Transactions of the Korean Nuclear Society Autumn Meeting
Gyeongju, Korea, October 26-27, 2017

Deep Learning for NPP Event Classification

Ju Hyun Back and Man Gyun Na*
Department of Nuclear Engineering Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju, Korea
**Corresponding author: magyna@chosun.ac.kr*

1. Introduction

If an event or accident occur in nuclear power plants, plant operators will try to find out abnormal plant states by monitoring the temporal trends of several important parameters. However, operators are provided with a part of information and also, there may be not enough time to recognize and diagnose about corresponding situation. So, it is very difficult for operators to predict the progression of the events by observing the trends of some parameters on large display panels in the main control room. In addition, during a series of accident progression, the operators will face hundreds of instrument readings that show some typical patterns of that accident [1].

In case of the accidents that happen in a nuclear power plant (NPP), it is very important for the operator to identify its accidents in early time because these can raise the problems such as unexpected reactor trip, design basis accident (DBA), etc. Therefore, in order to effectively manage the accidents, trends of major parameters shortly after the accidents have to be observed and NPP accidents have to accurately be identified to provide its information for operators and technicians.

In this regard, the objective of this study is to identify the accidents when the accidents happen in a NPP. In this study, we applied the deep learning network (DNN) model to classify the initiating events of critical accidents such as loss of coolant accidents (LOCA), total loss of feedwater (TLOFW), station blackout (SBO), steam generator tube rupture (SGTR), main steam line break (MSLB), and feedwater line break (FWLB). Input variables to the DNN are the initial integral values of the signal measured in the reactor coolant system (RCS), steam generator, and containment vessel after reactor trip. The proposed DNN model is verified by using the simulation data of the modular accident analysis program (MAAP4) code [2].

2. Methods and Results

2.1 Deep Learning

Deep learning is a machine learning technique using a deep neural network (DNN). Fig. 1 shows the concept of deep learning. Modeling is done using training data, and a learning rule is used as an algorithm for model building. Through this process, a DNN model is finally created.

2.2 Deep Neural Network (DNN)

Fig. 2 shows the DNN model. The DNN model consists of input layer, many hidden layers, and output layer. Input signals enter the input layer, pass through the hidden layers, and exit to the output layer. The signals are multiplied by corresponding weights and delivered to each hidden layer node. The node of hidden layers is obtained by the sum of the weights. Then, the node outputs the value which is calculated by inputting the sum of the weights to the activation function. The output layer receives the signal from the last hidden layer, and then outputs the final result.

2.3 Improvement of the DNN

The reason that the neural network with deeper layers yielded poorer performance was that the network was not properly trained. There are three difficulties in training process of the deep neural network with the back-propagation algorithm. The three difficulties are vanishing gradient, overfitting, and computational load [3].

2.3.1 Vanishing Gradient.

The vanishing gradient in the training process with the back-propagation algorithm occurs when the output error is more likely to fail to reach the farther nodes. The back-propagation algorithm trains the neural network by propagating the output error backward to the hidden layers. However, since the error hardly reaches the front hidden layer, the weight cannot be adjusted. Therefore, the hidden layers close to the input layer are not well properly trained [3]. Fig. 3 shows the vanishing gradient.

A solution to the vanishing gradient is the use of the rectified linear unit (ReLU) function as the activation function. The error is better transmitted than the existing sigmoid function. Eq. (1) is the definition of the ReLU function.

$$\varphi(x)=\begin{cases} x, & x>0 \\ 0, & x\le 0 \end{cases} \tag{1}$$

It produces zero for negative inputs and conveys the input for positive inputs. Fig. 4 shows the ReLU function. In the case of the sigmoid function, the output of the neural network node is not more than 1 even if the input value is large.

To implement the back-propagation algorithm, we need a derivative of the ReLU function. The derivative of the ReLU function is given by Eq. (2) according to the definition of the ReLU function [3].

$$\varphi'(x) = \begin{cases} 1, & x > 0 \\ 0, & x \le 0 \end{cases} \quad (2)$$

2.3.2 Overfitting.

If the number of hidden layers in the DNN model increases, the model becomes more complicated due to the increase of weights number. Therefore, the DNN model becomes vulnerable to the overfitting problem.

Solution to the overfitting is a method of dropout, which trains only some of the randomly selected nodes rather than the entire network. Fig. 5 shows the concept of the dropout. Some nodes are randomly selected at a constant rate and their outputs are set to be zero to deactivate the nodes [3]. The dropout effectively prevents overfitting by continuously changing nodes and weights in the training process. In addition, the use of huge amounts of training data is also very helpful in preventing overfitting. This is because the deep neural network depends less on a specific data.

2.3.3 Computational Load.

The number of weights increases exponentially with the number of hidden layers, thus requiring more training data. This ultimately needs more calculations to be done. Also, it takes longer to learn. This problem has been relieved to a considerable extent by using high-performance hardware such as GPU.

2.4 Application to NPP event classification

DNN model is used for classifying the data of the non-linear form. Input variables of the DNN model are composed of the signals measured at RCS, steam generator, and containment vessel. After reactor trip, major accidents is classified by using very short time integral values of the measured signals [1].

The data was obtained using MAAP4 code. Input variables of DNN model are integral values of 13 simulated sensor signals. The total simulation number of accident scenarios is 620. The acquired data are divided into training data and test data. The training data consist of 190 hot-leg LOCAs, 190 cold-leg LOCAs, 190 SGTR, 2 SBO, 2 MSLB, and 2 TLOFW. The test data consist of 10 hot-leg LOCAs, 10 cold-leg LOCAs, 10 SGTR, 1 SBO, 1 MSLB, and 1 TLOFW.

In this paper, DNN model was used to classify seven types of events in NPPs. NPP events are classified by the trained DNN model as shown in Fig. 6. As a result, perfect classification of events in case of no measurement errors is shown in Table I. That is, perfect classification was accomplished even though pretty short time measurement values were used.

Since the aforementioned results were generated from simulated data, it was assumed that there were no measurement errors in the input signals. Now, six types of measurement errors are assumed to check the effect of the measurement error on the proposed algorithm: minus 3% bias error, plus 3% bias error, minus 5% bias error, plus 5% bias error, random errors less than 3% and random error less than 5%. Table II shows the result under the assumption of measurement errors. Despite of measurement error, the DNN model classifies NPP events accurately more than 99%.

Table III shows the result of the case when the safety system works. Each of the safety systems was operated with delay.

Table IV and V show the comparison between DNN model and support vector classification (SVC) model [1]. If there is no measurement error, both performances are perfect. On the other hand, if there is a measurement error, the performance of DNN is slightly lowered. In case of DNN. If DNN is used with small amount of data, its performance is not good. However, if a large amount of data are acquired and the number of events increases, DNN performance will be improved.

Table VI and VII show the result according to the change of the dropout rate. In case of no measurement errors, it is shown that the performance is better when dropout rate is small.

Table I: Transient classification

Performance result	Integrating time (sec)	No. of Misclassification	
		Training data	Test data
DNN	3	0	0
	5	0	0
	10	0	0

Table II: Classification results using the DNN model with measurement errors

Performance result	Integrating time (sec)	Total data = 620					
		-3% bias error	3% bias error	-5% bias error	5% bias error	Random (below3%)	Random (below5%)
No. of Misclassification	3	4	7	3	8	6	6
	5	2	2	3	4	2	3
	10	3	3	4	7	4	4

Table III: Transient classification (safety system actuation)

Performance result	Total data = 620	
	Integrating time (sec)	No. of Misclassification
DNN	3	4
	5	2
	10	3

Table IV: Comparison (without measurement error)

Performance result	Integrating time (sec)	No. of Misclassification
DNN	3	0
	5	0
	10	0
SVC	3	0
	5	0
	10	0

Table V: Comparison (with measurement error)

Performance result	Integrating time (sec)	-3% bias error	3% bias error	-5% bias error	5% bias error	Random (below3%)	Random (below5%)	
DNN	3	4	7	3	8	6	6	
	5	2	2	3	5	2	3	
	10	3	3	4	7	4	4	
SVC	3	0	1	1	2	1	1	
	5	1	1	1	2	0	1	
	10	2	4	4	7	0	0	

Table VI: The result from change of drop rate (without measurement error)

Performance result	Drop rate (%)	No. of Misclassification	
		Training data	Test data
DNN	0	0	0
	3	0	0
	5	0	0
	10	1	0
	20	8	1

Table VII: The result from change of drop rate (with measurement error)

Performance result	Total data = 620						
	Drop rate (%)	-3% bias error	3% bias error	-5% bias error	5% bias error	Random (below3%)	Random (below5%)
DNN	0	4	7	3	8	6	6
	3	4	7	3	8	6	6
	5	4	7	3	8	6	6
	10	4	7	3	8	6	6
	20	4	7	3	8	6	6

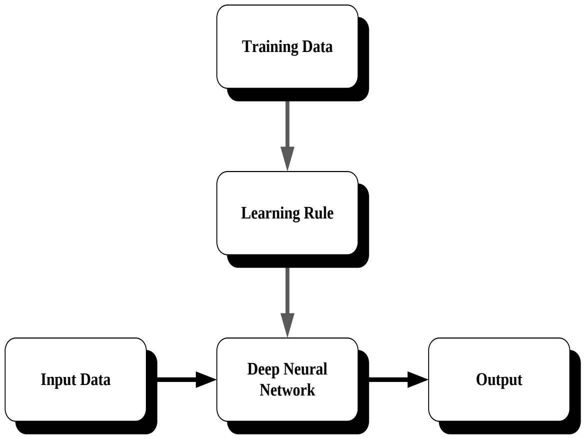

Fig. 1. The concept of deep learning

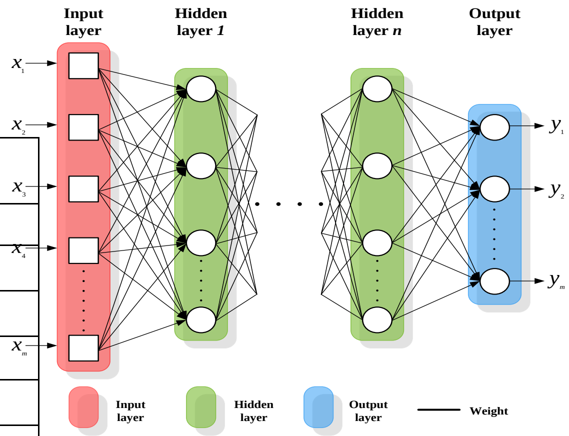

Fig. 2. DNN model

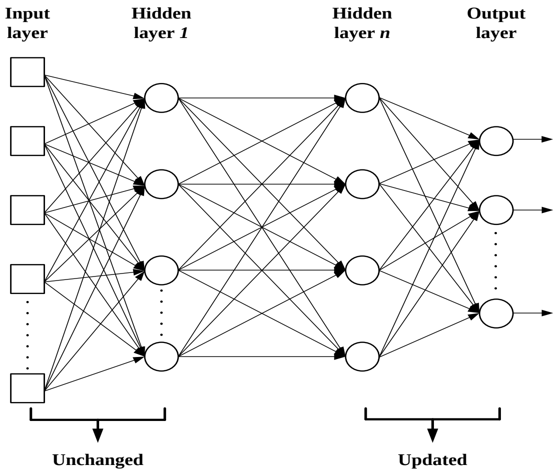

Fig. 3. Vanishing gradient

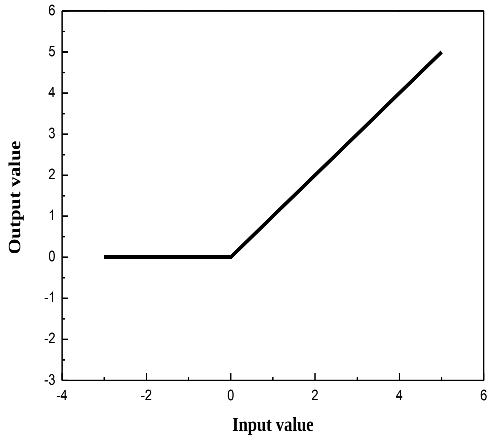

Fig. 4. Graph of ReLU function

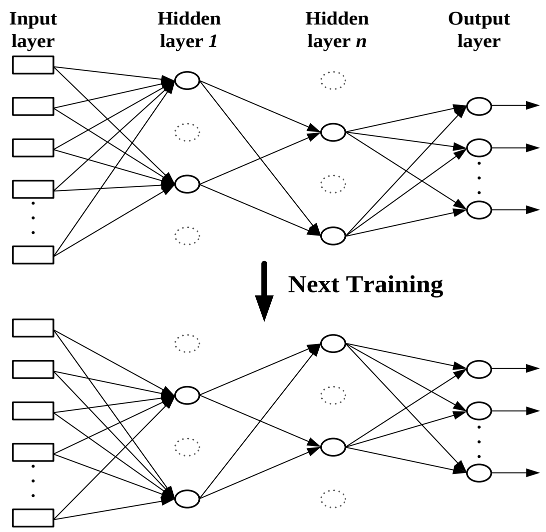

Fig. 5. The concept of dropout

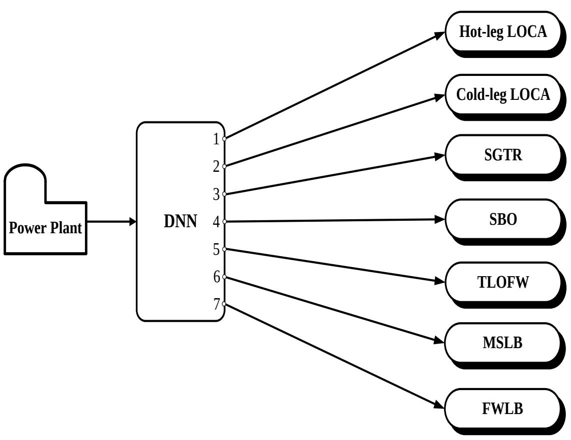

Fig. 6. Event classification using the DNN model

3. Conclusions

In this study, the proposed DNN model is verified by using the simulation data of MAAP4 code. We used an initial integral value of the simulated sensor signals to identify the NPP accidents. The training data was used to train the DNN model. And, the trained model was confirmed using the test data. As a result, it was known that it can accurately classify seven events. Since the proposed algorithm uses initial data after reactor trip and the initial simulation data was known to be accurate, it can be effectively used in an actual NPPs as well. By providing accurate information for accidents in a NPP, it will be helpful for the operators to rapidly respond to the accident situations. Also, if more data will be gained in the future, it is expected that better performance will be shown by using deep learning as the core technology of the fourth industrial revolution.

REFERENCES

[1] J. H. Back, K. H. Yoo, and M. G. Na, Identification of NPP accidents using support vector classification, *proc. Of KNS Autumn Mtg*, Gyeongju, Korea, Oct. 27-28, 2016.
[2] MAAP4-Modular Accident Analysis Program for LWR Power Plants, User's Manual, prepared by Fauske and Associates, LLC for EPRI, Project RP3131-02, May 1994-june 2005.
[3] S. P. Kim, MATLAB Deep Learning, Apress, pp. 102-119, 2017.