Volume : 4, Issue: 1
January - June 2014

ISSN : 2229 - 3515

International Journal of ADVANCES IN SOFT COMPUTING TECHNOLOGY

Editor-in-Chief
Dr. Vaka Murali Mohan

Published by
BHAVANA RESEARCH CENTER

Information Retrieval and Summarization of Documents Using Artificial Neural Networks

Sunil Manohar Reddy. K[1]*., Dr. G. Ravindra Babu[2]., S. Krishna Mohan Rao[3]

1. Department of CSE, Aryabhata Institute of Technology & Science, Hyderabad, AP, India
2. Dept. of CSE, Trinity College of Engg. & Tech., Peddapally, Karimanagar (Dt), AP, India
3. Department of CSE, Siddhartha Institute of Engineering & Technology, Hyderabad, AP, India

KEYWORDS

Key Extraction; Algorithm; Artificial Neural Networks; Term Frequency; Document Frequency

Abstract: *This chapter presents the Neural Network based appearance for information retrieval from a heap of documents. The information retrieval is basically to take a query as input where a query consists of a set of words known as phrase and then to find out the documents out of heap of the documents which are relevant to input query and then these set of found out relevant documents are used to find out the summary of each relevant document to make it more compact in terms of information. So this paper directly saves the time in terms of finding out the information about anything, because as we straight away gets the main information. This paper is comprised of two phases. Firstly it searches the documents that are relevant to user queries by some effective techniques. Secondly the document can also be summarized as per the user requirement. Both the phases use Neural Network approach.*
.

1. INTRODUCTION

The very first collection of scientific papers those with key phrases are in the minority, say nothing of web pages. For documents with no key phrases, it is necessary to assign key phrases with each of these documents, either manually or automatically. Manual key phrase assignment is tedious and time – consuming so automatic methods benefit both the developers and the users of large document collections consequently several automatic key phrase extraction algorithms have been proposed based on different techniques.

Kea is a key phrase extraction algorithm based on Naive Bayes Classifier. Three attributes of each remaining phrase are calculated, whether or not it is an exemplar key phrase of the document, the distance into a document that it first occurs and its TF x DF value. The distance value is real and in the range 0 to 1, indicating the proportion of the document occurring before a phrases first appearance. TF x DF is a standard information retrieval metric that estimates how specific a phrase is to a document. TF [Term Frequency] is a measure of how frequently a phrase occurs in a particular document. DF [Document Frequency] is a measure of how many other documents contain the phrase.

The candidate phrases from each document are considered and used to construct a Navie Bayes Classifier that predicts whether or not a given phrases is a key phrases based on its distance and TF x DF attributes. Jai – Bing wang et al. [2005] have proposed a key phrase extraction algorithm inspired by the protein biosynthesis processes. A document considered as an individual from a population, a document corpus, and document are composed of passages, divided into sentences built upon words. Following analogy, they hypothesized that two documents written in the same language or semantically related would show similar document genomes. The document genomes are extracted and then translated into significance proteins. Khosraw Kaikhah

*** MR. SUNIL MANOHAR REDDY. K**
Assistant Professor
Department of CSE
Aryabhata Institute of Technology & Science,
Hyderabad, AP, India
Ph. No: 91- 9866482481
E- Mail: sunil186@gmail.com

[2004] has proposed a model for key phrase extraction based on supervised machine learning and combinations of the baseline methods. Experiments show that a combination of a relatively high number of baseline methods is very successful for academic papers. Kin Keng Lai et al. [2006] have proposed a phrase extraction algorithm based on supervised learning.

A document is treated as a set of phrases, which must be classified as either positive or negative examples of key phrases based on examination of their features. In this papers we can make up the deficiencies of previous models and are proved to be more effective, rebuilt, and preside. As a well trained 3 layered Neural Network gives accurateness up to 100% with the help of the learning technique model can be trained to work over a more general set of documents.

ANALYSIS

The whole problem basically is solved using Neural Networks. There are two ANN which are used for finding out the relevance and for finding out the summary too respectively.

Finding Relevant Document

Phase 1 is to find out the relevance of documents with respect to query. In this paper we propose to key phrase extraction approach based on Back propagation. In order to determine whether or not a phrase is a key phrase, the following features of a phrase in a given document are considered the Terms Frequency TF, and the Inverted Document Frequency IDF, whether or not the phrase appears in the title or heading of the given document, and its distribution in the paragraphs of the given document.

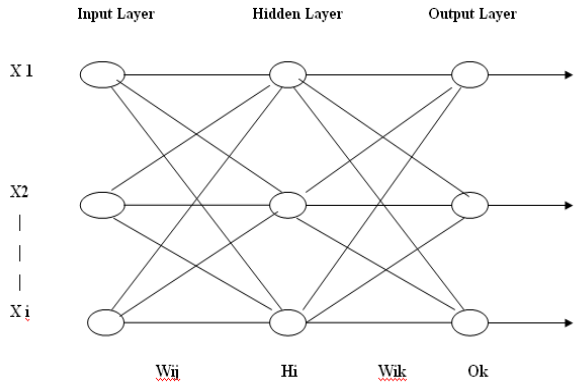


Fig1: A Multilayer Feed – Forward Neural Network

These features of a phrase are input to a multilayer Feed – Forward Neural Network and with which key phrases are selected using back propagation. The Back propagation algorithm performs learning as a multilayer Feed-Forward Neural Network. An example of such a network is shown in Figure 1.

A training sample $X=[x_1, x_{2,\ldots\ldots}x_i]$ is feed to the input layer weighted connections exit between each layer, where W_{ij} denotes the weight from a unit of j in are layer to a unit i in the previous layer. The inputs correspond to the attributes measured for each training sample. The inputs are feed simultaneously into a layer of units making up the input layer. The weighted outputs of these units are in turn fed simultaneously to a second layer of neuron like units, known as a hidden layer and so on. The number of hidden layers is arbitrary although in practice, generally only one is used. The weighted outputs of the last hidden layer are input to units making up to the output layer, which events the networking prediction for given samples.

Before applying the Neural Network to key phrase extraction features for judging whether or not a phrase is a key phrase should be defined. We have the following consideration and hypothesis when deciding to adopt what features to determine a key phrase. TF and IDF are standard information retrieval metrics that estimate have specific a phrase is to document, title and headings are the most common summaries in documents and generally describe the main issue discussed in this paper. Therefore we use the following four features to determine whether a phrase is a key phrase or not, the normalized term frequency TF, whether or not as phrase appears in the title or heading THS, and the normalized paragraph distribution frequency PDF.

TF: The normalized frequency of the phrase in the given document for a phrase i and a document "d". Let n_i be the number of times the phase '*i*' is mentioned in the document d, then the TF of *i* is computed by the formula.

$TF = n_i [\max_{led}, n_i]$

IDF: For a phrase i and a document set with size Q, let q_i be the number of document in

which the phrase 'i' appears then the IDF of i is computed by the formula IDF.

IDF = log [Q/ q_i]

PDF: PDF is a structural measure feature for a phrase 'i' and a document d, let m_i be the number of paragraphs of the document d in which the phrase 'i' appears, then the PDF of 'i' is computed by the formula

PDF = mi [max_{lled} m_i]

THS: Existence of the phrase in the title or heading of a given document of the phrase appears in the title or heading of a given document. THS is equal to 1 otherwise 0.

Therefore we have the neural networks model of key phrase extraction as shown in the Fig 2.

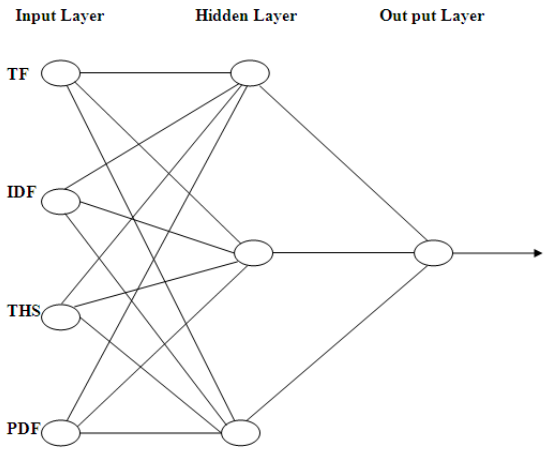


Fig 2: Neural Networking Key phase Extension Model

In the input layer, there are four neurons corresponding to a phrase's four features in a given document on the output layer there is only one neuron corresponding to whether or not a phrase is a key phrase, in the training stage if a phrase is key phrase, and then O_i is equal to 1 otherwise 0. In the key phrase extraction stage, if the output O for a feature vector input of a phrase i is more than 0.5 then i is extracted as a key phrase, otherwise i is not a key phrase.

The Phase 2 is to find out the summary of the found out relevant document. That is also done using ANN basically prepared to adjust itself in a way to give us the summary. The first step in summarization by extraction is the identification of important features. There are two distinct types of features: non– structured features and structured features. One group of researcher utilize only non-structured features such features include sentence length, sentence location, term prominence, presence of words occurring in title, and presence of proper names. On the other hand a group of researchers attempt to exploit structural relation between units of consideration.

For the summarization purpose we convert a sentence into a set of features. Each document is converted into list of sentences. Each sentence is represented as vector f1, f2- composed of 7 features.

Table 1 – The 7 Features of a Document

Feature	Description
f_1	Paragraph follows title
f_2	Paragraph location in document
f_3	Sentence location in paragraph
f_4	First sentence in paragraph
f_5	Sentence length
f_6	Number of thematic words in the sentence.
f_7	Number of title words in the sentence.

Features f_1 to f_4 represent the location of the sentence within the document or within its paragraph. It is expected that in structured documents such as news articles, these features would contribute to selecting summary sentences. Feature f_5 sentence length is useful for filtering out short sentences such as datelines and author names commonly found in news articles. Feature f_6 the number of thematic words in the sentence, relative to the maximum possible. Finally feature f_7 indicted the number of title words in the sentence, relative to the minimum possible. It is obtained by counting the number of matches between the content words in a sentence and the words in the title. This value is then normalized by the maximum number of matches. This feature is expected to be important because the salience of a sentence may be affected by the number of words in the sentence also appearing in the title. These features may be changed or new features may be added. The selection of

features plays an important role in determining the type of sentences that will be selected as part of the summary and therefore would influence the performance of the Neural Network.

TEXT SUMMARIZATION PROCESS

There are three phases in the process: Neural Network Training, Feature Fusion, and Sentence Selection. The first step involves training a Neural Network to recognize the type of sentences that should be included in the summary. Second step, Feature Fusion prunes the neural and collapses hidden layer unit activations into discrete values with identified frequencies. This step generalizes the important features that must exist in the summary sentences by fusing the features and finding trends in the summary sentences. The third step is sentence selection uses the modified neural network to filter the text and select only the highly ranked sentences. This step controls the selection of the summary sentences in terms of their importance.

Neural Network Training

The first phase of the process in values training the Neural Networks to learn the types of sentences that should be included in the summary. This is accomplished by training the network with sentences in several feet paragraphs where each sentence is identified as to whether it should be included in the summary or not. This is done by a human reader. The Neural Network learns the patterns inherent are sentences that should be included in the summary. We use a three layered feed forward neural networks, which has been proved to be universal function approximate. It can discover the patterns and approximate the inherent function of any data to an accuracy of 100% as long as there are no contradictions in the data set. We use a gradient method for training the network where the energy function is a combination of error function and a penalty function.

The goal of training is to search for the global minima of the energy function. The total energy function to be minimized during the training process is:

$F(W,V) = E(W, V) + P(W, V)$

The error function to be minimized is the mean error

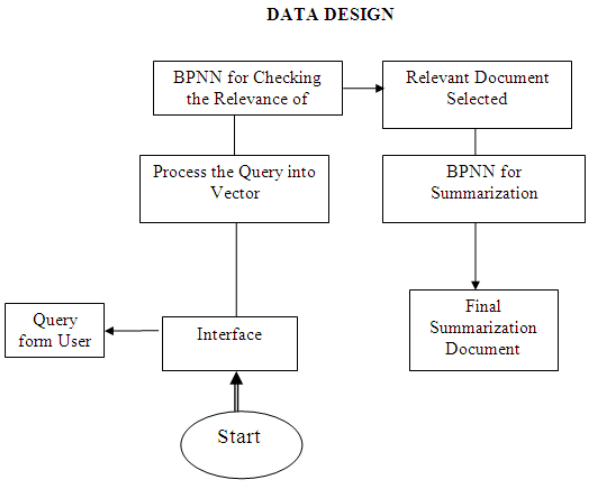
$$E(W, V) = \tfrac{1}{2} \sum_{m=i}^{l} \sum_{k=l}^{n} (O_{lk} - D_{lk})$$

Sentence Selection

Once the network has been trained, proved and generalized, it can be used as a tool to determine whether or not each sentence should be included in the summary. This phase is accomplished by providing control parameters for the derived radius and frequency of hidden layer. Activation clusters to select highly ranked sentences. The sentence ranking is directly proportional to cluster frequency and inversely proportional to cluster radius. Only sentences that satisfy required cluster boundary and frequency of all hidden layer neurons are selected as high ranking summary sentences. These selected sentences possess common features inherent in the majority of summary.

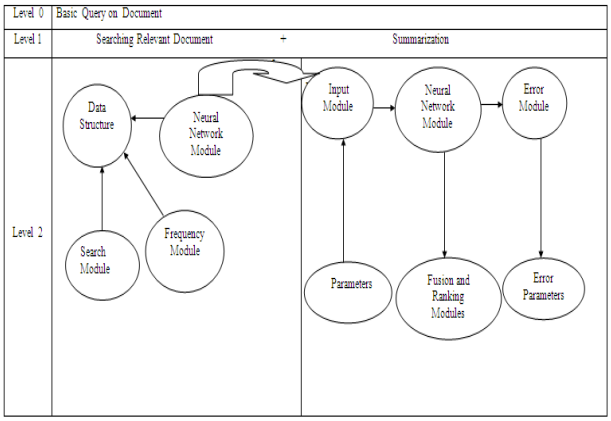
MODELLING

The whole programme is basically designed as in Hierarchal model. That means we can increase specifications to go into functions and modules. First of all on the top we have a query which is supposed to be used to find out the output. Now on the top we have a query which is supposed to be used to find out the output. Now on the more specified level we have two basic branches of the programme that is first the ANN to calculate the relevance of the document regarding to the query. And the second part is the finding out the summary of the relevant document came out. Now these two basic branches can be further sub divided into more sub branches. These sub branches will be functions. The main functions do the tasks as calculation of errors, conversion of the sentence of into vectors. Hence whole program is divided into hierarchy.

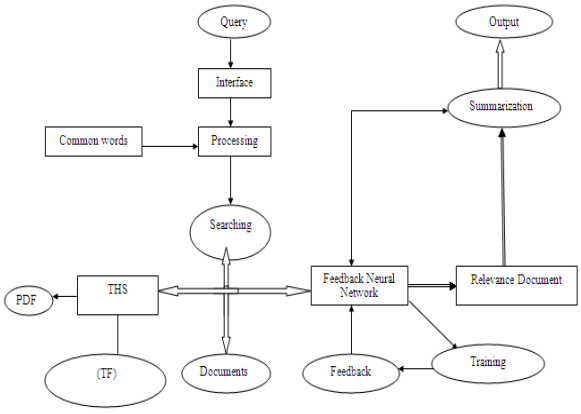


- So we can see, first of all it starts with the query from the user to the interface.
- Query is processed into the vector to move the feed able to the ANN.
- BPNN is applied to find out the relevance of the document.
- Relevant document is selected and its feed into the ANN to calculate the summary.
- Summarization formulae are applied and according the threshold value sentences are selected which are to be included in the summary.
- Final summarized document is prepared and is moved to user as output.

ARCHITECTURE DESIGN



COMPONENT LEVEL DESIGN

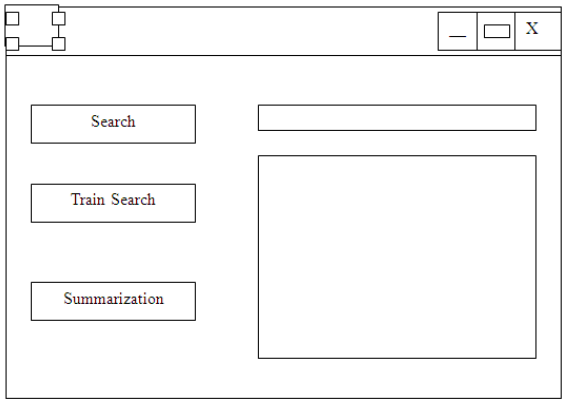


Form searching is used while searching relevant documents and training the search.

Button – Train Search: does training of ANN.
Button – Search: Searches the relevant document.

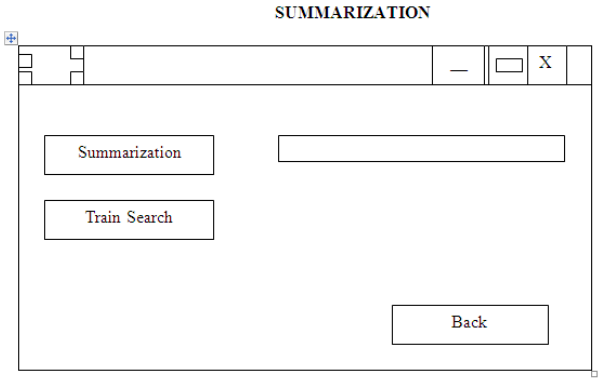
Button – Summarization: Navigates form 2 to summarization part.

INTERFACE DESIGN



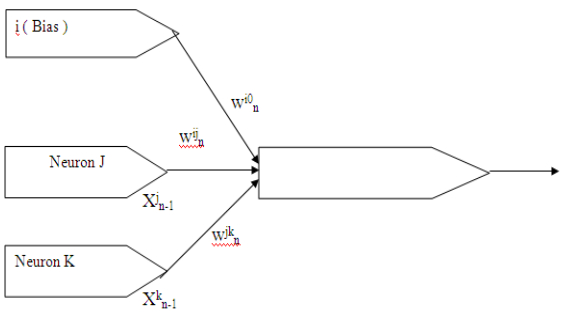
Form summarization is used while summarization of documents and training the summarization.
Button – Train Search: Does training of ANN
Button – Summarize: Searches the document given in TextBox 1.
Button – Back: Navigates to form 1 to searching part.

SUMMARIZATION



NEURAL NETWORK IMPLEMENTATION

Previous Layer	Current Layer
[That is (n+1) th Layer]	[That is nth Layer]
There are C_{n-1}	There are Cn
Neurons in the layer	Neurons in the layer.



Where

$$Y^{i}_{n} = \sum_{i=0}^{1} w^{i0}_{n} \cdot X^{i}_{n-1}$$

and F( ) is the equating function.
X^{i}_{n} is the output of the ith neuron in layer n.
X^{i}_{n-1} is the output of jth neuron in layer n-1.
X^{k}_{n-1} is the output of kth neuron in layer n-1.
w^{ij}_{n} is the weight that the ith neuron in layer n applies to the output of the jth neuron from layer (n-1).
W^{ik}_{n} is the weight that the ith neuron in the layer n applied to the output of the kth neuron layer n-1.

$X^{i}_{n} = F(Y^{i}_{n}) = F(\sum_{l=0}^{C_{n-1}} w^{il}_{n} \cdot X^{l}_{n-1})$ is the general feed – forward equation.

EXPERIMANTAL RESULTS

Data Structure implementation has been done in c++ complied with general computer. Artificial Neural Networks (ANN) part and front end has been implemented in C# and .net 2008. Data base used is Microsoft Accesses. Starting Value of weights ranged from -0.5 to +0.5
For the above weights number of lines produced in the summary was zero.

After 100 Iterations The Learning Weights Are As Follows

0.0568094769997215| 0.767351350217993| 0.0131426131055833| 0.142924089010274| 0.785954740767022| 0.117250243774683| 0.0110335433215638| 0.0955804303251604| 0.324243194911415| 0.948557456424796| 0.425760356314596| 0.817320798491337| 0.968365672891918| 0.191883458424532| 0.953557801748066| 0.337514840182593| 0.175180144562234| 0.298549410612996| 0.77445901307557| 1.01464369898248| 0.597650302723465| 0.288714156113868| 0.658574340578659| 0.471094434421505| 0.82866477532322| 0.684887404917459| 0.897884419942798| 0.131677865536698| 0.41222449157252| 0.760270405036117| 0.0570462550542181| 0.317065814702556| 0.983619029148712| 0.312912534561404| 0.586077355459592| 0.798531603026977| 0.000668451608778225| 0.126660098108167| 0.773423143769338| 0.104825834161831| 0.00154274937516922| 0.0692580824492356| 0.311769033414546| 0.932293465522634| 0.413237759316914| 0.804896388878412| 0.955789380195175| 0.165561110548608| 0.941183640251235| 0.321250849280493| 0.162657547564549| 0.286125001000155| 0.761882720378826| 0.988321351106572| 0.585176141226634| 0.272450165211768| 0.646051743580976| 0.458670024808664| 0.816088482626476| 0.658565057014533| 0.885410258445967| 0.115413874634589| 0.40469985215957| 0.74784599543129| 0.0444699623574851| 0.290743466826633| 0.971144967651881| 0.296648543659304| 0.573554758461908| 0.786107193414052| 0.000329413287625 37| 0.1259391630002415| 0.286804469897738| 0.03662880394456 31| 0.299165175511313| 0.0298385876498249| 0.376343884279076| 0.0753608474815749|

We observed that from the above weights the summary produced is about ¼ th of the document. After Further Iterations The Current Weights Are As Follows

0.*165099239276757| 0.485119134358364| 0.422726489878206| 0.386750176915324| 0.73668399813931| 0.964231715133878| 0.0724115602445788| 0.221056905157162| 0.0333903474734955| 1.07526655237959| 0.6710791898341 03| 0.0755907894317067| 0.614119916732013| 0.795480346156557| 0.938923315660308| 0.555602930465674| 0.982578858730069| 0.198942461687784| 0.812877230482119| 0.851310009012190| 0.80368435940296| 0.1087496981 52956| 0.568599745900997| 0.9932470257217 12| 0.304063236239858| 0.702432179163927| 0.879086637891184| 0.629356259493971| 0.6 41621850269087| 0.325720917198064| 0.2696 21700397048| 0.192670011461761| 0.1406234 3319298| 1.04020977646173| 0.455879742496926| 0.665905551907919| 0.418829981614718| 0.382446279486618| 0.732002609910715| 0. 960937108577259| 0.0647452757877446| 0.21 6335105702784| 0.0294938392100048| 1.0709 6265495086| 0.666397801605508| 0.0722961828750933| 0.606453632275175| 0.7907585437 02179| 0.935026807396811| 0.5512990330369 62| 0.977897470501473| 0.195647855131169| 0.805210946025281| 0.846588206557815| 0.7 99787851139463| 0.104445800724247| 0.5639 18357672402| 0.989952419165093| 0.2963969 51783022| 0.697710376709549| 0.8751901296 27687| 0.625052362065259| 0.6369404620404 92| 0.322426310641451| 0.261955415940212| 0.187948209007383| 0.136726924929488| 1.0 35905879579033| 0.451198354268337| 0.6626109 453513| 0.167217743632587| 0.093292658260 2336| 0.20291961602136| 0.195036436734023| 0.0978781368907781| 0.288559434974699| 0 .332038740898236| 0.0800081534834757*

CONCLUSION AND FUTURE RESEARCH

This paper has successfully implemented the Artificial Neural Network approach for searching the relevant documents and summarization of documents. Basic idea is to find out the key phrases of the documents or retrieving the important information from the documents. We use not only the TF (Term Frequency) and PDF (Paragraph Distribution Frequency) but also the structural features to determine whether a phrase is a key phrase or not. The experimental results show the performance of the information retrieval system has been greatly improved with the learning implying that the proposed ANN technique provided are effective solution to text information retrieval. To increase the effectiveness of the paper linguistics can be useful to improve the summary structure by making the summary more meaningful.

REFERENCES

1. Abduladheem A et al. [2005]: *Hybrid wavelet-network neural /FFT nural phoneme recognition.* Proceedings of the 2nd International Conference on Information Technology. Al-Zaytoonah University of Jordan.

2. Agarwal KK et al. [2004]: *A neural net-based approach to test oracle.* ACM SIGSOFT, 29, 1-6.

3. Arbib M A et al.[2003]: *The Handbook of Brain Theory and neural networks*, Cambridge, MIT Press

4. Archer et al. [1993]: *Application of the back propagation neural network algorithm with monotonic constraints for two group classification problem.* Decision Sciences, 24(1),60-75.

5. Atalla M J et al. [1996]: *Model updating using neural networks.* Virginea Polytechnic Institute and State University.

6. Chen J R et al . [1990]: *Step size variation methods for accelerating the back propagation algorithm.* Proceedings of the International Joint Conference on Neural Networks, Washington, DC, 601-604.

7. Farrel K R et al. [1994]: *Speaker recognition using neural networks and conventional classifiers.* IEE Trans.on Speech and Audio Proc., 194-205.

8. Franzini MA et al.[1987]: *Speech recognition with back propagation.* Proceedings of the IEEE/Ninth Annual Conference of the Engineering in Medicine and Biology Society, Boston.MA, 9,1702-1703.

9. Goffe W L et al. [1994]: *Global optimization of statistical functions with simulated annealing.* Journal of Econometrics, 60 , 65-99.

10. Grzeszczuk R et al.[1998]: *Neuron animator fast neural network emulation and control of physics based models.* New York, USA, ACM Press, 9-20.

11. Guo F et al.[1999]: *Finite element analysis bases Hopfield neural network model for solving nonlinear electromagnetic field problems.* IJCNN '99, (6), 4399-4403.

12. Haykin S et al. [1994]: *Neural networks:* A comprehensive foundation. Prentice-Hall, New Jersey, USA.

13. Holt et al . [1990]: *Convergence of back propagation in neural networks using a log-likelihood cost function.* Electron Letters, 26, 1964-1965.

14. Hsiung JT et al. [1990]: *Should back propagation be replaced by more effective optimization algorithms.* Proceedings of the International Joint Conference on Neural Networks, (IJCNN), 7,353-356.

15. Izui et al . [1990]: *Analysis of neural networks with redundancy*. Neural Computation 2.226-238.

16. Jr A J Metal.[1994]: *The Numerical Solution of Linear ordinary differential equations by feedforward neural networks.* Math. Comput Modelling, 19,1-25.

17. Lagaris I et al.[1998]: *Artificial Neural networks for solving ordinary and partial differential equations.* IEEE Transactions on Neural Networks , 9, 987-1000.

18. Laurene Fausett [1994]: *Fundamentals of Neural Networks*. Prntiece Hall, Englewood Cliffs, New Jersey.

19. Lee et al . [1991]: *Improvement of function approximation capability of back propagation neural networks.*

Proceedings of the International Joint Conference on Neural Computation. 1, 541-551.

20. Liang Y C et al.[2001]: *A neural-network-based method of model reduction for the dynamic simulation of memes..* Journal of Micromechanics and Micro Engineering, 11, 226-233.

21. Lin K et al.[1990]: *A counter-propagation neural network for function approximation.* Man and Cybernetics, IEEE, 382-384.

22. Low T S et al.[1992]: *The use of finite elements and neural networks for solution of inverse electromagnetic problems.* IEE Transactions on Magnetics,28, 2811-2813.

23. Matsuoka et al . [1991]: *Back propagation based on the logarithmic error function and eliminating of local minima. Proceedings of the International Joint Conference of Neural Networks.* Singapore, 2, 1117-1122.

24. Peterson C et al. [1989]: *A new method for mapping optimization problems onto neural networks.* International Journal of Neural systems,1,3-22.

25. Rowley H et al . [1998] : *Neural network - based face detection.*

26. Salchenberger et al . [1992]: *Neural networks. A new tool for predicting thrift failures.* Decision Sciences, 23, 899-916.

27. Shag et al . [1996]: *Global optimization for neural network training.* Computer, March, 45-54.

28. Sietsma et al . [1991] : *Creating artificial neural networks that generalize*. Neural Networks,2, 67-79.

29. Sun X et al.[2003]: *Solving partial differential equations in real-time using artificial neural network signal processing as an alternative to finite-element analysis.* International Conference of Neural Networks and Signal Processing, 1, 381-384.

30. van Ooyen et al . [1992]: *Improving the convergence of the back propagation algorithm.* Neural Networks 5, 465-471.

31. Werbos P J et al.[1990]: *Back propagation thought time: What it does and How to do it.* IEEE, 1550-1560.

32. Werbos, P et al . [1993]: *The Roots of Back propagation, From Ordered Derivatives to Neural Networks& Political Forecasting.* John Wiley, NY.

33. Zurada jacek M [1992]: *Introduction to Artificial Neural Systems*. West Publishing, St.Paul , New York.