Neural Process Lett (2007) 26:159–175
DOI 10.1007/s11063-007-9049-6

Iterative Contextual Recurrent Classification of Chromosomes

César Martínez · Alfons Juan · Francisco Casacuberta

Received: 18 October 2006 / Accepted: 9 August 2007 / Published online: 18 September 2007

Abstract Recurrent connectionist models provide a method to represent dynamic patterns in a neural network. In this work we present a method for chromosome classification based on an almost unexplored neural network technique for this task. A partially recurrent connectionist model, the Elman network, is managed to capture the dark and light band patterns of the different classes. The proposed method is completed with the formulation of the ICC (iterative contextual classification) algorithm in order to restrict the classification to the cell context, and is applied to the neural network results. The *Copenhagen* data set was used in the experiments, where a cross-validation method was applied in order to obtain statistically representative results using the complete corpus. The entire system obtained very good results for this task, improving the performance of other neural network approaches.

Keywords Chromosome classification · Recurrent neural networks · Elman network · Context-dependent classification · Iterative algorithm

Abbreviations

RNN	Recurrent neural network
EN	Elman network
ICC	iterative contextual classification
CER	classification error rate
ML	Maximum likelihood

C. Martínez (✉)
Facultad de Ingeniería, Universidad Nacional de Entre Ríos, Ruta 11, Km. 10., Paraná, Entre Ríos E3100XAD, Argentina
e-mail: cmartinez@bioingenieria.edu.ar

C. Martínez
Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral, Ciudad Universitaria Barrio El Pozo, Santa Fe S3001XAI, Argentina

A. Juan · F. Casacuberta
DSIC/ITI, Universitat Politècnica de València, Camí de Vera s/n, València 46071, Spain
e-mail: ajuan@dsic.upv.es

F. Casacuberta
e-mail: fcn@dsic.upv.es

1 Introduction

Cytogenetics is the study of genetic constitution of individuals. At cell scale, a number of distinct bodies called *chromosomes* are the objects used to represent the genetic material. The classification task consists of two steps: (1) taking a microphotograph of a metaphase cell, where all the chromosomes are disordered and sometimes overlapping, and then (2) obtaining the *karyotype*. Considering a normal, nucleated human cell, which contains 46 chromosomes, the karyotype consists of a structure showing the complete set organized into 22 pairs of matching homologous chromosomes (the *autosomes*), and two sex chromosomes. The pairs of autosomes are ordered by length and designated with class numbers 1–22, whereas the sex chromosomes are denoted by the letters *X* and *Y* (pair *XX* for a normal female cell, and *XY* for a normal male cell). Figure 1 shows a typical karyogram of a normal male cell. This study is a clinical routine in newborns and a very important one in other cases, since it allows the detection of structural and numerical anomalies of the cell, which correspond to physical abnormalities or different pathologies of the individual. For a thorough discussion of biological and genetic aspects, we refer the reader to [1].

In the initial development of an automatic classification system, the number of chromosomes of each class in a cell was left aside, what constitutes the so-called *context-independent* (or context-free) classification. In order to resemble the methodology used by the cytogenetists in the clinical routine, this information can be exploited as a restriction in the class assignment, thus improving the error rate in the so-called *contextual* (or context-dependent) classification.

Among others, solutions found in literature consisted of discrete and continuous hidden Markov models [2,3]; homologue pairing via a maximum likelihood (ML) criterion [4,5] and Bayesian methods of finding an adequate statistical distribution. For the last case, the model

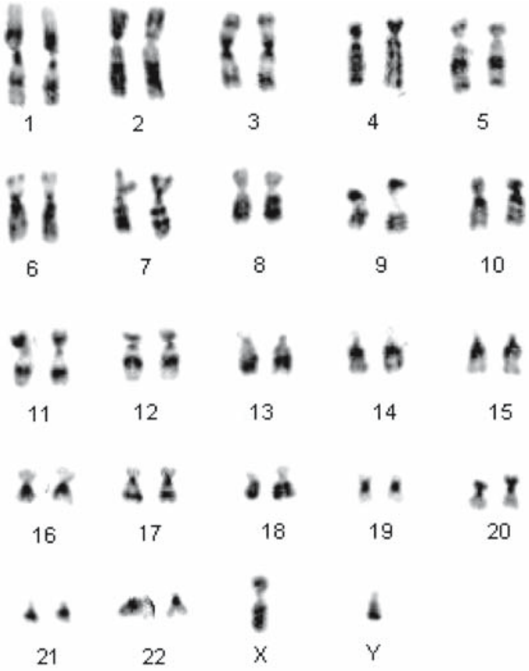

Fig. 1 Karyogram of a normal cell in metaphase (images obtained from the Copenhagen data set)

evolved from the classical Gaussian distribution [6,7] to elliptic and quadratically asymmetric distributions [8]. In the field of neural networks, the multilayer perceptron was applied [9–11] with the drawback that feature vectors have the same length for all the classes. In order to overcome this problem, some applications were attempted that worked only with geometrical features [12] or mappings [13,14]. Other architectures explored include Probabilistic Neural Network [15], which does not require an intensive training process; and hierarchical systems to produce a classification from broad groups to each one of the chromosome class [16].

The context-dependent classification involves a rearrangement of the chromosomes in order to apply the karyotype information, and this operation was also addressed in literature. A model to pair chromosomes based on a ML classifier was proposed in [17], working only on normal cells. In [6] the authors compared the performance of a context-independent ML classifier against a Bayesian classifier that moves chromosomes from one populated class to other classes with too few chromosomes (with penalties for bad assignments) and an adaptation to this problem of the well-known transportation algorithm in linear programming [18]. More recently, a maximum-weight graph matching for homologue pairing was proposed in [4], which optimize a ML criterion finding all the homologue pairs simultaneously. The same group then proposed a method that jointly optimizes the classification and pairing problems based on a three-dimensional ML assignment, including refinements for handling incomplete cells [5].

This paper describes the development of a new method for the classification of microscopic greyscale images of chromosomes, focused on two main goals. The first one consisted of adapting a type of connectionist system not extensively explored for the task in literature, the recurrent neural networks (RNN). These networks have been formulated to cope with dynamic patterns, in the sense that the input at any time is a small piece of the complete pattern, and the class assignment of the network is taken after the complete pattern presentation. Using this paradigm, the chromosomes of a cell are first separately classified by the RNN in a context-independent framework, using a sliding window containing a piece of the banding pattern as input to the RNN. The second goal was to formulate an algorithm to apply the restriction to contextual classification once all the chromosomes of a cell had been previously classified by the RNN. This algorithm looks for the pair of chromosomes of each class by an iterative method that optimizes the previous results and was called ICC, which stands for iterative contextual classification.

The organization of the paper is as follows. Section 2 exposes the architecture and operation of the RNN. Section 3 gives the formulation of the ICC algorithm for contextual classification. In Sect. 4, the data set and the feature extraction method are explained. Section 5 shows the experiments carried out in this work and the results obtained, along with the discussion drawn from this study. Finally, Sect. 6 concludes the paper.

2 Elman Networks

The formulation of the multilayer perceptron had a great impact on the development of connectionist models, given that a complete spatial representation of the objects is adequate. However, in a number of applications, the patterns are built from a temporal series of events, and they usually have unequal durations. In this work, we adapted the unidimensional concept of temporal signal processing to the bidimensional field of images, where they can be thought as temporal series of dark and light bands along the longitudinal medial axis. According to this reasoning, the time concept is given by the sequence of slices (henceforth, frames) that form the bands, and the temporal differences in duration are given by the differences in length between classes, as can be seen in Fig. 1.

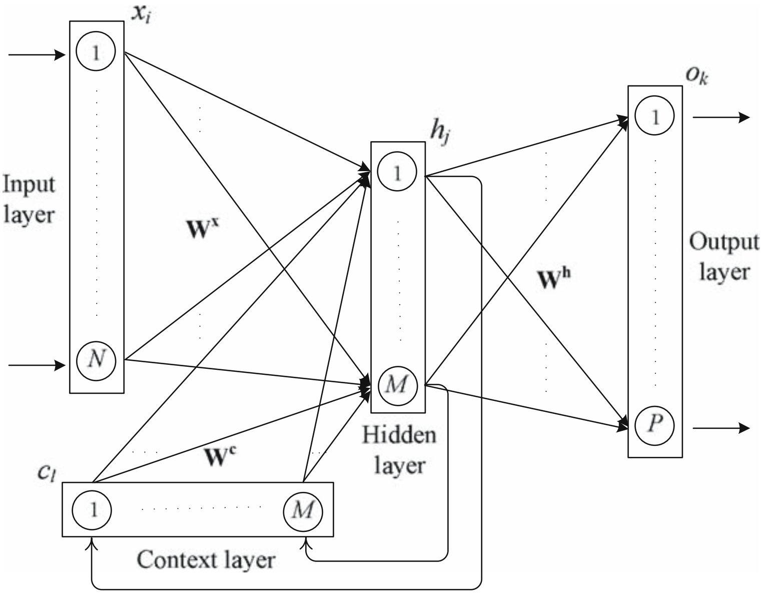

Fig. 2 Illustration of the architecture of the Elman network. The notation x_i is used to mean the inputs to the network, h_j and c_l means the activations of the hidden and context nodes respectively, whereas o_k denote the outputs of the network

An important issue is how to obtain a good representation of the temporal nature of these patterns in connectionist models. The multilayer perceptron is clearly not suitable for this kind of application because it produces an static input-output mapping, and works with patterns of the same length. An approach that provides connectionist models with a type of memory, where the past events can be used to improve the network decision on the actual event, is the general proposition of RNN [19–21]. In these models, there exist links from hidden or output units to the input layer, so a weighted version of the temporal series extends the input pattern acting as the desired memory.

Among the different RNN proposed in literature, the Elman network (EN) presents some advantages because of its internal time representation. The connections are mainly feedforward such as in a multilayer perceptron, so when processing a pattern, the network receives a frame in its input layer and propagates the activations to the hidden layer. The activations of this layer are, as well, forwarded to the output layer, and also forwarded to a set of units that constitute the *context layer*. Then, when the following frame is presented to the network, the activation values of the context layer are also forwarded to the hidden layer. In this manner, the hidden layer takes action by evaluating a kind of augmented input, considering the actual frame and a history of most recent activations of the network, which is provided by the context layer [22].

Next, we review the learning process. Let us define the structure of an EN as an input layer, a hidden layer with its context layer, and an output layer, as shown in Fig. 2. Here, we denote with N, M and P the dimensions of the input, hidden and output layer, respectively. As the outputs of the hidden layer are copied to the context layer, both have the same dimension.

Let $\mathbf{W^x} = \{w^x_{ij}\}$, $\mathbf{W^h} = \{w^h_{jk}\}$ and $\mathbf{W^c} = \{w^c_{lj}\}$ be the weight matrices for the input-to-hidden, hidden-to-output and context-to-hidden links respectively; with $1 \le i \le N$, $1 \le j, l \le M$, and $1 \le k \le P$. The weights for the hidden-to-context links are fixed to 1.

Let us define the linear output of each node as the linear combination of its inputs and weights, for the output and hidden layer respectively, as:

$$O_k(t) = \sum_{j=1}^{M} w^h_{jk}\, h_j(t), \tag{1}$$

$$H_j(t) = \sum_{i=1}^{N} w^x_{ij}\, x_i(t) + \sum_{l=1}^{M} w^c_{lj}\, c_l(t). \tag{2}$$

Here, x_i is the ith input to the network, whereas the activations of each node are obtained in the forward pass, as described by the following difference equations:

$$o_k(t+1) = f\left(O_k(t)\right), \quad 1 \le j \le M, \quad \text{for the output layer} \tag{3}$$

$$h_j(t+1) = f\left(H_j(t)\right), \quad 1 \le i \le N, \quad \text{for the hidden layer} \tag{4}$$

$$c_l(t+1) = h_j(t), \quad \forall\, l = j, \quad \text{for the context layer} \tag{5}$$

where $f(\cdot)$ is a non-linear function, specifically the logistic function $f(x) = 1/(1+e^{-x})$ used in this work. At time $t = 0$, when the first frame of a new input pattern is presented to the network, the activations of the hidden and context layer are set to 0.

Once the output of the network was obtained, the second step of the training process is the backward pass. In this stage, the difference between target and actual values is taken as the network error, and the accumulation over all the frames of each pattern is the function the backpropagation tries to minimize.

The error function at the output units, for the complete pattern, is defined as:

$$E = \frac{1}{2} \sum_{t=1}^{R} \sum_{k=1}^{P} \Big[s_k(t) - o_k(t) \Big]^2, \tag{6}$$

where R is the length of the pattern (number of frames) and $s_k(t)$ is the target output at time t.

By applying the general error-correction method of the backpropagation with momentum term and the derivative chain rule, the weights are updated according to the following formulas:

$$\Delta w^h_{jk} = \sum_{t=1}^{R} \Big[\eta\, h_j(t)\, \delta_k(t) + \mu\, \Delta w^h_{jk}(t) \Big], \tag{7}$$

$$\Delta w^x_{ij} = \sum_{t=1}^{R} \Big[\eta\, x_i(t)\, f'_j\Big(H_j(t)\Big) \sum_{k=1}^{p} \Big(\delta_k(t) w^h_{jk}\Big) + \mu\, \Delta w^x_{ij}(t) \Big], \tag{8}$$

$$\Delta w^c_{lj} = \sum_{t=1}^{R} \Big[\eta\, c_l(t)\, f'_j\Big(H_j(t)\Big) \sum_{k=1}^{p} \Big(\delta_k(t) w^h_{jk}\Big) + \mu\, \Delta w^c_{lj}(t) \Big], \tag{9}$$

with

$$\delta_k(t) = f'_k\Big(O_k(t)\Big)\, \Big(s_k(t) - o_k(t)\Big). \tag{10}$$

Here, η is the learning parameter, which specifies the step width of the gradient descent; μ is the momentum term, which allows the addition of a proportional amount of the previous weight change; and $f'(\cdot)$ represents the derivation of the activation function.

Once a complete pattern has been processed by the network, the activations of the context layer are reset so when a new pattern is presented, the network does not have the history of activations of the predecessor.

This learning process is repeated until the error value reaches some satisfactory level.

Figure 2 shows the architecture of the EN employed in this work, whereas details of implementation and operation of the EN for the chromosome classification task are given in Sect. 4.2.

3 Iterative Algorithm for Contextual Classification

We formulated an algorithm to restrict the context-independent classification carried out by the EN to the cell context, so it works with the complete set of chromosomes pertaining to a cell. Essentially, the algorithm consists of the reassignment of the classes, previously assigned by the EN, on those chromosomes that were classified with low confidence. The confidence on the initial classification is based on the values obtained in the output layer of the network. The main process is implemented by means of an iterative search of the two chromosomes for classes 1–22 and the sex chromosomes. Iteratively, the particular pattern that is considered is the pair that obtains the highest combined confidence for a given class and the reassigned classes until that round. In this way, the karyotype information is taken into account in order to fix up some erroneous class assignments that could be made by the neural network. The algorithm was named ICC, which stands for iterative contextual classification.

The requirement of the ICC algorithm is, for each chromosome, a vector with the values obtained in the complete output layer, normalized so the sum is equal to 1. Henceforth, these values are referred to as *network probabilities*. This information is arranged in a $N \times C$ matrix $\mathbf{M}$, where N is the number of chromosomes of the cell and C is the number of classes to be determined. The network probabilities are then rearranged in decreasing order (denoted in Algorithm 1 by the function 'reorderProbabilities()'), so the resulting matrix have the following properties:

$$\sum_{k=1}^{C} \mathbf{M}(p,k) = 1, \quad 1 \leq p \leq N,$$
$$\mathbf{M}(p,1) > \mathbf{M}(p,2) > \cdots > \mathbf{M}(p,C), \quad 1 \leq p \leq N.$$

The core of the algorithm is the search of a *solved class*, which is that one with only 2 patterns in the cell having its highest network output probabilities for that class. As an additional requirement, a minimum bound for both probabilities of 0.8 is set in order to ensure an initial classification with good certainty. Thus, the ICC considers that the patterns of a solved class were well classified by the EN, and therefore does not include them in the posterior class reassignment.

Before the iterative search, the ICC registers all the solved classes resulting from the EN classification, but those patterns are not considered in advance. In successive rounds, the algorithm finds the rest of pairs by the *renormalization method*: for each pattern, their output probabilities according to the unsolved classes are recalculated. After that, the search of new solved classes is applied. In the case of no solved class in a round, the ICC finds the pair of

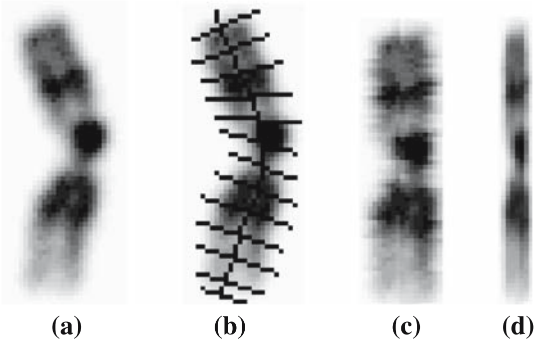

Fig. 3 Illustration of the complete preprocessing and feature extraction method. (**a**): original chromosome from the Copenhagen data set, (**b**): superposition of longitudinal axis and slices (one out of five slices were drawn for visual convenience), (**c**): unfolded chromosome, (**d**): pattern of 9 filtering points per slice

patterns with the highest probability product for a given class, and reassigns those patterns to the solved classes set. This scheme is repeated until all the classes have been solved.

The ICC does a separate treatment of the sex chromosomes, because normal cells have two chromosomes (combination X–X for female or X–Y for male cells), for which the search is restricted to 23 classes: the 22 autosomes plus one sex class, allowing the mentioned combinations. For the classification of abnormal cells, the algorithm takes into account the cases of cells with a missing chromosome, and with an extra chromosome. For the first case, it looks for the best 22 pairs and then assigns the remaining pattern to the last unsolved class. In this way, if the pattern is assigned to the sex class, it receives the X or the Y label according to its highest output probability. For the second case, once all the classes have been solved with their corresponding pairs, the extra chromosome is assigned to the class with highest output probability.

Algorithm 1 lists the complete algorithm for the generic case of a normal cell (the alternatives for abnormal cells can be easily sketched).

4 Experimental Framework

4.1 Corpus, Preprocessing and Feature Extraction

The data set of chromosome images used in this work was the *Cpa* corpus, a corrected and augmented version of the prior Copenhagen corpus called *Cpr*.

The corpus contains the segmented images of 2,804 karyotyped human metaphase cells; 1,344 of which are female and 1,460 are male. There is one missing chromosome in 26 cells and one extra chromosome in 37 cells, and one cell contains 48 chromosomes [8].

The chromosomes may present some bending differences among cells, and the features need to be extracted in an invariant format, so an unfolding of the chromosome is applied along its longitudinal axis (the medial line along which the dark and light bands are alternating). This operation is carried out by re-sampling the image over a series of perpendicular lines to the axis called *slices*, placed at equidistant distance. The unfolded version of the chromosome is generated by horizontally arranging the slices, organized in a column. Figure 3 shows the different stages of the feature extraction method: (a) a chromosome image as obtained in the metaphase stage, (b) the longitudinal axis and the transverse lines, (c) the unfolded version of the chromosome, and (d) the features calculated over the transverse lines. Next is the detailed description of the complete method.

```
Data:
    $N$: number of patterns of the cell,
    $C$: number of chromosome classes,
    $\mathbf{M}$: $N \times C$ matrix of output probabilities given by the EN.
Result:
    $\mathbf{R} = \{(t, o)\}$: $N \times 2$ matrix of pairs (teaching output class, reassigned class) for each pattern.
    $V$: vector containing the solved classes.
    $T$: vector containing the patterns solved.

subroutine SearchSC($\mathbf{M}$, $V$, $T$, $c$)
$c \longleftarrow$ false
for $1 \leq k \leq C$ do
    if $\left[\right.$ (# of patterns with output class $= k) = 2 \ \wedge$ (both probabilities $> 0.8$)$\left.\right]$ then
        $T \Leftarrow$ both pattern index $p'$
        $V \Leftarrow$ chromosome class $k$
        $\mathbf{R} \Leftarrow \{$teaching output of patterns $p', k\}$
        $c \longleftarrow$ true
    end
end
return c

initialization
    $\mathbf{R}, T, V \Leftarrow \emptyset$
begin
    reorderProbabilities($\mathbf{M}$)
    searchSC($\mathbf{M}$, $V$, $T$, $c$)                          /* search for solved classes */
    while $\#V < C$ do                                       /* -> Iterative search <- */
        for $1 \leq p \leq N,\ p \notin T$ do                             /* renormalization */
            $s \longleftarrow \sum_{k=1, k \notin V}^{C} \mathbf{M}(p, k)$
            $\mathbf{M}(p, :) \longleftarrow \mathbf{M}(p, :)/s$
        end
        searchSC($\mathbf{M}$, $V$, $T$, $c$)                  /* search for new solved classes */
        if $c$=false then                                        /* no classes solved */
            reorderProbabilities($\mathbf{M}$)
            $P, Q \Leftarrow \emptyset$
            for $1 \leq k \leq C,\ k \notin V$ do                   /* select pairs of patterns */
                                                             /* from unsolved classes */
                $Q(k) \Leftarrow$ the two patterns with highest probability between all the patterns of class $k$
                $P(k) \Leftarrow$ the probability product of patterns in $Q(k)$
            end
            $i \longleftarrow \mathrm{argmax}_i\ P(i)$                                /* get the best pair */
            $V \Leftarrow$ output class of patterns in $Q(i)$
            $T \Leftarrow$ patterns in $Q(i)$
            $\mathbf{R} \longleftarrow \{$teaching output of patterns in $Q(i)$,output class of patterns in $Q(i)\}$
        end
    end
end
```

Algorithm 1. ICC (iterative contextual classification) algorithm.

A key step in the preprocessing is to derive a good estimation of the longitudinal axis of the chromosome. Thus, the first operation is to fill the holes present in some images due to acquisition problems, which could cause some cut of the axis. The filling is carried out by means of a contour chain algorithm. Other problem in the axis calculus is the rough border, which could produce an axis with more than two terminals. To solve this problem, two mathematic morphology algorithms, dilation and erosion, are applied to smooth the border. Finally,

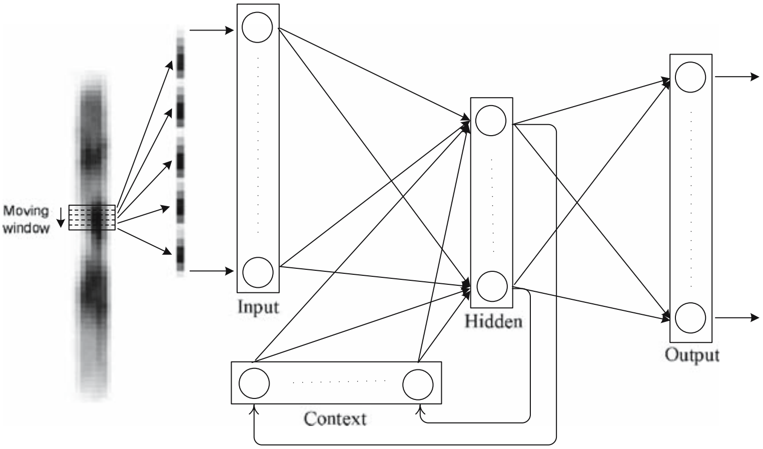

Fig. 4 Illustration of the use of an Elman network for chromosome classification (chromosome image obtained from the Copenhagen data set)

to obtain the axis, the skeletonization of the chromosomes [6,23] is applied in a sequence of three steps. The first one consists of obtaining a first approach of the skeleton by means of the Gonzalez–Woods algorithm [24]: it iterates two passes on the contour points to mark and delete them if a number of requirements are met, which prevent excessive erosion or obtaining disconnected points. The obtained skeleton has all the points within the chromosome, so the second step consists of extending the ends of the skeleton until covering the total length of the chromosome, in tangent direction at each end point. This skeleton is 4-adjacent, so the third step is to clean some points by the application of the Hilditch algorithm [25] in order to obtain an 8-adjacent axis. This algorithm scans the chromosome from left to right and from top to bottom, checking a series of conditions to flag the contour points and delete those that do not belong to the skeleton.

This study deals with the characterization of band patterns for each class working in a *local* manner, so we introduce here a new set of measurements of grey values in each slice, which constitute a frame. The measurements consisted of the output of a 2-D average filter. The set was composed of 9 filtering points uniformly spaced between the beginning and the end of each slice (which is usually larger than 9 pixels), perpendicularly to the medial axis. The filter applied to the image $f(x, y)$ had the following form:

$$g(x, y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} m_{s+a,t+b}\, f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} m_{s+a,t+b}}, \tag{11}$$

with $a = b = 2$ and $m_{s,t}$ being the coefficients of a weighted averaging spatial mask with the following distribution:

$$\mathbf{m} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 2 & 16 & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

where the decreasing weights from the center reduce the blurring introduced in the sampling process [24].

In Fig. 3(d), the slices are drawn in the horizontal lines, with the 9 points of the frames given by the pixels of each line.

4.2 Neural Networks

For the operation of the Elman networks, we proceed as follows. The banding pattern of the chromosomes is given by the slices of filtering points as obtained in the previous section. One way of network processing is to feed the input layer with the slices, one per time, and apply the algorithms described in Sect. 2 for training and testing. In a preliminary work [26] it was demonstrated that the network performance is greatly improved if the input layer is fed with a small portion of the banding pattern made up of a number of consecutive frames, constituting a *moving-window* over the image. The window is composed by arranging the frames consecutively as a vector, where the index 1–9 of the window contains the first frame, index 10–18 the second frame and so on. At each time, the window serves as input to the network, so the network is allowed to work not only with the actual frame under analysis, but also with a local context of the banding pattern. The sliding movement of the window resembles the temporal behaviour of the input pattern processing: once a window has been processed, a new window is taken from the next frames. In this work, the window is comprised of the concatenation of 5 consecutive slices.

The topology of the network implemented in this work consisted of an input layer of 45 units, in order to fit a moving-window of 5 frames, with 9 points each. The hidden layer had a variable number of units, which was adjusted in the experiments. The activations of the hidden units are copied on a one-to-one basis, so the context layer had the same number of units that the hidden layer. The output layer had 24 units, one for each of the classes to manage, without context layer.

The weights of the forward connections were randomly initialized from the interval $[-1.0, 1.0]$. The context units extend the input layer by feeding back the outputs of the hidden layer with non-trainable weights fixed to 1.0 [22], and there are no self-recurrent links from the context units to themselves.

As was detailed in Sect. 2, the standard backpropagation algorithm with momentum-term for partial recurrent networks was used as the learning function. After initial experiments for parameters adjustment, the learning rate parameter was set to 0.2 and the momentum term to 0.1. The maximum difference $d_j = t_j - o_j$ between a teaching output t_j and the value o_j of each output unit was set to 0.1, meaning that values above 0.9 were treated as 1, while values below 0.1 as 0, preventing overtraining of the network.

Figure 4 illustrate the use of EN for chromosome classification, as used in this work. The window is shown with the frames rotated by 90 degrees with respect to the medial axis of the chromosome.

5 Results and Discussion

For all the experiments, the classification performance of the systems was evaluated through the *classification error rate* (CER), considering the error on the test-set as given by CER $= w/(w + r) \times 100\%$, where w is the number of patterns wrong classified and r is the number of those correctly classified.

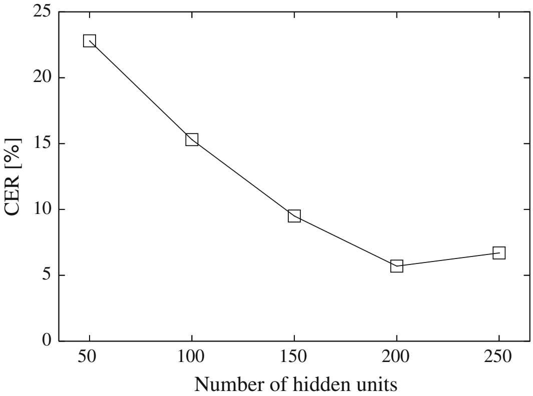

Fig. 5 Results for the adjustment of the network architecture, changing the size of the hidden layer. The minimum error rate obtained was 5.7%, for 200 neurons

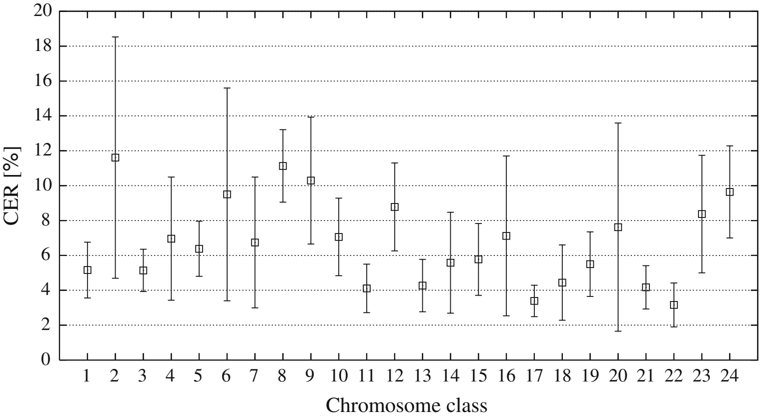

Fig. 6 CER's mean and standard deviation (central and end marks) for each chromosome class, as obtained from the seven partitions of the context-independent cross-validation experiment

The operation of a network strongly depends on the chosen structure and the selected parameters. Initial experiments to adjust the size of the hidden layer were carried out on a neural network (prior to the complete experimentation) using a subcorpus of the complete training set. The number of hidden units was changed from 50 to 250 units and each time the network was trained up to the generalization point. The optimum number of hidden units was found as the one that allows for obtaining the minimum CER. This fixes the network architecture for the remaining experiments to 45/200/24 (number of input/hidden/output units), as shown in Fig. 5.

The experiment with the complete database is made in two stages. The first one consists of the chromosome classification in a context-independent framework, as performed by the EN. The second one corresponds to the restriction to cell-context classification, as given by the application of the ICC algorithm.

For the context-independent stage, a cross-validation strategy in seven disjoint partitions was applied, in order to reduce the bias in the estimation of the classifier performance. Each partition consisted of 400 cells for the test set, and the remaining 2,400 cells have been used for the training set. Seven different networks were trained using these partitions.

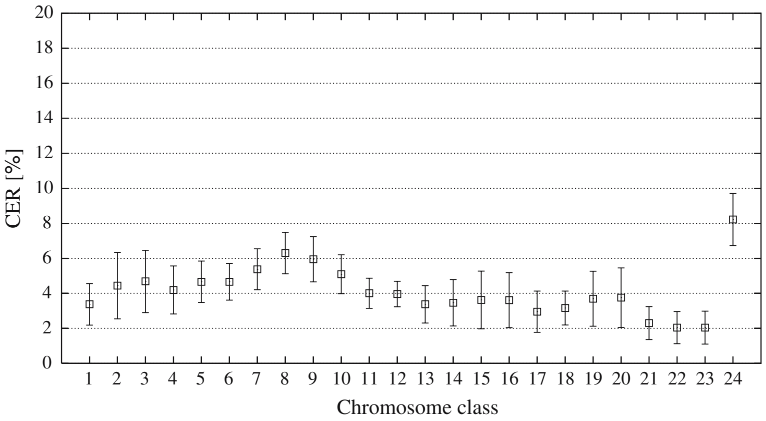

Fig. 7 CER's mean and standard deviation (central and end marks) obtained after the application of the ICC algorithm

Table 1 Results for the complete cross-validation experiment

Mode	p_1	p_2	p_3	p_4	p_5	p_6	p_7	Mean
Context-independent	6.0	6.7	6.1	8.3	7.5	5.6	6.8	6.7
Context-dependent	3.6	3.8	3.7	4.9	4.8	2.7	4.0	3.9

In rows, the CER for the context-independent classification and after the application of the ICC algorithm. In columns, each one of the p partitions of the experiments

Figure 6 summarizes some statistics of the CER value as obtained in the context-independent experiment. For each chromosome class, the error bars indicate the CER's arithmetic mean (central mark) and the standard deviation (end marks) of the seven partitions. This figure clearly shows the natural difficulty of the chromosome classification task for the EN, where some classes (i.e., 17 and 22) are being considerable better estimated than others (i.e., 2 and 20). Also it can be observed the large dispersion in the results between partitions for some classes, where CER values above 20% are reached.

The context-independent error rate for the complete experiment was 6.7%, obtained as the arithmetic mean CER of the seven partitions. The ICC algorithm was then applied to the context-independent results, producing a mean CER of 3.9%. Table 1 summarizes the results obtained for each partition, contrasting the two approaches. Figure 7 shows the mean and standard deviation for the seven partitions, as in Fig. 6. It can be easily observed the improvement in the CER for each chromosome class, and the significant reduction in the dispersion of the results between partitions.

The computational cost of the EN training of each partition was approximately 600 CPU hours on a Pentium IV-based PC. The application of the ICC algorithm takes a mean time of 1.6 ms per cell, depending on the previous network classification.

In order to validate our results and the performance of the ICC algorithm proposed here, we evaluated the statistical significance of the error rates for the context-independent classification (ε_{ci}) versus that obtained after the application of the ICC algorithm (ε_{cd}) [27]. We

Table 2 Results for different classifiers using Copenhagen chromosomes

System	Test-set error
Multilayer perceptron with reduced-sized output layer [11]	11.7(%)
Standard multilayer perceptron [10]	6.5(%)
Hierarchical neural network [16]	5.6(%)
Continuous hidden Markov model-ICC [3]	4.6(%)
Elman network-ICC (this work)	3.9(%)

obtained that Pr($\varepsilon_{ci} > \varepsilon_{cd}$)> 99.999%, showing the important benefit of the application of the ICC algorithm in the classification.

For comparison purposes with the other neural networks approaches working with the Copenhagen chromosomes, we present in Table 2 the summary of results for our work; a multilayer perceptron with a reduced-sized output layer proposed in [11]; a standard multilayer perceptron with one hidden layer proposed in [10]; and a hierarchical system that classifies the patterns firstly in 7 major groups of classes, and then in one of the 24 karyotype classes, proposed in [16]. Also, results of previous work with the use of continuous hidden Markov models as the context-independent classifier [3] are included. As can be seen, the system proposed in this work improves the error rates of the above-mentioned systems. This allows us to validate the hypothesis that the dynamic treatment of the patterns carried out by the recurrent neural network would add rich information to the classification.

With respect to the state-of-the-art results, the performance obtained in this work constitutes, to our knowledge, the best results obtained using the standard preprocessing technique and only taking into account the banding patterns. Regarding this subject, in [26] we showed that the recurrent neural network clearly outperforms the multilayer perceptron when using only the banding pattern as input. Instead, the context-independent classification could be improved if other widely employed features (e.g., global measurements, centromeric index) were used together with other well-known classifiers for this task. In this sense, the preliminary classification of each chromosome is relatively more important to determine the final error rate than the re-arrangement stage, because the ICC performance depends exclusively on the output probabilities assigned for every class.

The ICC algorithm applied here obtains a mean reduction of about 50% over the context-independent error rate (3 points), as it was presented in Table 1. This is a good performance for a re-arrangement method, given that in [6] the authors found that the transportation algorithm reduced the error rate but in a rather small amount (from 6.5% to 4.4% on the Copenhagen dataset). The assignment approach that combine the classification and pairing of chromosomes proposed in [5] performs better than the transportation algorithm (nearly 33%, because this last method does not exploit the feature similarities within a cell), obtaining reductions of about 3 points.

In order to lower the error rate we have to deal with statistical robustness. In [8] the authors presented other complex, ad-hoc preprocessing technique, the *dominant points* that use a different axis calculation, combined with the *variants* method which obtains multiple feature vectors by chromosome and allows to process the ambiguities that could appear in the feature extraction process (e.g., polarities). The method obtains an error rate of 0.61% using a statistically robust model (with *elliptical symmetry*) on the Cpa corpus, and an error rate of

1.8% on a personal dataset called Pki [28]. An open question is to investigate if these lower rates could be achieved with a system based on neural networks that combine the advantages of statistical robust models and proper treatment of outliers.

Even though the chromosome images of the corpus had been acquired at different times, and they have also some flaws due to segmentation errors of overlapping chromosomes, the preprocessing and feature extraction presented here have proved to be robust against variability in the images. This scheme overcomes the robustness limitations reported by other authors [7].

There is also a subtle issue about a piece of information also provided by the Cpa corpus. The karyogram as shown in Fig. 1 have all the chromosomes oriented top-down in a particular form: the short arm is located above the long arm, with the *centromere* being the constriction that separates both arms, in some classes located in the center of the chromosome whereas, in others, is located eccentrically. The orientation, or polarity, is the same in the training and testing phase, so the system does not take into account the polarity information as in [7]. Other works make of the polarity an issue that must also be solved by the classifier, which is known in the training phase and estimated in the preprocessing step for testing [17], or the above-mentioned method of variants is applied [29].

The behavior of the ICC algorithm depends to a great extent not only on the winner class obtained by the neural network for each chromosome but also on the complete vector of *network probabilities* of matrix M. Thus, the classification of the complete cell must be made with good accuracy mainly in the first places of these vectors, in order to accomplish a subsequent rearrangement that significantly lowers the error rate. The complete method achieves, for 68% of the cells, a correct classification of 100%, obtaining reductions of more than 10% in the error rate. 21% of the cells could not be classified with high accuracy by the neural network, and the method reduces the error rate but the cells are not classified with 100% of accuracy. On the other hand, for 7% of the cells, the application of the ICC obtains the same error rate that the previous one obtained by the neural network, whereas 4% of the cells are finally worse classified. Figure 8 show the confusion matrix before and after the application of the ICC algorithm, for a good and a bad example of two normal cells: we represent in rows the chromosomes and in columns the karyotype classes, with the black boxes meaning a correct classification, and the gray boxes meaning an error. In the first case, the neural network achieves 89% of correct classification and the ICC is able to correct all the previous errors. In the second case, the neural network assigns two extra chromosomes to class 7 (chromosomes number 12 and 15). Then, the ICC algorithm chooses two of the four chromosomes in this class to move them, but the wrong pair is chosen (chromosomes number 14 and 15). This introduces an error on chromosome 14 while maintaining the previous errors on chromosomes 12 and 15, so that the accuracy of the classification is reduced from 95% to 93%.

6 Conclusions and Future Works

A new approach to chromosome classification was proposed based on an almost unexplored neural network for this task. The EN has managed to work with a type of local information along the chromosome: a sampled version of the banding pattern. This information is treated as a temporal unidimensional vector, where the time concept resembles the way the chromosome image is explored along its longitudinal axis. Therefore, the EN can manage patterns of different lengths, a key characteristic of the karyotype, and is a drawback with the multilayer perceptron-based approaches. This paper presents follow-up research on

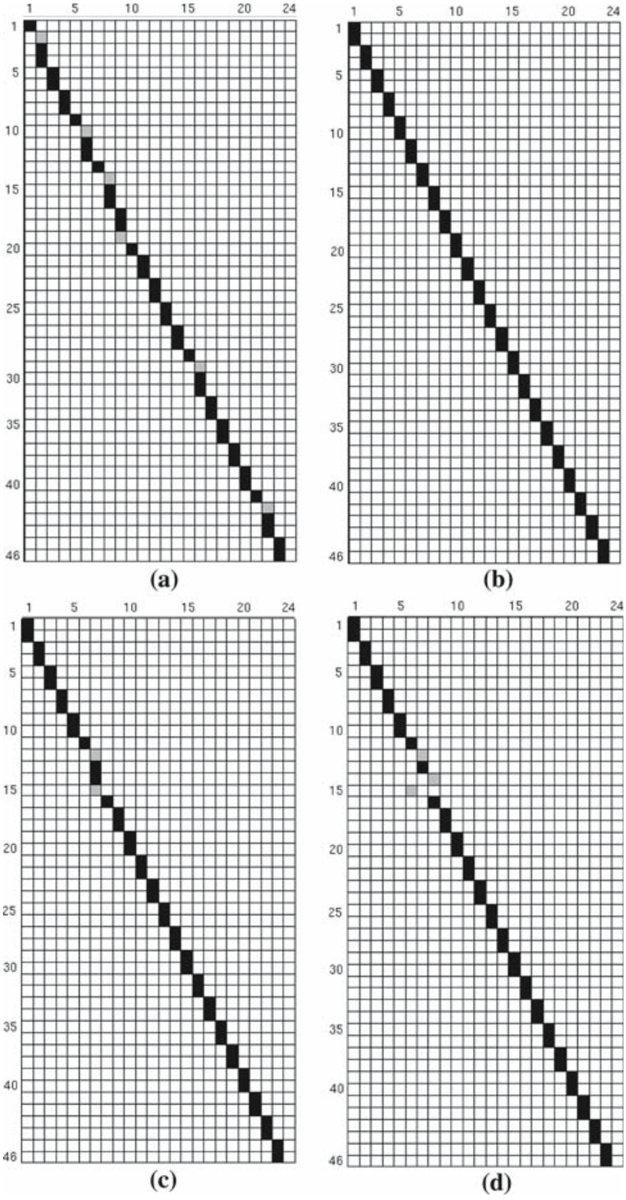

Fig. 8 Confusion matrix for two cells showing the context-independent classification on the left and context-dependent on the right. The first cell was classified with an accuracy of 89% by the neural network (**a**), after corrected to a 100% by the ICC algorithm (**b**). The second cell obtained a 95% by the neural network (**c**), after reduced to 93% (**d**)

a preliminary work in which the EN was applied to context-independent chromosome classification [26], with promising results on a reduced set of the complete corpus considered here.

A context-dependent classification method, the ICC (iterative contextual classification) algorithm was formulated. It performs a relabelling of the classes assigned by the EN by using the karyotype information for a number of expected chromosomes for each class in a cell. The method proposed here provides a significant improvement with respect to other neural network works reported by different authors in this context, and also outperforms the results obtained by the authors using continuous hidden Markov models as the context-independent classifier.

The results obtained constitute a very good performance for this task, considering that the preprocessing and feature extraction scheme was attempted in order to follow the guidelines reported in the literature. In addition, by working only with the banding profiles, the complexity and computational cost of the feature extraction block was reduced.

Some improvements could be applied to the system in order to obtain a lower context-independent error rate, which would facilitate the relabelling performed by the ICC algorithm. In the feature extraction stage, for example, the skeleton technique used to obtain the longitudinal axis has the disadvantage of being non-parametric. This causes some problems in the slice calculation, which can be seen mainly in short chromosomes and in several-bended chromosomes (almost circle-shaped), and could be overcomed with a parametric approach as it was explored in [8,29]. Also, new features like Fourier or wavelet descriptors or the canonical pattern of all the chromosomes as in [4], could be incorporated in order to add useful information to the classification stage.

Acknowledgements The authors wish to thank Dr. Gunter Ritter (*Fakultät für Mathematik und Informatik, Universität Passau, Passau, Germany*) for kindly providing the database of chromosome images used in this work, and the anonymous reviewers for their valuable comments which contributed to improve this paper. Work supported by "Universidad Nacional de Entre Ríos" under grants UNER-PICT 2002 11-12700 A and UNER-PME03/00151, and "Universidad Nacional del Litoral" under grant PROMAC-POS 2004.

References

1. Hartl D, Jones E (1998) Genetics: principles and analysis. Gareth Stevens Publishing
2. Gregor J, Thomason MG (1996) A disagreement count scheme for inference of constrained Markov networks. In: Miclet L, de la Higuera C (eds) Grammatical inference: learning syntax from sentences, vol 1147 of Lecture Notes in Computer Science. Springer, pp 168–178
3. Martínez C, García H, Juan A, Casacuberta F (2003) Chromosome classification using continuous hidden Markov models. In: Proc. of the First Iberian Conference IbPRIA 2003, vol 2652 of Lecture Notes in Computer Science. Mallorca (Spain), June 2003. Springer-Verlag, pp 494–501
4. Wu X, Biyani P, Dumitrescu S (2004) Globally optimal classification and pairing of human chromosomes. In: Proc. of the 26 Annual Int. Conf. of the IEEE EMBS. San Francisco, CA, USA, September 2004, pp 2789–2792
5. Biyani P, Wu X, Sinha A (2005) Joint classification and pairing of human chromosomes. IEEE Trans Comput Biol Bioinform 2:102–109
6. Carothers A, Piper J (1994) Computer-aided classification of human chromosomes: a review. Stat Comput 4:161–171
7. Wu Q, Liu Z, Chen T, Xiong Z, Castleman K (2005) Subspace-based prototyping and classification of chromosome images. IEEE Trans Image Proce 14(9):1277–1287
8. Ritter G, Schreib G (2001) Using dominant points and variants for profile extraction from chromosomes. Pattern Recognit 34:923–938
9. Errington P, Graham J (1993) Application of artificial neural networks to chromosome classification. Cytometry 14:627–639
10. Cho J (2000) Chromosome classification using backpropagation neural networks. IEEE Eng Med Biol Magaz 19(1):28–33
11. Delshadpour S (2003) Reduced size multi layer perceptron neural network for human chromosome classification. In: Proc. of the 25 Annual Int. Conf. of the IEEE EMBS, vol 3. Cancun, Mexico pp 2249–2252

12. Mousavi P, Fels S, Ward R, Lansdorp P (2001) Classification of homologous human chromosomes using mutual information maximization. In: Proc. of the Int. Conf. on Image Processing, vol 2. pp 845–848, (October 2001)
13. Lerner B (1998) Toward a completely automatic Neural-Network-based human chromosome analysis. IEEE Trans Syst Man Cybern-Part B: Cybern 28(4):544–552
14. Lerner B, Guterman H, Aladjem M, Dinstein I (1999) A comparative study of neural network based feature extraction paradigms. Pattern Recognit Lett 20:7–14
15. Sweeney W, Musavi M, Guidi J (1994) Classification of chromosomes using a probabilistic neural network. Cytometry 16:17–24
16. Cho J, Ryu S, Woo S (2004) A study for the hierarchical artificial neural network model for giemsa-stained human chromosome classification. In: Proc. of the 26 Annual Int. Conf. of the IEEE EMBS. San Francisco, CA, USA, September 2004, pp 4588–4591
17. Piper J, Granum E (1989) On fully automatic feature measurement for banded chromosome classification. Cytometry 10:242–255
18. Tso M, Kleinschmidt P, Mitterreiter I, Graham J (1991) An efficient transportation algorithm for automatic chromosome karyotyping. Pattern Recogn Lett 12:117–126
19. Waibel A, Hanazawa T, Hinton G, Shikano K, Lang K (1999) Phoneme recognition using time-delay neural networks. IEEE Trans Acoust Speech Signal Proce 37(3):328–339
20. Dorffner G (1996) Neural networks for time series processing. Neural Network World 6(4):447–468
21. Gupta L, McAvoy M, Phegley J (2000) Classification of temporal sequences via prediction using the simple recurrent neural network. Pattern Recognit 33:1759–1770
22. Elman JL (1990) Finding structure in time. Cogn Sci 14(2):179–211
23. Piper J (1982) Interactive image enhancement and analysis of prometaphase chromosomes and their band patterns. Anal Quant Cytol 4(3):233–240
24. Gonzalez R, Woods R (2002) Digital image processing. Prentice-Hall
25. Hilditch C (1969) Linear skeletons from square cupboards. Machine Intell 19:403–420
26. Martínez C, Juan A, Casacuberta F (2002) Using recurrent neural networks for automatic chromosome classification. In: Proc. of the Int. Conf. on Artificial Neural Networks ICANN 2002, vol 2415 of Lecture Notes in Computer Science. Madrid (Spain), (August 2002). Springer-Verlag, pp 565–570
27. Toutenburg H (2002) Statistical analysis of designed experiments. Springer
28. Ritter G, Schreib G (2000) Profile and feature extraction from chromosomes. In: Proc. of the 15th International Conference on Pattern Recognition (ICPR '00), Vol 2. pp 287–290
29. Ritter G, Pesch C (2001) Polarity-free automatic classification of chromosomes. Comput Stat Data Anal 35(3):351–372