Using spreadsheets as learning tools for computer simulation of neural networks

Serhiy Semerikov[1,*], *Illia* Teplytskyi[1], *Yuliia* Yechkalo[2], *Oksana* Markova[2], *Vladimir* Soloviev[1], and *Arnold* Kiv[3]

[1]Kryvyi Rih State Pedagogical University, 54, Gagarina Ave., Kryvyi Rih, 50086, Ukraine
[2]Kryvyi Rih National University, 11, Vitali Matusevich St., Kryvyi Rih, 50027, Ukraine
[3]Ben-Gurion University of the Negev, Beer Sheva, Israel

Abstract. The article substantiates the necessity to develop training methods of computer simulation of neural networks in the spreadsheet environment. The systematic review of their application to simulating artificial neural networks is performed. The authors distinguish basic approaches to solving the problem of network computer simulation training in the spreadsheet environment, joint application of spreadsheets and tools of neural network simulation, application of third-party add-ins to spreadsheets, development of macros using the embedded languages of spreadsheets; use of standard spreadsheet add-ins for non-linear optimization, creation of neural networks in the spreadsheet environment with-out add-ins and macros. The article considers ways of building neural network models in cloud-based spreadsheets, Google Sheets. The model is based on the problem of classifying multi-dimensional data provided in "The Use of Multiple Measurements in Taxonomic Problems" by R. A. Fisher. Edgar Anderson's role in collecting and preparing the data in the 1920s-1930s is discussed as well as some peculiarities of data selection. There are presented data on the method of multi-dimensional data presentation in the form of an ideograph developed by Anderson and considered one of the first efficient ways of data visualization.

1 Introduction

The Fourth Industrial Revolution (Industry 4.0) has become a system-related challenge for the scientific community [46]. Industry 4.0 is primarily characterized by evolution and convergence of nano-, bio-, information and cognitive technologies to enhance high quality transformations in economic, social, cultural and humanitarian spheres. Professionals dealing with development and introduction of the sixth techno-logical paradigm technologies determine to a great extent whether our country is able to ride the wave of Industry 4.0 innovations. Therefore, extensive implementation of ICT is a top priority of Ukraine's higher education updating in order to form a professionally competent specialist able to ensure the country's innovative development.

According to the Decree of the Cabinet of Ministers of Ukraine "Certain issues of specifying medium-term priorities of the national-level innovative activity for 2017-2021" (2016), developing modern ICT and robotics, particularly cloud technologies, computer training systems and technologies of mathematical informatics (intellectual simulation, informational security, long-term data storage and "big data" management, artificial intelligence systems) are nationally and socially important directions of the innovative activity [28; 29]. The Decree of the Cabinet of Ministers of Ukraine "Certain issues of specifying medium-term priorities of the sectoral-level innovative activity for 2017-2021" (2017) specifies that these directions accompanied by smart web-technologies and cloud computing make the basis for creating and defining themes for scientific researches and technical (experimental) developments as well as for forming the state order of training ICT specialists.

For the past 25 years, the authors have been developing the concept of systematic computer simulation training at schools and teachers' training universities. The concept ideas have been generalized and presented in the textbook [53]. Spreadsheets are chosen to be the leading environment for computer simulation training, their application discussed in articles [47]. Using spreadsheet processors (autonomous, integrated and cloud-based) as examples, the authors demonstrate components of teaching technology of computer simulation of determined and stochastic objects and processes of various nature. The systematic training of simulation provides for changing and integrating simulation environments ranging from general (spreadsheets) to specialized subject-based ones. While teaching computer simulation of intellectual systems specialized languages and programming environments are traditionally used. They can be easily mastered by first-year students [1]. One of the most wide-spread languages, Scheme, is offered to be applied to teaching computer simulation of classical mechanics at universities [51]. Extensive application of artificial intelligence in everyday life calls for students' early acquaintance with its models

[*] Corresponding author: semerikov@gmail.com

and methods including neural network-based while teaching informatics at secondary schools [48]. It conditions the need for developing training methods of computer simulation of neural networks in the general-purpose simulation environment, i.e. spreadsheets.

2 Literature review and problem statement

The first description of spreadsheet application to teaching neural network simulation of visual phenomena dates back to 1985 and belongs to Thomas T. Hewett, Professor of the Department of Psychology of Drexel University [17]. In [16] there are de-scribed simple models of microelectrode recording of two neuron types of neural activity – receptors and transmitters localized in two brain-hemispheres. Thomas T. Hewett offered psychology students to independently choose coefficients of intensifying or reducing input impulses to achieve the desired output: “... the simulations can be designed in such a way that the student is able to "experiment" with a simulation-experiment both in the sense of discovering the characteristics of an unknown model and in the sense of modifying various components of a known model to see how the simulation is affected” [16, p. 343]. This approach implies simultaneous studying a neural network and understanding its functioning as psychology students conclude the laws of the neural impulse spread by applying the trial-and-error method.

In his article [8], James J. Buergermeister, Professor of Hospitality and Tourism Management of University Wisconsin-Stout, associates electronic spreadsheet application with basic principles of training technology and methods of data processing (Fig. 1). The author does not work out the methods of applying electronic spreadsheets to neural network simulation in detail, yet, the presented scheme reveals such basic steps as data obtainment, semantic coding, matching with an etalon, etc.

Since 1988, Murray A. Ruggiero, one of the pioneers of autotrading, has been developing Braincel, an application for Microsoft Excel 2.1C, which is a set of twenty macros to solve tasks of image recognition by artificial neural network tools [23]. At the beginning of 1991, Murray A. Ruggiero received a patent “Embedding neural networks into spreadsheet applications” [45], which describes an artificial neural net-work with a plurality of processing elements called neurons arranged in layers. They further include interconnections between the units of successive layers. A network has an input layer, an output layer, and one or more “hidden” layers in between, necessary to allow solutions of non-linear problems. Each unit (in some ways analogous to a biological neuron: dendrites – input layer, axon – output layer, synapses – weights [43], soma – summation function) is capable of generating an output signal which is determined by the weighted sum of input signals it receives and an activation function specific to that unit. A unit is provided with inputs, either from outside the network or from other units, and uses these to compute a linear or non-linear output. The unit’s output goes either to other units in subsequent layers or to outside the network. The input signals to each unit are weighted by factors derived in a learning process.

When the weight and activation function factors have been set to correct levels, a complex stimulus pattern at the input layer successively propagates between the hidden layers, to result in a simpler output pattern. The network is “taught” by feeding it a succession of input patterns and corresponding expected output patterns. The net-work “learns” by measuring the difference at each output unit between the expected output pattern and the pattern that it just produced. Having done this, the internal weights and activation functions are modified by a learning algorithm to provide an output pattern which most closely approximates the expected output pattern, while minimizing the error over the spectrum of input patterns. Neural network learning is an iterative process involving multiple lessons. Neural networks have the ability to process information in the presence of noisy or incomplete data and yet still generalize to the correct solution.

In his patent, Murray A. Ruggiero details a network structure (multi-level), an activation function (sigmoidal), a coding method (polar), etc. He presents a mathematical apparatus for network training and determines a method of data exchange between a spreadsheet processor nucleus and an add-in to it. The patent author suggests storing input data in columns, maximum and minimum values for each column of input data, the number of learning patterns. Data can be normalized or reduced to the polar range [0; 1] both in spreadsheets and add-ins.

In his article of 1989, Paul J. Werbos, the pioneer of the backpropagation method for artificial neural network training [55] demonstrates how to make the corresponding mathematical apparatus simpler to use it directly in the spreadsheet processor. The cycling character of training is supported by a macro that exchanges data between lines to avoid restrictions on the number of iterations because of the limited number of lines on a sheet of a separate spreadsheet. Some other authors suggest applying a similar approach of macros application [14; 57].

The authors of [24] in Chapter 2 “Neural Nets in Excel” give an example of applying the non-linear optimization tool, Microsoft Excel Solver, to forecasting stock prices using the “grey-box” concept, in which the model is evident, yet, the details of its realization are hidden.

In their article of 1998 [15], Tarek Hegazy and Amr Ayed from the Department of Civil Engineering at University of Waterloo distinguish the corresponding steps. Unlike [44], the authors suggest using bipolar data normalization (over the range of [–1; 1]) and a hyperbolic tangent as an activation function. Three addins for Microsoft Excel are used to determine weighting factors – the standard Solver and third-party add-ins (NeuroShell2 and GeneHunter by Ward Systems Group). Experiment results reveal that the best result is provided by the optimizing general-purpose tool (Solver) and not by specialized ones. In spite of the fact that “Journal of Construction Engineering and Management” does not refer to educational editions, the article [15] and the paper [7] by their structure and focus on details can be considered the first description of methodic of using spreadsheets for neural network simulation.

In their article of 2012 [43], Thomas F. Rienzo and Kuriakose K. Athappilly from Haworth College of Business at Western Michigan University consider model illustrating the process of machine learning as networks examine training data would provide another. Authors incorporate the stepwise learning processes of artificial neural network in a spreadsheet containing (1) a list or table of training data for binary input combinations, (2) rules for target outputs, (3) initial weight factors, (4) threshold values, (5) differences between target outputs and neural network transformation values, (6) learning rate factors, and (7) weight adjustment calculations. Unlike the previous ones, this model is invariant to the spreadsheet and does not call for applying any third-party addins.

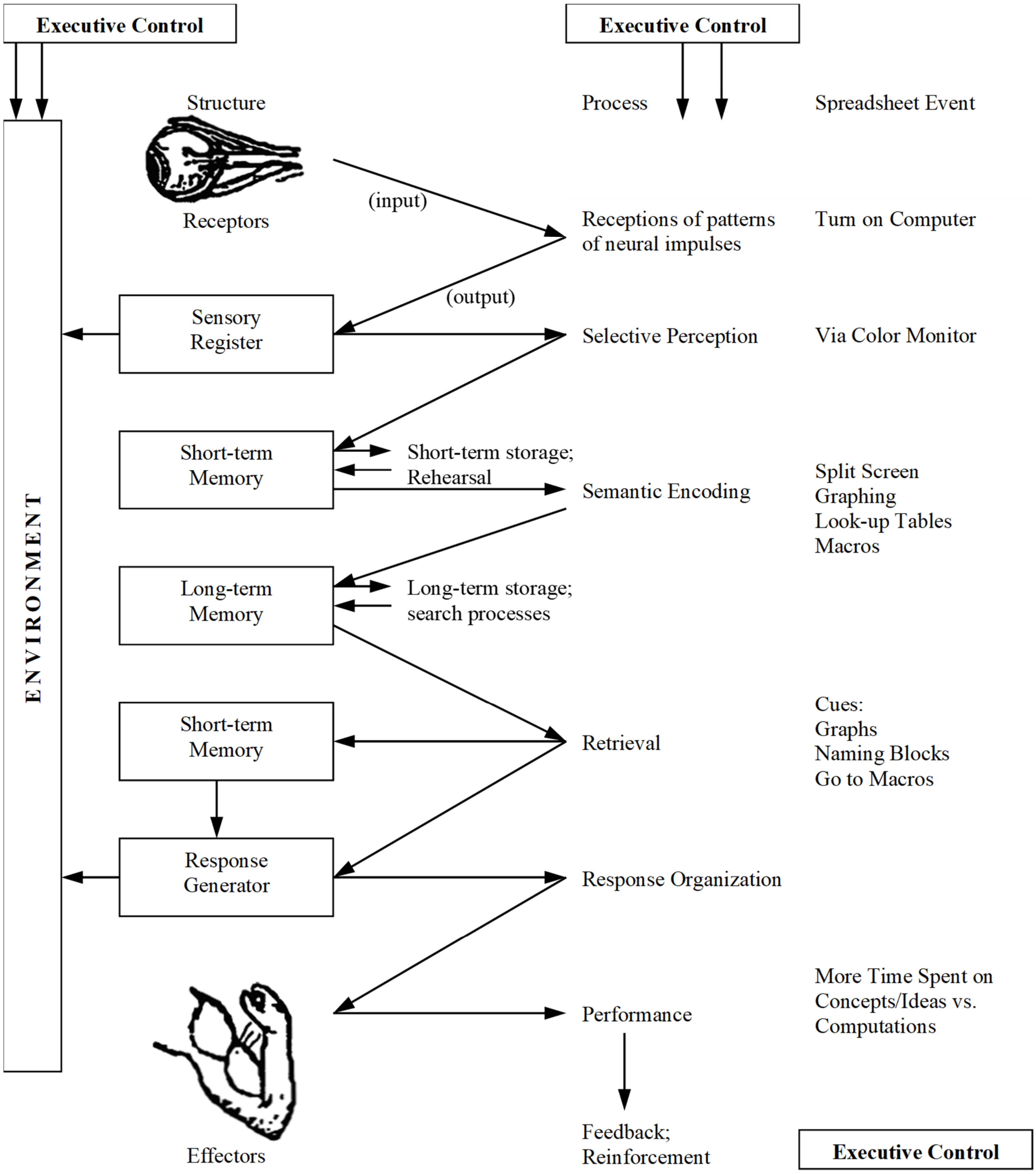


Fig. 1. The information-processing model using spreadsheet events (according to [8])

In [28] the role of neural network simulation in the training content of the special course "Foundations of Mathematical Informatics" is discussed. The course is developed for students of technical universities (future IT-specialists) and aimed at breaing a gap between theoretic computer science and its practical application to software, system and computing engineering. CoCalc is justified as a training tool for mathematical informatics in general and neural network modelling in particular. The elements of CoCalc techniques for studying the topic "Neural network and pattern recognition" within the special course "Foundations of Mathematical Informatics" are shown.

The authors of [47] distinguish basic approaches to solving the problem of network computer simulation training in the spreadsheet environment, joint application of spreadsheets and tools of neural network simulation. In [48], there are opportunities for applying spreadsheets to introducing essentials of machine learning [31] at secondary and higher school as well as some elements of their application to solving problems of pattern classification. Thus, using spreadsheets as a tool for teaching basics of machine learning creates conditions for early and simultaneously deeper mastering of corresponding models and methods of mathematical informatics [2].

The conducted review makes it possible to find the following solutions of the problem of computer simulation teaching to neural networks in the spreadsheet environment:

- joint application of spreadsheets and neural network tools [32], in which data is exported to the unit calculating weighting factors imported to spreadsheets and used in calculations;
- application of third-party add-ins for spreadsheets ([15; 23; 45]), according to which structured spreadsheet data is processed in the add-in, calculation results are arranged in spreadsheet cells;
- macros development ([7; 14; 55; 57]) enables direct software control over neural network training and creation of a user's specialized interface;
- application of standard add-ins for optimization ([15; 24]) calls for transparent network realization and determination of an optimization criterion (minimization of a squared deviation total of the calculated and etalon outputs of the network);
- creation of neural networks in the spreadsheet environment without add-ins and macros [43] requires transparent realization of a neural network with evident de-termination of each step of adjustment of its weighting factors.

The advantage of the first solution is its flexibility as one can choose any relevant combinations of the simulation environments, yet, their integration level is usually insufficient. The closed character of the second solution and its binding to a certain software platform make it relevant to be applied to solving various practical tasks and irrelevant for neural network simulation training as a network becomes a black box for a user. The fourth solution is partially platform-dependent as a neural network becomes a grey box for a user. The final solution is totally mobile and offers an opportunity to regard the model as a white box, thus making it the most relevant for initial mastering of neural network simulation methods.

3 The aim and objectives of the study

The research is aimed at considering mathematical models of neural networks realized in spreadsheet environment. To accomplish the set goal, the following tasks are to be solved: (1) to study historical models of neural networks; (2) to distinguish learning tools of computer simulation of neural networks in the spreadsheet environment; (3) to substantiate the chosen dataset to develop a model; (4) to develop a demonstration model of an artificial neural network using cloud-based spreadsheets.

4 Early neural networks models: from William James to Walter Pitts

Russell C. Eberhart and Roy W. Dobbins [12] suggest dividing the history of artificial network development into four stages. The first stage, the Age of Camelot, starts with "The Principles of Psychology" (1890) by the American psychologist, William James, who formulates the elementary law of association: "When two elementary brain processes have been active together or in immediate succession, one of them, on reoccurring, tends to propagate its excitement into the other" [22, p. 566]. The elementary law of association (the elementary principle) is closely related to the concepts of associative memory and correlational learning. In the authors' opinion [12], James seemed to foretell the notion of a neuron's activity being a function of the sum of its inputs, with past correlation history contributing to the weight of interconnections: "The amount of activity at any given point in the brain-cortex is the sum of the tendencies of all other points to discharge into it, such tendencies being proportionate (1) to the number of times the excitement of each other point may have accompanied that of the point in question; (2) to the intensity of such excitements; and (3) to the absence of any rival point functionally disconnected with the first point, into which the discharges might be diverted" [22, p. 567].

In "Psychology" (1892), an abridged re-edition of "The Principles of Psychology", James formulates basic principles of the image recognition theory: "We know, in short, a lot about it, whilst as yet we have no acquaintance with it. Our perception that one of the objects which turn up is, at last, our qucesitum, is due to our recognition that its relations are identical with those we had in mind, and this may be a rather slow act of judgment. Every one knows that an object may be for some time present to his mind before its relations to other matters are perceived. Just so the relations may be there before the object is." [21, p. 275].

"The Bulletin of Mathematical Biophysics" has been an advanced platform for approbating network models and methods since the moment of its foundation by Nicolas Rashevsky [11]. It should be no surprise as Rashevsky invented one of the first models of the neuron [40] and started the idea of artificial neural networks. The basic idea was to use a pair of linear differential equations and a nonlinear threshold operator:

$$\text{Input} = I(t)$$

$$\begin{cases} \dfrac{\partial e}{\partial t} = AI(t) - ae \\ \dfrac{\partial j}{\partial t} = BI(t) - bj \end{cases}$$

$$\text{Output} = H(e - j - \theta)$$

where θ is the threshold, e and j could represent excitation and inhibition or the amount or concentration of two substances within a neuron, H (x) is the Heaviside operator (takes positive values to 1, and non-positive values to 0). This gives an easy way to model the all-or-none firing of a neuron – Rashevsky showed that this simple model was able to model many of the known experimental results for the behavior of single neurons. He also made the point that networks of these model neurons could be connected to give quite complicated behavior and even serve as a model for a brain [11].

In his article of 1941 [56], Gale J. Young shows that the Rashevsky's two-factor model of nerve excitation can account for sustained inhibition or enhancement by a sequence of stimulus pulses, and for the decrease in the reinforcement period with each successive pulse of the train.

Developing Rashevsky's ideas, his student Alston Scott Householder, who gave his name to the known linear transformation describing a reflection about a plane or hyperplane containing the origin, and a class of root-finding algorithms used for functions of one real variable with continuous derivatives up to some order, in his article of 1940 [19], suggests a parameter measuring the "strength" of the inhibitory neurons acting among the terminal synapses. In [20], he describes the activity parameter as a characteristic of the fiber which is assumed to be different from zero, but it may be either positive (when the fiber is excitatory in character) or negative (when the fiber is inhibitory in character).

Thus, at the beginning of 1942, the theory of biological neural networks based on Rashevsky's continuous two-factor model was created and intensively developed. As remembered by J. A. Anderson and E. Rosenfeld, at the boundary of two decades, Walter Pitts was introduced to Nicolas Rashevsky by Rudolf Carnap, and accepted in to his mathematical biology group [10]. In his early publication, Pitts suggests "a new point of view in the theory of neuron networks is here adumbrated in its relation to the simple circuit: it is shown how these methods enable us to extend considerably and unify previous results for this case in a much simpler way" [36, p. 121]. With due consideration of Householder's articles, Pitts determines the total conduction time of a fiber as the sum of its conduction time and the synaptic delay at the postliminary synapse. Pitts was the first to use spreadsheet abstraction and discrete description of neural network functioning by determining a corresponding algorithm: Pitts was the first to use spreadsheet abstraction and discrete description of neural network functioning by determining a corresponding algorithm: "The excitation-pattern of [neural circuit] C may be described in a *matrix E*, of n rows and an infinite number of columns, each of whose elements e_{rs} represents the excitation at the synapse s_r during the interval (s, s+1). The successive entries in the excitation *matrix E* may be computed recursively from those in its first column – these are the quantities λ_r – by the following rule, whose validity is evident: Given the elements of the p-th column, compute those of the p+l-st thus: if the element e_{ip} is negative or zero, place σ_{i+l} in the i+l-st row and p+1-st column, or in the first row of the p+l-st column if i=n. Otherwise put σ_{i+l}+$a_i e_{ip}$, in this place. We shall say that C is in a *steady-state* during a series of n intervals (s, s+1), ..., (s+n–1, s+n) if, for every p between s and s+n, the p-th and p+n-th columns of E are identical. If s is the smallest integer for which this is the case, we shall say that the steady state begins at the interval (s, s+1)" [36, pp. 121–122]. The suggested algorithm describes a parallel neural network [36, p. 122]. Rather than analyzing the steady-state activity of networks, Pitts was more concerned with initial nonequilibrium cases, and how a steady state could be achieved [2, p. 18].

The results provided by Pitts in his articles on the linear theory of neuron networks (the static problem [38] and the dynamic problem [37]), enabled him to draw two essential conclusions: (1) it is possible to find a set of independent networks each of which consists of n simple circuits with one common synapse (*rosettes*), such that network arises by running chains from the centers of the rosettes to various designated points outside: but none back, so that the state of the whole network is determined by the states of the separate rosettes independently – Pitts calls networks of this kind *canonical networks* [37, p. 29]; (2) given any finite network, it is possible to find a set of independent rosettes such that the excitation function of network for every region is a linear combination of those of the rosettes – i. e., we can reduce any network to a canonical network having the same excitation function [37, p. 31]. Thus, in his article of 1943, Pitts solves the inverse network problem, "which is, given a preassigned pattern of activity over time, to construct when possible a neuron-network having this pattern" [37, p. 23] by allowing creating problem-oriented neural networks. Tara H. Abraham indicates that adopting Householder's model of neural excitation, Pitts develops a simpler procedure for the mathematical analysis of excitatory and inhibitory activity in a simple neuron circuit, and aimed to develop a model applicable to the most general neural network possible [2].

"Psychometrika", the official journal of the Psychometric Society (both founded in 1935 by Louis Leon Thurstone, Edward Lee Thorndike and Joy Paul Guilford), is devoted to the development of psychology as a quantitative rational science. It has become another mouthpiece of Nicolas Rashevsky and his students, whose articles examine statistical methods, discuss mathematical techniques, and advance theory for evaluating behavioral data in psychology, education, and the social and behavioral sciences generally. Pitts's article "A general theory of learning and conditioning" has been published in this journal. Part I [34] deals only with the case where the stimuli and responses are wholly independent, so that transfer and generalization do not occur, and proposes a law of variation for the reaction-tendency, which takes into ac-count all of classical conditioning and the various sorts of inhibition affecting it. Part II [35] extends a mathematical theory of non-symbolic learning and conditioning, still under the hypothesis of complete independence, to cases where reward and punishment are involved as motivating factors. The preceding results are generalized to the case where stimuli and responses are related psychophysically, thus constituting a theory of transfer, generalization, and discrimination.

Another article of 1943, "A logical calculus of the ideas immanent in nervous activity" [30], published again in "Bulletin of Mathematical Biophysics", has resulted from cooperation of Warren Sturgis McCulloch and Walter Pitts and is considered one of the most famous papers on artificial neural networks. They stated five physical assumptions for nets without circles [30, p. 118]:

1. The activity of the neuron is an "all-or-none" process [any nerve has a finite threshold and the intensity of excitation must exceed this for production of excitation – once produced, the excitation proceeds independently of the intensity of the stimulus].
2. A certain fixed number of synapses must be excited within the period of latent addition [time during which the neuron is able to detect the values present on its in-puts, the synapses – typically less than 0.25 msec] in order to excite a neuron at any time, and this number is independent of previous activity and position on the neuron.
3. The only significant delay within the nervous system is synaptic delay [time delay between sensing inputs and acting on them by transmitting an outgoing pulse, – typically less than 0.5 msec].
4. The activity of any inhibitory synapse absolutely prevents excitation of the neuron at that time.
5. The structure of the net does not change with time.

The neuron described by these five assumptions is known as the McCulloch-Pitts neuron [12, p. 17]. In the same way as propositions in propositional logic can be "true" or "false," neurons can be "on" or "off" – they either fire or they do not: this formal equivalence allowed them to argue that the relations among propositions can correspond to the relations among neurons, and that neuronal activity can be represented as a proposition [29, p. 19].

In [30], there is a set of theorems that "does in fact provide a very convenient and workable procedure for constructing nervous nets to order, for those cases where there is no reference to events indefinitely far in the past in the specification of the conditions" [30, pp. 121–122]. McCulloch and Pitts appear to be the first authors since William James to describe a massively parallel neural model. The theories they developed were important for a number of reasons, including the fact that any finite logical expression can be realized by networks of their neurons.

Combining simple "logical" neurons in chains and cycles, the authors show that the brain is able to perform any logical operation and arbitrary logical calculations. The paper is essential for developing computing machines as it allows creating a universal computer operating with logical expressions (in the hands of John von Neumann, the McCulloch-Pitts model becomes the basis for the logical design of digital computers [11, p. 180]): "It is easily shown: first, that every net, if furnished with a tape, scanners connected to afferents, and suitable efferents to perform the necessary motor-operations, can compute only such numbers as can a Turing machine; second, that each of the latter numbers can be computed by such a net; and that nets with circles can be computed by such a net; and that nets with circles can compute, without scanners and a tape, some of the numbers the machine can, but no others, and not all of them. This is of interest as affording a psychological justification of the Turing definition of computability and its equivalents, Church's λ-definability and Kleene's primitive recursiveness: If any number can be computed by an organism, it is computable by these definitions, and conversely." [30, pp. 121–122].

In the same issue of "Bulletin of Mathematical Biophysics", in which [30] was published, Herbert Daniel Landahl (the first doctoral student in Rashevsky's mathematical biology program at the University of Chicago, who became the second President of the Society for Mathematical Biology in 1981), Warren Sturgis McCulloch and Walter Pitts published a short (about 3 pages), yet essential addition [25], suggesting a method for converting logical relations among the actions of neurons in a net into statistical relations among the frequencies of their impulses. In the presented theorem, they detailed transition from Boolean calculations (in "true" and "false") to probabilistic ones (numbers within [0; 1]): the conjunction sign ∨ is replaced by +, the disjunction sign (single dot) is replaced by ×, negation ~ is replaced by «1 –», etc. The correspondence expressed by this theorem connects the logical calculus of the [30] with previous treatments of the activity of nervous nets in mathematical biophysics and with quantitatively measurable psychological phenomena.

The monograph by Householder and Landahl "Mathematical Biophysics of the Central Nervous System" has become a kind of conclusion of the discussed period [18]. In Paul Cull's opinion, there is no unambiguous answer to the question which model is better, the Rashevsky continuous model or the McCulloch-Pitts discrete model: "For some purposes, one model is better, but for other purposes, the other model is better. Rashevsky and Landahl were quick to notice, that in physics, one often averaged over a large set of discrete events to obtain a continuous model which represented the large scale behavior of a system, and so they posited that the continuous neuron model might be suitable for modeling whole masses of neurons even if each individual neuron obeyed a discrete model. In the hands of Householder and Landahl, this observation led to the idea of modeling psychological phenomena by neural nets with a small number of continuous model neurons. In particular, they found that the cross-couple connection [Fig. 2] was extremely useful. For such problems as reaction time, enhancement effects, flicker phenomena, apparent motion, discrimination and recognition, they were able to fit these models to experimental data and to use their models to predict phenomena that could be measured and verified" [11, p. 180].

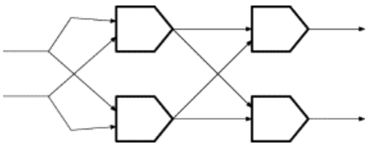


Fig. 2. This cross-couple connection of four neurons is capable of modeling a large number of phenomena (according to [46])

In 1945, Rashevsky wrote about [30] and [25]: "authors show that by applying logical calculus, it is possible to construct any complicated network having given properties. One could attempt to construct by the method of McCulloch and Pitts a network that would represent all modes of logical reasoning, and then apply the usual methods of mathematical biophysics to derive some quantitative relations between different manifestations of the processes of logical thinking" [39, p. 146]. "It seems somewhat awkward to have to construct by means of Boolean algebra first a "microscopic circuit" and then obtain a simpler one by a transition to the "macroscopic" picture. We should expect that a generalization of the application of Boolean algebra should be possible so as to permit its use for the construction of networks in which time relations are of a continuous rather than of a quantized, nature" [41, p. 211].

Rashevsky intensively develops the apparatus created by McCulloch and Pitts in his further papers. In [42] a theory of such neural circuits is developed which provide for formal logical thinking. Predicate apparatus application enables Rashevsky synthesizing huge neural networks from single-type fundamental elements of McCulloch-Pitts.

Telson Wei develops another approach to matrix representation of a neural net-work [54]. The structure of a complete or incomplete neural net is represented here by several matrices: the intensity matrix E, the connection matrix D, the structural matrix T, the diagonal inverse threshold-matrix H, and activity vector $\boldsymbol{a}$ from [26; 27].

In their paper [49], Alfonso Shimbel and Anatol Rapoport (pioneered in the modeling of parasitism and symbiosis, researching cybernetic theory) develop a probabilistic approach to the theory of neural nets: neural nets are characterized by certain parameters which give the probability distributions of different kinds of synaptic connections throughout the net. In their further papers, they consider steady states in random nets and contribution to the probabilistic theory of neural nets: randomization of refractory periods and of stimulus intervals, facilitation and threshold phenomena, specific inhibition and various models for inhibition.

The last joint article by Pitts and McCulloch, "How we know universals the perception of auditory and visual forms", in "Bulletin of Mathematical Biophysics" came out in 1947. "Numerous nets, embodied in special nervous structures, serve to classify information according to useful common characters. In vision they detect the equivalence of apparitions related by similarity and congruence, like those of a single physical thing seen from various places. In audition, they recognize timbre and chord, regardless of pitch. The equivalent apparitions in all cases share a common figure and define a group of transformations that take the equivalents into one another but pre-serve the figure invariant. So, for example, the group of translations removes a square appearing at one place to other places; but the figure of a square it leaves invariant. ... We seek general methods for designing nervous nets which recognize figures in such a way as to produce the same output for every input belonging to the figure. We endeavour particularly to find those which fit the histology and physiology of the actual structure." [33, pp. 127–128]

Thus, the models and methods developed by Pitts and McCulloch have created a foundation for designing a new type of computers – neurocomputers based on human brain principles and able to solve tasks of recognizing distorted (noisy) images.

5 Edgar Anderson and his Iris data set

Edgar Shannon Anderson (November 9, 1897 – June 18, 1969) was born in Forestville, New York. According to George Ledyard Stebbins, from an early age he exhibited both superior intelligence and a great interest in plants, particularly in cultivating them and watching them grow [50, p. 4].

He went to Michigan Agricultural College at the age of sixteen, just before his seventeenth birthday, knowing already that he wanted to be a botanist. After completing his degree, he accepted a graduate position at the Bussey Institution of Harvard University. After leaving Harvard with his doctor's degree in 1922, Anderson spent nine years at the Missouri Botanical Garden, where he was a geneticist and Director of the Henry Shaw School of Gardening; at the same time he was Assistant Professor, later Associate Professor, of Botany at Washington University in St. Louis. During this period, he developed the beginnings of his highly original and effective methods for looking at and recording variation in plant populations, as well as his keen interest in the needs and progress, both scientific and personal, of students in botany. His training in genetics had given him habits of precision and mathematical accuracy in ob-serving and recording variation in natural populations that were entirely foreign to the taxonomists of that period [50, p. 5].

Through contacts with Jesse Greenman, Curator of the Garden Herbarium, he became aware of the enormous complexity and extent of the variation present in any large plant genus and of the need for understanding the origin of species as a major step in evolution. On extensive field trips he began to realize that a great amount of genetic variation exists within most natural populations of plants. This realization led him to the conclusion that "if we are to learn anything about the ultimate nature of species we must reduce the problem to the simplest terms and study a few easily recognized, well differentiated species" [6, p. 243].

He first selected *Iris versicolor*, the common blue flag, because he believed it to be clearly defined, and it was common and easily observed. Initially, this appeared to be a mistaken choice, since he soon found that *Iris versicolor* of the taxonomic manuals was actually two species, which, after preliminary analysis, he could easily tell apart. He then set himself the task of finding out, by a careful analysis of populations throughout their geographic areas, how one of these species could have evolved from the other. He recorded several morphological characters in more than 2,000 individuals belonging to 100 populations, data far more extensive than those that any botanist had yet obtained on a single species.

In order to enable these data to be easily visualized and compared, he constructed the first of his highly original and extremely useful series of simplified diagrams or ideographs (Fig. 3). By examining them, he reached the conclusion that the variation within each of his two species was of another order from the differences between them; no population of one species could be imagined as the beginning of a course of evolution toward the other. He therefore concluded that speciation in this example was not a continuation of the variation that gave rise to differences between populations of one species, and started to look for other ways in which it could have taken place. The current literature offered a possible explanation: hybridization followed by chromosome doubling to produce a fertile, stable, true-breeding amphidiploid. To apply this concept to *Iris*, he had to find a third species that would provide an alter-nate parent for one of those studied. Going to the herbarium, he found it: an undescribed variety of *Iris setosa*, native to Alaska.

All of his data, including counts of chromosome numbers, agreed with the hypothesis that *Iris versicolor* of northeastern North America had arisen as an amphiploid, one parent being *Iris virginica* of the Mississippi Valley and the Southeast Coast and the other being *Iris setosa* var. interior of the Yukon Valley, Alaska. This was one of the earliest demonstrations that a plant species can evolve by hybridization accompanied or followed by chromosome doubling. Moreover, it was the first one to show that amphiploid or allopolyploid species could be used to support hypotheses about previous distribution of species.

Anderson's research into Iris resulted in all the techniques in his later successful work, namely:

1. careful examination of individual characteristics of plants growing in nature and progeny raised in the garden;
2. reduction of this variation to easily visualized, simple terms by means of scatter diagrams and ideographs;
3. extrapolation from a putative parental species and supposed hybrids to reconstruct the alternative parent;
4. development of testable hypotheses by synthesizing data from every possible source.

The *Iris* research was Anderson's chief accomplishment during his first period at the Missouri Botanical Garden. Toward the end of this period, in 1929-1930, he received a National Research Fellowship to study in England. There he was guided chiefly by geneticist J. B. S. Haldane, but he also studied cytology under C. D. Darlington and statistics with R. A. Fisher. Haldane introduced him to the mutants of *Primula sinensis*, which he analyzed in collaboration with Dorothea De Winton. Their joint research was the first effort in plant material to relate pleiotropic gene action to growth processes.

In 1931 Anderson went to Harvard, where he stayed until 1935, as an arborist at the Arnold Arboretum. He returned to the Missouri Botanical Garden in 1935 and remained there for the rest of his life. Returning to his study of the genus Iris, he and several students analyzed a complex variation pattern of populations found in the Mississippi delta region [5].

Anderson integrated his new experience with past memories, popular accounts of his methods of research, and his general philosophy of life in the book "Plants, Man and Life" [4] published in 1952. It is a combination of scientific knowledge, folklore of Latin American and other countries, and Anderson's comments on early herbalists and the habits of taxonomists and botany professors, plus a bit of philosophy. One of his chief contributions to plant science, the pictorialized scatter diagram, is presented for the first time in its final form in a chapter entitled, characteristically, "How to Measure an Avocado" (Fig. 3).

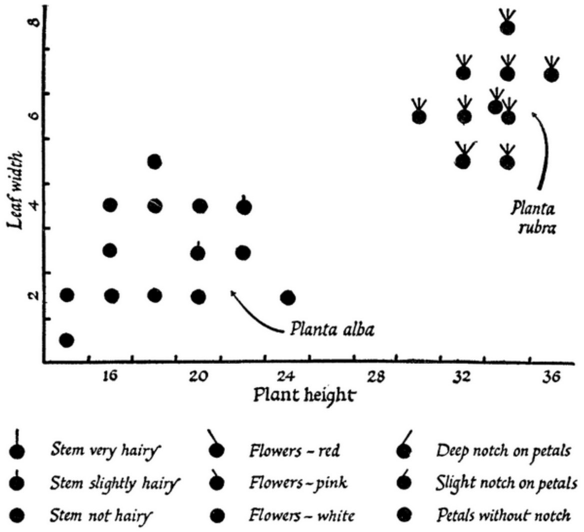


Fig. 3. Anderson's pictorialized scatter diagram [4, p. 97]

Anderson's article of 1936 [3] was his last work dedicated to the problem of *Iris* origin and classification. In his introduction to the article, Anderson not only expressed his gratitude to his English teachers, but also directly indicated that "Dr. Wright, Prof. J. B. S. Haldane, and Dr. R. A. Fisher have greatly furthered the final analysis of the data, though they are in no way responsible for the imperfections of the work or of its presentation." [3, p. 458].

In 1936, Sir Ronald Aylmer Fisher published the article "The Use of Multiple Measurements in Taxonomic Problems" indicating that "Table I shows measurements of the flowers of fifty plants each of the two species *Iris setosa* and *I. versicolor*, found growing together in the same colony and measured by Dr E. Anderson, to whom I am indebted for the use of the data" [13, p. 179-180]. Fisher's article contained only three references two of which to Anderson's works – that of 1935 [5] and that of [3] marked with "(in the Press)". In 1936, Fisher was not the member of the editorial board of "Annals of the Missouri Botanical Garden". The only way of his being aware of Anderson's article [3], was their personal correspondence.

The set of data used by Fisher and collected by Anderson was introduced as “Iris flower data set” (or “Iris data set” and “Iris data”). The phrase “Fisher’s Iris data set” traditionally expresses Fisher’s role as the founder of linear discriminant analysis, but not the authorship of the data set.

Although Anderson never published these data, he described [5] how he collected information on irises: “For some years I have been studying variation in irises but never before have I had the good fortune to meet such quantities of material for observation. On the simple assumption that if current theories are true, one should be able to find evidence of continuing evolution in any group of plants, I have been going around the world looking as sharply as possible at variation in irises. On any theory of evolution the differences between individuals get somehow built up, in time, into the differences between species. That is to say that by one process or another the differences which exist between one plant of *Iris versicolor* and its neighbor are com-pounded into the greater difference which distinguishes *Iris versicolor* from *Iris setosa canadensis*. It is a convenient theory and if it is true, we should be able to find the beginnings of such a compounding going on in our present day species. For that reason I have studied such irises as I could get to see, in as great detail as possible, measuring iris standard after iris standard and iris fall after iris fall, sitting squat-legged with record book and ruler in mountain meadows, in cypress swamps, on lake beaches, and in English parks. The result is still merely a ten year’s harvest of dry statistics, only partially winnowed and just beginning to shape itself into generalizations which permit of summarization and the building of a few new theories to test by other means.

I have found no other opportunity quite like the field from De Verte to Trois Pistoles. There for mile after mile one could gather irises at will and assemble for comparison one hundred full-blown flowers of *Iris versicolor* and of *Iris setosa canadensis*, each from a different plant, but all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus. The result is, to ordinary eyes, a few pages of singularly dry statistics, but to the bio-mathematician a juicy morsel quite worth looking ten years to find.

After which rhapsody on the beauty of variation it must immediately be emphasized that *Iris setosa canadensis* varies but little in comparison with our other native blue flags. *Iris versicolor* in any New England pastures may produce ground colors all the way from mauve to blue and with hafts white or greenish or even sometimes quite a bright yellow at the juncture with the blade. *Iris setosa canadensis* by contrast is prevailingly uniform, its customary blue grey occasionally becoming a little lighter or a little darker or even a little more towards the purple, and its tiny petals producing odd variants in form and pattern, but presenting on the whole only a fraction of the variability of *Iris versicolor* from the same pasture.

The reasons for this uniformity are not far to seek. Its lower chromosome number is one, but a discussion of that and its bearings on the whole problem would be a treatise in itself. More important probably is the fact that by geological and biological evidence, *Iris setosa canadensis* is most certainly a remnant, a relict *[sic]* of what was before the glacial period a species widely spread in northern North America.

If we take a map and plot thereon all known occurrences of *Iris setosa* and *Iris setosa canadensis*, we shall find the former growing over a large area at the northwest comer of the continent, and the latter clustering in a fairly restricted circle about the Gulf of St. Lawrence, while in the great intervening stretch of territory, none of these irises has been collected. This is a characteristic distribution for plants which were almost exterminated from eastern North America by the continental ice sheet, but while *[sic]* managed to persist in the unglaciated areas about the Gulf of St. Lawrence from which center they have later spread. In Alaska the species itself, *Iris setosa*, is apparently quite as variable as our other American irises.”

So, we should pay tribute to Edgar Anderson by naming this data set after him – Anderson’s Iris data set.

6 Model development

As indicated in [28], the special course “Foundations of Mathematical Informatics” final control of knowledge is a credit by the presentation of individual education and research projects on the artificial neural networks built by using CoCalc. Students can be offered to use cloud-based spreadsheets, Google Sheets, with the Solver additional cloud-based component (add-in) which is similar to “Solver” in Excel Online.

Let us consider the corresponding application method by taking a Anderson’s Iris data set to solve the pattern classification problem. Anderson’s Iris is composed of data on 150 measurements of three Iris species (Fig. 4) – *Iris setosa*, *Iris virginica* and *Iris versicolor*) – including 50 measurements for each species.

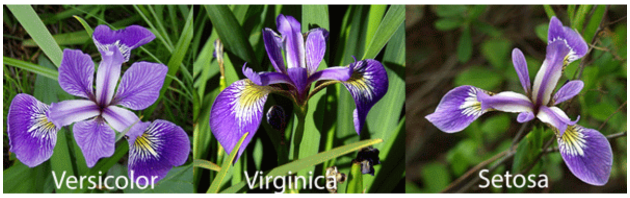


Fig. 4. Anderson’s Irises

There were measured four features (Fig. 5): sepal length (SL), sepal width (SW), petal length (PL), and petal width (PW).

To draw a grounded conclusion on the *Iris* type, we build a three-layered neural network with the following architecture (Fig. 6):

- the input layer is a four-dimensional arithmetical vector (x_1, x_2, x_3, x_4) the components of which are corresponding measured features of Anderson’s Irises (SL, SW, PL, PW) normalized according to the network activation function;
- the hidden layer has dimension 9 (the minimal required number according to Kolmogorov–Arnold representation theorem) and is described by the vector (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9);

- the output layer is a three-dimensional arithmetical vector (y_1, y_2, y_3) the components of which are probabilities indicating the correspondence of the data set to one of the three *Iris* types.

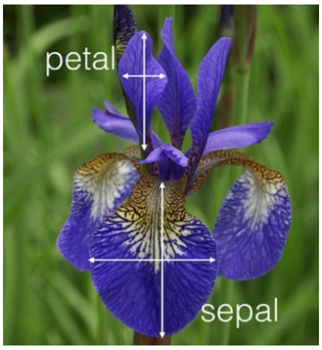


Fig. 5. Measurement features of Anderson's Irises

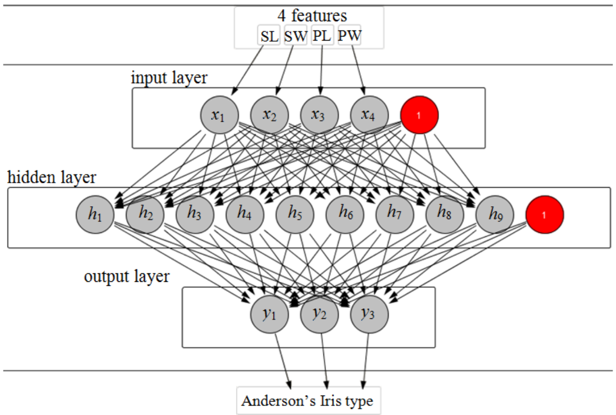
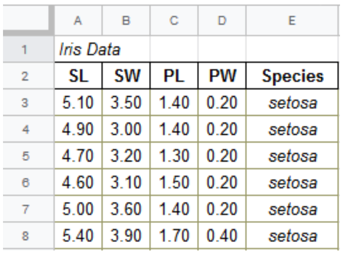


Fig. 6. Architecture of the neural network to solve the problem of Anderson's Iris classification

The bias neuron equal to 1 (marked red in Fig. 6) is added to the neurons of the input and hidden layers. The bias neurons are noted for not having synapses so they cannot be located in the output layer.

Let us first introduce Anderson's Irises into spreadsheets with the following values of cells: A1 is `Iris Data`, A2 is `SL`, B2 is `SW`, C2 is `PL`, D2 is `PW`, E2 is `Species`.

The table cells A3:E152 include Anderson's Irises (Fig. 7).



	A	B	C	D	E
1	*Iris Data*				
2	SL	SW	PL	PW	Species
3	5.10	3.50	1.40	0.20	*setosa*
4	4.90	3.00	1.40	0.20	*setosa*
5	4.70	3.20	1.30	0.20	*setosa*
6	4.60	3.10	1.50	0.20	*setosa*
7	5.00	3.60	1.40	0.20	*setosa*
8	5.40	3.90	1.70	0.40	*setosa*

Fig. 7. The fragment of the spreadsheet of Anderson's Irises

We cannot input the data of the given set into the input layer as the value of the four characteristics is beyond the range limits [0; 1]. The next step is normalization of columns A, B, C and D to meet the given range and coding of *Iris* types from column E.

Each *Iris* type is coded by the three-dimensional arithmetical vector: for *i*-Iris (*Iris setosa* is 1, *Iris versicolor* is 2, *Iris virginica* is 3) we set the *i*-th component in 1, and the other ones – in 0. To do this, we introduce the following values into the cells: G1 is `encoding`, G2 is `setosa`, H2 is `versicolor`, I2 is `virginica`, G3 is `=if($E3=G$2,1,0)`.

Next, we copy the formula from the cell G3 to the range G3:I152 and obtain the following model codes for the three *Iris* types: for *Iris setosa* – (1, 0, 0), for *Iris virginica* – (0, 0, 1) and for *Iris versicolor* – (0, 1, 0).

Each column is normalized separately. To perform this, we find minimum and maximum values by introducing the following values: E154 is `min`, E155 is `max`, A154 is `=min(A3:A152)`, A155 is `=max(A3:A152)`.

We apply the cells A154:A155 to the range B154:D155 and introduce the following values into the cells: K1 is `normalization`, K2 is x_1, L2 is x_2, M2 is x_3, N2 is x_4, K3 is `=(A3-A$154)/(A$155-A$154)`.

The latter formula is applied to the range K3:N152. Its essence is explained by:

$$\text{normalization} = \frac{\text{value} - \text{min}}{\text{max} - \text{min}}.$$

This approach results in the minimum value normalized to 0, while the maximum one – to 1.

According to the chosen architecture, we add the bias neuron to the four neurons of the input layer by introducing its name (x_5) into the cell O2 and its value (1) into the range O3:O152. On this stage, the input layer is formed as x_1, x_2, x_3, x_4, x_5.

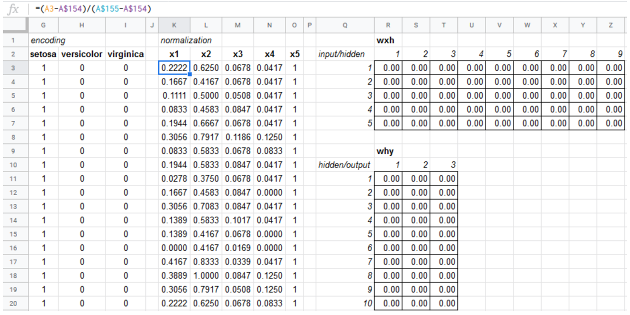
The next step includes transmission of a signal from the input layer to the hidden one of the neural network. We denote the weight coefficient of the synapse connecting the neuron x_i ($i = 1, 2, 3, 4, 5$) of the input layer with the neuron h_j ($j = 1, 2, ..., 9$) of the hidden layer by $w^{xh}{}_{ij}$, while the weight coefficient connecting the neuron h_j of the hidden layer with the neuron y_k ($k = 1, 2, 3$) of the input layer is denoted by $w^{hy}{}_{jk}$. In this case, the force of the signal coming to the neuron h_j of the hidden layer is determined as a scalar product of signal values on the input signals and corresponding weight coefficients. To determine a signal going further to the output layer, we apply the logistic function of activation $f(S) = 1/(1+e^{-S})$, where S is a scalar product. The formulae for determining the signals on the hidden and output layers will look like:

$$h_j = f\left(\sum_{i=1}^{4+1} x_i w_{i\ j}^{x\,\mathbf{h}}\right), y_k = f\left(\sum_{j=1}^{9+1} x_i w_{j\ k}^{\mathbf{h}\,y}\right).$$

Accordingly, two matrices should be created. The matrix w^{xh} of 5×9 contains weight coefficients connecting five neurons of the input layer (the first four contain normalized characteristics of Anderson's Irises, while the fifth one is the bias neuron) with the neurons of the hidden layer. The matrix w^{hy} of 10×3 contains weight coefficients connecting ten neurons of the hidden layer (nine of which are calculated and the tenth one is the bias neuron) with the neurons of the output layer. For the "untaught" neural network, initial values of the weight coefficients can be set either randomly or left undetermined or equal to zero. To realize the latter, we fill the cells with the following values: R1 is `wxh`, Q2 is `input/hidden`, R2 is `1`, S2 is `=R2+1`, Q3 is `1`, Q4 is `=Q3+1`, R3 is `0`, R9 is `why`, Q10 is `hidden/output`, R10 is `1`, S10 is `=R10+1`, Q11 is `1`, Q12 is `=Q11+1`, R11 is `0`.

To create the matrices, we should copy the cells R3 into the range R3:Z7, R11 – into R11:T20, S2 – into T2:Z2, Q4 – into Q5:Q7, S10 – into T10, Q12 – into Q13:Q20 (Fig. 8).

To calculate the scalar product of the vector row of the input layer values by the matrix vector-column of the weight coefficients w^{hy}, we should apply the matrix multiplication function: AB1 is `calculate the hidden layer`, AB2 is h_1, AC2 is h_2, AD2 is h_3, AE2 is h_4, AF2 is h_5, AG2 is h_6, AH2 is h_7, AI2 is h_8, AJ2 is h_9, AK2 is h_{10}, AB3 is `=1/(1+exp(-mmult($K3:$O3,R$3:R$7)))`, AK3 is `1`.



`=(A3-A$154)/(A$155-A$154)`

	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	*encoding*				*normalization*							**wxh**								
2	**setosa**	**versicolor**	**virginica**		**x1**	**x2**	**x3**	**x4**	**x5**		*input/hidden*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
3	1	0	0		0.2222	0.6250	0.0678	0.0417	1		*1*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	1	0	0		0.1667	0.4167	0.0678	0.0417	1		*2*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	1	0	0		0.1111	0.5000	0.0508	0.0417	1		*3*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	1	0	0		0.0833	0.4583	0.0847	0.0417	1		*4*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	1	0	0		0.1944	0.6667	0.0678	0.0417	1		*5*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	1	0	0		0.3056	0.7917	0.1186	0.1250	1											
9	1	0	0		0.0833	0.5833	0.0678	0.0833	1			**why**								
10	1	0	0		0.1944	0.5833	0.0847	0.0417	1		*hidden/output*	*1*	*2*	*3*						
11	1	0	0		0.0278	0.3750	0.0678	0.0417	1		*1*	0.00	0.00	0.00						
12	1	0	0		0.1667	0.4583	0.0847	0.0000	1		*2*	0.00	0.00	0.00						
13	1	0	0		0.3056	0.7083	0.0847	0.0417	1		*3*	0.00	0.00	0.00						
14	1	0	0		0.1389	0.5833	0.1017	0.0417	1		*4*	0.00	0.00	0.00						
15	1	0	0		0.1389	0.4167	0.0678	0.0000	1		*5*	0.00	0.00	0.00						
16	1	0	0		0.0000	0.4167	0.0169	0.0000	1		*6*	0.00	0.00	0.00						
17	1	0	0		0.4167	0.8333	0.0339	0.0417	1		*7*	0.00	0.00	0.00						
18	1	0	0		0.3889	1.0000	0.0847	0.1250	1		*8*	0.00	0.00	0.00						
19	1	0	0		0.3056	0.7917	0.0508	0.1250	1		*9*	0.00	0.00	0.00						
20	1	0	0		0.2222	0.6250	0.0678	0.0833	1		*10*	0.00	0.00	0.00						

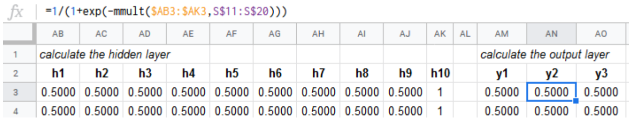
Fig. 8. The fragment of the spreadsheet after coding and normalization of the output data and creation of the matrices of the weight coefficients

Next, we copy the cell AK3 into the range AK4:AK152, while AB3 – into AB3:AJ152.

Considering the fact that all the matrix elements of the weight coefficients w^{xh} equal to zero, after duplicating the formulae, the calculated elements of the hidden layer will be equal to 0.5.

In the same way, we calculate the output layer elements: AM1 is `calculate the output layer`, AM2 is y_1, AN2 is y_2, AO2 is y_3, AM3 is `=1/(1+exp(-mmult($AB3:$AK3,R$11:R$20)))`.

Next, we copy the cell AM3 to the range AM3:AO152 (Fig. 9).



`=1/(1+exp(-mmult($AB3:$AK3,S$11:S$20)))`

	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO
1	*calculate the hidden layer*											*calculate the output layer*		
2	**h1**	**h2**	**h3**	**h4**	**h5**	**h6**	**h7**	**h8**	**h9**	**h10**		**y1**	**y2**	**y3**
3	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	1		0.5000	0.5000	0.5000
4	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	1		0.5000	0.5000	0.5000

Fig. 9. The fragment of the spreadsheet of calculating the hidden and output layers with initial values of the weight coefficients

Neural network training is performed by varying weight coefficients so that with each training step the difference between the calculated values of the output layer and the desired (reference ones) reduces. To solve the problem, the three-dimensional vectors resulted from coding of the three *Iris* types are reference.

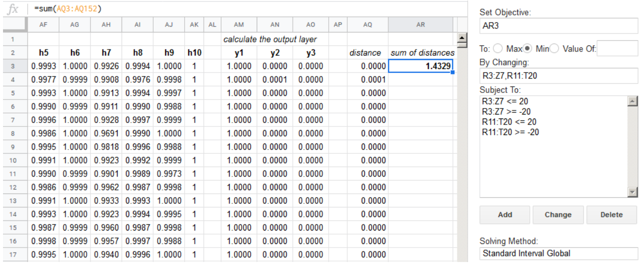
To find the difference between the calculated and the reference output vectors we apply the Euclidean distance: AQ2 is `distance`, AR2 is `sum of distances`, AQ3 is `=sqrt((AM3-G3)^2+(AN3-H3)^2+(AO3-I3)^2)`, AR3 is `=sum(AQ3:AQ152)`.

Next, we copy the cell AQ3 to the range AQ4:AQ152. The cell AR3 contains general deviation of the calculated output vectors from the reference ones.

Under this approach, the neural network training can be treated as an optimization problem in which the target function (the sum of distances in the cell AR3) will be minimized by varying the matrix weight coefficients w^{xh} (the range R3:Z7) and w^{hy} (the range R11:T20). To solve this problem, application of cloud-based spreadsheets (Google Sheets) is not enough and it is necessary to install an additional cloud-based component (add-in) Solver.

Adjustment of the add-in Solver to solve the set goal: the target function (Set Objective) is minimized (To: Min) by changing the values (By Changing) of the matrix weight coefficients in the range (Subject To) from –10 to +10 by one of the optimization methods (Solving Method).

To reduce the total distances, the actions with Solver can be done repeatedly as it is expedient to experiment with combination of various optimization methods by changing the variation limits of the weight coefficients. It is not necessary to try to reduce the value of the total distances to zero as this can be a greater (quite smaller) value (Fig. 10).



`=sum(AQ3:AQ152)`

	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR
1								*calculate the output layer*					
2	**h5**	**h6**	**h7**	**h8**	**h9**	**h10**		**y1**	**y2**	**y3**		*distance*	*sum of distances*
3	0.9993	1.0000	0.9926	0.9994	1.0000	1		1.0000	0.0000	0.0000		0.0000	**1.4329**
4	0.9977	0.9999	0.9908	0.9976	0.9998	1		1.0000	0.0001	0.0000		0.0001	
5	0.9993	1.0000	0.9913	0.9994	0.9997	1		1.0000	0.0000	0.0000		0.0000	
6	0.9990	0.9999	0.9911	0.9990	0.9988	1		1.0000	0.0000	0.0000		0.0000	
7	0.9996	1.0000	0.9928	0.9997	0.9999	1		1.0000	0.0000	0.0000		0.0000	
8	0.9986	1.0000	0.9691	0.9990	1.0000	1		1.0000	0.0000	0.0000		0.0000	
9	0.9995	1.0000	0.9818	0.9996	0.9988	1		1.0000	0.0000	0.0000		0.0000	
10	0.9991	1.0000	0.9923	0.9992	0.9999	1		1.0000	0.0000	0.0000		0.0000	
11	0.9990	0.9999	0.9901	0.9989	0.9973	1		1.0000	0.0000	0.0000		0.0000	
12	0.9986	0.9999	0.9962	0.9987	0.9998	1		1.0000	0.0000	0.0000		0.0000	
13	0.9991	1.0000	0.9933	0.9993	1.0000	1		1.0000	0.0000	0.0000		0.0000	
14	0.9993	1.0000	0.9923	0.9994	0.9995	1		1.0000	0.0000	0.0000		0.0000	
15	0.9987	0.9999	0.9960	0.9987	0.9998	1		1.0000	0.0000	0.0000		0.0000	
16	0.9998	0.9999	0.9957	0.9997	0.9988	1		1.0000	0.0000	0.0000		0.0000	
17	0.9995	1.0000	0.9940	0.9996	1.0000	1		1.0000	0.0000	0.0000		0.0000	

Set Objective: AR3
To: ○ Max ● Min ○ Value Of:
By Changing: R3:Z7,R11:T20
Subject To:
R3:Z7 <= 20
R3:Z7 >= -20
R11:T20 <= 20
R11:T20 >= -20
Solving Method: Standard Interval Global

Fig. 10. Optimization results

On the assumption of the chosen coding method, the output vector actually contains three probabilities: y_i denotes the probability of the given sample being the *i*-type Iris, where $i = 1$ for *Iris setosa*, 2 for *Iris versicolor* and 3 for *Iris virginica*. Then, to find out which *Iris* type describes the input vector (SL, SW, PL, PW), the most probable component should be determined.

To do this, we fill the cells in the following way: AT2 is `Calculated Iris species`, AT3 is `=if(max(AM3:AO3)=AM3,$G$2,if(max(AM3:AO3)=AN3,$H$2, $I$2))`, AU3 is `=if(AT3=E3, "right!", "wrong")`.

Next, the range AT3:AU3 is copied to the range AT4:AU152.

The obtained result enables us to visualize pattern recognition simulated in spread-sheets. The built model will be considered relevant in all 150 cases, the column AU contains the value "right!".

To check the limits of the built model application, we try to input the vector not coinciding with any reference input vector. For this, we copy the table row 152 to 158 and delete the content of the cells E158:I158, AQ158, AU158. We introduce averaged values borrowed from the description of *Iris versicolor* in the article by Anderson [33, p. 463]: 5.50, 2.75, 3.50 and 1.25. The reference values $x_1 = 0.3333$, $x_2 = 0.3125$, $x_3 = 0.4237$, $x_4 = 0.4792$ are conveyed to the input layer, while on the hidden layer there are calculated $h_1 - h_9$ and the values of the output layer $y_1 = 0.0000$, $y_2 = 1.0000$, $y_3 = 0.0000$. As the maximum value of the output layer 1.0000 corresponds to the other *Iris* type, we can conclude that *Iris versicolor* is identified.

7 Conclusions

1. Extensive application of artificial intelligence in everyday life calls for students' early acquaintance with its models and methods including neural network-based while teaching informatics at secondary schools. It conditions the need for developing training methods of computer simulation of neural networks in the general-purpose simulation environment, i.e. spreadsheets.
2. Basic solutions of the problem of computer simulation training of neural networks in the spreadsheet environment include: 1) joint application of spreadsheets and network simulation tools; 2) application of third-party add-ins to spreadsheet processors; 3) macros development using embedded languages of spreadsheet processors; 4) application of standard spreadsheet add-ins for non-linear optimization; 5) creation of neural networks in the spreadsheet environment without add-ins and macros.
3. Neural network simulation competences should be formed through mastering models based on the historical and genetic approach. The review of papers on computational neuroscience of its early period allows determining three groups of models, which are helpful for developing corresponding methods: the continuous two-factor model of Rashevsky, the discrete model of McCulloch and Pitts, and the discrete-continuous models of Householder and Landahl.
4. Edgar Anderson appeared to be not a simple botanist whose data were the basis for Fisher's known method. Anderson's Irises resulted from his long experience of working out relevant models to describe changes in specific populations by means of a limited number of characteristics. Yet, Anderson had also coped with the opposite problem of building simple multi-dimensional data interpretation 40 years before Chernoff faces appeared [9].
5. The described methods of applying cloud-based spreadsheets as a tools for training mathematical informatics can enable solution of all basic problems of neural net-work simulation. The only limitation is not so much the volume of a spreadsheet as the memory space and the speed of the device processing it. In the special course projects if the limitation is overcome, this becomes a stimulus for replacing the simulation environment by a more relevant one [52].

References

1. H. Abelson, G.J. Sussman, J. Sussman, *Structure and Interpretation of Computer Programs*, 2nd edn. (MIT Press, Cambridge, 1996)
2. T.H. Abraham, (Physio)logical circuits: The intellectual origins of the McCulloch-Pitts neural networks. Journal of the History of the Behavioral Sciences. **38**(1), 3–25 (2002). doi:10.1002/jhbs.1094
3. E. Anderson, The Species Problem in Iris. Annals of the Missouri Botanical Garden. **23**(3), 457–469+471–483+485–501+503–509 (1936). doi:10.2307/2394164.
4. E. Anderson, *Plants, Man and Life* (University of California Press, Boston, 1952)
5. E. Anderson, Bulletin of the American Iris Society. **59**, 2–5 (1935)
6. E. Anderson, The Problem of Species in the Northern Blue Flags, Iris versicolor L. and Iris virginica L. Annals of the Missouri Botanical Garden. **15**(3), 241–332 (1928). doi:10.2307/2394087
7. A.S. Ayed, Master thesis, Memorial University, 1997
8. J.J. Buergermeister, in Restructuring Training and Education through Technology, ed. by D.W. Dalton. 32nd Annual Conference of the Association for the Development of Computer-Based Instructional Systems, San Diego, California, October 29-November 1, 1990. (ADCIS International, Columbus, 1990), pp. 214–220
9. H. Chernoff, Journal of the American Statistical Association. **68**(342), 361–368 (1973)
10. J.D. Cowan in *Talking nets: An oral history of neural networks*, ed. by J.A. Anderson, E. Rosenfeld (MIT Press, Cambridge, 1998), pp. 97–124
11. P. Cull, The mathematical biophysics of Nicolas Rashevsky. BioSystems. **88**(3), 178–184 (2007). doi:10.1016/j.biosystems.2006.11.003
12. R.C. Eberhart, R.W. Dobbins, in *Neural Network PC Tools: A Practical Guide*, ed. by R.C. Eberhart, R.W. Dobbins (Academic Press, San Diego, 1990), pp. 9–34
13. R.A. Fisher, The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics. **7**(2), 179–188 (1936). doi:10.1111/j.1469-1809.1936.tb02137.x
14. R.S. Freedman, R.P. Frail, F.T. Schneider, B. Schnitta, in *Proceedings First International Conference on Artificial Intelligence Applications on Wall Street*, Institute of Electrical and Electronics Engineers, New York, 9–11 Oct. 1991
15. T. Hegazy, A. Ayed, Neural Network Model for Parametric Cost Estimation of Highway Projects. Journal of Construction Engineering and Management. **124**(3), 210–218 (1998). doi:10.1061/(ASCE)0733-9364(1998)124:3(210)
16. T.T. Hewett, Teaching Students to Model Neural Circuits and Neural Networks Using an Electronic Spreadsheet Simulator. Behavior Research Methods,

Instruments, & Computers. **17**(2), 339–344 (1985). doi:10.3758/BF03214406

17. T.T. Hewett, *Using an Electronic Spreadsheet Simulator to Teach Neural Modeling of Visual Phenomena*. (Drexel University, Philadelphia, 1985)
18. A.S. Householder, H.D. Landahl, *Mathematical Biophysics of the Central Nervous System* (Principia Press, Bloomington, 1945)
19. A.S. Householder, A neural mechanism for discrimination: II. Discrimination of weights. Bulletin of Mathematical Biophysics. **2**(1), 1–13 (1940). doi:10.1007/BF02478027
20. A.S. Householder, A theory of steady-state activity in nerve-fiber networks I: Definitions and Preliminary Lemmas. Bulletin of Mathematical Biophysics. **3**(2), 63–69 (1941). doi:10.1007/BF02478220
21. W. James, *Psychology* (Henry Holt and Company, New York, 1892)
22. W. James, *The Principles of Psychology* (Henry Holt and Company, New York, 1890)
23. S.J. Johnston, InfoWorld. **13**(7), 14 (1991)
24. D.A. Kendrick, P.R. Mercado, H.M. Amman, *Computational Economics* (Princeton University Press, Princeton, 2006)
25. H.D. Landahl, W.S. McCulloch, W. Pitts, A statistical consequence of the logical calculus of nervous nets. Bulletin of Mathematical Biophysics. **5**(4), 135–137 (1943). doi:10.1007/BF02478260
26. H.D. Landahl, R. Runge, Outline of a matrix calculus for neural nets. Bulletin of Mathematical Biophysics. **8**(2), 75–81 (1946). doi:10.1007/BF02478464
27. H.D. Landahl, A matrix calculus for neural nets: II. Bulletin of Mathematical Biophysics. **9**(2), 99–108 (1947). doi:10.1007/BF02478296
28. O. Markova, S. Semerikov, M. Popel, CoCalc as a Learning Tool for Neural Network Simulation in the Special Course "Foundations of Mathematic Informatics". (CEUR Workshop Proceedings, 2018), http://ceur-ws.org/Vol-2104/paper_204.pdf. Accessed 30 Nov 2018
29. O.M. Markova, S.O. Semerikov, A.M. Striuk, H.M. Shalatska, P.P. Nechypurenko, V.V. Tron, Implementation of cloud service models in training of future information technology specialists. (CEUR Workshop Proceedings, 2019), http://ceur-ws.org/Vol-2433/paper34.pdf. Accessed 10 Sep 2019
30. W.C. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics. **5**(4), 115–133 (1943). doi:10.1007/BF02478259
31. T.M. Mitchell, Key Ideas in Machine Learning. http://www.cs.cmu.edu/%7Etom/mlbook/keyIdeas.pdf. Accessed 28 Jan 2019
32. O.S. Permiakova, S.O. Semerikov, *Zastosuvannia neironnykh merezh u zadachakh prohnozuvannia* (The use of neural networks in forecasting problems), in *Materials of the International Scientific and Practical Conference "Young scientist of the XXI century"*, KTU, Kryviy Rih, 17–18 November 2008
33. W. Pitts, W.S. McCulloch, How we know universals the perception of auditory and visual forms. Bulletin of Mathematical Biophysics. **9**(3), 127–147 (1947). doi:10.1007/BF02478291
34. W. Pitts, A general theory of learning and conditioning: Part I. Psychometrika. **8**(1), 1–18 (1943). doi:10.1007/BF02288680
35. W. Pitts, A general theory of learning and conditioning: Part II. Psychometrika. **8**(2), 131–140 (1943). doi:10.1007/BF02288697
36. W. Pitts, Some observations on the simple neuron circuit. Bulletin of Mathematical Biophysics. **4**(3), 121–129 (1942). doi:10.1007/BF02477942
37. W. Pitts, The linear theory of neuron networks: The dynamic problem. Bulletin of Mathematical Biophysics. **5**(1), 23–31 (1943). doi:10.1007/BF02478116
38. W. Pitts, The linear theory of neuron networks: The static problem. Bulletin of Mathematical Biophysics. **4**(4), 169–175 (1942). doi:10.1007/BF02478112
39. N. Rashevsky, Mathematical biophysics of abstraction and logical thinking. Bulletin of Mathematical Biophysics. **7**(3), 133–148 (1945). doi:10.1007/BF02478314
40. N. Rashevsky, Outline of a physico-mathematical theory of excitation and inhibition. Protoplasma. **20**(1), 42–56 (1933). doi:10.1007/BF02674811
41. N. Rashevsky, Some remarks on the boolean algebra of nervous nets in mathematical biophysics. Bulletin of Mathematical Biophysics. **7**(4), 203–211 (1945). doi:10.1007/BF02478425
42. N. Rashevsky, The neural mechanism of logical thinking. Bulletin of Mathematical Biophysics. **8**(1), 29–40 (1946). doi:10.1007/BF02478425
43. T.F. Rienzo, K.K. Athappilly, Introducing Artificial Neural Networks through a Spread-sheet Model. Decision Sciences Journal of Innovative Education. **10**(4), 515–520 (2012). doi:10.1111/j.1540-4609.2012.00363.x
44. M.A. Ruggiero, *Cybernetic Trading Strategies: Developing a Profitable Trading System with State-of-the-Art Technologies* (John Wiley & Sons, New York, 1997)
45. M. Ruggiero, US Patent 5,241,620, 31 Aug 1993
46. K. Schwab, N. Davis, *Shaping the Fourth Industrial Revolution* (Portfolio Penguin, London, 2018)
47. S.O. Semerikov, I.O. Teplytskyi, Yu.V. Yechkalo, A.E. Kiv, Computer Simulation of Neural Networks Using Spreadsheets: The Dawn of the Age of Camelot. (CEUR Work-shop Proceedings, 2018), http://ceur-ws.org/Vol-2257/paper14.pdf. Accessed 21 Mar 2019
48. S.O. Semerikov, I.O. Teplytskyi, Metodyka uvedennia osnov Machine learning u shkilnomu kursi informatyky (Methods of introducing the basics of

Machine learning in the school course of informatics), in *Problems of informatization of the educational process in institutions of general secondary and higher education.* Ukrainian scientific and practical conference, Kyiv, October 09, 2018. (Vyd-vo NPU imeni M. P. Drahomanova, Kyiv, 2018), pp. 18–20

49. A. Shimbel, A. Rapoport, A statistical approach to the theory of the central nervous system. Bulletin of Mathematical Biophysics. **10**(2), 41–55 (1948). doi:10.1007/BF02478329
50. G.L. Stebbins, *Edgar Anderson 1897-1969*. (National Academy of Sciences, Washington, 1978)
51. G.J. Sussman, J. Wisdom, *Structure and interpretation of classical mechanics*, 2nd edn. (MIT Press, Cambridge, 2015)
52. I.O. Teplytskyi, O.I. Teplytskyi, A.P. Humeniuk, New computer technology. **6**, 67–68 (2008)
53. I.O. Teplytskyi, *Elementy kompiuternoho modeliuvannia* (Elements of computer simulation), 2nd edn. (KSPU, Kryvyi Rih, 2010)
54. T. Wei, On matrices of neural nets. Bulletin of Mathematical Biophysics. **10**(2), 63–67 (1948). doi:10.1007/BF02477433
55. P.J. Werbos, Maximizing long-term gas industry profits in two minutes in Lotus using neural network methods. Transactions on Systems Man and Cybernetics. **19**(2), 315–333 (1989). doi:10.1109/21.31036
56. G. Young, On reinforcement and interference between stimuli. Bulletin of Mathematical Biophysics. **3**(1), 5–12 (1941). doi:10.1007/BF02478102
57. T. Zaremba, in *Neural Network PC Tools: A Practical Guide*, ed. by R.C. Eberhart, R.W. Dobbins (Academic Press, San Diego, 1990), pp. 251–283