PERGAMON

Neural Networks 14 (2001) 1307–1321

Neural Networks

www.elsevier.com/locate/neunet

Contributed article

Bi-directional computing architecture for time series prediction

Hiroshi Wakuya*, Jacek M. Zurada

Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292, USA

Received 6 December 1999; revised 14 May 2001; accepted 14 May 2001

Abstract

A number of neural network models and training procedures for time series prediction have been proposed in the technical literature. These models studied for different time-variant data sets have typically used *uni-directional* computation flow or its modifications. In this study, on the contrary, the concept of *bi-directional* computational style is proposed and applied to prediction tasks. A bi-directional neural network model consists of two subnetworks performing two types of signal transformations bi-directionally. The networks also receive complementary signals from each other through mutual connections. The model not only deals with the conventional future prediction task, but also with the past prediction, an additional task from the viewpoint of the conventional approach. An improvement of the performance is achieved through making use of the future-past information integration. Since the coupling effects help the proposed model improve its performance, it is found that the prediction score is better than with the traditional uni-directional method. The bi-directional predicting architecture has been found to perform better than the conventional one when tested with standard benchmark sunspots data.

Keywords: Time series prediction; Bi-directional computation style; Bi-directional neural network model; Future prediction; Past prediction; Dynamic neuron; Sunspots data

1. Introduction

This world is filled with various time dependent phenomena. For example, our speech and heart rates change in respect to power, frequency, and so on, with time in millisecond order. An exchange rate between US dollars and Japanese yen also changes in hourly, daily, monthly or yearly intervals. This time-varying information, usually sampled with constant period, is called 'time series' or 'temporal sequences'. In general, it is difficult to estimate future values of time series because past information disappears with time unless an appropriate memory system is provided. This is not the only reason for prediction difficulties, but is unique to time series processing, so how to preserve it without major loss is important when time series prediction is attempted.

The task of time series prediction has been undertaken by many researchers. Numerous prediction methods have been examined for more than several decades. The most popular techniques have used linear systems framework. An auto-regressive moving average (ARMA) model is one of the most popular models in this category. Although easy to implement and understand, the ARMA model does not produce a good score because most systems to be estimated possess nonlinear characteristics (Gershenfeld & Weigend, 1993). Nonlinear techniques have, therefore, recently received greater attention. Connectionist techniques based on neural information processing are examples of recently developed nonlinear approaches, which have resulted in new prediction models. One of the critical aspects of neural networks-based predictors is how to implement a short-term memory to handle past information because a traditional multi-layer perceptron (MLP) does not have any dynamics and feedback loops.

A simple way to address this issue when dealing with time series is to make use of spatially-converted temporal patterns (Gorman & Sejnowski, 1988; Sejnowski & Rosenberg, 1987; Weigend, Huberman & Rumelhart, 1990). It is also possible to achieve such a conversion automatically with an array of unit-delay elements called a tapped delay line model. Another method is to introduce dynamical properties into a static neural network. In this category, Wan (1993) proposed a finite impulse response (FIR) neural network, an MLP model with originally scalar synaptic connections replaced with FIR filters. The other method makes use of feedback pathways to circulate signals inside the model. It is called a recurrent model. Kleinfeld (1986) proposed a mutually connected network

* Corresponding author.

E-mail addresses: wakuya@aivo.spd.louisville.edu (H. Wakuya), jmzura02@athena.louisville.edu (J.M. Zurada).

0893-6080/01/$ - see front matter
PII: S0893-6080(01)00087-9

with delayed feedback to produce a cyclic pattern. Jordan (1986) presented a model which is equivalent to an MLP with feedback pathways from output neurons to hidden ones through exponential-decay memories. Elman (1990) proposed yet another model of an MLP with feedback connections around the hidden layer for sentence generation. An attempt to summarize these approaches and related neural network architectures from the viewpoint of a short-term memory system is provided by Mozer (1993).

Once the network architecture for prediction is chosen, it then becomes important to estimate its size which could produce a good score of time series prediction tasks. It is said that a generalization performance improves when the network size becomes smaller, but a smaller one may not train sufficiently well for the task. Two kinds of training methods are commonly used to determine an appropriate size of the network. One is called a destructive method and starts from a large-sized network which gradually reduces its components (Ishikawa, 1996; Le Cun, Denker & Solla, 1990; Mozer & Smolensky, 1989; Sietsma & Row, 1988). The other is a constructive method which assumes a small-sized initial network and adds extra neurons when required (Moody & Utans, 1995).

As mentioned before, most conventional models for time series prediction are generally based on a *uni-directional* computation style, i.e. current signals are applied to the system as an input, past signals are also applied to the system's input or preserved as inner representation of the system, and predicted future signals are derived from the system as an output. Numerous studies have attempted to improve their performance through modifications of both network architectures and their training algorithms mentioned above (Cholewo & Zurada, 1997a,b; Geva, 1998; Moody & Utans, 1995; Mozer, 1993; Saad, Prokhorov & Wunsch, 1998; Wan, 1993; Weigend et al., 1990).

In this study, on the contrary, the concept of *bi-directional* computation not only predicts a certain *future* value, but also makes use of a certain *past* value. The method couples those two processes and is especially suitable for time series prediction. Furthermore, its effectiveness is demonstrated based on computer simulations.

In the following part of this paper, a neural network model for bi-directional computation style is described in Section 2. Secondly, computer simulations using sunspots data are performed to provide a comparison between the proposed model and the conventional ones in Section 3. Thirdly, discussion on the bi-directional computation style observed in biological systems is made in Section 4. Finally, conclusions from this study are summarized in Section 5.

2. Neural network model for bi-directional computation style

2.1. Basic concept of bi-directional computation

The term 'bi-directional computation' means that two signal transformations, which are direct and inverse transformations, exist and their performance is improved through their coupling effects. Its original idea was proposed by Wakuya, Futami and Hoshimaya (1994) as a sensorimotor neural network model for temporal sequence generation and recognition. Called a bi-directional neural network model, it consists of two subnetworks and can deal with two kinds of signal transformations bi-directionally. To apply this bi-directional model to the time series prediction, each subnetwork is trained for a task for direct or inverse transformation as follows.

- To predict time series at a certain *future* point.
- To predict time series at a certain *past* point.

Fig. 1 shows the outline of signal flow within the bi-directional computing architecture. In this figure, the upper half part deals with the future prediction task, a transformation

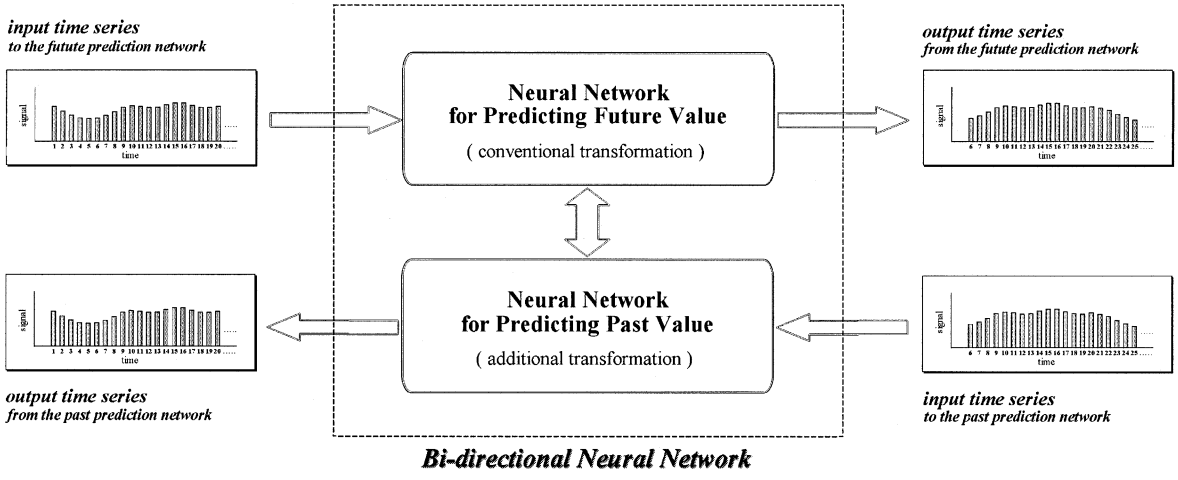


Fig. 1. Outline of signal flow in the bi-directional computing architecture for time series prediction.

from the present signal to the future signal, while the lower half deals with the past prediction task, a transformation from the desired future signal to the present signal. These two transformations cooperate with each other through their mutual connections. Based on the signal transformations to be trained, hereafter these two subnetworks are referred to as a future prediction system and a past prediction system, respectively. It is obvious that the transformation performed in the future prediction system is equivalent to the conventional task and the transformation performed in the past prediction system is an extra one from the viewpoint of the aim for time series prediction. Therefore, an improvement of the future prediction system's performance as a result of coupling with the past prediction system is the center of attention in this paper.

A similar concept was also proposed a few years ago (Schuster & Paliwal, 1997), but it is important to address here that the proposed bi-directional neural network model in this paper can deal with any time series in real-time. So, there are no restrictions on the length of time series applied to the model.

2.2. Bi-directional neural network model

2.2.1. Structure of bi-directional model

Fig. 2 shows a schematic diagram of a bi-directional neural network model (Wakuya et al., 1994). In this figure, each circle and arrow indicate a layer of neurons without feedback connections and a matrix of weights between two successive layers, respectively. This model consists of two mutually connected layered recurrent subnetworks which perform two signal transformations bi-directionally. Each subnetwork is a modified version of Elman's model (Elman, 1990). The upper network computes the transformation for the future prediction task, while the bottom one computes the transformation for the past prediction task. Fig. 3 shows the expanded future prediction subnetwork, where each neuron and weight are indicated by a triangle and a line, respectively. The neurons in the first layer are simple buffers to transfer the input signals applied to the future prediction subnetwork. Each input neuron in this model receives only one kind of time series and handles its components one by one serially. As mentioned later, a single input neuron is provided in this study. The neurons in the remaining three layers compute the weighted sum of all incoming signals as their output signals. As shown in Eqs. (11) and (12) in the next subsection, the neurons in the second and third layers use a regular nonlinear activation function, while the neurons in the fourth layer use a linear function. Dynamic neurons (shaded triangles) which replace simple buffers as the state layer have fixed one-to-one connections (dashed lines) with the second layer neurons, and this modification makes it possible to preserve the recent past information effectively without any loss of signal transmission speed between input and output nodes. And handling this localized state information denoted as $\boldsymbol{s}^{[F]}$ in

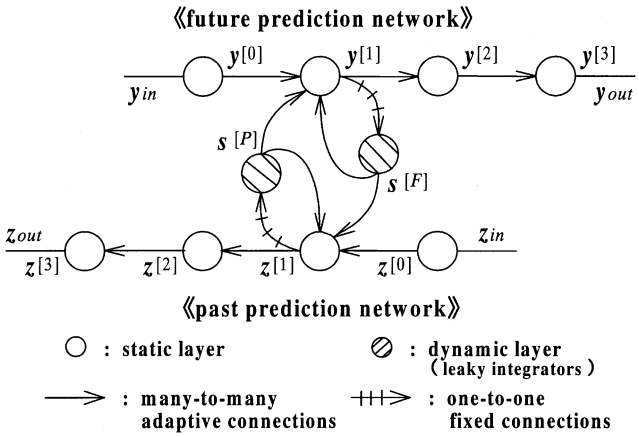


Fig. 2. A bi-directional neural network model.

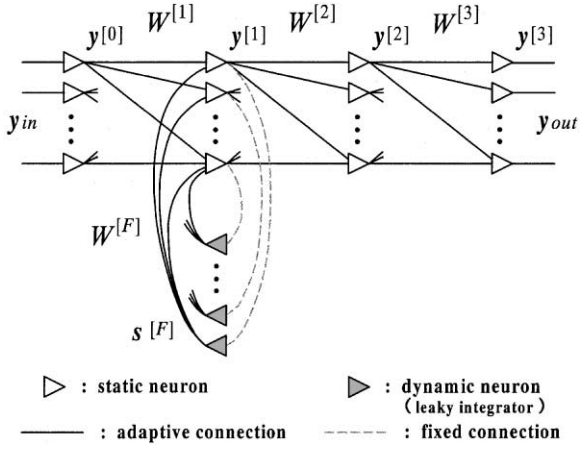


Fig. 3. A future prediction subnetwork in the bi-directional model of Fig. 2.

Fig. 2 and the similar information in the past prediction subnetwork denoted as $\boldsymbol{s}^{[P]}$ jointly, both subnetworks compute state transitions for the future and past prediction processing.

On the contrary, it also can be seen that the bi-directional model is a structuralized version of the modified Elman's model mentioned above in order to develop a bi-directional signal transformation easily. This structuralization is performed such that all static layers are divided into two parts to perform a signal transformation for the future prediction task and for the past prediction, and all weights crossing over to the other subnetwork are removed except those through the state neurons. With this procedure, it is possible to build a multi-directional neural network model such as a tri-, a tetra-, or a penta-directional one if such a model is required.

2.2.2. Signal transmissions equations for bi-directional model

As shown in Fig. 2, the bi-directional model consists of four layers with the output signals in the future production

system denoted as $y_i^{[0]}$, $y_i^{[1]}$, $y_i^{[2]}$, $y_i^{[3]}$ ($i = 1, 2, \ldots, n_l$) where n_l indicates the number of neurons in the l-th layer.[1] Signals in the past prediction system are denoted similarly as $z_i^{[0]}$, $z_i^{[1]}$, $z_i^{[2]}$, $z_i^{[3]}$ ($i = 1, 2, \ldots, n_l$). Then, their signal transmission equations are defined as follows:

$$y_i^{[0]} = [\mathbf{y}_{\text{in}}]_i, \tag{1}$$

$$y_i^{[1]} = f_1(u_i^{[y1]}) = f_1\left(\sum_j w_{ij}^{[1]} y_j^{[0]} + \sum_j w_{ij}^{[\text{F}]} s_j^{[\text{F}]} + \sum_j w_{ij}^{[\text{P}]} s_j^{[\text{P}]}\right), \tag{2}$$

$$\tau \frac{\mathrm{d}s_j^{[\text{F}]}}{\mathrm{d}t} + s_j^{[\text{F}]} = y_j^{[1]}, \tag{3}$$

$$y_i^{[2]} = f_2(u_i^{[y2]}) = f_2\left(\sum_j w_{ij}^{[2]} y_j^{[1]}\right), \tag{4}$$

$$[\mathbf{y}_{\text{out}}]_i = y_i^{[3]} = f_3(u_i^{[y3]}) = f_3\left(\sum_j w_{ij}^{[3]} y_j^{[2]}\right), \tag{5}$$

$$z_i^{[0]} = [\mathbf{z}_{\text{in}}]_i, \tag{6}$$

$$z_i^{[1]} = f_1(u_i^{[z1]}) = f_1\left(\sum_j v_{ij}^{[1]} z_j^{[0]} + \sum_j v_{ij}^{[\text{P}]} s_j^{[\text{P}]} + \sum_j v_{ij}^{[\text{F}]} s_j^{[\text{F}]}\right), \tag{7}$$

$$\tau \frac{\mathrm{d}s_j^{[\text{P}]}}{\mathrm{d}t} + s_j^{[\text{P}]} = z_j^{[1]}, \tag{8}$$

$$z_i^{[2]} = f_2(u_i^{[z2]}) = f_2\left(\sum_j v_{ij}^{[2]} z_j^{[1]}\right), \tag{9}$$

$$[\mathbf{z}_{\text{out}}]_i = z_i^{[3]} = f_3(u_i^{[z3]}) = f_3\left(\sum_j v_{ij}^{[3]} z_j^{[2]}\right), \tag{10}$$

where $\mathbf{y}_{\text{in}}/\mathbf{z}_{\text{in}}$ and $\mathbf{y}_{\text{out}}/\mathbf{z}_{\text{out}}$ are input and output signals of the future/past prediction system, respectively. In this chain of signal transmission formulae, Eqs. (3) and (8) indicate that the dynamic neurons are of first order where the time constant τ stands for their decay property. These neurons preserve context information of time series in the first hidden layer locally and provide it at any past points ideally, if paying no attention on its contents, to predict future values' accuracy. Furthermore, their dynamics can be expressed by a simple equation in the recursive form, so that this type of dynamic neurons is quite easy to perform in each computer simulation program. They are also equivalent to leaky integrators which are known as one of the famous short-term memory systems. Note that two types of context information $\boldsymbol{s}^{[\text{F}]}$, $\boldsymbol{s}^{[\text{P}]}$ are integrated at the first hidden layer which computes $\mathbf{y}^{[1]}$, $\mathbf{z}^{[1]}$ as can be seen from Eqs. (2) and (7) in more detail. Transfer functions f_1, f_2, f_3 are defined by the following customary equations:

$$f_1(x) = f_2(x) = \frac{1}{1 + \exp(-x)}, \tag{11}$$

$$f_3(x) = x. \tag{12}$$

It is clear from Eq. (12) that the last layer consists of linear activation function neurons to remove restriction of the output signal's range.

2.2.3. Training procedure for bi-directional model

The proposed bi-directional model has two pairs of input and output terminals or two subnetworks for future prediction and past prediction, therefore global training is achieved by repetition of two kinds of alternative local training. One is a training phase for the future prediction system and the other is that for the past prediction system (Fig. 4). In the training phase for the future prediction system, only the future prediction system adapts its weights (thick arrows in Fig. 4(a)). Its error function is defined as

$$e_{\text{f}} = \sum_t \sum_i \{[\mathbf{y}_{\text{out}}(t)]_i - d_i^{[\text{F}]}(t)\}^2, \tag{13}$$

where $\mathbf{d}^{[\text{F}]}$ is a desired signal for the future prediction system's output and t is a time index at the training point. Then, adaptation of all weights in the future prediction system such as $w_{\mu\xi}^{[3]}$, $w_{\mu\xi}^{[2]}$, $w_{\mu\xi}^{[1]}$, $w_{\mu\xi}^{[\text{F}]}$, $w_{\mu\xi}^{[\text{P}]}$ is determined by the real-time recurrent learning (RTRL) algorithm (Williams & Zipser, 1989) as follows:

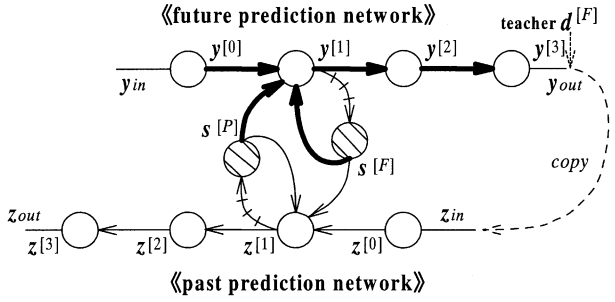
$$\Delta w_{\mu\xi}^{[3]} = -\eta_{\text{f}} \frac{\partial e_{\text{f}}}{\partial u_\mu^{[y3]}} y_\xi^{[2]}, \tag{14}$$

$$\Delta w_{\mu\xi}^{[2]} = -\eta_{\text{f}} \frac{\partial e_{\text{f}}}{\partial u_\mu^{[y2]}} y_\xi^{[1]}, \tag{15}$$

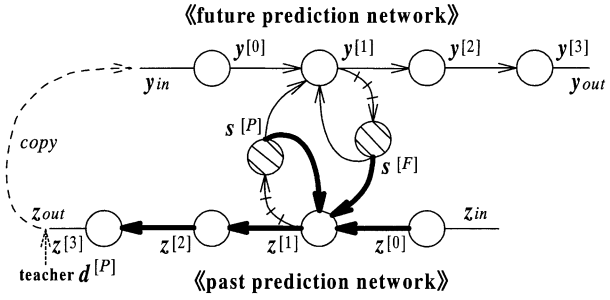
$$\Delta w_{\mu\xi}^{[1]} = -\eta_{\text{f}} \left\{ \frac{\partial e_{\text{f}}}{\partial u_\mu^{[y1]}} y_\xi^{[0]} + \sum_i \frac{\partial e_{\text{f}}}{\partial u_i^{[y1]}} w_{i\mu}^{[\text{F}]} \left(1 + \tau \frac{\mathrm{d}}{\mathrm{d}t}\right)^{-1} [f_1'(u_\mu^{[y1]}) y_\xi^{[0]}] \right\}, \tag{16}$$

$$\Delta w_{\mu\xi}^{[\text{F}]} = -\eta_{\text{f}} \left\{ \frac{\partial e_{\text{f}}}{\partial u_\mu^{[y1]}} s_\xi^{[\text{F}]} + \sum_i \frac{\partial e_{\text{f}}}{\partial u_i^{[y1]}} w_{i\mu}^{[\text{F}]} \left(1 + \tau \frac{\mathrm{d}}{\mathrm{d}t}\right)^{-1} [f_1'(u_\mu^{[y1]}) s_\xi^{[\text{F}]}] \right\}, \tag{17}$$

[1] The number of the hidden neurons is determined arbitrarily, while the number of the input and the output neurons is determined by the time series prediction task uniquely. In this study, a single neuron is provided in the input and the output layers (Section 3.1).



(a) training phase for the future prediction system
— future prediction mode —



(b) training phase for the past prediction system
— past prediction mode —

Fig. 4. A training procedure of the bi-directional model. All connections depicted by thick arrows are modified in each training phase.

$$\Delta w_{\mu\xi}^{[\mathrm{P}]} = -\eta_{\mathrm{f}} \frac{\partial e_{\mathrm{f}}}{\partial u_{\mu}^{[y1]}} s_{\xi}^{[\mathrm{P}]}, \tag{18}$$

where η_{f} is a learning rate, and $(1 + \tau(\mathrm{d}/\mathrm{d}t))^{-1}$ is an operator of the first order decay. In these equations, backpropagating errors circulating inside the bi-directional model are considered only the first lap of the recurrent connections within each subnetwork (e.g. $\mathbf{y}^{[1]} \rightarrow \mathbf{s}^{[\mathrm{F}]} \rightarrow \mathbf{z}^{[1]}$ in Fig. 4(a) and $\mathbf{z}^{[1]} \rightarrow \mathbf{s}^{[\mathrm{P}]} \rightarrow \mathbf{z}^{[1]}$ in Fig. 4(b)) and the rest is omitted as negligibly small for simplicity.[2]

During this training phase, a pathway for copying from the future prediction system's output $\mathbf{y}_{\mathrm{out}}$ to the past prediction system's input $\mathbf{z}_{\mathrm{in}}$ is prepared based on an analogy of auditory feedback in biological systems (a dashed arrow in Fig. 4(a)). When we are speaking, we are not only generating the voice, but also listening to it and always confirming the result of our speech. Therefore, people with congenital hearing defect are faced with a lot of difficulties because of the lack of this feedback loop. Even with normal vocal organs, it is quite difficult to acquire and maintain speaking skills in the absence of the feedback.

[2] In the bracket $\{\cdot\}$ of Eqs. (16) and (17), the first term corresponds to the element of direct backpropagating errors from the successive layer $\mathbf{y}^{[2]}$ and the second term is the element of recurrent backpropagating errors through the state layer $\mathbf{s}^{[\mathrm{F}]}$.

Similarly, the error function for the past prediction system is defined as

$$e_{\mathrm{p}} = \sum_{t} \sum_{i} \{[\mathbf{z}_{\mathrm{out}}(t)]_i - d_i^{[\mathrm{P}]}(t)\}^2, \tag{19}$$

and weight updates during the training phase for the past prediction system are also defined as follows:

$$\Delta v_{\mu\xi}^{[3]} = -\eta_{\mathrm{p}} \frac{\partial e_{\mathrm{p}}}{\partial u_{\mu}^{[z3]}} z_{\xi}^{[2]}, \tag{20}$$

$$\Delta v_{\mu\xi}^{[2]} = -\eta_{\mathrm{p}} \frac{\partial e_{\mathrm{p}}}{\partial u_{\mu}^{[z2]}} z_{\xi}^{[1]}, \tag{21}$$

$$\Delta v_{\mu\xi}^{[1]} = -\eta_{\mathrm{p}} \left\{ \frac{\partial e_{\mathrm{p}}}{\partial u_{\mu}^{[z1]}} z_{\xi}^{[0]} + \sum_{i} \frac{\partial e_{\mathrm{p}}}{\partial u_{i}^{[z1]}} v_{i\mu}^{[\mathrm{P}]} \left(1 + \tau \frac{\mathrm{d}}{\mathrm{d}t}\right)^{-1} [f_1'(u_{\mu}^{[z1]}) z_{\xi}^{[0]}] \right\}, \tag{22}$$

$$\Delta v_{\mu\xi}^{[\mathrm{P}]} = -\eta_{\mathrm{p}} \left\{ \frac{\partial e_{\mathrm{p}}}{\partial u_{\mu}^{[z1]}} s_{\xi}^{[\mathrm{P}]} + \sum_{i} \frac{\partial e_{\mathrm{p}}}{\partial u_{i}^{[z1]}} v_{i\mu}^{[\mathrm{P}]} \left(1 + \tau \frac{\mathrm{d}}{\mathrm{d}t}\right)^{-1} [f_1'(u_{\mu}^{[z1]}) s_{\xi}^{[\mathrm{P}]}] \right\}, \tag{23}$$

$$\Delta v_{\mu\xi}^{[\mathrm{F}]} = -\eta_{\mathrm{p}} \frac{\partial e_{\mathrm{p}}}{\partial u_{\mu}^{[z1]}} s_{\xi}^{[\mathrm{F}]}. \tag{24}$$

As a result of simplicity, as mentioned above, computation complexity for the bi-directional model in each training epoch is estimated at about twice $O(n^2)$, where n is the number of neurons in each subnetwork's hidden layer, and the number of the input and output neurons is assumed single, respectively.

Hereafter, these two kinds of operational modes are referred to as a future prediction mode and a past prediction mode, respectively.

3. Computer simulations for time series prediction

3.1. Training task and its procedure

Numerous neural network architectures and training algorithms for time series prediction tasks have been studied and various kinds of time series have been used to evaluate their performance. Among them, sunspots data[3] is one of the most popular datasets (Cholewo & Zurada, 1997a,b; Geva, 1998; Weigend et al., 1990) often used as a benchmark test. In this paper, these sunspots data are also adopted as one of the examples to discuss the effectiveness of the bi-directional

[3] Available from the web site at http://www.stern.nyu.edu/~aweigend/Time-Series/Data/Sunspots.Yearly

computation architecture. This is why a single time series makes it possible to use a simple network which has one input neuron and one output neuron, and few possibilities of influence by noisy factors make it easy to forecast future events. The first 100 years of the normalized annual data[4] for 280 years (AD 1700–1979) are used for training as follows.

- In the training phase for the future prediction system, $x(t)$ is applied as an input signal to $\mathbf{y}_{\text{in}}(t')$ and $x(t+a)$ is assigned as a teacher signal at $\mathbf{y}_{\text{out}}(t')$.
- In the training phase for the past prediction system, $x(t+a)$ is applied as an input signal to $\mathbf{z}_{\text{in}}(t')$ and $x(t)$ is assigned as a teacher signal at $\mathbf{z}_{\text{out}}(t')$.

Here, $x(t)$ is sunspots data at year t, a is the prediction step, and t' is the time step of computer simulations. Because a single time series (sunspots data) is used here, the input/teacher signal is provided to only the first neuron in the input/output layer.

In this paper, both subnetworks in the bi-directional model consist of four layers, and each of them has one, nine, nine and one neurons, respectively, not including the bias neuron. The time constant of the dynamic neurons is assumed to equal 10 times the minimum unit for time step ($\tau = 10$) in each computer simulation program. The initial weights are chosen as random numbers with a uniform distribution within $[-1, 1]$, and 10 different initial weight sets are used to avoid statistical bias (trial #1–10). Also, in order to eliminate any prior information bias from the initial state, 500 time-step free signal transmission is provided to converge to a stable point of the model's state space before presenting any patterns. The training procedure for the bi-directional model is performed by a batch process with sufficiently small learning rates ($\eta_{\text{f}} = \eta_{\text{p}} = 0.005$) to start the bi-directional model's training smoothly. Any choices larger than 0.005 induce the model to update weights radically at the beginning of the training phase, and it makes it difficult to recover from the rapid change of weights. The training procedure is repeated until the total square error of both systems (Eqs. (13) and (19)) becomes less than 0.2, equivalent to 0.2% error at every training point. If both total squared errors do not converge after a certain maximum number of training epochs, this trial is labeled as a 'failure' and further training is stopped. Each annual normalized sunspots number for both input data and teacher data is applied for five time steps ($\Delta t' = 5\Delta t$). During this five-time-step period, input data are applied constantly, while teacher data are applied only every fifth time step. This is why dynamic neurons in the bi-directional model produce hardly any rapid signal changes.

[4] A detailed procedure for normalization is described in Weigend et al. (1990).

After training is finished, each trained network's generalization performance is expressed by an index called an average relative variance (*ARV*) (Weigend et al., 1990) defined as

$$ARV = \frac{\sum_{t=1}^{T} \{x(t) - \hat{x}(t)^2\}}{\sum_{t=1}^{T} \{x(t) - \bar{x}\}^2}, \tag{25}$$

where $\{x(t)|t = 1, 2, \ldots, T\}$ indicates the true value set of the series (desired signals) and $\{\hat{x}(t)|t = 1, 2, \ldots, T\}$ denotes the predicted value set (actual output signals), and $\bar{x}$ is the average of $\{x(t)\}$. Eq. (25) can be rewritten with the variance σ^2 and the length T of the time series to be predicted as follows:

$$ARV = \frac{1}{\sigma^2 T} \sum_{t=1}^{T} \{x(t) - \hat{x}(t)\}^2. \tag{26}$$

This index can be seen as one of the universal measures of prediction quality. For example, perfect prediction is indicated by $ARV = 0$, while just average performance yields $ARV = 1$.

To clarify the explanations above, they are summarized again as follows.

- Training of each model is evaluated with the total squared error only involving the numerator of Eq. (25),
- Generalization performance is evaluated with the average relative variance shown in Eq. (25).

Each model's performance mentioned above is examined under the two kinds of test modes, one is for the future prediction system and the other is for the past prediction system. Hereafter, both of them are also referred to as a future prediction mode and a past prediction mode, respectively. Their signal flows are equivalent with the ones shown in Fig. 4.

To compare the performance of the uni-directional model with the performance of the bi-directional model, the future and past prediction systems are trained and examined independently. The future prediction system with twice the number of hidden neurons, equivalent to the total number of hidden neurons in the bi-directional model, is also tried because the number of hidden neurons has a close relationship to the model's signal processing abilities. Although the number of weights also affect the model's performance, the size of the hidden layer has been chosen as identical in this paper because of simplicity. For reference, each number of all models used in this study is summarized in Table 1.

3.2. Training epoch number to be required

At the beginning of the computer simulation, a number of required training epochs is considered. It is generally said that a training curve decays exponentially with the length of

Table 1
Number of hidden neurons and weights including bias neurons

Item	Bi-directional model			Uni-directional model		
	Future pred. system	Past pred. system	Total	Future pred. system	Future pred. system (II)[a]	Past pred. system
Architecture	1–9–9–1	1–9–9–1	–	1–9–9–1	1–18–18–1	1–9–9–1
Neuron	29	29	58	29	56	29
Bias neuron	3	3	6	3	3	3
Total	32	32	64	32	59	32
Weights	261	261	522	180	684	180
Bias connection[b]	19	19	38	19	37	19
Total	280	280	560	199	721	199

[a] Future prediction system with 18 neurons in each hidden layer.
[b] Connection from the bias neuron.

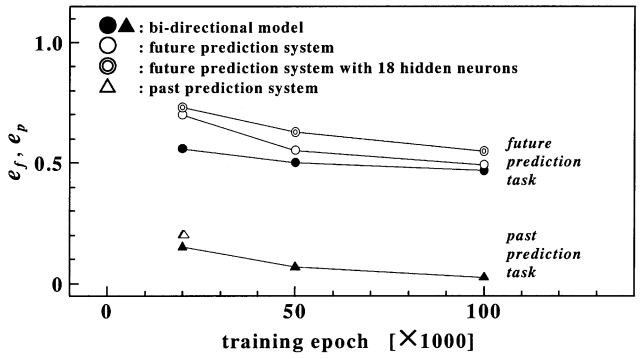


Fig. 5. Change of the total squared error e_f defined by Eq. (13) and e_p by Eq. (19) observed in each model. All trials of the past prediction system (△) finished training successfully before 20,459 epochs.

the training period. Training periods of 20,000, 50,000 and 100,000 epochs are provided for each model in this paper. The simplest task such as single-year prediction ($a = 1$) is used to confirm this requirement. Throughout the training period, all trials except for the past prediction system train unsuccessfully. Fig. 5 shows their total squared errors averaged over each 10 trials. According to this figure, there are no significant improvements observed after 20,000-epoch training. In particular, the results trained for 100,000 epochs are almost the same as those trained for 50,000. Therefore, it seems to be sufficient to train each model for 20,000 epochs, or for 50,000 epochs including some margin to assure convergence to a minimum point in the error space. Of course, it is also important to consider each model's computational load. For instance, the bi-directional model's is about twice as much as either the future prediction system's or the past prediction system's and about half as much as the one of the future prediction system with 18 neurons in each hidden layer, following the estimation of computation complexity mentioned in the last part of Section 2.2. However, allowing for such difference on computational load, there are no significant improvements observed after 20,000-epoch training. Moreover, the prediction quality produced by each trained network, which is shown in Section 3.4, is more important than the computational time from the viewpoint of practical use for time series prediction. An example of each model after 20,000-epoch training is shown in Table 2.

3.3. Performance comparison of four different models

According to Table 2, only the past prediction system can

Table 2
Results of computer simulations for the single-year prediction task. All decimals in this table indicate the total squared errors e_f, e_p in each model after 20,000-epoch training

Trial #	Bi-directional model		Uni-directional model		
	Future pred. system	Past pred. system	Future pred. system	Future pred. system (II)	Past pred. system (iter.#)
1	0.597	0.171	0.734	0.730	0.200 (16,482)
2	0.566	0.182	0.693	0.742	0.205
3	0.563	0.124	0.681	0.721	0.200 (15,428)
4	0.531	0.140	0.717	0.682	0.200 (16,064)
5	0.555	0.124	0.645	0.725	0.200 (18,653)
6	0.609	0.175	0.731	0.722	0.200 (14,713)
7	0.509	0.167	0.718	0.703	0.200 (15,342)
8	0.537	0.138	0.701	0.797	0.200 (12,770)
9	0.546	0.121	0.663	0.763	0.200 (16,082)
10	0.576	0.161	0.706	0.709	0.200 (11,056)
Average	0.559	0.150	0.699	0.729	0.201 (–)
Rate	0/10		0/10	0/10	9/10

learn this single-year prediction task successfully. It is obvious that the future prediction task is more difficult to train than the past prediction one. Because the desired past prediction system's output $\mathbf{z}_{\text{out}}(t')$ is the same as the past prediction system's input $\mathbf{z}_{\text{in}}(t'-5)$, where five time steps are equivalent to one year ($\Delta t' = 5\Delta t$), the past prediction task is quite easy to train. On the contrary, the desired future prediction system's output $\mathbf{y}_{\text{out}}(t')$ is not the same as any future prediction system's input applied before, the future prediction task is not so easy to train. In spite of this poor score on training for time series prediction, the comparison of the averaged total error for the bi-directional computation style (the second column) and that for the traditional uni-directional one (the fourth column) manifests that the proposed architecture performs better than the conventional one which estimates only the future value from the present and the past information. In order to clarify the improvement achieved by the bi-directional architecture, an index for improvement quality (*IIQ*) is defined as follows:

$$IIQ = \frac{[e]_{\text{bi-direct.comp.}}}{[e]_{\text{uni-direct.comp.}}} = \frac{[ARV]_{\text{bi-direct.comp.}}}{[ARV]_{\text{uni-direct.comp.}}}, \tag{27}$$

where e is the total squared error defined by Eqs. (13) or (19), and *ARV* is the average relative variance defined by Eq. (26). Positive bi-directionalization effect is for $IIQ < 1$, while negative is manifested by $IIQ \geq 1$. With this index, the bi-directionalization effect in the future prediction task is estimated from the average data in the second and the fourth columns of Table 2 as follows:

$$IIQ_f = \frac{[\bar{e}_f]_{\text{bi-direct.model}}}{[\bar{e}_f]_{\text{future pred.system}}} = \frac{0.559}{0.699} \approx 0.800. \tag{28}$$

In general, a trial number in Table 2 indicates the trial which starts training from the same initial weights. The model denoted as 'future prediction system (II)' containing 18 hidden neurons cannot provide the same initial state because its structure is completely different and does not correspond to the other three nine-hidden-neuron models. As seen from Table 2, the future prediction system with 18 hidden neurons (the fifth column) cannot match the performance of the bi-directional model. Therefore, this 18-hidden-neuron model is not described again in the rest of this paper.

As the result of this discussion, it can be said that future–past information integration in the bi-directional model improves the model's performance.

3.4. Performance of the bi-directional neural network model

In order to fully exploit and understand the bi-directional model for time series prediction it is important to investigate the model's behavior. First, responses of the input and output neurons in the bi-directional model are shown in Fig. 6 (trial #1 in Table 2). The upper figure shows the responses in the future prediction mode and the lower one is in the past prediction mode. In both figures, the applied time series is identical with that used for training and every filled circle indicates the training point. In the future prediction mode, a sawtooth waveform at the future prediction system's output layer (thick black line) is particularly noticeable. According to responses of all hidden neurons in the bi-directional model, which is not shown here for brevity, sharpness of the sawtooth waveform becomes greater as the signal proceeds to the final layer in the future prediction system. Although the reason for this sharp sawtoothed wave is not clear, possible explanations are:

1. the bi-directional model tries to keep past information to predict future events; and
2. the dynamic neurons in the bi-directional model have only a relatively small time constant and they lose their past information rapidly.

To fulfill the requirement (1) and condition (2) simultaneously, the bi-directional model can do nothing but generate a large amplitude output to offset attenuation in advance. On the contrary, there are no sawtooth waveforms in the past prediction mode because the bi-directional model does not have to preserve its past information so much.

Secondly, responses of all state neurons which play an important role in time series processing for the future and/or past prediction tasks are shown in Figs. 7 and 8. They consist of two subfigures and show

responses of all state neurons $s_i^{[\text{F}]}$ ($i = 1, 2, \ldots, 9$) in the future prediction system,
responses of all state neurons $s_i^{[\text{P}]}$ ($i = 1, 2, \ldots, 9$) in the past prediction system,

in the future prediction mode and the past prediction one, respectively. A signal flow in each mode is equivalent to that shown in Fig. 4. According to these two figures, responses of all state neurons are almost the same in spite of the bi-directional model's operational mode. Then, similarity on the bi-directional model's response enables production of cooperative actions, namely, the *past* prediction system's output $\mathbf{z}_{\text{out}}$ is quite similar to the *future* prediction system's input $\mathbf{y}_{\text{in}}$ in the *future* prediction mode (Fig. 6(a)), and vice versa (Fig. 6(b)). These facts imply that prediction of future values is executed as comparing its validity with the only acquired transformation matrices, not using any actual future information but feedback signal from the future prediction system's output layer, for past prediction. Therefore, it can be said that this future–past information integration helps the bi-directional model to predict future points accurately and it results in a better performance than the conventional uni-directional model.

To compare with Fig. 6, responses of the future prediction system and the past independently trained one are also investigated. Generally speaking, their behavior is quite similar to those of the corresponding subnetwork in the bi-directional model shown in Fig. 6(a, b).

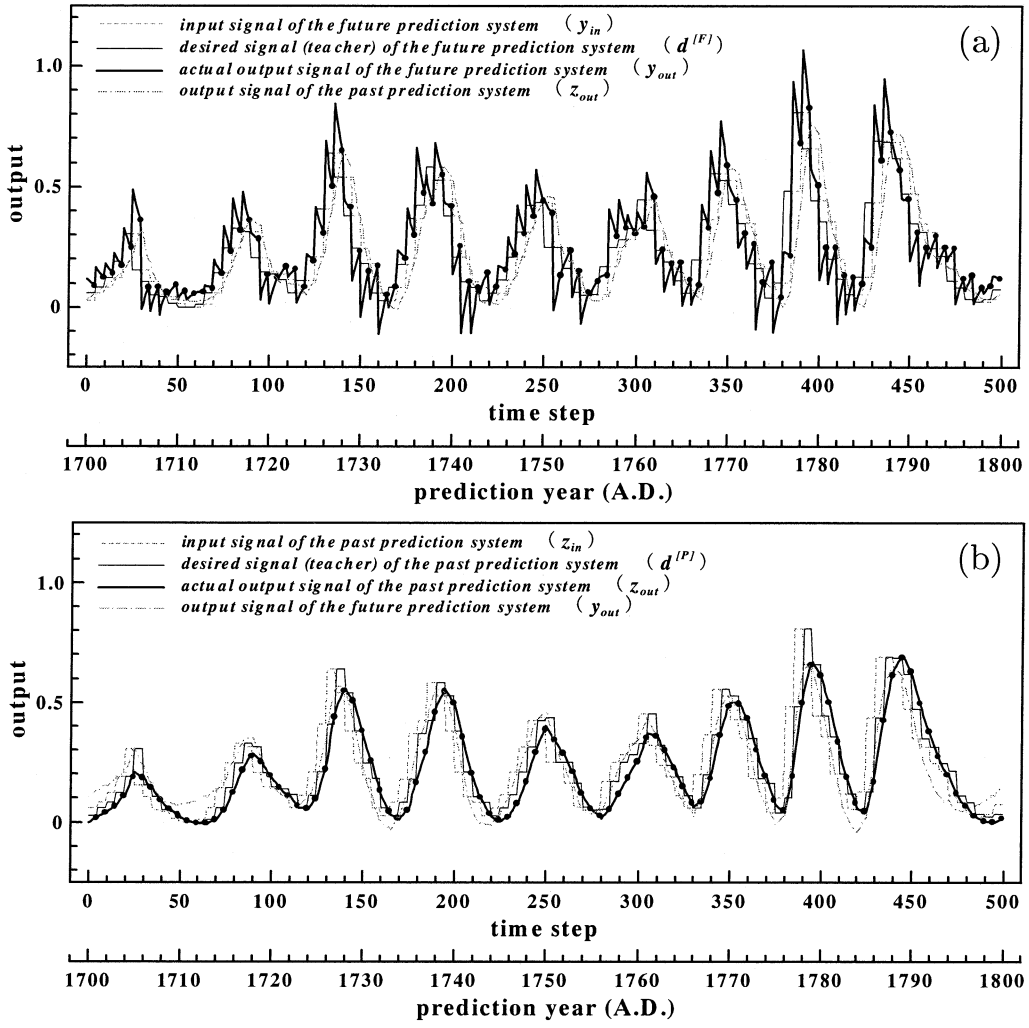


Fig. 6. Responses of the input and output neurons in the bi-directional neural network model after 20,000-epoch training. Trial #1 in Table 2 (a) Future prediction mode; (b) Past prediction mode.

Fig. 9 shows the prediction quality produced by the trained networks. The test data for these experiments are described in Table 3. Most of them are a completely new data set but some of them partially contain a training data set. According to this figure, the past prediction tasks (▲, △) indicate a good score, but the future prediction tasks (●, ○) are not satisfactory. Nevertheless, its performance is improved by future–past information integration in the bi-directional model (●) compared with the traditional uni-directional one (○).

By the way, it is said generally that longer training period contributes to better performance unless overtraining effects occur. Of course, it is important to consider such effects to achieve good results for forecasting future events. The model tries to fit gradually into the major features and then to the minor features, and in this study, good scores on prediction quality are confirmed with the test data sets and can be seen from Fig. 9. Then, there is no evidence of overtraining effects in these trained networks.

3.5. Dependency against size of prediction year ahead

So far, only the simplest task of single-year prediction ($a = 1$) has been dealt with. In this subsection, multi-year prediction tasks ($a > 1$) are tried. Table 4 shows a summary of computer simulation results after 20,000-epoch training. According to this table, all trials except the past prediction system for one- and two-year prediction have failed. One of major reasons for this training difficulty is caused by the existence of the dynamic neurons in the bi-directional model. They form a short-term memory system to keep the past information in the model, but it can be seen from Eqs. (3) and (8), that their output decays exponentially with time, so it becomes difficult to sustain the information for very long. Fig. 10 shows the result of multi-year prediction performance produced by the trained networks. In general, the score of the future prediction task (●, ○) is worse than that of the past prediction task (▲, △). Nevertheless, future prediction with the bi-directional computation method (●)

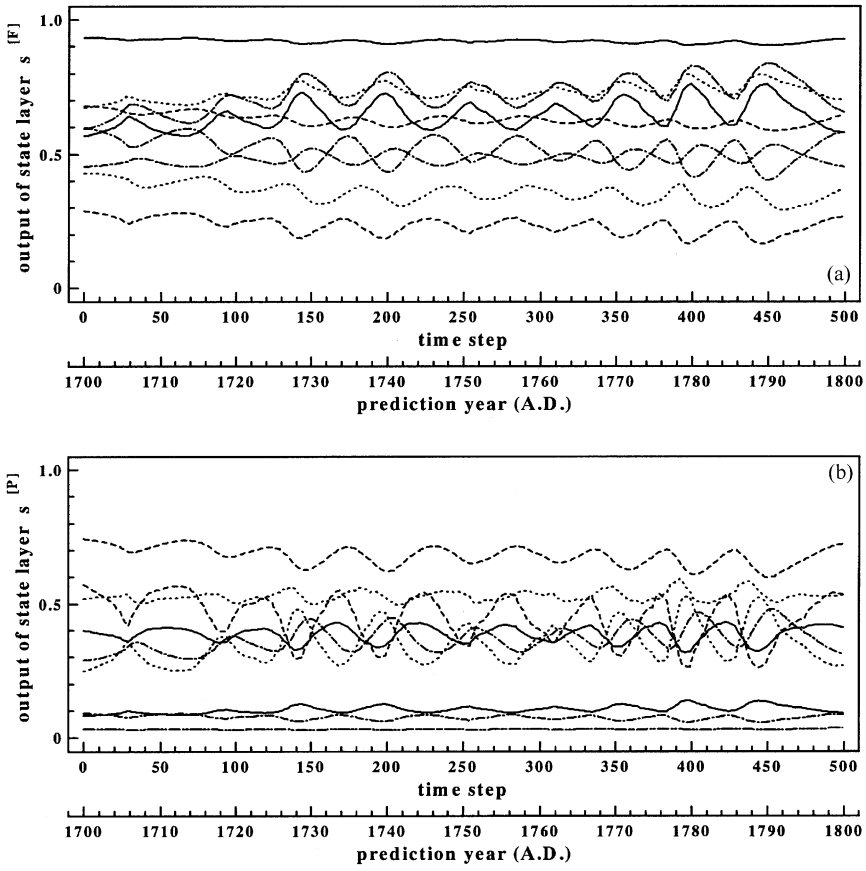


Fig. 7. Responses of the all state neurons in the bi-directional model during the *future* prediction mode (a) Responses in the future prediction system $s^{[F]}$; (b) Responses in the past prediction system $s^{[P]}$.

performs better than that with the conventional uni-directional computation method (○). It suggests that the developed transformation matrices for past prediction help to improve the future prediction performance of the bi-directional model through the future–past information integration. However, this improvement vanishes as the training difficulties of the past prediction task increase. According to this figure, it can be seen that two-year prediction is the critical one for this series of computer simulations using sunspots data.

For reference, responses of the bi-directional model trained in each task are described below. In the two-year prediction task, responses of the output neurons in both the future and the past prediction systems contain sharp sawtooth waveforms. Following the discussions in Section 3.4, the emergence of the sawtooth waveforms means the increase of training difficulties so that it can be considered that the bi-directional model is approaching the critical point to accurately predict the future value.

Figs. 11–13 show three-, five- and 10-year prediction tasks, respectively, and indicate unsatisfactory responses. All conditions are the same as those in Fig. 6. The 10-year prediction task (Fig. 13) seems to produce normal responses, but all responses contain one-cycle lag. The sunspots data set is roughly a cyclic series with a period of approximately 10 years. Therefore, prediction 10 years ahead can be seen as just an identity transformation from the viewpoint of each model, because the model tries to find any relationships between the input and the teacher patterns based only on their similarity, and it tries to minimize their distance. This is caused by the nature of dynamic neurons whose information content decays with time, namely, the more recent information outweighs the previous one. It is quite natural for each model to try to develop such an identity transformation, but the efforts always end in failure because there are no meaningful relationships in it. Both dull peaks and vibrations observed in the model's responses can be seen as the results of this meaningless training. In the five-year prediction task (Fig. 12), the model can barely produce periodical signals with the desired phase shift, which is equivalent to a half cycle of sunspots data series. This fact prevents the model from finding the correct correspondence between the input pattern and the teacher pattern, so that it fails to acquire the appropriate amplitude of the responses. In the three-year prediction task (Fig. 11), the model can produce responses similar to those developed in the two-year prediction task, but such responses do not pass slightly the criterion mentioned in Section 3.1, i.e. total squared errors defined by Eqs. (13) and (19) do not converge

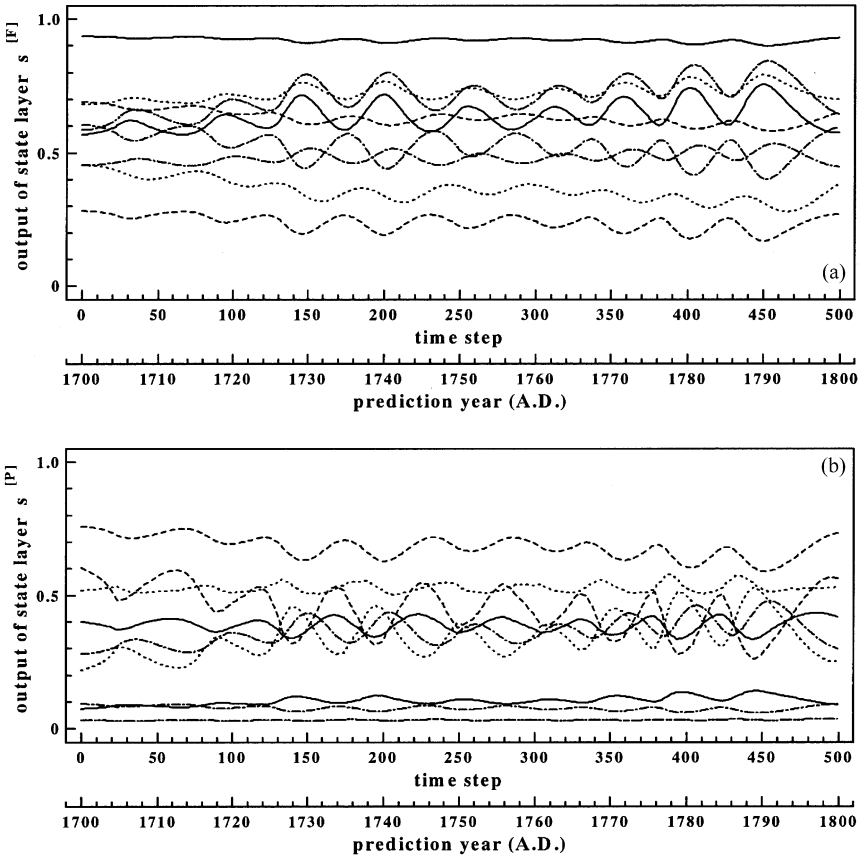


Fig. 8. Responses of the all state neurons in the bi-directional model during the *past* prediction mode (a) Responses in the future prediction system $s^{[F]}$; (b) Responses in the past prediction system $s^{[P]}$.

less than 0.2 after 20,000-epoch training, so that this task is labeled as a failure.

3.6. Dependency against time constant of dynamic neurons

Throughout the series of computer simulations on multi-year prediction tasks, a two-year prediction task is critical to successful training. As mentioned above, time series processing greatly depends on the nature of dynamic neurons defined by the time constant τ (Eqs. (3) and (8)). Of course, the larger the value, the longer the past information can be preserved. Then, using a larger time constant of the dynamic neurons seems to be one of the solutions when dealing with multi-year prediction tasks. To confirm its observation, another computer simulation for a three-year prediction task has been made. According to Table 5 showing the summary of simulations, the performance has changed for the worse. It is true that a larger time constant of the dynamic neurons makes it possible to sustain the past information, but at the same time it also makes it difficult to process and accept the present information. In other

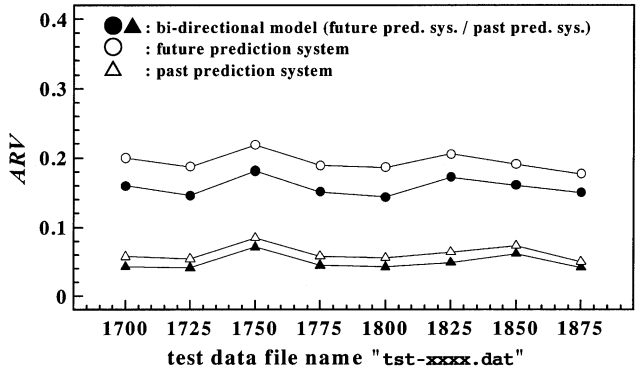


Fig. 9. Prediction quality with the trained networks.

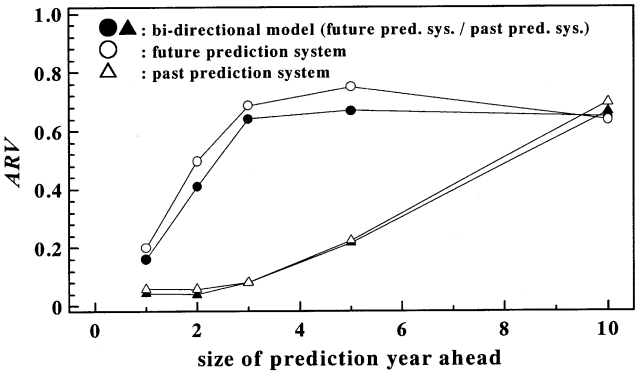


Fig. 10. Results of multi-year prediction using the test data 'tst-1700.dat' after 20,000-epoch training.

Table 3
Test data sets for Fig. 9

File name	Period		Training portion
	Begin–end yr	1700----1750----1800----1850----1900----1950----2000	
tst-1700.dat	1700–1799	**********	100%
tst-1725.dat	1725–1824	**********	75%
tst-1750.dat	1750–1849	**********	50%
tst-1775.dat	1775–1874	**********	25%
tst-1800.dat	1800–1899	**********	0%
tst-1825.dat	1825–1924	**********	0%
tst-1850.dat	1850–1949	**********	0%
tst-1875.dat	1875–1974	**********	0%

The test data "tst–1700.dat" in the first row is equivalent to the training data.

Table 4
Results of multi-year prediction ahead after 20,000-epoch training

Prediction size	Bi-directional model			Uni-directional model			
	Future pred. system	Past pred. system	(Rate)	Future pred. system	(Rate)	Past pred. system	(Rate)
1-year[a]	0.559	0.150	(0/10)	0.699	(0/10)	0.201	(9/10)
2-year	1.417	0.133	(0/10)	1.722	(0/10)	0.200	(10/10)
3-year	2.207	0.285	(0/10)	2.365	(0/10)	0.281	(0/10)
5-year	2.291	0.759	(0/10)	2.572	(0/10)	0.789	(0/10)
10-year	2.235	2.337	(0/10)	2.200	(0/10)	2.447	(0/10)

[a] Results of 1-year prediction task are the same as those in Table 2.

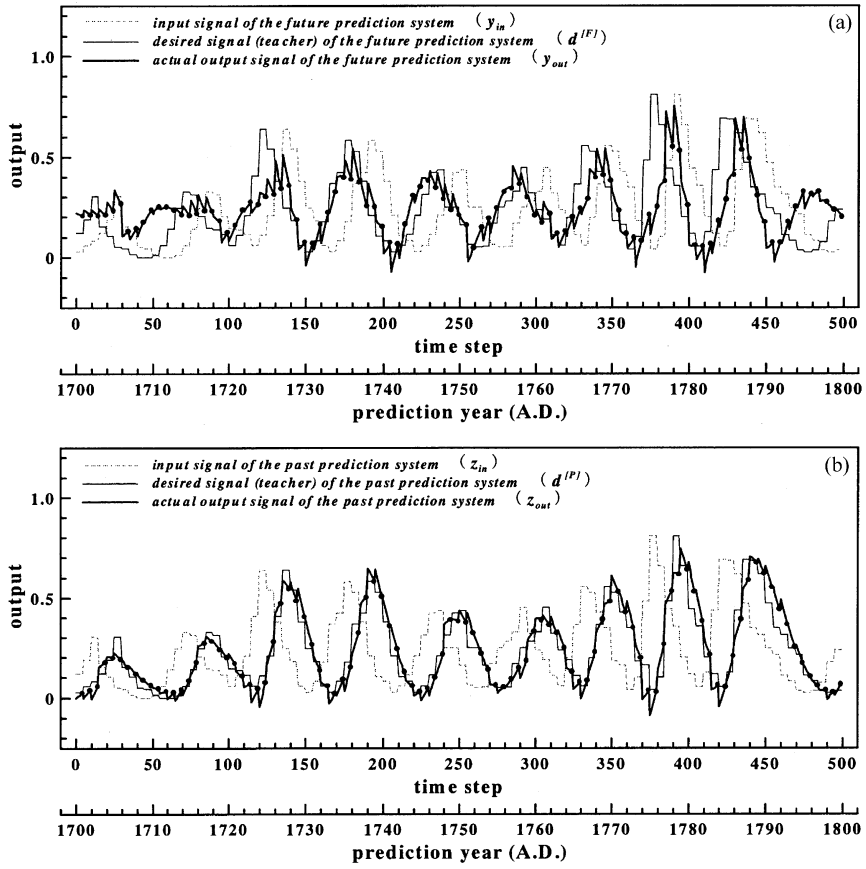


Fig. 11. Responses of the bi-directional model in *three*-year prediction task. Trial #1 in Table 4 (a) Responses in the future prediction system $s^{[F]}$; (b) Responses in the past prediction system $s^{[P]}$.

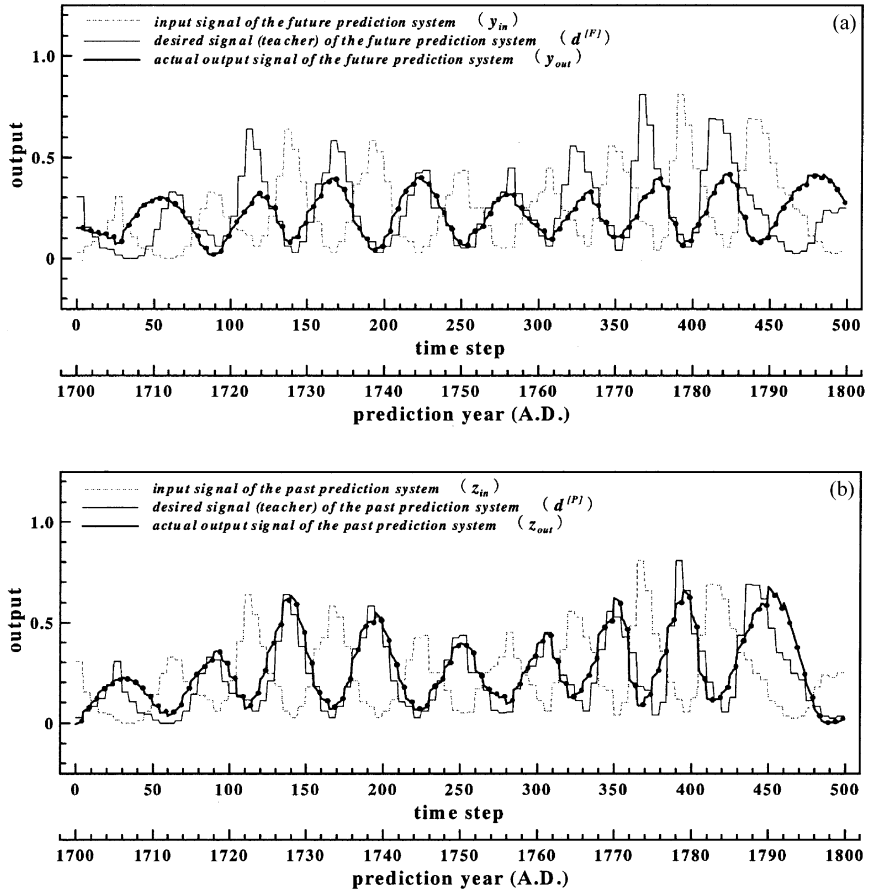


Fig. 12. Responses of the bi-directional model in *five*-year prediction task. Trial #1 in Table 4 (a) Responses in the future prediction system $s^{[F]}$; (b) Responses in the past prediction system $s^{[P]}$.

words, a smaller time constant makes the model easy to process and accept the present information, but it makes it difficult to sustain the past information to predict future values. Therefore, increasing the time constant of the dynamic neurons does not improve the network's quality. Actually, a response of the future prediction system in the bi-directional model is almost constant and there are no signs of state transitions.

Another possibility to improve the model's trainability is to adopt different types of dynamic neurons as a short-term memory system of the bi-directional model. According to the classification by Mozer (1993), there exist other types of neural network memories. Among them, Gamma memory (De Vries & Principe, 1992) has the property on 'high depth and high resolution'. So, it is quite conceivable that using other memory could result in improved performance for multi-year prediction tasks if their appropriate parameters, such as the depth and the resolution, could be found.

4. Discussion

The focus of this paper has been to ascertain whether the model's performance for time series prediction is improved by introducing the future–past information integration. A number of computer simulations focused on the comparison of the proposed technique of bi-directional computation and the conventional technique of uni-directional computation. Some results in this study may not indicate scores better than those examined by other researchers with a variety of useful methods such as cross-validation, but this does not mean that the proposed technique is worse than other techniques. The fact that the proposed *bi-directional* computing technique is better than the conventional *uni-directional* technique under the plain condition, without any additional methods, is confirmed. Therefore, further studies of bi-directional computational architecture with these useful methods such as cross-validation, applying various network size, the variable learning rate, the proper selection of initial weights, and the sufficient length of a training period will be required.

The basic idea of bi-directional computation originally comes from a biological nervous system. Such a system consists of both a motor control system and a sensory reception system. They are working cooperatively to perform various kinds of complex activities such as

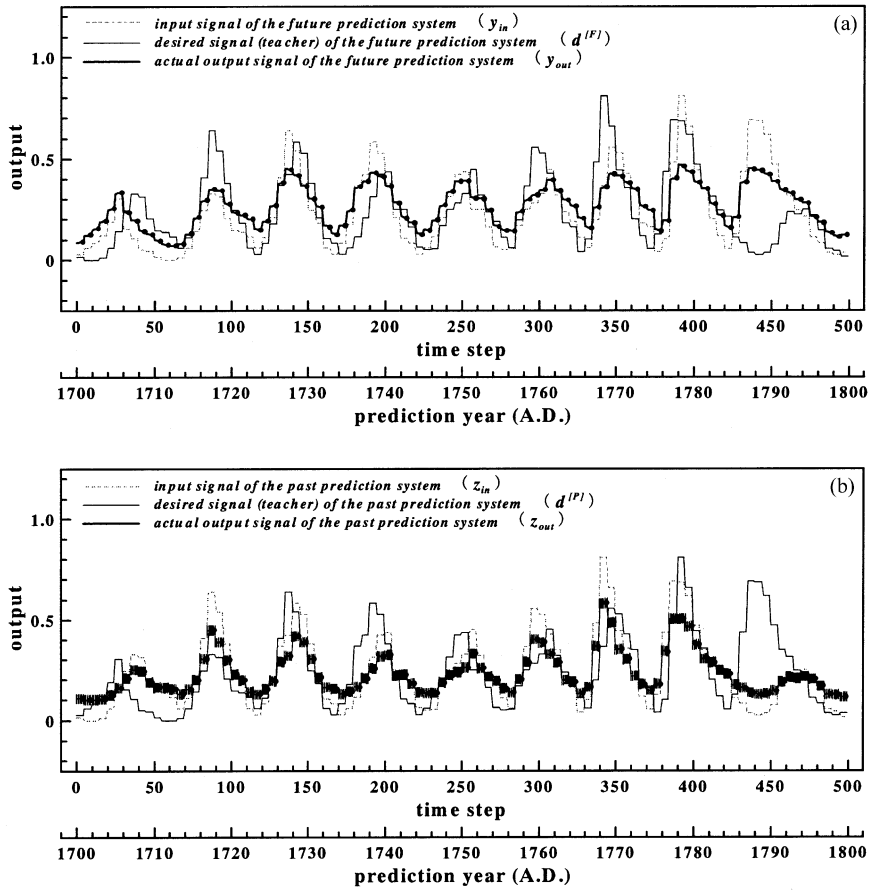


Fig. 13. Responses of the bi-directional model in *10*-year prediction task. Trial #1 in Table 4 (a) Responses in the future prediction system $s^{[F]}$; (b) Responses in the past prediction system $s^{[P]}$.

Table 5
Total squared errors e_f, e_p for different time constants after 20,000-epoch training

Time constant	Bi-directional model			Uni-directional model			
	Future pred. system	Past pred. system	(Rate)	Future pred. system	(Rate)	Past pred. system	(Rate)
10[a]	2.207	0.285	(0/10)	2.365	(0/10)	0.281	(0/10)
50	3.348	0.339	(0/10)	3.437	(0/10)	0.454	(0/10)
100	3.336	0.778	(0/10)	3.360	(0/10)	1.463	(0/10)

[a] Results of time constant $\tau = 10$ are the same as those of 3-year prediction in Table 4.

motor control, speech perception, visual recognition, and so on (Lee, 1950; Liberman, Delattre & Cooper, 1952). Quite often we feel it is easier to memorize some words, when using other assisting organs in both motor and sensory systems. Making use of the coupling effects turns out to be a very effective means of acquiring various tasks. In addition, it has been observed that some disorders in the sensorimotor cooperation cause a lot of difficulties (Lee, 1950). There have been reports on coupling effects in biological models (Wakuya & Shida, 1997, 1998, 1999; Wakuya et al., 1994), but there are few studies on application-oriented tasks such as the one discussed in this paper. Therefore, it is quite valuable to note such a viewpoint from a biological system even if some risk of ending in failure remains.

5. Conclusions

In this paper, the concept of bi-directional computation is applied to time series prediction tasks. One of its major features is an improvement of the model's performance based on direct and inverse signal transformations and their coupling effects. Through computer simulations using sunspots data, it has been confirmed that the prediction quality of the bi-directional model is better than the conventional uni-directional one. The uni-directional model corresponds to a subnetwork in the bi-directional model, trained independently. As a result, it can be said that future–past information integration improves the model's performance, especially in the future prediction

task. Multi-year prediction tasks have also been investigated, but only two-year ahead prediction has yielded accurate results because of the nature of dynamic neurons used in the proposed bi-directional neural network model.

Acknowledgements

The authors would like to thank Dr Tomasz J. Cholewo for his helpful comments and suggestions. This work was sponsored in part by the Systems Research Institute of the Polish Academy of Sciences, 01-447 Warszawa, Poland, ul. Newelska 6.

References

Cholewo, T., & Zurada, J. M. (1997). Neural network tools for stellar light prediction. In *Proceedings of the IEEE Aerospace Conference, 3, Snowmass, CO* (pp. 415–422).

Cholewo, T., & Zurada, J. M. (1997). Sequential network construction for time series prediction. In *Proceedings of the IEEE International Joint Conference on Neural Networks, Houston, TX* (pp. 2034–2039).

De Vries, B., & Principe, J. C. (1992). The gamma model —a new neural model for temporal processing. *Neural Networks*, *5*, 565–576.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Gershenfeld, N. A. & Weigend, A. S. (1993). The future of time series. In A. S. Weigend & N. A. Gershenfeld (Eds.), *Time series prediction: forecasting the future and understanding the past. Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis held in Santa Fe, NM, May 14–17* (pp. 1–70). Reading, MA: Addison-Wesley.

Geva, A. B. (1998). ScaleNet—multiscale neural-network architecture for time series prediction. *IEEE Transactions on Neural Networks*, *9*, 1471–1482.

Gorman, R. P., & Sejnowski, T. J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, *1*, 75–89.

Ishikawa, M. (1996). Structural learning with forgetting. *Neural Networks*, *9*, 509–521.

Jordan, M. L. (1986). *Serial order: a parallel distributed processing approach*. ICS Report 8604, Institute for Cognitive Science, University of California, San Diego.

Kleinfeld, D. (1986). Sequential state generation by model neural networks. *Proc. Natl. Acad. Sci. USA*, *83*, 9469–9473.

Le Cun, Y., Denker, J. S., & Solla, S. A. (1990). Optimal brain damage. In D. S. Touretzky, *Advances in neural information processing systems* (pp. 598–605), vol. 2. San Mateo, CA: Morgan Kaufmann.

Lee, B. S. (1950). Effect of delayed speech feedback. *Journal of the Acoustical Society of America*, *22*, 824–826.

Liberman, A. M., Delattre, P., & Cooper, F. S. (1952). The role of selected stimulus-variables in the perception of the unvoiced stop consonants. *American Journal of Psychology*, *65*, 497–516.

Moody, J., & Utans, J. (1995). Architecture selection strategies for neural networks: application to corporate bond rate prediction. In A. -P. Refenes, *Neural networks in the capital markets* (pp. 277–300). Chichester, UK: Wiley.

Mozer, M. C. (1993). Neural net architectures for temporal sequence processing. In A. S. Weigend & N. A. Gershenfeld (Eds.), *Time series prediction: forecasting the future and understanding the past. Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis held in Santa Fe, NM, May 14–17* (pp. 243–264). Reading, MA: Addison-Wesley.

Mozer, M. C., & Smolensky, P. (1989). Using relevance to reduce network size automatically. *Connection Science*, *1*, 3–17.

Saad, E. W., Prokhorov, D. V., & Wunsch II, D. C. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks*, *9*, 1456–1470.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*, 2673–2681.

Sejnowski, T. J., & Rosenberg, C. R. (1987). Parallel networks that learn to pronounce English text. *Complex Systems*, *1*, 145–168.

Sietsma, J., & Row, R. J. F. (1988). Neural net pruning—why and how. In *Proceedings of International Conference on Neural Networks, San Diego, CA* (vol. 1, pp. 325–333).

Wakuya, H., & Shida, K. (1997). An integrated model for motor control and sensory reception with a bi-directional neural network model (in Japanese). *Transactions of the Institute of Electronics, Information and Communication Engineers*, *D-II* (J80 D-II), 1929–1938.

Wakuya, H., & Shida, K. (1998). A study on an integrated neural network model of motor control and sensory reception systems. In *Proceedings of the Second International Conference on Knowledge-Based Intelligent Electronic Systems, vol. 3, Adelaide, Australia* (pp. 497–504).

Wakuya, H., & Shida, K. (1999). Acquired sensorimotor coordinated signal transformation in a bi-directional neural network model. In *Proceedings of 1999 International Joint Conference on Neural Networks, CD-ROM, Washington, DC*.

Wakuya, H., Futami, R., & Hoshimiya, N. (1994). A bi-directional neural network model for generation and recognition of temporal patterns (in Japanese). *Transactions of the Institute of Electronics, Information and Communication Engineers*, *D-II* (J77-D-II), 236–243.

Wan, E. A. (1993). Time series prediction by using a connectionist network with internal delay lines. In A. S. Weigend & N. A. Gersenfeld (Eds.), *Time series prediction: forecasting the future and understanding the past. Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis held in Santa Fe, NM, May 14–17* (pp. 195–217). Reading, MA: Addison-Wesley.

Weigend, A. S., Huberman, B. A., & Rumelhart, D. E. (1990). Predicting the future: a connectionist approach. *International Journal of Neural Systems*, *1*, 193–209.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*, 270–280.