Published in final edited form as:
*Semin Musculoskelet Radiol*. 2020 February ; 24(1): 3–11. doi:10.1055/s-0039-3401041.

# Artificial Intelligence Explained for Nonexperts

**Narges Razavian, PhD**[1], **Florian Knoll, PhD**[2], **Krzysztof J. Geras, PhD**[3]

[1]Department of Radiology and Population Health, NYU Langone Health and NYU Center for Data Science, New York, New York

[2]Department of Radiology, NYU Langone Health, New York, New York

[3]Department of Radiology, NYU Langone Health and NYU Center for Data Science, New York, New York

## Abstract

Artificial intelligence (AI) has made stunning progress in the last decade, made possible largely due to the advances in training deep neural networks with large data sets. Many of these solutions, initially developed for natural images, speech, or text, are now becoming successful in medical imaging. In this article we briefly summarize in an accessible way the current state of the field of AI. Furthermore, we highlight the most promising approaches and describe the current challenges that will need to be solved to enable broad deployment of AI in clinical practice.

### Keywords

deep learning; artificial intelligence; review; medical imaging

## Artificial Intelligence and Machine Learning

Artificial intelligence (AI) is a broad field that investigates constructing computer programs that exhibit some form of intelligent behavior. Examples of such behavior include playing board games, recognizing speech, identifying objects in images, understanding relations between entities in text, or translating text between two languages. Machinelearning[1–3] is currently a dominant approach to AI, based on the principle of *learning from data*. This principle can be contrasted with attempting to write a program explicitly to achieve a specific task, which was the focus of many early attempts at creating AI. ►Fig. 1 shows the relations between different subfields of AI.

Before exploring further, we must define a few key terms. Every machine learning model has the *input*, that is, what the model is given. In medical imaging the input would simply be a single image, a set of images, or a sequence of images. Learning models also have the

Address for correspondence Narges Razavian, PhD, Department of Radiology and Population Health, NYU Langone Health, and NYU Center for Data Science, Ste. 639, 227 East 30th Street, New York, NY 10016 (narges.razavian@nyulangone.org).

Conflict of Interest
Florian Knoll reports grants from the National Institutes of Health during the conduct of the study. In addition, he is one of the inventors of US patent 20,170,309,019: A1: System, method and computer-accessible medium for learning an optimized variational network. Narges Razavian and Krzysztof J. Geras have declared no conflicts of interest for this article.

*output*, that is, what we are trying to predict. For example, in the context of medical imaging, this can be a diagnosis for the given input. Every learning algorithm also has a *training objective,* a quantity we want to minimize in the process of training that indicates how good our model is at predicting the output given the input on the training data. For most *parametric* models, that is, models that contain parameters, the process of training consists of presenting the learning algorithm with the training data multiple times, allowing the model to adjust its parameters to the data. The process of adjusting the parameters to the data typically consists of iteratively taking a gradient of the training objective with respect to the parameters and changing the values of the parameters in the direction of the gradient. This training procedure is called the *gradient descent*.

Machine learning is divided into a few broad paradigms, among which the most commonly used are unsupervised learning, supervised learning, and reinforcement learning. Unsupervised learning aims at discovering the structure in the data that has no labels or categories assigned to training examples. The most common unsupervised learning task is clustering that consists of grouping similar examples together according to some predefined similarity metric. The purpose of this task to learn good feature representations from unannotated data sets and to enable discovering patterns in the data that would otherwise be difficult to notice. Recently, self-supervised learning, which is a special case of unsupervised learning, has emerged as a strong representation learning method. Self-supervised learning models attempt to use parts of the unlabeled data to predict other parts,[4–6] for example, using a black-and-white image and predicting the color,[4] or predicting how to organize several patches of an image correctly after we have shuffled them.[5] These methods now are the top performers in natural language processing applications[7–9] and have shown promise in various imaging tasks.

Although all of the learning paradigms just described have many important applications, supervised learning, in which every training example comes with an expected output, is the most frequently used in practice. The most common examples of supervised learning tasks are classification and segmentation. In classification the output is representing a label for the entire input. In segmentation, the output contains a separate label for every element of the input.

Reinforcement learning can be considered the third learning paradigm in machine learning. In this task, the algorithm is trained to optimize a reward that can only be computed after a long sequence of actions, where each action by itself is difficult to assign a numeric reward, but the final reward is easy to specify. Winning a long board game versus losing or learning to control a drone through movements of many motors are examples of reinforcement learning. A few recent publications on AlphaGo,[10] Poker,[11] or several strategic video games[12,13] all rely on reinforcement learning to learn to take actions that lead to winning the game. A related work on magnetic resonance imaging (MRI) research showed that new MR sequences can be learned via reinforcement learning.[14]

Among the three subfields of machine learning, supervised learning has contributed the most to medical applications of AI.

## Supervised Learning Models

Supervised learning is the paradigm that has had the most practical success in recent years. Some examples of routinely used supervised learning models not based on neural networks include logistic regression, decision trees, and support vector machines. The common factor in all of these methods is that although the decision process they use to arrive at the classification decision might be very complex, they do not learn any intermediate *representations* of the data. That implies these methods can only work well if the input features they are presented with are very predictive to begin with. In some applications, where the input features have a very well-defined meaning, this is not much of an issue. However, in computer vision classification tasks, for example, it is very difficult to go straight from the input pixels to the label for the entire image. It would be much more desirable to have a method to extract more abstract features from the pixels before attempting to classify an image. This observation and the fact that it is very difficult to define this sort of transformation explicitly is the key to understanding why neural networks work so well for difficult perceptual problems compared with other methods that do not learn intermediate representations for the data.

## Neural Networks and Deep Learning

Neural networks are learning models that consist of *layers*. In general, we can define the neural network as a composition of functions with learnable parameters acting on the input, usually denoted by **x**. The output of the neural network would then be computed as $f_k(f_{k-1}(f_{k-2}(\ldots f_1(\mathbf{x})\ldots)))$. All functions in this expression, except for the most outer one, represent one *hidden* layer. The most outer function represents the *output layer* and is usually different because its output is the final prediction produced by the network. For classification it is usually a softmax function with the number of outputs equal to the number of possible classes. For a segmentation task, a softmax function would also be used, but there would be one set of outputs for each pixel in the input. The *architecture* of a neural network is defined by the number of different layers, a design of each layer individually, and how the layers are connected to each other. The subfield of machine learning studying neural networks, usually consisting of many layers, is often referred to as *deep learning*. ►Fig. 2 shows an example of a simple neural network.

## The Success of Deep Learning

Although neural networks have been known for a long time, it is only in the past decade that they have emerged as the most successful methods for perception tasks including computer vision and speech recognition, as well as, more recently, for natural language processing. Currently, we are witnessing an explosion of applications of neural networks to different domains: radiology, physics, genetics, drug design, digital pathology, and beyond. Several important factors contribute to the recent dominance of deep learning methods and their only moderate practical success in previous decades.

First, among all available models for supervised learning, from logistic regression to random forest to neural networks, deep learning models are the largest models in terms of number of parameters and capacity to learn complex tasks. The success of these models came from the ability to learn hierarchical feature representations *and* the logic to combine these learned features. This entails more than hundreds of thousands to millions of parameters. As an example, the first deep learning model that achieved top results in the Image-Net[15] computer vision benchmark in 2013[16] has 60 million parameters. In machine learning, as a model's parameter count increases, larger data sets are required to avoid overfitting. In the earlier decades of AI development, digital data sets were small and expensive to store, so most neural network models were trained and tested on small data sets. Deep learning models will not be able to train well and generalize to unseen test cases when data sets are too small, and only after the so-called big data era were these models able to finally showcase their superior learning capacity.

Second, training models with large parameters via back-propagation involve millions of operations for every training example that becomes prohibitively slow when using large data sets. The use of graphical processing units (GPUs) rather than central processing units (CPUs) was important for the successful training of the first and every deep learning model in 2009[17] and beyond. GPUs were primarily designed and optimized for the fast parallel computation necessary in high-resolution graphical displays and the gaming industry. The matrix multiplication operations in deep learning methods turned out to benefit from the same optimizations. Switching from CPU to GPU computation improved the computational speed of larger networks by orders of magnitude. This led to a reduction in the training time of these networks, from months and years to days.

Third, in machine learning, the practice of open-source tool development and public release of libraries, and emphasis on reproducibility and usability, which accelerated only in the past decade, has been a major contributor to the exponential growth of research and development in deep learning. As communities of thousands of developers built and maintained open-sourced deep learning toolkits, different laboratories were able to explore architectural innovations faster, without having to reinvent the toolkits again. Several open-source general-purpose deep learning libraries include Caffe,[18] Torch,[19] Theano,[20] Tensorflow,[21] and Pytorch.[22] All major models have been released in multiple environments, allowing adaptation by other researchers. In the same trend, availability of benchmark evaluations such as the ImageNet annual competition[15] or the GLUE natural language processing benchmark[9] have led to faster innovation cycles within the deep learning field. Methods can be compared one to one, on the same evaluation task, without having to implement all previous existing methods.[17,43]

## Convolutional Neural Networks

Among the deep neural network architectures, convolutional neural networks (CNNs; also known as *convnets*)[23] and their varieties are currently enjoying the greatest practical success. Convnets have a special structure of connections between layers, resembling the human visual cortex, that is particularly well suited for perceptual tasks. This special structure of connections in convnets is implemented through two unique types of layers: a convolutional layer and a pooling layer. ►Fig. 3 shows an example of a simple convnet. Thanks to these two types of layers, convnets have fewer parameters, making them less prone to overfitting, and they are invariant (i.e., the change in the input does not change the output) to some transformations that are common in image data such as small translations and rotations. These invariances allow them to generalize well to new data.

Early versions of convnets, such as LeNet,[24] which had just a few convolutional layers, were effective in simple visual tasks, for example, digit and letter recognition. More complex versions of convnets[16,25–29] were later shown to achieve superior performance in more complex tasks involving natural and medical images. Currently, an overwhelming majority of neural network architectures used in computer vision are relying on the concepts introduced in convnets.

Next we offer a few examples, illustrating how the ideas just described are applied in practical neural network architectures.

## Neural Network Example 1: ResNet

Until 2015 all of the most popular CNNs were designed under the assumption that only two consecutive layers in a neural network should be directly connected to each other. This principle was refuted by the success of *residual networks,* also known as *ResNets*.[28] They are very similar to the more conventionally designed convnets such as AlexNet[16] or VGGNet,[25] except that some of their nonconsecutive hidden layers are directly connected.

To understand the novelty of ResNets, we need to introduce some simple notation. Let's denote some intermediate layer in the network as $h_k$ and the layer following that layer as $h_{k+1}$. In standard convnets, the mathematical relation between two consecutive layers can be written as $h_{k+1} = f(h_k;W)$, where f is some nonlinear function, parameterized by W. In contrast, residual networks have a slightly different structure. They are built with *residual units* that are connected such that $h_{k+1} = f(h_k;W) + h_k$. That is, an output of a residual unit is computed as the input of the residual unit plus the input after some nonlinear transformation. In summary, the special structure of connections in residual units allows the network to learn to push the information toward the output without transforming it in some layers.

Residual networks consist of several, sometimes hundreds, of such residual units. Thanks to the residual connections in ResNets, it is possible to train extremely powerful networks that can be even 1,000 layers deep. ResNets were shown experimentally to be faster and easier to train, as well as better at generalizing new data. For example, in 2014, the VGGNet was the best model in the ImageNet Large Scale Visual Recognition Challenge. Before ResNets were introduced, VGGNet achieved the classification error of 7.4%. In the challenge next year, the error by the winning ResNet was reduced to 3.57%. Interestingly, this improvement was achieved while dramatically reducing the number of learnable parameters. For example, the 16-layer VGGNet has ~ 138 million parameters, whereas the more accurate 110-layer resnet[30] has only 1.7 million parameters.

An overwhelming majority of the most successful convolutional networks currently used in practice are based on ResNets or their derivatives such as DenseNets.[29]

## Neural Network Example 2: U-Net

Several computer vision tasks require localization of an object or a class within the image that increases the complexity of the models by requiring a prediction for every pixel in the image. The application domains span from self-driving cars and robotics to biomedical imaging, and each application domain has its own requirements. In self-driving cars, computational speed is a priority because the segmentation model is needed over live video streams at ≥ 30 frames per second. In contrast, in biomedical imaging applications, accuracy is prioritized. U-Net[31] is a segmentation model developed initially for biomedical imaging tasks, with a focus on improved model performance and accuracy rather than computational speed.

The U-Net model was built on a prior model known as a fully convolutional network,[32] a fast architecture that uses standard convolution and pooling layers at different resolutions to classify each pixel in the image. In U-Net architecture, the model similarly combines segmentation and spatial information at various resolutions. An input is processed via a convolution layer and then down-sampled via a pooling layer. The resulting processed input goes through the same convolution and pooling operation four times. What makes U-Net distinct is that the final result of the down-sampling is transformed and then up-sampled hierarchically, reversing the down-sampling layers one by one. In addition, each up-sampling step has access to its corresponding down-sampling level's information.

By preserving spatial information at various resolutions, the U-Net architectures avoid having to learn the spatial information from data, and all patches of the image contribute to the training of the convolution layer parameters. As a result, U-Net models are extremely data efficient, and unlike classification models, they can be trained with as few as 20 or 30 fully annotated data points, depending on the task. This advantage has made U-Nets among the most commonly used models in radiology and medical imaging tasks.

## Neural Network Example 3: Variational Network

Neural networks can also be used for the task of image reconstruction, which means the generations of actual human-interpretable images from the measurement data that are acquired by the imaging hardware. The individual tasks required in this step depend on the physics of the acquisition device. For example, radiographs and computed tomography (CT) acquire data about the attenuation of ionizing radiation that is sent through the body, MRI acquires a signal from the excitation of polarized hydrogen atoms, and sonography uses ultrasound waves to probe the acoustic impedance of the tissue. From a mathematical point of view, all these tasks can be formulated in what is commonly known as an inverse problem: using observations to determine the underlying origin that caused a certain measurement.

In medical imaging, deep learning was introduced recently to relax the constraints on data acquisition while still producing diagnostically equivalent images. A neural network for image reconstruction receives as input the raw measurement data from the imaging system and (in the case of supervised methods, which are predominantly used at the time of this writing), a ground truth reference image that is obtained by conventional methods from uncorrupted measurement data. Note that the measurement data usually live in different domains than the reconstructed images. For example, MRI data are acquired in Fourier space, and CT data have the form of sinogram projection data. The network then takes the form of a regularizer in a constrained optimization problem that finds the solution that is both consistent with the input measurement data as well as whose statistics are closest to the image statistics of the uncorrupted training data.

One particular example of a neural network architecture that has been proposed to solve this class of problems is the so-called variational network. This approach was recently proposed for MRI with reduced scan times[33,34] and CT imaging with a reduced dose.[35] This architecture is inspired by classic iterative optimization procedures that are used to solve inverse problems as well as by developments in compressed sensing.[36] Each layer of the variational network corresponds to one iteration step of a gradient-based optimization. Another design feature of the variational network is that it uses knowledge of the imaging physics inside the architecture. For example, in MRI imaging, the Fourier transform operator itself is not learned but hardcoded explicitly inside the network architecture. This design feature is also used in other recently proposed architectures.[37–39] In contrast, other recently proposed approaches[40] learn the complete mapping between raw measurement data to the final image. Architectures that follow this design principle require the use of fully connected layers that leads to a substantially higher number of model parameters in comparison with architectures like the variational network that use only convolutional layers.

## A Brief Timeline of Recent CNN Architectural Developments

Several architectural elements of neural networks in use today have been the result of scores of theoretical and empirical findings by the machine learning community. Reviewing the timeline and motivations for the development of these elements can help put into context the motivations and decisions made by researchers in the past few years.

In 2012 a deep learning model developed by Krizhevsky et al[16] reduced the error of ImageNet object classification benchmark by almost 50% compared with traditional machine learning methods. Compared with the CNN models developed in the 1990s by LeCun et al,[23] Krizhevsky and colleagues introduced several unique architectural and engineering aspects: GPU computation to reduce training time from order of months to days, a novel regularization method termed *dropout,*[41] and use of a *ReLU*[42] nonlinearity instead of traditionally used *tanh* function. Since then, the annual ImageNet competition has been dominated by CNNs with new architectural innovations almost every year. In both 2013 and 2014, top performing models OverFeat[27] and VGG[25] showed that architectures could get much larger and deeper while still generalizing well to new data sets and not overfitting to training data. OverFeat had five convolutional and pooling layers, and VGG had 16 to 19 convolutional and pooling layers, as opposed to two layers in the model by Krizhevsky et al. The number of parameters of these models went from 60 million (Krizhevsky et al) to 144 million (in OverFeat and VGG). Around this time, open-source deep learning tools such as Caffe[18] (based on C++ and Python), Torch[19] (based on Lua), and Theano[20] (based on Python) were commonly used by various groups. The implementation of these winning and other architectures was quickly open-sourced by authors and used by the broader AI community.

In 2014, in addition to VGG, another model achieved top performing results. This architecture, termed *inception,*[26] introduced the concept of multi-resolution kernels to allow more flexibility to the model while keeping the number of parameters small. Although variants of this architecture such as Inception v3[43] were as deep as 300 layers, their total number of parameters was only ~ 23 million. This model is still used heavily in medical imaging. As models gradually got deeper, reaching 300 layers in inception, one major challenge emerged: Models would no longer train well via the standard backpropagation method because the lower layers in the architecture received a much smaller training signal. This phenomenon is known as the gradient vanishing problem[44] and had existed since the introduction of recurrent neural networks. In 2014, skip connections that connect nonconsecutive layers and create a shortcut to better pass training signal emerged in CNN architectures. One variant of such models was highway networks.[45] The 2015's top performing model in the ImageNet challenge was ResNet[28] that solved the depth challenge by introducing *residual blocks* that had two convolution layers along with a connection that skipped these layers altogether. This led to a model with a depth of 110 layers (~ 52 residual blocks) with only 1.7 million parameters that achieved state of the art in computer vision.

In November 2015, the open-source release of the deep learning library TensorFlow[21] (based on Python) further accelerated the speed of AI innovations and model sharing.

Because skip connections had shown performance gains across all architectural designs in the early 2015, in the field of image segmentation, Ronneberger et al[31] used the concept to create a hierarchical architecture that can use low- and high-level details of an image to create the segmentation mask for that image. This architecture was called *U-Net*, which achieved impressive performances for various image segmentation tasks. These architectures have all been used heavily in medical imaging and radiology applications.

The focus of the innovation has shifted toward increasing model capacity in the past 3 years. Availability of larger data sets, introduction of more open-source deep learning libraries such as PyTorch[22] (in 2016), and progress in unsupervised learning methods have encouraged researchers to increase model sizes interchangeably and devise new techniques that will improve training of these larger models.

DenseNet,[29] the winner of the 2016 ImageNet challenge, showed that accumulating output of convolutional layers together creates superior performance at the cost of 25.6 million parameters. In the field of natural language processing, architectures as large as 340 million[7] or even 1.6 billion parameters,[8] which use unsupervised[46,47] and self-supervised learning,[4,5,48] have now achieved impressive results in various text understanding and generation tasks,[7] and such improvements are not yet fully translated to computer vision.

The public sharing of trained models, open-sourcing all codes, benchmarks such as ImageNet, and the emphasis on reusability has led to an accelerated rate of progress in the field of architectural design and adaptation that is expected to continue and impact all application areas including medical imaging.

## Deep Learning Is the Right Tool for Medical Imaging

Just as in computer vision tasks for natural images, in medical image analysis we are trying to solve difficult learning tasks. It is not feasible to define explicitly a transformation of the input that would preserve the information we care about and discard what is not relevant to the final task. Therefore, to find an appropriate representation, we need to learn what it should be from data. Neural networks are the right tool to do that. These networks also allow us to build end-to-end predictors that, by design, can seamlessly integrate information of various kinds.

For example, for cancer diagnosis tasks, it would be very difficult, if not impossible, to enumerate and characterize exhaustively all possible types of potentially malignant and benign findings. It would then be challenging to describe these findings in low-level terms that could be used to design features useful for a simple learning algorithm, such as a decision tree. Neural networks have the ability to extract these features from data automatically, without the help of an expert. It is also, in principle, easy to imagine a neural network that can take as input all images in the examination, demographic information on the patient, their personal and family cancer history, and output a prediction for this examination. With backpropagation, all parameters of a neural network can be optimized jointly in a single training loop.

## Challenges and Limitations of Current Methods

Although the future of applications of deep learning to medical imaging is definitely bright, a lot of research in the principles of neural networks is necessary to realize the full potential of this family of models and make them a clinically practical tool. Currently most commonly used deep learning architectures have several serious limitations.

### Medical Images Are Very Different from Natural Images

The most successful and widely applied neural network architectures are designed for natural images that differ from medical images in multiple ways. For example, whereas popular neural network architectures are designed to work with two-dimensional images that are a few hundred pixels on either side, medical images are often bigger than 1,000 × 1,000 pixels or even three-dimensional. Furthermore, objects of interests in natural images are usually relatively large in relation to other items in the image. Therefore, down-sampling natural images usually does not erase the information that is necessary for deep networks to learn. In contrast, in medical images the objects that determine the class are often very small. Down-sampling such images would make learning impossible. Additionally, medical images frequently come in sets or sequences rather than individually, which is also quite unusual for natural images. This discrepancy implies that not all practices and network design principles considered standard for natural images are necessarily beneficial when learning with medical images.

### Splitting Training, Testing, and Validation Data Correctly

Deep learning models, especially the larger models commonly used today, have the ability to overfit to a small data set, meaning that they memorize every detail of the training input.

This ability can lead to highly inflated validation and test results, if the data points used in the validation or test set are either from the same patient as a data point in the training set or from the same tissue. This is often termed *data leakage.* The correct way to set up any deep learning study is to make sure the training, validation, and test data are first split randomly according to patients. This ensures that no data from the same patient reside in training, validation, and test sets simultaneously.

This simple but important potential pitfall has impacted several previously published articles, making the landscape of peer-reviewed scientific research difficult to navigate. A recent review[49] analyzed 33 published works on Alzheimer's classification from MRI and found that > 50% of the published work that had reported very high accuracies had suffered from one or more cases of data leakage between training, validation, and test sets. Similar reviews in other domains are necessary and have yet to be conducted.

## Confidence and Calibration

Classification and segmentation models are typically trained via optimizing a probabilistic training objective or loss function, most often simply maximizing the log likelihood of the correct class, pixel-wise or image-wise. The probabilistic nature of the loss function gives the illusion that a model's predicted probability can be used as a measure of confidence. However, in practice, deep learning models do not typically produce calibrated predictions,[50] meaning that the probability they assign to an outcome does not reflect the true probability of the outcome occurring in the population for that specific input. Raghu et al[51] recently showed that models trained to directly predict uncertain cases (via supervised learning and using radiologist disagreement as a measure of uncertainty) have a better uncertainty estimate compared with using model-predicted probabilities as a measure of confidence. This issue has direct consequences for deployment of AI models in clinics. Consequently, the trust and reliability of these models has remained an open issue in AI in health care applications.

## Difficulty of Interpretation

As models grow in complexity and size, generating human-interpretable explanations for the model's internal steps and the final prediction becomes increasingly difficult. This difficulty is in part due to a lack of objective definitions for interpretability,[52] in part due to the large number of features contributing to the predictions of deep learning models, making it hard for a human to aggregate the information, and in part due to the challenge of separating confirmation bias from correct feature explanations.[53]

In recent years, several techniques for explanation and visualizations of deep learning models have been proposed. Methods for explaining deep learning models include deconvolution and occlusion,[54] saliency map generation methods such as backpropagation,[55,56] guided backpropagation,[57] integrated gradients,[58] and methods that directly encode explainable layers into the architecture, for example class activation mapping[59] and attention-based[60] models such as caption generation models.[61]

At the moment, objective evaluations of models that generate an explanation over the input is difficult, and this is a common problem in any generative model.[62] As a result, many of the proposed methods for explaining deep learning models suffer from confirmation bias.[53] In particular, it was shown that variants of the commonly used guided backpropagation are unreliable. Similar studies are needed for evaluating other existing explanation methods. Overall, these challenges make interpretable deep learning an open problem, one that is still not fully solved.

### Hidden Variables, Robustness, and Safety

As deep learning models get larger and stronger, they become able to look naively for any pattern that differentiates different outcomes of interest, even if the pattern is not directly related to the pathology or outcome. Zech et al[63] found that when prevalence of a disease changed during deployment, standard deep learning models trained on one cohort failed to generalize to a new cohort. In their example, the model had learned to focus on the presence of metallic tokens that often appeared in radiographs in specific sites, to adjust the probability of chest radiograph pathologies.When a technician’s pattern of metallic token usage changed in a new institution, models performance dropped significantly. This issue has major implications in model deployment, regulation, and safety. Should an institution deploy a trained model that is only validated elsewhere? What regulations are necessary to avoid systematic errors and performance fluctuations in this case? In addition, today’s standard AI models are incapable of producing a warning about a distribution shift when the deployment environment has a different distribution than the development environment. A few related studies have started to address these questions, but the results are still in the preliminary stages and requires further research.[64]

Even though many of the problems cited here are being researched in the field of machine learning, it will take time to validate these results and to apply them to medical imaging. Furthermore, although there are already promising results in using deep learning for medical imaging, the translation from research to clinical application will not be easy. With few exceptions, techniques in most articles are evaluated on test sets that are different (usually a lot easier) from what can be seen in a clinical environment. Unfortunately, very few clinically realistic data sets are currently available. Data sets of this kind would allow for an objective comparison between different models, and they would provide a realistic assessment of their performance if deployed in practice.

## Further Reading

A comprehensive review of applications of deep learning in the field of musculoskeletal imaging is available in Gyftopoulos et al,[65] Hirschmann et al,[66] and Burns et al.[67] For more background on the development of deep learning, LeCun and colleagues[68] provide an in-depth review. For an extensive commentary on the pitfalls and challenges of AI applications, Riley[69] and Wiens et al[70] cover different focus areas, all of which are necessary and basic knowledge for anyone who makes decisions about using AI in practice.

A few publicly available large musculoskeletal data sets accompanied by benchmarks are also available including ChestX-ray8,[71] CheXpert,[72] MIMIC-CXR,[73] MURA,[74] and knee MRI,[75] among others.

## Financial Disclosure

We acknowledge support from the National Institutes of Health under grants R01-EB024532, P41-EB017183, and R21 EB027241.

## References

1. Bishop CM. Pattern Recognition and Machine Learning. New York, NY: Springer; 2006
2. Hastie T, Tibshirani R, Friedman J, Franklin J. The elements of statistical learning: data mining, inference and prediction. Math Intell 2005;27(02):83–85
3. Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge, MA: MIT Press; 2016
4. Zhang R, Isola P, Efros AA. Colorful image colorization. Computer Vision – ECCV 2016 Available at: https://arxiv.org/abs/1603.08511. Accessed January 8, 2020
5. Noroozi M, Favaro P. Unsupervised learning of visual representations by solving jigsaw puzzles. Computer Vision – ECCV 2016 Available at: https://arxiv.org/abs/1603.09246. Accessed January 8, 2020
6. Agrawal P, Carreira J, Malik J. Learning to see by moving. Proceedings of the IEEE International Conference on Computer Vision. Available at: https://arxiv.org/abs/1505.01596. Accessed January 8, 2020
7. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. arXiv [csCL]. 10 2018 Available at: http://arxiv.org/abs/1810.04805. Accessed January 8, 2020
8. Lample G, Conneau A. Cross-lingual language model pretraining. arXiv [csCL]. 1 2019 Available at: http://arxiv.org/abs/1901.07291. Accessed January 8, 2020
9. Wang A, Singh A, Michael J, Hill F, Levy O, Bowman S. GLUE: a multi-task benchmark and analysis platform for natural language understanding. arXiv [csCL]. 4 2018 Available at: http://arxiv.org/abs/1804.07461. Accessed January 8, 2020
10. Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go without human knowledge. Nature 2017;550(7676):354–359 [PubMed: 29052630]
11. Brown N, Sandholm T. Superhuman AI for multiplayer poker. Science 2019;365(6456):885–890 [PubMed: 31296650]
12. Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with deep reinforcement learning. arXiv [csLG]. 12 2013 Available at: http://arxiv.org/abs/1312.5602. Accessed January 8, 2020
13. Arulkumaran K, Cully A, Togelius J. AlphaStar: an evolutionary computation perspective. arXiv [csNE]. 2 2019 Available at: http://arxiv.org/abs/1902.01724. Accessed January 8, 2020
14. Zhu B, Liu J, Koonjoo N, Rosen B, Rosen MS. AUTOmated pulse SEQuence generation (AUTOSEQ) and neural network decoding for fast quantitative MR parameter measurement using continuous and simultaneous RF transmit and receive. ISMRM Annual Meeting & Exhibition. Vol 1090 2019 Available at: http://www.enc-conference.org/portals/0/Abstracts2019/ENC20198520.4608VER.2.pdf. Accessed January 8, 2020
15. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. Available at: www.image-net.org/papers/imagenet_cvpr09.pdf. Accessed January 8, 2020
16. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, eds. Advances in Neural Information Processing Systems 25. Red Hook, NY: Curran Associates; 2012: 1097–1105
17. Raina R, Madhavan A, Ng AY. Large-scale deep unsupervised learning using graphics processors. In: Proceedings of the 26th Annual International Conference on Machine Learning. ICML '09 New York, NY: ACM; 2009:873–880
18. Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional Architecture for Fast Feature Embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia MM '14. New York, NY: ACM; 2014:675–678
19. Collobert R, Bengio S, Mariéthoz J. Torch: a modular machine learning software library. Available at: https://infoscience.epfl.ch/record/82802/files/rr02-46.pdf. Accessed January 8, 2020

20. Bergstra J, Breuleux O, Bastien F, et al. Theano: a CPU and GPU math expression compiler In: Proceedings of the Python for Scientific Computing Conference (SciPy). Vol 4 Austin, TX; 2010 Available at: https://conference.scipy.org/scipy2010/slides/james_bergstra_theano.pdf. Accessed January 8, 2020

21. Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. arXiv [csDC]. 3 2016 Available at: http://arxiv.org/abs/1603.04467. Accessed January 8, 2020

22. Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch. 10 2017 Available at: https://openreview.net/pdf?id=BJJsrmfCZ. Accessed October 14, 2019

23. LeCun Y, Boser BE, Denker JS, et al. Handwritten digit recognition with a back-propagation network In: Touretzky DS, ed. Advances in Neural Information Processing Systems 2. San Francisco, CA: Morgan-Kaufmann; 1990:396–404

24. LeCun Y, Boser B, Denker JS, et al. Backpropagation applied to handwritten zip code recognition. Neural Comput 1989;1(04): 541–551

25. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv [csCV]. 9 2014 Available at: http://arxiv.org/abs/1409.1556. Accessed January 8, 2020

26. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015:1–9

27. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. OverFeat: Integrated recognition, localization and detection using convolutional networks. arXiv [csCV]. 12 2013 Available at: http://arxiv.org/abs/1312.6229. Accessed January 8, 2020

28. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. arXiv [csCV]. 12 2015 Available at: http://arxiv.org/abs/1512.03385. Accessed January 8, 2020

29. Huang G, Liu Z, van der Maaten L, Weinberger KQ. Densely connected convolutional networks. arXiv [csCV]. 8 2016 Available at: http://arxiv.org/abs/1608.06993. Accessed January 8, 2020

30. He K, Zhang X, Ren S, Sun J. Identity mappings in deep residual networks. Computer Vision – ECCV 2016. Available at: http://arxiv.org/abs/1603.05027. Accessed January 8, 2020

31. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. arXiv [csCV]. 5 2015 Available at: http://arxiv.org/abs/1505.04597. Accessed January 8, 2020

32. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015:3431–3440

33. Hammernik K, Klatzer T, Kobler E, et al. Learning a variational network for reconstruction of accelerated MRI data. Magn Reson Med 2018;79(06):3055–3071 [PubMed: 29115689]

34. Knoll F, Hammernik K, Kobler E, Pock T, Recht MP, Sodickson DK. Assessment of the generalization of learned image reconstruction and the potential for transfer learning. Magn Reson Med 2019;81(01):116–128 [PubMed: 29774597]

35. Kobler E, Muckley M, Chen B, et al. Variational deep learning for low-dose computed tomography. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2018:6687–6691

36. Lustig M, Donoho D, Pauly JM. Sparse MRI: the application of compressed sensing for rapid MR imaging. Magn Reson Med 2007;58(06):1182–1195 [PubMed: 17969013]

37. Aggarwal HK, Mani MP, Jacob M. MoDL: Model-Based Deep Learning architecture for inverse problems. IEEE Trans Med Imaging 2019;38(02):394–405 [PubMed: 30106719]

38. Schlemper J, Caballero J, Hajnal JV, Price AN, Rueckert D. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. IEEE Trans Med Imaging 2018;37(02):491–503 [PubMed: 29035212]

39. Qin C, Schlemper J, Caballero J, Price AN, Hajnal JV, Rueckert D. Convolutional recurrent neural networks for dynamic MR image reconstruction. IEEE Trans Med Imaging 2019;38(01):280–290 [PubMed: 30080145]

40. Zhu B, Liu JZ, Cauley SF, Rosen BR, Rosen MS. Image reconstruction by domain-transform manifold learning. Nature 2018;555 (7697):487–492 [PubMed: 29565357]

41. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15:1929–1958

42. Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10); 2010:807–814

43. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016:2818–2826

44. Hochreiter S The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int J Uncertain Fuzziness Knowl Based Syst 1998;06(02):107–116

45. Srivastava RK, Greff K, Schmidhuber J. Highway networks. arXiv [csLG]. 5 2015 Available at: http://arxiv.org/abs/1505.00387. Accessed January 8, 2020

46. Yang J, Parikh D, Batra D. Joint unsupervised learning of deep representations and image clusters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016: 5147–5156

47. Jayaraman D, Grauman K. Slow and steady feature analysis: higher order temporal coherence in video. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Doi:10.1109/cvpr.2016.418

48. Doersch C, Gupta A, Efros AA. Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision; 2015:1422–1430

49. Wen J, Thibeau-Sutre E, Samper-Gonzalez J, et al. Convolutional neural networks for classification of Alzheimer's disease: overview and reproducible evaluation. arXiv [csLG]. 4 2019 Available at: http://arxiv.org/abs/1904.07773. Accessed January 8, 2020

50. Guo C, Pleiss G, Sun Y, Weinberger KQ. On calibration of modern neural networks. arXiv [csLG]. 6 2017 Available at: http://arxiv.org/abs/1706.04599. Accessed January 8, 2020

51. Raghu M, Blumer K, Sayres R, et al. Direct uncertainty prediction for medical second opinions. arXiv [csLG]. 7 2018 Available at: http://arxiv.org/abs/1807.01771. Accessed January 8, 2020

52. Lipton ZC. The myth of model interpretability. Available at: https://arxiv.org/pdf/1606.03490.pdf. Accessed January 8, 2020

53. Adebayo J, Gilmer J, Muelly M, Goodfellow I, Hardt M, Kim B. Sanity checks for saliency maps. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, R Garnett, eds. Advances in Neural Information Processing Systems 31. Red Hook, NY: Curran Associates; 2018:9505–9515

54. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. Computer Vision – ECCV 2014. Available at: http://arxiv.org/abs/1311.2901. Accessed January 8, 2020

55. Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks. arXiv [csCV]. 12 2013 Available at: http://arxiv.org/abs/1312.6199. Accessed January 8, 2020

56. Olah C, Mordvintsev A, Schubert L. Feature visualization. Distill 2017;2(11):e7

57. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M. Striving for simplicity: the all convolutional net. arXiv [csLG]. 12 2014 Available at: http://arxiv.org/abs/1412.6806. Accessed January 8, 2020

58. Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. arXiv [csLG]. 3 2017 Available at: http://arxiv.org/abs/1703.01365. Accessed January 8, 2020

59. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016:2921–2929

60. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv [csCL]. 9 2014 Available at: http://arxiv.org/abs/1409.0473. Accessed January 8, 2020

61. Xu K, Ba J, Kiros R, et al. Show, attend and tell: neural image caption generation with visual attention. arXiv [csLG]. 2 2015 Available at: http://arxiv.org/abs/1502.03044. Accessed January 8, 2020

62. Theis L, van den Oord A, Bethge M. A note on the evaluation of generative models. arXiv [statML]. 11 2015 Available at: http://arxiv.org/abs/1511.01844. Accessed January 8, 2020

63. Zech JR, Badgeley MA, Liu M, Costa AB, Titano JJ, Oermann EK. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. PLoS Med 2018;15(11):e1002683 [PubMed: 30399157]

64. Subbaswamy A, Schulam P, Saria S. Preventing failures due to dataset shift: learning predictive models that transport. arXiv [statML]. 12 2018 Available at: http://arxiv.org/abs/1812.04597. Accessed January 8, 2020

65. Gyftopoulos S, Lin D, Knoll F, Doshi AM, Rodrigues TC, Recht MP. Artificial intelligence in musculoskeletal imaging: current status and future directions. AJR Am J Roentgenol 2019;213(03): 506–513 [PubMed: 31166761]

66. Hirschmann A, Cyriac J, Stieltjes B, Kober T, Richiardi J, Omoumi P. Artificial intelligence in musculoskeletal imaging: review of current literature, challenges, and trends. Semin Musculoskelet Radiol 2019;23(03):304–311 [PubMed: 31163504]

67. Burns JE, Yao J, Summers RM. Artificial intelligence in musculoskeletal imaging: a paradigm shift. J Bone Miner Res 2019; 8 9

68. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521 (7553):436–444 [PubMed: 26017442]

69. Riley P Three pitfalls to avoid in machine learning. Nature 2019; 572(7767):27–29 [PubMed: 31363197]

70. Wiens J, Saria S, Sendak M, et al. Do no harm: a roadmap for responsible machine learning for health care. Nat Med 2019;25(09):1337–1340 [PubMed: 31427808]

71. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017:2097–2106

72. Irvin J, Rajpurkar P, Ko M, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. arXiv [preprint arXiv:1901 07031]. 2019 Available at: https://arxiv.org/abs/1901.07031. Accessed January 8, 2020

73. Johnson AEW, Pollard TJ, Berkowitz SJ, et al. MIMIC-CXR: a large publicly available database of labeled chest radiographs. arXiv [csCV]. 1 2019 Available at: http://arxiv.org/abs/1901.07042. Accessed January 8, 2020

74. Rajpurkar P, Irvin J, Bagul A, et al. MURA: large dataset for abnormality detection in musculoskeletal radiographs. arXiv [physics.med-ph] 12 2017 Available at: http://arxiv.org/abs/1712.06957. Accessed January 8, 2020

75. Zbontar J, Knoll F, Sriram A, et al. fastMRI: an open dataset and benchmarks for accelerated MRI. arXiv [csCV]. 11 2018 Available at: http://arxiv.org/abs/1811.08839. Accessed January 8, 2020

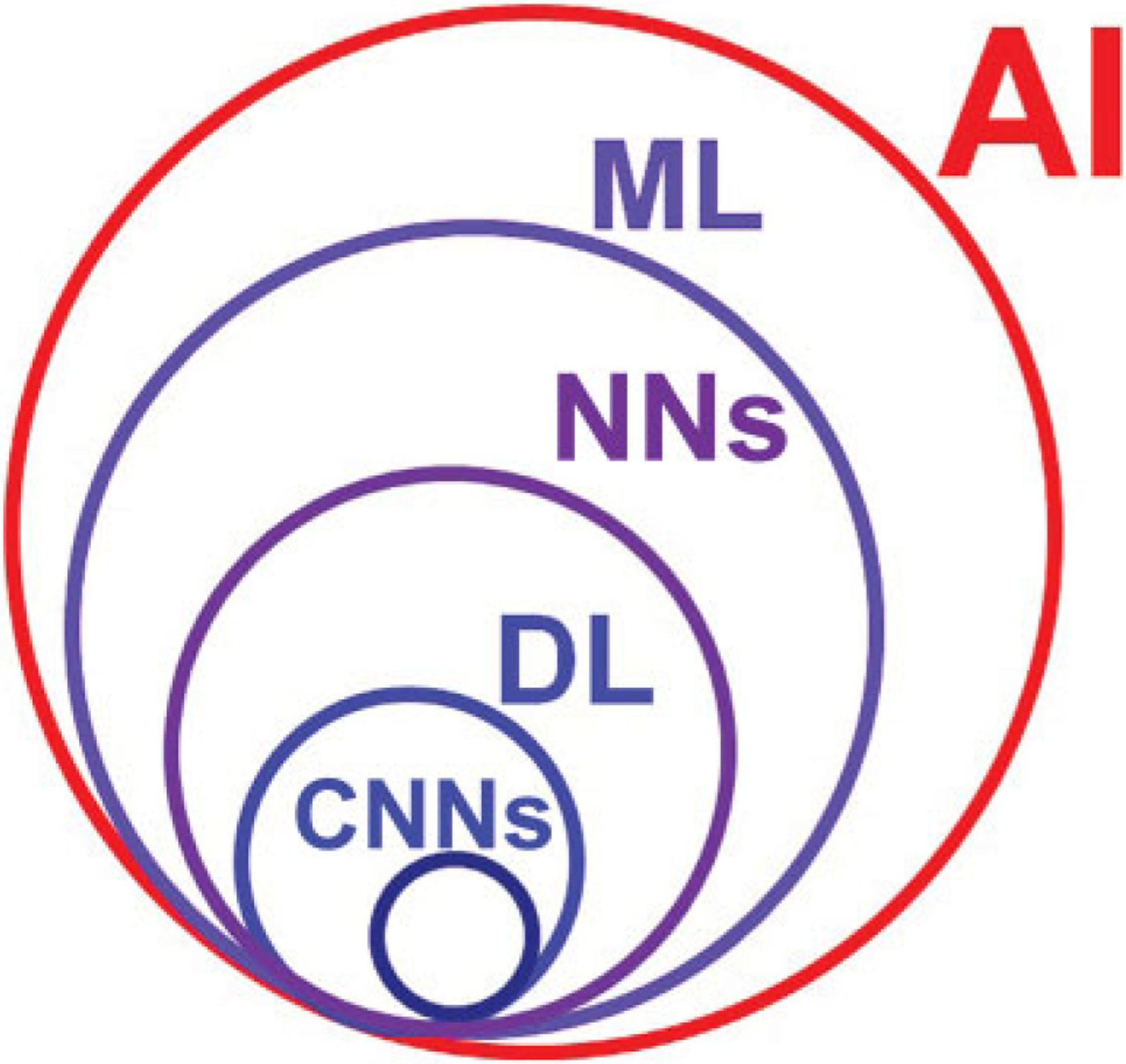

**Fig. 1.**

Relationship between different fields: artificial intelligence (AI), machine learning (ML), neural networks (NNs), deep learning (DL), and convolutional neural networks (CNNs).

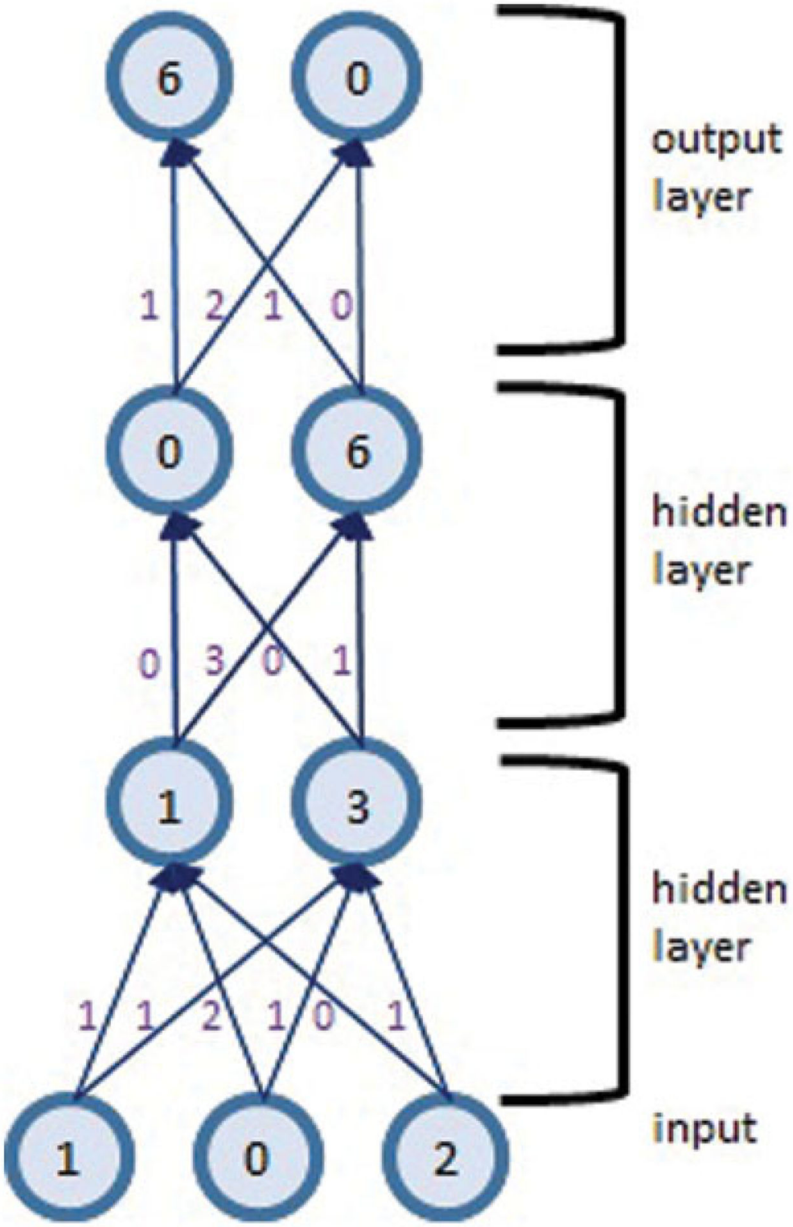

**Fig. 2.**

Illustration of a simple neural network with fully connected layers. The numbers on the edges are the learnable parameters of the network, and the numbers inside neurons represent their activations. The input to the network is a vector [1, 0, 2] that is transformed by the first layer into [1, 3], then by the following layer into [0, 6], and finally into [6, 0] by the output layer. For illustration purposes, we are assuming the activation values for each neuron of the network are computed as a simple weighted average of the values in the neurons in the preceding layer. In reality, after computing the weighted average, some nonlinear function is applied. This is necessary to allow the network to represent complicated transformations between the input and the output.

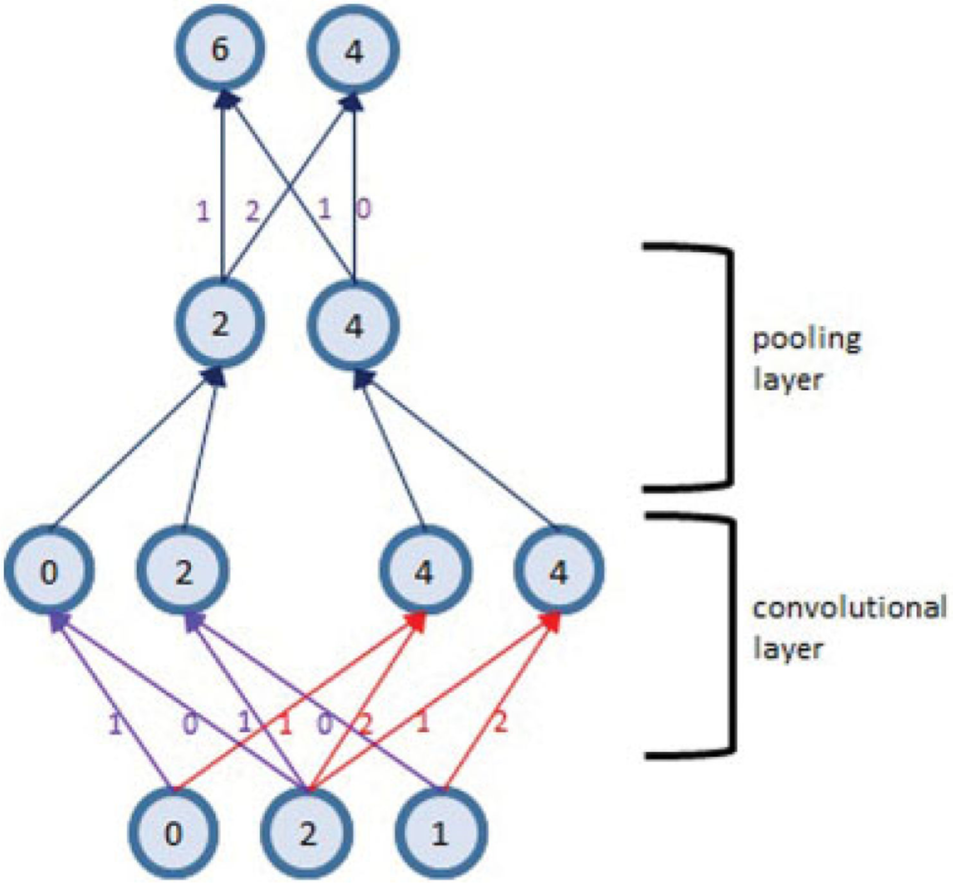

**Fig. 3.**

Illustration of a simple convolutional network for a one-dimensional input. The two unique types of layers illustrated here are the convolutional layer and the pooling layer. The special property of the convolutional layer is that it involves applying the same *filters* at different locations. The convolutional layer in this network contains two filters that transform the input into two *feature maps*. The pooling layer illustrated above is a *max pooling* layer that simply computes the maximum of the activations of the neurons in the preceding layer. The same ideas can be easily generalized to two-dimensional and three-dimensional inputs.