2014 International Symposium on Nonlinear Theory and its Applications
NOLTA2014, Luzern, Switzerland, September 14-18, 2014

Back Propagation Learning Based on an IDL Model

Yuta HORIUCHI† and Yoshihiro HAYAKAWA‡ Takeshi ONOMI† Koji NAKAJIMA†

†Laboratory for Brainware/Laboratory for Nanoelectronics
and Spintronics Research Institute of Electrical Communication, Tohoku University
2-1-1 Katahira, Aoba-ku, Sendai, 980-8577 Japan
‡Sendai National College of Technology
4-16-1 Ayashi-tyuuou,Aoba-ku,Sendai,989-3128 Japan
Email: horiuchi@nakajima.riec.tohoku.ac.jp, hayakawa@sendai-nct.jp,onomi@riec.tohoku.ac.jp, hello@riec.tohoku.ac.jp

Abstract—The Inverse function Delayed (ID) model was proposed as one of novel neural models. The ID model has an ability of oscillation, and this model can solve some local minimum problem in combinatorial optimization problems. However, it needs large calculation cost, and it is difficult to apply for large size combinational optimization problems. This problem was solved by Inverse function Delay-Less (IDL) model in combinational optimization problems. But learning process of IDL model has not been discussed yet. This study is to build a hierarchical network by using IDL model, and to derive a back propagation learning with IDL model. Finally, we discuss the performance of back propagation learning based on an IDL model.

1. Introduction

Recent computer's weak point is biological information processing, such as learning, pattern recognition, etc. Neural networks are expected to apply to those processing[1]. The neuronal models to constitute neural network are divided into two models. One is the macro type model. It has a purpose that investigates dynamics of a network. Another is micro type model. It has a purpose that investigates neuronal movement of the biology. However, an Inverse function Delayed (ID) model[2] has those two features. ID model has time delay between its internal state and output. By setting a negative slope to inverse output function, it can generate negative resistance field.

In solving combinational optimization problem using neural network, Hopfield models have local minimum problems. However, the ID model can avoid local minimum problems. Because it can set the negative resistance effect to a local minimum. So a solution exploration can escape from local minima[3][4][5].

On the other hand, the most typical process of the neural network is learning. In supervised learning, there are local minimum problems like combinatorial optimization problems. A preliminary research attempted to introduce a negative resistance effect to this learning process, and the new learning method which had a negative resistance effect was proposed[6][7]. It build, a hierarchical network using ID model, and it was trained through back propagation learning . The BP learning using ID model has better performance than normal BP learnings.

However, the learning using the ID model has a problem having a larger calculation cost rather than normal learnings. Because it has two different size time constants, ID model time constant and learning time constant. It is difficult to apply for a large size learning. Solving combinational optimization problem using ID model had the same problem.

This calculation cost problem was solved by Inverse function Delay-Less (IDL) model in combinational optimization problems[8]. IDL model improved calculation speed rather than ID model. This model has only one time constant. And it has the same ability as ID model.

IDL is a new model. So, learning performance of IDL model has not been discussed yet. This study is to build a hierarchical network using by IDL model, and we derive back propagation learning with IDL model. Finally, we compare IDL model with conventional back propagation model.

2. Inverse function Delayed (ID) model

This section explains about the ID model[2]. The model introduces biological time delay for output of Hopfield model[1]. The ID model is described as Eq. (1) and Eq. (2). u_i denotes internal state. w_{ij} denotes connection weight from neuron j to neuron i. The self connection weight denotes W. x_i denotes output of neuron i. τ_u and τ_x are time constant of internal state and output respectiuely. $f(u_i)$ is a output sigmoid function. $g(x_i)$ is an inverse function of $f(u_i)$. Eq. (2) decides the output of neuron. The condition of time constant is ($\tau_u >> \tau_x$).

$$\tau_u \frac{du}{dt} = \sum_j w_{ij} x_j + W x_i + h_i - u_i \qquad (1)$$

$$\tau_x \frac{dx_i}{dt} = u_i - g(x_i) \qquad (2)$$

Next, we explain about negative resistance effect of the ID model. Negative resistance affects network dynamics. The Eq. (1) is inserted to Eq. (2), we obtain Eq. (3) and

Eq. (4). The Eq. (3) will consider as movement of a particle in potential energy structure. d^2x_i/dt^2 denotes inertia, dx_i/dt denotes viscous resistance, and the right side of the equation denotes the force exerted by potential. Eq. (4) shows the coefficient functions of viscous resistance $\eta(x_i)$. This region of $\eta(x_i) < 0$ represents the negative resistance. By setting N shape as $g(x)$, we can get negative resistance effect.

$$\tau_x \frac{d^2x_i}{dt} + \eta(x_i)\frac{dx_i}{dt} = -\frac{1}{\tau_u}(g(x_i) - Wx_i - \sum_j w_{ij}x_j - h_i) \quad (3)$$

$$\eta(x_i) = \frac{dg(x_i)}{dx_i} + \frac{\tau_x}{\tau_u} \quad (4)$$

3. Inverse function Delay-less (IDL)model

In this paper, we use Eq. (6) as the neuron model[8]. The IDL model removes time delay of an ID model. By erasing time construct τ_x of Eq. (3), we can get the IDL model. We can consider Eq. (5) as the movement of a particle without inertia. $g(x_i)$ restricts the output range x from 0 to 1. $g(x_i)'$ controls the particle velocity. Therefor, by algorithmic limiting of output $g(x_i)'$ may be changed to unrelated function with $g(x_i)$. This setting doesn't influence to potential. The velocity amplitude function is changed to $A_x(x_i)$. The shape of $A_x(x_i)$ is similar to the differential form of the sigmoid function. It does not change the essence of the neuron model. By introducing the velocity amplitude function and a higher amplitude setting inside output space, IDL model can get like as negative resistance effect of ID model.

$$\frac{dx_i}{dt} = -\frac{1}{\tau_u g'(x_i)}(g(x_i) - Wx_i - \sum_j w_{ij}x_j - h_i) \quad (5)$$

$$\Delta x_i = -\frac{A(x_i)}{\tau_u}(g(x_i) - Wx_i - \sum_j w_{ij}x_j - h_i) \quad (6)$$

$$A_x(x_i) = \frac{\alpha_x}{1 + \exp\beta_x(0.5 - L_x - x_i)} - \frac{\alpha_x}{1 + \exp\beta_x(0.5 + L_x - x_i)} \quad (7)$$

4. Backpropagation learning using IDL model

This section explains about the method of backpropagation with IDL model. First, we explain about hierarchal network using IDL model. Next, we derive equations for IDL backpropagation learning.

4.1. Hierarchal network using IDL

The input layer structure normal neuron, and input signal passes through the input layer to a hidden layer. The hidden layer and an output layer take IDL models. Each layer is denoted by l. The behavior of IDL model from layer $l-1$ to layer l is written in Eq. (8) .IDL model has time construct τ_u. The neurons in the same layer are denoted by subindex numbers i, $n_{[l]}$ showed the number of layer neuron. The input signal to layer l is $x_j^{[l-1]}$, and connection weight is $w_{ij}^{[l]}$, $u_i^{[l]}$ showed internal state of IDL neuron i. $h_i^{[l]}$ is bias.

$$\Delta x_i^{[l]} = A_x(x_i^{[l]})(\sum_{j=1}^{n_{[l-1]}} w_{ij}^{[l]}x_j^{[l-1]} + Wx_i^{[l]} + h_i^{[l]} - g(x_i^{[l]})) \quad (8)$$

In this research, we define the velocity amplitude (VA) function of the IDL neuron as X VA function, and the VA function of the IDL back propagation as W VA function. W VA function is explained at next subsection.

4.2. Backpropagation using IDL model

Next, we derive BP learning using IDL model. In this research, for simplicity, we set only neuron for the output layer. It is necessary to set error function to update connection weight. We provide P patterns as training data, input $\vec{x^{[1]}}(p)$ and output teacher signal $T(p)$. Eq. (9) shows the error function. It is square error between output and teacher signal. The connection weight w_{1i} is updated according to Eq. (10).

$$E = \frac{1}{2}\sum_{p=1}^{P}(T(p) - x_1^{[3]}(p))^2 \quad (9)$$

$$\Delta w_{1i}^{[3]} = -\epsilon \sum_{p=1}^{P} \frac{\partial E}{\partial w_{1i}^{[3]}(p)} \quad (10)$$

A method of BP learning at consecutive time is suggested[9]. Using time constant τ_w, Eq. (10) is changed to consecutive time, it becomes Eq. (11).

$$\tau_w \frac{dw_{1i}^{[3]}}{dt} = -\sum_{p=1}^{P} \frac{\partial E(p)}{\partial w_{1i}^{[3]}} \quad (11)$$

$$= -\sum_{p=1}^{P} \frac{\partial E}{\partial x_1^{[3]}(p)} \frac{\partial x_1^{[3]}(p)}{\partial \theta_1^{[2]}(p)} \frac{\partial \theta_1^{[2]}(p)}{\partial w_{1i}^{[3]}(p)} \quad (12)$$

Eq. (11) is expanded using chain rule of differentiation, it becomes Eq. (12). Let us introduce a connection-weight time-constant τ_w ($\tau_w >> \Delta$). To consider time constant τ_w, Eq. (8) resemble following Eq. (13).And Eq. (13) becomes Eq. (14). Because τ_w is very much larger than time constant Δ, so $\Delta x_i^{[l]}$ is approximately 0 in learning period. The sum input signal from previous layer is θ_i in Eq. (15).

$$0 = A_x(x_i^{[l]})(\theta_i^{[l-1]} + Wx_i^{[l]} + h_i^{[l]} - g(x_i^{[l]})) \quad (13)$$

$$g(x_i^{[l]}) = Wx_i^{[l]} + \theta_i^{[l-1]} + h_i^{[l]} \quad (14)$$

$$\theta_i^{[l-1]} = \sum_{j=1}^{n_{[l-1]}} w_{ij}^{[l]}x_j^{[l-1]} \quad (15)$$

In IDL model using inverse output function g, $\partial x_1^{[3]}(p)/\partial\theta_1^{[2]}$ is changed to Eq. (16). We put

$g(x_1^{[3]}(p))' = 1/A_w(x_1^{[3]}(p))$ as IDL model. $A(x_i^{[l]})$ is described in Eq. (17). Each equation of the error function Eq. (9) and the sum of input signal Eq. (15) are insert to Eq. (12), it becomes Eq. (18).

$$\frac{\partial x_1^{[3]}(p)}{\partial \theta_1^{[2]}} = \frac{1}{\frac{\partial \theta_1^{[2]}}{\partial x_1^{[3]}(p)}} = \frac{1}{g(x_1^{[3]}(p))' - W} = \frac{1}{1/A_w(x_1^{[3]}(p)) - W} \quad (16)$$

$$A_w(x_i) = \frac{\alpha_w}{1+\exp\beta_w(0.5 - L_w - x_i)} - \frac{\alpha_w}{1+\exp\beta_w(0.5 + L_w - x_i)} \quad (17)$$

$$\tau_w \frac{dw_{1i}^{[3]}}{dt} = \sum_{p=1}^{P} \frac{1}{1/A_w(x_1^{[3]}(p)) - W} \left(T(p) - x_1^{[3]}(p)\right) x_i^{[2]}(p) \quad (18)$$

The connection weights are updated by online learning at each training data. Eq. (18) becomes Eq. (19).

$$\tau_w \frac{dw_{1i}^{[3]}}{dt} = \frac{1}{1/A_w(x_1^{[3]}(p)) - W} \left(T(p) - x_1^{[3]}(p)\right) x_i^{[2]}(p) \quad (19)$$

The equation of connection weight for update between output later and hidden layer using IDL model is Eq. (19).

Next, we describe about the connection weight update equation between input neuron i and hidden layer neuron j under the condition of one output neuron for input pattern P. The error function differentiate using connection weight w_{ij} like Eq. (11), it changes to Eq. (20). Eq. (20) is the equation for update connection weight. Eq. (20) is expanded using chain rule of differentiation, it becomes Eq. (21). By erasing partial differentiation of Eq. (21), it becomes Eq. (22). Eq. (22) shows update equation of connection weight between input layer and hidden layer.

$$\tau_w \frac{dw_{ij}^{[2]}}{dt} = -\frac{\partial E(p)}{\partial w_{ij}^{[2]}} \quad (20)$$

$$= -\frac{\partial E}{\partial x_1^{[3]}(p)} \frac{\partial x_1^{[3]}(p)}{\partial \theta_1^{[2]}(p)} \frac{\partial \theta_1^{[2]}(p)}{\partial x_j^{[2]}(p)} \frac{\partial x_j^{[2]}(p)}{\partial \theta_i^{[1]}(p)} \frac{\partial \theta_i^{[1]}(p)}{w_{1i}^{[3]}(p)} \quad (21)$$

$$\tau_w \frac{dw_{ij}^{[2]}}{dt} = \frac{1}{1/A_w(x_i^{[2]}(p)) - W} \frac{1}{1/A_w(x_1^{[3]}(p)) - W} \left(T(p) - x_1^{[3]}(p)\right) x_j^{[1]}(p) w_{1i}^{[3]} \quad (22)$$

In this research, the VA function of the IDL back propagation is referred to W VA function. The shape of each W VA function is the same.

5. Research of Learning ability

We inspected whether the BP learning of the IDL model can learn through XOR problem simulation. At first, we show a simulation condition. Next we describe a simulation results.

5.1. The Simulation condition

We investigate the learning ability through convergence rate of XOR problem. We use 3 layers network. The structure of network is [input, hidden, output]=[2,3,1]. The time constant and time notching are $\tau_w = 1.0, \tau_x = \delta t = 0.01$, respectively learns until $1 \times \tau_w$. Inverse output function is shown Eq. (23). We put $C = 2.0$ for inverse output function. The self connection weight is $W = 0$. Training pattern is given at random. If the error function became, $E < 0.01$, it was assumed that the learning would be converged at the right answer. The time over of BP learning set $5 \times 10^5 \tau_w$.

$$g(x_i) = f^{-1}(x_i) = \frac{1}{C} \log \frac{x_i}{1 - x_i} \quad (23)$$

We argue about the effect of the X and W VA function. So, we simulate using following condition.

- Condition 1: W VA function is constant, we change only X VA function.
- Condition 2: X VA function is constant, we change only W VA function.
- Condition 3: We change both of X and W VA function at the same time

We set $\beta_x = \beta_w = 10.0$ for the inclination of VA function. The Uniformity of VA function, $\alpha = 10.0, L = 0.01$. The simulation results are as follow; Across axle is width L of VA function $A(x_i)$. The vertical axis is the mean convergence probability for 100 times trial of XOR problem. In the same condition, we tried normal BP learning for XOR problem. The result of average convergence rate is 92%.

5.2. Result for change of X VA function

The figure1 shows the average convergence probability changing X VA function. Compared with normal BP, convergence probability improves in some area. Convergence probability was reduced for a wide L_x. Because the output of IDL model becomes 0 or 1 by wide VA-function. Therefor BP learning be stagnant.

5.3. Result for change of only W VA function

The figure2 shows convergence rate as a function of L in W VA function. Compare with normal BP, each parameter has high convergence rate. Because W VA function has a function corresponding to the learning rate of BP. Convergence is faster because the connection weight to be updated at a time is increased.

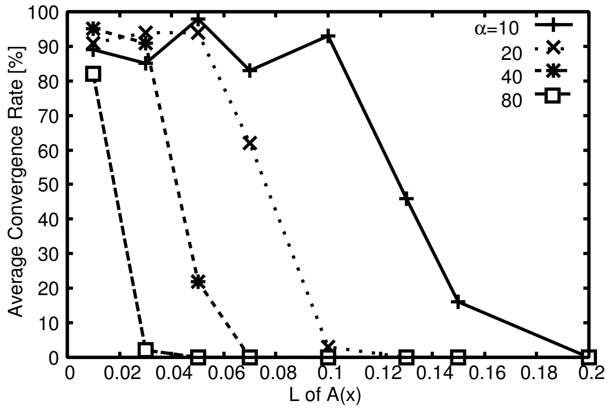

Figure 1: Convergence rate with X VA function

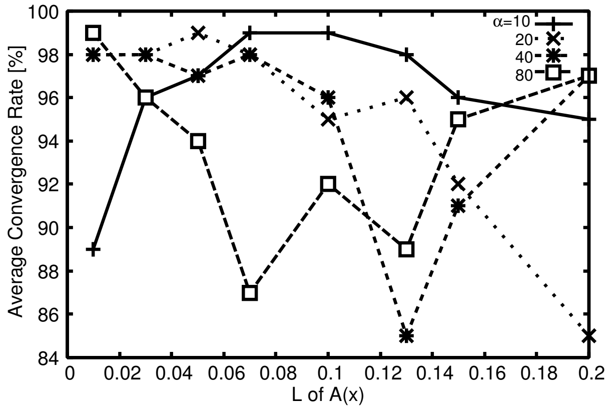

Figure 2: Convergence rate with W VA function

5.4. Result for change of both X and W VA function

The Figure3 shows the result of both X and W VA function. By simultaneously changing the acceleration function of the two, convergence rate is better than the condition 2 on some parameters. By widening the width of the VA function, convergence rate becomes to 0 like condition 1.

6. Conclution

In this research, we considered the IDL model BP learning. First, we constructed hierarchical network using the IDL model. Next, we derived the BP learning using the IDL model. Through the computer simulation, IDL model can learn XOR problem. Setting W VA function of IDL model has higher convergent performance than normal BP learning. In future, we will research about detail effect of W VA function.

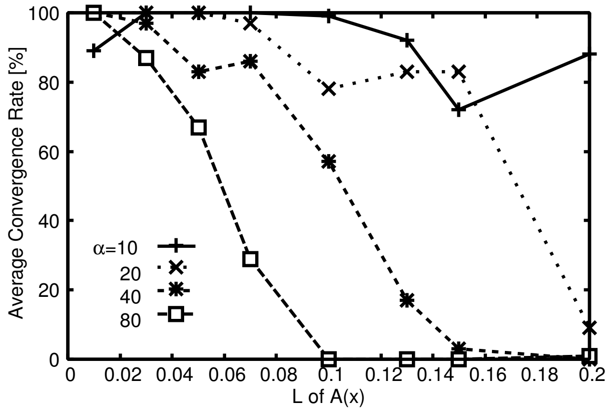

Figure 3: Convergence rate with X and W VA function

References

[1] Hopfield J.J. Tank,D.W."Neural computational of decisions in optimization problems",Biol.Cybern.vol.52,pp.141-152,1985.

[2] Akari Sato,Yoshihiro HAYAKAWA,Koji NAKAJIMA,"Avidance of the Permanent Oscillation Statee in the Inverse function Delayed Nerual Network"

[3] Akari SATO, Yoshihiro HAYAKAWA, Koji NAKAJIMA,"The Parameter Dependence of the Inverse Function Delayed Model on the Success Rate of Combinatorial Optimization Problems",J89-A(11), 960-972, 2006-11-01(Japanese)

[4] Yoshihiro HAYAKAWA, Koji NAKAJIMA, "The Method to solve Optimization Problems on a Neural Network using Inverse Delayed Model", NC2001-187, pp.151-158, 2002.(Japanese)

[5] Yoshihiro HAYAKAWA, Jun FUKUHARA,Tatsuaki DENDA, Koji NAKAJIMA"The Property of Inverse Delayed Model", NC2003-4, pp.19-24, 2003.(Japanese)

[6] Jun FUKUHARA, Yoshihiro HAYAKAWA, Koji NAKAJIMA. ,"BackPropagation with ID Model",NLP2003-92(2003-10)(Japanese)

[7] Jun FUKUHARA, Yoshihiro HAYAKAWA, Koji NAKAJIMA, "Learning capability of the ID neural network",NLP2004-4(2004-5)(Japanese)

[8] Yuto WATANABE, Yoshihiro HAYAKAWA, Shigeo SATO, Koji NAKAJIMA. "Inverse Function Delayless" (IDL) Model ,2012-12-NLP (Japanese)

[9] Owens, A.J. ,E.I. du Pont de Nemours Co. Inc., Wilmington, DE, USA, Filkin, D.L."Efficient training of the back propagation network by solving a system of stiff ordinary differential equations" Neural Networks, 1989. IJCNN.