

# THE USE OF NEURAL NETWORKS FOR STRUCTURAL SEARCH ON WEB

Oana Gogan

Computer Science Department, "G.Asachi" Technical University, 11, Copou Bd., Iasi, Romania  
ogogan@eureka.cs.tuiasi.ro

Sabin Corneliu Buraga

Faculty of Computer Science, "A.I.Cuza" University, 16, G-ral Berthelot Street, Iasi, Romania  
busaco@infoiasi.ro

## Abstract:

We propose a document structure based method to search hypermedia information, using the neural network approach. The search activity can be divided into three major parts. In the first part, we are searching the information using a traditional search engine according with user's query and we are storing first found pages. The second part's goal is to encode the Web pages structural information (some users want to retrieve Web documents without tables or other users want the graphical information to be placed on bottom of the Web pages, etc.). We'll retain only the position and the occurrences of some HTML elements and attributes, building a matrix. The elements of this matrix will be encoded by means of a special operator to obtain an integer number. After this procedure, each page will be denoted by its structural information number. In the third phase, we will choose a self-organizing feature maps neural network based on the competitive learning. With this method, the user will be able to formulate complex and flexible queries. Our approach can be applied to all structured documents based on SGML/XML meta-languages.

**Keywords:** Web, neural networks, search, structured information, hypermedia

## INTRODUCTION

The Web, the world's largest hypertext structure, is often described as the multimedia section of the Internet. Despite many theoretical and technical advances, relatively slight scientific studies about the structural search activities were written. In the present, the most common approach in searching information on Web is the keywords based method. The growing of hypermedia information available on Internet shows the weakness of this traditional Web search technique. Recent statistics

show that the Internet, and especially the Web space, represents a huge information repository. Search services generally can be distinguished according to how they accumulate and organize their meta-information: *automatic acquisition and indexing* (performed by Web robots) and *manual acquisition and categorization* (accomplished by trained information specialists). Each method is not enough in the present.

Fichtner emphasizes the major problem of searching for data on Web: "*90% of all search attempts lead to almost endless lists of ridiculous Web sites, which contain the searched words purely by chance but have nothing in common with the desired topic – hits are a pure matter of luck*".

The need of intelligent knowledge discovery is crucial.

## OUR PROPOSAL

In this paper, we propose a document structure based method to search hypermedia information, using the neural network approach.

In our idea, the search activity can be divided into three major parts. In the first part, we are searching the information using a traditional search engine (e.g. *AltaVista* or *Excite*) according with user's query and we are storing first  $N$  ( $N=20$  or  $N=50$  usual) found Web pages. Of course, we'll retain the corresponding addresses (URLs) of these documents.

The user will be able to formulate complex and flexible queries, such as: "microprocessor" + "documentation" + without applets + with <3 tables on top + <10 paragraphs. The query language is generated by a context-free grammar and it can be analyzed by a classical syntactic processor. The given keywords (e.g. "documentation") will be used by the search engine and the remaining expressions (e.g. with <3 tables on top) will be processed in the activity of structural search.

## STRUCTURAL INFORMATION ENCODING

The second part's goal is to encode the Web pages structural information (according to the given possibility of querying language, some users want to retrieve Web documents without tables or without script programs or other users want the graphical information to be placed on bottom of the Web pages and maximum 10 paragraphs etc.). We'll retain only the position (top, middle, and bottom) and the occurrences of some HTML elements and attributes (e.g. <p>, <table>, <img>, <embed>, <applet> and so on), building a matrix. The elements of the matrix will be encoded by means of a special operator to obtain an integer number. For implementation issues, we consider that this integer number is stored on 64 bits. If there is more then one page associated with the same number, we will keep only one. After this procedure, each page will be denoted by its *structural information number*.

We use the following five HTML elements (tags) for our structural search approach: <p> (paragraph), <img> (image), <object> (multimedia or generic object), <table> (tabular data), and <a> (anchor). For each element, we will keep three values that represent the occurrences of that element on top, middle and bottom of the Web page. Each such a value will be stored on 4 bits (for all these values we are using  $3 \times 5 \times 4 = 60$  bits). The structural information number will compress these 15 values (3 positions times 5 considered elements) and 4 additional bits (Boolean values) which specify if there are scripts (given by <script>), Java applets (<applet>), style definitions (integrated by <style> or <link> elements or "style" attribute) and sound content (<embed> or <bgsound>) included into the HTML source of a Web page. In addition, using this method, for the user's searching request will be computed the query structural information number.

These numbers will be used in the next phase.

### An example

Let consider the query: "multimedia" + "documents" + with <7 paragraphs on top + with <2 images on bottom + <5 tables on middle + <10 paragraphs on bottom + 0 links + no multimedia content. We consider only the < relational operator. In the future, we'll adopt other relational operators that can appear in the query expressions.

After the keywords suppression, we will have the following matrix:

$$\begin{pmatrix} 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 10 & 2 & 0 & 0 & 0 \end{pmatrix}$$

where on rows we wrote the position (top, middle, bottom) occurrences of the elements and each column correspond to a HTML tag in this order: <p>, <img>,

<embed>, <table>, <a>. The computed structural information number can be encoded using the matrix and the four Boolean conditions in the described manner.

## THE USE OF NEURAL NETWORKS

Therefore, we will have  $N$  different numbers, which will be the input for a neural network, in the third phase. Because we need that only one page will be selected, according with the user's request, we will choose a self-organizing feature maps neural network. That kind of neural network is based on the competitive learning. The output neurons interact for being activated so only one will be activated, for input set. The winner neuron of the competition is named "*winner-takes-all*".

A usual way to introduce that kind of competition is to use laterally connections (ways of reaction) between the output neurons. In a self-organizing feature maps neural network, the neurons are placed in the knot of a lattice, with one dimension (Fig. 1).

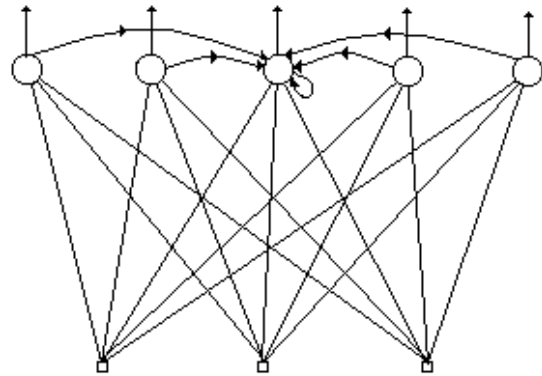


Figure 1. Mono-dimension lattice with direct and lateral connections for the neuron from the center of the lattice.

In the lattice, we have direct connections, between input and output layers, and lateral connections, between the neurons from the output layer. The direct connections are used for the obtaining of a selective reply at a certain input stimulus.

The neural network feed forward with lateral connections has two important characteristics:

- The network tends to focus its activity in local clusters;
- The place of these clusters depends by the nature of the input signals.

So, will be  $N+1$  inputs for the neural network:  $Num_1, Num_2, \dots, Num_N, Num_T$ , where,  $Num_1, Num_2, \dots, Num_N$  are the  $N$  numbers for the codification of the founded pages from the second part and  $Num_T$  is the codification of user's desired page. The values  $W_{j1}$ ,

$W_{j2} \dots W_{jN}, W_{jT}$  are the synaptic weights of  $j$  neuron (the  $W_j$  vector).

The numbers  $c_{j,-K}, \dots, c_{j,-1}, c_{j,0}, c_{j,1}, \dots, c_{j,K}$  are the synaptic weights of the lateral reaction, connected at  $j$  neuron, where  $K$  is the radius of the lateral interaction. The outputs of the network will be  $y_1, y_2 \dots y_N, y_T$ .

For  $j$  neuron, the output are described by the following equation:

$$y_j = \varphi \left[ I_j + \sum_{k=-K}^{k=K} c_{j,k} y_{j+k} \right] \quad (1)$$

where  $I_j$  is the stimulus of  $j$  neuron:

$$I_j = \sum_{i=1}^N W_{ji} N_i + W_{jT} N_T \quad (2)$$

and  $\varphi(\cdot)$  is a nonlinear function, which limits the values of  $y_j$  and assures that  $y_j$  is positive (we will use the function from equation (3)).

$$\varphi(t) = \frac{1}{1 + e^{-t}} \quad (3)$$

Using a relaxation technique, we can reform the equation (1) like one with differences:

$$y_j(n+1) = \varphi \left( I_j + \beta \sum_{k=-K}^{k=K} c_{jk} y_{j+k}(n) \right) \quad (4)$$

where  $y_j(n+1)$  is the output of  $j$  neuron at  $n+1$  moment and  $\beta$  is a positive factor, which controls the rate of convergence of the relaxation process.

If  $\beta$  is bigger enough, then in the final state, corresponding with  $n \rightarrow \infty$ , the values of  $y_j$  will be concentrated in the interior of spatial agglomeration (cluster), called activity bubble.

The bubble is centered in a output neuron, for which the value of initial response  $y_j(0)$ , caused by the stimulus  $I_j$  is maximum:

- If the positive reaction is strong, then the bubble becomes large;
- If the negative reaction is increased then the bubble becomes tight; if that reaction is too strong, the bubbles will not be created anymore.

### Auto-organization algorithm

The input patterns are presented one by one. For a certain type of input, will be active only one output

neuron.

The main mechanisms of the neural network are:

- A lattice with one dimension, which calculates the values of the activation function ( $y_j$ );
- A mechanism which compares these values and choose the neuron with the biggest value;
- A mechanism for the activation of the selected neuron and its neighborhood;
- A mechanism for the adapting of the neuron's weights.

A good way to choose the neuron with the highest activation is to calculate an error by gradient type and choose the minimum value:

$$E_j = \sum_{i=1}^N \frac{1}{2} (y_T - y_j W_{ji})^2 \quad (5)$$

At the end of the learning process, the index of that minimum value is the index of searched page (the best fit for the user request).

We will note with  $\Lambda_p(n)$  the positional neighborhood of the winner neuron. Its dimension will be variable during the time of neural running. After the find of the winner, the weights will be modify with the relation:

$$W_j(n+1) =$$

$$= \begin{cases} W_j(n) + \eta(n) [Num_j - W_j(n)] & \forall j \in \Lambda_p(n) \\ W_j(n), & \forall j \notin \Lambda_p(n) \end{cases} \quad (6)$$

where  $\eta(n)$  is a positive number, which controls the rate of the weights modification  $W_j$ .

The equation (6) will has as effect the modification of the weights  $W_j$  from the neighborhood of the winner.

The process of bubbles creation is critically dependent by the way of modification of the parameters  $\eta(n)$  and  $\Lambda_p(n)$ .

First, the parameter  $\eta(n)$  will have the value 1,  $\eta(0) = 1$ , and then it will be decrease in time depending of the number  $n$  of iterations. In a first step, in the first 1000 iterations  $\eta(n)$  will be more then 0.1. This step is a phase of ordering, when the weights suffers a big modification for the topological order. In the next step,  $\eta(n)$  will be maintained at small values, around 0.01, for a fine modification of the weights. That is the phase of convergence. We will run the network during 5000 iterations, so the rule used for  $\eta(n)$  is a linear one:

$$\eta(n) = 1 - \frac{n}{5000}, n \leq 4999 \quad (7)$$

At the beginning of the network run, we will have a big neighborhood, which will be decrease later. That means we will use first a strong positive lateral reaction and then we will create the negative lateral reaction.

The single output neuron will give the best-found Web page according to user's request and the result will correspond to desired document. The user will obtain the *URL (Uniform Resource Locator)* of this page to browse its content. As we seen, this URL was stored after the classical searching activity performed by the traditional search engines.

## CONCLUSSIONS AND FURTHER WORK

Our approach can be applied to all structured documents based on *SGML (Standard Generalized Markup Language)* and *XML (Extensible Markup Language)* meta-languages, such as *SMIL (Synchronized Multimedia Integration Language)*, used for hypermedia synchronized presentations on Web, to search in multimedia corpora.

In addition, our proposed method can be used in conjunction to *XML-GL*, a graphical language for querying structured and semi-structured data stored on hypertext databases or on Web. Another solution is to use *Metalog* language based on *RDF (Resource Description Framework)* that provides a logical view of metadata present on Web.

After other theoretical studies that must be tried out, we intend to develop and to experiment a software Web tool using the neural network approach for structural search on Internet.

## REFERENCES

Catledge, L.D., Pitkow, J.E. – “Characterizing Browsing Strategies in the World Wide Web”, Proc. 3<sup>rd</sup> Int. World Wide Web Conf., Darmstadt, Apr.1995

Cover, R. - The SGML/XML homepage (March 2000): <http://www.oasis-open.org/cover/xml.html>

Gütl, C et al. – “Future Information Harvesting and Processing on the Web”, European Telematics: advancing the information society Conf. Proc., Barcelona, Feb.1998

Haykin, S. – “Neural Networks: A Comprehensive Foundation”, IEEE Press, New York, 1994

Jenkins, C. et al. - "Automatic RDF Metadata Generation for Resource Discovery", WWW8 Conference Proc., Canada, Elsevier Science, May 1999

Lippmann, R. – “An Introduction to Computing with Neural Nets”, IEEE ASSP Magazine, Apr.1987

Marchiori, M. – “The Quest for Correct Information on the Web: Hyper Search Engines”, WWW6 Conf. Proc., France, Elsevier Science, 1997

Modjeska, D., March, A. – “Structure and Memorability of Web Sites”, Technical Report, Computer Science Research Institute, Toronto, 1997

Skillicorn, D.B. – “Structured Parallel Computation in Structured Documents”, Journal of Universal Computer Science, vol.3, no.1, 1997

\*\*\* - World Wide Web Consortium's Technical Reports: <http://www.w3.org/TR>

\*\*\* - QL'98, The Query Languages Workshop Proc., Boston, December 1998