Some Aspects of a Complexity Theory for Continuous Time Systems

Marco Gori[1] and Klaus Meer[2,⋆]

[1] Dipartimento di Ingegneria dell'Informazione
Università di Siena, Via Roma 56, I-53100 Siena, Italy
[2] Syddansk Universitet Odense, Dept. of Mathematics and Computer Science
Campusvej 55, DK-5230 Odense M, Denmark

Abstract. In this paper we survey previous work by the authors defining a complexity measure for certain continuous time systems. Starting point are energy functions of a particular structure. Global minimizers of such energies correspond to solutions of a given problem, for example an equilibrium point of an ordinary differential equation. The structure of such energies is used to define complexity classes for continuous problems and to obtain completeness results for those classes. We discuss as well algorithmic aspects of minimizing energy functions.

1 Introduction

The use of analog systems as computational models has attracted increasing interest in recent years. One way to formalize computation in this framework is to consider a differential equation and follow a trajectory until a solution, e.g., an equilibrium point, is reached. There are many interesting and open problems related to such an approach, ranging from the question of setting up a complexity theoretic framework for such dynamical systems (including notions of complexity classes, reducibility, completeness etc.) to concrete solution algorithms. For an excellent up to date survey on related questions see [3] and the literature cited in there. An older yet very readable survey is [6].

In this paper we discuss a general framework for measuring the complexity of analog systems introduced in [4].

Based on the notion of a *problem* we define complexity classes in dependence of the structural complexity of certain energy functions. Those functions are related to the solutions of a problem instance through their global minimizers. This gives a way to introduce complexity classes which mimic classical P and NP as well as the polynomial hierarchy, and to obtain completeness results.

Both the strength and weakness of this approach may be are its abstractness. On the negative side one might expect a complexity theory for continuous time

⋆ Partially supported by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778 and by the Danish Agency for Science, Technology and Innovation FNU. This publication only reflects the author's views.

S.B. Cooper, B. Löwe, and A. Sorbi (Eds.): CiE 2007, LNCS 4497, pp. 554–565, 2007.

systems to be more concrete. However, on the positive side the approach is not based on how to specify a complexity measure for following a trajectory.

Section 2 recalls the definition of a problem and introduces the above mentioned complexity classes in our model. We then discuss the main results concerning completeness for the introduced classes. In Section 3 we outline how the approach can be made more concrete by adding as well a measure for following trajectories. As examples we consider linear system solving and the perceptron learning algorithm.

Proof details can be found in [4].

2 The General Framework

A problem in our setting is defined as a binary relation over the space $\mathbb{R}^\infty := \bigoplus_{i\geq 1} \mathbb{R}^i$ of finite sequences of real numbers.

Definition 1. *A* PROBLEM *Π is a relation in $\mathbb{R}^\infty \times \mathbb{R}^\infty$.* SOLVING A PROBLEM *means that on input $d \in \mathbb{R}^n$ for some $n \in \mathbb{N}$, a vector $y \in \mathbb{R}^k$ for some $k \in \mathbb{N}$ is computed such that $(d, y) \in \Pi$. Usually we require the output dimension k to be polynomially related to n, i.e. there exists a polynomial p such that $k = p(n)$ for all $n \in \mathbb{N}$.*

Remark 1. We shall frequently use the notation $\Pi(d)$ to denote a solution y such that $(d, y) \in \Pi$, even though Π may not be a function.

Example 1. The following examples are typical for our framework:

a) The problem of solving linear equations is given by

$$\Pi := \{(A, b, y) | A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, y \in \mathbb{R}^n \text{ such that } A \cdot y = b\}.$$

In terms of Definition 1 we thus have $d := (A, b)$.

b) The problem of finding a separating hyperplane for two classes $X^+ \subset \mathbb{R}^n$, $X^- \subseteq \mathbb{R}^n$ of patterns is given by

$$\Pi := \{(X^+, X^-, w, \delta) | w^T x \geq \delta \ \forall\, x \in X^+, w^T x \leq -\delta \ \forall x \in X^-\}.$$

Again, in terms of Definition 1 it is $d := (X^+, X^-)$ and $y := (w, \delta)$.

c) Consider a dynamical system $\frac{dx}{dt} = F(x(t))$ for which an equilibrium point is searched. This can be formalized in different ways. One appropriate possibility is to consider $d := F$ as the first component of a problem Π and to look for a y which is an equilibrium point of F. Thus

$$\Pi = \{(F, y) \in \mathbb{R}^\infty \times \mathbb{R}^\infty \mid y \text{ is an equilibrium of } \frac{dx}{dt} = F(x(t))\}\ .$$

The above definition requires F to be representable in a certain way as a point in $\mathbb{R}^\infty$. This is, for example, the case if F is given as a rational function.

We are interested in characterizing the complexity of a problem through the structure of certain energy functions in the following sense. Such energies are associated to the problem in a uniform way by considering a family $\{E_n\}_{n\in\mathbb{N}}$ of functions for each dimension n. Every E_n is a function depending on two blocks d and w of variables. The block $d \in \mathbb{R}^n$ is taken to represent an input instance of a problem Π. The block $w \in \mathbb{R}^m$ will be related to a solution y of Π for input d. Once more, the dimension m should be polynomially related to n, that is $m = q(n)$ for a polynomial q. This polynomial is given together with the family $\{E_n\}$.

Moreover, for each n and fixed $d \in \mathbb{R}^n$ the function $w \to E_n(d,w)$ is supposed to have a global minimum. Such a minimum w^* can be used by an additional (computationally easy to perform) algorithm to yield a solution of the particular instance d for our problem.

Another main point of the definition is the way how these energies can be computed. This will be crucial for defining a complexity measure later on. We use straight-line programs for this purpose.

Definition 2. *(Straight-line programs) For an operation set $\mathcal{O}$ a* STRAIGHT-LINE PROGRAM *of input dimension T over $\mathcal{O}$ is a sequence $\beta_1, \dots, \beta_\ell$ of operations defined as follows. Every β_i is either of the form $\beta_i := c$ for a constant $c \in \mathbb{R}$ or of the form $\beta_i := \beta_j \circ \beta_k$, where $\circ \in \mathcal{O}$ and $j,k \in \{-T+1, \dots, 0, 1, \dots i-1\}$. For any T-dimensional real input $x_1, \dots, x_T$, to the first T registers $\beta_{-T+1}, \dots, \beta_0$ we assign the values $\beta_{-T+1} := x_1, \beta_{-T+2} := x_2, \dots, \beta_0 := x_T$. A computation of the program then proceeds in the obvious manner assigning the corresponding values to the $\beta_i, i \geq 1$. The result is supposed to be computed in β_ℓ. The* SIZE *or* COMPUTATION TIME *of the program is the number ℓ of operations performed.*

In our framework, input variables for an SLP are chosen as $d_1, \dots, d_n$ and $w_1, \dots, w_{q(n)}$. Thus, $T = n + q(n)$. We are interested in computing a real valued function $E_n : \mathbb{R}^{n+q(n)} \to \mathbb{R}$. We want to combine SLPs in a uniform way in order to relate them to functions from $\mathbb{R}^\infty \to \mathbb{R}^\infty$.

Definition 3. *A family $\mathcal{E} := \{E_n\}_{n\in\mathbb{N}}$ of SLPs, every E_n of input dimension $n + q(n) \in \mathbb{N}$ for a fixed polynomial q, is called to be* UNIFORMLY POLYNOMIALLY BOUNDED *if there exists an algorithm which on input $n \in \mathbb{N}$, computes a description of E_n and runs in polynomial time with respect to n. We call such a family an SLP energy family.*

In this paper we restrict the operation set $\mathcal{O}$ to be $\{+, -, *\}$, but more general sets are thinkable, see [4]. Thus, our energies basically are multivariate polynomials and can be treated in the framework of the BSS model of computation, see [2]. We suppose the reader to be familiar with this model.

2.1 Complexity Classes

For defining the complexity of a problem we now look for the structure of related families of energy functions. At this level the complexity of a problem will be independent of the question how to find a global minimum of an energy. The latter problem is addressed in the next section.

Next, we introduce certain complexity classes denoted by $\mathbf{U}, \mathbf{NU}$, and $\mathbf{PU}$ that are relevant in our framework.

Definition 4. *Let Π be a problem.*

a) Π belongs to the class $\mathbf{U}$ if there exists an SLP energy family $\{E_n\}_n$ together with another family $\{N_n\}$ of SLPs which is uniformly given by a polynomial time BSS machine such that the following is true:
  i) There is a fixed polynomial q such that every E_n is a map $E_n : \mathbb{R}^n \times \mathbb{R}^{q(n)} \to \mathbb{R}$;
  ii) For any fixed $d \in \mathbb{R}^n$ the function $w \to E_n(d,w)$ is unimodal;
  iii) If w^* is a global minimizer of $w \to E_n(d,w)$ for given d, then we can compute a solution $\Pi(d)$ using the SLP $N_{q(n)}$, i.e. $(d, N_{q(n)}(w^*)) \in \Pi$.

b) Problem Π belongs to class $\mathbf{NU}$ if items i) and iii) above hold, but $w \to E_n(d,w)$ has not to be unimodal. Clearly, it is $\mathbf{U} \subseteq \mathbf{NU}$.

c) Π belongs to the continuous-time polynomial hierarchy $\mathbf{PU}$ if the following holds: there exist an SLP energy family $\{E_n\}_n$ and a function $N : \mathbb{R}^\infty \to \mathbb{R}^\infty$ computable by a uniform family of SLPs in polynomial time such that:
  i) There is a fixed polynomial q such that every E_n is a map $E_n : \mathbb{R}^n \times \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \ldots \times \mathbb{R}^{n_k} \to \mathbb{R}$, where $\sum_{i=1}^{k} n_i = q(n)$.
  ii) If for given d the point w_1^* is a solution for the choice of variables w_1 in the optimization problem

$$\min_{w_1} \max_{w_2} \ldots \min_{w_k} \; E_n(d, w_1, \ldots, w_k),$$

  then we can compute a solution of Π for input d as $(d, N(w_1^*))$. The same should hold w.r.t. every $\widehat{w}_1$ such that $\max_{w_2} \ldots \min_{w_k} \; E_n(d, \widehat{w}_1, \ldots, w_k)$ does not exist. Above, the last optimization operation is min *if k is odd and* max *if k is even. The problem belongs as well to $\mathbf{PU}$ if the optimization starts with* max .

The classes $\mathbf{U}$ and $\mathbf{NU}$ can be seen as a natural counterparts of $\mathbf{P}$ and $\mathbf{NP}$ in our framework. We thus conjecture the obvious inclusion $\mathbf{U} \subset \mathbf{NU}$ to be proper.

In order to speak about complete problems finally the following definition is needed.

Definition 5

a) Let Π_1 and Π_2 be two problems. We say that Π_1 is SLP-reducible in polynomial time *to Π_2 if there exist two functions ϕ and ϕ^* from $\mathbb{R}^\infty \to \mathbb{R}^\infty$, both computable in polynomial time by a uniform SLP in the BSS model of computation, such that*

$$\forall d \in \mathbb{R}^\infty \quad \Pi_1(d) = \phi^* \left(\Pi_2 \left(\phi(d)\right)\right)$$

  Note that since $\Pi_2 \left(\phi(d)\right)$ might not be unique (cf. Remark 1) we require Φ^* to compute a solution of $\Pi_1(d)$ for any possible value of $\Pi_2 \left(\phi(d)\right)$.

*b) A problem $\Pi \in \mathbf{NU}$ is $\mathbf{NU}$-*complete *if every other problem in $\mathbf{NU}$ is SLP-reducible in polynomial time to Π. Similarly for $\mathbf{PU}$-*completeness.

2.2 Completeness Results

The following results show the existence of complete problems for **NU** and **PU**. For proofs we refer to [4].

Theorem 1

a) There exist **NU***-complete problems with respect to the operation-set* $\mathcal{O} := \{+,-,*\}$ *and SLP-reducibility.*

b) The following quadratic optimization problem is **NU***-hard with respect to the operation set given in a): Given a linear objective function* $f : \mathbb{R}^n \rightarrow \mathbb{R}$ *together with finitely many constraints* $h_1(x) = 0, \ldots, h_m(x) = 0, x \in \mathbb{R}^n$*, where the* h_i *are polynomials of degree at most* 2*, find a solution point of* $\min\{f(x)|h_i(x) = 0, 1 \leq i \leq m\}$*. Thus,* **QP** *is the problem defined by tuples* $(f, h_1, \ldots, h_m, x)$ *such that* x *is a global minimizer of the constrained optimization problem* $\min f(x)$ *subject to* $h_i(x) = 0, 1 \leq i \leq m$*.*

For the polynomial hierarchy it can be shown

Theorem 2. *The following problem is* **PU***-complete under SLP reductions: Given a polynomial* f *of degree* 4 *in* n *blocks* $w, X_2, \ldots, X_n$ *of variables, compute a minimizer of the function*

$$w \rightarrow \max_{X_2} \min_{X_3} \ldots \max_{X_n} f(w, X_1, \ldots, X_n)$$

Again, the last optimization operation is min *if* n *is odd and* max *if* n *is even.*

To get a rough idea of how the proof works consider a problem in **NU** with an input d and attached energy $w \rightarrow E_n(d, w)$. The decision problem: Given z and d, does z minimize $w \rightarrow E_n(d, w)$ is in co-NP$_{\mathbb{R}}$ over the reals. Using the common reduction arguments from [2] one realizes that by means of a max-min problem such a minimizer can be found (if existing). In general, a related argument establishes the problem in the statement to be **PU**-complete by finding a reduction of a given problem in the hierarchy to the former that increases the number of alternations of max and min by one.

We consider it to be important in our framework to extend the list of complete problems.

3 A Discretization for Trajectory Following: Guiding Examples

The framework analyzed above results in a split of the relevant parts contributing to the complexity of continuous time problems. The first deals with the structure and the SLP complexity of an underlying energy function. As mentioned in the introduction, this is advantageous in that the approach is independent of concrete measures for following trajectories of the given ODE. On the other side, this approach remains abstract when it comes to approximating concrete solutions. Thus, a second part contributes to the complexity of a problem. There is so far no real consensus about what a unifying approach for measuring this second part should be. For a much deeper discussion of different approaches we refer once more to [3].

In this final section we discuss briefly one natural such attempt. It adds a typical measure for the steepest descent algorithm finding minima of energy functions occuring in the above approach. We outline the resulting complexity statements for the first two problems of Example 1. Not surprising, the results obtained that way resemble well known properties for numerical solutions of those problems. Since most of the calculations are quite standard we just outline them.

Let $\{E_n\}_n$ be an SLP energy family as above. W.l.o.g. we assume that 0 is the minimal value for each E_n. For the moment we fix the input $d \in \mathbb{R}^n$ and suppress it and its size n notationally, i.e., instead of $E_n(d,w)$ we write $E(w)$. Choosing a start value w_0 we consider (see [8] for more on terminal dynamics) the terminal attractor equation

$$\frac{d}{dt}w(t) \;=\; -\frac{E_0}{\sigma}\cdot\frac{D_wE(w(t))}{\|D_wE(w(t))\|^2}\;, \tag{1}$$

where $E_0 := E(w_0)$ and $\sigma > 0$ is fixed, together with its Euler-discretization

$$w_{k+1} \;=\; w_k - \tau\cdot\frac{E_0}{\sigma}\cdot\frac{D_wE(w_k)}{\|D_wE(w_k)\|^2} \tag{2}$$

with step size $\tau > 0$. It is easy to see that the exact solution $\tilde{w}(t)$ of (1) for $t < \sigma$ satisfies

$$E(\tilde{w}(t)) = E_0\cdot(1-\frac{t}{\sigma}), t \in [0,\sigma).$$

Thus, the terminal attractor approaches a point with energy value 0 in finite time $t = \sigma$. The following theorem addresses the complexity of the discretization procedure (2).

We shall first study the number of discretization steps necessary in order to achieve a point w^* such that the energy value satisfies $E(w^*) \le \epsilon$ for a given precision $\epsilon > 0$. The theorem below is proven by a straightforward calculation using Taylor's formula.

Theorem 3. *Let E be an energy function as above and let E_0 denote the energy value for a starting point w_0 (see below). For a pair $(\epsilon_1,\epsilon_2), \epsilon_1 > \epsilon_2 > 0$ consider the differential equation*

$$\frac{d}{dt}w(t) \;=\; -\frac{E_0}{\sigma}\cdot\frac{D_wE(w(t))}{\|D_wE(w(t))\|^2}\;,$$

and its Euler-discretization

$$w_{k+1} \;=\; w_k - \tau\cdot\frac{E_0}{\sigma}\cdot\frac{D_wE(w_k)}{\|D_wE(w_k)\|^2}$$

Suppose that there exist bounds $L(\epsilon_1,\epsilon_2)$ and $H(\epsilon_1,\epsilon_2)$ on the set $\Omega(\epsilon_1,\epsilon_2) := \{\tilde{w} \in \Omega | \epsilon_1 \ge E(\tilde{w}) \ge \epsilon_2\}$ such that:

(i) $\|D_w E(\tilde{w})\| \geq L(\epsilon_1, \epsilon_2) > 0$ *for all* $\tilde{w} \in \Omega(\epsilon_1, \epsilon_2)$ *as well as*
(ii) $\|D_w^2 E(\tilde{w})\| \leq H(\epsilon_1, \epsilon_2)$ *for all* $\tilde{w} \in \Omega(\epsilon_1, \epsilon_2)$*, where the norm is the operator norm corresponding to the Euclidean vector norm.*

Then starting from a point w_0 *in* $\Omega(\epsilon_1, \epsilon_2)$ *a point* w^* *such that* $E(w^*) \leq \epsilon_2$ *can be reached in*

$$k(\epsilon_1, \epsilon_2) := O\left(\frac{\epsilon_1^2 \cdot H(\epsilon_1, \epsilon_2)}{\epsilon_2 \cdot L(\epsilon_1, \epsilon_2)^2}\right)$$

many discretization steps of step size

$$\tau(\epsilon_1, \epsilon_2) = O\left(\frac{\epsilon_2 \cdot L(\epsilon_1, \epsilon_2)^2 \cdot \sigma}{\epsilon_1^2 \cdot H(\epsilon_1, \epsilon_2)}\right).$$

After having reduced the energy value below ϵ_2 we let ϵ_2 play the role of ϵ_1 and choose a new target value for the energy. This results in considering a sequence $\{\epsilon_k\}_{k\in\mathbb{N}}$ which is used in the steepest descent algorithm until we have reduced the energy below a given accuracy ϵ. The number of iteration steps then is given by

$$\sum_{k=1}^{K^*} \frac{1}{2} \cdot \frac{H(\epsilon_{k-1}, \epsilon_k) \cdot \epsilon_{k-1}^2}{L(\epsilon_{k-1}, \epsilon_k)^2 \cdot \epsilon_k} \, , \tag{3}$$

where $\epsilon_0 := E_0$ is the first energy value we start with. The goal now is to determine how this sum depends on the required accuracy ϵ and to choose reasonable sequences $\{\epsilon_k\}$ to keep it as small as possible.

A typical choice is given by $\epsilon_k := \frac{E_0}{2^k}$. Then K^* in the above formula has to be taken such that $\frac{E_0}{2^{K^*}} \leq \epsilon$ for a given precision $\epsilon > 0$, i.e. $K^* := \lceil \log(\frac{E_0}{\epsilon}) \rceil$.

Remark 2.
(a) It should be clear from the above arguments that the ratio $\kappa(E, \epsilon_{k-1}, \epsilon_k) := \frac{H(\epsilon_{k-1}, \epsilon_k)}{L(\epsilon_{k-1}, \epsilon_k)^2}$ can be interpreted as a condition number for the problem of minimizing the energy on the set $\Omega(\epsilon_{k-1}, \epsilon_k)$. The limit behaviour of $\kappa(E, \epsilon_{k-1}, \epsilon_k)$ for $k \to \infty$ for an optimal sequence $\{\epsilon_k\}$ is a measure of the conditioning of minimizing the energy.
(b) The above analysis can be carried out for other numerical procedures as well (e.g., for higher degree discretizations). Since we want to focus on outlining the general framework we restrict ourselves to the Euler-method.

Clearly, with respect to applying a concrete numerical algorithm like the Euler method additional requirements related to the energies are necessary. One such is that an approximation of a global minimum still can be used to obtain (for example through the SLP family $\{N_n\}$ of Definition 4) a suitable approximation of a solution of the given problem. Here follow two classical problems that can be treated completely that way.

3.1 Example: Linear Systems

Let us consider a square linear system $A \cdot x = b$ with regular matrix $A \in \mathbb{R}^{n \times n}$. Define an energy

$$E(A, b, w) \;=\; \frac{1}{\|b\|^2} \cdot \|A \cdot w - b\|^2 \;,$$

where $\| \bullet \|$ denotes the Euclidean norm. We have

$$D_w E(\tilde{w}) = 2 \cdot \frac{A^T \cdot (A \cdot \tilde{w} - b)}{\|b\|^2} \;\text{ and }\; D_w^2 E(\tilde{w}) = 2 \cdot \frac{A^T \cdot A}{\|b\|^2} \;.$$

A is regular, so E is unimodal and the only critical point w^* of E is the solution. It clearly satisfies $E(w^*) = 0$. Moreover,

$$\begin{aligned} \|D_w E(A, b, \tilde{w})\| &= 2 \cdot \frac{\|(A^T)^{-1}\| \cdot \|A^T \cdot (A \cdot \tilde{w} - b)\|}{\|(A^T)^{-1}\| \cdot \|b\|^2} \\ &\geq \frac{2 \cdot \|A \cdot \tilde{w} - b\|}{\|(A^T)^{-1}\| \cdot \|b\|^2} \\ &= \frac{2 \cdot \sqrt{E(A, b, \tilde{w})}}{\|A^{-1}\| \cdot \|b\|} \end{aligned}$$

Thus, in the terminology of Theorem 3 we get for $\epsilon_1 > \epsilon_2 > 0$ the bounds

$$H(\epsilon_1, \epsilon_2) \leq \frac{2 \cdot \|A^T\| \cdot \|A\|}{\|b\|^2} \text{ and } L(\epsilon_1, \epsilon_2) \geq \frac{2 \cdot \sqrt{\epsilon_2}}{\|A^{-1}\| \cdot \|b\|}.$$

For an application of Theorem 3 we choose the sequence $\epsilon_k := \frac{1}{2^k}, \epsilon_0 := 1 = E(0) =: E_0$, a step size $\tau(\epsilon_{k-1}, \epsilon_k) := \frac{4\sigma \cdot \epsilon_k^2}{\epsilon_{k-1}^2 \cdot \|A\|^2 \cdot \|A^{-1}\|^2}$ and get as a bound on the number of steps

$$\begin{aligned} &\frac{1}{2} \cdot \sum_{k=1}^{K^*} \frac{2 \cdot \|A^T\| \cdot \|A\|}{\|b\|^2} \cdot \left(\frac{1}{2^{k-1}}\right)^2 \cdot \frac{(2^k)^2}{4} \cdot \|A^{-1}\|^2 \cdot \|b\|^2 \\ &= \sum_{k=1}^{K^*} \|A\|^2 \cdot \|A^{-1}\|^2 \\ &= \|A\|^2 \cdot \|A^{-1}\|^2 \cdot \lceil \log(\tfrac{1}{\epsilon}) \rceil \end{aligned}$$

since $K^* := \lceil \log(\frac{1}{\epsilon}) \rceil$ iterations are sufficient to reduce the energy to a value $\leq \epsilon$.

Several remarks are in charge. The quantity $\|A\| \cdot \|A^{-1}\|$ of course is well known as the condition number of a square matrix. So it is no surprise that in comes into our analysis. The complexity of the algorithm to minimize the energy also depends (besides on the number of iterations) on the complexity of evaluating $D_w E$. Thus, the approach taken in the previous section is important as well here. The latter evaluation complexity is bounded by the complexity of performing two matrix-vector multiplications, which is $O(n^2)$. Thus, for well-conditioned families of matrices, i.e, if the condition number can be bounded by a (known) constant, we get a number $O(n^2 \cdot \log \frac{1}{\epsilon})$ of arithmetic operations. Note that in this case the step sizes can be easily computed as well.

A related example can be found in [1], where the ranking problem for webpages is considered.

3.2 Example: Separating Hyperplane

In this subsection we want to show how our general framework gives back qualitatively the results obtained by the well known Perceptron learning algorithm, see [5]. Relations between the perceptron algorithm and steepest descent methods have been studied previously, see, for example, [7].

Given two finite sets X^+, X^- of points in $\mathbb{R}^n$ and a $\delta > 0$, the task is to find a $w \in \mathbb{R}^n$ such that

$$w^T \cdot x \geq \delta \ \ \forall \, x \in X^+ \text{ and } \ w^T \cdot x \leq -\delta \ \ \forall \, x \in X^- \ . \tag{4}$$

An energy for this problem can be defined as:

$$E(X^+, X^-, w) := \sum_{x \in X^+} \beta(w^T \cdot x - 2 \cdot \delta) + \sum_{x \in X^-} \beta(-w^T \cdot x - 2 \cdot \delta) \ ,$$

where

$$\beta(t) := \begin{cases} t^4 & t \leq 0 \\ 0 & t > 0 \end{cases}$$

The idea behind using this energy is as follows: First, it is not hard to see that E is twice differentiable and unimodal. If w is a hyperplane doing a correct separation, then for $x \in X^+$ we obtain $w^T \cdot x \geq \delta > 0$; similarly for $x \in X^-$. The energy is not necessarily vanishing in such a separating hyperplane w; however, E is vanishing in $2 \cdot w$. Note that β is penalizing those hyperplanes that do not separate the test sets sufficiently good, even though such a hyperplane might solve the initial problem. Note as well that any w satisfying $E(w) < \delta^4$ *is* a separating hyperplane, even though it might not be a global minimizer of E.

We compute upper and lower bounds according to Theorem 3.

i) Let $\epsilon_1 > \epsilon_2 > 0$. We compute an upper bound for $\|D^2_w E(X, w)\|$ on the set $\Omega(\epsilon_1, \epsilon_2)$. Instead of the operator norm $\|D^2_w E\|_2$ induced by the Euclidean vector norm we use the well-known estimation

$$\|D^2_w E\|_2 \leq \sqrt{\|D^2_w E\|_1 \cdot \|D^2_w E\|_\infty} \ ,$$

where $\| \bullet \|_\infty$ denotes the maximal sum of absolute values of row entries and $\| \bullet \|_1$ does the same for the column sums.

Suppose that for $X := X^+ \cup X^-$ it is $|X| =: m$, i.e. there are m many test points; let $B_X > 0$ denote a bound such that $\|x\|_\infty \leq B_X$ for all $x \in X$. For $i, j \in \{1, \ldots, n\}$ we get

$$\frac{\partial^2 E}{\partial w_i \partial w_j} = \sum_{\substack{x \in X^+ \\ w^T \cdot x < 2 \cdot \delta}} 12 \cdot (w^T \cdot x - 2 \cdot \delta)^2 \cdot x_i \cdot x_j + \\ + \sum_{\substack{x \in X^- \\ w^T \cdot x > -2 \cdot \delta}} 12 \cdot (w^T \cdot x + 2 \cdot \delta)^2 \cdot x_i \cdot x_j .$$

Using norm equivalence in $\mathbb{R}^n : \ \|z\|_2 \leq \sqrt{n} \cdot \|z\|_4 \ \forall \ z \in \mathbb{R}^n$ together with the assumption that $w \in \Omega(\epsilon_1, \epsilon_2)$ easy calculations result in

$$\|D_w^2 E(X,w)\|_\infty \ \leq \ 12 \cdot n^2 \cdot \sqrt{\epsilon_1} \cdot B_X^2$$

as well as

$$\|D_w^2 E(X,w)\|_1 \ \leq \ 12 \cdot n^2 \cdot \sqrt{\epsilon_1} \cdot B_X^2$$

and thus the same bound holds for $\|D_w^2 E(X,w)\|_2$.

ii) For obtaining a lower bound for $\|D_w E(X,w)\|$ on $\Omega(\epsilon_1, \epsilon_2)$ consider once more a separating hyperplane $\tilde{w}$ and apply the Cauchy-Schwartz inequality

$$\|D_w E(X,w)\|_2 \ \geq \|\tilde{w}\|_2^{-1} \cdot |\tilde{w}^T \cdot D_w E(X,w)| \ .$$

Now

$$|\tilde{w}^T \cdot D_w E(X,w)| \geq 4\delta \cdot \sum_{\substack{x \in X^+ \\ w^T \cdot x < 2\delta}} |w^T \cdot x - 2\delta|^3 + \sum_{\substack{x \in X^- \\ w^T \cdot x > -2\delta}} |w^T \cdot x + 2\delta|^3 \geq 4 \cdot \delta \cdot E(w)^{\frac{3}{4}} .$$

using the norm inequality $\|z\|_3 \geq \|z\|_4$ for any $z \in \mathbb{R}^n$, where $\|z\|_p := \left(\sum\limits_{i=1}^{n} z_i^p \right)^{\frac{1}{p}}$ for $p \in \mathbb{N}$.

Altogether, we obtain as lower bound on $\Omega(\epsilon_1, \epsilon_2)$:

$$\|D_w E(w)\|_2 \ \geq \ 4 \cdot \|\tilde{w}\|_2^{-1} \cdot \delta \cdot \epsilon_2^{\frac{3}{4}} .$$

iii) With these bounds we can compute the number of steps necessary to get an ϵ-approximate solution. We apply Theorem 3 with the following quantities: $\epsilon_0 := E(X,0) = 16 \cdot m \cdot \delta^4; \epsilon_k := \frac{E(\epsilon_0)}{2^k}, \epsilon > 0$ fixed and

$$H(\epsilon_{k-1}, \epsilon_k) \leq 12 \cdot n \cdot B_X^2 \cdot \sqrt{\epsilon_{k-1}} \ , \ L(\epsilon_{k-1}, \epsilon_k) \geq 4 \cdot \delta \cdot \|\tilde{w}\|_2^{-1} \cdot \epsilon_k^{\frac{3}{4}}$$

for any separating hyperplane $\tilde{w}$; the latter should be taken in the analysis so to minimize the norm (note that $\tilde{w}$ is not used in the algorithm).

Thus, the number of steps to reduce the energy from ϵ_{k-1} to ϵ_k is bounded by

$$\frac{1}{2} \cdot \frac{H(\epsilon_{k-1}, \epsilon_k) \cdot \epsilon_{k-1}^2}{L(\epsilon_{k-1}, \epsilon_k)^2 \cdot \epsilon_k} \leq \frac{3}{\sqrt{2}} \cdot \frac{n^2 \cdot B_X^2 \cdot \|\tilde{w}\|_2^2}{\delta^2}.$$

With $K^* := O(\log \frac{E_0}{\epsilon})$ many iterations we thus need

$$O\left(\frac{n^2 \cdot B_X^2 \cdot \|\tilde{w}\|_2^2}{\delta^2} \cdot \log \frac{E_0}{\epsilon}\right)$$

many steps. Finally, recalling that $\epsilon < \delta^4$ is a sufficient choice for obtaining a separating hyperplane we end up with

$$O\left(\frac{n^2 \cdot B_X^2 \cdot \|\tilde{w}\|_2^2}{\delta^2} \cdot \log m\right)$$

many steps for the Euler-discretization. For the computation of the gradient in each step an upper bound of order $O(n \cdot m)$ is obvious.

These bounds pretty well correspond to the bounds known from the perceptron convergence theorem, see [5]. The important difference is that our algorithm results as just one specific example from a much more general framework, that hopefully can be applied to a larger class of problems as well.

4 Conclusion and Acknowledgement

We have studied a framework for measuring the complexity of analog systems. The latter is based on the notion of a problem. In its more abstract part the approach allows to define complexity classes independently of particular trajectory following algorithms. The existence of complete problems for such classes was established. In a second part we considered steepest descent algorithms for discretized versions of our problems resulting in a concrete running time analysis. Two such examples were discussed.

We still believe that a lot of questions have to be investigated. To get a better overview what has been done so far and which problems are waiting to be solved let us finally refer once again to the survey by Bournez and Campognolo [3].

We would like to thank the anonymous referees for some helpful remarks.

References

1. Bianchini, M., Gori, M., Scarselli, F.: Inside PageRank. ACM Transactions on Internet Technology 5(1), 92–128 (2005)
2. Blum, L., Cucker, F., Shub, M., Smale, S.: Complexity and Real Computation. Springer, Heidelberg (1998)
3. Bournez, O., Campagnolo, M.L.: A Survey on Continuous Time Computations. Preprint (2006)
4. Gori, M., Meer, K.: A step towards a complexity theory for analog systems. Mathematical Logic Quarterly 48(1), 45–58 (2002)

5. Minsky, M., Papert, S.: Perceptrons. The MIT Press, Cambridge (1969)
6. Orponen, P.: A survey of continuous-time computation theory. In: Du, D.-Z., Ko, K.-I. (eds.) Advances in Algorithms, Languages, and Complexity, pp. 209–224. Kluwer Academic Publishers, Dordrecht (1997)
7. Shynk, J.J.: Performance Surfaces of a Single-Layer Perceptron. IEEE Transactions on Neural Networks 1(3), 268–274 (1990)
8. Zak, M.: Introduction to terminal dynamics. Complex Systems 7, 59–87 (1993)