Journal of the Korean Nuclear Society
Volume 30, Number 3, pp.222~234, June 1998

On-line Estimation of DNB Protection Limit via a Fuzzy Neural Network

Man Gyun Na
Chosun University
375 Seoseok-dong, Dong-gu, Kwangju 501-759, Korea

(Received July 25, 1997)

Abstract

The Westinghouse $OT\Delta T$ DNB protection logic heavily restricts the operation region by applying the same logic for a full range of operating pressure in order to maintain its simplicity. In this work, a fuzzy neural network method is used to estimate the DNB protection limit using the measured average temperature and pressure of a reactor core. Fuzzy system parameters are optimized by a hybrid learning method. This algorithm uses a gradient descent algorithm to optimize the antecedent parameters and a least-squares algorithm to solve the consequent parameters. The proposed method is applied to Yonggwang 3&4 nuclear power plants and the proposed method has 5.99 percent larger thermal margin than the conventional $OT\Delta T$ trip logic. This simple algorithm provides a good information for the nuclear power plant operation and diagnosis by estimating the DNB protection limit each time step.

1. Introduction

The core protection system philosophy is to define a region of permissible operation in terms of power, pressure, temperature, flow rate and 3-D power distribution, and to trip the reactor automatically when the limits of this region are approached. The protection system of the conventional pressurized water reactor designed by Westinghouse is an analog system. However, the Korea Standard Nuclear Power Plant (KSNPP) and the recently designed nuclear reactors employ a digital protection system. The CE-type nuclear power plants which KSNPP is based on, employ the Core Protection Calculator System (CPCS) which continuously calculates DNBR and Local Power Density (LPD) in order to assure that the specified acceptable fuel design limits on DNB and centerline melt are not exceeded during anticipated operational occurrences. The CPCS has approximately 6,000 constants and the CPCS is designed by deciding the CPCS constants [1]. This large number of constants makes the software V&V (Verification and Validation) more difficult.

The DNB correlations provide the expected value of fuel rod surface heat flux that will cause DNB for various coolant conditions and flow geometries. The ratio of the expected DNB heat flux to the actual fuel rod heat flux at a particular time during an incident is called the DNBR at that time. A correlation limit DNBR (e.g., 1.3 for W-3 correlation or 1.22 for ERB-2 correlation) is established based on the variance of the correlation such that there is a 95 percent probability at a 95 percent confidence level that DNB will not occur when the calculated DNBR is at the correlation limit DNBR. The conservative design method that the calculated DNBR is greater than the correlation limit DNBR on the limiting power rod, is established by considering all parameters at fixed conservative values. The variable value design method used in this work establishes $DNBR \geq$ the correlation limit DNBR on the limiting power rod by statistically combining the effects of uncertainties of the input parameters. Therefore, the design limit DNBR (e.g., 1.54) applicable to all Condition I and II events is determined by utilizing the DNBR sensitivities and variances in three input parameter categories : plant operating parameters, nuclear and thermal parameters and fabrication parameters [2].

Since the conventional Westinghouse DNB protection logic is implemented on analog circuits, the logic must be very simple. The Westinghouse $OT\Delta T$ protection logic heavily restricts the operation region by applying the same logic for a full range of pressure in order to maintain its simplicity. However, if the DNB protection logic is implemented in a digital processor, a little complexity may be allowed to increase the thermal (or operation) margin.

The objective of this work is to estimate the DNB protection limit according to operating conditions by using a fuzzy neural network. Fuzzy system parameters such as membership functions and the connectives between layers in a neural network are tuned by a hybrid learning method to minimize the errors between the target values and the trained values. The proposed method was applied to Yonggwang 3&4 nuclear power plants.

2. Design of a Fuzzy Neural Network System

A fuzzy system consists of situation and action pairs. Conditional rules described in if/then statements are generally used. Adapting fuzzy systems for on-line application would be the desirable objective. Such neuronal improvements of fuzzy systems as well as the fuzzification of neural network systems aim at exploiting the complementary nature of the two approaches; the fuzzy and neural network systems. Their composite is usually called as a fuzzy neural network system. Its simple description will be given first and then applied to Yonggwang 3&4 nuclear power plants in the next section.

In a fuzzy system, the i-th rule can be described using the first-order Sugeno-Takagi type [3] as follows :

$$\text{if } x_1 \text{ is } A_{i1} \text{ AND } \cdots \text{ AND } x_m \text{ is } A_{im}, \quad \text{then } y \text{ is } f_i(x_1, \cdots, x_m) \tag{1}$$

where

$x_1, \cdots, x_m$: input values to the fuzzy inference system (m = the number of input),

$A_{i1}, \cdots, A_{im}$: antecedent membership functions of each input variable for the i-th rule [i = 1, ···, n(= the number of rules)],

$$f_i(x_1, \cdots, x_m) = \sum_{j=1}^{m} q_{ij}x_j + r_i \text{ : the output value of the } i\text{-th rule.} \tag{2}$$

Generally, there is no restriction on the shape of a membership function. In this paper, the following symmetric Gaussian membership function is used :

$$w_{ij}(x_j) = e^{-\frac{(x_j - c_{ij})^2}{2\sigma_{ij}^2}}, \tag{3}$$

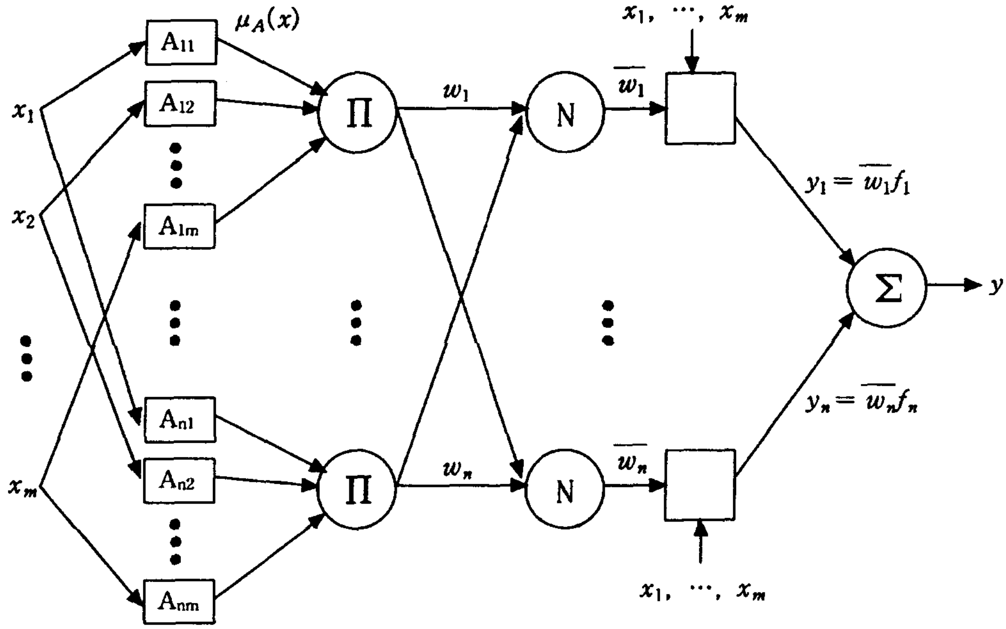

Fig. 1. Fuzzy-neural Network.

where c_{ij} is the center position of a peak of a membership function for the i-th rule and the j-th input and σ_{ij} is the sharpness for the i-th rule and the j-th input.

The output of an arbitrary i-th rule f_i is composed of the first-order polynomial of inputs as given in eq. (2). Then, from the weight average method, the output of the fuzzy inference with n rules is given as follows :

$$y = \sum_{i=1}^{n} \overline{w}_i \cdot f_i, \tag{4}$$

where

$$\overline{w}_i = \frac{w_i}{\sum_{i=1}^{n} w_i}, \tag{5}$$

$$w_i = \prod_{j=1}^{m} \mu_{ij}(x_j) \tag{6}$$

The fuzzy neural network system described above is shown in Fig. 1. The system architecture consists of five layers with as follows :

Layer 1 : generate the membership grades ($\mu_{ij}(x_j)$),

Layer 2 : calculate the firing strengths of each rule via multiplication (refer to eq. (6)),

Layer 3 : calculate the ratio of the i-th rule's firing strength to the sum of all rules' firing strengths (refer to eq. (5)),

Layer 4 : calculate rule outputs based on the consequent parameters ($\overline{w}_i \cdot f_i$),

Layer 5 : sum all the inputs from layer 4 (refer to eq. (4)).

Fuzzy system parameters such as membership functions and the connectives between layers in a fuzzy neural network must be optimized for good performance. This is accomplished by adapting the antecedent parameters (membership function parameters) and consequent parameters (the polynomial coefficients of the consequent part) so that a specified objective function is minimized.

The adaptation methods of most fuzzy neural systems rely on the gradient-descent optimization [4]-[7]. However, in this work, a hybrid learning method which Jang [8] had proposed, was used to optimize the antecedent and consequent parameters because the method shows better performance than any other method. The method combines the backpropagation gradient descent method and the least-squares method. It uses a gradient descent algorithm to optimize the antecedent parameters and a least-squares algorithm to solve the consequent parameters. When an output layer of the network consists of a linear combination of the previous layer' s output, the weights can be solved by using the least-squares method rather than trained iteratively with the backpropagation algorithm. Since it uses two different algorithms, the training method is called a hybrid learning method. The consequent parameters are updated first using a least-squares method and then the antecedent parameters are updated by backpropagating the errors that still exist.

The algorithm which the consequent parameters is updated by, will be introduced first. Equation (4) can be expressed as follows :

$$y = \overline{w}_1(x_1 q_{11} + \cdots + x_m q_{1m} + r_1) + \overline{w}_2(x_1 q_{21} + \cdots + x_m q_{2m} + r_2) + \cdots + \overline{w}_n(x_1 q_{n1} + \cdots + x_m q_{nm} + r_n). \quad (7)$$

Also, equation (7) can be written in different form by using vectors :

$$y = [\overline{w}_1 x_1 \cdots \overline{w}_n x_1 \cdots \cdots \overline{w}_1 x_m \cdots \overline{w}_n x_m \ \ \overline{w}_1 \cdots \overline{w}_n] \begin{bmatrix} q_{11} \\ \vdots \\ q_{n1} \\ \vdots \\ q_{1m} \\ \vdots \\ q_{nm} \\ r_1 \\ \vdots \\ r_n \end{bmatrix}. \quad (8)$$

In case that there exist N data pairs (x_1, $x2$, $\cdots$, x_m, y), the output y and inputs x_1 , $\cdots$, x_m are N-dimensional column vectors. Therefore, equation (8) can be expressed as follows :

$$Y = XW, \quad (9)$$

where

Y : a N-dimensional column vector,
X : a $N \times [(m+1)n]$ matrix,
W : a $(m+1)n$-dimensional column vector which consists of weights,
n : the number of rules,
m : the number of inputs,
N : the number of data pairs.

If there exists the inverse of the matrix X, the weight vector W which consists of the consequent parameters can be solved easily. However, there does not usually exist its inverse. For example, it is necessary that the matrix X should be a square matrix for the matrix invertible. However, if it is not a square matrix, we can solve the vector W by using its pseudoinverse. Therefore, a pseudoinverse can be in general used to solve the matrix W :

$$W = (X^T X)^{-1} X^T Y \quad (10)$$

This method involves inverting $X^T X$, which can cause numerical problems when the columns or rows of $X^T X$ are dependent. However, since we use DNB data at different average temperature and pressure of a reactor, there does not exist a problem like this.

The gradient decent algorithm is used where the antecedent parameters are updated by backpropagating the errors that still exist after the consequent parameters were adapted. The gradient descent method optimizes the parameters of the membership functions by minimizing the objective function defined as follows :

$$E=\frac{1}{2}\sum_{p=1}^{N}(y_{rp}-y_p)^2=\sum_{p=1}^{N}E_p, \quad (11)$$

where

N : the number of data pairs,

y_{rp} : the target output for the p-th input data $(x_1, \cdots, x_m)$,

y_p : the calculated output from a fuzzy neural network for the same input data,

$$E_p=\frac{1}{2}(y_{rp}-y_p)^2 \quad (12)$$

To minimize the above objective function with respect to the antecedent parameters, the following gradient descent algorithm is applied :

$$a_{ij}(k+1) = a_{ij}(k)-\frac{\eta_a}{N}\sum_{p=1}^{N}\frac{\partial E_p}{\partial a_{ij}}, \quad (13)$$

where

a_{ij} : arbitrary membership function parameters c_{ij}, and σ_{ij},

k : learning step,

η_a : the learning rate for parameter a_{ij},

the subscript 'i' denotes the rule number and 'j' denotes the antecedent number which is the same as the input number of the fuzzy neural network.

In order to update the antecedent parameters, it is required to evaluate $\frac{\partial E_p}{\partial a_{ij}}$. The following relationship for the adaptation of the membership function parameters (antecedent parameters) is used :

$$\frac{\partial E_p}{\partial a_{ij}} = \frac{\partial E_p}{\partial y_p}\frac{\partial y_p}{\partial w_i}\frac{\partial w_i}{\partial w_{ij}}\frac{\partial w_{ij}}{\partial a_{ij}}, \quad (14)$$

where

$$\frac{\partial E_p}{\partial y_p} = -(y_{rp}-y_p), \quad (15)$$

$$\frac{\partial y_p}{\partial w_i} = \frac{(f_i-y_p)}{\sum_{i=1}^{n}w_i}, \quad (16)$$

$$\frac{\partial w_i}{\partial w_{ij}} = \frac{w_i}{w_{ij}(x_j)}. \quad (17)$$

In eq. (14), $\frac{\partial w_{ij}}{\partial a_{ij}}$ depends on the input membership function shape. Since the symmetric Gaussian membership functions is used as mentioned previously, the following derivative is obtained in order to update the parameter c_{ij} :

$$\frac{\partial w_{ij}}{\partial c_{ij}} = w_{ij}\frac{(x_j-c_{ij})}{\sigma_{ij}^2}. \quad (18)$$

Also, in order to update the parameter σ_{ij}, the following derivative is needed :

$$\frac{\partial w_{ij}}{\partial \sigma_{ij}} = w_{ij}\frac{(x_j-c_{ij})^2}{\sigma_{ij}^3}. \quad (19)$$

As a result, the parameter adaptation is performed by the following equations :

$$c_{ij}(t+1) = c_{ij}(t)+\frac{\eta_a}{N}\sum_{p=1}^{N}[y_{rp}-y_p] \times (f_i-y_p)\overline{w_i}\frac{(x_j-c_{ij})}{\sigma_{ij}^2}, \quad (20)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t)+\frac{\eta_\sigma}{N}\sum_{p=1}^{N}[y_{rp}-y_p] \times (f_i-y_p)\overline{w_i}\frac{(x_j-c_{ij})^2}{\sigma_{ij}^3}. \quad (21)$$

The gradient decent algorithm depends much on the learning rate and the learning rate is usually updated from the previous learning rate as follows :

$$\text{if } E_p(k) < E_p(k-1), \text{ then } \eta(k) = 1.1\eta(k-1),$$
$$\text{or if } E_p(k) \geq E_p(k-1), \text{ then } \eta(k) = 0.8\eta(k-1). \quad (22)$$

Therefore, the consequent parameters are solved by eq. (10) and the antecedent parameters are calculated by eqs. (20) and (21). Using the optimized parameters, the ΔT (the temperature difference between the hot leg and the cold leg) value is calculated which DNB may take place at a measured pressure and a measured average temperature. The ΔT protection limit is established based on the measurement errors and the variance of the estimated value so that there is a 95 percent probability at a 95 percent confidence level that DNB will not occur when the calculated ΔT is at the ΔT protection limit. Therefore, the ΔT protection limit is defined as follows :

$$\Delta T_{sp} = \overline{\Delta T} - \varepsilon \Delta T_0 - 1.645\sigma \quad (23)$$

where

$\overline{\Delta T}$: the ΔT calculated by the proposed algorithm,

ΔT_o : the rated ΔT,

ε : measurement uncertainty (refer to table 1),

σ : the standard deviation of difference between the actual values and the estimated values.

When the measured ΔT is greater than ΔT_{sp}, the nuclear power plant is tripped.

3. Application to Yonggwang 3 & 4

The applied operation period is the first fuel cycle of Yonggwang 3&4 Units. The design critical heat flux correlation is CE-1 and the actual correlation limit DNBR is 1.19. The design limit DNBR is 1.38 (including 1.8% rod bow and 1% HID grid factors) for typical cell of Yonggwang 3&4 units (refer to [1]). However, in this work, COBRA code with the W-3 correlation was used for DNBR calculation and the uncertainty of the W-3 is greater than that of CE-1. The actual correlation limit DNBR for the W-3 correlation is 1.3. Although the different code was used, all the DNBR input parameters were assumed to have the same sensitivities and uncertainties as calculated by CETOP code (refer to table 2) and it is considered to have different CHF correlations only. Based on the assumptions, the design limit DNBR was assumed to be 1.493 (refer to [1]). Also, in order to generate DNB data, the same assumptions which had been used in obtaining the Westinghouse DNB protection limit were applied. The major assumptions used in calculation are as follows [9] :

Table 1. Measurement Errors of DNBR Calculations [1].

Parameters	Range R	Variance ($\sigma^2 = R^2/12$)
Errors		
Calibration		
1. Calorimetric	4.0%	1.3
2. Tavg (±2°F)	4.9%	2.01
3. Pressure (±8psi)	1.5%	0.19
Signal linearity, reproducibility, and bistable error	10.73%	9.6
Total Variance		13.10 (σ=3.62%)
Setpoint Uncertainty		5.96%(1.645σ)

1) The axial power distribution is an 1.55 chopped cosine shape defined as follows [1] :

$$y = 1.55\cos\left(\frac{0.98654\pi z}{L_o}\right), \quad (24)$$

where

L_o = active core length and $-L_o/2 \leq z \leq L_o/2$.

2) The nuclear enthalpy rise hot channel factor ($F^N_{\Delta H}$) is a design hot channel factor $F^N_{d\Delta H}$ for 100 percent rated or greater power levels, and for power levels less than the rated power $F^N_{\Delta H}$ is given by

$$F^N_{\Delta H}(P) = F^N_{d\Delta H}[1 + 0.3(1-P)], \quad (25)$$

Table 2. DNBR Sensitivities and Uncertainties of Yonggwang 3&4 [1].

Parameter	Nominal Value (μ_i)	Standard Deviation (δ_i)	Sensitivity (S_i)	δ_i / μ_i
1. Primary Coolant Flow Rate	1.0	0.025	1.3937	0.0250
2. Core Power	1.0	0.01	−1.8789	0.0100
3. Core Inlet Temperature [degree F]	564.5	1.5	−8.1070	0.0027
4. Primary System Pressure	2250	30	2.2852	0.0134
5. Nuclear Enthalpy Rise Hot Channel Factor	1.55	0.0243	−1.2455	0.0157
6. Nuclear Enthalpy Rise Hot Channel Factor	1.0	0.015	−0.4492	0.0150
7. Engineering Heat Flux Hot Channel Factor	1.0	0.015	−1.0021	0.0150
8. T/H Code	-	0.025	1.0	0.0250

where P = power level.

3) The coolant flow rate is the design value which is usually about 5% less than the best estimate flow.
4) The bypass flow is excluded from the available core flow.
5) The coolant flow to the hottest coolant channel is reduced by 5 percent.

Except for the above parameters, the other DNB input parameters were considered to be nominal values.

As the core power and inlet temperature vary at the given pressure, the vessel average hot-leg temperature is calculated using COBRA code when the minimum DNBR of the limiting power rod is equal to the design limit DNBR. The inlet and outlet temperatures where the minimum DNBR of the limiting power rod is equal to the design limit DNBR, were given in table 3.

In order to train the fuzzy neural network, the DNB data given in table 3 were used. The number of the network inputs is two and the inputs to the network are the pressure x_1 and average temperature x_2. Also, the target output y_r is ΔT. These inputs and output consist of 189 data pairs (x_1, x_2, y_r).

The membership functions of the input were selected as the Gaussian function as described previously. Therefore, the tuning parameters are the center value c_{ij} and standard deviation s_{ij} of a peak of the Gaussian function.

The following initial values were assumed in order to perform the proposed algorithm :

c_{i1} : randomly distributed between 1730 and 2500$psia$ (i=1, ⋯, n),

c_{i2} : randomly distributed between 570 and 660°F (i=1, ⋯ ,n),

s_{i1} = 50$psia$ (i=1, ⋯, n),

s_{i2} = 10°F (i=1, ⋯, n),

learning rate : η_{c1}=15, η_{c2}=5, η_{s1}=3, η_{s2}= 0.7.

Under the above initial conditions, the number of rules was selected to be 5. When the network is trained 2000 times, the results are shown in Figs. 2∼10. The initial membership functions are shown in Fig. 2. The trained membership functions are shown in Fig. 3. The trained membership functions are very different from the initial ones. Also, the consequent parameters are given in table 4. Figures 4 and 5 show the actual and estimated values. Figure 6 shows the distribution of the errors between the estimated ΔT s and target ΔT s which consist of 189 data. Its distribution is similar to the Gaussian distribution. Based on this distribution, the standard deviation σ is 0.1164°F. In order to have more conservative feature, the measurement uncertainty and the 1.645 σ are subtracted from the estimated ΔT to obtain the DNB protection limit. It is known from Fig. 6 that the calculated values are almost the same as the actual values.

Table 3. DNB Data (Inlet and Outlet Temperature[°F]) for Y/G 3&4 (*DNBR* =1.493).

Power[%] / Press [psia]	118	110	105	100	95	90	85	80	75	Saturated Temp
2415	577.10 644.66	587.45 648.90	593.95 651.51	600.28 654.17	604.75 655.44	609.37 656.78	614.30 658.33	619.30 660.05	624.30 661.94	663.20
2385	575.60 643.47	586.00 647.75	592.51 650.41	598.95 653.09	603.45 654.37	608.07 655.76	613.01 657.33	618.12 659.07	623.15 660.98	661.37
2355	574.12 642.28	584.57 646.62	591.10 649.32	597.67 652.04	602.17 653.32	606.82 654.73	611.77 656.37	616.97 658.13	622.03 659.51	659.51
2325	572.68 641.11	583.17 645.51	589.73 648.23	596.40 651.00	600.92 652.30	605.57 653.71	610.55 655.39	615.77 657.20	620.94 657.65	657.65
2295	571.20 639.97	581.73 644.37	588.37 647.16	595.06 649.97	599.67 651.28	604.37 652.73	609.35 654.41	614.57 655.77	619.85 655.77	655.77
2265	569.72 638.83	580.36 643.29	587.03 646.10	593.73 648.93	598.47 650.29	603.17 651.75	608.18 653.45	613.42 653.82	618.80 653.82	653.82
2235	568.25 637.70	579.03 642.21	585.70 645.04	592.46 647.92	597.27 649.30	602.00 650.79	607.03 651.88	612.29 651.88	617.77 651.88	651.88
2205	566.82 636.61	577.70 641.13	584.40 644.01	591.16 646.89	596.07 648.31	600.83 649.83	605.88 649.92	611.17 649.92	616.71 649.92	649.92
2175	565.39 635.47	576.38 640.05	583.13 643.01	589.90 645.90	594.81 647.31	599.68 647.92	604.74 647.92	610.05 647.92	615.62 647.92	647.92
2145	563.98 634.31	575.08 638.99	581.83 641.95	588.67 644.92	593.59 645.91	598.53 645.91	603.62 645.91	608.95 645.91	614.55 645.91	645.91
2115	562.53 633.12	573.80 637.95	580.59 640.94	587.44 643.91	592.37 643.91	597.41 643.91	602.54 643.91	607.88 643.91	613.49 643.91	643.91
2085	561.12 631.96	572.55 636.93	579.35 639.93	586.24 641.87	591.19 641.87	596.32 641.87	601.47 641.87	606.85 641.87	612.46 641.87	641.87
2055	559.76 630.84	571.25 635.93	578.13 638.93	585.04 639.80	590.03 639.80	595.18 639.80	600.41 639.80	605.82 639.80	611.52 639.80	639.80
2025	558.43 629.75	569.95 634.86	576.93 637.72	583.87 637.72	588.87 637.72	594.05 637.72	599.38 637.72	604.80 637.72	610.55 637.72	637.72
1995	557.10 628.65	568.65 633.79	575.76 635.63	582.70 635.63	587.75 635.63	592.95 635.63	598.35 635.63	603.82 635.63	609.60 635.63	635.63
1965	555.76 627.55	567.35 632.72	574.56 633.38	581.55 633.38	586 60 633.38	591.83 633.38	597.33 633.38	602.87 633.38	608.63 633.38	633.38
1935	554.44 626.47	566.08 631.14	573.39 631.14	580.40 631.14	585.48 631.14	590.73 631.14	596.33 631.14	601.92 631.14	607.68 631.14	631.14
1905	553.14 625.40	564.83 628.89	572.22 628.89	579.30 628.89	584.40 628.89	589.67 628.89	595.31 628.89	600.97 628.89	606.79 628.89	628.89
1875	551.86 624.34	563.63 626.65	571.05 626.65	578.23 626.65	583.35 626.65	588.65 626.65	594.37 626.65	600.08 626.65	605.93 626.65	626.65
1845	550.65 623.35	562.43 624.41	569.90 624.41	577.17 624.41	582.33 624.41	587.68 624.41	593.42 624.41	599.22 624.41	605.10 624.41	624.41
1815	549.43 622.16	561.23 622.16	568.78 622.16	576.17 622.16	581.35 622.16	586.73 622.16	592.52 622.16	598.40 622.16	604.32 622.16	622.16

• hot-leg boiling in the dotted cells

Table 4. Trained Consequent Parameters.

p_{i1}	1.4889e−001	2.4912e−002	1.3239e−001	2.3921e−002	3.4772e−002
p_{i2}	−1.9947e+000	−7.8633e−001	−1.9889e+000	−8.1617e−001	−1.0561e+000
r_i	9.7091e+002	4.8803e+002	1.0005e+003	5.0814e+002	6.3214e+002

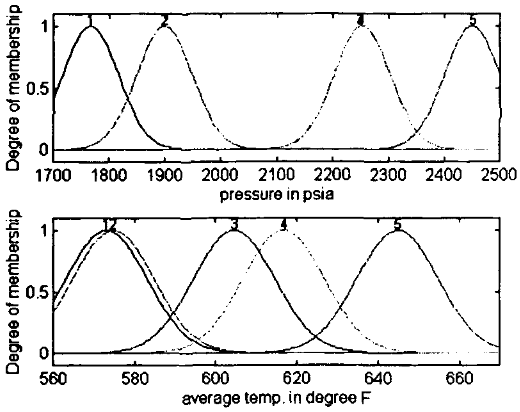

Fig. 2. Initial Input Membership Functions.

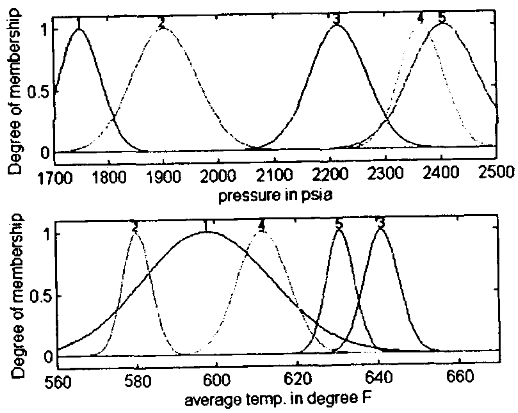

Fig. 3. Final Input Membership Functions.

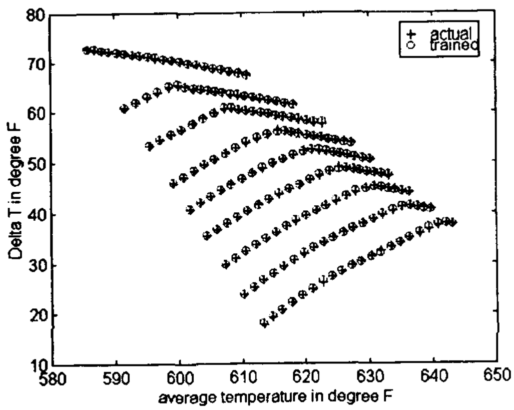

Fig. 4. Comparison of Actual and Trained ΔTs Versus Average Temperature.

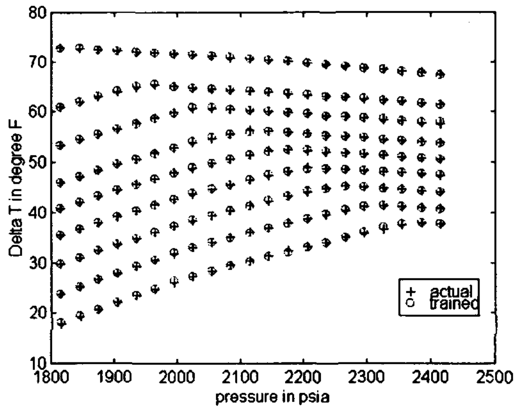

Fig. 5. Comparison of Actual and Trained ΔTs Versus Pressure.

Figure 7 shows the sum of the squared-errors between the estimated ΔT s and the actual ΔT s at each learning step and Fig. 8 shows the maximum error at each step. As the learning keeps going, *these values decrease gradually.* These values increase at some steps but the gradient descent algorithm is stabilized by adjusting the learning rates (refer to eq. (22)).

Figures 9 and 10 show the results after the proposed algorithm was applied to data which had *not been used for the learning.* Only table 3 is used for learning of this algorithm. DNB

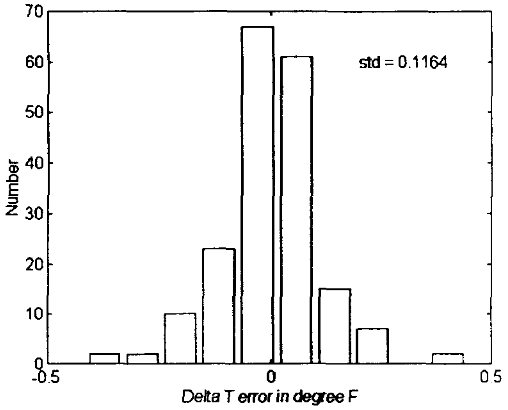

Fig. 6. Distribution of the Errors Between the Estimated Values and the Actual Ones.

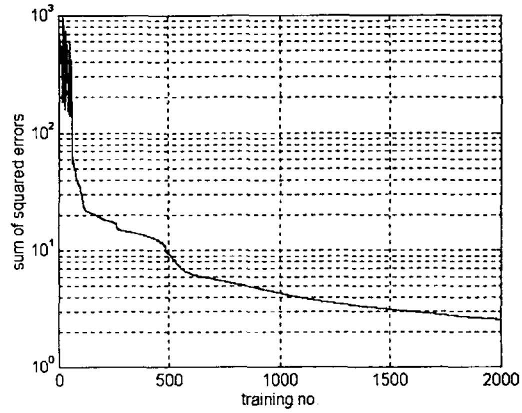

Fig. 7. Sum of the Squared Errors of Trained Values Versus Learning Step.

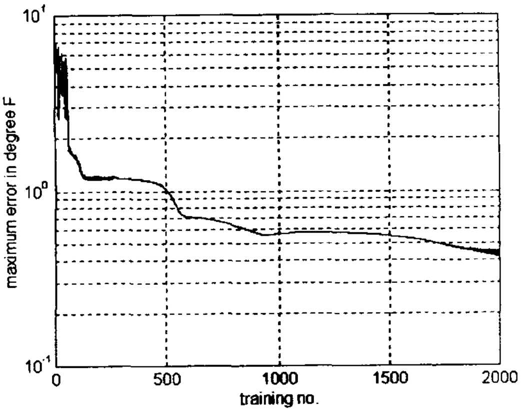

Fig. 8. Maximum Error of Trained Values Versus Learning Step.

protection limit at pressures 2250*psia* and 2000 *psia* which do not exist in table 3, were calculated from the trained network. From the figures, although the algorithm was applied to the arbitrary measured pressure and average temperature, it is known that it gives an accurate DNB protection limit.

A steady-state thermal margin was compared between the Westinghouse DNB protection system and the proposed one. The thermal margin may be defined as $\Delta T_{sp}/\Delta T_0$ at nominal cold leg temperature and RCS pressure. The nominal cold leg temperature and RCS pressure are 564.5°F and 2250 *psia*, respectively. The rated ΔT, ΔT_0 is given as 56.5°F and the ΔT protection limit calculated by the proposed algorithm is 69.365°F at the nominal cold leg temperature and RCS pressure. Therefore, the thermal margin of the proposed algorithm is 122.77%.

For an 1.55 chopped cosine shape at steady-state condition, the Westinghouse OTΔT trip setpoint is determined from the following equation :

$$\Delta T_{sp} = \Delta T_0 [K_1 - K_2 (T_{avg} - T_{avg0}) + K_3(P - P_0)], \quad (26)$$

where

ΔT_{sp} = setpoint value of ΔT,

ΔT_0 = indicated ΔT at nominal plant conditions,

T_{avg} = measured average coolant temperature,

T_{avg0} = reference average coolant temperature at nominal plant conditions of rated power,

P = measured RCS pressure,

P_0 = reference RCS pressure at nominal plant conditions of rated power,

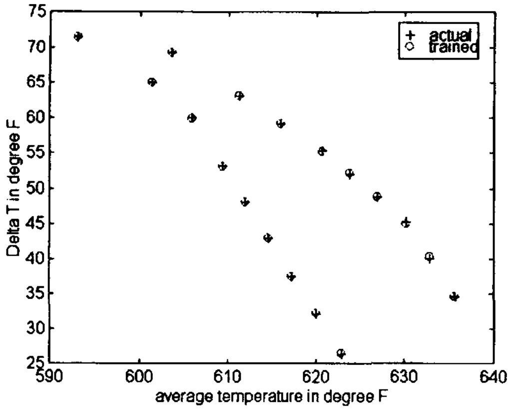

Fig. 9. Comparison of Actual and Trained ΔTs Versus Average Temperature Using Nontrained Data (Data at 2000 and 2500 Psia).

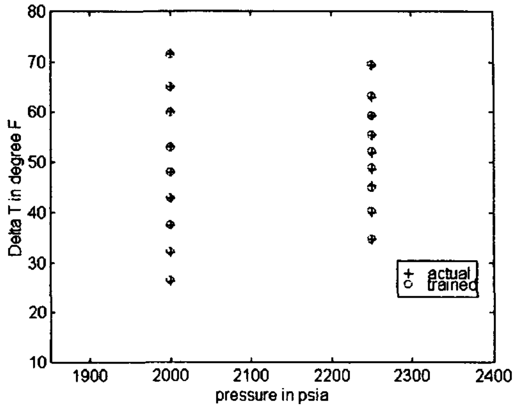

Fig. 10. Comparison of Actual and Trained ΔTs Versus Pressure Using Nontrained Data (Data at 2000 and 2500 Psia).

K_1 is a preset manually adjustable bias, and K_2 and K_3 are preset manually adjustable gains.

The constants K_1, K_2 and K_3 can be determined by solving three simultaneous equations derived from the four intersection points (refer to [10]). Each resulting equation is tested for various pressures inside the defined pressure region in order to assure that all the DNB limits are covered. Generally, two of the equations derived from the two intersection points are found to provide protection over each pressure range. The final equation is selected based on maximum available operating margin.

The final OTΔT setpoint is determined by adjusting K_1 based on appropriate allowances for uncertainties and equipment and measurement errors (refer to table 1). The thermal margin is calculated by substituting the following equation into Eq. (26) :

$$T_{avg} = \frac{\Delta T}{2} + T_c = \frac{\Delta T_{sp}}{2} + T_c, \quad (27)$$

where

T_c = cold-leg temperature.

Since $(T_{avg} - T_{avg0}) = \frac{\Delta T_0}{2}\left[\frac{\Delta T_{sp}}{\Delta T_0} - 1\right]$, the thermal margin of the Westinghouse DNB protection system may be written as

$$\frac{\Delta T_{sp}}{\Delta T_o} = \frac{K_1 + K_2 \cdot \frac{\Delta T_o}{2}}{1 + K_2 \cdot \frac{\Delta T_o}{2}}, \quad (28)$$

where

K_1 = 1.2382, (from ref. [10] and table 1)

K_2 = 0.014846, (from ref. [10])

ΔT_0 = 56.5°F.

The conventional $OT\Delta T$ trip logic has the thermal margin 116.78 percent. The proposed method has 5.99% larger thermal margin than the conventional $OT\Delta T$ trip logic.

4. Conclusions

A fuzzy neural method was applied to estimate the DNB protection limit using the measured average temperature and pressure. Fuzzy system parameters such as the membership functions and the connectives between layers in a neural network are optimized by a hybrid learning method. The learning method uses a gradient descent algorithm to optimize the antecedent parameters and a least-squares algorithm to solve the consequent parameters. The consequent parameters are updated first using a least-square method and then the antecedent parameters are updated by backpropagating the errors that still exist.

The network was trained by using 189 DNB data of Yonggwang 3 and 4 units. The DNB data are the inlet and outlet temperatures where the minimum DNBR of the limiting power rod is equal to the design limit DNBR at a given pressure. The inputs to the fuzzy neural network are the average temperature and pressure of the reactor core and the output is the temperature difference between inlet and outlet ΔT. Therefore, the ΔT, which induces DNB at a given average temperature and pressure, is estimated from this algorithm. The measurement error and the uncertainty of the estimation algorithm are subtracted from the estimated ΔT in order to establish the setpoint ΔT so that this algorithm has some conservative features.

The proposed algorithm has 5.99 percent larger thermal margin than the conventional $OT\Delta T$ logic. It is recommended to accomplish more realistic and exact DNB protection limit by adding the coolant flow rate and axial shape to the input of the fuzzy neural network using the DNB data on the coolant flow rate. Because the estimation algorithm gives sufficient conservative values with more than 95%/95% probability/confidence level although this algorithm is implemented by the fuzzy and neural-network method, it is desired that one is dazzled by the term 'fuzzy'.

Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Center for Advanced Reactor Research at Korea Advanced Institute of Science and Technology.

References

1. G.S. Auh, D.H Hwang, and S.H. Kim, "A Steady-State Margin Comparison Between Analog and Digital Protection Systems," J. KNS, Vol. 22, No. 1, pp. 45-57 (1990).
2. H. Chelemer, L.H. Boman and D.R. Sharp, "Improved Thermal Design Procedure," WCAP-8567 (1975).
3. T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Trans. System, Man, Cybern.*, Vol. 1, pp. 116-132, 1985.
4. H. Takagi, et. al., " NN-Driven Fuzzy Reasoning," Intl. J. Approximate Reasoning, Vol. 5, No. 3, pp. 191-212 (1991).
5. I. Hayashi, et. al., "Construction of Fuzzy Inference Rules by NDF and NDFL," Intl. J. Approximate Reasoning, Vol. 6, pp. 241-266 (1992).
6. H. Ichihashi, et. al., "Computed Tomography by Neuro-fuzzy Inversion," in Proceedings of 1993 Intl. Joint Conference on Neural Networks, Part 1, Nagoya, Oct. 25-29, pp. 709-712 (1993).
7. H. Nomura, et. al., "A Self Tuning Method of Fuzzy Reasoning by Genetic Algorithms," in Fuzzy Control Systems, A. Kandel and G. Langholz, eds., CRC Press, Boca Raton, FL,

pp. 338-354 (1994).
8. J. S. Jang, "Self-learning Fuzzy Controller Based Inference Systems," IEEE Trans. on Neural Network, Vol. 3, No. 5, pp. 714-723 (1992).
9. K.I. Han, "$OP\Delta T$ and $OT\Delta$T Trip Setpoint Generation Methodology," J. KNS, Vol. 16, No. 2, pp. 106-115 (1984).
10. Man Gyun Na, A Study on the Enhancement of the DNB Protection Logic, Center for Advanced Reactor Research, CARR/AI&C, Feb. (1997).