UDK 004.896.6

Bogach I.V.,
Kovenko V.A

NEURAL NETWORKS AND SPHERES OF THEIR USAGE

Vinnytsia National Technical University

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, НН (скорочено від Нейронні Мережі) , машинне навчання, поріг, зміщення, синапси, архитектура прямого зв'язку, точність, перцептрон, пороговий нейрон, прихований шар, шар входу, шар виходу, оптимізатори.

Abstract

The basics of working with Neural Networks are exposed in this article, the model of the network called "perceptron", made by Frank Rosenblatt is mainly emphasized. In addition, the main programming languages, that provide the opportunity to fulfil Neural Networks by making software related to it are highlighted.

Keywords

Neural networks, NN (stands for Neural Networks), machine learning, ML, threshold, bias, synapses, feedforward architecture, accuracy, perceptron, binary threshold neuron, hidden layer, input layer, output layer, optimizers.

Neural Networks is just a method of machine learning and is not a standalone artificial intelligent, 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 that this technology is better.

The three subthemes will be exposed in this article:

1.The aims of NN.

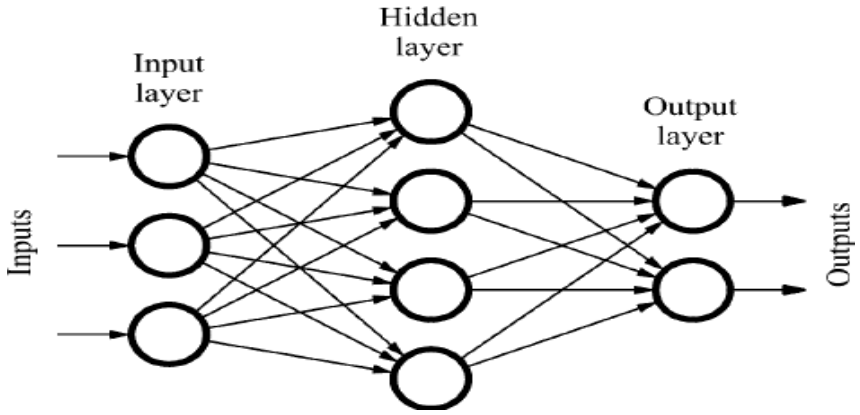
2. Basic concepts of NN.

3. Programming languages used to implement NN.

Nowadays Machine Learning algorithms are very popular and without exaggerating in subsequent years the interest towards this theme will only be increasing. The improving of these technologies leads humanity to the new horizons and discoveries. That's why it is more than important to be familiar with this technology, even just on an intuitive level. In the 21th century people got used to their gadgets and new science inventions. In average, everyone encounter technologies based on machine learning every day. For example, let's consider setting a finger-print block on your phone. In order to set it, your phone have to remember your fingerprint and then recognize is it with an accuracy of 100 percent. The *NN* is used to do it. The same thing with the face block and so on. Or what about the applications that give us an opportunity to recognize the whole song and its author by listening just to the small part of it? *NN* is used to fulfil this function. So the main fields of using NN is 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.

1. The aims of Neural Networks

So 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 NN's that are used in machine learning are simplified in thousands or even millions times to adjust them to the computers' power. Let's 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 is 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 review the *perceptron* example. However, there are several things that 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*, this the one in which all the calculations are done. And the last one is an *output layer*, neurons of which store the output. You can see the example of this architecture below (picture 1):



Picture 1 – Example of NN architecture

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* that is often represented as *w* 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. Now, knowing some basic expressions related to *NN*, we can head towards our example.

2. Basic concepts of Neural Networks

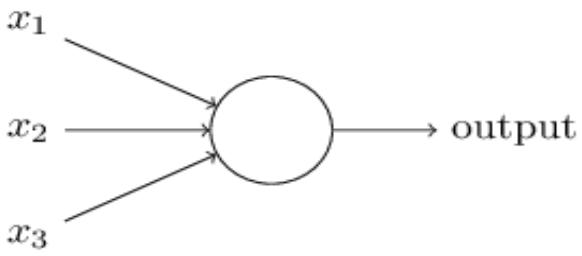
Let's imagine the situation in which we have to decide either visit the cheese festival or not. There are some circumstances that can influence our decision:

1. Going there with a friend.
2. Using a car to get to the destination.
3. The weather condition is really adorable.

We will use a primitive *NN* and *binary threshold neurons* to make a decision (picture 2).

As you see on the picture 2, the *NN* representing this situation is constructed of 3 layers. The *input layer* consists of three neurons which is the same number of circumstances that I had described before, so that x_1 corresponds to the first clause, x_2 to the second one and x_3 to the third one. As we speak about binary threshold neurons, the input will be 0 or 1 depending on the verity of our circumstances. Now, assume that you have neither friends nor car, but the weather is sunny and the sky is surprisingly cloudless. That means

that the input of the first and second neurons is 0, and the input of the last one is 1, as this condition



Picture 2 – Example of NN with 3 layers

you have neither friends nor car, but the weather is sunny and the sky is surprisingly cloudless. That means that the input of the first and second neurons is 0, and the input of the last one is 1, as this condition is true. Then, we go ahead to the *hidden layer*. As it had been said recently, all the computations are done in this layer. The formula presented below is used to do it:

$$\sum_j w_j x_j . \tag{1}$$

Where *w* stands for weight of ***synapse*** and in our situation it represents the importance of a single condition. In our case let's suppose that the importance of the first and second conditions equals 0.2 and the importance of the third one is 0.6. Then, in order to understand what is the right decision towards this situation we should use this formula:

$$output = \begin{cases} 0 \quad if \ \sum_j wx \ + bias \ \leq 0, \\ 1 \quad if \ \sum_j wx \ + bias \ > 0. \end{cases} \tag{2}$$

If output equals 0 or is less than 0, means that the best decision is to stay at home and not to visit the festival. Otherwise, going to festival is a really brilliant idea. In our example you should identify bias by yourself. Let it be -4 for instance, then in order to choose the right decision, we should do such calculations as:

$$\begin{gathered} w_1 x_1 + w_2 x_2 + w_3 x_3 + bias; \\ 0.2 \cdot 0 + 0.2 \cdot 0 + 0.6 \cdot 1 - 4 < 0 \ \rightarrow \ \text{output} = 0. \end{gathered} \tag{3}$$

After some calculations we see that we got a negative number which leads us to the conclusion that it's not recommended to visit the festival. It was just a simple example of a primitive *NN*, in reality no one initializes biases and weights by himself. Such thing as *optimizers* are used to find the best values for biases and weights, about which you can learn yourself.

3. Programming languages used to implement Neural Networks

We had already described something about the basic concepts of *NN*, but in practice the one should implement it into the code so that the machine could understand it. 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 using 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*. Any way 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*. Below you can see the example of *NN* in Python using *keras*:

```
import numpy as np
from keras.utils import to_categorical
from keras import models
from keras import layers
from keras.datasets import imdb
(training_data,    training_targets),    (testing_data,    testing_targets)    =
imdb.load_data(num_words=10000)
data = np.concatenate((training_data, testing_data), axis =0)
targets = np.concatenate((training_targets, testing_targets), axis =0)

def vectorize(sequences, dimensions=10000):
      results = np.zeros((len(sequences), dimensions))
      for i, sequence in enumerate(sequences) :
           results[i, sequence] = 1
       return results

data = vectorize(data)
targets = np.array(targets).astype("float32")

test_x = data[:10000]
test_y = targets[:10000]
train_x = data[10000:]
train_y = targets[10000:]

model = models.Sequential()
# input - layer
model.add(layers.Dense(50, activation = "relu", input_shape=(10000,
)))
# Hidden - Layers
model.add(layers.Dropout(0.3, noise_shape=None, seed=None))
model.add(layers.Dense(50, activation = "relu"))
model.add(layers.Dropout(0.2, noise_shape=None, seed=None))
model.add(layers.Dense(50, activation = "relu"))
# Output- Layer
model.add(layers.Dense(1, activation = "sigmoid"))
model.summary()
# compiling the model
model.compile(
     optimizer = "adam",
     loss = "binary_crossentropy",
     metrics = ["accuracy"]
)
results = model.fit(
     train_x, train_y,
     epochs= 2,
     batch_size = 500,
     validation_data = (test_x, test_y)
)
print("Test-Accuracy:", np.mean(results.history["val_acc"]))
```

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 medal, as Python is a dynamic language, meaning that it doesn't has ta strict typization, it is much slower than other languages as Java, C++ , Ruby and so on. 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 language and environment for analyzing and manipulating the data for statistical purposes. Using R, the 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 of 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, for cracking the business associated problems.

And the last programming language I wanted to notice is Java. Java can also be considered as a good choice for AI development. Artificial intelligence has lot to do with search algorithms, artificial neural networks and genetic programming. Java provides many benefits: easy use, debugging ease, package services, simplified work with large-scale projects, graphical representation of data and 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 a basic knowledge concerning *NNs* and *Machine Learning* algorithms. It was just a small step towards this wide theme, and as machine learning is in great demand it is improving every day, thus the one should always be in search of new knowledge. To sum up, machine learning are such kinds of technologies which we encounter every day and soon having even a little skill related to it will be essential.

THE LIST OF USED LITERATURE

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.

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.

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

Ковенко Володимир Андрійович, the student of group 1IST-18b, the faculty of computer systems and automation, Vinnytsia National Technical University