Mathematical computation for 3D face reconstruction

Matlab Software

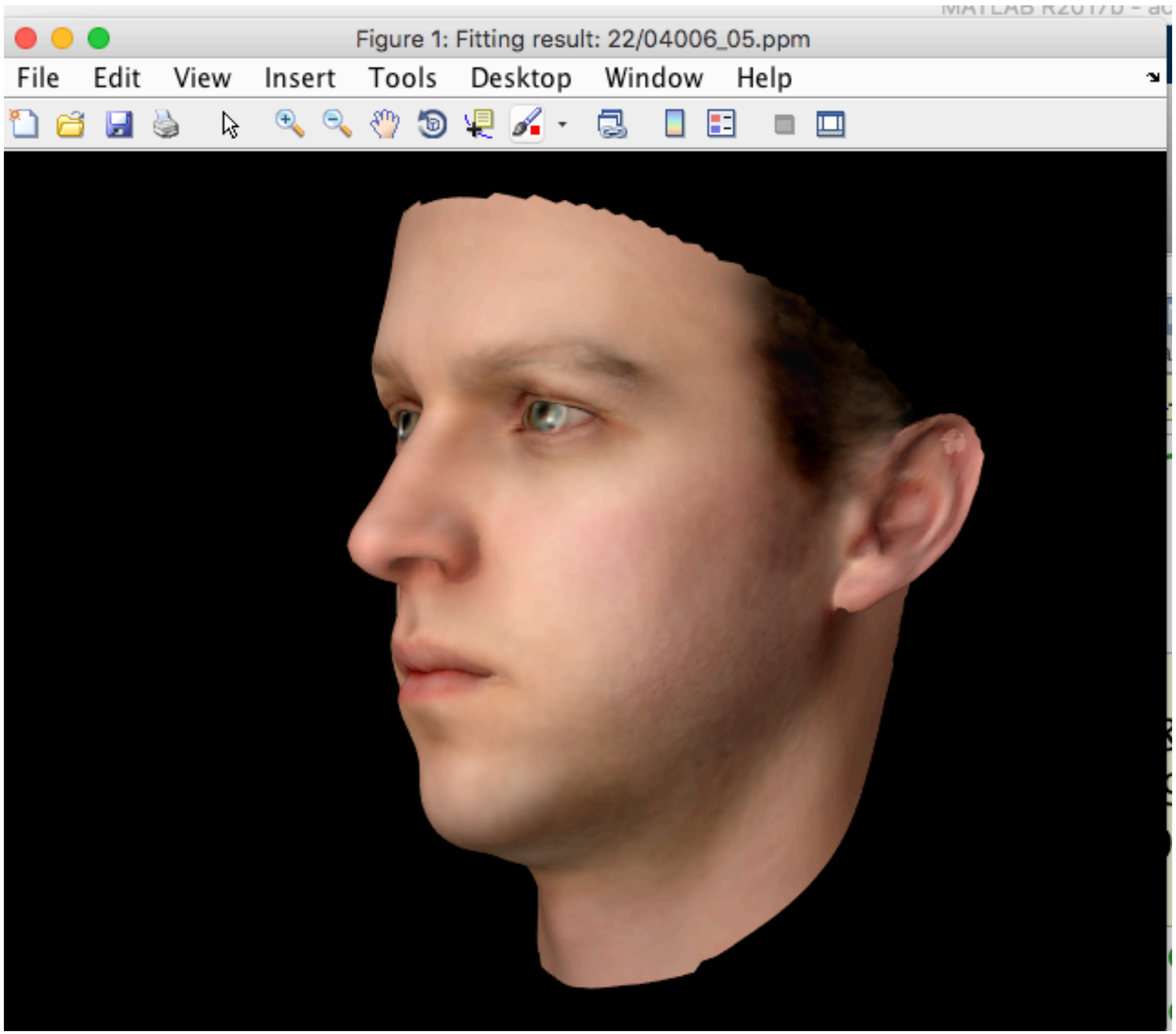


Figure 1: Fitting result: 22/04006_05.ppm
File Edit View Insert Tools Desktop Window Help

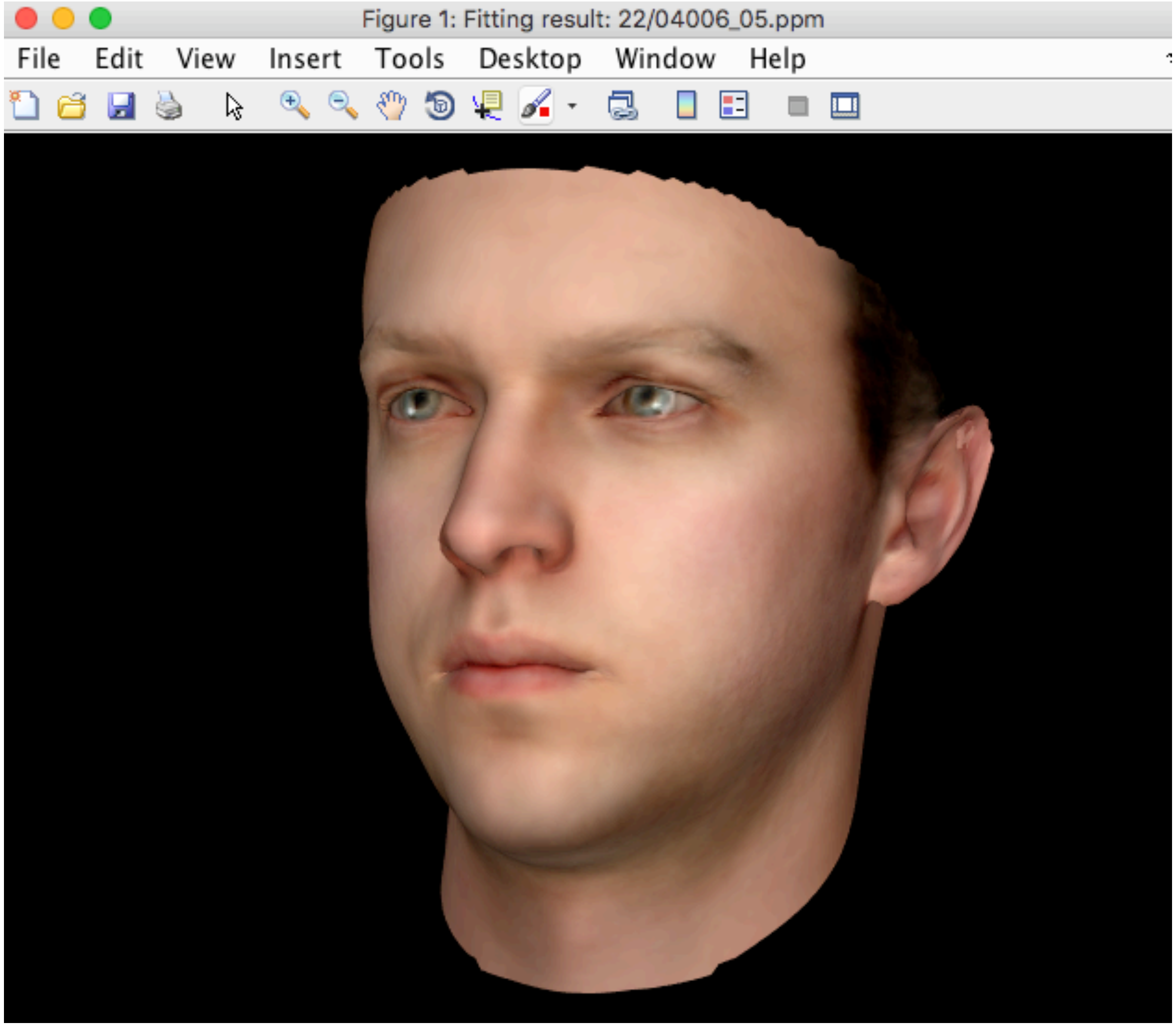


Figure 1: Fitting result: 22/04006_05.ppm
File Edit View Insert Tools Desktop Window Help

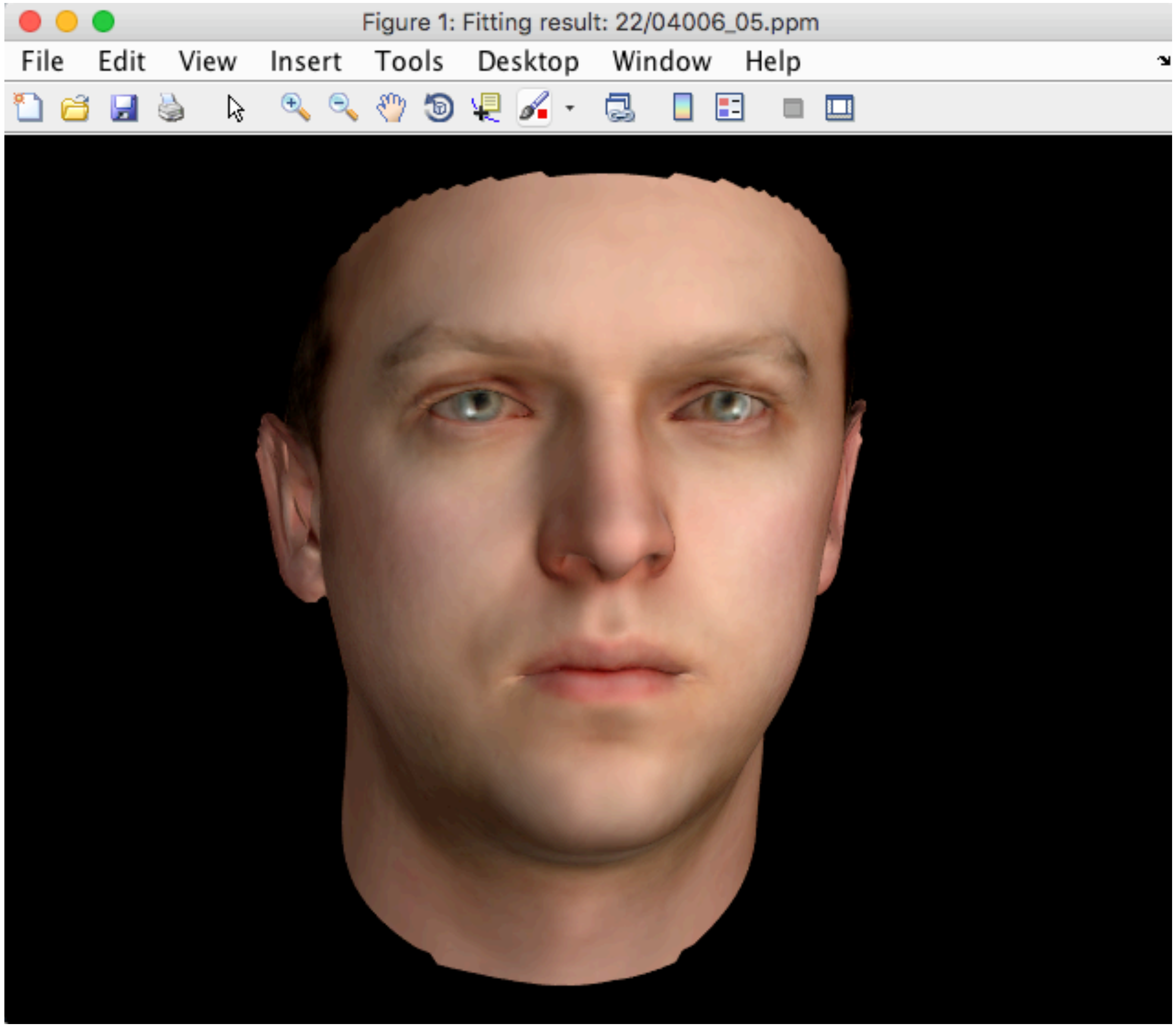


Figure 1: Fitting result: 22/04006_05.ppm
File Edit View Insert Tools Desktop Window Help

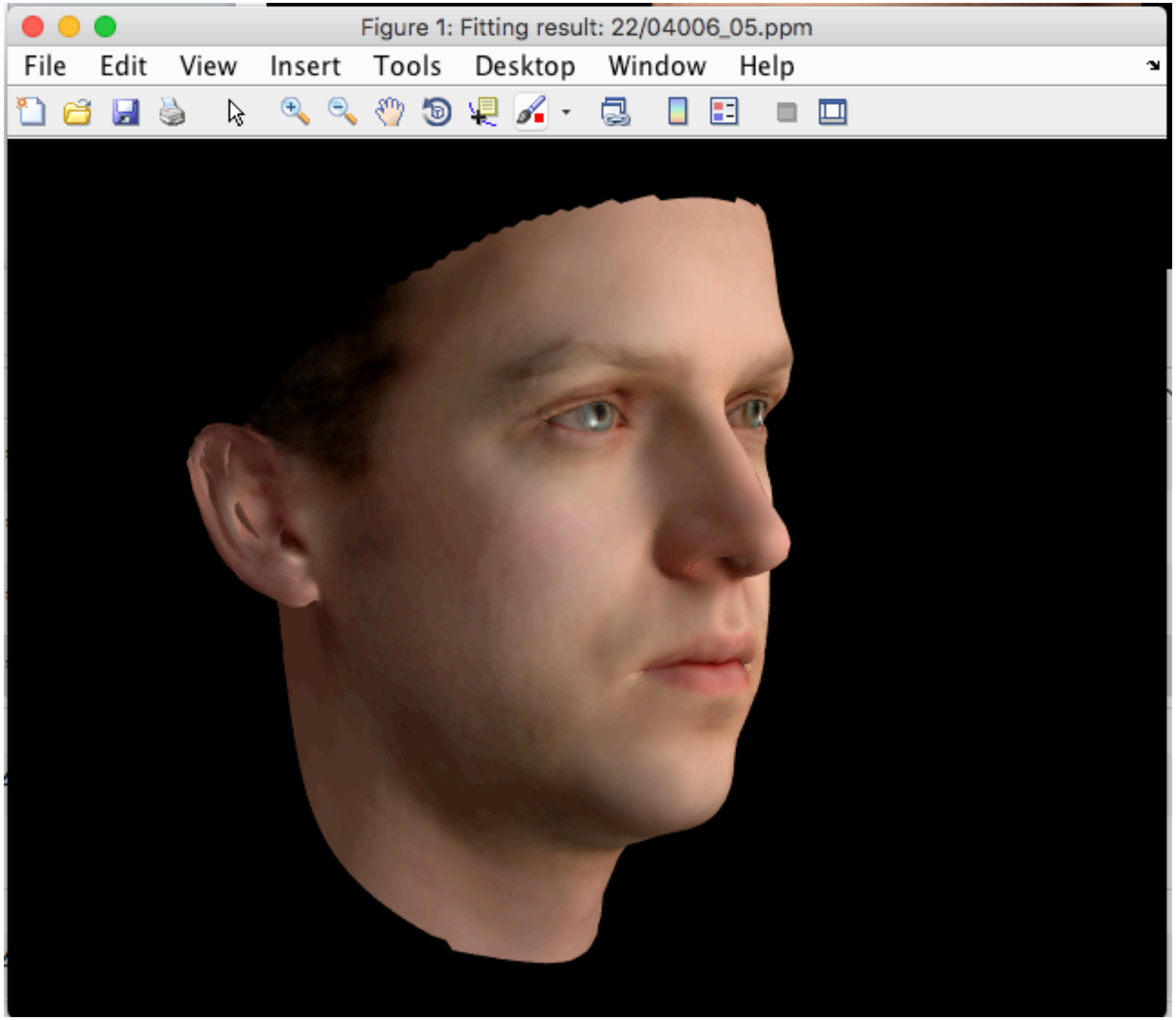


Figure 1: Fitting result: 22/04006_05.ppm
File Edit View Insert Tools Desktop Window Help

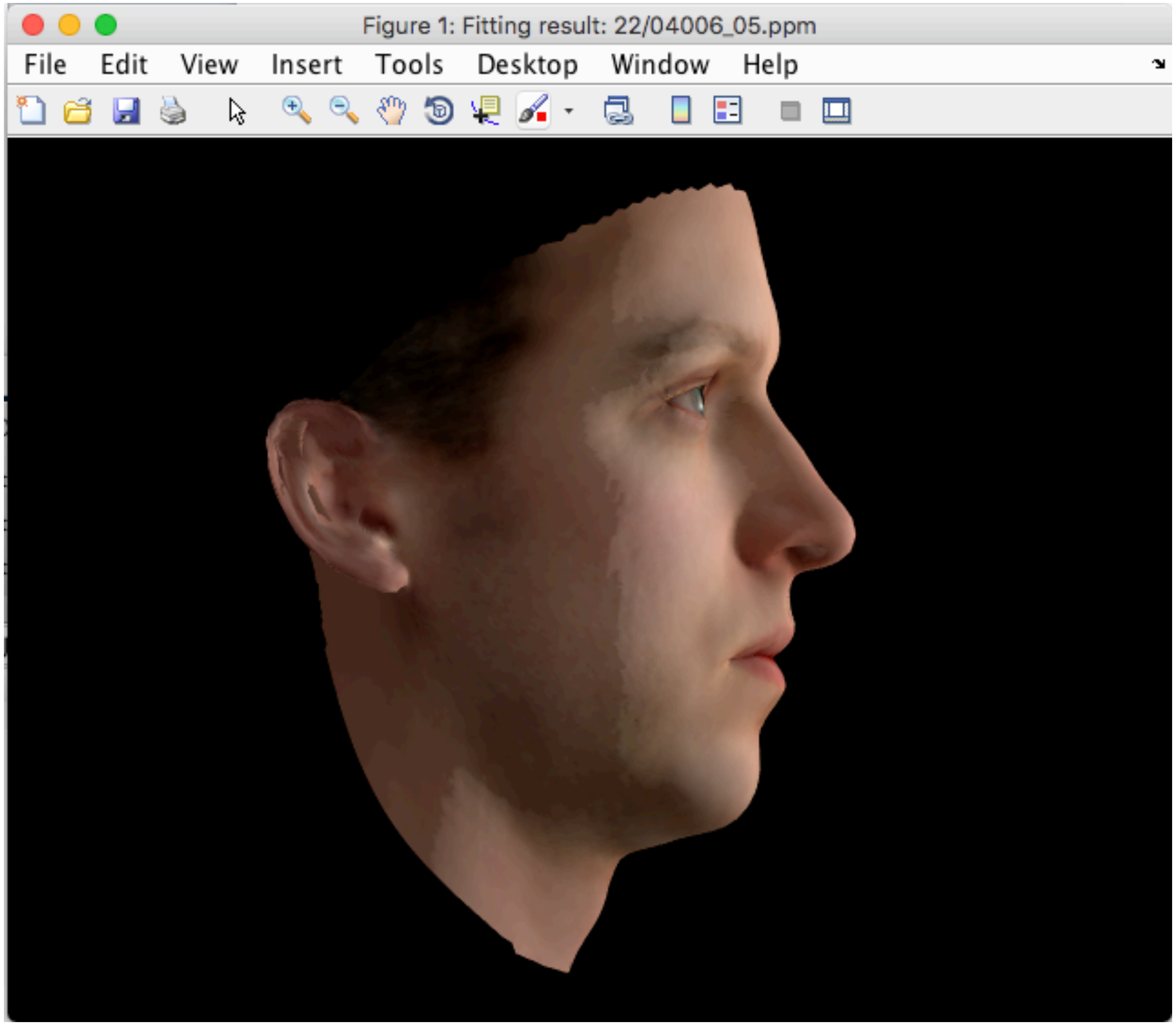
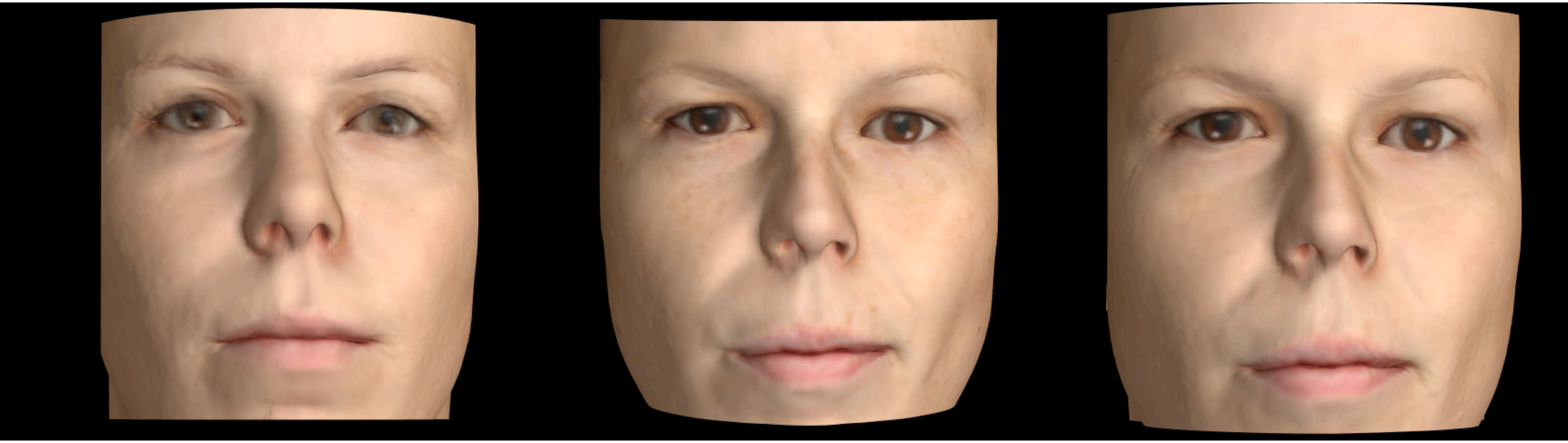


Figure 1: Fitting result: 22/04006_05.ppm
File Edit View Insert Tools Desktop Window Help



Handwriting 99 Multiplication

Handwriting Mult...

打開

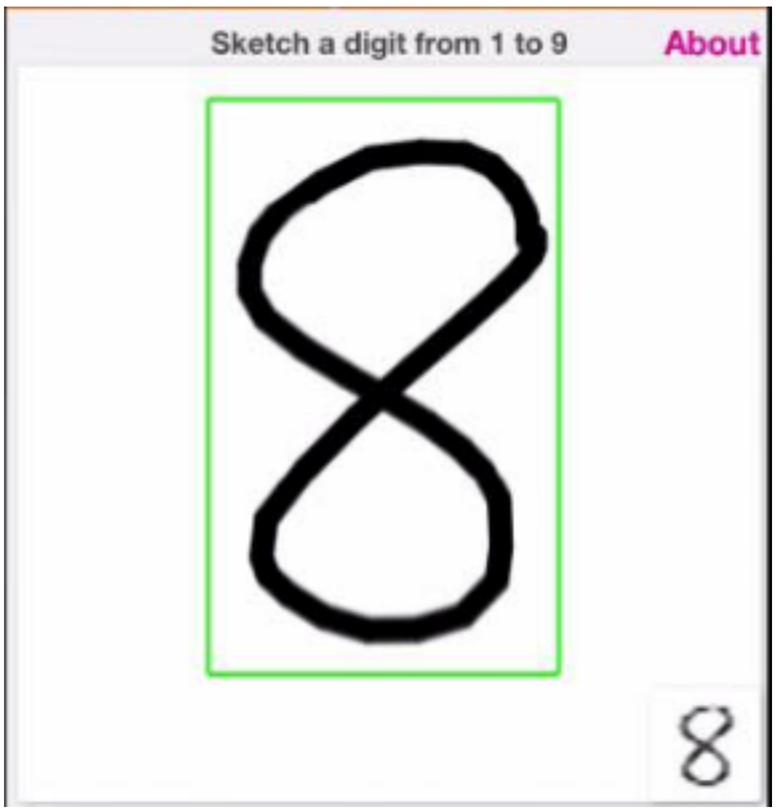
4.7 ★★★★★

10份評分

4+

年齡

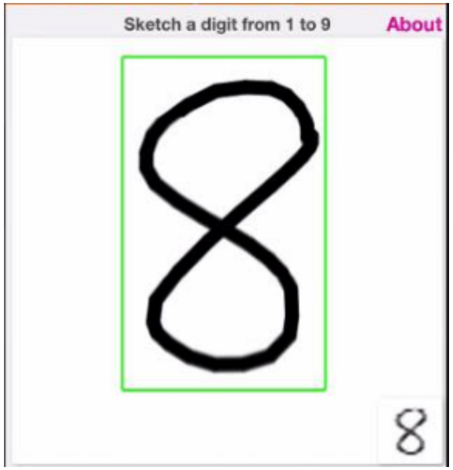
domain: a set of 280,000 hw-digit images



0000000100

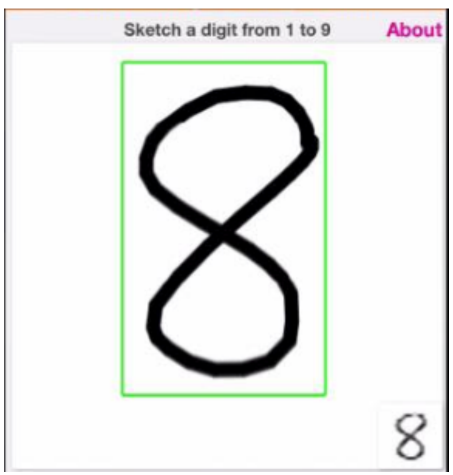
codomain

- {0,1,2,3,4,5,6,7,8,9}

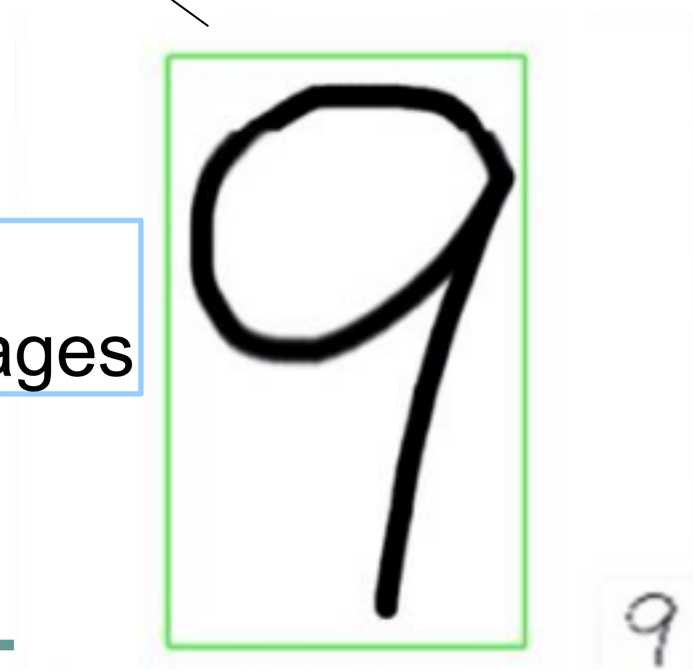


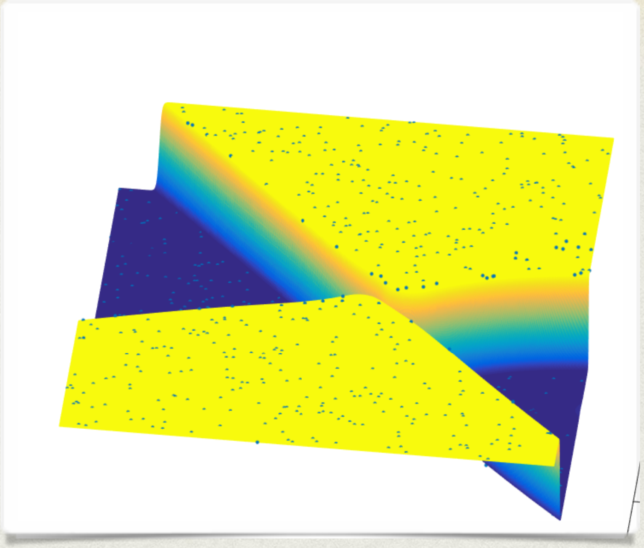
codomain

- {0,1,2,3,4,5,6,7,8,9}



280,000
hw-digit images

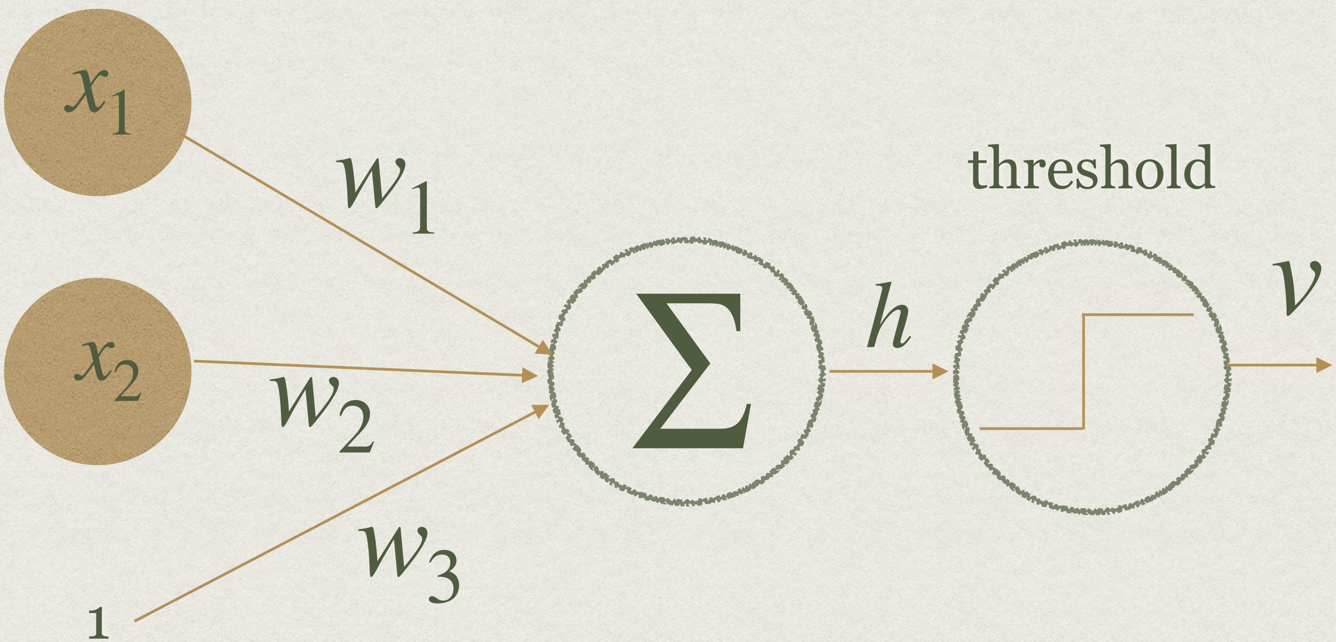




AI DEEP LEARNING AND APPROXIMATION

DISCRETE NEURON

$d = 2$ dimension



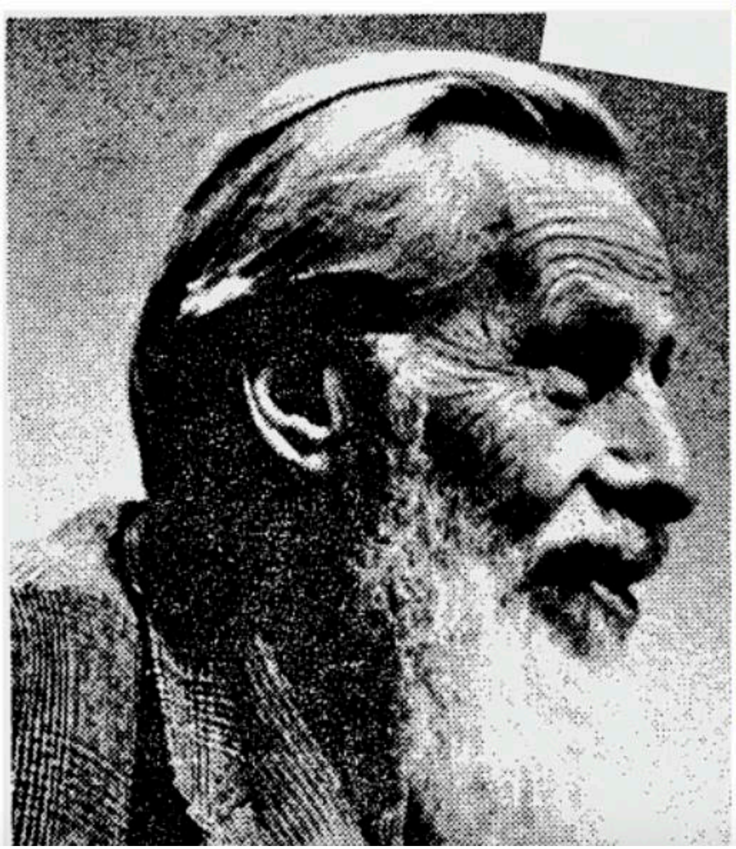
First artificial neurons: The McCulloch-Pitts model

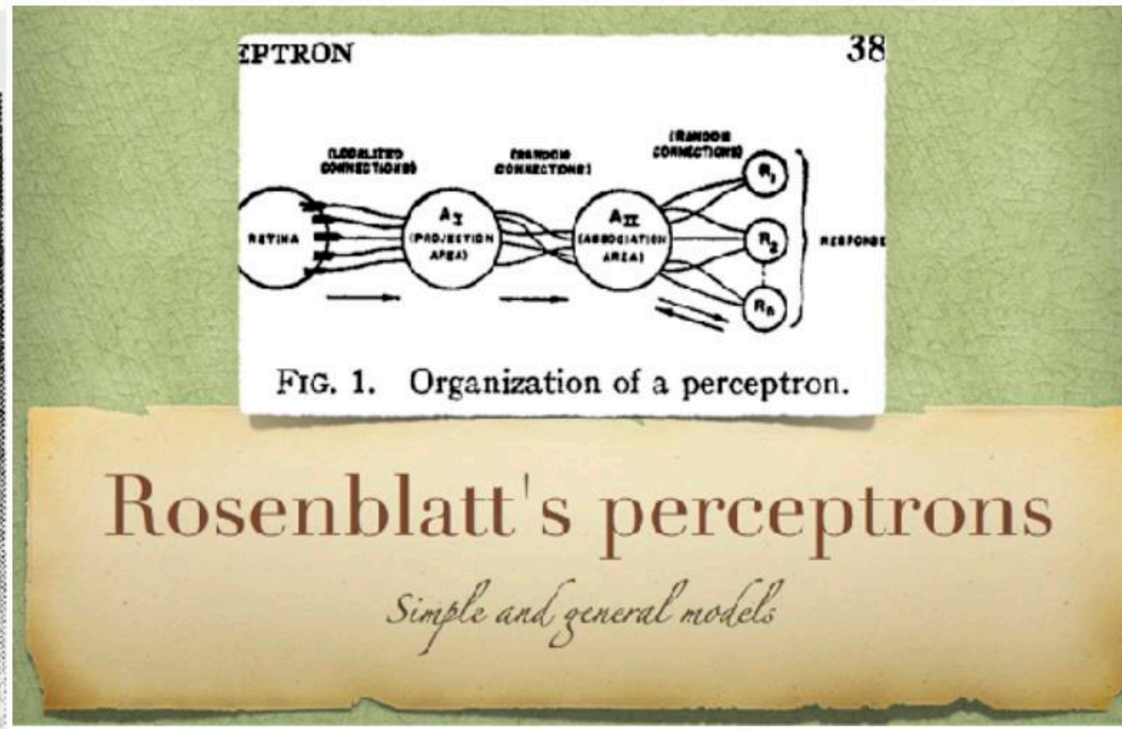
1898 - 1969

ASC FORUM Volume VI, Number 2 -Summer 1974

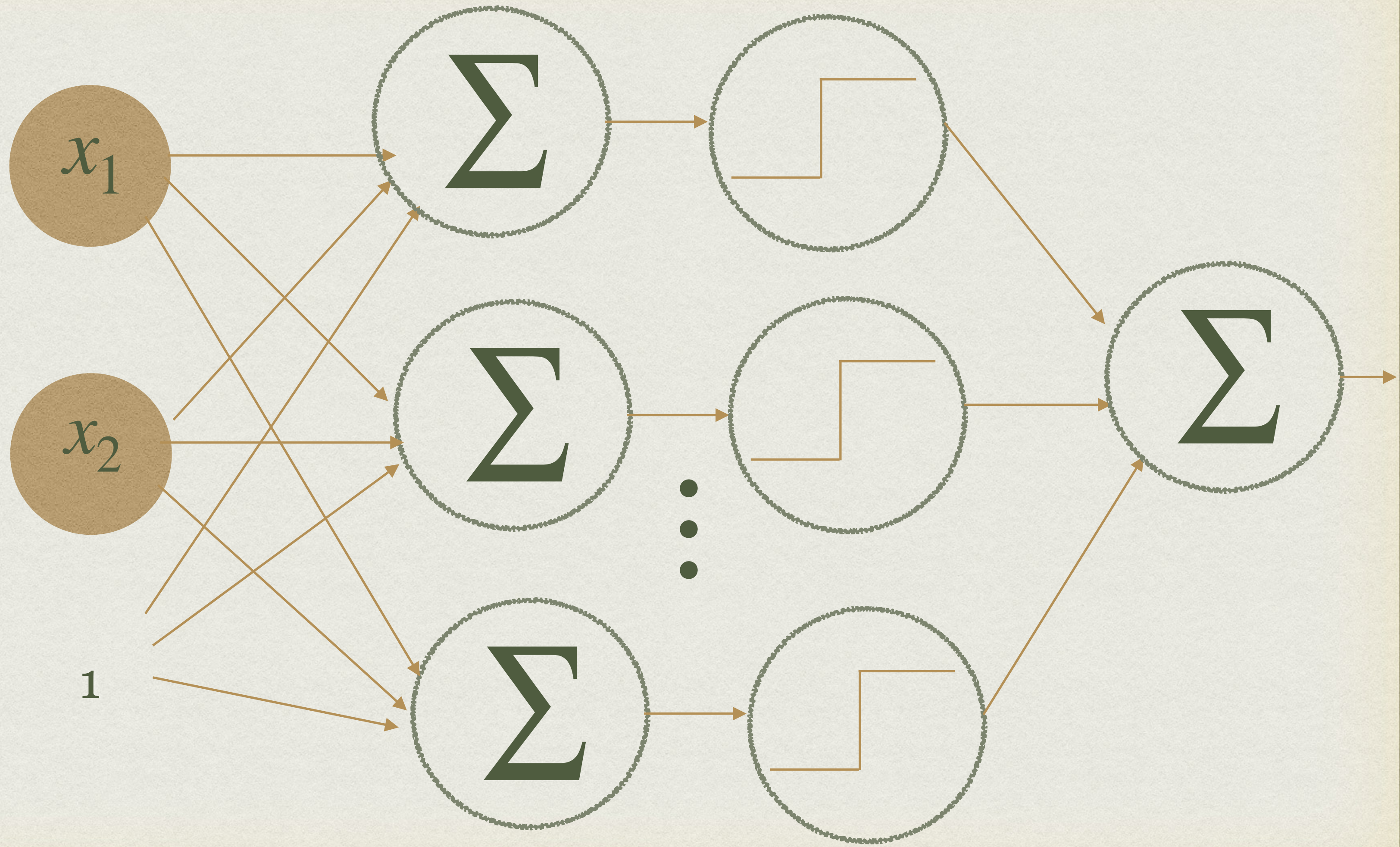
WARREN STURGIS MCCULLOCH

麥卡洛克-皮茨模型





A DISCRETE NEURAL NETWORK

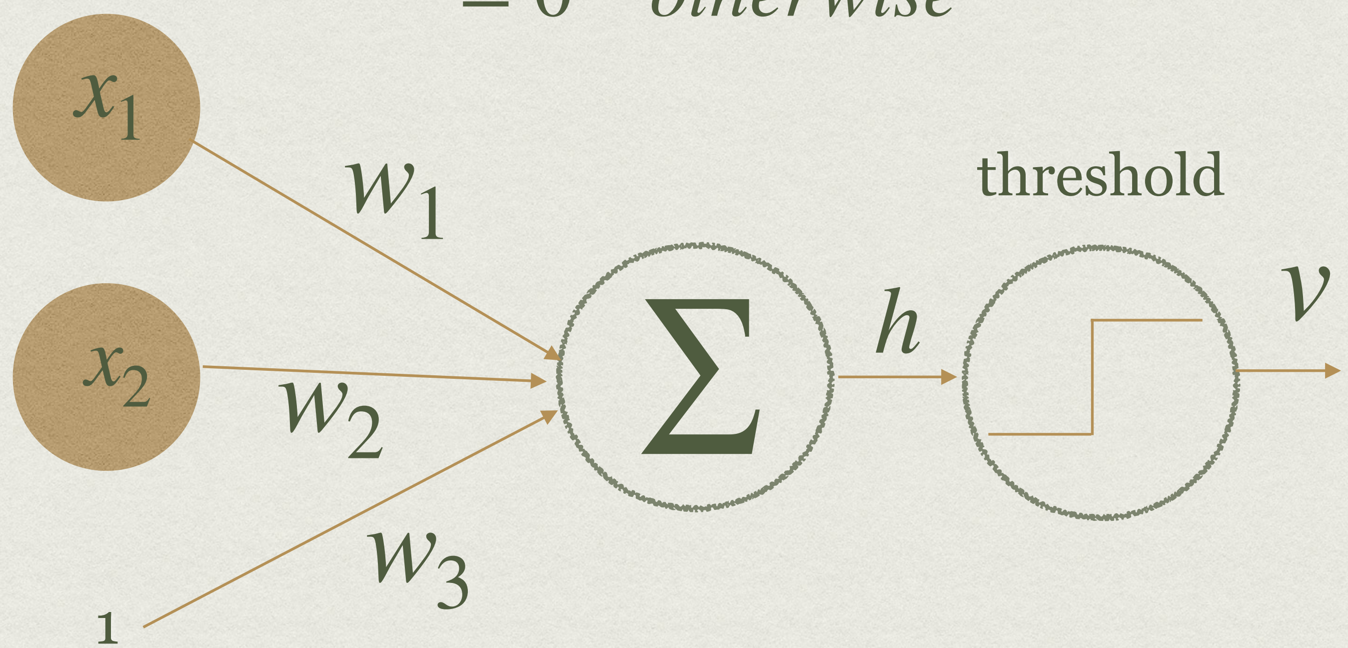


BINARY NEURON

$$h = w_1 x_1 + w_2 x_2 + w_3$$

$$v = 1 \quad if \quad h \geq 0$$

$$= 0 \quad otherwise$$



APPROXIMATION

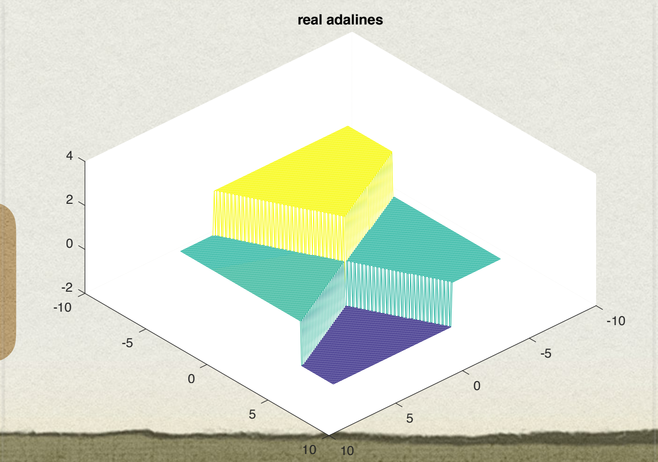
$$y = sign(2x_1 + x_2 - 1) - sign(x_1 - 2x_2 + 1)$$

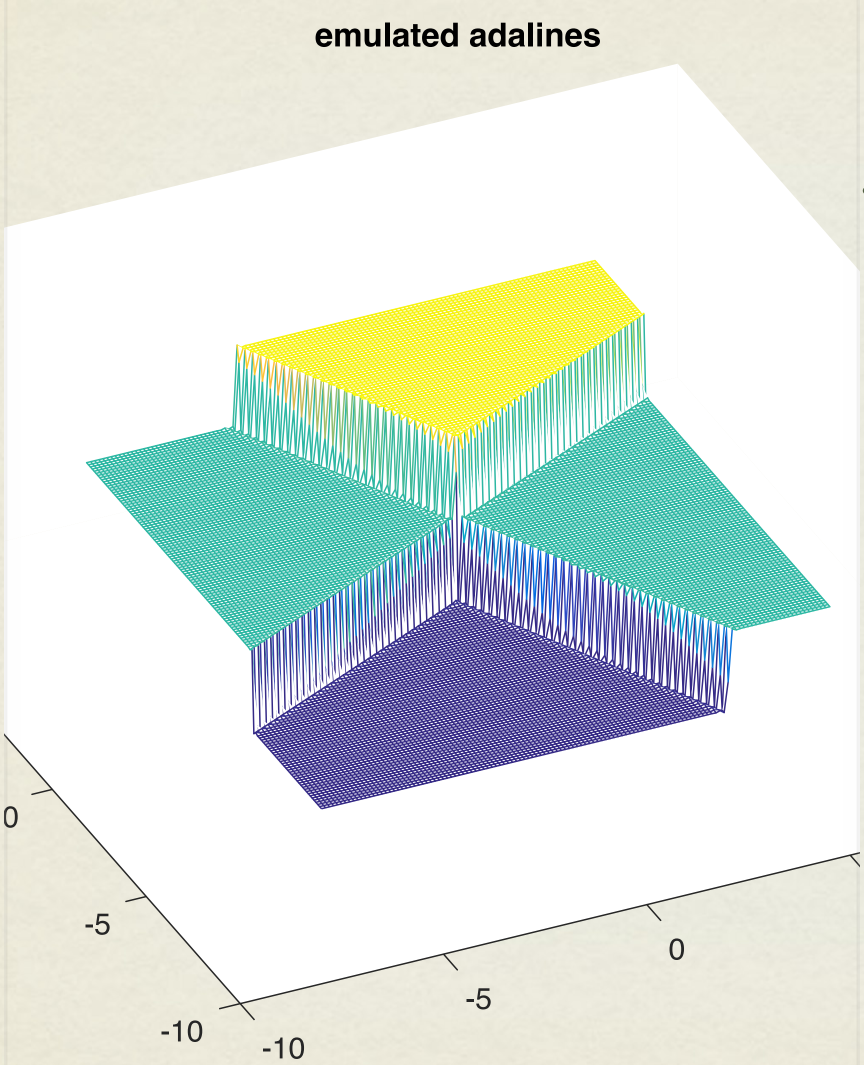
input
output

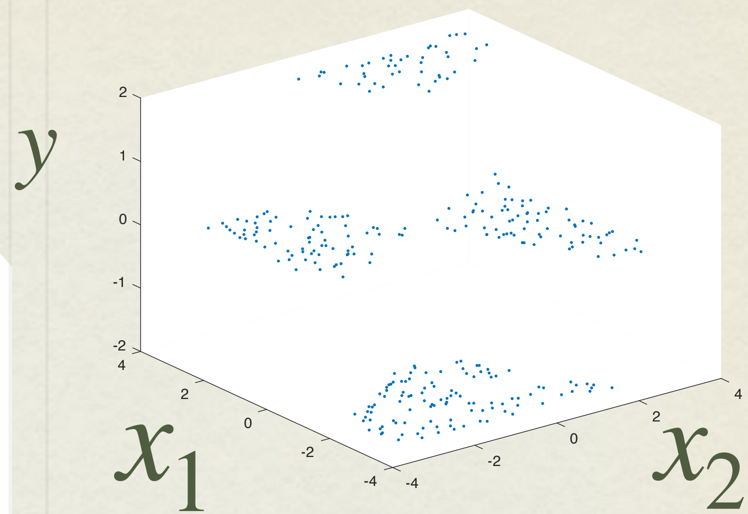
Learning a discrete neural network

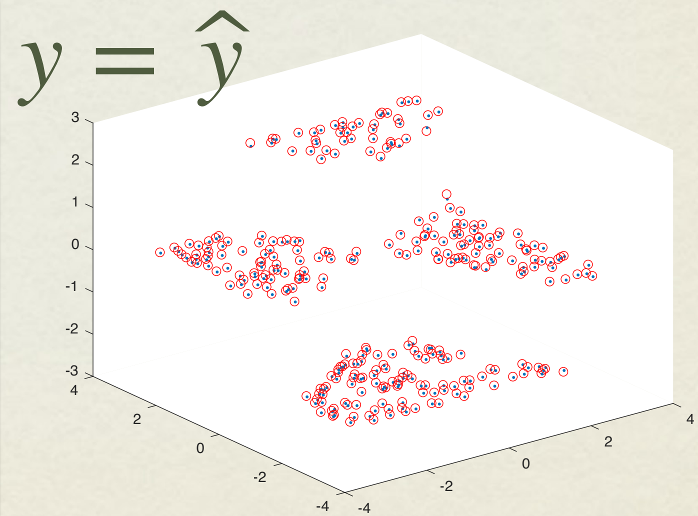
optimal interconnections

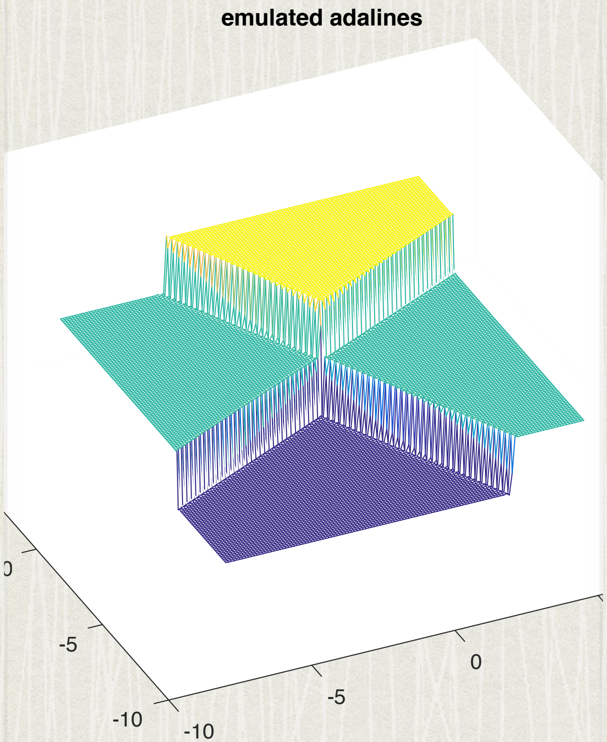
An approximating network

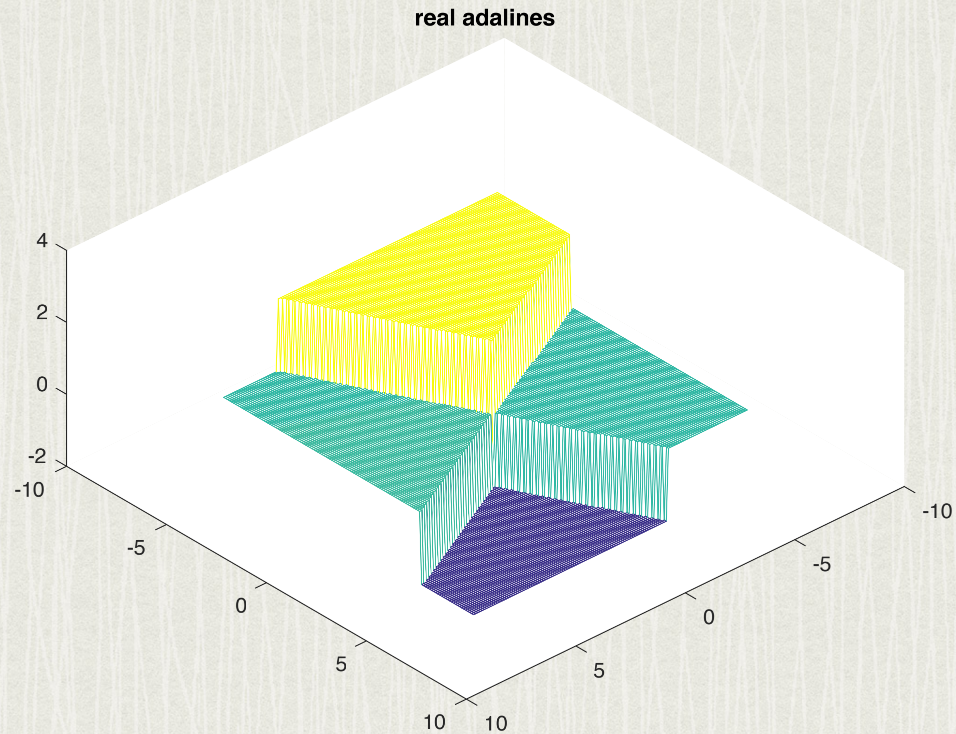



emulated adalines


y
x_1
x_2


$y = \hat{y}$





Adaptive Linear Element

ADALINE of Widrow

APPROXIMATION

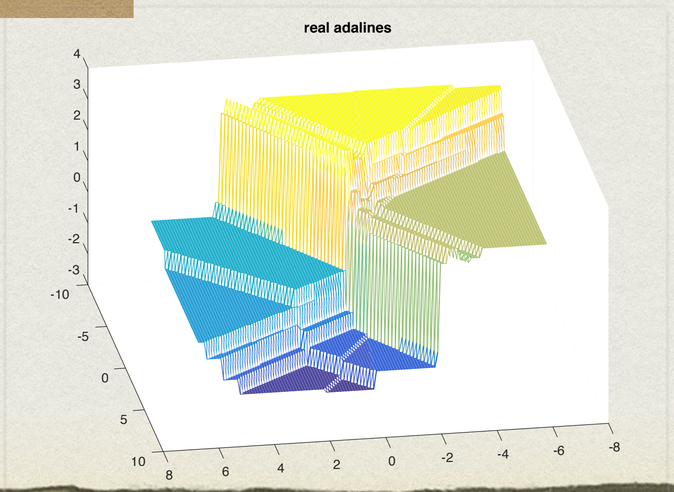
$$y = tanh(2x_1 + x_2 - 1) - tanh(x_1 - 2x_2 + 1)$$

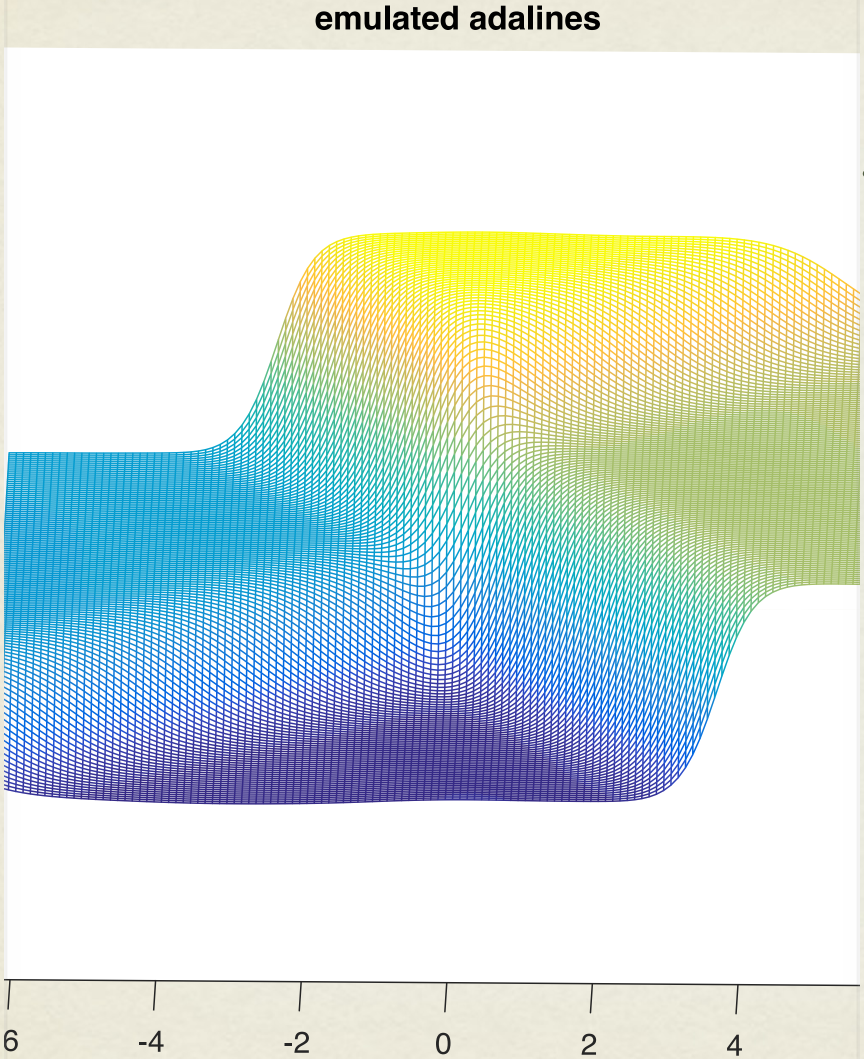
input
output

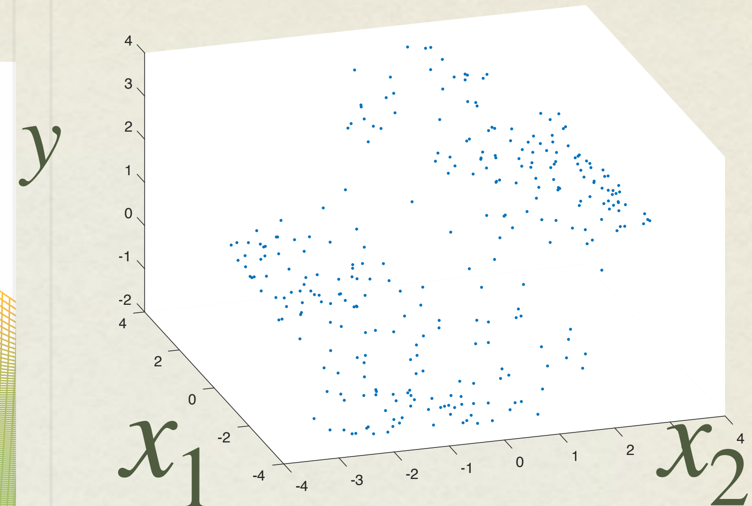
Learning a discrete neural network

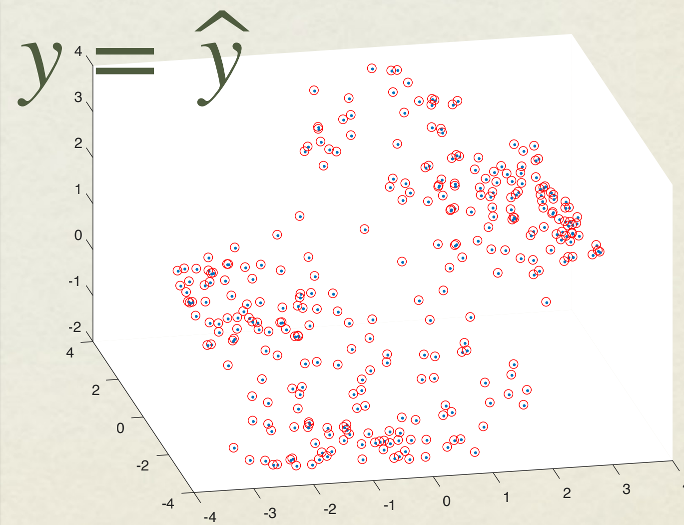
optimal interconnections

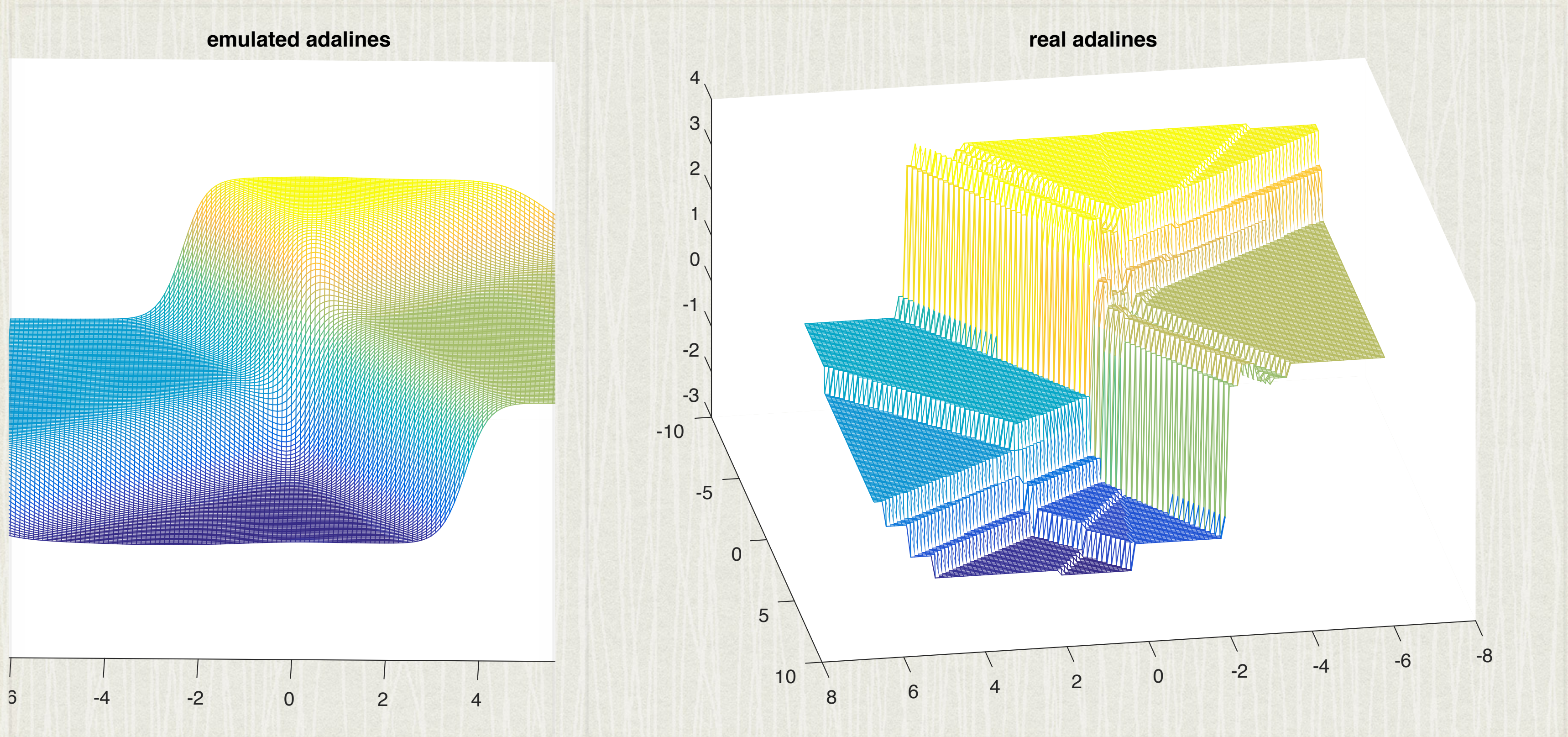
An approximating network




emulated adalines
-6
-4
-2
0
2
4

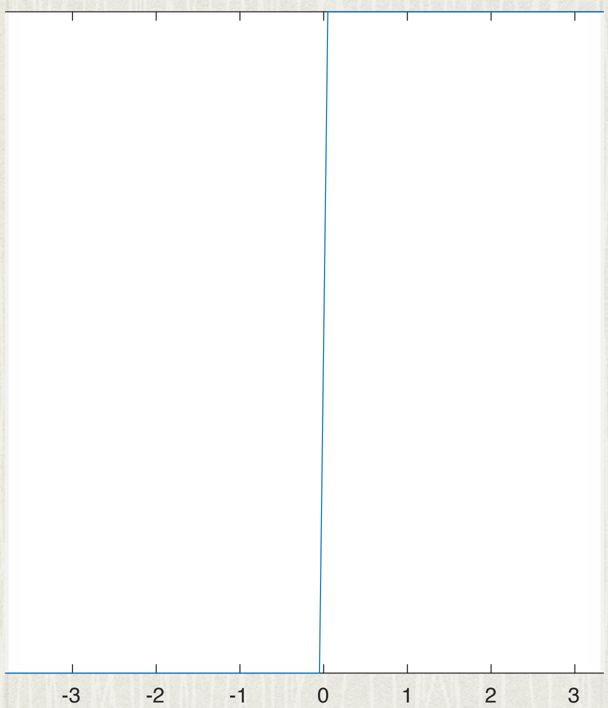

y
x_1
x_2

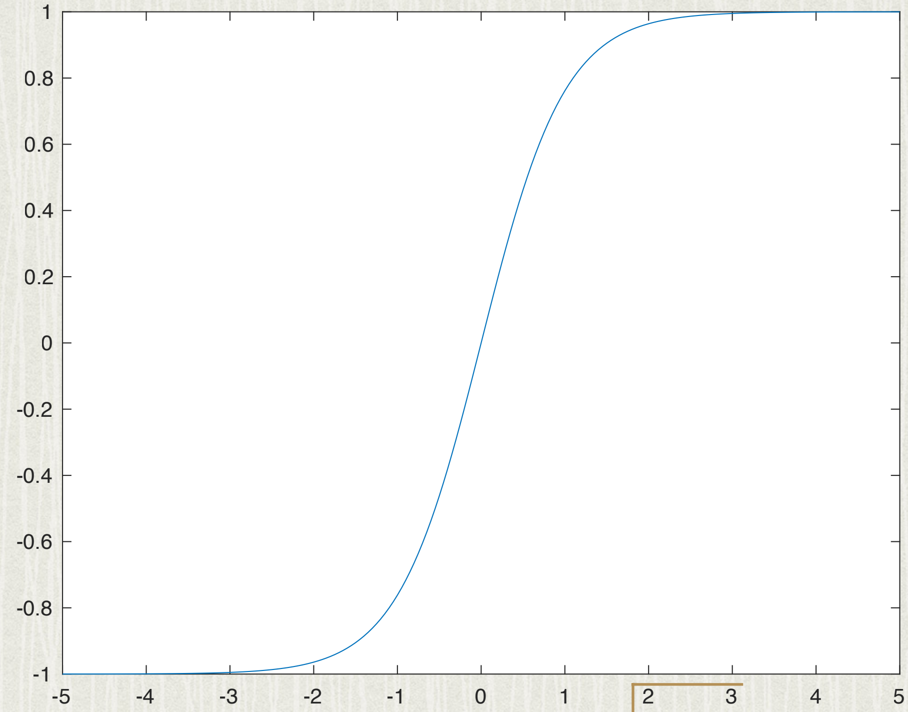

$y = \hat{y}$



Adaptive Linear Element

ADALINE of Widrow



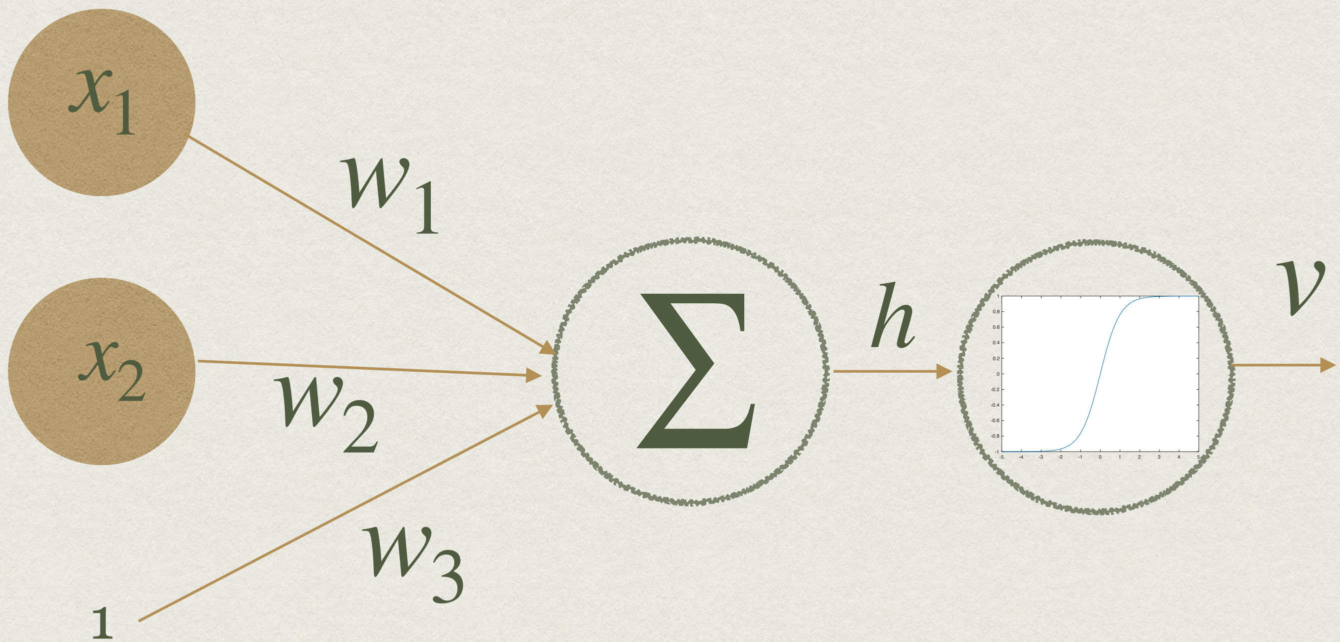


HYPER TANGENT

Differentiable sigmoid function

CONTINUOUS NEURON

$d = 2$ dimension

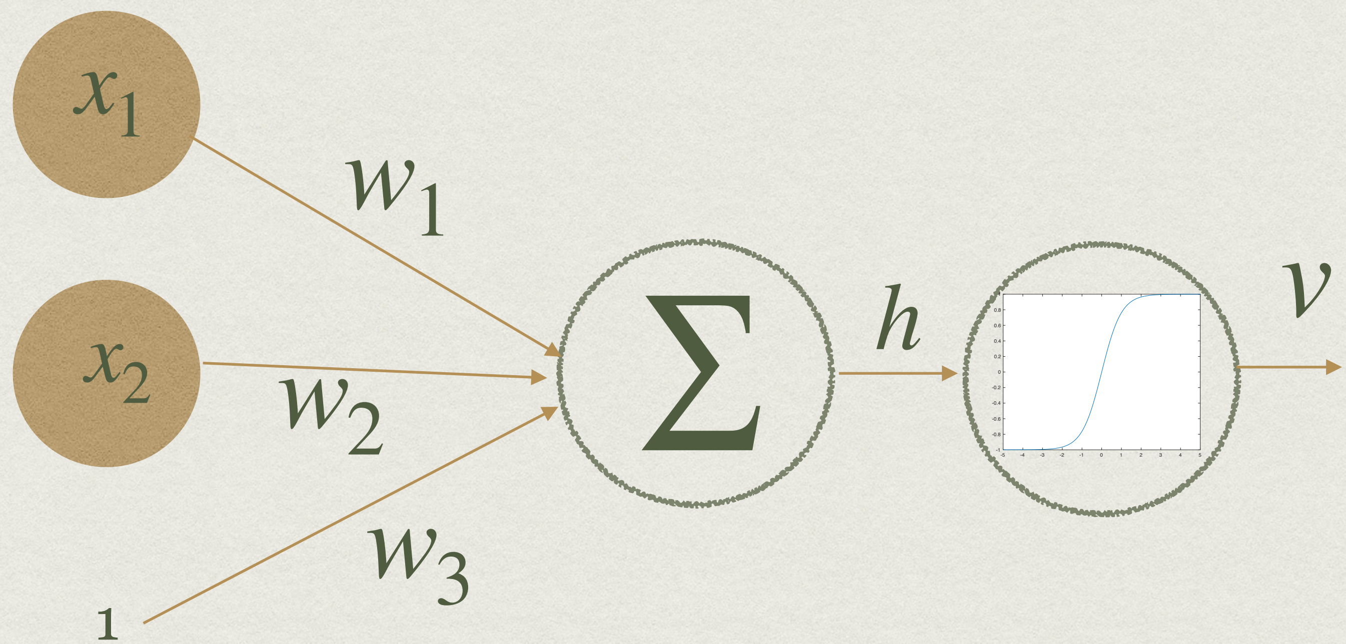


perceptron of Rossenblatt

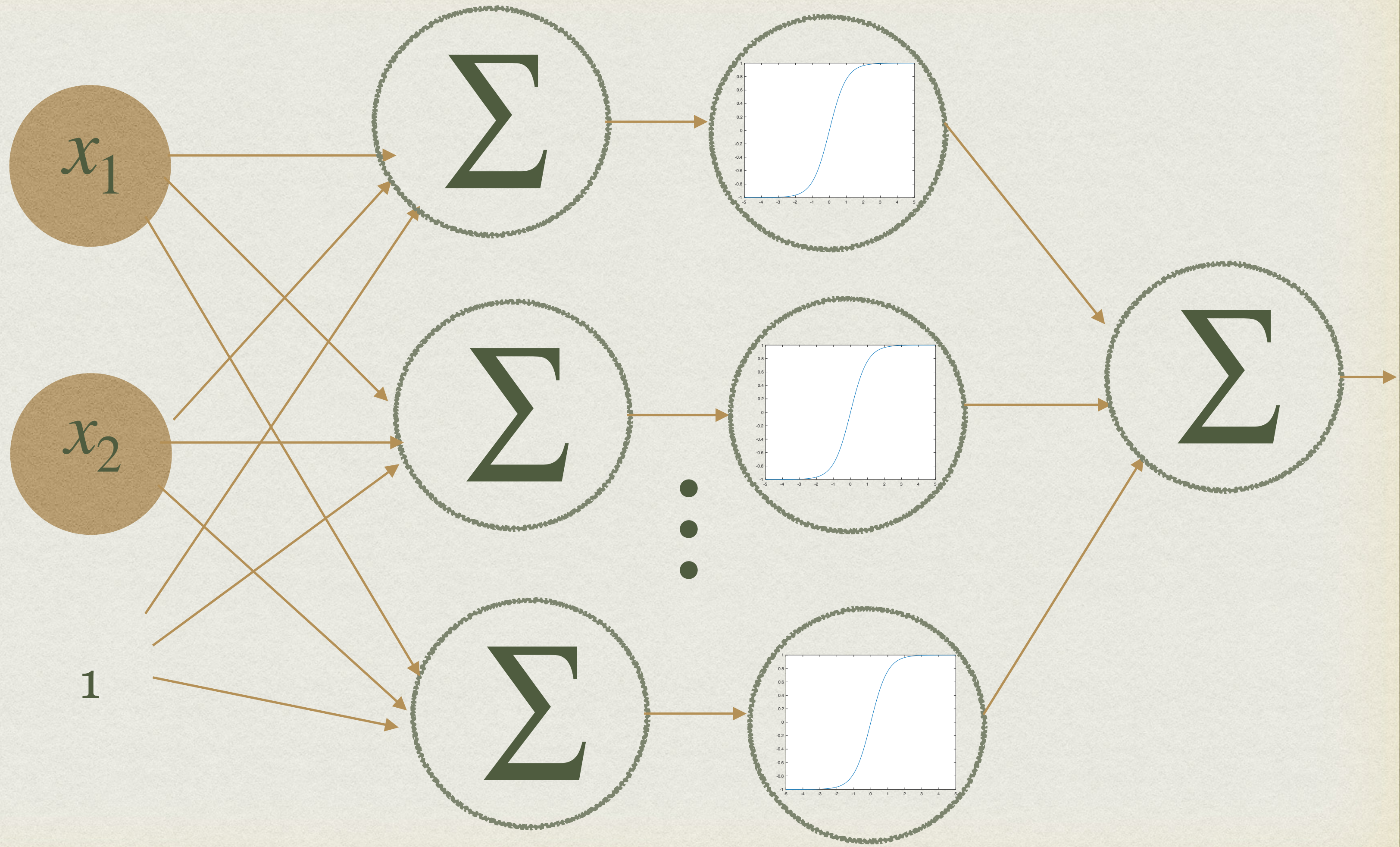
PERCEPTRON

$$h = w_1 x_1 + w_2 x_2 + w_3$$

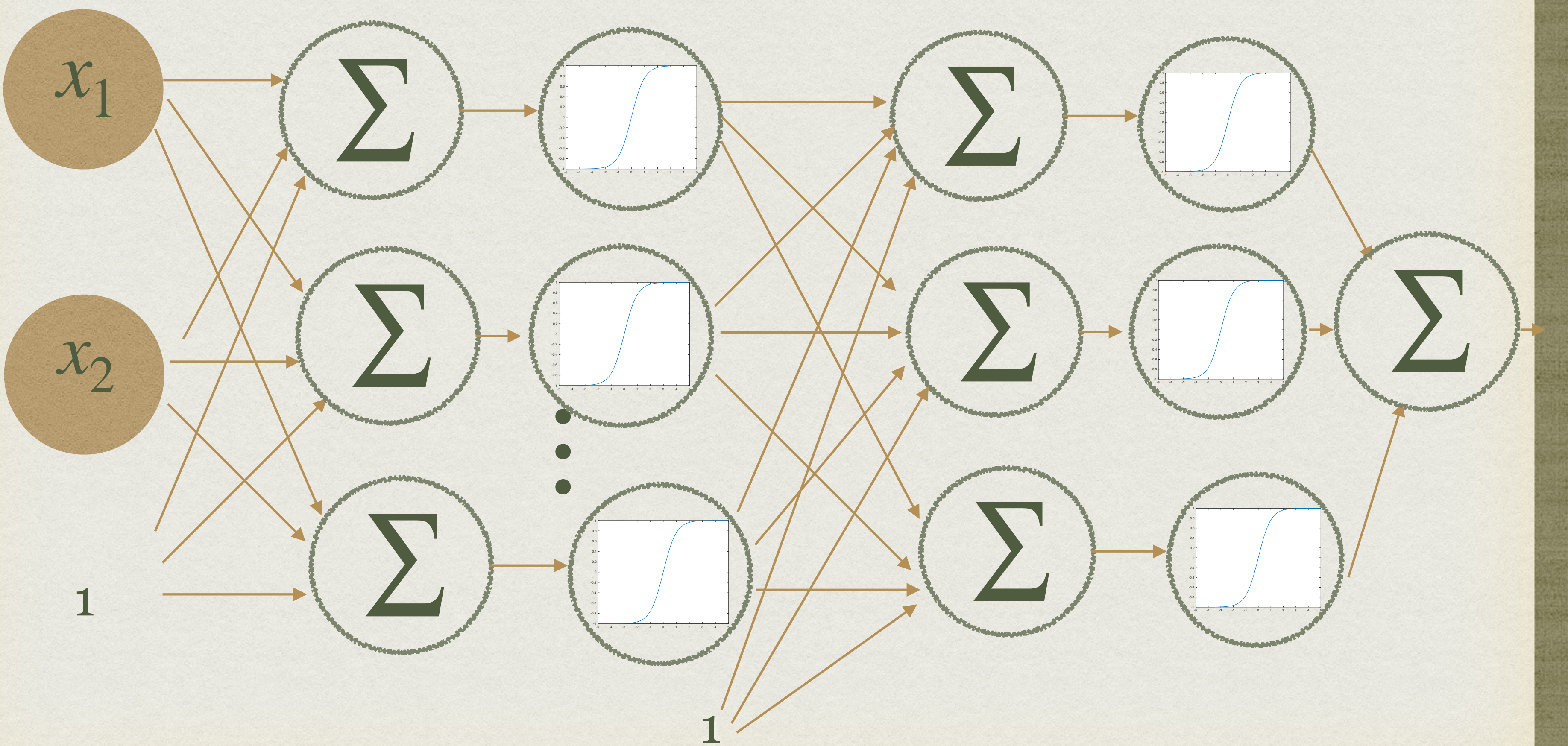
$$v = tanh(h)$$



A MULTILAYER NEURAL NETWORK



A DEEP NEURAL NETWORK



APPROXIMATION

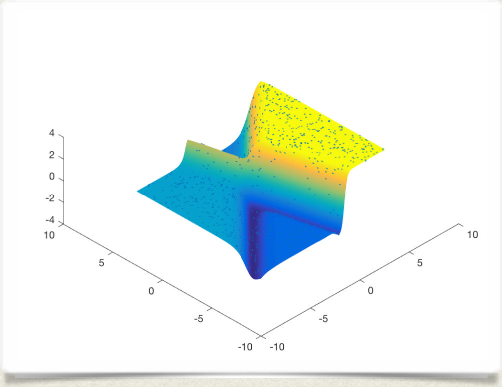
$$v_1 = tanh(2x_1 + 0.5x_2 - 1)$$
$$v_2 = tanh(x_1 - x_2 + 1)$$
$$y = 2cos(2v_1 - v_2 - 1) + sin(v_1 + v_2 - 1)$$

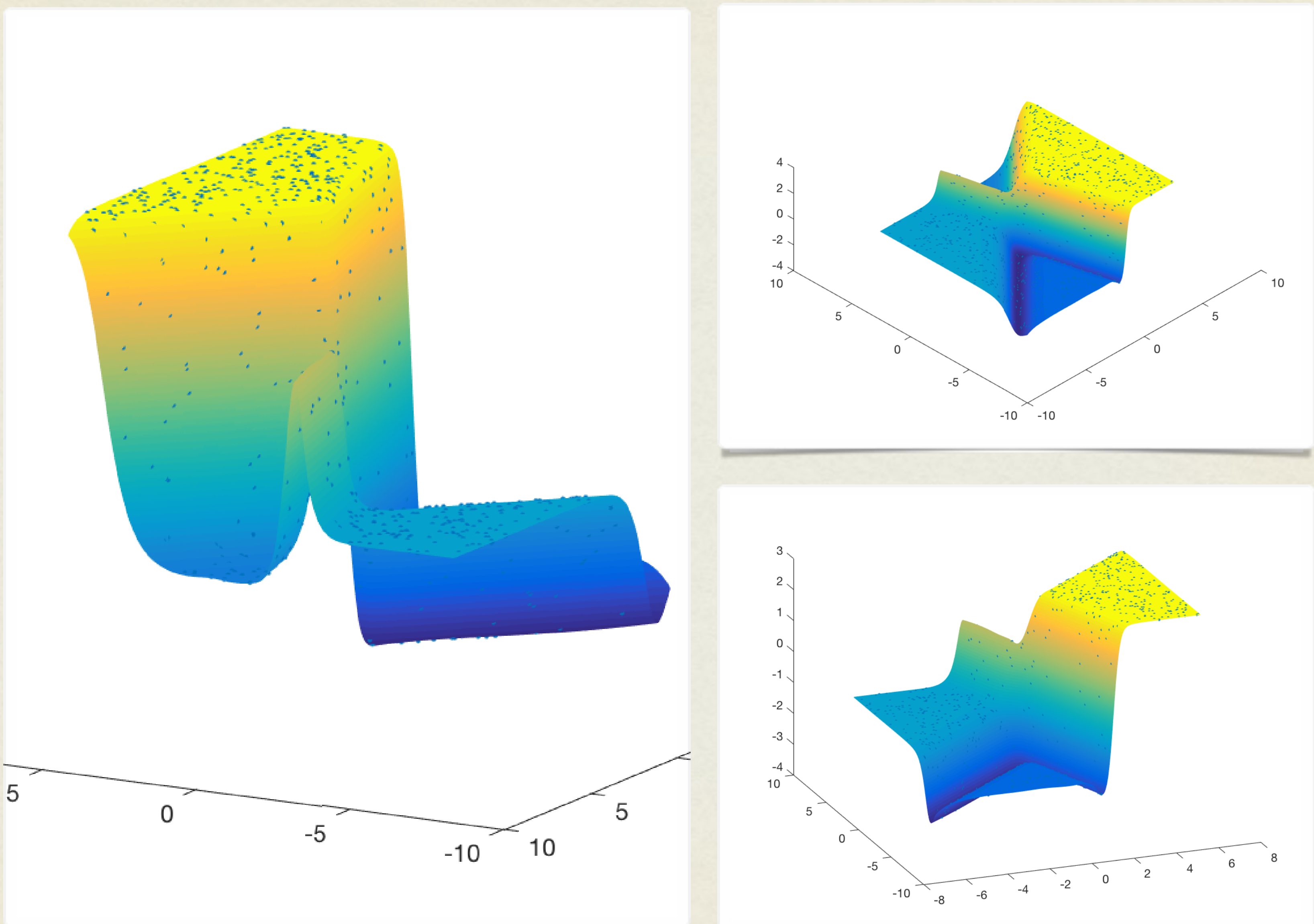
input
output

Learning a deep neural network

optimal interconnections

An approximating network



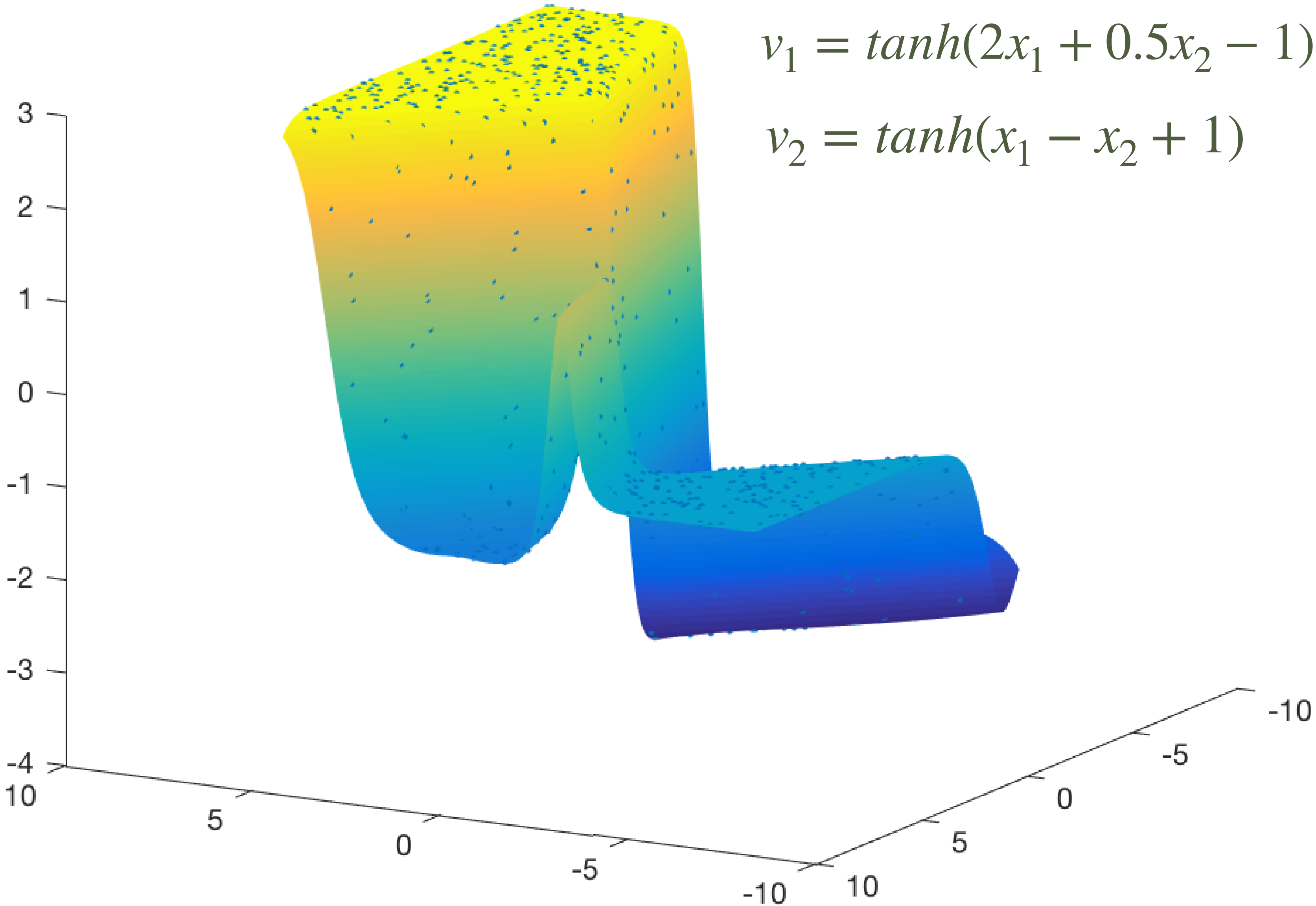



```
v = [tanh(2*x(:,1)+0.5*x(:,2)-1) tanh(x(:,1)-x(:,2)+1)];
y= 2*cos(2*v(:,1)-v(:,2)-1)+sin(v(:,1)+v(:,2)-1);
```

$$y = 2cos(2v_1 - v_2 - 1) + sin(v_1 + v_2 - 1)$$

$$v_1 = tanh(2x_1 + 0.5x_2 - 1)$$

$$v_2 = tanh(x_1 - x_2 + 1)$$



APPROXIMATION

$$v_1 = sign(2x_1 + 0.5x_2 - 1)$$

$$v_2 = sign(x_1 - x_2 + 1)$$

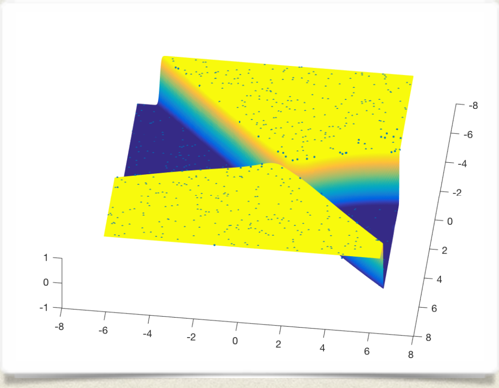
$$y = sign(v_1 v_2)$$

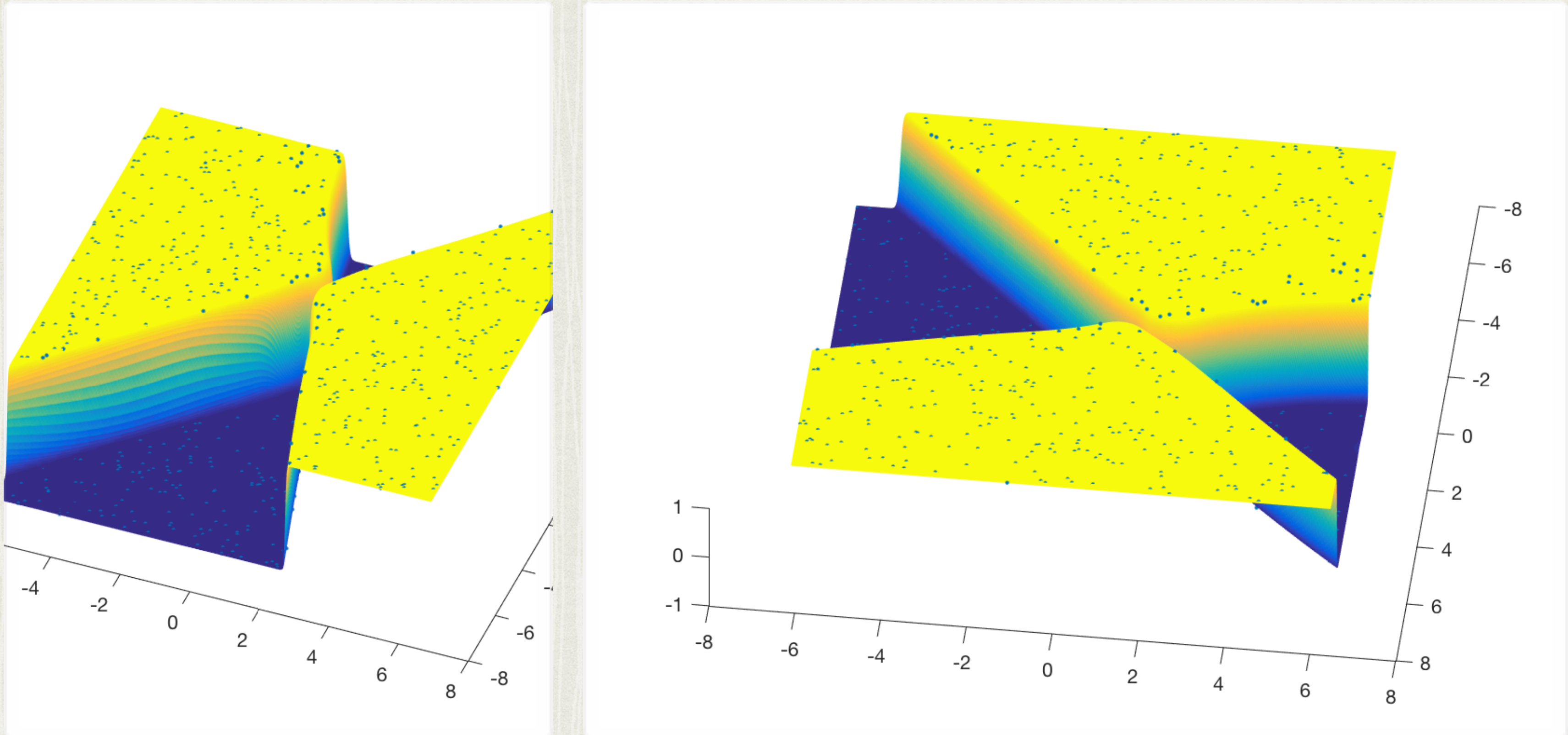
input
output

Learning a deep neural network

optimal interconnections

An approximating network



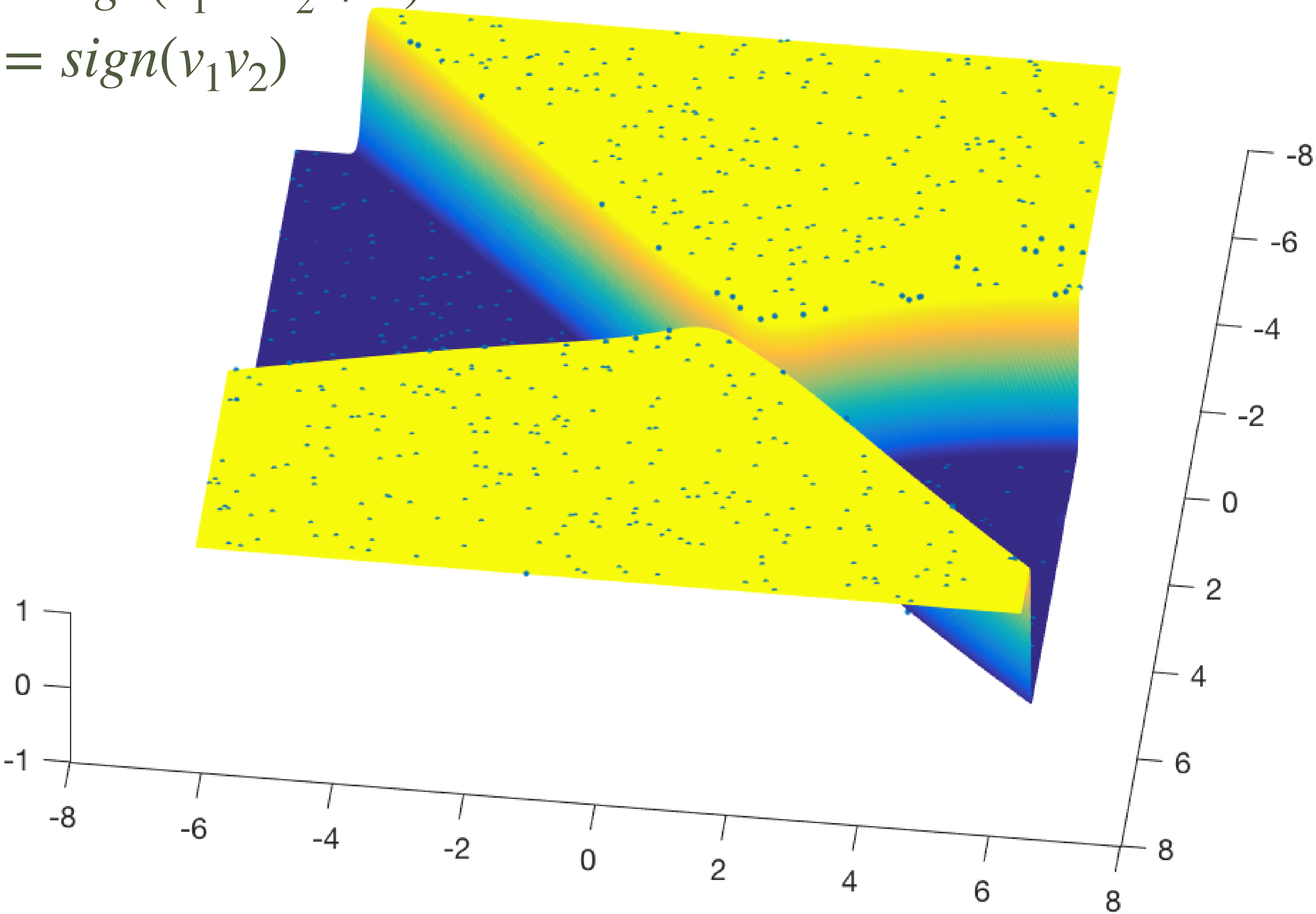



```
v = [sign(2*x(:,1)+0.5*x(:,2)-1) sign(x(:,1)-x(:,2)+1)];
y= sign(v(:,1).*v(:,2));
```

$$v_1 = sign(2x_1 + 0.5x_2 - 1)$$

$$v_2 = sign(x_1 - x_2 + 1)$$

$$y = sign(v_1 v_2)$$



MatconvNet/examples/fast_rcnn

fast_rcnn_demo

Detections for class 'car'

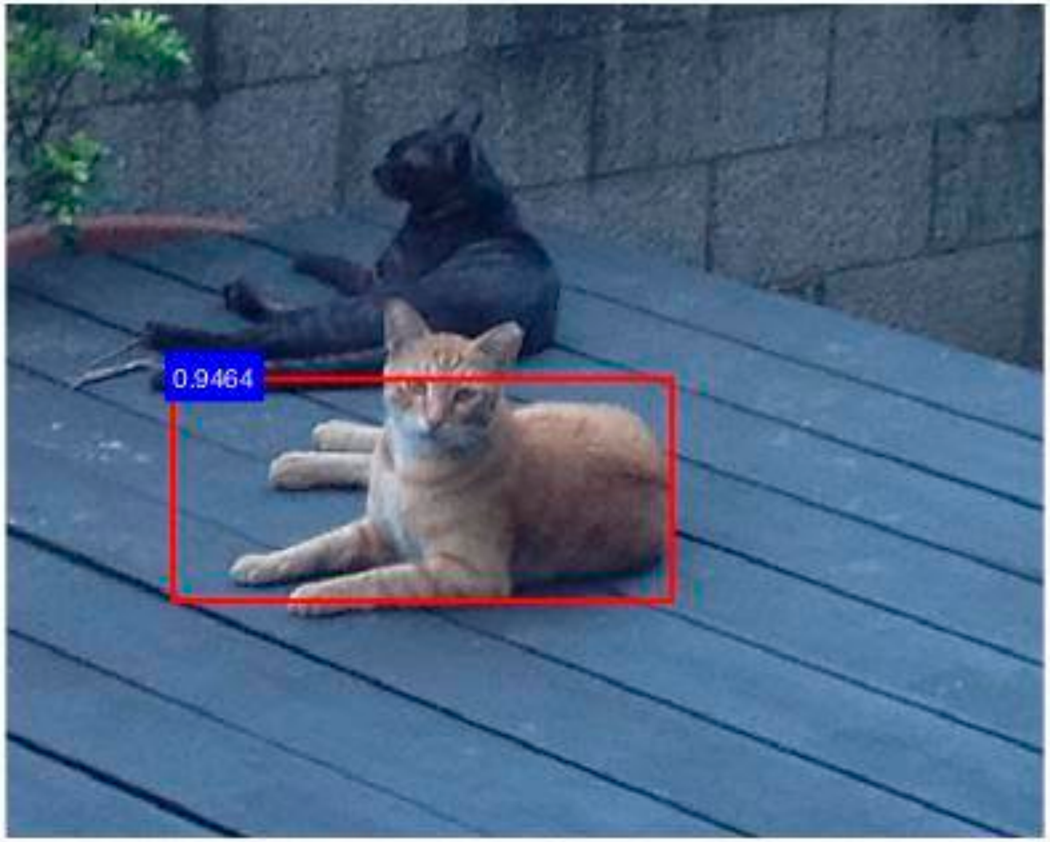
Problem
Architecture
Learning
Data
Execution
Results
your comments
.
.
.
Reference

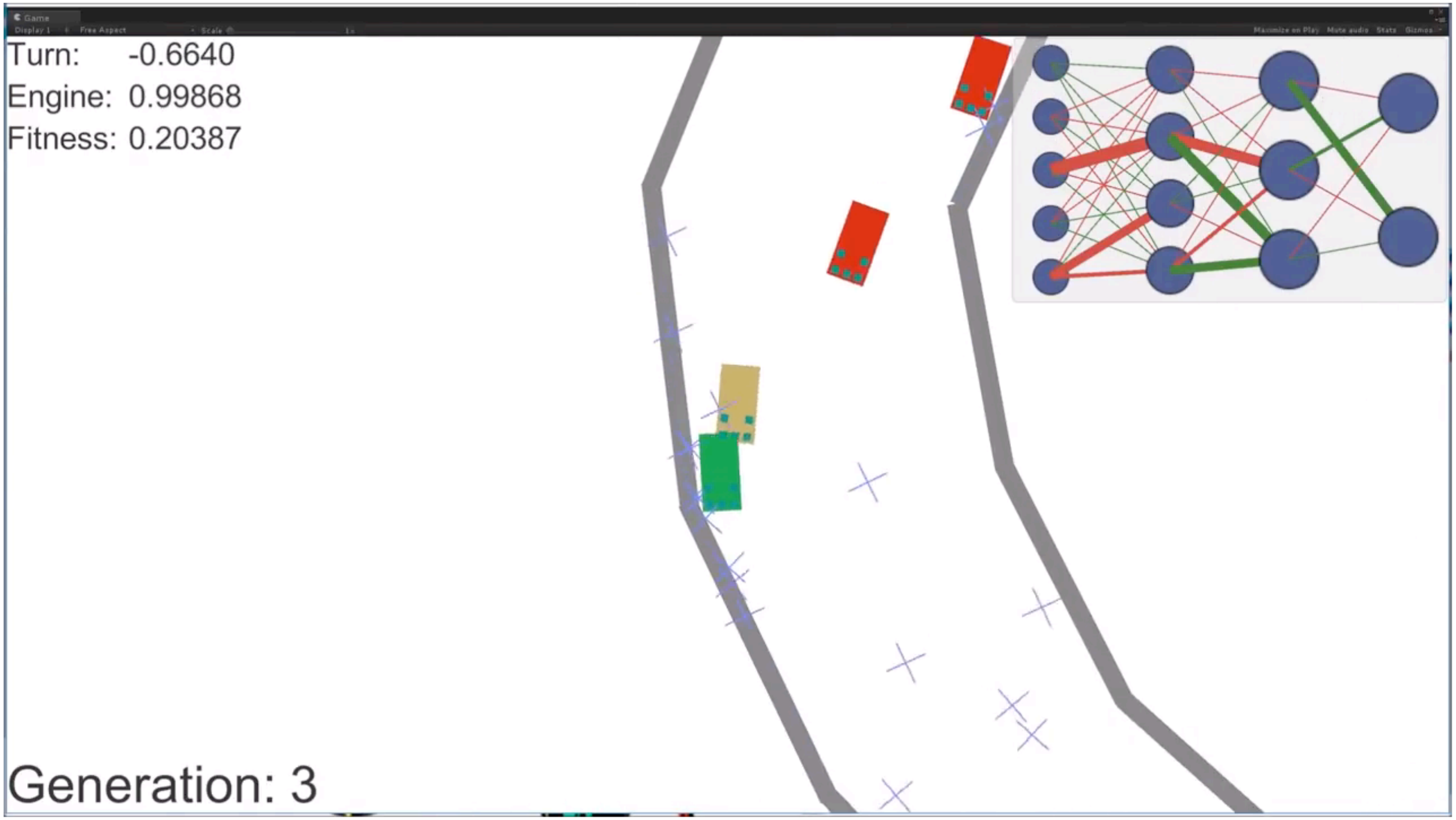
Please download again [fast-rcnn-vgg16-pascal07-dagnn]

http://www.vlfeat.org/matconvnet/pretrained/

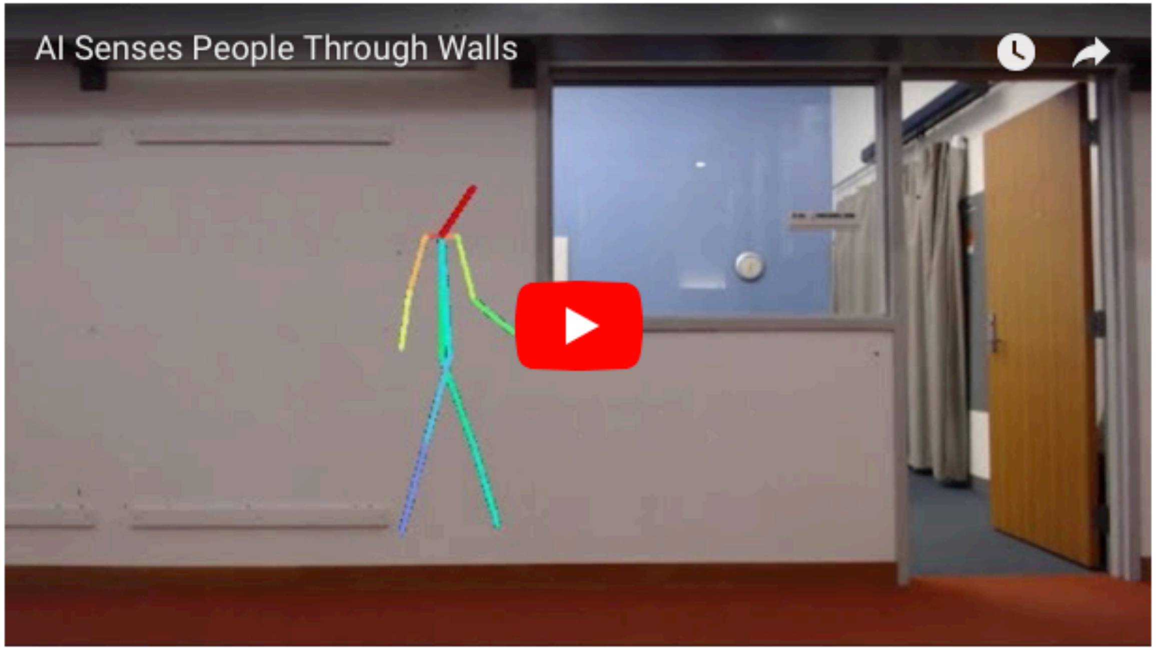
Detections for class 'cow'

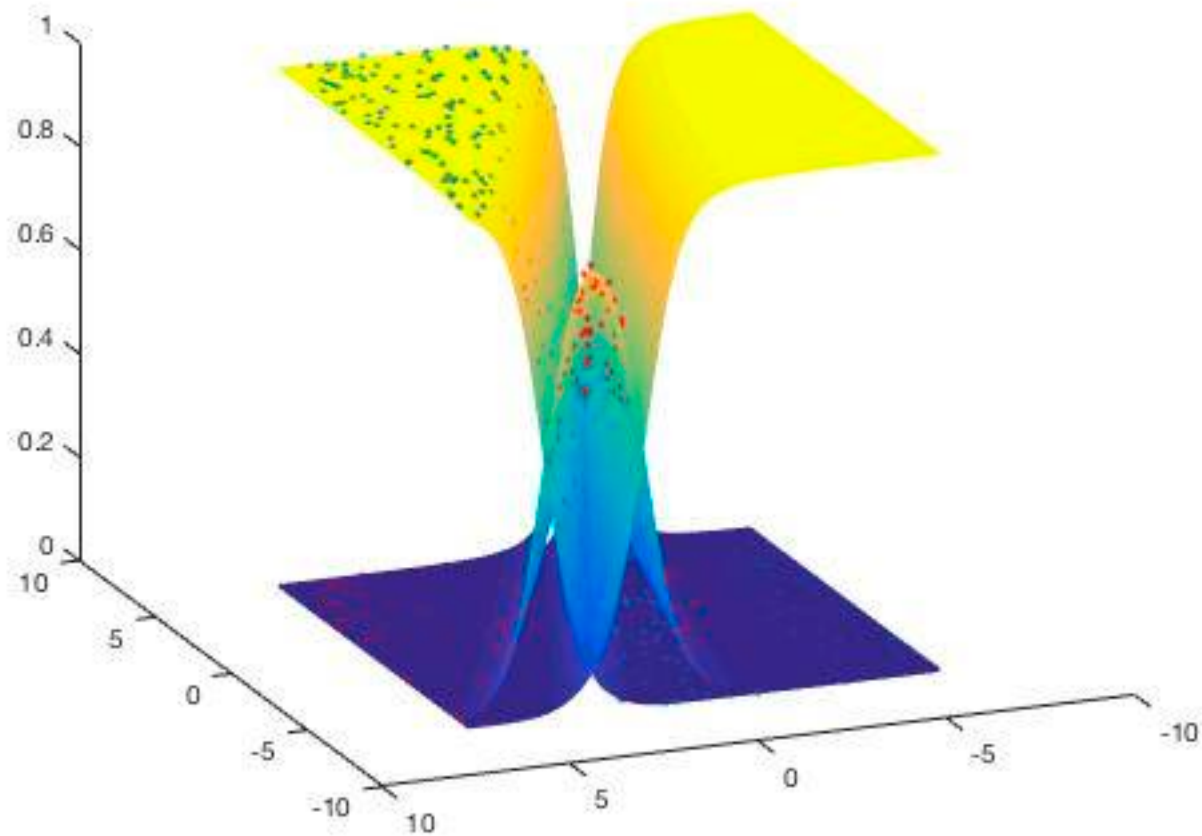
Detections for class 'cat'





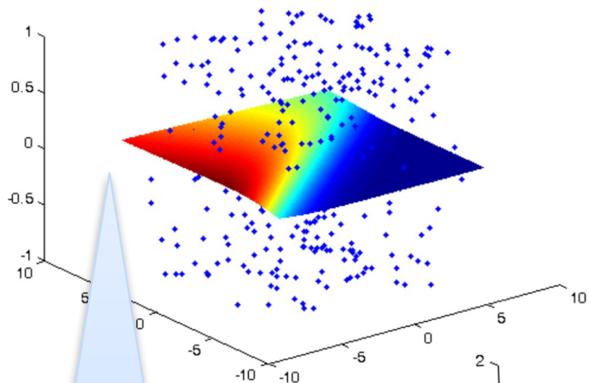
Deep Learning Cars


AI Senses People Through Walls


1
0.8
0.6
0.4
0.2
0
10
5
0
-5
-10
10
5
0
-5
-10

Nonlinear function approximation

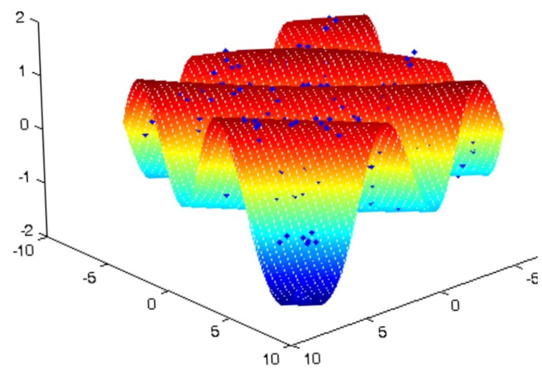
- Given samples from a high-dimensional nonlinear single-valued mapping, the goal is to optimize adaptable parameters for faithful approximation



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

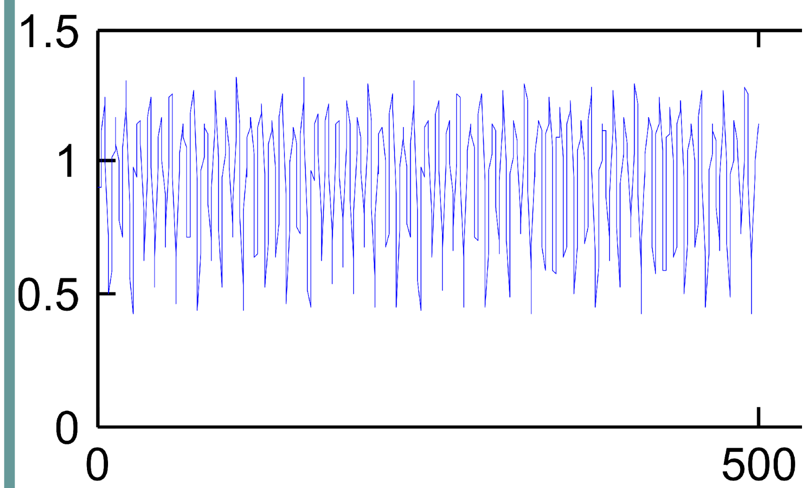
An adaptable network mapping

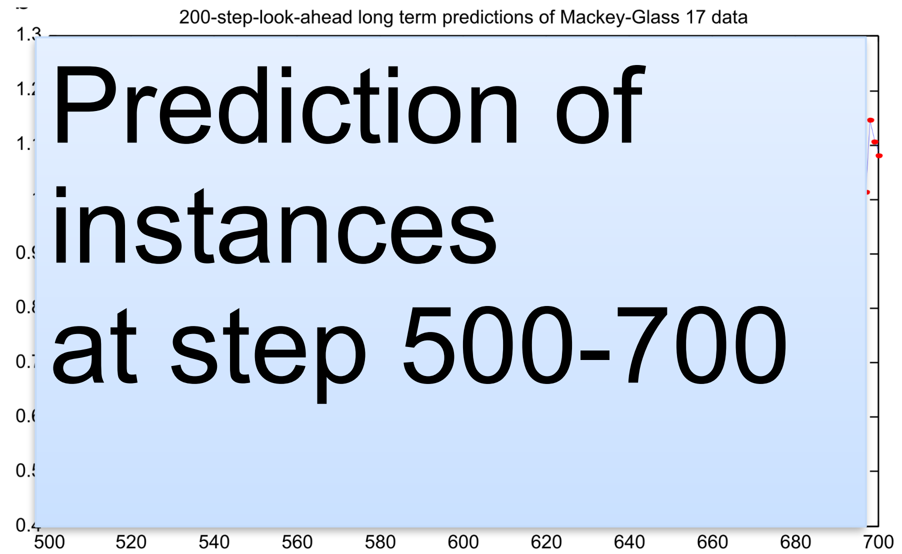
Optimal network parameters



Data driven long-term prediction

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





200-step-look-ahead prediction.

Table 1
Target functions.

$f_1(\mathbf{x}) = \sin(x_1 + x_2)$
$f_2(\mathbf{x}) = x_1^2 + x_2^2$
$f_3(\mathbf{x}) = 0.5x_1^2 - 0.9x_2^2$
$f_4(\mathbf{x}) = \exp(-0.05x_1^2 - 0.09x_2^2)$
$f_5(\mathbf{x}) = \sin([1, -1]^T x) + \exp(-x^T A x)$
$f_6(\mathbf{x}) = \tanh(0.8x_1 + 0.2x_2) + \sin(0.3x_1 - 0.9x_2)$
$f_7(\mathbf{x}) = 0.5\sin(x_1 + x_2) + 0.2x_1 - 0.2x_2$
$f_8(\mathbf{x}) = \exp(-(\mathbf{x} - \mathbf{w}_1)^T A(\mathbf{x} - \mathbf{w}_1)) + \exp(-(\mathbf{x} - \mathbf{w}_2)^T B(\mathbf{x} - \mathbf{w}_1))$
$f_9(\mathbf{x}) = f_8(\mathbf{x}) + 0.5\ \sin(x_1 + 0.3x_2) + 0.5\ \sin(0.2x_1 - 0.8x_2)$
$f_{10}(x) = \sin(x_1 + x_2 + x_3) + \cos(x_1 + x_2 + x_3)$
$f_{11}(x) = \tanh(x_1 + x_2 + x_3 + x_4)$

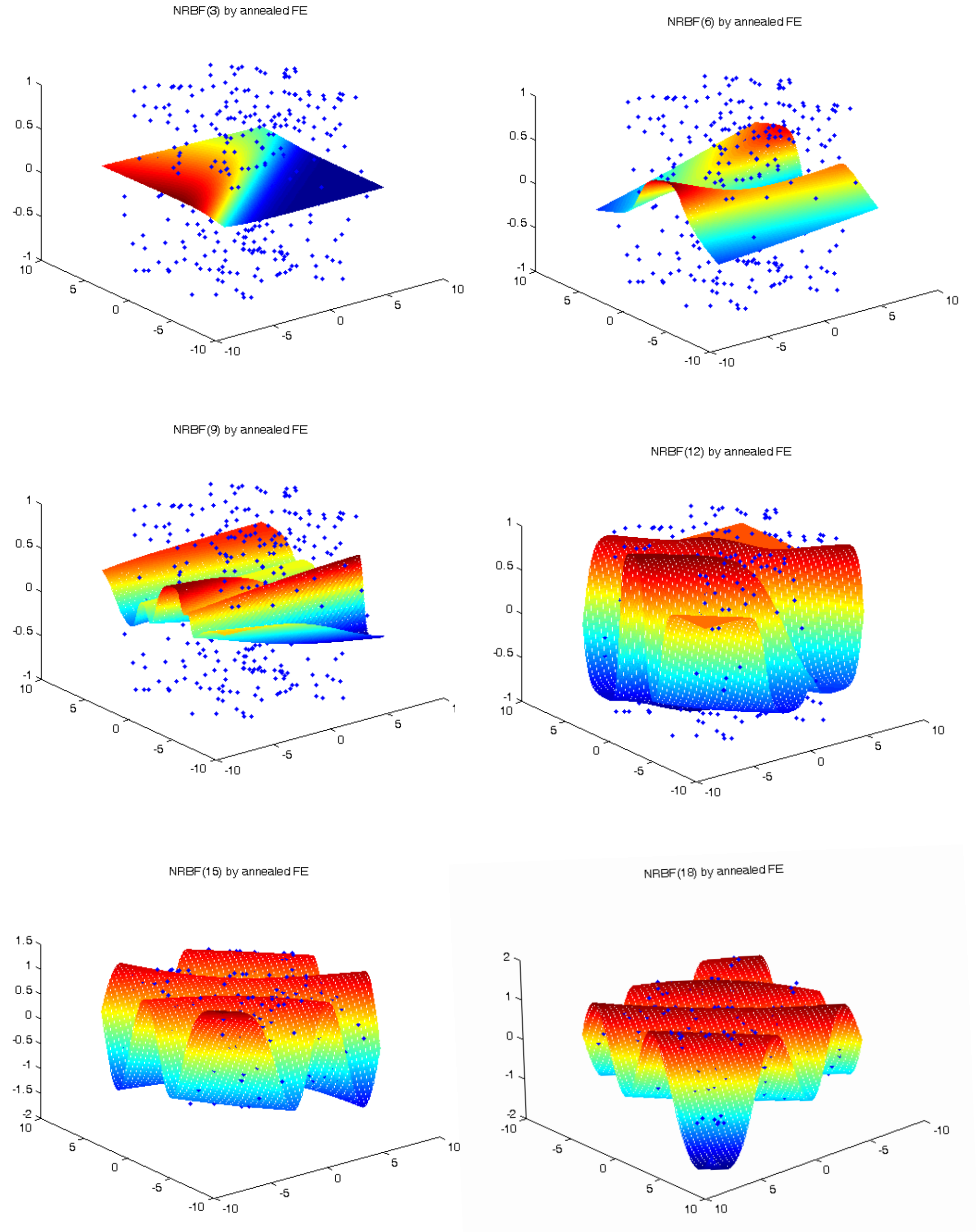

NRBF(3) by annealed FE
NRBF(6) by annealed FE
NRBF(9) by annealed FE
NRBF(12) by annealed FE
NRBF(15) by annealed FE
NRBF(18) by annealed FE

Figure 4

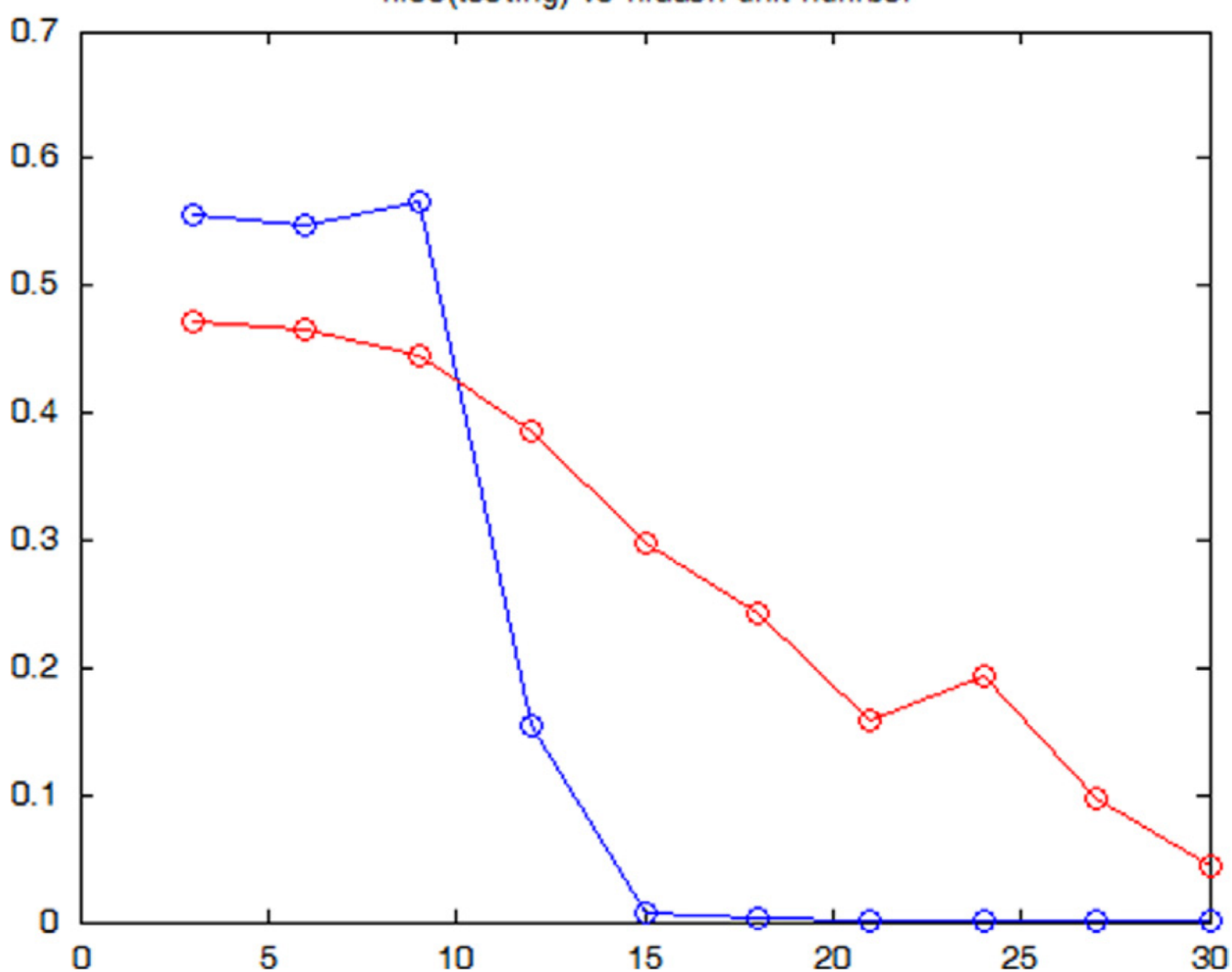


Fig. 5. Mean square testing errors of annealed competitive learning (blue curve) and the Rätsch method (red curve) in approximating f_1 versus the numbers of hidden units. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

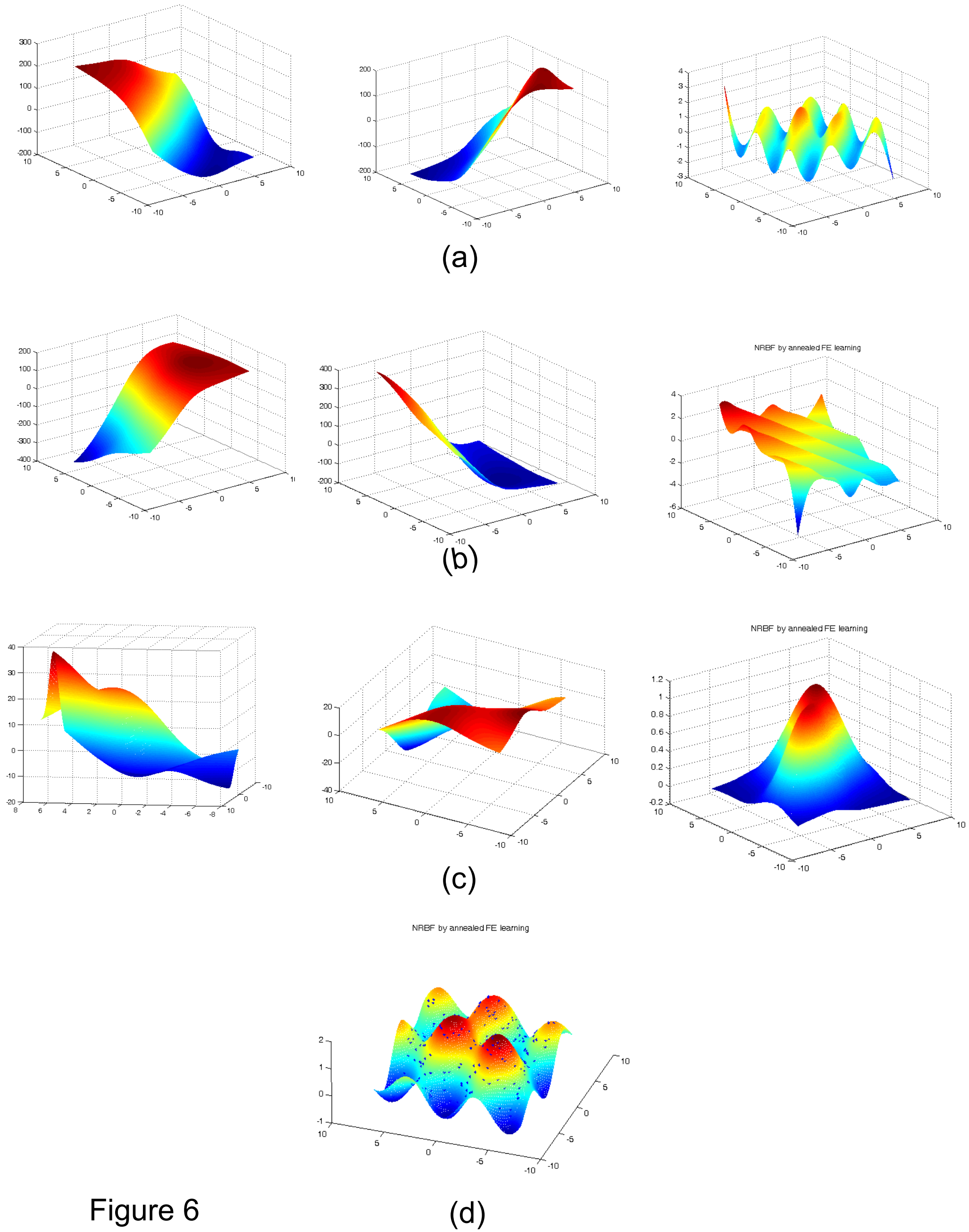
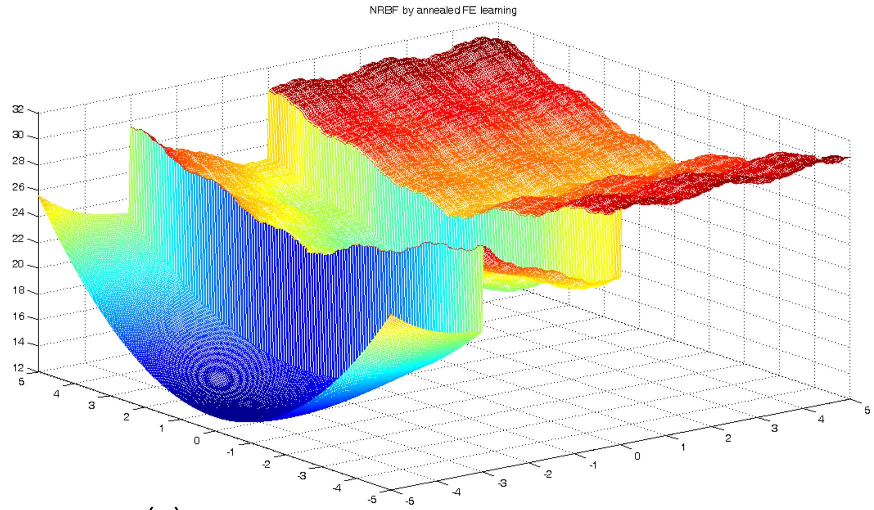
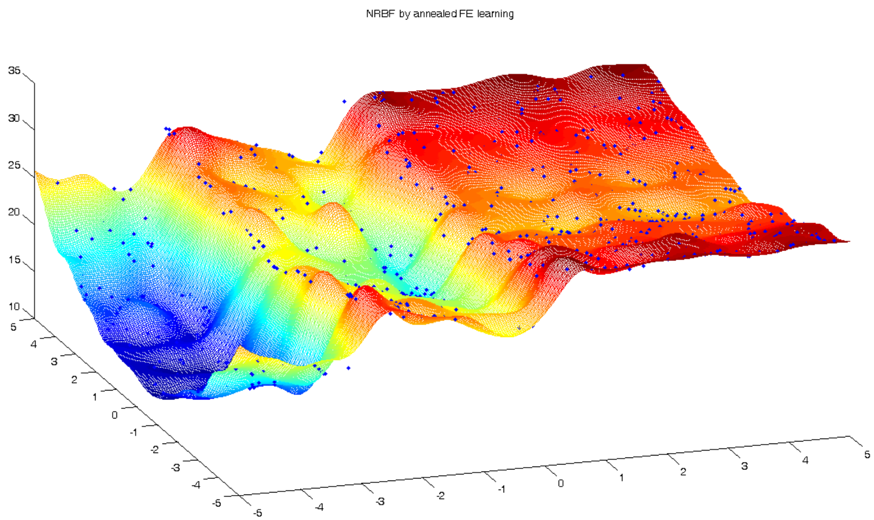

(a)
NRBF by annealed FE learning
(b)
NRBF by annealed FE learning
(c)
NRBF by annealed FE learning
(d)

Figure 6



(a)



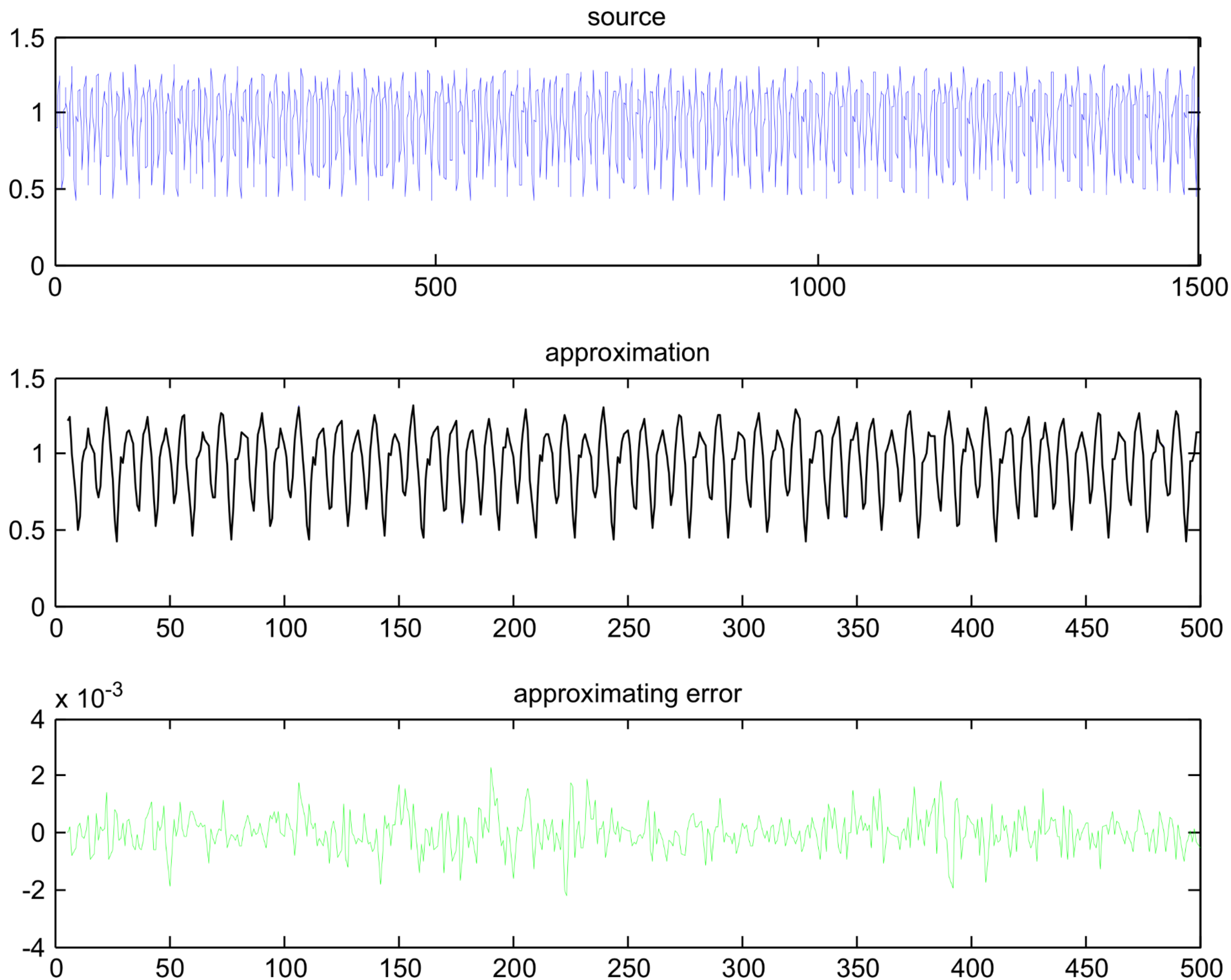
(b)

Figure 7

Chaotic differential function approximation

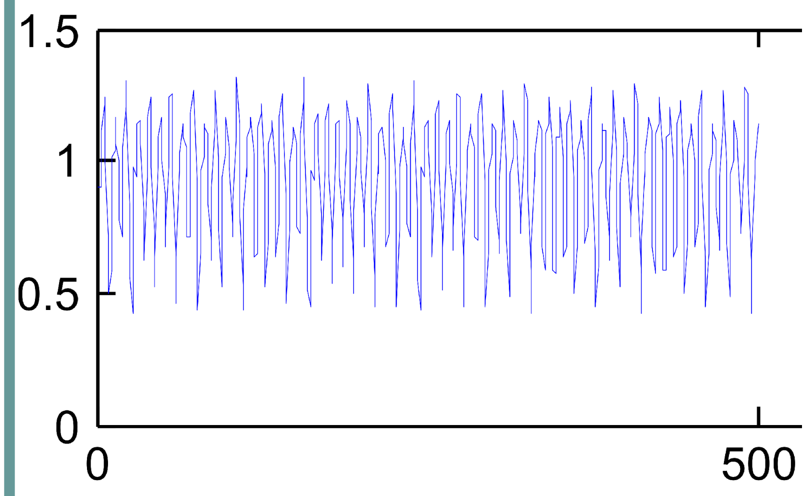
$$\frac{\partial x}{\partial t}=\frac{ax(t-\tau)}{1+x^{c}(t-\tau)}-bx(t),$$

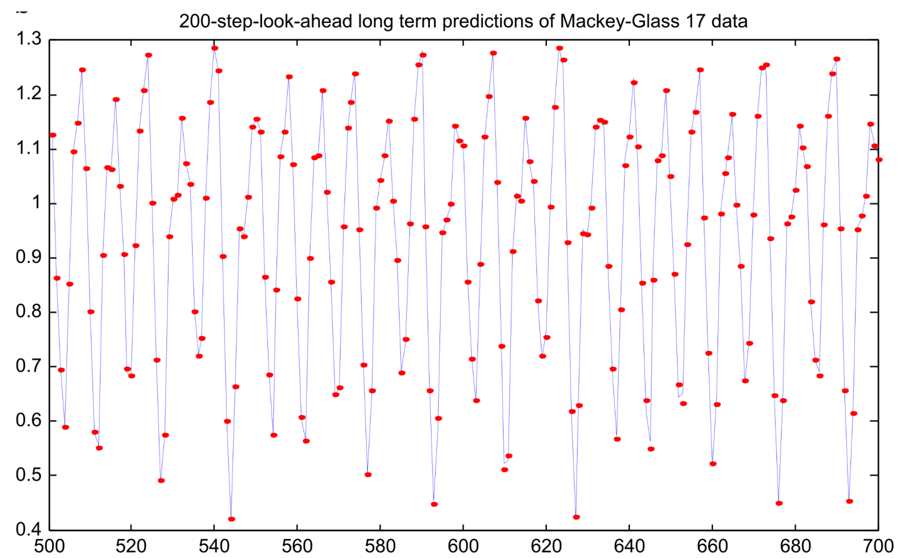
$\tau=17$, $a=0.2$, $c=10$ and $b=0.1$.


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

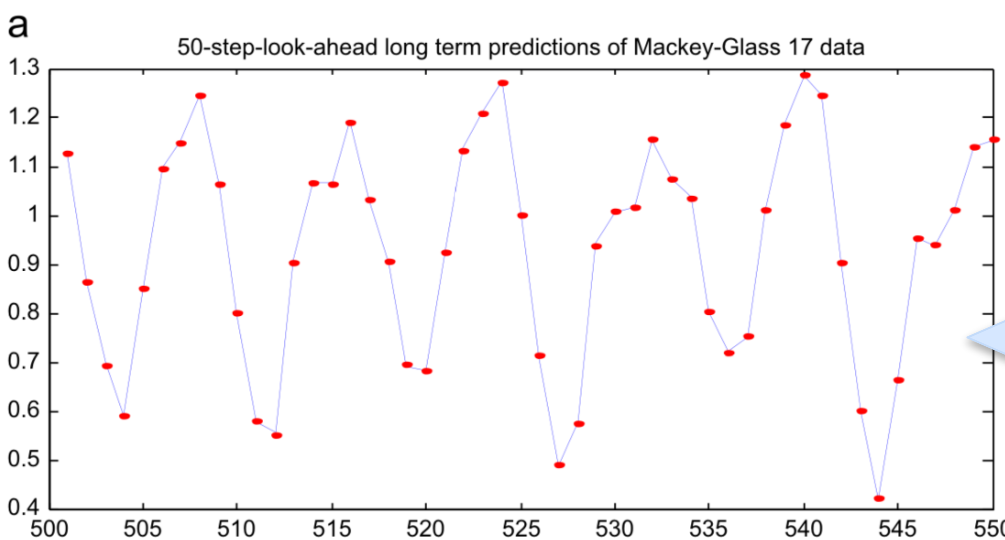
Data driven long-term prediction

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

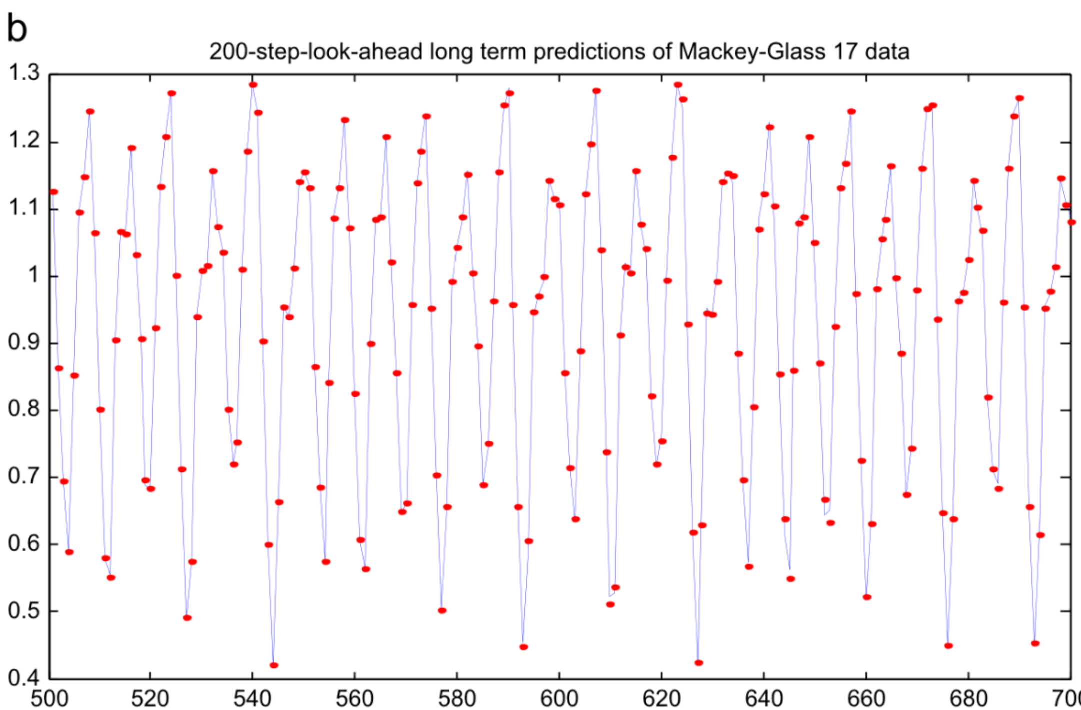




200-step-look-ahead prediction.



correlation coefficient 0.9999



correlation coefficient 0.9993

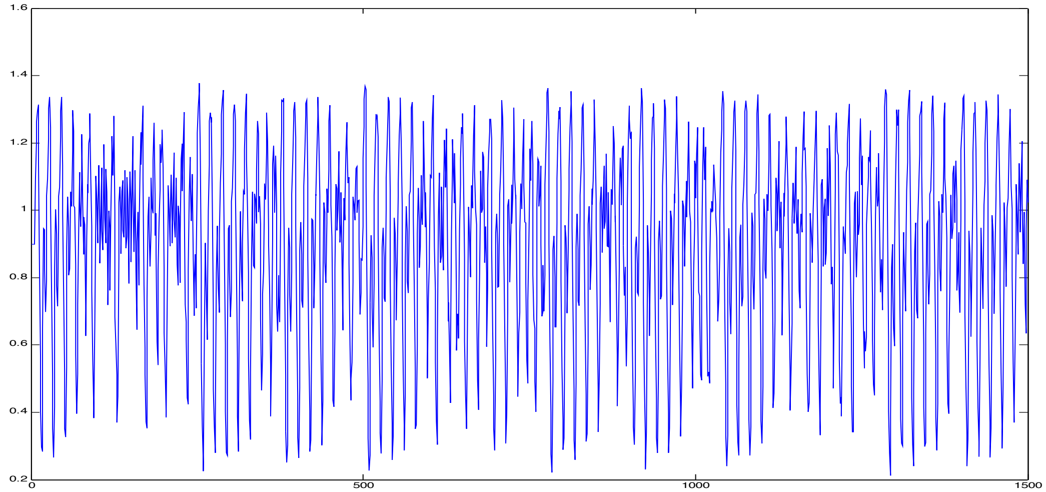
Fig. 10. n-step-look-ahead predictions of MG17 time series with $n = 50$ and $n = 200$ by annealed cooperative–competitive learning with $K = 2$. (For interpretation of the

Mackey-Glass 30

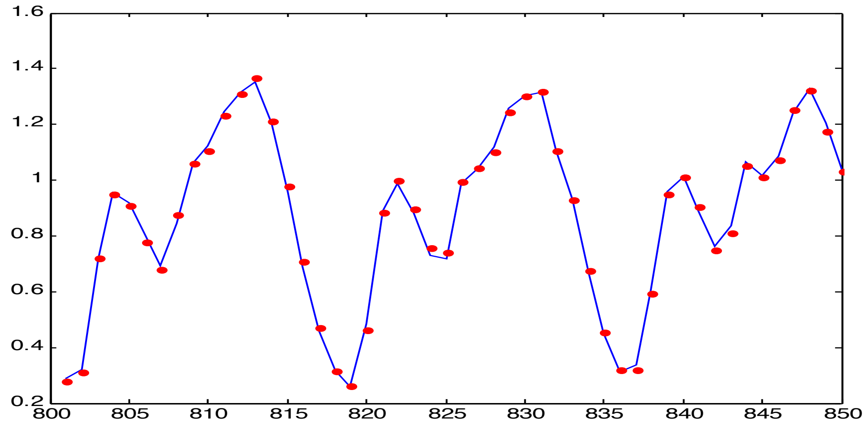
$$\frac{\partial x}{\partial t} = \frac{a x(t-\tau)}{1 + x^{c}(t-\tau)} - b x(t),$$

$\tau = 30$ $a = 0.2$, $c = 10$ 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.

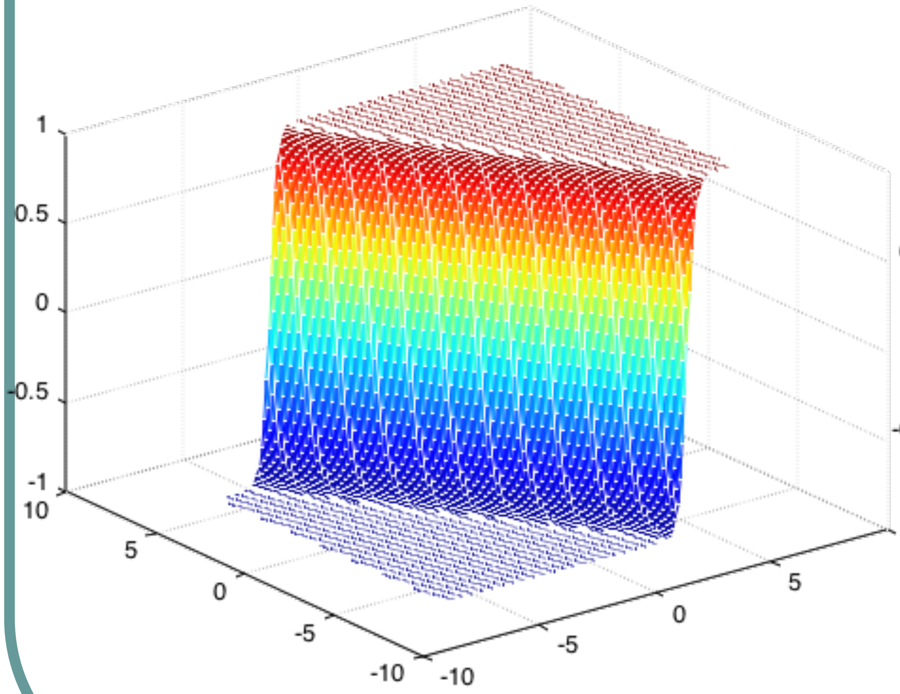
Function Approximation

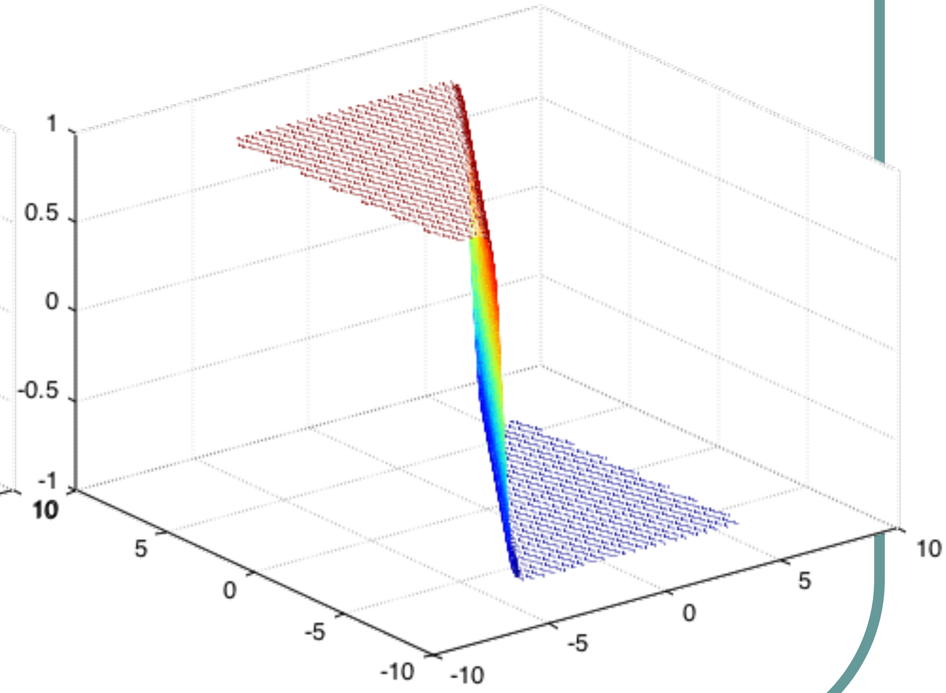
- Multiple input variables
- Nonlinear mapping from domain to range

tanh

$$f(x1,x2) = y_1 = 2x_1 + 3x_2$$

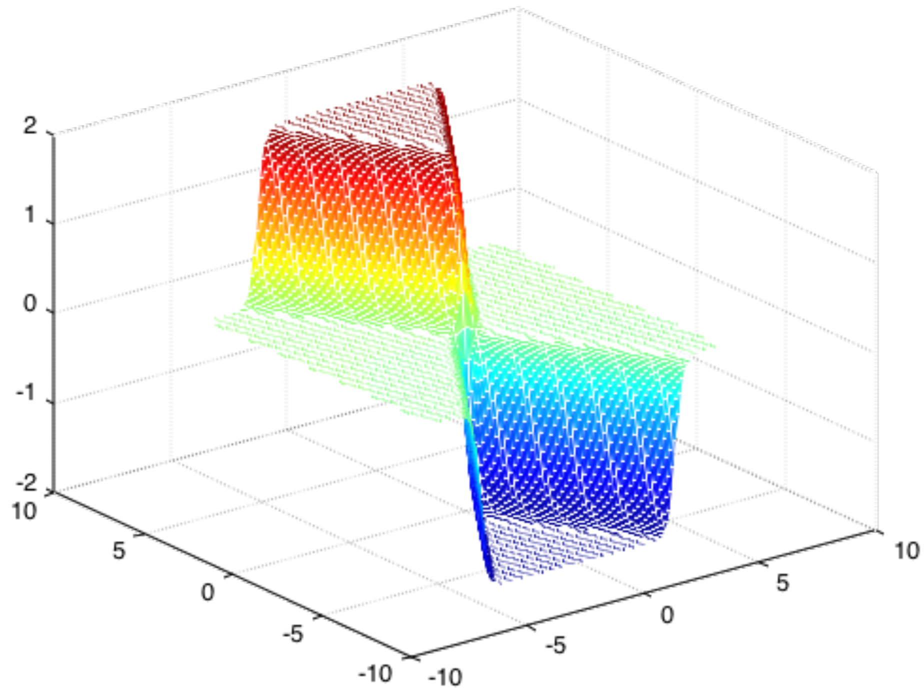
$$f(x1,x2) = y_2 = 2x_1 - 3x_2$$





Two post-nonlinear projections

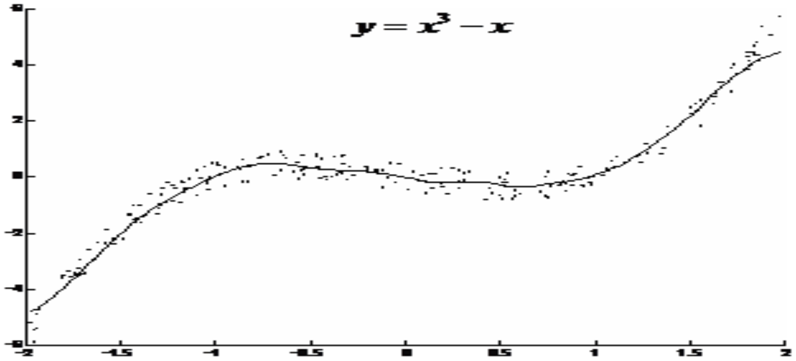
$$f(x1, x2) = \tanh(2x_1 + 3x_2) + \tanh(2x_1 - 3x_2)$$

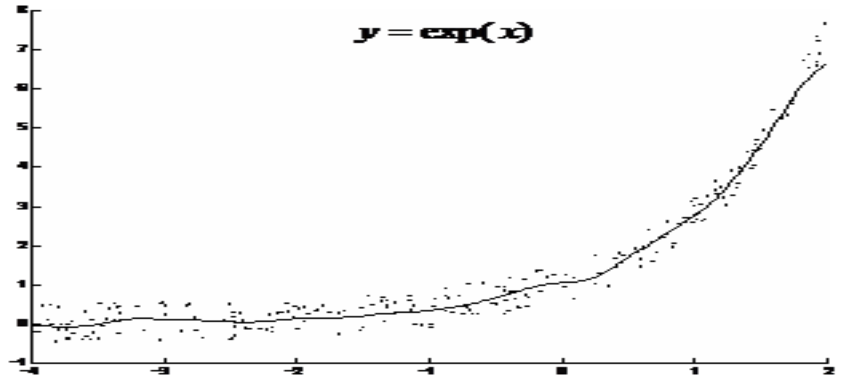


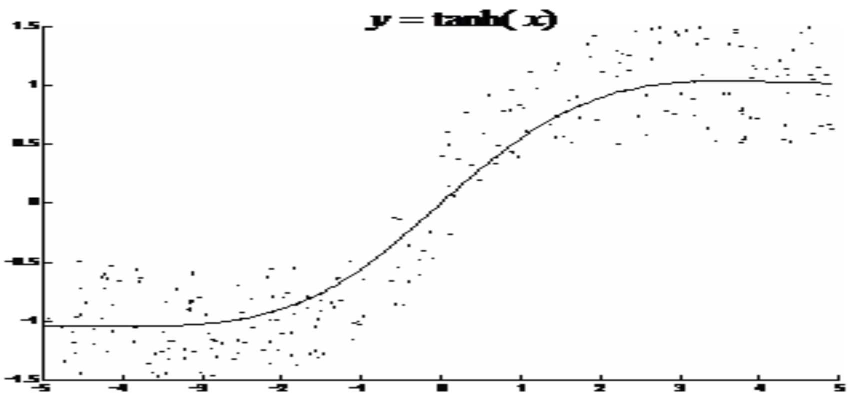
Intelligence computations

- Neural networks
- Machine Learning
- Data analysis
- Numerical computations

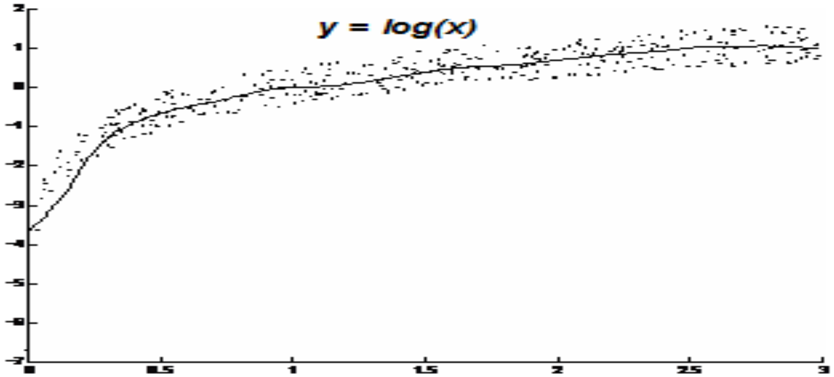
Function approximation

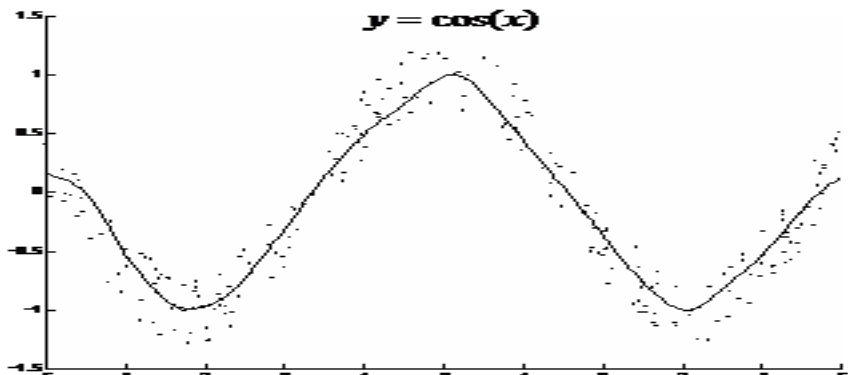






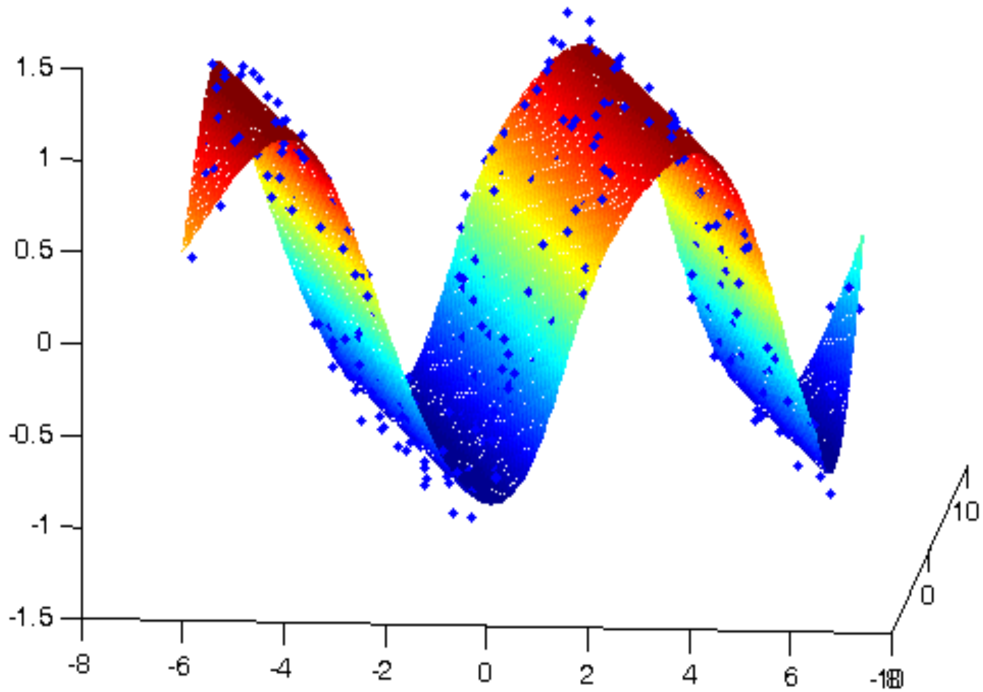
One-dimensional function approximation





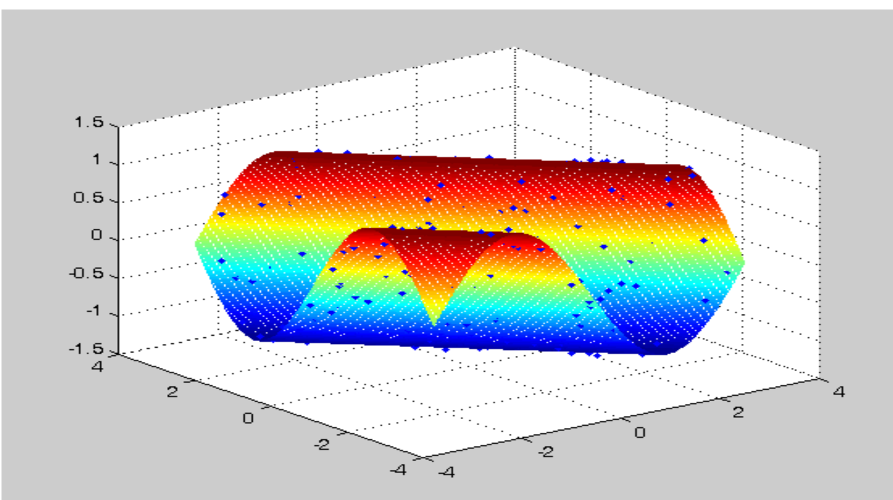
LM learning for MLP

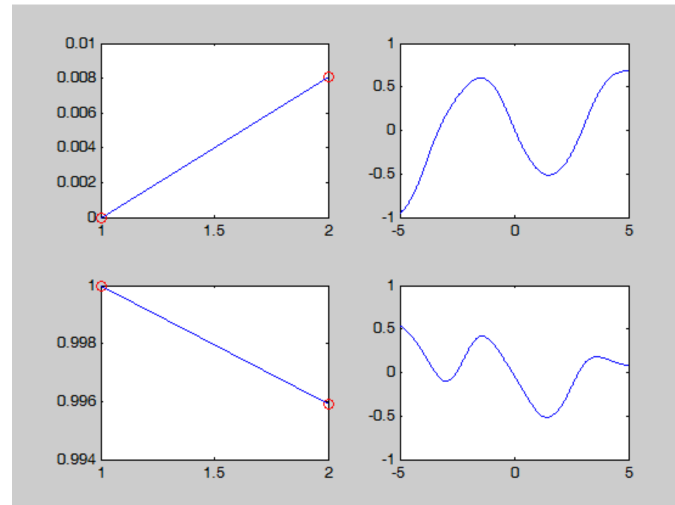
Two-dimensional Function Approximation

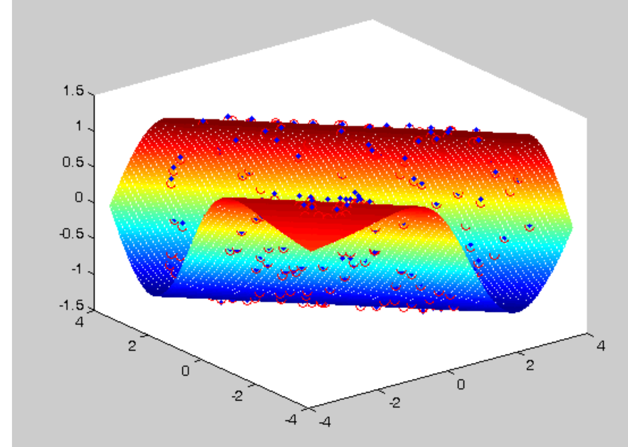


Learning MLPotts networks

Two-dimensional function approximation
by learning MLPotts networks
(Wu 2008)

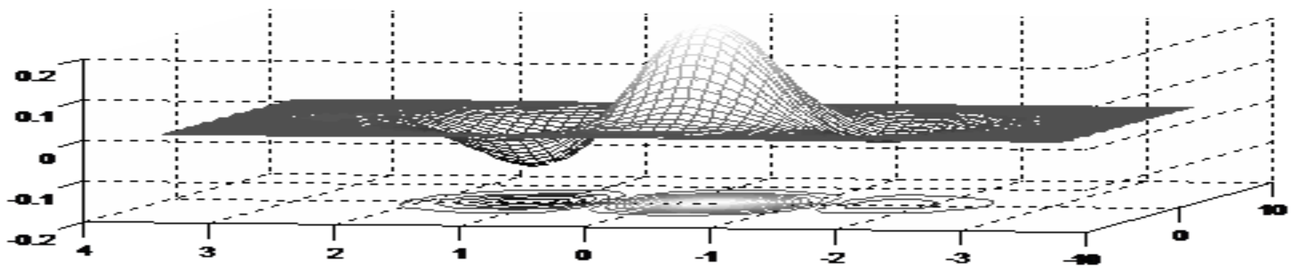


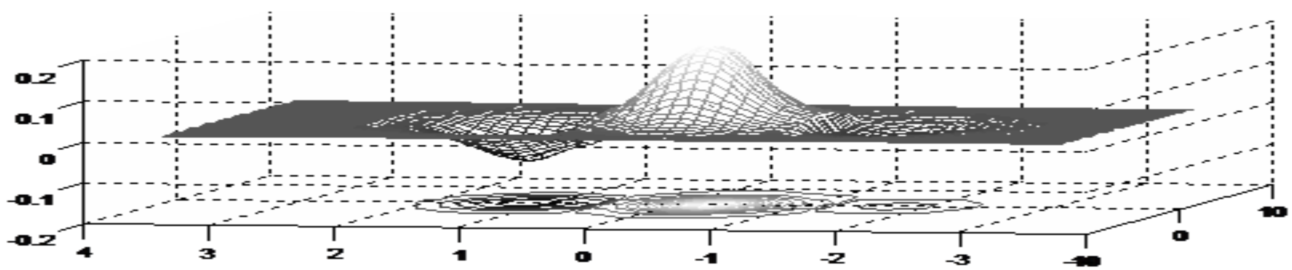




Approximating Gabor function

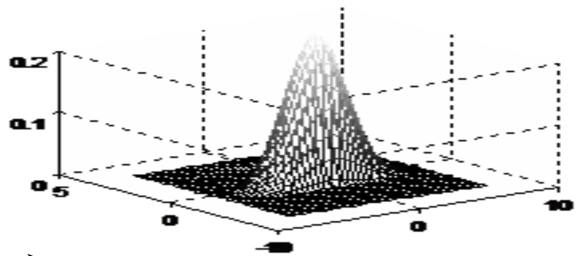
Learning generalized adalines (Wu et al 2006)



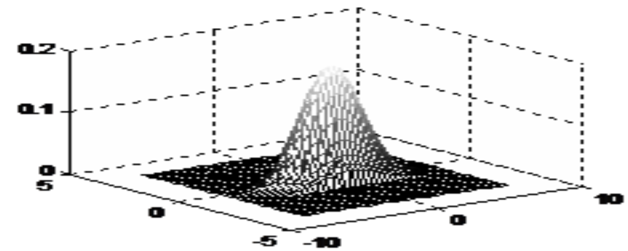


Learning gadaline networks

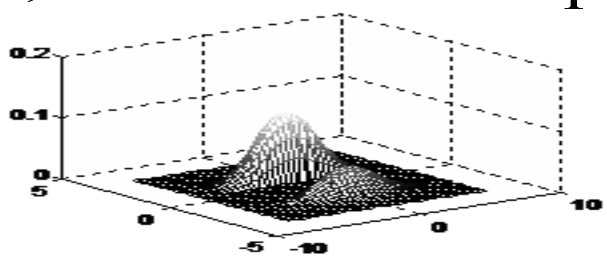
$y_7^{+}(\mathbf{z})$



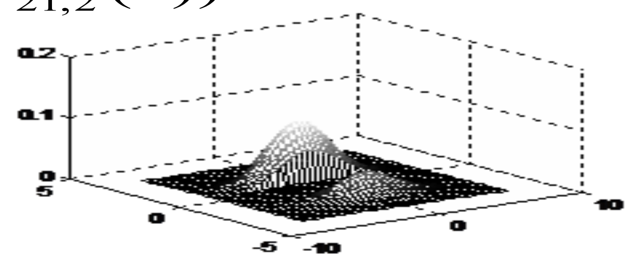
$\exp(G_{21,2}^{+}(\mathbf{z}))$



$y_7^{-}(\mathbf{z})$



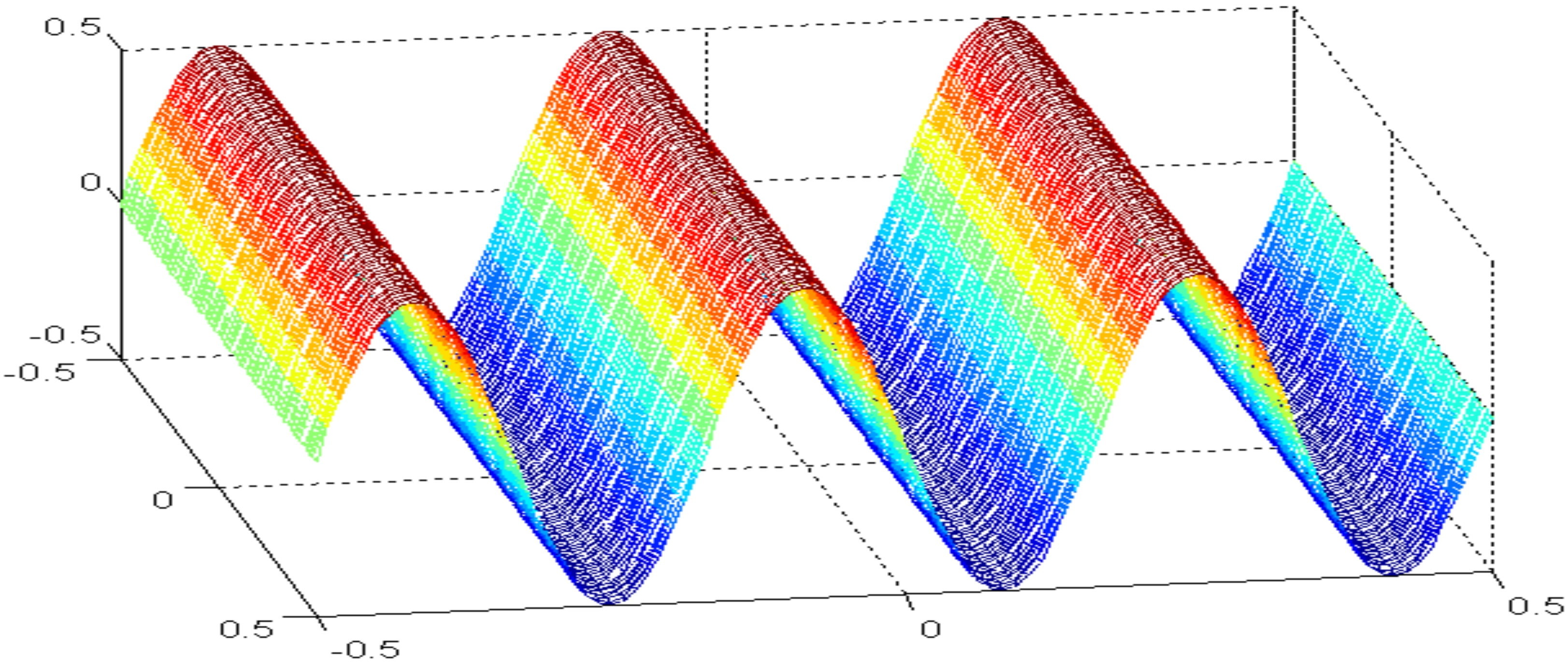
$\exp(G_{21,2}^{-}(\mathbf{z}))$

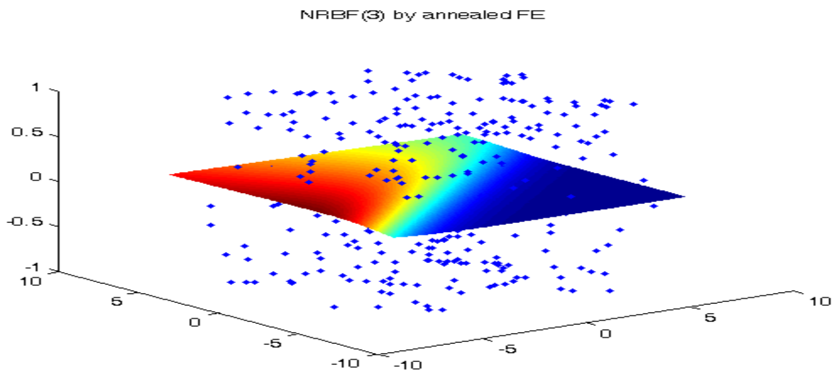


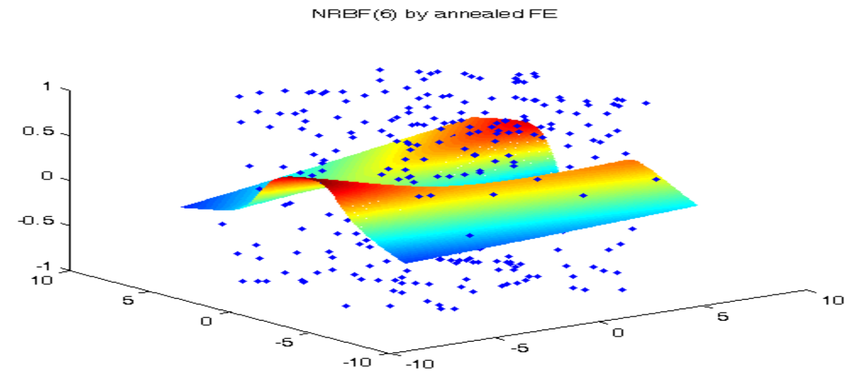
Sinusoidal function approximation

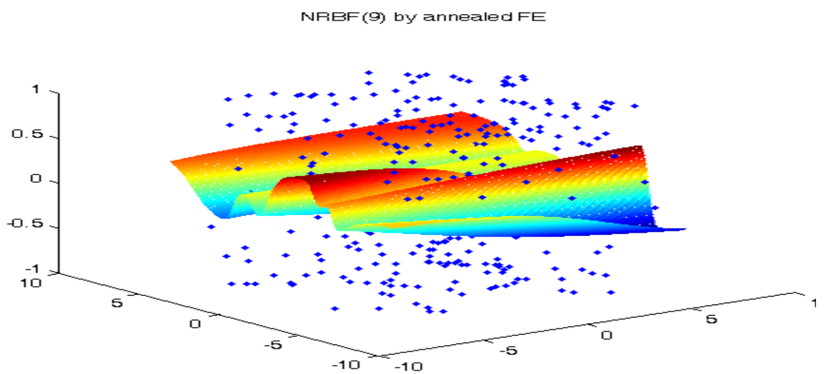
Learning gadaline networks (Wu et al 2006)

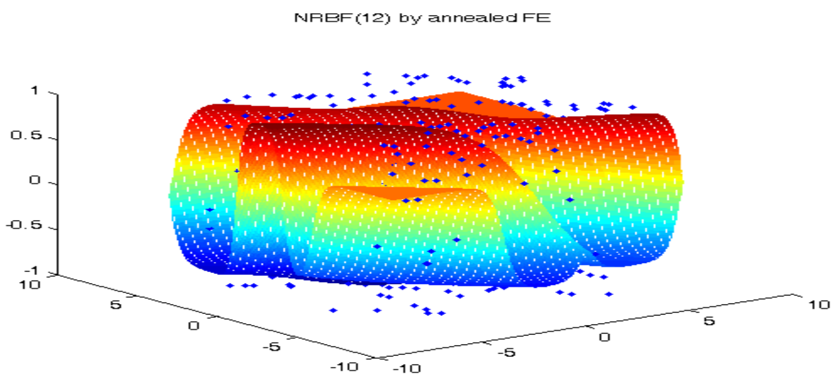
$y(\mathbf{z})$

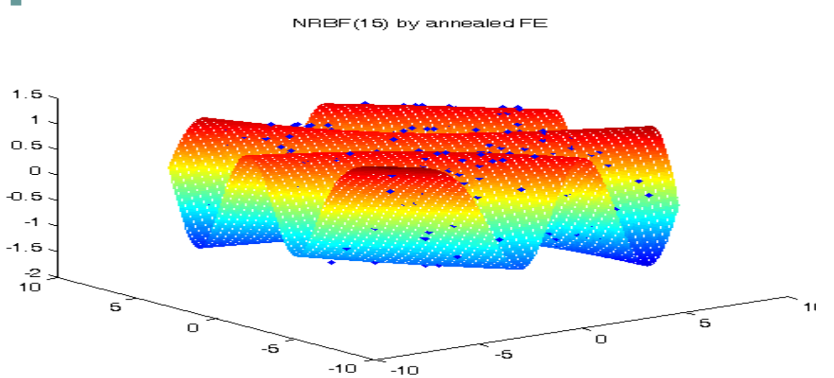


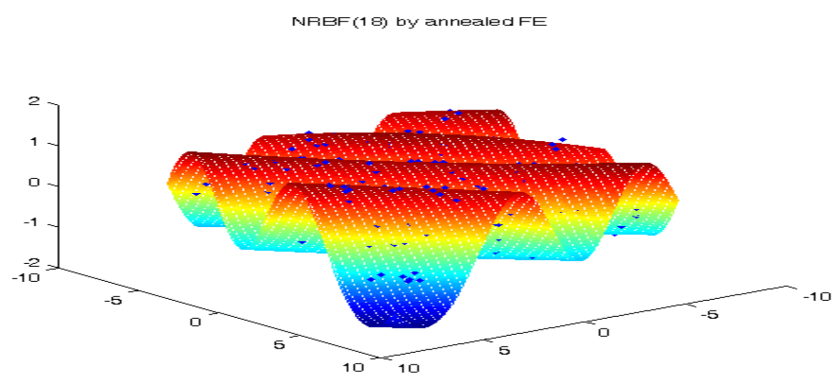

NRBF(3) by annealed FE

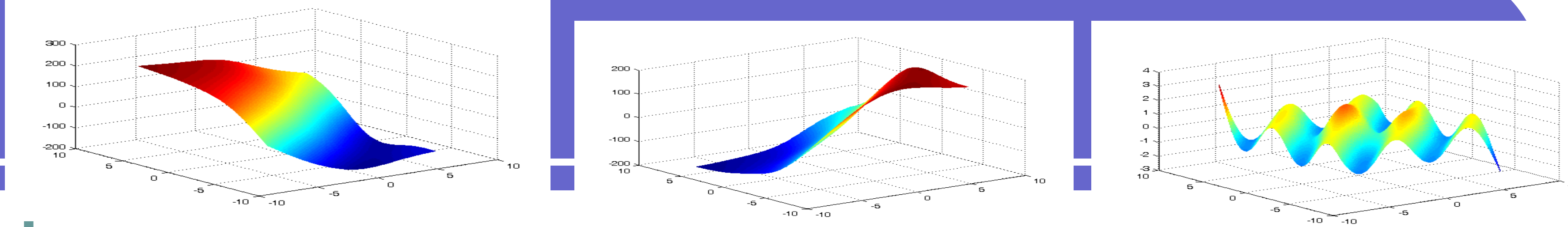

NRBF(6) by annealed FE


NRBF(9) by annealed FE

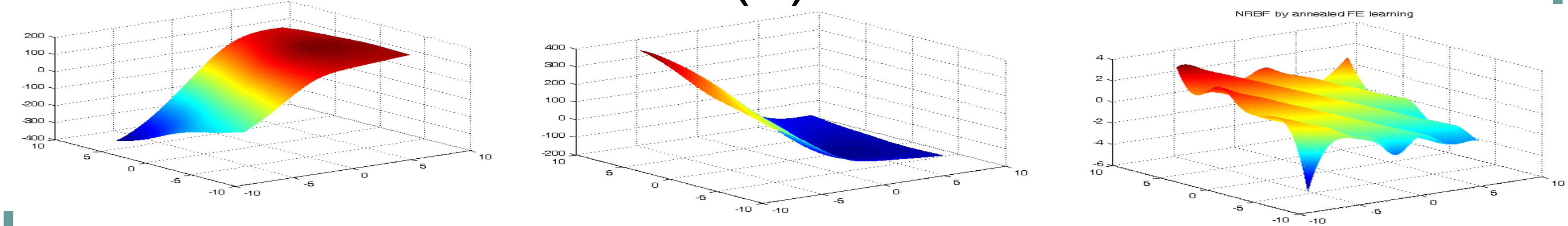

NRBF(12) by annealed FE


NRBF(15) by annealed FE

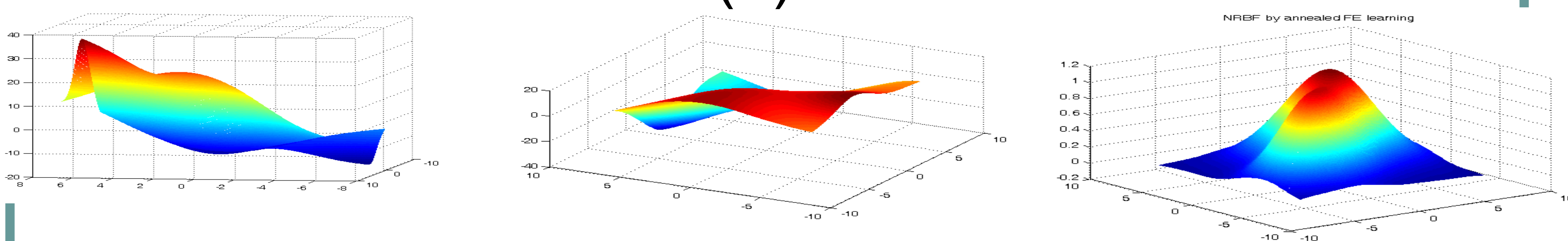

NRBF(18) by annealed FE



(a)



(b)



(c)

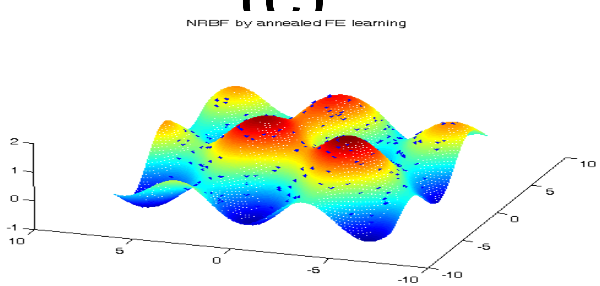
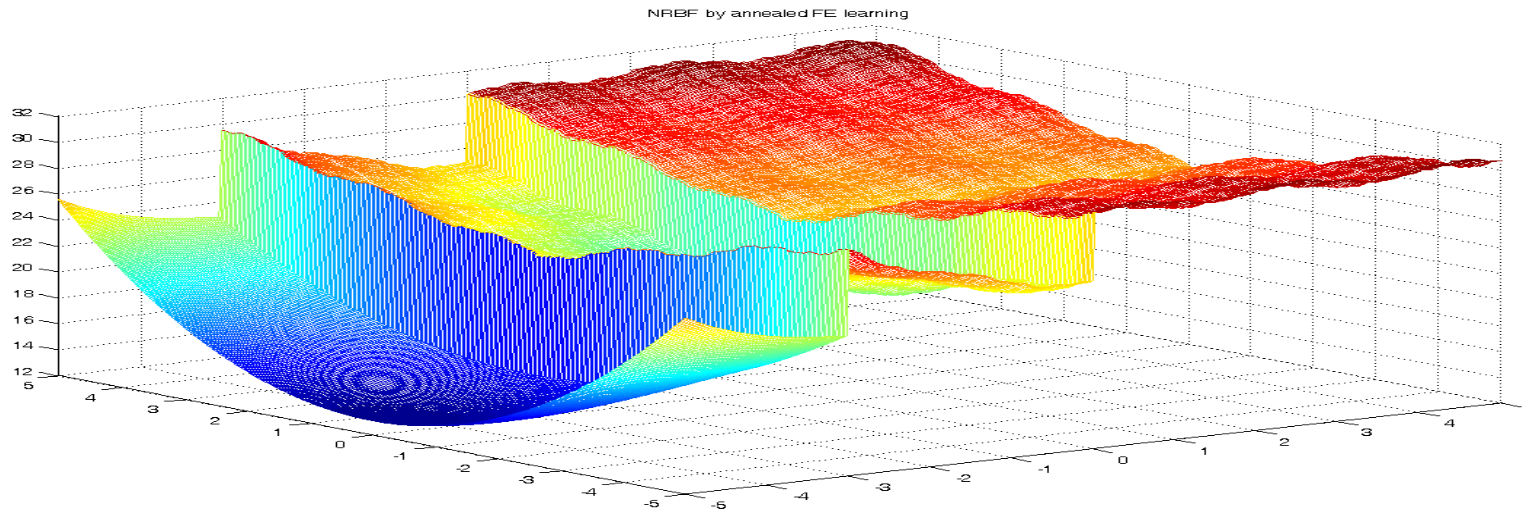
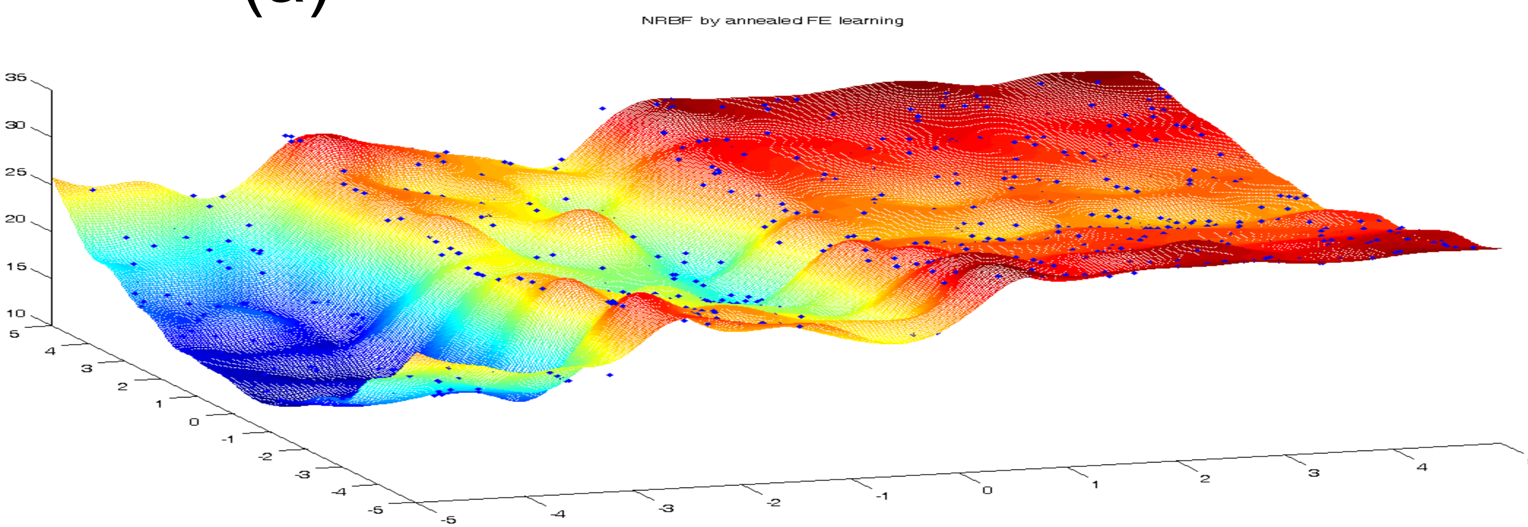


Figure 6

(d)



(a)



(b)

Figure 8

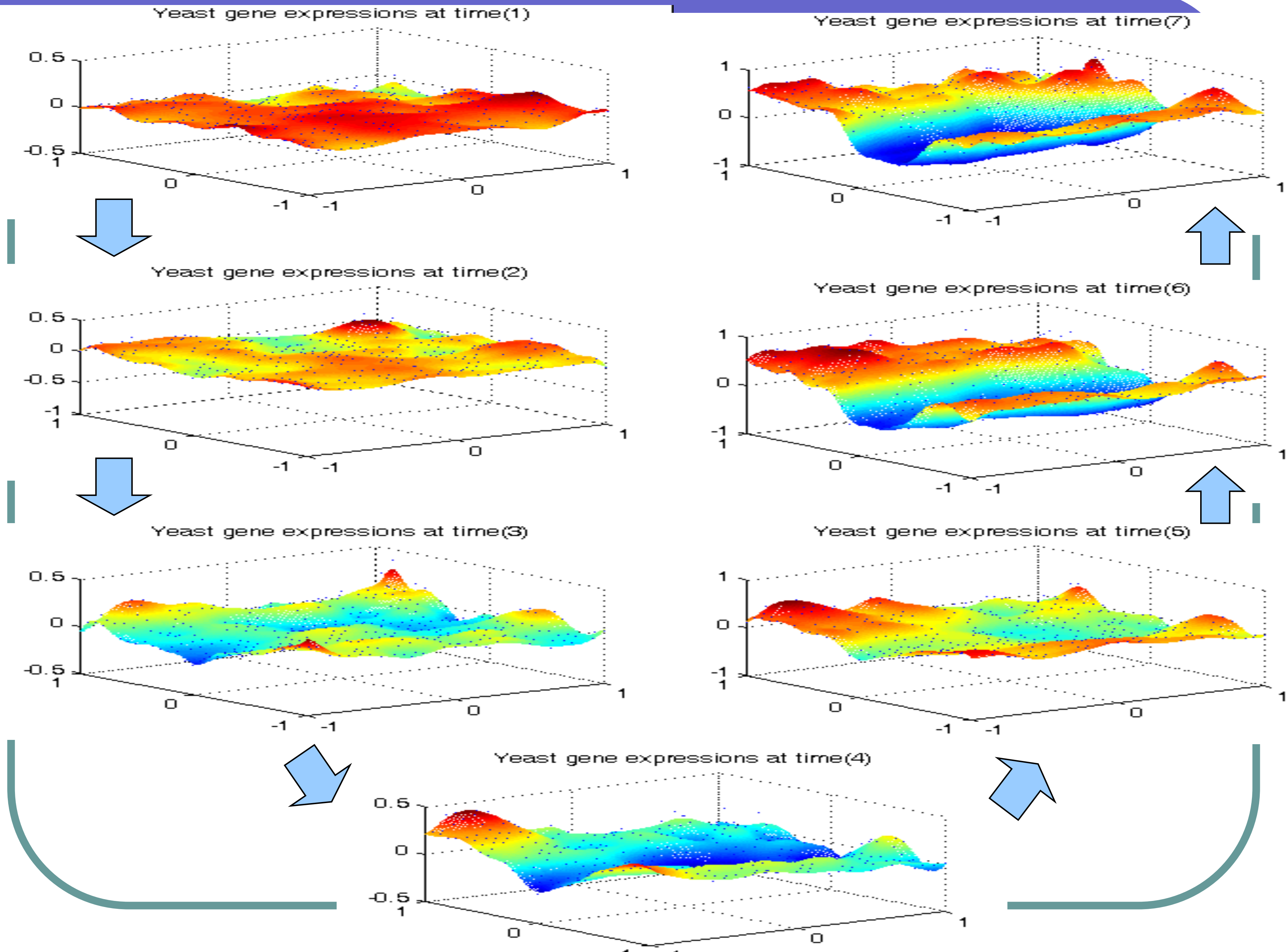

Yeast gene expressions at time(1)
Yeast gene expressions at time(2)
Yeast gene expressions at time(3)
Yeast gene expressions at time(4)
Yeast gene expressions at time(5)
Yeast gene expressions at time(6)
Yeast gene expressions at time(7)

Figure 9

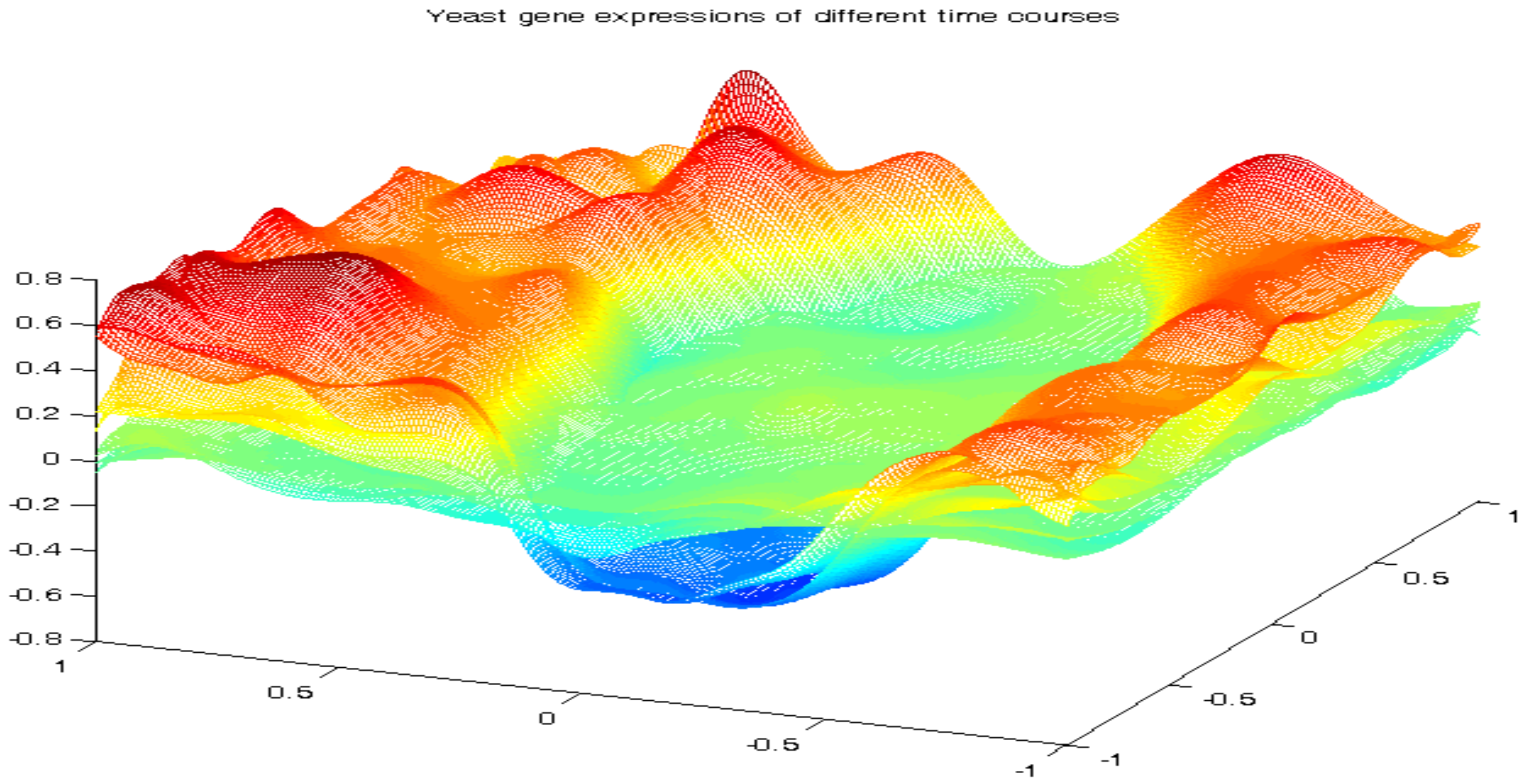
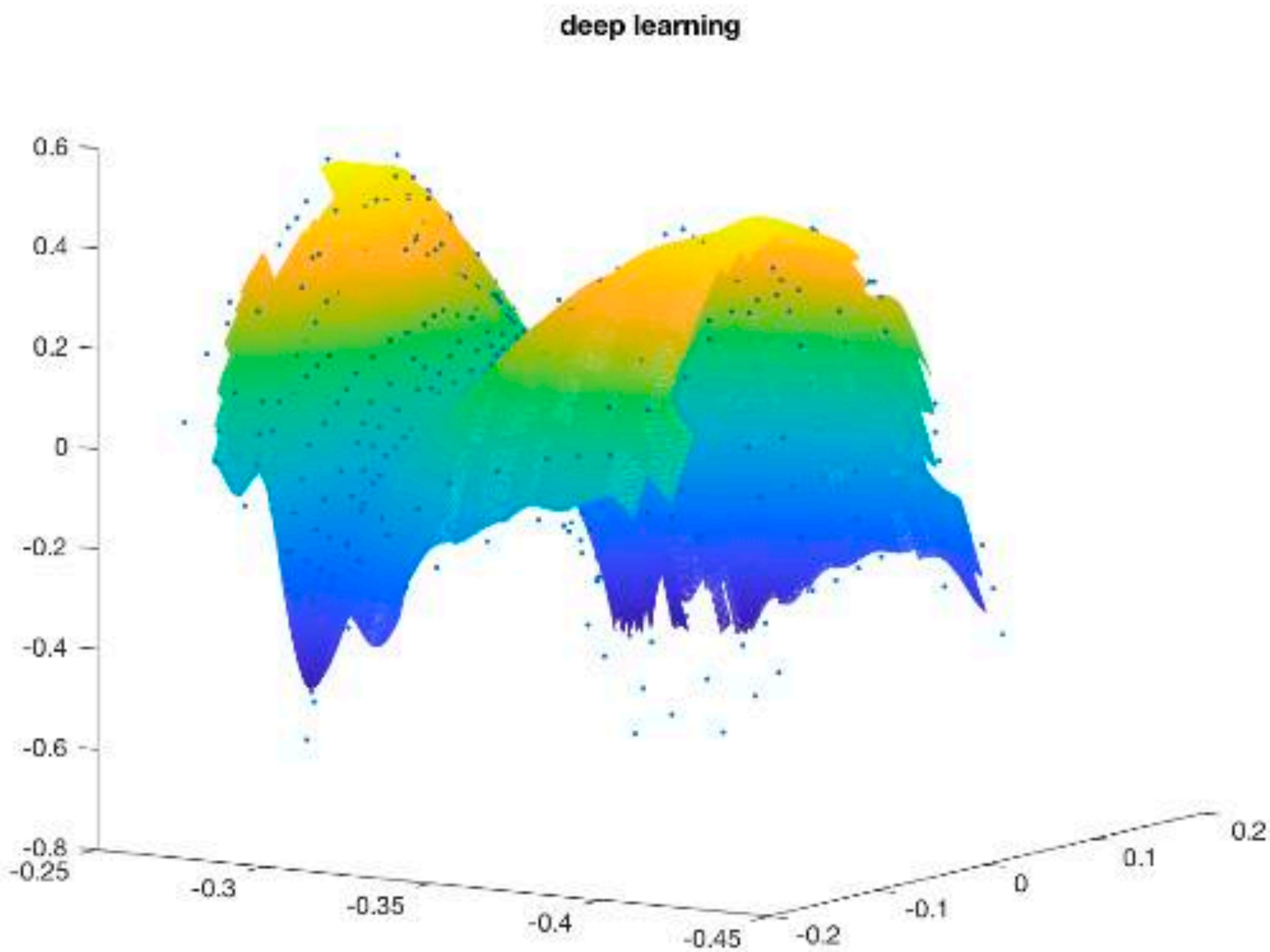
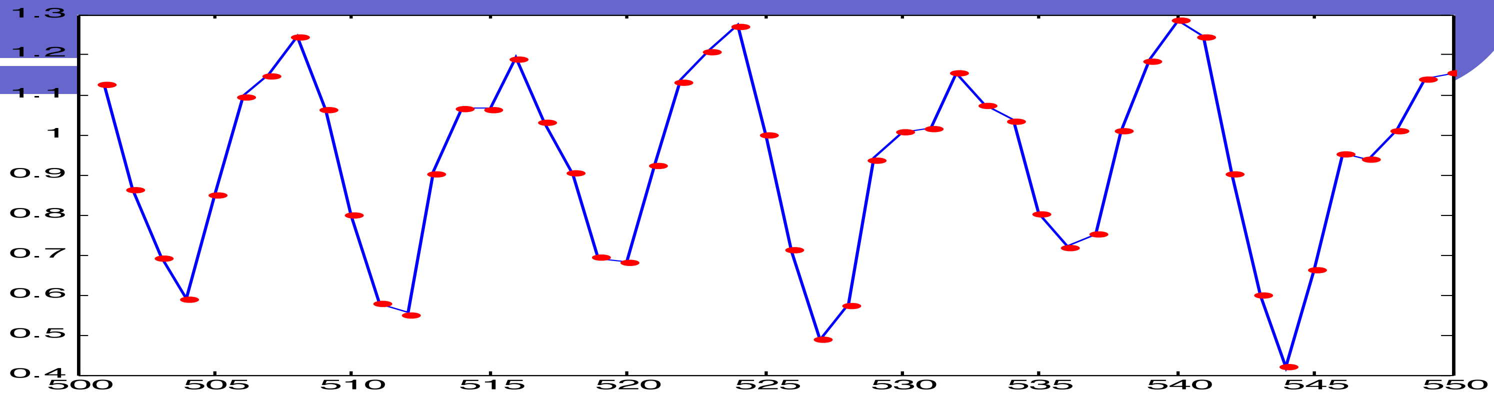

Yeast gene expressions of different time courses
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1

Figure 10



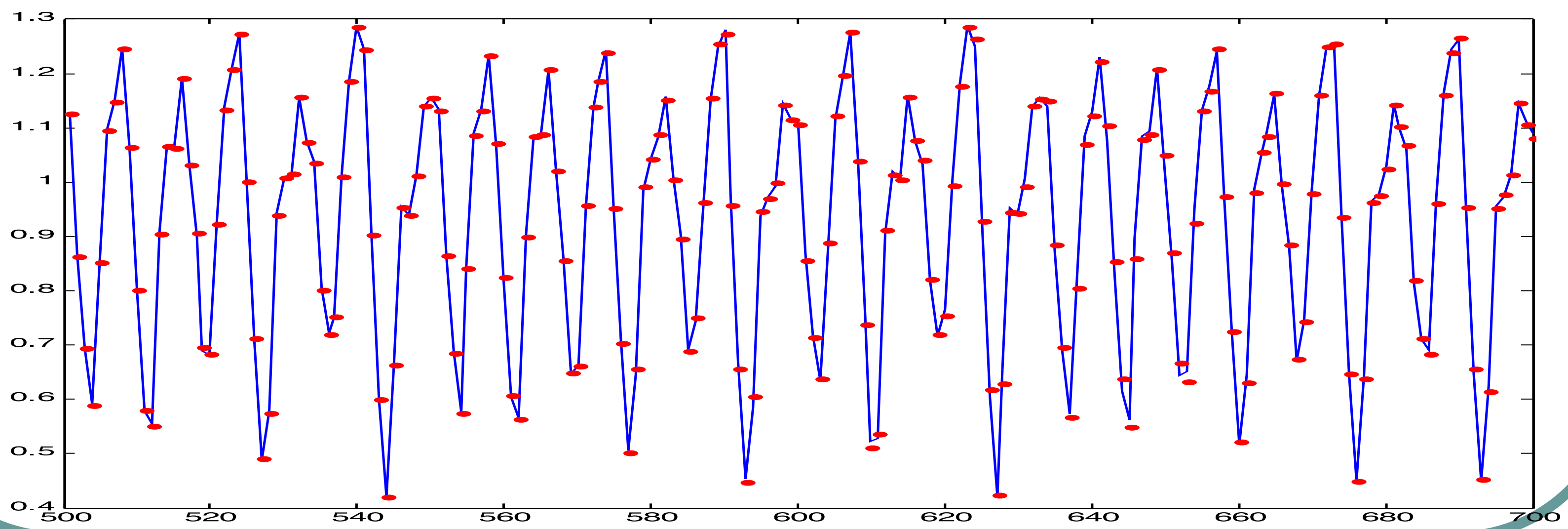
deep learning
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.25
-0.3
-0.35
-0.4
-0.45
-0.2
-0.1
0
0.1
0.2

50-step-look-ahead long term predictions of Mackey-Glass 17 data



(a)

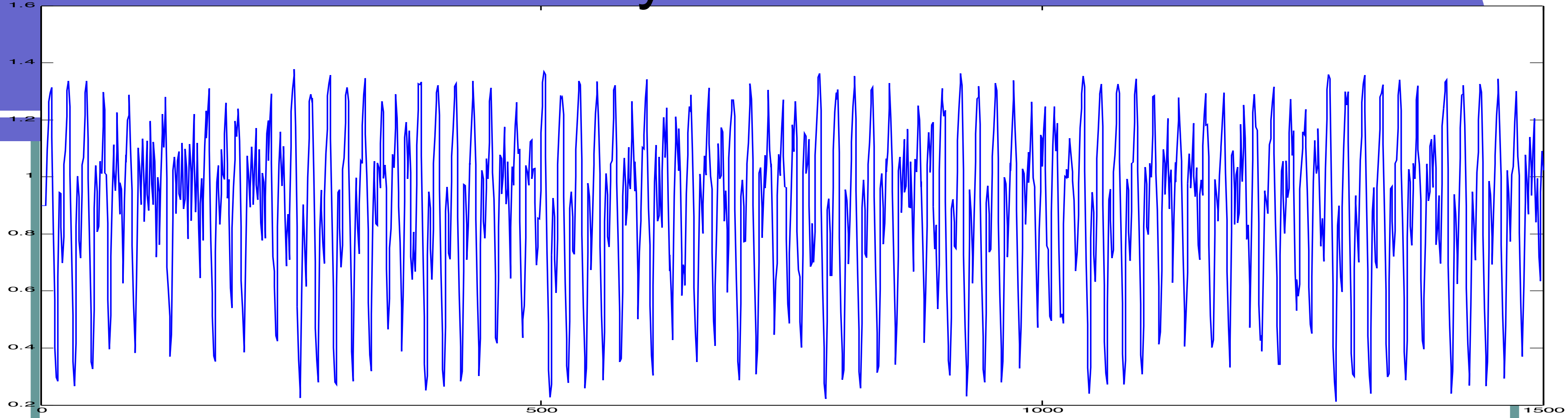
200-step-look-ahead long term predictions of Mackey-Glass 17 data



(b)

Figure 11

Mackey-Glass 30 data



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

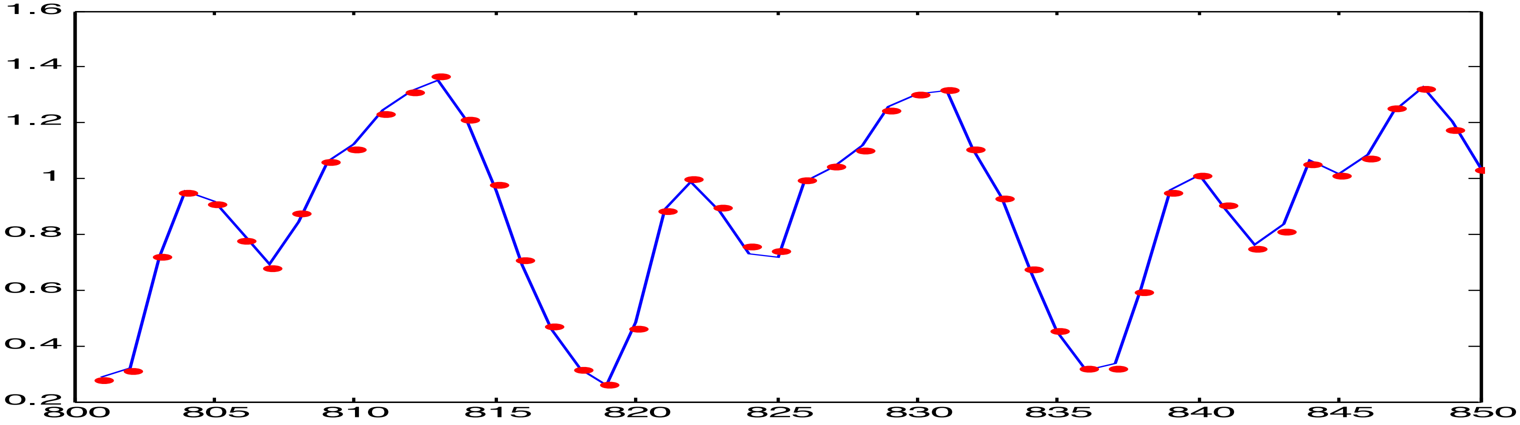
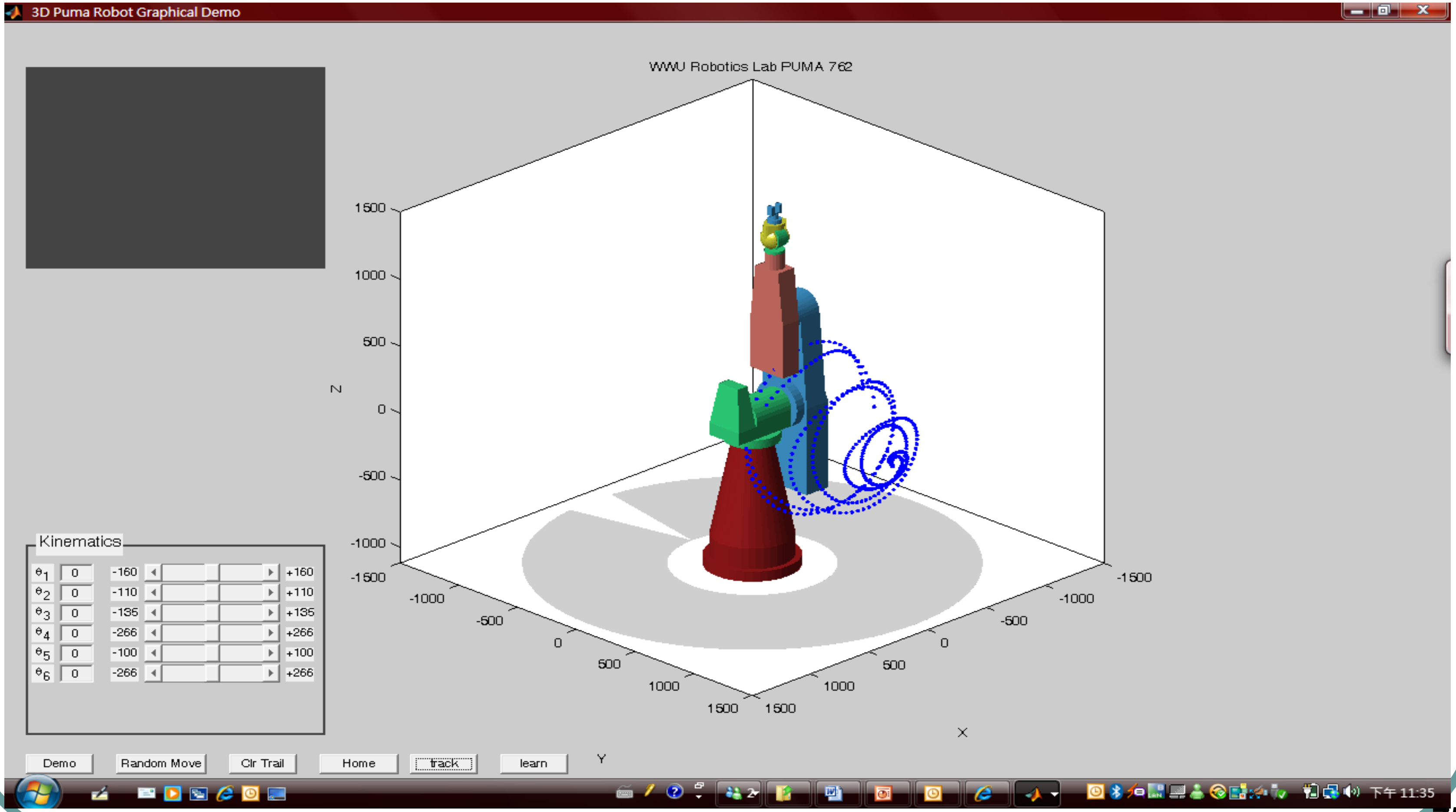
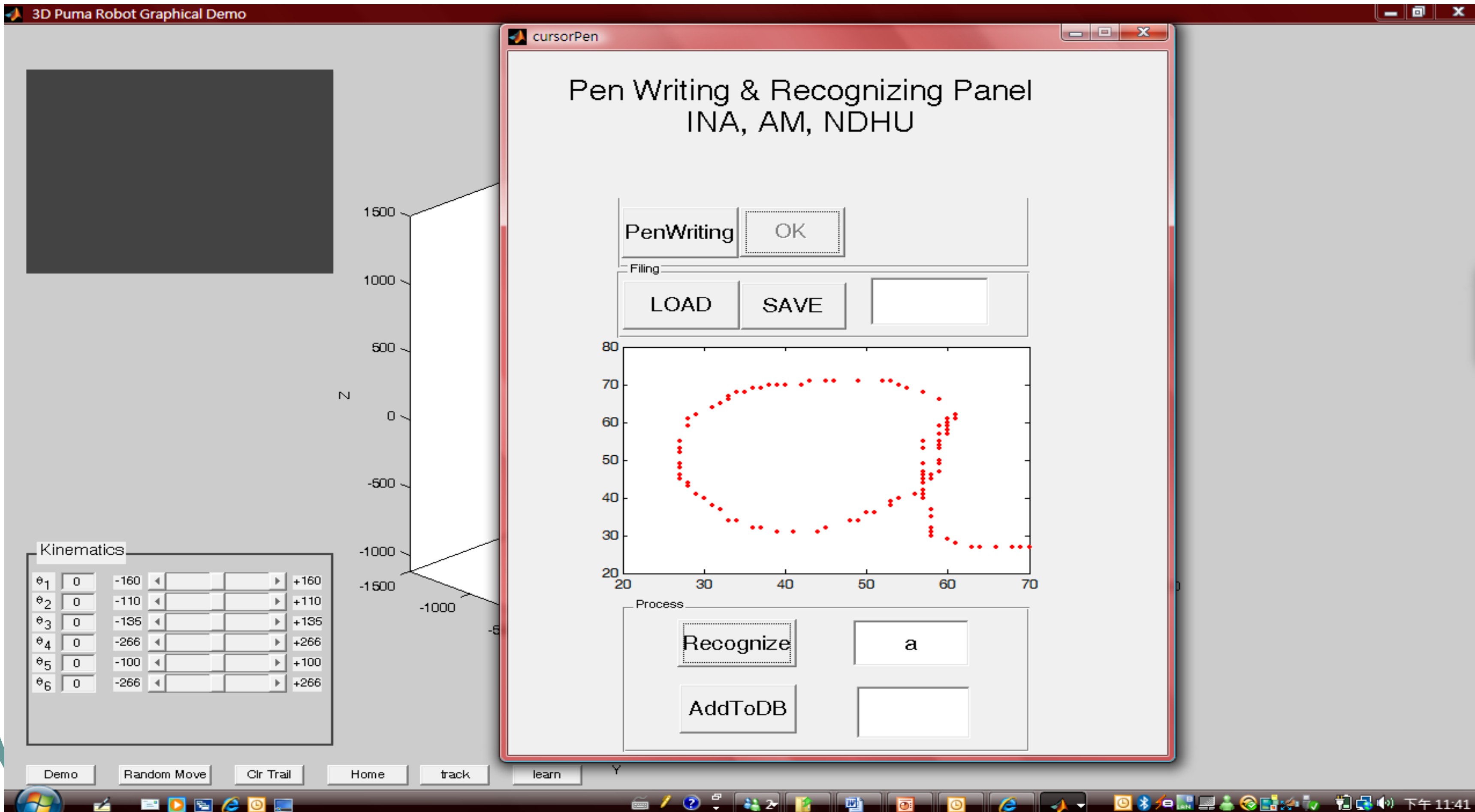


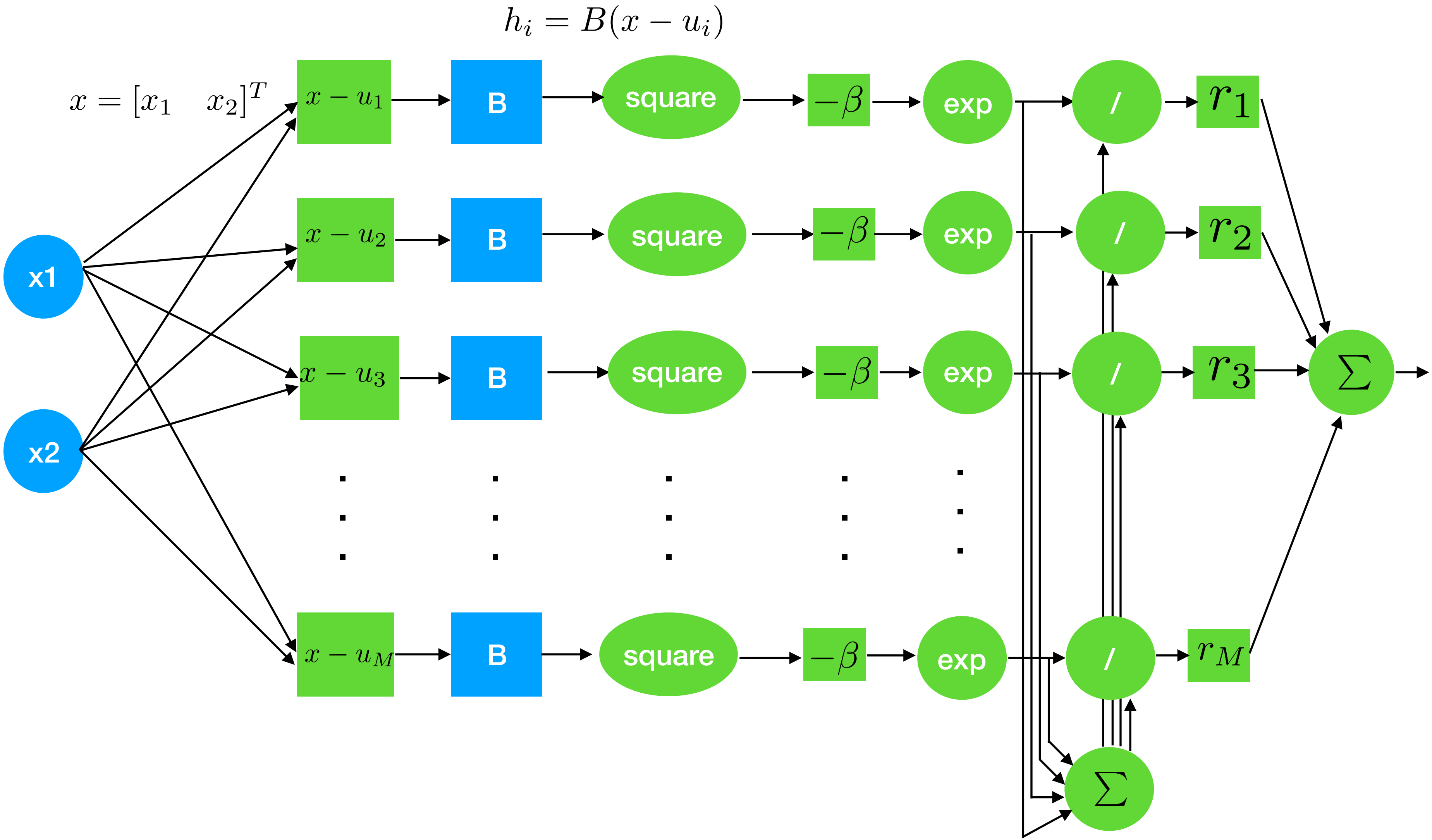
Figure 12

Robot arm control

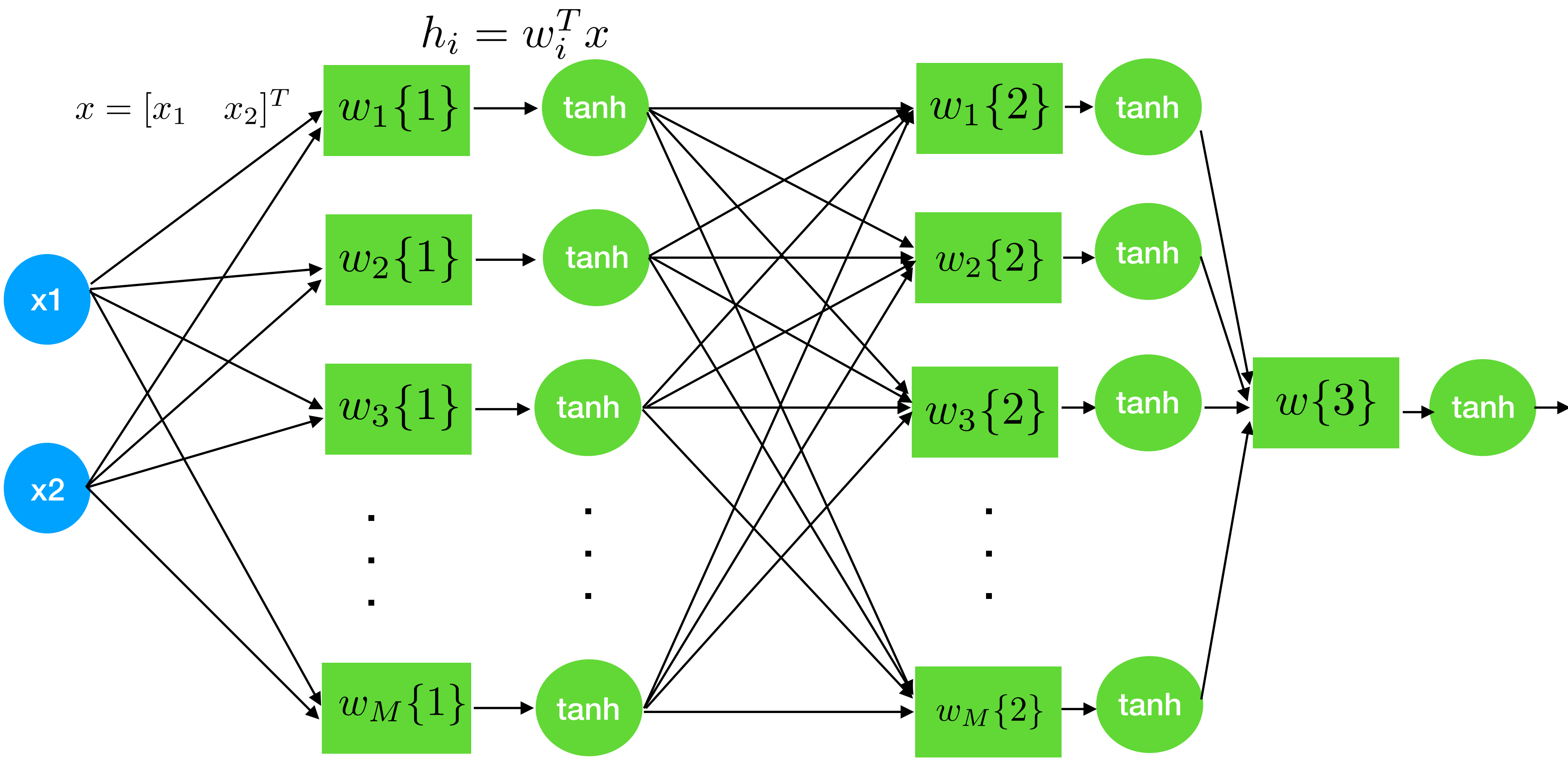


Pen writing recognition

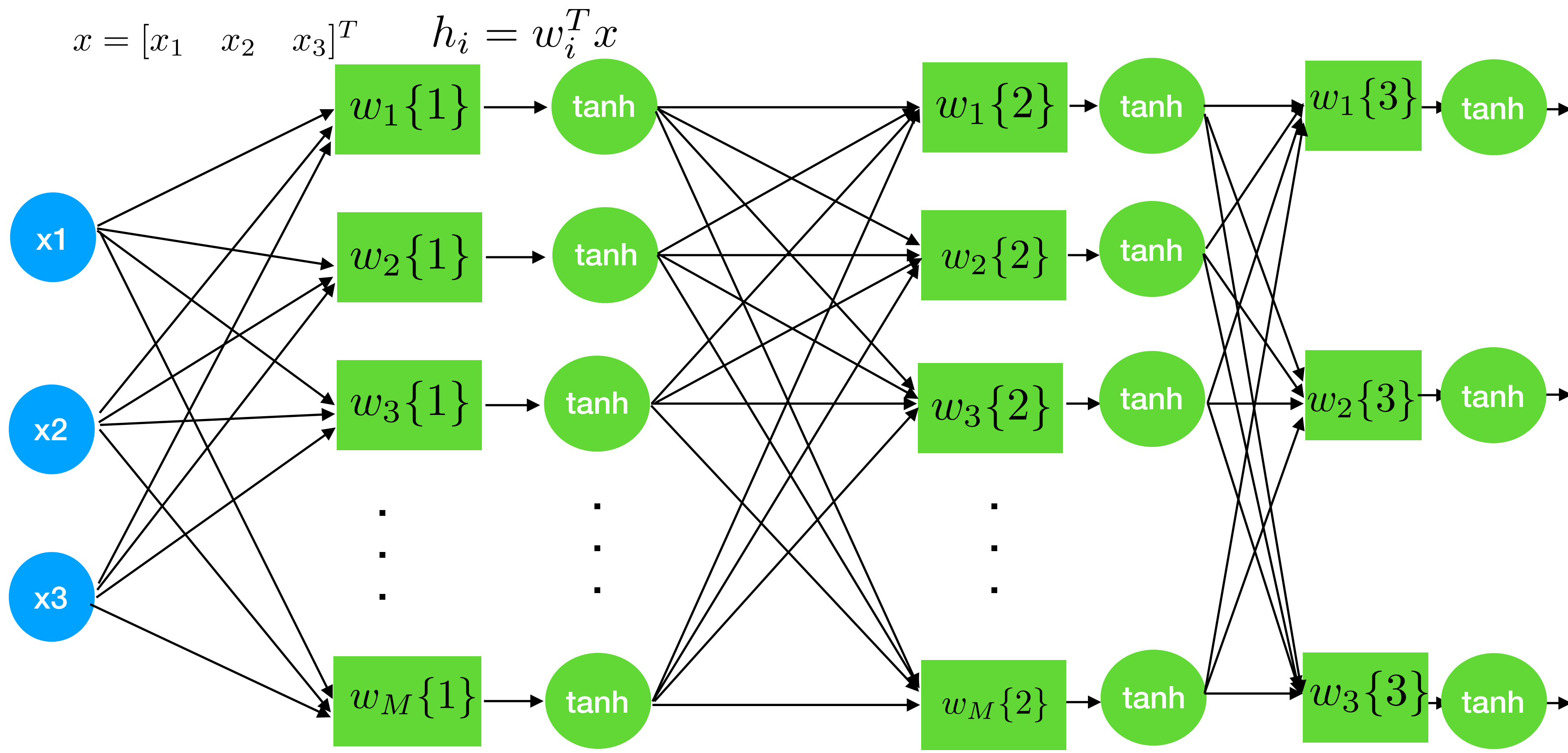




$h_i = B(x - u_i)$
$x = [x_1 \quad x_2]^T$
x1
x2
$x - u_1$
$x - u_2$
$x - u_3$
$x - u_M$
B
square
$-\beta$
exp
/
r_1
r_2
r_3
r_M
$\sum$

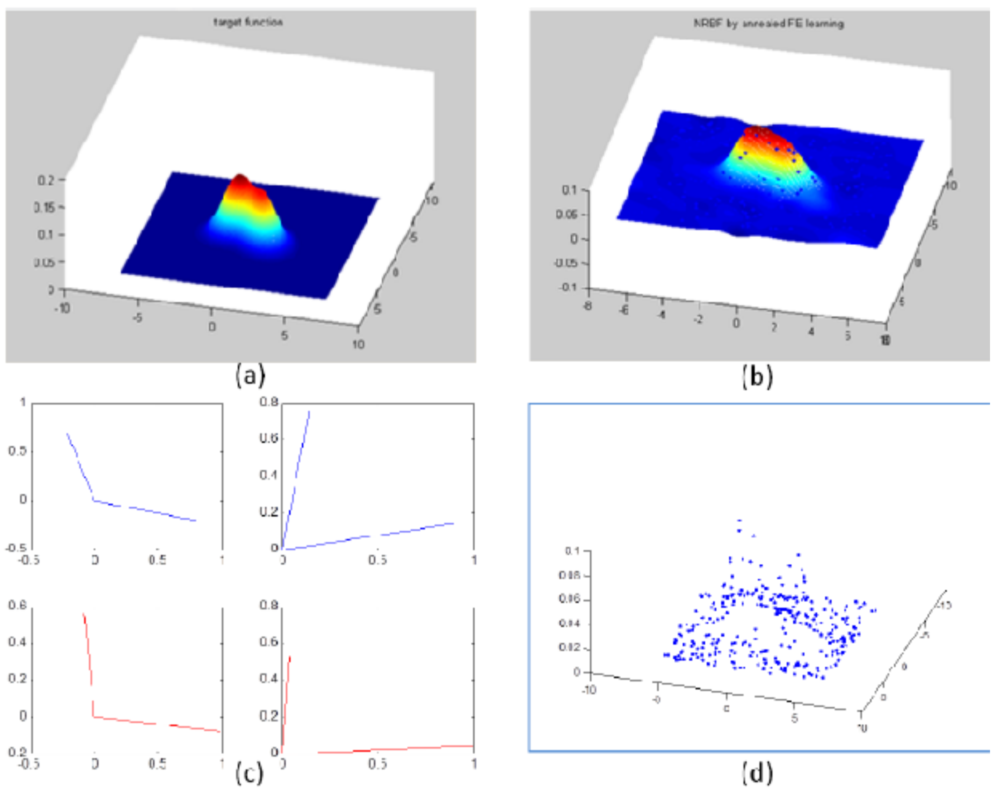


$h_i = w_i^T x$
$x = [x_1 \quad x_2]^T$
x1
x2
$w_1\{1\}$
$w_2\{1\}$
$w_3\{1\}$
$w_M\{1\}$
tanh
$w_1\{2\}$
$w_2\{2\}$
$w_3\{2\}$
$w_M\{2\}$
$w\{3\}$



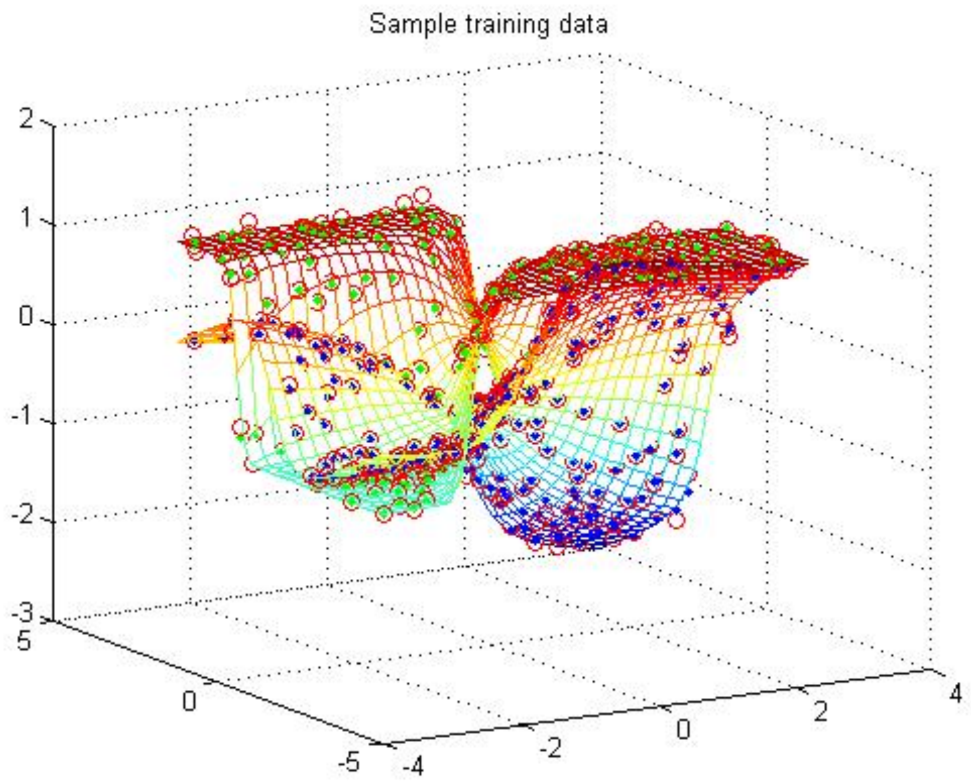
$x = [x_1 \quad x_2 \quad x_3]^T$
$h_i = w_i^T x$
x1
x2
x3
$w_1\{1\}$
$w_2\{1\}$
$w_3\{1\}$
$w_M\{1\}$
tanh
$w_1\{2\}$
$w_2\{2\}$
$w_3\{2\}$
$w_M\{2\}$
$w_1\{3\}$
$w_2\{3\}$
$w_3\{3\}$

Covariance Matrix Analysis

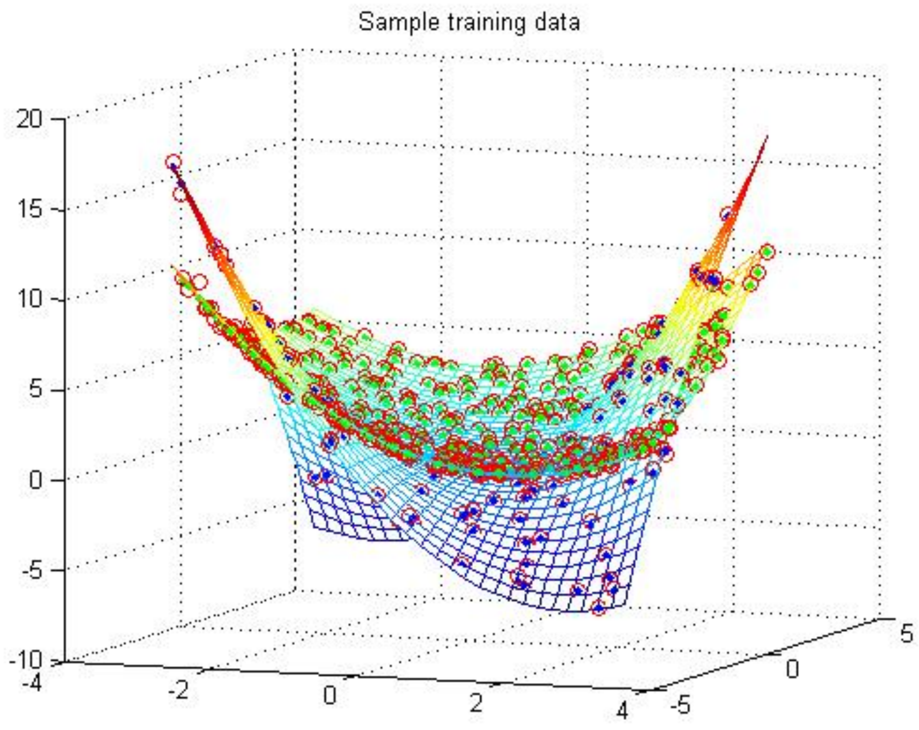


Function approximation v.s One-to-many mapping

	function approximation	one-to-many mapping
type	an RBF network	multiple high-order RBF networks
approximation	one-to-one mapping	one-to-many mapping
order	single-order posterior interconnections	high-order posterior interconnections
learning	supervised learning	supervised learning
data type	paired data without weights	paired data with weights
control system	forward kinematics	inverse kinematics
modular type	single module	multiple modules
objective function	mean square error	weighted square error



- P=2,k=25
- 2 tanh function
- mse=0.0022



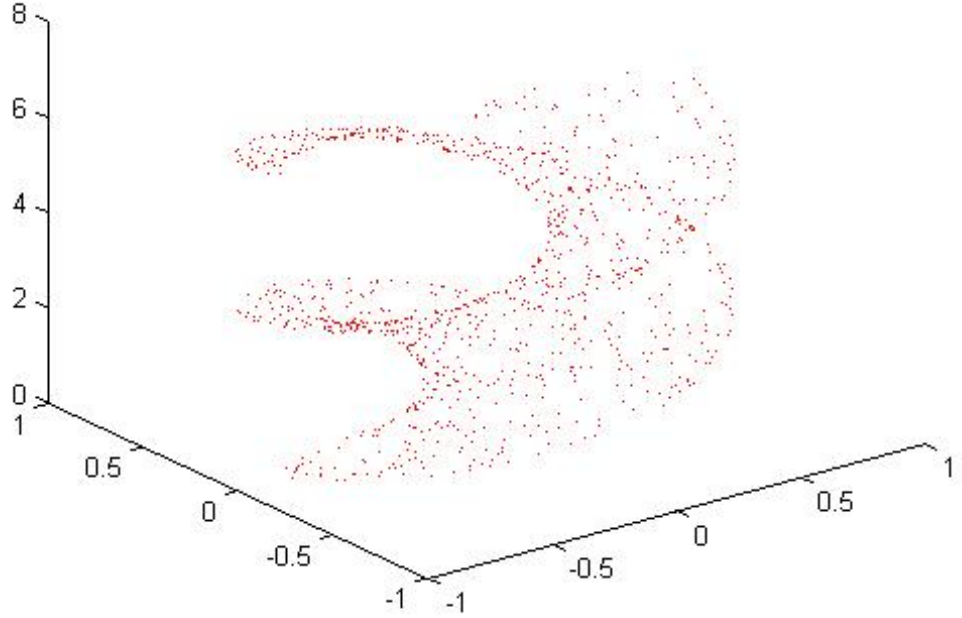
- P=2,k=25
- 2quadratic function
- Mse=0.0089

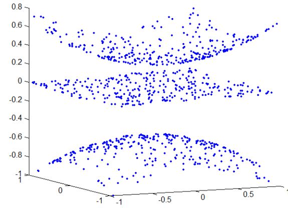
One-to-many function approximation to g_1

- P=5;k=M=30;N=1000

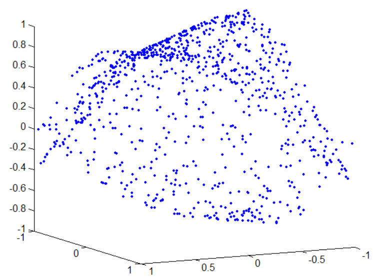
$$a_1 = g_1(q_1, q_2)$$

$$a_1 \in (0, 2\pi)$$

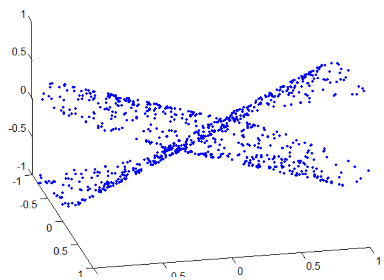




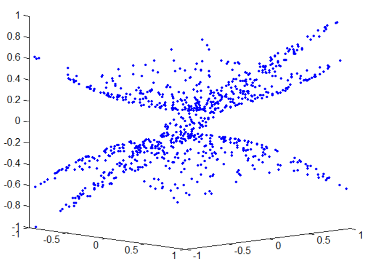
$$f_1(x_1,x_2) = x_1^2 + x_2^2 + 1$$
$$f_2(x_1,x_2) = -x_1^2 - x_2^2 - 2$$
$$f_3(x_1,x_2) = 0.1x_1 + 0.1x_2$$



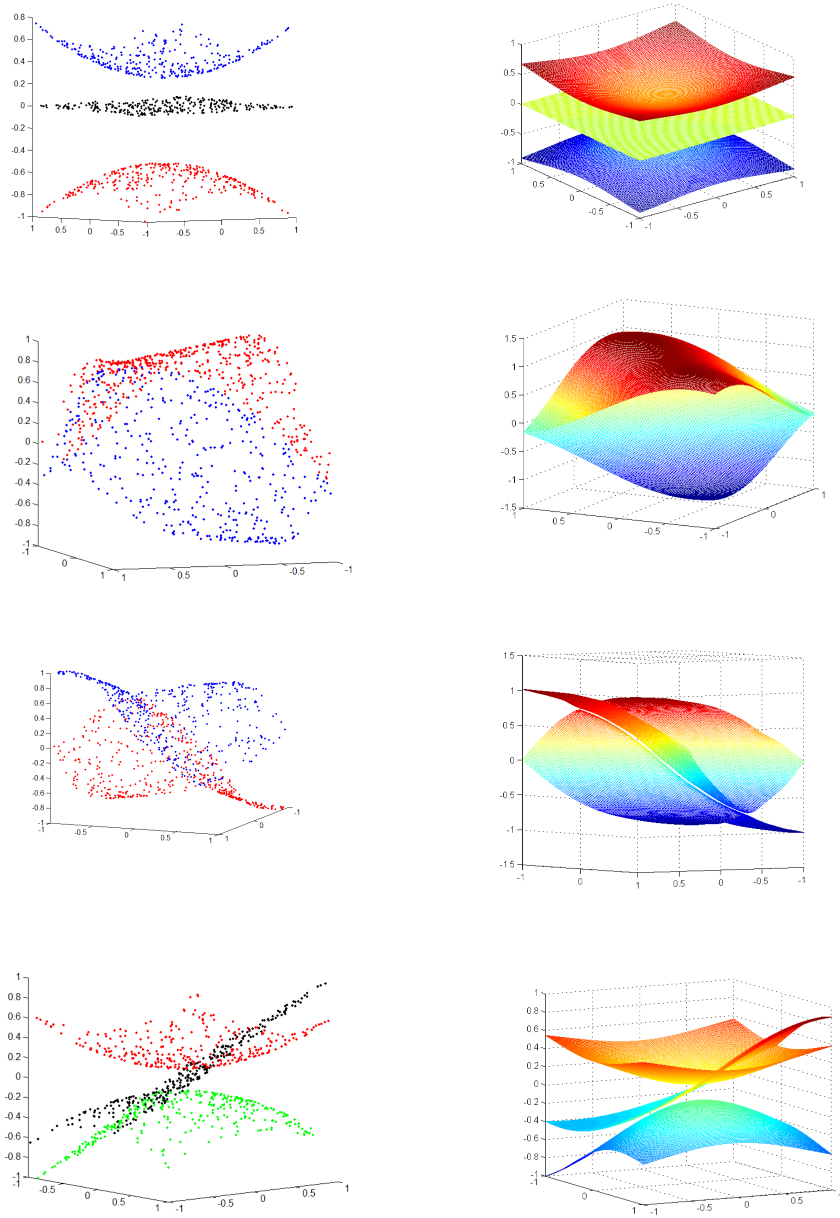
$$f_1(x_1,x_2) = \sin(x_1 + x_2)$$
$$f_2(x_1,x_2) = \cos(x_1 - x_2)$$



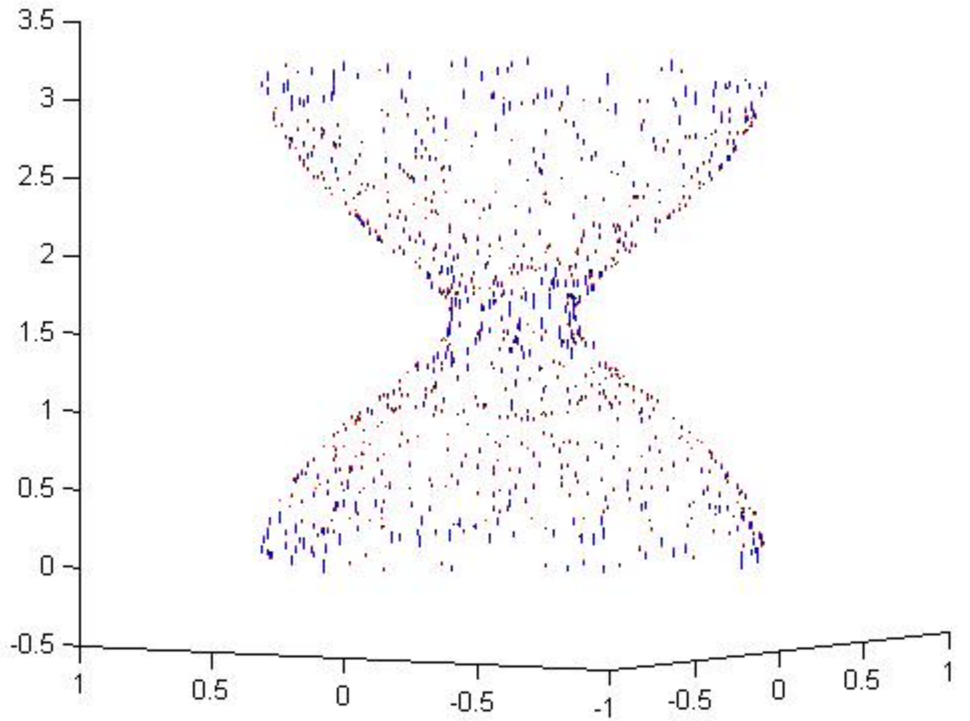
$$f_1(x_1,x_2) = \tanh(x_1 + x_2)$$
$$f_2(x_1,x_2) = \tanh(x_1 - x_2)$$

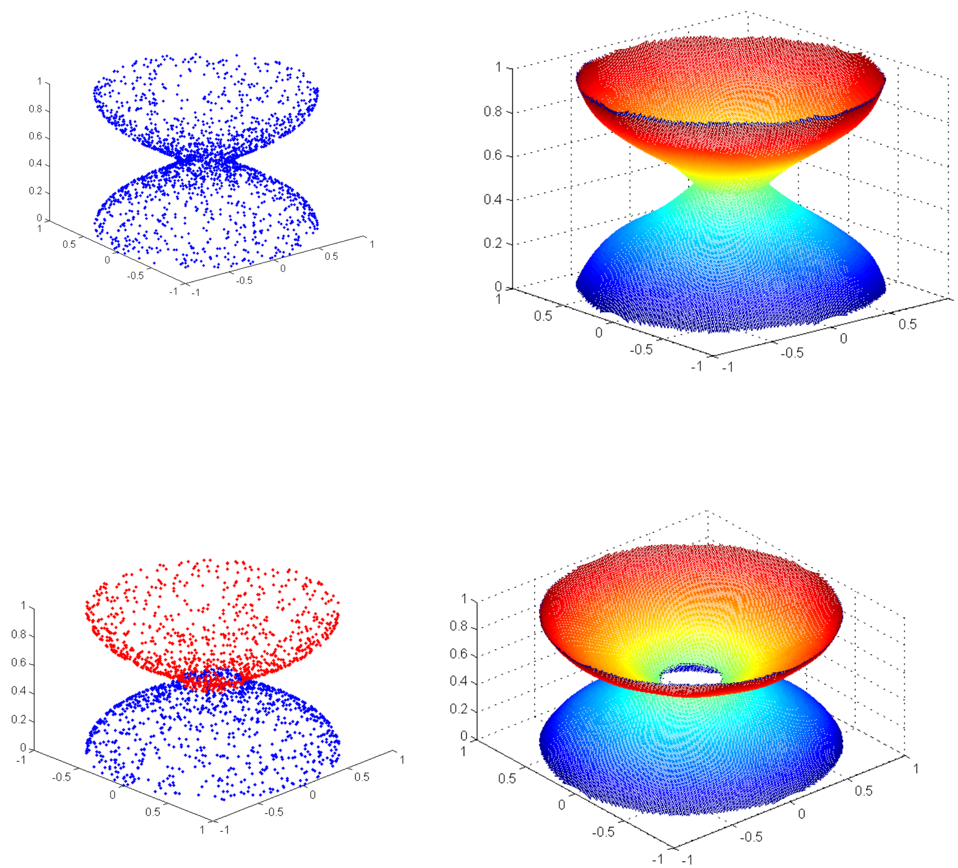


$$f_1(x_1,x_2) = x_1^2 + x_2^2 + 0.5$$
$$f_2(x_1,x_2) = -x_1^2 - x_2^2 - 0.5$$
$$f_3(x_1,x_2) = 2x_1 + 2x_2$$

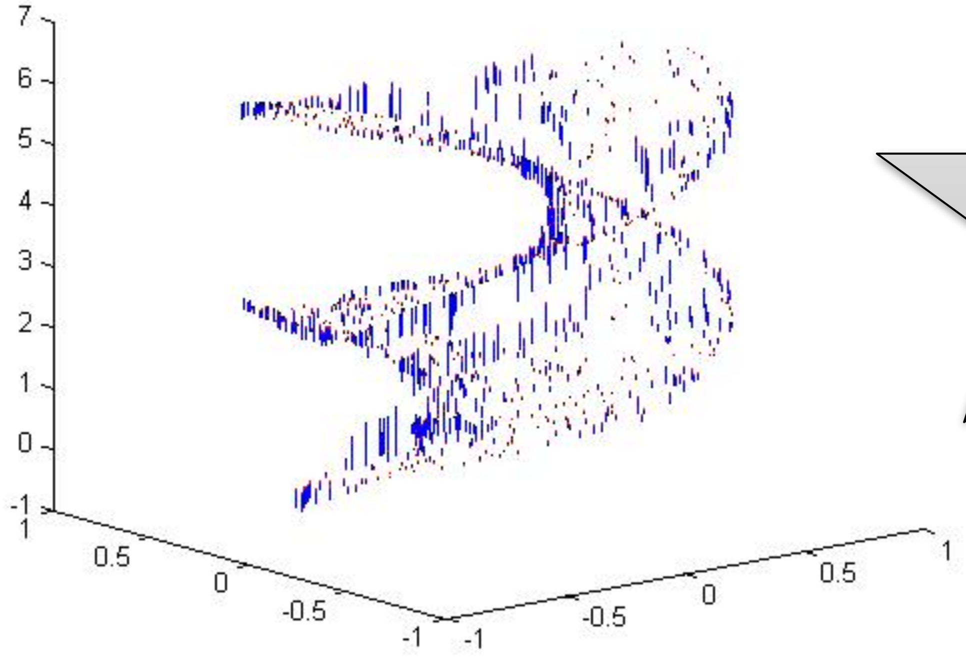


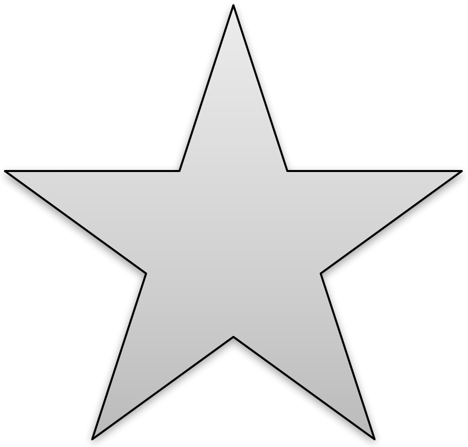
Approximating functions

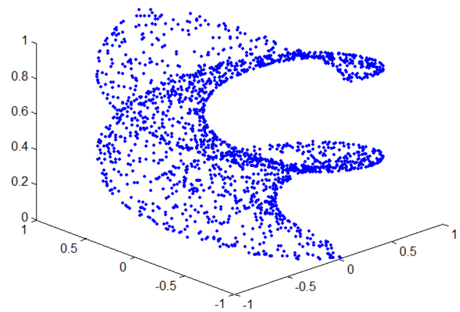


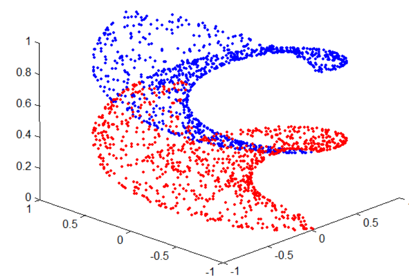


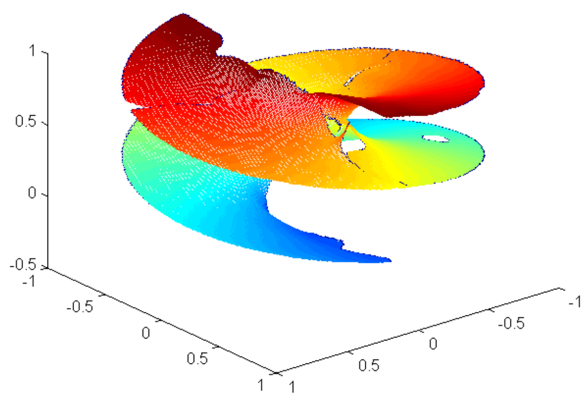
Approximating functions

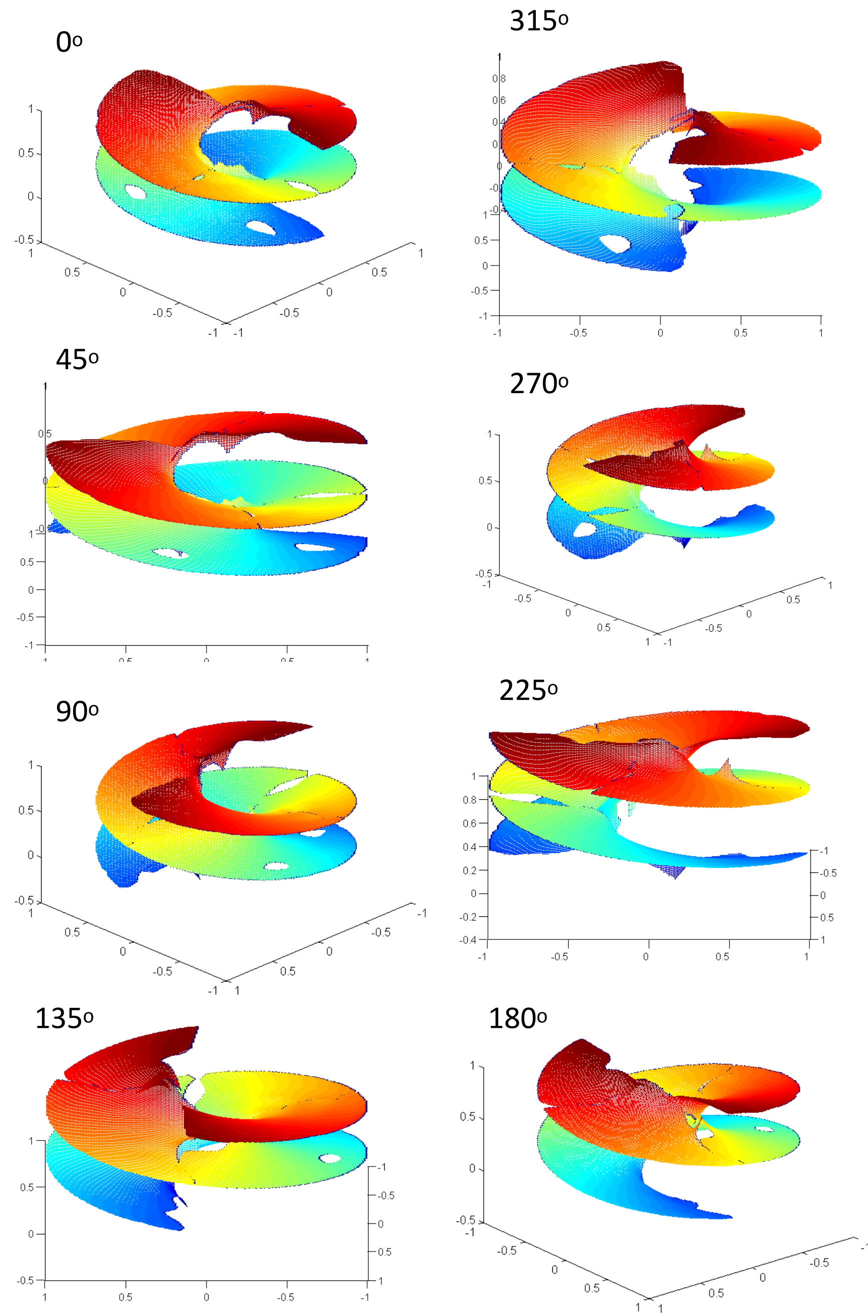


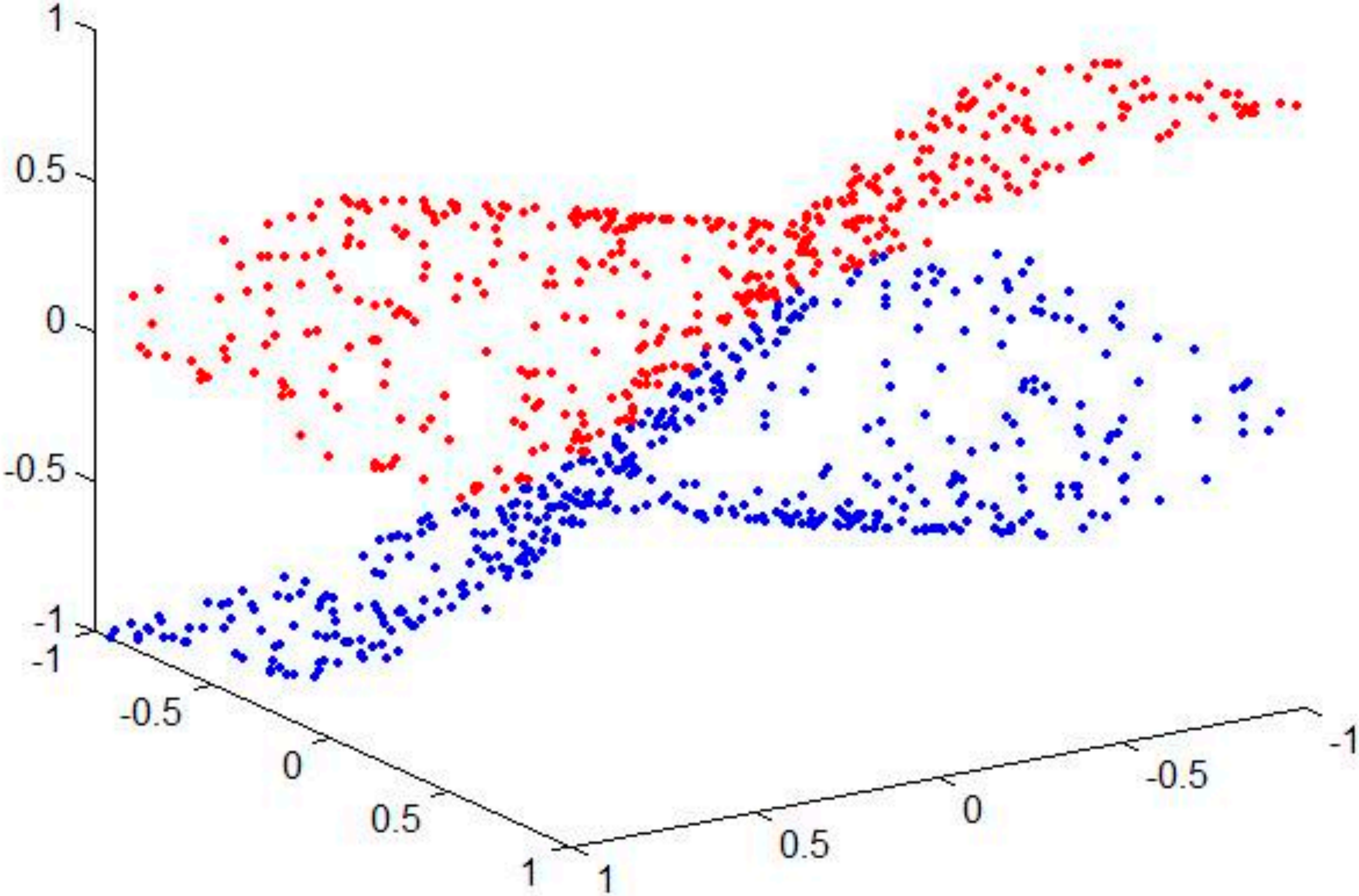



1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

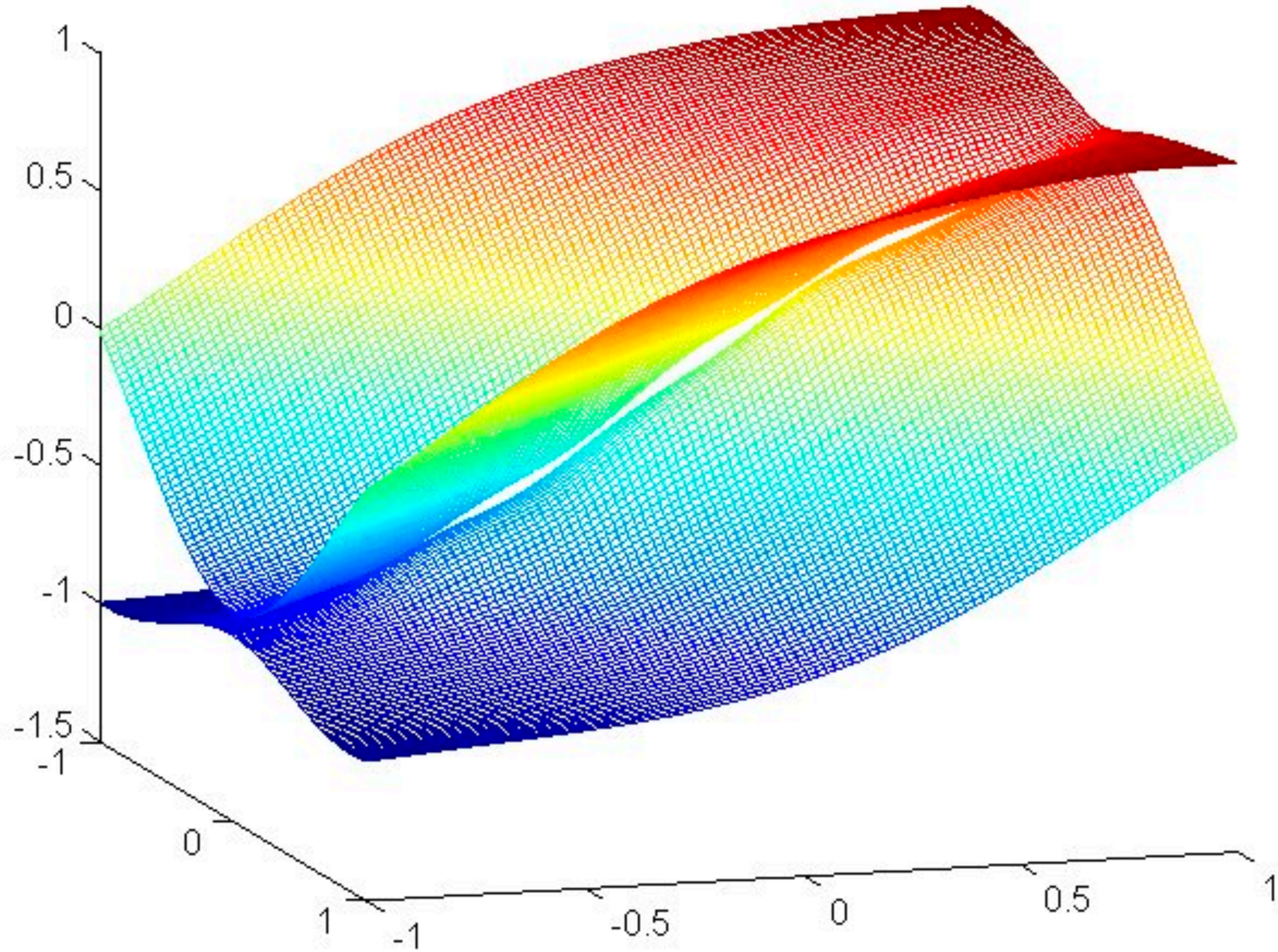

1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1


1
0.5
0
-0.5
-1
-0.5
0
0.5
1
1
0.5
0
-0.5
-1

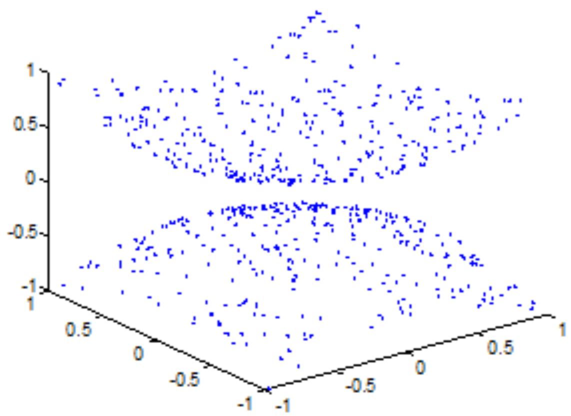
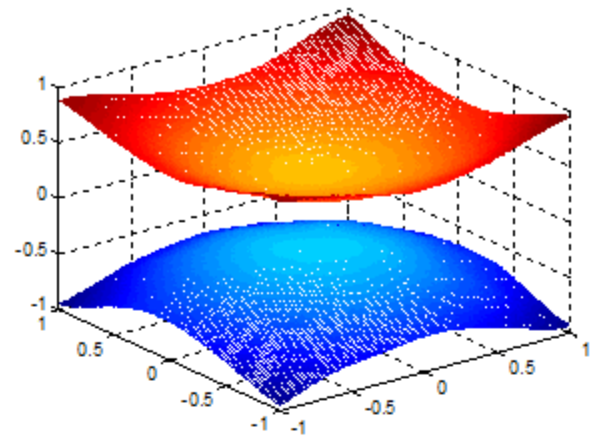

0o
315o
45o
270o
90o
225o
135o
180o



1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1
0.5
0
-0.5
-1

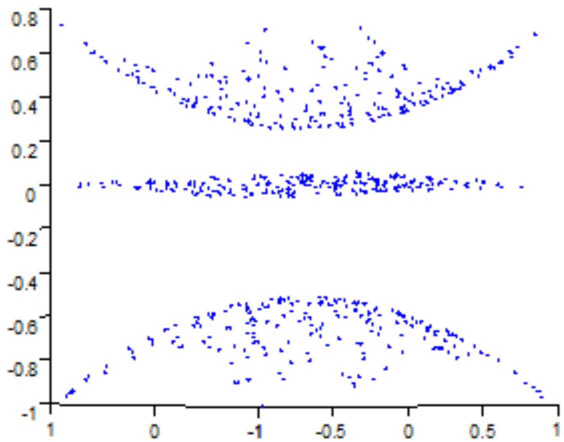
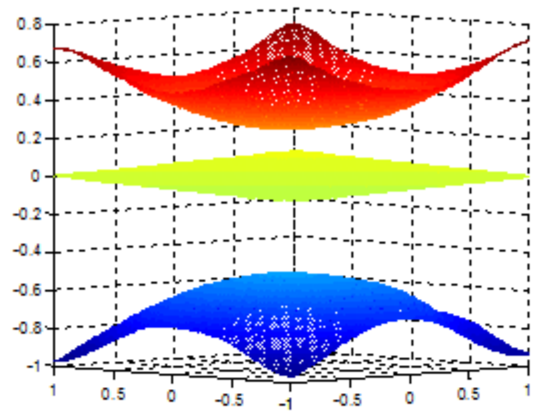

1
0.5
0
-0.5
-1
-1.5
-1
0
1
-1
-0.5
0
0.5
1

2-type deterministic transition

Original data	deterministic rule
	

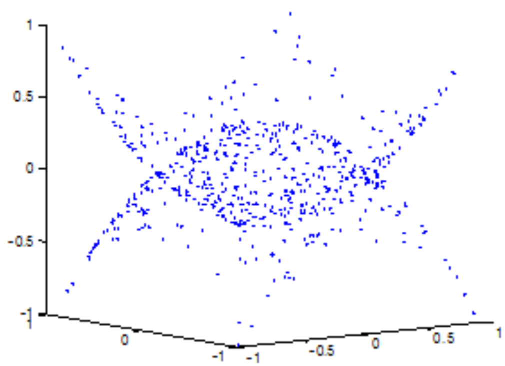
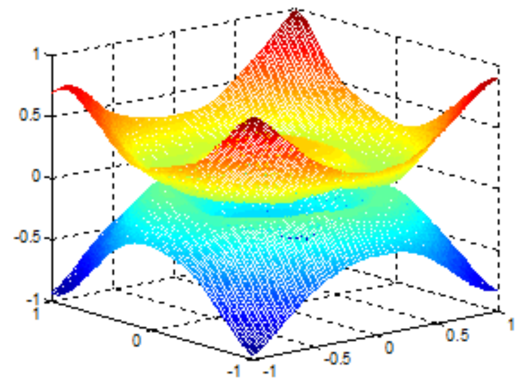
	target prediction (learning)	state inference	target prediction (aenn)
error	0.000257	0.000758	0.000281

3-type deterministic transition

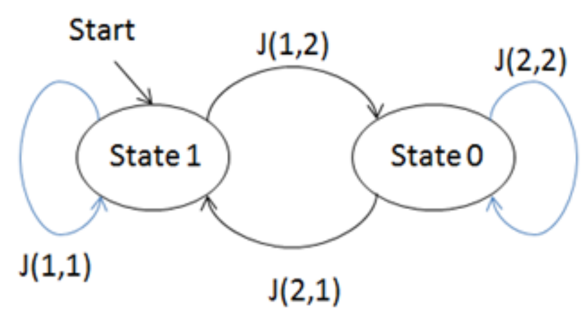
<table>
<tr><th>Original data</th><th>deterministic rule</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

	target prediction (learning)	state inference	target prediction (aenn)
error	0.000211	0.000911	0.000230

2-type Stochastic transition

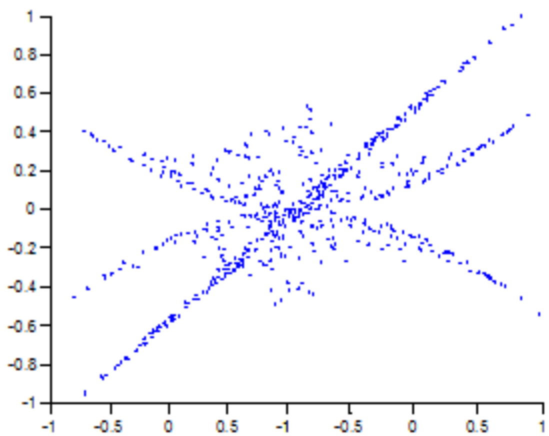
<table>
<tr><td>Original data</td><td>Stochastic rule</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Probability</td><td>Stochastic automata state transition</td></tr>
<tr><td>

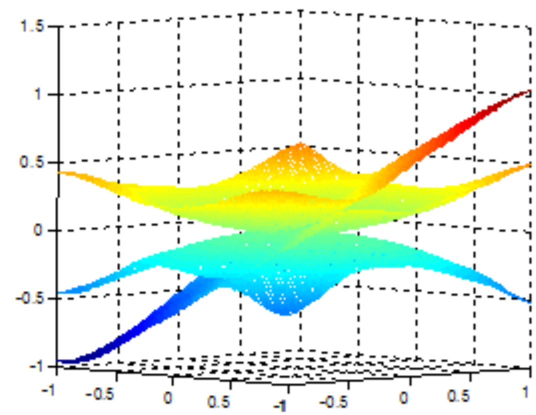
$$J = \begin{matrix} 0.4045 & 0.5955 \\ 0.6345 & 0.3655 \end{matrix}$$

</td><td>
</td></tr>
</table>

3-type Stochastic transition

Original data	Stochastic rule

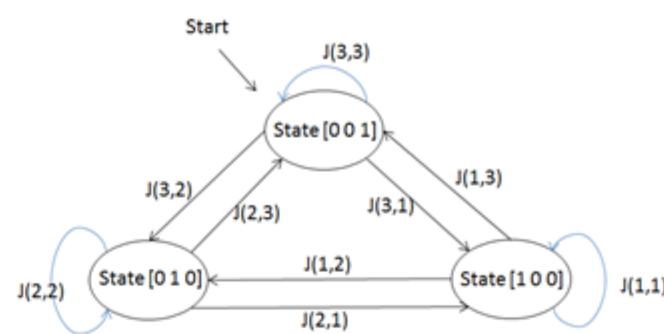




Probability	Stochastic automata state transition

J =

$$\begin{matrix} 0.1181 & 0.7014 & 0.1806 \\ 0.2925 & 0.7075 & 0 \\ 0.0645 & 0.7742 & 0.1613 \end{matrix}$$



Differential function approximation

- Neural Networks

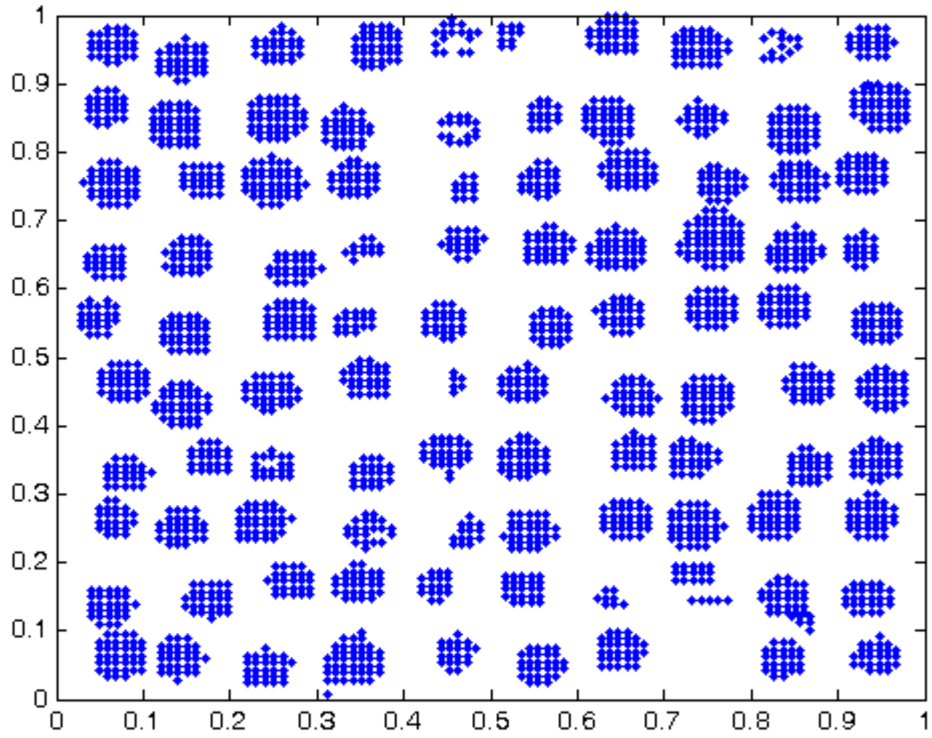
$$v(t)=\frac{dx}{dt}$$

$$=F(v(t),x(t))$$

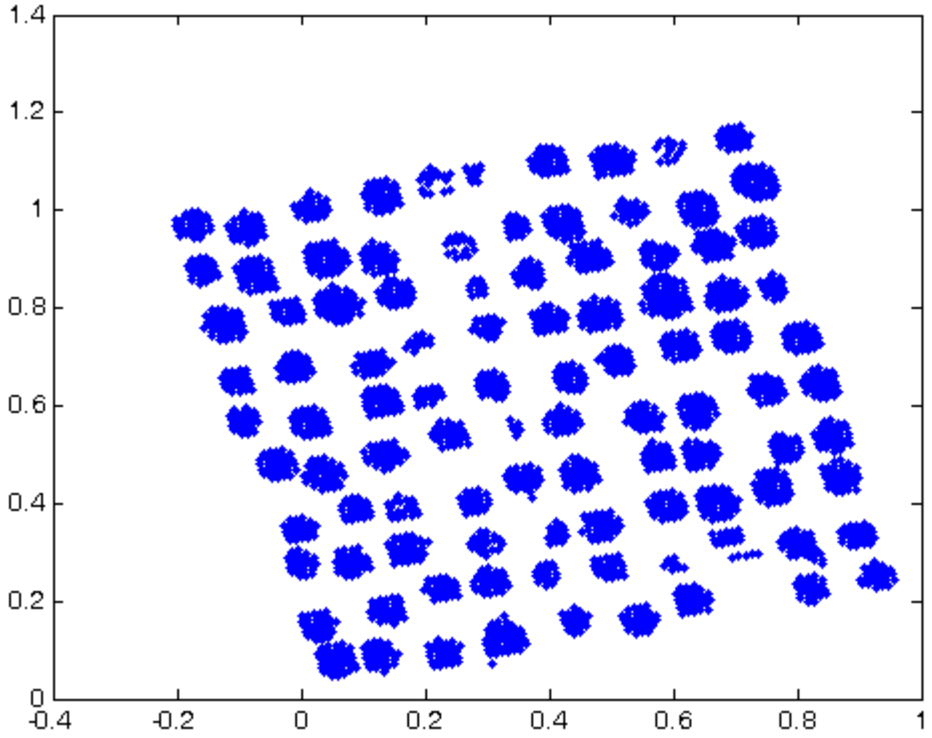
Learning neural networks to approximate differential equations

Clustering analysis

Find data clusters

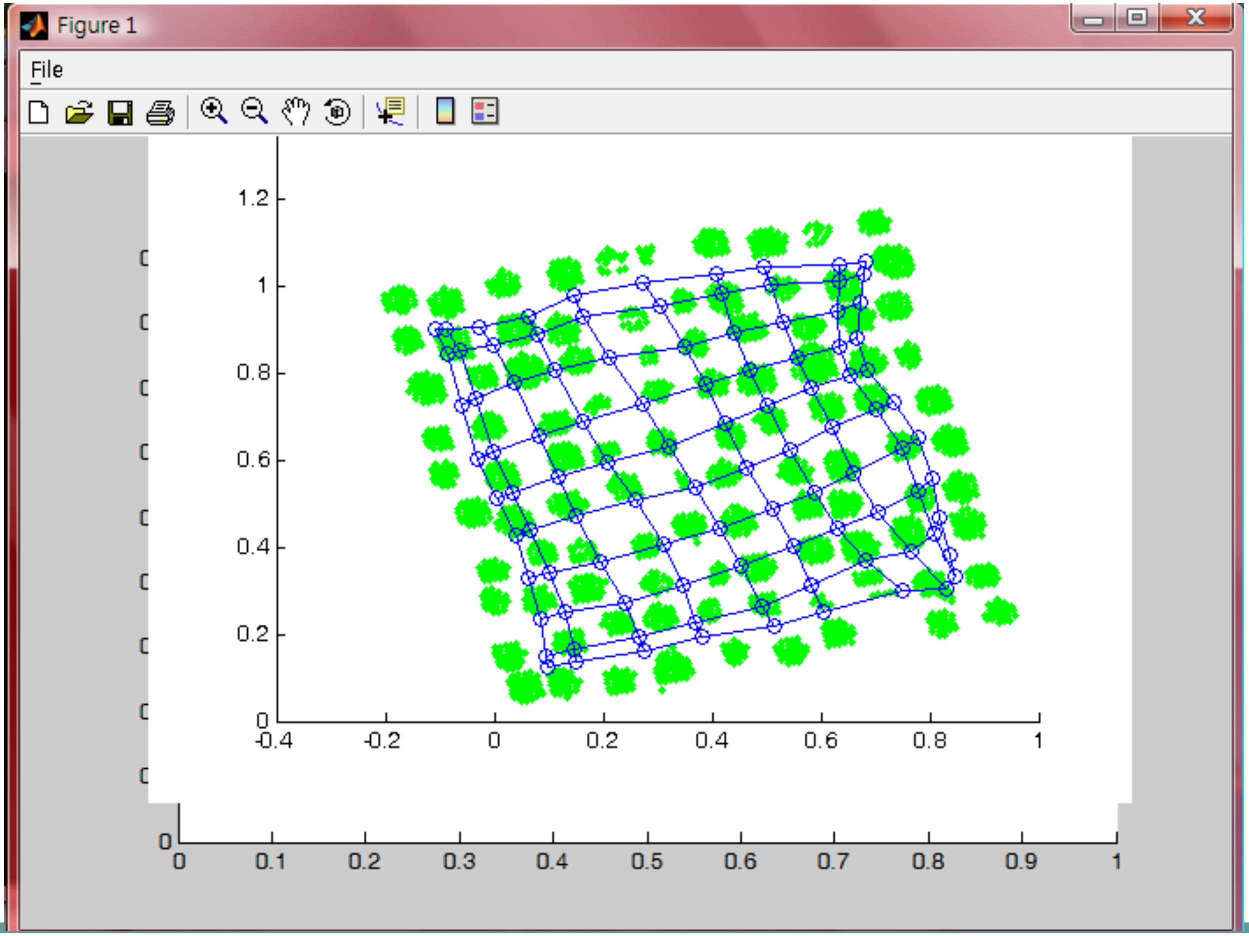


Rotated distributed clusters

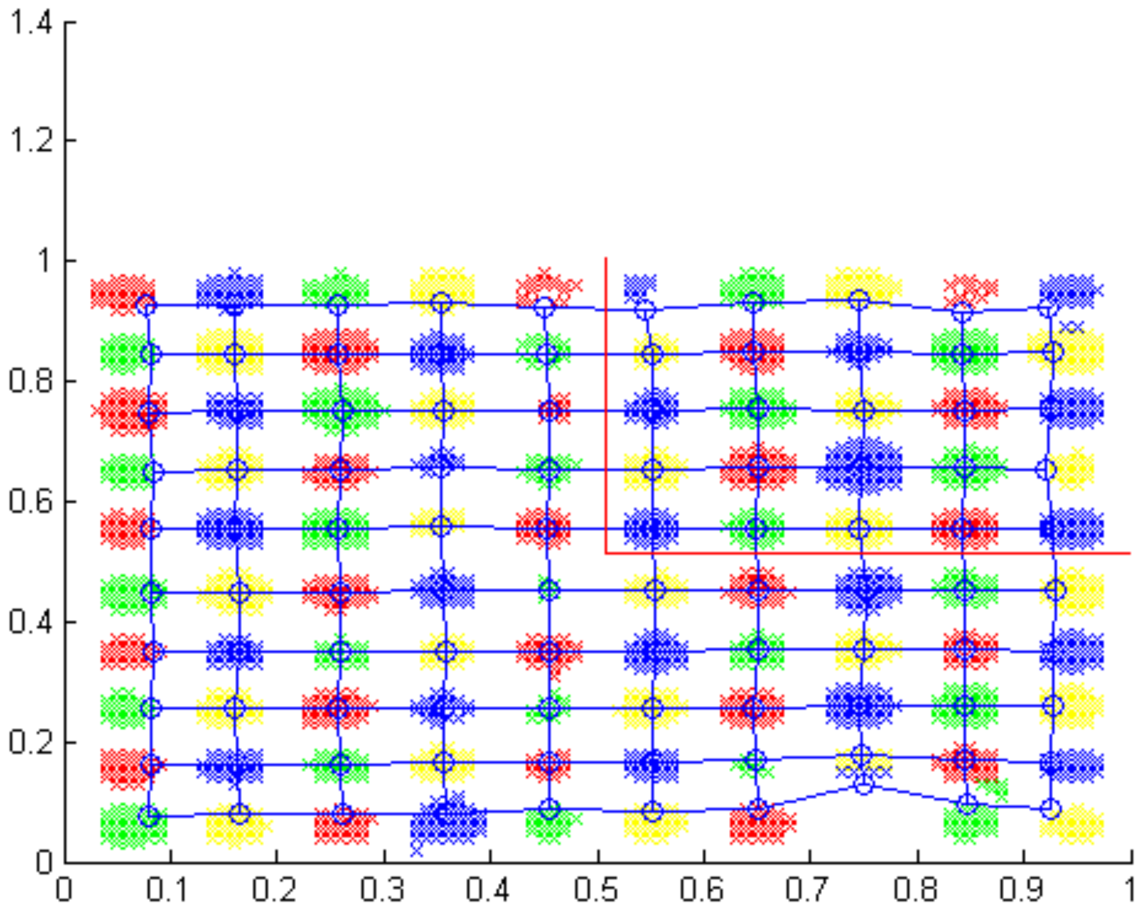


Deformable gridding

Place a lattice to structure distributed clusters



Self-organization



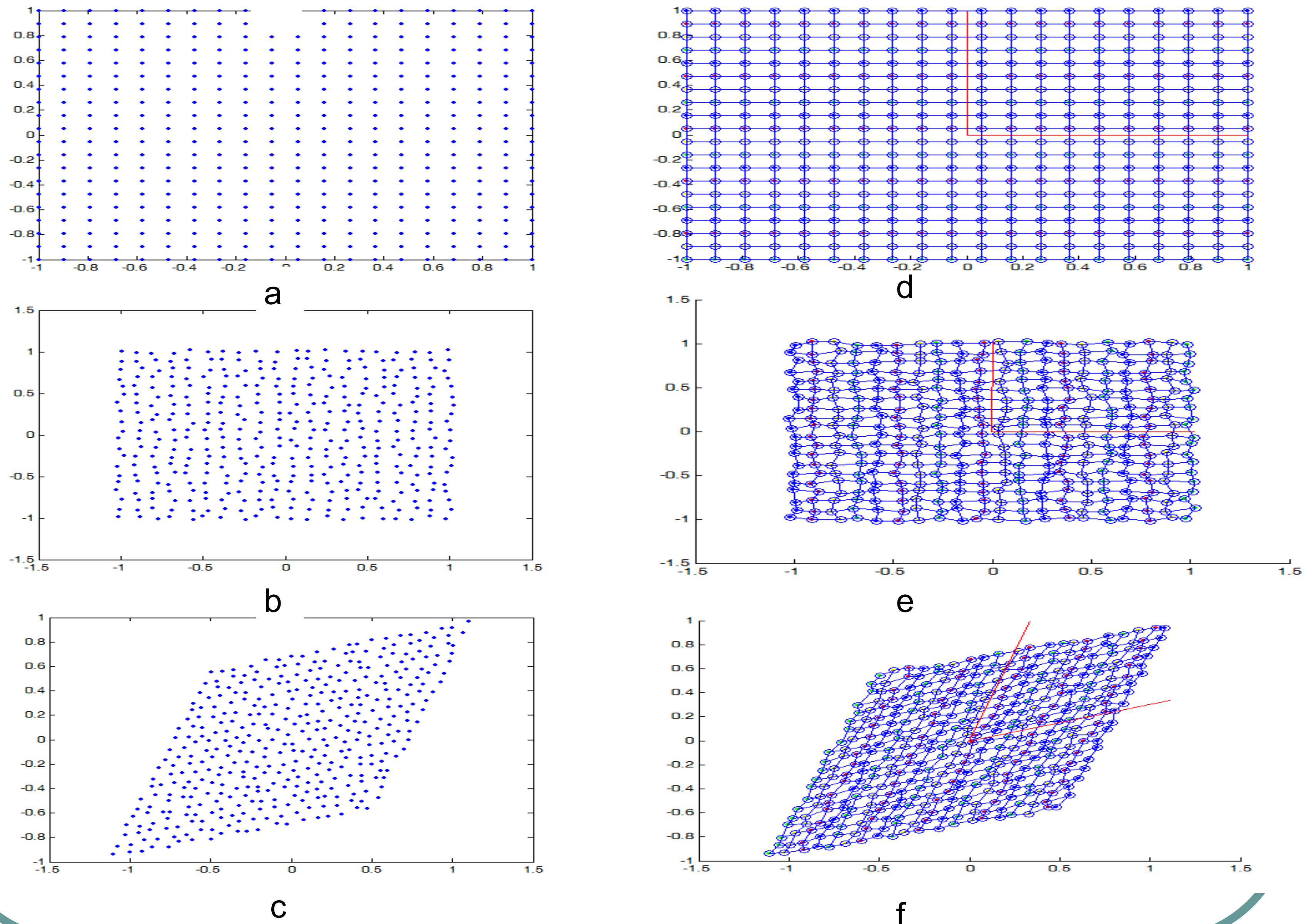


Figure 3

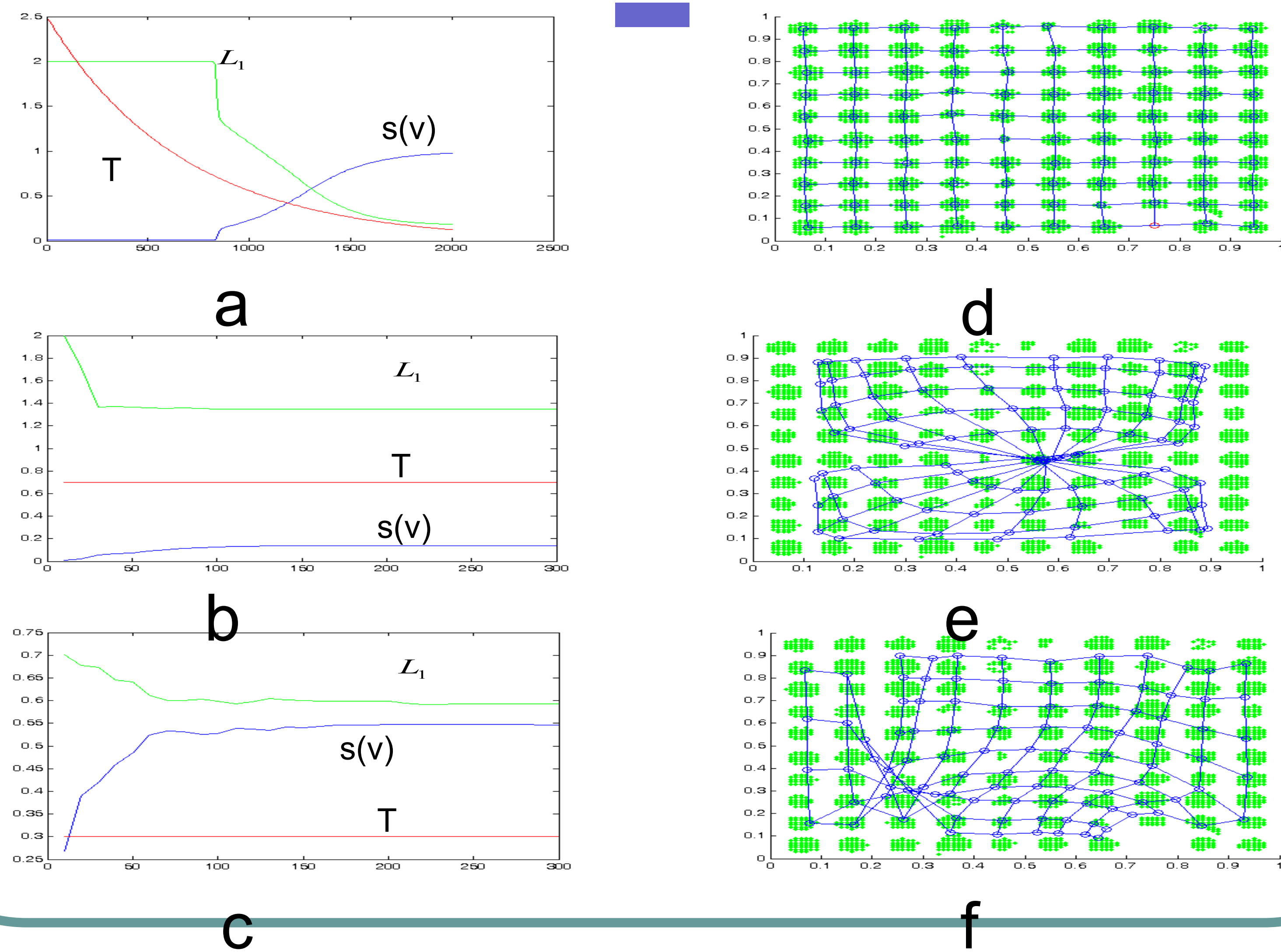
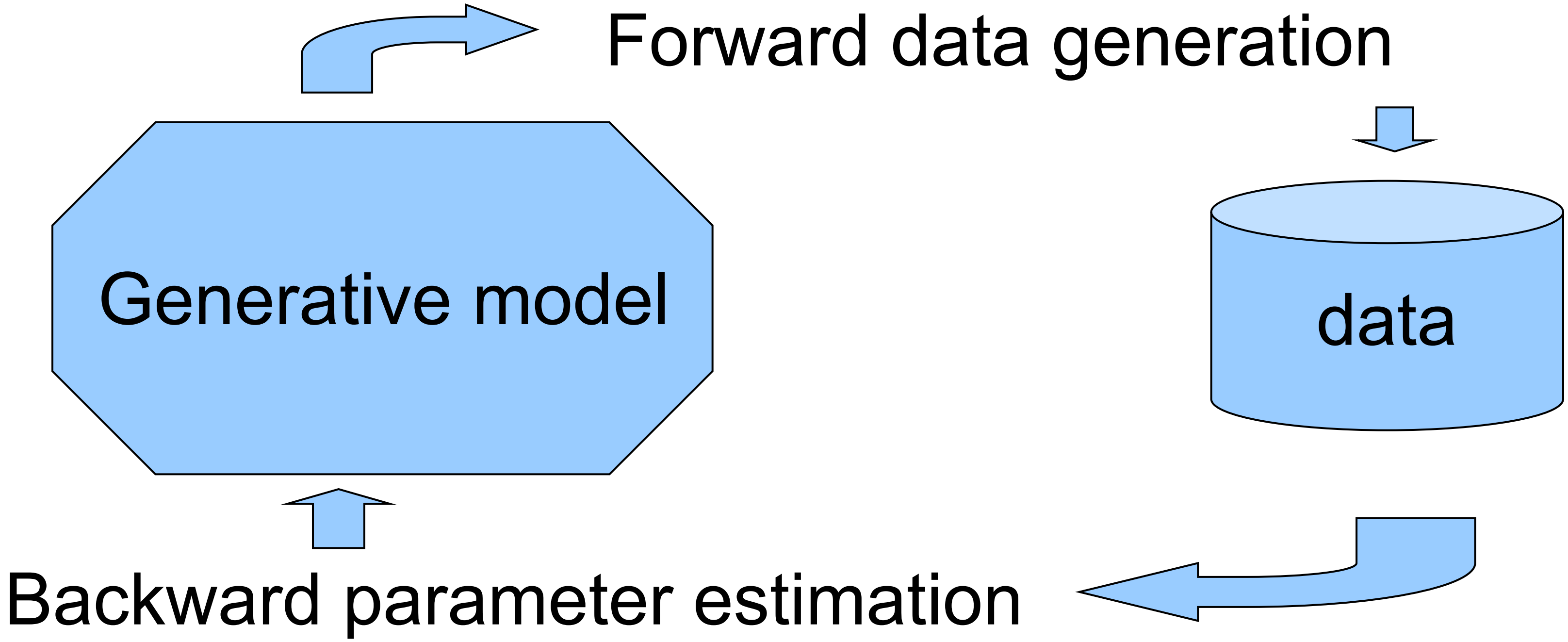
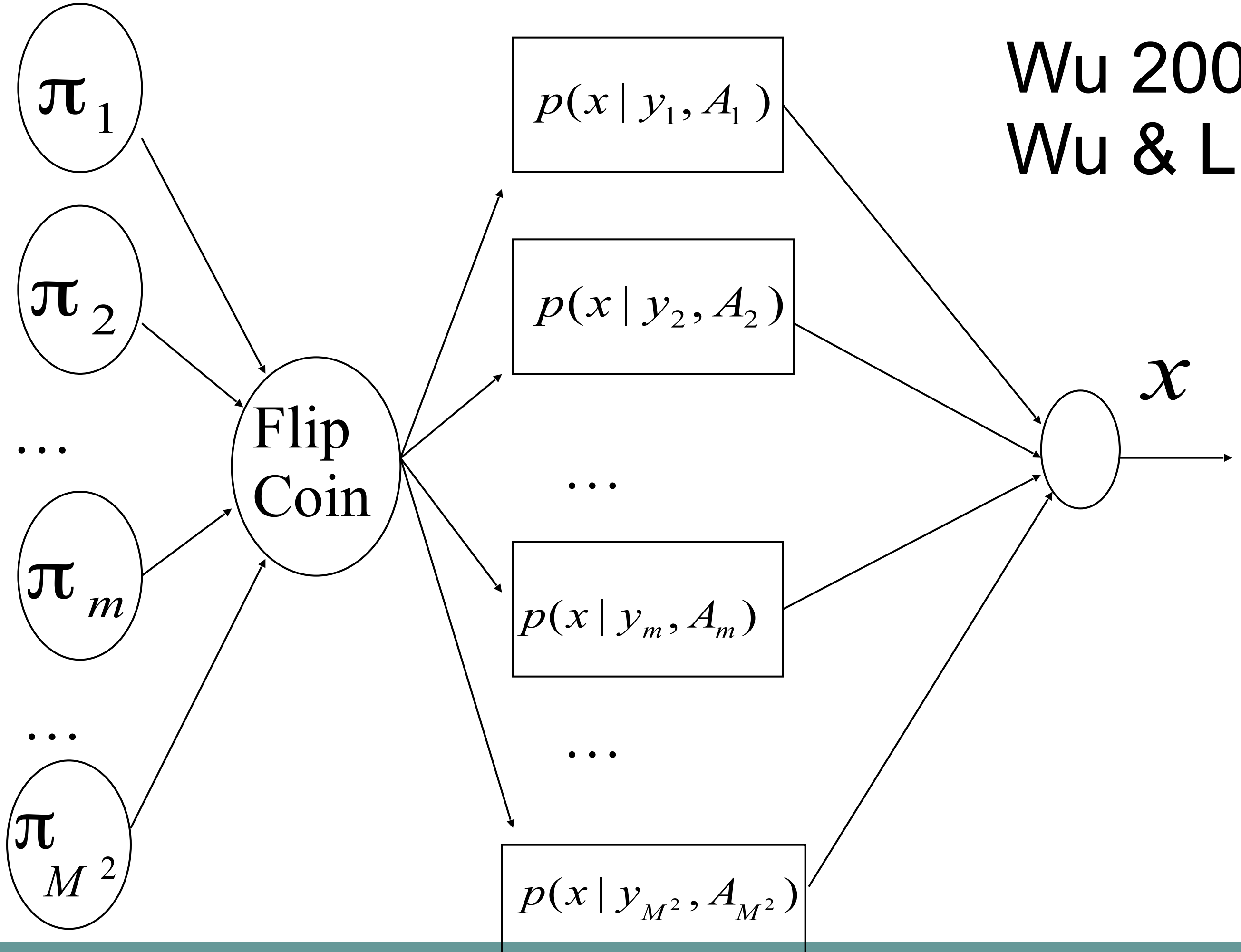


Figure 6

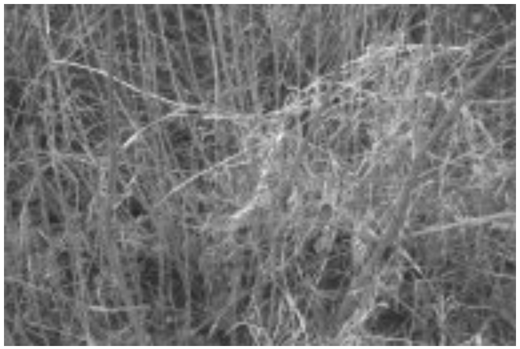

Forward data generation
Generative model
data
Backward parameter estimation

Gaussian Mixture Generative model

Wu 2002
Wu & Lin 2002



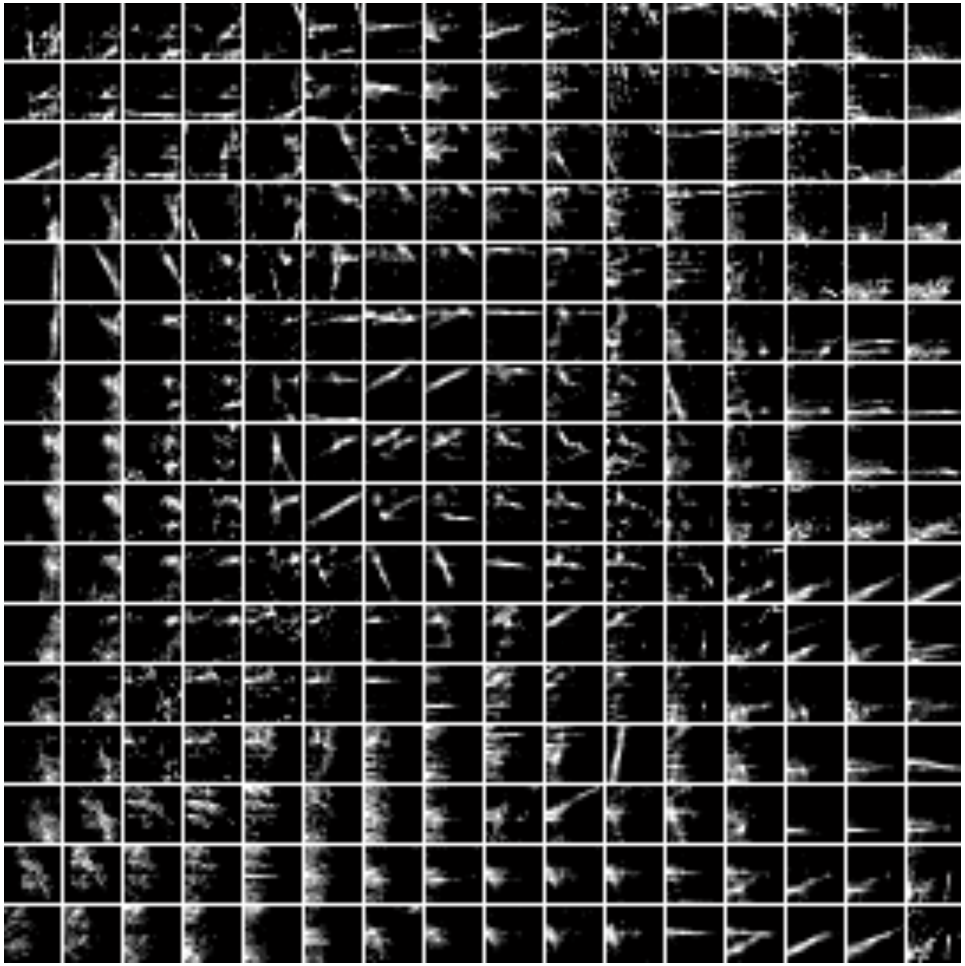
Analysis of Natural images

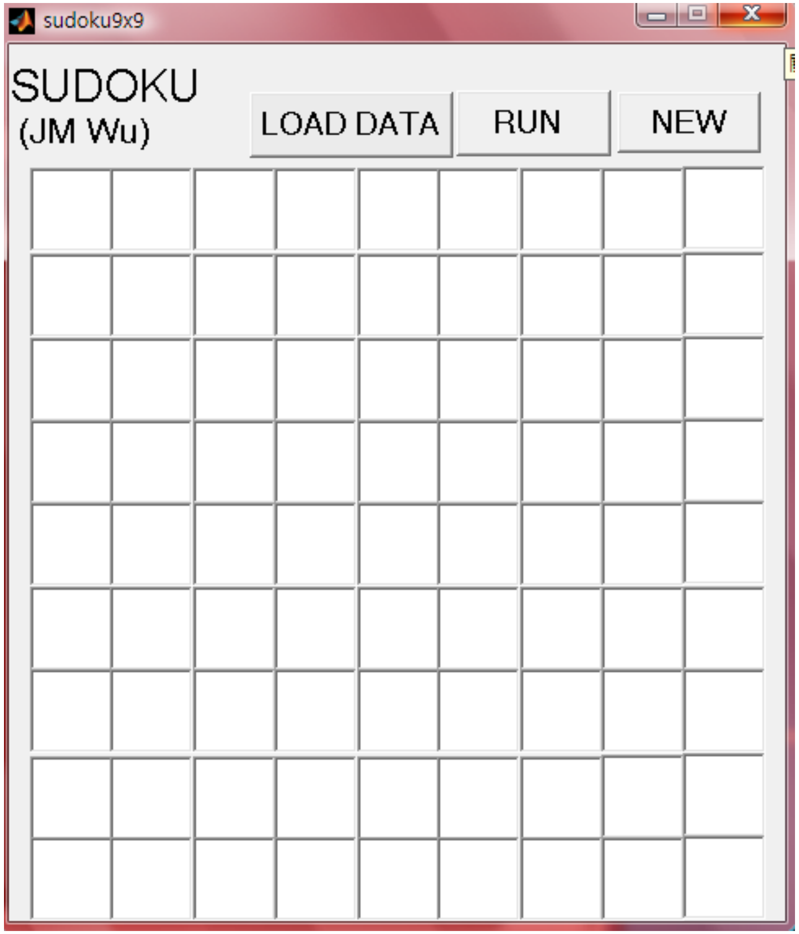
  

Generative models of natural images

Local means



Sudoku



Sudoku

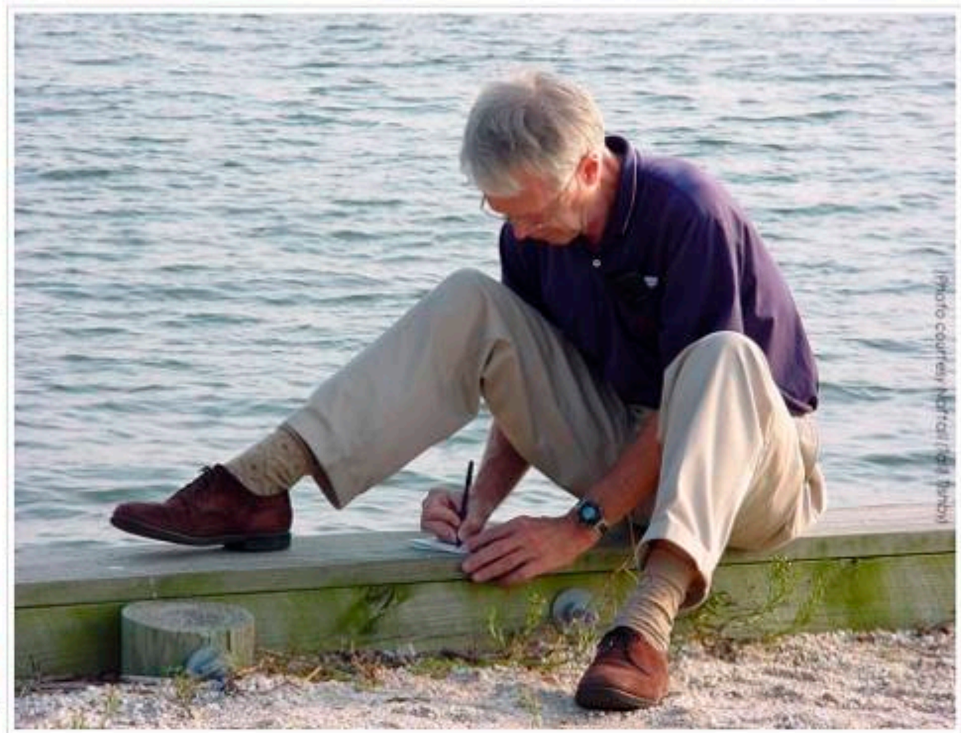

sudoku9x9
SUDOKU
(JM Wu)
LOAD DATA
RUN
NEW
4 5 2
8 6
2 5 8 4
6 7 9
1 3 6
3 1 7
1 9 2 6
5 7
6 3 9

速讀與速讀聯想記憶體

Donald Olding Hebb

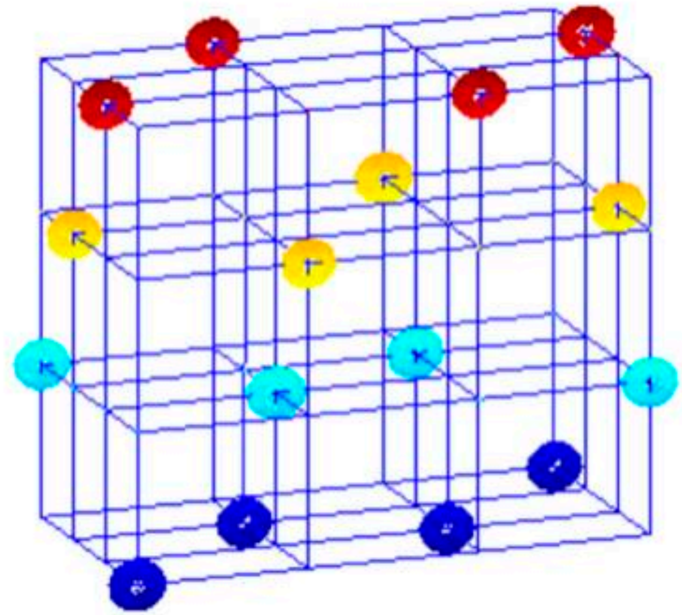
John Joseph Hopfield

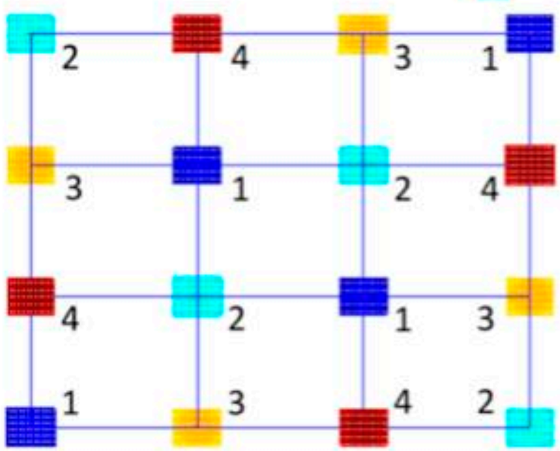
Neural Networks



Sudoku

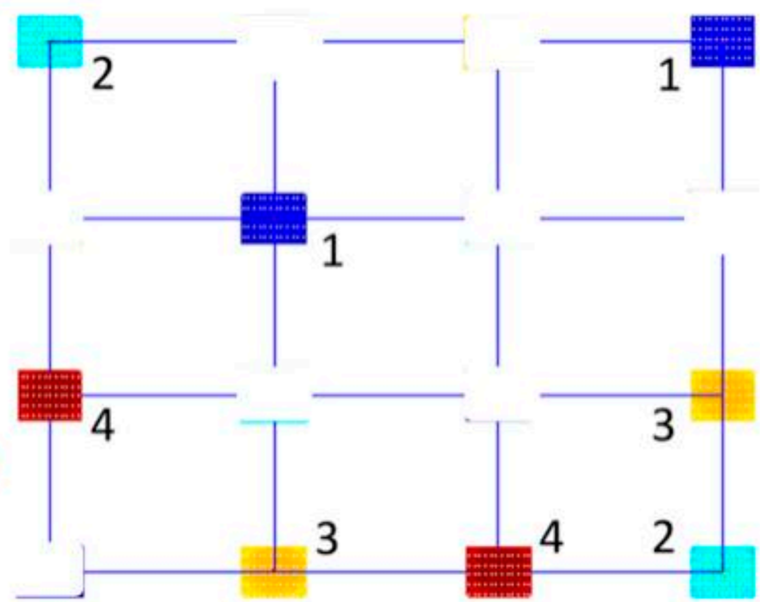
整數規劃數學求解
0 與 1的數學規劃式





S =

2	4	3	1
3	1	2	4
4	2	1	3
1	3	4	2



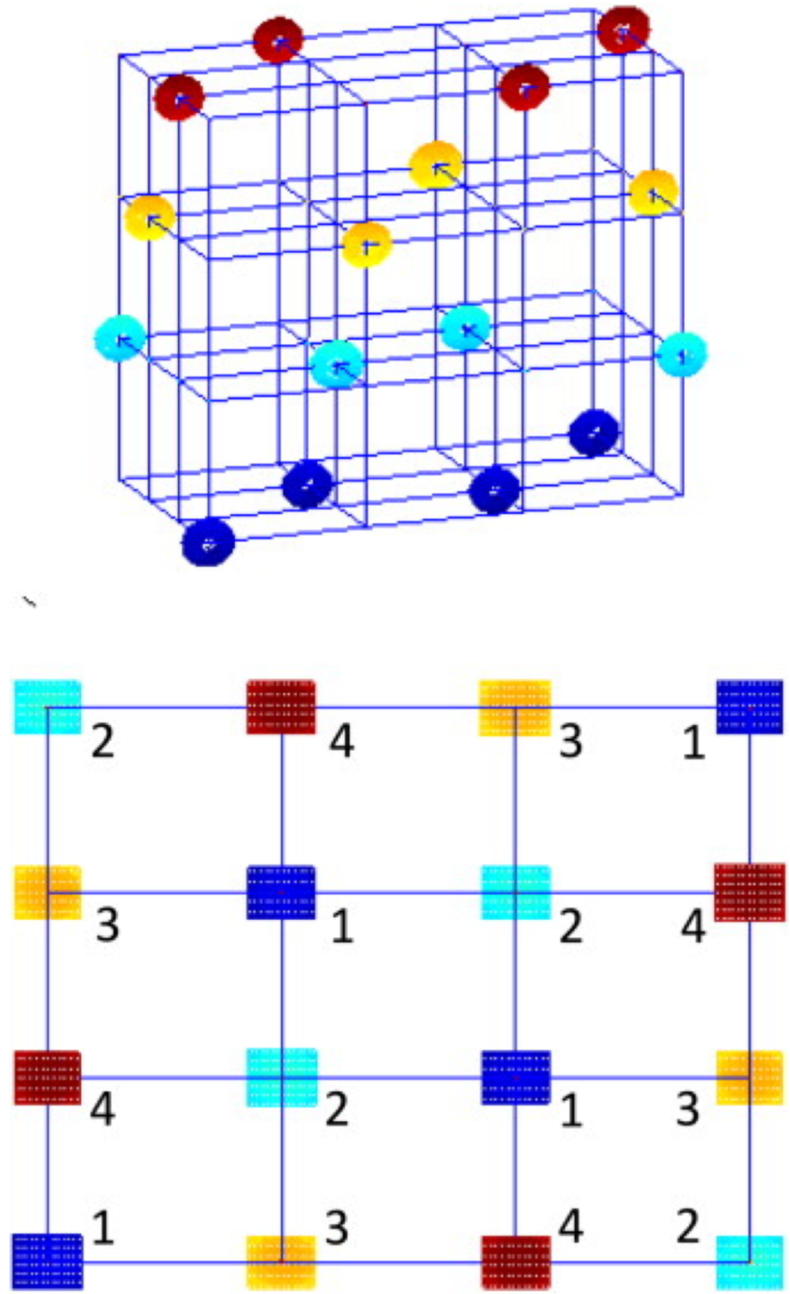


Fig. 6. Bipolar neural activations for Sudoku encoding of $K = 4$.

research direction of Sudoku associative memory. The goal is to achieve automatic error detection, error correction and restoration of Sudoku-rule embedded patterns subject to fewer partial clues, condense clues and perturbed or damaged clues.

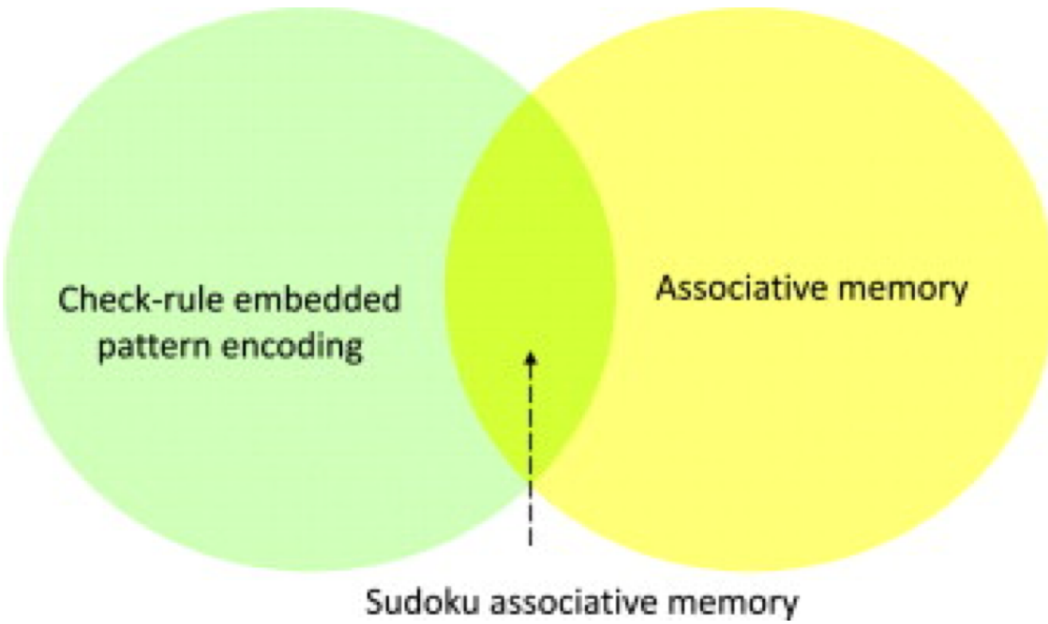
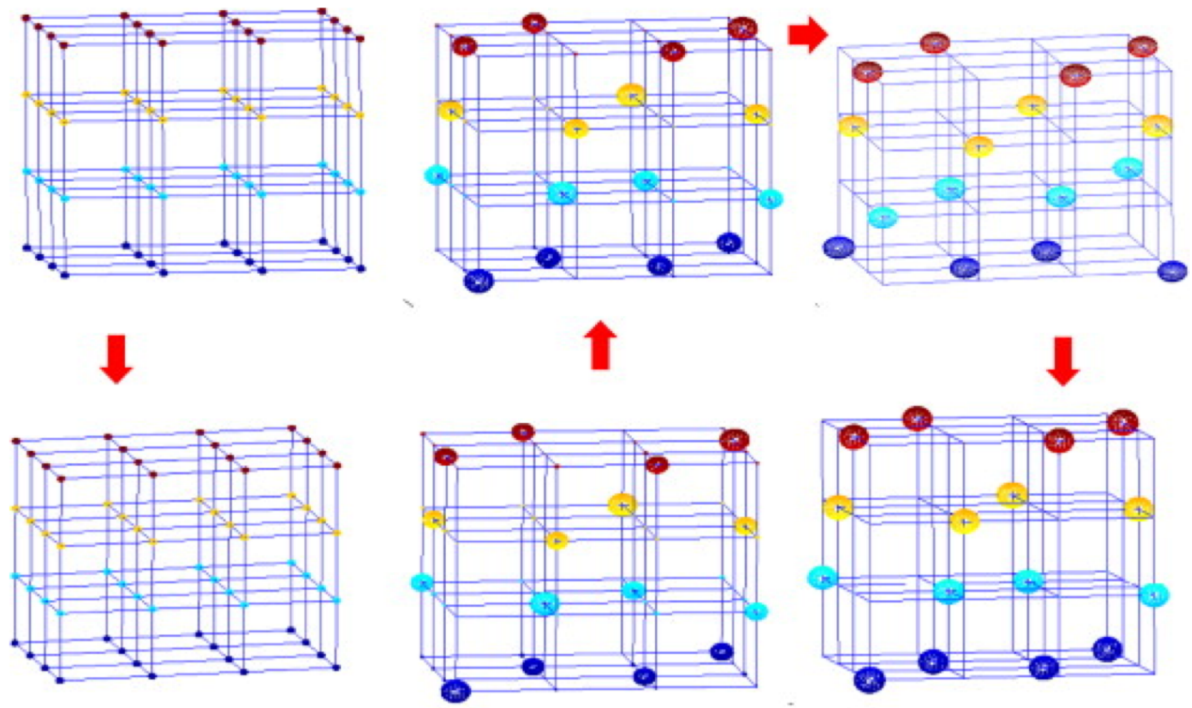
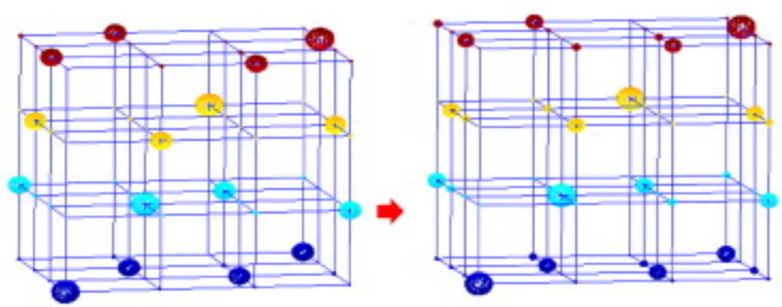


Fig. 7. The concept of developing SAM based on check-rule embedded pattern encoding and associative memory.





2	4	3	1
3	1	2	4
4	2	1	3
1	3	4	2

Fig. 10. Evolution of neural activations for general Sudoku restoration with $K = 4$ along an annealing process.

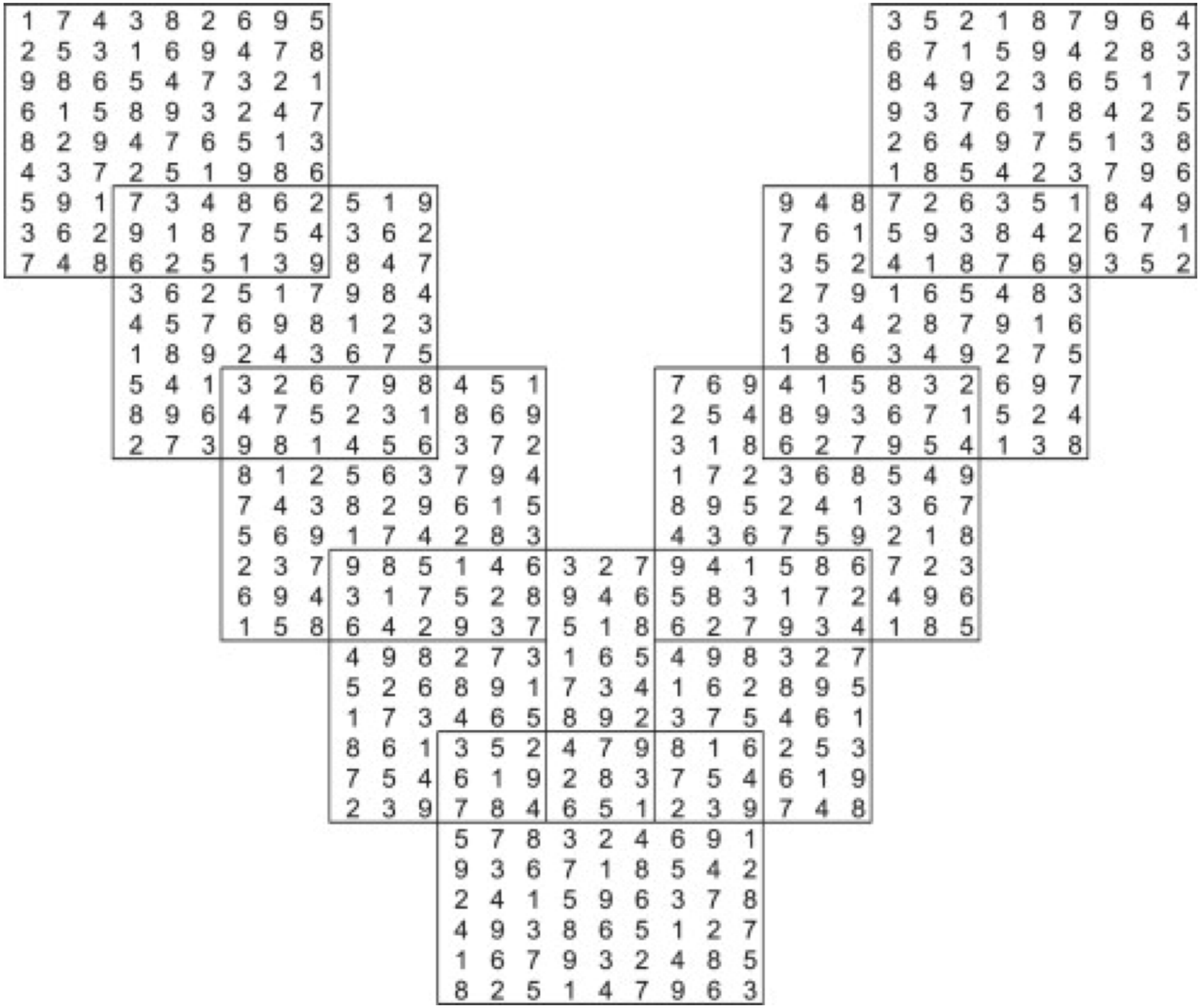


Fig. 12. A V-shape compound Sudoku pattern.

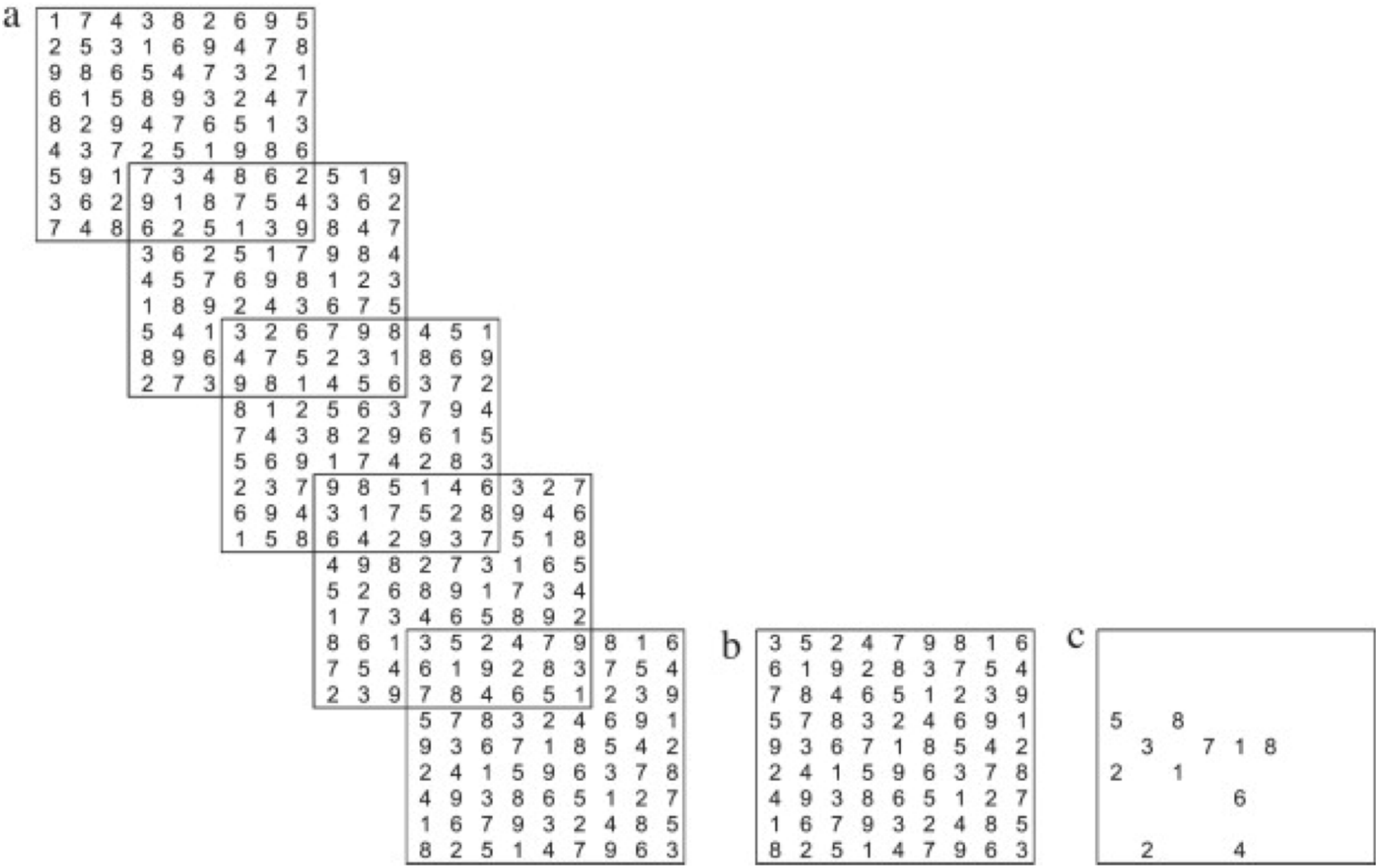


Fig. 13. Different partial clues for V-shape compound Sudoku pattern restoration, (a) a left part of the V-shape compound pattern, (b) a central pattern, and (c) a damaged central pattern.

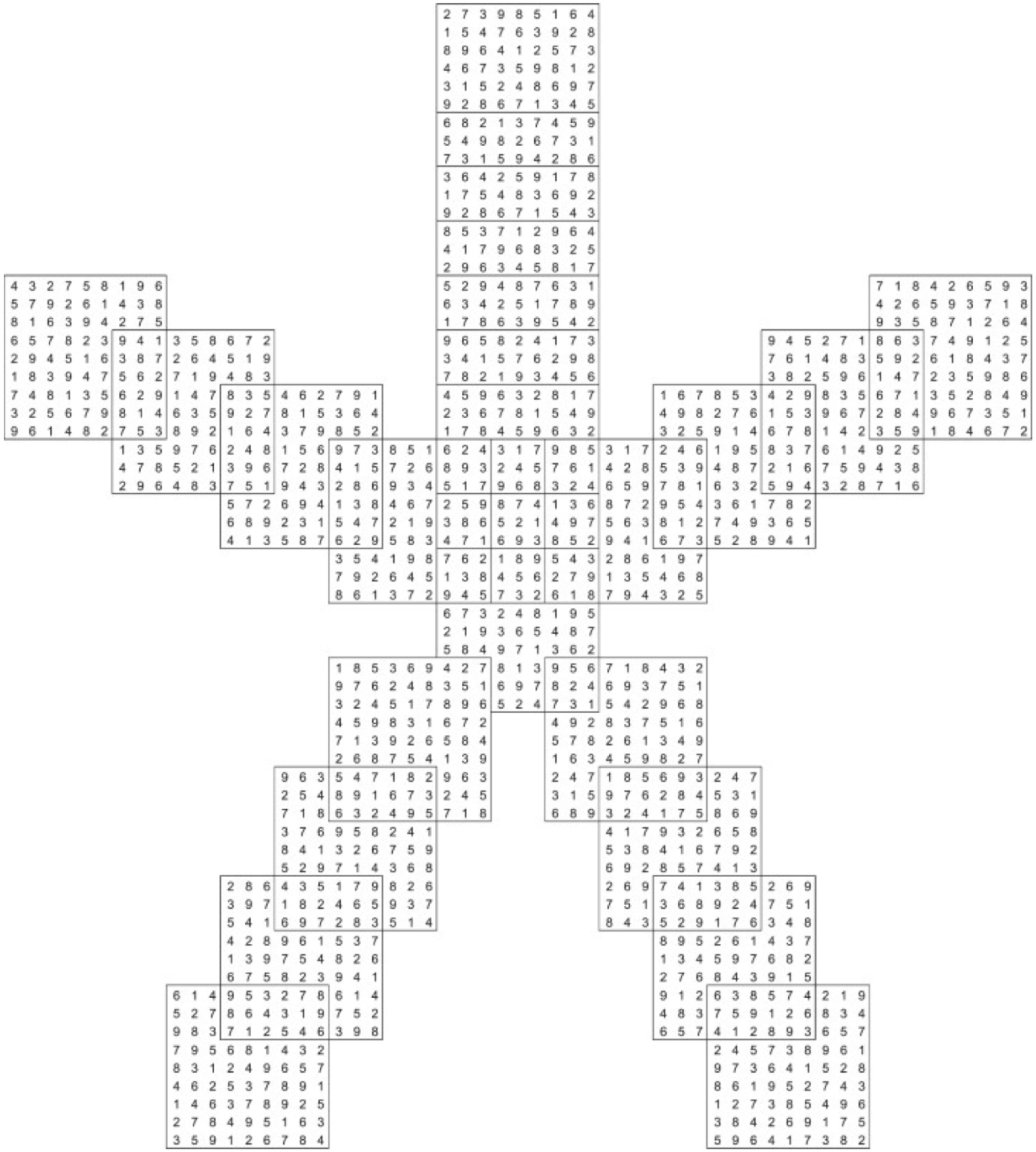


Fig. 14. A starfish-shape compound Sudoku pattern.

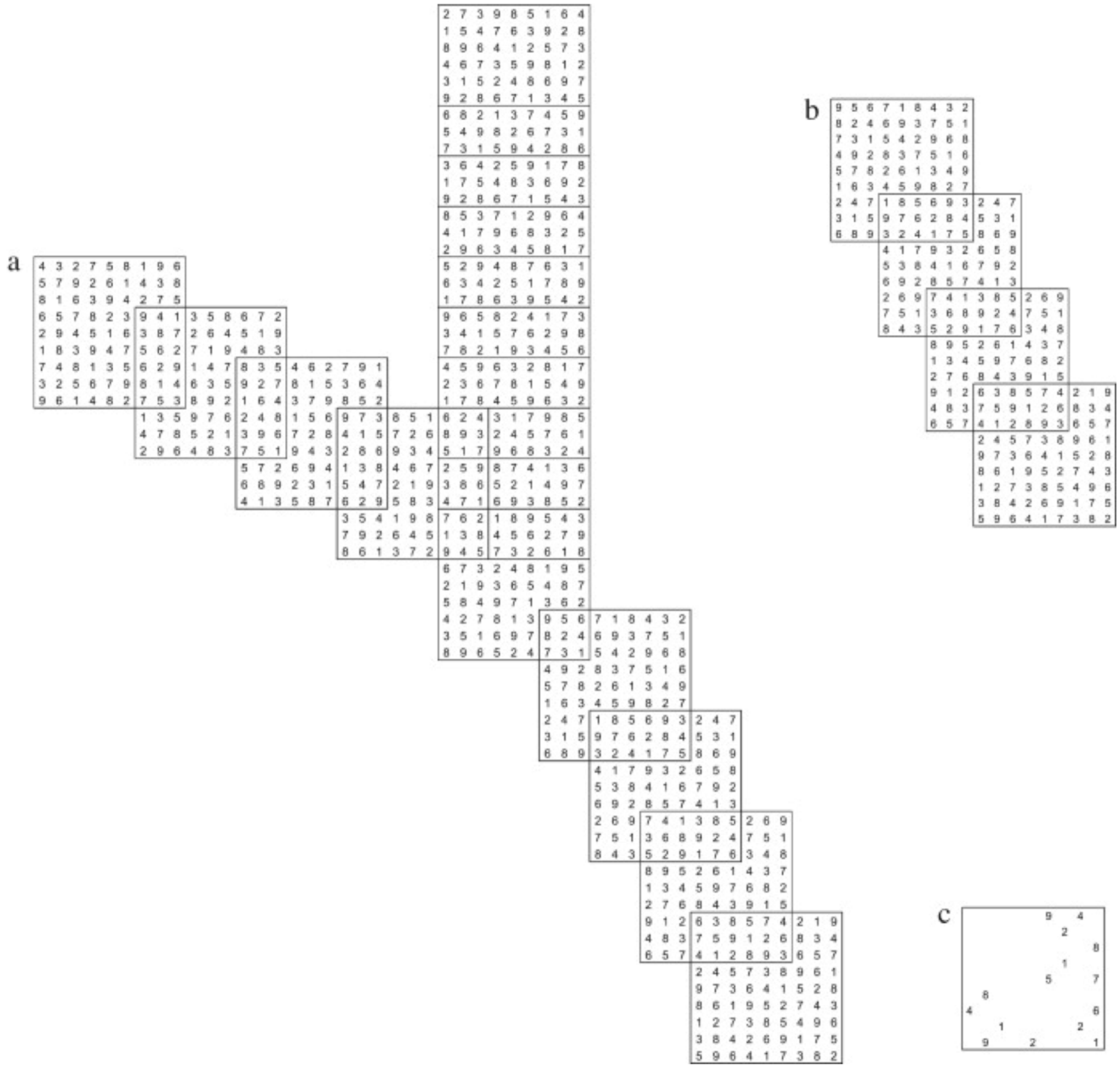
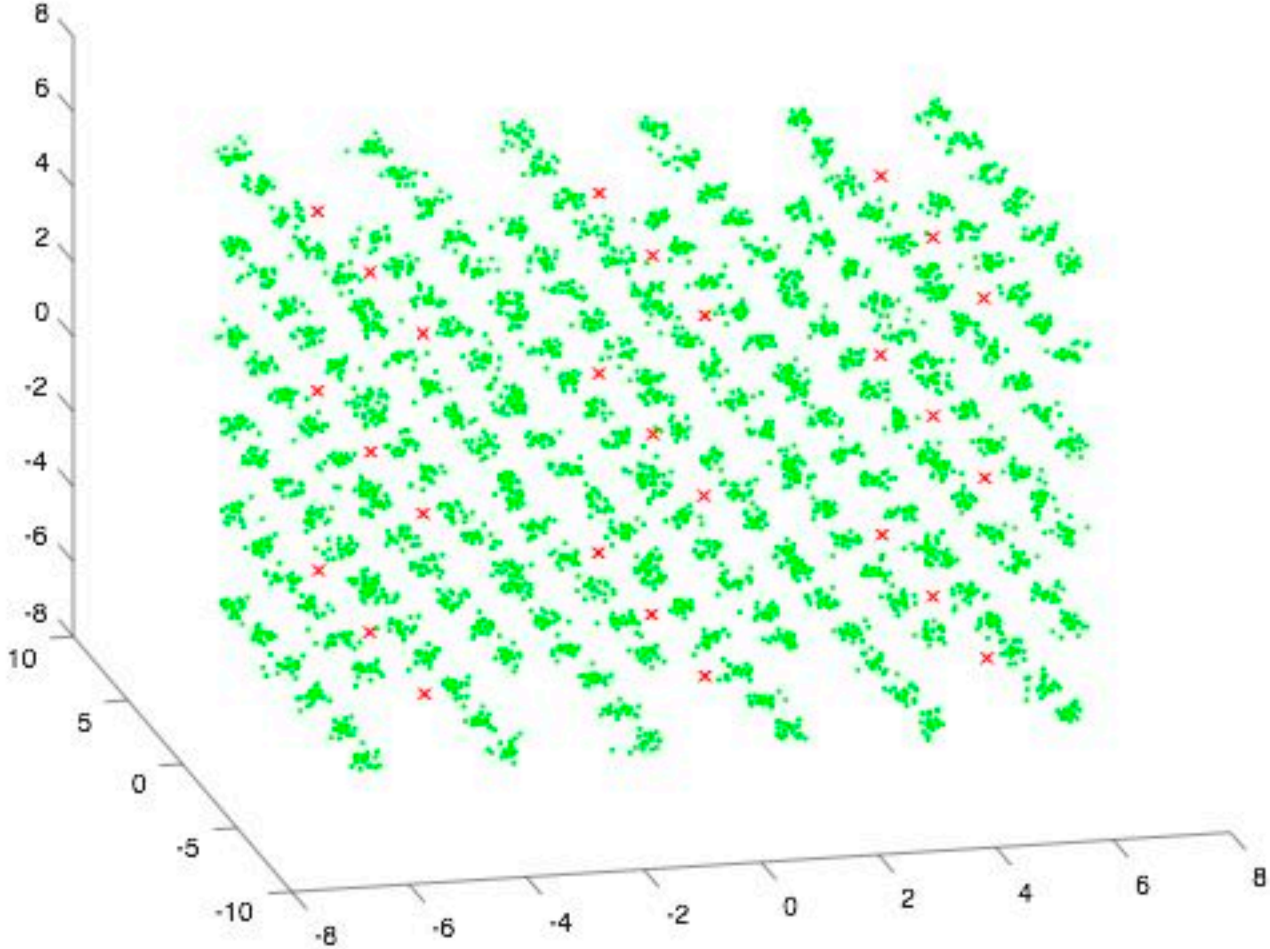
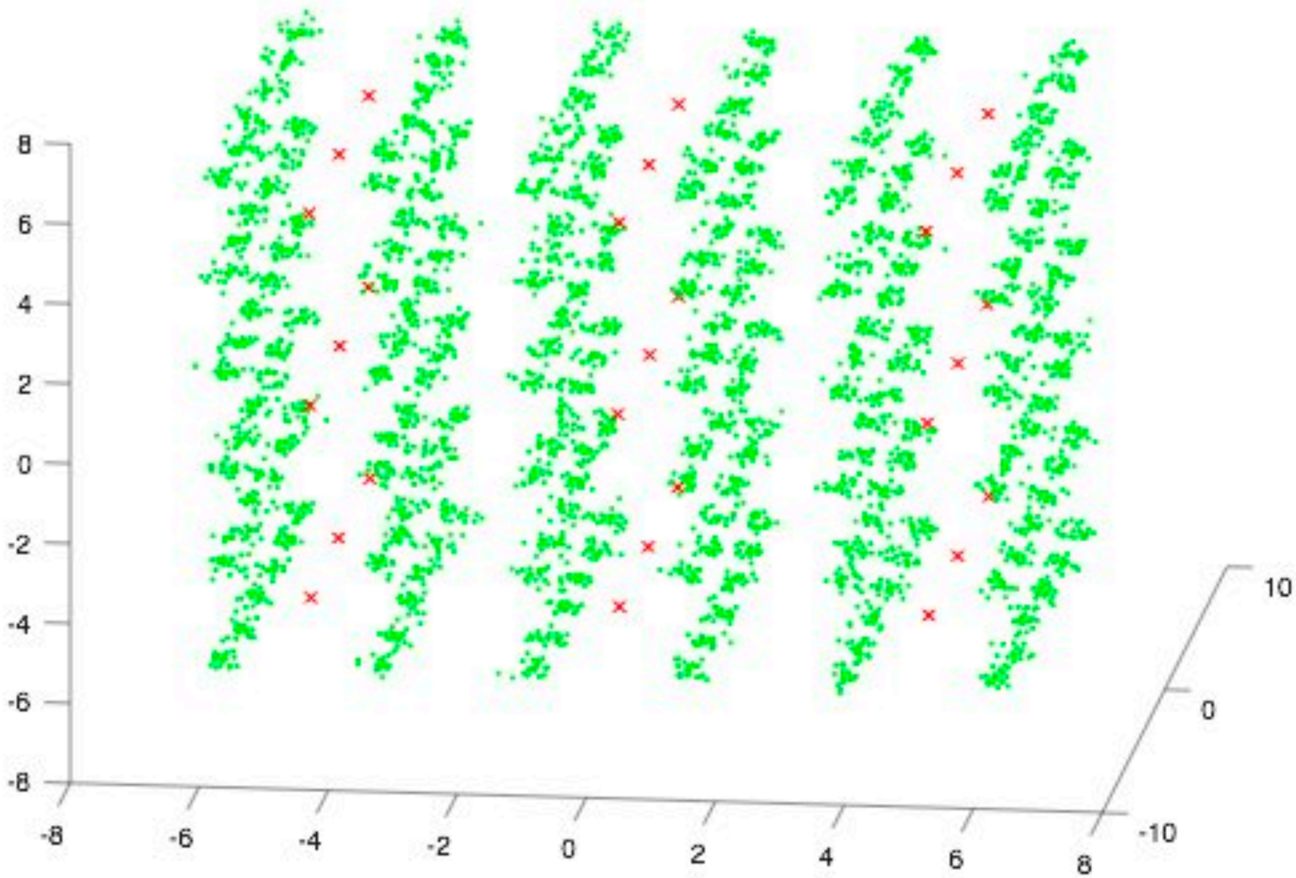
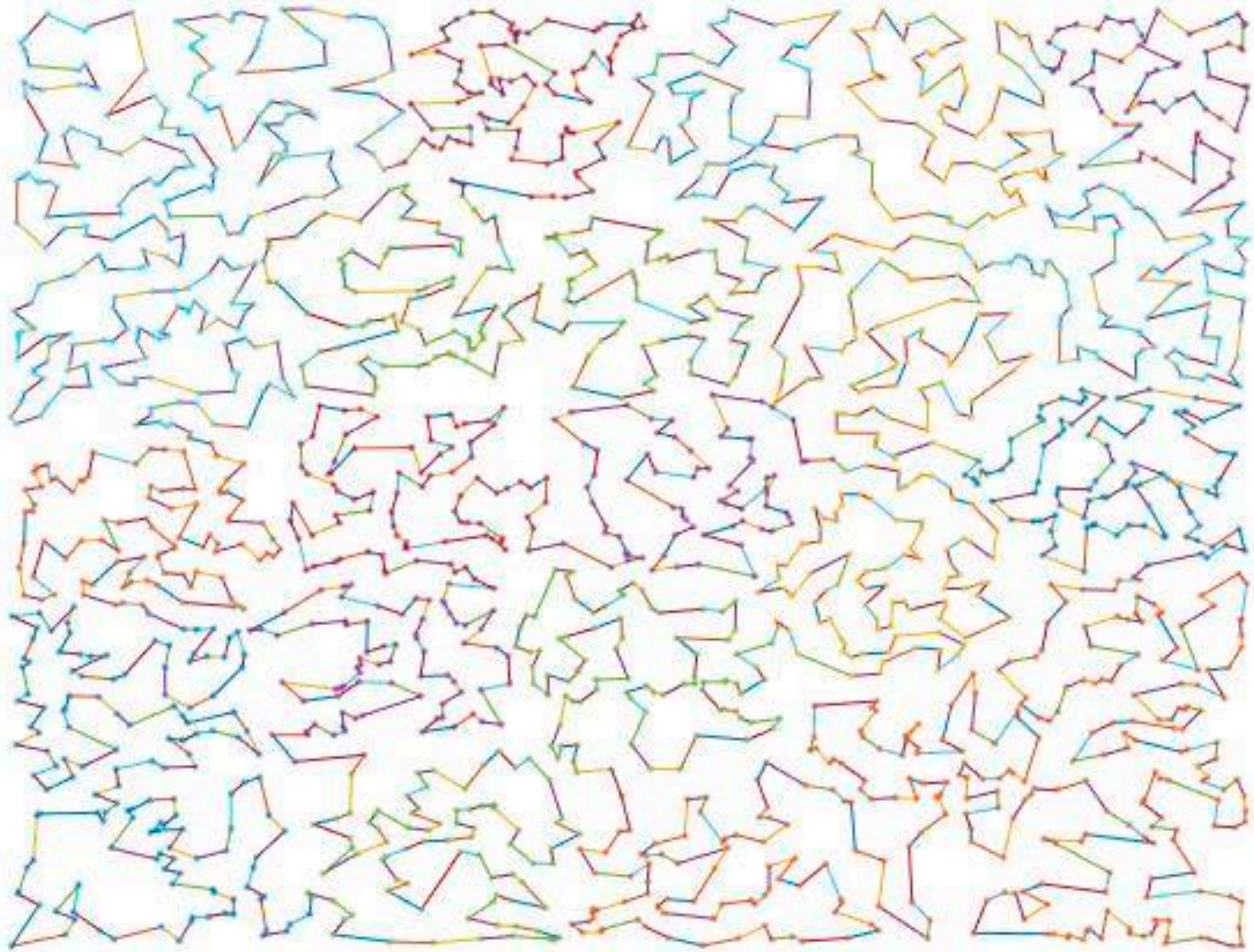
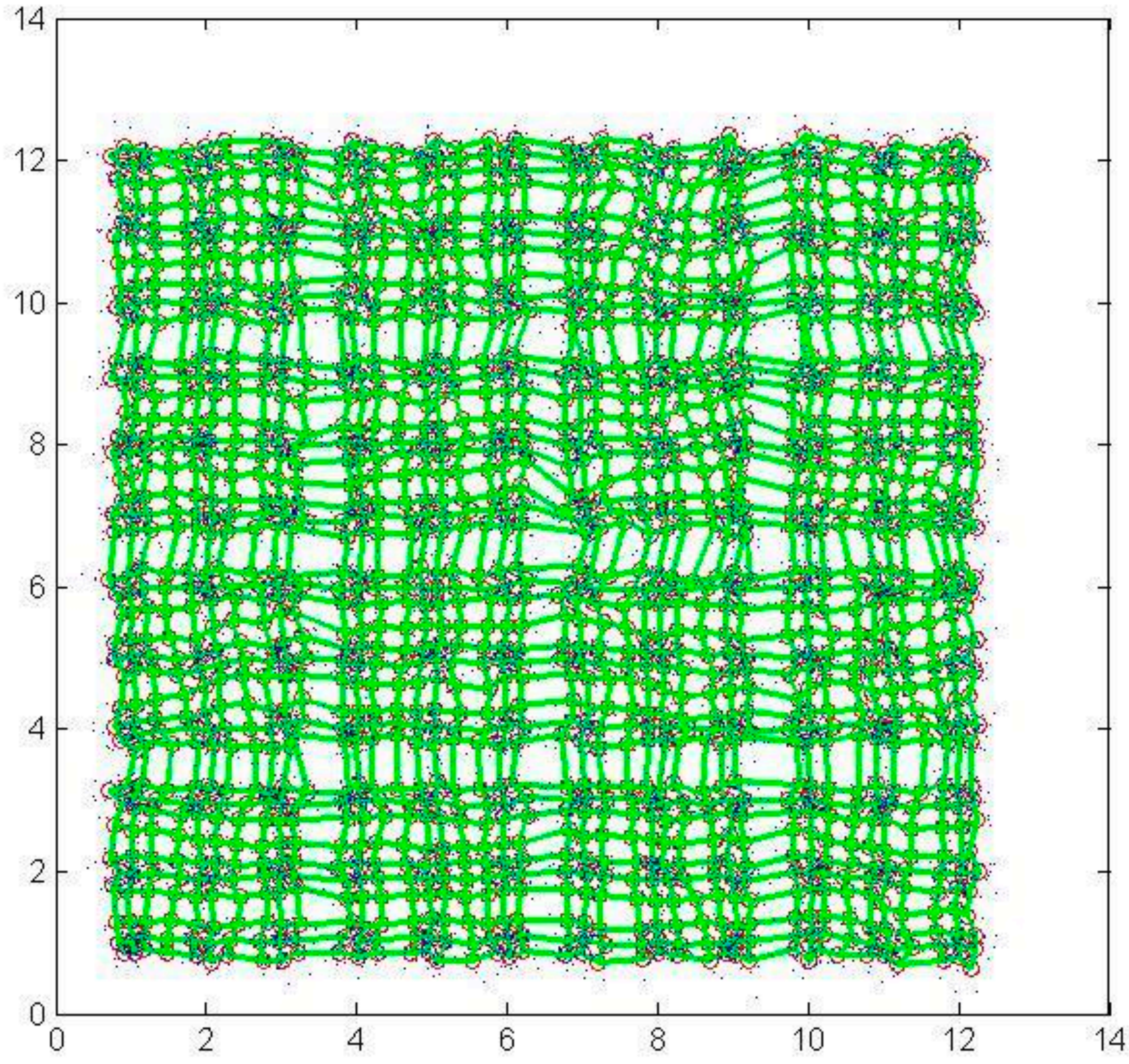


Fig. 15. Different partial clues for starfish shape compound Sudoku pattern restoration, (a) three tentacles, (b) one tentacle, and (c) a damaged seed pattern.


8
6
4
2
0
-2
-4
-6
-8
10
5
0
-5
-10
-8
-6
-4
-2
0
2
4
6
8

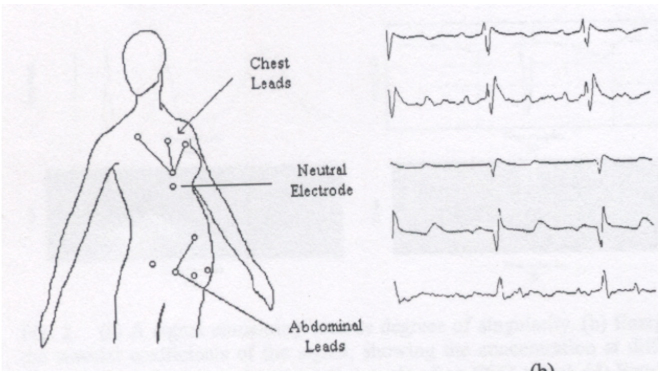

8
6
4
2
0
-2
-4
-6
-8
-8
-6
-4
-2
0
2
4
6
8
10
0
-10

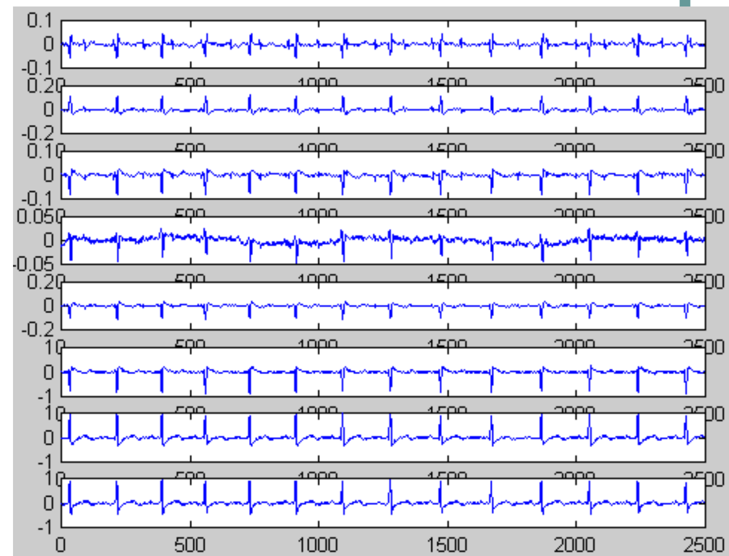




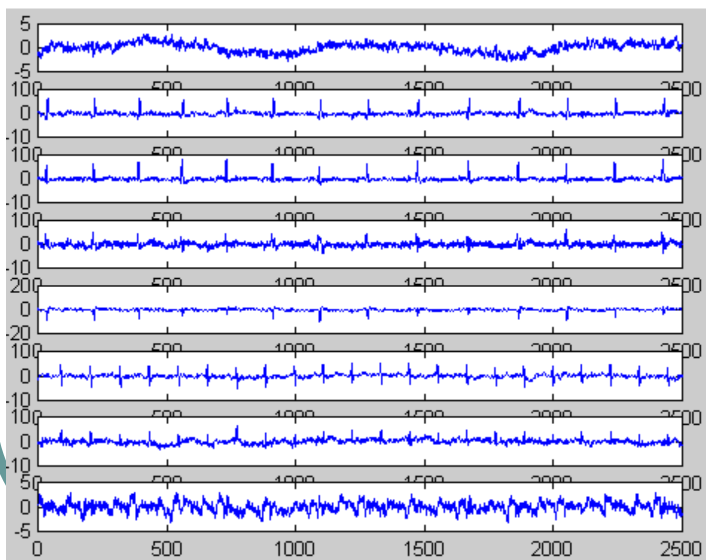
14
12
10
8
6
4
2
0
0
2
4
6
8
10
12
14

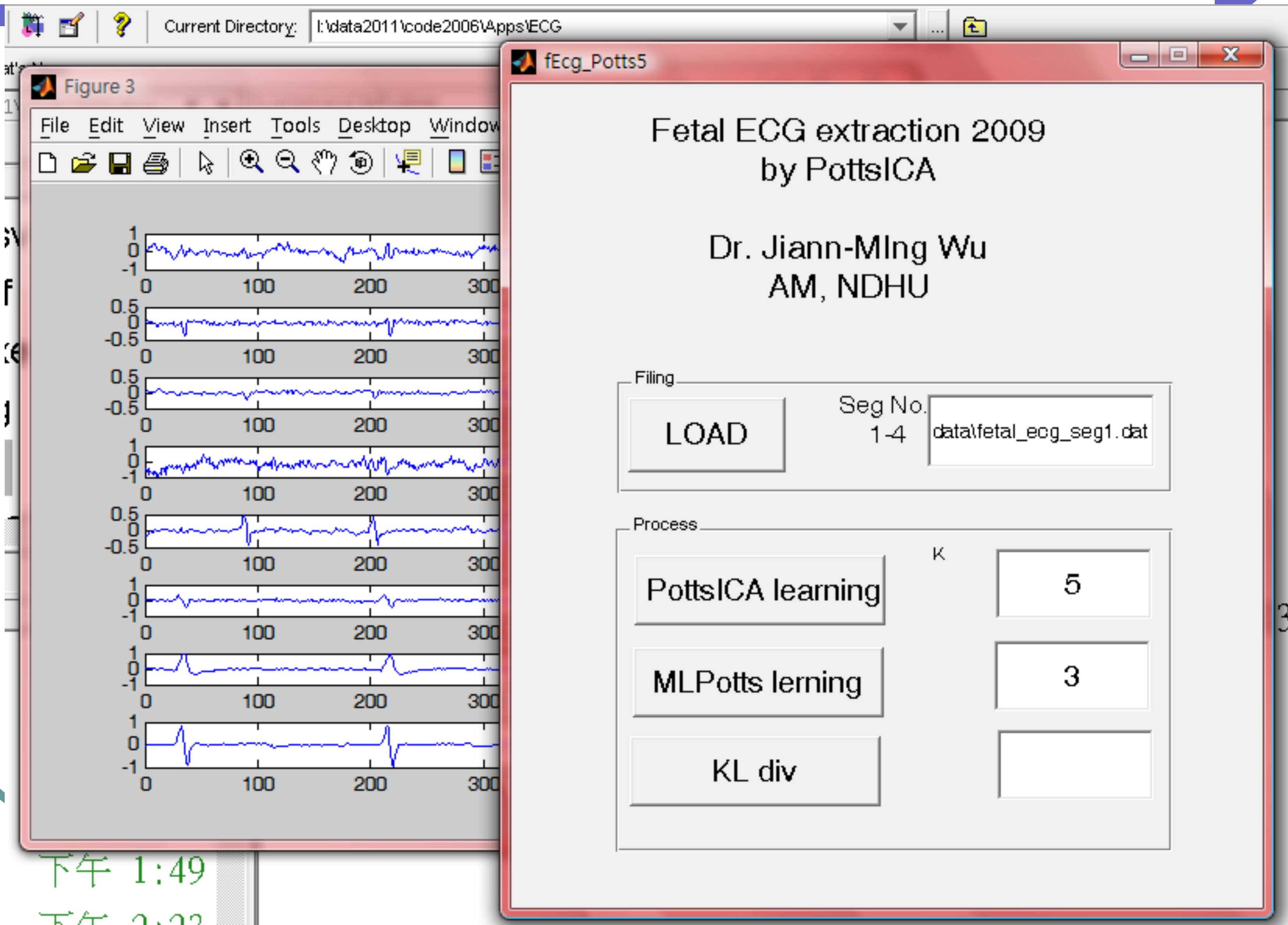
Blind source separation – fetal ECG





BSS

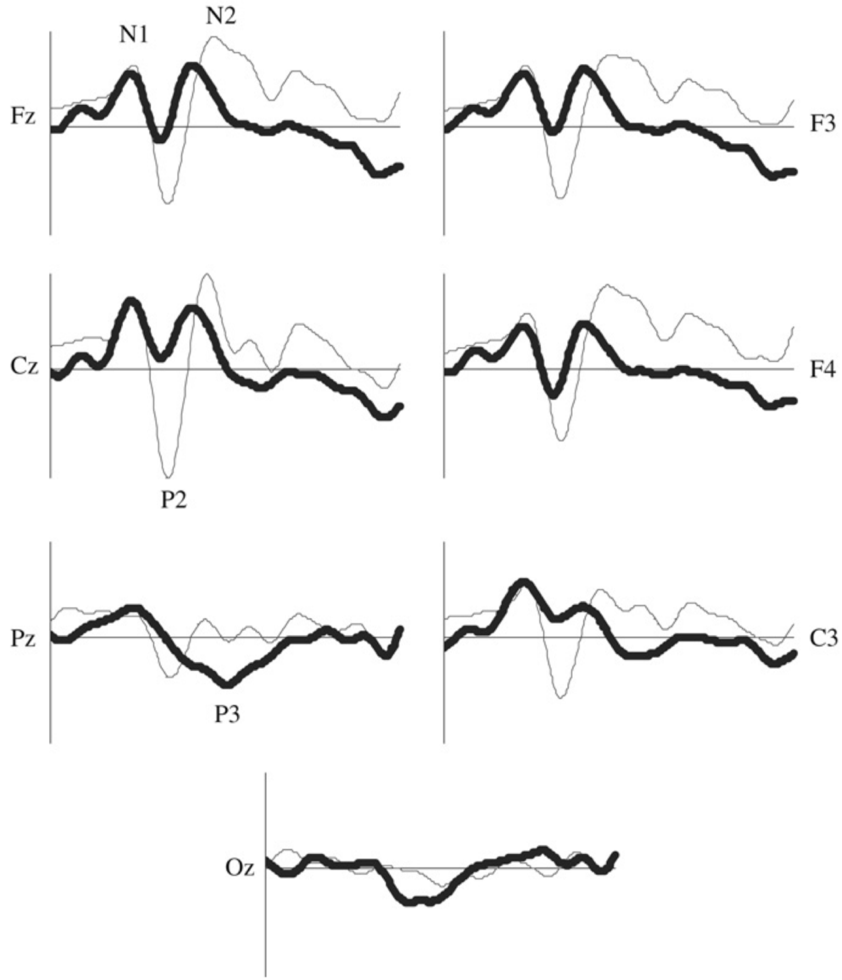



Current Directory: I:\data2011\code2006\Apps\ECG
Figure 3
File Edit View Insert Tools Desktop Window
fEcg_Potts5
Fetal ECG extraction 2009
by PottsICA
Dr. Jiann-MIng Wu
AM, NDHU
Filing
LOAD
Seg No.
1-4
data\fetal_ecg_seg1.dat
Process
K
PottsICA learning
5
MLPotts lerning
3
KL div
下午 1:49

Mixed Facial images

Wu et al 2008

ERP(event related potential)



n

Fig. 11. Observed ERPs.

ICs of ERP (Wu et al 2008)

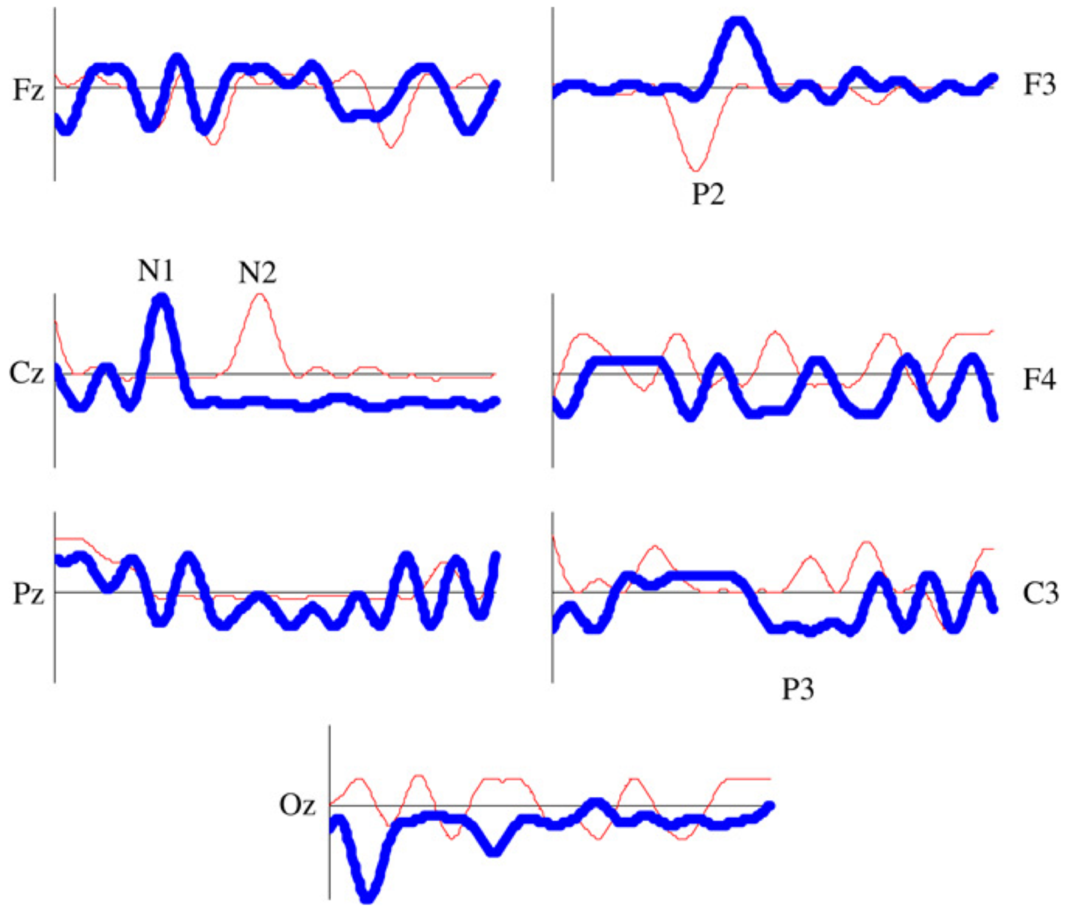


Fig. 12. Independent components obtained by AemICA for blind separation of ERPs.