Impact of Training Dataset Reduction on the Prediction Accuracy of Nuclear Design Parameters using Convolutional Neural Network

Siarhei Dzianisau, Muhammad Farid Khandaq, Deokjung Lee*
Department of Nuclear Engineering, Ulsan National Institute of Science and Technology, 50 UNIST-gil, Ulsan, 44919
**Corresponding author: deokjung@unist.ac.kr*

1. Introduction

These days, Artificial Neural Networks (ANN) gained significant popularity among researchers in various fields of knowledge. The first publicly recognized success of ANNs happened in the area of image recognition and computer vision. The ability of neural networks to learn specific features of different objects through training on thousands of images immediately became desired in various areas of research.

Once applied in other industries, it was quickly discovered that not every field of knowledge has readily available thousands of samples that can be used for training an ANN model. In fact, in most cases researchers needed to create training data for their models from scratch, often spending hundreds of hours and noticeable material resources for that purpose. As a result, we see a huge difference in how various people prepare their data. Thus, in our first work [1], we used 500 modelled Loading Pattern (LP) samples to predict several nuclear design parameters. In another work [2], we used 108,000 training samples to predict homogenized 2-group macroscopic cross-sections (XS) for the purpose of being used in a nodal diffusion code. Some of the latest publications include works of Lee et al [3], who generated 82,500 data samples to train a Convolutional Neural Network (CNN) for diagnosing abnormal accidents during a Nuclear Power Plant (NPP) operation. Or, even more that that, Guo et al [4] prepared 18,816 photographs of Fuel Assembly (FA) cladding scratches and other damages that were obtained using an actual experimental equipment and materials.

Though each of described examples has different application of an ANN and different data preparation technique, there is one common thing for all of them. In each of those studies, the training dataset size was chosen based on either the current capability of the research team and equipment or on the researcher's intuition, which led to an assumption that the given data size is enough for producing an unbiased result. However, such approach does not provide evidence on whether the chosen dataset size is not too large or too small for a certain application. Therefore, the outcomes of given studies need to be further tuned in order to be used by commercial companies or industry, in which every usage of computation equipment or material resources is desired to be minimized.

In this study, we are investigating whether the training dataset size for producing a general-case prediction using ANN can be significantly lowered, by that saving decent amounts of time and resources. For this purpose, we prepared our training data using techniques discussed in Section 2. After that, we used the prepared data for training a CNN that is described in Section 3. The factual outcomes of our study and the analysis of the results can be found in Section 4.

2. Data Preparation

In this work, we chose a CNN regression model that predicts multiple nuclear design parameters as our testing tool. In order to evaluate the importance of having a larger training data size, a large dataset consisting of 40,192 training samples and 5,120 validation samples was generated using our in-house nodal diffusion code RAST-K [5]. In addition to that, a testing dataset consisting of 1,280 samples and based on different from the training dataset types of fuel was generated. This separation assures that our training data does not overlap with the testing data, by that providing a general case application scenario, which is primarily desired by industry. All used fuel types for training and testing datasets are listed in Table I. The reactor type chosen for this study is a Small Modular Reactor (SMR) described in [6].

Table I: Fuel Assembly Types used For Training and Testing Datasets

FA Type	Enrichment, %	Burnable Absorber
Training Dataset		
16 × 16 Type A	1.5	No
16 × 16 Type B	2.5	No
16 × 16 Type C	3.5	No
16 × 16 Type D	4.5	No
16 × 16 Type E	5.5	No
Testing Dataset		
16 × 16 Type F	3.09	No
16 × 16 Type G	4.13	No
16 × 16 Type H	4.95	No

To avoid uncertainties that might get introduced in case of using a free-range mini-batch size, we set the total size of the training dataset and the validation dataset, as well as all used fraction sizes of studied datasets as multiples of 256. A total of 166 target reactor design parameters including Cycle Length, Critical Boron Concentration, values of various pin peaking factors, normalized fuel assembly power and burnup distributions were intended to be predicted using an ANN. A few examples of the target parameter distributions in the studied datasets of different sizes are shown in Fig. 1. The color codes for the given examples are the following: orange for 256 samples, green for 512 samples, pink for 768 samples, and blue for 40,192 samples.

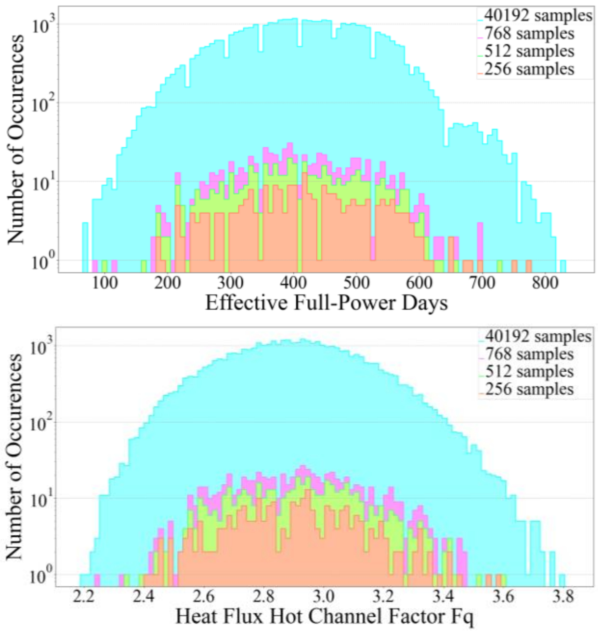

Fig. 1. Distribution of target parameters for different dataset sizes.

The prepared training data was further processed and used for a CNN training as discussed in the following section.

3. Neural Network Description and Methodology

In order to reduce uncertainties that are not directly related to the neural network training, we applied multiple mitigation techniques. The data preparation approach was described in the previous section, while the particular methodology related to the CNN model training is explained in Subsections 3.1 and 3.2 below.

3.1. CNN Architecture

For this study, a CNN shown in Fig. 2 was used. All layers of this ANN except for the very last output layer were followed by a ReLU activation function. The output layer was followed by a "sigmoid" activation function in order to provide a smoother output shape of the data prediction.

As stated in Section 2 of this paper, the dataset size for all our evaluated cases was chosen as a multiple of 256. That was done to train all cases using a mini-batch of 256 samples. In this case, each mini-batch of any dataset size used in our study would be perfectly filled with training samples in order to get even weight adjustment during training for all cases.

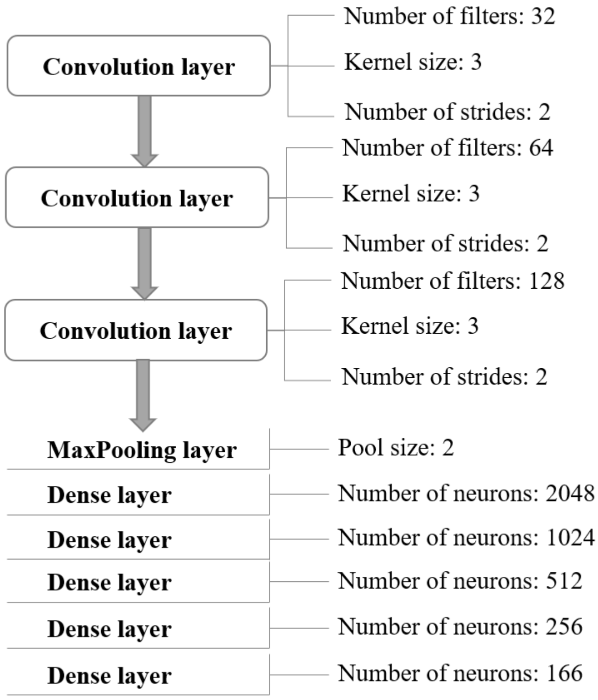

Fig. 2. CNN architecture used in this study.

For each training case, all hyperparameters were kept constant. In addition to before-mentioned mini-batch size, we used Adam optimizer with learning rate 0.0007 and default decay options. Mean Absolute Error (MAE) was chosen as the loss function for all cases, the validation dataset was fixed as described in Section 2, and the training data shuffle was performed at each epoch. The neural network was built and trained using Python 3.7 and TensorFlow 2.4 on nVidia RTX 2060 SUPER 8 Gb graphics card.

3.2. Testing Methodology

For proper measurement of the prediction accuracy for different dataset sizes, the case matrix shown in Table II was defined.

Table II: Training Dataset Size Arrangement for this Study

Training dataset size (weight adjustments per 1 epoch)
Initial Training Dataset Sizes
256 (1), 768 (3), 1280 (5), 1792 (7), 2304 (9), 2816 (11), 3328 (13), 3840 (15), 4352 (17), 4864 (19), 6400 (25), 9472 (37), 12544 (49), 15616 (61), 18688 (73), 21760 (85), 24832 (97), 27904 (109), 30976 (121), 34048 (133), 37120 (145), 40192 (157)
Additional Training Dataset Sizes
256 (1), 512 (2), 768 (3), 1024 (4), 1280 (5)

In order to account for the uncertainties that occur during training a model, we used an ensemble model training that consists of 10 trainings for each training dataset size given in Table II. After that, we applied the trained ensemble models to the testing data and found the average relative error for all 166 predicted parameters for each model in the ensemble. This value can be called a Combined Mean Relative Difference, and for brevity, we will call this value an Error Metrics (EM). Since our testing dataset was evaluated by 10 different models of the ensemble, we were able to calculate Mean and Standard Deviation of EM for each of the studied test cases using Pandas module of Python. Calculation of the testing dataset predictions was performed after each epoch of the training process for each model. By doing that, we intended to save the model configuration that shows the lowest possible EM for the testing data. For each particular case, we set the maximum number of training epochs to 5,000, and we set a rule that terminates the training if there is no improvement in the testing data EM over the latest 150 epochs. The final product of each training is the model with the lowest testing data EM.

4. Results and Discussion

Once all the initial data preparation and model training conditions were satisfied, we obtained the following results for our testing dataset. Below, we provide the Mean and Standard Deviation values for the chosen testing data as functions of the training dataset size and the estimated time required to generate that training dataset. In order to calculate the time, we used an averaged value of 19 seconds per one LP calculation using our code RAST-K and Intel Core i7 7700K at 4.37 GHz. The results of calculations are shown in Fig. 3 and in Table III.

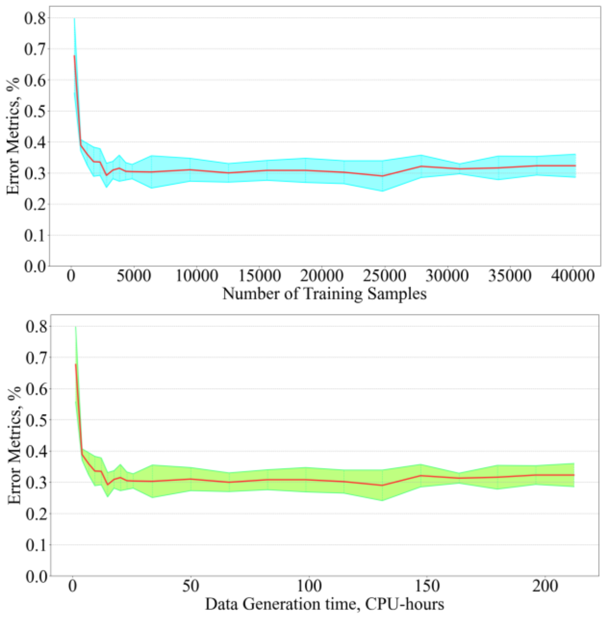

Fig. 3. Mean and Standard Deviation of the Testing Dataset Error Metrics as functions of the training dataset size and data generation time.

From the results shown above, it can be observed that there is a "saturation" point in the number of training data samples, exceeding which does not result in improvement of the EM for the testing data. Therefore, the time and resources spent on preparing that excess training data could have been saved. However, the value of Standard Deviation for the testing data obtained using 256-sample training dataset is noticeably higher compared to using other training dataset sizes. This can be related to the chosen termination rule that stops the code if the testing data EM does not improve over the latest 150 epochs. In particular, as shown in Table II, the weights are being updated only once per epoch for that smallest dataset case, which may seem to be unfair compared to other cases. To partially account for that, and to add more resolution to the lower-end dataset size area, we ran the same calculation again using the Additional Training dataset sizes from Table II. Contrary to our previous approach, we ran each of those cases for all 5,000 epochs without applying the termination rule. Our expectation was to see the reduction of Mean and Standard Deviation of the testing dataset for all additionally evaluated testing dataset cases. The result for the additional 10-times ensemble evaluation on the previously discussed testing data is given in Fig. 4 and in Table III.

Table III: Testing Dataset Prediction Results using Different Size of Training Dataset

Training Dataset Size, # samples	Mean, %	Standard Deviation
Initial Evaluation of the Testing Dataset Prediction		
256	0.677	0.120
768	0.389	0.018
1280	0.360	0.035
1792	0.336	0.047
2304	0.335	0.043
2816	0.292	0.039
3328	0.309	0.028
3840	0.315	0.042
4352	0.305	0.028
4864	0.304	0.023
6400	0.303	0.052
9472	0.310	0.037
12544	0.300	0.030
15616	0.308	0.032
18688	0.308	0.039
21760	0.302	0.037
24832	0.290	0.049
27904	0.321	0.036
30976	0.313	0.016
34048	0.316	0.038
37120	0.323	0.030
40192	0.323	0.037
Additional Evaluation of the Testing Dataset Prediction		
256	0.424	0.022
512	0.310	0.030
768	0.300	0.037
1024	0.288	0.041
1280	0.274	0.030

The produced result shows that our initial expectation is confirmed, and the values of Mean and Standard Deviation for the testing dataset using additionally trained models are reduced. At the same time, the graph shown in Fig. 4 demonstrates a clearer transition to the "saturation" point.

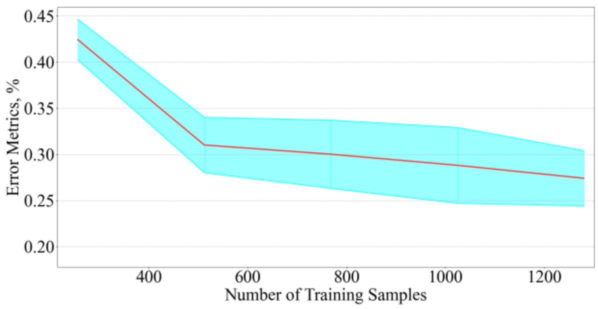

Fig. 4. Updated Mean and Standard Deviation of the Testing Dataset Error Metrics as functions of the training dataset size.

5. Conclusion

In this paper, we presented the results of our study on how the training dataset size impacts the accuracy of the trained ANN model. It was found that for a general case prediction there is a "saturation" dataset size, exceeding which does not result in the accuracy improvement. This can be partially explained by the choice of the testing dataset in our study. Unlike many other studies, we did not build the testing data from the same dataset as the training data. Instead, we created a new testing dataset that is using different fuel enrichments and therefore illustrates a more generalized use-case. The particular difference between training data and testing data used in our study is shown in Table I, and the detailed explanation of the testing dataset evaluation methodology is discussed in Subsection 3.2.

In our future studies, we would like to perform similar tests using a testing dataset within the same range as the training data. We expect that the "saturation" point for that specific case would be much higher compared to the testing data used in our study. Finally, though we attempted to eliminate as many uncertainties and obstacles as possible, there are still ways to improve and generalize our result. In particular, we would like to test more application scenarios (classification and regression), more reactor models, and to apply the no-termination rule used for the Additional Training cases to all training cases.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT).
(No. NRF-2019M2D2A1A03058371).

REFERENCES

[1] S. Dzianisau, H. Kim, C. Kong, D. Yun, D. Lee, Development of Barcode Model for Prediction of PWR Core Design Parameters Using Convolutional Neural Network, Transactions of the Korean Nuclear Society Autumn Meeting, Goyang, Korea, Oct 23-25, 2019.

[2] S. Dzianisau, J. Choe, A. Cherezov, D. Lee, Macroscopic Cross-Section Generation for Nodal Code RAST-K Using Artificial Neural Network, Transactions of the Korean Nuclear Society Autumn Meeting, Changwon, Korea, Dec 16-18, 2020.

[3] G. Lee, S. J. Lee, C. Lee, A Convolutional Neural Network Model for Abnormality Diagnosis in a Nuclear Power Plant, Applied Soft Computing, Vol. 99, 106874, 2021.

[4] Z. Guo, Z. Wu, S. Liu, X. Ma, C. Wang, D. Yan, F. Niu, Defect Detection of Nuclear Fuel Assembly Based on Deep Neural Network, Annals of Nuclear Energy, Vol. 137, 107078, 2020.

[5] 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.

[6] J. Jang, J. Choe, S. Choi, M. Lemaire, D. Lee, H. Shin, Conceptual Design of Long-cycle Boron-free Small Modular Pressurized Water Reactor with Control Rod Operation, International Journal of Energy Research, Vol. 44(8), pp. 6463-6482, 2020.