nature biotechnology

PERSPECTIVE

Deep learning in biomedicine

Michael Wainberg[1,2], Daniele Merico[1], Andrew Delong[1] & Brendan J Frey[1]

Deep learning is beginning to impact biological research and biomedical applications as a result of its ability to integrate vast datasets, learn arbitrarily complex relationships and incorporate existing knowledge. Already, deep learning models can predict, with varying degrees of success, how genetic variation alters cellular processes involved in pathogenesis, which small molecules will modulate the activity of therapeutically relevant proteins, and whether radiographic images are indicative of disease. However, the flexibility of deep learning creates new challenges in guaranteeing the performance of deployed systems and in establishing trust with stakeholders, clinicians and regulators, who require a rationale for decision making. We argue that these challenges will be overcome using the same flexibility that created them; for example, by training deep models so that they can output a rationale for their predictions. Significant research in this direction will be needed to realize the full potential of deep learning in biomedicine.

Driving cars[1], beating humans at their own games[2,3], generating images in the style of other images[4], transcribing speech[5], and translating text[6], deep learning has increasingly captivated the imagination of artificial intelligence (AI) researchers and the general public. In recent years, the approach has also captured the attention of clinicians, for example, aiding physicians in object detection using radiography, computed tomography or magnetic resonance imaging (MRI) data. A common goal of computer modeling in these problem domains is human-level AI: recapitulating complex actions already performed well by humans, but with greater precision.

In contrast to the above applications, a unique aspect of biomedical data is that they are often uninterpretable by the naked eye. For example, upon the completion of the Human Genome Project, geneticist Eric Lander famously quipped: "Genome. Bought the book. Hard to read." Humans are not naturally good at reading the genome, interpreting multidimensional MRI data or predicting target–drug interactions. For biomedical applications (see **Boxes 1–3**), we need AI and computational modeling that can make inferences and deliver insights that humans cannot.

Since the 1960s, computational intelligence and biology have both undergone striking advances (**Fig. 1a**)—sometimes synergistically, such as when the human genome was sequenced. However, the recent acceleration in the production of large-scale biomedical datasets (**Fig. 1b**,**c**) using high-throughput technologies has created an opportunity to re-envision biology and medicine using deep learning. For instance, there are now over 1 million genome datasets, each containing 10 gigabases on average.

In this Perspective, we provide an overview of machine learning and then focus on the subfield of deep learning. We go beyond retrospective reviews of deep learning[7] and its application to biology and medicine[8–15] by describing both technical challenges (for example, how to improve generalization performance in the presence of confounding variables) and implementation challenges (for example, how to gain widespread adoption among physicians, drug developers and regulatory agencies). Finally, we give our outlook for the prospects for deep learning approaches in biology and biomedicine.

Machine learning

Machine learning is a broad class of methods for reasoning and making inferences about data. A popular form of machine learning is supervised learning, which encompasses such methods as linear and logistic regression, random forests, gradient boosting, support vector machines, supervised deep learning and hybrids with other approaches, such as genetic algorithms. The goal of supervised learning is to build a model that can predict a property of an item, called its label, target, response variable or output, using various features that are known about the item, called input features, explanatory variables or input. For example, in computer vision, the input may be an image and the desired output might be a list of detected objects. In proteomics, the input might be an amino acid sequence and the prediction might be a representation of the three-dimensional structure of the resulting protein.

In machine learning, a model can be thought of as a machine with many tunable knobs, which are called parameters or weights. Tuning a knob changes the mathematical function that transforms inputs into predictions. To train a model, we first need a set of training inputs for which the desired predictions, or training labels, are known. Also, we need a way of quantitatively comparing the predictions for those training inputs to the known values. This measure or metric is called a loss function or an error function, and the number it computes is called the error or loss. A model with randomly configured knobs will make many mistakes and have high training error, but a good training algorithm will reconfigure the knobs so that most predictions match the training labels and the training error becomes low (**Fig. 2a**,**b**). Once training is complete, the model can be applied to new input conditions (**Fig. 2c**).

Learning methods are distinguished by the mathematical functions they are capable of learning, and by the assumptions they make about the likely relation between features and labels. For instance, in linear regression, the assumption is that a label can be predicted using a weighted sum of its corresponding features. This is a restrictive function because it presumes the absence of interactions between features.

[1]Deep Genomics Inc., MaRS Discovery District, Toronto, Ontario, Canada. [2]Department of Computer Science, Stanford University, Stanford, California, USA. Correspondence should be addressed to B.J.F. (contact@deepgenomics.com).

Received 12 October 2017; accepted 1 August 2018; published online 6 September 2018; doi:10.1038/nbt.4233

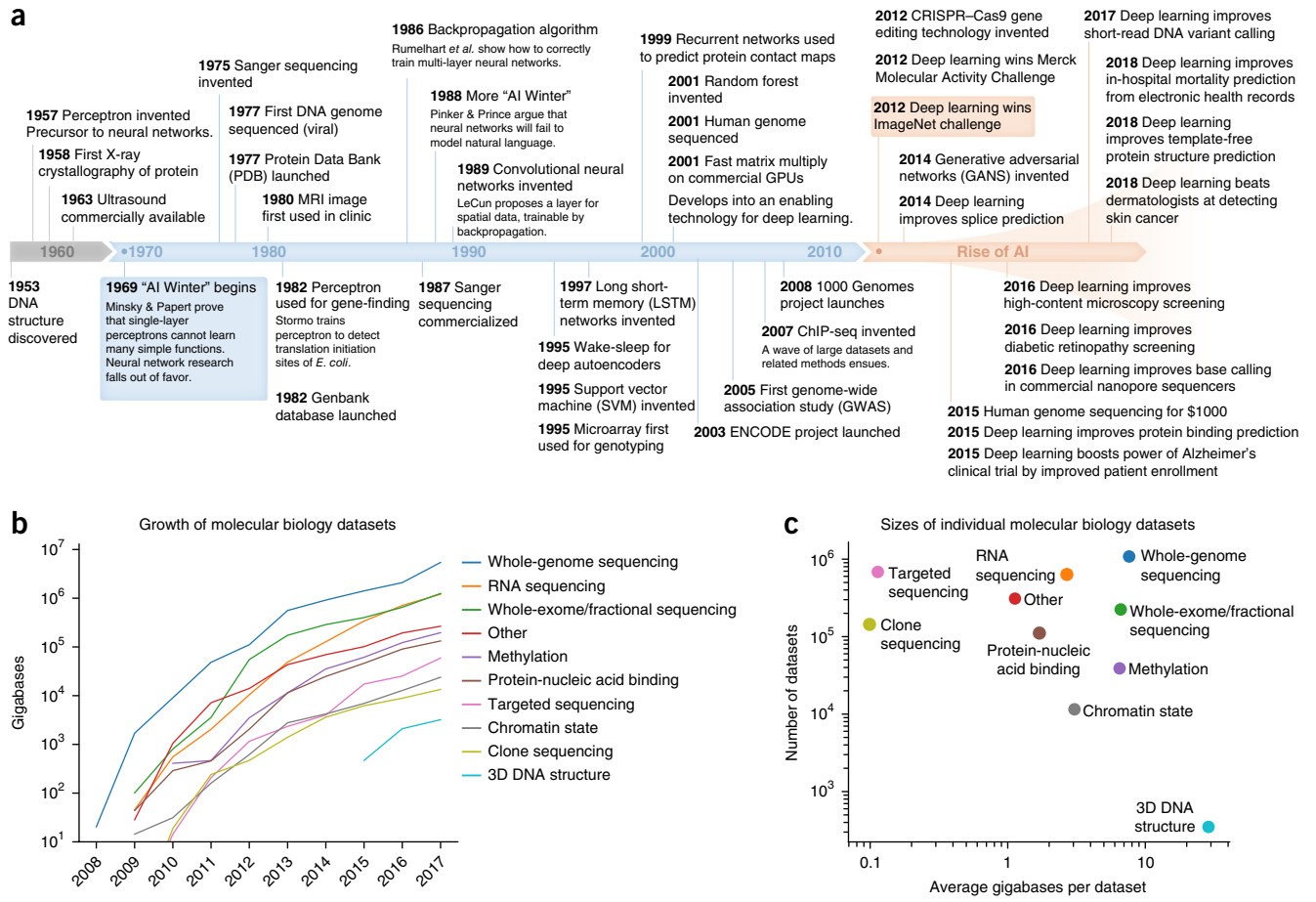

Figure 1 The rise of molecular biology, deep learning and data-driven biomedicine. (**a**) A timeline of milestones in machine learning and biomedicine. (**b**) An example of the explosive recent growth in biomedical data: genomic sequence data (gigabases) over time for various assays[63]. Note the logarithmic scale. (**c**) Number of datasets versus average dataset size for each category. The Sequence Read Archive (SRA) designations for the assays included in each category are as follows: whole-genome sequencing: WGS, WGA, synthetic-long-read; RNA sequencing: RNA-seq, miRNA-seq, EST, ncRNA-seq, FL-cDNA; whole-exome or fractional sequencing: WXS, RAD-seq, WCS; other: OTHER, other, Tn-seq, FINISHING, CTS, VALIDATION; methylation: bisulfite-seq, MeDIP-seq, MBD-seq, MRE-seq; protein–nucleic acid binding: ChIP-seq, RIP-seq, SELEX; targeted sequencing: AMPLICON, targeted-capture; chromatin state: DNase-hypersensitivity, MNase-seq, ATAC-seq, FAIRE-seq; clone sequencing: POOLCLONE, CLONE, CLONEEND; 3D DNA structure: Hi-C, ChIA-PET. Datasets without base count metadata were excluded from gigabase counts and datasets without run metadata were excluded from dataset counts.

For example, it could not accurately model a transcription factor that binds to two distinct patterns in a DNA sequence.

Deep learning

In contrast to linear regression, deep learning is very flexible in how it allows the labels to relate to the input features: labels are functions of intermediate variables (also known as hidden variables, intermediate features, nodes or neurons), which are in turn functions of other intermediate variables, and so on, until some intermediate variables are functions of the input features.

A deep neural network (DNN) can be viewed as a mathematical function built by composing simple transformations called layers, so that the outputs of one layer feed into the inputs of the next. For example, multiple logistic regression is a classical type of layer (**Fig. 3a**, top). Another popular layer is composed of rectified linear units, in which the element-wise sigmoid σ used in logistic regression is replaced with a rectification step that passes the input through but clamps negative values to zero. It is the multiple sequential layers that gives deep learning its name. By contrast, linear and logistic regression are models with only one layer; that is, they are shallow learning models.

The idea of deep learning is that stacks of transformations are extremely powerful and flexible in the kinds of relationships that they can model (**Fig. 3b**) while still being trainable. The most commonly used training method is backpropagation, which iteratively adjusts all weights (the knobs in **Fig. 2a**) so as to minimize the error between predictions and training labels. Backpropagation is named for the backward (output-to-input) flow of computation when determining how much to adjust each weight, which makes efficient reuse of intermediate values that were computed by the forward pass (**Fig. 3c**). The power of deep learning frameworks, such as PyTorch and TensorFlow, is that, given any user-defined model, they automatically derive the correct set of computations needed for backpropagation, no matter how deep or complex the architecture.

Genetic algorithms, sampling methods and other techniques for training DNNs are an important area of current research. A genetic algorithm recently outperformed some implementations of backpropagation when training DNNs to play Atari video games (I. Sutskever, OpenAI, personal communication).

Deep learning in practice

A strength of deep learning is its ability to learn end to end, automatically discovering multiple levels of representation to achieve a prediction task, where the outputs of one level become the input features for the next level. Low-level features (for example, DNA sequence motifs or patterns in a pathology image) and higher-level features (for example, disrupted mRNA splicing or asymmetrical skin lesions), as well as outputs (for example, the detection of cancer), can all be learned jointly from data, reducing or eliminating the need for manual feature engineering done before training. Early stages of DNNs are often similar to classic low-level models (for example, position-weight matrices for DNA sequences and edge detectors for medical images), but are learned jointly with, and in support of, higher-level outcomes (**Fig. 3d**). Deep learning can also easily model complex interactions between features, such as how different transcription factors compete for influence at the same binding site. This makes deep learning a natural choice for modeling hierarchical systems and systems with many interacting components.

Another important strength of deep learning is its ability to use intermediate variables for different but related tasks. For example, a hypothetical intermediate variable that detects the presence of an RNA secondary structure could be used in subsequent layers to detect a protein–RNA interaction, a microRNA target or the formation of a splicing lariat. In theory, sharing intermediate variables across different tasks during learning, in a procedure called multi-task learning[16] (see "Deep learning supports highly flexible architectures"), can elucidate intermediate variables that are more mechanistically relevant and increase the effective amount of data used for training, leading to increased accuracy.

For less experienced users, deep learning is less likely to work out of the box than simpler machine learning methods. In such cases, achieving optimal prediction accuracy may require the tuning of model settings, or hyperparameters. For instance, how large should one make each layer; what is the internal connectivity pattern, or architecture, of the DNN; and how fast should one adjust the parameters or weights during training (see "Deep learning supports highly flexible architectures")? Fortunately, hyperparameters can be selected in a statistically rigorous way via hyperparameter search techniques that use validation or cross-validation examples that are held out from training. Hyperparameter search is particularly necessary to avoid the DNN memorizing the training examples without learning any generalizable patterns, a problem called overfitting, which can occur owing to the large number of parameters.

In general, the amount of data required to accurately train a DNN is larger than for other machine learning models, although this depends strongly on the number of parameters in the model and the system being modeled. One practical way to evaluate whether a problem would benefit from more training data is to fit a curve to the model's validation accuracy after randomly subsampling the training dataset to various sizes. Successful applications of deep learning to biomedicine (see **Boxes 1**–**3**) have used anywhere from thousands to millions of training examples. We recommend comparing the accuracy of deep learning and simpler models, such as Lasso, ElasticNet and gradient boosting, on one's problem of interest to determine whether deep learning is beneficial.

Deep learning is computationally intensive, and specialized computer hardware, such as graphics processing units (GPUs), are frequently employed to train models within a reasonable time. However, with advances in computational power and the availability of software for automated hyperparameter selection[17], deep learning is fast becoming more accessible to nonexpert users. Frameworks such as PyTorch and TensorFlow are transformative to productivity, both for machine learning research and for application-oriented work. Equipped with a commercial GPU, software skills and an understanding of the appropriate layers, a scientist working today can design a model that suits their data and train it in under 100 lines of code. In the near future, the models themselves will be proposed by AI and then systematically evaluated, all with cloud computing. A computational biologist will be able to rapidly and cheaply receive a state-of-the-art model, no matter the nature of their data. Deep learning may well be even easier than training a random forest or a support vector machine is today.

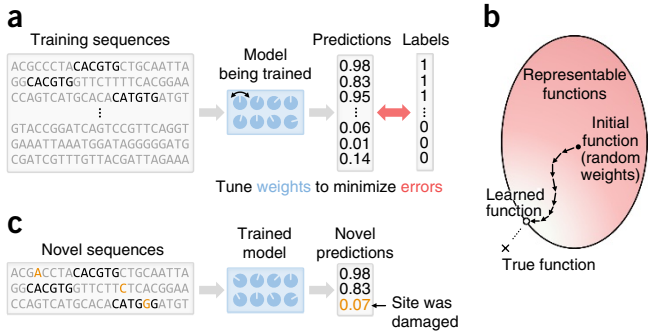

Figure 2 Supervised machine learning. (**a**) The concept of supervised learning, depicted as a pattern-recognition task over DNA sequences. The model has a set of parameters, shown as knobs. For neural networks, parameters are called weights and are typically initialized to small random values. Here, the highlighted patterns indicate binding sites of the Myc transcription factor and the training labels represent experimentally measured probabilities of binding according to, for example, a ChIP-seq assay. (**b**) The weights are then tuned by a learning algorithm to minimize the discrepancy between the model's prediction and the training label, which is quantified and called the training error or loss. (**c**) *In silico* mutagenesis (see **Box 1**) can be used to predict whether a variant would impact Myc binding.

In data-limited situations, deep learning is well suited to leverage large datasets on related problems to improve performance, in an approach called transfer learning[18], and with large enough datasets the performance of deep learning is unparalleled. We expect the relative advantage of deep learning over other supervised machine learning methods to only grow over time, given the ongoing explosion in genomic data generation (**Fig. 1b**,**c**).

Deep learning supports highly flexible architectures

Many advancements in deep learning have come from the introduction of new layer designs. The simplest type of layer, called a fully connected or dense layer, is one in which every input is connected to every output (**Fig. 3a**, top).

However, fully connected layers are suboptimal when patterns have the same meaning regardless of where they appear in the input features. For instance, an object detection model trained on images of pedestrians, but where the pedestrians always appeared in the center of the image, would not be able to recognize pedestrians anywhere else in the image, because every part of the image would have different weights. Alternatively, a fully connected model trained to recognize the motif CACGTG (**Fig. 2a**) would require all instances of the motif to be aligned to the same position, but in general the motif could occur anywhere in the sequence so it would be necessary to scan the sequence to determine its precise location.

A convolutional layer avoids the above problem by tying together the weights during training (**Fig. 3a**, bottom), so that every region of the input ends up with the same weights and can detect the same

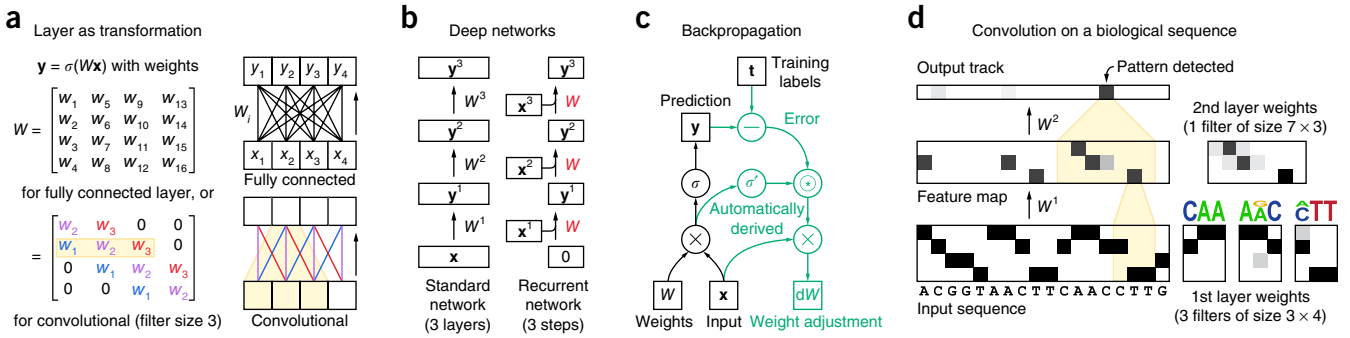

Figure 3 Key concepts in deep learning. (**a**) A standard neural network layer is made of multiple logistic regression models (top). Output vector **y** is computed by applying a nonlinear function (such as a sigmoid) to a linear or affine transformation of **x**; bias parameters are also typical, but omitted for clarity. A convolution layer takes the same form (bottom), but has a sparsity and weight-sharing structure that effectively scans one or more pattern detectors along spatial dimensions of the input. (**b**) In a DNN, complex input–output relationships can be modeled by composing simple layers. The best combinations of layers and weight-sharing schemes will depend on the nature of the data. For example, recurrent networks are useful for time-series data because they learn a single transformation *W* that incorporates new observations with a memory of previous observations when applied iteratively (recurrently) over time. (**c**) The 'compute graph' for the standard fully connected layer from **a** is shown (black), along with the extended compute graph automatically derived for backpropagation (green), which tends to involve element-wise and matrix operations. (**d**) Convolutional networks can learn to recognize patterns in biological sequences such as regulatory elements.

patterns. Convolutional layers that operate on genomic sequences can be thought of as a series of motifs or pattern detectors, each of which is scanned across the sequence in a similar manner to the position-weight matrices used in genomics (**Fig. 3d**). DNNs with multiple stacked convolutional layers, called convolutional neural networks, are the state of the art for image recognition[19].

A recurrent layer is designed to handle sequential inputs where information must be integrated over long distances. Recurrent layers are structurally similar to hidden Markov models, but capable of capturing much more complexity in their states and transitions. Like convolutional layers, recurrent layers scan input sequences element by element, but unlike convolutional layers, recurrent layers also store a memory of earlier parts of the sequence and use this memory, in combination with the current value they are reading, to output a value at each step. This chaining allows a recurrent layer to remember previously observed patterns as it accommodates new inputs. For instance, when given a set of RNA sequences, a recurrent layer may remember having already observed a donor splice site while encountering a candidate acceptor splice site. Bidirectional recurrent layers[20] are an important direction-agnostic extension of recurrent layers and are able to store memories of both previous and subsequent elements of the sequence. A recurrent layer can itself be viewed as a DNN (**Fig. 3b**, right). Networks with recurrent layers are the state of the art for language tasks, such as speech recognition[5] and machine translation[6].

Deep learning also supports a variety of other training configurations, including unsupervised, semi-supervised, multi-modal and multi-task learning.

Unsupervised learning. The goal of unsupervised learning is to identify efficient representations of the dataset without using labels. Then, these representations may be examined by experts (for example, for identifying disease subphenotypes) or used as automatically learned feature descriptors that are fed into supervised machine learning methods (for example, for detecting tumors in medical images). Clustering and principal component analysis are simple forms of unsupervised learning, but more advanced unsupervised deep learning methods are particularly promising in biomedicine. Unsupervised learning can be viewed as finding efficient ways of compressing data, or as finding representations that disentangle the factors that account for variation in the data.

Two highly successful classes of unsupervised DNN are deep generative models and autoencoders. While these two classes are for the most part treated separately, they are closely related. For instance, one of the earliest training methods, the wake-sleep algorithm, jointly trains a deep variational autoencoder and a deep generative model[21].

The goal of a deep generative model is to generate data points that are similar to the ones it was trained on. For instance, deep generative models could be trained on enhancer sequences and then asked to generate candidate enhancers, which could then be validated with a massively parallel reporter assay. Generative adversarial networks[22], a recent innovation in generative modeling, aim to generate examples that are indistinguishable from the real examples the model was trained on. They do this by explicitly asking a second network, called the discriminator, to distinguish the real examples from the generated ones while they progressively refine the generated examples to attempt to fool the discriminator into thinking they are real.

As in principal component analysis, the goal of an autoencoder is to compress each data point into a lower-dimensional representation (called an embedding) while preserving as much information as possible. For instance, an autoencoder could be trained to compress a dataset of 500-bp DNA sequences into a list of 10 real numbers per sequence, which could be used to approximately reconstruct the 500 bp sequence; these embeddings of length 10 could be clustered using any standard clustering technique to discover groups of functionally similar sequences. A clustering algorithm could be applied directly to the 500-bp sequences, but it might not perform as well because the sequences are large and may be redundant or contain irrelevant portions. Autoencoders have also been used in genomics to compress the rich, high-dimensional information contained in gene expression datasets[23], and in drug discovery to obtain compact representations of drug-like molecules[24]. Autoencoders and deep generative models are not mutually exclusive: some types of autoencoders, such as the powerful variational autoencoder, are formulated as generative models.

Semi-supervised learning. Semi-supervised learning is a fusion of supervised and unsupervised learning that can leverage datasets in which some data points have labels and others do not. This is a

Box 1 Applications to genetic data

Determining the relation between genotype and phenotype is the most central problem in genetics. The association paradigm adopted by the genome-wide association study (GWAS) relies on collecting enough individuals to statistically establish association between variation at a genetic locus and phenotype. Identification of specific genetic variants can pinpoint relevant biological pathways and suggest targets for drug discovery. Aggregate results can be further used to model disease risk imparted by germline and common genetic variation, using a linear additive model called the polygenic risk score. The association paradigm is best suited for genetically complex traits or disorders, like height or schizophrenia, for which common genetic variation explains a large fraction of phenotype variance. Datasets comprising tens of thousands of individuals often are required to discover statistically significant associations[36].

Beyond genotype–phenotype association is the study of the functional effects of variants: that is, how genetic variation produces phenotypes by mechanistically altering molecular, cellular or physiological processes. This is particularly crucial for variants of large effect, typically implicated in Mendelian disorders and cancer. For instance, for a gene whose complete loss of function has been implicated in a Mendelian disease, every novel variant causing complete loss of function can be inferred to contribute to the disease[37]. An important advantage of this approach is the elucidation of drug targets, which require a mechanistic understanding of intervention at the molecular level. In contrast, mechanistic characterization is more challenging for variants discovered purely using GWAS, which typically have small effect sizes, are noncausal because of linkage disequilibrium with nearby variants, and are not easily amenable to gene mapping because they usually impact distal regulatory elements[36,38]. In addition, the number of genes that can contribute to a given phenotype via small perturbations may be much larger than critical genes clustered in core pathways[39]. These challenges impact the capability of GWAS to illuminate pathways contributing to disease etiology and suggest therapeutic targets.

The most disruptive variant effects can be determined using genome annotations and relative simple rules (for example, frameshift and stopgain effects). Deep learning is well suited to modeling the molecular phenotypes of genetic variants impacting transcription, splicing, transcript stability and translation regulation: the hierarchical complexity of these processes makes simpler models like position-weight matrices suboptimal[9,40].

Prominent applications of deep learning to molecular phenotypes include SPIDEX[31], which predicts the percent-spliced-in of cassette exons across tissues from DNA sequence and other features; DeepBind[41], which predicts transcription factor and RNA-binding protein binding from DNA and RNA sequences; Basset[42], which predicts DNase hypersensitivity from DNA sequence; DeepSEA[43] and DanQ[44], which simultaneously predict transcription factor binding, histone modifications and DNase hypersensitivity from DNA sequence; DeepCpG[45], which predicts whether a CpG is methylated using single-cell methylation; and TITER[46], which predicts translation initiation sites from DNA sequence, among many others. In general, deep learning is critical in constructing quantitative models of molecular phenotypes, interfacing DNA sequence to the growing body of molecular activity data produced using next generation sequencing and other technologies.

DNNs can be trained using the reference genome sequence and annotations (for example, known splice sites) or molecular profiles (for example, exon inclusion in cell types of interest). Using a technique known as *in silico* mutagenesis, these reference data models can then be used to predict the effect of genetic variants: the model is run for the reference and the mutated sequence, and the difference between the two predictions corresponds to the variant effect on the molecular phenotype[31]. A complementary approach is to train DNNs using mutagenesis data. However, reference data are currently easier and cheaper to generate experimentally. Hundreds of thousands to millions of data points can be produced for reference data, as opposed to hundreds or thousands for most mutagenesis experiments. Larger-scale massively parallel reporter assays or high-throughput CRISPR–Cas9 screens may change the status quo[47].

DNNs of molecular phenotype can be used to mechanistically characterize variants discovered by GWAS, raising the as-yet-unanswered question of whether they can be used to increase statistical power—for instance, by up-weighting variants producing larger molecular phenotypes. Additionally, polygenic risk methods currently rely on linear additive models, where weights are derived directly from non-mechanistic genotype–phenotype associations and where complex interactions between variants are ignored. In principle, DNNs can produce significant improvements by modeling more complex relationships. However, a challenge is how to apply deep learning when there are more features (variants) than training examples (individuals), which is usually the case. Flexible architectures involving convolutional layers, unsupervised learning and semi-supervised learning could be helpful here.

In contrast to image recognition, where convolutional neural networks simplify the prediction problem by encoding strong assumptions about the general structure of images, it is not clear what prior assumptions can be used to derive a highly effective DNN architecture for modeling GWAS data. What may be the future path toward solving this problem? We argue that deep models of molecular phenotypes may provide an initial analysis layer, determining what the variant effects are on gene expression levels. Additional layers would have to handle network-level gene interactions, comprising cellular and physiological processes. This presents a challenge to the community. Recent research has taken advantage of the large number of data points available for yeast double mutants and their effect on cellular growth, proposing a DNN architecture informed by Gene Ontology functional gene annotations[48]. Although this approach constitutes an interesting first proof of concept, we argue that we ultimately need deep learning systems able to learn regulatory networks jointly from genotype-phenotype data and other experimental data while incorporating biological knowledge.

particularly common scenario in genomics, where only a small fraction of the genome may have high-quality labels for a specific problem; similar scenarios exist in other areas of biomedicine. For instance, to predict the effects of genetic variants, semi-supervised learning could be used to extrapolate the results of medium-throughput saturation mutagenesis experiments to the rest of the genome. Semi-supervised deep learning may be performed with deep generative models[25].

Multi-modal and multi-task learning. A striking advantage of deep learning over other machine learning methods is its ability to

Box 2 Applications to QSAR modeling in drug discovery

Drug discovery involves both target identification and the determination of compounds with good on-target effect and minimal off-target effects. The first problem, target identification, can benefit from DNNs of molecular effect (see **Box 1**), but here we focus on the second problem, predicting the biological activities of drug candidates toward targets of interest. For drugs, accurate quantitative structure–activity relationship (QSAR) models would facilitate discovery of small molecules having high activity toward a therapeutic target and—to minimize the likelihood of side effects—low activity toward off-target proteins.

A central complicating factor in QSAR modeling is the structural complexity of proteins, which is typically orders of magnitude greater than that of small molecules. Indeed, even predicting a protein's tertiary structure from polypeptide sequence is an unsolved problem; for instance, deep learning efforts in protein structure prediction from sequence have largely focused on secondary structure using recurrent neural networks[49,50] because of the intractability of predicting tertiary structure from sequence.

One way of dealing with this structural complexity is to ignore it, by modeling only the small molecule and not the protein, using ligand-based models. In its simplest form, this way of solving the problem involves experimentally measuring the activities of many small molecules against a single protein, then training a model to predict these activities from features of the small molecules (often called molecular descriptors or fingerprints), such as the counts and arrangements of atoms and functional groups within the molecule. Alternatively, molecular descriptors may be automatically derived from chemical structures using a model such as an autoencoder[23] (see "Deep learning supports highly flexible architectures"), or even learned on the fly using a specialized neural network architecture[51,52]. Once trained, the model may then be used to predict the activities of other, unmeasured small molecules against the same protein.

An improved version of this approach uses multi-task deep learning (see "Deep learning supports highly flexible architectures") to train a joint model across several proteins, which conceptually allows the model to learn from fewer data by exploiting the fact that similar small molecules tend to have correlated activities across multiple proteins. In 2012, the Merck Molecular Activity challenge, a competition to perform QSAR modeling across 15 protein targets, was won by a team using an ensemble of multiple models including both single-task and multi-task DNNs[53,54]. This multi-task approach was later scaled up to >200 targets[55], with the authors showing that performance did not saturate but instead continued to improve as more targets and tasks were added.

A key limitation of both single- and multi-task ligand-based models is that, because proteins are not modeled explicitly, the model is limited in its ability to generalize to proteins not in the training set[55]. This leads to a perverse situation in which the proteins most in need of predictions—because they have the least available data—are also the hardest to predict accurately[56]. A recent, promising approach[56] uses three-dimensional convolutional layers to directly model the structure of the protein, in addition to the structure of the small molecule, enabling the DNN to generalize to completely novel proteins with no experimental biological activity data. Unlike traditional QSAR models, the output of this approach is whether the small molecule interacts with the protein, rather than its biological activity toward the protein. This kind of deep learning approach is complementary to established methods, such as molecular dynamics simulations: whereas molecular dynamics is advantageous in data-limited scenarios, deep learning is data-driven and can avoid any incorrect assumptions made by molecular dynamics.

naturally integrate input data from multiple modalities and targets from multiple tasks. Multi-modal learning, in which input data from different modalities is used for training, can be accomplished by building a separate submodule for each data type and then feeding the outputs of all the submodules into a subsequent layer in the network. These submodules can perform standard DNN operations (for example, convolutional and recurrent layers) directly on the raw data, as an alternative to assay-specific feature engineering. For instance, chromatin immunoprecipitation sequencing (ChIP-seq) data for multiple histone marks can be combined with chromatin accessibility data (for example, from DNase-seq) to predict variant function.

Multi-task learning[19], in which output targets from different tasks are used for training, is enabled by feeding the outputs of earlier layers into subnetworks that each output a label for a corresponding task. If different tasks benefit from the detection of similar patterns, this effectively provides substantially more data for training earlier layers, since each additional output label acts like an additional training case (see **Box 2**).

DNNs are modular: the above elements and many more, such as deep reinforcement learning, can be combined, automatically or with human guidance, in diverse ways to identify creative and highly effective solutions[26].

Generalization, reliability and performance of deployed models

When machine learning models are deployed in real-world applications, it is important that performance guarantees be provided. Compared with shallow models, a deep model can go wrong in many more ways, making performance considerations both more important and more challenging.

Standard statistical and machine learning methods should be applied, such as: selecting hyperparameters of the model using cross-validation; testing the model using held-out data to evaluate performance before deployment; assessing prediction confidence intervals using the bootstrap or a Bayesian method; and analyzing the sensitivity of the model's output to certain parameters, input features and training cases. The fact that DNNs may have hundreds to millions of times more parameters than shallow models presents a challenge to the research community going forward. For example, it is often desirable to quantify the uncertainty in the output of a DNN that is due to limited training data, biased training data, insufficient information at the input, or inherent biological noise. In principle, Bayesian deep learning[27,28] can be used, but other methods, such as test-time dropout[29] and the bootstrap[30], may work better in practice. For example, we previously used Bayesian deep learning to classify disease variants[31].

In biology and medicine, the training conditions are likely to be quite different from the application conditions. This training–application gap is characterized by the following challenges, which need to be addressed by careful consideration and new approaches:

- **Target mismatch.** The target that is most important to users may not match the target used for training. For example, the

Box 3 Applications to medical imaging

Today, deep learning applies to imaging more directly than any other area of biology and medicine. The longstanding focus on image understanding in AI has led to breakthroughs in classification, segmentation, registration, object detection and tracking. All of these tasks are encountered in biomedical imaging, across diagnostics, pathology, high-content screening, molecular imaging and more. Even though biomedical imaging is distinct in many ways, with multi-modal and three-dimensional data being prominent, it is the most immediate beneficiary of deep learning. In fact, even off-the-shelf models applied to biomedical images have met with surprising success, despite most parameters having been trained to classify natural scenes[57–59].

In drug discovery, high-content cell imaging is a powerful platform for phenotypic screening. Image analysis may require, for example, detecting changes in subcellular localization[60] or discerning the difference between healthy, diseased, stressed or even novel cell phenotypes. Hand-engineered tools like CellProfiler[61] have long been used to quantify basic cytometrics like cell count, size, shape or fluorescence, but higher-level interpretations are harder to automate. This is because intraclass variation can appear large relative to the distinguishing features of interclass variation. Deep learning is opening a path for image cytometry to meet or even exceed human-level interpretation while keeping pace with the scale of image acquisition enabled by robotics.

In image diagnostics, stakes can be high for patients and for insurance providers. A common concern for DNNs is that they are black boxes, with no way to interpret how the algorithm arrived at a diagnosis. A strategy to address this is to highlight image regions that were consequential to the prediction, allowing diagnosticians to check the supporting evidence at a glance. Such regions can be computed by sensitivity analysis, asking "what perturbation to this image would change its diagnosis?" Deep learning provides the computational machinery to answer this question, similarly to how *in silico* mutagenesis can be used to identify important sequence features (see **Box 1**).

In our view, the trend of treating human predictions as ground truth misses a great opportunity to exceed human performance. For example, radiologists interpret a billion images worldwide each year, but autopsies have identified major errors in up to 20% of cases[62]. Autonomous systems should be trained against the most conclusive diagnostic available, such as multi-expert consensus or biopsy, even if those outcomes are only known downstream of the imaging diagnostic scenario or require tracking electronic health records.

The US Food and Drug Administration has already approved deep learning systems for heart segmentation, tumor tracking and retinopathy detection. These early systems are refinements on existing clinical decision support tools that practitioners use every day. As the outputs of these systems become more trusted in the field, new systems will emerge to take on responsibility for higher-level diagnostic decisions.

model was trained using tumor size as the target, whereas in the application the most relevant target is survival time.

- **Loss function mismatch.** The loss function used for training may not match the loss function that is important to users. For example, the training loss function is squared error in predicting tumor size, but the physician only cares about whether the tumor exceeds a certain size.
- **Data mismatch and selection bias.** The collection of training data may have been done in a way that does not match the application conditions and introduces bias. For example, training data were collected at a specific hospital, introducing a bias as to which types of patients were seen.
- **Nonstationary environments.** If the environment changes over time, the conditions at application will have drifted compared with those at training. For example, for a model that takes as input sequencing read counts, the quality of the reads may improve over time.
- **Reactive and adversarial environments.** Application of the model may alter the environment in a way that was not accounted for during training. For example, if a patient's treatment is altered using the model, a second application of the model to that patient may no longer be valid. In some cases, the environment actively changes to undermine the value of the model. For example, HIV evolves to escape predicted vaccines.
- **Confounding variables and causality.** Learned relationships between two variables may in fact be due to a third, unobserved variable, and this correlation may be mistaken for causation. For instance, a genetic variant may be strongly associated with a disease indication, but in fact this association is due to a different variant that causes the disease and that co-occurs with the associated variant because of linkage disequilibrium. Identifying causal relationships can help to bridge the training–application gap because these relationships do not change when training and application conditions change.

Establishing performance guarantees and stakeholder trust

Deep learning has ushered in an era of medicine wherein we can imagine human experts relying on AI and machine learning. Suppose we have built a model that can accurately diagnose a patient's disease-causing mutation and generate a tailor-made therapy that is safe and effective. A major subsequent challenge is establishing the trust of stakeholders in the deep learning approach before deployment and use. Stakeholders include patients, friends and family, physicians, ethics review boards, professional societies, diagnostic laboratories, biopharmaceutical companies, technology providers, insurance providers and regulators. Although previous research on such topics as model interpretability[32] and causality[33] provides helpful background, in this section we outline a different, stakeholder-centric view of the challenges ahead.

A stakeholder will either place their trust in the hands of another agent, such as an expert, an institution or a regulatory agency, or will need to be directly convinced that the model is trustworthy. Stakeholders refer to one another when establishing trust, but they are convinced in different ways and they assess benefits, costs and risks of decisions differently. For example, a patient may seek to survive longer, whereas a regulator may be looking for 50% or more of patients to survive longer.

Performance. When assessing benefits, costs and risks, stakeholders are looking for performance guarantees in the form of metrics, such as the fraction of patients that benefit from a drug. Surrogate metrics are used for training, such as mean squared error in predicting a drug response biomarker. For a specific metric, the highest level of machine learning and statistical expertise is required, both for training the model and for assessing how it will perform when deployed.

Careful attention must be paid to a range of issues, which include data preprocessing, optimization, model selection, overfitting, outliers, context dependence, missing information, confounding variables, and environments that are nonstationary, reactive or adversarial.

It is important to establish the metrics that stakeholders will use. In a genetic variant–calling application, what sensitivity and specificity is acceptable for regulatory approval? In a molecular diagnostics application, what rates of false positives and false negatives are acceptable to the professional association that oversees diagnostics? In a drug development application, what is the tradeoff between accuracy in predicting the effect of a drug on its target versus its toxicity-inducing effects? How does the answer change when the stakeholder is a biopharmaceutical company or a regulatory agency? This raises the issue that different stakeholders will use different metrics, and these metrics may not be known ahead of time when the model is built. Consequently, the performance guarantees should be robust to the metric used, which can partly be addressed by training and testing using different metrics, possibly using a multi-task framework. At a minimum, the deployed model must be consistent with facts that are known to be true, regardless of the metric used, and stakeholders may reasonably demand that the models be interrogated to provide evidence.

Rationale. Stakeholders seek to develop their own rationale for how the model will behave, so that they can gain confidence in the model using common sense (the 'smell test'), intuition, thought experiments, and discussion with other stakeholders.

Good rationales almost always rely on causal explanations that the stakeholder can be convinced to be true, so information must be provided about causal relationships. For example, models that reflect causal relationships can be used to develop therapeutic interventions (see **Box 3**). Previously, we developed a DNN that takes DNA sequence as input and predicts exon splicing, and we applied it to the spinal muscular atrophy gene *SMN2* to identify potential therapies that rescue aberrant splicing[31]. A model that distinguishes exons may take as input the frequencies of protein-binding sequence motifs and codon enrichment. Although changes in protein-binding motifs are likely to alter exon splicing, changes in codon enrichment are not likely to do so because the spliceosome does not mechanistically read codons. So, even though both are correlated with splicing changes, a therapy that targets protein-binding motifs is more likely to be effective.

Assumptions should not be made about what constitutes a good rationale or causal explanation. Instead, it is important to study stakeholders and engage them in advance, by listening, teaching and learning, to establish expectations. Reading their literature and attending their conferences will assist in understanding the variables they will use to construct their own rationales and causal explanations. The model should be built so that information pertaining to those variables can be made available to each stakeholder, along with user documentation, literature and expert advice, so that they can infer a rationale.

It should be kept in mind that a good rationale is one that holds up to adversarial challenges, in which the assumptions, intermediate conclusions and causal variables are poked and prodded, often from different perspectives, especially ones that were not incorporated into the training procedure. Stakeholders often gain confidence through interaction, by asking unexpected questions to see if the model's decision can be rationalized from different perspectives. For this purpose, it may be necessary to support interactive testing with stakeholders, including the ability to produce hypotheses and to test them experimentally.

Transparency. This refers to how easily a stakeholder can examine the model and understand, or explain, how the model operates when combining the inputs to produce the output, regardless of how accurate the model is. For example, in a linear model, a positive parameter indicates that increasing the input will lead to an increase in the output. Transparency can enable experts to determine whether a model conforms to existing scientific knowledge or whether some aspects are suspicious, and decide on follow-up experiments or additional data acquisition to validate the model and explore potential confounding factors.

Whether a model is transparent should not be confused either with whether it accurately represents the phenomenon being modeled or with whether the operational explanation is useful for a specific task. In the above example, even though the parameter is positive, increasing the input in the real world may cause the output to decrease because in reality the output nonlinearly depends on the inputs or the input co-varies with another input that has an opposite and stronger effect. Attending the emergency room may be positively correlated with mortality, but that does not mean the emergency room should be always avoided.

Transparency provides one way to build a rationale, but transparency is neither necessary nor sufficient for producing a good rationale. For example, if a model is very inaccurate, transparency will be of little value in producing a rationale, and if a model is trained to output a good rationale, transparency may not be necessary.

In the context of establishing trust, a commonly held belief is that DNNs are disadvantaged because they are nontransparent 'black boxes'[32,34]. However, this does not justify using an oversimplified, albeit more transparent, model such as linear regression. Whether a model is linear, shallow or deep says nothing about the value of the model for the purposes of stakeholder needs. The fact is that complex and hierarchical biological phenomena, such as transcriptional regulation, often can be better modeled using deep learning. If a DNN provides a more accurate model, then the crucial question is how to make the operation of the model transparent.

Improving the transparency of DNNs is an active area of research. Interestingly, in a DNN, the effect of an input feature on the output, conditional on the values of other input features, can be determined by adjusting the input feature, recomputing the output, and examining the change using an approach such as *in silico* mutagenesis. An attractive aspect of this approach is that it reveals the effect of changing the input feature in the context of the other input features.

Model interpretability. Another concept that is widely discussed is model interpretability. Here, it is expected that the stakeholder will interpret information derived by dissecting the model or its output. However, while interpretability seems to be a desirable and possibly useful property for a model to have, the vagueness of the definition has led to a disconnect between its implemented forms and the needs of stakeholders[35]. Theoretical work on inferring causality[33] is relevant to the above topics, but a major limiting factor is that existing techniques break down when there are hidden variables, which biology is fraught with. In contrast to interpretability and inferring causality, goals pertaining to performance, rationale and transparency can be more clearly defined and more successfully implemented.

Future perspective

Deep learning will radically transform human wellness and healthcare. But how will it be integrated seamlessly into a health management system of the future? Imagine the following scenario:

On her way to work, a woman is notified by her cell phone that she should stop by the local drug store and have a blood and urine test. This notification is produced by an AI system that has access to her health care records, medical images, genome data, periodically updated blood transcriptomes and metabolomes, and historical data profiling her heart rate, blood pressure, muscle strength and other psychomotor indicators. The recommendation is based on an analysis of similar observational and control-normalized data from hundreds of millions of people, including her relatives, as well as millions of cell biology datasets consisting of quadrillions of training cases.

The blood test detects an alteration in her transcriptome and the urine test detects a corresponding alteration in her metabolome suggesting the onset of a neuromuscular degenerative disorder. She is not surprised, because her data had previously indicated that this event was likely to happen sometime in the next year. In fact, her mother had the option of having the associated pathogenic variant edited out of her DNA in utero, *but an AI system assessed that the probability of undesired side effects was sufficiently high that her mother chose not to.*

After receiving this news, she is offered a genetic medicine precisely engineered to be optimal according to an analysis of her data, including her genome and her transcriptome.

Medicines designed with the assistance of AI-driven systems have been shown to be safer than an evening stroll downtown and to achieve a high level of efficacy in 99 out of 100 applications. At this point, AI systems have been shown to be more accurate than animal studies, including those in nonhuman primates, for predicting the safety of compounds in humans. Consequently, the systems themselves have received approval from regulatory agencies.

She selects the medicine and it arrives at the office of a nearby therapeutic counselor the next day. She meets with the counselor to discuss aspects of the treatment regime. Over the next year, as she administers the medicine, her devices continue to record information, including relevant psychomotor indicators, such as arm strength when she performs physical activity and speed of her pace and gait when she walks. She stops by the drug store once every 2 weeks to have her metabolome monitored.

Within 1 year, all evidence shows that the neuromuscular degeneration has been halted. Further, her data have been automatically incorporated into the AI systems to provide better medical treatment for other people in the future.

Although the above scenario seems far-fetched from our present viewpoint, it is one interpretation of how medical practice may undergo a major disruption in the coming years. What we do know is that medicine is already being transformed by the exponential growth in genetic, molecular, biometric and chemical data being collected via a burgeoning array of mobile sensors and medical devices. It is also clear that biology and medicine are too complex for any one individual, or indeed any group of individuals, to accurately understand or to act upon without the support of intelligent computer systems. Our view is that deep learning is the most promising technology for intelligently incorporating huge amounts of data and modeling complex systems. It follows that deep learning will play a key role in the future of biomedicine.

ACKNOWLEDGMENTS

Our perspectives were influenced by conversations with many people, including members of Deep Genomics, B. Andrews, Y. Bengio, B. Blencowe, C. Boone, D. Botstein, C. Francis, A. Heifets, G. Hinton, T. Hughes, P. Hutt, R. Klausner, E. Lander, Y. LeCun, A. Levin, Q. Morris, B. Neale, S. Scherer and J.C. Venter.

COMPETING INTERESTS

All authors are, or recently were, employees of Deep Genomics, an AI therapeutics company, which is using deep learning to identify the genetic determinants of disease and to develop therapies.

1. Waldrop, M.M. Autonomous vehicles: no drivers required. *Nature* **518**, 20–23 (2015).
2. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
3. Silver, D. *et al.* Mastering the game of Go without human knowledge. *Nature* **550**, 354–359 (2017).
4. Gatys, L.A., Ecker, A.S. & Bethge, M. Image style transfer using convolutional neural networks. in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* https://doi.org/10.1109/CVPR.2016.265 (2016).
5. Graves, A., Mohamed, A.-R. & Hinton, G. Speech recognition with deep recurrent neural networks. in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* https://doi.org/10.1109/icassp.2013.6638947 (2013).
6. Sutskever, I., Vinyals, O. & Le, Q.V. Sequence to sequence learning with neural networks. in. *Neural Information Processing Systems* **2014**, 3104–3112 (2014).
7. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
8. Angermueller, C., Pärnamaa, T., Parts, L. & Stegle, O. Deep learning for computational biology. *Mol. Syst. Biol.* **12**, 878 (2016).
9. Leung, M.K.K., Andrew, D., Babak, A. & Frey, B.J. Machine learning in genomic medicine: a review of computational problems and data sets. *Proc. IEEE* **104**, 176–197 (2016).
10. Mamoshina, P., Vieira, A., Putin, E. & Zhavoronkov, A. Applications of deep learning in biomedicine. *Mol. Pharm.* **13**, 1445–1454 (2016).
11. Min, S., Lee, B. & Yoon, S. Deep learning in bioinformatics. *Brief. Bioinform.* **18**, 851–869 (2017).
12. Gawehn, E., Hiss, J.A. & Schneider, G. Deep learning in drug discovery. *Mol. Inform.* **35**, 3–14 (2016).
13. Jurtz, V.I. *et al.* An introduction to deep learning on biological sequence data: examples and solutions. *Bioinformatics* **33**, 3685–3690 (2017).
14. Ching, T. *et al.* Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* **15**, 20170387 (2018).
15. Baldi, P. Deep learning in biomedical data science. *Annu. Rev. Biomed. Data Sci.* **1**, 181–205 (2018).
16. Ruder, S. An overview of multi-task learning in deep neural networks. Preprint at https://arxiv.org/abs/1706.05098 (2017).
17. Liu, H., Simonyan, K., Vinyals, O., Fernando, C. & Kavukcuoglu, K. Hierarchical representations for efficient architecture search. Preprint at https://arxiv.org/abs/1711.00436 (2017).
18. Weiss, K., Khoshgoftaar, T.M. & Wang, D. A survey of transfer learning. *J. Big Data* **3**, 9 (2016).
19. Krizhevsky, A., Sutskever, I. & Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2017).
20. Schuster, M. & Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **45**, 2673–2681 (1997).
21. Hinton, G.E., Dayan, P., Frey, B.J. & Neal, R.M. The wake-sleep algorithm for unsupervised neural networks. *Science* **268**, 7761831 (1995).
22. Goodfellow, I.J. *et al.* Generative adversarial networks. Preprint at https://arxiv.org/abs/1406.2661 (2014).
23. Tan, J., Ung, M., Cheng, C. & Greene, C.S. Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders. *Pac. Symp. Biocomput.* **2015**, 132–143 (2015).
24. Gómez-Bombarelli, R. *et al.* Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **4**, 268–276 (2018).
25. Kingma, D.P., Rezende, D.J., Mohamed, S. & Welling, M. Semi-supervised learning with deep generative models. Preprint at https://arxiv.org/abs/1406.5298 (2014).
26. Pham, H., Guan, M.Y., Zoph, B., Le, Q.V. & Dean, J. Efficient neural architecture search via parameter sharing. Preprint at https://arxiv.org/abs/1802.03268 (2018).
27. MacKay, D.J.C. A practical Bayesian framework for backpropagation networks. *Neural Comput.* **4**, 448–472 (1992).
28. Neal, R.M. *Bayesian Learning for Neural Networks* (Springer, Berlin and Heidelberg, Germany, 1996).
29. Gal, Y. & Ghahramani, Z. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. Preprint at https://arxiv.org/abs/1506.02142 (2015).
30. Efron, B. Bootstrap methods: another look at the jackknife. *Ann. Stat.* **7**, 1–26 (1979).
31. Xiong, H.Y. *et al.* RNA splicing. The human splicing code reveals new insights into the genetic determinants of disease. *Science* **347**, 1254806 (2015).
32. Lipton, Z.C. The mythos of model interpretability. Preprint at https://arxiv.org/abs/1606.03490 (2016).
33. Pearl, J. Causal inference in statistics: an overview. *Stat. Surv.* **3**, 96–146 (2009).

34. Shrikumar, A., Greenside, P., Shcherbina, A. & Kundaje, A. Not just a black box: learning important features through propagating activation differences. Preprint at https://arxiv.org/abs/1605.01713 (2016).
35. Hoskins, R.A. *et al.* Reports from CAGI: the critical assessment of genome interpretation. *Hum. Mutat.* **38**, 1039–1041 (2017).
36. Visscher, P.M. *et al.* 10 years of GWAS discovery: biology, function, and translation. *Am. J. Hum. Genet.* **101**, 5–22 (2017).
37. Richards, S. *et al.* Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. *Genet. Med.* **17**, 405–424 (2015).
38. Timpson, N.J., Greenwood, C.M.T., Soranzo, N., Lawson, D.J. & Richards, J.B. Genetic architecture: the shape of the genetic contribution to human traits and disease. *Nat. Rev. Genet.* **19**, 110–124 (2018).
39. Boyle, E.A., Li, Y.I. & Pritchard, J.K. An expanded view of complex traits: from polygenic to omnigenic. *Cell* **169**, 1177–1186 (2017).
40. Wagih, O., Merico, D., Delong, A. & Frey, B.J. Allele-specific transcription factor binding as a benchmark for assessing variant impact predictors. Preprint at *bioRxiv* https://doi.org/10.1101/253427 (2018).
41. Alipanahi, B., Delong, A., Weirauch, M.T. & Frey, B.J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* **33**, 831–838 (2015).
42. Kelley, D.R., Snoek, J. & Rinn, J.L. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.* **26**, 990–999 (2016).
43. Zhou, J. & Troyanskaya, O.G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods* **12**, 931–934 (2015).
44. Quang, D. & Xie, X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.* **44**, e107 (2016).
45. Angermueller, C., Lee, H.J., Reik, W. & Stegle, O. DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome Biol.* **18**, 67 (2017).
46. Zhang, S., Hu, H., Jiang, T., Zhang, L. & Zeng, J. TITER: predicting translation initiation sites by deep learning. *Bioinformatics* **33**, i234–i242 (2017).
47. Shendure, J. & Fields, S. Massively parallel genetics. *Genetics* **203**, 617–619 (2016).
48. Ma, J. *et al.* Using deep learning to model the hierarchical structure and function of a cell. *Nat. Methods* **15**, 290–298 (2018).
49. Baldi, P., Brunak, S., Frasconi, P., Soda, G. & Pollastri, G. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics* **15**, 937–946 (1999).
50. Pollastri, G., Przybylski, D., Rost, B. & Baldi, P. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins* **47**, 228–235 (2002).
51. Duvenaud, D. *et al.* Convolutional networks on graphs for learning molecular fingerprints. Preprint at https://arxiv.org/abs/1509.09292 (2015).
52. Kearnes, S., McCloskey, K., Berndl, M., Pande, V. & Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J. Comput. Aided Mol. Des.* **30**, 595–608 (2016).
53. Dahl, G.E., Jaitly, N. & Salakhutdinov, R. Multi-task neural networks for QSAR predictions. Preprint at https://arxiv.org/abs/1406.1231 (2014).
54. Ma, J., Sheridan, R.P., Liaw, A., Dahl, G.E. & Svetnik, V. Deep neural nets as a method for quantitative structure-activity relationships. *J. Chem. Inf. Model.* **55**, 263–274 (2015).
55. Ramsundar, B. *et al.* Massively multitask networks for drug discovery. Preprint at https://arxiv.org/abs/1502.02072 (2015).
56. Wallach, I., Dzamba, M. & Heifets, A. AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. Preprint at https://arxiv.org/abs/1510.02855 (2015).
57. Liu, Y. *et al.* Detecting cancer metastases on gigapixel pathology images. Preprint at https://arxiv.org/abs/1703.02442 (2017).
58. Wang, D., Khosla, A., Gargeya, R., Irshad, H. & Beck, A.H. Deep learning for identifying metastatic breast cancer. Preprint at https://arxiv.org/abs/1606.05718 (2016).
59. Lakhani, P. & Sundaram, B. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology* **284**, 574–582 (2017).
60. Kraus, O.Z. *et al.* Automated analysis of high-content microscopy data with deep learning. *Mol. Syst. Biol.* **13**, 924 (2017).
61. Carpenter, A.E. *et al.* CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.* **7**, R100 (2006).
62. Bruno, M.A., Walker, E.A. & Abujudeh, H.H. Understanding and confronting our mistakes: the epidemiology of error in radiology and strategies for error reduction. *Radiographics* **35**, 1668–1676 (2015).
63. Leinonen, R., Sugawara, H. & Shumway, M. The Sequence Read Archive. *Nucleic Acids Res.* **39**, D19–D21 (2011).