Inverse Design of Pixelated Antenna Based on Residual Neural Network

Mengyue Li
College of Electronic and Information Engineering, NUAA
Nanjing, China
euygnem@126.com

Yi Wang
College of Electronic and Information Engineering, NUAA
Key Laboratory of Radar Imaging and Microwave Photonics, NUAA
Nanjing, China
jflsjfls@nuaa.edu.cn

Min Zhu
Key Laboratory of Radar Imaging and Microwave Photonics, NUAA
Jinling Institute of Technology
Nanjing, China
zomi@jit.edu.cn

***Abstract*—A deep-learning-based antenna inverse design method is proposed, which utilizes a residual neural network (ResNet) to construct an inverse model to directly predict the antenna structure from the target's scattering parameters. The one-to-many problem can be solved to some extent by introducing a pre-trained forward model. The simulation results show that the method can effectively predict a reasonable antenna structure and ensure its physical feasibility, providing an efficient way for smart antenna design.**

Keywords*—*antenna inverse design; deep learning; residual neural network

I. Introduction

Antenna design is critical in wireless communication, radar and RF circuit design. Traditional antenna design method relies on empirical and simulation optimization, which makes it difficult to efficiently obtain an antenna structure that meets specific requirements.

In recent years, the idea of deep learning has been widely used in the field of antenna design [1], [2]. Most of the reported deep learning models are forward modeling, where the geometric parameters are set as inputs and the EM response is the output. Inverse design is an emerging approach where appropriate geometric parameters are directly derived from the expected EM response. The convolutional neural network (CNN), which is most commonly used, is capable of effectively extracting complex data features. However, gradient vanishing or gradient explosion problems often occur during CNN training, affecting the convergence speed and prediction accuracy of the model [3]. To solve these problems, the residual neural network (ResNet) is introduced into antenna inverse design to improve gradient propagation and enhance model performance.

In this paper, we propose an antenna design method based on ResNet, as shown in Fig. 1. In addition to the inverse model, we introduce a pre-trained forward model to ensure that the structure obtained from the inverse design can correctly reproduce the target's scattering parameters (S11). The one-to-many problem can be solved to some extent by calculating S11 parameter from the predicted structure and comparing it to optimize the output of the inverse model [4]. The loss-weighted sum of the two models is used as the total loss of the inverse model, and then the inverse model continues to be trained iteratively. Simulation results demonstrate that the ResNet proposed outperforms the traditional CNN in antenna inverse design.

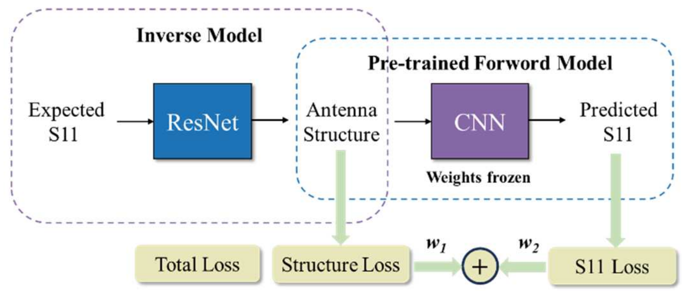


Fig. 1. Framework of proposed model.

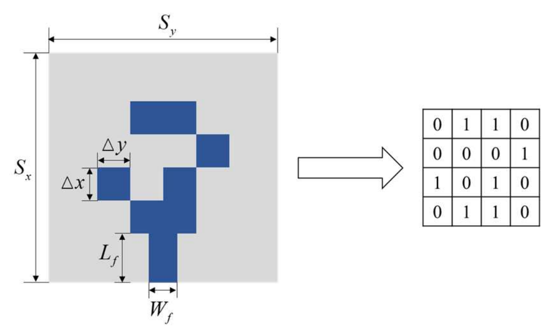


Fig. 2. Sample of a pixelated antenna and its matrix representation. ($S_x = S_y =$ 20 mm; $\Delta x = \Delta y = 3$ mm; $L_f = 4$ mm; $W_f = 2$ mm)

II. Method

A. Dataset generation

To make the antenna design more flexible, a prototype rectangular patch antenna is designed and evenly divided into 4 $\times$4 pixels (each is a smaller rectangle). Each unit is coded with 1 indicating metal covered unit and 0 indicating only substrate. An illustration is shown in Fig. 2. To feed the antenna, the two pixels connected to the microstrip line are fixed to 1 [5]. The height of substrate is 0.5mm and the relative permittivity is 2.65.

978-1-7334677-1-1© 2025 ACES

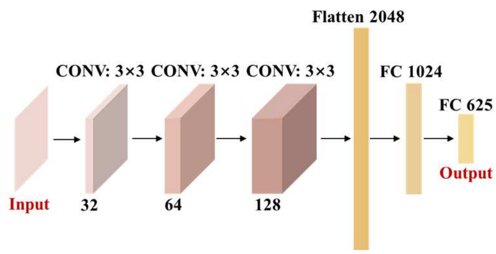


Fig. 3. Architecture of the forward model.

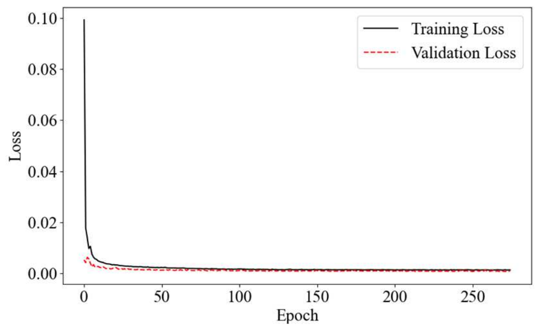


Fig. 4. Training loss and validation loss for forward model.

The frequency range is from 2 to 22 GHz with 625 sampling points selected at equal intervals.

We use CST-Python co-simulation to generate the dataset. Except for the two fixed elements, the remaining 14 elements are randomly generated, resulting in a total of 2^{14} (16384) possible structures. A total of 1912 antennas and their S11 data are generated, which are distributed in an 80:20 ratio for training, and validation, respectively.

B. Forward Model Training

The architecture of the forward model is shown in Fig. 3, consisting of three convolutional layers and two fully connected layers. The loss function is set with the mean square error function:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{1}$$

where n is the number of samples, y_i is the actual value of the i-th sample, and $\hat{y}_i$ is the predicted value of the i-th sample.

We set the batch size to 64 and the learning rate to 0.001. The training and validation results of the network are shown in Fig. 4. At epoch=250, the training loss and validation loss are 0.00132 and 0.0008, respectively. Notably, we include Dropout in the network because it randomly drops neurons while training, making model parameter updates unstable and thus increasing training losses. But at validation, the Dropout turns off and all neurons participate in the calculation, resulting in the validation loss that may be lower than the training loss [6].

C. Inverse Model Training

The ResNet is constructed by inserting skip connections to an ordinary CNN. The architecture of the inverse model, as shown in Fig. 5, contains an initial convolution layer, three residual blocks (each consisting of two sub-blocks), and two fully connected layers.

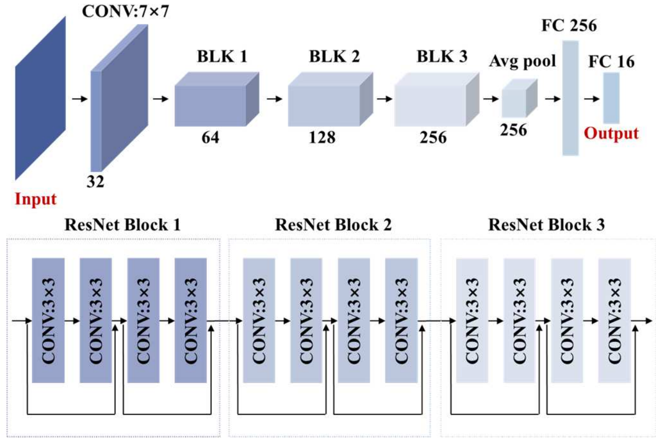


Fig. 5. Architecture of the inverse model.

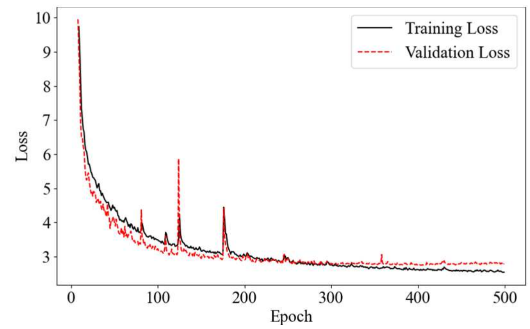


Fig. 6. Training loss and validation loss for inverse model (using ResNet).

Fig. 1 shows that the expected scattering parameter P is processed by the inverse model to obtain the designed antenna structure S' and then processed by the forward model to obtain the expected scattering parameters P' of the network. The total loss function L_{total} is given by

$$L_{total} = w_1 * \mathrm{MSE}\left(S, S'\right) + w_2 * \mathrm{MSE}\left(P, P'\right) \tag{2}$$

where MSE (S, S') is the structure loss, MSE (P, P') is the scattering parameter loss, and w_1 and w_2 are their weights respectively.

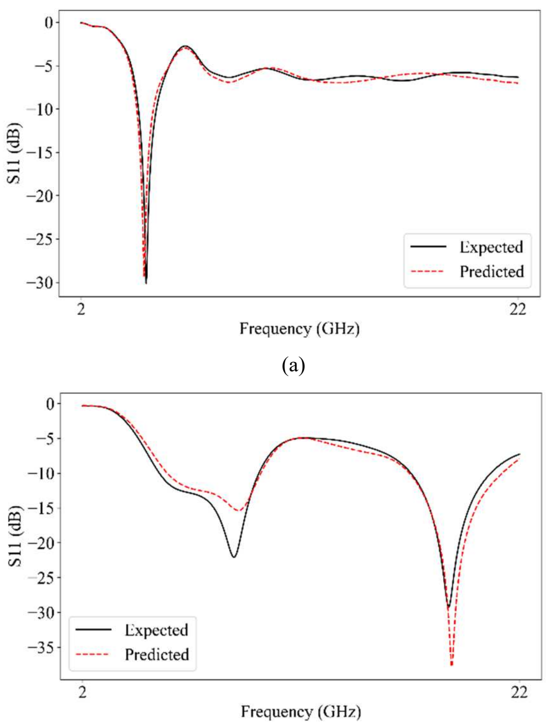
Since the antenna structure is represented by a 0/1 matrix of 4 × 4, the initial structure predicted by the model needs to be converted to the same 0 or 1. The Sigmoid function is used as the activation function of the last fully connected layer, and its formula is given as

$$\mathrm{Sigmoid}\left(x\right) = \frac{1}{1+e^{-x}} \tag{3}$$

which maps the input to the output in the range 0 to 1. A threshold function is then set to determine whether the output is 0 or 1, as follows

$$f\left(x\right) = \begin{cases} 0, x \le 0.5 \\ 1, x > 0.5 \end{cases} \tag{4}$$

The ResNet is trained with a batch size of 64 for 500 epochs. The weights are updated with the Adam optimizer and w_1 and w_2 are set to 1 and 10, respectively. The final training loss is 2.46 and validation loss is 2.82, as shown in Fig. 6.



(a)

(b)

Fig. 7. Comparison of the expected S11 and predicted S11. (a) Single band antenna. (b) Dual-band antenna.

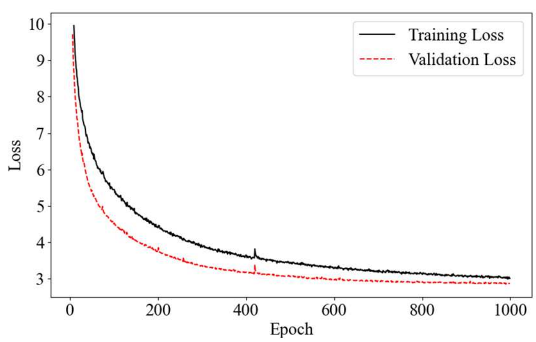


Fig. 8. Training loss and validation loss for inverse model (using CNN).

III. Application Examples

To verify the feasibility of the proposed model, we select two different antenna requirements: (a) single band and (b) dual-band, for which S11 is used as the input to the neural network. The results are shown in Fig. 7(a) and (b). Once the model is successfully trained, the expected antenna structure can be designed in seconds.

Besides, we use CNN as a reverse model of the network. Specifically, it consists of four convolutional layers and two fully connected layers. In addition, the Sigmoid function is also used as the activation function. Under the same data set, the loss function of CNN is shown in Fig. 8. In contrast to ResNet, CNN's MSE drops to about 2.86 when epoch=993, while ResNet's MSE drops to about 2.81 when epoch=373. It is not difficult to conclude that using ResNet's model is more efficient.

IV. Conclusion

In this paper, we introduce a new inverse antenna design method. Specifically, we have designed a pixelated antenna with the help of a residual neural network and a pre-trained forward model. A single-band antenna and a dual-band antenna are successfully designed, which proves the feasibility of the proposed model. In addition, the ResNet’s prediction process has also been compared with using convolutional neural networks, showing that the proposed method is more efficient.

V. Acknowledgment

This work is supported by the Key Laboratory of Radar Imaging and Microwave Photonics, Ministry of Education, China, No. NJ20230007.

References

[1] A. Gupta, E. A. Karahan, C. Bhat, K. Sengupta and U. K. Khankhoje, "Tandem Neural Network Based Design of Multiband Antennas," IEEE Transactions on Antennas and Propagation, vol. 71, no. 8, pp. 6308-6317, Aug. 2023.

[2] L. -Y. Xiao, W. Shao, F. -L. Jin, B. -Z. Wang and Q. H. Liu, "Inverse Artificial Neural Network for Multiobjective Antenna Design," IEEE Transactions on Antennas and Propagation, vol. 69, no. 10, pp. 6651-6659, Oct. 2021.

[3] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770-778, 2016.

[4] Y. Lu, J. Liu, Z. Zong and Z. Wei, "Multiphysics Inverse Design of Frequency-Selective Surface by Data-Physics-Driven Deep Neural Network," IEEE Transactions on Antennas and Propagation, vol. 72, no. 11, pp. 8739-8749, Nov. 2024.

[5] M. Ding, R. Jin, and J. Geng, “Optimal design of ultra wideband antennas using a mixed model of 2-D genetic algorithm and finite-difference time-domain,” Microw., Opt. Technol. Lett., vol. 49, no. 12, pp. 3177–3180, Dec. 2007.

[6] Hinton G E, Srivastava N, Krizhevsky A, "Improving neural networks by preventing co-adaptation of feature detectors," Computer Science, vol. 3, no. 4, pp. 212-223, 2012.