


## RESEARCH ARTICLE

WILEY

# A pruning method of feedforward neural network based on contribution of input components for digital predistortion of power amplifier

Guobo Zhao<sup>1</sup>  | Guizhen Wang<sup>2</sup> | Yingchao Lin<sup>2</sup> | Shulan Li<sup>1</sup> |  
Cuiping Yu<sup>1</sup> | Yuanan Liu<sup>1</sup>

<sup>1</sup>School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, China

<sup>2</sup>Department of Wireless and Terminal Technology, China Mobile Research Institute, Beijing, China

## Correspondence

Cuiping Yu, Beijing Key Laboratory of Work Safety Intelligent Monitoring, School of Electronic Engineering, Beijing University of Posts and Telecommunications, 100876 Beijing, China.

Email: [yucuiiping@bupt.edu.cn](mailto:yucuiiping@bupt.edu.cn)

## Funding information

National Natural Science Foundation of China, Grant/Award Numbers: 61922016, 62090015, 61821001

## Abstract

In this paper, a pruning method based on contribution of input components is proposed to optimize the structure of Feedforward neural network (FNN) model for digital predistortion of power amplifier. The optimized hyperparameters are the specific input components. The pruning algorithm is divided into two stages. The stage I is used to rough rank the importance of input components and the stage II is used to modify the results of stage I. Then, we can get an importance ranking of input components. By deleting unimportant input components, a pruned FNN model with fewer input components can be obtained. Test results show that the proposed algorithm can greatly simplify the FNN model on the premise of ensuring the predistortion performance.

## KEYWORDS

behavior model, feedforward neural network (FNN), power amplifiers (PAs), pruning

## 1 | INTRODUCTION

The digital predistortion (DPD) technique has been widely used and proved to be a linearization method suitable for the power amplifiers (PAs).<sup>1,2</sup>

In recent years, with the continuous increase of signal bandwidth and the use of highly efficient topologies of PAs, in some application scenarios, the neural network model can achieve better linearization effect than the behavior model based on polynomial.<sup>3,4</sup> The real-valued focused time-delay line neural network (RVFTDNN) proposed in<sup>5</sup> divided the I-channel and Q-channel components of input signal, and the nonlinearity and memory effect of PAs were considered. The augmented real-valued time-delay neural network (ARVTDNN)<sup>6</sup> added the envelope components of input signal to the input layer of RVFTDNN to improve the

modeling capability. In<sup>7</sup>, a two hidden layers artificial neural network (2HLANN) was proposed. R. Hongyo et al.<sup>8</sup> increased the number of hidden layers and used deep neural network (DNN) to improve modeling accuracy. In addition, some other adjusted FNN models were proposed for DPD, such as complex-Chebyshev functional link neural network (CCFLNN)<sup>9</sup> cascaded memory polynomial-neural network (MP-NN),<sup>10</sup> and so forth. When PAs have strong nonlinearities with significant memory effects, these FNN models can provide high performance but high complexity. To optimize the FNN model, Wang et al.<sup>11</sup> used genetic algorithms to optimize the two-hidden-layer NN model. Yu et al.<sup>12</sup> balanced the network depth, the number of neurons and the combination of input components to obtain a suitable DNN model. However, the hyperparameters of input layer (memory depth and nonlinear order) concerned by

published optimization methods<sup>11,12</sup> were related to multiple input components. A small change in the hyperparameters of input layer may lead to a huge change in performance.

In this paper, the proposed pruning method focuses on each input component, which will lead to a smoother performance change and a more suitable FNN model can be found. After the input components of input layer are pruned, the pruned FNN model can be further simplified by adjusting the number of hidden layer neurons.

## 2 | FNN MODEL

Compared with other FNN models, ARVTDNN<sup>6</sup> has a very rich input configuration and excellent performance, so it is recommended to choose the input configuration of ARVTDNN to perform pruning. In addition, the structure that is similar to the ARVTDNN has strong expansibility for other DPD applications, such as the codesigned crest factor reduction technique.<sup>13</sup> For

considering the versatility of DPD scenario, ARVTDNN was selected as the example in our paper. The structure of the ARVTDNN is shown in Figure 1. The input vector is written as

$$\mathbf{X} = [x_I(n), x_I(n-1) \dots x_I(n-D), x_Q(n) \dots x_Q(n-D), |x(n)| \dots |x(n-D)| \dots |x(n)|^K \dots |x(n-D)|^K], \quad (1)$$

where  $x_I(n)$  and  $x_Q(n)$  are the I/Q components of the signal  $x(n)$  at sampling time  $n$ ,  $D$  is the maximum memory depth and  $K$  is the maximum nonlinear order.

Output vector is

$$\mathbf{Y} = [y_I(n), y_Q(n)], \quad (2)$$

where  $y_I(n)$  and  $y_Q(n)$  are the I/Q components of the output signal  $y(n)$ .

The output of the  $i$ th neuron in the  $m$ th layer is written as

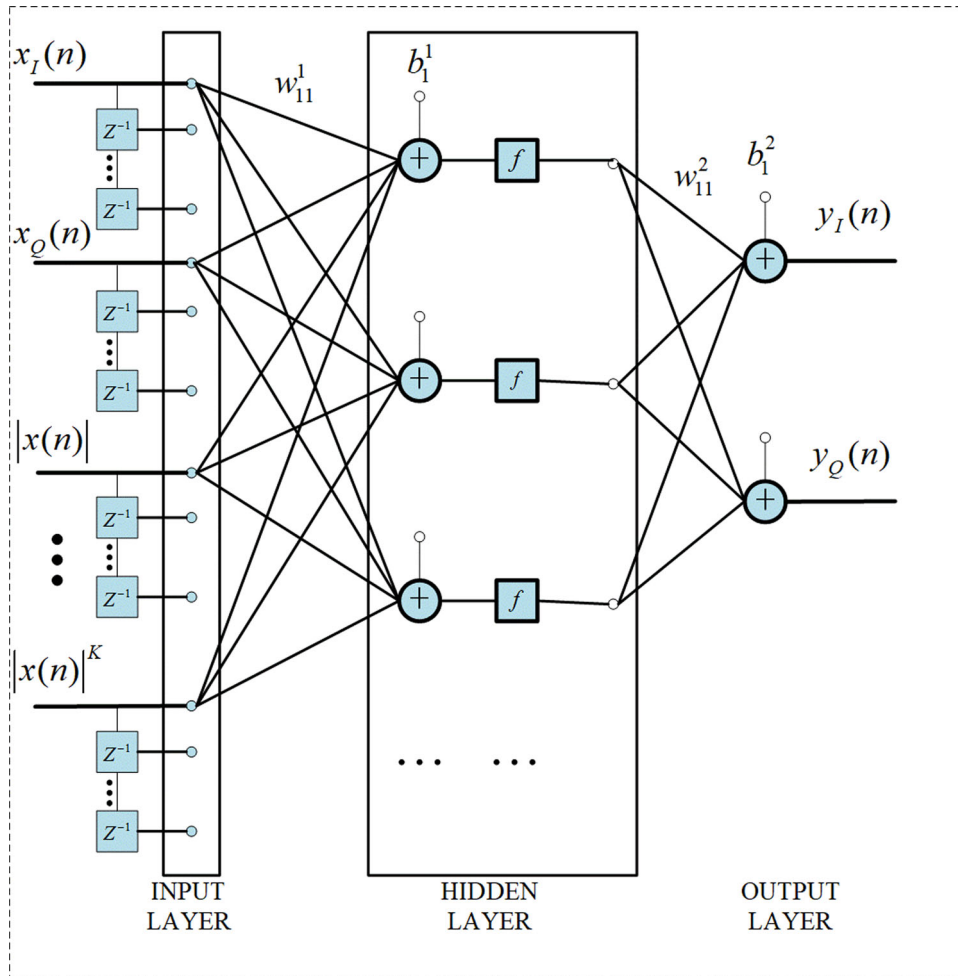


FIGURE 1 The structure of the ARVTDNN model

$$a_i^m = f \left( \sum_{j=1}^J w_{ij}^m a_j^{(m-1)} + b_i^m \right), \quad (3)$$

where  $w_{ij}^m$  is the connection weight from the  $j$ th neuron in the  $(m-1)$ -th layer to the  $i$ th neuron in the  $m$ th layer,  $a_j^{(m-1)}$  is the output of the  $j$ th neuron in the  $(m-1)$ -th layer,  $b_i^m$  is the bias.  $f()$  is the activation function. The hidden layer uses hyperbolic tangent sigmoid as the activation function. The output layer uses a linear activation function.

The model performance is evaluated with the sum of squares of errors (SSE) between the measured PA output and the estimated model output, which is written as

$$SSE = \sum_{n=1}^N [(I_{\text{est}}(n) - I_{\text{mea}}(n))^2 + (Q_{\text{est}}(n) - Q_{\text{mea}}(n))^2], \quad (4)$$

where  $I_{\text{est}}(n)$  and  $Q_{\text{est}}(n)$  are the I/Q components of the estimated model output, while  $I_{\text{mea}}(n)$  and  $Q_{\text{mea}}(n)$  are the I/Q components of the measured PA output, respectively.

### 3 | PRUNING ALGORITHM

An algorithm to prune the FNN model for predistortion of PA is proposed in this section. The proposed method can be used in a variety of FNN models for predistortion, such as ARVTDNN, DNN, MPNN, and so forth, to reduce the scale and implementation complexity of model. The idea of the algorithm is to rank the input components of input layer according to their importance, and then remove the unimportant input components. After pruning, a pruned FNN model with low complexity and meeting the performance requirements can be obtained.

#### STAGE I

**Input:**  $\mathbf{X}$ ,  $D$ ,  $K$

**Output:**  $S$

1. **for**  $i = 1: (K+2)D$
2. get the input matrix  $\mathbf{X}_R^{(i)}$  of the reduced model  $\text{NN}^{(i)}$  by removing  $x_i$  from  $\mathbf{X}$
3. **if**  $(i-1) \bmod D = 0$
4. initialize randomly  $\mathbf{W}_I^{(i)}$
5. **else**

6.  $\mathbf{W}_I^{(i)} \leftarrow \mathbf{W}_O^{(i-1)}$

7. **end if**

8. train  $\text{NN}^{(i)}$  and save  $\mathbf{W}_O^{(i)}$

9. calculate SSE and save it to the set  $S$

10. **end for**

#### STAGE II

**Input:**  $\mathbf{X}$ ,  $S$ ,  $SSE^{(0)}$ ,  $T$ ,  $N$

**Output:**  $\mathbf{X}_P$

1. according to set  $S$  from small to large, reorder the input components of  $\mathbf{X}$  to get  $\mathbf{X}^{(0)}$ .
2. **for**  $n = 1: ((K+2)D-1)$
3. get input matrix  $\mathbf{X}^{(n)}$  of model  $\text{NN}^{(n)}$  by removing  $x_n$  from  $\mathbf{X}^{(n-1)}$
4. train  $\text{NN}^{(n)}$  and calculate  $SSE^{(n)}$  according to (4)
5. **if**  $|SSE^{(n)} - SSE^{(n-1)}| \geq T$
6.  $s_n \leftarrow \max\{S\}$
7. **end for**
8. According to the set  $S$  from small to large, remove  $[(K+2)D - N]$  input components from  $\mathbf{X}$  to get  $\mathbf{X}_P$

The algorithm is divided into two stages. In the stage I, referring to the method in Mozer and Smolensky,<sup>14</sup> the contribution of the input component is preliminarily determined by comparing the error changes before and after the input component is deleted, which is equivalent to directly evaluating the contribution of the input component by the error after the removal of the input component. The larger the error, the more important this input component is.

In STAGE I, the initial FNN model  $\text{NN}^{(0)}$  is determined.  $\mathbf{X}$  is the input matrix of  $\text{NN}^{(0)}$ . To evaluate the importance of the input component  $x_i$ , a reduced model  $\text{NN}^{(i)}$  is constructed by removing the input component  $x_i$  from  $\text{NN}^{(0)}$ .  $\mathbf{X}_R^{(i)}$  is the input matrix of the reduced model  $\text{NN}^{(i)}$ .  $\mathbf{W}_I^{(i)}$  is the initial coefficient matrix of the reduced model  $\text{NN}^{(i)}$ , and  $\mathbf{W}_O^{(i)}$  is the local optimal coefficient matrix of the reduced model  $\text{NN}^{(i)}$  after training.

When calculating the SSE of different reduced models, retraining reduced models is required every time. To save the computing time, the inheritance initialization was proposed. It should be noted that the

inheritance initialization based on the concept of transfer learning<sup>15</sup> introduces prior knowledge compared with the random initialization, which made the inheritance initial point of training closer to the local optimum of the model than the random initial point, but this did not change the local optimal solution of the model, so the inheritance initialization can speed up the training without affecting the performance, which was also proved by the test results. All reduced models for evaluating memoryless input components ( $x_l(n)$ ,  $x_Q(n)$ ,  $|x(n)|$ , ...,  $|x(n)|^K$ ) are initialized randomly. The initial solution of the reduced model for evaluating the importance of memory input components will adopt the local optimal solution of the previous adjacent reduced model. As shown in Figure 2, the inheritance initialization proposed in the stage I can greatly speed up the convergence of the reduced models. The random initialization tends to converge within 20 iterations, and the proposed inheritance initialization tends to converge within 10 iterations. This makes the run time simulated under MATLAB of the stage I reduce by 42%. At the same time, Figure 2 also shows that the inheritance initialization can converge to get almost the same performance as the random initialization.

All SSE values are saved to the set  $S$ . The larger the SSE value of  $NN^{(i)}$ , the more important the input component  $x_i$  is. So, a rough rank of input components can be obtained through STAGE I.

In the model with redundancy, when some input components are removed alone, it makes little impact to the overall performance. However, when the system is pruned, once these input components that originally have little impact on the overall performance are removed, they will have a great impact on the overall performance.

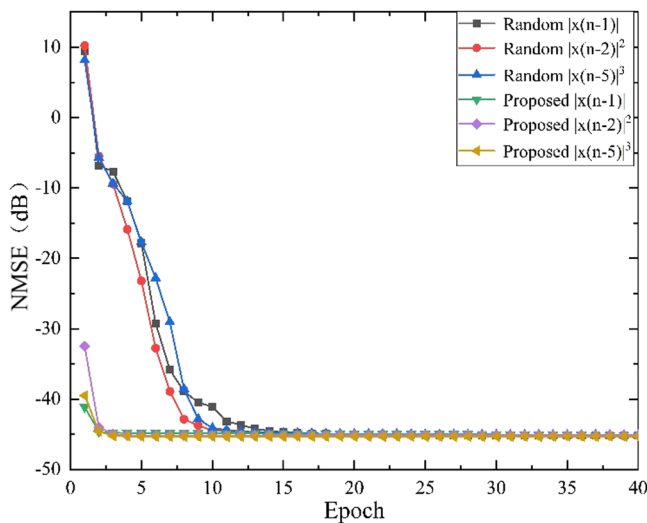


FIGURE 2 Convergence performance in test 1

Therefore, take the rank obtained in the stage I as the index of the stage II, then remove the input components from the initial FNN model one by one in order until there is only one input component left, and decide whether to correct the ranking of an input component by the magnitude of the error change before and after deleting it.

In STAGE II, the set  $S$  is obtained by stage I,  $SSE^{(0)}$  is the SSE value of the initial FNN model  $NN^{(0)}$ ,  $T$  is a threshold and  $N$  is the number of input components that need to be pruned. According to stage I, reorder input components of  $\mathbf{X}$  according to their importance to get  $\mathbf{X}^{(0)} = [x_1, x_2, \dots, x_{(K+2)D}]$ , where  $x_1$  is the input component corresponding to the minimum SSE in stage I. Then,  $n$  ( $n = 1 : ((K+2)D - 1)$ ) unimportant input components are removed from  $\mathbf{X}$  to obtain the input matrix  $\mathbf{X}^{(n)}$  of the model  $NN^{(n)}$ . If the change of SSE from  $NN^{(n-1)}$  to  $NN^{(n)}$  does not exceed  $T$ , it means that the currently input component is not important for network fitting; if the change of SSE from  $NN^{(n-1)}$  to  $NN^{(n)}$  exceeds  $T$ , it means that the performance drops sharply, and the currently input component  $x_n$  is important. So, we give the  $s_n$  corresponding to the input component  $x_n$  to the maximum value of the set  $S$  to revise its importance ranking. By this way, a more accurate ranking is obtained.  $N$  is the number of input components to be reserved, representing the pruning degree. The value of  $N$  is used to weigh the performance and running complexity of the proposed model. The appropriate  $N$  can be selected according to different application scenarios. After deleting unimportant input components, the input matrix  $\mathbf{X}_p$  of the pruned FNN model is obtained. The optimal number of hidden neurons of the pruned FNN model can be determined empirically.

## 4 | EXPERIMENTAL RESULTS

To verify the proposed algorithm, a test bench was set up. we used SMBV100B vector signal generator (VSG) to generate RF signal. After being attenuated, the output signal of the PA was down-converted and sampled by FSW-43. To better verify the effectiveness of the proposed algorithm, we conducted two sets of tests, considering different PA types, different bandwidths and different PAPR.

In test 1, the ZHL-16W-43-S + PA was tested at 2.4 GHz with the maximum output power at 42 dBm, and a 5 G NR signal with 100 MHz bandwidth (the peak-to-average power ratio was 7 dB) was used. the sampling rate was 491.52 MHz.

During the test, the initial FNN model was ARVTDNN,<sup>6</sup> and  $D$  and  $K$  of the ARVTDNN were selected as 8 and 3 respectively, and the hidden layer using 40 neurons can provide enough computing power. So, the initial FNN model

was 40-40-2 (this structure is used to represent the structure of FNN model, which means that there are 40 input neurons, 40 hidden layer neurons, and 2 output neurons). 20000 samples were used to train the model. In the training process, Levenberg-Marquardt (L-M) algorithm was used to update the coefficients. In the stage II,  $T = 1$  dB was selected.

Figure 3 shows the adjacent channel power ratio (ACPR) performance of the different pruned models when the number of input components takes different values. When the number of input components is reduced from 40 to 10, the performance changes slowly, and when the number of input components is less than 10, the performance deteriorates rapidly. Therefore,  $N$  value is chosen as 30, which means that the proposed

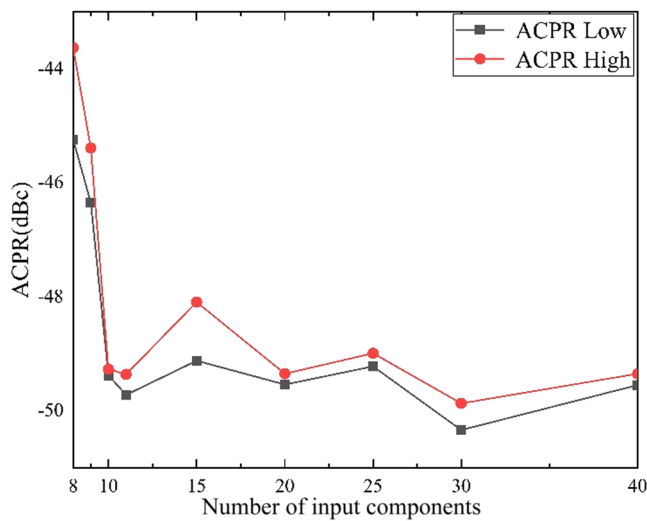


FIGURE 3 Performance changes caused by pruning in test 1

model only needs 10 input components. After the input components are optimized, the optimal number of hidden neurons in the proposed model is determined to be 13 by empirical method.

Table 1 shows the performance comparison of the proposed model with other models. It can be found that the proposed model needs the least input components, but its performance is far better than RVFTDNN,<sup>5</sup> and close to that of ARVTDNN<sup>6</sup> and DNN.<sup>12</sup> Compared with other models, the proposed model needs the least number of coefficients. Therefore, the proposed model can effectively reduce the complexity of the FNN model on the premise of maintaining performance.

In test 2, the Doherty PA designed by our team was also used to verify the performance of the proposed method. the Doherty PA was tested at 2.55 GHz with the maximum output power at 43 dBm, and an OFDM signal with 20 MHz bandwidth (the peak-to-average power ratio was 8 dB) was used. the sampling rate was 122.88 MHz.

Table 2 shows that the proposed method is effective, and the network structure is simple enough compared with other models on the premise of ensuring good enough performance. Compared with ARVTDNN,<sup>6</sup> the NMSE and the worst ACPR performance change within 1 dB, and the number of model coefficients is reduced from 842 to 262, which is reduced by 69%.

## 5 | CONCLUSION

In this paper, a pruning method based on contribution of input components of FNN model for predistortion is proposed. By judging the importance of each input

Models	Structure	NMSE (dB)	ACPR (dBc) ( $\mp 100$ MHz)	Num. model coefficients
RVFTDNN <sup>5</sup>	16-20-2	-36.14	-43.48/-41.46	382
ARVTDNN <sup>6</sup>	40-16-2	-43.18	-49.96/-49.37	690
DNN <sup>12</sup>	24-15-15-15-2	-42.14	-48.49/-48.63	887
Proposed model	10-13-2	-41.75	-49.16/-48.53	171

TABLE 1 Performance and complexity of models in test 1

Models	Structure	NMSE (dB)	ACPR (dBc) ( $\mp 20$ MHz)	Num. model coefficients
RVFTDNN <sup>5</sup>	16-30-2	-32.44	-39.20/-29.33	572
ARVTDNN <sup>6</sup>	25-30-2	-44.58	-52.68/-52.45	842
DNN <sup>12</sup>	15-15-15-15-2	-43.27	-50.98/-50.97	752
Proposed model	10-20-2	-43.92	-51.75/-52.02	262

TABLE 2 Performance and complexity of models in test 2



component, the unimportant input components are deleted from the initial FNN model. The test results show that the proposed model can greatly simplify the model structure on the premise of ensuring the predistortion performance.

## ACKNOWLEDGMENT

This study was supported by the National Natural Science Foundations of China (Nos. 61922016, 62090015 and 61821001).

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## ORCID

Guobo Zhao  <https://orcid.org/0000-0002-9799-5125>

## REFERENCES

1. Ghannouchi FM. Power amplifier and transmitter architectures for software defined radio systems. *IEEE Circuits Syst Mag*. 2010;10(4):56-63.
2. Fager C, Eriksson T, Barradas F, Hausmair K, Cunha T, Pedro JC. Linearity and efficiency in 5G transmitters: new techniques for analyzing efficiency, linearity, and linearization in a 5G active antenna transmitter context. *IEEE Microw Mag*. 2019;20(5):35-49.
3. Guillena E, Li W, Montoro G, Quaglia R, Gilabert PL. Reconfigurable DPD based on ANNs for wideband load modulated balanced amplifiers under dynamic operation from 1.8 to 2.4 GHz. *IEEE Trans Microw Theory Techn*. 2022;70(1):453-465.
4. Hu X, Liu Z, Yu X. et al. Convolutional neural network for behavioral modeling and predistortion of wideband power amplifiers. *IEEE Trans Neural Netw Learn Syst*. 2022;33(8):3923-3937.
5. Rawat M, Rawat K, Ghannouchi FM. Adaptive digital predistortion of wireless power amplifiers/transmitters using dynamic real-valued focused time-delay line neural networks. *IEEE Trans Microw Theory Techn*. 2010;58(1):95-104.
6. Wang D, Aziz M, Helaoui M, Ghannouchi FM. Augmented real-valued time-delay neural network for compensation of distortions and impairments in wireless transmitters. *IEEE Trans Neural Netw Learn Syst*. 2019;30(1):242-254.
7. Mkadem F, Boumaiza S. Physically inspired neural network model for RF power amplifier behavioral modeling and digital pre-distortion. *IEEE Trans Microw Theory Techn*. 2011;59(4):913-923.
8. Hongyo R, Egashira Y, Hone TM, Yamaguchi K. Deep neural network-based digital predistorter for Doherty power amplifiers. *IEEE Microw Wireless Compon Lett*. 2019;29(2):146-148.
9. Li M, Liu J, Jiang Y, Feng W. Complex-Chebyshev functional link neural network behavioral model for broadband wireless power amplifiers. *IEEE Trans Microw Theory Techn*. 2012;60(6):1979-1989.
10. Chu J, Chen W, Chen L, Feng Z. A cascaded memory polynomial-neural network behavior model for digital predistortion. 2020 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO). 2020:1-3.
11. Wang S, Roger M, Sarrazin J, Lelandais-Perrault C. Hyperparameter optimization of two-hidden-layer neural networks for power amplifiers behavioral modeling using genetic algorithms. *IEEE Microw Wireless Compon Lett*. 2019;29(12):802-805.
12. Yu X, Hu X, Liu Z, Wang C, Wang W, Ghannouchi FM. A method to select optimal deep neural network model for power amplifiers. *IEEE Microw Wireless Compon Lett*. 2021;31(2):145-148.
13. Wang S, Roger M, Sarrazin J, Lelandais-Perrault C. Augmented iterative learning control for neural-network-based joint crest factor reduction and digital predistortion of power amplifiers. *IEEE Trans Microw Theory Techn*. 2020;68(11):4835-4845.
14. Mozer MC, Smolensky P. Skeletonization: a technique for trimming the fat from a network via relevance assessment. *Adv Neural Inf Process Syst*. 1988;107:115.
15. Ando RK, Zhang T. A framework for learning predictive structures from multiple tasks and unlabeled data. *J Mach Learn Res*. 2005;6:1817-1853.

**How to cite this article:** Zhao G, Wang G, Lin Y, Li S, Yu C, Liu Y. A pruning method of feedforward neural network based on contribution of input components for digital predistortion of power amplifier. *Microw Opt Technol Lett*. 2023;65:98-103. doi:10.1002/mop.33465