2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Vibrate Synchronize Function Neural Network Model – Its Backgrounds

Yoshitsugu Kakemoto Shinichi Nakasuka

***Abstract*—VSF–Network, Vibrate Synchronize Function Network, is a hybrid neural network combining a Chaos Neural Network with a hierarchical network. VSF–Network is designed for symbol learning by a neural network. It finds unknown parts of input data by comparing to learned pattern and it learns unknown patterns using unused part of the network. The new patterns are learned incrementally and they are represented as sub-networks with unused parts of hierarchical neural network. Combinations of patterns are represented as combinations of the sub-networks. The combinations of symbols are represented as combinations of the sub-networks. In this paper, the two theoretical backgrounds of VSF–Network are introduced. At the first, an incremental learning framework with Chaos Neural Networks is introduced. Next, the pattern recognition with the combined with symbols is introduced. By Stochastic Catastrophe Model, the authors explain the combined pattern recognition. Through an experiment, both the incremental learning capability and the pattern recognition with pattern combination.**
Index Terms: Incremental learning, Chaos Neural network, Nonlinear dynamics, Stochastic Catastrophe Model.

I. Introduction

The purpose of our research is developing a model of symbol-generation using neural networks. Model of symbol-generation have been proposed in the past years. Inamura[1] has proposed a model about stochastic behavior identification and symbol-generation. On the model, symbols are learned through the following steps.

1) At the first stage, patterns are learned by abstraction of input data. The pattern becomes a prototype of a symbol or low-level units of symbol. We refer this level pattern as proto-symbol.
2) Combinations of proto-symbol are learned by refining learned proto-symbol.
3) The refined symbols and the combinations of them are maintained.

Not only the patterns abstracted from input data but learning relations of learned patterns also are key component of the process. In these steps, the key of symbol-generation are summarized as following two points.

- Patterns are proto-symbol and they are learned by abstracting input data.
- The relations among acquired proto-symbol are abstracted and maintained.

In the case of the neural network model for symbol-generation, incremental learning of proto-symbol and the significant property of proto-symbol that they can be used be used in combination or alone are key component of the model. We have proposed VSF–Network[2] to implement incremental learning of symbols and pattern recognition with their combination. And we have reported our model and its performance in recent years[3], [4]. In this paper, we show its theoretical backgrounds and the results of the experiments related to the backgrounds. At the first section, we show theoretical background of incremental learning. In the following section, we show also the background of the pattern recognition by proto-symbol combination.

Yoshitsugu Kakemoto : JSOL Corp. 2-5-24, Harumi,Chuo-ku,Tokyo JAPAN (email:kakemoto.yoshitsugu@jsol.co.jp, kakemoto@gmail.com)
Shinichi Nakasuka : The University of Tokyo,7-3-1 Hongo,Bunkyo-ku, Tokyo, JAPAN (email: nakasuka@space.t.u-tokyo.ac.jp

II. Incremental Learning and Pattern Recognition by Chaos Neural Network

A. Incremental Learning

As for neural networks, learning of symbol is an instance of incremental learning[5]. The reason is that the neural network has to learn incrementally new patterns keeping learned patterns. On the incremental learning by neural network, correlations of learned patterns take an important role for the learning. Lin and Yao[6] have proposed Negative Correlation Leaning model as a model of the incremental learning. By the proposed method, the neural networks learn incrementally with increasing neurons based on the correlation between learned pattern and newly input pattern. Because the suitable number of neurons cannot be defined before learning, it is reasonable to determine the number of neurons according to progress of learning. Furthermore, there are redundant neurons in learned natural networks. Over-learning for patterns is caused by the redundant neurons. If we increase neurons in neural networks unconditionally, the problem is getting worse.

One way to solve the problem is that reusing neurons and learning new patterns incrementally, if there are neurons that do not participate pattern recognition on neural networks. By our model, new patterns that have low correlation to learned patterns are learned by reusing a part of neurons in the neural networks. VSF–Network learns patterns incrementally by dividing the network into sub-networks, if it has redundant neurons during incremental learning. The redundant neurons are found by CNN, Chaos Neural Network[7]. CNN checks whether redundant neurons exist or not in hidden layer of HNN, hierarchical neural networks. If any redundant neuron is not found, VSF-Network begins to increase neurons in hidden layer of HNN. In this paper, we assume that there are several redundant neurons in VSF–Networks after several learning phases.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

B. Pattern Recognition by Chaos Neural Network

Malsburg[8] proposed a hypothesis that the binding problem can be solved by synchronized firing among neurons. The binding problem is the problem how a number of recognize object can be recognize. In this model, each pattern configuring recognized objects corresponds to each group of synchronized firing neurons, and each pattern is recognized by this group. Based on this hypothesis, we apply the synchronized neuron group to recognize partial pattern configuring a recognized object, and we determine whether input data has unknown parts for memorized patterns.

By CNN, we can find chaotic retrieval dynamics in addition to normal dynamics by associative memory[9]. The chaotic retrieval dynamics can be find when input pattern is patchy patterns to stored patterns. If CNN works as an associative memory, internal status of neurons change periodically. Sometimes, statues of neurons match statuses of other neurons. This phenomenon is called the synchronized oscillation. In Fig.1, outputs from elements of GCM for epochs t ($t = 1, \cdots, 100$) are plotted. A part of outputs show synchronized oscillation and other outputs show isolated behavior to other elements.

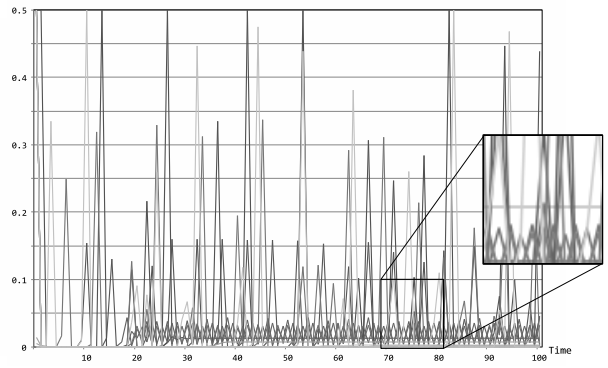

Fig. 1. Statuses of GCM with time evolution

The iTh. output x_i of CNN that has M input neurons and N chaos neurons is defined by (1).

$$x_i(t+1) = f\left[\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)\right] \quad (1)$$

In (1), an input term ξ_i, a feedback termη_i of internal status and an inhibitory ζ_i are respectively defined as follows.

$$\begin{aligned}\xi_i(t+1) &= \sum_{j=1}^{N} v_{ij} \sum_{d=0}^{t} A_j(t-d) \\ &= k_s \xi_i(t) + \sum_{j=1}^{N} v_{ij} A_j(t)\end{aligned} \quad (2)$$

$$\begin{aligned}\eta_i(t+1) &= \sum_{j=0}^{N} w_{ij} \sum_{d=0}^{t} k_m^d f\left\{x_j(t-d)\right\} \\ &= k_n \eta_i(t) + \sum_{j=1}^{N} w_{ij} x_j(t)\end{aligned} \quad (3)$$

$$\begin{aligned}\zeta_i(t+1) &= -\alpha \sum_{d=0}^{t} k_r^d f\left\{x_i(t-d)\right\} - \theta_i \\ &= k_r \zeta_i(t) - \alpha x_i(t) - \theta(1-k_r)\end{aligned} \quad (4)$$

In (2), (3) and (4), $A(t)$ is input at a time t and v_{jk} is the connection weight between the k Th. element of input pattern and the chaos neuron j. w_{ij} is the connection weight between chaos neuron i and chaos neuron j. k_s, $k_n (\simeq k_m^d)$, and $k_r (\simeq k_r^d)$ is a parameter for the each term. α is the parameter and θ_i is the threshold for the inhibitory term. f is Sigmoid function defined by (5).

$$f_a(x) = \frac{1}{1 + e^{-ax}} \quad (5)$$

The ξ_i is the self-interaction and η_i is an interacting term to other neuron j ($\neq i$), so CNN is a system that a status of neuron i diffuses to entirety of CNN through η and each neuron interacts to others. By this reason, CNN is an instance of Globally Coupled Mapping[10], GCM, that is a derivation of Coupled Map Lattice[11].

N neurons CNN as GCM is defined by (6).

$$\begin{aligned}y_i =& (1-\varepsilon) f^a(x_i) \\ &+ \frac{\varepsilon}{N} \sum_{j=1, j\neq i}^{N} f_a(x_j) \quad (1 \leq i \leq N)\end{aligned} \quad (6)$$

In (6), a is a parameter of a sigmoid function f, ε is a strength parameter of interactions among elements. Various dynamics on GCM have been reported. Komuro[12] analyzed dynamics on GCM using traverse Lyapunov exponents. Traverse Lyapunov exponents on an invariant manifold show stability of a trajectory to complementary spaces. Retracting observed on GCM is that several oscillators discretely oscillated are synchronized with other oscillators and show an identical oscillation. The fact that each element in synchronizing group converges to an invariant manifold was shown in his analysis.

Based on this finding and assumption that the invariant manifold corresponds to an equilibrium point, we can assume relation between stored patterns in an associative memory and a synchronized group on CNN. That is, patterns correlated with learned pattern reach equilibrium point corresponding to stored pattern as the result of the time-evolution. In contrast, other elements that have low correlation with the learned patterns do not reach any equilibrium point and they are in a unstable state.

C. Correlation between Associative Memory and Patterns

The dynamics of CNN is described as a dynamics of associative memory. On associative memory, information is stored as distributed pattern and required information is retrieved as partial information[13]. Hopfield[14] shows that dynamic system (7) has attractors and a part of them are equilibrium points, if weight matrix is a symmetric and output function is a monotonic increase. Kadone[15] studied about correlation between learning patterns and attractors of retrieved pattern reducing equilibrium points of the attractor. A time evolution

of retrieving process by CNN is defined as (7).

$$\begin{aligned} x(t+1) &= f(u(t)) \\ u(t) &= WX(t) - \alpha I \end{aligned} \tag{7}$$

On (7), u is an inner status of a neuron and W_{ij} is a connection weight between neuron i and j. α is a parameter to be 0 to on-diagonal element of W_{ii} As shown in (8), connection weights W are learned is with adding self correlation of the weights, if new patterns are input.

$$W = \sum_{\mu=1}^{p} \xi_\mu \xi_\mu^T - \alpha I \tag{8}$$

ξ_μ is a N column vector for a pattern μ.

We express highly correlated part of patterns μ,ν $(\mu \neq \nu)$ as $\xi_{\cdot,a}$, low correlated part of the patters is expressed as $\xi_{\cdot,c}$ and $N = N_a + N_c$. N_a and N_c show the number of elements of $\xi_{\cdot,a}$ and $\xi_{\cdot,c}$ A correlation between ξ_μ, ξ_ν is defined by (9).

$$\begin{aligned} &\frac{1}{N}\xi_{\cdot,a}^T \xi_{\cdot,a} \simeq 1, \\ &\frac{1}{N}\xi_{\cdot,a}^T \xi_{\cdot,c} \simeq \begin{cases} 1 & (\mu = \nu) \\ O(\frac{1}{\sqrt{N_c}}) & (\mu \neq \nu) \end{cases} \end{aligned} \tag{9}$$

A status of equilibrium point of (7) is determined by $Wf(u)$.

$$\begin{aligned} Wf(u) &= \left(\sum_{\mu=1}^{p} \xi_\mu, {\xi_\mu}^T - \alpha I \right) \cdot f(u) \\ &= \sum_{\mu}^{} \xi_{\mu=1}^{p} {\xi_\mu}^T \cdot f(u) - \alpha f(u) \end{aligned} \tag{10}$$

Here $\xi_\mu = \xi_{\mu,a} + \xi_{\mu,c}$, so (10) is rewritten as

$$\begin{aligned} &\sum_{\mu} (\xi_{\mu,a} + \xi_{\mu,c}) {\xi_\mu}^T \cdot f(u) - \alpha f(u) \\ =&\sum_{\mu} \xi_{\mu,a} {\xi_\mu}^T \cdot f(u) + \sum_{\mu} \xi_{\mu,c} {\xi_\mu}^T \cdot f(u) - \alpha f(u) \end{aligned} \tag{11}$$

We assume $f(u) = \xi_\nu$ and yield $\xi_{\mu,a}\xi_\mu^T \cdot \xi_\nu \simeq \xi_{\mu,a}$ from the correlation (9) and we obtain

$$\begin{aligned} &\frac{N_a}{N} \sum_{\mu} \xi_{\mu,a} + \frac{N_c}{N} \sum_{\mu} {\xi_{\mu,c}}^T \xi_\mu \xi_{\nu,c} - \alpha \xi_\nu \\ =&\frac{N_a}{N} \sum_{\mu} \xi_{\mu,a} + \frac{N_c}{N} \sum_{\mu} O(\frac{1}{\sqrt{N}}) \xi_{\nu,c} - \alpha \xi_\nu. \end{aligned} \tag{12}$$

The first term of (12) has a high correlated part to ξ_μ and $\xi_\nu \simeq \xi_{\mu,a}$. Therefore,

$$\frac{N_a}{N} \sum_{\mu} \xi_\mu + \left(\frac{O\left(1/\sqrt{N}\right)}{N} - \alpha \right) \xi_\nu. \tag{13}$$

If N_a is greater than N_c, (13) reaches an equilibrium point ξ_μ. If N_c is greater than a certain amount of value, (13) can not reach any an equilibrium point. As shown in the previous section II-B, clustered elements on GCM behavior on stable manifolds but unclustered elements cannot reach any stable manifold. If governed dynamics of GCM and associative memory are same one, the equilibrium points on associative memory correspond to the stable manifold. Based on this discussion, we conclude that CNN has an ability to find unknown part from input pattern. We apply CNN to find unknown part of input data at the hidden layer of a hierarchical neural network.

III. Recognition of Combinational Patterns

All proto-symbols learned by VSF–Network do not always need for recognition. If a pattern for single proto-symbol is given, VSF–Network retrieves a pattern for single proto-symbol. The probability density learned by VSF–Network is a multimodal distribution in a range of values and it is a unimodal distribution in other range. This means that multiple components or single component of limit mixture model are selected based on situations. To implement these dynamical process on VSF–Network, we developed weight update rule based on stochastic catastrophe theory[16], [17].

A. Patterns Combination and Mixture model

According to the neural manifold proposed by Amari[18], learning process and its results can be considered an approximation of probability density. Let M is a space spanned by a multi-layer neural network. The space M is called as neural manifold for the multi-layer neural network. The manifold is the space coordinates system with θ and a point on M expresses a probability distribution $p(y \mid x; \theta)$. Here θ is connection weight among neurons and $\theta = (w_1, \cdots, w_m; v_1, \cdots, v_n$ in the case of 3-layer neural network.

The distribution of y that x is given is expressed by a conditional probabilistic distribution Let a probability density of input x is $q(x)$, then the joint distribution for the input and output is

$$p(y \mid x; \theta) = q(x) p(y \mid x; \theta). \tag{14}$$

If learning by hierarchical neural network is a distribution estimation of values corresponding to input, the learning by VSF–Network is an estimation of a limited mixture model for a probability density function $p(x|a)$ with a parameter c_k

$$f\left(x \,|a\right) = \sum_{k=1}^{K} c_k p\left(x \,|a_k\right) \tag{15}$$

where $p(x|a_k)$ is a probability density corresponding to each proto-symbol.

One of learning algorithm for θ is the back-propagation. That is defined as

$$e\left(x_t, y_t; \theta\right) = \frac{1}{2} |y_t - f\left(x_t; \theta\right)|^2.$$

The stochastic descent method changes parameter θ_t at time t by

$$\theta_{t+1} = \theta_t - \eta \nabla e, \tag{16}$$

based on the expected values x_t, y_t. In (16), η is learning constant. ∇ is gradient and it is defined as

$$\nabla e = \left(\frac{\partial e}{\partial \theta_1}, \cdots, \frac{\partial e}{\partial \theta_N} \right). \tag{17}$$

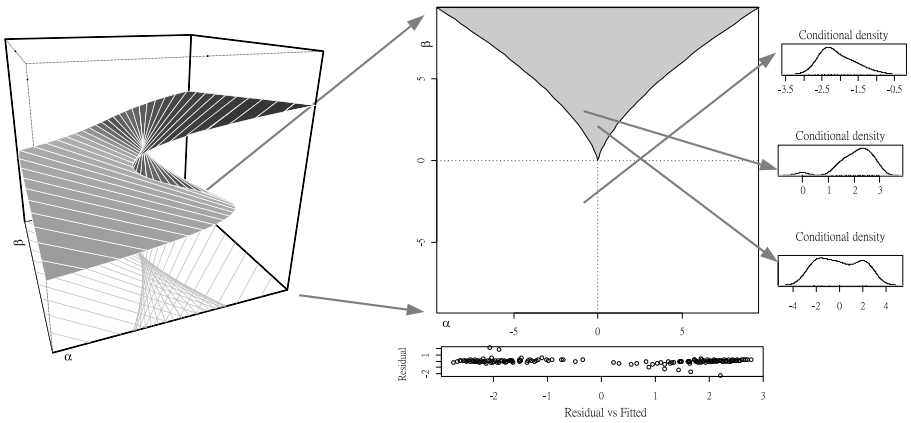

Fig. 2. An Overview of Stochastic Catastrophe model

B. Stochastic Catastrophe Model

On VSF–Network, there are several singular points, because equilibrium points are singular points from another perspective. In general, there are singular points on manifold spanned by non-liner function. At neighborhood of singular point, bifurcation occurs. The number of value corresponding to one variable changes at bifurcation point. For example, the number of real root of 3 degree equation $x^3 - cx = 0$ is 1 at the left side of the point $c = 0$ and it is 3 at the right site of the point $c = 0$. Based on the condition of parameters, there are two case of mapping,

$$f(x) \to y_1, y_2, y_3,$$
$$f(x) \to y.$$

Thom proposed catastrophe theory[19] about the bifurcations at equilibrium points to explain a sudden change of status at bifurcation point. We consider bifurcation at singular point on neural a manifold based on catastrophe theory. Existence of multiple solutions for a variable x means that multiple parameters for distributions exist. And existence of multiple parameters for distributions means that multiple components of limit mixture model are selected. If each of the components correspond to sub-network of VSF–Network, we can implement automatic sub-network selection based on input.

Cobb[16] applied catastrophe theory into multimodal density estimation and the estimation of multimodal probability density through use of the bifurcation of solution at the singular points was proposed. We apply the model to implement a combinational pattern recognition of the proto-symbols by VSF-Network.

We introduce the density function of stochastic catastrophe theory according to a series of studies by Cobb[16], [17], [20]. At the first, Cobb introduces a class of probability densities f_k that is expressed in the general form

$$f_k(x) = \xi(\beta) \exp\left[-\int^x \frac{g(s)}{v(s)} ds\right] \tag{18}$$

In (18), ξ is a normalized function, $g(x) = \beta_0 + \beta_1 x + \cdots + \beta_k x^k, k > 0$ and $v(x)$ has one of the following principal forms.

$$v(x) = \begin{cases} 1 & -\infty < x < \infty & ; \text{Type N} \\ x & 0 < x < \infty & ; \text{Type G} \\ x^2 & 0 < x < \infty & ; \text{Type I} \\ x\,(1-x) & 0 < x < 1 & ; \text{Type B} \end{cases}$$

The densities described by (18) are a generalization of the Pearson system. On differentiation with respect to x, (18) yields

$$\frac{f'(x)}{f(x)} = -\frac{g(x)}{v(x)}, \tag{19}$$

which contains Pearson's differential equation as a special case. So, this density function can be applied various parametric densities.

In the Pearson system, the degree k of the polynomial g is one and the degree of v is at most two. The polynomial g will be called the shape polynomial for the density f. We are principally concerned with the multimodal forms that appear when the degree of g exceeds one. The maximum number of modes possible in a given family is determined by the degree of its shape polynomial, k. From (19), it may be seen that the critical points of the density (i.e., those points x such that $f'(x) = 0$) are exactly the roots of $g(x)$. Whether such a point is a mode or an antimode (a relative minimum) depends on the sign of $g''(x) - \{g'(x)\}^2$. At the roots of $v(x)$ the density either has a zero ($f(x) \to 0$) or a pole ($f(x) \to \infty$), depending on the coefficients of $g(x)$. The only exceptions to this occur at points that are roots of both $g(x)$ and $v(x)$.

The generalized family of Pearson distributions may also be characterized in terms of nonlinear diffusion processes. Let $2\mu(x) = g(x) - v'(x)$, and $\delta^2(x) = v(x)$. Then $f(x)$ is the stationary density of a stochastic process x that is governed by the SDE, stochastic differential equation

$$dX_t = -\mu\,(X_t)\,dt + \delta\,(X_t)\,dW_t \tag{20}$$

where W_t is a standard Wiener process. The stochastic flow defined by (16) is also governed by the SDE (20).

In the Type N cases, namely $v(x) = 1$, these equilibrium points are exactly the modes and antimodes of the corresponding probability density function. In these cases, modes

correspond to stable equilibrium points, while antimodes correspond to unstable equilibrium points. Bimodal stationary densities can arise when there is but one corresponding stable equilibrium.

To simplify the notation, let $N_k(x)$ refer to the density of the Type N family of degree k for permissible k. The N_k densities have as their principal member the normal density, N_1. The general form for an N_k density is

$$N_k(x) = \xi \exp\left[\theta_1 x + \theta_2 x^2 + \cdots + \theta_{k+1} x^{k+1}\right] \qquad (21)$$

where $\theta_j = -\beta_{j-1}/j$. N_k has finite moments of all orders if k is odd and $\theta_{k+1} < 0$. Cobb[16] proposed a solution of SDE (20) that satisfies the general form for an N_k density. Grasman[21] proposed stochastic catastrophe density function based on Cobb[16], which is expressed as

$$f(y) = \frac{\psi}{\sigma^2} \exp\left[\frac{\alpha(y-\lambda) + \frac{1}{2}\beta(y-\lambda)^2 - \frac{1}{4}(y-\lambda)^4}{\sigma^2}\right] \qquad (22)$$

In (22), ψ is a normalizing constant, λ merely determines the origin of scale of the state variable and δ is a scale parameter. β is called the bifurcation factor, as it determines the number of modes of the density function, while α is called asymmetry factor as it determines the direction of the skew of the density. The density is symmetric, if $\alpha = 0$ and becomes left or right skewed depending on the sign of α.

An overview about the relation between stochastic catastrophe model and multimodal densities generated by the density (21) is shown in Fig.2. β is called the bifurcation factor, as it determines the number of modes of the density function, while α is called asymmetry factor as it determines the direction of the skew of the density. The density is symmetric, if $\alpha = 0$ and becomes left or right skewed depending on the sign of α.

IV. VSF–Network and Selective Weight Update

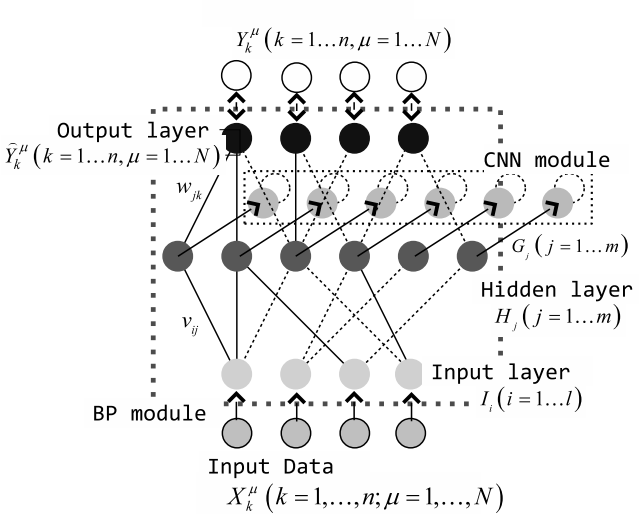

Fig. 3. An Overview of VSF-Network

In Fig.3, an overview of VSF–Network is shown. VSF–Network is composed of BP–module,a hierarchical network, and CNN–module that is CNN. CNN–module finds known or unknown parts of an input data and an used part of hidden layer neurons of BP–module. BP–module is trained with the selective weight update rule based on information from CNN-Module.

A. Learning Procedure

VSF–Network works for the incremental learning only, so the learning of VSF–Network is assumed that the initial connection weights among layers have been learned before the its incremental learning. The learning of VSF–Network is performed as follows.

1) Data are input to the input layer of BP–module.
2) The outputs of the hidden layer in BP–module are input to CNN–module and they are used as the initial state of each neuron of CNN–module.
3) From the initial state, CNN–module performs the retrieval process based on the dynamics of (1) for times $t = 1, \cdots, T$. The consistent rates defined by (31) are calculated.
4) The rest process of the forward path on BP–Module is performed. The error $E_{k,\mu}$ between the output $\hat{Y}_k$ of BP-module and the target output Y^μ for the input data is calculated.
5) The connection weights among layers are updated based on the weight update rule defined by (29) and (30).

B. Selective Weights Updating

By the previous section, we can identify unknown parts of inputs and identify redundant neurons with equilibrium point analysis for attractors of pattern retrieving on associative memory. If inputs for associative memory are statuses of hidden layer of multi-layer network, the retrieved patterns of the associative memory are statuses of hidden layer of the multi-layer network. The statuses based on learning experiences of the multi-layer network. If inputs for CNN are statuses of hidden layer, we can find redundant neurons at hidden layer that do not relate to pattern recognition. On CNN, redundant neurons that do not relate to pattern recognition show asynchronous oscillation while the network retrieves patterns. With updating only connection weights for neurons that show asynchronous oscillation, the network can learn new pattern without degradation of capability for learned patterns. From these standpoints, we have been proposed weight updating method for neural network[3]. We can summarize properties of the selective weight updating rule as the following three points.

- If multi-layer network has redundant neurons, then the weights are updated.
- The connection weights for neurons that show synchronous oscillation to other neurons are not updated.
- The connection weights for neurons that show asynchronous oscillation to other neurons are updated.

The delta rule for multi-layer network can be changed based on this selective weights update rule as follows. In (23), the ΔW_{ij} is the delta value for the weight between the iTh. input layer neuron and the jTh. hidden layer neuron. ΔW_{jk} is the delta value for the weight between the jTh. hidden layer neuron and the Th.kTh. output layer neuron.

$$\Delta W_{ij} = \begin{cases} \eta \frac{\partial E_{ij}}{\partial W_{ij}} & (\lambda_i \leq P) \\ 0 & (\lambda_i > P) \end{cases}, \quad \Delta W_{jk} = \begin{cases} \eta \frac{\partial E_{jk}}{\partial W_{jk}} & (\lambda_i \leq P) \\ 0 & (\lambda_i > P) \end{cases}. \tag{23}$$

Here η means coefficient for update, E_{jk} means learning error, λ_i is degree of coincidence calculated with (31) and P means threshold for λ_i.

C. Density Estimation and Selective Weights Updating

The estimation problem for these multimodal densities can be stated where we give the type and degree of the density, estimate the coefficient vector $\beta = (\beta_0, \beta_1, \cdots, \beta_k)$. Grasman[21] proposed estimation method for the density function (22) based on methods proposed by Cobb[20]. The core of Grasman's method is the fitting method that performs maximum likelihood estimation of all the parameters from (24) to (26) for observed dependent variables $Y_{i1}, \cdots, Y_{ip}$, and independent variables $x_{i1}, \cdots, x_{ip}$, for subjects $i = 1, \cdots, n$, the distribution of

$$y_i = w_0 + w_1 Y_{i1} + \cdots + w_p Y_{ip}, \tag{24}$$

where $w_0, \cdots, w_p$ are the first order coefficients of a polynomial approximation to the transformation. (24) is modeled by (22), with $\alpha \mapsto \alpha_i$ and $\alpha \mapsto \alpha_i$, where

$$\alpha = a_0 + a_1 X_{i1} + \cdots + a_p X_{ip} \tag{25}$$

$$\beta = b_0 + b_1 X_{i1} + \cdots + b_p X_{ip}. \tag{26}$$

The negative log-likelihood for a sample of observed values $(x_{i1}, \cdots, x_{iq}, y_{i1}, \cdots, y_{ip}), i = 1, \cdots, n$ is

$$L(a, b, w; Y, X) = \sum_{i=0}^{n} \log \psi_i - \sum_{i=0}^{n} \left[\alpha_i y_i + \frac{1}{2} \beta_i y_i^2 - \frac{1}{4} y_i^4 (y - \lambda)^4 \right] \tag{27}$$

The learning by VSF–Network is a minimizing L with respect to the parameters $w_0, \cdots, w_p$, $a_0, \cdots, a_p$, $b_0, \cdots, b_p$. From a standpoint of the subject of this section, the main target of learning by VSF–Network is an estimation of parameters controlling mode of densities. An equilibrium point, as a function of the control parameters α and β, are solutions to the equation

$$\alpha + \beta y - y^3 = 0. \tag{28}$$

This equation has one solution if $\lambda = 27\alpha - 4\beta^3$, which is known as Cardan's discriminant, is greater than zero, and has three solutions if $\lambda < 0$. Because VSF–Network performs incremental learning, so it should update the parameters to $\lambda < 0$. λ is determined by the parameters α and β, so VSF–Network should learn unknown patterns to increase β without changing parameter corresponding to known patterns. For the purpose, we apply the correlations among weights by the term $cor_{ki,kj}$. The selective weight update rule (23) is modified as,

$$\Delta W_{ij} = \begin{cases} \eta \frac{\partial E_{ij}}{\partial W_{ij}} & (\lambda_i \leq P) \\ 0 & (\lambda_i > P) \end{cases}, \quad \frac{\partial E_{ij}}{\partial W_{ij}} = (1.0 - |cor_{ki,kj}|)^{-1} \sum_{j=1}^{n} \frac{\partial E_{jk}}{\partial W_{jk}} f'(H_i^{\mu}). \tag{29}$$

Here $cor_{ki,kj}$ is correlation between neuron ki and kj in hidden layer, and H_i is output from neuron i in the hidden layer. The update rule between neuron k in output layer and neuron j in hidden layer is also modified as,

$$\Delta W_{jk} = \begin{cases} \eta \frac{\partial E_{jk}}{\partial W_{jk}} & (\lambda_i \leq P) \\ 0 & (\lambda_i > P) \end{cases}, \quad \frac{\partial E_{jk}}{\partial W_{jk}} = \left(\prod_{i=1}^{m} (1.0 - |cor_{ij}|) \right)^{-1} \sum_{j=1}^{n} E^{\mu} f'(O_k^{\mu}) \tag{30}$$

VSF–Network measures degree of coincidence between chaos neuron i and $j$at a part of the times $t = 1, \cdots, T$. This measurement is implemented with a correlation integral(31) based on Heaviside function (32).

$$C(r) = \frac{1}{n^2} \sum_{i,j=1 i \neq j}^{n} H(|x_i - x_j|) \tag{31}$$

$$H(t) = \begin{cases} 0 & (t < \Theta) \\ 1 & (t \geq 0) \end{cases} \tag{32}$$

Here x_i and x_j show a status of chaos neuron i and j and Θ is a threshold of function eq:Hviside.

V. Experiment and Result

In this section, we show a basic capability of VSF–Network through an experiment. The task for the experiment is a learning of avoiding obstacles by a rover. The goal of this task is that a rover learns an obstacle avoidance. The rover learns whether it can avoid an obstacle when the rover begins turning to the left from the point placed in. When the rover can avoid the obstacle, the output is $= 1$ otherwise $= 0$. It has the following three conditions about obstacle setting.

- Condition 1: T-Junction obstacle
- Condition 2 : Simple obstacle
- Condition 3 : Combined obstacle

We show an overview of these conditions in Fig 5.

The procedure of the experiment are described as fellows.

- The initial step
  - The initial weight for BP–Module is learned by multi-layer network using m records of from condition 1 of each task.
- The incremental learning step
  - We provide the patterns differ from the patterns that are assigned at the previous step. The input data is n records from the condition 2.

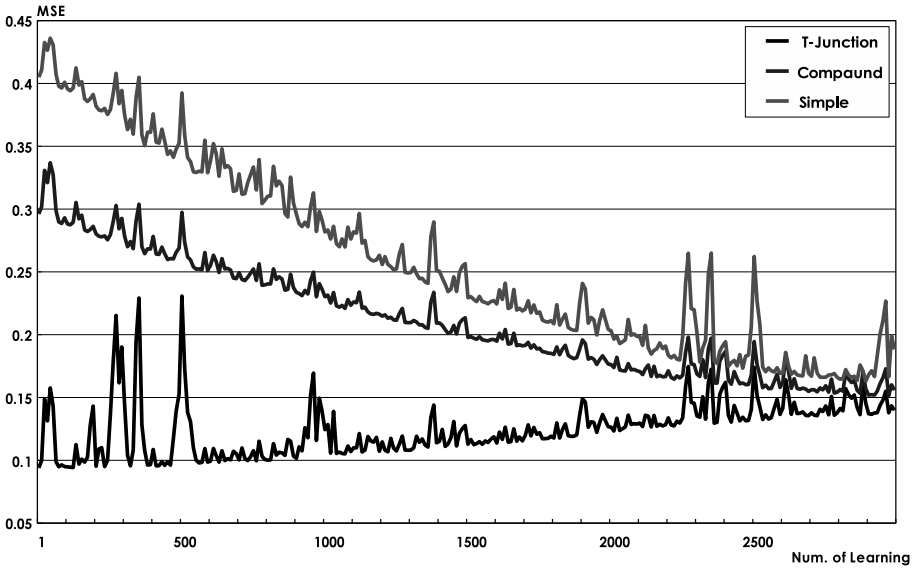

Fig. 4. Results of Experiment

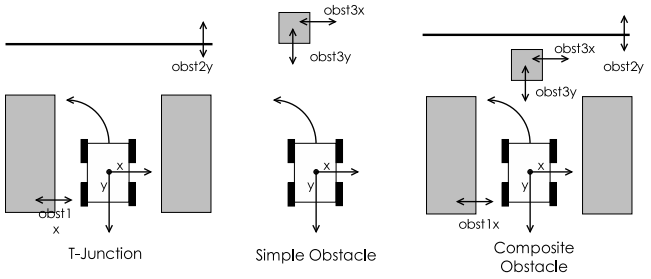

Fig. 5. Conditions of Task

- The step for testing learning performances.
  - We compare MSE (33) of each trial of the incremental learning to show the effect of the learnings.
  - To confirm the combination form, the data combined the condition1 and condition 2 are used in this step.

MSE, Mean Squired Error, is defined as

$$\mathrm{MSE}(\hat{y}) = \mathrm{E}\left[(\hat{y} - y)^2\right]. \tag{33}$$

Here $\hat{y}$ is an output from a network and y is an expected value for an input data.

In TABLE I, we show parameters and those settings of VSF–Network for the experiment. The parameters for CNN are set to show weak chaotic status. The number of learning at the initial step is $20,000$ and the number of learning at the initial step is $3,000$.

We show the result of incremental learning by VSF–Network for this task. Fig.4 shows the changes of MSE with the progress of the incremental learning for this task.

For incremental learning of the task, the effect of VSF–network is observed. VSF-Network learns new patterns incrementally and its weights are not destroyed. The combined patterns are learned without learning with the progress of incremental learning. After a certain number of incremental learning, MSE of every incremental learning reaches a equilibrium status. VSF–Network incrementally learns by reusing neurons which are considered as inconsequential neurons for identification of learned patterns. The incremental learning stops when redundant neuron is lost.

TABLE I
PARAMETER SETTING

Parameter	Description	Value
ε	Gradient of Sigmoid function	1.0
k_s	Coefficient of (2)	0.95
k_n	Coefficient of (3)	0.1
k_r	Coefficient of (4)	0.95
α	Inhibitory parameter of (4)	2.0
θ	Threshold for (4)	0.2
τ_{hst}	Time width of vibration period	100
η	Coefficient of selective weight update rule	0.1
P	Threshold for λ_i in selective weight update rule	0.01
Θ	Threshold for (32)	0.01

VI. CONCLUSION AND FUTURE WORKS

In this paper, we show the theoretical backgrounds of VSF–Network. The background consists of two parts. The first background is incremental learning and Chaos Neural Network. VSF–Network can identify unknown parts of input data and a part of neurons for reusing based on a dynamics of Chaos Neural Network. It learns only unknown parts of input data.

Another background is the estimating theory for multimodal or mixture probability density. The ability of VSF–Network for recognizing combined patterns that are learned by every sub-

network can be explained by geometrical properties of solution space.

Finally, we show that VSF–Network can recognize the combined patterns only if it have learned parts of the patterns.

The next step of our research concerns a detail consideration of the its dynamics. The current our theoretical scheme depends on two different bases. The first part of our scheme, the part for incremental leaning, is based on deterministic dynamics. The second part of our scheme, the part for recognition by proto-symbol combination, is based on stochastic dynamics. Originally, these are dynamics on an identical manifold, they should be discussed on the same condition.

REFERENCES

[1] T. Inamura, H. Tanie, and Y. Nakamura, "Proto-symbol development and manipulation in the geometry of stochastic model for motion generation and recognition." IEICE, Tech. Rep. NC2003-65, 2003.

[2] Y. Kakemoto and S. Nakasuka, "The learning and dynamics of vsf-network," in *Proc. of ISIC 2006*, 2006, pp. 1625–1630.

[3] ——, "The dynamics of incremental learning by vsf-network.." in *Proc.ICANN '2009*, 2009.

[4] ——, "Neural assembly generation by selective connection weight updating." in *Proc. IjCNN '2010*, 2010.

[5] C. Giraud-Carrier, "A note on the utility of incremental learning," *AI Communications*, vol. 13, pp. 215–223, 2000.

[6] M. Lin, K. Tang, and X. Yao, "Incremental learning by negative correlation leaning." in *Proc. of IJCNN 2008*, 2008.

[7] T. Aihara, T. Tanabe, and M. Toyoda, "Chaotic neural networks," *Phys. Lett.*, vol. 144A, pp. 333–340, 1990.

[8] von der Malsburg, *Am I thinking assemblies ? in Brain Theory (eds by G.Palm and A.Aertsen*. Springer–Verlag(Berlin), 1986.

[9] S. Uchiyama and H. Fujisaki, "Chaotic itinerancy in the oscillator neural network without lyapunov functions." *Chaos*, vol. 14, pp. 699–706, 2004.

[10] K.Kaneko, "Chaotic but regular posi-nega switch among coded attractors by cluster size variation," *Phys. Rev. Lett.*, vol. 63, p. 219, 1989.

[11] K. Kaneko, "Period-coupling of link-antilink patterns, quasi-periodicity in antiferrolike structure and spatial intermmittency in coupled map lattice. - toward a prelude to a field theory of chaos -," *Prog. Theor. Phys*, vol. 72, p. 1112, 1984.

[12] M. Komuro, "A mechanism of chaotic itinerancy in globally coupled maps," in *Dynamical Systems (NDDS 2002),*, 2002.

[13] N. Matsumoto, M. Okada, Y. Sugase, and S. Yamane., "Neuronal mechanisms encoding global-to-fine information in inferior-temporal cortex." *Journal of Computational Neuroscience*, vol. 18, pp. 85–103, 2005.

[14] J. Hopfield, "Neurons with graded response have collective computational properties like those of two-stage neurons." *Proceedings of the National Academy of Sciences of U.S.A.*, vol. 81, pp. 13 088–3092, 1984.

[15] H. Kadone and Y. Nakamura, "Symbolic memory of motion patterns using hierarchical bifurcations of attanctors in an associative memory model," *Journal of Robot Society of Japan*, vol. 25, no. 2, pp. 249–258, 2007.

[16] L. Cobb and R. Ragade, "Applications of catastrophe theory in the behavioral and life." *Behavioral Science*, vol. 79, no. 23, pp. 291–, 1978.

[17] C. L and W. B, "Statistical catastrophe theory: An overview." *Mathematical Modelling*, vol. 23, no. 8, pp. 1–27, 1980.

[18] S. Amari, "Information geometry and its application – 8," *ISCIE Journal of Systems, Control and Information*, vol. 49, no. 8, pp. 337–343, 08 2005.

[19] R.Thom, *Stability and Morphogenesis.:Essai Dúne Theorie Generale Des Modeles.* W. A. Benjamin, California, 1973.

[20] L. Cobb, B. Koppstein, and N. H. Chen, "Estimation and moment recursion relations for multimodal distributions of the exponential family." *Journal of the American Statistical Association*, vol. 78, pp. 124–130, 1983.

[21] R. Grasman, H. van der Maas, and E. Wagenmakers, "Journal of statistical software," *Mathematical Modelling*, vol. 32, no. 8, pp. 1–27, 2009.

[22] C.M.Bishop, *Pattern Recognition and machine Learning*. Springer–Verlag, 2006.