Chapter 1

Deep learning for computational biology

1.1 Introduction

Machine learning methods are general-purpose approaches to learn functional relationships from data without the need to define them a priori [29, 56, 58]. In computational biology, their appeal is the ability to derive predictive models without a need for strong assumptions about underlying mechanisms, which are frequently unknown or insufficiently defined. As a case in point, the most accurate prediction of gene expression levels is currently made from a broad set of epigenetic features using sparse linear models [13, 38] or random forests [49]; how the selected features determine the transcript levels remains an active research topic. Predictions in genomics [50, 59], proteomics [74], metabolomics [39], or sensitivity to compounds [17] all rely on machine learning approaches as a key ingredient.

Most of these applications can be described within the canonical machine learning workflow, which involves four steps: data cleaning and pre-processing, feature extraction, model fitting, and evaluation (Figure 1.1 (A)). It is customary to denote one data sample, including all covariates and features as input x (usually a vector of numbers), and label it with its response variable or output value y (usually a single number) when available.

A supervised machine learning model aims to learn a function $f(x) = y$ from a list of training pairs $(x_1, y_1), (x_2, y_2), \dots$ for which data are recorded (Figure 1.1 (B)). One typical application in biology is to predict the viability of a cancer cell line when exposed to a chosen drug [17, 55]. The input features x would capture somatic sequence variants of the cell line, chemical makeup of the drug, and its concentration, which together with the measured viability (output label y) can be used to train a support vector machine, a random forest classifier or a related method (functional relationship f). Given a new cell line (unlabelled data sample x^*) in the future, the learnt function predicts its survival (output label y^*) by calculating $f(x^*)$, even if f resembles more of a black box, and its inner workings of why

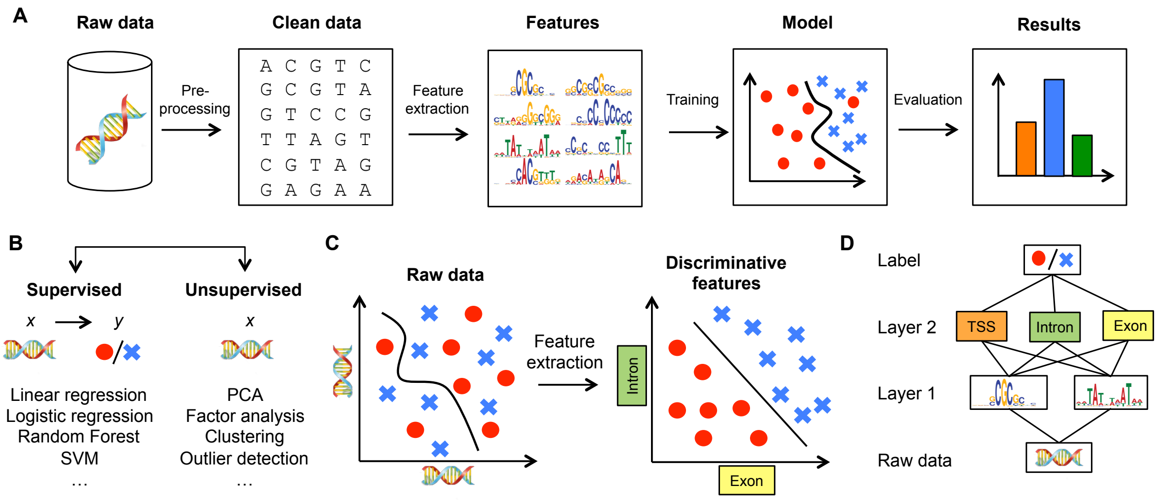

Figure 1.1 Machine learning and representation learning. (A) The classical machine learning workflow can be broken down into four steps: data pre-processing, feature extraction, model learning, and model evaluation. (B) Supervised machine learning methods relate input features x to an output label y, whereas unsupervised method learn factors about x without observed labels. (C) Raw input data are often high dimensional and related to the corresponding label in a complicated way, which is challenging for many classical machine learning algorithms (left plot). Alternatively, higher-level features extracted using a deep model may be able to better discriminate between classes (right plot). (D) Deep networks use a hierarchical structure to learn increasingly abstract feature representations from the raw data.

particular mutation combinations influence cell growth are not easily interpreted. Both regression (where y is a real number), and classification (where y is a categorical class label) can be viewed in this way. As a counterpart, unsupervised machine learning approaches aim to discover patterns from the data samples x itself, without the need for output labels y. Methods such as clustering, principal components analysis, and outlier detection are typical examples of unsupervised models applied to biological data.

The inputs x, calculated from the raw data, represent what the model 'sees about the world', and their choice is highly problem specific (Figure 1.1 (C)). Deriving most informative features is essential for performance, but the process can be labour-intensive and requires domain knowledge. This bottleneck is especially limiting for high dimensional data; even computational feature selection methods do not scale to assess the utility of vast number of possible input combinations. A major recent advance in machine learning is automating this critical step by learning a suitable representation of the data with deep artificial neural networks [11, 44, 68] (Figure 1.1 (D)). Briefly, a deep neural network takes the raw data at the lowest (input) layer, and transforms them into increasingly abstract feature representations by successively combining outputs from the preceding layer in a data-driven manner, encapsulating highly complicated functions in the process (Box 1). Deep learning is now one of the most active fields in machine learning and has been shown to improve performance in image- and speech recognition [16, 27, 31, 43, 82], natural language understanding [7, 51, 73, 80], and most recently, in computational biology [4, 14, 18, 40, 47, 76, 78, 83].

The potential of deep learning in high throughput biology is clear: in principle, it allows to better exploit the availability of increasingly large and high-dimensional datasets (e.g. from DNA sequencing, RNA measurements, flow cytometry, or automated microscopy) by training complex networks with multiple layers that capture their internal structure (Figure 1.1 (C)). The learned networks discover high-level features, improve performance over traditional models, increase interpretability and provide additional understanding about the structure of the biological data.

In this review, we discuss recent and forthcoming applications of deep learning, with a focus on applications in regulatory genomics and biological image analysis. The goal of this review is not to provide comprehensive background on all technical details, which can be found in more specialized literature [10, 11, 15, 25, 68]. Instead, we aim to provide practical pointers and the necessary background to get started with deep architectures, review current software solutions, and give recommendations for applying them to data. The applications we cover are deliberately broad to illustrate differences and communalities between approaches; reviews focusing on specific domains can be found elsewhere [22, 48, 53, 60]. Finally, we discuss both the potential and possible pitfalls of deep learning and contrast these methods to traditional machine learning and classical statistical analysis approaches.

1.1.1 Artificial neural networks

An artificial neural network, initially inspired by neural networks in the brain [21, 54, 65] consists of layers of interconnected compute units (neurons). In the canonical configuration, the network receives data in an input layer, which are then transformed in a nonlinear way through multiple hidden layers, before final outputs are computed in the output layer (Figure 1.2 (A)). Neurons in a hidden or output layer are connected to all neurons of the previous layer. Each neuron computes a weighted sum of its inputs, and applies a nonlinear activation function to calculate its output (Figure 1.2 (B)). The most popular activation function is the Rectified linear unit (ReLU, (Figure 1.2 (B)), since it allows faster learning compared to alternatives (e.g. sigmoid or tanh unit) [24]. The depth of a neural network corresponds to the number of hidden layers, and the width to the maximum number of neurons in one of its layers. As it became possible to train networks with larger numbers of hidden layers, artificial neural networks were rebranded to “deep networks”.

The weights between neurons are free parameters that capture the model’s representation of the data, and are learned from input/output samples. Learning minimizes a loss function that measures the fit of the model output to the true label of a sample (Figure 1.2 (A), bottom). This minimization is challenging, since the loss function is high dimensional and non-convex, similar to a landscape with many hills and valleys (Figure 1.2 (C)). It took several decades before the backward propagation algorithm was first applied to compute a loss function gradient via chain rule for derivatives [66], ultimately enabling efficient training of neural networks using stochastic gradient descent. During learning, the predicted label is compared with the true label to compute a loss for the current set of model weights. The loss is then backward propagated through the network to compute the gradients of the loss function and update (Figure 1.2 (A)). While learning in deep neural networks remains an active area of research, existing software packages (Table 1) can already be applied without knowledge of the mathematical details involved.

Alternative architectures to such fully connected feedforward networks have been developed for specific applications, which differ in the way neurons are arranged. These include convolutional neural networks, which are widely used for modelling images (Box 2), recurrent neural networks for sequential data [51, 72], or restricted Boltzmann machines [32, 67] and autoencoders [2, 33, 41] for unsupervised learning. The choice of network architecture and other parameters can be made in a data driven and objective way by assessing the model performance on a validation dataset.

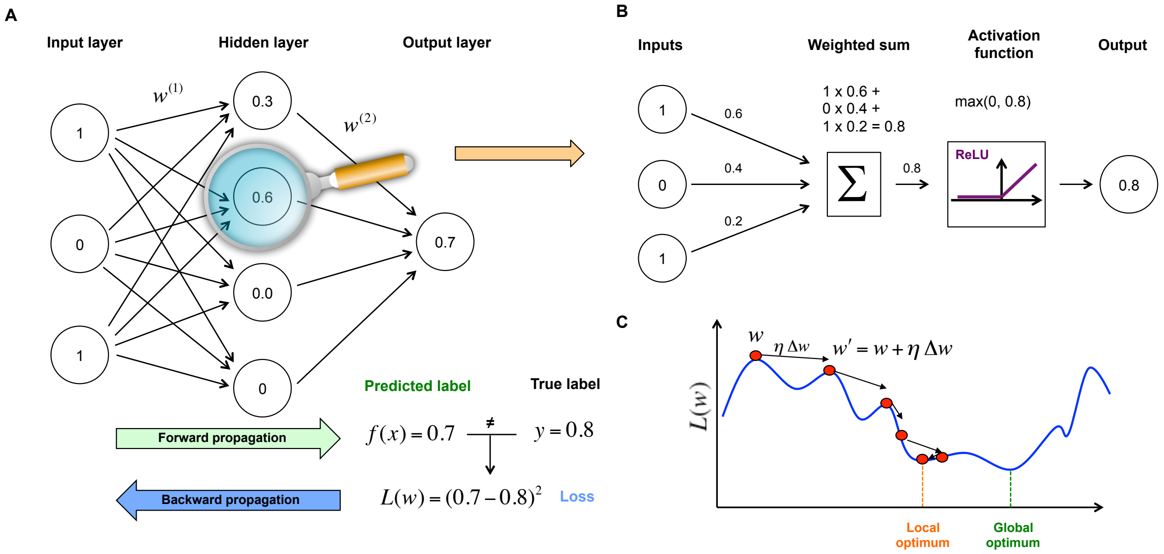

Figure 1.2 Building blocks and learning principles of a neural network. (A) Fully connected feedforward neural network with one input layer, hidden layer, and output layer. Each layer i consists of neurons which are connected to all neurons of the previous layer with weights $w(i)$. Given input x, neuron activations are calculated and forward propagated to the output layer to obtain a prediction $f(x)$. (B) Zoom-in view into one neuron, which computes the weighted sum of its inputs and applies a rectification function that thresholds negative signals to 0, and passes through positive signal. (C) Gradient-based optimization of the loss function $L(w)$. In each step, the current weight vector (red dot) is moved along the direction of steepest descent Δw (direction arrow) by learning rate η (length of vector). Decaying the learning rate over time allows to explore different domains of the loss function by jumping over valleys at the beginning of the training (left side), and fine-tune parameters with smaller learning rates in later stages of the model training.

1.2 Deep learning for regulatory genomics

Conventional approaches for regulatory genomics relate sequence variation to changes in molecular traits. One approach is to leverage variation between genetically diverse individuals to map quantitative trait loci (QTL). This principle has been applied to identify regulatory variants that affect gene expression levels [57, 63], DNA methylation [9, 23], histone marks [28, 79], and proteome variation [3, 8, 61, 77] (Figure 1.3 (A)). Better statistical methods have helped to increase the power to detect regulatory QTLs [37, 62, 64, 70], however any mapping approach is intrinsically limited to variation that is present in the training population. Thus, studying effects of rare mutations in particular requires extremely large datasets.

An alternative is to train models that use variation between regions within a genome (Fig 3A). Splitting the sequence into windows centred on the trait of interest gives rise to tens of thousands of training examples for most molecular traits even when using a single individual. Even with large datasets, predicting molecular traits from DNA sequence is challenging due to multiple layers of abstraction between effect of individual DNA variants and the trait of interest, as well as the dependence of the molecular traits on a broad sequence context and interactions with distal regulatory elements.

The value of deep neural networks in this context is twofold. First, classical machine learning methods cannot operate on the sequence directly, and thus require predefining features that can be extracted from sequence based on prior knowledge (e.g. the presence of absence of single-nucleotide variants (SNVs), k-mer frequencies, motif occurrences, conservation, known regulatory variants, or structural elements). Deep neural networks can help circumventing the manual extraction of features by learning them from data. Second, because of their representational richness, they can capture nonlinear dependencies in the sequence, interaction effects, and span wider sequence context at multiple genomic scales. Attesting to their utility, deep neural networks have been successfully applied to predict splicing activity [47, 81], specificities of DNA- and RNA binding proteins [4], or epigenetic marks and to study the effect of DNA sequence alterations [40, 83].

1.2.1 Early applications of neural networks in regulatory genomics

The first successful applications of neural networks in regulatory genomics replaced a classical machine learning approach with a deep model, without changing the input features. For example, Xiong et al. considered a fully connected feedforward neural network to predict the splicing activity of individual exons. The model was trained using more than $1,000$ pre-defined features extracted from the candidate exon and adjacent introns. Despite the relatively low number of 10,700 training samples in combination with the model complexity, this method achieved substantially higher prediction accuracy of splicing activity compared to simpler approaches, and in particular was able to identify rare mutations implicated in splicing misregulation.

1.2.2 Convolutional designs

More recent work using convolutional neural networks (CNNs) allowed direct training on the DNA sequence, without the need to define features [4, 5, 40, 83]. The CNN architecture allows to greatly reduce the number of model parameters compared to a fully connected network by applying convolutional operations to only small regions of the input space and by sharing parameters between regions. The key advantage resulting from this approach is the ability to directly train the model on larger sequence windows (Box 2, Figure 1.3 (B), Figure 1.4).

Alipanahi et al. considered convolutional network architectures to predict specificities of DNA- and RNA binding proteins [4]. Their DeepBind model outperformed existing methods, was able to recover known and novel sequence motifs, and could quantify the effect of sequence alterations and identify functional SNVs. A key innovation that enabled training the model directly on the raw DNA sequence was the application of a one-dimensional convolutional layer. Intuitively, the neurons in the convolutional layer scan for motif sequences and combinations thereof, similar to conventional position-weight matrices [71]. The learning signal from deeper layers informs the convolutional layer which motifs are most relevant. The motifs recovered by the model can then be visualized as heatmaps or sequence logos (Figure 1.3 (D)).

1.2.3 In silico prediction of mutation effects

An important application of deep neural networks trained on the raw DNA sequence is to predict the effect of mutations in silico. Such model-based assessments of the effect of sequence changes complement methods based on QTL mapping, and can in particular help to uncover regulatory effects of rare SNVs or to fine-map likely causal genes. An intuitive approach for visualizing such predicted regulatory effects are mutation maps [4], whereby the effect of all possible mutations for a given input sequence is represented in a matrix view (Figure 1.3 (E)). The authors could further reliably identify deleterious SNVs, by training an additional neural network with predicted binding scores for a wild type and mutant sequence (Figure 1.3 (C)).

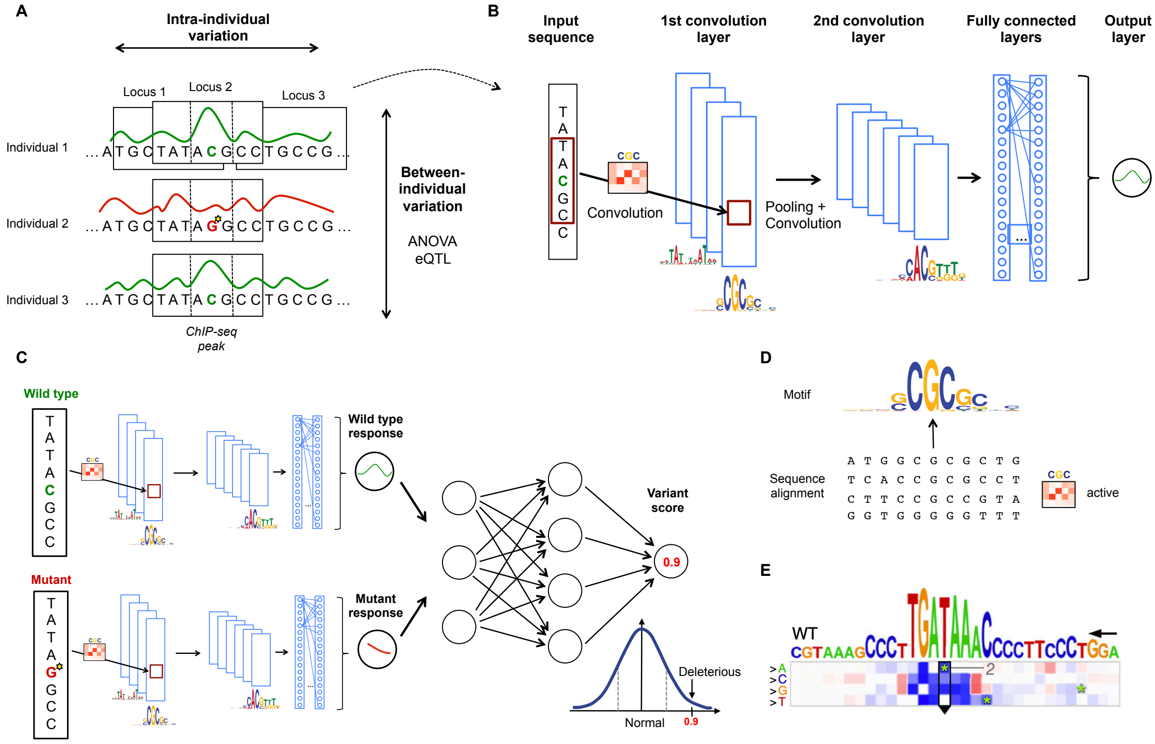

Figure 1.3 Principles of using neural networks for predicting molecular traits from DNA sequence. (A) DNA sequence and the molecular response variable along the genome for three individuals. Conventional approaches in regulatory genomics consider variations between individuals, whereas deep learning allows exploiting intra-individual variations by tiling the genome into sequence DNA windows centred on individual traits, resulting in large training datasets from a single sample. (B) One-dimensional convolutional neural network for predicting a molecular trait from the raw DNA sequence in a window. Filters of the first convolutional layer (example shown on the edge) scan for motifs in the input sequence. Subsequent pooling reduces the input dimension, and additional convolutional layers and can model interactions between motifs in the previous layer. (C) Response variable predicted by the neural network shown in (B) for a wild type and mutant sequence is used as input to an additional neural network that predicts a variant score and allows to discriminate normal from deleterious variants. (D) Visualization of a convolutional filter by aligning genetic sequences that maximally activate the filter and creating a sequence motif. (E) Mutation map of a sequence window. Rows correspond to the four possible base pair substitutions, columns to sequence positions. The predicted impact of any sequence change is colour coded. Letters on top denote the wild type sequence with the height of each nucleotide denoting the maximum effect across mutations (Figure panel adapted from Alipanahi et al.).

1.2.4 Joint prediction of multiple traits and further extensions

Following their initial successes, convolutional architectures have been extended and applied to a range of tasks in regulatory genomics. For example, Zhou and Troyanskaya considered these architectures to predict chromatin marks from DNA sequence. The authors observed that the size of the input sequence window is a major determinant of model performance, where larger windows (now up to 1kb) coupled with multiple convolutional layers enabled capturing sequence features at different genomic length scales. A second innovation was to use neural network architectures with multiple output variables (so called multi-task neural networks), here to predict multiple chromatin states in parallel. Multi-task architectures allow learning shared features between outputs, thereby improving generalization performance, and markedly reducing the computational cost of model training compared to learning independent models for each trait [14].

In a similar vein, Kelley et al. developed the open-source deep learning framework Basset, to predict DNase-I hypersensitivity across multiple cell types and to quantify the effect of SNVs on chromatin accessibility. Again, the model improved prediction performance compared to conventional methods and was able to retrieve both known and novel sequence motifs that are associated with DNase-I hypersensitivity. A related architecture has also been considered by Angermueller et al. to predict DNA methylation states in single-cell bisulfite sequencing studies [5]. This approach combined convolutional architectures to detect informative DNA sequence motifs with additional features derived form neighbouring CpG sites, thereby accounting for methylation context. Most recently, Koh et al. applied CNNs to de-noise genome-wide chromatin immunoprecipitation followed by sequencing data in order to obtain a more accurate prevalence estimate for different chromatin marks [42].

At present, CNNs are among the most widely used architectures to extract features from fixed sized DNA sequence windows. However, alternative architectures could also be considered. For example, recurrent neural networks (RNNs) are suited to model sequential data [51], and have been applied for modelling natural language and speech [12, 16, 26, 31, 73, 80], protein sequences [1, 76], clinical medical data [12], and to a limited extent DNA sequences [46]. RNNs are appealing for applications in regulatory genomics, because they allow modelling sequences of variable length, and to capture long-range interactions within the sequence and across multiple outputs. However, at present, RNNs are more difficult to train than CNNs, and additional work is needed to better understand the settings where one should be preferred over the other.

Complementary to supervised methods, unsupervised deep learning architectures learn low-dimensional feature representations from high-dimensional unlabelled data, similarly

Draft - v1.1 Monday 20th February, 2017 – 22:11

to classical principal components analysis or factor analysis, but using a non-linear model. Examples of such approaches are stacked autoencoders [77], restricted Boltzmann machines, and deep belief networks [33]. The learnt features can be used to visualize data or as input for classical supervised learning tasks. For example, sparse autoencoders have been applied to classify cancer cases using gene-expression profiles [20], or to predict protein backbones [52]. Restricted Boltzmann machines can also be used for unsupervised pre-training of deep networks to subsequently train supervised models of protein secondary structures [69], disordered protein regions [19, 18], or amino-acid contacts [19]. Skip-gram neural networks have been applied to learn low-dimensional representations of protein sequences and improve protein classification [6]. In general, unsupervised models are a powerful approach if large quantities of unlabelled data are available to pre-train complex models. Once trained, these models can help to improve performance on classification tasks, for which smaller numbers of labelled examples are typically available.

1.2.5 Convolutional neural network

Convolutional neural networks (CNNs) were originally inspired by cognitive neuroscience and Hubel and Wiesel's seminal work on the cat's visual cortex, which was found to have simple neurons that respond to small motifs in the visual field, and complex neurons that respond to larger ones [35, 34].

CNNs are designed to model input data in the form of multi-dimensional arrays, such as two-dimensional images with three colour channels [30, 36, 43, 45, 75, 82], or one-dimensional genomic sequences with one channel per nucleotide [4, 5, 40, 83]. The high dimensionality of these data (up to millions of pixels for high-resolution images) render training a fully connected neural network challenging, as the number of parameters of such a model would typically exceed the number of training data to fit them. To circumvent this, CNNs make additional assumptions on the structure of the network, thereby reducing the effective number of parameters to learn.

A convolutional layer consists of multiple maps of neurons, so called feature maps or filters, with their size being equal to the dimension of the input image (Figure 1.4). Two concepts allow reducing the number of model parameters: local connectivity and parameter sharing. First, unlike in a fully connected network, each neuron within a feature map is only connected to a local patch of neurons in the previous layer, the so-called receptive field. Second, all neurons within a given feature map share the same parameters. Hence, all neurons within a feature map scan for the same feature in the previous layer, however at different locations. Different feature maps might, for example, detect edges of different orientation in an image, or sequence motifs in a genomic sequence. The activity of a neuron is obtained by computing a discrete convolution of its receptive field, i.e. computing the weighted sum of input neurons, and applying an activation function.

In most applications, the exact position and frequency of features is irrelevant for the final prediction, such as recognizing objects in an image. Using this assumption, the pooling layer summarizes adjacent neurons by computing, for example, the maximum or average over their activity, resulting in a smoother representation of feature activities. By applying the same pooling operation to small image patches that are shifted by more than one pixel, the input image is effectively down-sampled, thereby further reducing the number of model parameters.

A CNN typically consists of multiple convolutional and pooling layers, which allows learning more and more abstract features at increasing scales from small edges, to object parts, and finally entire objects. One or more fully connected layers can follow the last pooling layer. Model hyper-parameters such as the number of convolutional layers, number of feature maps, or the size of receptive fields are application dependent and should be strictly selected on a validation data set (see below).

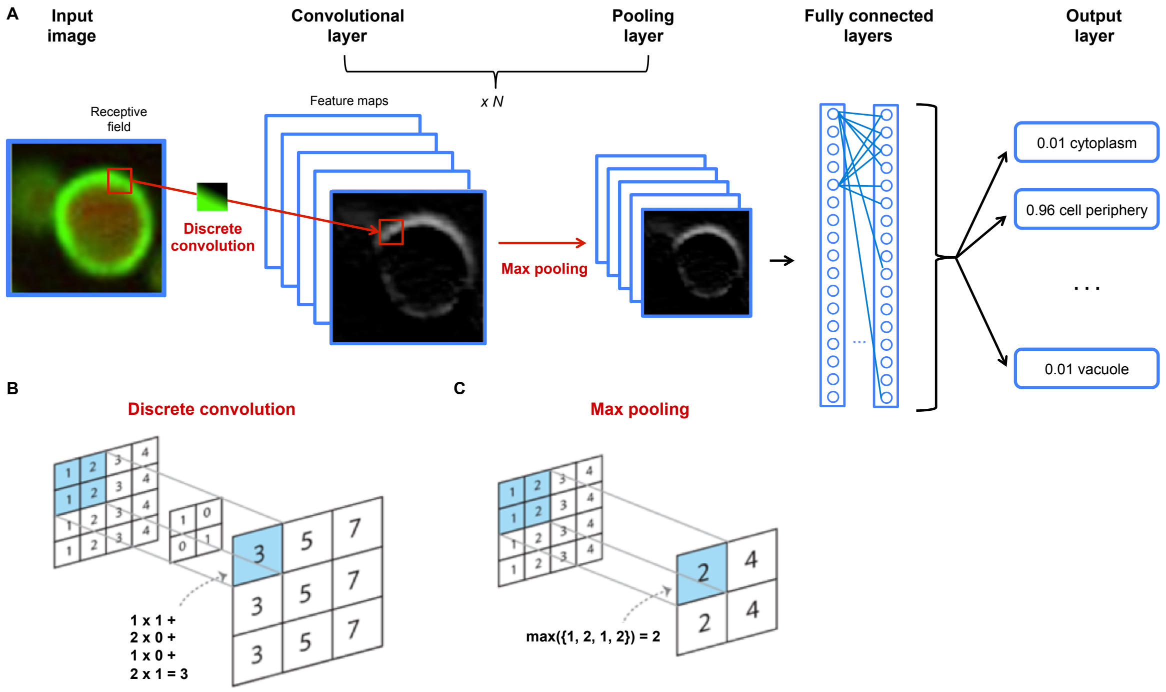

Figure 1.4 Convolutional neural networks (CNN). (A) A typical CNN consists of a number of convolutional and pooling layers, two fully connected layers, and one output layer. Each convolutional layer consists of multiple feature maps, with neurons responding to a particular feature in a receptive field (red square). One feature map responding to the membrane of a cell at a particular angle is highlighted on the edge. (B) Neuron activities result from a discrete convolution of their receptive field. (C) Max pooling computes the maximum neuron activity over a small patch, reducing the dimension of a convolutional layer.

References

[1] Michalis Agathocleous, Georgia Christodoulou, Vasilis Promponas, Chris Christodoulou, Vassilis Vassiliades, and Antonis Antoniou. Protein secondary structure prediction with bidirectional recurrent neural nets: Can weight updating for each residue enhance performance? In *Artificial Intelligence Applications and Innovations*, pages 128–137. Springer. ISBN 3-642-16238-X.

[2] Guillaume Alain, Yoshua Bengio, and Salah Rifai. Regularized auto-encoders estimate local statistics. pages 1–17.

[3] F. W. Albert, S. Treusch, A. H. Shockley, J. S. Bloom, and L. Kruglyak. Genetics of single-cell protein abundance variation in large yeast populations. 506(7489):494–7. ISSN 1476-4687 (Electronic) 0028-0836 (Linking). doi: 10.1038/nature12904.

[4] Babak Alipanahi, Andrew Delong, Matthew T. Weirauch, and Brendan J. Frey. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. 33: 831–838. ISSN 1087-0156. doi: 10.1038/nbt.3300.

[5] Christof Angermueller, Heather Lee, Wolf Reik, and Oliver Stegle. Accurate prediction of single-cell DNA methylation states using deep learning. page 055715. doi: 10.1101/055715. URL http://biorxiv.org/content/early/2017/02/01/055715.

[6] Ehsaneddin Asgari and Mohammad R. K. Mofrad. ProtVec: A Continuous Distributed Representation of Biological Sequences. 10:e0141287. ISSN 1932-6203. doi: 10.1371/journal.pone.0141287.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate.

[8] A. Battle, Z. Khan, S. H. Wang, A. Mitrano, M. J. Ford, J. K. Pritchard, and Y. Gilad. Genomic variation. Impact of regulatory variation from RNA to protein. 347(6222):664–7. ISSN 1095-9203 (Electronic) 0036-8075 (Linking). doi: 10.1126/science.1260793.

[9] Jordana T Bell, Athma A Pai, Joseph K Pickrell, Daniel J Gaffney, Roger Pique-Regi, Jacob F Degner, Yoav Gilad, and Jonathan K Pritchard. DNA methylation patterns associate with genetic and gene expression variation in HapMap cell lines. 12(1):R10.

[10] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer. ISBN 3-642-35288-X.

[11] Yoshua Bengio, Aaron Courville, and Pierre Vincent. Representation learning: A review and new perspectives. 35(8):1798–1828. ISSN 0162-8828.

[12] Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. Distilling Knowledge from Deep Networks with Applications to Healthcare Domain.

[13] C. Cheng, K. K. Yan, K. Y. Yip, J. Rozowsky, R. Alexander, C. Shou, and M. Gerstein. A statistical framework for modeling gene expression using chromatin features and application to modENCODE datasets. 12(2):R15. ISSN 1474-760X (Electronic) 1474-7596 (Linking). doi: 10.1186/gb-2011-12-2-r15.

[14] George E. Dahl, Navdeep Jaitly, and Ruslan Salakhutdinov. Multi-task Neural Networks for QSAR Predictions.

[15] Li Deng. Deep Learning: Methods and Applications. 7:197–387. ISSN 1932-8346, 1932-8354. doi: 10.1561/2000000039.

[16] Li Deng and Roberto Togneri. Deep dynamic models for learning hidden representations of speech features. In *Speech and Audio Processing for Coding, Enhancement and Recognition*, pages 153–195. Springer.

[17] F. Eduati, L. M. Mangravite, T. Wang, H. Tang, J. C. Bare, R. Huang, T. Norman, M. Kellen, M. P. Menden, J. Yang, X. Zhan, R. Zhong, G. Xiao, M. Xia, N. Abdo, O. Kosyk, Niehs-Ncats-Unc Dream Toxicogenetics Collaboration, S. Friend, A. Dearry, A. Simeonov, R. R. Tice, I. Rusyn, F. A. Wright, G. Stolovitzky, Y. Xie, and J. Saez-Rodriguez. Prediction of human population responses to toxic compounds by a collaborative competition. 33(9):933–40. ISSN 1546-1696 (Electronic) 1087-0156 (Linking). doi: 10.1038/nbt.3299.

[18] Jesse Eickholt and Jianlin Cheng. DNdisorder: Predicting protein disorder using boosting and deep networks. 14:88, . ISSN 1471-2105. doi: 10.1186/1471-2105-14-88.

[19] Jesse Eickholt and Jianlin Cheng. Predicting protein residue-residue contacts using deep networks and boosting. 28:3066–3072, . ISSN 1367-4803. doi: 10.1093/bioinformatics/bts598.

[20] Rasool Fakoor, Faisal Ladhak, Azade Nazi, and Manfred Huber. Using deep learning to enhance cancer diagnosis and classification.

[21] BWAC Farley and W Clark. Simulation of self-organizing systems by digital computer. 4(4):76–84. ISSN 2168-2690.

[22] Erik Gawehn, Jan A. Hiss, and Gisbert Schneider. Deep Learning in Drug Discovery. 35(1):3–14. ISSN 1868-1751. doi: 10.1002/minf.201501008.

[23] J Raphael Gibbs, Marcel P van der Brug, Dena G Hernandez, Bryan J Traynor, Michael A Nalls, Shiao-Lin Lai, Sampath Arepalli, Allissa Dillman, Ian P Rafferty, and Juan Troncoso. Abundant quantitative trait loci exist for DNA methylation and gene expression in human brain. 6(5):e1000952. ISSN 1553-7404.

[24] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. pages 315–323.

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. URL http://www.deeplearningbook.org.

[26] A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6645–6649. doi: 10.1109/ICASSP.2013.6638947.

[27] Alex Graves. Generating Sequences With Recurrent Neural Networks.

[28] F. Grubert, J. B. Zaugg, M. Kasowski, O. Ursu, D. V. Spacek, A. R. Martin, P. Greenside, R. Srivas, D. H. Phanstiel, A. Pekowska, N. Heidari, G. Euskirchen, W. Huber, J. K. Pritchard, C. D. Bustamante, L. M. Steinmetz, A. Kundaje, and M. Snyder. Genetic Control of Chromatin States in Humans Involves Local and Distal Chromosomal Interactions. 162(5):1051–65. ISSN 1097-4172 (Electronic) 0092-8674 (Linking). doi: 10.1016/j.cell.2015.07.048.

[29] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: Data mining, inference and prediction. 27(2):83–85. ISSN 0343-6993.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition.

[31] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, and Tara N Sainath. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. 29(6):82–97. ISSN 1053-5888.

[32] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade*, pages 599–619. Springer. ISBN 3-642-35288-X.

[33] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. 313(5786):504–507. ISSN 0036-8075.

[34] David H Hubel and Torsten N Wiesel. The period of susceptibility to the physiological effects of unilateral eye closure in kittens. 206(2):419, .

[35] DH Hubel and TN Wiesel. Shape and arrangement of columns in cat's striate cortex. 165(3):559, .

[36] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153. doi: 10.1109/ICCV.2009.5459469.

[37] H. M. Kang, C. Ye, and E. Eskin. Accurate discovery of expression quantitative trait loci under confounding from spurious and genuine regulatory hotspots. 180(4):1909–25. ISSN 0016-6731 (Print) 0016-6731 (Linking). doi: 10.1534/genetics.108.094201.

[38] R. Karlic, H. R. Chung, J. Lasserre, K. Vlahovicek, and M. Vingron. Histone modification levels are predictive for gene expression. 107(7):2926–31. ISSN 1091-6490 (Electronic) 0027-8424 (Linking). doi: 10.1073/pnas.0909344107.

Draft - v1.1 Monday 20th February, 2017 – 22:11

[39] D. B. Kell. Metabolomics, machine learning and modelling: Towards an understanding of the language of cells. 33:520–4. ISSN 0300-5127 (Print) 0300-5127 (Linking). doi: 10.1042/BST0330520.

[40] D. R. Kelley, J. Snoek, and J. Rinn. Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks. Advance online. ISSN 1549-5469 (Electronic) 1088-9051 (Linking). doi: 10.1101/gr.200535.115.

[41] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes.

[42] Pang Wei Koh, Emma Pierson, and Anshul Kundaje. Denoising genome-wide histone ChIP-seq with convolutional neural networks. page 052118. doi: 10.1101/052118. URL http://biorxiv.org/content/early/2017/01/27/052118.

[43] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. pages 1097–1105.

[44] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. 521:436–444, . ISSN 0028-0836. doi: 10.1038/nature14539.

[45] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. 1(4):541–551, .

[46] Byunghan Lee, Taehoon Lee, Byunggook Na, and Sungroh Yoon. DNA-Level Splice Junction Prediction using Deep Recurrent Neural Networks.

[47] Michael K. K. Leung, Hui Yuan Xiong, Leo J. Lee, and Brendan J. Frey. Deep learning of the tissue-regulated splicing code. 30:i121–i129, . ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btu277.

[48] Michael KK Leung, Andrew Delong, Babak Alipanahi, and Brendan J Frey. Machine Learning in Genomic Medicine: A Review of Computational Problems and Data Sets. . ISSN 0018-9219.

[49] J. Li, T. Ching, S. Huang, and L. X. Garmire. Using epigenomics data to predict gene expression in lung cancer. 16 Suppl 5:S10. ISSN 1471-2105 (Electronic) 1471-2105 (Linking). doi: 10.1186/1471-2105-16-S5-S10.

[50] M. W. Libbrecht and W. S. Noble. Machine learning applications in genetics and genomics. 16(6):321–32. ISSN 1471-0064 (Electronic) 1471-0056 (Linking). doi: 10.1038/nrg3920.

[51] Zachary C. Lipton. A Critical Review of Recurrent Neural Networks for Sequence Learning.

[52] James Lyons, Abdollah Dehzangi, Rhys Heffernan, Alok Sharma, Kuldip Paliwal, Abdul Sattar, Yaoqi Zhou, and Yuedong Yang. Predicting backbone C angles and dihedrals from protein sequences by stacked sparse autoencoder deep neural network. 35(28):2040–2046. ISSN 1096-987X.

[53] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov. Applications of Deep Learning in Biomedicine. 13(5):1445–54. ISSN 1543-8392 (Electronic) 1543-8384 (Linking). doi: 10.1021/acs.molpharmaceut.5b00982.

[54] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. 5(4):115–133. ISSN 0007-4985.

[55] M. P. Menden, F. Iorio, M. Garnett, U. McDermott, C. H. Benes, P. J. Ballester, and J. Saez-Rodriguez. Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties. 8(4):e61318. ISSN 1932-6203 (Electronic) 1932-6203 (Linking). doi: 10.1371/journal.pone.0061318.

[56] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. *Machine Learning: An Artificial Intelligence Approach*. Springer Science & Business Media. ISBN 3-662-12405-X.

[57] S. B. Montgomery, M. Sammeth, M. Gutierrez-Arcelus, R. P. Lach, C. Ingle, J. Nisbett, R. Guigo, and E. T. Dermitzakis. Transcriptome genetics using second generation sequencing in a Caucasian population. 464(7289):773–7. ISSN 1476-4687 (Electronic) 0028-0836 (Linking). doi: 10.1038/nature08903.

[58] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press. ISBN 0-262-01802-0.

[59] K. Märtens, J. Hallin, J. Warringer, G. Liti, and L. Parts. Predicting quantitative traits from genome and phenome with near perfect accuracy. 7:11512. ISSN 2041-1723 (Electronic) 2041-1723 (Linking). doi: 10.1038/ncomms11512.

[60] Yongjin Park and Manolis Kellis. Deep learning for regulatory genomics. 33:825–826. ISSN 1087-0156. doi: 10.1038/nbt.3313.

[61] L. Parts, Y. C. Liu, M. M. Tekkedil, L. M. Steinmetz, A. A. Caudy, A. G. Fraser, C. Boone, B. J. Andrews, and A. P. Rosebrock. Heritability and genetic basis of protein level variation in an outbred population. 24(8):1363–70, . ISSN 1549-5469 (Electronic) 1088-9051 (Linking). doi: 10.1101/gr.170506.113.

[62] L. Parts, O. Stegle, J. Winn, and R. Durbin. Joint genetic analysis of gene expression data with inferred cellular phenotypes. 7(1):e1001276, . ISSN 1553-7404 (Electronic) 1553-7390 (Linking). doi: 10.1371/journal.pgen.1001276.

[63] J. K. Pickrell, J. C. Marioni, A. A. Pai, J. F. Degner, B. E. Engelhardt, E. Nkadori, J. B. Veyrieras, M. Stephens, Y. Gilad, and J. K. Pritchard. Understanding mechanisms underlying human gene expression variation with RNA sequencing. 464(7289):768–72. ISSN 1476-4687 (Electronic) 0028-0836 (Linking). doi: 10.1038/nature08872.

[64] B. Rakitsch and O. Stegle. Modelling local gene networks increases power to detect trans-acting genetic effects on gene expression. 17(1):33. ISSN 1474-760X (Electronic) 1474-7596 (Linking). doi: 10.1186/s13059-016-0895-2.

[65] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. 65(6):386. ISSN 1939-1471.

Draft - v1.1 Monday 20th February, 2017 – 22:11

[66] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. 5(3):1.

[67] Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep Boltzmann machines. pages 693–700.

[68] Juergen Schmidhuber. Deep Learning in Neural Networks: An Overview. 61:85–117. ISSN 08936080. doi: 10.1016/j.neunet.2014.09.003.

[69] Matt Spencer, Jesse Eickholt, and Jianlin Cheng. A deep learning network approach to ab initio protein secondary structure prediction. 12(1):103–112. ISSN 1545-5963.

[70] O. Stegle, L. Parts, R. Durbin, and J. Winn. A Bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eQTL studies. 6(5):e1000770. ISSN 1553-7358 (Electronic) 1553-734X (Linking). doi: 10.1371/journal.pcbi.1000770.

[71] Gary D Stormo, Thomas D Schneider, Larry Gold, and Andrzej Ehrenfeucht. Use of the ‘Perceptron’algorithm to distinguish translational initiation sites in E. coli. 10(9): 2997–3011. ISSN 0305-1048.

[72] Ilya Sutskever. Training recurrent neural networks.

[73] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. pages 3104–3112.

[74] A. L. Swan, A. Mobasheri, D. Allaway, S. Liddell, and J. Bacardit. Application of machine learning to proteomics data: Classification and biomarker identification in postgenomics biology. 17(12):595–610. ISSN 1557-8100 (Electronic) 1536-2310 (Linking). doi: 10.1089/omi.2013.0017.

[75] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision.

[76] Søren Kaae Sønderby and Ole Winther. Protein Secondary Structure Prediction with Long Short Term Memory Networks.

[77] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. 11:3371–3408. ISSN 1532-4435.

[78] Kun Wang, Kan Cao, and Sridhar Hannenhalli. Chromatin and genomic determinants of alternative splicing. pages 345–354. ACM. ISBN 1-4503-3853-4.

[79] S. M. Waszak, O. Delaneau, A. R. Gschwind, H. Kilpinen, S. K. Raghav, R. M. Witwicki, A. Orioli, M. Wiederkehr, N. I. Panousis, A. Yurovsky, L. Romano-Palumbo, A. Planchon, D. Bielser, I. Padioleau, G. Udin, S. Thurnheer, D. Hacker, N. Hernandez, A. Reymond, B. Deplancke, and E. T. Dermitzakis. Population Variation and Genetic Control of Modular Chromatin Architecture in Humans. 162(5):1039–50. ISSN 1097-4172 (Electronic) 0092-8674 (Linking). doi: 10.1016/j.cell.2015.08.001.

[80] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic Memory Networks for Visual and Textual Question Answering. .

[81] Hui Y. Xiong, Babak Alipanahi, Leo J. Lee, Hannes Bretschneider, Daniele Merico, Ryan K. C. Yuen, Yimin Hua, Serge Gueroussov, Hamed S. Najafabadi, Timothy R. Hughes, Quaid Morris, Yoseph Barash, Adrian R. Krainer, Nebojsa Jojic, Stephen W. Scherer, Benjamin J. Blencowe, and Brendan J. Frey. The human splicing code reveals new insights into the genetic determinants of disease. 347:1254806, . ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1254806.

[82] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer. ISBN 3-319-10589-2.

[83] J. Zhou and O. G. Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. 12(10):931–4. ISSN 1548-7105 (Electronic) 1548-7091 (Linking). doi: 10.1038/nmeth.3547.