Modified Self-Organizing Maps for Line Extraction in Digitized Text Documents

J.M. Alonso-Weber, I.M. Galván and A. Sanchis

Departamento de Informática, Universidad Carlos III de Madrid,
Avenida de la Universidad 30, 28911, Leganés, Madrid.
jmaw@ia.uc3m.es, {igalvan, masm}@inf.uc3m.es

Abstract.. Different authors have developed modifications of the Kohonen Self-Organizing Maps to solve known combinatorial optimization problems. In this paper a modification of the Kohonen Map is proposed to solve the detection of white inter-text spaces in a digitized plain text documents. The idea relies on the fact that line extraction problem has several features which match easily with Kohonen networks, although an adaptation to the problem of the original learning rule has to be made at first. A test with different digitized text images is performed showing the ability to segment lines.

1 Introduction

The Self-Organizing Map (SOM) of Kohonen [1][2] consists of a group of nodes (neurons) placed in some type of topology, such as a lineal ribbon or a two- or multidimensional matrix. In this topology the concept of vicinity is defined in such a way that two contiguous nodes are considered as immediate neighbours, while those nodes which are not contiguous present a smaller degree of neighbourhood. This relationship is measured with some metric, usually the Euclidean distance. This deliberately chosen structure can be considered as a discreet and finite space in which each one of the nodes represents one of the possible space values.

Each one of the nodes receives a series of stimuli, simultaneous in time, that come from a finite group of neurons (called input neurons). The stimuli that transfer the input neurons to each of the output neurons are always numeric and grouped in a set (input pattern). Therefore, we will also have an input space with so many dimensions like input neurons, in which the patterns will be distributed in an arbitrary way.

For each of these patterns the SOM will activate the most “similar” neuron in the input space. The measure of similarity is carried out with a function of distance (Euclidean or any other metric distance). Afterwards, the active (or winning) neuron (i.e. the nearest one to the pattern) will modify its parameters (co-ordinates) moving towards the input pattern. Additionally, all the nodes are moved towards the input pattern in some amount that diminishes with a decreasing neibourhood relation to the winning node, and with an increasing simulation time. This variable dragging dynamics allows the SOM to spread inside the input space adapting to the forms of the distribution of patterns: at first (when the dragging is bigger) adopting a vague and global approach, and later on, tuning to the contours of the distribution of patterns in the input space.

Some properties present in these SOMs are:

1. Locations in the input space with higher pattern densities are assigned a higher number of map nodes, while locations with few or no patterns at all do not allocate any nodes.

2. The projection performed from the input into the output space preserves the neibourhood properties of the distribution in the input space. This means that patterns which are close to each other in the input space, compete for the same node or for nodes which are neighbours in the map topology.

Many applications have been studied for very different types of problems. Most of them are related with the topics of classification, vector quantification, dimensionality reduction, or even information retrieval.

A particular modification of these SOMs was carried out to approach the well-known Travelling Salesman Problem [3]. It relied on the idea of simulating the behaviour of an elastic net (previously described in [4]) which is shaped into a minimal distance route by the attraction of the individual cities distributed in a space plane. The resulting application is simple, elegant and with interesting results, but also presents restrictions that limit its use exclusively to the Euclidean and symmetrical variant of the TSP.

The purpose of this work is to modify some parameters and the dynamic behaviour of the traditional SOM in order to achieve a more general framework for solving some Combinatorial Optimization Problems. A first small application of this modified SOM is tested on a line extraction problem in digitized plain text documents.

Interesting types of problems are found in the context of digital image matching. Inside this problem, the 2D shape matching can be approached following the Kohonen SOMs philosophy. In the literature there are many different shape description methods [5, 6, 7], none of the methods were found to work well on different kind of shapes, and sometimes the methods are very domain specific, hardly to extend to other shapes. Another interesting problem is the localization of characters in a document image. That implies several operations such as determining the skew [8], separating picture from the text [9] and portioning the text into columns, lines and words, which is accomplished through a segmentation process [10].

The interest of this paper is to approach the line detection problem in a document image, this is to identify not-printed zones in the image. The problem is solved using the Kohonen network, but an adaptation of the original learning rule to the problem is needed.

In section 2, the adaptation of the model’s general dynamics is described. Section 3 is deals with the experiments done to validate the proposal and section 4 contains some conclusions and future work.

2 Self-Organizing Maps for Solving Line Extraction Problems in Digitized Texts.

The idea of applying the SOMs on the Line Extraction Problem relies on the fact that this problem has several features which match easily with the SOFM. Plain text consists of a number of text lines which are interspersed with white space lines. Usually, this white space lines will be rather straight, and the separation between text lines will be rather periodic in size. Furthermore, text lines will be parallel and grouped in a more or less rectangular appearance. Although this seems to be a fairly structured problem, in practice it might be difficult to determine how the single text lines are arranged. Several problems and types of distortion can appear when scanning a document: a skew, mechanical distortions which convert square texts arrangements into trapezoidal ones, curved lines and uneven contrast, brightness or colour casts can confuse text from white discrimination.

A slight skew (of even 1º) in the scanned document might force to use Hough Transforms or other time-consuming techniques to recognize the correct white inter-text separation spaces.

Algorithm description

The underlying idea for using the SOM for this problem is the idea that white inter-text spaces consist of several white regions which traverse the document in a continuous, straight and parallel way. As noted before, Kohonen Maps tend to dedicate more nodes to regions in input space with a higher pattern density. Moreover, neighbouring patterns in input space tend to compete for neighbouring nodes in the SOM. The idea is to try to project the white inter-text lines onto node lines defined on a two-dimensional SOM. There is a need to modify the SOM dynamics, as we also intend to detect the straight and parallel arrangement of text lines. The main basic ideas of this modifications are:

1. "Horizontal" node lines of the SOM should evolve into rigid lines through time. (Horizontal is deliberately quoted because it refers to the node lines that match horizontal white inter-text lines, which in a skewed digitized document could be a relative concept).
2. Starting from a low quantity of node lines, new lines should be inserted where needed, i.e. when the map detects unmatched white inter-text spaces, or when a node line is placed crossing a printed text line.

As it seems, horizontal node lines have an important purpose. Vertical node lines might not be expected to have such a relevance, but also have an influence in the segmentation quality, as it will be shown in the experimental section.

Also expected is the ability to detect skew in the scanned text image. Uneven contrast, brightness and colour cast problems will be partially eliminated with a special pattern generation procedure. This procedure will be described at first. Afterwards, the learning algorithm of the SOM is explained, concluding with the changes induced in the dynamics.

Pattern generation

The task of this procedure is to convert the white inter-text pixels into patterns that will be used for the SOM learning procedure. Being the scanned text image a digitized image composed by *NxM* pixels, we will consider d_{ij} ($0 \leq i < N, 0 \leq j < M$) as the value for each pixel located at line *i* and column *j*. As usual, those pixel values range from 0 to 255, where 0 represents a white dot and 255 the black one. Since the segmentation process has to make out white zones from the printed ones in the image, an appropriate selection of pixels d_{ij} must be carried out. The final data should be pairs $P_{ij} = (i, j)$ which correspond to pixels with no ink. The procedure to select the most relevant pixels is defined as follows:

A threshold *S* (with $0 < S < 255$) is defined (usually about 128). Each pixel value d_{ij} is taken as a real white dot when

$$d_{ij} \leq S \qquad (1)$$

Applying this only criterion generates a huge amount of patterns. Useless information might be also included because pixels inside the hole part of letters could be selected. Additionally, in some written documents, such as photocopies or newspapers, there will be a lot of noisy data due to intermediate gray tones which can not easily identified as originally being white or ink dots.

In order to avoid the previously described situations, additional criterions are used to select pixels. Only those pixels whose value are a local minimum inside a square around them, are considered. The size of the square neighbourhood is given by *nxn*. The pixel at *(i, j)* should meet:

$$d_{ij} < d_{kl}\,, \quad n > 0, \;\; \forall\, k, j \;\; with\; |i - k| \leq n, \; k \neq i, \quad |\,j - l| \leq n, \;\; l \neq j \qquad (2)$$

The maximum order of the minimum is defined as $n(d_{ij})$, the maximal value of *n* at which the pixel is still an absolute minimum inside the neighbourhood.

In order to avoid local minima due to scanner sampling errors that belong to big white surfaces, another restriction is added. For each local minimum d_{ij}

$$\exists\, d_{kl} \quad d_{kl} > S, \, with\; |i - k| = n, \quad |\,j - l| = n \qquad (3)$$

This limits the pattern selection to white pixels located close to written text. Two limits, O_{inf} and O_{sup} are used to reduce the amount of the pixels.

$$O_{inf} < n(d_{ij}) < O_{sup} \qquad (4)$$

For those pixels that meet (1), (2), (3) and (4) a pattern set $P_{ij} = (i, j)$ is generated for training the SOM.

Learning algorithm

As stated, a two-dimensional Kohonen map will be used, in which the output nodes, denoted as C_{pq}, are distributed onto a grid of dimension *v x h*. The index *p* ($0 \leq p < v$) and *q* ($0 \leq q < h$) determine the position of nodes in the grid (row *p*, column *q*). There is a metric *dt()* defined on the grid that measures the neighbourhood relation between output nodes. As usual, this metric will use the Euclidean distance. If C_{pq} and C_{rs} are output nodes, the distance over the map between the nodes is given by eq. (5):

$$dt(C_{pq} , C_{rs}) = ((p - r)^2 + (q - s)^2)^{1/2} \qquad (5)$$

Each node C_{pq} in the map has associated a real weight vector, whose dimension is given by the pattern dimension. In our case, two weights or parameters are used, denoted as X_{Cpq} and $Y_{Cpq,}$, that represent the position of the nodes over the image (input or pattern space). Since the image has *NxM* pixels, the weights must verify:

$$0 \leq Y_{Cpq} < N, \quad 0 \leq X_{Cpq} < M, \;\; 0 \leq p < v, \;\; 0 \leq q < h \qquad (6)$$

1. The weights X_{Cpq}, Y_{Cpq} are randomly generated at the first time. When patterns P_{ij} are presented to the network, the weights of the output nodes are iteratively updated using the following learning procedure:

2. For each pattern $P_{ij}=(i,j)$ the winning unit, called C^*_{pq}, in the map is calculated. The winning unit will be the output node with the weight vector closest to the current pattern P_{ij}:

$$Dist(C^*_{pq}, P_{ij}) \; < Dist(C_{rs}, P_{ij}), \quad \forall\, r \neq p, s \neq q \qquad (7)$$

where *Dist()* is the Euclidean distance (in the input space). Taking into account that $P_{ij}=(i,j)$, the distance is given by eq. (8):

$$Dist(C_{rs}, P_{ij}) = ((Y_{Crs} – i)^2 + (X_{Crs} – j)^2)^{½} \qquad (8)$$

3. Once the winning node is identified, all weights in the network are adapted using eq. (9) and (10):

$$Y'_{Crs} = Y_{Crs} + fgn\,(g, C^*_{pq} , C_{rs}\,) * (i - Y_{Crs}) \qquad (9)$$
$$X'_{Crs} = X_{Crs} + fgn\,(g, C^*_{pq} , C_{rs}\,) * (j - X_{Crs}) \qquad (10)$$

The learning rule moves all nodes in the map into the direction of the pattern $P_{ij}=(i,j)$ and the amount of this movement is given by the neighbourhood function *fgn ()* :

$$fgn\,(g, C^*_{pq} , C_{rs}\,) = g * e^{–k} \qquad (11)$$

$$k = dt(C^*_{pq} , C_{rs})^2 / g^2 \qquad (12)$$

where *dt()* is the distance over the map defined in eq. (5); and *g* is the gain parameter which decreases between two complete iterations. Several iterations are needed to go from a high gain value to the low one. The time dependence of the gain parameter is described in eq. (13):

$$g= g(t) = g_0*(1 - \alpha)^t \tag{13}$$

where t , the time variable, is increased by one unit once a complete iteration is performed; g_0 and α are the only fixed parameters. g_0 gives the starting dragging force of the nodes, and α determines the number of iterations used to complete the development of the SOM.

The neighbourhood function defined in eq. (11) is used to move the weights in such a way that units close to the winner, as well as the winner unit, have their weights changed appreciably. The winner unit will undergo the maximum movement, while the rest of the nodes in the map are modified depending of the their closeness to the winner unit. The value of neighbourhood function for the winner $C*_{pq}$ node is g, from eq. (11):

$$fgn\ (g, C^*_{pq} , C^*_{rs}) = = g * e^{-0} = g \tag{14}$$

because $k = dt(C^*_{pq} , C^*_{rs})^2 / g^2 = 0$

For any other node in the map, $k > 0$. Hence the value follows:

$$fgn\ (g, C^*_{pq} , C_{rs}) < fgn\ (g, C^*_{pq} , C^*_{pq}) \tag{15}$$

4. Once the steps 2 and 3 are repeated for all patterns an iteration concludes. The t parameter is incremented and the new gain parameter is computed. So far the usual learning procedure for SOMs.

5. At this point some additions are introduced to modify the SOM’s behaviour and to apdapt it to the line extraction problem (described in the next section).

6. The procedure is stopped when a lower bound value of the gain parameter is reached. At very low gain rates, no visible modifications are performed in the SOM, which means that further iterations are of no use.

Modified Dynamics

As stated, after each iteration some new calculations are performed.
For each node, a new position in the input space is computed. At first:

$$C_{pq} = (C_{(v-1)q} - C_{1q})*p + C_{1q} \tag{16}$$

and afterwards:

$$C_{pq} = (C_{p(h-1)} – C_{p1})*q + C_{p1} \tag{17}$$

The effect of eq. (15) is that the horizontal node lines evolve into straight lines during the SOM expansion. Eq. (16) is needed to equilibrate the vertical movements induced by eq. (15). In addition, it allows the SOM to acquire an approximate square appearance which is desirable as we expect text blocks to have a rectangular layout.

The insertion of new line nodes takes place in case following the next considerations. For each C_{pq} we will have a set of patterns $\mathrm{W}_{Cpq}= \{P_{kl}\}$ for which C_{pq} is the winning node and that contains $S_{Cpq} =|\{\{P_{kl}\}|$ patterns.

$$D_{Cpq}= (\sum_{Pkl} dist\ (C_{pq}\ , P_{kl})) / S_{Cpq} \tag{18}$$

$$L_p = \sum_q D_{Cpq} \tag{19}$$

Eq. (18) is useful for identifying those nodes that cover big extensions of patterns. After each iteration, node lines with the maximal value for eq. (19) are duplicated, i.e. a new line node is inserted in the map, straight before or after. This allows the map to allocate new node lines where more than one inter-text lines are covered with a single node-line.

3 Experimental Results

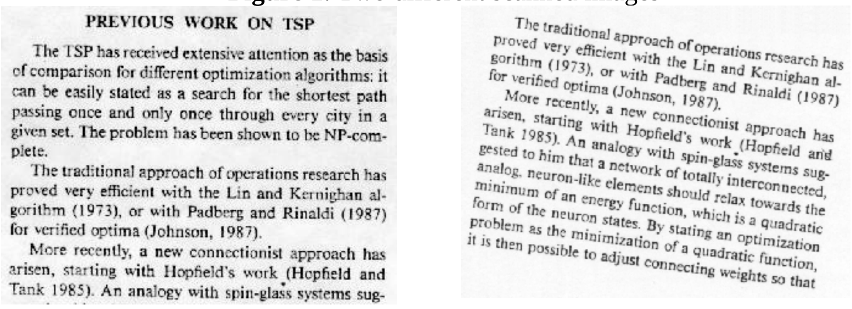
The proposed approach is tested with the problem of segmentation in digitized text images (as books or newspapers pages), finding their lines, and after separating each one of the characters. This problem is the previous step in the character recognition in written texts and allows showing the main characteristics of the SOMs, as the invariant with rotations in the text, the usefulness of transforming flexibility into rigidity and the dynamic generation of nodes and sub-structures in the map.

Experimental framework

The images used for the following experiments were scanned using a flatbed scanner at 1200dpi as colour documents. No contrast, brightness nor any other enhancement was performed. From the obtained images some interesting portions were selected, cut out and resized (about a 60%) to 320x240 pixels. Then colour information was converted into a grey-scale using the average between Red, Green and Blue intensities.

Two different cases were selected (see Figure 1). From one photocopied article (10 years old, with an intense yellow cast and a skew of about 0.75º) two different portions were selected. The same procedure was repeated on a photocopy of the same article (which eliminated the skew, the yellow cast, increased contrast, and a loss of some details). This second photocopy was also scanned with a skew of about 7º. Additionally, one hand-written text was scanned.

Figure 1: Two different scanned images



PREVIOUS WORK ON TSP

The TSP has received extensive attention as the basis of comparison for different optimization algorithms: it can be easily stated as a search for the shortest path passing once and only once through every city in a given set. The problem has been shown to be NP-complete.

The traditional approach of operations research has proved very efficient with the Lin and Kernighan algorithm (1973), or with Padberg and Rinaldi (1987) for verified optima (Johnson, 1987).

More recently, a new connectionist approach has arisen, starting with Hopfield's work (Hopfield and Tank 1985). An analogy with spin-glass systems sug-

Original photocopied article. Portion 1. 0.75º skew

The traditional approach of operations research has proved very efficient with the Lin and Kernighan algorithm (1973), or with Padberg and Rinaldi (1987) for verified optima (Johnson, 1987).

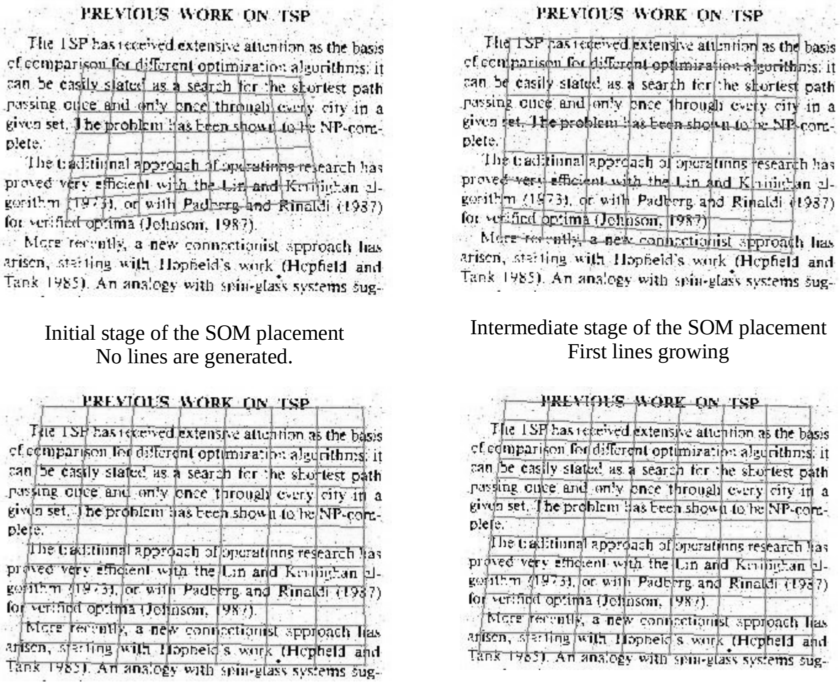
More recently, a new connectionist approach has arisen, starting with Hopfield's work (Hopfield and Tank 1985). An analogy with spin-glass systems suggested to him that a network of totally interconnected, analog, neuron-like elements should relax towards the minimum of an energy function, which is a quadratic form of the neuron states. By stating an optimization problem as the minimization of a quadratic function, it is then possible to adjust connecting weights so that

Secondary photocopy. Portion 2. 7º Skew.

Experiment Evaluation

The previously mentioned digitized text portions were presented to the modified SOM simulator. For each text, different simulations were performed starting with different initial sized maps.

Figure 2: A simulation sample



PREVIOUS WORK ON TSP

The TSP has received extensive attention as the basis of comparison for different optimization algorithms: it can be easily stated as a search for the shortest path passing once and only once through every city in a given set. The problem has been shown to be NP-complete.

The traditional approach of operations research has proved very efficient with the Lin and Kernighan algorithm (1973), or with Padberg and Rinaldi (1987) for verified optima (Johnson, 1987).

More recently, a new connectionist approach has arisen, starting with Hopfield's work (Hopfield and Tank 1985). An analogy with spin-glass systems sug-

Initial stage of the SOM placement
No lines are generated.

Intermediate stage of the SOM placement
First lines growing

Final stage of the SOM.
No more node lines are generated.

Final placement of the SOM.

The resultant segmentation was then evaluated. Several situations and error types were taken into account:

A. More than one node line allocated for one single white inter-text line.
B. A node line crosses a white inter-text line diagonally.
C. A node line crosses a text line.
D. A missing node line in a white inter-text line.
E. Excessive or insufficient map skew to match the documents skew.

Experimental Results

For each simulation carried out the type and rate of failed matches is indicated.

The simulation parameters were $g_0 = 10.0$, $\alpha = 0.1$ and $g_{stop} = 0.01$

Number of initial nodes (n)	**5**	**6**	**8**	**10**
Processed Images	**Error types and (quantities)**			
Original photocopy. Portion 1	C(1)	C(1)	C(1)	none
Original photocopy. Portion 2	C(3) E	C(4) E	C(2) E	C(1) E
Secondary photocopy. Portion 1	C(1) B(1)	C(2) B(1)	C(1)	none
Secondary photocopy. Portion 2	C(1) E	none	none	none
Skewed photocopy (7º skew). Portion 1	C(many) E	C (many)	C(many) E	C(many) E
Original photocopy (7º skew). Portion 2	none	none	none	C(3) E
Hand-written text 1.	none	none	none	*

* The hand-written text has only 8 text lines.

The first conclusion when studying the results is that the error rate is lower when using a higher number of nodes per line in the SOM. The only cases that doesn't follow this statement are the skewed text portions. A first inspection of this case determines that the number of generated patterns are in both cases lower than 1300. For a final SOM of 14x10 nodes, this means that the average pattern number per node is about 10, which might be somewhat low to allow correct development of the map. In the case of the original photocopy, the low contrast of the image seems to give a lower number of patterns.

Nevertheless it seems reasonable to state that a higher number of vertical rows leads to a better segmentation. The drawback of this is the need for a higher computation time.

Some fine-tuning of the pattern generation process (as might be deduced from the final placement in Figure 2) and of the SOM dynamics should be arise into a more robust segmentation .

4. Conclusions and future works

Conclusions from this work about the SOMs ability to segment lines in digitized texts are that it is a feasible approach in plain text documents and an interesting alternative to other sophisticated techniques used in signal processing. It relies on known features of text documents which are easily detected with the modified SOM model.

As a guide for future work, there are several different paths:

1. Modifying the pattern selection procedure, for a more robust pattern selection.
2. Trying to use more complex and inhomogeneous SOMs in order to segment more complex documents (multicolumn, with images and textures).
3. Speeding up the SOM´s learning algorithm.
4. Including the ability to segment individual letters and characters, once each text line is segmented.

References

[1] T. Kohonen: “Self-Organized formation of topologically correct feature maps”, *Biological Cybernetics*, Vol. 43 (2), 1982.

[2] T. Kohonen: “Self-Organization and Associative Memory”, *Springer Verlag*, Berlin, 1989.

[3] B. Angéniol, Gaël De La Croix Vaubois and J. Le Texier: “Self-Organizing Feature Maps and the Travelling Salesman Problem”, *Neural Networks*, Vol. 1, pp. 289-293, 1988.

[4] R. Durbin, D. Willshaw: “An Analogue Approach to the Travelling Salesman Problem Using An Elastic Net Method”, *Nature*, 326, 689-691.

[5] G. Taubin and D.B. Cooper, “Recognition and Positioning of Rigid Objects using Algebraic Moment Invariants”, SPIE, Geometric Methods in Computer Vision, pp. 175-186, 1991.

[6] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem and J. S. B. Mitchell. “An Efficiently Computable Metric for Comparing Polygonal Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, 3, pp. 209-216, 1991.

[7] J.M. Molina, M.J. Martin, P. Isasi and A. Sanchis, “A Fuzzy Reasoning System for Boundary Detection in Radiological Images”, IEEE International Conference on Fuzzy Systems, Vol. 2, pp. 1524-1529, 1998.

[8] O. Okun, M. Pietikainen and J. Sauvola. “Robust Document skew Detection Based on Line Extraction”. Proc. of the 11th Scandinavian Conference on Image Analysis (SCIA'99), June 7-11, Kangerlussuaq, Greenland, 457-464. 1999.

[9] C. Strouthopoulus and N. Papamarkos. “Text idenfication for document image analysis using a neural network”. Image and Vision Computing, 16, 879-896, 1998.

[10] M. Nadler. “A survey of Document Segmentation and Coding Techniques”. Computer Vision and Image Processing, 28, 240-262, 1984.