

Machine learning in bioinformatics

Indrajeet Kumar, Surya Pratap Singh and Shivam

Graphic Era Hill University, Dehradun, India

26.1 Introduction

The field of study that enables computers to learn things without being programmed is explicitly referred to as the term machine learning. Machine learning was developed for the new capability of computers. The main aim of bringing machine learning into the market was just to reduce the man workforce and train the computers in such a way that tough and complicated tasks could be performed by the computers without the involvement of human beings. Today, we find that most of the well-known organizations use different machine learning algorithms for their reliability and better results. Some of the major examples of machine learning are data mining (prominently used in medical records), web-click data, recognition of handwriting, natural language processing, selfcustomizing of programs, and understanding the human learning. The machine learning algorithm is also used in gene findings, genome annotation, protein structure prediction, gene expression analysis, drug discovery, etc. (Bhatt, Kumar, Vijayakumar, Singh, & Kumar, 2020; TAIR, 2019). The overview of the artificial intelligence component is given in Fig. 26.1.

Machine learning algorithms are mainly of two main categories that are supervised machine learning and unsupervised machine learning (Bhatt et al., 2020). Fig. 26.2 is an overview of machine learning and its categories. A detailed description of supervised and unsupervised learning is given here.

26.2 Supervised learning

Supervised learning has been an important algorithm in machine learning ideology over the period. Undoubtedly, supervised learning has helped a lot to improve performance and provide better results. In this type of machine learning

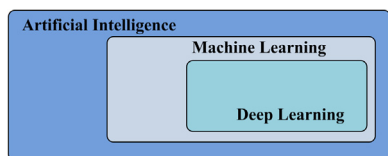


FIGURE 26.1 Overview of artificial intelligence components.

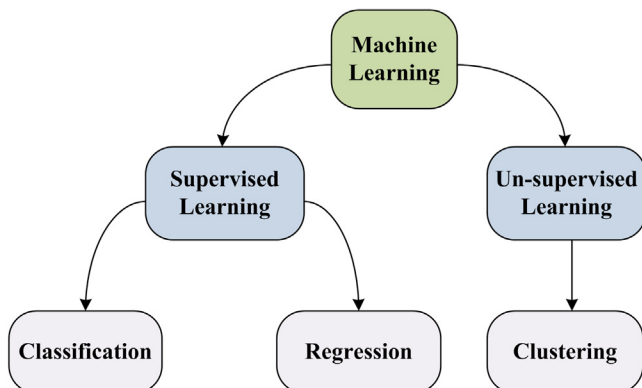


FIGURE 26.2 Overview of machine learning and its categories.

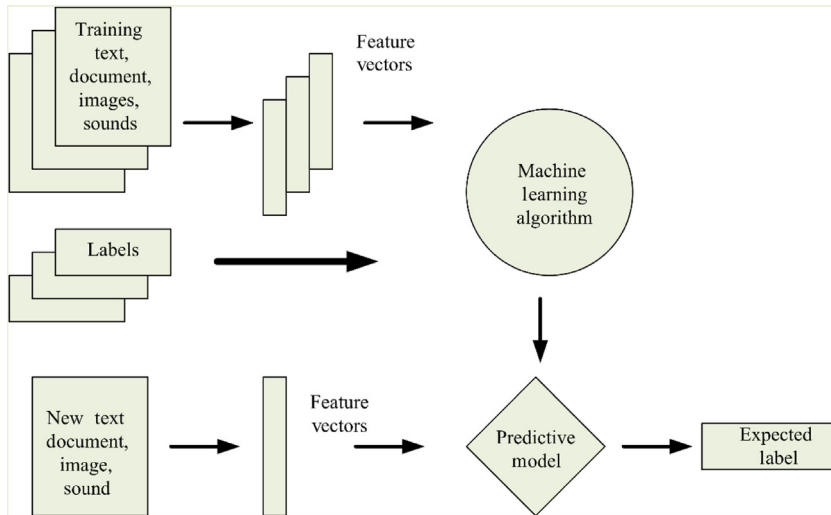


FIGURE 26.3 Flow diagram of a supervised machine learning algorithm.

algorithm, we are given a dataset and thus it is already known that how our output will look alike, coming across the idea and keeping in mind that there is a mapping between the input and the output. Fig. 26.3 shows the functionality of supervised machine learning (Caruana & Niculescu-Mizil, 2006) where the training set is passed with the respective label. From Fig. 26.3, it has also been observed that the test sample is directly passed to the trained model for testing purposes.

Supervised learning can be simply learned with a real-life example like, let us consider a father–son duo, who recently bought a new car. The son urges his father that he wants to learn how to drive the car, so in this case, the father trains the son first with already knowing what the results would look like once his son is trained properly. The same case goes with supervised learning where the machine is trained in such a way that our desired output can be achieved. House price prediction, breast cancer, liver cancer, brain cancer, etc., are a few of the important examples where the concept of supervised learning is applied. Describing supervised learning in a single line so supervised learning is the algorithm in which a computer is being taught how to learn something. Two major components of supervised learning are regression and classification. Let us understand the functionality of regression and classification separately.

26.2.1 Classification

As with regression, classification is also an important concept in machine learning under supervised learning algorithms. As the name itself suggests categorization, so classification is a widely used technique for prediction, as in classification, one can directly predict by categorizing the dataset into two categories, one for the positive cases whereas other for the negative cases and thus it makes the job less difficult for the prediction. To make it clearer, first, let us look at a smaller definition that can be used to bring out the methodology of classification in a précised manner or simpler words. Classification is a well known approach to label the given set of structured or unstructured data into a pre-defined category. Now, their categories can be in the form of true/false, 0/1, or positive/negative. Classification is mostly used in medical aspects as it makes it easy to understand whether the patient is diagnosed with any problem or not because there are no other means than positive or negative. Refer to the example below to make it clearer (Kumar, Bhadauria, Virmani, & Thakur, 2017a,b).

Suppose, you look for software to examine the bank account of each customer and for each account, predict whether the account has been hacked or compromised, so in this case classification is used. The reason to illustrate this is simple as to examine whether the account has been hacked or compromised, we have two cases. So here, the setting of value depends upon the user. For the case of the hacked account, the values can be set to 1 and for a compromised account, the value can be set to 0 which gives a clear view of whether the account is hacked or compromised. Another example of classification is the detection of emails whether they are spam or not.

26.2.2 Supervised machine learning in bioinformatics

From the last decades supervised learning is very frequently used in bioinformatics for protein structure characterization (Heffernan et al., 2015; Lyons et al., 2014), biomedical imaging (Alipanahi, DeLong, Weirauch, & Frey, 2015; Hua,

Hsu, Hidayati, Cheng, & Chen, 2015; Suk & Shen, 2013; Zhang et al., 2016), drug discovery (Mathé, Sagot, Schiex, & Rouzé, 2002), protein classification (Salzberg, 1995), anomaly classification (Hua et al., 2015; Suk & Shen, 2013), and so on. In this type of work, data set preparation and its bifurcation are very important. If the available dataset is a benchmark dataset and labeled by experts, then it is useful, and the task is pretty easy for the unlabeled dataset. The performance of each model is evaluated in the form of error rate, classification accuracy, individual class accuracy, receiver operating characteristic curve, sensitivity, and specificity (Kumar et al., 2017a,b). The model is validated using the two most widely used approaches, that is, cross-validation with K -fold and leave-one-out type cross-validation. The overall performance of the developed system depends on the extracted features. It has also been found that every extracted feature is not prominent for the problem, so the selection of the prominent features is a very crucial task. There are so many methods available for feature selection, such as genetic algorithm, nature-inspired optimization algorithm, principal component analysis, and linear discriminant analysis.

After the selection of prominent features, the next step is to select a classification model. There are so many frequently used classifiers, such as Naïve Bayes classifier (Minsky, 1961), decision tree (Breiman, Friedman, Stone, & Olshen, 1984), random forest (Breiman et al., 1984; Vapnik, 2013), Adaboost classifier (Breiman et al., 1984; Vapnik, 2013), support vector machine (SVM) (Kumar et al., 2017a,b), neural network (Kumar et al., 2017a,b), probabilistic neural network (Kumar et al., 2017a,b), and k -nearest neighbor (k -NN) classifier (Kumar et al., 2017a,b). The brief details regarding classifiers in bioinformatics are also found in the article (Breiman et al., 1984; Minsky, 1961; Vapnik, 2013; Kumar et al., 2017a,b).

The regression and classification go hand in hand and each technique has its own importance and is the most important technique of supervised learning. Furthermore, let us look at the second type of machine learning algorithm that is unsupervised learning in detail.

26.3 Unsupervised machine learning

Unsupervised machine learning is the second type of machine learning algorithm after supervised learning in machine learning that allows addressing problems or situations with little idea or sometimes even no idea about how the results will look like (Carter, Dubchak, & Holbrook, 2001; Ghahramani, 2003). Under unsupervised learning, the deriving of the feedbacks that are based on the results of predictions made is not present. Fig. 26.4 shows the working of supervised machine learning.

To understand unsupervised learning more simply, let us look at the same example of a father and a son that was considered in the above section, that is, of supervised machine learning. Taking the case of unsupervised learning into the consideration, when the son urges that he wants to learn how to drive the car, he is given the car without any training and with having no idea of what the actual result will be, whereas in supervised learning, proper training was given, and the results were already known. This is how supervised learning and unsupervised learning differ from each other. Below is a problem discussed that will give a better view for the machine learning algorithms when taken into the

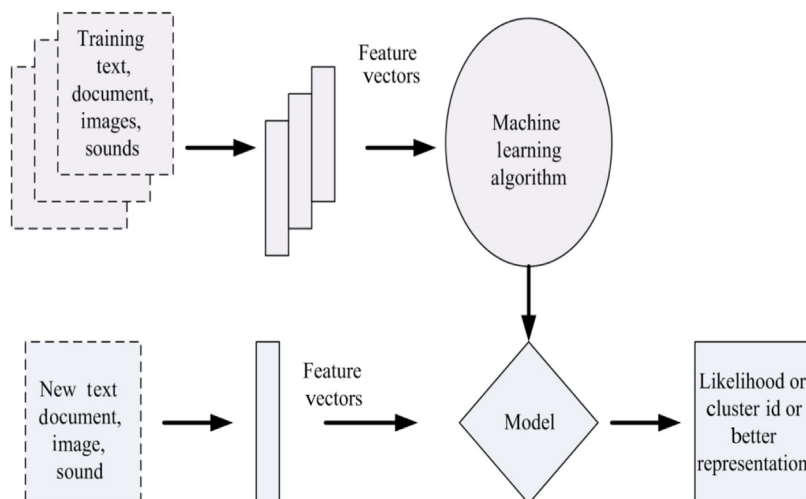


FIGURE 26.4 Flow diagram of the unsupervised machine learning algorithm.

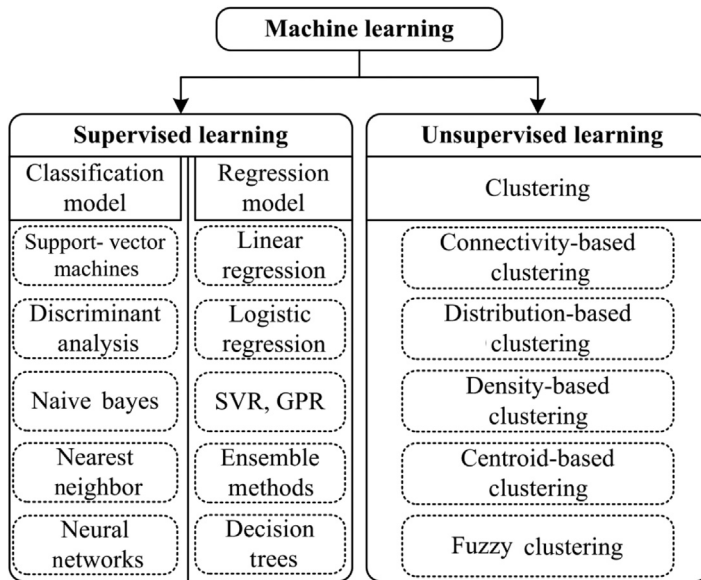


FIGURE 26.5 Classification of the machine learning algorithm.

consideration together, whether which algorithm should be used as one of the most difficult tasks in machine learning is choosing the right algorithm at the right time (Fig. 26.5).

26.4 Problems to understand supervised learning and unsupervised learning

From the given set of following examples, which algorithm will be used for:

1. Examining given emails whether they are spam or not spam.
2. Given a set of news reports that are found on the web page, further grouping them into a set of articles about the same story.
3. Given a database of customer's data, further automatically discover insurance segments and group the customers into different insurance segments.
4. Given a dataset of patients that are diagnosed as either being COVID positive or not.

Thus let us look at each of the cases individually and see which algorithm suits best for their prediction and why? Looking at Case I, so this is an example of supervised learning (classification) as here, the data can be easily categorized into two different cases and assign 0 to nonspam and 1 to spam. Now, when taken the Case II into the consideration, so this is an example of unsupervised learning as the input is although of the same category but is different from each other. Case III is also an example of unsupervised learning and is similar to Case II. The last case, that is, Case IV is an example of supervised learning (classification) and is similar to Case I. By now, it is clear that how supervised and unsupervised learning differ from each other. From the above paragraph, cases refer to the four examples that were used in the unsupervised section that depicted which algorithm should be used for the following cases. Thus Case I, Case II, Case III, and Case IV refers to the examples a, b, c, and d, respectively, used in the unsupervised learning section.

26.5 Regression

Regression is an important concept of supervised learning algorithms. It has been found that regression is used when we have many predictions possible and thus to get the best results of the prediction, we use regression. An example will make this clearer but before looking at the example first, let us look at a proper definition for the regression methodology. When predictions of the results within a continuous function or rather try to map the input variables to some particular continuous function, regression is used (Qasim, Amin, & Omer, 2020).

Suppose a manufacturer has a large inventory of identical items and wants to predict that how many out of those items he will be able to sell over the next 3 months, so in this case regression is used. The reason to illustrate this problem is that if the manufacturer has thousands of items, so when considering all the items together at once then this can

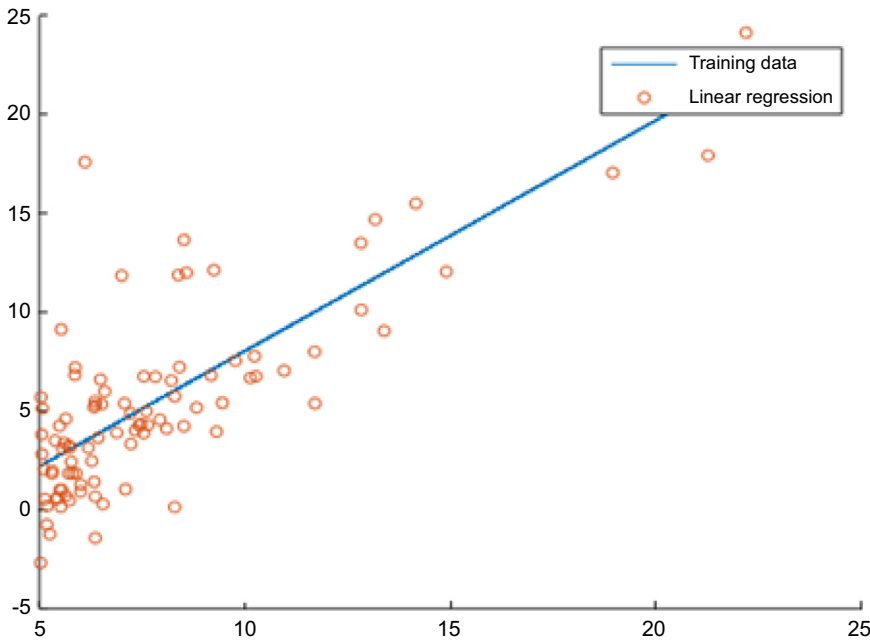


FIGURE 26.6 Resemblance of linear regression.

be considered as real values or continuous values. Regression is mostly used in predictions like house pricing. Now, let us look at another approach involved in supervised learning that is classification.

Now, let us look at linear regression with one variable and understand the study of main units that are involved in it. But before that, the question is what is linear regression? In machine learning, linear regression is an algorithm having a constant slope and the results that are predicted are continuous in nature. They are mainly used for the prediction of house prices, product sales, market beneficiaries, etc., but do not involve the processes of categorization like classifying into spam/spam, true/false, and so on. Fig. 26.6 shows the resemblance of linear regression. It indicates how the training examples are fit to generate a graph along with the values when linear regression is applied. A few of the main units that are involved in linear regression with one variable are hypothesis, cost function, and gradient descent. Let us look at each one of them separately.

26.5.1 Hypothesis

Hypothesis is a function that takes the input x and then results in the estimated value of y (Haahtela, 2019) as shown in Fig. 26.7. Kindly refer to Eq. (26.1) to see how a hypothesis is represented.

$$HQ(x) = Q_0 + Q_1x \quad (26.1)$$

where Q is the dataset ($Q = 0, 1, 2, \dots, n$), x is the input variable, and H denotes hypothesis function.

26.5.1.1 Cost function

Cost function is another important component/unit that is used for the training of the dataset. It is denoted by J . By using the cost function, the accuracy of the hypothesis function can be measured. The other name that can be used to denote cost function is the squared error cost function and is used for most regression problems (Oymak, Dulek, & Gezici, 2020). The mathematical expression of the cost function is given in Eq. (26.2) and its practical implementation is given in Eq. (26.3).

$$J = \frac{1}{2m} \sum_{i=1}^m [HQ(x_i - y_i)]^2 \quad (26.2)$$

$$J = \frac{1}{2m} \times \text{sum}((h-y)^2) \quad (26.3)$$

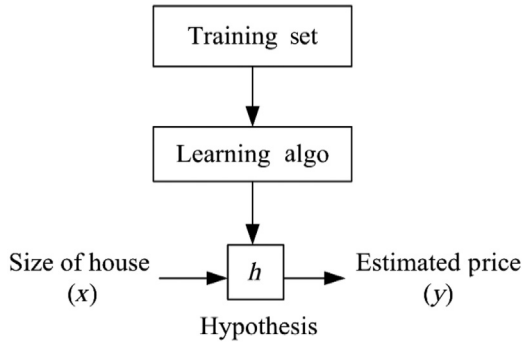


FIGURE 26.7 Hypothesis function.

From the above two equations, that is, Eqs. (26.2) and (26.3), J denotes the cost function whereas x and y are the values along the x -axis and y -axis, respectively, and H is a hypothesis, Q is the dataset, m is the number of the training examples, $HQ (x_i - y_i)^2$ or $(h - y)^2$ is the minimized form of $J(Q_1)$. Here, Q can be a single dataset as well as multiple datasets ($Q = 1, 2, 3, \dots, n$).

An important factor to keep in mind is that in linear regression, the cost function is always a bowl-shaped function that is also known as a convex function. Some of the well-known biological examples involved in machine learning are data analysis of high-throughput microarray, prediction of protein functions, prediction of enzyme functions, understanding of the makers diseases, etc. Now, let us look at the next component involved in linear regression that is gradient descent. Regression model can be helpful in finding the effect of smoking on cancer or survival behavior of patients in ICU. A similar type of problem can be also handled by the regression model in bio-medical science.

26.5.1.2 Gradient descent

Another important component present in linear regression is gradient descent and is denoted by delta Δ . Gradient descent starts with the same Q_0, Q_1 (say $Q_0 = 0, Q_1 = 0$) and keeps changes the Q_0, Q_1 to reduce the $J(Q_0, Q_1)$ until the minimum state is reached. Gradient descent is referred to as simultaneous update and can converge to a local minimum even if the learning rate α is fixed. The practical expression for gradient descent is given in Eq. (26.4).

$$\text{delta} = \frac{1}{m} \times (x' \times x \times \text{theta} - x' \times y) \quad (26.4)$$

26.5.1.3 Logistic regression algorithm

Though the name has regression involved in it, still logistic regression algorithm is a part of classification under supervised learning (Rezapour, Molan, & Ksaibati, 2020). Logistic regression algorithm over time has been the most frequent algorithm used in machine learning, always producing output between 0 and 1 and is expressed in Eq. (26.5). In the below equations, H is a hypothesis, Q is the dataset where Q can be a single dataset as well as multiple datasets, x is the predicted area, g is the sigmoid function, T denotes the transpose of the training dataset Q , $HQ(x)$ denotes the estimated probability that $y = 1$ on input x , e is exponential, and z is a particular range.

$$0 \leq HQ(x) \leq 1 \quad (26.5)$$

The hypothesis representation of logistic regression algorithm is discussed below:

$$HQ(x) = g(Q^T x) \quad (26.6)$$

$$g(z) = \frac{1}{1 + e^z} \quad (26.7)$$

On combining Eqs. (26.6) and (26.7), we get,

$$HQ(x) = \frac{1}{1 + e^{-Q^T x}} \quad (26.8)$$

The sigmoid function is also called a logistic function and a practical representation of the sigmoid function is given in Eq. (26.9).

$$g = 1./(1.0 + \exp(-z)) \quad (26.9)$$

Here, g is denoted as sigmoid function and above is the formula that is used to find the sigmoid function over a dataset. The main significance of the sigmoid function or g is that its range lies between 0 and 1 and thus is used only for the models in which the probability is predicted as an output.

Over a period of time, it has been found that overfitting has been a big problem seen in the training of the datasets. In technical terms, if we have too many features, the learned hypothesis may fit the training set very well but might fail to generate new examples. This is termed as the problem of overfitting. In simpler words, it can be said that overfit can be used to make accurate predictions for the examples that are in the training set, but it may fail to generalize to make predictions accurately on new, and on previous examples that are unseen. Well, to overcome the problem of overfitting, there are several options to address this problem and are discussed below.

Option 1: Reduce the number of features. Under this addressing, the trainer manually selects the features to keep and involves the model selection algorithm. The model selection algorithm is an algorithm that automatically decides whether which features need to be kept and which not. Though this is a preferred approach, still has a big disadvantage and that is, throwing away some features may result in throwing away some information. Another option that is addressed to overcome the problem of overfitting is discussed next.

Option 2: Regularization. Another solution used for the problem of overfitting is regularization that keeps all the features but reduces the magnitudes/values of the parameters Q_j . The important point to keep in mind about regularization is that when the number of feature more, regularization works well and each of them contributes to the prediction of y . The mathematical expression for the cost function of regularization is discussed in Eq. (26.10).

$$J = \frac{1}{2m} \left\{ \sum_{i=1}^m [HQ(x_i - y_i)]^2 + \lambda \sum_{j=1}^n \cdot Q_j^2 \right\} \quad (26.10)$$

where the practical expression of cost function and gradient descent for regularization is given in Eqs. (26.11) and (26.12), respectively. In the below equation, sum, m , and theta are the essential components that are used to determine cost function and gradient descent:

$$J = \frac{1}{m} \times \text{sum}(-y' \log(a_theta) - (1 - y)' \times \log(1 - a_theta)) + XX \quad (26.11)$$

$$XX = \frac{\text{lamda}}{2 \times m} \times \text{sum}(\text{theta}(2:\text{length}(\text{theta})) \times \text{theta}(2:\text{length}(\text{theta})))$$

$$\text{grad} = \frac{1}{m} \times \text{sum}(X \times \text{repmat}(\text{sigmoid}(X \times \text{theta}) - y, 1, \text{size}(X, 2))) \quad (26.12)$$

and the gradient descent for regularized linear regression is given in Eq. (26.13).

$$\begin{aligned} \text{grad}(:, 2:\text{length}(\text{grad})) &= \text{grad}(:, 2:\text{length}(\text{grad})) + YY' \\ YY &= (\text{lambda}/m) \times \text{theta}(2:\text{length}(\text{theta})) \end{aligned} \quad (26.13)$$

From the above given equations, lambda is the average number of events per intervals, gradient descent is denoted as grad, length(grad) denotes the entire length of the gradient descent, 2 is the starting range for training of examples, sigmoid is the activation function, m is the number of the training examples, y is the value on Y -axis, X is the predicted probability as an output when $y = 1$, Y is the predicted probability as an output when $x = 1$, repmat is a function that is used to return an array that contains n number of copies of grad in a row and column dimensions, and theta is the practical term used for the features of the dataset.

Now, let us look at another important methodology that comes under machine learning, that is, the theory and modulation of neural networks. The neural network plays an important role in making machine learning a better source. First, let us understand what neural networks actually means and then focus on the practical expressions of cost function and sigmoid gradient descent used in neural networks. A neural network is considered as giving the brain to the machines to increase the performance and produce better results on its own. A neural network is used in unsupervised learning that was designed as a human brain to the machines. A neural network comprises of multiple layers/nodes and each node is responsible for a particular computation in simpler terms. Suppose there are two layers, so each unit of layer 1 is connected to layer 2 as shown in Fig. 26.8.

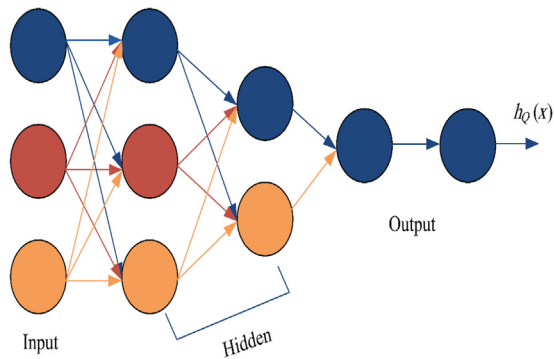


FIGURE 26.8 Architecture of neural network.

From Fig. 26.8, 1st layer is always the input layer that takes the input, the last layer is always the output layer that produces the output, and the rest of the layers are called hidden layers. Here, $h_Q(x)$ refers to the number of features used for output prediction.

In many manuscripts, both the terms $H_Q(x)$ and $h_Q(x)$ are used so $H_Q(x)$ is referred to as probability distribution whereas $h_Q(x)$ is referred to as hypothesis representation.

26.6 Clustering

An algorithm that is used to find the clusters is called clustering. The applications of clustering algorithms are social networking analysis, market segmentation, organizing computing clusters, and analysis of astronomical data (Understand galaxy/space formation). The types of clustering algorithm are given as:

1. K-means clustering;
2. density-based clustering;
3. distribution-based clustering; and
4. fuzzy clustering.

Choosing the right number of clusters has always been an important factor to watch out for, but before that let us focus on the K-means algorithm that plays a vital role in unsupervised learning.

26.6.1 K-means clustering

Over a period of time, the K-means algorithm (Windarto et al., 2019) has been the most popular and widely used clustering algorithm. In this algorithm, two points are initialized at random that are called as cluster centroid. Suppose a training example has two points, out of which one is red whereas the other is blue, so when the training examples are more in number, then both points come across each other that helps to easily differentiate the examples based on the colors. The two major steps involved in this algorithm are

1. clustering assignment step and
2. loop centroid step.

The above stanza discussed is the clarification of step 1 that is the clustering assignment step. Now let us look at step 2 that is the loop centroid step. For loop centroid, the procedure is the same as the clustering assignment step. The only factor that makes these two steps different from each other is that in clustering assignment steps, points can be placed anywhere whereas, in the loop centroid step, the points are always at the middle (Fig. 26.9).

26.6.2 Density-based clustering

Density-based clustering model uses the concept of data point density to create a cluster. It segregates multiple regions having a different density within the data space of the parent cluster. The frequently used density models are density-based spatial clustering of applications with noise and ordering points to identify the clustering structure.

The density-based clustering model is suitable when the dataset contains noise and some outliers. In this approach, randomly a point has been picked and the picked point does not belong to any cluster or it is assigned as an outlier.

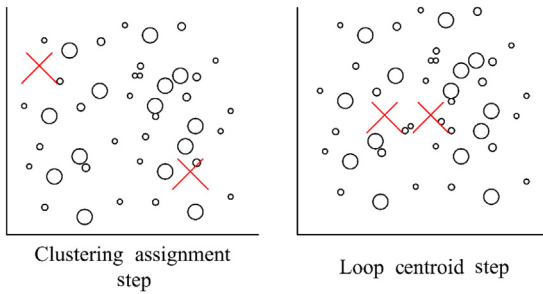


FIGURE 26.9 Clustering assignment step and loop centroid step.

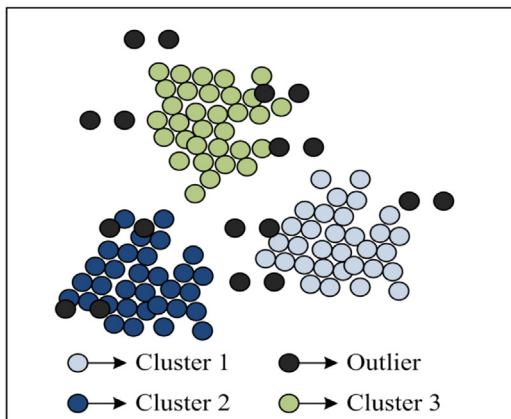


FIGURE 26.10 Representation of density-based clustering.

After that, calculate the neighborhood to find out the selected point is the core point or not. If yes, then start clustering around the selected point, and if not then labeled as an outlier. After this, expand the cluster by adding all directly accessible points to the cluster. Now carry out “neighborhood-jumps” to discover all density-accessible points and add them to the cluster. If an outlier is added, modify that point’s type from outlier to border point. The similar steps are repeated until all points are either assigned to a cluster or elected as an outlier. Fig. 26.10 shows the representation of the density-based clustering. In Fig. 26.10, three different clusters are defined, and black color points are referred to as outlier.

26.6.3 Distribution-based clustering

Distribution-based clustering is suitable when the data consists of distributions like Gaussian distributions, probability distributions, and so on. Fig. 26.11 shows the three Gaussian distribution-based clustering.

It has been observed that for three clusters, three different clustering centers are used and each one is based on Gaussian distribution. It can also be seen as the distance increases from the centroid the probability of that belongingness decreases. According to this principle, distribution-based clustering is implemented in biomedical science for discrimination between normal and infected tissue clustering, such as brain hemorrhage detection, breast abnormalities segmentation, and brain tumor segmentation.

26.6.4 Fuzzy clustering

Fuzzy clustering operates similarly to the K-means clustering method. It allows a cluster to be partially more than one cluster but in the case of K-mean, only one cluster is allowed. Let us assume that a cluster K . Suppose we have K clusters and a set of variables like $m_{i1}, m_{i2}, \dots, m_{ik}$, represents the probability value for object i , which is classified into k cluster where m_{ik} represents the membership value. The value of m_{ik} varies between 0 and 1 and the aggregation or the sum of all the values is equal to always 1. Fuzzy clustering is suitable in biomedical engineering, such as the visualization of internal tissues, organs, and ailment diagnosis of tumor and heart and vascular-related disease, adequate scheduling of treatment and surgical preparation, and image registration.

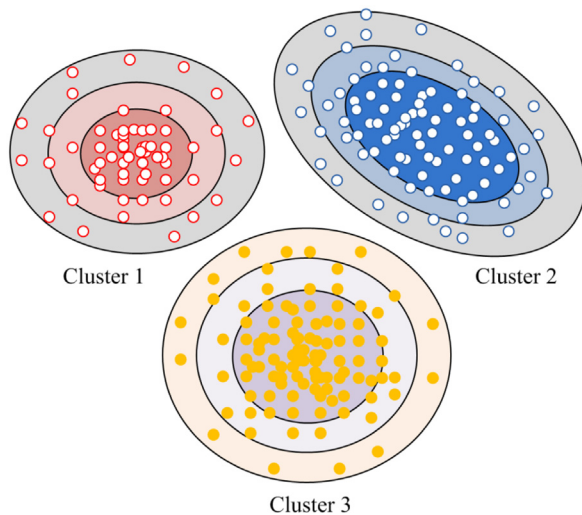


FIGURE 26.11 Representation of distribution-based clustering.

26.7 Unsupervised learning in bioinformatics

In bioinformatics, microarray data can be analyzed by taking the help of unsupervised machine learning methods. Among the various methods, clustering is widely acceptable for gene expression analysis. Initially, [Sheng, Moreau, De Smet, Marchal, and De Moor \(2005\)](#) observed that the clustering algorithm (K-mean) is suitable for finding the first generation of gene expression. It has been also found that the few challenging tasks like cluster number identification, outlier's identifications, or computational complexity cannot be obtained with the help of a clustering algorithm. Therefore the second generation of clustering methods, such as biclustering algorithms ([Sheng, Moreau, & De Moor, 2003](#)) and selforganizing tree algorithms ([Herrero, Valencia, & Dopazo, 2001](#)), can handle the previously mentioned problems.

26.8 Application of machine learning

The application of machine learning is not limited to only image segmentation, object classification, clustering, and regressions. It is also applicable in bioinformatics, such as gene findings and genome annotation, protein structure prediction, gene expression analysis, complex interaction modeling in biological systems, and drug discovery ([Shen, Zhang, Han, & Huang, 2020](#)).

26.8.1 Genome annotation

In recent research scenario, the discovery of the genomic sequences that comprised of functional regions is the initial stride in genome annotation. Machine learning approach is used to detect significant features, such as micro-RNAs, protein-coding genes, and long noncoding RNAs, polyadenylation sites, DNAs I hypersensitive sites, chromatin states, and *cis*-regulatory elements. The characterization of these identities is potentially classified by using machine learning methods. The performance of the used model depends on the heterogeneity of the dataset. In past studies, SVM, k-NN, an artificial neural network, decision tree, Bayesian classifier, and convolutional neural network (CNN) were used for such type of task and it has also been found that the SVM with different kernel functions shown promising results for task related to protein classifications ([Ding & Kihara, 2019](#); [Ying & Lin, 2020](#); [Cuevas et al., 2019](#); [Sperschneider, 2019](#); [Peters, Sinecen, Kizilkaya, & Thomas, 2020](#); [Rodgers-Melnick, Culp, & DiFazio, 2013](#)).

26.8.2 Protein structure prediction

Machine learning and deep learning algorithms are prominently used in bioinformatics, such as sequencing data. In this, DNA sequence, RNA sequence, and DNase sequence, are frequently predicted by taking the help of the machine and deep learning models by using feature extractions from genomic sequence ([Baldi, Brunak, Frasconi, Soda, & Pollastri, 1999](#); [Hochreiter, Heusel, & Obermayer, 2007](#); [Min, Lee, & Yoon, 2017](#)). The secondary structure (SS) of protein estimation is also done with the help of deep learning models in a very effective manner. For this task three important features (the amino acid residues themselves, the PSSM information, and the Atchley factors) were used and

TABLE 26.1 List of tools and machine learning algorithms in bioinformatics.

Problem objective	Machine learning models	Prominent-considerable features	References
Gene identification and expression	SVM, hidden Markov model, random forest, CNN	Genomic sequences, mapped RNA-seq transcripts, orthologous sequences, sequence signals, sequence compositions, sequence length, sequence segments, presence/absence of motifs, and motif pairs	Ding and Kihara (2019) and Ying and Lin (2020)
Protein–protein interaction	SVM, hidden Markov model, random forest	Protein subcellular localization, expression patterns, features derived from protein structure, such as conserved interaction sites and hydrophobicity	Ding and Kihara (2019)
Gene ontology	CNN, decision trees, k-NN, Naïve Bayes	Gene expression, predicted secondary structure, homology, membership in enzyme families, interacting proteins	Sperschneider (2019) and Rodgers-Melnick et al. (2013)
Genomic prediction	SVM, probabilistic neural network	Presence/absence of genetic makers and pairs of genetic makers, direct genetic makers for PNN	Peters et al. (2020) and Cuevas et al. (2019)

CNN, convolutional neural network; k-NN, k-nearest neighbor; SVM, support vector machine; PNN, probabilistic neural network.

the deep models were trained by using the principle of residual window training. After multiple iterations of training, an efficient model was observed and yields an accuracy of 80% (Spencer, Eickholt, & Cheng, 2014). The three-dimensional structure predictions are also crucial for experts and such type of task can be also performed effectively by deep learning models like recurrent neural network. To perform such type of task, upstream and downstream information plays a significant role in accounting for the prediction of 3-D structure. A previously published article also reports the important input information, such as RNA-seq, DNA-seq, DNase-seq, and ChIP-seq, for performing protein structure prediction. The list of tools and the important algorithms under machine learning that has been discussed in previously published studies are listed in Table 26.1.

26.8.3 Research area in bioinformatics with deep learning

In the current age of research, machine learning is further driven or précised and called deep learning. So many deep learning models are available, such as alexnet (Lu, Lu, & Zhang, 2019), resnet-50, resnet-101, mobilenet, googlenet (Anand, Shanthi, Nitish, & Lakshman, 2020; Wen, Li, & Gao, 2019), and VGG-16 (Theckedath & Sedamkar, 2020). Each model is completely based on the principle of deep neural network and CNN (Bhatt et al., 2020). CNN multilayer neural network is used and at each layer convolutional operations have been performed for the feature extraction. The structure of the CNN model is given in Fig. 26.12.

The CNN model is comprised of the input layer, convolutional layer, fully connected layer, and output layer. It has more than one hidden layer; therefore it is also called a deep neural network. The combination of convolutional layers is used for low-level and high-level feature extraction and extracted features are self-learned by the model. A fixed-size filter and nonlinear activation function are applied to the input image to get a feature map for each convolutional layer. The feature map is passed to the next convolutional layer by feature pooling. A similar type of task is used for each convolutional layer. The filter is also known as kernel window and the frequently used nonlinear activation function is “ReLU,” “sigmoid,” “tanh,” “leaky ReLU,” etc. Before the output layer, one important layer is used called the fully connected layer or softmax layer. Its main use is to convert the output of the ending layer of the proposed model into an essential probability distribution. We use the softmax function so that our output layers can communicate with each other and are aware of the result.

26.9 Discussion

After undergoing the deep studying of literature, it has been observed that there are still allots of works that have to be done in the field of bioinformatics using machine learning. If the proposed model performs a little bit better than the previously published model, then it would be a huge achievement in the field of bioinformatics. In the near future, machine learning, and deep learning will play an important role in other problems in bioinformatics as a prediction of

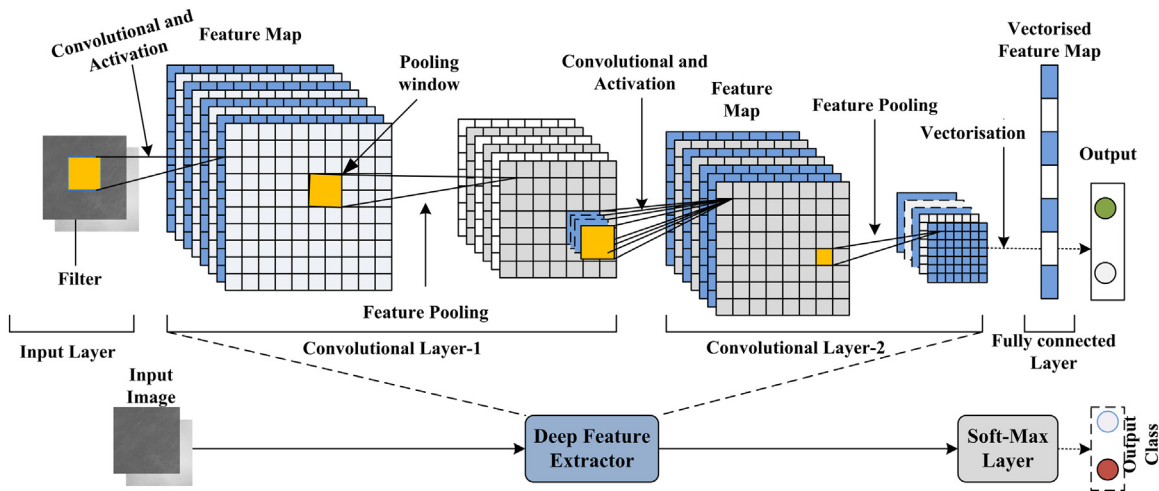


FIGURE 26.12 Convolutional neural network architecture.

beta-sheet, DNA boundaries extraction, DNA promoter area extraction, RNA SS, protein functional domain extraction, and so on. In the same domain, the proper selection of a deep learning model is a challenging task. Therefore there is a need for development in the field of multimodal deep learning model development. To implement a deep learning model effectively, a few challenges must be faced like limited dataset, converting the black-box model into a white-box model, selection of a well deep learning model that is appropriate, selection of appropriate hyperparameters, training, and testing dataset preparation, data cleaning, estimation of error, and some regulatory parameters.

26.10 Conclusion

This article provides an introduction to machine learning from the beginning. It is also pointed out that the machine learning subset of artificial intelligence and in the same fashion machine learning is extended to deep learning for better performance of the developed system. Initially, an introduction to machine learning is described, and later two important types of supervised, and unsupervised machine learning are discussed, and their different variations are also provided. The applications and benefits of machine learning models are also discussed. In previous studies, it has also been observed that most of the studies were based on the classification problem or regression problem. But the machine learning model is not limited to only these applications. It can also be suitable for the long noncoding RNAs, protein-coding genes, polyadenylation sites, micro-RNAs, DNAs, I hypersensitive sites, chromatin states, and *cis*-regulatory elements, and using the significant feature extraction approaches. At the end of the manuscript, some applications of bioinformatics with suitable machine learning models and a list of important features for the respective problems are briefly described. A similar type of problem can also be solved with the help of deep learning models and get better results with respect to a machine learning-based approach.

Conflict of interest

The authors declare that they have no conflict of interest

References

- Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology*, 33(8), 831–838.
- Anand, R., Shanthi, T., Nitiash, M. S., & Lakshman, S. (2020). *Face recognition and classification using GoogleNET architecture. Soft computing for problem solving* (pp. 261–269). Singapore: Springer.
- Baldi, P., Brunak, S., Frasconi, P., Soda, G., & Pollastri, G. (1999). Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics (Oxford, England)*, 15(11), 937–946.
- Bhatt, C., Kumar, I., Vijayakumar, V., Singh, K. U., & Kumar, A. (2020). The state of the art of deep learning models in medical science and their challenges. *Multimedia Systems*, 27, 599–613.

- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC Press.
- Carter, R. J., Dubchak, I., & Holbrook, S. R. (2001). A computational approach to identify genes for functional RNAs in genomic sequences. *Nucleic Acids Research*, 29(19), 3928–3938.
- Caruana, R. & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the twenty-third international conference on machine learning* (pp. 161–168).
- Cuevas, J., Montesinos-López, O., Juliana, P., Guzmán, C., Pérez-Rodríguez, P., González-Bucio, J., & Crossa, J. (2019). Deep kernel for genomic and near infrared predictions in multi-environment breeding trials. *G3: Genes, Genomes, Genetics*, 9(9), 2913–2924.
- Ding, Z., & Kihara, D. (2019). Computational identification of protein-protein interactions in model plant proteomes. *Scientific Reports*, 9(1), 1–13.
- Ghahramani, Z. (2003). *Unsupervised learning. Summer school on machine learning* (pp. 72–112). Berlin, Heidelberg: Springer.
- Haahtela, T. (2019). A biodiversity hypothesis. *Allergy*, 74(8), 1445–1456.
- Heffernan, R., Paliwal, K., Lyons, J., Dehzangi, A., Sharma, A., Wang, J., & Zhou, Y. (2015). Improving prediction of secondary structure, local backbone angles and solvent accessible surface area of proteins by iterative deep learning. *Scientific Reports*, 5(1), 1–11.
- Herrero, J., Valencia, A., & Dopazo, J. (2001). A hierarchical un-supervised growing neural network for clustering gene expression patterns. *Bioinformatics (Oxford, England)*, 17(2), 126–136.
- Hochreiter, S., Heusel, M., & Obermayer, K. (2007). Fast model-based protein homology detection without alignment. *Bioinformatics (Oxford, England)*, 23(14), 1728–1736.
- Hua, K. L., Hsu, C. H., Hidayati, S. C., Cheng, W. H., & Chen, Y. J. (2015). Computer-aided classification of lung nodules on computed tomography images via deep learning technique. *Oncotargets and Therapy*, 8.
- Kumar, I., Bhadauria, H. S., Virmani, J., & Thakur, S. (2017a). A classification framework for prediction of breast density using an ensemble of neural network classifiers. *Biocybernetics and Biomedical Engineering*, 37(1), 217–228.
- Kumar, I., Bhadauria, H. S., Virmani, J., & Thakur, S. (2017b). A hybrid hierarchical framework for classification of breast density using digitized film screen mammograms. *Multimedia Tools and Applications*, 76(18), 18789–18813.
- Lu, S., Lu, Z., & Zhang, Y. D. (2019). Pathological brain detection based on AlexNet and transfer learning. *Journal of computational science*, 30, 41–47.
- Lyons, J., Dehzangi, A., Heffernan, R., Sharma, A., Paliwal, K., Sattar, A., & Yang, Y. (2014). Predicting backbone C α angles and dihedrals from protein sequences by stacked sparse auto-encoder deep neural network. *Journal of Computational Chemistry*, 35(28), 2040–2046.
- Mathé, C., Sagot, M. F., Schiex, T., & Rouzé, P. (2002). Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Research*, 30(19), 4103–4117.
- Min, S., Lee, B., & Yoon, S. (2017). Deep learning in bioinformatics. *Briefings in Bioinformatics*, 18(5), 851–869.
- Minsky, M. (1961). Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1), 8–30.
- Oymak, B., Dulek, B., & Gezici, S. (2020). Sensor selection and design for binary hypothesis testing in the presence of a cost constraint. *IEEE Transactions on Signal and Information Processing over Networks*, 6, 617–632.
- Peters, S. O., Sinecen, M., Kizilkaya, K., & Thomas, M. G. (2020). Genomic prediction with different heritability, QTL, and SNP panel scenarios using artificial neural network. *IEEE Access*, 8, 147995–148006.
- Qasim, M., Amin, M., & Omer, T. (2020). Performance of some new Liu parameters for the linear regression model. *Communications in Statistics-Theory and Methods*, 49(17), 4178–4196.
- Rezapour, M., Molan, A. M., & Ksaibati, K. (2020). Analyzing injury severity of motorcycle at-fault crashes using Machine Learning techniques, decision tree and logistic regression models. *International Journal of Transportation Science and Technology*, 9(2), 89–99.
- Rodgers-Melnick, E., Culp, M., & DiFazio, S. P. (2013). Predicting whole genome protein interaction networks from primary sequence data in model and non-model organisms using ENTS. *BMC Genomics*, 14(1), 608.
- Salzberg, S. (1995). Locating protein coding regions in human DNA using a decision tree algorithm. *Journal of Computational Biology*, 2(3), 473–485.
- Shen, Z., Zhang, Q., Han, K., & Huang, D.S. (2020). A deep learning model for RNA-protein binding preference prediction based on hierarchical LSTM and attention network. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.
- Sheng, Q., Moreau, Y., & De Moor, B. (2003). Biclustering microarray data by Gibbs sampling. *Bioinformatics (Oxford, England)*, 19(suppl_2), ii196–ii205.
- Sheng, Q., Moreau, Y., De Smet, F., Marchal, K., and De Moor, B. (2005). Advances in cluster analysis of microarray data. *Data analysis and visualization in genomics and proteomics*, 153–173.
- Spencer, M., Eickholt, J., & Cheng, J. (2014). A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(1), 103–112.
- Sperschneider, J. (2019). Machine Learning in plant–pathogen interactions: Empowering biological predictions from field scale to genome scale. *New Phytologist*, 228, 35–41.
- Suk, H. I. & Shen, D. (2013, September). Deep learning-based feature representation for AD/MCI classification. In *Proceedings of the international conference on medical image computing and computer-assisted intervention* (pp. 583–590). Berlin/Heidelberg: Springer.
- TAIR. (2019). The Arabidopsis information resource. <https://www.arabidopsis.org/portals/genAnnotation/genome_snapshot.jsp>. Accessed 30.09.20.
- Theckedath, D., & Sedamkar, R. R. (2020). Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. *SN Computer Science*, 1(2), 1–7.

- Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media.
- Wen, L., Li, X., & Gao, L. (2019). A transfer convolutional neural network for fault diagnosis based on ResNet-50. *Neural Computing and Applications*, 32, 6111–6124.
- Windarto, A. P., Siregar, M. N. H., Suharso, W., Fachri, B., Supriyatna, A., Carolina, I., & Toresa, D. (2019). Analysis of the K-means algorithm on clean water customers based on the province. *Journal of Physics: Conference Series*, 1255(1), 012001.
- Ying, K. C., & Lin, S. W. (2020). Maximizing cohesion and separation for detecting protein functional modules in protein-protein interaction networks. *PLoS One*, 15(10), e0240628.
- Zhang, S., Zhou, J., Hu, H., Gong, H., Chen, L., Cheng, C., & Zeng, J. (2016). A deep learning framework for modeling structural features of RNA-binding protein targets. *Nucleic Acids Research*, 44(4), e32-e32.