Research Article

Ranking of Search Requests in the Digital Information Retrieval System Based on Dynamic Neural Networks

Viera Bartosova,1 Svetlana Drobyazko,2 Sergii Bogachov,3 Olga Afanasieva,4 and Maria Mikhailova5

1University of Zilina, Univerzitna 8215, 010 26, Zilina, Slovakia
2The European Academy of Sciences Ltd., 71-75 Shelton Street, London WC2H 9JQ, UK
3Financial University Under the Government of the Russian Federation, 49 Leningradsky Prospekt, Moscow 125993, Russia
4Moscow Aviation Institute (National Research University), Volokolamskoe Shose 4, Moscow 125993, Russia
5Medical Sciences, Sechenov First Moscow State Medical University, 8-2 Trubetskaya Str., Moscow 119991, Russia

Correspondence should be addressed to Svetlana Drobyazko; svetlana.drobyazko@yahoo.com

Received 20 November 2021; Revised 1 March 2022; Accepted 15 March 2022; Published 11 April 2022

Academic Editor: Yu Zhou

Copyright © 2022 Viera Bartosova et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The article is devoted to the problem of optimization of search request ranking algorithms in the digital information retrieval system. The algorithm of functioning of the neural network ranking unit based on Hopfield neural network is built. The ability to generate a ranked list of pages found as a result of the request in the digital information retrieval system can be provided by solving two problems of integer optimization: the problem of assignment of combinatorial sets of criteria for assessing the relevance of web page search and the problem of sorting of numbers—relevance values. The architecture of the neural network model based on the dynamic Hopfield neural network with binary output function designed for combinatorial optimization of the final list of documents found in the digital information retrieval system was synthesized. Promising variants of neural network models with binary output function of neurons for synthesis of the optimal evaluation plan with a combinatorial set of criteria by solving the problem of assignment were built. It has been proven that the built models differ in the rules for determining the coefficients of synaptic connections and external shifts; each of the created rules can be used independently or in different combinations with one another. In the course of analytical research, it was found that the optimization formulation of the problem of sorting of relevance values of search pages is identical to the problem of assignment of combinatorial groups of evaluation criteria provided that the elements of the performance matrix of the latter are defined as linear combinations of relevance values.

1. Introduction

In the process of development of the Internet and the growth of data from local area networks, the probability of the existence of the required information increases, and the possibility of finding it decreases. Proper search for the necessary information becomes a common problem. New methods and tools are needed that solve the problems of relevant search in extremely large amounts of information. The goal of digital information retrieval systems (DIRS) developers is to provide the user with a search engine results page (SERP) that is as relevant as possible to the content of the personal request (to ensure relevance and accuracy of the search), while generating as much SERP as possible containing the requested information (to ensure completeness of the search).

The development of search engines is impossible without their intellectualization, which includes the use of semantic-syntactic analysis of texts, natural language tools, intelligent algorithms for determining the significance and information resources for the user, obtaining additional information about documents by analyzing hypertext structure and user preferences, and many other tools. In the DIRS, a request is generated as a keyword or a combination of keywords related to logical operations. To search for the same textual information, different keywords are used, the choice of
which is subjective [1]. But even if the keywords are defined and the request consists of only one word, the search result may differ when using different search engines. This is due to different procedures for indexing the text of documents in different search engines, as well as due to the fact that the search result can be represented as a very large list of found documents. The number of documents found in the list when searching by a keyword can be reduced by continuing the search, but already among these found documents (a number of search engines provide this option), using another keyword as a request. Increasing the number of keywords and continuing the search procedure in the found documents allows to reduce the number of found documents to a reasonable value, at which viewing the text of documents on the topic of interest is already real. That is why the right choice of a sufficient number of keywords, and even better word combinations, is a challenge, especially when working with unfamiliar material. Neural networks are becoming a software and system necessity that can effectively solve these problems. All of this determines the high level of relevance and necessity of this research.

2. Related Work

Among all variety of types of problems of mathematical programming, there are problems in which the set of acceptable alternatives is known or infinite, and it is based on certain rules of combinatorics. Such problems constitute a class of combinatorial optimization problems. Since the problems of combinatorial programming are reduced to the choice of a finite set of alternatives, in principle, each such problem may be solved by a complete search. It is natural to state the question of the complexity of the algorithms that implement this method. In this case, when the authors say about a dynamic binary Hopfield neural network, they can get a higher effect. As is known, in static neural networks, the absence of feedback guarantees their unconditional stability. They cannot enter in some mode when an output continuously switches between states and is not suitable for use. But this very desirable property is not achieved for free; networks without feedback have more limited capabilities compared to those with feedback. Since networks with feedback have pathways for transmission of signals from outputs to inputs, the response of such networks is dynamic; that is, after using a new input, the output is calculated and, transmitted via feedback, modifies the input [2]. Then, the output is calculated again; the process is repeated again and again. For a stable network, successive iterations lead to smaller output changes until the output becomes constant. The problem of stability is solved by the Hopfield neural network, which is a subset of networks with feedback, which outputs eventually reach a steady state. This is the main advantage.

A number of fundamental works on the theory of intelligent search processes in DIRS [2–4] state that without effective ranking search results lose importance because they may include references to tens and hundreds of thousands of documents. In such conditions, ranking is a general problem imperative, and the main problems of development of scientific bases of architectural principles of DIRS are insufficient theoretical elaboration of approaches used in practice, relatively low level of use of developed mathematical mechanisms, and lag of theoretical developments from rapidly changing search needs of computer users.

A number of scientists and experts [5–8] point out that the ranking of requests in search engines is a rather complex process. Developers are constantly trying to improve ranking algorithms, usually pursuing two major goals—to improve search quality and reduce the possibility of artificial influences on the ranking of results. One or another DIRS can take into account many factors, which one way or another affect the state of a particular document in the issuance of a particular request. At the present stage of technical development, most DIRSs are based on various complex modifications of the same ranking method [9, 10]. The method is reduced to calculation of the relevance of SERP to the search phrase based on two factors—internal and external ones [11]: document relevance = (internal factor (search phrase) + external factor (search phrase)) * confidence coefficient. An internal factor in the ranking method is roughly the density of the search phrase in the body of the page multiplied by the coefficient of thematic relevance of the page to the search phrase. An external factor in the ranking method is the number of links to the page with the text of the link corresponding to the search phrase multiplied by the coefficient of thematic relevance of links to the search phrase and the confidence coefficient of the link to the website. Specific changes to this formula apply to all documents in the search engine database. Then, the issuance is sorted in descending order of relevance. Documents, whose relevance is less than a certain threshold, do not appear in the issuance [12].

According to [13, 14], the above global difficulties of search in DIRS cause low adequacy of the information found on request of the user, i.e., return of a large volume of uninformative pages by the system. The problem may be exacerbated by the low speed of receiving answers from the Internet, the need for the user to view all found documents and evaluate their information content in a nonautomated mode. An alternative to secondary search procedures can be the development of fast algorithms for selecting and sorting returned SERPs according to the functions of significance of information resources and relevance of requests. The problem of optimal representation of a limited set of SERPs from some set found in order to maximize the total relevance of the request is presented by the problem of combinatorial optimization in this scientific work. The formation of the final list of found documents is provided by a recurrent procedure of application of the developed combinatorial algorithms.

3. The Method of Ranking of DIRS Requests Based on Neural Network Solving of Combinatorial Problems

3.1. The General Sequence of the Method. The importance of research and development of working algorithms is determined by the need to maintain and develop high
performance of DIRS with a large amount of indexed information and compliance with the vector criterion of significance. In practice, this often leads to the impossibility of applying those algorithms that have proven themselves well in experimental studies in DIRS. In our case, high-performance requirements for DIRS are provided by a neural network computing basis, which allows to parallelize tasks; that is, the algorithm allows to perform resource-intensive precision ranking operations for requests that have a chance to be in high enough places in the issuance by the results of a rougher assessment of their relevance. The SERP ranking method developed by the author is based on two facts:

(1) The increase in the number of search objects contradicts the need to implement in a limited time interval complex assessments of the relevance of content of the information found to the requirements of the search request. Therefore, existing information technologies do not provide effective implementation of search engines without parallelization of their methods and algorithms or exponential increase in the productivity of computing resources.

(2) The complexity of the mathematical support of the algorithms for presenting the found documents and the complexity of the tasks solved using DIRS themselves do not exclude the set of criteria of relevance. In accordance with this set, the selection criteria for the found documents are already presented as Pareto-optimal. This feature is poorly reflected in the algorithmic support of the final stage of search and presentation of SERPs in DIRS [15, 16].

The general sequence of the method of ranking before delivery of search result to the user that was developed by the authors can be presented by four stages:

(1) The set “AI” of content correspondences (significance criteria) between the information request and SERP (search result) is dynamically determined, and also the initial set “P” of the found pages which is exposed to ranking is designated.

(2) The table of correspondence of the set “P” of the found pages with the set of significance criteria is built. The table defines the initial data for solving the ranking problem within the combinatorial assignment problem.

(3) A neural network model for solving the ranking problem based on a dynamic binary Hopfield neural network is formed.

(4) Having initialized the neural network with random input vectors, one obtains the required sequence of SERP indices in accordance with a given set of relevance criteria in DIRS.

The algorithm of search for information relevant to the information needs of the user is the main advantage of DIRS. By relevance, the authors mean the correspondence between the required and the retrieved information. Construction of the optimal sequence of application of certain tools at each step of the search determines its effectiveness. The proposed method allows to solve the problem of choice and can give a clear idea of the types, purposes, and properties of DIRS. Within the two main classes of information retrieval systems: (1) search engines and (2) search directories, the authors identify the search categories: (a) by keywords; (b) with Boolean logic of word combinations; (c) by word combinations; (d) taking into account the distance between words; (e) case sensitive; (f) semantic (conceptual); (g) according to a pattern (similarity); and (h) by document margins. The method of search engine operation proposed by the authors provides automatic indexing of a large number of documents, but it has no advanced tools of artificial intelligence for expert evaluation of information. This explains the low relevance of the results of search engines (relevance means the degree of correspondence of the search results to the user query). Search directories provide more relevant results at the expense of manual preprocessing of queries by editors.

3.2. Representation of the Method of Ranking by the Neural Network Solution of Combinatorial Sorting of SERPs. The first stage consists in determining the set of content correspondences “AI” and the formation of the initial set of ranked search pages—a trivial method that mainly depends on the characteristics of a certain DIRS. The formed set of content correspondences is denoted as a set of significance criteria “V” (importance of search request) of DIRS at generation of SERPs, \( V = \{v_{uj}, u = 1, \ldots, U; j = 1, \ldots, M \} \), where \( M \) is the total number of DIRS request significance criteria. Suppose that as a result of a request a set of search pages \( D \) is found \( \{d_j\}, j = 1, \ldots, M \), according to each criterion \( v_{uj} = 1, \ldots, U \), has a certain relevance \( r_{uj} = 1, \ldots, j = 1, \ldots, M \). From a given set of search pages \( M \), it is necessary to form a set of pages \( S \) in which the total \( S \) is relevant: \( \sum_{u=1}^{U} \sum_{j=1}^{S} r_{uj} \), by all criteria of significance will be maximum [17]. To ensure the completeness of the selection from the set of alternatives, we will consider the total relevance of \( r_{uj} \) of any SERP \( d_j \) in relation to a group of various combinations of criteria \( v_u \) (Figure 1).

We assume that the \( I \)th group of DIRS request criteria consists of \( k \) criteria with \( U \), where \( U \) is the total number of DIRS request significance criteria. The number of variants of alternate selection of \( k \) criteria with \( U \) is \( U!/(U-k)! \). In the formed subsets of \( k \) criteria, each group has its own position. However, from the permutations of the criteria in the space covered by the subset, the number of the corresponding document does not change.

The number of subsets with different potencies is \( U \). Therefore, the maximum number of groups of criteria for the significance of the request (or works in terms of the assignment problem) will be

\[
N = \frac{\sum_{k=1}^{U} U!}{k!(U-k)!}.
\]
where $U$ is the total number of criteria for the significance of the request in DIRS.

In this regard, there arises a need to develop an efficient and fast neural network algorithm to solve the problem of assignment of one SERP from set $S$ to each group of request criteria that the problem of ranked representation of found DIRS documents is reduced to.

At the second stage of the ranking method implementation, it is necessary to distribute SERPs by groups of criteria in such a way that each page is preferably evaluated by one group of criteria, each group of criteria is preferably used for evaluation of one page, and the total relevance of the SERP by all combinatorial groups of criteria of request significance is maximum [18].

We designate through $R = [r_{ji}] - N \times M$ the performance matrix (Table 1), whose elements $r_{ji}$ form the relevance of the document with the number $j$ relative to the group of criteria with the number $i$. $M$ is the total number of SERPs found by DIRS, and $N$ is the total number of combinatorial groups of criteria of request significance.

We designate through $X = \|x_{ji}\| - N \times M$ the matrix of the unknown, whose element $x_{ji}$ takes the value “1,” if an SERP with the number $j$ will be evaluated as having maximum relevance by the group of criteria with the number $i$, and the value “0” otherwise. We present the constraints of the mathematical model as a system of equations:

$$\begin{align*}
\sum_{j=1}^{M} x_{ji} & \leq 1, \ i = 1, N, \\
\sum_{i=1}^{N} x_{ji} & \leq 1, \ j = 1, M, \\
x_{ji} & \in \{0, 1\}, \ j = 1, M, i = 1, N.
\end{align*}$$

In the given system, the first equation means that each SERP is evaluated by one group of criteria. According to the second equation, each group of criteria is used to evaluate one search page. The conditions of the third equation are natural constraints on the introduced variables. Given this, we determine the destination matrix $X$, which forms the criterion of optimality:

$$F(X) = \sum_{i=1}^{M} \sum_{j=1}^{N} r_{ji}x_{ji} \longrightarrow \max.$$  \hspace{1cm} (3)

Problems (2)-(3) are called assignment problems with an additive criterion of optimality. When considering the assignment problem in its standard form, it is assumed that the number of different combinatorial groups of criteria is equal to the number of SERPs: $M = N$. It is not difficult to demonstrate that by introducing fictitious groups of criteria or fictitious SERPs found, the open mathematical model (2) is made equivalent to the following model [19]:

**Table 1: Input parameters of the assignment problem in DIRS.**

| Criteria group | 1  |  |  | Criteria group | N  |  |  | Criteria group | N  |
|----------------|----|---|---|----------------|----|---|---|----------------|----|
| SERP 1         | $r_{11}$ | ... | $r_{1i}$ | ... | $r_{1N}$ | ... | ... | ... | ... |
| ...            | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| SERP j         | $r_{ji}$ | ... | $r_{ji}$ | ... | $r_{jN}$ | ... | ... | ... | ... |
| ...            | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| SERP M         | $r_{Mi}$ | ... | $r_{Mi}$ | ... | $r_{MN}$ | ... | ... | ... | ... |
Based on the fact that the matrix of condition constraints (4) is absolutely unimodular (an integer matrix is called absolute or completely unimodular, if any of its minors is equal to ("1," "-1," or "0"); then any pivot of the mathematical model (4) is an integer, hence the equivalence of the mathematical models (2) and (4) [20]. In addition, since conditions (4) and conditions of invariance of variables automatically imply that variables cannot be more than "0," the initial mathematical model (2) is equivalent (in terms of finding the optimal solution to the assignment problem) to the mathematical model with constraints (4), conditions $M=N$ and constraints $x \geq 0$, $j=1,2,...,M$. Suppose, for example, that the number of SERP documents found exceeds the number of criteria (groups of criteria) $N$. We introduce additional fictitious criteria (groups of criteria) with indices: $j=N+1$, ..., $M$. Suppose that the coefficients of the assignment table are equal to zero. In this case, we obtain the problem formulated in the standard form. If in the optimal plan of this problem $x_{ji}=1$, then the search page $j$ is evaluated by a fictitious criterion (group of criteria), that is, remains without work [21]. The method using a neural network model allows to rank SERPs both in the case of many $V$ (significance criteria) and in the extreme case of $V=1$.

We reformulate the ranking problem as follows. Suppose there is a set of numbers—the values of the relevance of SERPs found as a result of a DIS request: $r_j$, $i \in N=1,...,n$. It is necessary to arrange the numbers in ascending order, that is, to find this permutation of indices $j=\pi(i)$, then $\forall j \in N, r_{j+1} \leq r_j$. We build the solution of this problem on the basis of using a neural-like model synthesized above to solve combinatorial optimization problems. For the purpose, we reduce the formulated problem of sorting of numbers to some optimization problem. To do this, we prove this statement. Suppose that there is an arbitrary set of numbers $r_j$, $i \in N=1,...,n$ and a monotonically increasing series of positive numbers $\{a_j\}$, $j \in N=1,...,n$, such that $\forall j \in N, a_{j+1} \geq a_{j} + 1$. On the set of various substitutions $\{\pi(i)\}=\Pi$, $j=\pi(i)$ and set the linear functional as follows:

$$J[\pi(i)] = \sum_{j=1}^{n} a_j r_j. \quad (5)$$

We designate permutation $j^* = \pi^*(i)$, which sorts in ascending order a given arbitrary set of numbers $r_{(j)}$, that is, satisfies the condition:

$$\pi^*(i) = \arg\max_{\pi(i) \in \Pi} \{ J[\pi(i)] \} = \arg\max_{\pi(i) \in \Pi} \left\{ \sum_{j=1}^{n} a_j r_j \right\}. \quad (6)$$

To prove this statement, we take an arbitrary substitution $\pi(i)$, which generates the next permutation of the initial set of numbers $(r_1, r_2, \ldots, r_k, \ldots, r_p, \ldots, r_n)$ (Figure 2(a)). For certainty, we assume that $k < l$ and $r_k \geq r_l$; that is, this pair of numbers in this permutation is not sorted in the sense of the problem formulated above. We take the second substitution $\pi^*(i)$ obtained from the first transposition of the $k$th and $l$th elements. This substitution will correspond to the permutation of the initial set of numbers: $(r_1, r_2, \ldots, r_l, r_k, \ldots, r_p, \ldots, r_n)$, which differs from the first one in that the number $r_k$ is in the $k$th place and the number $r_l$ is accordingly in the $l$th place (Figure 2(b)). In such a permutation, this pair of numbers will be sorted in the format of the problem.

We determine the values of the functional (5) on the substitutions $\pi(i)$, $l \neq \pi^*(i)$:

$$J[\pi(i)] = \sum_{j \in N} a_j r_j + a_k r_k + a_l r_l \quad (7)$$

$$J[\pi^*(i)] = \sum_{j \in N} a_j r_j + a_k r_k + a_{l} r_{l}. \quad (7)$$

and find their difference: $\Delta kl = J[\pi^*(i)] - J[\pi(i)] = a_k r_k - a_l r_l - (a_k r_k + a_l r_l) = (a_k - a_l) \times (r_k + r_l)$. On the strength of the accepted assumption $k < l$ and $r_k \geq r_l$, so, firstly, $r_k + r_l \geq 0$; secondly, $a_k - a_l \geq 0$, as the series of numbers $\{a_j\}$ is monotonically increasing by definition. From which, it follows

$$\Delta kl = J[\pi^*(i)] - J[\pi(i)] \geq 0. \quad (8)$$

that is, the value of the input linear functional (5) in the permutation, in which some pair of numbers is sorted, is greater than or equal to the value of this functional in the permutation, which differs only in that the selected pair of numbers is not sorted.

At the third and fourth stages of the method implementation, a neural network model for solving the ranking problem based on a dynamic binary Hopfield neural network is formed and used.

4. Binary Hopfield Neural Network-Based SERP Ranking Model

4.1. Synthesis of Architecture and Parameters of the Hopfield Neural Network. We consider the neural network interpretation of the problem of ranking by many criteria as a problem of assignment provided that the problem of assignment is reduced to the standard form (the number of groups of criteria is equal to the number of ranked documents) [23, 24]. We define the architecture of the neural...
network by solving problem (2) under constraints (3). We include in consideration a network of binary neurons, which is a matrix with dimension of \( n \times n \), where \( n = N - M \) is the number of SERP or groups of criteria. A ranking model can be based on a neural network (Figure 3) containing arbitrary feedbacks by which the transmitted firing returns to the neuron and it performs its function again [25].

In dynamic neural networks, instability is manifested in wandering changes in the states of neurons, which does not lead to steady states. In the general case, the answer to the question of the stability of the dynamics of an arbitrary system with feedbacks is extremely difficult and is still open [26] (Figure 3).

It is assumed that the discrete Hopfield neural network used have the following characteristics: (1) one layer of elements (input elements representing the input sample are not taken into account); (2) each element is connected to all other elements, but not to itself; (3) only one element is refreshed in one stage; (4) elements are refreshed randomly, but on average each element must be refreshed to the same extent (with the same frequency); and (5) element output is limited to "0" or "1"; that is, the output function is binary [27].

The Hopfield network is recurrent in the sense that, for each input sample, the network output is reused as input until a steady state is reached. After "start," the appropriately organized (programmed) neural network changes its state, gradually moving to steady state. This mode identifies the result—a SERP evaluation plan with many groups of criteria, which may not match the exact solution. Random search procedures are usually used to clarify the result [28].

It is convenient to assume that the Hopfield network has no input elements, as the input vector simply determines the initial values of the activity of the elements. The elements are refreshed when all elements transmit their activity values according to the available weighted value of connections, after which the sum of terms is calculated (i.e., a scalar term is taken). The activity value of the element is based on the use of some activation rule. Each integer variable \( x_{ij} \) is matched to the output signal of the \( ij \)th neuron \( u_{ij} \), which is in the \( i \)th row and in the \( j \)th column of the network matrix.

\[
\begin{align*}
\sum_{j=1}^{n} u_{ji} &= 1, \forall i \in \{1, N\} \quad (10) \\
\sum_{i=1}^{n} u_{ij} &= 1, \forall j \in \{1, N\} \quad (11) \\
\sum_{j=1}^{n} u_{ji} &= n \quad (12) \\
\Phi(u) &= \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} r_{ji} \rightarrow \max, \quad (13)
\end{align*}
\]

where \( u_{ij} \) is Hopfield neural network output value (Figure 3) and \( R_{ji} \) is performance matrix value (Table 1) whose elements \( r_{ji} \) represent the relevance of the search page with the number \( j \) in relation to the criterion (group of criteria) with the number \( i \).

We build the energy function \( E^0(u) \), whose minimization ensures the fulfillment of constraints (11)–(13) and the solution of problem (14). We build it in the following form:

\[ E^0(u) = E^0_p(u) + E^0_q(u), \quad (14) \]

where the last term provides the optimization of the cost function and up to constant \( F > 0 \) is unambiguously determined as follows:

\[ E^0(u) = -F \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} r_{ji}, \quad (15) \]

and the first term ensures the fulfillment of constraints and can be built in several ways. According to the first of them,
this component of the energy function being built has the following form:

$$E_\psi^0(u) = \frac{A}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} u_{ji} + \frac{B}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} u_{ji}$$

$$+ \frac{C}{2} \left( \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} - n^2 \right),$$

(16)

where $A$, $B$, and $C$ are positive constants. The first term takes the minimum and zero value only if each row of the matrix $\{u_{ij}\}$ contains no more than one element, the second term takes the minimum zero value if each column of the matrix contains no more than one element, and finally the third term takes the minimum zero value if the whole matrix $\{u_{ij}\}$ contains exactly $n$ elements.

The built function $E_\psi^0(u)$ reaches its minimum in all states that correspond to the set of constraints (11)–(13) and represent a plan of assignments. According to the second method to build this component of the energy function being built, we have

$$E_\psi^0(u) = \frac{A}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} - 1^2 + \frac{B}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} - 1^2,$$

(17)

where the first term takes the minimum zero value only if any row of the matrix $\{u_{ij}\}$ contains only one firing neuron, and the second term takes the minimum zero

---

Figure 3: Hopfield neural network in the problem of SERP ranking ($x$: input search requests within a separate DIRS, $y$: neuron outputs, and $w$: values of offsets and synaptic connections).

Figure 4: Assignment plan: Hopfield neural network matrix in the state of rest.
value if any column of the matrix contains exactly one firing neuron.

In general, the above function takes the minimum zero value only in states that meet constraints (11)-(12) and represent a plan of assignments. Combining the function (16) with the function (17) or the function (18), we build the energy function in its finished form:

$$E^0(u) = \frac{A}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji}u_{ji} + \frac{B}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} + \sum_{i=1}^{n} u_{ji}u_{ji}$$

$$+ \frac{C}{2} \left( \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} - n^2 \right) - F \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji}$$

(18)

or

$$E^0(u) = \frac{A}{2} \left( \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} - 1^2 \right)$$

$$+ \frac{B}{2} \left( \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} - 1^2 \right) - F \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji}$$

(19)

We define network parameters having compared one of the obtained functions with the energy function put down in the general form:

$$E(u, T, I) = \frac{A}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{\mu=1}^{n} \sum_{\nu=1}^{n} T_{j\mu \nu} \mu_{ji}u_{ji} \mu_{ji} + \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji}I_{ji}$$

(20)

where $T_{j\mu \nu}$ is the coefficient of connection between the input of the $j$th neuron and the output of the $\mu$th neuron and $I_{ji}$ is shift of the $j$th neuron.

In this formula, for the energy function of the network, the time parameter is intentionally omitted due to the fact that in determining the synapses and external shifts, it does not play a significant role for both discrete-time networks and continuous-time networks. Moreover, this formula should be used when determining the parameters of synthesized networks, both with discrete and continuous states. The reason for this is the fact that the energy functions of networks with discrete and continuous states differ only in the presence of the integral term in the latter, which does not explicitly depend on the values of synapses and external shifts [29, 30].

In order to determine the parameters of the network in accordance with the built energy function (19), we give the formula for this function in the form

$$E^0(u) = \frac{A}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{\mu=1}^{n} \sum_{\nu=1}^{n} \mu_{ji}u_{ji} - Cn \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} - F \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji} - C_{n^2}$$

(21)

and equate the coefficients for the linear and quadratic terms of the last formula and energy (21). The last term can be excluded from consideration, as it does not depend on the state of the network. Matching of linear terms allows to determine the value of external shifts, and matching of quadratic terms allows to determine the synaptic connections between neurons. Analysis of the first term of the built energy function indicates that any neuron in the network must have synaptic connections with the coefficient—$A$ with all neurons of the row of the same name (condition $\mu = i$) except the considered neuron itself (condition $\nu \neq i$).

The second term dictates the presence of connections with the coefficient—$B$ between the neurons of the column of the same name (condition $\nu = j$) in addition to its own feedback (condition $\mu = i$). The third term indicates that all neurons in the network are connected to each other by synapses with the coefficients—$C$. With the help of the Kronecker symbol—$\delta_{ji}$, we form the resulting formula for synaptic connections of the network in the form

$$T_{j\mu \nu} = -A\delta_{\mu}(1 - \delta_{\nu}) - B\delta_{\nu}(1 - \delta_{\mu}) - C =$$

$$= -A\delta_{\mu} - B\delta_{\nu} + (A + B)\delta_{\mu}\delta_{\nu} - C, i, j, \mu, \nu \in 1, n$$

(22)

Analysis of the fourth and fifth terms of the built energy function shows that all neurons in the network must be fed external offsets in the form

$$I_{ji} = -Cn - Fc_{ji}i, j \in 1, n.$$  

(23)

As a rule, in practical problems, it is assumed that $F = 1$ and $A = B$; then all nonzero connections have the same weight, which is equal to $A$. Moreover, when analyzing formulas (23) and (24), one can see that the presence of global connections with the coefficient $C$ of each neuron with each one in the end state of the network corresponds to the plan of assignments ensures the application of a combined signal, which is equal to $Cn$ and compensated by the constant offset $Cn$ to any neuron from all other neurons. Therefore, to simplify the structure of network synapses, global connections with weight $C$ and part of offset $Cn$ can be ignored in the first approximation [31]. In this case, a simplified network structure for the synthesis of the optimal SERP assessment plan, by solving the assignment problem, can be represented as shown in Figure 5.

The required model of a neural network with binary output function of neurons contains a matrix of $nn$ neurons, each of which is fed an external offset equal to the corresponding performance $I_{ji} = -r_{ji}$, and the output signal of any neuron $u_{ji}$, with the coefficient $A_{i}$, is fed to the inputs of all neurons of the same rows and columns. In order not to complicate the graphical model (Figure 5.), the connections and shifts of only one neuron $N_{i}$ are shown in it. An example of a plan of assignments is represented by a set of excited (put in bold) neurons. Another variant of network parameters for the optimal SERP assessment plan can be obtained using the built energy function in the form of (20). Similar to the above procedure, we give this expression in a new form (21):
In addition to the above functions, one can use various combinations of functions (17) and (18) to build the energy function $E^0(u)$. As a result, one will obtain different variants of neural network parameters. When building a neural network with continuous states, it is necessary to provide conditions for finding rest points in the corners of the $n$-dimensional cube of its state space. Another way to ensure strict binarity of the output signals of neurons in steady states is to add to the energy function being built an additional component that reaches a minimum value in the states of the network in which the output signals of neurons take the value “0” or “1.” An example of such a function is as follows:

$$E^0(u) = \frac{A}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji}^2 + \frac{B}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji}^2 \mu \nu - A \sum_{j=1}^{n} u_{ji},$$

(24)

and compare the coefficients for linear and quadratic terms, discarding the last term. As a result, we obtain

$$E^0(u) = \frac{A}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji}^2 + \frac{B}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji}^2 \mu \nu - A \sum_{j=1}^{n} u_{ji},$$

(27)

from where we define the following parameters of the search neural network:

$$T_{jiuv} = -A \delta_{ju} \delta_{iv} - B \delta_{iu} \delta_{iv} \mu \nu - C + G \delta_{ju} \delta_{iv},$$

$$I_{ji} = -C n - F r_{ji}, j, \mu, \nu \in \Gamma, n.$$  

(28)

Combining formulas (27) and (21), we can build the energy function as follows:

$$E^0(u) = \frac{A}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji}^2 + \frac{B}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{ji}^2 \mu \nu - A \sum_{j=1}^{n} u_{ji}$$

(29)

where the required network parameters are determined as follows:

$$T_{jiuv} = -A \delta_{ju} \delta_{iv} - G \delta_{ju} \delta_{iv},$$

$$I_{ji} = (A + B - G) / 2 - F r_{ji}, j, \mu, \nu \in \Gamma, n.$$  

(30)

The ranking of documents in the case of a single criterion of significance, i.e., in the extreme case of $V = 1$, is implemented under the following assumption: there are many values of relevance of the SERP list found as a result of the DIRS request: $|r_i|, i \in N = 1, ..., n$. As a result of ranking, the numbers are arranged in ascending order of indices $j = \pi(i)$ so that $\forall j \in 1 \ldots (n - 1), r_j \leq r_{j+1}$.

The pair of numbers $[r_i, r_j]$ was chosen arbitrarily. Similar proofs can be provided for any other pair of numbers. Specifically, condition (9) will be valid for different pairs of adjacent elements. Due to the fact that any permutation can be represented by the superposition of transposition of adjacent elements, the maximum value of the linear function (5) will be achieved at the permutation in
which all pairs of adjacent elements are sorted, which corresponds to the above solution of SERP sorting problem [32, 33].

Thus, the initial problem of finding a sorting permutation is reduced to the optimization of the linear function with a set of various permutations, whose solution is defined in the form of (6). To solve this problem using a neural network, we present its neural network interpretation. We assume an arbitrary substitution \( j = \pi(i) \) in the form of a matrix of neurons \( [V_{ij}] \) with \( nxn \) dimension as follows:

\[
(j = \pi(i)) \Rightarrow (V_{ij} = 1), \forall i, j \in \mathbb{1}, n.
\] (31)

At the content level, the excited state of the neuron \( V_{ij} = 1 \) corresponds to the fact that the \( i \)th element of the initial set of values of relevance \( r \) occupies the \( j \)th position in the substitution \( \pi(i) \). There must be exactly one excited neuron in any row and in each column of such a matrix, so the limitations of the form are valid:

\[
\sum_{i=1}^{n} V_{ij} = 1, \forall j \in \mathbb{1}, n.
\] (32)

The neural network interpretation of the optimizing function (5) will be presented as follows:

\[
\phi(V) = \sum_{j=1}^{n} a_j \sum_{i=1}^{n} r_{ij} V_{ij} \rightarrow \max_{M}.
\] (33)

Comparing the last formula with the results of the neural network interpretation of the assignment problem in the forms (11)–(14), we conclude that the optimization formulation of the SERP sorting problem is identical to the formulation of the assignment problem provided that the elements of the performance matrix are defined as follows:

\[
c_{ij} = a_j r_{ij}, i, j \in 1, \ldots, n.
\] (34)

Therefore, the further building of the neural network for solving the problem (6) will completely coincide with the building of the network for solving the SERP assignment problem. The parameters of networks for solving both the first and second problems will be determined by the same formula [34, 35]. Specifically, using the parameters of the network to solve the assignment problem taking into account the condition (35), we obtain the network parameters to solve the sorting problem in the following form:

\[
T_{j\mu v} = -A \delta_{j\mu} - \delta_{jv},
\]

\[
I_{j\mu} = -(A + B) - Fr_{j\mu}, j, \mu, v \in \mathbb{1}, n.
\] (35)

Using another formula to determine the parameters of the network for solving the assignment problem, one can obtain a family of neural networks to solve the problem of sorting data in DIRS.

4.2. Use of the Model through Relaxation of the Hopfield Network Energy Function. At the fourth stage of the method implementation, the Hopfield neural network is initiated by an input random vector that “relaxes” to its energy minimum, and the output results of each of the neurons are interpreted as an index according to which DIRS requests should be ranked. The input vector specifies the initial states of all elements—neurons. The element to be updated is selected randomly. The selected element receives weighted signals from all other neurons and changes its state. Another element is selected and the process is repeated. The neural network reaches the point when none of its elements, being already selected for updating, does not change its state. In the general case, the end rest point \( U^* \), to which the network will pass in the process of energy minimization, will be determined by its initial state \( U \) and the relief of the energy function \( E(U, T, I) \) in the space of states \( M \) [36, 37]. We turn to the simplified version of the network to solve the problem of assignment presented in Figure 3.

In this case, the relief of the energy function will be determined by the module of synaptic connections \( T = A \) and the performance matrix \( [r_{ij}] \). With a rather wide range of values of the elements of this matrix and an arbitrary value of the module of connections \( T \), the network not only does not guarantee the transition to the end state, which corresponds to the global maximum of the optimizing function but may have end steady states that do not correspond to any plan of assignments. This situation can be addressed by introducing additional matching of the values of the module of connections and external shifts of the network, which are the input data of the problem, by scaling the latter [38]. The selection of the scaling factor of the input data of the problem is based on the results of the study of the influence of the module of connections \( T \) on the relief of the energy function \( E \) for some specific variant of the performance matrix.

We consider a practical version of the problem with the size of \( 2\times2 \) with the performance matrix as follows:

\[
r_{ji} = \left( \begin{array}{cc} 1 & 2 \\ 4 & 3 \end{array} \right).
\]

Each initial state \( U \) of the network is matched by a four-bit binary code, which is a rolled matrix \( \{u_{ij}\} \), and the change in the energy function and its components depending on the state of the network is examined. A graph of change in the own component of the energy function is presented in Figure 6:

\[
E_{0} = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{\mu=1}^{n} \sum_{v=1}^{n} T_{j\mu v} u_{ji} u_{j\mu v},
\] (36)
determined by the value of the module of connections and obtained at \( T = 1 \).

Regardless of the value of the module of connections, this component takes the minimum (zero) value in the states corresponding to the plans (“1001” and “0110”) or characterized by the presence of no more than one excited neuron. The full line shows the graph of the change of the forced component of the energy function \( E_{c} = - \sum_{j=1}^{n} \sum_{\mu=1}^{n} I_{j\mu} u_{ji} u_{j\mu v} \), which is determined by external shifts in accordance with the performance matrix and does not depend on the value of the module of connections. Graphs of changes in the full energy function \( E = E_{0} + E_{c} \) at different values of the module of connections \( T \) are shown by solid lines. As can be seen from the analysis of the presented
results, at rather small values of the module of connections the relief of the energy function is complex and multi-extreme. Despite the fact that in this case the four-dimensional space of the network states is reflected on the integer axis, even such a simplified representation allows to conclude that with increasing value of $T$ the relief of the energy function becomes more regular, the number of local minima decreases, and global extremum is identified, which corresponds to the optimal plan of assignments "0110" [39]. More constructive conclusions can be drawn from the results of the study of the quality of solving the problem of assignment at different values of the module of connections. For the purpose, we consider the problem of assignment with the matrix:

$$
\begin{bmatrix}
10 & 5 & 4 & 6 \\
6 & 4 & 9 & 7 \\
1 & 8 & 3 & 6 \\
5 & 3 & 7 & 2
\end{bmatrix}
$$

Totally, in this case, there are 24 possible plans of assignments of criteria of evaluation of the found SERPs where the values of cost function are within the range from 11 to 32. The last value corresponds to the optimal solution, which is the only one. A family of initial states of the network $U^{(0)}$ is set, each of which consists of one excited neuron and all other excited neurons. Thus, by sequentially sorting all the neurons, a series of sixteen implementations of the evolutionary processes of the network from different initial states with the above selected performance matrix and the specific value of the module of connections $T$ is carried out. The results of solving the problem in six series, which correspond to the values of the module of connections from 5 to 10 that are presented in Figure 7.

At large values of $T$, close to the maximum element of the performance matrix $r_{\text{max}}$ out of sixteen solutions in a series six results give the optimal plan of assignments. With that, ten other solutions of the series correspond to local maxima, whose cost function value is not less than 28. This condition is maintained with a further increase in the module of connections. The module of connections is reduced to a value close to the average for all elements of the performance matrix, which leads to an increase in the number of strictly optimal solutions, a decrease in the number of locally optimal solutions, and the emergence of stable conditions not meeting the plans of assignments, whose number is growing.

Based on the above results, the following practical recommendations can be made. When solving the problem of assignment, using a neural network, the value of the module of synaptic connections should be within the range from the average to the maximum element of the performance matrix [40]. In practice, when changing the values of the module of synaptic connections is a difficult technical problem, this recommendation should be implemented through scaling the input performance values before their input into the network, in the form of neuron shifts at the given value of $T$. The scaling factor $k_m$ is selected from the range $k_m \in [r_{\text{mid}/T}; r_{\text{max}/T}]$, and the shift signals are defined in the form of $l_{ji} = r_{ji}/k_m$, $j, i = 1, \ldots, n$. The final selection is made in each specific case depending on the required quality of the solution and the available time [41]. If it is necessary to promptly obtain a solution of the problem that is not necessarily fundamental, but close enough to it by a single call to the network, the scaling factor is selected, which is close to $k_m = r_{\text{max}/T}$. If it is necessary to obtain strictly optimum solution and there is a possibility to call to the network several times, the factor is selected, which is close to $k_m = r_{\text{mid}/T}$, a series of solutions for various initial states of the network is found, and among the found plans of assignments of groups of criteria for SERPs, the best plan is assigned.

5. Neural Network Request Ranking Algorithm for DIRS

We define the basic neural network request ranking algorithm for DIRS. A general block diagram of the developed neural network ranking algorithm with a vector relevance criterion is given in Figure 8. In the first three stages, the input data of the algorithm are formed [42, 43]. At the first stage, the number of $Z$ starts of neural network relaxation is set to its energy minimum.

At the second stage, based on statistical, dynamic internal, and dynamic external factors, the vector of relevance criteria of the found SERPs is formed: $v_i = l_{i\ldots N}$. Since the
At the eighth stage, an unambiguous interpretation of the output signals of neurons binary Hopfield neural network is carried out. Interpretation of neural network outputs allows to obtain the desired sequence of SERP indices according to a given set of DIRS relevance criteria, and the total relevance value of \( R^*_r \) on the rth start of the neural network. Stages 7–9 are performed after achievement of the set quantity of 2 starts of relaxation of a neural network to its energy minimum (check of the condition \( r \geq Z \) at stage 9). At the final stage 10, the SERP presentation sequence is selected, whose total relevance value, \( R^*_r \), \( r = 1, \ldots, Z \), is maximum.

6. Practical Implementation of the Neural Network Search Request Ranking Algorithm in DIRS

The developed and implemented algorithms and methods for synthesizing the parameters of the neural network ranking unit allow determining the parameters of the network up to constant coefficients and can be used alone or in various combinations with each other. In this regard, the experimental results of comparing the quality of solving problems of assignment on neural-like networks, the parameters of which were determined at different values of constants in equations, are of practical interest. The authors consider the example of equations that define the parameters of the neural network model in the ranking unit for solving the problem of assignment with dimension of 20 to 200 links found on Internet. To evaluate the quality of the solution, the authors use the parameter:

\[
\Phi_{\hat{c}}(c) = \frac{\phi_{\hat{c}}(c) - \phi(c, x_{\text{opt}})}{\phi(c, x_{\text{min}}) - \phi(c, x_{\text{opt}})},
\]

where \( \phi_{\hat{c}}(c) \) is the average value of the cost function for the whole set of valid solutions \( x \) for a specific set of search request parameters.

In this equation, \( c \) denotes a set of input data of the problem, which, depending on the type of problem, will be a performance matrix \([c_{ij}]\). The values of \( \phi_{\hat{c}}(c) \) and \( \phi(c, x_{\text{opt}}) \) are determined by a complete search. To evaluate the statistically average value of quality \( \Phi_w(c) \), different variants of the input data randomly selected from the range \([0,1]\) are considered, for each of which 10 solutions of different initial conditions of evolution of the network \( V^d \) are obtained. Table 1 presents the results of evaluation of the statistically average value of quality \( \Phi_w(c) \) for solving the problem of assignment for different dimensions and the above rules for determining network parameters (Table 2).

The above results testify to quite acceptable quality of problem solving by means of the synthesized models of the neural network ranking unit.

To determine the efficiency of neural network ranking compared to traditional ranking methods, experiments were conducted with various search requests and samples of web links from 20 to 200. Table 3 summarizes the data on the time of problem solving.
Table 2: Results of the experimental evaluation of the quality of the solution of the vector ranking problem with the number of ranked links from 20 to 200.

| Network parameters (equations) | 20          | 40          | 80          | 100         | 200         |
|--------------------------------|-------------|-------------|-------------|-------------|-------------|
| (4.6)                          | +0.980      | 0.960       | 0.975       | 0.988       | 0.887       |
| (4.7)                          | 1.000       | 0.999       | 0.999       | 0.958       | 0.898       |
| (4.8)                          | 1.000       | 0.999       | 1.000       | 1.000       | 0.899       |

Table 3: Results of experimental comparison of the efficiency of problem solving with the number of ranked links from 20 to 200.

| Ranking algorithm                      | Number of ranked documents in a request |
|----------------------------------------|-----------------------------------------|
| Vector neural network ranking          | 5,1667e - 004 2,1667e - 002 (10 model runs) 0,1211 (5 model runs) |
| Scalar neural network ranking          | 6,379e - 004 9,8623ce - 002 (10 model runs) 0,5274 (5 model runs) |
| Sorting the relevance of found links by the “bubble” method | 1,1437e - 003 0,5274 |
| Sorting the relevance by selection of the largest element | 1,3496e - 003 0,8305 |
| Ranking by sorting the array of values by insert | 2,0054e - 003 0,9846 |
| Exhaustive search                      | 10.2159     |

Table 4: Examples of 20 search requests of the prototype of DIRS search engine.

| No. | Search request | No. | Search request |
|-----|----------------|-----|----------------|
| 1   | Syllabus       | 11  | Bright         |
| 2   | Wear           | 12  | Kalman filter  |
| 3   | Periodic       | 13  | Hill           |
| 4   | Analyzing the response | 14 | Probably an increase |
| 5   | Revive         | 15  | Tragic         |
| 6   | Test period    | 16  | Ophthalmology  |
| 7   | Speak          | 17  | Link           |
| 8   | Waiting for the storm | 18 | Verse          |
| 9   | Exemplary subject | 19 | Connect       |
| 10  | Album          | 20  | Early          |

Table 5: Time of ranking operation (c) depending on the sample size in the sequential implementation of the block.

| Search request | N = 20       | N = 80       | N = 200      |
|----------------|--------------|--------------|--------------|
| Syllabus       | 0.37 × 10^{-3} | 0.133 × 10^{-2} | 0.854       |
| Wear           | 0.13 × 10^{-3} | 0.932 × 10^{-2} | 0.86        |
| Periodic       | 0.273 × 10^{-3} | 0.563 × 10^{-2} | 0.34       |
| Analyzing the response | 0.57 × 10^{-3} | 0.3 × 10^{-2} | 0.618     |
| Revive         | 0.34 × 10^{-3} | 0.758 × 10^{-2} | 0.62       |
| Test period    | 0.8545 × 10^{-3} | 0.34 × 10^{-2} | 0.32     |
| Speak          | 0.62 × 10^{-3} | 0.348 × 10^{-2} | 0.68     |
| Waiting for the storm | 0.45 × 10^{-3} | 0.598 × 10^{-2} | 0.63     |
| Exemplary subject | 0.864 × 10^{-3} | 0.555 × 10^{-2} | 0.51     |
| Album          | 0.12 × 10^{-3} | 0.995 × 10^{-2} | 0.436     |
| Bright         | 0.386 × 10^{-3} | 0.774 × 10^{-2} | 0.68     |
| Kalman filter  | 0.56 × 10^{-3} | 0.578 × 10^{-2} | 0.657     |
| Hill           | 0.78 × 10^{-3} | 0.42 × 10^{-2} | 0.942     |
| Probably an increase | 0.655 × 10^{-3} | 0.233 × 10^{-2} | 0.85     |
| Tragic         | 0.30 × 10^{-3} | 0.177 × 10^{-2} | 0.773     |
| Ophthalmology  | 0.471 × 10^{-3} | 0.893 × 10^{-2} | 0.6     |
| Link           | 0.89 × 10^{-3} | 0.64 × 10^{-2} | 0.835     |
| Verse          | 0.535 × 10^{-3} | 0.88 × 10^{-2} | 0.579     |
| Connect        | 0.644 × 10^{-3} | 0.45 × 10^{-2} | 0.848     |
| Early          | 0.55 × 10^{-3} | 0.518 × 10^{-2} | 0.87     |

Table 4 shows examples of search requests of the prototype of DIRS search engine.

Investigation of the sequential dynamics of the neural network block during the ranking of web links. Although within this variant of the organization of the Internet ranking block, in turn, a number of subvariants can be identified, the latter will differ slightly from each other in their efficiency, so without loss of generality, the authors settled on one of them. It is implemented in two main steps. At the first step, the calculation of binary functions from the current variables of the model state is carried out. It is implemented in the form of a sequential process of variable sampling: the sum of Hopfield neural network inputs from the transfer buffer to the calculation unit of the binary output function, the implementation of the output function, and the transfer of the final output of one neuron to the buffer. The second step is to sequentially determine for each neuron the right part of the numerical scheme for obtaining the final difference. Adding the execution time of two consecutive steps, the authors obtain the desired estimate of the implementation time of one ranking step for a sequential version of the organization of the block. Table 5 presents the results of measuring the ranking time of different requests depending on the size of the sample being ranked.

In the structure of the implementation of this variant of the ranking block, there is an element of parallel summation of N+I number of neurons. This summation is carried out using the tree of two-input software adders. The most significant contribution to the duration τ, which depends on dimension of the model, is the time of calculating the sum \( \sum_{i=1}^{n} w_{ij} \hat{q}_{ij} \), where \( w_{ij} \) and \( \hat{q}_{ij} \) are neuron output and the size of its bias. The contribution of sum calculation to the total time can be reduced by using adders with a large number of inputs.

The family of intermediate variants of parallelization is naturally generated by the assumption of the presence of not one, as was the case in an absolutely sequential version, but several traditional serial processors, each of which simulates a group of neurons in an absolutely sequential scheme. The more processors, the less time τ. The limit case when the number of
Table 6: Time of ranking operation (c) depending on the sample size in the parallel implementation of the block.

| Search request     | N = 20       | N = 80       | N = 200      |
|--------------------|--------------|--------------|--------------|
| Syllabus           | 0.62 × 10⁻⁴  | 0.156 × 10⁻³ | 0.976 × 10⁻² |
| Wear               | 0.826 × 10⁻⁴ | 0.386 × 10⁻³ | 0.68 × 10⁻²  |
| Periodic           | 0.64 × 10⁻⁴  | 0.185 × 10⁻³ | 0.8 × 10⁻²   |
| Analyzing the response | 0.428 × 10⁻⁴  | 0.567 × 10⁻³ | 0.462 × 10⁻² |
| Revive             | 0.596 × 10⁻⁴  | 0.28 × 10⁻³  | 0.486 × 10⁻² |
| Test period        | 0.95 × 10⁻⁴   | 0.653 × 10⁻³ | 0.593 × 10⁻² |
| Speak              | 0.396 × 10⁻⁴  | 0.56 × 10⁻³  | 0.542 × 10⁻² |
| Waiting for the storm | 0.605 × 10⁻⁴  | 0.97 × 10⁻³  | 0.457 × 10⁻² |
| Exemplary subject  | 0.864 × 10⁻⁴  | 0.74 × 10⁻³  | 0.51 × 10⁻²  |
| Album              | 0.76 × 10⁻⁴   | 0.869 × 10⁻³ | 0.436 × 10⁻² |
| Kalman filter      | 0.461 × 10⁻⁴  | 0.578 × 10⁻³ | 0.542 × 10⁻² |
| Hill               | 0.146 × 10⁻⁴  | 0.42 × 10⁻³  | 0.47 × 10⁻²  |
| Probably an increase | 0.475 × 10⁻⁴  | 0.233 × 10⁻³ | 0.825 × 10⁻² |
| Tragic             | 0.486 × 10⁻⁴  | 0.177 × 10⁻³ | 0.589 × 10⁻² |
| Ophthalmology      | 0.565 × 10⁻⁴  | 0.893 × 10⁻³ | −0.46 × 10⁻² |
| Link               | 0.982 × 10⁻⁴  | 0.475 × 10⁻³ | 0.728 × 10⁻² |
| Verse              | 0.214 × 10⁻⁴  | 0.79 × 10⁻³  | 0.5 × 10⁻²   |
| Connect            | 0.47 × 10⁻⁴   | 0.33 × 10⁻³  | 0.48 × 10⁻²  |
| Early              | 0.445 × 10⁻⁴  | 0.764 × 10⁻³ | 0.974 × 10⁻² |

neurons is equal to the number of processors \( \tau \) will be determined by the time of absolutely sequential implementation of the model of one neuron. Approximately for this family of parallelization variants, one can consider \( \tau \sim N^{2/\pi} \).

An additional number of families of different variants gives hypotheses about the parallelization of the implementation of certain operations in the model of one neuron. Table 6 presents the results of measuring the ranking time of different requests depending on the size of the sample being ranked.

A comparison of Tables 5 and 6 shows a significant gain in ranking time with an increase in the number of ranked web links and the degree of parallelization of the neural network RIS (intellectual subsystem) block. Knowing the methods of parallelization of the neural network and substituting the corresponding values of the parallelization parameters, we can evaluate the efficiency of this specific example of the ranking block based on the RIS server. In conclusion, it should be noted that the methods considered in the study do not exhaust all possible variants for parallelization, and hence all possible variants for hardware implementation of the Hopfield neural network in the web search ranking block. For example, a broad family of parallelization variants allows using pipelined methods of arithmetic operations in the ranking block, which may be one of the areas of further studies on creation of a class of digital neuro accelerators for web search. The results of experimental tests of neural network ranking procedures showed high efficiency of the latter in the adapted multicore architecture of hardware and software.

7. Conclusion

An analysis of the existing capabilities of modern information retrieval systems and problems of intelligent search for textual information related to the need to solve discrete optimization problems was carried out. The analysis showed that as of today the neural network is not fully used in information retrieval systems to solve problems of optimal representation of the found SERPs. The analysis of neural network models of optimization of information retrieval procedures applicable in algorithms of information retrieval systems was carried out. So, the choice of method to solve the optimization problem in the operation of DIRS is determined by the original set of alternatives. With a small number of ranked SERPs, the use of accurate methods becomes effective; otherwise, heuristic methods are used for solving a specific optimization problem where neural networks have advantages. The solution of optimization problems with the help of neural networks is based on the properties of the latter to minimize the energy function of the stability of states corresponding to some local minima of the network energy.

A method of ranking documents in DIRS based on a neural network solution of combinatorial problems of assignment and sorting was developed. The solution of these problems with the help of neural network is due to the need to solve an integer problem of optimizing the large dimension of the found SERPs. The architecture of the neural network model of ranking of SERPs based on the dynamic binary Hopfield neural network was created. Several variants of neural network models with binary output function of neurons were created for the synthesis of the optimal evaluation plan based on a combinatorial set of criteria by solving the problem of assignment. The models differ in the rules for determining the coefficients of synaptic connections and external shifts, and practical recommendations for their selection are provided.

The algorithm of operation of the neural network ranking unit at various values of significance criterion was developed. During analytical studies, it was established that the optimization formulation of the problem of sorting of the values of relevance of documents is identical to the formulation of the problem of assignment of combinatorial groups of evaluation criteria, provided that the elements of the performance matrix of the latter are defined as linear combinations of relevance values.

Data Availability

No data were used to support this study.

Consent

Consent is not applicable.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Authors’ Contributions

Conceptualization was prepared by V. B. and S. D.; methodology and investigation were conducted by S. D.; software and resources were obtained by S. B.; validation was
References

[1] V. Maurya, P. Pandey, and L. S. Maurya, “Effective information retrieval system,” International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 4, p. 787, 2013.
[2] W. Arms, Manuscript of Digital Libraries, MIT Press, Cambridge, MA, USA, 2002.
[3] T. M. Cover and T. A. Joy, Elements of Information Theory, John Wiley & Sons Interscience Publication, Hoboken, USA, 1991.
[4] G. Kowalski and M. T. Maybury, Information Storage and Retrieval Systems, Springer, New York, NY, USA, 2005.
[5] F. Crestani, “Spoken query processing for interactive information retrieval,” Data & Knowledge Engineering, vol. 41, no. 1, pp. 105–124, 2002.
[6] P. A. Mohan, Residue Number Systems: Algorithms and Architectures, Kluwer Academic Pub, Boston, Florida, USA, 2002.
[7] E. Greengrass, “Information retrieval: a survey by ed greengrass,” Information Retrieval, vol. 141-163, 2002.
[8] E. Brynjolfsson and A. McAfee, Race against the Machine: How the Digital Revolution Is Accelerating Innovation, Driving Productivity, and Irreversibly Transforming Employment and the Economy, Digital Frontier Press, Lexington, Massachusetts, USA, 2012.
[9] G. Forman, “A pitfall and solution in multi-class feature selection for text classification,” in Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, January, 2004.
[10] W. Xue and H. Deng, “Unstructured queries based on mobile user context,” International Journal of Perceptive Computing and Communications, vol. 8, no. 4, pp. 368–394, 2012.
[11] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral Networks and Locally Connected Networks on Graphs,” pp. 1312–6203, 2013, https://arxiv.org/abs/1312.6203.
[12] K. Jarvelin and J. Kekalainen, “IR evaluation methods for retrieving highly relevant documents,” in In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, N. J. Belkin, P. Ingwersen, and M. K. Leong, Eds., pp. 41–48, ACM, New York, NY, USA, 2010.
[13] Y. Chang, I. Ounis, and M. Kim, “Query reformulation using automatically generated query concepts from a document space,” Information Processing & Management, vol. 42, no. 2, pp. 453–468, 2006.
[14] G. Chowdhury, Introduction to Modern Information Retrieval, Facet Publishing, UK., edition, 2010.
[15] A. F. Ward, “Supernormal: how the Internet is changing our minds and our memories,” Psychological Inquiry, vol. 24, 2013.
[16] K. Agbele, E. Ayetiran, and O. Babalola, “A context-adaptive ranking model for effective information retrieval system,” International Journal of Information Science, vol. 8, no. 1, pp. 1–12, 2018.
[36] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Hoboken, New Jersey, USA, 2010.

[37] M. Neumann, R. Garnett, C. Bauckhage, and K. Kersting, “Propagation kernels: efficient graph kernels from propagated information,” *Machine Learning*, vol. 102, no. 2, pp. 209–245, 2016.

[38] C. T. Meadow, B. R. Boyce, D. H. Kraft, and C. L. Barry, *Text Information Retrieval Systems*, Academic Press, Cambridge, Massachusetts, USA, 2007.

[39] B. Croft, D. Metzler, and T. Strohman, *Search Engines: Information Retrieval in Practice*, Pearson, New York, NY, USA, 2010.

[40] C. J. Van Rijsbergen, *The Geometry of Information Retrieval*, Cambridge University Press, Cambridge, MA, USA, 2004.

[41] V. Sutton, *Innovations in Information Retrieval: Perspectives for Theory and Practice*, Library Management, 2012.

[42] M. Levene, “Search engines: information retrieval in practice,” Pearson Education, New York, NY, USA, 2011.

[43] S. Buettcher, C. Clarke, and G. Cormack, *Information Retrieval: Implementing and Evaluating Search Engines*, MIT Press, Cambridge, MA, USA, 2010.