Physics in Medicine & Biology

TOPICAL REVIEW

An introduction to deep learning in medical physics: advantages, potential, and challenges

To cite this article: Chenyang Shen *et al* 2020 *Phys. Med. Biol.* **65** 05TR01

IOP Publishing Phys. Med. Biol. 65 (2020) 05TR01 (29pp) https://doi.org/10.1088/1361-6560/ab6f51

Physics in Medicine & Biology

IPEM Institute of Physics and Engineering in Medicine

RECEIVED
18 June 2019

REVISED
17 December 2019

ACCEPTED FOR PUBLICATION
23 January 2020

PUBLISHED
3 March 2020

TOPICAL REVIEW

An introduction to deep learning in medical physics: advantages, potential, and challenges

Chenyang Shen[1,2], Dan Nguyen[1], Zhiguo Zhou[1], Steve B Jiang[1], Bin Dong[3] and Xun Jia[1,2]

[1] Medical Artificial Intelligence and Automation (MAIA) Laboratory, Department of Radiation Oncology, University of Texas Southwestern Medical Center, Dallas, TX, United States of America
[2] Innovative Technology Of Radiotherapy Computation and Hardware (iTORCH) Laboratory, Department of Radiation Oncology, University of Texas Southwestern Medical Center, Dallas, TX, United States of America
[3] Beijing International Center for Mathematical Research (BICMR), Peking University, Beijing, People's Republic of China

E-mail: xun.jia@utsouthwestern.edu, dongbin@math.pku.edu.cn and steve.jiang@utsouthwestern.edu

Keywords: deep learning, artificial intelligence, deep neural network

Abstract

As one of the most popular approaches in artificial intelligence, deep learning (DL) has attracted a lot of attention in the medical physics field over the past few years. The goals of this topical review article are twofold. First, we will provide an overview of the method to medical physics researchers interested in DL to help them start the endeavor. Second, we will give in-depth discussions on the DL technology to make researchers aware of its potential challenges and possible solutions. As such, we divide the article into two major parts. The first part introduces general concepts and principles of DL and summarizes major research resources, such as computational tools and databases. The second part discusses challenges faced by DL, present available methods to mitigate some of these challenges, as well as our recommendations.

1. Introduction

Artificial intelligence (AI) refers to the intelligence achieved by computer systems. Developing and employing AI techniques to solve important problems has become a central topic of many disciplines. Within the broad scope of AI, machine learning (ML) has been a popular topic for decades because of its capability of solving practical problems by learning from data. Over the past few years, deep learning (DL), a subcategory of ML, has achieved remarkable performance surpassing traditional ML approaches across a wide spectrum of different areas as a consequence of the availability of large-scale datasets, innovative DL technologies, advanced model training algorithms, as well as rapidly growing computing powers. In the classical image classification problem, a deep neural network (DNN) evaluated on the ImageNet 2012 classification dataset achieved an error rate of 3.57% (He *et al* 2016a), even lower than the human classification error rate of 5.1% (Russakovsky *et al* 2015). In the context of the Go game, AlphaGo (DeepMind Technologies Limited, London, UK) armed with DNNs successfully defeated the best professional human Go players (Silver *et al* 2016, 2017). These examples, together with many others, have led to the burst of DL research, exerting substantial impacts on our daily life.

Not surprisingly, powerful DL tools have also been introduced to solve problems in medicine. Numerous studies have demonstrated the power of DL in a variety of problems ranging from disease diagnosis (Esteva *et al* 2017, Rajpurkar *et al* 2017, Mendelson 2018), where an DL agent achieved a performance comparable or better than well trained clinicians, to data mining in health informatics (Ravì *et al* 2017), where the hidden structures in healthcare data were discovered and employed to support decision making. Specific to the medical physics community, the research interests on DL have also experienced a rapid and continuous growth in a relatively short period of time. To illustrate this fact, we searched papers that have 'artificial intelligence' or 'deep learning' in the title or abstract published in two major medical physics journals *Physics in Medicine and Biology* and *Medical Physics*, as well as in the medical physics category of *arXiv.org*, the world largest electronic archive of preprints in physics and mathematics. Figure 1 presents the result with the vertical axis displayed in a logarithmic scale. Apparently, there has been an exponential growth in the number of publications since 2015. This has also

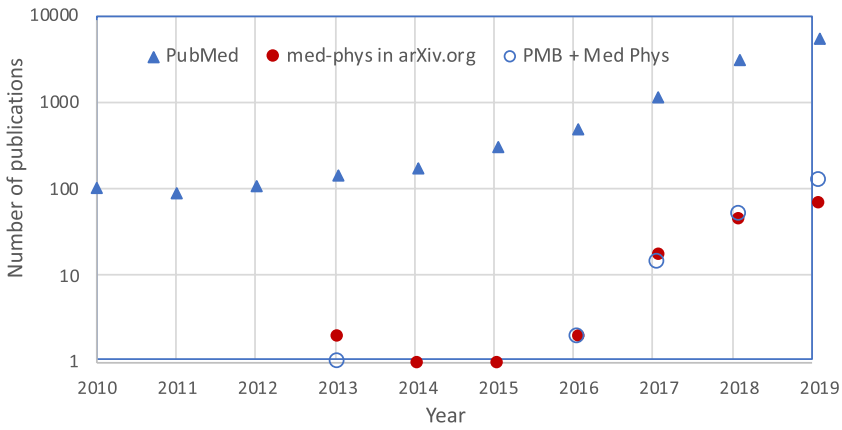

Figure 1. Number of DL-related research articles published in each year. The search for year 2019 only covered the first two months and the number was linearly annualized to the full year.

matched the overall trend of research interest on DL in medicine, as quantified by the number of publications obtained in the *PubMed* database with the same searching criteria.

Along with the rapid growth of research activities and achievements, it came to the point that we should look at the initial but substantial successes gained in a relatively short period of time. As such, multiple review articles on DL in healthcare, or specifically in medical physics, have been published (Miotto *et al* 2017, Thompson *et al* 2018b, Sahiner *et al* 2019). Meanwhile, DL has triggered extensive discussions regarding its role in medical physics research and clinical practice (Tang *et al* 2018, Thompson *et al* 2018a, Xing *et al* 2018, Sensakovic and Mahesh 2019).

These publications have successfully served their roles in terms of demonstrating the interests on DL, summarizing achievements and learnt lessons, as well as discussing future directions. However, we think there is also a strong need for a review article written from an educational aspect targeting researchers in the medical physics community. In particular, it is desirable to introduce to our community technical aspects of DL, available tools and resources, fundamental functions of DL from the mathematical perspective, and its capabilities and limitations. Having such a concise and focused review article could help researchers who are interested in DL, but may not have been trained in this area, to become familiar with this technology. This will facilitate their research and inspire novel research directions and activities. Meanwhile, we have to admit that DL has its own limitations. It has not yet been fully understood mathematically at this point why DL is so powerful in some problems. Discussing potential challenges of DL will help researchers to establish objective interpretations of this technology and to generate impactful DL studies in medical physics.

With these goals in mind, we prepared this topical review article. Instead of focusing on summarizing the remarkable achievements so far, this article will present a detailed introduction to the DL technology and in-depth discussions on it. The rest of this article is divided into two main parts. Sections 2–4 belong to the first part, which presents the DL technology as well as available research tools and resources. The purpose of this part is to give readers a brief, but hopefully comprehensive overview of DL. Since we focus on introducing the DL technology to researchers who are not trained in this area, a lot of concepts and terminologies will be provided at a very fundamental level. Experts in DL may find these sections basic and may skip them. Due to a limited space, it is not possible to cover all topics in a very detailed way. We will include important concepts for one to quickly gather necessary information, and provide references for further reading. The second part of this article, sections 5–7, provides more in-depth discussions on the fundamentals behind DL. We will introduce mathematical aspects of DL and discuss challenges of which we should be aware. We will also present potential solutions to some of the challenges, as well as our recommendations. Finally, we will conclude the review article in section 8 with an outlook into the near future.

2. What is artificial intelligence/machine learning/deep learning?

2.1. AI, machine learning, and deep learning

Artificial intelligence (AI) refers to the intelligence achieved by machines, which is in contrast to the natural intelligence of humans. In general, it broadly encompasses the capabilities of any devices or systems to take actions for successfully achieving specific goals (Jackson 1985, Nilsson and Nilsson 1998). In this sense, any models, algorithms, or computer programs designed by humans to tackle certain tasks requiring human intelligence can be generally considered as AI. Hence, the scope of the term AI covers a wide spectrum of problems including, but not limited to perception, recognition, analysis, and decision-making using a machine or computer.

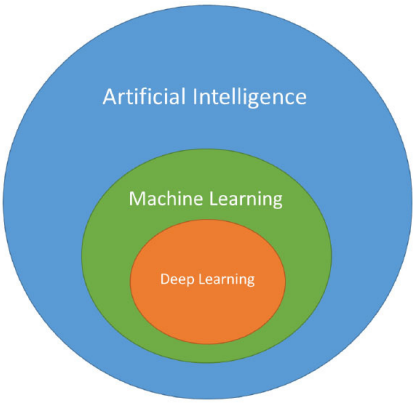

Figure 2. Relationship among artificial intelligence, machine learning, and deep learning.

Machine learning (ML) is a subcategory of methods within the broad scope of AI (figure 2), which specifically refers to numerical algorithms and models established to analyze data and derive or learn decision-making capabilities to achieve certain tasks (Alpaydin 2009). In other words, ML deals with data. Its goal is to conclude the hidden pattern embedded in the data under practical constraints, such as data size and quality. The derived pattern can then be used to solve the problem of interest. Take the classical problem of image classification as an example, an ML method would try to draw a boundary to separate different classes by analyzing the dataset of images. For a new image to be classified, it is compared to the learnt boundary to decide the class that it belongs to.

First introduced by Aizenberg *et al* (2000), *Deep learning* (DL) is a group of methods within ML (figure 2). Therefore, the general goal of DL is aligned with that of ML. What differentiates DL from other ML techniques is that DL employs large-scale hierarchical models with multi-layer architectures to automatically generate comprehensive representations and to learn complicated inherent patterns of the data (LeCun *et al* 2015). In contast, classical ML methods uses hand-crafted features manually extracted from data as input and relies on relatively simple models to represent inherent data patterns. In recent years, DL is becoming increasingly popular in both research and applications because of its feasibility granted by advanced numerical algorithms, high computing power, and available large-scale datasets, as well as its impressive performance as compared to traditional ML methods.

2.2. Deep neural network

The most commonly employed models in DL are *deep neural networks* (DNNs), which are essentially a type of artificial neural network (ANN) (McCulloch and Pitts 1943) but with a large number of layers. The building block of a DNN is a *neuron* designed in analogy to the neural cell of a human. Each artificial neuron consists of four major components: a group of input signals, a linear operation, a non-linear operation, often termed as activation function, and its output signal (figure 3(a)). The neuron takes its inputs and first performs a linear operation on them. The resulting data are fed into an activation function, commonly a non-linear function, to generate the neuron's output. A typical form of the activation function is rectified linear unit (ReLU) (Nair and Hinton 2010) outputting zero if the input value is negative, and the same value as the input otherwise, although other function forms are often used. A large number of neurons are connected under a certain structure to form a DNN, where the output of a neuron is fed to another neuron as one of the inputs.

Typically, the DNN contains a number of layers (e.g. figure 3(b)). Neurons in one layer receive information from the previous layer, and after processing it, pass the result to the next layer. Any layer embedded between the input of the DNN and the output is termed as a *hidden layer*. In mathematical terms, let us denote the input data as a vector $x = [x^1, x^2, x^3, \ldots]$, the data at the ith hidden layer as $h_i = [h_i^1, h_i^2, h_i^3, \ldots]$, and the output data as $y = [y^1, y^2, y^3, \ldots]$. We further denote operations acting on the input of the $i$thlayer as $f_i(\cdot|\theta_i)$, where θ_i represents all relevant parameters defining the operations. Note this operation contains both the linear and non-linear part as mentioned previously. With a number of n layers, the basic neural network operations can be written as:

$$\begin{aligned} h_1 &= f_0(x|\theta_0), \\ h_i &= f_{i-1}(h_{i-1}|\theta_i), \\ y &= f_n(h_n|\theta_n). \end{aligned} \tag{1}$$

It is easy to see that this can be written in the form of a composite function:

$$y = DNN(x|\theta) = f_n(f_{n-1}(f_{n-2}(\cdots f_1(f_0(x|\theta_0)|\theta_1))\cdots|\theta_{n-1})|\theta_n), \tag{2}$$

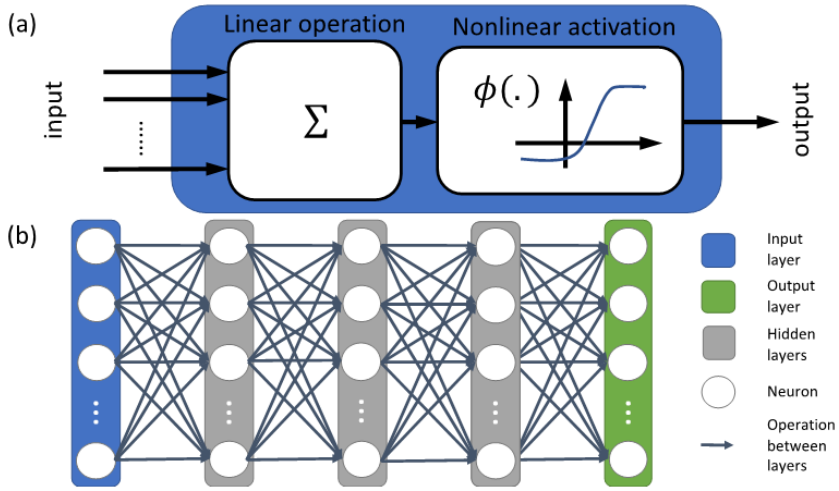

Figure 3. (a) Structure of a neuron. Σ indicates a linear operation acting on the input variables, and $\phi(.)$ is the nonlinear activation function. (b) An example of a fully connected DNN model with a multi-layer structure.

where the function $DNN\ (x|\theta)$ is a mapping from the input to the output, and θ represents the set of θ_i to define this DNN mapping. The total number of layers indicates the *depth* of a DNN, while the number of neurons in a layer gives its *width*.

It is worth mentioning that a DNN does not necessarily organize its neurons in a layered format and more complicated architectures may be employed. For example, skip or residual connections can be added to connect neurons in non-adjacent layers, as in the popular ResNet (He *et al* 2016a). Mathematically, a hidden layer calculation with a skip connection between the layers i and j can be written as

$$h_i = f_{i-1}\left(f_{i-2}\left(\ \cdots\ h_j\right)\right) + h_j, \text{ for certain } i > j. \tag{3}$$

Regardless of the exact formulation, the commonality among DL models is that they utilize many consecutive layers of calculation. Popular operation types of these layers include, but not limited to full connection (Ivakhnenko and Lapa 1965), convolution (Fukushima 1980, LeCun *et al* 1990, LeCun and Bengio 1995), max-pooling/up-sampling (Scherer *et al* 2010, Ciresan *et al* 2011), batch normalization (Ioffe and Szegedy 2015), ReLU (Nair and Hinton 2010), sigmoid-shaped functions, the soft-max function (Goodfellow *et al* 2016) etc. A typical scheme of DL is to first apply a linear operation on the previous layer followed by a non-linear operation. It is very uncommon to use two linear operations sequentially, since this is equivalent to using a single linear operation. In addition, a normalization scheme, including batch (Ioffe and Szegedy 2015), layer (Lei Ba *et al* 2016), instance (Ulyanov *et al* 2016), or group (Wu and He 2018) normalization, may be applied either directly before or after the non-linear operation, which tends to improve the convergence speed during the network training process.

To date, a large number of DNN architectures have been designed for different applications. For example, fully connected DNNs are the most general form that plays an important role in numerous studies. Convolutional neural networks (CNNs) employ convolutional operations to effectively extract image features and are widely used in image-related projects. Recurrent neural networks (RNNs) have a recurrent mechanism that can handle data with a temporal structure. Most of successful applications so far incorporate one, or the combination of these DNNs. Major characteristics of these DNNs are summarized in Table 1 and we will discuss them in detail in the following subsections.

2.2.1. *Fully connected deep neural network*

In principle, a *fully connected DNN*, in which every pair of neurons are linked using pair-specific connections between two adjacent layers, is the most general form of DNNs. Figure 3(b) shows the basic design of a fully connected DNN. The number of neurons in the input, output and hidden layers may vary, depending on many attributes such as the data format or the intended use of the model.

While the fully connected DNN is a very general network form, it is not used very often in practice by itself. For a specific problem of interest, it is of importance to design a special network form, which can be viewed as by removing some connections of a fully connected DNN purposely, to make the network easier to train and to improve computational performance. The design of the specific network structure can be very creative, and often be specifically tailored based on the researcher's domain knowledge and experience. For instance, various CNN (Krizhevsky *et al* 2012, Cho *et al* 2014, Simonyan and Zisserman 2014, Ronneberger *et al* 2015) and RNN (Hochreiter and Schmidhuber 1997, Graves *et al* 2014) architectures are designed specifically for the processing of image data and sequential data, respectively.

Table 1. Commonly used DNNs.

Networks	Characteristics
Fully connected DNN	1. All neurons in one layer are connected with all neurons in the adjacent layer using pair-specific connections 2. There is a large number of trainable network parameters 3. Typically, a large amount of data and computational resources are required to determine parameters
CNN	1. Convolution and pooling operations are involved to connect adjacent layers 2. Typically, there are considerably less trainable network parameters compared to a fully connected DNN 3. Often used to handle image-related tasks
RNN	1. Feedback mechanisms are incorporated in hidden layers to realize a recurrent structure 2. Often used to handle time series data

2.2.2. Convolutional neural network

Convolutional neural networks (CNNs) proposed by LeCun *et al* (1989) effectively use convolutional layers for image-related tasks. The convolutional computation utilizes a kernel and convolves with the previous layer's image data to produces new images, called feature images, and feeds them to the next layer. Pooling operations, such as max-pooling or average-pooling, can be added after convolution to reduce resolution of the feature images. This allows for the model to reduce computational cost and to view and analyze the images at multiple resolution scales. In addition to these layers, CNNs can still incorporate other operations such as fully connected layers, batch/layer/instance/group normalizations, non-linear activations, etc.

Recently, a number of researchers have revealed the tremendous performance of CNNs on image related tasks. This is largely ascribed to the capability of analyzing images at different resolution scales that comes with convolutional and pooling operations. For instance, a basic structure of a classification CNN inspired by LeNet (El-Sawy *et al* 2016) is shown in figure 4(a). After analyzing the images at different scales through the first few convolutional layers, the extracted information is gathered and further processed in the last a few fully connected layers to generate the final output. Several variants of CNN model structures for classification have been proposed, such as AlexNet (Krizhevsky *et al* 2012), VGGNet (Simonyan and Zisserman 2014), and GoogleNet (Szegedy *et al* 2015).

CNNs have also been employed widely for image-to-image translation tasks. Different from those for image classification, a CNN in an image translation task outputs an image that has a one-to-one pixel or voxel correspondence with the input. One of the common uses of such CNNs is for image segmentation, where the output is the segmented region maps. A popular architecture designed for this task is the U-net (Ronneberger *et al* 2015), and its 3D variant, the V-net (Milletari *et al* 2016), which for the first time introduced the Dice loss layer widely incorporated nowadays. A basic structure of U-net style architecture is depicted in figure 4(b). Because of the multiple layers of convolutions and pooling operations to change image resolutions among layers, U-net's and V-net's architectures allow effective calculation and combination of both local and global features. Many variations of these two networks have appeared in literature, typically with modifications designed for specific tasks of the studies.

2.2.3. Recurrent neural network

Recurrent neural networks (RNN) are neural networks with a feedback mechanism in the hidden layers, as shown in figure 5. Because of the recurrent nature, an RNN can be equivalently viewed as a series of stacked networks with identical structures. RNN was designed to effectively learn from sequential data, such as writing, speech, time series data, decision pathways, etc. Structure-wise, RNNs can be created using fully connected or convolutional style layers, as well as other aforementioned DNN operations. The original RNN structure was found to be limited to only short sequences of data mainly due to the so-called unstable gradient issues in propagating memory from previous iterations. To mitigate this issue, a long short-term memory (LSTM) model (Hochreiter and Schmidhuber 1997) was proposed, which added extra mechanisms for remembering/forgetting past information (forget gate), adding new data into the memory (input gate), and calculating desired output for that iteration (output gate). With these modifications, LSTM is capable of learning and performing on much longer sequences of data, and has largely replaced the basic RNN for most modern tasks. While the LSTM is one of the most popular version of RNN, other recurrent networks have been devised as well, including bi-directional RNN (Schuster and Paliwal 1997, Mikolov *et al* 2010, Sak *et al* 2014, Zaremba *et al* 2014), gated recurrent unit (GRU) (Cho *et al* 2014), neural turing machines (Graves *et al* 2014), etc.

2.2.4. Other network structures

In the modern DL setting, most frameworks utilize the aforementioned fully connected DNNs, CNNs, RNNs, or a combination of them to achieve a high performance for a given task. However, there exists less conventional

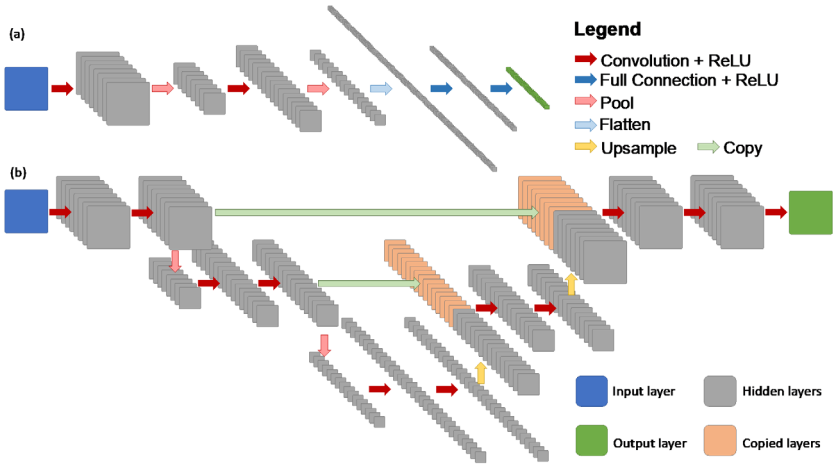

Figure 4. Basic structure of (a) CNN for classification tasks and (b) a U-net style architecture. Up-sample is an operation to increase the dimension of data via a user-defined interpolation method. Flatten indicates the operation to reshape a high-dimensional data array into a 1D vector.

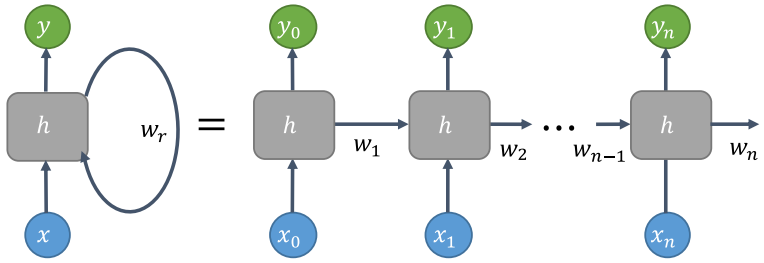

Figure 5. Illustration of the RNN architecture (left). x is the input data, h indicates the RNN, y is the output of RNN, and w_r is the recurrent information. This network is equivalent to an unfolded chain-like architecture (right) with each block being a copy of the same network. Each block passes information to its successor.

network structures, such as deep belief network (DBN) (Lee *et al* 2009) and deep Boltzmann machine (DBM) (Mohamed *et al* 2009). In the interest of space, we will not present their details and readers can find more from relevant references.

2.3. Training of a machine learning/deep learning model

Using an ML/DL approach to solve a problem requires a training stage to develop the model. Specifically, *model training* refers to the process of determining the model parameters based on observed data (training data). Training the ML/DL model is often formulated mathematically as solving an optimization problem, where the goal is to find the model parameters that minimize a loss function. For a DNN model, this can be expressed as

$$\theta^* = \min_{\theta} L(\theta) = L\left(DNN\left(x|\theta\right)\right), \tag{4}$$

where θ^* is the set of parameters of the trained model. The loss function $L\left(\cdot\right)$ is problem specific. Take the image classification problem as an example, if we were given a set of images x_i and corresponding classification labels y_i, the loss function can be naturally defined as $L\left(\theta\right) = \sum_i ||y_i - DNN\left(x_i|\theta\right)||^2$. Minimizing this loss function explicitly enforces the agreement between the predicted label $DNN\left(x_i|\theta\right)$ and the ground truth label y_i by minimizing the difference between them. In practice, the loss functions can be defined creatively in different forms, e.g. cross entropy for classification, Dice similarity coefficient for segmentation (Dice 1945), depending on specific considerations.

It is important to note that equation (4) is often a non-convex optimization problem. This means that it is difficult to find the global minimum. This optimization problem is typically solved via a gradient-based algorithm that updates the solution θ as

$$\theta^{k+1} = \theta^k - \lambda \nabla_\theta L(\theta), \tag{5}$$

where k is the index of iterations, $\nabla_\theta L$ is the gradient term with respect to θ, and λ is called the learning rate. For the loss function involving a complex DNN, it is complicated to evaluate the gradient term. Rumelhart *et al* (1986) proposed to use a technique called backpropagation, which effectively employs the chain rule in calculus

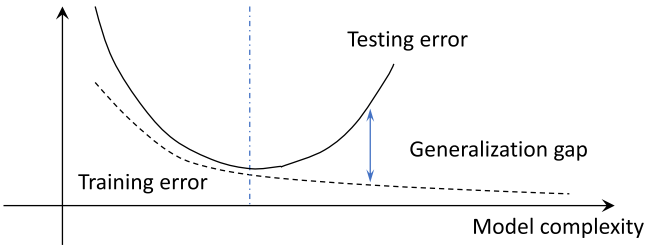

Figure 6. Illustration of training and testing errors as a function of model complexity.

to compute gradient and update the parameters θ at each layer. Although this technique is very computationally efficient, it still does not guarantee a global minimum. To help both the local minimum issue as well as to alleviate the problem of potentially a large memory usage, DNNs are often trained using stochastic optimization methods, such as stochastic gradient descent (SGD) or adaptive moment estimation (ADAM) (Kingma and Ba 2014). While these methods differ slightly, such as additional momentum calculations or adaptive learning rates, a common scheme is that only a portion of the training data is randomly selected at a given step to update the solution. More specifically, the complete training scheme consists of a number of *epochs*. Before each epoch, the training data is shuffled and split into a number of small portions called *batches*. The training process loops over all these batches in each epoch, each time using data from a batch to update the model. A number of training epochs are needed to yield convergence or satisfactory results. Note that the data used to update the model in different epochs are the same, but the data batches are randomly generated.

Similar to solving other optimization problems, training the DL model involves many user-defined parameters in the algorithm, called hyper-parameters. Examples include learning rate, number of epochs, batch sizes, and dropout rate (Srivastava *et al* 2014), etc. The values of these hyper-parameters affect the solution and hence resulting model performance. One usually repeatedly adjusts these hyper-parameters to achieve a satisfactory performance of the trained model.

A typical setup when constructing a ML/DL model is to use a portion of available data, called *training dataset*, for model training, i.e. to solve the optimization problem in equation (4) and set aside a smaller, hold-out *testing dataset*, to evaluate the model performance, after the training stage is completed. The loss function $L(\cdot)$ used to train a ML/DL model can often be viewed as a certain form of error, when evaluating the model performance on the datasets. The error evaluated on the testing dataset is typically larger than that on the training set. The difference between the two is termed as the *generalization gap*, see figure 6 for a graphical illustration. This gap tends to indicate the model's *generalizability*. Being able to generalize means that the model trained on the training dataset performs well on the testing dataset that has not been seen by the training process. Hence, we expect that the data will likely perform equally well for future data, when we apply the trained model to solve the problem. In this case, the generalization gap is small. On the other hand, a large gap means bad model generalizability. Generally speaking, using a model with a high complexity can reduce the training error, as shown in figure 6. However, this does not mean the trained model becomes more accurate. After a certain point, the model starts to be capable of 'memorizing' the training data, as characterized by the increasing testing error and generalization gap. Since the goal of a ML/DL study is to build a model using training data and to apply it on unseen future data, we would like to find the sweet spot where the generalization gap is minimized, so that the trained model has the best generalizability.

In the regime of DL, studies have observed a quite surprising phenomenon: a very complex DL model with an extremely large number of trainable parameters may also generalize well after being trained on a large dataset (Neyshabur *et al* 2018, Novak *et al* 2018). The exact mathematical reason for this phenomenon is still unclear, although there are some hints. Numerous studies have been actively investigating this direction (Belkin *et al* 2018). More detailed discussion on this will be given in section 6.2.

2.4. Feasibility and popularity of deep learning

Since Ivakhnenko and Lapa (1965) established the first effective DNN, DL has been around for more than 50 years. Yet the lack of large-scale datasets and the limited computing power impeded its applications. These obstacles were recently overcome, making DL a feasible solution to many problems. Formulated in a hierarchical multi-layer architecture, a DNN consists of a number of hidden layers. The large number of network parameters naturally requires the collection of a massive training dataset. Creating such a large-scale dataset was expensive, labor intensive, and time consuming until very recently, when fast development in data acquisition/sharing techniques and storage capability have been advanced. One of the most impressive examples is the ImageNet database (Deng *et al* 2009, Krizhevsky *et al* 2012, Russakovsky *et al* 2015) where more than 14 million fully annotated images were collected from the Internet.

In addition, training a DNN is usually formulated as solving a large-scale optimization problem, demanding a huge computing power. Given the relatively limited computing power in the past decades, it was not practical to use DNN. This issue has been eased by the substantial increase in memory and computing power of modern computers/workstations. In particular, graphics processing unit (GPU) has been employed in the scientific computing regime (Pratx and Xing 2011, Jia *et al* 2014a, Despres and Jia 2017). Its remarkable power to parallelize computations substantially improves the efficiency of training DL models, making it affordable to use DNN in real-world applications.

One of the main reasons making DL popular is its flexibility in handling different problems. One critical step for the success of a classical ML model is to extract concise representations of the data, namely feature extraction. Due to the relatively limited flexibility of traditional ML models, a concise and complete representation of the data is very important to feed the ML model with relevant information for accurate learning. Take a typical image classification problem to classify images into cats and dogs as an example. A classical ML approach typically requires manually extracting relevant features from input images, e.g. shape, size, color, etc, and feeding them into the ML model to predict the output class label. However, this step is very challenging, and often needs problem-specific and creative thinking. In contrast, DL can be conducted through a purely data-driven end-to-end approach. The flexibility of a DNN is large, so that it is capable of establishing a mapping directly from the input to the output, bypassing the feature extraction step. For the cat/dog classification problem, a DL model can be constructed to learn a mapping directly from the image to the class label using a large amount of labeled training data. The researcher does not have to explicitly specify what features the model should look at. After training, the feature extraction step is inherently incorporated in the built model.

Additionally, the remarkable performances beyond most of traditional ML methods in a spectrum of problems also contribute to DL's popularity. Examples include natural language processing (Mikolov *et al* 2010, Manning *et al* 2014), face recognition (Schroff *et al* 2015), image classification (Krizhevsky *et al* 2012, Russakovsky *et al* 2015), playing board games (Silver *et al* 2016, 2017), and self-driving (Bojarski *et al* 2016), to name a few.

3. Learning strategies of deep learning

Depending on how a ML model is trained, we generally divide the learning strategies into supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. The same classification can be applied to learning strategies of DL, as it is a subcategory of ML. In this section, we will briefly introduce the characteristics of each category and present some examples.

3.1. Supervised learning

Supervised learning (SL) refers to the learning strategy that trains a model by using labeled data with input–output correspondence and by explicitly enforcing the compliance of the model to this correspondence. SL is the most straightforward and effective learning strategy, since the learning goal is clearly defined by the paired input data and output target. With the widely available DL platforms (see details in section 4), it is quite easy to set up a DL model as well as to perform training via SL. On the other hand, SL has its limitation of requiring the co-existence of input data and corresponding output target. For instance, each training image has to be annotated with an explicit class label for the image classification task. Such a strict requirement on dataset diminishes the practical value of SL in many contexts, especially for those applications where the targets of data are hard to obtain.

As many medical problems can be formed as mappings from the input to the output side, SL is probably the most obvious approach to solve these problems in a DL way. The researchers could be very creative in terms of defining the inputs and outputs. Rather than enumerating the large number of successful DL models trained with SL, we will briefly show a few representative examples.

Given the success of DL in image-related areas, applications in medical imaging are certainly warranted. For instance, remarkable success has been achieved using DL-based methods trained with SL to perform segmentation (mapping from anatomy image to organ maps) (Ronneberger *et al* 2015, Roth *et al* 2015, Cha *et al* 2016, Guo *et al* 2016, Hu *et al* 2016, Milletari *et al* 2016, Hu *et al* 2017, Ibragimov and Xing 2017, Balagopal *et al* 2018, Ren *et al* 2018, Saffari *et al* 2018, Chen *et al* 2019a, Jung *et al* 2019a, 2019b), to improve image quality (mapping from a low-quality image to the corresponding high-quality image) (Han *et al* 2016, Chen *et al* 2017a, Gjesteby *et al* 2017a, 2017b, Kang *et al* 2017, Kelly *et al* 2017, Hansen *et al* 2018, Liang *et al* 2018, Maier *et al* 2018a, 2018b, Rivenson *et al* 2018, Schlemper *et al* 2018, Xie *et al* 2018, Zhang and Yu 2018a, Zhu *et al* 2018), to improve image resolution (mapping from a low-resolution image to the corresponding high-resolution image) (Oktay *et al* 2016, Pham *et al* 2017, Chen *et al* 2018, Iqbal *et al* 2019), to convert images between different modalities (mapping from one image modality to another) (Han 2017, Fu *et al* 2019, Kazemifar *et al* 2019, Liu *et al* 2019a, 2019b), and to conduct disease diagnosis based on medical images (mapping from an image to a label of diagnosis result) (Zhang *et al* 2014, Hua *et al* 2015, Kumar *et al* 2015, Shen *et al* 2015, Cheng *et al* 2016, Litjens *et al* 2016, Sun *et al* 2016, Rajpurkar *et al* 2017, Wang *et al* 2017, Chen *et al* 2019b) etc.

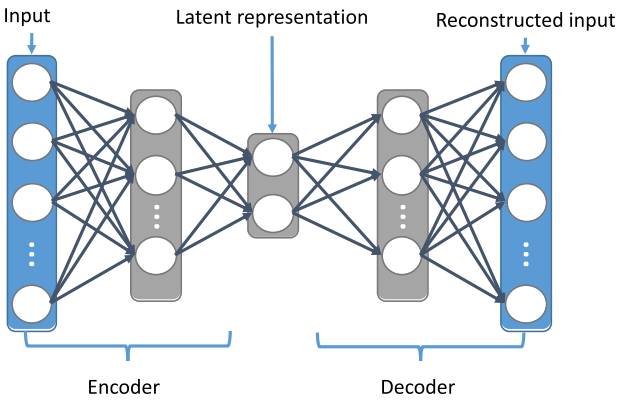

Figure 7. An example of a deep auto-encoder architecture. The left and right halves of the network are encoder and decoder, respectively. The vector in the middle is the representation of the data in the latent space.

Applications of SL can go beyond image-related problems. Zhen *et al* developed a DL-based model trained with SL to build a mapping from radiation dose distribution to rectum toxicity after radiotherapy (Zhen *et al* 2017). Similar strategy has also been applied for distant metastasis prediction after radiotherapy in head and neck cancer (Diamant *et al* 2019). In the radiotherapy treatment planning regime, SL were employed to train DL models to establish a relationship between a patient image to the best achievable dose distribution (Nguyen *et al* 2019a, 2019b, 2019c). This approach has also been used for treatment plan quality assurance purpose (Tomori *et al* 2018, Nyflot *et al* 2019).

3.2. Unsupervised learning

In contrast to SL, *unsupervised learning* (USL) is a group of methods that purely rely on input data and seek for the inherent data patterns without requiring any target information. In this sense, the learning objective is not explicitly described and the learning process is fully driven by the data itself. A representative application of USL is data clustering, where the data samples need to be grouped into several clusters without any prior knowledge regarding group labels of sample, sometimes not even the number of clusters in the dataset.

In DL, one typical example of USL is the training of a deep auto-encoder (Vincent *et al* 2010, Ngiam *et al* 2011). Auto-encoder (AE) refers to the model building two-way mappings between the original data space and a latent space, i.e. a vector space of a relatively lower dimension than the original data space while keeping most of the data information in the original space. While AE can be achieved using many ML approaches, a *deep auto-encoder* (DAE) refers to an AE model constructed using a DNN architecture, as shown in figure 7. The function that maps from the original data space to the latent space is called an *encoder*, whilst a *decoder* stands for the mapping that reconstructs the data from the latent representation. The detailed network structure of a DAE is quite flexible. For example, the network can be CNN based, fully connected DNN based, or a mixture of them, depending on the specific context. Essentially, a DAE simply aims at creating a concise representation of the original data on a latent space with most of the information from the original space preserved, such that the data can be exactly recovered based on the latent representations. To achieve this goal, training a DAE is performed by simply enforcing the agreement between the original data and the corresponding output from the decoder. No task-specific information or target is involved in the training process, as required by the definition of USL. Several variants of DAE have been proposed such as sparse autoencoder (Xu *et al* 2016), denoising autoencoder (Vincent *et al* 2008) and variational autoencoder (Sønderby *et al* 2016).

DAE has its broad applications in many different areas. Within the scope of medical physics, DAE has been successfully applied to seek for a low-dimension representation of patient CT images, such that high-quality low-dose CT images can be reconstructed by restricting the solution to the trained DAE-based manifold as prior information (Wu *et al* 2017, Ma *et al* 2018). It has also been employed to enhance the robustness and quality for real-time MRI and CT image reconstruction (Mehta and Majumdar 2017). A DAE-based unsupervised deep feature learning algorithm was developed for medical image analysis (Guo *et al* 2016, Chen *et al* 2017b). DAE has also been utilized for other medical physics related tasks, such as breast density segmentation (Kallenberg *et al* 2016), multiple organ detection (Shin *et al* 2013), and breast cancer nuclei detection (Xu *et al* 2016).

In addition to DAE, other types of USL have also been employed in medical physics field. For instance, deep Boltzmann machine was applied to tackle the problem of heart motion tracking for treatment planning (Wu *et al* 2018).

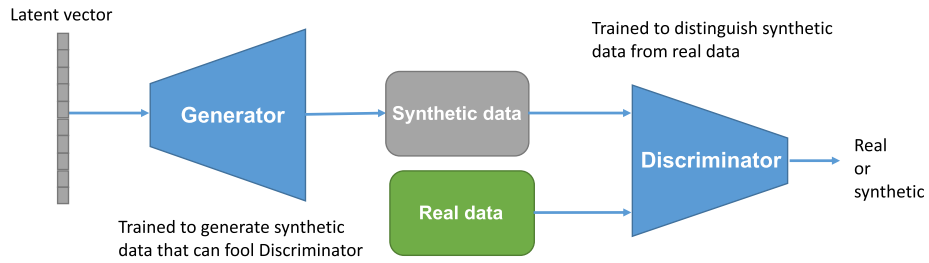

Figure 8. Structure of GAN. It consists of a generator and a discriminator. The generator is a network producing synthetic data by taking latent vectors as input. The discriminator is another network trained to distinguish real and synthetic data.

3.3. Semi-supervised learning

Semi-supervised learning (semi-SL) falls in between SL and USL. It broadly refers to the training strategies designed for applications with target information only partially available, which holds true for many real-world applications. Hence, semi-SL methods have attracted great research interests. The goal of semi-SL is to fully utilize data with/without targets rather than simply treating the problem via SL using only the data with labels, or formulating the problem via USL while completely ignoring the available labels.

One of the most widely used semi-SL techniques in the DL area is generative adversarial networks (GAN) (Goodfellow *et al* 2014). A general GAN structure is illustrated in figure 8. A generative network (generator) and a discriminative network (discriminator) are trained simultaneously to fight against each other. The goal of a discriminator is to distinguish real and synthetic samples in a way similar to human perception, whilst the generator is trained to produce examples that are realistic enough to fool the discriminator. To handle the unlabeled data in semi-supervised learning, semi-supervised GAN has been proposed (Springenberg 2015, Odena 2016, Kumar *et al* 2017). For each unlabeled data, instead of specifying which label it should receive, the semi-supervised GAN forces it to belong to one of the possible categories with a large probability based on the underlying pattern in data as well as the labeled data available in dataset.

Variants of GAN architecture have been incorporated into the medical physics field. A recent study (Liang *et al* 2018) has successfully adopted the CycleGAN (Zhu *et al* 2017) to generate synthesized CT images from CBCT images for adaptive radiation therapy without fully relying on paired CT-CBCT data. Zhang *et al* proposed a deep adversarial network for biomedical image segmentation by utilizing unannotated images (Zhang *et al* 2017). Nie *et al* established an attention based approach using a confidence network for adversarial learning to tackle the image segmentation problem (Nie *et al* 2018). Madani *et al* put forward a semi-supervised GAN model to solve domain adaptation problem for chest x-ray classification (Madani *et al* 2018).

There are also many other types of semi-SL approaches established in DL. In medical physics field, Feng *et al* proposed a progressive semi-SL strategy for MRI segmentation that gradually enlarges the training dataset along training steps by including reasonable unlabeled data (Feng *et al* 2018). Bai *et al* designed an iterative strategy to alternatively train the DNN model and estimate labels of unlabeled data for cardiac MRI segmentation (Bai *et al* 2017). A deep multi-planar co-training strategy was developed for multi-organ segmentation (Zhou *et al* 2018), where pseudo-labels were generated for unlabeled data. The scheme was also employed in Sun *et al* (2017), in which a graph based semi-SL method was put forward for breast cancer diagnosis with majority of unlabeled data.

3.4. Reinforcement learning

Reinforcement learning (RL) (Sutton and Barto 2018) is a ML/DL strategy that enables a model, or more frequently referred as an *agent*, to learn by interacting with an environment, an existing system that produces states based on the agent's actions. This is illustrated in figure 9. Essentially, RL tries to train the agent to make decisions to maximize a reward based on the interactions between the agent and environment. In *deep reinforcement learning* (DRL) (François-Lavet *et al* 2018), a deep neural network is incorporated to model the agent. This approach incorporates the superior perception ability of DL into the RL framework to improve the decision-making performance for complex tasks. Different from previously mentioned learning strategies, during the process of model training, DRL utilizes a reward function obtained from the environment to improve the model. Specifically, it trains DL models in a natural trial-and-error learning strategy similar to that of a human. The deep model, namely agent, is constructed to learn decision-making by observing the reaction of the environment, i.e. how the environment changes its state in response to the decision. The quality of the decision, good or bad, is quantified by the reward function. Through a series of interactions with the environment, the agent can learn how to make appropriate decisions based on the observed state to maximize its reward.

Modern applications, such as in (Mnih *et al* 2013, 2015), often combine DRL with Q-learning (Watkins and Dayan 1992). Instead of aiming at maximizing the reward obtained through the next interaction, the goal of

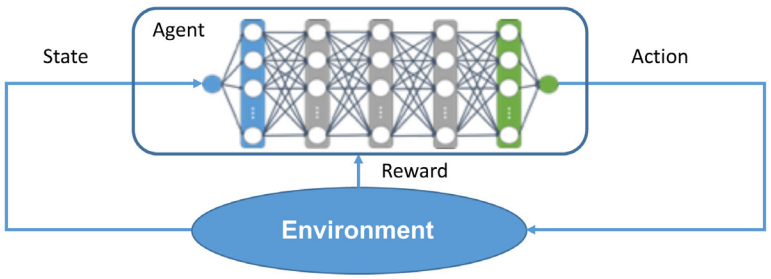

Figure 9. Illustration of the DRL process. An environment indicates an existing system that produces states after interacting with an agent. The gain in the interaction is evaluated by a pre-defined reward function. An agent in the form a deep neural network observes the state from the environment and determines an action. The overall goal of DRL is to train a deep neural network-based agent to maximize the reward automatically by interacting with the environment.

Q-learning is to predict the total reward to be obtained in the sequence of future interactions. Deep Q-learning has been successfully applied to achieving many real-world problems, such as playing Atari games (Mnih *et al* 2013, 2015) and the game of Go (Silver *et al* 2016, 2017).

DRL essentially tries to mimic human's decision-making behaviors, which holds a strong potential for solving medical physics problems requiring human inputs. It has been successfully applied to automatically adjust regularization parameters in iterative CT reconstruction (Shen *et al* 2018, 2019d). In addition, Shen *et al* established an intelligence virtual treatment planner that is able to automatically operate a treatment planning engine to generate clinically acceptable plans in a human-like fashion. This idea has been tested in high-dose-rate brachytherapy (Shen *et al* 2019a), as well as in external beam radiation therapy (Shen *et al* 2019b, 2019c). DRL was also employed to automate the decision process of adaptive radiotherapy for non-small cell lung cancer (Tseng *et al* 2017). A multimodal image registration strategy was also developed based on the deep context reinforcement learning (Ma *et al* 2017).

4. Available research tools and datasets

4.1. Deep learning frameworks

Due to the exploding popularity of DL in recent years, a number of computational packages and frameworks have been established to simplify development and deployment of DL models, with large levels of support backed by companies. All of these packages now handle basic and generic operations encountered in DL to allow for a relatively straightforward implementation. For instance, commonly used layers in a DNNs, such as fully connected layer, 1D/2D/3D convolution layer, batch normalization, dropout layer, max-pooling layer, etc, are predefined in these DL frameworks. They also provide readily usable activation functions including ReLU, sigmoid, soft-max, etc. Users can simply call some functions to set up a DNN suitable for their own tasks. These DL frameworks have built-in optimization algorithms that can calculate gradient/momentum and perform backpropagation operations to train a network. These features allow researchers to begin at a higher level of development, and focus more on solving their own problems, rather than spending a lot of efforts on implementing the DL model and training it. The following paragraphs will present common DL packages available to researchers. We also summarize them in table 2.

Currently having the largest community usage and support, TensorFlow developed by the Google Brain Team in 2015 is the most popular DL framework (Abadi *et al* 2016). TensorFlow was designed with production and scalability in mind, making it very popular in the industrial setting where quickly pushing prototypes to deployment is essential. Moving forward, TensorFlow 2.0 now has Eager Execution enabled by default, which allows for faster debugging, immediate run time, dynamic computational graphs, and custom gradients, ultimately leading to faster prototyping and development from a research standpoint. TensorFlow 2.0 fully integrates Keras (Chollet 2015) as the default mechanism, a high-level application programming interface (API) that wraps around the core frameworks like TensorFlow, Microsoft Cognitive Toolkit, and Theano. It contains a simple, easy to utilize interface to access the packages' core operations. The integration of Keras into TensorFlow as the default creates a simple and seamless method for model development, training, validation, testing, and deployment.

While TensorFlow gets the award for the largest community and support, PyTorch (Paszke *et al* 2017) gets the award for the fastest growth. Released in 2016, PyTorch is the Python-based version and successor of Torch, a popular ML and scientific computing framework written based on Lua and released in 2002. In contrast to TensorFlow that started with a scalable and production ready type of framework and added Eager mode at a later time point, PyTorch followed a reversed path. They designed the framework to be more 'Pythonic' (follows the conventions and language use of Python), allowing for dynamic computation graphs from the beginning. These

Table 2. Summary of commonly used DL packages.

DL packages	Language	Characteristics
TensorFlow (Abadi *et al* 2016)	Python	• Largest community usage and support • Easy implementation • Integrated Keras API to simplify implementation • Large number of examples, e.g. available at https://github.com/hwalsuklee/tensorflow-generative-model-collections
PyTorch(Paszke *et al* 2017)	Python	• Fastest growing community usage and support • Easy implementation • Large number of examples, e.g. available at https://github.com/znxlwm/pytorch-generative-model-collections
Caffe (Jia *et al* 2014b)	C++	• Large community usage and support (less popular than TensorFlow or PyTorch) • Merged with PyTorch
Theano (Bergstra *et al* 2010, 2011, Bastien *et al* 2012)	Python	• Limited community usage and support • Ceased major development
MXNet (Chen *et al* 2015)	Multi-language	• Limited community usage and support • High efficiency
CNTK (Seide and Agarwal 2016)	Multi-language	• For commercial-level distributed DL
DL4J (Deeplearning4j 2016)	Java	• Limited community usage • Suitable for applications in Java
Chainer (Tokui *et al* 2015)	Python	• Limited community usage • Early adopter of 'define-by-run' DL
MATLAB	Multi-language	• Large community usage and support • Non-open-source • Slower update compared to other open-source packages • Available interface with other packages

aspects made PyTorch great for rapid prototyping and gain immense popularity among the academic community. More recently, with the release of PyTorch 1.0, there was a much heavier focus on paving the path from research to production and scalability.

Even with TensorFlow/Keras and PyTorch taking the top spots in popularity, there are many other frameworks available for ML/DL tasks. Developed by Berkeley AI Research (BAIR), Caffe (Jia *et al* 2014b) is the next most popular framework. It is written in C++ with a Python interface. An attempt to create its successor, Caffe2, was started by Facebook in 2017, but the project development was merged into PyTorch in spring of 2018. The next popular ML framework, Theano (Bergstra *et al* 2010, 2011, Bastien *et al* 2012) was developed by the Montreal Institute for Learning Algorithms (MILA) at the Université de Montréal. In fall of 2017, with rising popularity of TensorFlow, it was announced that Theano would cease any major development. However, minor updates are still added, and some experts today still rely on Theano for ML tasks. The Apache MXNet (Chen *et al* 2015) was developed by the Apache Software Foundation. It is written in multiple languages, including Python, C++, MATLAB, R, Julia, JavaScript, Scala, Go, and Perl. While less popular, MXNet is a very high-performance framework, typically training faster with less computational resource demand. Microsoft Cognitive Toolkit, CNTK (Seide and Agarwal 2016), is Microsoft Research's implementation for commercial-level distributed DL, and can be added as a library for Python, C#, or C++. Deeplearning for Java, DL4J (Team 2016), is a computing framework written for Java and the Java Virtual Machine (JVM). Chainer (Tokui *et al* 2015) is a purely Python-based framework developed by Preferred Networks, Inc. It is one of the early adopters and popularizers of the 'define-by-run' format of DL, which is now the concept in both PyTorch and Eager Execution in TensorFlow. Last but not the least, MATLAB (The MathWorks, Inc., Natick, Massachusetts, United States) also has its own DL package, but is usually behind the aforementioned open-source packages. However, MATLAB has recognized this and allows developers to import models built in other frameworks such as TensorFlow, Keras, PyTorch, MXNET, Caffe, etc, to be used in it.

For most researchers starting DL studies, either TensorFlow or PyTorch is a great framework to begin with, as they offer plenty of tools that will satisfy majority of users' needs. Other less popular frameworks may find their utility in special use cases. For example, a developer that needs an application written in Java may choose to use DL4J for a more seamless integration of the DL development and deployment in the Java environment, as opposed to training the model in Python with Tensorflow, and then porting the trained model over to Java.

Beyond that, it is largely due to personal preference. Even through TensorFlow and PyTorch began with different philosophies, their latest releases are beginning to converge in terms of their capabilities and available tools. Both PyTorch and TensorFlow have fairly large community backings, with many models and codes available freely online (table 2). In general, a number of popular model implementations can be easily found through a quick online search query.

As for coding languages, for scientific research, it is often recommended to use Python in order to have access to TensorFlow and PyTorch. Python is the most popular language used by the DL community, and is one of the most utilized languages for industry. It is highly recommended to install Python via the Anaconda Distribution, a Python and R distribution focused on scientific computing. It simplifies package and environment management, and contains most of the core packages needed for scientific computing.

4.2. Datasets

It is widely recognized that DL models are data hungry, and that, in the medical realm, the dataset size is often limited. In the era of big data for ML, enormous efforts have been spent to aggregate medical data and make them publicity available. Here we only point out a few representative ones.

The National Cancer Institute (NCI) provides a large resource index for researchers (https://www.cancer.gov/research/resources), including a large list of over 80 available databases and datasets. The datasets cover research areas that include cancer treatment, cancer biology, cancer omics, screening and detection, cancer heath disparities, cancer and public health, cancer diagnosis, cancer statistics, cancer prevention, causes of cancer, bioinformatics, and cancer survivorship. Of these datasets, The Cancer Imaging Archive (TCIA) (Clark *et al* 2013) is a notable database and is constantly growing. Funded by NCI, TCIA consists of mostly DICOM images of CT, MRI, and PET, as well as organ structures, radiation therapy plans, and dose data. Another notable archive, The Cancer Genome Atlas (TCGA) (Tomczak *et al* 2015) is a database to catalogue major genomic mutations that cause cancer, with the goal to improve diagnostic methods and treatment standards.

Moreover, within the medical physics community, various challenges are hosted every year by different organizations, e.g. the grand challenge from American Association of Physicists in Medicine (AAPM) and the multimodal Brain Tumor Segmentation (BRATS) challenge from the Medical Image Computing and Computer Assisted Intervention Society (MICCAI). Datasets released from these challenges are important resources for different problems.

5. Mathematical aspects of deep neural networks

Previous sections have presented a brief introduction on DL, major techniques and resources. Starting from here, we will switch gears to discuss technical aspects of DL to gain some insights about its capability and potential challenges.

5.1. Mathematical interpretations of deep neural networks

From mathematical perspective, a DNN can be viewed as an effective tool that is able to approximate a function arbitrarily well under suitable mathematical conditions (Hornik *et al* 1989, Hornik 1991, Pinkus 1999). Both the depth and width of a neural network are among the most important factors that affect its approximation power. As we have defined in section 2.2, depth refers to the total number of layers in a DNN and width refers to the number of neurons in a layer. Earlier results on the approximation property, i.e. universal approximation, suggested that a wide class of functions can indeed be approximated by neural networks with only one hidden layer, although the number of neurons may increase exponentially, as we increase the required level of accuracy of the approximation (Cybenko 1989, Funahashi 1989, Barron 1993). Later, many studies showed that the depth of DNNs helps with the approximation. For example, approximation with DNNs leads to an exponential or polynomial reduction in the number of neurons while maintaining the same level of approximation accuracy (Cohen *et al* 2016, Eldan and Shamir 2016, Liang and Srikant 2016, Mhaskar *et al* 2016). Furthermore, Delalleau and Bengio (2011), Telgarsky (2015) and Telgarsky (2016) presented concrete examples that there exist functions that can be more efficiently represented with DNNs rather than shallow networks. Recently, Yarotsky (2018) analyzed the dependence of optimal approximation rate on the depth for ReLU activated DNNs. When approximating a multivariate polynomial, Rolnick and Tegmark (2018) mathematically proved that the total number of neurons in DNNs should grow linearly with respect to the number of variables of the polynomial. In He *et al* (2018), the authors investigated the connection between linear finite element functions and ReLU activated DNNs. They proposed an efficient ReLU activated DNN structure to represent any linear finite element functions and theoretically established that at least two hidden layers are needed in a ReLU activated DNN to represent any linear finite element functions. More recently, Shen *et al* (2019e) provided an intriguing analysis on ReLU activated DNNs via a nonlinear approximation with composite dictionaries. They demonstrated the advantage of depth over width quantitatively. Other than generic DNNs, theoretical analysis on the popular

ResNet (He *et al* 2016a, 2016b) were also provided (Veit *et al* 2016, Lin and Jegelka 2018, Ma and Wang 2019). Lu *et al* (2017) investigated the efficiency of the depth of ReLU activated DNNs from a different angle by proving that there exist classes of wide neural networks which cannot be realized by any narrow network whose depth is no more than a polynomial bound, indicating that network depth is more effective than width. On the other hand, Hanin (2019) proved that there is a minimal width of ReLU activated DNNs to guarantee approximation of continuous functions. Their results indicated that a good DNN cannot be too narrow either, otherwise we cannot approximate continuous functions even with infinite depth.

It is also of note that, in addition to viewing a DNN as an effective function approximation tool, there are also other mathematical interpretations of DNNs. For example, because of the recursively composite structure, it is natural to view a DNN as a certain dynamic system (Cessac 2010). Specifically, Weinan (2017) made an inspiring observation that ResNet can be viewed as the forward Euler scheme solving an ordinary differential equation, and linked training of DNNs with the optimal control problems. Similar observations were also made by other groups (Chang *et al* 2017, Li and Shi 2017, Sonoda and Murata 2017, Chang *et al* 2018) and rigorous justification of the link was given (Thorpe and van Gennip 2018, Weinan *et al* 2019). These interpretations are useful in certain contexts (Gregor and LeCun 2010). Due to space consideration, we will not go into more details. Interested readers can refer to relevant literatures.

5.2. Fundamental requirements to build a successful deep neural network model

As mentioned in section 2.1, the fundamental problem of ML (including DL) is to decipher the unknown function $y = f^*(x)$, from a set of observed training data A_{train}. The unknown $f^*(x)$ is approximated by a function $f(x|\theta)$, where θ represents parameters of the function to be determined through a training process. In the context of DL, this function takes the form of DNN, as specified in equation (2). After training, performance of the solution function $y = f(x|\theta)$ is evaluated in a test set of data A_{test}.

The underlying challenge of ML/DL is that the training is performed by minimizing a certain loss function, e.g. $L(\theta) = \sum_{(x_i,y_i)\in A_{train}} y_i - f(x_i|\theta)^2$, defined on the observed training dataset, while the trained model is evaluated on the testing dataset. Both the training and the testing datasets are finite and may be of a relatively small size. These datasets may not completely represent the structure of the unknown function and data distribution. In principle, the proper loss function to recover $y = f^*(x)$ should be $L^*(\theta) = \sum_{(x,y)} y - f(x|\theta)^2$. Note the difference between the actual loss function $L(\theta)$ and the desired loss function $L^*(\theta)$ in terms of the range of summation. In other words, it is desired to train the model in the complete dataset covering all the points (x, y). However, this is impossible, as otherwise we would already have all the information about the unknown function and there would be no need for solving the ML/DL problem.

With these in mind, there are two fundamental requirements for a successful ML/DL process. (1) The training and testing datasets should be representative of the function $y = f^*(x)$, so that it is unbiased to use the actual loss function $L(\theta)$ defined with the training dataset as a proxy of the desired loss function $L^*(\theta)$ and to evaluate the model performance on the testing dataset. (2) For the specific problem of interest, the function $f(x|\theta)$ used to approximate the unknown function $y = f^*(x)$ based on data should have sufficient *capacity* and *flexibility*. Here capacity measures the scope of mappings that the function $f(x|\theta)$ is able to learn by adjusting model parameters θ. High flexibility is a consequence of high capacity, as it permits the flexible use of the model $f(x|\theta)$ to approximate the unknown but potentially very complex mapping $y = f^*(x)$, such as mapping from an image to a class label.

Based on the discussion in section 5.1 looking at DNNs from the function approximation perspective, a DNN serves the purpose of providing a class of functions with a large capacity that can be trained to flexibly approximate a very complex mapping from the input to the output. It is this fact that contributes to the successful applications of DNNs to solve a number of problems. Take a typical problem of classifying a picture into cat or dog group as an example, a successful function achieving this goal, i.e. mapping an image into a class label, is likely very complicated in its mathematical form. The function has to extract some features from the image, based on which the two targeted groups can be discriminated. Although describing these features verbally is relatively easy for a human, describing them rigorously in a mathematical form is apparently difficult. Nonetheless, after training a DNN with a large number of images, it can easily learn the mapping function, although described in a neural network format. In this example, the flexibility and capacity attributes of DNN are of central importance.

Nonetheless, the first requirement on data distribution for the success of DL is beyond the reach of DNNs. In other words, given a sufficient amount of data, a DNN is capable of extracting complex patterns hidden in the data, yielding a successful DL application. Yet a DNN cannot generate additional information not contained in data.

6. Challenges for deep learning in general and in medical physics problems

After a brief introduction in the last section about how DNNs can meet in part the requirements of ML, we will discuss specific challenges that DL is facing, including both generic challenges as a ML tool and specific ones in the medical physics contexts. We would like to emphasize that presenting these challenges does not mean using DL to solve medical physics problems is impractical. On the contrary, we hope the presentation could help us understanding DL better, yielding impactful studies with scientific and practical values.

6.1. Data size

Data size is the fundamental challenge for all ML/DL approaches. Numerous studies have demonstrated the success of DL in problems with sufficient data. However, applying DL to problems with small-size datasets is dangerous.

This data size issue is typically discussed in combination with the dimensionality of the problem of interest. Let us denote the number of data samples with n and the dimensionality with p. The severity of this data size problem indeed depends on both quantities. The fundamental cause of this problem is that each observed data sample only carries information locally valid at the sample location, or at most in a region surrounding the sample's location, if we assume smoothness of information with respect to the data sample location. Unfortunately, for a fixed number of n samples, as the dimension p increases, these samples become sparser and sparser in the space. Specifically, suppose we place n samples in a unit cube of dimension p. Each data point only carries information in a small region around it, e.g. a small ball of a radius of ε. The total volume of all the ε-balls around the data points is $nV_p(\varepsilon) = n\varepsilon^p V_p(1)$, where $V_p(1)$ is the volume of a p-dimensional unit sphere. Given the volume of the unit cube $1^p = 1$, the ratio of the volume that we have measured information to the entire volume is $n\varepsilon^p V_p(1)$. For a fixed n, this ratio quickly approaching zero. This implies that for a fixed n, as the problem dimension p increases, there are more and more gap spaces between samples, in which there is no information measured. Hence it becomes increasingly difficult to draw conclusions about the global data structure of the unknown function to be reconstructed from the observed samples. This is often called the *curse of dimensionality*, indicating that the number of needed samples grows rapidly with dimensionality.

While the comprehensiveness of the training data is critically important to DL methods, and ML methods in general, a proper metric to evaluate data comprehensiveness is still missing. It is hard to quantify how well the inherent pattern of one specific type of data is represented by the training data. Moreover, instead of the original data space, the inherent pattern is often embedded in a task-related low-dimensional latent subspace. Unfortunately, such a latent space is generally unknown and it is a particularly challenging task to estimate its dimensionality. Although it is not exact, the dimensionality of the original data space p is often considered as an alternative measure to represent the complexity of data. For general ML purposes, it usually requires number of training samples n to be much larger than p to ensure comprehensiveness of training data. A problem with small n and large p is usually challenge to solve.

It is worth mentioning that the data size n refers to the number of independent samples, rather than the number of features measured per sample. There is a clear cut between the notion of big data and big-size data. Big data refers to the situation with a large number of independent samples. A dataset containing only a few samples but numerous measurements per sample is only big in its size, but may not be considered as big data. Meanwhile, independence of data samples is problem specific. In many image processing contexts, images can be broken into patches which can be considered as a large number of independent samples. It is necessary to justify independence based on problems of interest.

Coming to medical physics problems, data size is a particular concern for several reasons. Data collection for most of the medical problems is often time-consuming, labor-intensive, and expensive. The creation of a useful training dataset usually requires notable domain knowledge from clinicians. Moreover, data pre-processing, which may also require substantial efforts, is commonly needed for medical data, given the sometimes-poor standardization, distinct clinical protocols, and inevitable human errors in data collection. Furthermore, although a relatively large pool of historical data might be available at some large medical institutions, the amount of data available for each specific task is likely still limited. On top of all these issues, legal and privacy concerns are additional factors that prevent assembling large-scale medical datasets and sharing them among institutions. As a consequence, medical datasets are often in the regime of small n. On the other hand, many medical tasks are actually large p problems due to the involvement of high-dimensional data such as multi-modality medical images and other patient medical records. It is also often easier to measure a lot of information per patient than to have a large number of patients. Combining these facts, the small n large p nature of many problems indeed poses a significant challenge for ML/DL in medical applications.

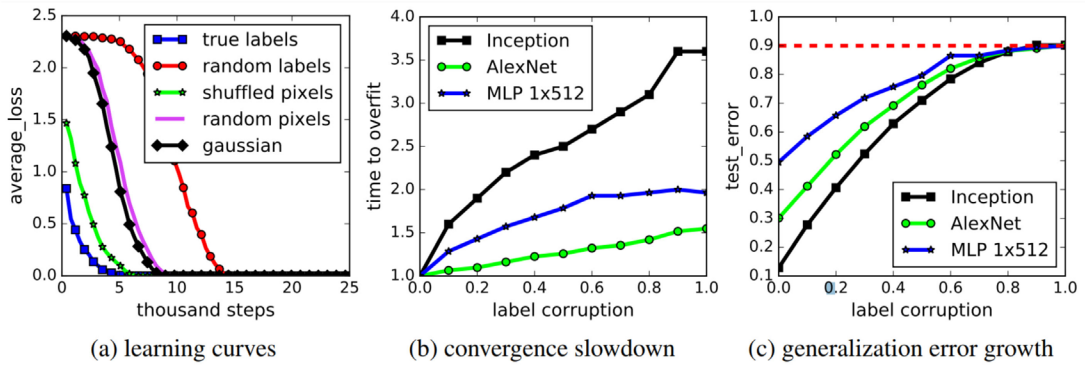

Figure 10. Figure 1 from Zhang *et al* (2016). Fitting random labels and random pixels on CIFAR10. (a) Shows the training loss of various experiment settings decaying with the training steps. (b) Shows the relative convergence time with different label corruption ratio. (c) Shows the test error (also the generalization error since training error is 0) under different label corruptions.

6.2. Overfitting

Overfitting refers to the situation where a ML/DL model is trained to closely or even exactly fit a particular set of training data, but fail to learn the general underlying data pattern to maintain generality. This typically happens when the number of parameters in an ML/DL model is too large to be justified by the available training data.

6.2.1. Overfitting in deep learning

To date, almost all the successful DL applications were achieved on large-scale datasets. In these examples, although the number of network parameters is larger than the number of training data, it has been empirically and surprisingly observed that the developed DNN models still generalized well. The mathematical reasons behind this phenomenon is yet unknown. A recent study (Belkin *et al* 2018) has shed some lights to bridge the classical ML and the DL regimes. They provided several examples to show that over-parametrization, i.e. using a large model with more parameters than data, helps to improve the performance of DL models. However, we would like to emphasize that these conclusions were drawn based on the assumption that the amount of training data is sufficient to well approximate the complete data distribution. Generality of the conclusions to different problems and the theory behind remain to be further explored.

On the other hand, in the cases with a relatively small-size dataset, the concern of overfitting should be always kept in mind. This is particularly true because of the large capability of DNN to fit data, which in fact is one of its virtues and at the same time one of its drawbacks. One interesting study was reported by Zhang *et al* (2016), which received the best paper award of 2017 International Conference on Learning Representations (ICLR). One of the key results is depicted in figure 10, where large scale DL models were trained to fit different datasets synthesized based on CIFAR10 dataset (Krizhevsky and Hinton 2009), a widely used public image dataset in the computer science community. More specifically, the authors generated multiple synthetic datasets by randomly assigning fake image labels, shuffling image pixels, adding random noise to image pixels, and replacing the images by random Gaussian noise etc. As such, the generated training data set lost 'correct' information and hence the models built on it did not have meanings. However, as shown in figure 10(a), the DL model was still able to perfectly fit the original CIFAR10 dataset as well as all the synthesized with 0% training loss, although it took more training steps to achieve so (figure 10(b)). Note that there were no true data patterns to learn in the synthetized datasets, since either image labels or image data were corrupted and hence the trained model did not generalize anymore in these cases (figure 10(c)). These results clearly demonstrated the fact that large-scale DL models are powerful enough to memorize the whole training dataset.

6.2.2. Potential approaches to prevent overfitting

Multiple strategies have been designed to address the overfitting problem. One trivial way is to reduce the number of network parameters. However, the network capacity would be reduced as well, which may deteriorate the model performance. Hence, it requires insights about the problem of interest and the knowledge about the strengths and weaknesses of different network structures to design an effective network model.

Another effective and commonly employed approach is *data augmentation*, i.e. to expand a dataset by synthesizing additional realistic samples from available samples. The way of augmentation varies depending on the context. For instance, one common augmentation technique used in image analysis related tasks is to apply random translations, rotations, deformations, and adding low-level random noise to training images to create new training samples. A more advanced augmentation strategy is to perform interpolation based on the distribution of existing training data (Chawla *et al* 2002). More recently, the generative adversarial networks (GANs),

(Goodfellow *et al* 2014) have demonstrated its power to synthesize realistic training samples (see section 3.3). However, empirical evidence recently showed that GANs learn distributions defined on a fairly small region of the input data domain. Hence, the capability of GAN may be also limited in terms of generating truly independent data (Arora and Zhang 2017).

Synthesizing new data based on physics principles is also a potential approach to increase data size. For instance, in the problem of x-ray scatter estimation in cone beam CT (Maier *et al* 2018a, Nomura *et al* 2019), Monte Carlo simulation can be employed to generate realistic scatter signals based on patient anatomy and x-ray illumination setup (Jia *et al* 2012). Nonetheless, performance of models trained by these data critically depends on the realism of data synthesis. If there is a systematic bias of the synthesized data from real data, e.g. due to inaccurate modeling of the physics process, the trained models would give incorrect outputs, when being applied to real situations.

Incorporating regularizations to model parameters can also help preventing overfitting (Moody 1992). Classical regularizers including weight decay and sparsity have shown to be effective in many studies (Boureau and Cun 2008, Glorot *et al* 2011a, Krizhevsky *et al* 2012, Venkatesh *et al* 2017). Along the same line, more sophisticated approaches, such as group sparsity (Scardapane *et al* 2017) and structured sparsity (Wen *et al* 2016), were introduced. Moreover, layers and architecture specifically designed for DNN have also been put forward (Glorot *et al* 2011a, Wan *et al* 2013, Srivastava *et al* 2014). One of the most successful regularization techniques in DNN model is adding dropout layers (Srivastava *et al* 2014). It randomly turns off some neurons with a certain probability at each iteration of the training phase, so that the number of activated network parameters is reduced. Despite the remarkable success achieved by these methods, the study Zhang *et al* (2016) revealed that overfitting may still remain for large-scale DNNs, even when various regularizations are applied. Further efforts in developing more effective regularization techniques are definitely needed.

6.2.3. Monitoring overfitting with validation

Preventing DL models from overfitting is still an ongoing research direction, since no existing solution can achieve satisfactory performance in a general situation. Therefore, monitoring overfitting along the training process becomes essential.

One effective way is to have a stand-alone validation dataset and monitor the model performance on it along the training process, rather than to pursue the lowest training loss evaluated on the training dataset. Hyper-parameters, such as number of layers, size of each layer, learning rate, etc, can be tuned in order to reduce the validation loss in a trial-and-error fashion. Moreover, through monitoring the validation loss along the training process, it is also possible to select the model at the epoch when the smallest validation loss is obtained.

A more comprehensive way is to perform k-fold cross validation (Kohavi 1995). We illustrate the idea of k-fold cross validation in figure 11 using a simple three-fold case as an example. The dataset is first split randomly into training data and testing data. The testing data are saved for model evaluation after model development, while the training data are randomly split evenly into three folds. Multiple trials of training can be performed to develop multiple models. In each trial, one fold of data is left out as validation data, such that the performance of the model trained on data in other folds can be validated along the training steps. Similar to single-fold validation, one can adjust hyper-parameters according to the validation performance averaged over the k-fold cross validation process. With finely tuned hyper-parameters, the model achieving the best performance on training and validation data, or an integrated model generated via certain fusion techniques (Ngiam *et al* 2011) can be selected as the final model. Up to this point, the model should have only seen the training and validation datasets. Its performance will be finally evaluated on the reserved testing data.

Although these validation strategies have been shown to be very useful to identify overfitting, they do not guarantee a generalizable model, especially when the data size problem discussed in section 6.1 occurs. The reason is that, when the data are too sparse, the data space not observed in training and validation datasets allows a large freedom to manipulate the model without affecting training and validation performance. Meanwhile, when the validation dataset is not comprehensive enough to represent the complete data distribution, validity of using validation loss to prevent overfitting degenerates, as the model would be tuned to favor the validation data, but not the testing data.

We also emphasize that the testing data should not be involved in the training process in any form, before the final model is constructed and ready for model evaluation. A high model performance evaluated on the testing dataset likely indicates a successful model development. On the other hand, if the model performance is tested to be low, the model development is likely failed. Further tuning the model to improve performance on the testing data risks information leakage from the testing dataset to the training process and bias the model development. In this case, evaluating model performance on testing dataset is not a true test anymore but rather becomes validation.

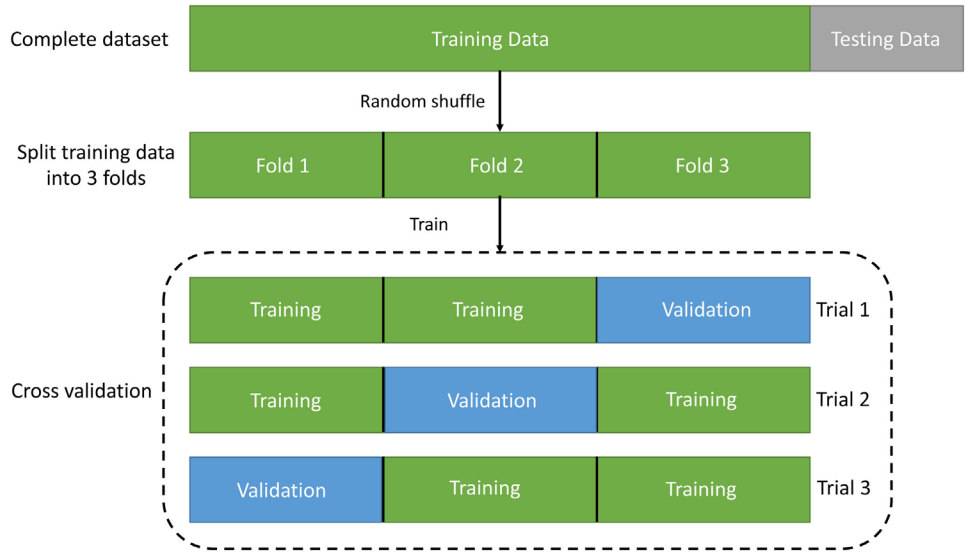

Figure 11. Illustration of three-fold cross validation.

6.3. Interpretation

Interpretability of a model refers to the level of transparency in knowledge and information that the ML/DL model extracts from input data. Interpretability helps to understand and validate the correctness of the established model, and hence it is crucial to many real-world applications. For instance, let us assume a model has been developed to identify babies in images. An interpretable model provides information that it extracts for such decision making, e.g. a baby's face in the image. Based on this fact, one can judge if the model is trained to identify the true information, or it mistakenly incorporates other incorrect information in the images to make the decision, such as by identifying a milk bottle. Interpretability probably requires more attentions in medical applications than in other applications, given the fact that resulting DL models may potentially influence the clinical decision-making process, and thereby the patient care quality, safety, and outcome.

Given the complicated multi-layer structures and numerous numerical operations performed by each layer, interpreting DL models is often challenging by itself. A DL model is sometimes referred as a 'black box', since users have little knowledge about what is inside the highly nonlinear model. This has become one of the main obstacles that prevent further applications of DL models in many real-world tasks.

To date, extensive studies have been performed to investigate and enhance interpretability of DL models. A recent survey in (Zhang and Zhu 2018b) gave a comprehensive review about the current status. Major efforts have been devoted to two aspects, i.e. enhancing interpretability of DL models and developing interpretable models. To interpret existing DL models, t-SNE embedding was proposed to map DNN representations onto a low-dimensional space to visualize the relationship among data (Maaten and Hinton 2008). A number of other methods have also been developed to visualize regions in input data that DL models consider to be important (Simonyan *et al* 2013, Springenberg *et al* 2014, Zeiler and Fergus 2014, Mahendran and Vedaldi 2015, Dosovitskiy and Brox 2016, Nguyen *et al* 2017, Samek *et al* 2017). Meanwhile, interpretable DNN architectures have been put forward (Che *et al* 2016, Chen *et al* 2016, Sturm *et al* 2016, Zhang *et al* 2018). Among them, Zhang *et al* (2018) proposed a typical interpretable structure by facilitating disentangle representation, while Sabour *et al* (2017) designed the capsule network to extract the rationale of detection qualitatively. Several pioneer studies have been conducted to design interpretable architectures for medical related applications (Che *et al* 2016, Sturm *et al* 2016).

6.4. Model uncertainty and robustness

The constructed DL model itself has *uncertainty* for three reasons. First, there are typically more network parameters than training data, which makes the optimization problem essentially underdetermined. The resulting model after the training stage depends on many factors, such as initial solution, hyper-parameters, etc. Second, due to the highly nonconvex form of the loss function, the training of a DL model ends up with one of the local minima. Conducting training multiple times does not necessarily produce the same model or models with the same performance. Third, the training algorithms are often of a stochastic nature. For instance, the data are split randomly into batches at each epoch. This fact introduces discrepancy among models even trained with the identical network structure using the same dataset.

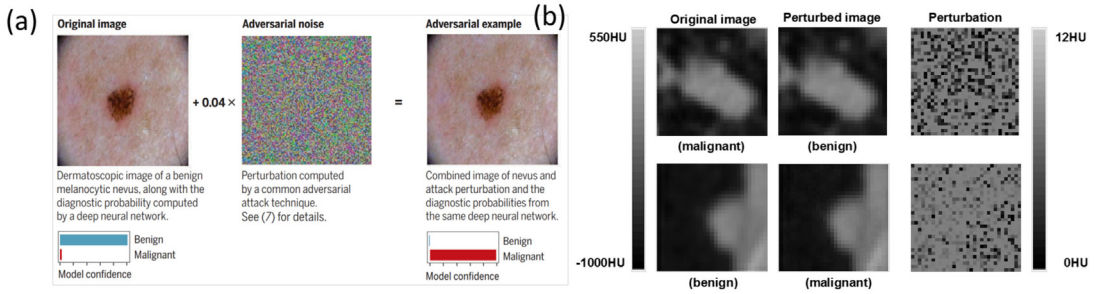

Figure 12. (a) Examples from Finlayson *et al* (2019), used with permission. Model prediction and confidence level are shown. (b) An example showing prediction results a CT-based lung nodule classification network is changed by adding a noise to input CT images. Predicted labels from the DL model are given below images.

Robustness is another important aspect to consider, when evaluating the performance of a trained DL model, particularly in the medical context. It refers to the sensitivity of the model output with respect to perturbations to the input variables. A high sensitivity means poor robustness. As perturbations always exist in the real world, a DL model with poor robustness can lead to unstable performance, and thereby deteriorate its practical value. Unfortunately, it has been recognized lately that DL models are not quite robust. Multiple recent studies have shown empirical results, demonstrating that output of DL models may be sensitive to small changes on input data (Kurakin *et al* 2016, Papernot *et al* 2016, Akhtar and Mian 2018, Yuan *et al* 2019).

Specific to the healthcare context, since the established DL model has the potential to affect clinical decision making, robustness of DL models strongly correlates with the patient safety and healthcare quality. Yet a recent article in Science discussed deep vulnerability of DL models in healthcare (Finlayson *et al* 2019). Figure 12(a) is one example, showing the output of an DL model predicting skin cancer is altered by adding a small-scale noise to the input image. The same behavior has also been observed in other DL models for image-based diagnosis of pneumothorax from chest x-ray, diabetic retinopathy from retinal fundoscopy (Finlayson *et al* 2019), and the classification of lung nodules based on CT images (Tsai *et al* 2019) (figure 12(b)).

Although theoretical guidelines about this robustness issue are few and far between, we can still summarize some practical insights on how to improve model robustness. One effective way is to enhance diversity of the training dataset. More specifically, if the model is found to be not robust, we may include those vulnerable samples into the training dataset and further refine the DL models, such that the models observe those samples and learn the proper way to defend against the perturbations (Yuan *et al* 2019). This process can be repeatedly performed, until the robustness level is satisfactory.

6.5. Correlation, causality, and incomplete information

A DNN can be very powerful to decipher the correlation between input and output variables, even a very complex one that is difficult to capture by other traditional ML methods. Yet it faces the same two challenges as other ML methods. First, it is hard to answer the question whether the discovered correlation is real or not. While one can be very creative in terms of using a DNN to establish a relationship between two variables, it may be dangerous to do so, if it is hard to justify that there is indeed a relationship between them. There are many examples on the Internet showing spurious correlations between different quantities around us. This problem is further compounded by the large flexibility and capacity of a DNN. The DNN can easily generate a mapping between the input and output variables with a high level of accuracy, and hence one may tend to believe the observed relationship is real. The second challenge is that the observed correlation does not necessarily mean *causality*, i.e. one is the cause of the other. While correlation is often acceptable to support decision making in medicine, it is highly desirable to derive causality, which would help us apply the DL model in the correct context.

In many situations, the input variables only contain partial information to determine the output variables due to incomplete measurements. For instance, encouraging results have been achieved in medical image reconstruction using only a limited amount of measurements by mapping from a low-quality reconstructed image to a high-quality one with a DNN. In this case, it is important to think where the additional information is from. In the image reconstruction example, it is likely that the training process learns inherent prior knowledge about images, e.g. its appearance, shape, size, etc. The prior knowledge is appended to the input low-quality image to derive the output. Understanding the sources of the information provided by the network is valuable for not only gaining insights about the technologies itself, for further improving the methods, but also for a safe and confident deployment of the developed techniques in routine applications.

6.6. Effective and efficient learning

On the technical side, one has to admit that current DL technologies are still at their infancy. It typically requires a large amount of data to train the model and the training process usually converges slowly. This is in stark contrast with human intelligence that can easily and quickly understand situations and problems based on only a few observations. Hence, it is a central topic overcoming the data-hunger nature in medical regime and making the learning more efficient.

Transfer learning (Pan *et al* 2010) is one of the effective approaches to reduce the requirement on data size. As its name suggested, transfer learning refers to a method where a model developed for one task is reused as the starting point for another model on a second, but related task. After a DNN model is trained for the first task, it is believed that the model already gains certain knowledge about the problem of interest. Hence, the network parameters can be reused as the initial solutions, when training the same network on the second related task using limited training data. Optionally, at the retraining stage, the first DNN model can be adapted or adjusted to better suit the second problem. Take an image classification problem in a medical context as an example, the publicly available DNNs such as VGG model may be reused, since the model has been successfully developed to recognize key features in an image, such as edges. Although the initial DNN model are trained using image data probably unrelated to the second problem, the capability of recognizing edges and other features from the pre-training stage is expected to be critical for the second problem, and hence should be preserved.

Network structure is very important for each specific task, as a predefined network with a specific structure can reduce the number of network links and hence unknown parameters as compared to using a fully connected network, making the training more effective. Therefore, before starting a DL study, it is worthwhile to carefully design the network structure based on the targeted problem. Meanwhile, it is equally important to avoid using an inappropriate network structure. Additionally, the emerging research of neural architecture search (Elsken *et al* 2018), a subfield of automated ML, shows potential in searching for effective DNN architectures for different datasets and tasks.

One important reason why a human is able to quickly learn from a small-size dataset is the capability of reasoning. However, this is quite difficult for a network. Current DL approaches mainly rely on data. It is expected that, if we could incorporate rule-based reasoning capability in a DL model, the training process could be more effective.

Because of the importance of improving learning effectiveness, not only in medical area, there have been a vast number of approaches developed over the years to address this issue, such as few-shot learning (Fink 2005, Li *et al* 2006), imitation learning (Ho and Ermon 2016, Duan *et al* 2017), meta learning (Santoro *et al* 2016), domain adaptation (Glorot *et al* 2011b), etc. As it is impossible to enumerate all of them, interested readers could refer to relevant publications.

On the computation side, using GPU programming has become the standard practice to achieve a high computational efficiency. Many modern GPUs and GPU workstations are designed specifically for DL purposes. For instance, some modern GPUs supports the use of the half precision (16-bit) floating point format, because lower precision calculations seem to be not critical for DL (Micikevicius *et al* 2017). This reduces the computational load and memory requirement compared to the conventional 32-bit floating point format for single precision operations.

7. Recommendations

Before concluding this review article, we will present some recommendations regarding practices on DL research. These recommendations are derived based on the extensive discussions about the insights of DNNs, their challenges, and potential solutions. We first summarize commonly encountered challenges as well as the corresponding recommendations in table 3, and then provide more discussions below.

7.1. Graphics processing unit

As mentioned previously, training of a DNN usually requires extensive computations due to a large model size and the amount of data. It is beneficial to use GPU programing to overcome this challenge. While there are a few high-end GPU workstations from different vendors built specifically for DL purposes, almost all modern GPU cards can be used in a standard computer workstation or cluster to enhance computational efficiency. On the software side, most of the DL computational frameworks, such as TensorFlow and PyTorch, have supports of GPU-based computation, which allow users to employ GPUs without the need of in-depth knowledge in GPU programming.

7.2. Data and model construction

As data is the basis of DL, before performing a new study, the data should be carefully inspected to avoid issues such as errors, outliers, bias, and confounding factors, etc. Note that in many medical physics applications, it

Table 3. Commonly encountered challenges in DL and recommendations.

	Challenges	Recommendations
Model training	Training efficiency	• Employ high-end GPUs and well-established computational packages
	Data quality	• Understand and improve data quality
	Data quantity	• Increase data size
		• Use data augmentation and synthesize data
	Overfitting	• Start with a small-scale model
		• Use validation strategies to monitor overfitting
		• Use regularization strategies
Model evaluation	Model evaluation	• Test the model in testing dataset blind to training stage
		• Use appropriate and comprehensive metrics
	Interpretability	• Interpret models
		• Use interpretable models
	Robustness and uncertainty	• Be aware of robustness and uncertainty
		• Discuss model robustness and uncertainty, and evaluate if possible
Other	Reproducibility	• Use public datasets, if possible
		• Share data and model
		• Present research in detail

is challenging to get the real ground truth. For example, contours in image segmentation and labels in disease diagnosis are mostly given by clinicians. The data generation process is usually subjective and human error or uncertainty is almost unavoidable. Hence, having multiple clinicians to label the same dataset and properly integrating the labels are helpful to ensure validity of developed model.

Based on the specific task of interest, one should carefully select or design a DNN with an appropriate network structure. At this stage, it is important to keep the balance between model complexity and capacity. On one hand, a sufficiently complex DNN should be employed to allow accurately modeling the underlying complex data pattern. On the other hand, the model should be kept as simple as possible to reduce the number of unknown parameters and therefore the required amount of data. In the circumstance with a limited data size, designing suitable data augmentation strategies based on the specific context is encouraged and often helpful. Yet data augmentation strategies should be justified and examined to avoid potential bias.

Given the extraordinary capability of a DNN, the risk of overfitting is often high and one should be always alerted about this fact. Cross validation as presented in section 6.2.3 is an effective approach during the model training stage, when data size is limited. However, we need to keep in mind that it does not eliminate the risk of overfitting. In the case with an intermediate to large-size dataset, splitting the data into training, validation and test is recommended. Appropriate strategies to avoid overfitting, such as dropout, regularizations etc, could be employed during the training process. While repeated tuning hyper-parameters during the training step is unavoidable to achieve the best performance, it is very important to do so by observing model performance on the validation dataset only, but not the testing dataset.

7.3. Model evaluation

At the model evaluation stage, the model should be evaluated in a testing dataset that is completely blind to the training process to avoid information leakage from the testing dataset to the training stage, which may bias the constructed model to favor the testing dataset. Ideally, the testing dataset should be independent from the training dataset, e.g. from a different data source.

It is equally important to use proper evaluation metrics for the specific problem or context to objectively assess model performance. First, the evaluation metric has to be scientifically sound. For instance, for a model predicting a very rare disease, simply reporting model accuracy tested on a population dataset is likely not sufficient. A trivial model simply predicting no disease in all cases would give a very high accuracy level, but is practically useless due to poor sensitivity. More comprehensive evaluation metrics such as sensitivity, specificity, area under the curve, and many others should be employed for an objective evaluation from different perspectives. Second, evaluation metrics and passing criteria are problem specific. For many imaging related tasks, DL algorithms are very capable of generating visually very appealing images. However, the images have to be assessed in an appropriate context to evaluate if they meet requirements of the clinical task. For image-based diagnosis purpose, using metrics regarding low-contrast object detection is much more important than just presenting visually appealing images. The CT image intensity accuracy is critical for radiotherapy dose calculation, whereas geometry accuracy should be the main focus for cone beam CT in the context of image-guided radiotherapy.

7.4. Model interpretation

Correctly interpreting the resulting model is at the core of DL. Care should be given to properly and objectively make conclusions and statements. For instance, a DL model can discover correlations, but it would be typically challenging to establish causality. While correlation is acceptable in medicine, the fact that the constructed model reports correlation should be clearly stated to avoid misleading readers. With this in mind, human interpretation and judgement is of importance to avoid spurious results. One example is that the shoe sizes and student's intelligence quotient scores would be found correlated by a ML method, but neither of them is the cause of the other. The impact of confounding factor in this case, age, may not be identified by DL and it has to rely on the researcher to figure this out. Another situation is that the input data contain both the independent variables and other variables correlated to them, whereas the output only depends on the independent variables. A trained DL model may build a relationship not only between the output and the independent variables, but also incorrectly between the output and other correlated variables. Again, carefully examining and testing the developed DL model is critical to make sure the model is constructed properly.

It would be valuable in a study to use model interpretation techniques to understand the exact information that DNN extracts from the training data. This would help us to gain confidence and insights about the study and avoid over or incorrect interpretations. Meanwhile, we agree that it is not always possible to decipher reasons in the constructed DNN due to technical challenges. In this case, certain discussions and justifications regarding the end-to-end rationale learnt by a DNN model is important and encouraged. The discovered correlation should have a certain reasonable explanation to support it.

7.5. Model uncertainty and robustness

Every study has uncertainty, to a large or small extent. The uncertainty of the trained DL model may be caused by numerous factors including, but not limited to inherent noise in the data, limited data size, imperfectly selected model, training residual error, non-local minima of the training process, to name a few. The model robustness seems to be an issue caused by the DNN. Therefore, commenting on the model uncertainty and robustness is also an important aspect, so that readers can bear this in mind for effective use of the trained model.

The researchers are encouraged to perform comprehensive investigation and estimation about the magnitude and cause of uncertainty and robustness. For instance, the model may be trained multiple times to study the uncertainty due to the stochastic nature of the training process. Input data may be deliberately perturbed to investigate robustness of the trained DL model with respect to noise in the input data. We agree that such a comprehensive evaluation may be a tedious process and may be beyond the scope of some initial studies proposing a DL model. However, certain types of estimation are desired, so that the readers can assess the DL model from a more objective angle. For example, for a DL model claiming a better performance than a classical ML model but with the uncertainty of a similar size to, or even larger than the performance margin, it is probably inappropriate to claim the definitive advantage of the DL model.

7.6. Study reproducibility

Study reproducibility is an important feature of science (Open Science Collaboration 2015). In the DL field, there has been an increasing need to enhance research reproducibility. Doing so will not only be critical for the deep understanding of published techniques, it would also be vital for the community to continuously advance by building success on top of existing successes.

Data is the cornerstone of DL. Different from other fields where large-scale standard datasets are frequently available, e.g. the CIFAR-10 and CIFAR-100 datasets for image classification tasks, medical physics field has relatively less publicly available datasets. Section 4 has listed a few. When performing new studies, we recommend to consider using public datasets as the first priority. This would put studies from multiple groups on the same ground, facilitating cross comparisons. In addition, the sizes of public datasets are often larger than those of private datasets, which is a feature favored by DL research. Quality of these datasets is expected to be carefully inspected before being opened to public, releasing the users from tedious tasks of data cleaning.

Along the same vein, data sharing is another route to overcome the data-hunger nature of DL. Nonetheless, data sharing in medicine is challenging due to not only technical, but also practical and legal concerns. Standard solution for large-scale as-needed data sharing is yet unclear. Efforts from the community are encouraged to promote data sharing and to develop novel techniques to achieve this goal under practical constraints.

Another key factor to enhance reproducibility is to provide sufficient details in publications. This includes not only the typically presented information such as network structure, training data, algorithms, etc, but also a comprehensive list of hyper-parameter values, training strategies etc. It is also noted that it is probably impractical to list all the details in a paper. For instance, some hyper-parameters are dynamically adjusted during the training process. Hence, it is suggested to clearly specify those key parameters and details that may significantly contribute to the results.

Meanwhile, we point out that training a DL model has inherent randomness. Examples include the commonly used stochastic gradient descent algorithm (Bottou 2010) and the dropout strategy (Srivastava *et al* 2014). The randomness in the training process, together with the highly non-convex landscape of the loss function, poses a barrier to exactly reproduce a reported training process, as each time running the training process would land at a certain local minimum following a random trajectory. On the other hand, it is more important for others to reproduce the claimed performance in a study, as opposed to the exact training process.

7.7. Do not forget classical machine learning models

As the last note, while we are continuously impressed by DL and are devoted to study DL, it is not a bad idea to keep eyes open about classical ML models for two reasons. First, when both classical and DL models can achieve the same, or similar performance, classical models may be preferred because of less computational demand, low requirement on data size, transparent model meanings, robustness etc. Second, even though DL model attain a better performance, classical models may still offer some insights about the problem and data, which are valuable for further improving the DL models.

8. Concluding remarks

In this review article, we did not review extensively the rapid advancements of DL in medical physics area over the past several years, as existing review articles have conducted excellent jobs for this purpose. Instead, we focused our presentations on introducing DL technologies and discussing challenges faced by DL. This choice was made to provide medical physics researchers interested in DL an objective overview about the method and to help them start the endeavor. Because of limited space and efforts, our presentations cannot be complete. We hope the presentation can serve as an initial point for researchers to explore more in depth.

DL is a very capable and potentially very impactful tool to advance medical physics in near future. As is true for any other powerful tools, DL is not perfect and faces its own challenges. The exact mathematical theory behind DL is still lacking. The applications of DL in certain medical physics problems require careful and insightful thinking. Attention should be payed, when using DL technologies to solve problems in clinical practice.

Over a very short period, DL has achieved tremendous success in a spectrum of problems in medical physics. The way forward is bright and challenging. While there are still low-hanging fruits in certain areas, there is still a long way to develop accurate, robust, and clinically impactful DL tools to ultimately bring the potential of DL from bench to the bed side and to eventually benefit patient care. Achieving this goal would require a close collaboration among physicists, mathematicians, computer scientists, data scientists, and clinicians. We hope that researchers will continuously perform novel and impactful studies towards realizing this goal and improve healthcare with this amazing technology.

Acknowledgments

This work is supported in part by National Institutes of Health grants R01CA227289, R37CA214639, and R01CA237269. Dr Bin Dong is supported in part by Beijing Natural Science Foundation grant Z180001 and National Natural Science Foundation of China grant 11831002. Drs Chenyang Shen and Dan Nguyen acknowledge the support by the Seed grants from the Department of Radiation Oncology, University of Texas Southwestern Medical Center.

ORCID iDs

Chenyang Shen https://orcid.org/0000-0001-6490-6555
Dan Nguyen https://orcid.org/0000-0002-9590-0655

References

Abadi M *et al* 2016 Tensorflow: large-scale machine learning on heterogeneous distributed systems (arXiv:1603.04467)
Aizenberg I, Aizenberg N N and Vandewalle J P 2000 *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications* (Berlin: Springer) (https://doi.org/10.1007/978-1-4757-3115-6)
Akhtar N and Mian A 2018 Threat of adversarial attacks on deep learning in computer vision: a survey *IEEE Access* **6** 14410–30
Alpaydin E 2009 *Introduction to Machine Learning* (Cambridge, MA: MIT Press)
Arora S and Zhang Y 2017 Do gans actually learn the distribution? An empirical study (arXiv:1706.08224)
Bai W *et al* 2017 Semi-supervised learning for network-based cardiac MR image segmentation *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention* (Berlin: Springer) pp 253–60
Balagopal A, Kazemifar S, Nguyen D, Lin M-H, Hannan R, Owrangi A and Jiang S 2018 Fully automated organ segmentation in male pelvic CT images *Phys. Med. Biol.* **63** 245015
Barron A R 1993 Universal approximation bounds for superpositions of a sigmoidal function *IEEE T. Inform. Theory* **39** 930–45

Bastien F, Lamblin P, Pascanu R, Bergstra J, Goodfellow I, Bergeron A, Bouchard N, Warde-Farley D and Bengio Y 2012 Theano: new features and speed improvements (arXiv:1211.5590)
Belkin M, Hsu D, Ma S and Mandal S 2018 Reconciling modern machine learning and the bias-variance trade-off *PNAS* **116** 15849–54
Bergstra J *et al* 2011 Theano: Deep learning on gpus with python *NIPS 2011, BigLearning Workshop* (Granada, Spain: Citeseer) pp 1–48
Bergstra J, Breuleux O, Bastien F, Lamblin P, Pascanu R, Desjardins G, Turian J, Warde-Farley D and Bengio Y 2010 Theano: a CPU and GPU math expression compiler *Proc. of the Python for scientific computing Conf. (SciPy) (Austin, TX)*
Bojarski M *et al* 2016 End to end learning for self-driving cars (arXiv:1604.07316)
Bottou L 2010 *Large-Scale Machine Learning with Stochastic Gradient Descent* (Berlin: Springer) pp 177–86
Boureau Y-L and Cun Y L 2008 Sparse feature learning for deep belief networks *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 1185–92
Cessac B 2010 A view of neural networks as dynamical systems *Int. J. Bifurcat. Chaos* **20** 1585–629
Cha K H, Hadjiiski L, Samala R K, Chan H P, Caoili E M and Cohan R H 2016 Urinary bladder segmentation in CT urography using deep-learning convolutional neural network and level sets *Med. Phys.* **43** 1882–96
Chang B, Meng L, Haber E, Ruthotto L, Begert D and Holtham E 2018 Reversible architectures for arbitrarily deep residual neural networks *Thirty-Second AAAI Conf. on Artificial Intelligence*
Chang B, Meng L, Haber E, Tung F and Begert D 2017 Multi-level residual networks from dynamical systems view (arXiv:1710.10348)
Chawla N V, Bowyer K W, Hall L O and Kegelmeyer W P 2002 SMOTE: synthetic minority over-sampling technique *J. Artif. Intell. Res.* **16** 321–57
Che Z, Purushotham S, Khemani R and Liu Y 2016 Interpretable deep models for ICU outcome prediction *AMIA Annual Symp. Proc.* (Bethesda, MD: American Medical Informatics Association) p 371
Chen H, Zhang Y, Zhang W, Liao P, Li K, Zhou J and Wang G 2017a Low-dose CT via convolutional neural network *Biomed. Opt. Express* **8** 679–94
Chen L, Shen C, Zhou Z, Maquilan G, Albuquerque K, Folkert M R and Wang J 2019a Automatic PET cervical tumor segmentation by combining deep learning and anatomic prior *Phys. Med. Biol.* **64** 085019
Chen L, Zhou Z, Sher D, Zhang Q, Shah J, Pham N-L, Jiang S and Wang J 2019b Combining many-objective radiomics and 3D convolutional neural network through evidential reasoning to predict lymph node metastasis in head and neck cancer *Phys. Med. Biol.* **64** 075011
Chen M, Shi X, Zhang Y, Wu D and Guizani M 2017b Deep features learning for medical image analysis with convolutional autoencoder neural network *IEEE Trans. Big Data* (https://doi.org/10.1109/TBDATA.2017.2717439)
Chen T *et al* 2015 Mxnet: a flexible and efficient machine learning library for heterogeneous distributed systems (arXiv:1512.01274)
Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I and Abbeel P 2016 Infogan: interpretable representation learning by information maximizing generative adversarial nets *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 2172–80
Chen Y, Xie Y, Zhou Z, Shi F, Christodoulou A G and Li D 2018 Brain MRI super resolution using 3D deep densely connected neural networks *2018 IEEE 15th Int. Symp. on Biomedical Imaging* (Piscataway, NJ: IEEE) pp 739–42
Cheng J-Z, Ni D, Chou Y-H, Qin J, Tiu C-M, Chang Y-C, Huang C-S, Shen D and Chen C-M 2016 Computer-aided diagnosis with deep learning architecture: applications to breast lesions in US images and pulmonary nodules in CT scans *Sci. Rep.* **6** 24454
Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H and Bengio Y 2014 Learning phrase representations using RNN encoder-decoder for statistical machine translation *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing* pp 1724–34
Chollet F 2015 *Keras* (Github) (https://github.com/fchollet/keras)
Ciresan D C, Meier U, Masci J, Gambardella L M and Schmidhuber J 2011 Flexible, high performance convolutional neural networks for image classification *Twenty-Second Int. Joint Conf. on Artificial Intelligence*
Clark K *et al* 2013 The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository *J. Digit. Imaging* **26** 1045–57
Cohen N, Sharir O and Shashua A 2016 On the expressive power of deep learning: a tensor analysis *Conf. on Learning Theory* pp 698–728
Cybenko G 1989 Approximation by superpositions of a sigmoidal function *Math. Control Signals Syst.* **2** 303–14
Deeplearning4j 2016 Deeplearning4j: open-source distributed deep learning for the JVM *Apache Softw. Foundat. License* **2**
Delalleau O and Bengio Y 2011 Shallow versus deep sum-product networks *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 666–74
Deng J, Dong W, Socher R, Li L-J, Li K and Fei-Fei L 2009 Imagenet: a large-scale hierarchical image database *2009 IEEE Conf. on Computer Vision and Pattern Recognition* (Piscataway, NJ: IEEE) pp 248–55
Despres P and Jia X 2017 A review of GPU-based medical image reconstruction *Phys. Medica* **42** 76–92
Diamant A, Chatterjee A, Vallières M, Shenouda G and Seuntjens J 2019 Deep learning in head & neck cancer outcome prediction *Sci. Rep.* **9** 2764
Dice L R 1945 Measures of the amount of ecologic association between species *Ecology* **26** 297–302
Dosovitskiy A and Brox T 2016 Inverting visual representations with convolutional networks *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition* pp 4829–37
Duan Y, Andrychowicz M, Stadie B, Ho O J, Schneider J, Sutskever I, Abbeel P and Zaremba W 2017 One-shot imitation learning *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 1087–98
Eldan R and Shamir O 2016 The power of depth for feedforward neural networks *Conf. on Learning Theory* pp 907–40
El-Sawy A, Hazem E-B and Loey M 2016 CNN for handwritten arabic digits recognition based on LeNet-5 *Int. Conf. on Advanced Intelligent Systems and Informatics* (Berlin: Springer) pp 566–75
Elsken T, Metzen J H and Hutter F J 2018 Neural architecture search: a survey *J. Mach. Learn. Res.* **20** 1–21
Esteva A, Kuprel B, Novoa R A, Ko J, Swetter S M, Blau H M and Thrun S 2017 Dermatologist-level classification of skin cancer with deep neural networks *Nature* **542** 115
Feng Z, Nie D, Wang L and Shen D 2018 Semi-supervised learning for pelvic MR image segmentation based on multi-task residual fully convolutional networks *2018 IEEE 15th Int. Symp. on Biomedical Imaging* (Piscataway, NJ: IEEE) pp 885–8
Fink M 2005 Object classification from a single example utilizing class relevance metrics *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 449–56
Finlayson S G, Bowers J D, Ito J, Zittrain J L, Beam A L and Kohane I S 2019 Adversarial attacks on medical machine learning *Science* **363** 1287–9
François-Lavet V, Henderson P, Islam R, Bellemare M G and Pineau J 2018 An introduction to deep reinforcement learning *Found. Trends® Mach. Learn.* **11** 219–354

Fu J, Yang Y, Singhrao K, Ruan D, Chu F I, Low D A and Lewis J H 2019 Deep learning approaches using 2D and 3D convolutional neural networks for generating male pelvic synthetic CT from MRI *Med. Phys.* **46** 3788–98
Fukushima K 1980 Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position *Biol. Cybern.* **36** 193–202
Funahashi K-I 1989 On the approximate realization of continuous mappings by neural networks *Neural Networks* **2** 183–92
Gjesteby L, Yang Q, Xi Y, Shan H, Claus B, Jin Y, De Man B and Wang G 2017a Deep learning methods for CT image-domain metal artifact reduction *Proc. SPIE* **10391** 103910W
Gjesteby L, Yang Q, Xi Y, Zhou Y, Zhang J and Wang G 2017b Deep learning methods to guide CT image reconstruction and reduce metal artifacts *Proc. SPIE* **10132** 101322W
Glorot X, Bordes A and Bengio Y 2011a Deep sparse rectifier neural networks *Proc. of the 14th Int. Conf. on Artificial Intelligence and Statistics* pp 315–23
Glorot X, Bordes A and Bengio Y 2011b Domain adaptation for large-scale sentiment classification: a deep learning approach *Proc. of the 28th Int. Conf. on Machine Learning* pp 513–20
Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT press)
Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial nets *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 2672–80
Graves A, Wayne G and Danihelka I 2014 Neural turing machines (arXiv:1410.5401)
Gregor K and LeCun Y 2010 Learning fast approximations of sparse coding *Proc. of the 27th Int. Conf. on Int. Conf. on Machine Learning* (Madison, WI: Omnipress) pp 399–406
Guo Y, Gao Y and Shen D 2016 Deformable MR prostate segmentation via deep feature learning and sparse patch matching *IEEE Trans. Med. Imaging* **35** 1077–89
Han X 2017 MR-based synthetic CT generation using a deep convolutional neural network method *Med. Phys.* **44** 1408–19
Han Y S, Yoo J and Ye J C 2016 Deep residual learning for compressed sensing CT reconstruction via persistent homology analysis (arXiv:1611.06391)
Hanin B 2019 Universal function approximation by deep neural nets with bounded width and ReLU activations *Mathematics* **7** 992
Hansen D C, Landry G, Kamp F, Li M, Belka C, Parodi K and Kurz C 2018 ScatterNet: A convolutional neural network for cone-beam CT intensity correction *Med. Phys.* **45** 4916–26
He J, Li L, Xu J and Zheng C 2018 ReLU deep neural networks and linear finite elements (arXiv:1807.03973)
He K, Zhang X, Ren S and Sun J 2016a Deep residual learning for image recognition *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition* pp 770–8
He K, Zhang X, Ren S and Sun J 2016b Identity mappings in deep residual networks *European Conf. on Computer Vision* (Berlin: Springer) pp 630–45
Ho J and Ermon S 2016 Generative adversarial imitation learning *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 4565–73
Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80
Hornik K 1991 Approximation capabilities of multilayer feedforward networks *Neural Netw.* **4** 251–7
Hornik K, Stinchcombe M and White H 1989 Multilayer feedforward networks are universal approximators *Neural Netw.* **2** 359–66
Hu P, Wu F, Peng J, Bao Y, Chen F and Kong D 2017 Automatic abdominal multi-organ segmentation using deep convolutional neural network and time-implicit level sets *Int. J. Comput. Assist. Radiol. Surg.* **12** 399–411
Hu P, Wu F, Peng J, Liang P and Kong D 2016 Automatic 3D liver segmentation based on deep learning and globally optimized surface evolution *Phys. Med. Biol.* **61** 8676
Hua K-L, Hsu C-H, Hidayati S C, Cheng W-H and Chen Y-J 2015 Computer-aided classification of lung nodules on computed tomography images via deep learning technique *OncoTargets Ther.* **8** 2015–22
Ibragimov B and Xing L 2017 Segmentation of organs-at-risks in head and neck CT images using convolutional neural networks *Med. Phys.* **44** 547–57
Ioffe S and Szegedy C 2015 Batch normalization: Accelerating deep network training by reducing internal covariate shift *Int. Conf. on Machine Learning* pp 448–56
Iqbal Z *et al* 2019 Super-Resolution 1H magnetic resonance spectroscopic imaging utilizing deep learning *Front Oncol.* **9** 1010
Ivakhnenko A G and Lapa V G 1965 *Cybernetic Predicting Devices* (New York: CCM Information Corporation)
Jackson P C 1985 *Introduction to Artificial Intelligence* (New York: Dover Publications)
Jia X, Yan H, Cervino L, Folkerts M and Jiang S B 2012 A GPU tool for efficient, accurate, and realistic simulation of cone beam CT projections *Med. Phys.* **39** 7368–78
Jia X, Ziegenhein P and Jiang S B 2014a GPU-based high-performance computing for radiation therapy *Phys. Med. Biol.* **59** R151
Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S and Darrell T 2014b Caffe: convolutional architecture for fast feature embedding *Proc. of the 22nd ACM Int. Conf. on Multimedia* (New York: ACM) pp 675–8
Jung H, Gonzalez Y, Shen C, Klages P, Albuquerque K and Jia X 2019a Deep-learning–assisted automatic digitization of applicators in 3D CT image-based high-dose-rate brachytherapy of gynecological cancer *Brachytherapy* **18** 841–51
Jung H, Shen C, Gonzalez Y, Albuquerque K and Jia X 2019b Deep-learning assisted automatic digitization of interstitial needles in 3D CT image based high dose-rate brachytherapy of gynecological cancer *Phys. Med. Biol.* **64** 215003
Kallenberg M *et al* 2016 Unsupervised deep learning applied to breast density segmentation and mammographic risk scoring *IEEE Trans. Med. Imaging* **35** 1322–31
Kang E, Min J and Ye J C 2017 A deep convolutional neural network using directional wavelets for low-dose x-ray CT reconstruction *Med. Phys.* **44** e360–e375
Kazemifar S, McGuire S, Timmerman R, Wardak Z, Nguyen D, Park Y, Jiang S and Owrangi A 2019 MRI-only brain radiotherapy: assessing the dosimetric accuracy of synthetic CT images generated using a deep learning approach *Radiother. Oncol.* **136** 56–63
Kelly B, Matthews T P and Anastasio M A 2017 Deep learning-guided image reconstruction from incomplete data (arXiv:1709.00584)
Kingma D and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)
Kohavi R 1995 A study of cross-validation and bootstrap for accuracy estimation and model selection *Ijcai (Montreal, Canada)* pp 1137–45
Krizhevsky A and Hinton G 2009 Learning multiple layers of features from tiny images *Technical Report* University of Toronto
Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 1097–105
Kumar A, Sattigeri P and Fletcher T 2017 Semi-supervised learning with gans: Manifold invariance with improved inference *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 5534–44

Kumar D, Wong A and Clausi D A 2015 Lung nodule classification using deep features in CT images *2015 12th Conf. on Computer and Robot Vision* (Piscataway, NJ: IEEE) pp 133–8
Kurakin A, Goodfellow I and Bengio S 2016 Adversarial examples in the physical world (arXiv:1607.02533)
LeCun Y and Bengio Y 1995 Convolutional networks for images, speech, and time series *The Handbook of Brain Theory and Neural Netw.* **3361** 1995
LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44
LeCun Y, Boser B E, Denker J S, Henderson D, Howard R E, Hubbard W E and Jackel L D 1990 Handwritten digit recognition with a back-propagation network *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 396–404
LeCun Y, Boser B, Denker J S, Henderson D, Howard R E, Hubbard W and Jackel L D 1989 Backpropagation applied to handwritten zip code recognition *Neural Comput.* **1** 541–51
Lee H, Pham P, Largman Y and Ng A Y 2009 Unsupervised feature learning for audio classification using convolutional deep belief networks *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 1096–104
Lei Ba J, Kiros J R and Hinton G E 2016 Layer normalization (arXiv:1607.06450)
Li F-F, Fergus R and Perona P 2006 One-shot learning of object categories *IEEE Trans. Pattern Anal. Mach. Intell.* **28** 594–611
Li Z and Shi Z 2017 Deep residual learning and pdes on manifold (arXiv:1708.05115)
Liang S and Srikant R 2016 Why deep neural networks for function approximation? (arXiv:1610.04161)
Liang X, Chen L, Nguyen D, Zhou Z, Gu X, Yang M, Wang J and Jiang S 2018 Generating synthesized computed tomography (CT) from cone-beam computed tomography (CBCT) using CycleGAN for adaptive radiation therapy (arXiv:1810.13350)
Lin H and Jegelka S 2018 ResNet with one-neuron hidden layers is a Universal Approximator *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 6172–81
Litjens G *et al* 2016 Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis *Sci. Rep.* **6** 26286
Liu F, Yadav P, Baschnagel A M and McMillan A B 2019a MR-based treatment planning in radiation therapy using a deep learning approach *J. Appl. Clin. Med. Phys.* **20** 105–14
Liu Y *et al* 2019b MRI-based treatment planning for proton radiotherapy: dosimetric validation of a deep learning-based liver synthetic CT generation method *Phys. Med. Biol.* **64** 145015
Lu Z, Pu H, Wang F, Hu Z and Wang L 2017 The expressive power of neural networks: A view from the width *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 6231–9
Ma C and Wang Q 2019 *A priori* estimates of the population risk for residual networks (arXiv:1903.02154)
Ma G, Shen C and Jia X 2018 Low dose CT reconstruction assisted by an image manifold prior (arXiv:1810.12255)
Ma K, Wang J, Singh V, Tamersoy B, Chang Y-J, Wimmer A and Chen T 2017 Multimodal image registration with deep context reinforcement learning *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention* (Berlin: Springer) pp 240–8
Maaten L v d and Hinton G 2008 Visualizing data using t-SNE *J. Mach. Learn. Res.* **9** 2579–605
Madani A, Moradi M, Karargyris A and Syeda-Mahmood T 2018 Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation *2018 IEEE 15th Int. Symp. on Biomedical Imaging* (Piscataway, NJ: IEEE) pp 1038–42
Mahendran A and Vedaldi A 2015 Understanding deep image representations by inverting them *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition* pp 5188–96
Maier J, Berker Y, Sawall S and Kachelrieß M 2018a Deep scatter estimation (DSE): feasibility of using a deep convolutional neural network for real-time x-ray scatter prediction in cone-beam CT *Proc. SPIE* **10573** 105731L
Maier J, Sawall S, Knaup M and Kachelrieß M 2018b Deep scatter estimation (DSE): accurate real-time scatter estimation for x-ray CT using a deep convolutional neural network *J. Nondestruct. Eval.* **37** 57
Manning C, Surdeanu M, Bauer J, Finkel J, Bethard S and McClosky D 2014 The Stanford CoreNLP natural language processing toolkit *Proc. of 52nd Annual Meeting of the Association For Computational Linguistics: System Demonstrations* pp 55–60
McCulloch W S and Pitts W 1943 A logical calculus of the ideas immanent in nervous activity *Bull. Math. Biophys.* **5** 115–33
Mehta J and Majumdar A 2017 RODEO: robust DE-aliasing autoencOder for real-time medical image reconstruction *Pattern Recognit.* **63** 499–510
Mendelson E B 2018 Artificial intelligence in breast imaging: potentials and limitations *Am. J. Roentgenol.* **212** 293–9
Mhaskar H, Liao Q and Poggio T 2016 Learning functions: when is deep better than shallow (arXiv:1603.00988)
Micikevicius P *et al* 2017 Mixed precision training (arXiv:1710.03740)
Mikolov T, Karafiát M, Burget L, Černocký J and Khudanpur S 2010 Recurrent neural network based language model *11th annual Conf. of the Int. Speech Communication Association*
Milletari F, Navab N and Ahmadi S-A 2016 V-net: Fully convolutional neural networks for volumetric medical image segmentation *2016 4th Int. Conf. on 3D Vision* (Piscataway, NJ: IEEE) pp 565–71
Miotto R, Wang F, Wang S, Jiang X and Dudley J T 2017 Deep learning for healthcare: review, opportunities and challenges *Briefings Bioinform.* **19** 1236–46
Mnih V *et al* 2015 Human-level control through deep reinforcement learning *Nature* **518** 529–33
Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D and Riedmiller M 2013 Playing atari with deep reinforcement learning (arXiv:1312.5602)
Mohamed A-R, Dahl G and Hinton G 2009 Deep belief networks for phone recognition *Nips Workshop on Deep Learning for Speech Recognition And Related Applications (Vancouver, Canada)* p 39
Moody J E 1992 The effective number of parameters: an analysis of generalization and regularization in nonlinear learning systems *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 847–54
Nair V and Hinton G E 2010 Rectified linear units improve restricted boltzmann machines *Proc. of the 27th Int. Conf. on Machine Learning* pp 807–14
Neyshabur B, Li Z, Bhojanapalli S, LeCun Y and Srebro N 2018 Towards understanding the role of over-parametrization in generalization of neural networks (arXiv:1805.12076)
Ngiam J, Khosla A, Kim M, Nam J, Lee H and Ng A Y 2011 Multimodal deep learning *Proc. of the 28th Int. Conf. on Machine Learning* pp 689–96
Nguyen A, Clune J, Bengio Y, Dosovitskiy A and Yosinski J 2017 Plug & play generative networks: conditional iterative generation of images in latent space *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition* pp 4467–77
Nguyen D, Barkousaraie A S, Shen C, Jia X and Jiang S 2019a Generating Pareto optimal dose distributions for radiation therapy treatment planning *Medical Image Computing and Computer Assisted Intervention MICCAI 2019 (Lecture Notes in Computer Science* vol 11769) (Berlin: Springer) pp 59–67

Nguyen D, Jia X, Sher D, Lin M-H, Iqbal Z, Liu H and Jiang S B 2019b Three-dimensional radiotherapy dose prediction on head and neck cancer patients with a hierarchically densely connected U-net deep learning architecture *Phys. Med. Biol.* **64** 065020
Nguyen D, Long T, Jia X, Lu W, Gu X, Iqbal Z and Jiang S 2019c A feasibility study for predicting optimal radiation therapy dose distributions of prostate cancer patients from patient anatomy using deep learning *Sci. Rep.* **9** 1076
Nie D, Gao Y, Wang L and Shen D 2018 ASDNet: Attention based semi-supervised deep networks for medical image segmentation *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention* (Springer) pp 370–8
Nilsson N J and Nilsson N J 1998 *Artificial Intelligence: a New Synthesis* (Burlington, MA: Morgan Kaufmann)
Nomura Y, Xu Q, Shirato H, Shimizu S and Xing L 2019 Projection-domain scatter correction for cone beam computed tomography using a residual convolutional neural network *Med. Phys.* **46** 3142–55
Novak R, Bahri Y, Abolafia D A, Pennington J and Sohl-Dickstein J 2018 Sensitivity and generalization in neural networks: an empirical study (arXiv:1802.08760)
Nyflot M J, Thammasorn P, Wootton L S, Ford E C and Chaovalitwongse W A 2019 Deep learning for patient-specific quality assurance: identifying errors in radiotherapy delivery by radiomic analysis of gamma images with convolutional neural networks *Med. Phys.* **46** 456–64
Odena A 2016 Semi-supervised learning with generative adversarial networks (arXiv:1606.01583)
Oktay O *et al* 2016 Multi-input cardiac image super-resolution using convolutional neural networks *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention* (Springer) pp 246–54
Open Science Collaboration 2015 Estimating the reproducibility of psychological science *Science* **349** aac4716
Pan S J and Yang Q 2010 A survey on transfer learning *IEEE Trans. Knowl. Data Eng.* **22** 1345–59
Papernot N, McDaniel P, Jha S, Fredrikson M, Celik Z B and Swami A 2016 The limitations of deep learning in adversarial settings *2016 IEEE European Symp. on Security and Privacy* (IEEE) pp 372–87
Paszke A *et al* 2017 Automatic differentiation in PyTorch *NIPS 2017 Workshop Autodiff* (https://openreview.net/forum?id=BJJsrmfCZ)
Pham C-H, Ducournau A, Fablet R and Rousseau F 2017 Brain MRI super-resolution using deep 3D convolutional networks *2017 IEEE 14th Int. Symp. on Biomedical Imaging* (IEEE) pp 197–200
Pinkus A 1999 Approximation theory of the MLP model in neural networks **8** 143–95
Pratx G and Xing L 2011 GPU computing in medical physics: a review *Med. Phys.* **38** 2685–97
Rajpurkar P *et al* 2017 Chexnet: radiologist-level pneumonia detection on chest x-rays with deep learning (arXiv:1711.05225)
Ravì D, Wong C, Deligianni F, Berthelot M, Andreu-Perez J, Lo B and Yang G 2017 Deep learning for health informatics *IEEE J. Biomed. Health Inform.* **21** 4–21
Ren X, Xiang L, Nie D, Shao Y, Zhang H, Shen D and Wang Q 2018 Interleaved 3D-CNN s for joint segmentation of small-volume structures in head and neck CT images *Med. Phys.* **45** 2063–75
Rivenson Y, Zhang Y, Günaydın H, Teng D and Ozcan A 2018 Phase recovery and holographic image reconstruction using deep learning in neural networks *Light Sci. Appl.* **7** 17141
Rolnick D and Tegmark M 2018 The power of deeper networks for expressing natural functions *Int. Conf. on Learning Representations (ICLR)* (https://openreview.net/forum?id=SyProzZAW)
Ronneberger O, Fischer P and Brox T 2015 U-net: convolutional networks for biomedical image segmentation *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention* (Berlin: Springer) pp 234–41
Roth H R, Lu L, Farag A, Shin H-C, Liu J, Turkbey E B and Summers R M 2015 Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation *Int. Conf. on Medical Image Computing And Computer-Assisted Intervention* (Berlin: Springer) pp 556–64
Rumelhart D E, Hinton G E and Williams R J 1986 Learning representations by back-propagating errors *Nature* **323** 533–36
Russakovsky O *et al* 2015 Imagenet large scale visual recognition challenge *Int. J. Comput. Vis.* **115** 211–52
Sabour S, Frosst N and Hinton G E 2017 Dynamic routing between capsules *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 3856–66
Saffari N, Rashwan H A, Herrera B, Romani S, Arenas M and Puig D 2018 On improving breast density segmentation using conditional generative adversarial networks *CCIA* pp 386–93
Sahiner B, Pezeshk A, Hadjiiski L M, Wang X, Drukker K, Cha K H, Summers R M and Giger M L 2019 Deep learning in medical imaging and radiation therapy *Med. Phys.* **46** 1–36
Sak H, Senior A and Beaufays F 2014 Long short-term memory recurrent neural network architectures for large scale acoustic modeling *15th Annual Conf. of the Int. Speech Communication Association*
Samek W, Wiegand T and Müller K-R 2017 Explainable artificial intelligence: understanding, visualizing and interpreting deep learning models (arXiv:1708.08296)
Santoro A, Bartunov S, Botvinick M, Wierstra D and Lillicrap T 2016 Meta-learning with memory-augmented neural networks *Proc. of The 33rd Int. Conf. on Machine Learning* ed B Maria Florina and Q W Kilian pp 1842–50 (*Proc. of Mach. Learn. Res.*)
Scardapane S, Comminiello D, Hussain A and Uncini A 2017 Group sparse regularization for deep neural networks *Neurocomputing* **241** 81–9
Scherer D, Müller A and Behnke S 2010 Evaluation of pooling operations in convolutional architectures for object recognition *Int. Conf. on Artificial Neural Networks* (Berlin: Springer) pp 92–101
Schlemper J, Caballero J, Hajnal J V, Price A N and Rueckert D 2018 A deep cascade of convolutional neural networks for dynamic MR image reconstruction *IEEE Trans. Med. Imaging* **37** 491–503
Schroff F, Kalenichenko D and Philbin J 2015 Facenet: A unified embedding for face recognition and clustering *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition* pp 815–23
Schuster M and Paliwal K K 1997 Bidirectional recurrent neural networks *IEEE Trans. Signal Process.* **45** 2673–81
Seide F and Agarwal A 2016 CNTK: Microsoft's open-source deep-learning toolkit *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (New York: ACM) p 2135
Sensakovic W F and Mahesh M 2019 Role of the medical physicist in the health care artificial intelligence revolution *J. Am. College Radiol.* **16** 393–4
Shen C, Gonzalez Y, Chen L, Jiang S and Jia X 2018 Intelligent parameter tuning in optimization-based iterative CT reconstruction via deep reinforcement learning *IEEE Trans. Med. Imaging* **37** 1430–9
Shen C, Gonzalez Y, Klages P, Qin N, Jung H, Chen L, Nguyen D, Jiang S B and Jia X 2019a Intelligent inverse treatment planning via deep reinforcement learning, a proof-of-principle study in high dose-rate brachytherapy for cervical cancer *Phys. Med. Biol.* **64** 115013
Shen C, Gonzalez Y, Nguyen D, Chen L, Jiang S and Jia X 2019b Virtual treatment planner for radiation therapy via deep reinforcement learning *Int. Conf. on the Use of Computers in Radiation Therapy*

Shen C, Nguyen D, Gonzalez Y, Chen L, Jiang S and Jia X 2019c Operating a treatment planning system using a deep-reinforcement-learning based virtual treatment planner for intensity-modulated radiation therapy treatment planning *Annual Meeting of American Association of Physicists in Medicine*
Shen C, Tsai M-Y, Gonzalez Y, Chen L, Jiang S B and Jia X 2019d Quality-guided deep reinforcement learning for parameter tuning in iterative CT reconstruction *Proc. SPIE* **11072** 1107203
Shen W, Zhou M, Yang F, Yang C and Tian J 2015 Multi-scale convolutional neural networks for lung nodule classification *Int. Conf. on Information Processing in Medical Imaging* (Berlin: Springer) pp 588–99
Shen Z, Yang H and Zhang S 2019e Nonlinear approximation via compositions *Neural Networks* **119** 74–84
Shin H-C, Orton M R, Collins D J, Doran S J and Leach M O 2013 Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 1930–43
Silver D *et al* 2016 Mastering the game of Go with deep neural networks and tree search *Nature* **529** 484
Silver D *et al* 2017 Mastering the game of Go without human knowledge *Nature* **550** 354
Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition (arXiv:1409.1556)
Simonyan K, Vedaldi A and Zisserman A 2013 Deep inside convolutional networks: visualising image classification models and saliency maps (arXiv:1312.6034)
Sønderby C K, Raiko T, Maaløe L, Sønderby S K and Winther O 2016 Ladder variational autoencoders *Advances in Neural Information Processing Systems* pp 3738–46
Sonoda S and Murata N 2017 Double continuum limit of deep neural networks *ICML Workshop Principled Approaches to Deep Learning*
Springenberg J T 2015 Unsupervised and semi-supervised learning with categorical generative adversarial networks (arXiv:1511.06390)
Springenberg J T, Dosovitskiy A, Brox T and Riedmiller M 2014 Striving for simplicity: The all convolutional net (arXiv:1412.6806)
Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: a simple way to prevent neural networks from overfitting *J. Mach. Learn. Res.* **15** 1929–58
Sturm I, Lapuschkin S, Samek W and Müller K-R 2016 Interpretable deep neural networks for single-trial EEG classification *J. Neurosci. Methods* **274** 141–5
Sun W, Tseng T-L B, Zhang J and Qian W 2017 Enhancing deep convolutional neural network scheme for breast cancer diagnosis with unlabeled data *Comput. Med. Imaging Graph.* **57** 4–9
Sun W, Zheng B and Qian W 2016 Computer aided lung cancer diagnosis with deep learning algorithms *Proc. SPIE* **9785** 97850Z
Sutton R S and Barto A G 2018 *Reinforcement Learning: an Introduction* (Cambridge, MA: MIT Press)
Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition* pp 1–9
Tang X, Wang B and Rong Y 2018 Artificial intelligence will reduce the need for clinical medical physicists *J. Appl. Clin. Med. Phys.* **19** 6–9
Telgarsky M 2015 Representation benefits of deep feedforward networks (arXiv:1509.0810186)
Telgarsky M 2016 Benefits of depth in neural networks *29th Annual Conference on Learning Theory (Proc. of Machine Learning Research* vol 49) pp 1517–39
Thompson R F *et al* 2018a Artificial intelligence in radiation oncology imaging *Int. J. Radiat. Oncol. Biol. Phys.* **102** 1159–61
Thompson R F *et al* 2018b Artificial intelligence in radiation oncology: A specialty-wide disruptive transformation? *Radiother. Oncol.* **129** 421–6
Thorpe M and van Gennip Y 2018 Deep limits of residual neural networks (arXiv:1810.11741)
Tokui S, Oono K, Hido S and Clayton J 2015 Chainer: a next-generation open source framework for deep learning *Proc. of Workshop on Machine Learning Systems (LearningSys) in the 29th Annual Conf. on Neural Information Processing Systems* pp 1–6
Tomczak K, Czerwińska P and Wiznerowicz M 2015 The cancer genome atlas (TCGA): an immeasurable source of knowledge *Contemp. Oncol.* **19** A68
Tomori S, Kadoya N, Takayama Y, Kajikawa T, Shima K, Narazaki K and Jingu K 2018 A deep learning-based prediction model for gamma evaluation in patient-specific quality assurance *Med. Phys.* **45** 4055–65
Tsai M-Y, Shen C and Jia X 2019 Evaluating robustness of deep learning based lung nodule classification *AAPM Annual Meeting Science Council Session TU-HI-SAN2-11*
Tseng H H, Luo Y, Cui S, Chien J T, Ten Haken R K and Naqa I E 2017 Deep reinforcement learning for automated radiation adaptation in lung cancer *Med. Phys.* **44** 6690–705
Ulyanov D, Vedaldi A and Lempitsky V 2016 Instance normalization: The missing ingredient for fast stylization (arXiv:1607.08022)
Veit A, Wilber M J and Belongie S 2016 Residual networks behave like ensembles of relatively shallow networks *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 550–8
Venkatesh G, Nurvitadhi E and Marr D 2017 Accelerating deep convolutional networks using low-precision and sparsity *2017 IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (Piscataway, NJ: IEEE) pp 2861–5
Vincent P, Larochelle H, Bengio Y and Manzagol P-A 2008 Extracting and composing robust features with denoising autoencoders *Proc. of the 25th Int. Conf. on Machine Learning* (New York: ACM) pp 1096–103
Vincent P, Larochelle H, Lajoie I, Bengio Y and Manzagol P-A 2010 Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion *J. Mach. Learn. Res.* **11** 3371–408
Wan L, Zeiler M, Zhang S, Le Cun Y and Fergus R 2013 Regularization of neural networks using dropconnect *Int. Conf. on Machine Learning* pp 1058–66
Wang X, Peng Y, Lu L, Lu Z, Bagheri M and Summers R M 2017 Chestx-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition* pp 2097–106
Watkins C J and Dayan P 1992 Q-learning *Mach. Learn.* **8** 279–92
Weinan E 2017 A proposal on machine learning via dynamical systems *J. Commun. Math. Stat.* **5** 1–11
Weinan E, Han J and Li Q 2019 A mean-field optimal control formulation of deep learning *J. Res. Math. Sci.* **6** 10
Wen W, Wu C, Wang Y, Chen Y and Li H 2016 Learning structured sparsity in deep neural networks *Advances in Neural Information Processing Systems* (New York: Curran Associates, Inc.) pp 2074–82
Wu D, Kim K, El Fakhri G and Li Q 2017 Iterative low-dose CT reconstruction with priors trained by artificial neural network *IEEE Trans. Med. Imaging* **36** 2479–86
Wu J *et al* 2018 A deep Boltzmann machine-driven level set method for heart motion tracking using cine MRI images *Med. Image Anal.* **47** 68–80
Wu Y and He K 2018 Group normalization *Proc. of the European Conf. on Computer Vision* pp 3–19
Xie S, Yang C, Zhang Z and Li H 2018 Scatter artifacts removal using learning-based method for CBCT in IGRT system *IEEE Access* **6** 78031–7

Xing L, Krupinski E A and Cai J 2018 Artificial intelligence will soon change the landscape of medical physics research and practice *Med. Phys.* **45** 1791–3

Xu J, Xiang L, Liu Q, Gilmore H, Wu J, Tang J and Madabhushi A 2016 Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images *IEEE Trans. Med. Imaging* **35** 119–30

Yarotsky D 2018 Optimal approximation of continuous functions by very deep ReLU networks (arXiv:1802.03620)

Yuan X, He P, Zhu Q and Li X 2019 Adversarial examples: Attacks and defenses for deep learning *IEEE Trans. Neural Netw. Learn. Syst.* **30** 2805–24

Zaremba W, Sutskever I and Vinyals O 2014 Recurrent neural network regularization (arXiv:1409.2329)

Zeiler M D and Fergus R 2014 Visualizing and understanding convolutional networks *European Conf. on Computer Vision* (Berlin: Springer) pp 818–33

Zhang C, Bengio S, Hardt M, Recht B and Vinyals O 2016 Understanding deep learning requires rethinking generalization (arXiv:1611.03530)

Zhang F, Song Y, Cai W, Lee M-Z, Zhou Y, Huang H, Shan S, Fulham M J and Feng D D 2014 Lung nodule classification with multilevel patch-based context analysis *IEEE Trans. Biomed. Eng.* **61** 1155–66

Zhang Q, Nian Wu Y and Zhu S-C 2018 Interpretable convolutional neural networks *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition* pp 8827–36

Zhang Q-S and Zhu S-C 2018a Visual interpretability for deep learning: a survey *Frontiers Inf. Technol. Electron. Eng.* **19** 27–39

Zhang Y and Yu H 2018b Convolutional neural network based metal artifact reduction in x-ray computed tomography *IEEE Trans. Med. Imaging* **37** 1370–81

Zhang Y, Yang L, Chen J, Fredericksen M, Hughes D P and Chen D Z 2017 Deep adversarial networks for biomedical image segmentation utilizing unannotated images *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention* (Berlin: Springer) pp 408–16

Zhen X, Chen J, Zhong Z, Hrycushko B, Zhou L, Jiang S, Albuquerque K and Gu X 2017 Deep convolutional neural network with transfer learning for rectum toxicity prediction in cervical cancer radiotherapy: a feasibility study *Phys. Med. Biol.* **62** 8246

Zhou Y, Wang Y, Tang P, Bai S, Shen W, Fishman E K and Yuille A L 2018 Semi-supervised multi-organ segmentation via deep multi-planar co-training (arXiv:1804.02586)

Zhu B, Liu J Z, Cauley S F, Rosen B R and Rosen M S 2018 Image reconstruction by domain-transform manifold learning *Nature* **555** 487

Zhu J-Y, Park T, Isola P and Efros A A 2017 Unpaired image-to-image translation using cycle-consistent adversarial networks *Proc. of the IEEE Int. Conf. on Computer Vision* pp 2223–32