Optimization of Training Dataset Size for Predicting Homogenized Macroscopic Cross-Sections using Deep Neural Network

Siarhei Dzianisau, Korawit Saeju, 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

During the last decade, Artificial Neural Networks (ANN) became a solid instrument for solving complex and widespread problems such as image recognition or defect (anomaly) detection. At the same time, since ANNs are essentially Machine Learning (ML) algorithms, they are being successfully applied for numerical data prediction and analysis. For example, we used ANN-based surrogate models for accelerating our in-house reactor analysis codes as well as reactor simulation methodologies. Thus, we applied a Convolutional Neural Network (CNN) to the problem of reactor design parameters estimation [1]. Later, we found that using an ANN for whole-core parameters prediction was not as promising as building a surrogate ANN model for homogenized macroscopic cross-section (XS) generation [2]. That study led us to development of a brand-new nodal diffusion code system – RAST-AI. The establishment of the RAST-AI code system is briefly discussed in [3].

To train a neural network model that can predict XS data required by RAST-AI, we needed to significantly optimize our data preparation workflow. Since the idea of RAST-AI is to work with a free-range geometry, we would have to deal with way too many free-range parameters, such as Fuel Assembly (FA) layout, each Fuel Pin (FP) fuel configuration and enrichment, various operation parameters such as boron concentration (BOR), fuel temperature (TFU) change due to power change, moderator temperature (TMO), burnup, etc. Even the FP configuration alone can create a significant problem, since each 16x16 FA contains 236 FP, which can be generalized to 32 unique FP positions as a result of octant symmetry. This could theoretically lead to over $2 \cdot 10^{35}$ unique FA configurations in case of having all unique FP compositions.

Though it sounds like a tremendously complex problem, it is something that ANNs are known to successfully deal with. Thus, a typical image recognition problem consists of theoretically infinite number of each class images, which can only be reduced to finite (though, still extremely large) number if we restrict the pixel size of images to some reasonably small value. Therefore, if we generalize the training data as much as possible, we would be able to train a model that can detect underlying dependencies between the input and output and make a prediction with reasonable accuracy.

In our last publication [4], we attempted to generalize the data preparation for an ANN in terms of the required training dataset size. We found that for a general case prediction, the training dataset size can be greatly reduced. In this paper, we are discussing how this result and methodology was applied to the problem of macroscopic XS generation.

2. Methodology

As stated in the Introduction, the problem size that we have to deal with, is extremely large. Therefore, from the very beginning we decided to make the model as general as possible. In particular, we decided to use 5 fuel enrichment types (from 1.0 wt% to 5.0 wt%), and to distribute given enrichments in such a way that any number of training geometries in our dataset would have a uniform distribution of FP enrichments. This means that in each FP position within FA each enrichment will appear equal number of times. We expect that such a rule would be beneficial for an ANN. In addition, we decided to use 90 branch calculations in which the values of BOR, TMO and TFU are picked from the list shown in Table I.

Table I: Range of FP Enrichments and Branch Parameters used in Training and Testing Datasets

Enrichment, %	BOR, ppm	TMO, K	TFU, K
1.0	0.1	300	300
2.0	300	450	600
3.0	600	584*	850*
4.0	700*	600	900
5.0	900		1200
	1200		1500
	1500		

* Default case

The values marked as the default values in Table I are not part of the actual branch calculations and are calculated only once for the default case. Other values are interchanged in the abovementioned branch calculation, thus producing the total number of 90 per each geometry. The idea behind the given numbers is to cover as wide range of parameters as possible, while keeping a wide gap to save computation resources.

Once the training and testing data structure was defined, we used our in-house lattice physics code STREAM [5] for calculating homogenized XS and pin power shapes for an arbitrary number of geometries. The data produced by STREAM was used for training and testing the ANN model. In all studied cases, we separated training and testing data geometries so that there was no

overlap of geometry between that data. The final model was also evaluated using a completely independent data that is using random enrichments in range between 1.0 and 5.0 wt%. Unlike in training data, we did not fix the enrichment values to certain numbers in this additional testing dataset. Therefore, it consists of a continuous range of enrichments that do not have any pattern inside FA. Finally, the branch parameters of the additional testing dataset also contain non-fixed values of BOR, TMO and TFU, which should represent the ultimate general case testing problem.

3. Dataset optimization

3.1. Optimization of branch number

To start the dataset optimization, we decided to generate 1900 training geometries and 100 validation geometries. For the testing data, additional 100 geometries were generated using the same technique. We applied all the rules and methods established in Section 2. In particular, we separated geometries of training data and geometries of validation data, since our goal was to train a general case model. As a result, the total number of training samples is 171,000 (1900 geometries with 90 branches each).

Before we can determine the optimal number of data samples, we can find how many branch cases are needed for such kind of problem. As stated earlier, we arbitrarily chose the number of branch calculations as 90, which is in fact lower than in our previous studies [2, 3], where the total number of branches was chosen as 294. The problem with having many branch calculations is that each branch calculation takes the same amount of time as any other calculation, including those with different geometries. Therefore, in case of having limited computation resources, a hard choice needs to be made in order to choose the number of branch calculations versus the number of geometries. From our previous experience, we empirically observed that the number of branches plays lesser role compared to the number of geometries. To evaluate it using a systematic approach, we made the following steps.

First, shuffled the training dataset so that each 1900 training samples picked from it contained 1900 different geometries on average. Then, we randomly picked 3 smaller datasets from the training dataset. The chosen numbers of samples were 5760, 23040, and 46080. Since the data was shuffled beforehand, each of those sub-datasets contained around 1900 geometries and variable number of branch points for each of those geometries. Divided by 1900, those sample number yield around 3.03, 12.12, 24.25 of branch sizes per geometry.

These chosen smaller datasets were evaluated using the neural network model that we developed for this task, and which is described in [2]. To improve the variance of the model prediction, we decided to use the ensemble model approach as discussed in [4]. Each training dataset was used for 10 models training, each training took 7000 epochs, the learning rate was chosen as 0.00012, which is lower than the default 0.001, thus producing less fluctuations during the model training and less variance in the final result. The mini-batch size was fixed for all models produced in this study and was chosen as 256.

The results for all obtained ensemble models are shown in Table II.

Table II: Mean Relative Difference of the Testing Dataset versus Predicted using Ensemble ANN models

Branches (avg.)	Mean	STD
3.03	0.782	0.036
12.12	0.694	0.023
24.25	0.69	0.039

The results given in Table II demonstrate that further increase of the branch number provides very little benefit to the trained model accuracy.

To make the result more practically useful, the results from Table II were approximated using built-in capability of Microsoft Excel. It was found that Power function has the most suitable shape for such task. General form of a Power equation can be stated as:

$$y = Ax^b \tag{1}$$

In Eq. (1), y is the target function, which in our case is Mean Relative Difference (MRD), x is the argument (in our case – number of training samples), A and b – coefficients that are to be determined either manually or using specialized software. In our case, the value of b was found below 0, which shows the trend of MRD change with increased number of samples x.

With the obtained function, it is possible to estimate the MRD values for various numbers of training samples, including those that are way beyond the current dataset size. However, it is often more desirable to further process this estimation and produce Relative Gain (RG) of MRD per every additional training sample.

The philosophy behind RG is the following. In our case, training samples are produced using computer code calculation. Each sample (regardless of branch or geometry) can be assumed to have fixed average time to be generated. Therefore, generating every new sample will add up to both the total computation time and to the gained accuracy of the resulting model. However, the time increase per sample is linear by definition, while the accuracy gain (or reduction of MRD) is close to the Power function shape, which is non-linear and decreasing as can be found from the data in Table II. Hence, at some finite number of samples the value of RG is expected to decrease so low that it would not be reasonable to spend extra resources for further generating samples.

In case of having a table of discrete dataset sizes similar to Table II, the value of RG for each additional number of samples can be found as:

$$RG = A\frac{y_{N-1} - y_N}{x_N - x_{N-1}} \quad (2)$$

In Eq. (2), (*N-1*) parameters are found for the previous number of samples (smaller number of samples *x* and larger MRD *y* than in (*N*) case). (*N*) parameters are found for the current (or target) number of samples (larger *x* and smaller *y* than in (*N-1*) case). *A* is a weighting parameter, which can be defined or assigned depending on other conditions. Regardless of the parameter *A* value, the shape of the obtained RG function is expected to have the trend as shown in Figure 1.

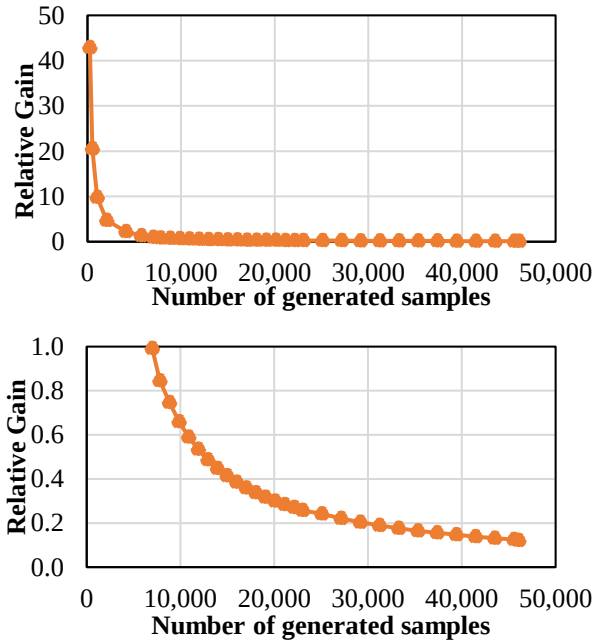

Fig 1. Shape of Relative Gain function for branches

The choice of the optimal RG value is expected to be experience-based, since it may vary based on the weighting parameter value or the particular data, as well as available computation resources. In our case, we decided to choose the number of samples where the value of RG shown in Figure 1 is close to 1.0. For this particular problem it was found around 7000 samples, which in return yields around 3.68 branches per geometry. It is important to note that this value is obtained not for certain branches but for random branch samples from the fixed list of Table I. Therefore, we can expand this dataset further by generating additional geometries with 3-4 random branches from the list.

3.2. Optimization of number of fuel assembly geometries

Based on the result obtained in Section 3.1, we know that the optimal number of branches for our particular problem can be assumed around 4 per geometry. Therefore, we can use the same methodology to obtain the target number of geometries for chosen number of branches. Using the same ensemble ANN model methodology as discussed in Section 3.1, we can obtain the results for the number of geometries with 4 random branches per geometry as shown in Table III.

Table III: Mean Relative Difference of the Testing Dataset versus Predicted using Ensemble ANN models

Geometries	Mean	STD
128	11.87	1.333
512	1.577	0.194
1216	0.911	0.037

Compared to the results shown in Table II, we can see the confirmation of the result obtained in Section 3.1. The impact of branches is less dominant compared to the impact of geometries. At the same time, the tested cases with different number of geometries show signs that adding more geometries can further improve the result. Using the same RG calculation technique, we found that the optimum number of geometries should be around 7000 as shown in Fig. 2.

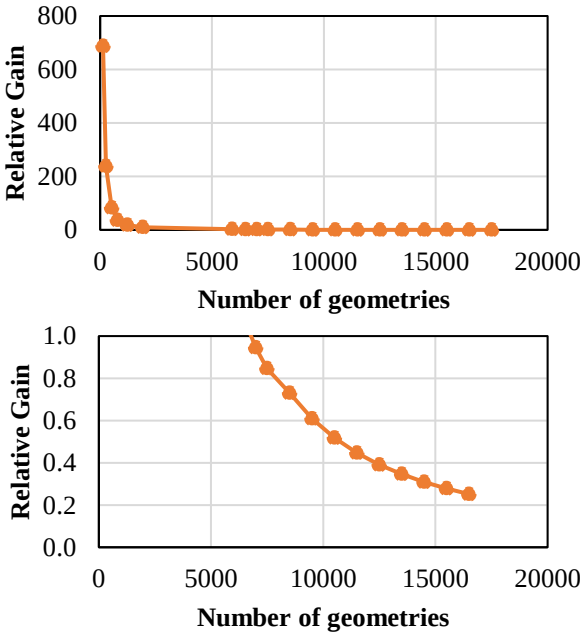

Fig 2. Shape of Relative Gain function for geometries

This time, we decided to test the estimated MRD value obtained using this method. For this purpose, we generated additional 8000 training geometries with 4 random branches each. We made sure that these additional geometries do not duplicate those used for testing and validation. This new data was merged with the already existing 1900 geometries, picking only 4 branches for the original cases. The estimation was predicting the MRD value around 0.275%. However, the actual ensemble ANN training showed the value of 0.313%, which is higher than the estimation due to uncertainty of the chosen extrapolation model. At the

same time, the higher actual MRD value means that the real RG for this case is even lower, which proves the efficiency of the chosen approach.

Finally, the result of ANN model prediction for the additional testing data with continuous random enrichments and branch parameters was tested with the last generated model. The MRD for that testing data was found around 0.66%, which is higher than the original testing data (0.313%). At the same time, this result can be considered reasonable since the additional testing dataset demonstrates an extreme level of generalization. As mentioned in the end of Section 2, the additional testing dataset is using all random FP enrichments continuously ranging from 1.0 to 5.0 wt% placed at random places of the FA, as well as random branch parameters that have continuous values in range of training data values. Therefore, such kind of chaotic data structure was not seen by a neural network neither at the stage of training, nor at the stage of validation, and can hardly be expected in real world practical scenarios.

4. Conclusions

In this study, we demonstrated the method of training dataset optimization, which was used for preparing training data for our ANN model that is predicting homogenized macroscopic XS, pin-wise power distribution inside FA and other parameters required for nodal diffusion calculation. The abovementioned method was applied to the optimization of branch calculation number per one FA geometry and to optimization of the total number of training FA geometries.

In case of number of branches, it was shown that this value can be significantly reduced (namely from original 90 to 4) without significant impact on the final result. As for the number of geometries, the final generated number stopped at the value of 9900, which results in 39,600 total training samples (4 branches per geometry). This number is significantly lower than the originally generated 171,000 samples (1900 geometries with 90 branches each). At the same time, the Mean Relative Difference obtained using independent geometry testing data was found significantly lower for the optimized dataset (0.313% versus the original 0.719%).

In future studies, we are planning to continue our work on the topic by adding further optimization steps that can save even more computation time and disk resources.

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 (online).

[3] S. Dzianisau, J. Choe, A. Cherezov, D. Lee, Acceleration of Nodal Diffusion Calculations Using Machine Learning-Driven Generation of Homogenized Macroscopic Cross-Sections, ANS M&C 2021, Raleigh, North Carolina, October 3–7, 2021 (accepted).

[4] S. Dzianisau, M. F. Khandaq, D. Lee, Impact of Training Dataset Reduction on the Prediction Accuracy of Nuclear Design Parameters using Convolutional Neural Network, Transactions of the Korean Nuclear Society Spring Meeting, Jeju, Korea, May 13-14, 2021 (online).

[5] 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.