Progress In Electromagnetics Research, Vol. 172, 41–49, 2021

Phase Synthesis of Beam-Scanning Reflectarray Antenna Based on Deep Learning Technique

Tao Shan, Maokun Li*, Shenheng Xu, and Fan Yang

Abstract—In this work, we investigate the feasibility of applying deep learning to phase synthesis of reflectarray antenna. A deep convolutional neural network (ConvNet) based on the architecture of AlexNet is built to predict the continuous phase distribution on reflectarray elements given the beam pattern. The proposed ConvNet is sufficiently trained with data set generated by array-theory method. With radiation pattern and beam direction arrays as input, the ConvNet can make real-time and fairly accurate predictions in milliseconds with the average relative error below 0.7%. This paper shows that deep convolutional neural networks can "learn" the principle of reflectarray phase synthesis due to their inherent powerful learning capacity. The proposed approach may provide us a potential scheme for real-time phase synthesis of antenna arrays in electromagnetic engineering.

1. INTRODUCTION

Reflectarray is an important kind of high-gain antennas, which usually consists of a flat reflecting surface and an illuminating feed horn [1]. By combining advantages of both parabolic reflector and array antennas, reflectarrays have many promising features, such as high efficiency, low profile, low cost, and easy fabrication [2]. In the design and applications of reflectarrays, phase synthesis is a vital step. It computes the phase-shift of each unit cell given the desired beam pattern. Many array antennas require rapid phase synthesis, such as beam-scanning reflectarrays [3] and phased arrays.

Conventional phase synthesis methods include analytical methods and nonlinear optimization methods. Analytical methods [4–6] are fast but not accurate enough for complex beam. Nonlinear optimization methods are usually applied to solve complex phase synthesis problems, including genetic algorithm (GA) [7, 8], particle swarm optimization (PSO) [9], simulated annealing [10], gradient based method [11], etc. These methods convert phase synthesis problems into optimization problems with constraints that are solved via iterative procedure. The iterative scheme demands large computing resources and cannot provide real-time response.

Various efforts are made to accelerate phase synthesis, such as real-valued GA [12], the accelerated gradient calculation [11], parallel computation on Graphics Processing Units (GPU) [13]. Learning-based methods are also applied to speed up phase synthesis with good accuracy. Artificial neural networks (ANNs) help shape beams in various application scenarios, such as characterizing EM response of reflectarray elements [14], multiple source tracking [15], adaptive beamforming, and inference cancellation [16]. Support vector machines (SVMs) are also applied to model the reflection coefficients of reflectarray unit cells [17, 18].

Recently, deep learning techniques are experiencing a huge leap forward due to the development of high-performance computing. Deep learning techniques demonstrate powerful learning ability and achieve remarkable successes in natural language, image and video processing [19–21]. Deep learning

Received 13 September 2021, Accepted 22 December 2021, Scheduled 30 December 2021

* Corresponding author: Maokun Li (maokunli@tsinghua.edu.cn).
The authors are with the Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing, China.

techniques have also attracted the attention from many other fields, including physics and engineering. Various works have been reported, for example, computational fluid dynamics [22], antennas and propagation [23], inverse scattering problems [24, 25], biomedical imaging [26], Poisson's equation solver [27], and programmable metasurface [28]. The common approach of these works is to sufficiently train the deep neural network with massive data in the offline stage and then the trained network can provide reliable computations in the online stage.

In this work, we explore the feasibility of applying deep learning techniques to phase synthesis of a reflectarray antenna. As a starting point, we design and train a deep convolutional neural network to predict unit cells' phase-shifts given single beam pattern. The proposed deep convolutional neural network (ConvNet) is built based on the architecture of AlexNet [20]. AlexNet has proven its excellent performance in ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2010 and 2012. The input of ConvNet includes radiation pattern and beam direction information arrays. The output is an array of continuous phase-shift corresponding to the input beam pattern. Both training and testing data sets are generated by the array-theory method. The average relative error of the predicted and desired phase-shift distribution can reach below 0.7%. The corresponding radiation patterns are also in a good agreement, which further validates the feasibility of the proposed method. This study is more complete and detailed with expanded numerical results compared with the previous conference paper [29].

This paper is organized as follows: Section 2 introduces the array-theory method and the proposed deep convolutional neural network. Section 3 demonstrates numerical examples and evaluates performance of the proposed deep convolutional neural network. Conclusions and further discussions are summarized in Section 4.

2. FORMULATION

2.1. Array-Theory Method

Array-theory method stems from the conventional array antenna theory [6]. It is widely used to calculate the radiation pattern in the modeling and design of reflectarrays. The typical structure of reflectarray antenna is illustrated in Figure 1. It consists of an illuminating horn antenna and a reflecting surface on which unit cells are periodically distributed.

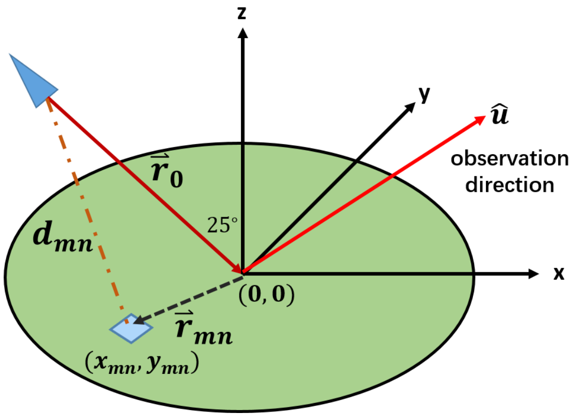

Figure 1. The typical structure of reflectarray.

The radiation pattern of a reflectarray can be formulated using array-summation technique [6]:

$$E(\theta,\varphi)=\sum_{m=1}^{M}\sum_{n=1}^{N}\cos^{q_e}\theta\frac{\cos^{q_f}\theta_{f,mn}}{|\vec{r}_{mn}-\vec{r}_0|}\,e^{-jk_0(|\vec{r}_{mn}-\vec{r}_0|-\vec{r}_{mn}\cdot\hat{u})}\cos^{q_e}\theta_{e,mn}e^{j\phi_{mn}} \tag{1}$$

where r_{mn}, r_0, and $\hat{u}$ denote the position vector of the mn-th unit cell, horn antenna, and observation direction, respectively; $\phi(m,n)$ represents the reflection phase of the mn-th unit cell. In order to simplify the calculation, array-theory method uses the cosine q model to approximate the radiation patterns of the horn antenna and each unit cell of the reflectarray. In Eq. (1), q_f and q_e denote different q factors of the horn antenna and reflectarray unit cell. As illustrated in Figure 1, the distances between unit cells and the illustrating horn antenna are different. Then the incident phase of each unit cell that is introduced by the horn antenna is different, which can be written as:

$$\phi_{inc}(m,n) = -k_0 d_{mn} \tag{2}$$

where d_{mn} denotes the distance between the mn-th unit cell and the horn antenna; k_0 denotes the propagation constant. To generate a main beam in the direction (θ_r, ϕ_r), the reflection phase of the mn-th unit cell of a reflectarray can be written as:

$$\phi(m,n) = -k_0 \sin\theta_r \cos\varphi_r x_{mn} - k_0 \sin\theta_r \sin\varphi_r y_{mn} \tag{3}$$

where (x_{mn}, y_{mn}) is the Cartesian coordinate of the mn-th unit cell. Its phase-shift to form the main beam can be obtained:

$$\phi_{ele}(m,n) = \phi(m,n) - \phi_{inc}(m,n) = k_0 d_{mn} - k_0 \sin\theta_r (\cos\varphi_r x_{mn} + \sin\varphi_r y_{mn}) \tag{4}$$

2.2. ConvNet Model

A deep convolutional neural network is built based on the structure of AlexNet [20]. Figure 2 shows the architecture of the proposed ConvNet. The ConvNet consists of five convolutional layers and three full connected layers(FCNs). The convolutional kernel sizes from the first to fifth layer are 11 × 11, 5 × 5, 3 × 3, 3 × 3 and 3 × 3 respectively. The stride of the 11 × 11 convolutional kernel is 2 while the stride of others is 1. The input and output channels of convolutional kernels are denoted in Figure 2. ReLUs are included in all layers except the last layer to introduce nonlinearity and max pooling are included in the first, second and fifth layer of ConvNet to perform downsampling. The local response normalization (LRN) is applied in the first and second convolutional layer to improve the generalization ability. LRN is inspired by the lateral inhibition mechanism in the biological neuron system [20]. It can normalize the output of convolutional layers based on the competition scheme, which can be written as [20]:

$$b^i_{x,y} = \frac{a^i_{x,y}}{\left(k + \alpha \displaystyle\sum_{j=\max(0,\frac{i-n}{2})}^{\min(N-1,\frac{i+n}{2})} (a^j_{x,y})^2\right)^{\beta}} \tag{5}$$

where $b^j_{x,y}$ is the output of LRN; n is the number of neighboring kernels at the same spatial position; N is the total number of kernels in one layer; $a^i_{x,y}$ is the output of i-th convolutional kernel with ReLU

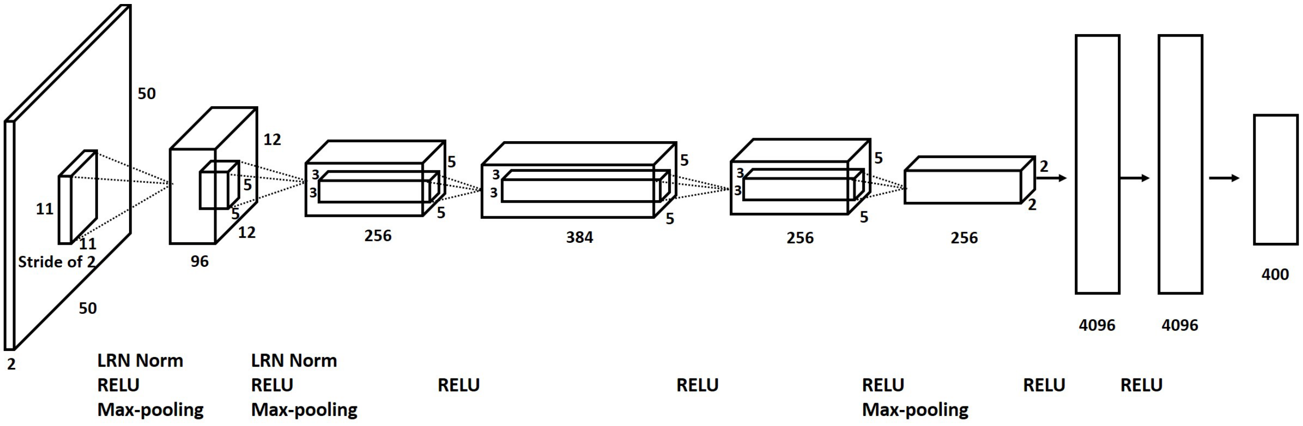

Figure 2. Architecture of the proposed convolutional neural network.

nonlinearity at the spatial position (x, y). In Eq. (5), α, β, n, and k are hyper parameters and set as 0.0001, 0.75, 4, and 1. The output size of FCN is the same as the number of unit cells in the reflectarray. Dropout [30] is applied into the first two FCNs and the dropout rate is set as 0.7. Dropout can reduce the computational complexity and prevent over-fitting by random dropping some connections in FCNs [30].

The input of the ConvNet includes two arrays representing the desired radiation pattern and beam direction information, as shown in Figure 3. The radiation pattern is calculated in the uv plane ($u = \sin\theta\cos\varphi, v = \sin\theta\sin\varphi$). The beam direction (θ_r, φ_r) can also be transformed into a point (u_r, v_r) in the uv plane. Therefore, the beam direction can be represented as a two-dimensional array, of which every element is the distance between the (u_r, v_r) and (u_i, v_j). It can be written as:

$$dist = \sqrt{(u_i - u_r)^2 + (v_j - v_r)^2} \tag{6}$$

where i, j is the index of each element in the beam direction array. The value of the center point is set to 0 as the anchor point, as shown in Figure 3. The output of the ConvNet is the continuous phase-shift distribution given the desired radiation pattern.

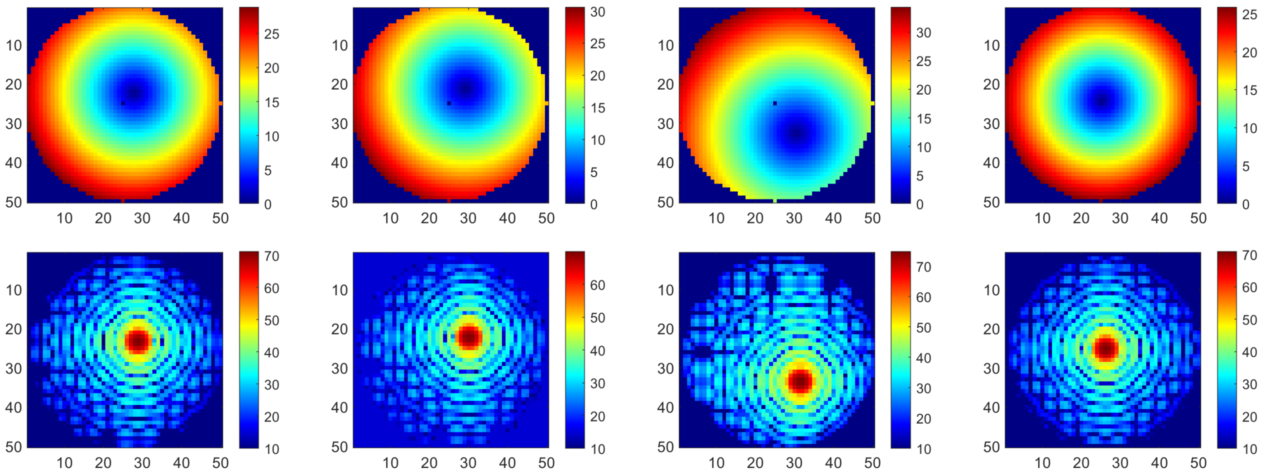

Figure 3. Four examples of input pairs of ConvNet. The first row is the beam direction array and the second row is the radiation pattern (unit: dB).

The objective function of the ConvNet is designed as:

$$f_{obj} = \frac{1}{N_1 N_2} \sum_{j=1}^{N_1} \sum_{i=1}^{N_2} \left| \frac{\log 10(\phi_{i,j}) - \log 10(\phi'_{i,j})}{\log 10(\phi'_{i,j})} \right|^2 \tag{7}$$

where (i, j) is the index of each unit cell; ϕ' and ϕ denote the continuous phase-shift distribution computed by the array-theory method and predicted by the ConvNet, respectively. Eq. (7) evaluates the difference between the logarithm values of both ϕ' and ϕ, which can help stabilize the training process by avoiding fast attenuation of phase-shift values.

3. RESULTS AND ANALYSIS

The numerical experiment takes a square aperture 20 × 20 reflectarray as an example. Its operation frequency is 12.5 GHz, and the square aperture's length is 10λ. The direction of incident wave is fixed as ($\varphi_i = 180°, \theta_i = 25°$). The elevation angle φ_r of the reflected wave varies from 0° to 179°, and the azimuth angle θ_r varies from 1° to 60°. The sampling interval of the reflected direction is 1°. We generate 10800 samples based on the array-theory method, of which 8640 samples are randomly selected as training data, and 2160 samples are used as testing data.

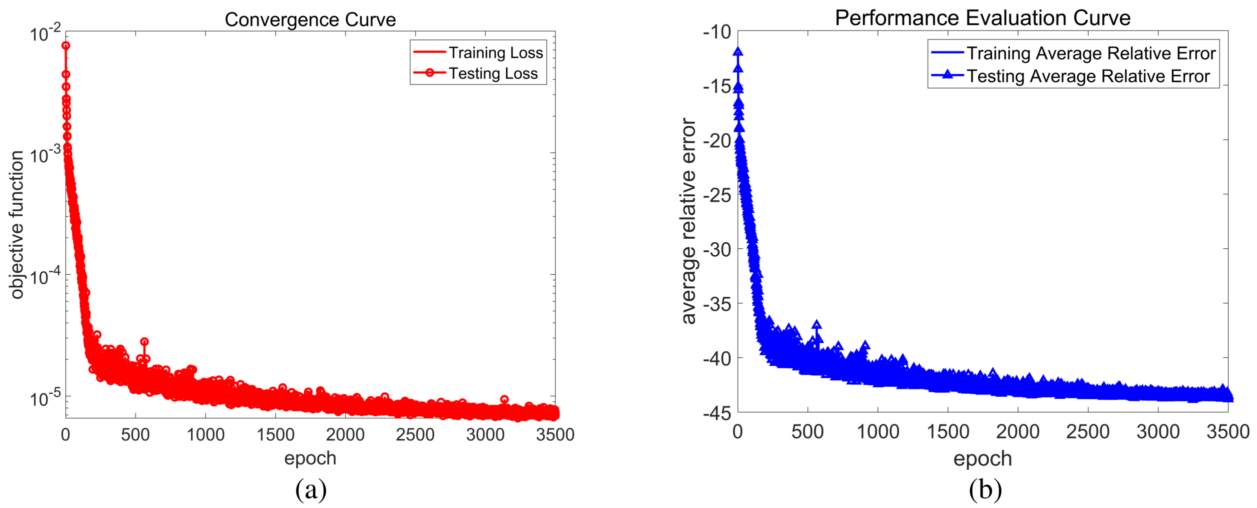

Figure 4. Convergence curve of the objective function and performance evaluation function, (a) convergence curve of objective function, (b) performance evaluation curve.

The input of the ConvNet is two 50 × 50 arrays, as shown in Figure 3, and the output is a 20 × 20 array representing the phase-shift distribution on the reflectarray surface. The ConvNet is implemented on Tensorflow, and the computation platform is an Nvidia P100 GPU card. With Eq. (7) as the objective function, the ConvNet is trained by the adaptive moment estimation (Adam) optimizer [31] that is a first-order stochastic gradient optimization algorithm. The learning rate of Adam optimizer is initialized at 0.00005, and it is multiplied by 0.5 every 250 epochs. The mini-batch training scheme is employed to take 40 data samples as a batch to update the parameters of ConvNet. The convergence curve is shown in Figure 4(a). The convergence curve drops sharply with the large learning rate at the beginning of training. Then with the decrease of learning rate, the convergence falls marginally until convergence. The training loss agrees well with the testing loss, which means that the ConvNet is trained sufficiently. For a more detailed evaluation of the ConvNet's performance, the average relative error is defined as the evaluation function of the ConvNet. For a single unit cell's phase-shift, the average relative error can be written as:

$$re\_error(m,n) = \frac{|\phi(m,n)_{ConvNet} - \phi(m,n)_{array-theory}|}{|\phi(m,n)_{array-theory}|} \tag{8}$$

where $\phi(m,n)_{ConvNet}$ and $\phi(m,n)_{array-theory}$ are mn-th unit cell's phase-shift predicted by the ConvNet and computed by the array-theory method. Then the average relative error of the i-th samples can be written as:

$$re\_error_{aver_i} = 20\log 10\left(\frac{\sum_m \sum_n rel\_error(m,n)}{\sum_m \sum_n 1}\right) \tag{9}$$

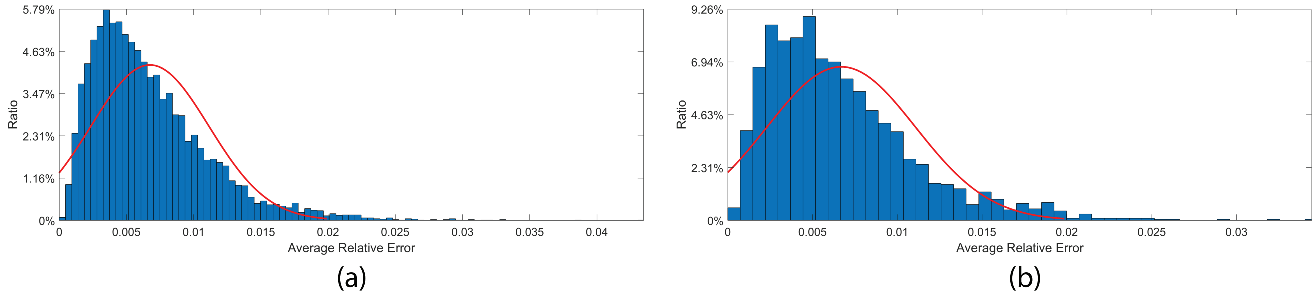

Figure 5. Average relative error histogram of training and testing data set, (a) training data set, (b) testing data set.

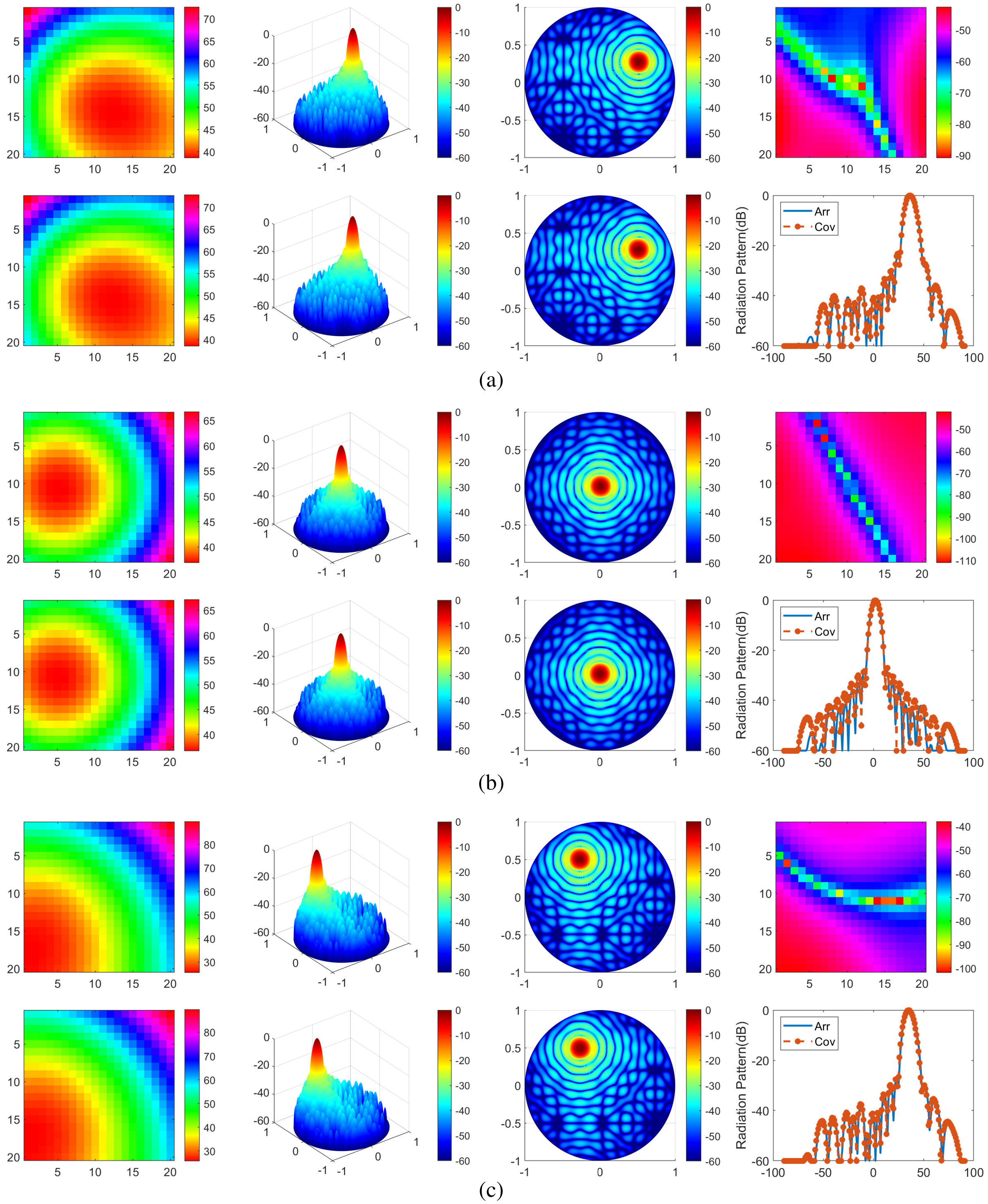

Figure 6. Phase-shift distributions and their 3D and 2D radiation patterns computed by the array-theory method and the ConvNet. In each subfigure, the first row (from left to right): phase-shift distribution obtained by the array-theory method, the corresponding 3D and 2D beam pattern, the average relative error distribution (dB) between radiation patterns computed by array-theory method and the ConvNet; the second row (from left to right): phase-shift distribution obtained by the ConvNet, the corresponding 3D and 2D beam pattern, the beam pattern comparison on the principle plane of these two methods. (a) Example 1. (b) Example 2. (c) Example 3.

Figure 4(b) shows the convergence curve of the evaluation function during the training and testing process. The average relative error finally converges below -43 dB (0.7%). Figure 5(a) and Figure 5(b) show the average relative error histogram of all samples in the training and testing data sets, respectively. The mean and standard deviation of the average relative errors are $0.0067, 1.9223 \times 10^{-05}$ for training data set and $0.0067, 1.9229 \times 10^{-05}$ for testing data set. The training and testing data sets have almost the same distribution of average relative errors. It is consistent with the convergence curve of the performance evaluation function. It also validates that the ConvNet is trained sufficiently, and there exists no overfitting. Figure 6 shows three results randomly selected from the testing data set. The phase-shift distributions obtained by the array-theory method and the ConvNet reach a good agreement. Thus, the discrepancy between the corresponding radiation patterns is very small. The detailed comparison on the principle plane shows that minor differences concentrate on side lobes, and the main beams are almost the same. As to computing time, the ConvNet takes 0.00017 s to generate a phase-shift distribution given the beam pattern. This reveals a good potential that the deep learning techniques may provide accurate and real-time responses in the phase synthesis of reflectarrays.

4. CONCLUSIONS AND DISCUSSIONS

In this paper, we investigate the feasibility of applying deep learning techniques to phase synthesis of reflectarrays. As a starting point, the phase synthesis of a 20 × 20 beam-scanning reflectarray is taken as an example. A deep convolutional neural network is built and sufficiently trained, then it can compute fairly accurate phase-shift distributions in milliseconds. The final average relative error reaches below 0.7%. This work may open a new door for real-time phase synthesis of reflectarrays in complex application scenarios. It also shows that deep learning techniques can capture the inner law between radiation patterns and the corresponding phase-shift distributions via learning process. In the future work, we will try to apply deep learning techniques to compute more complex phase synthesis problems.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61971263, in part by the National Key Research and Development Program of China under Grant 2018YFC0603604, in part by the Beijing Innovation Center for Future Chip, in part by the Institute for Precision Medicine, Tsinghua University, Beijing, China, and in part by Tencent Foundation through the XPLORER PRIZE.

REFERENCES

1. Huang, J. and J. A. Encinar, *Reflectarray Antennas*, Vol. 30, John Wiley & Sons, 2007.
2. Mao, Y., S. Xu, F. Yang, and A. Z. Elsherbeni, "A novel phase synthesis approach for wideband reflectarray design," *IEEE Transactions on Antennas and Propagation*, Vol. 63, No. 9, 4189–4193, 2015.
3. Yang, H., F. Yang, X. Cao, S. Xu, J. Gao, X. Chen, M. Li, and T. Li, "A 1600-element dual-frequency electronically reconfigurable reflectarray at X/Ku-band," *IEEE Transactions on Antennas and Propagation*, Vol. 65, No. 6, 3024–3032, 2017.
4. Barbiere, D., "A method for calculating the current distribution of Tschebyscheff arrays," *Proceedings of the IRE*, Vol. 40, No. 1, 78–82, 1952.
5. Chakraborty, A., B. Das, and G. Sanyal, "Beam shaping using nonlinear phase distribution in a uniformly spaced array," *IEEE Transactions on Antennas and Propagation*, Vol. 30, No. 5, 1031–1034, 1982.
6. Nayeri, P., F. Yang, and A. Z. Elsherbeni, *Reflectarray Antennas: Theory, Designs, and Applications*, Wiley-IEEE Press, 2018.

7. Johnson, J. M. and Y. Rahmat-Samii, "Genetic algorithm optimization and its application to antenna design," *Proceedings of IEEE Antennas and Propagation Society International Symposium and URSI National Radio Science Meeting*, Vol. 1, 326–329, IEEE, 1994.
8. Lommi, A., A. Massa, E. Storti, and A. Trucco, "Sidelobe reduction in sparse linear arrays by genetic algorithms," *Microwave and Optical Technology Letters*, Vol. 32, No. 3, 194–196, 2002.
9. Robinson, J. and Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics," *IEEE Transactions on Antennas and Propagation*, Vol. 52, No. 2, 397–407, 2004.
10. Ferreira, J. A. and F. Ares, "Pattern synthesis of conformal arrays by the simulated annealing technique," *Electronics Letters*, Vol. 33, No. 14, 1187–1189, 1997.
11. Prado, D. R., A. F. Vaquero, M. Arrebola, M. R. Pino, and F. Las-Heras, "Acceleration of gradient-based algorithms for array antenna synthesis with far-field or near-field constraints," *IEEE Transactions on Antennas and Propagation*, Vol. 66, No. 10, 5239–5248, 2018.
12. Mahanti, G., A. Chakraborty, and S. Das, "Phase-only and amplitude-phase only synthesis of dual-beam pattern linear antenna arrays using oating-point genetic algorithms," *Progress In Electromagnetics Research*, Vol. 68, 247–259, 2007.
13. Capozzoli, A., C. Curcio, A. Liseno, and G. Toso, "Fast, phase-only synthesis of aperiodic reflectarrays using NUFFTs and CUDA," *Progress In Electromagnetics Research*, Vol. 156, 83–103, 2016.
14. Robustillo, P., J. Zapata, J. A. Encinar, and J. Rubio, "Ann characterization of multi-layer reflectarray elements for contoured-beam space antennas in the Ku-band," *IEEE Transactions on Antennas and Propagation*, Vol. 60, No. 7, 3205–3214, 2012.
15. El Zooghby, A. H., C. G. Christodoulou, and M. Georgiopoulos, "A neural network-based smart antenna for multiple source tracking," *IEEE Transactions on Antennas and Propagation*, Vol. 48, No. 5, 768–776, 2000.
16. El Zooghby, A. H., C. G. Christodoulou, and M. Georgiopoulos, "A neural-network-based linearly constrained minimum variance beamformer," *Microwave and Optical Technology Letters*, Vol. 21, No. 6, 451–455, 1999.
17. Prado, D. R., J. A. Lopez-Fernandez, G. Barquero, M. Arrebola, and F. Las-Heras, "Fast and accurate modeling of dual-polarized reflectarray unit cells using support vector machines," *IEEE Transactions on Antennas and Propagation*, Vol. 66, No. 3, 1258–1270, 2018.
18. Prado, D. R., J. A. López-Fernández, M. Arrebola, and G. Goussetis, "Support vector regression to accelerate design and crosspolar optimization of shaped-beam reflectarray antennas for space applications," *IEEE Transactions on Antennas and Propagation*, Vol. 67, No. 3, 1659–1668, 2018.
19. Collobert, R. and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," *Proceedings of the 25th International Conference on Machine Learning*, 160–167, 2008.
20. Krizhevsky, A., I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, Vol. 25, 1097–1105, 2012.
21. Ng, J. Y.-H., M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4694–4702, 2015.
22. Guo, X., W. Li, and F. Iorio, "Convolutional neural networks for steady flow approximation," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 481–490, 2016.
23. Massa, A., D. Marcantonio, X. Chen, M. Li, and M. Salucci, "DNNs as applied to electromagnetics, antennas, and propagation — A review," *IEEE Antennas and Wireless Propagation Letters*, Vol. 18, No. 11, 2225–2229, 2019.
24. Chen, X., Z. Wei, M. Li, and P. Rocca, "A review of deep learning approaches for inverse scattering problems (invited review)," *Progress In Electromagnetics Research*, Vol. 167, 67–81, 2020.
25. Wei, Z. and X. Chen, "Deep-learning schemes for full-wave nonlinear inverse scattering problems," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 57, No. 4, 1849–1860, 2018.

26. Li, M., R. Guo, K. Zhang, Z. Lin, F. Yang, S. Xu, X. Chen, A. Massa, and A. Abubakar, "Machine learning in electromagnetics with applications to biomedical imaging: A review," *IEEE Antennas and Propagation Magazine*, 2021.

27. Shan, T., W. Tang, X. Dang, M. Li, F. Yang, S. Xu, and J. Wu, "Study on a fast solver for poisson's equation based on deep learning technique," *IEEE Transactions on Antennas and Propagation*, Vol. 68, No. 9, 6725–6733, 2020.

28. Shan, T., X. Pan, M. Li, S. Xu, and F. Yang, "Coding programmable metasurfaces based on deep learning techniques," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Vol. 10, No. 1, 114–125, 2020.

29. Shan, T., M. Li, S. Xu, and F. Yang, "Synthesis of reflectarray based on deep learning technique," *2018 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC)*, 1–2, IEEE, 2018.

30. Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv:1207.0580, 2012.

31. Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.