*Transactions of the Korean Nuclear Society Autumn Meeting*
*Gyeongju, Korea, October 26-27, 2023*

# Research on Surrogate Models of Space-time Neutron Dynamics with Artificial Neural Networks

Xiaoqi Li, Youqi Zheng*, Xianan Du, Yongping Wang, Huabei Yin, Bowen, Xiao
*School of Nuclear Science and Technology, Xi'an Jiaotong University, Xi'an, 710049, China*
**Corresponding author: yqzheng@mail.xjtu.edu.cn*

***Key words:*** neutron spatiotemporal dynamics, Artificial neural network, Physical information guidance

## 1. Introduction

During the operation of the reactor, the neutron flux distribution will be affected by the introduction of external reactivity, which will lead to the change of the power distribution shape everywhere in the reactor core, and ultimately affect the thermal and safety analysis of the reactor core [1]. Therefore, in the core physics calculation program it is essential to simulate the time-dependent process, that is, calculate the neutron dynamics. This calculation can be performed by solving the point-reactor dynamic equation or the neutron space-time dynamic equation [2][3]. It is fast to solve the point-reactor dynamic equation, but it cannot simulate the spatial variation of neutron flux density or power. The spatial variation of neutron flux density or power can be accurately described by solving the neutron space-time dynamics equation. However, the computational is of huge complexity, which cannot meet the actual analysis needs such as reactor core design and operation state simulation. Therefore, it is significant to study the neutron dynamics method with both accuracy and efficiency for reactor operation and safety analysis.

In recent years, artificial neural network has developed rapidly, but in the field of nuclear core physics calculation, it still has not played a similar application effect as other fields [4]. The reason can be summarized as follows.

1) The calculation of the physical quantities involved in the neutron spatiotemporal dynamics is complex, which not only needs to solve the time-dependent transport equation, but also needs to consider the delayed neutron effect. It would be important to decide the surrogate object.

2) It is costly to obtain the training samples for neural networks. Neural networks need to rely on a large number of samples for training, and the time and computing resources consumed by sample acquisition also need to be considered as a cost in the evaluation of new methods.

3) Black-box properties of neural networks. The mathematical and physical information of the problem itself is not involved in the training of the network, and the extrapolation generalization of the obtained network is poor.

In this paper, the prediction of time series problem is transformed into the prediction of the coefficients of a time-dependent function, where the function is based on a spatial mesh, that is, each mesh under investigation has a set of coefficients to be predicted. According to these coefficients, the power distribution at any time can be obtained by substituting them into the target time.

## 2. Theoretical introduction

### *2.1 Timing function of power*

The surrogate problem in this paper is based on the calculation of the neutron space-time dynamics problem, and the construction of the function refers to the point reactor dynamics theory. The dynamic equation of the reactor considering the delayed neutron effect is usually called the dynamic equation of the point reactor model [5].

$$\frac{dn(t)}{dt}=\frac{\rho(t)-\beta}{\Lambda}n(t)+\sum_{i}^{6}\lambda_i C_i(t) \tag{2-1}$$

$$\frac{dC_i(t)}{dt}=\frac{\beta_i}{\Lambda}n(t)-\lambda_i C_i(t) \quad i=1,2,\cdots,6 \tag{2-2}$$

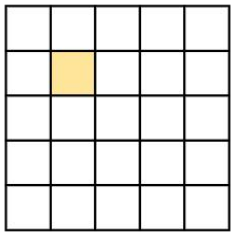
Where, $n(t)/C_i(t)$ are the changes of neutron density and the precursor nucleus concentration of group i delayed neutrons with time, respectively. $\rho(t)$ is the change of reactivity with time; $\Lambda/\beta_i/\lambda_i$ are the neutron generation time, group i delayed neutron share and delayed neutron constant, respectively, are the characteristic parameters of the reactor; $\beta$ is the total percentage of delayed neutron. When step reactivity is introduced, the general solution of the problem will consist of seven exponential items:

$$n(t)=n_0\left(A_1e^{\omega_1 t}+A_2e^{\omega_2 t}\cdots+A_7e^{\omega_7 t}\right) \tag{2-3}$$

Where Ai is the amplitude parameter, $\omega_i$ is the frequency parameter, and n0 is the neutron density in the initial state. For a given disturbance, if the values of the parameters $A_i$ and $\omega_i$ can be determined, the neutron density at any time can be accurately calculated from the above equation. In this paper, based on the time function expression in the point-reactor model, the value of each observation mesh is performed as prediction target, and a neural network model for rapid prediction of core power with disturbance as input is built.

### *2.2 Neural Network model*

The model built in this paper takes the location and the magnitude of the disturbance as variables, and expresses the introduction position in the form of "one-hot code". The value of the undisturbed position is 0, and the disturbance position is the measure of the disturbance amount (e.g. Material density change, control rod position change, etc.), as shown in Figure 1-(a), the disturbance x is introduced at the mesh (2,2), then the form of the input is the array shown in Figure 1-(b).



$$\begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Fig.1. Take the form of a "one-hot code" as input, with the input position filled with the quantity of introduction, and the other position is 0

Since the prediction target is the coefficient of the function expression, the coefficients need to be extracted from the training samples, and the Curve_fit function in Python is used in this work. The Curve_fit function is based on optimization, which automatically fits the coefficients according to the function form provided by the user, the coefficients to be fitted, and the input and output values. Due to the large number and nonlinearity of the parameters to be fitted, if the parameters are fitted without constraints, messy parameter combinations that deviate from the physical background of the problem will be obtained, which is not conducive to the subsequent training of neural networks. Therefore, a set of initial values and constraints should be provided when using curve_fit. In this paper, it is considered that there is a slight difference between the relative change of individual mesh power and the relative change of total power when reactivity is introduced. Therefore, the fitting parameter group of mesh power over time can be found according to the parameter group corresponding to the total power change function. The obtained reactivity equation (2-4) is derived from the point-heap equation,

$$\rho = \frac{l\omega}{1+l\omega} + \frac{1}{1+l\omega}\sum_{i=1}^{6}\frac{\beta_i \omega}{\omega+\lambda_i} \qquad (2\text{-}4)$$

Where, $l$ / $\beta_i$ / $\lambda_i$ are the neutron lifetime, the delayed neutron share, and the delayed neutron constant respectively; are the characteristic parameters of the reactor; given the introduced reactive disturbance ρ, the corresponding 7 ω can be solved. Since this equation is of 7th order and cannot be solved directly, graphical methods are generally used, as shown in Figure 2. In this paper, the gradient descent solver tool grad_opt is used to calculate the corresponding seven ω values under the given disturbance in the 7 solution domains, respectively. grad_opt is an optimization tool for solving the minimum value of a function provided by the DAKOTA[6] toolbox, which can quickly calculate the optimal solution within a given value range. The solved ω is then used to calculate the amplitude parameter A according to equations (2-5).

$$A_j = \rho_0 \left\{ \omega_j \left[ \Lambda + \sum_{i=1}^{6} \frac{\beta_i \lambda_i}{\left(\omega_i + \lambda_i\right)^2} \right] \right\}^{-1} \qquad (2\text{-}5)$$

At this time, the parameter set corresponding to the expression of the power change of the full reactor after the introduction of reactivity has been obtained. This set of parameters is used as the initial value and provided to curve_fit to fit the power change function of the each mesh.

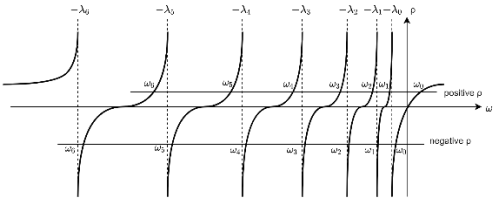



Fig. 2. Schematic of the graphical method for the solution of the reactivity equation

So far, the agency problem that this network can deal with is the introduction of one step disturbance. In this work, a sequence network model is built to fit the situation where the disturbances are introduced successively. Recurrent neural networks (RNNs)[7] are a special family of neural networks for processing and analyzing sequential data such as time series data. The RNN model performed in this work is illustrated as shown in figure 3, where the input is the disturbances presented in time series and

the same with the output.

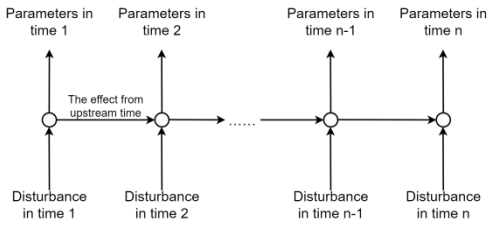



Fig. 3. The structure of RNN in this work

### *2.3 Calculation flow*

The computational flow of this work is showed in figure 5.

1) Firstly, the Python script is used to generate the samples of random disturbance position and random disturbance amount, transform them into input form, and pass them to the neutron spatiotemporal dynamics calculation program SARAX-TRANSIENT[8], and output the power change state corresponding to each sample with time.

2)Secondly, the curve_fit function in Python is used to extract the parameter combination of each sample, which is used as the input label to make the training sample set. In order to obtain a reasonable fitting effect, the initial value of the parameter combination is calculated according to the total amount of each sample.

3) The training samples were provided to the neural network prediction model built using Keras framework for training;

4) After the 1) -3) training operation is completed, the change function associated with time of the core power shape for predicting the same type of reactivity introduction event can be obtained.

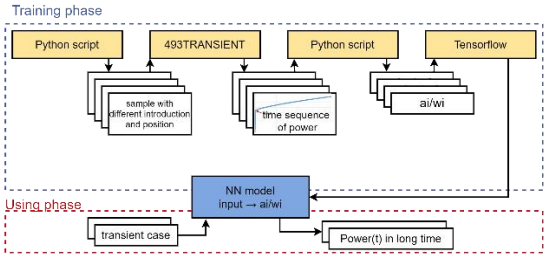



Fig.4. Flow chart

## 3. Numerical calculations

In this paper, the model provided by ANL-BSS16[9] one-dimensional benchmark problem is taken as an example to introduce the process of training and using neural networks. In the benchmark problem model, zones 2 and 6 are filled with active material containing fission medium, and reactivity is introduced by means of density change. This is shown in Figure 4.

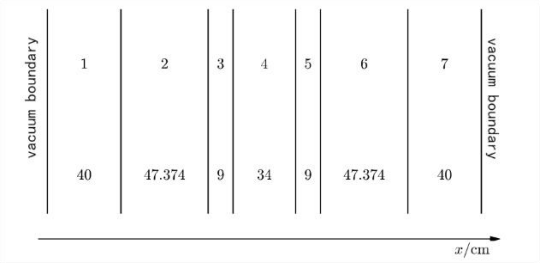



Fig. 5. ANL-BSS16 benchmark problem

Figure 6 shows the schematic diagram of the computational mesh of the model, the yellow mesh area is the mesh that may introduce disturbance, and the power changes of a total of 30 meshes will be used as the target for calculation and prediction.

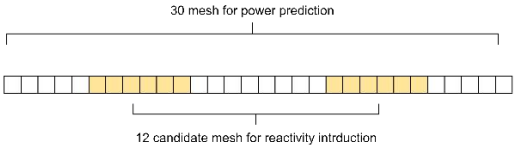



Fig. 6. Modeling

### *3.1 Selection of model parameters*

The combination terms of the seven groups of $A_i$ and $\omega_i$ in Equations (2-4) were selected within the acceptable range of accuracy. Taking the model described in 3.1 as an example, as shown in Figure 7, 2.1% positive density disturbance (corresponding to +0.11β reactivity) was introduced into mesh No. 7 of the model, and the power change with time was observed at mesh No. 10. To obtain the highest function fitting accuracy, all seven exponential fitting terms should be retained.

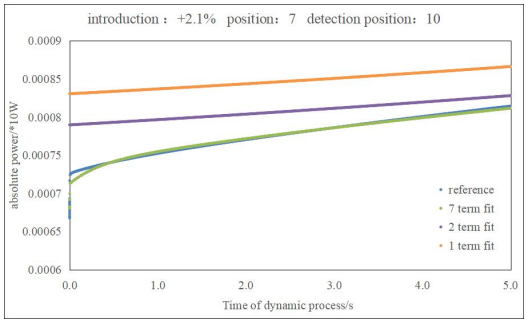



Fig.7. Confirmation of fitting function accuracy: Take the sample with 2.1% density change as an example

### *3.2 Model Construction*

Samples with different disturbance locations and different amounts of disturbance were randomly generated, and the dynamics of 2s was calculated using the SARAX-TRANSIENT program, and the power level changes of 30 meshes were output. Curve-fit function is used

to fit and record sample parameters and generate sample sets for mesh training and validation. The neural network is built using the Python-based Keras framework, and the structural information is shown in Table 1.

Table 1: the neural network structure in this work

| **Convolutional Layer** | | |
|---|---|---|
| **Layer** | **Number of convolution kernels** | **Size of convolution kernels** |
| c1 | 8 | 5 |
| **Fully Connected Layer** | | |
| **Layer** | **Number of node** | **Activation Function** |
| f1 | 32 | ReLU |
| f2 | 64 | ReLU |
| f3 | 128 | ReLU |
| f4 | 64 | ReLU |
| f5 | 32 | ReLU |
| f6 | 30 | ReLU |

### *3.3 Calculation results*

A) Single Introduction

The prediction function is constructed by the trained network and brought into the test sample respectively. The fitting parameters of the problem shown in Figure 6 are predicted and brought back to the time function. The comparison results with the reference values are shown in Figure 8. The predicted values are in good agreement with the 7-term fitting equation, and then fit with the reference values as expected.

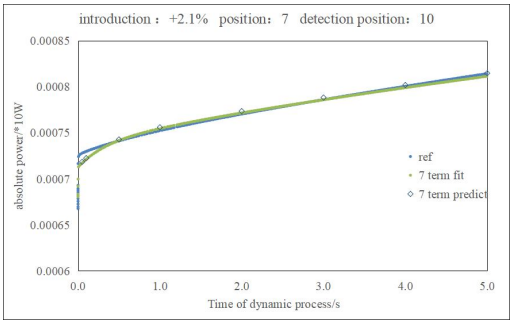



Fig. 8. Comparison between prediction and reference

Several groups of test samples were randomly generated, and the trained network was used to predict, taking the sample shown in Figure 9 as an example, the error bars in are 1% and 3%[1] respectively, where the results are as expected.

[1] The expected error is 10% and we use 1% and 3% error bars for an envelope of the results.

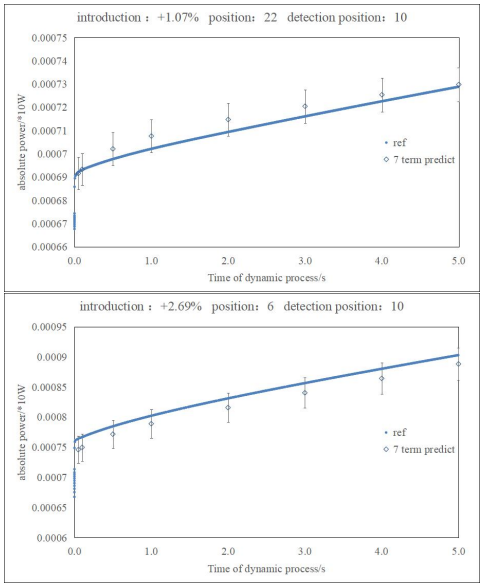



Fig. 9. Test sample results (reactivity introduced +0.056β and +0.14β, respectively)

B) Multiple introduction

Generate a set of samples randomly for the multi-point introduction problem. Since the input mode has the characteristic of "one-hot code", it can be used directly without changing the network structure. After training, the network can predict the introduction problem of random combinations, and the error distribution is shown in Figure 10. Among them, since the calculation target in this paper is the power distribution, the 30 meshes of the current problem are used as 30 comparison objects to compare with the reference value to ensure good local features when calculating the error distribution. The error of the 200*30 comparison results here is controlled within 10% and is normally distributed.

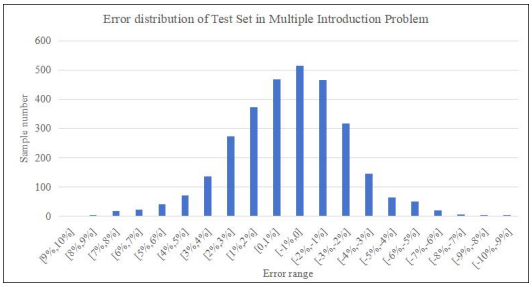



Fig. 10. Error distribution of Test Set in Multiple Introduction Problem

C) Time series introduction

The same with A) and B), a set of samples in time series has been generated. An example is selected as demonstration to show the way of fitting the mesh power with successive disturbance as showed in fig , where the reactivity introduction in time 0, 0.5s, 1s, 1.5s

and 2s is +0.032β, +0.042β, +0.044β, -0.033β and -0.009β respectively. The parameters of exponential function will change with the new introduction while still be effected by the history, which is suitable for performing RNN, as showed in figure 11.

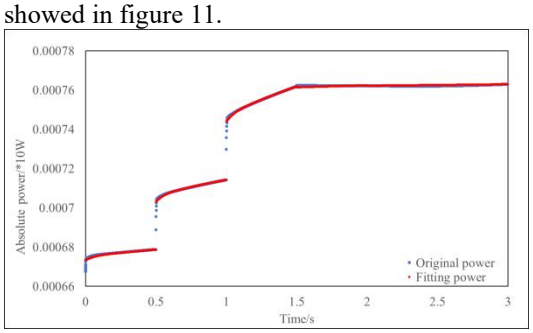



Fig. 11. The fitting of mesh power in serial disturbance problem with peicewise exponential function

In this work, the number of time sequence is 5, and the final mesh power is selected for error evaluation. The error distribution is showed in figure 12 . Since error will accumulate in the process, where the initial power of next time step is the terminal power of last step, the error of final results will relatively larger, which can be enveloped by 20%.

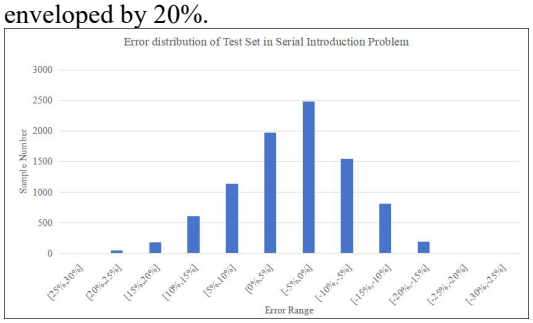



Fig. 12. Error distribution of test set in serial introduction problem.

## 4. Conclusion and Prospect

In this paper, the dynamic change process of nuclear core power is fitted by the time function expression, and the convolutional neural network is used to predict the change of the expression coefficient with the input, and the continuous-time prediction problem is converted into the prediction of a small number of discrete coefficients. The test results are good on the one-dimensional model ANL-BSS16, which provides a new idea for the surrogate model research of nuclear core transient calculation. The work of this paper will continue to be carried out to model, train and test on 3D models, and consider more complex working conditions to achieve the ultimate goal of assisting to solve the actual needs of engineering.

## REFERENCES

[1] A. Zhang, J. Teng, Y. Ju, T. Yang. Prediction of Nuclear Reactor Core Thermal Power Based on Artificial Neural Network[J]. Computer Simulation, 2021

[2]Z. Cai. Nuclear Power Reactor Neutron Dynamics[M]. Beijing: National Defense Industry Press, 2005

[3]Z. Huang. Fundamentals of Nuclear Reactor Dynamics [M]. Beijing: Peking University Press, 2014

[4]Q. Huang, S. Peng, J. Deng, etc. A review of the application of artificial intelligence to nuclear reactors: Where we are and what's next[J]. Heliyon, 2023

[5]Z. Xie. Physical Analysis of Nuclear Reactor [M]. Xi 'an: Xi 'an Jiaotong University Press, 2004

[6]D. Keith , E. Michael, G. Gianluca, et al, Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.11 User's Manual. 2019

[7]D. Rumelhart, G. Hinton, R. Williams, Learning representations by back-propagating errors. Nature 323, 533–536 (1986).

[8] H. Lu. Research on node method of neutron diffusion and transport based on triangle mesh [D]. Xi 'an: Xi 'an Jiaotong University, 2007

[9]ANL. NATIONAL ENERGY SOFTWARE CENTER: BENCHMARK PROBLEM BOOK