International Journal of Fuzzy Logic and Intelligent Systems, vol. 8, no. 4, December 2008 pp. 264-269

Formulation of the Neural Network for Implicit Constitutive Model (Ⅱ) : Application to Inelastic Constitutive Equations

Joon-Seong Lee*, Eun-Chul Lee* and Tomonari Furukawa**

***Dept. of Mechanical System Engineering, Kyonggi University**
****Dept. of Mechanical Engineering, The Univ. of New South Wales, Sydney**

Abstract

In this paper, two neural networks as a material model, which are based on the state-space method, have been proposed. One outputs the rates of inelastic strain and material internal variables whereas the outputs of the other are the next state of the inelastic strain and material internal variables. Both the neural networks were trained using input-output data generated from Chaboche's model and successfully converged. The former neural network could reproduce the original stress-strain curve. The neural network also demonstrated its ability of interpolation by generating untrained curve. It was also found that the neural network can extrapolate in close proximity to the training data.

Key Words : Multilayer Neural Network, State Space Method, Modern Control Theory, Implicit Constitutive Model

1. Introduction

Problems of using explicit constitutive equations have been the difficulty of determining an appropriate parameter set and the inaccuracy of the model itself. The former problem has been overcome by the parameter identification approach referred to in Ref. [1]. For the latter, we have to either introduce a more complex explicit model or replace it by an implicit constitutive equation.

Multilayer feedforward neural networks, explained in this paper, have been proposed for material modeling by a couple or researchers as introduced in Ref. [2]. Yamamoto's model[3] is not as strong as the other two models as a consequence that the neural-based model is made with the help of Ramberg-Osgood model, thereby not being completely implicit. Nevertheless, this dose not mean that the other methods are most appropriate. One of the deficiencies of Ghaboussi's model[4] is that path-dependence is achieved by taking only the past three points. It is needless to say that we have to increase the number of past points to describe the hysteresis behavior of materials, although the increase of the number causes the dimension of the input space to be large. Miyazaki's architecture[5], in comparison, is rather the straight imitation of Ghaboussi's model. In order to describe the path-dependence, the architecture therefore redundantly used two components; internal variables and the past three points. Another serious problem, common in both the models, are that the increments $\Delta\sigma$ and ΔY are used as inputs, which is very sensitive to experimental data. This can yield very unstable neural networks particularly if the measurements are subject to errors, thereby not being suitable. Proposed by the authors in this paper is a multi neural network, the formulation of which is based on mathematical warrants. A general neural network constitutive model is presented in conjunction with the state-space method, which can describe any dynamical system.

Manuscript received Aug. 7, 2008; revised Dec. 11, 2008.
Corresponding author: jslee1@kyonggi.ac.kr

2. State-Space Method

The idea of state space comes from the state-variable method of describing differential equations. In this method, dynamical systems are described by a set of first-order differential equations in variables called the "state", and the solution may be visualized as a trajectory in space. The method is particularly well suited to performing calculations by computer.

Use of the state-space approach has often been referred to as modern control theory[6-8], whereas use of transfer-function based methods such as root locus and frequency response have been referred to as classical control design. Advantage of state-space design are especially apparent when engineers design controllers for systems with more than one control input or more than one sensed output. A further advantage of state-space design is that the system representation provides a complete internal description of the system, including possible internal oscillations or instabilities that might be hidden by inappropriate cancellations in the transfer-function (input/ output) description.

The motion of any finite dynamic system can be expressed as a set of first-order ordinary differential equations. This is often referred to as the state-variable representation. In general, a nonlinear dynamic system is given by

$$\dot{x}^t = \Phi(x^t, u^t) \tag{1}$$

with initial conditions :

$$x\big|_{t=t_0} = x^0, \tag{2}$$

where $x^t \in R^n$ is a set of n state variables and $u^t \in R^r$ a set of r control inputs. $\Phi : R^n \times R^r \to R^n$ is assumed to be continuously differentiable with respect to each of its arguments.

For example, Newton's law for a single mass M moving in one dimension x under force F is

$$M\ddot{x} = F. \tag{3}$$

If we define one state variable as the position $x_1 = x$ and the other state variable as the velocity $x_2 = \dot{x}$, this equation can be written as

$$\dot{x}_1 = x_2 \tag{4}$$

$$\dot{x}_2 = \frac{F}{M}. \tag{5}$$

This first-order linear differential equations can be concisely expressed using matrix notation. If we collect the state into a column vector x, and the coefficients of the state equations into a square matrix A, and the coefficients of the input into the column matrix B, these equations can be written in matrix form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{F}{M} \tag{6}$$

or

$$\dot{x} = A'x + Bu \tag{7}$$

where A is the system matrix and B is the input matrix. More generally the equation is hence represented by Eq. (1).

Now, we are concerned with discrete-time systems, which can be represented by difference equation corresponding to the differential equations given in Eq. (1). This takes the form:

$$x^{k+1} = \Phi\left(x^k, u^k\right) \tag{8}$$

where xk and uk are discrete time sequences. xk+1 is calculated by integrating Eq. (1) with respect to time as follows:

$$x^{k+1} = x^k + \Delta t\, \dot{x}^t \tag{9}$$

where Δt is the time increment between k and k+1.

One of some recent papers[9] demonstrated that neural networks can be used effectively for the identification and control of nonlinear dynamical systems. The structure of the controller used in the paper was dependent on the structure of the nonlinear system under consideration, of which some partial knowledge was assumed. This work was extended by Refs.[10,11], considering a unified approach for the control of higher order systems where full state information is not available, and where no specific information about the structure of the system is assumed.

Previous work by the dynamics community, as a conclusion, proved that multilayer neural networks can emulate a system whose structure is unknown but its input-output data are obtainable. Up to now, though not much discussion was done for their comparison, two major neural network architectures have been intensively used [12,13];

Case I Output $\dot{x}_k$ from x^k and u^k,
Case II Output x^{k+1} from x^k and u^k,

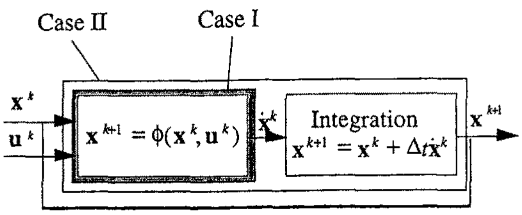
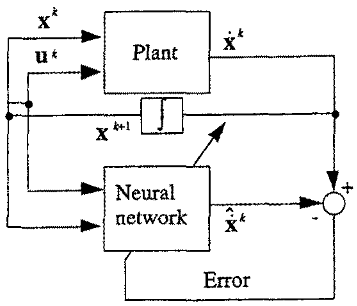
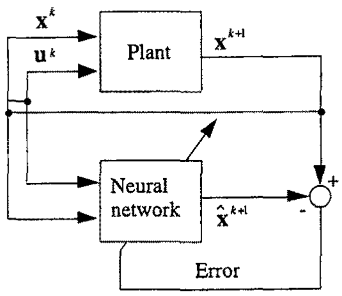


Fig. 1. Schematic diagram of state-space method



(a) Case I

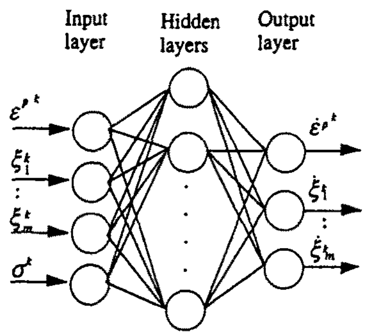


(b) Case II

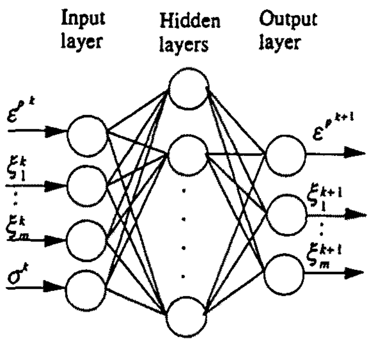
Fig. 2. Block diagrams of training neural networks

and they are illustrated in Fig. 1. Clearly Case II is more complicated than Case I because of the additional integration term. Fig. 2 and Fig. 3 shows the training of the neural network from the input-output data of the system.

If the material behavior can also be described in state-space form, we can apply neural networks to emulate the behaviors and use the networks as a material model. In the next section, we will propose the general state-space formulation of material behavior.



(a) Case I



(b) Case II
Fig. 3. Neural network architectures

3. State Space Representation of Material Behaviors

Under uniaxial loading, let the inelastic strain be εp and a set of ζ internal variables ξ∈Rζ, the inelastic material behaviors can typically have the following form:

$$\dot{\varepsilon}^{p^t} = \hat{\dot{\varepsilon}}^{p}(\varepsilon^{p^t}, \xi^t, \sigma), \tag{10}$$

$$\dot{\xi}^t = \hat{\dot{\xi}}(\varepsilon^{p^t}, \xi^t, \sigma), \tag{11}$$

where σ is the stress. Note here that internal variables can be kinematic and isotropic variables or anything else, depending on materials to be described. Comparing these equations with the state-space equation (1), we can find that the inelastic strain and internal material variables correspond to the state variables whereas the stress acts as a control input. The dynamics of the equations can be specified by giving the initial conditions of the state variables

$$\dot{\varepsilon}^{p}\Big|_{t=t_0} = \varepsilon^{p^0}, \tag{12}$$

$$\xi\Big|_{t=t_0} = \xi^0. \tag{13}$$

and the control inputs for all t accordingly. Similar to Eq. (8), Eqs. (10) and (11) can be restated for discretization:

$$\varepsilon^{p^{k+1}} = \hat{\varepsilon}^{p}\left(\varepsilon^{p^k}, \xi^k, \sigma^k\right), \tag{14}$$

$$\xi^{k+1} = \hat{\xi}\left(\varepsilon^{p^k}, \xi^k, \sigma^k\right). \tag{15}$$

Note that the control input of dynamical systems, being known for all t, are normally independent of the state variables, but the control input of the inelastic material are the stress and is therefore derived from the state variables.

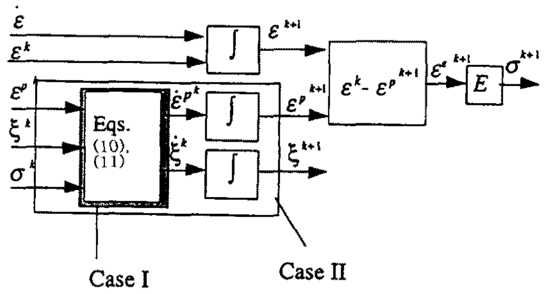
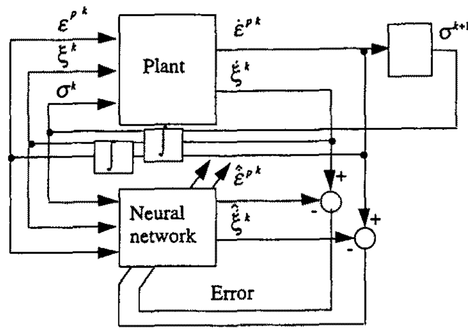
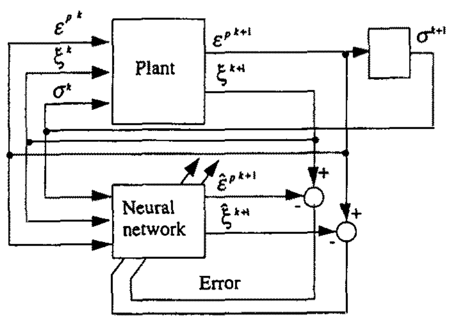


Fig. 4. Block diagrams for simulation



(a) Case I



(b) Case II
Fig. 5. Block diagrams of training neural networks

In accordance to the previous section, we can propose two neural networks;

Case I Output $\dot{\varepsilon}^{p^k}$ and $\dot{\xi}^k$ from $\varepsilon^{p^k}, = \xi^k$ and σ^k,
Case II Output $\varepsilon^{p^{k+1}}$ and ξ^{k+1} from $\varepsilon^{p^k}, = \xi^k$ and σ^k.

The proposed neural network architectures and the block diagram describing the derivation of the stress are respectively shown in Figs. 3 and 4, which block diagrams for training the networks are illustrated in Fig. 5. The advantage of the neural network architecture to the others are clearly the following points:

(a) Simplicity : Only one neural network is used, compared to Miyazaki's model[5], which uses two neural networks independently, and the input layer consists of only the latest information of strain, internal variables and stress.

(b) Generality : Depending on the material to be chosen, any kind of internal variable can be used as far as the material can have the state-space representation.

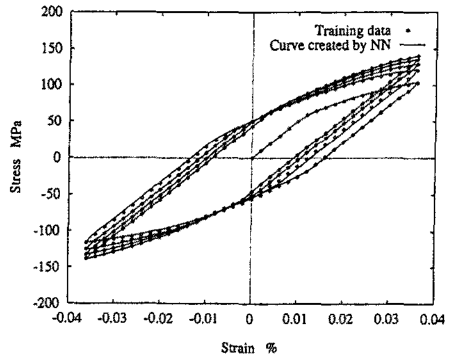


Fig. 7. Stress-strain plots created by training data and the curve created by neural network

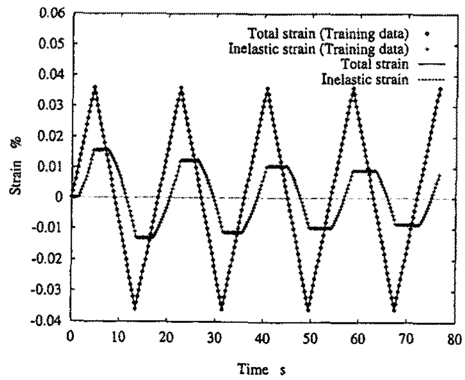


Fig. 8. Training data of stresses and corresponding curve created by neural network

4. Numerical Examples

The error development of the training and validation set until 10,000 trainings is shown in Fig. 6. Clearly, the error is approaching to zero, indicating that the neural network is learning the material law. Figs 7-9 show stress-strain curve, strain-time and stress-time curves of the training data and the corresponding curves created by the neural network. It is seen that the curve by the network is well correlated with the training data, indicating that the neural network formulation can mimic Chaboche's model

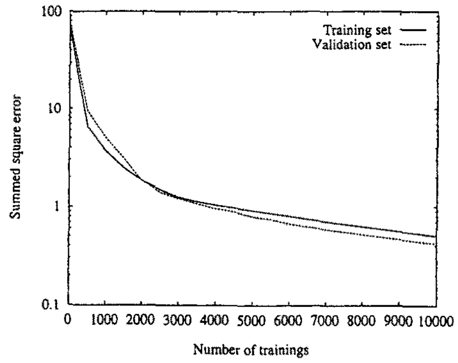


Fig. 6. Error development of training and validation data (Case I)

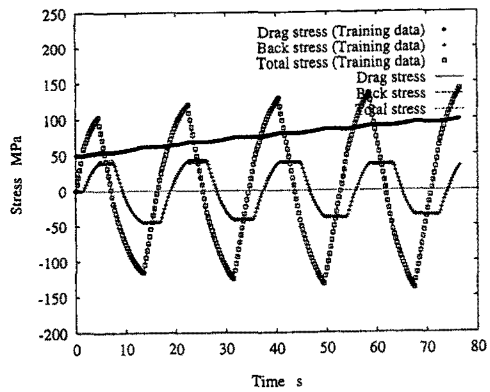


Fig. 9. Training data of strains and corresponding curve created by neural network

Now that we found that the proposed network could reproduce the training data, we will investigate the interpolative capability of the network. For example, Fig. 10 and Fig. 11 shows the strain-time and stress-time curves by Chaboche's model, and their equivalence created by the neural network, with cyclic strain range of $\pm 0.04\%$. This result indicates that the neural network can create a curve similar to the exact curve extrapolatively is the extrapolation is adjacent. However, the peak of the second cycle of back stress shows large errors, indicating that there is no guarantee in extrapolation.

Although the training and validation data depicted in Fig. 7 were used for training, the outcome from the trained network extremely bad. Therefore, the network was trained with the training data obtained at every computer simulation during the material tensile behavior.

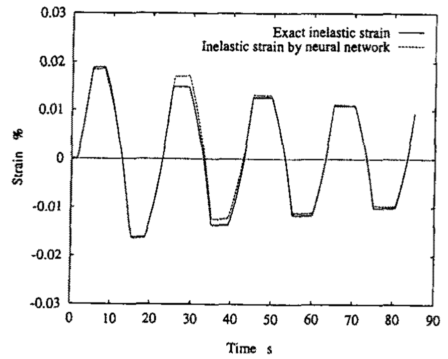


Fig. 10. Exact strain curve and curve by neural network (Case I, Maximum strain range : 0.04%)

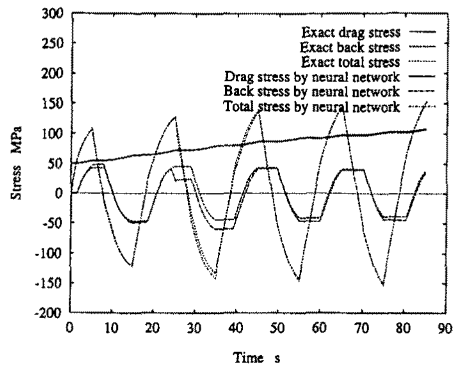


Fig. 11. Exact stress curves and curves by neural network (Case I, Maximum strain range : 0.04%)

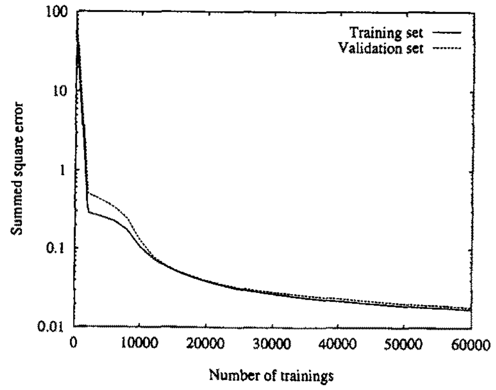


Fig. 12. Error development of training and validation data (Case II, Maximum strain range: 0.04%)

The error development of the training and validations sets until 60,000 trainings is shown in Fig. 12. However, Fig. 12 illustrate that the neural network curve deviates from the exact curve. The reason for the deviation can be easily explained from the curves of the inelastic strain and back stress in Fig. 13, which increases its deviation from the exact curves as time goes.

5. Conclusions

Two neural networks as a material model, which are based on the state-space method, have been proposed.

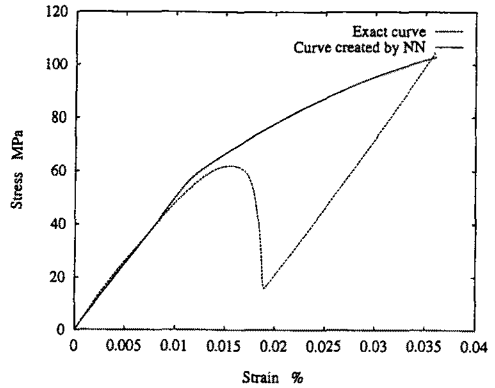


Fig. 13. Exact stress-strain curve and curve created by neural network (Case II, Maximum strain range: 0.04%)

One outputs the rates of inelastic strain and material internal variables whereas the outputs of the other are the next state of the inelastic strain and material internal variables. Both the neural networks were trained using input-output data generated from Chaboche's model and successfully converged. The former neural network could reproduce the original stress-strain curve. The neural network also demonstrated its ability of interpolation by generating untrained curve. It was also found that the neural network can extrapolate in close proximity to the training data. Therefore, the neural network can replace Chaboche's model completely by its interpolative capability if more training data with different conditions are used. This obviously means that the network is superior to all the existing explicit constitutive models by the fact that it is not governed by the explicitness of the model equations and their internal parameters. The curve created by the latter network, in contrast, deviated from the training data significantly. In conclusion, the former neural network has shown its validity as an powerful material model.

Training data for the proposed neural network nevertheless cannot be easily obtained from the actual experimental data. The next version will refer to strategies for extracting the training data from experiments.

References

[1] Thomas B.B, *Experimental Research in Evolutionary Computation*, Springer Berlin Heidelberg, 2006.

[2] Furukawa, T., J.S. Lee and E.C. Lee, "Constitutive Parameter Identification of Inelastic Equations Using an Evolutionary Algorithm", *Journal of Korean Institute of Intelligent Systems*, in Print.

[3] Yamamoto, K., "Modeling of Hysteretic Behavior with Neural Network and its Application to Non-linear Dynamic Response Analysis", *Applications of Artificial Intelligence in Engineering*, Computational Mechanics Publications and Elsevier Applied Science, pp. 475-486, 2002.

[4] Ghaboussi, J., Garrett Jr., J.H., and Wu, X., "Knowledge-Based Modeling of Material Behavior with Neural Networks", *Journal of Engineering Mechanics*, Vol. 117, No. 1, pp. 132-153, 1991.

[5] Miyazaki, H., "Inelastic Analysis Using a Neural Network Constitutive Law", *M.Sc. Thesis*, The University of Tokyo, 1999.

[6] Franklin, G.F., Powell, J.D. and Emami-Naeini, A., *Feedback Control of Dynamic Systems*, Addison Wesley.

[7] G.S. Kumar and A. Kishore, "Generalized State-Space Modeling of Three Hpase Self-Excited Induction Generator for Dynamic Characteristics and Analysis", *Journal of Electrical Engineering & Technology*, Vol. 1, No. 4, pp. 482-489, 2006.

[8] S.R. Rimmalapudi, S.S. Williamson, A. Nasiri and A. Emadi, "Validation of Generalized State Space Averaging Method for Modeling and Simulation of Power Electronic Converters for Renewable Energy Systems", *Journal of Electrical Engineering & Technology*, Vol. 2, No. 2, pp. 231-240, 2007.

[9] Furukawa, T., Okuda, H. and Yagawa, G., "A Neural Network Constitutive Law Based on Yield and Back Stresses", *The 18th Computational Mechanics Conference*, in Print,

[10] J. Lu, C. Feng, S. Xu and Y. Chu, "Observer Design For A Class of Uncertain State-Delayed Nonlinear Systems", *International Journal of Control, Automation, and Systems*, Vol. 4, No. 4, pp. 448-455, 2006.

[11] Goh, C.J., "Model Reference Control of Nonlineare System via Implicit Function Emulation", *Internal Report*, Department of Mathematics, University of Western Australia, 1991.

[12] Liu Puyin, "An Efficient Algorithm of Polygonal Fuzzy Neural Network", *Proc. of 4th WSEAS International Conference on Soft Computing, Optimization, Stimulation and Manufacturing Systems*, April 21-23, USA, 2004.

[13] F.L. Chung and J.C. Duan, "On Multistage Fuzzy Neural Network Modeling", *IEEE Trans. on Fuzzy Systems*, Vol. 8, pp. 125-142, 2000.

Joon-Seong Lee

He received B.Sc. (1986) and M.Sc. (1988) degrees in Mechanical Eng. from SungKyunKwan University, Suwon, Korea. He worked as a professor during 1988-1991 in the Dept. of Mechanical Eng. of Korea Military Academy. He received Ph.D. degree in engineering from the University of Tokyo in 1995. Now he is a professor in the Dept. of Mechanical System Engineering of Kyonggi University. His current research interest is the development of an intelligent simulation system that will effectively combine numerical simulations such as the finite element method and soft computing techniques, and its application to various artifacts such as micromachines.

Phone : 031-249-9813
Fax : 031-244-6300
E-mail : jslee1@kyonggi.ac.kr

Eun-Chul Lee

He received B.Sc. degree in Mechanical Engineering from Kyonggi University, Korea, in 2007. He is currently Master course in the same Department. His research interests include intelligent systems, fuzzy control and application engineering, etc.

Phone : 031-249-9818
Fax : 031-244-6300
E-mail : lec9853@hanmail.net

Tomonari Hurukawa

He received B.Sc. degree in Mechanical Engineering from Waseda University, Tokyo, Japan, in 1991, the M.Sc. degree in Mechanical Engineering from The University of Sydney, Australia. in 1993, and the Ph.D. degree in Mechanical System Engineering from the University of Tokyo, Japan, in 1996. From 1997-1998, he was a lecturer at the University of Tokyo. He is currently Assistant Professor in the Department of Mechanical Engineering, the University of New South Wales, Sydney, Australia. His research interests are in the areas of Fuzzy rule, Genetic Algorithm, and Neural network problems.