Overview of Machine Learning: Part 2 Deep Learning for Medical Image Analysis

Article *in* Neuroimaging Clinics of North America · October 2020

DOI: 10.1016/j.nic.2020.06.003

5 authors, including:

William Trung Le
University of Montreal Hospital Research Centre
13 PUBLICATIONS **38** CITATIONS
SEE PROFILE

Farhad Maleki
The University of Calgary
37 PUBLICATIONS **298** CITATIONS
SEE PROFILE

Francisco Perdigón Romero
Ericsson Canada
33 PUBLICATIONS **146** CITATIONS
SEE PROFILE

Reza Forghani
McGill University Health Centre
127 PUBLICATIONS **3,617** CITATIONS
SEE PROFILE

Overview of Machine Learning: Part 2
Deep Learning for Medical Image Analysis

William Trung Le, BSc[a,b,1], Farhad Maleki, PhD[c,1], Francisco Perdigón Romero, MSc[a], Reza Forghani, MD, PhD[c,d,e,f], Samuel Kadoury, PhD[a,b,*]

KEYWORDS

• Deep learning • Medical imaging • Health care • Convolutional neural network • Image registration • Image synthesis • Treatment planning • Radiology

KEY POINTS

- Radiological imaging data for H&N contains a wealth of information suitable for feature extraction using deep learning methods to characterize various pathologies.
- Convolutional neural networks (CNN) have recently become highly effective in multiple medical imaging tasks including anatomical classification, segmentation and registration, as well as disease progress prediction, and image reconstruction.
- Various experimental and ethical considerations still need to be addressed to ensure successful deployment of deep learning models in clinical settings.

INTRODUCTION

Innovations in deep learning continue to result in breakthroughs across multiple fields, including computer games,[1,2] autonomous driving,[3] predictive health care,[4,5] and natural language processing.[6,7] These breakthroughs can be attributed to improvements in graphical processing units technology allowing faster model training[8]; the emergence of accessible development frameworks such as TensorFlow,[9] Keras,[10] PyTorch,[11] MXNet,[12] and Caffe[13]; and the increasing volume of data available for model training.[14] Artificial neural networks (ANN) were originally modeled after the biological neurons and their synaptic connections in the brain and can be considered the building blocks for deep learning architectures.[15–17] With ANNs, the flow of synaptic pulses can be seen as a hierarchical feature extractor.[8] As information is propagated through the layers of neurons, the internal representation captures increasingly more abstract and complex relationships. This is the motivation for the development of "deep" networks, with many layers for processing large and complex data.

Preprint submitted to *Neuroimaging Clinics* February 27, 2020.
Funding Information: Dr Samuel Kadoury is a Canada Research Chair (CRC) in Medical Imaging and Assisted Intervention.
[a] Polytechnique Montreal, PO Box 6079, succ. Centre-ville, Montreal, Quebec H3C 3A7, Canada; [b] CHUM Research Center, 900 St Denis Street, Montreal, Quebec H2X 0A9, Canada; [c] Augmented Intelligence & Precision Health Laboratory (AIPHL), Department of Radiology and Research Institute of the McGill University Health Centre, 1001 Decarie Boulevard, Montreal, Quebec H4A 3J1, Canada; [d] Segal Cancer Centre, Lady Davis Institute for Medical Research, Jewish General Hospital, 3755 Cote Ste-Catherine Road, Montreal, Quebec H3T 1E2, Canada; [e] Gerald Bronfman Department of Oncology, McGill University, Montreal, Quebec, Canada; [f] Department of Otolaryngology - Head and Neck Surgery, McGill University, Montreal, Quebec, Canada
[1] Co-first author.
* Corresponding author.
E-mail address: samuel.kadoury@polymtl.ca

Neuroimag Clin N Am 30 (2020) 417–431
https://doi.org/10.1016/j.nic.2020.06.003

This was shown in 2012 during the ImageNet Large-Scale Visual Recognition Challenge,[18] previously dominated by classic computer vision methods. The AlexNet model used a deep convolutional neural network (CNN), leading to a substantial reduction in error rate compared with the classic computer vision methods.[19] This success revolutionized image processing research and applications.

In medical imaging, computed tomography (CT), MR imaging, and ultrasound images are prevalent. These imaging modalities can be used for diagnosis, treatment planning, disease monitoring, and evaluation of response to therapy. Traditionally, radiologists perform these analyses, relying on human-discernible visual features. However, this manual approach requires years of specialized training, and the inherent complexity of these types of images can make certain manual tasks and the subjective process laborious, time-consuming, and prone to interobserver variability. Furthermore, increasing evidence suggests that the complex quantitative information on medical images is underutilized using current approaches. Deep learning thus offers multiple benefits compared with previous techniques: the ability to perform medical image analysis for identification of high-order complex features, performing different classification tasks or predictive modeling, and accelerating image processing tasks. These advantages will facilitate deployment in the clinical workflow. This article provides the fundamental background required to understand and develop deep learning models used for medical image processing. The authors cover the main deep learning architectures such as feedforward and recurrent neural networks and their variants and also provide use cases of such applications in medical image processing.

DEEP LEARNING ARCHITECTURES

Feedforward Neural Networks

In a feedforward network, the information flow between computational units in the network can be represented as a directed acyclic graph, where the information flows from the inputs to outputs. In other words, there is no path in which the output of a computational unit is fed back into itself. Typical examples of feedforward networks include multilayer perceptrons (MLPs), convolutional neural networks, autoencoders, and generative adversarial networks (GANs).

Multilayer Perceptrons

MLPs are the quintessential feedforward networks. The term is sometimes used interchangeably with feedforward networks. An MLP, as depicted in **Fig. 1**, consists of several layers, each with one or more artificial neurons. Each neuron accepts one or more inputs. First, each input is multiplied by a weight. Then the summation of all weighted inputs and a bias value are calculated. Next, an activation function is applied to the summation, and the output of the activation function is considered as the neuron's output. The outputs of the neurons from each layer are used as inputs for the neurons in the next layer of the network.

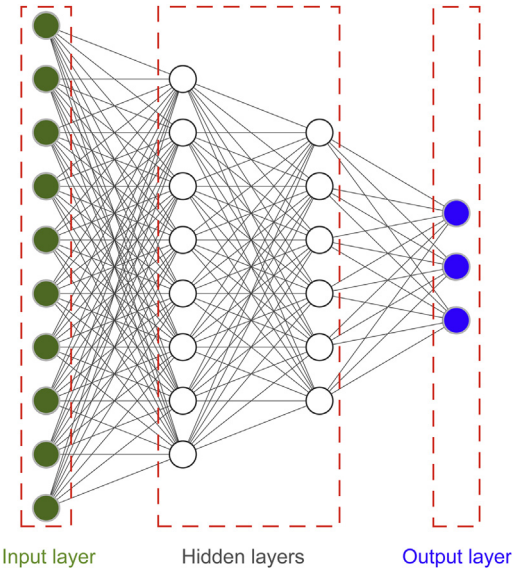

Fig. 1. A multilayer perceptron. Each artificial neuron is represented by a circle in white or blue color. The network consists of an input layer (in *green*), an output layer (in *blue*), and 2 hidden layers (in *white*). Each edge represents a weight for the network. These weights define the network and are determined through a training process.

Network weights, often referred to as network parameters, define the output of the network for a given input. These weights are often initialized randomly. Then during an iterative process, referred to as the training process (see Section 3), the optimal weight assignment for the network is sought. During the training process, a loss function is used to measure the optimality of a given weight assignment for the network.

Activation functions are used to introduce nonlinearity to the network and make it possible for the network to learn complex nonlinear functions. Rectified linear unit (ReLU) and its variants, Sigmoid, and Tanh are commonly used in neural networks.[8,20,21] ReLU and its variants are commonly used for neurons in hidden layers. Sigmoid is commonly used in the output layers of

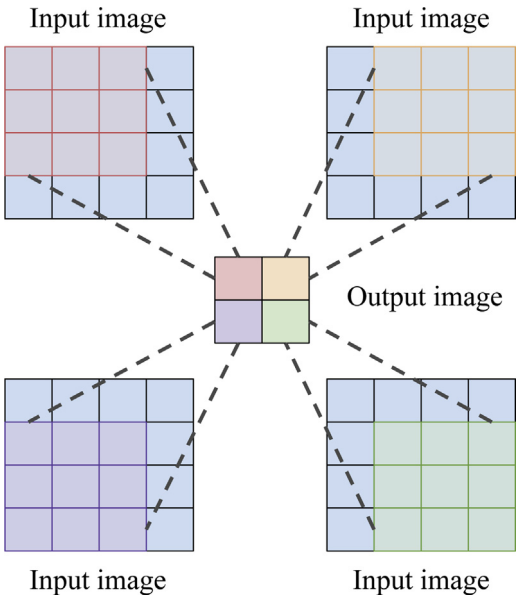

Fig. 2. A 3 × 3 convolution. A convolution slides over the input and generates an output. This allows the same local feature to be extracted anywhere within the input, which means fewer parameters to be learned.

networks designed for classification. In multiclass classification, softmax function (also known as softargmax) is applied to the output of the network to generate a class probability distribution.

Convolutional neural networks

Training MLP for tasks with a large number of inputs requires a substantial amount of training examples that might not always be available. For example, using a 1000 by 1000 grayscale image (width: 1000 and height: 1000) leads to an input dimension of 1,000,000. Building an MLP for processing such images is impractical. In addition, there are high spatial dependencies between neighboring pixels in an image. CNNs use these spatial dependencies and build on MLPs by replacing neurons with convolutions (Fig. 2) and pooling operators.[8] These operators are applied to a neighborhood rather than the whole input. This approach has 2 primary advantages: sparse connectivity and parameter sharing.[8] Using convolutional kernels reduces the number of parameters compared with using fully connected layers in MLPs. This reduces the memory requirements and makes it possible to achieve a better performance. Also, the kernel is repeatedly applied to each region of the image: this allows its parameters to be shared across the image. This is in contrast to MLPs where a weight is only associated with an input from the previous layer.

A convolution operator—also referred to as a convolutional kernel or filter—computes a weighted linear combination of its parameters and the corresponding inputs from the region where the filter is applied.

A CNN consists of several convolution or pooling layers (Fig. 3). Each convolution layer includes several kernels. The output of each layer can then be used as the input for the next layer. Because convolutions act as local feature extractors, pooling operations are necessary to allow global feature combinations. A pooling operation summarizes the corresponding inputs from the region where the operator is applied. For example, the output of a max-pooling operator is the maximum of the corresponding values where the operator is applied, and this can be used to reduce the dimensions of the intermediate representations. Pooling operators can reduce the memory requirement and the sensitivity to small translations in the input, as well as increase the effective receptive field for future convolution operators.

Autoencoders

Autoencoders are a family of feedforward networks commonly used for providing a low-dimensional representation of data.[8] An advantage of autoencoders is that they can be trained in an unsupervised manner without labeled data. An autoencoder has 2 components: an encoder and a decoder. For a given input x, the encoder tries to provide a low-dimensional representation y from x. The decoder, on the other hand, tries to reconstruct the input x using y, that is, the low-dimensional representation generated by the encoder. The intuition behind autoencoders is that if y is an accurate representation of x, it should hold enough information to reconstruct x.

The autoencoder can be trained end-to-end using the backpropagation algorithm (see Section 3). Fig. 4 illustrates a typical autoencoder, which is also referred to as an undercomplete autoencoder. Several other variants of autoencoders exist: contractive autoencoders,[22,23] denoising autoencoders,[24] and variational autoencoders.[25]

Sensitivity to small variations and noise in the training data is one of the challenges an autoencoder might encounter. To address this challenge, contractive autoencoders add a specific component to the loss function to penalize network weights that lead to sensitivity to small variation in the data. In mathematical terms, this component corresponds to the Frobenius norm of the Jacobian matrix of the activations in the encoder. In denoising autoencoders, a different approach is used to address these challenges. During the training process, inputs are corrupted by

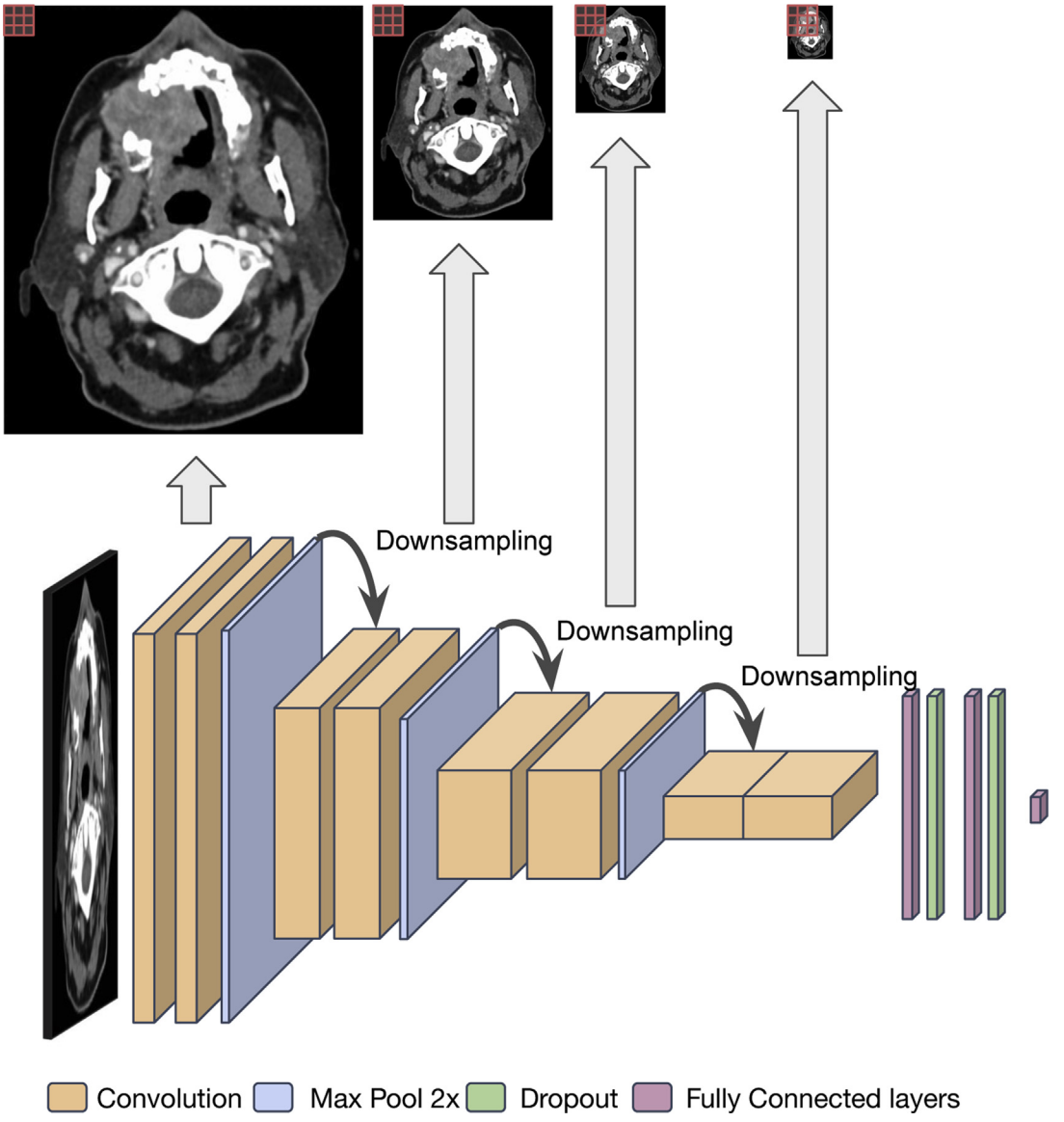

Fig. 3. A CNN-based network for image classification. Successive convolution and downsampling through max pooling reduce the input size while increasing the kernel's receptive field. The original images at the top were used to represent the increase in the receptive field while being a reference for the dimensionality reduction of the intermediate representations as the result of applying max pooling. For an image classification task, the CNN component is followed by an MLP to convert the deep features generated by CNN to a class probability distribution.

introducing small random noises. Then the network is trained to learn the actual input before the corruption. Corrupted inputs can be made through a stochastic process such as adding Gaussian noise or masking noise.[26]

Generative adversarial networks

GANs are composed of 2 networks: a generator and a discriminator, as shown in Fig. 5.[27] These networks are trained in parallel with opposite goals. The generator synthesizes data from scratch, often using random inputs. The discriminator on the other hand receives either a ground truth from the target domain or a synthetic output, which is produced by the generator, and tries to distinguish the true outputs from the synthetic ones.

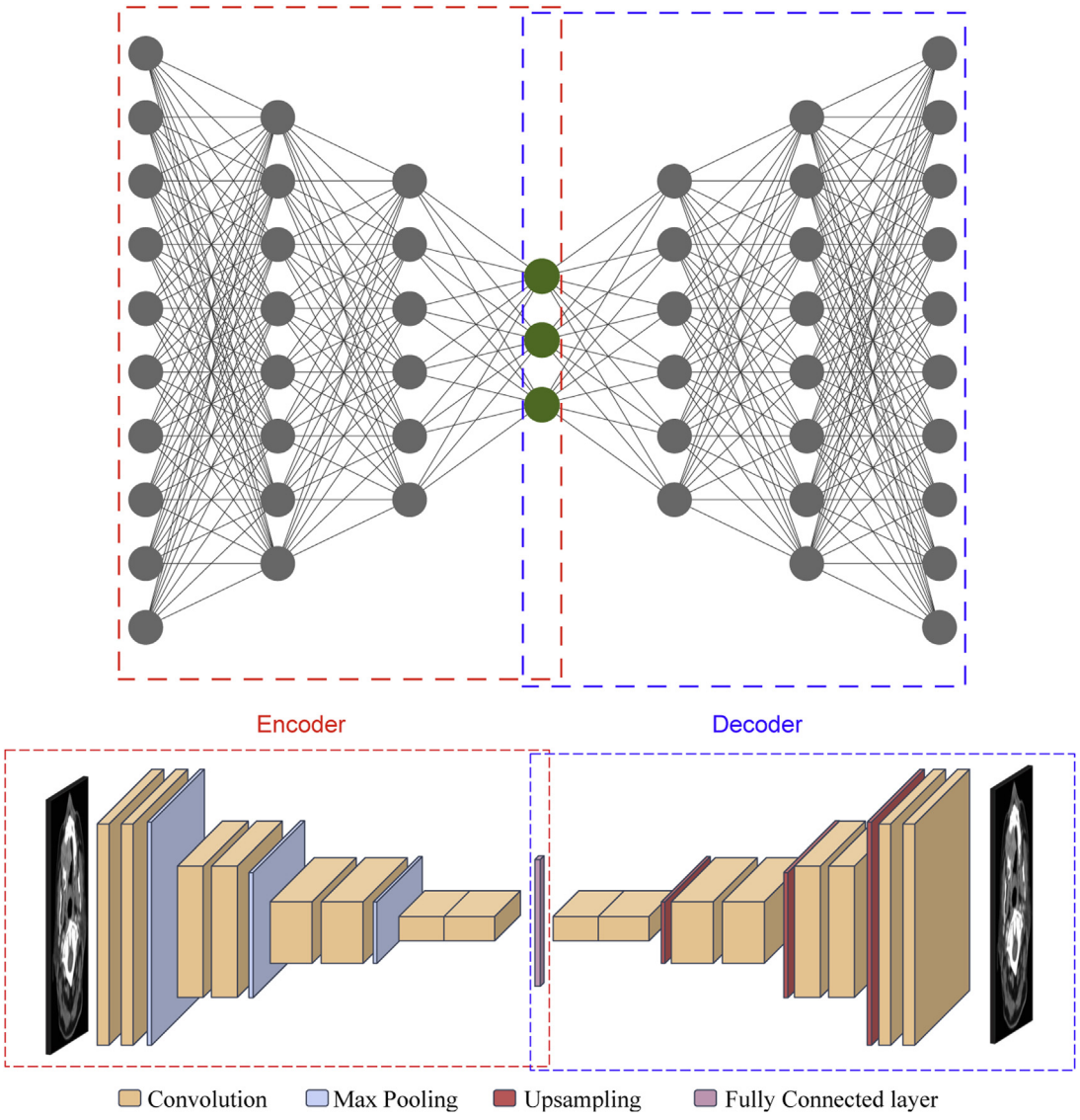

Fig. 4. Typical autoencoders based on MLP (*top*) and CNN (*bottom*). The network consists of 2 components: the encoder and the decoder. The encoder, represented within the red box, transforms the input to a lower dimensional space. For the MLP-based autoencoder, the low-dimensional representation is the output of the 3 neurons (*green*). For the CNN-based autoencoder the representation is the output of the convolution layer (*green*). The decoder's task is to reconstruct the input using the low-dimensional representations generated by the encoder.

During the training process, generators incorporate feedback from the discriminator to improve its performance, where the performance is measured based on the percentage of synthesized data classified by the discriminator as being from the target domain. Training a GAN model is a balancing act between the generator and the discriminator performance. An overperformance by either side leads to an overfitting situation where the synthesized output becomes meaningless. Convergence thus occurs when a Nash equilibrium is attained: real and synthetic data become indistinguishable.

In medical imaging, GANs are often used for image registration[28–30] or image synthesis tasks.[31–35] In registration tasks, the generator outputs a transformation, and the discriminator must distinguish the warped image from the alignment target image. Synthesis tasks often revolve around generating a new synthetic view of the input image, such as MR imaging to CT synthesis[35] or generation of 3-dimensional (3D) organ volumes from 2D single-slice scans.[36,37] GAN may also be used for data augmentation, as discussed in Section 3 (see **Fig. 5**).

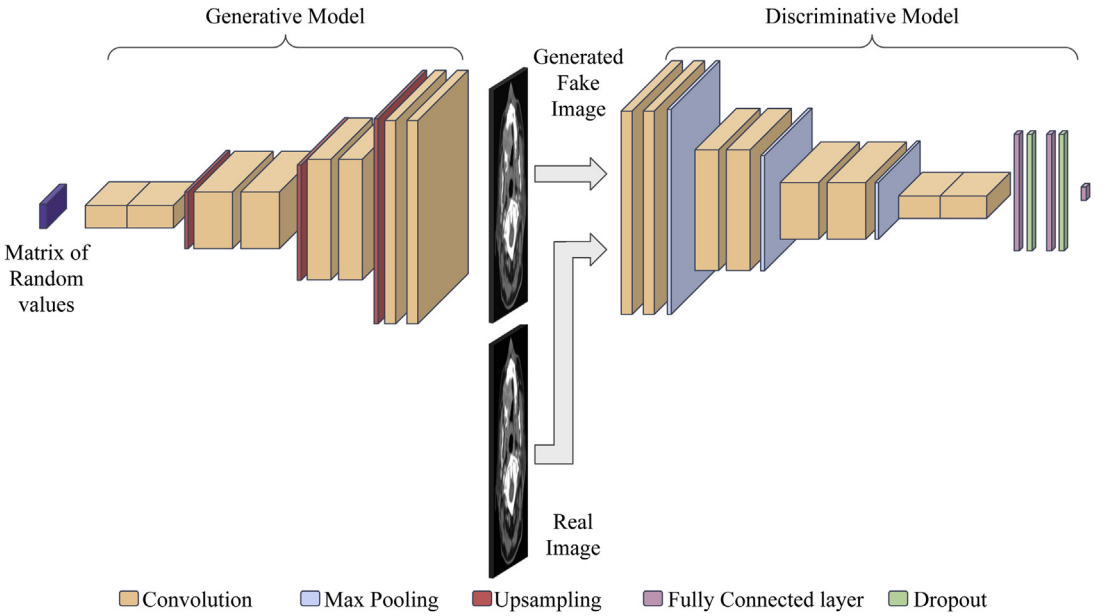

Fig. 5. A typical GAN for image synthesis. The generator synthesizes data aiming to deceive the discriminator. The discriminator, on the other hand, tries to classify the synthesized data and data from the target domain accurately.

Recurrent Neural Networks

Sequential data are a common data type in scientific and engineering applications. Elements of a sequence have a temporal relationship. Also, these sequences might have different lengths. Because of these characteristics, MLPs fall short in processing sequential data such as text, audio, and video. Recurrent neural networks (RNNs) have been designed to work with such data types.

Fig. 6 (panel A) illustrates a schematic view of an RNN. Given a sequence $S = x_1,\ldots,x_n$, at time step t, x_t is fed to the network. The network preserves a summary of previous data elements, that is, $x_1,\ldots,x_{t-1}$. This summary is often called a latent variable or hidden state and denoted by h_{t-1}. At the time step t, the network uses h_{t-1} and x_t for generating an output y_t as well as a new hidden state h_t. The initial value of the hidden state (h_0) is often set to be a zero vector.

Long short-term memory (LSTM)[38] and gated recurrent units (GRU)[39,40] are 2 commonly used RNN architectures. These architectures use gating mechanisms to preserve longer dependencies in input sequences. LSTM uses an internal state (cell) to preserve information from the key elements of the input sequence. LSTM uses an input gate, an output gate, and a forget gate to control the flow of information in the network, that is, to update the cell and hidden states. GRU is a

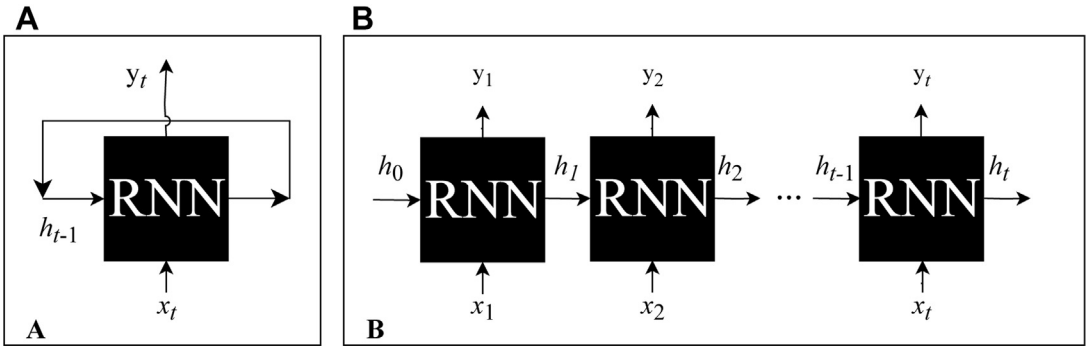

Fig. 6. A schematic view of a typical RNN. The RNN has been visualized as a black box. At each time step t, the RNN accepts a hidden state (h_{t-1}) from the previous time steps ($x_1,\ldots,x_{t-1}$) as well as an input. The RNN then generates an output (y_t) and a new hidden state (h_t). Panel B shows the computational logic of the same network. The initial hidden state (h_0) is often set to be a zero vector.

variation of LSTM. It is a simpler architecture with no cell and with only reset and update gates. GRU has less parameters due to its lack of output gate and may achieve higher performance when working with small datasets.[40] In general, LSTMs have more capacity and are more powerful in comparison to GRUs.[41]

TRAINING AND VALIDATION OF DEEP LEARNING MODELS

Training Deep Learning Models

After data acquisition and network architecture design/selection, the network should be trained to determine the weights that lead to an optimal or near-optimal solution for the task at hand. First, network weights are initialized through a stochastic process. Xavier initialization[42] and Kaiming initialization[43] are some commonly used network initialization methods. The training of deep learning networks is carried out through an optimization process, where the objective is to minimize the error obtained through a cost function. This function—also referred to as loss function—represents the difference between the network outputs from the desired outputs. To correct the error, the backpropagation algorithm is used to modify network weights depending on their contribution to the calculated loss. This process is repeated several times until a stopping criterion is achieved.[17]

The loss function may vary depending on the task at hand. For regression tasks, where the model outputs are continuous values, the most commonly used loss functions are mean squared error and mean absolute error, also referred to as L1 loss. For classification tasks, where the model output is a categorical variable, the most common loss functions are cross-entropy and hinge loss. For segmentation tasks, the most commonly used loss functions are pixel accuracy, intersection over union—also referred to as Jaccard Index—and Dice coefficient.

Backpropagation

The learning process for a neural network consists of 2 phases: a stimulus presentation phase followed by a model adaptation phase. In the first step, an input is fed to and processed by the network, resulting in a prediction output that is compared with the expected output using a loss function. This difference, also called the error or loss value, can then be used to numerically update the parameters (weights and biases) of the network in order to minimize the loss value. The backpropagation algorithm calculates the gradient of the loss function with respect to network parameters through the chain rule, which is a mathematical formula for calculating the derivatives of a composite function in an efficient way. After calculating the gradients, the network parameters can then be updated incrementally to minimize the future loss with the given input.[8]

Considerations for Training Deep Learning Models

Bias versus variance

Differentiating training error, generalization error, and Bayes error are crucial for developing and deploying deep learning models successfully. Training error refers to the error made by a model when applied to the data used during model training. Generalization error, on the other hand, is the error made by a model when applied to previously unseen data. These data must not be used for the training or fine-tuning steps nor as well as for the design or selection of the model architecture. Finally, the Bayes error—also known as irreducible error—is the lowest achievable error for a task using a given dataset. Most often it is not possible to mathematically calculate Bayes error. Therefore, an estimate for it is used: for example, the predictions made by a council of experts can be used for estimating Bayes error. It should be noted that unlike training and generalization errors, Bayes error is independent of a given model; rather, it is defined for a task based on a given dataset. Simply put, Bayes error is an indicator of the best possible algorithm performance for a task given a specific dataset.

When developing deep learning models, the available data are often divided into 3 subsets: training, validation, and test data. Training data are used to train the model through an optimization process. Test data are used to estimate the generalization error for the trained model. Validation data are used to tune model hyperparameters, for example, number of layers in an MLP, and also to avoid overfitting to the training data.

Overfitting and underfitting are 2 fundamental concepts in machine learning. Overfitting happens when a model achieves a small training error but relatively large generalization error, that is, there is a falsely optimistic performance of the model. Underfitting, on the other hand, happens when a model achieves a large training error relative to the Bayes error.

Overfitting and underfitting can also be explained by the bias-variance trade-off. Bias error refers to the error made due to erroneous or oversimplifying assumptions made when building a model. Errors due to a high variance happen when the model becomes too sensitive to small

fluctuations or noise in the training data. Thus, a model with high bias leads to underfitting, whereas a model with high variance leads to overfitting. Using linear regression to model a highly nonlinear relationship between dependent and independent variables leads to high bias error and such a model will underfit. On the other hand, using a deep MLP for capturing a linear relationship leads to high variance error and such a model will overfit. Therefore, finding a bias-variance trade-off is essential in machine learning applications.

Among common approaches to deal with overfitting are using larger training datasets and controlling model complexity through the proper use of regularization, dropout,[44] and early stopping.[8] Also, changing network architecture is used to achieve a trade-off between bias and variance. Underfitting usually happens due to insufficient training or model capacity. A more detailed discussion of machine learning algorithm validation is provided in a separate review article in this issue.

Data augmentation

One approach to deal with overfitting is to use large-scale datasets for training deep learning models. However, this is often not an option in the biomedical domain, as it requires resources that might not be accessible in a timely and cost-effective way; because of legal, ethical, or privacy considerations; or because the disease being studied is rare or the biological process of interest (eg, a complex molecular profile) may only be available on a small set of patients. Data augmentation refers to computational methods used to generate new samples from existing ones. These new samples, together with the original samples, can then be used for training a machine learning model. These methods have been extensively used in image processing tasks.[45] Image augmentation methods include simple transformations such as cropping, affine transformations, color space transformation, random erasing, noise injection, elastic deformation,[46] as well as more complex approaches such as GAN-based data augmentation methods.[32]

Data imbalance

A common challenge when working with medical data is data imbalance, also known as class imbalance. This imbalance happens when a disproportionate ratio of observations exists in each class or observation category. For a dataset, the class imbalance is measured using the imbalance ratio defined as the ratio of the number of observations in the majority class to that of the minority class. The majority and minority classes are defined to be the classes with the largest and smallest number of observations, respectively.

Class imbalance, if not addressed properly, might hamper developing generalizable models. Many methods for addressing class imbalance exist.[47] Oversampling, undersampling, and data augmentation are 3 common approaches for addressing data imbalance. These techniques attempt to generate a new balanced dataset from the original imbalanced one. Oversampling introduces extra samples into classes with a smaller number of observations. For each class, these additional samples are generated through resampling with replacement from the members of the class itself. In undersampling, on the other hand, samples from classes with a larger number of observations are discarded to reduce the imbalance. A shortcoming of these resampling techniques is that they change the data distribution, and the model trained with the resampled data might not achieve the optimal results. In addition, undersampling might lead to overfitting for datasets with high imbalance ratios. Data augmentation can also be used to introduce more samples to the class(es) with a smaller number of samples. Another method for dealing with data imbalance is defining a loss function to be in favor of classes with a small number of observations, that is, to impose larger penalties for the error made using samples from small classes when training the model.

Transfer learning

Transfer learning refers to using knowledge gained from a source domain to solve problems in a target domain that is related to, but not the same as, the source domain. Because of the resource-intensive nature of deep learning, transfer learning has been widely used.[5,48–51]

In the presence of large public datasets, such as ImageNet,[18] an initial model is trained on these datasets. Then the pretrained model is fine-tuned on a target domain, where the required resources for training the model from scratch might not be available. A large number of pretrained models exist for image processing tasks. Therefore, transfer learning has been widely used for medical image analysis, where acquiring large datasets is often impractical.

For classification tasks, the number of classes in the target domain might differ from that of the pretrained model. Therefore, the classifier component of the pretrained model is replaced with a new classifier that suits the target domain. This new resulting model is then trained using the available data from the target domain.

Deep Learning for Medical Imaging

Medical imaging with a broad range of domains, modalities, and tasks is a field rich in potential for deep learning applications. In this section, the authors provide use cases of applications of deep learning in medical imaging.

Classification

Classification is a process through which the category of each observation must be predicted among a set of predetermined categories. Detecting the presence of a disease, classifying a lesion, and predicting the response to a line of treatment are examples of classification tasks conducted using medical imaging. The resulting models can provide added diagnostic or prognostic benefits in a noninvasive manner. In this section, the authors cover some classification use cases in medical imaging.

- To predict transarterial chemoembolization response level in hepatocellular carcinoma, Peng and colleagues[49] used a ResNet model[52] pretrained on the ImageNet dataset[18] and fine-tuned on their internally collected 789 patients' CTs and achieved 85.1% and 82.8% prediction accuracies.
- For tracking disability progression of patients with multiple sclerosis after 1 year, Tousignant and colleagues[50] proposed a CNN approach based on an Inception model[53] trained on a proprietary brain MR imaging dataset with 5 different sequences as inputs. They achieved a 66.0% receiver operating characteristic area under the curve (AUC) score, which was improved to 70.1% with the introduction of 2 extra lesion mask sequences as input images.
- Using transfer learning, Zhou and colleagues[51] built 3 different binary classifiers for distinguishing benign and malignant tumors from CT images. Their first model, which was called the image-level model, was built based on an Inception model[53] adopted for binary classification of CT slices. In their second and third models, which were called patient-level models, they first extracted a feature vector for each slice of the CT. These features were extracted from a trained version of the image-level model. The second model concatenated these feature vectors and fed them into a max pooling layer, followed by a shallow MLP-based binary classifier. The third model used a GRU-based binary classifier that accepted these feature vectors as sequential inputs. They reported that the patient-level models achieve better results compared with the image-level models.
- To predict breast cancer risk from mammograms, McKinney and colleagues[5] developed an ensemble of 3 deep learning systems, namely lesion model, breast model, and case model. The average of the predicted risk scores was then used as the ensemble prediction. For each patient, they used craniocaudal and mediolateral oblique views of the left and right breasts. In the lesion model, they used RetinaNet[54] to detect regions of interest (ROIs) and their corresponding confidence scores. Then 10 ROIs with the highest confidence score as well as their corresponding regions from contralateral breast were selected. Each ROI and its corresponding region were fed to a MobileNetV2[55] to predict a cancer risk score. Ten calculated risk scores were then combined to calculate a case-level score. The Breast model used a ResNet[56] module as the feature extractor. Then per breast features were fed to another residual network[52] to make the case-level prediction. Case model also used a ResNet feature extractor.[52] The extracted features for each view of the left and right breast were concatenated and fed to an MLP for making the case-level predictions. They trained and evaluated the model using 2 large datasets and reported reductions in false-positive rates of 9.4% and 2.7% and reduction in false-negative rates of 5.7% and 1.2% for these datasets compared with predictions made by experts.

Segmentation

In semantic segmentation, the goal is to delineate the boundaries of anatomic or pathologic structures of interest. The results are used to guide treatment target planning, to aid multimodal image registration, or to aid operative navigation systems. U-Net[57] and its variants have been widely used for segmentation in the medical imaging field. **Fig. 7** illustrates U-Net architecture.

- Hashemi and colleagues[58] developed a 3D fully convolutional architecture based on DenseNet blocks and an asymmetric loss for infant brain MR imaging white matter, gray matter, and cerebrospinal fluid segmentation. By having the model learn 2 of the 3 labels in parallel and taking the complement to generate the contours for the third (gray matter), they achieved a 96% Dice score.
- Sun and colleagues[59] demonstrated that hippocampus segmentation can be improved by using a V-Net[60] coupled with an auxiliary loss

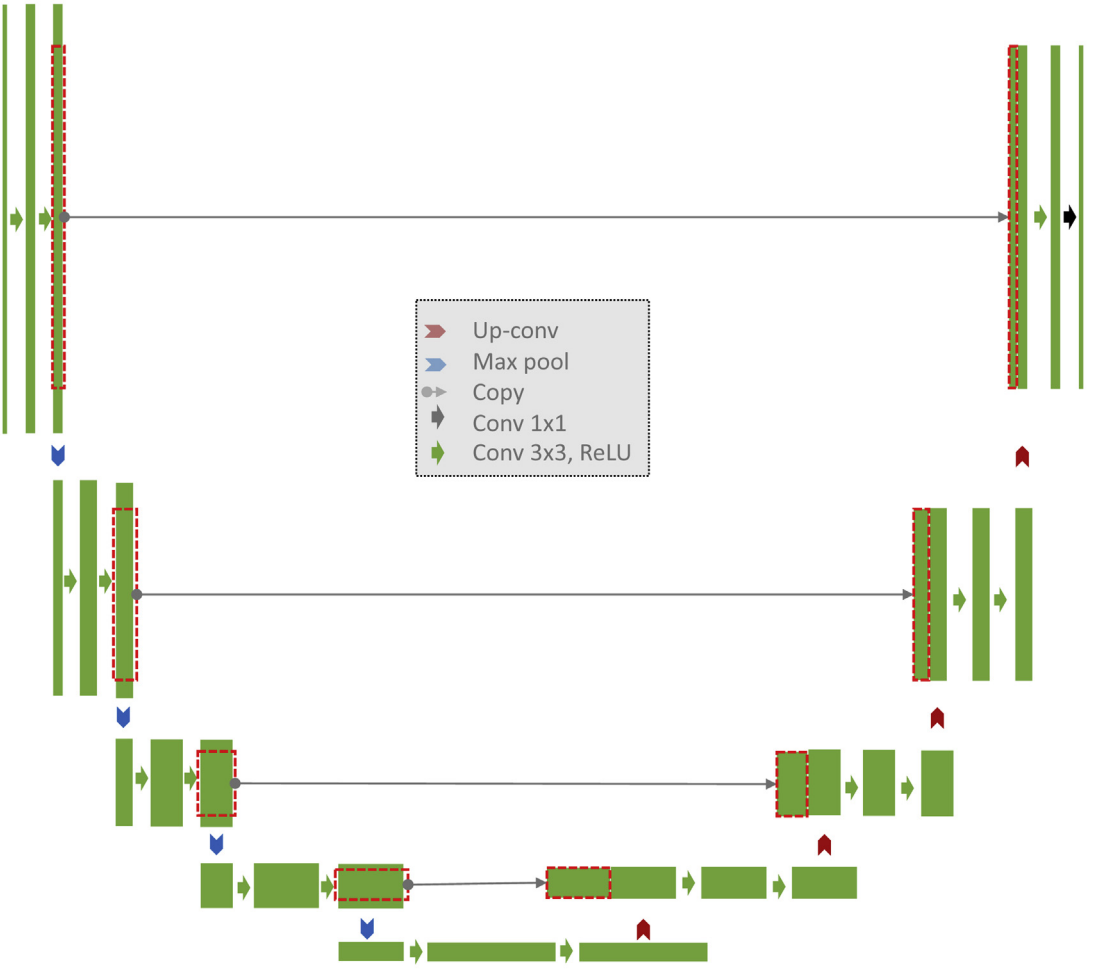

Fig. 7. U-Net architecture[57] is made of a contracting path followed by an expansive path. In the contracting path, a sequence of convolution and pooling operators produces a dense representation of the input image. In the expansive path, a sequence of convolution and upsampling operators is used to generate the segmentation mask. To be able to use both high-level and low-level features, U-Net used a crop and copy operator.

for brain state classification. Hippocampal size is a useful feature for monitoring Alzheimer disease progression. They showed that taking the embedded representation of the network and the output mask as input to a classification subnetwork as an additional constraint, segmentation results improved to 91.6% Dice score.

- Yin and colleagues[61] developed a novel method to segment kidney ultrasound images using a combination of 2 deep learning models. The first model computes an organ boundary distance map using a pretrained version of VGG model[62] feature extractor modified with atrous convolutions.[63] This distance map has the advantage of reducing the pixel-wise organ to background class imbalance. The predicted kidney boundary map was then fed to a pretrained DeepLab[64] segmentation network. Yin and colleagues reported a 93% Dice score on their privately acquired dataset.

Registration

Image registration is the task of aligning 2 images together. Image registration is required when a one-to-one mapping between 2 acquired images is not possible by simply overlaying one image on the other. This is the case when these images are from different scan times, causing a shift in the anatomic structures due to movement. These misalignments might imply different visual features of the same area. In this section, the authors

provide examples of deep learning applications for image registration.

- de Vos and colleagues[65] introduced an unsupervised method that takes a pair of images through a CNN and outputs affine or B-spline deformation parameters. It shows comparable performance to conventional methods for cardiac cine MR imaging and chest CT registration while being substantially faster to compute.
- Balakrishnan and colleagues[66] introduced the VoxelMorph method, which instead outputs a deformation field via a U-Net model, outperforming the state-of-the-art results on brain MR imaging registration tasks.
- Shen and colleagues[67] developed a 2-step ensemble model combining both the affine and the deformable transformations. Their U-Net generated a velocity field, from which a deformation field was computed. This has the advantage of ensuring better fluid mechanics properties and smoother deformations.
- Eppenhof and colleagues[68] devised a method to train the U-Net–based model on pulmonary CT by starting with lower resolution images and progressively increasing them. To do so, they restricted the depth of their model and increased it over time, which allows the model to learn coarse-to-fine grained deformations.

Reconstruction, Synthesis, and Denoising

In certain medical applications, converting images between different modalities might be necessary due to better resolution, contrast to visualize clinically relevant details, or reducing radiation doses. Therefore, a model should synthesize an instance of anatomic structures in the target modality using another modality as input. Paired multimodal datasets are often rare in the medical domain; this makes GANs a well-suited option for such image synthesis tasks.

- Yang and colleagues[69] used a CycleGAN[70] to synthesize CT from MR for patients with brain tumor. This method consists of enforcing the model to be able to generate modalities from either direction. In their qualitative evaluation, experts could not distinguish the real from the synthesized pairs.
- Rubin and Abulnaga[71] developed a CT to MR model based on conditional GANs, a variant where both the generator and discriminator also accept a source domain image as input. They showed that the resulting MR imaging yielded improvements in their pipeline to improve stroke lesion segmentation.
- Bass and colleagues[72] developed a novel variant of the conditional GAN with added capsule blocks[73,74] for cortical axons microscope images. They showed that the resulting images were realistic enough to be used to train a segmentation pipeline that tested successfully on real datasets.

In the medical imaging field, reconstruction refers to transforming images from a physical domain—eg, intensity variations of the electromagnetic field for MR imaging—to images that humans can understand. Generally, these reconstruction processes are computationally expensive and difficult to parallelize. Deep learning has been used to learn and enhance such transformations.[75,76] Examples include accelerating the use of deep learning to enable acceleration of acquisition of an MR imaging sequence while preserving diagnostic information or the use of deep learning for denoising medical images.[77,78] Often noisy images prevent the correct diagnosis; therefore, denoising can add a substantial value to diagnosis and treatment workflow. To achieve such a task, a model must receive an image as input and generate a denoised version of the image as the output. Therefore, fully convolutional autoencoders have been used for such a task.[77,78]

LIMITATIONS AND FUTURE DIRECTIONS

Unlike traditional machine learning models, where the performance plateaus at some point and using more data does not improve the model performance leading to an underfitting situation, the performance of deep learning models improves as the number of training examples increases. Therefore, data availability is essential for developing deep learning models. Data availability can pose a problem in the medical imaging field where training data are scarce due to the lack of infrastructure, expert annotations, or patient privacy concerns or simply due to the rarity of the phenotype/disease under study.

Transfer learning techniques have been considered to alleviate the scarcity of training data. Using large pretrained models to extract latent features has proved to be successful in many medical imaging tasks.[5,48–51] These pretrained models are trained on public datasets from domains different from the medical domain. For example, many available pretrained models have been trained on the ImageNet dataset, which is a collection of

ordinary pictures. Therefore, the extracted features from these pretrained models could potentially be improved if the pretrained models were instead trained on medical image datasets. Medical images often have different visual characteristics and a higher resolution and pixel intensity depth. Considering the availability of many small- and medium-sized public dataset of medical images—eg, datasets from the Grand Challenge in Medical Imaging and the Cancer Imaging Archive—a data aggregation attempt for building a large-scale medical imaging dataset and providing pretrained models based on this dataset may benefit the research community and facilitate the deployment of methods built on these domain-specific pretrained models in clinical setting.

Medical images come in a variety of modalities such as radiographs, ultrasound, CT, MR imaging, PET, and other nuclear medicine scans, to mention the most common ones. One may also consider digital histopathology slides as medical images. These modalities have different spatial and depth resolution. Also, using different modality-specific acquisition parameters—T1 versus T2 weighting of MR imaging scans or adding a contrast agent—provides trade-off for visibility of certain substructures. Finally, medical images are also affected by device- or manufacturer-specific differences, varying reconstruction parameters that can vary between institutions and even within the same institution, and different maneuvers or positioning that may be used for targeted evaluation depending on the specific body part. These factors pose a new dimension of complexity on top of the complex nature of regular image processing tasks. Most of the deep learning models in the literature often demonstrated state-of-the-art performance only for a subset of these parameters and often on certain anatomic structures. This could hinder the deployment of such models in the clinical setting, as these models might not generalize well when applied to data acquired with different acquisition settings or data acquired from different devices or manufacturers.

The other major limitation of deep learning is interpretability. Traditional machine learning methods offer varying levels of transparency due to the nature of the inputs: features must be carefully selected during training, and the resulting model prediction can be queried against those features. Comparatively, deep learning models are often considered as black boxes. Although there exist methods for visualization of the intermediate representation of neural networks, explainability can also represent an obstacle for unleashing the full potential and widespread adoption of deep learning algorithms in the medical domain.

SUMMARY

In this review, the authors cover the main practical concepts required for understanding and developing deep learning solutions in medical imaging and review the principal deep learning architectures, and general examples of different common categories of applications were provided. They also described the main considerations for training deep learning models. More specifically, underfitting and overfitting and the remedies for addressing these issues are explained. Data imbalance—which is a ubiquitous phenomenon in the medical domain—and how it may affect the performance of deep learning models are also discussed, and some of the solutions for alleviating the effect of data imbalance on the generalizability of deep learning models are specified. Finally, some of the limitations and challenges facing the deployment of deep learning solutions in the clinical setting are mentioned.

Deep learning has opened new horizons in medical image analysis and health care predictive modeling more broadly, and its true potential is still largely untapped. However, for successful development of algorithms that are generalizable and have the potential to be deployed in the health care setting, it is essential to understand the fundamentals as well as pitfalls of this powerful technology.

REFERENCES

1. Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go without human knowledge. Nature 2017;550(7676):354–9.
2. Vinyals O, Babuschkin I, Czarnecki WM, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature 2019;575(7782):350–4.
3. Fridman L, Brown DE, Glazer M, et al. MIT autonomous vehicle technology study: Large-scale deep learning based analysis of driver behavior and interaction with automation. arXiv preprint arXiv:1711.06976 1 (2017).
4. Lin ED, Hefner JL, Zeng X, et al. A deep learning model for pediatric patient risk stratification. Am J Manag Care 2019;25(10):e310–5.
5. McKinney SM, Sieniek M, Godbole V, et al. International evaluation of an AI system for breast cancer screening. Nature 2020;577(7788):89–94.
6. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Adv Neural Inf Process Syst 2017;5998–6008.

7. Devlin J, Chang M.-W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv: 1810.04805.
8. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge: MIT Press; 2016.
9. Abadi M, Barham P, Chen, J, et al. Tensorflow: A system for large-scale machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). Savanah (GA); 2016. p. 265–83.
10. Chollet F. Keras. 2015. Available at: https://github.com/fchollet/keras. Accessed July 29, 2020.
11. Paske A, Gross S, Massa F, et al. PyTorch: An imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 2019;32:8024–35.
12. Chen T, Li M, Li Y, et al. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems, arXiv preprint arXiv: 1512.01274.
13. JiaY, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding, arXiv preprint arXiv:1408.5093.
14. Manyika J, Chui M, Brown B, et al. Big data: the next frontier for innovation, competition, and productivity, Tech. rep. New York: McKinsey Global Institute; 2011.
15. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 1943;5(4):115–33.
16. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. Psychol Rev 1958;65(6):386–408.
17. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323(6088):533–6.
18. Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database. IEEE Conference on computer vision and Pattern Recognition, IEEE. Miami (FL), June 20, 2009.
19. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, et al, editors. Advances in neural information processing systems. Denver (CO): Curran Associates, Inc; 2012. p. 1097–105.
20. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. Fort Lauderdale (FL); 2011. p. 315–23.
21. Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. Proceedings of the 30th Annual International Conference on Machine Learning, Vol. 30. Atlanta (GA); 2013. p. 3.
22. Rifai S, Vincent P, Muller X, et al. Contractive autoencoders: explicit invariance during feature extraction. Proceedings of the 28th International Conference on International Conference on Machine Learning. Bellevue (WA): ACM; 2011. p. 833–840.
23. Rifai S, Mesnil G, Vincent P, et al. Higher order contractive auto-encoder. Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Athens (Greece): Springer; 2011. p. 645–60.
24. Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th International Conference on Machine Learning. Helsinki (Finland): ACM; 2008. p. 1096– 1103.
25. Kingma DP, Welling M. Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114.
26. Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 2010;11:3371–408.
27. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. Adv Neural Inf Process Syst 2014;27:2672–80.
28. Fan J, Cao X, Xue Z, et al. Adversarial similarity network for evaluating image alignment in deep learning based registration. In: Frangi A, Schnabel J, Davatzikos C, et al, editors. Medical image computing and computer Assisted Intervention – MICCAI 2018. Cham (Switzerland): Springer International Publishing; 2018. p. 739–46.
29. Mahapatra D, Antony B, Sedai S, et al. Deformable medical image registration using generative adversarial networks. 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). Washington, DC: IEEE; 2018. p. 1449–1453.
30. Kazeminia S, Baur C, Kuijper A, et al. GANs for medical image analysis, arXiv preprint arXiv:1809.06222.
31. Bi L, Kim J, Kumar A, et al. Synthesis of positron emission tomography (PET) images via multi-channel generative adversarial networks (GANs). In: Frangi A, Schnabel J, Davatzikos C, et al, editors. Molecular imaging, reconstruction and analysis of Moving body organs, and stroke imaging and treatment. Cham (Switzerland): Springer International Publishing; 2017. p. 43–51.
32. Yi X, Walia E, Babyn P. Generative adversarial network in medical imaging: A review. Med Image Anal 2019. https://doi.org/10.1016/j.media.2019.101552.
33. Nie D, Trullo R, Lian J, et al. Medical image synthesis with deep convolutional adversarial networks. IEEE Trans Biomed Eng 2018;65(12):2720–30.
34. Frid-Adar M, Diamant I, Klang E, et al. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing 2018;321:321–31.
35. Nie D, Trullo R, Lian J, et al. Medical image synthesis with context-aware generative adversarial

networks. In: Frangi A, Schnabel J, Davatzikos C, et al, editors. Medical image computing and computer Assisted Intervention - MICCAI 2017. Cham (Switzerland): Springer International Publishing; 2017. p. 417–25.
36. Cerrolaza JJ, Li Y, Biffi C, et al. 3D fetal skull reconstruction from 2DUS via deep conditional generative networks. In: Frangi A, Schnabel J, Davatzikos C, et al, editors. Medical image computing and computer Assisted Intervention – MICCAI 2018. Springer International Publishing; 2018. p. 383–91.
37. Biffi C, Cerrolaza JJ, Tarroni G, et al. 3d high-resolution cardiac segmentation reconstruction from 2d views using conditional variational autoencoders, arXiv preprint arXiv:1902.11000.
38. Schmidhuber J, Hochreiter S. Long short-term memory. Neural Comput 1997;9(8):1735–80.
39. Cho K, Van Merrïenboer B, Bahdanau D, et al. On the properties of neural machine translation: Encoder-decoder approaches, arXiv preprint arXiv: 1409.1259.
40. Chung J, Gulcehre C, Cho K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. NIPS 2014 Workshop on Deep Learning. Montreal (QC), December 8, 2014.
41. Weiss G, Goldberg Y, Yahav E. On the practical computational power of finite precision rnns for language recognition. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Melbourne (Australia): Association for Computational Linguistics; 2018. p. 740–5.
42. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Sardinia (Italy): PMLR; 2010. p. 249–56.
43. He K, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE International Conference on Computer Vision. Las Condes (Chile): IEEE; 2015. p. 1026–34.
44. Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15(1):1929–58.
45. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. J Big Data 2019;6(1):60.
46. Nalepa J, Marcinkiewicz M, Kawulok M. Data augmentation for braintumor segmentation: A review. Front Comput Neurosci 2019;13:83.
47. Branco P, Torgo Li, Ribeiro R. A survey of predictive modeling on imbalanced domains. ACM Comput Surv 2016;49(2):1–31.
48. Yang Q, Zhang Y, Dai W, et al. Transfer learning. Cambridge: Cambridge University Press; 2020.
49. Peng J, Kang S, Ning Z, et al. Residual convolutional neural network for predicting response of transarterial chemoembolization in hepatocellular carcinoma from CT imaging. Eur Radiol 2019; 30(1):413–24.
50. Tousignant A, Lemaître P, Precup D, et al. Prediction of disease progression in multiple sclerosis patients using deep learning analysis of MRI data. Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning, Vol. 102. London (UK): MIDL; 2019. p. 483–92.
51. Zhou L, Zhang Z, Chen Y-C, et al. A deep learning-based radiomics model for differentiating benign and malignant renal tumors. Translational Oncol 2019;12(2):292–300.
52. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas (NV): IEEE; 2016. p. 770–8.
53. Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas (NV): IEEE; 2016. p. 2818–26.
54. Lin T.-Y, Goyal P, Girshick R, et al. Focal loss for dense object detection. Proceedings of the IEEE International Conference on Computer Vision. Venice (Italy): IEEE; 2017. p. 2980–8.
55. Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City (UT): IEEE; 2018. p. 4510–20.
56. He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks. European Conference on Computer Vision. Amsterdam (The Netherlands): Springer; 2016. p. 630–45.
57. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. Munich (Germany): Springer; 2015. p. 234–41.
58. Hashemi SR, Prabhu SP, Warfield SK, et al. Exclusive independent probability estimation using deep 3d fully convolutional densenets for isointense infant brain MRI segmentation, arXiv preprint arXiv: 1809.08168.
59. Sun J, Yan S, Song C, et al. Dual-functional neural network for bilateral hippocampi segmentation and diagnosis of alzheimer's disease. Int J Comput Assist Radiol Surg 2020;15(3):445–55.
60. Milletari F, Navab N, Ahmadi S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. 2016 Fourth International Conference on 3D Vision (3DV), IEEE, Stanford University, California, October 25, 2016. p. 565–71.

61. Yin S, Peng Q, Li H, et al. Automatic kidney segmentation in ultrasound images using subsequent boundary distance regression and pixelwise classification networks. Med Image Anal 2020;60: 101602.
62. Simonyan K, Zisserman A. Very deep convolutional networks for largescale image recognition, arXiv preprint arXiv: 1409.1556.
63. Chen L, Papandreou G, Kokkinos I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, arXiv preprint. arXiv:1606.00915.
64. Chen L, Papandreou G, Kokkinos I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, arXiv preprint arXiv:1606.00915.
65. de Vos BD, Berendsen FF, Viergever MA, et al. A deep learning framework for unsupervised affine and deformable image registration. Med Image Anal 2019;52:128–43.
66. Balakrishnan G, Zhao A, Sabuncu MR, et al. Voxelmorph: A learning framework for deformable medical image registration, arXiv preprint arXiv: 1809.05231.
67. Shen Z, Han X, Xu Z, et al. Networks for joint affine and nonparametric image registration, arXiv preprint arXiv:1903.08811.
68. Eppenhof KAJ, Lafarge MW, Pluim JPW. Progressively growing convolutional networks for end-to-end deformable image registration. In: Angelini ED, Landman BA, editors. Medical imaging 2019: image processing, ociety of Photo-Optical Instrumentation Engineers (SPIE). Bellingham (WA): SPIE Press; 2019. p. 338–44.
69. Yang H, Sun J, Carass A, et al. Unpaired brain MR-to-CT synthesis using a structure-constrained CycleGAN, arXiv preprint arXiv:1809.04536.
70. Zhu J.-Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE International Conference on Computer Vision. Venice (Italy): IEEE; 2017. p. 2223–2232.
71. Rubin J, Abulnaga SM. CT-To-MR conditional generative adversarial networks for ischemic stroke lesion segmentation. IEEE Int Conf Healthc Inform 2019;(2019):1–7.
72. Bass C, Dai T, Billot B, et al. Image synthesis with a convolutional capsule generative adversarial network. Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning, Vol. 102 of Proceedings of Machine Learning Research. London (UK): PMLR; 2019. p. 39–62.
73. LaLonde R, Bagci U, Capsules for object segmentation, arXiv preprint arXiv:1804.04241.
74. Sabour S, Frosst N, Hinton GE, Dynamic routing between capsules, arXiv preprint arXiv:1710.09829.
75. Sriram A, Zbontar J, Murrell T, et al, GrappaNet: Combining parallel imaging with deep learning for multi-coil MRI reconstruction, arXiv preprint arXiv: 1910.12325.
76. Akagi M, Nakamura Y, Higaki T, et al. Deep learning reconstruction improves image quality of abdominal ultra-high-resolution CT. Eur Radiol 2019;29(11): 6163–71.
77. Gondara L. Medical image denoising using convolutional denoising autoencoders. 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). Barcelona (Spain): IEEE; 2016. p. 241–6.
78. Jifara W, Jiang F, Rho S, et al. Medical image denoising using convolutional neural network: a residual learning approach. J Supercomput 2019; 75(2):704–18.