

Terminal Attractor Algorithms and the Class of Unimodal Loading Problems

M. Bianchini*, S. Fanelli**, M. Gori*, and M. Protasi**

* *Dipartimento di Sistemi e Informatica, Università di Firenze*

Via di Santa Marta, 3 – 50139 Firenze – Italy

Tel. +39 (55) 479.6265 – Fax +39 (55) 479.6363

e-mail: marco{monica}@mcculloch.ing.unifi.it

www: <http://www.dsi.unifi.it/~marco>

** *Dipartimento di Matematica, Università di Roma “Tor Vergata”*

Via della Ricerca Scientifica – 00133 Roma – Italy

Tel. +39 (6) 7259.4681 – Fax +39 (6) 7259.4699

e-mail: fanelli{protasi}@mat.utovrm.it

Abstract

The effectiveness of connectionist models in emulating intelligent behaviour and solving significant practical problems is strictly related to the capability of the learning algorithms to find optimal or near-optimal solutions, and to generalise to new examples. This paper deals with optimal learning and provides a unified viewpoint of most significant results in the field.

We briefly review proposals for discovering optimal solutions and give some general guidelines for performing successful optimisation. Most importantly, we show some intriguing links between optimal learning and the computational complexity of loading problems. We prove that all problems giving rise to unimodal error functions have $\mathcal{O}(1)$ as a complexity upper bound, thus suggesting that they belong to the same class, defined on the basis of computational requirements.

1 Learning sub-optimal solutions

Supervised learning in multilayered networks can be accomplished thanks to Backpropagation (BP), which attempts to obtain interconnection weights that minimise pattern misclassifications. Though there are many variants of the basic scheme, they are all based on the minimisation of a cost function through the use of gradient descent for a particular nonlinear least squares fitting problem. Thus BP is subject to the local minima entrapment and indeed many examples have been found in which such minima occur. Nevertheless, in spite of this theoretical remark, it is generally claimed that even when these pathologies do occur, their domain of attraction is *small*, or they are not *true* minima [1], allowing network trained through BP to classify correctly. Unfortunately, neither of the above assertion is, in fact, generally correct, as examples in [2, 3] have shown. However, as for their relevance in most interesting practical applications, it should not be forgotten that the cited problems have a significantly different structure, typically due to the data redundancy. As pointed out in [3], “*It is entirely possible that “real” problems — as opposed to mathematically constructed ones — will not share these pathologies.*” Therefore, it becomes even more urgent to extensively characterize all the features which would cause *real* problems to be incorrectly faced by neural networks.

1.1 Spurious and structural local minima

There have been some efforts to understand the behaviour of BP in feedforward networks with no hidden layer. In [2], it was proposed an example illustrating that, with a linearly separable training set, a network performing gradient descent may get stuck in a solution which fails to separate the data, thus pessimistically concluding that BP fails where perceptron succeeds. Nevertheless, the analysis of such cases reveals that these *spurious* local minima are due to an *unproper* joined choice of the cost function, the nonlinear neuron functions, and the target values. A quick glance makes it clear how these examples only hold by choosing unsuitable targets (typically different from the asymptotic squashing function values). Using instead a Least Mean Square (LMS) threshold cost function [4], where values “*beyond*” the targets are not penalised, these counterexamples cease to exist. Under the latter assumption, in fact, a convergence theorem, strictly related to that of Perceptron, holds [5, 6] also for networks with a hidden layer. On the other hand, if we look at the problem of supervised learning in general, the shape of the cost function depends on several elements. Keeping fixed the pattern of connectivity, squashing and cost functions still play quite an important role. As a result, different choices of the cost may lead to optimisation problems with different minima. Moreover, the existence of local minima is due in general to the shape of the error function. In [3] it was shown an example of local minimum in a single-layered network which is remarkably different from those described above, since it involves the problem structure.

1.2 Premature network saturation

Problems of sub-optimal learning may also arise when learning with “*high*” initial weights. In literature, this is referred to as “*premature saturation*” (see e.g.: [7]). The problems deriving from high weights are essentially due to the neuron saturation, that, in turn, makes the error backpropagation very hard. Obviously, the neuron saturation is strictly related to the neuron fan-in. It is important to emphasise that by augmenting the fan-in, the probability of neuron saturation is increased. These considerations, together with the fact that local minima emerge from problems where a sort of symmetry exists in both network and data, suggest choosing the initial weights neither too high nor too small.

An interesting way of facing the premature saturation problem is to use the “*relative cross-entropy metric*”. The main difference of this metric with respect to the ordinary quadratic or LMS cost is that the erroneous saturation of output neurons does not lead to plateaux, but to very high values of the error function. In fact, the large plateaux associated with other costs do not represent local minima and, consequently, do not attract the learning trajectory toward sub-optimal solutions. However, the computational burden for escaping from similar “*entrapments*” may be extremely serious, due to the limited numerical precision. When using the relative cross-entropy metric, the repulsion from the previous erroneous configurations is much more effective, for there are no plateaux but surfaces with high gradient, and underflow errors are likely to be avoided.

These analyses suggest that the problem of premature saturation in multilayered networks must be carefully considered, but that there are also effective techniques to approach it.

2 Learning with no local minima

This section contains theoretical results aimed at guaranteeing local minima free error surfaces under some hypotheses on network and data. The identification of similar conditions ensures global optimisation just by using simple gradient descent learning algorithms, while their interest is motivated by the comparison with the Perceptron Learning algorithm [8, 9] for which “*perfect separation*” is guaranteed under the assumption of linearly separable patterns. Moreover, roughly speaking, the Backpropagation convergence is guaranteed for “*many input*” and “*many hidden unit*” networks.

The general analysis for the case of multilayered networks given in [5] provides some theoretical con-

ditions ensuring local minima free error surfaces. In particular, the so called “*pyramidal networks*”, commonly used in pattern recognition, fulfill the above conditions. In [5] the already cited analysis is also specialised for the case of linearly separable patterns, which is likely to hold for patterns represented by “*many coordinates*”. The following theorem introduces some hypotheses primarily concerning the network architecture, but also the relationship between the network and the learning environment.

Theorem 1 *The cost function $E^{LMS}(\mathcal{N}, \mathcal{L}_e)$ is local minima free if the network \mathcal{N} is pyramidal (i.e. $n(l+1) \leq n(l)$, $l = 1, \dots, L-1$), the weight layer matrices \mathcal{W}_l , $l = 1, \dots, L-1$, are full rank matrices, and the associated learning environment \mathcal{L}_e meet the hypothesis:*

$$Ker[(\mathcal{X}_0^e)'] \cap \mathcal{S}_1^y = \{0\}, \quad (1)$$

where $\mathcal{S}_1^y \subset \mathbb{R}^{T, n(1)}$ represents the set of all the δ -errors $\mathcal{Y}_1 \doteq [y_{i(1)}(1), \dots, y_{i(1)}(T)]'$, $y_{i(1)}(t) \doteq \partial E_t / \partial a_{i(1)}(t)$ generated by varying the weights in the weight space Ω and \mathcal{X}_0^e is the input-matrix. ■

We can think of Theorem 1 as a first general attempt to investigate the presence of stationary points in the error surface in the case of pyramidal networks. From this general point of view, the problem is very complex and the role of this theorem is essentially that of moving all the difficulties to condition (1). A case in which it holds is when all the patterns are linearly independent, since in that case $Ker[(\mathcal{X}_0^e)'] = \{0\}$. It is worth mentioning that if $Ker[(\mathcal{X}_0^e)'] = \{0\}$ is satisfied, then the hypothesis only involves the learning environment. This is a very desirable property but, on the other hand, when the patterns are linearly independent, the number of patterns T cannot be greater than the input cardinality $n(0)$.

In order to discover meaningful conditions with a straightforward and practical interpretation, it is essential to investigate the case of patterns that are separable by a family of surfaces. The following theorem deals with the simplest case of linearly separable patterns and specialises the results given in Theorem 1 under this new assumption.

Theorem 2 *The cost function $E^{LMS}(\mathcal{N}, \mathcal{L}_e)$ is local minima free if the network and the learning environment satisfy the hypotheses:*

- **Network.** *The network has only one hidden layer ($L = 2$) and C outputs, where C is the number of classes. Full connections are assumed from the input to the hidden layer which is divided into C sub-layers, while connections are only permitted from any sub-layer to the associated output.*
- **Learning environment.** *All the patterns of \mathcal{L}_e are linearly separable and exclusive coding is used for the output.* ■

The hypothesis on the architecture is not very restrictive. No output interaction is assumed, that is the outputs are computed independently to each other. Moreover modularity assures such architectures to learn faster than those with fully-connected layers. The hypothesis of linearly separable patterns suggests a comparison with Rosenblatt’s Perceptron [8]. It is well-known that this hypothesis is also sufficient for guaranteeing, in the case of the simple Perceptron, the convergence of the δ -rule [8, 9, 10] to configurations where all the patterns are correctly classified. Nevertheless, in the case of multilayered networks, the assumption of linearly separable patterns is only sufficient to guarantee the convergence of a gradient descent learning algorithm. Moreover, the generalisation to new examples is significantly better for networks with a hidden layer.

On the other hand, in [11, 12], the absence of local minima is guaranteed for networks with one hidden layer and as many hidden neurons as patterns. In the case of networks with many hidden units the convergence can be established in very general situations, but unfortunately, the resulting architectures have a poor capability of generalisation.

Finally, the results given for feedforward networks can be extended also to other multilayered architectures having different types of neurons. Recently, in [13] was analysed the problem of optimal learning for radial basis functions. Under the assumption that the patterns are separable by hyperspheres, which turns out to be a sort of “*dual condition*” of linear separability for inner product based neurons, it can be proved that the cost function is local minima free.

Theorem 3 *Cost function $E^{LMS}(\mathcal{N}, \mathcal{L}_e)$ is local minima free if the network \mathcal{N} and the learning environment \mathcal{L}_e satisfy the hypotheses:*

- **Network.** *The network has C outputs, where C is the number of classes. Full connections are assumed from the input to the hidden layer which is divided into C sub-layers, while connections are only permitted from any sub-layer to the associated output.*
- **Learning environment.** *All the patterns of \mathcal{L}_e are separated by hyperspheres and exclusive coding is used for the output.* ■

3 Theoretical problems with optimal learning

We introduce the concept of “*non-suspect problems*” to address situations in which there exists a canonical gradient descent scheme that turns out to be the optimal algorithm. Conversely, we define “*suspect*” a problem in which classical learning algorithms based on optimisation are likely to fail [14]. This seems to stress the role of neural networks for solving non-suspect problems but, on the other hand, seems to raise also a warning on the actual capabilities of such learning systems for problems exhibiting “*high*” computational complexity.

3.1 Canonical form of gradient descent learning

Let us consider the following learning scheme:

$$\frac{dW}{dt} = -\gamma \nabla_W E = f(t, W), \quad (2)$$

where $E(W)$ is the cost function and $W \in \mathbf{R}^m$ is the weight vector.

Let us choose $\gamma \doteq \Psi(E)/\|\nabla_W E\|_2^2$, being Ψ a non-negative continuous function such that $\Psi(E) = 0$ if and only if $E = 0$. Based on this choice of the learning rate, the dynamics of the error function becomes

$$\frac{dE}{dt} = (\nabla_W E)^T \frac{dW}{dt} = (\nabla_W E)^T \left(-\frac{\Psi(E)}{\|\nabla_W E\|_2^2} \nabla_W E \right) = -\Psi(E),$$

which makes the cost function continuously decreasing to zero. In the case of unimodal functions the configurations for which $\nabla_W E = 0$ are singular points that attract the learning trajectory [15].

In this paper, we are interested in finding terminal attractors [15] and, particularly, in minimising the time t_e required to approach the optimal solution. The choice $\Psi(E) = \eta$ fulfills our needs. Consequently $t_e = E_0/\eta$ and, in particular, selecting $\eta = E_0/\sigma$, the terminal attractor is approached for $t_e = \sigma$, independently of the problem at hand, while the corresponding weight updating equation becomes $\frac{dW}{dt} = -\eta \frac{\nabla E}{\|\nabla E\|_2^2}$. As shown in the following, this way of forcing the dynamics leads to establish intriguing links between the concept of non-suspect problems and their computational complexity.

Let us now consider the following discrete version of equation (2):

$$W_{k+1} = W_k - \tau_k \frac{\eta \nabla_k E}{\|\nabla_k E\|_2^2}, \quad (3)$$

where k is the index associated with the continuous time t and τ_k is the quantisation step. This discrete version comes out just from taking Euler's approximation of dW/dt .

Definition 1 Equation (3) is a consistent approximation of equation (2) with $\Psi(E) = \eta$ provided that, $\forall \epsilon_a > 0$, $\exists \{\tau_k\}$ such that, for all points arising from the chosen quantisation, $|E(t) - E_k| < \epsilon_a$, being $E(t)$ and E_k the error functions associated with the continuous and discrete weight updating equations, respectively.

The most interesting case takes place when the quantisation step τ_k is constant during the learning trajectory (i.e. $\tau_k = \tau$). Let us assume that we have to load n examples into a parametrical system that is based on m weights.

Definition 2 Difference equation (3) is non-suspect if:

1. there exists an initialization algorithm having a complexity lower or equal to $\mathcal{O}(m n)$ acting with constant quantisation steps such that the condition: $\left| \frac{\partial E(W_{j,k})}{\partial W_{j,k}} \right| > \epsilon_s$ (where $W_{j,k}$ denotes the j -th weight at the k -th iteration) is met during the learning trajectory apart from the global minimum;
2. there exists $h > 0$ such that

$$\left| \frac{\partial^2 E}{\partial W_i \partial W_j} \right| < h \quad \forall i, j = 1, \dots, m$$

holds uniformly with n and m .

The following theorem can be established.

Theorem 4 Let us assume that the difference equation (3) is non-suspect. Then, $\forall \epsilon_e > 0$, the difference equation (3) is a consistent approximation of the differential equation (2) in the domain $\mathcal{D}_{\epsilon_e} \doteq \{W \in \mathbf{R}^m : E(W) > \epsilon_e\}$, when choosing quantization steps no higher than

$$\tau^* = 2\sigma \left[\frac{\epsilon_s^2 \epsilon_e}{h E_0^2} \right]. \quad (4)$$

Moreover, $E(W_{k^*}) \leq \epsilon_e$ holds after at least k^* iteration steps of equation (3), being

$$k^* = \frac{1}{2} \left[\frac{h E_0^2}{\epsilon_s^2 \epsilon_e} \right]. \quad (5)$$

3.2 Links to computational complexity

Lemma 1 Let $\mathcal{E}_{m,n} = \{\mathcal{N}_m, \mathcal{L}_{\epsilon,n}, E_{m,n}\}$ be an experiment using multilayer network $\mathcal{N}_m(W)$ and data $\mathcal{L}_{\epsilon,n}$. Then Backpropagation is the optimal algorithm for computing the gradient and takes $\mathcal{O}(m n)$. ■

The following theorem establishes a formal link between families of non-suspect loading problems and their computational complexity.

Theorem 5 Let $\mathcal{F}_P^{\Xi} \doteq \{\mathcal{P}_{m,n}^l : \Xi\}$ be a family of non-suspect loading problems associated with the experiment $\mathcal{E}_{m,n} = \{\mathcal{N}_m, \mathcal{L}_{\epsilon,n}, E_{m,n}\}$. Then the Backpropagation algorithm included in the weight updating equations (3) is optimal and takes $\mathcal{O}(m n)$. ■

Notice that the bound $\mathcal{O}(m n)$ can be derived by exploiting the structure of feedforward networks where the gradient is computed by Backpropagation. According to Lemma 1, the neural hypothesis leads to the lower bound. Theorem 5 has straightforward consequences that arise from considering some theoretical results on non-suspect loading problems. If one establishes that a given family of loading problem is non-suspect, than we can assess that for such family the complexity bound is $\mathcal{O}(m n)$.

References

- [1] L. G. C. Hamey, “The analysis of local minima in feed-forward neural networks.” Submitted, 1995.
- [2] M. Brady, R. Raghavan, and J. Slawny, “Back-propagation fails to separate where perceptrons succeed,” *IEEE Transactions on Circuits and Systems*, vol. 36, pp. 665–674, 1989.
- [3] E. Sontag and H. Sussman, “Backpropagation can give rise to spurious local minima even for networks without hidden layers,” *Complex Systems*, vol. 3, pp. 91–106, 1989.
- [4] E. Sontag and H. Sussman, “Backpropagation separates when perceptrons do,” in *International Joint Conference on Neural Networks*, vol. 1, (Washington DC), pp. 639–642, IEEE Press, June 1989.
- [5] M. Gori and A. Tesi, “On the problem of local minima in backpropagation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-14, pp. 76–86, January 1992.
- [6] P. Frasconi, M. Gori, and A. Tesi, “Backpropagation for linearly separable patterns: a detailed analysis,” in *IEEE International Conference on Neural Networks*, vol. 3, (San Francisco, (CA)), pp. 1818–1822, IEEE Press, March-April 1993.
- [7] Y. Lee, S. Oh, and M. Kim, “The effect of weights on premature saturation in back-propagation learning,” in *International Joint Conference on Neural Networks*, vol. 1, (Seattle), pp. 765–770, July, 8–12 1991.
- [8] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanism*. Washington D.C.: Spartan Books, 1962.
- [9] M. Minsky and S. Papert, *Perceptrons — Expanded Edition*. Cambridge: MIT Press, 1988.
- [10] N. Nilsson, *Learning Machines*. New York: McGraw-Hill, 1965. Reissued as *Mathematical Foundations of Learning Machines*, Morgan Kaufmann Publishers, San Mateo, CA, 1990.
- [11] T. Poston, C. Lee, Y. Choie, and Y. Kwon, “Local minima and backpropagation,” in *International Joint Conference on Neural Networks*, vol. 2, (Seattle, (WA)), pp. 173–176, IEEE Press, July 1991.
- [12] X. Yu, “Can backpropagation error surface not have local minima?,” *IEEE Transactions on Neural Networks*, vol. 3, pp. 1019–1020, November 1992.
- [13] M. Bianchini, P. Frasconi, and M. Gori, “Learning without local minima in radial basis function networks,” *IEEE Transactions on Neural Networks*, vol. 6, pp. 749–756, May 1995.
- [14] M. Bianchini, F. Fanelli, M. Gori, and M. Protasi, “Unimodal loading problems.” To be published on *Mathematics of Neural Networks: Models Algorithms and Applications*, Eds. S.W. Ellacott, J.C. Mason, and I.J. Anderson, Kluwer Academic Operations Research/Computer Science Interfaces.
- [15] M. Bianchini, M. Gori, and M. Maggini, “Does terminal attractor backpropagation guarantee global optimization?,” in *International Conference on Artificial Neural Networks*, vol. 1, (Sorrento, Italy), pp. 377–380, Springer-Verlag, May, 26–29, 1994.