IDIAP RESEARCH REPORT

IDIAP

Martigny - Valais - Suisse

# On the Complexity of Recognizing Regions Computable by Two-Layered Perceptrons

Eddy N. Mayoraz †

IDIAP–RR 98-03

March 1998

revised in September 1998

published in
Annals of Mathematics and Artificial Intelligence **24**-1/4 (1998) 129-153

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

† IDIAP—Dalle Molle Institute for Perceptive Artificial Intelligence, P.O.Box 592, CH-1920 Martigny, Valais, Switzerland Eddy.Mayoraz@idiap.ch

IDIAP Research Report 98-03

# On the Complexity of Recognizing Regions Computable by Two-Layered Perceptrons

Eddy N. Mayoraz

March 1998

revised in September 1998

published in
Annals of Mathematics and Artificial Intelligence **24**-1/4 (1998) 129-153

**Abstract.** This work is concerned with the computational complexity of the recognition of $\mathcal{LP}_2$, the class of regions of the Euclidian space that can be classified exactly by a two-layered perceptron. Some subclasses of $\mathcal{LP}_2$ of particular interest are also studied, such as the class of iterated differences of polyhedra, or the class of regions $V$ that can be classified by a two-layered perceptron with as only hidden units the ones associated to $(d-1)$-dimensional facets of $V$. In this paper, we show that the recognition problem for $\mathcal{LP}_2$ as well as most other subclasses considered here is $NP$-Hard in the most general case. We then identify special cases that admit polynomial time algorithms.

**Acknowledgements**: This work was started while the author was a postdoctoral fellow at RUTCOR — Rutgers University Center for Operations Research — supported by the Swiss National Science Foundation and by DIMACS — Center for Discrete Mathematics and Computer Science. The author is thanksful to Dr. Motakuri Ramana for very helpful discussions and his participation in the proof of Theorem 5.

# 1 Introduction

Classification is a basic ability of multilayer perceptrons and is the common point between most of the applications of these computational models. As far as multilayer perceptrons based on linear threshold processing units with real inputs and a binary output are concerned, it is natural to ask which regions $V$ of the Euclidian space $\mathbb{R}^d$ are *recognized* by such devices, i.e. such that the output of the network is True if and only if the input is in $V$. Defining the *depth* of a multilayer perceptron by the length of the longest path from an input node to an output node, let $\mathcal{LP}_k$ denote the class of regions of $\mathbb{R}^d$ that can be recognized by a multilayer perceptron of depth $k$. The characterization of $\mathcal{LP}_1$ is obvious, since this class contains nothing but half-spaces. The hierarchy $\mathcal{LP}_1 \subset \mathcal{LP}_2 \subset \ldots \subset \mathcal{LP}_k \subset \ldots$ collapses already at level 3, since any finite union of polyhedra in the Euclidian space can be recognized by a depth 3 multilayer perceptron [Lip87, ZAW91].

The only interesting issues in this area are thus related to $\mathcal{LP}_2$ and several of them have been already addressed [GC90, ZAW92, Sho93, TTK93, Gib96, SGM98, BKPM97]. Various necessary conditions, as well as sufficient ones, have been proposed, and the relationships between them have been studied. However, these studies were essentially of geometrical nature, and even if algorithms were sometimes proposed to check the membership of a region in a particular class [ZAW92], there was no attempt made to determine the exact complexity of any of the recognition problems.

The aim of this work is to address the question of the computational complexity of the problem of recognizing whether a given region $V \subset \mathbb{R}^d$ is in $\mathcal{LP}_2$. This issue is of particular interest for example for constructive neural networks algorithms building a single hidden layer during the learning process. A key component of these algorithms is to be able to decide, with the information at hand, whether or not the task to be performed is computable by a single hidden layer architecture. Since the combinatorial complexity is the aspect investigated here, the problem will be expressed in combinatorial terms, by opposition to several other papers on the characterization of $\mathcal{LP}_2$, which were mostly of geometrical flavor. Here and there, the interest of this translation into a combinatorial language will be illustrated by interpreting in terms of classification with a two-layered perceptron some well known results in Boolean function theory.

The paper will be organized as follows. A formalization of the problem, as well as some notations are presented in the next section. In Section 3 we will discuss the difficulty of the global issue and introduce some subclasses of $\mathcal{LP}_2$, of particular interest with regard to computational complexity. The negative complexity results are reported in Section 4 and some tractable situations are analyzed in Section 5. Final remarks and open questions will conclude the paper.

# 2 Formalization of the recognition problem

The functions computed by two-layered perceptrons as considered in the present paper are defined from the $d$ dimensional Euclidian space $\mathbb{R}^d$ onto the Boolean set $\{0,1\}$ and are of the form:

$$f(\boldsymbol{x}) = g(h_1(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})) \tag{1}$$

where $g : \{0,1\}^m \to \{0,1\}$ as well as $h_i : \mathbb{R}^d \to \{0,1\}, i = 1, \ldots, m$ are linear threshold functions, i.e.

$$h_i(\boldsymbol{x}) = \mathrm{sgn}(w_{i0} + \boldsymbol{x}^\top \boldsymbol{w}_i)\ , \tag{2}$$

$$g(\boldsymbol{b}) = \mathrm{sgn}(t_0 + \boldsymbol{b}^\top \boldsymbol{t})\ . \tag{3}$$

The $w_{ij}$ and $t_i, i = 1, \ldots, m, j = 1, \ldots, d$ are the *coefficients* of the linear threshold functions, and the $w_{i0}$ and $t_0$ are the *thresholds*. In what follows, coefficients and thresholds will all be real values. The sign function is defined as $\mathrm{sgn}(r) = 1$ if $r \geq 0$ and $\mathrm{sgn}(r) = 0$ otherwise.

**Remark 2.1** The value $\mathrm{sgn}(0)$ has been set arbitrarily to 1 but this choice has no implications for the class of functions computable by a $k$-layered perceptron, $k > 1$. Indeed, let $\mathrm{sgn}^0$ be defined as sgn except for $\mathrm{sgn}^0(0) = 0$. On the one hand, observe that $t_0$ can always be chosen distinct from $-\boldsymbol{b}^\top \boldsymbol{t}$ for all $\boldsymbol{b} \in \{0,1\}$ and thus, substituting sgn with $\text{sgn}^0$ in the output function (function $g$) does not change anything. On the other hand, the equivalence $\text{sgn}^0(x) = 1 - \text{sgn}(-x)$ shows that sgn and $\text{sgn}^0$ can be interchanged in any function of the intermediate levels (e.g. $h_i$) assuming a modification of its input weights and threshold ($\boldsymbol{w}_i$, $w_{i0}$), its output weights ($t_i$) as well as the threshold $t_0$ of the function $g$.

**Definition 2.1** A region $V \subset \mathbb{R}^d$ is in $\mathcal{LP}_2$ if and only if its characteristic function can be written in the form of Equations (1-3).

For a study of the computational complexity of the recognition of $\mathcal{LP}_2$, the form of the input, i.e. the way region $V \in \mathbb{R}^d$ is provided, has to be precisely specified. This is the purpose of the following section.

## 2.1 From geometry to combinatorics

Let us first introduce some basic terminology in the theory of Boolean functions. A *Boolean function* is a mapping $f : \{0,1\}^m \to \{0,1\}$. A *partial Boolean function* is a mapping $f : D \subset \{0,1\}^m \to \{0,1\}$, and $D$ is called the *domain of* $f$. An *extension* of a partial Boolean function $f$ is any Boolean function that coincides with $f$ on its domain. The vectors in $f^{-1}(1)$ (resp. $f^{-1}(0)$) are called *true points* (resp. *false points*) of $f$. A Boolean function $f$, complete or partial, is *threshold*, if there exists a hyperplane in $\mathbb{R}^m$ separating the true points of $f$ from its false points. Clearly, a partial Boolean function is threshold if and only if it has a threshold extension.

The operators $\wedge$, $\vee$ and $\overline{x}$ (over-line) stand for conjunction, disjunction and negation, and when they are applied to Boolean vectors, a coefficient-wise application of the operator is meant. We will denote by $\boldsymbol{e}^i$ the Boolean vector $e^i_i = 1, e^i_j = 0, \forall j \neq i$. A *literal* is either a Boolean variable or its negation. A conjunction of literals is a *term*, and a *Disjunctive Normal Form* or *DNF*, is a disjunction of terms. Any Boolean function can be expressed by a DNF. For a term $t$ and a DNF $D$, $|t|$ and $|D|$ denotes the number of literals in $t$, and the number of terms in $D$, respectively.

For a set $X$, $2^X$ denotes the set of mappings from $X$ onto $\{0,1\}$. For any subset $V \subset \mathbb{R}^d$, $V^c$ (resp. $V^o$) stand for the closure (resp. the interior) of $V$ according to the usual topology of $\mathbb{R}^d$, $\overline{V}$ denotes the complement $\mathbb{R}^d \backslash V$, while the border of $V$, defined as $V^c \cap \overline{V}^c$, is denoted $V^\delta$. For any collection $\mathcal{E}$ of subsets of $\mathbb{R}^d$:

$$
\begin{aligned}
\widetilde{\mathcal{E}} &= \{V \subset \mathbb{R}^d \mid V \in \mathcal{E} \text{ or } \overline{V} \in \mathcal{E}\}, \\
\mathcal{P}_{\mathcal{E}} &= \{V \subset \mathbb{R}^d \mid V = \bigcap_{i \in I} E_i,\ E_i \in \mathcal{E},\ |I| < \infty\}, \\
\mathcal{U}_{\mathcal{E}} &= \{V \subset \mathbb{R}^d \mid V = \bigcup_{i \in I} P_i,\ P_i \in \mathcal{P}_{\mathcal{E}},\ |I| < \infty\}.
\end{aligned}
$$

Letting $\mathcal{L}$ denote the set of all closed half-spaces of $\mathbb{R}^d$, i.e. $\mathcal{L} = \mathcal{LP}_1$, the elements of $\mathcal{P}_{\mathcal{L}}$ (resp. $\mathcal{P}_{\widetilde{\mathcal{L}}}$) are called *polyhedra* (resp. *pseudo-polyhedra*). The collection $\mathcal{U}_{\widetilde{\mathcal{L}}}$ comprises of unions of finitely many pseudo-polyhedra and in what follows, it will be always assumed that the input $V$ of our problem belongs to $\mathcal{U}_{\widetilde{\mathcal{L}}}$.

A finite subset $\mathcal{H} \subset \mathcal{L}$ is called an *arrangement*. The set $\mathcal{P}_{\widetilde{\mathcal{H}}}$ with the inclusion relation is a finite lattice. The *cells* of $\mathcal{H}$ are the minimal non-empty elements of the lattice $\mathcal{P}_{\widetilde{\mathcal{H}}}$, and $\mathcal{C}_{\mathcal{H}}$ denotes the set of cells of $\mathcal{H}$. For an arrangement $\mathcal{H} = \{H_1, \ldots, H_m\}$, the following natural injective mapping bridges the geometrical problem with combinatorics : $\phi_{\mathcal{H}} : \mathcal{C}_{\mathcal{H}} \to \{0,1\}^m$ is defined so that the $k^{\text{th}}$ component of the Boolean vector $\phi_{\mathcal{H}}(C)$ is 1 if $C \subset H_k$ and 0 if $C \subset \overline{H_k}$. Let us call *domain* of the arrangement $\mathcal{H}$, the subset $\phi_{\mathcal{H}}(\mathcal{C}_{\mathcal{H}})$ of $\{0,1\}^m$ and denoted $D_{\mathcal{H}}$.

Since any element of $\mathcal{U}_{\widetilde{\mathcal{H}}}$ can be expressed as a union of cells of $\mathcal{H}$, $\mathcal{U}_{\widetilde{\mathcal{H}}} = 2^{\mathcal{C}_{\mathcal{H}}}$ and the one-to-one mapping $\phi_{\mathcal{H}}$ from $\mathcal{C}_{\mathcal{H}}$ to $D_{\mathcal{H}}$ provides a one-to-one mapping $\Phi_{\mathcal{H}}$ from $\mathcal{U}_{\widetilde{\mathcal{H}}}$ to $2^{D_{\mathcal{H}}}$ as follows: $\Phi_{\mathcal{H}}(V)(\boldsymbol{b}) = 1$ *iff* $\phi_{\mathcal{H}}^{-1}(\boldsymbol{b}) \subset V$. Finally, to each possible expression of a region $V \in \mathcal{U}_{\widetilde{\mathcal{H}}}$ as a union of elements of $\mathcal{P}_{\widetilde{\mathcal{H}}}$, corresponds in a natural way a DNF expression for the function $\Phi_{\mathcal{H}}(V)$, with one term for each component of the union.

For $V \in \mathcal{U}_{\widetilde{\mathcal{L}}}$, any $\mathcal{H} \subset \mathcal{L}$ such that $V \in \mathcal{U}_{\widetilde{\mathcal{H}}}$ is called a *basis* of $V$. A halfspace $H$ belonging to any basis of $V$ is an *essential halfspace* of $V$. If $\mathcal{H}$ is a basis of a region $V$, any arrangement $\mathcal{G}$ containing $\mathcal{H}$ is also a basis of $V$. Thus, $\mathcal{H} \subset \mathcal{G}$ implies $\mathcal{U}_{\widetilde{\mathcal{H}}} \subset \mathcal{U}_{\widetilde{\mathcal{G}}}$. For two arrangements $\mathcal{H} \subseteq \mathcal{G}$ and a function $f : D_{\mathcal{H}} \to \{0,1\}$, the unique function $g : D_{\mathcal{G}} \to \{0,1\}$ defined as $g = \Phi_{\mathcal{G}}(\Phi_{\mathcal{H}}^{-1}(f))$ is called the *expansion* of $f$ from $D_{\mathcal{H}}$ to $D_{\mathcal{G}}$.

In the remaining part of this paper, it will always be assumed that a region $V \subset \mathbb{R}^d$ is described in a combinatorial way, i.e. by the complete specification of one of its bases $\mathcal{H}$, along with a DNF expressing the partial Boolean function $\Phi_{\mathcal{H}}(V)$. With these definitions, our recognition problem can be stated as follows:

> $\mathcal{LP}_2$-RECOGNITION: Given an arrangement $\mathcal{H} \subset \mathcal{L}$ and a DNF of a partial Boolean function $f : D_{\mathcal{H}} \to \{0,1\}$, is there an arrangement $\mathcal{G} \supseteq \mathcal{H}$ such that the expansion of $f$ from $D_{\mathcal{H}}$ to $D_{\mathcal{G}}$ is threshold ?

To conclude this subsection, let us illustrate the previous definitions through a simple illustration.

## 2.2 A simple example

Consider the region $V = P_1 \cup P_2 \subset \mathbb{R}^2$, where $P_1 = \{\boldsymbol{x} \mid x_1 \geq 1,\ x_2 \geq 0\}$ and $P_2 = \{\boldsymbol{x} \mid x_1 < 0\}$. Since $P_1$ and $P_2$ are clearly 2 pseudo-polyhedra, $V$ is in $\mathcal{U}_{\widetilde{\mathcal{L}}}$. The simplest basis of $V$, i.e. minimal in the sense of inclusion, is $\{H_1, H_2, H_3\}$, where $H_1 = \{\boldsymbol{x} \mid x_1 \geq 0\}$, $H_2 = \{\boldsymbol{x} \mid x_1 \geq 1\}$ and $H_3 = \{\boldsymbol{x} \mid x_2 \geq 0\}$. In this basis, $P_1 = H_2 \cap H_3$ and $P_2 = \overline{H_1}$. Figure 1(a) pictures the region $V$ as well as its basis $\{H_1, H_2, H_3\}$.

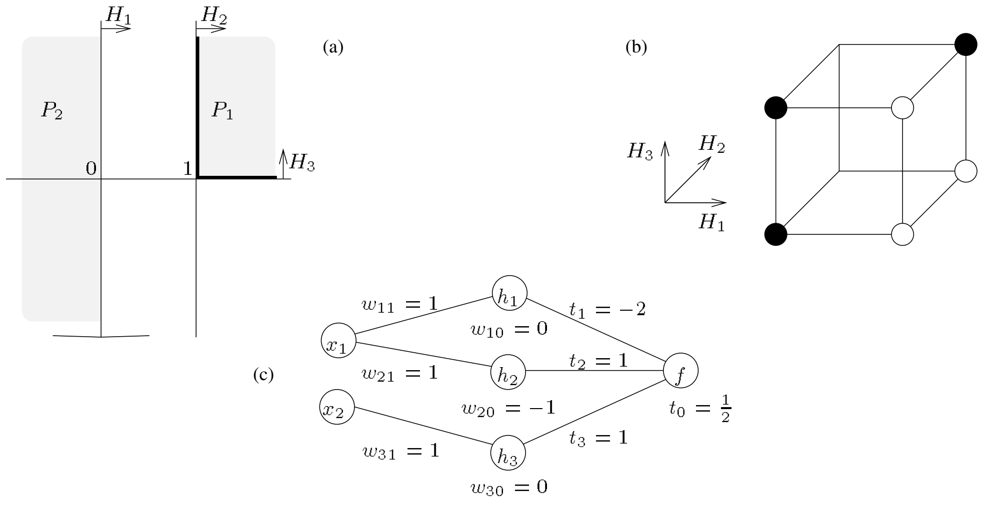


Figure 1: Simple example of region $V \in \mathcal{U}_{\widetilde{\mathcal{L}}}$.

A region $V$ union of two pseudo-polyhedra $P_1$ and $P_2$ is represented in (a), as well as a simple basis $\mathcal{H} = \{H_1, H_2, H_3\}$. (b) illustrates the partial Boolean function defined on $D_{\mathcal{H}} \in \{0,1\}^3$ corresponding to the region $V$. Since this partial Boolean function is threshold, a representation of the two-layered perceptron classifying correctly $V$ is given in (c) where all the non-zero parameters of Equations (2) and (3) are given.

Figure 1(b) illustrates the hypercube $\{0,1\}^3$. $D_{\mathcal{H}}$ is the union of black and white vertices and this bicoloration represents the function $f = \Phi_{\mathcal{H}}(V)$. Observe that in this case, the black points can be separated from the white points by a plane, therefore, $\Phi_{\mathcal{H}}(V)$ is threshold. Thus, there is no need to expand it to a larger arrangement in order to conclude that $V$ is in $\mathcal{LP}_2$. Indeed, if functions $h_1, h_2$ and $h_3$ in Equation (2) are the characteristic functions of $H_1, H_2$ and $H_3$, then $g$ in Equation (3) determined by $t_0 = \frac{1}{2}$ and $\boldsymbol{t} = (-2, 1, 1)$, complete the expression of the characteristic function of $V$ as of the form of Equation (1). The corresponding two-layered perceptron is shown in Figure 1(c).

## 2.3 Monotonic Boolean functions

A Boolean function $f : \{0,1\}^d \to \{0,1\}$ is *monotonic in its $i^{th}$ variable*, if either $f(\boldsymbol{b}) \geq f(\boldsymbol{b} \vee \boldsymbol{e}^i)\ \forall \boldsymbol{b} \in \{0,1\}^d$, or $f(\boldsymbol{b}) \leq f(\boldsymbol{b} \vee \boldsymbol{e}^i)\ \forall \boldsymbol{b} \in \{0,1\}^d$. A *monotonic* function is monotonic in each of its variables. Monotonicity is a well known necessary condition for a Boolean function to be threshold [Mur71]. Moreover, this concept can be extended to partial Boolean functions, simply by requiring that the above inequalities hold only for the $\boldsymbol{b}$ such that the function is defined for both, $\boldsymbol{b}$ and $\boldsymbol{b} \vee \boldsymbol{e}^i$. Obviously, a partial Boolean function which is not monotonic in this sense, has no monotonic extension. The essence of Lemma 1 is that if the monotonicity condition does not hold for a particular function, the property cannot be recovered by expanding the function. To cope with some degenerated cases, an additional technical assumption must be made for this result to hold in general.

**Lemma 1** If a partial Boolean function $f$ defined on $D_{\mathcal{H}}$ for an arrangement $\mathcal{H}$ is non-monotonic due to a variable $i$ and four vectors $\boldsymbol{b}_0^k$ and $\boldsymbol{b}_1^k = \boldsymbol{b}_0^k \vee \boldsymbol{e}^i,\ k = 1, 2$ (say $\boldsymbol{b}_1^1$ and $\boldsymbol{b}_0^2$ are true points and $\boldsymbol{b}_0^1$ and $\boldsymbol{b}_1^2$ are false points of $f$), and if the four cells $C_l^k = \phi_{\mathcal{H}}^{-1}(\boldsymbol{b}_l^k),\ k = 1, 2,\ l = 0, 1$, are such that

$$C_0^{k^c} \cap C_1^{k^c} \supset B^k, \text{ a } (d-1)\text{-dimensional open ball in } H_i^\delta, \tag{4}$$

then $f$ has no monotonic expansion.

Figure 2 illustrates an example of the situation of Lemma 1.

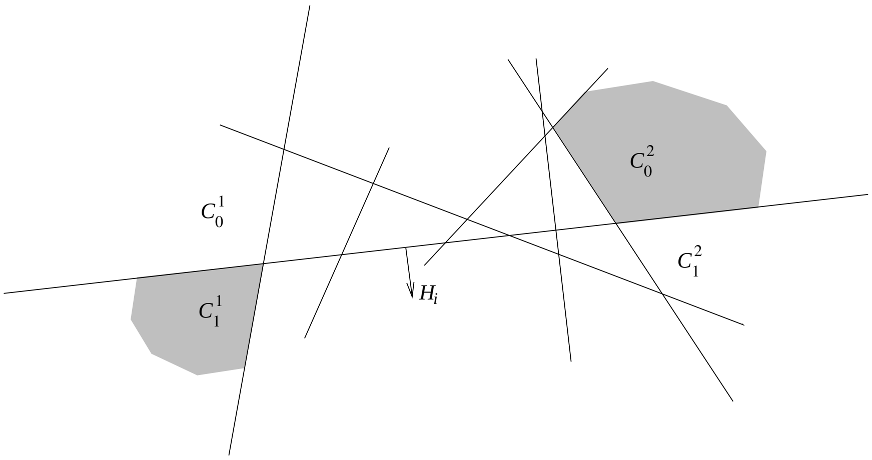


Figure 2: Region providing a Boolean function non-monotonic in variable $i$.

**Proof**: Using mathematical induction, it suffices to show that the expansion of $f$ from $\mathcal{H}$ to $\mathcal{G} = \mathcal{H} \cup \{H\}$ is not monotonic, for any $H \in \mathcal{L} \backslash \mathcal{H}$. The proof is established by showing that for any $H \in \mathcal{L} \backslash \mathcal{H}$, there exist four *non-empty* cells $C_l^k \cap X^k,\ l = 0, 1,\ k = 1, 2$, where

$X^k$ is adequately chosen either as $H$ or as $\overline{H}$. The images of such four cells through $\phi_{\mathcal{G}}$ provide clearly a contradiction to the monotonicity of the expansion of $f$ to $\mathcal{G}$ in variable $i$.

Fix $k$ in $\{1,2\}$. First ($\blacklozenge^1$), suppose that $H$ and $H_i$ do not have the same support hyperplane, i.e. $H^\delta \neq H_i^\delta$. Choose $X^k = \overline{H}$ if $B^k \cap \overline{H} \neq \emptyset$ and $X^k = H$ otherwise. In both cases, it can be verified that $B^k \cap X^{k^o} \neq \emptyset$. This is obvious in the first case, since $\overline{H}$ is already open. In the second case, $B^k \subset H$, and because the support hyperplanes of $H$ and of $H_i$ where lies $B^k$ are different, one concludes that $B^k \subset H^o$. By definition of $B^k$, $B^k \subset C_l^{k^c}$ and thus $C_l^{k^c} \cap X^{k^o} \neq \emptyset$. Since $X^{k^o}$ is open, one conclude $C_l^k \cap X^{k^o} \neq \emptyset$ and thus $C_l^k \cap X^k \neq \emptyset$.

($\blacklozenge^2$), suppose that $H$ and $H_i$ have the same support hyperplane. Since $H \in \mathcal{L}\backslash\mathcal{H}$, the only possibility is that $H = \overline{H_i}^c$. Set $X^k = H$. By definition of $C_0^k$, $C_0^k \subset \overline{H_i}$ and thus $C_0^k \cap H \neq \emptyset$. It remains to prove that the same is true for $C_1^k$. By definition of $C_1^k$, $C_1^k \subset H_i$. ($\blacktriangle^1$) If $B^k \cap C_1^k \neq \emptyset$, the sought property $C_1^k \cap H \neq \emptyset$ holds because $B^k \subset H \cap H_i$. ($\blacktriangle^2$), suppose now that $B^k \cap C_1^k = \emptyset$. By definition, $B^k \subset C_1^{k^c}$, thus $B^k \subset C_1^{k^c} \backslash C_1^k$, which means that in the expression of $C_1^k$ as finite intersection of elements of $\mathcal{H}$ or their complements, appears a term $\overline{H_j}$, where $H_j = \overline{H_i}^c$, i.e. $H_j = H$, which contradicts $H \in \mathcal{L}\backslash\mathcal{H}$. △

The technical assumption (4) is minimal for the purpose of Lemma 1. In other works mentioning a similar result, assumption (4) is replaced by stronger requirements [ZAW91, Gib96]. For example, (4) follows if all the cells of the arrangement are *full-dimensioned*, i.e. contain an open ball of dimension $d$. The so-called "general position" assumption is even stronger than the latter, since it requires that the intersection of any $k$ support hyperplanes of the arrangement is a sub-space of dimension at most $d-k$ if $k \leq d$ and is empty if $k > d$.

**Lemma 2** If a partial Boolean function $f$ defined on $D_{\mathcal{H}}$ for an arrangement $\mathcal{H}$ containing only full-dimensioned cells is non-monotonic, $f$ has no monotonic expansion.

**Proof**: Let say that $f$ is non-monotonic due to 4 cells $C_l^k,\ k=1,2, l=0,1$ defined as in Lemma 1. Since the $C_l^k$ are full-dimensioned, for a fixed $k \in \{1,2\}$, the set of all possible points $\boldsymbol{x} \in H_i^\delta$ and of the form $\boldsymbol{x} = \lambda \boldsymbol{x}_0^k + (1-\lambda)\boldsymbol{x}_1^k$ ($\boldsymbol{x}_l^k \in C_l^k$), is full-dimensioned in $H_i^\delta$ and thus (4) holds. △

Lemma 1 (a similar result can be stated with Lemma 2) conjugated with the fact that monotonicity is a necessary condition for thresholdness, leads to a slight generalization of a well known result (see for example Theorem 1 in [ZAW92]).

**Corollary 3** A region $V \subset \mathbb{R}^d$ is not in $\mathcal{LP}_2$ if for any basis $\mathcal{H}$ of $V$, there exist $H_i \in \mathcal{H}$ and four cells $C_l^k \in \mathcal{C}_{\mathcal{H}},\ k=1,2,\ l=0,1$, such that

- $\phi_{\mathcal{H}}(C_1^k) = \phi_{\mathcal{H}}(C_0^k) \vee \boldsymbol{e}^i,\ k=1,2,$
- $C_1^1 \cup C_0^2 \subset V$ and $C_0^1 \cup C_1^2 \subset \overline{V}$,
- $C_0^{k^c} \cap C_1^{k^c}$ contains a $(d-1)$-dimensional ball, open in the support hyperplane of $H_i$.

This example illustrates the interest of expressing these problems in combinatorial terms and more importantly, it brings some insight on the complexity of our recognition problem, since it is *NP*-Complete to decide whether a Boolean function given by an arbitrary DNF is monotonic or not.

## 2.4 2-summability of Boolean functions

A Boolean function, complete or partial, is 2-*summable* if there exist two true points $\boldsymbol{b}^1$ and $\boldsymbol{b}^2$, not necessarily distinct, and similarly two false points $\boldsymbol{b}^3$ and $\boldsymbol{b}^4$, not necessarily distinct, such that

$$\boldsymbol{b}^1 + \boldsymbol{b}^2 = \boldsymbol{b}^3 + \boldsymbol{b}^4. \tag{5}$$

It is well known that a 2-summable function is not threshold. In fact, the 2-summability notion can be extended to $k$-summability where both sums in (5) contain $k$ components, and a function is threshold *iff* it is not $k$-summable for any $k$. Contrary to the non-monotonicity, the 2-summability property is not necessarily preserved through expansions. Figure 4 in Section 3 will describe a 2-summable function $f : D_{\mathcal{H}} \to \{0,1\}$ with threshold expansions. Nevertheless, 2-summability can play a role in the proof of non-memberships to $\mathcal{LP}_2$, such as in Proposition 4.

**Proposition 4** Let $H_i = \{\boldsymbol{x} \mid x_i \geq 0\}$ and $H'_i = \{\boldsymbol{x} \mid x_i \leq -1\}$ for $i = 1,\dots,d$, be $2d$ hyperplanes two by two parallel. The region $V = \bigcap_{i=1}^d H_i \ \cup \bigcap_{i=1}^d H'_i$ is not in $\mathcal{LP}_2$.

This region $V$ is a generalization to the $d$ dimensional space, of a planar figure already mentioned in the literature. The proof will be established by generalizing the argument given in [Gib96], showing that the example pictured in Figure 3 is not in $\mathcal{LP}_2$.

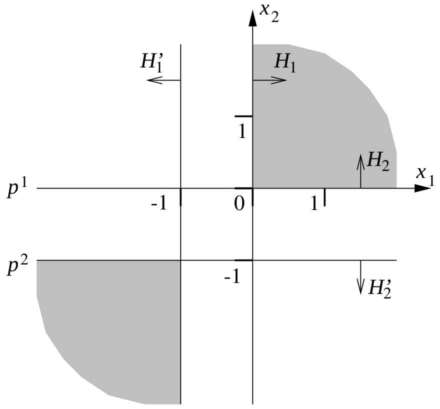


Figure 3: Two-dimensional version of the region of Proposition 4.

**Proof**: Let us assume that $V \in \mathcal{LP}_2$ and denote by $\mathcal{G} = \{H''_1,\dots,H''_m\}$ a basis of $V$, such that $\Phi_{\mathcal{G}}(V)$ is threshold. Consider the following two parallel lines in $\mathbb{R}^d$: $\boldsymbol{p}^1(\alpha) = (\alpha, 0, \dots, 0)$ and $\boldsymbol{p}^2(\alpha) = (\alpha, -1, \dots, -1)$, $\alpha \in \mathbb{R}$, and let $I \subset \{1,\dots,m\}$ be the set of indices of halfspaces of $\mathcal{G}$ containing one line and not intersecting the other. Note that all halfspaces with index in $I$ have their first coefficient $w''_1 = 0$, and that the basic halfspaces $H_i$ and $H'_i$, $i = 2,\dots,d$ are in that collection.

Let $\alpha_+ \geq 0$ (resp. $\alpha_- \leq -1$) be a large enough positive (resp. negative) value such that there is no halfspace in $\mathcal{G}\backslash\{H''_i : i \in I\}$ whose support hyperplane separates $\boldsymbol{p}^1(\alpha_+)$ from $\boldsymbol{p}^2(\alpha_+)$ (resp $\boldsymbol{p}^1(\alpha_-)$ from $\boldsymbol{p}^2(\alpha_-)$). Such two values exist by definition of $I$. Let us partition the remaining indices $\{1,\dots,m\}\backslash I$ into two parts $J$ and $K$ as follows: $H''_i, i \in J$ contains either $\boldsymbol{p}^1(\alpha_+)$ and $\boldsymbol{p}^2(\alpha_+)$, or $\boldsymbol{p}^1(\alpha_-)$, but not the four of them; while $H''_i, i \in K$ either contain the four points, or contain none of them.

The four vectors $\boldsymbol{b}^1_+, \boldsymbol{b}^2_-, \boldsymbol{b}^1_-, \boldsymbol{b}^2_+ \in \{0,1\}^d$, defined as images through $\phi_{\mathcal{G}}$ of the four cells containing the points $\boldsymbol{p}^1(\alpha_+), \boldsymbol{p}^2(\alpha_-), \boldsymbol{p}^1(\alpha_-)$ and $\boldsymbol{p}^2(\alpha_+)$, respectively, satisfy the 2-summability Equation (5). Indeed, $\boldsymbol{p}^1(\alpha) \in V$ for $\alpha \geq 0$, $\boldsymbol{p}^2(\alpha) \in V$ for $\alpha \leq -1$, and

$$(\boldsymbol{b}^1_+ + \boldsymbol{b}^2_-)_i \quad = \quad \left\{ \begin{array}{cl} 1 & \text{if } i \in I \cup J \\ 0 \text{ or } 2 & \text{if } i \in K \end{array} \right\} \quad = \quad (\boldsymbol{b}^1_- + \boldsymbol{b}^2_+)_i \ \ \forall i = 1,\dots,m.$$

This contradicts the assumption that $\Phi_{\mathcal{G}}(V)$ is threshold. $\triangle$

# 3 Subclasses of $\mathcal{LP}_2$ of particular interest

The $\mathcal{LP}_2$-Recognition problem involves two parts of different natures. On the one hand, a "good" super-arrangement $\mathcal{G}$ of $\mathcal{H}$ has to be found. This is essentially a geometrical issue and requires the knowledge on the relative position of the half-spaces of $\mathcal{H}$. On the other hand, one has to be able to check whether a partial Boolean function $g : D_{\mathcal{G}} \to \{0,1\}$ is threshold. This second aspect of the problem is purely combinatorial and requires only the knowledge of the partial Boolean function $g$.

A very simple region $V$ is presented in Figure 4 to illustrate these two parts of our recognition problem. The basis $\mathcal{H}$ of $V$ is composed of the six half-spaces $\mathcal{H} = \{H_{+,+}, H_{-,+}, H_{-,-}, H_{+,-}, H_+, H_-\}$ where the first four half-spaces are defined by $H_{\imath,\varsigma} = \{\boldsymbol{x} \mid \imath x_1 + \varsigma x_2 \geq 1\}$, $\imath, \varsigma \in \{+,-\}$, and the last two by $H_\imath = \{\boldsymbol{x} \mid \imath x_2 \leq \delta\}$, $\imath \in \{+,-\}$. $V$ is defined as the union of the two facing triangles $P_+, P_-$ with height controlled by the parameter $\delta$: $P_\imath = H_{+,\imath} \cap H_{-,\imath} \cap H_\imath$, $\imath \in \{+,-\}$. The images of $P_+$ and $P_-$ through $\Phi_{\mathcal{H}}$ are the vertices $\boldsymbol{b}^+ = (1,1,0,0,1,1)$ and $\boldsymbol{b}^- = (0,0,1,1,1,1)$ respectively. The two unbounded regions in $H_+ \cap H_-$ are mapped onto the vertices $\boldsymbol{b}^1 = (1,0,0,1,1,1)$ and $\boldsymbol{b}^2 = (0,1,1,0,1,1)$. Since $\boldsymbol{b}^+ + \boldsymbol{b}^- = (1,1,1,1,2,2) = \boldsymbol{b}^1 + \boldsymbol{b}^2$, the partial Boolean function $\Phi_{\mathcal{H}}(V)$ is 2-summable and consequently it is not threshold. However, $V \in \mathcal{LP}_2$, and two possible ways of enlarging the basis $\mathcal{H}$, for any arbitrary $\delta$, in order to obtain a threshold expansion of $f$ are presented in Figure 4(a) and (b).

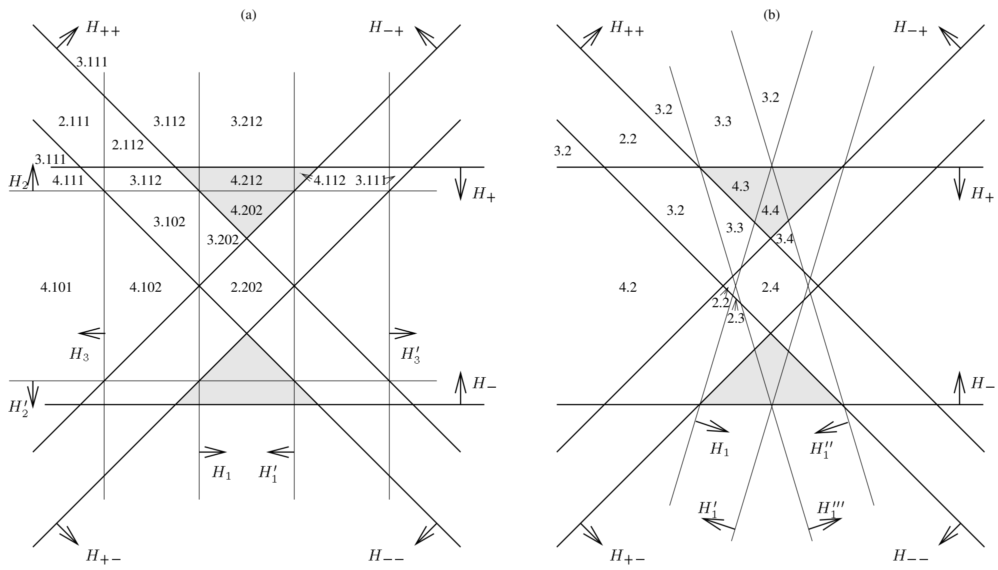


Figure 4: Two constructions solving the two triangles problem.

Two different super-arrangements $\mathcal{G}$ are proposed. The numbers in each cell of $\mathcal{G}$ indicates the value of the dot product $\boldsymbol{b}^\top \boldsymbol{t}$ in Equation (3) for the following choice of $\boldsymbol{t}$: in (a), if $\mathcal{G} = \{H_{+,+}, H_{-,+}, H_{-,-}, H_{+,-}, H_+, H_-, H_1, H_1', H_2, H_2', H_3, H_3'\}$, $\boldsymbol{t} = (1,1,1,1,1,1,\frac{1}{10},\frac{1}{10},\frac{1}{100},\frac{1}{100},\frac{1}{1000},\frac{1}{1000})$; in (b), if $\mathcal{G} = \{H_{+,+}, H_{-,+}, H_{-,-}, H_{+,-}, H_+, H_-, H_1, H_1', H_1'', H_1'''\}$, $\boldsymbol{t} = (1,1,1,1,1,1,\frac{1}{10},\frac{1}{10},\frac{1}{10},\frac{1}{10})$. Clearly, for a threshold $t_0 = -4.1115$ in (a) and $t_0 = -4.25$ in (b), the region $V$ is correctly recognized in both cases. We intentionally chose weights as powers of 10 in order to illustrate the sequence of introduction of each group of half-spaces, and thus to give an idea on how this process can be extended for any $\delta$. By considering the integer parts of different powers of 10 times the values reported in the cells, one can read on these two pictures the remaining error at each step, which consists in cells of $V$ and cells out of $V$ with the same values.

This simple example in the plane suggests that $\mathcal{LP}_2$-Recognition is likely hard, and in particular, it might not be in *NP*. Indeed, the two super-arrangements proposed in Figure 4(a) and (b) are of linear size with the parameter $\delta$ of region $V$, that is, the number of hidden units of the multilayered perceptrons is growing exponentially with a compact encoding of the instance $V$ of $\mathcal{LP}_2$-Recognition.

However, we did not succeed neither in finding a super-arrangement with polylogarithmic size in $\delta$, nor in proving polynomial lower bounds on the size of super-arrangements leading to a threshold expansion. These issues motivated the consideration of subclass of $\mathcal{LP}_2$ introduced in the following subsection.

## 3.1 Two-layered computation with respect to a basis

**Definition 3.1** A region $V \in \mathbb{R}^d$ defined on a basis $\mathcal{H}$ is said to be recognized *on the basis* $\mathcal{H}$ by a two-layered perceptron if its characteristic function can be expressed in the form of Equation (1), where each $h_i$ is a characteristic function of one element of $\mathcal{H}$. A region $V \in \mathbb{R}^d$ is in the class $\overline{\mathcal{LP}_2}$ if there exists a basis $\mathcal{H}$ of $V$, minimal in the sense of the inclusion, such that $V$ is recognized on $\mathcal{H}$ by a two-layered perceptron.

Note that this definition assumes the existence of a minimal basis with the desired property. It is worth mentioning that some region $V$ have more than one minimal basis and Figure 5 illustrates a region that can be recognized by a two-layered perceptron according to one basis but not to another one. However, in the particular case where $V$ is a union of full-dimensioned pseudo-polyhedra (i.e. with non-empty interior in $\mathbb{R}^d$), it can be shown that $V$ has a unique minimal basis.

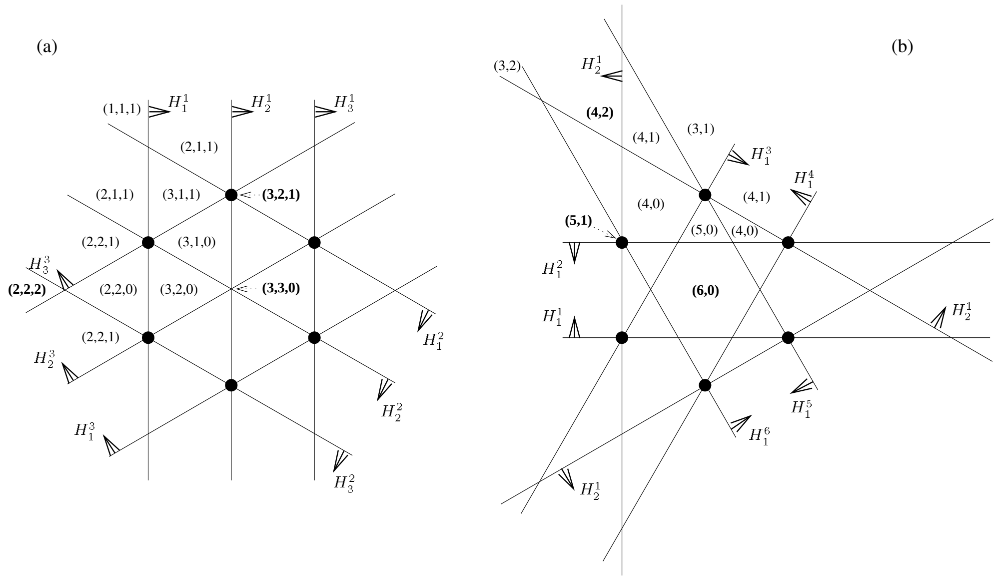


Figure 5: Impact of the choice of the basis for a region $V$.

Two different bases are proposed for a region $V$ composed of six points in the plane, such that none of them is in the convex hull of the others. Since each point is a degenerate polyhedron defined by at least 3 closed half-planes, a basis of $V$ will contain at least $\frac{3\times 6}{2} = 9$ elements (each half-plane can be used for at most 2 points). Thus, the two bases are minimal. Rotations of $2\pi/3$ preserve both figures, and in addition, figure (b) is unchanged by a symmetry along the horizontal axis. Thus, there is no need to distinguish half-planes with the same subscript. Each cell of the arrangement in (a) is labeled by a triplet $(n_1, n_2, n_3)$, where $n_j$ is the number of half-planes $H_j^i$ containing this cell. In figure (b) each cell is labeled by a pair in a similar way. The first arrangement leads to a threshold partial Boolean function $f$ by the following choice: the weights 3,1,2 are associated to half-planes $H_1^i$, $H_2^i$, $H_3^i$ respectively, and the threshold $t_0$ is $-12.5$. On the contrary, a 2-summability situation occurs in (b) with cells of $V$ labeled $(5,1)$ and cells out of $V$ labeled $(6,0)$ and $(4,2)$, since $(6,0)+(4,2)=(5,1)+(5,1)$.

From the definition of $\overline{\mathcal{LP}_2}$ it follows that $\overline{\mathcal{LP}_2} \subset \mathcal{LP}_2$ and this inclusion is proper since there are examples in $\mathcal{LP}_2 \backslash \overline{\mathcal{LP}_2}$ such as the one of Figure 4.

## 3.2 Iterated differences of polyhedra

Another interesting subclass of regions of $\mathcal{LP}_2$ is provided by a necessary condition for two-layered perceptron computation, proposed independently in two different studies [ZAW92, Sho93].

**Definition 3.2** A region $V \subset \mathbb{R}^d$ is an *iterated difference of polyhedra* (resp. pseudo-polyhedra) if it can be expressed as $V = P_1 \backslash (P_2 \backslash (\ldots P_k) \ldots)$, where $P_i \in \mathcal{P}_{\mathcal{L}}$ (resp. $P_i \in \mathcal{P}_{\tilde{\mathcal{L}}}$), $i = 1, \ldots, k$. The class of iterated differences of polyhedra (resp. pseudo-polyhedra) is denoted $\mathcal{D}$ (resp. $\tilde{\mathcal{D}}$).

For example, the region $V = \overline{H_1} \cup (H_2 \cap H_3)$ of Figure 1 is in the class $\mathcal{D}$ since it can be expressed as $\mathbb{R}^d \backslash (H_1 \backslash (H_2 \cap H_3))$.

The inclusion $\tilde{\mathcal{D}} \subset \mathcal{LP}_2$ relies on the fact that $P \backslash V \in \mathcal{LP}_2$ for any pseudo-polyhedron $P$ and any region $V \in \mathcal{LP}_2$, which follows directly from two simple properties of threshold functions, namely

1. the negation of a threshold function is threshold;
2. the conjunction between a single variable and a threshold function is threshold.

In [ZAW92], the authors proposed the following algorithm for the recognition of class $\mathcal{D}$, where the operator $\Theta$ is the closure of the convex hull, denoted $\text{conv}^c$.

| | |
|---|---|
| input: | $V \subset \mathbb{R}^d$; |
| initialization: | $V_0 := V$; $l := 0$; |
| main loop: | `while` $V_l \neq \emptyset$ `and` ( $l < 2$ `or else` $P_l \neq P_{l-1}$) `loop` |
| | $l := l + 1$; |
| | $P_l := \Theta(V_{l-1})$; |
| | $V_l := P_l \backslash V_{l-1}$; |
| | `end loop` |
| output: | $P_1 \backslash (P_2 \backslash (\ldots P_{l-1} \backslash (P_l \backslash V_l)) \ldots) \;=:\; V$ |

`Algo`$(\Theta)$: Recognition of iterated differences of polyhedra.

Obviously, if `Algo`$(\text{conv}^c)$ stops with $V_l = \emptyset$, $V$ is an iterated difference of polyhedra. The authors proved that the converse is also true (Theorem 3 in [ZAW92]), i.e. that if $V \in \mathcal{D}$, `Algo`$(\text{conv}^c)$ stops with $V_l = \emptyset$. However, they only conjectured that `Algo`$(\text{conv}^c)$ could not cycle, or in other words, that if $V \notin \mathcal{D}$, it will always stop with $P_l = P_{l-1} \neq \emptyset$.

At a first glance, one might believe that choosing $\Theta$ simply as the convex hull would lead to an algorithm `Algo`(conv) for the recognition of $\tilde{\mathcal{D}}$, but as mentioned by the authors, the convex hull of the difference between two pseudo-polyhedra is not necessarily a pseudo-polyhedron (e.g. a closed halfspace $H$ minus a half-hyperplane of its support). Moreover, with Algo$(\text{conv}^c)$ in mind we cannot conclude that $\mathcal{D} \subset \overline{\mathcal{LP}}_2$, since the computation of the convex hull will add non essential halfspaces. Finally, the main weakness of Algo$(\text{conv}^c)$ is its complexity, given that

- there is no proof that it always stops,
- even if $V \in \mathcal{D}$, there is no bound on the number of iterations,
- the computation of the convex hull is exponential in $d$.

Most of these drawbacks can be circumvented, if the convex hull is replaced by a more appropriate operator [May97].

**Definition 3.3** Given a collection $\mathcal{E}$ of regions of $\mathbb{R}^d$, the operator $\text{hull}_{\mathcal{E}}$ is defined as follows :

$$\forall X \subset \mathbb{R}^d, \quad \text{hull}_{\mathcal{E}}(X) = \bigcap_{E \in \mathcal{E},\ E \supset X} E \ .$$

There are relationships between convex hull and operators $\mathrm{hull}_{\mathcal{L}}$ and $\mathrm{hull}_{\widetilde{\mathcal{L}}}(X)$ (see [May96]). It can be shown that $\mathrm{conv}^c \equiv \mathrm{hull}_{\mathcal{L}}$, while $\mathrm{conv}(X) \subseteq \mathrm{hull}_{\widetilde{\mathcal{L}}}(X)$, with a proper inclusion in the pathological cases identified in [ZAW92] where $\mathrm{conv}(X)$ is not a pseudo-polyhedron.

Thus, operator $\mathrm{hull}_{\widetilde{\mathcal{L}}}$ solves the problem highlighted in [ZAW92] and requiring the usage of $\mathrm{conv}^c$ instead of conv in `Algo`. The transcription of Theorem 3 in [ZAW92] for $\mathtt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{L}}})$ is straightforward and it proves that $\mathtt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{L}}})$ stops with $V_l = \emptyset$ if and only if $V \in \tilde{\mathcal{D}}$.

Similarly, for any arrangement $\mathcal{H} \subset \mathcal{L}$, the same theorem can be translated to show that $\mathtt{Algo}(\mathrm{hull}_{\mathcal{H}})$ (resp. $\mathtt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{H}}})$) stops with $V_l = \emptyset$ if and only if $V$ is an iterated difference of polyhedra belonging to $\mathcal{P}_{\mathcal{H}}$ (resp. pseudo-polyhedra belonging to $\mathcal{P}_{\widetilde{\mathcal{H}}}$). The result of Proposition 5 is more interesting but its proof is long and technical and is reported in the annex.

**Proposition 5** $\mathtt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{H}}})$ stops with $V_l = \emptyset$ if and only if $V \in \mathcal{U}_{\widetilde{\mathcal{H}}} \cap \tilde{\mathcal{D}}$.

In particular, this result implies that if a region $V$ can be expressed as an iterated difference of arbitrary pseudo-polyhedra, it can also be expressed as an iterated difference of pseudo-polyhedra, all belonging to $\mathcal{P}_{\widetilde{\mathcal{H}}}$, for an arbitrary basis $\mathcal{H}$ of $V$. In other words:

**Corollary 6** $\tilde{\mathcal{D}} \subset \overline{\mathcal{LP}}_2$.

Proposition 5 means that deciding whether a region $V$ belongs to $\tilde{\mathcal{D}}$ can be done using $\mathtt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{H}}})$, where $\mathcal{H}$ is an arbitrary basis of $V$. The interest of $\mathtt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{H}}})$ for a finite $\mathcal{H}$ lies in its complexity, as it is expressed by the next two lemmas.

**Lemma 7** For a finite $\mathcal{H}$, $\mathtt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{H}}})$ always stops after at most $|\mathcal{H}|$ steps.

**Proof**: First, observe that the sequence $P_1, P_2, \ldots$ is nested decreasing, since for any $l \geq 1$, $P_l \supset P_l \cap V_{l-1}$ implies $\mathrm{hull}_{\widetilde{\mathcal{H}}}(P_l) \supset \mathrm{hull}_{\widetilde{\mathcal{H}}}(P_l \cap V_{l-1})$, which is rephrased as $P_l \supset P_{l+1}$. This inclusion is proper except at the last iteration.

Let $S_l \subset \mathcal{H}$ be the set of halfspaces such that their support hyperplanes have a non-empty intersection with the interior of $P_l$. Since $P_l$ and $P_{l+1} \in \mathcal{P}_{\widetilde{\mathcal{H}}}$ and $P_l \supsetneq P_{l+1}$, the expression of the latter as intersection of elements of $\widetilde{\mathcal{H}}$ must contain at least one element of $S_l$. Thus, $S_l \supsetneq S_{l+1}$, which implies that the total number of iterations is bounded by the number of halfspaces (a halfspace and its complement could not occur simultaneously in the definition of the $P_l$s). $\triangle$

Thus, substituting the closure of the convex hull with $\mathrm{hull}_{\widetilde{\mathcal{H}}}$ allows the recognition of $\tilde{\mathcal{D}}$ instead of $\mathcal{D}$, avoids the problem of the conjecture in [ZAW92] (since no halfspace is added in the algorithm) and even bounds the number of iterations linearly with the input size of the problem.

**Lemma 8** For an arrangement $\mathcal{H}$ in $\mathbb{R}^d$ and a region $X \in \mathcal{U}_{\widetilde{\mathcal{H}}}$, union of $s$ pseudo-polyhedra, the computation of $\mathrm{hull}_{\mathcal{H}}(X)$ (resp. $\mathrm{hull}_{\widetilde{\mathcal{H}}}(X)$) is polynomial in $d$, $|\mathcal{H}|$ and $s$.

**Proof**: Let $\mathcal{G}$ be any finite subset of $\widetilde{\mathcal{L}}$ (e.g. $\mathcal{H}$ or $\widetilde{\mathcal{H}}$). The computation of $\mathrm{hull}_{\mathcal{G}}(X)$ requires that for each halfspace $G \in \mathcal{G}$ and each pseudo-polyhedron $P$ defining $X$, we check if $P \subset G$. This is done by testing if $P \cap \overline{G} = \emptyset$. It requires to check the non-feasibility of a system of at most $|\mathcal{H}|$ inequalities, which can be done by linear programming in a time polynomial in the number of inequalities and the number $d$ of variables. $\triangle$

Lemmas 7 and 8 are positive news for the complexity of the recognition of $\tilde{\mathcal{D}}$ using $\mathtt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{H}}})$, since they resolve all the problems identified by the authors of $\mathtt{Algo}(\mathrm{conv}^c)$ in [ZAW92]. However, it will be shown in the next section that this recognition problem is still intractable, and this is due to the complexity of computing the difference of sets.

# 4 Intractability results

Negative complexity results are presented for the recognition problems of the classes described in Sections 2 and 3, in the general case.

**Remark 4.1** Even if $V \subset \mathbb{R}^d$ is given by a basis $\mathcal{H}$ and a partial function $\Phi_{\mathcal{H}}(V)$ defined on the very special domain $D_{\mathcal{H}}$, it still generalizes the case of complete Boolean function. Indeed, for any arbitrary complete function $f : \{0,1\}^d \rightarrow \{0,1\}$, there is an arrangement $\mathcal{H}$ in $\mathbb{R}^d$ and a region $V \in \mathcal{U}_{\tilde{\mathcal{H}}}$ such that $f = \Phi_{\mathcal{H}}(V)$. For that, it suffices to choose $\mathcal{H}$ composed of $d$ halfspaces so that their support hyperplanes intersect in a single point. Therefore, all the questions we might ask in our setting will be at least as hard as their equivalent formulation for complete Boolean functions.

In particular, since it is *NP*-Hard to decide whether an arbitrary DNF represents a threshold function, $\overline{\mathcal{LP}}_2$-Recognition is *NP*-Hard. On the other hand, the same problem is polynomially solvable if the given DNF is monotonic, so $\overline{\mathcal{LP}}_2$-Recognition might be polynomial in the case of monotonic inputs.

**Lemma 9** $\overline{\mathcal{LP}}_2$-Recognition is in co-*NP*.

**Proof**: Given a region $V \subset \mathbb{R}^d$ and a minimal basis $\mathcal{H}$, let $D_{\mathcal{H}} = T \cup F$ be a partition of the domain of the function $\Phi_{\mathcal{H}}(V)$ into its set of true points and false points. By definition of thresholdness, $\Phi_{\mathcal{H}}(V)$ is threshold if and only if the following system of inequalities in the variables $(t_0, \boldsymbol{t})$ has a feasible solution:

$$\begin{array}{rcll} t_0 + \boldsymbol{b}^\top \boldsymbol{t} & \geq & 0 & \forall \boldsymbol{b} \in T \\ t_0 + \boldsymbol{b}^\top \boldsymbol{t} & < & 0 & \forall \boldsymbol{b} \in F \end{array} \tag{6}$$

By the standard separation theorem the system of inequalities (6) has no solution if and only if the convex hulls of $T$ and $F$ intersect, or in other words, if and only if the following system has a solution:

$$\begin{array}{rcll} \sum_{\boldsymbol{b}^i \in T} \alpha_i \boldsymbol{b}^i & = & \sum_{\boldsymbol{b}^j \in F} \beta_j \boldsymbol{b}^j & \\ \sum_i \alpha_i & = & 1 & \\ \sum_j \beta_j & = & 1 & \\ \alpha_i & \geq & 0 & \forall i \\ \beta_j & \geq & 0 & \forall j \end{array} \tag{7}$$

Since System (7) has $d+2$ equations of rational coefficients, Caratheodory's theorem implies that there exists a solution $(\alpha, \beta)$ with at most $d+2$ non zero variables.

A concise certificate showing that $\overline{\mathcal{LP}}_2$ is in co-*NP* is thus provided by at most $d+2$ non zero values $\alpha_i$ and $\beta_i$ defining a solution of (7), as well as the corresponding boolean points $\boldsymbol{b}^i$. The certificate can be verified in polynomial time, since

- the membership to $D_{\mathcal{H}}$ for each of the $\boldsymbol{b}^i$ given in the certificate can be checked by solving a linear program of $d$ variables and $|\mathcal{H}|$ constraints;
- the outcome of the function $\Phi_{\mathcal{H}}(V)$ for each $\boldsymbol{b}^i$ is obtained by evaluating the DNF given as input;
- the fact that the certificate provides a solution of (7) can be checked in $O(n^2)$ operations.

$\triangle$

In our further considerations on the classes $\mathcal{D}$ and $\tilde{\mathcal{D}}$, the following equality will be useful:

**Lemma 10** For any sequence $X_1 \supset X_2 \supset \ldots \supset X_s$ of subsets of $\mathbb{R}^d$, the following equality hold:

$$X_1 \backslash (X_2 \backslash (\ldots X_s) \ldots) = \biguplus_{i=1}^{s/2} (X_{2i-1} \cap \overline{X_{2i}}) \uplus \begin{cases} \emptyset & \text{if } s \text{ is even} \\ X_s & \text{if } s \text{ is odd} \end{cases} .$$

**Proof**: The equality is obvious for $s = 0, 1, 2$. The general statement follows easily by induction on $s$ (the statement holds for $s+2$ if it holds for $s$). △

**Lemma 11** $\tilde{\mathcal{D}}$-Recognition is in *NP*.

**Proof**: First, observe that if $A = A_1 \backslash (A_2 \backslash (\ldots A_k) \ldots)$ for any sets $A, A_1, \ldots, A_k$, there exists some sets $A'_1, \ldots, A'_k$ such that $A'_1 \supset \ldots \supset A'_k$ and $A = A'_1 \backslash (A'_2 \backslash (\ldots A'_k) \ldots)$. This sets can be obtained for example by setting $A'_1 = A_1$ and by choosing $A'_i = A_i \cap A'_{i-1}$ iteratively for $i = 2, \ldots, k$. Consequently, if the $A_i$ are pseudo-polyhedra, we can find pseudo-polyhedra $A'_i$ with the inclusion property.

A certificate showing that a region $V$ is in $\tilde{\mathcal{D}}$ is provided by the sequence of pseudo-polyhedra $P_1, \ldots, P_k$ such that $P_1 \supset P_2 \supset \ldots P_k$, and $V = P_1 \backslash (P_2 \backslash (\ldots P_k) \ldots)$. The length $k$ of this sequence is at most $|\mathcal{H}|$ by Lemma 7, so the certificate is of polynomial size in $d$ and $|\mathcal{H}|$.

The inclusion $A \subset B$ between two pseudo-polyhedra can be checked by solving $|B|$ linear programs of $d$ variables and $|A| + 1$ constraints, where $|X|$ for a pseudo-polyhedron $X$ denotes the number of half-spaces used in the intersection defining $X$. So we can check in a time polynomial in $d$ and $|\mathcal{H}|$, whether the sequence of pseudo-polyhedra given in the certificate is so that $P_1 \supset \ldots \supset P_k$. Then, the iterated difference can be computed using Lemma 10, and it is easy to verify that the whole computation can be done in polynomial time in $d$, $k$ and $|\mathcal{H}|$. △

**Theorem 12** $\mathcal{LP}_2$-Recognition and $\mathcal{D}$-Recognition are *NP*-Hard, $\overline{\mathcal{LP}_2}$-Recognition is co-*NP*-Complete, and $\tilde{\mathcal{D}}$-Recognition is *NP*-Complete.

**Proof**: By Lemmas 9 and 11, it remains to show that the four recognition problems are *NP*-Hard. This will be done by a reduction to SAT.

An instance of SAT is given as a DNF of a Boolean function $a : \{0,1\}^d \to \{0,1\}$

$$a(\boldsymbol{b}) = \bigvee_{i=1}^{d} ( \bigwedge_{j \in I_i^+} b_j \ \wedge \ \bigwedge_{j \in I_i^-} \overline{b_j} \ ).$$

Consider the Euclidian space $\mathbb{R}^d$ and the arrangement $\mathcal{H}$ containing the following $2d$ half-spaces:

$$H_i = \{\boldsymbol{x} \mid x_i \geq 0\}, \quad H'_i = \{\boldsymbol{x} \mid x_i \leq -1\} \ \ \forall i = 1, \ldots, n.$$

Let us call $b_i$ the Boolean variable associated with the half-space $H_i$ ($[b_i = 1] = \Phi_{\mathcal{H}}(H_i)$), and $b'_i$ the Boolean variable associated with the half-space $H'_i$ ($[b'_i = 1] = \Phi_{\mathcal{H}}(H'_i)$) for all $i = 1, \ldots, d$.

Consider the region $V \subset \mathbb{R}^d$ based on these $2d$ half-spaces, which coincides with $\Phi_{\mathcal{H}}^{-1}(a)$ in the orthant $\bigcap_{i=1}^d \overline{H'_i}$, and which contains the remaining part of the space except orthant $\bigcap_{i=1}^d H'_i$. In other words, $V = \Phi_{\mathcal{H}}^{-1}(f)$, where $f : \{0,1\}^{2d} \to \{0,1\}$ is defined by Equation (8):

$$f(b_1, \ldots, b_n, b'_1, \ldots, b'_n) \ = \ ( \ a(b_1, \ldots, b_n) \wedge (\bigwedge_{i=1}^{d} \overline{b'_i}) \ ) \ \vee \ \bigvee_{i \neq j} b'_i \overline{b'_j}. \qquad (8)$$

Since $\mathcal{D} \subset \tilde{\mathcal{D}} \subset \overline{\mathcal{LP}_2} \subset \mathcal{LP}_2$, the 4 complexity results are proved if the two following statements hold:

(i) $V \in \mathcal{D}$ if $a$ is a tautology;

(ii) $V \notin \mathcal{LP}_2$ if there is a $\boldsymbol{b} \in \{0,1\}^d$ such that $a(\boldsymbol{b}) = 0$.

Let us first prove (i). If $a$ is a tautology:

$$f(\boldsymbol{b}) \quad = \quad (\bigwedge_{i=1}^{d} \overline{b'_i}) \ \vee \ \bigvee_{i \neq j} b'_i \overline{b'_j} \quad = \quad \bigvee_{i=1}^{d} \overline{b'_i}.$$

and $V = \Phi_{\mathcal{H}}^{-1}(f)$ is simply given by

$$V \quad = \quad \bigcup_{i=1}^{d} \overline{H'_i} \quad = \quad \mathbb{R}^d \setminus \bigcap_{i=1}^{d} H'_i,$$

which is obviously in $\mathcal{D}$.

To prove (ii), let us assume that there is at least one vector $\boldsymbol{b} \in \{0,1\}^d$ for which $a$ is false.

Case 1: $\exists \boldsymbol{b}^0 \in \{0,1\}^d, a(\boldsymbol{b}^0) = 0$, with $b_j^0 = 0$ for at least one $j$.

Let us consider the following two vectors of $\{0,1\}^{2d}$:

$$\begin{aligned} \boldsymbol{b}^1 &= ( \quad 0,\ldots,0 \quad ,1,\ldots,1) \\ \boldsymbol{b}^2 &= ( \qquad \boldsymbol{b}^0 \qquad ,0,\ldots,0). \end{aligned}$$

With $f$ defined by Equation (8), we have $f(\boldsymbol{b}^1) = f(\boldsymbol{b}^2) = 0$ and $f(\boldsymbol{b}^1 - \boldsymbol{e}'_j) = f(\boldsymbol{b}^2 + \boldsymbol{e}'_j) = 1$, where $\boldsymbol{e}'_j$ is the vector with all components 0 except the $d+j^{\text{th}}$ which is 1. Thus, the condition of Proposition 3 is fulfilled and $V \notin \mathcal{LP}_2$.

Case 2: $a(1,\ldots,1) = 0$, and $\forall \boldsymbol{b} \in \{0,1\}^d, \boldsymbol{b} \neq (1,\ldots,1), a(\boldsymbol{b}) = 1$.

The region $V = \Phi_{\mathcal{H}}^{-1}(f)$ is the whole space $\mathbb{R}^d$ except the two opposite orthants $\bigcap_{i=1}^d H_i$ and $\bigcap_{i=1}^d H'_i$. In other words, it is the complement of the region of Proposition 4 and consequently it is not in $\mathcal{LP}_2$.

$\triangle$

# 5 Tractable cases

The complexity results of Theorem 12 rely essentially on the fact that it is hard to find a DNF expression for the complement of a region defined by a DNF, since it is already hard to decide whether this complement is empty or not (SAT). Let us first analyze the complexity of several of our recognition problems whenever not only $V$ is available as input, but also $\overline{V}$.

## 5.1 $V$ and $\overline{V}$ are available

**Remark 5.1** Note that even though the number of cells of an arrangement does grow exponentially with $d$, in many practical situations, $\Phi_{\mathcal{H}}(V)$ and $\Phi_{\mathcal{H}}(\overline{V})$ can be provided by two DNFs whose total size is polynomial in $d$ and $|\mathcal{H}|$, for any arbitrary basis $\mathcal{H}$ of $V$. For example, the class of *regular* functions introduced in [Win62] is an important class of Boolean functions with the property that the complement $\overline{f}$ of a function $f$ can be expressed by a DNF of size polynomial in $d$ and in the size of any DNF of $f$ [PS85].

In what follows, a slightly more general setting will be used. Suppose that two disjoint regions $V$ and $W$ are given by a common basis $\mathcal{H}$ and two DNFs for $\Phi_{\mathcal{H}}(V)$ and $\Phi_{\mathcal{H}}(W)$. The question related to a class of regions $\mathcal{X}$ will be to decide whether there exists a region $S \in \mathcal{X}$, which is a *sandwich region* between $V$ and $\overline{W}$, i.e. such that $V \subset S \subset \overline{W}$.

This makes more sense with respect to the applications, since in most practical cases we have a positive area that has to be separated from a negative area, and there might be some intermediate regions without any constraints. For the computational complexity point of view, this setting is also more convenient, since it is easy to verify that $V \cap W = \emptyset$, while checking whether $V \cup W = \mathbb{R}^d$ is *NP*-Hard. Indeed, in the particular case of the remark at the beginning of Section 4, the problem is to check whether the DNF $D_V \vee D_W$ is a tautology or not, where $D_X$ is the DNF given for $\Phi_{\mathcal{H}}(X),\ X = V, W$.

**Proposition 13** Given a basis $\mathcal{H}$ and two DNFs expressions $D_1$ and $D_2$, it can be checked in polynomial time whether there is a region $S$ in $\mathcal{D}$ ( resp. in $\tilde{\mathcal{D}}$) such that $\Phi_{\mathcal{H}}^{-1}(D_1) \subset S \subset \overline{\Phi_{\mathcal{H}}^{-1}(D_2)}$.

**Proof**: Let $V$ and $W$ denote $\Phi_{\mathcal{H}}^{-1}(D_1)$ and $\Phi_{\mathcal{H}}^{-1}(D_2)$ respectively. Consider the following variation of $\texttt{Algo}(\Theta)$.

| | |
|---|---|
| input: | $V, W \subset \mathbb{R}^d$; |
| initialization: | $V_0 := V$; $V_{-1} := W$; $l := 0$; |
| main loop: | **while** $V_l \neq \emptyset$ **and** ( $l < 2$ **or else** $P_l \neq P_{l-1}$) **loop** |
| | $l := l + 1$; |
| | $P_l := \Theta(V_{l-1})$; |
| | $V_l := P_l \cap V_{l-2}$; |
| | **end loop** |
| output: | $P_1 \backslash (P_2 \backslash (\ldots P_{l-1} \backslash (P_l \backslash V_l)) \ldots)$ |

$\texttt{Algo}'(\Theta)$: Recognition of iterated differences of polyhedra, without computation of sets differences.

The remaining part of the proof holds for any operator $\Theta$ such that $\Theta(X) \supseteq X$, $X \subset Y \Rightarrow \Theta(X) \subset \Theta(Y)$ and $\Theta(\Theta(X)) = \Theta(X)$. $P_{l+1} = \Theta(P_l \cap V_{l-2}) \subset \Theta(P_l) = P_l$, thus $\texttt{Algo}'$ produces a nested decreasing sequence of sets, which means that Lemma 10 can be used and more importantly, that the same argument as in Lemma 7 applies to show that the number of steps is linear in $|\mathcal{H}|$. Moreover, other relations characterizing $\texttt{Algo}$ are still valid in $\texttt{Algo}'$, e.g.

$$V_l \cap \overline{V_{l-1}} = \emptyset. \tag{9}$$

Indeed, by induction, $V_{l-1} \cap \overline{V_{l-2}} = \emptyset$, and by construction $V_l \subset V_{l-2}$, thus $V_l \cap \overline{V_{l-1}} = \emptyset$.

The interest of $\texttt{Algo}'$ is that it does not compute any difference of sets. The computation of the intersection of two regions based on $\mathcal{H}$ is achieved by computing the intersection of all pairs of pseudo-polyhedra from the two regions. The computation of each such intersection requires the resolution of a linear program with as many inequality constraints as halfspaces in the two pseudo-polyhedra. Moreover, by Lemma 8, the computation of $P_l$ from $V_{l-1}$ is polynomial for both $\Theta = \mathrm{hull}_{\mathcal{H}}$ and $\Theta = \mathrm{hull}_{\tilde{\mathcal{H}}}$.

To complete the proof, it remains to verify that the output of $Algo'(\Theta)$ fulfill the sandwich condition for the input $V, W$, i.e. it contains $V$ and it is contained into $\overline{W}$. This is obvious for $l = 0, 1$, so let proceed by induction. Using Lemma 10, it suffices to show the following statements.

$$\begin{aligned}
P_l \cap \overline{V_l} &\supset V_{l-1}, && (10)\\
(P_{l-1} \cap \overline{P_l}) \cup V_l &\supset V_{l-2}, && (11)\\
(P_{l-1} \cap \overline{P_l}) \cup V_l &\subset P_{l-1} \cap \overline{V_{l-1}}, && (12)\\
(P_{l-2} \cap \overline{P_{l-1}}) \cup (P_l \cap \overline{V_l}) &\subset (P_{l-2} \cap \overline{V_{l-2}}). && (13)
\end{aligned}$$

These properties allow to make a single or a double induction step, depending whether $l$ is odd or even and whether one has to prove that the output of $\texttt{Algo}'(\Theta)$ contains $V$ or is a subset of $\overline{W}$. The verification of these four statements is straightforward. $V_{l-1} \subset P_l$, by a basic property of $\Theta$, and $V_{l-1} \subset \overline{V_l}$ by (9), which proves (10).

$$(P_{l-1} \cap \overline{P_l}) \cup V_l = (P_{l-1} \cap \overline{P_l}) \cup (P_l \cap V_{l-2}).$$

Distributing the union of the two intersections into an intersection of four unions and simplifying each of these unions, one get

$$(P_{l-1} \cap \overline{P_l}) \cup V_l = P_{l-1} \cap (\overline{P_l} \cup V_{l-2}). \tag{14}$$

Since $V_{l-2}$ clearly belongs to the two sides of the intersection in the right-hand-side of (14), this proves (11). The right-hand-side of (14) can also be rewritten as $\overline{P_{l-1} \cap \overline{P_l \cap \overline{V_{l-2}}}}$. Since $V_{l-1} \subset P_l$ and from (9), $V_{l-1} \subset \overline{V_{l-2}}$, then $V_{l-1} \subset P_l \cap \overline{V_{l-2}}$, or equivalently $\overline{V_{l-1}} \supset \overline{P_l \cap \overline{V_{l-2}}}$, which achieves the proof of (12).

$$\begin{aligned}(P_{l-2} \cap \overline{P_{l-1}}) \cup (P_l \cap \overline{V_l}) &= (P_{l-2} \cap \overline{P_{l-1}}) \cup (P_l \cap (\overline{P_l} \cup \overline{V_{l-2}}))\\ &= (P_{l-2} \cap \overline{P_{l-1}}) \cup (P_l \cap \overline{V_{l-2}})\\ &= P_{l-2} \cap (\overline{P_{l-1}} \cup (P_l \cap \overline{V_{l-2}})).\end{aligned}$$

Since $P_{l-1} \supset V_{l-2}$, $\overline{V_{l-2}}$ contains the outer parenthesis of the last equation above. This concludes the proof of (13). △

In the remaining part of this section, the complexity of $\overline{\mathcal{LP}}_2$-Recognition is addressed in the sandwich setting. It is well known that if $D_1$ is a positive DNF (no literal is negated) and $D_2$ is a negative DNF (all literals appear negated), it is easy to decide whether there is a threshold function $f$ such that $\overline{D_2} \geq f \geq D_1$. Indeed, such an $f$ exists if and only if there is a hyperplane having each minimal true point of each term of $D_1$ on its positive side and each maximal true point of each term of $D_2$ on its negative side. The quest for such a hyperplane is reduced to the resolution of a linear program with one inequality for each term in $D_1$ and $D_2$. Proposition 14 states that this problem is still easily solvable when $D_1$ and $D_2$ are not given in monotonic forms.

**Proposition 14** Given two arbitrary DNFs $D_1$ and $D_2$, it can be decided in polynomial time whether there is a sandwich function $f$, i.e. $\overline{D_2} \geq f \geq D_1$, which is threshold.

**Proof**: In order to characterize the most critical true point of a term, without assuming anything on the monotonicity of the function in any particular variable $i$, we need to split every coefficient $t_i$ of Equation (3) into two variables $t_i^+$ and $t_i^-$, with the property that $t_i^+ t_i^- = 0$ and $t_i = t_i^+ - t_i^-$. With that decomposition in mind, consider the following linear program:

$$t_0 + \sum_{i \in I^+} (t_i^+ - t_i^-) \; - \sum_{i \notin I^+ \cup I^-} t_i^- \quad > 0 \qquad \forall t \text{ in } D_1 \;\; t = \bigwedge_{i \in I^+} b_i \wedge \bigwedge_{i \in I^-} \overline{b_i} \tag{15}$$

$$t_0 + \sum_{i \in I^+} (t_i^+ - t_i^-) \; + \sum_{i \notin I^+ \cup I^-} t_i^+ \quad < 0 \qquad \forall t \text{ in } D_2 \;\; t = \bigwedge_{i \in I^+} b_i \wedge \bigwedge_{i \in I^-} \overline{b_i} \tag{16}$$

$$t_0,\ t_i^+,\ t_i^- \quad \geq 0 \quad . \tag{17}$$

We will complete the proof by showing that this system of inequalities has a solution $(t_0, \boldsymbol{t}^+, \boldsymbol{t}^-)$ if and only if $(t_0, \boldsymbol{t}^+ - \boldsymbol{t}^-)$ separates the true points of $D_1$ from the true points of $D_2$.

First observe that if the system (15-17) is feasible, then it has a solution satisfying $t_i^+ t_i^- = 0$ for every $i = 1, \ldots, |\mathcal{H}|$. Indeed, if $t_i^+$ and $t_i^-$ are both non-zero in a feasible solution, the solution will remain feasible if the $t_i^\varsigma$ are replaced by $t_i^\varsigma - \min\{t_i^+, t_i^-\}$, for $\varsigma = +, -$. Therefore, each pair $(t_i^+, t_i^-)$ in a solution of (15-17) is completely determined by the difference $t_i = t_i^+ - t_i^-$.

Assume that the system (15-17) has a solution $(t_0, \boldsymbol{t})$. For the sake of contradiction, suppose that there is one true point $\boldsymbol{b}$ of $D_1$ for which Equation (3) evaluates to 0 (the case when a true point of $D_2$ is evaluate to 1 is treated similarly). Let $t = \bigwedge_{i \in I^+} b_i \ \wedge\ \bigwedge_{i \in I^-} \overline{b_i}$ be a term of $D_1$ which has $\boldsymbol{b}$ as true point. In particular, this implies that $b_i = 1$, $\forall i \in I^+$, and $b_i = 0$, $\forall i \in I^-$. With $J^+ = \{i \mid i \notin I^+ \cup I^-$, and $t_i > 0\}$ and $J^- = \{i \mid i \notin I^+ \cup I^-$, and $t_i < 0\}$, we get:

$$
\begin{aligned}
0 &> t_0 + \boldsymbol{b}^\top \boldsymbol{t} \\
&= t_0 + \sum_{i \in I^+} t_i + \sum_{i \in J^+} t_i^+ b_i - \sum_{i \in J^-} t_i^- b_i \\
&\geq t_0 + \sum_{i \in I^+} t_i - \sum_{i \in J^-} t_i^-,
\end{aligned}
$$

which contradicts the feasibility of the solution $(t_0, \boldsymbol{t})$, since the inequality of type (15) corresponding to this term $t$ is not satisfied.

On the contrary, assume that $(t_0, \boldsymbol{t})$ separates correctly the true points of $D_1$ from those of $D_2$. Then, for any term $t = \bigwedge_{i \in I^+} b_i \ \wedge\ \bigwedge_{i \in I^-}$ of $D_1$ (the terms of $D_2$ can be analyzed similarly) and for any true point $\boldsymbol{b}$ of $t$, $t_0 + \boldsymbol{b}^\top \boldsymbol{t} > 0$. In particular, if $b_i = 1$ iff $i \in I^+ \cup J^-$, the previous relation implies that the inequality of type (15) corresponding to $t$ is satisfied. $\triangle$

This result can be rephrased in terms of regions and $\overline{\mathcal{LP}}_2$-Recognition as follows:

**Corollary 15** Given two regions $V$ and $W$ based on a common basis $\mathcal{H}$ and specified by two DNFs $D_1$ for $\Phi_{\mathcal{H}}(V)$ and $D_2$ for $\Phi_{\mathcal{H}}(W)$ such that all the true points of $D_1$ and $D_2$ are in $D_{\mathcal{H}}$, it can be decided in polynomial time whether there exists a sandwich region $S \in \overline{\mathcal{LP}}_2$, $V \subset S \subset \overline{W}$.

For an immediate application of Proposition 14, it is necessary to assume that all true points of $D_1$ and $D_2$ are in $D_{\mathcal{H}}$. Without it, the resolution of the system (15-17) could lead to the wrong conclusion that there is no sandwich $f \in \overline{\mathcal{LP}}_2$ due to true points corresponding to no cell of the arrangement. Although we suspect that this assumption is not necessary, we did not succeed in proving the result without it.

In order to illustrate Corollary 15 and the proof of Proposition 14, let us consider once again the region illustrated in Figure 1. For the basis $\mathcal{H} = \{H_1, H_2, H_3\}$ represented by the Boolean variables $b_1, b_2$ and $b_3$, a possible choice for the two DNFs $D_1$ and $D_2$ for $V$ and $W = \overline{V}$, satisfying the hypothesis of Corollary 15 is

$$D_1 = (\overline{b_1} \wedge \overline{b_2}) \vee (b_1 \wedge b_2 \wedge b_3), \quad D_2 = (b_1 \wedge \overline{b_2}) \vee (b_1 \wedge \overline{b_3}).$$

The linear program of the form (15-17) associated to $D_1$ and $D_2$ is the following:

$$
\begin{array}{llllllll}
t_0 & & & & & & -t_3^- & > 0 \\
t_0 & +t_1^+ & -t_1^- & +t_2^+ & -t_2^- & +t_3^+ & -t_3^- & > 0 \\
t_0 & +t_1^+ & -t_1^- & & & +t_3^+ & & < 0 \\
t_0 & +t_1^+ & -t_1^- & +t_2^+ & & & & < 0\,,
\end{array}
$$

and the solution $(t_0, t_1^+, t_1^-, t_2^+, t_2^-, t_3^+, t_3^-) = (\frac{1}{2}, 0, 2, 1, 0, 1, 0)$, proposed in Figure 1(c), satisfies this set of inequalities.

### 5.2 Euclidian space of fixed size

In many practical situations, the size $d$ of the Euclidian space is small, while the number of halfspaces can be very large. Therefore, in this section on tractable cases, it is worth highlighting that some of the recognition problems discussed in this paper become easy when $d$ is a constant. Even the cases of $d = 2, 3$ themselves are quite interesting for certain applications of Neural Networks such as Computer Vision.

**Proposition 16** $\overline{\mathcal{LP}_2}$-Recognition, $\tilde{\mathcal{D}}$-Recognition and $\mathcal{D}$-Recognition are polynomial in the size of the basis and in the size of the DNF given as input.

**Proof**: The maximal number of cells contained in an arrangement of $h$ hyperplanes in $\mathbb{R}^d, n \leq h$ is given by

$$N(h,n) = \sum_{i=0}^{d} \binom{h}{i},$$

a formula many times rediscovered under various forms, but which was already known to Ludwig Schläfli in the middle of the previous century. This expression proves that the number of cells increases exponentially with $d$, but polynomially with $h$.

Let a region $V \subset \mathbb{R}^d$ be given as input by a basis $\mathcal{H}$ and a DNF expression $D$ for $\Phi_{\mathcal{H}}(V)$. If the set of all cells of $\mathcal{H}$, or equivalently the domain $D_{\mathcal{H}}$, can be enumerated in time polynomial in $|\mathcal{H}|$ and $D$, then this proof follows from Propositions 15 and 13.

To enumerate the set of cells, we can proceed by induction on the size of $|\mathcal{H}|$. If $\mathcal{H} = \emptyset$, $\mathbb{R}^d$ is the unique cell. The list of cells of the arrangement $\mathcal{H} = \{H_1, \ldots, H_h\}$ can be produced by checking for each cell $P$ of the arrangement $\{H_1, \ldots, H_{h-1}\}$, the intersections $P \cap H_h$ and $P \cap \overline{H_h}$, and by introducing in the list the non-empty intersections. Since $N(h,n)$ is monotonic increasing in $h$, the whole process requires only a polynomial computational time. Obviously, if we are just interested in the computation of a DNF for $\Phi_{\mathcal{H}}(\overline{V})$ in order to apply the algorithm in the proof of Proposition 14, this can be done more efficiently than by enumerating the whole domain $D_{\mathcal{H}}$, still by using induction on the number of halfspaces. $\triangle$

# 6 Conclusion and open problems

In this work, we essentially showed that the problem of deciding whether a region $V$ of the Euclidian space can be computed by a two-layered perceptron is difficult. Several other decision problems, variants of this general question, are also examined and it turns out that all of them are also hard, as long as the dimension of the Euclidian space is not fixed, or the complement of the region $V$ is not available as input.

Among the unanswered questions, it is worth mentioning two that we consider of particular interest.

> Does there exist a region $V \subset \mathbb{R}^d$ that can be computed by a two-layered perceptron, but that requires a number of hidden units growing exponentially with a compact binary encoding of the region $V$ ?

Note that even if this question is answered positively for a very simple example such as the one of Figure 4, it does not prove that $\mathcal{LP}_2$-Recognition is hard in some favorable cases provided by a fixed size Euclidian space or by the sandwich framework. Indeed, there could exist a certificate to the existence of a 2-layered network without expressing it. In the case of Euclidian space of size 2, examples of such certificates can be found in [Gib96, SGM98] or in [BKPM97].

The second question is concerned with the generalization, to the particular type of partial functions defined on arrangements, of the problem of deciding whether a monotonic DNF is threshold.

> Is there a polynomial time algorithm deciding on the thresholdness of a monotonic partial Boolean function over $h$ arguments whose domain of definition is given by an arrangement of $h$ hyperplanes in a Euclidian space ?

We are convinced that this interesting issue as well as most of the open questions related to the topic developed in this paper, requires a deeper understanding of the structure of the set of vertices $D_{\mathcal{H}}$. In a first approximation, if the arrangement is in general position in $\mathbb{R}^d$, one can say that

- $D_{\mathcal{H}}$ is a union of $d$-dimensional faces of the hypercube $\{0,1\}^{|\mathcal{H}|}$,
- each of these faces shares a $d-1$-dimensional face with $n$ other such faces, $d \leq n \leq 2d$,
- $D_{\mathcal{H}}$ contains no face of dimension greater than $d$.

But, this is still not enough to characterize completely $D_{\mathcal{H}}$ and probably much more could be said on its structure.

# Annex

**Proposition 5**. $\texttt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{H}}})$ stops with $V_l = \emptyset$ if and only if $V \in \mathcal{U}_{\widetilde{\mathcal{H}}} \cap \tilde{\mathcal{D}}$.

**Proof**: $\texttt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{H}}})$ returns an expression of $V$ as an iterated difference of pseudo-polyhedra of $\mathcal{P}_{\widetilde{\mathcal{H}}}$ and of a residual set $V_l$. Thus, if $V \notin \mathcal{U}_{\widetilde{\mathcal{H}}}$, a fortiori $V_l \neq \emptyset$. So, let assume that $V \in \mathcal{U}_{\widetilde{\mathcal{H}}}$ and show that $\texttt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{L}}})$ stops with $V_l = \emptyset$ if and only if $V \in \tilde{\mathcal{D}}$. The proof will be established by showing that if $\texttt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{L}}})$ stops after $L$ iterations with $V_L = \emptyset$, $\texttt{Algo}(\mathrm{hull}_{\widetilde{\mathcal{H}}})$ stops as well after $L$ iterations, with $V_L = \emptyset$.

To simplify the notations, let denote by $\mathrm{comp}_{\mathcal{E}}$ the operator defined as

$$\mathrm{comp}_{\mathcal{E}}(X) = \mathrm{hull}_{\mathcal{E}}(X) \backslash X.$$

If $\{V_i\}_{i\geq 0}$ and $\{W_i\}_{i\geq 0}$ are two series of sets defined as $V_0 = W_0 = V$, $V_{i+1} = \mathrm{comp}_{\widetilde{\mathcal{H}}}(V_i)$, and $W_{i+1} = \mathrm{comp}_{\widetilde{\mathcal{L}}}(W_i)$, $\forall i \geq 0$, the goal is to show that

$$W_L = \emptyset \;\Rightarrow\; V_L = \emptyset. \tag{18}$$

To prove (18), it is sufficient to show

$$W_i = \mathrm{comp}_{\widetilde{\mathcal{L}}}(V_{i-1}),\ \forall i > 0. \tag{19}$$

Indeed, using (19), $W_L = \emptyset \Rightarrow \mathrm{comp}_{\widetilde{\mathcal{L}}}(V_{L-1}) = \emptyset$. By definition of $\mathrm{comp}_{\widetilde{\mathcal{L}}}$, this implies $\mathrm{hull}_{\widetilde{\mathcal{L}}}(V_{L-1}) = V_{L-1}$, which means that $V_{L-1} \in \mathcal{P}_{\widetilde{\mathcal{L}}}$. But $V_{L-1} \in \mathcal{U}_{\widetilde{\mathcal{H}}}$ by construction of the series $\{V_i\}_{i\geq 0}$, thus $\mathrm{comp}_{\widetilde{\mathcal{H}}}(V_{L-1}) = V_{L-1}$, which implies $V_L = \emptyset$.

The proof of (19) follows if

$$\forall X \in \mathcal{U}_{\widetilde{\mathcal{H}}},\ \mathrm{comp}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{L}}}(X)) = \mathrm{comp}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{H}}}(X)). \tag{20}$$

Let demonstrate that (20) implies (19) by induction on $i$. As $V = V_0 = W_0$, setting $X = V$ in (20), one get $W_2 = \mathrm{comp}_{\widetilde{\mathcal{L}}}(V_1)$. Suppose now that (19) is true for every $i \leq j$. $W_{j+1} = \mathrm{comp}_{\widetilde{\mathcal{L}}}(W_j) = \mathrm{comp}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{L}}}(V_{j-1}))$ by induction. Using (20), one get $W_{j+1} = \mathrm{comp}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{H}}}(V_{j-1})) = \mathrm{comp}_{\widetilde{\mathcal{L}}}(V_j)$, which completes the induction step.

Let try to simplify Relation (20).

$$\begin{aligned}
\mathrm{comp}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{L}}}(X)) &= \mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{L}}}(X)) \cap \overline{\mathrm{comp}_{\widetilde{\mathcal{L}}}(X)} \\
&= (\mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{L}}}(X)) \cap \overline{\mathrm{hull}_{\widetilde{\mathcal{L}}}(X)}) \cup (\mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{L}}}(X)) \cap X).
\end{aligned}$$

The first parenthesis in this last expression is empty, because $\mathrm{comp}_{\widetilde{\mathcal{L}}}(X) \subset \mathrm{hull}_{\widetilde{\mathcal{L}}}(X) \Rightarrow \mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{L}}}(X)) \subset \mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{hull}_{\widetilde{\mathcal{L}}}(X)) = \mathrm{hull}_{\widetilde{\mathcal{L}}}(X)$. Thus,

$$\mathrm{comp}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{L}}}(X)) = \mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{L}}}(X)) \cap X. \tag{21}$$

Similarly, one can derive

$$\mathrm{comp}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{H}}}(X)) = \mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{H}}}(X)) \cap X. \tag{22}$$

and (20) is rephrased as

$$\forall X \in \mathcal{U}_{\widetilde{\mathcal{H}}},\ \mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{L}}}(X)) \cap X = \mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{H}}}(X)) \cap X. \tag{23}$$

The left part is obviously included in the right part, since $\mathrm{comp}_{\widetilde{\mathcal{L}}}(X) \subseteq \mathrm{comp}_{\widetilde{\mathcal{H}}}(X)$. So it remains to prove the other inclusion. Observe that

$$\mathrm{comp}_{\widetilde{\mathcal{L}}}(X) = \mathrm{hull}_{\widetilde{\mathcal{L}}}(X) \cap \overline{X} = \mathrm{hull}_{\widetilde{\mathcal{L}}}(X) \cap \mathrm{hull}_{\widetilde{\mathcal{H}}}(X) \cap \overline{X} = \mathrm{hull}_{\widetilde{\mathcal{L}}}(X) \cap \mathrm{comp}_{\widetilde{\mathcal{H}}}(X),$$

thus the remaining goal is

$$\forall X \in \mathcal{U}_{\widetilde{\mathcal{H}}},\ \mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{H}}}(X)) \cap X \ \subset\ \mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{H}}}(X) \cap \mathrm{hull}_{\widetilde{\mathcal{L}}}(X)) \cap X. \tag{24}$$

Pick $\boldsymbol{x} \in \mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{comp}_{\widetilde{\mathcal{H}}}(X)) \cap X$. $\boldsymbol{x} \in \mathrm{conv}(\mathrm{comp}_{\widetilde{\mathcal{H}}}(X))$ and let $\boldsymbol{x}^i$ denote points in $\mathrm{comp}_{\widetilde{\mathcal{H}}}(X)$ such that $\boldsymbol{x} = \sum_i \lambda_i \boldsymbol{x}^i$, $\lambda_i > 0$, $\sum_i \lambda_i = 1$. $\boldsymbol{x}^i \in \mathrm{comp}_{\widetilde{\mathcal{H}}}(X) \Rightarrow \boldsymbol{x}^i \in \mathrm{hull}_{\widetilde{\mathcal{H}}}(X) \Rightarrow [\boldsymbol{x}, \boldsymbol{x}^i] \subset \mathrm{hull}_{\widetilde{\mathcal{H}}}(X)$, where $[\boldsymbol{x}, \boldsymbol{x}^i]$ denotes the line segment from $\boldsymbol{x}$ to $\boldsymbol{x}^i$.

For any $\boldsymbol{x}^i$, there exists $\boldsymbol{y}^i \in [\boldsymbol{x}, \boldsymbol{x}^i] \cap \mathrm{hull}_{\widetilde{\mathcal{L}}}(X) \cap \overline{X}$. Indeed, if $[\boldsymbol{x}, \boldsymbol{x}^i] \cap \mathrm{hull}_{\widetilde{\mathcal{L}}}(X) \cap \overline{X} = \emptyset$, then the border of $X$ and the border of $\mathrm{hull}_{\widetilde{\mathcal{L}}}(X)$ coincide on the segment $[\boldsymbol{x}, \boldsymbol{x}^i]$ in say $\boldsymbol{z}^i$. This means that there is $H_i \in \widetilde{\mathcal{H}}$, with $\boldsymbol{z}^i \in H_i^\delta$ and $H_i \supset X$. But in this case, $\boldsymbol{x}^i = \boldsymbol{z}^i$ which contradicts the fact that $\boldsymbol{x}^i \in \mathrm{comp}_{\widetilde{\mathcal{H}}}(X)$.

Say $\boldsymbol{y}^i = \alpha_i \boldsymbol{x} + (1-\alpha_i)\boldsymbol{x}^i$, $\alpha \in [0,1)$. Thus,

$$\begin{aligned}
\boldsymbol{x} &= \sum_i \lambda_i \boldsymbol{x}^i \\
&= \sum_i \lambda_i \frac{\boldsymbol{y}^i - \alpha_i \boldsymbol{x}}{1-\alpha_i} \\
\boldsymbol{x} + \sum_i \frac{\alpha_i \lambda_i}{1-\alpha_i}\boldsymbol{x} &= \sum_i \frac{\lambda_i}{1-\alpha_i}\boldsymbol{y}^i \\
\boldsymbol{x} &= \sum_i \frac{\lambda_i}{(1-\alpha_i)(1+\sum_j \frac{\alpha_j \lambda_j}{1-\alpha_j})}\boldsymbol{y}^i.
\end{aligned}$$

It can be easily checked that the coefficients in front of the $\boldsymbol{y}^i$s are non-negative and sum to 1, thus $\boldsymbol{x}$ is a convex combination of the $\boldsymbol{y}^i$s. By definition, $\boldsymbol{y}^i \in \mathrm{hull}_{\widetilde{\mathcal{L}}}(X) \cap \overline{X} = \mathrm{comp}_{\widetilde{\mathcal{L}}}(X)$ and $\mathrm{comp}_{\widetilde{\mathcal{L}}}(X) \subset \mathrm{comp}_{\widetilde{\mathcal{H}}}(X)$. Thus, $\boldsymbol{y}^i \in \mathrm{hull}_{\widetilde{\mathcal{L}}}(X) \cap \mathrm{comp}_{\widetilde{\mathcal{H}}}(X)$, and finally, $\boldsymbol{x} \in \mathrm{conv}(\mathrm{hull}_{\widetilde{\mathcal{L}}}(X) \cap \mathrm{comp}_{\widetilde{\mathcal{H}}}(X)) \subset \mathrm{hull}_{\widetilde{\mathcal{L}}}(\mathrm{hull}_{\widetilde{\mathcal{L}}}(X) \cap \mathrm{comp}_{\widetilde{\mathcal{H}}}(X))$, which completes the proof. $\triangle$

# References

[BKPM97] G. Brightwell, C. Kenyon, and H. Paugam-Moisy. Multilayer neural networks : one or two hidden layers ? In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9 (NIPS*96)*, pages 148–154. MIT Press, 1997.

[GC90] Gavin J. Gibson and Colin F. N. Cowan. On the decision regions of multilayer perceptrons. *Proceedings of the IEEE*, 78(10):1590–1594, 1990.

[Gib96] Gavin J. Gibson. Exact classification with 2-layered nets. *J. Computer and System Sciences*, 1996.

[Lip87] R. P. Lippmann. An introduction to computing with neural nets. *IEEE, Acoustic, Speech and Signal Processing*, 4:4–22, 1987.

[May96] Eddy Mayoraz. On variations of the convex hull operator. IDIAP-RR 96-06, IDIAP, 1996. `ftp://ftp.idiap.ch/pub/mayoraz/Publications/RRR-06-96.ps.gz`.

[May97] Eddy Mayoraz. On the complexity of recognizing iterated differences of polyhedra. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Proceedings of ICANN'97, Lausanne*, number 1327 in Lecture Notes in Computer Science, pages 475–480, 1997.

[Mur71] Saburo Muroga. *Threshold Logic and Its Applications.* John Wiley & Sons, New York, 1971.

[PS85] U. N. Peled and B. Simeone. Polynomial-time algorithms for regular set-covering and threshold synthesis. *Discrete Applied Mathematics*, 12:57–69, 1985.

[SGM98] Catherine Z. W. Hassell Sweatman, Gavin J. Gibson, and Bernard Mulgrew. Exact classification with two-layer neural nets in $n$ dimensions. *Discrete Applied Math*, 81:19–39, 1998.

[Sho93] Ron Shonkwiler. Separating the vertices of n-cubes by hyperplanes and its application to artificial neural networks. *IEEE Transactions on Neural Networks*, 4(2):343–347, March 1993.

[TTK93] Haruhisa Takahashi, Etsuji Tomita, and Tsutomu Kawabata. Separability of internal representations in multilayer perceptrons with application to learning. *Neural Networks*, 6:689–703, 1993.

[Win62] R. O. Winder. *Threshold Logic.* PhD thesis, Department of Mathematics, Princeton University, 1962.

[ZAW91] P. J. Zwietering, E. H. L. Aarts, and J. Wessels. The design and complexity of exact multilayered perceptrons. *International Journal of Neural Systems*, 2:185–199, 1991.

[ZAW92] P. J. Zwietering, E. H. L. Aarts, and J. Wessels. Exact classification with two-layered perceptrons. *International Journal of Neural Systems*, 3(2):143–156, 1992.