

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

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

# Outline

- Radial Basis Function (RBF)
  - RBF networks
  - NRBF (Normalized Radial Basis Functions)
- Introduction
- A model-oriented Mahalanobis-NRBF neural module
- Annealed KLD minimization
- Annealed cooperative–competitive learning of multiple Mahalanobis-NRBF
- Nonlinear function approximations
- Chaotic differential function approximation
- Conclusions

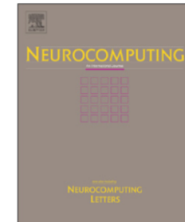


ELSEVIER

Contents lists available at [ScienceDirect](http://www.sciencedirect.com)

## Neurocomputing

journal homepage: [www.elsevier.com/locate/neucom](http://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 Mahalanobis normalized radial basis functions (NRBF) with applications to nonlinear function approximation and chaotic differential function approximation. A multilayer neural network is extended to be composed of multiple Mahalanobis-NRBF modules. Each module activates normalized outputs of radial basis functions, determining Mahalanobis radial distances based on its own adaptable weight matrix. An essential cooperative scheme well decomposes learning a multi-module network to sub-tasks of learning individual modules. Adaptable network interconnections are asynchronously updated module-by-module based on annealed cooperative–competitive learning for function approximation under a physical-like mean-field annealing process. Numerical simulations show outstanding performance of annealed cooperative–competitive learning of a multi-module Mahalanobis-NRBF network for nonlinear function approximation and long term look-ahead prediction of chaotic time series.

© 2014 Elsevier B.V. All rights reserved.

- [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

July 1989

C.B.I.P. Paper No. 31

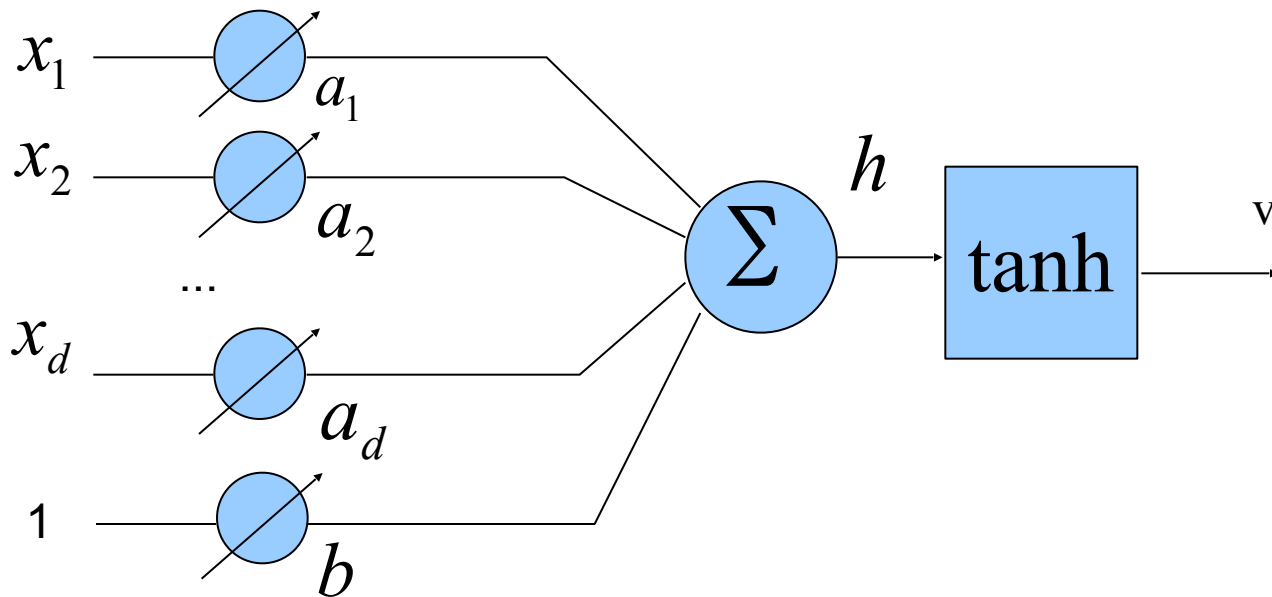
**A Theory of Networks for Approximation and Learning**

**Tomaso Poggio and Federico Girosi**

# Perceptrons

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

$$v = \tanh(h = a_1x_1 + a_2x_2 + \dots + a_dx_d + b)$$



# RBF Network function

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

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

*Network* parameter

$$\theta = \{\mathbf{w}_i\}_i \cup \{\boldsymbol{\mu}_i\}_i \cup \{\sigma_i\}_i$$

Normalized RBF

$A=I$

Euclidean Distance

Normalization

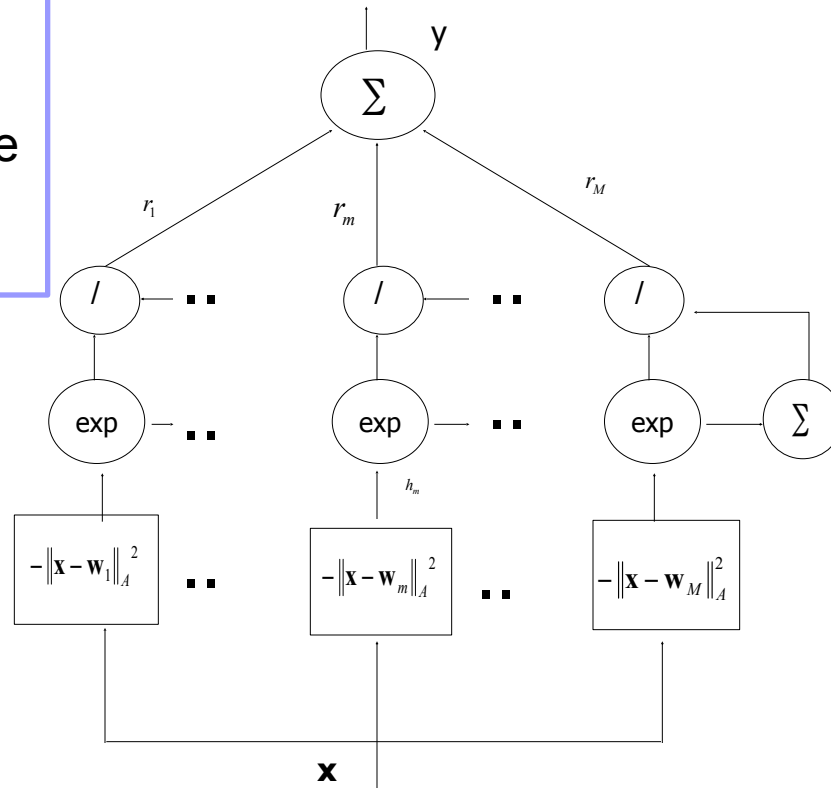


Figure 2



# Mahalanobis-NRBF modules

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

## A generative model for paired predictors targets

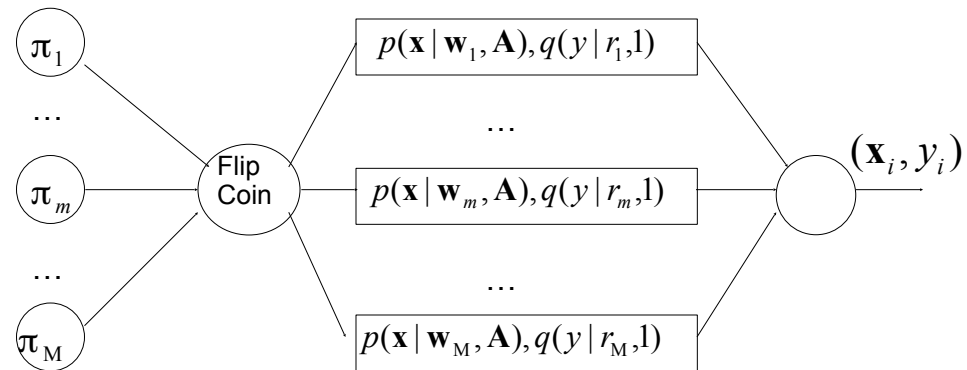


Figure 1

A generative model theoretically characterizes data formation [15,16]. Here it is organized with multiple sub-models, each consisting of paired normal random variables respectively generating paired predictors and targets. Each pair of predictor and target is oriented from one and only one sub-model. Given training data are mixtures of samples from joined sub-models. By Potts encoding of exclusive memberships [15,16], fitting a generative model could be decomposed to sub-tasks of fitting joined sub-models and formulated as a mixed integer programming, which involves constrained optimization with respect to discrete integer variables and continuous model parameters. Since the fitting criterion is not differentiable, its optimization with respect to discrete and continuous variables is resolved by annealed Kullback–Leibler divergence (KLD) minimization, which has been devised for solving self-organization [15,17] and classification [16].

# A network of multiple Mahalanobis-NRBF modules

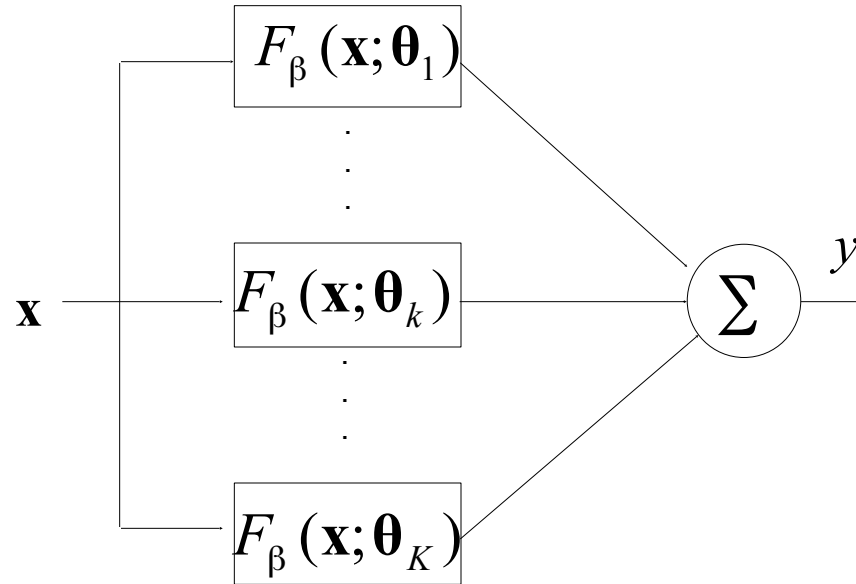


Figure 3

# A Mahalanobis-NRBF module

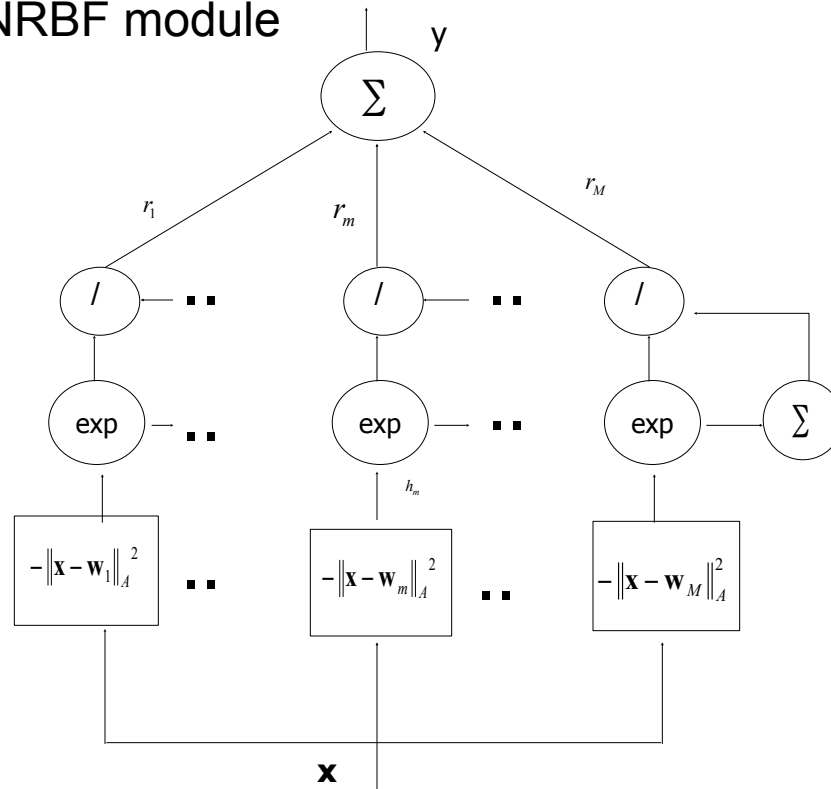


Figure 2

# Annealed Cooperative Learning

- The proposed cooperative learning asynchronously updates network interconnections module-by-module under a physical-like mean-field annealing process. By an essential cooperative learning scheme the local target of learning an individual module is set to compensate the error of approximating the global target by outputs of the remaining modules. Asynchronous updating ensures minimizing the global error by refining each individual module of minimizing local errors. By the proposed annealed cooperative-competitive learning, the model-oriented multi-module architecture is shown feasible for resolving nonlinear function approximation and chaotic differential function approximation.

# paired normal random variables

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

and

$$\begin{aligned} q_m(y) &\equiv q(y|r_m, 1) \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y - r_m)^2}{2}\right), \end{aligned} \quad (2)$$

where  $d$  is the input dimension,  $T$  denotes matrix transpose,  $\mathbf{A}$  is a positive definite  $d \times d$  matrix that denotes the inverse of a common covariance matrix, and  $\mathbf{w}_m$  and  $r_m$  respectively denote means of predictors and targets in a sub-model.

Each time a generative model randomly selects one of joined sub-models according to a set of prior probabilities, triggering to produce paired instances,  $(\mathbf{x}_i, y_i)$ , where  $\mathbf{x}_i \in R^d$  and  $y_i \in R$ . Each  $(\mathbf{x}_i, y_i)$  possesses its exclusive membership to joined sub-models. All generated paired data form a training or testing set. The exclusive membership of  $(\mathbf{x}_i, y_i)$  to joined sub-models is encoded by a multi-state Potts variable [15,16,18–22]. Potts encoding facilitates decomposition of fitting a generative model to sub-tasks of fitting individual sub-models. By Potts encoding this work translates model fitting to a mixed integer programming that aims to minimize an energy function with respect to dependent variables, including built-in model parameters and discrete exclusive memberships. Since the energy function contains discrete variables, it is not differentiable. Its minimization could not be directly resolved by typical gradient-based methods.



As in previous works [15,16], sub-models in a generative model share a common covariance matrix  $\mathbf{A}^{-1}$ . This helps expressing expected exclusive memberships in terms of Mahalanobis distances and associating the conditional expectation of target  $y$  to  $\mathbf{x}$  with the input-output relation of a Mahalanobis-NRBF module. If  $\mathbf{A}$  is diagonal,  $p_m$  is factorial and components of predictors are statistically independent. Otherwise  $\mathbf{A}$  plays a role of compensating statistical dependency among components of predictors, serving as the sole weight matrix of measuring Mahalanobis radial distances in a Mahalanobis-NRBF neural module.

# A Mahalanobis-NRBF neural module

- Conditional probability

$$\begin{aligned} h_m &\equiv \|\mathbf{x} - \mathbf{w}_m\|_{\mathbf{A}}^2 \\ &= (\mathbf{x} - \mathbf{w}_m)^T \mathbf{A} (\mathbf{x} - \mathbf{w}_m). \end{aligned}$$

Let  $\boldsymbol{\delta} = (\delta_1, \dots, \delta_M)^T$  be a Potts variable whose possible outcomes belong  $\Xi_M = \{\mathbf{e}_m\}_m$ , where  $\mathbf{e}_m$  denotes a unitary binary vector with the  $m$ th bit one and others zeroes. It follows  $\delta_m$  belongs  $\{0, 1\}$  and the sum of binary elements in  $\boldsymbol{\delta}$  equals one. The exclusive membership  $\boldsymbol{\delta}$  of  $\mathbf{x}$  is encoded by  $\mathbf{e}_{m^*}$  where

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

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

# Conditional expectation



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

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

# Overlapping membership

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

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

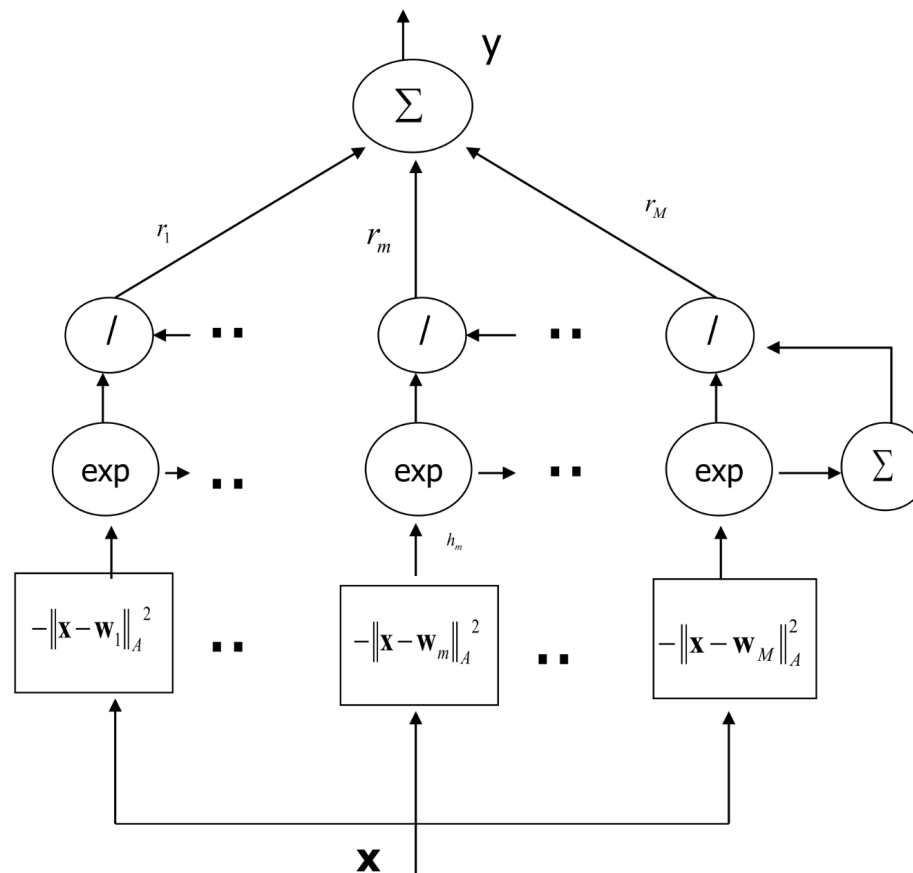
$$\sum_m v_m = 1,$$

it follows

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

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

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





# Annealed KLD minimization

- Mathematical frameworks
- Annealed competitive learning

# design cost

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

# Fitting a generative model

Let  $\boldsymbol{\delta}_i = (\delta_{i1}, \dots, \delta_{iM})^T$  be a Potts variable that encodes the exclusive membership of  $\mathbf{x}_i$  to  $M$  joined sub-models, where  $\delta_{im}$  is binary and  $\boldsymbol{\delta}_i$  belongs  $\Xi_M$ . Encoding  $\boldsymbol{\delta}_i$  to  $\mathbf{e}_m$  means that  $\mathbf{x}_i$  is closest to  $\mathbf{w}_m$  and oriented from the  $m$ th sub-model. Let  $X_m = \{\mathbf{x}_i | \boldsymbol{\delta}_i = \mathbf{e}_m\}$  denote collection of predictors with memberships identical to  $\mathbf{e}_m$ . Let  $L_m$  denote the log likelihood of fitting  $p_m$  to  $X_m$ . Summing up all negative  $L_m$  induces a fitting criterion

$$E_1 = - \sum_m L_m,$$

# a mixed energy function

$$\begin{aligned} E(\Lambda, \boldsymbol{\theta}) &= E_1 + \lambda E_2 \\ &= \frac{1}{2} \sum_i \sum_m \delta_{im} (\mathbf{x}_i - \mathbf{w}_m)^T \mathbf{A} (\mathbf{x}_i - \mathbf{w}_m) - \frac{N}{2} \log |\mathbf{A}| + \frac{\lambda}{2} \sum_i \sum_m \delta_{im} (r_m - y_i)^2, \end{aligned} \quad (12)$$

where  $\lambda$  is non-negative and  $\Lambda$  denotes collection of exclusive memberships. Minimizing  $E$  subject to constraints,

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

# Boltzmann distribution

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

where  $\beta$  denotes the inverse of a temperature-like parameter and  $\Lambda|\boldsymbol{\theta}$  denotes the case of fixing  $\boldsymbol{\theta}$ .  $\Lambda$  is composed of  $N$   $M$ -state random variables. The summation over the configuration space of  $\Lambda$  is intractable in computation for large  $M$  and  $N$ . A factorial form is considered. The joint pdf of  $\Lambda$  is approximated by the product of marginal pdfs of all  $\delta_{im}$ . The factorization is realized by minimizing [17] the Kullback–Leibler divergence (KLD) that measures the expected ratio of the product of marginal pdfs to the joint pdf. Since marginal pdfs of all  $\delta_{im}$  can be totally determined by expectations of all  $\delta_{im}$ , the KLD induces a tractable free energy function. Relevant derivations have been presented in the previous

# 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  $\delta_{im}$  and  $\mathbf{u}$  denotes collection of auxiliary variables  $u_{im}$ . Minimizing the KLD along an annealing process has been shown feasible for resolving the mixed integer programming [17].  $\psi_{\beta}$  is differentiable with respect to means of Potts variables as well as model parameters. Its saddle point can be determined by iteratively executing multiple sets of interactive dynamics. Under an annealing process, a free energy function eventually evolves to recover the original mixed energy function. Similar tractable free energy functions have been formulated in previous works [6,15–19] for solving complex tasks in the field of neural networks.

# Annealed competitive learning

The proposed annealed KLD minimization mainly tracks the saddle point of  $\psi_\beta$  under an annealing process. Since  $\beta$  modulates the freedom degree of random variables, the contribution of entropies to  $\psi_\beta$  (13) inversely proportional to  $\beta$  vanishes at sufficiently large  $\beta$ , where  $\psi_\beta$  reduces to recover original  $E$ . An annealed competitive learning addresses on tracking memberships from overlapping to exclusive forms under an annealing process, which schedules  $\beta$  from sufficiently small to high values to emulate physical annealing as in [18,19].  $\psi$  is differentiable with respect to individual expectations in  $\langle \Lambda \rangle$  as well as model parameters in  $\theta$ . By setting

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

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

# Interactive Dynamics

$$u_{im} = -\frac{\partial E}{\partial \langle \delta_{im} \rangle} = -\frac{1}{2}(\mathbf{x}_i - \mathbf{w}_m)^T \mathbf{A}(\mathbf{x}_i - \mathbf{w}_m) - \frac{\lambda}{2}(r_m - y_i)^2, \quad (14)$$

$$\langle \delta_{im} \rangle = \frac{\exp(\beta u_{im})}{\sum_l \exp(\beta u_{il})}. \quad (15)$$

The mean configuration  $\langle \Lambda \rangle$  determined by (14)–(15) under each intermediate  $\beta$  feedbacks to refine model parameters in  $\theta$ . Setting zero to derivatives,  $\partial \psi / \partial \mathbf{w}_m$ ,  $\partial \psi / \partial A_{ab}$  and  $\partial \psi / \partial r_m$  derived in [Appendix A](#), leads to the following updating rules:

$$\mathbf{w}_m = \frac{\sum_i \langle \delta_{im} \rangle \mathbf{x}_i}{\sum_i \langle \delta_{im} \rangle} \quad (16)$$

$$\mathbf{A} = (\mathbf{B}^{-1})^T \quad (17)$$

$$r_m = \frac{\sum_i \langle \delta_{im} \rangle y_i}{\sum_i \langle \delta_{im} \rangle} \quad (18)$$

where the element in matrix  $B$  is defined by

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



# optimal $\beta$

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

where

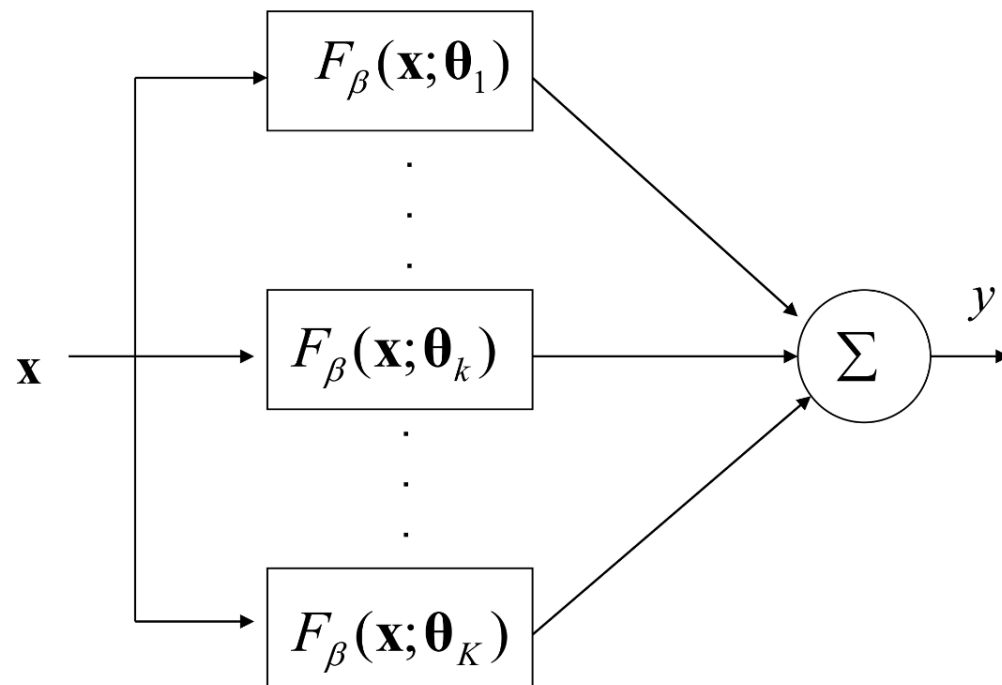
$$D_{S, \beta}(\boldsymbol{\theta}) = \frac{1}{2} \sum_i \|y_i - F_{\beta}(\mathbf{x}_i; \boldsymbol{\theta})\|^2.$$

# Annealed cooperative– competitive learning of multiple Mahalanobis-NRBF modules

# A multi-module Mahalanobis-NRBF network

The multi-module network in Fig. 3 consists of  $K$  Mahalanobis-NRBF modules. Let  $\theta_k$  denote collection of parameters in the  $k$ th Mahalanobis-NRBF module. The mapping by a multi-module network is expressed by

$$G(\mathbf{x}) = \sum_k F_\beta(\mathbf{x}; \theta_k).$$



(21)

# 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  $F_\beta$  (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

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

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

$$y_i[k] = y_i - \sum_{l \neq k} \hat{y}_i[l], \quad (22)$$

## Appendix B. A procedure for annealed competitive learning

- 1 Set  $\beta$  sufficiently small,  $\alpha$  near and less than one,  $\lambda$  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  $\gamma$  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  $\gamma$  is close enough to one, halt, otherwise go to step 2.

## Appendix C. A procedure for annealed cooperative-competitive learning

1. Input all  $(\mathbf{x}_i, y_i)$ , and set  $\beta$  sufficiently small,  $\alpha$  near and less than one and  $\theta_k$  to  $\theta_0$  for all  $k$ .
2. Determine  $\gamma_k$  for each  $k$ . If the mean of all  $\gamma_k$  is greater than a 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  $\theta_k$ .
    - (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.

---

$$f_1(\mathbf{x}) = \sin(x_1 + x_2)$$

$$f_2(\mathbf{x}) = x_1^2 + x_2^2$$

$$f_3(\mathbf{x}) = 0.5x_1^2 - 0.9x_2^2$$

$$f_4(\mathbf{x}) = \exp(-0.05x_1^2 - 0.09x_2^2)$$

$$f_5(\mathbf{x}) = \sin([1, -1]^T \mathbf{x}) + \exp(-\mathbf{x}^T A \mathbf{x})$$

$$f_6(\mathbf{x}) = \tanh(0.8x_1 + 0.2x_2) + \sin(0.3x_1 - 0.9x_2)$$

$$f_7(\mathbf{x}) = 0.5 \sin(x_1 + x_2) + 0.2x_1 - 0.2x_2$$

$$f_8(\mathbf{x}) = \exp(-(\mathbf{x} - \mathbf{w}_1)^T A (\mathbf{x} - \mathbf{w}_1)) + \exp(-(\mathbf{x} - \mathbf{w}_2)^T B (\mathbf{x} - \mathbf{w}_1))$$

$$f_9(\mathbf{x}) = f_8(\mathbf{x}) + 0.5 \sin(x_1 + 0.3x_2) + 0.5 \sin(0.2x_1 - 0.8x_2)$$

$$f_{10}(\mathbf{x}) = \sin(x_1 + x_2 + x_3) + \cos(x_1 + x_2 + x_3)$$

$$f_{11}(\mathbf{x}) = \tanh(x_1 + x_2 + x_3 + x_4)$$

---

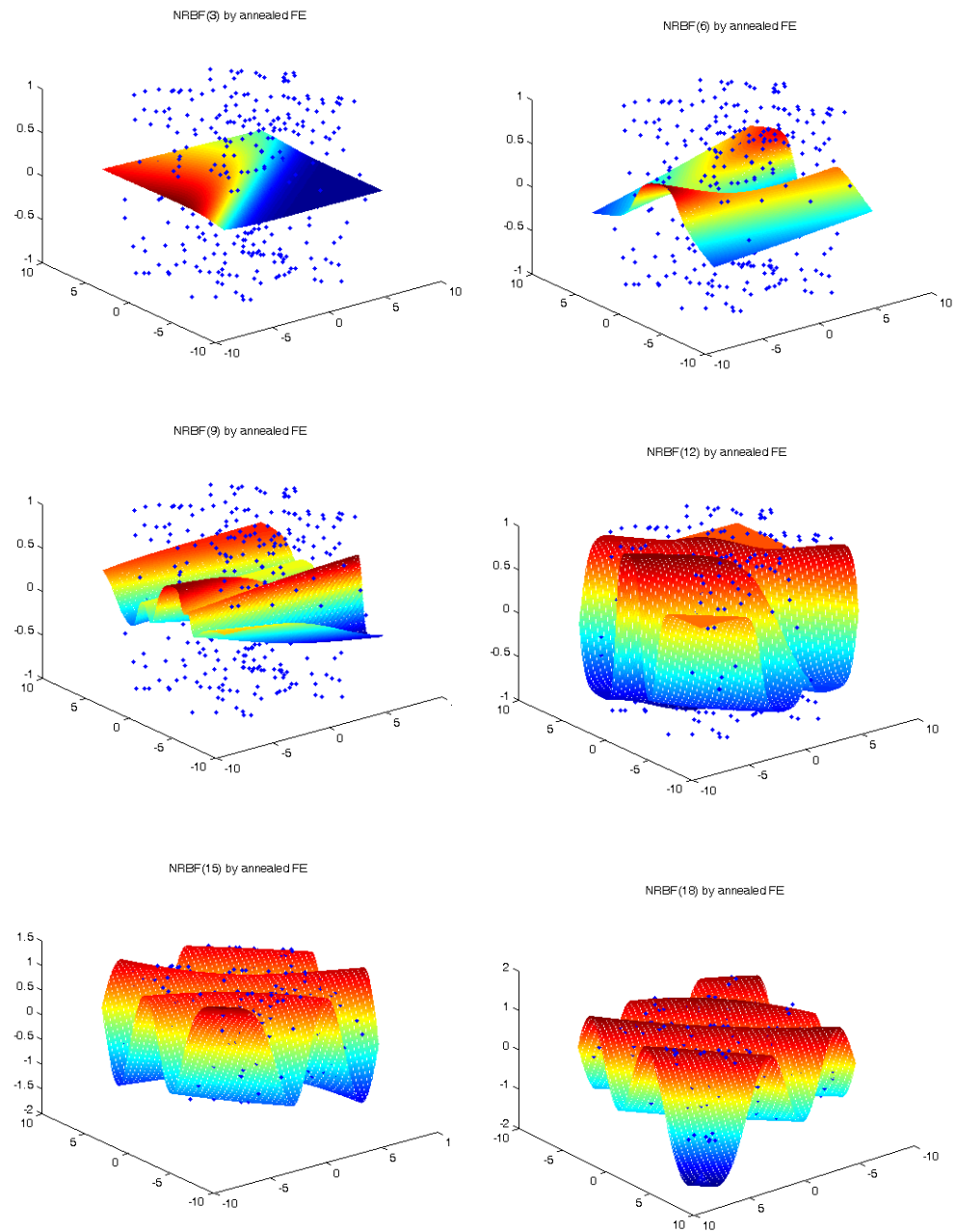
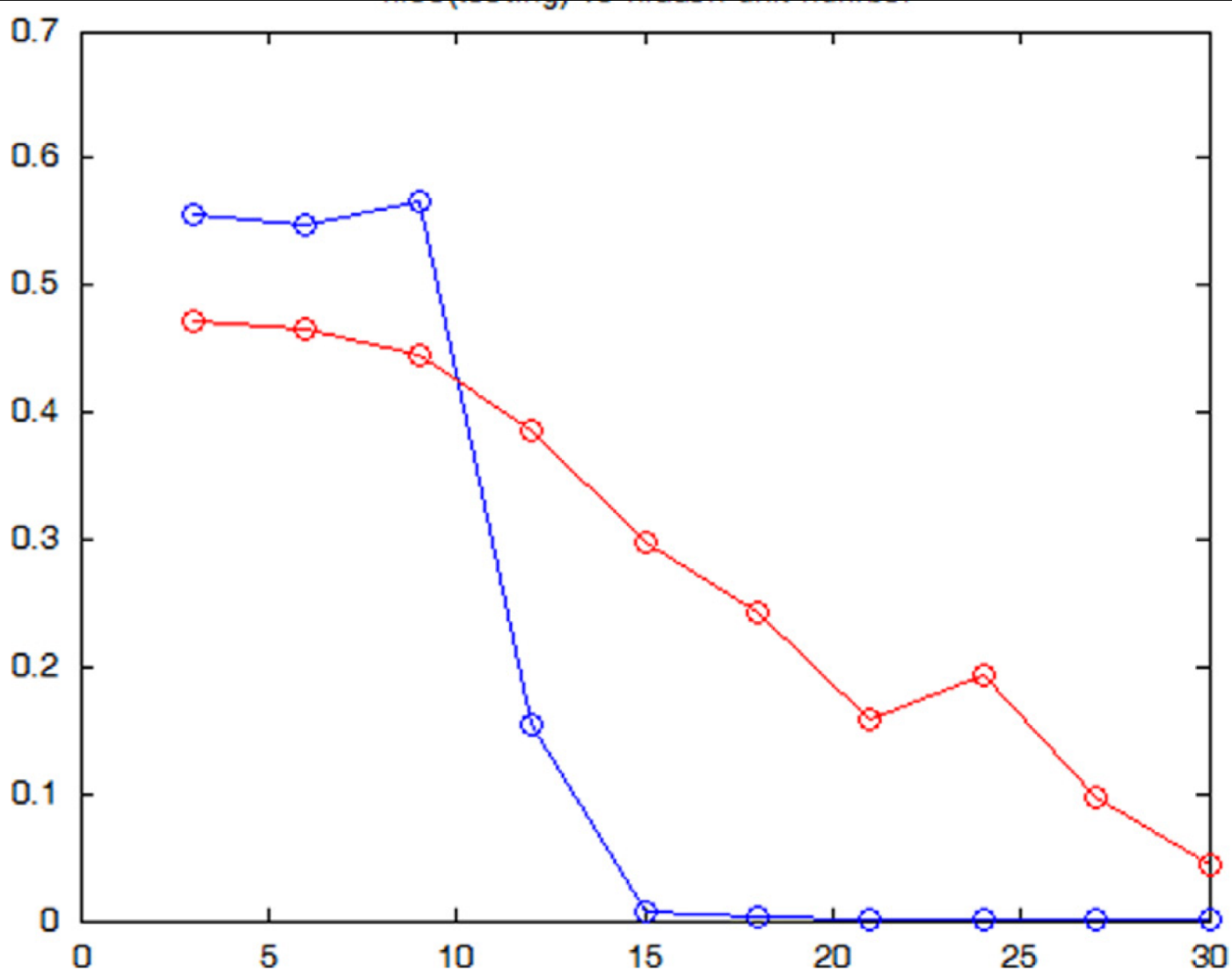


Figure 4



**Fig. 5.** Mean square testing errors of annealed competitive learning (blue curve) and the Rättsch 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.)

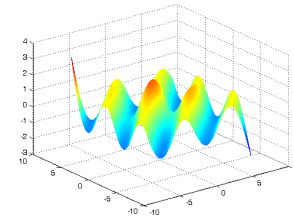
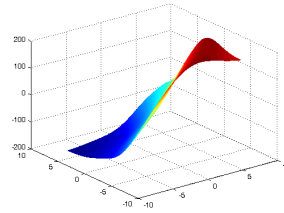
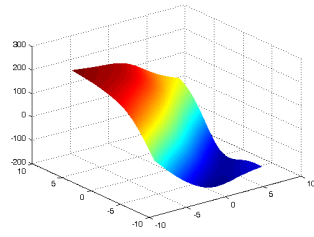
**Table 2**

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

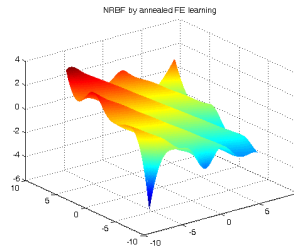
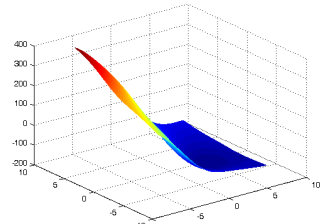
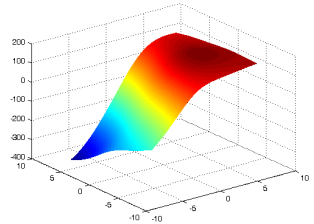
	Mean square error			
	Mahalanobis-NRBF ( $K=1, M=41$ )	Euclidean-RBF (Rätsch, $M=41$ )	Euclidean-RBF (LM, $M=41$ )	MLP (LM, $M=41$ )
<b>Training</b>				
$f_1$	$3.4e-5 \pm 0$	$1.4e-2 \pm 1.1e-5$	$1.8e-3 \pm 9.5e-7$	$9.6e-5 \pm 8.2e-9$
$f_2$	$1.2e-3 \pm 4.5e-7$	$4.1e-1 \pm 4.7e-3$	$1.4e0 \pm 2.5e-1$	$2.3e-2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

**Table 3**  
Quantitative performances of the relevant methods for approximating  $f_5$ - $f_8$ .

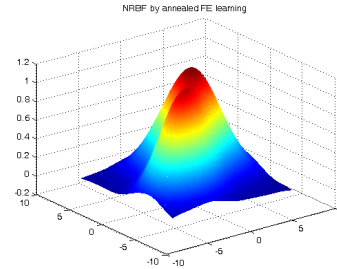
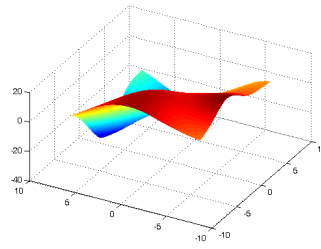
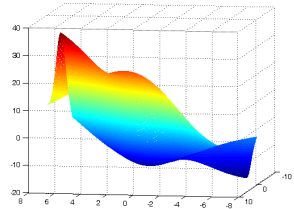
	Mean square error				
	Mahalanobis-NRBF ( $K=2, M=41$ )	Euclidean-RBF (Rätsch, $M=41$ )	Euclidean-RBF (LM, $M=41$ )	MLP (LM, $M=41$ )	MLP (BP, $M=41$ )
<b>Training</b>					
$f_5$	$1.0e-5 \pm 0$	$1.4e-2 \pm 1.6e-5$	$4.4e-3 \pm 3.9e-6$	$1.5e-3 \pm 4.1e-6$	$9.8e-2 \pm 1.2e-2$
$f_6$	$1.7e-5 \pm 2.2e-11$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



(a)

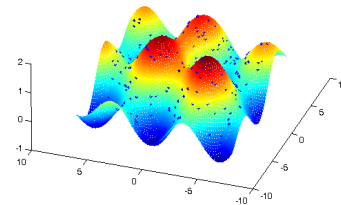


(b)



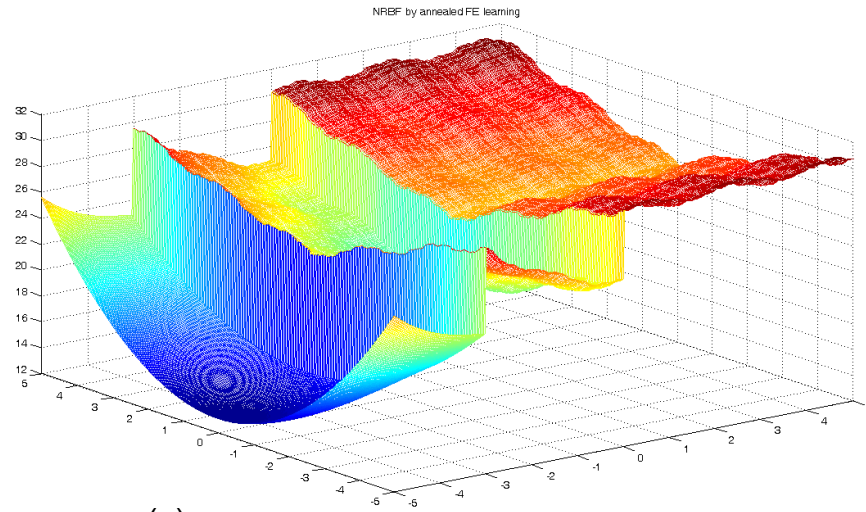
(c)

NREBF by annealed FE learning

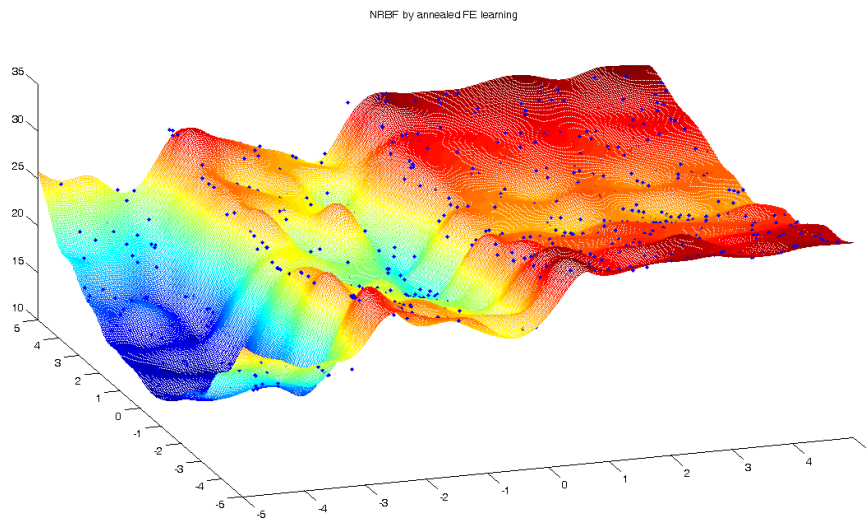


(d)

Figure 6



(a)



(b)

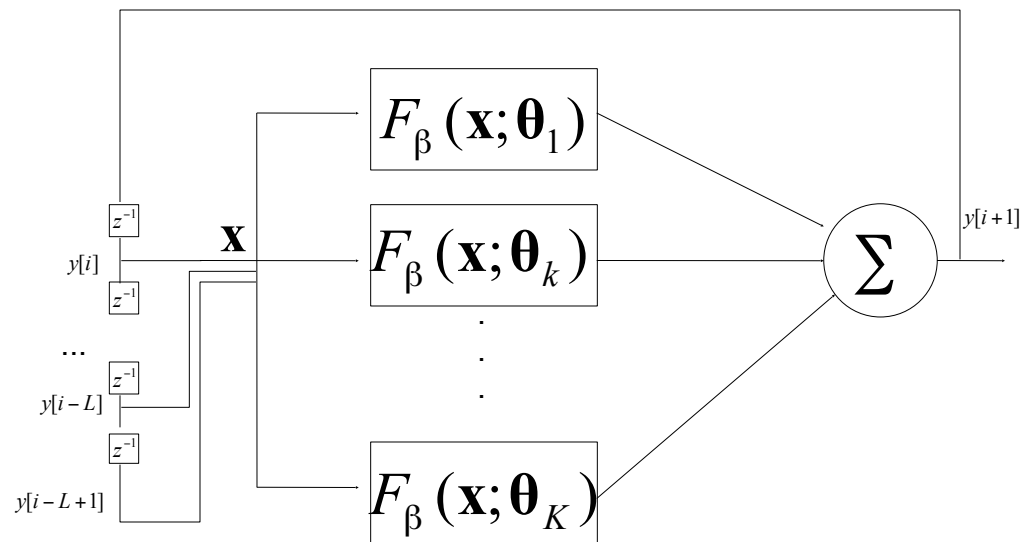
Figure 7



# Chaotic differential function approximation

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

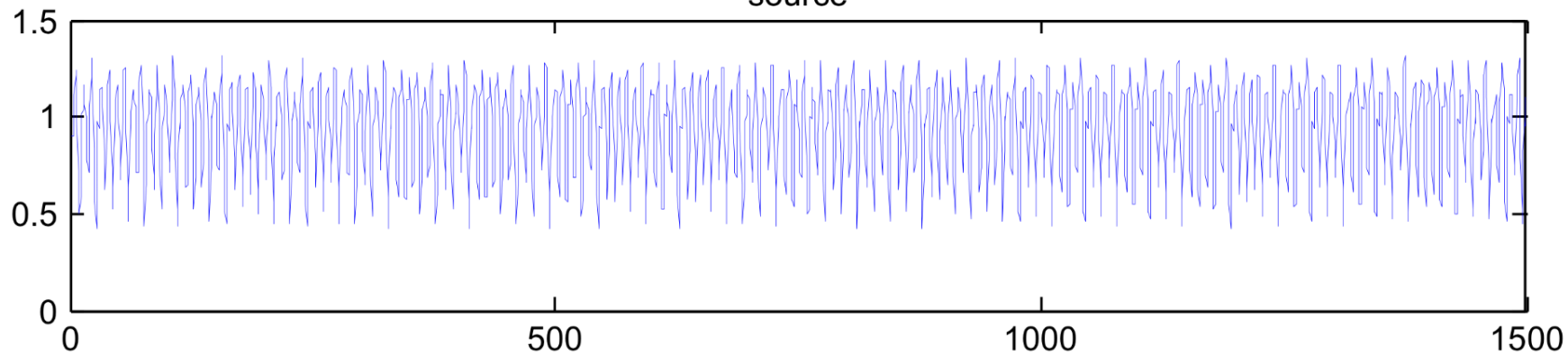
$\tau = 17, a = 0.2, c = 10$  and  $b = 0.1$



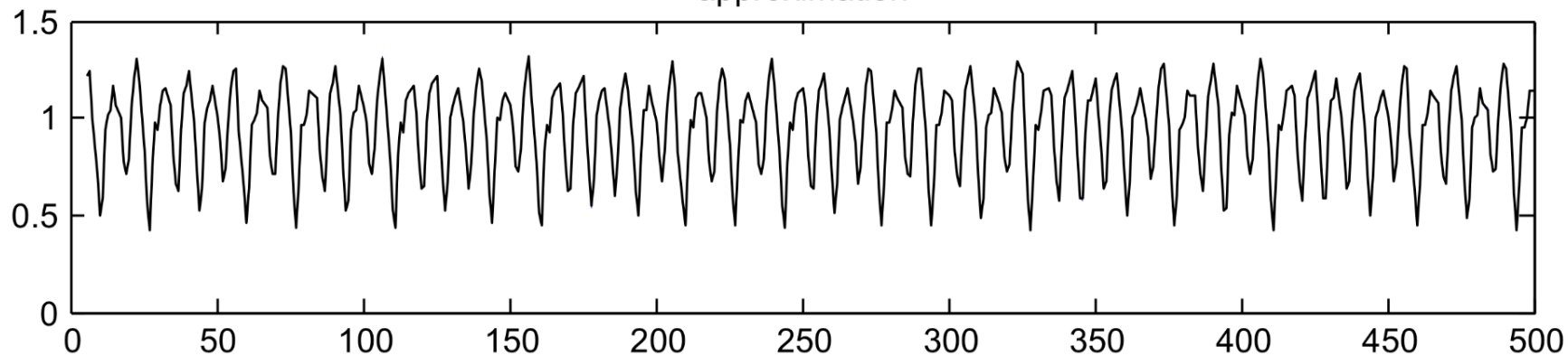
$$\mathbf{o}_t = f(\mathbf{x}_t = (\mathbf{o}_{t-L}, \mathbf{o}_{t-L+1}, \dots, \mathbf{o}_{t-1})^T),$$

Figure 9

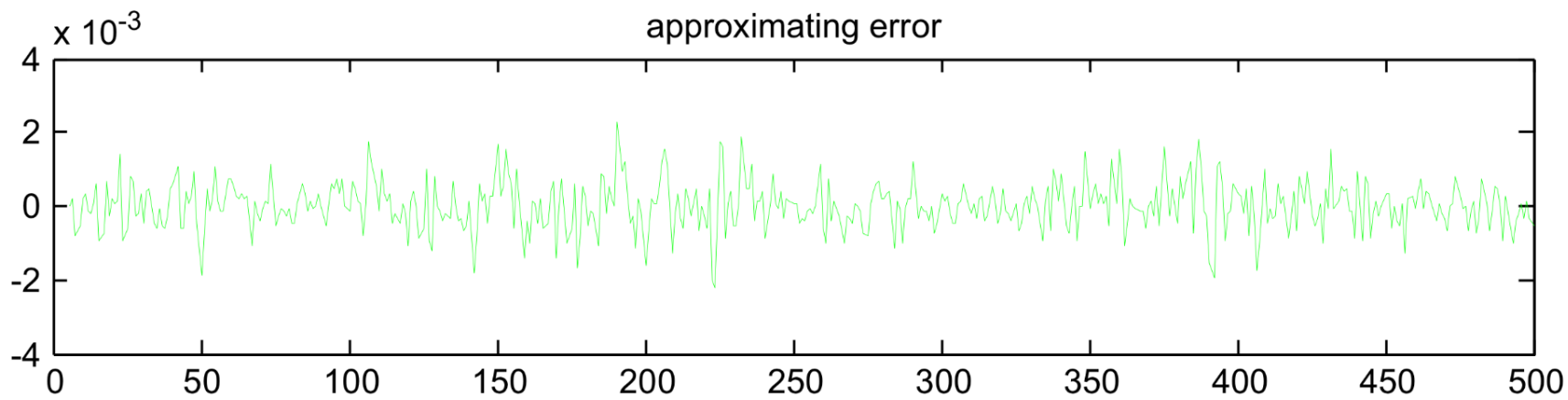
source



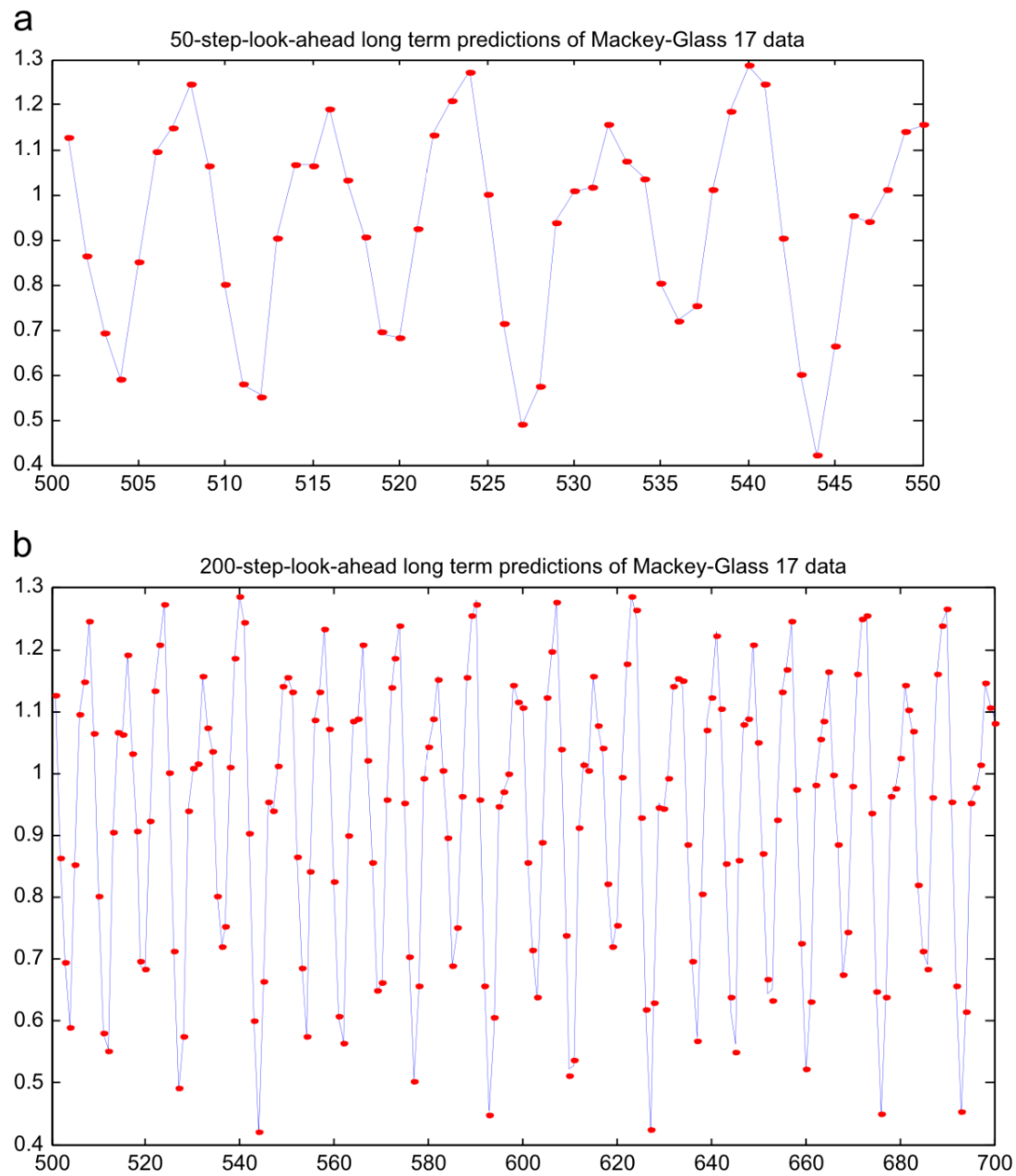
approximation



approximating error

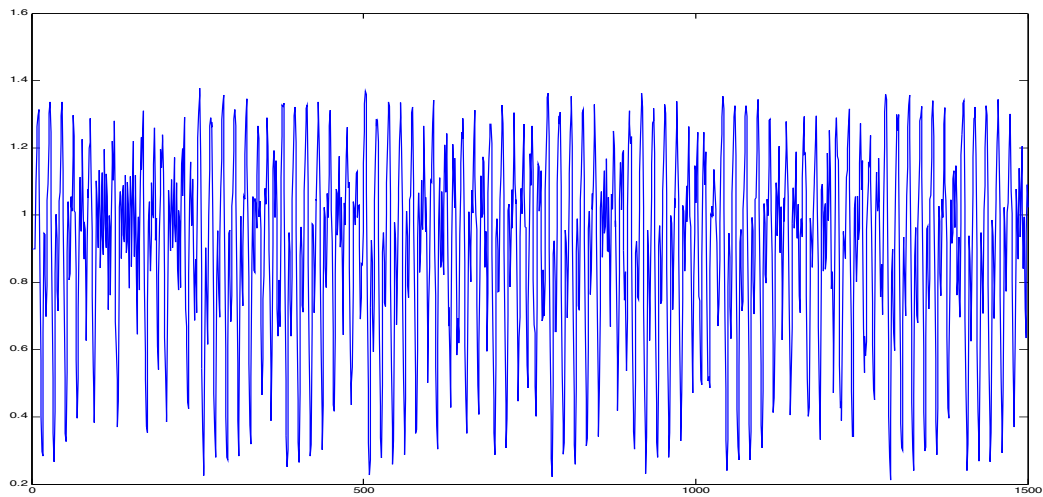


Approach	$mse_{S_1}$	$mse_{S_2}$	$D_{O_2}^{50}$	
	mean $\pm$ var	mean $\pm$ var	mean $\pm$ var	min
Mahalanobis-NRBF modules ( $K=3$ )	$4.68e-7 \pm 0$	$2.22e-6 \pm 0$	$5.00e-3 \pm 1.77e-5$	$2.18e-3$
MLP-LM (8)	$5.22e-5 \pm 3.51e-10$	$6.20e-5 \pm 6.77e-10$	$4.08e-2 \pm 2.92e-5$	$2.95e-2$
MLP-LM (15)	$4.61e-5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



**Fig. 10.**  $n_2$ -step-look-ahead predictions of MG17 time series with  $n_2 = 50$  and  $n_2 = 200$  by annealed cooperative-competitive learning with  $K=2$ . (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Mackey-Glass 30 data



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

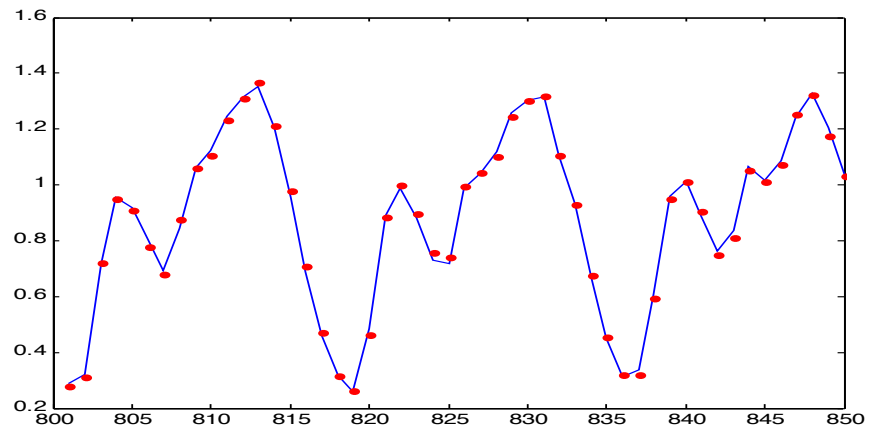
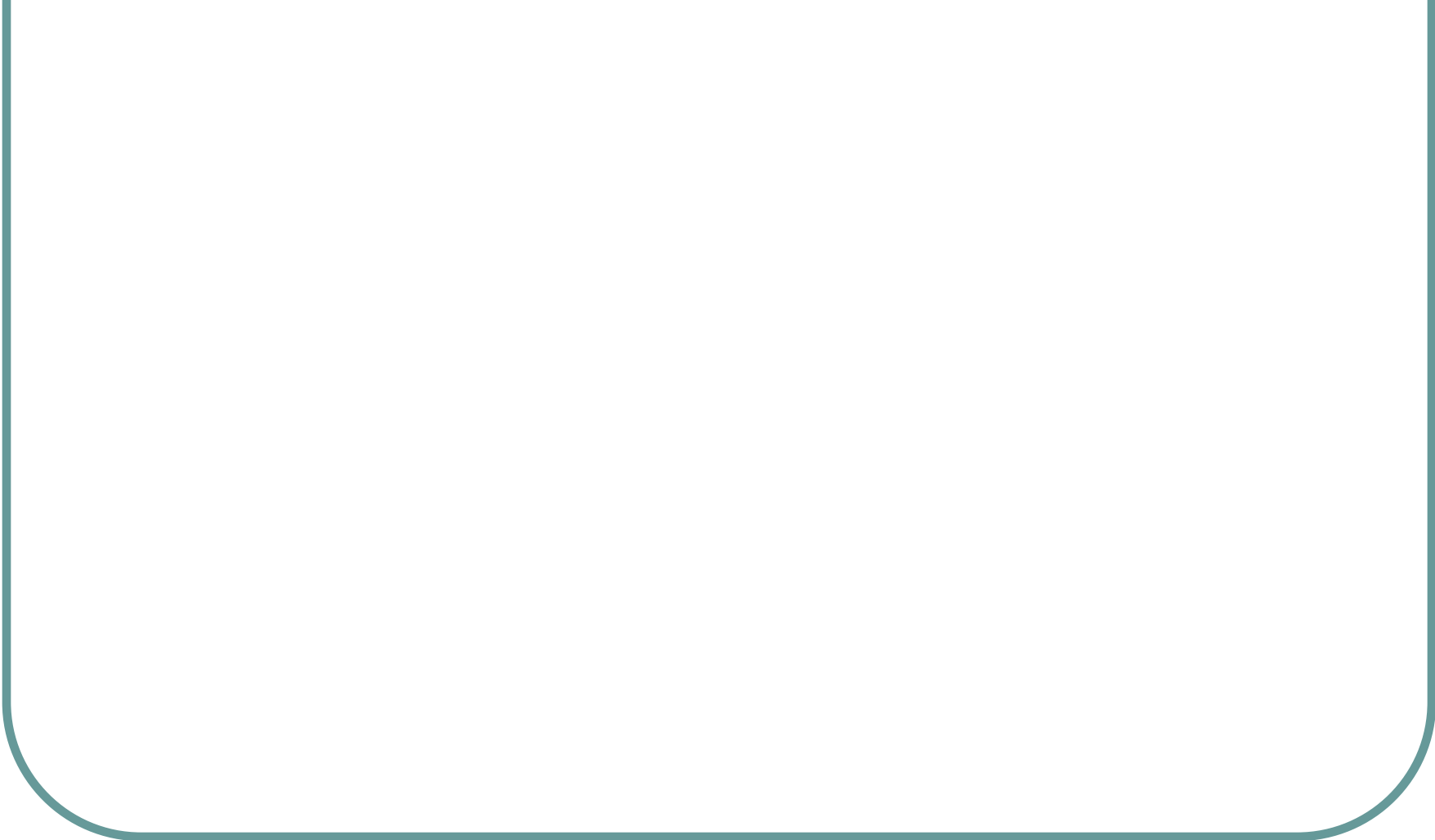


Figure 11





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

where the delay  $\tau$  is set to 1.6.

# Conclusions and discussions