Introduction to RBF Trainer 1.0

This software, RBF Trainer 1.0, is used for RBF network design, in order to solve function approximation problems and pattern classification problems. It contains the properties:

- Both first and second order gradient descent methods are implemented;
- Parameters, including input weights ***u***, output weights ***w***, centers ***c*** and widths ***σ***, can be adjusted, as shown in Fig. 1;
- Various options for initial settings, including randomly selection, fixed values, reading initial settings from file and the recently developed error correction algorithm;
- Both off-line and on-line (for the error correction algorithm) training procedures are implemented.
- Designed RBF networks can be tested, by both training data and new data.

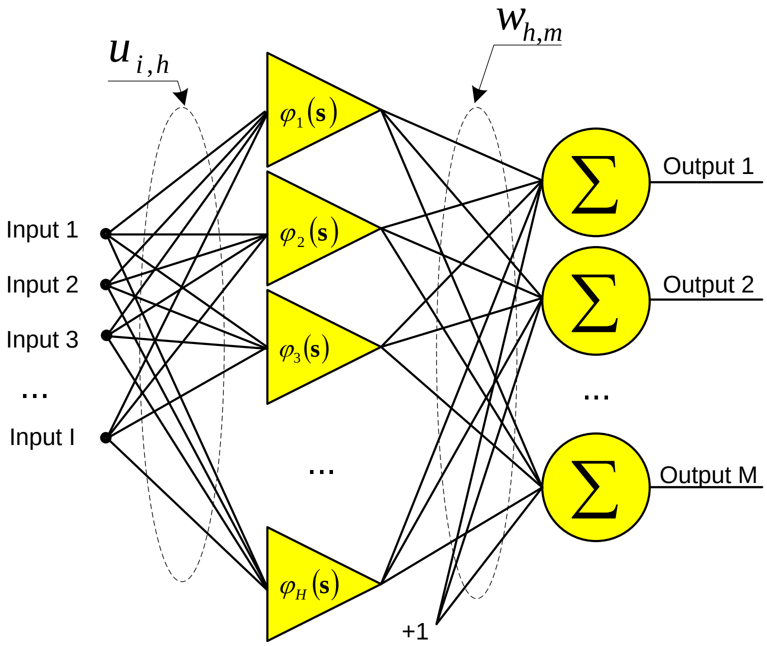


Fig. 1 General architecture of RBF network, with *I* inputs, *H* RBF units and *M* outputs; parameters include input weight matrix ***u*** (*H*×*I*), output weight vector ***w*** (*(H+1)*×*M*), center matrix ***c*** (*H*×*I*) and width vector ***σ*** (*1*×*H*)

For a given pattern *p*, the output of each RBF unit *h* is calculated by:

$$\varphi_h(\mathbf{x}_p) = \exp\left(-\frac{\left\|\mathbf{x}_p \bullet \mathbf{u}_h - \mathbf{c}_h\right\|^2}{\sigma_h}\right)$$

Where: $\mathbf{x}_p \bullet \mathbf{u}_h = \lfloor x_{p,1}u_{1,h} \quad x_{p,2}u_{2,h} \quad \cdots \quad x_{p,i}u_{i,h} \cdots \quad \rfloor$, and ||•|| represents the computation of Euclidean norm.

Then the output *m* of the network is calculated by

$$o_{p,m} = \sum_{h=1}^{H} w_{h,m}\varphi_h(\mathbf{x}_p) + w_{0,m}$$

1. *Files*

The software consists of 4 files: *RBF_Trainer.exe*, *PlotFor1D.m*, *PlotFor2D.m* and this file *Intro.pdf*.

RBF_Trainer.exe: execute file for RBF trainer;

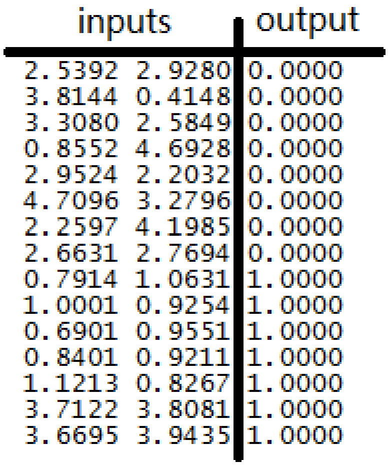
PlotFor1D.m: used for plotting the test results of designed RBF networks (2-D only);

PlotFor2D.m: used for plotting the test results of designed RBF networks (3-D only);

Intro.pdf: used for introducing how to use the RBF trainer.

In order to use the RBF trainer, there should be other files, including the *data files* (*.dat or *.txt), *initial files* (*.ini), *training result files* (.txt) and *testing output files* (.txt).

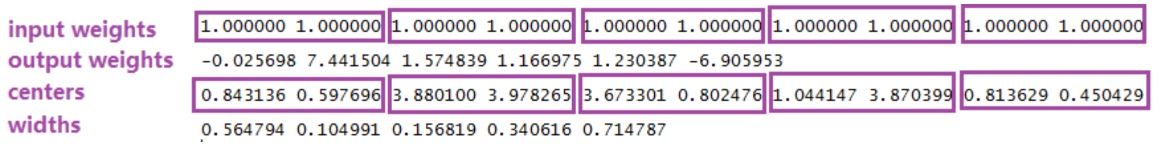
Data files: in "*.dat" or "*.txt" format. Each line consists of inputs and one output. The number of lines is equal to the number of patterns.



```
2.5392 2.9280 0.0000
3.8144 0.4148 0.0000
3.3080 2.5849 0.0000
0.8552 4.6928 0.0000
2.9524 2.2032 0.0000
4.7096 3.2796 0.0000
2.2597 4.1985 0.0000
2.6631 2.7694 0.0000
0.7914 1.0631 1.0000
1.0001 0.9254 1.0000
0.6901 0.9551 1.0000
0.8401 0.9211 1.0000
1.1213 0.8267 1.0000
3.7122 3.8081 1.0000
3.6695 3.9435 1.0000
```

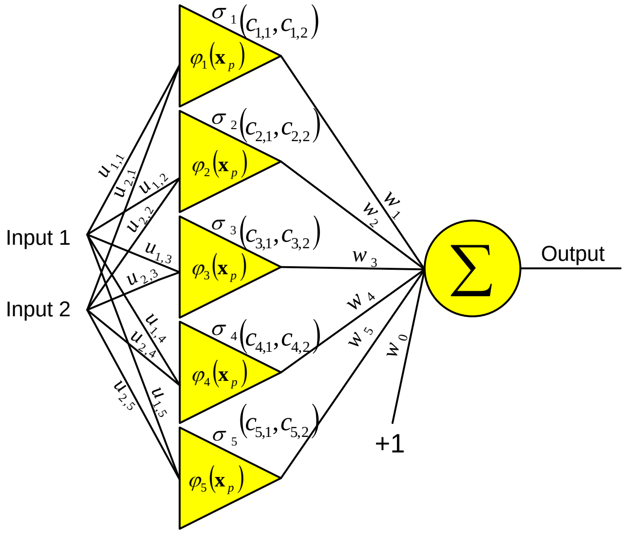
Fig. 2 Data file with two inputs and one output

Initial files: in "*.ini" format. The initial files consist of four lines of data: the first line is the input weights, the second line of the output weights, the third line is the centers and the fourth line is the widths. Fig. 3 shows a sample of initial file, consists of parameters of 5 RBF units, as shown in Fig. 4.



```
input weights    1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
output weights   -0.025698 7.441504 1.574839 1.166975 1.230387 -6.905953
centers          0.843136 0.597696 3.880100 3.978265 3.673301 0.802476 1.044147 3.870399 0.813629 0.450429
widths           0.564794 0.104991 0.156819 0.340616 0.714787
```

Fig. 3 Initial file, with 5 RBF units

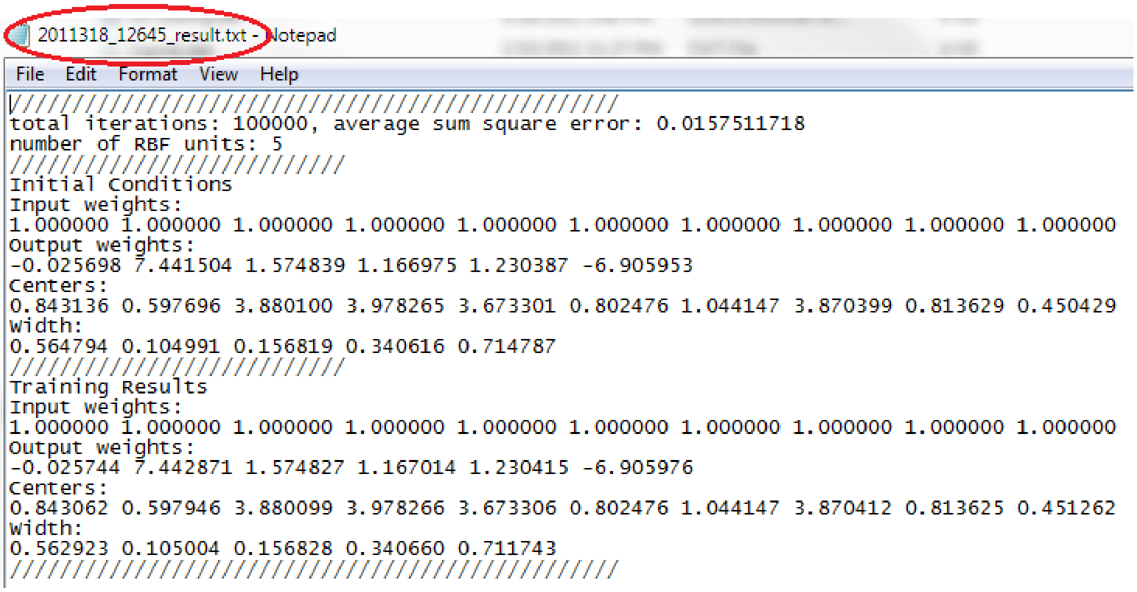


$\sigma_1(c_{1,1}, c_{1,2})$
$\varphi_1(\mathbf{x}_p)$
$\sigma_2(c_{2,1}, c_{2,2})$
$\varphi_2(\mathbf{x}_p)$
$\sigma_3(c_{3,1}, c_{3,2})$
$\varphi_3(\mathbf{x}_p)$
$\sigma_4(c_{4,1}, c_{4,2})$
$\varphi_4(\mathbf{x}_p)$
$\sigma_5(c_{5,1}, c_{5,2})$
$\varphi_5(\mathbf{x}_p)$
$u_{1,1}$
$u_{2,1}$
$u_{1,2}$
$u_{2,2}$
$u_{1,3}$
$u_{2,3}$
$u_{1,4}$
$u_{2,4}$
$u_{2,5}$
$u_{1,5}$
Input 1
Input 2
w_1
w_2
w_3
w_4
w_5
w_0
+1
Σ
Output

Fig. 4 Five RBF units with initialization in Fig. 3

The parameters in Fig. 3 is organized as
Input weights: $u_{1,1}$, $u_{2,1}$, $u_{1,2}$, $u_{2,2}$, $u_{1,3}$, $u_{2,3}$, $u_{1,4}$, $u_{2,4}$, $u_{1,5}$, $u_{2,5}$
Output weights: w_0, w_1, w_2, w_3, w_4, w_5
Centers: $c_{1,1}$, $c_{1,2}$, $c_{2,1}$, $c_{2,2}$, $c_{3,1}$, $c_{3,2}$, $c_{4,1}$, $c_{4,2}$, $c_{5,1}$, $c_{5,2}$
Widths: σ_1, σ_2, σ_3, σ_4, σ_5

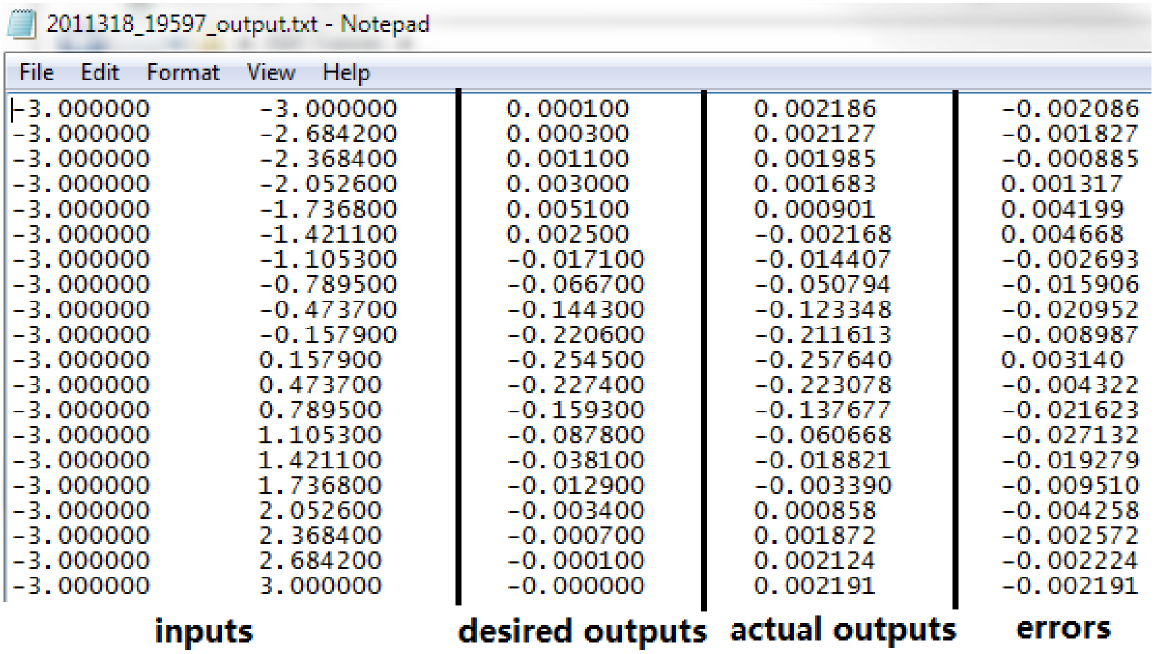
Training result files: in "{year}{month}{day}_{hour}{minute}{second}_ result.txt" format, as shown in Fig. 5. They contain the training information, including *the number of RBF units*, *training iterations*, *average training error*, *initial parameter settings* and *final training results*.


2011318_12645_result.txt - Notepad
File Edit Format View Help

```
//////////////////////////////////////////////////
total iterations: 100000, average sum square error: 0.0157511718
number of RBF units: 5
///////////////////////////
Initial Conditions
Input weights:
1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
Output weights:
-0.025698 7.441504 1.574839 1.166975 1.230387 -6.905953
Centers:
0.843136 0.597696 3.880100 3.978265 3.673301 0.802476 1.044147 3.870399 0.813629 0.450429
Width:
0.564794 0.104991 0.156819 0.340616 0.714787
///////////////////////////
Training Results
Input weights:
1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
Output weights:
-0.025744 7.442871 1.574827 1.167014 1.230415 -6.905976
Centers:
0.843062 0.597946 3.880099 3.978266 3.673306 0.802476 1.044147 3.870412 0.813625 0.451262
Width:
0.562923 0.105004 0.156828 0.340660 0.711743
//////////////////////////////////////////////////
```

Fig. 5 Training result file

Testing output files: in "{year}{month}{day}_{hour}{minute}{second}_ output.txt" format. Each row consists of inputs, desired output, actual output and error, as shown in Fig. 6.

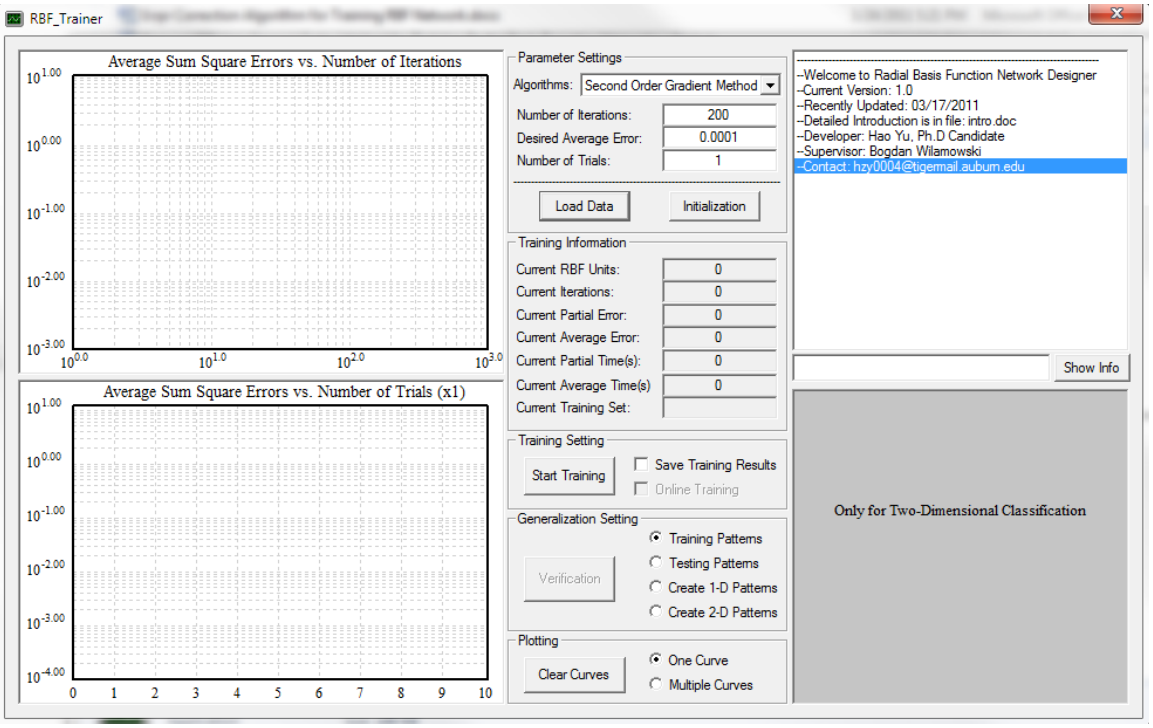

2011318_19597_output.txt - Notepad
File Edit Format View Help
inputs
desired outputs
actual outputs
errors

```
-3.000000    -3.000000     0.000100     0.002186    -0.002086
-3.000000    -2.684200     0.000300     0.002127    -0.001827
-3.000000    -2.368400     0.001100     0.001985    -0.000885
-3.000000    -2.052600     0.003000     0.001683     0.001317
-3.000000    -1.736800     0.005100     0.000901     0.004199
-3.000000    -1.421100     0.002500    -0.002168     0.004668
-3.000000    -1.105300    -0.017100    -0.014407    -0.002693
-3.000000    -0.789500    -0.066700    -0.050794    -0.015906
-3.000000    -0.473700    -0.144300    -0.123348    -0.020952
-3.000000    -0.157900    -0.220600    -0.211613    -0.008987
-3.000000     0.157900    -0.254500    -0.257640     0.003140
-3.000000     0.473700    -0.227400    -0.223078    -0.004322
-3.000000     0.789500    -0.159300    -0.137677    -0.021623
-3.000000     1.105300    -0.087800    -0.060668    -0.027132
-3.000000     1.421100    -0.038100    -0.018821    -0.019279
-3.000000     1.736800    -0.012900    -0.003390    -0.009510
-3.000000     2.052600    -0.003400     0.000858    -0.004258
-3.000000     2.368400    -0.000700     0.001872    -0.002572
-3.000000     2.684200    -0.000100     0.002124    -0.002224
-3.000000     3.000000    -0.000000     0.002191    -0.002191
```

Fig. 6 Testing output file for a two-dimension case

2. *Graphical User Interface*

Fig. 2 shows the graphical user interface of the RBF trainer:



Algorithms: Second Order Gradient Method

Number of Iterations: 200

Desired Average Error: 0.0001

Number of Trials: 1

Current RBF Units: 0

Current Iterations: 0

Current Partial Error: 0

Current Average Error: 0

Current Partial Time(s): 0

Current Average Time(s): 0

Current Training Set:

□ Save Training Results

□ Online Training

● Training Patterns

○ Testing Patterns

○ Create 1-D Patterns

○ Create 2-D Patterns

● One Curve

○ Multiple Curves

--Welcome to Radial Basis Function Network Designer
--Current Version: 1.0
--Recently Updated: 03/17/2011
--Detailed Introduction is in file: intro.doc
--Developer: Hao Yu, Ph.D Candidate
--Supervisor: Bogdan Wilamowski
--Contact: hzy0004@tigermail.auburn.edu

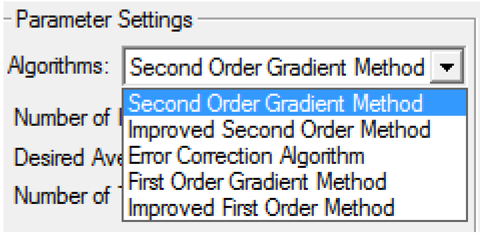
Fig. 7 Graphical user interface of the RBF Trainer

Let us explain the function of each part.

Training Algorithm Selection

As shown in Fig. 8, there are five algorithms: “second order gradient method”, “improved second order method”, “error correction algorithm”, “first order gradient algorithm” and “improved first order algorithm”.

- Second Order Gradient Method: using second order algorithm to adjust parameters, including output weights, centers and widths.
- Improved Second Order Method: using second order algorithm to adjust parameters, including input weights, output weights, centers and widths.
- Error Correction Algorithm: using second order algorithm to adjust parameters, including output weights, centers and widths. In this algorithm, the RBF units are added one by one, initialed with optimally selected centers.
- First Order Gradient Method: using first order gradient descent method to adjust parameters, including output weights, centers and widths. Momentum is used to improve the performance.
- Improved First Order Method: using first order gradient descent method to adjust parameters, including input weights, output weights, centers and widths. Momentum is applied to improve to performance.



Parameter Settings

Algorithms: Second Order Gradient Method

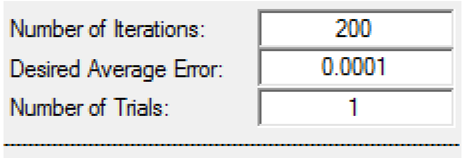
Second Order Gradient Method
Improved Second Order Method
Error Correction Algorithm
First Order Gradient Method
Improved First Order Method
Number of I
Desired Ave
Number of

Fig. 8 Algorithm selection

Parameter Setting:

There are three general parameters: number of iterations, desired average error and number of trials.

- Number of Iterations: the training process will keep running until (a) reach the desired average error, or (b) reach the setting number of iterations, or (c) manually stopped.
- Desired Average Error: smallest average sum square error, as the destination of training.
- Number of Trials: number of repeating times for training, with different initial conditions.



Number of Iterations: 200

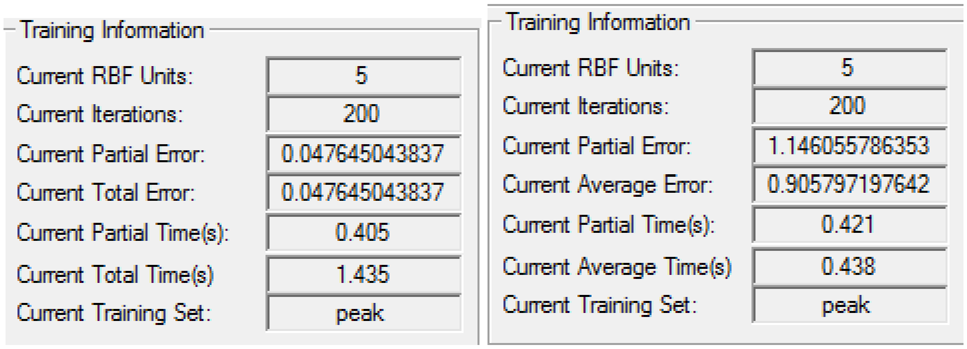
Desired Average Error: 0.0001

Number of Trials: 1

Fig. 9 General Parameters

Training Parameter Monitor:

This window is used to monitor the training process:



Training Information

Current RBF Units: 5

Current Iterations: 200

Current Partial Error: 0.047645043837

Current Total Error: 0.047645043837

Current Partial Time(s): 0.405

Current Total Time(s) 1.435

Current Training Set: peak

(a)

Training Information

Current RBF Units: 5

Current Iterations: 200

Current Partial Error: 1.146055786353

Current Average Error: 0.905797197642

Current Partial Time(s): 0.421

Current Average Time(s) 0.438

Current Training Set: peak

(b)

Fig. 10 Parameters monitor: (a) error correction algorithm; (b) other algorithms

Plotting area

Fig. 11 shows the plotting of average SSEs as the increasing of number of iterations. The scales of x and y axis will be adjusted automatically, during the training process. Also, by click the right button of mouse, the popup menu will be helpful for manually setting the scales of x and y axis.

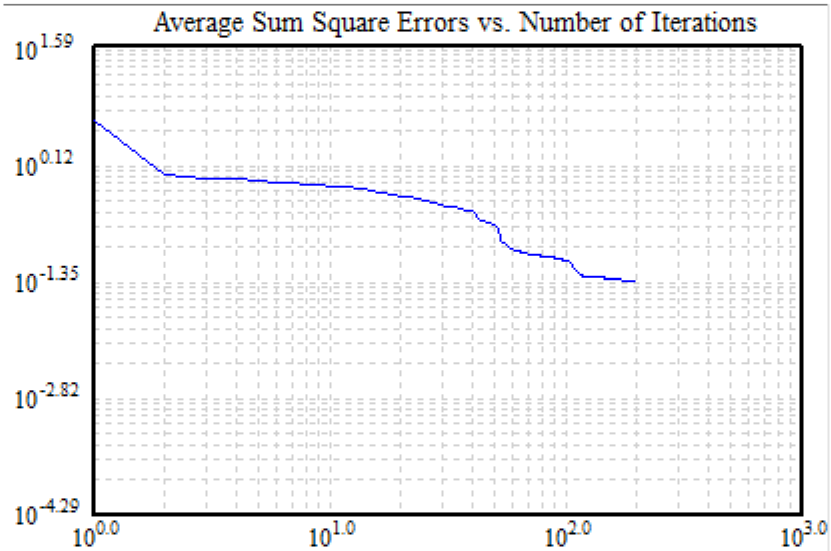


Fig. 11 Plotting of Average Sum Square Errors vs. Number of Iterations

Also the final average SSE of each training process is plotted for different trials in Fig. 12. Different results will be obtained because of the different initial conditions.

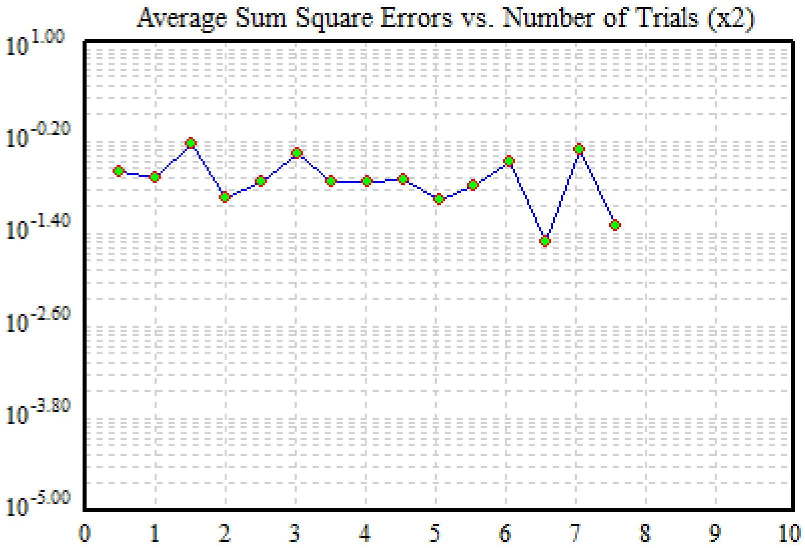


Fig. 12 Plotting of Average Sum Square Errors vs. Number of Trials

For error correction algorithm, the coordinates in Fig. 12 will change as shown in Fig. 13, which presents the average SSE changes when the number of RBF units increases.

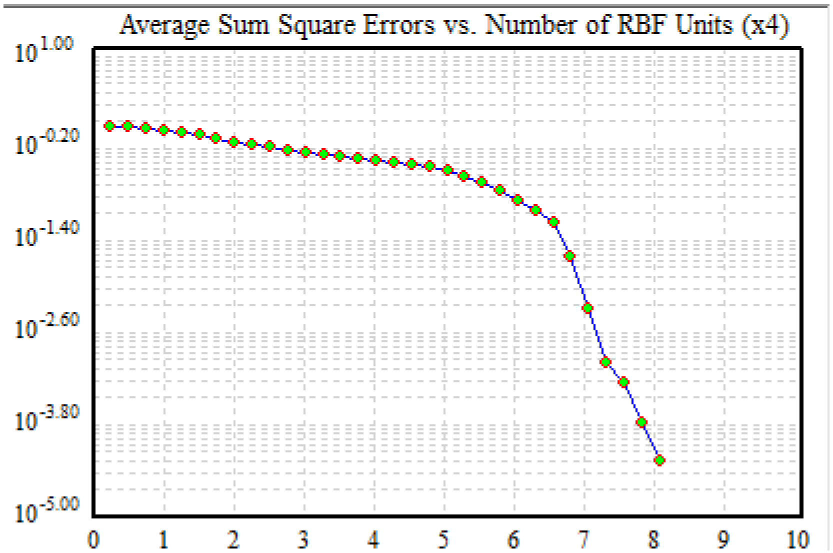


Fig. 13 Plotting of average sum square errors as the increase of number of RBF units

The plotting in Fig. 14 is only for two-dimension classification problem.

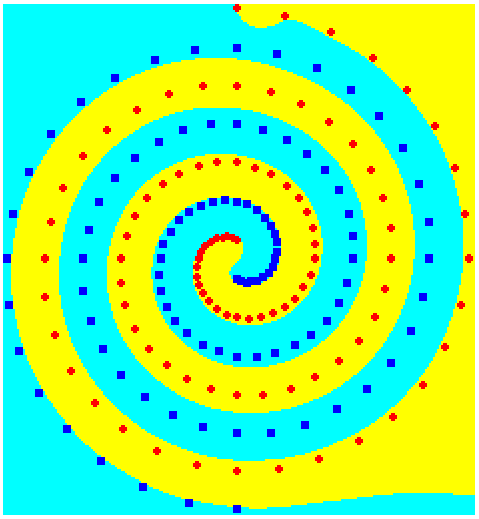
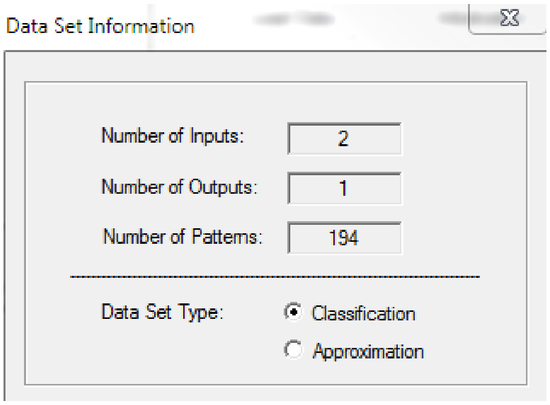


Fig. 14 Plotting of average sum square errors as the increase of number of RBF units

Training Control

- Load Data: used for training data load. After selecting the data file, the data information will be shown as in Fig. 15. Notice that data set type should be chosen properly for correct plotting.



Data Set Information

Number of Inputs: 2

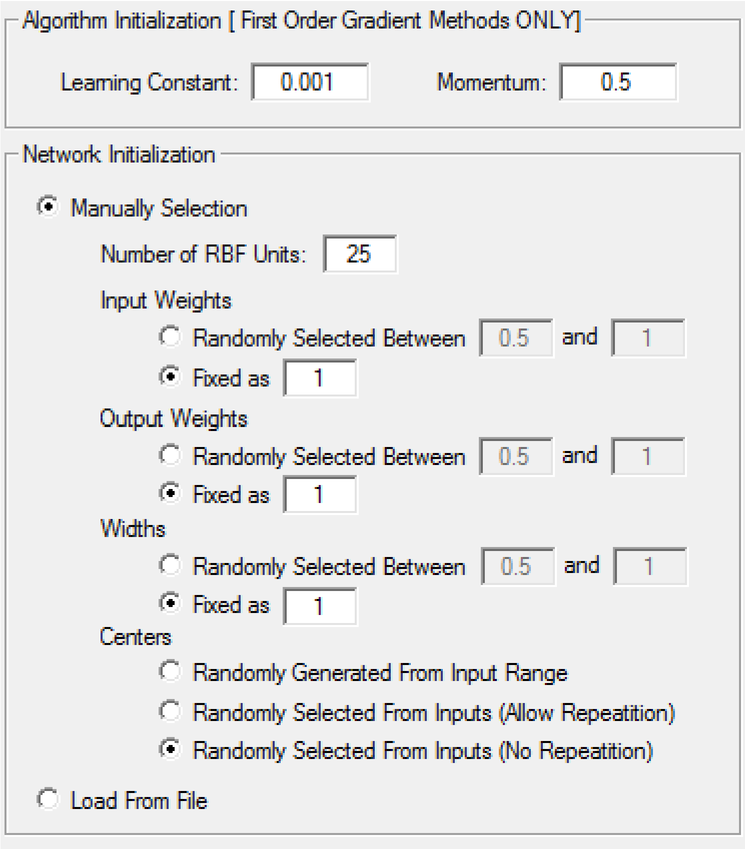
Number of Outputs: 1

Number of Patterns: 194

Data Set Type: ● Classification ○ Approximation

Fig. 15 Training data information

- Initialization: used for training parameter initialization. As shown in Fig. 16 Learning constant and Momentum are used for first order algorithms. Network parameters, including input weights, output weights, widths and centers can be chosen manually or read from file. For error correction algorithm, the initialization is different as shown in Fig. 17 and only one parameter is required.



Algorithm Initialization [First Order Gradient Methods ONLY]

Learning Constant: 0.001 Momentum: 0.5

Network Initialization

● Manually Selection

Number of RBF Units: 25

Input Weights

○ Randomly Selected Between 0.5 and 1

● Fixed as 1

Output Weights

○ Randomly Selected Between 0.5 and 1

● Fixed as 1

Widths

○ Randomly Selected Between 0.5 and 1

● Fixed as 1

Centers

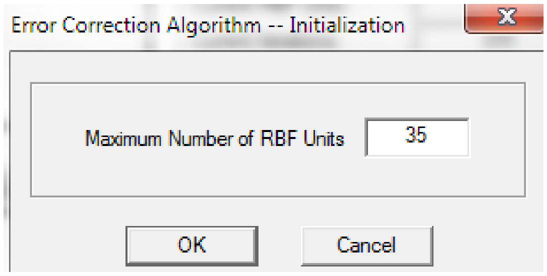
○ Randomly Generated From Input Range

○ Randomly Selected From Inputs (Allow Repeatition)

● Randomly Selected From Inputs (No Repeatition)

○ Load From File

Fig. 16 Initial condition settings for all algorithms except error correction algorithm



Error Correction Algorithm -- Initialization

Maximum Number of RBF Units: 35

Fig. 17 Initial condition settings for error correction algorithm

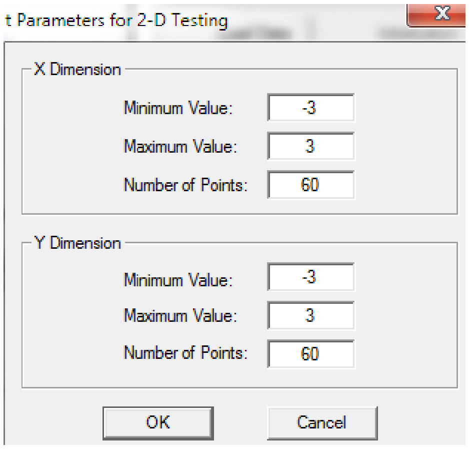
- Start Training: this button is used to start the training process after "Load Data" and "Initialization" steps. When the training process is running, the "Start Training" becomes "Stop Training". The training process will be stopped if it is clicked.

Plotting Control

- Clear Curves: all plotting will be removed if this button is clicked.
    - One Curve: only one curve will be shown for each case or trial
    - Multiple Curves: all curves will be shown at the same coordinate

Testing Control

- Training Patterns: the trained RBF networks will be tested by training data set
- Testing Patterns: the trained RBF networks will be tested by testing data set which should have the same format as the training data set
- Create 1-D Patterns: used for one-dimensional input testing. New data will be generated and applied for testing automatically, once the range and number of the points are selected.
- Create 2-D Patterns: used for two-dimensional inputs testing. New data will be generated and applied for testing automatically, once the range and number of the points for each dimension are selected.



t Parameters for 2-D Testing

X Dimension

Minimum Value: -3

Maximum Value: 3

Number of Points: 60

Y Dimension

Minimum Value: -3

Maximum Value: 3

Number of Points: 60

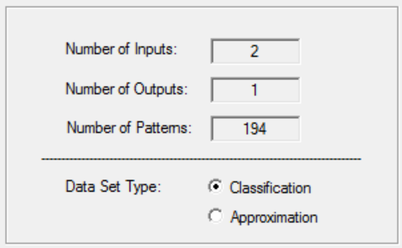
Fig. 18 Settings of range and number of the data points for 2-D inputs

3. *Examples*

In this section, let us apply three examples to show how to use the RBF Trainer.

Two-spiral Problem

a. Click “Load Data” button, load data file “2spiral.dat” in “Examples”→“Two spiral” folder;



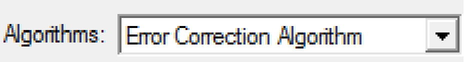
Number of Inputs: 2

Number of Outputs: 1

Number of Patterns: 194

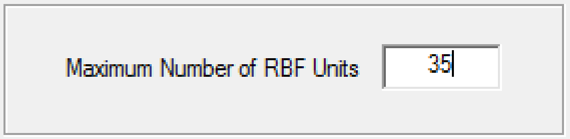
Data Set Type: ● Classification ○ Approximation

b. Select “Error Correction Algorithm”



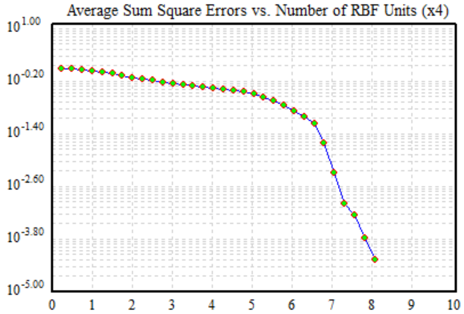
Algorithms: Error Correction Algorithm

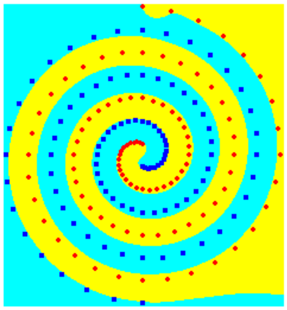
c. Click “Initialization”, set 35 as the maximum number of RBF units

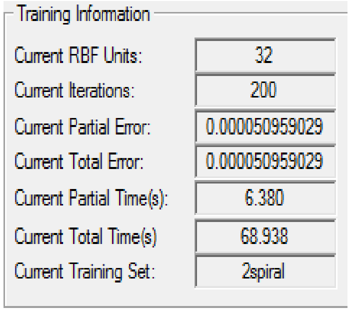


Maximum Number of RBF Units 35

d. Click “Start Training”. The training process will be ended when average training SSE less than 0.0001. The results is shown in figures below.







Training Information

Current RBF Units: 32

Current Iterations: 200

Current Partial Error: 0.000050959029

Current Total Error: 0.000050959029

Current Partial Time(s): 6.380

Current Total Time(s) 68.938

Current Training Set: 2spiral

Peak Surface Approximation

a. Click “Load Data”, load data file “peak.dat” in “Examples”→ “Peak” folder.

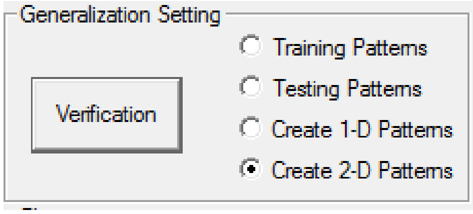
b. Select “Error Correction Algorithm”

c. Click “Initialization”, set 1 as the maximum number of RBF units

d. Click “Start Training”

e. Wait until training stops when the number of iterations reaches 200.

f. In generalization area, choose “Create 2-D Patterns”



Generalization Setting

Verification

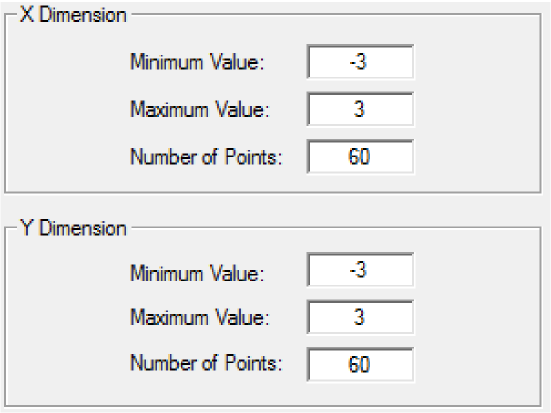
○ Training Patterns

○ Testing Patterns

○ Create 1-D Patterns

● Create 2-D Patterns

g. Click “Verification” and fill in the editing boxes as shown in figure below



X Dimension

Minimum Value: -3

Maximum Value: 3

Number of Points: 60

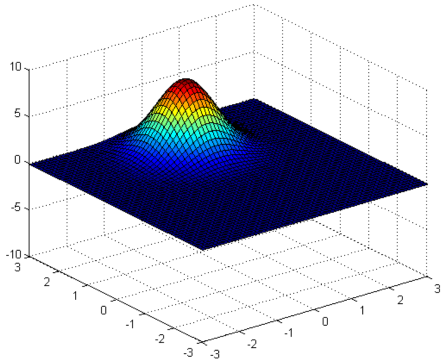
Y Dimension

Minimum Value: -3

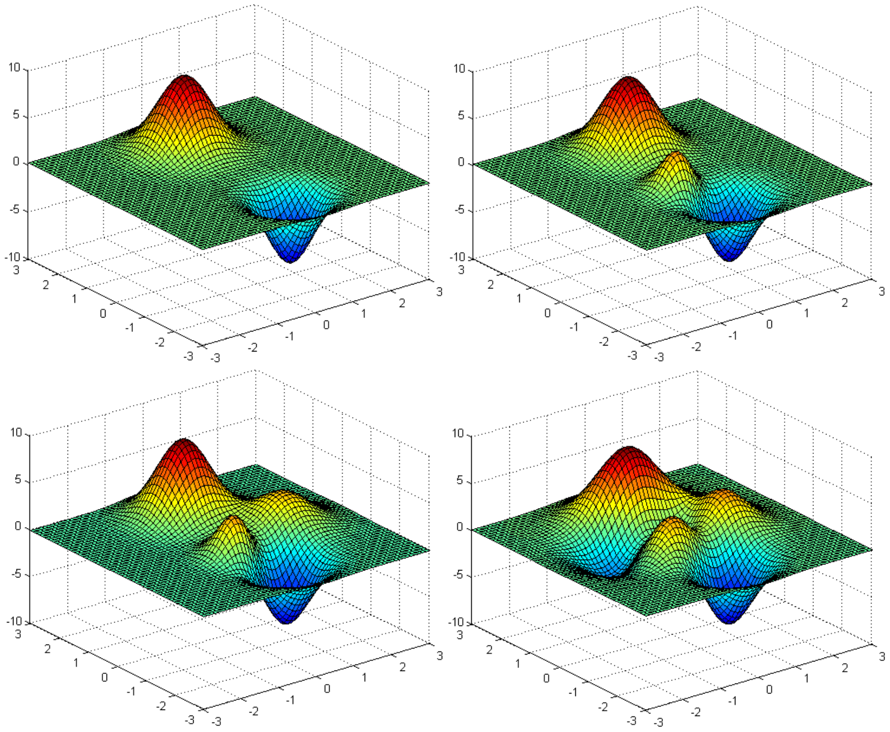
Maximum Value: 3

Number of Points: 60

h. Open MALAB, load file “PlotFor2D.m” and execute it.



i. Repeat steps c to h, for maximum number of RBF units equal to 2, 3, 4 and 5, respectively. The results shown below.



Online Training

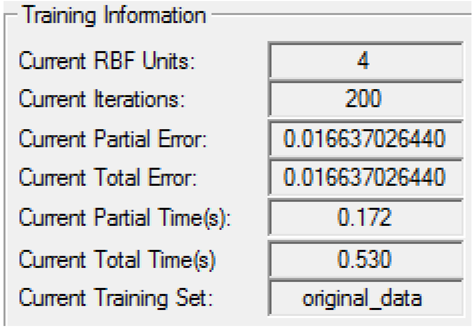
a. Click “Load Data”, load data file “original_data.txt” in “Examples”→ “Online Training” folder.

b. Choose “Error Correction Algorithm”

c. Click “Initialization”, set 4 as the maximum number RBF units

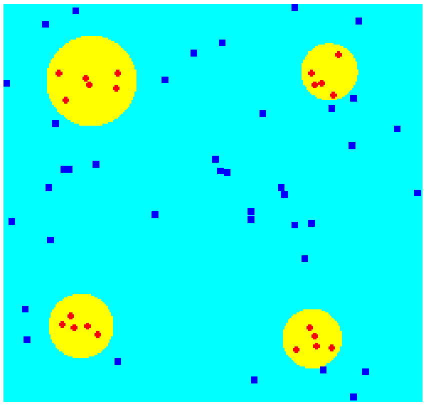
d. Click “Start Training”

e. After the training process the classification result is shown

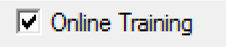


Training Information

Current RBF Units:	4
Current Iterations:	200
Current Partial Error:	0.016637026440
Current Total Error:	0.016637026440
Current Partial Time(s):	0.172
Current Total Time(s)	0.530
Current Training Set:	original_data

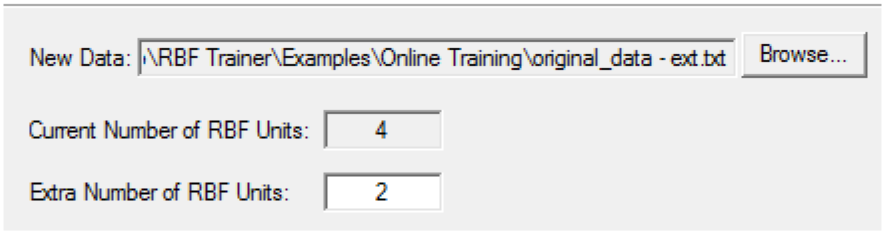


f. Click “Online Training”



☑ Online Training

g. Click “Browse…” to load data “original_data - ext.txt”; set the extra number of RBF units as 2

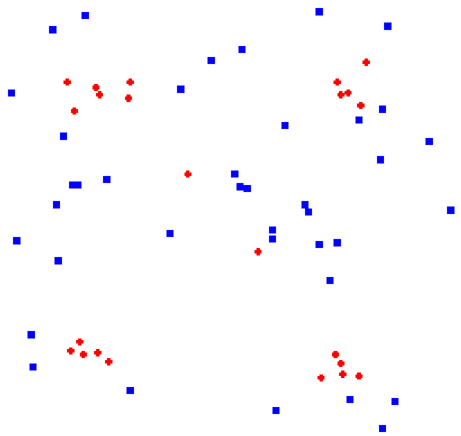


New Data: \RBF Trainer\Examples\Online Training\original_data - ext.txt

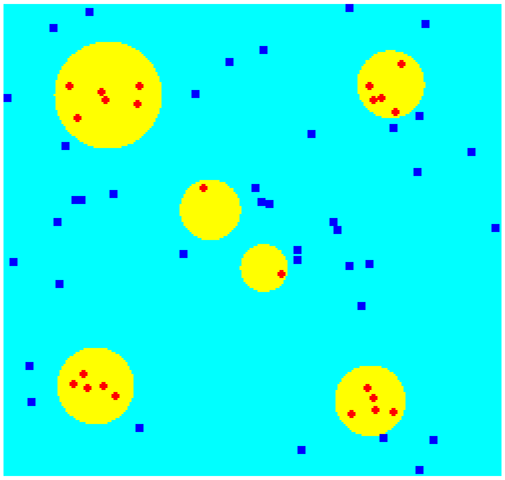
Current Number of RBF Units: 4

Extra Number of RBF Units: 2

h. Click “OK”, then the training data set changes as shown in figure below. Two more red points are added.



i. Click “Start Training”. After training, the results shown as



The RBF Trainer will keep being updated.

Notice: In this version, the RBF Trainer is only used for problems with one output case.