Neural Processing Letters **3:** 113–121, 1996.

A Relaxation Algorithm for Estimating the Domain of Validity of Feedforward Neural Networks

MARCELLO PELILLO
Dipartimento di Matematica Applicata e Informatica, Università 'Ca' Foscari' di Venezia, Via Torino 155, 30173 Venezia Mestre, Italy
E-mail: pelillo@dsi.unive.it

Key words: convex hull, generalization, neural networks, relaxation

Abstract. We expand on a recent paper by Courrieu which introduces three algorithms for determining the distance between any point and the interpolation domain associated with a feedforward neural network. This has been shown to have a significant relation with the network's generalization capability. A further neural-like relaxation algorithm is presented here, which is proven to naturally solve the problem originally posed by Courrieu. The algorithm is based on a powerful result developed in the context of Markov chain theory, and turns out to be a special case of a more general relaxation model which has long become a standard technique in the machine vision domain. Some experiments are presented which confirm the validity of the proposed approach.

1. Introduction

One of the ultimate criteria for judging the quality of a given neural network problem solution is its generalization ability, that is, how well will the network perform when presented with patterns never seen during learning? In a recent paper, Courrieu [6] attempted to provide an answer to this question. He demonstrated how the generalization performance of a feedforward neural network depends significantly on the location of the generalization patterns with respect to the network's domain of validity, which corresponds to the convex hull of the set of learning points. He therefore posed the problem of calculating the distance between an arbitrary point and a given convex polytope, and developed three simple algorithms to accomplish this. The first is a conventional gradient descent procedure, the second is a four-layer recurrent neural network which essentially approximates the first, and the third makes use of a circumscribed sphere to approximate the polytope.

In this paper, a further neural-like algorithm for solving the problem originally posed by Courrieu [6] is presented. Based on a powerful result of use in the theory of Markov processes, the proposed network model is proven to have an energy function which rules its dynamical behavior and drives the system towards low-energy configurations. This property is therefore exploited to make the network solve Courrieu's problem in a completely natural fashion. Interestingly enough, the algorithm proposed here turns out to be but a special instance of a more general class

of parallel distributed models, generally known as *relaxation labeling processes*, which were heuristically introduced by Rosenfeld et al. [21] for solving certain constraint satisfaction problems arising in vision. Since then the algorithm has been successfully employed in a variety of difficult tasks in pattern recognition and computer vision, and is still attracting the interest of many investigators (see, e.g., [11]). Despite its heuristic derivation, the algorithm has been recently shown to possess interesting dynamical properties [17] and learning capabilities [19], and turns out to be closely related to certain mechanisms in the early stages of the human visual system [2, 23]. It may be also of some interest to point out that the proposed dynamical scheme was independently derived and studied as a model of evolution in population genetics [8] and, based on such ideas, a similar strategy was more recently used to solve certain combinatorial optimization problems [14].

The rest of the paper is organized as follows: Section 2 formally states the problem we intend to solve in terms of minimizing a convex functional over a certain polytope in Euclidean space. Section 3 presents the proposed neural network model, proves some interesting properties, and discusses how to configure the network so as to solve the problem. Some experimental results are presented in Section 4 which illustrate the effectiveness of the proposed approach.

2. Problem Formulation

Let $G = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ be a finite set of points in the Euclidean space $\Re^n$, and let conv(G) denote the convex hull of G, that is the smallest convex set containing G. Let $\mathcal{K}_m$ denote the following polytope in $\Re^m$ (see Figure 1):

$$\mathcal{K}_m = \left\{ \lambda \in \Re^m \ : \ \lambda_i \geq 0, \text{ all } i = 1 \ldots m, \quad \text{and} \quad \sum_{i=1}^{m} \lambda_i = 1 \right\},$$

and consider the $n \times m$ real matrix defined as $X = [\mathbf{x}_1 \ \ \mathbf{x}_2 \ \ \cdots \ \ \mathbf{x}_m]$. It is well known that conv(G) can be written as

$$\text{conv}(G) = \{ \mathbf{v} \in \Re^n \ : \ \mathbf{v} = X\lambda \quad \text{for some} \quad \lambda \in \mathcal{K}_m \}.$$

Courrieu [6] called the following measure

$$E(\mathbf{y}, G) = \min_{\lambda \in \mathcal{K}_m} \|X\lambda - \mathbf{y}\|_2 \tag{1}$$

the *exteriority* of $\mathbf{y}$ to conv(G), which is nothing but the Euclidean distance between $\mathbf{y}$ and its closest point in conv(G). He experimentally demonstrated how the exteriority measure can provide useful information about the ability of neural networks to generalize well. Specifically, the generalization error was shown to become higher as the exteriority of the generalization points increases; on the other hand, low exteriority values do not necessarily imply that the network will respond

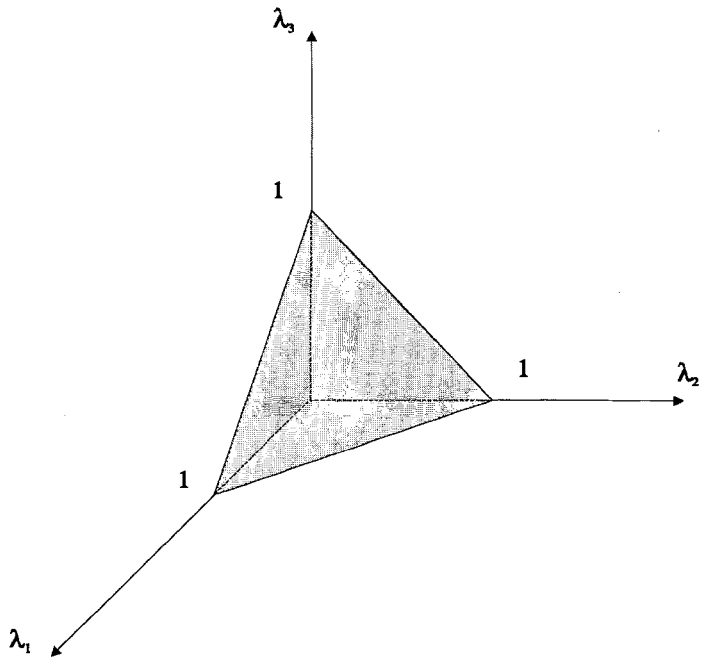

Figure 1. The polytope $\mathcal{K}_3$.

correctly, but in this case the generalization error is typically smaller than for larger exteriority values. In addition, Courrieu proposed a straightforward procedure for extracting the vertices of conv(G), which completely (and more economically) characterize the polytope. It is based on the observation that a point $\mathbf{x} \in G$ is a vertex of conv(G) if and only if it has a nonzero exteriority to conv($G - \{\mathbf{x}\}$).

For convenience, the problem of evaluating $E(\mathbf{y}, G)$ is translated into the equivalent (but more manageable) constrained quadratic programming problem

$$\begin{aligned} &\text{minimize } C(\lambda) = \tfrac{1}{2}\|X\lambda - \mathbf{y}\|_2^2 \\ &\text{subject to } \lambda \in \mathcal{K}_m. \end{aligned}$$

It is a well-known fact that the functional C is convex (strictly convex indeed if the vectors $\mathbf{x}_1, \cdots, \mathbf{x}_m$ happen to be linearly independent), and this implies that all local minima of C are also global minima. Any descent procedure is therefore guaranteed to approach the global optimal solution in this case, without the risk of becoming trapped into poor local minima.

It is interesting to note that a similar optimization problem, known as the problem of 'optimal stability', also arises in the context of learning in perceptron networks; there the goal is to derive the network's weights so as to ensure larger basis of attraction [1, 12, 22]. Moreover, our problem turns out to be closely related

to that of determining whether a given set of prototype vectors can be stored in a Hopfield-style associative memory [10, Theorem 6.1].

3. Estimating Exteriority by Relaxation

The proposed neural network model for calculating the exteriority of a point $\mathbf{y}$ to a given convex hull polytope conv(G) consists of m pairwise interconnected computational units, one for each point in G. Let w_{ij} be the strength of the connection from unit i to unit j, and let s_i denote an external input signal associated with unit i. It is assumed that both the weights and the external signals are nonpositive, i.e., $w_{ij} \leq 0$ and $s_i \leq 0$, for all $i, j = 1 \ldots m$.[1] Note that unipolar networks, like the one we are proposing here, turn out to be advantageous in many applications, especially when hardware implementation is a concern [7]. In the following discussion, W will denote the $m \times m$ real-valued nonpositive matrix having w_{ij} as its (i, j) entry, and $\mathbf{s}$ will represent the m-dimensional nonpositive vector of the external signals.

Let $\sigma_i(t)$ represent the state of unit i at time t, and define the state of the network as a whole at time t to be the vector $\sigma(t) = (\sigma_1(t), \cdots, \sigma_m(t))^T$, where 'T' denotes transposition. The system works as follows. It starts out with an initial state vector $\sigma(0) \in \mathcal{K}_m$ and iteratively and synchronously updates its own state according to the following dynamical equation

$$\sigma_i(t+1) = \frac{\sigma_i(t) q_i(t)}{\sum_{j=1}^{m} \sigma_j(t) q_j(t)}, \quad i = 1 \ldots m \tag{2}$$

where

$$q_i(t) = \sum_{j=1}^{m} w_{ij} \sigma_j(t) + s_i, \quad i = 1 \ldots m \tag{3}$$

is the net input to unit i at time t. The process evolves until a fixed point is reached, i.e., until $\sigma(t+1) = \sigma(t)$.

Because of the normalization factor present in Equation (2) and the unipolarity condition, the network performs essentially a mapping of the domain $\mathcal{K}_m$ onto itself, provided that $\sigma(0) \in \mathcal{K}_m$. Levinson et al. [13], in a rather different context, offered a simple geometrical interpretation for transformations like (2). Let λ be a point in $\mathcal{K}_m$, and let $\mathbf{q}$ denote the m-vector composed of the q_i's, as defined in (3). Moreover, let $\mathbf{z}$ be the m-vector whose ith component is given by the component-wise product between λ and $\mathbf{q}$, i.e., $z_i = \lambda_i q_i$. Then, it is readily seen that the vector obtained by applying the transformation (2) to λ is simply the intersection of the vector $\mathbf{z}$ – or its extension – with the hyperplane defined by $\sum_{i=1}^{m} \lambda_i - 1 = 0$. As an aside, we note that output normalization has now become

a common practice within the neural network community; this can be regarded as a multi-input generalization of the more familiar logistic nonlinearity, and can be easily implemented into physical circuitry [5]. It can also be considered as a form of 'soft' competition among hypotheses, an approach that contrasts with the classical winner-take-all view [15].

In a very interesting paper, Baum and Eagon [3] proved a powerful theorem which turns out to be the basis of the work reported in this paper. Here, we present Baum and Eagon's result in a slightly different and simplified form.

Theorem 1 (Baum–Eagon) *Let $P(\lambda)$ be a polynomial in the variables $\{\lambda_i\}$ with nonpositive coefficients, and let λ be a point of the domain $\mathcal{K}_m$. Define the mapping $\mu = \mathcal{M}(\lambda)$ as*

$$\mu_i = \frac{\lambda_i \frac{\partial P(\lambda)}{\partial \lambda_i}}{\sum_{j=1}^{m} \lambda_j \frac{\partial P(\lambda)}{\partial \lambda_j}}, \qquad i = 1 \ldots m . \tag{4}$$

Then $P(\mathcal{M}(\lambda)) < P(\lambda)$, unless $\mathcal{M}(\lambda) = \lambda$.

Indeed, Baum and Eagon's result was originally proven for the special case of homogeneous polynomials. In a subsequent paper, however, Baum and Sell [4] extended the original theorem to nonhomogeneous polynomials, and proved that the inequality still holds for all points lying on the segment connecting λ and $\mathcal{M}(\lambda)$. They also provided an analysis of the asymptotic behavior of the transformation $\mathcal{M}$ in the vicinity of local extrema. As noted by Baum and Sell [4], the mapping defined previously makes use of first derivatives only and yet is able to make finite steps while decreasing P. This contrasts sharply with conventional gradient methods, for which a decrease in the objective function is guaranteed only when infinitesimal steps are taken, and determining the optimal step size entails computing higher-order derivatives. The Baum–Eagon inequality provides an effective iterative means for optimizing polynomial functions over a domain of probability values and, in fact, it has served as the basis for many statistical estimation procedures. More recently, its usefulness in the field of speech recognition has been proven extensively [13].

Now, let us turn to our neural network model, and suppose that the weight matrix is symmetric ($w_{ij} = w_{ji}$). By simply applying the Baum–Eagon Theorem, we can assert that the network possesses the following strict Liapunov (or energy) function which is minimized in $\mathcal{K}_m$ as the process evolves:

$$L(\lambda) = \frac{1}{2}\lambda^T W \lambda + \mathbf{s}^T \lambda + \kappa , \tag{5}$$

where κ is an arbitrary constant (of either sign). Put another way, we have

$$L(\sigma(t+1)) < L(\sigma(t)) , \qquad \text{all } t \geq 0 \tag{6}$$

unless $\sigma(t+1) = \sigma(t)$. This property follows immediately from the fact that, when W is symmetric, we get

$$\frac{\partial L(\lambda)}{\partial \lambda_i} = \sum_{j=1}^{m} w_{ij}\lambda_j + s_i \,, \qquad i = 1 \ldots m \tag{7}$$

which means that the mapping performed by the network is identical to that defined in the Baum–Eagon Theorem.

Returning to our original problem, recall that calculating the exteriority of a point $\mathbf{y}$ to a given convex hull polytope conv(G) amounts to minimizing in $\mathcal{K}_m$ a quadratic polynomial which is explicitly written as

$$C(\lambda) = \frac{1}{2}\lambda^T X^T X \lambda - \mathbf{y}^T X \lambda + \frac{1}{2}\mathbf{y}^T \mathbf{y} \,. \tag{8}$$

In light of the above discussion, it is therefore easy to map the problem of estimating the exteriority measure onto a relaxation network of the type described above. To accomplish this, in fact, simply put

$$W = X^T X \tag{9}$$

and

$$\mathbf{s} = -X^T \mathbf{y} \,. \tag{10}$$

The network, starting from an initial state $\sigma(0)$, will iteratively minimize C and will eventually converge to a fixed point $\sigma^* \in \mathcal{K}_m$ which corresponds to a minimum of the cost function.[2] Owing to the convexity of C, σ^* will be also the global minimum of C, so that

$$E(\mathbf{y}, G) = \sqrt{2C(\sigma^*)} \tag{11}$$

irrespective of the starting point. However, since the process cannot leave the boundary of $\mathcal{K}_m$, it is preferable that the relaxation search begin with an interior point, i.e., $\sigma_i(0) > 0$ for all i. A reasonable choice, also adopted in the experiments reported in the next section, can be to start the process with $\sigma(0) = (1/m, 1/m, \cdots, 1/m)^T$ which corresponds to the center of $\mathcal{K}_m$.

As a final remark, observe that Equations (9) and (10) do not guarantee that W and $\mathbf{s}$ will contain nonpositive values, as required. Fortunately, this problem can be easily overcome by performing a simple linear scaling. Let $\hat{w}$ and $\hat{s}$ be the maximum positive values of W and $\mathbf{s}$, respectively, (put $\hat{w} = 0$ and $\hat{s} = 0$ if no such values exist) and construct the matrix W' as $w'_{ij} = w_{ij} - \hat{w}$, and the vector $\mathbf{s}'$ as $s'_i = s_i - \hat{s}$. Trivially, both W' and $\mathbf{s}'$ contain nonpositive values. Now, consider the polynomial $C'(\lambda) = \frac{1}{2}\lambda^T W' \lambda + \mathbf{s}'^T \lambda + \kappa$. By direct computation, it is simple

to see that $C'(\lambda) < C'(\mu)$ if and only if $C(\lambda) < C(\mu)$, for all $\lambda, \mu \in \mathcal{K}_m$. This means that if we have a descent procedure for C' in $\mathcal{K}_m$, then this will be also a descent procedure for C, and *vice versa*. Put another way, should W or $\mathbf{s}$ contain positive values, the network with weight matrix W' and external inputs $\mathbf{s}'$ not only will minimize C', but also the original cost C (and, indeed, all those functions which agree with C' in $\mathcal{K}_m$, up to a constant).

4. A Numerical Example

In order to assess the validity of the proposed algorithm in estimating the exteriority measure, some simulations were carried out over a simple toy problem. The task consisted of calculating the exteriority of an input point in the square $[-2,2] \times [-2,2]$ to the convex hull of the set $G = \{(-1,-1),(-1,1),(1,-1),(1,1)\}$ (which is the square $[-1,1] \times [-1,1]$). To accomplish this, a four-neuron relaxation network was constructed with weight matrix and external signals obtained as described before. Next, a thousand points were generated randomly in the square $[-2,2] \times [-2,2]$, and given as input to the network (via the external signals, as seen in the previous section). The process was allowed to iterate until the (squared) distance between two successive state vectors became smaller than $\epsilon = 10^{-7}$. A median number of 81 iterations were needed for the relaxation network to converge.

To evaluate the goodness of the solutions found by the relaxation process, the following quality measure introduced by Protzel (1990) was adopted:

$$Q = \frac{C_{ave} - C_{relax}}{C_{ave} - C_{opt}}\,, \tag{12}$$

where C_{ave} is the average cost function calculated by collecting a sufficient number of random points in $\mathcal{K}_m$, C_{relax} is the cost value of a given solution found by the network, and C_{opt} represents the global optimal value of the cost function, which is proportional to the square of the 'real' exteriority of the input point, determined in a separate calculation. In our problem, the true exteriority is readily calculated according to the following straightforward procedure:

$$E(\mathbf{y}, G) = \begin{cases} 0\,, & \text{if } |y_1| \leq 1 \text{ and } |y_2| \leq 1 \\ |y_1| - 1\,, & \text{if } |y_1| > 1 \text{ and } |y_2| \leq 1 \\ |y_2| - 1\,, & \text{if } |y_1| \leq 1 \text{ and } |y_2| > 1 \\ \sqrt{(|y_1| - 1)^2 + (|y_2| - 1)^2}\,, & \text{if } |y_1| > 1 \text{ and } |y_2| > 1 \end{cases}$$

where y_1 and y_2 represent the coordinates of the input point $\mathbf{y}$. Notice that, from the definition of Q, we have $Q = 0$ if $C_{relax} = C_{ave}$, and $Q = 1$ if $C_{relax} = C_{opt}$.

For each of the thousand relaxation runs, the quality measure Q was calculated and then averaged. The average value of Q was found to be 9.9998×10^{-1}, which clearly illustrates how the network is always able to find the globally optimal solution and can provide very accurate estimates of the exteriority measure. As

observed before, this is not surprising because of the convexity of the energy function.

5. Concluding Remarks

In this paper, a unipolar relaxation neural network has been presented which is able to estimate the distance between an arbitrary input point and a given convex polytope. As shown by Courrieu [6], this measure can be helpful in predicting the generalization performance of artificial neural networks. The validity of the proposed model has been demonstrated both theoretically and experimentally.

The neural network algorithm developed here exhibits a number of advantages over the neural-like counterpart developed by Courrieu, which essentially approximates a gradient procedure. First, in contrast with Courrieu's model which consists of four layers of highly-specialized units, ours has a much more simple and homogeneous architecture and therefore lends itself well to physical implementation. A second difference between the two models which is worth mentioning is that the one presented here does not make use of any working parameter. This is not true for Courrieu's algorithm, which needs a parameter that defines the size of the steps taken along gradient direction. As Courrieu himself admitted, the choice of this parameter poses some problems for too small a value slows down convergence, while too high a value can result in a divergent oscillation of the iterative process. A further nice feature of the proposed neural network is the existence of an energy function which monotonically decreases along network's trajectories. This makes the model far more general than presented here and suggests using it for solving arbitrary optimization problems, in exactly the same way as Hopfield and Tank [9] did with their popular neural algorithm. As a matter of fact, some experiments conducted recently with a similar (but more general) parallel relaxation algorithm have demonstrated the effectiveness of this kind of models in solving well-known intractable optimization problems [16, 18].

Acknowledgments

The author is grateful to Manfred Opper for pointing out the relations to population genetics and the optimal stability problem, and to Roberto Ligonzo for carrying out the experiments presented in the paper.

Notes

1. The nonnegative case can be treated in a completely analogous way and will be therefore ignored in what follows.
2. In practice, the process can be stopped when
$$\|\sigma(t+1) - \sigma(t)\|_2^2 < \epsilon$$
where ϵ is a small predetermined constant which affects the precision of the solution found.

References

1. J. K. Anlauf and M. Biehl, "The AdaTron: an adaptive perceptron algorithm", Europhys. Lett., Vol. 10, No. 7, pp. 687–692, 1989.
2. D. H. Ballard, G. E. Hinton and T. J. Sejnowski, "Parallel visual computation", Nature, Vol. 306, pp. 21–26, 1983.
3. L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology", Bull. Am. Math. Soc., Vol. 73, pp. 360–363, 1967.
4. L. E. Baum and G. R. Sell, "Growth transformations for functions on manifolds", Pacif. J. Math., Vol. 27, No. 2, pp. 211–227, 1968.
5. J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition", in F. Fogelman-Soulié and J. Hérault (eds) Neurocomputing: Algorithms, Architectures, and Complexity, pp. 227–236, Springer-Verlag: Berlin, 1990.
6. P. Courrieu, "Three algorithms for estimating the domain of validity of feedforward neural networks", Neural Networks, Vol. 7, No. 1, pp. 169–174, 1994.
7. J. S. Denker, "Neural network refinements and extensions", in J.S. Denker (ed) Neural Networks for Computing, pp. 121–128, American Institute of Physics: New York, 1986.
8. W. J. Ewens, Mathematical Population Genetics, Springer-Verlag: Berlin, 1979.
9. J. J. Hopfield and D. W. Tank, " 'Neural' computation of decisions in optimization problems", Biol. Cybern., Vol. 52, pp. 141–152, 1985.
10. Y. Kamp and M. Hasler, Recursive Neural Networks for Associative Memory, Wiley: New York, 1990.
11. J. Kittler and J. Illingworth, "Relaxation labeling algorithms – a review", Image Vision Comput., Vol. 3, pp. 206–216, 1985.
12. W. Krauth and M. Mézard, "Learning algorithms with optimal stability in neural networks", J. Phys. A, Vol. 20, pp. L745–L752, 1987.
13. S. E. Levinson, L. R. Rabiner and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition", Bell Syst. Tech. J., Vol. 62, No. 4, pp. 1035–1074, 1983.
14. H. Mühlenbein, M. Gorges-Schleuter and O. Krämer, "Evolution algorithms in combinatorial optimization", Parallel Computing, Vol. 7, pp. 65–85, 1988.
15. S. J. Nowlan, "Maximum likelihood competitive learning", in D.S. Touretzky (ed) Advances in Neural Information Processing Systems 2, pp. 574–582, Morgan Kaufmann: San Mateo, CA, 1990.
16. M. Pelillo, "Relaxation labeling processes for the traveling salesman problem", Proc. Int. J. Conf. Neural Networks (Nagoya, Japan), pp. 2429–2432, 1993.
17. M. Pelillo, "On the dynamics of relaxation labeling processes", Proc. IEEE Int. Conf. Neural Networks (Orlando, FL), pp. 1006–1011, 1994.
18. M. Pelillo, "Relaxation labeling networks for the maximum clique problem", J. Artif. Neural Networks (Special Issue on "Neural Networks for Optimization"), Vol. 2, No. 4, pp. 313–327, 1995.
19. M. Pelillo and M. Refice, "Learning compatibility coefficients for relaxation labeling processes", IEEE Trans. Pattern Anal. Machine Intell., Vol. 16, No. 9, pp. 933–945, 1994.
20. P. W. Protzel, "Comparative performance measure for neural networks solving optimization problems", Proc. Int. J. Conf. Neural Networks (Washington, DC), Vol. II, pp. 523–526, 1990.
21. A. Rosenfeld, R. A. Hummel and S. W. Zucker, "Scene labeling by relaxation operations", IEEE Trans. Syst. Man Cybern., Vol. 6, No. 6, pp. 420–433, 1976.
22. P. Ruján, "A fast method for calculating the perceptron with maximal stability", J. Phys. I France, Vol. 3, pp. 277–290, 1993.
23. 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.