Superresolution of multi-frequency signals using multilayer neural network supervised by backpropagation algorithm

SUPERRESOLUTION OF MULTI-FREQUENCY SIGNALS USING MULTILAYER NEURAL NETWORK SUPERVISED BY BACKPROPAGATION ALGORITHM

Kazuyuki HARA+ **Kenji NAKAYAMA**++

+ Graduate School of Nat. Sci. & Tech., ++ Faculty of Tech., Kanazawa Univ.
2-40-20, Kodatsuno, Kanazawa, 920 JAPAN
E-mail: hara@haspnnl.ec.t.kanazawa-u.ac.jp
nakayama@haspnnl.ec.t.kanazawa-u.ac.jp

ABSTRACT Multi-frequency signal classification is discussed using multilayer neural networks supervised by the backpropagation algorithm. Several novel properties of the neural network are provided. First, the neural network can detect the frequencies, which cannot be represented by the discrete Fourier transform (DFT). Second, the neural network can be realized with real coefficients. DFT, however, requires the complex coefficients $\exp(\pm j\omega nT/N)$. Finally, the number of the inner products of the connection weights and the input signal samples is sufficiently smaller than that of the output samples, required in digital filters with real coefficients.

I INTRODUCTION

Advantage of multilayer neural networks supervised by the backpropagation (BP) algorithm is to extract common properties, features or rules, which can be used to classify data included in several groups [1]. Especially, when it is difficult to analyze the common features using conventional methods, the supervised learning, using the known input and output data, becomes very useful. This application field includes, for instance, pronunciation of English text, speech recognition, image compression, sonar target analysis, stuck market prediction and so on [2]-[6].

In this paper, classification performance of the neural networks is discussed based on frequency analysis. Multi-frequency signals are employed for this purpose. Since the number of the input units in the neural network is finite, sampled data are taken into account. In order to analyze frequency components of the discrete signals, discrete Fourier transform (DFT) is usually used. In this case, the sampling points on the frequency axis discretely locate. Therefore, the representable frequencies are limited by the sampling frequency and the number of samples.

II MULTI-FREQUENCY SIGNALS

Multi-frequency signals are defined by

$$x_{pm}(n) = \sum_{r=1}^{R} A_{mr}\sin(\omega_{pr}nT+\phi_{mr}), \quad n=1\sim N, \quad \omega_{pr}=2\pi f_{pr} \tag{1}$$

M samples of $x_{pm}(n)$, $m=1\sim M$, are included in the pth group X_p as follows:

$$X_p = \{x_{pm}(n),\ m=1\sim M\}, \quad p=1\sim P \tag{2}$$

P signal groups, X_p, $p=1\sim P$, are assumed.

T is a sampling period. The signals have N samples. In the same group, the same frequencies are used.

$$F_p = [f_{p1}, f_{p2}, \ldots, f_{pR}] \text{ Hz}, \quad p=1\sim P \tag{3}$$

Amplitude A_{mr} and phase ϕ_{mr} are different for each frequency. They are generated as random numbers, uniformly distributed in the following ranges.

$$0 < A_{mr} \leqq 1 \tag{4a}$$
$$0 \leqq \phi_{mr} < 2\pi \tag{4b}$$

III MULTILAYER NEURAL NETWORK

3.1 Network Structure

A two-layer neural network is taken into account. N samples of the signal $x_{pm}(n)$ are applied to the input layer in parallel. The nth input unit receives the sample at nT. Connection weight from the nth input unit to the jth hidden unit is denoted by w_{nj}. The input of the jth hidden unit is given by

$$net_j = \sum_{n=0}^{N-1} w_{nj} x_{pm}(n) \tag{5}$$

The input net_j is transferred through the following sigmoid function.

$$y_j = f(net_j) = \frac{1}{1 + e^{-(net_j + \theta_j)}} \tag{6}$$

Letting connection weight from the jth hidden unit to the kth output unit be w_{jk}, the input of the kth output unit is given by

$$net_k = \sum_{j=1}^{J} w_{jk} y_j \tag{7}$$

Furthermore, the final output is obtained by

$$y_k = f(net_k) \tag{8}$$

The number of output units is equal to that of the signal groups P. The neural network is trained so that a single output unit responds to one of the signal groups.

3.2 Training and Classification

The set of signals is categorized into training and untraining data sets, X_{Tp} and X_{Up}, respectively. Their elements are expressed by $x_{Tpm}(n)$ and $x_{Upm}(n)$.

$$X_p = [X_{Tp}, X_{Up}] \tag{9}$$
$$X_{Tp} = \{x_{Tpm}(n),\ m=1\sim M_T\} \tag{10a}$$
$$X_{Up} = \{x_{Upm}(n),\ m=1\sim M_U\} \tag{10b}$$

The neural network is trained by using $x_{Tpm}(n)$, $m=1\sim M_T$, for the pth group. After the training is completed, the untraining signals $x_{Upm}(n)$ are applied to the neural network, and the output is calculated following Eqs.(5)-(8). For the input signal $x_{Upm}(n)$, if the pth output y_p has the maximum value, then the signal is exactly classified. Otherwise, the network fails in classification.

IV SIMULATION OF MULTI-FREQUENCY SIGNAL CLASSIFICATION

4.1 Two-Groups with Three Frequencies

(Case-1.1) Alternate Grouping of Frequencies

Two frequency sets are determined as follows:

$$F_1 = [1, 2, 3] \quad \text{Hz}$$
$$F_2 = [1.5, 2.5, 3.5] \quad \text{Hz}$$

The sampling frequency is chosen to be 10 Hz, that is T=0.1 sec. The number of samples is N=10. Therefore, the signals are sampled in the range $0 \leq nT < 1$ sec, and n=0 ~ 9. The training signal sets for each group include 200 signals, that is $x_{Tpm}(n)$, m=1 ~ 200 for p=1 and 2. 3600 signals are used for the untrained signal sets, that is $x_{Upm}(n)$, m=1 ~ 1800 for p=1 and 2.

Figure 1 shows examples of the signals. The frequencies, included in $x_{1m}(n)$ and $x_{2m}(n)$, are located alternately. The observation interval and the number of samples are limited. Therefore, it is very difficult to distinguish $x_{1m}(n)$ and $x_{2m}(n)$, based on their waveform.

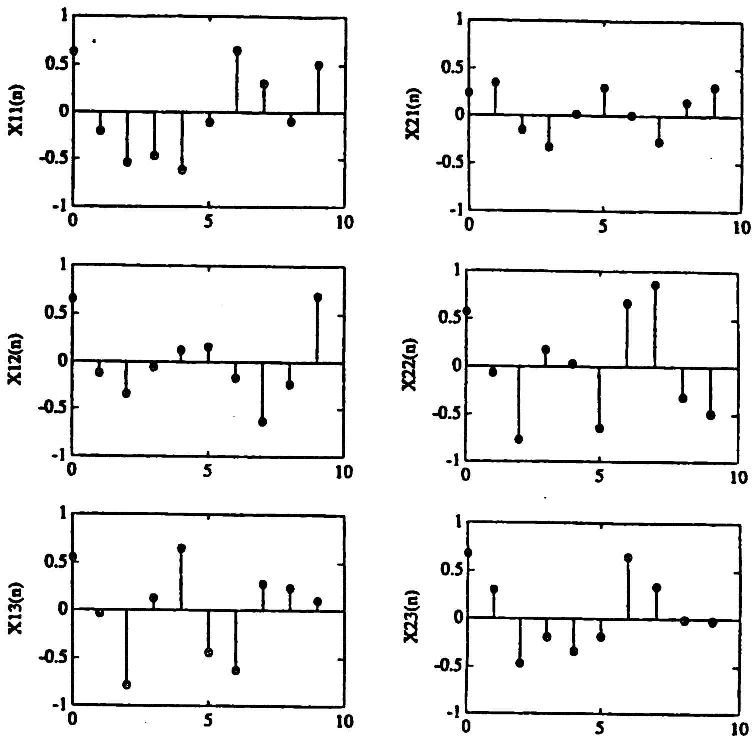

Fig.1 Examples of multi-frequency signals $x_{1m}(n)$ and $x_{2m}(n)$ in Case-1.1.

The training by BP algorithm can converge with a single hidden unit, that is a single-layer network. Figure 2 illustrates the connection weights from the input units to the hidden unit.

Since the training converged, 100% of the training signals $x_{Tpm}(n)$ were successfully classified. For the untraining signals, the classification rate is 99.5 %. Thus, highly exact classification can be achieved.

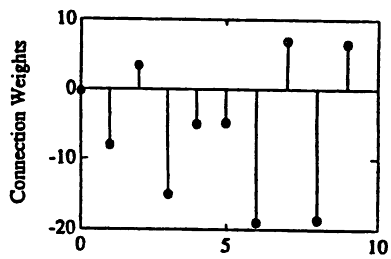

Fig.2 Connection weights from input layer to hidden layer in Case-1.1.

(Case-1.2) Similar Frequencies

The following similar frequencies are employed.

$$F_1 = [1, 2, 3] \quad \text{Hz}$$
$$F_2 = [1.1, 2.1, 3.1] \quad \text{Hz}$$

Since the frequencies in both sets are very close to each other, it is more difficult to distinguish the signals based on their waveform. The BP algorithm can converge, and the accuracy for the untraining signals is still 99.5%.

4.2 Two-Groups with Five Frequencies

(Case-2.1) Training Signals with 10 Samples

The frequencies are chosen to be

$$F_1 = [1, 1.5, 2, 2.5, 3] \quad \text{Hz}$$
$$F_2 = [1.25, 1.75, 2.25, 2.75, 3.25] \quad \text{Hz}$$

The number of samples is N=10, and the sampling period is T=0.1 sec. The observation interval is $0 \leqq nT < 1$ sec, and $n=0 \sim 9$. The training did not completely converge. Percentages of exact classification are 75.3% and 81.5% for $x_{T1m}(n)$ and $x_{T2m}(n)$, respectively. For the untraining signals $x_{U1m}(n)$ and $x_{U2m}(n)$, the accuracies are a little decreased to 71.8% and 79%, respectively.

(Case-2.2) Training Signals with 15 Samples

The frequencies are chosen to be

$$F_1 = [1, 2, 3, 4, 5] \quad \text{Hz}$$
$$F_2 = [1.5, 2.5, 3.5, 4.5, 5.5] \quad \text{Hz}$$

The number of samples is N=15, using T=1/15 sec. Therefore, the sampling points locate in $0 \leqq nT < 1$ sec, and $n=0 \sim 14$.

The training can converge using a single hidden unit. Thus, 100% classification is possible for $x_{T1m}(n)$ and $x_{T2m}(n)$. Furthermore, 99.9% of the untraining signals can be classified. Comparison between Cases-2.1 and 2.2 will be discussed in Sec. V.

4.3 Three or Five Frequency Sets

(Case-3.1) Three Frequency Sets

The following frequency sets are used.

$$F_1 = [1, 2, 3] \quad \text{Hz}$$
$$F_2 = [1.33, 2.33, 3.33] \quad \text{Hz}$$
$$F_3 = [1.67, 2.67, 3.67] \quad \text{Hz}$$

The number of samples is N=10, and the sampling period is T=0.1 sec. X_{T1} and X_{T2} have 200 signals each. Totally, 5400 signals are used for the untraining data sets X_{U1} and X_{U2}.

The number of the hidden units is varied from 1 to 4. In any case, the training did not completely converge. The classification accuracies are 91.2%, 98.0% and 99.5% for 1, 2 and 4 hidden units, respectively. For the untraining data, the accuracies are 90.1%, 97.0% and 98.6%, respectively. More hidden units did not improve the classification rates.

(Case-3.2) Five Frequency Sets with 10 Samples

The following frequency sets are chosen.

$$F_1 = [1, 2, 3] \quad \text{Hz}$$
$$F_2 = [1.2, 2.2, 3.2] \quad \text{Hz}$$
$$F_3 = [1.4, 2.4, 3.4] \quad \text{Hz}$$

F_4 = [1.6, 2.6, 3.6] Hz
F_5 = [1.8, 2.8, 3.8] Hz

The sampling period is T=0.1 sec, and the number of samples is N=10. 200 training signals are used for each group, that is $x_{Tpm}(n)$, m=1～200 for p=1,2,3,4 and 5. 1800 untraining signals are examined for each group, that is $x_{Upm}(n)$, m=1～1800 for p=1,2,3,4 and 5.

The training did not completely converge. The number of hidden units was varied from 1 to 8. Percentages of the exact classification, with 3 and 8 hidden units, are listed in Table 1. By increasing hidden units, the classification rates can be improved. 5～8 hidden units are appropriate.

Table 1 Classification rates [%] in Cases-3.2 and 3.3.

CASE		Three hidden units		Eight hidden units	
		X_{Tp}	X_{Up}	X_{Tp}	X_{Up}
3.2	F_1	98.0	96.7	98.0	96.3
	F_2	73.0	70.9	79.5	75.3
	F_3	65.0	71.6	98.5	98.1
	F_4	68.0	73.6	89.0	87.2
	F_5	77.0	73.7	98.0	97.8
	Mean	76.2	77.3	92.6	90.9
3.3	F_1	99.5	98.0	99.5	98.4
	F_2	82.5	84.3	99.5	98.5
	F_3	91.5	92.2	99.0	99.1
	F_4	94.0	90.3	99.0	97.2
	F_5	99.5	99.1	100.0	99.6
	Mean	93.4	92.8	99.4	98.6

(Case-3.3) Five frequency Sets with 20 Samples

The following frequency sets are taken into account.

F_1 = [1, 4, 7] Hz
F_2 = [1.5, 4.5, 7.5] Hz
F_3 = [2, 5, 8] Hz
F_4 = [2.5, 5.5, 8.5] Hz
F_5 = [3, 6, 9] Hz

The sampling period is 0.05 sec, and the number of samples is 20. Thus, the sampling points locate in $0 \leq nT < 1$ sec, and n=0～19.

The classification rates are also listed in Table 1, which are improved from Case-3.2. Comparison between Cases-3.2 and 3.3 will be discussed in Sec. V.

4.4 Estimation Based on Euclidean Distance

Similarity between the training and untraining signals is evaluated using Euclidean distance as follows:.

$$D_{pq}(m,m') = \left\{\frac{1}{N}\sum_{n=1}^{N}(x_{Tpm}(n) - x_{Tqm'}(n))^2\right\}^{1/2} \qquad (11)$$

Table 2 shows the above Euclidean distance. a_i and c_i indicate some of $x_{U1m}(n)$ and $x_{U2m}(n)$, exactly classified, respectively. b_i and d_i correspond to some of $x_{U1m}(n)$ and $x_{U2m}(n)$, not exactly classified, respectively. The signals, having the minimum distance to the above signals, were searched for from X_{T1} and X_{T2}. As a result, Table 2 shows the minimum Euclidean distance between a_i, b_i, c_i, d_i and some of $x_{Tpm}(n)$.

Table 2 Minimum Euclidean distance in Case-1.1.

Signals $X_{Upm}(n)$		Minimum Distance X_{T1}	X_{T2}
X_{U1}	a_1	0.167	0.184
	a_2	0.127	0.182
	a_3	0.167	0.184
	a_4	0.148	0.110
	a_5	0.118	0.114
	b_1	0.114	0.257
	b_2	0.114	0.084
	b_3	0.130	0.190
X_{U2}	c_1	0.123	0.078
	c_2	0.130	0.089
	c_3	0.219	0.170
	c_4	0.100	0.130
	c_5	0.118	0.123
	d_1	0.167	0.138
	d_2	0.192	0.228
	d_3	0.105	0.110

The Euclidean distance between $x_{Upm}(n)$ and $x_{Tpm}(n)$ in the same group is not always smaller than that between $x_{Upm}(n)$ and $x_{Tqm}(n)$, $p \neq q$, in the different groups. Furthermore, the minimum distances for a_i, c_i and b_i, d_i are almost the same. These results indicate that the multi-frequency signals $x_{1m}(n)$ and $x_{2m}(n)$ cannot be distinguished based on the Euclidean distance.

V COMPARISONS BETWEEN NEURAL NETWORK AND DFT

5.1 Observable Frequencies by DFT

Generally speaking, multi-frequency signals can be analyzed by Fourier analysis. For the sampled data, DFT is used [7].

As defined in Sec.2.1, the sampling frequency is given by $fs = 1/T$. The sampling points on the frequency axis, that is representable frequency points, are given by ifs/N Hz, $i=0,1,2,\ldots$, which satisfy $0 \leq ifs/N < fs/2$ Hz.

5.2 Limitation of Frequency Detection by DFT

Table 3 summarized the observable frequencies, signal frequencies, the number of hidden units and percentages of exact classification for each case. From this table, all frequencies cannot be represented by DFT. This means that the classification problems discussed in this paper, cannot be inherently solved by DFT. Some examples to demonstrates this fact are shown in the following.

Table 3 Classification rates [%] for each case, with the suitable number of hidden units.

CASE	Observable Frequencies	Frequency Sets	Hidden Units	Accuracy[%] X_{Tp}	Accuracy[%] X_{Up}
1.1	0,1,2,3,4	$F_1 = [1,2,3]$	1	100	99.8
		$F_2 = [1.5,2.5,3.5]$		100	94.5
1.2	0,1,2,3,4	$F_1 = [1,2,3]$	1	100	99.7
		$F_2 = [1.1,2.1,3.1]$		100	99.9
2.1	0,1,2,3,4	$F_1 = [1,1.5,2,2.5,3]$	5	82.8	77.2
		$F_2 = [1.25,1.75,2.25,2.75,3.25]$		86.8	82.3
2.2	0,1,2,3,4 5,6,7	$F_1 = [1,2,3,4,5]$	1	100	99.9
		$F_2 = [1.5,2.5,3.5,4.5,5.5]$		100	99.9
3.1	0,1,2,3,4	$F_1 = [1,2,3]$	4	99.0	93.5
		$F_2 = [1.33,2.33,3.33]$		99.5	98.8
		$F_3 = [1.67,2.67,3.67]$		100	99.6
3.2	0,1,2,3,4	$F_1 = [1,2,3]$	8	98.0	96.3
		$F_2 = [1.2,2.2,3.2]$		79.5	75.3
		$F_3 = [1.4,2.4,3.4]$		98.5	98.1
		$F_4 = [1.6,2.6,3.6]$		89.0	87.2
		$F_5 = [1.8,2.8,3.8]$		98.0	97.8
3.3	0,1,2,3,4, 5,6,7,8,9	$F_1 = [1,4,7]$	8	99.5	98.4
		$F_2 = [1.5,4.5,7.5]$		99.5	98.5
		$F_3 = [2,5,8]$		99.0	99.1
		$F_4 = [2.5,5.5,8.5]$		99.0	97.2
		$F_5 = [3,6,9]$		100	99.6

Figure 3 shows examples for amplitude responses of $x_{Tpm}(n)$, p=1(o), 2(*), 3(x), in Case-3.1. $x_{T1m}(n)$ can be roughly recognized, because its frequencies are observable. On the contrary, $x_{T2m}(n)$ and $x_{T3m}(n)$ cannot be distinguished due to lack of their frequencies.

Furthermore, in Case-1.2, the signals $x_{U1m}(n)$ are regenerated by setting their amplitude for each frequency to 50% of that of $x_{U2m}(n)$, as follows:

$$x_{U1m}(n) = 0.5\sum_{r=1}^{R} A_{2mr}\sin(\omega_{1r}nT + \phi_{1mr}) \tag{12a}$$

$$x_{U2m}(n) = \sum_{r=1}^{R} A_{2mr}\sin(\omega_{2r}nT + \phi_{2mr}) \tag{12b}$$

Figure 4 shows the amplitude responses for both signals obtained by DFT. From this result, $x_{U2m}(n)$ may be classified into the first group X_1, because its

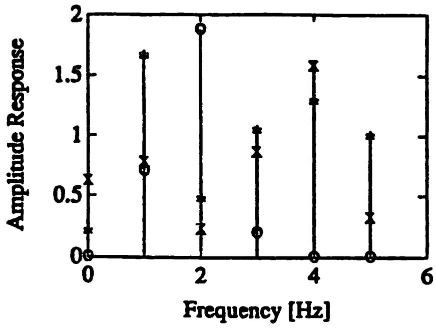

Fig.3 Amplitude responses for $x_{Tpm}(n)$, p=1, 2, 3, denoted with o, *, x, respectively.

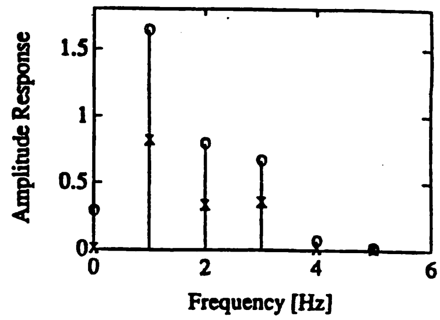

Fig.4 Amplitude responses for $x_{Upm}(n)$, p=1, 2, defined by Eqs.(12a) and (12b), denoted by x and o, respectively.

frequency components at 1, 2 and 3 Hz are greater than those of $x_{U1m}(n)$. On the contrary, the neural network can distinguish these signals with 99.5% accuracy.

5.3 Performance of Neural Network

As discussed in the previous sections, the neural network can classify the multi-frequency signals, whose frequencies are not completely represented by DFT. The classification performance of the neural network is also depend on what percentage of the frequencies can be observable as shown in Table 3.

In Case-2.1, 1.5, 2.5 Hz in F_1 and all frequencies in F_2 cannot be represented by DFT. As a result, the percentages of classification is not so high. On the other hand, in Case-2.2, all frequencies in F_1 can be represented by DFT. Therefore, the accuracies are drastically improved. In Case-3.2, F_1 is only representable. On the other hand, in Case-3.3, F_1, F_3 and F_6 can be represented. Therefore, the classification rates are improved.

Although performance of the neural network still depends on the observable frequencies, it is not necessary to represent all frequencies. About 30~50% of the frequencies are sufficient to achieve high classification rates. This point should be noted as essential difference between the neural network and DFT.

The number of hidden units, required to obtain high accuracy, is proportional to the number of the signal groups. That could be expected following the discussions in [8].

5.4 Design Problem of Real Coefficients

There is another typical difference between the neural network and DFT. The latter needs complex coefficients $\exp(\pm j\omega nT/N)$. On the other hand, the neural network can be realized with real coefficients. One example is shown here.

In Case-1.1, the number of samples is increased to 20, while the same sampling period T=0.1 sec is used. Therefore, the signals are sampled in the interval $0 \leq nT < 2$ sec, at every 0.1 sec. The frequency points, which can be represented by DFT, become 0, 0.5, 1, ..., 4.5 Hz, which include all frequencies in F_1 and F_2.

After the neural network was trained, the amplitude and phase responses of the connection weights were calculated through DFT. The frequency components in the first group $x_{1m}(n)$ can be emphasized, and those of the second group $x_{2m}(n)$ are suppressed. This amplitude response could be expected based on DFT analysis. However, there is no direction on the phase response. Different phase response was used to generate another connection weights. These weights, however, did not work well. The classification accuracy is about 63%. Because the problem is a choice between two things, 63% is very low accuracy. Thus, the phase response obtained by BP algorithm has significant meaning, which cannot be designed by the conventional methods.

5.5 Filtering Method

Frequency component extraction with real coefficients is also possible using digital filters [7]. The digital filters can continuously sweep the frequency axis. The output signal is obtained through the convolution sum of the input signal and the impulse response of the digital filter. The amplitude response of the transfer function can be designed so as to amplify one of $x_{pm}(n)$ and to suppress the other. However, in order to remove effects of the phase response, the mean square of y(n) is required over N samples.

The neural network, trained with a single hidden unit, requires only one inner products of the input signal and the connection weights. However, the digital filters cannot detect the frequency components with one output sample.

VI CONCLUSIONS

The multi-frequency signal classification problems have been discussed using the multilayer neural network supervised by BP algorithm. Several novel properties have been provided. First, the neural network can classify the sets of frequencies. Some of them cannot be represented by DFT. Second, the neural network can be realized with real coefficients. DFT needs the complex coefficients $\exp(\pm j\omega nT/N)$. Finally, the number of the inner products, required in the neural network, is sufficiently smaller than that of the output samples, required in the digital filters.

REFERENCES

[1]D.E.Rumelhart and J.L.McClelland et al, Parallel Distributed Processing, MIT Press, 1986.

[2]T.J.Sejnowski and C.R.Rosenberg,"Parallel networks that learn to pronounce English text", Complex Systems, vol.1, pp.145-168, 1987.

[3]J.L.Elman and D.Zipser,"Learning the hidden structure of speech", Tech. Rep. 8701, Univ. of California, San Diego 1987.

[4]G.W.Cottrell, R.Munro and D.Zipser,"Image compression by backpropagation: An example of extentional programming", Tech. Rep. 8702, Univ. of California, San Diego 1987.

[5]R.P.Gorman and T.J.Sejnowski,"Analysis of hidden units in a layered network trained to classify sonar targets", Neural Networks, vol.1, pp.75-90 1988.

[6]T.Kimoto et al.,"Stock market prediction system with modular neural networks", Proc. IJCNN'90, San Diego, pp. I -1-6, 1990.

[7]A.V.Oppenheim and R.W.Shafer, Digital Signal Processing, Prentice-Hall, Inc., 1975.

[8]K.Nakayama, S.Inomata and Y.Takeuchi,"A digital multilayer neural network with limited binary expression", Proc. IJCNN'90, San Diego, pp. II -587-592, 1990.