Fifth IFSA World Congress (1993), 1013-1016

# An Input-correlated Neuron Model and Its Learning Characteristics

**Takeshi Yamakawa*†, Toru Aonishi**, Eiji Uchino*† and Tsutomu Miki*†**

* Department of Control Engineering and Science
Kyushu Institute of Technology, Iizuka, Fukuoka 820, Japan

** Department of Biophysical Engineering
Osaka University, Toyonaka, Osaka 560, Japan

***Abstract*** This paper describes a new type of neuron model, the inputs of which are interfered with one another. It has a high mapping ability with only single unit. The learning speed is considerably improved compared with the conventional linear type neural networks. The proposed neuron model was successfully applied to the prediction problem of chaotic time series signal.

***Key words:*** input-correlated neuron, high-speed learning, prediction of chaotic signal, system identification

## 1. Introduction

The multilayer neural networks have been applied to a number of problems in many areas other than an engineering area. However, there are some problems such as: (i) there are no reasonable design policies for the network structures, e.g. number of layers, number of units in each hidden layer, and so on, (ii) it takes much time for learning with use of error back-propagation type algorithm, (iii) the gradient descent scheme can become stuck in local minima of the error function, and so on.

The purpose of this paper is to propose a new type of neuron model, i.e. input-correlated neuron, which has a high mapping ability with only single unit. It is proved mathematically that there exist no local minima for the single-layer network of the proposed neurons (i.e. with no hidden layers). The computer simulations tell that the learning speed is considerably improved compared with the conventional linear type neural networks. The single unit of the proposed neuron model can solve the XOR problem or more complex problems with linearly nonseparable patterns, which are impossible with use of the conventional linear type neural networks with no hidden layers.

†The authors are also with the Fuzzy Logic Systems Institute (FLSI), Iizuka, Fukuoka 820, Japan.

Finally, the proposed input-correlated neuron model has been successfully applied to the prediction problem of future time history of signal, which shows chaotic behavior. The computer simulation results are quite well and promising.

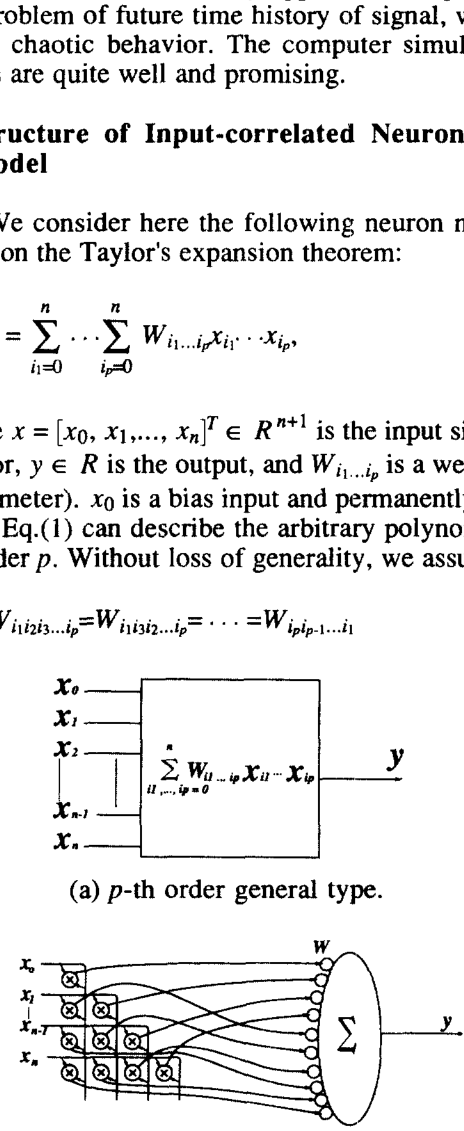
## 2. Structure of Input-correlated Neuron Model

We consider here the following neuron model based on the Taylor's expansion theorem:

$$y = \sum_{i_1=0}^{n} \cdots \sum_{i_p=0}^{n} W_{i_1 \ldots i_p} x_{i_1} \cdots x_{i_p}, \tag{1}$$

where $x = [x_0, x_1, \ldots, x_n]^T \in R^{n+1}$ is the input signal vector, $y \in R$ is the output, and $W_{i_1 \ldots i_p}$ is a weight (parameter). $x_0$ is a bias input and permanently set to 1. Eq.(1) can describe the arbitrary polynomial of order $p$. Without loss of generality, we assume:

$$W_{i_1 i_2 i_3 \ldots i_p} = W_{i_1 i_3 i_2 \ldots i_p} = \cdots = W_{i_p i_{p-1} \ldots i_1} \tag{2}$$



(a) $p$-th order general type.

(b) Second order type.

Fig.1. Structure of an input-correlated neuron model.

for all the combinations of $(i_1, i_2, \ldots, i_p)$.

If it is confined up to the second order polynomials, Eq.(1) has the following quadratic form:

$$y = x^T W x, \tag{3}$$

where $W$ is the weight matrix that is symmetrical. The structure of this input-correlated neuron model is shown in Fig.1.

## 3. Learning Algorithm of Input-correlated Neuron

Fig.2 shows the general multilayer feedforward network of the proposed neurons. The input-output relation for the $k$-th neuron in layer $n$ is described by:

$$y_k^n = \sum_{i_1=0}^{M_{n+1}} \cdots \sum_{i_p=0}^{M_{n+1}} W^{n\,k}_{i_1 \ldots i_p} y^{n+1}_{i_1} \cdots y^{n+1}_{i_p}, \tag{4}$$

where $M_n$ is the number of units in layer $n$, $W^{n\,k}_{i_1 \ldots i_p}$ is the weight assigned, and $y_k^n$ is the output of the $k$-th neuron in layer $n$.

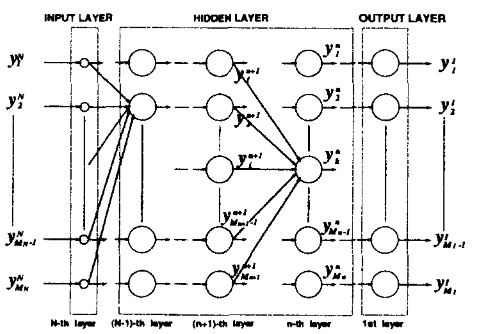



Fig.2. Multilayer feedforward network of the input-correlated neurons.

Here we define the error to be minimized over the training cycle as follows:

$$V = \frac{1}{2}\sum_{q=1}^{r} \| y^1(q) - \hat{y}(q) \|^2$$
$$= \frac{1}{2}\sum_{q=1}^{r}\sum_{k=1}^{M_1} \left(y_k^1(q) - \hat{y}_k(q)\right)^2 = \sum_{q=1}^{r} v(q), \tag{5}$$

where $y^1(q) = \left[y_1^1(q), \ldots, y_{M_1}^1(q)\right]^T$ is the output of the network, and $\hat{y}(q) = \left[\hat{y}_1(q), \ldots, \hat{y}_{M_1}(q)\right]^T$ $(q=1,\ldots,r)$ is the training data.

The gradient descent scheme [1] is employed here to reduce the error $V$ through the adjustment of weight $W^{n\,k}_{i_1 \ldots i_p}$. That is, the dynamics of weight is described by:

$$\frac{dW^{n\,k}_{i_1 \ldots i_p}}{dt} = -\frac{\partial V}{\partial W^{n\,k}_{i_1 \ldots i_p}}. \tag{6}$$

Now, the derivative of $V$ with respect to $W^{1\,k}_{i_1 \ldots i_p}$ in layer 1 becomes:

$$\frac{\partial V}{\partial W^{1\,k}_{i_1 \ldots i_p}} = \sum_{q=1}^{r} \frac{\partial v(q)}{\partial y_k^1(q)} \cdot \frac{\partial y_k^1(q)}{\partial W^{1\,k}_{i_1 \ldots i_p}}. \tag{7}$$

Then, the second term of the right hand side of Eq.(7) is calculated from Eq.(4) as:

$$\frac{\partial y_k^1(q)}{\partial W^{1\,k}_{i_1 \ldots i_p}} = y^2_{i_1}(q) \cdots y^2_{i_p}(q). \tag{8}$$

Now, by putting:

$$r_k^1(q) \equiv \partial v(q)/\partial y_k^1(q) = y_k^1(q) - \hat{y}_k(q), \tag{9}$$

thus one obtains:

$$\frac{\partial V}{\partial W^{1\,k}_{i_1 \ldots i_p}} = \sum_{q=1}^{r} r_k^1(q) y^2_{i_1}(q) \cdots y^2_{i_p}(q). \tag{10}$$

In the similar manner, the derivative of $V$ with respect to $W^{n\,k}_{i_1 \ldots i_p}$ in layer $n$ becomes:

$$\frac{\partial V}{\partial W^{n\,k}_{i_1 \ldots i_p}} = \sum_{q=1}^{r} \frac{\partial v(q)}{\partial y_k^n(q)} \cdot \frac{\partial y_k^n(q)}{\partial W^{n\,k}_{i_1 \ldots i_p}}. \tag{11}$$

Then, by putting:

$$r_k^n(q) \equiv \frac{\partial v(q)}{\partial y_k^n(q)} = \sum_{l=1}^{M_{n-1}} \frac{\partial v(q)}{\partial y_l^{n-1}(q)} \cdot \frac{\partial y_l^{n-1}(q)}{\partial y_k^n(q)}, \tag{12}$$

one finally obtains:

$$r_k^n(q) = p \sum_{l=1}^{M_{n-1}} r_l^{n-1}(q) \times \sum_{j_1=0}^{M_n} \cdots \sum_{j_{p-1}=0}^{M_n} W^{n-1\,l}_{j_1 \ldots j_{p-1} k} y^n_{j_1}(q) \cdots y^n_{j_{p-1}}(q) \tag{13}$$

with considerations of $r_l^{n-1}(q) = \partial v(q)/\partial y_l^{n-1}(q)$ in layer $n$-1 and Eqs.(2) and (4). Furthermore, one obtains from Eq.(4):

$$\frac{\partial y_k^n(q)}{\partial W^{n\,k}_{i_1 \ldots i_p}} = y^{n+1}_{i_1}(q) \cdots y^{n+1}_{i_p}(q). \tag{14}$$

Thus, one finally obtains:

$$\frac{\partial V}{\partial W^{n\,k}_{i_1 \ldots i_p}} = \sum_{q=1}^{r} r_k^n(q) y^{n+1}_{i_1}(q) \cdots y^{n+1}_{i_p}(q). \tag{15}$$

Then, the learning algorithm of weight $W^{n\,k}_{i_1\dots i_p}$ becomes as follows:

$$\frac{dW^{n\,k}_{i_1\dots i_p}}{dt} = -\sum_{q=1}^{r} r^n_k(q) y^{n+1}_{i_1}(q)\cdots y^{n+1}_{i_p}(q), \qquad (16)$$

where $r^n_k(q)$ is determined recursively by using Eqs.(9) and (13).

Eq.(16) can be transformed into the following discrete form:

$$W^{n\,k}_{i_1\dots i_p}(\tau+1) = W^{n\,k}_{i_1\dots i_p}(\tau) - \xi\sum_{q=1}^{r} r^n_k(q) y^{n+1}_{i_1}(q)\cdots y^{n+1}_{i_p}(q), \qquad (17)$$

where $\xi$ is called a learning rate factor that affects the speed of learning. In case the variation of weight is small, the following approximated expression can be employed:

$$W^{n\,k}_{i_1\dots i_p}(\tau+1) = W^{n\,k}_{i_1\dots i_p}(\tau) - \xi r^n_k(q(\tau)) y^{n+1}_{i_1}(q(\tau))\cdots y^{n+1}_{i_p}(q(\tau)) \qquad (18)$$

with $q(\tau)=\tau+1-ir$ $(ir\leq\tau<(i+1)r)$. In order to accelerate the convergence of the error back-propagation learning algorithm, the momentum method is often used in the following form:

$$\Delta W^{n\,k}_{i_1\dots i_p}(\tau+1) = \alpha\Delta W^{n\,k}_{i_1\dots i_p}(\tau) - \xi r^n_k(q(\tau)) y^{n+1}_{i_1}(q(\tau))\cdots y^{n+1}_{i_p}(q(\tau)), \qquad (19)$$

$$W^{n\,k}_{i_1\dots i_p}(\tau+1) = W^{n\,k}_{i_1\dots i_p}(\tau) + \Delta W^{n\,k}_{i_1\dots i_p}(\tau), \qquad (20)$$

where $\Delta W^{n\,k}_{i_1\dots i_p}(\tau)$ is the weight increment and $\alpha$ is a positive momentum factor that suppresses oscillation and improves the learning performance [1,2].

Page limitations preclude an inclusion of the proof that there exist no local minima for the single-layer network of the proposed neurons (i.e. with no hidden layers).

## 4. Application to Prediction of Chaotic Behavior

In order to verify the effectiveness of the proposed neuron model, application is made to the prediction problem of the time series data [3,4]. The prediction of the time series data is very important in many areas, e.g. signal processing, meteorology, economics, and so on. A time series signal through delay elements is applied to the single unit of proposed neuron as shown in Fig.3. This system can learn the mapping from $\{s_t, s_{t-1},\dots, s_{t-m+1}\}$ to $\{s_{t+1}\}$. This corresponds to, so to speak, the identification of the nonlinear autoregressive model by neural network. Once the learning has been completed, it can achieve the one step ahead prediction of its time series signal. If we feed the output of the neuron directly back to the input of the delay chain, the far future prediction of time series signal can be achieved based on the time series data up to the present.

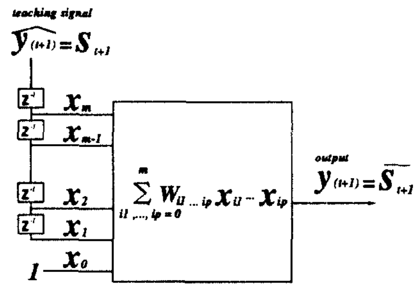



Fig.3. Structure of single unit of neuron which achieves the prediction of time series signal. $z^{-1}$ is the unity delay element.

Here we employ the following chaos dynamical system [5] as the target model to be identified:

$$s_{t+1} = (1+ab)s_t - bs_tv_t, \qquad (21)$$

$$v_{t+1} = (1-b)v_t + bs_t^2 \qquad (22)$$

with $a$=0.45 and $b$=1.9. The problem here is to predict the time history of $s_t$ by using the proposed neuron model.

First we have to determine the followings: (i) the order of polynomials, (ii) the number of delay elements. As for (i), we employ the 3rd order polynomials. Then as for (ii), we follow the Takens' theorem [6], i.e. we employ the embedding dimension $m$ as the number of delay elements.

The embedding dimension is determined by the following procedure [7]. First the correlation integration:

$$C_m(r) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} H(r-|s_i^m-s_j^m|) \qquad (23)$$

is evaluated by using the time series data $s_i^m = [s_i, s_{i-1},\dots, s_{i-m+1}]$, where $H(x)$ ($H(x)$=1 when $x\geq0$; $H(x)$=0 when $x<0$) is the Heaviside function, and $N$ is the number of data. The correlation dimension $D_2^m$ of the attractor is defined by:

$$D_2^m = \lim_{r\to0}\frac{\log C_m(r)}{\log r}. \qquad (24)$$

If the system is deterministic, $D_2^m$ approaches to a constant value as $m$ increases. This constant value is the dimension of the attractor, and $m$ which gives that value is the embedding dimension. Fig.4 shows the correlation integration and the dimension of the attractor for the chaotic signal $s_t$. We can see from this figure that the dimension of the attractor is about 1.6 and its embedding dimension is about 5. Therefore, the neuron model with 5 delay elements is employed in this experiment. Fig.5 shows the prediction results of time history of $s_t$ in Eq.(21) by using the proposed neuron model. The learning is achieved by Eqs.(19) and (20), and the time series data $s_t$ from $t$=0 to $t$=204 was used for learning. The momentum factor $\alpha$ is 0.5 and the learning rate factor $\xi$ is 0.005. In the prediction procedure after $t$=205 (i.e. after learning is completed), the output of the neuron was directly fed back to the input of the delay chain.

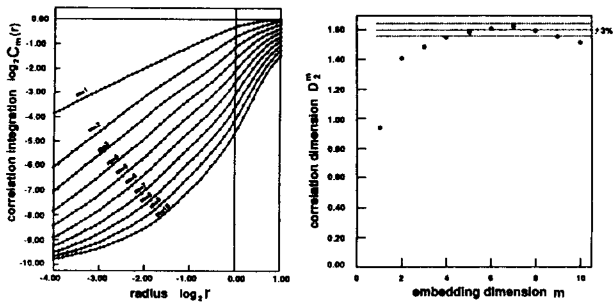



Fig.4. The correlation integration and the dimension of the chaotic signal $s_t$.

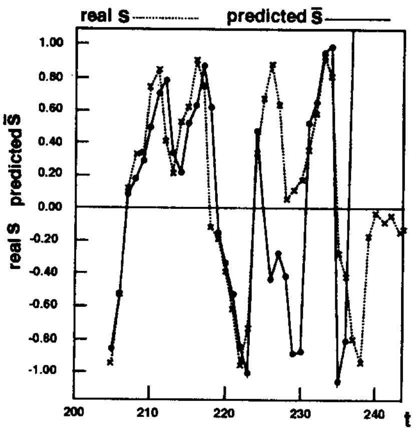



Fig.5. Prediction results of time history of $s_t$ by using the single unit of proposed neuron.

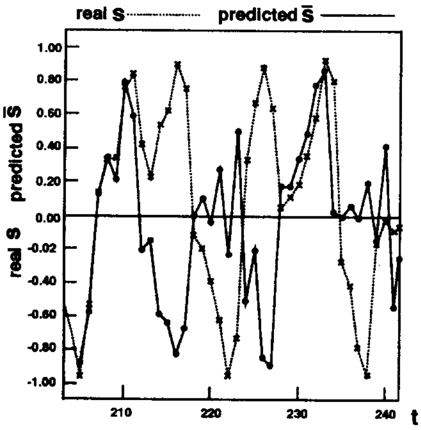



Fig.6. Prediction results of time history of $s_t$ by using the conventional neural network.

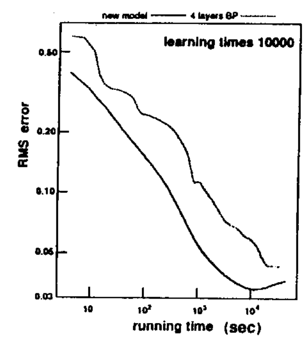



Fig.7. RMS error versus running time for the single unit of the proposed neuron and the conventional neural network.

The prediction results by the conventional neural network (4-layer network, 5 units for input and hidden layers, 1 unit for output layer, error back-propagation learning with the momentum factor 0.5 and the learning rate factor 0.1) are shown in Fig.6. The difference of the prediction performance is clear. The RMS (Root Mean Square) error versus running time for the single unit of the proposed neuron and the conventional neural network is shown in Fig.7 (this experiment was achieved on a personal computer with coprocessor).

## 5. Conclusions

In this paper, the input-correlated neuron model has been proposed. The back-propagation type learning algorithm has been derived, and it is proved that there exist no local minima for the single-layer network of the proposed neurons. It has been shown by the computer simulations that the learning speed is very fast compared with the conventional neural networks. The prediction of the future time history of chaotic signal has been achieved only by use of the single unit of the proposed neuron. As the proposed neuron has the high mapping ability, it could be applied to complex problems in many areas.

### References

[1] J.M. Zurada, *Introduction to Artificial Neural Systems*, West Publishing, St. Paul, 1992.

[2] D.E. Rumelhart and J.L. McClelland, *Explorations in Parallel Distributed Processing*, MIT Press, Cambridge, 1989.

[3] B. Müller and J. Reinhardt, *Neural Networks*, Springer-Verlag, Berlin, 1990.

[4] T. Yamakawa, E. Uchino, T. Miki, and H. Kusanagi, "A neo fuzzy neuron and its applications to system identification and prediction of the system behavior," *Proc. of the 2nd Int. Conf. on Fuzzy Logic and Neural Networks*, pp.477-483, 1992.

[5] E.N. Lorenz, "Computational chaos — a prelude to computational instability," *Physica D*, vol.35, pp.299-317, 1989.

[6] F. Takens, "Detecting strange attractors in turbulence," *Lecture Note in Mathematics*, 898, pp.366-381, Springer-Verlag, 1981.

[7] K. Aihara, *Chaos*, Science Publishing, Tokyo, 1990.