

# Suspiciousness of Loading Problems

P. Frasconi, M. Gori  
Università di Firenze (Italy)

www: <http://www-dsi.ing.unifi.it/marco>

S. Fanelli, and M. Protasi  
Università di Roma “Tor Vergata” (Italy)

email: [fanelli@mat.utovrm.it](mailto:fanelli@mat.utovrm.it)

## Abstract

*We introduce the notion of suspect families of loading problems in the attempt of formalizing situations in which classical learning algorithms based on local optimization are likely to fail (because of local minima or numerical precision problems). We show that any loading problem belonging to a non-suspect family can be solved with optimal complexity by a canonical form of gradient descent with forced dynamics (i.e., for this class of problems no algorithm exhibits a better computational complexity than a slightly modified form of backpropagation). The analyses of this paper suggest intriguing links between the shape of the error surface attached to parametrical learning systems (like neural networks) and the computational complexity of the corresponding optimization problem.*

**Index Terms-** Computational complexity, gradient descent, local minima, suspiciousness.

## 1. Introduction

Most experiments with multilayer perceptrons (MLP) and Backpropagation (BP) are performed in a sort of “magic” atmosphere where one supplies data properly to the network and begins learning without knowing whether or not the experiment will be successful in terms of both optimal convergence and generalization. As a matter of fact one uses a *trial and error* scheme aimed at adjusting the architecture in subsequent experiments so as to meet the desired requirements. To some extent, this way of performing experiments is inherently plagued by the suspect that the numerical optimization algorithm might fail discovering an

optimal solution.

There are at least three reasons that may lead an experiment to fail. First, the chosen architecture may not be able to solve the given problem as no solution exists for performing the required mapping (a representational problem). Second, the presence of local minima [4] may seriously limit the chance of discovering an optimal solution (a problem inherent to local optimization techniques such as gradient descent). Finally, because of the presence of plateaus or very abrupt zones in the error surface, any “reasonable” choice of the discretization step may be ineffective (a numerical precision problem). These potential sources of failure give rise to a sort of *suspiciousness* that turns out to be the unpleasant companion of every experiment.

In this paper we focus on classification tasks and we formally introduce the class  $\mathcal{AS}$  of *families of non-suspect loading problems*, based on some conditions about the shape of the error surface. We show that any loading problem in  $\mathcal{AS}$  can be solved by a canonical form of gradient descent algorithm whose computational complexity is  $\Theta(mn)$ , being  $m$  the number of weights and  $n$  the number of examples. To the best of our knowledge, this is a novel theoretical result that emphasizes the role of neural information processing systems. Basically, for this class of learning problems, there is no algorithm which exhibits a computational complexity better than neural networks.

## 2. The loading problem in parametrical systems

The learning machine we consider has the form of a general *parametrical system*

$$Y = \mathcal{S}(U, W), \quad (1)$$

where the input  $U \in \mathcal{R}^\ell$  is mapped to the output  $Y \in \mathcal{R}$  by means of the weight vector  $W \in \mathcal{R}^m$ . Neural networks like multilayer perceptrons and radial basis functions are special interesting case of such parametrical systems. To make explicit the two dimensions that characterize the network (input size  $\ell$  and number of weights  $m$ ), we shall write in the following  $\mathcal{S}_{m,\ell}$ .

The given parametrical system is charged of learning a concept  $c$  defined as a partition of some (abstract) sample space  $X$  into a positive set  $X^+$  and a negative set  $X^-$ . The elements of  $X$  need to be represented as real vectors in order to be used as input to the network. To this purpose we introduce an injective operator  $r_\ell(\cdot) : X \rightarrow \mathcal{R}^\ell$  that maps an abstract entity  $x$  into a representation  $U = r(x)$ . The learning procedures determines a weight vector  $W$  on the basis of a set of examples

$$\mathcal{L}_n \doteq \{(U_q, d_q), U_q \in \mathcal{R}^\ell, d_q \in \{d^-, d^+\}, q = 1 \dots n\}$$

where  $d^-, d^+ \in \mathcal{R}$  are target values associated to the class, i.e.  $d_q = d^+$  iff  $r^{-1}(U_q) \in X^+$ .

Given the pair  $(\mathcal{S}_{m,\ell}, \mathcal{L}_n)$ , the output-target data fitting is measured by means of a proper error function, defined as following:

$$E(\mathcal{S}_{m,\ell}, \mathcal{L}_n) \doteq \frac{1}{n} \left( \sum_{q=1}^n e(Y_q(U_q, W) - d_q) \right), \quad (2)$$

where  $e(\cdot)$  is some distance in  $\mathcal{R}$ . Any loading problem can be compactly represented by the triple  $\mathcal{P}_{m,n,\ell} \doteq \{\mathcal{S}_{m,\ell}, \mathcal{L}_n, E(\cdot, \cdot)\}$ , where the task is that of finding the weights that minimize the function  $E(\cdot, \cdot)$ . We say that the loading problem is approximately correctly solved with approximation  $\delta$  if the learning algorithm determines a weight vector such that  $E(\mathcal{S}_{m,\ell}, \mathcal{L}_n) < \delta$ . Note that such condition allows us to limit the average error associated with each example  $q$  of  $\mathcal{L}_n$ , but  $\delta$  and the fraction of correctly loaded examples are not the same quantity.

Negative results have been established in the literature concerning the intractability of the loading problem [5]. These results, however, do not take into account the degrees of freedom available to a designer when attempting to solve the original concept learning problem. These degrees of freedom are: 1) the choice of the input representation; 2) the choice of the architecture; 3) the choice of the training examples (when possible). In order to formalize the practical situation faced when designing an adaptive network for

concept learning, instead of a *single* loading problem we consider a *family of loading problems* induced by the concept  $c$ :

$$\mathcal{F}_c^\Xi \doteq \{\mathcal{P}_{m,n,\ell} : \mathcal{P}_{m,n,\ell} = \xi(c), \xi(\cdot) \in \Xi\}$$

where  $\xi(\cdot)$  is a function that transforms a concept  $c$  into a particular loading problem, by making choices about input representation, architecture and training examples (such a function is usually “computed” by a designer faced with the problem of having a machine to learn  $c$ ). The family of loading problems is described by picking up  $\xi(c)$  from a family  $\Xi$  of “design rules.” The rules in  $\Xi$  specify how scale up  $\ell$ ,  $m$ , and  $n$  (and, eventually, how to modify the connectivity of the network). Typically, such rules should also aim to guarantee a satisfactory generalization to new examples. Thus, as an example, the design rules family  $\Xi$  might be constrained to contain all those loading problems in which the number of training samples  $n$  is at least equal to the sample complexity, as computed in the PAC learning framework [1].

### 3. Canonical form of gradient descent learning

Let us consider the following learning scheme

$$\frac{dW}{dt} = -\gamma \nabla_W E = f(t, W), \quad (3)$$

where  $E(W)$  denotes the value taken on by the error function  $E(\mathcal{S}_{m,\ell}, \mathcal{L}_n)$  (for a fixed architecture and a fixed training set) for a given weight vector  $W \in \mathcal{R}^m$ . Let us choose  $\gamma$  as

$$\gamma \doteq \frac{\Psi(E)}{\|\nabla_W E\|^2}, \quad (4)$$

being  $\Psi$  a non-negative continuous function of  $E$ . Based on this choice of the learning rate, the dynamics of the error function becomes

$$\begin{aligned} \frac{dE}{dt} &= (\nabla_W E)^T \frac{dW}{dt} \\ &= (\nabla_W E)^T \left( -\frac{\Psi(E)}{\|\nabla_W E\|^2} \nabla_W E \right) \\ &= -\Psi(E), \end{aligned} \quad (5)$$

which makes the error function continuously decreasing to zero. As pointed out in [6], the choice of the adaptive learning rate produces a *forced dynamics* for the error function

that, to some extent, is equivalent to the choice of a linear “sliding surface” in Sliding Mode Control [8].

In this paper, we are interested in finding terminal attractors and, particularly, in minimizing the time  $t_e$  required to approach the optimal solution. For this reason, the choice of  $\Psi(E) = \eta$  fulfills our needs. In particular, *if we choose  $\eta = E_0/\sigma$ , then the terminal attractor is approached for  $t_e = \sigma$  independently of the problem at hand* and the corresponding weight updating equation becomes

$$\frac{dW}{dt} = -\frac{E_0}{\sigma} \frac{\nabla_W E}{\|\nabla_W E\|^2}. \quad (6)$$

As shown in the following sections, this way of forcing the learning dynamics leads us to establish intriguing links between non-suspiciousness and computational complexity.

## 4. Discretization of the learning trajectory

The analysis carried out in the previous section holds for continuous computational models. Let us now consider the following discrete version of equation (6)

$$W_{k+1} = W_k - \tau \frac{\eta(\nabla_W E)(W_k)}{\|(\nabla_W E)(W_k)\|^2}, \quad (7)$$

where the discrete time  $k$  is related to the continuous time  $t$  and to the quantization step  $\tau$  by  $t = \tau k$ . Although far away from representing the best numerical approximation of equation (6), this simple discretization process turns out to be adequate for our theoretical purposes.

**Definition 1** *Difference equation (7) is a consistent approximation of differential equation (6) provided that  $\forall \epsilon_a > 0 \exists \tau > 0$  such that  $\forall t = k\tau$ ,  $|E(t) - E_k| < \epsilon_a$ , being  $E(t)$  and  $E_k$  the functions associated with the differential and difference equations, respectively.*

**Definition 2** *A weight initialization algorithm  $\mathcal{O}_i^s$  is called a strong initialization oracle for discrete equation (7), provided that it can predict an initial value  $\tilde{W}$  such that  $\forall \epsilon_e > 0, \exists \epsilon_s > 0$  such that  $\forall j = 1, \dots, m$ , the condition  $|\frac{\partial E(W_{j,k})}{\partial W_{j,k}}| > \epsilon_s$  (where  $W_{j,k}$  denotes the  $j$ -th weight at the  $k$ -th iteration) is met everywhere<sup>1</sup>, apart from  $k$  such that  $E(W_k) \leq \epsilon_e$ .  $\epsilon_s$  is referred to as the gradient stopping parameter.*

<sup>1</sup>In order to establish the asymptotic results ( $n \rightarrow \infty$ ) of section 5, this assumption can slightly be relaxed. In that case, one needs that this conditions holds for a number of gradient coordinates  $\rho(m)$  such that  $\lim_{m \rightarrow \infty} \rho(m)/m = 1$ .

An algorithm  $\mathcal{O}_i^w$  which can only guarantee that  $\|(\nabla_W E)(W_k)\| > \epsilon_s$  (i.e., in norm instead of component-wise) is met everywhere during the learning trajectory will be referred to as a *weak initialization oracle*. Unlike the strong one, for which  $\|(\nabla_W E)(W_k)\| > \epsilon_s \sqrt{m}$  holds<sup>2</sup> during the learning trajectory, there is no scaling up of the norm of the gradient with the dimension of the learning system.

It is worth mentioning that the requirement  $|\frac{\partial E}{\partial W_i}| > \epsilon_s \forall i$  can be slightly relaxed. It suffices indeed that such requirement be met for any set of parameters obtained by a rotation of  $W$ . For, let  $W \in \mathcal{R}^m$  be the original vector of parameters and  $\tilde{W} \in \mathcal{R}^m$  be a vector obtained by a rotation according to  $\tilde{W} = PW$ , being  $P^{-1} = P'$ . The gradients with respect to these vectors are related by  $(\nabla_W E)(\tilde{W}) = P'(\nabla_W E)(W)$ , which implies  $\|(\nabla_W E)(\tilde{W})\| \leq \|P'\| \|(\nabla_W E)(W)\|$ , that is  $\|(\nabla_W E)(\tilde{W})\| \leq \|(\nabla_W E)(W)\|$ , being  $\|P'\| = 1$ . Consequently, if an oracle guarantees one of the two norm inequalities  $\|(\nabla_W E)(\tilde{W})\| > \epsilon_s$ ,  $\|(\nabla_W E)(\tilde{W})\| > \epsilon_s \sqrt{m}$  then the same inequalities hold for the gradient with respect to  $W$ .

A special case in which the oracles can be trivially found is the case of local minima free error functions (it suffices to generate a random point in the weight space). In this context, the research, recently carried out in ([4]), aimed at determining conditions that guarantee the absence of local minima turns out to be interesting.

The following discretization analysis is based on some assumptions that will be referred to as *non-suspiciousness conditions* for difference equation (7).

**Definition 3** *We say that difference equation (7) is strongly non-suspect if*

1. *there exists a strong initialization oracle  $\mathcal{O}_i^s$  capable of guessing the initialization of equation (7), whose complexity is independent of  $n$  and  $m$ ;*
2. *there exist  $h > 0$  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  $m$ .*

<sup>2</sup>Since  $\|(\nabla_W E)(W_k)\| = \sum_{i=1}^m (\frac{\partial E}{\partial W_i})^2 \geq m \epsilon_s^2$  we have  $\|(\nabla_W E)(W_k)\| \geq \epsilon_s \sqrt{m}$ .

Weak non-suspiciousness can be similarly defined.

**Theorem 1** *Let us assume that the difference equation (7) is strongly non-suspect. Then,  $\forall \epsilon_e > 0$ , difference equation (7) is a consistent approximation of differential equation (6) in domain  $\mathcal{D}_{\epsilon_e} \doteq \{W \in \mathcal{R}^m : E(W) > \epsilon_e\}$ , when choosing quantization steps no higher than*

$$\tau^* = 2\sigma \left[ \frac{\epsilon_s^2 \epsilon_e}{hE_0^2} \right]. \quad (8)$$

Moreover,  $E(W_{k^*}) \leq \epsilon_e$  holds after at least  $k^*$  iteration steps of equation (7), being

$$k^* = \frac{1}{2} \left[ \frac{hE_0^2}{\epsilon_s^2 \epsilon_e} \right]. \quad (9)$$

*Proof:* When updating the parameters from  $W_k$  to  $W_{k+1}$ , according to equation (7), the corresponding discrete form of the error function changes from  $E_k$  to  $E_{k+1}$ . This variation can be evaluated as follows by using Taylor's theorem in  $W_k$

$$\begin{aligned} E_{k+1} &= E_k \\ &+ (\nabla_W E)(W_k)'(W_{k+1} - W_k) \\ &+ \frac{1}{2}(W_{k+1} - W_k)' \mathcal{H}(\xi_k)(W_{k+1} - W_k), \end{aligned} \quad (10)$$

where  $'$  denote the transpose operator and  $\xi_k$  is a proper value in a neighborhood of  $W_k$ . Because of equation (7), we get

$$E_{k+1} = E_k - \tau\eta + \frac{1}{2}(W_{k+1} - W_k)' \mathcal{H}(\xi_k)(W_{k+1} - W_k).$$

Since  $E(t) = E_0 - \eta t$ , the chosen approximation gives rise to an error in the above equation that involves the second-order term only<sup>3</sup>.

Once  $W(t)$  is restricted to  $\mathcal{D}_{\epsilon_e}$ , the correspondent function  $f(t, W)$  of differential equation (6) has a bounded partial derivative with respect to its second variable  $W$ , and its solution has a bounded second derivative. Under these condition Euler's approximation (7) of equation (6) converges to the exact solution as  $\tau \rightarrow 0$  (see [3], p. 26). Since  $E \in \mathcal{C}^2$ , function  $E$  inherits the convergence of  $W$  and, therefore, the condition of consistent approximation of equation (1) holds.

<sup>3</sup>Note that this property holds because of the special choice of function  $\Psi(E)$ . In general a discretization error is introduced also concerning the first-order term.

We can always choose  $\tau$  such that the algorithm is stopped in  $k^* \leq t_e/\tau$  steps. To obtain a bound on  $\tau$ , observe that the following inequality must hold for the approximation error  $\epsilon_a \doteq \epsilon_e$ :

$$\frac{1}{2} \frac{t_e}{\tau} \max_k (|(W_{k+1} - W_k)' \mathcal{H}(\xi_k)(W_{k+1} - W_k)|) \leq \epsilon_e. \quad (11)$$

According to the hypotheses, there exists a strong oracle  $\mathcal{O}_i^s$  guaranteeing that  $\|(\nabla_W E)(W_k)\| > \sqrt{m} \epsilon_s$  holds during the gradient descent, apart from absolute minima configurations, and therefore

$$\begin{aligned} &\frac{1}{2} \frac{t_e}{\tau} \max_k (|(W_{k+1} - W_k)' \mathcal{H}(\xi_k)(W_{k+1} - W_k)|) \\ &\leq \frac{1}{2} \frac{t_e}{\tau} \max_k (\|W_{k+1} - W_k\| \|\mathcal{H}(\xi_k)\| \|W_{k+1} - W_k\|) \\ &\leq \frac{1}{2} \frac{t_e}{\tau} H \left( \frac{\eta\tau}{\epsilon_s \sqrt{m}} \right)^2 \leq \frac{1}{2} \frac{t_e}{\tau} m h \left( \frac{\eta\tau}{\epsilon_s \sqrt{m}} \right)^2. \end{aligned} \quad (12)$$

Hence, inequality (11) holds if  $\tau$  is chosen according to (8). The previous inequalities are based on the assumption of using the Euclidean norm in the weight space and the induced spectral norm for the Hessian matrix. From the last one we can immediately determine the limit value of the quantization step  $\tau^*$ , given by equation (8). Finally, given the value of  $\tau^*$ , we can easily derive the number of steps given by equation (9).  $\diamond$

The condition that seems to be more demanding in this theorem is the one which involves the existence of an initialization oracle. An intriguing conclusion is that the quantization step for attaining the desired approximation is *independent of the special error function involved* and is related to a few parameters only  $(\epsilon_e, \epsilon_s, E_0, h)$ . Note that, because of Definition (2), the error has an upper bound independently of  $n$  and, since we consider the case of one output only, because of the squashing function in the output neuron, the error is also bounded independently of the number of parameters  $m$ . In general, bounding uniformly the coordinates of the Hessian is not as simple and must be carefully analyzed ([2]).

## 5. Non-suspect Loading and the connectionist assumption

**Definition 4** *Let us consider a family of loading problems  $\mathcal{F}_c^\Xi$  induced by a concept  $c$ . We say that this family is*

strongly non-suspect,  $\mathcal{F}_c^\Xi \in \mathcal{NS}$ , if  $\exists \tilde{m}, \tilde{n}$  such that for any  $\mathcal{P}_{m,n,\ell}$  with  $m > \tilde{m}, n > \tilde{n}$ , the difference equation (7) is non-suspect.

Note that this definition has an asymptotic nature and that it assumes practical relevance once the number of parameters is chosen in such a way that a satisfactory generalization to new examples is guaranteed. In fact, growing the network size has a favorable impact on the quality of the error function. In the limit case of a two-layered network having as many hidden neurons as training example, the error function can be proven to be local minima free (see e.g. [7, 9]). Thus, if the class of design choices  $\Xi$  would contain rules that tend to produce relatively large networks for relatively limited number of examples, the family of loading problems would always be non-suspect. Of course, such a design strategy would have very little practical significance. We have a less trivial case of local minima free error functions when the training examples are linearly separable [4]. Linearly separable patterns may be obtained by increasing the input dimension  $\ell$  (for example using polynomial preprocessing); clearly, this may be another design strategy not compatible with generalization, unless plentiful training data is available.

Finding initialization oracles with the required computational constraints in cases in which  $E(\mathcal{S}_{m,\ell}, \mathcal{L}_n)$  is multimodal seems to be hard and is an open problem. Let us focus on the case of multilayer perceptrons.

**Theorem 2** *Let  $\mathcal{F}_c^\Xi$  be a strongly non-suspect family of loading problems induced by concept  $c$ . Then, for any loading problem  $\mathcal{P}_{m,n,\ell} \in \mathcal{F}_c^\Xi$  the algorithm (7) with stopping criterion  $E(W) < \epsilon_e$  is optimal and takes  $\Theta(m n)$ .*

*Proof:* Under the assumption of the theorem, the number of steps required for meeting the stopping criterion is independent of the problem at hand, being difference equation (7) strongly non-suspect and, consequently,  $h$  independent of  $m, n$ . In the case of multilayer perceptrons, the gradient can exactly be computed by using Backpropagation, which takes  $\Theta(m n)$  (see e.g. [4])<sup>4</sup>. Moreover, no algorithm can load the weights of  $\mathcal{N}_m$  with complexity lower than that required by difference equation (7), since all the  $n$  examples must be inspected by any candidate loading algorithm, that must also take all the  $m$  weights into account for minimizing the error function.  $\diamond$

<sup>4</sup>Note that ordinary numerical approximations based on weight perturbation would take  $O(m^2 n)$ .

Note that the bound  $\Theta(m n)$  comes out from exploiting the structure of feedforward networks for which the gradient can be computed by Backpropagation. Unfortunately, the optimality property of equation (7) seems to be lost once relying on *weak* initialization oracles. In that case, the same analysis of Theorem 2 leads us to conclude that the solution of loading problems according to canonical gradient descent takes  $O(m^2 n)$ , since those oracles can only ensure that  $\|(\nabla_W E)(W_k)\| > \epsilon_s$  during the gradient descent ([2]).

Theorem 2 has straightforward consequences that arise from considering some theoretical results on non-suspect families of loading problems. If a given family  $\mathcal{F}_c^\Xi \in \mathcal{NS}$ , then we obtain the complexity bound  $\Theta(m n)$  for that family. For instance, it can be proven that a strongly non-suspect family of loading problems can be created for learning the weights of a single layer perceptron ([2]).

## 6. Conclusions

In this paper, we have shown that there are some intriguing links between the computational complexity of the loading problem and the form of the associated error surface. In particular, we have proven that loading problems giving rise to unimodal error functions can be solved optimally by an MLP with lower bound on the complexity of  $\Theta(m n)$  which holds, for instance, in the case of learning linearly separable patterns.

Moreover, once there is evidence on the intractability of a given loading problem, our theoretical results allow us to conclude that such problem is suspect. This is nothing special, apart from the way the face of the computational complexity is revealed for this class of problems. The suspiciousness becomes a more straightforward concept to understand where troubles arise and, hopefully, a concept that could help face them more effectively.

## Acknowledgments

We thank M. Bianchini, M. Maggini, F. Scarselli, and F. Schoen for fruitful discussions and comments on an earlier draft of this paper.

## References

- [1] E. Baum and D. Haussler. What size net gives valid generalization? *Neural Computation*, 1(1):151–160, 1989.
- [2] P. Frasconi, M. Gori, S. Fanelli, and M. Protasi. Links between suspiciousness and computational complexity. Technical report, Dipartimento di Sistemi e Informatica, Università di Firenze, October 1995.
- [3] G. Golub and J. Ortega. *Scientific Computing and Differential Equations*. Academic Press, Inc., 1992.
- [4] M. Gori and A. Tesi. On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-14(1):76–86, January 1992.
- [5] J. Judd. *Neural Network Design and the Complexity of Learning*. The MIT Press, Cambridge, London, 1990.
- [6] A. Parlos, B. Fernandez, A. Atyla, J. Muthusami, and W. Tsai. An accelerated algorithm for multilayer perceptron networks. *IEEE Transactions on Neural Networks*, 5:493–497, 1994.
- [7] T. Poston, C. Lee, Y. Choie, and Y. Kwon. Local minima and backpropagation. In *International Joint Conference on Neural Networks*, volume 2, pages 173–176, Seattle, (WA), July 1991. IEEE Press.
- [8] J. Slotine and S. S. Sastry. Tracking control of non-linear systems using sliding surfaces, with applications to robot manipulators. *International Journal of Control*, 38:465–492, 1983.
- [9] X. Yu. Can backpropagation error surface not have local minima? *IEEE Transactions on Neural Networks*, 3(6):1019–1020, November 1992.