**UDK** 004.896.6

**Medvedieva S.O.**
**Bogach I.V.**
**Kovenko V.A.**

# NEURAL NETWORKS IN MASHINE LEARNING

Vinnytsia National Technical Universityy

***Анотація***

*В статті розглянуті основи роботи з нейронними мережами, особливу увагу приділено моделі мережі під назвою «перцептрон», запровадженої Френком Розенблаттом. До того ж було розкрито тему найпоширеніших мов програмування, що дозволяють втілити нейронні мережі у життя, шляхом створення програмного забезпечення, пов'язаного з ними.*

**Ключові слова:** нейронні мережі, машинне навчання, поріг, зміщення, синапси, архитектура прямого зв'язку, точність, перцептрон, пороговий нейрон, прихований шар, шар входу, шар виходу, оптимізатори.

***Abstract***

*The paper covers the basic principles of Neural Networks' work. Special attention is paid to Frank Rosenblatt's model of the network called "perceptron". In addition, the article touches upon the main programming languages used to write software for Neural Networks.*

**Keywords:** Neural Networks, Machine Learning, threshold, bias, synapses, feedforward architecture, accuracy, perceptron, binary threshold neuron, hidden layer, input layer, output layer, optimizers.

Neural Networks (NN) is just a method of machine learning and is not a standalone artificial intelligence, though it is a part of it. Besides Neural Networks, there are lots of other methods of machine learning, for instance: Random Forest and K-nearest neighbors. Anyway, new architectures of Neural Networks proved to be a promising technology.

Nowadays Machine Learning (ML) algorithms are very popular and without exaggeration in subsequent years the interest towards this theme will only be increasing. The improvements in the technologies lead humanity to the new horizons and discoveries. That's why it is more than important to be familiar with the notion of NN, even just on an intuitive level. In the 21th century, people have got used to their gadgets and the latest scientific inventions. On average, everyone encounters technologies based on machine learning every day. For example, let us consider setting a finger-print block on your phone. In order to do it, your phone has to remember your fingerprint and then recognize it with an accuracy of 100 percent. The *NN* is used to do it. The same thing concerns the face block. Or what about the applications that give us an opportunity to recognize the whole song and its author by listening just to a small part of it? *NN* is used to fulfil this function. So, the main fields of using NN deal with classifying and making decisions (for instance, we need to decide whom to give a credit having a credit history about the interested people), recognition and prediction.

Thus, what is the *Neural Network*? As the name implies, this technology is built around the neurons and their activity towards each other. Human's brain also has such a component as neural network, but the NNs used in machine learning are simplified in thousands or even millions times to adjust them to the computers' power. Let us think about neurons in *NNs* like about the rooms that store some value. There are five types of neurons: *linear, binary threshold, rectifier linear, stochastic binary* and *sigmoid.* The difference between them lies in value that they output. In our example we will use *binary threshold neurons* and an algorithm called *perceptron*. *Binary* threshold neuron is available to store and output only binary values (0 and 1). This fact makes it seem useless in most cases. In order to understand how the *forward propagation* works,

we should consider the *perceptron* example. However, several things are better to be learnt before heading to the named instance. *NNs* usually consist of multiple layers. The first one is called an *input layer*, this is a layer which contains input neurons with their value. The second one is a *hidden layer,* the one in which all the calculations are done. And the last one is an *output layer,* neurons of which store the output.

Such an architecture is called *feedforward*, as every neuron in the previous layer is connected to each one of the next layer. The connections between neurons are called *synapses*, and they store some value called *weights*. *Weight* is often represented as *w* and it is a value responsible for the importance of information stored in neurons. There is also a value called *threshold*, the main aim of which is to regulate the output data depending on the value computed in the *hidden layer,* but more often the other concept called *bias* is used (*bias = -threshold)*. By the way, the *NN* can have more than one *hidden layer* and this can allow our network to learn better and to vary data patterns.

There are lots of languages that can provide us with the implementation of machine learning algorithms, but they all differ in the complexity of fulfilling this task.

In this passage we will tell you about 3 languages that are in great demand in the sphere of *Machine Learning* and Artificial Intelligence.

The first one is a dynamic programming language named Python. As this language is dynamic, it has a simplified syntax, so that the time-consuming program made with the help of, for example C++, can be written in Python much faster and easier. Additionally, there are special frameworks for implementing machine learning algorithms in Python, so that the one can focus on the architecture of his *NN* or on the other necessary features, than spend lots of time on making a handwritten *NN*. Anyway, you can make a handwritten *NN* without additional libraries in order to boost your programming skill and get the wider understanding of the *NN* process. Speaking about frameworks for *Machine Learning* in Python, the most famous are: *tensorflow*, *pytorch*, *numpy*, *keras* and *scikit-learn*.

Even without knowing programming languages anyone will say that there aren't many rows of code here for such complicated technologies as *Machine Learning* algorithms. But there is also the opposite side of the medal, as Python is a dynamic language, meaning that it doesn't has a strict typization, it is much slower than the other languages like Java, C++ , Ruby, etc. However, we strongly recommend using exactly this language for *Machine Learning*, as the libraries for these tasks supported in Python are improving day-by-day.

The other language used for *ML* is called R. R is one of the most effective languages and environments for analyzing and manipulating the data for statistical purposes. Using R, one can easily produce well-designed publication-quality plot, including mathematical symbols and formulas where needed. Apart from being a general purpose language, R has numerous packages like RODBC, Gmodels, Class and Tm which are used in the field of machine learning. These packages make the implementation of *Machine Learning* algorithms easy and can be applied for cracking the business associated problems.

And the last programming language I want to notice is Java. Java can also be considered as a good choice for AI development. Artificial intelligence has much to do with search algorithms, artificial neural networks and genetic programming. Java is quite beneficial: it is easy to use, easy to debug, gives a number of package services, simplifies work with large-scale project; it produces graphical representation of data and gives better user interaction. It also has the incorporation of Swing and SWT (the Standard Widget Toolkit). These tools make graphics and interfaces look appealing and sophisticated.

In this article, you've got some basic understanding of *NNs* and *Machine Learning* algorithms. As machine learning is in great demand nowadays and is improving every day, every specialist in the area should always be in search of new knowledge.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. A short online-book about basic concepts of NN. [Electronic resource]. - Mode of access: URL: http://neuralnetworksanddeeplearning.com/index.html.

2. Fitting NN in R example. [Electronic resource]. - Mode of access: URL: https://datascienceplus.com/fitting-neural-network-in-r/.

3. Fitting NN in Python example. [Electronic resource]. - Mode of access: URL: https://towardsdatascience.com/how-to-build-a-neural-network-with-keras-e8faa33d0ae4.
4. Fitting NN in Java example. [Electronic resource]. - Mode of access: URL: https://dzone.com/articles/designing-a-neural-network-in-java.

***Медведєва Світлана Олександрівна,*** викладач кафедри іноземних мов, Вінницький національний технічний університет

***Богач Ілона Віталіївна,*** кандидат технічних наук, доцент кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет

***Ковенко Володимир Андрійович***, студент групи 1ІСТ-18б, Факультет комп'ютерних систем та автоматики, Вінницький національний технічний університет

***Svitlana Medvedieva,*** the teacher of the department of foreign languages, Vinnytsia National Technical University

***Ilona Bogach,*** PhD, Associate Professor of the department of automation and intelligent information technologies, Vinnytsia National Technical University

***Volodymyr Kovenko***, the student of group 1IST-18b, the Faculty of Computer Systems and Automation, Vinnytsia National Technical University