



# Recent Advances of Deep Learning in Bioinformatics and Computational Biology

Binhua Tang<sup>1,2\*</sup>, Zixiang Pan<sup>1†</sup>, Kang Yin<sup>1</sup> and Asif Khateeb<sup>1</sup>

<sup>1</sup> Epigenetics & Function Group, Hohai University, Nanjing, China, <sup>2</sup> School of Public Health, Shanghai Jiao Tong University, Shanghai, China

## OPEN ACCESS

### Edited by:

Juan Caballero,  
Universidad Autónoma de Querétaro,  
Mexico

### Reviewed by:

Wenhai Zhang,  
Hengyang Normal University, China  
Zhuliang Yu,  
South China University of Technology,  
China

### \*Correspondence:

Binhua Tang  
bh.tang@hhu.edu.cn

<sup>†</sup>These authors have contributed  
equally to this work

### Specialty section:

This article was submitted to  
Bioinformatics and Computational  
Biology,  
a section of the journal  
Frontiers in Genetics

**Received:** 20 August 2018

**Accepted:** 27 February 2019

**Published:** 26 March 2019

### Citation:

Tang B, Pan Z, Yin K and Khateeb A  
(2019) Recent Advances of Deep  
Learning in Bioinformatics and  
Computational Biology.  
*Front. Genet.* 10:214.  
doi: 10.3389/fgene.2019.00214

Extracting inherent valuable knowledge from omics big data remains as a daunting problem in bioinformatics and computational biology. Deep learning, as an emerging branch from machine learning, has exhibited unprecedented performance in quite a few applications from academia and industry. We highlight the difference and similarity in widely utilized models in deep learning studies, through discussing their basic structures, and reviewing diverse applications and disadvantages. We anticipate the work can serve as a meaningful perspective for further development of its theory, algorithm and application in bioinformatic and computational biology.

**Keywords:** computational biology, bioinformatics, application, algorithm, deep learning

## INTRODUCTION

Deep learning is the emerging generation of the artificial intelligence techniques, specifically in machine learning. The earliest artificial intelligence was firstly implemented on hardware system in the 1950s. The newer concept with the more systematic theorems, named machine learning, appeared in the 1960s. And its newly-evolved branch, deep learning, was first brought up around the 2000s, and soon led to rapid applications in different fields, due to its unprecedented prediction performance on big data (Hinton and Salakhutdinov, 2006; LeCun et al., 2015; Nussinov, 2015).

The basic concepts and models in deep learning have derived from the artificial neural network, which mimic human brain's activity pattern to intelligentize the algorithms and save tedious human labor (Mnih et al., 2015; Schmidhuber, 2015; Mamoshina et al., 2016). Although deep learning is an emerging subfield recently from machine learning, it has immense utilizations spreading from machine vision, voice, and signal processing, sequence and text prediction, and computational biology topics, altogether shaping the productive AI fields (Bengio and LeCun, 2007; Alipanahi et al., 2015; Libbrecht and Noble, 2015; Zhang et al., 2016; Esteva et al., 2017; Ching et al., 2018). Deep learning has several implementation models as artificial neural network, deep structured learning, and hierarchical learning, which commonly apply a class of structured networks to infer the quantitative properties between responses and causes within a group of data (Ditzler et al., 2015; Liang et al., 2015; Xu J. et al., 2016; Giorgi and Bader, 2018).

The subsequent paragraphs mainly summarize the essential concepts and recent applications of deep learning, together highlight the key achievements and future directions of deep learning, especially from the perspectives of bioinformatics and computational biology.

## ESSENTIAL CONCEPTS IN DEEP NEURAL NETWORK

### Basic Structure of Neural Network

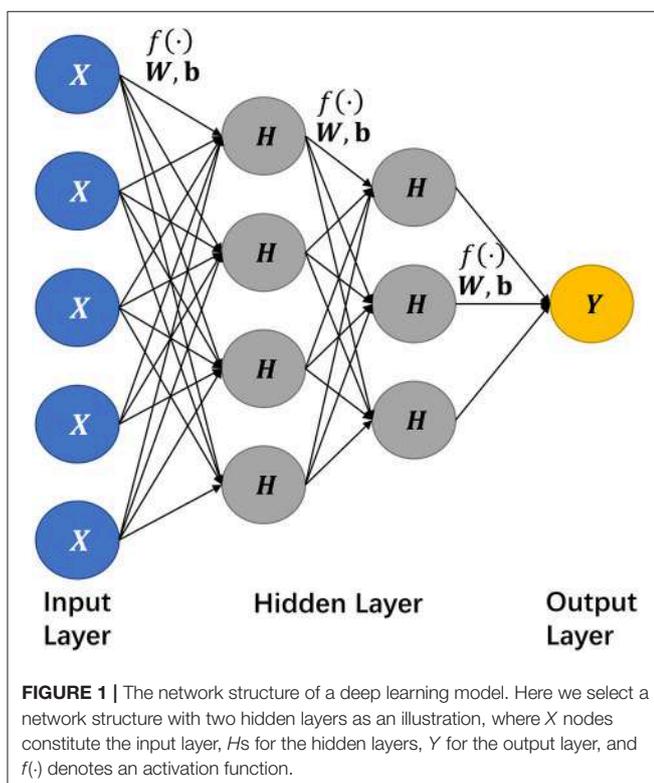
Neural network is a class of information processing modules, frequently utilized in machine learning. Within a multi-layer context, the basic building units, namely neurons, are connected to each other among the adjacent layers via internal links, but the neurons belonging to the same layer have no connection, as depicted in **Figure 1**.

In **Figure 1**, each hidden layer processes its inputs via a connection function denoted as below,

$$h_{W,b}(X) = f(W^T X + b) \quad (1)$$

where  $W$  refers to the weight and  $b$  for bias. When all input layer neurons are active, each input neuron will multiply their respective weight matrix and the output will be summed up with a bias, which then will be fed into an adjacent hidden layer. Although the input-output formalization may repeat similarly among hidden layers, there is usually no direct connection between neurons within the same layer. And activation function is to quantify the connection between two neighboring neurons across two (hidden) layers.

Specifically, the input of the activation function is the combination  $W^T X + b$  denoted in Equation (1), and the function output is then fed into the next neuron as a new input. Following the connection formula, the former input feature can be extracted to the next layer; by this means the features can be



well-extracted and refined further. And the performance of the feature extraction depends significantly on the selection of the activation function.

Before training the network structure, the input raw datasets are usually separated into two or three groups, namely a training set and a test set, sometimes a validation set to examine the performance of previously trained network models, as depicted in **Figure 2**. In practice, the original datasets are separated stochastically to avoid the potential local tendency, but the proportion of each set can be determined manually.

### Learning by Training, Validation, and Testing

Normally, training a neural network refers to a process the network self-tunes its parameters or weights to meet the prespecified performance criteria, thus the trained model can be further used in regression or classification purposes. As depicted in **Figure 2**, generally a complete dataset collected from a specific experiment beforehand can be split into the training and testing, and even validation sets, then followed by conventional tasks as model training, validation and performance comparison.

During training with initial batches of data samples, model parameters and their characteristics normally can be tuned by various learning paradigms, including appropriate activation and rectification functions. Then the trained network should be further tested or even validated with the other batch of samples, to acquire high robustness and satisfactory predictability, the processes of which are often referred as model testing and validation.

Usually, the three procedures above are faithfully implemented in conventional machine learning studies; and even in its quickly-evolving subfield, deep learning, the similar paradigm is always observed (LeCun et al., 2015; Schmidhuber, 2015).

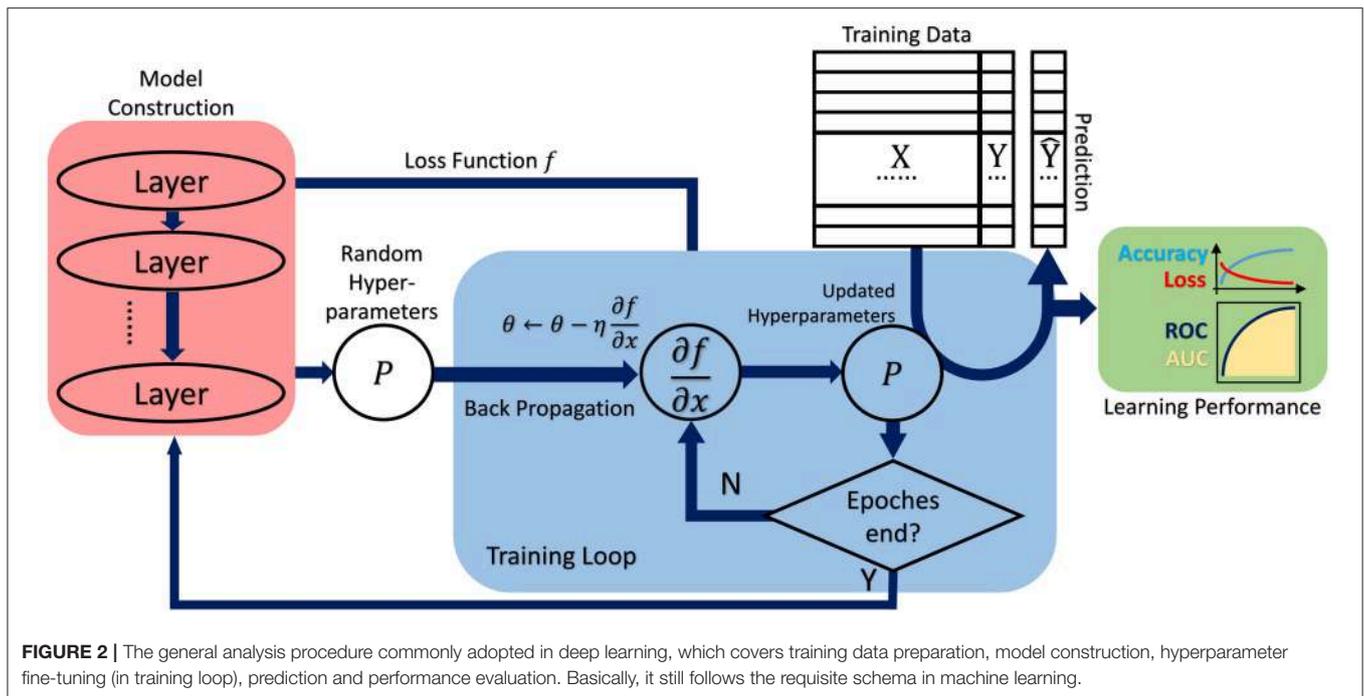
### Activation and Loss Function

After training completed, the neural network can perform regression or classification task on testing data, while there usually exists the difference between the predicted outputs and actual values. And the difference should be minimized to acquire optimal model performance.

Within a certain layer, error reduction requires scaling it back within a preset range before passing it onto the next layer of neurons. Activation herein is defined to control neurons' outputs in "active" or "inactive" status, using those non-linear functions as rectified linear unit (ReLU), tanh, and logistic (Sigmoid or soft step) (LeCun et al., 2015).

Besides, a loss function herein is to measure the total difference between the predicted and accurate values, through fine-tuning in backpropagation process. And it acts as an ending threshold for parameter optimization by means of iteratively evaluating the trained models.

With activation function in each neuron throughout diverse layers, a training procedure will continue searching a whole hyperparameter space till the ending threshold, compare and detect an optimal parameter combination by minimizing the preset loss function.



## TYPICAL ALGORITHMS AND APPLICATIONS

With the substantial progresses in advanced computation and Graphic Processing Unit (GPU) technologies, systematic interrogation into massive data to understand its inherent mechanisms becomes possible, especially through deep learning approaches. Hereinafter, we illustrated several frequently utilized models in deep learning literatures, in both recent computation theories and diverse applications.

### Recurrent Neural Network

Recurrent Neural Network (RNN) is a deep learning model different from traditional neural networks, since the former can integrate the previously learned status through a recurrent approach, namely backpropagation; while traditional neural network usually outputs prediction based on the status of the current layer.

Compared with traditional network models, RNN only has one hidden layer but it can unfold horizontally, and multi-vertical-groups are enabled to utilize most of the previous results, namely “using memory”.

As depicted in **Figure 3**, the hidden layer neuron  $H_n$  is defined by Equation (2),

$$H_n = \sigma_1(W_{1,n}^T H_{n-1} + W_{2,n}^T X_n + b_{1,n}) \quad (2)$$

where  $W_{1,n}$  and  $W_{2,n}$  represent weight matrix,  $b_{1,n}$  is a bias matrix, and  $\sigma(\cdot)$  (usually  $\tanh(\cdot)$ ) for an activation function. Thus, each layer will generate a partial of output from the current hidden layer neuron with a weight matrix  $W_{3,n}$  and bias  $b_{2,n}$ ,

defined by Equation (3),

$$\hat{Y}_n = \sigma_2(W_{3,n} H_n + b_{2,n}) \quad (3)$$

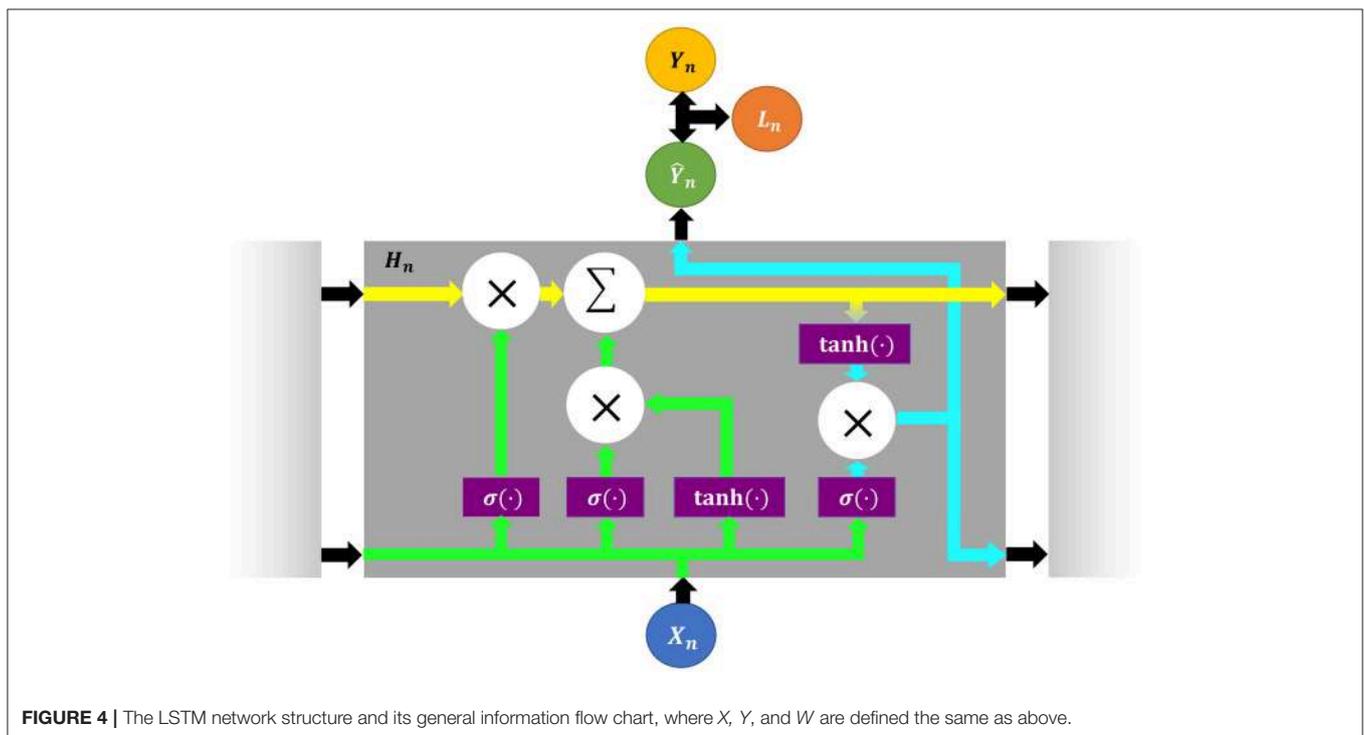
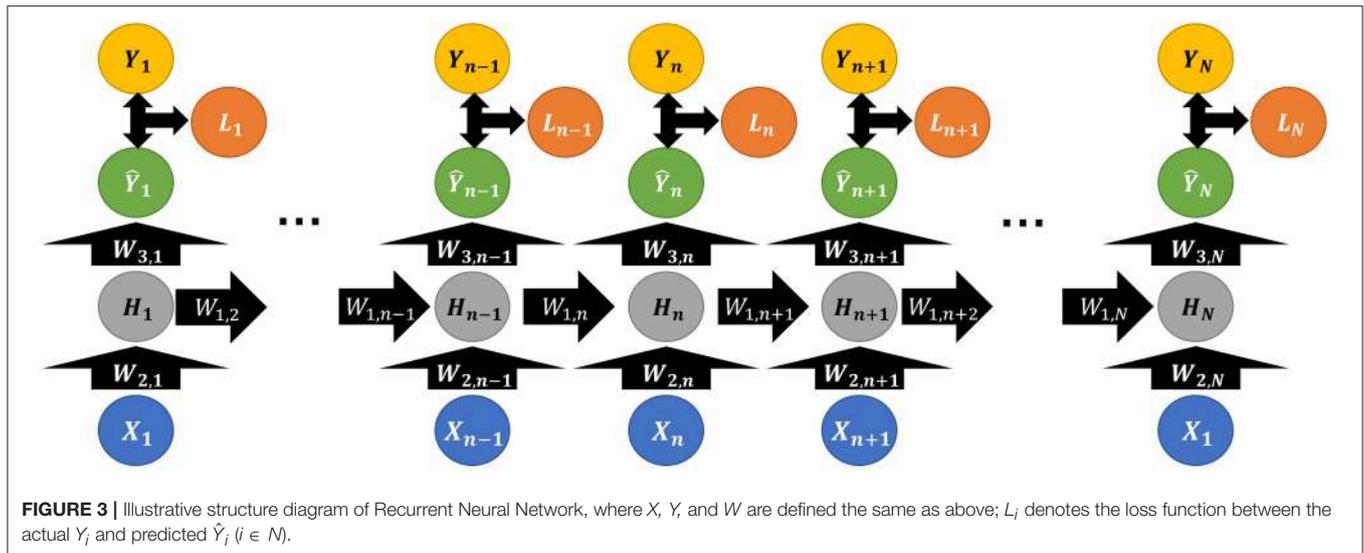
And the total loss  $L_{total}$  will be the sum of the loss functions from each hidden layer, defined as below,

$$L_{total} = \sum_{n=1}^N L_n = \sum_{n=1}^N L(\hat{Y}_n, Y_n) \quad (4)$$

Thus, fine tuning of RNN backpropagation is based on three weights,  $W_{1,n}$ ,  $W_{2,n}$ , and  $W_{3,n}$ . Since the multi-parameter setting in weights adds to the optimization burden, RNN usually performs worse than Convolutional Neural Network (CNN) in terms of fine-tuning. But frequently it is ensembled with CNN in diverse applications, such as dimension reduction, image, and video processing (Hinton and Salakhutdinov, 2006; Hu and Lu, 2018). Angermueller et al. proposed an ensembled RNN-CNN architecture, DeepCpG, on single-cell DNA methylation data, to better predict missing CpG status for genome-wide analysis; together the model’s interpretable parameters shed light on the connection between sequence composition and methylation variability (Angermueller et al., 2017). Section Autoencoder will specifically discuss CNN and its typical applications.

Moreover, RNN outperforms those conventional models as logistic regression and SVM, and it can be implemented in various environments, accelerated by GPUs (Li et al., 2017). Due to its structural characteristics, RNN is suitable to deal with long and sequential data, such as DNA array and genomics sequence (Pan et al., 2008; Ray et al., 2009; Jolma et al., 2013; Lee and Young, 2013; Alipanahi et al., 2015; Xu T. et al., 2016).

But RNN cannot interact with hidden neurons far from the current one. To construct an efficient framework

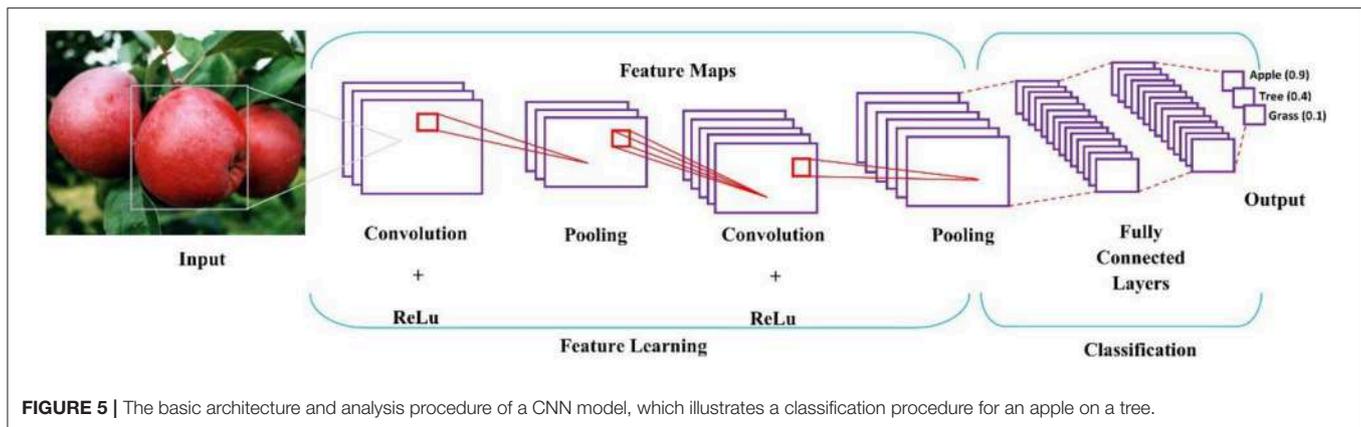


of recalling deep memory, many improved algorithms have been proposed, like BRNN in protein secondary structure prediction (Baldi et al., 1999), and MD-RNN in analyzing electron microscopy and MRIs of breast cancer samples (Kim et al., 2018).

LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) are two recently-improved derivatives of RNN to solve the long-time dependence issues. GRU shares a similar structure with LSTM, which has several gates used for modeling its memory center. The current memory output is jointly influenced by its current input feature, the context (namely the

past influence), and the inner action toward the input, as shown in Figure 4.

In Figure 4, the yellow track refers to an input gate transferring its total past features, and is accessible for any new feature to be added. The green track is a mixture of an input gate and its former hidden layer neurons; and it decides what to omit, namely resetting activation function close to 0, and what to be updated into the yellow track. The blue track is the output gate integrating the inner influence from the yellow track, and it decides the output of the current hidden neurons and what to be passed to the next hidden neuron.



Recently an attention-based architecture, DeepDiff, utilizes a hierarchy of LSTM modules to characterize how various histone modifications cooperate simultaneously, and it can effectively predict cell-type-specific gene expression (Sekhon et al., 2018).

## Convolutional Neural Network

Convolutional neural networks (CNN or ConvNet) are suitable to process information in the form of multiple arrays (LeCun et al., 2015; Esteva et al., 2017; Hu and Lu, 2018). To reduce the parameters without compromising its learning capacity is the general design principle of CNN (LeCun et al., 2015; Krizhevsky et al., 2017). And each convolution kernel's parameters in CNN are trained by the backpropagation algorithm.

Especially in image-related applications, CNN can cope with pixel scanning and processing, thus it greatly accelerates the implementation of optimized algorithms into practice (Esteva et al., 2017; Quang et al., 2018). Structurally, CNN consists of linear convolution operation, followed by nonlinear activators, pooling layers, and deep neural network classifier, depicted in Figure 5.

In Figure 5, several filters are applied to convolve an input image, and its output is subsampled as a new input into the next layer; and convolution and subsampling processes are repeated till high level features, namely shapes, can be extracted. The more layers a CNN model has, the higher-level features it will extract.

In feature learning, convolution operation is to scan a 2D image with a given pattern, and calculate the matching degree at each step, then pooling identifies the pattern presence in the scanned region (Angermueller et al., 2016). Activation function defines a neuron's output based on a set of given inputs. The weighted sum of inputs is passed through an activation function for non-linear transformation. A typical activation function returns a binary output, 0 or 1; when a neuron's accumulation exceeds a preset threshold, the neuron is activated and passes its information to the next layers; otherwise, the neuron is deactivated. Sigmoid, tanh, ReLU, leaky ReLU, and softmax are the commonly used activation functions (LeCun et al., 2015; Schmidhuber, 2015).

Through pooling layers, pixels are stretched to a single column vector. The vectorized and concatenated pixel information is fed into dense layers, known as fully connected layers for

further classification. The fully-connected layer renders the final decision, where CNN returns a probability that an object in the image belongs to a specific type.

Following the fully-connected layer is a loss layer, which adjusts their weights across the network. A loss function is used to measure the model performance and inconsistency between the actual and predicted values. Model performance increases with decreasing of the loss function. For an output vector  $y_i$  and an input  $x=(x_1, x_2, \dots, x_n)$ , the mapping loss function  $L(\cdot)$  between  $x$  and  $y$  is defined as,

$$L(y_i, \hat{y}_i) = \frac{1}{n} \sum_{i=1, j=1}^{n, k} \varphi[y_i, f(x_i, \sigma_i, \omega_{ij}, b_i)] \quad (5)$$

where  $\varphi$  denotes an empirical risk for each output,  $\hat{y}_i$  for the  $i$ -th prediction,  $n$  the total number of training samples,  $k$  the count of the weights  $\omega_{ij}$  and  $b_i$  the bias for the activation function  $\sigma_i$ .

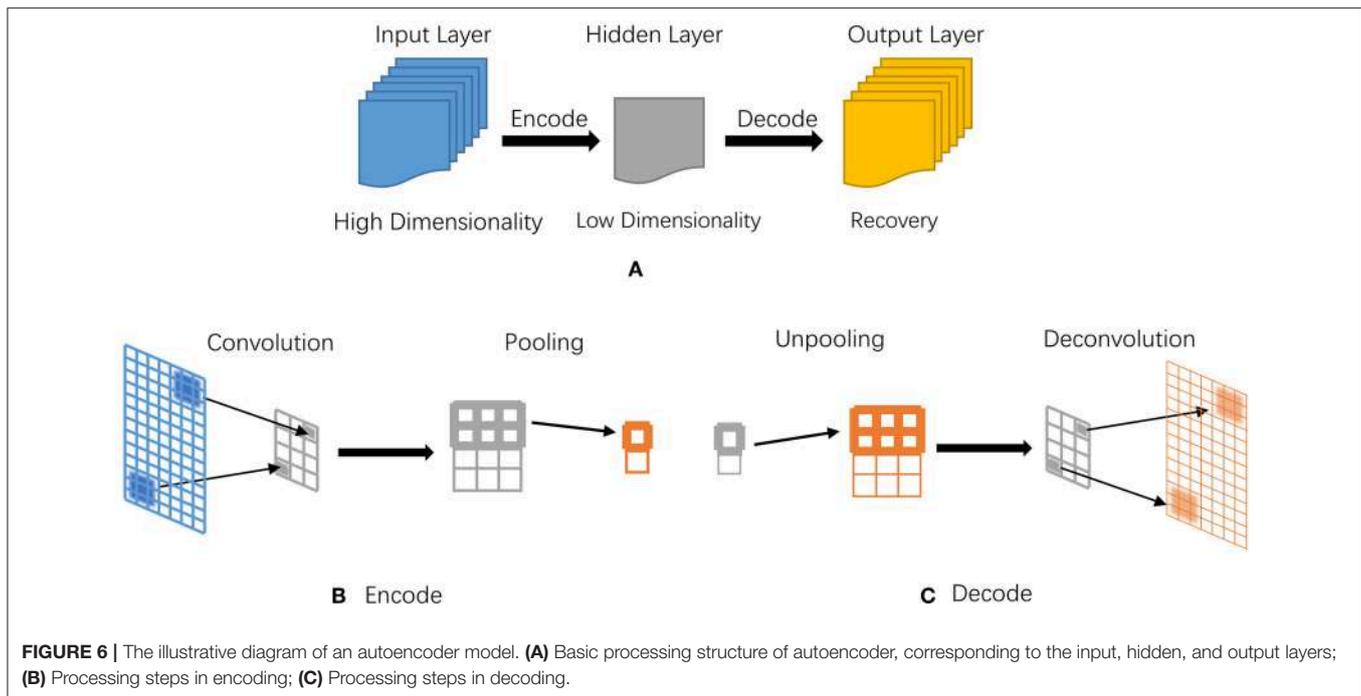
Recently, CNN has been adopted rapidly in biomedical imaging studies for its outstanding performance in computer vision and concurrent computation with GPUs (Ravi et al., 2017). Usually convolution-pooling structure can better learn imaging features from CT scans and MRI images from head trauma, stroke diagnosis and brain EPV (enlarged perivascular space) detection (Chilamkurthy et al., 2018; Dubost et al., 2019).

In recent computational biology, a discriminative CNN framework, DeepChrome, is proposed to predict gene expression by feature extraction from histone modification. And the deep learning model outperforms traditional Random Forests and SVM on 56 cell types from REMC database (Singh et al., 2016).

Furthermore, CNN can be combined with other deep learning models, such as RNN to predict imaging content, where CNN encodes an image and RNN generates the corresponding image description (Angermueller et al., 2016). Till now, quite a few variants of CNN have been also proposed in diverse classification applications, like AlexNet with GPU support and DQN in reinforcement learning (Mnih et al., 2015).

## Autoencoder

Through an unsupervised manner, autoencoder is another typical artificial neural network, designed to precisely extract coding or



representation features using data-driven learning (Min et al., 2017; Zeng et al., 2017; Yang et al., 2018). For high-dimensional data, it is time-consuming and infeasible to load all raw data into a network, thus dimension reduction or compression is a necessity in preprocessing of raw data.

Autoencoder can compress and encode information from the input layer into a short code, then after specific processing, it will decode into the output closely matching the original input. **Figure 6** illustrates its basic model structure and processing steps.

Convolution and pooling are two major steps in encoder, depicted in **Figure 6B**; while decoder has two complete opposite steps, namely unpooling and deconvolution in **Figure 6C**. Both convolution and pooling can compress data while preserving the most representative features in two different ways. Convolution involves continuously scanning data with a rectangle window, for example a  $3 \times 3$  size; after each scanning, the window moves to a next position, namely pixel, by replacing the oldest elements with new ones, together with convolution operation. After the whole scanning and convolution, pooling is utilized to deeper compress on redundancy.

Similar to traditional PCA in dimension reduction to some extent, but autoencoder is more robust and effective in extracting data features for its non-linear transformation in hidden layers. Given an input  $x$ , the model extracts its main feature and generates  $\hat{x} = Wb$ , where  $W$  and  $b$  denote weighting and bias vectors, respectively. Commonly, the output cannot fit the input precisely, which can be measured with a loss function in mean squared error (MSE) defined in Equation (6),

$$L(W, b) = \frac{1}{m} \sum_{i=1}^m (\hat{x} - x)^2 \quad (6)$$

Thus, the learning process is to minimize the loss  $L$  after iterative optimization.

Recently, sparse autoencoder (SAE) is frequented discussed for its admirable performance in dimension reduction and denoising corrupted data. And the loss function in SAE is defined in Equation (7),

$$L_{SAE} = L(W, b) + \beta \sum_k KL(\rho || \hat{\rho}_k) \quad (7)$$

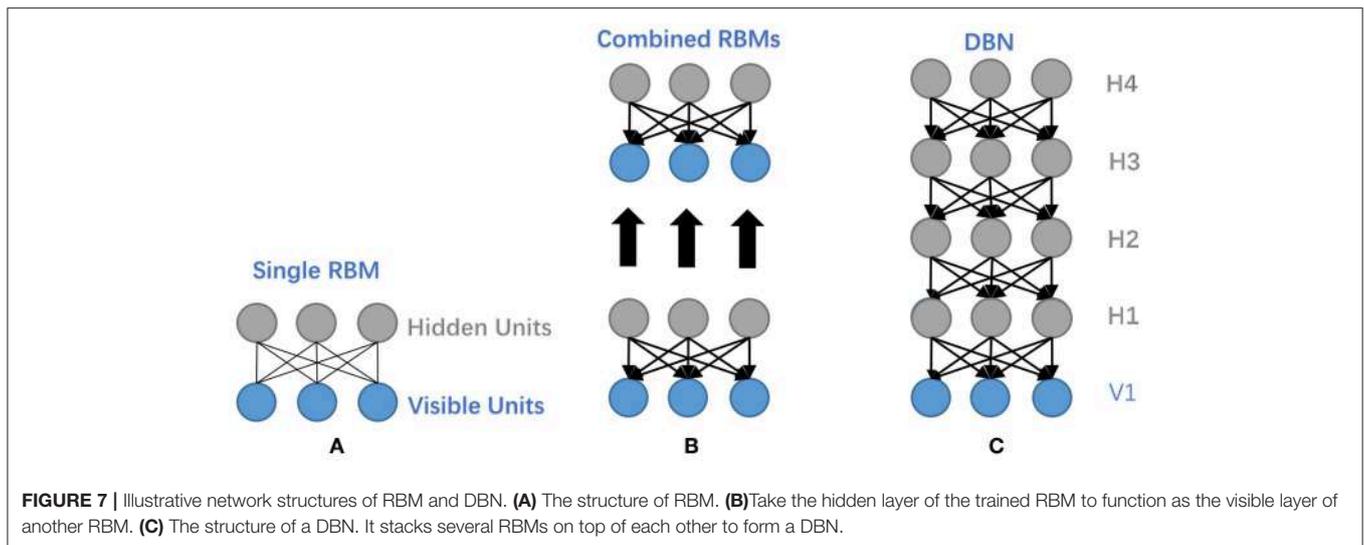
where KL refers to KL-divergence in Equation (10),  $\rho$  for the activation level of neurons, usually set as 0.05 in condition of sigmoid, indicating most neurons are inactive,  $\rho_k$  for the average activation level of neuron  $k$ , and  $\beta$  for the regularization coefficient.

$$KL(\rho || \hat{\rho}_k) = \rho \log \frac{\rho}{\hat{\rho}_k} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_k} \quad (8)$$

where  $\hat{\rho}_k$  represents the average activation level of test samples, and  $x^{(i)}$  is the  $i$ -th test sample in Equation (9).

$$\hat{\rho}_k = \frac{1}{m} \sum_i [a_j(x^{(i)})] \quad (9)$$

For high dimensional data, multiple autoencoders can be stacked to act as a deep autoencoder (Hinton and Salakhutdinov, 2006). And this architecture may lead to vanishing gradient, due to its gradient-based and backpropagation learning, and the current solutions include adopting ReLu activation and dropout (Szegedy et al., 2015; Krizhevsky et al., 2017). During configuration and pretraining, the model weights can be acquired by greedy layer-wise training, then the network can be fine-tuned with the backpropagation algorithm.



Many variations of autoencoder have been proposed recently, such as sparse autoencoder (SAE), denoising autoencoder (DAE). Typically, stacked sparse autoencoder (SSAE) was proposed to analyze high-resolution histopathological images in breast cancer (Xu J. et al., 2016). By using SAE with three iterations, Heffernan et al. reported the successful prediction of protein secondary structure, local backbone angles, and solvent accessible surface area (Heffernan et al., 2015). Miotto et al. introduced a stack of DAEs to predict features from a large scale of electronic health records (EHR), via an unsupervised representation approach (Miotto et al., 2016). Ithapu et al. proposed a randomized denoising autoencoder marker (rDAM) to predict future cognitive and neural decline for Alzheimer diseases, with its performance surpassing the existing methods (Ithapu et al., 2015).

## Deep Belief Network

As a generative graphical model, Deep Belief Network (DBN) is composed of multiple Restricted Boltzmann Machines (RBM) or autoencoders stacked on top of each other, where each hidden layer in subnetworks serves as a visible layer for the next layer (Hinton et al., 2006). The main network structures of RBM and DBN are depicted in **Figure 7**, where it manifests the construction relations between the two network models.

DBN trains layer by layer in an unsupervised greedy approach to initialize network weights, separately; then it can utilize the wake-sleep or backpropagation algorithm during fine-tuning. While for traditional backpropagation used in fine-tuning, DBN may encounter several problems: (1) requiring labeled data for training; (2) low learning rate; (3) inappropriate parameters tending to acquire local optimum.

Within recent applications, Plis et al. classified schizophrenia patients based on brain MRIs with DBN (Plis et al., 2014); in drug design based on high-throughput screening, DBN was exploited to perform quantitative structure activity relationship (QSAR)

study. And the result showed that the optimization in parameter initialization highly improves the capability of DNN to provide high-quality model predictions (Ghasemi et al., 2018). DBN was also used to study the combination of resting-state fMRI (rs-fMRI), gray matter, and white matter data by exploiting the latent and abstract high-level features (Akhavan Aghdam et al., 2018). Meanwhile, DBN and CNN were compared to prove that deep learning has better discriminative results and holds promise in the medical image diagnosis (Hua et al., 2015).

## Transfer Learning in Deep Learning

Besides the above deep learning models, transfer learning is frequently utilized in specific cases without sufficient labeling information or dimensionality (Pan and Yang, 2010). Although conceptually it does not belong to deep learning, due to its transferability of high-level semantic classification for deep neural network, transfer learning has gained emerging notices from deep learning fields (O'Shea et al., 2013; Anthimopoulos et al., 2016).

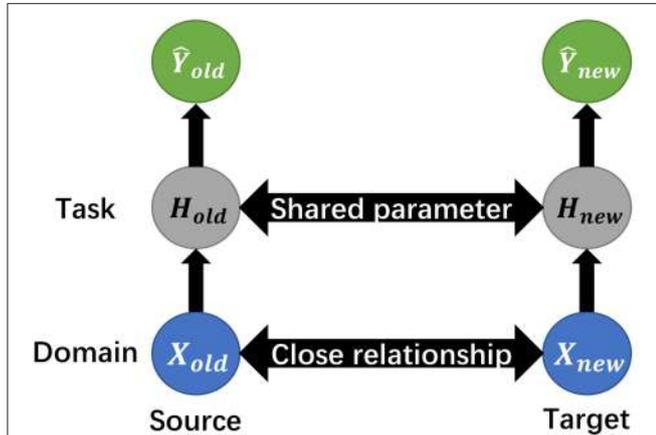
In quite a few deep learning studies, transfer learning enables a previously-trained model to transfer its optimized parameters to a new model, thus to implement the knowledge transmission and reduce repetitive training from scratch, as depicted in **Figure 8**.

Normally, source and target domains have certain statistical relationship or similarity that directly affects the transferability. The domain contains the original dataset, for example image matrix, and the task refers to certain processes, like classification or pattern recognition. The mission of transfer learning includes transferring not only the parameters like weight, but the concentrated small-size matrix from the origin data domain called knowledge distillation.

The knowledge distillation usually uses both "hard target" and "soft target" to train the model and obtain lower information entropy. The below softmax function is usually utilized to soften

the sparse data and excavate its inherent features,

$$f(\alpha_k) = \frac{e^{\frac{\alpha_k}{T}}}{\sum_k e^{\frac{\alpha_k}{T}}} \quad (10)$$



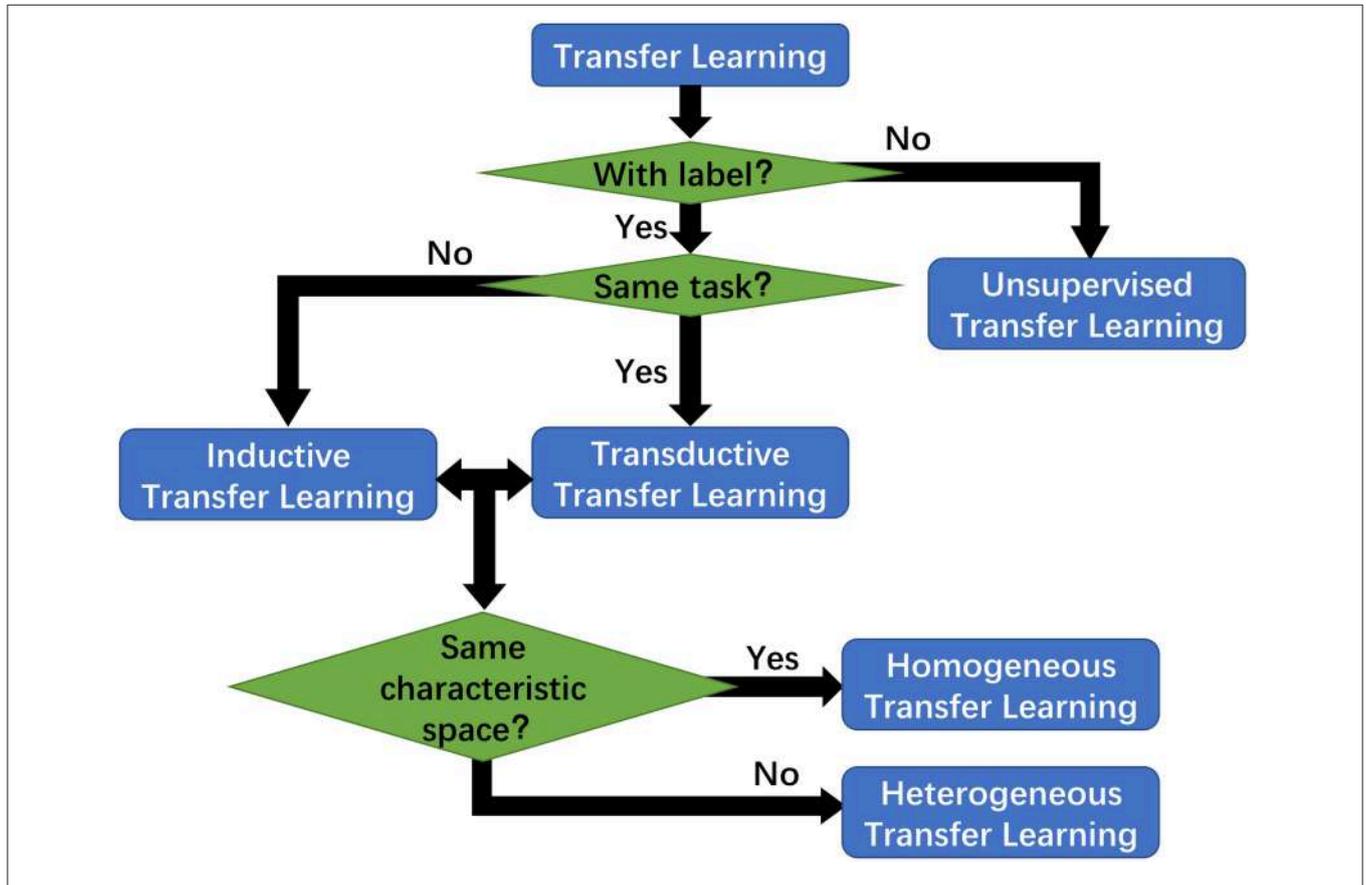
**FIGURE 8** | The schematic illustration of transfer learning. Given source domain and its learning task, together with target domain and respective task, transfer learning aims to improve the learning of the target prediction function, with the knowledge in source domain and its task.

where the logical judger  $\alpha_k$  is the input,  $f(\cdot)$  is to soft target data and can offer smaller gradient variance,  $k$  denotes the  $k$ -th segmented data slice. The parameter  $T$  is called temperature and the larger  $T$  is, the softer the target is.

Furthermore, transfer learning is categorized into instance-based, feature-based, parameter-based and relation-based derivatives, depicted in **Figure 9**. Currently transfer learning is frequently discussed in the deep learning fields for its great applicability and performance. Ensembled with CNN, transfer learning can attain greater prediction performance of interstitial lung disease CT scans (Anthimopoulos et al., 2016). It was also used as a ligament between the multi-layer LSTM and conditional random field (CRF), and the result showed that the LSTM-CRF approach outperformed the baseline methods on the target datasets (Giorgi and Bader, 2018).

### CONCLUSIONS

Within the work, we comprehensively summarized the basic but essential concepts and methods in deep learning, together with its recent applications in diverse biomedical studies. Through reviewing those typical deep learning models as RNN, CNN, autoencoder, and DBN, we highlight that the specific application scenario or context, such as data feature and model applicability,



**FIGURE 9** | Transfer learning has several derivatives categorized by the labeling information and difference between the target and source.

are the prominent factors in designing a suitable deep learning approach to extract knowledge from data; thus, how to decipher and characterize data feature is not a trivial work in deep-learning workflow yet. In recent deep learning studies, many derivatives from classic network models, including the network models depicted above, manifest that model selection affects the effectiveness of deep learning application.

Secondly, for its limitation and further improvement direction, we should revisit the nature of the method: deep learning is essentially a continuous manifold transformation among diverse vector spaces, but there exist quite a few tasks cannot be converted into a deep learning model, or in a learnable approach, due to the complex geometric transform. Moreover, deep learning is generally a big-data-driven technique, which has made it unique from conventional statistical learning or Bayesian approaches. Thus, it is a new direction for deep learning to integrate or embed with other conventional algorithms in tackling those complicated tasks.

Thirdly, when it comes to innovation in computational algorithm and hardware. As an inference technique driven by big data, deep learning demands parallel computation facilities of high performance, together with more algorithmic breakthroughs and fast accumulation of diverse perceptual data, it is achieving pervasive successes in many fields and applications. Particularly in bioinformatics and computational biology, which is a typical data-oriented field, it has witnessed the remarkable changes taken place in its research methods.

Finally, as unprecedented innovation and successes acquired with deep learning in diverse subfields, some even argued that

deep learning could bring about another wave like the internet. In the long term, deep learning technique is shaping the future of our lives and societies to its full extent. But deep learning should not be misinterpreted or overestimated either in academia or AI industry, and actually it has lots of technical problems to solve due to its nature. In all, we anticipate this review work will provide a meaningful perspective to help our researchers gain comprehensive knowledge and make more progresses in this ever-faster developing field.

## AUTHOR CONTRIBUTIONS

BT conceived the study. ZP, KY, AK, and BT drafted the application sections and revised and approved the final manuscript.

## FUNDING

This work was supported by the Natural Science Foundation of Jiangsu, China (BE2016655 and BK20161196), and the Fundamental Research Funds for China Central Universities (2019B22414). This work made use of the resources supported by the NSFC-Guangdong Mutual Funds for Super Computing Program (2nd Phase), and the Open Cloud Consortium sponsored project resource, supported in part by grants from Gordon and Betty Moore Foundation and the National Science Foundation (USA) and major contributions from OCC members.

## REFERENCES

- Akhavan Aghdam, M., Sharifi, A., and Pedram, M. M. (2018). Combination of rs-fMRI and sMRI data to discriminate autism spectrum disorders in young children using deep belief network. *J. Digit. Imaging*. 31, 895–903. doi: 10.1007/s10278-018-0093-8
- Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* 33:831–838. doi: 10.1038/nbt.3300
- Angermueller, C., Lee, H. J., Reik, W., and Stegle, O. (2017). DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome Biol.* 18:67. doi: 10.1186/s13059-017-1189-z
- Angermueller, C., Pärnamaa, T., Parts, L., and Stegle, O. (2016). Deep learning for computational biology. *Mol. Syst. Biol.* 12:878. doi: 10.15252/msb.20156651
- Anthimopoulos, M., Christodoulidis, S., Ebner, L., Christe, A., and Mougiakakou, S. (2016). Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. *IEEE Trans. Med. Imag.* 35, 1207–1216. doi: 10.1109/TMI.2016.2535865
- Baldi, P., Brunak, S., Frasconi, P., Soda, G., and Pollastri, G. (1999). Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics* 15:937. doi: 10.1093/bioinformatics/15.11.937
- Bengio, Y., and LeCun, Y. (2007). “Scaling learning algorithms toward AI,” in *Large-Scale Kernel Machines*, eds L. Bottou, O. Chapelle, D. DeCoste and J. Weston (Cambridge, MA: The MIT Press).
- Chilamkurthy, S., Ghosh, R., Tanamala, S., Biviji, M., Campeau, N. G., Venugopal, V. K., et al. (2018). Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study. *Lancet* 392, 2388–2396. doi: 10.1016/S0140-6736(18)31645-3
- Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., et al. (2018). Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* 15:20170387. doi: 10.1098/rsif.2017.0387
- Ditzler, G., Polikar, R., Member, S., Rosen, G., and Member, S. (2015). Multi-layer and recursive neural networks for metagenomic classification. *IEEE Trans. Nanobiosci.* 14:608. doi: 10.1109/TNB.2015.2461219
- Dubost, F., Adams, H., Bortsova, G., Ikram, M. A., Niessen, W., Vernooij, M., et al. (2019). 3D regression neural network for the quantification of enlarged perivascular spaces in brain MRI. *Med. Image Anal.* 51, 89–100. doi: 10.1016/j.media.2018.10.008
- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542:115–118. doi: 10.1038/nature21056
- Ghasemi, F., Mehridehnavi, A., Fassihi, A., and Pérez-Sánchez, H. (2018). Deep neural network in QSAR studies using deep belief network. *Appl. Soft Comput.* 62, 251–258. doi: 10.1016/j.asoc.2017.09.040
- Giorgi, J. M., and Bader, G. D. (2018). Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics* 34, 4087–4094. doi: 10.1093/bioinformatics/bty449
- Heffernan, R., Paliwal, K., Lyons, J., Dehzangi, A., Sharma, A., Wang, J., et al. (2015). Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Sci. Rep.* 5:11476. doi: 10.1038/srep11476
- Hinton, G. E., Osindero, S., and Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural. Comput.* 18, 1527–1554. doi: 10.1162/neco.2006.18.7.1527
- Hinton, G. E., and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science* 313, 504–507. doi: 10.1126/science.1127647

- Hu, Y., and Lu, X. (2018). Learning spatial-temporal features for video copy detection by the combination of CNN and RNN. *J. Vis. Commun. Image Rep.* 55, 21–29. doi: 10.1016/j.jvcir.2018.05.013
- Hua, K. L., Hsu, C. H., Hidayati, H. C., Cheng, W. H., and Chen, Y. J. (2015). Computer-aided classification of lung nodules on computed tomography images via deep learning technique. *Oncotargets Ther.* 8:2015–2022. doi: 10.2147/OTT.S80733
- Ithapu, V. K., Singh, V., Okonkwo, O. C., Chappell, R. J., Dowling, N. M., and Johnson, S. C. (2015). Imaging-based enrichment criteria using deep learning algorithms for efficient clinical trials in mild cognitive impairment. *Alzheimer's Dement.* 11:1489–1499. doi: 10.1016/j.jalz.2015.01.010
- Jolma, A., Yan, J., Whittington, T., Toivonen, J., Nitta Kazuhiro, R., Rastas, P., et al. (2013). DNA-binding specificities of human transcription factors. *Cell* 152, 327–339. doi: 10.1016/j.cell.2012.12.009
- Kim, Y., Sim, S. H., Park, B., Lee, K. S., Chae, I. H., Park, I. H., et al. (2018). MRI assessment of residual breast cancer after neoadjuvant chemotherapy: relevance to tumor subtypes and MRI interpretation threshold. *Clin. Breast Cancer* 18, 459–467.e1 doi: 10.1016/j.clbc.2018.05.009
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436. doi: 10.1038/nature14539
- Lee, T. I., and Young, R. A. (2013). Transcriptional regulation and its misregulation in disease. *Cell* 152, 1237–1251. doi: 10.1016/j.cell.2013.02.014
- Li, A., Serban, R., and Negrut, D. (2017). Analysis of a splitting approach for the parallel solution of linear systems on GPU cards. *SIAM J. Sci. Comput.* 39, C215–C237. doi: 10.1137/15M1039523
- Liang, M., Li, Z., Chen, T., and Zeng, J. (2015). Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach. *IEEE/ACM Trans. Comput. Biol. Bioinf.* 12, 928–937. doi: 10.1109/TCBB.2014.2377729
- Libbrecht, M. W., and Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nat. Rev. Genet.* 16:321–322. doi: 10.1038/nrg3920
- Mamoshina, P., Vieira, A., Putin, E., and Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. *Mol. Pharmaceut.* 13, 1445–1454. doi: 10.1021/acs.molpharmaceut.5b00982
- Min, S., Lee, B., and Yoon, S. (2017). Deep learning in bioinformatics. *Brief Bioinform.* 18, 851–869. doi: 10.1093/bib/bbw068
- Miotto, R., Li, L., Kidd, B. A., and Dudley, J. T. (2016). Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Sci. Rep.* 6:26094. doi: 10.1038/srep26094
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236
- Nussinov, R. (2015). Advancements and challenges in computational biology. *PLoS Comput. Biol.* 11:e1004053. doi: 10.1371/journal.pcbi.1004053
- O'Shea, J. P., Chou, M. F., Quader, S. A., Ryan, J. K., Church, G. M., and Schwartz, D. (2013). pLogo: a probabilistic approach to visualizing sequence motifs. *Nat. Methods* 10, 1211–1212. doi: 10.1038/nmeth.2646
- Pan, Q., Shai, O., Lee, L. J., Frey, B. J., and Blencowe, B. J. (2008). Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nat. Genet.* 40, 1413–1415. doi: 10.1038/ng.259
- Pan, S. J., and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22, 1345–1359. doi: 10.1109/TKDE.2009.191
- Plis, S. M., Hjelm, D. R., Salakhutdinov, R., Allen, E. A., Bockholt, H. J., Long, J. D., et al. (2014). Deep learning for neuroimaging: a validation study. *Front. Neurosci.* 8:229. doi: 10.3389/fnins.2014.00229
- Quang, D., Guan, Y., and Parker, S. C. J. (2018). YAMDA thousandfold speedup of EM-based motif discovery using deep learning libraries and GPU. *Bioinformatics* 34, 3578–3580. doi: 10.1093/bioinformatics/bty396
- Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., et al. (2017). Deep learning for health informatics. *IEEE J. Biomed. Health Inform.* 21, 4–21. doi: 10.1109/JBHI.2016.2636665
- Ray, D., Kazan, H., Chan, E. T., Peña, L. C., Chaudhry, S., Talukder, S., et al. (2009). Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins. *Nat. Biotechnol.* 27, 667–670. doi: 10.1038/nbt.1550
- Schmidhuber, J. (2015). Deep learning in neural networks: an overview. (2015). *Neural. Net.* 61:85. doi: 10.1016/j.neunet.2014.09.003
- Sekhon, A., Singh, R., and Qi, Y. (2018). DeepDiff: DEEP-learning for predicting DIFFerential gene expression from histone modifications. *Bioinformatics* 34, i891–i900. doi: 10.1093/bioinformatics/bty612
- Singh, R., Lanchantin, J., Robins, G., and Qi, Y. (2016). DeepChrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics* 32, i639–i648. doi: 10.1093/bioinformatics/btw427
- Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). “Going deeper with convolutions,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9.
- Xu, J., Xiang, L., Liu, Q., Gilmore, H., Wu, J., Tang, J., and Madabhushi, A. (2016). Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. *IEEE Trans. Med. Imaging* 35, 119–130. doi: 10.1109/TMI.2015.2458702
- Xu, T., Zhang, H., Huang, X., Zhang, S., and Metaxas, D. N. (2016). “Multimodal deep learning for cervical dysplasia diagnosis,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Boston, MA), 115–123.
- Yang, W., Liu, Q., Wang, S., Cui, Z., Chen, X., Chen, L., and Zhang, N. (2018). Down image recognition based on deep convolutional neural network. *Inform. Process. Agric.* 5, 246–252. doi: 10.1016/j.inpa.2018.01.004
- Zeng, K., Yu, J., Wang, R., Li, C., and Tao, D. (2017). Coupled deep autoencoder for single image super-resolution. *IEEE Trans. Cybernet.* 47, 27–37. doi: 10.1109/TCYB.2015.2501373
- Zhang, S., Zhou, J., Hu, H., Gong, H., Chen, L., Cheng, C., and Zeng, J. (2016). A deep learning framework for modeling structural features of RNA-binding protein targets. *Nucleic Acids Res.* 44:e32. doi: 10.1093/nar/gkv1025

**Conflict of Interest Statement:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2019 Tang, Pan, Yin and Khateeb. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.