Autoassociative Learning in Relaxation Labeling Networks

Marcello Pelillo [1] and Anna M. Fanelli [2]

[1] *Dipartimento di Matematica Applicata e Informatica, Università "Ca' Foscari" di Venezia*
Via Torino 155, 30173 Venezia Mestre, Italy

[2] *Dipartimento di Informatica, Università di Bari*
Via Orabona 4, 70126 Bari, Italy

Abstract

We address the problem of training relaxation labeling processes, a popular class of parallel iterative procedures widely employed in pattern recognition and computer vision. The approach discussed here is based on a theory of consistency developed by Hummel and Zucker, and contrasts with a recently introduced learning strategy which can be regarded as heteroassociative*, i.e., what is actually learned is the association between patterns rather than the patterns themselves. The proposed learning model is instead* autoassociative *and involves making a set of training patterns* consistent*, in the sense rigorously defined by Hummel and Zucker; this implies that they become local attractors of the relaxation labeling dynamical system. The learning problem is formulated in terms of solving a system of linear inequalities, and a straightforward iterative algorithm is presented to accomplish this. The learning model described here allows one to view the relaxation labeling process as a kind of asymmetric associative memory, the effectiveness of which is demonstrated experimentally.*

1 Introduction

Relaxation labeling processes are a broad class of parallel cooperative procedures widely employed in computer vision and pattern recognition. They were pioneered by Rosenfeld, Hummel, and Zucker [1] in the mid-1970s and have since found applications in a variety of different problems (see [2] for a review). The basic idea behind relaxation labeling is that complex, global computations can be done in a simple cooperative network of locally-interacting processing units, each representing a specific hypothesis. This is commonly believed to be the style of computation actually implemented in the early stages of biological visual systems (e.g. [3]).

One of the most important problems that arise in applying a relaxation labeling algorithm to a practical task concerns finding a suitable set of *compatibility coefficients*. These are real-valued quantities that express the degree of agreement between different hypotheses, and effectively embody all the knowledge about the problem at hand. Traditional interpretations of compatibility coefficients have been in terms of statistical measures such as, for example, correlation [1] or mutual information [4]. The work by Peleg [5] and Hancock and Kittler [6] also supports this view.

In a recent work, we have tackled the problem from a radically different perspective [7]. Instead of deriving the compatibilities of relaxation labeling by making recourse to probabilistic analysis, we try to *learn* them from a set of labeled data, in exactly the same way that neural networks do. Besides the obvious benefits of employing an automated procedure which is capable of determining the compatibility model in an "optimal" manner, the learning-based approach contributes to make the fields of relaxation labeling and neural networks closer. It also strengthens the biological plausibility of relaxation labeling processes as a mechanism of visual computation actually implemented in the brain, a hypothesis that has been forcefully put forward by Zucker and his collaborators [8].

The approach described in [7] consists of deriving the compatibility strengths in such a way that the "distance" between what the relaxation process produces upon presentation of an input pattern and what is expected to produce be as small as possible. Technically, this amounts to minimizing a certain cost function in the space of compatibility coefficients, a problem that can be solved by either classical gradient methods or more attractive global optimization procedures like genetic algorithms, as described in [9]. The input to the learning algorithm in this case is a set of training exemplars in the form of ordered pairs of objects and labels. The objects are given as input to a local measurement process whose output will then initialize the relaxation process, while the labels represent the desired final responses. Borrowing the terminology from the neural network domain [10], this can be regarded as a *heteroassociative* approach to learning, which means that what is actually learned (or stored) is the association between pairs of patterns, not the patterns themselves. A potential drawback of this kind of approach is that the derived compatibility coefficients *do* depend on the initial local measurement process. In some cases this may be advantageous, because in this way we effectively provide the learning algorithm with more problem-dependent information. However, in some circumstances it may result in poor generalization performance. This may be true when the initial local process is particularly noisy, thereby introducing noise in the learning process.

In this paper, we develop what may be called an *autoassociative* approach to the relaxation labeling learning task which, as a by-product, avoids this kind of problems. In this case, the learning problem consists of making a set of training patterns *consistent*, in the rigorous sense defined by Hummel and Zucker [11]. This is equivalent to saying that the learning data, viewed as points in state space, become attractive fixed points for the relaxation labeling dynamical system. We shall see that this amounts to solving a (sparse) system of linear inequalities, and shall describe an interesting iterative procedure to accomplish this, which has the distinguished feature of being able to always produce an "optimal" solution (in the sense of Chebyshev) when no exact solutions exist. It will also be seen how the proposed learning approach allows us to develop a powerful asymmetric associative memory model based on the dynamical properties of the relaxation labeling process. The validity of the model is confirmed experimentally.

We mention that essentially the same idea to derive the compatibility coefficients of relaxation labeling was succinctly suggested by Hummel [12]. As a matter of fact, he did not suggest any practical way to accomplish this, and did not apparently recognize the potential applications of the model.

2 Relaxation Labeling Processes

Relaxation labeling processes were developed to solve the so-called (continuous) labeling problem, where one has to assign labels to objects so as to satisfy a set of domain-specific constraints. Let $\mathbf{B} = \{b_1, \cdots, b_n\}$ and $\mathbf{\Lambda} = \{\lambda_1, \cdots, \lambda_m\}$ denote respectively the set of objects and the set of labels of the problem at hand. Moreover, let the constraints be quantitatively expressed in terms of a four-dimensional matrix of real-valued compatibility coefficients $R = \{r_{ij}(\lambda, \mu)\}$: the component $r_{ij}(\lambda, \mu)$ measures the strength of compatibility between the two hypotheses "λ is on object b_i" and "μ is on object b_j." High values mean compatibility while low values mean incompatibility.

Let $p_i(\lambda)$ represent the degree of confidence of the hypothesis "label λ is on object b_i." It is assumed that $p_i(\lambda) \geq 0$ and $\sum_\lambda p_i(\lambda) = 1$, so that the m-dimensional vector $\bar{p}_i = (p_1(\lambda_1), \ldots, p_n(\lambda_m))^T$ can be considered as the probability distribution of labels for object b_i. By putting together the $\bar{p}_i$'s we obtain a weighted labeling assignment for the objects of $\mathbf{B}$ that will be denoted by $\bar{p}$, and will be conveniently considered as an $n \times m$ matrix. We will find it useful to introduce the space of weighted labeling assignments:

$$\mathcal{K} = \left\{ \bar{p} \in \Re^{nm} : p_i(\lambda) \geq 0 \text{ and } \sum_\lambda p_i(\lambda) = 1 \right\}$$

which is a linear convex set of $\Re^{nm}$. Every vertex of $\mathcal{K}$ represents an *unambiguous* labeling assignment which assigns exactly one label to each object. The set of these labelings will be denoted by $\mathcal{K}^*$.

Hummel and Zucker [11] developed a general theory of consistency for the labeling problem which is the basis of the work reported here. The entire development of the theory is basically a generalization of the notion of consistency for unambiguous labelings which is more easily understood. Consider a labeling $\bar{p} \in \mathcal{K}$. The degree of agreement between the hypothesis that b_i is labeled with label λ and the context can be quantified by a linear *support* function

$$q_i(\lambda; \bar{p}) = \sum_j \sum_\mu r_{ij}(\lambda, \mu) p_j(\mu) \ . \qquad (1)$$

Now, let $\bar{p} \in \mathcal{K}^*$ be an unambiguous labeling assignment, and let $\lambda(i)$ denote the label assigned to b_i by $\bar{p}$ (i.e., $p_i(\lambda(i)) = 1$). It seems reasonable to say that $\bar{p}$ is consistent if and only if the assigned label of each object receives the greatest support at that object. This corresponds to having

$$q_i(\lambda; \bar{p}) \leq q_i(\lambda(i); \bar{p}) \qquad (2)$$

for all i and λ or, equivalently, $\sum_\lambda v_i(\lambda) q_i(\lambda; \bar{p}) \leq \sum_\lambda p_i(\lambda) q_i(\lambda; \bar{p})$, for all $\bar{v} \in \mathcal{K}^*$.

By analogy, a weighted labeling assignment $\bar{p} \in \mathcal{K}$ is said to be *consistent* provided that

$$\sum_\lambda v_i(\lambda) q_i(\lambda; \bar{p}) \leq \sum_\lambda p_i(\lambda) q_i(\lambda; \bar{p}) \qquad (3)$$

for all $i = 1 \ldots n$ and $\bar{v} \in \mathcal{K}$. Furthermore, if strict inequalities hold in (3), for all $\bar{v} \neq \bar{p}$, then $\bar{p}$ is said to be *strictly consistent*. It can be easily shown that for unambiguous labelings conditions (2) and (3) are equivalent [11]. After defining the notion of consistency and proving some useful characterizations, Hummel and Zucker showed that when the compatibility matrix happens to be symmetric (i.e., $r_{ij}(\lambda, \mu) = r_{ji}(\mu, \lambda)$), then a sufficient condition for a labeling $\bar{p}$ to be consistent is that it is a local minimum of the following "energy" function which is a measure of labeling's (in)consistency

$$A(\bar{p}) = - \sum_{i,\lambda} \sum_{j,\mu} r_{ij}(\lambda, \mu) p_i(\lambda) p_j(\mu) \ . \qquad (4)$$

A relaxation labeling process takes as input an initial labeling assignment $\bar{p}^{(0)} \in \mathcal{K}$ and iteratively adjusts it taking into account the compatibility model using an updating formula of the form $\bar{p}^{(t+1)} = f(\bar{p}^{(t)}, \bar{q}^{(t)})$, where t is the time index. The process evolves until (at least in theory) a fixed point is reached, which means that $\bar{p}^{(t+1)} = \bar{p}^{(t)}$. In practice, it is customary to stop the process when the distance between two successive labelings becomes negligible, or after a predetermined number of iterations.

The most popular form for the function f, which was also used in the experiments reported later in this work, is as follows:

$$p_i^{(t+1)}(\lambda) = \frac{p_i^{(t)}(\lambda) q_i^{(t)}(\lambda)}{\sum_\mu p_i^{(t)}(\mu) q_i^{(t)}(\mu)} \tag{5}$$

provided that the compatibility coefficients are nonnegative. This corresponds to the original nonlinear scheme developed heuristically by Rosenfeld *et al.* [1].

Recently it has been shown that despite its completely heuristic derivation, the original relaxation scheme (5) possesses a number of interesting properties [13]. Firstly, when the compatibility matrix R is symmetric then A turns out to be a (strict) Liapunov function for the process, which means that it is monotonically decreasing along nonconstant trajectories. Secondly, and even more interestingly, it can be proven that strictly consistent labelings act as local attractors for the dynamical system defined by (5), whether or not the matrix R happens to be symmetric. This means that when started sufficiently close to a strictly consistent labeling $\bar{e}$, the relaxation process will tend to $\bar{e}$ as time increases. This property is formalized in the following theorem, which was originally proven by Elfving and Eklundh [14] in a slightly simplified form (see [13] for proof).

Theorem 1 *Let $\bar{e} \in \mathcal{K}^*$ be a strictly consistent labeling. Then $\bar{e}$ is an asymptotically stable equilibrium point (and hence a local attractor) for the relaxation labeling scheme defined in formula (5).*

This is the analog to the fundamental local convergence result that Hummel and Zucker proved for a different relaxation algorithm (which makes use of a computationally expensive projection operator) that turns out to be an approximation of the standard Rosenfeld *et al.*'s scheme [11, Theorem 9.1]. Its interestingness stems essentially from the fact that no restriction on the structure of the compatibility matrix is imposed. This result is the basis for the autoassociative learning approach described in this paper.

3 Problem Formulation

The learning approach described in this paper is based on the assumption that we have access to a set of learning patterns

$$\Xi = \left\{ \xi^{(1)}, \cdots, \xi^{(P)} \right\} \tag{6}$$

where each $\xi^{(\gamma)}$, $\gamma = 1 \dots P$, is simply an ordered list of labels, i.e.,

$$\xi^{(\gamma)} = \xi_1^{(\gamma)} \xi_2^{(\gamma)} \cdots \xi_n^{(\gamma)} \tag{7}$$

with $\xi_i^{(\gamma)} \in \mathbf{\Lambda}$, for all $i = 1 \dots n$. For each learning pattern $\xi^{(\gamma)}$, consider the corresponding unambiguous labeling assignment $\bar{p}(\xi^{(\gamma)}) \in \mathcal{K}^* \subset \Re^{nm}$, which is just an alternative (but equivalent) representation of $\xi^{(\gamma)}$: $p_i(\lambda; \xi^{(\gamma)}) = 1$, if $\lambda = \xi_i^{(\gamma)}$, and $p_i(\lambda; \xi^{(\gamma)}) = 0$ otherwise.

The key idea behind the proposed learning approach follows from the observation that since the learning patterns *are* instances of the problem we intend to solve, they must be "consistent" with the problem's constraints. This naturally suggests the following strategy: find a compatibility matrix R so that the learning patterns become consistent (in the following, we will find it useful to "linearize" the compatibility matrix and consider it as an n^2m^2-dimensional column vector $\bar{r}$). Indeed we want the learning patterns to be strictly consistent. In fact, from the preceding section (Theorem 1) we know that if started in the vicinity of one of them, the relaxation process will eventually converge toward it.

For the labeling $\bar{p}(\xi^{(\gamma)})$, $\gamma = 1 \dots P$, to be strictly consistent the following relation must hold

$$q_i(\lambda; \bar{p}(\xi^{(\gamma)})) < q_i(\xi_i^{(\gamma)}; \bar{p}(\xi^{(\gamma)})) \tag{8}$$

for all $\lambda \neq \xi_i^{(\gamma)}$ and $i = 1 \dots n$. Simple algebraic manipulations yield

$$\sum_j r_{ij}(\lambda, \xi_j^{(\gamma)}) - \sum_j r_{ij}(\xi_i^{(\gamma)}, \xi_j^{(\gamma)}) < 0 \tag{9}$$

which is a system of $Pn(m-1)$ linear inequalities in the n^2m^2 unknowns $\{r_{ij}(\lambda, \mu)\}$.

System (9) can be compactly represented as

$$C\bar{r} < \bar{0} \tag{10}$$

where C is the $Pn(m-1) \times n^2m^2$ matrix defined as

$$C(\gamma, i, \lambda; j, k, \mu, \eta) = \begin{cases} +1\ , & \text{if } j = i, \mu = \lambda, \eta = \xi_k^{(\gamma)} \\ -1\ , & \text{if } j = i, \mu = \xi_j^{(\gamma)}, \eta = \xi_k^{(\gamma)} \\ 0\ , & \text{otherwise} \end{cases}$$

(for notational convenience we use a three-component index for the rows and a four-component index for the columns), $\bar{r}$ is the unknown compatibility vector, and $\bar{0}$ is the null vector.

In practical applications, it is customary when solving systems of linear inequalities to introduce a "margin," which also ensures larger basins of attraction [15]. Accordingly, our system is rewritten as

$$\sum_j r_{ij}(\lambda, \xi_j^{(\gamma)}) - \sum_j r_{ij}(\xi_i^{(\gamma)}, \xi_j^{(\gamma)}) \leq \kappa \qquad (11)$$

or, equivalently,

$$C\bar{r} \leq \kappa\bar{1} \qquad (12)$$

where κ is some predetermined negative constant (the margin), and $\bar{1}$ is the unity vector.

4 Solving the System: The Eremin Algorithm

Many algorithms have been developed for solving systems of linear inequalities. The most immediate approach (also suggested in [12]) is to make use of the well-known simplex algorithm. Another popular method is the *relaxation* method developed by Agmon [16] (not to be confused with our relaxation labeling process). A common characteristic of these methods, however, is that they require the system to be *compatible*, i.e., to admit at least a solution. Alternative procedures, like the one developed by Ho and Kashyap [17] are able to automatically determine the solution of a system if there is one, and just to report that no solutions exist when this is not the case.

Even if in certain restricted circumstances our system can be proven to have a solution, in general this is certainly not the case. All these methods are therefore inappropriate in our case, for when the system is incompatible we would like at least to get a "good" approximation solution of it. Fortunately, this kind of algorithm is available. This was proposed in a rather obscure paper by Russian mathematician Eremin in the early 1960s [18]. The attractive feature of Eremin's method (which is a simple variant of Agmon's relaxation algorithm) is that it *does* solve the system when it happens to be compatible, and yields the "best" approximation solution (in the sense of Chebyshev) when this is not the case. The major point is that it does not require any *a-priori* knowledge about the system's compatibility, being able to handle the two cases in a completely automatic manner.

Eremin's method applies to systems of the general form $\sum_i a_{ji}x_i \leq b_j$. In our description, however, we will tailor it to our problem (11), or its equivalent formulation (12). Let $\bar{r} \in \Re^{n^2m^2}$ be an arbitrary compatibility vector, and put

$$\Delta(\bar{r}) = \max_{\gamma,i,\lambda} \left\{ \sum_j r_{ij}(\lambda, \xi_j^{(\gamma)}) - \sum_j r_{ij}(\xi_i^{(\gamma)}, \xi_j^{(\gamma)}) - \kappa \right\}.$$

Moreover, put

$$d(\bar{r}) = \begin{cases} \Delta(\bar{r})\ , & \text{if } \Delta(\bar{r}) \geq 0 \\ 0\ , & \text{otherwise} \end{cases}$$

which is a continuous and convex function of $\bar{r}$. Eremin called the number $\epsilon_0 = \min_{\bar{r}} d(\bar{r})$ the *defect* of system (11), which can be interpreted as the smallest ϵ for which the system $\sum_j r_{ij}(\lambda, \xi_j^{(\gamma)}) - \sum_j r_{ij}(\xi_i^{(\gamma)}, \xi_j^{(\gamma)}) - \kappa \leq \epsilon$ is compatible. Clearly if $\epsilon_0 = 0$ then the original system (11) is compatible. When the system is incompatible (i.e., $\epsilon_0 > 0$) the number ϵ_0 is called the Chebyshev deviation of the system and any point $\bar{r}$ satisfying the system

$$\sum_j r_{ij}(\lambda, \xi_j^{(\gamma)}) - \sum_j r_{ij}(\xi_i^{(\gamma)}, \xi_j^{(\gamma)}) - \kappa \leq \epsilon_0 \qquad (13)$$

will be called the point of its Chebyshev approximation. Furthermore, a sequence of points $\bar{r}_0, \bar{r}_1, \bar{r}_2, \ldots \in \Re^{n^2m^2}$ will be called *solving* for the system (11) if it converges to some solution of the system (13).

Now, let $\{\lambda_k\}$ be a sequence of positive numbers such that $\lambda_k \to 0$ as $k \to \infty$, and $\sum_k \lambda_k = +\infty$ (e.g., $\lambda_k = 1/(k+1)$), and let $\bar{r}_0$ be an arbitrary initial point. Starting from $\bar{r}_0$ Eremin's algorithm produces a sequence of points $\{\bar{r}_k\}$ according to the following scheme

$$\bar{r}_{k+1} = \bar{r}_k - \lambda_{k+1} d(\bar{r}_k) C(\gamma_k, i_k, \lambda_k;\ :)^T \qquad (14)$$

where $C(\gamma_k, i_k, \lambda_k;\ :)$ is the row of the coefficient matrix C corresponding to the equation for which the value $d(\bar{r}_k)$ is attained (if more than one such equations exist, that with the least index is taken). Eremin [18] proved the following result.

Theorem 2 (Eremin, 1962) *The sequence $\{\bar{r}_k\}$ defined in equation (14) is solving for the system (11).*

In other words, the preceding theorem states that if the system (11) happens to be compatible, then the sequence $\{\bar{r}_k\}$ will converge toward one of its solutions. Otherwise, it will converge toward a point of its Chebyshev approximation.

To conclude this section we need to clarify a final point. If we are to use the relaxation labeling algorithm defined in formula (5), then we need the compatibility coefficients to be nonnegative. Note, however, that there is no guarantee that the solution provided by the Eremin procedure will satisfy this constraint. A first possible approach to solve this problem is to add a set of "feasibility" constraints to our system (11). In our case, however, it is more convenient to initially derive arbitrary compatibility coefficients and then to scale them linearly so as to make them nonnegative (i.e., by simply adding the smallest negative coefficient). It can be readily seen that this has no effect on the structure of the space of consistent labelings [13].

5 An Application: Building an Asymmetric Associative Memory

The autoassociative learning model discussed in this paper naturally leads us to view the relaxation labeling process as a novel kind of multi-valued asymmetric associative memory. This consists of an $n \times m$ densely interconnected relaxation labeling network, where n is the word's length and m is the number of possible values at each site. The connection strengths between units are determined by the compatibility matrix R, i.e., $r_{ij}(\lambda, \mu)$ is the weight on the connection between units (i, λ) and (j, μ). The unit indexed (i, λ) updates its own state according to formula (5), and its activation value $p_i(\lambda)$ can therefore be thought of as the probability that λ be the correct value for word's site i. From Theorem 1, we know that if started in the vicinity of a strictly consistent labeling, the network will eventually converge toward it, whether or not the connection strengths are symmetric. If the resulting labeling corresponds to a memory pattern, we say that the memory has "recalled" that pattern. The learning algorithm described in the previous section can therefore naturally be employed as a means to "store" a given set of patterns in the memory.

To assess the validity of this memory model some experiments were conducted aimed at testing its error-correction capabilities as well as evaluating its storage capacity. We concentrated on binary memories, i.e., $\mathbf{\Lambda} = \{0, 1\}$. Note that, contrary to tradition, in our model each memory site has two computational units associated with it, namely one for the value "0" and the other for the "1." It is interesting to observe that this kind of representation, which involves "opposing pairs," is indeed the one preferred by many biological perceptual systems. Retinal ganglion cells with on- and off-center receptive fields are perhaps the most common example, but examples from other sensory systems abound [19].

The learning set used in this study was taken from a recent paper by Zhuang *et al.* [20] and consists of ten binary digits on a 5×5 matrix, as shown in Fig. 1. The training was carried out using the Eremin procedure described in the previous section, using different margin values $\kappa = -10, -20, -30, -50, -100$.

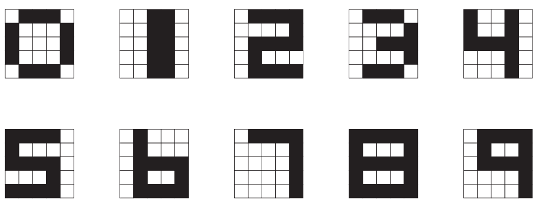

Figure 1. Training set used in the experiments.

To experimentally test that the stored patterns had finite attraction basins, ten noisy versions of the corresponding unambiguous labelings were generated for each of them, using a continuous Gaussian noise with mean 0 and variance 0.1 (this was followed by a successive normalization step to ensure that the noisy labelings still belonged to $\mathcal{K}$). These 100 labelings were later given as input to the relaxation network which in the 100% of the cases quickly recalled the original ones.

Later, the error-correction capability of the model was tested. For each memory pattern, ten perturbed versions were generated by randomly flipping exactly d bits (for varying d), and the corresponding unambiguous labelings were obtained. In addition, since unambiguous labelings turn out to be fixed points for the relaxation scheme (5), a further Gaussian noise was inflicted as described before. The relaxation network was then allowed to run for (at most) 1,000 iterations, and the resulting (weighted) labelings were converted into binary patterns by a simple maxima selection criterion. The resulting binary patterns were then compared with the original ones, and a *success* was recorded when a perfect match was achieved. Fig. 2 shows the percentage of successful recalls as a function of d, for various values of the margin κ. As can be seen, there is no significant difference in performance, but it seems that there is no advantage in going beyond a certain limit which, in our case, may be taken as $\kappa = -50$. Our results compare favorably with those obtained on the same training set by Zhuang *et al.* [20], who developed a sophisticated learning algorithm for Hopfield memories (*cfr.* their Fig. 4(a)). Their results, in turn, were by far superior to those obtained using the standard Hebb rule [21].

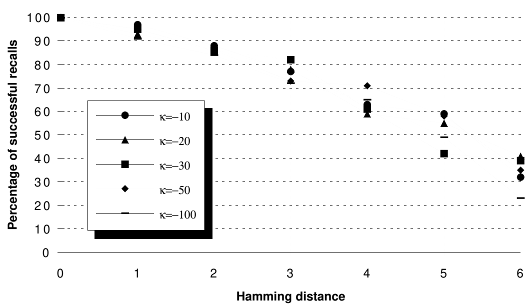

Figure 2. Error-correction performance as a function of the Hamming distance, for various values of the margin κ.

Our next goal was to estimate the storage capacity of the proposed memory. To do so, the following experiment was conducted. We fixed the word's length at $n = 10$ and tried to store s desired memories, for increasing values of s. In this case, the margin value $\kappa = -50$ was used. For each value of s, 100 such tests were performed each involving exactly s randomly generated patterns. After the learning phase, we tested if the desired memories were made strictly consistent by the learning algorithm. As seen, this is equivalent to saying that the patterns were stored into the memory. A test was considered successful if the percentage of stored memories was greater than or equal to 90%. We then calculated the percentage of successful tests for each value of s, and Fig. 3 shows the results obtained.

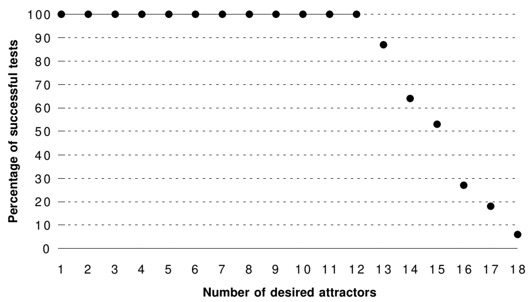

Figure 3. Storage capacity of the relaxation labeling associative memory.

From this preliminary study we can say that the capacity of our model is approximately $1.2n$. This figure is by far superior to the capacity of Hebb-trained associative memories, which is $0.15n$ [21], and is also substantially larger than the estimated capacity of the powerful model developed by Zhuang *et al.* in [20], which was $0.8n$ for $n = 10$.[1] In order to fully assess the storage capacity of the model, however, further experiments for larger values of n are needed.

6 Conclusions

In this paper, we have proposed a new, autoassociative algorithm to train relaxation labeling processes. The approach is based on a formal theory of consistency developed in a landmark paper by Hummel and Zucker [11]. After introducing the basic relaxation labeling formulas and discussing their fundamental dynamical properties, we have formulated the autoassociative learning problem as one of solving a system of linear inequalities. Among the many existing algorithms for solving such systems we have chosen a relaxation-style procedure developed by Russian mathematician Eremin. This has the attractive feature of being able to automatically yield the best approximation solution when no exact solution exist. The proposed learning algorithm has been tested over an associative memory application, where memory patterns are stored as attractive fixed points of the relaxation labeling dynamical system. The preliminary results obtained indicate that our model exhibits error-correction performance that are competitive with sophisticated Hopfield-style memories, and turns out to have a substantially higher storage capacity.

[1] In their experiments, however, they performed 1,000 tests for each value of s.

References

[1] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, “Scene labeling by relaxation operations,” *IEEE Trans. Syst. Man Cybern.*, vol. 6, pp. 420–433, 1976.

[2] J. Kittler and J. Illingworth, “Relaxation labeling algorithms—A review,” *Image Vision Comput.*, vol. 3, pp. 206–216, 1985.

[3] D. H. Ballard, G. E. Hinton, and T. J. Sejnowski, “Parallel visual computation,” *Nature*, vol. 306, pp. 21–26, 1983.

[4] S. Peleg and A. Rosenfeld, “Determining compatibility coefficients for curve enhancement relaxation processes,” *IEEE Trans. Syst. Man Cybern*, vol. 8, pp. 548–555, 1978.

[5] S. Peleg, “A new probabilistic relaxation scheme,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 2, pp. 362–369, 1980.

[6] E. R. Hancock and J. Kittler, “Combining evidence in probabilistic relaxation,” *Int. J. Pattern Recognition Artificial Intell.*, vol. 3, pp. 29–51, 1989.

[7] M. Pelillo and M. Refice, “Learning compatibility coefficients for relaxation labeling processes,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 933–945, 1994.

[8] S. W. Zucker, A. Dobbins, and L. Iverson, “Two stages of curve detection suggest two styles of visual computation,” *Neural Computation*, vol. 1, pp. 68–81, 1989.

[9] M. Pelillo, F. Abbattista, and A. Maffione, “An evolutionary approach to training relaxation labeling processes,” *Pattern Recognition Lett.*, vol. 16, pp. 1069–1078, 1995.

[10] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, CA, 1991.

[11] R. A. Hummel and S. W. Zucker, “On the foundations of relaxation labeling processes,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, pp. 267–287, 1983.

[12] R. A. Hummel, “A design method for relaxation labeling applications,” in *Proc. Natl. Conf. Artificial Intell. (AAAI-83)*, Washington, DC, 1983, pp. 168–171.

[13] M. Pelillo, “The dynamics of nonlinear relaxation labeling processes,” *J. Math. Imaging Vision*, to appear.

[14] T. Elfving and J.-O. Eklundh, “Some properties of stochastic labeling procedures,” *Comput. Graph. Image Processing*, vol. 20, pp. 158–170, 1982.

[15] B. M. Forrest, “Content addressability and learning in neural networks,” *J. Phys. A*, vol. 21, pp. 245–255, 1988.

[16] S. Agmon, “The relaxation method for linear inequalities,” *Canad. J. Math*, vol. 6, pp. 382–392, 1954.

[17] Y.-C. Ho and R. L. Kashyap, “An algorithm for linear inequalities and its applications,” *IEEE Trans. Elect. Comp.*, vol. 14, pp. 683–688, 1965.

[18] I. I. Eremin, “Iteration method for Chebyshev approximations for sets of incompatible linear inequalities,” *Soviet Math. Doklady*, vol. 3, pp. 570–572, 1962.

[19] D. H. Hubel, *Eye, Brain, and Vision*. Scientific American Library, New York, 1988.

[20] X. Zhuang, Y. Huang, and F. A. Yu, “Design of Hopfield content-addressable memories,” *IEEE Trans. Signal Processing*, vol. 42, pp. 492–495, 1994.

[21] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proc. Natl. Acad. Sci. USA*, vol. 79, pp. 2554–2558, 1982.