Nuclear Engineering and Technology 51 (2019) 723–730

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Nuclear reactor vessel water level prediction during severe accidents using deep neural networks

Young Do Koo [a], Ye Ji An [a], Chang-Hwoi Kim [b], Man Gyun Na [a, *]

[a] Department of Nuclear Engineering, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju, 61452, Republic of Korea
[b] Nuclear ICT Research Division, Korea Atomic Energy Research Institute, 989-111 Daedeok-daero Yuseong-gu, Daejeon, 34039, Republic of Korea

ARTICLE INFO

Article history:
Received 28 October 2018
Received in revised form
14 December 2018
Accepted 24 December 2018
Available online 27 December 2018

Keywords:
Deep neural networks
Genetic algorithm
Reactor vessel water level
Safety-critical instrumentation signals
Severe accident

ABSTRACT

Acquiring instrumentation signals generated from nuclear power plants (NPPs) is essential to maintain nuclear reactor integrity or to mitigate an abnormal state under normal operating conditions or severe accident circumstances. However, various safety-critical instrumentation signals from NPPs cannot be accurately measured on account of instrument degradation or failure under severe accident circumstances. Reactor vessel (RV) water level, which is an accident monitoring variable directly related to reactor cooling and prevention of core exposure, was predicted by applying a few signals to deep neural networks (DNNs) during severe accidents in NPPs. Signal data were obtained by simulating the postulated loss-of-coolant accidents at hot- and cold-legs, and steam generator tube rupture using modular accident analysis program code as actual NPP accidents rarely happen. To optimize the DNN model for RV water level prediction, a genetic algorithm was used to select the numbers of hidden layers and nodes. The proposed DNN model had a small root mean square error for RV water level prediction, and performed better than the cascaded fuzzy neural network model of the previous study. Consequently, the DNN model is considered to perform well enough to provide supporting information on the RV water level to operators.

1. Introduction

Acquiring instrumentation signals (also called sensor signals) from nuclear power plants (NPPs) is essential for their safe long-term operation. NPPs comprise various facilities and systems including the reactor coolant system (RCS) (or primary system), which contains the reactor coolant passing through the reactor core. Maintaining the integrity of the NPP can be accomplished by the operators taking necessary action based on the information from many instruments used to monitor and diagnose its state. As the safety of the primary system is traditionally a main concern, various instrumentation signals such as temperature, pressure, water level, neutron flux in the RCS, and hydrogen gas concentration in the containment, are considered as safety-critical signals and accident monitoring variables.

However, acquisition of these safety-critical signals can be constrained on account of reliability degradation or failure of the instruments, and eventually an abnormal state of an NPP can progress to severe accident circumstances by missing the critical moment for the operator's decision for mitigation. For instance, among the safety-critical signals, nuclear reactor vessel (RV) water level, which is the focus of this study, is directly related to the reactor core cooling. Although a heated junction thermocouple (HJTC) is used to measure RV water level during accidents in the Optimized Power Reactor 1000 (OPR1000), its precision can decline when bubbles in the RV are adsorbed onto the HJTC due to the coolant and steam co-existing at the same temperature and pressure.

Therefore, the RV water level was predicted using deep neural networks (DNNs) by applying only a few signals from the NPP, to provide monitoring support information to the operators during severe accidents. The DNN [1] employs the basic artificial neural network (ANN) structure and its performance mainly depends on its hidden layer scale. Since the number of hidden layers varies according to the subject or data type, many attempts are required to determine the optimal number of layers. In this study, the numbers of hidden layers and nodes of the DNNs were selected using a genetic algorithm (GA) [2,3] which is a technique that mimics the

* Corresponding author. Department of Nuclear Engineering, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju, 61452, Republic of Korea.
E-mail address: magyna@chosun.ac.kr (M.G. Na).

https://doi.org/10.1016/j.net.2018.12.019

evolutionary process of living organisms to obtain a near optimal result.

Signal data are required as DNNs are mainly trained through backpropagation and gradient-descent algorithms [4]. However, actual data of NPP accidents rarely exist. Accordingly, the accident data were obtained by simulating loss-of-coolant accidents (LOCAs) at hot- and cold-legs, and steam generator tube (SGT), which may occur in the NPPs, using the modular accident analysis program (MAAP) code [5]. These LOCA data simulated by the MAAP code comprised diverse simulated signals numerically expressed. From the simulated signals, only three were provided as input to the DNNs to predict the RV water level as instrumentation signals cannot be ensured during severe accidents.

In this paper, the prediction performance of the DNN model for the RV water level by employing limited information was evaluated and compared with the performance of the cascaded fuzzy neural network (CFNN) model of the previous study [6].

2. Deep learning method for RV water level prediction

Artificial intelligence (AI) is described as a technology that enables computers to perform tasks such as classification, prediction, and recognition equally well or better than human beings. AI methods applied in various industrial fields perform well in object recognition and classification, voice recognition, and regression problems. Recently, the performance of deep learning methods was enhanced by high-performance hardware, such as upgraded central or graphics processing units, as well as the utilization of efficient computational techniques for training. For these reasons, deep learning is generally considered as a state-of-the-art method among AI methods in a wide range of fields.

In the NPP field, satisfactory state monitoring and diagnosis, and variable prediction were achieved in many studies by mainly using representative machine learning models of AI methods. To check performance of a deep learning method, therefore, DNNs [1] were used to predict a safety-related variable in this study. Although the efficacy of AI methods differs depending on the intrinsic characteristics, deep learning methods are fundamentally based on a neural network structure inspired by the inter-connections between neurons in the human brain. The DNNs proposed in this paper can be considered as a simpler method in the aspects of structure and weight flow as the basic ANN framework is utilized as it is for DNNs.

2.1. Deep neural networks

DNNs can be briefly described as ANNs with multiple hidden layers between the input and output layers to enable complex computation and high performance (refer to Fig. 1). The main features of DNNs are that the three types of layers of the ANN structure are utilized without a need for specific processing layers designed by the users (e. g. fuzzy neural networks (FNNs) [7]), and it is trained from the data using a general-purpose learning algorithm [1]. In this study, the DNNs were trained and optimized by the backpropagation and gradient descent algorithms [4], which is the most common form of the machine learning including deep learning [1]. Specifically, the training and optimization were performed in two steps; forward propagation for calculating predicted values and errors, and backpropagation for updating the weights.

As indicated in Fig. 2, the sum of the values calculated by multiplying the input x_i and initial weight w_{ij} for each node in the input layer was transferred to the first hidden layer. These values were used for computing z_j, the outputs of the nodes in the first hidden layer, by applying an activation function.

$$z_j = f\left(\sum\left(w_{ij}x_i + b_j\right)\right) \tag{1}$$

where subscripts i and j denote the indexes of the nodes in the input and first hidden layers, respectively.

From the second hidden layer to the output layer, the outputs of the nodes were computed using calculated values from the nodes in a previous layer and the activation function expressed as Eq. (2). Therefore, it can be the output of the overall DNNs or just the output of a hidden layer depending on the number of all the layers.

$$z_l = f\left(\sum\left(w_{jl}z_j + b_l\right)\right) \tag{2}$$

where subscript l denotes the index of the nodes in the following layer.

The outputs from the nodes in the hidden layers were transferred from the former layers to the following layers by forward propagation and revised iteratively until reaching the output layer. Finally, the values predicted by the DNNs, $\hat{y}$, were calculated in the output layer.

After forward propagation, a cost function was calculated using the errors between the predicted values $\hat{y}$ and the reference (or targeted) values y for backpropagation and gradient descent. Generally, a bowl-shaped convex function is used as the cost function to easily approach the global minimum:

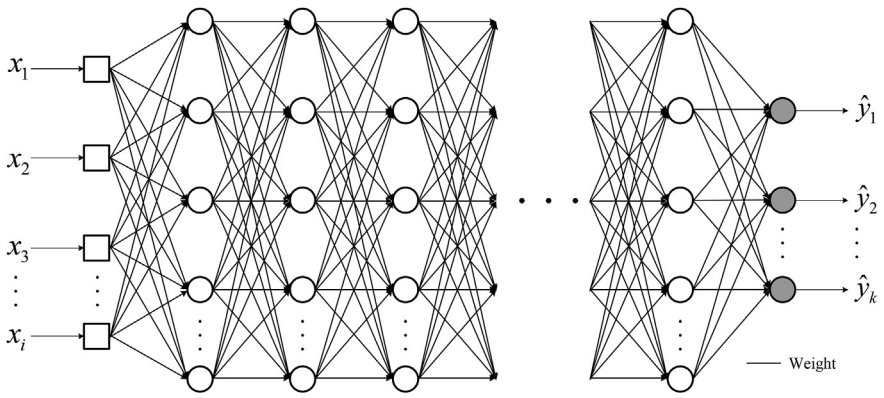

Fig. 1. Example of deep neural networks.

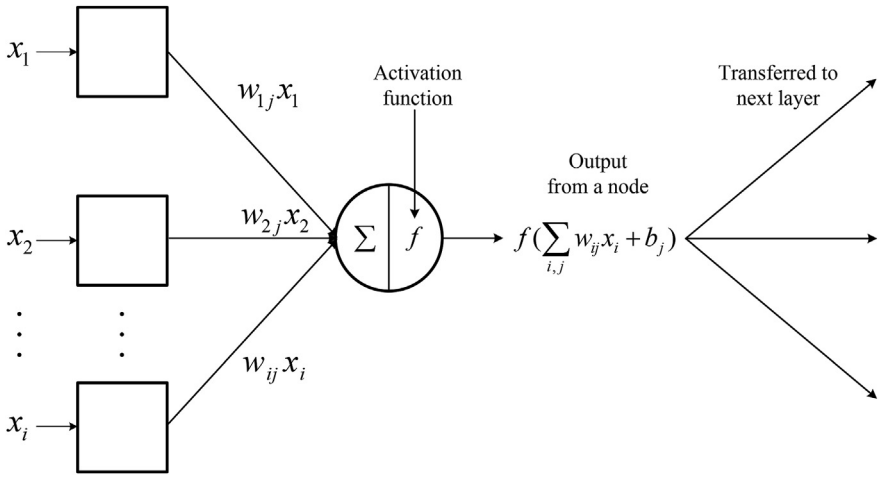

Fig. 2. Single artificial neuron in the first hidden layer.

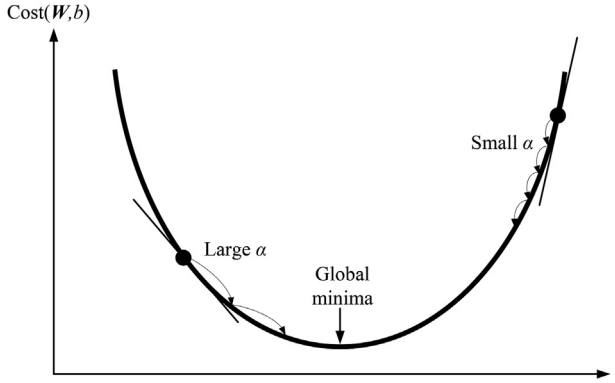

Fig. 3. Weight update by gradient descent in the cost function.

$$Cost(\boldsymbol{W}, b) = E_s = \sum_{k=1}^{n} (\hat{y}_k - y_k)^2 \tag{3}$$

where n is the number of applied data.

The weights established through the forward propagation step were updated using Eq. (4) by propagating E_s in Eq. (3) backwards and calculating the gradient. The gradient descent iterations for the cost function were stopped when a global minimum, the lowest point or a point where the weight is optimized for the learning data in the cost function, was reached (see Fig. 3).

$$w_{ij}^{new} = w_{ij}^{old} - \alpha \frac{dE_s}{dw_{ij}} \tag{4}$$

where the learning rate α ($0<\alpha<1$) controls the gradient descent step size for the cost function. If α is large, the learning pace of the DNNs can be faster, but it can be vulnerable to divergence. On the contrary it can converge into a local minimum if α is too small.

The training of the DNNs in this study continued until the predefined criterion was met or the maximum number of epochs was reached.

2.2. Optimized network structure of DNNs using genetic algorithms

Deep learning algorithms including DNNs have many hyper-parameters in common determining their performances. Therefore, it is important to find the optimal settings in order to establish a deep learning model with the desired accuracy. Many attempts are required to determine these settings since deep learning methods have more hyper-parameters to be considered than machine learning methods. Especially, the numbers of hidden layers and nodes are main parameters influential in the performance and their optimal values generally depend on data provided to the DNNs. In this study, the hidden layers and nodes of the DNNs were automatically optimized using a GA [2,3], instead of manually adjusting them.

The GA utilized to find a solution in a search or optimization problem denotes a technique that artificially models the evolutionary process of living organisms through genetic operations such as selection, crossover, and mutation. As a generation proceeds in the GA, populations of chromosomes are arbitrarily formed and then repeatedly substituted with new populations of chromosomes generated by the genetic operations [8]. In this study, the optimal numbers of hidden layers and nodes were determined when the most suitable candidate solution to a problem had been identified by assessing the chromosomes with the fitness function. The fitness function was utilized to evaluate how well the chromosomes in the populations fitted the objective by assigning scores to each chromosome. As indicated in Eq. (5), the scores for the chromosomes were calculated using the errors from the training procedure of the DNNs employing learning and validation data sets. Fig. 4 shows the optimization process of the DNNs using the GA. The network structure of the DNNs can be optimized over a number of generations.

$$F = \exp(-\lambda_1 E_l - \lambda_2 E_{l\cdot\max} - \lambda_1 E_v - \lambda_2 E_{v\cdot\max}) \tag{5}$$

where E_l and $E_{l\cdot\max}$ are the root mean square error (RMSE) and maximum error for the learning data, and E_v and $E_{v\cdot\max}$ denote the RMSE and maximum error for the validation data, respectively. λ_1 and λ_2 are weighted values for the two types of the errors. The weight values depend on how well we can estimate the reactor vessel water level by using the DNN model. If the RV water level prediction can be modeled easily (that is, the error values are small), the weight values can be large for a proper fitness value. Note that the maximum fitness value is one. Also, λ_2 is smaller than λ_1 as it can be inferred easily.

Although the presented GA technique was computationally intensive [8], the optimal numbers of hidden layers and nodes were determined by the GA. In addition, once the optimized network

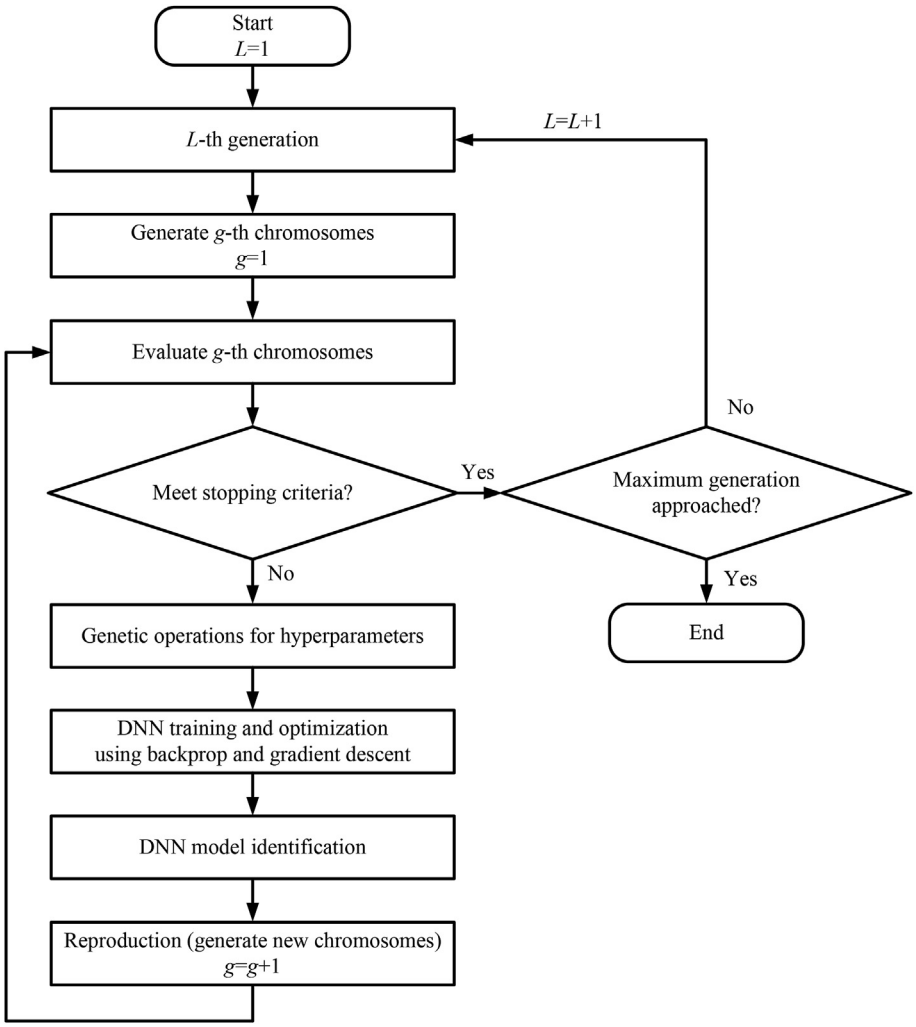

Fig. 4. Optimization procedure for DNN model using genetic algorithm.

structure has been determined, the DNN model performs well by getting close to the global minimum, and its computing time for the prediction problem considerably decreases.

2.3. Other hyper-parameters of DNNs

As the numbers of hidden layers and hidden nodes were properly selected using the GA, the DNN model can obtain the required performance for the objectives. However, the DNN model can be susceptible to the vanishing gradient problem if its network scale is too large. As expressed in Fig. 5, vanishing gradient means that the errors from the training procedure are not well propagated backwards from the final hidden layer to the first one through the deep network. This phenomenon easily happens since activation functions based on multiplication are used in the hidden nodes. S-shaped curve, outputs of the sigmoid function (or logistic function) used as an activation function taper off and its gradients become nearly zero as it passes through the network (refer to Fig. 6). Eventually, training of DNNs can be constrained in large-scale networks due to this conventional problem of deep learning. Thus, the bipolar sigmoid function was used for the activation function of the DNNs in this study as it is less vulnerable to the vanishing gradient in the DNN than the logistic function. The range of derivative of the sigmoid function is between 0 and 0.25 while that of the bipolar sigmoid, of which output range is similar to the hyperbolic tangent function, is between 0 and 1 [9]. Thus, an error calculated from the output layer can be well propagated backward through deeply stack hidden layers using the bipolar sigmoid. Moreover, the DNN model with the bipolar sigmoid function showed better performance than other activation functions including well-known rectified linear unit function for RV water level prediction. Fig. 7 indicates the outputs of the bipolar sigmoid function.

It is known that the performance of DNNs can be improved by applying a large amount of data to the larger-scale neural networks [10]. However, the overfitting problem of the model being overtrained for the learning data (too accurate for the learning data) can occur even in proper hidden layers. To prevent overfitting, cross-validation was applied in consideration of the volume of data used and the network structure optimized by the GA. Cross-validation is a method separating the data into three types, namely learning, validation, and test data. The learning data set that accounts for the highest percentage of the data was used for the DNN model training and the validation data set was utilized to check whether overfitting occurs while in training. Generally, the errors for learning and validation data are the smallest when the DNN model is optimally trained. If the training of the DNN model continues and the error for the validation data set starts to increase,

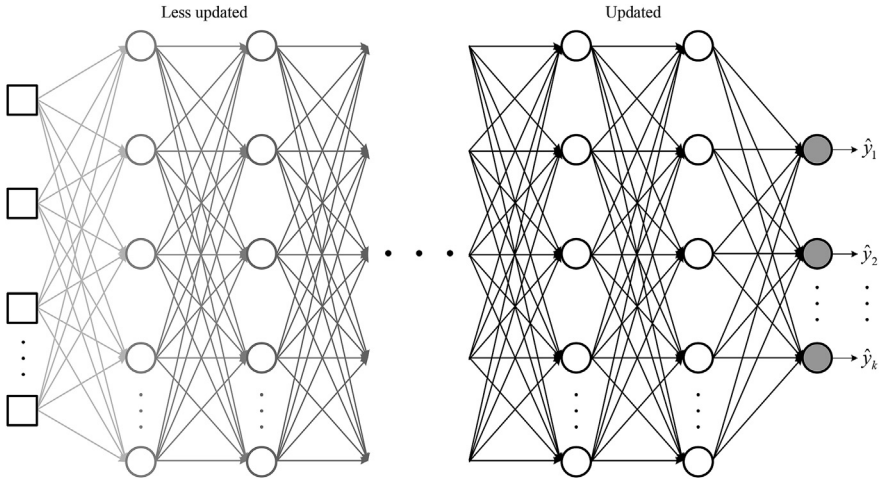

Fig. 5. Illustration of vanishing gradient problem.

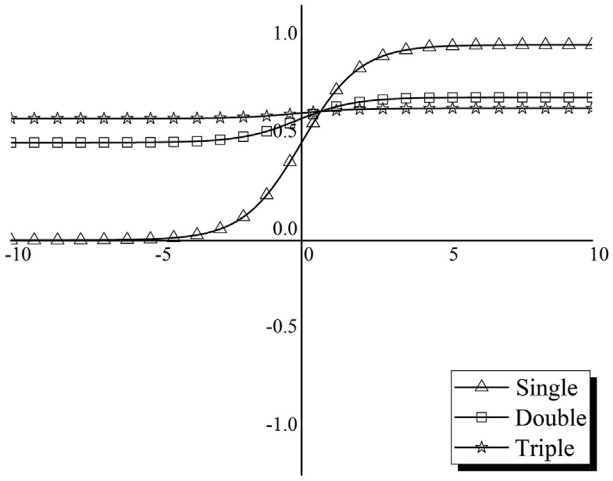

Fig. 6. Outputs of the sigmoid function.

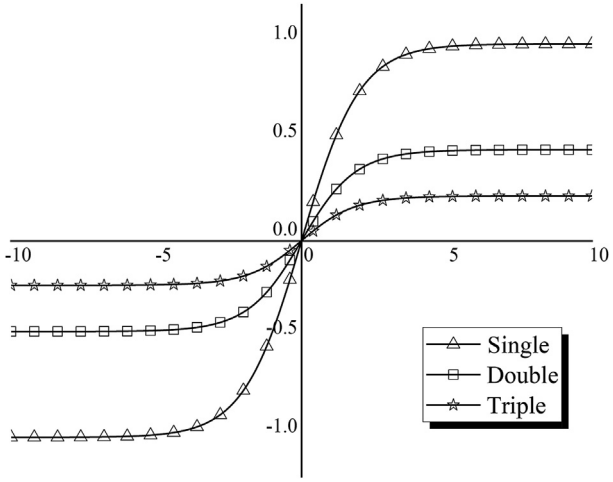

Fig. 7. Outputs of the bipolar sigmoid function.

it is regarded that overfitting occurs beyond this point. The test data set was used to check how well the developed DNN model fitted the independent data. Accordingly, the error for the test data set was considered a measure of performance.

3. Acquired data for DNN model training

3.1. Simulated data for postulated accidents

NPP data are needed to train the DNNs as a RV water level prediction model with required accuracy and to verify the developed DNN model after the training. However, data acquisition is constrained due to a lack of actual accident data of NPPs. Hence simulated data for the postulated accidents such as LOCAs at the hot- and cold-legs and steam generator tube rupture (SGTR) were obtained for the OPR1000 using the MAAP code [5], which is a software tool used for severe accident analysis for pressurized water reactors and boiling water reactors. Various NPP accidents including LOCAs can be simulated according to pre-set accident scenarios to obtain accident data containing information of the NPPs under the corresponding accident circumstances. These obtained simulated data sets were considered as reference or target data and provided as input to the DNNs.

In this study, simulated data were obtained on the LOCAs and SGTRs in accordance with the accident scenario that the safety injection tank and containment spray system function well while the high-pressure and low-pressure safety injection systems did not function normally [5]. A total number of 600 simulated data, comprising 200 hot-leg LOCAs, 200 cold-leg LOCAs, and 200 SGTRs, classified according to the break sizes, were generated. For the LOCAs at the hot- and cold-legs, 30 simulated data were for the smaller break sizes and the rest of the data were for the larger break sizes, respectively. In case of the SGTRs the data were separated into 100 each for the smaller and larger rupture sizes, respectively.

3.2. Simulated instrumentation signals in the data

The simulated data on the LOCAs and SGTRs from the MAAP code consisted of time-integrated values of many simulated signals generated from the facilities in the RCS after the reactor trip using Eq. (6). Simulated signals (elapsed time after the reactor trip,

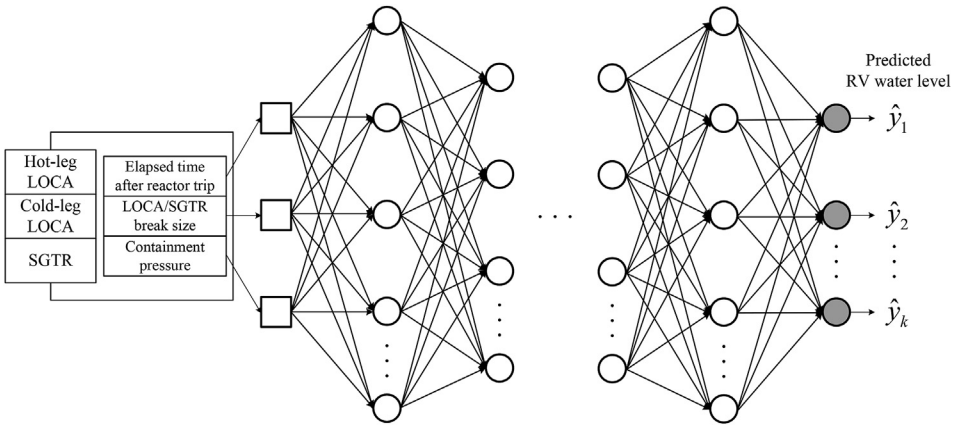

Fig. 8. Prediction of RV water level using DNN model.

estimated LOCA break size, containment pressure, RCS pressure, sump water level, collapsed RV water level, boiled-up RV water level, hydrogen concentration in containment) were included in the simulated data for each postulated accident in this study.

$$x_j = \int_{t_s}^{t_s+\Delta t} g_j(t)dt, \quad j = 1, 2, \cdots, m \tag{6}$$

where t_s is a reactor trip time, Δt is the integration time span, $g_j(t)$ is the type of signal, and m is the number of the used simulated signals. The simulated data on the LOCAs at hot- and cold-legs comprised the time-integrated values from 10 min to 3 days after the reactor trip. The time-integrated values for the signals from 60 min to 3 days after the reactor trip were included in the SGTR data.

As acquisition of a variety of the instrumentation signals from the NPPs can be constrained under severe accident circumstances, a few signals such as the elapsed time after the reactor trip and the containment pressure were provided as inputs to the DNNs (see Fig. 8). The errors of the DNN model were calculated by comparing the predicted RV water levels from the DNN model, as the outputs, with the RV water levels of the acquired simulated data, as the targeted values.

Since the break size and position are not accurately measured and identified in actual LOCA and SGTR circumstances, these factors have to be determined. According to previous studies on LOCA diagnosis [11–13], they can be identified quite accurately. In this study, thus, the estimated LOCA break size (or the estimated SGTR size) was considered as an applicable input to predict the RV water level. In addition, the containment pressure signal was also used as another input for the DNNs as the internal condition of the containment is expected to be in milder condition than that of the RV or reactor coolant boundary [6].

For cross-validation, the values of the three simulated signals in the data were separated into three types of data sets. First, 100 data points for each signal in the LOCAs and SGTRs were selected as the test data at fixed intervals. The validation data points were chosen in a similar way as the test data after the test data points had been removed. The remaining data were used as the learning data points.

4. Prediction results for RV water level using DNN model

4.1. Prediction performance of the DNN model

As mentioned previously, the test data set was used to verify the developed DNN model and its performance was measured by its RMSEs. Table 1 shows the RMSEs and maximum errors of the DNN model for the test and learning data according to the break sizes at each accident location. Most of the RMSEs for the test data were a little larger than those of the learning data while all the maximum errors for the test data were much smaller than those of the learning data. Therefore, it is accepted that a well-trained DNN model was established in this study, which can accurately predict the RV water level. The numbers of hidden layers and hidden nodes as optimized by the GA are also shown in Table 1.

Figs. 9 and 10 indicate the prediction performances of the DNN model for the test data. Fig. 9(a), 9(b), and 9(c) show the prediction results of the RV water level according to the elapsed time after the reactor trip for the LOCAs and SGTRs with small break sizes, respectively. For the SGTR case, the predicted values from the DNN model (indicated by “crosses”) tracked the targeted values (indicated by “circles”) with relatively small errors. Even though the

Table 1
Prediction results of RV water level using DNN model and its optimized network structure.

Break size	LOCA location	Test data		Learning data		Number of hidden layers (number of hidden nodes)
		RMS error (m)	Maximum error (m)	RMS error (m)	Maximum error (m)	
Small	Hot-leg LOCA	0.11	0.44	0.08	0.67	6 (12-20-9-16-18-17)
	Cold-leg LOCA	0.09	0.37	0.06	0.83	6 (18-16-15-10-8-19)
	SGTR	0.03	0.10	0.03	0.37	5 (20-18-12-19-19)
Large	Hot-leg LOCA	0.04	0.12	0.03	0.40	9 (18-10-7-12-17-18-13-12-16)
	Cold-leg LOCA	0.13	0.78	0.07	1.13	3 (20-16-7)
	SGTR	0.07	0.33	0.05	0.38	4(19-14-20-12)

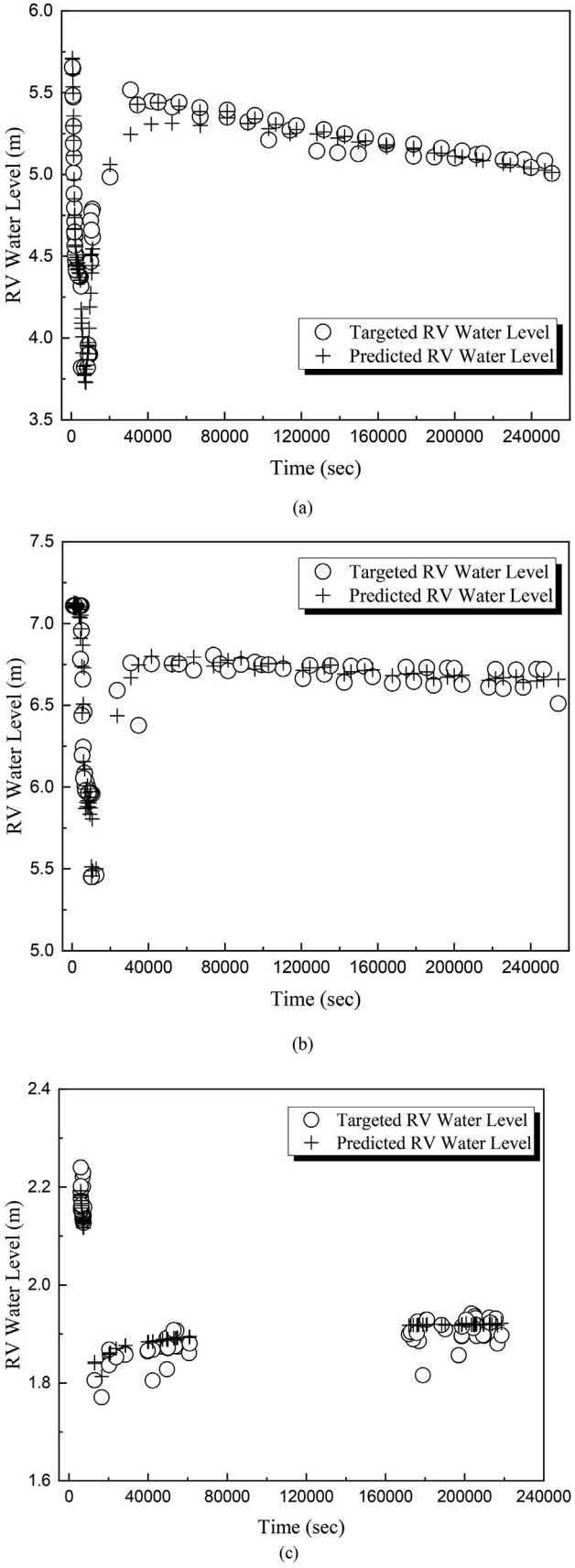

Fig. 9. Prediction results of DNN model for test data under hot- and cold-leg LOCAs and SGTR with small break size. (a) Hot-leg LOCA. (b) Cold-leg LOCA. (c) SGTR.

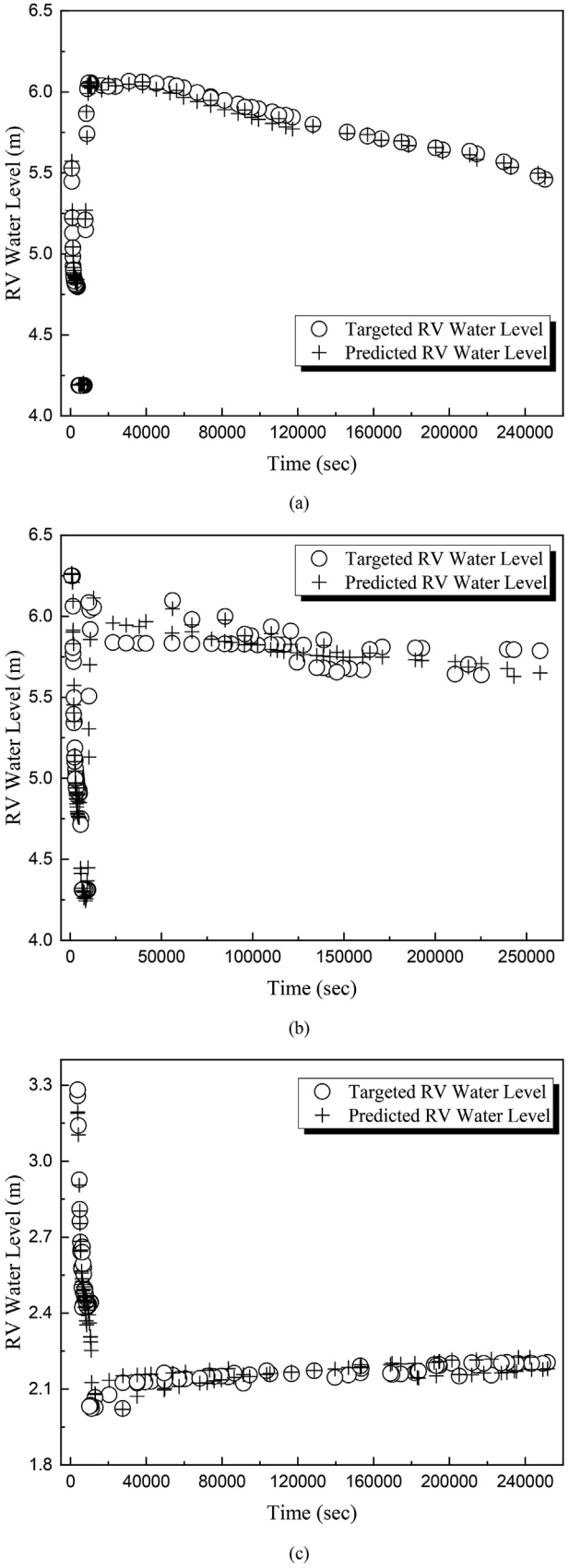

Fig. 10. Prediction results of DNN model for test data under hot- and cold-leg LOCAs and SGTR with large break size. (a) Hot-leg LOCA. (b) Cold-leg LOCA. (c) SGTR.

Table 2
Performance comparison for RV water level prediction using DNN and CFNN models.

Break size	LOCA location	Error for test data			
		DNN model		CFNN model	
		RMS error (m)	Maximum error (m)	RMS error (m)	Maximum error (m)
Small	Hot-leg LOCA	0.11	0.44	0.32	1.76
	Cold-leg LOCA	0.09	0.37	0.20	0.79
	SGTR	0.03	0.10	0.22	0.82
Large	Hot-leg LOCA	0.04	0.12	0.07	0.64
	Cold-leg LOCA	0.13	0.78	0.15	0.60
	SGTR	0.07	0.33	0.50	3.24

DNN model showed some undesirable prediction results arising from the significant change in RV water level over a short time interval in all the LOCA cases, no excessively over- or under-predicted values were present overall. Fig. 10(a), 10(b), and 10(c) indicate the predicted RV water levels compared with the targeted values according to the elapsed time after the reactor trip for the LOCAs and SGTRs with large break sizes.

The RV water level prediction error were analyzed for the test data according to the accident locations and sizes. Since the errors are distributed within approximately 25 cm with a 95% confidence level, it is confirmed that the DNN model provide quite accurate estimate of the RV water level. The DNN model has good accuracy especially in the hot-leg LOCA and SGTR cases.

4.2. Comparison of prediction performance with a machine learning method

The performances of the proposed DNN model and the CFNN model [6] for the RV water level prediction were compared in this paper. As listed in Table 2, the DNN model had much smaller RMSEs and maximum errors than the CFNN model in all the accident cases and especially in the large SGTR case. It is therefore concluded that the DNN model is an enhanced method to more accurately predict the RV water level than the CFNN model. The reason why the DNN model is superior is that its network structures optimized by the GA were able to be more effectively trained even using the same data. That is, the DNN model with the optimized hidden networks was able to infer a more proper regression function for RV water level prediction than the CFNN model of which the design parameters determining a shape of a membership function were optimized by the GA.

5. Conclusions

In this study, the RV water level was predicted by applying only a few signals to the DNN model to provide supporting information under severe accident circumstances as the required safety-critical instrumentation signals cannot be ensured on account of instrument degradation or failure. As a result of the RV water level prediction, it is concluded that the developed model is accurate as shown by its small RMSEs and maximum errors for the test data. Moreover, as the errors of the proposed model were much smaller than those of the CFNN model, it can be regarded that the DNN model performs better for the RV water level prediction. These results were obtained by utilizing the GA for automatically selecting the numbers of hidden layers and nodes. Although the DNN model has different optimized network structures according to the LOCA break location and size of the postulated accidents, the developed model was found capable of predicting the RV water level as a safety-critical signal.

Consequently, the applicability of the DNN model as a method to provide supporting information for checking the RV water level and utilizing the severe accident management guideline for mitigation of severe accidents was confirmed in this study. Furthermore, if additional studies to predict other accident monitoring variables are carried out in future, the supplementary applicability of the deep learning method for a comprehensive accident management support system can be verified.

Conflict of interest

The authors have no conflicts of interest to declare.

Acknowledgments

This work was supported by the National Research Foundation of Korea grant funded by the Korean government (MSIP), (Grant No. NRF-2018M2A8A4025978) and the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant, funded by the Korean Government (MOTIE) (Grant No. 20184030202180).

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep Learning, Nature 521 (2015) 436–444.
[2] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Massachusetts, 1989.
[3] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Massachusetts, 1996.
[4] D. Rumelhart, G. Hinton, R. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.
[5] R. Henry, et al., MAAP4: Modular Accident Analysis Program for the LWR Power Plants, User's Manual, Fauske and Associates Inc., 1994-2005.
[6] D.Y. Kim, K.H. Yoo, G.P. Choi, J.H. Back, M.G. Na, Reactor vessel water level estimation during severe accidents using cascaded fuzzy neural networks, Nucl. Eng. Tech. 48 (3) (2016) 702–710.
[7] M.G. Na, On-line estimation of DNB protection limit via a fuzzy neural network, Nucl. Eng. Tech. 30 (3) (1998) 222–234.
[8] M.G. Na, Y.J. Lee, I.J. Hwang, A smart software sensor for feedwater flow measurement monitoring, IEEE Trans. Nucl. Sci. 52 (6) (2005) 3026–3034.
[9] D. Lewy, ANN Activation Function - Comparison, PyData Warsaw, Warsaw, Poland, 2017. October 18-20, 2017.
[10] A. Ng, K. Katanforoosh, Y.B. Mourri, Neural Networks and Deep Learning, Coursera, 2017.
[11] M.G. Na, W.S. Park, D.H. Lim, Detection and diagnostics of loss of coolant accidents using support vector machines, IEEE Trans. Nucl. Sci. 55 (1) (2008) 628–636.
[12] S.H. Lee, Y.G. No, M.G. Na, K.I. Ahn, S.Y. Park, Diagnostics of loss of coolant accidents using SVC and GMDH models, IEEE Trans. Nucl. Sci. 58 (1) (2011) 267–276.
[13] K.H. Yoo, Y.D. Koo, M.G. Na, Identification of LOCA and estimation of its break size by multiconnected support vector machines, IEEE Trans. Nucl. Sci. 64 (10) (2017) 2610–2617.