Structure Extraction in Printed Documents Using Neural Approaches

Abdel Belaïd, Yves Rangoni

Structure Extraction in Printed Documents Using Neural Approaches

Abdel Belaïd and Yves Rangoni

University Nancy 2 - LORIA, Campus Scientifique, 615 rue du Jardin Botanique, 54600 Villers-Lès-Nancy, France
{abelaid,rangoni}@loria.fr

Summary. This paper addresses the problem of layout and logical structure extraction from image documents. Two classes of approaches are first studied and discussed in general terms: data-driven and model-driven. In the latter, some specific approaches like rule-based or formal grammar are usually studied on very stereotyped documents providing honest results, while in the former artificial neural networks are often considered for small patterns with good results. Our understanding of these techniques let us to believe that a hybrid model is a more appropriate solution for structure extraction. Based on this standpoint, we proposed a Perceptive Neural Network based approach using a static topology that possesses the characteristics of a dynamic neural network. Thanks to its transparency, it allows a better representation of the model elements and the relationships between the logical and the physical components. Furthermore, it possesses perceptive cycles providing some capacities in data refinement and correction. Tested on several kinds of documents, the results are better than those of a static Multilayer Perceptron.

1 Introduction

Automatic structure extraction remains a very challenging problem due to the inherent complexity of documents. For raster images of documents, the gap between physical and logical structure is huge. It is difficult to model the intermediate steps and the relationships between the original image blocks and recognized layout structures, and to maintain consistency between the processing steps in the recognition process. It is also difficult to handle image noise, layout variations and artifacts produced during processing.

In spite of the numerous researches done in this way, the investigation made in this area is prudent:

- recognition has been limited to few structures (less than 10, let say 5 in average), essentially in editorial documents (i.e. books, articles, reports, etc.), often accompanied by a DTD (Document Type Definition), making the recognition more stereotyped;

- recognition methodology has been limited to translating DTD knowledge and its application on the document. The methods were mainly oriented toward context-free grammars and tree or graph comparisons, and often considered as limited in their ability to handle complex situations.

Certainly, the literature provides many approaches to structural recognition, but their application to document analysis is not straightforward and their advantages often equal their drawbacks.

There are two main approaches to document layout analysis: those based primarily on information manipulation, and those based primarily of perceiving features in data. Considering the information manipulation aspect, two sub-categories exist:

- model-driven (e.g. systems using rules or grammars). They use and formalize knowledge well, and are precise and fast but are dependent on an expert to guide their actions. Unfortunately, they do not generalize well, and have been found sensitive to variation and noise;
- data-driven, starting from low-level data. Their classes should represent very well the structure elements, but data description is not easy and the convergence is not assured. However, contrary model-driven methods, they remain very general and flexible as their adaptation to new documents is easier.

Considering the perception aspect, here also two points of view can be distinguished:

- global to local which is often assimilated to top-down approach. The process is based on a segmentation refinement: here the progress seems to be made continuously and safely but if an error is introduced in the beginning, it remains during all the process.
- local to global or bottom-up approaches. These labeling-based methods start from fine to coarse building progressively the context. In this case, a lot of unused features have to be extracted and managed.

As indicated above, all the methods investigated in the literature present some limitations. Hence, the solution that seems to be appropriate for document structure analysis is a hybrid approach in the sense where it mixes both aspects: data consistency and perceptual approaches for the processing methodology.

This paper is organized as follows: section 2 discusses the use of Artificial Neural Network (ANN) approaches in Document Analysis and Recognition (DAR) area, specifically for recognition tasks involving the physical structure of a document. Section 3 focuses on the ANN based solution for logical structure extraction. Finally, section 4 gives some perspectives about the use of ANN in logical structure analysis.

2 Neural networks in document analysis and recognition

2.1 Physical or geometrical layout analysis

In Document Analysis and Recognition (DAR), Artificial Neural Networks (ANN) have been devoted mainly to preprocessing tasks or recognition of small patterns as isolated characters. As detailed by Marinai et al. in [1], such kind of use include binarization, noise reduction, skew detection, and character thinning. The MultiLayer Perceptron (MLP) is used for example in [2] to binarize images for character segmentation. After a segmentation phase based on gray level histogram analysis, the authors feed a MLP with pixel values within a 5×5 windows. In [3], a Self Organizing Map (SOM) and a MLP are applied on the image to classify the pixels according to their gray levels or color values. Another use of ANN is for noise elimination such as in [4] by applying Kalman filtering.

For images representing characters, various ANN models dealing with printed or handwritten scripts have been experimented. Main of them proceeds directly on the images as the inputs are often composed of the image pixel values. Le Cun et al. [5] have provided a very interesting survey on various ANN models related to handwritten words. Similar architectures (convolutional) were used by [6] for handwritten digits recognition, complemented by a SOM to correct the rejections. For each rejected character, a SOM is trained and associated to the MLP to make possible the correction. Garris et al. [7] used an enhanced MLP for the same problem, where the enhancements are focused on neuron activation functions, regularization and Boltzmann pruning.

Hence, these examples show clearly that ANN are able to deal with local variations in a document image during recognition.

2.2 Logical structure analysis

There are few works on logical structure recognition using ANN. Indeed most of the approaches are model-driven. The model contains the description of the physical elements of the document and their associated logical labels. The recognition procedure tries to identify these associations.

Usually, these models are either trees or grammar rules. In both cases, a syntactical analysis procedure is employed to perform the structure labeling [8]. For example, Brugger et al. [9] use a generalized n-gram (with n=3) to represent geometrical relationship between the text blocks, then an optimization method to match the current input with a global model or a sub-tree of this model. Hu et al. [10] use dynamic parsing and fuzzy logic to be more flexible when analyzing the logical structure. A rule-based system is employed by Niyogi et al. [11] with a top-down backward-chaining strategy. Their system "DeloS" handles about 160 rules in three levels for classification, reading order and logical structure analysis.

Although this methodology seems natural as it transcribes a known structure hierarchy of the document and works very well for simple documents, its application on more complex document becomes difficult and causes many errors. In fact, the use of deterministic models fails because of the rigidity in the application of the rules. Furthermore, these kinds of models are often created manually, leading the operator to select and tune himself a lot of parameters. This can explain the limits of such models when applied on real images where the structure is complex and does not fit exactly the general model. The inherent noise of the input image can sometimes introduce errors in the interpretation of the elements.

To face this problem, a data-driven method seems more appropriate. ANN can provide a good solution because they learn from examples, are robust, insensitive to noise and have a generalization capacity. Furthermore, the ANN based solution will avoid the drawbacks of model-based method provided that knowledge must be integrated. Indeed as mentioned in [1], the classical use of MLP is not sufficient to tackle the problem. Existing methods are focused on the tuning of the MLP to resolve the problem and not really on new architectures. The idea is to use a model which is not only based on MLP but which can integrate the structural aspect of the problem.

Two types of ANN can be considered:

- static ANN (with MLP configurations) can adapt to structured patterns by cleverly integrating the structure in the topology as made by [12];
- dynamic ANN by transforming the temporal chain in structured version as in [13, 14].

These two architectures will be described in the following.

3 Neural networks for structured patterns

Neural networks are suitable to handle classification problems with static information. For several applications including logical structure analysis, the patterns to deal with are in a structured domain. ANN are designed to classify unstructured patterns and cannot deal directly with tree or graph structures. However, we can find models which can take into account the structured patterns either in a dynamic or a static version.

3.1 Static networks

The best know type of static network is the MLP because it is the easiest to implement, its training algorithm is well known and it has been applied successfully to different kinds of data.

As mentioned in section 2, its use is generally devoted to physical element recognition where there are no or few structures to interpret. All research done on this kind of network does not focus on the model topology but more on the way MLP is applied to the specific task.

3.2 Dynamic networks

In order to take into account the temporal dimension of some real-world problems, a dynamic network can be applied rather than a static one.

The Time Delay NN (TDNN) is a straightforward solution that unfolds the time sequence onto several static models through a Tapped Delay Line (TDL) [15]. The same approach can be done with the RBFN (Radial Basis Function Network) to take into account the temporal dimension [16].

Feedback dynamic methods as recurrent networks integrate feedback contrary to feed forward systems. The learning is recursive and consequently more complex to undertake. The output Feedback based systems use the network outputs in a second TDL besides the classical one's as in TDNN [17].

State feedback methods feedback connections between neurons are introduced: each neuron contributes to all components of the state vector.

Time Hopfield Networks (THN) [18] are mono-layer networks in which all the possible interconnections are used. The Continuous THN (CTHN) is well known as it can handle oscillations or even chaotic phenomenon. The Discrete THN (DTHN) is similar to the previous one's but here the activation function is hard limiter and not a sigmoid.

Continuous Time Recurrent Neural Network (TRNN) [19] is quite similar to CTHN: there is one layer of fully connected neurons, the difference is in the differential equation managing the dynamic process. The same analogy is done for the Discrete Time Recurrent Neural Networks (DTRNN) with its hard-limiter function [20]. The DTRNN can simulate deterministic finite automate. In such ANN, the training stage is more complicated and two main solutions can be seen in the literature. The first totally converts the network into a feed forward version by unfolding the network over time. The second method consists in the use recursive version of the gradient descent.

3.3 Dynamic networks for structured patterns

The previous dynamic networks have been developed to process sequences of patterns but adaptations to structured patterns can be found in the literature.

Küchler and Goller [13] propose an approach to classify structured patterns. The patterns considered are those represented by a Direct Acyclic Graph (DAG) or by a Rooted LDAG (i.e. a graph with only one root node, i.e. one node with in degree zero). The network topology, in a static view, corresponds to the folding of the DAG in a feed forward MLP. The first layers compute the folding part (i.e. inputs through DAG representation) and the following layers constitute the transformation part (Fig. 1).

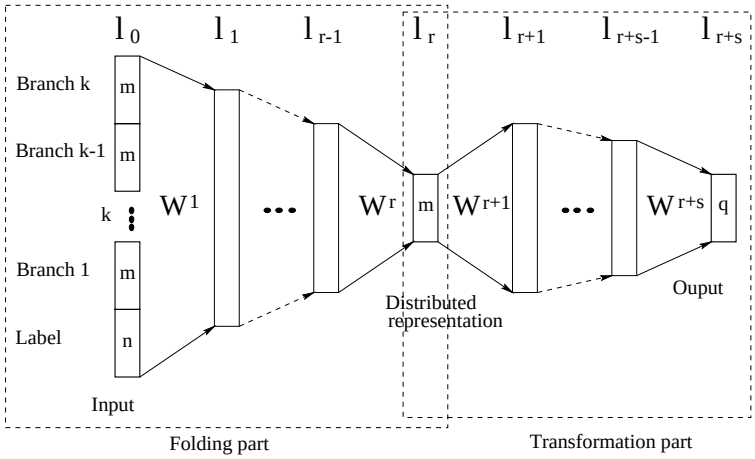

Fig. 1. Küchler et al. generic folding architecture

The input contains the vertex labels for distributed representation of DAG. The last layer corresponds to the task specific output. The network dynamics are defined as follows:

$$o_j^{(l+1)}(t) = f\left(\sum_i o_i^{(l)}(t) w_{ij}^{(l+1)} + \theta_j^{(l+1)}\right) \tag{1}$$

where $o_i^{(l)}(t)$ is the output of neuron i in the layer l at recursion stage t, $\theta_i^{(l)}$ is the bias associated with neuron i at layer l, $w_{ij}^{(l+1)}$ the weight of the connection between neuron i in layer l and neuron j in layer $l+1$ and f the sigmoid function.

The authors use a modified version of the Back-Propagation Through Time (BPTT) algorithm where the structure of a labeled DAG is incorporated in the error measurement

$$E = \sum_{i=1}^{p} \sum_{j=0}^{q-1} \frac{1}{2} \left([t_i]_j - o_j^{(r+s)}(root(s_i))\right)^2 \tag{2}$$

where *root* denotes the function mapping structures to their root nodes, s_i are in the general symbolic domain and t_i define by $\Xi(s_i) = t_i$ with Ξ being the function to be approximated.

Thanks to a special gradient descent technique called Back-Propagation Through Structure (BPTS), the network can be trained. Experimentation has been done on 2-classes classification problems on logical terms. The results are very promising: 99% for the training and 98% for the test.

Sperduti et al. [14] propose another dynamic NN extended to structural patterns. The main idea is to generalize a recurrent neuron in a "Generalized

Recursive Neuron" (GNR). The approach is different from the standard one which focuses on the tree-structure encoding in a fixed input vector. The GRN considers the outputs of the unit for all the vertices which are pointed by the current input vertex.

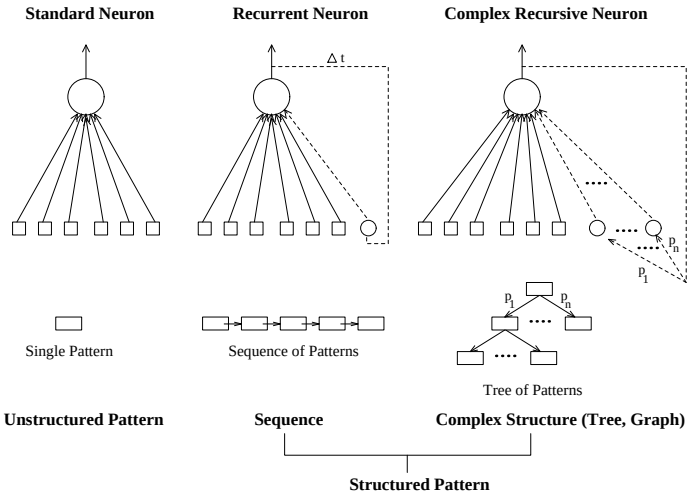

Fig. 2. Neuron models for different input domains

Figure 2 shows the standard models for structured and unstructured of patterns, on the right side, the presented configuration can represent any graph or tree structure thanks to the proposed GRN. Usually, in a standard neuron the output is given by:

$$o^{(s)} = f\left(\sum_i w_i I_i\right) \tag{3}$$

where f is non-linear function such as the sigmoid, I the input vector and w the weight vector. In the recurrent version, the output depends on time:

$$o^{(r)}(t) = f\left(\sum_i w_i I_i(t) + w_s o^{(r)}(t-1)\right) \tag{4}$$

where $o^{(r)}(t-1)$ is the previous output at time $t-1$ that is weighted by w_s and added to the activation formulae. In the GRN the output $o^{(g)}(x)$ depends on a vertex in the graph and computed recursively on the output performed for all the vertices pointed by it. The output is given by:

$$o^{(g)}(x) = f\left(\sum_i^{N_L} w_i l_i + \sum_{j=1}^{\text{out\_degree}_X(x)} \hat{w}_j o^{(g)}(\text{out}_X(x,j))\right) \tag{5}$$

where x is a vertex of a graph X, N_L the unit number encoding the label l attached to the current input x, $\hat{w}_j$ the weights on the recursive connections and $out_X(x, j)$ the out nodes of the graph X attached to the node x. The graph is encoded to fit with the GRN representation (Fig 3).

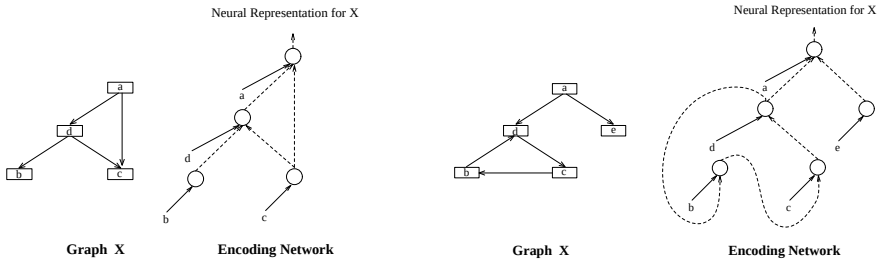

Fig. 3. On the left side, the network encoding for an acyclic graph is shown. On the right side, the encoding network for a cyclic graph is shown

The authors have extended five supervised algorithms for ANN to handle the GRN: back propagation through structure, real-time recurrent learning, LRAAM-based networks and simple recurrent networks, cascade-correlation for structures, and neural trees.

For example the Back propagation Through Structure (BPTS) is simply as in [13] an expression of the back propagation through time. The trick consists in unfolding through time the recurrent network in an equivalent and fully feed forward network. As a consequence, the transformed network can be trained using the back propagation algorithm. For the GRN, the network is decomposed into two parts: an encoding function Ψ and a classification function Φ such as

$$o(X) = \Phi(\Psi(X)) \tag{6}$$

Using standard back propagation, the weights are modified using (7) and (8):

$$\Delta \mathrm{W}_\Phi = -\eta \frac{\partial \mathrm{Error}(\Phi(y))}{\partial \mathrm{W}\Phi} \tag{7}$$

$$\Delta \mathrm{W}_\Psi = -\eta \frac{\partial \mathrm{Error}(\Psi(y))}{\partial \mathrm{y}} \frac{\partial y}{\partial \mathrm{W}_\Psi} \tag{8}$$

Two cases must be treated separately in the case of a DAG and graphs with cycles. With DAG, Küchler et al. [13] algorithm can be used. The training is computed by back propagation of the error from the feed forward network through the encoding network of each structure. For cyclic graphs, recurrent back propagation must be considered.

Real-Time Recurrent Learning can also be extended. For DAG the extension does not present particular problems, the cyclic graphs are more difficult

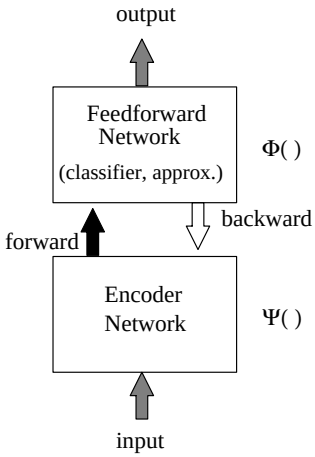

Fig. 4. The encoding part of the NN passes encoded structures to the classifier, the classifier returns the deltas used by the encoder to adapt its weights

to extend and require different situation according to global cycle presence. Thanks to the "Strongly Connected Component" and "Component Graph" notion, the cyclic graphs can be considered as many acyclic graphs and solved more easily.

Labeling Recursive Auto Associative Memory (LRAAM) [21], another model to represent labeled structures, is trained by a combination of a supervised method and an unsupervised one. For structured pattern recognition, Sperduti uses this LRAAM to produce a compressed representation of the structure, then he uses an MLP to carry out the classification.

GRN can be also extended to the cascade-correlation algorithm developed by Fahlmane and Lebiere [22]. This model generates a standard ANN by using an incremental approach for classification of unstructured patterns. The starting network $\mathcal{N}_0$ is a network with no hidden nodes trained using LMS. If $\mathcal{N}_0$ cannot resolve the problem, a hidden unit u_1 is added so that the correlation between the output of the unit and the residual error of the network $\mathcal{N}_0$ is maximized. The weights of $u1$ are frozen and the remaining weights are retained. If the retained network $\mathcal{N}_1$ cannot solve the problem, the network is further grown by new hidden units which are connected (with frozen weights) with all the inputs and previous hidden units. The resulting network is a cascade of nodes. Sperduti et al. extend the output of the k^{th} to GRN using (9) where $w_{(v,j)}$ is the weight of the k^{th} hidden unit associated with the output of the v^{th} hidden unit computed on the j^{th} component pointed by x. $\bar{w}_q^{(k)}$ is the weight of the connection from q^{th} hidden unit and the k^{th} hidden unit. Learning is performed as in standard cascade-correlation with the difference that the equations are recurrent on the structures.

$$
\begin{aligned}
o^{(k)}(x) &= f(\alpha + \beta + \gamma) \\
\alpha &= \sum_{i}^{N_L} w_i^{(k)} l_i \\
\beta &= \sum_{v=1}^{k} \sum_{j=1}^{\text{out\_degree}_X(x)} \hat{w}_{(v,j)}^{(k)} o^{(v)}(\text{out}_X(x,j)) \\
\gamma &= \sum_{q=1}^{k-1} \bar{w}_q^{(k)} o^{(q)}(x)
\end{aligned}
\tag{9}
$$

GRN can also be adapted to neural trees. The advantage of this kind of model is to build the structure on the fly and not be restricted to a static structure as in a feed forward network. New classes are learnt incrementally with supervised or unsupervised training. The extension of this network to a structured version is done by analogy: each discriminator associated with each node of the tree is replaced by a generalized recursive discriminator.

Experiments on GRN have been carried out on several classification tasks. The data are randomly generated. For small size structures (tree depth between 3 and 6) the results obtained on classification problems are nearly perfects for the training (near 100%) and very good for the testing (average of 95% and sometimes 100% with a good choice of hidden units and learning parameters).

There is more and more work about dynamic networks, and although they are not oriented directly towards logical structure extraction, it seems that the previous contributions can be easily extended to this kind of application. However, these recurrent techniques present some drawbacks compared to static ANN:

- they are time and memory consuming;
- the convergence is more difficult to reach as there are more local minima;
- the convergence is slower, decreasing the training step make the training more and more slow;
- there are more numerical errors that create serious repercussion on network's convergence;
- the gradient explosion occurs quickly on long sequences. The more the sequence is long and the more the global error is large.

On top of that, the presented dynamic neuronal methods can deal with logical structure recognition but are not sufficient. In addition to the inherent limitations, the structures to be recognized need to be known and fixed throughout the training and recognition. This it is not necessarily true in real world applications.

3.4 Transparent neural network for handwriting recognition

As seen in previous section, dynamic ANN can be extended to deal with structured patterns. The well known static ANN such as MLP can be also improved to handle structured patterns.

In [23], Côté et al. propose a perceptual model, Perceptro, for handwritten word recognition. The proposed method is based on McClelland and Rumelhart's reading mode [24]. Two questions are explored: what kinds of features are detected and how the information concerning the meaning of a word is accessed. The key is to integrate a knowledge representation in the network. Indeed, trying to use a standard network with distributed representation, such as the MLP, cannot deal correctly with handwritten recognition. That is why in [23] a network with local representation is chosen for the kernel of their approach. The Interactive Activation Model of [24] is a neural network with local knowledge representation, parallel processing of information, and gradual propagation of activation between adjacent levels of neurons. The original activation is given by

$$\begin{aligned} A_i(t+\delta t) &= A_i(t) - \theta_i(A_i(t) - r_i) + E_i(t) \\ E_i(t) &= \begin{cases} n_i(t)(M - A_i(t)) & \text{if } n_i(t) > 0 \\ n_i(t)(A_i(t) - m) & \text{if } n_i(t) < 0 \end{cases} \\ n_i(t) &= \sum_j (\alpha_{ij} - \beta_{ij}) a_j(t) \end{aligned} \tag{10}$$

where θ_i is a decreasing constant, r_i the activation threshold, $E_i(t)$ the neighborhood contribution, M and m superior and lower activation bounds, α_{ij} and β_{ij} the positive and negative stimulation from j to i, and $a_j(t)$ the activation of node j

Recognition is performed trough several bottom-up and top-down processes. The physical features extracted from the image are specific to the problem: primary (e.g. ascender, descender) secondary (e.g. loop, bar) and face-up/face-down valley (e.g. connected components of the background between the lower and upper contours of the word). The architecture of the system is general enough to handle hierarchical organized interpretation. The authors have chosen three levels of neurons: feature, letter, and word (Fig. 5).

The connections between adjacent levels are excitatory and bi-directional. The connections are only bottom-up between the feature letter and the letter level. The weights are determined according to a priori knowledge. Thanks to an active and passive neuron system, the network can reach the solution after several cycles (until saturation) of bottom-up and top-down processes called perceptual cycles. The system generates hypothesis, validates them and may insert letter candidates in the right place using already validated letters (Fig. 6).

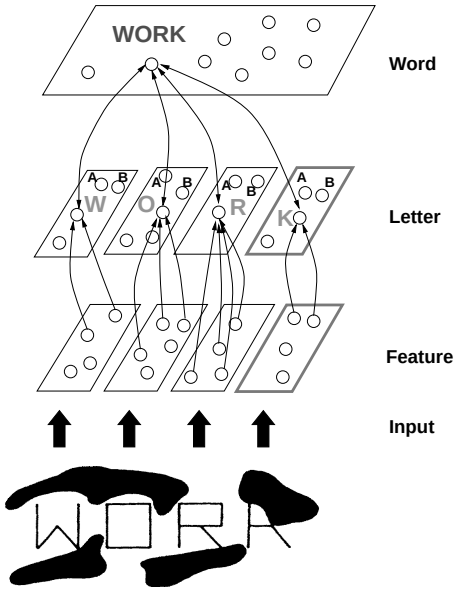

Fig. 5. Côté et al. hierarchically organized ANN model

Experiments have been made on CENPARMI database (French and English handwritten cheques), using 184 pattern for training and 2929 for testing achieve form 85.3% for word length 3 up to 100% with word length 9.

In [25], Maddouri et al. propose an extension of the Perceptro model [23]. They use a geometrical correction method to improve the performances of the Arabic handwritten word recognition system developed. The recognition proceeds in cycles of global and local observations. The global observations try to detect apparent features of the words. They create hypotheses on the word label. To carry out the recognition from different kinds of information, a normalization stage is done on the word edges to improve the local observations. Indeed, contrary to printed words or characters, the handwritten text needs a powerful normalization stage to handle the variability in position, size, rotation, slant, and distortion. The authors have chosen a Fourier based solution to eliminate this variability. The whole recognition process is summarized in Figure 7.

The top-down and bottom-up cycles are carried out thanks to the TNN model (right part of the schema), the local observation comes from the normalization of features such as ascenders, descenders, diacritics, and loops (left part of Fig. 7).

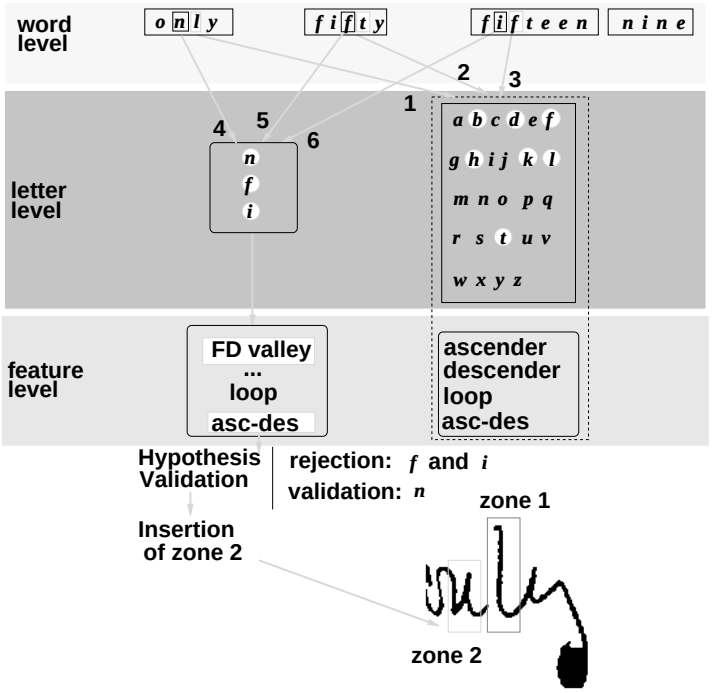

Fig. 6. Top-down process: feedback and insertion

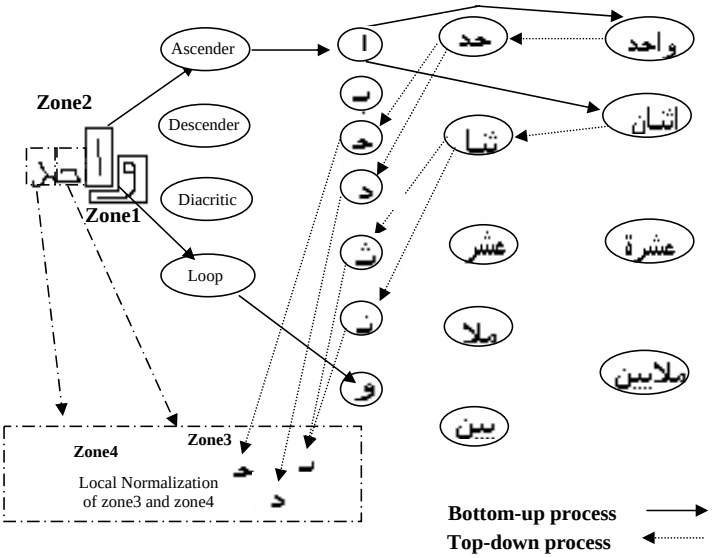

Fig. 7. Local normalization and global recognition

The normalization is performed on the boundary of the word: a detection of the contour is done first, then a Freeman chain code is generated, the next step consists in computation of Fourier coefficient of the chain-encoded contour and finally, the coefficients are normalized to cope with variability. To obtain the final normalized character, a reverse Fourier transformation is applied to the latest normalized coefficient. The reader can refer to [25] to see how the boundary normalization is carried out. When the word is normalized, a metric distance is used to evaluate the difference between the current word and printed references.

3.5 Perceptive structured neural network for logical structure analysis

In [12], Rangoni et al. propose a quite similar TNN for logical structure recognition in document images. The hierarchically organized interpretation is kept and transposed to handle editorial documents. Each neuron corresponds to an interpretable concept and is attached to an element of the logical structure. Excluding the first layer composed of input physical features, the following layers unfold the interpretation by introducing fine concepts in the first layers and general concepts in the latest layers (Fig. 8).

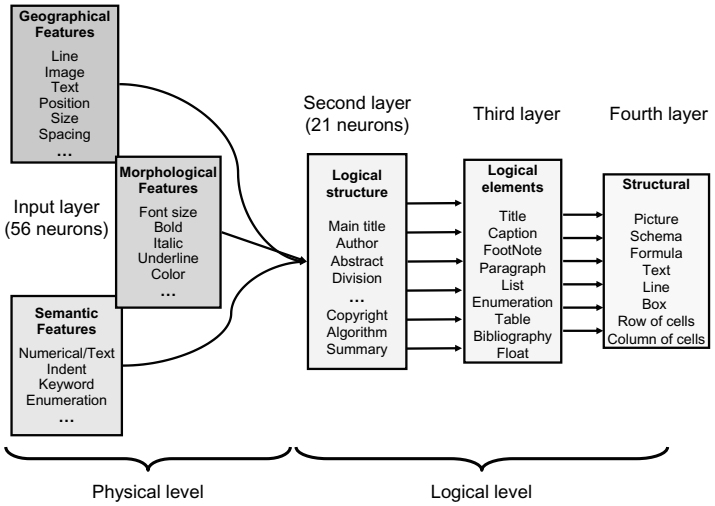

Fig. 8. Topology for scientific articles

If a DTD is present, it can be helpful to set the neurons meaning: the hierarchy included in the DTD can be unfolded to form the layers and the neurons. Contrary to other models [23, 25] the network is fully connected and the neurons can be inhibitors.

Training and recognition

As the relations between the layers are not straightforward, a training phase similar to MLP is proceeded to set all the weights.

In the back propagation algorithm, the error $E_p(w)$ between the desired output d_q and the computed output $o_{L,q}$ is minimized for each pattern p

$$\begin{aligned} E_p(w) &= \frac{1}{2}\sum_{q=1}^{N_L}(o_{L,q}(x_p) - d_q(x_p))^2 \\ o_{l,j} &= f\left(\sum_{i=0}^{N_{l-1}} w_{l,j,i}o_{l-1,i}\right) \end{aligned} \tag{11}$$

As a consequence, the weight between the unit i in layer l and unit j in layer $l+1$ is modified as follows

$$w_{l,i,j} \rightarrow w_{l,i,j} - \mu\sum_{p=1}^{P}\frac{\partial E_p(w)}{\partial o_{l,j}} f'\left(\sum_{m=0}^{N_{l-1}} w_{l,j,m}o_{l-1,m}\right) o_{l-1,i} \tag{12}$$

In case of [12], all the neurons carry interpretable concepts and the desired output is known for all the units. So, the partial term is given by:

$$\forall l, \frac{\partial E_p(w)}{\partial o_{l,j}} = o_{l,j}(x_p) - d_j(x_p) \tag{13}$$

and the network can be trained as a cascade of mono-layer perceptrons.

The model is on the one hand data-driven thanks to the training stage and on the other hand model-driven due to the integration of knowledge inside the topology. This kind of ANN is called Transparent Neural Network (TNN) in contrast to the "blackbox" aspect of MLP. For document logical layout analysis, we have named this the Perceptive Structured NN (PSNN).

The aim of the final layers is to bring context during the perceptive cycles as the previous authors used these to simulate the "word superiority effect" on letters. As the network is feed forward, the learning of the network is the same as an MLP but here the training is done separately between each consecutive pair of layers because all the desired outputs are known.

During the recognition step, the network is used as an MLP but after each propagation, the outputs are analyzed. If the output vector is close to a basis vector (14 & 15) the pattern is considered classified, otherwise the following layers are taking into account to bring context. $M(O)$ gives a vector with at least one component with high value, $\Gamma(O)$ give a vector where one component has a value very high compared to other components.

$$M(O) = \|O\|_\infty > \varepsilon \qquad \text{with} \qquad 0 \ll \varepsilon < 1 \tag{14}$$

$$\Gamma(O) = \frac{n((\sum O_i)^2 - \sum O_i^2)}{(n-1)(\sum O_i)^2} < \eta \quad \text{with} \quad 0 < \eta \ll 1 \tag{15}$$

As these layers contain more global information, they are more robust and accurate. They are used to generate hypothesis on the pattern. The context manages the correction of the input features. Once a label is supposed to be the good one, the input vector is modified according to this hypothesis and according to the knowledge extracted from the training database. Indeed, several representative samples are extracted from database and are matched with the current input. The input is modified to be close to a representative sample and another perceptive cycle is completed and so on until no ambiguities persist (Fig. 9).

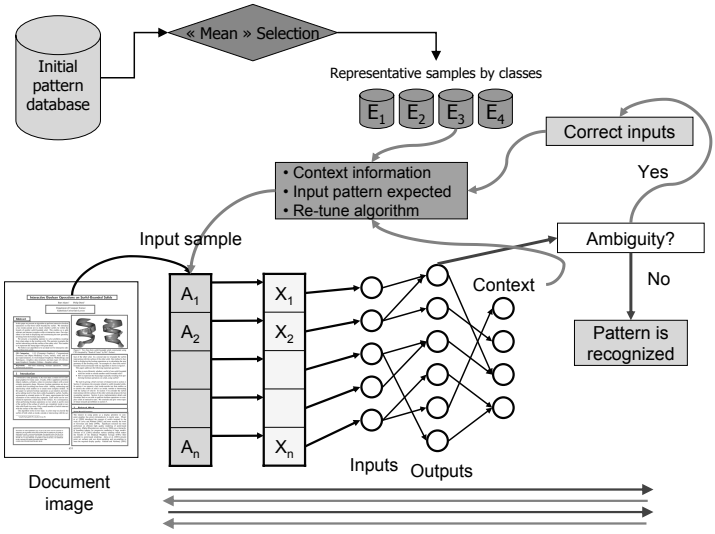

Fig. 9. Recognition in Perceptive Structured Neural Network

There are several methods to determine these representative samples; unfortunately there is no exact solution. Some approaches have been investigated. Methods using optimization produce mathematically perfect sample but they do not correspond to real-world interpretable solutions. Methods that are more straightforward can produce appropriate samples: mean sample for only one representative or a k-NN for select several samples per class. Others methods can be performed during the training stage: In [26], a new learning method is presented which can produce from a very small subset of the global database an MLP almost as efficient as trained on the whole database. The subset arisen from the algorithm will provide the representative samples.

Input feature clusterization

The perceptive cycles in this PSNN allow bottom-up and top-down resolution and refine the recognition. However, if too many recognition cycles have to be done, the task could be very time consuming because a lot of physical extraction must be completed. On top of that, some of the inputs are high-level (given by OCR) and slow down the logical structure recognition. In order to face this problem, a manual selection has been used to trim down the extractions. To simulate global and local vision, the input features are partitioned into clusters using a data categorization. Instead of feeding the network with the whole features for each cycle, the features are given progressively during the recognition and only if the pattern is too ambiguous (Fig. 10).

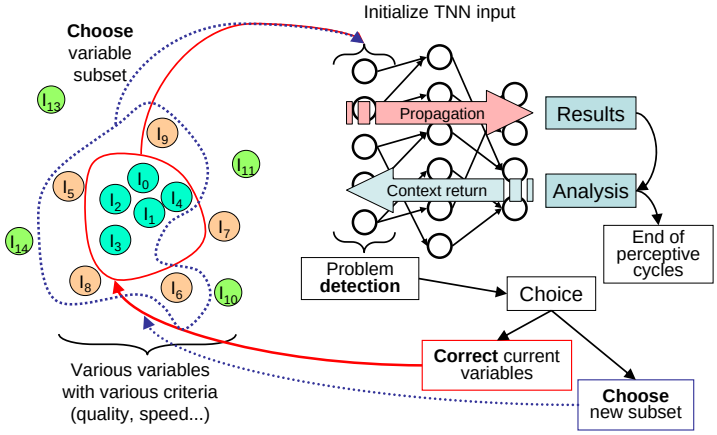

Fig. 10. Perceptive cycles: propagation, analyze, context return, correction, input feature selection

Subsets of feature are computed according to their extraction time and their predictive capacities. The first criterion is trivial as the extraction can be timed by experiments or by analyzing the algorithm complexity. Evaluating the predictive power of a set of features is more complicated as there is no optimal solution to do this. The literature proposes two main approaches: filter-based methods and wrapper methods [27]. The filter methods only use the sample database to score the feature, they are fast to evaluate the feature separately but do not produce good groups. On the other side, wrapper methods consider variables but they need the classifier to produce the groups. The method presented in [12] is based on a filter approach but can compute groups at the same time with ordered predictive power and less redundancy inside each group (Fig. 11).

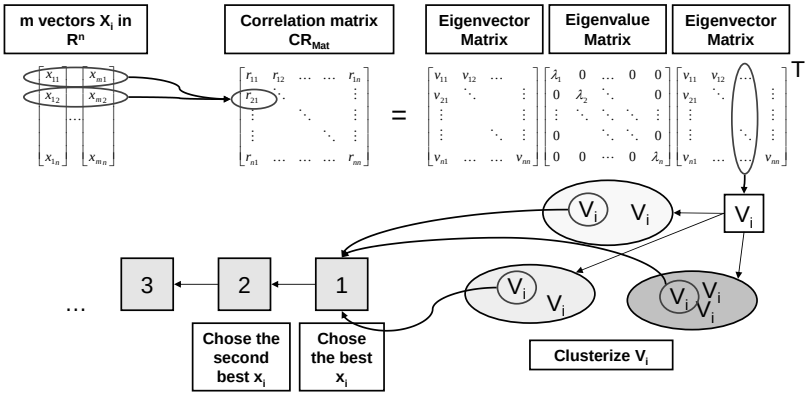

Fig. 11. Input feature clustering

By combining the input feature correction and selection, the PSNN is able to adapt the computation amount according to pattern complexity without adding too much processing time.

Experimentations

The system has been tested on scientific articles (Fig. 13). Physical inputs are mainly extracted by commercial OCR, others are computed by using existing ones. There are also 21 logical lables that cover any document image (Fig. 12).

After four perceptive cycles, the recognition rate increase to 91.7% which is 10 points better than a classical MLP (Tab. 1).

Logical	Physical		
	Geometrical	Morphological	Semantic
Title	Text	Bold	IsNumeric
Author	Image	Italic	KeyWords
Email	Table	Underlined	%KnownWords
Locality	Other	Strikethrough	%Punctuation
Abstract	x position	UpperCase	Bullet
Key words	y position	Small Capitals	Enum
CR Categories	Width	Subscript	Language
Introduction	Height	Superscript	Baseline
Paragraph	NumPage	Font Name	
Section	UpSpace	Font Size	
SubSection	BottomSpace	Scaling	
SubSubSection	LeftSpace	Spacing	
List	RightSpace	Alignment	
Enumeration		LeftIndent	
Float		RightIndent	
Conclusion		FirstIndent	
Bibliography		NumLines	
Algorithms		Boxed	
Copyright		Red/Green/Blue	
Acknowledgments			
Page number			

Fig. 12. Input feature clustering

Classes		PSNN			
	MLP	C_1	C_2	C_3	C_4
Whole	81.6%	45.2	78.9	90.2	91.7%
Best	86.9%	66.7	85.3	85.3	99.3%
Worst	0.0%	0.0	0.0	4.0	28.6%
Time	1	0.7	1.45	1.85	2.40

Table 1. Logical structure classification for MLP and for PSNN

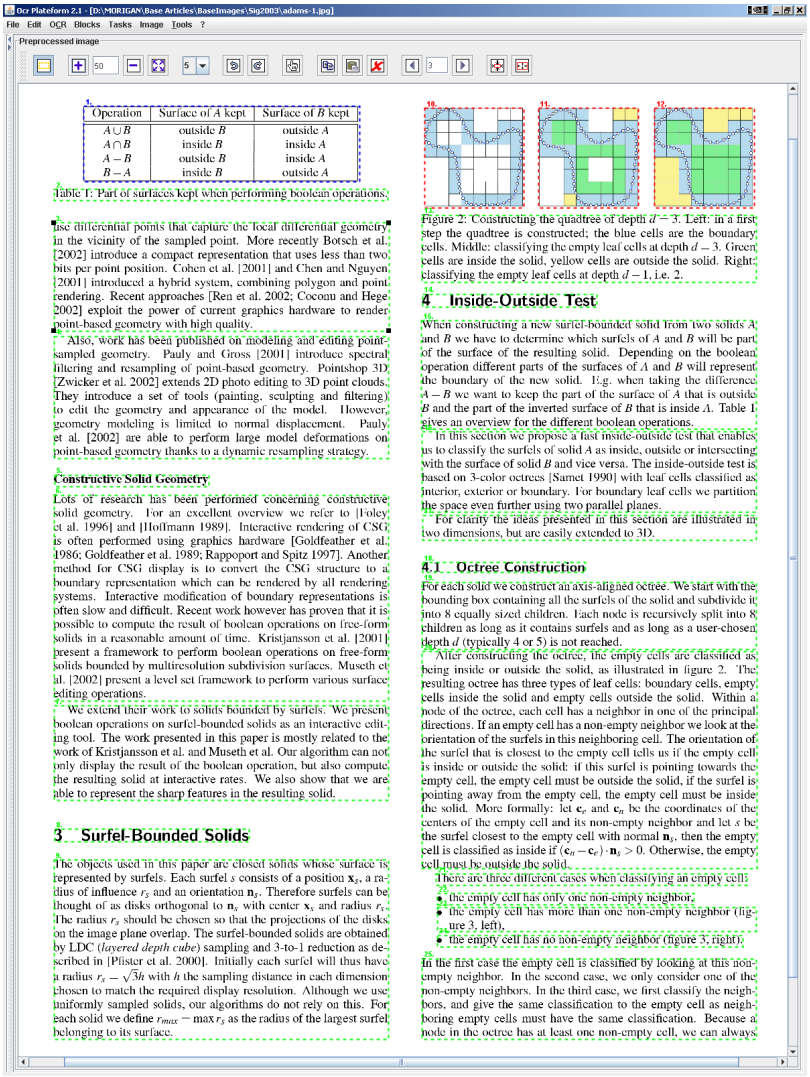

Fig. 13. Document sample

4 Conclusion and perspectives

We outlined in this paper several categories of approaches used for the extraction of document structures from raster images. After a description of their properties, advantages and drawbacks, we focused on neural approaches for their capacities in noise absorption and generalization capacities. Their application in document analysis was a real challenge for us as they were not considered for this kind of structured data. The idea of McClelland to propose a perceptive model with different cycles allowing a dynamic and progressive approximation of the problem was the basis of our investigations. After the study of others dynamic models, we proposed a specific one called Perceptive Structured Neural Network which can be applied for logical structure recognition. This model allows us to process several categories of structures based only on physical data. After few cycles, the behavior of the system is better than an MLP's. Besides, it gives us the possibility to refine the outputs by correcting the inputs accordingly.

Although dynamic ANN are able to deal with structured patterns, they are not still used for document logical layout analysis. Besides, static networks have been used far less than pure model-driven approaches. All the works presented in the section 3 show how to extend classical models to deal with such a problem. The neuronal approach is accessible and can be as competitive as grammar or rule based systems. It is obvious that, as mentioned in Nagy et al. [28], domain specific knowledge appears essential for document interpretation.

The proposed PSNN can be improved in a different way: the data-driven methods may be improved by introducing hidden layers between each layer of interpretable concepts. The "transparency" property will be lost but the system will be more accurate and have better generalization capacities.

Another approach could integrate transparency in a dynamic network or adding dynamic properties to PSNN. A simply output feedback-based PSNN will have more feedback information when using the context. On top of that, the context will be taking into account not only during the recognition but also during the training stage.

References

1. Marinai, S., Gori, M., Soda, G.: Artificial neural networks for document analysis and recognition. Pattern Analysis And Machine Intelligence **27**(1) (2005) 23–35
2. Chi, Z., Wong, K.: A two-stage binarization approach for document images. International Symposium on Intelligent Multimedia, Video and Speech Processing (2001) 275–278
3. Hamza, H., Smigiel, E., Belaïd, A.: Neural based binarization techniques. International Conference on Document Analysis and Recognition **4**(8) (2005) 317–321
4. Whichello, A.P., Yan, H.: Linking broken character borders with variable sized mask to improve recognition. Pattern Recognition **29**(8) (1995) 1429–1435
5. Le Cun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
6. Cecotti, H., Belaïd, A.: Rejection strategy for convolutional neural network by adaptive topology applied to handwritten digits recognition. International Conference on Document Analysis and Recognition (8) (2005) 765–769
7. Garris, M.D., Wilson, C.L., Blue, J.L.: Neural network-based systems for handprint OCR applications. IEEE Transactions on Image Processing **7**(8) (1998) 1097–1112
8. Mao, S., Rosenfeld, A., Kanungo, T.: Document structure analysis algorithms: A literature survey. SPIE Electronic Imaging **50**(10) (2003) 197–207
9. Brugger, R., Zramdini, A., Ingold, R.: Modeling documents for structure recognition using generalized n-grams. International Conference on Document Analysis and Recognition **1**(4) (1997) 56–60
10. Hu, T., Ingold, R.: A mixed approach toward an efficient logical structure recognition from document images. Electronic Publishing: Origination, Dissemination, and Design **6**(4) (1993) 457–468
11. Niyogi, D., Srihari, S.N.: Knowledge-based derivation of document logical structure. Third International Conference on Document Analysis and Recognition **1** (1995) 472–475
12. Rangoni, Y., Belaïd, A.: Document logical structure analysis based on perceptive cycles. Document Analysis Systems **1**(7) (2006) 117–128
13. Küchler, A., Goller, C.: Inductive learning in symbolic domains using structure-driven recurrent neural networks. Lecture Notes in Computer Science (1137) (1996) 183–197
14. Sperduti, A., Starita, A.: Supervised neural networks for the classification of structures. IEEE Transactions on Neural Networks **8**(3) (1997) 714–735
15. Hertz, J., Krogh, A., Palmer, R.G.: Introduction to the theory of neural computation. Addison Wesley (1991)
16. Moody, J., Darken, C.J.: Fast learning in networks of locally-tuned processing units. Neural Computation (1) (1989) 281–294
17. Narendra, K.S., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks **1**(1) (1990) 4–27
18. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences of the United States of America **79**(8) (1982) 2554–2558
19. Pineda, F.J.: Dynamics and architecture for neural computation. Journal of Complexity **4**(3) (1988) 216–245

20. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural Computation **1**(2) (1989) 270–280
21. Sperduti, A., Starita, A., Goller, C.: Learning distributed representations for the classification of terms. Proceedings of International Joint Conference on Artificial Intelligence **1**(40) (1995) 509–515
22. Fahlman, S.E., Lebiere, C.: The cascade-correlation learning architecture. Advances in Neural Information Processing Systems **2** (1990) 524–532
23. Côté, M., Lecolinet, E., Cheriet, M., Suen, C.: Automatic reading of cursive scripts using a reading model and perceptual concepts. the Perceptro system. International Journal on Document Analysis and Recognition **1**(1) (1998) 3–17
24. McClelland, J., Rumelhart, D.: An interactive activation model of context effects in letter perception. Psychological Review (88) (1981) 375–407
25. Maddouri, S.S., Amiri, H., Belaïd, A.: Local normalization towards global recognition of Arabic handwritten script. Document Analysis and Systems (2000)
26. Vajda, S., Rangoni, Y., Cecotti, H., Belaïd, A.: A fast learning strategy using data selection for feedforward neural networks. International Workshop on Frontiers in Handwriting Recognition (10) (2006)
27. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research (3) (2003) 1157–1182
28. Nagy, G.: Twenty years of document image analysis in PAMI. Pattern Analysis and Machine Intelligenc **22**(1) (2000) 38–62