American Journal of Epidemiology

Vol. 192, No. 11
https://doi.org/10.1093/aje/kwad107
Advance Access publication:
May 3, 2023

Practice of Epidemiology

Deep Learning for Epidemiologists: An Introduction to Neural Networks

Stylianos Serghiou* **and Kathryn Rough**

* Correspondence to Dr. Stylianos Serghiou, Prolaio, Inc., 6929 N. Hayden Road, Suite C4-441, Scottsdale, AZ 85250 (e-mail: dr.serghiou@gmail.com).

Initially submitted February 6, 2022; accepted for publication April 24, 2023.

Deep learning methods are increasingly being applied to problems in medicine and health care. However, few epidemiologists have received formal training in these methods. To bridge this gap, this article introduces the fundamentals of deep learning from an epidemiologic perspective. Specifically, this article reviews core concepts in machine learning (e.g., overfitting, regularization, and hyperparameters); explains several fundamental deep learning architectures (convolutional neural networks, recurrent neural networks); and summarizes training, evaluation, and deployment of models. Conceptual understanding of supervised learning algorithms is the focus of the article; instructions on the training of deep learning models and applications of deep learning to causal learning are out of this article's scope. We aim to provide an accessible first step towards enabling the reader to read and assess research on the medical applications of deep learning and to familiarize readers with deep learning terminology and concepts to facilitate communication with computer scientists and machine learning engineers.

artificial intelligence; deep learning; epidemiologic methods; machine learning; modeling; neural networks; prediction

Abbreviations: AI, artificial intelligence; CNN, convolutional neural network; FNN, feed-forward neural network; GRU, gated recurrent unit; LSTM, long short term memory; ReLU, rectified linear unit; RNN, recurrent neural network.

In 1998, researchers used longitudinal data on 7 carefully curated predictors from 5,345 individuals to build the Framingham Risk Score for 10-year risk of coronary heart disease (1). In the subsequent decades, it became one of the best known and most frequently utilized medical risk prediction models. Investigators prospectively collected data from a 12-year cohort study and fitted a Cox model to predictors identified using decades of domain-specific knowledge. Although not typically regarded as such, the Framingham Risk Score and many other popular scores (2–4) are early applications of machine learning in medicine (5).

Deep learning is a subset of machine learning methods, recent advancements that have led to breakthroughs in tasks that are not easily handled by more traditional methods, including image recognition (6, 7), language translation (8), text-to-speech generation (9), and text synthesis (10, 11). While some deep learning techniques were proposed in the 1980s, the increased availability of large data sets (12) and better computing resources (13) have led to dramatic performance improvements in recent years.

Deep learning techniques are increasingly being applied to tasks in health and medicine (14), although few have reached the stage of clinical implementation (15). In medicine, studies demonstrate that models can use chest x-rays to diagnose pneumonia on par with radiologists (16) and use electronic health records to predict acute kidney injury 2 days in advance (17), among many other applications (18, 19). In population health, Google (Mountain View, California) Street View can be used to predict the demographic make-up of US neighborhoods (20) and predict health outcomes (21). These results have led to excitement, despite few prospective evaluations and several methodological concerns (22–25).

With the abundance of research in this domain, it is important that epidemiologists and other health researchers can critically engage with and contribute to research using deep learning. This review offers an accessible introduction to the basics of deep learning from an epidemiologic perspective. It covers fundamental principles of machine learning, an explanation of common deep learning architectures,

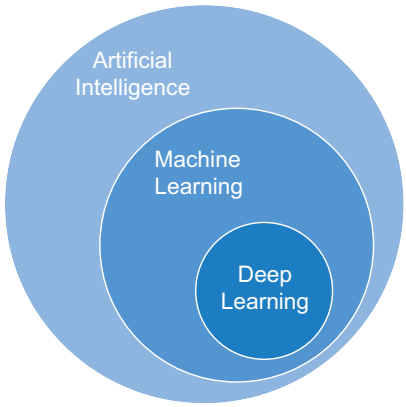

Figure 1. Deep learning is a subfield of machine learning, which is a subfield of artificial intelligence. This image was adapted from Goodfellow et al. (34, p. 9).

and summarizes training, evaluation, and deployment of models.

MACHINE LEARNING FUNDAMENTALS

In the mid-1980s, researchers developed DXplain, one of the first automated diagnostic decision support systems (26, 27) based on human-curated rules and existing information; for example, chest pain and shortness of breath could indicate myocardial infarction, pulmonary embolism, aortic dissection, or other conditions.

DXplain is an "expert system," a form of artificial intelligence (AI) outside the domain of machine learning (Figure 1). "Artificial intelligence" (28, 29) was a phrase coined in the 1950s to encompass a broad collection of machine abilities traditionally attributed to intelligent beings, including image recognition, text summarization, and commonsense reasoning. The term "AI" does not constrain the methods used to achieve these goals. Early work in the field of AI, including DXplain, focused on creating decision systems that followed hard-coded logical rules. In contrast, machine learning is a data-driven approach to AI that relies on "the ability to learn without being explicitly programmed" (30).

Most machine learning algorithms can be classified as supervised, unsupervised, or reinforcement learning approaches. In supervised learning, observations in the data set need "ground truth" labels, and the algorithm learns to identify patterns in the data that are indicative of a certain label. Unsupervised approaches learn useful properties of the data set, such as clustering (e.g., deriving ways of phenotyping sepsis (31)), without requiring any labels. Reinforcement learning is related to causal learning and uses trial and error to learn which actions generate the greatest rewards. Supervised learning algorithms are the focus of this article.

In supervised learning, each training example, or observation, in the data set has 2 key components: its features (e.g., "variables" or "predictors") and its label (e.g., "outcomes").

Table 1. Analogous Epidemiologic Terms for Common Machine Learning Vocabulary

Machine Learning Term[a]	Analogous Epidemiology Term or Concept
Training example	Observation, individual
Feature	Predictor, covariate, independent variable
Label	Outcome, response variable
Noisy labels	Outcome with measurement error
Feature engineering	Data preprocessing
Weights	Model coefficients
Bias term	Model intercept
Training	Model fitting
Training set	Derivation set
Bagging	Bootstrap model selection
Model output	Prediction
Sigmoid classifier	Logistic regression
Softmax classifier	Multinomial logistic regression
L1 regularization	LASSO regression
L2 regularization	Ridge regression
Confusion matrix	Contingency table, 2 ×2 table
Recall	Sensitivity
Precision	Positive predictive value

Abbreviation: LASSO, least absolute shrinkage and selection operator.

[a] See also, Table 2 in Mooney and Pejaver (94).

Based on internal parameters, the learning algorithm processes the features and produces an output, which can be compared with the ground-truth label. As the learning algorithm views more training examples, it adjusts its parameters to minimize the difference between its output and ground truth. These differences are quantified using a loss function. The further the model's output from ground truth, the greater the loss. A perfect model would have a loss of zero. This may sound familiar; linear and logistic regression are both supervised learning algorithms.

In addition to regression, numerous other machine learning algorithms fall outside the scope of deep learning, including support vector machines, naive-Bayes algorithms, and decision trees. (Bi et al. (32) provided a thorough review of these methods in an earlier issue of the *Journal*.) Terms used in the field of machine learning tend to vary from those used in the medical literature (Table 1).

DEEP LEARNING

Deep learning is a collection of machine learning methods, in which stacked processing layers are used to create abstract representations of data, creating an artificial "neural network" (33, 34). These processing layers form an interconnected path from input features (e.g., age, smoking status,

systolic blood pressure, etc.) to the model's output (e.g., risk of heart disease). Initially inspired by mechanistic theories of brain physiology, each layer consists of processing units called "neurons" or "nodes."

Neural networks may be considered a more flexible approach to fitting prediction models. Prediction modeling typically involves numerous preprocessing decisions: selecting a subset of the available predictors using prior knowledge or a statistical procedure (e.g., stepwise regression), discretizing continuous predictors, introducing polynomials, or using interaction terms. In deep learning, with enough data (which may vary from hundreds to millions of examples, depending on task complexity and approach), these preprocessing steps become unnecessary because of the flexibility gained by multiple processing layers and nonlinear data transformations, known as activation functions (33).

Structure

The building blocks of neural networks are neurons, which perform 2 simple operations. First, the neuron calculates a weighted sum of the inputs; these weights are randomly chosen at the start of model training and progressively revised to improve model performance (i.e., minimize loss) throughout the learning process. During the second step, the neuron applies a nonlinear mathematical transformation, an activation function, to that weighted sum.

While both operations are simple, neurons can be extremely powerful in aggregate. By adding more layers (and more neurons per layer), it is possible to model highly complex functions, including nonlinearities and interactions, without any further specification (Figure 2). The universal approximation theorem (35–37) demonstrates that a sufficiently deep (i.e., many layers) or wide (i.e., many neurons) neural network can approximate any continuous mathematical function.

In fact, a regression model can be expressed as a simple neural network. A network with an input layer, an output layer applying the logistic function, and no hidden layers (or hidden layers of neurons applying linear activation functions) will behave equivalently to logistic regression (Figure 3).

Activation functions

After the weighted sum of the inputs is calculated by the neuron, an activation function applies a mathematical transformation. In theory, both linear and nonlinear transformations may be used, but nonlinear functions are nearly always used to increase the network's capacity to model nonlinearities in the data.

The rectified linear unit (ReLU) is a simple activation function used extensively in deep learning. If the input is positive, it outputs the value of the input. If the output is negative or zero, it outputs zero.

Hyperparameters and training

"Hyperparameter" is a term used to describe any modifiable or "tunable" modeling choice. When using regression for predictive modeling, one could consider modeling choices, such as the use of higher-order terms or interaction terms, to be hyperparameters. In addition to increasing the complexity of the model structure, deep learning also increases the number of hyperparameters: the number of layers, the number of neurons in layers, parameters for regularization, and batch size (i.e., how many examples are shown to the model before updating its weights in training), among others. These hyperparameters define the structure of the neural network and dictate how it will be trained. Values of hyperparameters have a large impact on model performance and should be reported to enhance reproducibility.

The learning rate is among the most important hyperparameters. Unlike linear regression, loss functions cannot be formulaically minimized in deep learning models. Instead, an iterative approach known as gradient descent is commonly used. Gradient descent is a procedure that identifies the direction of steepest decrease in the local loss landscape, like water flowing downhill. In simple terms, calculating the gradient tells us in which direction we should adjust parameter values. The magnitude of the change made is partly determined by the learning rate hyperparameter. The learning rate is critical to the success of the gradient descent algorithm; too large a learning rate can result in a failure to

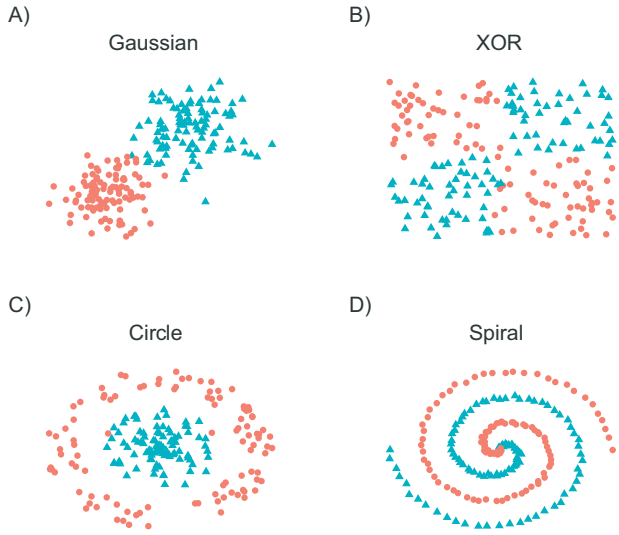

Figure 2. Classification in increasingly nonlinear data. Suppose that in panels A–D our goal is to classify each data point (dots and triangles) as belonging to either the pink group or the teal group based on its *x*- and *y*-coordinates. A regression-based classifier with a single parameter per predictor could classify data (A), but it would lack the capacity to model more complicated nonlinear functions (A–C). Its capacity could be increased by adding splines, polynomials, or interaction terms; yet, even with such approaches, performance in the spiral plot (D) would remain extremely poor. In deep learning models, capacity can be increased by adding more layers, adding more neurons per layer, changing the activation function, and changing other modeling choices collectively known as hyperparameters. You can interactively test the impact of such choices in Google Playground (https://playground.tensorflow.org/), from which the data were adapted.

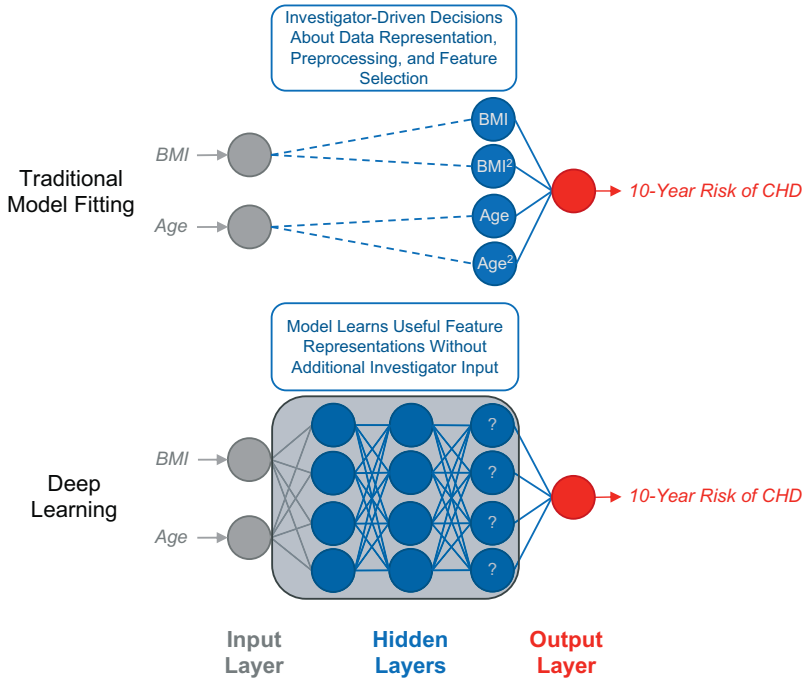

Figure 3. Traditional epidemiology vs. deep learning. The traditional process to data analysis in epidemiology is one in which the researcher uses a priori knowledge to engage in "feature discovery" by selecting, transforming, and modifying available data into the most appropriate features for the task at hand. The researcher then uses a method of choice, such as logistic regression, to fit a model. Deep learning differs in that the process of feature discovery can depend solely on the data at hand (i.e., it is data-driven) and is off-loaded to the neural network rather than being performed by the researcher. BMI, body mass index; CHD, coronary heart disease.

converge, and too small will make the model train slowly and inefficiently.

DEEP LEARNING ARCHITECTURES

Fully connected neural networks

Feed-forward neural networks (FNN), also known as fully connected neural networks, multilayer perceptrons, or dense neural networks, are the fundamental type of deep learning networks. They consist of 1 or more fully connected layers (Figure 4). In a fully connected layer, each neuron receives the output of all neurons from the previous layer. Based on learned weights, neurons calculate a weighted average of these inputs, apply an activation function, and propagate their output to neurons in the next layer.

Examples in health research. Avati et al. (38) used the electronic health records of 221,284 patients and 13,654 different features to predict all-cause mortality within the following year. To do so, they split the available data into training, validation, and test sets at a ratio of 8:1:1. They trained an FNN with 18 fully connected layers of 512 neurons each. The neurons in these layers used an activation function closely related to the ReLU, a scaled exponential linear unit (SELU). The output layer used a single neuron with a logistic activation function to output a probability. Their model correctly identified 1 in 3 deaths at the prespecified tolerance of 1 in 10 false alarms.

Convolutional neural networks

Convolutional neural networks (CNNs) are a family of deep learning models designed for images. However, they can also be applied to other data types (e.g., medical records) (39, 40). The networks have 3 core components: convolutional layers, pooling layers, and fully connected layers. These layers can be rearranged into different architectures of varying complexity.

Convolutional layers. For several reasons, FNNs are poorly suited to process images. They require an inefficiently large number of trainable parameters. Digital images are represented to machine learning models as grids of tens of thousands of pixels, each with a numerical red/green/blue (RGB) value. If each RGB value is a feature and each neuron is fully connected, the exponential increase in the number of weights from adding neurons and layers quickly becomes an issue. Further inefficiencies arise because weights learned by neurons are not shared in FNNs; if the network learns to locate an object of interest in one area of the frame, it will need to learn the pattern again if the object is shifted. Additionally, FNNs lack an inherent structure to compare a given pixel with the pixels around it and are only capable of processing images of a fixed size.

These issues motivated the creation of convolutional layers (Figure 5). Conceptually, convolutional layers recognize patterns across an image by maintaining a consistent but small number of weights. Each convolutional layer is a

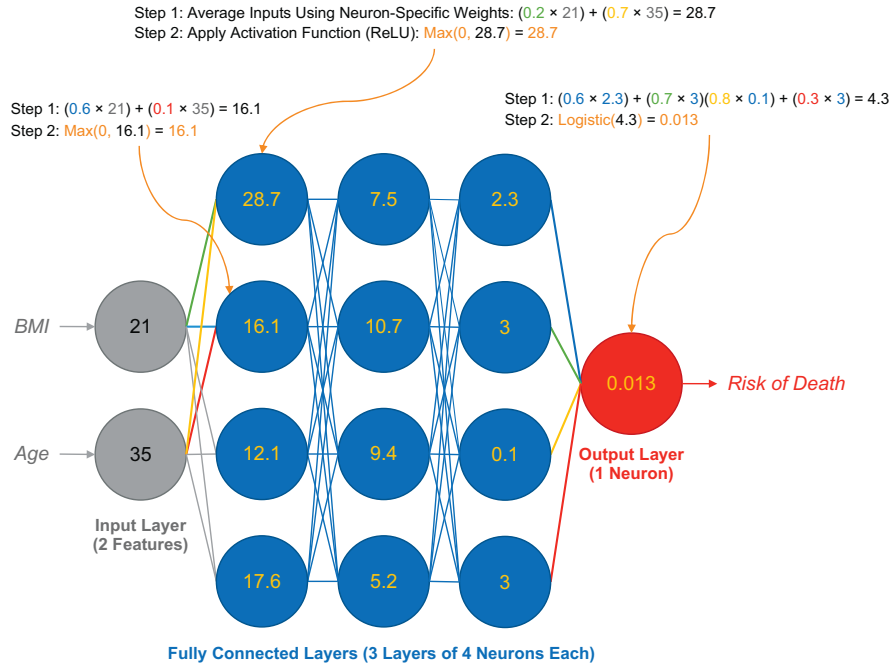

Figure 4. The basic architecture of feed-forward neural networks (FNNs). This figure demonstrates an FNN that predicts mortality based on age (in years) and the body mass index (BMI, calculated as weight (kg)/height (m)^2). FNNs consist only of fully connected layers. This figure demonstrates 3 fully connected layers, each of which consists of 4 neurons (in blue). Starting from the input layer (in gray), each neuron receives an input, takes a weighted average of that input using arrow-specific weights, applies an activation function (in orange; here we demonstrate a rectified linear unit (ReLU): max(0, x)) and propagates the activated weighted average to each of the neurons of the next hidden layer. The depicted FNN is a 4-layer neural network because it consists of 4 layers of learned parameters (3 hidden layers, 1 output layer). Note that there can be more than 1 node in the output layer in what is known as "multitask learning."

square (e.g., 1×1, 3×3, 5×5) of learned weights, known as the "filter" or "kernel". The filter is first applied at the top left of an image, and each weight in the filter (e.g., 9 weights for a 3×3 filter) corresponds to a pixel. Each convolutional layer can have multiple filters, the same way a fully connected layer can have multiple neurons. A pixel's value is multiplied by the corresponding weight and then summed to create a weighted average representation. The filter then moves to the right by a prespecified number of pixels or "stride," and repeats the process of multiplying pixel values by weights. Continuing this process across the whole image is a "convolution." As with fully connected layers, an activation function is applied to the output of the filter before being fed to the next layer.

Convolutions create "translation invariant" representations, meaning their output is consistent regardless of where in the picture the object may be. Often, they are conceptualized as "feature detectors" because they can represent specific features: Early layers detect basic features, including edges and outlines of shapes; later layers detect more complex features, such as the eyes or the mouth of a face.

Pooling layers. Convolutional layers are interspersed with pooling layers, which aggregate information across rectangular "neighborhoods" of an image. This reduces the size of the representation and helps neural networks identify specific features, regardless of their location within the image; they play a critical role in the performance of CNNs (41). Typically, pooling layers will take the maximum value of the neighborhood ("max pooling"), but they can also take the average ("average pooling"). Unlike most other types of layers discussed, no activation function is applied to the output of pooling layers, nor do they contain any learned parameters.

Examples in health research. In 2017, Esteva et al. (42) published a paper evaluating the ability of a CNN to classify skin lesion photographs into the risk of having each one of 2,032 different dermatologic diseases. Authors found the model achieved performance on par with 21 board-certified dermatologists in prediction of keratinocyte carcinomas versus benign seborrheic keratoses and malignant melanomas versus benign nevi.

The study used a previously developed CNN architecture, Inception-v3 (43). The network contained repeated convolution and pooling layers, and had been originally trained to classify ImageNet, a nonmedical data set of over 14 million photographs from more than 21,000 categories (12, 44). Esteva et al. leveraged this "pretrained" model and refitted the final fully connected layers using the dermatology images and labels in their data set. Repurposing pretrained CNNs for medical tasks is a frequently used strategy known as "transfer learning"; otherwise, training performant CNN architectures from scratch can require enormous resources.

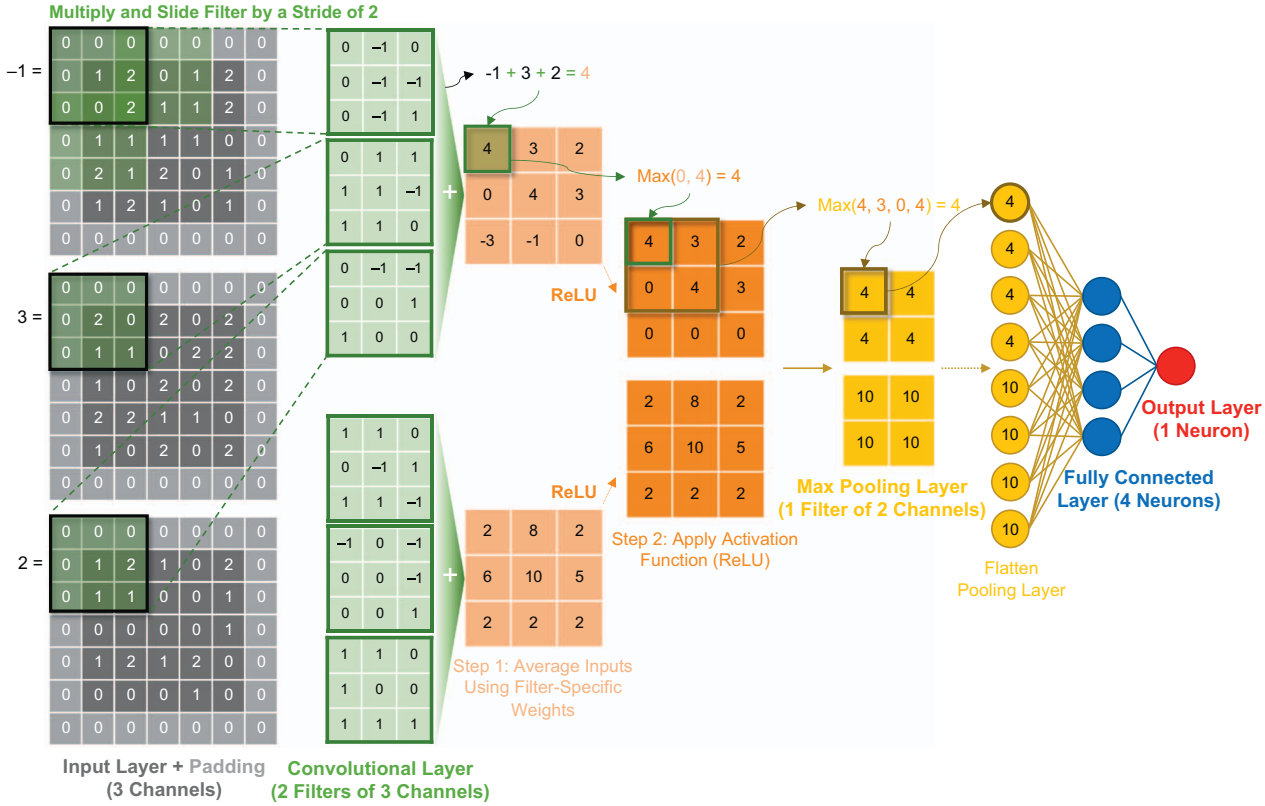

Figure 5. The basic architecture of convolutional neural networks (CNNs). CNNs are characterized by their inclusion of convolutional (in green) and pooling (in yellow) layers. Each of these layers consists of 1 or more filters (= kernels) of identical dimensions and of as many channels as the input (in gray). The input in this figure consists of 3 channels, in reminiscence of the standard red/green/blue (RGB) channels of an image (but the input does not need to be an image and does not need to be constrained to 3 channels). Each channel of each filter is applied to the top left corner of the input, each input is multiplied by its respective weight (i.e., term-by-term), and all values are then summed across channels. The filter is then moved by a predetermined stride (i.e., the number of squares by which the filter will move) to the right and down to cover the whole input. A padding of zeros is used so that the filter always perfectly fits into the input. An activation function is then applied to each of the totals (in this case, ReLU is applied to each of the 9 totals of each filter), filter-specific outputs are stacked onto each other (such that the output of each filter now represents a new channel) and the stacked output is propagated to the next layer. A convolutional layer is typically followed by a pooling layer (in this case, max pooling), which is applied across the output of the convolutional layer in a similar fashion as described above. Note that no activation is applied after max pooling. Finally, it is common to complete a CNN architecture with one or more fully connected layers (blue), where each rectangle of the max pooling output represents a single feature input to the fully connected layer. Image adapted from the CS231n course at Stanford University (95). ReLU, rectified linear unit.

Recurrent neural networks

There are many tasks where the order of model inputs matters, such as the sequence of words in natural language processing or the sequence of notes in music recognition. Recurrent neural networks (RNNs) process data in a sequential fashion, achieving better-than-human ability in tasks such as speech recognition (45).

Recurrent layer. In its simplest form, a recurrent layer is a fully connected layer that feeds into itself; outputs at one time step become inputs at the next time step (Figure 6). Take for example the phrase "the quick brown fox jumps over the lazy dog." We first create a mathematical representation for each word. A simple representation could be an indicator vector (i.e., a vector where the position corresponding to the specific word equals 1 and all other positions to 0). However, the indicator method is inefficient. Words can alternatively be represented as embeddings, where words that are similar are represented by vectors that are close to one another in vector space. Using embeddings tends to improve model performance, and many of these embeddings are open-sourced and freely available (10, 46, 47).

A basic RNN unit consists of a single layer and an activation function, and it will process the sequence from left to right. The vector representing the first word, "the," is propagated through the layer, and an activation function is applied. The output will be concatenated with the vectorized representation of "quick," passed through the same network, and the output will again be concatenated with the representation of the next word. This process continues as the model iterates through the entire sequence. Notice that the RNN unit remains unchanged; we are simply looping through the

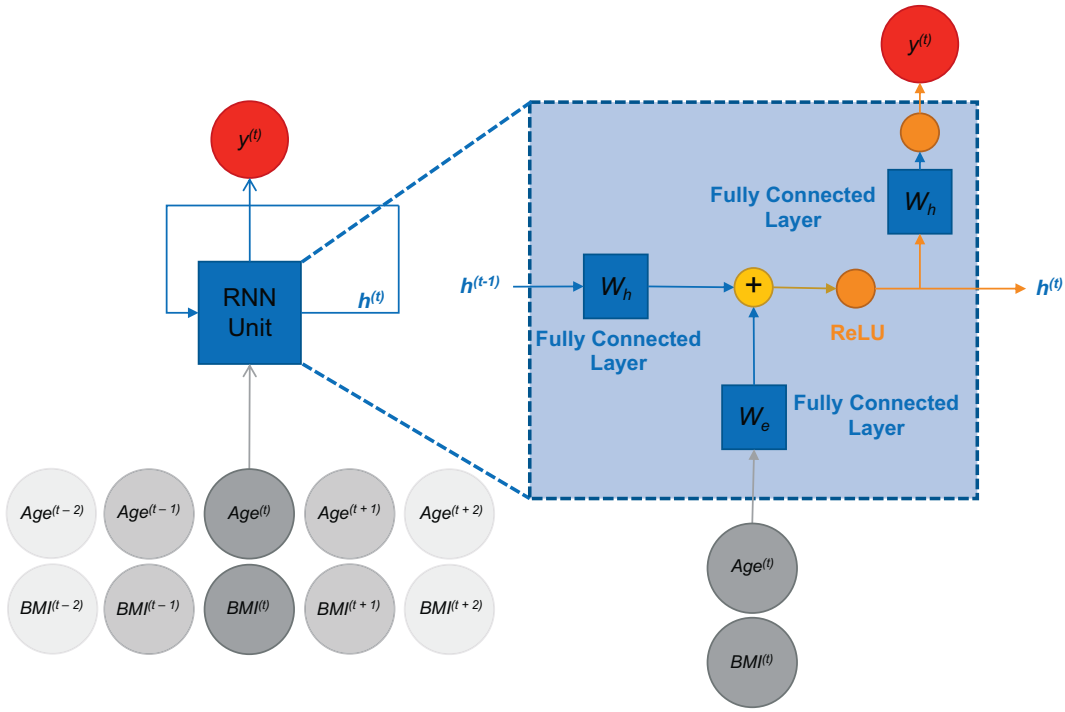

Figure 6. The basic architecture of recurrent neural networks (RNNs). RNNs are characterized by their inclusion of recurrent layers. For each point in time t, there is an input or embedding (in gray), a hidden state, $h^{(t)}$ (in blue), and an output observed at time t, $y^{(t)}$ (in red) (in this example, risk of death at time t). Each recurrent layer takes a weighted average of input from the previous time point, $h^{(t-1)}$, adds it to a weighted average of the input from the current time point, adds a bias term, applies an activation function (e.g., ReLU, in orange) and propagates the activated weighted average, $h^{(t)}$, to the next time point (adapted from Jeewandara (96)). BMI, body mass index; ReLU, rectified linear unit.

inputs. Predictions or inferences can be made using outputs of the model at any stage (i.e., a prediction can be made after each item in the sequence or after the entire sequence has been processed).

Simple RNN units often fail to retain key contextual information that occurred earlier in the sequence (48). Instead, many RNNs use other types of units, such as gated recurrent units (GRU) (49) or long short term memory (LSTM) units (50). LSTMs have an additional "memory" state that allows storage of information from previous hidden states, in addition to the hidden state maintained by the simple RNN unit. At each subsequent time step along the sequence, the LSTM takes 2 inputs: the hidden state from the previous time step and the values of the predictors at the current time step. Using these inputs, it determines what new information to retain, what old information to forget, and what information to pass along to the next time step as an output. This last value is the new hidden state at the current time point and captures information that was seen recently, as well as several time steps before.

Examples in health research. Use of RNN architectures has often led to higher performance than standard feed-forward networks (14). Choi et al. (51) used an RNN to predict the risk of being diagnosed with heart failure in a matched case-control sample within the next 12 months. Embeddings were used to capture clinical events in the timeline of a patient (e.g., being diagnosed with pneumonia, having a chest x-ray, or being prescribed amoxicillin). A GRU was used to propagate through each event in the sequence it was recorded. At the end of the sequence of clinical events, the model output was a probability of heart failure.

Modern directions. There has been substantial work on alternative architectures for sequences that mitigate some of the limitations of RNNs. RNNs cannot organically represent events occurring at varying or irregular time intervals; recent work proposed RNNs that work on a continuum, rather than discrete event units (52). Even LSTMs and GRUs have limited capacity to capture long-term dependencies; transformer models use a mechanism called "attention" (53) to simultaneously process all sequence elements, which has led to substantial performance improvements (8, 10).

Alternative architectures

CNNs and RNNs represent only a portion of deep learning architectures. Other models include deep generative models (54), Bayesian neural networks (55, 56), graphical models (57), and general adversarial networks (58–60), although we are unable to describe these in detail in this review.

MODEL FITTING

Deep learning models can have substantial capacity. Given data and labels, the network can "discover" intermediate representations that facilitate translation of a given input into the desired output, referred to as "end-to-end" learning. Neural networks can use, combine, and create intermediate representations of features through learned weights. Figuring out which models are "optimal" typically requires partitioning the data, fitting multiple models with different specifications, and choosing the best-performing model according to a metric of interest.

Training, validation, and test sets

Using the same data to fit a model and evaluate it will lead to an overoptimistic estimate of performance in the target population. To prevent this, machine learning typically splits the available data into 3 sets: the training, validation, and test sets (the training-validation split can be avoided if cross-validation is used). The split ratio can vary depending on the size of the data set (e.g., 60:20:20 for 10–10,000 examples, vs. 98:1:1 for 1 million examples). The training set can be used to train a variety of models. Model performance is compared by measuring performance on the validation set. The "held-out" test set should be used only once, to assess performance after final model selection.

Hyperparameter tuning

There is limited theoretical understanding of which modeling choices will work well for a given machine learning task. Choosing hyperparameters often requires iterative experimentation, a process known as tuning. Candidate values can be found through random search (values randomly selected independently of one another) or Bayesian hyperparameter optimization (61) (conditioning on the observed performance of previous hyperparameter combinations to inform which combination of hyperparameters should be tested next).

Regularization

The ultimate goal of fitting a model is to generalize well to a target population that has not yet been "seen" by the model. Models with sufficient capacity can "memorize" training data—they fit to noise rather than signal—and become "overfitted." In contrast, models with inadequate capacity will poorly model the underlying function; this is often referred to as "underfitting." Overfitting is diagnosed when a model performs very well in the training set but poorly in the validation set. Underfitted models will perform poorly both in the training and validation sets.

Regularization helps to minimize overfitting by trading a moderate increase in model bias with a large decrease in model variance (62). There are several methods to regularize models; the most widely used approaches help avoid large weights by adding the absolute value of each weight (L1 regularization), or their squares (L2 regularization), to the loss function. L1 regularization also helps drive small weights to 0, enabling feature selection (i.e., ignoring some features). In addition to these methods, deep learning utilizes many other methods of regularization, such as adding random noise (63) to the inputs, dropout (64), and batch normalization (65).

Performance metrics

A number of metrics can be used to quantify the performance of the model, and appropriate performance metrics will depend on the task. For binary classification tasks, common metrics include recall (sensitivity in epidemiologic literature), precision (positive predictive value in epidemiologic literature), accuracy, area under the receiver operating curve (AUROC), area under the precision-recall curve (AUPRC), and calibration.

AUPRC is a particularly useful metric for measuring performance when there is a large imbalance in the prevalence of different outcome labels (66, 67). It is a measure of average positive predictive value, across all values of sensitivity, and varies in value from the prevalence of the outcome in the sample (no predictive ability) to 1 (perfect predictive ability).

Calibration measures how well the expected risk corresponds to the observed risk and may be assessed using a calibration curve, the Greenwood-Nam-D'Agostino test (68) (a modified version of the Hosmer-Lemeshow χ^2 statistic), or the Brier score.

Reporting multiple metrics with confidence intervals and a clinically meaningful interpretation of the result is generally good practice. On their own, standard machine learning metrics do not quantify potential benefits or harms to the patient.

Approach to using deep learning in your research

For minimal code or code-free approaches to fitting neural networks, readers can consider tools such as the h2o package (69) in R (R Foundation for Statistical Computing, Vienna, Austria). Readers interested in a user-friendly package to train neural networks or reuse pretrained neural networks can consider the Keras package (70) in Rand relevant introductory tutorials (71). For those interested in engaging with the field more deeply, consider reading *Deep Learning* (https://www.deeplearningbook.org) (34) or enrolling in freely available web courses through such services as Deep Learning.AI (https://www.deeplearning.ai/) (72). Table 2 provides additional educational resources.

LIMITATIONS AND REAL-WORLD CHALLENGES

Limitations of deep learning in epidemiology

Even though the limitations of deep learning have been extensively reviewed elsewhere (73, 74), limitations in training, interpretability, and generalizability are particularly relevant to epidemiologists.

First, deep learning tends to thrive in settings where outcomes depend on nonlinear combinations of thousands of variables, and hundreds of thousands of examples (or more) are available for training. It has been successful in

Table 2. Fundamental Resources in Deep Learning

Resource	Description	Link
Books		
Deep Learning	Introduction to the fundamentals of deep learning	http://www.deeplearningbook.org
Dive Into Deep Learning	An applied approach to deep learning with example code	https://d2l.ai/index.html
Online resources		
Chris Oha's blog	Accessible overview of foundational topics in deep learning	https://colah.github.io/
deeplearning.ai	7-day trial, $49/month after that for the Coursera subscription (financial aid available)	https://www.deeplearning.ai/
fast.ai	Free of charge	https://www.fast.ai/
Machine Learning Crash Course With TensorFlow APIs	Free of charge	https://developers.google.com/machine-learning/crash-course/
Learn by Kaggle	Free of charge	https://www.kaggle.com/learn/overview
Deep learning frameworks		
Keras	Available for R (https://www.r-project.org/), Python (https://www.python.org/), and Julia (https://julialang.org/). An accessible framework to build neural networks with minimal code	https://keras.io/
TensorFlow	Available for R and Python	https://www.tensorflow.org/
PyTorch	Available for R and Python	https://pytorch.org/

image classification, where data sets like ImageNet contain 14 million images (12) and the typical image is cropped into 256 × 256 pixels (i.e., 65,536 pixels, where each pixel is a model input). It has shown less benefit in the typical epidemiologic setting, where data sets have hundreds to thousands of examples, and outcomes may be largely explained by tens of variables (75–77). However, with bigger data sets, more complex data types (e.g., text, images), and novel data-efficient training methods (e.g., transfer learning, semi-supervised learning, creation of synthetic data), deep learning can be a useful tool in epidemiology (78–82).

Second, deep learning predictions are difficult to interpret due to their use of nonlinear combinations of variables. In contrast, linear and logistic regression have coefficients that are more readily interpretable. However, there are several interpretability methods that can be applied to deep learning. One commonly used method quantifies feature saliency by occlusion, meaning that the impact of a given area of an image on the model output is measured by replacing it with a gray square (83). Tomašev et al. (17) generalized this method to electronic health record data in research to predict acute kidney injury, identifying baseline creatinine, 48-hour creatinine, and serum calcium as the most predictive features. Other interpretability approaches include analysis of weights, activations, gradients, and attention (84).

Third, deep learning models can have limited generalizability. The ability of deep learning models to capture complex predictive relationships can identify patterns unique to the population or setting that gave rise to the training data. This can increase model performance in a specific population while decreasing generalizability to other populations (85). While the regularization approaches described above can improve external validity, prospective validation studies are necessary to understand the generalizability of deep learning algorithms.

Practical challenges for real-world use of deep learning

The use of machine learning–enabled technologies in real-world clinical settings presents numerous practical challenges (73).

First, model performance tends to degrade with time, and the performance originally measured in the test set overestimates real-world performance. This phenomenon is known as the "training-serving skew." We can view this as a failure of the model to generalize to the population it is currently being used in (a lack of external validity). This can have several root causes, including temporal shifts in the input features (e.g., changes in patient behavior or in clinical or operational practices) or differences between the population included in the training set versus actual users of the technology. Continuous performance monitoring is important for detection of performance degradation, and deployed deep learning models are often retrained on fairly regular schedules to mitigate this issue.

Second, the use of machine learning in clinical settings also has the potential to propagate or increase existing disparities in health care (86). Ensuring fairness requires a holistic approach (87), including consideration of biases in formulation of the machine learning task, training set composition, labeling of observations, nonrandom missingness of data, and real-world use of the models.

Third, the widespread availability of sophisticated machine learning may have privacy implications, particularly for deidentified data in the public domain (88). Research has demonstrated that some genetic and aggregated accelerometer data are reidentifiable using machine learning (89, 90). In contrast, deep learning–based methodologies have also been applied to improve and scale deidentification of free-text medical records (91, 92). Federated learning, a technique that allows the training of machine learning models in a decentralized fashion, may also decrease the need for data sharing.

CONCLUSION

In a 1970 commentary, physician William Schwartz mused, "Indeed, it seems probable that in the not too distant future the physician and the computer will engage in frequent dialogue, the computer continuously taking note of history, physical findings, laboratory data, and the like, alerting the physician to the most probable diagnoses and suggesting the appropriate, safest course of action" (93, p. 1258). While this vision has certainly not been realized, deep learning may enable tooling that facilitates parts of it. Deep learning presents real opportunities for improving the quality of care provided to patients, as well as numerous challenges.

We hope this review has provided you with an accessible first step towards building the foundation needed to engage with research that uses deep learning, either as a collaborator, reviewer, or critical reader. Epidemiologists have a role to play in the development of these technologies, particularly in measuring their real-world impact and safety.

ACKNOWLEDGMENTS

Author affiliations: Prolaio, Inc., Scottsdale, Arizona, United States (Stylianos Serghiou); Meta-Research Innovation Center at Stanford, School of Medicine, Stanford University, Stanford, California, United States (Stylianos Serghiou); and Global Epidemiology and Outcomes Research, IQVIA Germany, Frankfurt, Hessen, Germany (Kathryn Rough).

This work was supported by Google, LLC, Mountain View, California. Stylianos Serghiou and Kathryn Rough were employed by Google, LLC, at the time this article was originally drafted.

The data and code used to produce Figure 2 can be found at https://github.com/serghiou/deep-learning-for-epidemiologists. No other raw data or code was used in preparing this manuscript.

We thank Dr. Michael Howell (Google) for his feedback on earlier versions of this work.

An earlier version of this work was presented at the annual meeting of the Society for Epidemiologic Research, June 18–21, 2019, Minneapolis, Minnesota.

An earlier version of this work was published online. Serghiou S, Rough K. Deep learning for epidemiologists: an introduction to neural networks. *arXiv*. 2022. (https://doi.org/10.48550/arXiv.2202.01319).

Conflict of interest: none declared.

REFERENCES

1. Wilson PW, D'Agostino RB, Levy D, et al. Prediction of coronary heart disease using risk factor categories. *Circulation.* 1998;97(18):1837–1847.
2. Knaus WA, Wagner DP, Draper EA, et al. The APACHE III prognostic system. Risk prediction of hospital mortality for critically ill hospitalized adults. *Chest.* 1991;100(6):1619–1636.
3. Lip GYH, Nieuwlaat R, Pisters R, et al. Refining clinical risk stratification for predicting stroke and thromboembolism in atrial fibrillation using a novel risk factor-based approach: the Euro Heart Survey on Atrial Fibrillation. *Chest.* 2010;137(2):263–272.
4. Wells PS, Ginsberg JS, Anderson DR, et al. Use of a clinical model for safe management of patients with suspected pulmonary embolism. *Ann Intern Med.* 1998;129(12):997–1005.
5. Beam AL, Kohane IS. Big data and machine learning in health care. *JAMA.* 2018;319(13):1317–1318.
6. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, et al., eds. *Advances in Neural Information Processing Systems 25*. Red Hook, NY: Curran Associates, Inc.; 2012:1097–1105.
7. He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks. In: *Computer Vision—ECCV 2016*. Cham, Switzerland: Springer International Publishing; 2016:630–645.
8. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Guyon I, Luxburg UV, Bengio S, et al., eds. *Advances in Neural Information Processing Systems 30*. Red Hook, NY: Curran Associates, Inc.; 2017:5998–6008.
9. van den Oord A, Dieleman S, Zen H, et al. WaveNet: a generative model for raw audio [preprint]. *arXiv.* 2016. (https://doi.org/10.48550/arXiv.1609.03499). Accessed November 18, 2022.
10. Devlin J, Chang M-W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [preprint]. *arXiv.* 2018. (https://doi.org/10.48550/arXiv.1810.04805). Accessed November 18, 2022.
11. Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners. https://life-extension.github.io/2020/05/27/GPT%E6%8A%80%E6%9C%AF%E5%88%9D%E6%8E%A2/language-models.pdf. Accessed November 29, 2022.
12. Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009:248–255.
13. Krizhevsky A. One weird trick for parallelizing convolutional neural networks [preprint]. *arXiv.* 2014. (https://doi.org/10.48550/arXiv.1404.5997). Accessed November 18, 2022.
14. Xiao C, Choi E, Sun J. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *J Am Med Inform Assoc.* 2018;25(10):1419–1428.

15. Shah P, Kendall F, Khozin S, et al. Artificial intelligence and machine learning in clinical development: a translational perspective. *NPJ Digit Med*. 2019;2(1):69.
16. Rajpurkar P, Irvin J, Zhu K, et al. CheXNet: radiologist-level pneumonia detection on chest x-rays with deep learning [preprint]. *arXiv*. 2017. (https://doi.org/10.48550/arXiv.1711.05225). Accessed November 18, 2022.
17. Tomašev N, Glorot X, Rae JW, et al. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*. 2019;572(7767):116–119.
18. Ravi D, Wong C, Deligianni F, et al. Deep learning for health informatics. *IEEE J Biomed Health Inform*. 2017; 21(1):4–21.
19. Topol EJ. High-performance medicine: the convergence of human and artificial intelligence. *Nat Med*. 2019;25(1):44–56.
20. Gebru T, Krause J, Wang Y, et al. Using deep learning and Google Street View to estimate the demographic makeup of neighborhoods across the United States. *Proc Natl Acad Sci U S A*. 2017;114(50):13108–13113.
21. Nguyen QC, Khanna S, Dwivedi P, et al. Using Google Street View to examine associations between built environment characteristics and U.S. health outcomes. *Prev Med Rep*. 2019;14:100859.
22. Goldstein BA, Navar AM, Pencina MJ, et al. Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review. *J Am Med Inform Assoc*. 2017;24(1):198–208.
23. Chen JH, Asch SM. Machine learning and prediction in medicine—beyond the peak of inflated expectations. *N Engl J Med*. 2017;376(26):2507–2509.
24. Emanuel EJ, Wachter RM. Artificial intelligence in health care: will the value match the hype? *JAMA*. 2019;321(23): 2281–2282.
25. Rajkomar A, Dean J, Kohane I. Machine learning in medicine. *N Engl J Med*. 2019;380(14):1347–1358.
26. Octo Barnett G, Cimino JJ, Hupp JA, et al. DXplain: an evolving diagnostic decision-support system. *JAMA*. 1987; 258(1):67–74.
27. Miller RA. Medical diagnostic decision support systems—past, present, and future: a threaded bibliography and brief commentary. *J Am Med Inform Assoc*. 1994;1(1): 8–27.
28. McCarthy J, Minsky ML, Shannon CE, et al. *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence*. 1955. http://jmc.stanford.edu/articles/dartmouth/dartmouth.pdf. Accessed April 24, 2023.
29. McCarthy J. Recursive functions of symbolic expressions and their computation by machine. Part I. *Commun ACM*. 1960; 3(4):184–195.
30. Samuel AL. Some studies in machine learning using the game of checkers. *IBM J Res Dev*. 1959;3(3):210–229.
31. Seymour CW, Kennedy JN, Wang S, et al. Derivation, validation, and potential treatment implications of novel clinical phenotypes for sepsis. *JAMA*. 2019;321(20): 2003–2017.
32. Bi Q, Goodman KE, Kaminsky J, et al. What is machine learning? A primer for the epidemiologist. *Am J Epidemiol*. 2019;188(12):2222–2239.
33. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015; 521(7553):436–444.
34. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. 1st ed. Cambridge, MA: MIT Press; 2016.
35. Cybenko G. Approximation by superpositions of a sigmoidal function. *Math Control Signals Syst*. 1989;2(4):303–314.
36. Leshno M, Lin VY, Pinkus A, et al. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw*. 1993;6(6): 861–867.
37. Hanin B, Sellke M. Approximating continuous functions by ReLU nets of minimal width [preprint]. *arXiv*. 2017. (https://doi.org/10.48550/arXiv.1710.11278). Accessed November 18, 2022.
38. Avati A, Jung K, Harman S, et al. Improving palliative care with deep learning. *BMC Med Inform Decis Mak*. 2018; 18(suppl 4):122.
39. Razavian N, Marcus J, Sontag D. Multi-task prediction of disease onsets from longitudinal laboratory tests. *Proc Mach Learn Res*. 2016;56:73–100.
40. Yang Z, Huang Y, Jiang Y, et al. Clinical assistant diagnosis for electronic medical record based on convolutional neural network. *Sci Rep*. 2018;8(1):6329.
41. Boureau Y-L, Ponce J, LeCunn Y. A theoretical analysis of feature pooling in visual recognition. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. 2010:111–118.
42. Esteva A, Kuprel B, Novoa RA, et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*. 2017;542(7639):115–118.
43. Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016:2818–2826.
44. State-of-the-Art: image classification on ImageNet. 2020. https://paperswithcode.com/sota/image-classification-on-imagenet. Accessed, July 17, 2020.
45. Shoham Y, Perrault R, Brynjolfsson E, et al. Artificial intelligence index: 2017 annual report. 2017. https://hai.stanford.edu/ai-index-2017. Accessed November 29, 2022.
46. Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Burges CJC, Bottou L, Welling M, et al., eds. *Advances in Neural Information Processing Systems 26*. Red Hook, NY: Curran Associates, Inc.; 2013: 3111–3119.
47. Pennington J, Socher R, Manning C. GloVe: global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014:1532–1543.
48. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Netw*. 1994;5(2):157–166.
49. Cho K, van Merrienboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014:1724–1734.
50. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735–1780.
51. Choi E, Schuetz A, Stewart WF, et al. Using recurrent neural network models for early detection of heart failure onset. *J Am Med Inform Assoc*. 2017;24(2):361–370.
52. Rubanova Y, Chen RTQ, Duvenaud D. Latent ODEs for irregularly-sampled time series. In: *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*. 2019. https://papers.nips.cc/paper_files/paper/2019/file/42a6845a557bef704ad8ac9cb4461d43-Paper.pdf. Accessed April 24, 2023.
53. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: *International Conference on Learning Representations (ICLR)*. 2015. https://iclr.cc/archive/www/lib/exe/fetch.php%3Fmedia=iclr2015:bahdanau-iclr2015.pdf. Accessed April 24, 2023.

54. Oussidi A, Elhassouny A. Deep generative models: survey. *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*. 2018. https://doi.org/10.1109/isacv.2018.8354080. Accessed April 24, 2023.
55. Bishop CM. *Bayesian Methods for Neural Networks*. Birmingham, UK. Aston University; 1995.
56. Mullachery V, Khera A, Husain A. Bayesian neural networks [preprint]. *arXiv*. 2018. (https://doi.org/10.48550/arXiv.1801.07710). Accessed November 18, 2022.
57. Johnson MJ, Duvenaud DK, Wiltschko A, et al. Composing graphical models with neural networks for structured representations and fast inference. In: Lee DD, Sugiyama M, Luxburg UV, et al., eds. *Advances in Neural Information Processing Systems 29*. Red Hook, NY: Curran Associates, Inc; 2016:2946–2954.
58. Senior AW, Evans R, Jumper J, et al. Improved protein structure prediction using potentials from deep learning. *Nature*. 2020;577(7792):706–710.
59. Jordon J, Yoon J, van der Schaar M. PATE-GAN: generating synthetic data with differential privacy guarantees. In: *International Conference on Learning Representations*. 2018. https://openreview.net/pdf?id=S1zk9iRqF7. Accessed April 24, 2023.
60. Yahi A, Vanguri R, Elhadad N, et al. Generative adversarial networks for electronic health records: a framework for exploring and evaluating methods for predicting drug-induced laboratory test trajectories. In: *Neural Information Processing Systems: Machine Learning for Health (NeurIPS ML4H)*. 2017. https://arxiv.org/abs/1712.00164. Accessed April 24, 2023.
61. Bergstra J, Yamins D, Cox DD. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning*. 2013:115–123.
62. Belkin M, Hsu D, Ma S, et al. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proc Natl Acad Sci.* 2019;116(32):15849–15854.
63. Bishop CM. Training with noise is equivalent to Tikhonov regularization. *Neural Comput.* 1995;7(1):108–116.
64. 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):1929–1958.
65. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning, Volume 37*. 2015:448–456.
66. Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*. 2006:233–240.
67. Leisman DE. Rare events in the ICU: an emerging challenge in classification and prediction. *Crit Care Med.* 2018;46(3):418–424.
68. D'Agostino RB, Nam B-H. Evaluation of the performance of survival analysis models: discrimination and calibration measures. *Handbook of Stat.* 2003;23:1–25.
69. LeDell E, Poirier S. H2O AutoML: scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*. 2020. https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf. Accessed November 29, 2022.
70. Chollet F, Allaire J, Falbel D, et al. R interface to Keras. *Keras Team.*2017; https://github.com/rstudio/keras. Accessed on April 21, 2020.
71. Arnold T. R Interface to the Keras Deep Learning Library. *kerasR*. https://cran.r-project.org/web/packages/kerasR/vignettes/introduction.html. Accessed July 26, 2022.
72. DeepLearning.AI. Courses. 2022; https://www.deeplearning.ai/courses/. Accessed November 20, 2022.
73. Kelly CJ, Karthikesalingam A, Suleyman M, et al. Key challenges for delivering clinical impact with artificial intelligence. *BMC Med.* 2019;17(1):195.
74. Ghassemi M, Naumann T, Schulam P, et al. A review of challenges and opportunities in machine learning for health. *AMIA Jt Summits Transl Sci Proc.* 2020;2020:191–200.
75. Puddu PE, Menotti A. Artificial neural network versus multiple logistic function to predict 25-year coronary heart disease mortality in the Seven Countries Study. *Eur J Cardiovasc Prev Rehabil.* 2009;16(5):583–591.
76. Puddu PE, Menotti A. Artificial neural networks versus proportional hazards Cox models to predict 45-year all-cause mortality in the Italian rural areas of the Seven Countries Study. *BMC Med Res Methodol.* 2012;12(1):100.
77. Christodoulou E, Ma J, Collins GS, et al. A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *J Clin Epidemiol.* 2019;110:12–22.
78. Ouali Y, Hudelot C, Tami M. An overview of deep semi-supervised learning [preprint]. *arXiv*. 2020. (https://doi.org/10.48550/arXiv.2006.05278). Accessed November 18, 2022.
79. Zhuang F, Qi Z, Duan K, et al. A comprehensive survey on transfer learning. *Proc IEEE.* 2021;109(1):43–76.
80. Aggarwal R, Sounderajah V, Martin G, et al. Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis. *NPJ Digit Med.* 2021;4(1):65.
81. Si Y, Du J, Li Z, et al. Deep representation learning of patient data from electronic health records (EHR): a systematic review. *J Biomed Inform.* 2021;115:103671.
82. Xie F, Yuan H, Ning Y, et al. Deep learning for temporal data representation in electronic health records: a systematic review of challenges and methodologies. *J Biomed Inform.* 2022;126:103980.
83. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: *Computer Vision—ECCV 2014*. Cham, Switzerland: Springer International Publishing; 2014:818–833.
84. Molnar C. *Interpretable Machine Learning*. 1st ed. Victoria, Canada: Leanpub; 2020.
85. Goodman SN, Goel S, Cullen MR. Machine learning, health disparities, and causal reasoning. *Ann Intern Med.* 2018;169(12):883–884.
86. Obermeyer Z, Powers B, Vogeli C, et al. Dissecting racial bias in an algorithm used to manage the health of populations. *Science.* 2019;366(6464):447–453.
87. Rajkomar A, Hardt M, Howell MD, et al. Ensuring fairness in machine learning to advance health equity. *Ann Intern Med.* 2018;169(12):866–872.
88. Murdoch B. Privacy and artificial intelligence: challenges for protecting health information in a new era. *BMC Med Ethics.* 2021;22(1):122.
89. Schadt EE, Woo S, Hao K. Bayesian method to predict individual SNP genotypes from gene expression data. *Nat Genet.* 2012;44(5):603–608.
90. Na L, Yang C, Lo C-C, et al. Feasibility of Reidentifying individuals in large National Physical Activity Data Sets from which protected health information has been removed with use of machine learning. *JAMA Netw Open.* 2018;1(8):e186040.

91. Ahmed T, Aziz MMA, Mohammed N. De-identification of electronic health record using neural network. *Sci Rep.* 2020; 10(1):18600.
92. Murugadoss K, Rajasekharan A, Malin B, et al. Building a best-in-class automated de-identification tool for electronic health records through ensemble learning. *Patterns (N Y).* 2021;2(6):100255.
93. Schwartz WB. Medicine and the computer. The promise and problems of change. *N Engl J Med.* 1970;283(23): 1257–1264.
94. Mooney SJ, Pejaver V. Big data in public health: terminology, machine learning, and privacy. *Annu Rev Public Health.* 2018;39(1):95–112.
95. Stanford CS class CS231n: Convolutional neural networks for visual recognition. https://cs231n.github.io/convolutional-networks/. Accessed February 2, 2022.
96. Jeewandara T. Wave physics as an analog recurrent neural network. 2020; https://phys.org/news/2020-01-physics-analog-recurrent-neural-network.html. Accessed February 2, 2022.