Self-Organizing Map and MLP Neural Network - A Practical Use

Cao Thang, 2011

This material guides you to use Self-Organizing Map (SOM) and MLP Neural Networks (NN) in some practical applications. It was used to introduce NN to some Japanese students. The author thought that it might be useful for the other students so he re-prepared it, for your reference. Because of time limitation, it may have mistakes. Please tell the author at spiceneuroAT gmail DOT com or http://spiceneuro.wordpress.com Thank you.
If you are interested in NN and SOM, you can use free software, SpiceSOM and SpiceMLP, download here
Some data presented here also already in "Data" folder when you setup SpiceSOM and SpiceMLP software. All results here are modeled by SpiceSOM and SpiceMLP.
Thank you.

CONTENTS

1. Iris Flower Data Set

Iris dataset consists of 150 samples of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor), 50 samples for each species. Features were measured from each sample with the length and the width of sepal and petal in centimeters.
Details are in http://archive.ics.uci.edu/ml/datasets/Iris

Iris setosa

Iris versicolor

Iris virginica

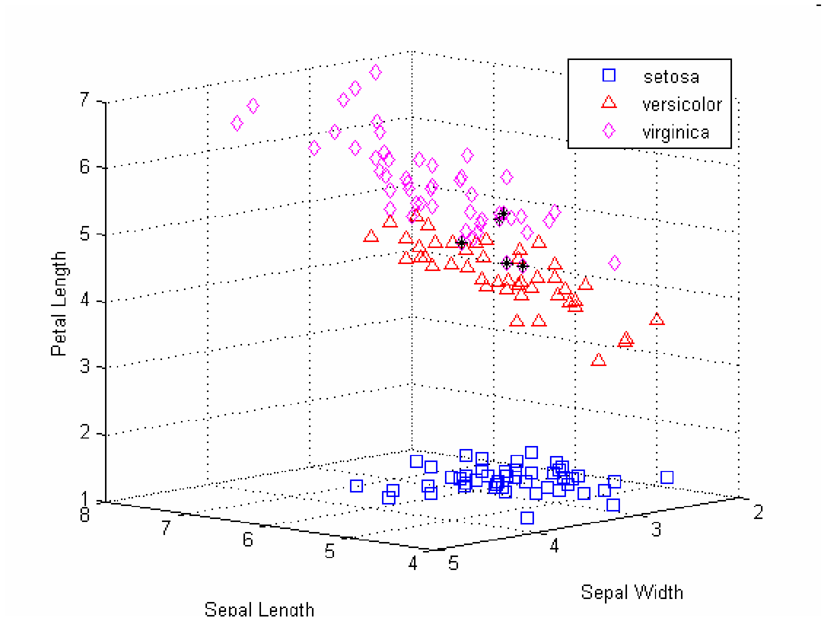


Fig.1. Iris flower photos (wikipedia)

1.1. Data Preparation

Classify Iris dataset by a NN with 4 inputs and 1 outputs. We code output of data as shown in table 1:

Table 1. Input and Output of a MLP NN

Data	Output
Iris setosa	0.0
Iris versicolor	0.5
Iris virginica	1.0

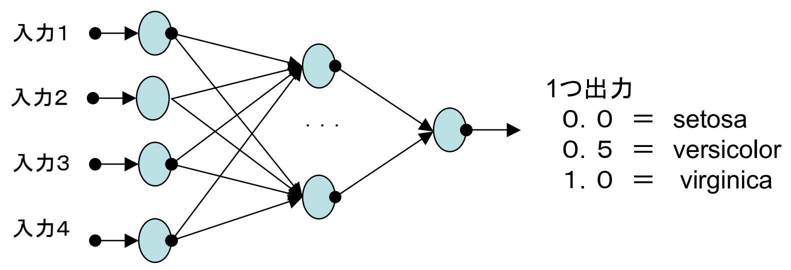
ID	Sepal Length	Sepal Width	Petal Length	Petal Width	Output	Species
1	5.1	3.5	1.4	0.2	0.0	setosa
...	...	...	...	...	...	...
51	7	3.2	4.7	1.4	0.5	versicolor
...	...	...	...	...	...	...
150	5.9	3	5.1	1.8	1.0	virginica

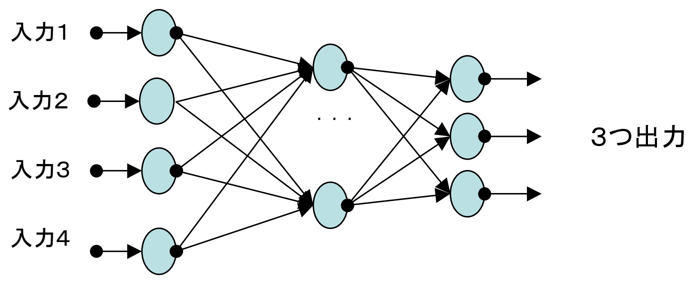
Classify Iris dataset by a NN with 4 inputs and 3 outputs. We code output of data as in the table 2:

Table 2. Input and Output of a MLP NN

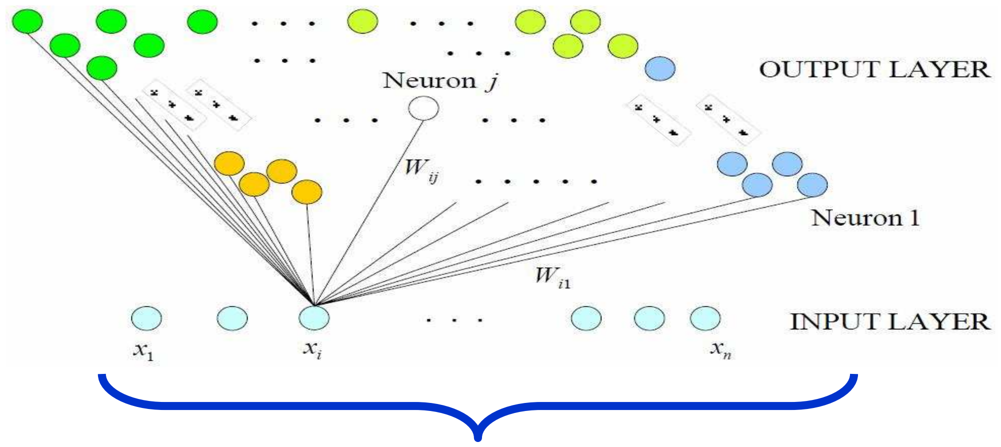
Data	Output 1	Output 2	Output 3
Iris setosa	1.0	0.0	0.0
Iris versicolor	0.0	1.0	0.0
Iris virginica	0.0	0.0	1.0

ID	Sepal Length	Sepal Width	Petal Length	Petal Width	Output 1	Output 2	Output 3	Species
1	5.1	3.5	1.4	0.2	1.0	0.0	0.0	setosa
…	…	…	…	…	…	…	…	…
51	7	3.2	4.7	1.4	0.0	1.0	0.0	versicolor
…	…	…	…	…	…	…	…	…
150	5.9	3	5.1	1.8	0.0	0.0	1.0	virginica





Classify Iris dataset by a Self-Organizing Map (SOM). Since data for SOM does not require outputs as data for MLP NN, so we prepare data as in the Fig.2.



SOM's inputs

ID	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
1	5.1	3.5	1.4	0.2	setosa
…	…	…	…	…	…
51	7	3.2	4.7	1.4	versicolor
…	…	…	…	…	…
150	5.9	3	5.1	1.8	virginica

Fig.2. Data for SOM

1.2. Classify Data with SpiceSOM

Classify Iris data with SOM size 8x10 neurons, we have an output map and corresponding output table as shown on Fig.3. On the output table, we easily notice that labels of versicolor and virginica are fall on both two neuron (0, 1) and neuron (0, 3). The other single neurons are matched with only one single label. That means the SOM misclassified versicolor and virginica data at neuron (0, 1) and neuron (0, 3), and correctly classify at the other neuron. On the output map we also see data with setosa labels are separated far on a side, data with versicolor and virginica labels located closer each other. Intuitionally, we can see setosa species has size and shape that are different from versicolor and virginica. And versicolor and virginica species have slightly similar size and shape and sometimes we can not distinguish these two species if only by their sepal and petal sizes.

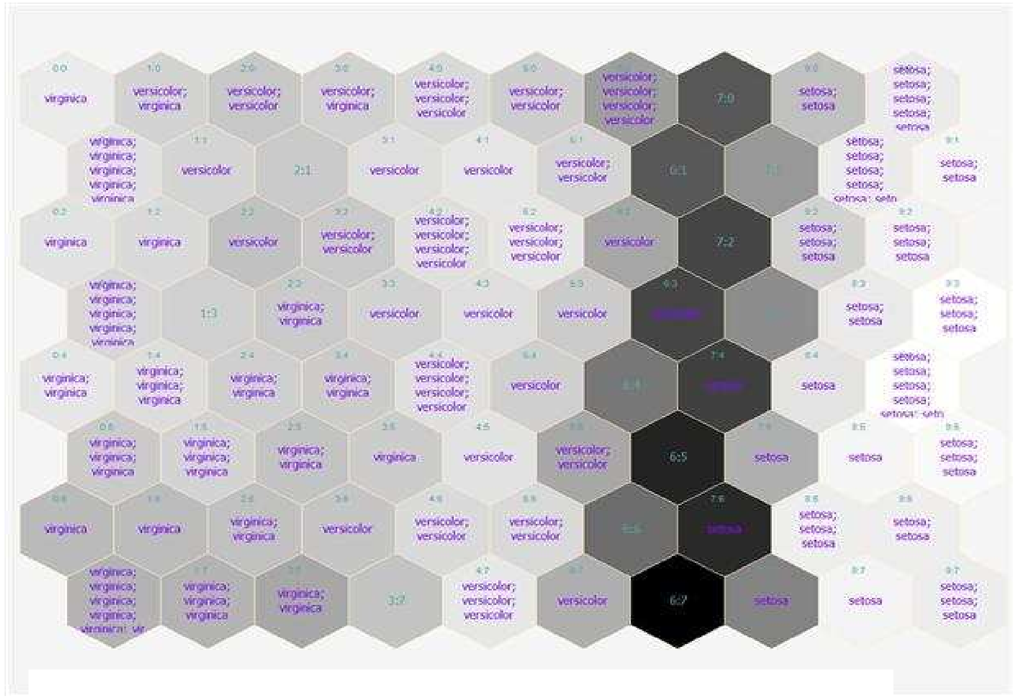


Fig.3. Output Map of SOM, trained by Spice-SOM with Iris Data

Label: virg = virginica   vers = versicolor   seto = setosa

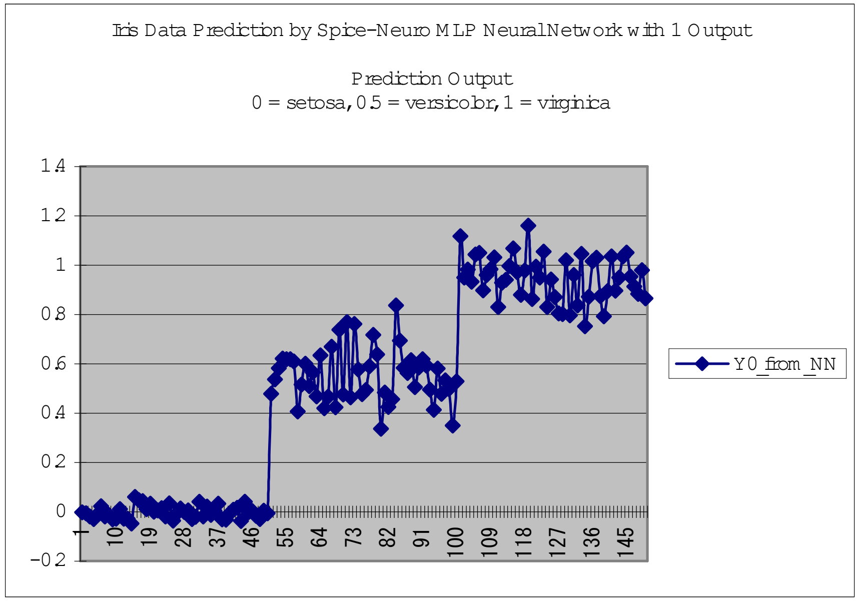
No.	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9
Y0	virg	vers; virg	vers	vers; virg	vers	vers	vers		seto	seto
Y1	virg	vers		vers	vers	vers			seto	seto
Y2	virg	virg	vers	vers	vers	vers	vers		seto	seto
Y3	virg		virg	vers	vers	vers	vers		seto	seto
Y4	virg	virg	virg	virg	vers	vers		seto	seto	seto
Y5	virg	virg	virg	virg	vers	vers		seto	seto	seto
Y6	virg	virg	virg	vers	vers	vers		seto	seto	seto
Y7	virg	virg	virg		vers	vers		seto	seto	seto

1.3. Classify Data with SpiceNeuro

Classify Iris data with MLP NN, we have output graphs as shown in Fig.4.

With MLP 1 output, we notice that with threshold 0.2 the MLP NN classifies 100% correctly for setosa species. With threshold 0.8 there is 1 dataset of versicolor and 1 dataset of virginica that are misclassified.

With MLP MLP 3 outputs, we can see with threshold 0.5 the MLP NN classifies 100% correctly for setosa but there are misclassified 4 dataset of versicolor and 3 dataset of virginica. In this case classification of MLP 1 output is better than classification of MLP 3 outputs.



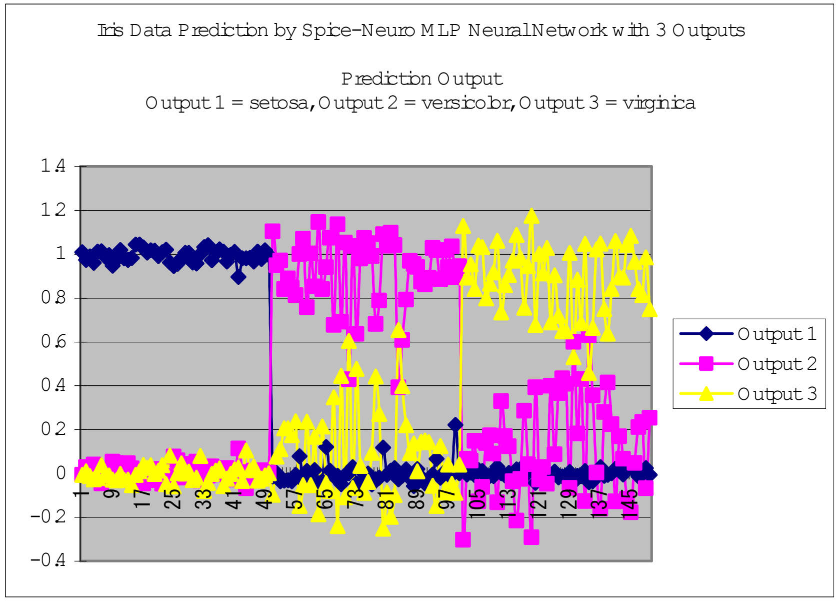


Fig.4. Output Graph of MLP with Iris Data

2. Students's Score Data

Suppose that we have a table of students' scores in a semester. By SOM, we want to group students into several groups based on their scores.

2.1. Data Preparation

To let Spice-SOM can read the score data, we prepare the data as the following table 3. Please note that after a student' name are his/her average score in brackets just for easily understanding the output map. You can see this score data on "Data" folder of SpiceSOM.
Have a SOM size 6x10 learned, we have an output map and output table as Figs.5-6.
On the output map, you can easily see that students whose high grades are arranged closer each other, and low grade students are also closer each other.

Bảng 3. Dữ liệu điểm học sinh cho SOM

No	English	Algebra	Geome try	…	Analysis	Power System	Management Methodology	Geological System	Name
1	5	7	7	…	7	5	5	5	Pham Kieu Anh (6.0)
2	5	9	8	…	6	8	6	5	Cung Hong Hien (7.0)
…	…	…	…	…	…	…	…	…	…
99	7	8	9	…	4	7	9	7	Vo Mai Manh (6.4)
100	7	7	7	…	5	6	7	7	Pham La Trinh (6.0)

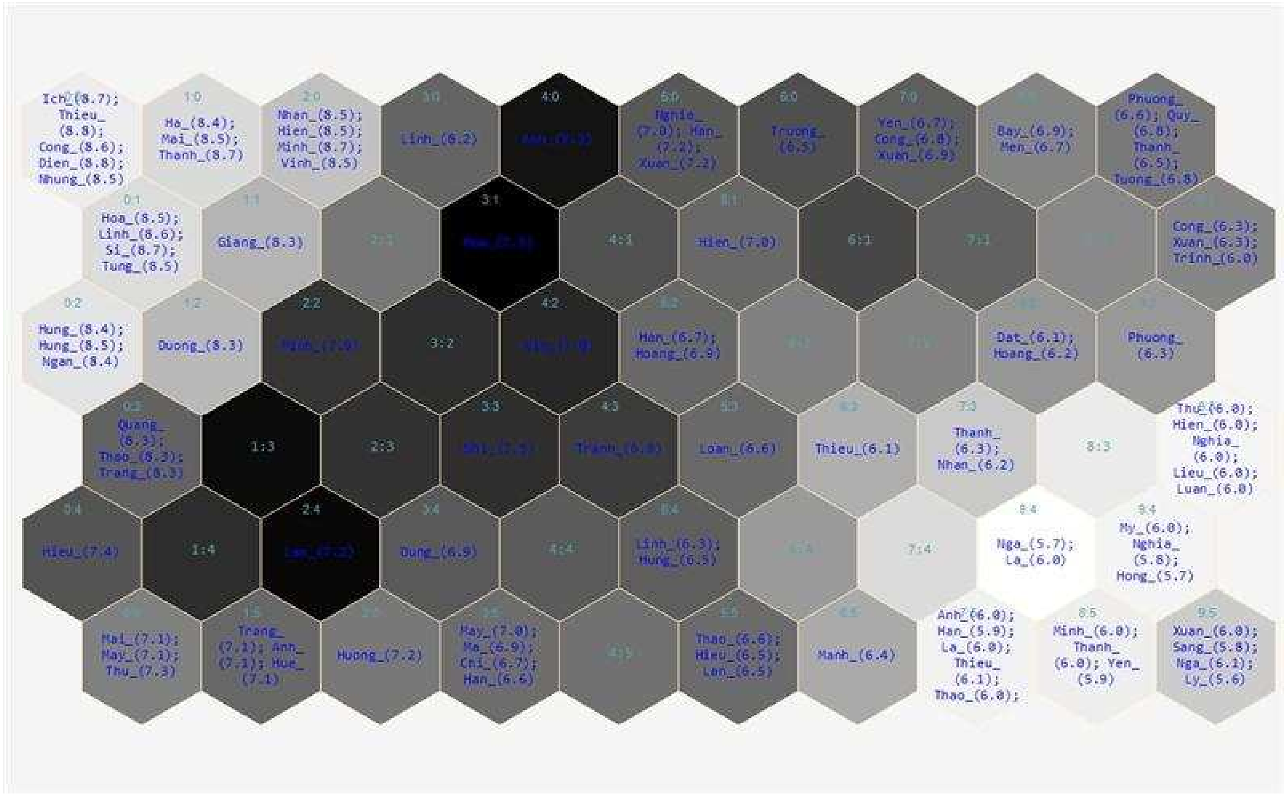


Fig 5. Output Map of SOM, trained by Spice-SOM with Students' score Data

No.	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9
Y0	Ich_(8.7);Thieu_(8.8);Cong_(8.6);Dien_(8.8);Nhung_(8.5)	Ha_(8.4);Mai_(8.5);Thanh_(8.7)	Nhan_(8.5);Hien_(8.5);Minh_(8.7);Vinh_(8.5)	Linh_(8.2)	Anh_(7.3)	Nghia_(7.0);Han_(7.2);Xuan_(7.2)	Truong_(6.5)	Yen_(6.7);Cong_(6.8);Xuan_(6.9)	Bay_(6.9);Men_(6.7)	Phuong_(6.6);Quy_(6.8);Thanh_(6.5);Tuong_(6.8)
Y1	Hoa_(8.5);Linh_(8.6);Si_(8.7);Tung_(8.5)	Giang_(8.3)		Hoa_(7.5)		Hien_(7.0)				Cong_(6.3);Xuan_(6.3);Trinh_(6.0)
Y2	Hung_(8.4);Hung_(8.5);Ngan_(8.4)	Duong_(8.3)	Minh_(7.9)		Kim_(7.0)	Han_(6.7);Hoang_(6.9)			Dat_(6.1);Hoang_(6.2)	Phuong_(6.3)
Y3	Quang_(8.3);Thao_(8.3);Trang_(8.3)			Nhi_(7.5)	Tranh_(6.9)	Loan_(6.6)	Thieu_(6.1)	Thanh_(6.3); Nhan_(6.2)		Thu_(6.0);Hien_(6.0);Nghia_(6.0);Lieu_(6.0);Luan_(6.0)
Y4	Hieu_(7.4)		Lan_(7.2)	Dung_(6.9)		Linh_(6.3);Hung_(6.5)			Nga_(5.7);La_(6.0)	My_(6.0);Nghia_(5.8);Hong_(5.7)
Y5	Mai_(7.1);May_(7.1); Thu_(7.3)	Trang_(7.1); Anh_(7.1);Hue_(7.1)	Huong_(7.2)	May_(7.0);Ma_(6.9);Chi_(6.7);Han_(6.6)		Thao_(6.6);Hieu_(6.5);Lan_(6.5)	Manh_(6.4)	Anh_(6.0);Han_(5.9);La_(6.0);Thieu_(6.1);Thao_(6.0); Minh_(5.8)	Minh_(6.0);Thanh_(6.0);Yen_(5.9)	Xuan_(6.0);Sang_(5.8);Nga_(6.1);Ly_(5.6)

Fig. 6. Output Table of SOM, trained by Spice-SOM with Students' score Data

3. Face/Nonface Classification

To detect faces on images, a popular method is use Haar-like feature and Adaboost algorithm. SOM and MLP NN can be used here but the recognition speed is slow. Here we present examples using MLP NN and SOM to classify face frames, to illustrate how to use MLP NN and SOM on the practice.

3.1. Data Preparation

Length of a data vector depends on its feature. Suppose we have a set of images size *m x n*, if we use pixel value as feature we will have a *m x n* - length vector for each image. If we use pixel value histogram as feature, we will have a 256 - length for each image. If we use other features, Histogram of Oriented Gradient for instance, represented vector will depend on parameters from that the feature was created. This material does not focus on how to create image features.
To classify images by MLP, besides feature vectors we should code desired output, for example we use one output with value 1.0 as face, 0.0 as non face. To classify images by SOM, we need just need feature vectors and their labels as illustrated in Figs.7-8.

ID	feature vector	Output	label
1		1.0	face
2		1.0	face
…	…	…	…
n		0.0	nonfacce
n+1		0.0	nonfacce
…	…	…	…

Fig.7. Data for MLP NN

ID	feature vector	label
1		face
2		face
…	…	…
n		nonfacce
n+1		nonfacce
…	…	…

Fig.8. Data for SOM

3.2. Classify face and non-face images by Spice-SOM

The following figures 9 and 10 illustrate Output Maps of SOM after several training with different SOM sizes. Training data is 400 face images downloaded from http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html and 700 non-face downloaded randomly from internet. Data vectors are histogram of gradient inputs, length is 324. On this illustrated output map, a neuron just display the first images fallen on it (in fact there are some neurons on that several images are fallen, and there are also neurons on that no images are fallen).

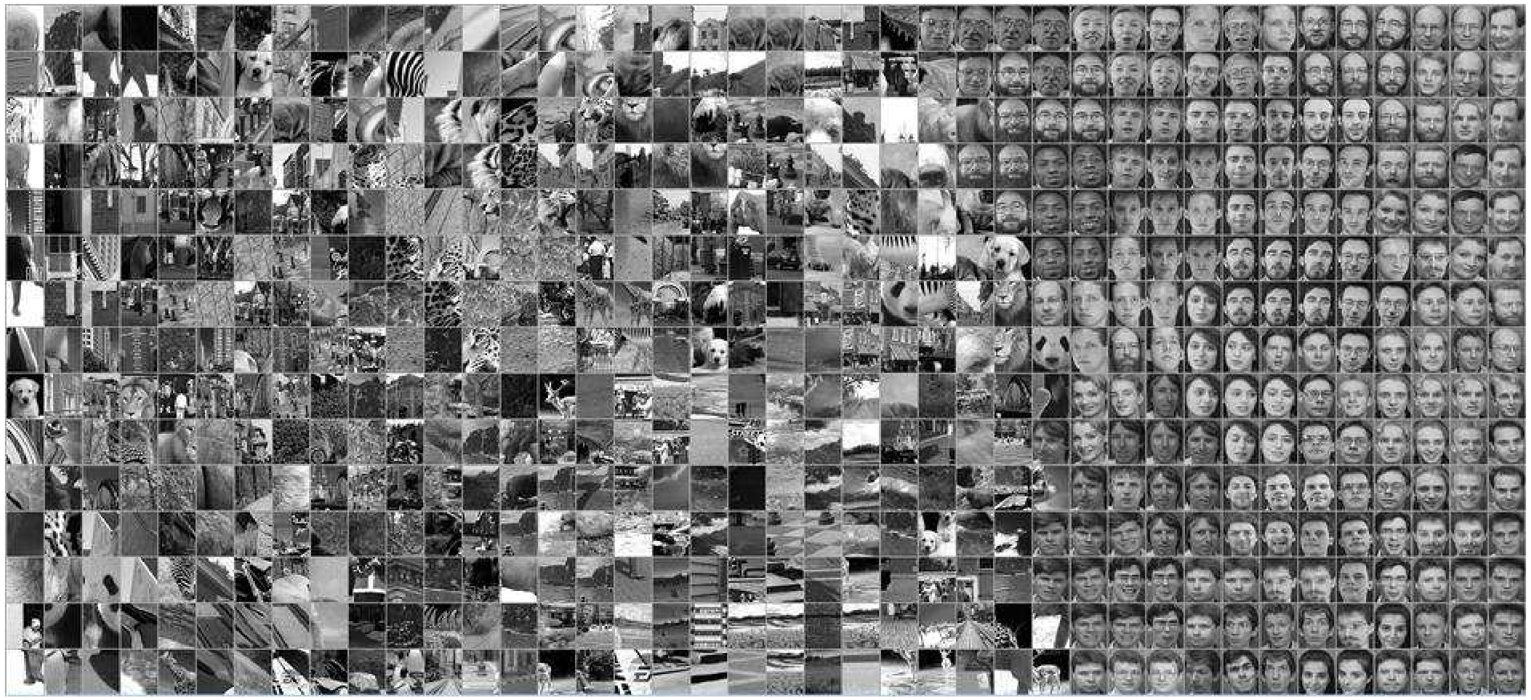


Fig. 9. Output Map of a SOM

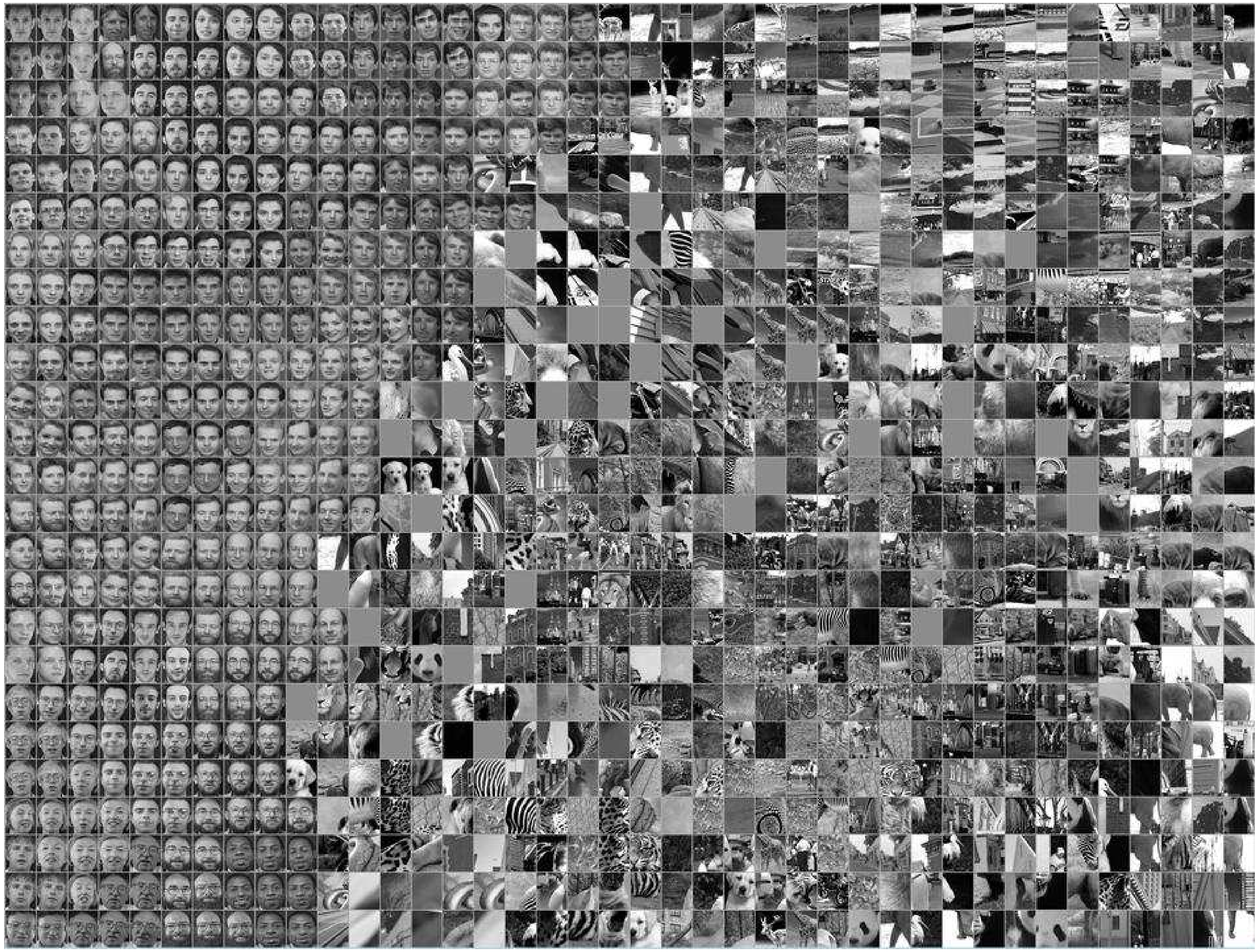
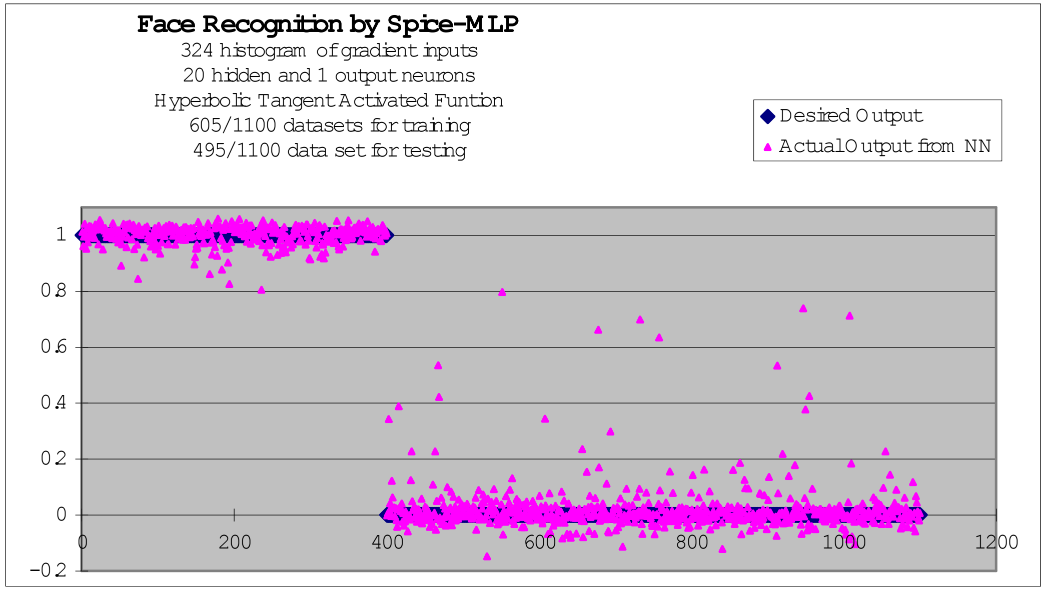


Fig.10. Output Map of a SOM

3.3. Classify face and non-face images by Spice-MLP

Fig.11 shows training error, actual output from NN and desired output, feature are histogram of gradient feature whose length is 324. The MLP has 20 hidden and 1 output neurons, Hyperbolic Tangent Activated Functions are used on both hidden and output layers. 605/1100 datasets were used for training (55%) and 495/1100 data set were used for testing (45%).

Fig.12 shows Receiver Operating Characteristic – ROC and accuracies with several thresholds on testing set. We can see when threshold for face/non-face is 0.52, the MLP classify correctly 98.7% this testing set.



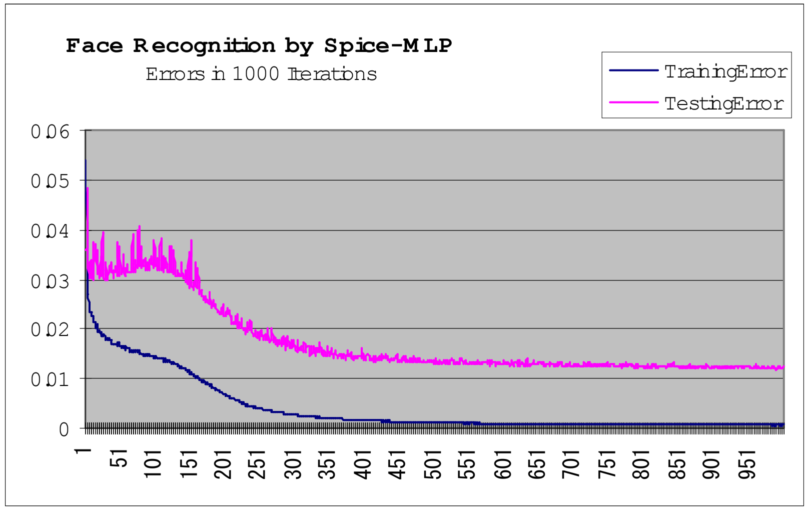
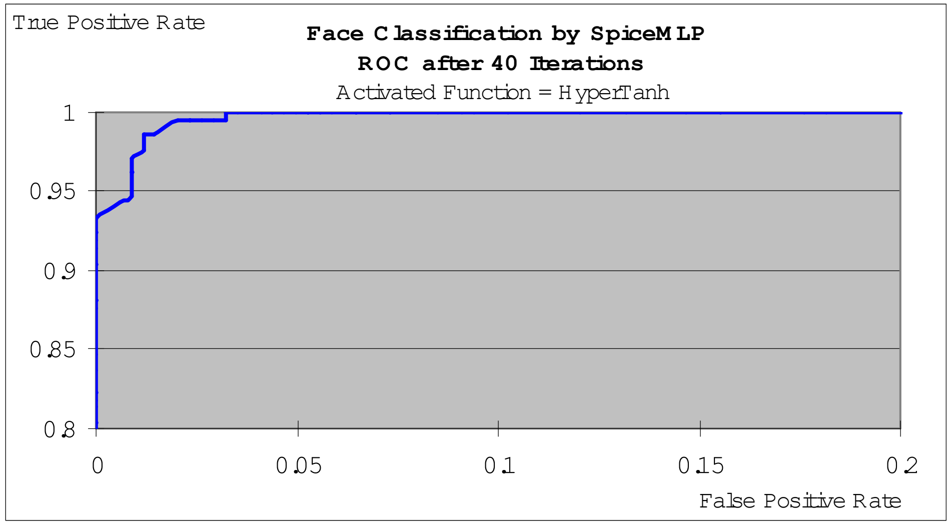


Fig.11. Outputs and Training Errors of a MLP with face/non-face data



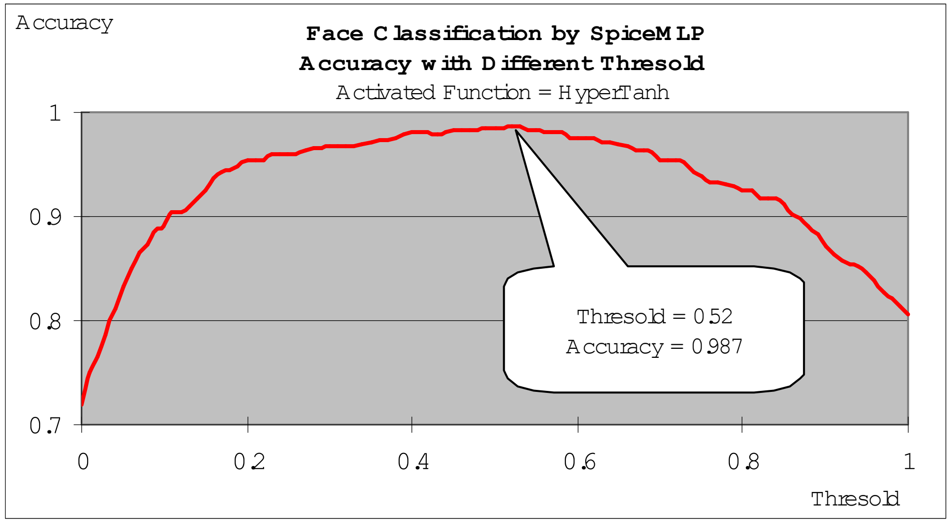


Fig.12. ROC and Accuracy Lines by a MLP with face/nonface data

4. Classify pedestrians

4.1. Data Preparation

Data on this example is got from 924 pedestrian images (http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html) and 1100 non pedestrian images randomly downloaded from internet..

4.2. Classify pedestrians by SpiceSOM

The following Figs.13-14 illustrate Output Maps of SOM after several training with different SOM size, feature is histogram of gradient inputs and feature length is 810.

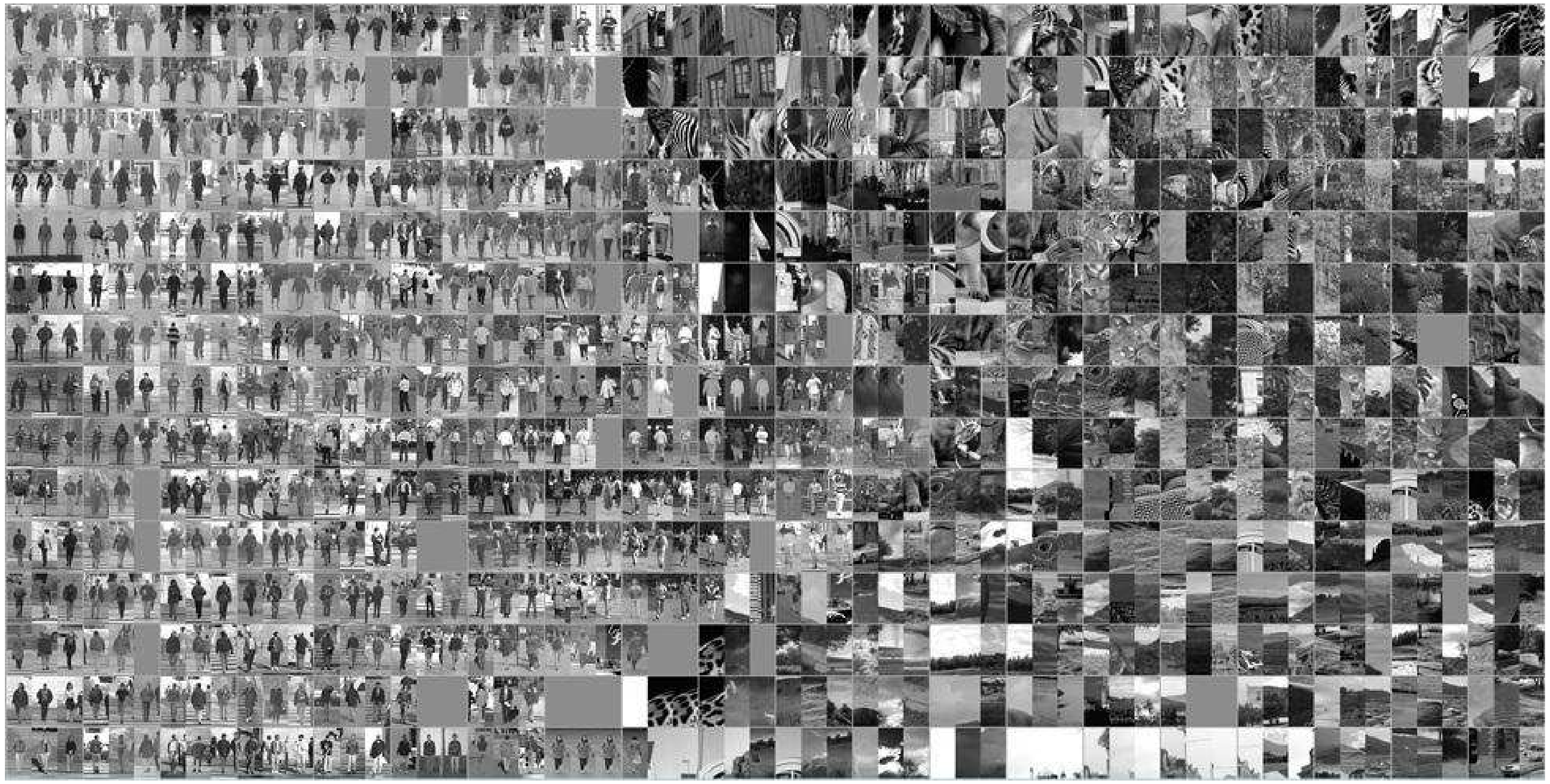


Fig.13. Output Map of a SOM with Pedestrian Data

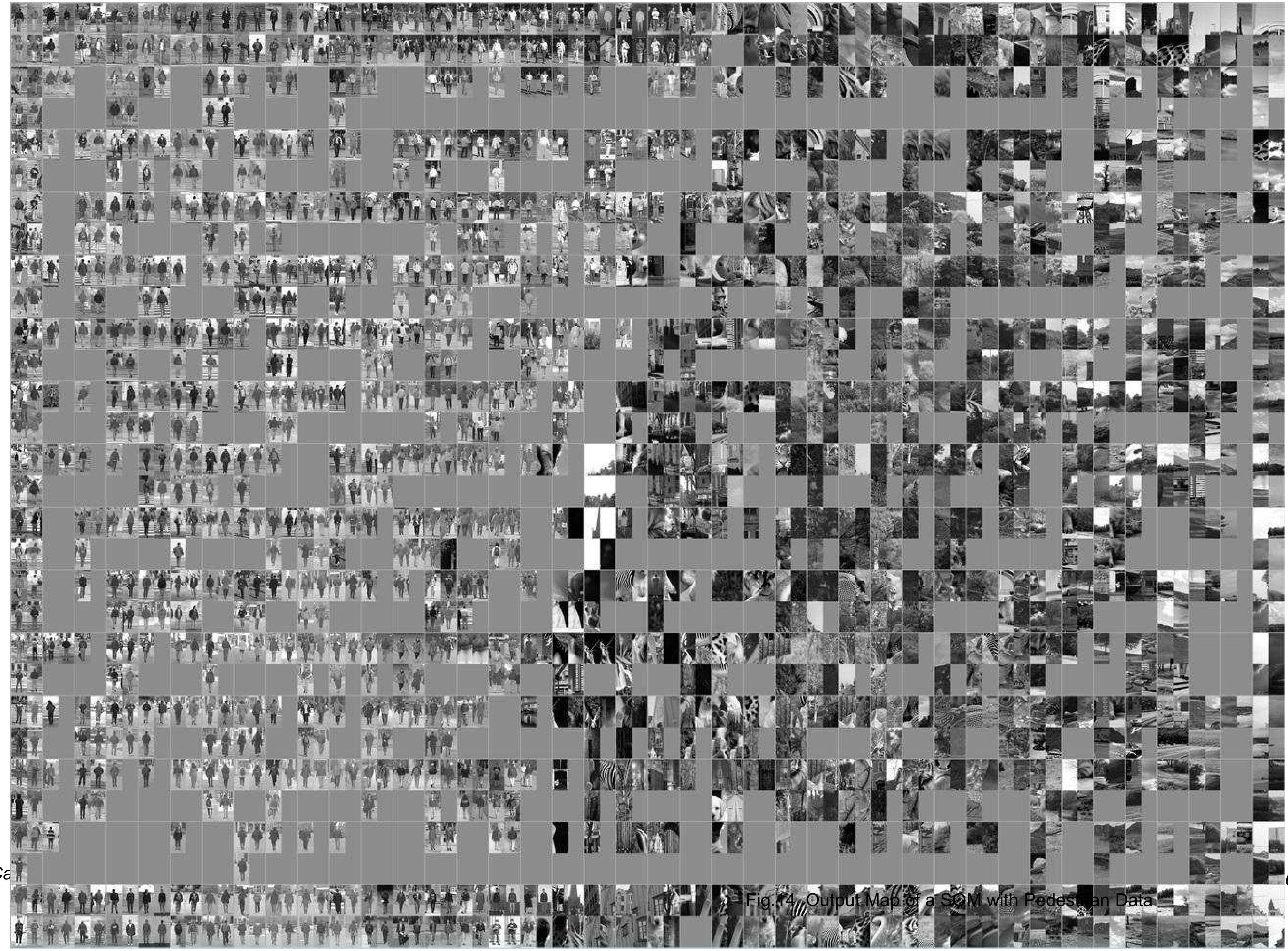


Fig. 14 Output Map of a SOM with Pedestrian Data

4.3. Classify pedestrians by SpiceMLP

Fig.15 shows training error, actual output from NN and desired outputs of a MLP 5 hidden and 1 output neurons, Hyperbolic Tangent Activated Function are used in both hidden and output layers. 1113/2024 datasets were used for training (55%) and 911/2024 data set were used for testing (45%).
Fig.16 illustrates ROC curves on testing set when MLP NN just initialized (without training) and after training with 1, 2, 3, 10, 20 iterations. We can notice that when MLP initializes, ROC curve looks like ROC curve of a random selection. The MLP NN converges relatively fast after several iterations.

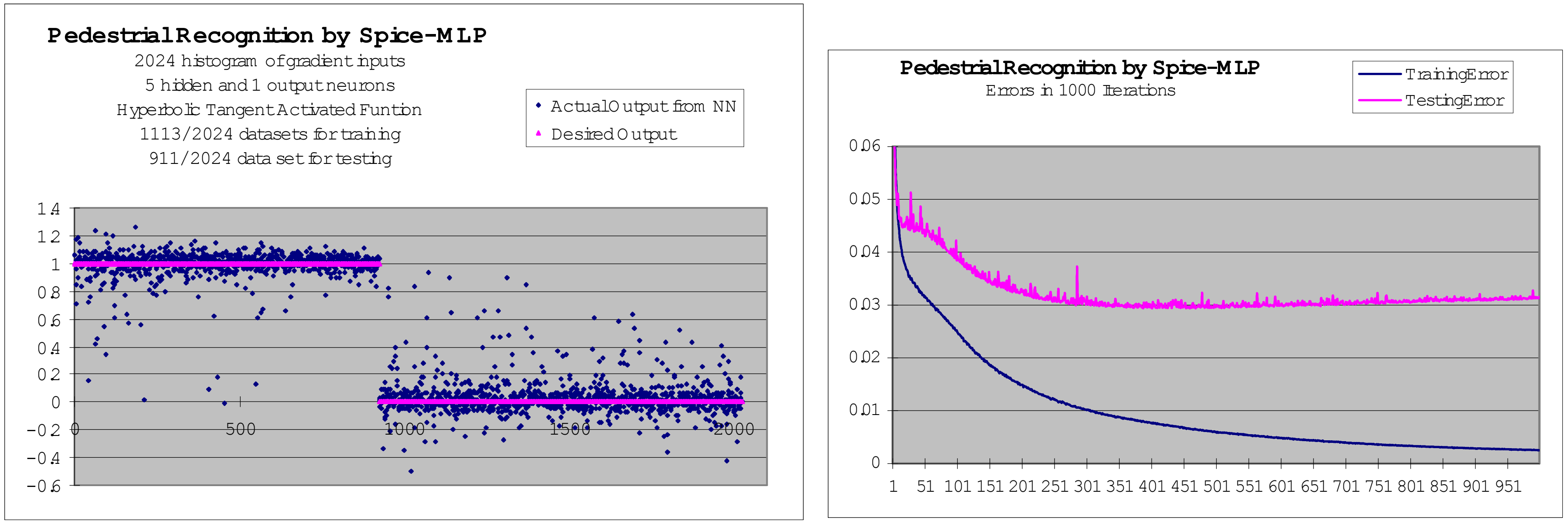


Fig.15. Outputs and Training Errors of a MLP with Pedestrial Data

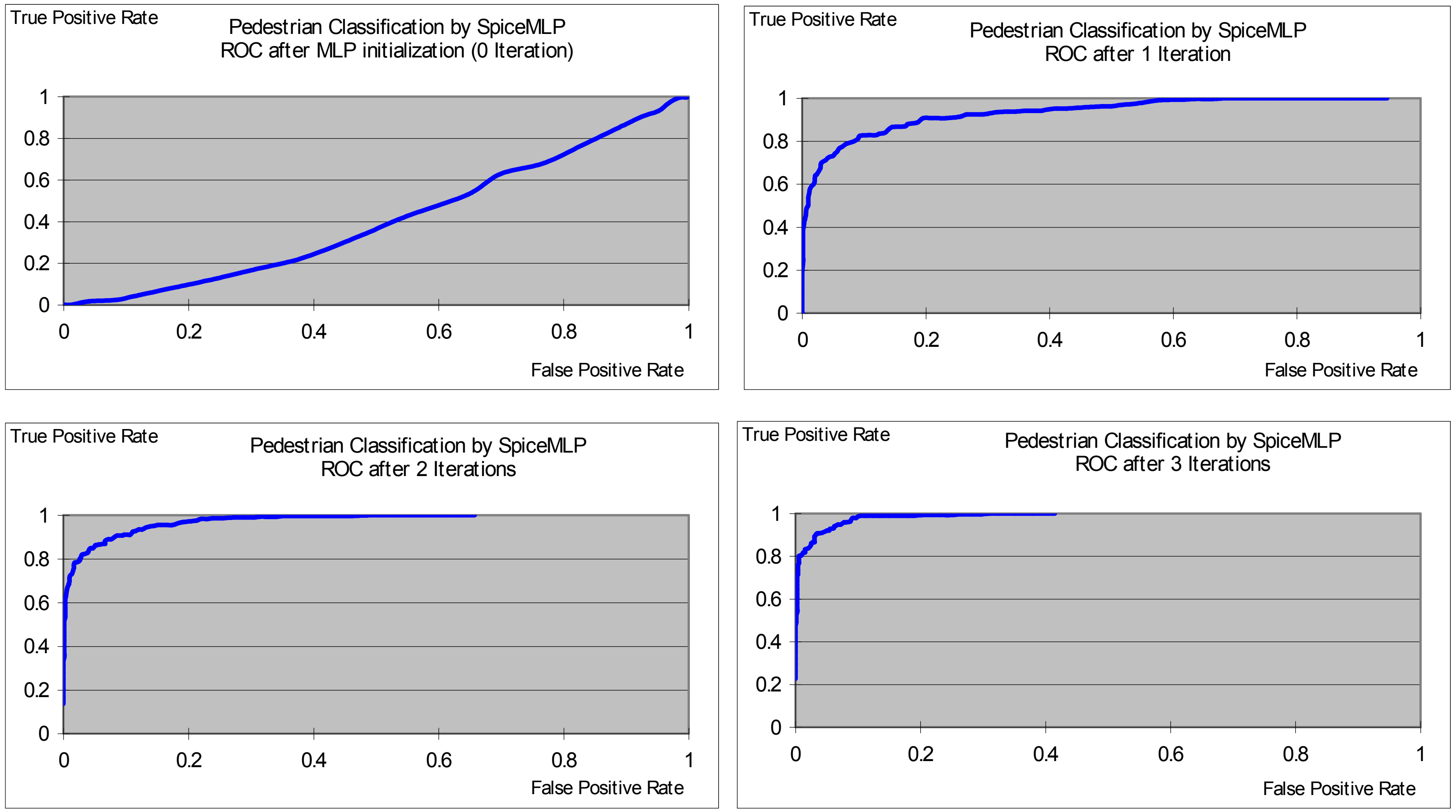
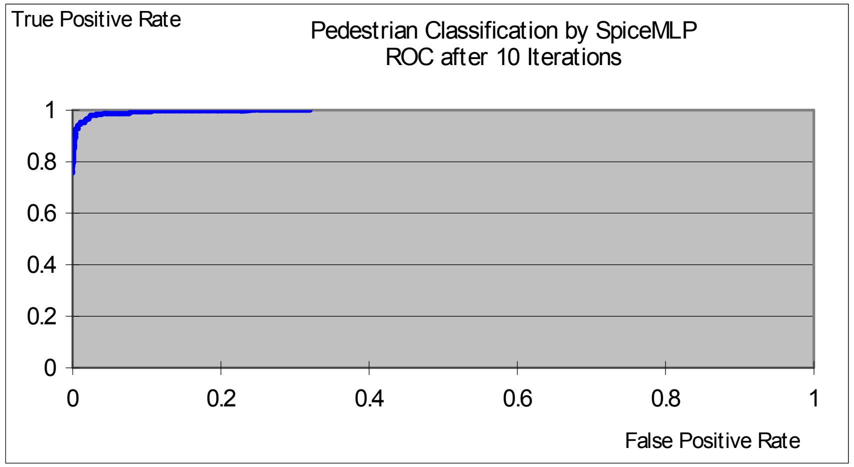


Fig.16. ROC after several training iteration (continued in next page)



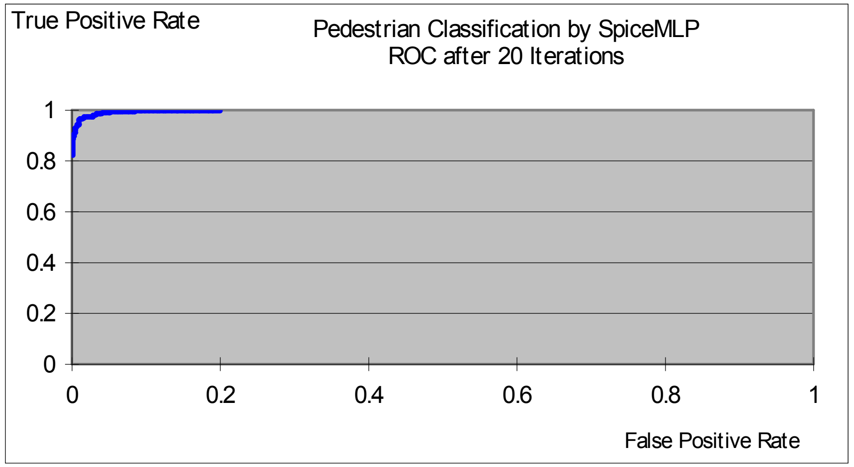
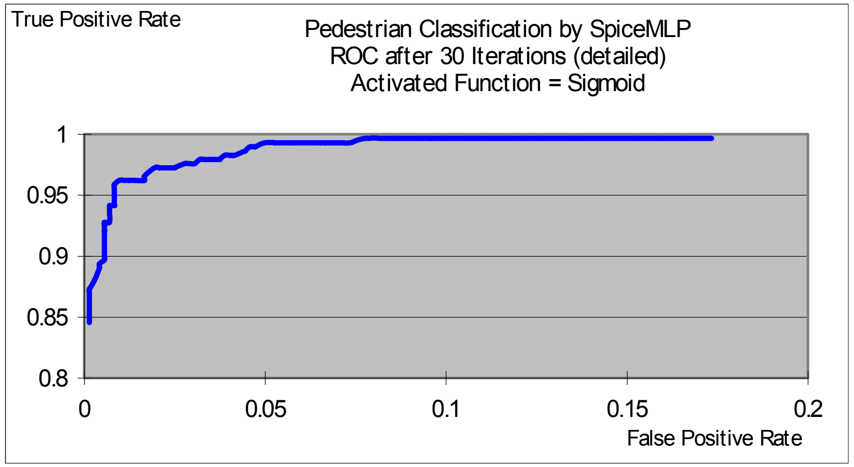


Fig.16. ROC after several training iteration (continued)

Fig.17 illustrates receiver operating characteristic (ROC curve) and accuracies with several thresholds after 30 training iterations. For the testing sets, with pedestrial/non-pedestrial threshold = 0.75, MLP neural network classifies correctly 98.2%.



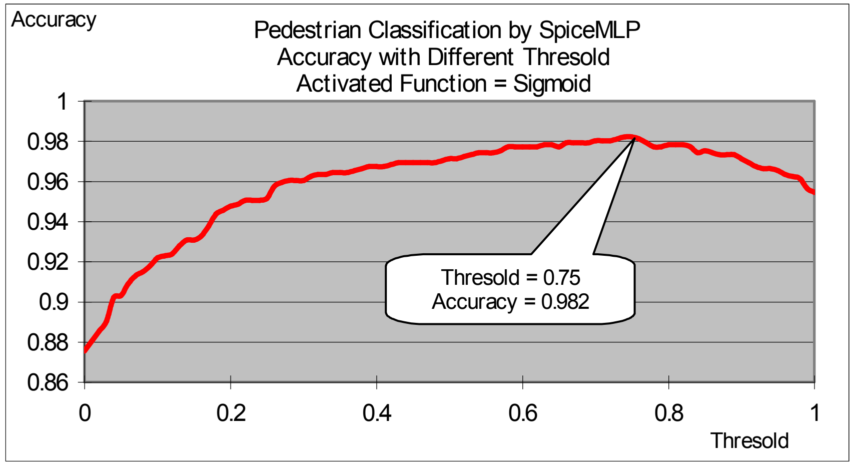


Fig.17. ROC and Accuracy Lines by a MLP with pedestrial data

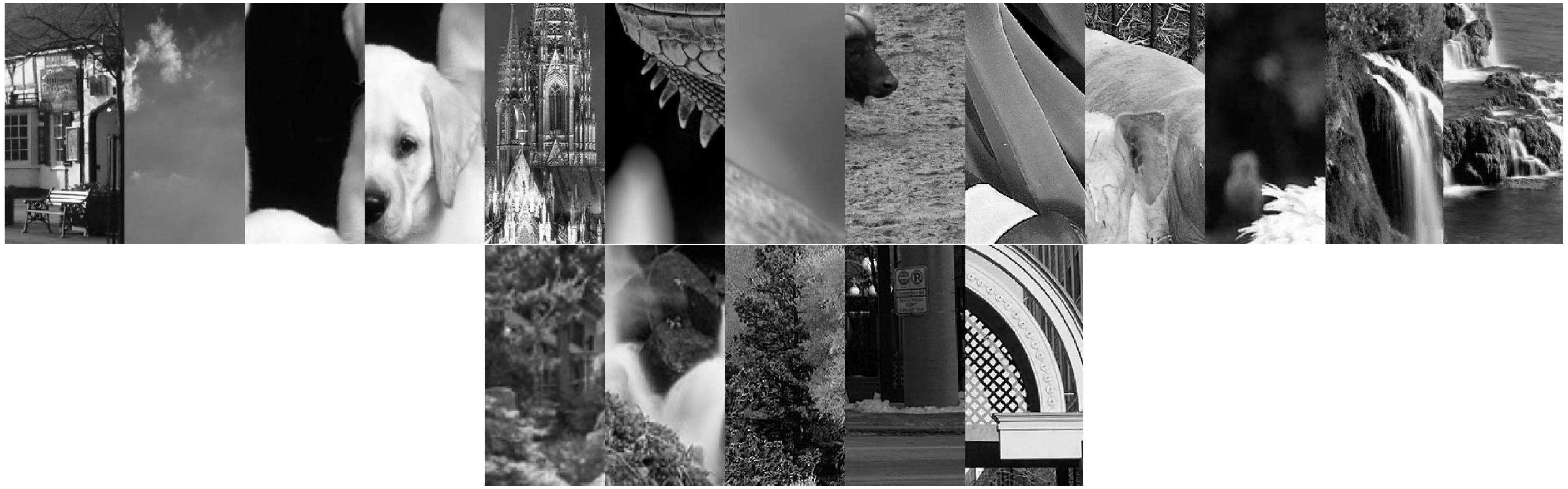


Fig.18.a. Some images that are misclassified non-pedestrial to pedestrial with pedestrial/non-pedestrial threshold = 0.5

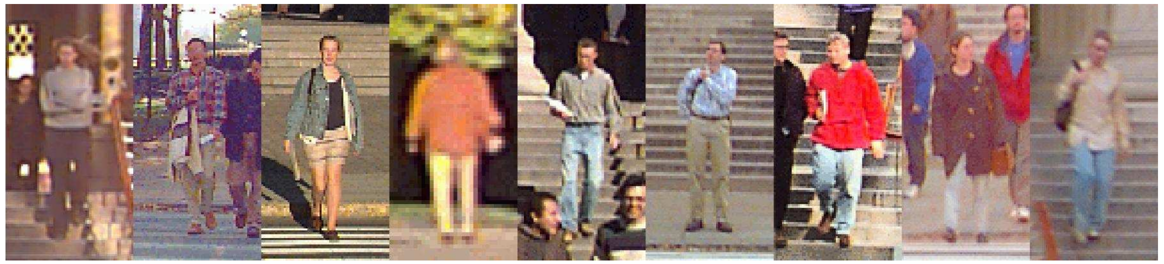


Fig.18.b. Some images that are misclassified pedestrian to non-pedestrian with pedestrial/non-pedestrial threshold = 0.5.

5. Car Image Classification

As same as face and pedestrian classifications, the following figs illustrate Output Maps of SOM after several training with different SOM sizes, car samples are downloaded from http://cogcomp.cs.illinois.edu/Data/Car/

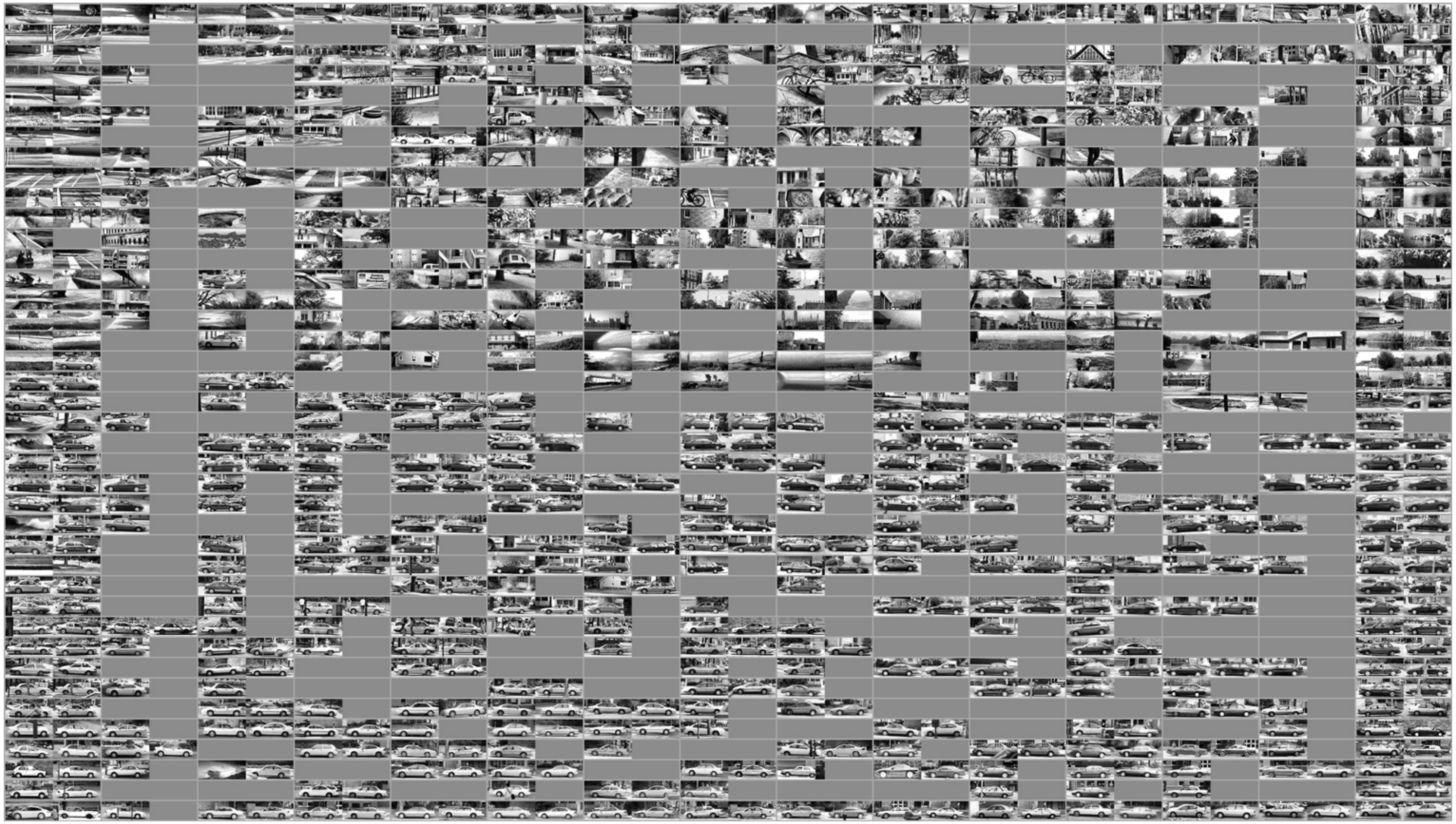


Fig.19.a. Output Map of a SOM with Car Data

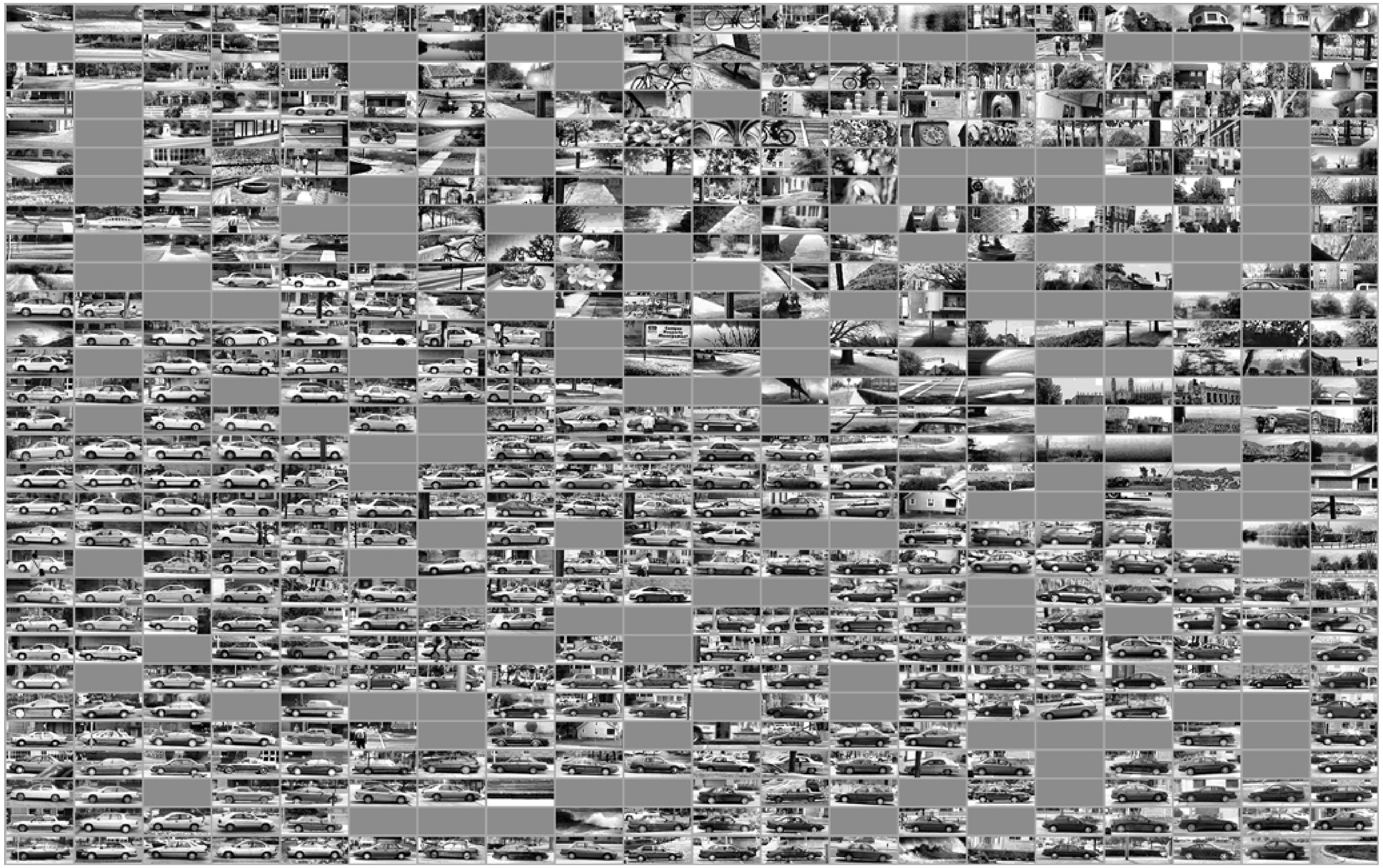


Fig.19.b. Output Map of a SOM with Car Data

6. Stock Index Prediction

One of interesting applications of NN is stock price prediction. Based on statistical market data, NN can predict stock price in next few days with a high accuracy.

To predict stock price accurately by NN, the most importance thing is to find suitable market data including prices, information of macro and micro economic ..., appropriately code them so that the NN can learn and generalize rules and relations hidden inside the data.

There are several methods to predict stock price by NN. This material presents a simple method: predict stock prices in next few days based on the past prices.

6.1. Data Preparation

You can download index of NASDAQ Stock Data from http://www.dailyfinance.com/historical-stock-prices/ For example from 3_12_1990 to 1_12_2010, data would have form as shown in table 4.

Table 4. NASDAQ Stock Price

High	Low	Open	Close	Day_Month_Year
362.2798	359.0498	361.3198	361.3198	3_12_1990
364.2898	360.4199	364.1099	364.1099	4_12_1990
370.9700	364.0198	370.8699	370.8699	5_12_1990
378.0298	370.9199	372.2898	372.2898	6_12_1990
372.3499	369.0198	371.5398	371.5398	7_12_1990
…	…	…	…	…
2545.4100	2515.4800	2519.8900	2543.1200	24_11_2010
2541.4900	2522.4000	2525.9100	2534.5600	26_11_2010
2531.0300	2496.8300	2522.2400	2525.2200	29_11_2010
2510.7100	2488.6100	2497.1200	2498.2300	30_11_2010
2558.2900	2534.8100	2535.1900	2549.4300	1_12_2010

From this data, you want to predict index for next days. For instance you want to predict average index (Open + Close)/2.0 based on past average indexes.

The data you have now is time series data, to have it learnt by MLP NN you should re-prepare it. Here is an example.
From columns of Open and Close indexes, make a new column that contains average values of the Open and Close indexes.
Suppose you want to use average indexes of 15 past days to predict index of one next day, so you make a row that consists 16 average indexes of 16 continuous days. That also means you will use a MLP NN with 15 inputs and one output. Your data should look like data in table 5:

Table 5. NASDAQ Stock Index data that are re-prepared so that it can be leant by a MLP NN

Today - 14	Today - 13	Today - 12	…	Today - 2	Today - 1	Today	Tomorrow	Label
361.3198	364.1099	370.8699	…	371.22	372.2998	373.5999	372.4099	21_12_1990
364.1099	370.8699	372.2898	…	372.2998	373.5999	372.4099	372.3999	24_12_1990
370.8699	372.2898	371.5398	…	373.5999	372.4099	372.3999	371.0498	26_12_1990
372.2898	371.5398	371.47	…	372.4099	372.3999	371.0498	371.2	27_12_1990
…	…	…	…	…	…	…	…	…
2573.305	2578.305	2575.455	…	2520.705	2499.58	2531.505	2530.235	24_11_2010
2578.305	2575.455	2575.03	…	2499.58	2531.505	2530.235	2523.73	26_11_2010
2575.455	2575.03	2571.545	…	2531.505	2530.235	2523.73	2497.675	29_11_2010
2575.03	2571.545	2544.88	…	2530.235	2523.73	2497.675	2542.31	30_11_2010

Similarly, if you want to use indexes of 20 past days to predict indexes of next 3 days, you make your data like the above procedure and use a MLP NN with 20 inputs and 3 outputs.
The following example uses average index of 15 past days to predict index of next day (15 inputs, 1 putput), from 21_12_1990 to 30_11_2010, by SpiceMLP. Number of Datasets is 5026. 80% (4021) datasets used for training and 20% (1005) datasets used for testing. The learning iterations is 1000.

```
Number of trained data: 4021.  Number of tested data: 1005
Taken iterations: 1000         Number of Inputs: 15
Number of Outputs: 1
```

This data is already on "Data" folder when you install the SpiceMLP neural network program.

6.2. Finding suitable parameters for the MLP NN

To let MLP NN generalize your data well we need to assign it suitable parameters. Normally MLP parameters depend much on your data. A parameter may good for this data but bad for the others. Here we introduce a simple finding method: with the same training and testing data, change a parameter to find its relatively optimum value. Please note that before training, we should normalize inputs and outputs data. In this example we normalize data by Linear method.

First, we find a suitable number of hidden neurons.

Table 6. Errors when using HyperTanh for hidden and output layers, changing number of hidden neurons.

Number of Hidden Neurons	Errors	
	Training	Testing
15	9.01E-05	9.35E-05
12	9.22E-05	1.12E-04
10	8.62E-05	1.04E-04
8	9.01E-05	9.36E-05
6	6.54E-05	6.94E-05
4	6.49E-05	6.65E-05
3	4.43E-05	4.25E-05
2	4.28E-05	4.01E-05

We can see that number of hidden neuron = 2 may be better. Next we find activated function for the output layer.

Table 7. Errors when using HyperTanh activated function for hidden layer, number of hidden neurons = 2, changing activated function of output layer.

Activated Function		Errors	
Hidden Layer	Output Layer	Training	Testing
HyperTanh	Identity	3.94E-05	3.38E-05
HyperTanh	Sigmoid	7.35E-05	1.08E-04
HyperTanh	ArcTan	6.97E-05	6.41E-05
HyperTanh	ArcSinh	6.24E-05	6.47E-05
HyperTanh	Sin	5.05E-05	4.83E-05
HyperTanh	Gaussian	5.48E-05	5.77E-05
HyperTanh	XSinX	5.54E-05	5.53E-05

We can see that activated function for output layer = Identity seems better. Finally we find activated function for hidden layer.

Table 8. Error when using Identity activated function for output layer, number of hidden neurons = 2, changing activated function of hidden layer.

Activated Function		Errors	
Hidden Layer	Output Layer	Training	Testing
Sigmoid	Identity	4.33E-05	3.93E-05
HyperTanh	Identity	3.94E-05	3.38E-05
ArcTan	Identity	4.03E-05	4.37E-05
ArcSinh	Identity	4.05E-05	3.99E-05
Sin	Identity	4.07E-05	4.18E-05
Gaussian	Identity	4.43E-05	4.25E-05
XsinX	Identity	4.72E-05	4.98E-05

We can see that activated function for hidden layer = HyperTanh seems better.
So we choose number of hidden neurons = 2, activated function for hidden layer = HyperTanh, activated function for output layer = Identity.

6.3. Train the MLP NN

Train your MLP NN several times and select one with smallest training and testing error. You will have trained information like the following

```
Last trained information
 Activated Function for Hidden Layer: HyperTanh
 Activated Function for Output Layer: Identity
 Final Learning rate: 0.03308719
 Final MSE of Training Set: 4.138118E-05
 Final MSE of Testing Set: 3.423549E-05
 Number of trained data: 4021
 Number of tested data: 1005
 Taken iterations: 1000
```

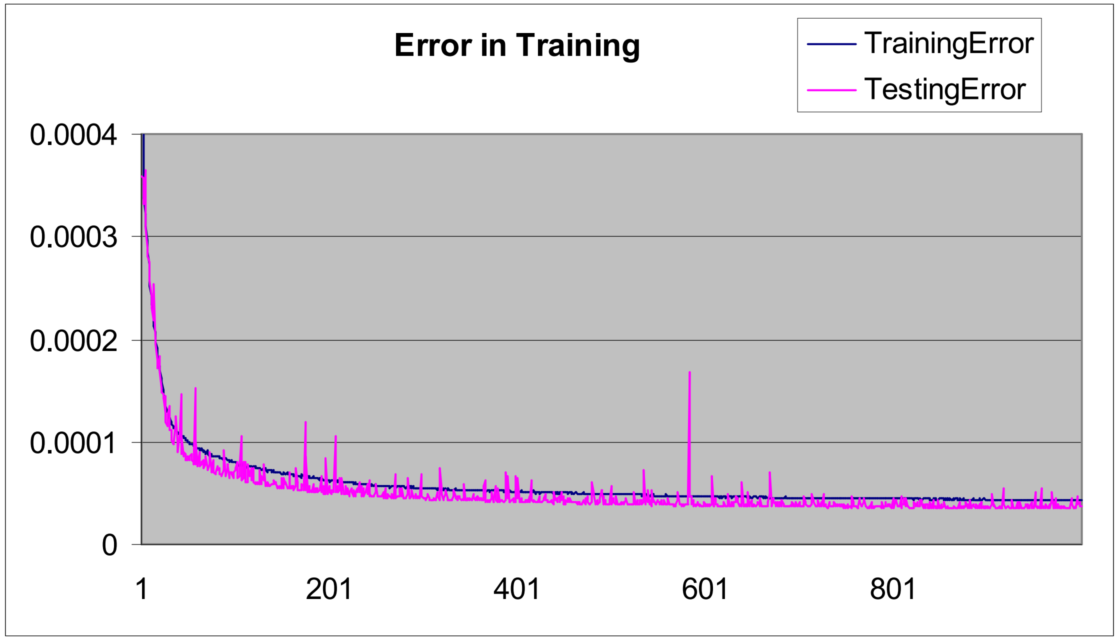


Fig. 20. Training Error when learning NASDAQ Stock prices

6.4. Test the modeling data

After training the MLP NN, we can see output graphs of MLP NN for training or testing data on "Data Visualization" Tab. The graphs look like the graph shown on Fig.21

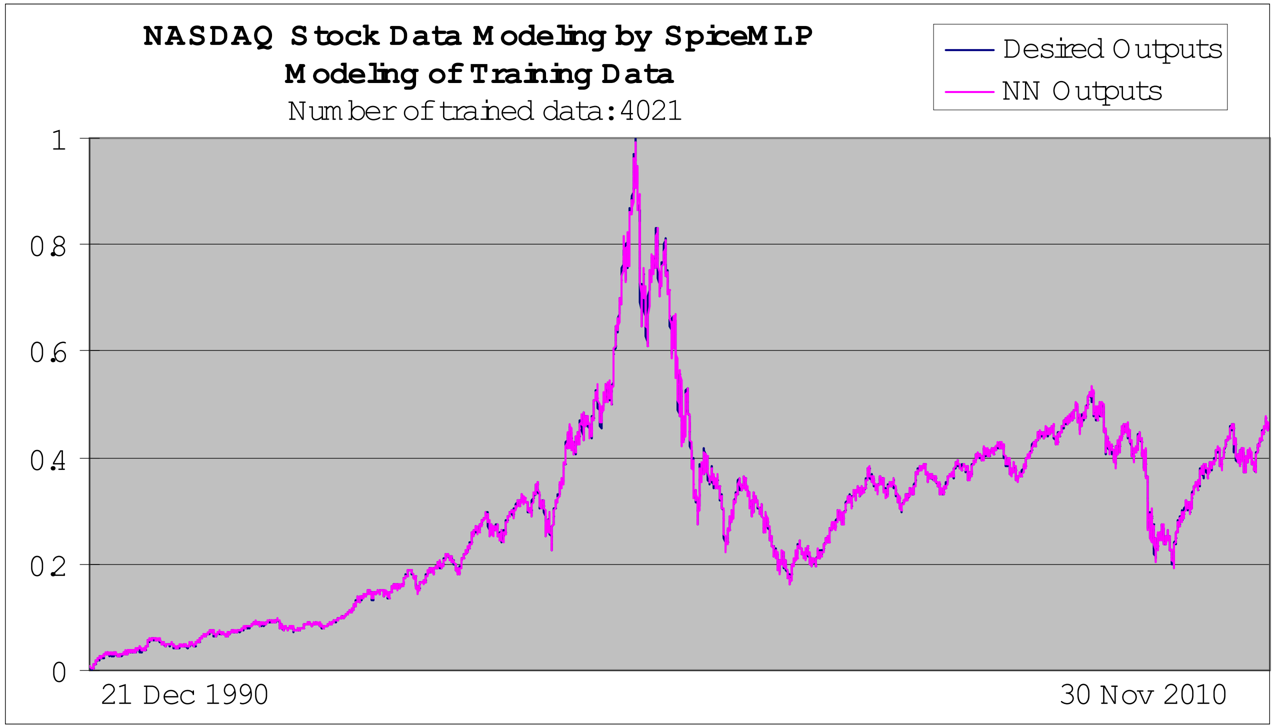


Fig. 21. Outputs of Training Data (NASDAQ Stock prices)

We can see that with training data, outputs of MLP NN are almost as same as desired outputs. With the testing data, we have the graph shown on Fig.22.a:

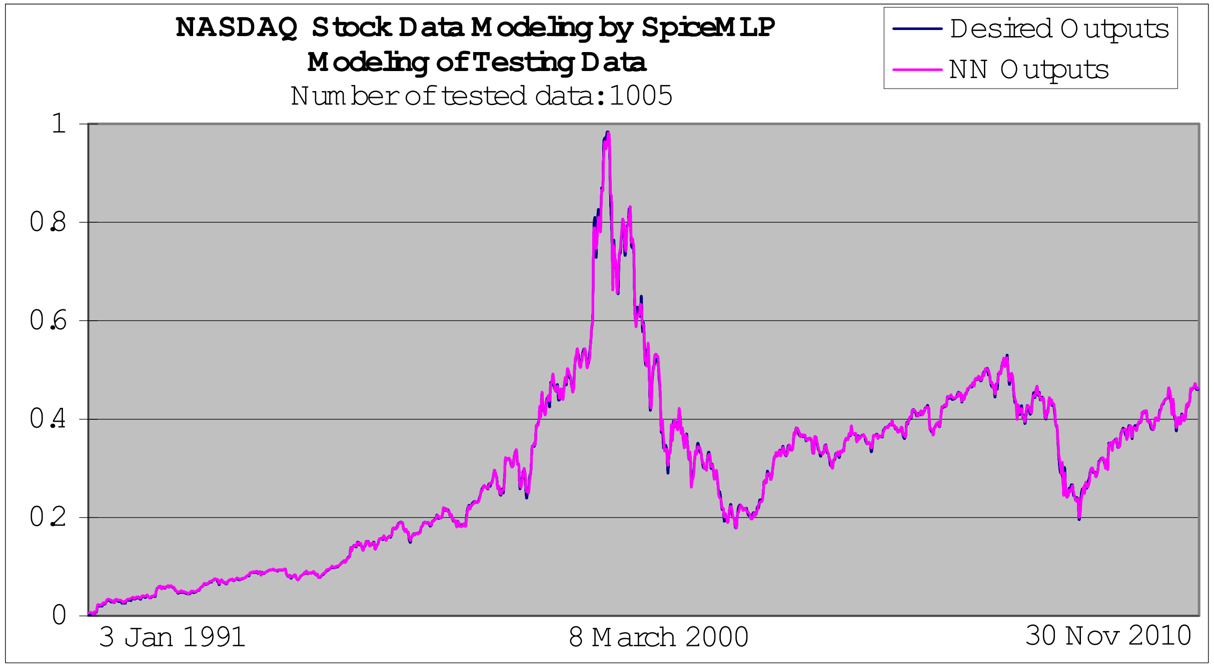


Fig. 22.a. Outputs of Testing Data (NASDAQ Stock prices)

In short term we have the graph shown on Fig.22.b.

With testing data, outputs of MLP NN are also almost as same as desired outputs. It is easy to see that MLP NN learn well with our data, how ever there are some points that have small errors.

Here are some question for you

- ✧ How to reduce error in MLP prediction?
- ✧ Are there any other data that can be used for stock prediction?
- ✧ Could we apply the above method to other problem?
- ✧ How we prepare data for non-time series broblem?

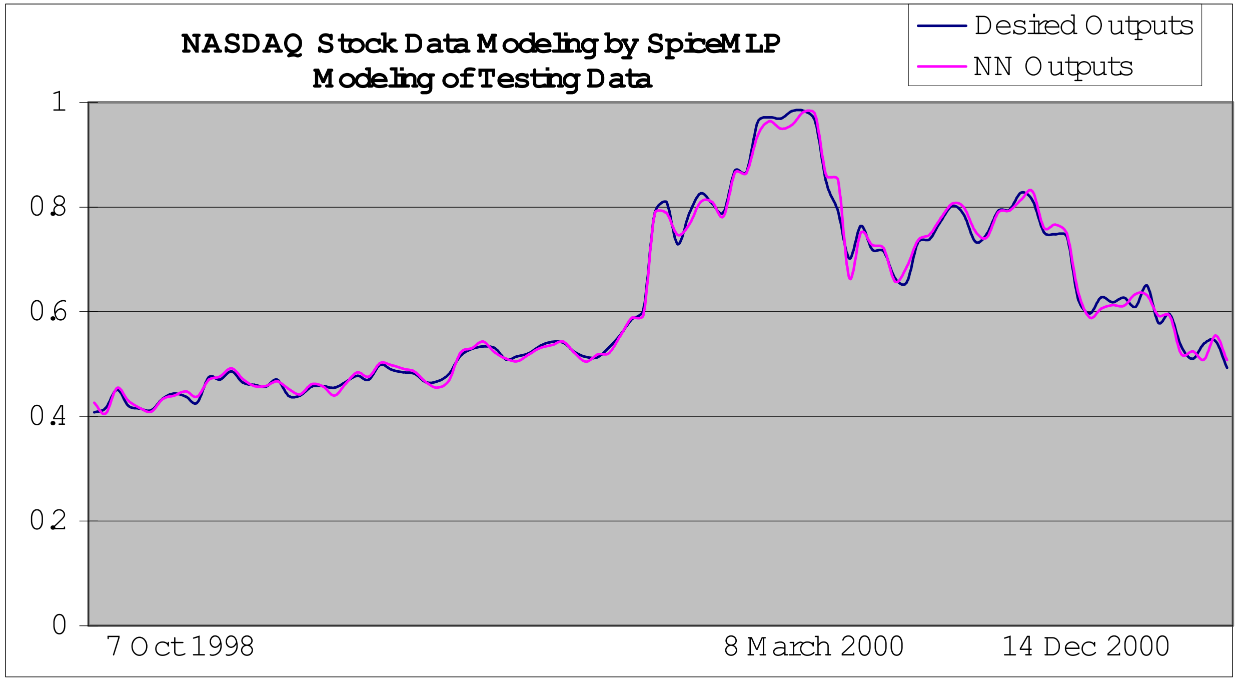


Fig. 22.b. Outputs of Testing Data (NASDAQ Stock prices)

7. Current Exchange Prediction

Suppose you want to predict currency exchange o Canada Dollar/US Dollar and Canada Dollar/Japanese Yen. You can download its data at http://www.bankofcanada.ca/rates/exchange/10-year-lookup/

The currency exchange of Canada Dollar/US Dollar and Canada Dollar/Japanese Yen downloaded from above link would have the following form.

USD->CAD	CAD->USD	Date	JPY->CAD	CAD->JPY	Date
1.5657	0.6387	2001/10/12	0.012948	77.232005	2001/10/12
1.5579	0.6419	2001/10/15	0.012883	77.621672	2001/10/15
1.5619	0.6402	2001/10/16	0.01288	77.639752	2001/10/16
…	…	…	…	…	…
1.5981	0.6257	2001/11/8	0.01331	75.13148	2001/11/8
1.6021	0.6242	2001/11/9	0.013325	75.046904	2001/11/9
Bank holiday	Bank holiday	2001/11/12	Bank holiday	Bank holiday	2001/11/12
1.5981	0.6257	2001/11/13	0.013158	75.999392	2001/11/13
1.5916	0.6283	2001/11/14	0.013082	76.440911	2001/11/14
1.5868	0.6302	2001/11/15	0.012961	77.154541	2001/11/15
…	…	…	…	…	…

For example you use two data: CAD->USD and CAD->JPY, and you use data of 30 past days to predict currency exhange of the next 5th day. That means you will use data of Today-30, Today-29, ..., Today-1,Today to predict currentcy exchange of Today+5. So you will have 60 inputs (30 inputs for CAD->USD and 30 inputs for CAD->JPY), 2 outputs for CAD->USD and CAD->JPY. The following example uses data of the 15 past days to predict data of the next 5th day, you will have 30 inputs and 2 outputs.

With the above data, you should remove "Bank Holiday" lines and then prepare data as the following table.

	CAD->USD					CAD->JPY					CAD->USD	CAD->JPY	LABEL
ID	Today-14	Today-13	…	Today-1	Today	Today-14	Today-13	…	Today-1	Today	Today+5	Today+5	Today Date
1	0.6387	0.6419		0.6302	0.6285	77.232005	77.621672		77.23797	76.581406	0.6257	75.13148	2001/11/1
2	…	…	…	…	…	…	…	…	…	…	…	…	…

You may notice that CAD->USD has values on [0.6199,1.0905] while CAD->JPY has values on [68.918, 123.885]. Because these two ranges of values are much different, if you normalize all data once, CAD->USD will have very small values and CAD->JPY will have large values. This will lead MLP leant with poor generalization.

To have MLP trained with good generalization, it's better to normalize CAD->USD and CAD->JPY seperately. On "Data" folder of the Spice-MLP, there are data without normalization (CAD_USD_JPN_2489_data_30inputs_2outputs.csv) and data that already normalized seperately by Linear method (CAD_USD_JPN_Normalized_2489_data_30inputs_2outputs.csv). The data has 2489 currency exhange datasets with 30 inputs, 2 outputs, from 2001/11/1 to 2011/10/3.

Randomly select 70% of data (1742 datasets) for training and 30% of data (747 datasets) for testing data. Let the MLP lean, we have a training information such as following:

```
Activated Function for Hidden Layer: HyperTanh

Activated Function for Output Layer: Linear

Final Learning rate: 0.003811921

Final MSE of Training Set: 0.0004247656

Final MSE of Testing Set: 0.0004671731

Number of trained data: 1742

Number of tested data: 747

Taken iterations: 1000
```

Save trained data to csv file and draw graphs, we have graphs of outputs of MLP as illustrated in Fig. 23.

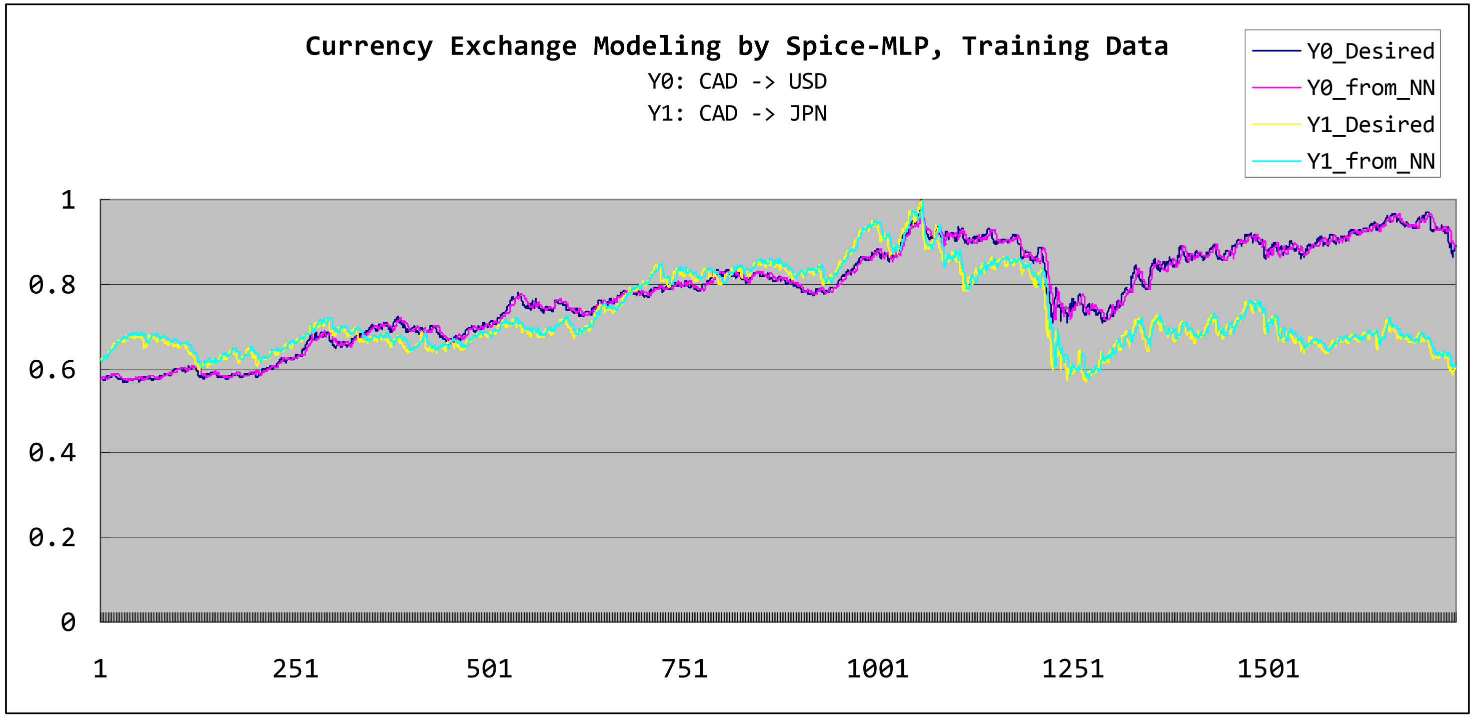


Fig. 23.a. Outputs of Training Data (CAD->USD and CAD->JPY)

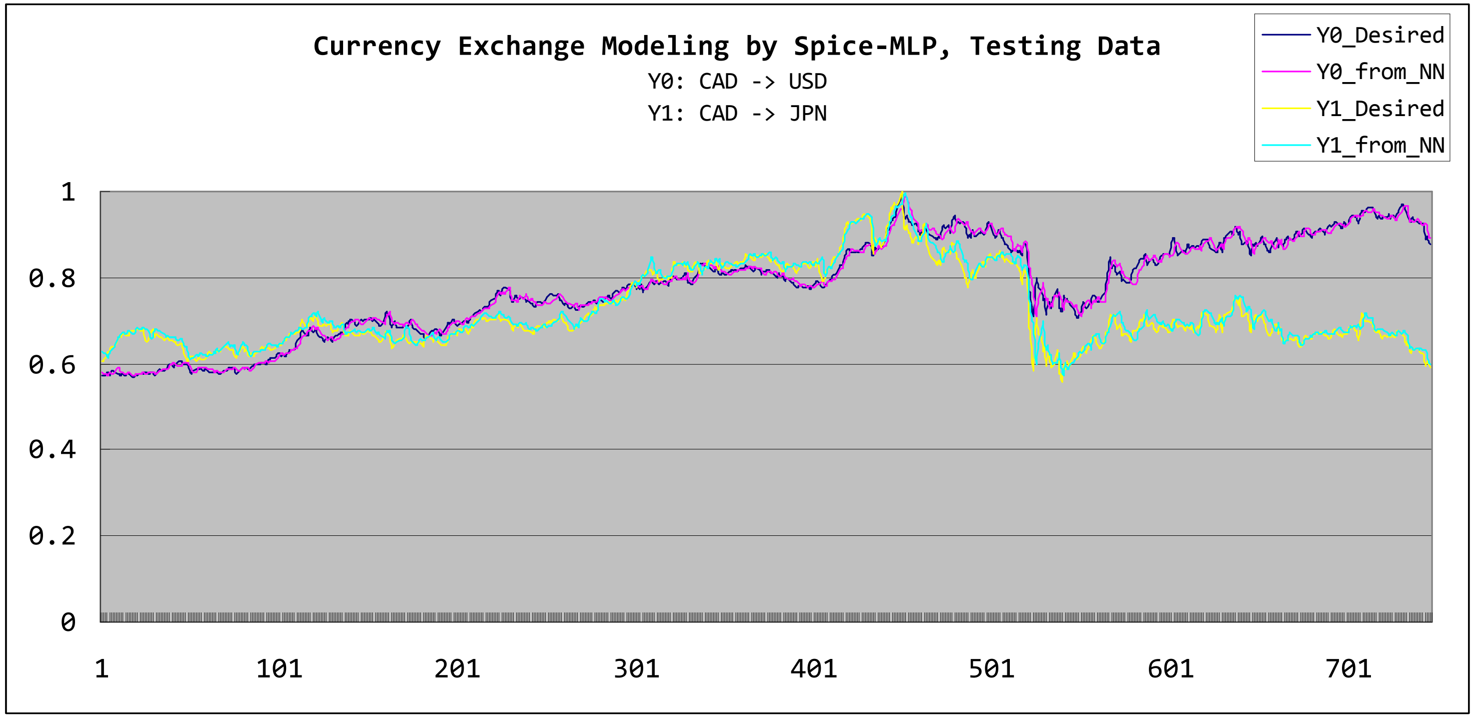


Fig. 23.b. Outputs of Testing Data (CAD->USD and CAD->JPY)

From graphs on Fig. 23, we can see that our MLP lean well with both CAD->USD and CAD->JPY data. However there are some points outputs of MLP and desired outputs are slightly different.

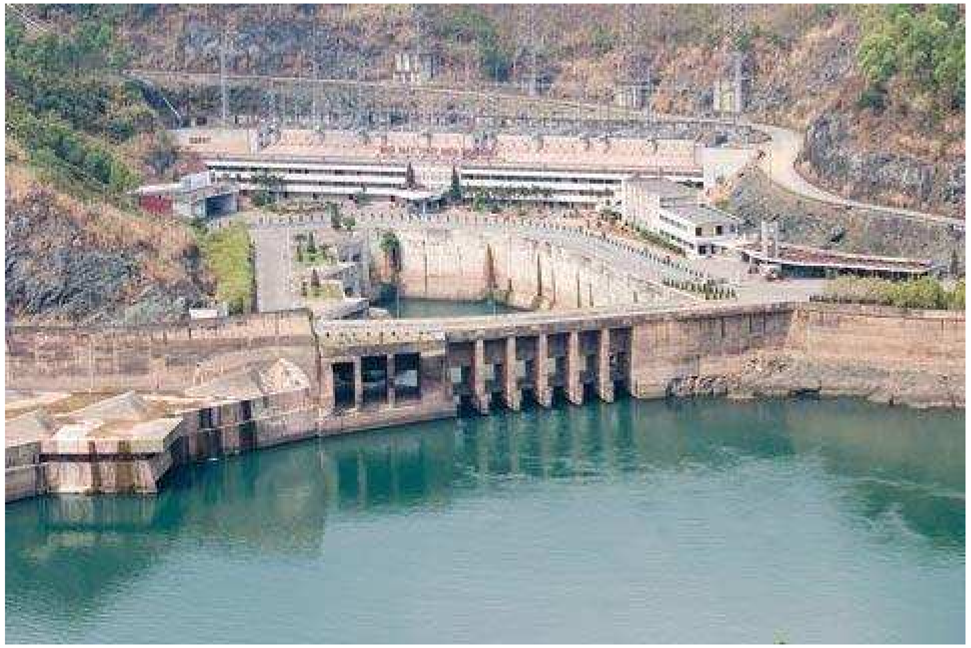
Here are some questions for your reader:

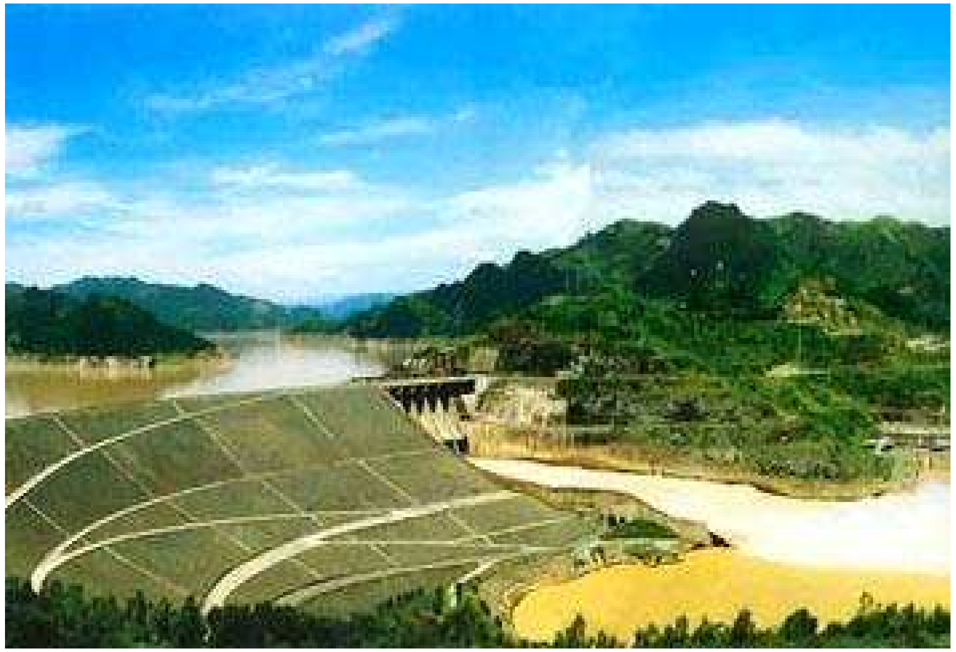
- ✧ Adjust data and MLP parameters so that the MLP learn with an error and accuracy that your desired.
- ✧ The above example use data of 15 past days to predict data of the next 5th day. If we want to use data of the past 60 days to predict data of the next 10th and 15th days, how we should do?

8. Forecast Discharge Hydrograph of Hoabinh lake, Hoabinh Hydroelectric Power plant

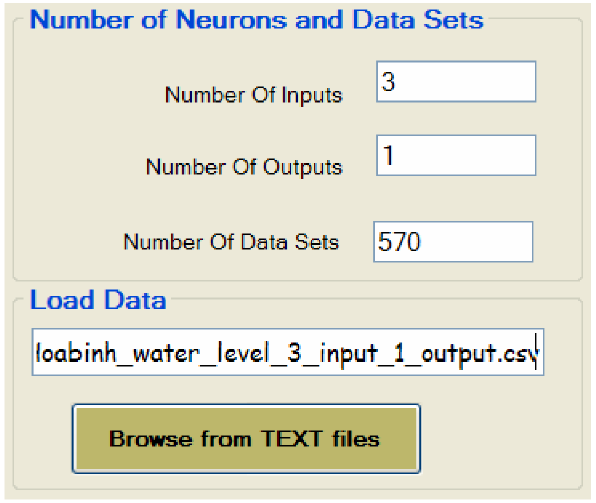
Base on past discharge hydrograph of a lake, we can forecast its discharge hydrograph in near future. In "Data" folder of Spice-MLP there is file named "Hoabinh_water_level_3_input_1_output.csv", this file contains data of discharge hydrographs of the lake of Hoabinh hydroelectric power plan. The data has 3 inputs: today-discharge Q(t), pasth 10th day-discharge Q(t-10), past 20th day-discharge Q(t-20) and the next 10th day-discharge. Number of data is 570 in that there are 480 training data (line 2 to line 481) and 90 testing data (line 482 to line 571). This data is provided by MS. Pham Thi Hoang Nhung, Hanoi Water Resource University.

Here is short descripton about Hoabinh dam from wiki and its photo from internet: "The Hoabinh dam is located in Hòa Bình of the Hoa Binh Province, in Vietnam. It measures 128 m (420 ft) in height, and 970 m (3,182 ft) in length. The power plan produces up to 8,160 GWh of power annually. Construction on the rockfill dam began in November 1979 and was completed in 1994".



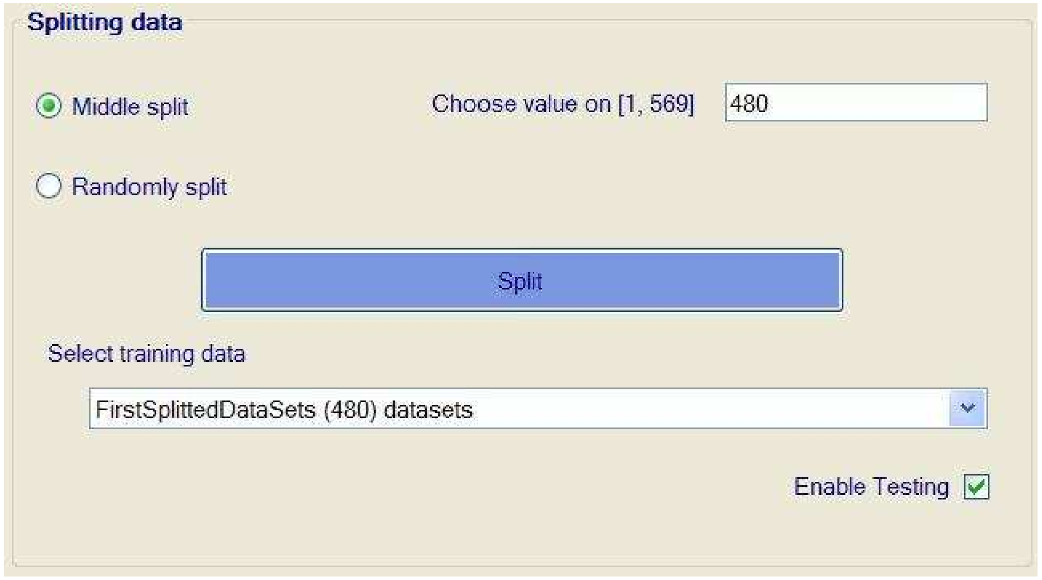


With the data "Hoabinh_water_level_3_input_1_output.csv", you can load data as illustrated in Fig.24, normalize data with Linear method, split training and testing data as shown in Fig.25.



Number Of Inputs: 3
Number Of Outputs: 1
Number Of Data Sets: 570
loabinh_water_level_3_input_1_output.csv

Fig. 24. Load Data with Hoabinh_water_level_3_input_1_output.csv


Splitting data

(● Middle split) Choose value on [1, 569] 480

○ Randomly split

Select training data: FirstSplittedDataSets (480) datasets

Enable Testing ✓

Fig. 25. Split Data for training and testing

Select activated functions for hidden and output layers, train the MLP for some iterations, save modeling data on a csv file, draw output graphs, we will have graphs of training and testing data as shown in Fig.26 and Fig.27.

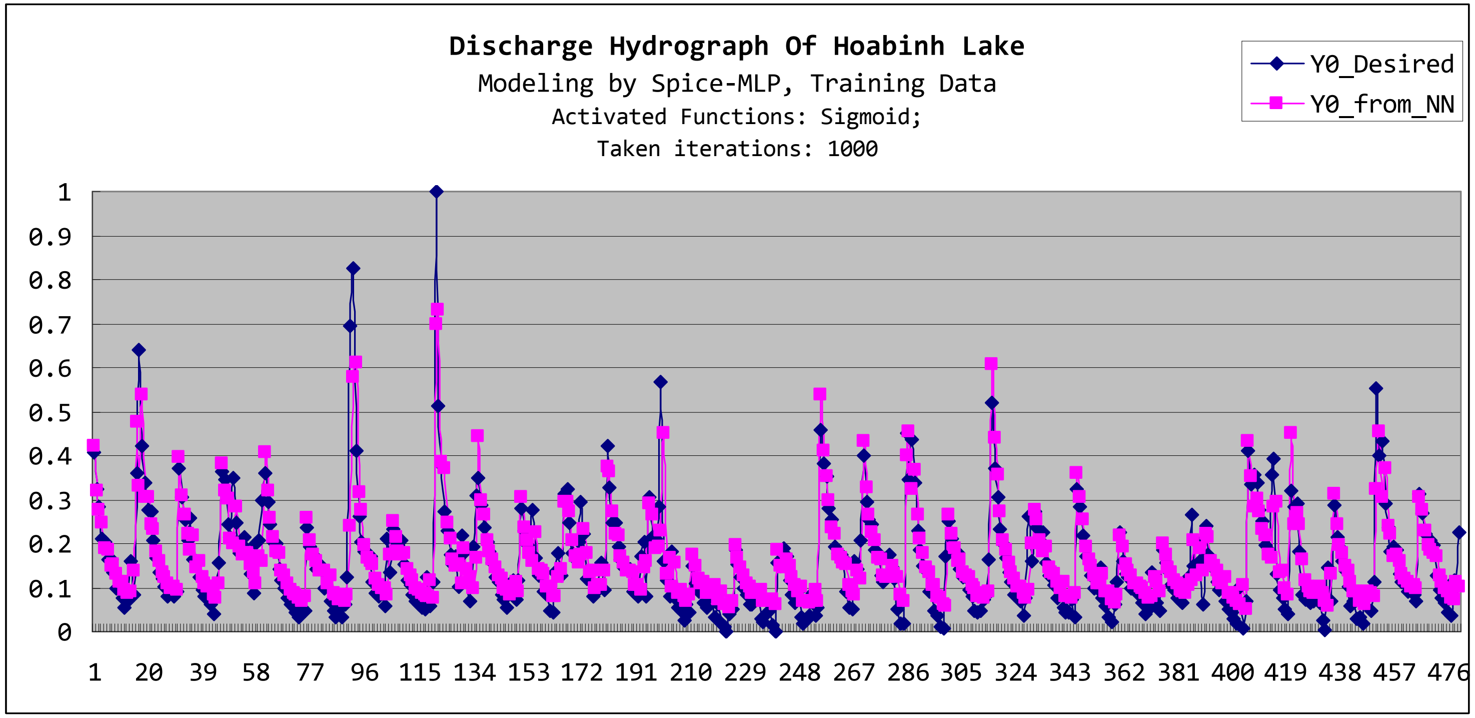


Fig. 26. Discharge Hydrograph Of Hoabinh Lake, Training Data

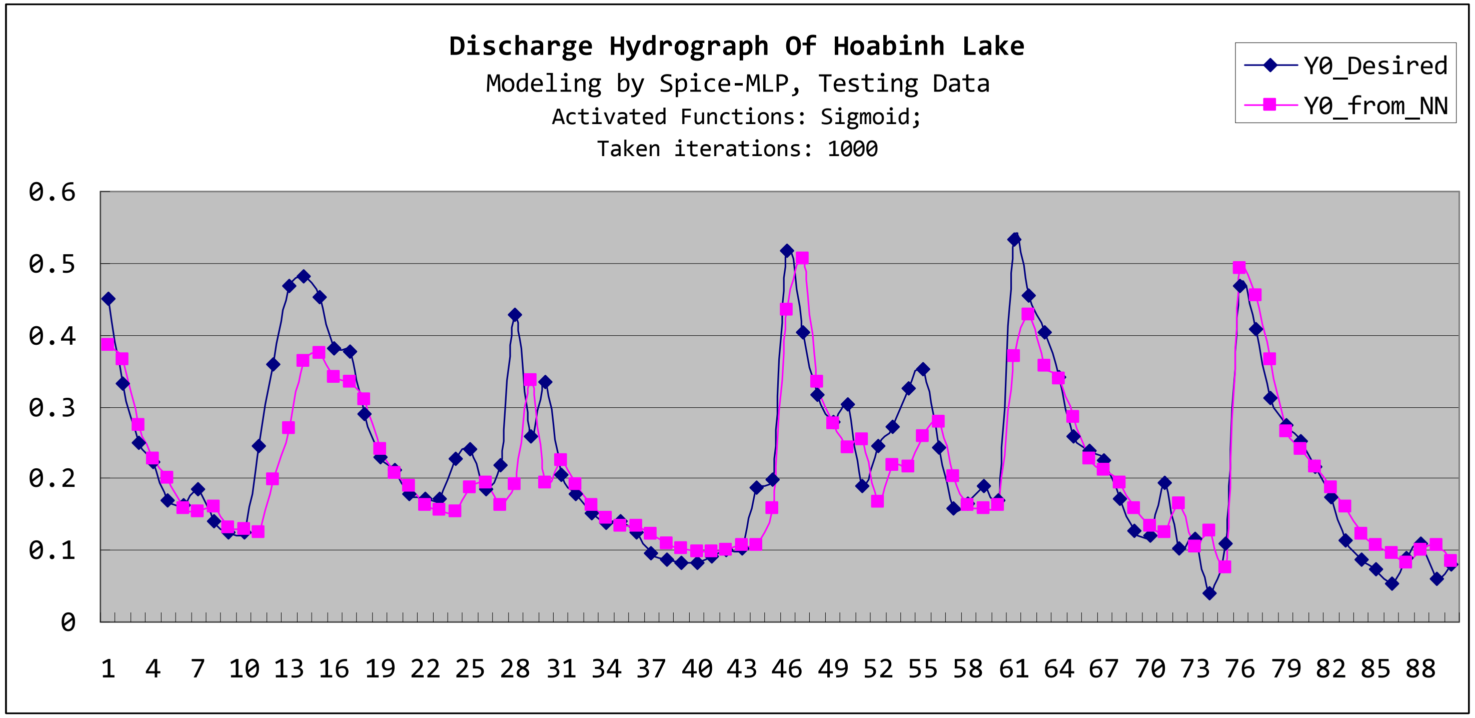


Fig. 27. Discharge Hydrograph Of Hoabinh Lake, Testing Data

We can notice that the MLP leant well, however there are some points that output of MLP and desired output have a little different. How to reduce these different, that is a question for you - your readers.

```
SPICE-NEURO by Cao Thang 2004-2011
Last trained information;
 Activated Function for Hidden Layer: Sigmoid;
 Activated Function for Output Layer: Sigmoid;
 Final Learning rate: 0.03572268;
 Final MSE of Training Set: 0.003680485;
 Final MSE of Testing Set: 0.003794955;
 Number of trained data: 480;
 Number of tested data: 90;
 Taken iterations: 1000;
```

Fig. 28. Discharge Hydrograph Of Hoabinh Lake modeling by Spice-MLP, Training Information

9. Vietnamese Aodai Classification

Fig. 29. shows two output maps of Vietnamese Aodai in two learning by Spice-SOM.

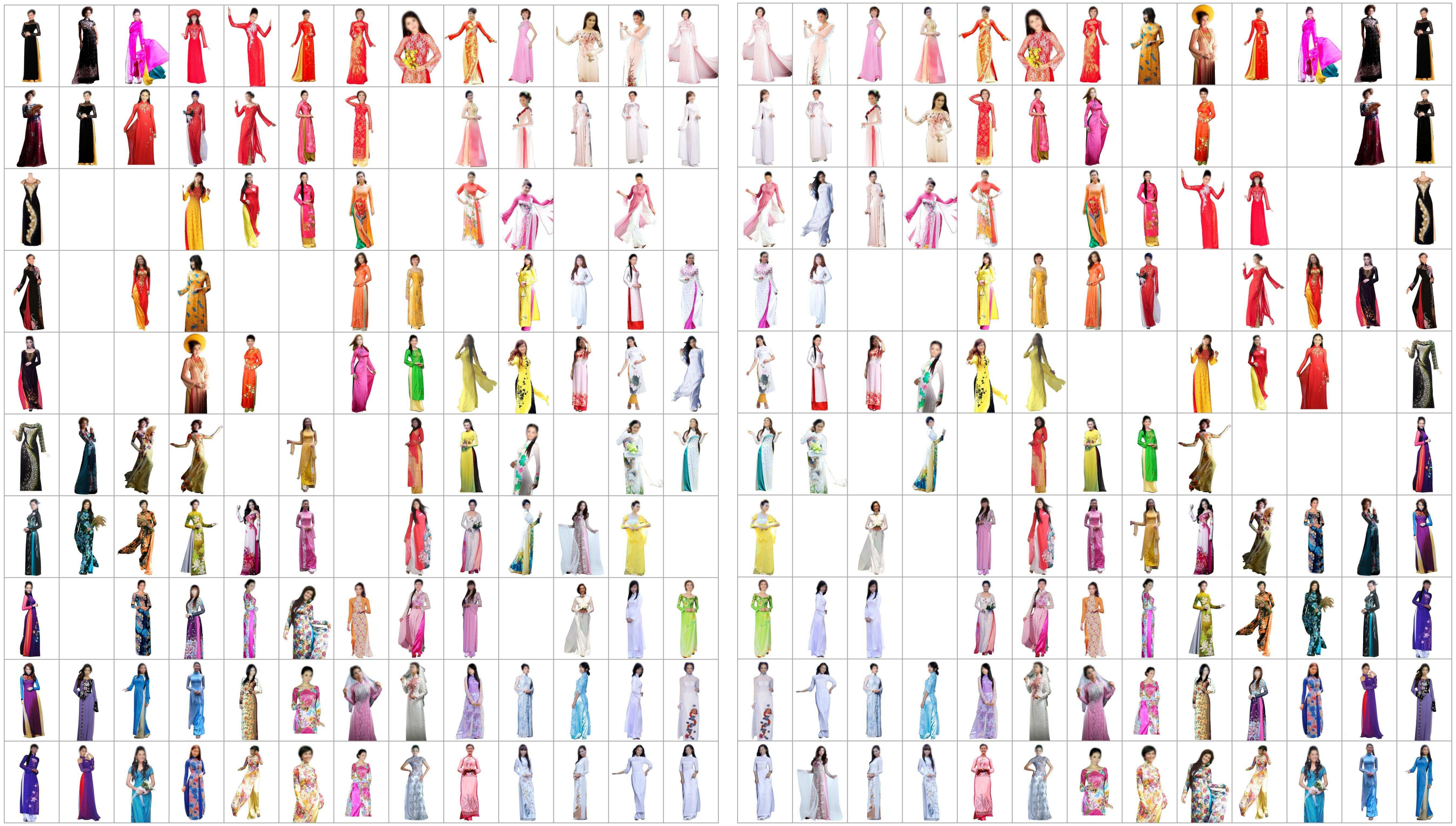


Fig. 29. Vietnamese Aodai grouped by SOM in two learning

10. Conclusion

This material guides how to use and apply NN into practical applications. The author hope that it would be useful for your research.

Thank you for your reading. Good luck for your health, study, job and enjoy everything you like.