ANNUAL REVIEWS

Annual Review of Biomedical Data Science

Deep Learning in Biomedical Data Science

Pierre Baldi

Department of Computer Science, Institute for Genomics and Bioinformatics, and Center for Machine Learning and Intelligent Systems, University of California, Irvine, California 92697, USA; email: pfbaldi@uci.edu

Keywords

neural networks, machine learning, omic data, biomedical imaging, electronic medical records

Abstract

Since the 1980s, deep learning and biomedical data have been coevolving and feeding each other. The breadth, complexity, and rapidly expanding size of biomedical data have stimulated the development of novel deep learning methods, and application of these methods to biomedical data have led to scientific discoveries and practical solutions. This overview provides technical and historical pointers to the field, and surveys current applications of deep learning to biomedical data organized around five subareas, roughly of increasing spatial scale: chemoinformatics, proteomics, genomics and transcriptomics, biomedical imaging, and health care. The black box problem of deep learning methods is also briefly discussed.

1. INTRODUCTION

In essence, deep learning is a rebranding of neural networks as they were developed and used in the 1980s, direct descendants of early efforts to model the brain and cognitions in the 1940s and 1950s (1, 2). Deep learning is also a direct descendant of shallow learning, which can be traced back to the early work on linear regression by Gauss and Legendre. The relatively recent rebranding and expansion of deep learning is powered by trends and progress in two main areas: (*a*) big data and (*b*) computing power, primarily in the form of clusters of CPUs/GPUs (central/graphics processing units) and the cloud. The deep learning expansion has also led to the development of robust, well-maintained deep learning software libraries, such as TensorFlow. These trends have enabled successful applications of deep learning to many engineering areas, ranging from computer vision to speech recognition, natural language processing, and games, to name just a few (3).

With deep learning at the center of artificial intelligence (AI) and machine learning, it should come as no surprise that there are numerous applications of deep learning in biomedical data science. What is less appreciated is that some of the earliest applications of deep learning were precisely in this area and that the complexity of the problems raised by the life sciences have inspired over the years the development of novel deep learning methods. In fact, it is essential to acknowledge a strong two-way relationship between deep learning and the life sciences: In one direction, deep learning helps the life sciences, both broadly, by providing powerful methods for analyzing biomedical data, and more narrowly, by providing simplified but useful computational models for neuroscience. In the other direction, it is our knowledge of the human brain that has provided the fundamental source of inspiration for AI and deep learning, while all areas of the life sciences have provided challenging problems that have inspired researchers to push the boundaries of deep learning methods.

2. BIOMEDICAL DATA

In deep learning, everything begins with the data. Biomedical data can be massive but also extremely heterogeneous, ranging from small molecules to omic data (e.g., genomic, proteomic, transcriptomic, metabolomic), biomedical imaging data, clinical data, and electronic medical records. The data quality can also be very heterogeneous, and the data types can vary from analog to digital and text and can come with complex associated structures such as sequences, trees, and other graphs, which can often vary in size. Central to the application of deep learning methods to biomedical data is the development of methods that can handle different kinds of data, in particular variable-size structured data.

Biomedical data also span many orders of magnitude in both space and time and covers a myriad of different phenomena. Improvement in sensors and other instruments, as well as computers, databases, and the Internet, coupled with the development of novel high-throughput methods such as high-throughput sequencing, has often led to a deluge of data. However, even in the era of big data, the data landscape remains very variable in terms of the volume of available data. Biologists in general have been good citizens in making data, for instance omic data, as publicly available and downloadable as possible. But data are much less readily available in the chemical, pharmaceutical, or clinical sciences, and much remains to be done in these areas in order to overcome the numerous commercial, legal, and other societal barriers to produce sufficient data while also addressing privacy and other legitimate concerns. Overcoming these barriers is essential to enable real scientific progress powered by deep learning that can benefit everyone. Finally, even in some areas where the data appear to be plentiful, it may still not be sufficient to address certain questions. For example, we do have access to thousands of human genome sequences, but not billions, and billions may be required to detect certain subtle effects.

Thus, in short, one must be aware of the complexity and variability of the data landscape. While deep learning applications tend to benefit from the availability of large amounts of data, there are still plenty of areas of research where deep learning must be applied in regimes where data are not so plentiful. In these areas, several techniques can be helpful, ranging from regularization methods, including some relatively new forms of regularization such as dropout (4, 5), to early stopping, semisupervised approaches that try to leverage both labeled and unlabeled data, and data augmentation methods where training or testing sets are expanded using natural or artificially generated data, for instance, by adding appropriate noise or applying appropriate group transformations and other manipulations to the original data.

3. ARCHITECTURES AND ALGORITHMS

The neural networks used in deep learning are networks of simple computational units connected by synaptic weights. The computational units typically compute a weighted average of their inputs, weighted by the incoming synaptic weights, and then apply a linear or nonlinear function to this weighted average. These computational units can be connected into complex architectures. Some of the main architectures that have been developed over the years are as follows. A neural network is feedforward if its connectivity graph does not contain any directed cycles. A neural network is recurrent if its connectivity graph contains at least one directed cycle. A neural network is layered if its neurons can be partitioned into layers with connections generally running from one layer to the next. A neural network is convolutional if it contains layers of units, where each unit has the same synaptic weights but is connected to different patches of the input or the previous layer. Thus, these units apply the same operation to different portions of their inputs, similarly to a mathematical convolution. Convolutional neural networks (CNNs) are typically used in computer vision applications. An autoencoder neural network uses the same data both as input and output target, thereby producing hidden representations of the data in its hidden units, for instance, in compressive fashion. Weight sharing is a design and training technique whereby different neurons in the same network are constrained to have the same weights. For instance, weight sharing is widely used in CNNs. Weight sharing has the effect of reducing the total number of free parameters and thus can help control overfitting. In addition, it can be used to apply the same operation uniformly to different portions of the corresponding input, as in CNNs, to ensure translation-invariant recognition. A siamese neural network typically consists of two identical subnetworks with the same shared weights, together with a third network, often reduced to a simple difference operator, which takes the output of the two subnetworks and produces a final output representing the degree of similarity, or the ordering, between the two inputs presented to the two subnetworks. A long short-term memory (LSTM) is a kind of recurrent neural network building block capable of learning or storing contextual information over different temporal or spatial length scales (6, 7).

It is well known that neural networks have universal approximation properties in the sense that they can implement any Boolean function or approximate any continuous function defined on a compact set to any degree of approximation (8, 9). Learning is the process by which synaptic weights are adjusted in order to implement a function, which is typically known only through examples. Thus, it can also be viewed as a process for storing the information contained in the training examples into the synaptic weights and as a process for writing a computer program from examples. The main algorithm for storing information in the synaptic weights has been stochastic gradient descent with respect to error functions representing the mismatch between what the network does and what it ought to do, measured on the training data. Stochastic gradient descent can be computed efficiently using the backpropagation algorithm, a straightforward application of the chain rule.

A question that often comes up and is often misunderstood is, when is an architecture deep versus shallow? The mathematically correct answer is that as soon as there are hidden units (i.e., units that are not input or output units and thus do not have targets), the network is deep and leads to deep learning. While this review focuses on deep learning in neural networks, it must be noted that other forms of deep learning exist based on graphical probabilistic models (e.g., deep Bayesian networks, Boltzmann machines) and that hidden Markov models (HMMs), which are widely used in biological sequence analysis (10, 11), already implement a form of deep learning since the transitions between hidden states are not observed. Finally, different forms of deep learning can be combined (12), as described below.

3.1. Dealing with Variable-Size Structured Data

Feedforward neural networks are routinely used to process input vectors or tensors of fixed size. For instance, in a computer vision application, all the images can be viewed as fixed-size matrices. However, in many biomedical applications, the data come in a different format, where data items vary in size and are often associated with an underlying graphical structure. This is the case, for example, of (*a*) small molecules, (*b*) nucleotide or amino acid sequences, (*c*) protein or other contact maps, (*d*) phylogenetic trees, (*e*) natural language sequences, and (*f*) natural language parse trees. In many of these cases, the size of the data items can vary greatly, and using a feedforward network tuned to the maximal possible size is usually not a good approach. Thus, in these cases, a recursive network must be used. A recursive neural network is a neural network consisting of submodules that occur repeatedly. A recurrent network unfolded in time is an example of a simple recursive network. Recursive neural network approaches can also be viewed as message-passing algorithms, where variables associated with nodes or edges of a graph are updated as a function of the variables present in their graph neighborhood. The update functions are implemented by neural networks and these networks can be shared in space or time. More concretely, when designing recursive neural networks to process variable-size structured data, it is useful to distinguish two basic approaches: the inner approach and the outer approach (13).

3.2. Inner and Outer Recursive Neural Network Approaches

In the inner approach, the data model must be captured by a directed acyclic graph, and neural networks are used to compute a vector at each node of the graph as a function of the vectors associated with the parent nodes. If the graph has a regular structure, for example, like a chain, grid, or some other lattice, then the networks associated with similar nodes can be shared—hence the recursive aspect. For example, in the case of sequences, these can be modeled using Bayesian networks (14, 15) such as HMMs (or factorial HMMs, input–output HMMs, etc.) associated with directed acrylic graphs. In this case, the inner approach uses two recursive neural networks, one for the transitions and one for the emissions (**Figure 1**). The approach is called inner because the neural networks are used to crawl the graphs associated with the data from the inside. For sequences, it is possible to crawl them in both directions, from the start to the end and vice versa, and this leads to bidirectional recursive neural networks (BRNNs) (16).

In contrast, in the outer approach, neural networks are built recursively in a direction that grows perpendicularly to the data. For instance, in the case of sequences, all the networks in the first generation may look at a fixed window over the input sequence in order to produce an output vector. In the second generation, all the networks may look at a window of consecutive first-generation vectors and produce a second generation of vectors, and so forth (**Figure 2**). Naturally, the inner and outer approaches are not exclusive and can be combined.

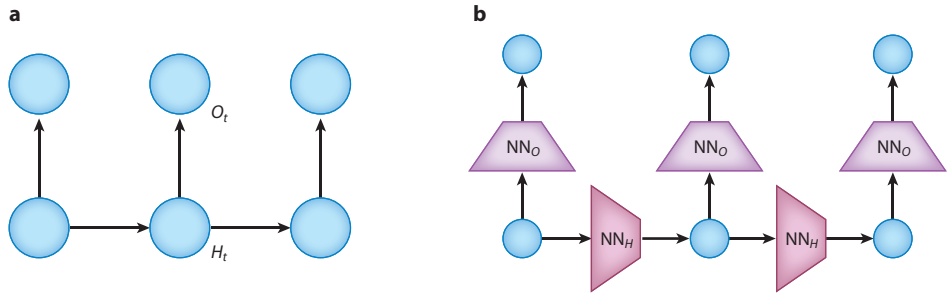

Figure 1

Recursive neural network inner approach for sequences. (*a*) Acyclic graph corresponding to a hidden Markov model for sequences, with hidden state H_t and output O_t at time t. (*b*) Derived recursive neural network consisting of two separate networks, one to compute hidden state vectors and one to compute output vectors, shared at all positions t and trainable by backpropagation because of the acyclic nature of the underlying graph. Adapted from Reference 13.

The next sections survey some of the applications of deep learning to biomedical data, organized roughly by spatial scales into a nonexhaustive list of areas. In some cases, a partial historical perspective of the field is provided, but with no attempt at being exhaustive, as the literature is already very substantial.

4. DEEP LEARNING IN CHEMOINFORMATICS

Small organic molecules—including, for instance, all nucleotides, amino acids, and metabolites—come up in many questions ranging from drug design to natural product retrosynthesis, metabolomic analyses, and molecular bioengineering. Within this broad space, there are two different classes of questions: those related to the molecules themselves and those related to the corresponding chemical reactions.

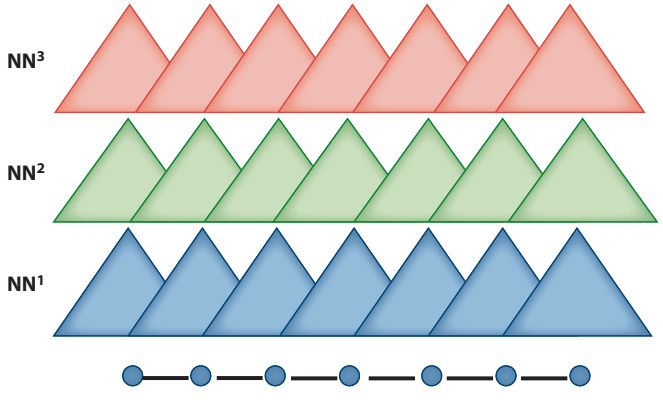

Figure 2

Recursive neural network inner approach for sequences. Neural networks are stacked in a direction perpendicular to the sequence. Networks can be shared within each level of the hierarchy and process data coming from increasingly larger sequence windows as the hierarchy is ascended. Adapted from Reference 13.

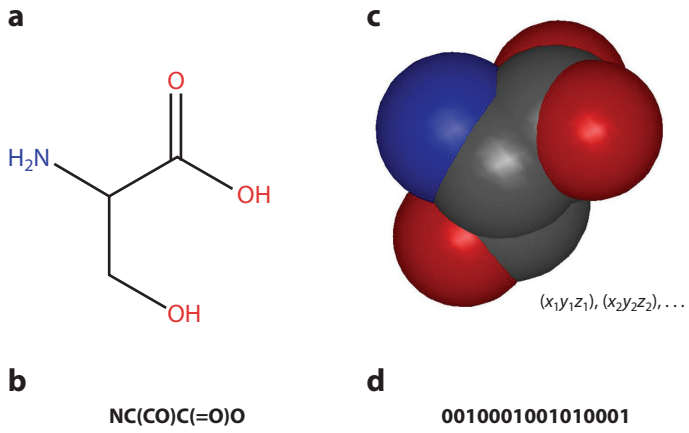

Figure 3

Molecules can be represented in several different ways: (*a*) graphs (or adjacency matrices) labeled by atom and bond types; (*b*) SMILES (simplified molecular-input line-entry system) strings (note that hydrogen atoms are omitted, carbon atoms are implicit, parentheses denote branching, and the equals sign signifies a double bond); (*c*) three-dimensional structures, which can be described by a list of (x, y, z) coordinates associated with the nuclei of the atoms in the molecule; and (*d*) fingerprints consisting of long, sparse, binary vectors of fixed length, where a "1" denotes the presence of a particular feature in the molecule and "0" is absence. A typical feature basis consists of labeled paths or labeled trees up to a certain size.

4.1. Molecules

For the first class of questions, there are on the order of 100 million small molecules that have been discovered in living systems or synthesized by chemists, and these are available in public repositories such as PubChem. Furthermore, the number of virtual small molecules that could be synthesized is astronomical—greater than the number of atoms in the universe. Thus, the central problem here is to be able to predict the physical, chemical, and biological properties of real or virtual small molecules in order to search them. The list of such properties is virtually infinite, including, for example, the prediction of the melting point of a molecule, its degree of solubility in water (often an important consideration for drugs), its degree of affinity for a particular protein, or its degree of toxicity. Unfortunately, publicly available data sets of small molecules annotated with some of their properties tend to be fairly rare and small, raising challenges for deep learning approaches.

Nevertheless, deep learning approaches for the prediction of the properties of small organic molecules have been developed, leveraging in particular the variety of representations that can be used to represent such molecules (**Figure 3**). Indeed, molecules can be represented by labeled graphs or matrices representing atoms and their bonds, strings known as SMILES (simplified molecular-input line-entry system), or lists of atoms with the three-dimensional (3D) coordinates of their nuclei. All these representations are structured and variable in size. Fixed-size vector representations are also possible in the form of molecular fingerprints. These are typically long, sparse, binary vectors, indexing the presence or absence of particular features in the corresponding molecule. Deep learning methods have been developed and can be applied to each of these representations.

For instance, inner approaches in the form of BRNNs and LSTMs can readily be applied to the SMILES sequence format, and similarly for outer approaches. The outer recursive approach has been applied to the graph representation of molecules (17). Application of the inner approach at the level of the molecular graphs requires directed acyclic graphs and thus is faced by two

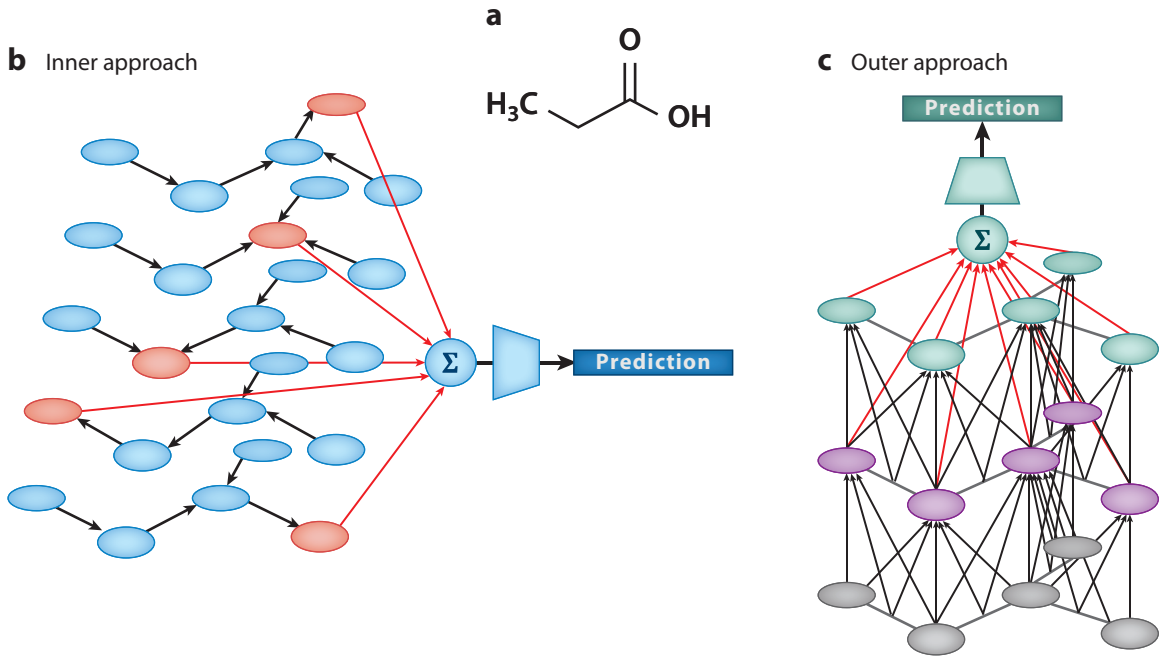

Figure 4

Schematic representation of inner and outer recursive neural network approaches for small molecules represented by labeled graphs (*a*). (*b*) In the inner approach, several copies of the graph are used. In each copy, a different sink node (*red bubbles*) is used, and all the edges are oriented toward the sink node. A shared neural network is used to crawl each acyclic graph from the inside, starting from the source nodes and producing an output vector at each sink node. The resulting output vectors are aggregated and passed to a final prediction neural network. (*c*) In the outer approach, neural networks are stacked in a direction perpendicular to the molecular graph. The shared network in the first layer (*purple bubbles*) receives inputs from a node and its immediate neighbors and produces an output vector associated with each node. A similar process is repeated in the following layers of the stack. The outputs from all layers, or just from the top layer (*green bubbles*), are aggregated and passed to a final prediction neural network.

major challenges: (*a*) The edges of molecular graphs are not directed and (*b*) molecular graphs are not acyclic. One solution (18) to address this problem is essentially to use all possible acyclic orientations of a molecule, using a shared neural network to crawl all these orientations and pass the aggregated crawling results through a final neural network to produce the final prediction. Considering all possible orientations is feasible because small organic molecules have relatively few atoms and bonds (**Figure 4*a***). Application of feedforward neural networks to fixed-size fingerprint vectors is also possible, especially after a dimensionality reduction step to reduce the dimension of typical fingerprint vectors, which tend to be very long and very sparse. In short, this is an area where deep learning methods are in place and what is lacking the most are the data. Finally, it should also be noted that deep learning methods are being applied to facilitate quantum and molecular mechanics calculations (19, 20), although these approaches are still far from being able to scale up to chemical space.

4.2. Reactions

Being able to predict the outcome of chemical reactions, including biochemical reactions, is also a fundamental scientific problem with many applications. Again, deep learning methods have been developed and applied to this problem (21, 22) and can leverage the multiple representations

available both for the reactants and for the reactions. In particular, SMIRKS strings are used to represent reactions, basically by using SMILES strings to represent the reactants and the products, with the important addition of atom numbering and atom mapping between the left- and right-hand sides of the reaction. One possible strategy is to operate at the level of elementary reactions or arrow-pushing mechanisms, which conceptually correspond to the transfer of an electron from an orbital source to an orbital sink. Chemists view global reactions (e.g., bromination of an alkene) as a sequence of elementary reactions. Thus, a reasonable strategy used in the work cited above is, given a set of reactants, to first identify the potential sources and sinks of electrons. A first deep learning approach is used to reduce and rank the list of possible sources and sinks. Given two reactants, there may be 10 potential sources and 10 potential sinks, leading to 100 potential elementary reactions. Ranking these reactions is a machine learning problem that can be addressed by deep learning. Thus, a second deep learning step uses a siamese network to compare source–sink pairs and identify the most favorable ones. The process can be iterated in order to build multistep reactions. Another obvious deep learning approach for reactions is to use recursive networks, for instance, bidirectional LSTM networks, to operate at the level of SMIRKS sequences, either for elementary reactions or for global reactions. Thus, in short, useful deep learning methods are also in place to handle chemical reactions, and one of the main challenges is again to find suitable training data, as there are no large, publicly downloadable databases of chemical reactions.

5. DEEP LEARNING IN PROTEOMICS

High-throughput sequencing and other technologies, including synthetic biology technologies, continue to rapidly increase the number of available protein sequences, significantly outpacing the rate at which these proteins can be characterized experimentally, for instance, by using X-ray crystallography. Thus, tools for predicting protein structural and functional properties play an important role in high-throughput proteomics to sift through large numbers of protein sequences in order to predict their annotations, for instance, in support of integrative systems biology approaches, pathway analyses, drug target discovery, and to identify relevant candidates that can be targeted for experimental and other follow-up studies.

5.1. Protein Structures

For brevity, this section focuses on globular proteins, which represent roughly 75% of a typical proteome. While membrane proteins are very important, and many of the deep learning methods developed for globular proteins can also be applied to membrane proteins (e.g., 23), the training sets available are far smaller due to significantly greater experimental challenges in determining their structures.

Predicted 3D structures, with different degrees of accuracy, can be used for a variety of tasks (24), such as protein classification, inference of functional relationships, identification of functional sites or epitopes, support of site-directed mutagenesis experiments, and molecular docking in virtual screening.

Any approach to 3D structure prediction must deal with the fundamental fact that protein structures are invariant under translations and rotations. Thus, it would not be a good idea to try to predict 3D coordinates directly from the primary structure alone using a neural network. To address this fundamental issue of invariance, one must break down the problem modularly in several steps, using intermediate representations that are translation and rotation invariant, such as the protein contact map or the sequence of its dihedral angles.

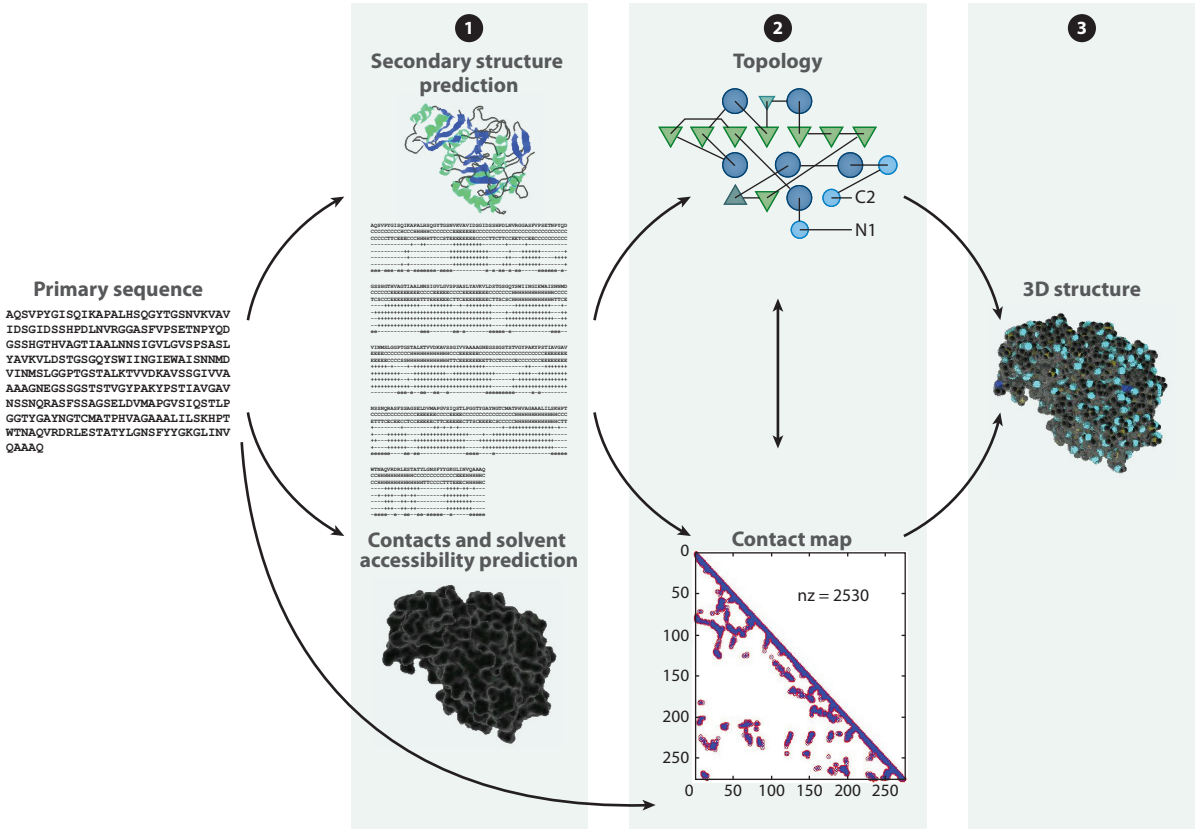

Figure 5

Pipeline of three-dimensional (3D) protein structure prediction by deep learning in three main stages, starting from the protein sequence: (❶) prediction of structural features such as secondary structure and relative solvent accessibility, (❷) prediction of coarse- and fine-grained contact maps invariant to translations and rotations, and (❸) prediction of backbone and side chain 3D coordinates.

An approach based on contact maps is illustrated in **Figure 5** (25–28). More precisely, the first step starts from the protein sequence, possibly in conjunction with evolutionary profiles, to leverage evolutionary information and predicts several structural features, such as classification of amino acids into secondary structure membership classes (e.g., alpha helices, beta strands, coils) or relative exposure classes (e.g., surface, buried) (16, 29–31).

The second step uses the primary structure and the structural features to predict a topological representation in terms of contact or distance maps. The contact map is a 2D representation of neighborhood relationships consisting of an adjacency matrix at some distance cutoff, typically in the range of 6–12 Å at the amino acid level. The distance map replaces the binary values with pairwise Euclidean distances. Fine-grained contact or distance maps are derived at the amino acid level, whereas coarse-grained contact or distance maps can be derived at the level of secondary structure elements, for instance, from the pairwise distances between the centers of gravity of these elements.

The third step in the pipeline predicts the actual 3D coordinates of the atoms in the protein using the constraints provided by the previous steps, primarily from the 2D contact maps, but also possibly from other constraints provided by physical or statistical potentials. Effective methods to

address the third step have been developed in the NMR (nuclear magnetic resonance) literature (32, 33) and elsewhere (34, 35) and typically use ideas from distance geometry, molecular dynamics, and stochastic optimization to recover 3D coordinates from contacts. Suffice it to say that the problem of recovering a 3D structure, up to mirror symmetries, from a good contact map is essentially solved. Clearly, it is the distant contacts that are important and difficult to predict, and simulations show that recovering about 50% of the distant contacts is sufficient in most cases to derive good 3D structures. Thus, deep learning methods must focus on the first and second steps of the pipeline.

5.2. Protein Secondary Structure and Other Structural Features

In the first step of the pipeline, the prediction of secondary structure and, to a lesser extent, relative solvent accessibility, has played a central role in this field. Chou & Fasman (36, 37) first developed a simple statistical approach for predicting the location of helices, strands, and coils along the protein sequence—basically by looking at amino acid propensities—using the small data available in the Protein Data Bank (PDB) in the 1970s. The accuracy at the time was around 60%. A first breakthrough, and probably the first application of deep learning in computational biology, came a few years later (38), where a fully connected network with a single hidden layer was able to achieve an accuracy of about 64%, rapidly followed by other similar applications (39, 40). The next breakthrough (41, 42) brought the accuracy to about 74%. While this work used more sophisticated feedforward deep learning architectures, the major contribution at the time was the realization of the importance of using evolutionary profiles in the input, as opposed to the primary structure alone. Other improvements rapidly followed (43). With the benefit of hindsight, up to this time, all the deep learning applications to secondary structure prediction were essentially based on the outer, or convolutional, approach. In the late 1990s, the first inner approaches were developed in the form of BRNNs, and these brought the accuracy to around 80% (16, 31); similar results were also obtained with large ensembles of feedforward networks (44). Using ensembles of BRNNs, 82.3% is the best accuracy reported so far without using alignments to PDB sequences to refine the results at the output level (45, 46). Finally, using BRNNs but also alignments to sequences in the PDB, the accuracy has been further increased to about 95% (47), suggesting that the problem of secondary structure prediction is now basically solved.

Indeed, there are fundamental reasons why an accuracy of 100% should not be expected: (*a*) the presence of disordered regions; (*b*) the ambiguities inherent in the definitions of secondary structure or relative solvent accessibility, as reflected by the imperfect correlation between several programs for determining these features from PDB files; (*c*) the errors and uncertainties contained in the PDB; and (*d*) the structure-determining role of the solvent and other molecules, from ions to chaperone proteins, which are not taken into consideration by the current prediction methods.

Thus, in about three and a half decades, through a series of twists and turns, the accuracy for secondary structure prediction has gone from 60% to 95%. Similar trends can be seen for the prediction of secondary structure into eight classes or for other structural features such as relative solvent accessibility. The challenges have spearheaded the creation of new deep learning methods, which have helped solve the problem, although it is fair to say that the increase in the number of sequences and structures in the protein databases has played the major role in the improvements. Although the main challenges in this highly studied area have been addressed, several open problems still remain. After all, unlike the deep learning methods discussed above, when proteins fold in vivo or even in vitro, they do not use sequence or structural similarity at all. This suggests that our understanding of protein structural features is far from complete and

points to at least two interesting technical challenges for the foreseeable future: (*a*) to predict secondary structure and other structural features with an accuracy of about 80% or more, using no similarity to known proteins at all (i.e., no input profiles), and (*b*) to predict secondary structure and other structural features with an accuracy of 85% or more, using sequence similarity alone (profiles), but no structural similarity. Key to addressing these challenges, besides increasing the size of the training sets, may be the development of deep learning methods that can better exploit information contained in larger input windows, ideally covering the entire protein length.

5.3. Protein Contacts and Contact Maps

In the second step of the pipeline, there has also been progress on the problem of predicting contact or distance maps using deep learning; however, this problem is harder and still unsolved. Various algorithms for the prediction of amino contacts (30, 48–51), contact numbers (52), distances (53–56), and coarse- and fine-grained contact maps (57–61) have been developed. Deep learning methods for special classes of contacts, such as disulphide bridges (62), have also been developed. In terms of methods, the 1D inner approach based on BRNNs, used for secondary structure prediction, can be generalized to two dimensions for contact map prediction (**Figure 6*a***). This gives rise to grid recursive neural networks with one input grid, four hidden grids oriented toward each one of the four cardinal corners, and one output grid corresponding to the predicted contacts. The corresponding deep architecture can be implemented using five recursive neural networks, one for computing the output as a function of the input vector and the four hidden vectors, and one for each hidden grid to compute the corresponding hidden vector as a function of its grid neighbors and the input vector (26, 27).

In terms of outer approaches, it is possible to consider layered architectures, where shared neural networks (within layers or across layers) operate on the contact map to progressively refine it (63). While targets for the intermediate contact maps are usually not available, the targets provided by the known contact maps are available at the top of this process, and error gradients can be backpropagated through the entire stack due to the acyclic nature of the underlying graphs (**Figure 6*b***).

Additional work in this area includes the application of deep learning, using stacked autoencoders, to the prediction of dihedral angles (64). Deep learning methods have also been applied to address the problems of predicting the coordinates of side chains (65) and to begin to tackle problems of quaternary structure prediction (66). In recent years, it has become apparent, again using alignments, that correlated mutations (67, 68) provide a powerful tool for inferring contacts, and these methods are being incorporated into deep learning algorithms (69). While good progress is being made, this is an area where data again play a critical role and additional data are needed. While the PDB has been growing rapidly, after redundancy reduction, it still contains only a few thousand sequences available for ab initio contact prediction training.

Finally, there is also a fairly extensive body of work using similar shallow and deep learning methods to predict other features related to structure, including stability (70), solubility (71), disordered regions (72, 73), domains (74), fold recognition (75, 76), short linear binding regions (77), binding regions within disordered regions (78), and bioactive peptides (79) within proteins.

This is also an area where other forms of deep learning, in terms of deep graphical models, have been used. For instance, in Reference 80, an ensemble of stacks of restricted Boltzmann machines is used to predict residue–residue contacts from sequences. Similarly, deep belief network methods are used in Reference 81 to predict disordered regions, in Reference 82 to predict secondary structures, and in Reference 83 for fold recognition and template ranking (see also Reference 84).

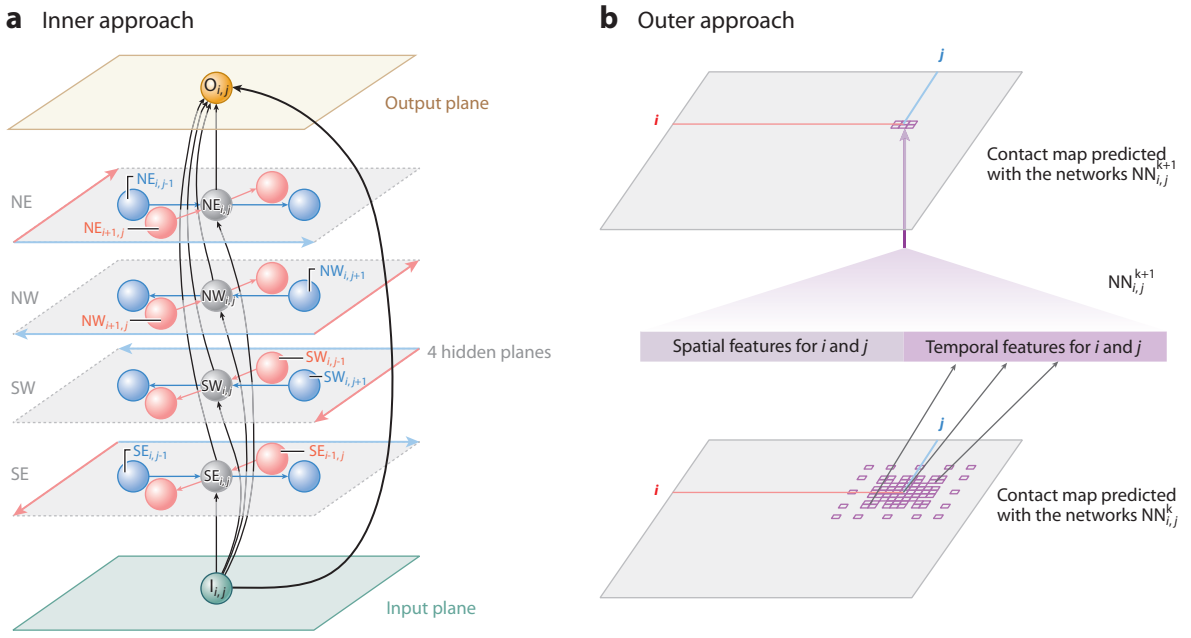

Figure 6

Schematic representation of recursive neural network inner and outer approaches for contact map prediction. (*a*) In the inner approach, the prediction at position (i, j) is computed by a shared neural network as a function of the input vector and four hidden vectors at position (i, j). The input vector may contain information coming from a window around amino acid i and amino acid j, including sequence and profile information, as well as predicted features such as secondary structure and relative solvent accessibility. Each of the four hidden vectors resides in a different hidden plane. The directed acyclic graph associated with each hidden plane is a grid where all the edges are oriented toward one of the four cardinal corners. For instance, the hidden vector in the northeast hidden plane at position (i, j) is computed by a neural network, which is shared within the plane, as a function of the corresponding input vector and the two neighboring hidden vectors in the same grid. Thus, this neural network is used to crawl the northeast hidden grid. In total, five neural networks are first trained and then used recursively to produce the predictions. (*b*) In the outer approach, neural networks are stacked on top of the contact map plane. Only one level of the stack is shown on the right. At level $k + 1$ of the stack, there is a shared neural network common to all (i, j) positions in that level. The input to the network consists of spatial or static features, corresponding to the same input vector used in the inner approach, and temporal or dynamic features, corresponding to a local receptive field of hidden vectors produced at the previous level in the stack. Intuitively, each level tries to refine the predictions produced by the previous level. Adapted from References 26 and 63.

5.4. Protein Functional Features

This section briefly considers the prediction of other features that are not directly related to the structure of a protein in any obvious way and often inform its function. These include the global classification of a protein into classes (e.g., kinase, RNA-binding protein) or whether it contains a particular signal or binding site (e.g., signal peptide, glycosylation site). These predictions are often made from the protein sequence, possibly augmented with profiles, protein sequence composition (e.g., frequencies of amino acids and amino acid pairs), protein length, and structural features (e.g., secondary structure, relative solvent accessibility). Seminal early work in this area has been the development of effective deep learning methods for identifying prokaryotic and eukaryotic signal peptides (85) leading to the SignalP software. Signal peptides control the entry of virtually all proteins to the secretory pathway in both eukaryotes and prokaryotes, and the SignalP software has been one of the most successful software in bioinformatics. SignalP (85) trains two neural networks from the same data: a signal peptide/nonsignal peptide network, whose output can be

interpreted as the probability of the input position belonging to the signal peptide, and a cleavage site/noncleavage site network, whose output can be interpreted as the probability of the position being the first in the mature protein (position +1 relative to the cleavage site). Output from the two networks are then combined to produce the final prediction. Similar work has been done for a host of other signals and protein properties, such as posttranslational glycosylation and phosphorylation of proteins (86), showing that useful functional properties of proteins can be robustly inferred even when the structure is not known (87). Outer deep learning approaches have also been used to predict, for instance, subcellular localization (88), peptide bioactivity (89, 90), and various forms of protein classification (91, 92).

Finally, inner and outer deep learning methods have also been applied to various problems in computational immunology (93), for instance to predict T cell and B cell epitopes (94–96), protein antigenicity (97), and MHC (major histocompatibility complex) peptide binding (98).

6. DEEP LEARNING IN GENOMICS AND TRANSCRIPTOMICS

This section looks at deep learning and its applications to DNA, RNA, gene, and genome sequences, as well as gene expression. The earliest precursor in this area was the use of nonlinear shallow learning to predict ribosome binding sites using the perceptron algorithm (99). First systematic applications of deep learning methods in this area were focused on the prediction of splice sites (100, 101) and coding regions (102). Some of this work has also led to interesting approaches for detecting error in databases (103). The development of HMM-based methods and the abundance of genomic sequences have somewhat reduced the interest in these approaches. Another relatively early and orthogonal application of deep learning to the genomic field was actually for the problem of base calling in early sequencers (104, 105), a problem that still continues today with new technologies (106).

Not surprisingly, current modern applications of deep learning in genomics are focusing on the analysis of actual DNA or RNA sequences and the inference of functional properties and phenotypic consequences associated with mutations. For instance, CNNs are used in Reference 107 to predict sequence specificities of DNA- and RNA-binding proteins, and further refinements have used convolutional recurrent neural networks and LSTMs (108). A related cell-specific application using ChIP-seq (chromatin immunoprecipitation sequencing) data is developed in Reference 109, complemented by Reference 110, where convolutional recurrent neural networks in the form of convolutional bidirectional LSTMs are used to predict cell type–specific transcription factor binding using reference binding data from ChIP-seq and DNase-seq (DNase I hypersensitive sites sequencing) data. In Reference 111, CNNs are used to predict chromatin accessibility, histone modification, and transcription factor binding from raw DNA sequences, and the predictions are subsequently used to infer the functional effect of noncoding single nucleotide polymorphisms (SNPs) or indels. This work has been complemented (112) using features from annotated data, such as evolutionary scores or the distance to the nearest transcription start site, and extended (113) using convolutional recurrent neural networks, including bidirectional LSTMs (**Figure 7**). Related work (114) has also used CNNs to predict open chromatin regions from DNase-seq data with the goal of better understanding the regulatory code and assessing the functional effects of SNPs. Deep learning in the form of CNNs and recurrent neural networks, specifically bidirectional gated recurrent networks, has also been used to predict the methylation state of CpG dinucleotides at the single-cell level (115).

A different application of CNNs to the problem of predicting and understanding sequence conservation is described in Reference 116. Finally, a hybrid architecture combining a pretrained deep neural network and an HMM has been applied to the identification of replication domains

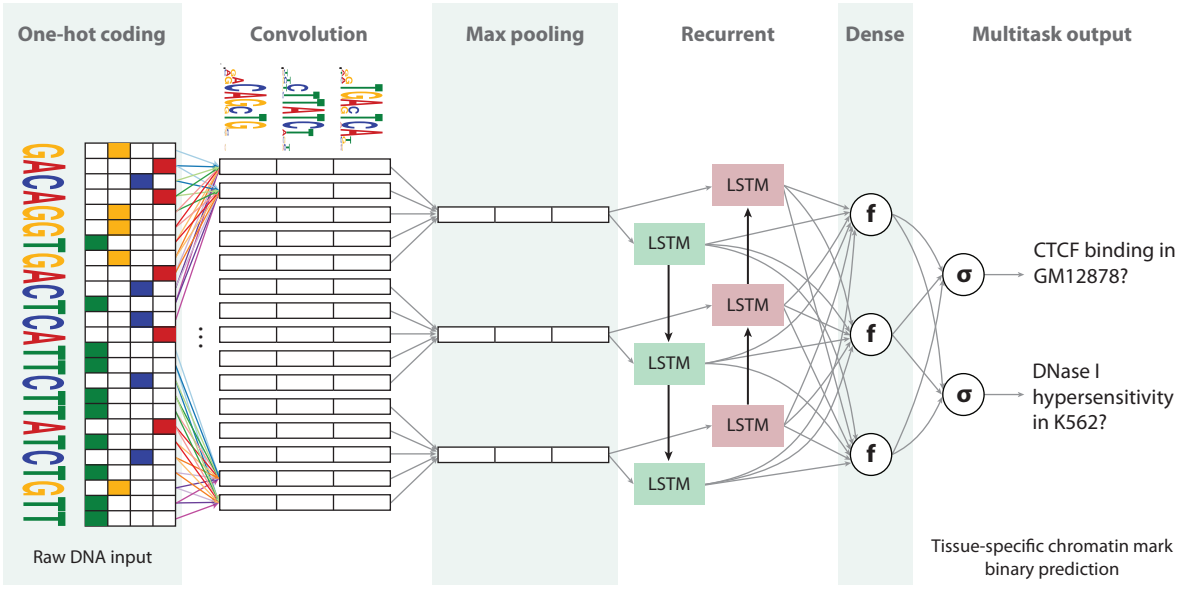

Figure 7

Deep hybrid convolutional and bidirectional long short-term memory (LSTM) network for DNA sequence analysis (113). The input DNA sequence is first one-hot encoded into a four-row bit matrix. A convolution layer acts as a motif scanner across the input matrix to produce an output matrix with a row for each convolution kernel and a column for each position in the input (minus the width of the kernel). Max pooling reduces the size of the output matrix along the spatial axis, preserving the number of channels. The subsequent bidirectional LSTM layer considers the orientations and spatial distances between the motifs. The bidirectional LSTM outputs are flattened into a layer as inputs to a final feedforward neural network, which produces a vector of probability predictions of epigenetic marks to be compared via a loss function to the true target vector. For example, the model will predict whether the protein CTCF in the GM12878 cell line binds to the input sequence or whether the input sequence is DNase I hypersensitive in the K562 cell line.

using replication timing profiles (117), and deep restricted Boltzman machines and belief networks have been applied to the prediction of RNA-binding protein binding sites (118) from RNA sequences and their profiles, as well as predicted RNA secondary and tertiary structural features.

Finally, on the gene annotation and transcriptomic sides of things, one of the earliest applications of deep learning is Reference 119, where neural networks are used in an unsupervised manner to cluster gene expression patterns. More recently, deep autoencoders are used in Reference 120 for gene ontology annotation predictions, and deep feedforward networks are used to predict patterns of alternative splicing using mouse RNA-seq (RNA sequencing) data (121) and to predict splicing in human tissue and its relationship to disease (122). Feedforward neural networks are also used in Reference 123 for gene expression regression, specifically for inferring the expression level of all the genes using the expression levels of only ∼1,000 landmark genes. Finally, there have been recent applications of deep learning in circadian biology. Deep feedforward neural networks, trained primarily on synthetic data, have been used to identify which species (transcripts, proteins, or metabolites) are periodic in the corresponding omic time series, in particular, to identify circadian patterns of oscillations. In addition, compressive autoencoders with sine and cosine bottleneck units have been trained to infer time from single-time-point transcriptome experiments (124) in the GEO (Gene Expression Omnibus) database.

7. DEEP LEARNING IN BIOMEDICAL IMAGING

It is not surprising that CNNs are the workhorse of applications of deep learning to biomedical images. The earliest systematic application of deep learning to biomedical image data is found in Reference 125, where deep siamese CNNs are developed and used in a biometric application to recognize fingerprints. In recent years, CNNs and their variants have achieved superhuman performance (126) in image classification and other vision tasks, and thus applications to biomedical images are rapidly expanding. One example is the application of deep neural networks to each pixel of electron microscopy images to segment neuronal membranes (127). On the medical side, many applications have focused on cancer. For instance, in Reference 128, deep neural networks are used for mitosis detection in breast cancer histology images; in Reference 129, a deep CNN is used in the second tier of a two-tiered, coarse-to-fine cascade framework to refine the candidate lesions from the first tier for sclerotic spine metastases detection in computed tomography (CT) images; in Reference 130, multiscale CNNs are developed for lung nodule detection in CT images; in Reference 131, a GoogLeNet-based method is adopted for automated detection and diagnosis of metastatic breast cancer in whole-slide images of sentinel lymph node biopsies; in Reference 132, a multiresolution approach using deep siamese neural networks is developed for spinal metastasis detection; finally, an application to skin cancer identification is described in Reference 133. There have been also applications of deep learning to other diseases, such as the detection of diabetic retinopathy in retinal fundus photographs (134), or to regression problems (as opposed to classification problems); for instance, in Reference 135, CNNs are used to process mammograms, infer the degree of breast calcification, and assess the corresponding risk of cardiovascular disease (**Figure 8**).

Several trends and technical points are worth noticing in this area. First, essentially the same techniques can be applied to several different imaging modalities and scales, such as electron microscopy, ultrasound imaging, photographs, video, CAT (computed axial tomograph) scans, MRI (magnetic resonance imaging) scans, X-ray scans, and so forth.

Second, there are publicly available deep CNNs that have already been trained on large image data sets for general purpose computer vision, sometimes achieving superhuman performance (126). These trained networks can easily be reused in many biomedical imaging problems simply by slightly modifying the top of the architecture and retraining just the top, thus reusing the already pretrained, general-purpose, low-level feature detectors. It is also possible to use the pretrained architecture to initialize the weights but then retrain everything. Using pretrained architectures often provides excellent performance and also a comparison baseline for training newly designed, problem-specific architectures. These pretrained architectures usually have many parameters; however, the danger of overfitting is limited because they have already been trained with many images.

Third, in applications where the size or the speed of the architecture becomes an issue (e.g., deep learning on cell phones), custom architectures need to be trained. In some cases, there are also approaches for converting deep architectures into almost-as-good shallow ones, a process sometimes called model compression. This can be achieved by first training a deep architecture and then training a shallow architecture using as targets the analog outputs of the trained deep architecture. These analog outputs contain more information (also called dark knowledge) than the binary targets associated with the original training set in classification tasks (136, 137).

Fourth, deep learning can be used not only for classification or regression tasks on biomedical images, but also for localization and segmentation. Given a training set that contains bounding boxes around regions of interest (e.g., polyps), one can train a network that does both classification (e.g., presence or absence of polyps) and regression on the location and size of the bounding boxes

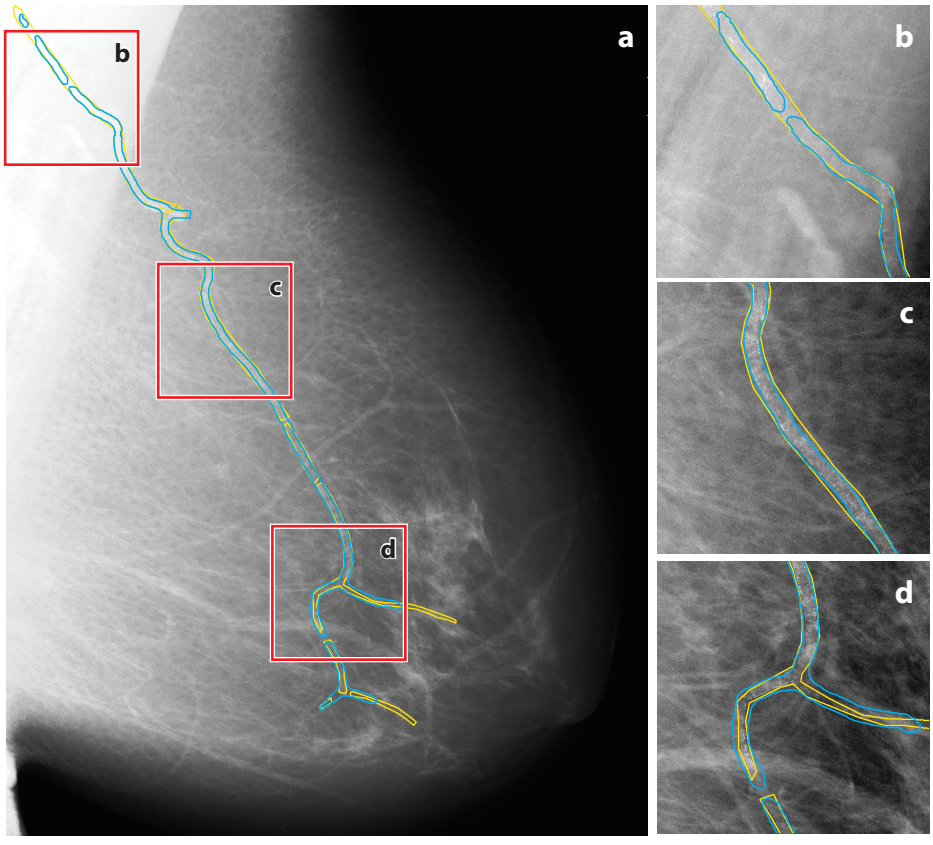

Figure 8

(*a*) Application of deep learning to the detection of breast arterial calcifications (BACs) in mammogram images, in which the true BACs are marked by yellow contours, and the automatically detected regions are marked by blue contours. (*b–d*) The magnified views of the three regions of interest in panel *a* (*red squares*). Deep learning networks are used to automatically identify BACs and predict the overall level of calcification (135).

(138). For instance, segmentation can be cast as a classification task at the pixel level and therefore is usually more time consuming, but it can be addressed using deep learning (139, 140) and carried off-line.

Fifth, it is often the case that biomedical images are actually 3D: two spatial dimensions and one temporal dimension (videos), or three spatial dimensions (scans). Training deep architectures can take days or weeks, but once trained, they can process one image in a few milliseconds. Thus, one can easily process video data, as in the case of colonoscopies (141), by identifying the presence/absence and location of polyps in multiple frames within each second and using the results of neighboring frames to produce more robust inferences. Likewise, in the case of three spatial dimensions, it is possible to process multiple 2D slices and use one slice and its neighboring slices to derive more robust inferences (e.g., 132). Finally, it is also possible to use 3D CNNs. 3D convolutions are computationally expensive but can be done for relative small volumes (e.g., $3 \times 3 \times 3$) with current technology. Large 3D convolutional filters still present significant computational challenges.

In any case, there is a general sense that this is an area where all the basic technical problems are already solved, and there ought to be an explosion of applications in the coming years. As larger and larger data sets are aggregated and used for training, and as societal and other barriers are addressed, one can expect an increasing number of deep learning systems to make their way into real biological and medical applications.

8. DEEP LEARNING IN HEALTH CARE

There is also a rapidly growing set of applications of deep learning to electronic medical records and health care data, recently reviewed in References 142 and 143, as well as related concerns (144). Examples of work in this area include detection of cardiac arrhythmia (145) from electrocardiograms; prediction of postoperative in-hospital mortality (146) from clinical data; phenotype discovery using clinical data (147); and use of neural network autoencoders and topic modeling techniques to learn compact, useful-for-prediction representations of electronic health records (148) (see also 149). One important challenge in the latter area is how to incorporate temporal information, for instance, the notion of disease trajectories (150), into end-to-end, specialized or more generic, predictive systems (151, 152).

9. CONCLUSION: THE BLACK BOX QUESTION

As should be obvious from this necessarily nonexhaustive review, there have been plenty of deep learning applications to biomedical data over the past 35 years, and the pace is only increasing. As with any exponential growth phase, what remains ahead exceeds what has already been done, and one can expect these applications to continue to expand and yield important scientific, biomedical, and clinical breakthroughs. As in the case of biomedical imaging, one may predict that deep learning will become a commodity—in fact, in many cases, the technical problems have already been solved. The real challenges lie in the data and the various societal barriers to technological changes. While there is still plenty of room for applying deep learning methods to narrowly focused problems, it is clear that one important direction of research is to broaden their applicability toward general AI for biomedical data and beyond.

This review has examined how deep learning can be applied to make sense of biomedical data. However, there is a second, narrower application of deep learning in neuroscience, where deep learning approaches, including deep reinforcement learning, can be used to build coarse models of brain circuitry, brain learning, and brain function. This line of research also has a long history (153) and has gone through several ups and downs. While in the down periods, artificial neural networks have been dismissed as being overly simplistic to the point of having little to do with the brain, it is becoming increasingly clear that this is not the case (154) and that neural networks provide a fundamental model for a style of computing where information is stored holographically in synaptic weights, rather than in digital memory addresses (155, 156).

Finally, one important issue that comes up all the time when neural networks are trained from data is the black box issue: the sense of insecurity that arises from not knowing how a neural network solves a particular task. The significance of this issue is compounded by recent developments in a variety of adversarial methods that can fool neural networks. This is a complex question that requires a multifaceted answer.

First, a certain degree of opacity is inevitable. After all, training is a way of storing the training data into the synapses of a network. But synaptic storage is fundamentally different from the digital, address-based storage of our computers. It is a holographic form of storage in the sense that each training example makes a small contribution to each synaptic weight. Unlike digital storage of the

training examples, synaptic storage of the training examples is inherently opaque: The training examples cannot be recovered from the weights in a fundamental way. This is already the case for linear regression. The weights in linear regression tell you something about the sufficient statistics of the data (e.g., mean, covariance) but do not allow one to recover the training set.

Second, if one really wants to, it is possible to open the black box, essentially by conducting neurophysiological experiments and studying the behavior of each neuron under all kinds of inputs. This is often prohibitively time consuming and can only be done in favorable circumstances for a subset of neurons (e.g., neurons in the early layers of a CNN). Other weaker but faster techniques have been developed to explain the behavior of a trained neural network essentially by propagating the information backward (157) or by training a simple classifier using examples in the close neighborhood of a test example (158).

Third, the practical solution to address the problem is to build a modular system where particular behaviors or errors can easily be isolated and corrected. For example, a deep learning system trained to classify images as containing polyps or no polyps may be opaque. But if the system also draws bounding boxes around the polyps it finds, then in the case of a false positive, the bounding box provides a fairly precise example of signal that leads to confusion. This provides clues as to why the network fails and how to fix it. Likewise, a mechanistic system for predicting a chemical reaction allows one to isolate which mechanistic step—within a chain of steps—is wrongly predicted, and within that step, which are the sources and sinks of electrons that were mislabeled or the source–sink pairs that were improperly ranked, again providing important clues for understanding the errors and the avenues for fixing them.

Fourth, while the desire to understand and control is fundamental to human nature, one must be honest and recognize that most of the objects we use and trust everyday are black boxes. Most of us do not know how our cars, computers, or cell phones work in detail. Wonder of wonders, each and every one of us has no clue of how his brain works yet trusts it blindly at every second of life. As the man who introduced the term "synapse" put it,

> A shower of little electrical leaks conjures up for me, when I look, the landscape; the castle on the height, or, when I look at him, my friend's face, and how distant he is from me they tell me. Taking their word for it, I go forward and my other senses confirm that he is there. (159, p. 113)

DISCLOSURE STATEMENT

The author is not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

This work was supported in part by NIH grant GM123558 and DARPA grant HR0011624519. I would like to acknowledge useful discussions with Jianlin Cheng, Klaus-Robert Müller, Gianluca Pollastri, Daniel Quang, Jürgen Schmidhuber, David Westergaard, and Xiaohui Xie.

LITERATURE CITED

1. McCulloch W, Pitts W. 1943. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* 7:115–33
2. Rosenblatt F. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65(6):386–408

3. Schmidhuber J. 2015. Deep learning in neural networks: an overview. *Neural Netw*. 61:85–117
4. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learning Res.* 15(1):1929–58
5. Baldi P, Sadowski P. 2014. The dropout learning algorithm. *Artif. Intell.* 210:78–122
6. Hochreiter S, Schmidhuber J. 1997. Long short-term memory. *Neural Comput*. 9(8):1735–80
7. Greff K, Srivastava RK, Koutnk J, Steunebrink BR, Schmidhuber J. 2017. LSTM: a search space odyssey. *IEEE Trans. Neural Netw. Learning Syst.* 28(10):2222–32
8. Cybenko G. 1989. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* 2(4):303–14
9. Hornik K, Stinchcombe M, White H. 1990. Universal approximation of an unknown function and its derivatives using multilayer feedforward networks. *Neural Netw*. 3:551–60
10. Krogh A, Brown M, Mian IS, Sjölander K, Haussler D. 1994. Hidden Markov models in computational biology: applications to protein modeling. *J. Mol. Biol.* 235:1501–31
11. Baldi P, Chauvin Y, Hunkapillar T, McClure M. 1994. Hidden Markov models of biological primary sequence information. *PNAS* 91(3):1059–63
12. Baldi P, Chauvin Y. 1996. Hybrid modeling, HMM/NN architectures, and protein applications. *Neural Comput*. 8(7):1541–65
13. Baldi P. 2018. The inner and outer approaches for the design of recursive neural networks architectures. *Data Min. Knowl. Discov*. 32(1):218–30
14. Koller D, Friedman N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: MIT Press
15. Murphy KP. 2012. *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press
16. Baldi P, Brunak S, Frasconi P, Pollastri G, Soda G. 1999. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics* 15:937–46
17. Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, et al. 2015. Convolutional networks on graphs for learning molecular fingerprints. *Proc. Int. Conf. Neural Inf. Process. Syst., 28th, Montreal, Can., 7–12 Dec.*, ed. C Cortes, DD Lee, M Sugiyama, R Garnett, pp. 2224–32. Cambridge, MA: MIT Press
18. Lusci A, Pollastri G, Baldi P. 2013. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. *J. Chem. Inf. Model.* 53(7):1563–75
19. Schütt KT, Arbabzadah F, Chmiela S, Müller KR, Tkatchenko A. 2017. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* 8:13890
20. Chmiela S, Tkatchenko A, Sauceda HE, Poltavsky I, Schütt KT, Müller K-R. 2017. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv*. 3(5):e1603015
21. Kayala MA, Azencott CA, Chen JH, Baldi P. 2011. Learning to predict chemical reactions. *J. Chem. Inf. Model.* 51(9):2209–22
22. Kayala MA, Baldi P. 2012. Reactionpredictor: prediction of complex chemical reactions at the mechanistic level using machine learning. *J. Chem. Inf. Model.* 52(10):2526–40
23. Randall A, Cheng J, Sweredoski M, Baldi P. 2008. TMBpro: secondary structure, β-contact and tertiary structure prediction of transmembrane β-barrel proteins. *Bioinformatics* 24(4):513–20
24. Baker D, Sali A. 2001. Protein structure prediction and structural genomics. *Science* 294:93–96
25. Baldi P, Pollastri G. 2002. A machine learning strategy for protein analysis. *IEEE Intell. Syst.* 17(2):28–35
26. Baldi P, Pollastri G. 2003. The principled design of large-scale recursive neural network architectures—DAG-RNNs and the protein structure prediction problem. *J. Mach. Learn. Res.* 4:575–602
27. Cheng J, Randall AZ, Sweredoski M, Baldi P. 2005. Scratch: a protein structure and structural feature prediction server. *Nucleic Acids Res.* 33:W72–76
28. Cheng J, Tegge AN, Baldi P. 2008. Machine learning methods for protein structure prediction. *IEEE Rev. Biomed. Eng.* 1:41–49
29. Baldi P, Brunak S. 2001. *Bioinformatics: The Machine Learning Approach*. Cambridge, MA: MIT Press. 2nd ed.
30. Pollastri G, Baldi P, Fariselli P, Casadio R. 2001. Prediction of coordination number and relative solvent accessibility in proteins. *Proteins* 47:142–53

31. Pollastri G, Przybylski D, Rost B, Baldi P. 2001. Improving the prediction of protein secondary strucure in three and eight classes using recurrent neural networks and profiles. *Proteins* 47:228–35
32. Nilges M, Clore GM, Gronenborn AM. 1988. Determination of three-dimensional structures of proteins from interproton distance data by dynamical simulated annealing from a random array of atoms. *FEBS Lett*. 239:129–36
33. Nilges M, Clore GM, Gronenborn AM. 1988. Determination of three-dimensional structures of proteins from interproton distance data by hybrid distance geometry-dynamical simulated annealing calculations. *FEBS Lett*. 229:317–24
34. Vendruscolo M, Kussell E, Domany E. 1997. Recovery of protein structure from contact maps. *Fold. Des.* 2:295–306
35. Vassura M, Margara L, Di Lena P, Medri F, Fariselli P, Casadio R. 2008. FT-COMAR: fault tolerant three-dimensional structure reconstruction from protein contact maps. *Bioinformatics* 24(10):1313–15
36. Chou PY, Fasman GD. 1974. Conformational parameters for amino acids in helical, β-sheet, and random coil regions calculated from proteins. *Biochemistry* 13(2):211–22
37. Chou PY, Fasman GD. 1974. Prediction of protein conformation. *Biochemistry* 13(2):222–45
38. Qian N, Sejnowski TJ. 1988. Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.* 202:865–84
39. Bohr H, Bohr J, Brunak S, Cotterill RMJ, Lautrup B, et al. 1988. Protein secondary structure and homology by neural networks. The α-helices in rhodopsin. *FEBS Lett*. 241(1–2):223–28
40. Holley LH, Karplus M. 1989. Protein secondary structure prediction with a neural network. *PNAS* 86(1):152–56
41. Rost B, Sander C. 1997. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.* 232:584–99
42. Rost B, Sander C. 1994. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins* 19:55–72
43. Riis SK, Krogh A. 1996. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *J. Comput. Biol.* 3:163–83
44. Petersen TN, Lundegaard C, Nielsen M, Bohr H, Bohr J, et al. 2000. Prediction of protein secondary structure at 80% accuracy. *Proteins* 41(1):17–20
45. Mooney C, Pollastri G. 2009. Beyond the twilight zone: automated prediction of structural properties of proteins by recursive neural networks and remote homology information. *Proteins* 77(1):181–90
46. Mirabello C, Pollastri G. 2013. Porter, PaleAle 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. *Bioinformatics* 29(16):2056–58
47. Magnan CN, Baldi P. 2014. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning, and structural similarity. *Bioinformatics* 30(18):2592–97
48. Shindyalov IN, Kolchanov NA, Sander C. 1994. Can three-dimensional contacts of proteins be predicted by analysis of correlated mutations? *Protein Eng*. 7:349–58
49. Olmea O, Valencia A. 1997. Improving contact predictions by the combination of correlated mutations and other sources of sequence information. *Fold. Des.* 2:S25–32
50. Fariselli P, Casadio R. 1999. Neural network based predictor of residue contacts in proteins. *Protein Eng*. 12:15–21
51. Fariselli P, Casadio R. 2000. Prediction of the number of residue contacts in proteins. *Proc. Conf. Intell. Syst. Mol. Biol., 8th, La Jolla, Calif., 16–23 Aug.*, ed. R Altman, pp. 146–51. Menlo Park, CA: AAAI Press
52. Pollastri G, Baldi P, Fariselli P, Casadio R. 2001. Improved prediction of the number of residue contacts in proteins by recurrent neural networks. *Bioinformatics* 17:S234–42
53. Aszodi A, Gradwell MJ, Taylor WR. 1995. Global fold determination from a small number of distance restraints. *J. Mol. Biol.* 251:308–26
54. Lund O, Frimand K, Gorodkin J, Bohr H, Bohr J, et al. 1997. Protein distance constraints predicted by neural networks and probability density functions. *Prot. Eng.* 10(11):1241–48
55. Gorodkin J, Lund O, Andersen CA, Brunak S. 1999. Using sequence motifs for enhanced neural network prediction of protein distance constraints. *Proc. Conf. Intell. Syst. Mol. Biol., 7th, Heidelberg, Ger., 6–10 Aug.*, ed. T Lengauer, R Schneider, P Bork, D Brutlag, J Glasgow, et al., pp. 95–105. Menlo Park, CA: AAAI Press

56. Kukic P, Mirabello C, Tradigo G, Walsh I, Veltri P, Pollastri G. 2014. Toward an accurate prediction of inter-residue distances in proteins using 2D recursive neural networks. *BMC Bioinform.* 15(1):6
57. Fariselli P, Olmea O, Valencia A, Casadio R. 2001. Prediction of contact maps with neural networks and correlated mutations. *Prot. Eng.* 14:835–43
58. Pollastri G, Baldi P. 2002. Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners. *Bioinformatics* 18:S62–70
59. Pollastri G, Vullo A, Frasconi P, Baldi P. 2006. Modular DAG-RNN architectures for assembling coarse protein structures. *J. Comput. Biol.* 13(3):631–50
60. Cheng J, Baldi P. 2007. Improved residue contact prediction using support vector machines and a large feature set. *BMC Bioinform.* 8(1):113
61. Wang S, Sun S, Li Z, Zhang R, Xu J. 2017. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLOS Comput. Biol.* 13(1):e1005324
62. Cheng J, Saigo H, Baldi P. 2006. Large-scale prediction of disulphide bridges using kernel methods two-dimensional recursive neural networks, and weighted graph matching. *Proteins* 62(3):617–29
63. Di Lena P, Nagata K, Baldi P. 2012. Deep architectures for protein contact map prediction. *Bioinformatics* 28:2449–57
64. Lyons J, Dehzangi A, Heffernan R, Sharma A, Paliwal K, et al. 2014. Predicting backbone Cα angles and dihedrals from protein sequences by stacked sparse auto-encoder deep neural network. *J. Comput. Chem.* 35(28):2040–46
65. Nagata K, Randall A, Baldi P. 2012. SIDEpro: a novel machine learning approach for the fast and accurate prediction of side-chain conformations. *Proteins* 80:142–53
66. Fariselli P, Pazos F, Valencia A, Casadio R. 2002. Prediction of protein–protein interaction sites in heterocomplexes with neural networks. *FEBS J.* 269(5):1356–61
67. Jones DT, Buchan DWA, Cozzetto D, Pontil M. 2011. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics* 28(2):184–90
68. Marks DS, Hopf TA, Sander C. 2012. Protein structure prediction from sequence variation. *Nat. Biotechnol.* 30(11):1072–80
69. Liu Y, Palmedo P, Ye Q, Berger B, Peng J. 2017. Enhancing evolutionary couplings with deep convolutional neural networks. *Cell Syst.* 6(1):65–74.e3
70. Cheng J, Randall A, Baldi P. 2006. Prediction of protein stability changes for single-site mutations using support vector machines. *Proteins* 62(4):1125–32
71. Magnan CN, Randall A, Baldi P. 2009. SOLpro: accurate sequence-based prediction of protein solubility. *Bioinformatics* 25(17):2200–7
72. Cheng J, Sweredoski M, Baldi P. 2005. Acurate prediction of protein disordered regions by mining protein structure data. *Data Min. Knowl. Discov.* 11(3):213–22
73. Volpato V, Alshomrani B, Pollastri G. 2015. Accurate ab initio and template-based prediction of short intrinsically-disordered regions by bidirectional recurrent neural networks trained on large-scale datasets. *Int. J. Mol. Sci.* 16(8):19868–85
74. Cheng J, Sweredoski MJ, Baldi P. 2006. DOMpro: protein domain prediction using profiles, secondary structure, relative solvent accessibility, and recursive neural networks. *Data Min. Knowl. Discov.* 13(1):1–10
75. Cheng J, Baldi P. 2006. A machine learning information retrieval approach to protein fold recognition. *Bioinformatics* 22(12):1456–63
76. Hou J, Adhikari B, Cheng J. 2017. DeepSF: deep convolutional neural network for mapping protein sequences to folds. arXiv:1706.01010 [cs.LG]
77. Mooney C, Pollastri G, Shields DC, Haslam NJ. 2012. Prediction of short linear protein binding regions. *J. Mol. Biol.* 415(1):193–204
78. Khan W, Duffy F, Pollastri G, Shields DC, Mooney C. 2013. Predicting binding within disordered protein regions to structurally characterised peptide-binding domains. *PLOS ONE* 8(9):e72838
79. Mooney C, Haslam NJ, Holton TA, Pollastri G, Shields DC. 2013. Peptidelocator: prediction of bioactive peptides in protein sequences. *Bioinformatics* 29(9):1120–26

80. Eickholt J, Cheng J. 2012. Predicting protein residue–residue contacts using deep networks and boosting. *Bioinformatics* 28(23):3066–72
81. Eickholt J, Cheng J. 2013. DNdisorder: predicting protein disorder using boosting and deep networks. *BMC Bioinform*. 14(1):88
82. Spencer M, Eickholt J, Cheng J. 2015. A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 12(1):103–12
83. Jo T, Hou J, Eickholt J, Cheng J. 2015. Improving protein fold recognition by deep learning networks. *Sci. Rep.* 5:17573
84. Cao R, Bhattacharya D, Hou J, Cheng J. 2016. DeepQA: improving the estimation of single protein model quality with deep belief networks. *BMC Bioinform*. 17(1):495
85. Nielsen H, Engelbrecht J, Brunak S, von Heijne G. 1997. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Prot. Eng.* 10(1):1–6
86. Blom N, Sicheritz-Pontén T, Gupta R, Gammeltoft S, Brunak S. 2004. Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence. *Proteomics* 4(6):1633–49
87. Jensen LJ, Gupta R, Blom N, Devos D, Tamames J, et al. 2002. Prediction of human protein function from post-translational modifications and localization features. *J. Mol. Biol.* 319(5):1257–65
88. Mooney C, Wang Y-H, Pollastri G. 2011. SCLpred: protein subcellular localization prediction by N-to-1 neural networks. *Bioinformatics* 27(20):2812–19
89. Mooney C, Haslam NJ, Pollastri G, Shields DC. 2012. Towards the improved discovery and design of functional peptides: Common features of diverse classes permit generalized prediction of bioactivity. *PLOS ONE* 7(10):e45012
90. Holton TA, Pollastri G, Shields DC, Mooney C. 2013. CPPpred: prediction of cell penetrating peptides. *Bioinformatics* 29(23):3094–96
91. Volpato V, Adelfio A, Pollastri G. 2013. Accurate prediction of protein enzymatic class by N-to-1 neural networks. *BMC Bioinform*. 14(1):S11
92. Asgari E, Mofrad MRK. 2015. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLOS ONE* 10(11):e0141287
93. Lund O, Nielsen M, Lundegaard C, Kesmir C, Brunak S. 2005. *Immunological Bioinformatics*. Cambridge, MA: MIT Press
94. Nielsen M, Lundegaard C, Worning P, Lauemøller SL, Lamberth K, et al. 2003. Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Prot. Sci.* 12(5):1007–17
95. Sweredoski MJ, Baldi P. 2008. PEPITO: improved discontinuous B-cell epitope prediction using multiple distance thresholds and half-sphere exposure. *Bioinformatics* 24(12):1459–60
96. Sweredoski MJ, Baldi P. 2009. COBEpro: a novel system for predicting continuous B-cell epitopes. *Prot. Eng. Des. Sel.* 22(3):113–20
97. Magnan CN, Zeller M, Kayala MA, Vigil A, Randall A, et al. 2010. High-throughput prediction of protein antigenicity using protein microarray data. *Bioinformatics* 26(23):2936–43
98. Vang YS, Xie X. 2017. HLA class I binding prediction via convolutional neural networks. *Bioinformatics* 33(17):2658–65
99. Stormo GD, Schneider TD, Gold L, Ehrenfeucht A. 1982. Use of the 'Perceptron' algorithm to distinguish translational initiation sites in *E. coli*. *Nucleic Acids Res*. 10(9):2997–3011
100. Brunak S, Engelbrecht J, Knudsen S. 1991. Prediction of human mRNA donor and acceptor sites from the DNA sequence. *J. Mol. Biol.* 220(1):49–65
101. Hebsgaard SM, Korning PG, Tolstrup N, Engelbrecht J, Rouzé P, Brunak S. 1996. Splice site prediction in *Arabidopsis thaliana* pre-mRNA by combining local and global sequence information. *Nucleic Acids Res*. 24(17):3439–52
102. Uberbacher EC, Mural RJ. 1991. Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *PNAS* 88(24):11261–65
103. Brunak S, Engelbrecht J, Knudsen S. 1990. Neural network detects errors in the assignment of mRNA splice sites. *Nucleic Acids Res*. 18(16):4797–801
104. Golden J, Garcia E, Tibbetts C. 1995. Evolutionary optimization of a neural network-based signal processor for photometric data from an automated DNA sequencer. *Proc. Annu. Conf. Evol. Progr., 4th, San Diego, Calif., 1–3 March*, ed. JR McDonnell, RG Reynolds, DB Fogel, pp. 579–601. Cambridge, MA: MIT Press

105. Ewing B, Hillier L, Wendl MC, Green P. 1998. Base-calling of automated sequencer traces using *phred*. I. Accuracy assessment. *Genome Res.* 8(3):175–85
106. Boža V, Brejová B, Vinař T. 2017. DeepNano: deep recurrent neural networks for base calling in MinION nanopore reads. *PLOS ONE* 12(6):e0178751
107. Alipanahi B, Delong A, Weirauch MT, Frey BJ. 2015. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* 33(8):831–38
108. Wang MD, Hassanzadeh HR. 2017. DeeperBind: enhancing prediction of sequence specificities of DNA binding proteins. bioRxiv 099754. **https://doi.org/10.1101/099754**
109. Qin Q, Feng J. 2017. Imputation for transcription factor binding predictions based on deep learning. *PLOS Comput. Biol.* 13(2):e1005403
110. Quang D, Xie X. 2017. Factornet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. bioRxiv 151274. **https://doi.org/10.1101/151274**
111. Zhou J, Troyanskaya OG. 2015. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods* 12(10):931–34
112. Quang D, Chen Y, Xie X. 2015. DANN: a deep learning approach for annotating the pathogenicity of genetic variants. *Bioinformatics* 31(5):761–63
113. Quang D, Xie X. 2016. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.* 44(11):e107
114. Kelley DR, Snoek J, Rinn JL. 2016. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.* 26(7):990–99
115. Angermueller C, Lee HJ, Reik W, Stegle O. 2017. DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome Biol.* 18(1):67
116. Li Y, Quang D, Xie X. 2017. Understanding sequence conservation with deep learning. bioRxiv 103929. **https://doi.org/10.1101/103929**
117. Liu F, Ren C, Li H, Zhou P, Bo X, Shu W. 2015. *De novo* identification of replication-timing domains in the human genome by deep learning. *Bioinformatics* 32(5):641–49
118. Zhang S, Zhou J, Hu H, Gong H, Chen L, et al. 2015. A deep learning framework for modeling structural features of RNA-binding protein targets. *Nucleic Acids Res.* 44(4):e32
119. Herrero J, Valencia A, Dopazo J. 2001. A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics* 17(2):126–36
120. Chicco D, Sadowski P, Baldi P. 2014. Deep autoencoder neural networks for gene ontology annotation predictions. *Proc. ACM Conf. Bioinform. Comput. Biol. Health Inform.*, *Newport Beach*, *Calif.*, *20–23 Sept.*, pp. 533–40. New York: Assoc. Comput. Mach.
121. Leung MKK, Xiong HY, Lee LJ, Frey BJ. 2014. Deep learning of the tissue-regulated splicing code. *Bioinformatics* 30(12):i121–29
122. Xiong HY, Alipanahi B, Lee LJ, Bretschneider H, Merico D, et al. 2015. The human splicing code reveals new insights into the genetic determinants of disease. *Science* 347(6218):1254806
123. Chen Y, Li Y, Narayan R, Subramanian A, Xie X. 2016. Gene expression inference with deep learning. *Bioinformatics* 32(12):1832–39
124. Agostinelli F, Ceglia N, Shahbaba B, Sassone-Corsi P, Baldi P. 2016. What time is it? Deep learning approaches for circadian rhythms. *Bioinformatics* 32(12):i8–17
125. Baldi P, Chauvin Y. 1993. Neural networks for fingerprint recognition. *Neural Comput.* 5(3):402–18
126. Cireşan DC, Meier U, Schmidhuber J. 2012. Multi-column deep neural networks for image classification. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, *25th*, *Providence*, *R.I.*, *16–21 June*, pp. 3642–49. New York: Inst. Electr. Electr. Eng.
127. Cireşan DC, Giusti A, Gambardella LM, Schmidhuber J. 2012. Deep neural networks segment neuronal membranes in electron microscopy images. *Proc. Int. Conf. Neural Inf. Process. Syst.*, *25th*, *Lake Tahoe*, *Calif.*, *3–6 Dec.*, ed. F Pereira, CJC Burges, L Bottou, KQ Weinberger, pp. 2843–51. Red Hook, NY: Curran Assoc.
128. Cireşan DC, Giusti A, Gambardella LM, Schmidhuber J. 2013. Mitosis detection in breast cancer histology images with deep neural networks. *Med. Image Comput. Comput.-Assist. Interv.*, *16th*, *Nagoya*, *Jap.*, *22–26 Sept.*, ed. K Mori, I Sakuma, Y Sato, C Barillot, N Navab, pp. 411–18. Berlin: Springer-Verlag

129. Roth HR, Yao J, Lu L, Stieger J, Burns JE, Summers RM. 2015. Detection of sclerotic spine metastases via random aggregation of deep convolutional neural network classifications. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, ed. J Yao, B Glocker, T Klinder, S Li, pp. 3–12. Cham, Switz.: Springer Int.
130. Shen W, Zhou M, Yang F, Yang C, Tian J. 2015. Multi-scale convolutional neural networks for lung nodule classification. *Int. Conf. Inf. Process. Med. Imaging*, ed. S Ourselin, D Alexander, CF Westin, M Cardoso, pp. 588–99. Cham, Switz.: Springer Int.
131. Wang D, Khosla A, Gargeya R, Irshad H, Beck AH. 2016. Deep learning for identifying metastatic breast cancer. arXiv:1606.05718 [q-bio.QM]
132. Wang J, Fang Z, Lang N, Yuan H, Su M-Y, Baldi P. 2017. A multi-resolution approach for spinal metastasis detection using deep siamese neural networks. *Comput. Biol. Med.* 84:137–46
133. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, et al. 2017. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542(7639):115–18
134. Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, et al. 2016. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA* 316(22):2402–10
135. Wang J, Ding H, Azamian F, Zhou B, Iribarren C, et al. 2017. Detecting cardiovascular disease from mammograms with deep learning. *IEEE Trans. Med. Imaging* 36(5):1172–81
136. Bucilua C, Caruana R, Niculescu-Mizil A. 2006. Model compression. *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 12th, Phila., Pa., 20–23 Aug.*, ed. L Ungar, M Craven, D Gunopulos, T Eliassi-Rad, pp. 535–41. New York: Assoc. Comput. Mach.
137. Sadowski P, Collado J, Whiteson D, Baldi P. 2015. Deep learning, dark knowledge, and dark matter. *Proc. Mach. Learn. Res.* 42:81–97
138. Redmon J, Divvala S, Girshick R, Farhadi A. 2016. You only look once: unified, real-time object detection. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 29th, Las Vegas, Nev., 26 June–1 July*, pp. 779–88. New York: Inst. Electr. Electr. Eng.
139. Masci J, Giusti A, Ciresan DC, Fricout G, Schmidhuber J. 2013. A fast learning algorithm for image segmentation with max-pooling convolutional networks. *IEEE Int. Conf. Image Process., 20th, Melb., Aust., 15–18 Sept.*, pp. 2713–17. New York: Inst. Electr. Electr. Eng.
140. Stollenga M, Beyon W, Liwicki M, Schmidhuber J. 2015. Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation. *Proc. Int. Conf. Neural Inf. Process. Syst., 28th, Montreal, Can., 7–12 Dec.*, ed. C Cortes, DD Lee, M Sugiyama, R Garnett, pp. 2998–3006. Cambridge, MA: MIT Press
141. Urban G, Tripathi P, Alkayali T, Mittal M, Jalali F, et al. 2017. Deep learning localizes and identifies polyps in real time with 96% accuracy in screening colonoscopy. *Gastroenterology*. In press
142. Obermeyer Z, Emanuel EJ. 2016. Predicting the future—big data, machine learning, and clinical medicine. *New Engl. J. Med.* 375(13):1216–19
143. Ching T, Himmelstein DS, Beaulieu-Jones BK, Kalinin AA, Do BT, et al. 2017. Opportunities and obstacles for deep learning in biology and medicine. bioRxiv 142760. **https://doi.org/10.1101/142760**
144. Cabitza F, Rasoini R, Gensini GF. 2017. Unintended consequences of machine learning in medicine. *JAMA* 318(6):517–18
145. Rajpurkar P, Hannun AY, Haghpanahi M, Bourn C, Ng AY. 2017. Cardiologist-level arrhythmia detection with convolutional neural networks. arXiv:1707.01836 [cs.CV]
146. Lee CK, Hofer I, Gabel E, Baldi P, Cannesson M. 2018. Development and validation of a deep neural network model for prediction of postoperative in-hospital mortality. *Anesthesiology*. In press
147. Lasko TA, Denny JC, Levy MA. 2013. Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. *PLOS ONE* 8(6):e66341
148. Miotto R, Li L, Kidd BA, Dudley JT. 2016. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Sci. Rep.* 6:26094
149. Beaulieu-Jones BK, Greene CS, et al. 2016. Semi-supervised learning of the electronic health record for phenotype stratification. *J. Biomed. Inform.* 64:168–78
150. Jensen AB, Moseley PL, Oprea TI, Ellesøe SG, Eriksson R, et al. 2014. Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients. *Nat. Commun.* 5:4022

151. Pham T, Tran T, Phung D, Venkatesh S. 2017. Predicting healthcare trajectories from medical records: a deep learning approach. *J. Biomed. Inform.* 69:218–229
152. Choi E, Bahadori MT, Schuetz A, Stewart WF, Sun J. 2016. Doctor AI: predicting clinical events via recurrent neural networks. *Proc. Mach. Learn. Res.* 56:301–18
153. Zipser D, Andersen RA. 1988. A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature* 331(6158):679–84
154. Yamins DLK, DiCarlo JJ. 2016. Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* 19(3):356–65
155. Baldi P, Sadowski P. 2016. A theory of local learning, the learning channel, and the optimality of backpropagation. *Neural Netw*. 83:61–74
156. Baldi P, Lu Z, Sadowski P. 2017. Learning in the machine: the symmetries of the deep learning channel. *Neural Netw*. 95:110–33
157. Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE* 10(7):e0130140
158. Ribeiro MT, Singh S, Guestrin C. 2016. "Why should I trust you?": Explaining the predictions of any classifier. *Proc. Conf. Knowl. Discov. Data Min., 22nd, San Franc., Calif., 13–17 Aug.* New York: Assoc. Comput. Mach.
159. Sherrington CS. 1951. *Man on His Nature*. Cambridge, UK: Cambridge Univ. Press. 2nd ed.