SECURITY CLASSIFICATION

REPOR

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AIM 1220

AD-A234 425

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Extensions of a Theory of Networks and Learning: Outliers and Negative Examples

5. TYPE OF REPORT & PERIOD COVERED
memorandum

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Tomaso Poggio, Federico Girosi, & Bruno Caprile

8. CONTRACT OR GRANT NUMBER(s)
S1-801534-2
DACA76-85-C-0010
N00014-85-K-0124

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Artificial Intelligence Laboratory
545 Technology Square
Cambridge, MA 02139

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

12. REPORT DATE
July 1990

13. NUMBER OF PAGES
24

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Office of Naval Research
Information Systems
Arlington, VA 22217

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
learning
networks
regularization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Learning an input-output mapping from a set of examples, of the type that many neural networks have been constructed to perform, can be regarded as synthesizing a approximation of a multi-dimensional function. From this point of view, this form of learning is closely related to regularization theory. The theory developed in Poggio and Girosi (1989) shows the equivalence between regularization and a class of three-

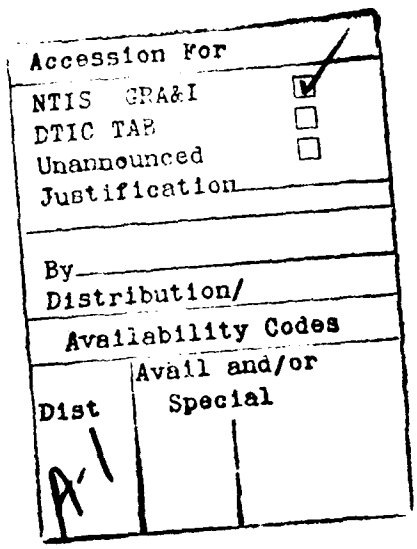
(continued on back)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered

Block 20 continued:

layer networks that we call regularization networks or Hyper Basis Functions. These networks are not only equivalent to generalized splines, but are also closely related to the classical Radial Basis Functions used for interpolation tasks and to several pattern recognition and neural network algorithms. In this note, we extend the theory by introducing ways of dealing with two aspects of learning: learning in the presence of unreliable examples and learning from positive *and* negative examples. These two extensions are interesting also from the point of view of the approximation of multivariate functions. The first extension corresponds to dealing with outliers among the sparse data. The second one corresponds to exploiting information about points or regions in the range of the function that are forbidden.



MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY
and
CENTER FOR BIOLOGICAL INFORMATION PROCESSING
WHITAKER COLLEGE

A.I. Memo No. 1220 July 1990
C.B.I.P. Paper No. 46

Extensions of a Theory of Networks for Approximation and Learning: outliers and negative examples

Federico Girosi, Tomaso Poggio and Bruno Caprile

Abstract

Learning an input-output mapping from a set of examples, of the type that many neural networks have been constructed to perform, can be regarded as synthesizing an approximation of a multi-dimensional function. From this point of view, this form of learning is closely related to regularization theory. The theory developed in Poggio and Girosi (1989) shows the equivalence between regularization and a class of three-layer networks that we call regularization networks or Hyper Basis Functions. These networks are not only equivalent to generalized splines, but are also closely related to the classical Radial Basis Functions used for interpolation tasks and to several pattern recognition and neural network algorithms. In this note, we extend the theory by introducing ways of dealing with two aspects of learning: learning in the presence of unreliable examples and learning from positive ***and*** negative examples. These two extensions are interesting also from the point of view of the approximation of multivariate functions. The first extension corresponds to dealing with outliers among the sparse data. The second one corresponds to exploiting information about points or regions in the range of the function that are forbidden.

This paper describes research done mainly at I.R.S.T. in Trento, Italy, and also within the Center for Biological Information Processing, in the Department of Brain and Cognitive Sciences, and at the Artificial Intelligence Laboratory. This research is sponsored by a grant from the Office of Naval Research (ONR), Cognitive and Neural Sciences Division; by the Artificial Intelligence Center of Hughes Aircraft Corporation (S1-801534-2). Support for the A. I. Laboratory's artificial intelligence research is provided by the Advanced Research Projects Agency of the Department of Defense under Army contract DACA76–85–C–0010, and in part by ONR contract N00014-85-K-0124.

1 Introduction

In previous papers (Poggio and Girosi, 1989, 1990) we have shown the equivalence between regularization and a class of three-layer networks that we called regularization networks and that are related to the classical interpolation technique of Radial Basis Functions.

Let $g = \{(\mathbf{x}_i, y_i) \in R^n \times R\}_{i=1}^N$ be a set of data that we want to approximate by means of a function f. The regularization approach (Tikhonov, 1963; Tikhonov and Arsenin, 1977; Morozov, 1984; Bertero, 1986) selects the function f that solves the variational problem of minimizing the functional

$$H[f] = \sum_{i=1}^{N}(y_i - f(\mathbf{x}_i))^2 + \lambda \|Pf\|^2 \tag{1}$$

where P is a constraint operator (usually a differential operator), $\|\cdot\|$ is a norm on the function space to which Pf belongs (usually the L^2 norm) and λ is a positive real number, the so called *regularization parameter*. The structure of the operator P, that is called "stabilizer", embodies the a priori knowledge about the solution, and therefore depends on the nature of the particular problem that has to be solved. We have shown (Poggio and Girosi, 1989) that the solution of the variational problem (1) has the following simple form:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x}; \mathbf{x}_i) + p(\mathbf{x})$$

where $G(\mathbf{x})$ is the Green's function (Stakgold, 1979) of the self-adjoint differential operator $\hat{P}P$, $\hat{P}$ being the adjoint operator of P, $p(\mathbf{x})$ is a linear combination of functions that span the null space of P, and the coefficients c_i satisfy a linear system of equations that depend on the N "examples", i.e. the data to be approximated. The form of the term $p(\mathbf{x})$ depends on the stabilizer that has been chosen and on the boundary conditions, and therefore on the particular problem that has to be solved (for instance, it is not needed in the case of P corresponding to a Gaussian or bell-shaped Green's function). For this reason, and since its inclusion does not modify the main conclusions, we will disregard it in the following. In the special case in which

P is an operator with radial symmetry, the Green's function G is radial and therefore the approximating function becomes:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\|\mathbf{x} - \mathbf{x}_i\|^2), \tag{2}$$

which is a sum of radial functions, each with its *center* $\mathbf{x}_i$ on a distinct data point. Thus the number of radial functions, and corresponding centers, is the same as the number of examples.

In this note we indicate how to extend our theory of learning from examples in order to deal with 1) occurence of unreliable examples, 2) negative examples. Both problems are also interesting from the point of view of classical approximation theory:

1. discounting "bad" examples corresponds to discarding, in the approximation of a function, data points that are outliers.

2. learning by using negative examples – in addition to positive ones – corresponds to approximating a function based not only on points to which the function must be close but also on points – or regions – that the curve associated with the function must avoid.

2 Unreliable data

Suppose that the set $g = \{(\mathbf{x}_i, y_i) \in R^n \times R\}_{i=1}^N$ of data has been obtained by random sampling a function f, defined on R^n, in presence of noise. We are interested in recovering the function f, or an estimate of it, from the set of data g. We take a probabilistic approach, and regard the function f and the data g as random, dependent, variables. Using Bayes theorem, it is possible to express the conditional probability $\mathcal{P}[f|g]$ of the function f given the examples g in terms of the a priori probability of f, $\mathcal{P}[f]$, and the conditional probability of g given f, $\mathcal{P}[g|f]$, that is equivalent to a model of the noise:

$$\mathcal{P}[f|g] \propto \mathcal{P}[g|f]\ \mathcal{P}[f]. \tag{3}$$

If the noise is Gaussian the probability $\mathcal{P}[g|f]$ can be written as:

$$\mathcal{P}[g|f] \propto e^{-\sum_{i=1}^{N} \beta_i (y_i - f(\mathbf{x}_i))^2} \tag{4}$$

where $\beta_i = \frac{1}{2\sigma_i^2}$ and σ_i is the variance of the noise related to the i-th data point. Under some assumption on the stochastic process f (Marroquin et al., 1987; Geman and Geman, 1984) it is possible to write the a priori probability $\mathcal{P}[f]$ in the following way:

$$\mathcal{P}[f] \propto e^{-\lambda \|Pf\|^2}$$

where P is a constraint operator (usually a differential operator), $\|\cdot\|$ is a norm on the function space to which Pf belongs (usually the L^2 norm) and λ a positive real number. This form of probability distribution gives high probability only to those functions for which the term $\|Pf\|^2$ is small, and embodies the a priori knowledge that one has about the system. For example if one knows that the function f that has been sampled is very smooth, in the sense that it does not vary too "quickly" in its domain, the operator P will be a differential operator of high degree.

Using Bayes theorem (3) the *a posteriori* probability of f can be written as

$$\mathcal{P}[f|g] \propto e^{-[\sum_{i=1}^{N} \beta_i (y_i - f(\mathbf{x}_i))^2 + \lambda \|Pf\|^2]}. \tag{5}$$

A simple way to obtain an estimate of the function f from the probability distribution (5) consists in taking the so called MAP (Maximum A Posteriori) estimate, that is the function that maximizes the a posteriori probability $\mathcal{P}[f|g]$, or minimizes the exponent in equation (5). Setting for simplicity all the variances σ_i equal to one fixed variance σ, and defining from here on

$$\Delta_i = y_i - f(\mathbf{x}_i) \ ,$$

the MAP estimate of f is then the minimum of the following functional:

$$H_0[f] = \frac{1}{2\sigma^2} \sum_{i=1}^{N} V(\Delta_i) + \lambda \|Pf\|^2 \tag{6}$$

where we have defined the quadratic function

$$V(x) = x^2$$

This is equivalent to the so called "regularization technique" (Tikhonov, 1963; Tikhonov and Arsenin, 1977; Morozov, 1984; Bertero, 1986) that has been extensively used in order to solve *ill-posed problems*, of which this is a particular example. The parameter λ, that is usually called "regularization parameter", determines the trade-off between the level of the noise and the strength of the assumptions about the solution, therefore controlling the compromise between the degree of smoothness of the solution and its closeness to the data.

In the approach outlined here we have assumed to know the variance of the noise associated with each data point, but this assumption is not always realistic. Sometimes we know that some of the data can be affected by a high amount of noise, or can be completely wrong. In order to deal with this situation, we regard the variances of the noise, as well as the unknown function, as random variables. Of course, some a priori knowledge about these variables, represented by an appropriate a priori probability distribution, is needed. Let us denote by β the set of random variables $\{\beta_i\}_{i=1}^N$. By means of Bayes theorem we can compute the joint probability of the function f and of the set β:

$$\mathcal{P}[f, \beta|g] \propto \mathcal{P}[g|f, \beta]\ \mathcal{P}[f]\ \mathcal{P}[\beta] \tag{7}$$

where $\mathcal{P}[g|f, \beta]$ is the same as in equation (4) and $\mathcal{P}[\beta]$ is the a priori probability of the set of variances β. The model above, that leads to standard regularization, is recovered by setting

$$\mathcal{P}[\beta] = \prod_{i=1}^{N} \delta(\beta_i - \beta_i^*)$$

where β_i^* are some fixed values. Depending on the a priori knowledge on β different models may arise, corresponding to different choices of $\mathcal{P}[\beta]$. Here we consider the following situation: we have knowledge that a certain percentage, ϵ, of data is spurious (we will call them "outliers") whereas a percentage $(1 - \epsilon)$ is characterized by a Gaussian noise distribution of variance β^*. Therefore there are only two possibilities: $\beta_i = \beta^*$, for the "true" data points, and $\beta_i = 0$, for the outliers. This situation leads to choosing the following probability distribution:

$$\mathcal{P}[\beta] = \prod_{i=1}^{N}[(1-\epsilon)\delta(\beta_i - \beta^*) + \epsilon\ \delta(\beta_i)]\ . \tag{8}$$

Given the a posteriori probability (7) we are mainly interested in computing an estimate of f. Thus what we really need to compute is the marginal posterior probability of f, $P_m[f]$, that is obtained integrating equation (7) over the variables β_i:

$$P_m[f] = \int_0^\infty \prod_{i=1}^{N} d\beta_i\ \mathcal{P}[f,\beta|g]$$

Using the model for $\mathcal{P}[\beta]$ described by equation (8) we obtain:

$$P_m[f] \propto e^{-\lambda\|Pf\|^2} \prod_{i=1}^{N} \int_0^\infty dx e^{-x\Delta_i^2}[(1-\epsilon)\delta(x-\beta^*) + \epsilon\ \delta(x)].$$

The integral yields

$$P_m[f] \propto e^{-\lambda\|Pf\|^2} \prod_{i=1}^{N} [\frac{1-\epsilon}{\epsilon} e^{-\beta^*\Delta_i^2} + 1]$$

In order to make clear the meaning of such a marginal probability distribution we rewrite as:

$$P_m[f] \propto e^{-(\beta^* \sum_{i=1}^{N} V_{eff}(\Delta_i) + \lambda\|Pf\|^2)}$$

where we have defined the *effective potential*

$$V_{eff}(x) = x^2 - \frac{1}{\beta}\ln(1 + e^{(\beta x^2 - \gamma)})$$

and we have set $\gamma = \ln\frac{1-\epsilon}{\epsilon}$. The MAP estimate for f given by this probability distribution is obtained by minimizing the functional

$$H_m[f] = \beta^* \sum_{i=1}^{N} V_{eff}(\Delta_i) + \lambda\|Pf\|^2)\ . \tag{9}$$

The introduction the random variables β_i leads, therefore, to a new minimization problem. Let us compare the functionals (9) and (6). The functional

(9) is similar to the standard regularization functional (6), the only difference being in the data term. In the standard regularization functional the data term consists of the sum over all the data of a quadratic function V of the interpolation error Δ_i, and its role is to enforce closeness of the solution to the data. In the last case the quadratic function V has been substituted by the function V_{eff}, depicted in figures (1) and (2), whose shape depends on the parameters β^* and ϵ.

Figure (1) shows the effective potential for different values of ϵ, and for $\beta^* = 1.0$. In the case of $\epsilon = 0$ we obviosly recover the regularization model, since

$$\lim_{\epsilon \to 0} V_{eff}(x) = V(x) = x^2.$$

When ϵ is different from zero $V_{eff}(x)$ has two different behaviours: quadratic in a neighborhood of the origin, and constant far away from it. The effect of this behavior is clear: closeness to the data is enforced only when the interpolation error is small. In particular we notice that:

$$\lim_{x \to 0} V_{eff}(x) = 2(1 - \epsilon)x^2.$$

When ϵ increases and approaches 1 the effective potential becomes flatter and flatter, which is equivalent to the effective variance of the noise becoming larger and larger.

Let us consider the case of positive values of γ, that corresponds to values of ϵ smaller than 0.5. This is the usual case, since ϵ represents the percentage of "true" data points. (2) In the limit of $\beta^* \to \infty$ the effective potential V_{eff} is quadratic if the absolute value of its argument is smaller than $\sqrt{\gamma}$ and constant otherwise (fig. 2). This corresponds to the situation in which we have "true" data points without noise: therefore data points are considered reliable if the interpolation error is smaller than a threshold ($\sqrt{\gamma}$) and their contribution neglected otherwise. In the case of negative values of γ, which is the case of a percentage of outliers greater than 50%, the effective potential, that is already flat, becomes even flatter when β^* increases. This case is not very interesting and in the following we will always make the assumption that $\gamma > 0$, that is $\epsilon < 0.5$.

The standard regularization functional and the functional (9) admit a simple physical interpretation. Let us consider for simplicity a function defined on a one-dimensional lattice. The value of the function $f(\mathbf{x}_i)$ at site i

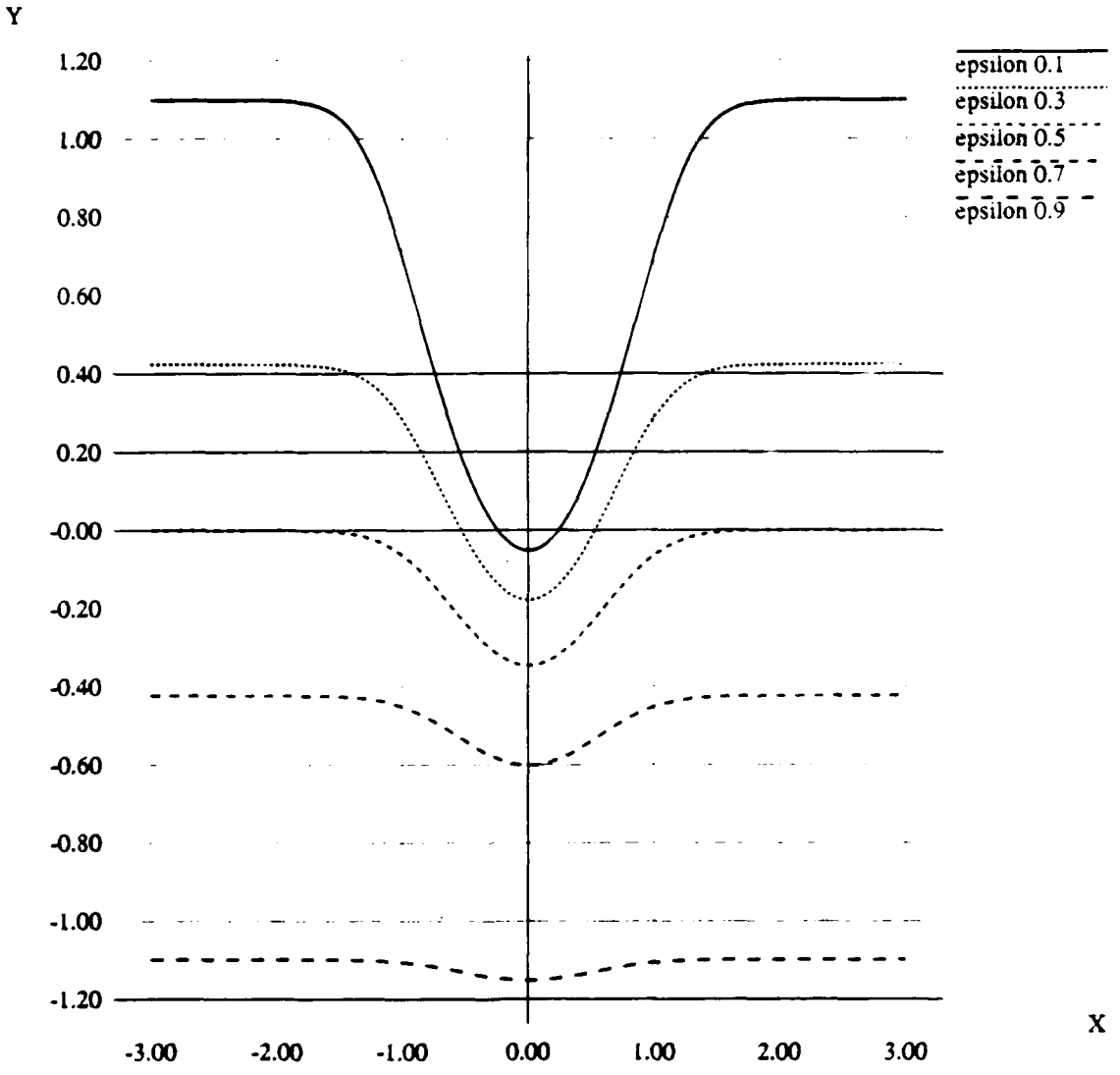


Figure 1: The effective potential for $\beta^* = 1$ and different values of ϵ.

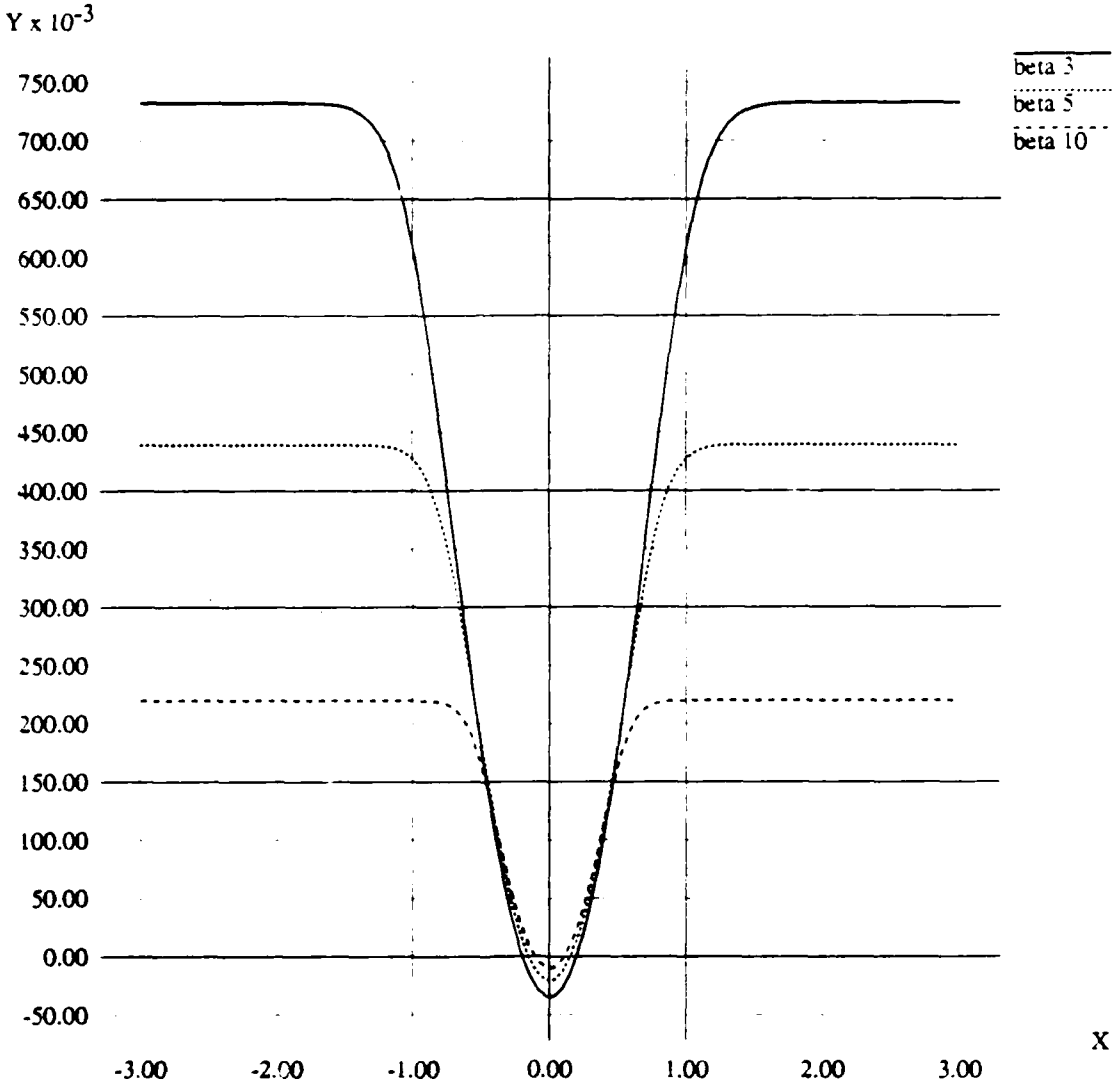


Figure 2: The effective potential for $\epsilon = 0.1$ and different values of β.

is regarded as the position of a particle that can move only in the vertical direction. The particle is connected by a spring to a point that corresponds to the data value y_i, and is also connected by springs to some neighboring particles. The size of the neighborhood can vary, but the overall effect is such that the values of the function at neighboring sites tend to be the same. The particle is attracted, with a quadratic potential, by the data point, but it is also attracted by the neighboring particles: the configuration of the system will be the one that minimizes the total energy, depending on the trade off between these two different effects. The energy of the system corresponds in this scheme to the standard regularization functional: the first term is associated to the springs connecting the particle to the data point, and the second term is associated to the the springs connecting neighboring particles, whose role is to enforce smoothness of the final configuration. The stabilizer is represented by the relative strength and the extension of the connections of the particles at neighboring sites: a stabilizer of high degree corresponds to a system in which a particle at a site is connected to particles at sites very far away.

The functional (9) admits a similar interpretation, the only difference being the kind of springs that connect the function value to the data point: in this case the potential energy of these springs is not quadratic anymore, that is the force associated to each spring does not grow linearly with its elongation. The potential energy becomes constant when the elongation is larger than the threshold ϵ, and the force (that is proportional to the first derivative of the potential energy) goes to zero. In a sense these springs break if we try to stretch them too much.

3 Negative examples

As we have seen in the previous section, standard regularization admits an interpretation in term of linear springs, whereas regularization in presence of unreliable data needs an interpretation in term of nonlinear springs, that break when the elongation is too large. Nonlinear springs have also been used to deal with discontinuities (Geiger and Girosi, 1989, 1990; Blake and Zisserman, 1987), and we show now another case in which they are very useful.

In many situations, further source of information about the function may

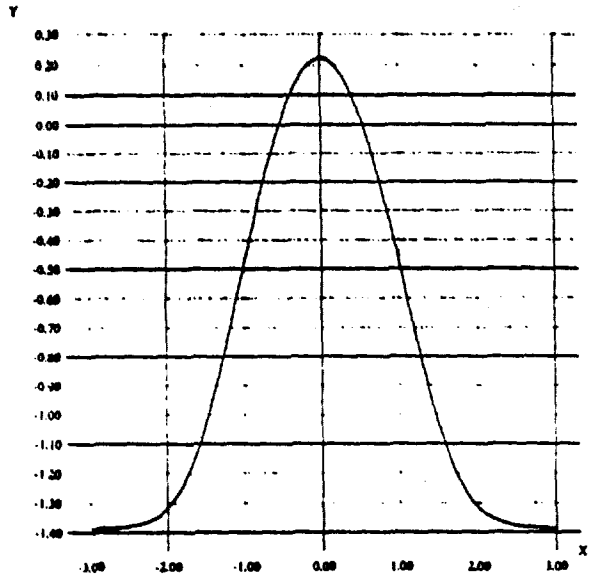


Figure 3: The potential associated to a repulsive spring for $\epsilon = 0.2$ and $\beta = 1$.

consist of knowing that its value at some given point has to be far from a given value (which, in this context, can be seen as a "negative example"). We shall account for the presence of negative examples by introducing a quadratic repulsive term (a sort of "repulsive" spring) in the regularization functional, one for each negative example (for a related trick, see Kass et al., 1987). However the introduction of such a term might make the regularization functional unbounded from below, because the repulsive spring will tend to push the value of the function up to infinity. The simplest way to prevent this occurency is to allow the spring constant to decrease with the increasing elongation, or in the extreme case, to break at some point. We can use the same model of nonlinear spring of the previous section, and just reverse the sign of the associated potential (see figure (3)).

If $\{(\mathbf{t}_i, y_i) \in R^n \times R\}_{i=1}^{K}$ is the set of negative examples, and if we define

$$\Delta_i = y_i - f(\mathbf{t}_i)$$

the regularizing functional can be written as

$$H[f] = \sum_{i=1}^{N} V(\Delta_i) - \sum_{\alpha=1}^{K} V_{eff}(\Delta_\alpha) + \lambda \|Pf\|^2 \ .$$

4 Solution of the variational problem

In this section we discuss the solution of the variational problem associated with the regularizing functionals of the previous sections. Since the cases of unreliable data and of negative examples are formally similar we will derive the equations only in the case of unreliable data. The functional to be minimized is

$$H_m[f] = \beta^* \sum_{i=1}^{N} V_{eff}(\Delta_i) + \lambda \|Pf\|^2) \ , \tag{10}$$

and the Euler-Lagrange equations for this functional have the form:

$$\hat{P} \ Pf(\mathbf{x}) = \frac{\beta^*}{2\lambda} \sum_{i=1}^{N} V'_{eff}(\Delta_i) \delta(\mathbf{x} - \mathbf{x}_i) \tag{11}$$

where $V'_{eff}(x)$ is the first derivative of $V_{eff}(x)$, that is

$$V'_{eff}(x) = \frac{2x}{1 + e^{\beta^* x^2 - \gamma}} \ .$$

We notice that the in the limit of $\epsilon \to 0$, that is in the case of springs that never break, γ goes to infinity and $V'(x) \to 2x$. In this case the standard regularization equations

$$\hat{P} \ Pf(\mathbf{x}) = \frac{\beta^*}{\lambda} \sum_{i=1}^{N} \Delta_i \delta(\mathbf{x} - \mathbf{x}_i) \tag{12}$$

are recovered. Equation (11) shows the same structure of that associated with the standard regularization case, and the solution can be derived using the Green function technique (Stakgold, 1979). As in the standard regularization case, (Poggio and Girosi, 1989) the solution will be a linear superposition of Green functions, one for each data point:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x}; \mathbf{x}_i) \tag{13}$$

where, in the general case,

$$c_i = \frac{\beta^* V'_{eff}(\Delta_i)}{2\lambda}.$$

We notice however that expression (13) is not the complete solution of the minimization problem. In fact all the functions that lie in the null space of the operator P are "invisible" to the smoothing term in the functional (10), so that the previous expansion is the solution *modulo* a term that lies in the null space of P. According to the considerations contained in section 1, in the following we will drop it from equations.

In order to find the vector $\mathbf{c}$ of coefficients c_i we substitute the expansion of equation (13) in the functional $H[f]$ defined in equation (10), that becomes a *function* $H^*(\mathbf{c})$ of the coefficients. Thus the vector $\mathbf{c}$ minimizes the function $H^*(\mathbf{c})$, which leads to the following set of equations:

$$\frac{\partial}{\partial c_k} H^*(\mathbf{c}) = 0 \quad k = 1, \ldots, N \tag{14}$$

Gradient descent is probably the simplest approach for attempting to find the solution to this minimization problem, though, of course, it is not guaranteed to converge. Several other iterative methods, such as versions of conjugate gradient and simulated annealing may be more appropriate than gradient descent, and their use is reccomended. In the gradient descent method the vector $\mathbf{c}$ that minimizes $H^*(\mathbf{c})$ is regarded as the stable fixed point of the following dynamical system:

$$\dot{\mathbf{c}} = -\omega \frac{\partial H^*(\mathbf{c})}{\partial \mathbf{c}} \tag{15}$$

where ω is a parameter determining the microscopic timescale of the problem and is related to the rate of convergence to the fixed point.

We consider for simplicity the case of positive definite Green's functions, that do not require any additional term in eq. (13). In this case it has been shown (Poggio and Girosi, 1989) that, with natural boundary conditions, we can write

$$\|Pf\|^2 = \mathbf{c} \cdot G\mathbf{c}.$$

where G is the symmetric matrix $(G)_{ij} = G(\mathbf{x}_i; \mathbf{x}_j)$ – its symmetry coming from the fact that the operator $\hat{P}P$ is self-adjoint.

Equations (15) have then the following form:

$$\dot{\mathbf{c}} = -\omega \frac{\partial}{\partial \mathbf{c}} \left[\beta^* \sum_{i=1}^{N} V_{eff}(\Delta_i) + \lambda \mathbf{c} \cdot G\mathbf{c} \right]$$

that, defining

$$\sigma_i = \frac{1}{1 + e^{-\beta(\epsilon - \Delta_i^2)}}; \quad (\Sigma)_{ij} = \sigma_i \delta_{ij}$$

can be written as

$$\dot{\mathbf{c}} = -2\omega G[(\beta^* \Sigma G + \lambda I)\mathbf{c} - \beta^* \Sigma \mathbf{y}] \tag{16}$$

wher I is the identity matrix. The vector $\mathbf{c}$ that mimizes $H^*(\mathbf{c})$ has then to satisfy the following set of *non linear* equations:

$$(\beta^* \Sigma G + \lambda I)\mathbf{c} = \beta^* \Sigma \mathbf{y}, \tag{17}$$

the non linearity being contained in the matrix Σ, that is a nonlinear function of the unknowns. Notice that

$$\lim_{\epsilon \to 0} \Sigma = I$$

and in this case the *linear* standard equations are recovered (Poggio and Girosi, 1989). The main implication of the nonlinearity is that the solution of these equations is not unique anymore, the different solutions corresponding to the local minima of the functional (10). Notice that it is straightforward to modify the previous gradient descent equations in order to take into account negative examples.

5 Experimental Results

In this section we describe some results that we obtained applying these techniques to very simple one-dimensional problems. We first discuss an

example unreliable data, and then a problem with negative examples. We used a gradient descent algorithm with adaptive step, running on a SUN 4 workstation. The code for these simulations has been written in Common Lisp, and in all the examples that we will describe in the next section, the time required for 100 iterations of the gradient descent algorithm was about 30 seconds. In the following figures data points are represented by large dots.

5.1 Unreliable data

We approximate the function $f(x) = \cos(x)$ in the interval $[-1, 1]$. The data set consisted of seven examples, randomly chosen from the graph of f. In order to create an outlier in the data set, we substituted the value of the fourth point with the value 1.5, that is 50% larger of the largest value of the other data points. The Green's function of the problem was a Gaussian of variance $\sigma = 0.3$, the parameter ϵ was set to 0.1, and the parameter β^* was set to 6. With this values of ϵ and β the effective potential was approximately constant for values of its argument larger than 1. In figure (4a) we show the result that is obtained applying standard regularization theory to approximate the data set. The value of the regularization parameter λ is 10^{-2}, and the result obtained after 200 iterations of the gradient descent algorithm is shown. The solution, that almost interpolates the data set, hardly resembles a cosine function, due to the outlier. If the springs are allowed to break, we obtain the result shown in figure (4b), after only 10 iterations of gradient descent: the spring of the outlier breaks, and the solution is not influenced by the outlier. Since the variance of the Gaussian Green's function is small ($\sigma = 0.3$) the solution has a "hole" in correspondence of the outlier, because there are no data there. A similar situation is shown in figures (4c) and (4d), the only difference being the value of the regularization parameter, that is ten times larger, that is $\lambda = 10^{-1}$. We notice that, since the Green's function is bounded, increasing the regularization parameter has the effect of decreasing the norm of the solution (see Poggio and Girosi, 1989). This effect is evident when comparing figures (4a) and (4b) with figures (4c) and (4d).

5.2 Negative examples

In order to test the negative example technique we choose again, as a function to be approximated, the cosine function $f(x) = \cos(x)$, randomly

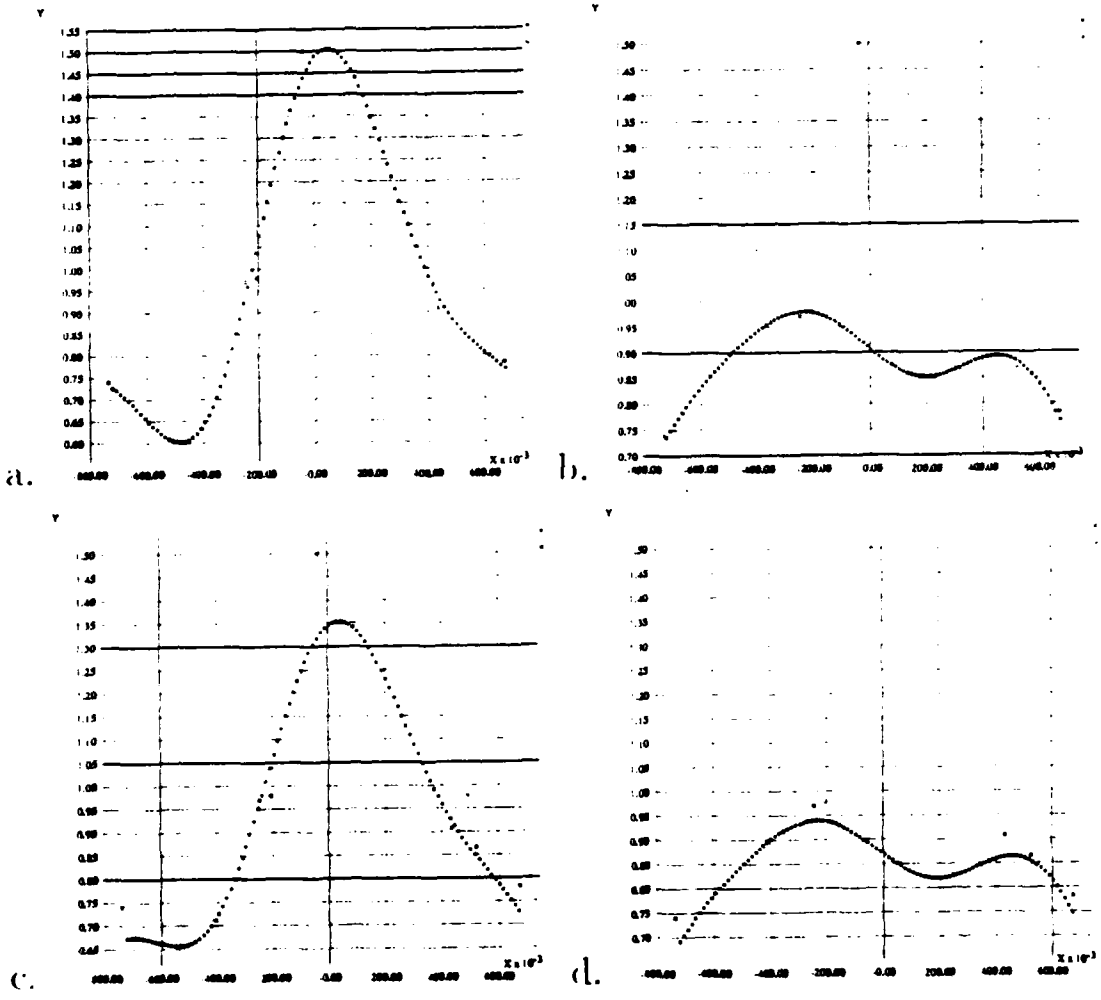


Figure 4: Approximation in presence of an outlier (the data point whose value is 1.5). Comparisons between standard regularization (4a) and (c)) and the extension introduced here (4b) and (d)). See text for explanation.

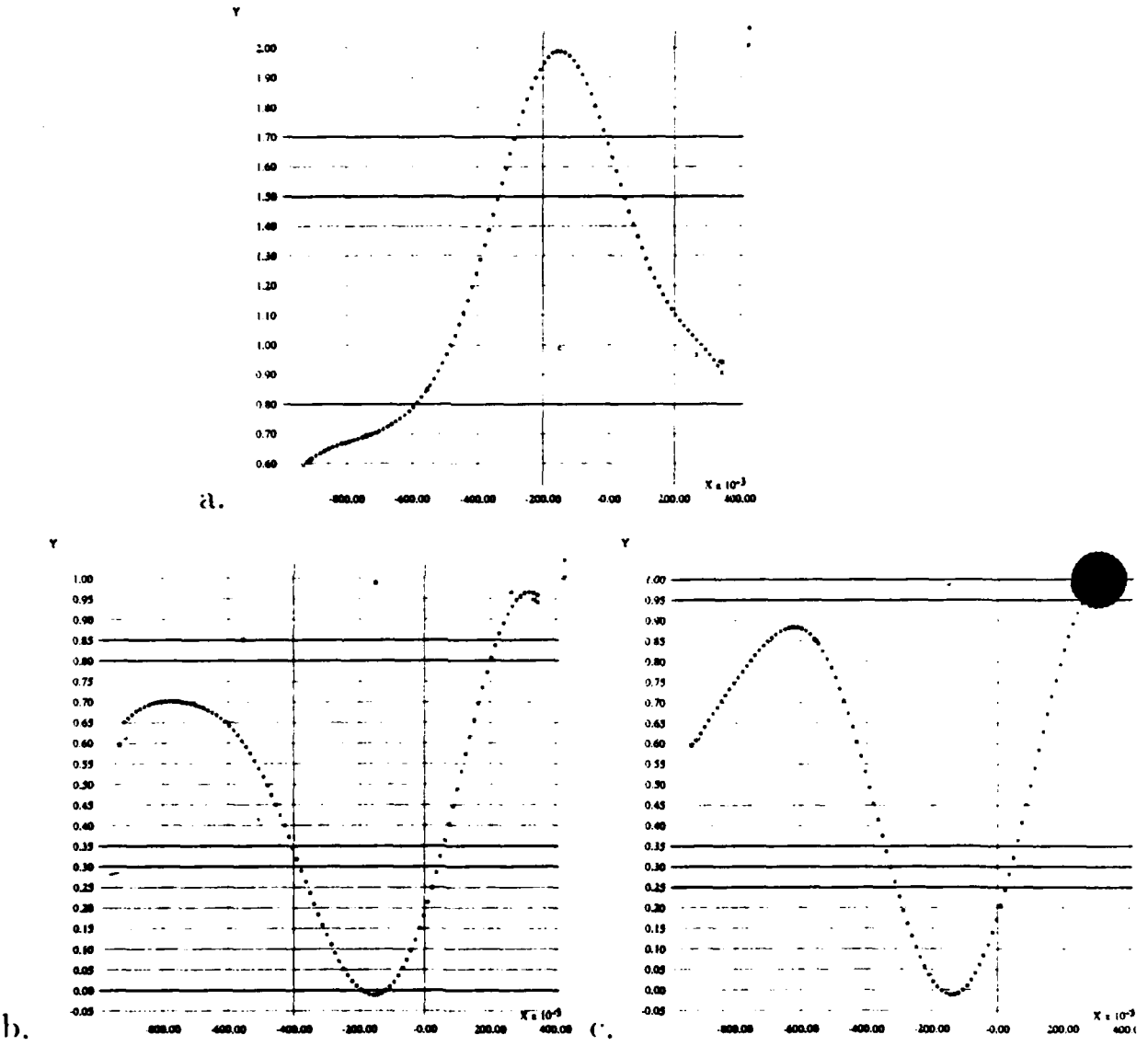


Figure 5: Negative examples.(a) and (b): the configurations corresponding to different minima of the functional. (c): the effect of increasing the attractiveness of the standard springs. See text for explanation

sampled at seven points in the interval $[-1, 1]$. In all the experiments the regularization parameter was set to zero, its role not being crucial in this case. The Green's functions we used were always Gaussians, with variance different from case to case. The fourth data point, whose coordinates were $(x, y) = (-0.15, 0.99)$, was selected as the negative example, and the parameters β^* and ϵ was the same as in the previous case, so that the springs could break if the elongation were larger than 1. This meant that the result had to be a function $f^*(x)$ that approximates the six "positive" examples, but such that $|f^*(-0.15) - 0.99| \geq 1$. There are clearly two possibilities: $(f^*(-0.15) - 0.99) \geq 1$ and $(0.99 - f^*(-0.15)) \geq 1$, corresponding to functions "passing above and below the negative example". These configurations corresponds to two different minima of the functional, and we expect to obtain one of these two configurations depending on the initial conditions of the gradient descent algorithm.

In figure (5a) and (5b) we show two results corresponding to two different local minima. Convergence was reached in 50 iterations, and in both cases the variance of the Gaussian is $\sigma = 0.2$. In figure (5a) we set as initial condition $c_i = y_i$, and in figure (5b) we set $c_i = 0.0$. In the first case the initial condition corresponds to a function that is "above" the data, while in the second case the initial function is zero everywhere, and then "below" the data. In the first case the final value of the "energy" of the system was $H = -0.996$, that is very close to the global minimum energy $H = -1.0$, while in the second case the energy was $H = -0.931$. Interpreting these results in terms of springs, it is evident, in figure (5b), that the spring on the left of the negative example is not sufficiently strong to pull up the solution to the datum. We then changed the elastic constant of this spring and of the corresponding one on the right of the negative example, setting their values to 10, that is ten times larger than the other ones. The result is shown in figure (5c), and it is clearly better than the one shown in figure (5b), its associated energy being $H = -0.995$, that is comparable with the value $H = -0.996$ of figure (5a).

From the previous result and many other experiments it is apparent that the energy landscape associated with this minimization problem could be very complicated, with many local minima corresponding to the two types of configurations ("above" and "below"). It is natural to ask whether during the gradient descent iterations the system naturally "jumps" from one of these configurations to the other one. The answer is given in figures (6a) and

(6b). In figure (6a) we show the configuration of the system corresponding to the iterations 1, 30, 31 and 40 of the gradient descent algorithm. The variance of the Gaussian Green's function is $\sigma = 0.8$, and the starting point of the descent procedure is $c_i = 0.0$. At the beginning the configuration is of type "below", because it is identically zero, and then it stabilizes around an interpolating function until iteration 30. At iteration 30 the system jumps in a configuration of type "above", whose energy is much lower, and then converges rapidly to a local mimimum. In figure (6b) the energy of the system is shown as function of the number of iterations: notice the jump at iteration 30, that probably corresponds to a discontinuity of the gradient of the energy surface.

In order to escape local minima we used a simple form of stochastic gradient descent, adding a white noise term to eq. (15). The noise term was used only to get out of local minima, that is it was switched on only when the energy decreased, from one iteration to the next one, of an amount lower than a small threshold (usually 10^{-8}). The usefulness of the noise is shown in figures (7a) and (6b). The data are the same of figures (6) and (5), but the break point of the spring of the negative example has value 1, the variance of the Gaussian Green's function is $\sigma = 0.4$ and the amplitude of the noise is 10^{-2}. In figure (7a) we show the result of the gradient descent algorithm without noise. Convergence was obtained after 25 iterations, and the result is not very good, corresponding to some local minimum. In figure (7b) we show the result of the stochastic gradient descent algorithm after 1000 iterations: the local minima have been escaped, and the result is almost a perfect interpolant on the "positive" examples.

Interesting effects take place if we raise the amplitude of the noise. In figure (8a), (8b) and (8c) we show what happens if, in the previous example, we set the amplitude of the noise to 10^{-1}, instead of 10^{-2}. The results of the stochastic descent procedure are shown at iterations 200, 500 and 2000. We notice that the system jumps from a configuration of type "below" to a configuration of type "above" and then to a configuration of type "below" again. This suggests that there are several local minima, and the noise makes the system jumping from one to another. In figure (8d) the energy of the system as function of the number of iterations is shown: notice that the algorithm does not inject the noise continuosly, but only when the energy stops decreasing.

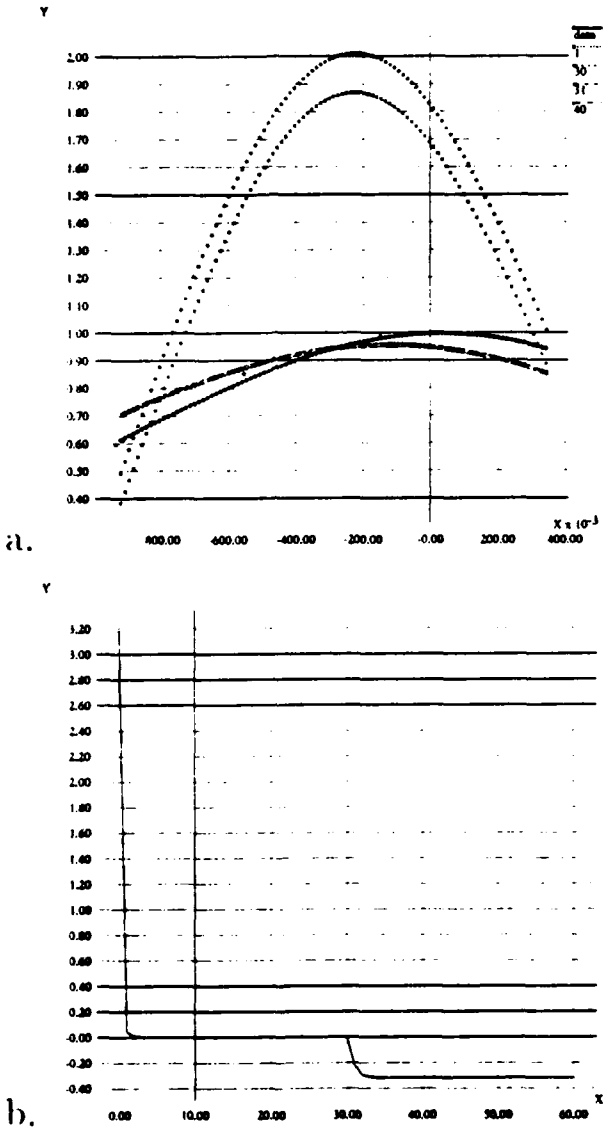


Figure 6: Negative examples. (a) Several configurations of the system are shown, while equilibrium is being approached. (b) The learning curve.

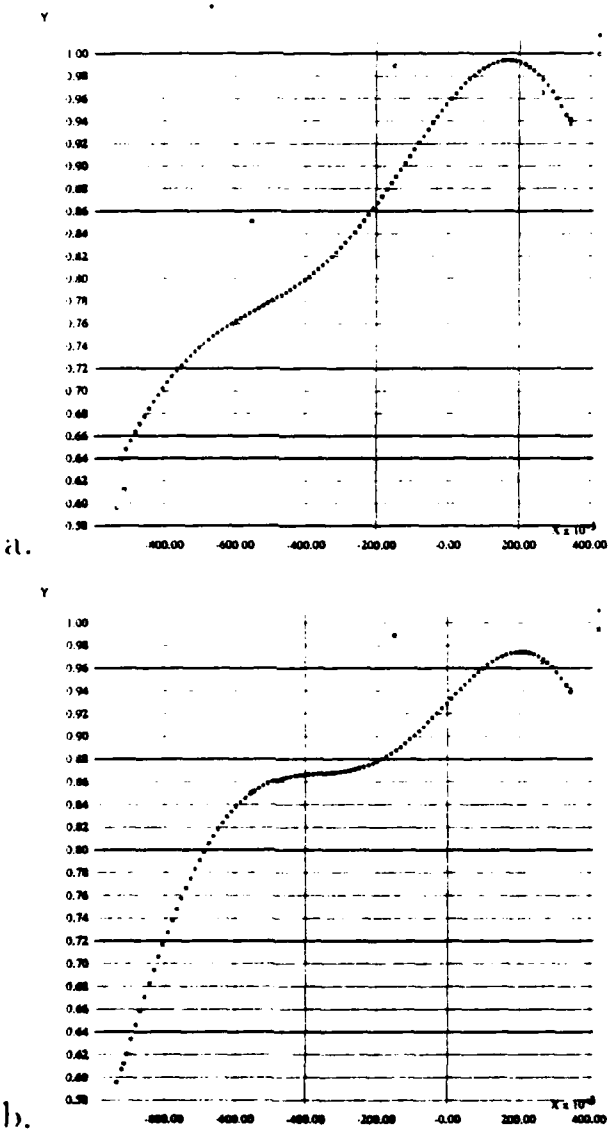


Figure 7: Negative examples. Results obtained without (a) and (b) noise.

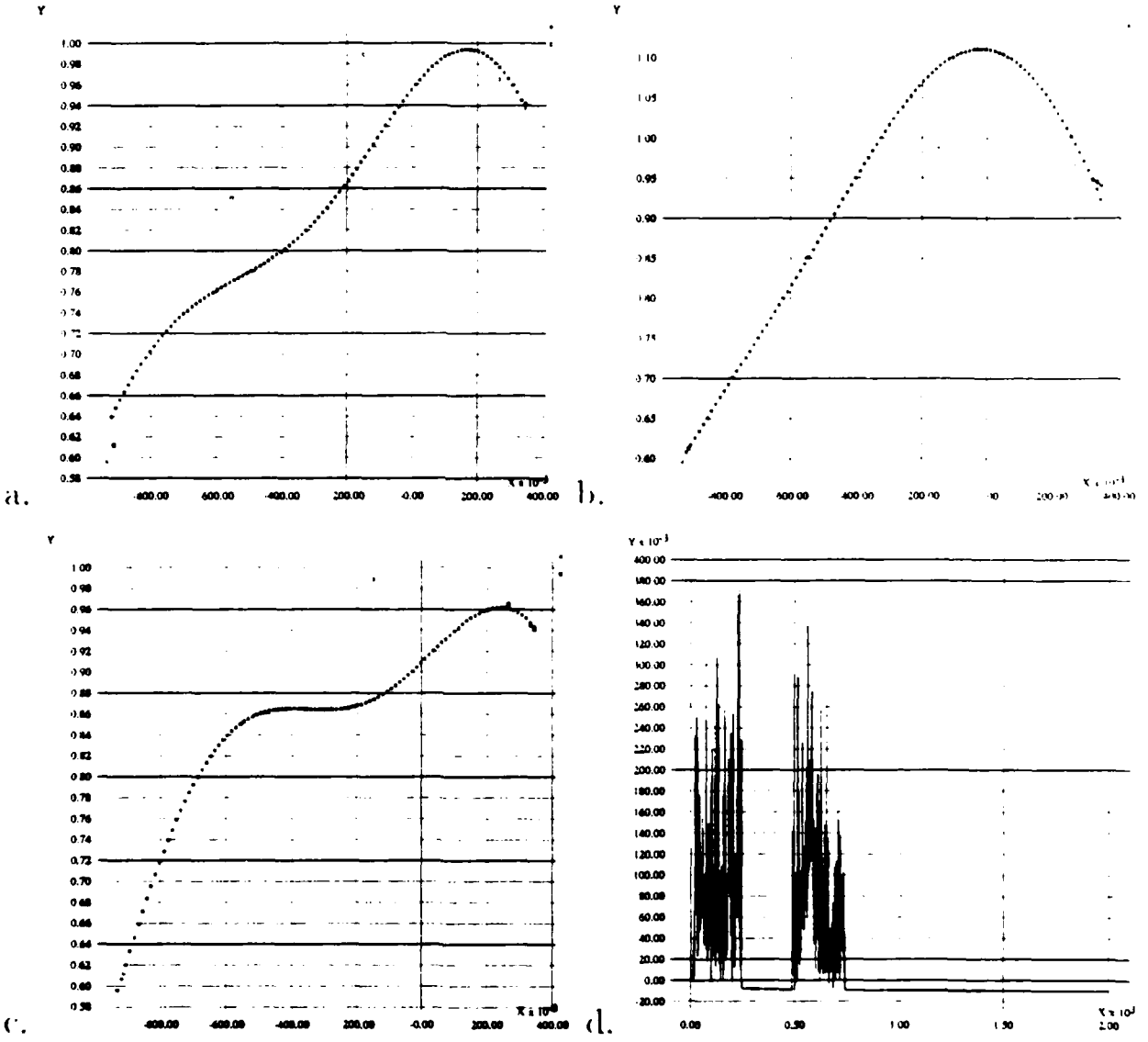


Figure 8: Negative examples. (a) (b), (c): the effect of an high level of noise: the system jumps from one local minima to an other one. (d): the learning curve.

6 Remarks

1. The first extension we have introduced here – to deal with unreliable data – may be important in problem of the type of surface reconstruction, as one encounters in computer vision. It may or may not be useful in problems of learning from examples.

2. The second extension – to exploit negative examples – is especially interesting for the problem of learning, where often negative examples are present (though they usually are less important than the positive ones). In some cases it may be useful also in problems of approximation of functions. There are situations in which one knows that certain regions of the range of the function are forbidden. Interestingly, this type of problems seems to have been ignored in the classical approach to function approximation (see Verri and Poggio, 1988 for related, simpler and more classical cases). The functional we considered, and then the type of spring we used, is feasible of further modifications, according to the a priori knowledge about the system. For example, the constraint that the values of a one dimensional function are bounded from above (and/or below) can be included using springs that are negative from one side and positive from the other side.

3. In both the extensions that we have presented the solution has the form (13), which has a simple interpretation in terms of feedforward networks with one layer of hidden units, of the same class of the regularization networks introduced in previous papers (Poggio and Girosi, 1989; 1990).

Acknowledgements We thank Cesare Furlanello for useful discussions and for a critical reading of the manuscript.

References

[1] M. Bertero. Regularization methods for linear inverse problems. In C. G. Talenti, editor, *Inverse Problems*. Springer-Verlag, Berlin, 1986.

[2] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, Mass., 1987.

[3] D. Geiger and F. Girosi. Parallel and deterministic algorithms for MRFs: surface reconstruction and integration. A.I. Memo No. 1114, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1989.

[4] D. Geiger and F. Girosi. Parallel and deterministic algorithms for MRFs: surface reconstruction and integration. In O. Faugeras, editor, *Lecture Notes in Computer Science, Vol. 427: Computer Vision - ECCV 90.* Springer-Verlag, Berlin, 1990.

[5] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:721–741, 1984.

[6] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the First International Conference on Computer Vision*, London, 1987. IEEE Computer Society Press, Washington, D.C.

[7] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:219–227, 1983.

[8] J. L. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *J. Amer. Stat. Assoc.*, 82:76–89, 1987.

[9] V.A. Morozov. *Methods for solving incorrectly posed problems.* Springer-Verlag, Berlin, 1984.

[10] T. Poggio and F. Girosi. A theory of networks for approximation and learning. A.I. Memo No. 1140, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1989.

[11] T. Poggio and F. Girosi. A theory of networks for learning. *Science*, 247:978–982, 1990.

[12] I. Stakgold. *Green's functions and boundary problems.* John Wiley and Sons, New York, 1979.

[13] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

[14] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems.* W. H. Winston, Washington, D.C., 1977.

[15] A. Verri and T. Poggio. Regularization theory and shape constraints. A.I. Memo No. 916, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1986.