2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

Data Mining CNN to Circuit Modeling

Mamoru Tanaka[1], Yuko Zennyoji[2] and Hisashi Aomori[1]

Department of Electrical and Elecrtonics Engineering, Sophia University[1]
7-1 Kioi-cho, Chiyoda-ku, Tokyo 102-8554, Japan
Email: mamoru.tanaka@gmail.com
NTT Data[2]

Abstract—This paper describes a machine learning of a data-mining cellular neural network (CNN) from analysis of covariance structure using Back Euler method. It is the implicit method which is the most practical method for solving stiff systems. At each learning iteration step for quasi-Newton method, the Hessian matrix approximation for the matrix including second-order partial derivatives is updated by using Davidon-Fletcher-Powell (DFP) method. By using both of Back Euler method and DFP method, the solution for stiff systems in the parameter space can be obtained. That is, our purpose is to determine the weight parameters $\boldsymbol{\theta}$ in the connection matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{T}$ and $\boldsymbol{e}$ by the machine learning method for equilibrium points of the CNN states equation $\dot{\boldsymbol{x}} = 0$. The structure of the data-mining CNN is considered from viewpoints of circuit modeling.

1. Introduction

Recently, neural machine learning methods have been used as data mining to acquire the important information from massive amount of data and to predict future. That is, the data mining is to find 'rule' as classification, prediction, I/O mapping and association by using machine learning algorithm. However, the CNN learning methods have not been proposed sufficiently.
We propose a cellular analysis of covariance structure to predict the meaningful information including continuous data. The circuit model is called a data-mining CNN. In this paper, Backward Euler method, one of methods for solving a nonlinear differential equation, is used in parameter space. The important point is that this method can solve stiff systems which have large difference among eigenvalues. The resulted parameters are used as weights on edges in the cellular signal flow graph (SFG) which works as a circuit model with V-I transformation devices and a prediction model for unknown input data. Our simulation for the model of "Purchase of a Car" shows good result. This machine learning algorithm will be in SDP(Sophia Dynamics Program) which has been used as nonlinear circuit program with Fast Automatic Differentiation. Also, the structure of the data-mining CNN is considered from circuit modeling viewpoints and from Chua's theorems[1].

2. State and Measurement Equations

The observed variable $\boldsymbol{y}$ is a visible information data which has been obtained from real human behavior, natural environment and so on. The average $\boldsymbol{\mu}$ and the real covariance matrix $\boldsymbol{S} = E((\boldsymbol{y}-\boldsymbol{\mu})(\boldsymbol{y}-\boldsymbol{\mu})') \in \boldsymbol{R}^{l\times l}$ is calculated by using the observed variable $\boldsymbol{y}$.

Let $\boldsymbol{x} \in \boldsymbol{R}^n$ and $\boldsymbol{u} \in \boldsymbol{R}^m$ be the inner and outer state variable vectors with input-branches and without input-branches respectively, then the cellular structural equation is expressed by

$$\dot{\boldsymbol{x}} = -\boldsymbol{x} + \boldsymbol{A}\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{T} \tag{1}$$

where $\boldsymbol{A} \in \boldsymbol{R}^{n\times n}$ and $\boldsymbol{B} \in \boldsymbol{R}^{n\times m}$ are coefficient weight matrices which express connections between the state variables, and $\boldsymbol{T} \in \boldsymbol{R}^n$ is the error vector for the state variables. Each nonlinear function of $\boldsymbol{f}(\boldsymbol{x}) \in \boldsymbol{R}^n$ is a piece-wise linear function.

The cellular measurement equation should be used to express the cause and result relations between the observed variables $\boldsymbol{y} \in \boldsymbol{R}^l$ and the state variables $\boldsymbol{x}$, $\boldsymbol{u}$. The measurement equation is given by

$$\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{C}\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{D}\boldsymbol{u} + \boldsymbol{e} \tag{2}$$

where $\boldsymbol{C} \in \boldsymbol{R}^{l\times n}$ and $\boldsymbol{D} \in \boldsymbol{R}^{l\times m}$ are coefficient weight matrices which express connections between the state variables $\boldsymbol{x}$,$\boldsymbol{u}$ and the observed variables $\boldsymbol{y}$ for its average $\boldsymbol{\mu}$ for $\boldsymbol{y}$, and $\boldsymbol{e} \in \boldsymbol{R}^l$ is the error variables for the observed variables.

For example,each i-th row vector of the matrix $\boldsymbol{A}$ includes a weight element w_{ij} on the edge from a cell C_j to C_i. Generally, all matrices are sparse.

In this paper, our purpose is to determine the parameters of $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{T}$ and $\boldsymbol{e}$ by machine learning method for equilibrium points of the CNN states equation $\dot{\boldsymbol{x}} = 0$ from viewpoints of circuit modeling.

3. Optimization

3.1. Fit Function

Let $\boldsymbol{z}$ be the model standardized vector, then it is given by

$$\boldsymbol{z} = \boldsymbol{y} - \boldsymbol{\mu}, \tag{3}$$

then,the CNN state equation can be changed to structural and measurement equations in equilibrium point as follows:

$$\begin{pmatrix} \boldsymbol{z} \\ \boldsymbol{x} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{A} \end{pmatrix} \begin{pmatrix} \boldsymbol{z} \\ \boldsymbol{f(x)} \end{pmatrix} + \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{D} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{B} \end{pmatrix} \begin{pmatrix} \boldsymbol{e} \\ \boldsymbol{T} \\ \boldsymbol{u} \end{pmatrix} \tag{4}$$

In the case that a connection in the data-mining CNN is not symmetrical, we must determine it such that if $a_{ij} \neq 0$,then we set $a_{ji} = 0$ without any loop for its equilibrium. And if $a_{ii} = 0$, it seems that each state variable is in the equilibrium point of linear region as $|x| \leq 1$ to become$\boldsymbol{f(x)} = \boldsymbol{x}$. This will be proved later. In the linear region, if we define as

$$\boldsymbol{E} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \end{pmatrix}$$

$$\boldsymbol{A}_0 = \left(\boldsymbol{I} - \begin{pmatrix} \boldsymbol{0} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{A} \end{pmatrix} \right)^{-1}$$

$$\boldsymbol{B}_0 = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{D} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{B} \end{pmatrix}$$

it is derived that

$$\boldsymbol{z} = \boldsymbol{E}\boldsymbol{A}_0\boldsymbol{B}_0 \begin{pmatrix} \boldsymbol{e} \\ \boldsymbol{T} \\ \boldsymbol{u} \end{pmatrix}. \tag{5}$$

where $(\boldsymbol{e}, \boldsymbol{T}, \boldsymbol{u})'$ is an outer vector with no input-branches.
Let $\boldsymbol{\Sigma} \in \boldsymbol{R}^{l \times l}$ be the covariance matrix $E(\boldsymbol{z}\boldsymbol{z}')$ for (5), then it is derived theoretically from the cellular structural and measurement equations as follow:

$$\boldsymbol{\Sigma} = \boldsymbol{E}\boldsymbol{A}_0\boldsymbol{B}_0\boldsymbol{\Phi}_0\boldsymbol{B}_0'\boldsymbol{A}_0'\boldsymbol{E}' \tag{6}$$

where $\boldsymbol{\Phi}_0$ defined as the covariance matrix of the vector $(\boldsymbol{e}, \boldsymbol{T}, \boldsymbol{u})'$ is a diagonal matrix by assumption. To use in (10), we define that $\boldsymbol{A}_! = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{A} \end{pmatrix}$.

By using the real covariance matrix $\boldsymbol{S} \in \boldsymbol{R}^{l \times l}$ and a parameter element vector $\boldsymbol{\theta} \in \boldsymbol{R}^p$, the fit function is defined as

$$f_c(\boldsymbol{\theta}) = \frac{1}{2} tr((\boldsymbol{S} - \boldsymbol{\Sigma})\boldsymbol{S}^{-1})^2 \tag{7}$$

where $tr(\mathbf{X})$ means the trace of a matrix. In order to minimize the fit function, the function $\boldsymbol{g}(\boldsymbol{\theta}) = \frac{\partial f_c(\boldsymbol{\theta})}{\partial \theta_i}$ is changed to

$$\frac{\partial f_c(\boldsymbol{\theta})}{\partial \theta_i} = tr \left(\frac{\partial f_c(\boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}(\boldsymbol{\theta})} \frac{\partial \boldsymbol{\Sigma}(\boldsymbol{\theta})}{\partial \theta_i} \right). \tag{8}$$

In this equation, we derive that

$$\frac{\partial f_c}{\partial \boldsymbol{\Sigma}} = \boldsymbol{S}^{-1}(\boldsymbol{\Sigma}_y - \boldsymbol{S})\boldsymbol{S}^{-1}.$$

and also

$$\frac{\partial f_c}{\partial \phi_{ij}} = \left(\boldsymbol{B}_0'\boldsymbol{A}_0'\boldsymbol{E}' \frac{\partial f}{\partial \boldsymbol{\Sigma}} \boldsymbol{E}\boldsymbol{A}_0\boldsymbol{B}_0 \right)_{ij} \times 2.$$

Similarly, we derive that

$$\frac{\partial f_c}{\partial \boldsymbol{B}_0} = 2\boldsymbol{A}_0'\boldsymbol{E}' \frac{\partial f}{\partial \boldsymbol{\Sigma}} \boldsymbol{E}\boldsymbol{A}_0\boldsymbol{B}_0\boldsymbol{\Phi}_0 \tag{9}$$

$$\frac{\partial f_c}{\partial \boldsymbol{A}_!} = 2\boldsymbol{A}_0'\boldsymbol{E}' \frac{\partial f}{\partial \boldsymbol{\Sigma}} \boldsymbol{E}\boldsymbol{A}_0\boldsymbol{B}_0\boldsymbol{\Phi}_0\boldsymbol{B}_0'\boldsymbol{A}_0'. \tag{10}$$

From the above equations, the derivative function $\boldsymbol{g}(\boldsymbol{\theta}) = \frac{\partial f_c(\boldsymbol{\theta})}{\partial \theta_i}$ is derived from the component(i, j).

3.2. New Learning Algorithm

The function (7) should be minimized such that the parameters at that time are determined. So we want to solve the equation:

$$\mathbf{g}(\boldsymbol{\theta}) = 0 \tag{11}$$

where $\mathbf{g}(\boldsymbol{\theta}) = \frac{\partial f_c}{\partial \boldsymbol{\theta}}$.
However, the convergence depends on initial value when the nonlinear equation (11) is solved by an iterative solution method. In order to escape the initial problem, we solve the following equation by using Backward Euler method.

$$\dot{\boldsymbol{\theta}} = \mathbf{g}(\boldsymbol{\theta}) \tag{12}$$

The Backward Euler method is implicit numerical integral method for solving stiff systems. This method approximates the solution at virtual time $t_{k+1} = t_k + h$ by solving the implicit equation:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + hg(\boldsymbol{\theta}_{k+1}) \tag{13}$$

where the gradient vector $\mathbf{g}(\boldsymbol{\theta}_k)$ is evaluated at $\boldsymbol{\theta}_k$.
Since this equation(13) may be nonlinear, solving it in general requires an iterative solution method. In this paper, quasi-Newton method is provided for solving the implicit equation. An implicit method requires the solution of a nonlinear equation at each time step. For each step of Backward Euler method, we use the Newton method.
Let

$$\mathbf{F}(\boldsymbol{\theta}_{k+1}) = \boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k - h\mathbf{g}(\boldsymbol{\theta}_{k+1}), \tag{14}$$

then the Newton method is described as

$$^{(n+1)}\boldsymbol{\theta}_{k+1} = {}^{(n)}\boldsymbol{\theta}_{k+1} - (\frac{\partial \mathbf{F}(^{(n)}\boldsymbol{\theta}_{k+1})}{\partial^{(n)}\boldsymbol{\theta}_{k+1}})^{-1}\mathbf{F}(^{(n)}\boldsymbol{\theta}_{k+1}). \tag{15}$$

The inverse matrix computation of the Jacobi matrix $(\frac{\partial \mathbf{g}(^{(n)}\boldsymbol{\theta}_{k+1})}{\partial^{(n)}\boldsymbol{\theta}_{k+1}})$ is not available or expensive. Then the approximation technique is used. That is,we replace $(\mathbf{I} - h\frac{\partial \mathbf{g}(^{(n)}\boldsymbol{\theta}_{k+1})}{\partial^{(n)}\boldsymbol{\theta}_{k+1}})^{-1}$ by Davidon-Fletcher-Powell(DFP) approximation (17).
The gradient vector $\mathbf{g}(\boldsymbol{\theta}_n) = \frac{\partial f_c}{\partial \boldsymbol{\theta}}$ is evaluated at $\boldsymbol{\theta}_n$. $\mathbf{H}(\boldsymbol{\theta}_n) \in \mathbf{R}^{p\times p}$ is the matrix of second-order partial derivatives of a function with respect to $\boldsymbol{\theta}_n$. This is called Hessian. It is important to use the inverse of the Hessian matrix in our algorithm. However, since the Hessian leads to algorithmic and computational complexities, an approximation technique of the inverse Hessian is often used. We use the DFP method which is one of quasi-Newton methods. The update formula is as follows:

$$\boldsymbol{H}_{n+1} = \boldsymbol{H}_n + \boldsymbol{A}_n - \boldsymbol{B}_n \tag{16}$$

$$\boldsymbol{A}_n = \frac{\boldsymbol{p}\boldsymbol{p}'}{\boldsymbol{p}'\boldsymbol{q}} \quad \boldsymbol{B}_n = \frac{\boldsymbol{H}_n'\boldsymbol{q}\boldsymbol{q}'\boldsymbol{H}_n}{\boldsymbol{q}'\boldsymbol{H}_n\boldsymbol{q}}$$

$$\boldsymbol{p} = -\alpha\boldsymbol{H}_n F(\boldsymbol{\theta}_{k+1}^{(n-1)}) \quad \boldsymbol{q} = F(\boldsymbol{\theta}_{k+1}^{(n)}) - F(\boldsymbol{\theta}_{k+1}^{(n-1)})$$

$$\mathbf{H}_{n+1} = \mathbf{H}_n + \frac{\mathbf{r}\mathbf{r}'}{\mathbf{r}'\mathbf{d}} - \frac{\mathbf{H}_n'\mathbf{d}\mathbf{d}'\mathbf{H}_n}{\mathbf{d}'\mathbf{H}_n\mathbf{d}} \tag{17}$$

where

$$\mathbf{r} = -\alpha\mathbf{H}_n\mathbf{g}(\boldsymbol{\theta}_n) \quad \mathbf{d} = \mathbf{g}(\boldsymbol{\theta}_{n+1}) - \mathbf{g}(\boldsymbol{\theta}_n)$$

.

4. Circuit Model

A circuit model which is constructed by the learning algorithm becomes a data-mining CNN described as

State Equation

$$C\frac{dx_{ij}(t)}{dt} = -\frac{1}{R_x}x_{ij}(t) + \sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l)y_{kl}(t)$$

$$+ \sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l)u_{kl} + T \tag{18}$$

Output

$$y_{ij}(t) = \frac{1}{2}(|x_{ij}+1| - |x_{ij}-1|) \tag{19}$$

where a cell $C(i,j)$ is placed in pixel $P(i,j)$ in 2D-plane.

Definition 1: A connection between a cell $C(k,l)$ and a cell $C(i,j)$ in CNN is called symmetry for $A_1(i,j;k,l) = A_1(k,l;i,j)$ and one-directional asymmetry for that if $A_0(i,j;k,l) \neq 0$ then $A_0(k,l;i,j) = 0$ without any loop.

For the type of each branch connection, we define each element of A-template by using a binary parameter Ω such that:

$$A_\Omega(i,j;k,l) = \begin{cases} 2A(i,j;k,l) & if \ \ \Omega = 0 \\ A(i,j;k,l) & if \ \ \Omega = 1 \end{cases} \tag{20}$$

where $\Omega = 0$ and $\Omega = 1$ mean asymmetrical and symmetrical connections respenctively.

Definition 2: if any connection is $\Omega = 0$ or $\Omega = 1$, the CNN is called a data-mining CNN.

All Cell have different templates in the data-mining CNN. The weights depends on the learning for the structure determined by a data-mining analyser. The structure is determined to satisfy the conditions of the definitions. The data-mining CNN must give a stable state in linear region [-1,1] of the piece-wise linear fuction in parallelism because it is a circuit model for the data mining. The dynamics will be converged to an equilibrium point according to the following theorem.

Theorem 1: The data-mining CNN energy function which is defined as

$$E(t) = -\sum_{(i,j)}\sum_{(k,l)}\{\Omega\frac{1}{2} + (1-\Omega)\}A_\Omega(i,j;k,l)y_{ij}(t)y_{kl}(t)$$

$$+\frac{1}{2R_x}\sum_{(i,j)} y_{ij}(t)^2$$

$$- \sum_{(i,j)}\sum_{(k,l)} B(i,j;k,l)y_{ij}(t)u_{kl} - \sum_{(i,j)} Ty_{ij}(t)/(21)$$

satisfies that:

$$\frac{dE(t)}{dt} \leq 0 \tag{22}$$

The proof is now done as:

Since the symmetries $A_1(i,j;k,l) = A_1(k,l;i,j)$ is satisfied for the symmetrical connection, the next equations are derived in the data-mining CNN.

$$\frac{d}{dt}\frac{1}{2}A_\Omega(i,j;k,l)y_{ij}(t)y_{kl}(t) + \frac{d}{dt}\frac{1}{2}A_\Omega(k,l;i,j)y_{kl}(t)y_{ij}(t)$$

$$= \quad A_\Omega(i,j;k,l)\frac{dy_{ij}(t)}{dt}y_{kl}(t) + A_\Omega(k,l;i,j)\frac{dy_{kl}(t)}{dt}y_{ij}(t).$$

One of above two terms is 0 for the asymmetrical connection. Taking it into consideration, we can derive that:

$$\frac{dE(t)}{dt} = -\sum_{(i,j)}\sum_{(k,l)}\{\Omega + (1-\Omega)\frac{1}{2}\}A_\Omega(i,j;k,l)\frac{dy_{ij}}{dx_{ij}}\frac{dx_{ij}(t)}{dt}y_{kl}(t)$$

$$+\frac{1}{R_x}\sum_{(i,j)}\frac{dy_{ij}}{dx_{ij}}\frac{dx_{ij}(t)}{dt}y_{ij}(t)$$

$$-\sum_{(i,j)}\sum_{(k,l)}B(i,j;k,l)\frac{dy_{ij}}{dx_{ij}}\frac{dx_{ij}(t)}{dt}u_{kl}(t)-\sum_{(i,j)}T\frac{dy_{ij}}{dx_{ij}}\frac{dx_{ij}(t)}{dt}$$

By changing the element $A_\Omega(i,j;k,l)$ to original element $A(i,j;k,l)$ by using the eqaution (20),the following equations are derived:

$$\begin{aligned}\frac{dE(t)}{dt} &= -\sum_{(i,j)}\frac{dy_{ij}}{dx_{ij}}\frac{dx_{ij}(t)}{dt}\\ &\cdot[\sum_{C(k,l)\in N_r(i,j)}A(i,j;k,l)y_{kl}(t)-\frac{1}{R_x}y_{ij}(t)\\ &+\sum_{C(k,l)\in N_r(i,j)}B(i,j;k,l)u_{kl}+T]\\ &=-\sum_{|x_{ij}|<1}\frac{dx_{ij}(t)}{dt}\cdot\\ &\cdot[\sum_{C(k,l)\in N_r(i,j)}A(i,j;k,l)y_{kl}(t)-\frac{1}{R_x}x_{ij}(t)\\ &+\sum_{C(k,l)\in N_r(i,j)}B(i,j;k,l)u_{kl}+T\ ].\end{aligned}\tag{23}$$

By substituting the CNN sate equation (18) to above equation, we can demonstrate that:

$$\frac{dE(t)}{dt}=-\sum_{|x_{ij}|<1}C\left[\frac{dx_{ij}(t)}{dt}\right]^2\le 0 \tag{24}$$

Thorem 2: If

$$A(i,j;i,j)<\frac{1}{R_x} \tag{25}$$

then, each cell converges to the linear region:

The proof should be done such as that in the paper[1].

5. Simple Circuit Model

It is very important that the coefficient matrices are sparse and its SFG is cellular structure. The cellular structural state equations of the model for "Purchase of a Car" are given by

$$\begin{pmatrix}x_1\\x_2\end{pmatrix}=\begin{pmatrix}0 & a_{12}\\0 & 0\end{pmatrix}\begin{pmatrix}f(x_1)\\f(x_2)\end{pmatrix}+\begin{pmatrix}1\\b_{21}\end{pmatrix}\begin{pmatrix}u_1\end{pmatrix}+\begin{pmatrix}T_1\\T_2\end{pmatrix} \tag{26}$$

$$\begin{pmatrix}y_1\\y_2\\y_3\\y_4\end{pmatrix}=\begin{pmatrix}\mu_1\\\mu_2\\\mu_3\\\mu_4\end{pmatrix}$$

$$+\begin{pmatrix}1 & 0\\c_{21} & 0\\0 & c_{32}\\0 & 1\end{pmatrix}\begin{pmatrix}f(x_1)\\f(x_2)\end{pmatrix}+\begin{pmatrix}0\\0\\0\\0\end{pmatrix}\begin{pmatrix}u_1\end{pmatrix}+\begin{pmatrix}e_1\\e_2\\e_3\\e_4\end{pmatrix} \tag{27}$$

Fig 1 shows the learning curves for the model of "Purchase of a Car". The number of steps for BE method is shown on the horizontal axis, and a value of the fit function is shown on the vertical axis.

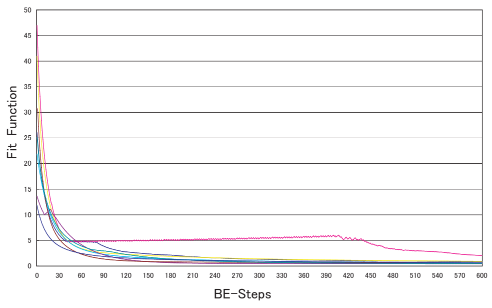

Figure 1: The simulation results of the model

6. Conclusions

A novel cellular learning analysis was proposed. The model by the learning is constructed by a circuit with current mode OTA's with nonlinearity.

Acknowledgments

This research is supported by the fund of Open Research Center Project from MEXT of Japanese Government (2007-2011).

References

[1] L. O. Chua and L. Yang, "Cellular neural networks: theory," *IEEE Trans. Circuits Syst.,* vol. 35, no. 10, pp. 1257–1272, Oct. 1988.

[2] Y. Zennyoji, N. Ohashi, M. Yamauchi, and M. Tanaka, "Cellular Analysis of Covariance Structure for Data Mining by Backward Euler Method,"In Proc. 2005 Intl. Symp. Nonlinear Theory and its Appl. (NOLTA 2005), Bruges, Belgium, Oct. 2005.