УДК 004.932.2

INTRODUCTION TO NEURAL NETWORKS AND DEEP LEARNING

БОБРОВА НАТАЛЬЯ ЛЕОНИДОВНА
к.т.н., доцент
СТЕЦКО ВАДИМ ЮРЬЕВИЧ,
ХАРКЕВИЧ АНТОН ПАВЛОВИЧ
магистранты
УО «Белорусский государственный университет информатики и радиоэлектроники»

Научный руководитель: Шевалдышева Елена Зигфридовна
к.ф.н., доцент
УО «Белорусский государственный университет информатики и радиоэлектроники»

Аннотация: В данной статье рассматриваются ключевые концепции, такие как структура нейрона, активационные функции и механизм обратного распространения ошибки. Также рассматривается роль глубокого обучения в различных областях.
Ключевые слова: Нейронные сети, глубокое обучение, машинное обучение, нейрон, матрица весов, смещение.

ВВЕДЕНИЕ В НЕЙРОННЫЕ СЕТИ И ГЛУБИННОЕ ОБУЧЕНИЕ

Bobrova Natalya Leonidovna,
Stetsko Vadim Yurievich,
Kharkevich Anton Pavlovich

Scientific adviser: Shevaldysheva Elena Zigfrididovna

Abstract: This article covers key concepts such as neuron structure, activation functions, and backpropagation. The role of deep learning in various fields is also discussed.
Keywords: Neural networks, deep learning, machine learning, neuron, weight matrix, bias.

Introduction

Machine learning is a broad field in the field of AI (artificial intelligence) that focuses on developing algorithms and models that can learn from data, identify patterns, and make predictions or decisions.

Based on this, neural networks are a subset of machine learning inspired by biological neurons. They are algorithmic models consisting of artificial neurons organized in layers. But traditional neural networks (such as multilayer perceptrons) have limited ability to solve complex problems, which has led to the development of deep learning.

It follows that deep learning is a subfield of neural networks that focuses on deep (multiple-layer) neural networks. One of the key ideas behind deep learning is the use of multiple layers to extract more abstract features from data. Deep neural networks can automatically learn representations of data at different levels of abstraction. This allows them to effectively solve complex problems in areas such as computer vision, natural language processing and others.

History of neural networks and deep learning

The first computer was created in 1623 by the German scientist Wilhelm Schickard [1]. It was a calculating device that could perform four arithmetic operations on six-digit numbers. About 340 years later, a group of researchers in the fields of neurobiology and neuroanatomy found that the brain is hundreds of billions of neurons connected to each other [2]. Understanding the functioning of a neuron and its connections allowed researchers to create mathematical models, which, in turn, provided the theoretical basis for the creation of artificial neural networks.

The first artificial neural networks were implemented in the form of electronic circuits. Later, in connection with the development of computer technology, artificial neural networks began to be implemented in the form of programs.

The development of neural networks can be divided into the following stages:

1. Early research (1950 - 1960).

Initial ideas for neural networks emerged, inspired by biological neurons, and the first perceptron (artificial neuron) was created, the first single-layer neural network that could solve linear problems. However, the limitations of the perceptron in solving more complex problems led to a halt in research in this area.

2. Quiet period (1970 - 1980).

Interest in neural networks is fading due to limitations in training and computing resources.

3. Revival (1980 - 1990).

In 1986, David Rumelhart, Geoffrey Hinton, and Ronald Williams introduced the backpropagation algorithm, which allowed multilayer neural networks to be trained efficiently [3], leading to a renewed interest in neural networks.

4. Development of various architectures (1990 - 2000).

Development of various neural network architectures, including multilayer perceptrons with multiple hidden layers.

5. A surge in the development of deep learning and convolutional neural networks (CNN) (2010 - present time).

Deep learning is becoming especially popular due to the advent of large amounts of data and computing power. Convolutional neural networks (CNNs) are being developed for image processing and are becoming a key technology in the field of computer vision. Deep neural networks have been successfully used in a wide range of areas, including speech recognition, natural language processing, robot and car control, medical diagnostics, and others.

These milestones reflect the evolution of neural networks from early attempts to model neurons to their practical and widespread use in modern artificial intelligence applications and research.

Based on the history of the development of neural networks and deep learning, it can be understood that the study and use of artificial neural networks began quite a long time ago - at the beginning of the 20th century, but they really became widely known a little later. This is due, first of all, to the fact that advanced (for that time) computing devices began to appear, the power of which was large enough to work with artificial neural networks.

Artificial neurons and connections

A neural network is a collection of neurons connected to each other in a certain way. An artificial neuron (perceptron) is an element that calculates the output signal from a set of input signals [4]. Figure 1 shows a neuron.

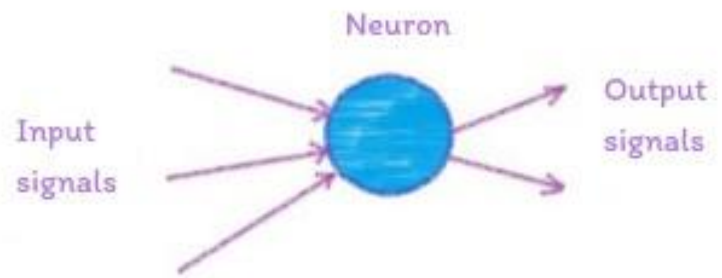

Fig. 1. Artificial Neuron

The perceptron takes input signals, weights them, and applies an activation function to generate an output signal. Mathematically, the operation performed by a neuron can be described is shown below:

$$y = f(\sum_{i=1}^{n} w_i x_i + b), \tag{1}$$

where y – is the output of the neuron;
f – activation function;
x_i – input signal from the previous neuron or external source;
w_i weight assigned to input signal x_i;
b – is a bias that allows the neuron to take into account certain default values.

The connections between neurons are weights that determine the strength and direction of signal transmission between neurons. The weight w_i in Formula 1 indicates how important the corresponding input signal x_i is. The process of training neural networks involves optimizing the weights to best fit the problem data.

The connections between neurons can be represented as a weight matrix W, where each $w_{i\,j}$ represents the weight between neuron i and neuron j.

An important feature of neural networks is the ability to automatically adjust connection weights (including by backpropagation) during the training process, which allows networks to extract and generalize information from data.

Activation functions

Activation functions introduce nonlinearity into the model, which allows neurons to approximate complex nonlinear relationships in the data. Classic activation functions that were used in the early stages of neural network development include the sign function, or sign, sigmoid, and hyperbolic tanh [5]. Also in recent years, an activation function such as the rectified linear unit (ReLU) has become popular.

1. Signum (sign)

This activation function is binary, it takes any real number as input and returns one of three values: -1, 0 or 1, depending on the sign of the input value x. If x is negative, then the function will return -1, if x is zero, then it will return 0, and if x is positive, then it will return 1.

$$sign(x) = \begin{cases} -1, & if\ x < 0 \\ 0, & if\ x = 0 \\ 1, & if\ x > 0 \end{cases} \tag{2}$$

2. Sigmoid

The sigmoid function is one of the first activation functions that was widely used in neural networks. It is an S-shaped curve and is defined by the following formula:

$$S(x) = \frac{1}{1 + e^{-x}}, \tag{3}$$

where x – weighted sum of inputs.

3. Hyperbolic tangent (tanh)

Hyperbolic tangent is another popular activation function that is limited to the range [-1, 1] and is also S-shaped.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4}$$

4. Linear rectified unit (ReLU)

ReLU is a simple threshold function that fires if the input is positive and returns zero otherwise.

$$ReLU(x) = \max(0, x) \tag{5}$$

Activation functions are key components of neural networks, and the choice of a particular function can significantly affect the network's ability to extract features and learn. It is important to select activation functions that best suit the specific task and network structure.

Forward and Backpropagation

Forward and backpropagation are fundamental concepts that underlie the training of neural networks. They describe the processes of data transfer in a network during training and error correction to improve network performance.

1. Feedforward

The process of feedforward starts with the input data and ends in the network. Inputs are passed through layers of neurons using activation functions, and each neuron has an output value based on weights and inputs. This can be represented mathematically as follows for layer l, where l – layer number:

– set the input data for the layer l: $X^{(l)}$;
– calculate the layer output l: $Z^{(l)} = W^{(l)}X^{(l)} + b^{(l)}$;
– apply activation function f to obtain layer activation l: $A^{(l)} = f(Z^{(l)})$.

This process is performed sequentially for each layer from input to output.

2. Backpropagation

Backpropagation is the process of adjusting weights in neural networks based on the received errors. The goal is to minimize the loss function, which estimates the difference between the predicted and expected results. The backpropagation process is carried out in several stages:

– calculation of the gradient of the loss function with respect to the weights and biases of the network (partial derivatives);
– using a gradient to correct network weights and biases using the gradient descent method or its modifications.

Mathematically (for the loss function L, weights W and biases b) this can be represented as follows:

– calculation of the gradient of the loss function with respect to the weights: $\nabla_W L$;
– calculating the gradient of the loss function with respect to displacements: $\nabla_b L$.

The weights and biases are then adjusted in the opposite direction of the gradient to reduce the loss function.

Forward and back propagation processes are the basis of neural network training, and they allow networks to extract features from data and adjust their weights to achieve better performance on end-use tasks.

Loss function and optimization

The loss function (objective function) is a metric that evaluates how well a model performs a task. It measures the difference between the model's predicted values and the true values in the training data. And optimization of the loss function is a process aimed at adjusting the parameters of the model so as to minimize this loss function. Each of these concepts is described in more detail below.

1. Loss function

Loss function L measures the difference between the model's predicted y ̂values and the true y values in the training data. Its choice depends on the machine learning task. Some of the most famous loss functions:

– mean squared error (MSE) used for regression problems, is the simplest and most frequently used;
– cross-entropy – used for classification problems;
– exponential (AdaBoost).

2. Optimization of the loss function

The goal of training neural networks is to minimize the loss function. This is achieved by adjusting the weights W and biases b of the network. One of the most common optimization methods is gradient descent. It uses the gradient of the loss function (a vector that contains all the partial derivatives of a given function with respect to its arguments, or a measure of the change in the function in each direction) of the loss with respect to the network parameters to update them in the direction of a decreasing loss function. The parameter update formula is shown below:

$$W \leftarrow W - \alpha \nabla_W L \quad (6)$$
$$b \leftarrow b - \alpha \nabla_b L, \quad (7)$$

where W – weight matrix;
b – bias vector;

α – learning rate;
$\nabla_W L$ – gradient of the loss function with respect to the weights;
$\nabla_b L$ – gradient of loss function with respect to displacements.

Optimizing the loss function is a key part of training neural networks. It allows the model to find optimal parameters that best fit the problem data. The choice of an appropriate loss function and optimization method depends on the specific problem and network structure.

Deep neural networks and convolutional networks

Deep Neural Networks (DNN) are a class of neural networks that have multiple layers of neurons between the input and output. This allows them to identify complex relationships in data and solve a variety of problems, including classification, regression and image segmentation. One type of deep neural network is Convolutional Neural Networks (CNN), which are well suited for processing spatially structured data such as images.

1. Deep Neural Networks (DNN)

Each neuron in a deep neural network layer is connected to neurons in the previous and next layers. The general view of a neuron operation in a deep neural network can be represented as follows:

$$A^{(l)} = f(W^{(l)}A^{(l-1)} + b^{(l)}) \tag{8}$$

where $A^{(l)}$ – layer activation l.

2. Convolutional neural networks (CNN)

Convolutional neural networks include convolutional layers that can automatically extract features from input images. The main operation in convolutional networks is convolution, which can be represented as follows:

$$O(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n), \tag{9}$$

where O – output pixel;
I – input image;
K – convolution kernel;
(i,j) – pixel coordinates.

Convolutional layers in CNNs are trained to automatically find various patterns and features in images, making them effective in pattern recognition tasks.

Definition of Deep Learning

Deep Learning is a subclass of machine learning in which neural network models with multiple layers are trained to automatically extract features from data. These models are called deep neural networks, and they are characterized by the fact that they have multiple hidden layers, which allows them to model complex dependencies in data.

Figure 2 shows the dependence of performance on the amount of input data of different neural networks. The graph shows that deep neural networks are most effective when there is a large amount of data.

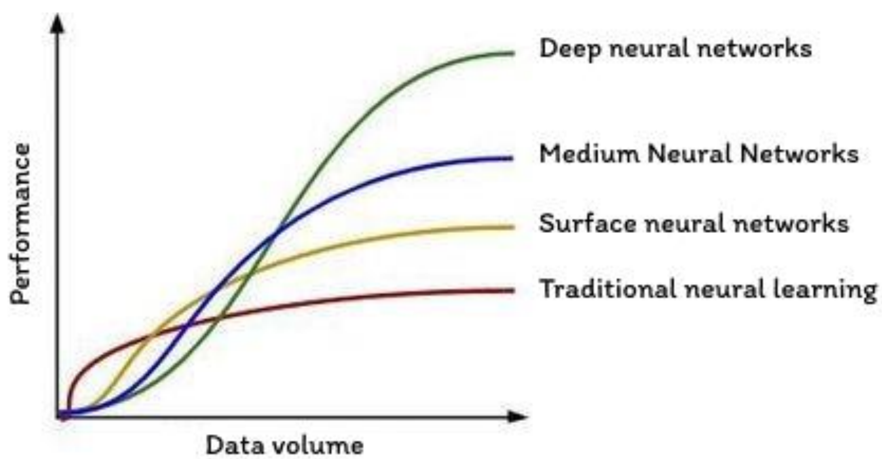

Fig. 2. Dependence of the performance of neural networks on the volume of input data

The basic definition of deep learning can be represented as a general formula for predicting y given input x in a neural network:

$$y = f(W^{(N)} \cdot f\left(W^{(N-1)} \cdot \ldots \cdot f\left(W^{(1)} \cdot x + b^{(1)}\right) + b^{(N-1)}\right) + b^{(N)}) \qquad (10)$$

Deep learning allows models to automatically extract complex features from data, such as images, sound, text, and other types of information. This makes it a powerful tool for a wide range of tasks, including computer vision, natural language processing, speech recognition and more.

The role of deep learning in improving the performance of neural networks

Feedforward neural networks work well for basic tasks, such as identifying simple patterns or classifying information. However, they will have difficulty solving more complex problems.

And deep learning algorithms can process and analyze huge amounts of data thanks to several hidden layers of abstraction. They can perform complex tasks such as natural language processing (NLP) and speech recognition.

Deep learning systems have a wide range of practical applications. Their ability to learn from data, extract patterns, and engineer features allows them to deliver cutting-edge performance.

Application of neural networks and deep learning

The use of neural networks and deep learning covers a wide range of fields and applications, their use leads to significant improvements in various areas, for example:

1. Computer vision – object and face recognition, image segmentation, object detection, optical character recognition (OCR), medical image processing and analysis.
2. Natural language processing (NLP) – machine translation, text and dialogue systems generation, information extraction and text classification.
3. Speech processing – speech recognition and synthesis, text-to-speech (TTS) and speech-to-text (ASR) systems, speech assistants and chatbots.
4. Autonomous systems – cars with autopilot and driving assistance systems, unmanned aerial vehicles, robots for industry and service.
5. Medical diagnostics – diagnosis of cancer and other diseases from images, analysis of biomedical data and images, personalized medicine.
6. Financial analytics – financial market forecasting, fraud detection, credit scoring and risky transactions.
7. Games and entertainment – generation of content and scenarios.
8. Industry and manufacturing - equipment failure prediction, supply chain and inventory management, quality control and defect detection.
9. Environmental modeling – climate and weather monitoring, environmental data and resource analysis, sustainable resource management.
10. Bioinformatics – analysis of genomic data, diagnosis and study of biological processes, prediction of drug interactions.

These examples demonstrate how neural networks and deep learning have applications in a variety of fields, and their ability to automatically extract features and summarize information makes them powerful tools for analyzing and solving complex problems. The application of neural networks continues to expand and has a significant impact on our daily lives, industry and scientific research.

Thus, understanding neural networks and deep learning is of great importance in the modern world. These technologies have driven innovation and revolutionized many industries, including computer vision, natural language processing, autonomous systems, medical diagnostics, and many others.

References

1. Gorbachevskaya E.N., Krasnov S.S. – History of the development of neural networks [Electronic resource] – Access mode: https://cyberleninka.ru/article/n/istoriya-razvitiya-neuronnyh-setey/viewer.
2. Tsoi, Yu.R. Evolutionary approach to setting up and training artificial neural networks / Yu.R. Tsoi, V.G. Spitsyn //Neuroinformatics. -2006. -T.1. -No.1. -P. 34-61.

3. Rumelhart, D., Hinton, G., & Williams, R. "Learning Representations by Back-Propagating Errors." Nature, 1986.

4. Gafarov F.M., Galimyanov A.F. – Textbook – Artificial neural networks and their applications, Kazan – 2018.

5. Williams Publishing House – Introduction to neural networks. Chapter 1 [Electronic resource] – Access mode: https://www.williamspublishing.com/PDF/978-5-907203-01-3/part.pdf.