NUCL SCI TECH (2021) 32:62
https://doi.org/10.1007/s41365-021-00901-2

Hybrid windowed networks for on-the-fly Doppler broadening in RMC code

Tian-Yi Huang[1] · Ze-Guang Li[1] · Kan Wang[2] · Xiao-Yu Guo[2] · Jin-Gang Liang[1]

Received: 4 January 2021 / Revised: 25 March 2021 / Accepted: 16 April 2021 / Published online: 14 June 2021

Abstract On-the-fly Doppler broadening of cross sections is important in Monte Carlo simulations, particularly in Monte Carlo neutronics-thermal hydraulics coupling simulations. Methods such as Target Motion Sampling (TMS) and windowed multipole as well as a method based on regression models have been developed to solve this problem. However, these methods have limitations such as the need for a cross section in an ACE format at a given temperature or a limited application energy range. In this study, a new on-the-fly Doppler broadening method based on a Back Propagation (BP) neural network, called hybrid windowed networks (HWN), is proposed to resolve the resonance energy range. In the HWN method, the resolved resonance energy range is divided into windows to guarantee an even distribution of resonance peaks. BP networks with specially designed structures and training parameters are trained to evaluate the cross section at a base temperature and the broadening coefficient. The HWN method is implemented in the Reactor Monte Carlo (RMC) code, and the microscopic cross sections and macroscopic results are compared. The results show that the HWN method can reduce the memory requirement for cross-sectional data by approximately 65%; moreover, it can generate k_{eff}, power distribution, and energy spectrum results with acceptable accuracy and a limited increase in the calculation time. The feasibility and effectiveness of the proposed HWN method are thus demonstrated.

Keywords Monte Carlo method · Reactor Monte Carlo (RMC) · On-the-fly Doppler broadening · BP network

1 Introduction

With the increase in the computing power of high-performance computing platforms, Monte Carlo neutronics-thermal hydraulics coupling has become an ideal approach for obtaining accurate results for the design and analysis of reactors. Traditional methods of linear interpolation with point-wise nuclear data require a large amount of memory resources to provide temperature-dependent microscopic cross sections for simulations [1]. On-the-fly Doppler broadening methods have been introduced to reduce the memory cost and enable thermal-hydraulics coupled reactor analysis [2]. Several on-the-fly Doppler broadening methods have been proposed to meet both the efficiency and memory requirements. Three methods, namely, Target Motion Sampling (TMS) [3, 4], windowed multipole [5], and a method based on regression model and fitting [6] have been proposed for the evaluation of cross sections across the resolved resonance energy range. Walsh studied an on-the-fly Doppler broadening method for the unresolved resonance energy range [7]. Pavlou and Ji proposed a thermal energy range method [8].

This work was supported by the Science Challenge Project (No. TZ2018001), the National Natural Science Foundation of China (Nos. 11775126, 11545013, 11775127), Young Elite Scientists Sponsorship Program by CAST (No. 2016QNRC001), and Tsinghua University Initiative Scientific Research Program.

✉ Ze-Guang Li
lizeguang@tsinghua.edu.cn

[1] Institute of Nuclear and New Energy Technology, Tsinghua University, Beijing 100084, China

[2] Department of Engineering Physics, Tsinghua University, Beijing 100084, China

For heavy nuclides such as U-235, the data in the resolved resonance energy range account for most of the nuclear data. Among the methods applicable in this range, the TMS method requires cross-sectional data at a given temperature, such as 0 K [9, 10], while the windowed multipole method divides the resolved resonance energy range into energy windows and uses approximations to evaluate cross sections [1], Yesilyurt et al. used thirteen parameters for broad cross sections at 0 K to perform Doppler broadening for any temperature in the range of 77–3200 K [6]. This method was further developed by Liu et al., and the number of parameters was made flexible [11]. Both the TMS method and the method proposed by Yesilyurt et al. require ACE data in their processes, and the latter requires a larger memory for storing the broadening parameters. The range of nuclides for which the windowed multipole method is applicable is limited.

In this paper, a new on-the-fly Doppler broadening method based on BP neural networks, called hybrid windowed networks (HWN), is proposed. The total memory requirements are reduced by approximately 65% compared with ACE data at the expense of efficiency over the resolved energy range. The BP neural networks are used to evaluate the cross sections. The neural network method, which is a type of machine learning method, simulates the structure of biological neurons through establishing artificial neural networks [12]. By relying mainly on multi-layer neuron nodes with various weights and biases, neutron networks can be used to solve complicated problems such as image [13] or audio [14] processing. Neural networks with simple structures also exhibit a satisfactory performance in data fitting [15]. The structures and training parameters of the networks reported herein were carefully determined to meet the needs of cross-sectional training.

In this method, the resolved resonance energy range is divided into windows. The networks trained for each window can be used independently so that the method can be easily combined with other on-the-fly Doppler broadening methods. The application range of this method can be set by the users to avoid unacceptable losses of efficiency.

The results confirm the feasibility of evaluating complex cross-sectional parameters through the use of neural networks. The potential of neural networks for memory saving is demonstrated in this work. Neural networks can be used to evaluate some of the parameters in the calculation of other developed on-the-fly Doppler broadening methods. Larger memory savings and higher accuracy might be achieved by incorporating the physics of Doppler broadening into the method.

The principle of BP neural networks and the HWN method are introduced in Sect. 2. In Sect. 3, the results of numerical tests conducted to verify the effectiveness, accuracy, and efficiency of the method are reported and discussed.

2 HWN Method

In the HWN method, the resolved energy is divided into energy windows based on the number of extreme points in the cross section. In each window, two BP networks are used to calculate the cross section at 200 K and broaden the cross section to temperatures in the range of 250–1600 K. ACE data are used to train the BP networks. The networks for each window can be used independently; thus, the scope of the method can be set easily.

Section 2.1 reviews the principle of BP neural networks. Section 2.2 describes the training and calculation process of the two networks within a window. Section 2.3 introduces the division of the resolved energy range and the parameter determination process.

2.1 BP Neural network

An efficient and memory-saving method for evaluating cross sections is needed for on-the-fly Doppler broadening. Neural networks, which are widely used in machine learning, need to be trained before they are used. Once the parameters of the networks are determined, the networks can be used to calculate the output from the given input. After training, the amount of calculations required is greatly reduced; therefore, this method can be used for on-the-fly Doppler broadening. A combination of the input and expected results is used in the training process, during which the weights and biases of all the neurons are adjusted. The deviation between the outputs of the network and the expected results is gradually reduced during training until the network meets the requirements.

Many studies on neural networks such as convolutional neural networks [16] and deep neural networks [17] have been conducted. Such networks have many hidden layers and neurons, as well as complex structures, resulting in low computational efficiencies. Because computing speed is important in Monte Carlo codes, a neural network with a simple structure is more suitable.

In this study, the back propagation (BP) network was used. The error back propagation algorithm proposed by Rumelhart [18] introduced in the following paragraphs is used for training this network. As shown in Fig. 1, a BP network consists of an input layer, hidden layers, and an output layer. Each layer contains a certain number of neurons. The simplicity of the structure and calculation steps of this type of network ensures its high computational efficiency.

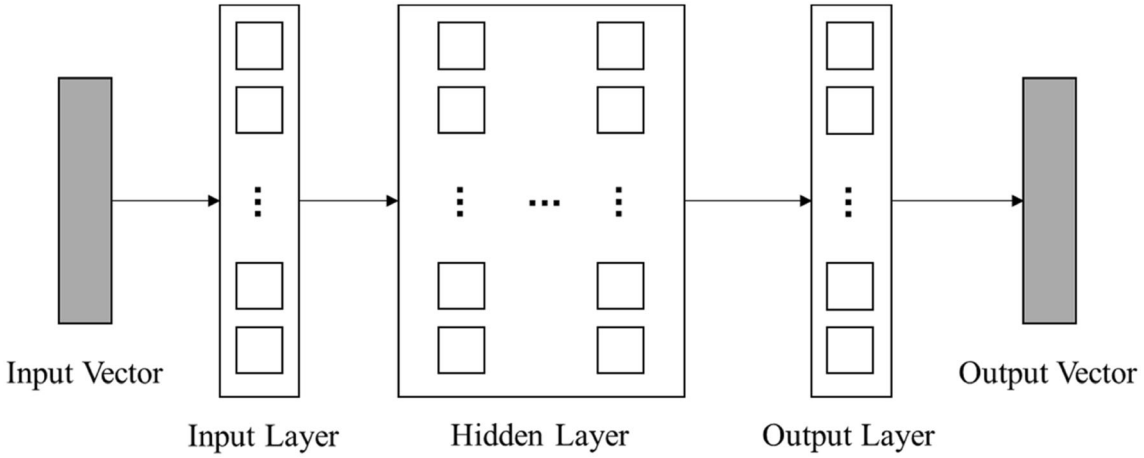


Fig. 1 Structure of BP neural network

The hidden and output layer nodes of the BP neural network are, respectively, described by

$$O_j = f_1\left(\sum_i \omega_{ij} x_i - b_j\right), \tag{1}$$

$$Y_k = f_2\left(\sum_j \omega_{jk} O_j - b_k\right). \tag{2}$$

In equations, O_j is the output of the hidden layer node; Y_k is the output value of the output layer node. f_1 and f_2 are the transfer functions of the hidden and output layer nodes, respectively. In general, f_1 is a nonlinear function, and f_2 can be a linear or nonlinear function. w_{ij} is the weight from the previous node to the hidden layer node. w_{jk} is the weight from the node of the last hidden layer to the node of the output layer. b_j and b_k are the biases of the hidden layer node and output layer node, respectively.

The weights and thresholds of the BP neural network in calculations are generally expressed in matrix form. Equations (1) and (2) are therefore expressed as Eqs. (3) and (4), respectively.

$$\boldsymbol{O}_{\mathrm{h}} = f_1\left(\boldsymbol{W}_{\mathrm{pretoh}}\boldsymbol{X}_{\mathrm{pre}} - \boldsymbol{B}_{\mathrm{h}}\right) \tag{3}$$

$$\boldsymbol{Y} = f_2(\boldsymbol{W}_{\mathrm{h\ to\ o}}\boldsymbol{O}_{\mathrm{h}} - \boldsymbol{B}_{\mathrm{o}}) \tag{4}$$

The subscript "pre" means the previous layer, which can be a hidden layer or an input layer. Subscript "h" and "o" means hidden layer and output layer, respectively. In Eq. (3), $\boldsymbol{O}_{\mathrm{h}}$ is the output value of the hidden layer. $\boldsymbol{W}_{\mathrm{pretoh}}$ is the matrix of weights from the previous layer to the hidden layer. $\boldsymbol{X}_{\mathrm{pre}}$ is the matrix of output values of the previous layer and $\boldsymbol{B}_{\mathrm{h}}$ is the matrix of bias of the hidden layer. In Eq. (4), $\boldsymbol{Y}$ is the matrix of output value of the output layer, and it is also the output value of the network. $\boldsymbol{W}$ and $\boldsymbol{B}$ in Eq. (4) have similar meanings to those in Eq. (3). The bias and output data of each layer are column vectors with lengths equal to the number of nodes in the layer. The weight is a matrix where the first and second dimensions are the number of nodes in the previous and current layers, respectively. The matrix elements correspond to those in Eqs. (1) and (2).

The neural networks in this study were trained using MATLAB. The output of the neural networks approached the target value over successive iterations. All the weights and biases in the network were adjusted during the training process, while the structure of the network, including the number of hidden layers and nodes in each layer, and the transfer function, remained unchanged.

The error back propagation algorithm was used for training the network. The error between the result in the training data and the corresponding result calculated by the network for a given input in the training data was propagated back to the parameters of each layer. The process is briefly described as follows.

The set of input vectors and the corresponding target vectors are denoted as

$$\boldsymbol{P} = (p_1, p_2, \ldots, p_n)^T, \tag{5}$$

$$\boldsymbol{T} = (t_1, t_2, \ldots, t_k)^T. \tag{6}$$

The square of errors between $\boldsymbol{P}$ and $\boldsymbol{T}$ are summed to defined R, which is the error of the network. The factor $1/2$ is to simplify the following process.

$$R = \frac{1}{2}\sum_k (Y_k - t_k)^2 \tag{7}$$

The goal of the training is to reduce the error. R is therefore expanded using Eq. (4) into the node parameters of the output layer as follows:

$$R = \frac{1}{2}\sum_k \left[f_2\left(\sum_j w_{jk} O_j - b_k\right) - t_k\right]^2. \tag{8}$$

Equation 8 can be further expanded using Eq. (1) as follows:

$$R = \frac{1}{2}\sum_k \left[f_2\left(\sum_j w_{jk} f_1\left(\sum_i w_{ij} x_i - b_j \right) - b_k \right) - t_k \right]^2. \tag{9}$$

The expansion ends here if the network has one hidden layer. Otherwise, R can be expanded layer by layer. The algorithm is illustrated for a one-hidden-layer network for which the transfer function of the output layer is given by

$$f_2(x) = x. \tag{10}$$

The partial derivatives of R in Eq. (8) are

$$\frac{\partial R}{\partial b_k} = -\left(\sum_j w_{jk} O_j - b_k - t_k \right), \tag{11}$$

$$\frac{\partial R}{\partial w_{jk}} = O_j \left(\sum_j w_{jk} O_j - b_k - t_k \right). \tag{12}$$

The partial derivatives are called the gradient values of the error. The parameters are updated using the gradient values in each training Iteration as follows:

$$b_{k(n+1)} = b_{k(n)} - \eta \left(\frac{\partial R}{\partial b_k} \right)_{(n)}, \tag{13}$$

$$w_{jk(n+1)} = w_{jk(n)} - \eta \left(\frac{\partial R}{\partial w_{jk}} \right)_{(n)}, \tag{14}$$

where η is the learning rate, which determines the ratio of the correction to the parameters. The amplitude of a single adjustment is directly proportional to η. The derivation shown in Eqs. (1–14) is applied to each individual parameter, and the end result of the derivation can be expressed in the form of a matrix as

$$\boldsymbol{\Delta} = \boldsymbol{Y} - \boldsymbol{T}. \tag{15}$$

The results of Eq. (13) and (14) for all the parameters can be collectively expressed as

$$\boldsymbol{B}_{\mathrm{o}(n+1)} = \boldsymbol{B}_{\mathrm{o}(n)} + \eta \boldsymbol{\Delta}, \tag{16}$$

$$\boldsymbol{W}_{\mathrm{h\ to\ o}(n+1)} = \boldsymbol{W}_{\mathrm{h\ to\ o}(n)} - \eta \boldsymbol{O}_{\mathrm{h}} \boldsymbol{\Delta}. \tag{17}$$

The training of the other parameter matrices can be described in a similar manner. The error plays a role in the parameter adjustment of each layer through the result deviation $\boldsymbol{\Delta}$ and propagates to each parameter matrix.

2.2 Training and calculation

This section describes how the two networks were trained and used in an energy window. The method for dividing the resolved energy is introduced in the next section. Both the energy division and the sequential computation using two networks are designed to reduce the complexity of the training target.

A temperature is chosen as the base temperature for the cross section of a nuclide. The cross section at this temperature, denoted as $\sigma_{T_0}(E)$, is a function of the neutron energy E. The cross-sectional broadening coefficient, K, is introduced, which is defined as the ratio of the cross section at a temperature not less than T_0 to the cross section at T_0.

$$K(E,T) = \frac{\sigma(E,T)}{\sigma_{T_0}(E)} (T \geq T_0) \tag{18}$$

K is a binary function with the independent variables E and T and denoted as $K(E,T)$. The features of $K(E,T)$ are much less complex than those of $\sigma(E,T)$. Thus, the ratio of the maximum value of $K(E,T)$ to its minimum value is much lower than that of $\sigma(E,T)$ within an energy window obtained by the method explained in Sect. 2.3. Therefore, it is expected that better results can be obtained using $K(E,T)$ as the input data for network training.

The HWN method is similar to the method proposed by Yesilyurt et al. [6], which also uses fitting parameters for on-the-fly Doppler broadening. However, the broadening coefficient K in the proposed method applies for all the energies in the corresponding energy window, whereas the method proposed by Yesilyurt et al. requires individual parameters for each energy point.

To reduce the memory required, a network denoted as Network 1 is trained to calculate the cross section at temperature T_0. The upper and lower bounds of the energy window are denoted as $E_{\min}$ and $E_{\max}$, respectively. The corresponding relationship between the input and output of the network is expressed as

$$\sigma'_{T_0}(E) = F_1(E) (E_{\min} \leq E \leq E_{\max}), \tag{19}$$

where σ'_{T_0} is the cross section calculated by Network 1 at T_0 and F_1 is the mapping relationship between the input energy and output cross section of Network 1. It should be noted that some errors exist between the results and the actual cross section.

Network 2 is trained to calculate the broadening coefficient, K. If K is used directly as the input data for training Network 2 and Eq. (18) is used to calculate the broadened cross section, the error of the calculation result will be the superposition of the errors of the two networks.

To reduce the error, the K value calculated by the following formula is used as the training target.

$$K(E,T) = \frac{\sigma(E,T)}{\sigma'_{T_0}(E)} (E_{\min} \leq E \leq E_{\max}, T \geq T_0) \tag{20}$$

The difference between Eqs. (20) and (18) is that the cross section at T_0 in Eq. (18) is replaced by the output of Network 1 in Eq. (20). Therefore, the influence of the accuracy of Network 1 on the final result can be eliminated. However, the complexity of K will increase if the error of Network 1 is extremely large, which may affect the accuracy of Network 2. It is therefore necessary to control the error of Network 1.

K in Eq. (20) is used as the input data to train Network 2. The temperature range of the network fitting is limited by the input data. A temperature range denoted as $T_{\min} \le T \le T_{\max}$ that could meet the actual requirements is selected. $T_{\min}$ should not be lower than T_0. The corresponding relationship between the input and output can therefore be expressed as

$$K'(E,T) = F_2(E,T)(E_{\min} \le E \le E_{\max}, T_{\min} \le T \le T_{\max}), \tag{21}$$

where K' is the broadening coefficient calculated by Network 2, and F_2 is the mapping relationship between the input energy and the output coefficient of Network 2. K' is a function of the independent variables E and T and is denoted as $K'(E,T)$.

By rewriting Eq. (20), the equation for on-the-fly Doppler broadening is obtained as

$$\begin{aligned}\sigma(E,T) &= \sigma'_{T_0}(E)K'(E,T) \\ &= F_1(E)F_2(E,T)(E_{\min} \le E \le E_{\max}, T_{\min} \le T \le T_{\max}).\end{aligned} \tag{22}$$

Equation (22) describes the Doppler broadening process. Network 1 is first used to calculate the cross section at the given energy E and basic temperature T_0. Then, the cross section is broadened to the temperature T using the broadening coefficient K calculated by Network 2.

The values of T_0, $T_{\min}$, and $T_{\max}$ were determined. The temperature ranges and their corresponding fields of study were summarized by Yesilyurt et al. [6] as shown in Table 1. Monte Carlo codes for reactors should be able to handle benchmarking calculations and reactor operation problems. Thus, $T_{\min}$ and $T_{\max}$ were set as 250 K and 1650 K, respectively. It should be noted K approaches 1 as T approaches T_0, which affects the accuracy of the neural network in the vicinity of T_0. As a result, training using data at $T = T_0$ should be avoided, and T_0 should not be close to $T_{\min}$. Considering the lower complexity of the cross-sectional curve at higher temperatures, the base temperature T_0 was chosen as 200 K to reduce the difficulty of the network training.

Table 1 Temperature ranges and their corresponding fields of study

Temperature range (K)	Field of study
77–293.6	Cold neutron physics
293.6–550	Benchmarking calculations
550–1600	Reactor operation
1600–3200	Accident condition

2.3 Window division and parameter determination

The cross-sectional curve in the resolved resonance energy range of heavy nuclides at a single temperature is very complex because of the large number of resonance peaks. The addition of the temperature dimension further increases the complexity. Therefore, it is impractical to train the neural network directly using the ACE data over the entire resolved resonance energy range as the input.

Dividing the resolved resonance energy range into energy windows can significantly reduce the difficulty of training in each window. Because similar processes are carried out for each window, the data should be divided evenly according to the training difficulty. Dividing the resonant peaks equally between the different windows is an easy and effective method. Each window has the same number of maximum points that correspond mainly to the positions of the resonance peaks within the resolved resonance energy range. The edges of the windows are set as the minimum points of the cross-sectional curve.

Because of the Doppler broadening effect, the resonant peaks are broadened at temperatures above 0 K. At $T_0 = 200$ K, some adjacent resonant peaks are combined into single peaks, leading to a reduction in the number of maximum points. The maximum points, rather than the resonant peaks at 0 K, were used for the deviation of the cross-sectional curve at T_0. Because the distributions of the resonant peaks differ for different heavy nuclides, the number of windows and the positions of the edges determined by this process will also be different.

The number of maximum points in each window was carefully determined. A smaller number of points for a given network would result in a higher number of divided windows and a corresponding increase in the memory required. In addition, if large number of peaks are included, the amount of data in each window will be extremely large, and the accuracy of the networks will be decreased. The number of points was set to 15 based on experience and testing. The aforementioned process was applied to the total cross-sectional curve at the base temperature of $T_0 =$ 200 K for window division.

Relatively large deviations near the boundary of the windows were often observed during network training. To avoid this, the windows were extended along both edges, as is shown in Fig. 2. The data in the extended window were used for neural network training, whereas the acquired

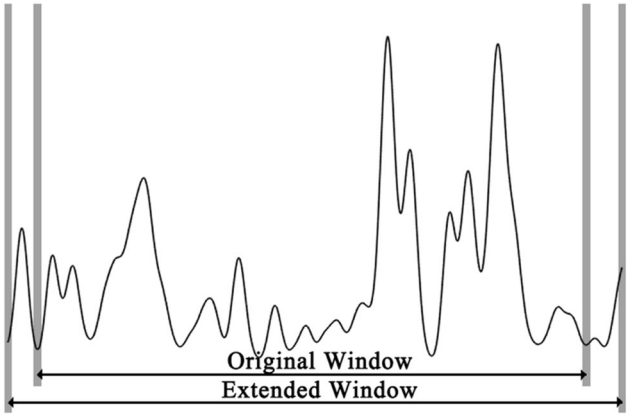


Fig. 2 Extension of window

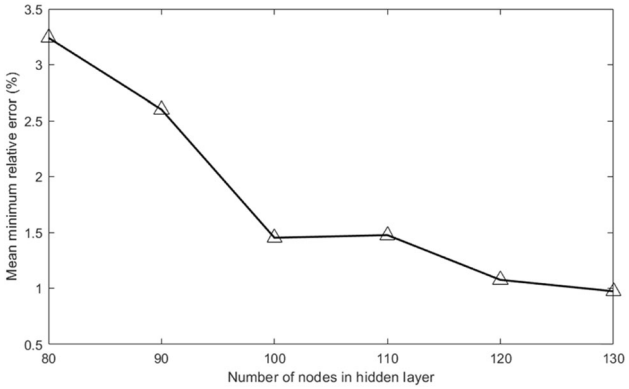


Fig. 3 Relationship between the number of nodes in the hidden layer and the mean minimum absolute relative error

neural network was used only within the range of the original window. The windows next to the unresolved resonance energy range and thermal energy range were only extended in the direction of the resolved resonance energy range.

The training results were greatly affected by the structure of the neural networks and the training parameters. The network parameters were determined according to the test results.

The training function *training* was used in network training. This approach has the advantages of a faster convergence speed and better accuracies. The transfer function of the hidden layer nodes is the *tansig* function in MATLAB, which is defined as

$$f_2(x) = \frac{2}{1 - \mathrm{e}^{-2x}} - 1. \tag{23}$$

A BP network with one hidden layer is sufficient for solving many problems. Networks with multiple hidden layers may show better performance in some cases [19]. A one-hidden-layer BP network was used for Network 1 (described in Sect. 2.2) to reduce the memory requirements. For Network 2, a two-hidden-layer network was used to avoid overfitting in the training of $K(E,T)$.

Tests were performed to determine the number of neurons in each hidden layer. The number of neurons in the hidden layer of Network 1 was determined first. The total cross-sectional data of U-235 at 0 K were divided into 181 windows, and 19 equally spaced windows were selected. Networks with 80, 90, 100, 110, 120, and 130 nodes in the hidden layer were trained using the data in the windows. Each network was trained with the data from each window 10 times, and the minimum value of the maximum absolute relative error was calculated. The geometric means of the error values for the 19 windows are shown in Fig. 3.

In general, the accuracy of the network increased with the number of nodes, and it was at an acceptable level when the number of nodes was 130. As shown in Fig. 3, a further increase in the number of nodes had a limited effect on the improvement of accuracy. Networks with 130 nodes in the hidden layers were used for most windows. A larger number of nodes were used in a few windows in which the relative error was extremely large.

Training and comparisons were performed to determine the number of nodes in the two hidden layers of Network 2. A reasonable maximum epoch number, training time limit, and accuracy target were set for training. Data from the second window of the U-235 total cross section were used to evaluate networks with different combinations of node numbers. The performance was determined using the percentage of data points where the absolute relative error with respect to the input data was less than 0.1%. The results are compared in Table 2. The results indicate that the best combination has 40 nodes in the first hidden layer and 20 nodes in the second hidden layer.

3 Numerical tests of HWN method

The HWN method was implemented in the Reactor Monte Carlo (RMC) code [20] for on-the-fly Doppler broadening of U-235. Data from the ENDF/B-VII.0 database were processed by NJOY [21] with an accuracy of no less than 0.001 to obtain the training ACE data. The resolved resonance energy range was divided into 80 energy windows, and the networks were trained for each window. cross-sectional data at 25-K intervals in the range of 250–1650 K were used to calculate the training target $K(E,T)$ in each window. The microscopic cross sections and macroscopic results were compared to prove the feasibility and effectiveness of the HWN method.

In Sect. 3.1, the accuracy and memory requirements of the HWN method are demonstrated by comparing the calculated microscopic cross section with the ACE data. In Sect. 3.2, the results of two macroscopic tests performed to

Table 2 Performance comparison of networks with different combinations of node numbers

Number of nodes in the first hidden layer N_1	Performance of networks with N_2 nodes in second hidden layer (%)				
	10	13	15	20	25
35	57.1	55.9	61.1	58.6	58.5
40	56.9	59.8	57.0	64.8	59.6
45	56.1	57.6	58.5	59.5	56.8
50	59.2	57.7	61.6	56.0	58.1
55	57.7	61.4	63.4	59.6	52.6

verify the accuracy and efficiency of the method are presented.

3.1 Microscopic accuracy and memory requirement comparison

The HWN method was applied to U-235 and U-238, which are representative heavy nuclides with important resonances. The total cross section, elastic scattering cross section, absorption cross section of U-235, and total cross section of U-238 were compared with the ACE data at the same temperature. The cross-sectional results and absolute relative errors with respect to the ACE data are plotted in Fig. 4, where the comparisons were made in the energy regions with strong resonances at 300 K and 700 K. It can be observed that the errors are low for most data points and that the cross sections evaluated are accurate at both temperatures. The relative errors fluctuate with the energy. The fluctuations are caused by the characteristics of neural networks with many nodes.

The HWN method and the windowed multipole method are both methods with new ways, which do not use continuous-energy data, of storing the cross-sectional data. Therefore, their memory requirements are lower than that of the ACE data. The relative errors of the HWN and windowed multiple [1] methods are compared in Table 3. The results show that the maximum relative errors are similar and that the average relative errors of both methods are within 0.1%.

The theoretical memory consumption of the network parameters and ACE cross-sectional data are compared. In the continuous-energy Monte Carlo code using the ACE data, the cross sections are stored in the form of double-precision floating-point numbers. Each double-precision floating-point number occupies eight bytes of memory. Both the energy grid and cross-sectional values are needed for cross-sectional evaluation. For the total cross section, elastic scattering cross section, and absorption cross section, four double-precision floating-point numbers, which occupy 32 bytes of memory, are needed for each energy point. In comparison, most of the parameters in the HWN method are stored in the form of double-precision floating-point numbers, and a few parameters are integers. The memory consumptions of the two methods are calculated using the number of parameters and the memory space needed for each parameter.

The ACE data at 0 K processed by NJOY were used for comparison. If on-the-fly Doppler broadening is not introduced to the Monte Carlo code, point-wise cross sections at more than a dozen temperatures will be needed for thermal-hydraulics coupled analysis. The method proposed by Yesilyurt et al. uses parameters that require several times the memory needed for the ACE data at 0 K. Point-wise data are also needed in the TMS method. As a result, the HWN method shows a significant reduction of the memory requirement compared to the 0 K ACE data.

The comparison results are listed in Table 4. The results show that for the three cross sections of U-235, the HWN method could reduce the memory requirements of cross-sectional data in the resolved resonance energy range by 66.1% as compared with the case of the 0 K ACE data. The memory requirement reduction was 65.9% over the entire energy range.

The networks for each window can be used independently because the training process of each window is independent. It is easy to set the scope of the method when the method is implemented in a Monte Carlo code. It is not necessary to store the 0 K ACE data within the selected windows if the HWN method is used. However, the speed of the cross-sectional evaluation in these windows will decrease. The efficiency drop is described in Sect. 4. Table 4 clearly shows that the resolved resonance energy range accounts for most the nuclear data. The memory optimization ratio is significant when the HWN method is used in some of the windows. Users can decide the efficiency to be compromised for saving memory.

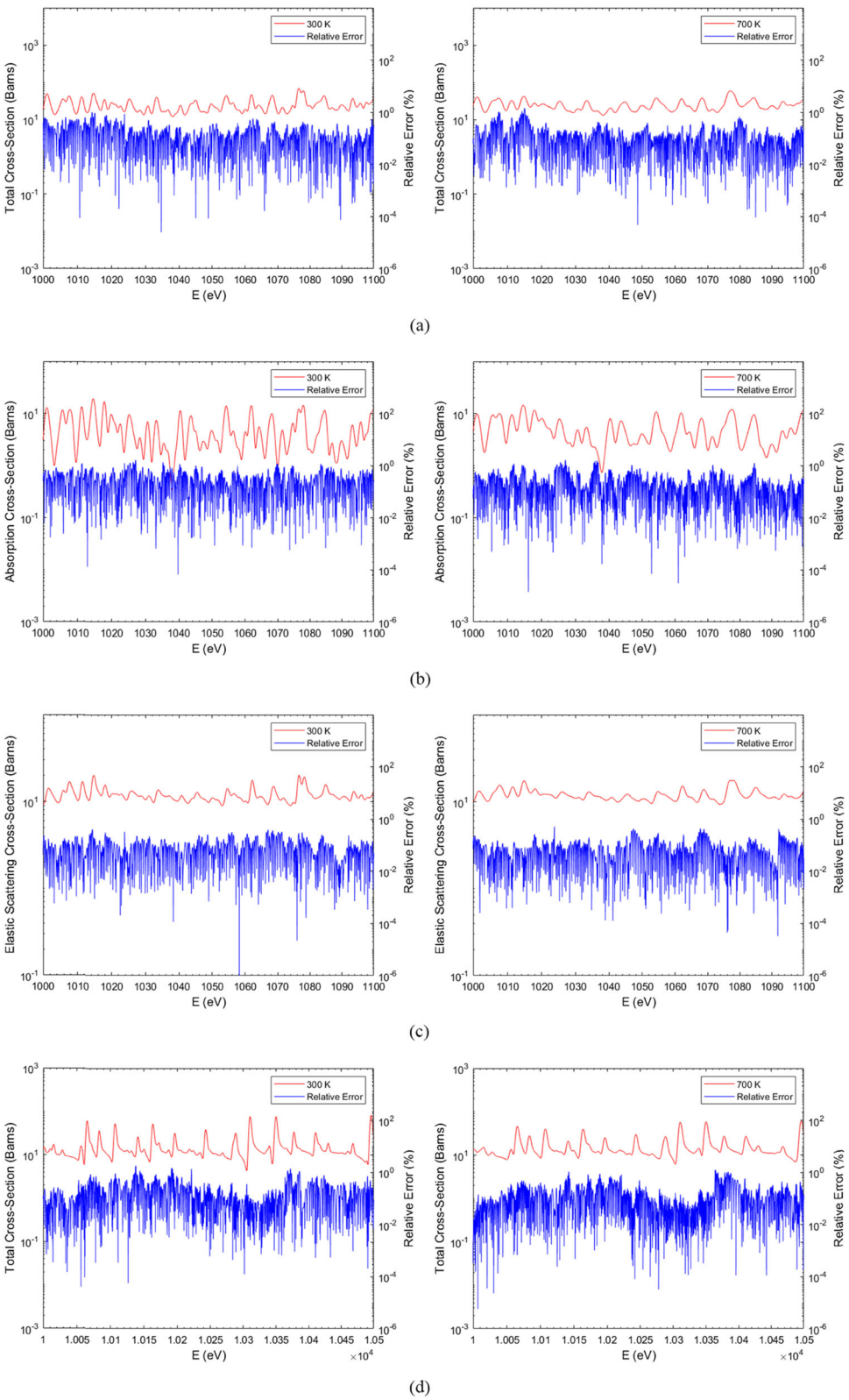


Fig. 4 (Color online) Comparison of microscopic cross sections at 300 K and 700 K. **a** Total cross section of U-235. **b** Absorption cross section of U-235 **c** Elastic scattering cross section of U-235. **d** Total cross section of U-238

Table 3 Comparison of relative errors of HWN and windowed multipole

Relative error	HWN method (%)	Windowed multipole (%)
Max relative error	~ 1	~ 1
Average error	< 0.1	< 0.1

3.2 Comparison of macroscopic test results

The HWN method was used for Doppler broadening in all windows of the resolved resonance energy in the HWN cases, while ACE data from the processed ENDF/B-VII.0 database at the same temperature were used in the ACE case. All calculations were performed using an Intel i7-9750H CPU without parallelism.

3.2.1 Concentric spheres

The first comparison example consisted of two concentric spheres, as shown in Fig. 5. The radii of the inner and outer spheres were 6.782 cm and 11.862 cm, respectively. The inner sphere was filled with Material 1. The space between the two spheres was filled with Material 2. The outside of the outer sphere was set to vacuum. The nuclide compositions of the materials are listed in Table 5. The temperature of each material was set to 300 K.

The calculation parameters are listed in Table 6, and the results are presented in Table 7. The deviation in k_{eff} is very small. There was a slight increase in the calculation time. The accuracy of the HWN method was therefore confirmed.

3.2.2 PWR assembly

The second comparison example is the PWR assembly model shown in Fig. 6. The model comprised an infinite cylinder with a cross section of 21.42 cm $\times$ 21.42 cm. There were 264 fuel rods and 25 pipes arranged in a 17 $\times$ 17 square. The fuel rods were cylinders with diameters of 0.8192 cm. Each fuel rod was surrounded by a 0.082 mm air layer and a 0.572 mm zirconium wall. The

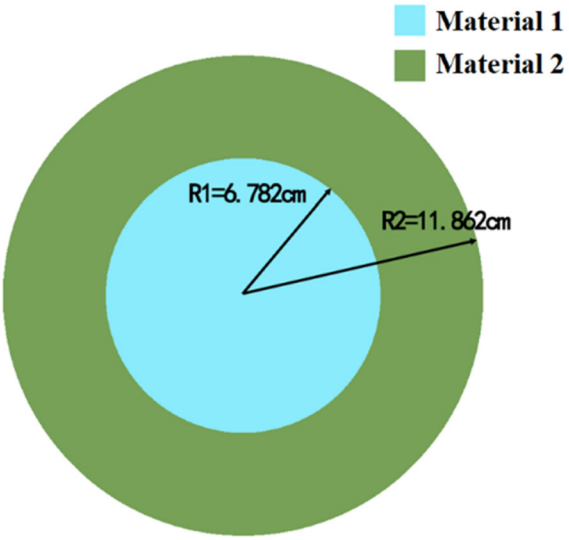


Fig. 5 (Color online) Concentric spheres example

Table 5 Nuclide composition of materials in concentric spheres example

Materials	Nuclides	Atomic density (10^{24}atoms $\cdot$ cm^{-3})
Material 1	U-235	4.4917×10^{-2}
	U-238	2.5993×10^{-3}
	U-234	4.9210×10^{-4}
Material 2	U-235	3.4428×10^{-4}
	U-238	4.7470×10^{-2}
	U-234	2.6299×10^{-6}

Table 6 Calculation parameters for the example of concentric spheres

Parameters	Value
Neutrons per cycle	100,000
Inactive cycles	400
Active cycles	1600

Table 4 Theoretical memory requirements of HWN and ACE data

Data saving mode for cross section	Theoretical memory requirement (MB)	Optimization ratio (%)
0 K ACE data in resolved resonance energy range	7.38	–
HWN	2.50	66.1
0 K ACE data in the whole range	7.40	–
HWN and 0 K ACE data outside resolved resonance range	2.52	65.9

Table 7 Calculation results of the concentric spheres example

Case	k_{eff}	Standard deviation	Calculation time (min)	Time ratio
ACE	0.995747	0.000048	597.3738	1.00
HWN	0.995748	0.000048	601.7461	1.01

inner and outer diameters of the pipes were 1.138 and 1.2294 cm, respectively. The pipes were filled with water, and the material of their walls was zirconium. The remainder of the assembly was filled with water. The nuclide compositions of the fuels are listed in Table 8. The temperature of all the materials was set to 700 K.

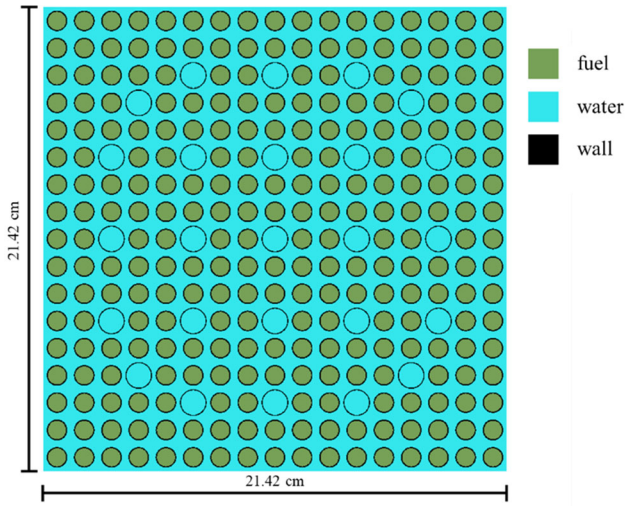


Fig. 6 (Color online) Schematic diagram of the PWR assembly

Table 8 Nuclide composition of fuel in the PWR assembly

Nuclide	Atomic density (10^{24} atoms/cm^{-3})
U-235	6.9100×10^{-3}
U-238	2.2062×10^{-1}
O-16	4.5510×10^{-1}

Table 9 Calculation parameters of PWR assembly

Parameters	Value
Neutrons per cycle	40,000
Inactive cycles	400
Active cycles	1600

The calculation parameters are listed in Table 9, and the results are presented in Table 10. The difference in k_{eff} between the two cases is within twice the standard deviation, indicating that there is no significant difference. The calculation time of the HWN case is 1.39 times that of the ACE case, which indicates that using the HWN method will prolong the calculation time. The calculation time is shorter if the method is not used in all the windows.

The neutron flux spectrum of the fuel in a fuel rod adjacent to the central pipe and the neutron flux inside each fuel rod and pipe were calculated for both the ACE and HWN cases. The results are presented in Fig. 7. The blocks in Fig. 7c do not represent the actual geometry, rather they the corresponding positions.

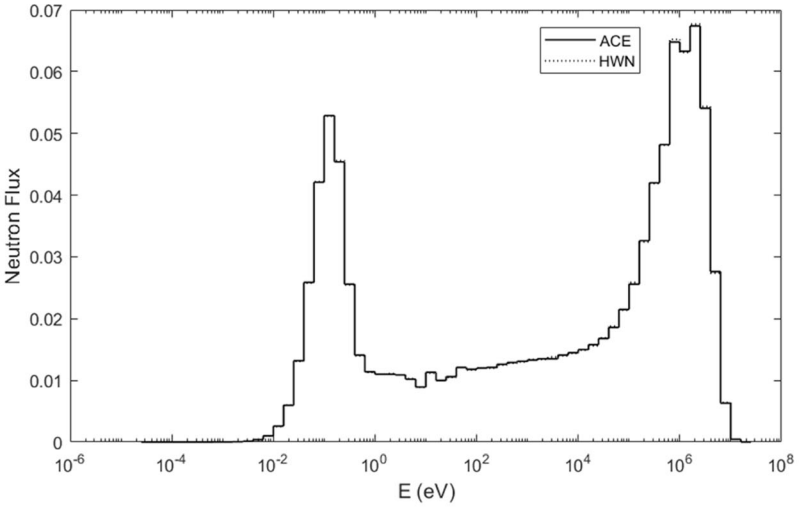
Figure 7a shows that the neutron flux spectra of the fuel rods are the same. Figure 7b shows that the relative deviations of the fluxes in most statistical intervals are within three times the standard deviation of the ACE case. Figure 7c compares the neutron fluxes of all the fuel rods in the assembly. There is no significant difference between the ACE and HWN cases. The blank grid squares represent the pipes whose fluxes are not shown in this figure. A numerical comparison shows that the deviation of most fuel rods and pipes is within twice the standard deviation, and the deviation of the remaining fuel rods and pipes is within three times the standard deviation except for a few fuel rods and pipes. The comparison results therefore demonstrate the accuracy of the proposed HWN method.

4 Conclusion

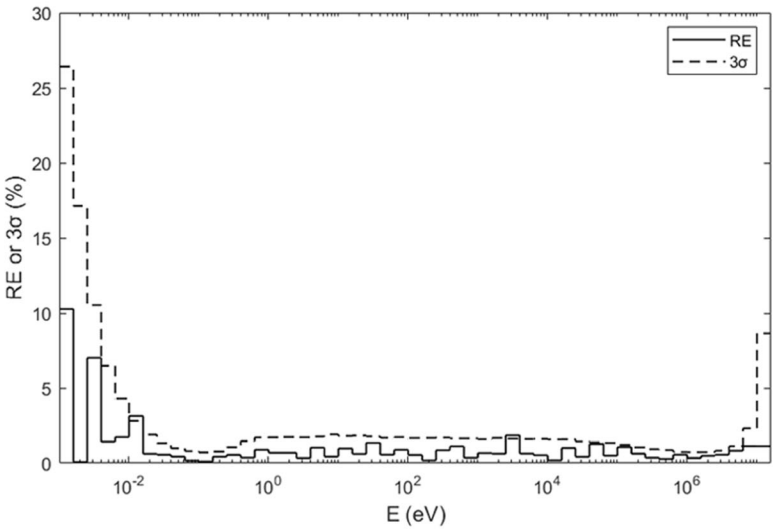
In this study, a hybrid windowed networks method for on-the-fly Doppler broadening was proposed and implemented in the RMC code. The resolved resonance energy range is divided into energy windows. In each window, two BP networks are trained to calculate the cross section at the base temperature and broaden the cross section to any temperature within the range of 250–1600 K. The

Table 10 Calculation results of PWR assembly

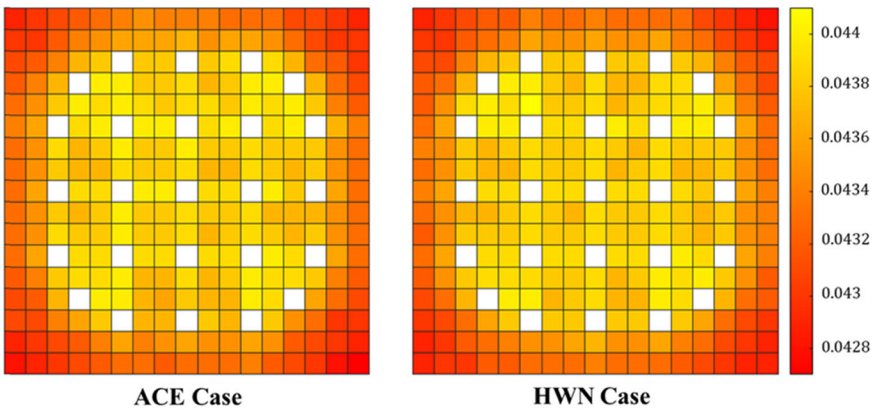
Case	k_{eff}	Standard deviation	Calculation time (min)	Time ratio
ACE	1.349355	0.000068	2174.0543	1.00
HWN	1.349255	0.000062	2961.4330	1.39



(a)



(b)



(c)

Fig. 7 (Color online) Comparison results of ACE and HWN cases. **a** Flux spectrum of fuel rod beside central pipe **b** Relative error and standard deviation of the flux spectrum **c** Flux of fuel rods

structures of the neural networks and training parameters are determined through calculations. Networks for the total cross section, absorption cross section, and elastic scattering cross section of U-235 were trained.

Microscopic cross-sectional comparison and macroscopic tests were performed to verify the utility and effectiveness of the HWN method. A comparison between the cross sections evaluated by this method and the ACE data shows the high accuracy of the proposed method. Macroscopic tests were conducted using RMC to verify the accuracy and efficiency of the method. The calculation time ratio between the HWN method and the ACE data for the PWR assembly calculation was 1.39. If the method is used with all the windows of the resolved resonance energy range, the theoretical memory consumption for U-235 nuclide can be reduced to 33.9% of the memory needed for ACE cross-sectional interpolation at 0 K. The theoretical memory consumption is reduced to 34.1% of the ACE data at 0 K if the ACE data outside the resolved resonance energy range are also included.

The HWN can be combined with other on-the-fly Doppler broadening methods or linear interpolation with point-wise nuclear data. Using the HWN method, users can compromise efficiency according to the memory saving requirement. If the predicted neutron flux is high in some windows, the use of this method in the remaining windows can significantly reduce the memory cost without comprising efficiency to a great extent.

The method proposed in this study should be further studied to improve its effectiveness. The calculation speed may be greatly improved by optimizing the calculation process, particularly the evaluation of the nonlinear transfer function. The accuracy of this method can be further improved by extending the training time or by choosing more suitable training parameters.

The HWN method is applicable to any heavy nuclide with a resolved resonance energy range. Because the training is performed with point-wise data, the method can also be applied to nuclides for which the windowed multipole method is inapplicable. The method can be applied to more nuclides, especially those that are important in reactor simulations or those for which it is difficult to apply the windowed multipole method.

The potential of the neural networks used in the HWN method for reducing the memory usage in the evaluation of complex parameters was demonstrated. The introduction of neural networks into other developed on-the-fly Doppler broadening methods may result in greater memory savings and higher accuracy.

Author contributions All authors contributed to the study conception and design. Data collection and analysis were performed by Tian-Yi Huang and Ze-Guang Li. The programming and tests are strongly supported by Kan Wang, Xiao-Yu Guo and Jin-Gang Liang. The first draft of the manuscript was written by Tian-Yi Huang and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

1. B. Forget, S. Xu, K. Smith, Direct Doppler broadening in Monte Carlo simulations using the multipole representation. Ann. Nucl. Energy **64**, 78–85 (2014). https://doi.org/10.1016/j.anucene.2013.09.043
2. K. Wang, S. Liu, Z. Li et al., Analysis of BEAVRS two-cycle benchmark using RMC based on full core detailed model. Prog. Nucl. Energ. **98**, 301–312 (2017). https://doi.org/10.1016/j.pnucene.2017.04.009
3. T. Viitanen, J. Leppänen, Effect of the target motion sampling temperature treatment method on the statistics and performance. Ann. Nucl. Energy **82**, 217–225 (2015). https://doi.org/10.1016/j.anucene.2014.08.033
4. J. Leppänen, M. Pusa, T. Viitanen et al., The serpent Monte Carlo code: status, development and applications in 2013. Ann. Nucl. Energy **82**, 142–150 (2015). https://doi.org/10.1016/j.anucene.2014.08.024
5. C. Josey, B. Forget, K. Smith. Efficiency and accuracy evaluation of the windowed multipole direct Doppler baradening method.in PHYSOR 2014, The Westin Miyako, Kyoto, Japan (2014).
6. G. Yesilyurt, W.R. Martin, F.B. Brown et al., On-the-fly Doppler broadening for Monte Carlo Codes. Nucl. Sci. Eng. **171**, 239–257 (2012). https://doi.org/10.13182/NSE11-67
7. J.A. Walsh, B. Forget, K.S. Smith et al., On-the-fly Doppler broadening of unresolved resonance region cross sections. Prog. Nucl. Energy **101**, 444–460 (2017). https://doi.org/10.1016/j.pnucene.2017.05.032
8. A.T. Pavlou, W. Ji, On-the-fly sampling of temperature-dependent thermal neutron scattering data for Monte Carlo simulations. Ann. Nucl. Energy **71**, 411–426 (2014). https://doi.org/10.1016/j.anucene.2014.04.028
9. S. Liu, Y. Yuan, J. Yu et al., Development of on-the-fly temperature-dependent cross sections treatment in RMC code. Ann. Nucl. Energy **94**, 144–149 (2016). https://doi.org/10.1016/j.anucene.2016.02.026
10. S. Liu, X. Peng, C. Josey et al., Generation of the windowed multipole resonance data using vector fitting technique. Ann. Nucl. Energy **112**, 30–41 (2018). https://doi.org/10.1016/j.anucene.2017.09.042
11. X. Liu, L. Deng, Z. Hu et al., Study of on-the-fly Doppler broadening in JMCT program. Acta Phys. Sin. **65**, 42–47 (2016). https://doi.org/10.7498/aps.65.092501
12. E.R. Caianiello, Mathematical theory of neural networks. IFAC Proc. Vol. **2**, 395–397 (1968). https://doi.org/10.1016/S1474-6670(17)68880-3
13. F. Cao, K. Yao, J. Liang, Deconvolutional neural network for image super-resolution. Neural Netw. **132**, 394–404 (2020). https://doi.org/10.1016/j.neunet.2020.09.017
14. H. Heo, B. So, I. Yang et al., Automated recovery of damaged audio files using deep neural networks. Digit. Invest. **30**, 117–126 (2019). https://doi.org/10.1016/j.diin.2019.07.007
15. Y. Li, J. Li, J. Huang et al., Fitting analysis and research of measured data of SAW micro-pressure sensor based on BP neural network. Measurement **155**, 107533 (2020). https://doi.org/10.1016/j.measurement.2020.107533

16. S. Zare, M. Ayati, Simultaneous fault diagnosis of wind turbine using multichannel convolutional neural networks. ISA Trans. **108**, 230–239 (2021). https://doi.org/10.1016/j.isatra.2020.08.021
17. R. Pahič, B. Ridge, A. Gams et al., Training of deep neural networks for the generation of dynamic movement primitives. Neural Netw. **127**, 121–131 (2020). https://doi.org/10.1016/j.neunet.2020.04.010
18. D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors. Nature **323**, 533–536 (1986). https://doi.org/10.1038/323533a0
19. P.S. Kumar, H.S. Behera, K.K. Anisha et al., Advancement from neural networks to deep learning in software effort estimation: perspective of two decades. Comp. Sci. Rev. **38**, 100288 (2020). https://doi.org/10.1016/j.cosrev.2020.100288
20. K. Wang, Z. Li, D. She et al., RMC – A Monte Carlo code for reactor core analysis. Ann. Nucl. Energy **82**, 121–129 (2015). https://doi.org/10.1016/j.anucene.2014.08.048
21. R.E. MacFarlane, A.C. Kahler, Methods for processing ENDF/B-VII with NJOY. Nucl. Data Sheets **111**, 2739–2890 (2010). https://doi.org/10.1016/j.nds.2010.11.001