World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 7.029

ARTIFICIAL NEURAL NETWORKS AS A TECHNOLOGY SHAPING OUR FUTURE

Oleg Namicheishvili, *Mikheil Ramazashvili, Zhuzhuna Gogiashvili and Natia Namicheishvili

Georgian Technical University.

Article Received on 08/02/2025 **Article Revised on 28/02/2025** **Article Accepted on 18/03/2025**

***Corresponding Author**
Mikheil Ramazashvili
Georgian Technical University.

ABSTRACT

From automatic image annotation to voice recognition, from computers capable of defeating champions in Go to autonomous vehicles, many recent successes in artificial intelligence are based on deep neural networks. Deep learning has become a key topic in discussions today.

Artificial neural networks are essentially non-parametric, potentially complex models: unlike linear regression, they allow for the creation of highly flexible models.

In this article, we discuss the fundamental principles of multilayer perceptrons and their training. We will also briefly touch on deep

(hierarchical) neural networks. The objectives are:

- Understanding the decision-making process of single-layer and multilayer perceptrons;
- Implementing the training procedure of a perceptron;
- Explaining the update mechanism of a multilayer perceptron using backpropagation;
- Understanding the key principles of building and training a multilayer perceptron;
- Understanding the tasks of deep learning.

1. INTRODUCTION: The Perceptron

The history of artificial neural networks dates back to the early 1950s, with their effectiveness in modeling complex nonlinear relationships being a key factor in their success.

The first artificial neural network was Frank Rosenblatt's perceptron (Rosenblatt, 1957).

This network was not deep; it had a single layer and limited modeling capabilities.

1.1 Model

A perceptron (Fig. 1) consists of an input layer where p neurons or modules (*block),* represented and each of them corresponds to individual input variables.

Each neuron transmits its input value to the next layer. This p neuron is usually supplemented with a shift module (block), which always outputs a value equal to 1.

The perceptron's first and only layer (after the input layer) contains a single neuron connected to all input units (block).

This neuron computes a linear combination $o(\vec{x}) = w_0 + \sum_{j=1}^{p} w_j x_j$ of incoming signals $x_1, x_2, \ldots, x_p$, applies an activation function a, and transmits the result to the output, which serves as the decision function of the perceptron. Thus, the perceptron is a *parametric model*.

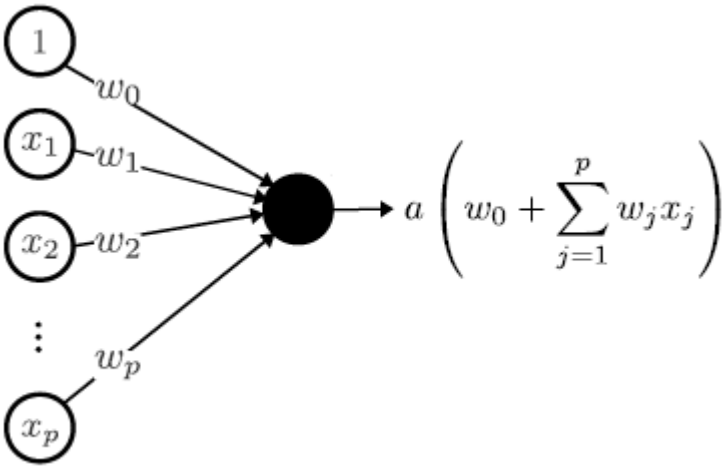

Fig. 1: Perceptron's architecture.

Activation Function. For regression tasks, the *identity function* suffices as the activation function.

For binary classification, different approaches can be used:

— A threshold function for direct binary prediction;

— A logistic function for probabilistic classification of belonging to the positive class.

Multiclass Classification. In the case of multiclass classification, the perceptron's architecture is modified such that the output layer consists of C neurons, where is the C number of classes. Each output $p+1$ neuron is connected to all preceding neurons

(accordingly, we will have a weight matrix representation $(p+1)C$) (Fig. 2).

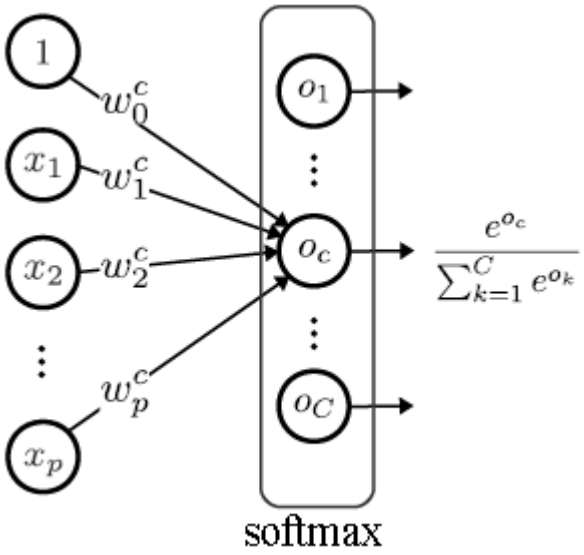

Fig. 2. – Multiclass perceptron's architecture.

1.2 Training

To train a perceptron, we aim to minimize empirical risk, similar to parametric regression models.

However, we assume that observations are available sequentially rather than simultaneously. This assumption is inspired by the plasticity of biological neural networks: they continuously adapt to incoming signals. Therefore, we employ an *incremental* learning algorithm designed for sequentially incoming observations.

Incremental and Autonomous Learning: *Autonomous* learning refers to a learning algorithm that operates on a single set of n observations. *Offline learning* is also commonly used (batch learning).

In contrast, *incremental* learning updates model parameters with each new observation. *Online learning* is also commonly used.

Learning Rate: In practice, the learning rate is often high at the beginning and gradually decreases as training progresses, ensuring stability in convergence. Such an approach can be compared to analogy algorithms.

2 Multilayer Perceptron

The perceptron's modeling capabilities are limited because it is a linear model. Initial enthusiasm surrounding neural networks waned in the early 1970s when researchers realized these limitations.

"The study of the perceptron has been very interesting, despite its serious limitations, and perhaps because of them. Some of its features are noteworthy: linearity, an intriguing learning theorem, and the simplicity of the parallel computing paradigm."

2.1 Architecture

A multilayer perceptron (MLP) introduces *intermediate layers (hidden layers)* between the input and output layers. Each neuron in these hidden layers receives inputs from the previous layer and passes outputs to the next layer. No feedback connections exist; hence, this is a *feed-forward* neural network.

Using nonlinear activation functions such as the logistic function or hyperbolic tangent function enables the creation of powerful nonlinear parametric models.
An example of an MLP with two hidden layers is shown in Fig. 3.

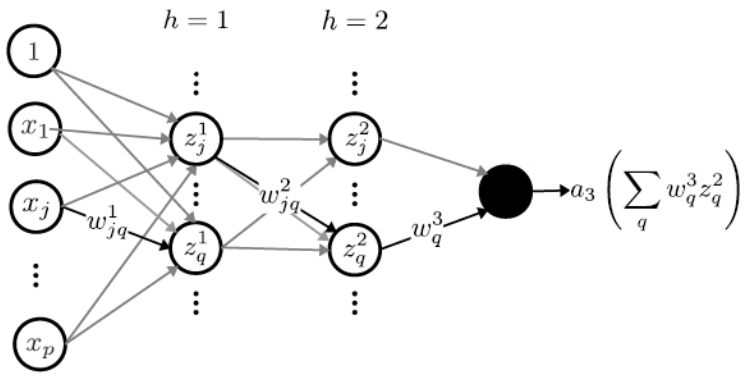

Fig.3 - Multilayer perceptron's architecture.

The number of layers and neurons per layer are considered hyperparameters and are fixed during training. To avoid overfitting, deep neural networks require large datasets for effective training.

Multilayer perceptron — this is a parametric model, the number of layers and the number of neurons in these layers are part of the hyperparameters: we assume that they are fixed and are not being trained. Thus, the more parameters, i.e. the number of connections, this model has,

the more intermediate layers and the number of neurons in these layers. To avoid overfitting, deep neural networks require large datasets for effective training.

2.2. Training via Backpropagation

Gradient Conditioning and Stability Issues

It is important to emphasize that minimizing empirical risk in MLPs is a non-convex optimization problem. Therefore, gradient descent-based algorithms do not guarantee convergence to a global minimum.

Recall from mathematics that the *inverse of a function* with respect to an argument measures the magnitude of the change in the value of the function with a small change in the argument.

Indeed, the gradient of the cost function with respect to the link weight is typically poorly conditioned, meaning that even a small perturbation of the initial conditions of the algorithm will lead to a completely different and unexpected result. Furthermore, the gradient with respect to the weight in one of the first intermediate layers is often unstable. It is either very small, which slows down the learning process, and this is known as the *vanishing gradient* problem, or it takes on increasingly larger values with each iteration, and this leads to the *exploding gradient* problem.

Initialization of connection weights, standardization of variables, and choice of learning rate and activation function affect the ability of a multilayer perceptron to converge to a good decision.

That is why training a multilayer perceptron is a difficult task. And only in 2006, the works of Geoffrey Everest Hinton (1947-), a British-Canadian cognitive psychologist and computer scientist, considered by the educated world to be the "Godfather of AI", and work of other researchers, as well as the increase in computer power, provided a way to overcome these difficulties and brought the study of neural networks back to the forefront.

Saturation is another problem associated with multilayer perceptrons, occurring when the logistic modules (blocks) return values that are very close to 0 either or 1.

Backpropagation

Despite these challenges, the primary method for training MLPs remains backpropagation (often abbreviated as backprop). Just like in the case of the perceptron, the intermediate layer

solution is based on the use of a gradient descent.

This will allow us to simplify calculations by using the *memoization method*, i.e. to avoid recalculating ratios that occur several times in our procedure.

More precisely, training a multilayer perceptron using the backpropagation method consists in alternating between a forward propagation phase for each observation $\left(\vec{x}^i, y^i\right)$ to be processed, in which the outputs of each neuron are calculated, and an error backpropagation phase, in which the weights are updated: 1) this update process starts with the weights that go from the last intermediate layer to the output module (block) and 2) continues "up" in the network to the input weights that connect it to the first intermediate layer.

2.3 Deep Learning and Neural Networks

From image analysis to autonomous vehicles and speech recognition, most recent advances in machine learning rely on deep neural nets (DNNs).

An MLP becomes **deep** when it has a sufficient number of layers. The exact *threshold* for depth varies among researchers.

Deep learning is also interested in many other architectures, such as Recursive Neural Networks (RNNs), namely Long Short-Term Memory (LSTMs) for modeling sequential (e.g., textual or temporal) data, and Convolutional Neural Networks (CNNs) or Capsule Neural Networks (CapsNets) for image processing. In all cases, the point is, in fact, to:
— use these architectures to create (potentially very complex) parametric models;
— train, teach the weight coefficients using a directed approach algorithm.

Challenges of Deep Learning. Training weights for deep neural networks is associated with technical difficulties: the optimization problem is not convex, and convergence to a “good” local minimum is not an easy task. The more complex the network, the more difficult the task, and progress in this area can only be achieved by creating methods that simplify its solution.

In addition, deep neural networks have a large number of parameters, and therefore these networks require large amounts of data to avoid unnecessary, overfitting.

Therefore, it is usually necessary to deploy them on distributed architectures.

So, despite its success in some areas of application, deep learning is difficult to implement and is not always the optimal approach for solving machine learning problems, especially when dealing with small datasets.

Deep neural networks are often associated with the *concept of learning representations* about a *subject or features about a subject (object).* Indeed, it can be considered that each subsequent intermediate layer learns to represent the data in a new way based on the representation of the previous layer, until it becomes possible to use a linear algorithm between the last intermediate layer and the network output. This aspect is used, in particular, in autoencoders.

3 CONCLUSION

- The perceptron is useful for training linear parametric models via iterative weight updates.
- The multilayer perceptron (MLP) extends modeling capabilities to nonlinear parametric models.
- MLPs are trained using *backpropagation*, which combines gradient-based optimization with *memoization* techniques the essence of which is to store the previous results of a function and use them later, instead of calling this function again, thereby ensuring high computational efficiency. It should not be confused with the word *memorization*, which simply means remembering (without specifying the purpose of this operation).
- Optimization for MLPs and deep networks is non-convex, and obtaining a good minimum is not easy.
- Neural networks are a technology that is already shaping our future, in which the following directions will be presented:

- Personalized learning, when a neural network analyzes each student's learning preferences, abilities, and pace, and based on that, creates an individual learning program for them.
- Healthcare, when neural networks are the basis for both diagnosis and treatment, provide a prognosis for the development of diseases and select an individual treatment regimen for each patient.
- Creative industry, when a neural network not only writes music, creates artwork and text, but also becomes a fully-fledged partner of a person in the creation of new art.

4 BIBLIOGRAPHY

1. Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4): 303–314.
2. Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press, Cambridge, MA. https://www.deeplearningbook.org/
3. Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2): 251–257.
4. Minsky, M. and Papert, S. (1972). *Perceptrons: an Introduction to Computational Geometry*. MIT Press, Cambridge, MA.
5. Novikoff, A. B. J. (1962). On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, Polytechnic Institute of Brooklyn, Vol. XII, pages 615–622: https://cs.uwaterloo.ca/~y328yu/classics/novikoff.pdf.
6. Rosenblatt, F. (1957). The perceptron. A perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, INC., Buffalo, N. Y., Project PARA, Prepared by: Frank Rosenblatt, Project Engineer : https://blogs.umass.edu/brain-wars/files/2016/03/rosenblatt-1957.pdf .
7. ფრანგიშვილი, ა.ი., ნამიჩეიშვილი, ო.მ., გოგიაშვილი ჟ.გ. (2020). ღრმა სწავლება, თბილისი : საგამომცემლო სახლი „ტექნიკური უნივერსიტეტი“, 155 გვ.