Arabic Document Image Classification Using Neural Networks

تصنيف الوثائق العربية باستخدام الشبكات العصبية

Abdallah Ali Al-Khorabi
Associate Professor
Sana'a University,
Sana'a, Yemen Republic
P. O. Box 1341, Fax 967-1-250514
a_alkhorabi@yahoo.com

Mohamed Abduallah Mansour
Postgraduate Student
University of Science and Technology
Sana'a, Yemen Republic
P. O. Box 1341

ملخص البحث

نظام تصنيف الوثائق العربية بالشبكات العصبية NNADICS هو مصنف وثائق عربية متكيف. بتدريب NADICS على عدد من صور الوثائق ذات الأنواع المختلفة فإنه يتصرف كمصنف للوثائق تبعاً لأنواعها وذلك لمقدرته على التمييز بين هذه الأنواع المختلفة. في هذا البحث تم تصميم NNADICS, بناءه, تجربته, وتقيمه. بعد تدريبه, يستخدم NNADICS في مرحلة التطبيق لتصنيف الوثائق المدخلة إليه. قبل تصنيف الوثائق يجب مسحها, معالجتها معالجة أولية, تحويلها إلى وثيقة ثنائية, ومن ثم إدخالها إلى NNADICS لتصنيف كل من محتوياتها إلى أحد الأصناف الأساسية "نص", "رسم خطي" ، أو "صورة". لقد حقق NNADICS معدل تصنيف متوسط قدره ٨٦%.

Abstract

The Neural Network Arabic Document Image Classification System (NNADICS) is an adaptive Arabic document classifier. By training NNADICS on a number of different document image types, NNADICS behaves as a multiple classifier, since it is capable for distinguishing between multiple document image types. NNADICS is designed, built, tested and evaluated. After training NNADICS a document image is applied to the system for classification. Before that the document image is scanned, pre-processed and binarized, and then applied to NNADICS to classify its contents to text, geometric, or photographic image type. NNADICS achieved an average of a 86% recognition rate as it is clearly demonstrated.

Keywords : Neural networks, Arabic document images, document image classification.

1. Introduction

Document classification systems has an important role in document image processing and storage. The document media might be in the form of paper (hard copy), or in computer based electronic form. The work described in this paper is

Accepted February 26, 2004.

concerned with the electronic analysis and characterization of paper-based input documents.

The verity of documents is almost unlimited and their form depends on their information content, editing style, and color content. Information content is content which is normally based on text, picture, and geometric types. Text consists of words constructed from a defined set of characters and symbols. Pictures are pixels of different values distributed in a certain form, and geometric are collections of straight or curved lines sometimes intermixed with text. The document image may be color image, gray-level image or binary image (Kasturi [1]).

Normally document processing is used to reduce document storage and transmission time/cost, facilitate document archiving and retrieval, provide easy access to broader user groups, and to facilitate information modification/ manipulation. Beside all these benefits, document processing and image classification is one step towards electronic office realization, which is the goal of research in this field (Al-Khorabi [2]).

Document processing involves functions that change the document image contents, orientation, position, size and/or representation. This may include binrization, noise reduction, skew detection and correction, partitioning, classification, understanding, and/or coding, or any combination of these operations accomplished in a pre-defined sequence. Desktop publishing systems are used to generate coded electronic documents (characters, lines, objects, ...etc.). This coded form is then used to produce hard copy documents. To automatically process hard copy documents it is nesseccary to trnsfer it back to electronic form, this is normally achieved by scanning the hard copy document. The generated electronic document is in a bitmap form (pixels representation) that has large size. To have the bitmap document image contents represented in the same code it had earlier when created using the desktop publishing system (group of objects), the document should be processed. Document image processing make the document in an acceptable form, more understandable, and/or with reduced size. An electronic document office requires document image processing techniques to transfer hard copy documents into perfect electronic documents (Al-Khorabi [2]).

2. Neural Networks

Artificial Neural Networks (NN) - also called parallel distributed processing systems, intended for modeling the organizational principles of the human central nervous system so that it resembles the brain in the following respects (Buchanan [3], Bose [4]):

1- Knowledge acquisition, through a learning process (Bose [4]).
2- Knowledge storing, through inter-neuron connection strengths (known as synaptic weights), (Bose [4]).
3- Knowledge consulting, through knowledge processing algorithms.

Neural networks may not only have different structures and topologies, but also they are distinguished from one another by the way they learn, the manner in which computations are performed (rule-based, fuzzy, even non-algorithmic) (Johansson [5]), and the component characteristics (activation function that

represents the transfer characteristics of the neurons or input/output description of the synaptic dynamics).

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network. Four different classes of *neural network architectures* may be identifed: single layer feedforward networks, multilayers feedforward networks, competitive networks, and, lattice structure networks (Johansson [5]).

A NN has the ability of *learning* from its environment, and to improve its performance through learning, the improvement in performance takes place over time in accordance with some prescribed measure. A neural network learns about its environment through an iterative process of adjustments applied to its synaptic weights and thresholds. Ideally the network becomes more knowledgeable about its environment after each iteration of the learning process (Johansson [5]).

Techniques used for a learning process are: Error-correction learning, Hebbian learning, competitive learning , and, Boltzmann learning (Johansson [5]).

3. NNADICS Design and Implementation

NNADICs consists of the following modules: System Interface (SI), Pre-Processor (PP), Statistics Module (SM), Learning Module (LM), Update NN Parameters Module (UNNPM), Application Module (AM), and, Display and/or Store Results Module (DSRM), (see Fig. 1).

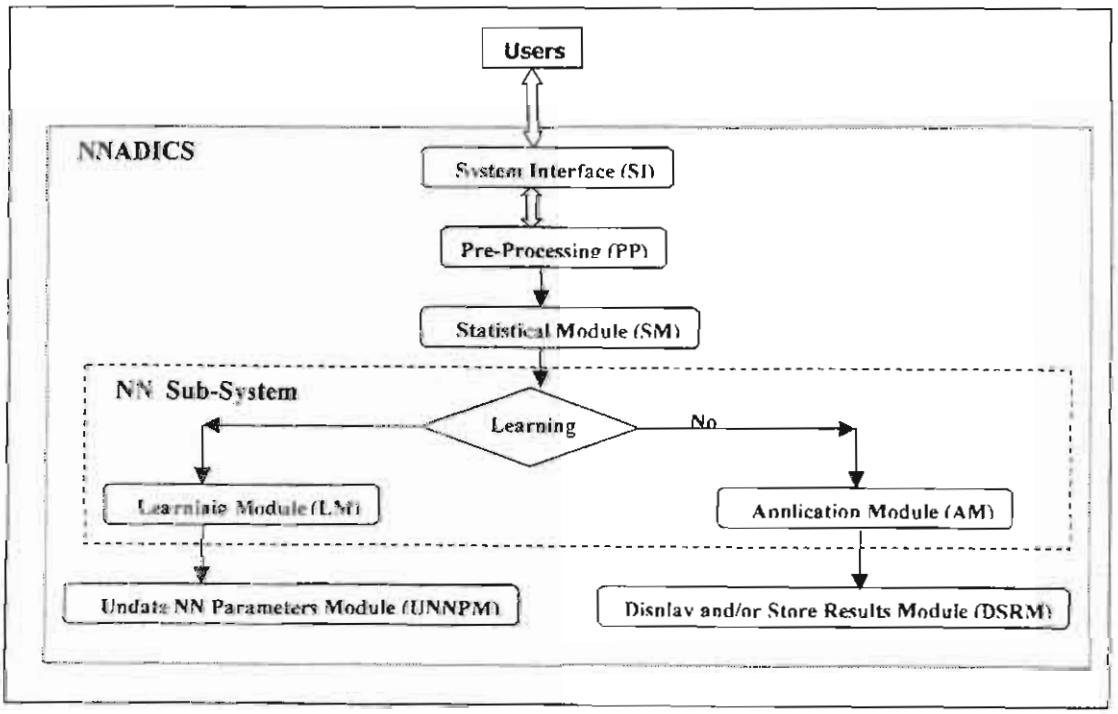

Figure (1) NNADICS Structure

The System Interface (SI) : is responsible for starting the system activities, triggering the system modules, and terminating the system activities.
The Pre-processor (PP) : is responsible for selecting the document, displaying the image, reducing the noise, binarizing and filtering the image, and converting the image from one type to another.
The Statistics Module (SM) : is responsible for obtaining the statistical parameters from the image bitmap, normalizing and storing these statistical parameters.
The NN sub-system within NNADICS include two modules, the Learning Module, and the Application Module. The two modules are back propagation units.
The Learning Module (LM) : is responsible for reading the image normalized statistical parameters, setting the neural network parameters (number of hidden layers, number of neurons, learning rate value , momentum value,...Etc), performing the NN teaching process for the neural network, displaying and storing the best results to be used in the application stage.
The Update NN Parameters Module (UNNPM) : is responible for updating the weighting vector that represents the NN parametrs.
The Application Module (AM) : is responsible for selecting and displaying the application document image, obtaining the bitmap statistical parameters from bitmap image and normalizing, appling the neural network , and deciding about the image type (text , picture or geometric).
The Display and/or Store Results Module (DSRM) : is responsible for presenting and storing the classification process results performed by AM.

The learning process of the back propagation NN sub-system is represented by the algorithm flowchart illustrated in figure (2). During the NN initialization, the network configuration, the synaptic weights, and the neurons activation functions are set. During the learning process for every iteration 'I' a number of input patterns 'P' are repeated. During each input pattern the neurons output are computed, and the connecting weights are updated. Iteration stops when an optimum error is obtained.

4. Results and Discussion

The project is implemented on a 120 Pantuim IBM compatible PC computer using C++ language. As described earlier, the system has two main parts. The first part is the "image processing sub-system" that is responsible for processing the document image. The second part is the "neural network sub-system" that represents the document classifier, it learns, stores, and uses knowledge to classify image.

The performance of the NN sub-system depends on a number of variables. These variables are features selected, number of hidden layers, number of neurons, number of iterations, momentum, learning rate, with or without threshold, number of patterns, number of document image types, number of outputs, and type of activation function. The combination of these variables, using different values gives a great deal of results. However, in order to simplify the neural network application, certain parameters were fixed at a time, and the

values of the remaining parameters were optimized. In this project all variables are fixed except the number of document image types(3 types; text, geometric, and photographic, or 4 types; text, geometric, photographic and large font text types) and the type of activation function are variables. These two variables gave four different combinations (methodoligies) for testing the NN sub-system, these methodoligies are summarized in table (1). Each methodology has been individualy trained on a set of 120 documents, applied on another set of documents, and gave a unique results.

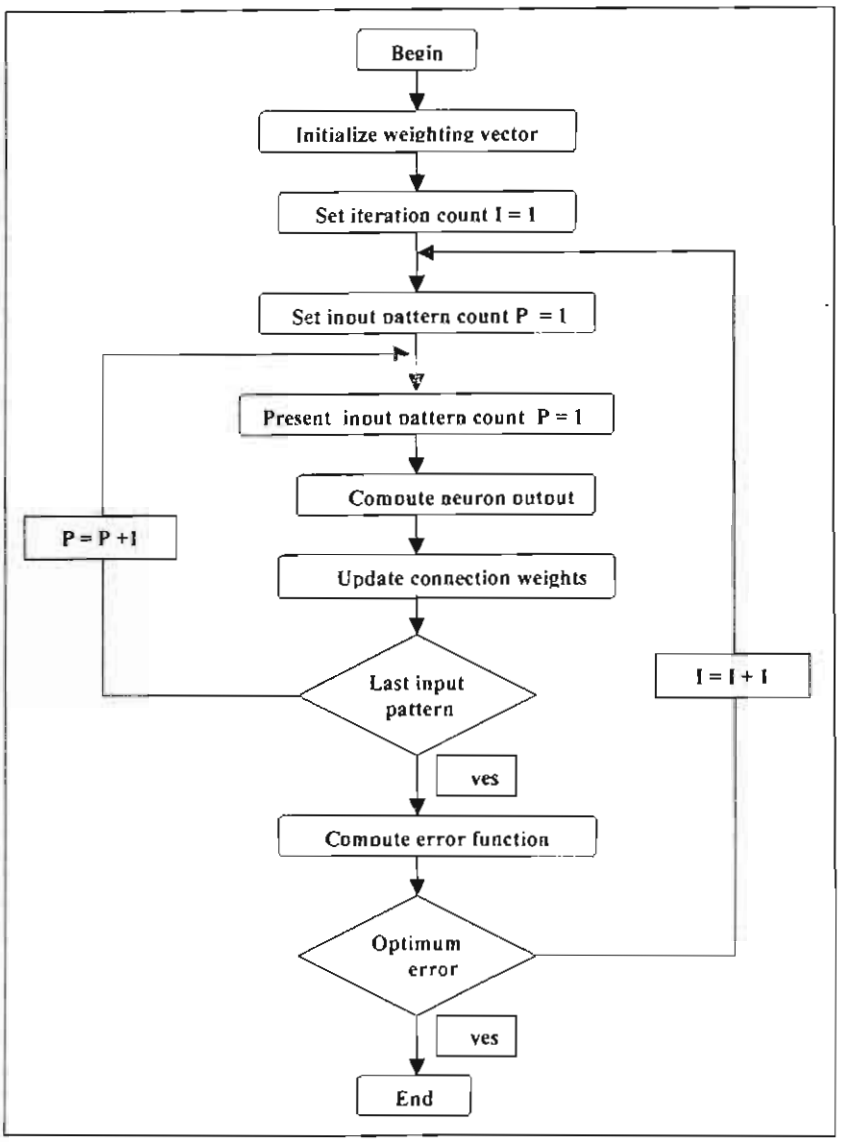

Figure (2) Back propagation learning algorithm flow chart.

Table (1) The four methodologies used to examine the NN sub-system

Method no	Activation function	No of document image types	No of learning iterations	Minimum error
1	Sigmoid	4	3000	0.001111
2	Sigmoid	3	3000	0.001373
3	Hyberbolic	4	2500	0.011355
4	Hyberbolic	3	1500	0.002500

In the four methods the neural network fixed variables all have the same values as follows:

NNADICS has threshold = (yes),
Number of Patterns for learning = 120 patterns.
Number of inputs = 12 neuron input
Number of outputs = 1
Number of hidden layers = 1 layer
Number of neurons in hidden layers = 10 neurons
Total number of Iterations for learning = 3000 iteration
Momentum = 0.89
Learning rate = 0.21

The first method expressed the best result (86% recognition rate), this is summraized in table (2).

Table (2) Results summary for the first method

Document image	Input doc	Correctly defined	Erroneously defined				% Successful	% Successful if large font text & text are the same in test stage
			Picture	Geome-tric	Text	Large Font text		
Pictures	32	29	-	3	0	0	90.63	90.63
Geometric	37	26	1	-	4	6	70.27	70.27
Text	29	25	0	0	-	4	86.21	100
Font Text	22	11	0	3	8	-	50	86.36
Totals	120	91	1	6	12	10	75.83	85.83

Using the first method the system defined correctly 29 document images as picture type from a total of 32 pictures, i.e 90.63% successfully defined pictures. Also it defined correctly 25 images as text from a total of 29 images. This is a second good result for the system i.e. 86.21% successfully defined text. The system is successfully defined 26 document as geometric from a total of 37 images, i.e. 70.27% successfully defined geometric. Finally the system defined correctly only 11 images as large font text from a total 22 images, i.e. 50% successfully defined as large font text. Totally the system defined correctly 91 image from a total of 120 images i.e. 75.83% successfully defined. If no deference is suggested between text and large font text, the results then are improved because the system defines correctly all documents text, and the total result is improved to 85.83% successfully defined.

The 2nd method gave 71% recognition rate, where the 4th method gave 68% recognition rate. The 3rd method was the worst since it gave 48% recognition rate only.

It was found that the classification performance of NNADICS is satisfactory, although tested with a limited set of input images in the experiments. However, extensive tests should be carried out using large number of input images, in order to establish accurately the classification performance of the proposed system.

5. Comparison between NNADICS and CBDI

To evaluate NNADICS performance, its results is compared with those systems designed for the same purpose, i.e. document image classification. CBDI (Classification for Binary Document Image) is a system designed by Abdallah Al-khorabi [1], to identify document image type, the system represents an adaptive nonlinear filter. CBDI is built as 3X3, 5X5, 7X7, or 9X9 non-linear filter, however the best results are experienced with the 3X3 and 5X5 filters. Table (3) shows comparison between NNADICS and CBDI in terms of successful identification rate, a 3X3 filter results for CBDI are selected. From table [3], it is clear that the performance of NNADICS is better than CBDI in text, and picture due to the type of features selected, and the neural network learning method. However NNADICS shows worse results for geometric due to the same reason.

Table (3) Successfully identification rate in CBDI and DICS.

Doc type	CBDI	NNADICS
Text (and large font)	52%	100%
Geometric	91%	70.27%
Picture	72%	90.63%
Totally	71.6%	85.83%

This shows that NNADICS makes an acceptable identification rate and can stand alone as one classifier for identifying different document types, compared to 4 different filters required by CBDI for the same purpose.

6. Conclusion

The main objective of this research project was to develop a system that classifies document images stored as bitmap files. The objective has been achieved with the proposed efficient NNADICS system. More than 90% of the images are captured, read, binarized, statistical information extracted, i.e. accepted as input to NNADICS, and then processed by NNADICS. The failure of 10% of the images to be accepted by NNADICS is due to images large size, new type images, or some other mistakes happened during pre-processing stage.

Currently, the system has performance limitations for documents, which contain equations. Most of the unrecognized or wrongly recognized image is due to the unknown weight matrix of the neural network module.

NNADICS achieved an average of 70% recognition rate. The 1st method expressed the best result 86% recognition rate , the 2nd method gave 71%

recognition rate, where the 4th method gave 68% recognition rate. The 3rd method was the worst since it gave 48% recognition rate only. This good result reflects the system capability, and the neural networks power as high performance classification systems.

7. References

[1] Kasturi R., "Introduction to Document Image Analysis Techniques" short course, The Pennsylvania State University, 1990.

[2] Al-Khorabi, Abdallah A. "A Non-Linear Filter for Document Image Classification" Journal of Science & Technology, university of Science & Technology, Yemen, Vol 2, No. 2, Dec. 1997.

[3] Buchanan B., and Shortliffe E., "Rule-based Expert Systems" , Addison-Weseley Publishing Company, 1984.

[4] Bose N. K., and Liang P. "Neural Network Fundamentals with Graphs, Algorithms, and Applications", Mc GRAW-HILL, 1993.

[5] Johansson E. M., Dowla F. U., and Goodman. D. M. "Backpropagation learning for multilayers feedforward neural networks using the conjugate gradient method". International Journal of Neural Systems, 1992.

[6] International Standardization Organization, "Information Processing - Text and Office Systems - ODA and Interchange Format", ISO-8613, Parts 1,2,4-8, 1989.