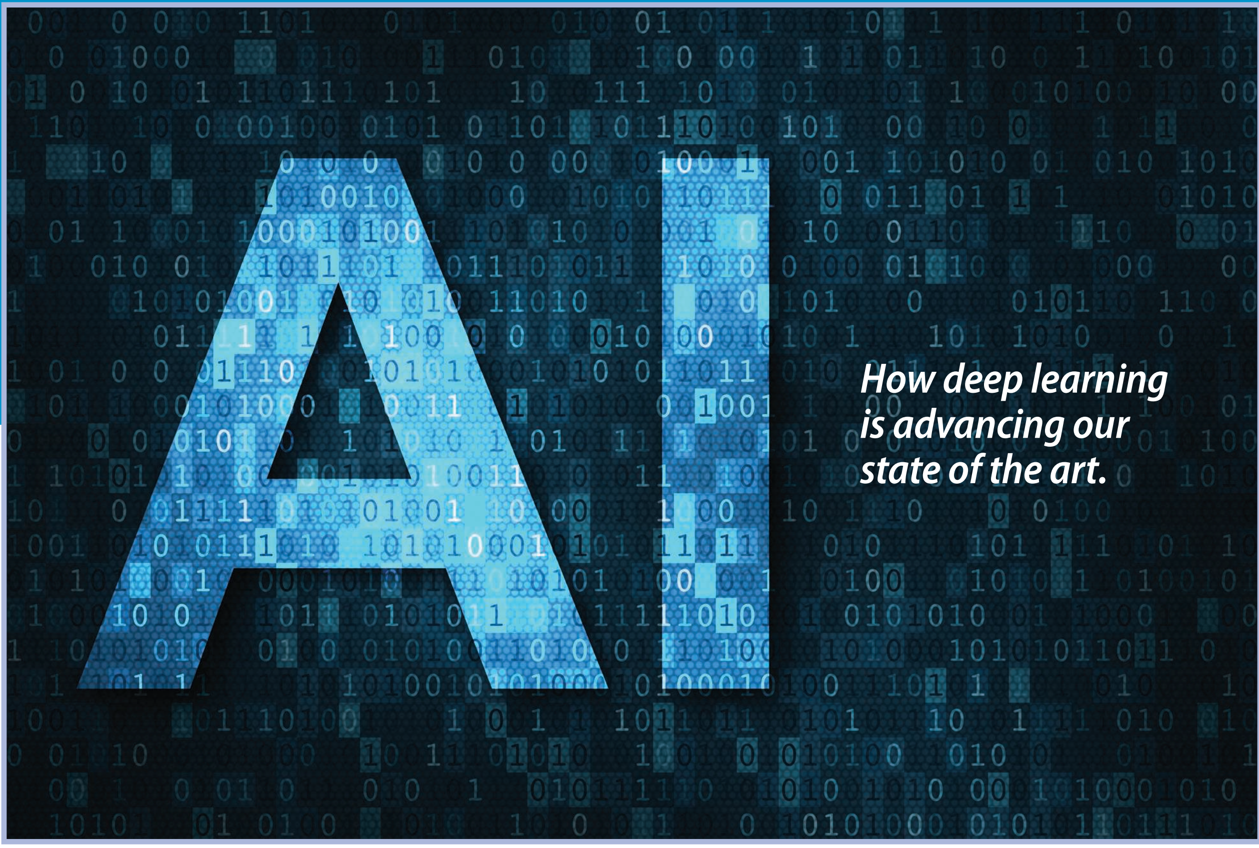


©SHUTTERSTOCK.COM/NICOELNINO

# The Explosion of Artificial Intelligence in Antennas and Propagation

Sawyer D. Campbell, Ronald P. Jenkins, Philip J. O'Connor, and Douglas H. Werner

The rise and proliferation of artificial intelligence (AI) has the potential to influence and disrupt many aspects of society as we currently know it. While it has been a long-held belief that robots are limited to physical tasks and therefore only capable of replacing blue collar workers, many concerns exist that continued developments will yield artificial minds capable of replacing knowledge workers. These advancements have led people to ask the question, "What will our society look like when AI is everywhere?" [1].

While that is hard to predict, perhaps the most recognized example of how AI could impact day-to-day life is driverless cars. What will happen to the truck drivers and couriers who transport goods across the country if driverless cars are fully realized? Moreover, what about the gas stations, hotels, and small towns along our interstates that are supported by the needs of those drivers? It is projected that many workers will need to develop new skills to operate within this new AI world

*Digital Object Identifier 10.1109/MAP.2020.3021433*
*Date of current version: 2 October 2020*

[2]. While some of these prospects are quite unsettling, other AI-enabled disruptions have the potential for massive positive changes going forward.

One area where AI currently excels is in classifying images, and recently AI algorithms have been shown to outperform human doctors at identifying cancer in radiological images—both in terms of speed and accuracy. Thus, AI could significantly accelerate radiological diagnosis while decreasing the cost for all—a win for humanity. In short, there stand to be potentially positive and negative impacts of continued AI developments. In fact, the rise of AI in more aspects of society has caused governments and nongovernmental organizations to draft ethics policies to guide future developments and use of AI. For example, the Organization for Economic Cooperation and Development's principles [3] include guidelines that AI should benefit people and the planet, drive sustainable development, be designed in a way that respects the rule of law and human rights, and that those who develop AI systems be held responsible for their function in line with the ethical guidelines. To this end, IEEE's slogan "Advancing Technology for Humanity" means that we have an implicit responsibility to ensure that these developments benefit all members of society.

In this article, we seek to update readers on the current state of the art (SOA) on applications of deep learning (DL) in problems of interest to the IEEE Antennas and Propagation Society (AP-S) community. Furthermore, a number of resources are provided to assist researchers to start using DL in their own research. Finally, we provide an outlook on DL's potential impact on the AP-S as well as how exciting new hardware advancements may lead to significant improvements in the way we currently perform computational electromagnetic (CEM) simulations.

## WHAT'S IN A NAME?

The quickening pace of technological developments in AI and machine learning (ML) has led to confusion with regard to what exactly these terms mean. Therefore, it is important to define AI and ML and distinguish these broader terms from the newer concept of DL in order to have a better understanding of how they are impacting our SOA.

AI can be considered as intelligence displayed by machines, which imitates cognitive tasks that humans associate with natural intelligence. If AI refers to any task resembling human intelligence, then ML refers to a subdomain of AI wherein a computer or robot must iteratively converge on a method for solving its problem or learn to solve a problem. This is often accomplished by recognizing patterns in data and inferring behaviors to make predictions. In ML, the learning process followed to achieve the final model is fundamental to its designation as AI because the learning itself is a demonstration of human intelligence, regardless of the simplicity of the final task.

While AI and ML have recently become increasingly mainstream concepts, they actually have a long history. Much of their rise in popularity is due to improvements made to artificial neural networks (ANNs). An ANN seeks to mimic the neural pathways in the brain and its mode of information processing (more information about their construction is provided in the following section). The first computational model for an ANN was introduce by McCulloch and Pitts in 1943. In 1956, AI was founded as an academic discipline and saw growing interest into the 1960s as the U.S. Department of Defense invested in the field. However, the 1970s were an "AI winter" due to a decrease in research funding, and researchers began exploring symbolic approaches for AI. Fortunately, AI research began to grow again in the mid-1980s as interest in ANNs renewed and accelerated with the advent of techniques such as backpropagation [4]. This interest continued into the 1990s and early 2000s when some specialized applications emerged, such as handwritten check signature recognition. However, broader interest in ANNs waned after the turn of the millennium, and it wasn't until recently that computational power became adequate to justify the training of ANNs for a broader range of problems.

In recent years, there have been tremendous efforts to accelerate the training of ANNs to be more efficient and for application to more complex problems. Perhaps the biggest advancement has been the practical development of deep neural networks (DNNs), which extend ANNs to include many hidden layers. DNNs improve the ability of ANNs to solve significantly more complex problems than shallow networks with few layers. Various network topologies, all based on DNNs, are classified under the umbrella term of DL, which has attracted much attention in the past few years. Figure 1 compares the relative search trends since 2004 as reported by Google for the terms *artificial neural network* and *deep learning*. Again, interest in ANNs had begun to wane during the early 2000s because the available computational resources were still lacking to take advantage of their full potential. However, several key events occurring in the past decade led to a tremendous rise in the popularity of DL and a resurgence of interest in ANNs. This rise began in 2009 with the "big bang" of DL when NVIDIA GPUs were first used to train DNNs. The leveraging of general-purpose GPU computing increased the training speed by two orders of magnitude compared to conventional CPU-based approaches [84]. Then in 2012, the DL revolution began when the DNN AlexNet won the ImageNet competition. Just three years later, another convolutional neural network (CNN) won the ImageNet competition and was the first to beat the accuracy benchmark set by human experts; this was the turning point when AI became better than humans at labeling images. Another momentous event happened in 2016 when the DNN-based AlphaGo system beat a human Go champion—an event previously thought impossible due to the extreme complexity of the game. Since then, we have seen the "democratization of AI" in which the widespread adoption of DL techniques has been enabled by a breadth of open source educational and software resources that are freely available to all. Moreover, the availability of cloud computing resources has enabled many companies to quickly exploit DL to improve their products and services. Before using DL in your research, it is helpful to understand how DNNs are constructed, how they work, common terminologies associated with them (see Table 1 for a summary of some of the most common DL-related terms), and to which types of problems they can be applied.

## DL BASICS

In the broadest sense, ML is best understood as learning by example [5]. By presenting a "smart" system with many examples of data or a desired function, patterns and features can be learned quickly without the need for the researcher to understand what is going on "under the hood" of the problem.

The blossoming of ML has coincided with noteworthy discoveries and recent successes of DL. These have been facilitated through some of DL's foundational properties: universal approximation and efficient backpropagation. As universal function approximators in both the feed-forward and recurrent formulations, deep models are applicable to an incredibly wide variety of problems from image classification to natural language processing [6], [7]. However, what takes DL from attractive to useful is efficient backpropagation [8]. Backpropagation is the process, typically found in supervised learning, where the error between a prediction and goal is back-fed through a deep model to directly recover an error gradient of all the network's free parameters. In practice, this means that finite differencing can be avoided. Thus, the act of computing an error gradient, which would otherwise be intractable for these topologies consisting of millions of free parameters, becomes no more computationally expensive than simply evaluating the model. Despite all the excitement about DL, training deep models is not much more complicated than a gradient descent optimization. Rather, the power of DL comes from the scale and scope of the models that can be trained.

### DESIGNING MODELS FOR LEARNING

Common to all deep models are a few building blocks that together enshrine these two critical properties (that is, universal approximation and efficient backpropagation). Deep models, as shown in Figure 2, are composed of a series of interconnected neurons called *nodes*. Signals applied to an input layer are passed on through various connection schemes to a hidden layer. Networks with at least three hidden layers are considered deep, each performing some function composition on the signal from the previous layer. At each node, all incoming connections from the previous layer are weighted and many options for activation functions are found in the literature and are commonly employed, but they are all nonlinear and monotonic. Early on, most used Sigmoid and tanh, but now the most popular is the rectified linear unit because it is faster to train with deep nets [9], [10]. The choice of activation function is often biologically inspired, with the goal of mimicking the nonlinear tipping point when a real neuron fires.

The simplest and most general topology of a DNN is a multilayer perceptron (MLP), where each hidden layer is fully connected to the next layer [11]. However, several variations on neuron and layer types have led to remarkably successful topologies. Convolutional layers constrain the weights used in the layer to be uniform, just with different connections. This is the basis of CNNs, which train many layers of image processing kernels that can be used for learned feature extraction [12]. Regularization techniques, like dropout and batch normalization, have also been introduced to help mitigate overfitting issues [13], [14]. Recurrent neural networks (NNs) not only pass signals interlayer but are also time varying and pass signals intralayer between time steps [15]. For these time-varying or sequential systems, long short-term memory nodes have made a significant impact in language models and natural language processing

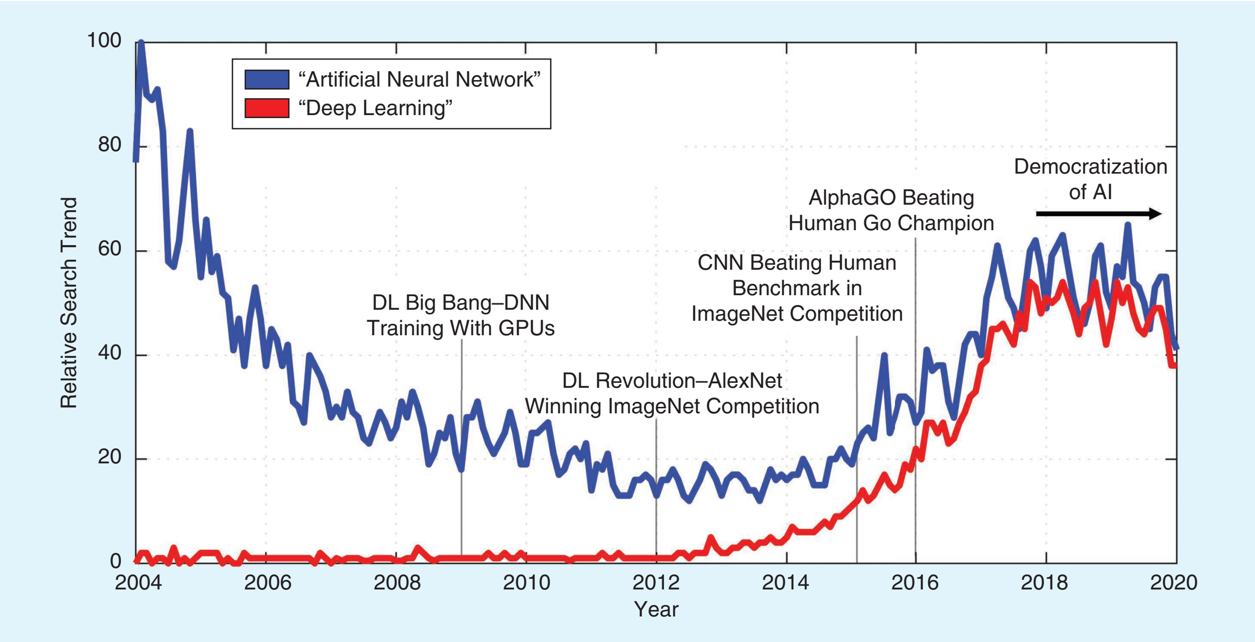


**FIGURE 1.** Relative search trends for topics "artificial neural network" (blue) and "deep learning" (red) since 2004, as reported by Google. Note the recent rise in interest in ANNs has coincided in step with the rise in popularity of DL. Key events such as the 2009 DL big bang and 2012 DL revolution have led to a significant rise in public awareness of DL.

**TABLE 1. A TAXONOMY OF AI AND DL TERMS.**

| Term | Abbreviation | Definition |
|---|---|---|
| Artificial intelligence | AI | Broadly, any engineered system that can perform actions exemplary of human or superhuman capabilities; colloquially, the threshold for the designation of AI is ill defined and always changing, especially across the many disciplines that utilize parts of the AI toolbox [25]. |
| Machine learning | ML | The study of algorithms and statistical methods used to perform a specific task without explicit instructions; the system improves, or "learns," through iteration; a subdomain of AI [26]. |
| Supervised learning | SL | An ML method that uses a large volume of data (input and output pairs) to train a model to predict new, unseen data similar to the training data; the two primary tasks are classification (determining which of a finite set of classes) and regression (predicting exact values on a continuous range of possibilities) [26]. |
| Unsupervised learning | UL | An ML method for learning unknown patterns in a data set; a type of self-organization; often used for clustering or feature extraction to better understand unlabeled data [26]. |
| Reinforcement learning | RL | An ML method in which a random agent attempts a given objective. This agent self-analyzes its actions and the outcomes of those actions to adjust its model through trial and error; similar to genetic algorithms, RL training schemes deal with the tradeoff of exploration versus exploitation to attempt new strategies or to perfect the best-learned strategies [27]. |
| Artificial neural network or neural network | ANN or NN | Computing systems based on a collection of connected units or nodes called *neurons*; for sufficiently large ANNs, proven to be universal approximators of any function; a computing system loosely inspired by biological NNs [28]. |
| Deep learning | DL | A method for using an ANN comprised of many hidden layers, ranging from a few to several hundreds of layers with potentially many thousands of nodes per layer [26], [29]. |
| Generative adversarial network | GAN | A class of ML systems wherein two ANNs compete to improve each of their models; the generative network learns to create inputs indistinguishable from the training data while the discriminative network learns to identify true data from data created by the generative network [30]. |
| Autoencoder | AE | An ANN designed to learn a reduced dimensionality of an input space; an autoencoder learns to map an input space back to itself while intermediately passing through a lower feature dimension; this lower dimension model can then be used as an approximation of the high-dimensional input space [31]. |
| Recurrent neural network | RNN | A class of ANNs that uses backward (or feedback) connections to enable a memory of internal states between successive passes to the network; RNNs can process collections of successive input data and are well posed for learning series data [15]. |
| Convolutional neural network | CNN | A class of ANNs that uses convolution to learn hierarchical patterns within data; CNNs learn generalized patterns across many spatial scales from their given data and are well posed for image data [12]. |
| Spiking neural network | SNN | Like neurons in the brain, spiking ANN activations develop iteratively, only firing once a threshold of consecutive activity is reached before the node is reset. There are currently asynchronous CMOS hardware solutions emulating this principle in development [32]. |
| Support vector machine | SVM | An ML method for determining the ideal hyperplane for separating a classification problem; given a set of training data, an SVM learns where to draw the line between classes in the hope that it will generalize to unseen data [24]. |
| Transfer learning | TL | A method for speeding up and improving the learning of a new task by using a previously trained ANN as the initialization for the new network; the method works best for closely related tasks and speed improvements can be greatest for very deep (multilayer) networks [33]. |

[16]. Although the scope of problems to which DL can be applied is extremely wide, those with DL experience are aware of just how sensitive the training process is to the designer's choice of network topology. For some problems, like the XOR problem, a single-layer solution would require an asymptotically large number of nodes, whereas for a deep network the number of nodes required is relatively small [17]. So although DL has the potential to solve any problem, it's up to the researcher 1) to choose the right topology and 2) to set up the learning environment to give the topology the best chance of converging and generalizing.

Although much effort has been dedicated to articulating principles for matching topologies with problem types, no one has yet found any hard-and-fast rules for this. Increasingly, researchers are looking into examples in the literature for the best topology for their problem. For a helpful chart that explains some of the basic topologies, see [18]. Coincident with the desire of researchers to reuse previously successful topologies, a relatively new approach called *transfer learning* (*TL*) has gained steam recently [19]. TL allows for a pretrained network to be repurposed for a new problem, with potentially massive reductions on the designer's side in the amount of data and training time required.

Regardless of the application or problem type, learning systems like these demand a lot of data. In fact, the development, acquisition, and curation of large data sets have become scholarly fields of their own, namely, data mining and big data [20]. Many open source data sets like MNIST [21] and data.gov [22] as well as online image and text-scraping tools have made the ML progress this decade possible. Additionally, there are many practices with regard to data sets that researchers have determined can aid the learning process. Common preprocessing tasks include data normalization (for example, zero-centered and unit variance) or selecting image sizes to take advantage of GPU architecture (for example, 224 × 224 pixels). These are motivated in part by properties of the backpropagation algorithm and network architecture, which are sensitive to biases and defects in the data. What remains to be seen is which kinds of data sets will become available in the next few years to DL researchers working in the areas of antennas and propagation (AP).

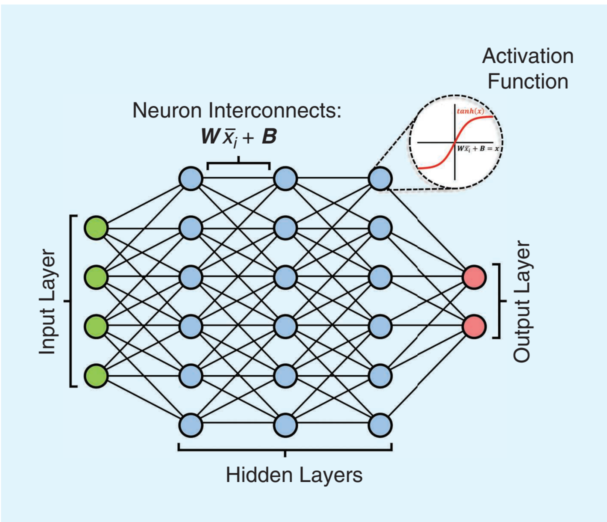


**FIGURE 2.** An example of a DNN, specifically a multilayer perceptron. Signals input on the left side travel through trained weights between layers to become inputs to activation functions for each node in the next layer.

### HOW DL IS USED FOR ML

ML can be broadly classified into three distinct categories: supervised, unsupervised, and reinforcement learning, each of which benefits from DL. These techniques are suitable for different kinds of problems, depending often on the kind of data available to the researcher. For instance, supervised learning is useful for direct function approximation, requiring data with labels [for example, input/output pairs, like $x$ and $f(x) = y$], whereas unsupervised learning clusters data by similarity (inputs only; for example, only $x$ and not $y$), and reinforcement learning discovers its own data through problem interaction.

Supervised learning is arguably the most prolific and successful type of DL, especially within the last two decades. This may be primarily due to its clear application as a function learner. The quintessential supervised learning example is that of classifying handwritten numerical digits. Using thousands of labeled images 0–9, an NN was trained to classify new digits [4]. However, this begs the question: Is supervised learning just glorified curve fitting? Although it certainly is the objective of supervised models to at least reproduce the examples it is shown, this is only the beginning of what these models are capable. Rather, the true goal of a supervised model is to go beyond the training data and make a model that generalizes to new examples of data that it has never seen before. Thus, to call supervised learning "just" interpolation would be an oversimplification. Indeed, work has been done to understand and validate the generalization of DNNs [23]. Although DNNs are the predominant form used in supervised learning (for example, MLPs and CNNs), other architectures like support vector machines (SVMs) are common as well [24]. A second domain of ML is unsupervised learning, a method for analyzing unknown data. An example might be identifying the number of unique voices talking in an audio clip. Unsupervised learning methods ideally would be able to cluster different audio segments into several groups by speaker and differentiate the number of speakers [34]. Deep models have found their way into this area in the form of autoencoders [31], although more traditional algorithms like $k$-means are commonly used as well.

The final area of ML, reinforcement learning, is somewhere between the other two. Without prelabeled data, reinforcement learning has an exploration component more akin to evolutionary algorithms. The system is given the ability both to generate its own new data and to analyze the quality of actions or predictions. Through trial and error, a reinforcement learning system is able to improve itself [35]. More traditional reinforcement learning can be found in the literature around temporal difference learning [36]. Newer work, like generative adversarial networks (GANs), have blurred the distinction between supervised and reinforcement learning by having an inherently reinforcement style system inscribed within a supervised learning framework [30].

## APPLICATIONS

Outside of the AP-S, the main AI applications are numerous, including image recognition, classification, and segmentation, natural language processing, drug discovery, fraud detection, and recommendation systems, to name a few. While the true utility of ML to the AP-S community is still being discovered, 2019 was a breakthrough year for the publication of AI-related techniques. In fact, a total of 109 articles mentioning either AI, ML, AANs, or DL were published by the AP-S (excluding conference proceedings), which represents an enormous increase over the preceding years (see Figure 3). Moreover, November 2019 saw the publication of the *IEEE Antennas and Wireless Propagation Letters* "special cluster" issue on ML applications in electromagnetics (EM) and AP, which included 25 articles on the SOA applications of ML in our field [37]. Most of the articles relied on some aspect of DL as the main driving force behind their developments. As interest and knowledge of DL grows, the number of new articles is expected to increase in 2020 and beyond.

Some recent review work by different contributors in the field has sought to understand how ML is best being applied in AP. The principles of learning by example in EM were introduced in [5], while [38] provides background for DNNs in EM up through early 2019. Other reviews have looked at ML in a subfield like remote sensing [39], next-generation wireless technologies [40], and ground-penetrating radar [41]. Problems that do not lend themselves well to DL are those where data are currently extremely "expensive" to acquire. This could be in terms of computational expenses [for example, time or random-access memory (RAM) limitations] or when measurements are extremely difficult or prohibitive to perform. It is expected that improvement to DNN algorithms and increases in computational power in the coming years will make such problems tractable for DL.

In the following sections, a sampling of recent work that applies ML in AP is broken down into a few categories that best delineate the different ways ML is being used, namely, forward modeling, inverse modeling, remote sensing, and inverse design.

### *FORWARD MODELING*

One highly attractive application of DL within AP is integration with or replacement of CEM solvers. This is the so-called forward problem of EM design—analyzing the scattering and matching performance of conductive or dielectric structures. Through DL, there is a possibility of large speedups by bypassing traditional solvers like method of moments (MOM), finite-element method (FEM), and finite difference time domain (FDTD). The approaches taken in these forward-modeling articles can predominantly be categorized as supervised learning since the goal is to learn some sort of specific scattering behavior. The latest contributions in this domain can be roughly categorized as general forward modeling, channel estimation, and TL.

Recent contributions in this area have been more general in some cases, like a Poisson's equation solver, which uses CNNs to analyze 2D statics [42]. Similarly, CNNs have also been used to take images of high residual error surface currents on various conducting structures and convert those to images of low-resolution error surface currents [43]. In both cases, a DL method lays the groundwork for large speedups of general solvers in the future. Scattering has also been modeled for more constrained types of structures and environments. For instance, Massa et al. used kriging to learn the scattering behavior of 2D conducting reflectarray elements, providing a number of base formulations for metaelements while learning their scattering based on a variation of their defining variables [44]. Also, Gaussian processes were used to perform direction of arrival estimation for random arrays of antennas. This technique was shown to be competitive or better than the SOA in several tests [45].

Channel state information (CSI) describes the quality of a connection between two points in space, accounting for path loss, scattering, power decay, and so forth. The process of discovering the CSI for an environment, called *channel estimation (CE)*, has also been a point of many DL applications. In [46], the authors generated an ensemble of learned models of the received signal strength using an evolutionary algorithm called the *Salp Swarm Algorithm*. A drone lifted a cell phone to several locations in space, and the authors used multiple measurements to construct a model of the path loss to different spatial locations away from a transmitting base station. In another article, an analysis of real satellite communication in conjunction with low-cost weather instruments was accurately fit using an MLP DNN to predict channel excess attenuation, and the authors believe this method will be a lower-cost option

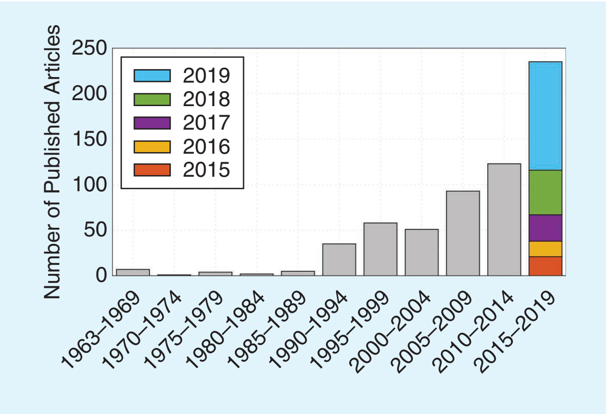


**FIGURE 3.** The number of articles that have appeared in AP-S publications (that is, *IEEE Transactions on Antennas and Propagation*, *IEEE Antennas and Wireless Propagation Letters*, and *IEEE Antennas and Propagation Magazine*) since 1963 grouped in five-year increments. The year 2019 represented an explosion of DL/ML in the AP community as nearly as many DL/ML articles were published as in the previous four years combined.

than other existing solutions [47]. The authors of [48] used ML to learn the diffraction loss of antennas transmitting around the human body via Gaussian process regression using real measurements of human bodies.

Perhaps the most exciting avenue for DL in forward modeling is that of TL of EM principles for scaling up problem sizes. For example, the authors of [49] developed a method for learning scattering responses from simpler targets, then compositing them together via TL to classify complex aircraft targets, outclassing the non-DL SOA (see Figure 4). With this method they developed a useful general model from a relatively small amount of data, which is very important for electrically large CEM problems. We'd like to see more work in the future in which TL can be used to leverage forward modeling to produce successful and powerful inverse scattering (IS) solutions and remote sensing solutions.

## IS AND REMOTE SENSING

Arguably, the most striking uses of DL in the AP-S community have been in the areas of IS as well as remote sensing and classification. IS uses a few receiving antennas to figure out the shape of a scattering structure, typically a highly underdetermined problem. Many of these examples were shown to beat the SOA by a significant margin. The new work discussed next falls into the categories of general IS, remote classification, and transmitter localization.

DL of IS problems has yielded several improvements over the SOA in a relatively short time. DeepNIS is a DNN for nonlinear electromagnetic IS that very effectively recovers scattering structures from a small number of receiving antennas [50]. This article and other similar approaches have used CNNs to perform IS using scattering simulated by dielectrics in the shape of handwritten digits from the MNIST database [51]. In [52], the authors looked at how the quality of the IS solution varies with changes to the network depth. They considered a series of different depths and showed that when a measure of input/output variation reaches a plateau, there is coincident stabilizing of the network training as well as generalization (see Figure 5). In a more application-focused version of the same concept, CNNs are used to perform IS to determine, via ground-penetrating radar, regions of increased resistivity underground. This method of "geophysical inversion" produces instantaneous results, as opposed to having to follow a gradient-based method every time, as is the case with other IS methods [53].

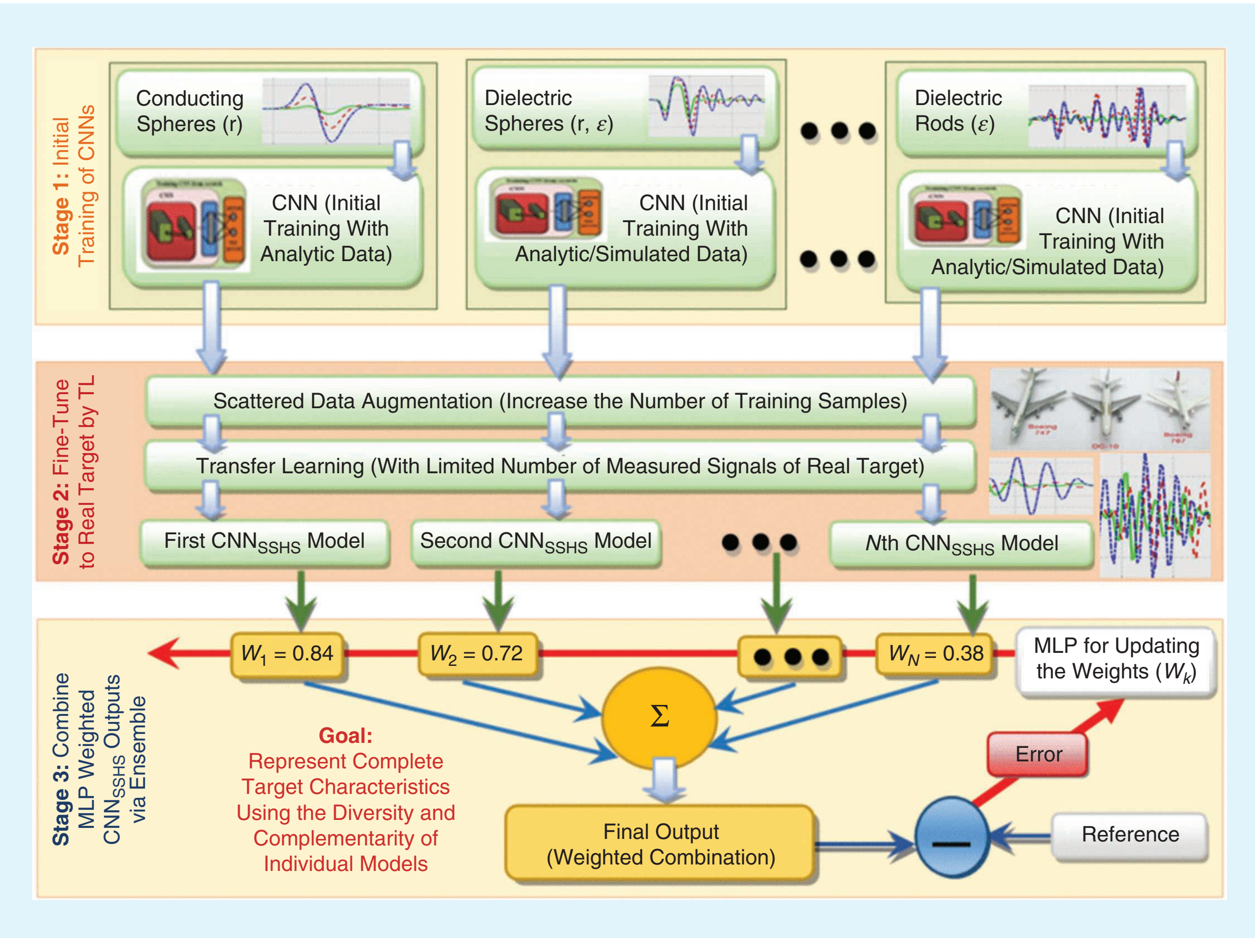


**FIGURE 4.** In this DL forward scattering/remote sensing hybrid, multiple CNNs were trained to discover the defining variables (for example, radius and permittivity) of several simple conducting and dielectric structures based on time-domain backscatter. Once trained, these CNNs were then integrated, via TL, into a method for classifying aircraft to a high degree of accuracy despite very limited actual aircraft test data [38].

Besides the specific case of using DL to recover structures from scattering analysis, many other articles have a broader scope, attempting to classify various features or metrics of an environment based on scattering data. For instance, [54] tested several different ML techniques (including ANN, SVM, and $k$-NN) to identify via S-parameters whether or not a ground-embedded anchor rod for high-power transmission has a fault. The authors achieved a high accuracy in predicting problematic anchors when training against data generated via an FEM. Similar uses have been to detect explosives underground with 87.02% accuracy using an SVM [55] or to measure the rough crowding of a room using a simple dipole, software-defined radio, and random forest classification (RFC) [56].

Whereas in the previous section DL of CSI via forward modeling was used to perform CE, an inverse method can instead localize emitters based on CSI. There are numerous examples in the literature of this use case, employing almost every kind of ML model demonstrated to work. RFC was used to localize emitters with some confidence region, trained on ray tracing a map of the environment [57]. In a more application-based take, a SVM was used to predict which of several rectennas in an environment should be active to best "burst charge" the system for a longer battery life. The model, based on real measurements, predicts where Wi-Fi will be strongest in an indoor environment, which yielded an average efficiency increase of 5% to 25% [58]. Finally, CSI along with other radio-frequency (RF) metrics were used to form a high-dimensional description of an environment, then $k$-means clustering (unsupervised learning) identified groupings, which could then be compared with known environments to do after-the-fact environment classification [59].

## INVERSE DESIGN

The improvements available through DL in AP have been the most obvious and sudden in the area of IS. Nevertheless, the domain with perhaps a larger growth in interest is that of inverse design. For instance, DL methods are beginning to have significant impact in optical metadevice design and optimization [60]. The articles discussed next have been broken down into two rough categories: DL models used in optimization and reinforcement learning. Distinguishing between these types is not always clear since reinforcement learning is in many ways an optimization procedure itself.

Much of the new inverse design work with DL has focused on using deep models as surrogate models before or after optimization. In [61], bat optimization was used to first generate wide-nulling antenna array solutions. A general regression NN was then trained on optimized results before being used to produce new patterns. In another article, NNs were trained on

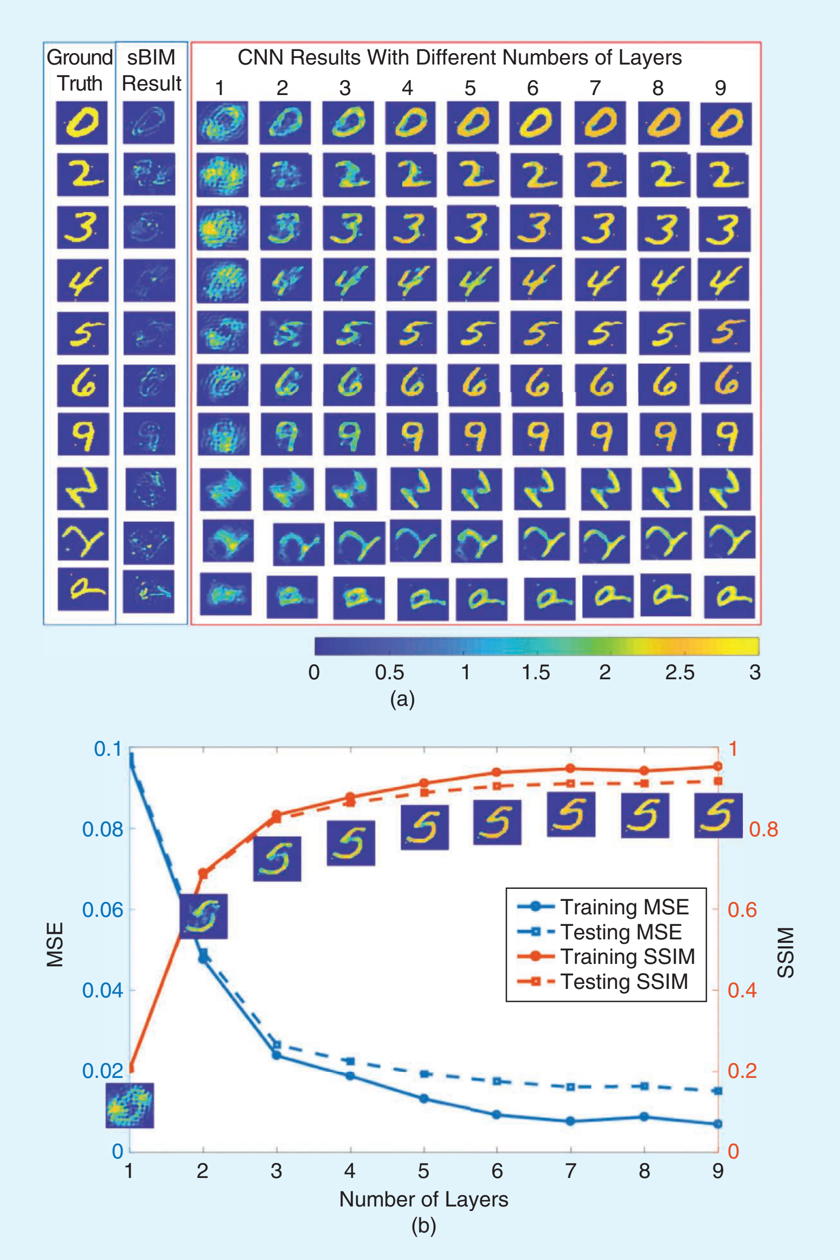


**FIGURE 5.** CNNs with different numbers of layers were trained to perform IS and reconstruct a dielectric shape based on scattered fields. (a) Images from the MNIST handwritten numbers data set were evaluated numerically as training data. (b) The mean square error (MSE) of the reconstructions decreases as the number of layers used in the CNN increases [52]. This solution along with other similar DL techniques [50], [51] outperform the previous best non-ML approaches by a wide margin. SSIM: structure similarity index.

metasurface shapes to learn diffraction efficiencies, then inverse methods were used to generate a shape with desired diffraction efficiency [62]. SVMs have been a popular candidate model for inverse design as well. They have been used alongside MOM or other full-wave techniques for the design of reflectarray unit cells [63], [64]. Furthermore, [65] has used DL to estimate the probability distributions of design variable uncertainties to speed up the optimization problem and promote RF design robustness. Finally, inverse DNNs can generate patterns based on input spectra [66], and a double-NN method has been used for generating and optimizing chiral metamaterials [67].

Perhaps the most exciting integration of DL into the inverse design process is through various forms of reinforcement learning. Preeminent in this regard are GANs, which train on both supervised examples and new unsupervised designs that are invented during the training process. Although not explicitly reinforcement learning methods, their inherently explorative properties make them very similar. The resultant "generator" networks have been used to find metaelement shapes with some desired transmittance spectrum [68], [69]. A multidiscriminator GAN was used for three-layer RF metasurface [70] and single-layer RF metasurface [71] design. Purer uses of reinforcement learning have taken the route of deep Q-learning—reinforcement for selecting circular unit cell diameters and material properties [72] and for generating nanoparticles [73].

## GETTING STARTED WITH DL

For those who are unfamiliar with DL, it may not be clear where to begin using these techniques. Essentially, there are three main areas researchers need to be familiar with regarding DNNs and DL: 1) education, 2) software, and 3) hardware. Fortunately, there exists a wide range of freely available educational and software resources. As for hardware, most readers will be able to use these techniques to some degree on conventional workstation computers, although significant improvements in DNN model training can be realized with specialized hardware. In the following section, we provide an overview of some of the most popular and useful resources readers can take advantage of to jump start the integration of DL and related techniques into their research.

### EDUCATION

One major cause of the democratization of AI is the open source movement and, with that, the availability of free coursework, which everyone can use to educate themselves on the fundamentals of DL. Some great educational resources include Andrew Ng's Coursera courses [74] for DL, free university lectures through Massachusetts Institute of Technology [75] and Stanford University [76], and fast.ai's courses [77] developed for making ML simple and accessible. DL's rapidly developing popularity and low barrier to entry have pushed ML into the forefront of open source tools, tutorials, and technology. Within these offerings, there are different kinds of courses users can take. Some provide more theoretical background about how NNs work (for example, Andrew Ng's Coursera courses [74]), whereas others focus on learning the tools first rather than leaning so heavily on the underlying mathematics and the technical side (for example, fast.ai's practical ML for programmers courses [77]). Moreover, many leaders in the field actively produce educational material, often for free, and develop tools specifically for DL newcomers. With these courses, one can quickly come up to speed with the fundamentals and some of the latest technological breakthroughs in the field of AI and DL. Moreover, many of these courses also integrate hands-on approaches with the use of software.

### SOFTWARE

Like educational courses, a number of open source software tools for DL can be freely used. Some of the most popular software tools include PyTorch [78], TensorFlow [79], and the MATLAB Deep Learning Toolbox [80]. Most of these tools offer the ability to train various DNN topologies with user-specified numbers of layers and nodes. Moreover, they allow users to perform training with backpropagation, create custom activation functions, and vary features such as the batch size, among others. Furthermore, most provide utilities for acquiring and managing data used for DNN training. Also, users can often load popular DNN topologies such as ResNet [81] or U-Net [82] to start modeling their data with a proven network. Ideally, these preexisting networks are chosen for problems that are similar to the original applications from the literature. It should be noted that some of these tools have different features and interfaces, so readers are encouraged to try multiple options to see what serves them best in integrating DL into their research. Finally, while capable of running entirely on the CPU, nearly all of the tools have extensions to work with NVIDIA GPUs to accelerate DNN network training.

### HARDWARE

While all of the software tools listed work with conventional x86 CPUs, there is no denying that the use of NVIDIA GPUs can greatly accelerate the training of DNNs. While many workstation computers will be compatible with the GPU-acceleration features offered by the aforementioned software tools, researchers interested in using DL for complex problems with lots of data may consider acquiring a dedicated GPU computing platform to satisfy their needs. These machines can come in a variety of form factors, including traditional workstation towers and special rackmount servers. Figure 6 shows a multi-GPU server intended for high-performance scientific computing and DL whose primary computing power is GPU rather than CPU based. Moreover, one such vendor, Lambda Labs, publishes information about what are the best cards to purchase for DL research both in terms of bleeding edge power as well as price and performance [83]. Having an awareness of what the latest product offerings are as well as how they perform can help researchers save money by purchasing the option that provides the best bang for their buck.

## OUTLOOK

The current status of DL research is exposing the SOA to constant disruption. For example, applications such as IS have already seen their SOA significantly improved by exploiting DL techniques.

Moreover, every new DL advancement, whether it be software or hardware related, can lead to an improvement in the SOA. For example, DeepMind has recently developed an approximation technique called *Sideways* for accelerating backpropagation up to five times for some CNN problems compared to conventional approaches [85]. This development could significantly reduce the cost of training DNNs as well as make some currently intractable problems feasible. Simply put, DL positions researchers in a world where the goalposts are constantly moving. Still, rest assured that DL cannot replace the power of analytical theory or the knowledge of experienced engineers. While DL has had a huge impact in a few key areas of interest to the AP-S community, future developments will enable DL to impact other areas.

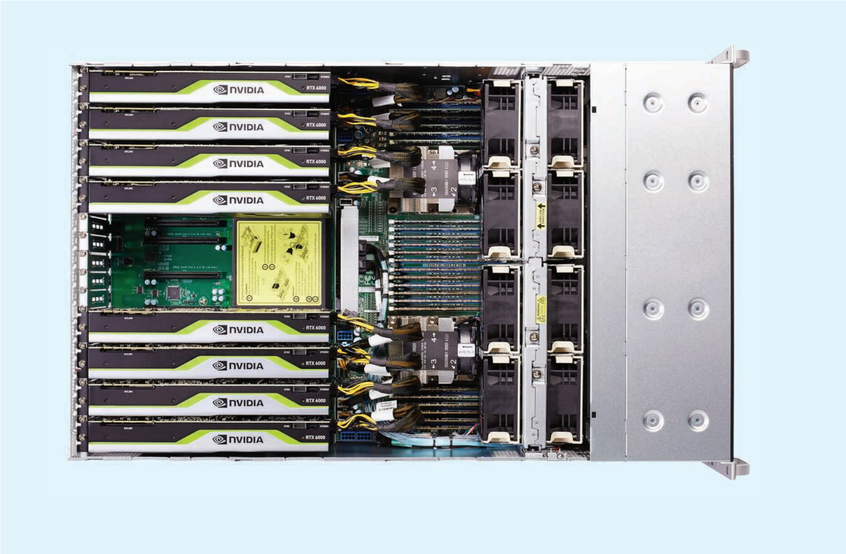

**FIGURE 6.** A high-performance GPU server with eight GPUs designed for DL and GPU-accelerated scientific computing. (Source: Adapted from [84]; used with permission from from Lambda Labs.)

Nevertheless, acquiring big data sets is still a tremendously expensive task for a number of RF and optical problems, especially those requiring full-wave modeling. Still, more data can be generated with increasingly efficient forward solvers and more powerful computational hardware. To this end, the multi-GPU workstations and compute nodes that are being produced to accelerate DNN model training are an extremely attractive option for accelerating CEM simulations as well. While adapting CEM codes to run on the GPU has been explored for some time [86], several challenges have limited the potential of GPU-accelerated CEM codes: 1) having to rewrite established code to run on the GPU, 2) the limited amount of dedicated GPU RAM compared to what is available to the CPU, 3) the scalability to many GPUs due to communication bottlenecks, and 4) the input–output communications bottleneck between the GPU and CPU, which robs many potential performance gains [87]. For these and other reasons, most commercial CEM codes either do not support or have extremely limited support for GPU acceleration. However, many advancements have recently been made that address these issues. In the past two years, the amount of dedicated GPU RAM has increased tremendously, with the 2017 NVIDIA Titan V cards having 12 GB and 2019 Titan RTX seeing a doubling to 24 GB. In a multi-GPU system, the amount of dedicated GPU RAM is now finally comparable to conventional x86-based workstation computers with which we are all familiar. Moreover, the recently introduced NVLink significantly increases the data transfer bandwidth between GPUs in multi-GPU compute nodes and has enabled significant acceleration of DNN training [88]. Furthermore, the recently developed NVIDIA GPUDirect Storage technology enables the GPU to bypass communication with the CPU and directly pass data to fast nonvolatile memory express storage through a blazing-fast peripheral component interconnect express bus for a remarkably improved data transfer rate. CEM solvers including MOM and FDTD can be accelerated with GPUs [89], while the element-level decomposition ability of the discontinuous Galerkin time-domain method makes it an extremely attractive candidate for GPU computing platforms [90]. Catching up with these recent hardware developments will help the AP-S community solve CEM problems faster while enabling the development of improved DL models.

For now, DL should be considered a tool to aid designers in both forward and inverse problems as well as inverse design (optimization). Faster solvers and more powerful optimizers are still needed, and developments in these areas will synergize with further developments in DL. Moreover, the increase in dedicated computational power is also a boon to engineers, although we need to catch up with these new developments. CEM codes should continue to be developed to exploit these new hardware advancements. As we begin this new decade, it will be interesting to see where these developments take us and what we will say when we look back at this period in time. It is a very interesting time indeed.

## ACKNOWLEDGMENT

The research is supported in part by DARPA EXTREME (contract HR00111720032).

## AUTHOR INFORMATION

***Sawyer D. Campbell*** (sawyer@psu.edu) is with the Pennsylvania State University, University Park. His research interests include applied computational electromagnetics, optimization and inverse-design methods, deep learning, nanophotonics, and optical design. He is a Member of IEEE.

***Ronald P. Jenkins*** (rpj1@psu.edu) is a Ph.D. student at the Pennsylvania State University, University Park. His research interests include computational electromagnetics, evolutionary algorithms, deep learning, multiobjective optimization, and nanophotonics. He is a Student Member of IEEE.

***Philip J. O'Connor*** (pjo5097@psu.edu) is an M.S. student in the Department of Electrical Engineering, the Pennsylvania State University, University Park. His research interests include machine learning applications in electrical engineering, specifically in applied electromagnetics.

***Douglas H. Werner*** (dhw@psu.edu) is with the Pennsylvania State University, University Park. His research interests include computational electromagnetics, optimization methods, advanced antenna design, and metamaterials. He is a Fellow of IEEE, Institution of Engineering and Technology, Electromagnetics Academy, Applied Computational Electromagnetics Society, and the Optical Society.

## REFERENCES

[1] S. Talty and J. Julien. "What will our society look like when artificial intelligence is everywhere?" Smithsonian Magazine. https://www.smithsonianmag.com/innovation/artificial-intelligence-future-scenarios-180968403/ (accessed Jan. 21, 2020).

[2] "Machine learning: The power and promise of computers that learn by example," Royal Society (Great Britain), London, 2017. [Online]. Available: https://royalsociety.org/~/media/policy/projects/machine-learning/publications/machine-learning-report.pdf

[3] "OECD principles on artificial intelligence," Organisation for Economic Co-operation and Development, Paris, France. Accessed: Jan. 21, 2020. [Online]. Available: https://www.oecd.org/going-digital/ai/principles/

[4] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989. doi: 10.1162/neco.1989.1.4.541.

[5] A. Massa, G. Oliveri, M. Salucci, N. Anselmi, and P. Rocca, "Learning-by-examples Techniques as Applied to Electromagnetics," *J. Electromagn. Waves Appl.*, vol. 32, no. 4, pp. 516–541, Mar. 2018. doi: 10.1080/09205071.2017.1402713.

[6] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signal Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989. doi: 10.1007/BF02551274.

[7] A. M. Schäfer and H. G. Zimmermann, "Recurrent neural networks are universal approximators," in *Proc. Int. Conf. Artificial Neural Networks (ICANN 2006)*, Berlin, 2006, pp. 632–640. doi: 10.1007/11840817_66.

[8] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade*, 2nd ed., G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 9–48.

[9] S. Sharma. "Activation functions in neural networks." Medium. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 (accessed Jan. 21, 2020).

[10] "Activation function." Wikipedia. https://en.wikipedia.org/wiki/Activation_function (accessed Jan. 19, 2020).

[11] H. Ramchoun, "Multilayer perceptron: Architecture optimization and training," *Int. J. Interactive Multimedia Artif. Intell.*, vol. 4, no. 1, pp. 26–30, 2016. doi: 10.9781/ijimai.2016.415.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017. doi: 10.1145/3065386.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," Mar. 2015, arXiv:1502.03167.

[15] A. Cleeremans, D. Servan-Schreiber, and J. L. McClelland, "Finite state automata and simple recurrent networks," *Neural Comput.*, vol. 1, no. 3, pp. 372–381, Sept. 1989. doi: 10.1162/neco.1989.1.3.372.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. doi: 10.1162/neco.1997.9.8.1735.

[17] D. S. Touretzky and D. A. Pomerleau, "What's hidden in the hidden layers?" *BYTE*, vol. 14, no. 8, pp. 227–233, Aug. 1989. doi: 10.5555/77174.77190.

[18] A. Tch. "The mostly complete chart of neural networks, explained." Medium. https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464 (accessed Jan. 21, 2020).

[19] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY: Curran Associates, 2014, pp. 3320–3328.

[20] D. Che, M. Safran, and Z. Peng, "From big data to big data mining: challenges, issues, and opportunities," in *Database Systems for Advanced Applications*, Berlin, Germany: Springer-Verlag, 2013, pp. 1–15.

[21] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," distributed by NIST, Gaithersburg, MD. Accessed: Jan. 28, 2020. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[22] Data.gov. Accessed: Jan. 28, 2020. [Online]. Available: https://www.data.gov/

[23] J. Kindermann and A. Linden, "Inversion of neural networks by gradient descent," *Parallel Comput.*, vol. 14, no. 3, pp. 277–286, Aug. 1990. doi: 10.1016/0167-8191(90)90081-J.

[24] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst.*, vol. 13, no. 4, pp. 18–28, July 1998. doi: 10.1109/5254.708428.

[25] D. L. Poole, A. K. Mackworth, and R. Goebel, *Computational Intelligence: A Logical Approach*. New York: Oxford Univ. Press, 1998.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. doi: 10.1038/nature14539.

[27] I. Salian. "SuperVize Me: What's the difference between supervised, unsupervised, semi-supervised and reinforcement learning?" NVIDIA Blog. https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/ (accessed Jan. 23, 2020).

[28] "Artificial neural network." Wikipedia. https://en.wikipedia.org/wiki/Artificial_neural_network (accessed Jan. 20, 2020).

[29] "Deep learning," NVIDIA Developer, Santa Clara, CA, Feb. 18, 2015. Accessed: Jan. 21, 2020. [Online]. Available: https://developer.nvidia.com/deep-learning

[30] I. Goodfellow et al., "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY: Curran Associates, 2014, pp. 2672–2680.

[31] P. Baldi, "Autoencoders, unsupervised learning and deep architectures," in *Proc. 2011 Int. Conf. Unsupervised Transfer Learning Workshop*, Washington, DC, 2011, vol. 27, pp. 37–50. doi: 10.5555/3045796.3045801.

[32] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997. doi: 10.1016/S0893-6080(97)00011-7.

[33] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. E. Soria, J. Martin, R. Magdalena, M. Martinez, and A. Serrano, Eds. Hershey, PA: IGI Global, 2009.

[34] C. Xu et al., "Crowd++: Unsupervised speaker count with smartphones," in *Proc. 2013 ACM Int. Joint Conf. Pervasive Ubiquitous Computing*, Zurich, Switzerland, pp. 43–52.

[35] "Learning dexterity." OpenAI. https://openai.com/blog/learning-dexterity/ (accessed Jan. 23, 2020).

[36] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach Learn.*, vol. 3, no. 1, pp. 9–44, Aug. 1988. doi: 10.1007/BF00115009.

[37] Z. Bayraktar, D. E. Anagnostou, S. K. Goudos, S. D. Campbell, D. H. Werner, and A. Massa, "Guest Editorial: Special cluster on machine learning applications in electromagnetics, antennas, and propagation," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2220–2224, Nov. 2019. doi: 10.1109/LAWP.2019.2945426.

[38] A. Massa, D. Marcantonio, X. Chen, M. Li, and M. Salucci, "DNNs as applied to electromagnetics, antennas, and propagation—A review," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2225–2229, Nov. 2019. doi: 10.1109/LAWP.2019.2916369.

[39] X. X. Zhu et al., "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017. doi: 10.1109/MGRS.2017.2762307.

[40] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017. doi: 10.1109/MWC.2016.1500356WC.

[41] X. L. Travassos, S. L. Avila, and N. Ida, "Artificial neural networks and machine learning techniques applied to ground penetrating radar: A review," *Appl. Comput. Inform.*, early access, Oct. 2018. doi: 10.1016/j.aci.2018.10.001.

[42] W. Tang et al., "Study on a Poisson's equation solver based on deep learning technique," in *Proc. 2017 IEEE Electrical Design Advanced Packaging Systems Symp. (EDAPS)*, pp. 1–3. doi: 10.1109/EDAPS.2017.8277017.

[43] B. Karaosmanoglu and O. Ergul, "Visual result prediction in electromagnetic simulations using machine learning," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2264–2266, Nov. 2019. doi: 10.1109/LAWP.2019.2939762.

[44] M. Salucci, L. Tenuti, G. Oliveri, and A. Massa, "Efficient prediction of the EM response of reflectarray antenna elements by an advanced statistical learning method," *IEEE Trans. Antennas Propag.*, vol. 66, no. 8, pp. 3995–4007, Aug. 2018. doi: 10.1109/TAP.2018.2835566.

[45] A. Gupta, C. G. Christodoulou, J. L. Rojo-Alvarez, and M. Martinez-Ramon, "Gaussian processes for direction-of-arrival estimation with random arrays," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2297–2300, Nov. 2019. doi: 10.1109/LAWP.2019.2934944.

[46] S. K. Goudos and G. Athanasiadou, "Application of an ensemble method to UAV power modeling for cellular communications," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2340–2344, Nov. 2019. doi: 10.1109/LAWP.2019.2926784.

[47] L. Bai, C.-X. Wang, Q. Xu, S. Ventouras, and G. Goussetis, "Prediction of channel excess attenuation for satellite communication systems at Q-band using artificial neural network," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2235–2239, Nov. 2019. doi: 10.1109/LAWP.2019.2932904.

[48] M. Khalily, T. W. C. Brown, and R. Tafazolli, "Machine-learning-based approach for diffraction loss variation prediction by the human body," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2301–2305, Nov. 2019. doi: 10.1109/LAWP.2019.2929289.

[49] M. Alper Selver, T. Toprak, M. Secmen, and E. Y. Zoral, "Transferring synthetic elementary learning tasks to classification of complex targets," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2267–2271, Nov. 2019. doi: 10.1109/LAWP.2019.2930602.

[50] L. Li, L. G. Wang, F. L. Teixeira, C. Liu, A. Nehorai, and T. J. Cui, "DeepNIS: Deep neural network for nonlinear electromagnetic inverse scattering," *IEEE Trans. Antennas Propag.*, vol. 67, no. 3, pp. 1819–1825, Mar. 2019. doi: 10.1109/TAP.2018.2885437.

[51] H. M. Yao, W. E. I. Sha, and L. Jiang, "Two-step enhanced deep learning approach for electromagnetic inverse scattering problems," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2254–2258, Nov. 2019. doi: 10.1109/LAWP.2019.2925578.

[52] L. Li, L. G. Wang, and F. L. Teixeira, "Performance analysis and dynamic evolution of deep convolutional neural network for electromagnetic inverse scattering," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2259–2263, Nov. 2019. doi: 10.1109/LAWP.2019.2927543.

[53] V. Puzyrev, "Deep learning electromagnetic inversion with convolutional neural networks," *Geophys. J. Int.*, vol. 218, no. 2, pp. 817–832, Aug. 2019. doi: 10.1093/gji/ggz204.

[54] D. C. P. Barbosa et al., "Machine learning approach to detect faults in anchor rods of power transmission lines," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2335–2339, Nov. 2019. doi: 10.1109/LAWP.2019.2932052.

[55] S. Gutierrez, F. Vega, F. A. Gonzalez, C. Baer, and J. Sachs, "Application of polarimetric features and support vector machines for classification of improvised explosive devices," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2282–2286, Nov. 2019. doi: 10.1109/LAWP.2019.2915047.

[56] T. C. de Brito Guerra, P. M. de Santana, M. M. de Medeiros Campos, M. de Oliveira Mattos, A. A. M. de Medeiros, and V. A. de Sousa, "RF-driven crowd-size classification via machine learning," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2321–2324, Nov. 2019. doi: 10.1109/LAWP.2019.2932076.

[57] M. N. de Sousa and R. S. Thoma, "Applying random forest and multipath fingerprints to enhance TDOA localization systems," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2316–2320, Nov. 2019. doi: 10.1109/LAWP.2019.2934466.

[58] A. Eid, S. Mghabghab, J. Costantine, M. Awad, and Y. Tawk, "Support vector machines for scheduled harvesting of Wi-Fi signals," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2277–2281, Nov. 2019. doi: 10.1109/LAWP.2019.2943250.

[59] M. I. AlHajri, N. T. Ali, and R. M. Shubair, "Indoor localization for IoT using adaptive feature selection: A cascaded machine learning approach," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2306–2310, Nov. 2019. doi: 10.1109/LAWP.2019.2915047.

[60] S. D. Campbell, D. Sell, R. P. Jenkins, E. B. Whiting, J. A. Fan, and D. H. Werner, "Review of numerical optimization techniques for meta-device design [Invited]," *Opt. Mater. Express*, vol. 9, no. 4, pp. 1842–1863, Apr. 2019. doi: 10.1364/OME.9.001842.

[61] X. Xiao and Y. Lu, "Data-based model for wide nulling problem in adaptive digital beamforming antenna array," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2249–2253, Nov. 2019. doi: 10.1109/LAWP.2019.2925419.

[62] S. Inampudi and H. Mosallaei, "Neural network based design of metagratings," *Appl. Phys. Lett.*, vol. 112, no. 24, p. 241,102, June 2018. doi: 10.1063/1.5033327.

[63] D. R. Prado, J. A. López-Fernández, G. Barquero, M. Arrebola, and F. Las-Heras, "Fast and accurate modeling of dual-polarized reflectarray unit cells using support vector machines," *IEEE Trans. Antennas Propag.*, vol. 66, no. 3, pp. 1258–1270, Mar. 2018. doi: 10.1109/TAP.2018.2790044.

[64] D. R. Prado, J. A. López-Fernández, M. Arrebola, and G. Goussetis, "Support vector regression to accelerate design and crosspolar optimization of shaped-beam reflectarray antennas for space applications," *IEEE Trans. Antennas Propag.*, vol. 67, no. 3, pp. 1659–1668, Mar. 2019. doi: 10.1109/TAP.2018.2889029.

[65] D. Kan, D. Spina, S. De Ridder, F. Grassi, H. Rogier, and D. V. Ginste, "A machine-learning-based epistemic modeling framework for textile antenna design," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2292–2296, Nov. 2019. doi: 10.1109/LAWP.2019.2933306.

[66] I. Malkiel, M. Mrejen, A. Nagler, U. Arieli, L. Wolf, and H. Suchowski, "Plasmonic nanostructure design and characterization via deep learning," *Light Sci. Appl.*, vol. 7, no. 1, pp. 60, Dec. 2018. doi: 10.1038/s41377-018-0060-7.

[67] W. Ma, F. Cheng, and Y. Liu, "Deep-learning-enabled on-demand design of chiral metamaterials," *ACS Nano*, vol. 12, no. 6, pp. 6326–6334, June 2018. doi: 10.1021/acsnano.8b03569.

[68] Z. Liu, D. Zhu, S. P. Rodrigues, K.-T. Lee, and W. Cai, "Generative model for the inverse design of metasurfaces," *Nano Lett.*, vol. 18, no. 10, pp. 6570–6576, Oct. 2018. doi: 10.1021/acs.nanolett.8b03171.

[69] D. Liu, Y. Tan, E. Khoram, and Z. Yu, "Training deep neural networks for the inverse design of nanophotonic structures," *ACS Photon.*, vol. 5, no. 4, pp. 1365–1369, Apr. 2018. doi: 10.1021/acsphotonics.7b01377.

[70] J. A. Hodge, K. Vijay Mishra, and A. I. Zaghloul, "Multi-discriminator distributed generative model for multi-layer RF metasurface discovery," in *Proc. 2019 IEEE Global Conf. Signal Information Processing (GlobalSIP)*, Ottawa, ON, Canada, pp. 1–5. doi: 10.1109/GlobalSIP45357.2019.8969135.

[71] J. A. Hodge, K. V. Mishra, and A. I. Zaghloul, "RF metasurface array design using deep convolutional generative adversarial networks," in *Proc. 2019 IEEE Int. Symp. Phased Array System Technology (PAST)*, Waltham, MA, pp. 1–6. doi: 10.1109/PAST43306.2019.9020794.

[72] I. Sajedian, H. Lee, and J. Rho, "Double-deep Q-learning to increase the efficiency of metasurface holograms," *Sci. Rep.*, vol. 9, no. 1, p. 10,899, Dec. 2019. doi: 10.1038/s41598-019-47154-z

[73] I. Sajedian, T. Badloe, and J. Rho, "Optimisation of colour generation from dielectric nanostructures using reinforcement learning," *Opt. Express*, vol. 27, no. 4, pp. 5874–5883, Feb. 2019. doi: 10.1364/OE.27.005874.

[74] "Deep learning." Coursera.org. https://www.coursera.org/specializations/deep-learning (accessed: Jan. 26, 2020).

[75] "Machine learning." OpenCourseWare.MIT.edu. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-867-machine-learning-fall-2006/ (accessed: Jan. 26, 2020).

[76] "Stanford engineering everywhere: CS229—Machine learning." SEE.Stanford.edu. https://see.stanford.edu/Course/CS229 (accessed: Jan. 26, 2020).

[77] "Practical deep learning for coders." Fast.ai.com. https://course.fast.ai/ (accessed: Jan. 26, 2020).

[78] PyTorch. https://www.pytorch.org (accessed: Jan. 22, 2020).

[79] TensorFlow. https://www.tensorflow.org/ (accessed: Jan. 21, 2020).

[80] "Deep Learning Toolbox." MathWorks.com. https://www.mathworks.com/products/deep-learning.html (accessed: Jan. 21, 2020).

[81] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[82] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proc. Medical Image Computing Computer-Assisted Intervention (MICCAI 2015)*, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[83] "benchmarks." Lambdalabs.com. https://lambdalabs.com/blog/tag/benchmarks/ (accessed: Jan. 27, 2020).

[84] "Lambda blade." Lambdalabs.com. https://lambdalabs.com/products/blade (accessed: Jan. 28, 2020).

[85] M. Malinowski, G. Swirszcz, J. Carreira, and V. Patraucean, "Sideways: Depth-parallel training of video models," 2020, arXiv:2001.06232

[86] M. J. Inman and A. Z. Elsherbeni, "Programming video cards for computational electromagnetics applications," *IEEE Antennas Propag. Mag.*, vol. 47, no. 6, pp. 71–78, Dec. 2005. doi: 10.1109/MAP.2005.1608730.

[87] T. Maceina, P. Bettini, G. Manduchi, and M. Passarotto, "Fast and efficient algorithms for computational electromagnetics on GPU architecture," *IEEE Trans. Nuclear Sci.*, vol. 64, no. 7, pp. 1983–1987, July 2017. doi: 10.1109/TNS.2017.2713888.

[88] "NVLink and NVSwitch." NVIDIA.com. https://www.nvidia.com/en-us/data-center/nvlink/ (accessed: Jan. 23, 2020).

[89] S. Peng and Z. Nie, "Acceleration of the method of moments calculations by using graphics processing units," *IEEE Trans. Antennas Propag.*, vol. 56, no. 7, pp. 2130–2133, July 2008. doi: 10.1109/TAP.2008.924768.

[90] H.-T. Meng, B.-L. Nie, S. Wong, C. Macon, and J.-M. Jin, "GPU accelerated finite-element computation for electromagnetic analysis," *IEEE Antennas Propag. Mag.*, vol. 56, no. 2, pp. 39–62, Apr. 2014. doi: 10.1109/MAP.2014.6837065.