Proceedings of the 3rd International Conference on Biological Engineering and Medical Science
DOI: 10.54254/2753-8818/20/20230753

Multi-layered perceptron and its applications in biotechnology

Haiqiao Zhu

Shanghai XiWai International School, SISU., 1100 Wenxiang Road, Songjiang District, Shanghai, China

1811000230@mail.sit.edu.cn

Abstract. Multi-layered perceptron (MLP) is the first artificial neural network with a complete structure, which is mainly used to perform the tasks of pattern classification and function regression. Its original idea was inspired by biological neural networks in animal brains. Based on the process of electrical signals traveling through biological neural networks, this similar structure was designed to receive, process, and transmit data just like the brain. Multi-layered perceptron uses a feedforward path to complete the prediction task and backpropagation to train itself and optimize its performance. Developing until now, artificial neural network pioneered by multi-layered perceptron has been closely related to our life, and many more advanced derivatives that are good at solving more complex problems have emerged. Although the development of multi-layered perceptrons belongs to artificial intelligence and machine learning, its applications can be helpful to researchers in diverse fields such as engineering, finance, and medicine. This paper will focus on multi-layered perceptron, introduce its developing history, network structure, and algorithm (mainly learning algorithm), and briefly discuss its application in the specific field of biotechnology.

Keywords: artificial neural network, multi-layered perception, prediction.

1. Introduction

Multi-layered perceptron (MLP), a simple and early invented type of artificial neural network, is commonly used to complete tasks such as pattern classification, function approximation and prediction, and regression [1]. It is designed to mimic the structure and functioning of the biological neural network, which is the complex system in the brain that enable animals to process and respond to information. In biological neural networks, each neuron has multiple dendrites extending from its soma, for receiving the signals from the presynaptic neurons. There is a firing mechanism in the soma where the electrical signals accumulate until reaching a threshold to be emitted as an action potential, along the axon to the next neuron. Neurons connect to each other through the synapse, allowing the signals to traverse the network. The postsynaptic potentials fall into two categories, excitatory postsynaptic potentials (EPSPs) which bring the membrane potential towards the threshold, and inhibitory postsynaptic potentials (IPSPs) which have the opposite effect. The neural network has the ability to learn from experiences by changing the strength of connections between neurons, known as synaptic plasticity. Similarly, MLPs are made up of artificial neurons called perceptrons, which carry computer algorithms to accept, process, and output data. The activation function in the algorithms corresponds to the firing mechanism of biological neurons, to decide whether the output will be positive or negative. The perceptrons in two adjacent layers

DOI: 10.54254/2753-8818/20/20230753

are connected by weights, and the exciting or inhibiting effect of each input on the next perceptron depends on the positive or negative value of weight. The MLPs can also continually learn and optimize by monitoring the errors in predicted results.

The individual perceptron was first invented by Frank Rosenblatt in 1957 [2], and was used to construct single-layered perceptron, which aimed at performing linearly separable classification. The subsequent invention of the more complex multi-layered perceptron was another milestone in the research of artificial neural networks. It was able to carry out contract enhancing, model differential, and XOR logic, according to the two papers of Later Stephen Grossberg in 1972 and 1973 [3]. Up to now, some more advanced artificial neural networks have been invented based on MLP, including convolutional neural networks (ConvNets) that aim at processing data in arrays [4], Transformers and GPT which are good at language-related tasks.

As a neural network with a simple structure, MLPs are widely applied to various fields. The commonest ones are image recognition [5], speech recognition [6] and natural language understanding [7]. Its main advantages are strong computing power coupled with highly autonomous pattern recognition, enabling it to simulate human roles to some extent in life and being a powerful assistance in scientific research to improve speed and precision. As a product of the study of structures in living things, MLP has in turn promoted human understanding of biological neural networks and also helps the reconstruction of brain circuits [8]. In addition, it is applied to biotechnology on drug development [9], protein structure prediction, gene expression analysis, and many other tasks. This article is going to introduce the structures and mathematical algorithms of multi-layered perceptron, starting with its premise, single-layered perceptron, and shortly discuss the applications of MLPs to biotechnology.

2. neural network structures and algorithms

2.1. Perceptron and single-layered perceptron

Perceptrons are the basic units in MLP that run all the algorithms. A single-layered perceptron is composed of only one layer containing one or more neurons. It is only able to complete simple and linearly separable classification tasks. A model of the typical single-layered perceptron for achieving binary classification is applied here (Figure 1). The followings are its components:

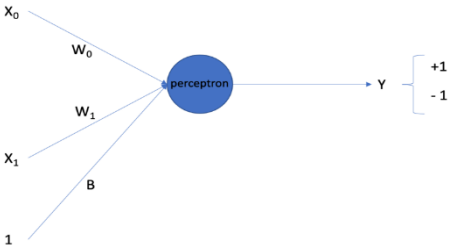

Figure 1. The structure of a single-layered perceptron for binary classification.

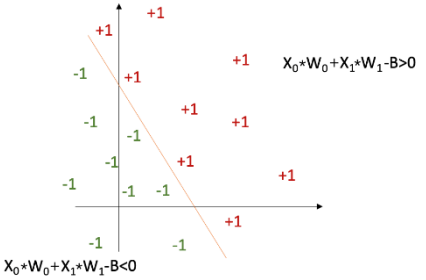

Figure 2. An example of the predicted results of binary classification.

DOI: 10.54254/2753-8818/20/20230753

A. Input (X_0, …, X_n) and output (Y_0, …, Y_n). Arbitrary numbers of external inputs can be passed to the perceptron. In this case, two values X_0 and X_1 are taken just to further simplified the model. One output Y is computed, and this can be either +1 or -1, corresponding to the two optional categories (Figure 2).

B. Weights (W_0, …, W_n). Each input has its associated weight when connecting to a perceptron, representing the degree of impact it has on the output. Two weights W0 and W1 here correspond to X_0 and X_1 respectively.

C. Bias (B). The bias is the threshold to determine whether the inputs would have a positive or negative impact on the output layer. Each perceptron in the neural network has an associated bias. The more negative it is, the easier the perceptron generates a positive output.

D. Perceptron. The internal algorithm of the perceptron consists of the below steps:

a) Each input is multiplied by its weight and all the results are summed up, i.e.,

$$\sum_{i=0}^{n} X_i * W_i \tag{1}$$

b) Check whether the sum exceeds the bias to decide if it has a positive influence on the output, i.e.,

$$Z = \sum_{i=0}^{n} X_i * W_i - B \tag{2}$$

c) Activation function. It is aimed to restrict the result from (b) into a specific range, normally from 0(or -1) to +1 inclusively. Several types of activation functions can be applied. The one used for binary classification problems is the bipolar step function (Figure 3), where +1 is taken for the positive parameter and -1 is taken for the negative parameter, i.e.,

$$\begin{cases} f(z) = 1 & z > 0 \\ f(z) = -1 & z < 0 \end{cases} \tag{3}$$

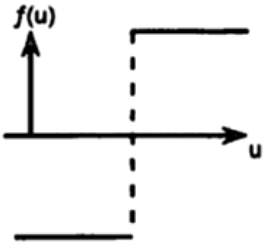

Figure 3. Bipolar step function.

Other common activation functions include step function, Sigmoid function, Tanh function, and ReLu function [2]. Ultimately, the values of +1 or -1 are plotted on the graph, separated by a linear function to indicate which category each of them belongs to.

2.2. Multi-layered perceptron

2.2.1. Prediction. Compared with single-layered perceptron, MLP adds intermediates called hidden layers. Since more than one layers are present, the outputs of the previous layer are passed to the next layer as the inputs. Each perceptron in MLP has a different bias and activation function, therefore MLP can perform more complicated tasks such as non-linear classification (e.g., XOR). The model used here consists of one hidden layer and two perceptrons in each layer (Figure 4).

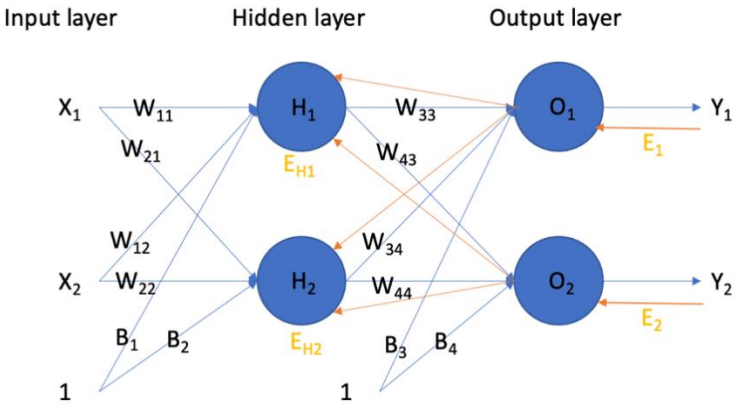

Figure 4. The structure of a multi-layered perceptron and its backpropagation path.

The working principle of MLP to predict outputs is the same as that of single-layered perceptron, except for some changes in detail. One of the main differences is the addition of a matrix algorithm when computing the sum of products of all inputs and weights, which can be seen in Eq.(4) since multiple perceptrons are connected to each input in pairs. Besides, the activation function mainly applied to MLP is called the Sigmoid function (i.e., $\delta(x) = \frac{1}{1+e^{-x}}$). It limits the output within a range of 0 to +1 inclusively, but with higher precision for each value (Figure 5).

$$\begin{bmatrix} w_{11} & w_{12} & B_1 \\ w_{21} & w_{22} & B_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} w_{11} * x_1 + w_{12} * x_2 + B_1 \\ w_{21} * x_1 + w_{22} * x_2 + B_2 \end{bmatrix} \tag{4}$$

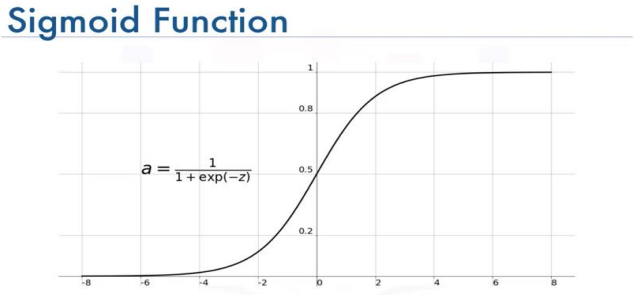

Figure 5. Sigmoid function.

2.2.2. Training. Training the MLP is an optimization process of weights and biases before the precision of prediction is refined to a satisfactory level. This is a process of supervised learning where the MLP is fed by a large amount of training data with known outputs. The parameters in the algorithm will be continuously adjusted until a zero difference presents between the generated output and the target (although this is just an ideal situation). The whole process contains the following steps:

A. Feedforward path. The training data is fed into the MLP. Outputs are gained using the same steps as in the prediction process.

B. Backpropagation. Compute the deviation between each output and the target, which is known as an error. The error is then passed backward for adjusting the algorithm parameters. This procedure is called backpropagation. A cost function about Wi and Bi is constructed to reflect the magnitudes of error when using different values of these parameters. The cost function is normally expressed as the mean squared error:

$$\text{cost function} = \sum_{i=1}^{n} [\![(Y_i]\!] - \text{target}_i)^2 \tag{5}$$

DOI: 10.54254/2753-8818/20/20230753

C. The more obvious deviation between each pair of output and target, the larger the function value, thus the worse performance of MLP. Therefore, the goal is to find the minimum of this cost function. (Note that in a real situation, due to many parameters in one MLP, the cost function is usually three-dimensional, so the task considered is to find the lowest point throughout a hook face (Figure 6). This process applies a mathematical method called gradient descent. A short view of the principle of gradient descent is given below.

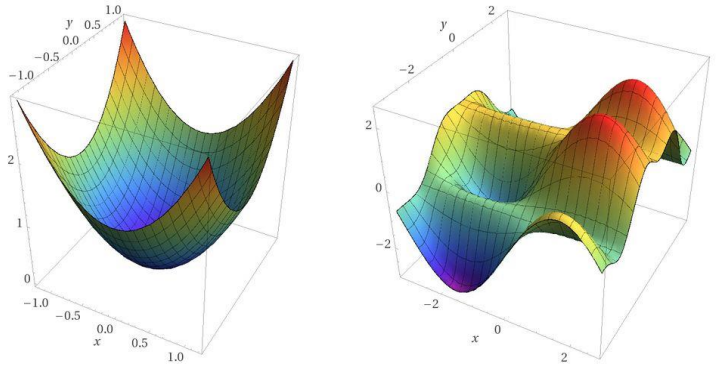

Figure 6. An example model of a high-dimensional function applying gradient descent.

a) Randomly generate the initial weights and biases, within a reasonable range. This action is equivalent to selecting a point on the cost function with the random values of independent variables.

b) Use derivation to get the function gradient at this point. Move along the function towards the direction that can lead to smaller function value, according to the line of the gradient. During the movement, the values of independent variables are altered based on the formulas:

$$\Delta W_i = E_{rror} * X_i * \text{learning rate}$$
$$\Delta B_i = E_{rror} * \text{learning rate} \quad (6)$$

where the learning rate is a hyperparameter that controls the step size of the movement at each iteration. This alteration is continuously iterated until the lowest point is reached. However, the algorithm may be stuck in a local minimum, which means it reaches one of the minimum stationary points of the cost function, but not the lowest point throughout the whole hook face (known as the global minimum). To avoid this situation to the greatest extent, here are several common solutions:

Random parameter initialization. Initial weights and biases are selected multiple times and randomly so that more different local minima are likely to be recorded in the iterations, and thus the global minimum is less likely to be ignored. The number of repetitions is determined by the complexity of the cost function. Generally, the more perceptrons in one layer of the MLP, the more complex the cost function, and the more random initializations are required to ensure that every local minimum is experienced. Note that even so, such an outcome can only ever be ideal.

1. Learning rate scheduling [1]. Since the learning rate represents the step size of the descent, choosing an appropriate value is crucial to the performance of the training process. A too-high learning rate can lead to unstable learning known as overshooting. A too-low learning rate can lead to slow convergence and inefficient training. One simple way of the learning rate schedule is to change its value once in each iteration, ensuring that the learning rate is always proportional to the gradient. This method can help the network learn quickly in the beginning and then fine-tune the weights as it approaches the minimum.

2. The momentum term represents the weighted average of the previous parameter updates. In momentum-based gradient descent, each parameter adjustment also takes into account the direction and magnitude of the previous parameter updates, rather than solely based on the current gradient of the cost function. This allows the algorithm to jump over the local minimum and continue in the direction of the global minimum. The momentum term is another hyperparameter, whose value determines the weighting given to the previous parameter updates. A higher momentum means that more emphasis will

DOI: 10.54254/2753-8818/20/20230753

be placed on the previous updates, while a lower momentum means that more weight will be given to the current gradient.

D. The above parameter adjustment process happens between two adjacent layers. In the backpropagation pathway, it is also necessary to infer the error of the former layer from that of the latter layer. The formulas are listed below.

$$E_{H1} = \frac{W_{11}}{W_{11} + W_{12}} * E_1 + \frac{W_{21}}{W_{21} + W_{22}} * E_2$$
$$E_{H2} = \frac{W_{12}}{W_{11}+W_{12}} * E_1 + \frac{W_{22}}{W_{21}+W_{22}} * E_2 \quad (7)$$

Note that the weights are also involved in the calculation, and the error of one former perceptron is proportional to its corresponding weight. That is because the larger the weight, the greater the influence of the error of the former layer on the latter layer.

In MLP training, tens of thousands of sets of training data are often used to achieve adequate accuracy. An amount of test data is then needed to go through the normal feedforward path, to test how the MLP works in real applications. Finally, MLP can come into service for data prediction.

3. Applications in biotechnology

In biology-related fields, the application of MLP can cover bioinformatics [10], pharmacology, medical imageology, and many other subdisciplines. This section will focus on two major issues that can be addressed by MLP, briefly explain how MLP can work on them, and introduce the recent research progress.

3.1. Drug design

At the drug discovery stage, which belongs to the early stages of drug development, researchers need to explore potential compounds that can be used for therapy, and then preliminarily test their safety and effectiveness, therefore it can lay a foundation for the subsequent preclinical research and clinical trials. At this stage, a method called virtual screening can be used. Features of molecules are extracted from a large-scale molecule database and input into MLP. The required features are typically molecular structures and associated properties. Through internal algorithms, MLP obtains the molecules' bioactivity, toxicity, solubility, and other properties as results. According to these, researchers can determine whether the molecule interacts with a specific biological target, such as an enzyme or a receptor, and if it can be absorbed harmlessly by the body. In addition to choosing from existing molecules, another method is to train MLP on a database of molecules with specific bioactivity, thus producing new molecules with similar properties.

At present, the application of deep learning to drug design is promising and relatively mature. Many pharmaceutical companies, such as Insilico Medicine [11, 12], have developed relevant technology. Noted that in real applications, MLPs are used together with other types of artificial neural networks to complete the prediction.

3.2. Gene expression analysis

Gene expression analysis is the classification or prediction of a biological function or phenotype based on the expression levels of certain genes. At the preparation stage, gene expression data is collected using technologies such as microarray or RNA sequencing. The data is then preprocessed to remove noise and input into MLP. The outputs are the genes that are differentially expressed in different samples or conditions. Therefore, this method can be used to diagnose genetic diseases and monitor the effects of drugs on certain gene expressions.

4. Conclusion

As a branch of artificial intelligence, MLP's promising applications that are being closely watched include autonomous vehicles, interactive game non-player characters (NPCs), disease diagnosis and personalized treatment planning, and so forth. According to the current research direction, the future improvements in its performance can be in the following aspects: (1) Deeper MLPs, which means the

DOI: 10.54254/2753-8818/20/20230753

MLPs with more hidden layers, are developing in order to adapt to the increasing task difficulty; (2) Reinforcement learning, which is the ability of neural networks to learn from experiences, can be implemented in more applications; (3) At present, the internal running process of artificial neural networks is not yet understood by a human. Developing explainable AI might help us better monitor its behaviors and optimize its performance.

References

[1] Gardner, M. W., & Dorling, S. R. “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences.” Atmospheric environment, 32(14-15), 2627-2636. (1998).

[2] Singh, J., and Banerjee, R. “A study on single and multi-layer perceptron neural network.” In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) 35-40 (2019)

[3] Freund, Y., Schapire, R. E. "Large margin classification using the perceptron algorithm." Machine Learning 37(3) 277–296 (1999).

[4] LeCun, Y., Bengio, Y., and Hinton, G. “Deep learning.” nature, 521(7553), 436-444 (2015).

[5] Farabet, C., Couprie, C., Najman, L. and LeCun, Y. “Learning hierarchical features for scene labeling.” IEEE Trans. Pattern Anal. Mach. Intell 35, 1915–1929 (2013).

[6] Hinton, G. et al. “Deep neural networks for acoustic modeling in speech recognition.” IEEE Signal Processing Magazine 29, 82–97 (2012).

[7] Doan, S., Conway, M., Phuong, T. M., and Ohno-Machado, L. “Natural language processing in biomedicine: a unified system architecture overview.” Clinical bioinformatics, 275-294 (2014).

[8] Helmstaedter, M. et al. “Connectomic reconstruction of the inner plexiform layer in the mouse retina.” Nature 500, 168–174 (2013).

[9] Dara, S., Dhamercherla, S., Jadav, S. S., Babu, C. M., and Ahsan, M. J. “Machine learning in drug discovery: a review.” Artificial Intelligence Review, 55(3), 1947-1999 (2022).

[10] Narayanan, A., Keedwell, E. C., and Olsson, B. “Artificial intelligence techniques for bioinformatics.” Applied bioinformatics, 1, 191-222 (2002)

[11] Zhavoronkov, A., Ivanenkov, Y.A., Aliper, A. et al. “Deep learning enables rapid identification of potent DDR1 kinase inhibitors.” Nat Biotechnol 37, 1038–1040 (2019).

[12] Miglani, A., and Kumar, N. “Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges.” Vehicular Communications, 20 100-114 (2019).