Nonlinear and chaotic differential function approximation by learning Mahalanobis-NRBF neural modules

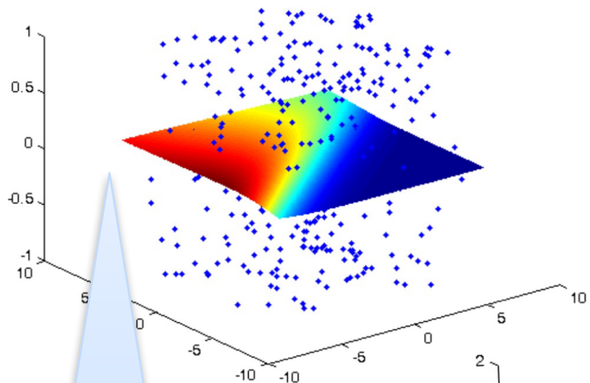
Jiann-Ming Wu
Department of Applied Mathematics
National Dong Hwa University

Outline

- Approximation
  - Nonlinear function approximation
  - Chaotic differential function approximation
- Neural Organization
  - A Mahalanobis-NRBF module
  - A network of K Mahalanobis-NRBF modules
- Learning
  - Annealed competitive learning
  - Annealed KLD minimization
  - Annealed cooperative–competitive learning of multiple Mahalanobis-NRBF
- Numerical simulations
  - Nonlinear function approximation
  - Chaotic differential function approximation
- Conclusions

Nonlinear function approximation

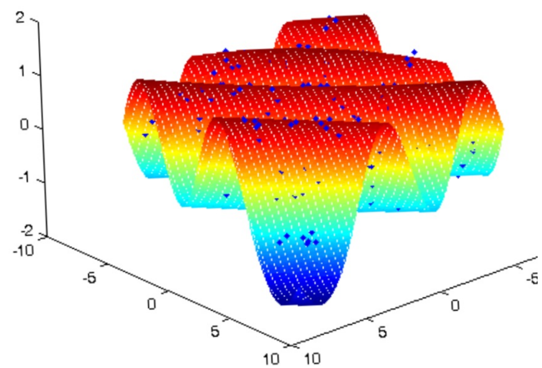
- Given samples from a high-dimensional nonlinear single-valued mapping, the goal is to optimize adaptable parameters for faithful approximation



$$y = F(x \mid \theta_{opt})$$

An adaptable network mapping

Optimal network parameters



CDFA(Chaitic differential FA) : Mackey-Glass 17

$$\frac{\partial x}{\partial t} = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t),$$

$$\tau = 17,\ a = 0.2,\ c = 10 \text{ and } b = 0.1$$

M. Mackey, L. Glass, Oscillation and chaos in physiological control systems, Science 197 (1977) 287.

Mackey-Glass 30

$$\frac{\partial x}{\partial t} = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t),$$

$\tau = 30$ $a = 0.2$, $c = 10$ and $b = 0.1$

CDFA: Nonlinear delay differential equations

$$\frac{\partial x}{\partial t} = x(t-\tau) - x^3(1-\tau),$$

where the delav τ is set to 1.6.

J.C. Sprott, A simple chaotic delay differential equation, Phys. Lett. A 366 (2007) 397–402.

Chaotic differential function approximation using Multilayer N rural Networks

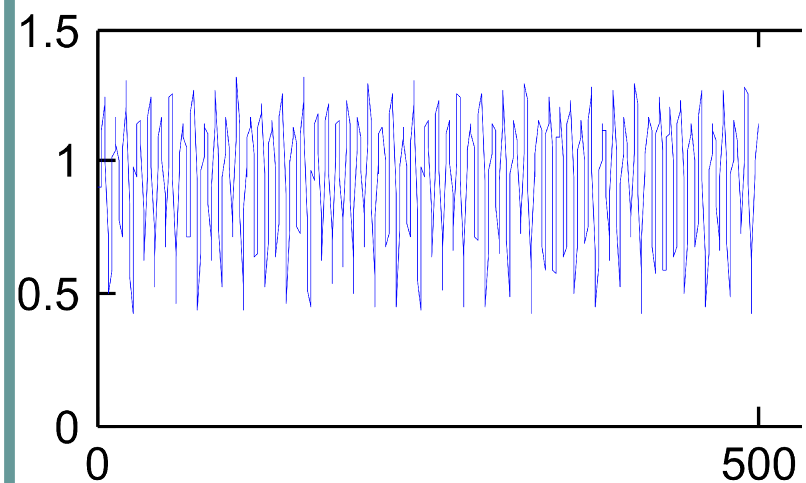
- Cechin, Pechmann, Oliveira, Chaos Solitons Fractal (2008)
- Mirzaee, Chaos Solitons Fractal (2009)
- Moody, Darken, Neural Computation (1989)
- Lin, Horne, Tino, IEEE Trans. Neural Netw. (1996)

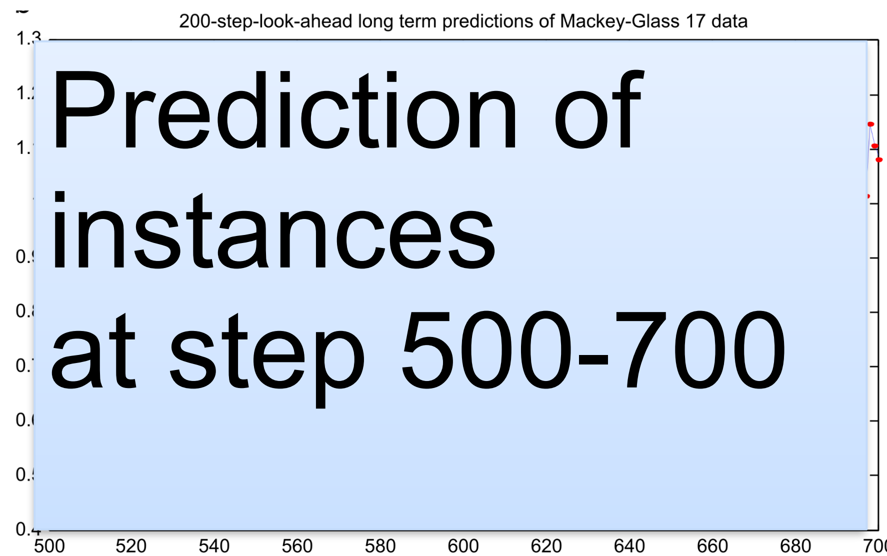
CDFA: goal and methodologies

- Goal
  - Long term look-ahead prediction
- Methodology
  - Data driven approaches
  - Recurrence relation modeling
  - Supervised learning of multilayer neural networks

Data driven long-term prediction

- MG(Mackey–Glass) 17 generated by RK(Runge-Kutta) 4

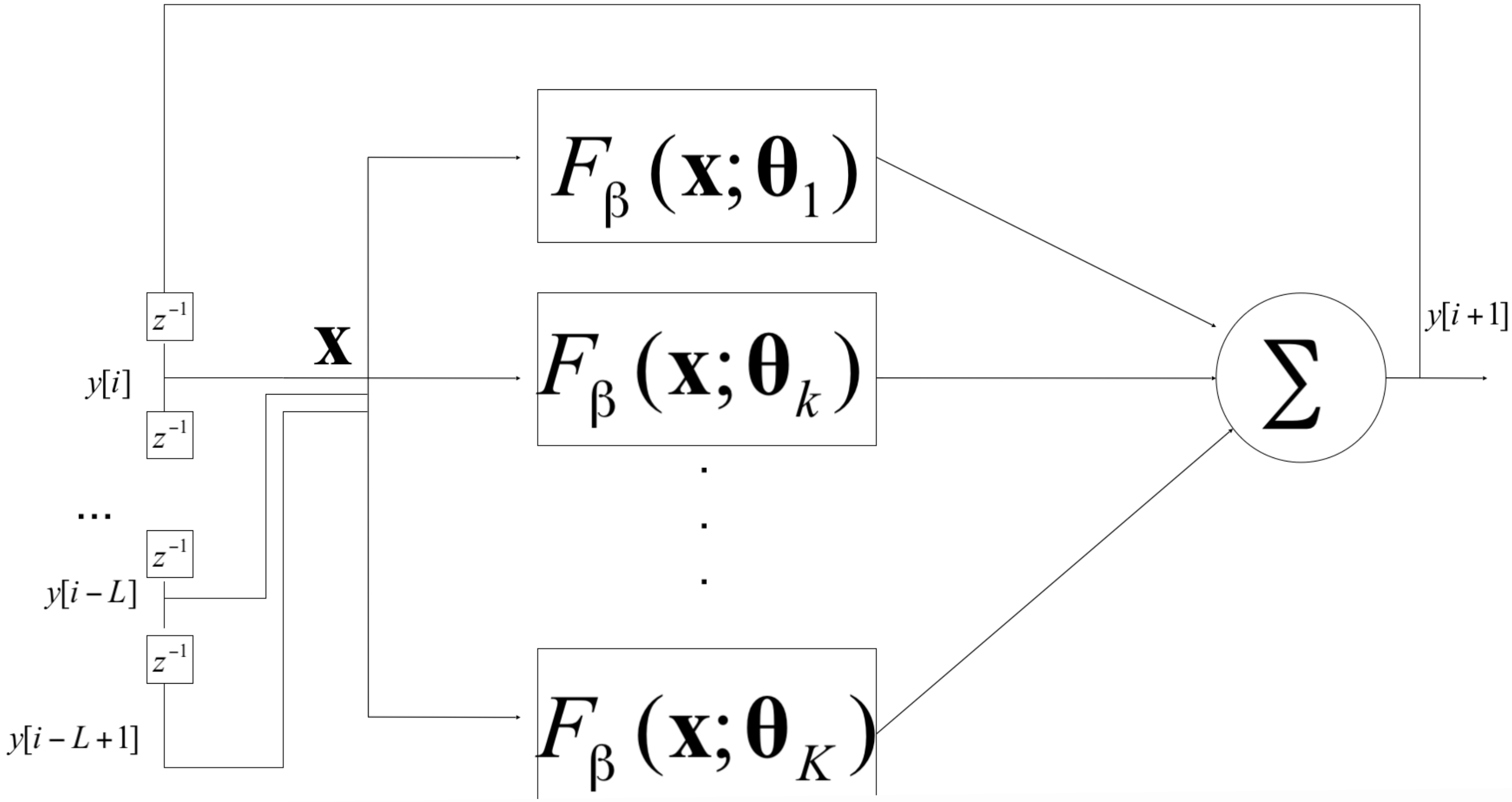




Prediction of instances at step 500-700

200-step-look-ahead prediction.

Nonlinear Recurrent Relation Modeling based on nonlinear function approximation



$$o_t = f(\mathbf{x}_t = (o_{t-L}, o_{t-L+1}, \ldots, o_{t-1})^T),$$

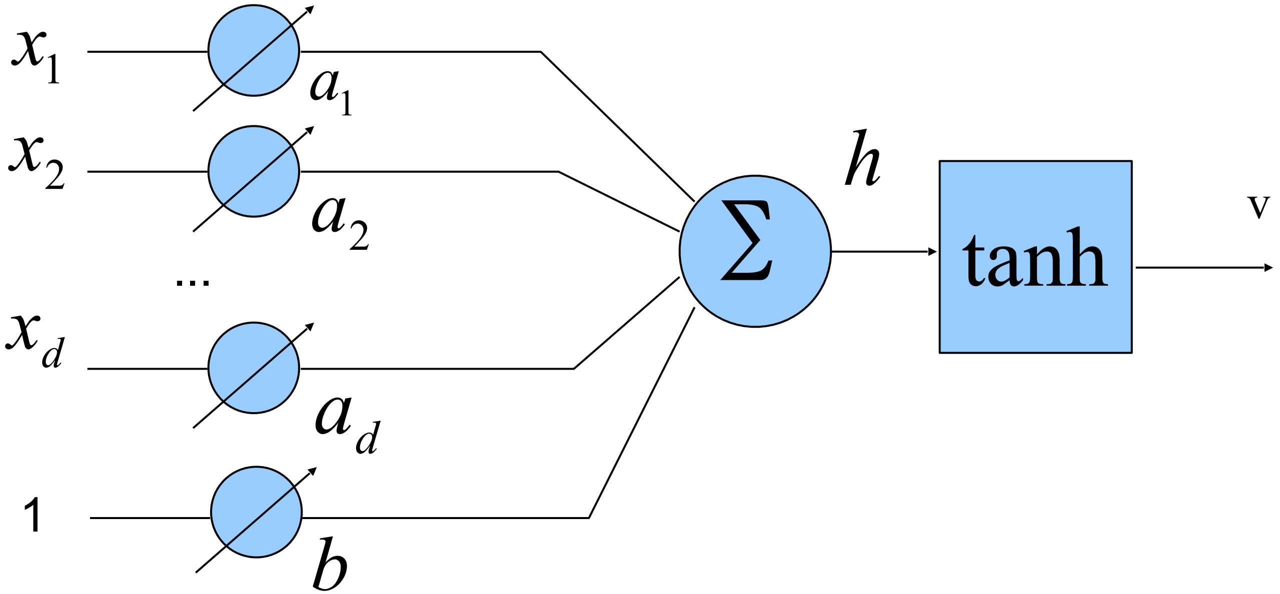
MNN:Projective basis functions

- Multilayer perceptrons, Rosenblatt (1962),
- Adalines, Widrow (1962)
- Multilayer Potts perceptrons, Wu (2008), IEEE NN
- Generalized Adalines, Wu, Hsu & Lin, (2006), IEEE NN

Perceptrons

- Rosenblatt (1962), Widrow (1962)
- Post-tanh (sigmoid-like) projection

$$v = \tanh(h = a_1 x_1 + a_2 x_2 + \ldots + a_d x_d + b)$$



MNN: Radial basis functions

RBF(Radial Basis Function) Network

$$y(t \mid \theta) = G(\mathbf{x}[t] \mid \theta)$$

$$= w_0 + \sum_{m=1}^{M} w_m \exp\left(-\frac{\left\| \mathbf{x}[t] - \boldsymbol{\mu}_m \right\|^2}{2\sigma_m^2}\right)$$

Network parameter

$$\theta = \{\mathrm{w}_\mathrm{i}\}_i \cup \{\boldsymbol{\mu}_i\}_i \cup \{\sigma_i\}_i$$

Previous works for RBF networks

[4] F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Spartan Books, Washington, DC, 1962.

[5] P.J. Werbos, Backpropagation: past and the future, in: Neural Networks, IEEE International Conference, vol. 1, 1988, pp. 343–353.

[6] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, MA, 1991.

[7] J. Moody, J. Darken, Fast learning in networks of locally-tuned processing units, Neural Comput. 1 (1989) 281–294.

[8] T. Poggio, F. Girosi, A theory of networks for approximation and learning, Proc. IEEE 78 (1990) 1481–1497.

[9] G. Rätsch, T. Onoda, K.R. Muller, Soft margins for AdaBoost, Mach. Learn. 42 (3) (2001) 287–320.

[10] R. Battiti, 1st-order and 2nd-order methods for learning – between steepest descent and Newton method, Neural Comput. 4 (1992) 141.

[11] C. Charalambous, Conjugate-gradient algorithm for efficient training of artificial neural networks, IEE Proc. G Circuits Devices Syst. 139 (1992) 301.

[12] M.T. Hagan, M.B. Menhaj, Training feedforward networks with the Marquardt algorithm, IEEE Trans. Neural Netw. 5 (6) (1994) 989–993.

[13] L. Ljung, System Identification – Theory for the User, Prentice-Hall, Englewood Cliffs, NJ, 1987.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY
and
CENTER FOR BIOLOGICAL INFORMATION PROCESSING
WHITAKER COLLEGE

A.I. Memo No.1140
C.B.I.P. Paper No. 31

July 1989

A Theory of Networks for Approximation and Learning

Tomaso Poggio and Federico Girosi

Mahalanobis-RBF

Neural Comput. 2002 Mar;14(3):689-713.

Natural discriminant analysis using interactive Potts models.

Wu JM.

Abstract

Natural discriminant analysis based on interactive Potts models is developed in this work. A generative model composed of piece-wise multivariate gaussian distributions is used to characterize the input space, exploring the embedded clustering and mixing structures and developing proper internal representations of input parameters. The maximization of a log-likelihood function measuring the fitness of all input parameters to the generative model, and the minimization of a design cost summing up square errors between posterior outputs and desired outputs constitutes a mathematical framework for discriminant analysis. We apply a hybrid of the mean-field annealing and the gradient-descent methods to the optimization of this framework and obtain multiple sets of interactive dynamics, which realize coupled Potts models for discriminant analysis. The new learning process is a whole process of component analysis, clustering analysis, and labeling analysis. Its major improvement compared to the radial basis function and the support vector machine is described by using some artificial examples and a real-world application to breast cancer diagnosis.

PMID: 11860688 [PubMed]

Mahalanobis-RBF

$$y(t \mid \theta) = G(\mathbf{x}[t] \mid \theta)$$

$$= w_0 + \sum_{m=1}^{M} w_m \exp\left(-\frac{\left\| \mathbf{x}[t] - \boldsymbol{\mu}_m \right\|_A^2}{2\sigma_m^2}\right)$$

$$h_m \equiv \| \mathbf{x} - \mathbf{w}_m \|_{\mathbf{A}}^2$$

$$= (\mathbf{x} - \mathbf{w}_m)^T \mathbf{A} (\mathbf{x} - \mathbf{w}_m).$$

Mahalanobis-NRBF modules

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Annealed cooperative–competitive learning of Mahalanobis-NRBF neural modules for nonlinear and chaotic differential function approximation

Jiann-Ming Wu *, Chun-Chang Wu, Ching-Wen Huang

Department of Applied Mathematics, National Dong Hwa University, Shoufeng, Hualien 974, Taiwan

ARTICLE INFO

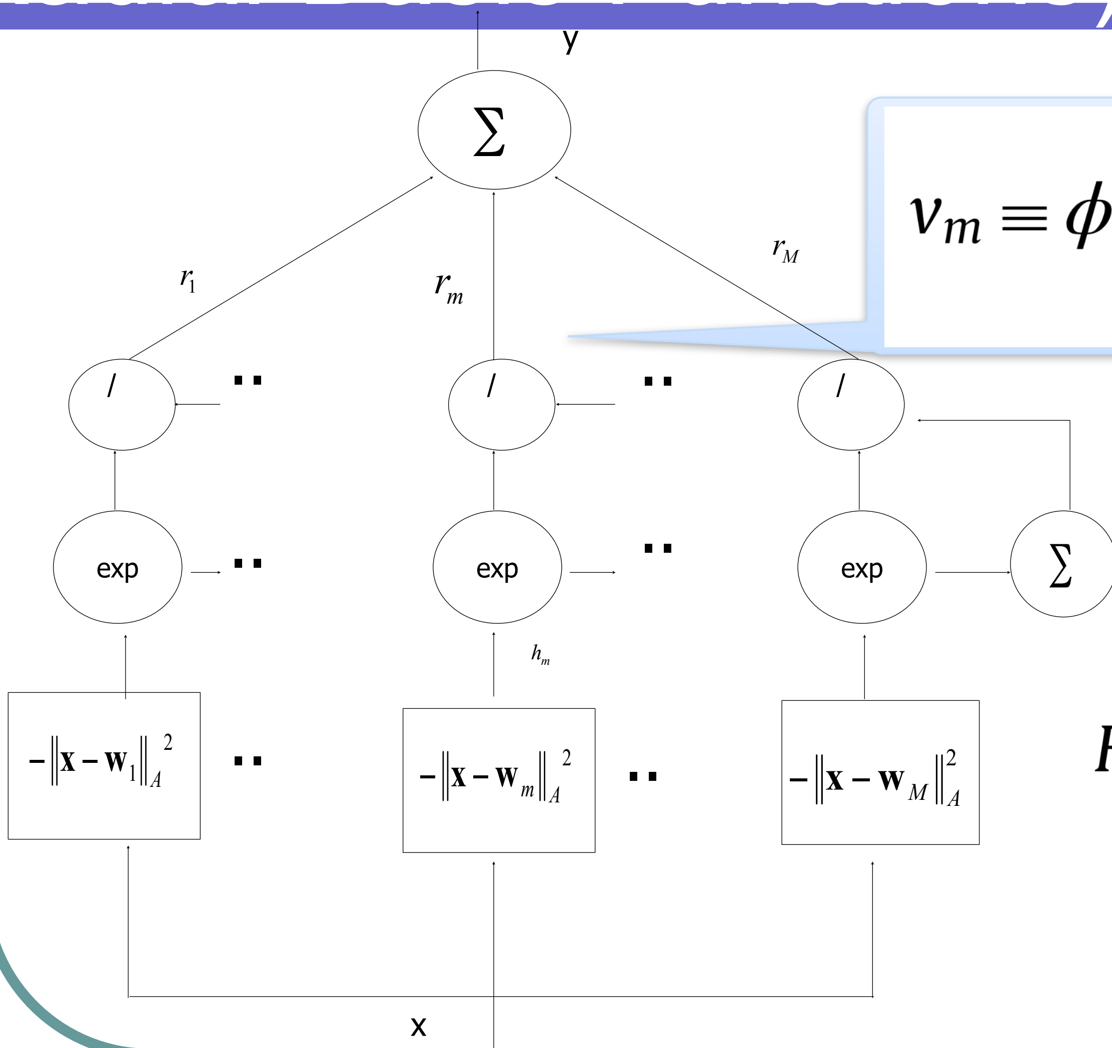
Article history:
Received 30 May 2013
Received in revised form
13 January 2014
Accepted 17 January 2014
Communicated by W.S. Hong

Keywords:
Multilayer neural networks
Free energy function
Mixed integer programming
Mean field annealing
Long term prediction
Chaotic time series

ABSTRACT

This work explores annealed cooperative–competitive learning of multiple modules of Mahalanobi normalized radial basis functions (NRBF) with applications to nonlinear function approximation an chaotic differential function approximation. A multilayer neural network is extended to be compose of multiple Mahalanobis-NRBF modules. Each module activates normalized outputs of radial basi functions, determining Mahalanobis radial distances based on its own adaptable weight matrix. A essential cooperative scheme well decomposes learning a multi-module network to sub-tasks of learnin individual modules. Adaptable network interconnections are asynchronously updated module-by module based on annealed cooperative–competitive learning for function approximation under physical-like mean-field annealing process. Numerical simulations show outstanding performance c annealed cooperative–competitive learning of a multi-module Mahalanobis-NRBF network for nonlinea function approximation and long term look-ahead prediction of chaotic time series.

Mahanobis-NRBF(Normalized Radial Basis Functions)



$$v_m \equiv \phi_m(\mathbf{x}) = \frac{\exp(-\beta h_m)}{\sum_j \exp(-\beta h_j)}.$$

$$\sum_m v_m = 1$$

$$F_\beta(\mathbf{x};\boldsymbol{\theta}) \equiv \mathbf{v}^T\mathbf{r}$$

$$= \sum_m r_m \phi_m(\mathbf{x}),$$

A network of multiple Mahalanobis-NRBF modules

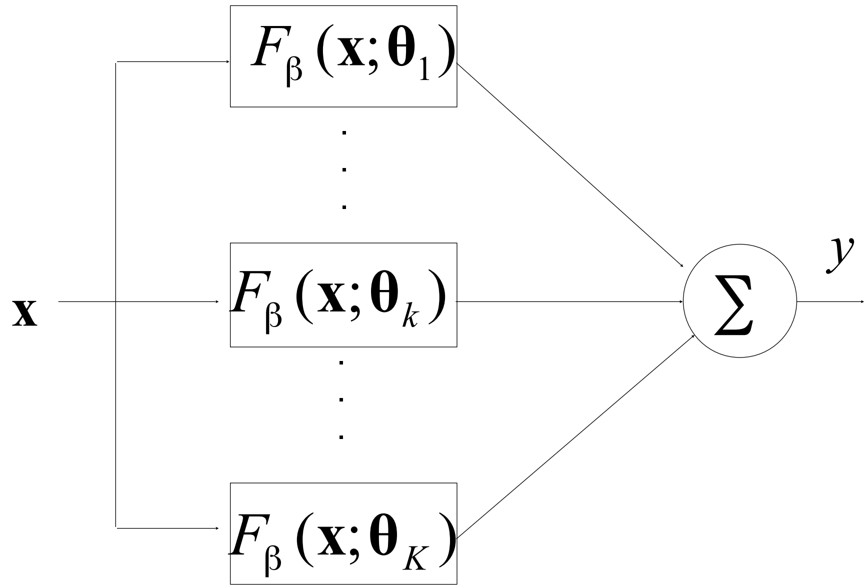
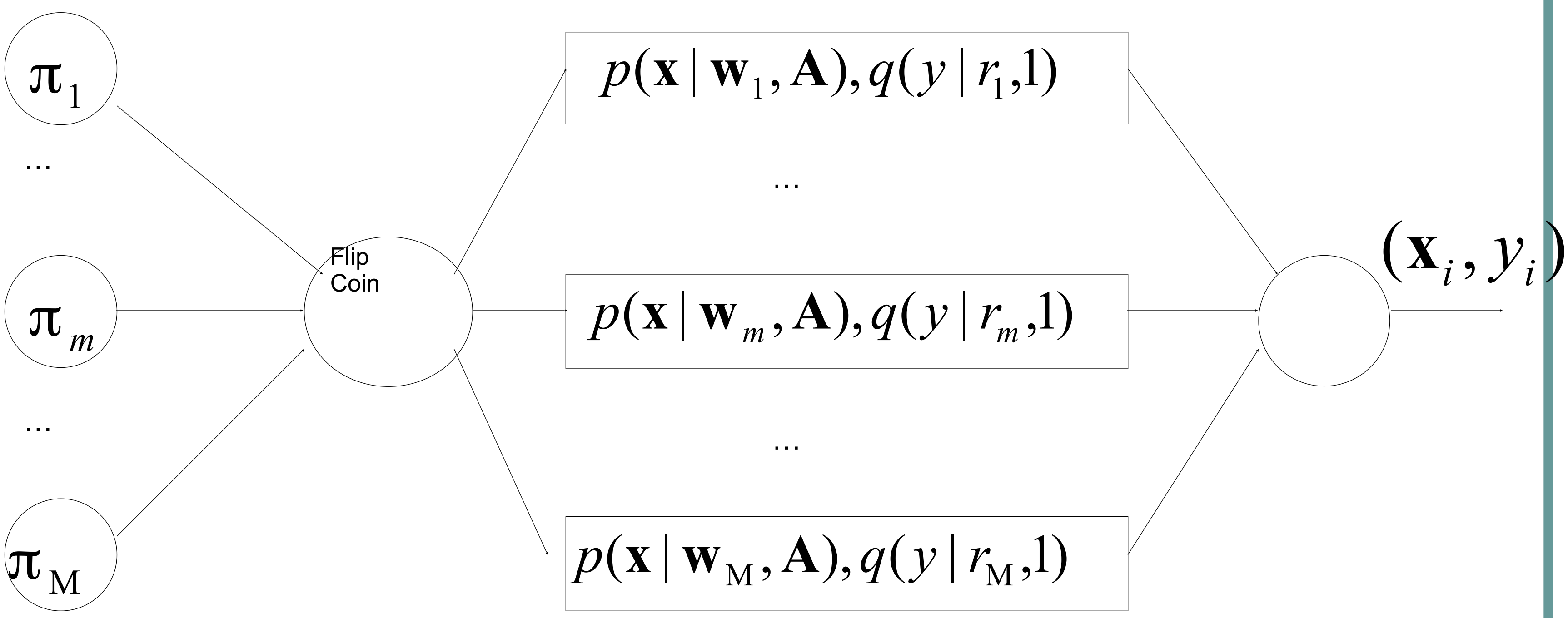


Figure 3

Mahalanobis-NRBF modules

- A generative model for paired predictors and targets
- Model-based Mahalanobis-NRBF module
- Annealed competitive learning
- A network of multiple Mahalanobis-NRBF modules
- Annealed competitive-cooperative learning

A generative model



Paired Training Data

$$S = \{(x_i, y_i)\}_i$$

A sub-model

share a common weight matrix

$$p_m(\mathbf{x}) \equiv p(\mathbf{x}|\mathbf{w}_m, \mathbf{A})$$

$$= \frac{1}{(2\pi)^{d/2}\sqrt{|\mathbf{A}^{-1}|}} \exp\left(-\frac{(\mathbf{x}-\mathbf{w}_m)^T \mathbf{A}(\mathbf{x}-\mathbf{w}_m)}{2}\right), \quad (1)$$

and

$$q_m(y) \equiv q(y|r_m, 1)$$

$$= \frac{1}{\sqrt{2\pi}} \exp\left(\frac{(y-r_m)^2}{2}\right), \quad (2)$$

Paired normal random variables

An exclusive membership to joined Gaussians

$$m^* = \arg\min_m h_m$$

Minimal distance to the local mean

$$= \arg\max_m p_m(\mathbf{x})$$

Maximal probability

$$h_m \equiv \|\mathbf{x} - \mathbf{w}_m\|^2_{\mathbf{A}}$$

$$= (\mathbf{x} - \mathbf{w}_m)^T \mathbf{A} (\mathbf{x} - \mathbf{w}_m).$$

Exclusive membership

A unitary vector of binary elements with the mth bit one and others zero

$$\boldsymbol{\delta} = (\delta_1, \ldots, \delta_M)^T \text{ in } \Xi_M = \{\mathbf{e}_m\}_m$$

membership $\boldsymbol{\delta}$ of $\mathbf{x}$ is encoded by $\mathbf{e}_{m^*}$

$$m^* = \arg\min_m h_m$$

$$= \arg\max_m p_m(\mathbf{x})$$

Conditional expectation of r to x

Target

Predictor

- < r | x >

$$F(\mathbf{x};\boldsymbol{\theta}) = \boldsymbol{\delta}^T \mathbf{r}$$

$$F(\mathbf{x};\boldsymbol{\theta}) = \sum_m \delta_m r_m$$

Overlapping membership of x

- $$\hat{\mathbf{v}} = (v_1, \ldots, v_M)^T \in [0,1]^M$$

$$v_m \propto \exp(-\beta h_m),$$

$$\sum_m v_m = 1,$$

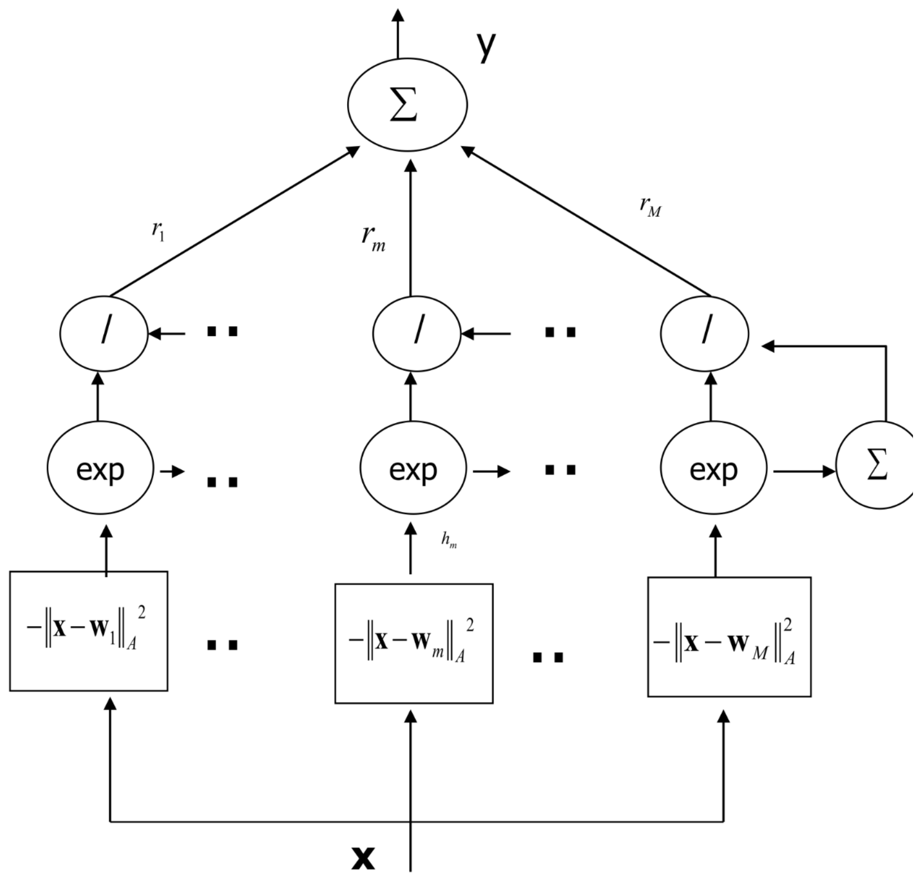
it follows

$$v_m \equiv \phi_m(\mathbf{x}) = \frac{\exp(-\beta h_m)}{\sum_j \exp(-\beta h_j)}.$$

Model-oriented Mahanobis-NRBF module

$$F_{\beta}(\mathbf{x};\boldsymbol{\theta}) \equiv \mathbf{v}^T\mathbf{r}$$

$$= \sum_m r_m \phi_m(\mathbf{x}),$$



Learning a Mahalanobis-NRBF module

- Supervised Learning subject to paired training data
- Model fitting to paired training data
  - Exclusive memberships and Potts encoding
  - Fitting individual sub-models

Annealed competitive learning

- Mathematical framework
  - A mixed integer programming
- Annealed KLD minimization
  - A mixed energy function is not differentiate with respect to discrete variables
  - Boltzmann assumption under thermal equilibrium
  - Annealed Kullback-Leibler Divergence minimization
  - A tractable free energy function

A mixed energy function

Discrete variables

Continuous variables

model fitting criteria

Mean square approximating error

$$E(\Lambda, \boldsymbol{\theta}) = E_1 + \lambda E_2$$
$$= \frac{1}{2}\sum_i \sum_m \delta_{im}(\mathbf{x}_i - \mathbf{w}_m)^T \mathbf{A}(\mathbf{x}_i - \mathbf{w}_m) - \frac{N}{2}\log|\mathbf{A}| + \frac{\lambda}{2}\sum_i \sum_m \delta_{im}(r_m - y_i)^2,$$

where λ is non-negative and Λ denotes collection of exclusive memberships. Minimizing E subject to constraints,

$$\sum_m \delta_{im} = 1, \quad \forall i,$$
$$\delta_{im} \in \{0, 1\}, \quad \forall i, m,$$

design cost

$$D_S(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{N} \|y_i - F(\mathbf{x}_i;\boldsymbol{\theta})\|^2,$$

$$F(\mathbf{x};\boldsymbol{\theta}) = \boldsymbol{\delta}^T \mathbf{r}$$

$$F(\mathbf{x};\boldsymbol{\theta}) = \sum_m \delta_m r_m$$

Boltzmann distribution

$$\Pr(\Lambda) \propto \exp(-\beta E(\Lambda|\boldsymbol{\theta})),$$

Inverse of a temperature-like parameter

Fixed continuous variables

A tractable free energy function

$$\psi_\beta(\langle\Lambda\rangle, \mathbf{u}|\boldsymbol{\theta}) = E(\langle\Lambda\rangle|\boldsymbol{\theta}) + \sum_i \sum_m \langle\delta_{im}\rangle u_{im} - \frac{1}{\beta}\sum_i \ln\left(\sum_m \exp(\beta u_{im})\right), \quad (13)$$

where $\langle\delta_{im}\rangle$ denotes the mean of δ_{im} and $\mathbf{u}$ denotes collection of auxiliary variables u_{im}.

KLD(Bullback-Leibler Divergence)
A ratio between the joint probability and the product of individual marginal probabilities of discrete binary variables

Annealed competitive learning

$$\frac{\partial \psi}{\partial \langle \delta_{im} \rangle} = 0 \quad \text{for all } i, m,$$

$$\frac{\partial \psi}{\partial u_{im}} = 0 \quad \text{for all } i, m,$$

A saddle point of KLD

Annealed KLD minimization :
Tracking the saddle point of KLD along a physical-like annealing process that schedules the temperature-like parameter from sufficiently high to low values

Wu and Hsu, Neuorcomputing 2011

Interactive Dynamics

- Mean Field Equations

Discrete variables

$$u_{im} = -\frac{\partial E}{\partial \langle \delta_{im} \rangle} = -\frac{1}{2}(\mathbf{x}_i - \mathbf{w}_m)^T \mathbf{A} (\mathbf{x}_i - \mathbf{w}_m) - \frac{\lambda}{2}(r_m - y_i)^2$$

$$\langle \delta_{im} \rangle = \frac{\exp(\beta u_{im})}{\sum_l \exp(\beta u_{il})}.$$

Updating rules

- Setting zero to $\partial\psi/\partial\mathbf{w}_m$, $\partial\psi/\partial A_{ab}$ and $\partial\psi/\partial r_m$

$$\mathbf{w}_m = \frac{\sum_i \langle\delta_{im}\rangle \mathbf{x}_i}{\sum_i \langle\delta_{im}\rangle}$$

$$\mathbf{A} = (\mathbf{B}^{-1})^T$$

$$r_m = \frac{\sum_i \langle\delta_{im}\rangle y_i}{\sum_i \langle\delta_{im}\rangle}$$

where the element in matrix B is defined by

$$B_{ab} = \frac{1}{N}\sum_i\sum_m \langle\delta_{im}\rangle (x_{ia} - w_{ma})(x_{ib} - w_{mb}).$$

Continuous variables

Appendix B. A procedure for annealed competitive learning

1 Set β sufficiently small, α near and less than one, λ positive, $\mathbf{A} = 0.01 \times \mathbf{I}$, and

$$\mathbf{w}_m = \frac{1}{N}\sum_i \mathbf{x}_i, \quad \langle \delta_{im} \rangle = \frac{1}{M}, \quad r_m = \frac{1}{N}\sum_i y_i.$$

2. If γ is less than a pre-determined threshold, apply small random perturbations to all $\langle \delta_{im} \rangle$.
3. Update all $\langle \delta_{im} \rangle$ by (14) and (15).
4. Update all $\mathbf{w}_m$ by (16).
5. Update $\mathbf{A}$ by (17).
6. Update r_m by (18).
7. $\beta \leftarrow \beta / \alpha$. If γ is close enough to one, halt, otherwise go to step 2.

optimal β

- $$\beta_{opt} = \arg\min_{\beta,\mathbf{r}} D_{S,\beta}(\boldsymbol{\theta}),$$

  where

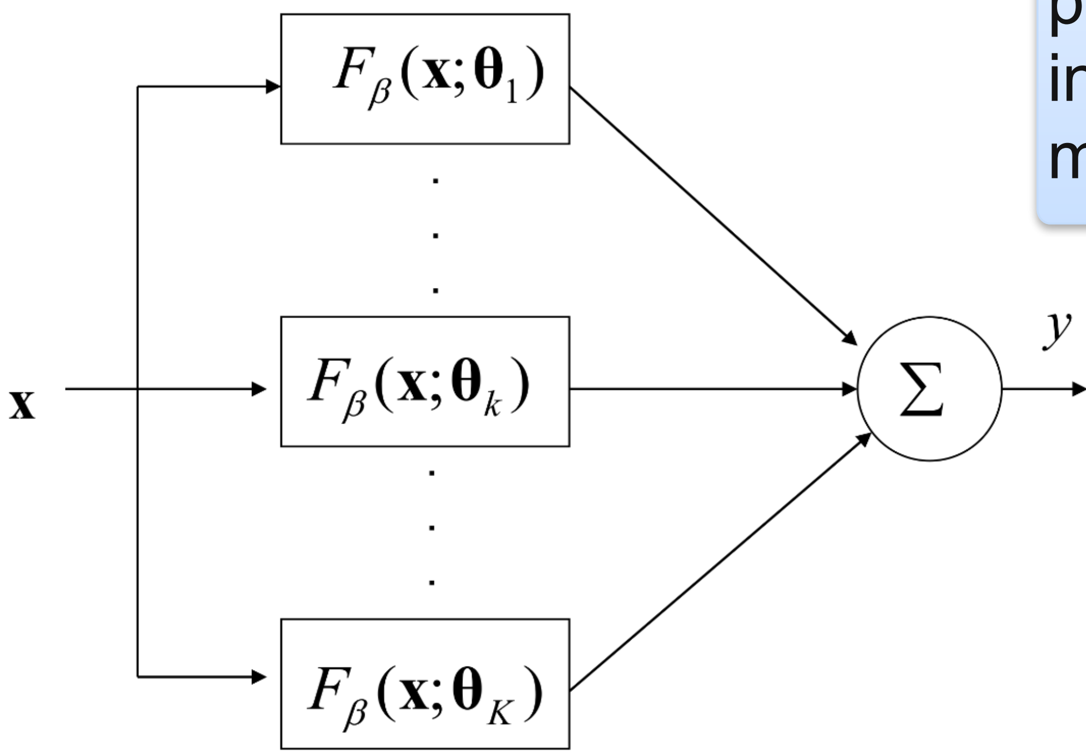
  $$D_{S,\beta}(\boldsymbol{\theta}) = \frac{1}{2}\sum_i \|y_i - F_\beta(\mathbf{x}_i;\boldsymbol{\theta})\|^2.$$

Annealed cooperative–competitive learning of multiple Mahalanobis-NRBF modules

A network of K Mahalanobis-NRBF modules

- $$G(\mathbf{x}) = \sum_k F_\beta(\mathbf{x}; \boldsymbol{\theta}_k).$$

Adaptable parameters in the kth module



Special cases

1. When $K > 1$ and $M = 1$, the denominator in (9) is ignored. In the occasion, G reduces to the mapping of a 3-layer RBF network explored in [23,24], where each hidden unit determines its external fields based on its own weight matrix, but the outputs of hidden units are not normalized.
2. When $M = 1$ and $\mathbf{A}_k = \mathbf{I}/\sigma_k^2$ for all k, G reduces to define the input–output relation of typical RBF networks explored in [6,8,9,25], where outputs of hidden units are not normalized and radial basis functions adopt Euclidean distances.

3. For $K=1$ and $M>1$, G reduces to F_β (10) that defines the mapping of a Mahalanobis-NRBF module, where external fields to hidden units are in terms of Mahalanobis distances based on a common weight matrix and all hidden units respond normalized activations.
4. When $K=1$ and $\mathbf{A}=\mathbf{I}/\sigma^2$, G reduces to characterize normalized RBF networks explored in [7,26–30], where external fields to hidden units are based on Euclidean distances.

Annealed cooperative–competitive learning

$$\widehat{y}_i[l] = F_\beta(\mathbf{x}_i; \boldsymbol{\theta}_l).$$

An individual module output

As in [36], a local target, denoted by $y_i[k]$ for module k, is set to compensate for the error of approximating y_i by the sum of the other $K-1$ modules, such as

$$y_i[k] = y_i - \sum_{l \neq k} \widehat{y}_i[l], \tag{22}$$

Cooperative mechanism of learning individual modules

Local target of an individual module

Appendix C. A procedure for annealed cooperative–competitive learning

1. Input all $(\mathbf{x}_i, y_i)$, and set β sufficiently small, α near and les
   than one and $\boldsymbol{\theta}_k$ to $\boldsymbol{\theta}_0$ for all k.
2. Determine γ_k for each k. If the mean of all γ_k is greater than
   predetermined threshold, halt.
3. Execute the following steps for each k asynchronously.
   (a) Calculate $y_i[k]$ by (22) for all i.
   (b) Employ $\{(\mathbf{x}_i, y_i[k])\}_i$ to update all $\mathbf{w}_m$, r_m and $\mathbf{A}_k$ in $\boldsymbol{\theta}_k$.
      (i) Update $\{\langle\delta_{im}\rangle\}$ by (14) and (15).
      (ii) Update all $\mathbf{w}_m$ by (16).
      (iii) Update all $\mathbf{A}_k$ by (17).
      (iv) Update all r_m by (18).
4. $\beta \leftarrow \beta/\alpha$. Go to step 2.

Nonlinear function approximation

Table 1
Target functions.

$$f_1(\mathbf{x}) = \sin(x_1 + x_2)$$
$$f_2(\mathbf{x}) = x_1^2 + x_2^2$$
$$f_3(\mathbf{x}) = 0.5x_1^2 - 0.9x_2^2$$
$$f_4(\mathbf{x}) = \exp(-0.05x_1^2 - 0.09x_2^2)$$
$$f_5(\mathbf{x}) = \sin([1, -1]^T x) + \exp(-x^T A x)$$

$$f_6(\mathbf{x}) = \tanh(0.8x_1 + 0.2x_2) + \sin(0.3x_1 - 0.9x_2)$$
$$f_7(\mathbf{x}) = 0.5\sin(x_1 + x_2) + 0.2x_1 - 0.2x_2$$
$$f_8(\mathbf{x}) = \exp(-(\mathbf{x} - \mathbf{w}_1)^T A(\mathbf{x} - \mathbf{w}_1)) + \exp(-(\mathbf{x} - \mathbf{w}_2)^T B(\mathbf{x} - \mathbf{w}_1))$$

$$f_9(\mathbf{x}) = f_8(\mathbf{x}) + 0.5\ \sin(x_1 + 0.3x_2) + 0.5\ \sin(0.2x_1 - 0.8x_2)$$
$$f_{10}(x) = \sin(x_1 + x_2 + x_3) + \cos(x_1 + x_2 + x_3)$$
$$f_{11}(x) = \tanh(x_1 + x_2 + x_3 + x_4)$$

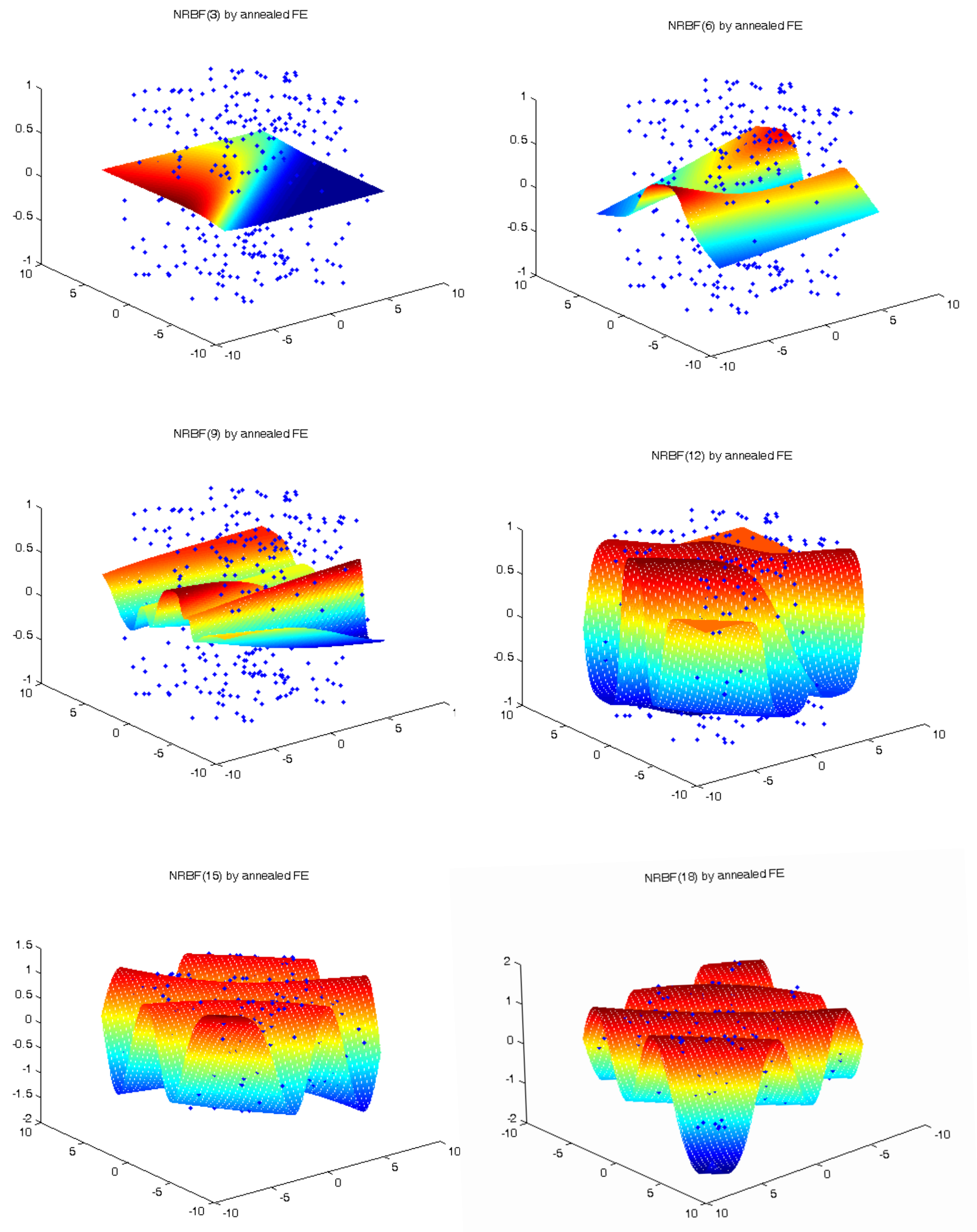

NRBF(3) by annealed FE
NRBF(6) by annealed FE
NRBF(9) by annealed FE
NRBF(12) by annealed FE
NRBF(15) by annealed FE
NRBF(18) by annealed FE

Figure 4

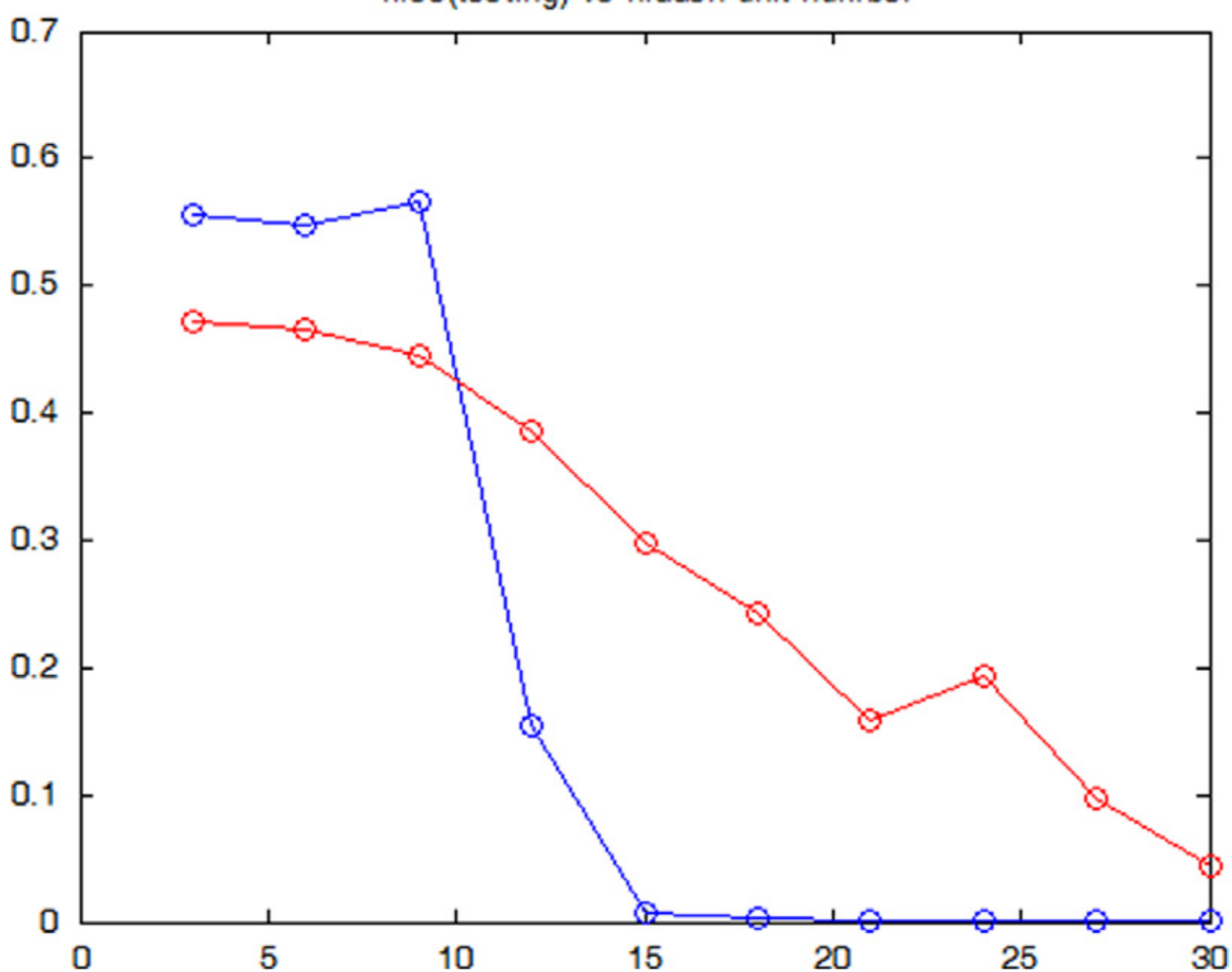


Fig. 5. Mean square testing errors of annealed competitive learning (blue curve) and the Rätsch method (red curve) in approximating f_1 versus the numbers of hidden units. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Table 2
Quantitative performances of the five relevant methods in approximating the first four target functions.

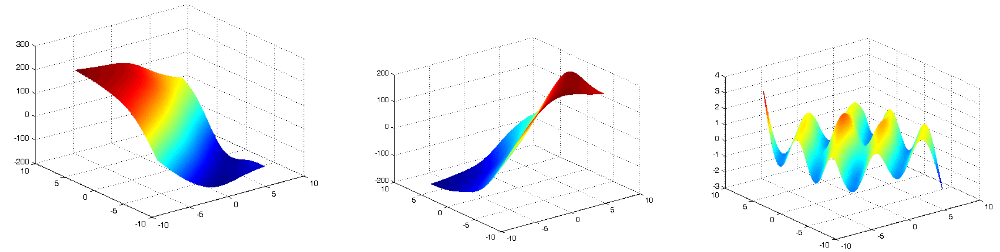
	Mean square error			
	Mahalanobis-NRBF ($K=1$, $M=41$)	Euclidean-RBF (Rätsch, $M=41$)	Euclidean-RBF (LM, $M=41$)	MLP (LM, $M=41$)
Training				
f_1	$3.4e-5 \pm 0$	$1.4e-2 \pm 1.1e-5$	$1.8e-3 \pm 9.5e-7$	$9.6e-5 \pm 8.2e-9$
f_2	$1.2e-3 \pm 4.5e-7$	$4.1e-1 \pm 4.7e-3$	$1.4e0 \pm 2.5e-1$	$2.3e-2 \pm 4.9e-5$
f_3	$1.0e-3 \pm 3.4e-7$	$8.5e-2 \pm 2.0e-4$	$2.2e-1 \pm 1.1e-3$	$4.9e-3 \pm 1.6e-6$
f_4	$2.7e-3 \pm 0$	$2.5e-3 \pm 0$	$2.7e-3 \pm 0$	$3.4e-3 \pm 1.2e-7$
Testing				
f_1	$7.4e-5 \pm 0$	$2.3e-2 \pm 3.0e-5$	$6.4e-3 \pm 4.9e-6$	$1.8e-3 \pm 1.1e-5$
f_2	$4.0e-3 \pm 4.6e-6$	$1.7e0 \pm 7.3e-2$	$6.0e0 \pm 3.1e0$	$3.4e-2 \pm 2.8e-4$
f_3	$1.5e-3 \pm 6.5e-7$	$1.2e-1 \pm 2.2e-4$	$6.2e-1 \pm 1.1e-2$	$7.0e-3 \pm 8.4e-6$
f_4	$3.7e-3 \pm 0$	$4.3e-3 \pm 0$	$4.1e-3 \pm 0$	$4.0e-3 \pm 2.9e-7$

	Mean square error	
	Mahalanobis-NRBF ($K=1$, $M=41$)	Euclidean-RBF (Rätsch, $M=41$)
Training		
f_1	$3.4e-5 \pm 0$	$1.4e-2 \pm 1.1e-5$
f_2	$1.2e-3 \pm 4.5e-7$	$4.1e-1 \pm 4.7e-3$
f_3	$1.0e-3 \pm 3.4e-7$	$8.5e-2 \pm 2.0e-4$
f_4	$2.7e-3 \pm 0$	$2.5e-3 \pm 0$
Testing		
f_1	$7.4e-5 \pm 0$	$2.3e-2 \pm 3.0e-5$
f_2	$4.0e-3 \pm 4.6e-6$	$1.7e0 \pm 7.3e-2$
f_3	$1.5e-3 \pm 6.5e-7$	$1.2e-1 \pm 2.2e-4$
f_4	$3.7e-3 \pm 0$	$4.3e-3 \pm 0$

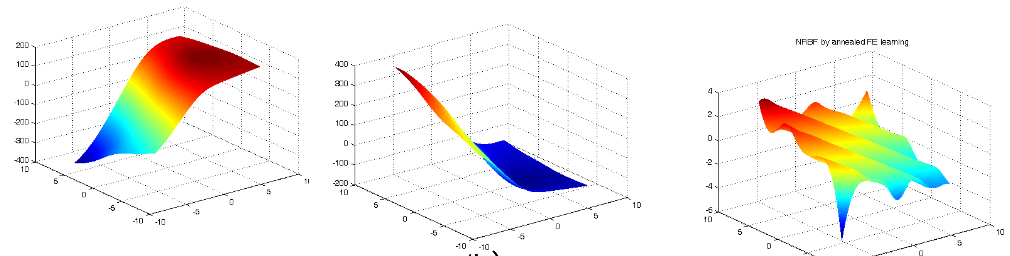
Table 3
Quantitative performances of the relevant methods for approximating f_5–f_8.

	Mean square error				
	Mahalanobis-NRBF (K=2, M=41)	Euclidean-RBF (Rätsch, M=41)	Euclidean-RBF (LM, M=41)	MLP (LM, M=41)	MLP (BP, M=41)
Training					
f_5	1.0e−5 ± 0	1.4e−2 ± 1.6e−5	4.4e−3 ± 3.9e−6	1.5e−3 ± 4.1e−6	9.8e−2 ± 1.2e−2
f_6	1.7e−5 ± 2.2e−11	1.5e−3 ± 5.3e−8	5.8e−4 ± 4.8e−8	3.2e−4 ± 9.2e−9	3.5e−2 ± 5.8e−6
f_7	4.1e−5 ± 0	2.7e−3 ± 1.7e−7	3.1e−3 ± 3.4e−7	1.3e−3 ± 7.9e−7	1.1e−1 ± 2.4e−5
f_8	4.0e−7 ± 0	5.4e−6 ± 0	1.4e−4 ± 0	6.9e−5 ± 5.3e−10	1.9e−3 ± 1.6e−7
Testing					
f_5	2.4e−4 ± 1.6e−7	2.8e−2 ± 3.4e−5	1.9e−2 ± 6.8e−5	2.2e−3 ± 6.4e−6	1.3e−1 ± 1.8e−2
f_6	6.3e−5 ± 4.2e−10	3.0e−3 ± 5.6e−7	1.2e−3 ± 1.3e−7	4.2e−4 ± 1.7e−8	4.9e−2 ± 8.4e−6
f_7	4.3e−4 ± 4.2e−8	4.5e−3 ± 9.5e−7	9.2e−3 ± 9.1e−7	2.7e−3 ± 5.2e−6	1.2e−1 ± 1.8e−5
f_8	2.4e−6 ± 0	7.4e−6 ± 0	2.8e−4 ± 1.6e−9	1.0e−4 ± 4.9e−10	2.2e−3 ± 5.4e−7

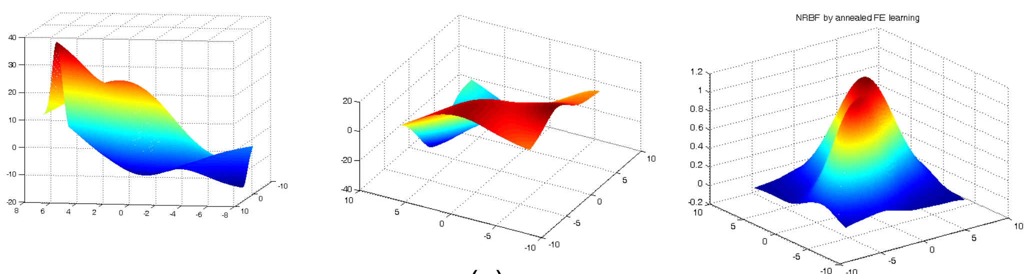
	Mean square error	
	Mahalanobis-NRBF ($K=2$, $M=41$)	Euclidean-RBF (Rätsch, $M=41$)
Training		
f_5	$1.0e-5 \pm 0$	$1.4e-2 \pm 1.6e-5$
f_6	$1.7e-5 \pm 2.2e-11$	$1.5e-3 \pm 5.3e-8$
f_7	$4.1e-5 \pm 0$	$2.7e-3 \pm 1.7e-7$
f_8	$4.0e-7 \pm 0$	$5.4e-6 \pm 0$
Testing		
f_5	$2.4e-4 \pm 1.6e-7$	$2.8e-2 \pm 3.4e-5$
f_6	$6.3e-5 \pm 4.2e-10$	$3.0e-3 \pm 5.6e-7$
f_7	$4.3e-4 \pm 4.2e-8$	$4.5e-3 \pm 9.5e-7$
f_8	$2.4e-6 \pm 0$	$7.4e-6 \pm 0$



(a)



(b)



(c)

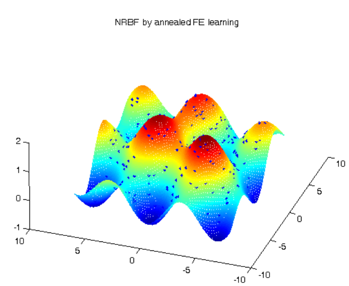
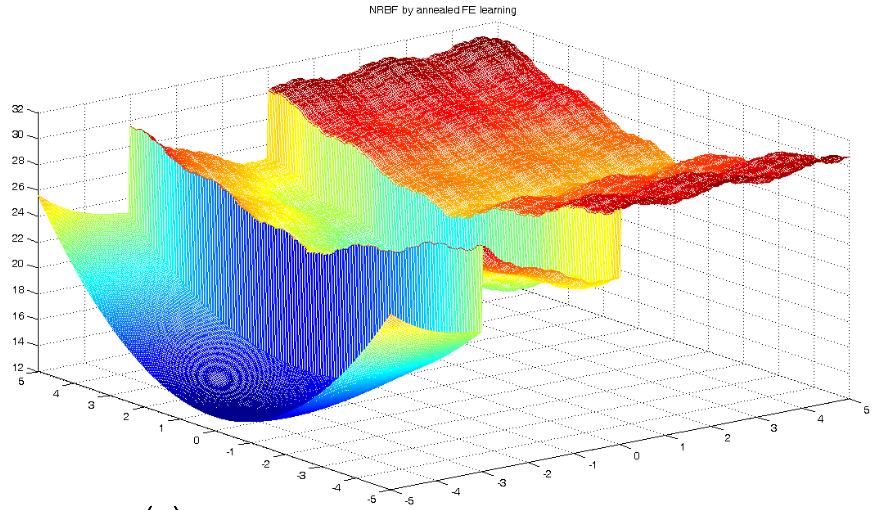
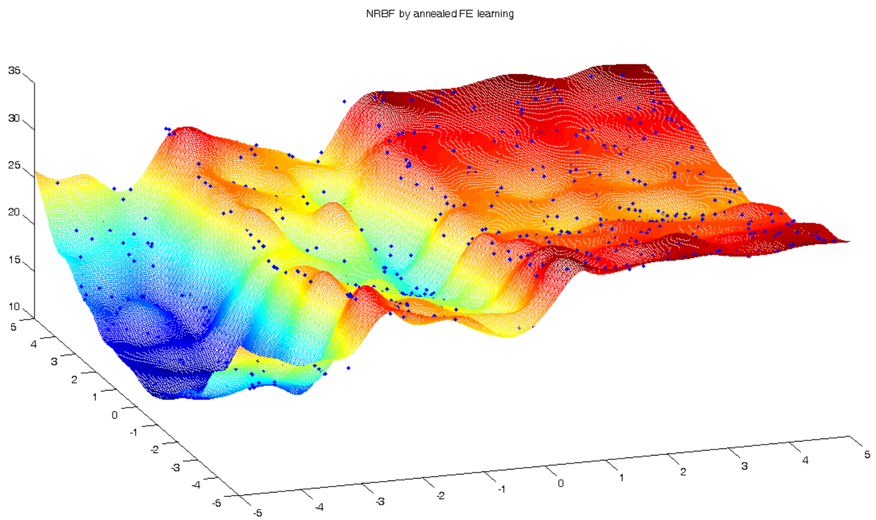


Figure 6

(d)


NRBF by annealed FE learning
32
30
28
26
24
22
20
18
16
14
12
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
-4
-3
-2
-1
0
1
2
3
4
5

(a)


NRBF by annealed FE learning
35
30
25
20
15
10
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
-4
-3
-2
-1
0
1
2
3
4
5

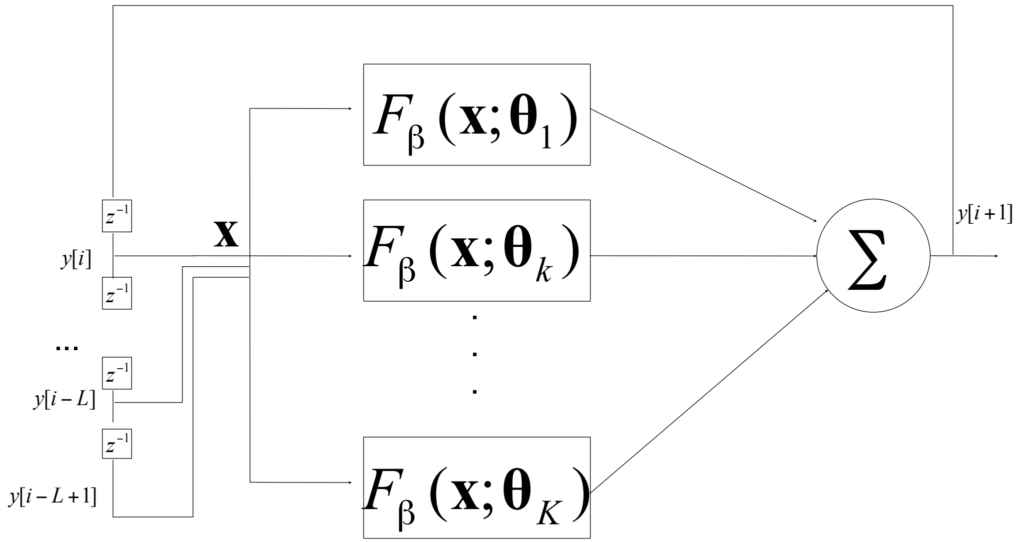
(b)

Figure 7

Chaotic differential function approximation

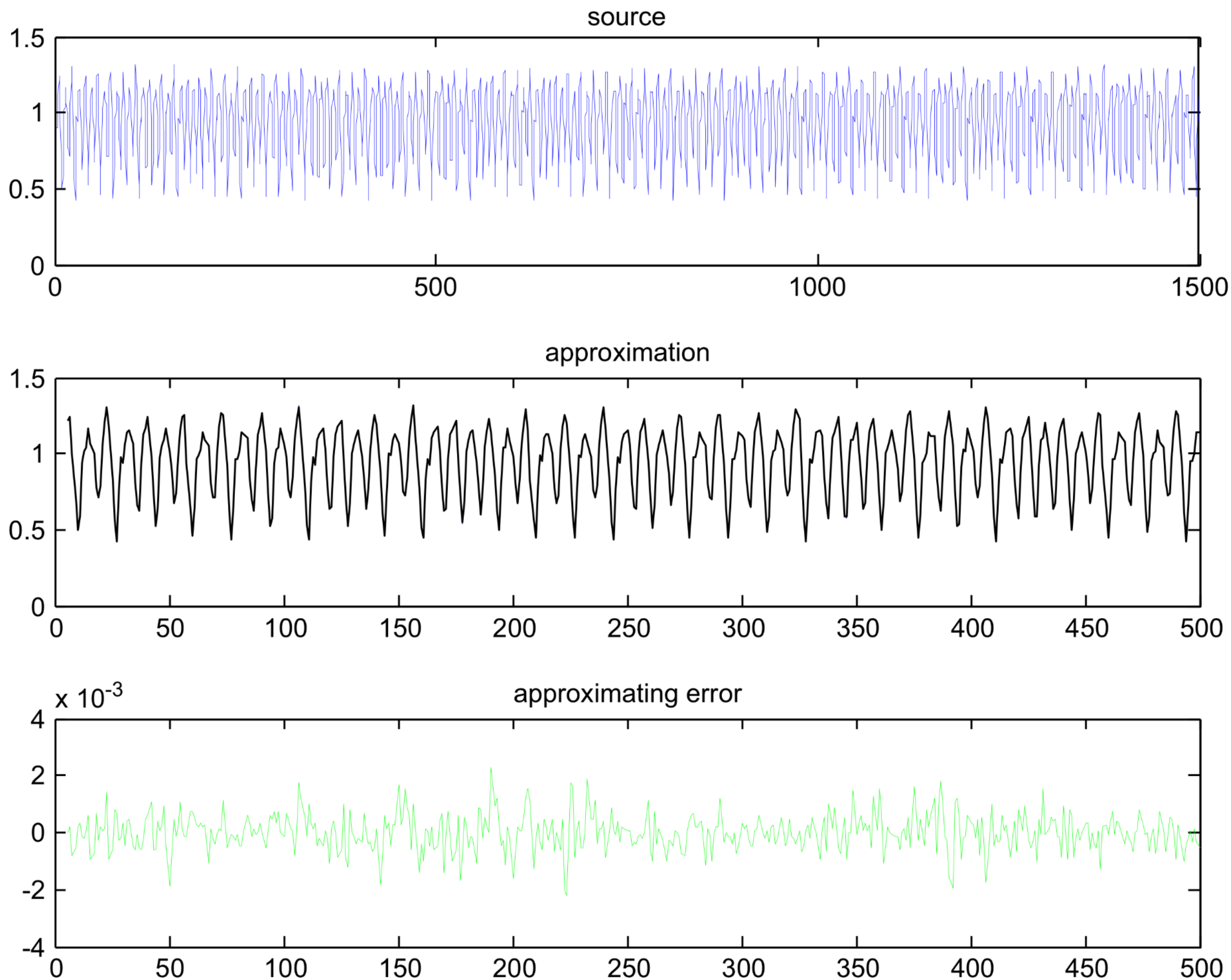
$$\frac{\partial x}{\partial t} = \frac{ax(t-\tau)}{1+x^{c}(t-\tau)} - bx(t),$$

$\tau = 17$, $a = 0.2$, $c = 10$ and $b = 0.1$



$$o_t = f(\mathbf{x}_t = (o_{t-L}, o_{t-L+1}, \ldots, o_{t-1})^T),$$

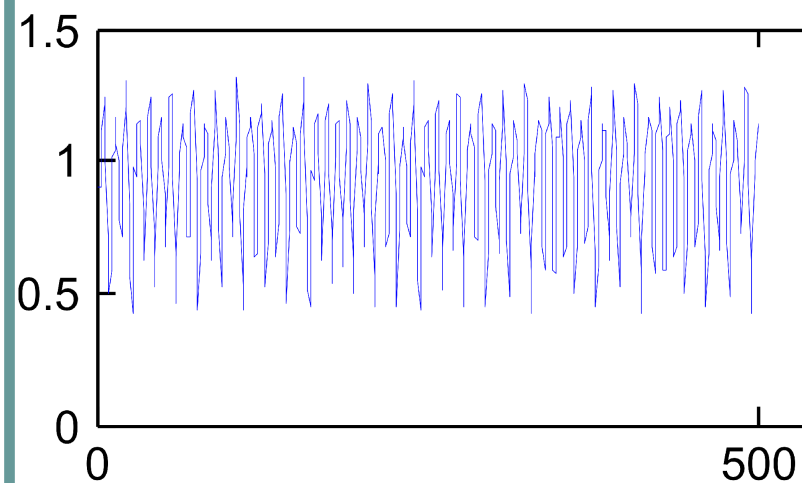
Figure 9

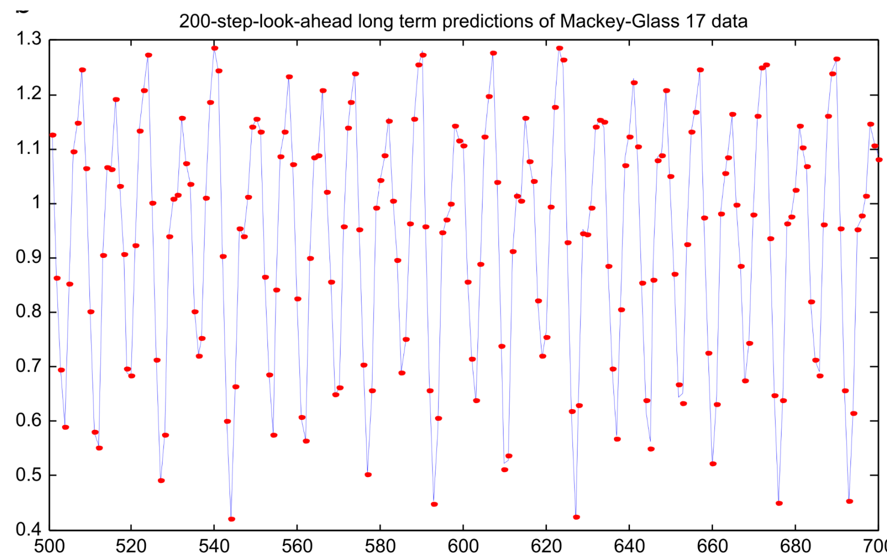


source
1.5
1
0.5
0
0
500
1000
1500
approximation
1.5
1
0.5
0
0
50
100
150
200
250
300
350
400
450
500
approximating error
x 10^-3
4
2
0
-2
-4
0
50
100
150
200
250
300
350
400
450
500

Data driven long-term prediction

- MG(Mackey–Glass) 17 generated by RK(Runge-Kutta) 4



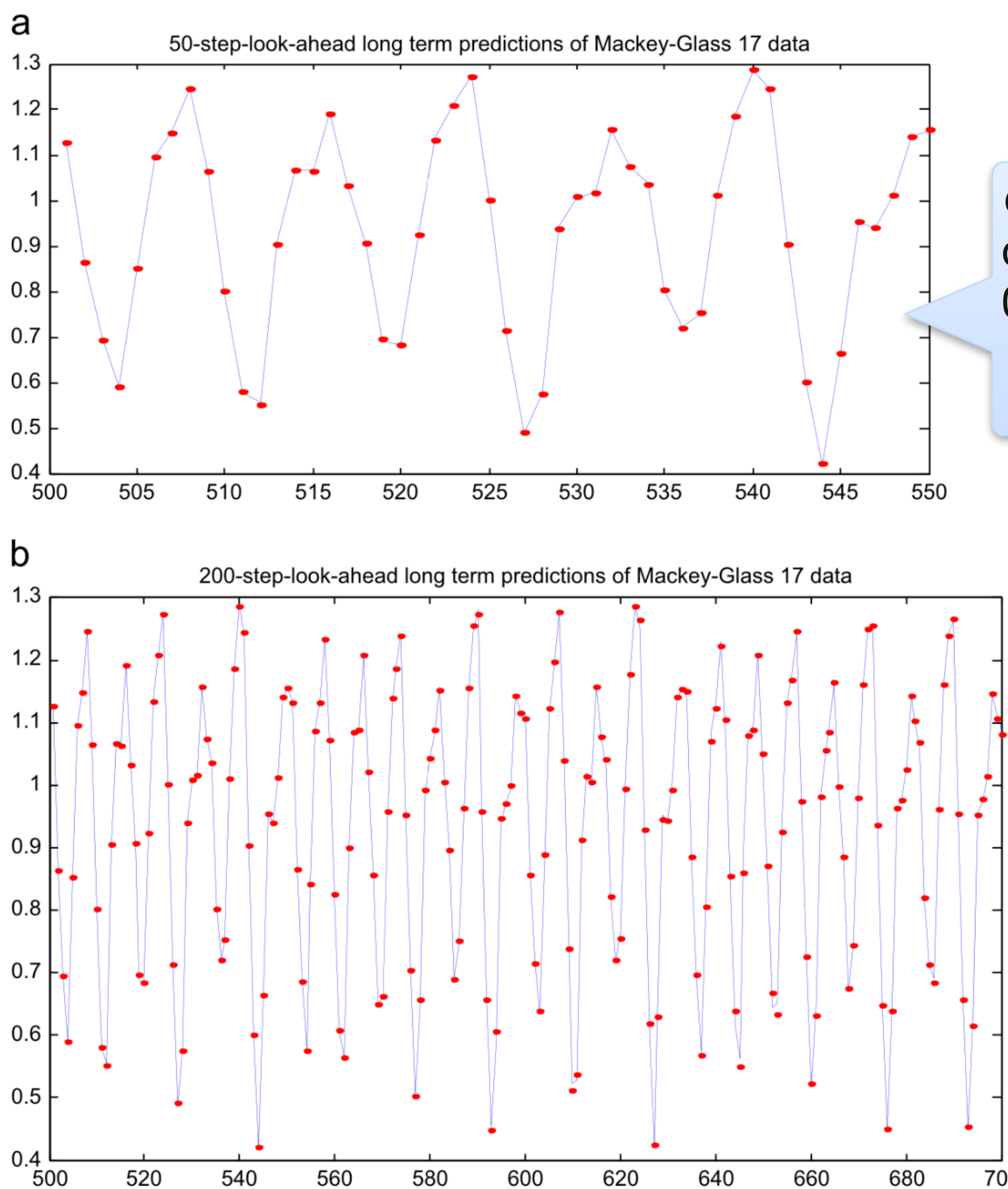


200-step-look-ahead prediction.

Approach	mse_{S_1}	mse_{S_2}	$D_{O_2}^{50}$	
	mean ± var	mean ± var	mean ± var	min
Mahalanobis-NRBF modules (K=3)	4.68e−7 ± 0	2.22e−6 ± 0	5.00e−3 ± 1.77e−5	2.18e−3
MLP-LM (8)	5.22e−5 ± 3.51e−10	6.20e−5 ± 6.77e−10	4.08e−2 ± 2.92e−5	2.95e−2
MLP-LM (15)	4.61e−5 ± 0	5.31e−5 ± 1.51e−10	4.28e−2 ± 1.07e−4	3.19e−2
RBF (Rätsch, M=30)	8.92e−5 ± 2.41e−9	7.10e−5 ± 2.06e−9	5.51e−2 ± 1.13e−4	2.30e−2

Table 5
The performances of the three relevant methods for n_2-step-look-ahead long term predictions with n_2 ranging from 50 to 200.

Approach	$D_{O_2}^{50}$	$D_{O_2}^{80}$	$D_{O_2}^{100}$	$D_{O_2}^{150}$	$D_{O_2}^{200}$
Mahalanobis-NRBF modules ($K = 3$)	2.18e−3	2.436e−3	3.44e−3	5.80e−3	5.78e−3
MLP-LM (8)	2.95e−2	7.17e−2	1.52e−1	2.15e−1	2.51e−1
MLP-LM (15)	3.19e−2	6.44e−2	1.49e−1	2.12e−1	2.49e−1
RBF (30)	2.30e−2	3.54e−2	6.66e−2	1.21e−1	1.82e−1



correlation coefficient 0.9999

correlation coefficient 0.9993

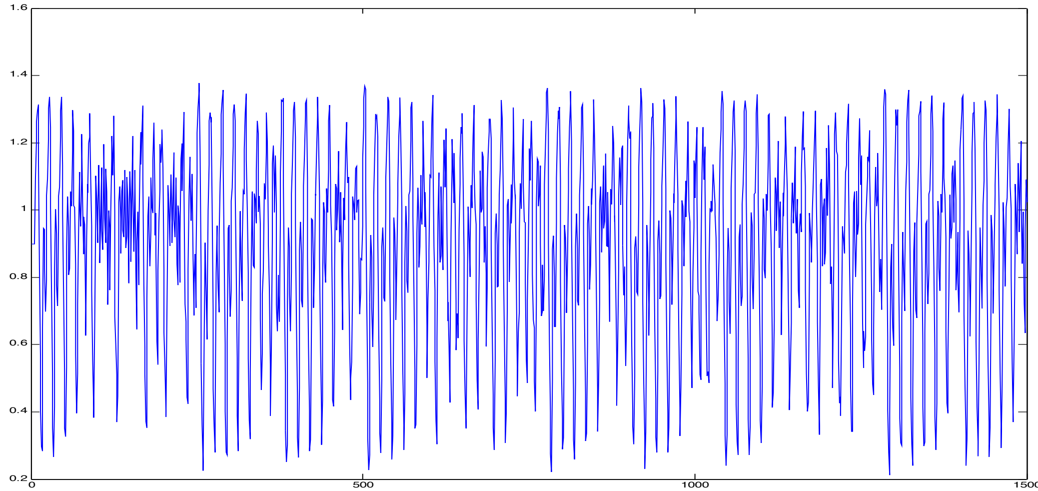
Fig. 10. n-step-look-ahead predictions of MG17 time series with n = 50 and n = 200 by annealed cooperative–competitive learning with K=2. (For interpretation of the

Mackey-Glass 30

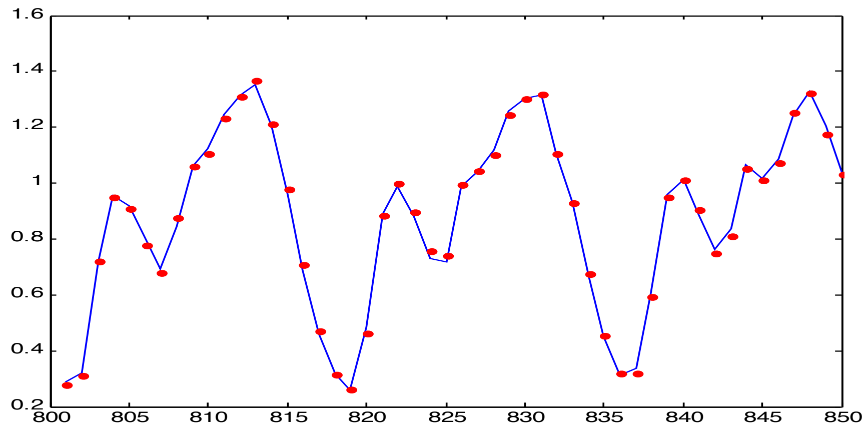
$$\frac{\partial x}{\partial t} = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t),$$

$\tau = 30$ $a = 0.2$, $c = 10$ and $b = 0.1$

Mackey-Glass 30 data



50-step-look-ahead long term predictions of Mackey-Glass 30 data



correlation coefficient 0.999

Figure 11

CDFA: Nonlinear delay differential equations

$$\frac{\partial x}{\partial t} = x(t-\tau) - x^3(1-\tau),$$

where the delav τ is set to 1.6.

J.C. Sprott, A simple chaotic delay differential equation, Phys. Lett. A 366 (2007) 397–402.

Table 6
Quantitative performances of learning multilayer neural networks for data driven forecasts of chaotic time series oriented from the delay differential equation (27).

Approach	mse_{S_1}	mse_{S_2}	$D_{O_2}^{n_2}$	
	mean ± var	mean ± var	mean ± var	min
Mahalanobis-NRBF modules ($K=3$)	3.0e−6 ± 0	7.0e−6 ± 0	1.3e−2 ± 1.0e−4	4.4e−3
MLP-LM (8)	4.8e−4 ± 9.5e−8	5.3e−4 ± 1.0e−7	1.7e−1 ± 1.7e−2	2.7e−2
MLP-LM (15)	8.4e−5 ± 4.0e−9	5.8e−5 ± 1.0e−9	3.3e−2 ± 3.6e−4	1.2e−2
RBF (Rätsch, $M=30$)	1.5e−3 ± 7.3e−8	1.7e−3 ± 3.5e−7	1.0e−1 ± 3.4e−4	7.3e−2

Conclusions and discussions

- Numerical simulations have shown annealed cooperative– competitive learning of the proposed multi-module Mahalano- bis-NRBF network effective and reliable for nonlinear function approximation and long term prediction of chaotic time series.

- The reliability and effectiveness of the proposed approach for nonlinear and chaotic differential function approximation relies on the success of annealed compe- titive–cooperative learning of multiple Mahalanobis-NRBF mod- ules, which introduce a system of manifold Mahalanobis distances

- In architecture, the proposed multi-module Mahalanobis-NRBF network spans a general functional scope well covering most existing networks of radial basis functions.