# Numerical Methods

AM NDHU
Jiann-Ming Wu
2026 Spring

數值方法論

# 電子書

# Introduction to Numerical Methods and MATLAB Programming for Engineers

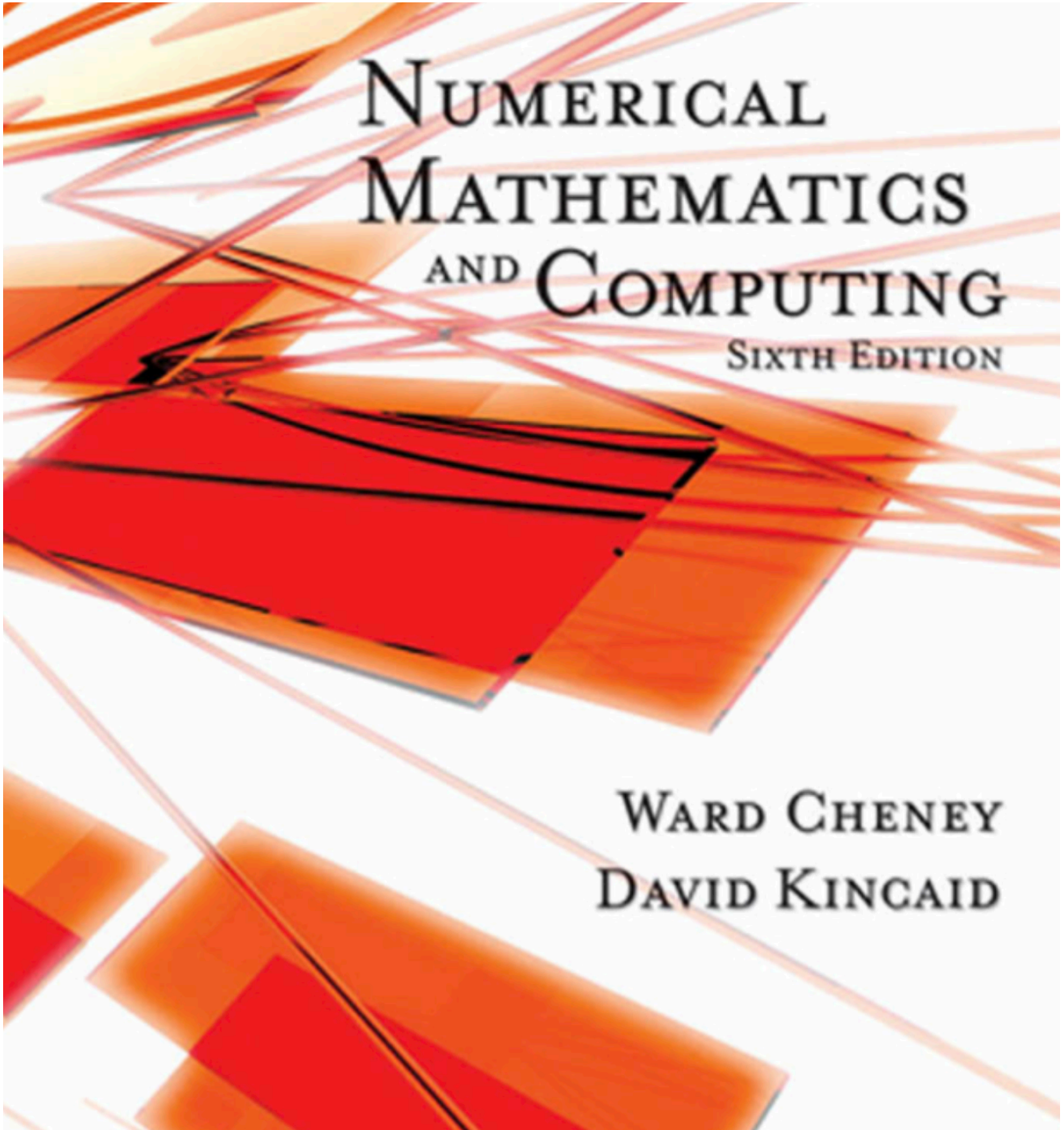
Todd Young and Martin J. Mohlenkamp
Department of Mathematics
Ohio University
Athens, OH 45701
youngt@ohio.edu

December 15, 2014

# Numerical Methods in Scientific Computing

## Volume I

**Germund Dahlquist**
Royal Institute of Technology
Stockholm, Sweden


NUMERICAL
MATHEMATICS
AND COMPUTING
SIXTH EDITION
WARD CHENEY
DAVID KINCAID

# Introduction to Numerical Analysis and Scientific Computing

Nabil R. Nassif and Dolly K. Fayyad

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

# NUMERICAL ANALYSIS
# MATHEMATICS OF SCIENTIFIC COMPUTING

**David Kincaid and Ward Cheney**

The University of Texas at Austin

$$x = \sqrt{3} = 1.???$$

$$abs(x^2 - 3) < 10^{-10}$$

$$abs(x^2 - 3) < 10^{-20}$$

$$abs(x^2 - 3) < 10^{-30}$$

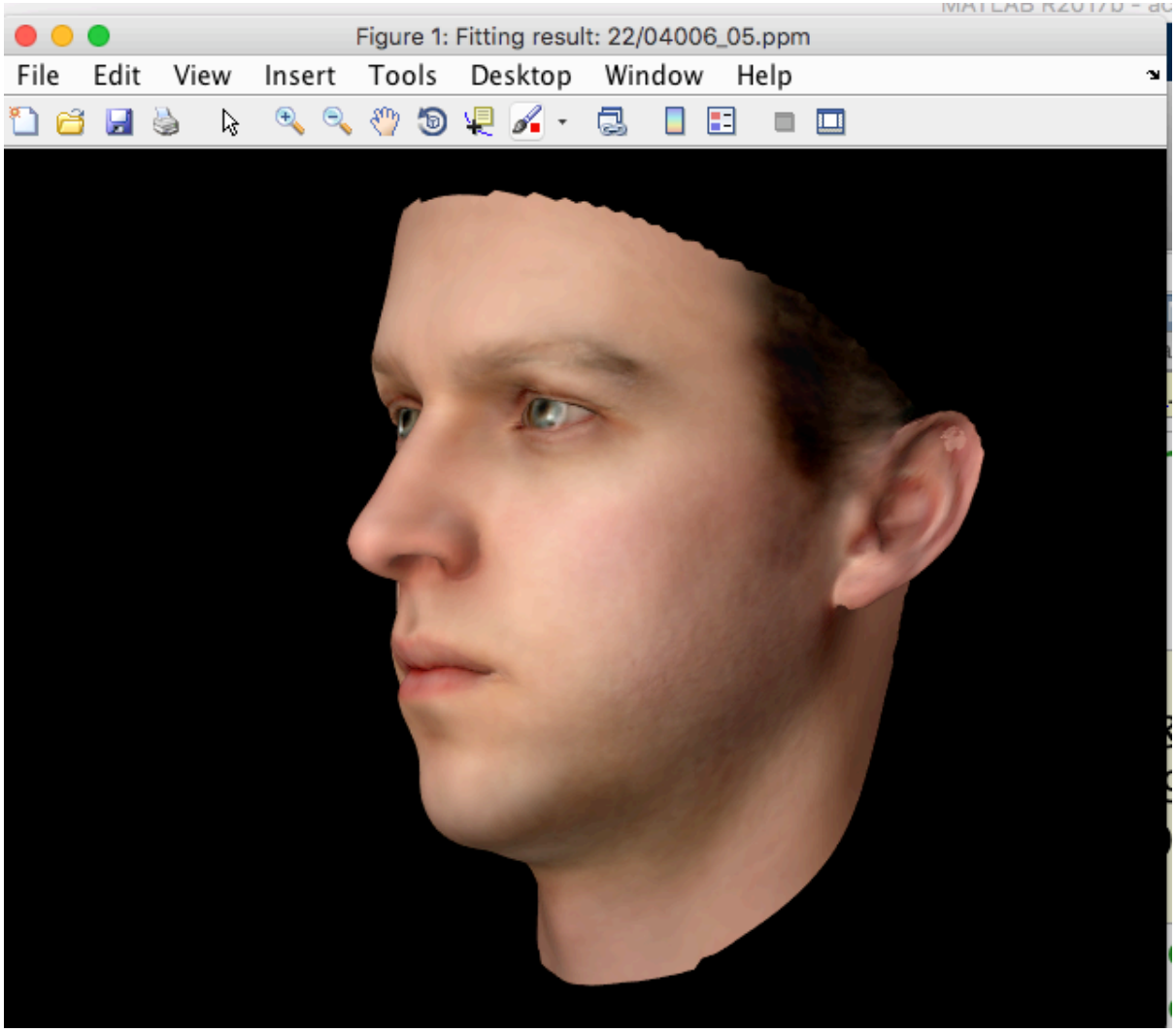
# Matlab Projects

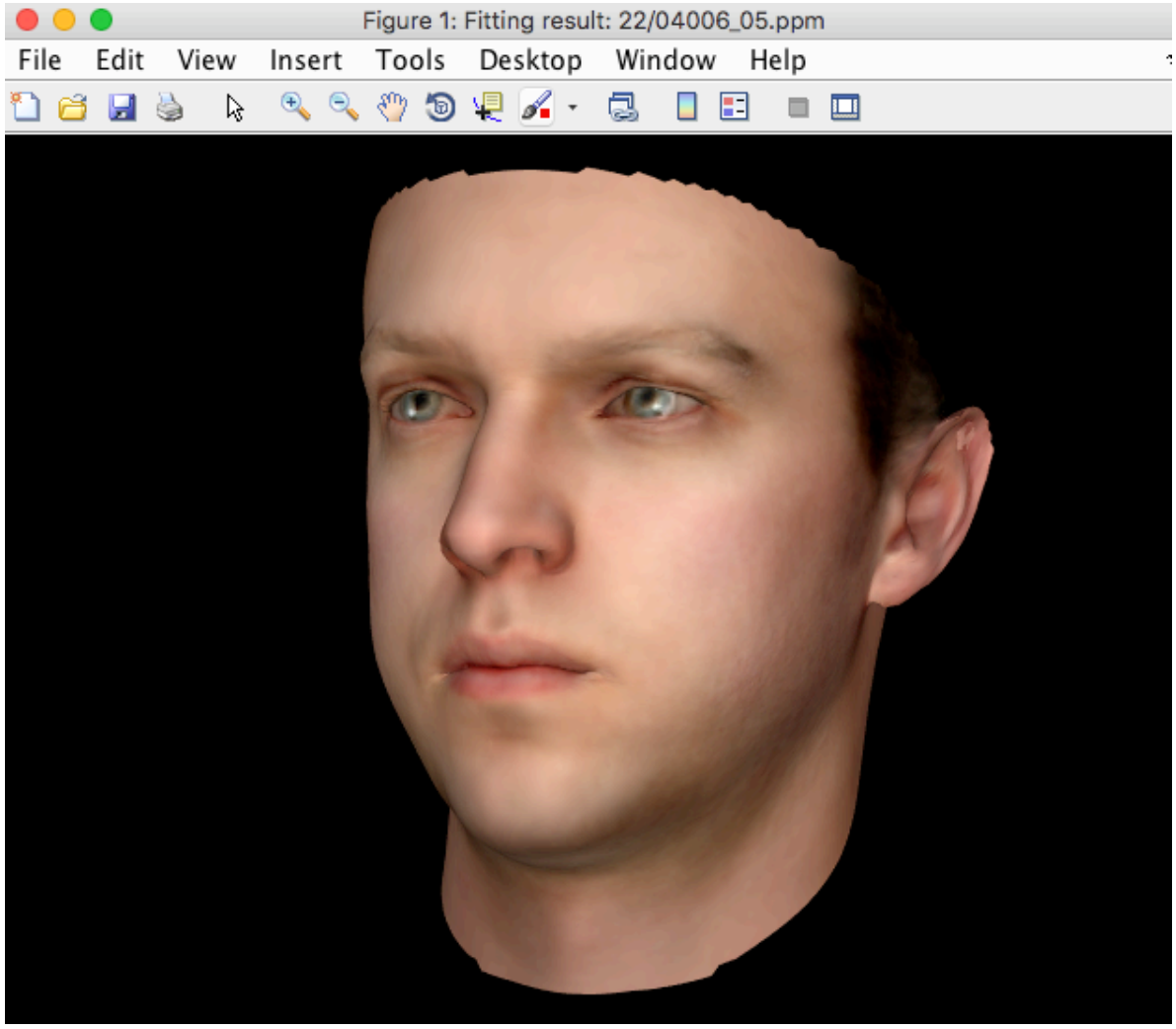
For Intelligent computation
Artificial Intelligent

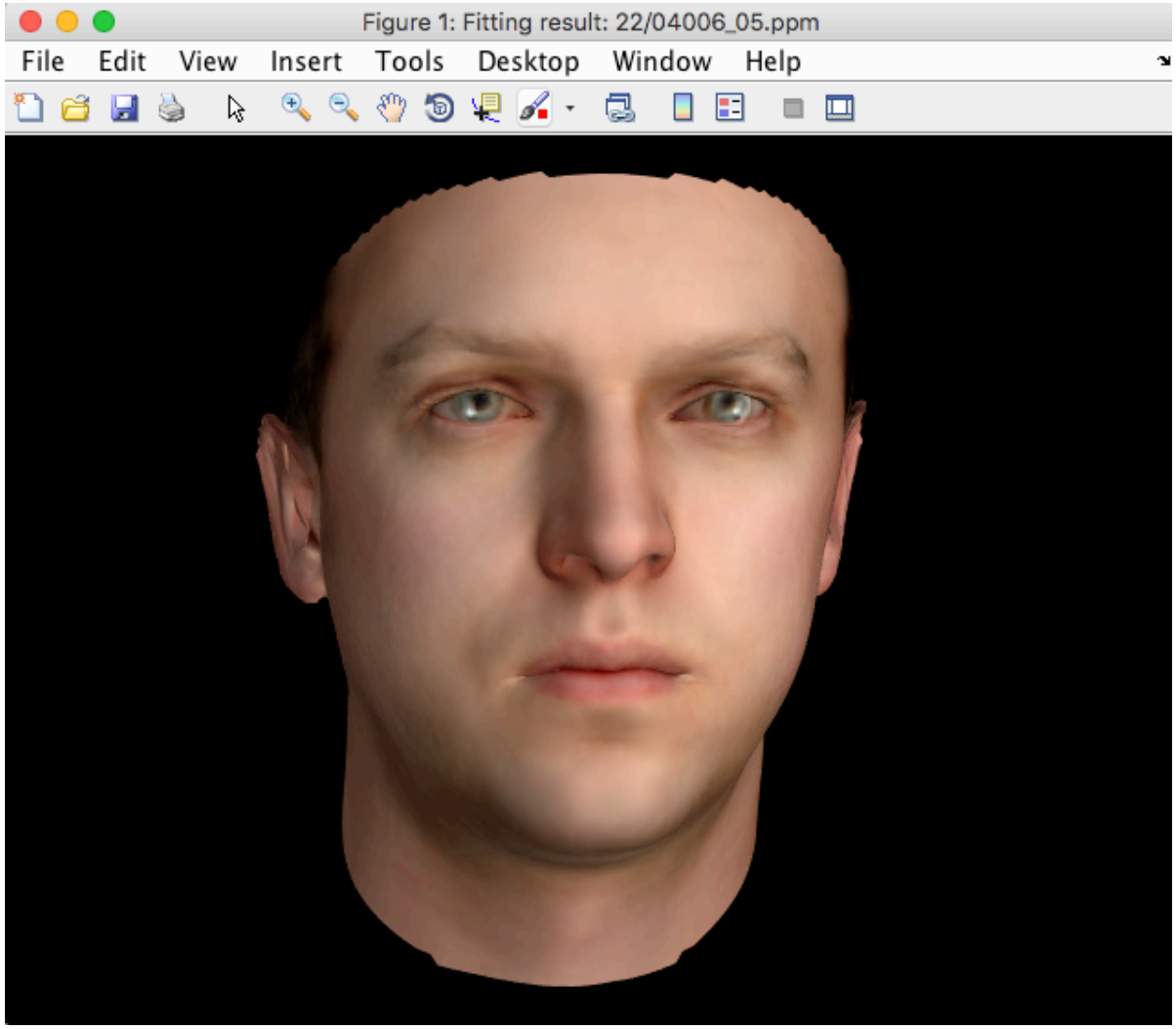
# Mathematical computation for 3D face reconstruction

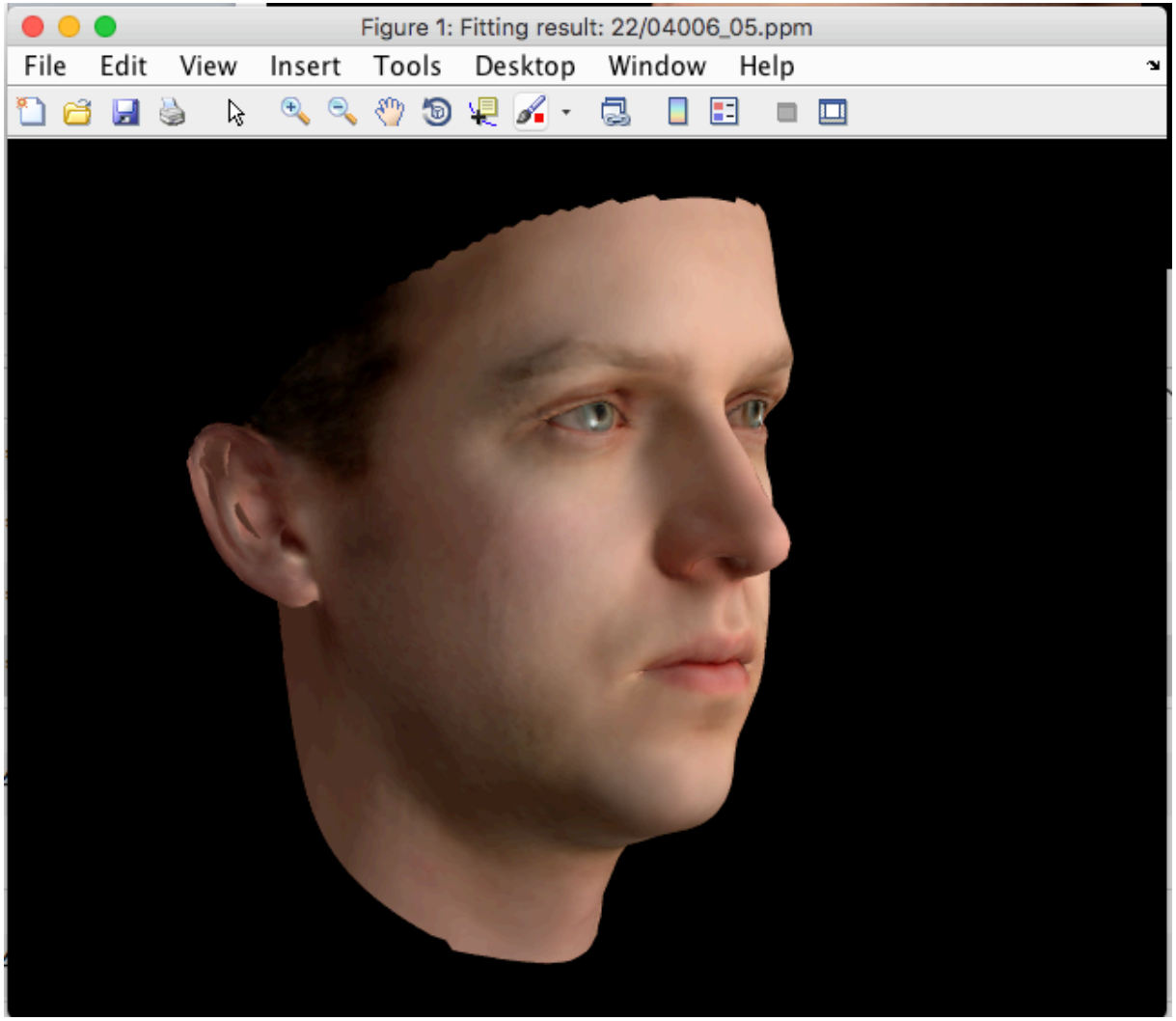
Matlab Software




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

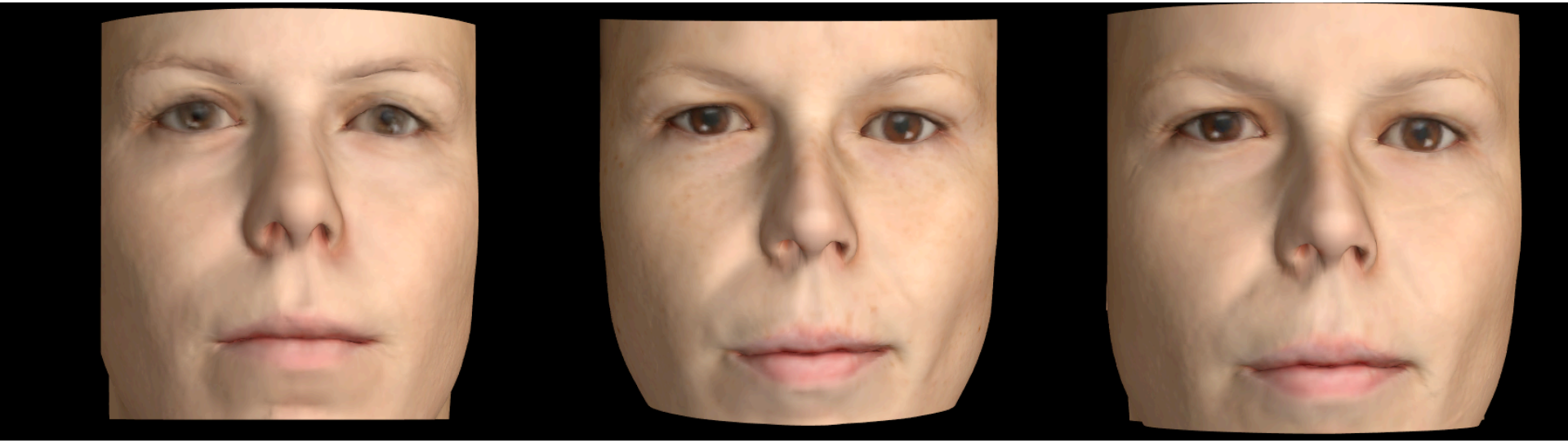

Figure 1: Fitting result: 22/04006_05.ppm
File Edit View Insert Tools Desktop Window Help

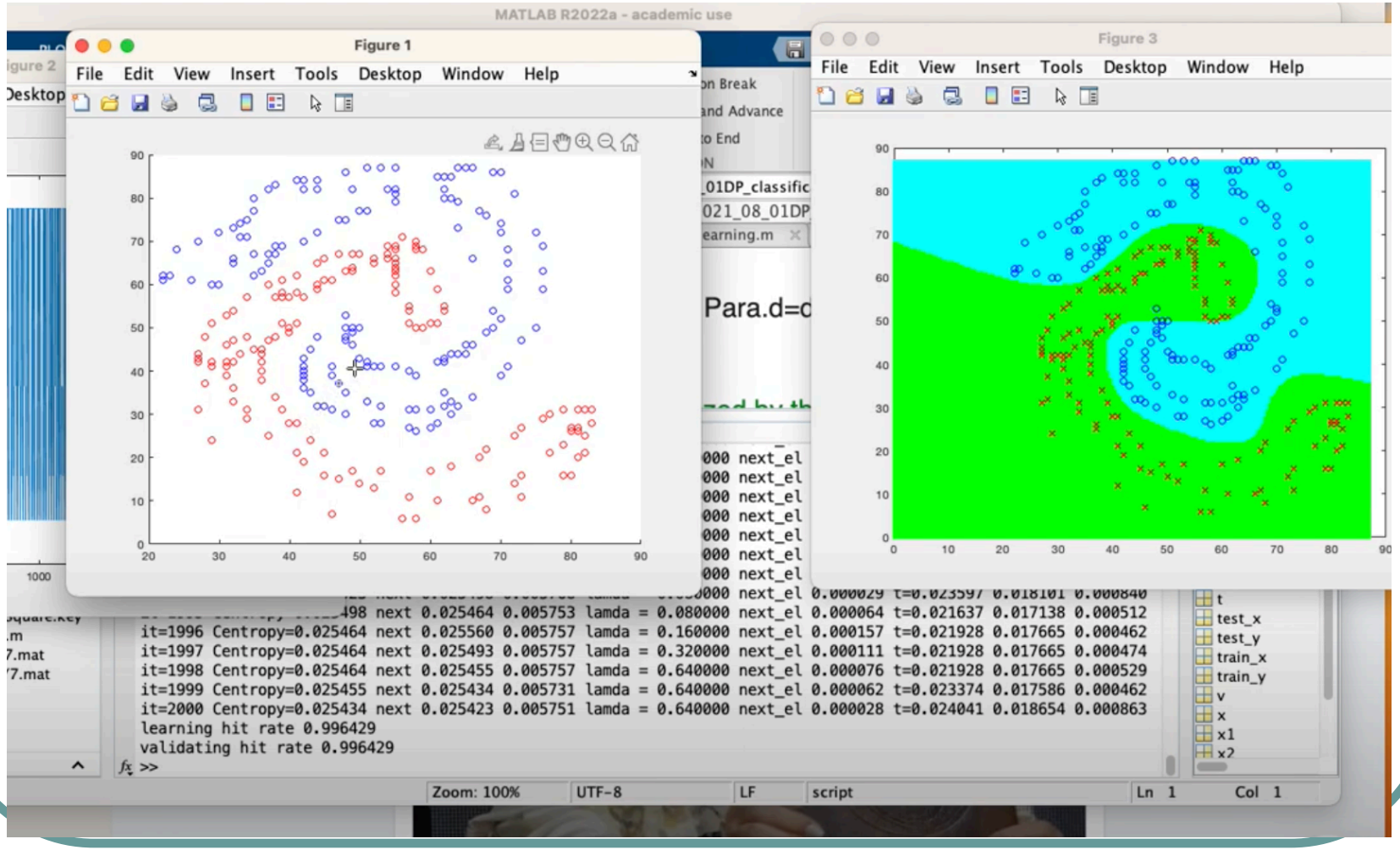

Figure 1: Fitting result: 22/04006_05.ppm
File
Edit
View
Insert
Tools
Desktop
Window
Help

Linear combination of templates

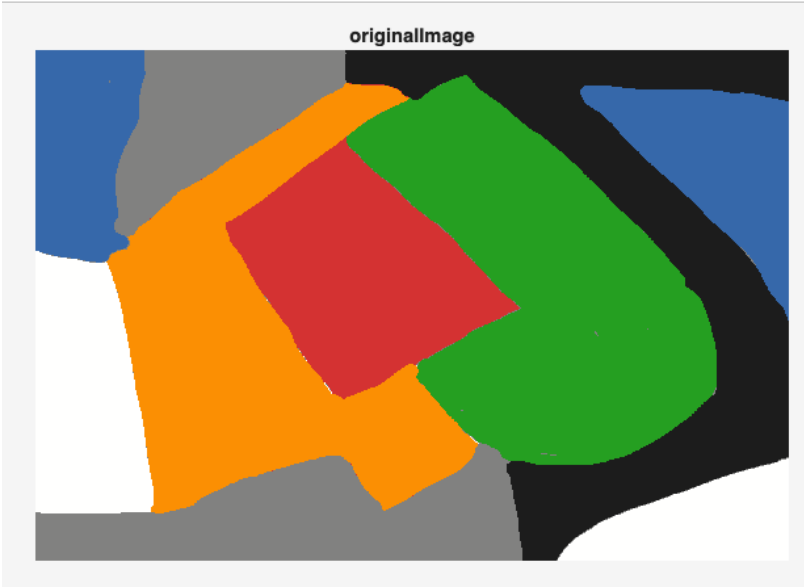
Original face

Nonlinear combination of templates




MATLAB R2022a - academic use
Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 3
File Edit View Insert Tools Desktop Window Help
it=1996 Centropy=0.025464 next 0.025560 0.005757 lamda = 0.160000 next_el 0.000157 t=0.021928 0.017665 0.000462
it=1997 Centropy=0.025464 next 0.025493 0.005757 lamda = 0.320000 next_el 0.000111 t=0.021928 0.017665 0.000474
it=1998 Centropy=0.025464 next 0.025455 0.005757 lamda = 0.640000 next_el 0.000076 t=0.021928 0.017665 0.000529
it=1999 Centropy=0.025455 next 0.025434 0.005731 lamda = 0.640000 next_el 0.000062 t=0.023374 0.017586 0.000462
it=2000 Centropy=0.025434 next 0.025423 0.005751 lamda = 0.640000 next_el 0.000028 t=0.024041 0.018654 0.000863
learning hit rate 0.996429
validating hit rate 0.996429
>>
Zoom: 100%
UTF-8
LF
script
Ln 1
Col 1

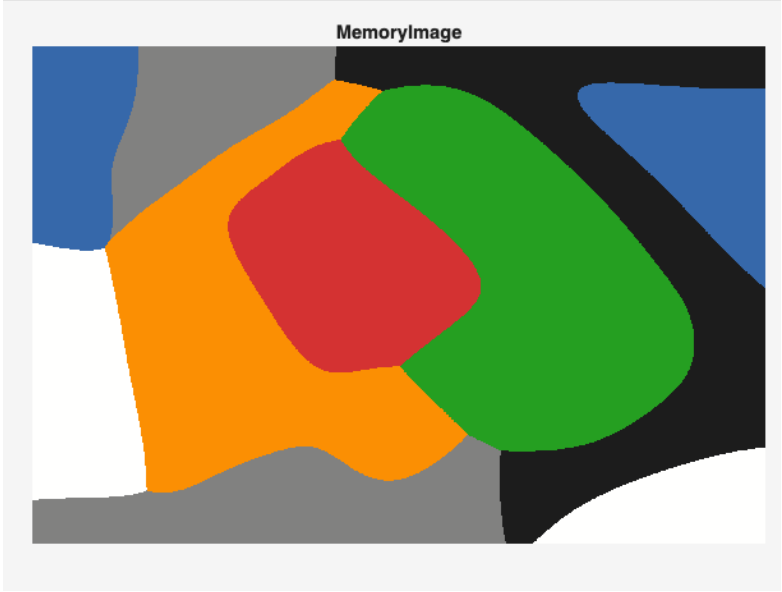
# Memory encoding and memory size reduction




1345

1968

1345*1968*3 bytes




Pixel location → 30 Neurons → 30 Neurons → 7 Neurons

1237 parameters = 1237 * 4 bytes

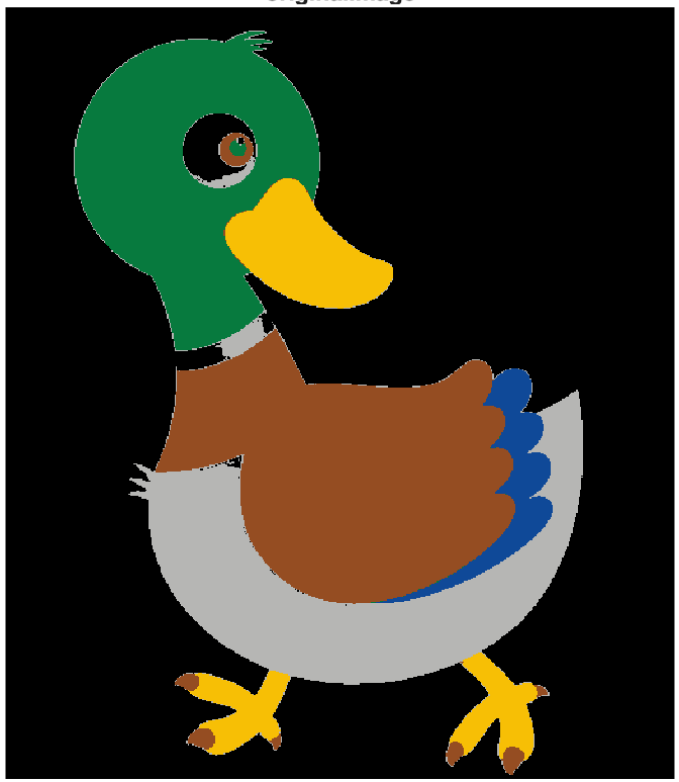
```
Command Window
K>> 1237 *  4 /(1345*1968*3)

ans =

   6.2310e-04
```

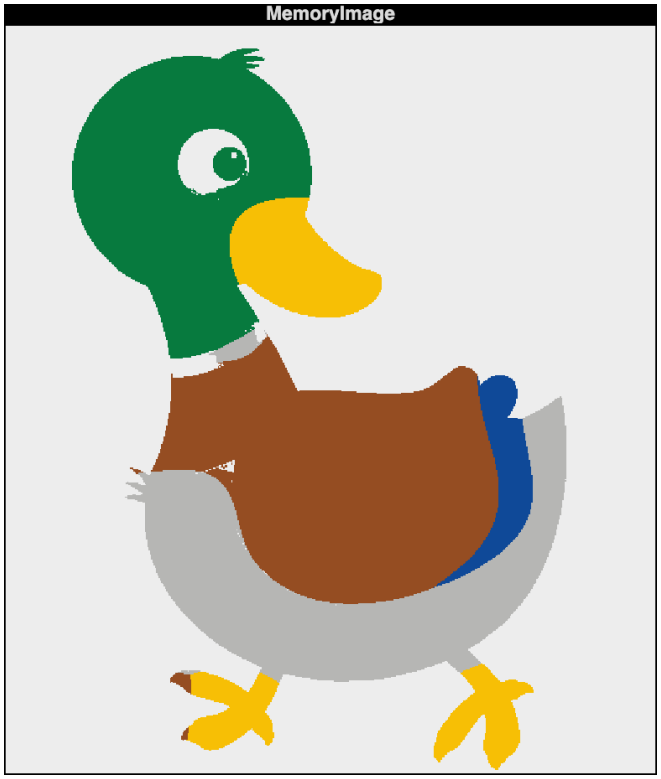
597

originalImage



518
597*518*3 bytes

MemoryImage



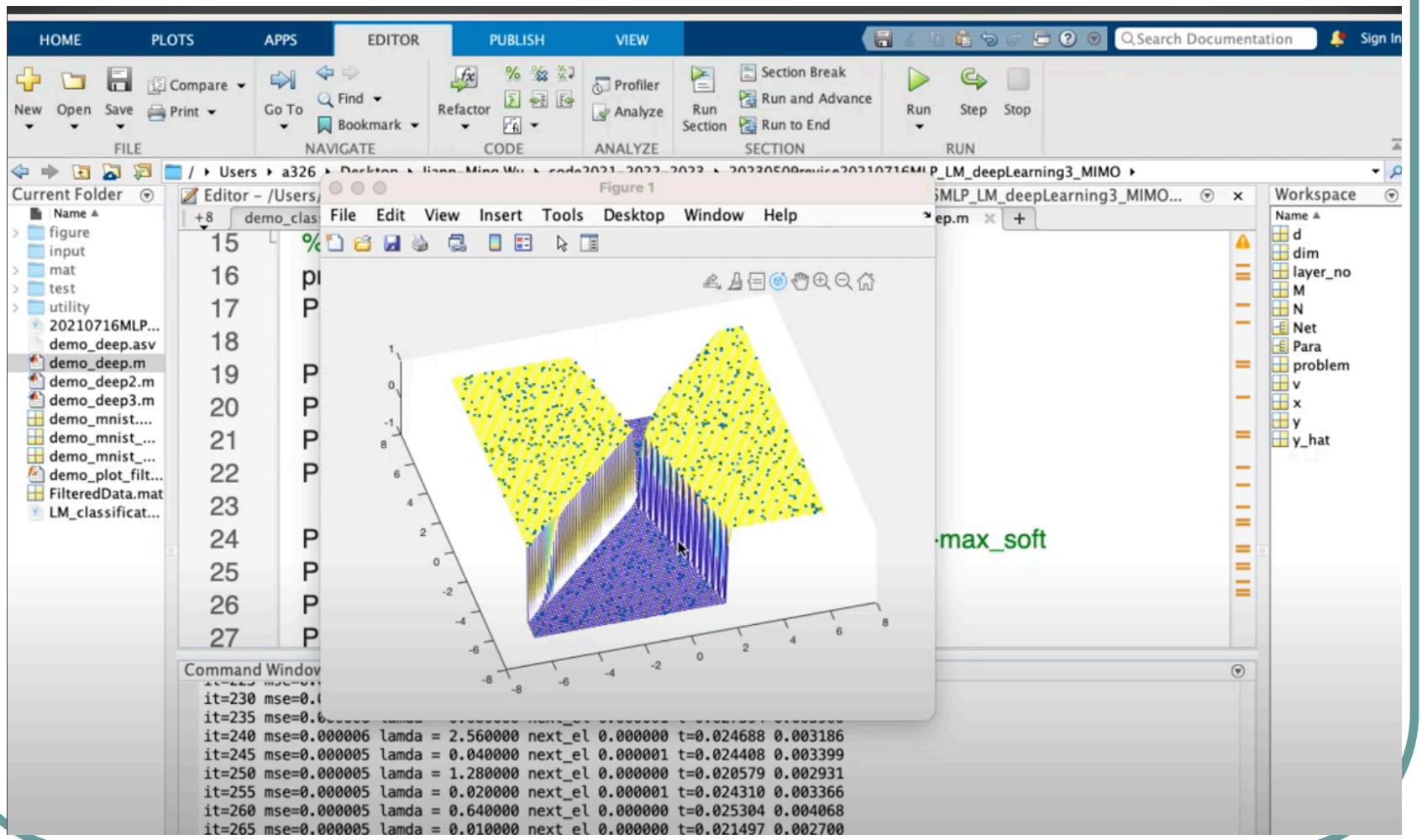
Pixel location → 200 Neurons → 40 Neurons → 7 Neurons

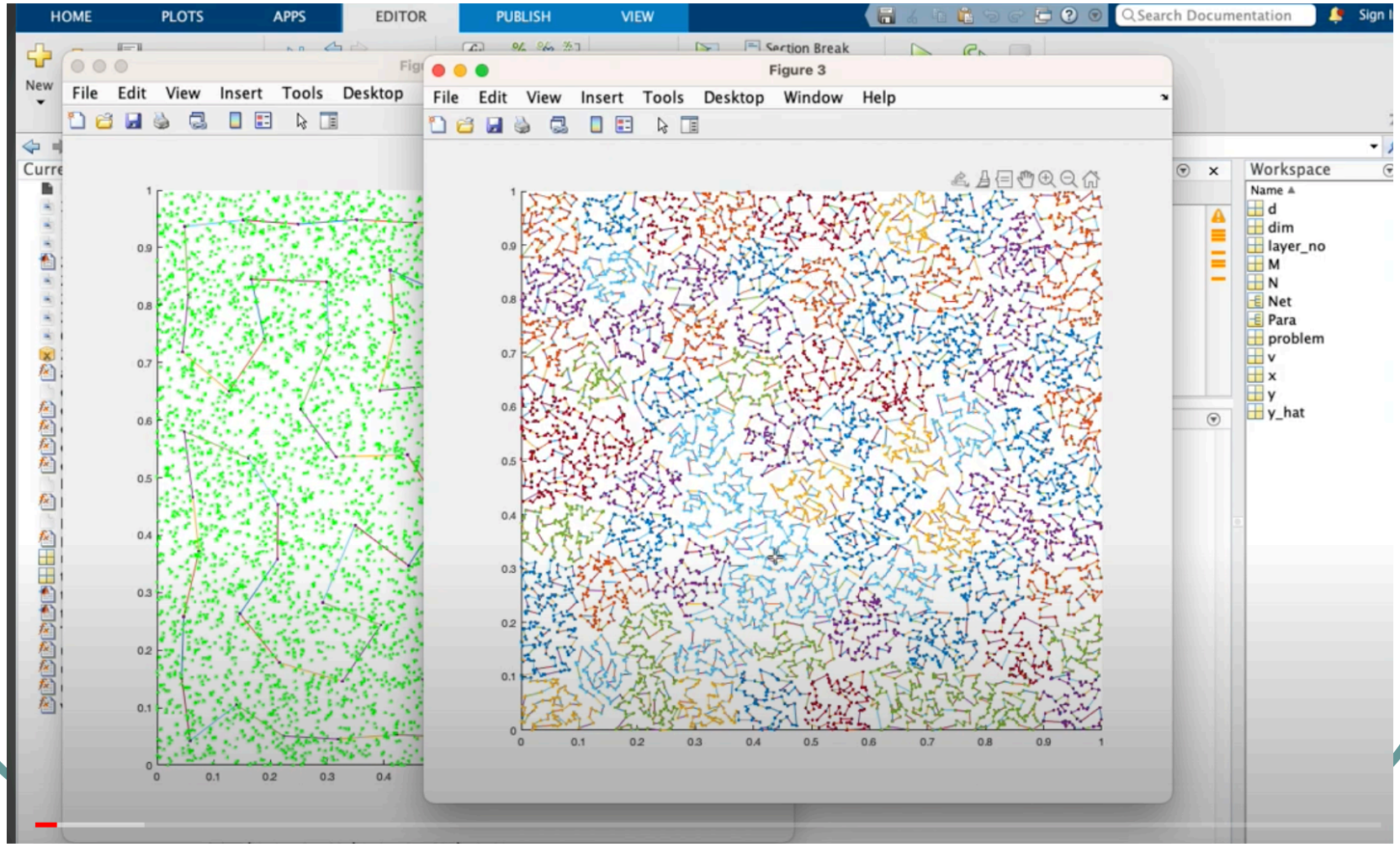
8927 parameters = 8927 * 4 bytes

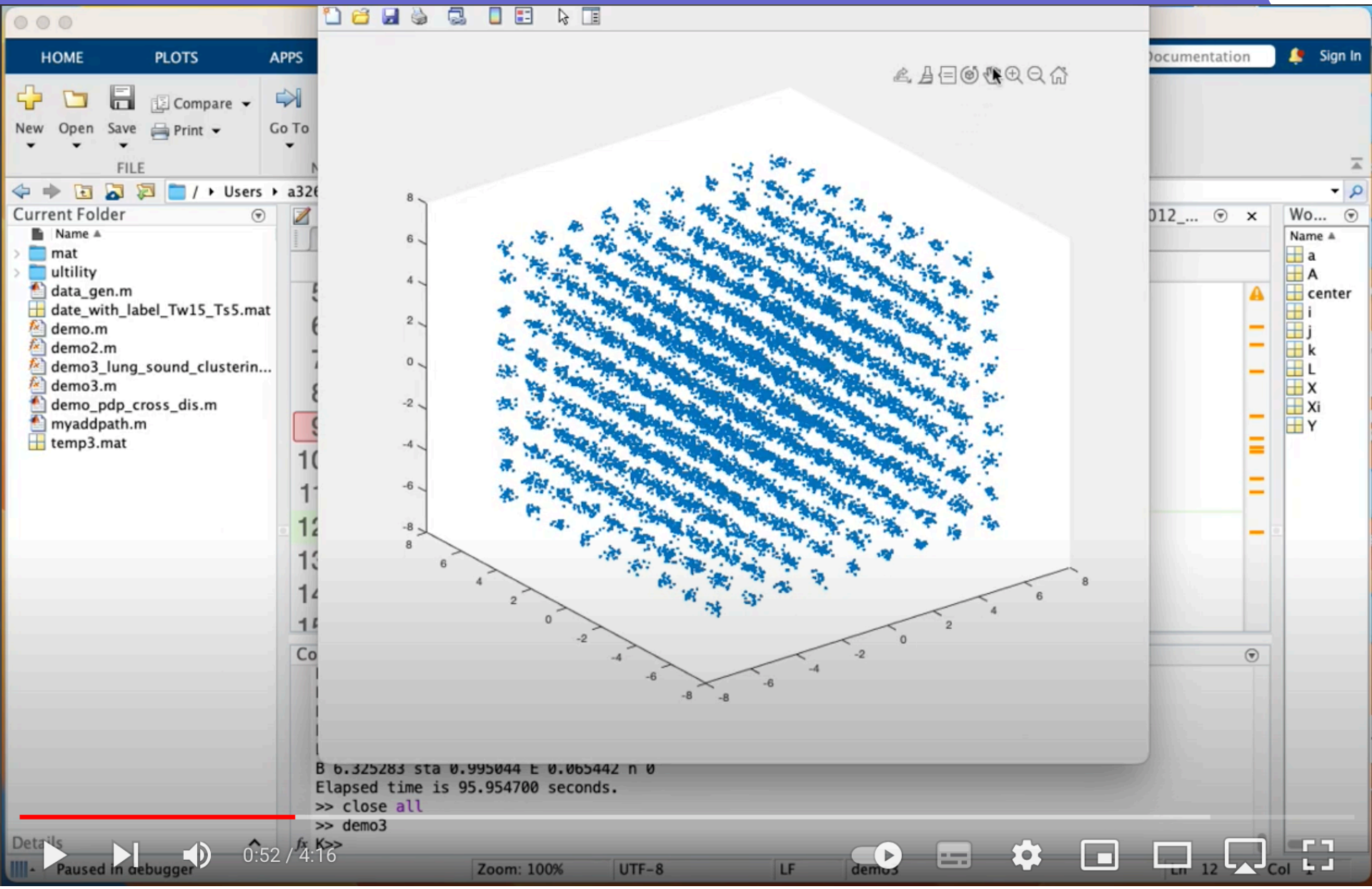
```
K>> 8927 * 4 /(597*518*3)

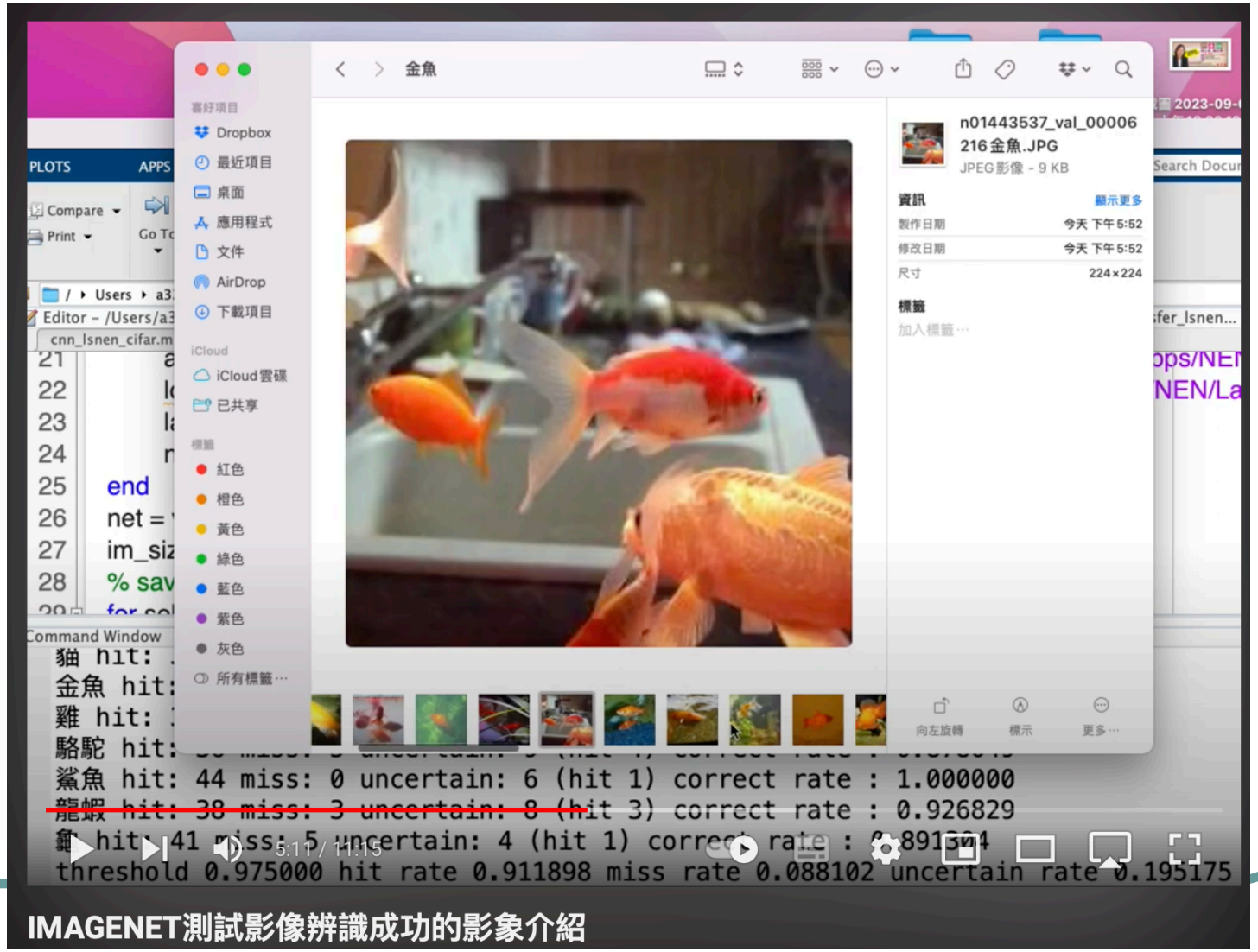
ans =

    0.0385
```

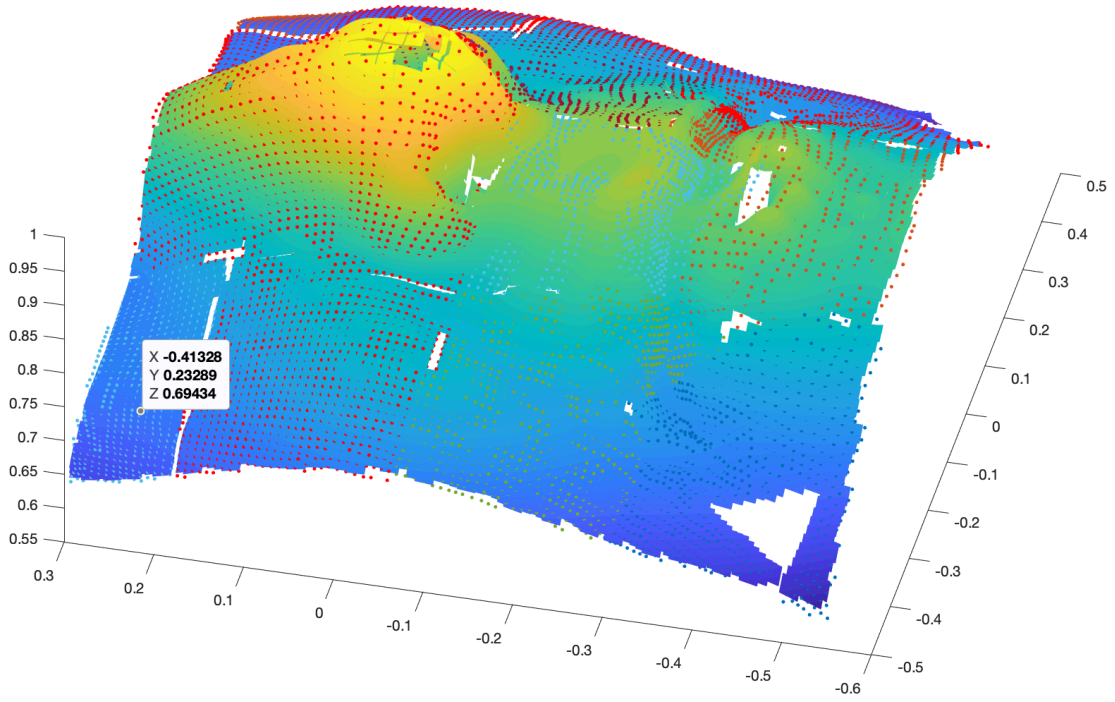

HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Compare
Print
Go To
Find
Bookmark
Refactor
Profiler
Analyze
Run Section
Section Break
Run and Advance
Run to End
Run
Step
Stop
FILE
NAVIGATE
CODE
ANALYZE
SECTION
RUN
Current Folder
Name
figure
input
mat
test
utility
20210716MLP...
demo_deep.asv
demo_deep.m
demo_deep2.m
demo_deep3.m
demo_mnist....
demo_mnist_...
demo_mnist_...
demo_plot_filt...
FilteredData.mat
LM_classificat...
Figure 1
File Edit View Insert Tools Desktop Window Help
Workspace
Name
d
dim
layer_no
M
N
Net
Para
problem
v
x
y
y_hat
Command Window
it=230 mse=0.0
it=235 mse=0.6
it=240 mse=0.000006 lamda = 2.560000 next_el 0.000000 t=0.024688 0.003186
it=245 mse=0.000005 lamda = 0.040000 next_el 0.000001 t=0.024408 0.003399
it=250 mse=0.000005 lamda = 1.280000 next_el 0.000000 t=0.020579 0.002931
it=255 mse=0.000005 lamda = 0.020000 next_el 0.000001 t=0.024310 0.003366
it=260 mse=0.000005 lamda = 0.640000 next_el 0.000000 t=0.025304 0.004068
it=265 mse=0.000005 lamda = 0.010000 next_el 0.000000 t=0.021497 0.002700


HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New
File Edit View Insert Tools Desktop
Figure 3
File Edit View Insert Tools Desktop Window Help
Workspace
Name
d
dim
layer_no
M
N
Net
Para
problem
v
x
y
y_hat


HOME
PLOTS
APPS
New
Open
Save
Compare
Print
Go To
FILE
Current Folder
Name
mat
ultility
data_gen.m
date_with_label_Tw15_Ts5.mat
demo.m
demo2.m
demo3_lung_sound_clusterin...
demo3.m
demo_pdp_cross_dis.m
myaddpath.m
temp3.mat
Name
a
A
center
i
j
k
L
X
Xi
Y
B 6.325283 sta 0.995044 E 0.065442 n 0
Elapsed time is 95.954700 seconds.
>> close all
>> demo3
K>>
Paused in debugger
Zoom: 100%
UTF-8
LF
0:52 / 4:16




IMAGENET測試影像辨識成功的影象介紹




圖九b、完整的臉頰、鼻子、嘴、下巴3D人臉平面幾何重構與近似

Research Insights

# MatConvNet Deep Learning and iOS Mobile App Design for Pattern Recognition

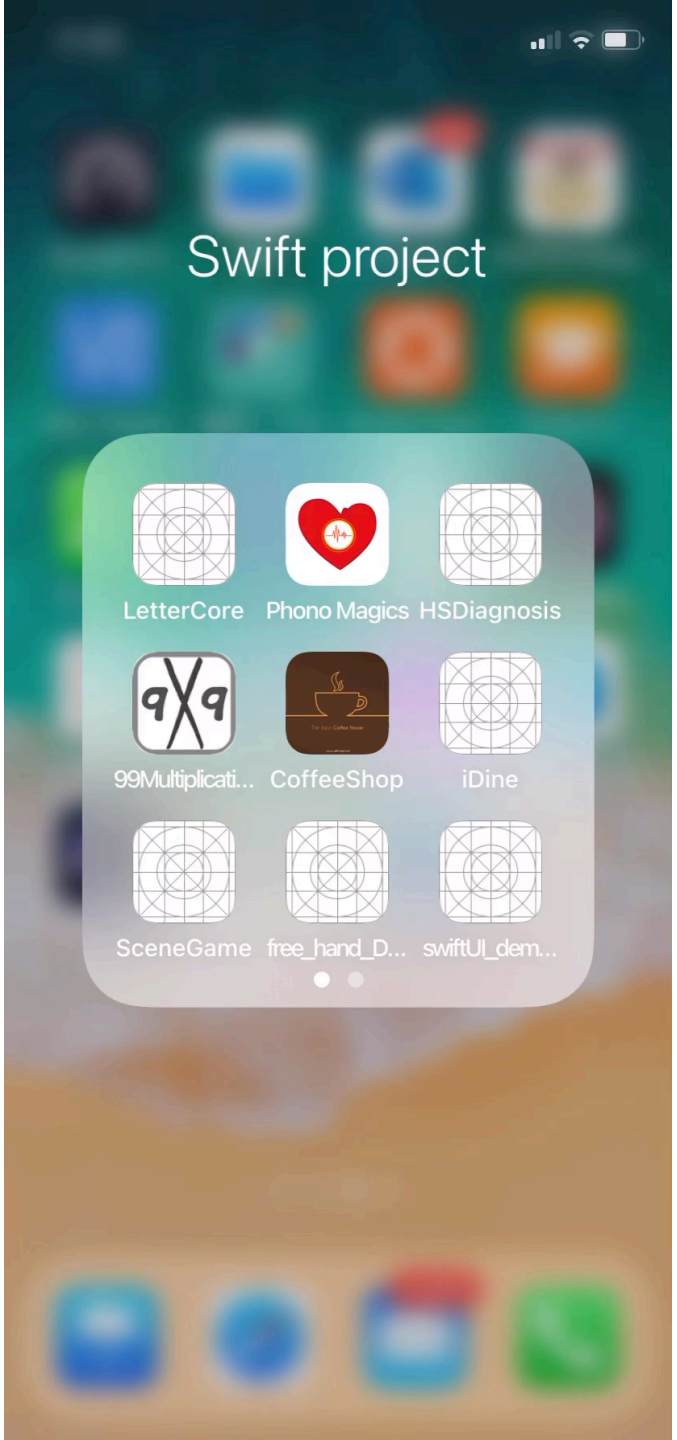
## Emerging Research and Opportunities

國立東華大學圖書館
W0073196



Jiann-Ming Wu and Chao-Yuan Tien

IGI Global
DISSEMINATOR OF KNOWLEDGE


Swift project
LetterCore
Phono Magics
HSDiagnosis
99Multiplicati...
CoffeeShop
iDine
SceneGame
free_hand_D...
swiftUI_dem...



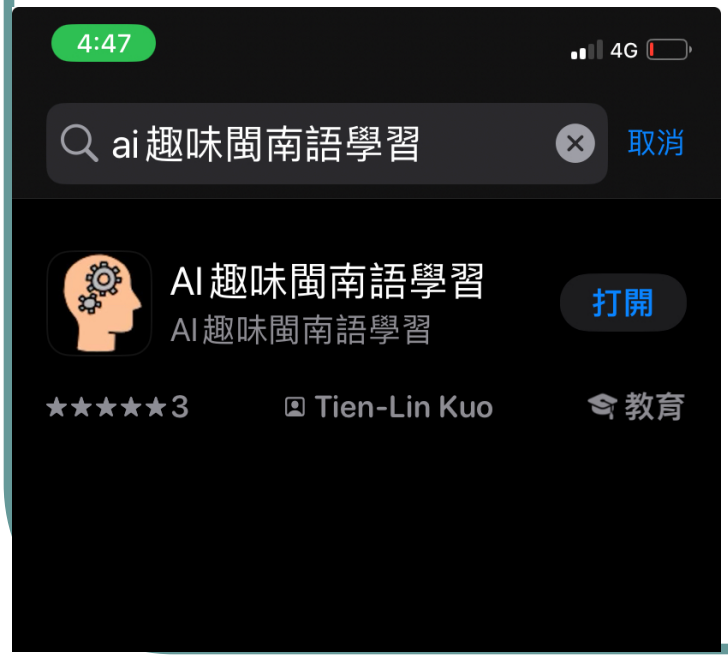
Return

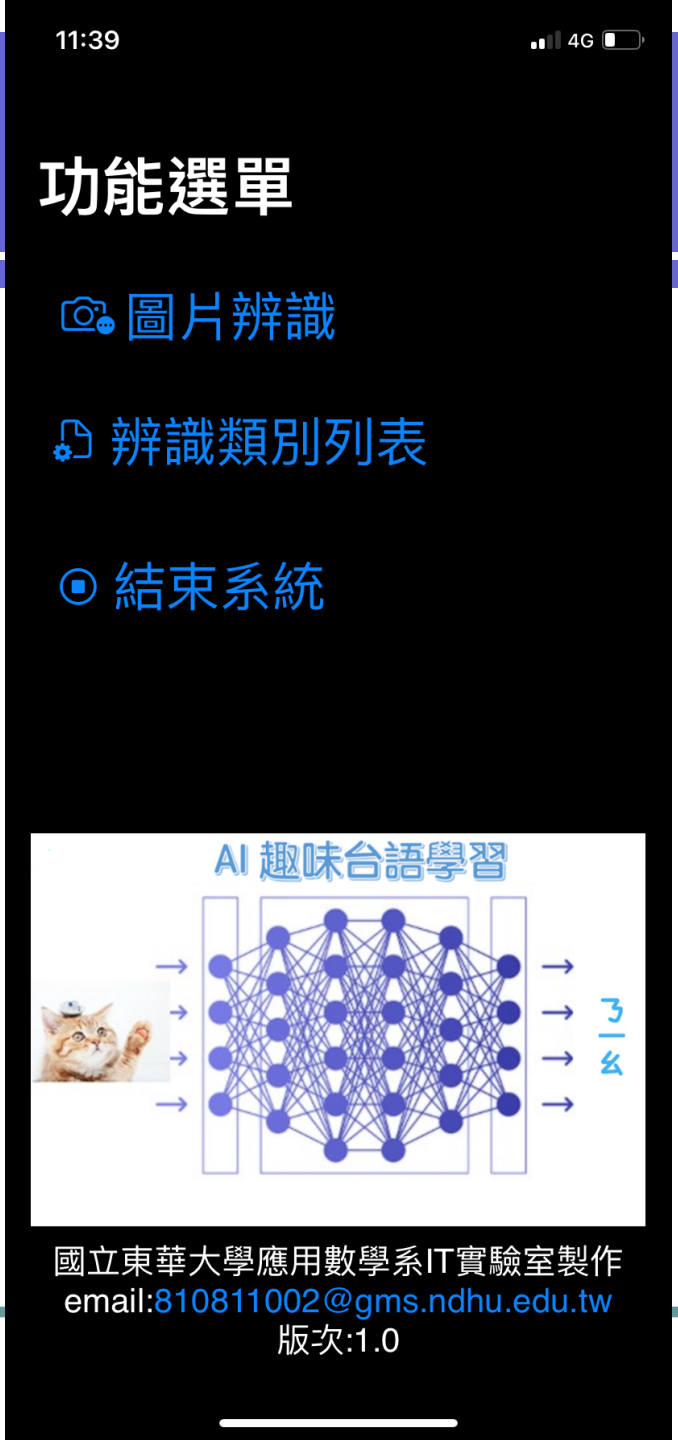
暢遊花蓮


11:39
4G
南澳
9
蘇澳-花蓮
太魯閣
秀林鄉
菲律賓
太平洋公園
壽豐鄉
花蓮縣
萬榮鄉
光復鄉
豐濱鄉
北回歸線
Legal
Maps

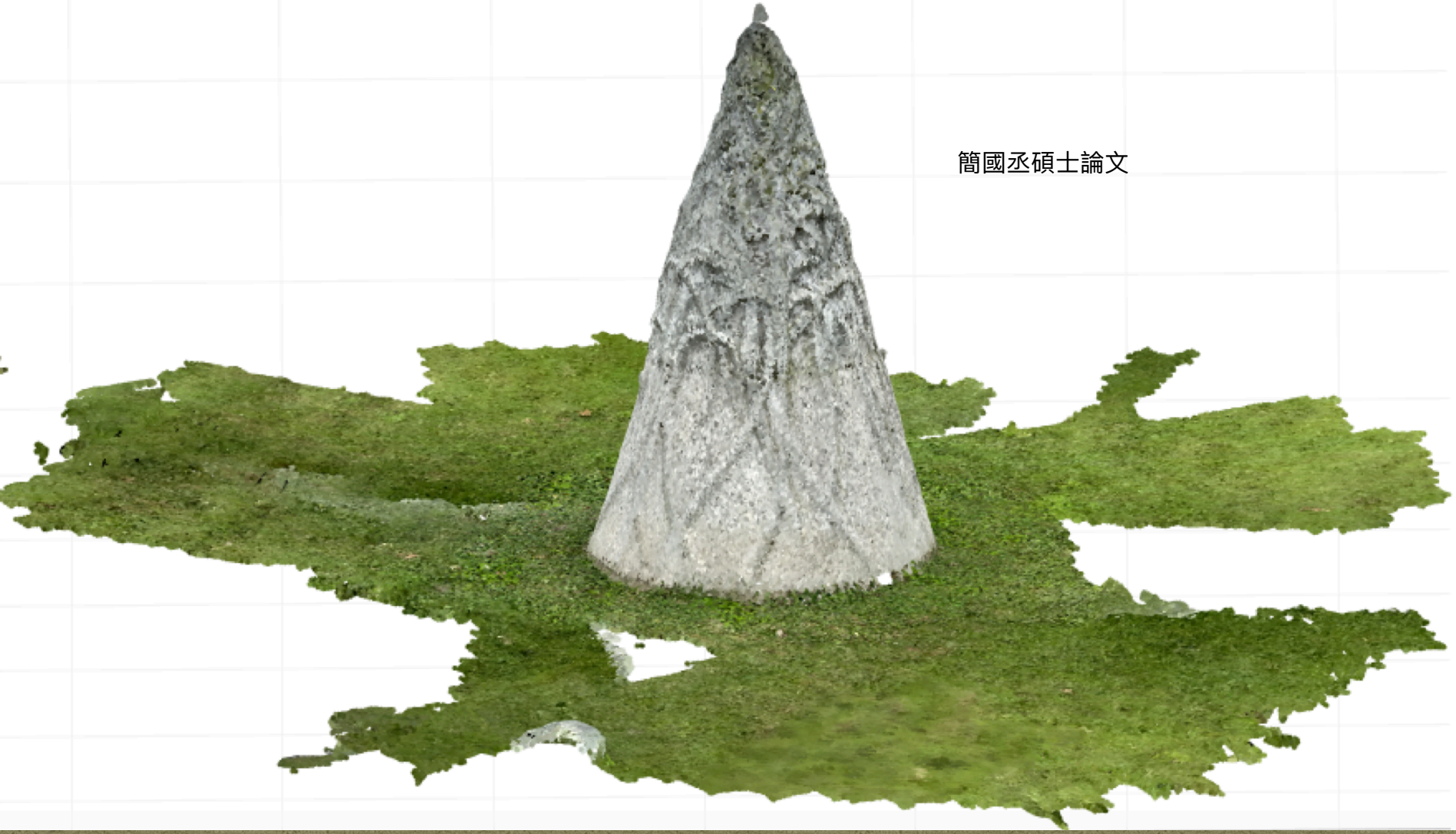

4:48
4G
搜尋
暢遊花蓮
HAO-YUAN HUANG
更新
年齡
4+
歲
類別
旅遊
開發者
HAO-YUAN H
新功能
版本記錄
版本 1.1
3 個月前
新增更清晰易懂的影片說明
預覽
8:42
建立旅遊清單，計算
最佳路徑
8:42
花蓮知名景點
Today
遊戲
App
Arcade
搜尋

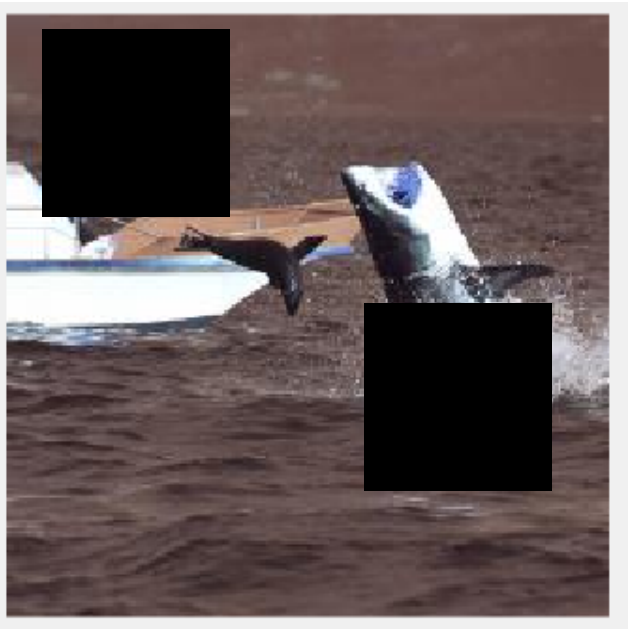
AI趣味閩南語學習




國立東華大學應用數學系IT實驗室製作
email:810811002@gms.ndhu.edu.tw
版次:1.0


簡國丞碩士論文



失去部分影像

回復完整的影像

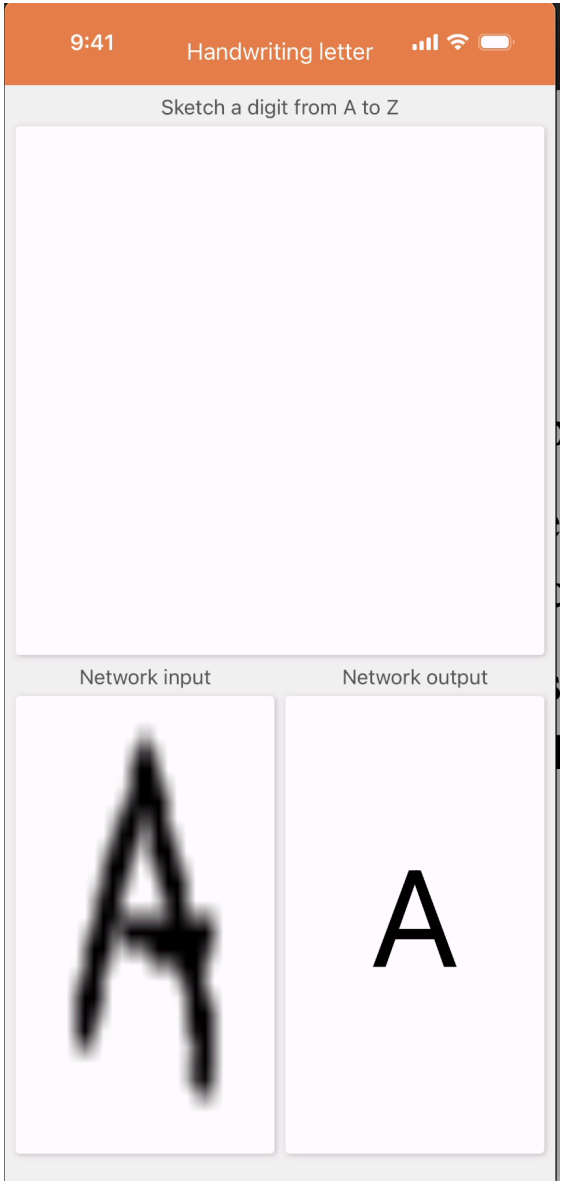


| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | | | 4 | | | | 1 | 6 |
| 2 | | | | 5 | 6 | 7 | | |
| | | | 8 | 7 | 1 | 4 | | 2 |
| 6 | 3 | | | 1 | | | 5 | |
| | | | | 8 | | | | |
| | 7 | | | 3 | | | 4 | 9 |
| 7 | | 6 | 2 | 4 | 8 | | | |
| | | 8 | 3 | 9 | | | | 4 |
| 3 | 9 | | | | 7 | | | 5 |

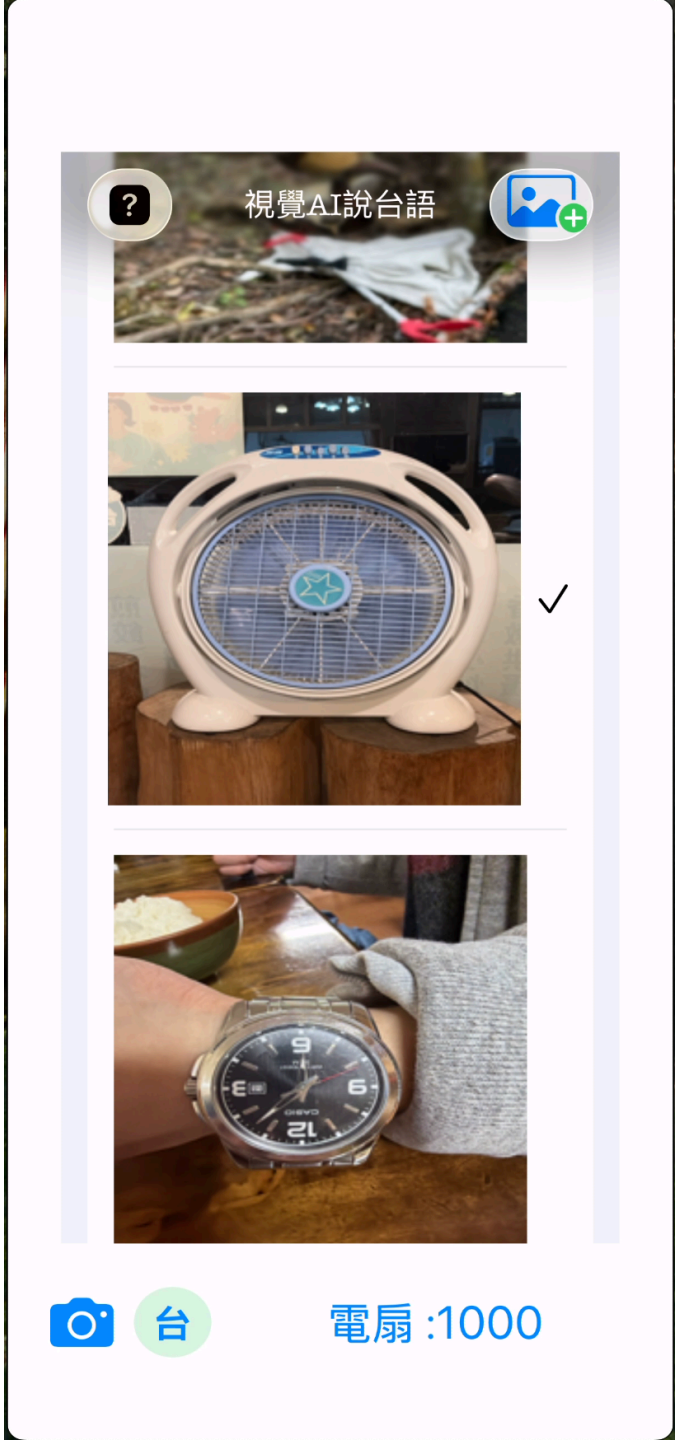
Sudoku puzzle resolution

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | 8 | 7 | 4 | 2 | 3 | 5 | 1 | 6 |
| 2 | 4 | 1 | 9 | 5 | 6 | 7 | 8 | 3 |
| 5 | 6 | 3 | 8 | 7 | 1 | 4 | 9 | 2 |
| 6 | 3 | 9 | 7 | 1 | 4 | 2 | 5 | 8 |
| 4 | 1 | 2 | 5 | 8 | 9 | 3 | 6 | 7 |
| 8 | 7 | 5 | 6 | 3 | 2 | 1 | 4 | 9 |
| 7 | 5 | 6 | 2 | 4 | 8 | 9 | 3 | 1 |
| 1 | 2 | 8 | 3 | 9 | 5 | 6 | 7 | 4 |
| 3 | 9 | 4 | 1 | 6 | 7 | 8 | 2 | 5 |

# Install to iPhone and execute on iPhone



Execute an App: 視覺AI說台語



# Shortest Path Optimization App



**MatConvNet: CNNs for MATLAB**

Obtaining MatConvNet

Documentation

Extensions

Getting started

Use cases

Other information

# MatConvNet: CNNs for MATLAB

Download Code & issues Pre-trained models Discussion forum

**MatConvNet** is a MATLAB toolbox implementing *Convolutional Neural Networks* (CNNs) for computer vision applications. It is simple, efficient, and can run and learn state-of-the-art CNNs. Many pre-trained CNNs for image classification, segmentation, face recognition, and text detection are available.

MathWorks®

# R2024a Release Highlights – MATLAB and Simulink

## Discover What's New

Get more out of MATLAB and Simulink by downloading the latest release.

Download now




LEGALLY USING IMAGES



pandas

# Python Interface

Convert between MATLAB tables and Python Pandas DataFrames

```
patients =

  5×3 table

                Age    Height    Weight
                ___    ______    ______

    Sanchez      38      71       176
    Johnson      43      69       163
    Zhang        38      64       131
    Diaz         40      67       133
    Brown        49      64       119
```

```
>> py.pandas.DataFrame(patients)

ans =

  Python DataFrame with properties:

          T: [1×1 py.pandas.core.frame.DataFrame]
         at: [1×1 py.pandas.core.indexing._AtIndexer]
      attrs: [1×1 py.dict]
       axes: [1×2 py.list]
    columns: [1×1 py.pandas.core.indexes.base.Index]
     dtypes: [1×1 py.pandas.core.series.Series]
      empty: 0
      flags: [1×1 py.pandas.core.flags.Flags]
        iat: [1×1 py.pandas.core.indexing._iAtIndexer]
      index: [1×1 py.pandas.core.indexes.base.Index]
       ndim: [1×1 py.int]
      shape: [1×2 py.tuple]
       size: [1×1 py.int]
     values: [1×1 py.numpy.ndarray]

                 Age  Height  Weight
  RowNameIndex
  Sanchez       38.0    71.0   176.0
  Johnson       43.0    69.0   163.0
  Zhang         38.0    64.0   131.0
  Diaz          40.0    67.0   133.0
  Brown         49.0    64.0   119.0
```

# Major Updates

Computer Vision Toolbox

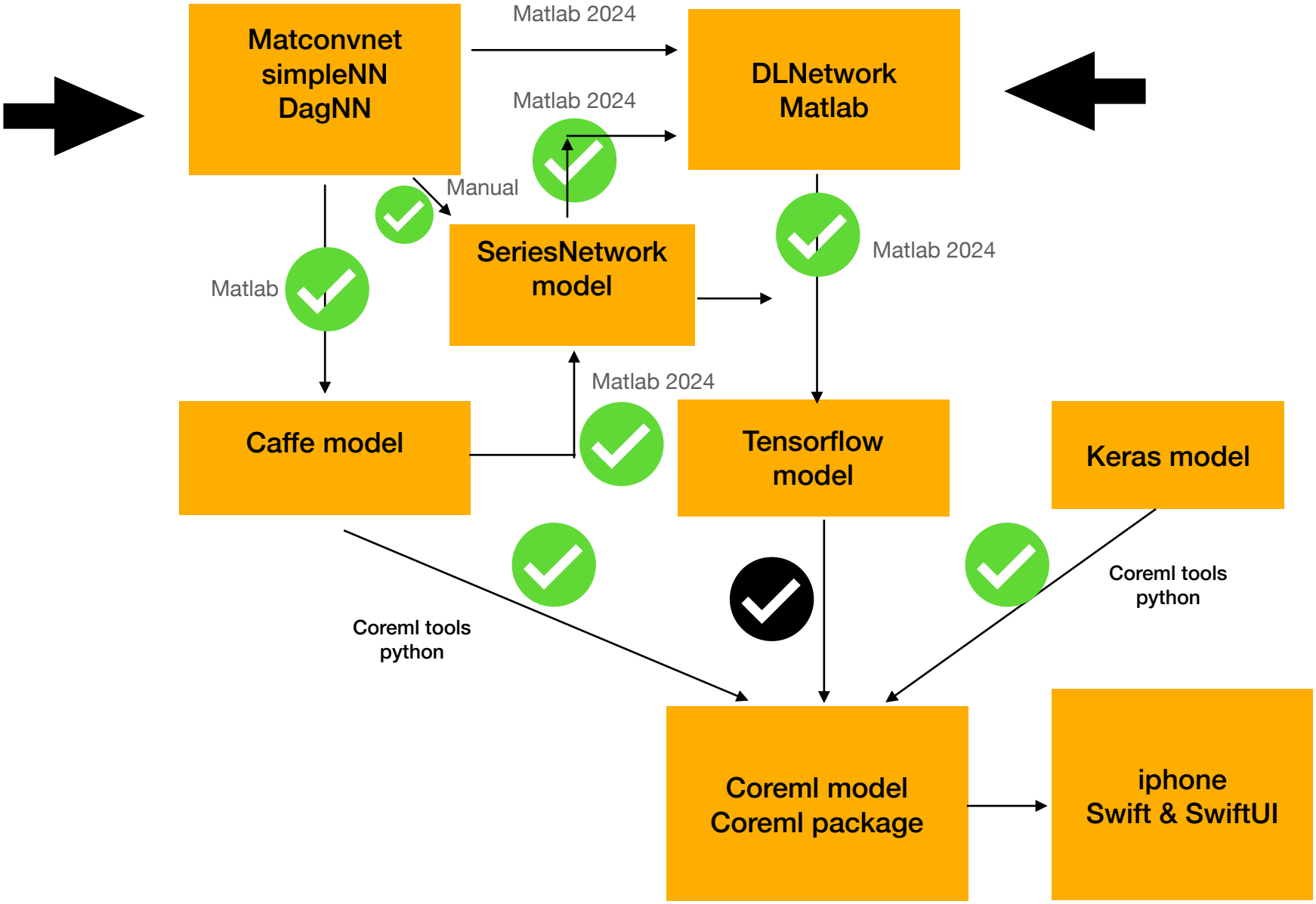
Deep Learning Toolbox

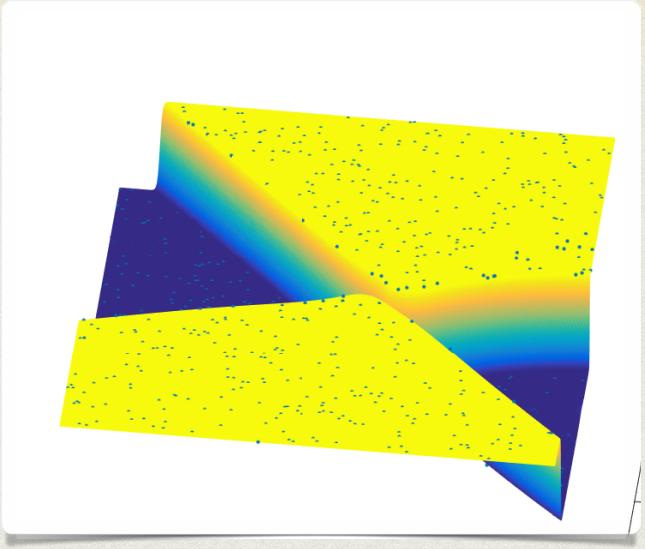
GPU Coder

Instrument Control Toolbox

Satellite Communications Toolbox

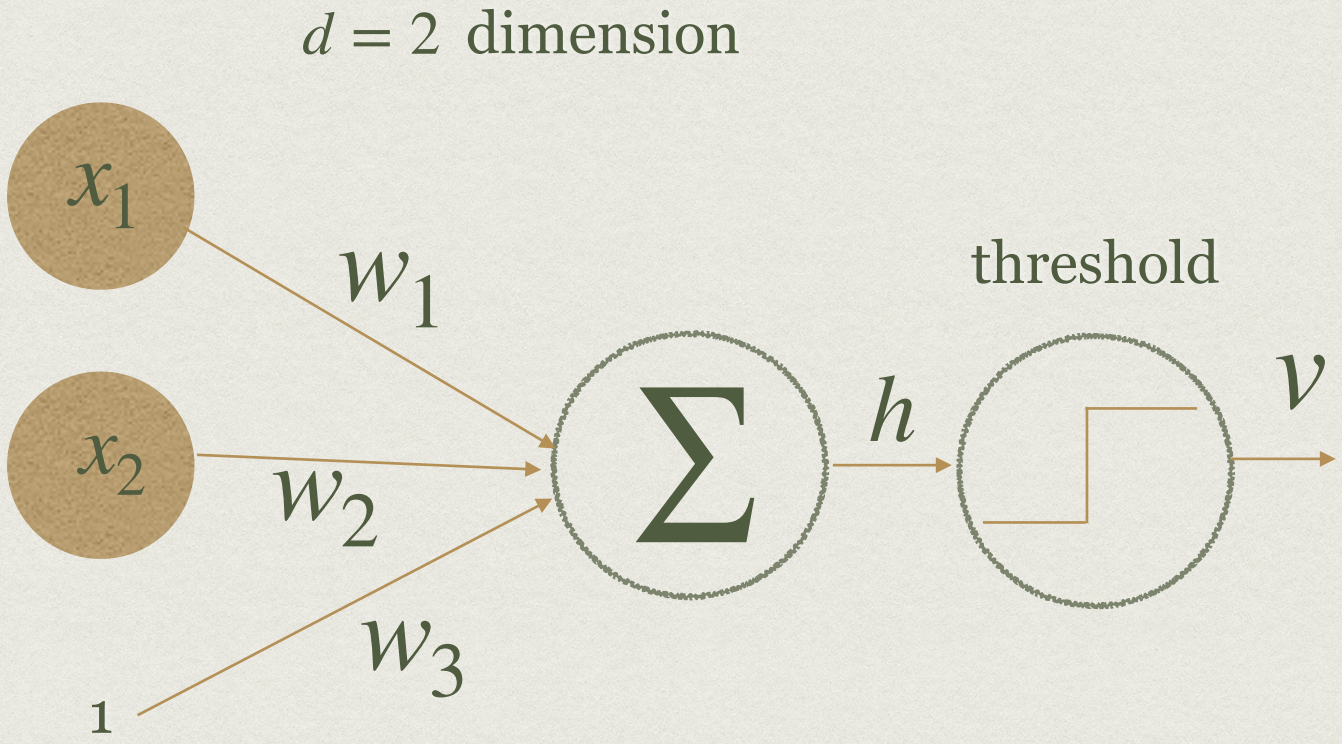
UAV Toolbox


Matconvnet
simpleNN
DagNN
Matlab 2024
DLNetwork
Matlab
Matlab 2024
Manual
SeriesNetwork
model
Matlab
Matlab 2024
Matlab 2024
Caffe model
Tensorflow
model
Keras model
Coreml tools
python
Coreml tools
python
Coreml model
Coreml package
iphone
Swift & SwiftUI



# AI DEEP LEARNING AND APPROXIMATION

# DISCRETE FUNCTION



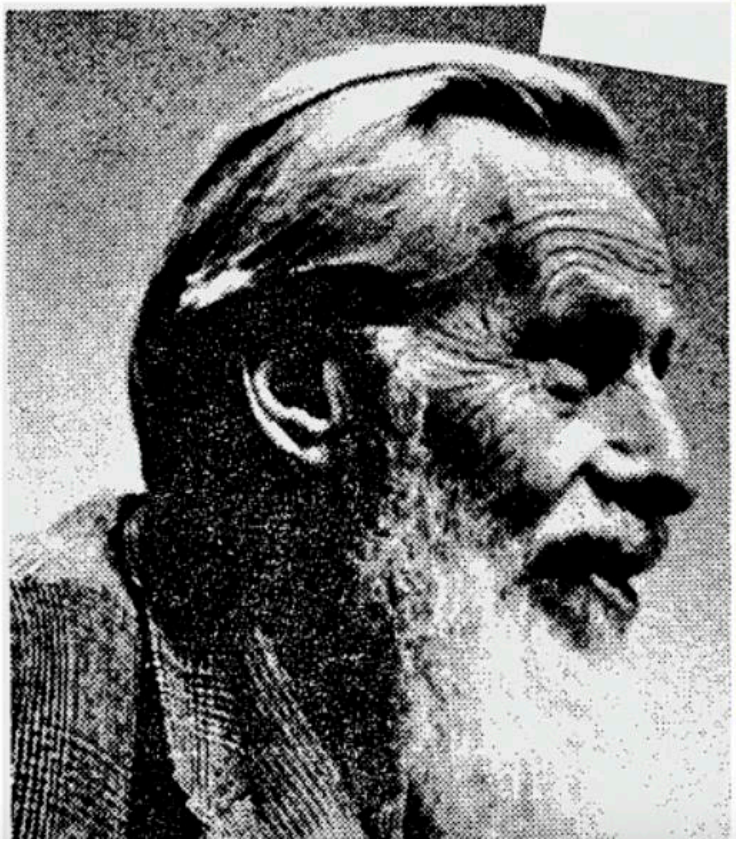
# First artificial neurons: The McCulloch-Pitts model

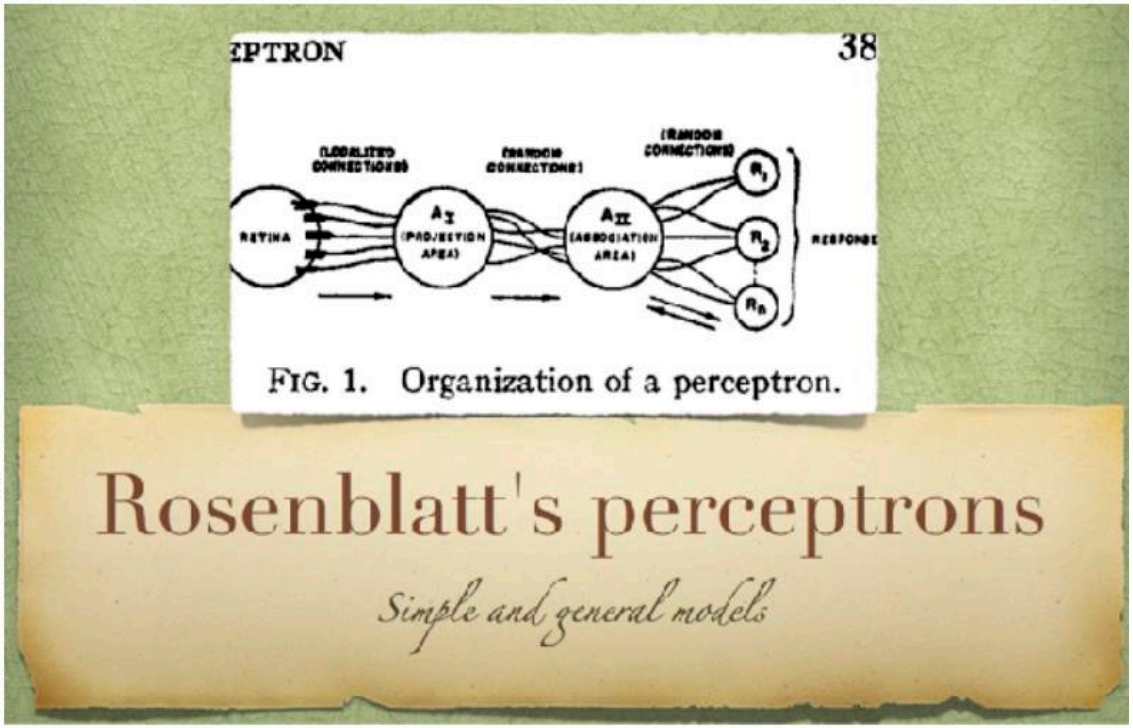
1898 - 1969

ASC FORUM Volume VI, Number 2 -Summer 1974

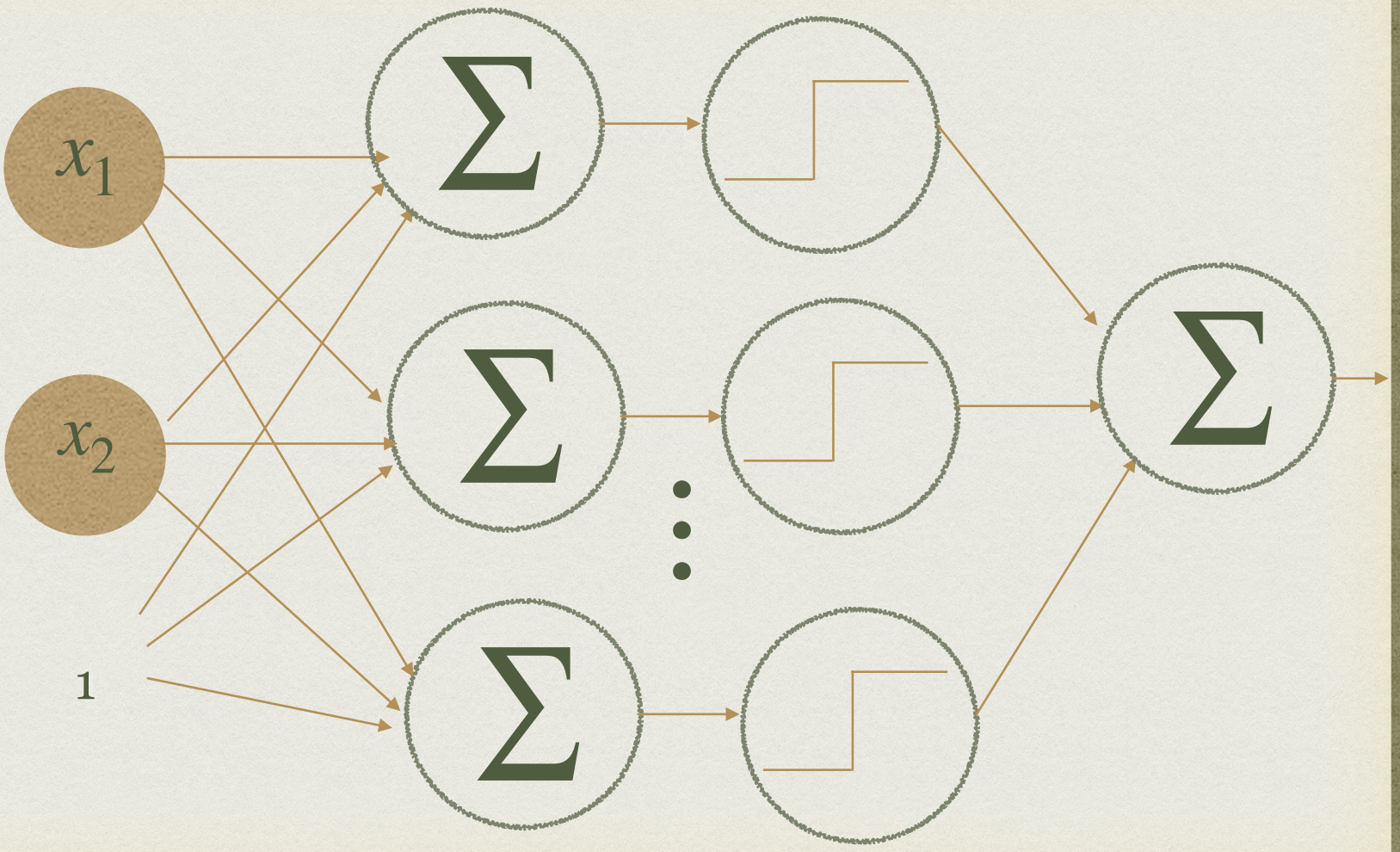
WARREN STURGIS MCCULLOCH

麦卡洛克-皮茨模型





# A DISCRETE NEURAL NETWORK



# APPROXIMATION

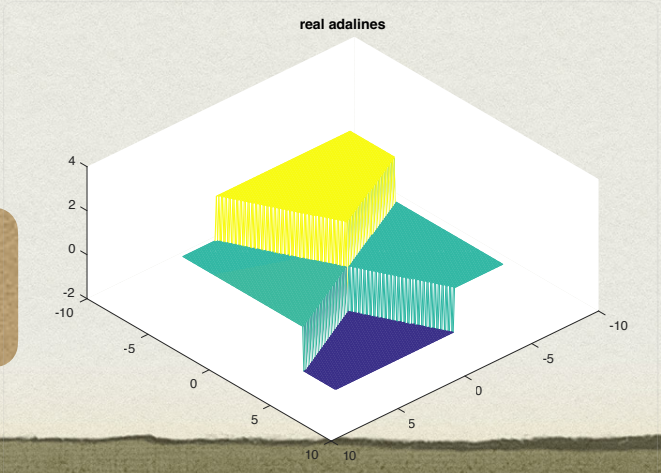
$$y = sign(2x_1 + x_2 - 1) - sign(x_1 - 2x_2 + 1)$$

input
output

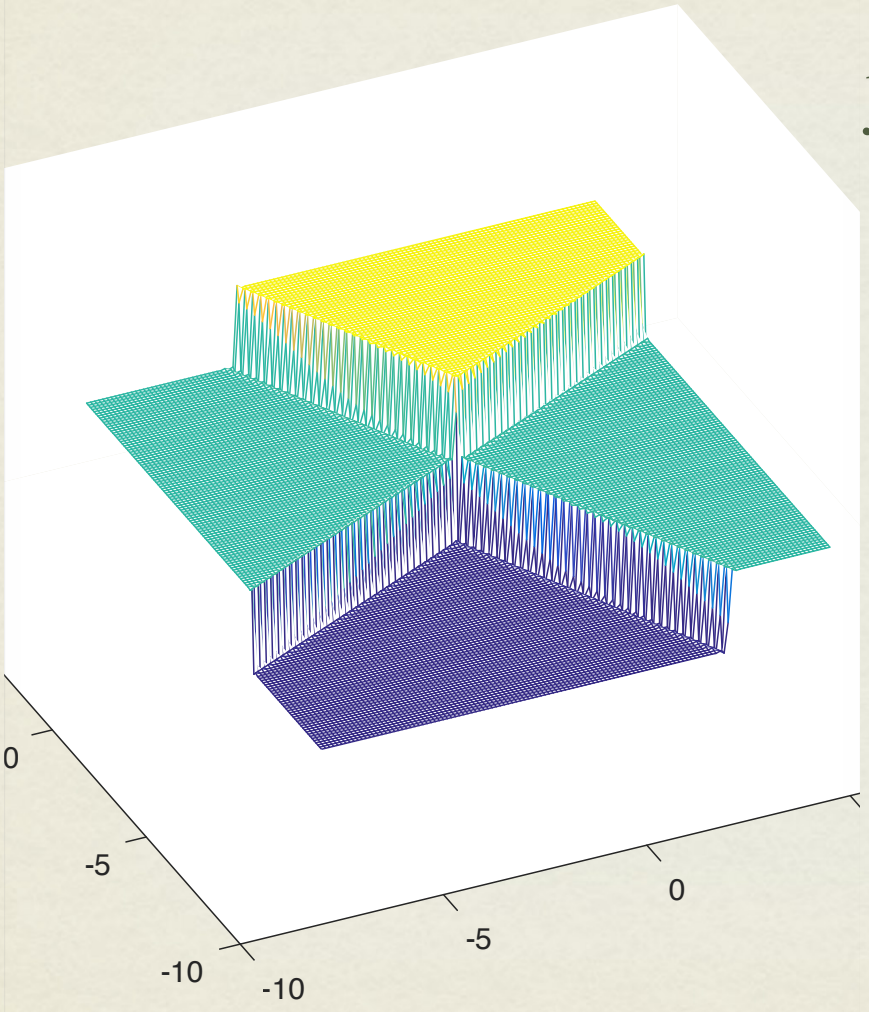
Learning a discrete neural network

optimal interconnections

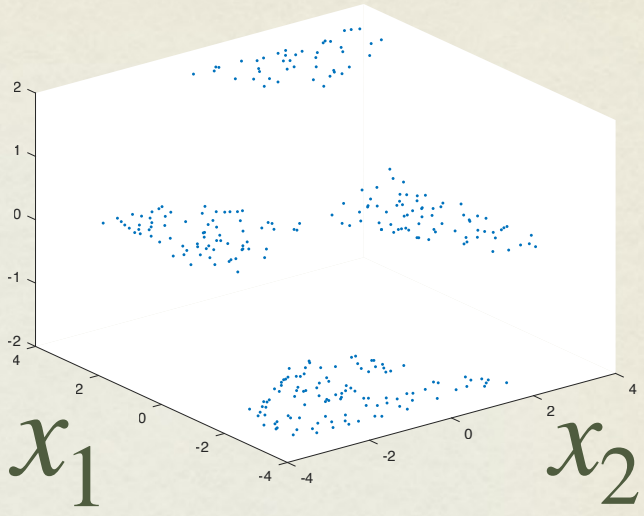
An approximating network



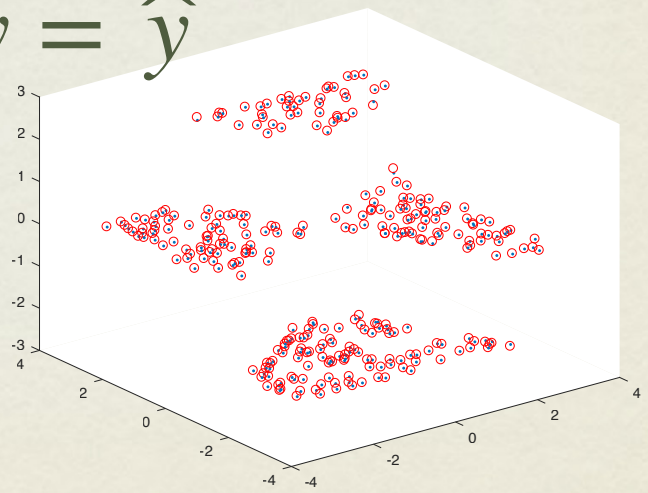
**emulated adalines**

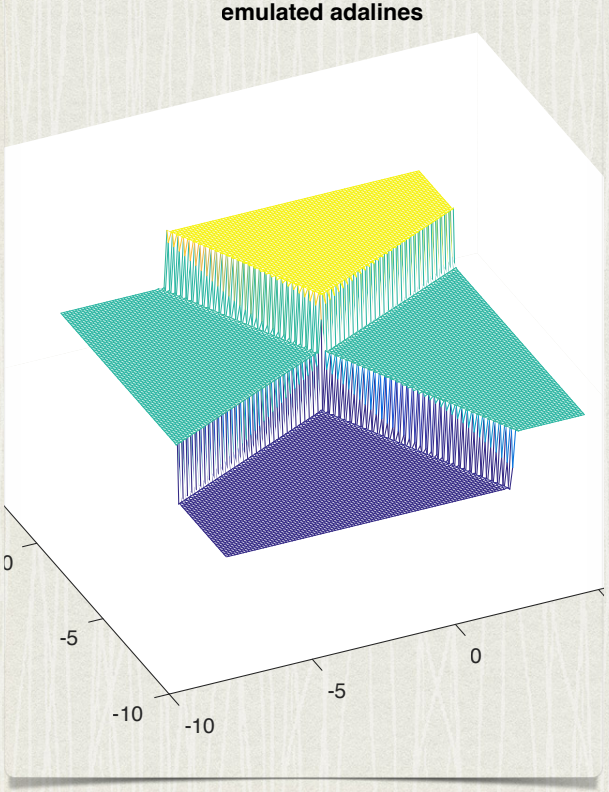


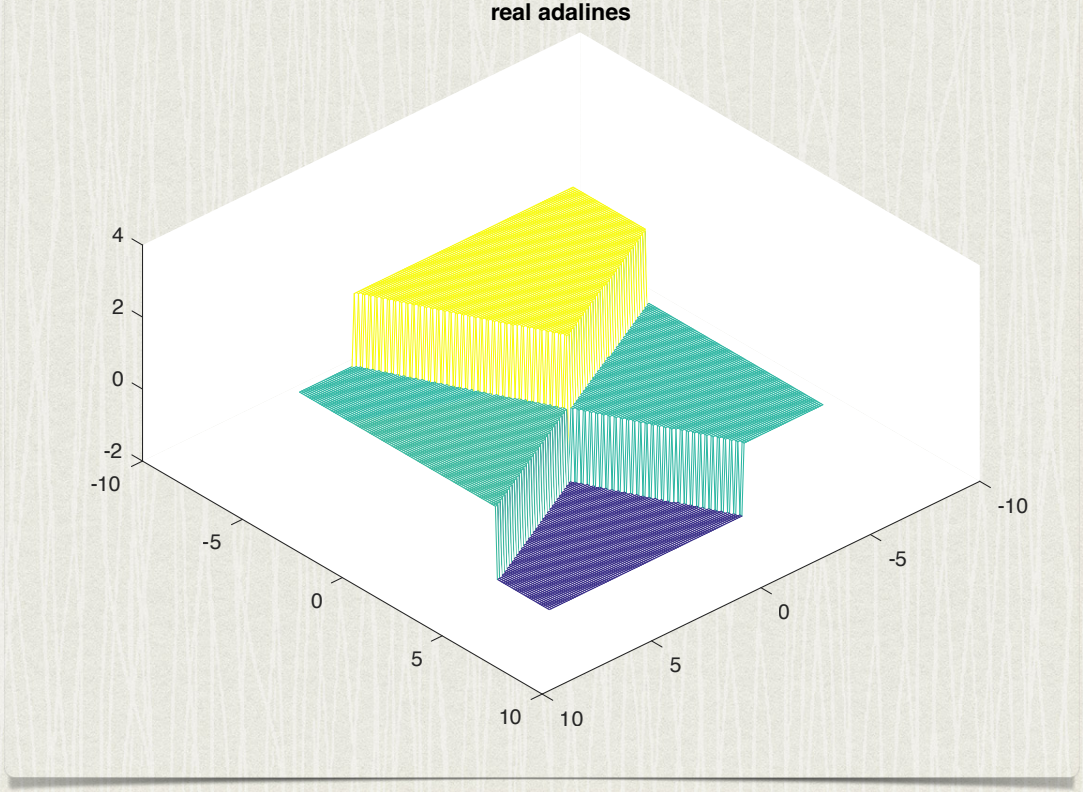

$y$




$y = \hat{y}$






*Adaptive Linear Element*

*ADALINE of Widrow*

# APPROXIMATION

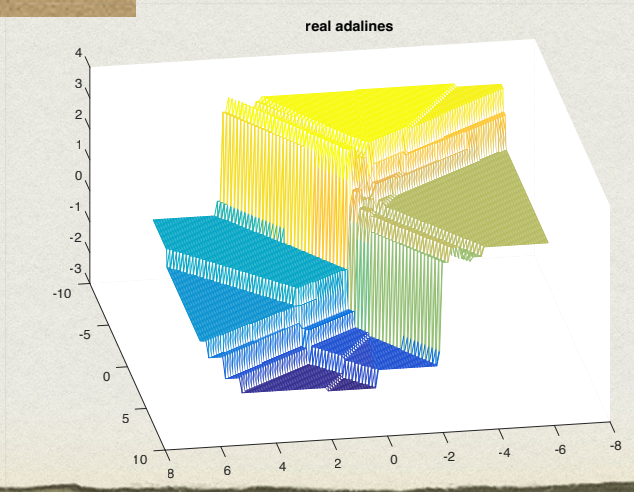
$$y = tanh(2x_1 + x_2 - 1) - tanh(x_1 - 2x_2 + 1)$$

input
output

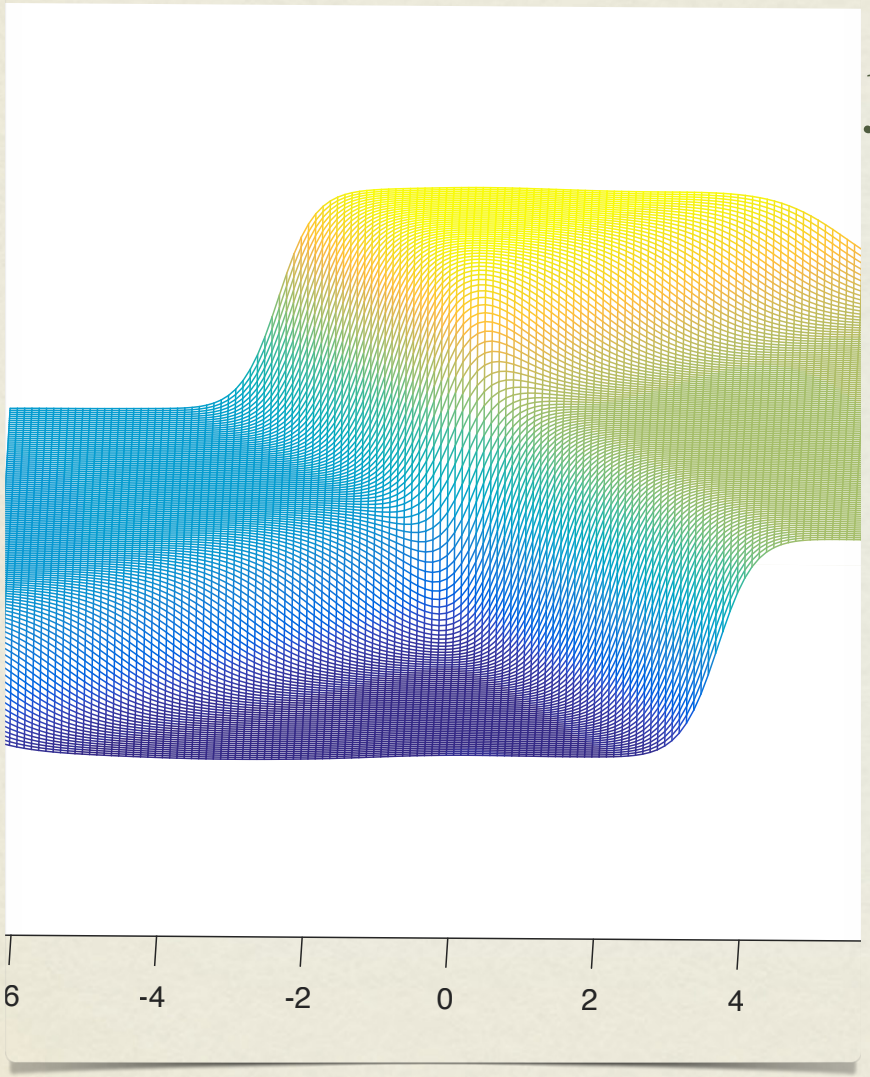
Learning a discrete neural network

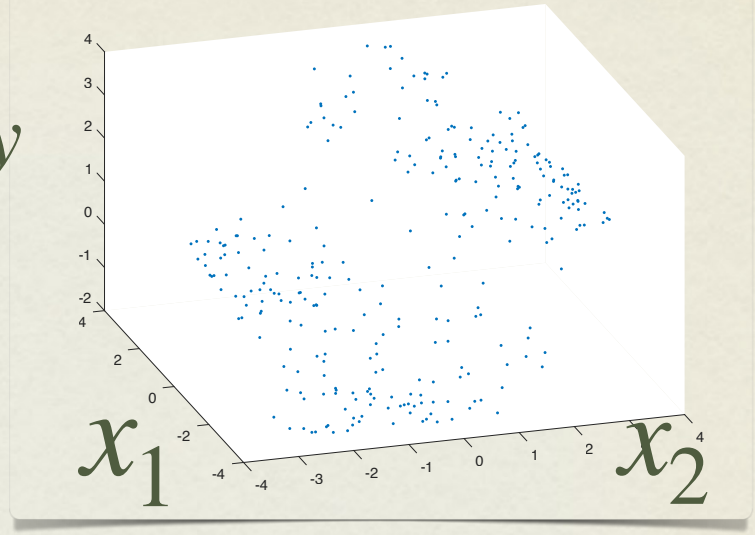
optimal interconnections

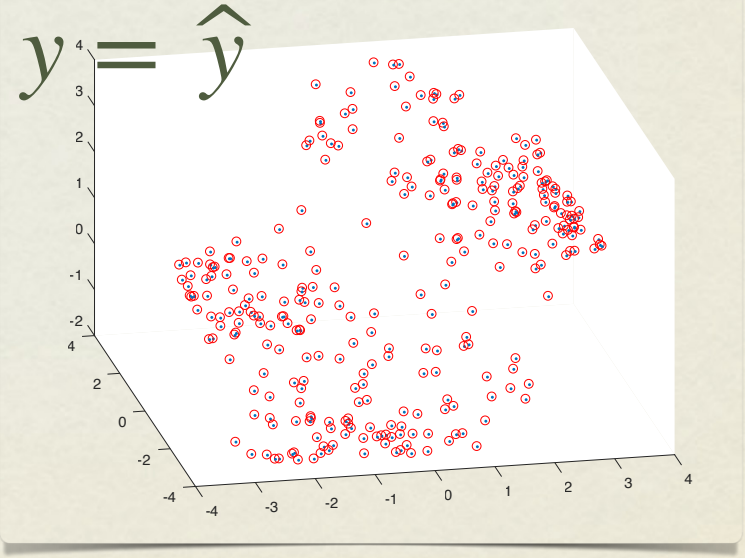
An approximating network

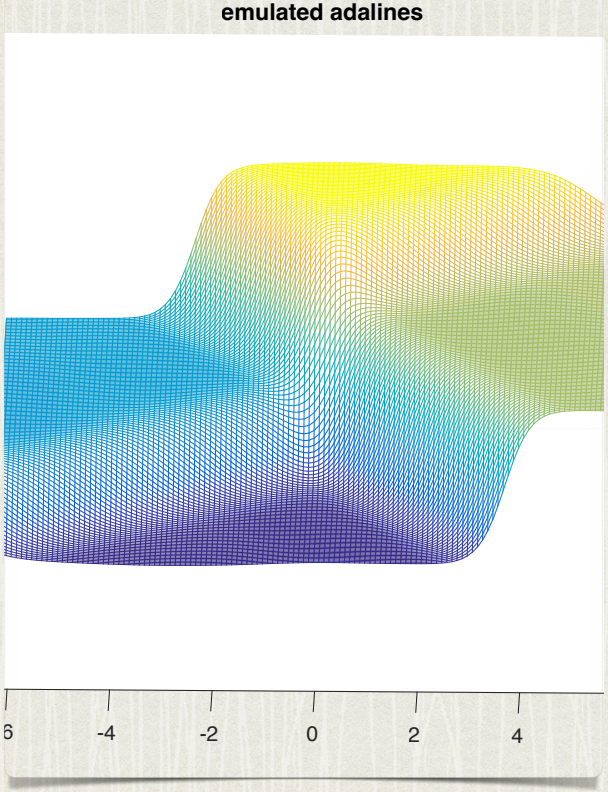


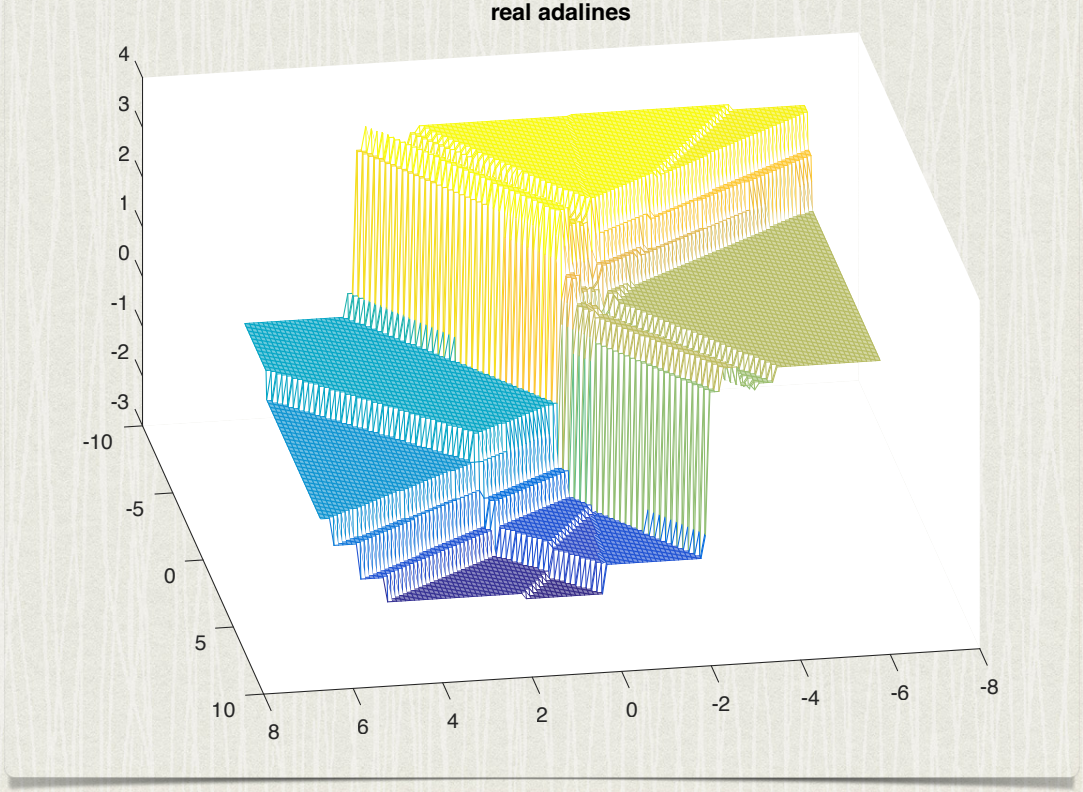
**emulated adalines**



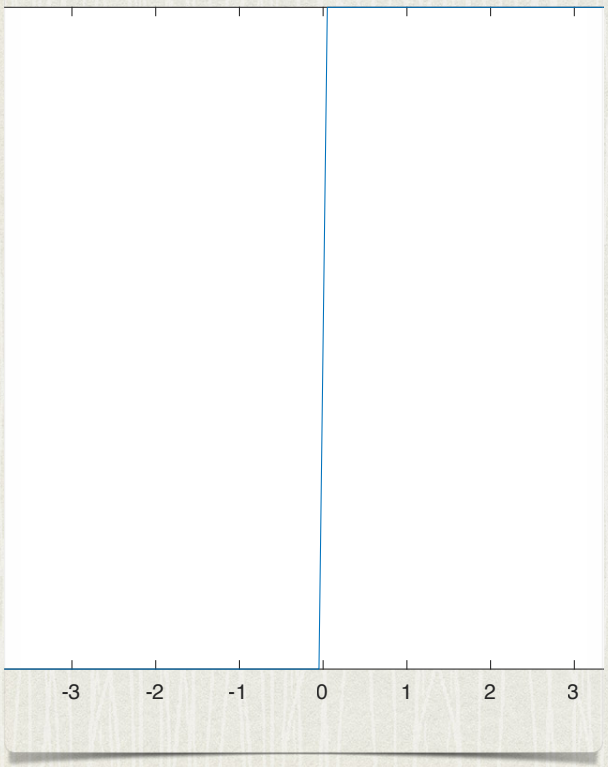



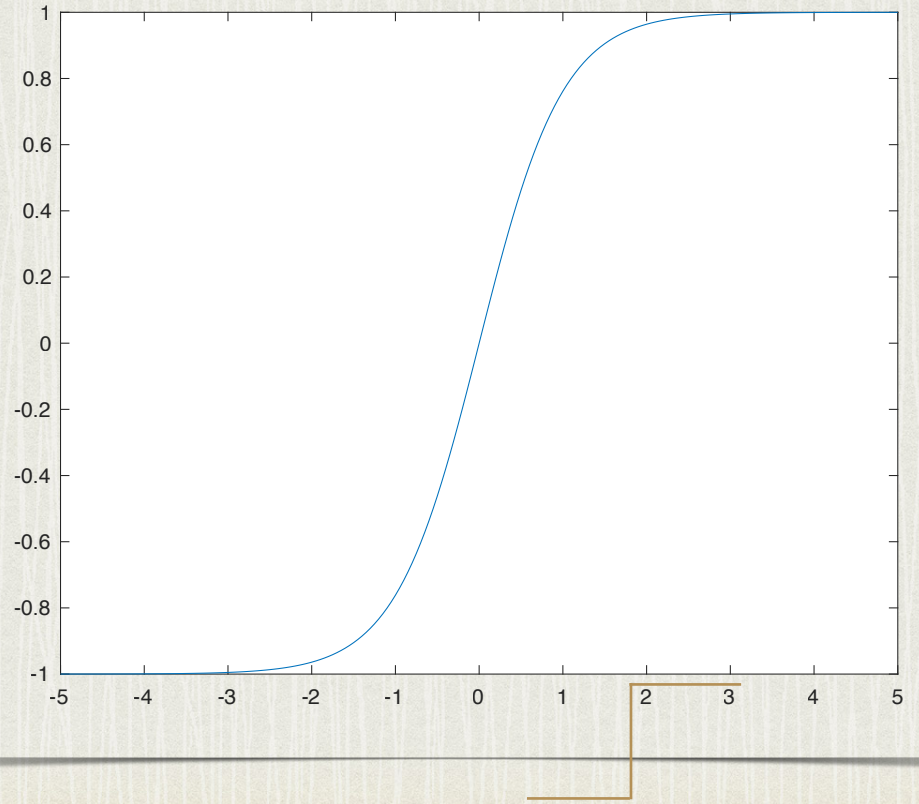



*Adaptive Linear Element*

# *ADALINE of Widrow*

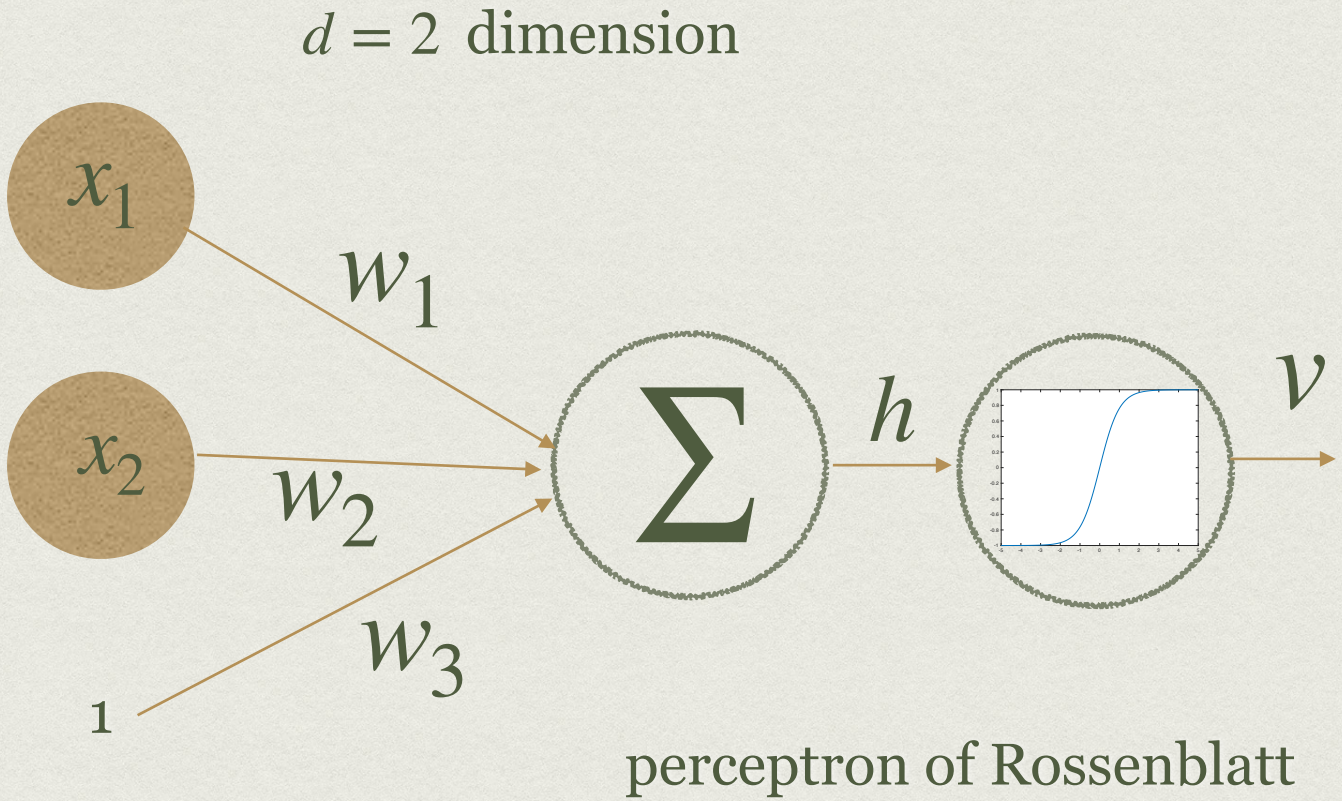



# HYPER TANGENT

*Differentiable sigmoid function*

# CONTINUOUS FUNCTION

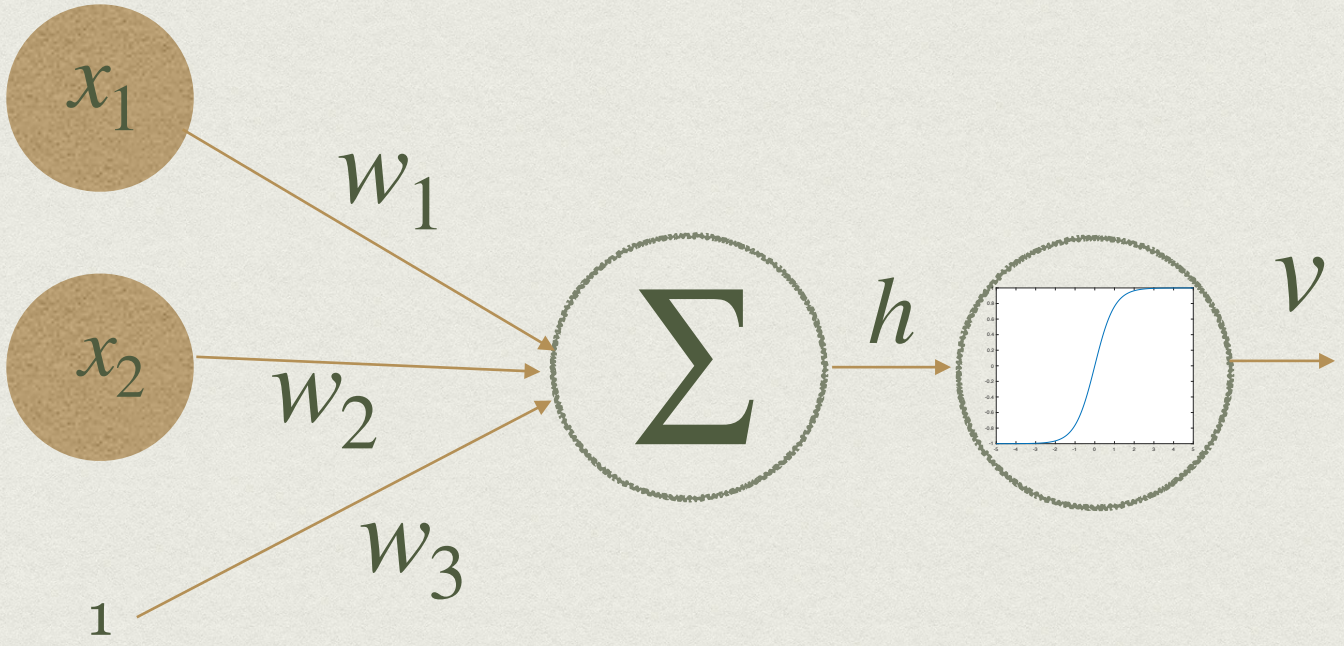



perceptron of Rossenblatt

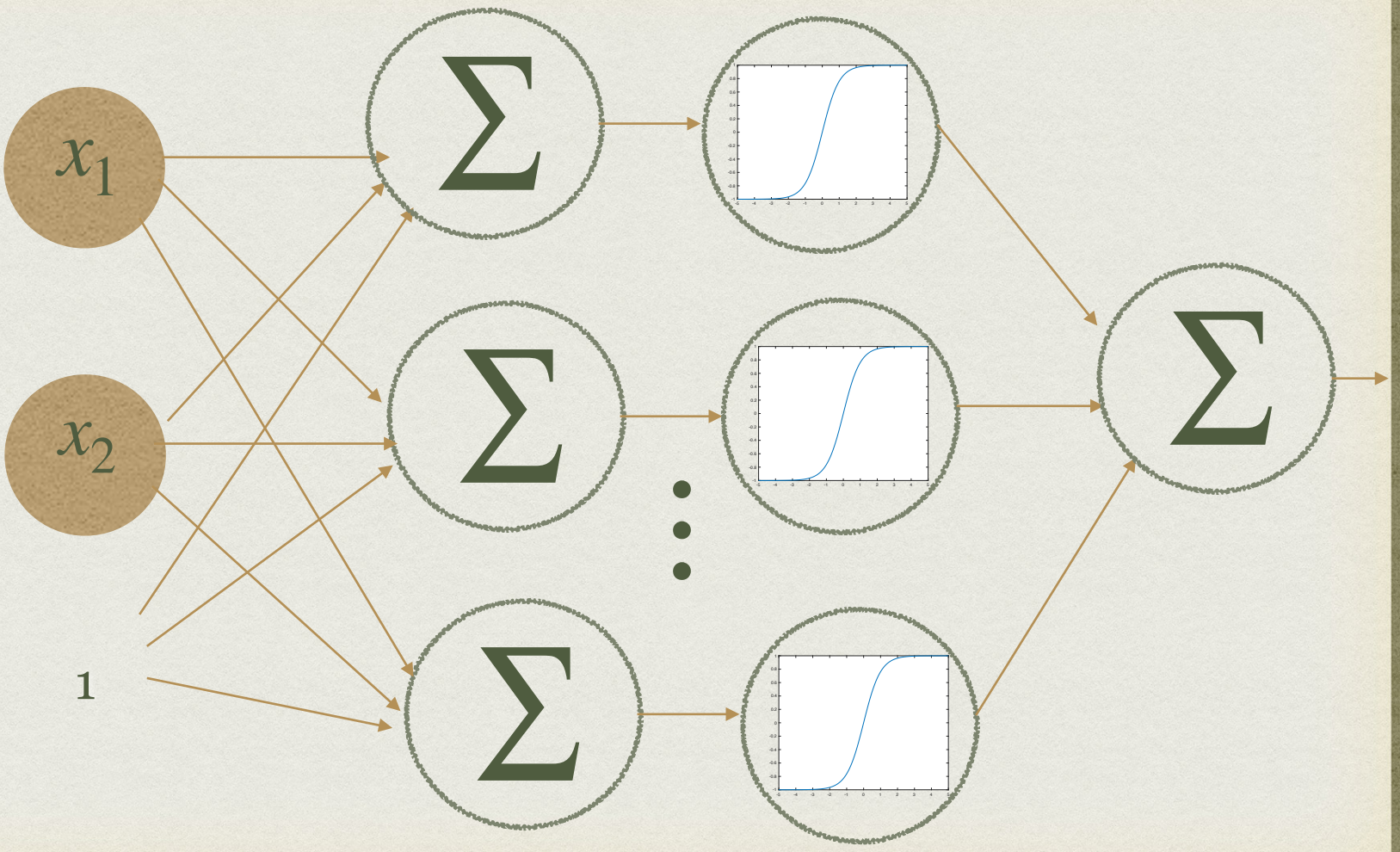
# AI PERCEPTRON

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

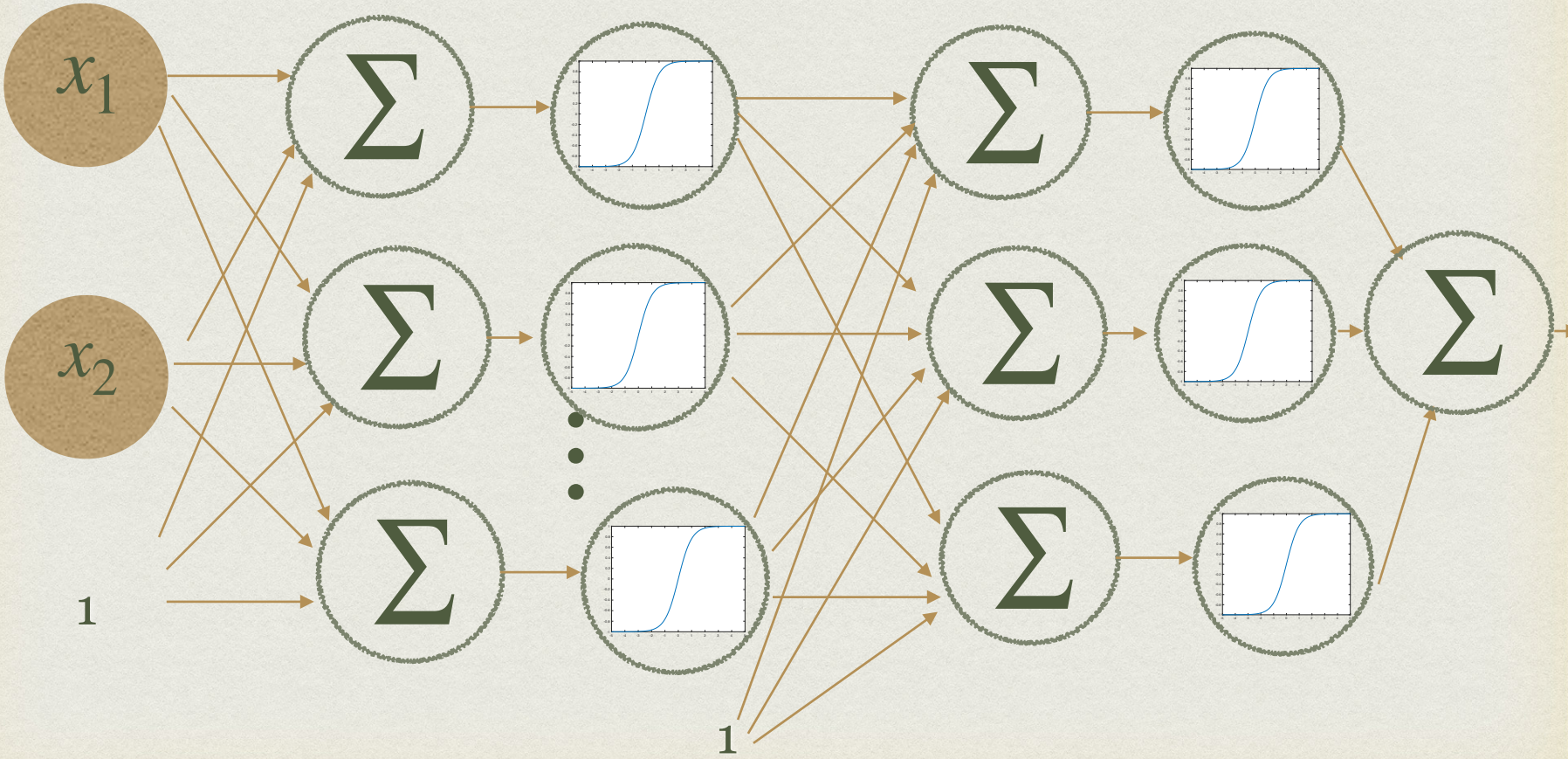
$$v = tanh(h)$$



# A MULTILAYER NEURAL NETWORK



# A DEEP NEURAL NETWORK



# APPROXIMATION

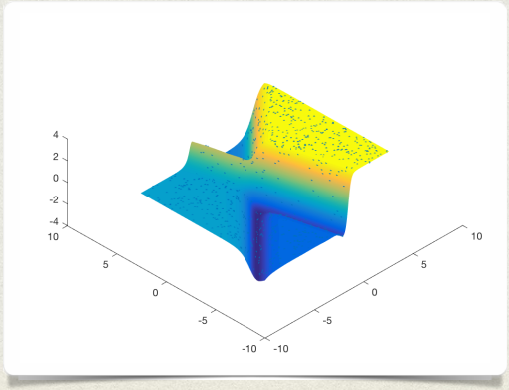
$$v_1 = tanh(2x_1 + 0.5x_2 - 1)$$
$$v_2 = tanh(x_1 - x_2 + 1)$$
$$y = 2cos(2v_1 - v_2 - 1) + sin(v_1 + v_2 - 1)$$

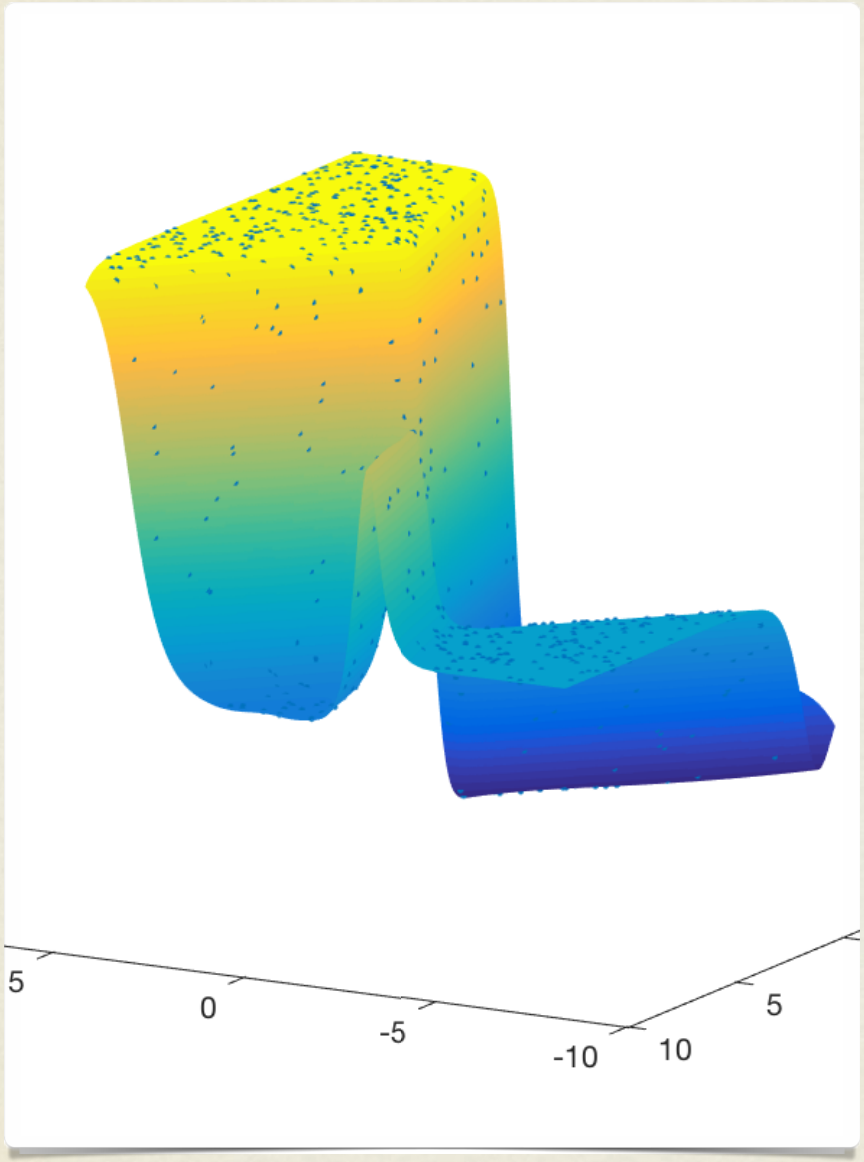
input
output

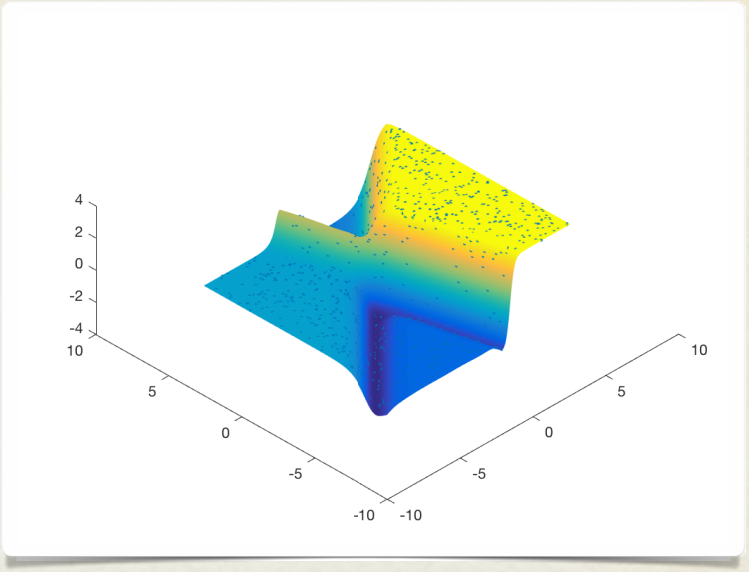
Learning a deep neural network

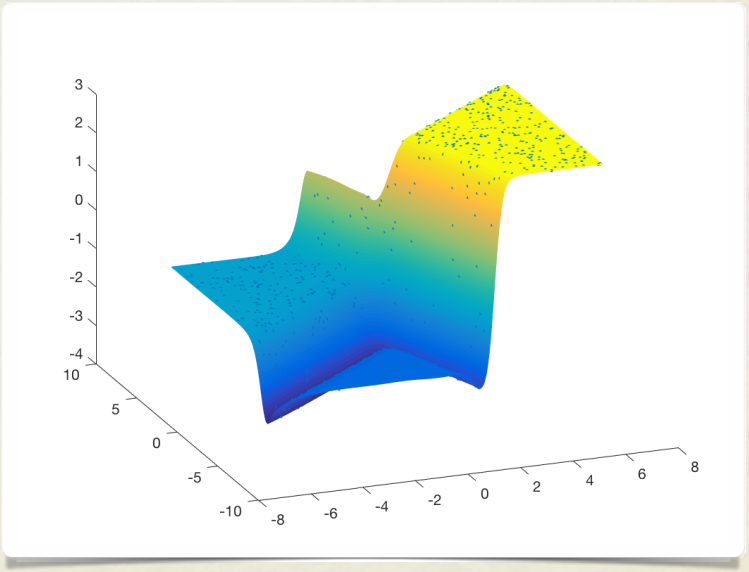
optimal interconnections

An approximating network





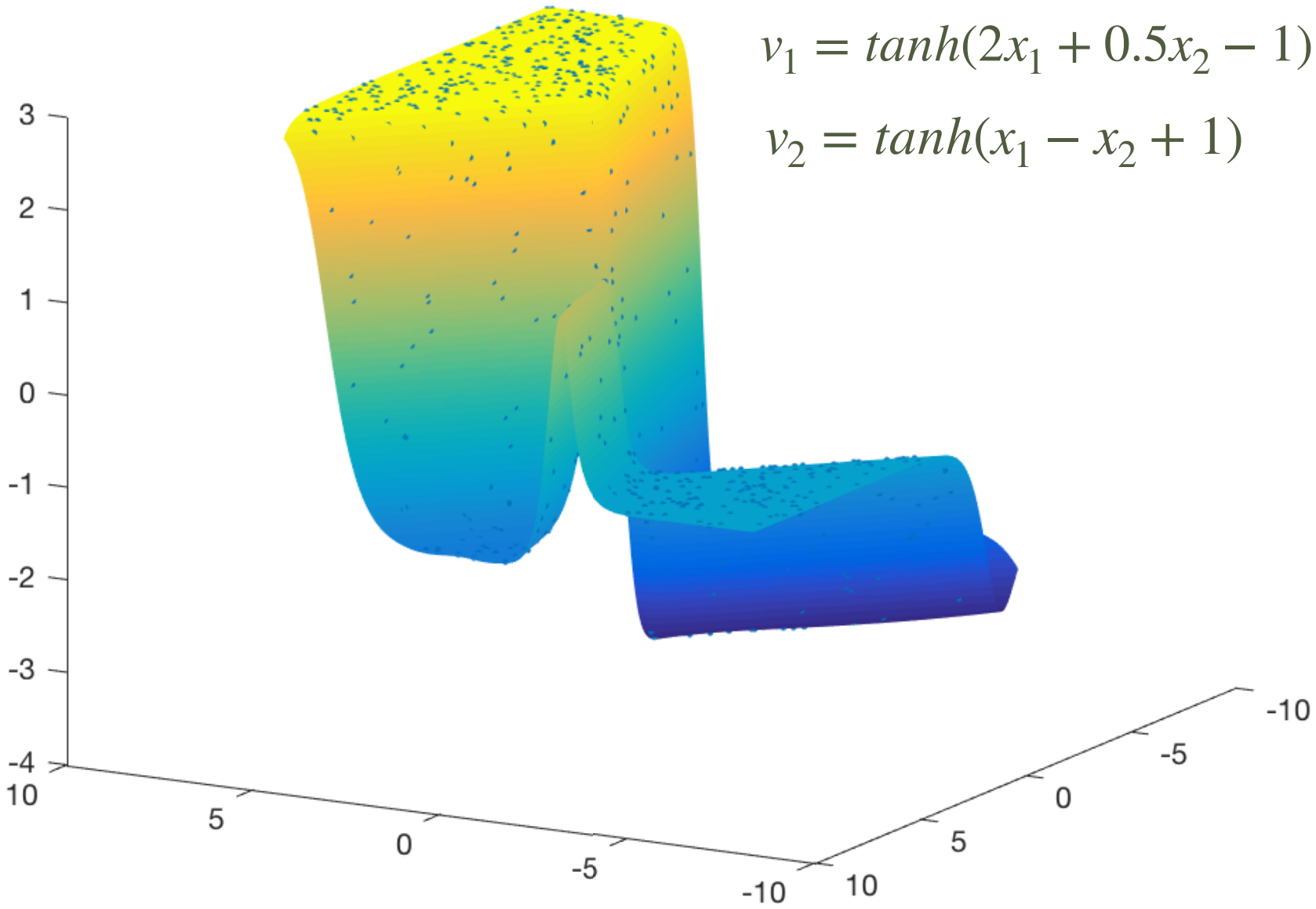




```
v = [tanh(2*x(:,1)+0.5*x(:,2)-1) tanh(x(:,1)-x(:,2)+1)];
y= 2*cos(2*v(:,1)-v(:,2)-1)+sin(v(:,1)+v(:,2)-1);
```

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

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

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



# APPROXIMATION

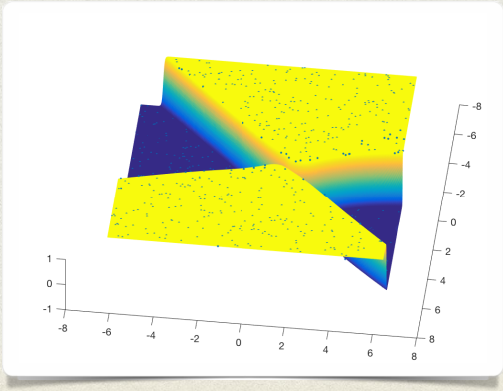
$$v_1 = sign(2x_1 + 0.5x_2 - 1)$$
$$v_2 = sign(x_1 - x_2 + 1)$$
$$y = sign(v_1 v_2)$$

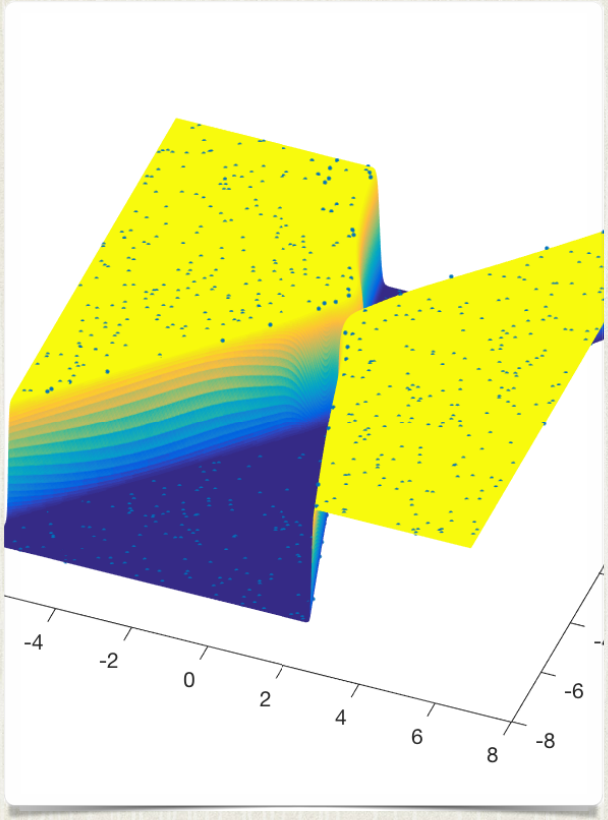
input
output

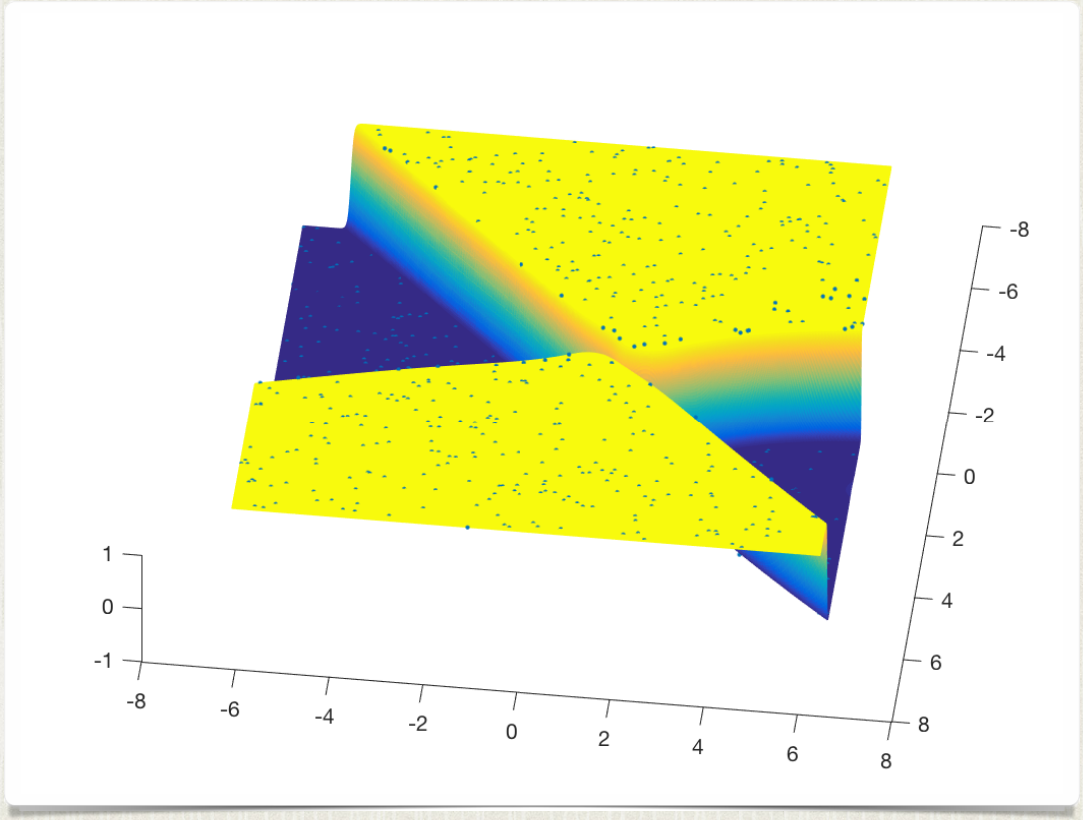
Learning a deep neural network

optimal interconnections

An approximating network

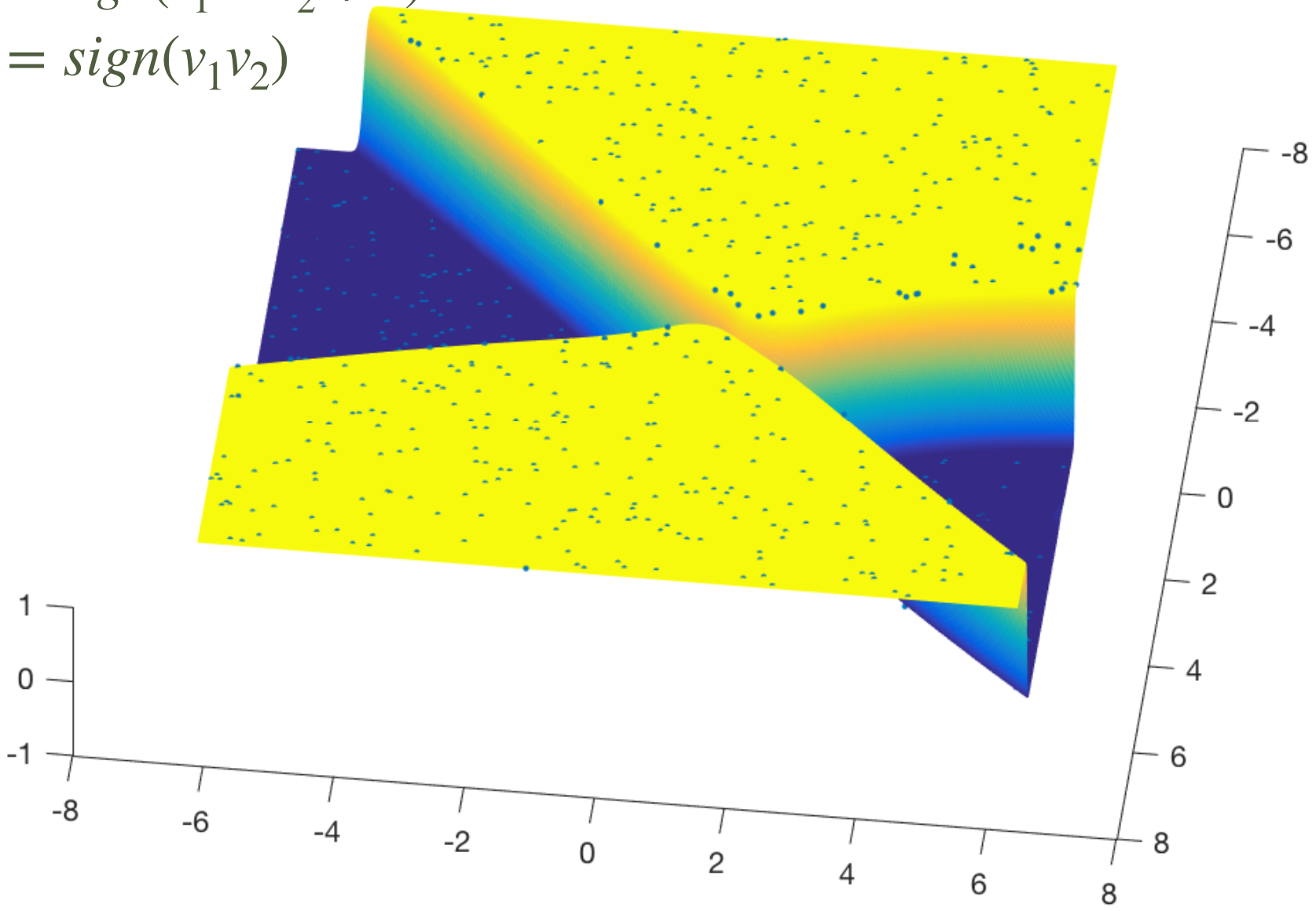





```
v = [sign(2*x(:,1)+0.5*x(:,2)-1) sign(x(:,1)-x(:,2)+1)];
y= sign(v(:,1).*v(:,2));
```

$$v_1 = sign(2x_1 + 0.5x_2 - 1)$$
$$v_2 = sign(x_1 - x_2 + 1)$$
$$y = sign(v_1 v_2)$$



## MatconvNet/examples/fast_rcnn

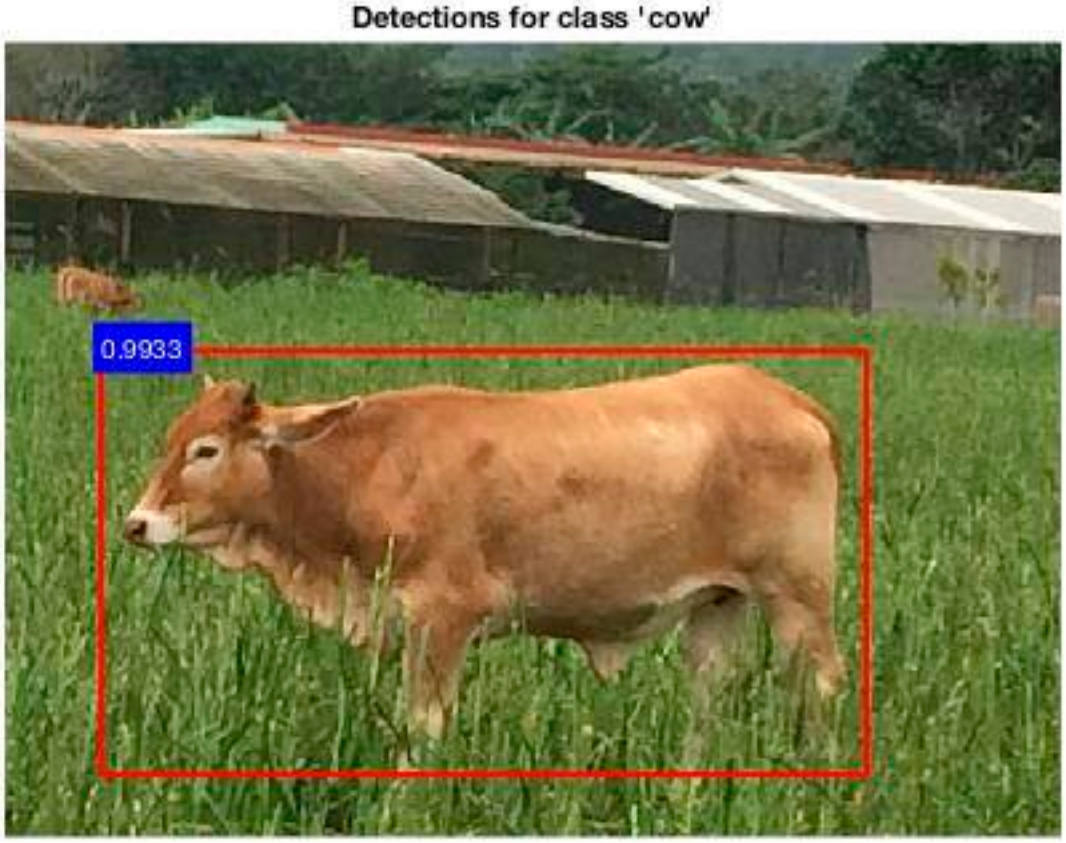
## fast_rcnn_demo



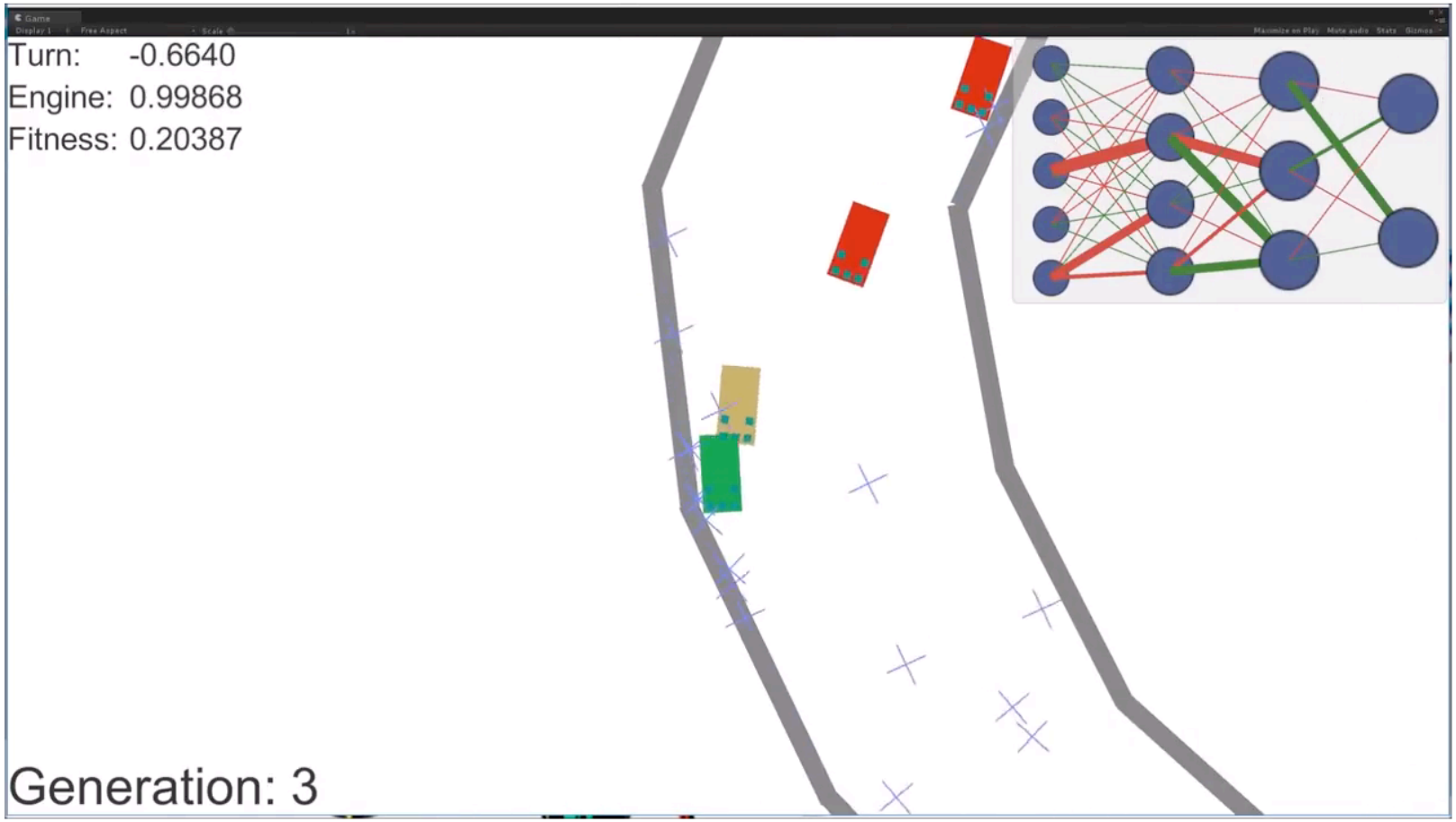

Problem
Architecture
Learning
Data
Execution
Results
your comments
.
.
.
Reference

**Please download again** fast-rcnn-vgg16-pascal07-dagnn

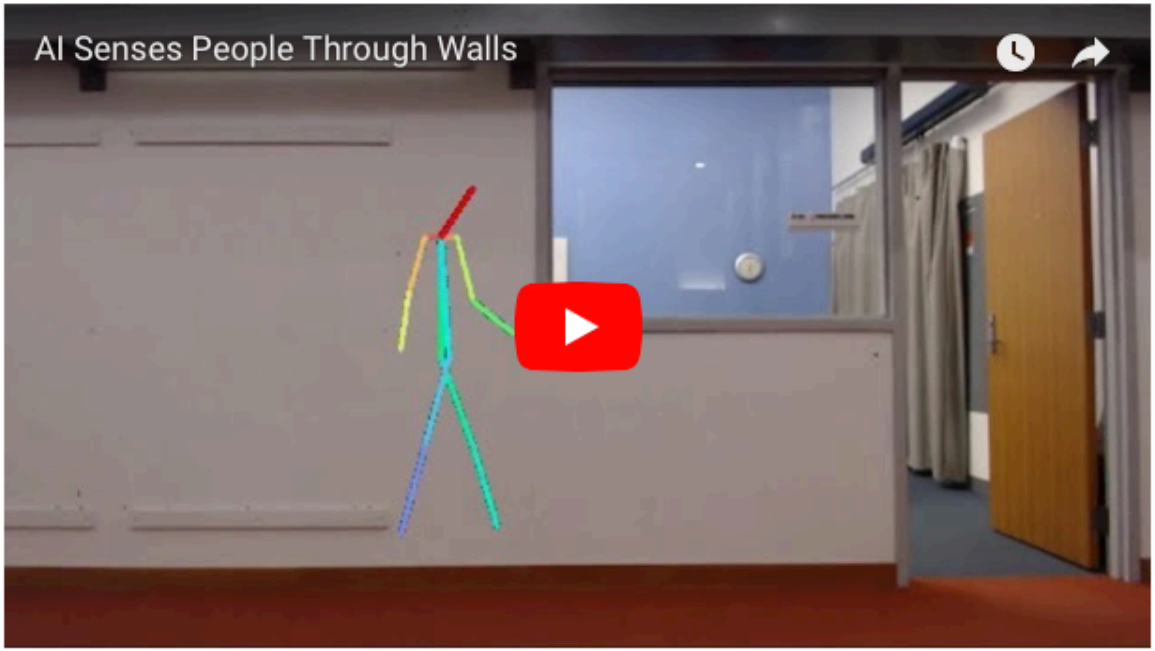
http://www.vlfeat.org/matconvnet/pretrained/

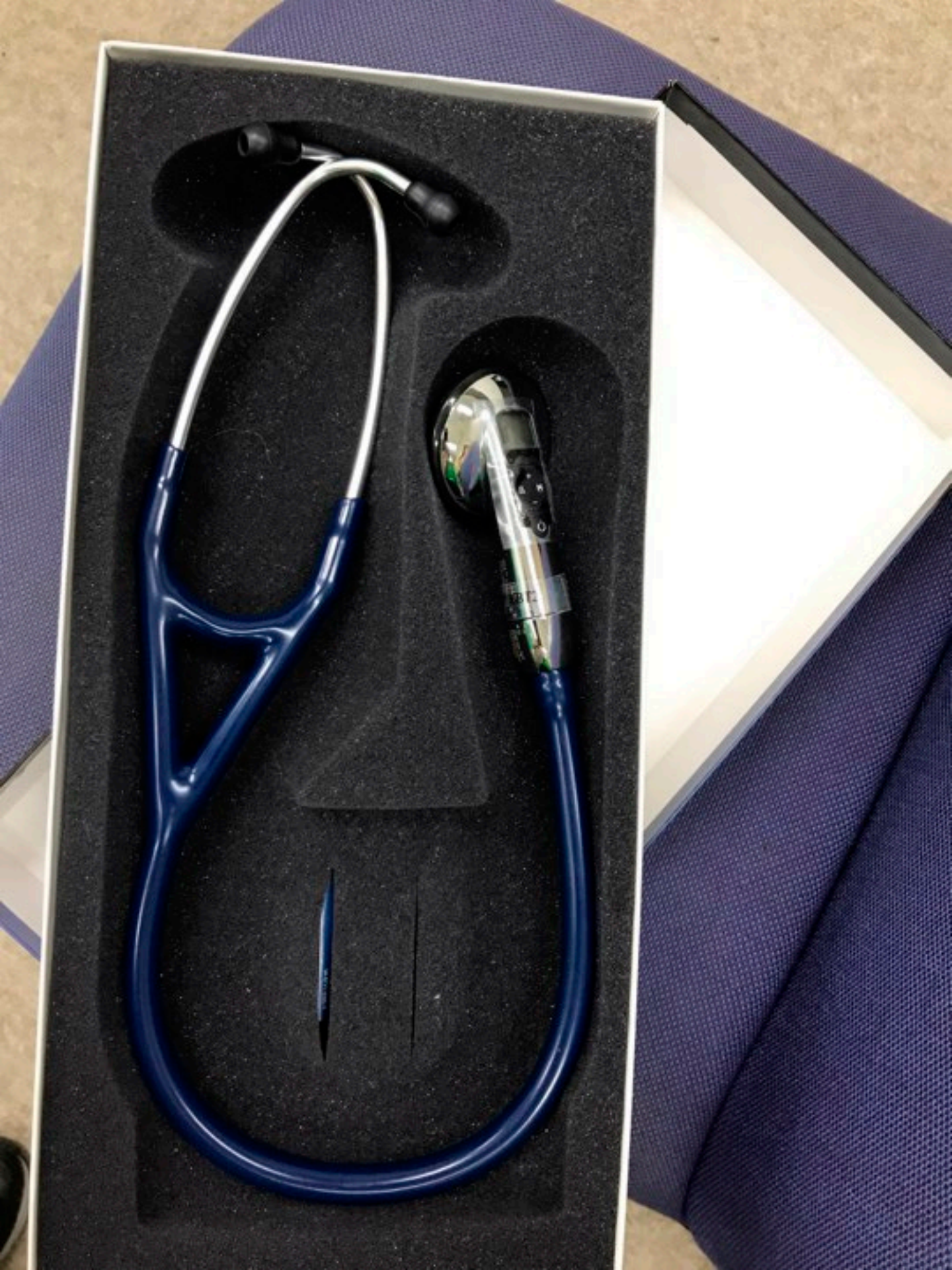

Detections for class 'cow'
0.9933

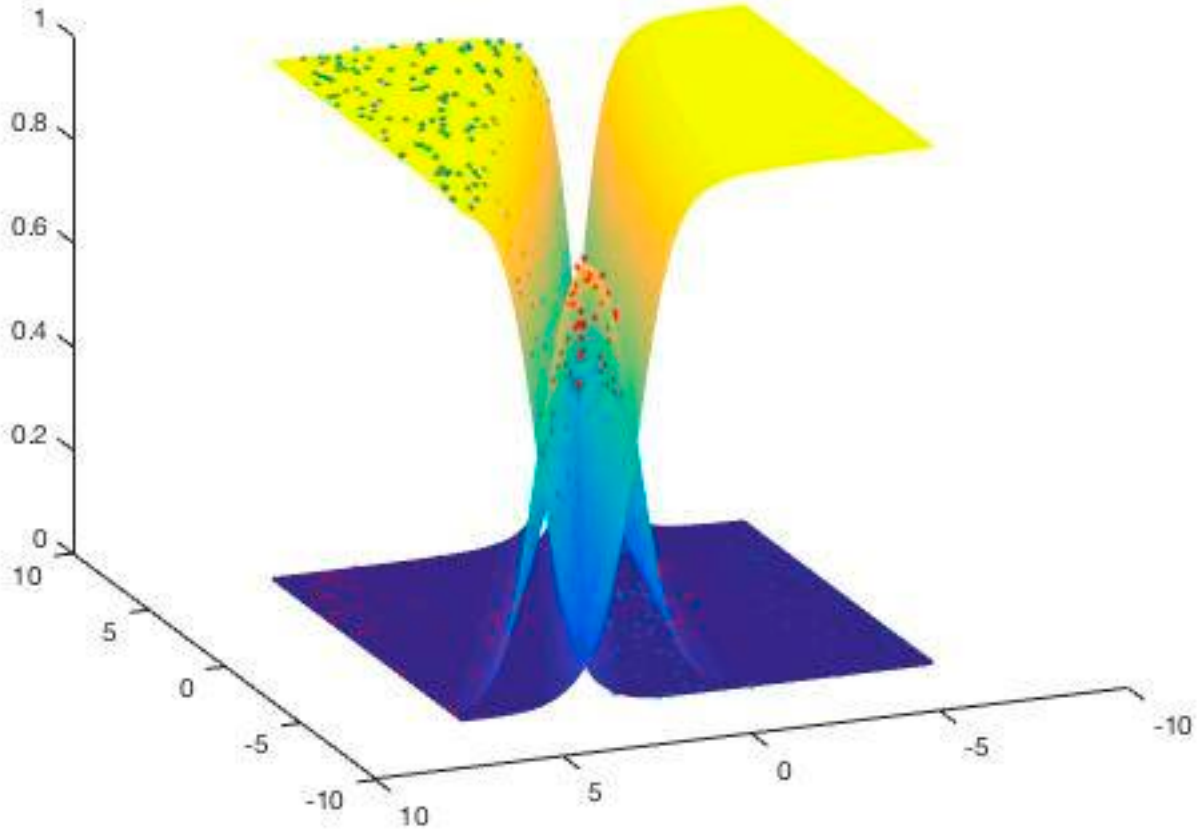

Detections for class 'cat'
0.9464




Deep Learning Cars

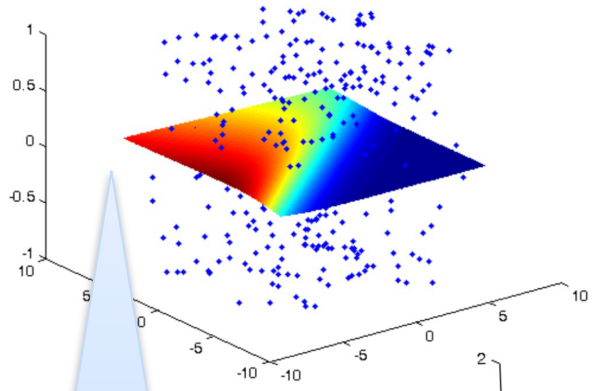

AI Senses People Through Walls




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

# Data oriented function approximation

- Adaptable function



An adaptable network mapping

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

Optimal network parameters

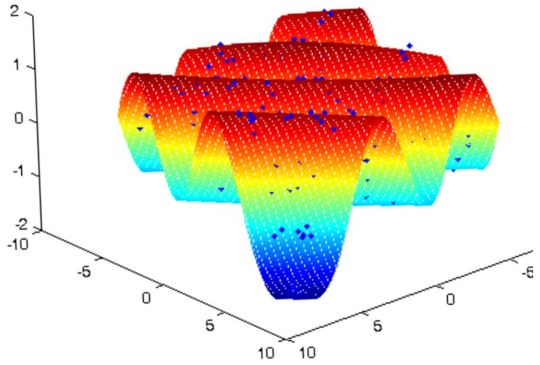


**Table 1**
Target functions.

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

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

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

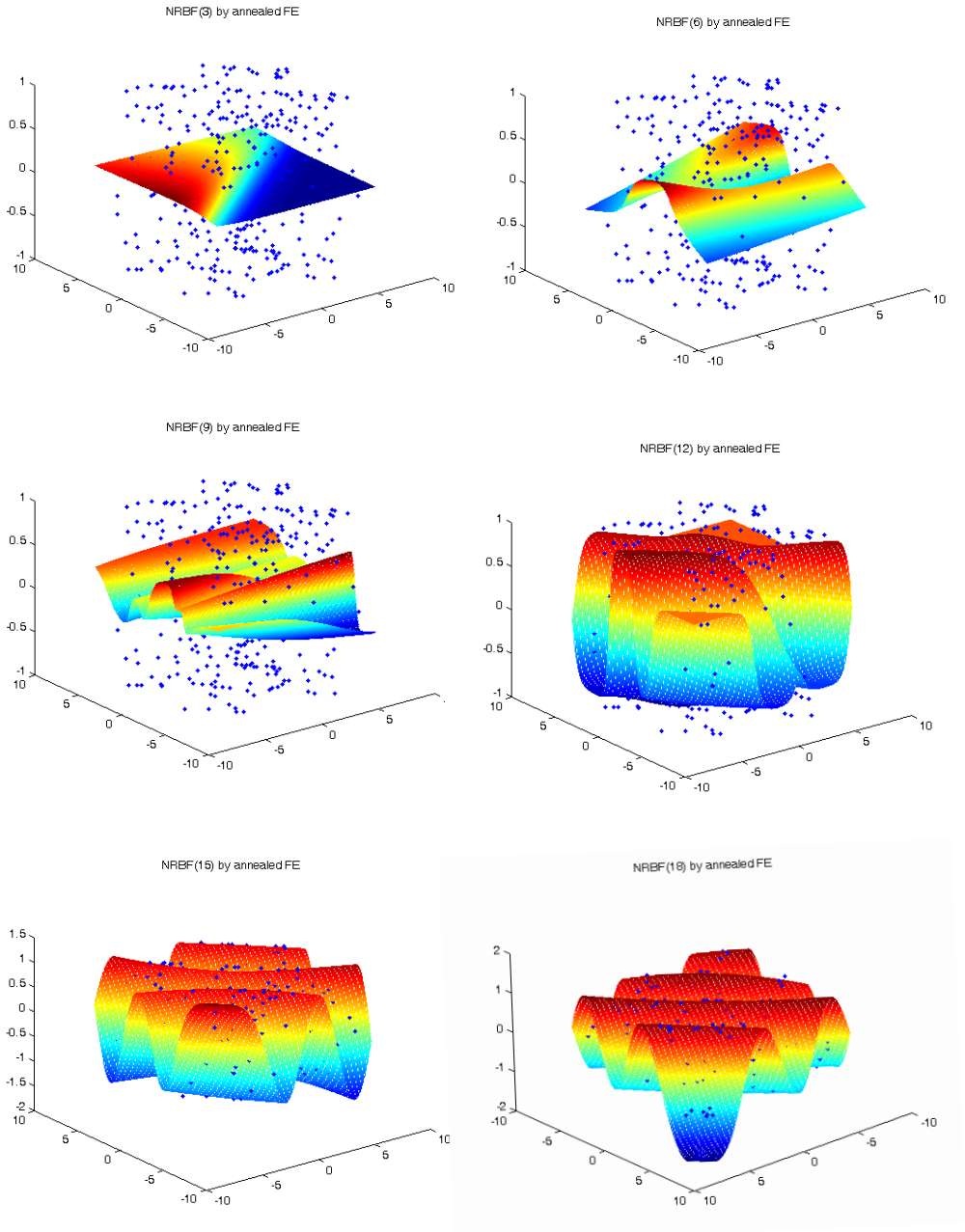

NRBF(3) by annealed FE
NRBF(6) by annealed FE
NRBF(9) by annealed FE
NRBF(12) by annealed FE
NRBF(15) by annealed FE
NRBF(18) by annealed FE

Figure 4

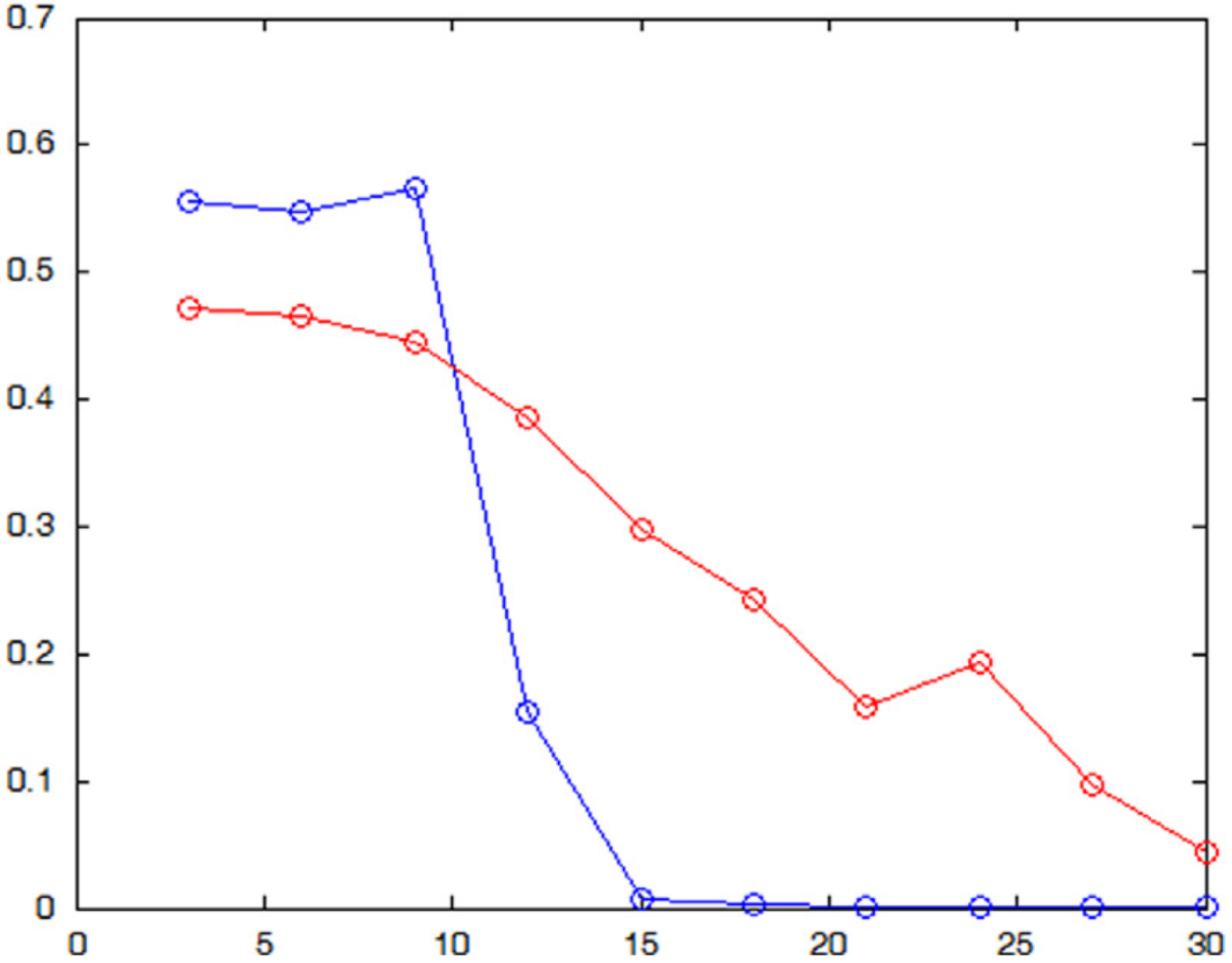
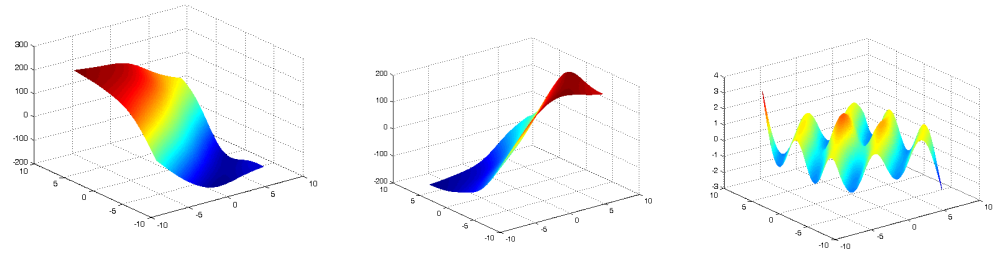
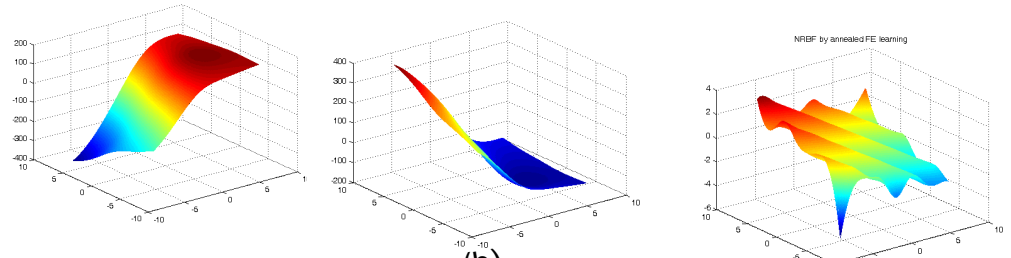


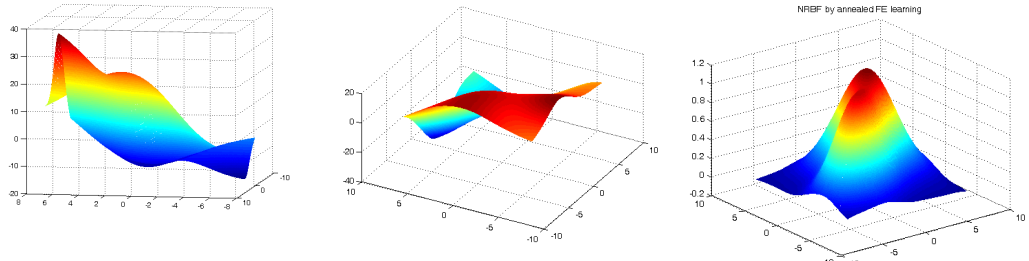

**Fig. 5.** Mean square testing errors of annealed competitive learning (blue curve) and the Rätsch method (red curve) in approximating $f_1$ versus the numbers of hidden units. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)



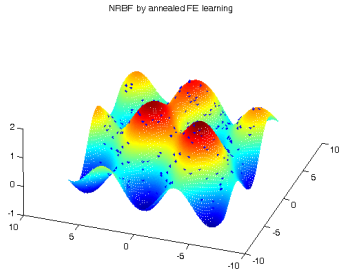
(a)



(b)

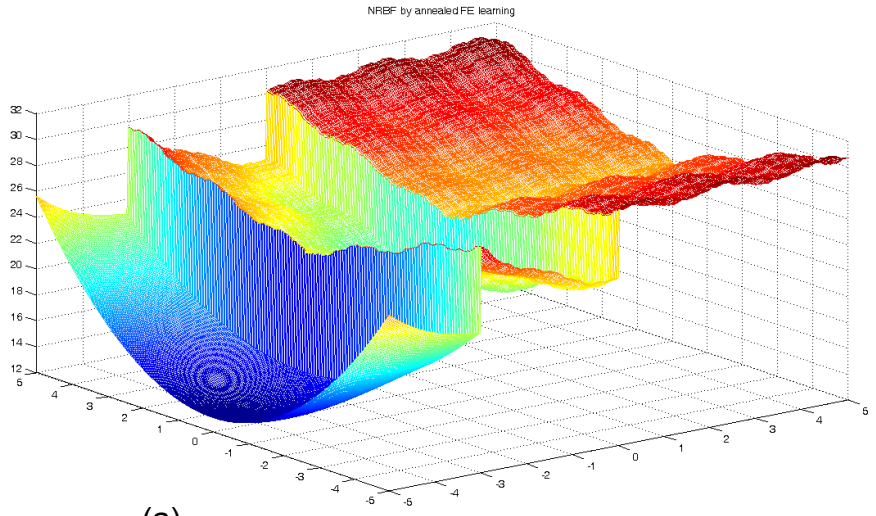


(c)

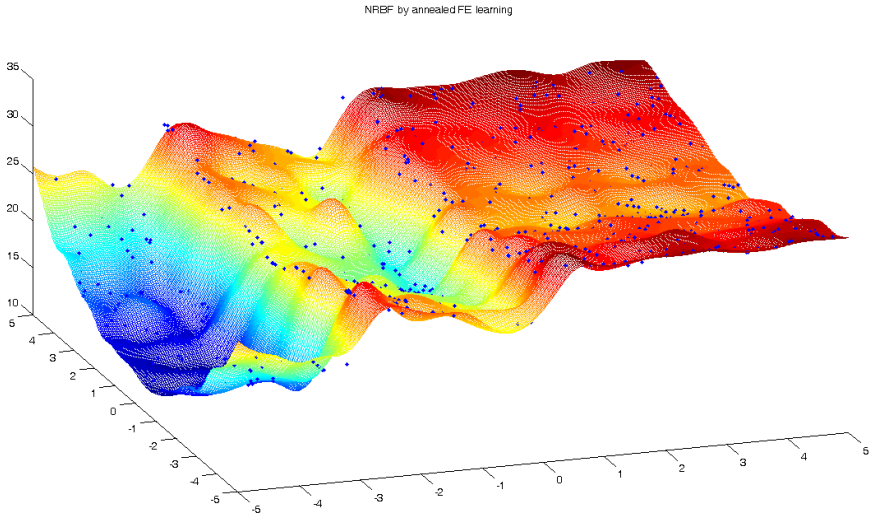



(d)

Figure 6




(a)



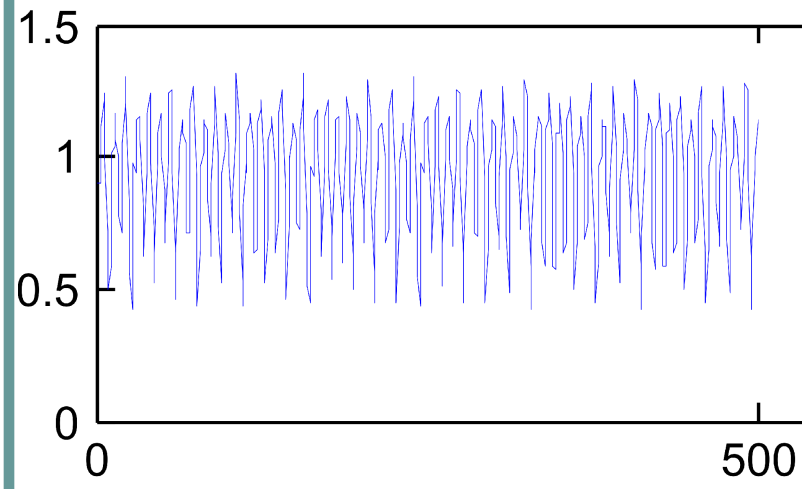

(b)

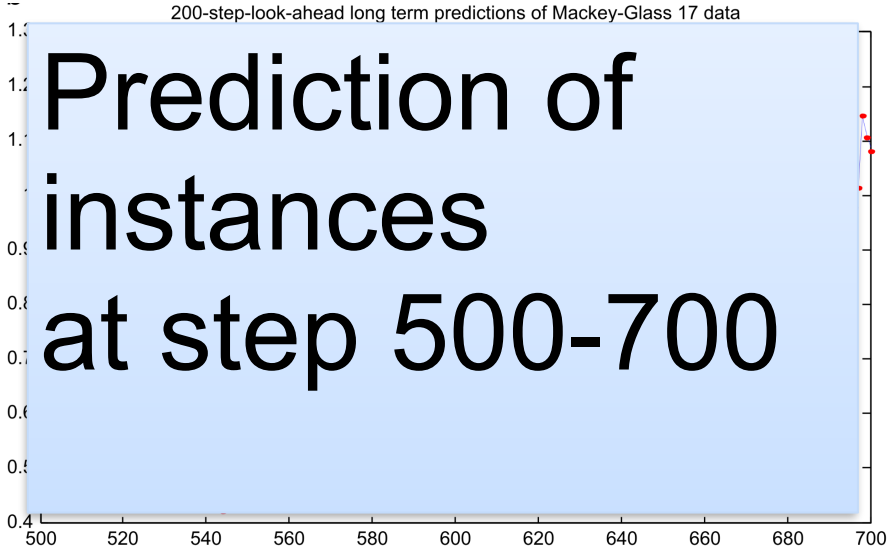
Figure 7

# Chaotic differential function approximation

# Data driven long-term prediction

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



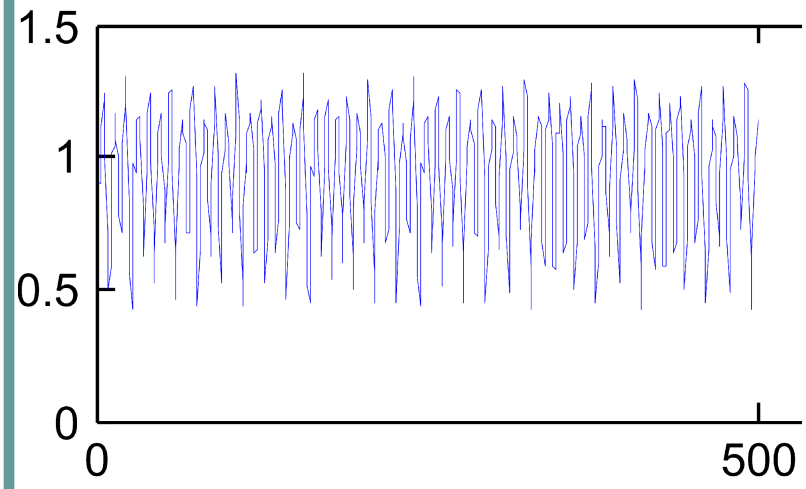



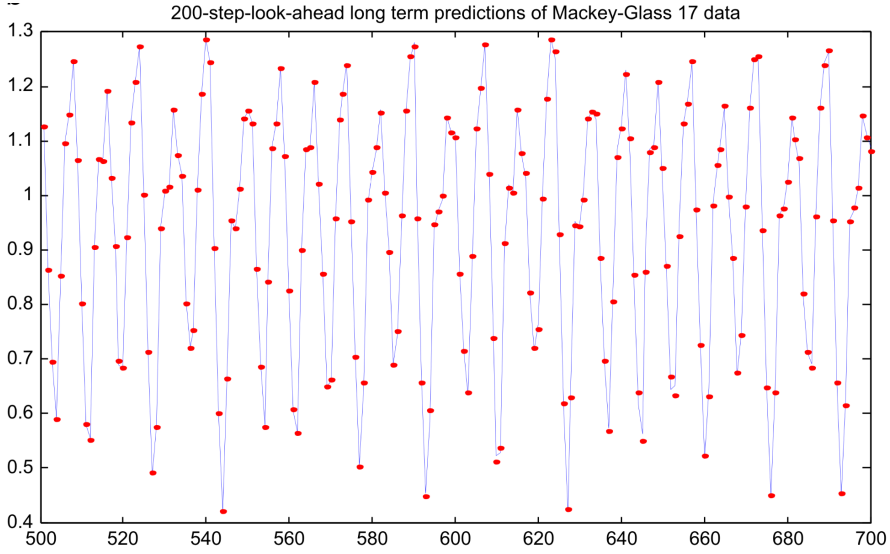

Prediction of instances at step 500-700

200-step-look-ahead prediction.

# Data driven long-term prediction

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

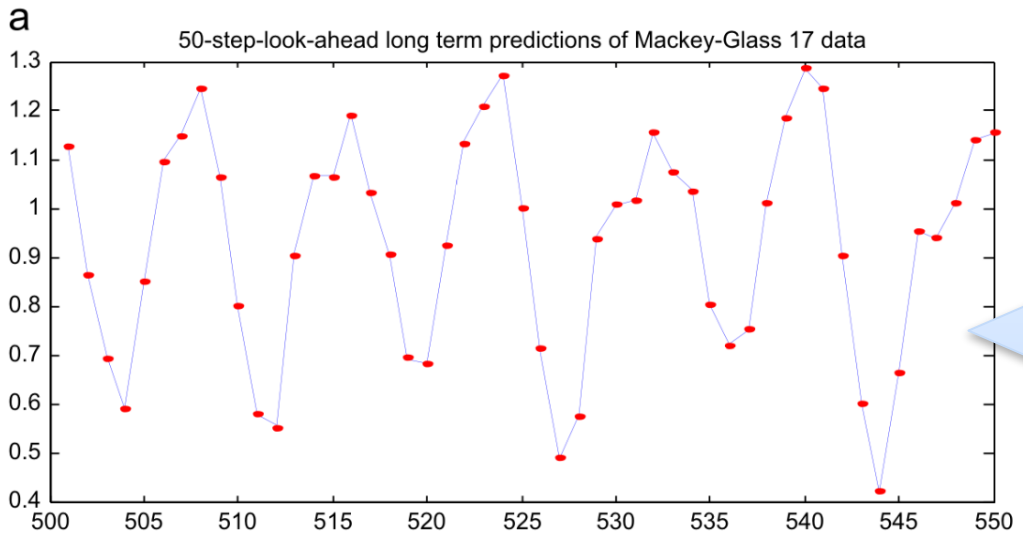


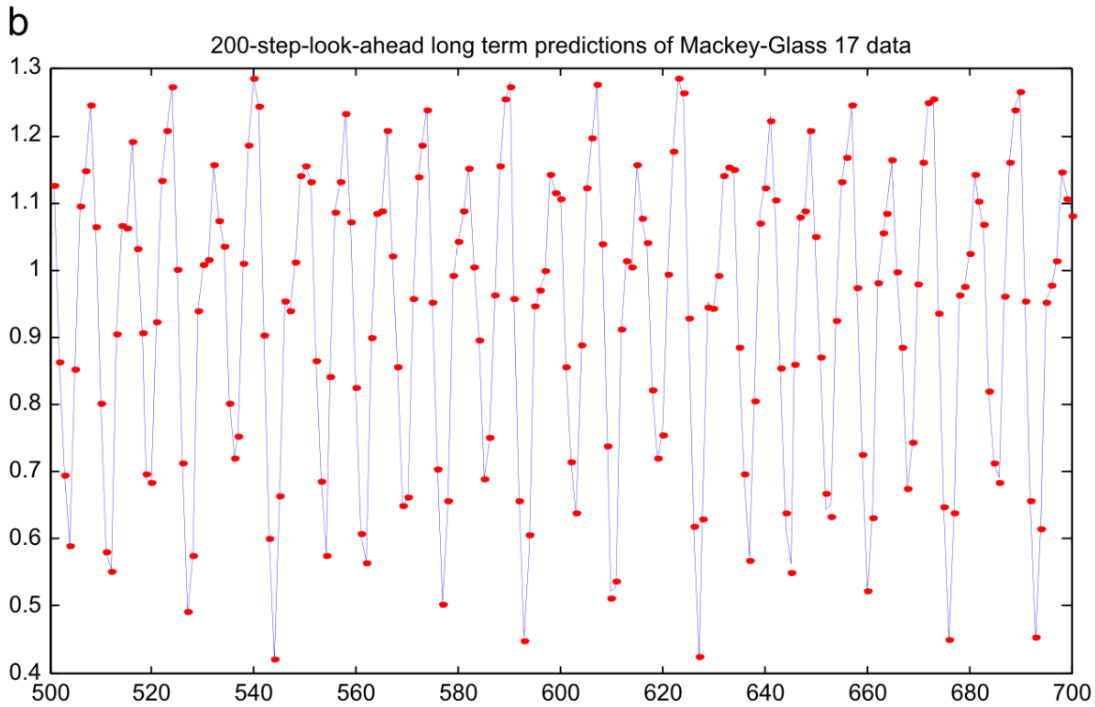

200-step-look-ahead prediction.

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

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




correlation coefficient 0.9999



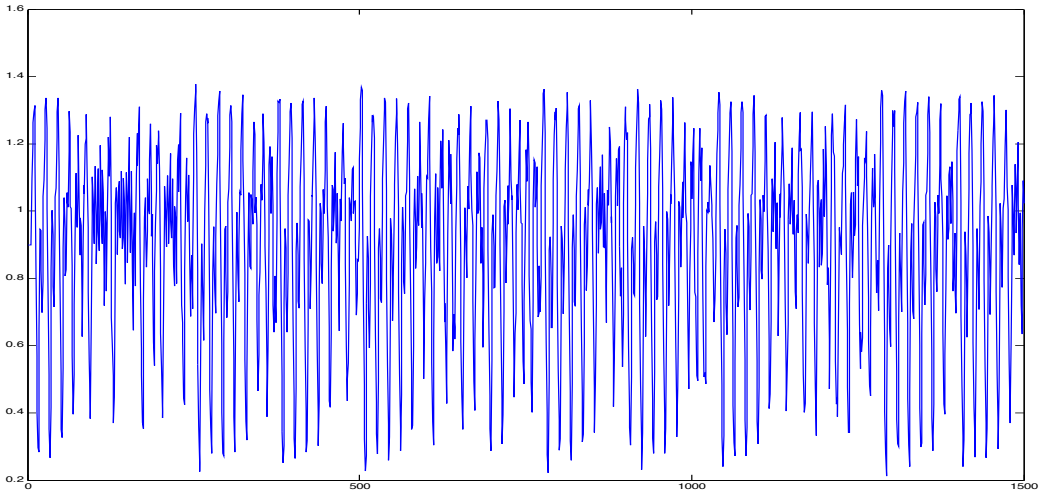

correlation coefficient 0.9993

# Mackey-Glass 30

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

$$\tau = 30 \quad a = 0.2, c = 10 \text{ and } b = 0.1$$

Mackey-Glass 30 data




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

correlation coefficient 0.999

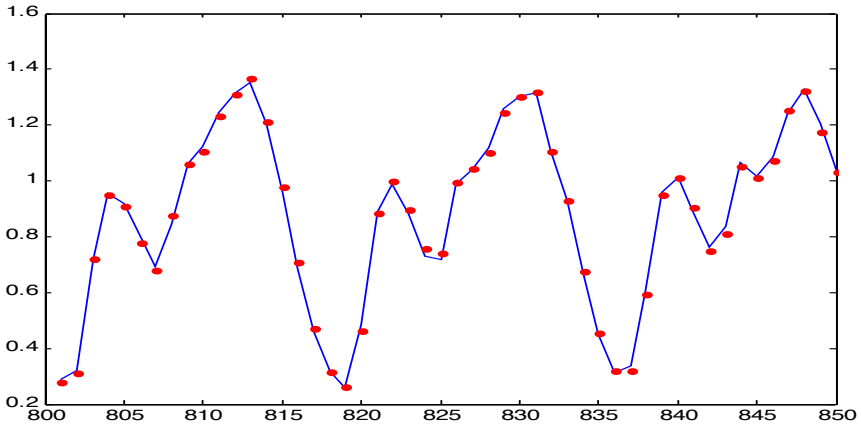



Figure 11

# CDFA: Nonlinear delay differential equations

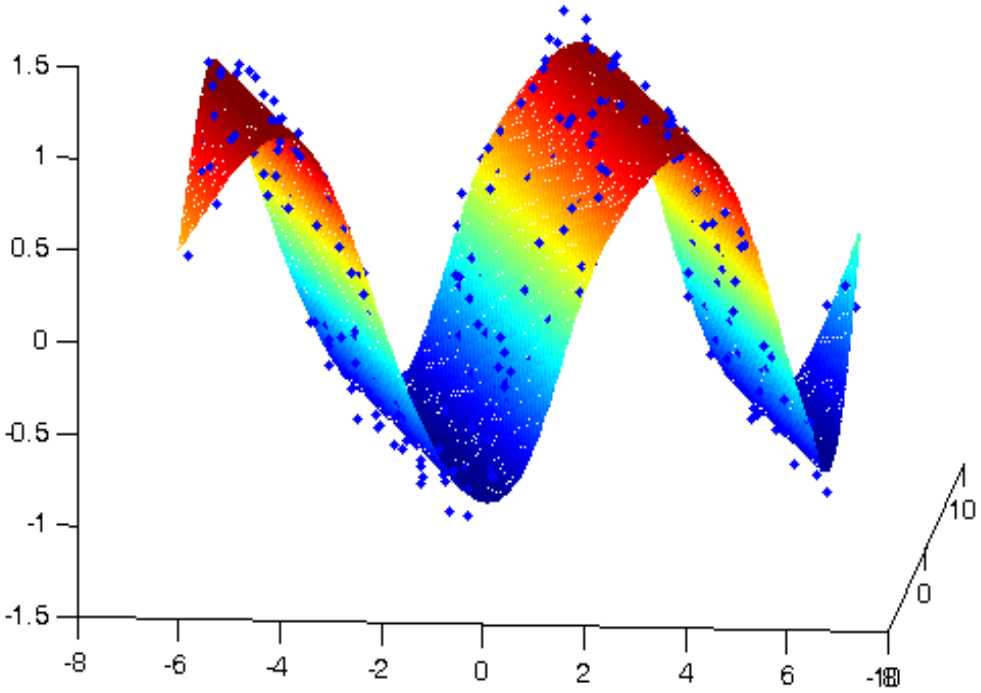
$$\frac{\partial x}{\partial t} = x(t-\tau) - x^3(1-\tau),$$

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

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

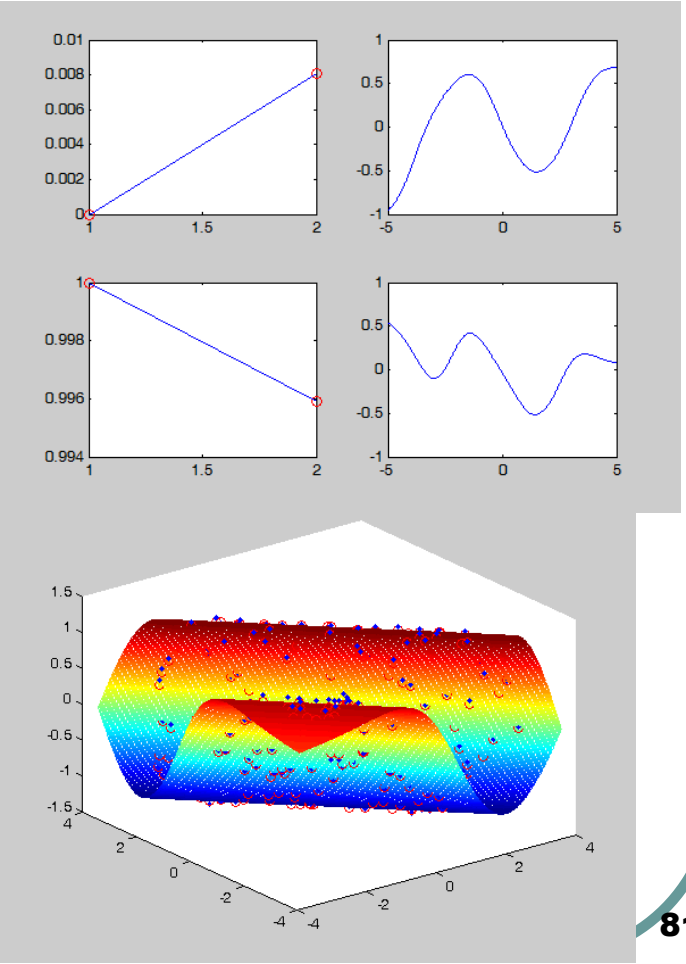
# LM learning for MLP

Two-dimensional Function Approximation



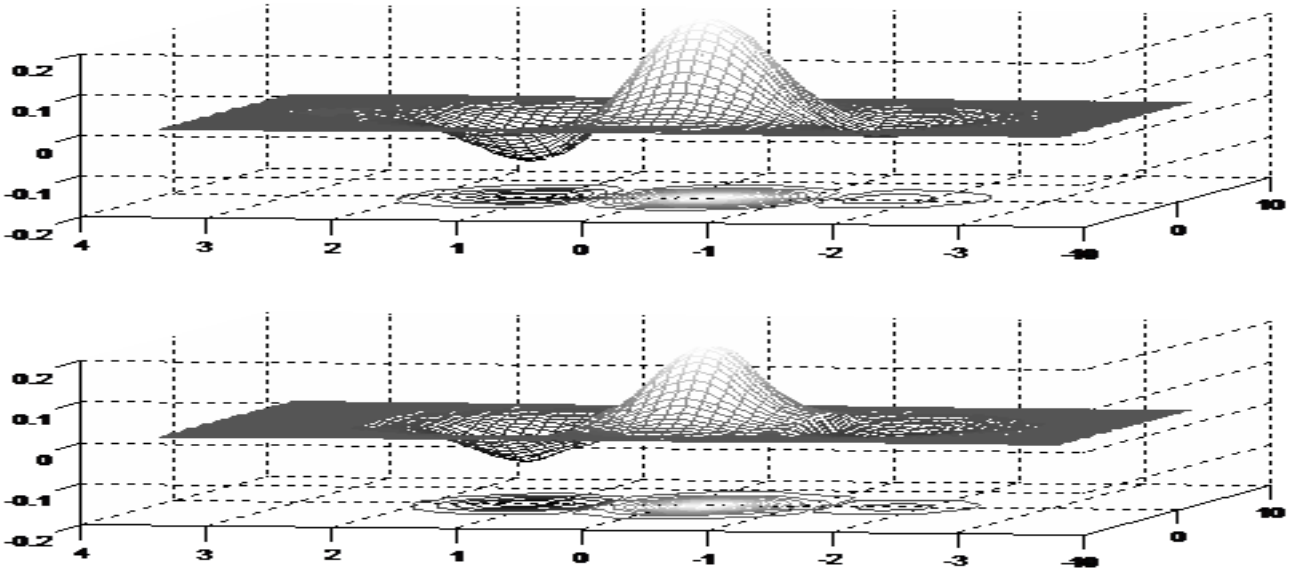
# Learning MLPotts networks

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



# Approximating Gabor function

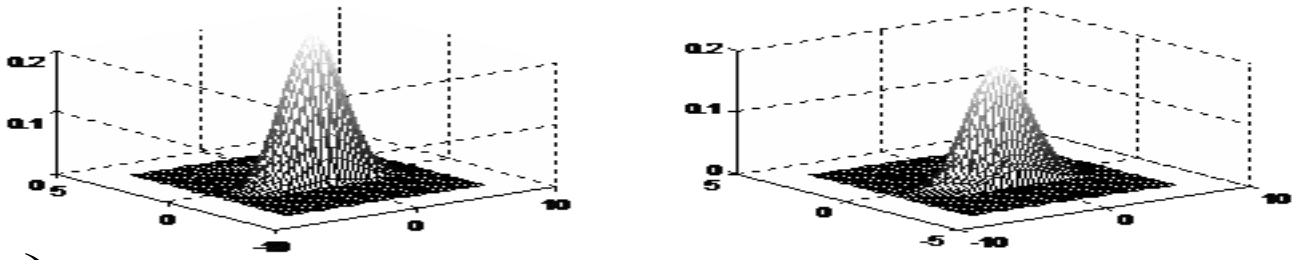
Learning generalized adalines (Wu et al 2006)



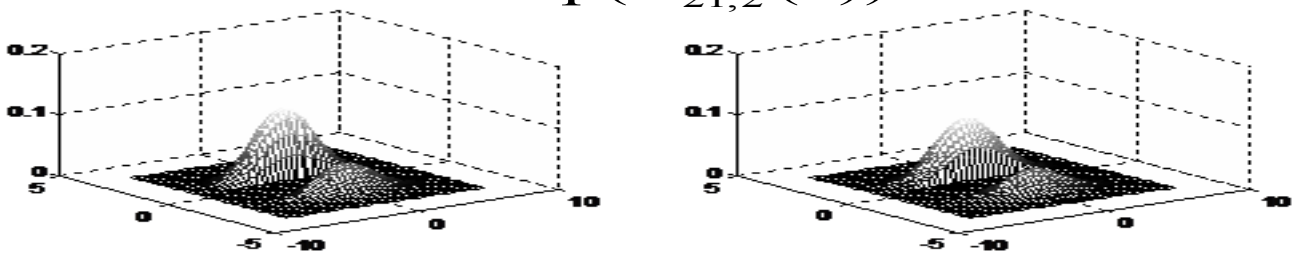
# Approximating Gabor function

## Learning gadaline networks

$y_7^+(\mathbf{z})$ $\exp(G_{21,2}^+(\mathbf{z}))$



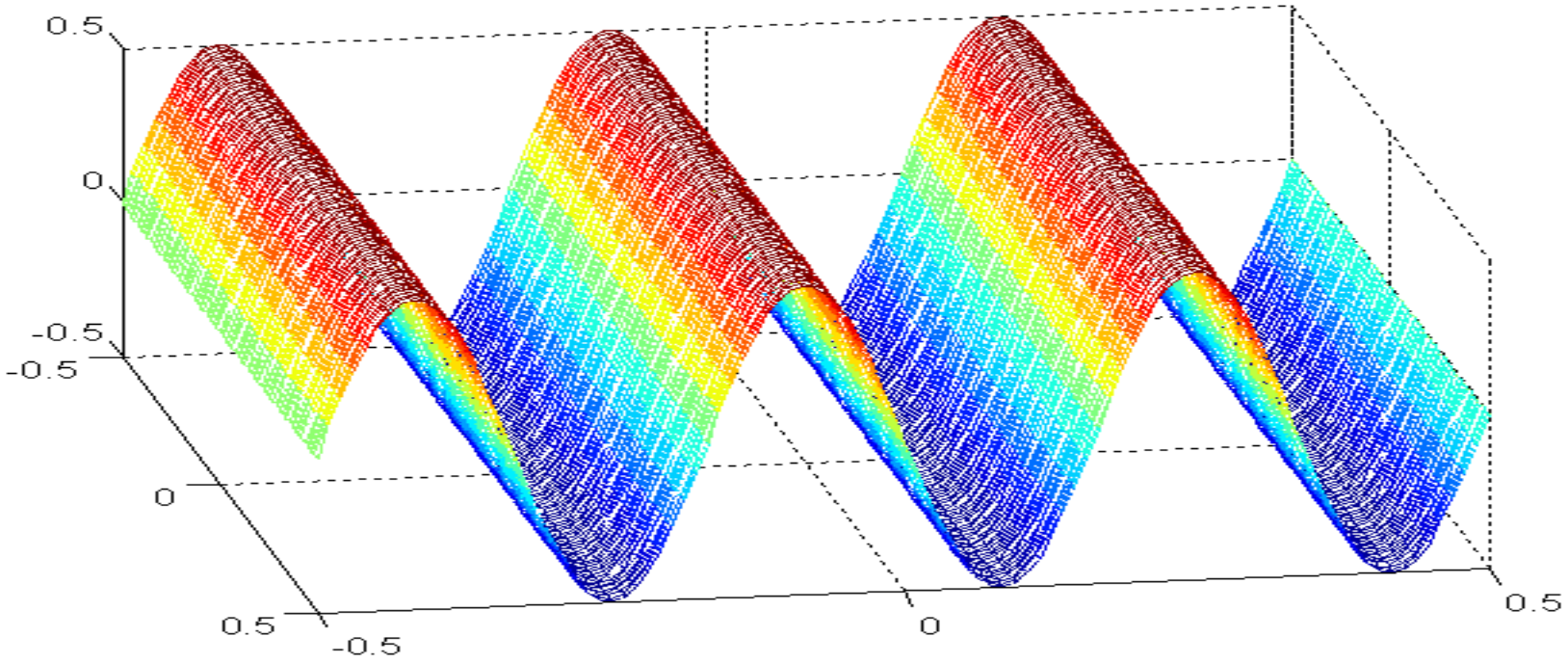
$y_7^-(\mathbf{z})$ $\exp(G_{21,2}^-(\mathbf{z}))$

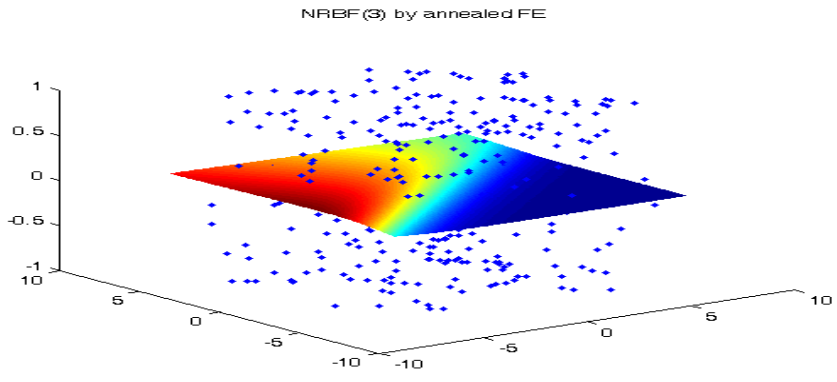


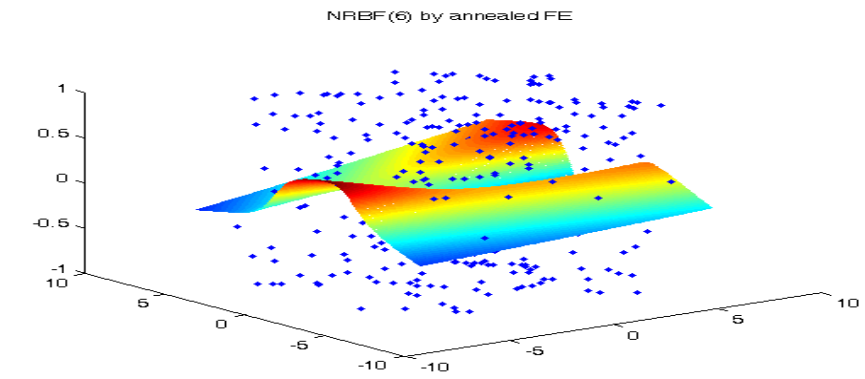
# Sinusoidal function approximation

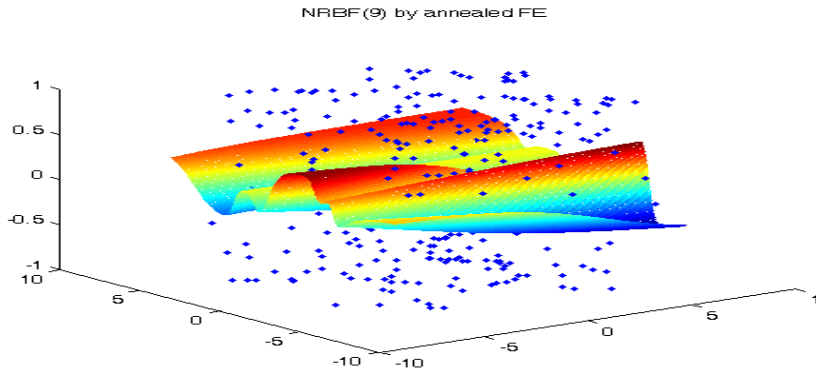
Learning gadaline networks (Wu et al 2006)

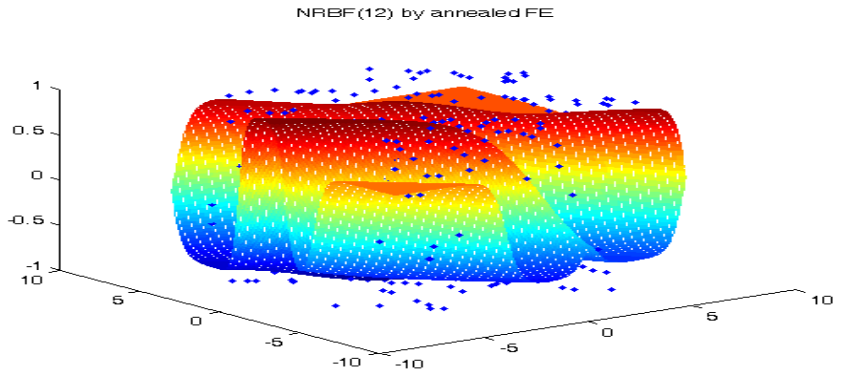
$y(\mathbf{z})$

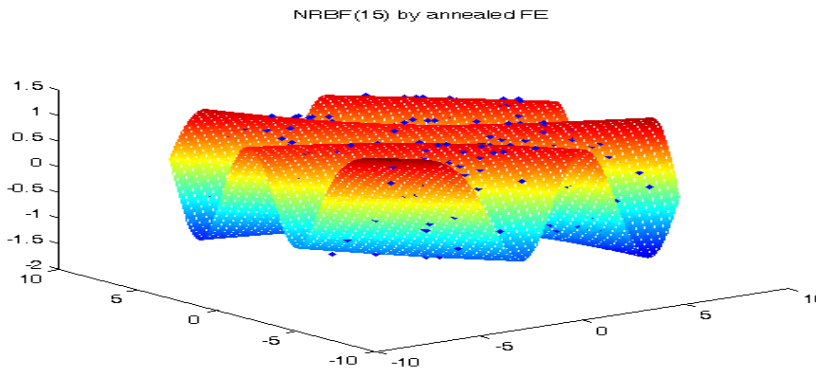


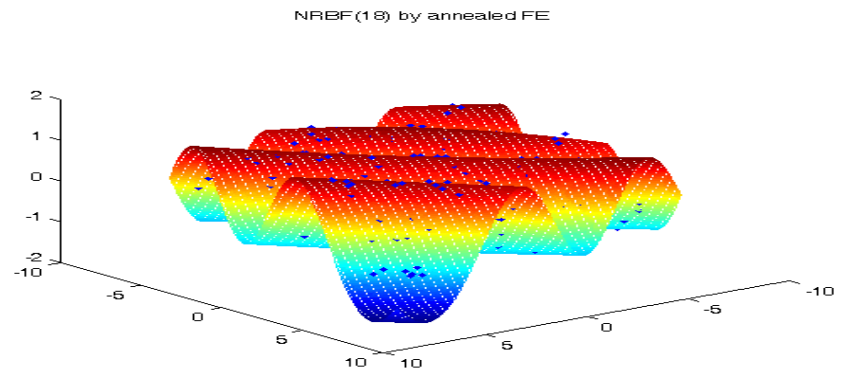

NRBF(3) by annealed FE

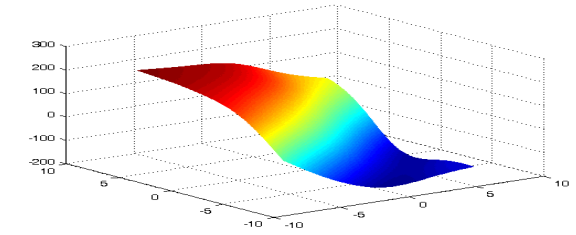

NRBF(6) by annealed FE

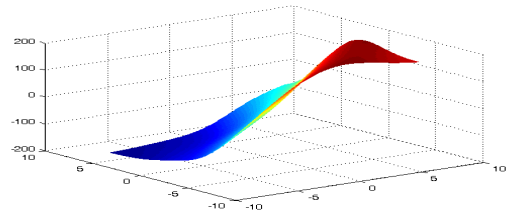

NRBF(9) by annealed FE

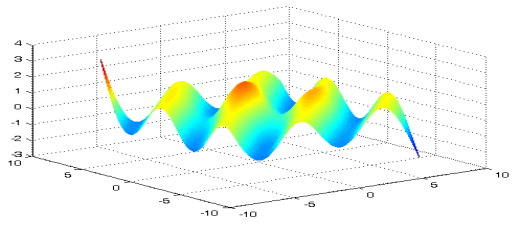

NRBF(12) by annealed FE


NRBF(15) by annealed FE

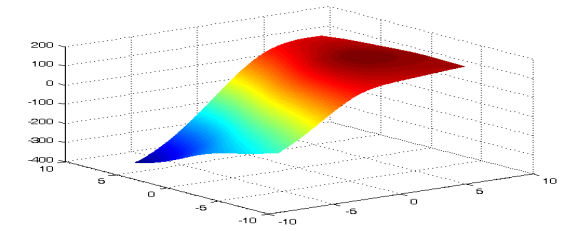

NRBF(18) by annealed FE

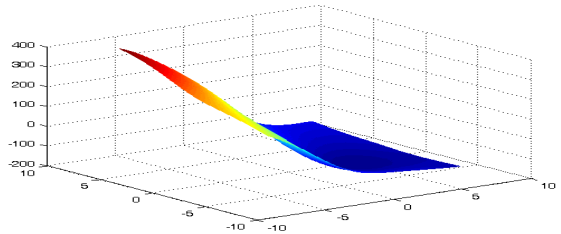


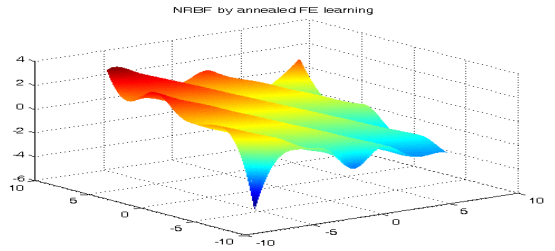




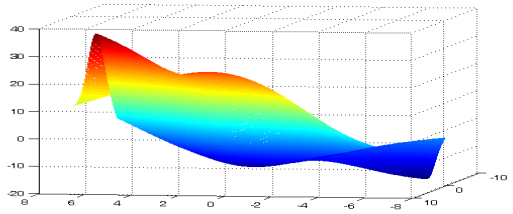
(a)

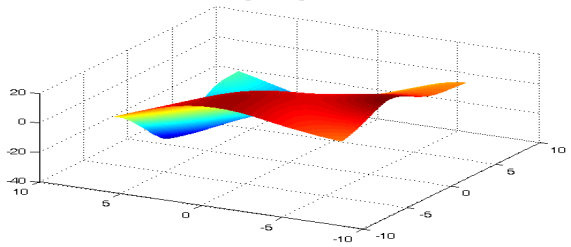


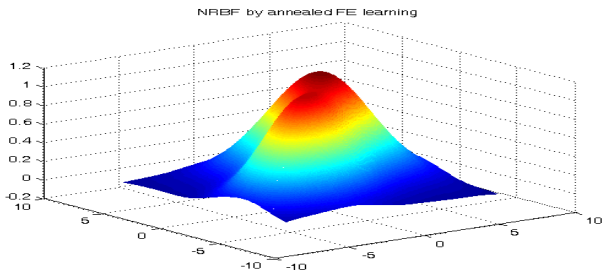




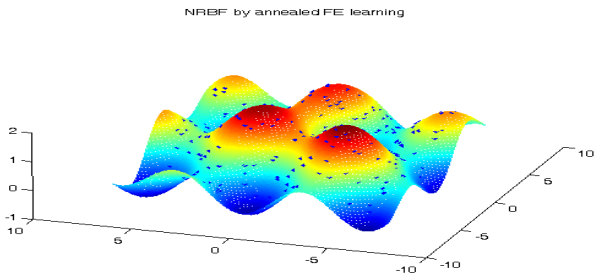

(b)





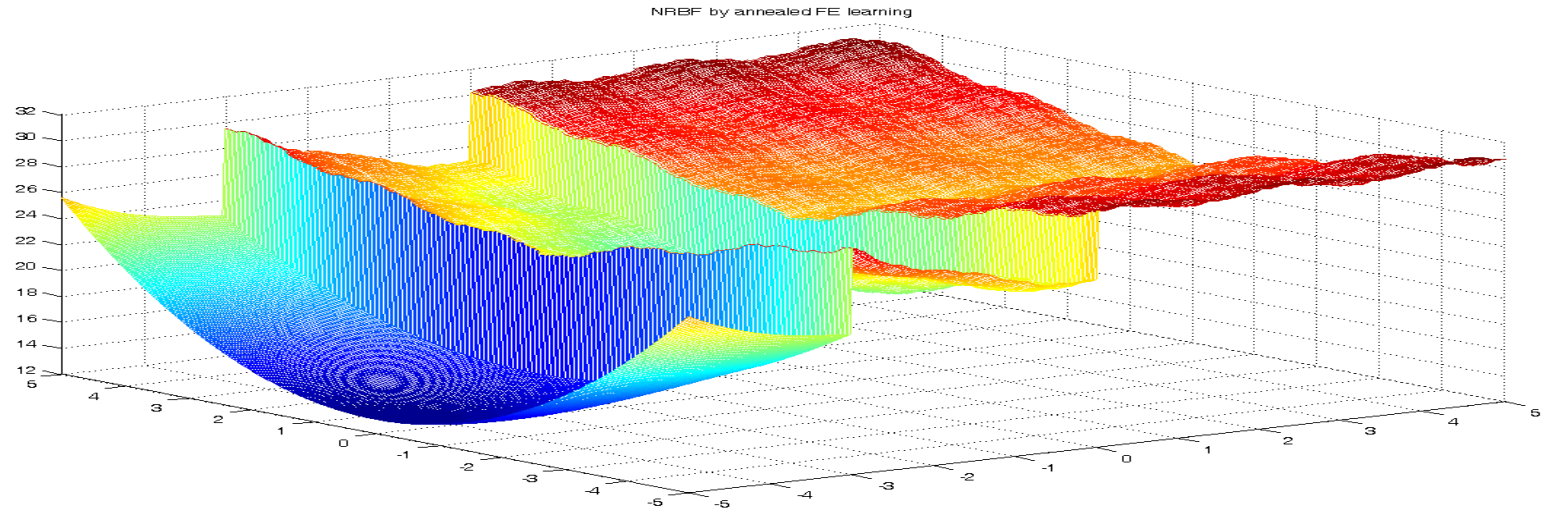



(c)

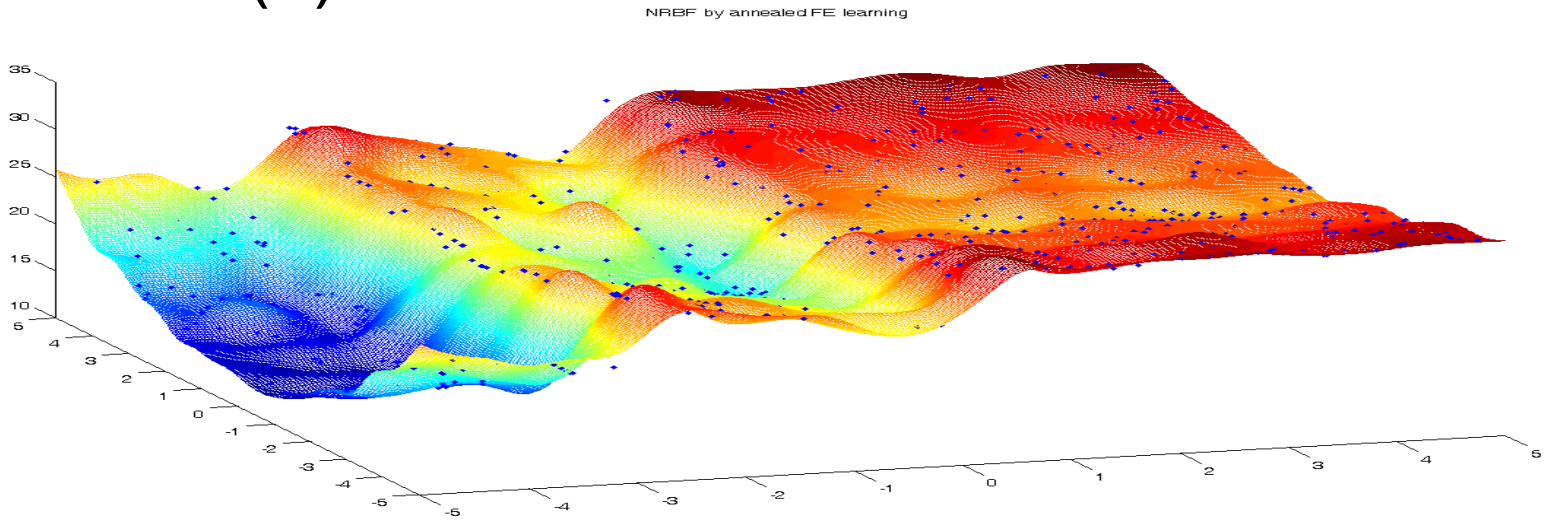



(d)

Figure 6




(a)




(b)

Figure 8

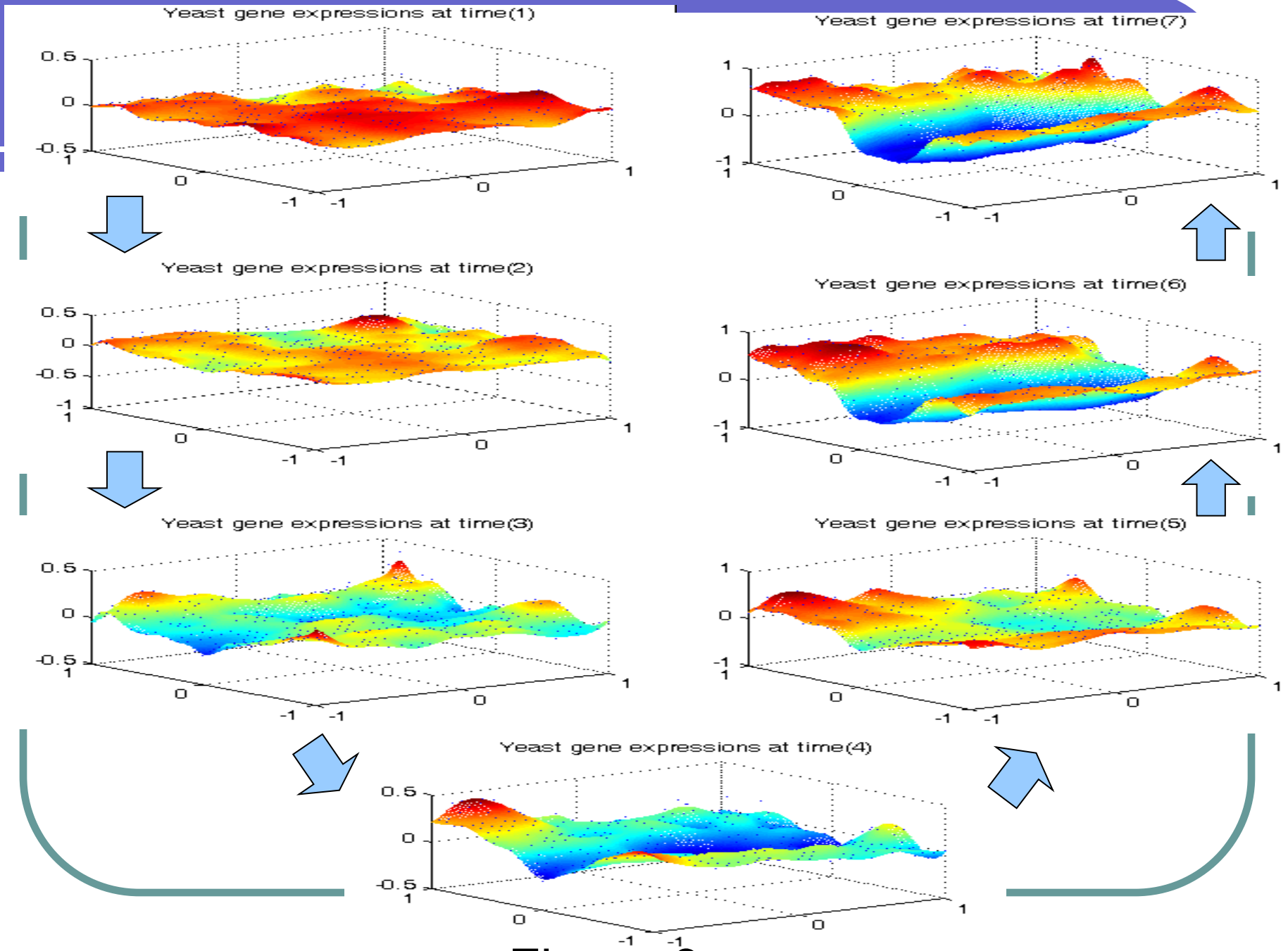

Yeast gene expressions at time(1)
Yeast gene expressions at time(2)
Yeast gene expressions at time(3)
Yeast gene expressions at time(4)
Yeast gene expressions at time(5)
Yeast gene expressions at time(6)
Yeast gene expressions at time(7)

Figure 9

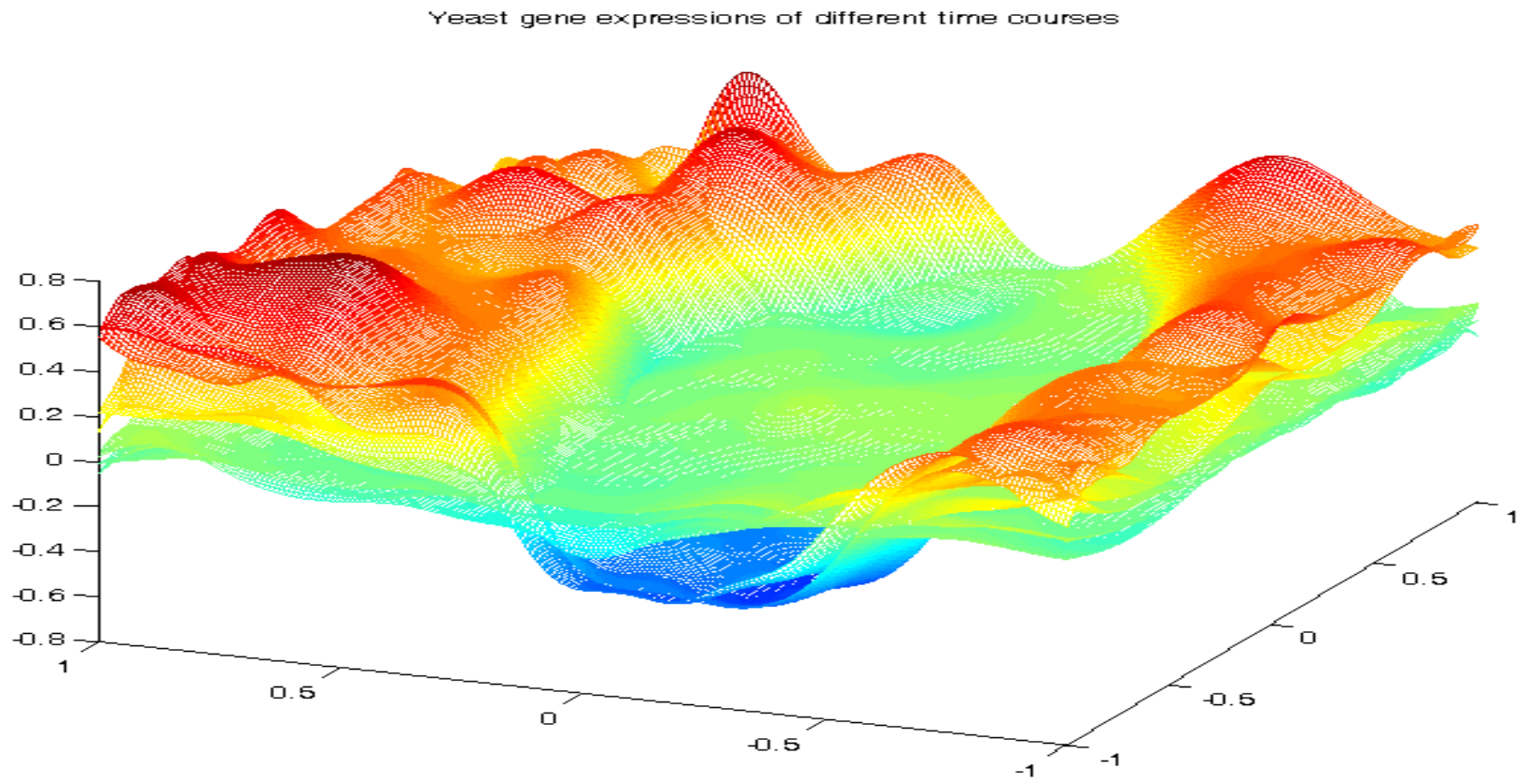
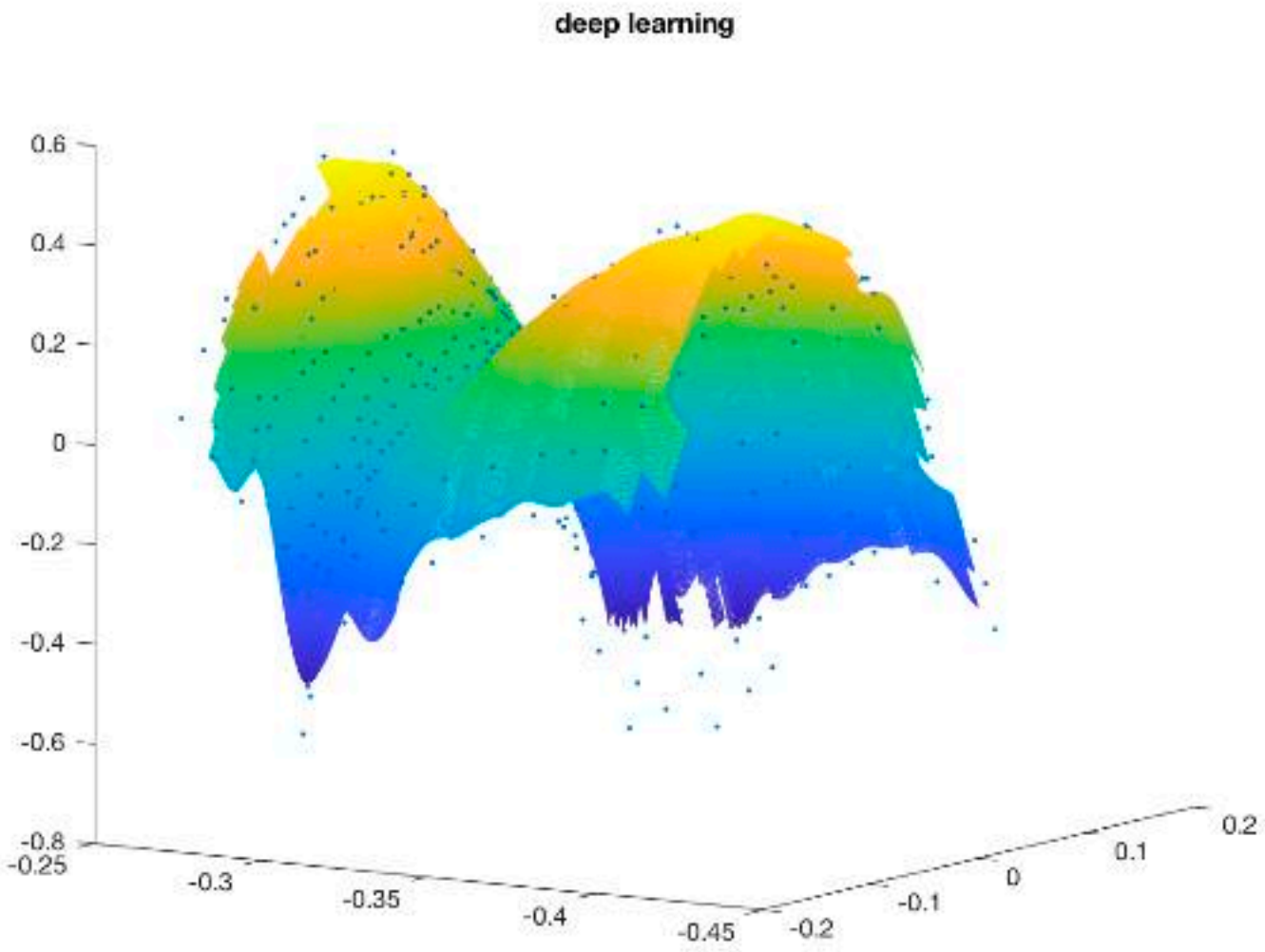
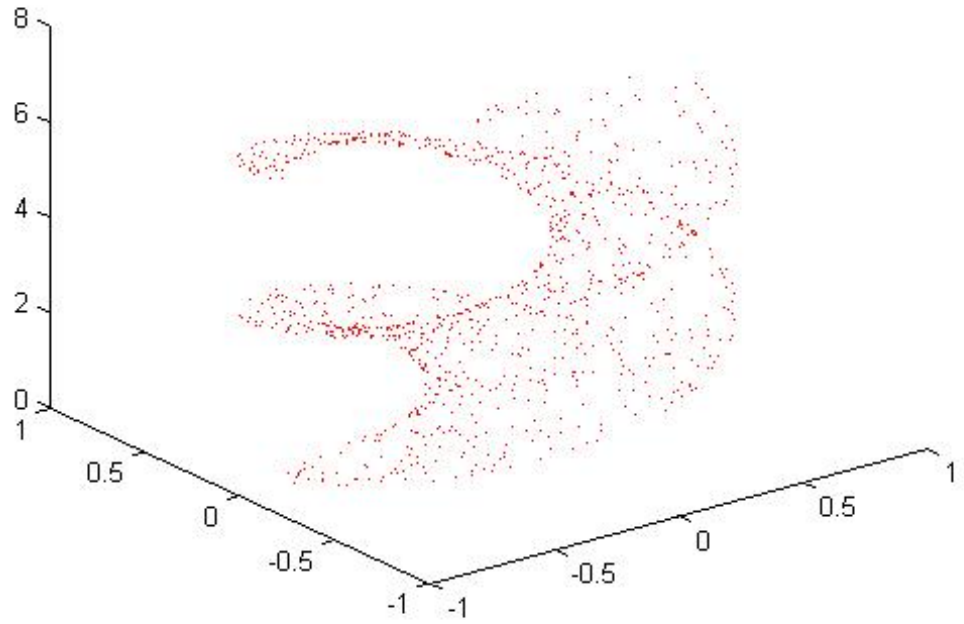

Yeast gene expressions of different time courses
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
1
0.5
0
-0.5
-1
1
0.5
0
-0.5
-1

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

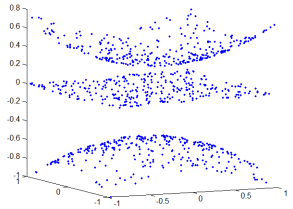
# One-to-many function approximation to $g_1$



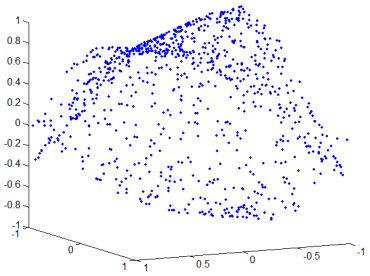

- P=5;k=M=30;N=1000

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

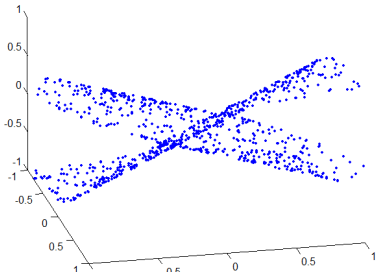
$$a_1 \in (0, 2\pi)$$



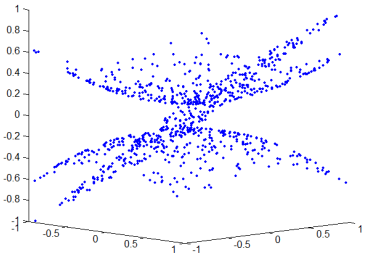
$$f_1(x_1, x_2) = x_1^2 + x_2^2 + 1$$
$$f_2(x_1, x_2) = -x_1^2 - x_2^2 - 2$$
$$f_3(x_1, x_2) = 0.1x_1 + 0.1x_2$$



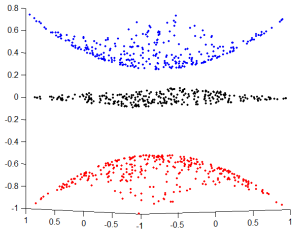
$$f_1(x_1, x_2) = \sin(x_1 + x_2)$$
$$f_2(x_1, x_2) = \cos(x_1 - x_2)$$

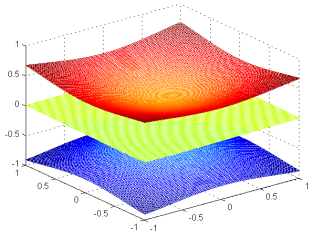


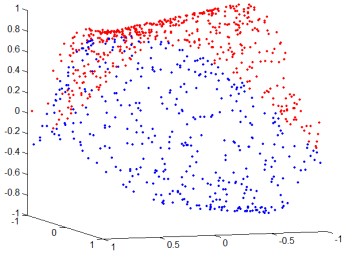
$$f_1(x_1, x_2) = \tanh(x_1 + x_2)$$
$$f_2(x_1, x_2) = \tanh(x_1 - x_2)$$

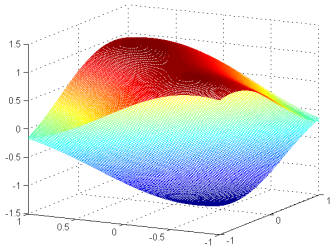


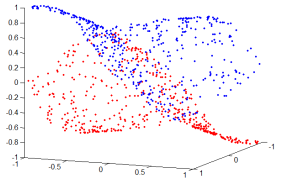
$$f_1(x_1, x_2) = x_1^2 + x_2^2 + 0.5$$
$$f_2(x_1, x_2) = -x_1^2 - x_2^2 - 0.5$$
$$f_3(x_1, x_2) = 2x_1 + 2x_2$$

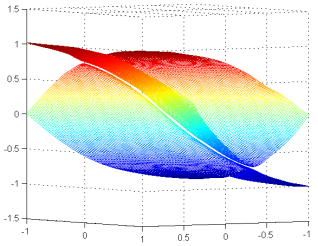


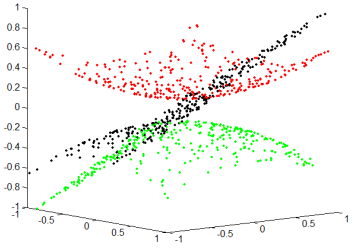


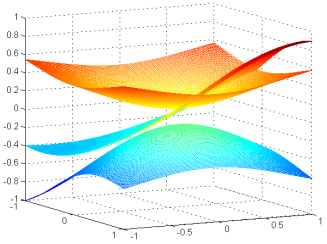




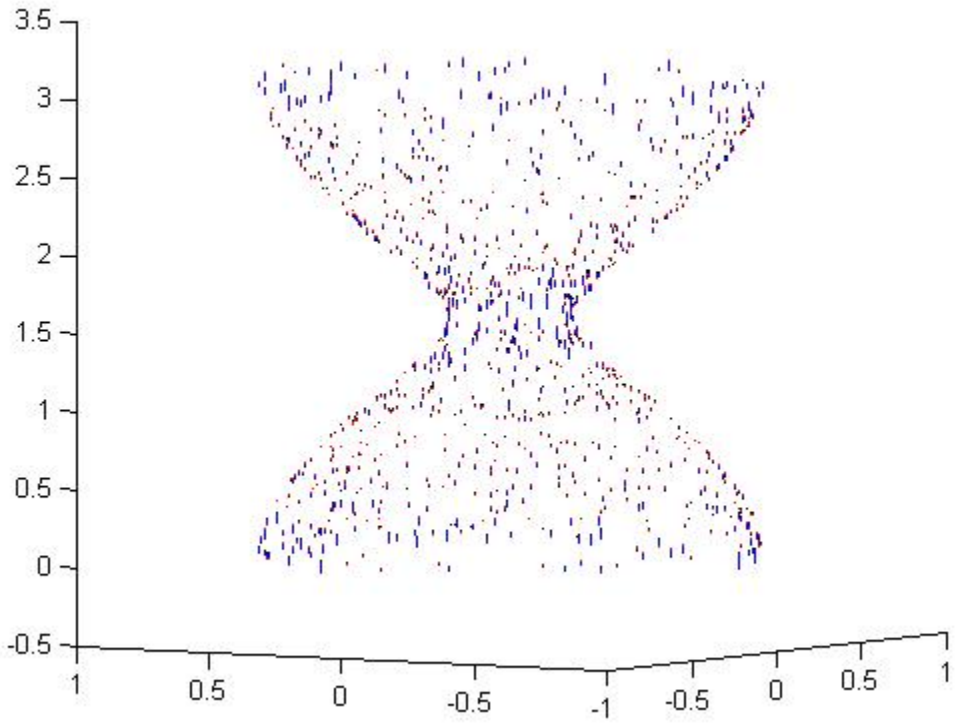


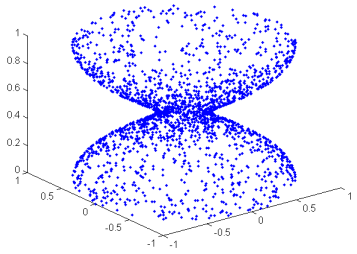


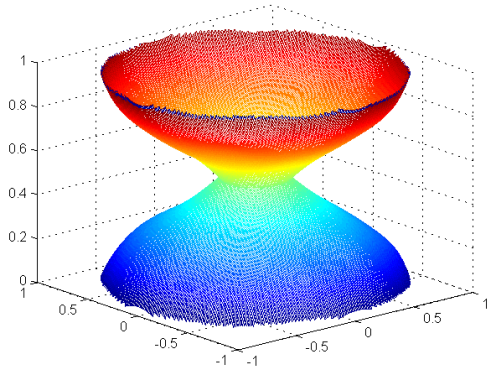


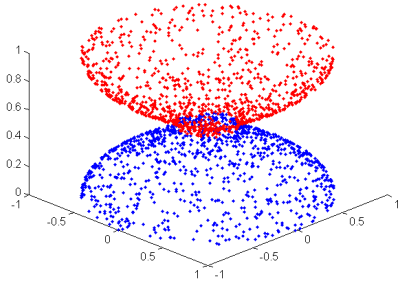


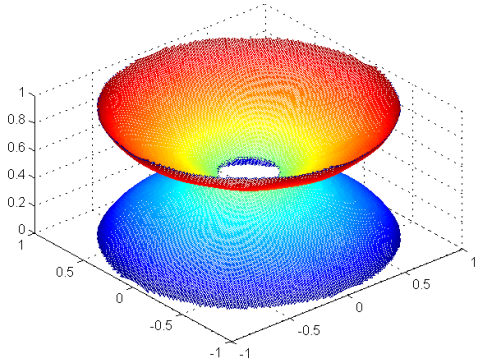
# Approximating functions



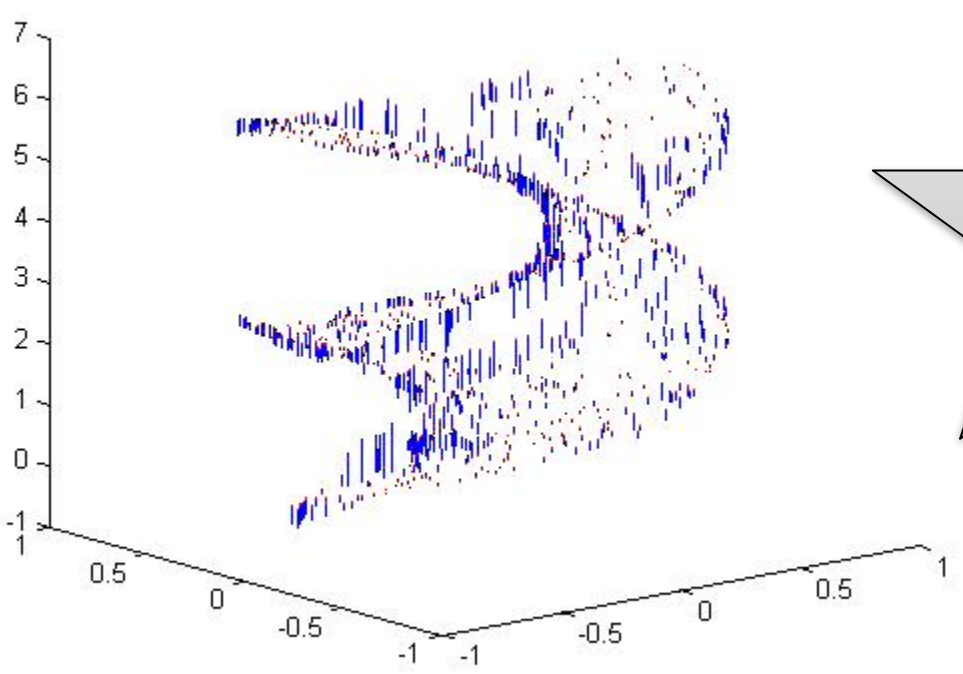

1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

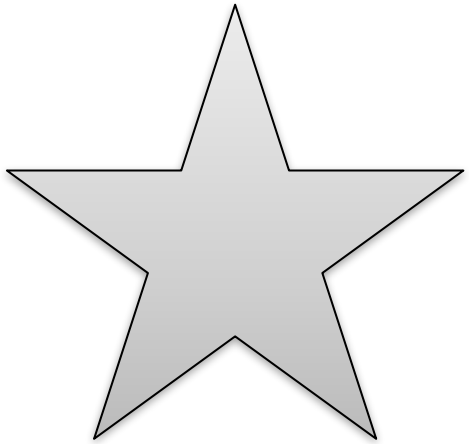

1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

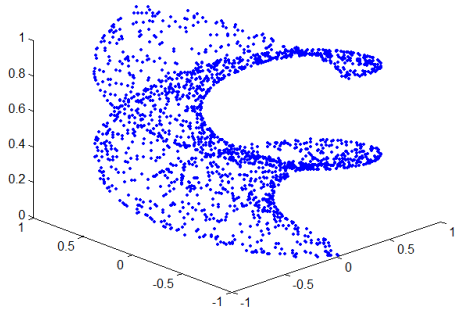

1
0.8
0.6
0.4
0.2
0
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1

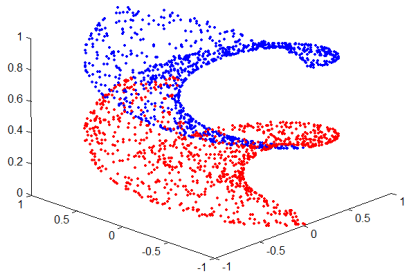

1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

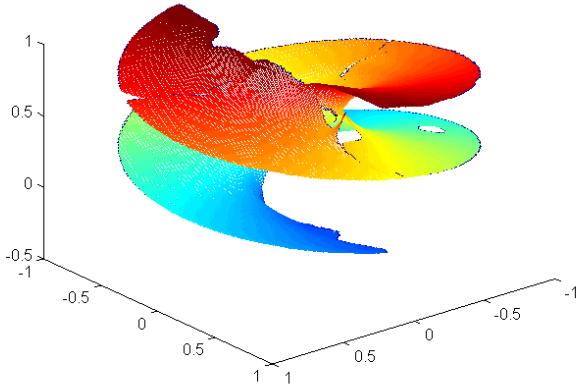
# Approximating functions



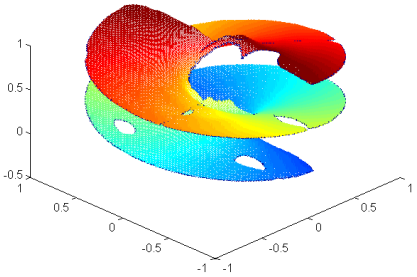




1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1

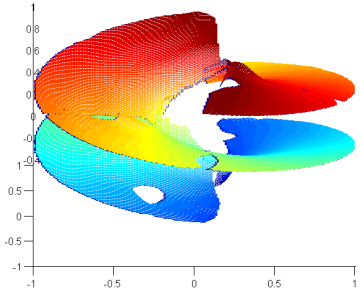

1
0.8
0.6
0.4
0.2
0
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1


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

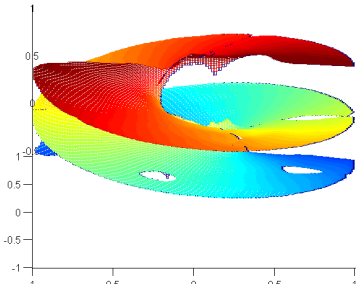
0°



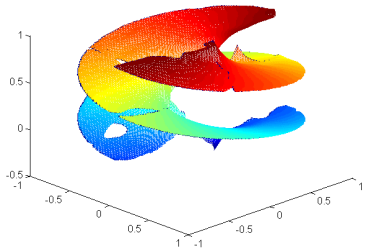
315°



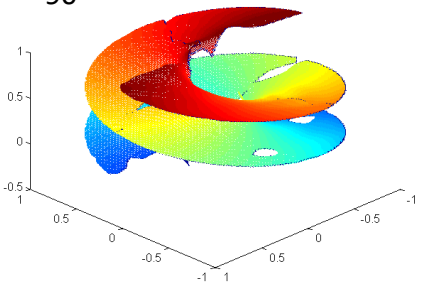
45°



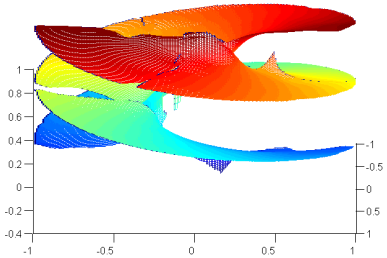
270°



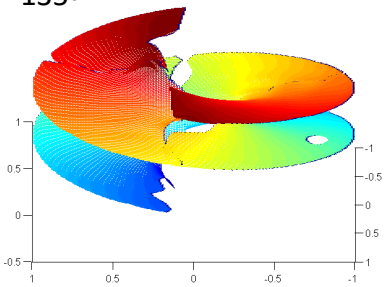
90°



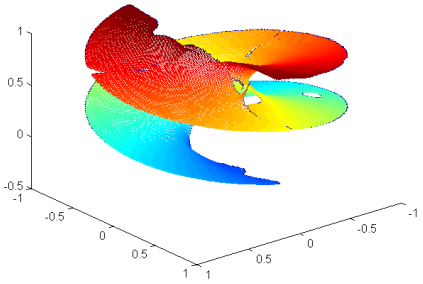
225°

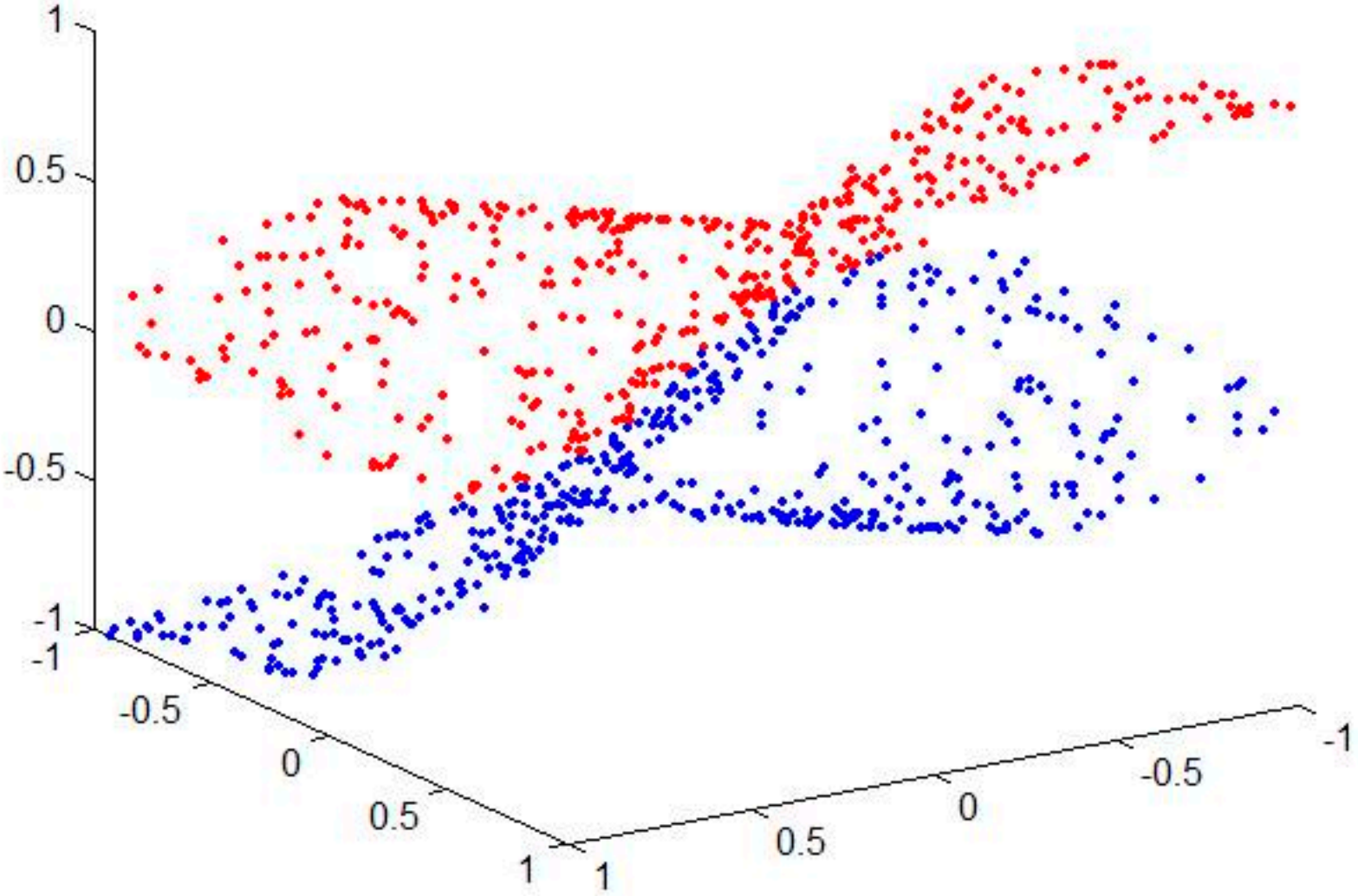


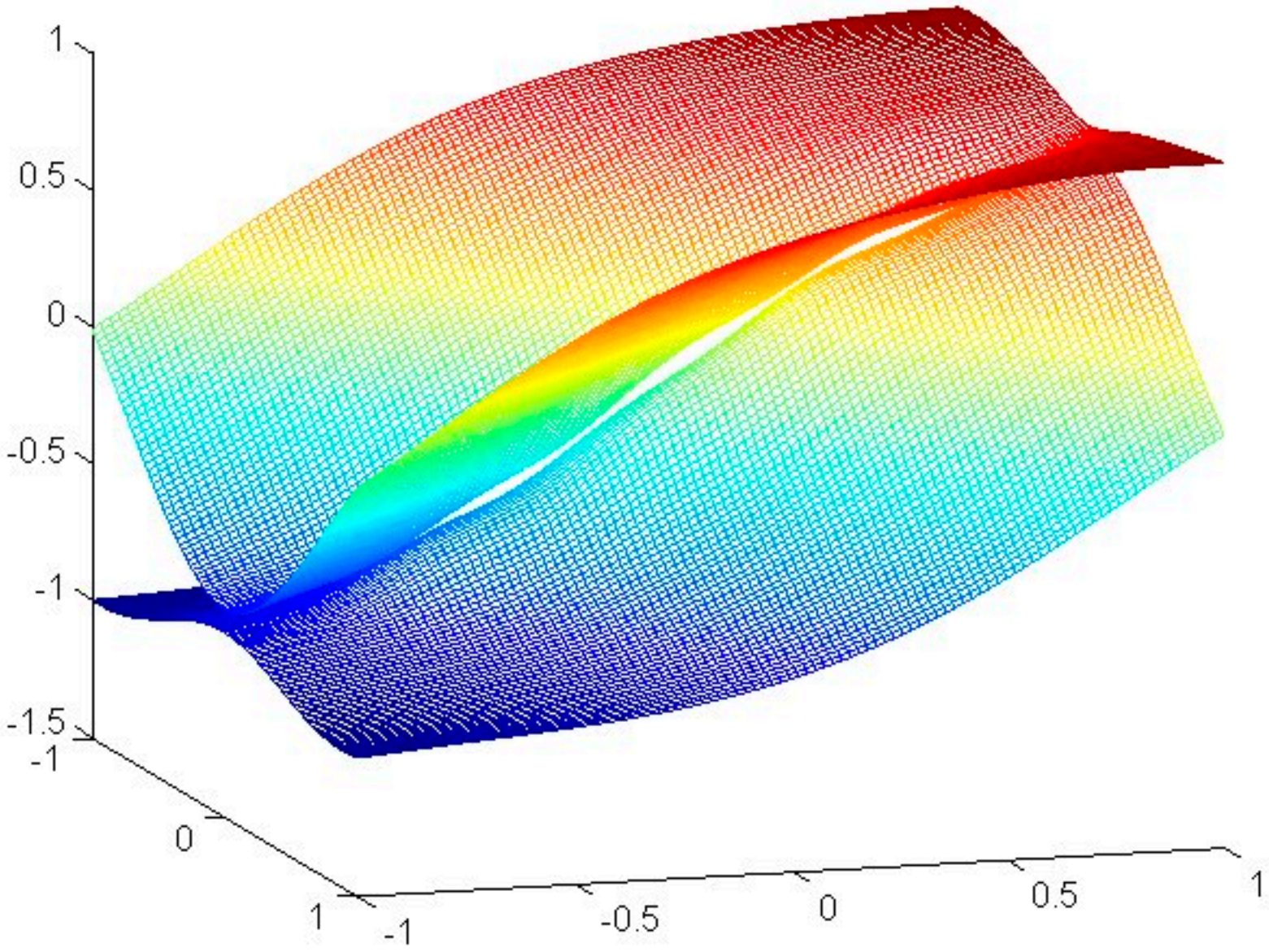
135°



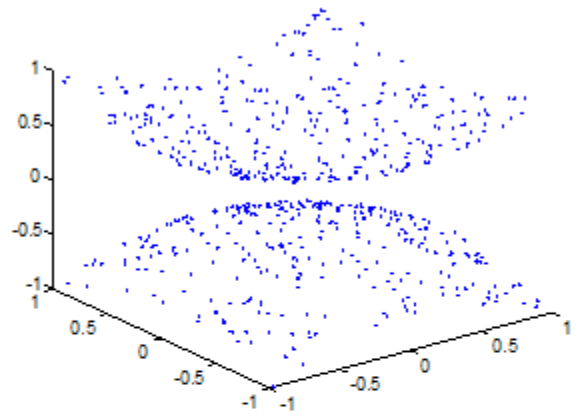
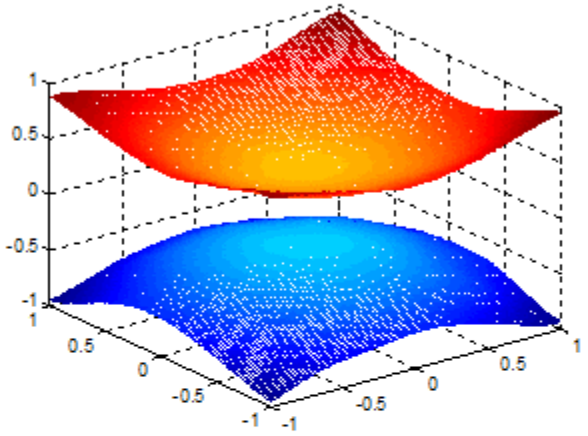
180°




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

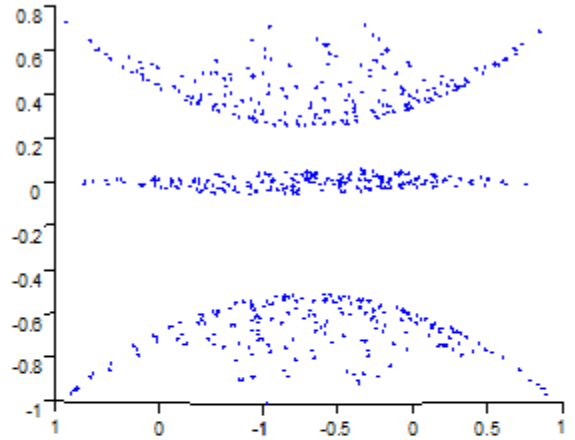
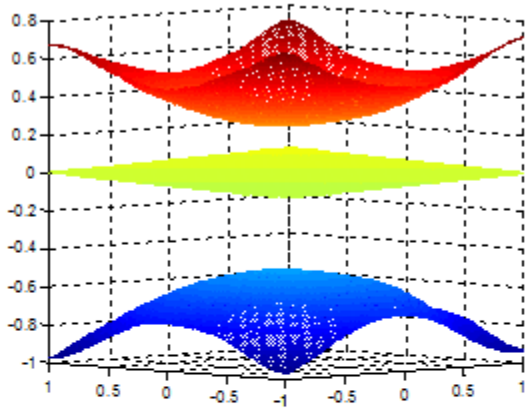

1
0.5
0
-0.5
-1
-1.5
-1
0
1
-1
-0.5
0
0.5
1

# 2-type deterministic transition

| Original data | deterministic rule |
| --- | --- |
|  |  |

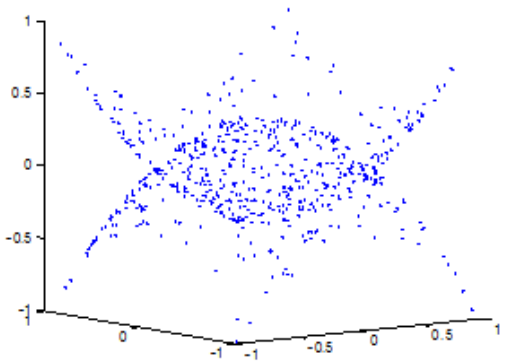
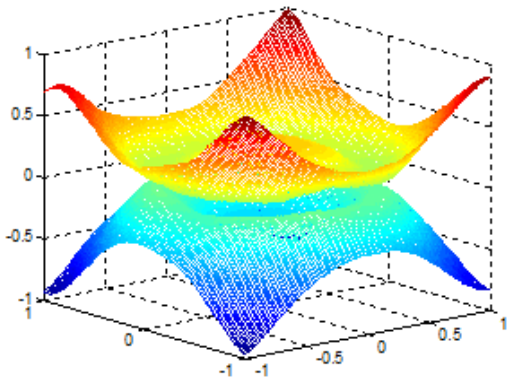
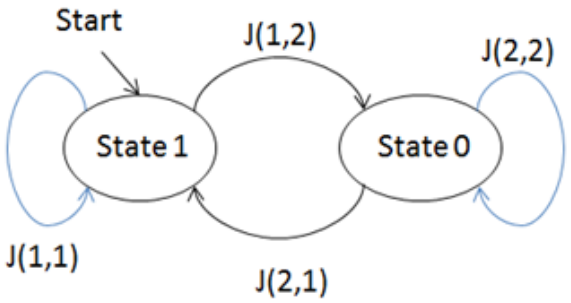
| | target prediction (learning) | state inference | target prediction (aenn) |
| --- | --- | --- | --- |
| error | 0.000257 | 0.000758 | 0.000281 |

# 3-type deterministic transition

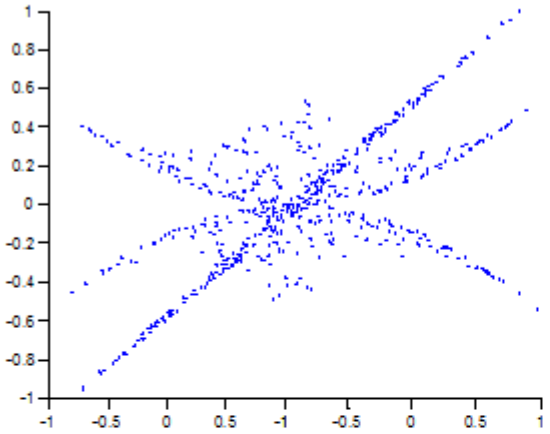
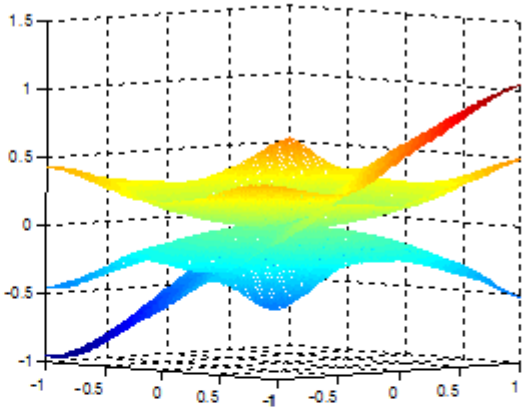
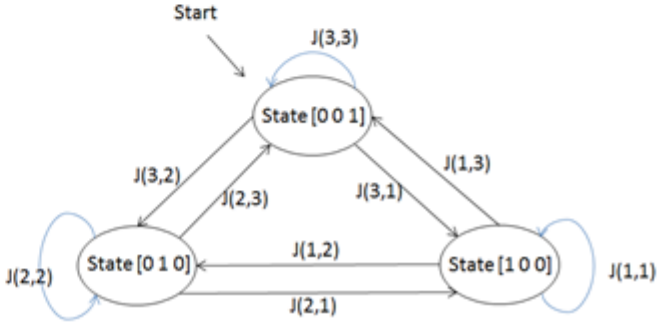
| Original data | deterministic rule |
| --- | --- |
|  |  |

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

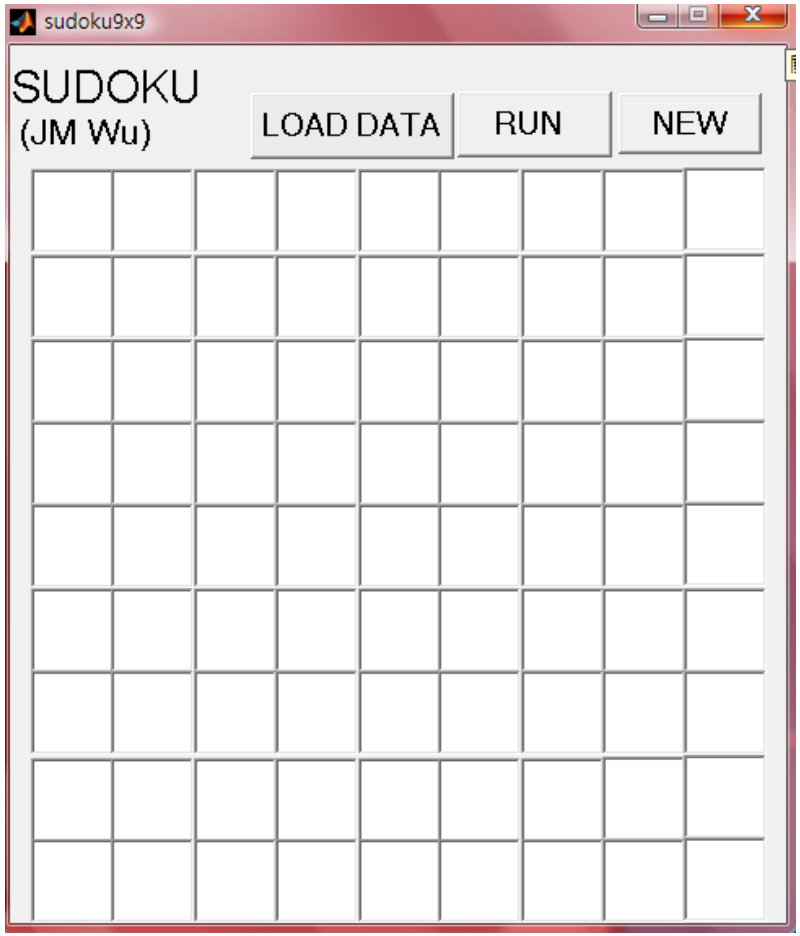
# 2-type Stochastic transition

| Original data | Stochastic rule |
| --- | --- |
|  |  |
| **Probability** | **Stochastic automata state transition** |
| J = <br> 0.4045 0.5955 <br> 0.6345 0.3655 |   |

# 3-type Stochastic transition

| Original data | Stochastic rule |
| --- | --- |
|  |  |
| **Probability** | **Stochastic automata state transition** |
| J = <br> 0.1181 0.7014 0.1806 <br> 0.2925 0.7075 0 <br> 0.0645 0.7742 0.1613 |  |

# 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

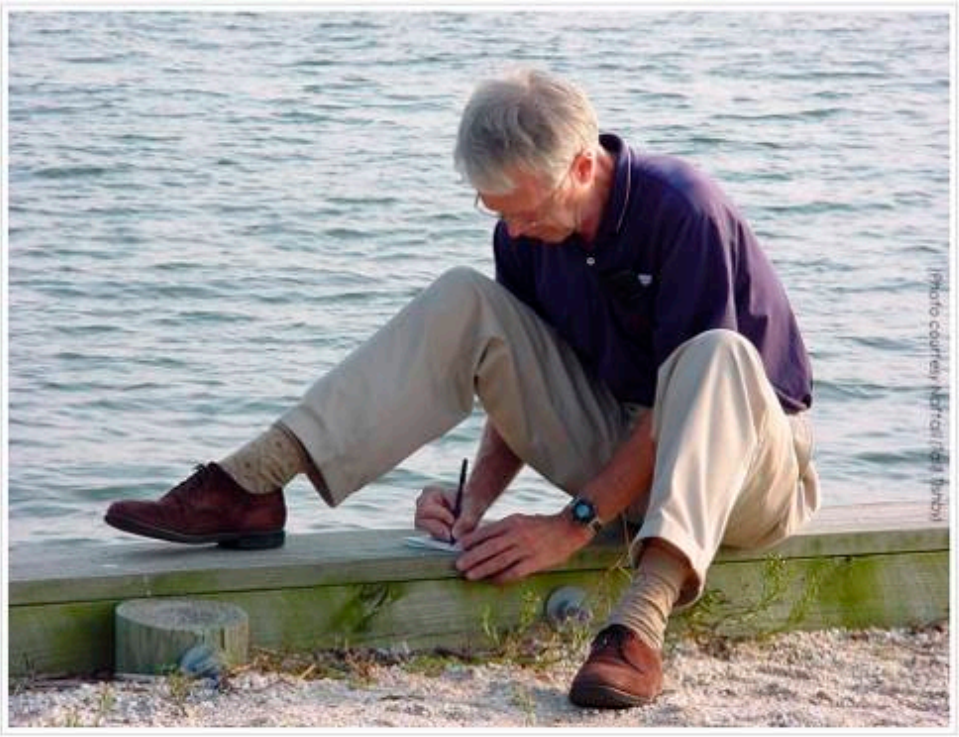
# 速讀與速讀聯想記憶體

Donald Olding Hebb



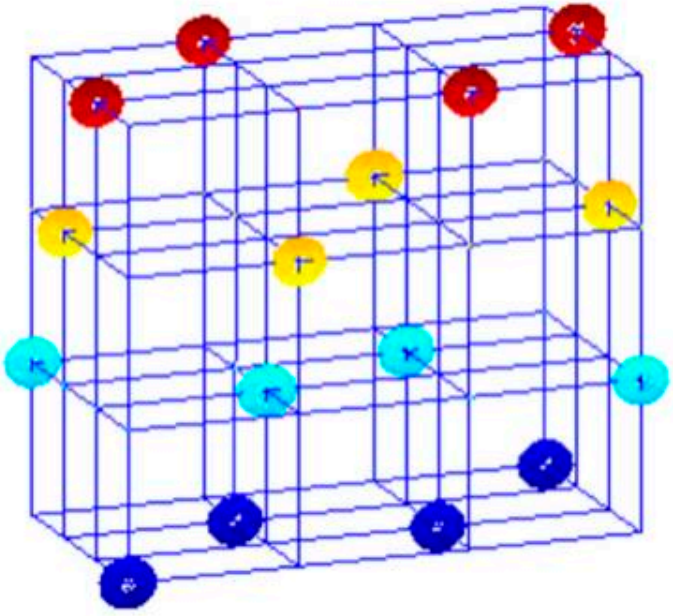
John Joseph Hopfield

Neural Networks



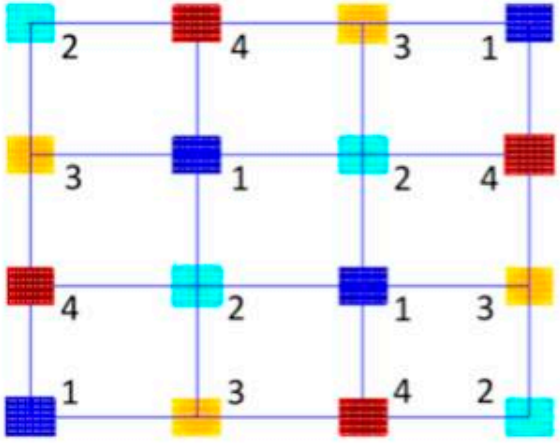
# Sudoku

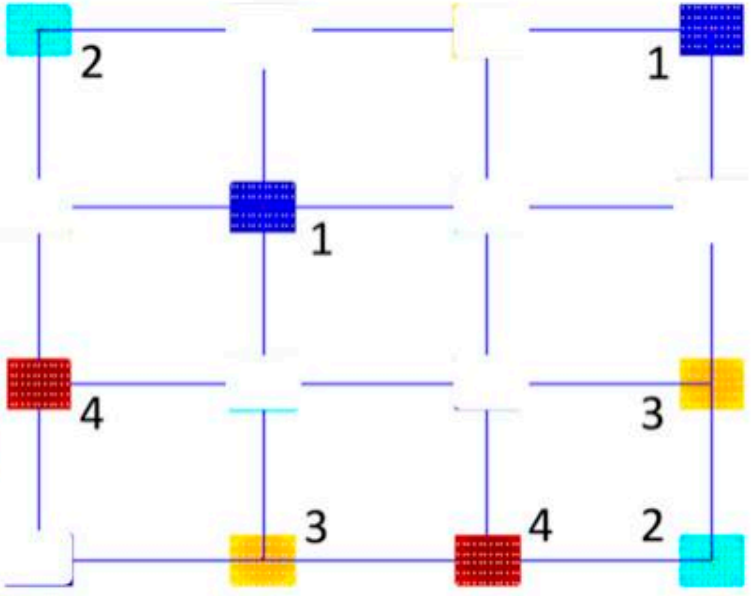
整數規劃數學求解
0 與 1的數學規劃式



S =

| | | | |
|---|---|---|---|
| 2 | 4 | 3 | 1 |
| 3 | 1 | 2 | 4 |
| 4 | 2 | 1 | 3 |
| 1 | 3 | 4 | 2 |



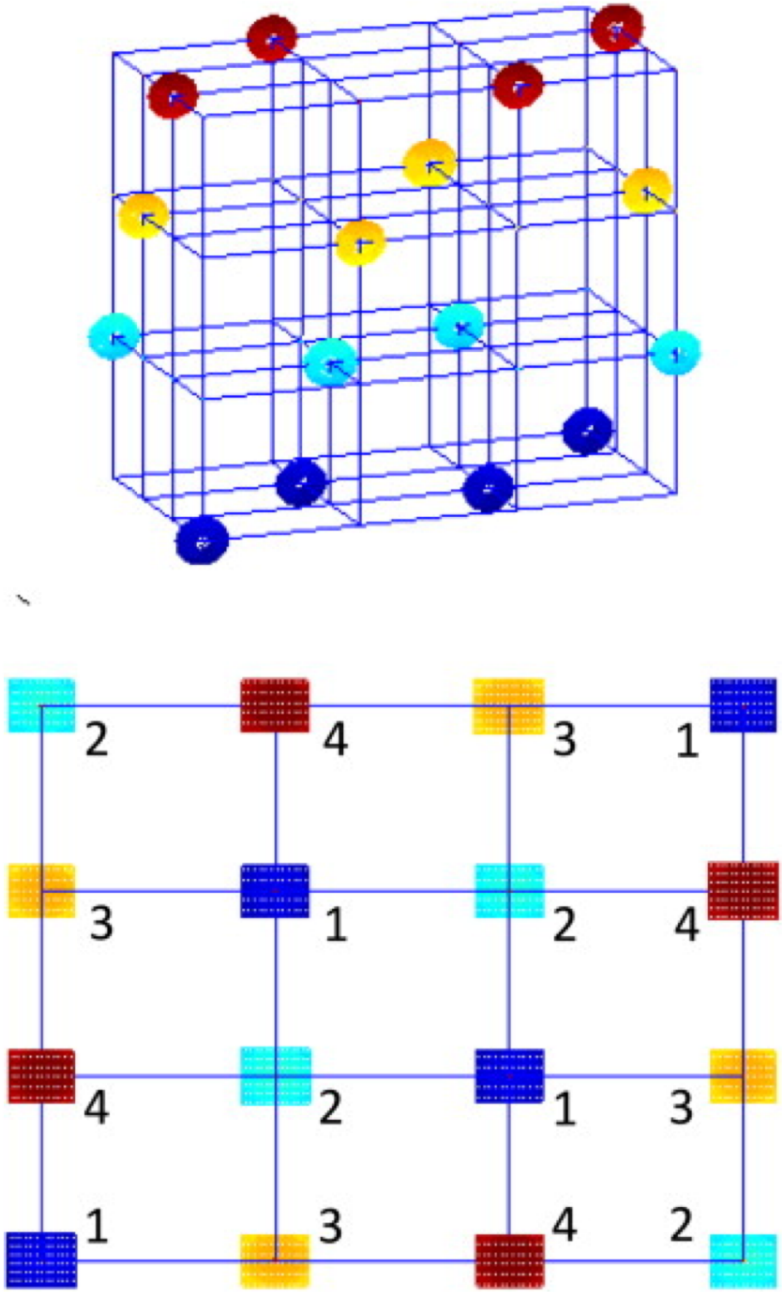



Fig. 6. Bipolar neural activations for Sudoku encoding of $K = 4$.

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.

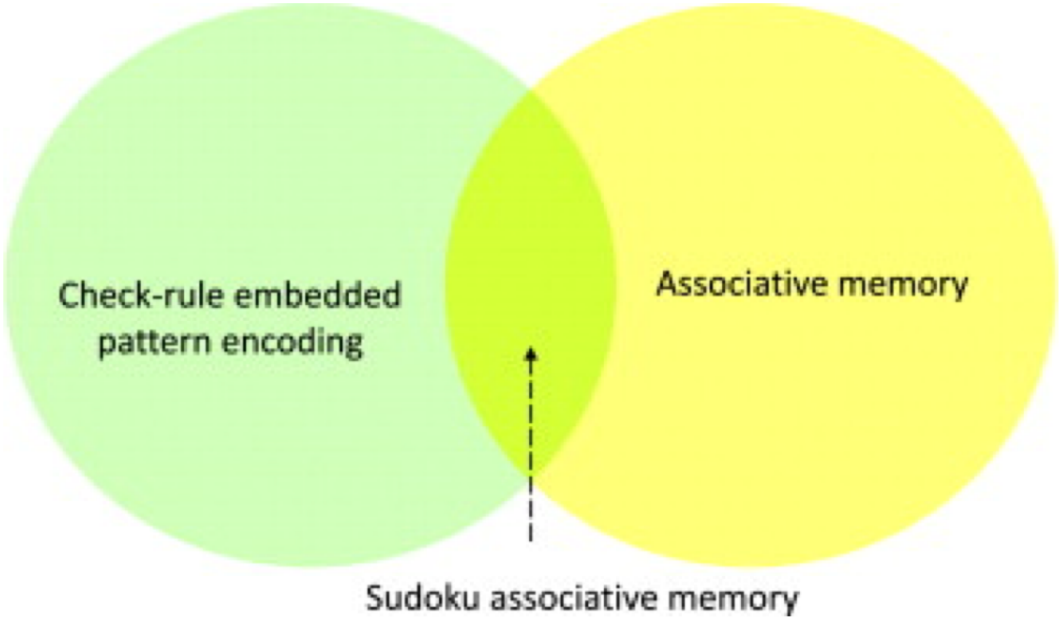
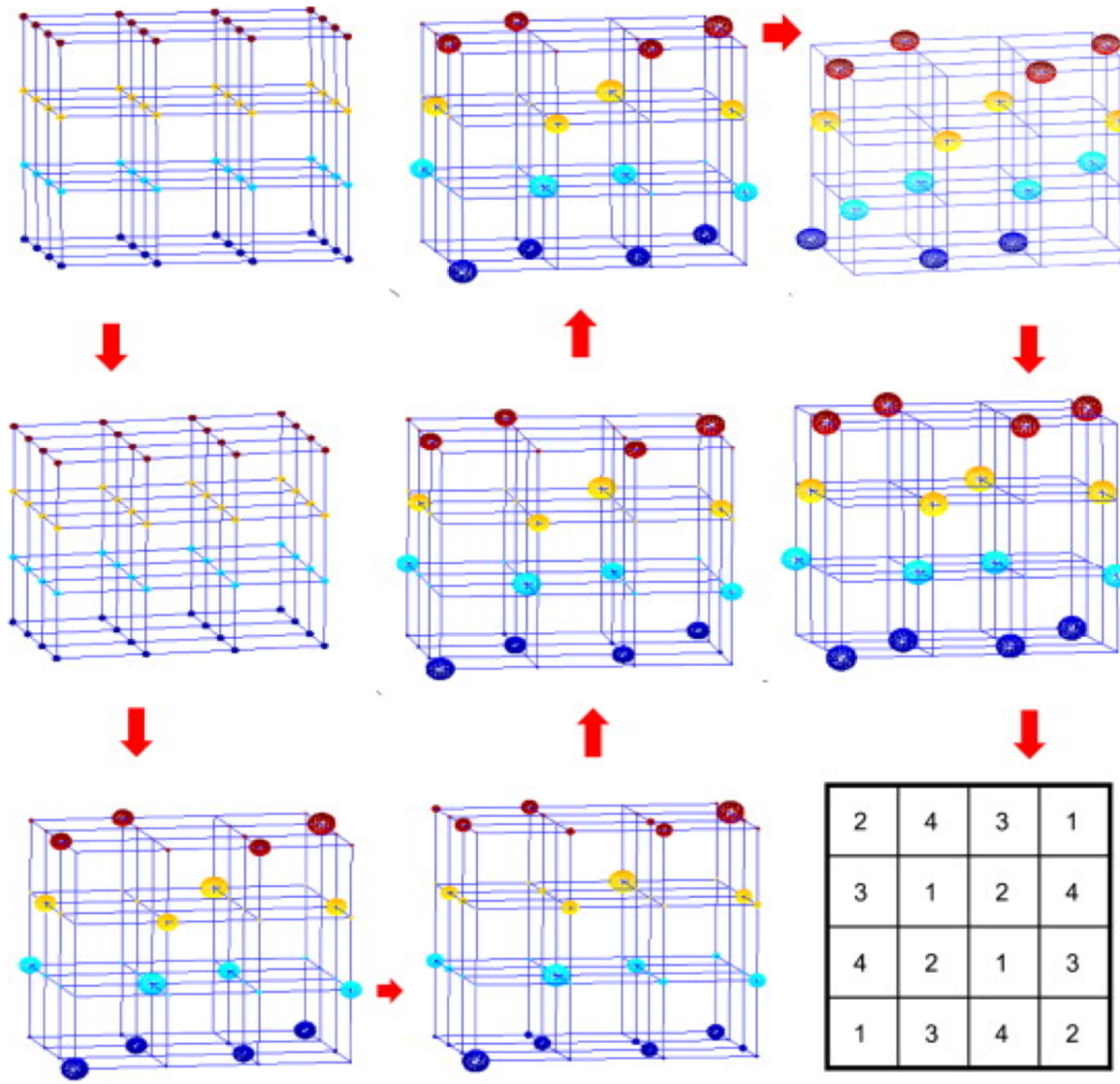



Fig. 7. The concept of developing SAM based on check-rule embedded pattern encoding and associative memory.



| | | | |
|---|---|---|---|
| 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.

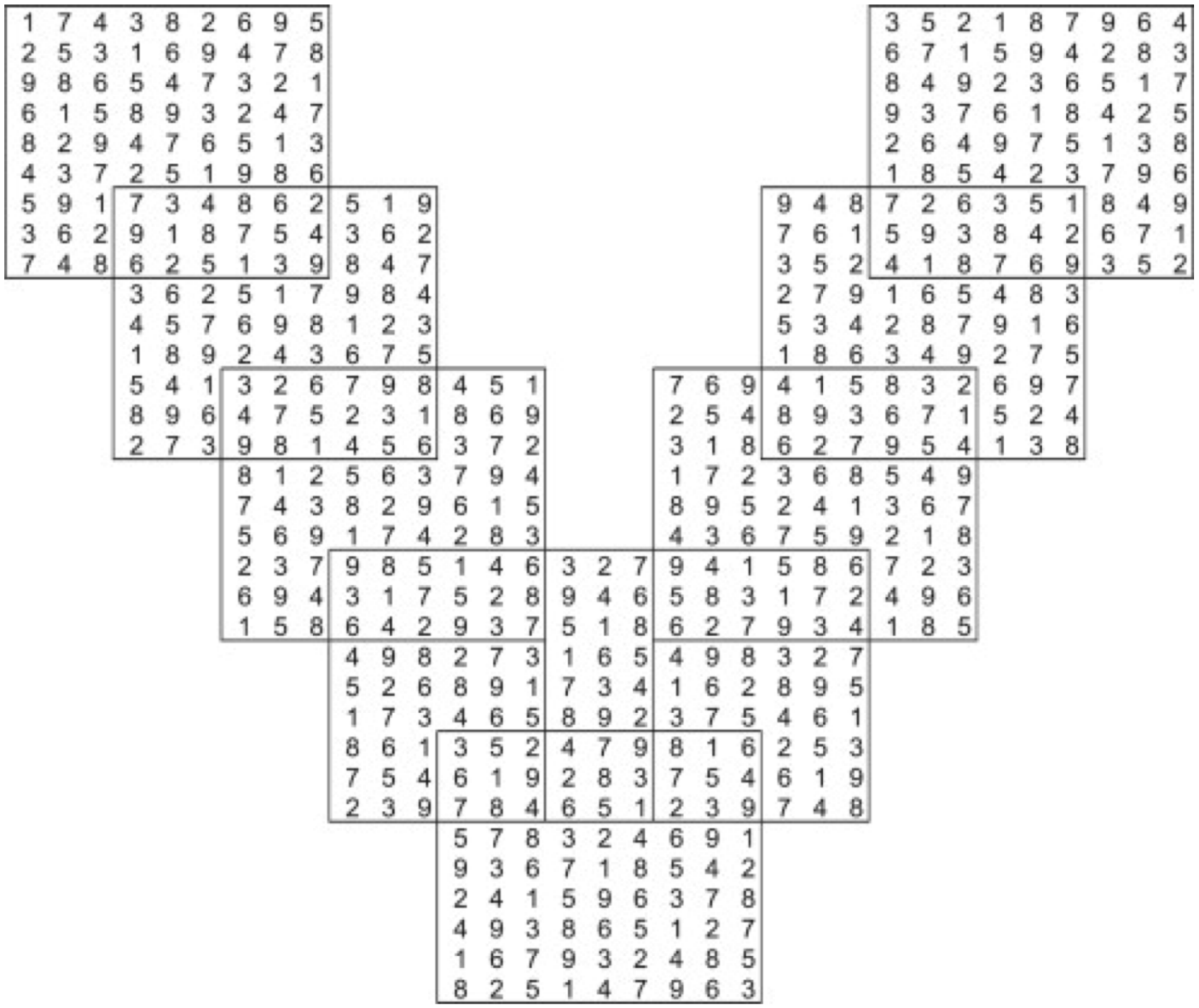



Fig. 12. A V-shape compound Sudoku pattern.

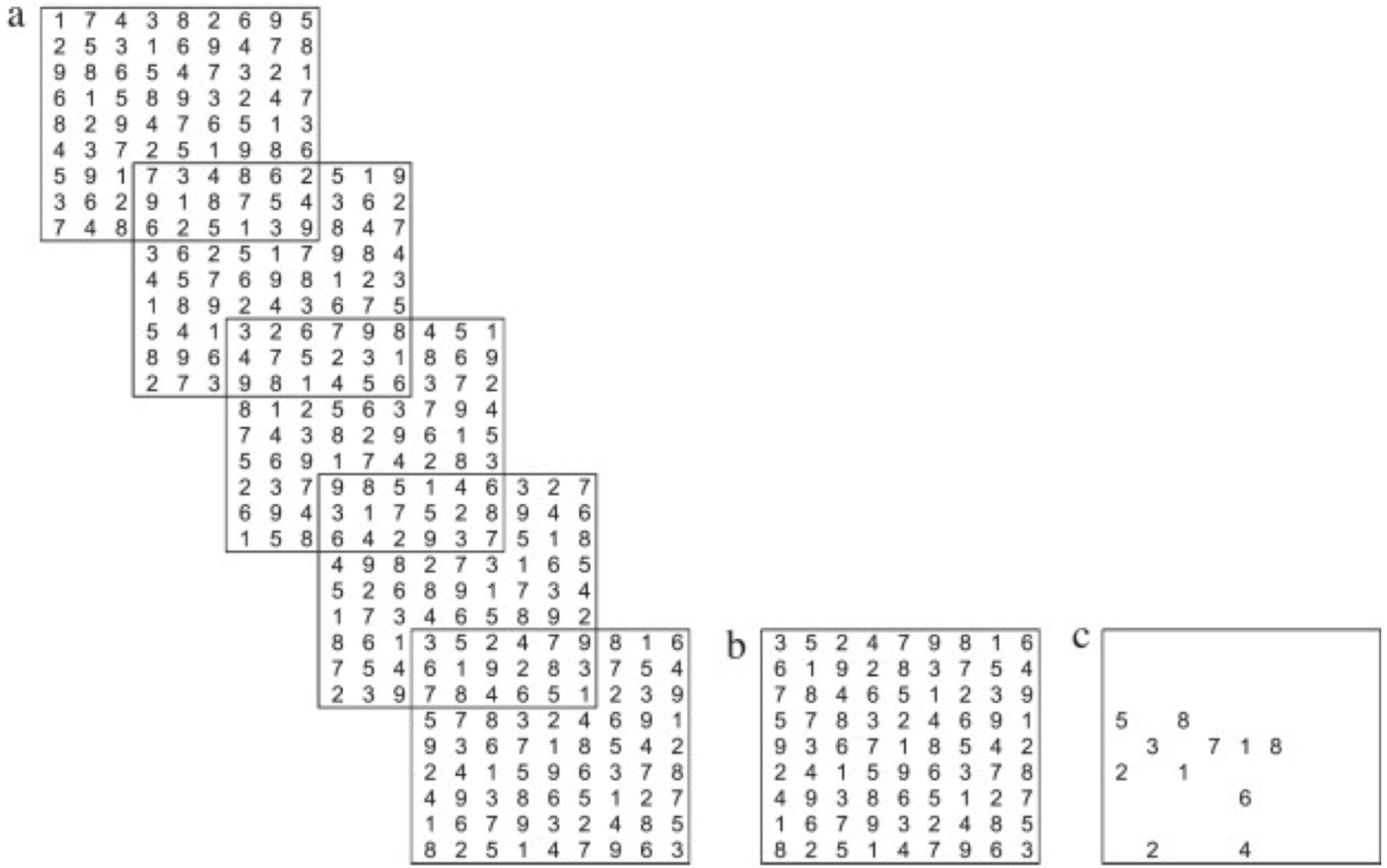



Fig. 13. Different partial clues for V-shape compound Sudoku pattern restoration, (a) a left part of

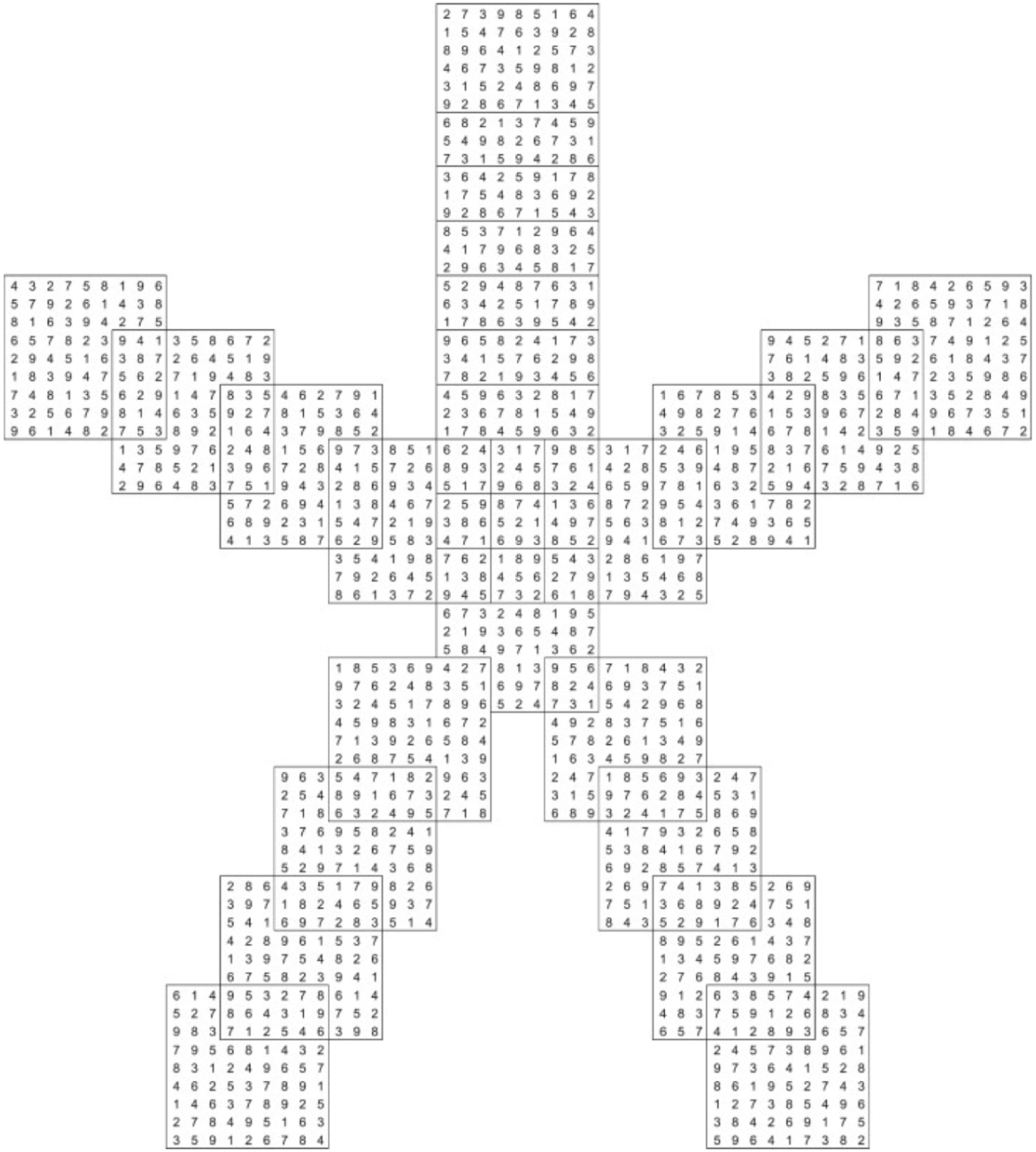


Fig. 14. A starfish-shape compound Sudoku pattern.

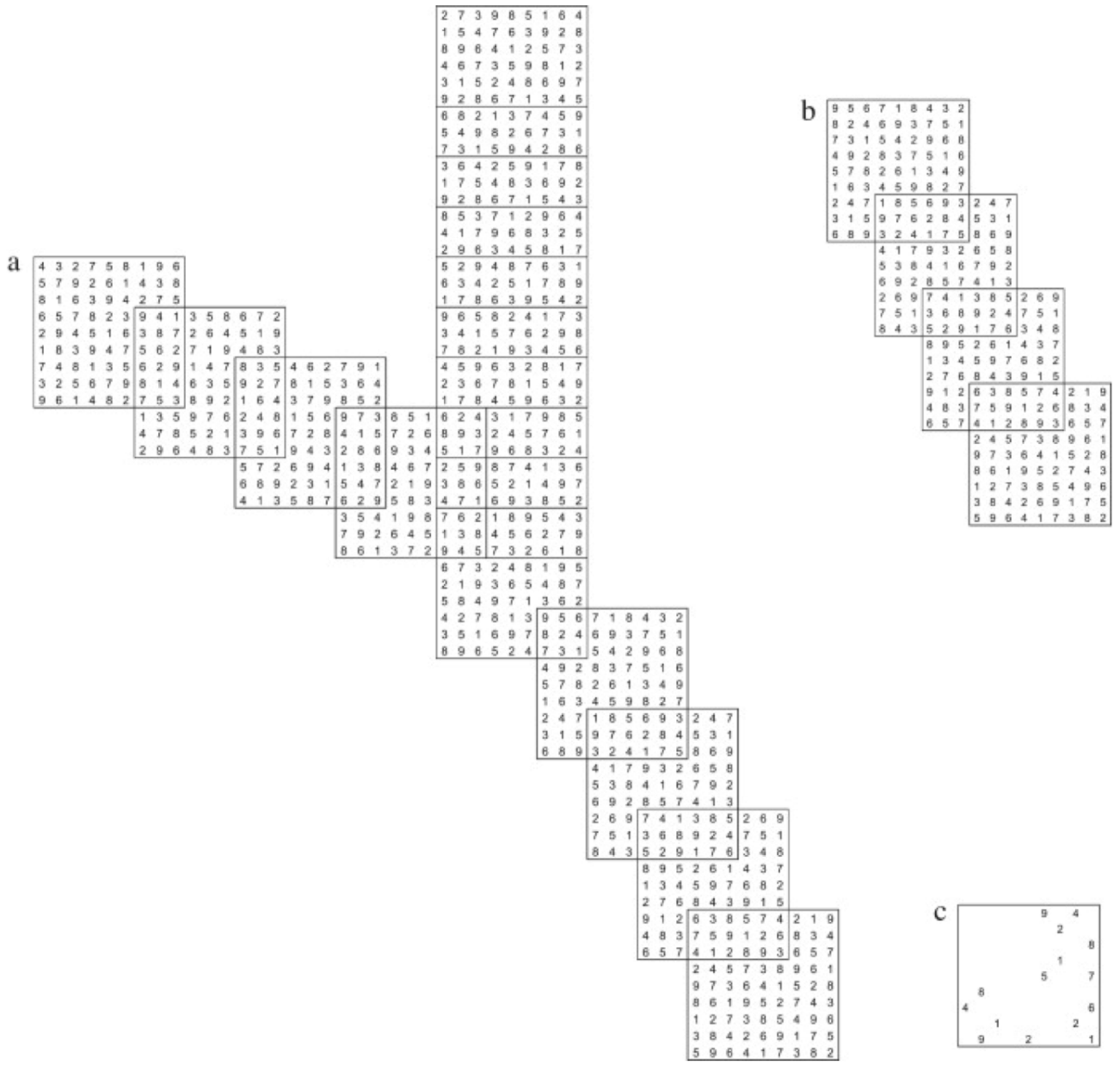
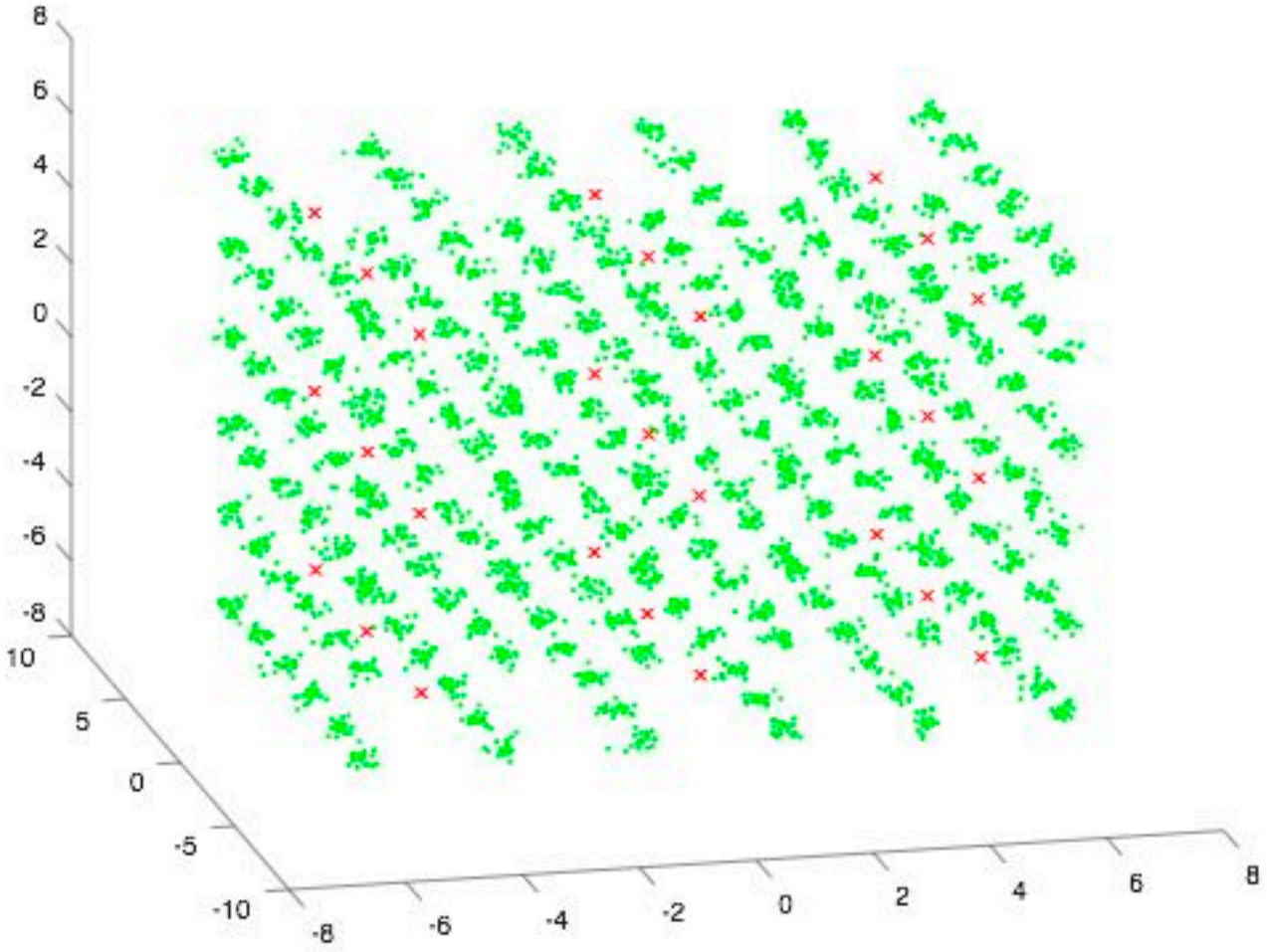
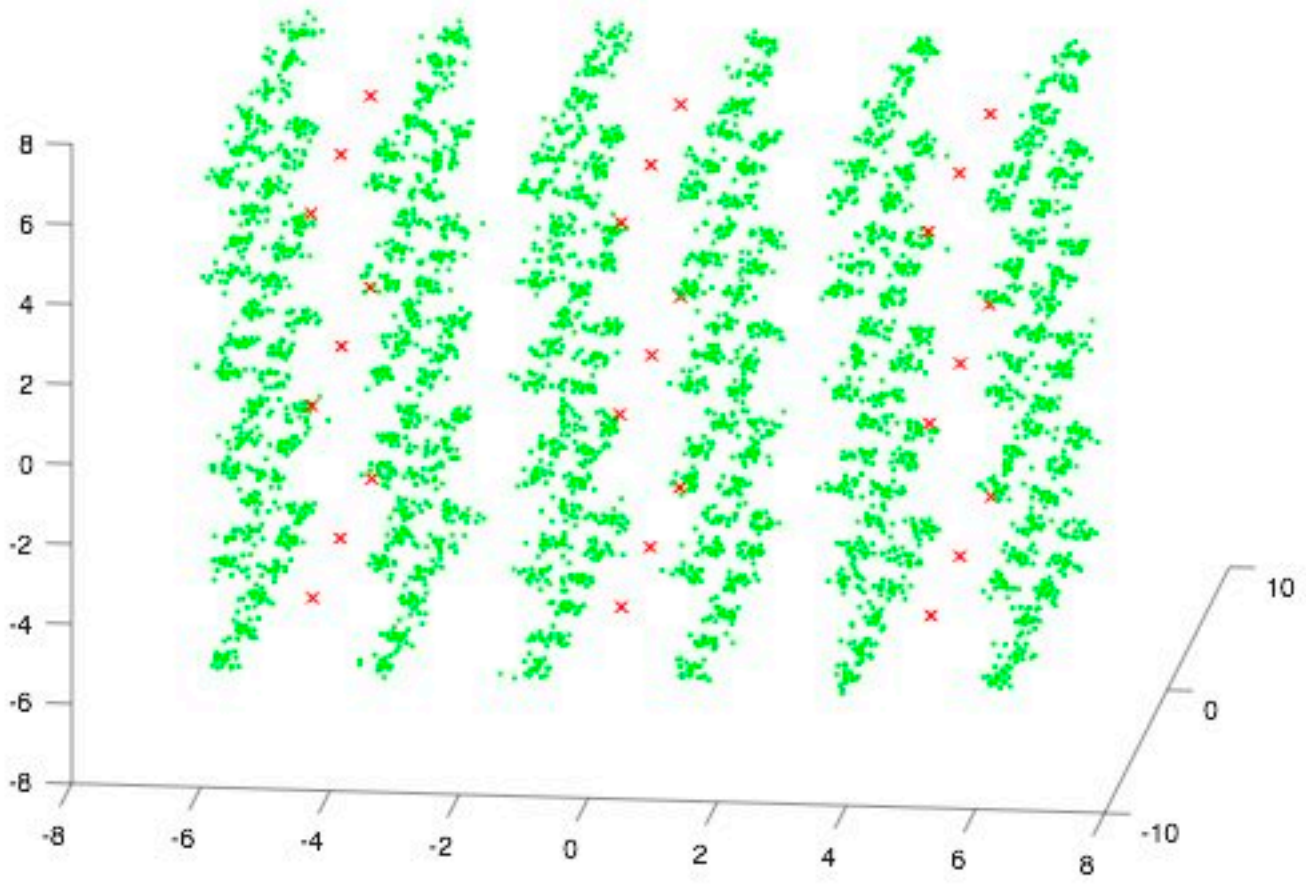
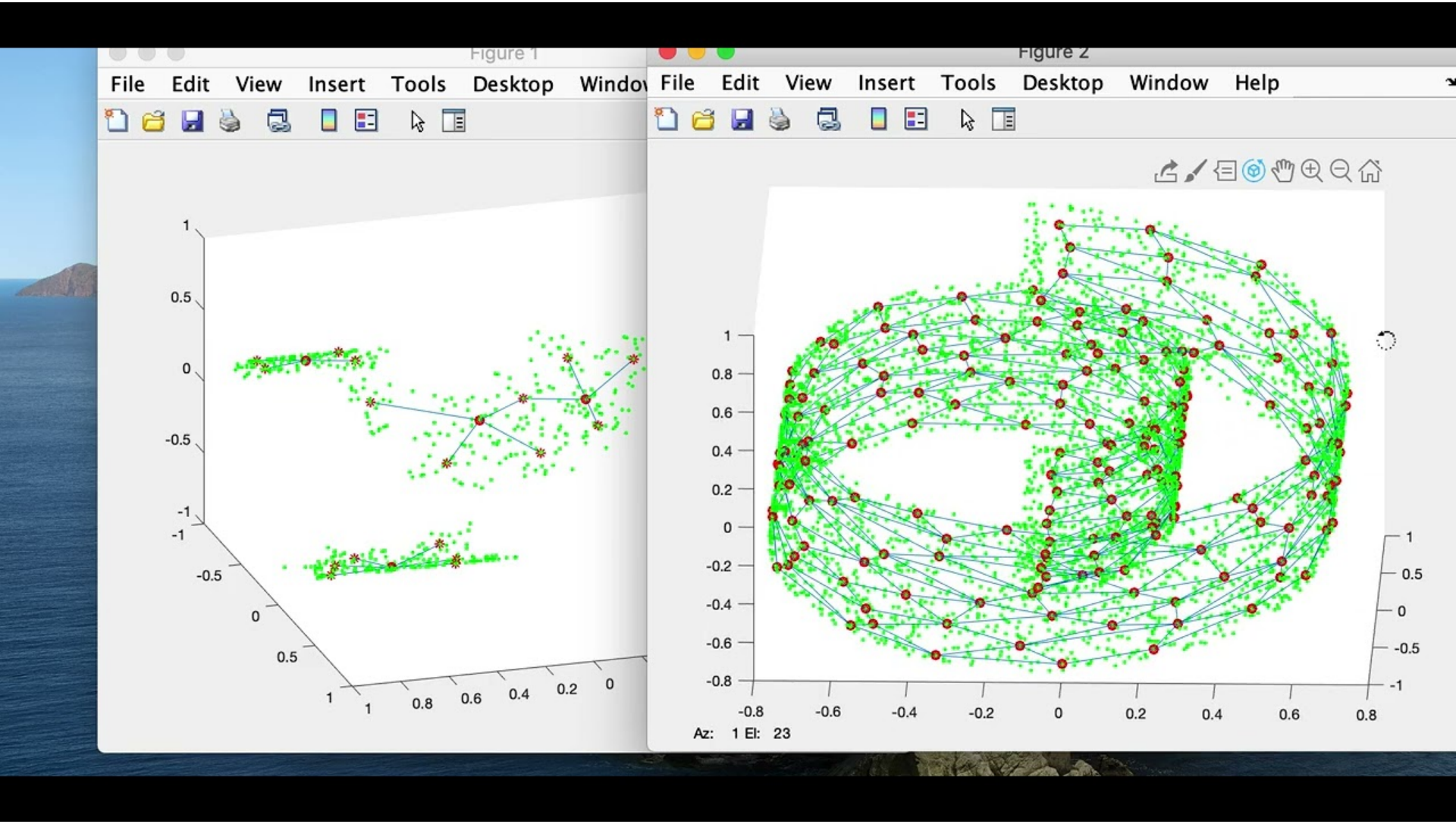
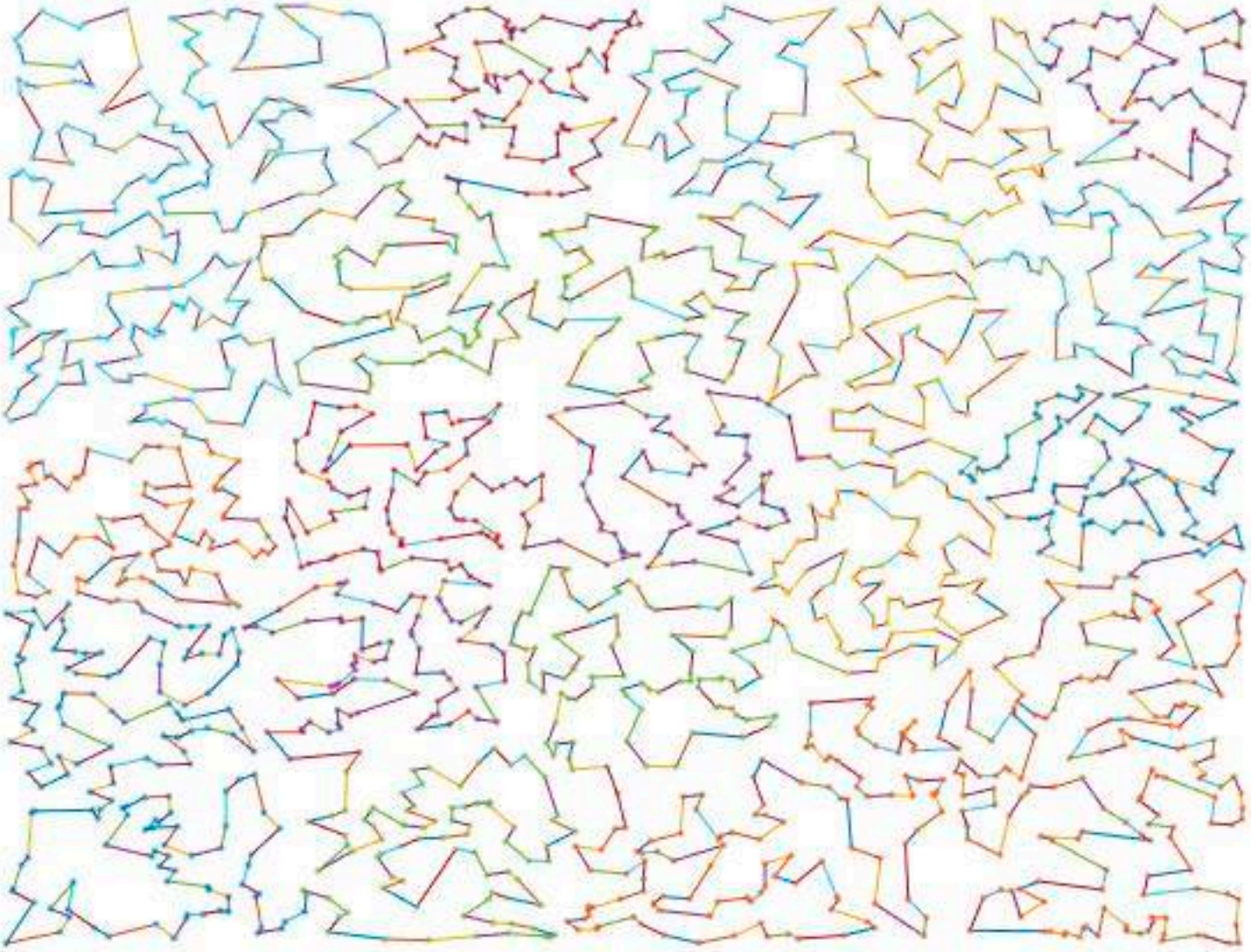


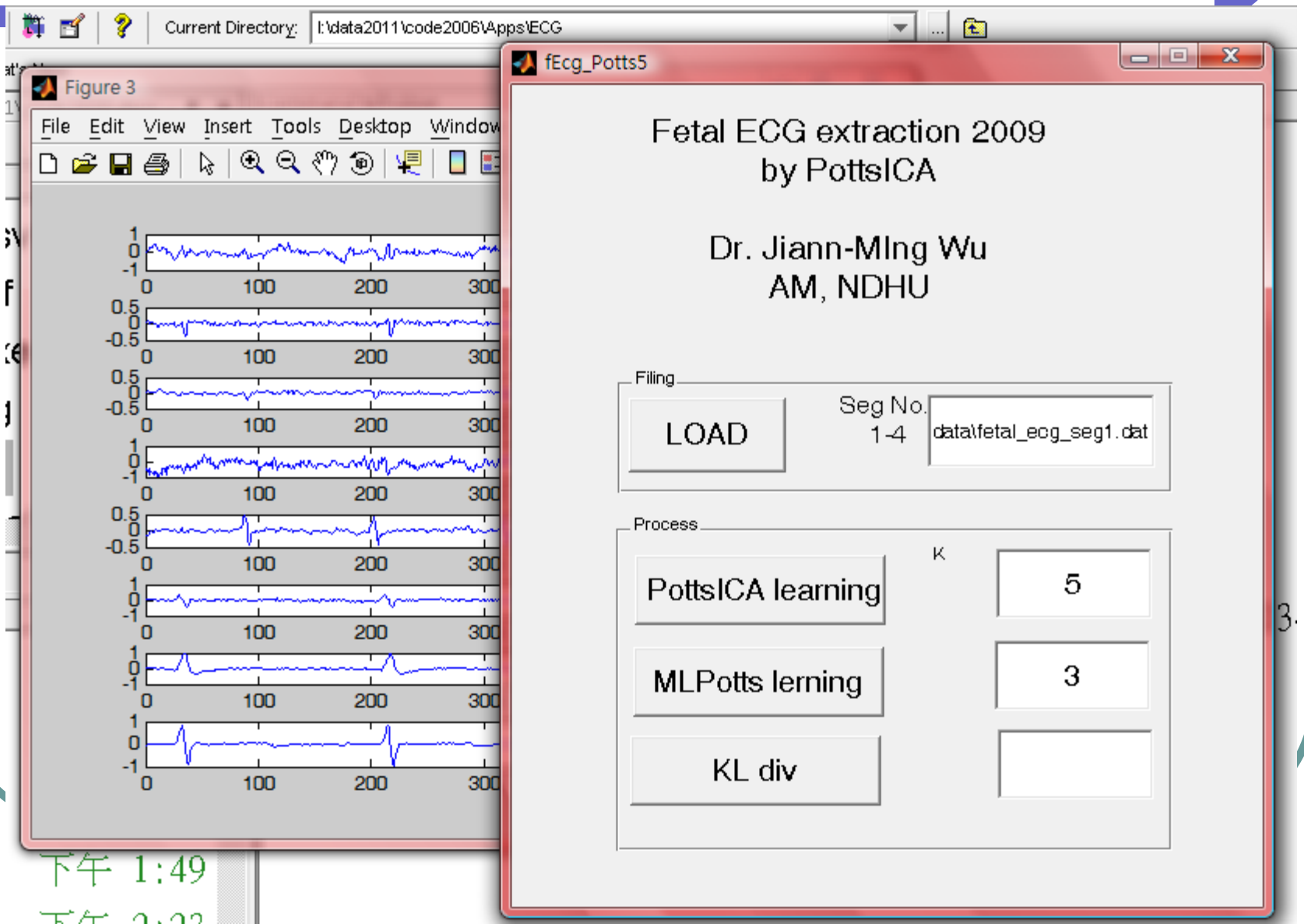

Fig. 15. Different partial clues for starfish shape compound Sudoku pattern restoration, (a) three tentacles, (b) one tentacle, and (c) a damaged seed pattern.


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


Figure 1
File Edit View Insert Tools Desktop Windo
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1
0.8
0.6
0.4
0.2
0
Figure 2
File Edit View Insert Tools Desktop Window Help
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
0.5
0
-0.5
-1
Az: 1 El: 23




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

# Mixed Facial images

## Wu et al 2008

sources

mixed images

AemICA

JadeICA



# ERP(event related potential)

J.-M. Wu et al. / Neural

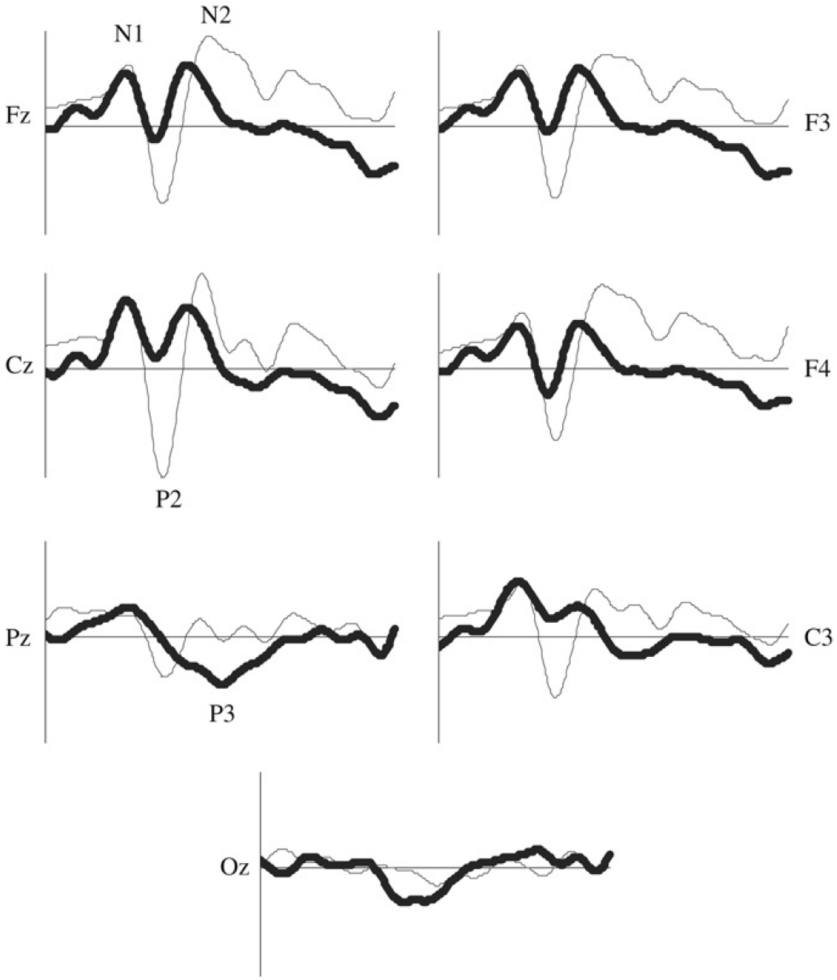



**Fig. 11.** Observed ERPs.

# ICs of ERP (Wu et al 2008)

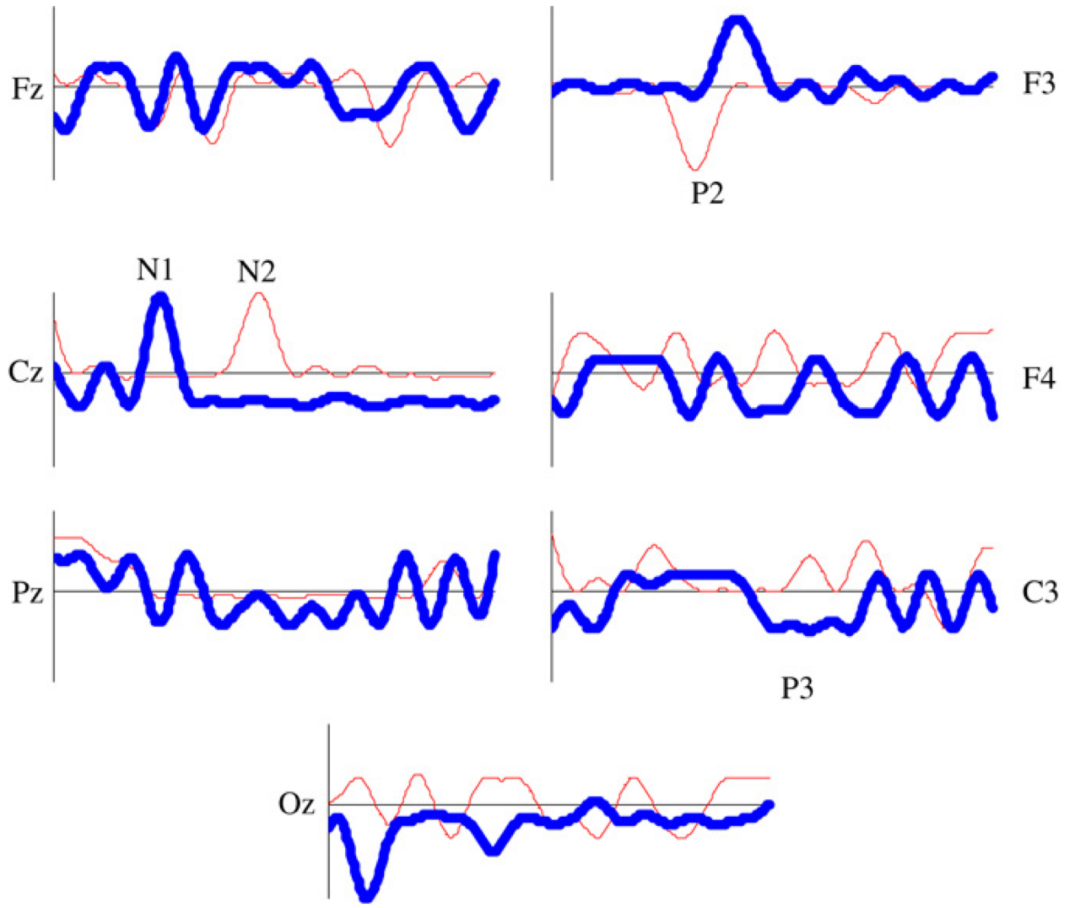



**Fig. 12.** Independent components obtained by AemICA for blind separation of ERPs.