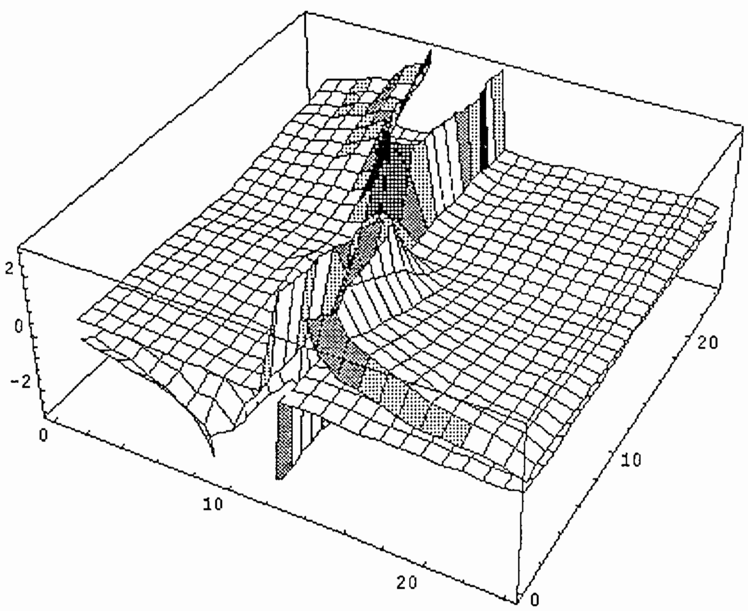


図 7: Network Output

参考文献

[1] Poggio, T., Girosi, F. "Networks for Approximation and Learning". Proceedings of the IEEE, 78(9), pp. 1481-1497, 1990.

[2] Girosi, F., Jones, M., Poggio, T. "Priors, Stabilizers and Basis Functions: from regularization to radial, tensor and additive splines". MIT Artificial Intelligence Laboratory A.I. Memo No. 1430, C.B.C.L. Paper No. 75, 1993.

[3] Shizawa, M. "Regularization Networks for Approximating Multi-valued Functions: Learning Ambiguous Input-Output Mappings from Examples". ATR Institute International.

[4] 志沢雅彦, 間瀬健二. "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". 電子情報通信学会論文誌 **J76**-D-II No.5, pp. 987-1003, 1993.

[5] 志沢雅彦. "標準正則化理論の多価関数への拡張と多重表面の復元". 電子情報通信学会信学技報 NC 93-34 (1993-09), pp. 25-32.

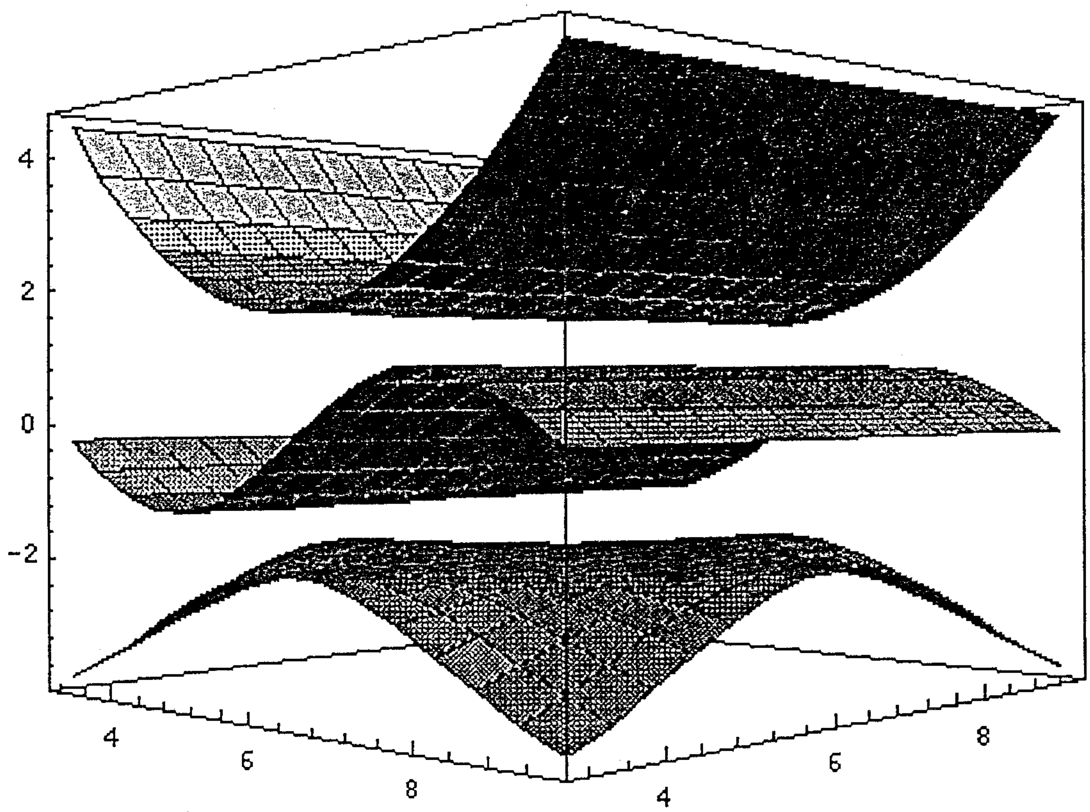
[6] 志沢雅彦. "多価関数を近似するネットワークとその学習アルゴリズム". 電子情報通信学会信学技報 NC 93-66 (1993-12), pp. 89-96.

多価正則化ネットワーク:
多重表面の復元および2重運動の検出への応用

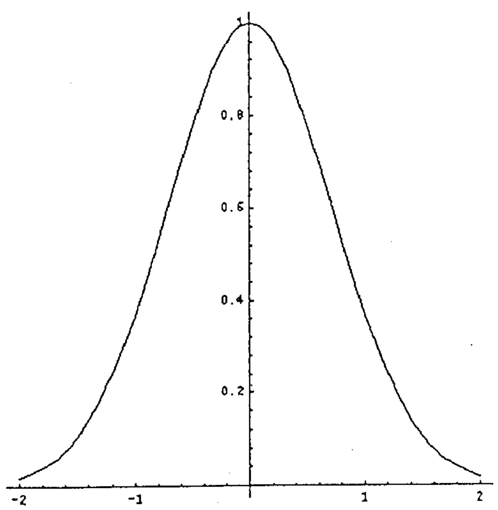
EDGARDO S. CUREG

FEB. 25, 1994

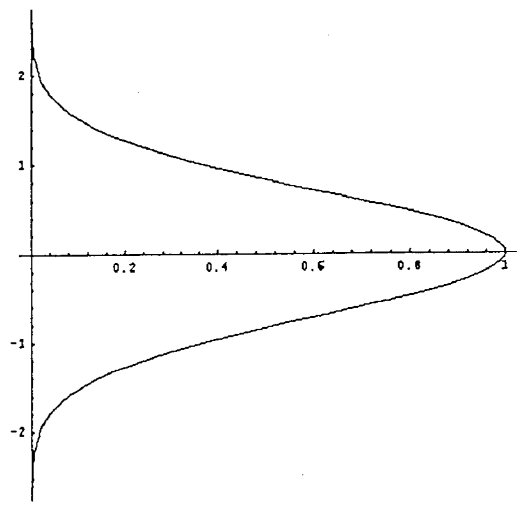
Multi-valued Functions



A MULTI-VALUED FUNCTION ON $\mathcal{R}^2$

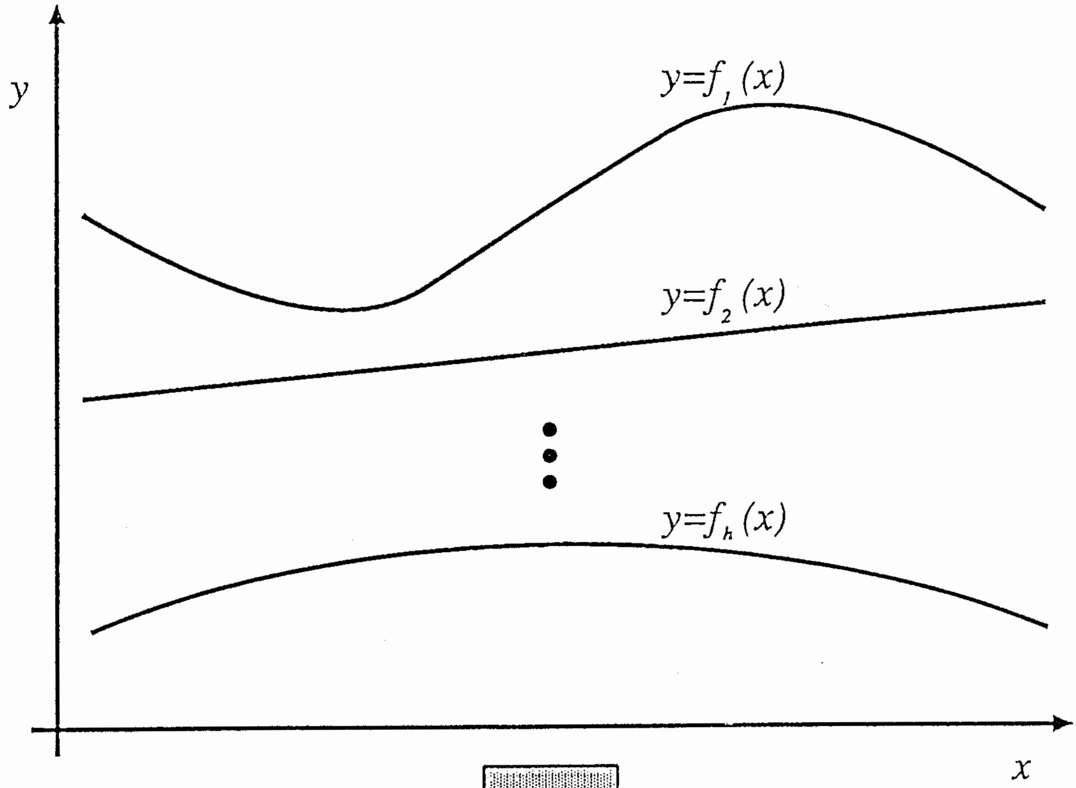


$y = f(x)$



$x = f^{-1}(x)$

MVSRT



Algebraic representation of an h-fold surface:

$$(y-f_1(\mathrm{x}))\ (y-f_2(\mathrm{x}))\ \cdots\ (y-f_h(\mathrm{x})) = 0$$

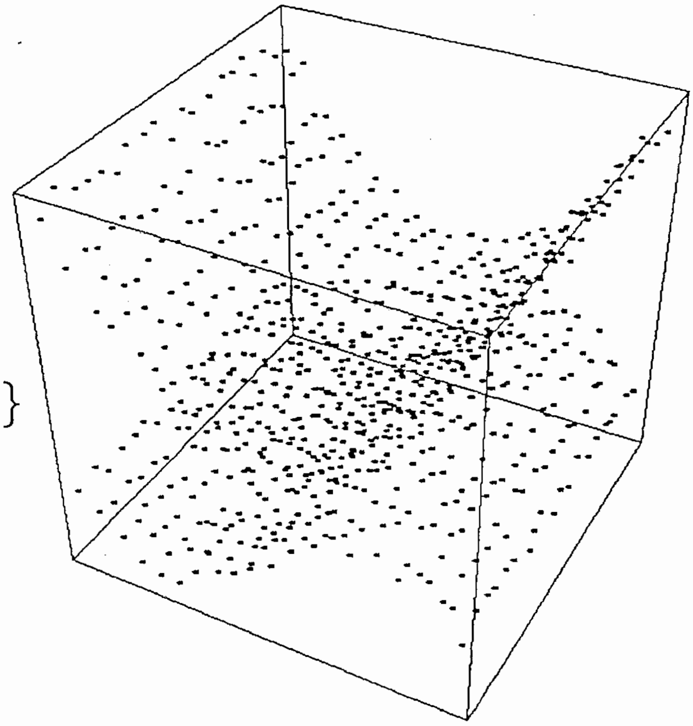
$$\begin{aligned}\prod_{i=1}^{h}(y-f_i(\mathrm{x})) &= F_1(\mathrm{x}) + yF_2(\mathrm{x}) + \cdots + y^{h-1}F_h(\mathrm{x}) + y^h \\ &= 0\end{aligned}$$

def.

$\Lambda(\mathrm{x}, y)$

$$\begin{aligned} F_1 &= (-1)^h f_1 f_2 \cdots f_h \\ &\vdots \\ F_{h-1} &= \sum_{i_1=1}^{h} \sum_{i_2=i_1+1}^{h} f_{i_1} f_{i_2} \\ F_h &= -\sum_{i=1}^{h} f_i \end{aligned}$$

MVSRT



Given: N data points

$$\{(\mathrm{x}_{(i)}, y_{(i)}) \mid i=1, \cdots, N\}$$

Regularization Problem:

Minimize the following functional w.r.t. F_k

$$E[F_1, F_2, \cdots, F_h] = \sum_{i=1}^{N} \{\Lambda(\mathrm{x}_{(i)}, y_{(i)})\}^2 + \sum_{k=1}^{h} \lambda_k \, ||S_k F_k(\mathrm{x})||^2$$

⇩

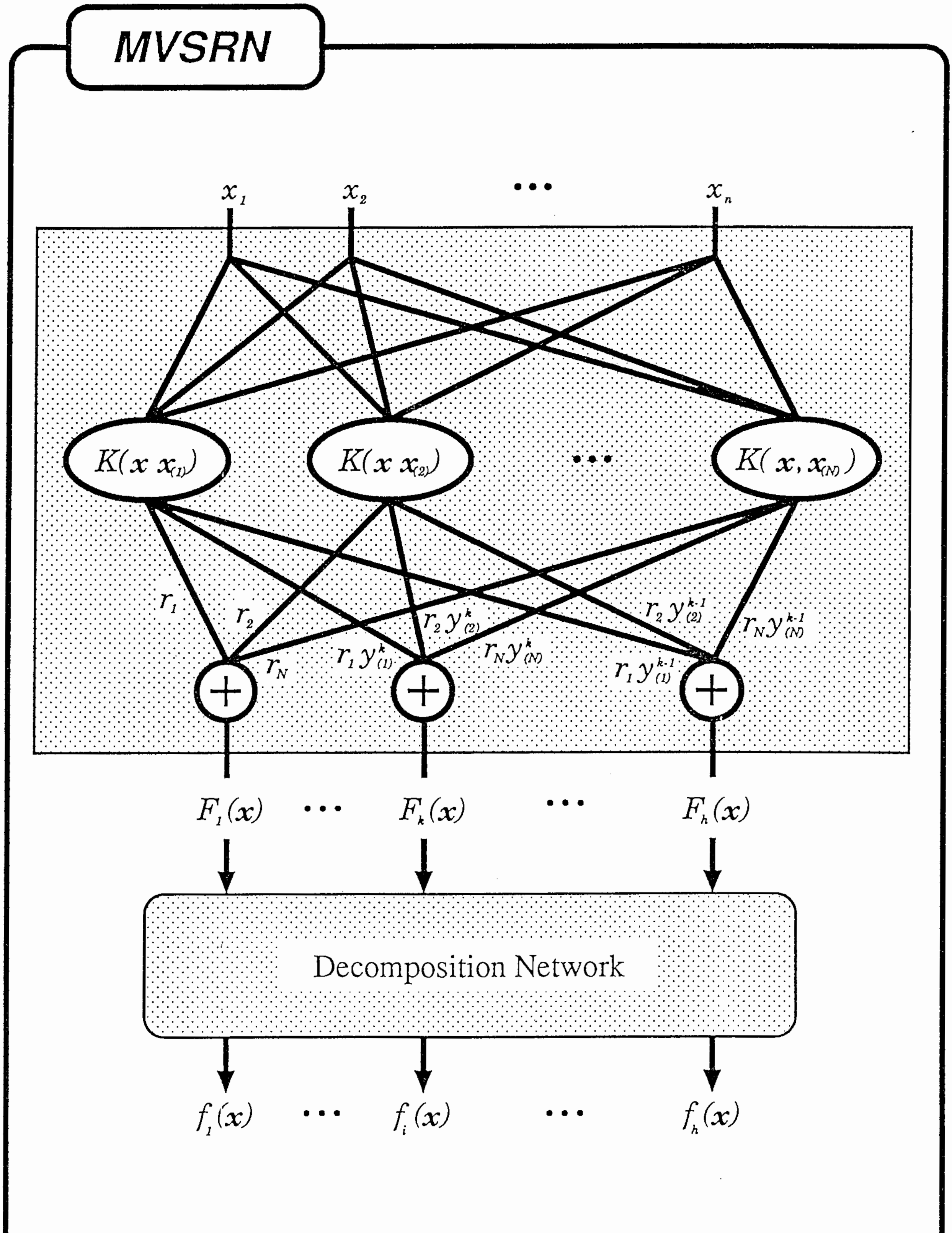
$$\sum_{i=1}^{N} (y_{(i)})^{k-1} \{\Lambda(\mathrm{x}_{(i)}, y_{(i)})\} \delta(\mathrm{x}-\mathrm{x}_{(i)}) + \lambda \, \hat{S} S F_k(\mathrm{x}) = 0$$

⇩

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

$$r_i = -\frac{1}{\lambda} \Lambda(\mathrm{x}_{(i)}, y_{(i)})$$

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



Multi-valued Standard Regularization Network

MVSRN

$$F_k(\mathbf{x}) = \sum_{i=1}^{N} r_i \left(y_{(i)}\right)^{k-1} K(\mathbf{x}, \mathbf{x}_{(i)}) \qquad (1)$$

$$r_i = -\frac{1}{\lambda} \Lambda(\mathbf{x}_{(i)}, y_{(i)}) \qquad (2)$$

Replace $\mathbf{x}$ by $\mathbf{x}_{(i)}$, i by j in (1)

Substitute in (2)

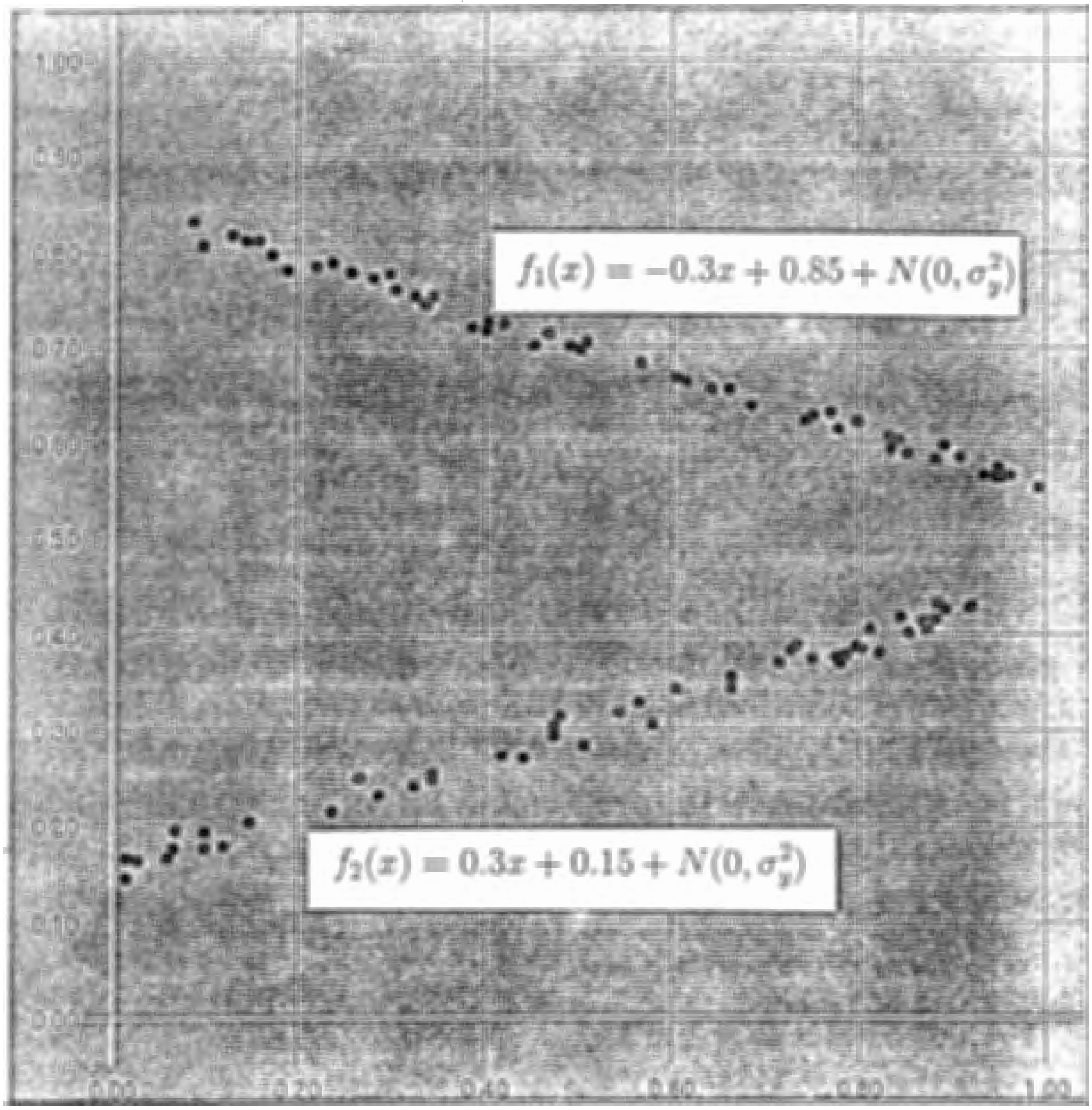
Learning algorithm for MVSRN:

Solve the following N-dimensional system of linear equations

$$\mathbf{K}\mathbf{r} + \mathbf{z} = 0$$

$$\begin{aligned}
\mathbf{K} &= (K_{ij}) = \left(\left\{ \sum_{k=1}^{h} \left(y_{(i)} y_{(j)}\right)^{k-1} \right\} K(\mathbf{x}_{(i)}, \mathbf{x}_{(j)}) + \lambda \delta_{ij} \right), \\
\mathbf{r} &= (r_1, \cdots, r_N)^T, \quad \text{and} \\
\mathbf{z} &= \left(\{y_{(1)}\}^h, \{y_{(2)}\}^h, \cdots, \{y_{(N)}\}^h \right)^T
\end{aligned}$$

Experiment No. 1



$$\sigma_y^2 = 0.01$$

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

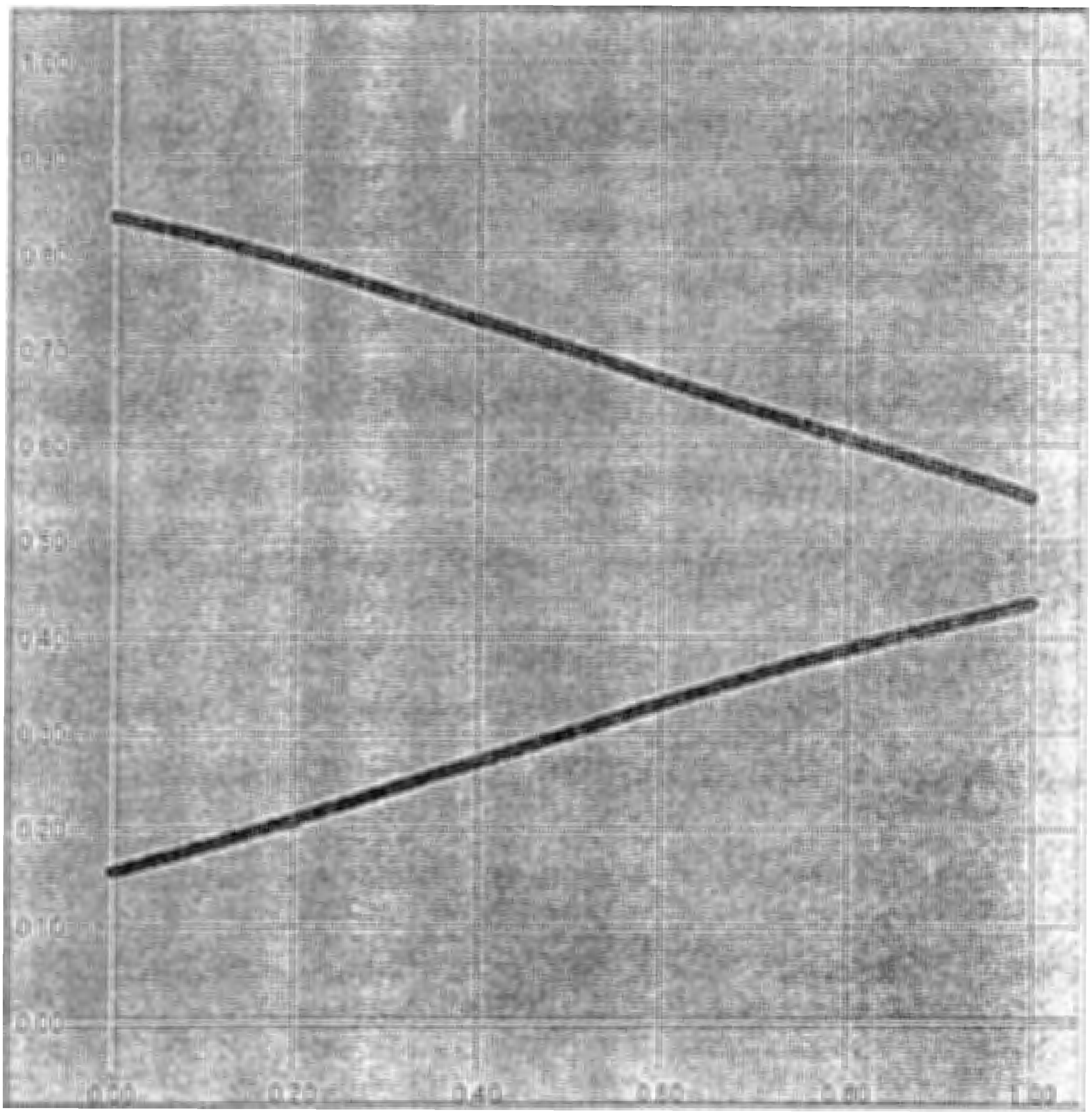
$$\sigma = 1.0$$

$$\lambda = 0.001$$

Learned Mapping

M = 500 Test Input Data:

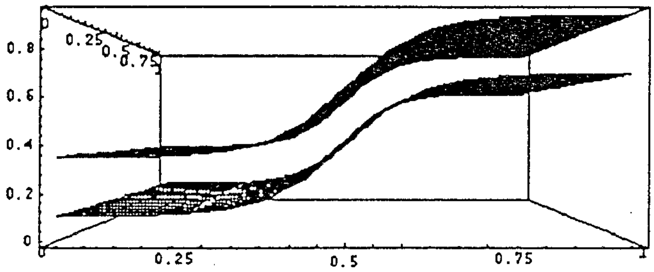
$$x_i = \frac{i}{M-1}, \quad i = 0, \cdots, M-1$$

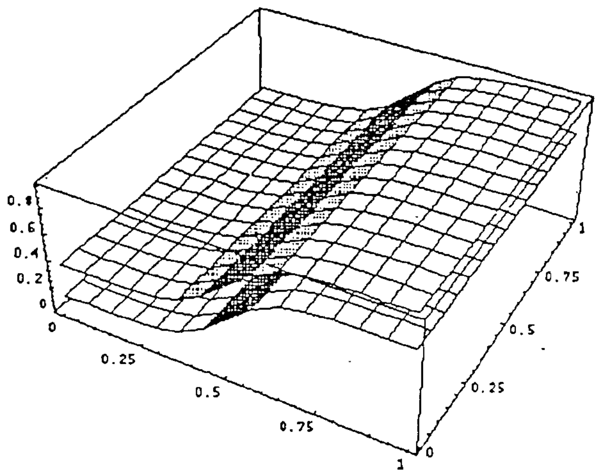


Experiment No. 2

$$f_1 : (x_1, x_2) \mapsto \frac{0.6}{1 + \exp(-15(x_1 - 0.5))} + 0.35$$

$$f_2 : (x_1, x_2) \mapsto \frac{0.6}{1 + \exp(-15(x_1 - 0.5))} + 0.10$$





$$\sigma_y = 0.02$$

$$K(\mathrm{x}, \mathrm{x}') = \exp\left(-\frac{||\mathrm{x} - \mathrm{x}'||^2}{2\sigma^2}\right)$$

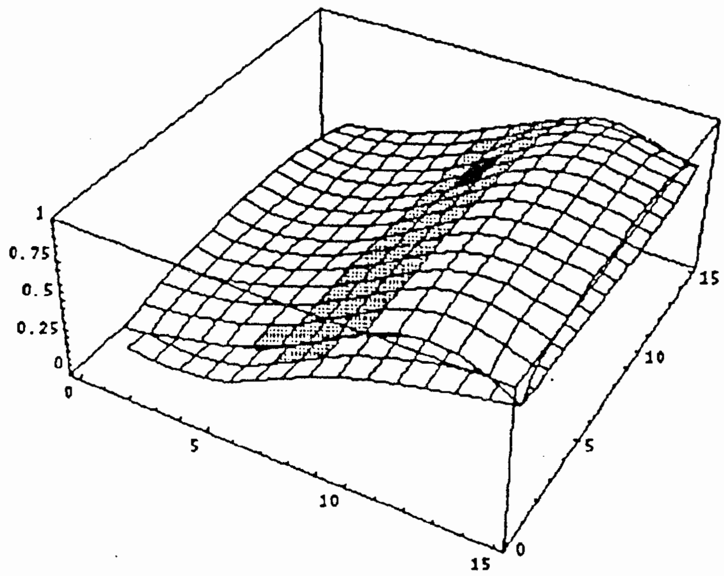
$$\sigma = 0.4$$

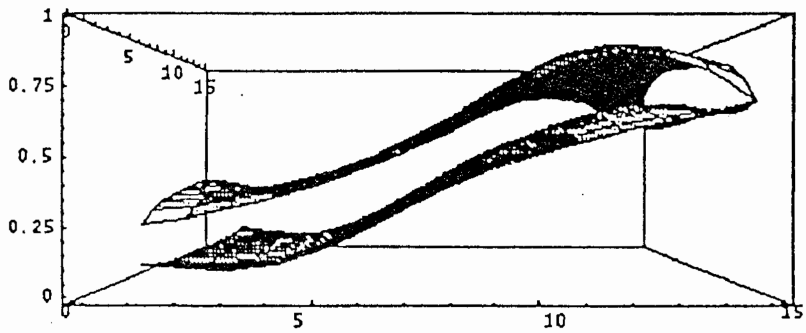
$$\lambda = 0.008$$

Learned Mapping

$M^2 = 225$ Test Input Data:

$$\left(\frac{i}{M-1}, \frac{j}{M-1}\right), \quad i,j=0,\cdots,M-1.$$





Learning of velocity detection module for motion transparency (1D motion)

(This is a toy problem.)

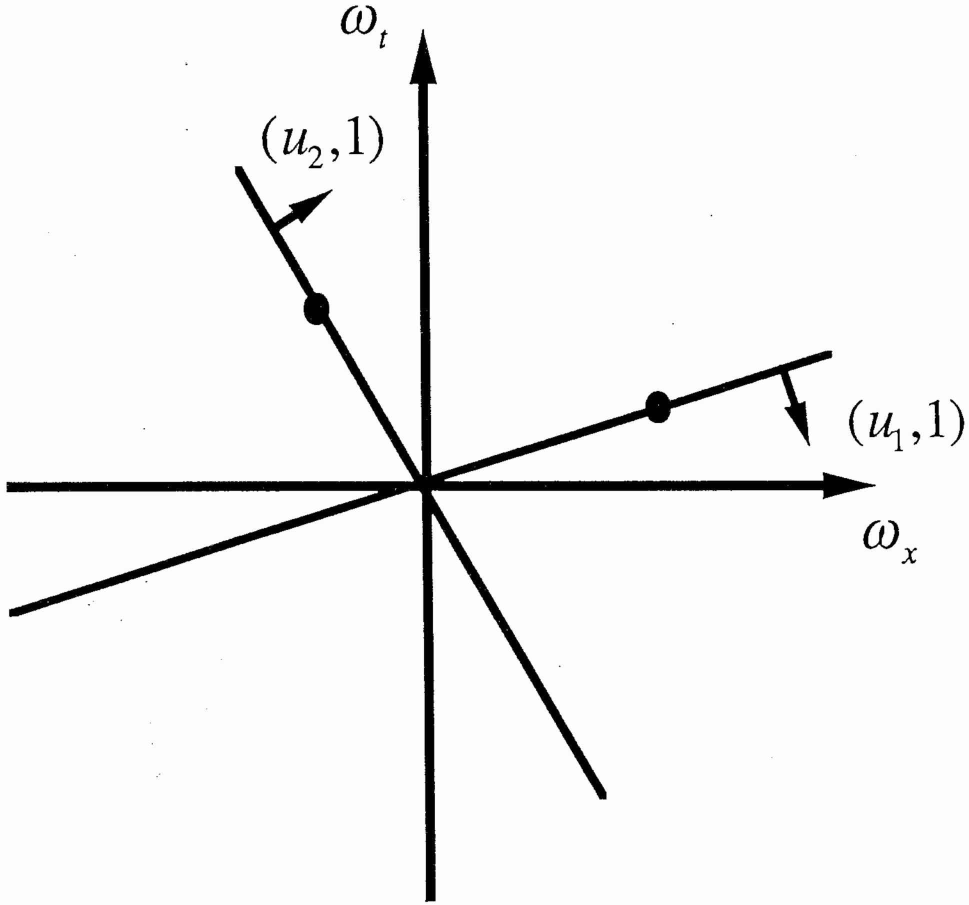
Colinearity constraints:

Constraint of motion #1

$$u_1\omega_x + \omega_t = 0$$

Constraint of motion #2

$$u_2\omega_x + \omega_t = 0$$



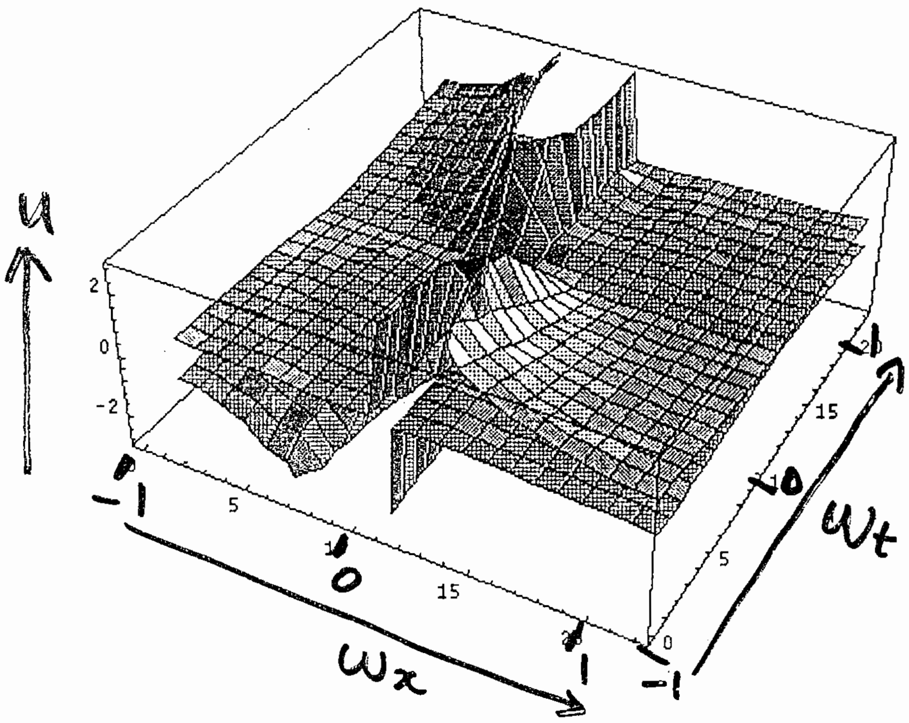
Constraint of motion transparency

$$u_1u_2\omega_x^{\ 2} + (u_1 + u_2)\omega_x\omega_t + \omega_t^{\ 2} = 0$$

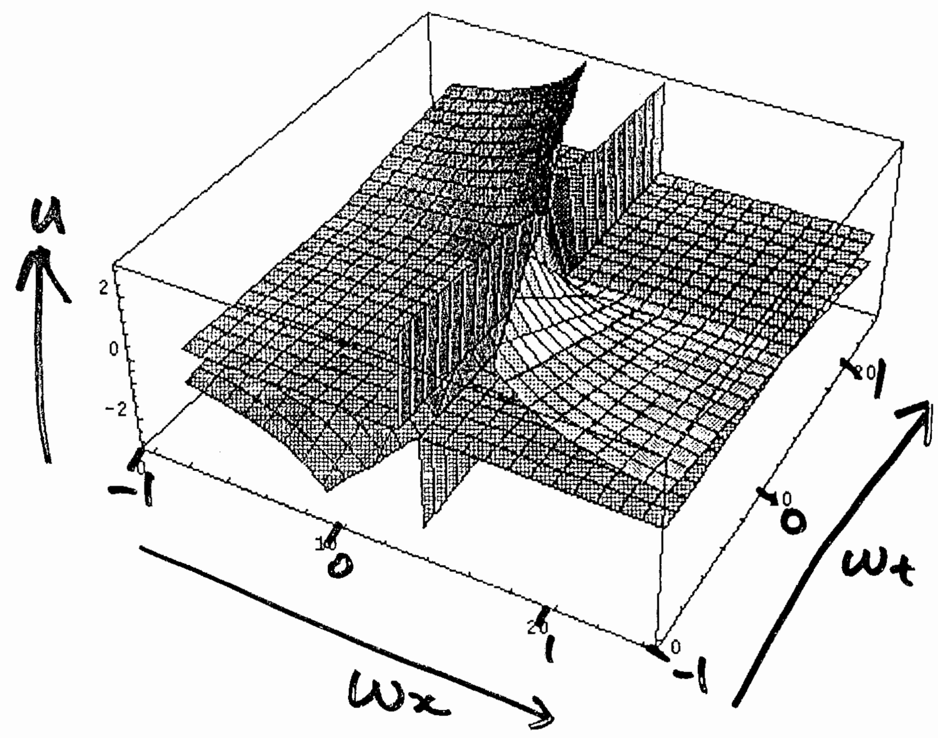
Desired mapping $R^4 \to R$, *two-valued*

$$(\omega_x,\omega_t)_1(\omega_x,\omega_t)_2 \to u_1, u_2$$

Learning samples: $\{(\omega_x,\omega_t)_{(i)}, u_{(i)}\}_{i=1,2,\cdots,N}$



MURN
(Gussian RBF)



Analytical Solution

● まとめ

◎ ワークステーションに多価正則化ネットワークをインプリメントした。

◎ 多重表面の復元に関する数値実験を行い、芳しい結果が得られた。

◎ １次元２重運動の検出モジュールを例からの学習によって実現した。

● 今後の課題

◎ ２次元２重運動の検出モジュールの実現