PAPER

Multi-Frequency Signal Classification by Multilayer Neural Networks and Linear Filter Methods

Kazuyuki HARA[†*] *and* Kenji NAKAYAMA[††], *Members*

SUMMARY This paper compares signal classification performance of multilayer neural networks (MLNNs) and linear filters (LFs). The MLNNs are useful for arbitrary waveform signal classification. On the other hand, LFs are useful for the signals, which are specified with frequency components. In this paper, both methods are compared based on frequency selective performance. The signals to be classified contain several frequency components. Furthermore, effects of the number of the signal samples are investigated. In this case, the frequency information may be lost to some extent. This makes the classification problems difficult. From practical viewpoint, computational complexity is also limited to the same level in both methods. IIR and FIR filters are compared. FIR filters with a direct form can save computations, which is independent of the filter order. IIR filters, on the other hand, cannot provide good signal classification due to their phase distortion, and require a large amount of computations due to their recursive structure. When the number of the input samples is strictly limited, the signal vectors are widely distributed in the multi-dimensional signal space. In this case, signal classification by the LF method cannot provide a good performance. Because, they are designed to extract the frequency components. On the other hand, the MLNN method can form class regions in the signal vector space with high degree of freedom. When the number of the signal samples is not so limited, both the MLNN and LF methods can provide the same high classification rates. In this case, since the signal vectors are distributed in the specific region, the MLNN method has some convergence problem, that is local minimum problem. The initial weights should be carefully determined around the optimum solution. Another point is robustness for noisy signal. The LFs can suppress wide-band noise by using very high-Q filters. However, the MLNN method can be also robust. Rather, it is a little superior to the LF method when the computational load is limited.

key words: *multilayer neural networks, signal classification, FIR filters, IIR filters, frequency selective classification*

1. Introduction

Recently, neural networks (NNs) [1] have been applied to the signal processing fields, including signal detection [2]–[6], digital demodulator [7], [8] and digital signal classification [9], [10]. In these applications, the NN methods can provide better performance. Furthermore, there are many papers comparing multilayer NNs (MLNNs) and statistical methods in the application point of view. For example, pattern classification performance, complexity of structure for implementation and computations have been taken into account in comparison in Tsoi [11], Atlas [12], Gish [13], and Lippmann [14], respectively. From these results, the MLNN method is recognized to be superior to linear filter (LF) methods under some conditions. However, these conditions have not been well discussed.

In this paper, comparison between the MLNN and the LF methods that used in signal classification is discussed. Usually, the MLNN method is seemed to be useful for arbitrary pattern classification. On the other hand, the LF methods are good for detecting the signals specified with frequency components. The purpose of this paper is to investigate usefulness of the MLNN method in the signal processing field, therefore, the latter signals are taken into account. Thus, the signals are classified based on their frequency components.

Furthermore, the observation period is very short. This means the number of the signal samples is set to be very small. Since, in this case, frequency information may be lost to some extent, the signal classification becomes more difficult. This kind of limitations appears in the digital communication, the signal processing, and the real time image processing [13] fields. From practical viewpoint, computational complexity is also limited. Namely, the comparison will be discussed based on length of the signal sequence and complexity of implementation.

Since the MLNN is a non-parametric model, the generalization for untrained data is an important criterion. Furthermore, robustness for noisy signal classification is also compared. Through theoretical and experimental results, we derive the conditions, under which we can estimate which method is useful in frequency selective signal classification.

Classification mechanisms of the MLNN method and the LF methods are explained in the Sects. 2 and 3, respectively. Moreover, in Sect. 4, classification performance as a pattern classifier is analyzed for the MLNN and the LF methods. Learning ability of the MLNN is also discussed in this section. Then, they are compared based on the analysis results. Computer simulations and discussions are given in Sect. 5.

Manuscript received May 30, 1996.
Manuscript revised December 5, 1996.

[†]The author is with Toyama Polytechnic College, Uozu-shi, 937 Japan.
[††]The author is with Department of Electrical and Computer Engineering, Faculty of Engineering, Kanazawa University, Kanazawa-shi, 920 Japan.
[*]Presently, the author is with Graduate School of Natural Science and Technology, Kanazawa University.

2. Signal Classification by Multilayer Neural Network

2.1 Multilayer Neural Networks

A two-layer neural network is considered here. N samples of the signal $\{x(n_0+n), n=0 \sim N-1\}$ is applied to the input layer. n_0 is a starting point of the observation. There are P signal classes. We use the notation $x_{pm}(n_0+n)$ to denote the p th class and the m th signal. The n th input unit receives $x_{pm}(n_0+N-n)$. The connection weight from the n th input unit to the j th hidden unit is w_{nj}. The input potential net_j and the output y_j of the j th hidden unit are given by

$$net_j = \sum_{n=1}^{N} w_{nj} x_{pm}(n_0+N-n) + \theta_j \tag{1}$$

$$y_j = f_H(net_j) \tag{2}$$

where, $f_H(\cdot)$ is an activation function of the hidden layer and θ_j is the bias of the j th hidden unit. The hidden unit output y_j is transferred to the output layer. The same process, as in the hidden unit layer, is carried out in the output layer. The number of the output units is equal to that of the signal classes. The MLNN is trained so that a single output unit responds to one of the signal classes.

2.2 Signal Classification Mechanism

In the case of using a single neuron, to classify the signal set $\boldsymbol{X}$, consist of two classes $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, in an N-dimensional space is viewed as dividing the N-dimensional space into two sub-spaces by a hyper-plane $\boldsymbol{w}^t\boldsymbol{x}=0$. Here, $\boldsymbol{w}$ is a set of connection weights and $\boldsymbol{x}$ is an input signal with N samples, that is equivalent to an N-dimensional vector. Each sub-space includes the signals belong to either $\boldsymbol{X}_1$ or $\boldsymbol{X}_2$. When this hyper-plane exists, the signal set $\boldsymbol{X}$ can be classified correctly. This can be expressed by

$$\left.\begin{array}{ll} \boldsymbol{w}^t\boldsymbol{x} > \alpha, & \boldsymbol{x} \in \boldsymbol{X}_1 \\ \boldsymbol{w}^t\boldsymbol{x} < \alpha, & \boldsymbol{x} \in \boldsymbol{X}_2 \end{array}\right\} \tag{3}$$

Monotonically increasing nonlinear functions, including a threshold function and a sigmoid function Eq. (4), are used as an activation function of the neuron.

$$y(net) = \frac{1}{1+e^{-net}} \tag{4}$$

Pattern classification ability of the MLNN has been studied by many researchers. Funahashi [16] has proved that the two-layer NN can approximate any continuous function with any accuracy if a large number of hidden units are used. The activation function is the sigmoid function. Gibson [15], Cover [17] and Makhoul [19] separately demonstrated that the MLNN has a high degree of freedom of forming sub-regions at the hidden layer. They pointed out that such a high flexibility realized with non-linear activation function and multilayered structure. It allows forming suitable regions to a complex data classification.

3. Signal Classification by Linear Filters

3.1 Filter Design and Implementation

In a finite impulse response (FIR) filter with a direct form [20], the output is given by

$$y_p(n) = \sum_{k=0}^{N-1} x(k+n_0)h_p(n-k), \tag{5}$$
$$h_p(n-k) = 0, n-k < 0$$

Here, $h_p(n-k)$ are the filter coefficients, by which the p th class signal can be extracted. To realize a high-Q filter, a very high-order transfer function is required. However, a linear phase is easily realized.

An infinite impulse response (IIR) filter [20] requires a low-order transfer function, that is a small number of coefficients. However, the recurrent structure requires higher computation than the FIR filter.

One of the IIR filter realization is a cascade form of the second-order circuits, whose transfer function is written as

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}. \tag{6}$$

The output $y(n)$ is calculated as

$$w(n) = x(n) - b_1 w(n-1) - b_2 w(n-2) \tag{7}$$
$$y(n) = a_0 w(n) + a_1 w(n-1) + a_2 w(n-2). \tag{8}$$

$w(n)$ is an internal variable as shown in Fig. 1. A high-Q filter can be realized using a low-order transfer function. However, the linear phase response cannot be guaranteed.

3.2 Signal Classification by Output Power

The same number of the filters as that of the signal

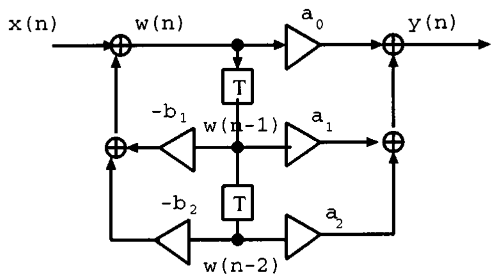


Fig. 1 Second order IIR filter.

classes are used in the signal classification. The p th class filter is designed to extract the frequency components of this class, and to suppress those of all the other classes. The power of the p th filter output S_p is calculated by

$$S_p = \sum_{n=n_1}^{n_1+K-1} y_p^2(n). \tag{9}$$

Where, $y_p(n)$ is the filter output and K is the number of the output samples. n_1 is the beginning of the steady state response. Classification is done by using the following criterion.

$$\text{If } \; S_p = \max_{p'}\{S_{p'}\} \; \text{ then } \; \boldsymbol{x} \in \boldsymbol{X}_p \tag{10}$$

that is the signal is classified into the p th class.

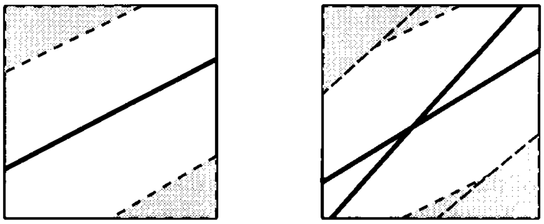
Next, computation complexity required in calculating the output power is discussed. The FIR filter with a direct form always needs N computations in calculating one output as shown in Eq. (5). One computation includes one multiplication and one addition. It is independent of the filter order denoted N_{FIR}. In other words, a very high-Q, that is high-order FIR filter can be used to achieve higher resolution without increasing in the memory capacity and the number of computations. The output samples in the steady state are used in calculating the output power.

On the other hand, the IIR filter has a recursive structure as shown in Fig. 1. In calculating the M th output $y(M)$, the filter should operates from $n = 0$ to $n = M$. Letting filter order be N_{IIR}, $y(M)$ requires $(5/2)N_{IIR}M$ computations. It is mainly determined by N_{IIR} and M, not N. Here, we assume the 2nd-order section needs 5 computations as shown in Fig. 1. Furthermore, $y(M)$ in the steady state should be used in estimating the output power. Thus, even though $N_{IIR} \ll N_{FIR}$, the IIR filter may require more computations than the FIR filter in estimating the output power.

3.3 Signal Classification Performance

Classification performance of the LF methods is investigated based on the spectrum distribution of the signals regarding the number of the signal samples.

When many samples are used to represent the input signals, the frequency components are almost the same as the original signal's. Then highly accurate signal classification is possible by the LF methods. To analyze a frequency component by a high-Q band-pass filter (BPF), difference of the output power between the input signals that include or not include the frequency components are obtained. Moreover, the output of the high-Q BPF nearly regarded as a sinusoidal waveform. Then it is possible to identify that the frequency component is included in an input signal or not with small number of filter output samples.



(a) N is large and k_l is small. (b) N is small and k_h is large.

Fig. 2 Signal detection region of FIR filter.

If the input signal is in the p th class, the outputs of the p th class filter $y_p(n)$ and the others $y_{p'}(n)$ satisfy the following equation.

$$\sum_{n=n_1}^{n_1+K_l-1} |y_p(n)| \gg \max \sum_{n=n_1}^{n_1+K_l-1} |y_{p'}(n)| \tag{11}$$

Where, K_l is the number of the filter outputs, and is assumed to be small. Supposing an appropriate threshold α, this condition can be replaced by

$$\sum_{n=n_1}^{n_1+K_l-1} |y_p(n)| = \sum_{n=n_1}^{n_1+K_l-1} \left| \sum_{k=0}^{N-1} x(k+n_0)h_p(n-k) \right| > \alpha. \tag{12}$$

In this equation, the right hand inequality forms some regions in an N-dimensional space, where the p th class signals are included. This region is called a signal detection region of the p th class. Figure 2 (a) shows a conceptual image of the signal detection regions given by Eq. (12) for two-dimensional signals. The shaded parts are the signal detection regions and the solid line shows a boundary of the regions that formed by $y_p(n) = 0$ in Eq. (12). The signals of the p th class are concentrated in the shaded parts and the other class signals are distributed around the boundary.

When the number of the signal samples is small, the frequency components or the spectrum distribution is distorted from those of the original signal's. Because, using a small number of the signal samples is equal to using a short interval window, and this affects the amplitude response of the signal. The signal detection region is formed by

$$\sum_{n=n_1}^{n_1+K_h-1} |y_p(n)| > \alpha. \tag{13}$$

Where $K_h > K_l$. The region specified by this inequality is wider than that given by Eq. (12). Equation (13) can be satisfied when some outputs take large values than α. Then the condition of the classification is relaxed by using many output samples. A conceptual image of this extended regions is illustrated as some shaded parts in Fig. 2 (b).

4. Comparison between MLNN and LF Methods

4.1 Degree of Freedom of Space Division

The degree of the freedom of forming the class region is discussed in the following. The filter coefficients to calculate $y(n)$ are $h_p(n-N+1) \sim h_p(n)$. Thus, a set of successive N coefficients is used to calculate $y(n)$. Let this set be $\boldsymbol{h}_p(n,N)$. There is strong correlation among $\boldsymbol{h}_p(n,N)$. In other words, they cannot be determined independently. They are designed to extract the necessary frequencies. $\boldsymbol{h}_p(n,N)$ corresponds to a set of the connection weights from the input to the hidden layers of the MLNN. These connection weights do not have any constraints for a select. They can be adjusted using the training data. Therefore, the MLNN can realize more flexible sub-regions, and is superior to the LF methods in pattern classification. However, the MLNN is dependent on the training data. The training should be done to achieve good generalization performance.

Discussions based on computer simulation will be given in Sect. 5.

4.2 Learning Ability and Convergence Property

Linear Filters:

When the frequency components of the signals are known in advance, the filter specification can be determined, and the filters can be designed to extract the necessary frequencies and suppress the unnecessary ones. Usually, high-Q amplitude and linear phase are desirable. On the other hand, when the frequencies are not known, the filters cannot be designed following some specifications, rather they should be designed through some training algorithms like "adaptive filters." In this paper, however, it is assumed that the frequency components of the signals are known, and the former case is taken into account.

Multilayer Neural Networks:

A main difference between the linear filters and the general neural networks is "non-linearity." Especially, if non-linear units are not used in the MLNN, it is equivalent to a set of the FIR filters. In this case, signal classification process in both methods becomes the same. This is induced by results of Sects. 2 and 3. On the other hand, supervised training algorithm, like the Back-propagation (BP) algorithm [1], is used to train the MLNNs. In this paper, the BP algorithm is used to train the MLNN. Thus, discussions on learning ability and convergence property are important. As described in Sect. 2, for the MLNN, the classification problem is equivalent to dividing the N-dimensional space into several sub-spaces. The following is discussed based on this point of view.

As mentioned before, the number of the signal samples is assumed to be small. This is further divided into the following two cases, (1) a very small number, and (2) a relatively small number.

In the case (1), the frequency components become vague. In other words, the regions, in which the signals of each class are distributed, are changed from their original distribution. Sometime, the class regions are mixed. However, if they are not overlapped, it is possible to separate the regions into the different classes. In this case, feature is not clear and the training is relatively difficult.

In the case (2), the signals include accurate frequency components, and they are distributed in some specified regions. The regions of the different classes are separated. However, the boundary between them may be complicated and narrow. Then, feature is clear and the training is easier than that of case (1).

Furthermore, the circuit complexity, which is mainly determined by the number of the hidden units, is practically important. In the MLNN method, to achieve complete separation, that is to form the complicated boundary, many hidden units are required. Then MLNN is required to adjust a number of connection weights. For this reason, the learning will slowly converge, and will be easily trapped at the local minimum. Therefore, the initial connection weights should be carefully determined.

On the other hand, if not enough number of the hidden units are used, the complete separation is impossible. However, relatively high classification rate can be obtained due to high degree of freedom of forming the boundary as mentioned in Sect. 2. In this case, stable and fast convergence can be obtained.

5. Simulation Results and Comparisons

Since, the MLNN method is useful for general pattern classification, then, to compare the both methods fairly, the following multi-frequency signals are used. The frequencies are located alternately between the signal classes, and the amplitude and the phase of each frequency component are generated randomly. Therefore, the signal waveforms of the different classes are similar. This kind of classification may be a difficult problem.

5.1 Multi-Frequency Signal

The p th signal class, denoted $\boldsymbol{X}_p$, includes M signals.

$$\boldsymbol{X}_p = \{x_{pm}(n), m = 1 \sim M, n = 0 \sim N-1\} \quad (14)$$

The multi-frequency signal is defined as follows:

$$x_{pm}(n) = \sum_{r=1}^{R} A_{mr} \sin(\omega_{pr} nT + \phi_{mr}) \quad (15)$$

where, $\omega_{pr} = 2\pi f_{pr}$, f_{pr} is the r th frequency component of the p th class. T is a sampling period. Amplitude A_{mr} and phase ϕ_{mr} of each frequency component are randomly generated in $(0,1]$ and $[0,2\pi)$, respectively. Two classes are used. The number of the signal samples is N=10 or N=20. The frequencies in one class (class 1) are 1, 2 and 3 Hz, and in the other class (class 2), 1.5, 2.5 and 3.5 Hz, respectively. A sampling frequency is 10 Hz. These frequencies can be scaled.

2000 input signals are prepared for each class. For the MLNN, 200 signals are used for training, and 1800 signals for testing. After the training converges, the training signals were perfectly classified. Thus, the MLNN is equivalently evaluated with 2000 signals. All the signals are used for testing the LF methods.

For noisy signals, the additive noise, uniformly distributed in $[-0.5, 0.5]$, is used. The SNR is about 6.5 dB.

5.2 Multilayer Neural Network Design

The MLNN with a single hidden layer is used. Minimization of the number of hidden units have been well discussed [23], [24]. In this paper, however, it is determined by experience. Almost the highest classification performance was obtained with three hidden units. The number of output units is equal to that of the signal classes. A single output unit is assigned to one class. This means the MLNN is trained so that a single output unit responds to one of the signal classes.

Back-propagation (BP) algorithm is used for training the networks. Both noise-free and noisy signals' sets are used in a training phase and a testing phase. The learning rate η and the momentum term coefficient α are 0.1 and 0.8, respectively, which are decided also by experience. The training is stopped when the mean squared error is less than 0.01 or the number of iterations exceeds 3000.

A ratio of the number of the correctly classified signals and the number of the entire testing signals, denoted "classification rate," is evaluated under several conditions. A signal is classified into the p th class if the p th output unit takes the maximum value.

5.3 Linear Filter Design

FIR filter

Figure 3 shows an example of the amplitude response of a 1000th-order FIR filter for the class 1. It has the peaks at frequencies 1, 2 and 3 Hz, and the band width is 0.02 Hz. The output signal of the FIR filter is calculated by Eq. (5) in the steady state.

IIR filter

The transfer function of the i th second order circuit is given as follows:

$$H_i(z) = \frac{1 - 2\cos\theta_{zi} z^{-1} + z^{-2}}{1 - 2r_i \cos\theta_{pi} z^{-1} + r_i^2 z^{-2}}. \tag{16}$$

The total transfer function is

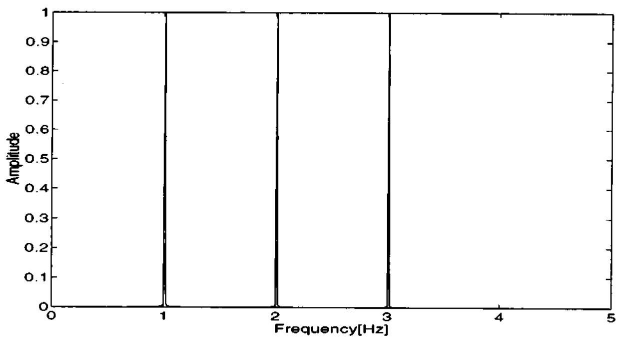


Fig. 3 Amplitude response of FIR filter designed to extract class 1 signals.

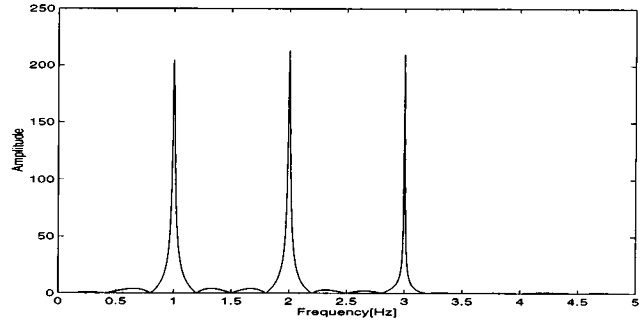


Fig. 4 Amplitude response of IIR filter to extract class 1 signals.

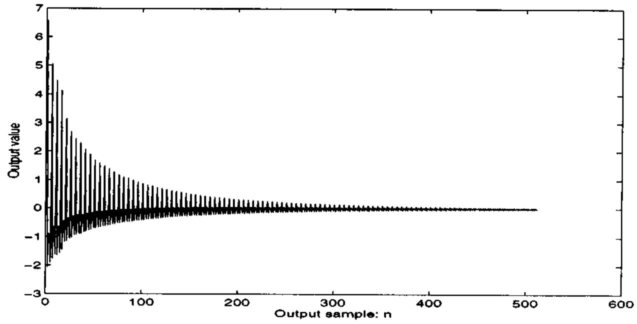


Fig. 5 Impulse response of class 1 filter.

$$H(z) = \prod_{i=1}^{I} H_i(z). \tag{17}$$

In order to realize a high-Q filter, fifteen zeros and three poles are used for each class. r_i in Eq. (16) for the class 1 are 0.9945, 0.995 and 0.9985, for the class 2, 0.994, 0.995 and 0.9985, respectively. The pole frequencies are 1.0, 2.0 and 3.0 Hz for the class 1, and 1.5,2.5 and 3.5 Hz for the class 2, respectively. All zeros locate on the unit cycle. Figure 4 shows the amplitude response of the class 1 filter. The impulse response is shown in Fig. 5.

By using this filter for classification of the multi-frequency signals, the classification rate for the noise free signals with 10 samples is 86.9%. To achieve this

Table 1 Probability of exact signal classification in percentage when computation is limited.

Methods	N=10		N=20	
	NFS	NS	NFS	NS
MLNN	97.8	86.9	97.7	91.5
FIR	4.7	3.7	100	87.5
IIR	0.0	0.0	49.0	49.0

N : Number of samples
NFS : Noise Free Signal, NS : Noisy Signal

Table 2 Probability of exact signal classification in percentage when computation is not limited.

Methods	N=10		N=20	
	NFS	NS	NFS	NS
MLNN	100	91.5	100	99.4
FIR	100	90.5	100	99.8
IIR	95.4	86.5	100	99.8

N : Number of samples
NFS : Noise Free Signal, NS : Noisy Signal

accuracy, 2000 output samples are required. This rate is not good compared with that of the FIR filter will be shown in Table 2. The reason is the phase distortion caused by the high-Q amplitude response. By using a lower-Q filter than the above, the classification rate was increased from 86.9% to 95.4%. In this case, r_i in Eq. (16) are changed to 0.94, 0.94 and 0.98 for the class 1, and 0.92, 0.935 and 0.98 for the class 2. In the lower-Q IIR filter, 200 output samples are required. On the other hand, since the FIR filter always can guarantee the linear phase, a very high-Q filter can be effectively used as shown in Fig. 3.

5.4 Computational Complexity

Normalized computational complexity (NCC) is defined to compare classification performance based on the same number of computations. The NCC for each method is listed on Table A·1 and A·2. The calculation method of NCC for each classification method is given in Appendix.

5.5 Signal Classification Results

The following two conditions are investigated; the number of computations is limited or not limited. In the former case, computations of the LF methods are decided as almost the same as in the MLNN method.

The classification rates with limited computations are listed in Table 1 in percentage. From this table, the MLNN method can provide higher performance than the LF methods. The classification rates of using the signals with 20 samples are better than those of the signals with 10 samples. In the LF methods, the classification rates are higher for 20 samples' signals than that of 10 samples'. Therefore, non-linearity is notable for 10 samples' signals and is not notable for 20 samples'.

Classification rate of the IIR filter for 20 sample signals is worse than that of FIR filter. The main reason of this difference comes from a recurrent structure of IIR filter. If the computation is limited, the output samples in the transient state become dominant in the output power, and accuracy is decreased.

In the case of the computation being not limited, the classification rates are shown in Table 2. The classification rates of the LF methods can be improved, and are almost the same in all methods.

The MLNN method uses the valley shape activation function [7] instead of the sigmoid function in the hidden layer.

5.6 Learning Ability of Multilayer Neural Network

As discussed in Sect.4.2, when a large number of hidden units are used, it is difficult to converge to the best solution. The initial connection weights should be carefully selected. When random numbers are used as the initial connection weights, the MLNN could not achieve good classification rates as the filters. However, by using the coefficients of the FIR filter as the initial connection weights, the MLNN achieved the same classification rates as the filters'.

In this case, the valley shaped function is used in the hidden unit. The valley shaped function rectifies unit input and it can detect the signal amplitude. From the following discussion, the initial connection weights between the input layer and the hidden units can be determined by the coefficients of the FIR filter. From Eqs. (1) and (5), the input of the hidden unit is correspond to the output of the FIR filter calculated by using a set of successive N coefficients. The output of the hidden unit using the valley shaped activation function is equal to the rectified output of the FIR filter. The input of the output unit is the weighted sum of the absolute value of the FIR filter outputs, which is correspond to the filter output power. Although, this function can be realized by using two sigmoid functions, the former can make fast convergence possible.

5.7 Robustness of MLNN to Noise Level Changes

Robustness for noise level changes is guaranteed by the filters. However, this kind of robustness is not always guaranteed by the MLNN. Then, the robustness of the MLNN for noise level changes is further investigated.

5.7.1 Analysis of Connection Weight

By comparing the Eqs. (1) and (5), the connection weights between a hidden unit and input layer correspond to the filter coefficients $h_p(n-k)$. Then the connection weight between the input layer and the hidden layer are analyzed by using Fourier transform. Figures 6 and 7 are the amplitude responses of the connec-

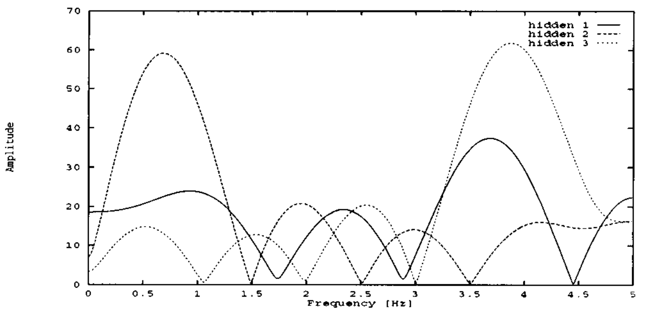


Fig. 6 Fourier transform of sets of connection weights trained with noise free signals.

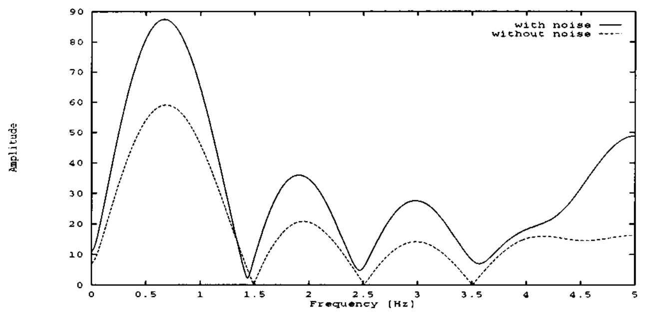


Fig. 8 Fourier transform of two sets of connection weights trained with noise free and noisy signals.

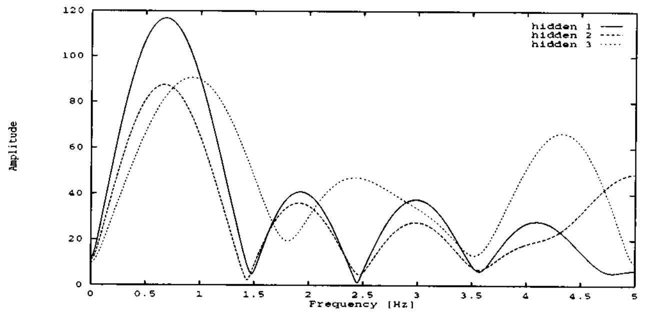


Fig. 7 Fourier transform of sets of connection weights trained with noisy signals.

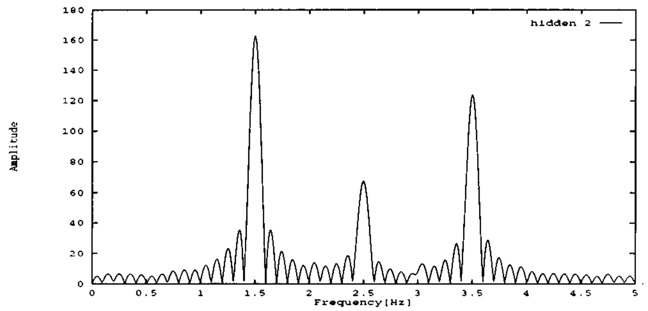


Fig. 9 Fourier transform of a set of connection weights trained with noise free signals. Number of hidden units is 100.

tion weights trained with noise free and noisy signals. The numbers of the input units and the hidden units are 10 and 3, respectively. From above two figures, the amplitude response of the connection weights suppressing other class frequencies. From these figures, the amplitude response of the connection weights suppressing other class frequencies.

When the MLNN is trained using noise free signals, there are two types of amplitude response; one is suppressing class 1 frequencies and the other is suppressing class 2 frequencies. The features of the noise free signal may be clear and the class regions in an N-dimensional space are separated clearly. However, the features of the noisy signals are dispersed by random noise, and the class regions are not clearly separated. Therefore, the training using noisy signals is more difficult than that of using the noise free signals. From the Fig. 7, only the amplitude response of suppressing class 2 frequency components is obtained, so, this result supports above discussion.

Figure 8 shows the amplitude responses of Figs. 6 and 7 in the same graph. The connection weights of the input layer and the 3rd hidden unit are used. From this figure, when the MLNN trained by noisy signals, the amplitude response slightly changed into flatter than that of trained using noise free signals. So, the MLNN adapted to the noisy signals by changing its connection weight to have insensitive amplitude response. The FIR filter has a sharp amplitude response and can sufficiently suppress non-interest frequencies. The MLNN, it does not have such a sharp amplitude response.

When the number of hidden units is increased from three to 100, the amplitude response of the connection weights is changed as shown in Fig. 9. In this case, the amplitude response is similar to that of the FIR filter.

From above analysis, the MLNN achieved amplitude response that can classify the signals. However, the amplitude response of the connection weights is changed due to number of hidden units. So, the MLNN can make a suitable amplitude response due to given number of hidden units.

As the training is converged, it is confirmed that the input signals are converted into the linearly separable output pattern at the hidden layer. Then the patterns can be classified by single layer NN of the hidden and the output layers. The connection weights between the hidden layer and the output layer are adjusted to emphasize the feature extracted by the connection weights from the input layer to the hidden layer. From the simulation results, some hidden units are activated for one class, then the connection weights from these hidden units to the output unit assigned to this class are relatively large.

5.7.2 Robustness to White Noise

For LF methods, robustness to the noise level change is guaranteed. So, when the noise level is reduced, classification rate will be better. The MLNN is trained with 20 samples including ±0.5 additive random noise. When the noise level is decrease to 0, the classification rate is reduced from 90.6% to 89.7%. In this case, 200 data for each class is used for training. So, generalization for smaller noise signals is not achieved. However, by increasing the number of the training signals from 200 to 400, the network provides the classification rates of 91.7% for ±0.2 additive noise, and 91.3% for the noise free signals, respectively. Thus, the robustness for noise level change can be guaranteed by training the network with a larger number of the noisy signals.

6. Conclusions

The signal classification by the MLNN and the LF methods have been compared with each other. The signals to be classified are specified by the multi-frequency components. The MLNN method is more efficient to classify the short period signal than the other under the limited computation. In other words, the MLNN has high-degree of freedom to combine the sub-regions at the hidden layer into the desired regions at the output layer. On the other hand, the LF methods do not have such freedom due to their linearity. Therefore, when the signal period is very short, and consequently their vectors are distributed widely and randomly in the signal space, the MLNN method is superior to the LF methods. Both methods, however, provide almost the same high performance for the relatively long period signal without the computational limitation. The MLNN method has good generalization and robustness for the un-training data and noisy signals, respectively. It has, however, some convergence problem, that is local minimum solutions, in the long period signal case. The initial weights should be carefully determined. The FIR filter with a direct form is more useful for detecting the short period signals than the IIR filter in classification accuracy and computational requirement. Because a very high-Q response with linear phase can be realized, and computation of an output is separated from the other, in which the number of computations is proportional to that of the input samples.

References

[1] D.E. Rumelhart, J.L. McCelland, and the PDP Research Group, "Parallel Distributed Processing," The MIT Press, 1986.

[2] J. Zhan and F. Li, "A self-organization neural network and detecting signals," Proc. WCNN'93, Portland, Oregon, vol.4, pp.748–751, 1993.

[3] R. Vanderbeek and A. Garper, "A back-propagation network for analog signal separation in high environments," Proc. IJCNN'92, Boltimore MD, vol.1, pp.664–669, 1992.

[4] K.A. Al-Mashouq and I.S. Reed, "The use of neural nets to combine equalization with decoding for severe intersymbol interference channels," IEEE Trans. Neural Network, vol.5, no.6, pp.982–988, 1994.

[5] D.M. Hummels, W. Ahmed, and M.T. Musavi, "Adaptive detection of small sinusoidal signals in non-gaussian noise using an RBF neural network," IEEE Trans. Neural Network, vol.6, no.1, pp.214–219, 1995.

[6] Z.H. Michalopoulou, L.W. Nolte, and D. Alexandrou, "Performance evaluation of multilayer perceptrons in signal detection and classification," IEEE Trans. Neural Network, vol.6, no.2, pp.381–386, 1995.

[7] K. Nakayama and K. Imai, "A neural demodulator for amplitude shift keying signal," Proc. ICANN'94, Sorrent, Italy, vol.2, pp.1017–1020, 1994.

[8] K. Ohnishi and K. Nakayama, "A neural demodulator for quadrature amplitude modulation signals," Proc. ICNN'96, Washington, DC, pp.1933–1938, 1996.

[9] K. Hara and K. Nakayama, "High resolution of multi-frequencies using multilayer networks trained by back-propagation algorithm," Proc. WCNN'93, Portland, Oregon, vol.4, pp.675–678, 1993.

[10] K. Hara and K. Nakayama, "Comparison of activation functions in multilayer neural network for pattern classification," Proc. ICNN'94, Orlando, Florida, vol.5, pp.2997–3002, 1994.

[11] A.C. Tsoi and R.A. Pearson, "Comparison of three classification techniques, CART, C4.5 and multi-layer perceptrons," Neural Information Processing System 3, pp.963–969, 1991.

[12] L. Atlas, R. Cole, J. Connor, M. El-Sharkawi, R.J. Mark II, Y. Muthusamy, and E. Barnard, "Performance comparison between backpropergation networks and classification trees on three real-world applications," Neural Information Processing System 2, pp.622–629, 1990.

[13] S.L. Gish and W.E. Blanz, "Comparing the performance of connectionist and statistical classifiers on a image segmentation problem," Neural Information Processing System 2, pp.614–621, 1990.

[14] K. Ng and R.P. Lippmann, "A comparative study of the practical characteristics of neural network and conventional pattern classifiers," Neural Information Processing Systems 3, pp.970–976, 1990.

[15] G.J. Gibson and C.F.N. Cowan, "On the decision regions of multilayer perceptrons," Proc. IEEE, vol.78, pp.1590–1594, 1990.

[16] K. Funahashi, "Approximate realization of identity mappings by three-layer neural networks," IEICE Trans., vol.J73-A, no.1, pp.139–145, 1990.

[17] T.M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," IEEE Trans. Electric Computers, vol.EC–14, pp.326–334, 1965.

[18] A. Kowalczyk, "Counting function theorem for multilayer networks," Neural Information Processing System 6, pp.375–382 1995.

[19] J. Makhoul, A. El-Jaroudi, and R. Schwarts, "Partitioning Capabilities of Two-Layer Neural Networks," IEEE Trans. Signal Processing, vol.39, no.6, pp.1435–1440, 1991.

[20] J.S. Lim and A.V. Oppenheim, "Advance Topics in Signal Processing," Prentice-Hall Inc., 1992.

[21] S. Haykin, "Neural networks expand SP's horizons," IEEE Signal Processing Magazine, pp.24–49, March 1996.

[22] K. Nakayama, S. Inomata, and Y. Takeuchi, "Reductions in number of bits for digital realization of multilayer neural network," IEICE Trans., vol.J73-D-II, no.8, pp.1336–1345, 1990.

[23] J. Sietsma and R.J.F. Dow, "Creating artificial neural networks that generalize," Neural Networks, vol.4, pp.67–79, 1991.

[24] T. Ueda, K. Takahashi, and S. Mori, "A structural learning for multi-layered neural networks by using fuzzy set–A

pruning weights and units," IEICE Trans., vol.J78-D-II, no.10, pp.1479–1490, 1995.

Appendix: Normalized Computation Complexity

Normalized computational complexity (NCC) and the number of the parameter for each method is listed in Tables A·1 and A·2. An inner product of N-dimensional vector is counted as unity in NCC. The number of elements of the free parameter is listed in these tables. The parameter for each method is explained in the followings. Since, parameters should be integers, so NCC for all methods is not exactly the same. The number of samples of the signal is denoted by N for the following methods.

A.1 Multilayer Neural Network

The NCC for the MLNN is calculated for the MLNN architecture that performs the highest classification for the training signals and for the test signals. After the training converges, the hidden unit outputs approach to '1' or '0' [22]. So, the sigmoid function can be replaced by a threshold function in the test phase. Therefore, the calculation of the sigmoid function is omitted from NCC.

Let the number of input units be N, the number of hidden units be M, and the number of the output units is P, respectively, NCC is given by $NCC = M + \frac{M}{N}P$. The parameter is the number of the hidden units.

A.2 FIR Filter

From the Eq. (5), NCC for one output calculation is unity. If the number of the signal classes is denoted by P, and the number of output samples required to calculate one output power be K, then NCC is calculated as $NCC = P \times K$. Here, the number of the output samples is the parameter.

A.3 IIR Filter

In the case of N_p pole frequencies and N_z zero frequencies are used and $N_z > N_p$, then N_z of the 2nd-order circuits are used. Each circuit includes five inner products of the signal and the filter coefficients, then computation for one filter output is $N_p \times 5 + (N_z - N_p) \times 3 = 2 \times N_p + 3 \times N_z$. Then $NCC = (2 \times N_p + 3 \times N_z)/N$. N_p is as the same as the number of frequencies included in one class. The number of the output samples is the parameter.

Table A·1 Normalized computational complexities and parameters.(Computation is limited.)

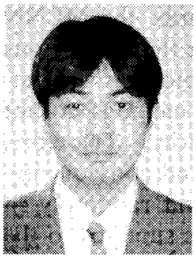
Methods	N=10		N=20	
	NCC	NP	NCC	NP
MLNN	3.6	3	3.3	3
FIR	4.0	2	4.0	2
IIR	5.1	1	5.1	2

N: Number of samples
NP: Number of elements of parameter

Table A·2 Normalized computational complexities and parameters.(Computation is not limited.)

Methods	N=10		N=20	
	NCC	NP	NCC	NP
MLNN	48	40	44	40
FIR	20	10	20	10
IIR	1020	200	510	200

N: Number of samples
NP: Number of elements of parameter



Kazuyuki Hara received the B.E. and M.E. degree in electronics engineering from Nihon University, Tokyo, Japan, in 1979 and 1981, respectively. From 1979 to 1987 he was involved in NEC Home Electronics Corporation. He joined Toyama Polytechnic College in 1987, where he is a Lecturer. He is currently a doctor course student of Graduate School of Natural Science and Technology, Kanazawa University. His current research interests include neural networks.

Kenji Nakayama received the B.E. and Dr. degrees in electronics engineering from Tokyo Institute of Technology (TIT), Tokyo, Japan, in 1971 and 1983, respectively. From 1971 to 1972 he was engaged in the research on classical network theory in TIT. He was involved in NEC Corporation from 1972 to 1988, where his research subjects were filter design methodology and signal processing algorithms. He joined the Department of Electrical and Computer Engineering at Kanazawa University, in Aug. 1988, where he is currently a Professor. His current research interests include neural networks and adaptive filters.