



Estimation of trapezoidal-shaped overlapping nuclear pulse parameters based on a deep learning CNN-LSTM model

Xing-Ke Ma,^a Hong-Quan Huang,^{a*} Xiao Ji,^a He-Ye Dai,^b Jun-Hong Wu,^a Jing Zhao,^a Fei Yang,^a Lin Tang,^c Kai-Ming Jiang,^a Wei-Cheng Ding^a and Wei Zhou^a

Received 5 January 2020

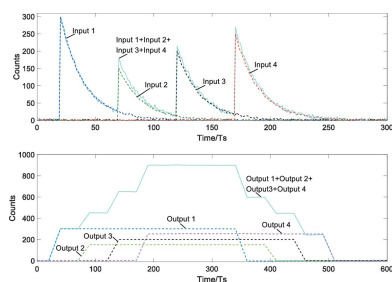
Accepted 31 March 2021

Edited by A. Stevenson, Australian Synchrotron, Australia

Keywords: nuclear pulse; trapezoidal shaping; deep learning; CNN-LSTM.

^aCollege of Nuclear Technology and Automation Engineering, Chengdu University of Technology, Dongsanlu, Erxianqiao, Chengdu 610059, People's Republic of China, ^bCollege of Foreign Language, Shanghai Maritime University, Haigang Avenue, Shanghai 201306, People's Republic of China, and ^cCollege of Electronic Information and Electrical Engineering, Chengdu University, No. 1 Shiling Street, Longquanyi District, Chengdu 610106, People's Republic of China. *Correspondence e-mail: huanghongquan@cdu.cn

The Long Short-Term Memory neural network (LSTM) has excellent learning ability for the time series of the nuclear pulse signal. It can accurately estimate the parameters (such as amplitude, time constant, *etc.*) of the digitally shaped nuclear pulse signal (especially the overlapping pulse signal). However, due to the large number of pulse sequences, the direct use of these sequences as samples to train the LSTM increases the complexity of the network, resulting in a lower training efficiency of the model. The convolution neural network (CNN) can effectively extract the sequence samples by using its unique convolution kernel structure, thus greatly reducing the number of sequence samples. Therefore, the CNN-LSTM deep neural network is used to estimate the parameters of overlapping pulse signals after digital trapezoidal shaping of exponential signals. Firstly, the estimation of the trapezoidal overlapping nuclear pulse is considered to be obtained after the superposition of multiple exponential nuclear pulses followed by trapezoidal shaping. Then, a data set containing multiple samples is set up; each sample is composed of the sequence of sampling values of the trapezoidal overlapping nuclear pulse and the set of shaping parameters of the exponential pulse before digital shaping. Secondly, the CNN is used to extract the abstract features of the training set in these samples, and then these abstract features are applied to the training of the LSTM model. In the training process, the pulse parameter set estimated by the present neural network is calculated by forward propagation. Thirdly, the loss function is used to calculate the loss value between the estimated pulse parameter set and the actual pulse parameter set. Finally, a gradient-based optimization algorithm is applied to update the weight by getting back the loss value together with the gradient of the loss function to the network, so as to realize the purpose of training the network. After model training was completed, the sampled values of the trapezoidal overlapping nuclear pulse were used as input to the CNN-LSTM model to obtain the required parameter set from the output of the CNN-LSTM model. The experimental results show that this method can effectively overcome the shortcomings of local convergence of traditional methods and greatly save the time of model training. At the same time, it can accurately estimate multiple trapezoidal overlapping pulses due to the wide width of the flat top, thus realizing the optimal estimation of nuclear pulse parameters in a global sense, which is a good pulse parameter estimation method.



© 2021 International Union of Crystallography

1. Introduction

Digital shaping methods are important in shaping nuclear pulse signals because some digital signal processing can be used to estimate the parameters of the nuclear signal, and greatly improve the performance of the nuclear instrument.

However, the overlap of adjacent nuclear pulses at high speed counting is difficult to avoid regardless of the shaping method. Therefore, the parameter estimation of overlapping nuclear pulses after digital shaping is still a problem (Huang *et al.*, 2017; Jiang *et al.*, 2017). For example, the exponential nuclear pulse in the trapezoidal shaping method is shaped into a trapezoidal nuclear pulse; thus, the signal is broadened to facilitate the estimation of the amplitude; however, the probability of the trapezoidal pulse overlapping is larger than that of the exponential nuclear pulse (Chen, 2009; Ren *et al.*, 2018). Many research institutions have conducted in-depth research on the formation, acquisition, identification, and parameter estimation of trapezoidal nuclear pulses in recent years (Xie, 2009; Zhang, 2006; Zhou *et al.*, 2015). However, the parameter estimation for heavily overlapping trapezoidal nuclear pulses is unsatisfactory (Xiao *et al.*, 2005; Zhou *et al.*, 2007; Tang *et al.*, 2018; Hong *et al.*, 2018). Deep learning technology is a popular intelligent science technology (Hinton & Salakhutdinov, 2006). It has hidden layers that contain many nonlinear transformation structures. Thus, its ability in fitting complex models by training a large amount of data is enhanced (LeCun *et al.*, 1998, 2015; Dorffner, 1996; Du *et al.*, 2018). At present, research related to introducing deep learning technology into nuclear pulse parameter estimation is still at the preliminary stage.

The Long Short-Term Memory neural network (LSTM) can be used to estimate the parameters of the nuclear pulse signal (Graves *et al.*, 2009, 2013; Pascanu *et al.*, 2013; Graves, 2013), because the nuclear pulse signal has the characteristics of a time series after discretization, and the LSTM (Hochreiter & Schmidhuber, 1997; Gers *et al.*, 2001) with cyclic structure is extremely effective in dealing with time series problems.

Unfortunately, directly using a nuclear pulse sequence as the input data of the LSTM network increases the complexity of the model because of its complex and diverse characteristics. Accordingly, the efficiency of the training model decreases. Therefore, a convolutional neural network (CNN) (Krizhevsky *et al.*, 2017) can be used to extract abstract features from the pulse sequence and then input these abstract features into the LSTM network. In this study, a parameter estimation method based on the deep learning CNN-LSTM model is proposed for the overlapped nuclear pulses shaped by several exponential decay nuclear pulses. This method is important in verifying the shaping algorithm and acquiring nuclear pulse parameters.

2. Principle and algorithm

2.1. Principle of trapezoidal shaping for exponential pulse

For the overlapping pulses obtained by the superposition of N exponentially decaying nuclear pulses, the mathematical model is as follows,

$$V_c(t) = \sum_{i=1}^N u(t - T_i) A_i \exp[(t - T_i)/\tau] + v(t), \quad (1)$$

where $u(t)$ represents the step signal, A_i is the amplitude coefficient of the i th nuclear pulse, T_i represents the occurrence time of the i th nuclear pulse, τ is the time constant, and $v(t)$ represents the noise in the detection process. Discretization is performed by the sampling period T_s , and the discretized exponential pulse is

$$V_c(kT_s) = \sum_{i=1}^N u(kT_s - T_i) A_i \exp[-(kT_s - T_i)/\tau]. \quad (2)$$

The original trapezoidal overlapping nuclear pulse sequence $V_o(mT_s)$ for parameter estimation is regarded as the result of the trapezoidal shaping of N exponential decay nuclear pulse stacking sequences $V_c(kT_s)$. Its mathematical model is as follows,

$$\begin{aligned} V_o(mT_s) = & 2V_o[(m-1)T_s] - V_o[(m-2)T_s] \\ & + \frac{1}{n_a} \left(V_c[(m-1)T_s] - V_c[(m-n_a-1)T_s] \right. \\ & - V_c[(m-n_b-1)T_s] + V_c[(m-n_c-1)T_s] \\ & - \exp(-T_s/\tau) \left\{ V_c[(m-2)T_s] \right. \\ & - V_c[(m-n_a-2)T_s] - V_c[(m-n_b-2)T_s] \\ & \left. \left. + V_c[(m-n_c-2)T_s] \right\} \right). \end{aligned} \quad (3)$$

In equations (2) and (3), $u(\dots)$ represents a step function; $k = 1, 2, 3, \dots, K$; K is the number of discrete points of $V_c(kT_s)$; τ is the decay time constant of the exponential pulse; T_s is the sampling period; A_i and T_i represent the amplitude and occurrence time of the i th exponentially decaying nuclear pulse, respectively; $n_a = t_a/T_s$; $n_b = (t_a + D)/T_s$; $n_c = t_c/T_s$; t_a is the rise time of the trapezoidal pulse; D is the flat top width of the trapezoidal pulse; $t_c = 2t_a + D$ represents the entire trapezoidal shaping time; and $m = 1, 2, 3, \dots, K + 2 + n_c$.

2.2. Parameter estimation of overlapping pulses

For the parameter estimation problem after N exponential decay nuclear pulse trapezoidal shapings, the steps to solve the problem mainly include the generation of data sets: the forward propagation estimate pulse parameter set, the back propagation update network weight, and the preservation after the completion of the model training.

2.2.1. Production of the data set. A data set with n samples is considered. The matrix representation of the data set is as follows,

$$\begin{bmatrix} [V_o(T_s)]_1 & [V_o(2T_s)]_1 & \cdots & [V_o((K+2+n_c)T_s)]_1 & \theta_1 \\ [V_o(T_s)]_2 & [V_o(2T_s)]_2 & \cdots & [V_o((K+2+n_c)T_s)]_2 & \theta_2 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ [V_o(T_s)]_n & [V_o(2T_s)]_n & \cdots & [V_o((K+2+n_c)T_s)]_n & \theta_n \end{bmatrix}. \quad (4)$$

Each row in matrix (4) represents data of one sample. n_c represents the entire trapezoidal shaping time (t_c) divided by the sampling period (T_s), $n_c = t_c/T_s$. The front $K + 2 + n_c$ data

of each row are the sampling value of the trapezoidal overlapping nuclear pulse. The trapezoidal overlapping nuclear pulse is assumed to be shaped according to the shaping method in Section 2.1. The parameters of the input signal $V_c(kT_s)$ before shaping are A_i ($i = 1, 2, \dots, N$), T_i ($i = 1, 2, \dots, N$) and τ . The rising edge time at the time of trapezoidal shaping is t_a , and the flat top width time is D . These parameters constitute the parameter set θ of this sample, that is, $\theta = [A_1, A_2, \dots, A_N, T_1, T_2, \dots, T_N, \tau, t_a, D]$; for example, the sampling values of the trapezoidal overlapping nuclear pulse $V_o(mT_s)$ corresponding to the i th sample are $[V_o(T_s)]_i, [V_o(2T_s)]_i, [V_o(3T_s)]_i, \dots, [V_o((K + 2 + n_c)T_s)]_i$; the parameter set of the i th sample becomes θ_i . The parameter set θ is randomly generated.

The data set is divided into a training set, a test set and a validation set according to a certain proportion. Among them, the training set is used to train the CNN-LSTM model, and the test set is utilized to test the generalization ability of the model after the model training is completed. The validation set is adopted to test whether the trained model has an overfitting phenomenon. If an overfitting phenomenon occurs, then the Dropout (Srivastava *et al.*, 2014; Bouthillier *et al.*, 2015) algorithm is used to modify the propagation structure of the neural network. The Dropout algorithm can discard the memory unit from the network according to a certain probability during the training process. The memory cells of the network are temporarily discarded randomly. Thus, a difference in the network of each batch training is observed. Such differences increase the generalization ability of the model. Accordingly, the occurrence of overfitting is effectively suppressed. Its mathematical model is as follows,

$$r_j^{(l)} \simeq \text{Bernoulli}(p), \quad (5)$$

$$\tilde{y}^{(l+1)} = r^{(l)} y^{(l)}, \quad (6)$$

where p is the probability that the CNN-LSTM memory cell stops propagating. $r_j^{(l)}$ is the retention probability of state information for the j th LSTM memory cell of the layer l network, which obeys the Bernoulli distribution. $y^{(l)}$ is the output information of the layer l network. $\tilde{y}^{(l+1)}$ is the input information of the layer $l + 1$ network.

2.2.2. Forward propagation. In training the CNN-LSTM forward propagation, the sampling values of the trapezoidal overlapping nuclear pulse $V_o(mT_s)$ corresponding to each sample in the training set is used as the input data of the CNN-LSTM model. We use CNN to extract abstract features of each sample, including convolution and pooling processes, for fully learning the training data by using the network. The abstract feature of the sample extracted by CNN is utilized as the input of LSTM using the structure of the forgetting gate, input gate, memory cell state and output gate of LSTM to sequentially iterate out the hidden state information h_m and the state information C_m of the memory cell. The hidden state information h_m is not only transferred to the next LSTM memory cell of the same layer with the state information C_m of the memory cell but also serves as the input information of the

LSTM network of the next layer. This way of information transfer function enables the CNN-LSTM model to map abstract features of data to higher dimensional network layers. The convolution process, pooling process, forgetting gate, input gate, memory cell state and output gate of CNN-LSTM forward propagation are realized according to steps (A), (B), (C), (D), (E) and (F) as follows:

(A) *Calculation of the convolution process.* The convolution process in CNN is mainly implemented by the convolutional layer, and the abstract features of the input data are extracted by multiple internal convolution kernels. For the i th sample, the input and output sequences of the CI convolution layer are assumed to be $[V_o^{CI-1}(mT_s)]_i$ and $[V_o^{CI}(mT_s)]_i$, respectively, and its mathematical model is expressed as

$$\begin{aligned} [V_o^{CI}(mT_s)]_i &= f_a \left\{ [(V_o^{CI-1} \otimes \omega^{CI})(mT_s)]_i \right\} \\ &= f_a \left\{ \sum_x^{C_s} [V_o^{CI-1}(s_0 mT_s + x) \omega^{CI}(x)]_i + b_i^{CI} \right\}, \quad (7) \end{aligned}$$

where $\omega^{CI}(x)$ is the convolution kernel, C_s is the convolution kernel size, and s_0 is the convolution step size. f_a is an activation function. Suppose that the total number of input sequences and output sequences of the convolution layer are M^{CI} and M^{CI+1} ; the calculation formula is as follows,

$$M^{CI} = \frac{M^{CI-1} + 2p - C_s}{s_0} + 1, \quad (8)$$

where p is the padding number, and $M^{CI-1} = K + 2 + n_c$ when $CI = 1$.

(B) *Pooling process.* After feature extraction in the convolution layer, the output feature image is transferred to the pooling layer for feature selection and information filtering. The pooling layer contains a pooling function, which replaces the result of a single point in the feature image with the feature image statistic of its neighboring area. The steps of selecting the pooling area in the pooling layer are the same as those of the convolution kernel scanning feature image, which are controlled by pool size, step size, and filling.

After the multi-layer CNN extracts abstract features from the original sample sequence, the complexity of the sequence $V_o(mT_s)$ is greatly decreased. At this time, the sequence is input into the LSTM network for further processing. We assume that the total number of sequences output by the last layer of CNN is $M^{CI} = M$. Accordingly, the number of sequences input by the LSTM first layer is $M^{L1} = M$, and we obtain

$$[V_o(mT_s)]_i = [V_o^{CI}(mT_s)]_i. \quad (9)$$

(C) *Calculation of the forgetting gate structure.* The forgetting gate structure can determine the degree of information discarding of the state of the memory cell,

$$f_m = \sigma \left(\sum_{m=1}^{M^{L1}} U_{i,m}^f [V_o(mT_s)]_i + \sum_{m=2}^{M^{L1}} W_{i,m}^f h_{m-1} + b_i^f \right), \quad (10)$$

where h_{m-1} is the hidden state information of the previous memory cell; $U_{i,m}^f$ and $W_{i,m}^f$ are the input and cyclic weights of

the m th sample value $[V_o(mT_s)]_i$ in the forgotten gate structure in the i th sample, respectively; b_i^f is the bias of the i th sample in the forgetting gate structure; and σ is the gate function, which is composed of the Sigmoid function. With this function, a value between 0 and 1 can be output to determine the retention probability of the state information. The formula is

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \quad (11)$$

(D) *Calculation of the input gate structure.* The input gate structure is used to calculate the newly increased state information in the interior of the memory cell. Its structure is like that of the forgetting gate. The parameters of weight and bias are U_i^g , W_i^g , and b_i^g . The mathematical model is as follows,

$$g_m = \sigma \left(\sum_{m=1}^{M^L} U_{i,m}^g [V_o(mT_s)]_i + \sum_{m=2}^{M^L} W_{i,m-1}^g h_{m-1} + b_i^g \right), \quad (12)$$

where $U_{i,m}^g$ and $W_{i,m}^g$ are the input and cyclic weights of the m th sampling value $[V_o(mT_s)]_i$ in the i th sample in the input gate structure, respectively; and b_i^g is the bias of the i th sample in the input gate structure.

(E) *Updating of memory cell status.* The candidate information vector \tilde{C}_m is created using the tanh function. The forgetting gate information, the state information of the previous memory cell, the input gate information, and the candidate information vector are regarded as the update elements of the state information of the current memory cell. The mathematical model for updating the state information is

$$\tilde{C}_m = \tanh \left(\sum_{m=1}^{M^L} U_{i,m}^C [V_o(mT_s)]_i + \sum_{m=2}^{M^L} W_{i,m-1}^C h_{m-1} + b_i^C \right), \quad (13)$$

$$\tanh(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)}, \quad (14)$$

$$C_m = f_m C_{m-1} + g_m \tilde{C}_m, \quad (15)$$

where C_m represents the state value of the memory cell at the current time; f_m represents the output value of the forgetting gate; C_{m-1} represents the state value of the memory cell at the previous time; g_m represents the output value of the input gate; \tilde{C}_m represents the candidate vector; $U_{i,m}^C$ and $W_{i,m}^C$ are the input and cyclic weights of the m th sampling value $[V_o(mT_s)]_i$ in the i th sample in the state update structure of the memory cell, respectively; and b_i^C is the bias of the i th sample in the state update structure of the memory cell.

(F) *Calculation of the output gate.* The output gate determines the hidden state information h_m . First, the vector containing the hidden state information h_{m-1} of the previous memory cell and the current pulse sequence information $[V_o(mT_s)]_i$ are transferred to the Sigmoid function. Then, the state information C_m of the memory cell is transferred to the tanh function. Finally, the output of the tanh function is multiplied by the output o_m of the Sigmoid function to

determine the hidden state information h_m . The hidden state information h_m is transmitted to the next layer network, and the state information C_m of the memory cell and this h_m are also transmitted to the next memory cell in the same layer. The mathematical model of the output gate is as follows,

$$o_m = \sigma \left(\sum_{m=1}^{M^L} U_{i,m}^o [V_o(mT_s)]_i + \sum_{m=2}^{M^L} W_{i,m-1}^o h_{m-1} + b_i^o \right), \quad (16)$$

$$h_m = o_m \tanh(C_m), \quad (17)$$

where $U_{i,m}^o$ and $W_{i,m}^o$ are the input and cyclic weights of the m th sampling value $[V_o(mT_s)]_i$ in the i th sample in the output gate structure, and b_i^o is the bias of the i th sample in the output gate structure. The forward propagation ends when the last layer of the LSTM network has predicted the set θ'_i of pulse parameters.

2.2.3. Back propagation. The back propagation training of the trapezoid overlapping nuclear pulse is based on the back propagation through time (BPTT) algorithm (Werbos, 1990; Graves & Schmidhuber, 2005). The weights and biases of each LSTM memory cell are randomly assigned when defining the neural network. Thus, the error between the predicted pulse parameter set θ'_i outputted by the single forward propagation iteration and the actual pulse parameter set θ_i in the training set can be calculated by the loss function. For the training set with q samples, the mean square error value of the parameter set θ_i is taken as the function value Loss_{MSE} of the loss function, that is, the calculation formula of the loss function is

$$\text{Loss}_{\text{MSE}} = \frac{1}{q} \sum_{i=1}^q (\theta_i - \theta'_i)^2. \quad (18)$$

The BPTT algorithm is used to feed the Loss_{MSE} and the gradient of the loss function back to the network to update the weight for reducing the error in the subsequent iteration.

2.2.4. Saving and application of the training model. After a CNN-LSTM model that can estimate the parameter set θ of the trapezoidal overlapping nuclear pulse is trained, the important information, such as the structure, weight, training configuration, and optimizer status, of the trained model is saved as a hierarchical data format 5 (HDF5) file. The sampled value of the trapezoidal overlapping nuclear pulse to be estimated is taken as the input of the CNN-LSTM model, and the output of the CNN-LSTM model is used to obtain the desired estimated pulse parameter set θ .

3. Examples

According to Xiao *et al.* (2005), Chen (2009) and Ren *et al.* (2018), trapezoidal shaping is a simple, fast, and efficient pulse shaping method. Increasing the flat top width of the trapezoidal pulse can increase the number of samples to guarantee the weak effect of noise on the shaped signal. However, a wide flat top width indicates a high probability of pulse overlap. When the time interval T_i of the adjacent exponential pulses is small, the overlap of the pulses is serious. As a result, the difficulty in estimating the pulse parameters increases.

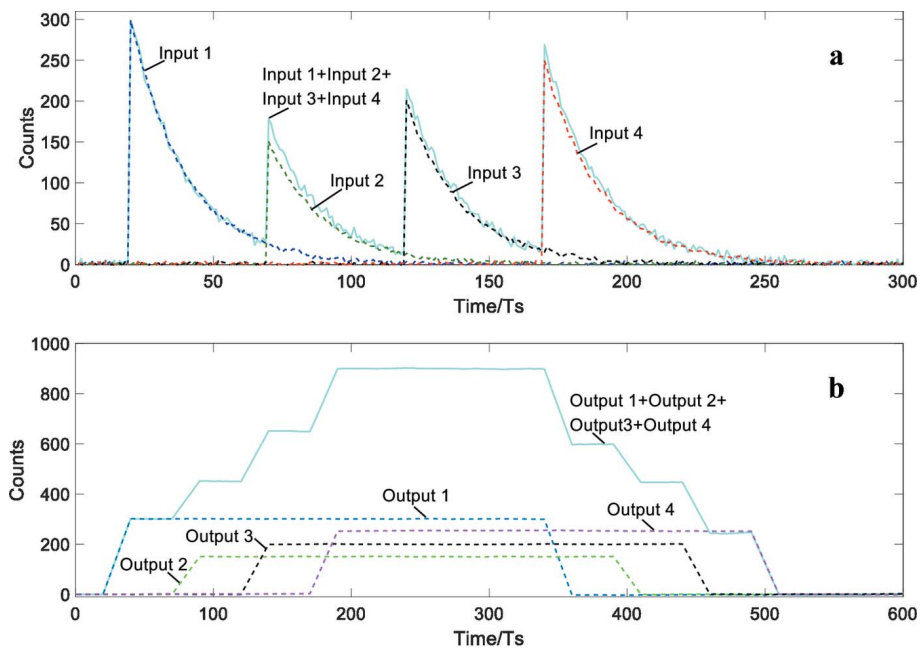


Figure 1 Comparison of four exponentially overlapping pulses before and after trapezoidal shaping. (a) The dashed line indicates the actual value before the exponential pulse shaping (Input x , $x = 1, 2, 3, 4$), and the solid line indicates the four exponential overlapping pulses (Input 1 + Input 2 + Input 3 + Input 4) before shaping. (b) The dashed line indicates the actual value of the exponential pulse after trapezoidal shaping (Output x , $x = 1, 2, 3, 4$), and the solid line indicates the four overlapping pulses after the trapezoidal shaping (Output 1 + Output 2 + Output 3 + Output 4).

Therefore, the examples in this paper explore the parameter estimation when the time interval of occurrence of multiple exponential pulses is relatively short and the trapezoidal shaping flat top width is relatively wide.

3.1. Example 1

Exponential pulses ‘Input 1’, ‘Input 2’, ‘Input 3’, and ‘Input 4’ were input into a trapezoidal shaping circuit with the characteristic time $\tau = 100$ ns. The values for the amplitude parameter, A_i , were 300, 150, 200, and 250 counts. The values for the time parameter, T_i , were 20, 70, 120, and 170 T_s , where T_s is the sampling period. The white noise standard deviation was 5 counts, and the sampling period, T_s , was 5 ns. The rise time of the trapezoidal shaping pulse, t_a , was 20 T_s . The flat top width of the trapezoidal pulse, D , was 300 T_s . The effect of shaping is shown in Fig. 1.

As shown in Fig. 1, trapezoidal shaping suppresses noise very well. However, the overlap of the shaped pulses is very serious due to the wide flat top width of the trapezoidal pulse. The time when the overlapping part of the pulse occurs is mainly concentrated in the front 600 T_s . Therefore, the number of input sequences of CNN-LSTM was set to 600 to save computing resources and improve training efficiency. A data set containing 10000 samples was considered, of which 72%, 8%, and 20% were considered for the training set, the validation set, and the test set, respectively. In the CNN model, two one-dimensional convolution layers and two pooling layers were set. The first convolutional layer had 60 convolution kernels, and the second convolutional layer had 10 convolution kernels. The convolution kernel size in both

convolutional layers was 3, the filling strategy was ‘same,’ the convolution kernel moving step size was 1, and the excitation function was ‘relu.’ The pooling layer used one-dimensional maximum pooling. In the LSTM model, four LSTM layers were set, and the Adam optimizer (Kingma & Ba, 2014) was used to set the learning rate to 0.0001 with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The number of training rounds (named epochs) was set to 16. In addition, groups of 10 samples were gathered into a batch of input networks (batch size = 10). An iterative graph of the loss value and accuracy in the training process are shown in Fig. 2. A structural diagram of the network model is shown in Fig. 3.

As shown in Fig. 2, the loss values of the algorithm on the training set and the validation set are both decreasing. Therefore, no over-fitting phenomenon occurs. In other words, this example does not need to add a Dropout layer. The

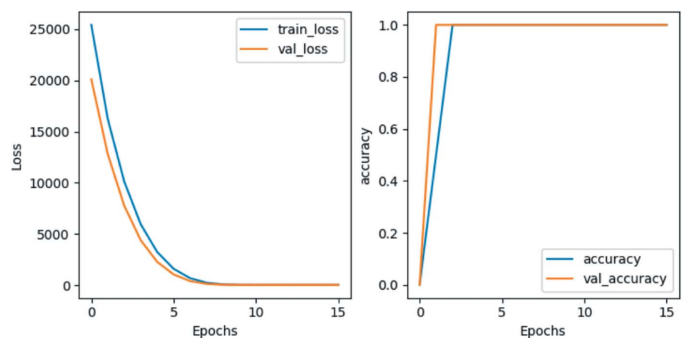


Figure 2 Iterative graphs of (left) the loss value and (right) the accuracy in the training process.

Table 1

Comparison of estimated and actual values of overlapping pulses in the CNN-LSTM model ($\tau = 100$ ns, $T_s = 5$ ns, $t_a = 20 T_s$, $D = 300 T_s$).

	A_1	A_2	A_3	A_4	T_1	T_2	T_3	T_4
Actual value	300	150	200	250	20	70	120	170
Estimated value of the CNN-LSTM model	299.83	149.68	200.21	249.79	20.19	70.25	120.26	170.08
Absolute error	0.17	0.32	0.21	0.21	0.19	0.25	0.26	0.08
Relative error (%)	0.06	0.21	0.11	0.08				

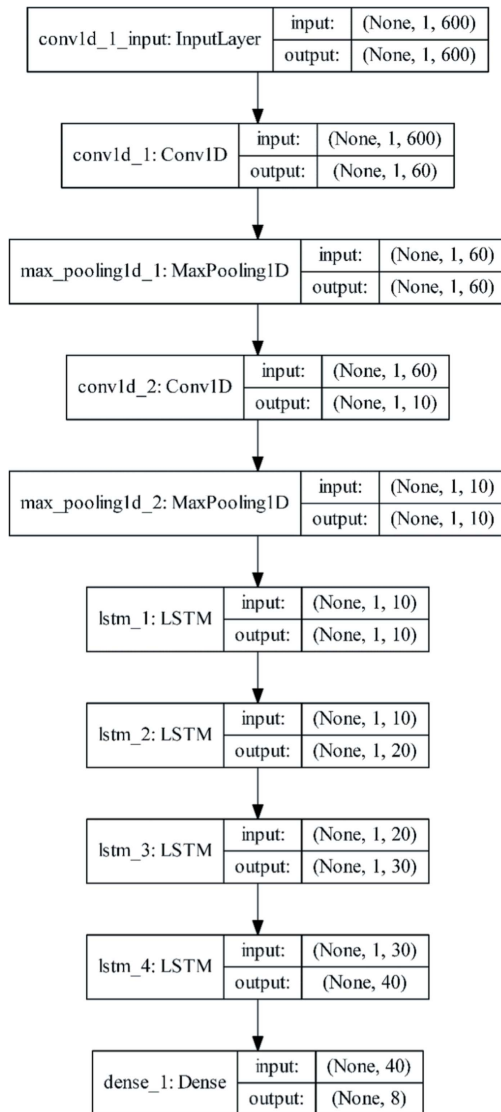


Figure 3
Structure diagram of the network model.

average accuracy of the trained model in the test set with 2000 samples was 100.00% (two decimal places were retained). The pulse parameters and errors estimated by the CNN-LSTM model (retaining two decimal places) are shown in Table 1. The estimation results of nuclear pulse parameters based on deep learning CNN-LSTM are shown in Fig. 4.

From the experimental results, the relative errors for amplitude parameter, A_i , were 0.06%, 0.21%, 0.11%, and

0.08%, and the absolute errors for time parameter, T_i , were 0.19, 0.25, 0.26 and 0.08. In a word, the trapezoidal overlapping nuclear pulse parameter estimation method based on the deep learning CNN-LSTM network is of great significance for improving the accuracy of radioactivity measurement, which partly solves the technical problem to accurately extract relevant information about the adjacent nuclear pulses due to the overlapping pulse signals. This method has great significance for improving the accuracy of radioactivity measurement.

3.2. Example 2

When the time interval (T_i) of the adjacent exponential nuclear pulses is short, the overlap between the pulses is serious. As a result, estimating the overlapping nuclear pulse parameters becomes very difficult. The purpose of this example was to verify the ability of the proposed method to estimate overlapping nuclear pulse parameters when the time interval between adjacent exponential nuclear pulses was small. Exponential pulses ‘Input 1’, ‘Input 2’, ‘Input 3’, and ‘Input 4’ were input into a trapezoidal shaping circuit with the characteristic time of $\tau = 100$ ns. The values for the amplitude parameter, A_i , were 300, 150, 200, and 250 counts. The values for the time parameter, T_i , were 20, 40, 60, and 80 T_s . The white noise standard deviation was 5 counts, and the sampling period, T_s , was 5 ns. The rise time of the trapezoidal shaping pulse, t_a , was 20 T_s . The flat top width of the trapezoidal pulse, D , was 300 T_s . The effect of shaping is shown in Fig. 5.

As shown in Fig. 5, the overlap of the pulses is very serious whether before or after shaping when the time interval between adjacent pulses is relatively short. This example used the same neural network model and training strategy as in Example 1 to verify the parameter estimation effect of the proposed method in the case of severe pulse overlap caused by short time interval. An iterative graph of the loss value and accuracy in the training process are shown in Fig. 6.

As shown in Fig. 6, the loss value of the algorithm on the training set and verification set is decreasing. This condition indicates that no over-fitting phenomenon occurs. Therefore, this example does not need to add the Dropout layer. The average accuracy of the trained model in the test set with 2000 samples was 100.00% (two decimal places were retained). The pulse parameters and errors estimated by the CNN-LSTM model (retaining two decimal places) are shown in Table 2. The estimation results are shown in Fig. 7.

According to the experimental results, the relative amplitude parameter (A_i) errors were 0.14%, 0.03%, 0.17% and 0.06%, and the absolute time parameter (T_i) errors were 0.10, 0.05, 0.12 and 0.26, respectively. Therefore, even

when the time interval between adjacent exponential pulses was short, the overlapping nuclear pulse shaping parameters estimated by the CNN-LSTM model still showed high precision.

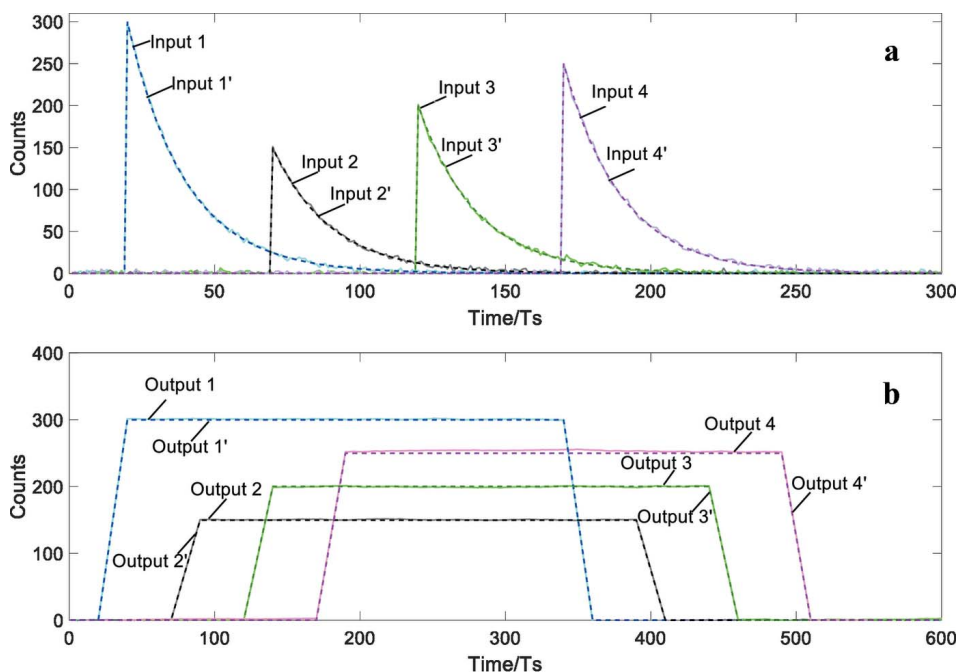


Figure 4 Estimation results of nuclear pulse parameters based on deep learning CNN-LSTM. (a) The solid line indicates the actual value of the exponential pulse (Input x , $x = 1, 2, 3, 4$), and the dashed line indicates the estimated value of the exponential pulse (Input x' , $x = 1, 2, 3, 4$). (b) The solid line indicates the actual value of the pulse after trapezoidal shaping (Output x , $x = 1, 2, 3, 4$), and the dashed line indicates the pulse estimation value after trapezoidal shaping (Output x' , $x = 1, 2, 3, 4$).

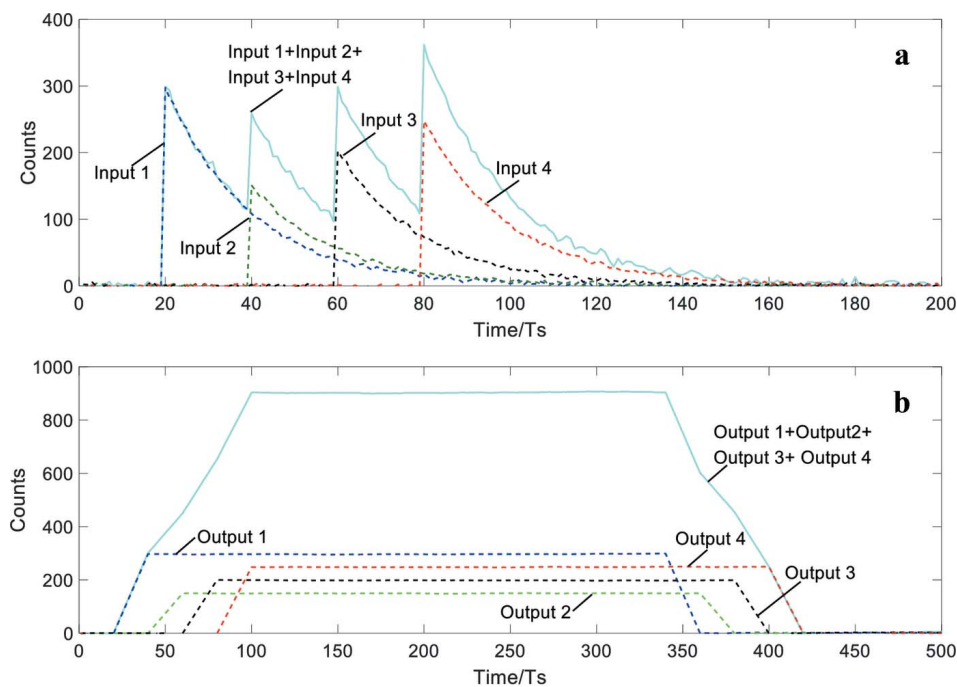


Figure 5 Comparison of four exponentially overlapping pulses before and after trapezoidal shaping. (a) The dashed line indicates the actual value before the exponential pulse shaping (Input x , $x = 1, 2, 3, 4$), and the solid line indicates the four exponential overlapping pulses (Input 1 + Input 2 + Input 3 + Input 4) before shaping. (b) The dashed line indicates the actual value of the exponential pulse after trapezoidal shaping (Output x , $x = 1, 2, 3, 4$), and the solid line indicates the four overlapping pulses after the trapezoidal shaping (Output 1 + Output 2 + Output 3 + Output 4).

Table 2

Comparison of estimated and actual values of overlapping pulses in the CNN-LSTM model ($\tau = 100$ ns, $T_s = 5$ ns, $t_d = 20 T_s$, $D = 300 T_s$).

	A_1	A_2	A_3	A_4	T_1	T_2	T_3	T_4
Actual value	300	150	200	250	20	40	60	80
Estimated value of the CNN-LSTM model	300.43	150.04	200.34	249.85	20.10	40.05	59.88	80.26
Absolute error	0.43	0.04	0.34	0.15	0.10	0.05	0.12	0.26
Relative error (%)	0.14	0.03	0.17	0.06				

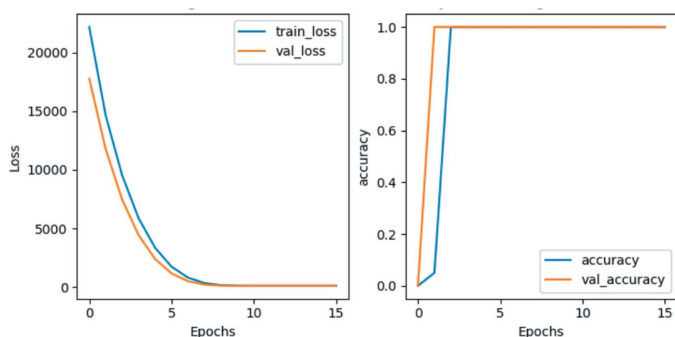


Figure 6
Iterative graphs of (left) the loss value and (right) the accuracy in the training process.

4. Conclusions

Traditional algorithms cannot extract signals from an entire sample due to the limitation in the mathematical model scale. The proposed method of parameter estimation based on deep learning technology for trapezoidal overlapping nuclear pulses obtained after trapezoidal shaping overcomes this limitation. The method uses the trapezoidal pulse sequence

and the shaping parameters of the exponential pulse as a sample set. The CNN-LSTM model is also allowed to establish a mapping relationship between each trapezoidal pulse sequence and its corresponding exponential pulse parameters through continuous training. The purpose of overlapping nuclear pulse parameter estimation is achieved. This method greatly reduces the rejection rate of the trapezoidal overlapping nuclear pulse and improves the accuracy and reliability of radioactivity measurement. The method is beneficial in analyzing the fluctuation of the signal parameter caused by the change in the response characteristics of the detector and its subsequent circuits, such as the fluctuation of the time constant of the exponential pulse signal. This method is important in verifying the nuclear instrument waveform shaping and energy spectrum drift correction algorithms, analyzing the relationship of parameters change with time and external conditions, and acquiring subsequent nuclear pulse parameters. At the same time, the trained model is saved in the HDF5 file format. Thus, other computer equipment can be used to complete the pulse parameter estimation by directly loading this model. With the increasing performance of portable devices with Android as the operating system, HDF5

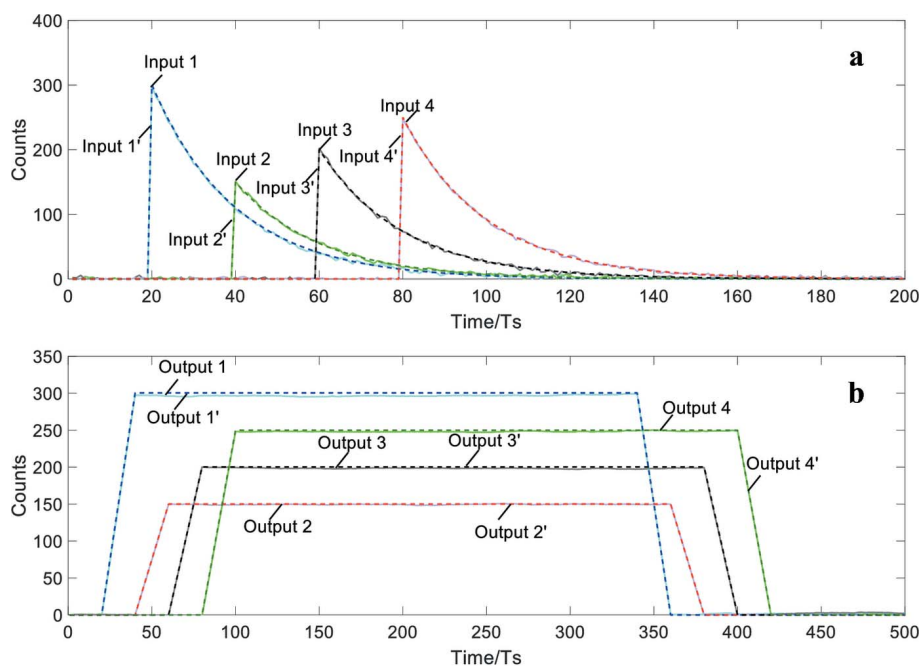


Figure 7
Estimation results of nuclear pulse parameters based on deep learning CNN-LSTM. (a) The solid line indicates the actual value of the exponential pulse (Input x , $x = 1, 2, 3, 4$), and the dashed line indicates the estimated value of the exponential pulse (Input x' , $x = 1, 2, 3, 4$). (b) The solid line indicates the actual value of the pulse after trapezoidal shaping (Output x , $x = 1, 2, 3, 4$), and the dashed line indicates the pulse estimation value after trapezoidal shaping (Output x' , $x = 1, 2, 3, 4$).

files can be deployed on these portable devices with simple modifications. This work provides support for the development of new-generation portable nuclear pulse recognition detectors.

Funding information

This work was supported by the National Natural Science Foundation of China (grants No.41774140 to Weicheng Ding and No.11675028 to Wei Zhou), the Scientific Research Fund of Sichuan Provincial Education Department (grant No. 18ZA0050 to Hongquan Huang), and the Sichuan Province Science and Technology Planning Project (grant No. 2021YJ0325 to Hongquan Huang).

References

- Bouthillier, X., Konda, K., Vincent, P. & Memisevic, R. (2015). *arXiv:1506.08700* [Stat. ML].
- Chen, L. (2009). PhD thesis, Tsinghua University, Beijing, China.
- Dorffner, G. (1996). *Neural Network World*, **6**, 447–468.
- Du, J., Hu, B. L., Liu, Y. Z., Wei, C. Y., Zhang, G. & Tang, X. J. (2018). *Spectrosc. Spectral Anal.* **38**, 1514–1519.
- Gers, F. A., Eck, D. & Schmidhuber, J. (2001). *Artificial Neural Networks – ICANN 2001*, International Conference Proceedings, Vol. 2130 of *Lecture Notes in Computer Science*, 21–25 August 2001, Vienna, Austria, edited by G. Dorffner, H. Bischof & K. Hornik, pp. 669–676.
- Graves, A. (2013). *arXiv:1308.0850* [cs. NE].
- Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H. & Schmidhuber, J. (2009). *IEEE Trans. Pattern Anal. Mach. Intell.* **31**, 855–868.
- Graves, A., Mohamed, A. & Hinton, G. (2013). *arXiv:1303.5778* [cs. NE].
- Graves, A. & Schmidhuber, J. (2005). *Neural Netw.* **18**, 602–610.
- Hinton, G. E. & Salakhutdinov, R. R. (2006). *Science*, **313**, 504–507.
- Hochreiter, S. & Schmidhuber, J. (1997). *Neural Comput.* **9**, 1735–1780.
- Hong, X., Zhou, J., Ni, S., Ma, Y., Yao, J., Zhou, W., Liu, Y. & Wang, M. (2018). *J. Synchrotron Rad.* **25**, 505–513.
- Huang, H. Q., Yang, X. F., Ding, W. C. & Fang, F. (2017). *Nucl. Sci. Tech.* **28**, 12.
- Jiang, K. M., Huang, H. Q., Yang, X. F. & Ren, J. F. (2017). *Nucl. Electron. Detect. Technol.* **37**, 121–124.
- Kingma, D. P. & Ba, J. (2014). *arXiv:1412.6980* [cs. LG].
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2017). *Commun. ACM*, **60**, 84–90.
- LeCun, Y., Bengio, Y. & Hinton, G. (2015). *Nature*, **521**, 436–444.
- Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). *Proc. IEEE*, **86**, 2278–2324.
- Pascanu, R., Gulcehre, C., Cho, K. & Bengio, Y. (2013). *arXiv:1312.6026* [cs. NE].
- Ren, Y. Q., He, J. F., Zhou, S. R., Ye, Z. X. & Yang, S. (2018). *Nucl. Electron. Detect. Technol.* **38**, 105–110.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). *J. Mach. Learn. Res.* **15**, 1929–1958.
- Tang, L., Zhou, J., Fang, F., Yu, J., Hong, X., Liao, X., Zhou, C. & Yu, S. (2018). *J. Synchrotron Rad.* **25**, 1760–1767.
- Werbos, P. J. (1990). *Proc. IEEE*, **78**, 1550–1560.
- Xiao, W. Y., Wei, Y. X. & Ai, X. Y. (2005). *J. Tsinghua Univ. (Sci. Technol.)*, **45**, 810–812.
- Xie, S. X. (2009). PhD thesis, University of Science and Technology of China, Hefei, Anhui, China.
- Zhang, R. Y. (2006). PhD thesis, Sichuan University, Chengdu, Sichuan, China.
- Zhou, Q. H., Zhang, R. Y. & Taihua, L. I. (2007). *J. Sichuan Univ. Nat. Sci. Ed.* **44**, 111–114.
- Zhou, W., Xiao, Y. J., Zhou, J. B., Hong, X. & Zhao, X. (2015). *J. Terahertz Sci. Electron. Inf. Technol.* **13**, 605–608.