

2106269-Kaicheng Li-MSc Project

by Kaicheng Li

Submission date: 07-Sep-2022 09:36PM (UTC+0100)

Submission ID: 185788604

File name: 32969_Kaicheng_Li_2106269-Kaicheng_Li-MSc_Project_430701_809830639.docx (1.73M)

Word count: 15012

Character count: 83337



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

**Optical response prediction of
optical structures
based on deep learning**

by
Kaicheng Li (2106269)

Supervisor: Gordon Dickers

Project submitted as part of the requirements for the
award of MSc Software Engineering and Artificial
Intelligence

September 2022

DECLARATION

I, Kaicheng Li declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed :

李开城 LI KAICHENG

Date : 2022.9.7

FORM OF CONSENT

I, Kaicheng Li hereby consent that my Project, submitted in candidature for the MSc Software Engineering and Artificial Intelligence degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed :

李开城 LI KAICHENG

Date : 2022.9.7

ABSTRACT

Novel optical devices capable of controlling optical flow down to the nanoscale have become an extremely dynamic area of research. For example, metasurfaces containing subwavelength structures can exhibit superior properties, opening the possibility for breakthroughs in widely used technologies such as imaging, sensing, and polarization control.

In traditional design methods of optical structures, attempts such as full-wave simulations require significant time and computational resources even for experienced researchers. Deep learning algorithms have been applied to the learning forward prediction problem in light, and have shown good results. Image processing also plays an important role in these studies to convert observed measurements into desired property values, an information processing process similar to that of computer vision and computer imaging. In recent years, deep learning algorithms, in which machines automatically learn patterns from large amounts of data, have shown certain advantages in optics.

Optical structure design strategies incorporating deep learning are expected to increase computational speed by orders of magnitude, and artificial neural networks are a potential solution to bypass the complex design process, allowing us to directly predict the optical response of optical structures or perform inverse design efficiently and accurately. The first is to replace forward simulations in iterative optimization with artificial neural networks. Forward predictive models are used to determine the optical response of geometric or material parameters, which can replace full-wave simulations.

The research content of this paper is the prediction of optical response of optical structures based on deep learning. The optical structure is represented in the form of an image, and the deep learning algorithm is used to predict the optical response of the optical structure to the incident electromagnetic wave, which is the prediction process of the system from the input optical structure image to the output optical response. By continuously adjusting the parameters of the initial neural network model, the model achieves relatively good results and can make accurate instance predictions.

Key Words: optical response prediction; deep learning; CNN

CONTENTS

1. INTRODUCTION.....	1
2. Literature Review	4
2.1. Micro-nano structured metasurfaces.....	4
2.2. Advantages and challenges of applying deep learning to optics	8
2.3. Design strategy of micro-nano structure based on neural network	12
3. Design strategy of micro-nano structure based on neural network	22
4. Data collection	25
5. Design and Implementation	31
5.1. Initial Design	31
5.2. Choice Of Model Parameters.....	33
5.3. Choice Of Hyper-Parameters	34
5.4. Intermediate Results.....	34
5.5. Final Model	35
6. Results and Discussion	37
7. Conclusions and Recommendations	39
REFERENCE	43
APPENDIX: PROGRAMMING CODE	46

1. INTRODUCTION

In recent years, metamaterials and metasurfaces have been widely used in microwave, terahertz and optical fields due to their unique electromagnetic response characteristics. Different from natural materials in nature, the optical response in metamaterials and metasurfaces is mainly determined by the shape, size, and arrangement of the microunits that make up their structures, so through rational structural design, people can almost freely manipulate light. Adjust its phase, amplitude and polarization properties. Compared with metamaterials with three-dimensional structures, two-dimensional metasurfaces have become a hot spot in the field of nanophotonics because of their simple fabrication process, small size, high integration, and flexible design. Early research work on metasurfaces mainly focused on surface plasmon devices based on metal materials. With the blue-shift of the operating band, especially to the near-infrared and visible light bands, a series of negative effects caused by the ohmic loss of metals are difficult to ignore. , which seriously affects the efficiency of the device.

At this time, researchers began to turn their attention to low-loss dielectric materials, and the research on dielectric metasurfaces began to become a key direction in the field of metamaterials. The design principle of dielectric metasurface devices is based on Mie scattering theory, which can be roughly divided into two categories according to their functions. By introducing a phase gradient on the incident surface of the light field, the functions of beam deflection, focusing and vortex light are realized. The other type is a resonant metasurface that mainly enhances the electromagnetic field strength in the metasurface, and uses the group resonance effect of all resonant units to form high-quality resonant modes and electromagnetic hot spots to enhance the interaction between light and matter in the structure. . In nanophotonics, light is used as a carrier for information transmission, and flexible processing and manipulation of light fields need to be realized with the help of some media that have special responses to light fields, such as materials with specific optical properties and materials constructed from these materials. Special optical structure. The most ideal optical materials should have the following properties: high refractive index in the optical frequency band, responsive to the electric and magnetic

fields of incident light, supporting optical chirality and anisotropy, and being able to confine the optical field in the wavelength range region. , regulating its amplitude, phase, and polarization information, etc. 1.

However, in nature, natural materials with all of the above properties are very rare. It is precisely based on the demand for new materials that scientists try to use the artificially fabricated extremely small nanostructure units as a kind of "meta-atom" according to a certain lattice law to construct a new artificial structure. Materials, also known as metamaterials today. From a macroscopic point of view, a metamaterial is an artificial composite material. From a microscopic point of view, a metamaterial is composed of a series of artificial structural units arranged periodically or non-periodically. The size of these artificial structural units is larger than that of real atoms. The lattice is much larger, but much smaller than the wavelength of the incident light, so its macroscopic properties can be described by equivalent medium theory [21]. Different from natural materials in nature, the properties of metamaterials are not only related to the chemical composition of the material itself, but also depend on the combination and arrangement of these artificial structural units. We know that it is very difficult to change the lattice structure of atoms in materials to obtain new electromagnetic response characteristics, but it is very easy to change the artificial structural units that make up metamaterials. Spatial distribution can enable metamaterials to achieve light-field response characteristics that many natural materials do not have in nature. The emergence of metamaterials breaks the limitations of the nature of materials, allowing people to manipulate, transmit and process light waves more freely.

Two-dimensional metamaterials, also known as metasurfaces, are ultrathin two-dimensional array structures composed of sub-wavelength-sized artificial structural units arranged in a plane according to certain rules. With their special response to the incident electromagnetic field, these sub-wavelength-sized structural units can act as secondary wave sources to control the amplitude, phase, and polarization properties of the incident light field. By rationally designing the structure and period of the subwavelength unit, the wavefront regulation of the optical field, anomalous reflection and transmission, and local field enhancement, etc. can be realized, resulting in the birth of a number of new optical

applications and devices, including flat lenses [58, 58] 59], optical vortex polarizers [60], magnetic mirrors [61-63], highly directional Huygens sources [64-74], modulation of spontaneous emission, metasurface plasmon stimulated emission amplification lasers [75,76] As well as sensing based on electromagnetic field hotspots [77], etc. Compared with metamaterials with three-dimensional structures, metasurfaces effectively avoid the problems of bulky volume, difficult preparation and high loss caused by the concept of equivalent parameter gradients in metamaterials. It is also more flexible and simpler. The flexible manipulation of light fields by metasurfaces greatly broadens researchers' ideas and directions for material design, and provides an important way for people to customize required optical devices and explore new functional optical devices. Most of the early researches on metasurfaces were based on metal materials. However, due to the inherent ohmic loss of metals, in the microwave and terahertz bands, the loss of metals is low, and the performance of the device can be guaranteed to a certain extent.

Deep learning has become the dominant method for artificial intelligence to solve complex data-driven problems. Initially applied almost exclusively to computer science fields such as image analysis and natural language processing, deep learning has rapidly entered a wide range of scientific fields including physics, chemistry, and materials science. Recently, deep neural networks have been introduced into the field of nanophotonics as an efficient method to obtain nonlinear mappings between the topology and composition of arbitrary nanophotonic structures and their associated functional properties. Fast processing speeds enabled by wavelength-scale light-matter interactions are propelling photonic chips as competitive candidates for next-generation processing units for large-scale deep learning inference. Deep neural networks are a general class of machine learning algorithms that utilize serial stacking of non-linear processing layers to achieve the capture and modeling of highly non-linear data relationships. Because neural networks are able to model physics-based nonlinear relationships, such as the relationship between the geometry of a photonic system and its electromagnetic response, they offer a whole new perspective on forward and inverse problems.

2. Literature Review

2.1. Micro-nano structured metasurfaces

At present, one of the most studied and mature directions in dielectric metasurfaces is wavefront control, as shown in Figure 1-5. Similar to the design idea of metal gradient metasurface, this kind of device mainly realizes the modulation of the wavefront of the light field by adjusting the phase and polarization of the light field scattered by the resonant element in the metasurface. Compared with metal materials, dielectric metasurfaces have lower absorption loss, higher transmission and diffraction efficiencies, and thus better device performance. In the past few years, there have been numerous reports in the literature on the application of wavefront-controlled dielectric metasurfaces, including planar lenses [58, 84], optical axicons [8], beam deflection [80], optical vortex polarizers [87], eliminating Chromatic aberration lenses [88, 89], beam shaping [90, 91] and polarizing beam splitter [92], etc. In 2016, the Capasso research team prepared a high aspect ratio two-dimensional array metasurface lens on titanium dioxide material [93], with a numerical aperture of up to 0.8, which can achieve diffraction-limited beam focusing in the visible light band, and the efficiency can reach up to 86%. The resolution reaches the sub-wavelength scale, the magnification is 170 times, and the imaging quality can be compared with the most advanced commercial lenses. This research achievement was named one of the top ten scientific and technological advances in 2016 by the journal Science. Immediately after that, gradient metasurfaces for wavefront regulation have attracted more attention, and related research results have been published in various journals. In addition, the applications based on the internal Mie resonances of nanostructured units in dielectric metasurfaces are also very extensive, including various types of perfect mirrors [94, 95] and magnetic mirrors [62, 96, 97, etc.], these optical devices. Its working principle is fundamentally different from that of traditional mirrors. The subwavelength thickness metasurface composed of silicon nanoresonators can achieve 100% reflectivity at the resonance wavelength in the near-infrared band.

Through the above-mentioned overview of the development history and research status of metamaterials and metasurfaces, we find that low-loss dielectric metasurfaces

are gradually becoming the core hotspot in nanophotonics research. According to the working principle and different functions, dielectric metasurfaces can be roughly divided into two categories: resonant and non-resonant (mainly wavefront-controlled metasurfaces). Wavefront control metasurfaces, also known as flat optics devices, are usually composed of a series of sub-wavelength-sized structural units arranged in a regular period or aperiodic, which can realize the control of the beam wavefront, including the amplitude, polarization, and phase distribution of the light. In the past few years, many breakthrough results have been achieved in this field, and many works have been rated as the top ten scientific and technological progress by "Science" magazine. The structural units in resonant metasurfaces are usually arranged uniformly periodic, and high quality factor resonant modes are formed through the interference of electromagnetic dipoles in the structure and the group resonance effect, such as Fano resonance, quasi-continuous domain bound states (quasi-BIC) mode, which can realize the functions of sensing, slow light, fluorescence enhancement and enhancement of various nonlinear effects. Among them, the research on dielectric-based resonant metasurfaces and surface-emitting devices is still in its infancy, using the interference effect between electromagnetic dipoles and multipoles in high-refractive-index dielectric resonators and the near-field between resonators. Coupling can achieve local enhancement of the electromagnetic field inside the structure and resonant modes with high quality factor. Furthermore, the highly localized electromagnetic field inside the resonant structure will strengthen the process of interaction between light and matter, thereby regulating the optical radiation of the system, and realizing the functions of luminescence enhancement and directional emission.

According to the working principle and different functions, dielectric metasurfaces can be roughly divided into two categories: resonant and non-resonant (mainly wavefront-controlled metasurfaces). Wavefront control metasurfaces, also known as flat optics devices, are usually composed of a series of sub-wavelength-sized structural units arranged in a regular period or aperiodic, which can realize the control of the beam wavefront, including the amplitude, polarization, and phase distribution of the light. In the past few years, many breakthrough results have been achieved in this field, and many works have been rated as the top ten scientific and technological progress by "Science" magazine. The

structural units in resonant metasurfaces are usually arranged uniformly periodic, and high quality factor resonant modes are formed through the interference of electromagnetic dipoles in the structure and the group resonance effect, such as Fano resonance, quasi-continuous domain bound states (quasi-BIC).) mode, which can realize the functions of sensing, slow light, fluorescence enhancement and enhancement of various nonlinear effects. Among them, the research on dielectric-based resonant metasurfaces and surface-emitting devices is still in its infancy, using the interference effect between electromagnetic dipoles and multipoles in high-refractive-index dielectric resonators and the near-field between resonators. Coupling can achieve local enhancement of the electromagnetic field inside the structure and resonant modes with high quality factor. Furthermore, the highly localized electromagnetic field inside the resonant structure will strengthen the process of interaction between light and matter, thereby regulating the optical radiation of the system, and realizing the functions of luminescence enhancement and directional emission.

The perceptron that appeared in the 1950s can be said to be the beginning of artificial neural networks. Since then, artificial neural network research has been actively carried out in the 1960s. However, in 1969, a book called *Perceptrons* was published, which revealed the fatal limitations of perceptrons and contained proofs. Then, in the 1970s, we entered a dark age where artificial neural networks were ignored by many academics. In the 1980s, artificial neural network research began to receive attention again. This can be attributed to the fact that a method was devised in 1986 to learn multilayer artificial neural networks by applying error backpropagation. In this era, Deep Neural Networks (DNN), Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) have been developed. In the 1990s, advanced forms of artificial neural networks appeared. The advanced recurrent neural network LSTM was released in 1997, and the advanced convolutional neural network LeNet-5 was released in 1998. In the 2000s, artificial neural networks started to gain attention in the name of deep learning. Gartner, a well-known information technology research organization, ranks deep learning as a top ten strategic technology. to tune. The proposal of this deep learning method immediately aroused great repercussions in the academic circle. Many world-renowned universities represented by

Stanford University and the University of Toronto have invested huge human and financial resources to conduct research in the field of deep learning. And then quickly spread to the industry. In 2012, in the famous ImageNet image recognition competition, the team led by Jeffrey Hinton adopted the deep learning model AlexNet to win the championship. AlexNet uses ReLU activation function to fundamentally solve the problem of gradient disappearance, and uses GPU to greatly improve the operation speed of the model. In the same year, the deep neural network-DNN technology co-led by the famous Professor Wu Enda of Stanford University and the world's top computer expert Jeff Dean has achieved amazing results in the field of image recognition, and successfully reduced the error rate from 26% to 26% in the ImageNet evaluation. 15%. The emergence of deep learning algorithms in the world competition has once again attracted the attention of academia and industry in the field of deep learning.

With the continuous progress of deep learning technology and the continuous improvement of data processing capabilities, in 2014, Facebook's DeepFace project based on deep learning technology, the accuracy rate of face recognition has reached more than 97%, which is almost the same as the accuracy rate of human recognition. no difference. Such results once again prove that the deep learning algorithm is unrivaled in image recognition.

In 2016, when AlphaGo, developed by Google based on deep learning, defeated the world's top Go player, Li Shishi, with a score of 4:1, the popularity of deep learning was immortal. Later, AlphaGo played against many world-class Go masters one after another, and they all won. This also proves that in the game of Go, robots based on deep learning technology have surpassed humans.

Problems to be solved by deep learning Classification Finding the categories of data is a problem. An example is the problem of classifying cars into one of the following categories based on characteristics such as length, width, height, wheel size, and engine horsepower: light car, semi-medium car, medium car, and large car. Clustering This is a matter of grouping instances of data. In other words, it is a set of data instances with similar characteristics. For example, look at characteristics such as length, width, height, wheel size, and engine horsepower of a car, and group similar instances together. Human

intervention is required to determine grouping results as compact cars in group 1, semi-compact cars in group 2, etc.

The problem of regression is to find the value of incomplete data. For example, in a data instance, if you know characteristics such as the width, height, wheel size, and engine horsepower of a car, but not the length, you can predict the length characteristics for an instance consisting of incomplete data. Values on other data instances.

Each problem can be solved from the perspective of machine learning, and the learning methods can be divided into supervised learning, unsupervised learning and reinforcement learning. Supervised Learning It is a method of training on labeled data, mainly dealing with classification and regression problems. It's relatively easy to learn and effective because having labels means having the right answer. However, tagging by data is often expensive. The cost can be money, time, or in many cases both. The training data is massive, from hundreds of millions to tens of millions, or even more. Sometimes it is impossible for humans to label this data. How unsupervised learning trains on unlabeled data. Mainly used to group data or analyze the characteristics of data. Since unsupervised learning is analyzing data without correct answers, there is no need to label the data. This means less cost to prepare data. However, in many cases, human intervention is required to determine the meaning of the clusters being analyzed.

2.2. Advantages and challenges of applying deep learning to optics

Photonic circuits beat their electronic counterparts regarding processor size, long vector lengths, and low-precision operation. Modern electronic computers have programmability, large-scale, high-yield manufacturing capabilities, the ability to implement in three dimensions, integrated signal recovery, and gain, and reliable memory solutions. High accuracy is another feature of today's advanced electronic systems that photonic systems still lack. The level of accuracy required by AI systems performing inference tasks varies. One potential use of optics and photon processing is the quick processing of scene information via optical inference with incoherent light, even under ambient illumination conditions. It finds widespread application in fields including artificial intelligence (AI), robotics, and driverless cars.

Given the scope and complexity of AI, we require processors with increased throughput at reduced power consumption. There is much room for adaptation in optical computing systems, especially in visual computing. Despite fifty years of research and development, current general optical photonic computing systems are still not ready for widespread deployment in actual applications. As GPU speeds and supervised machine learning methods have increased, so too has the efficiency with which deep neural networks (DNNs) can train with massive amounts of data. However, this complicates the use of DNNs on edge devices (cameras, autonomous driving, robotics, IoT devices, etc.). Optical neural networks (ONNs) and photonic circuits offer numerous benefits over traditional electrical computer systems, including high levels of parallelism, low power consumption, and minimal hardware requirements for operation. There are two main parts to the DNNs framework, and they are training and inference. The reasoning part of the system improves both all-optical and hybrid electro-optical systems. Convolutions, Fourier transformations, random projections, and other operations can all be carried out using linear optics components with extremely low power consumption and lightning-fast response times.

It is also possible to construct processing power on a light field as it travels across space or another medium, a technique known as a photonic circuit. Convolutional neural networks (CNNs) are most visual computing applications' preferred neural network design. One of their fundamental building blocks is the convolution of a wavefield with a complex-valued kernel to explain free-space propagation. In addition, it was shown that the complex medium is a visual representation of a neural network. The connection weights are random matrix coefficients, and the nonlinearity is a transition of intensity during camera detection, enabling direct classification without imaging. Computational imaging, phase retrieval, and reservation computing are the only examples of problems that could benefit from this mathematical recalculation of light propagation.

Given the scope and complexity of AI, we require processors with increased throughput at reduced power consumption. There is much room for adaptation in optical computing systems, especially in visual computing. Despite fifty years of research and development, current general optical photonic computing systems are still not ready for

widespread deployment in actual applications. As GPU speeds and supervised machine learning methods have increased, so too has the efficiency with which deep neural networks (DNNs) can train with massive amounts of data. However, this complicates the use of DNNs on edge devices (cameras, autonomous driving, robotics, IoT devices, etc.). Optical neural networks (ONNs) and photonic circuits offer numerous benefits over traditional electrical computer systems, including high levels of parallelism, low power consumption, and minimal hardware requirements for operation. There are two main parts to the DNNs framework, and they are training and inference. The reasoning part of the system improves both all-optical and hybrid electro-optical systems. Convolutions, Fourier transformations, random projections, and other operations can all be carried out using linear optics components with extremely low power consumption and lightning-fast response times.

When building cameras and lenses, optical engineers have a wide variety of options available to them, such as the degree to which individual sensor pixels are sensitive to various portions of the electromagnetic spectrum (PSFs). For camera encoder-decoder systems, some of the design challenges include optical as well as end-to-end image processing optimization. The difficulty that comes with the process of building application-specific imaging systems is in the designing and optimizing the engineering potential of such devices. These computational "depth" cameras can increase the effectiveness of more sophisticated loss functions, such as those used for object or picture categorization, when trained offline. One example of such a function is the classification of images. This training strategy, similar to classic approaches to computer vision, optimizes the parameters of other differentiable algorithms or the weights of neural networks in order to achieve optimal performance.

Approaches based on data-driven deep learning are utilized in order to find a solution to the optical microscopy inverse problem. DNNs can provide a quick framework for picture reconstruction and augmentation following a single training session, without the need for iterations, parameter tuning, or physical forward models. This is made possible by their ability to learn quickly.

Deep learning has a significant deal of potential use in the field of digital image processing. This does not imply that we have given up on the tried-and-true approaches of computer vision that were developed prior to the introduction of deep understanding. Researchers at the Irish Terry Institute of Technology developed an analysis that compares and contrasts the two different ways. Vision applications are becoming increasingly commonplace as a result of developments in deep learning, rapid app iteration, and enhanced device capabilities (such as CPU speed, memory size, energy usage, image sensor resolution, and optics). Deep learning can assist engineers enhance accuracy in areas such as image classification, semantic segmentation, object recognition, and simultaneous localization and map generation when compared to classic CV methods (SLAM). Deep learning applications are able to analyze the large volumes of video data currently present in systems because they use learnt neural networks rather than coded neural networks. This eliminates the requirement for human interpretation and correction of the data.

In contrast to CV approaches, the CNN models and frameworks of deep learning may be retrained with datasets that are particular to the application at hand, making it a highly flexible option. Traditional approaches to creating CVs have the disadvantage of asking you to select the information from each photograph that is most pertinent to your application. In addition, as the number of categories grows, the difficulty of feature extraction grows as well. To identify which characteristics most effectively distinguish each target audience, the CV engineer will need to make use of their judgment and carry out a large amount of trial and error. It will be necessary for the CV engineer to make modifications to the specifications of each feature in order to bring them into compliance with a diverse set of extra criteria. Learning from beginning to end is made possible by assigning the necessary class to each image that is stored in the computer's image collection. As a consequence of this, the neural network that is a part of the deep learning model "trains" on the input data in order to recognize patterns that are hidden within the image category. After this, the model automatically extracts the characteristics that are most significant and illustrative of the target category. Even though there are compromises to be made in terms of the amount of processing power and the amount of time spent training, it is generally accepted that DNNs perform better than older approaches. Deep learning is used in every one of the

industry's most successful solutions, which have all just been put into operation. Because of this, the knowledge and expertise needed to manually extract features have been supplanted by the knowledge and experience required to iterate using deep learning architectures. This is because of the way that deep learning works.

The CV industry has been severely impacted by the changes made to CNN, which have also made it easier to create targets. The increase in processing capacity and the availability of training data are inextricably linked to this phenomenon and cannot be considered separately. More than 3,000 citations have been made to the paper "ImageNet Classification Using Deep Convolutional Neural Networks," which is evidence of the growing interest in these kinds of architectures for CV.

Convolutional neural networks (CNNs) use convolutional kernels (filters) to identify picture features like edges. To recognize certain features, data scientists train convolution kernels and weighted matrices. The primary purpose of a convolutional neural network (CNN) is to search for characteristics necessary for detection by spatially convolving a kernel onto an input picture. As part of its feature detection process, a neural network performs a convolution operation, which involves computing the dot product of the convolution kernel and the region it overlaps with the input picture (the area of the original image the seed is looking at is called the receptive field).

To facilitate the learning of the convolution kernel weights, researchers included a bias term in the output of the convolutional layer, which is then fed into the nonlinear activation function. Sigmoid, TanH, and ReLU are just a few examples of popular nonlinear functions used as activation functions. Both the nature of the data and the task at hand in terms of categorization will guide the selection of an appropriate activation function. For example, ReLU more accurately depicts biology (whether neurons in the brain are active). Due to its improved robustness against the vanishing gradient problem and its ability to construct sparser and more effective representations, ReLU will outperform competing methods in image recognition applications.

2.3. Design strategy of micro-nano structure based on neural network

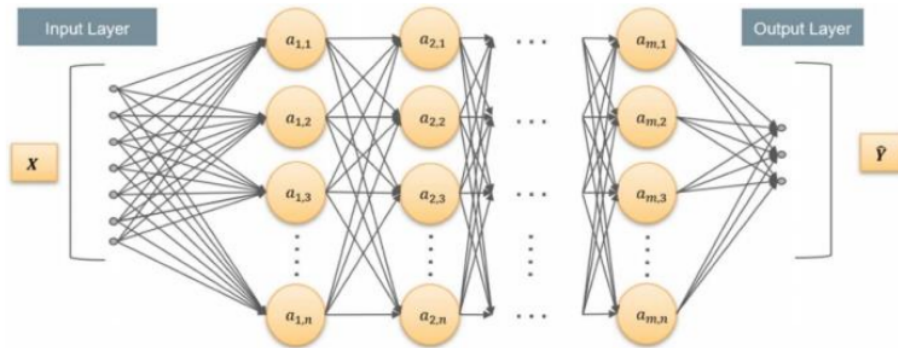
In recent decades, photonics has brought about transformation in a variety of scientific and commercial fields. Even for experienced researchers, developing sophisticated micro- and nanostructures and devices requires a significant investment of time. On the other hand, the construction of subwavelength structures in a rational manner enables us to manipulate light in any way we choose. Artificial neural networks, a subset of artificial intelligence, offer a promising alternative to conventional design approaches by allowing us to execute inverse design quickly and inexpensively or immediately predict the optical response of micro-nanostructures [10]. This makes artificial neural networks a subset of artificial intelligence that offers a promising alternative to conventional design approaches.

The generation of micro-nano structures by means of an artificial neural network (ANN) is a revolutionary approach that has proven to be highly effective [11-19]. Artificial neural networks, similar to the nonlinear brain conduction processes that occur naturally in the human body, have the ability to build implicit connections between inputs (such as geometric or material features) and outputs (such as optical responses). It is feasible to circumvent time-consuming and difficult design processes like numerical simulations and optimization with the help of artificial neural networks that have been suitably trained. This is made possible by the advent of deep learning. The majority of ANN models that are used in the process of producing micro- and nanostructures do both forward prediction and reverse design. The bold prediction network can be used to determine the optical response of a group of geometrical or material features rather of running full-wave simulations. This saves time and effort. The process of developing networks that are capable of properly replicating ideal structures based on a particular visual reaction is frequently more crucial and difficult. One of the most significant benefits offered by the ANN model is the ability to produce results quickly. In contrast to analog simulations [20-22], a forward prediction model that has been properly trained can yield meta-atom spectra in a matter of milliseconds. In addition, artificial neural networks are superior to rigorous simulations when it comes to locating links between physical structures and optical responses that are counterintuitive and not unique.

Data-driven artificial neural networks need huge training sets of geometric/material parameters and optical responses in order to solve the problem of micro nanostructured design. After the ANN model has been successfully trained on the training dataset, the test set, which represents a real-world situation, can be used to evaluate the model's effectiveness. While maintaining the same design structure, it should incorporate the newly acquired data.

The tasks that forwarding prediction networks are tasked with completing are typically carried out using these same four phases. The first phase involves the generation of a large pool of input structures and optical responses through the use of virtual experiments. It is vital to keep in mind that the output of neural networks is dependent on the quantity and quality of data to which they have access. This is something that must not be forgotten. Some research endeavors to provide one-of-a-kind training data with more dimensions [23], and does so by incorporating new data into the training model. This is done in an effort to improve the overall quality of the dataset that is used for training. After that, a synthetic neural network is built by employing a particular network design, such as a fully connected layer (FCL)-based neural network or a convolutional neural network. This allows the network to accurately simulate natural neural networks (CNN). Following this, the training dataset is continuously supplied into the network so that its weights and biases can be adjusted. Finally, well-trained ANNs have the ability to make predictions about the outcomes of input structures that are not included in the datasets used for training and testing.

The challenge of building backwards can be easily circumvented by switching the order of the input and output nodes of a network that otherwise retains its original structure. Nevertheless, in order to overcome certain roadblocks, you will need to employ more involved strategies and procedures. Ahead-of-Time Optical Response Prediction in Micro- and Nanostructures; Prediction in advance by employing a neural network that is completely connected.



As shown in Figure 1, a fully connected layer-based neural network consists of an input layer, a hidden layer, and an output layer. Beginning with the output of the preceding layer, each subsequent layer feeds it into a tensor operation whose parameters can be adjusted according to biases and weights. Since this is a linear process from start to finish, all input-output connections can only be simple linear transformations, making it impossible to construct complex relationships. Because the nonlinear "activation" process of the neuron is so important here, the activation function should be nonlinear to enable the ANN to approximate any function. In addition to spectrum prediction [24-25], reverse design makes use of ANNs with fully linked layers to accomplish a variety of tasks and enhance practical applications.

Given the data-driven nature of deep learning, the performance of artificial neural networks is heavily dependent on the training set, and when input deviates from the training set, prediction loss may increase. Thus, the difficulty of deep learning-assisted reverse engineering is in transferring the capabilities of artificial neural networks to datasets that are vastly unlike from the training data. For similar but distinct physical conditions, it is customary to design an entirely new training regimen. One technique to quicken the training of deep learning models in this case is to reduce the amount of computing data required for training. Y. Qu et al. [32] suggested a strategy for transmitting information across various physical settings.

The second use of forward prediction with a convolutional neural network.

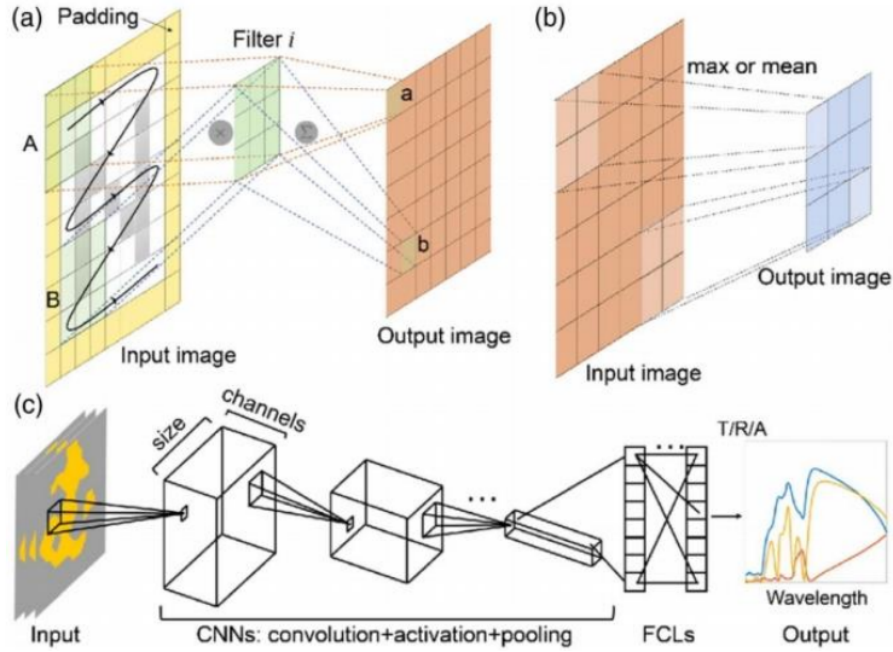
On sometimes, we get input that is particularly complex in the visual realm. Parameterizing the desired design and structure can be difficult if the structure of interest comprises a variety of fundamental shapes [33, 34] or is free-form [35, 36]. Converting 3D

models into 2D photos might be advantageous in some situations since it provides designers with more freedom of movement.

If FCL-based models are still being utilized, preprocessing is required to control the input images. Linear embeddings, like principal component analysis and random projection, can be used to reduce the dimensionality of the image and facilitate feature extraction. However, performance can suffer when these transformations disturb the connection between vertically adjacent pixels in a single image or when they remove information essential to the image's integrity. Another issue is the possible high size of the input dimension. Although the input dimension is often in the tens or hundreds for parametric input, a vectorized image with 6464 by 6464 pixels will result in a 4096-dimensional input vector.

Convolutional neural networks (CNN) are the most effective tool for dealing with the current scenario. The CNN takes an image as input and applies a series of filters to it along the horizontal and vertical axes without performing any preprocessing. To each image's sub-region, a filter's weight is multiplied with the sub-value region's to calculate the convolution operation.

Figure 2 shows the convolution, pooling operation and traditional CNN Schematic diagram of the workflow.



Because of their capacity to maintain local sections of the input, CNNs are commonly used in 2D image processing. Furthermore, CNNs can theoretically work in arbitrary dimensions, significantly broadening the range of structures that can be investigated. Plasmonic structures, for example, have been intensively studied in recent decades due to their unusual optical properties and far-reaching impact on other fields, making it all the more important and intriguing to establish a link between the design of plasmonic structures and the corresponding optical responses. In their 2019 paper [37], I. Sajedian et al. employed a convolutional neural network (CNN) and a recurrent neural network (RNN) to forecast the near-infrared absorption spectra of advanced plasmonic devices. CNNs help with visual feature extraction, whereas RNNs with gated recurrent unit layers handle spectral prediction.

Reverse Engineering

In the days before the development of algorithms for deep learning, the inverse design of micro- and nanostructures was typically performed through a combination of intuitive guesswork and laborious fine-tuning. The process of combining heuristic and numerical simulation approaches requires a significant investment of both time and

resources. It may take anywhere from a few hundred to several thousand iterations until the entire search of the parameter space for each new optimization objective has been completed. In recent years, there has been significant development achieved in fields such as deep learning, which has the potential to increase processing speed by orders of magnitude. [38] Research has shown that artificial neural networks are capable of solving both the forward simulation problem as well as the inverse problem that arises in iterative optimization.

Approaches based on deep learning are very effective at approximating solutions to particular issues, such as the optical response of micro- and nanostructures. In order to speed up the process of reverse engineering, a "forward neural network" could be employed as a proxy model instead of the more standard simulation methodologies.

In contrast to numerical simulations, artificial neural networks are able to offer predictions in a matter of milliseconds after being trained. Heuristic reverse engineering is still an iterative process that requires thousands of model calls and intermediate processing steps. This is because, despite the fact that replacing traditional physical simulations with artificial neural networks can significantly speed up the process, heuristic reverse engineering remains a tedious endeavor.

The use of forward artificial neural networks in evolutionary optimization as a surrogate model increases the possibility of the surrogate model converging to the singularity, and as a result, this method is not the most computationally efficient one. As a consequence of this, researchers in the academic world are working toward the goal of developing reverse engineering techniques that are exclusively based on artificial neural networks. The "one-to-many" conundrum, which happens when numerous methodologies may achieve the same design goal, is the most challenging impediment that reverse engineers need to overcome. The training of a forward solver network, which ultimately leads to the creation of a standard stable reverse-engineered network, is where the tandem network design gets its start. The evaluation of FCL is based on the actual reverse engineering network's training, despite the fact that it has been extensively applied to reverse engineering issues in tandem network topologies (the generator).

When training ANNs for inverse design, one of the challenges that can arise is the fact that different structures can yield optical responses that are mostly comparable to one another. If the training dataset comprises examples that have contradictory optical responses but distinct geometries, the neural network may have difficulty converging since the one-to-many mapping is not unique. At this particular data point, the mathematical gradient of the function that is being represented by the artificial neural network is exceptionally robust. The authors recommend a backwards-engineered network design as a solution to this problem. This architecture would be built on a forward-trained model and layers of projected spectra of fully-connected structures. In order to solve the one-to-many problem in a backwards manner, the network needs to be able to produce predictions that eventually converge on a frequency-spanning structure.

One further method for solving one-to-many puzzles is called the Conditional Generative Adversarial Network, or CGAN [40-43]. In addition to the design goal, CGAN will accept a "latent vector" as an input from the user. It is a series of arbitrary numbers, and the network adapts to new situations by taking advantage of the latent vector's numerous values. In addition to its utilization of latent vectors, CGAN is characterized by the loss function that it employs. It is necessary to perform fine-tuning on both the GAN network and the training hyperparameters in order to achieve convergent learning with a GAN. Equal amounts of effort must be spent training generators and discriminators for the adversarial loss function in order to achieve the best possible outcomes.

In other situations, complex visual responses are offered as the form of input, and the structure of interest is composed of a variety of basic shapes, combinations of these shapes, and/or freeforms. Both Conditional Generative Adversarial Networks (CGAN) and Convolutional Neural Networks (CNN) are beneficial for dealing with one-to-many issues and inverse design problems. When developing the architecture of a network, it is important to take these two types of networks into consideration.

S. So et al. [42] used a conditional deep convolutional generative adversarial network to reconstruct six fundamental forms of silver plasmonic structures (circle, square, cross, etc.) from a linearly polarized reflectance spectrum (cDCGAN). The generator and discriminator of a GAN-based model compete during the training phase. The generator

constructs structures from the input spectrum and noise vector in an attempt to fool the discriminator into thinking they are legitimate, using the discriminator's expertise with the training set. In this case, the noise vector is produced from a fluctuating distribution based on the supplied frequency range. This is done through the discriminator, which can tell the difference between a structure formed by the generator and one from the training dataset. After each input structure is pixelized into a 64x64 image, features from both images are retrieved using a CNN. Although the optimized discriminator improves with training over many iterations, it is still unable to distinguish between true and false inputs since the generator can build structures that are remarkably similar to the needed structures, opening the door to reverse engineering. graphical illustration

To generate images of nanostructures, Liu and colleagues used a generative adversarial network using spectra (GAN). Using statistical learning theory techniques, the authors show that their model can generate randomly selected test samples from a variety of geometric object families, including squares, circles, sectors, and crosses. The model is then asked to infer geometry from observed training set templates after being trained on samples from all families. By using a test set that included the complete geometric subfamily, Liu and colleagues examined how to increase the generalization of future tests. However, the projected geometry differs significantly from the original geometry, and the overall spectrum properties of the predicted geometry are comparable to those of the input spectrum, with minor differences [44]. This is because the authors decided to keep samples from a subset of the family in the test set and train their model on the remaining data set.

Because the relationship between input and output that a neural network learns is typically implicit, it is commonly referred to as a "black box." Because artificial neural networks can predict the optical responses of micro-nano structures, inverse design can be performed without the time-consuming design approach. Some published research make an admirable effort to illustrate each layer's output, revealing which properties (or functions) are learned by each layer. It is possible to discover new structures such as H-shapes, crosses, and others by directly extracting the relationship between micro-nano

structure and optical response from a trained neural network. New recommendations for optical device design are provided.

Another significant goal is to improve the adaptability of neural network models to novel situations. When applied to classical problems such as image recognition and natural language processing, it is preferable for artificial neural networks to first learn the information and distributions contained within natural images or language, and then attempt to reconstruct or approximate those distributions. Artificial neural networks are well-suited to learning and summarizing picture and language distribution. Simultaneously, it is simple to extend the technique to accommodate different picture or language types. Nonetheless, micronanostructures make inverse design more complex. The structure's information and distributions are less important than the physical laws that artificial neural networks must learn. In reverse engineering challenges, the scalability of well-trained neural networks remains a hurdle.

Deep learning and the fabrication of micro-nanostructures have advanced independently during the last few decades. The merging of these fields has resulted in some astonishing discoveries in recent years. On the one hand, the artificial neural network model is a powerful tool for expediting the optical design and implementation process while revealing some surprising structures and occurrences. Production restrictions and material attributes must be included into the design space for global optimization of devices and systems to attain this goal. Innovation and Development Methodology.

3. Design strategy of micro-nano structure based on neural network

We want to predict the optical response of the optical structure based on deep learning, that is, the process from the input optical structure image to the output response of the system. We can use the model based on convolutional neural network for the deep learning algorithm here.

Of course, we can also use models based on fully connected neural networks for the deep learning algorithm here, but in general, for image analysis performance, models based on convolutional neural networks are better than models based on fully connected neural networks. This has been proved by scholars' experiments on image processing and analysis. So based on this consideration, we use a model based on convolutional neural networks.

One part of the data set used for neural network training is 2D images of optical structures, and the other part is the optical response values corresponding to these structures obtained in the simulation software. Using pre-made optical structure images and a dataset of corresponding optical response values, the model is trained so that the neural network learns the hidden correspondence between the structure image and the corresponding optical response.

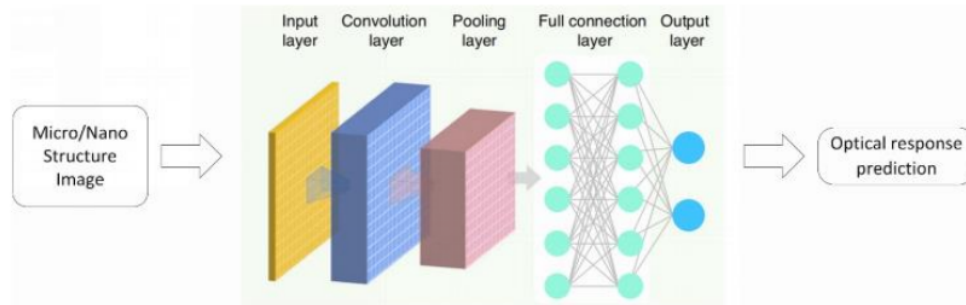


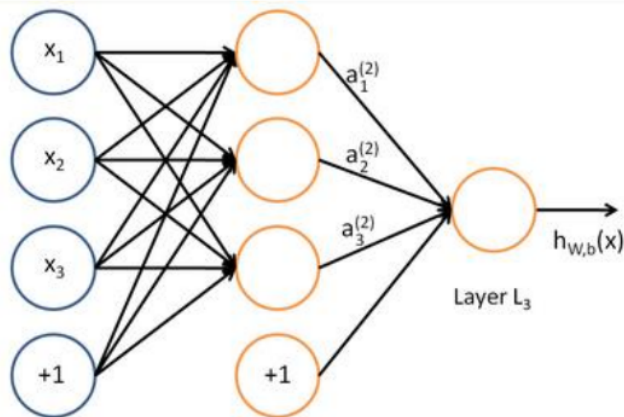
Fig 3. Schematic diagram of model structure

After the model training is completed, we can test the model, input the optical structure image in the test data set, and the neural network can predict the optical response. By comparing the error between the predicted value and the real value, we can

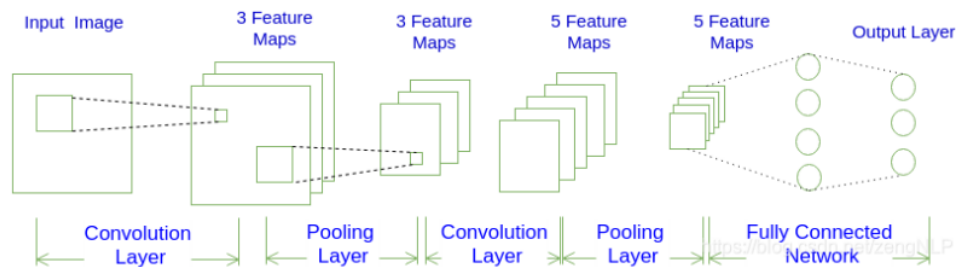
evaluate the performance of the neural network model. If the performance is better, then the neural network model can help us predict the optical response of the optical structure, instead of the traditional method such as numerical full-wave simulation, and improve the speed of calculation.

related to deep learning. The problem of regression is to find the value of incomplete data. For example, in a data instance, if you know characteristics such as the width, height, wheel size, and engine horsepower of a car, but not the length, you can predict the length characteristics for an instance consisting of incomplete data. Values on other data instances.

Commonly used networks ANN, CNN, neural network are shown in the following figure:



Schematic diagram of a convolutional neural network:



Convolution refers to the process of extracting features in sliding, which can be visually understood as the process of magnifying each step with a magnifying glass and taking pictures, and then stitching the taken pictures into a new large picture.

2D convolution is a fairly simple operation: we start with a small weight matrix, the convolution kernel, and let it gradually "scan" over the 2D input data. While the convolution kernel is "sliding", the product of the weight matrix and the scanned data matrix is computed, and the result is aggregated into one output pixel.

That is to say, each element in the matrix obtained after the convolution operation is obtained by the product of two matrices-these two matrices are the "size equal to the convolution kernel" segmented by the original matrix in the sliding process. "The matrix and convolution kernel (the convolution kernel is generally a 3×3 or 5×5 matrix) consists of several convolution layers, Pooling layers, and fully connected layers. There are all kinds of different convolutional neural networks you can build, and image recognition is one of the main areas where neural networks are used and involves identifying a large number of features hidden behind pixels of data.

Backpropagation is commonly used to compare the output with the error on the training data and calculate the error in the output. The layers immediately next to the output layer then adjust their weights, causing weight adjustments in subsequent inner layers until the error rate decreases. Deep learning requires a large dataset of different instances from which the model learns the features to look for and produces an output with a probability vector.

In order to verify the necessity of big data, let's take a look at 2 successful application scenarios of large-capacity training data:

1. The well-known modern face recognition system on Facebook called "DeepFace" deploys a set of 4 million facial images of more than 4000 identities and achieves 97.35% accuracy on the labeled dataset.

2. Other scholars such as Alex Krizhevsky and Geoffrey Hinton describe a robotic grasp learning model involving hand-eye coordination. To train their network, a total of 800,000 grasping attempts were collected, and the robotic arm successfully learned a wider variety of grasping strategies.

4. Data collection

Part of the data set used for neural network training is 2D images of optical structures. The structural images in the data set are free-form, and the image resolution is temporarily set to 32*32. You can use the commercial math software MATLAB to randomly place several images on the canvas. Randomly shaped rectangles for image creation.

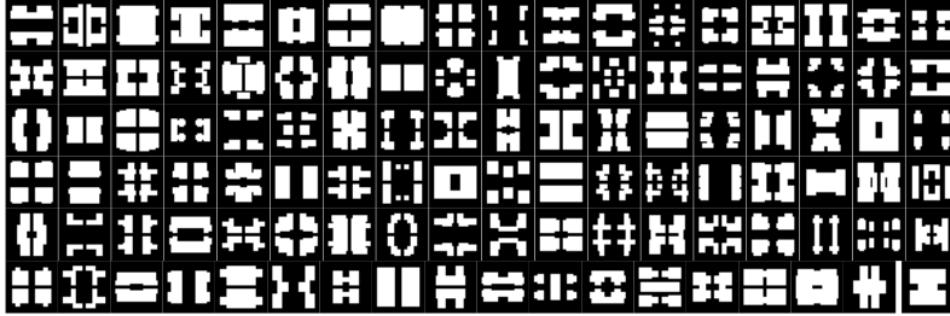


Figure Structural Image Example

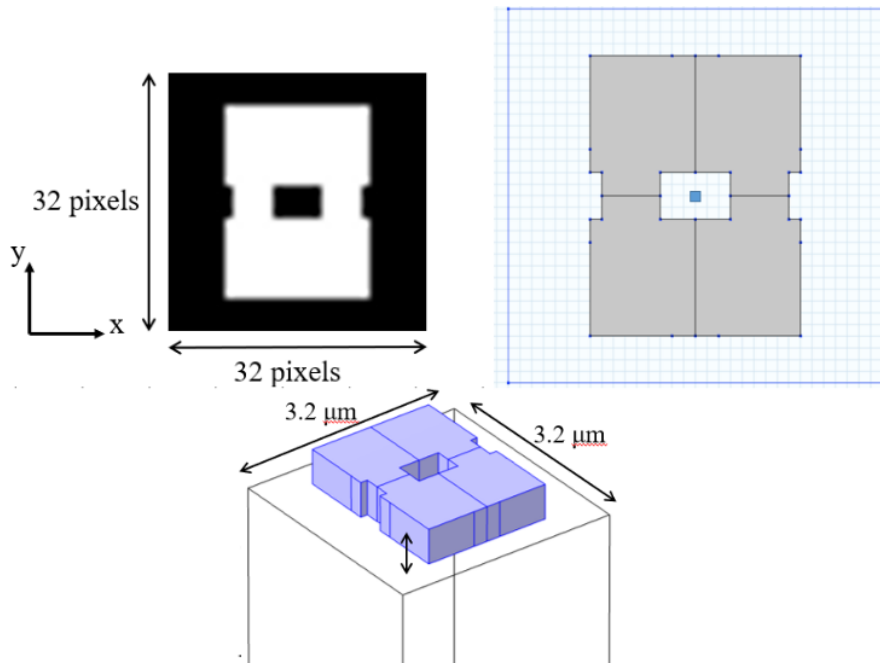
The other part is the optical response values corresponding to these structures obtained in the simulation software. After obtaining the structural image, the numerical full-wave simulation software COMSOL applied to optics is used to simulate the optical response corresponding to the structural image, which here refers to the outgoing electric field data at a frequency of 70THz. 2000 sets of structural images and corresponding optical response data were produced, 1400 sets of data were used for model training, and 600 sets of data were used for model testing.

Table Example of output electric field response values corresponding to structural images

0.038891	0.650528	0.475161	0.812854	0.653211	0.48126	0.431056	0.039718
0.638562	0.009862	0.539492	0.386106	0.234562	0.201752	0.242513	0.316733
0.58187	0.530398	0.481899	0.625667	0.135964	0.665295	0.710142	0.609867
0.089984	0.13366	0.483766	0.174973	0.268265	0.442031	0.552722	0.036621
0.431073	0.171979	0.30719	0.543057	0.149626	0.438042	0.130936	0.732968
0.547556	0.471833	0.311042	0.148483	0.116739	0.454672	0.542294	0.56943
0.581354	0.48754	0.219027	0.025405	0.025328	0.576952	0.112201	0.252956
0.120585	0.076795	0.558691	0.584515	0.524747	0.400147	0.266974	0.604111
0.489067	0.316272	0.433969	0.533831	0.344056	0.008096	0.019883	0.352523
0.181157	0.138846	0.569177	0.088892	0.653545	0.024591	0.415046	0.452367

The dataset we use here, in real life, usually only uses a small dataset. Models trained on a small number of observations tend to overfit and produce inaccurate results. So even if the available data is extremely limited, you need to understand how to avoid overfitting and get accurate predictions. The concepts of big data and data science are often mentioned together. It is believed that data science can draw some valuable insights from terabytes of massive data. Indeed, in theory it is possible. However, in practical scenarios, the data that can be used to solve the problem is usually limited. Because collecting a huge dataset can be very expensive, or simply impossible (for example, when doing time series analysis, there are only records from a specific time period). So often times there is no choice but to use a small dataset to get as accurate predictions as possible.

For the image fabrication method, the considered all-dielectric inter-unit is composed of a dielectric upper layer and a dielectric substrate. The top view of the unit is a square with a side length of $3.2\ \mu\text{m}$, and the pixels of the produced image are 32×32 pixels. Each image is obtained by the upper and lower quarter parts symmetrically, and the upper part is composed of 3 rectangles placed at random positions and random side lengths under the condition that the minimum side length is 3 pixels.



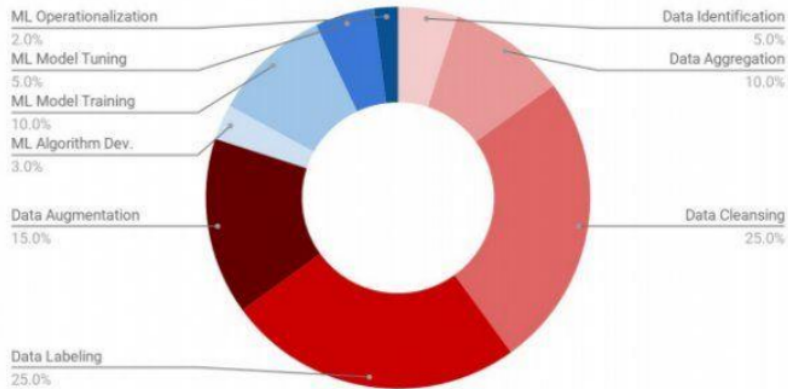
The upper layer is a high refractive index medium, and the lower layer is a low refractive index medium. To speed up the data collection process, all-dielectric components are generated only in the upper left quadrant of each lattice and then replicated symmetrically along the x and y axes to form the entire pattern. Full-wave electromagnetic simulation using COMSOL. For each element, perfect conducting surface and perfect magnetic permeating surface boundary conditions are used to calculate the transport and phase shift of the square lattice structure. The x-polarized plane wave is irradiated from the substrate side. Without identical structures, 2000 structures were selected and recorded for simulation and to generate electric field response values.

The evaluation and analysis of the data set, according to the experiment, can generate thousands or even tens of thousands of structural images, but in fact the similarity of these data is very high, so after comprehensive consideration, the number of data sets in thousands is selected. It can also be seen through the subsequent deep learning-based model establishment and training process that these data are sufficient to train a relatively qualified model.

The importance of datasets, there are 3 aspects to building a successful AI/ML model: algorithms, data and computation. While building accurate algorithms and application of computational skills is part of the process, a good machine learning project, from writing a model to field testing it, is based on using the right dataset, whether it be automated. From large-scale technological revolutions based on AI, such as driving cars, to building very simple algorithms, we all need data in the right format. According to recent research from Cognilytica, in which responses from organizations, institutions, and end-user businesses were recorded and analyzed to identify the time spent tagging, annotating, cleaning, enriching, and enriching data for machine learning models, the following graph amply illustrates at this point:

Percentage of Time Allocated to Machine Learning Project Tasks

Source: Cognilytica



Data is the input of deep learning, which is very important and easy to be ignored. Nowadays, machine learning has highlighted its irreplaceable importance in more and more fields, and people have begun to penetrate typical cases of machine learning from various fields. I hope it will be put into use on a large scale, and the difference between good training results and bad training results is not how good or bad your model is, but often how much and how good data you have to train your model. This is why many Internet companies (such as BAT, Google, Amazon) are now focusing on the field of big data sources in machine learning.

Let's go back to the example of cat pictures. You have developed a mobile app, users have uploaded various photos, and you want the computer to automatically distinguish which photos contain cats. At the same time, in order to train the model, your teammates found a large number of cat pictures (positive examples) and non-cat pictures (negative examples) from the Internet. They divided the data into a training sample set and a test sample set in a ratio of 7:3. From this data, they built a cat detector model that performed well on both the training and testing samples. However, when you apply this model to the mobile APP, you find that the model does not work, and the detection accuracy is extremely poor. You find that the pictures uploaded by users are very different from the pictures you collect from the Internet. Most of the pictures uploaded by users are taken with mobile

phones, so the pixels are low, blurry, and the lighting is not very good. Because all the training data you used before came from the Internet. Your algorithm model is not suitable for real data, i.e. photo data from mobile phones.

Before the era of big data, it was a very common practice to divide the data into training sample sets and test sample sets by 7:3. Practice has proved that this is indeed useful, but for more and more applications, their training data is not the same as the final actual data distribution, then it is not a good idea to do so. How to do it? Typically we define three datasets: the training dataset — used to train your algorithmic model. Development datasets — used to tune the learning model, such as tuning parameters, selecting features, etc. This part of the data is also known as the hold-out cross-validation dataset. Test dataset — used to evaluate model performance, this part of the dataset does not participate in model tuning and parameter updates. Once you've defined your dev and test datasets, your team can make a lot of other experiments (like trying other parameters on the learning algorithm) to find the best performing idea. Developing and testing datasets allows your team to quickly see if an algorithm works and how it performs. In other words, using dev and test datasets can guide your team to make the most important decisions and changes directly to your machine learning system. Therefore, you should choose the data that you will actually use in the end as a sample set for development and testing. Therefore, when your final data is inconsistent with your pre-collected training data, you'd better not simply use 30% of the available data as the test data set. If your APP is not yet online, or you don't have users yet, you may not be able to collect actual data from users at this time. But I still recommend that you find a way to find data that is similar to the real data. For example, ask your friends to help, take a picture of the cat with your mobile phone and send it to you. When your app is online, you can replace the approximate data in the previous development and test sample sets with real data. If you really can't find data that approximates the actual situation, you can use data collected from the Internet to train your model. But be aware of the risk of generalization that can result, i.e. when you train your model, it performs well, but when you use it, it performs poorly. You need to consider how much it costs to build better dev and test datasets. But never assume that the training data has the same distribution as the actual data. The test data set should try to choose

data similar to the final data, not like training, any data can be selected. We divide the data set into training data set and test data set. If we use the training data set to get a model, the test data set is a brand new sample for this model. If the model can also obtain the test data set, it will be very good. If the effect is good, the model is said to have a good generalization effect. If the model cannot get a good effect on the test data set, it means that the model does not have a good effect, and the generalization ability of the model is weak.

5. Design and Implementation

5.1. Initial Design

Deep learning is the application of artificial neural networks using modern hardware. It makes it possible to develop, train and use larger (more layers) neural networks than in the past.

Researchers have proposed thousands of types of specific neural networks, which are often modifications or adaptations of existing models. Sometimes there are completely new approaches. Three types of artificial neural networks are generally concerned. Namely: Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), these three classes of networks offer great flexibility and have been proven for decades in various useful and reliable in such problems. And they have proven useful and reliable in many problems over the past few decades. They also have many variants to help them specialize in different prediction problem frameworks and different datasets.

Now that we know which networks to focus on, let's see when each type of neural network can be used. Convolutional Neural Networks (CNNs) are designed to map image data to output variables. It has proven to be very effective, and it is the preferred method for any kind of prediction problem involving image data as input. The benefit of using CNNs is the ability to develop internal representations of 2D images. This allows the model to learn location and scale in the variant structure in the data, which is important when working with images.

Using CNNs: , Image Data, Classification Prediction Problems, Regression Prediction Problems In summary, CNNs are suitable for working with data with spatial relationships. If one hidden layer can solve the problem, we try to use one layer as much as possible. Too many hidden layers will increase the amount of calculation. If there are multiple hidden layers, it is best to have the same number of units in each hidden layer (Default), in general, the more hidden units, the better (but the amount of calculation will be large), because there are more parameters, and the fitting effect is better.

We can establish a relatively basic convolutional neural network architecture based on experience or first, and then adjust the parameters step by step on the basis of this

initial design to optimize the entire neural network model. Until we can no longer proceed with substantial effective optimization, we can stop and determine the final convolutional neural network model to use.

We need to find a suitable neural network architecture and adjust the parameters based on the architecture to determine the model with better performance.

The structural parameters of the initial model are shown in the figure below.

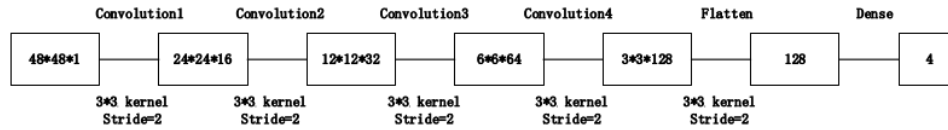


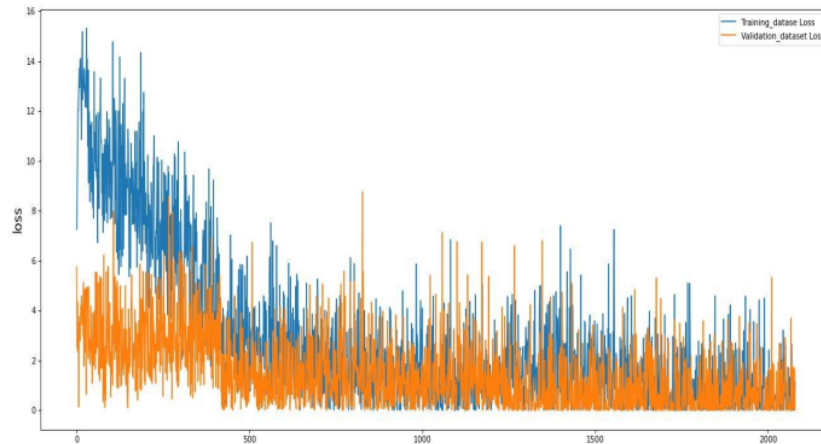
Fig Initial Model Structure

The structural parameters of the initial network model. 4 convolutional layers and 2 fully connected layers. The size of the convolution kernel is 3*3,

The optimizer optimizer is Adam, and the learning rate is kept at 0.001 during training. The training hyperparameters of the model are as follows: batchsize is 12 and the number of iterations is 2000. Output the loss Loss of training dataset and validation dataset after each iteration. At the same time, we plot these data to help us observe the training process more intuitively.

Based on the above parameter settings, we use the python language to build and train the model based on the pytorch deep learning framework. The loss loss curve of the

training process is shown in the following figure:



In order to help us optimize the neural network model, we need to choose performance measurement parameters to reflect the training effect of the neural network model. Here, we choose the loss function Loss and classification accuracy Accuracy of the model as performance metrics, based on which to observe the performance of the model during training. It can be seen from the above two figures that the model has a problem of overfitting. The more serious problem is that with the deepening of the training process, the loss value of the validation data set does not decrease but increases.

After determining the model, we can use the test data to test the model performance. We can calculate the average loss value of the model on the test set.

Based on the above initial model design, we try to improve the performance of the model mainly by adjusting the structural parameters of the neural network and some important hyperparameters.

5.2. Choice Of Model Parameters

The adjustment of the structural parameters of the neural network is mainly the adjustment of the number of filters and the number of network layers in the convolutional neural network

Changes in the number of layers.

A MaxPool network layer was added and downsampled using Stride 2 to progressively halve the size of the feature map. Also change the step size of the convolution layer to 1. Experiments show that this adjustment works better than using only convolutional layers.

Increase the number of filters in the convolutional layer.

Add weight initialization, using the `heiniInitialize` method for initialization.

The hidden layer is regularized before the output layer with a regularization coefficient of 0.1. Because the model overfitting problem is relatively serious, so try to alleviate the overfitting problem. Add another term to the loss function to make the "penalty" more severe. This method is called regularization. Regularization can make the data more general and thus prevent the neural network from overfitting.

There is no significant performance gain from increasing the number of model layers, so do not increase the number of model layers. The adjustment of the structural parameters of the neural network is mainly the adjustment of the number of filters and the number of network layers in the convolutional neural network

5.3. Choice Of Hyper-Parameters

☑ Select the SGD optimizer, where the initial learning rate lr is 0.01, the decay coefficient decay is 0.001, and the momentum is set to 0.9.

☑ In addition, when training the model, increase the batch size to 128.

5.4. Intermediate Results

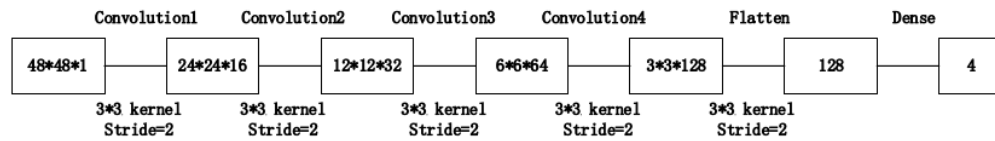
Adding the maximum pooling layer MaxPool, adding the weight initialization and adding the regularization have certain effects on improving the classification accuracy of the model on the training dataset and the generalization performance of the model. The improvement in model performance by tuning these parameters can be seen in the testing of the final model.

As shown in the figure below, if the activation function of the hidden layer uses sigmoid, the performance of the model is poor, and even has a serious performance indicator stagnation problem during training. Therefore, you should still use ReLU as the activation function for the hidden layer.

5.5. Final Model

After continuous adjustment of the initially designed convolutional neural network model, the final model with relatively improved performance can be obtained and its test results are as follows.

Through the adjustment of the above parameters, the final model structure is determined as follows:



Network model structure parameters: the input image first goes through 3 convolution layers, 1 pooling layer, and 3 fully connected layers, and the number of neurons in the final output layer is 1.

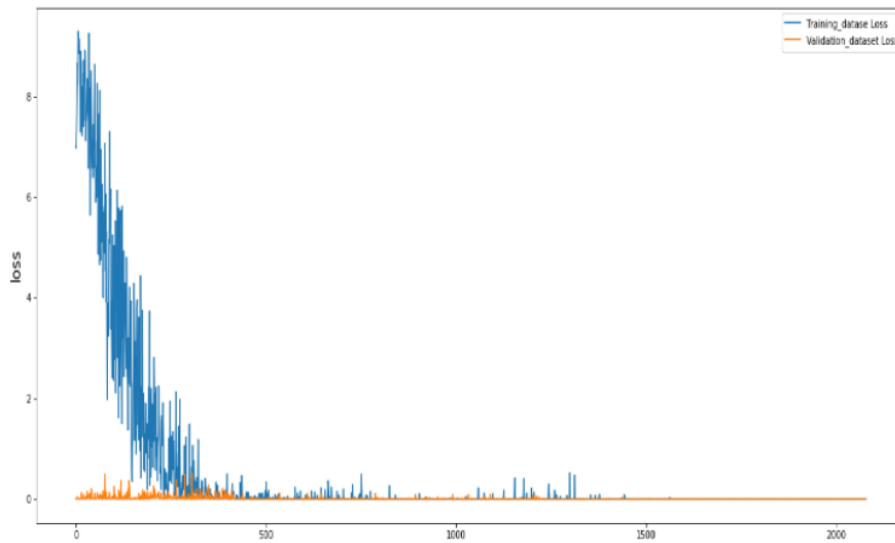
Other hyperparameters: The activation function of the hidden layer uniformly uses ReLU, the optimizer optimizer is Adam, and the learning rate is kept at 0.001 by default during the training process.

The training hyperparameters are as follows: the batch size is 12 and the number of iterations is 2000iterations. Output the loss Loss of training dataset and validation dataset after each iteration. At the same time, we plot these data to help us observe the training process more intuitively.

Loss record of the training process of the final model:

```
100% | 8333/8333 [04:38<00:00, 29.90it/s]
100% | 416/416 [00:03<00:00, 109.33it/s]
0% | 0/8333 [00:00<?, ?it/s]第1个epoch成功, 准确率为0.9827993121903426, 损失为0.13550426898311593, 测试集准确率1.0, 测试集损失1.3625016903299289e-05
100% | 8333/8333 [04:42<00:00, 29.53it/s]
100% | 416/416 [00:02<00:00, 162.88it/s]
第2个epoch成功, 准确率为0.9987799513387206, 损失为0.805046039354479174, 测试集准确率1.0, 测试集损失1.578617404488588e-05
100% | 8333/8333 [04:56<00:00, 28.07it/s]
100% | 416/416 [00:02<00:00, 158.96it/s]
0% | 0/8333 [00:00<?, ?it/s]第3个epoch成功, 准确率为1.0, 损失为3.667489812231819e-06, 测试集准确率1.0, 测试集损失3.824371002701345e-07
100% | 8333/8333 [04:57<00:00, 27.98it/s]
100% | 416/416 [00:02<00:00, 152.33it/s]
第4个epoch成功, 准确率为0.999719988859159, 损失为0.0013266812839612936, 测试集准确率1.0, 测试集损失1.461235403416455e-06
100% | 8333/8333 [04:59<00:00, 27.79it/s]
100% | 416/416 [00:02<00:00, 169.37it/s]
第5个epoch成功, 准确率为1.0, 损失为3.1427379307744546e-07, 测试集准确率1.0, 测试集损失3.4028966016802926e-08
```

Loss curve of the training process of the final model:



Loss value on test set: 3.4028966×10^{-8} , Test examples:

True Label	Predicted Value	
0.321582634	0.321500125	
0.504657256	0.503257256	
0.484006631	0.484633663	

6. Results and Discussion

In order to analyze and verify the effectiveness of the method, it is necessary to visually monitor the training process of the model, and perform performance test and instance verification on the test results.

When training the model using the training data, the first thing to ensure is that the loss of the convolutional neural network model converges, which proves that the model can be trained to a steady state on this data. Second, we need to tune the hyperparameters of the neural network model, such as the number of network layers, the size of the convolution kernel, the learning rate, and the number of neurons. Finally, when the model parameters are determined, the value of the neural network loss function needs to be plotted according to the number of training iterations. In general, the loss function value decreases gradually and tends to a small stable value.

After the training of the light response prediction model based on convolutional neural network is completed, several structured samples are randomly selected in the test data set for light response prediction, and compared with the real light response for verification, and a table is drawn to verify and prove the prediction effect of the model.

To help us optimize the neural network model, we need to choose the performance measurement parameters to reflect the training effect of the neural network model. Here, we choose the loss function loss and classification accuracy of the model as performance metrics and observe the performance of the model during the training process based on them.

From the loss change curve of the training process, it can be seen that the model can eventually converge after a period of training.

After the final model is determined, we can use the test data to test the performance of the model.

True Label	Predicted Value	
0.051434568	0.041532568	
0.606638882	0.633338882	
0.165289547	0.164889547	

From the performance metrics and test results of the training process of the convolutional neural network model on the above test dataset, it can be seen that the performance of the model has been improved after the parameter tuning in Chapter 3.

But the problem with the model is overfitting, because the accuracy on the training set is over 90%, which is much higher than the accuracy on the validation set. Although regularization is added to the model, the problem is not completely solved.

Moreover, the classification accuracy of the model is not very high on both the validation set and the test set, while the classification accuracy of the officer is very high on the training set. This indicates that the generalization performance of the model is not very good, which is a big problem to be solved. In order to analyze and verify the effectiveness of the method, it is necessary to visually monitor the training process of the model, and perform performance test and instance verification on the test results.

When training the model using the training data, the first thing to ensure is that the loss of the convolutional neural network model converges, which proves that the model can be trained to a steady state on this data. Second, we need to tune the hyperparameters of the neural network model, such as the number of network layers, the size of the convolution kernel, the learning rate, and the number of neurons. Finally, when the model parameters are determined, the value of the neural network loss function needs to be plotted according to the number of training iterations. In general, the loss function value decreases gradually and tends to a small stable value.

After the training of the light response prediction model based on convolutional neural network is completed, several structured samples are randomly selected in the test data set for light response prediction, and compared with the real light response for verification, and a table is drawn to verify and prove the prediction effect of the model. In order to analyze and verify the effectiveness of the method, it is necessary to visually monitor the training process of the model, and perform performance test and instance verification on the test results.

7. Conclusions and Recommendations

Photonics has brought about transformation in a variety of scientific and commercial fields. Even for experienced researchers, developing sophisticated micro- and nanostructures and devices requires a significant investment of time. On the other hand, the construction of subwavelength structures in a rational manner enables us to manipulate light in any way we choose. Artificial neural networks, a subset of artificial intelligence, offer a promising alternative to conventional design approaches by allowing us to execute inverse design quickly and inexpensively or immediately predict the optical response of micro-nanostructures [10]. This makes artificial neural networks a subset of artificial intelligence that offers a promising alternative to conventional design approaches.

The development trend of deep learning, deep neural network shows the development trend of deeper and deeper layers and more complex structure. In order to continuously improve the performance of deep neural networks, the industry continues to explore the network depth and network structure. The number of layers in neural networks has expanded to hundreds or even thousands of layers. With the deepening of the network layer, its learning effect is better and better. In 2015, Microsoft proposed ResNet with 152 layers of network depth for image classification tasks. For the first time, the accuracy rate exceeds the human eye. New network design structures are constantly proposed, which makes the structure of neural networks more and more complex. For example, Google proposed Inception network structure in 2014, Microsoft proposed residual network structure in 2015, and Huang Gao proposed densely connected network structure in 2016. These network architecture designs continuously improve the performance of deep neural networks.

The functions of deep neural network nodes are constantly enriched. In order to overcome the limitations of current neural networks, the industry has explored and proposed new neural network nodes, making neural networks more and more functional. In 2017, Jeffrey Hinton proposed the concept of capsule network, which treats capsules as network nodes, theoretically closer to the behavior of the human brain, and aims to overcome the limitations of convolutional neural networks in spatial layering and reasoning

capabilities. In 2018, scholars from DeepMind, Google Brain, and MIT jointly proposed the concept of graph networks and defined a new class of modules with relational inductive bias functions, aiming to empower deep learning through causal inference.

Deep neural network engineering application technology is deepening. Most deep neural network models have hundreds of millions of parameters, hundreds of megabytes of space, and a lot of computation. To solve this problem, the industry adopts model compression techniques to reduce the number and size of model parameters and reduce the amount of computation. Current model compression approaches include pruning trained models (e.g., pruning, weight sharing, and quantization) and designing more refined models (e.g., MobileNet). The modeling and parameter adjustment process of deep learning algorithm is tedious, and the application threshold is high. In order to reduce the application threshold of deep learning, the industry has proposed automatic machine learning (AutoML) technology, which can realize the automatic design of deep neural networks and simplify the use process.

Deep learning and various machine learning techniques continue to merge and evolve. Deep reinforcement learning technology was born from the fusion of deep learning and reinforcement learning, which fuses the perception ability of deep learning and the decision ability of reinforcement learning, and overcomes the fact that reinforcement learning is only suitable for discrete and low-dimensional states. Data Learning Control policy. In order to reduce the amount of data required to train deep neural network models, the industry introduces the idea of transfer learning and produces the technology of deep transfer learning. Transfer learning refers to the learning process of applying a model learned in an old domain to a new domain by exploiting similarities between data, tasks, or models. By migrating the trained model to similar scenarios, only a small amount of training data is needed to achieve better results.

Suggestions for future development, and strengthen the research on cutting-edge technologies such as graph networks, deep reinforcement learning, and generative adversarial networks. Due to the lack of important original research results and basic theoretical research contributions in the field of deep learning in China, American experts have proposed innovative and original concepts such as capsule network and graph

network, and Chinese research contributions are insufficient. In terms of deep reinforcement learning, most of the latest research results are proposed by researchers from foreign companies such as DeepMind and OpenAI, while there are no breakthrough research results in China. In recent years, Generative Adversarial Networks (GAN), a research hotspot, was proposed by Goodfellow, an American researcher. Google, Facebook, Twitter, Apple and other companies have proposed various improvements and application models, which have strongly promoted the development of GAN technology. But our country is less research achievements in this respect. Therefore, scientific research institutes and enterprises should be encouraged to strengthen the research on the combination of deep neural networks and causal inference models, generative adversarial networks, deep reinforcement learning and other cutting-edge technologies, and propose more original research results to enhance their influence. Global academic research.

Accelerate research on deep learning application technologies such as automated machine learning and model compression. Relying on the domestic market advantages and the growth advantages of enterprises, in response to the personalized application needs with Chinese characteristics, accelerate the research on deep learning application technology. Strengthen research on automated machine learning, model compression and other technologies, and accelerate the engineering application of deep learning. Strengthen the application research of deep learning in the field of computer vision, and further improve the accuracy of visual tasks such as target recognition, as well as the performance in practical application scenarios. Strengthen the application research of deep learning in the field of natural language processing, propose algorithm models with better performance, and improve the performance of applications such as machine translation and dialogue systems.

Deep learning has a significant deal of potential use in the field of digital image processing. This does not imply that we have given up on the tried-and-true approaches of computer vision that were developed prior to the introduction of deep understanding. Researchers at the Irish Terry Institute of Technology developed an analysis that compares and contrasts the two different ways. Vision applications are becoming increasingly commonplace as a result of developments in deep learning, rapid app iteration, and

enhanced device capabilities (such as CPU speed, memory size, energy usage, image sensor resolution, and optics). Deep learning can assist engineers enhance accuracy in areas such as image classification, semantic segmentation, object recognition, and simultaneous localization and map generation when compared to classic CV methods (SLAM). Deep learning applications are able to analyze the large volumes of video data currently present in systems because they use learnt neural networks rather than coded neural networks. This eliminates the requirement for human interpretation and correction of the data.

Given the scope and complexity of AI, we require processors with increased throughput at reduced power consumption. There is much room for adaptation in optical computing systems, especially in visual computing. Despite fifty years of research and development, current general optical photonic computing systems are still not ready for widespread deployment in actual applications. As GPU speeds and supervised machine learning methods have increased, so too has the efficiency with which deep neural networks (DNNs) can train with massive amounts of data. However, this complicates the use of DNNs on edge devices (cameras, autonomous driving, robotics, IoT devices, etc.). Optical neural networks (ONNs) and photonic circuits offer numerous benefits over traditional electrical computer systems, including high levels of parallelism, low power consumption, and minimal hardware requirements for operation. There are two main parts to the DNNs framework, and they are training and inference. The reasoning part of the system improves both all-optical and hybrid electro-optical systems. Convolutions, Fourier transformations, random projections, and other operations can all be carried out using linear optics components with extremely low power consumption and lightning-fast response times.

REFERENCE

- [1] Zuo C, Qian J, Feng S, et al. Deep learning in optical metrology: a review[J]. Light: Science & Applications, 2022, 11(1): 1-54.
- [2] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. nature, 1986, 323(6088): 533-536.
- [3] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.
- [4] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.
- [5] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
- [6] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.
- [7] Xu Y, Zhang X, Fu Y, et al. Interfacing photonics with artificial intelligence: an innovative design strategy for photonic structures and devices based on artificial neural networks[J]. Photonics Research, 2021, 9(4): B135-B152.
- [8] Yao K, Unni R, Zheng Y. Intelligent nanophotonics: merging photonics and artificial intelligence at the nanoscale[J]. Nanophotonics, 2019, 8(3): 339-366.
- [9] Zhang Q, Yu H, Barbiero M, et al. Artificial neural networks enabled by nanophotonics[J]. Light: Science & Applications, 2019, 8(1): 1-14.
- [10] Hegde R S. Deep learning: a new tool for photonic nanostructure design[J]. Nanoscale Advances, 2020, 2(3): 1007-1023.
- [11] Tu X, Chen Z M, Fu H Y 2019 Acta Phys. Sin. 68 104210(in Chinese) [涂鑫, 陈震旻, 付红岩 2019 物理学报 68 104210]
- [12] Liu Z, Zhu D, Rodrigues S P, et al. Generative model for the inverse design of metasurfaces[J]. Nano letters, 2018, 18(10): 6570-6576.
- [13] So S, Rho J. Designing nanophotonic structures using conditional deep convolutional generative adversarial networks[J]. Nanophotonics, 2019, 8(7): 1255-1261.
- [14] Mall A, Patil A, Tamboli D, et al. Fast design of plasmonic metasurfaces enabled by deep learning[J]. Journal of Physics D: Applied Physics, 2020, 53(49): 49LT01.
- [15] Liu Z, Zhu D, Rodrigues S P, et al. Generative model for the inverse design of metasurfaces[J]. Nano letters, 2018, 18(10): 6570-6576.
- [16] Jahani S, Jacob Z. All-dielectric metamaterials. Nature nanotechnology. 2016, 11: 23-36

- [17] Veselago V G. The electrodynamics of subatances with simultaneously negtive values of imgalign. *Physics-Uspekhi*. 1968, 10: 509-514
- [18] Lezec H J, Dionne J A, Atwater H A, et al. Negative refraction at visible frequencies. *Science*. 2007, 316: 430-432
- [19] Arnedo I, Illescas J, Flores M, et al Forward and backward leaky wave radiation in splitring-resonator-based metamaterials. *IET microwaves, antennas & propagation*. 2007, 165-68
- [20] English LQ, Wheeler, S, Shen Y, et al. Backward-wave propagation and discrete solitons in a left-handed electrical lattice. *Physics Letters A*. 2011, 375: 1242-1248
- [21] Grbic A, Eleftheriades G V, Flores M, et al Experimental verification of backw ard-wave radiation from a negative refractive index metamaterial. *Journal of Applied Physics*. 2002, 92: 5930-5935
- [22] Chen, J, Wang, Y, Jia B, et al. Observation of the inverse Doppler effect in negative index materials at optical frequencies. *Nature Photonics*. 2011, 5: 239
- [23] Lee S H, Park C M, Seo Y M, et al. Reversed Doppler effect in double negative metamaterials. *Physical Review B*. 2010, 81: 241102
- [24] Leong K M, Lai A, Itoh T, et al Demonstration of reverse Doppler effect using a lefthanded transmission line. *Microwave and optical technology letters*. 2006, 48: 545-547
- [25] Cheng M, Park C M, Seo Y M, et al. Forward-and backward-propagating Cerenkov radiation in strong chiral media. *Optics Express*. 2007, 15: 9793-9798
- [26] Pendry JB, Holden A, Stewart W, et al. Extremely low frequency plasmons in metallic mesostructures. *Physical Review Letters*. 1996, 76: 4773
- [27] Pendry J B, Holden A J, Robbins D J, et al. Magnetism from conductors and enhanced nonlinear phenomena. *IEEE Transactions on Microw ave Theory and Techniques*. 1999, 47: 2075-2084
- [28] Smith D R, Schultz, S, Markoĝ, P, et al. Determination of effective permittivity and perm eability of metamaterials from reflection and transmission coefficients. *Physical Review B*. 2002, 65: 195104
- [29] Zheludev N I. The road ahead for metamaterials. *Science*. 2010, 328: 582-583
- [30] WenshanC, Shalaev V. *Optical Metamaterials. Fundamentals and Applications*. Springerpp. xi. 2010, 3:9
- [31] Zhou JI, E conomon E N, Koschny T, et al. Unifying approach to left-handed material design. *Optics Letters*. 2006, 31: 3620-3622
- [32] Smith D R, Kroll N. Negative refractive index in left-handed m aterials. *Physical ReviewLetters*. 2000, 85: 2933
- [33] Houc, A A, Brock J B, Chuang I L. Experimental observations of a left-handed material that obeys Snell's law. *Physical Review Letters*. 2003, 90: 137401

- [34] Zhao H, Zhou J, Zhao Q, et al Magnetotunable left-handed material consisting of yttrium iron garnet slab and metallic wires. Applied Physics Letters. 2007, 91: 131107
- [35] Fredkin D, Ron A. Effectively left-handed (negative index) composite material. Applied Physics Letters. 2002, 81: 1753-175
- [36] Foteinopoulou S, Economou E N, Soukoulis C. Refraction in media with a negative refractive index. Physical Review Letters. 2003, 90: 107402
- [37] Kildishev AV, Cai W, Chettiar U K, et al. Negative refractive index in optics of metamaterials. JOSA B. 2006, 23: 423-433
- [38] Ramakrishna S A Physics of negative refractive index materials. Reports on Progress in Physics. 2005, 68: 449
- [39] Shalaev V M, Cai W, Chettiar U K, et al. Optical negative-index metamaterials. Nature Photonics. 2007, 1: 41
- [40] Shalaev V M, Cai W, Chettiar U K, et al. Negative index of refraction in optical metamaterials. Optics Letters. 2005, 30: 3356-3358
- [41] Smith D R, Pendry J B, Wiltshire MC, et al Metamaterials and negative refractive index. Science. 2004, 305: 788-792

APPENDIX: PROGRAMMING CODE

CNN - PROGRAMMING CODE

```
import os
import time

from matplotlib import pyplot as plt

from torchconf import My_model, my_transforms
import numpy as np
from torch import nn, optim, save
from torch.utils.data import DataLoader
from torchvision import datasets
from tqdm import tqdm

tr = "tes"
file_path = "D:/python/xiangmu/niu" # 路径换一下,换成你的解压目录,精确到
train和test的上一级目录。

# dataset = datasets.ImageFolder('D:/python/xiangmu/mu') # 没有
transform, 先看看取得的原始图像数据
# print(dataset.classes) # 根据分的文件夹的名字来确定的类别
# print(dataset.class_to_idx) # 按顺序为这些类别定义索引为0,1...
# print(dataset.imgs) # 返回从所有文件夹中得到的图片的路径以及其类别

dataset_train = datasets.ImageFolder(root=os.path.join(file_path, tr),
transform=my_transforms,)

# print(dataset_train.classes) # 输出标签
#
# exit()
# test_loader = torch.utils.data.DataLoader(dataset_test, batch_size=20,
shuffle=True)
# print(train_loader)

model = My_model()

optimizer = optim.Adam(model.parameters(), )
loss_function = nn.CrossEntropyLoss()

from torch_test import val_success, val_loss

train_losses = [] # 训练集总损失,用来绘制损失图

def plt_loss(train_losses):
    val_losses = val_loss() #
    # print(val_losses)
    # print(len(val_losses))
    x = [i for i in range(len(train_losses))]
    # print(len(x))
    # print(len(train_losses))
```

```

# exit()
figure = plt.figure(figsize=(20, 8), dpi=80)
plt.plot(x, train_losses, label='Training dataset Loss')
plt.plot(x, val_losses, label='Validation dataset Loss')
# plt.xlabel("iterations", fontsize=15)
plt.ylabel("loss", fontsize=15)
plt.legend()

# plt.grid() #显示网格线
plt.show()

def train(epoch):
    total_loss = [] # 用来存每次的 loss 最后算平均值
    model.train() # 模型训练一下

    train_loader = DataLoader(dataset_train, batch_size=12,
                               shuffle=True, drop_last=True)
    train_loader = tqdm(train_loader, total=len(train_loader))
    succeed = [] # 计算准确率
    for (images, labels) in train_loader:
        optimizer.zero_grad() # 梯度置0
        output = model(images) # 前向传播
        result = output.max(dim=1).indices
        succeed.append(result.eq(labels).float().mean().item()) # 准确率
    百分比
    loss = loss_function(output, labels) # 计算损失
    total_loss.append(loss.item()) # 每次损失加入进列表 最后算平均值
    train_losses.append(loss.item()) # 训练集损失函数输出图列表
    loss.backward() # 反向传播
    optimizer.step() # 优化 优化器更新

    save(model.state_dict(), './tor_test_models.pkl')
    save(optimizer.state_dict(), './tor_test_optimizer.pkl')
    # 打印一下成功率
    # print('第{}个 epoch 成功率, 损失为{}'.format(epoch,
    np.mean(total_loss), test()) # 可以存 test()最大的值 就是识别率的模型
    # val_vale = val_success() # 返回测试集的准确率和损失 返回一个列表 0 是准确
    率 1 是损失
    val_vale = val_success() # 返回测试集的准确率和损失 返回一个列表 0 是准确率
    1 是损失
    print('第{}个 epoch 成功率, 准确率为{}, 损失为{}, 测试集准确率(), 测试集损失
    {}'.format(
        epoch + 1, np.mean(succeed), np.mean(total_loss), val_vale[0],
        val_vale[1]))

for i in range(5):
    train(i)
plt_loss(train_losses)

```

```

from torchvision import datasets
from torch.utils.data import DataLoader
from torch import nn
from tqdm import tqdm
import torch
import numpy as np
from torchconf import My_model, my_transforms
import os

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

val_losses = [] # 测试集总损失集合, 用来输出总损失图

def val_loss():
    return val_losses

def val_success():
    total_loss = []
    # 实例化模型
    model = My_model()
    model = model.to(device)
    if os.path.exists("./tor_test_models.pkl"):
        # print('开始加载')
        model.load_state_dict(torch.load("./tor_test_models.pkl"))
    loss_function = nn.CrossEntropyLoss()

    ts = "tes" # 验证数据集
    file_path = "D:/python/xiangmu/niu" # 路径换一下, 换成你的解压目录, 精确到 train 和 test 的上一级目录。
    dataset_train = datasets.ImageFolder(root=os.path.join(file_path, ts), transform=my_transforms)

    train_loader = DataLoader(dataset_train, batch_size=12, shuffle=True, drop_last=True)
    dataloader = tqdm(train_loader, total=len(train_loader))

    succeed = [] # 计算准确率

    model.eval()
    with torch.no_grad():
        for images, labels in dataloader:
            # 获取结果
            images = images.to(device)
            labels = labels.to(device)
            output = model(images) # 返回数据格式为 tensor([[ -1.2366, 0.3810, -4.6937, 5.9513, -3.5925]])
            result = output.max(dim=1).indices # 返回最大标签索引
            # print(labels)
            # print(result)
            succeed.append(result.eq(labels).float().mean().item())
# 准确率百分比

    # 通过结果计算损失
    loss = loss_function(output, labels)
    total_loss.append(loss.item()) # 平均损失

```

```
        val losses.append(loss.item()) # 总损失，用来绘图
    # print(f'损失为{np.mean(total_loss)}')
    return [np.mean(succeed), np.mean(total_loss)]

if __name__ == '__main__':
    print(val success())
```

2106269-Kaicheng Li-MSc Project

GRADEMARK REPORT

FINAL GRADE

/100

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

