

# Radial Basis Functions

Toru Wakahara

[1] Radial Basis Functions の意義

[2] RBF と正確な内挿の問題

[3] RBF networks への拡張

- 第2層の学習

- 分類問題と事後確率の関係

- 基底関数の選択法と第1層の学習

[4] 多層パーセプトロンとの比較

# Radial Basis Functions の意義

## [1] 形式としての特徴

**MLP** は入力  $x$  と重みベクトル  $w$  の内積  $w^t x$  を変数とする非線形の活性化関数をもつユニットから構成される

**RBF** は入力  $x$  とプロトタイプ  $\mu$  との距離  $\|x - \mu\|$  を変数とする非線形の活性化関数をもつユニットから構成される

## [2] 概念としての重要性

関数の近似, 正則化理論, 雑音データの内挿, 確率密度関数の推定, 最適識別理論, ポテンシャル関数法, などと緊密にリンクを張る

# RBFと正確な内挿の問題 (1)

$d$ 次元空間の  $x$  から1次元空間  $t$  への写像で  $N$  個の点  $\{x_n\}$  での目標値  $\{t^n\}$  が与えられたとき,

$$h(x_n) = t^n, \quad n = 1, \dots, N$$

を満たす最適な内挿関数  $h(x)$  を求める

これを次の形の RBF で解くことができるだろうか？

$$h(\mathbf{x}) = \sum_{n=1}^N w_n \varphi(\|\mathbf{x} - \mathbf{x}_n\|)$$

# RBFと正確な内挿の問題 (2)

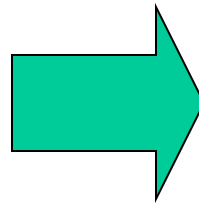
次の方程式を解けばよい

$$\Phi \mathbf{w} = \mathbf{t}$$

$$\mathbf{t} = (t^n), \quad \mathbf{w} = (w_n), \quad \Phi_{nn'} = \varphi(\|\mathbf{x}_n - \mathbf{x}_{n'}\|)$$

ほとんどの  $\varphi(\cdot)$  に対して  $\Phi$  は正則行列であり、  
正確な内挿を与える  $\mathbf{w}$  が次式で決定できる

$$\mathbf{w} = \Phi^{-1} \mathbf{t}$$



$$w_{nk} = \sum_{n'} (\Phi^{-1})_{nn'} t_k^{n'}$$

多次元への拡張

# RBFと正確な内挿の問題 (3)

RBF関数  $\varphi(\cdot)$  としては様々な形が用いられる

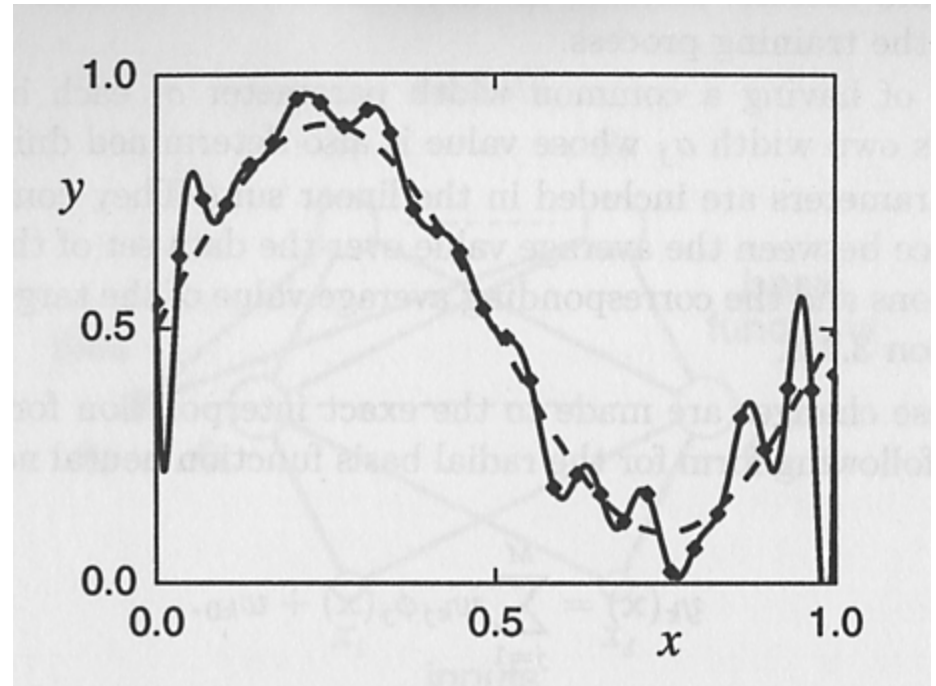
局所的:

$$\varphi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

非局所的:

$$\varphi(x) = x^3$$

$$\varphi(x) = x$$



雑音を含む正弦波形の30点  
 $\sigma = 0.067$  ← データ間隔の  
ほぼ2倍

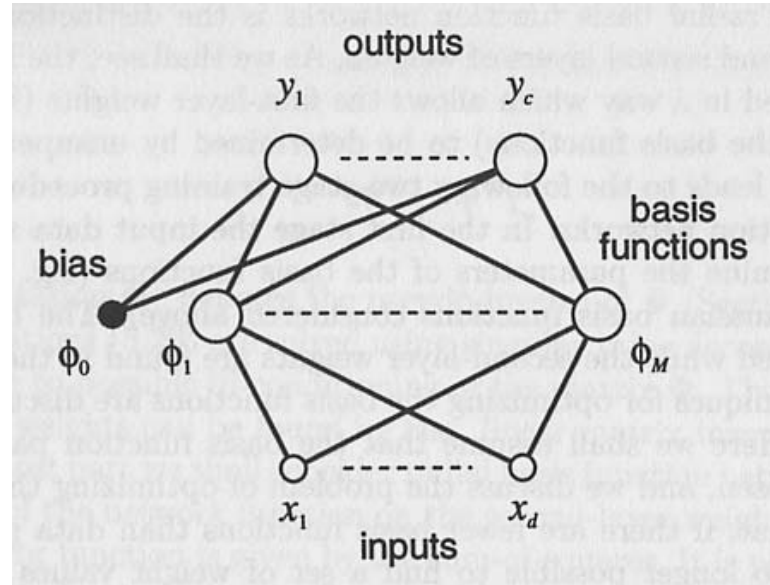
# RBF networks への拡張 (1)

▪ RBF の個数:

$$N \rightarrow M$$

▪ RBF のセンタ:

$$\mathbf{x}_n \rightarrow \boldsymbol{\mu}_j$$

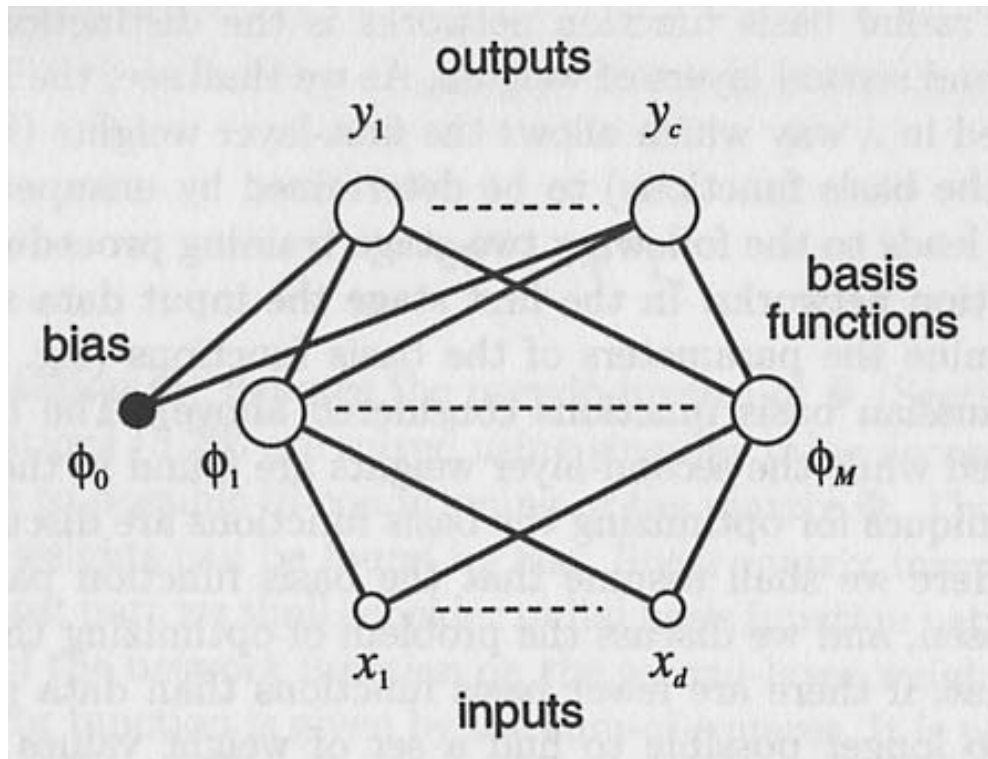


$$y_k(\mathbf{x}) = \sum_{j=1}^M w_{kj} \varphi_j(\mathbf{x}) + w_{k0} = \sum_{j=0}^M w_{kj} \varphi_j(\mathbf{x})$$

$$\varphi_j(x) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right) : \text{Gauss型の RBF}$$

# RBF networks への拡張 (2)

$$y_k(\mathbf{x}) = \sum_{j=0}^M w_{kj} \varphi_j(\mathbf{x})$$



第2層の重みベクトル  $w_{kj}$  は教師つきで最適な写像となるように学習する

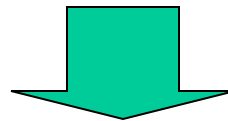
第1層の  $\varphi_j(\mathbf{x})$  への結線はセンタ  $\mu_j$  の成分  $\mu_{ji}$  に対応しており,  $\{\mathbf{x}_n\}$  の分布を最適に近似するように学習する

# RBF networks の第2層の学習 (1)

$$y_k(\mathbf{x}) = \sum_{j=0}^M w_{kj} \varphi_j(\mathbf{x}) \quad \rightarrow \quad \mathbf{y}(\mathbf{x}) = \mathbf{W}\boldsymbol{\varphi}$$

$$\mathbf{W} = (w_{kj}), \quad \boldsymbol{\varphi} = (\varphi_j)$$

$$E = \frac{1}{2} \sum_n \sum_k \{ y_k(\mathbf{x}_n) - t_k^n \}^2 \quad \rightarrow \quad \text{min for } \mathbf{W}$$



$$\frac{\partial E}{\partial w_{kj}} = \sum_{n=1}^N \left\{ \sum_{j'=0}^M w_{kj'} \varphi_{j'}(\mathbf{x}_n) - t_k^n \right\} \varphi_j(\mathbf{x}_n) = 0$$

# RBF networks の第2層の学習 (2)

$$\frac{\partial E}{\partial w_{kj}} = \sum_{n=1}^N \left\{ \sum_{j'=0}^M w_{kj'} \varphi_{j'}(\mathbf{x}_n) - t_k^n \right\} \varphi_j(\mathbf{x}_n) = 0$$

行列形式に書き換えると

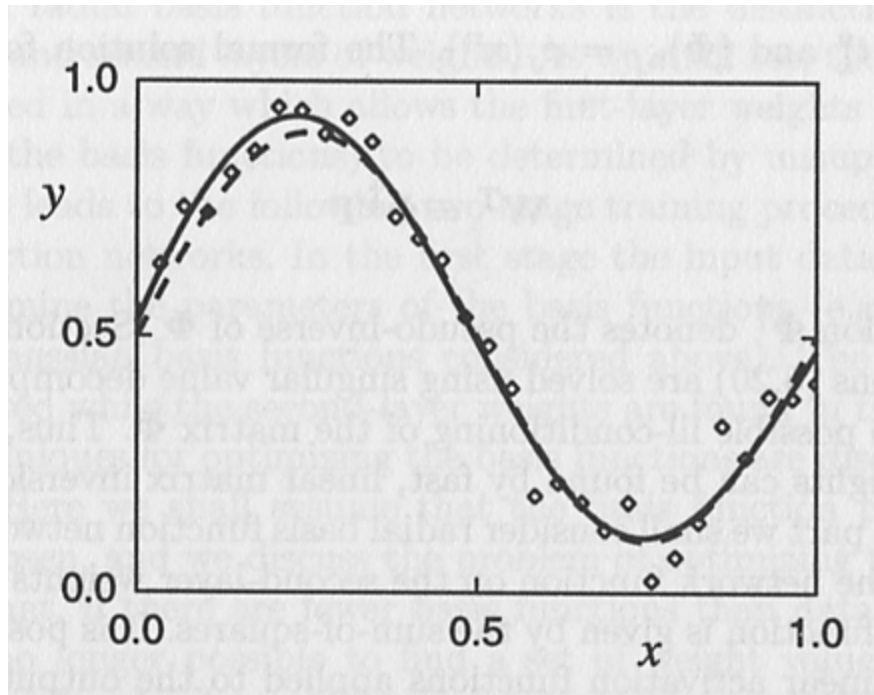
$$\Phi^t \Phi \mathbf{W}^t = \Phi^t \mathbf{T} \quad \rightarrow \quad \mathbf{W}^t = \Phi^+ \mathbf{T}$$

$$\mathbf{T}_{nk} = t_k^n, \quad \Phi_{nj} = \varphi_j(\mathbf{x}_n) \quad \Phi^+ = (\Phi^t \Phi)^{-1} \Phi^t$$

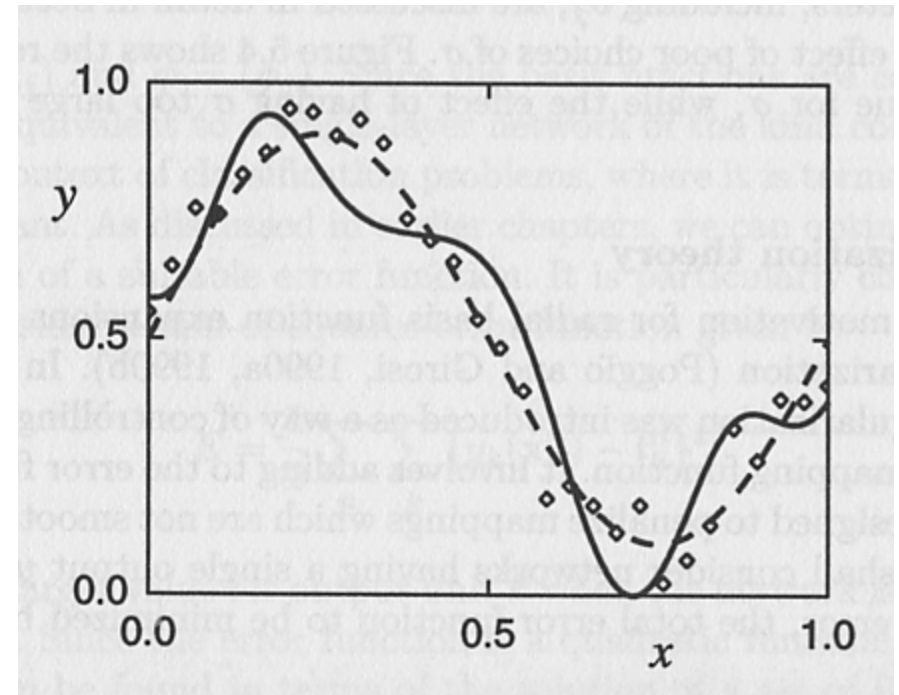
:一般逆行列

一般逆行列の計算には特異値分解(SVD)を利用

# RBF networks の第2層の学習 (3)



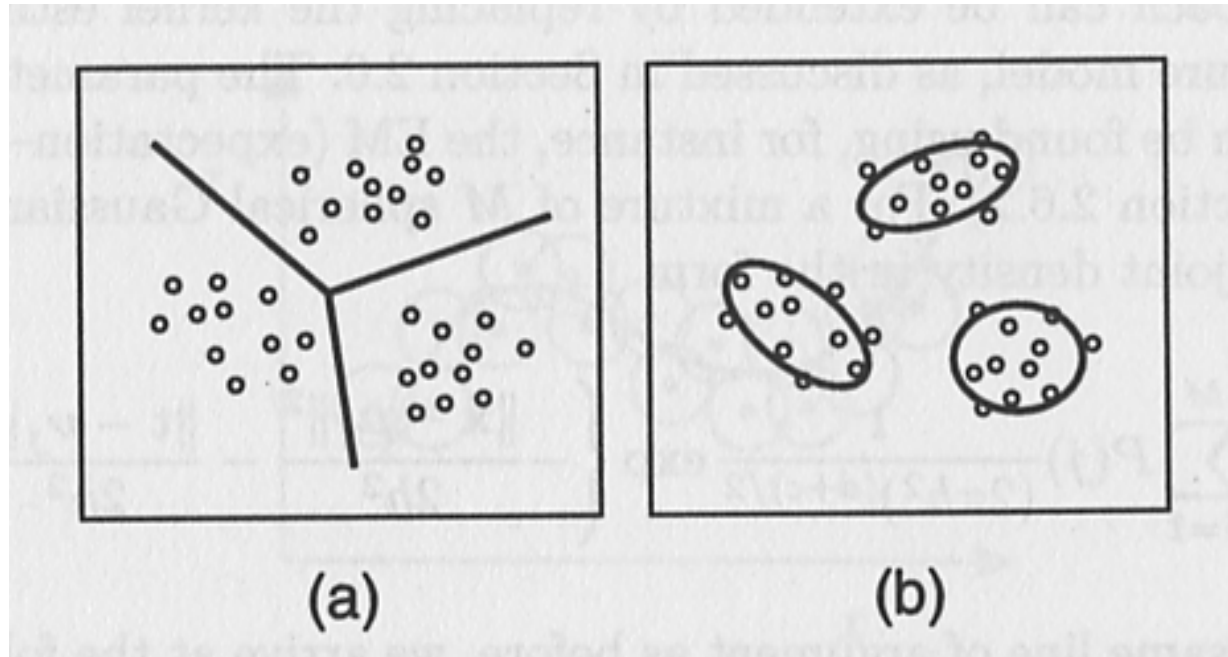
RBF ユニット数: 5個  
センタ:  $\{x_n\}$  から 5個を  
ランダムに選択  
 $\sigma = 0.4$  ← センタの平均  
間隔のほぼ2倍



$\sigma$  のみ変更:  $\sigma = 0.08$   
RBF による  $\{x_n\}$  分布の  
表現能力に難あり

↑  
第1層の学習の問題

# RBF networks と分類問題 (1)



MLP はクラスを  
超平面で分割する

RBF は各クラスの分布を  
核関数で表現する

## RBF networks と分類問題 (2)

各クラスの確率密度関数をクラス共通の  $M$  個の基底関数  $\{p(\mathbf{x} | j)\}_{j=1}^M$  で表現すると考える

$$p(\mathbf{x} | C_k) = \sum_{j=1}^M p(\mathbf{x} | j) P(j | C_k)$$

$$p(\mathbf{x}) = \sum_k p(\mathbf{x} | C_k) P(C_k) = \sum_{j=1}^M p(\mathbf{x} | j) P(j)$$

Bayes' theorem により事後確率は次式で得られる

$$P(C_k | \mathbf{x}) = \frac{\sum_{j=1}^M p(\mathbf{x} | j) P(j | C_k) P(C_k)}{\sum_{j'=1}^M p(\mathbf{x} | j') P(j')} \frac{P(j)}{P(j)}$$

# RBF networks と分類問題 (3)

RBF networks が事後確率を表現していると考える

$$P(C_k | \mathbf{x}) = \sum_{j=1}^M w_{kj} \varphi_j(\mathbf{x})$$

と等値すると次の関係  
が導かれる

$$\varphi_j(\mathbf{x}) = \frac{p(\mathbf{x} | j) P(j)}{\sum_{j'=1}^M p(\mathbf{x} | j') P(j')} = P(j | \mathbf{x})$$

$$w_{kj} = \frac{P(j | C_k) P(C_k)}{P(j)} = P(C_k | j)$$

どのような  
解釈が  
できるか?

# RBF networks の第1層の学習 基底関数の選択法 (1)

## 1. データ点のサブセットをセンタ $\mu_j$ とする方法

- ・ 無作為に学習データから  $M$  点をセンタとして選択する  
この場合, 最適性はまったく保証されない
- ・ 全点を初期選択し, 分類能力あるいは確率密度関数  $p(x)$  の表現能力の低下を最小限とする, などの評価基準を用いて, センタ数を逐次削減する

$\sigma_j$  は別の手順で決定する

- ・  $\sigma_j$  を共通に, センタ間の平均距離の整数倍とする
- ・ 各センタごとに  $L$  近傍センタへの平均距離を  $\sigma_j$  とする

# RBF networks の第1層の学習 基底関数の選択法 (2)

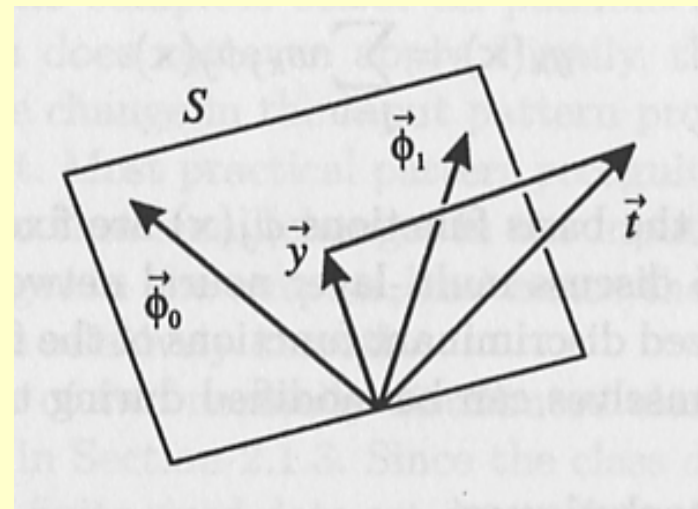
## 2. 正規直交最小2乗法

- ・ センタをデータ点から逐次選択するが, 出力空間  $S$  での教師データベクトル  $t$  との2乗エラー  $E$  を最大限に削減するようにデータ点を順に選んでいく

$$\boldsymbol{\varphi}_j = (\varphi_j(\mathbf{x}_n)), \quad \mathbf{t} = (t^n)$$

$$E = \frac{1}{2} \left\| \sum_{j=0}^M w_j \boldsymbol{\varphi}_j - \mathbf{t} \right\|^2$$

$\sigma_j$  は別の手順で決定する



# RBF networks の第1層の学習 基底関数の選択法 (3)

## 3. クラスタリング法

- ・ クラスタリングを行い, クラスタ中心をセンタとする  
例えば, *K-means* や *VQ algorithms* が利用される

$$J = \sum_{j=1}^M \sum_{n \in S_j} \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \rightarrow \min \text{ for } \{S_j\}_{j=1}^M$$

$$\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_{n \in S_j} \mathbf{x}_n$$

クラスタリング結果から  $\sigma_j$  あるいは  $\Sigma_j$  が計算できる

# RBF networks の第1層の学習 基底関数の選択法 (4)

## 4. 混合モデル法

- ・  $x$  の確率密度関数  $p(x)$  を Gaussian mixture models で推定して, 得られた Gaussian 型基底関数を RBF とする

$$p(\mathbf{x}) = \sum_{j=1}^M P(j) \varphi_j(\mathbf{x})$$

$$L = \prod_n p(\mathbf{x}_n) \rightarrow \max \text{ for } \left\{ \boldsymbol{\mu}_j, \sigma_j^2, P(j) \right\}_{j=1}^M$$

→  $P(j)$  情報は捨てて,  $\varphi_j(\mathbf{x})$  を RBF とする

# RBF と MLP の比較

比較内容	MLP	RBF
hidden units の出力	$d-1$ 次元の超平面の上で等しい	$d-1$ 次元の超楕円体の上で等しい
hidden units の勢力範囲	入力空間を多数ユニットで分散表現する	各 $x$ に対し少数ユニットのみが活性化する
network 構成の多様性	層数, 活性化関数, 結合法を選択できる	2層の単純構成を取る
学習法	全体での1段の教師つき学習	基底関数と第2層重みベクトルの2段階学習

# 課題4

正規雑音  $N(0.0, 0.05^2)$  を含む正弦波形:

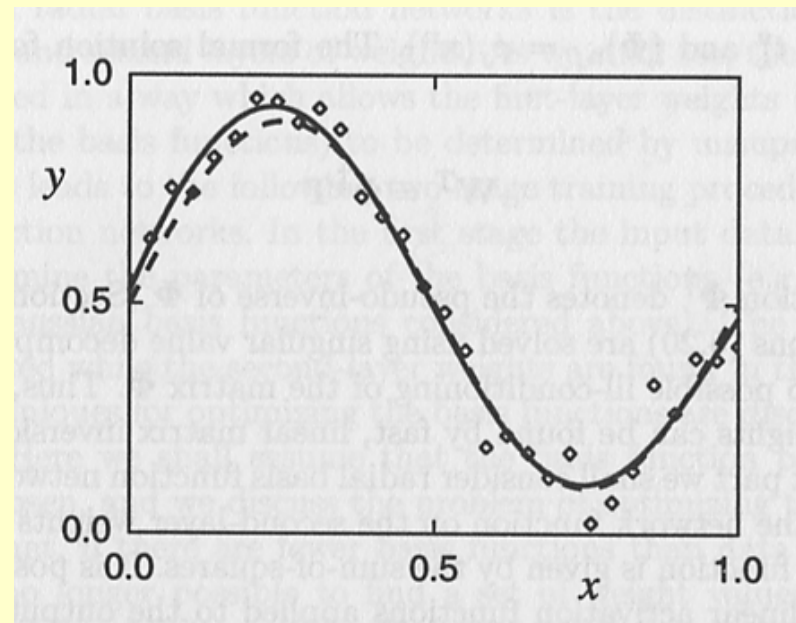
$$y = 0.5 + 0.4 \sin(2\pi x)$$

を  $[0, 1]$  で等間隔 (0, 1を含む) にサンプリングした30点に対し, RBFネットワークによる学習を計算機シミュレーションで確認しなさい.

RBF ユニット数 :  $1 \leq M \leq 30$

センタ  $\mu_j$  :  $[0, 1]$  を等間隔に  
( $M+1$ ) 分割して設定  
(*ex.*  $M = 1$  では 0.5)

共通の  $\sigma$  : センタ間の間隔  
の2倍



## 課題4(つづき)

$y(x_n) = \sum_{j=0}^M w_j \varphi_j(x_n)$  の重み係数  $w_j$  を算出する.

ただし,  $\varphi_0(x_n) = 1, \quad n = 1, \dots, N$

$\mathbf{t} = (t^n), \quad \Phi_{nj} = \varphi_j(x_n)$  より行列形式で表現すると

$$\Phi \mathbf{w} = \mathbf{t} \quad \rightarrow \quad \Phi^t \Phi \mathbf{w} = \Phi^t \mathbf{t}$$

$\therefore \mathbf{w} = \Phi^+ \mathbf{t}$  where  $\Phi^+ = (\Phi^t \Phi)^{-1} \Phi^t$  : 一般逆行列

$\Phi^+$  の計算には特異値分解(SVD)を活用する.

このため, ヘッダファイル `nrutil.h`, `svd.h` をインクルードし,  
関数 `dsvdcmp`, `dsvbksb` を用いなさい.

また, `solve_linear_eqs.c` が参考になります.

## 課題4(つづき)

誤差は  $RMSE$  (Root Mean Square Error) を用いて評価してみましょう. 定義は次式になります.

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N \left( y(x_n) - t^n \right)^2}$$

センタ数  $M$  がデータ点数  $N (= 30)$  に等しくなると, 上記の誤差  $RMSE$  は 0 になります.

余裕があれば, 解曲線の正弦波形, 雑音の乗った観測点, RBFネットワークで得られた曲線を, 20頁にあるような図にプロットしてみなさい.