Metamaterials Design Method based on Deep learning Database

Xiaoshu Zhou\(^1\), Qide Xiao\(^{1,2,*}\), Han Wang\(^{1,2,3,*}\)

1. Beijing Institute of Technology, Zhuhai, China
2. City University of Macau, Macau, China
3. Zhuhai Institute of Advanced Technology Chinese Academy of Sciences, China

*Email: wanghan@ziat.ac.cn

Abstract. In recent years, deep learning has risen to the forefront of many fields, overcoming challenges previously difficult to solve by traditional methods. In the field of metamaterials, there are significant challenges in the design and optimization of metamaterials, including the need for a large number of labeled data sets and one-to-many mapping when solving inverse problems. Here, we will use deep learning methods to build a metamaterial database to achieve rapid design and analysis methods of metamaterials. These technologies have significantly improved the feasibility of more complex metamaterial designs and provided new metamaterial design and analysis ideas.

1. Introduction

Recently, Professor Yongmin Liu from Northeastern University in the United States proposed a method for automatically designing and optimizing chirality and other properties by learning metamaterials' structure and properties through deep learning networks \([1]\). Chirality describes the structural characteristics that cannot be directly superimposed between the object and its mirror image. Just like a human hand, it cannot overlap in the same direction. This property is prevalent in nature and has aroused vital research interest from experts and scholars. It is widely used in many industries and fields, such as physical optics, electromagnetics, and drug synthesis. However, in nature, in most cases, the chiral optical response of natural materials is feeble. It cannot be detected and collected by our technology, which makes the research and development of chiral materials difficult. The appearance of metamaterials has reduced the weak chiral response of natural materials to a certain extent. By artificially designing the metamaterial structure, the chiral properties of the material can be effectively enhanced. Today, researchers have produced many metamaterials with obvious chirality, but these metamaterials still cannot meet chiral reaction research needs. Under normal circumstances, researchers will use a lot of calculations to design the structure of metamaterials. However, due to the complexity and intuitiveness of the metamaterial structure and its optical response, this traditional method relies on many calculations and cannot be performed quickly. To accurately obtain metamaterials' properties, it is even impossible to reverse design suitable metamaterials through the optically responsive structure.

In response to metamaterial calculation and simulation problems, Professor Liu Yongmin introduced the idea of deep learning to learn the structure and characteristics of metamaterials to achieve the purpose of designing and improving opponent metamaterials. Professor Liu designed a metamaterial structure with two open resonant rings superimposed in a cross form (the arm lengths of the two are \(L_1\) and \(L_2\), and the crossing angle of the two is \(\alpha\)), and by adding two dielectric layers (the thickness of the two is...
T1 and T2, respectively), and add a metal substrate at the bottom of the medium to make its transmittance equal to 0.

This material has different electromagnetic properties under different rotation modes. The method proposed by Professor Liu uses the reflection coefficient to describe its optical properties, and its arm-length, crossing angle and medium thickness are used as the input parameters of the network. In constructing the deep learning neural network, Professor Liu created a unique neural network structure with a bidirectional design. The forward network PN converts the 1*5 metamaterial structure into a 3*201 reflection coefficient used for metamaterials. Modeling the reverse network’s reflection coefficient uses a typical convolutional neural network structure, two consecutive convolutional layers, and a fully connected convolutional layer, and a well-trained forward-path is used in the PN full-cycle Training to model the spectra of metamaterials.

The forward network PN and the reverse network AN are directly combined through the forward combiner and the reverse combiner. At the same time, a tensor layer and an up-sampling module are added to control the network parameters to achieve better convergence. The tensor Unlike the fully connected layer, which combines the output of previous neurons linearly, the tensor layer can simulate the second-order relationship between pairs of variables. This relationship was initially proposed for the completion of the knowledge base, and recently in chemistry, The field has been applied. This two-way network combination method enables the network to have training and calculation functions for both the metamaterial structure and electromagnetic properties, which significantly improves the calculation ability and accuracy of the deep learning network. In the forward simulation of the bidirectional neural network constructed by Professor Liu, its deep learning model can quickly calculate the reflection coefficients and frequency points of different metamaterial structural parameters. In reverse design, its deep learning model can rely on the required metamaterial characteristics to design appropriate structural parameters. It is particularly worth mentioning that even if only basic parameters such as the reflection frequency point and amplitude in the metamaterial properties are provided, the two-way neural network model constructed by Professor Liu can also design a suitable physical structure of chiral metamaterials according to performance requirements.

Researcher Michael Mrejen of the Center for Nanoscience and Nanotechnology at Tel Aviv University and Mallkieltzik, a researcher of the School of Computer Science, published an article demonstrating a deep learning-based approach that can simplify the design of basic nanophotonics and metamaterial elements in recent decades [2]. Although many breakthroughs in the optical field have brought unprecedented imaging capabilities, their impact and penetration in daily life are hindered by complex and repetitive process modeling, nano-preparation, and nano-characterization cycles. The fundamental reason is that the calculation of optical response requires a lot of numerical calculations to solve Maxwell's equations, which is a very time-consuming and calculation-related process. More importantly, the problem is highly non-linear. Even with the most advanced numerical tools, the inverse problem of on-demand optical response design of metamaterial nanostructures is a daunting task. At the same time, although computer science has been used to deal with the complex functions of nanophotonic imaging, design, and characterization, these methods are either aimed at imaging and characterization with resolution beyond the diffraction limit or in the design process of photonics nanophotonic devices. Provide help. However, so far, few computing technologies can fully solve these two aspects of nano-plasmonic.

Based on the above problems, researchers Mallkieltzik and Michael Mrejen proposed a deep learning-based method to simplify the design of basic nanophotonics and metamaterial elements and provide them with unique, reliable, time-saving, and special functions [3]. Complex nanostructure fields. Accurate optical response characterization capability. This method solves the highly non-linear problem of reasoning tasks by creating a model containing two-way knowledge and uses the method of training two-way networks to complete the structural design and characterization tasks from optical response spectroscopy to metamaterial nanoparticles. In the bidirectional neural network they built, the first network GN uses three fully connected parallel architecture layers. Each layer contains three fully connected 100 neurons, and each set of data in each layer is sent to a different layer. Then connect three
groups of neurons to a "connected lawyer," and then connect another seven fully connected layers, each layer composed of 750 neurons. The last layer consists of eight neurons, which encode the calculated geometric shape. The second network SPN contains eight fully connected layers. The first layer receives eight output values, material properties, and polarization signs of the previously constructed network as input. Each inner layer consists of 1,000 neurons, which are entirely connected to the next layer. The last layer of the network consists of 43 neurons, and each neuron represents the Y value of the transmission map at a predefined wavelength and a fixed wavelength value. The neural network data set contains simulated samples of 3,000 wavelengths. Each piece is sampled based on 12 different nanostructure geometries, so each geometry has different side lengths, angles, and material properties. This method can not only accurately calculate the spectral response of the nanostructure but also solve the inverse problem and provide the geometric size and design of a single metamaterial nanostructure for the required optical response in a targeted manner. This deep learning method provides a method for the on-demand design of plasma and metamaterial structures and is suitable for sensing, targeted therapy, and other fields. In addition, the computing power of the deep learning model also provides a broad prospect for the multiple characterizations of nanostructures beyond the diffraction limit.

Zhang Qian, Liu Zhe, Xiang Wan, and others of Southeast University in China published a paper "Machine Learning Design of Anisotropic Digitally Encoded Metamaterials" and proposed a metamaterial design using the same isotropic machine learning technology method [4]. The difference between the ultra-thin planar array metamaterials and the three-dimensional subwavelength structure metamaterials lies in their small size and small profile. In recent years, scholars have proposed a variety of metamaterial surfaces to design radiating and scattering electromagnetic beams. The metasurface of these elements is arranged in the phase of gradient changes in the elements, such as abnormal reflection and refraction controlled by the generalized Snell's law. The conversion mode between space mode and metasurface, etc. Subsequently, related experts proposed the concept of digitally encoded metasurfaces, expounding various elements of encoded metasurfaces, such as relative phase information with a fixed phase difference, relative amplitude information with a fixed amplitude difference, and relative time information with a fixed time difference. Related scholars have proposed the design of the body of an anisotropic encoding element. The anisotropic encoding element must have a fixed phase difference (for example, 180°) orthogonal polarization to achieve the desired effect. Suppose various other anisotropic encoding element bodies with different phase differences (100 types) are realized. In that case, 100 different encoding elements must be designed to realize these phase differences, which cannot be promoted and applied. To this end, Zhang Qian et al. proposed using machine learning to design the surface of anisotropic digital coding elements. This method selects discrete random lattice elements composed of many square metal sub-blocks at the microscopic level. In digital coding, due to the micro-range coding mode, "1" and "0" indicate sub-blocks with and without metal. Due to its randomness and diversity, various compensation phases can be generated in various directions so that machine learning can design anisotropic digital code element surface evaluation and calculation systems. Then, by combining deep learning and binary particle swarm optimization (BPSO) for machine learning design, the best phase performance square code sub-block coding mode, structural features can be quickly obtained. The method of using machine learning to design the phase characteristics and structure of anisotropic metamaterials provides great potential and space for the design of metamaterials.

In this paper, based on the current deep learning research on metamaterials introduced above, this paper uses Matlab and CST Microwave Studio to simulate the electromagnetic properties of metamaterials. It proposes a method to calculate the electromagnetic properties of metamaterials using deep neural networks. Effectively simulate the electromagnetic properties of periodic metamaterial structures.

2. Method
In electromagnetics, periodic metamaterials are regarded as homogeneous media with the same thickness, and equivalent electromagnetic numerical calculation methods are used to analyze the properties of metamaterials. At present, the commonly used equivalent electromagnetic numerical
calculation methods include the analytical method, numerical method, and parameter inversion method. Analysis method. The analysis method is based on the metamaterial structure, constructing similar models, and inferring the conductive and magnetic parameters through the geometric dimensions of the model, which is intuitive and convenient. However, if the metamaterial structure is relatively complex, it will be more challenging to implement this method in the analysis process. The numerical method uses the method of moments (MoM), finite element method (FEM) and finite time-domain integration (FITD), and other electromagnetic numerical calculation methods to calculate electric field, current, magnetic field, and magnetic flux and other parameters, and replace the average electric field current ratio with a constant dielectric, replace the permeability with the average magnetic flux ratio, and then obtain the electromagnetic parameters of the metamaterial. By using the above methods, CST electromagnetic simulation software has been fully put into practical application. It can quickly and easily analyse the properties of metamaterials through modelling and simulation and uses internal adaptive grids to ensure the convergence of results and the credibility of numerical calculations [5-14].

The inversion method uses the reflection coefficient S11 and the transmission coefficient S21 to invert the electromagnetic parameters of the metamaterial.

\[
S_{11} = \frac{Z(1-e^{j2\pi d})}{1-Z^2e^{j2\pi d}}
\]

According to the S parameter, the characteristic impedance is obtained:

\[
Z = \pm \sqrt{\frac{(1+S_{11})^2 - S_{21}^2}{(1-S_{11})^2 - S_{21}^2}}
\]

\[
e^{j\pi d} = \frac{1-S_{11}^2 + S_{21}^2}{2S_{21}} \pm j\sqrt{\frac{1-(1-S_{11}^2+S_{21}^2)^2}{2S_{21}}}
\]

The dielectric constant, permeability and refractive index of metamaterials can be obtained from the following formulas:

\[
\begin{align*}
\mu &= nZ \\
\varepsilon &= n/Z \\
n &= \frac{1}{kd} \left\{ [\ln(e^{j\pi d})^n + 2n\pi] + j[\ln(e^{j\pi d})^n] \right\}
\end{align*}
\]

What needs to be emphasized here is that this method has certain adaptability in most cases, but in metamaterials with metal substrates, the electromagnetic parameters will change, which will affect the inversion effect.

In the experiment, the CST microwave studio was used to model and simulate metamaterials in the experiment to obtain the data required for the experiment. CST Microwave Studio is a software specially designed for high-frequency passive components and antenna simulation. The software uses finite element method (FEM) and finite integration (FI) and other electromagnetic simulation methods to quickly and effectively analyse antennas, couplers and filters, as well as various passive components, such as resonators. CST Microwave Studio uses the Finite Integral (FI) algorithm to improve the performance of the solver through ideal boundary fitting technology, so as to realize the simulation and calculation of electromagnetic parameters. CST Microwave Studio provides a variety of solution methods, such as time domain solver, frequency domain solver, and intrinsic solver, etc., and realizes the design and analysis of various electromagnetic equipment through the combination of optimizer and parameter scanning tool.

The electromagnetic simulation of CST Microwave Studio can be divided into the following steps:

1. Create a simulation project: Create a new project and define the appropriate units and background materials.
2. Establish geometric model: Establish or import the geometric model of passive components as required.

3. Set related parameters: set the frequency range, incident port and boundary conditions and other parameters.

4. Split structure grid: select the grid type to split the model for simulation and solution.

5. Perform solution simulation: Choose a suitable solver for simulation. The solution result can be obtained by creating basic matrix and adjoint matrix, calculating port mode, excitation and transient analysis and other processes.

The metamaterial structure constructed in this experiment uses "material1" with a dielectric constant of $4 \times 4$ as the background material, the upper patch material is PEC, the simulation frequency band is 8-13GHz, and the boundary conditions of the X and Y axes are set to Unit cell, That is, the model can be periodically extended in the X and Y axis directions, and the frequency domain solver in CST can be used to solve this problem. The analogy output is the S11 reflection coefficient of 49 sampling points. This experiment uses the idea of deep learning, so a large amount of data is needed as a training set for neural network training. In the above, CST Microwave Studio can only be used to model and simulate one passive device at a time, which faces a large amount of data requirements. Simply using CST Microwave Studio requires a lot of manpower and time. In response to this difficulty, this paper proposes a method that combines MATLAB and CST to model and simulate the properties of metamaterials, which greatly improves the efficiency of data collection and reduces the need for manpower. In MATLAB, the invoke0 function is used to call the CST control so that it can automatically initialize, set parameters, create models and splits, select the solver to be solved, and export data. A series of operations have realized the function of using MATLAB to control the CST super-control modeling material structure and simulation characteristics.

For the periodic metamaterial structure, we choose a periodic structure for experimental data collection. For the period unit, we divide it into $16 \times 16$ square units. These square sub-blocks are divided into two parts: the gray sub-block is covered by the metal film, and the blue sub-block does not cover anything. In the numerical code representation, in the microscopic view, the sub-blocks with metal and without metal respectively represent "1" and "0", that is, "1" represents PEC material, and "0" represents background material. In this way, we can obtain a $16 \times 16$ "0,1" matrix, which is equivalent to the periodic form of a periodic unit. For the $16 \times 16$ matrix, we choose the $8 \times 8$ matrix on the upper left as the reference, construct the $16 \times 16$ matrix in a symmetrical manner, construct the geometric model in the form of coding in MATLAB, and learn the convolutional neural network in depth. As input, use MATLAB to construct the required model in CST through matrix coding, and set the frequency range, incident wave direction and other parameters for simulation. The reflection coefficient S11 obtained by simulation is the characteristic parameter of the metamaterial we need. From the frequency, the real and imaginary parts are displayed in corresponding forms.

The deep learning network consists of an input layer, three convolutional pool layers, a fully connected layer and an output layer. The input is the equivalent matrix constructed above, and the output is the SU simulation result of CST Microwave Studio. The convolutional layer is located in the first part of the entire convolutional neural network and is used to extract the structural features of the metamaterial. The pooling layer is associated with the convolutional layer and is used to aggregate certain features to optimize the neural network. In this experiment, a convolutional neural network with three convolutional pool layers is constructed, which makes the feature map obtained by the neural network deeper and more effective. The first convolutional layer is associated with the input matrix. The input matrix is a parameter of $16 \times 16 \times 1$. We choose a convolution kernel with a size of $5 \times 5 \times 1$ to perform the convolution operation. Since the size of the feature map is inconsistent with the input matrix, we need to fill the boundary of 2 pixels to match the size of the feature map with the input matrix. After the first convolutional layer, the pooling layer is connected to aggregate certain functions, and then the maximum pooling method is used to perform the pooling operation. At this time, the input of the pooling layer is the feature map of the convolutional layer calculated above the "output". We use a pooling layer with a size of $2 \times 2 \times 1$ to pool the feature map, which is able to calculate the parameters of the pooled
After the first layer of convolution and pooling, we compress the original data with a size of 16*16 into the feature map data with a size of 8*8, which is used as the input of the second convolutional layer. The second convolution layer uses a convolution kernel with a size of 5 * 5 * 1 to perform convolution operations on the elements after the first convolution and merging. We fill 2 pixels to make the size of the feature map and the input matrix the same. For the feature map after the second convolution, we use the largest pooling layer with a size of 2 * 2 * 1 to perform the pooling operation. Through merging, the feature map obtained after merging is used as the input of the third convolutional layer. After three times of convolution and merging, we decompose the original data with a size of 16 * 16 * 1 into 2 * 2 * 128 feature maps, which can effectively extract the features of the original data and help the fully connected layer to classify the features. After convolution and merging, a fully connected layer needs to be used to classify the obtained features according to different groups. We have built a fully connected layer with a three-layer structure. Each corresponding layer contains 4096, 1024 and 512 neurons. The number of neurons in each layer gradually decreases, and the neurons in each layer are completely connected. The merged elements will be passed to the output layer by layer. The last layer is the output layer, which consists of 49 neurons. These neurons represent the S11 reflection coefficients of artificial metamaterials with different frequencies and specific structures. Based on this output, the characteristic spectrum of the metamaterial can be drawn.

The deep learning convolutional neural network used to calculate the feature parameters of metamaterials is constructed above.

3. Results

In this section, the constructed neural network will be used to calculate the characteristic parameters of the metamaterial. First, use the neural network to train the training set samples, add the path of the training set to the neural network training function, and extract 20 sets of data from the training set for training, and gradually loop the extraction each time, and store the network every time the neural network is performed. When training for 1000 times, these parameters are convenient for reusing network parameters, and the loss obtained in each printing will be printed out at the same time. The training process of the neural network is as follows:

![Neural network training and testing flowchart](image)

**Figure 1.** Neural network training and testing flowchart.
After many iterations, the training stops when the neural network loss function drops below $5 \times 10^{-3}$, and then reads the samples in the test set to test the network performance, that is, simulates the S-parameter characteristics of the metamaterials with different structures in the test set samples. In the experiment, for the 8*8 reference matrix of the metamaterial structure, we fixedly generate the 8*8 reference matrix containing 32-36 PEC patches as samples. The 16*16 metamaterial contains 32*4 PEC patches. We used the above-mentioned MATLAB and CST co-simulation method to obtain 1,000 sets of random samples, and then screened out metamaterial structures with frequency selective characteristics based on their S11 reflection coefficient maps, and then selected 20 sets that basically cover the 8-13 GHz frequency band. Metamaterial structures with good frequency selection characteristics at a certain frequency are used as samples to expand to form a data set.

In the constructed three-layer convolutional neural network, its convolution depth will affect the number of calculation parameters of the neural network, and it will also have a significant impact on the efficiency of the neural network.

In order to analyze the impact of different convolution depths on the convolutional neural network. We select 16-32-64, 32-64-128 and 64-128-256 three different depth models to train the neural network. The training time (unit: second) after a certain number of iterations is compared with the loss function. The situation is as follows:

![Figure 2. Comparison of training time consumption of different depth convolutional networks.](image)

When the convolution depth of the neural network is deepened, the number of parameters between networks will increase rapidly, and its training speed will gradually slow down. Among the three networks with different depths we have constructed, the network with a depth of 16-32-64 has a shallower network depth, which has the fastest training speed under different training times, with a depth of 32-64-128 Compared with the network with a depth of 16-32-64, the average increase in time is 30% under different training times. The network with a depth of 64-128-256 is compared with a network with a depth of 32-64-128 under different training times. The average increase in time is 70%.
Figure 3. Comparison of the influence of different depth convolutional networks on the loss function.

When the convolution depth in the neural network is deepened, the number of parameters between the networks will increase rapidly, and the loss function obtained by its training will be relatively small. Among the three different depth networks we constructed, the network with a depth of 64-128-256 has a deeper network depth, so it has the smallest loss function in various training times, but the network with a depth of 32-64-128 compared with the network with a depth of 64-128-256, its loss function is closer under different training times. The network with a depth of 16-32-64 has a loss compared to a network with a depth of 32-64-128 under different training times. The function has a relatively large gap. Based on the above comparison, it can be obtained that although the network with a depth of 16-32-64 has a faster training time, its loss function is far from other depths; The network with a depth of 64-128-256 can achieve a better minimization progress in the loss function, but the network with a relative depth of 32-64-128 does not have a great advantage in the loss function, and its time is far behind the other two deep networks; The network with a depth of 32-64-128 can obtain a relatively small loss function in a relatively short time, and can achieve a balance between time-consuming and loss function. Therefore, a network of this depth is selected when constructing the model. The characteristics of metamaterials are studied and calculated.

It can be seen from the foregoing that the pooling layer can reduce the size of the convolutional feature map, thereby reducing the number of parameters that need to be calculated, which is able to effectively avoid the over-fitting phenomenon in the neural network. In the actual use of neural networks, two commonly used pooling methods are maximum pooling and average pooling [15-20]. The pooling window size used in this design is 2*2. The two methods of maximum pooling and average pooling are used to pool the convolutional feature maps obtained above. The comparison of the loss functions after a certain number of iterations is as follows:
Figure 4. Comparison of the impact of different pooling methods on the loss function.

The above figure compares the impact of the maximum pooling and average pooling methods on the loss function, compared to the average pooling method. Maximum pooling maintains a better loss function in each training stage of the neural network designed in this design, so that the loss function converges more quickly, and better realizes the function of minimizing the loss function. Therefore, this design adopts the maximum pooling method in the calculation of metamaterial properties.

The activation function introduces a non-linear quantity to the neural network model, so that the neural network can handle complex data flexibly. We analysed the characteristics of several different activation functions of the convolutional neural network, processed the convolutional neural network constructed above with different activation functions. The comparison of the loss function after a certain number of iterations is as follows:

Figure 5. Comparison of the influence of different activation functions on the loss function.

As can be seen from the above figure, the Sigmoid function has a small convergence to the loss function in the initial training stage, which is able to achieve the convergence of the loss function after further training; The Tanh function is better than the Sigmoid function in the convergence of the loss function in the initial training stage. But in the later stage of training, its convergence rate is reduced, which cannot meet the function of minimizing the loss function; Compared with the Sigmoid function [21-28] and Tanh function [26-29], the ReLU function [30-33] guarantees the convergence of the loss
function during training, which is able to minimize the loss function quickly and effectively. Therefore, it is used in the neural network constructed by the metamaterial property calculation.

In addition, we use different optimizers (including Stochastic gradient descent method, SGD [34-38], momentum Stochastic gradient descent method [39-42], and adaptive moment estimation method, Adam [43-44]) to optimize the convolutional neural network constructed above. The SGD makes the loss function converge slowly, the convergence rate of the momentum SGD method in the early stage is similar to that of the Stochastic gradient descent method, but it can effectively converge the loss function in the later stage; The adaptive moment estimation method (Adam) can make the loss function converge quickly in the initial stage of neural network training, but it cannot converge further after the loss function converges to a certain extent in the later stage.

![Figure 6. Comparison of the influence of different optimizers on the loss function.](image)

Based on the characteristics of SGD and Adam in the convergence process of the loss function. In this experiment, the Adam is used in the initial training stage to make the loss function converge quickly and effectively. Then the momentum Stochastic gradient descent method is used to further optimize the initially convergent loss function, so that the loss function is gradually reduced to the required range.

4. conclusion
This article introduces the method of using deep learning in the calculation of metamaterial properties. The study expounds the joint simulation of MATLAB and CST to construct the metamaterial structure and generate the required data set. It also introduces the composition of the data set and its corresponding relationship, introduces the construction of the deep learning computing network in detail. This paper focuses on explaining the data processing process of the convolutional layer and the pooling layer in this design, and uses the built neural network Calculate the properties of metamaterials. At the same time, the influence of convolution depth, pooling method, activation function and optimization method on the loss function in the calculation of metamaterial properties is discussed, and the more suitable parameters and methods for each structure are pointed out. Finally, a metamaterial design method based on deep learning is proposed. Combining deep learning theory with metamaterial applications, comprehensively and comprehensively expounding the calculation method of metamaterial characteristic parameters based on deep learning, this study proposes a different idea from traditional electromagnetic calculation methods, novel and effective solutions to the complex calculation of metamaterial properties problem.
Acknowledgements
This research was funded by the 2020 Key Technology R & D Program of GuangDong Province, grant number of ZH01110405180056PWC, Zhuhai Technology and Research Foundation, grant number of ZH01110405180056PWC, Zhuhai Technology and Research Foundation, grant number of ZH22036201210034PWC, Zhuhai Basic and Application Research Project, grant number of ZH22017003200011PWC, 2017 key scientific research projects of Guangdong Province, grant number of 2017GXJK218 and the Science and Technology Innovation Committee of Shenzhen City, grant number of JCYJ2020109140820699.

References
[1] Wei M, Feng C and Yongming L 2018 Deep-learning-enabled on-demand design of chiral theory metamaterials ACS Nano 12(6): acs.nano.8b03659
[2] Itzik M, Micheal M, Achiya N et al. 2018. Plasmonic nanostructure design and characterization via Deep Learning Light: Science & Application 7(6):60
[3] Valley M T 2019 Metamaterials How Close Are We to a Klingon Cloaking Device or Harry Potter Invisibility Cloak? (No SAND2019-14245B) Sandia National Lab (SNL-NM) Albuquerque NM (United States)
[4] Zhang Q, Liu C, Wan X, Zhang L, Liu S, Yang Y and Cui T J 2019 Machine-learning designs of anisotropic digital coding metasurfaces Adv. Theory Simul. 2 1800132
[5] LeCun Yann Yosuhla Bengio and Geoffrey Hinton 2015 Deep learning nature 521: 436-444
[6] Qiu T, Shi X, Wang J, Li Y, Qu S, Cheng Q and Sui S 2019 Deep learning: a rapid and efficient route to automatic metasurface design Advanced Science 6(12): 1900128
[7] Nadell C C, Huang B, Malof J M and Padilla W J 2019 Deep learning for accelerated all-dielectric metasurface design Optics express 27(20): 27523-27535
[8] An S, Fowler C, Zheng B, Shalaginov M Y, Tang H, Li H & Zhang H 2019 A deep learning approach for objective-driven all-dielectric metasurface design ACS Photonics 6(12): 3196-3207
[9] John-Herpin A, Kavungal D, von Mücke L and Altug H 2021 Infrared Metasurface augmented by deep learning for monitoring dynamics between all major classes of biomolecules Advanced Materials 33(14): 2006054
[10] Liu Z, Zhu D, Rodrigues S P, Lee K T and Cai W 2018 Generative model for the inverse design of metasurfaces Nano letters 18(10): 6570-6576
[11] Sajedian I, Lee H and Rho J 2019 Double-deep Q-learning to increase the efficiency of metasurface holograms Scientific reports 9(1): 1-8
[12] Pestourie R, Mroueh Y, Nguyen T V, Das P and Johnson S G 2020 Active learning of deep surrogates for PDEs: application to metasurface design. npj Computational Materials 6(1): 1-7
[13] Mall A, Patil A, Tamboli D, Sethi A and Kumar A 2020 Fast design of plasmonic metasurfaces enabled by deep learning Journal of Physics D: Applied Physics 53(49) 49LT01
[14] Malkiel Itzik et al. 2018 Plasmonic nanostructure design and characterization via deep learning Light: Science & Applications
[15] Scherer D, Müller A and Behnke S 2010 Evaluation of pooling operations in convolutional architectures for object recognition In International conference on artificial neural networks (pp 92-101) Springer Berlin Heidelberg
[16] Li C et al. 2018 Hyperspectral remote sensing image classification based on maximum overlap pooling convolutional neural network Sensors 18(10): 3587
[17] Zhang Y D, Muhammad K and Tang C 2018 Twelve-layer deep convolutional neural network with stochastic pooling for tea category classification on GPU platform Multimedia Tools and Applications 77(17): 22821-22839
[18] Pagola M, Forcen J I, Barrenechea E, Fernández J and Bustince H 2017 A study on the cardinality of ordered average pooling in visual recognition *In Iberian Conference on Pattern Recognition and Image Analysis* (Springer Cham)

[19] Sharma T, Singh V, Sudhakaran S and Verma N K 2019 Fuzzy based pooling in convolutional neural network for image classification *In 2019 IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE) pp. 1-6

[20] Boureau Y L, Ponce J and LeCun Y 2010 A theoretical analysis of feature pooling in visual recognition. *In Proceedings of the 27th international conference on machine learning* (ICML-10) pp. 111-118

[21] Cybenko G 1989 Approximation by superpositions of a sigmoidal function *Mathematics Of Control Signals and Systems* 2(4): 303-314

[22] Kalman B L and Kwasny S C 1992 Why tanh: choosing a sigmoidal function. In [Proceedings 1992] *IJCNN International Joint Conference on Neural Networks* (IEEE) pp 578-581

[23] Barron A R 1993 Universal approximation bounds for superpositions of a sigmoidal function *IEEE Transactions on Information Theory* 39(3): 930-945

[24] Chandra P 2003 Sigmoidal function classes for feedforward artificial neural networks *Neural Processing Letters* 18(3): 205-215

[25] Kyurkchiev N. & Markov S. 2015. Sigmoidal functions: some computational and modelling aspects. Biomath Communications 1(2).

[26] Fan E 2000 Extended tanh-function method and its applications to nonlinear equations *Physics Letters A* 277(4-5): 212-218

[27] Parkes E J and Duffy B R 1996 An automated tanh-function method for finding solitary wave solutions to non-linear evolution equations *Computer Physics Communications* 98(3): 288-300

[28] Elwakil S A, El-Labany S K, Zahran M A and Sabry R 2002 Modified extended tanh-function method for solving nonlinear partial differential equations *Physics Letters A* 299(2-3): 179-188

[29] Abdou M A and Soliman A A 2006 Modified extended tanh-function method and its application on nonlinear physical equations *Physics Letters A* 353(6): 487-492

[30] Agarap A F 2018 Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375

[31] Eckle K and Schmidt-Hieber J 2019 A comparison of deep networks with ReLU activation function and linear spline-type methods *Neural Networks* 110: 232-242

[32] Schmidt-Hieber J 2020 Nonparametric regression using deep neural networks with ReLU activation function *The Annals of Statistics* 48(4): 1875-1897

[33] He J, Li L, Xu J and Zheng C 2018 Relu deep neural networks and linear finite elements. *arXiv preprint arXiv:1807.03973*

[34] Bottou L 2010 Large-scale machine learning with stochastic gradient descent *In Proceedings of COMPSTAT'2010* (Physica-Verlag HD) pp 177-186

[35] Shamir O and Zhang T 2013 Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes *In International conference on machine learning* PMLR pp 71-79

[36] Qian Q, Jin R, Yi J, Zhang L and Zhu S 2015 Efficient distance metric learning by adaptive sampling and mini-batch stochastic gradient descent (SGD) Machine Learning 99(3) 353-372

[37] Wijnhoven R G & de With P H N 2010 Fast training of object detection using stochastic gradient descent *In 2010 20th International Conference on Pattern Recognition* (IEEE) pp 424-427

[38] Liu Y, Gao Y and Yin W 2020 An improved analysis of stochastic gradient descent with momentum *arXiv preprint arXiv:200707989*

[39] Cutkosky A and Orabona F 2019 Momentum-based variance reduction in non-convex sgd. *arXiv preprint arXiv:1905.10018*
[40] Wang B, Nguyen T M, Bertozzi A L, Baraniuk R G and Osher S J 2020 Scheduled restart momentum for accelerated stochastic gradient descent. arXiv preprint arXiv:2002.10583

[41] Khan Z A, Zubair S, Alquhayz H, Azeem M and Ditta A 2019 Design of momentum fractional stochastic gradient descent for recommender systems IEEE Access 7 179575-179590

[42] Cutkosky A and Mehta H 2020 Momentum improves normalized sgd In International Conference on Machine Learning (PMLR) pp 2260-2268

[43] Zhang Z 2018 Improved adam optimizer for deep neural networks In 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS) pp 1-2

[44] Bock S and Wei M 2019 Non-convergence and limit cycles in the adam optimizer In International Conference on Artificial Neural Networks (Springer Cham) pp 232-243

[45] Bera S and Shrivastava V K 2020 Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification International Journal of Remote Sensing 41(7): 2664-2683