DIMACS Technical Report 95-17
June 1995

Sample Complexity for Learning Recurrent Perceptron Mappings

by

Bhaskar Dasgupta [*],[†]   Eduardo D. Sontag [‡],[§]

[*]DIMACS
Rutgers University
New Brunswick, NJ 08903
bhaskar@dimacs.rutgers.edu
[†]This research was supported in part by US Air Force Grant AFOSR-94-0293
[‡]Department of Mathematics
Rutgers University
New Brunswick, NJ 08903
sontag@hilbert.rutgers.edu
[§]This research was supported in part by US Air Force Grant AFOSR-94-0293

DIMACS is a cooperative project of Rutgers University, Princeton University, AT&T Bell Laboratories and Bellcore.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

Recurrent perceptron classifiers generalize the classical perceptron model. They take into account those correlations and dependences among input coordinates which arise from linear digital filtering. This paper provides tight bounds on sample complexity associated to the fitting of such models to experimental data.

1 Introduction

One of the most popular approaches to binary pattern classification, underlying many statistical techniques, is based on *perceptrons* or *linear discriminants*; see for instance the classical reference [9]. In this context, one is interested in classifying k-dimensional input patterns $v = (v_1, \ldots, v_k)$ into two disjoint classes A^+ and A^-. A perceptron P which classifies vectors into A^+ and A^- is characterized by a vector (of "weights") $\vec{c} \in \mathbb{R}^k$, and operates as follows. One forms the inner product $\vec{c}.v = c_1 v_1 + \ldots c_k v_k$. If this inner product is positive, v is classified into A^+, otherwise into A^-. (A variation allows for an additional constant term c_0, corresponding geometrically to a partition of $\mathbb{R}^k$ by a hyperplane not passing through the origin.)

In practice, given a large number of labeled ("training") samples $(v^{(i)}, \varepsilon_i)$, where $\varepsilon_i \in \{+, -\}$, one attempts to find a vector $\vec{c}$ so that $\vec{c}.v^{(i)}$ is positive when ε_i = "+" and negative (or zero) otherwise. Finding such a vector amounts to solving a linear programming problem, and recursive algorithms ("perceptron learning method") are popular for its solution. The resulting perceptron corresponding to one such vector $\vec{c}$ is then used to classify new, previously unseen, examples. The philosophy underlying this procedure is this: One starts with the hypothesis that the sets A^+ and A^- are indeed linearly separable, that is, there is hyperplane having them in opposite sides. In addition, it is assumed that the training samples are in either A^+ or A^-, and are labeled accordingly. Provided that the training set is large enough, a hyperplane separating the samples is a good approximation of a true separating hyperplane for A^+ and A^-. This philosophy can be made precise on the basis of sample complexity bounds ("VC dimension" as discussed below), and can be found in classical references (see e.g. [20]). These bounds give estimates of the number of random training samples needed so that a perceptron consistent with the seen samples will also, with high probability, perform well on unseen data; see in particular the exposition in [14].

Recurrent Perceptrons

In signal processing and control applications, the size k of the input vectors v is typically very large, so the number of samples needed in order to accurately "learn" an appropriate classifying perceptron is in principle very large. On the other hand, in such applications the classes A^+ and A^- often can be separated by means of a dynamical system of fairly small dimensionality. The existence of such a dynamical system reflects the fact that the signals of interest exhibit context dependence and correlations, and this prior information can help in narrowing down the search for a classifier. Various dynamical system models for classification appear from instance when learning finite automata and languages —see e.g. [10]— and in signal processing as a channel equalization problem (at least in the simplest 2-level case) when modeling linear channels transmitting digital data from a quantized source —see [3] and also the related paper [15].

When dealing with linear dynamical classifiers, the inner product $\vec{c}.v$ represents a convolution by a separating vector $\vec{c}$ that is the impulse-response of a recursive digital filter of some order $n \ll k$. Equivalently, one assumes that the data can be classified using a $\vec{c}$ that is *n-recursive*, meaning that there exist real numbers $r_1, \ldots, r_n$ so that

$$c_j = \sum_{i=1}^{n} c_{j-i} r_i\,, \quad j = n+1, \ldots, k\,.$$

Seen in this context, the usual perceptrons are nothing more than the very special subclass of "finite impulse response" systems (all poles at zero); thus it is appropriate to call the more general class "recurrent" or "IIR (infinite impulse response)" perceptrons. Some authors, particularly Back and Tsoi —see e.g. [1, 2]— have introduced these ideas in the neural network literature. There is also related work in control theory dealing with such classifying, or more generally quantized-output, linear systems; see [8, 13].

The problem that we consider in this paper is: if one assumes that there is an n-recursive vector $\vec{c}$ that serves to classify the data, and one knows n but not the particular vector, how many labeled samples $v^{(i)}$ are needed so as to be able to reliably estimate $\vec{c}$? More specifically, we want to be able to guarantee that any classifying vector consistent with the seen data will classify "correctly with high probability" the unseen data as well. The precise formulation is the terms of empirical means and computational leaning theory, and is reviewed below. Very roughly speaking, the main result is that the number of samples needed is proportional to the logarithm of the length k (as opposed to k itself, as would be the case if one did not take advantage of the recurrent structure). We also make some remarks on the actual computational complexity of finding a vector $\vec{c}$ consistent with the training data.

Sample Complexity and VC Dimension

We next very briefly review some (by now standard) notions regarding sample complexity, with the purpose of motivating the main results, which deal with the calculation of VC dimensions. For more details see the book [20], the paper [6], or the survey [14] (the particular terminology used here is as in the exposition [17]).

In the general classification problem, an input space $\mathbb{X}$ as well as a collection $\mathcal{F}$ of maps $\mathbb{X} \to \{-1, 1\}$ are assumed to have been given. (The set $\mathbb{X}$ is assumed to be either countable or an Euclidean space, and the maps in $\mathcal{F}$ are assumed to be measurable. In addition, mild regularity assumptions are made which insure that all sets appearing below are measurable, but details are omitted since in our context these assumptions are always satisfied.) Let W be the set of all sequences

$$w = (u_1, \phi(u_1)), \ldots, (u_s, \phi(u_s))$$

over all $s \geq 1$, $(u_1, \ldots, u_s) \in \mathbb{X}^s$, and $\phi \in \mathcal{F}$. An *identifier* is a map $\varphi : W \to \mathcal{F}$. The value of φ on a sequence w as above will be denoted as φ_w. The *error* of φ with respect to a probability measure P on $\mathbb{X}$, a $\phi \in \mathcal{F}$, and a sequence $(u_1, \ldots, u_s) \in \mathbb{X}^s$, is

$$\mathrm{Err}(P, \phi, u_1, \ldots, u_s) := \mathrm{Prob}\,[\varphi_w(u) \neq \phi(u)]$$

(where the probability is being understood with respect to P).

The class $\mathcal{F}$ is said to be (uniformly) *learnable* if there is some identifier φ with the following property: For each $\varepsilon, \delta > 0$ there is some s so that, for every probability P and every $\phi \in \mathcal{F}$,

$$\mathrm{Prob}\left[\mathrm{Err}(P, \phi, u_1, \ldots, u_s) > \varepsilon\right] < \delta$$

(where the probability is being understood with respect to P^s on $\mathbb{X}^s$).

In the learnable case, the function $s(\varepsilon, \delta)$ which provides, for any given ε and δ, the smallest possible s as above, is called the *sample complexity* of the class $\mathcal{F}$. It can be proved that learnability is equivalent to finiteness of the *Vapnik-Chervonenkis dimension* ν of the class $\mathcal{F}$, a combinatorial concept whose definition we recall later. In fact, $s(\varepsilon, \delta)$ is bounded by a polynomial in $1/\varepsilon$ and $1/\delta$ and is proportional to ν in the following precise sense (cf. [6]):

$$s(\varepsilon, \delta) \;\leq\; \max\left\{\frac{8\nu}{\varepsilon}\log\left(\frac{13}{\varepsilon}\right), \frac{4}{\varepsilon}\log\left(\frac{2}{\delta}\right)\right\}$$

(there is a similar lower bound), and this motivates the studies dealing with estimating VC dimension, as we pursue here.

It can also be proved that, if there is any identifier at all in the above sense, then one can always use the following naive identification procedure: pick any element ϕ which is consistent with the observed data. In the statistics literature —see [20]— this "naive technique" is a particular case of what is called *empirical risk minimization*. When there is an algorithm that allows computing such a function ϕ in time polynomial on the sample size, the class is said to be learnable in the PAC ("probably approximately correct") sense of Valiant (cf. [19]). In this paper we concentrate on the question of uniform learnability in the sample complexity sense, for recurrent perceptron concept classes, but we will also prove a result, in Section 5 regarding PAC learnability for such classes. Generalizations to the learning of real-valued (as opposed to Boolean) functions, by evaluation of the "pseudo-dimension" of recurrent maps, are discussed as well, in Section 6.

2 Definitions and Statements of Main Results

The concept of VC dimension is classically defined in terms of abstract concept classes. Assume that we are given a set $\mathbb{X}$, called the *set of inputs*, and a family of subsets $\mathcal{C}$ of $\mathbb{X}$, called the set of "concepts." A subset $X \subseteq \mathbb{X}$ is said to be shattered (by the class $\mathcal{C}$) if for each subset $B \subseteq X$ there is some $C \in \mathcal{C}$ such that $B = C \bigcap X$. The VC dimension is then the largest possible positive integer n (possibly $+\infty$) so that there is some $X \subseteq \mathbb{X}$ of cardinality n which can be shattered. An equivalent manner of stating these notions, somewhat more suitable for our purposes, proceeds by identifying the subsets of X with Boolean functions from X to $\{-1, 1\}$ (we pick $\{-1, 1\}$ instead of $\{0, 1\}$ for notational convenience): to each such Boolean function ϕ there is an associated subset, namely $\{x \in X \mid \phi(x) = 1\}$, and conversely, to each set $B \subseteq X$ one can associate its characteristic function ϕ_B defined on the set X. Similarly, we can think of the sets $C \in \mathcal{C}$ as Boolean functions on $\mathbb{X}$ and the intersections $C \bigcap X$ as the restrictions of such functions to X. Thus we restate the definitions now in terms of functions.

Given the set $\mathbb{X}$, and a subset X of $\mathbb{X}$, a *dichotomy* on X is a function

$$\delta : X \to \{-1, 1\}.$$

Assume given a class $\mathcal{F}$ of functions $\mathbb{X} \to \{-1, 1\}$, to be called the class of *classifier* functions. The subset $X \subseteq \mathbb{X}$ is *shattered* by $\mathcal{F}$ if each dichotomy on X is the restriction to X of some $\phi \in \mathcal{F}$. The *Vapnik-Chervonenkis dimension* $\mathrm{VC}(\mathcal{F})$ is the supremum (possibly infinite) of the set of integers κ for which there is some subset $X \subseteq \mathbb{X}$ of cardinality κ which can be shattered by $\mathcal{F}$.

Pick any two integers $n>0$ and $q \geq 0$. A sequence

$$\vec{c} = (c_1, \ldots, c_{n+q}) \in \mathbb{R}^{n+q}$$

is said to be *n-recursive* if there exist real numbers $r_1, \ldots, r_n$ so that

$$c_{n+j} = \sum_{i=1}^{n} c_{n+j-i} r_i \,, \quad j = 1, \ldots, q \,.$$

(In particular, every sequence of length n is n-recursive, but the interesting cases are those in which $q \neq 0$, and in fact $q \gg n$.) Given such an n-recursive sequence $\vec{c}$, we may consider its associated *perceptron* classifier. This is the map

$$\phi_{\vec{c}} : \mathbb{R}^{n+q} \to \{-1, 1\} : \quad (x_1, \ldots, x_{n+q}) \mapsto \operatorname{sign}\left(\sum_{i=1}^{n+q} c_i x_i\right)$$

where the sign function is understood to be defined by $\operatorname{sign}(z) = -1$ if $z \leq 0$ and $\operatorname{sign}(z) = 1$ otherwise. (Changing the definition at zero to be $+1$ would not change the results to be presented in any way.) We now introduce, for each two fixed n, q as above, a class of functions:

$$\mathcal{F}_{n,q} := \left\{ \phi_{\vec{c}} \mid \vec{c} \in \mathbb{R}^{n+q} \text{ is } n\text{-recursive} \right\}.$$

This is understood as a function class with respect to the input space $\mathbb{X} = \mathbb{R}^{n+q}$, and we are interested in estimating $\mathrm{VC}(\mathcal{F}_{n,q})$.

Our main result will be as follows (in this paper, all logarithms are understood to be in base 2):

Theorem 1 $\boxed{\max\left\{n, n\lfloor \log(\lfloor 1 + \frac{q-1}{n} \rfloor)\rfloor\right\} \leq \mathrm{VC}(\mathcal{F}_{n,q}) \leq \min\{n+q\,,\, 5.35n + 4n\log(q+1)\}}$.

The upper bound is a simple consequence of an argument based on parameter counts, and is given in Section 4. Much more interesting is the almost matching lower bound, which will involve a result on dual VC dimensions which we prove in Section 3.

Some particular cases are worth discussing. When $q = O(n)$ then both the upper and the lower bounds are of the type cn for some (different) constants c. If $q = \Omega(n^{1+\epsilon})$ (for any constant $\epsilon > 0$), then both the upper and the lower bounds are of the form $cn \log(\frac{q}{n})$ for some constants c. In this latter case, assume that one is interested in the behavior of $\mathrm{VC}(\mathcal{F}_{n,q})$ as $n \to +\infty$ while q grows polynomially in n; then the upper and lower bounds are both of the type $cn \log n$, for some constants c. If instead q grows exponentially on n, both the upper and lower bounds are polynomial in n.

The organization of the rest of the paper is as follows. In Section 3 we prove an abstract result on VC-dimension, which is then used in Section 4 to prove Theorem 1. In Section 5, we show that the consistency problem for recurrent perceptrons can be solved in polynomial time, for any fixed n; some recent facts regarding representations of real numbers and decision problems for real-closed fields, needed in this Section, are reviewed in an Appendix. Finally, Section 6 deals with bounds on the sample complexity needed for identification of linear dynamical systems, that is to say, the real-valued functions obtained when not taking "signs" when defining the maps $\phi_{\vec{c}}$. This requires a quick review of recent results dealing with the learning of real-valued functions; the basic result there is that upper bounds similar to the binary case hold.

3 An Abstract Result on VC Dimension

Assume that we are given two sets $\mathbb{X}$ and Λ, to be called in this context the set of *inputs* and the set of *parameter values* respectively. Suppose that we are also given a function

$$F \,:\, \Lambda \times \mathbb{X} \to \{-1, 1\}.$$

Associated to this data is the class of functions

$$\mathcal{F} \,:=\, \{F(\lambda, \cdot) : \mathbb{X} \to \{-1, 1\} \,|\, \lambda \in \Lambda\}$$

obtained by considering F as a function of the inputs alone, one such function for each possible parameter value λ. We will prove lower bounds in Theorem 1 by studying the VC dimension of classes obtained in this parametric fashion.

Note that, given the same data one could, dually, study the class

$$\mathcal{F}^* \,:\, \{F(\cdot, \xi) : \Lambda \to \{-1, 1\} \,|\, \xi \in \mathbb{X}\}$$

which obtains by fixing the elements of $\mathbb{X}$ and thinking of the parameters as inputs. It is well-known (and in any case, a consequence of the more general result to be presented below) that

$$\mathrm{VC}(\mathcal{F}) \geq \lfloor \log(\mathrm{VC}(\mathcal{F}^*)) \rfloor,$$

which provides a lower bound on $\mathrm{VC}(\mathcal{F})$ in terms of the "dual VC dimension." A sharper estimate is possible when Λ can be written as a product of n sets

$$\Lambda = \Lambda_1 \times \Lambda_2 \times \ldots \times \Lambda_n \tag{1}$$

and that is the topic which we develop next.

We assume from now on that a decomposition of the form in Equation (1) is given, and will define a variation of the dual VC dimension by asking that only certain dichotomies on Λ be obtained from $\mathcal{F}^*$. We define these dichotomies only on "rectangular" subsets of Λ, that is, sets of the form

$$L = L_1 \times \ldots \times L_n \subseteq \Lambda$$

with each $L_i \subseteq \Lambda_i$ a nonempty subset. Given any index $1 \leq \kappa \leq n$, by a *κ-axis dichotomy* on such a subset L we mean any function $\delta : L \to \{-1, 1\}$ which depends only on the κth coordinate, that is, there is some function $\phi : L_\kappa \to \{-1, 1\}$ so that $\delta(\lambda_1, \ldots, \lambda_n) = \phi(\lambda_\kappa)$ for all $(\lambda_1, \ldots, \lambda_n) \in L$; an axis dichotomy is a map that is a κ-axis dichotomy for some κ. A rectangular set L will be said to be *axis-shattered* if every axis dichotomy is the restriction to L of some function of the form $F(\cdot, \xi) : \Lambda \to \{-1, 1\}$, for some $\xi \in \mathbb{X}$.

Theorem 2 *If $L = L_1 \times \ldots \times L_n \subseteq \Lambda$ can be axis-shattered and each set L_i has cardinality r_i, then* $\mathrm{VC}(\mathcal{F}) \geq \lfloor \log(r_1) \rfloor + \ldots + \lfloor \log(r_n) \rfloor$.

Note that in the special case n=1 one recovers the result $\mathrm{VC}(\mathcal{F}) \geq \lfloor \log(\mathrm{VC}(\mathcal{F}^*)) \rfloor$. We will prove this theorem below, after a couple of small observations.

Remark 3.1 Assume that $L = L_1 \times \ldots \times L_n \subseteq \Lambda$ can be axis-shattered. Pick any indices (possibly equal) $\kappa_1, \kappa_2 \in \{1, \ldots, n\}$ and any functions $\phi_i : L_{\kappa_i} \to \{-1, 1\}$, $i = 1, 2$. By definition of axis-shattering, there exist elements $\xi_1, \xi_2 \in \mathbb{X}$, such that

$$F(\lambda_1, \ldots, \lambda_n, \xi_i) = \phi_i(\lambda_{\kappa_i}) \quad \forall (\lambda_1, \ldots, \lambda_n) \in L_1 \times \ldots \times L_n \,. \tag{2}$$

We then have:

(a) If $\kappa_1 = \kappa_2$ and $\xi_1 = \xi_2$ then $\phi_1 = \phi_2$.

(b) If $\kappa_1 \neq \kappa_2$ and $\xi_1 = \xi_2$ then both ϕ_1 and ϕ_2 are constant functions.

Property (a) is obvious. Property (b) is proved as follows. Without loss of generality, we may take $\kappa_1 = 1$ and $\kappa_2 = 2$. Now pick $\widehat{\lambda}_2, \ldots, \widehat{\lambda}_n$ arbitrarily. Then

$$\phi_1(\lambda) = F(\lambda, \widehat{\lambda}_2, \ldots, \widehat{\lambda}_n, \xi) = \phi_2(\widehat{\lambda}_2)$$

for all $\lambda \in L_1$, and a similar argument shows that ϕ_2 is constant as well. □

Lemma 3.2 Let $\mathcal{S} = \{s_1, s_2, \ldots, s_r\}$ be a set of cardinality $r = 2^m$, where m is a positive integer. Then there exists a set of m (distinct) dichotomies $\phi_1, \phi_2, \ldots, \phi_m$ on $\mathcal{S}$ with the following property: For each vector $(a_1, a_2, \ldots, a_m) \in \{-1, 1\}^m$, there exists a unique index $j \in \{1, \ldots r\}$ such that

$$\phi_i(s_j) = a_i \,, \quad i = 1, \ldots, m \,. \tag{3}$$

Moreover, none of the functions ϕ_i is a constant function.

Proof. Take as M the $m \times r$ matrix whose columns are the 2^m possible vectors in $\{-1,1\}^m$ and now let the functions ϕ_i be defined by the formula $\phi_i(s_j) = M_{ij}$ for all $1 \leq i \leq m$ and $1 \leq j \leq r$. Clearly, given any vector $(a_1, a_2, \ldots, a_m) \in \{-1,1\}^m$, there exists a unique index j such that $M_{ij} = a_i$ for $1 \leq i \leq m$, or equivalently, so that Equation (3) holds. By construction, $\phi_i \neq \phi_{i'}$ whenever $i \neq i'$, and since each row of M contains at least one 1 and one -1, none of the functions ϕ_i is a constant function. ∎

Proof of Theorem 2. We may assume without loss of generality that each $r_\kappa = 2^{m_\kappa}$ for some positive integers $m_1, \ldots, m_n$. This is because any possible indices so that $r_\kappa = 1$ can be dropped (and the result proved with smaller n), and, for each $r_\kappa > 1$, a subset L'_κ of L_κ, of cardinality $2^{\lfloor \log r_\kappa \rfloor}$, could be used instead of the original L_κ if r_κ is not a power of two.

To prove the Theorem, it will be enough to find n disjoint subsets $X_1, X_2, \ldots, X_n$ of $\mathbb{X}$, of cardinalities $m_1, \ldots, m_n$ respectively, so that the set $X = X_1 \bigcup X_2 \bigcup \ldots \bigcup X_n$ is shattered. Pick any $\kappa \in \{1, \ldots, n\}$. Consider the set $L_\kappa = \{l_{\kappa,1}, l_{\kappa,2}, \ldots, l_{\kappa,r_\kappa}\}$. By Lemma 3.2 applied to this set, there exists a set of m_κ distinct and nonconstant dichotomies $\phi_{\kappa,1}, \phi_{\kappa,2}, \ldots, \phi_{\kappa,m_\kappa}$ on L_κ so that, for any vector $(a_1, a_2, \ldots, a_{m_\kappa}) \in \{-1,1\}^{m_\kappa}$, there exists a unique index $1 \leq j_\kappa \leq r_\kappa$ so that

$$\phi_{\kappa,i}(l_{\kappa,j_\kappa}) = a_i\ , \quad i = 1, \ldots, m_\kappa\,. \tag{4}$$

Since L can be axis-shattered, each of the axis dichotomies $\phi_{\kappa,i}$ can be realized as a function $F(\cdot, \xi)$. That is, there exists a set inputs

$$X_\kappa \;=\; \{\xi_{\kappa,1}, \xi_{\kappa,2}, \ldots, \xi_{\kappa,m_\kappa}\}$$

so that, for each $i = 1, \ldots, m_\kappa$,

$$F(\lambda_1, \ldots, \lambda_n, \xi_{\kappa,i}) = \phi_{\kappa,i}(\lambda_\kappa)\ , \quad \forall (\lambda_1, \ldots, \lambda_n) \in L_1 \times \ldots \times L_n\,. \tag{5}$$

Note also that, by construction, $\xi_{\kappa,i} \neq \xi_{\kappa,i'}$ for $i \neq i'$, since the corresponding functions $\phi_{\kappa,i}$ are distinct (recall Remark 3.1, part (a)).

Summarizing, for each vector $(a_1, a_2, \ldots, a_{m_\kappa}) \in \{-1,1\}^{m_\kappa}$ and for each $\kappa \in \{1, \ldots, n\}$ there is some $1 \leq j_\kappa \leq r_\kappa$ so that

$$F(\lambda_1, \ldots, \lambda_{\kappa-1}, l_{\kappa,j_\kappa}, \lambda_{\kappa+1}, \ldots, \lambda_n, \xi_{\kappa,i}) \;=\; \phi_{\kappa,i}(l_{\kappa,j_\kappa}) \;=\; a_i\ , \quad i = 1, \ldots, m_\kappa \tag{6}$$

for all $\lambda_q \in L_q$ $(q \neq \kappa)$. We do this construction for each κ and define $X := X_1 \bigcup X_2 \bigcup \ldots \bigcup X_n$. Note that the sets X_κ are disjoint, since $\xi_{\kappa,i} \neq \xi_{\kappa',i'}$ whenever $\kappa \neq \kappa'$ (by part (b) of Remark 3.1 and the fact that the functions $\phi_{\kappa,i}$ are all nonconstant). The set X can be shattered. Indeed, assume given any dichotomy $\delta : X \to \{-1,1\}$. Using Equation (6), with the vector $a = (\delta(\xi_{\kappa,1}), \ldots, \delta(\xi_{\kappa,m_\kappa}))$ for each κ, it follows that for each $\kappa \in \{1, \ldots, n\}$ there is some $1 \leq j_\kappa \leq r_\kappa$ so that

$$F(l_{1,j_1}, \ldots, l_{n,j_n}, \xi_{\kappa,i}) \;=\; \delta(\xi_{\kappa,i})\ , \quad i = 1, \ldots, m_\kappa.$$

That is, the function $F(\lambda, \cdot)$ coincides with δ on X, when one picks $\lambda = (l_{1,j_1}, \ldots, l_{n,j_n})$. ∎

Note that the lower bound in the above result is almost tight, because by Lemma 4.2 there is a set of the form $L = L_1 \times \ldots \times L_n \subseteq \Lambda$ which can be axis-shattered and for which $\mathrm{VC}(\mathcal{F}) = O(n \log(rn))$, with cardinality of each L_i greater or equal to r for each i.

4 Proof of Main Result

We recall the following result; it was proved, using Milnor-Warren bounds on the number of connected components of semialgebraic sets, by Goldberg and Jerrum:

Fact 4.1 ([11]) Assume given a function $F : \Lambda \times \mathbb{X} \to \{-1, 1\}$ and the associated class of functions $\mathcal{F} := \{F(\lambda, \cdot) : \mathbb{X} \to \{-1, 1\} \mid \lambda \in \Lambda\}$. Suppose that $\Lambda = \mathbb{R}^k$ and $\mathbb{X} = \mathbb{R}^n$, and that the function F can be defined in terms of a Boolean formula involving at most s polynomial inequalities in $k + n$ variables, each polynomial being of degree at most d. Then, $\mathrm{VC}(\mathcal{F}) \leq 2k \log(8eds)$. □

Lemma 4.2 $\mathrm{VC}(\mathcal{F}_{n,q}) \leq \min\{n + q\,,\, 5.35n + 4n \log(q + 1)\}$

Proof. Since $\mathcal{F}_{n,q} \subseteq \mathcal{F}_{n+q,0}$,

$$\mathrm{VC}(\mathcal{F}_{n,q}) \leq \mathrm{VC}(\mathcal{F}_{n+q,0}) = n + q$$

where the last equality follows from the fact that $\mathrm{VC}(\mathrm{sign}(\mathcal{G})) = \dim(\mathcal{G})$ when $\mathcal{G}$ is a vector space of real-valued functions (the standard "perceptron" model). On the other hand, it is easy to see (by induction on j) that, for n-recursive sequences, c_{n+j} (for $1 \leq j \leq q$) is a polynomial in $c_1, c_2, \ldots, c_n, r_1, r_2, \ldots, r_n$ of degree exactly $j + 1$. Thus one may see $\mathcal{F}_{n,q}$ as a class obtained parametrically, and applying Fact 4.1 (with k=$2n$, s=1, d=$q + 1$) gives $\mathrm{VC}(\mathcal{F}_{n,q}) < 5.35n + 4n \log(q + 1)$. ■

Lemma 4.3 $\mathrm{VC}(\mathcal{F}_{n,q}) \geq \max\{n, n\lfloor\log(\lfloor 1 + \frac{q-1}{n}\rfloor)\rfloor\}$

Proof. As $\mathcal{F}_{n,q}$ contains the class of functions $\phi_{\vec{c}}$ with $\vec{c} = (c_1, \ldots, c_n, 0, \ldots, 0)$, which in turn being the set of signs of an n-dimensional linear space of functions, has VC dimension n, we know that $\mathrm{VC}(\mathcal{F}_{n,q}) \geq n$. Thus we are left to prove that if $q > n$ then $\mathrm{VC}(\mathcal{F}_{n,q}) \geq n\lfloor\log(\lfloor 1 + \frac{q-1}{n}\rfloor)\rfloor$.

The set of n-recursive sequences of length $n + q$ includes the set of sequences of the following special form:

$$c_j = \sum_{i=1}^{n} \alpha_i l_i^{j-1}\,, \quad j = 1, \ldots, n + q \tag{7}$$

where $\alpha_i, l_i \in \mathbb{R}$ for each $i = 1, \ldots, n$. (More precisely, this is a characterization of those n-recursive sequences of length $n + q$ for which the characteristic roots, that is, the roots of the polynomial determined by the recursion coefficients, are all real and distinct; such facts are classical in the theory of recurrences.) In turn, this includes the sequences as in Equation (7) in which one uses only $\alpha_1 = \ldots = \alpha_n = 1$. Hence, to prove the lower bound, it is sufficient to study the class of functions induced by

$$F : \mathbb{R}^n \times \mathbb{R}^{n+q} \to \{-1, 1\}\,, \quad (\lambda_1, \ldots, \lambda_n, x_1, \ldots, x_{n+q}) \mapsto \mathrm{sign}\left(\sum_{i=1}^{n}\sum_{j=1}^{n+q} \lambda_i^{j-1} x_j\right) \tag{8}$$

Let $r = \lfloor \frac{q+n-1}{n} \rfloor$ and let $L_1, \ldots, L_n$ be n disjoint sets of real numbers (if desired, integers), each of cardinality r. Let $L = \bigcup_{i=1}^{n} L_i$. In addition, if $rn < q+n-1$, then select an additional set B of $(q+n-rn-1)$ real numbers disjoint from L.

We will apply Theorem 2, showing that the rectangular subset $L_1 \times \ldots \times L_n$ can be axis-shattered. Pick any $\kappa \in \{1, \ldots, n\}$ and any $\phi : L_\kappa \to \{-1, 1\}$. Consider the (unique) interpolating polynomial

$$p(\lambda) = \sum_{j=1}^{n+q} x_j \lambda^{j-1}$$

in λ of degree $q+n-1$ such that

$$p(\lambda) = \begin{cases} \phi(\lambda) & \text{if } \lambda \in L_\kappa \\ 0 & \text{if } \lambda \in (L \cup B) - L_\kappa. \end{cases}$$

One construction of such a polynomial is via the Lagrange formula

$$\sum_{l \in L_\kappa} \phi(l) \frac{\Pi_{l_j \in L \cup B \,;\, l_j \neq l}(\lambda - l_j)}{\Pi_{l_j \in L \cup B \,;\, l_j \neq l}(l - l_j)} .$$

Now pick $\xi = (x_1, \ldots, x_{n+q-1})$. Observe that

$$F(l_1, l_2, \ldots, l_n, x_1, \ldots, x_{n+q}) = \operatorname{sign}\left(\sum_{i=1}^{n} p(l_i)\right) = \phi(l_\kappa)$$

for all $(l_1, \ldots, l_n) \in L_1 \times \ldots \times L_n$, since $p(l) = 0$ for $l \notin L_\kappa$ and $p(l) = \phi(l)$ otherwise. It follows from Theorem 2 that $\mathrm{VC}\,(\mathcal{F}_{n,q}) \geq n \lfloor \log(r) \rfloor$, as desired. ∎

5 The Consistency Problem

We next briefly discuss polynomial time learnability of recurrent perceptron mappings. As discussed in e.g. [18], in order to formalize this problem we need to first choose a *data structure* to represent the hypotheses in $\mathcal{F}_{n,q}$. In addition, since we are dealing with complexity of computation involving real numbers, we must also clarify the meaning of "finding" a hypothesis, in terms of a suitable notion of polynomial-time computation. Once this is done, the problem becomes that of solving the *consistency problem*:

> Given a set of $s \geq s(\varepsilon, \delta)$ inputs $\xi_1, \xi_2, \ldots, \xi_s \in \mathbb{R}^{n+q}$, and an arbitrary dichotomy $\Delta : \{\xi_1, \xi_2, \ldots, \xi_s\} \to \{-1, 1\}$ find a representation of a hypothesis $\phi_{\vec{c}} \in \mathcal{F}_{n,q}$ such that the restriction of $\phi_{\vec{c}}$ to the set $\{\xi_1, \xi_2, \ldots, \xi_s\}$ is identical to the dichotomy Δ (or report that no such hypothesis exists).

The representation to be used should provide an *efficient encoding* of the values of the parameters $r_1, \ldots, r_n, c_1, \ldots, c_n$: given a set of inputs $(x_1, \ldots, x_{n+q}) \in \mathbb{R}^{n+q}$, one should be able to efficiently check concept membership (that is, compute sign $(\sum_{i=1}^{n+q} c_i x_i)$). Regarding the precise meaning of polynomial-time computation, there are at least two models of complexity possible. The first, the *unit cost model* of computation, is intended to capture the algebraic complexity of the problem; in that model, each arithmetic and comparison operation on two real numbers is assumed to take unit time, and finding a representation in polynomial time means doing so in time polynomial on $s + n + q$. An alternative, the *logarithmic cost model*, is closer to the notion of computation in the usual Turing machine sense; in this case one assumes that the inputs $(x_1, \ldots, x_{n+q})$ are rational numbers, with numerators and denominators of size at most L bits, and the time involved in finding a representation of $r_1, \ldots, r_n, c_1, \ldots, c_n$ is required to be polynomial on L as well.

We study the complexity of the learning problem for constant n (but varying q). The key step is treating consistency, since if the decision version of a consistency problem is NP-hard, then the corresponding class is not properly polynomially learnable under the complexity theoretic assumption RP$\neq$NP, cf. [6]. For a suitable choice of representation, we will prove the following result:

Theorem 3 *For each fixed $n > 0$, the consistency problem for $\mathcal{F}_{n,q}$ can be solved in time polynomial in q and s in the unit cost model, and time polynomial in q, s, and L in the logarithmic cost model.*

Since $\text{VC}\,(\mathcal{F}_{n,q}) = O(n+n\log(q+1))$, it follows from here that the class $\mathcal{F}_{n,q}$ is learnable in time polynomial in q (and L in the log model). Our proof will consist of a simple application of several recent results and concepts, given in [4, 5, 16], which deal with the computational complexity aspects of the first-order theory of real-closed fields. Note that we do not study scaling with respect to n: for $q = 0$, this reduces to the still-open question of polynomial time solution of linear programming problems, in the unit cost model.

Proof of Theorem 3. For asymptotic results we may assume, without loss of generality, that $s > 2n$ from the bound of Theorem 1. We will use the representation discussed in the Appendix for the coefficients $c_1, \ldots, c_n, r_1, \ldots, r_n$, seen as vectors in $\mathbb{R}^k$, $k = 2n$. We first write the consistency problem as a problem of the following type:

($\star$) find some $c_1, \ldots, c_n, r_1, \ldots, r_n \in \mathbb{R}$ such that $\wedge_{i=1}^{s}\ (Q_i\ \Delta_i\ 0)$ (or report that no such parameter values exist)

where each Q_i is a certain real polynomial in the variables $r_1, \ldots, r_n, c_1, \ldots, c_n$ of degree at most $q + 1$, and Δ_i is the relation $>$ (resp. $\leq$) if $\delta(\xi_i) = 1$ (resp. $\delta(\xi_i) = -1$). Next, we determine all non-empty sign conditions of the set $\mathcal{Q} = \{Q_1 \ldots Q_s\}$. See Fact A.2 in the Appendix for an algorithm achieving this. For constant n, and this can be done in polynomial time in either the unit cost or the logarithmic cost model. Now, we check each non-empty sign condition to see if it corresponds to the given dichotomy Δ, i.e. if all the $(Q_i\ \Delta_i\ 0)$ hold. If there is no match, we report a failure. Otherwise, we output the representation of the coefficients $c_1, \ldots, c_n, r_1, \ldots, r_n$. ∎

6 Pseudo-Dimension Bounds

In this section, we obtain results on the learnability of linear systems dynamics, that is, the class of functions obtained if one does *not* take the sign when defining recurrent perceptrons. The connection between VC dimension and sample complexity is only meaningful for classes of Boolean functions; in order to obtain learnability results applicable to real-valued functions one needs metric entropy estimates for certain spaces of functions. These can be in turn bounded through the estimation of Pollard's pseudo-dimension. We next briefly sketch the general framework for learning due to Haussler (based on previous work by Vapnik, Chervonenkis, and Pollard) and then compute a pseudo-dimension estimate for the class of interest.

The basic ingredients are two complete separable metric spaces $\mathbb{X}$ and $\mathbb{Y}$ (called respectively the sets of inputs and outputs), a class $\mathcal{F}$ of functions $f : \mathbb{X} \to \mathbb{Y}$ (called the decision rule or hypothesis space), and a function $\ell : \mathbb{Y} \times \mathbb{Y} \to [0, r] \subset \mathbb{R}$ (called the loss or cost function). The function ℓ is so that the class of functions $(x, y) \mapsto \ell(f(x), y)$ is "permissible" in the sense of Haussler and Pollard. (This means that these functions must be Borel-measurable, and in addition measurability holds in a certain parametric sense uniformly over the class; the reader may consult [12] for details, but in any case, the assumptions hold in our application). The interest will be in choosing a function $f \in \mathcal{F}$, based on "training" data given by a sequence of "input/output samples"

$$w \;=\; w_1, \ldots, w_s \;=\; (u_1, v_1), \ldots, (u_s, v_s) \;\in\; (\mathbb{X} \times \mathbb{Y})^s \tag{9}$$

so that f can predict the "correct" output v associated to a new input u. It is the goal that the hypothesis f minimize the expected value of the loss $\ell(f(u), v)$, assuming that both the training data and the new input/output pair are picked according to the same probability distribution.

More precisely, let W be the set of all sequences as in Equation (9), with arbitrary $s \geq 0$; then an *identifier* is a map $\varphi : W \to \mathcal{F}$. The value of φ on a sequence $w \in W$ will be denoted by φ_w. Assume that P is any probability measure defined on the Borel sets of $\mathbb{X} \times \mathbb{Y}$. For any $f \in \mathcal{F}$, one may consider the expected error

$$\mathrm{Err}_f(P) := E_{(u,v)\in W}[\ell(f(u), v)]$$

(expectation being understood with respect to the distribution P) and in particular, given an identifier φ one may define its error after training, for any given sequence $w \in W$:

$$\mathrm{Err}_\varphi(P, w) = \mathrm{Err}_{\varphi_w}(P)\,.$$

Finally, there is a best-possible error when using functions in $\mathcal{F}$:

$$\mathrm{Err}_{\mathcal{F}}(P) := \inf_{f\in\mathcal{F}} \mathrm{Err}_f(P)\,.$$

The class $\mathcal{F}$ is said to be (uniformly) *learnable* with respect to the loss function ℓ if there is some identifier φ with the following property: for each $0<\varepsilon,\delta<1$ there is some integer $s(\varepsilon,\delta)$ so that, for each $s\geq s(\varepsilon,\delta)$ and each Borel probability distribution P on $\mathbb{X}\times\mathbb{Y}$,

$$\mathrm{Prob}\left[\mathrm{Err}_\varphi\left(P,(w_1,\ldots,w_s)\right)\;>\;\mathrm{Err}_{\mathcal{F}}(P)+\varepsilon\right]\;<\;\delta$$

(the probability being understood with respect to P^s on $(\mathbb{X}\times\mathbb{Y})^s$).

In the learnable case, the function $s(\varepsilon,\delta)$ which provides, for any given ε and δ, the smallest possible s as above, is called the *sample complexity* of the class $\mathcal{F}$. In partial analogy to the role of VC dimension for Boolean functions, the study of sample complexity for the present learning problem is related to another combinatorial quantity, namely Pollard's pseudo-dimension, which can be defined as follows (the property that we give is trivially equivalent to the one in e.g. [12, 14], but it is somewhat easier to check): Given a class of functions $\mathcal{F}$ from $\mathbb{X}$ to $\mathbb{Y}$, and a function $\ell:\mathbb{Y}\times\mathbb{Y}\to\mathbb{R}$, one may introduce, for each $f\in\mathcal{F}$, the function

$$A_{f,\ell}\;:\;\mathbb{X}\times\mathbb{Y}\times\mathbb{R}\to\{-1,1\}\;:\;(x,y,t)\mapsto\mathrm{sign}\left(\ell(f(x),y)-t\right)$$

as well as the class $\mathcal{A}_{\mathcal{F},\ell}$ consisting of all such $A_{f,\ell}$. The *pseudo-dimension* of $\mathcal{F}$ with respect to the loss function ℓ, denoted by $\mathrm{PD}\left[\mathcal{F},\ell\right]$, is defined as:

$$\mathrm{PD}\left[\mathcal{F},\ell\right]\;:=\;\mathrm{VC}\left(\mathcal{A}_{\mathcal{F},\ell}\right).$$

It is known (cf. [12, Lemma 1 in pp. 100 and Corollary 2 in pp. 114], as well as the discussion in [14, Theorem 4.1 and Remark 4.2*b*]) that a sufficient condition for $\mathcal{F}$ to be learnable with respect to ℓ is that $\mathrm{PD}\left[\mathcal{F},\ell\right]<\infty$, in which case one may pick

$$s(\varepsilon,\delta)\;\leq\;\frac{576r^2}{\varepsilon^2}\left(2.\mathrm{PD}\left[\mathcal{F},\ell\right]\,\ln\frac{48er}{\varepsilon}\;+\;\ln\frac{8}{\delta}\right)\,.$$

Moreover, in analogy with the identifier that picks any consistent concept, an identifier in this case can be obtained by empirical risk minimization: to achieve the above error for a given δ,ε, and given the samples $w=(u_1,v_1),\ldots,(u_s,v_s)$ with $s\geq s(\varepsilon,\delta)$, it is enough to pick φ_w to be any function $f\in\mathcal{F}$ with the property that:

$$\left|\sum_{i=1}^{s}\ell(f(u_i),v_i)-\inf_{h\in\mathcal{F}}\sum_{i=1}^{s}\ell(h(u_i),v_i)\right|<\frac{\varepsilon}{3}\,.$$

(This entails, in practice, the approximate solution of an optimization problem over $\mathcal{F}$, minimizing error over the training samples; in fact, a probabilistic algorithm can be used in that it is only necessary that this estimate holds with high probability, cf. [12, Lemma 1]). Of course, actually performing this minimization may be a nontrivial computational task and we will not deal with this question, which is related, in our application, to the topic of dynamic covers and Hankel approximations.)

In the application that interests us, we define, for any two nonnegative integers n, q, the class

$$\mathcal{F}'_{n,q} \;:=\; \left\{ \widehat{\phi}_{\vec{c}} \,\middle|\, \vec{c} \in \mathbb{R}^{n+q} \text{ is } n\text{-recursive} \right\}$$

where

$$\widehat{\phi}_{\vec{c}} \;:\; \mathbb{R}^{n+q} \to \mathbb{R} \;:\quad (x_1, \ldots, x_{n+q}) \;\mapsto\; \sum_{i=1}^{n+q} c_i x_i \;.$$

Lemma 6.1 Let p be a positive integer and assume that the loss function ℓ is given by $\ell(y_1, y_2) = |y_1 - y_2|^p$. Then,

$$\text{PD}\left[\mathcal{F}'_{n,q}, \ell\right] \leq 7.76n + 4n \log(p(q+1)) \;.$$

Proof. By definition, $\text{PD}\left[\mathcal{F}'_{n,q}, \ell\right] \leq \text{VC}\left(\mathcal{F}''_{n,q}\right)$, where, for each $\vec{c}$,

$$\psi_{\vec{c}} \;:\; \mathbb{R}^{n+q+2} \to \mathbb{R} \;:\quad (x_0, x_1, \ldots, x_{n+q}, y) \;\mapsto\; \ell\left(\sum_{i=1}^{n+q} c_i x_i, y\right) - x_0 \;,$$

and

$$\mathcal{F}''_{n,q} \;:=\; \left\{ \text{sign}\left(\psi_{\vec{c}}\right) \mid \; \vec{c} \in \mathbb{R}^{n+q} \text{ is } n\text{-recursive} \right\} .$$

The required bound follows from Fact 4.1, applied with $k = 2n$, $s = 4$, $d = p(q+1)$: $\text{VC}\left(\mathcal{F}''_{n,q}\right) \leq 4n \log(32ep(q+1)) < 7.76n + 4n \log(p(q+1))$. ∎

Note that, in order to apply this pseudo-dimension estimate to a learning problem, it is necessary to restrict inputs and parameters in such a manner that the values $\ell\left(\sum_{i=1}^{n+q} c_i x_i, y\right)$ fall in a finite range $[0, r]$. Alternatively, one can easily modify the above proof to deal with a bounded error function such as $\ell(y_1, y_2) = |y_1 - y_2|^p/(1 + |y_1 - y_2|^p)$ or $\ell(y_1, y_2) = \min\{|y_1 - y_2|^p, 1\}$.

A Appendix: Representations of Real Numbers and Decision Problems

We collect here some facts regarding Thom encodings of real numbers and their use in decision problems for real-closed fields.

Let $f(x)$ be a real univariate polynomial of degree d, and let α be a real root of f. The *Thom encoding* of α relative to $f(x)$, denoted $\text{Th}\,(\alpha, f)$, or just $\text{Th}\,(\alpha)$ if f is clear from the context, is the sign vector

$$\left(\text{sg}[f(\alpha)], \text{sg}[f'(\alpha)], \ldots, \text{sg}[f^{(d)}](\alpha))\right) \;\in\; \{-1, 0, 1\}^{d+1}$$

where $\text{sg}[x] = x/|x|$ if $x \neq 0$ and $\text{sg}[0] = 0$. It is known (cf. [7]) that $\text{Th}\,(\alpha, f)$ uniquely characterizes α among the roots of f.

In this paper, by a *representation* of a vector $(y_1, y_2, \ldots, y_k) \in \mathbb{R}^k$ we mean a vector

$$(f(t), g_0(t), \ldots, g_k(t), \rho)$$

consisting of:

(a) a univariate polynomial $f(t)$,

(b) $k+1$ univariate polynomials $g_0(t), \ldots, g_k(t)$, and

(c) a vector $\rho \in \{-1, 0, 1\}^{\deg(f)+1}$,

so that ρ is the Thom encoding $\mathrm{Th}\,(\alpha)$ of some root α of f, and $y_i = \frac{g_i(\alpha)}{g_0(\alpha)}$ for each $1 \leq i \leq k$. The polynomials are represented by vectors providing their degrees and listing all coefficients. When dealing with the logarithmic cost model, we assume in addition that the coefficients of the polynomials f and g_i are all rational numbers. In the unit cost model, the *size* of such a representation is defined to be the total number of reals needed so as to specify the coefficients, that is, the sum of the degrees of all the polynomials plus $k+3+\deg(f)$. In the logarithmic cost model, the size is the above plus the total number of bits needed in order to represent the coefficients of the polynomials, each written in binary as the quotient of two integers.

In the paper, we use these representations for the parameters defining concepts, while inputs are given directly as real numbers (rationals in the log model); thus we need to know that signs of polynomial expressions involving vectors represented in the above manner as well as reals can be evaluated efficiently. We next state a result that assures this. By the *complexity* of a multivariable polynomial $H(z_1, \ldots, z_q)$ we mean the sum of the number of nonzero monomials plus the sum of the total degrees of all these monomials (for instance, $2z_1^2z_2^3 - z_1^7$ has complexity $2+5+7 = 14$); in the log cost model, we assume that the coefficients of H are rational and we add the number of bits needed to represent the coefficients.

Lemma A.1 In the unit cost model, there is an algorithm $\mathcal{A}$ which, given a polynomial H of complexity h on variables $x_1, \ldots, x_l, y_1, \ldots, y_k$, and given real numbers $x_1, \ldots, x_l$ and a representation $(f(t), g_0(t), \ldots, g_k(t), \rho)$ of a vector $y_1, \ldots, y_k$, can compute $\mathrm{sg}[H(x_1, \ldots, x_l, y_1, \ldots, y_k)]$ in time polynomial on l, h, and the size of this representation. The same result holds in the logarithmic cost model, assuming that the inputs x_i are all rational, with time now polynomial on the size of these inputs as well. $\square$

Proof. Note that, in general, if $p_1(t)$ and $p_2(t)$ are two rational functions with numerator and denominators of degree bounded by d, then both $p_1(t)p_2(t)$ and $p_1(t) + p_2(t)$ are rational functions with numerator and denominator of degree at most $2d$. Moreover, these algebraic operations can be computed in time polynomial on d as well as, in the log model, on the size of coefficients. Working iteratively on all monomials of H, we conclude that it is possible to construct from the g_i's and x_j's, in polynomial time, two polynomials $R_1(t)$ and $R_2(t)$ with real (rational, in the log model) coefficients so that $H(x_1, \ldots, x_l, y_1, \ldots, y_k) = R_1(\alpha)/R_2(\alpha)$, where α is the root encoded by ρ. Note that

$$\mathrm{sign}\left(\frac{R_1(\alpha)}{R_2(\alpha)}\right) = \begin{cases} 1 & \text{if sign}\,(R_1(\alpha)) = \text{sign}\,(R_2(\alpha)) \text{ and } R_1(\alpha) \neq 0 \\ -1 & \text{otherwise} \end{cases}$$

Thus it is only necessary to evaluate sign $(R_i(\alpha))$, $i = 1, 2$. The evaluation can be done efficiently because of the following fact from [16]:

> There is an algorithm $\mathcal{B}$ with the following property. Given any univariate real polynomial $f(t)$, a real root α of f specified by means of its Thom encoding Th (α), and another univariate polynomial $g(t)$, $\mathcal{B}$ outputs sign $(g(\alpha))$, using a number of arithmetic operations polynomial on $\deg(f) + \deg(g)$; in the logarithmic cost model, if all input coefficients are rationals of size at most L, then $\mathcal{B}$ uses a number of bit operations polynomial on $\deg(f) + \deg(g) + L$.

This provides the desired $\mathrm{sg}[H(x_1, \ldots, x_l, y_1, \ldots, y_k)]$. ∎

The main reason that representations of the type $(f(t), g_0(t), \ldots, g_k(t), \rho)$ are of interest is that one can produce solutions of algebraic equations and inequalities represented in that form. We explain this next.

One says that a vector $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_s) \in \{-1, 0, +1\}^s$ is a *nonempty sign condition* for an ordered set of s real polynomials $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_s\}$ in $k < s$ real variables if there exists some point $(y_1, \ldots, y_k) \in \mathbb{R}^k$ such that $\sigma_i = \mathrm{sg}[\mathcal{P}_i(y_1, y_2, \ldots, y_k)]$ for all i; the corresponding point $(y_1, y_2, \ldots, y_k) \in \mathbb{R}^k$ is said to be a *witness* of σ.

Fact A.2 ([4, 5]) There is an algorithm $\mathcal{A}$ as follows. Given any set $\mathcal{P}$ of s real polynomials in $k < s$ variables, where each polynomial is of degree at most d, $\mathcal{A}$ computes, for each non-empty sign-condition of $\mathcal{P}$, the sign condition σ as well as a representation of a witness for σ. Moreover, $\mathcal{A}$ runs in $O((sd)^{O(k)})$ time in the unit cost model, and in the corresponding representation, $\deg(f) \leq (sd)^{O(k)}$. In the logarithmic cost model, assuming that coefficients of the given polynomials are rationals of size at most L, $\mathcal{A}$ runs in time $O(s^k d^{O(k)} L^{O(1)})$, and the degrees and coefficients of all the polynomials $f, g_0, \ldots, g_k$ (and, consequently the number of components in Th (α)) are rational numbers of size at most $O(d^{O(k)} L^{O(1)})$. □

References

[1] A.D. Back and A.C. Tsoi, *FIR and IIR synapses, a new neural network architecture for time-series modeling*, Neural Computation, 3 (1991), pp. 375–385.

[2] A.D. Back and A.C. Tsoi, *A comparison of discrete-time operator models for nonlinear system identification*, Advances in Neural Information Processing Systems (NIPS'94), Morgan Kaufmann Publishers, 1995, to appear.

[3] A.M. Baksho, S. Dasgupta, J.S. Garnett, and C.R. Johnson, *On the similarity of conditions for an open-eye channel and for signed filtered error adaptive filter stability*, Proc. IEEE Conf. Decision and Control, Brighton, UK, Dec. 1991, IEEE Publications, 1991, pp. 1786–1787.

[4] S. Basu, R. Pollack, and M.-F. Roy, *A New Algorithm to Find a Point in Every Cell Defined by a Family of Polynomials*, in *Quantifier Elimination and Cylindrical Algebraic Decomposition*, B. Caviness and J. Johnson eds., Springer-Verlag, to appear.

[5] S. Basu, R. Pollack, and M.-F. Roy,*On the Combinatorial and Algebraic Complexity of Quantifier Elimination*, Proc. 35th IEEE Symp. on Foundations of Computer Science, 1994.

[6] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth, *Learnability and the Vapnik-Chervonenkis dimension*, J. of the ACM, 36 (1989), pp. 929-965.

[7] M. Coste and M.F. Roy, *Thom's Lemma, the Coding of Real Algebraic Numbers and the Computation of the Topology of Semialgebraic sets*, J. Symbolic Computation, 5 (1988), pp. 121-129.

[8] D.F. Delchamps, *Extracting State Information from a Quantized Output Record*, Systems and Control Letters, 13 (1989), pp. 365-372.

[9] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[10] C.E. Giles, G.Z. Sun, H.H. Chen, Y.C. Lee, and D. Chen, *Higher order recurrent networks and grammatical inference*, Advances in Neural Information Processing Systems 2, D.S. Touretzky, ed., Morgan Kaufmann, San Mateo, CA, 1990.

[11] P. Goldberg and M. Jerrum, *Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers,* Machine Learning, 18, (1995): 131-148.

[12] D. Haussler, *Decision theoretic generalizations of the PAC model for neural nets and other learning applications*, Information and Computation, 100, (1992): 78-150.

[13] R. Koplon and E.D. Sontag, *Linear systems with sign-observations*, SIAM J. Control and Optimization, 31(1993): 1245 - 1266.

[14] W. Maass, *Perspectives of current research about the complexity of learning in neural nets*, in *Theoretical Advances in Neural Computation and Learning*, V.P. Roychowdhury, K.Y. Siu, and A. Orlitsky, editors, Kluwer, Boston, 1994, pp. 295-336.

[15] G.W. Pulford, R.A. Kennedy, and B.D.O. Anderson, *Neural network structure for emulating decision feedback equalizers*, Proc. Int. Conf. Acoustics, Speech, and Signal Processing, Toronto, Canada, May 1991, pp. 1517–1520.

[16] M.-F. Roy and A. Szpirglas, *Complexity of Computation on Real Algebraic Numbers*, J. Symbolic Computation (1990) 10: 39-51.

[17] E.D. Sontag, *Neural networks for control*, in *Essays on Control: Perspectives in the Theory and its Applications* (H.L. Trentelman and J.C. Willems, eds.), Birkhauser, Boston, 1993, pp. 339-380.

[18] György Turán, *Computational Learning Theory and Neural Networks: A Survey of Selected Topics*, in *Theoretical Advances in Neural Computation and Learning*, V.P. Roychowdhury, K.Y. Siu, and A. Orlitsky, editors, Kluwer, Boston, 1994, pp. 243-293.

[19] L.G. Valiant *A theory of the learnable*, Comm. of the ACM, 27, 1984, pp. 1134–1142

[20] V.N. Vapnik, *Estimation of Dependencies Based on Empirical Data*, Springer, Berlin, 1982.