Prediction of Pressurized Water Reactor Core Design Parameters Using Artificial Neural Network for Loading Pattern Optimization

Hyunbin Jang, Hyun Chul Lee*
School of Mechanical Engineering, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan, 46241, Korea
**Corresponding author: hyunchul.lee@pusan.ac.kr*

1. Introduction

The loading pattern optimization for reactor core is to find the most economical loading pattern of all loading patterns, satisfying the safety restriction requirements. If the reactor core design parameters for all loading patterns could be computed with numerical analysis codes, the optimal loading pattern could be obtained. However, it is limited because of the huge computation time. Therefore, various methods such as Simulated Annealing (SA) algorithm [1] have been studied to effectively perform calculations for finding the optimal loading pattern. If these design parameters such as the peaking factor and the cycle length could be calculated faster than the numerical analysis codes, the optimal loading pattern could be found faster.

In this study, reactor core design parameter prediction algorithms using an Artificial Neural Network (ANN) has been developed to replace the numerical analysis codes in the loading pattern optimization. A deep learning algorithm was used to improve the accuracy of predictions because it can solve more complex and nonlinear problems using multiple hidden layers for feature extraction and transformation. In addition, a system has been developed that automatically generates training data for predicting the reactor core design parameters using data from the Westinghouse 2-loop plant. In this paper, the peaking factor and the cycle length according to the loading patterns were predicted using the deep learning algorithms and automatically generated training data.

2. Automatic Generation of Training Data

The number of training data in the ANN algorithms greatly contributes to learning accuracy. Therefore, it is very important to obtain enough training data. In this study, the system for automatically generating training data was established, training data were produced using the STREAM/RAST-K 2.0 [2-4] code system. The STEAM code is a neutron transport analysis code for Light Water Reactor (LWR) core calculation and developed at Ulsan National Institute of Science and Technology (UNIST) and the RAST-K 2.0 code is a diffusion nodal code for Pressurized Water Reactor (PWR) core analysis developed by UNIST.

Based on the data of the Westinghouse 2-loop plant reactor core, the assembly calculation used for RAST-K input has been performed using the STREAM code, and the RAST-K core calculation has been performed using the assembly calculation result. The RAST-K 2.0 input files with random loading patterns are automatically generated. The training input parameters, the peaking factors and the cycle lengths are automatically extracted from the RAST-K 2.0 output file and updated with big data in the Comma-Separated Values (CSV) file format. The peaking factor is extracted from the maximum peaking factor in the cycle, and the cycle length is extracted when the Critical Boron Concentration (CBC) is 10 ppm. Fig. 1 shows an overall procedure of automatic generation of training data.

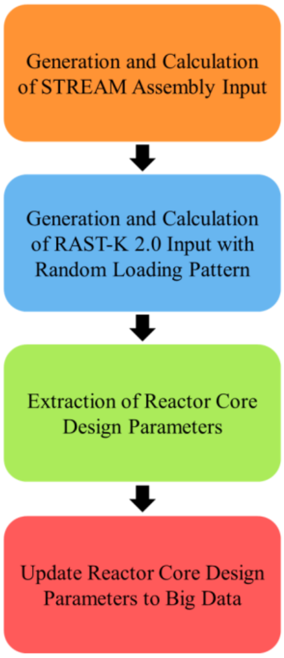

Fig. 1. Overall procedure of automatic generation of training data

3. Selection of Deep Learning Algorithm

Deep learning algorithms include various algorithms such as Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN). The RNN algorithm is suitable for learning time-sequenced data such as natural language processing. The CNN algorithm is suitable for image classification and image processing. The CNN algorithm recognizes features while preserving spatial information of the data. Therefore, The CNN algorithm was adopted for learning based on spatial information. In this study, the training result of the CNN algorithm is compared with the training result of the DNN algorithm which is the basic algorithm of deep learning.

Generally, the CNN algorithm consists of input layer, convolution layer, pooling layer, fully connected layer, and output layer. The convolution layer is a layer for extracting meaningful features, and the pooling layer is a layer for subsampling to reduce features. Fig. 2 shows a structure of a conventional convolutional neural network.

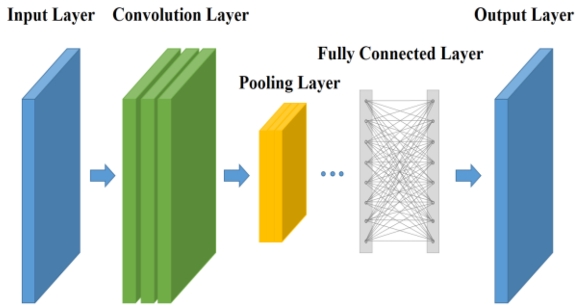

Fig. 2. Structure of the convolutional neural network

4. Results and Discussion

In this section, the training results of the DNN and CNN algorithms for the peaking factor and the cycle length are compared. The prediction algorithms were implemented using the TensorFlow library [5]. The DNN algorithm has 10 hidden layers, and the CNN algorithm has 6 convolution layers, 2 pooling layers and 3 fully connected layers. In the internal structure of the CNN algorithm, 3x3 convolution filters were used for the convolution layers, and 2x2 pooling filters and a max pooling method, extracting the largest value in the filter region, were used for the pooling layer. In addition, approximately 42,000 training data were used for training, and 6,000 test data were used to confirm training results. The number of total parameters is about 150,000 in both the DNN and CNN algorithms. Fig. 3 shows internal structures of the prediction algorithms

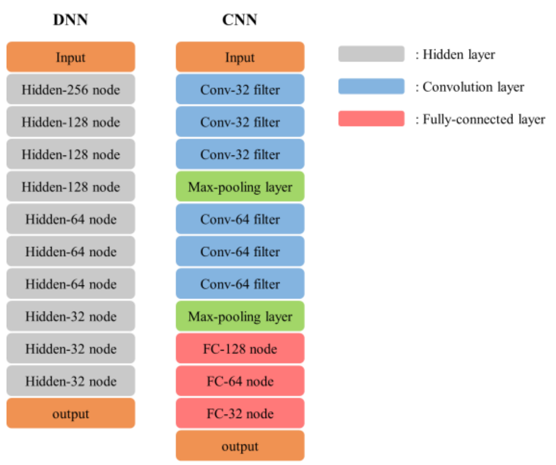

Fig. 3. Internal structures of the prediction algorithms

4.1 Training Result of Peaking Factor

As described in Section 2, big data generated automatically using the STREAM/RAST-K 2.0 code system are used as training data for peaking factor prediction. The results of the peaking factor prediction are shown in Fig. 4, Fig. 5 and Fig. 6. Fig. 4 shows the peaking factor training procedure as a Root Mean Square (RMS) error according to the training step. Fig. 5 and Fig. 6 show the training result of the peaking factor of the DNN and CNN algorithms. In Fig. 5 and Fig. 6, the x-axis is the target value calculated by the STREAM/RAST-K 2.0 code system, and the y-axis is the peaking factor predicted by the deep learning algorithms.

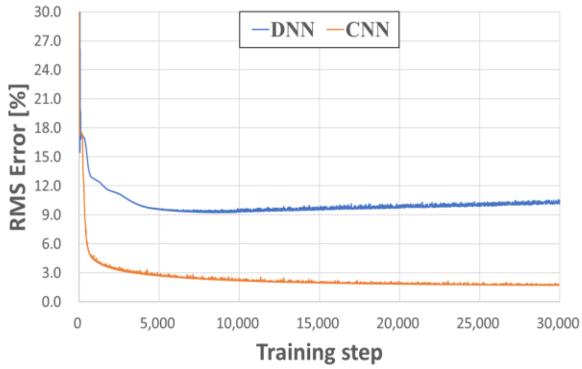

Fig. 4. Convergence process for peaking factor prediction

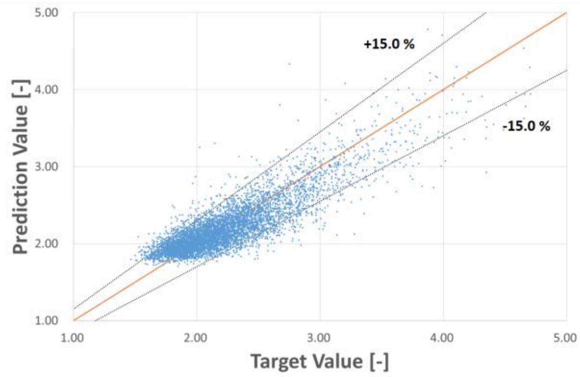

Fig. 5. Training result of peaking factor using DNN algorithm

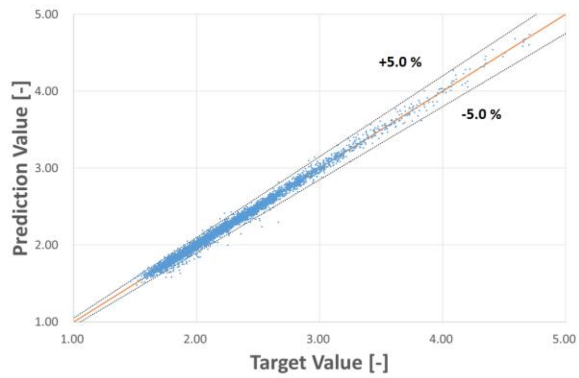

Fig. 6. Training result of peaking factor using CNN algorithm

The prediction results of the model trained by the DNN algorithm are 9.28% for RMS error and 60.91% for maximum error. However, 90.2% of the test data are obtained with the prediction error of less than 15%.The results of the CNN algorithm are 1.74% for RMS error, 13.81% for maximum error and 98.7% of the test data are obtained with the prediction error of less than 5%. The results show that the CNN algorithm performs much better than the DNN algorithm in the peaking factor prediction. Table I. summarizes the peaking factor prediction errors of the DNN and CNN.

Table I: Peaking factor prediction error of DNN and CNN

Algorithm	Prediction error	
	RMS (%)	Max (%)
DNN	9.28	60.91
CNN	1.74	13.81

4.2 Training Result of Cycle Length

The results of the cycle length prediction are shown in Fig. 7, Fig. 8 and Fig. 9. Fig. 7 shows the cycle length training procedure as RMS error according to the training step. Fig. 8 and Fig. 9 show the training results of the cycle length of the DNN and CNN algorithms. In Fig. 8 and Fig. 9, the x-axis is the target value, and the y-axis is the cycle length predicted by the deep learning algorithms.

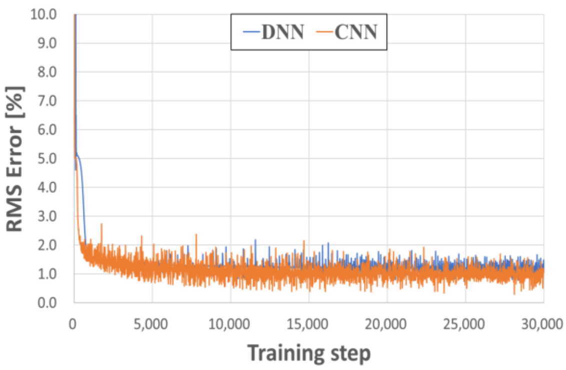

Fig. 7. Convergence process for cycle length prediction

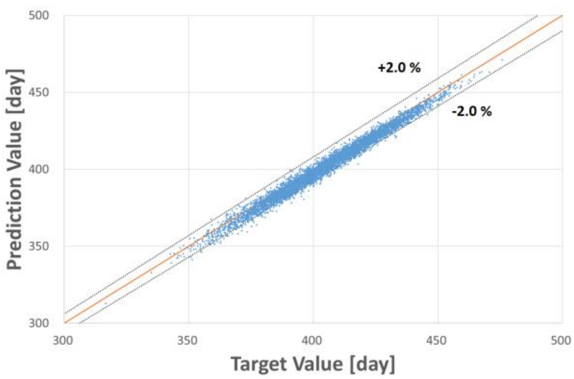

Fig. 8. Training result of cycle length using DNN algorithm

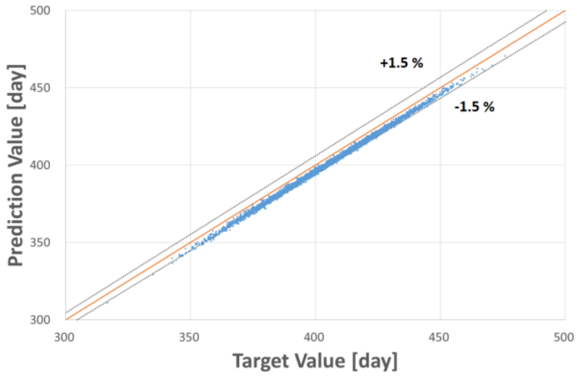

Fig. 9. Training result of cycle length using CNN algorithm

The prediction results of the DNN algorithm are 1.06% for RMS error, 3.98% for maximum error and 96.3% of the test data are obtained with the prediction error of less than 2%. The results of the CNN algorithm are 1.07% for RMS error, 2.12% for maximum error and 96.9% of the test data are obtained with the prediction error of less than 1.5%. The results show that the CNN algorithm has better cycle length prediction performance than the DNN algorithm. Table II. summarizes the cycle length prediction errors of the DNN and CNN.

Table II: Cycle length prediction error of DNN and CNN

Algorithm	Prediction error	
	RMS (%)	Max (%)
DNN	1.06	3.98
CNN	1.07	2.12

5. Conclusions

In this study, the peaking factor and the cycle length prediction algorithms using the deep learning algorithms, which can replace numerical analysis codes, has been developed. Prior to developing the prediction model, the system for automatically generating training data used in deep learning algorithms has been developed. The training data have been generated automatically using the STREAM/RAST-K 2.0 code system and the extracted the peaking factor and the cycle length have been updated with big data.

Based on the deep learning algorithms, the peaking factor and the cycle length prediction were performed with about 42,000 training data, and the prediction results were confirmed with 6,000 test data. The CNN algorithm shows higher performance than the DNN algorithm in both the peaking factor and the cycle length prediction. Especially, the CNN algorithm shows significantly higher performance than the DNN algorithm in the peaking factor prediction. However, even if the prediction algorithms developed using the CNN algorithm show better performance than the algorithm using DNN, the prediction performance is still insufficient.

Therefore, the future work of this study will focus on modifying the training data structures and the internal structure of the prediction algorithms to improve the performance of the peaking factor and the cycle length prediction algorithms. Furthermore, the system for automatically generating the big data and the prediction algorithms for the reactor core design parameters other than the peaking factor and the cycle length will be developed.

REFERENCES

[1] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by Simulated Annealing, Science, Vol.220, No.4598, pp.671-680, 1983.

[2] S. Choi, C. Lee, D. Lee, Resonance Treatment using Pin-Based Pointwise Energy Slowing-Down Method, Journal of Computational Physics, Vol.330, pp.134-155, 2017.

[3] J. Choe, S. Choi, P. Zhang, J. Park, W. Kim, H. Shin, H. Lee, J. Jung, D. Lee, Verification and validation of STREAM/RAST-K for PWR analysis, Nuclear Engineering and Technology, Vol.51, No.2, pp.356-368, 2019.

[4] F. Khoshahval, M. Park, H. Shin, P. Zhang, D. Lee, Vanadium, rhodium, silver and cobalt self-powered neutron detector calculations by RAST-K v2.0, Annals of Nuclear Energy, Vol.111, pp.644-659, 2018.

[5] M. Abadi, et al., TensorFlow: A System for Large-scale Machine Learning, Proceeding of the 12th USENIX conference on Operating Systems Desing and Implementation, pp.265-283, Nov 2016.