On the Complexity of Neural Network Representations

Thesis by
Kordağ Mehmet Kılıç

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2024
Defended December 7, 2023

Kordağ Mehmet Kılıç
ORCID: 0009-0005-6321-7113

ACKNOWLEDGEMENTS

Embarking on a PhD journey stands out as the most pivotal decision of my life, requiring me to relocate overseas and invest five transformative years in the United States. Throughout my life, I will always remember what my advisor, Professor Jehoshua Bruck, said one time in a class, that "Doing a PhD is the greatest gift one can give to oneself." His unwavering support, optimism, wisdom, and guidance were instrumental in shaping this profound experience.

I extend heartfelt gratitude to my collaborator, Jin Sima, whose kindness, assistance, and willingness for countless discussions were invaluable during and after his Caltech years.

I am deeply grateful to my PhD defense committee members —Professor Christopher Umans, Babak Hassibi, Erik Winfree, and Yaser Abu-Mostafa— for encouragement and invaluable feedback which played a crucial role in my academic journey.

Caltech became a second home, enriched by the support of colleagues and friends. The camaraderie within the Paradise Lab, particularly with Siddharth Jain, Netanel Raviv, and Bijan Mazaheri, enhanced my experience. Special mention goes to my friends Şahin Lale, Sinan Kefeli, Oğuzhan Teke, Taylan Kargın, Hikmet Yıldız, and Recep Can Yavaş—your companionship made my time at Caltech memorable.

Beyond Caltech, my circle of friends holds immeasurable significance, and I find myself deeply indebted to their enduring companionship. Cahit Topaktaş, Serdar Atalay, Buğra Felekoğlu, Bulut Aygüneş, Doğukan Yiğit Polat, and Alper Aksoy made my visits to Turkey truly special. Rock and metal concerts in Los Angeles with Emir Ceyani and Berk Tınaz were always a blast. Adem Derinel, Berkan Kılıç, Yiğit Ertuğrul, Semih Kurt, and Sarp Tanrıdağ—your friendship was a cornerstone of my college years. I cherish the memories of occasional hangouts with my brother-in-law Berk Vatansever, Omar Elsafty, Austin Flick, and Sebastian Hendricks-Rodriguez.

I must apologize to those friends not mentioned here, but your impact on my life is not overlooked.

During my time in the U.S., I had the privilege of encountering inspiring artistic souls, and I cherish the friendships forged. Gabriel Reoyo Pazos, Deborah and Isaiah Reoyo, Dennis Wise Rupp, Emma Berggren, and Helen Wu— your friendship, kindness, wisdom, musicality, and intensity will continue to resonate within me.

Last but certainly not least, my family deserves heartfelt appreciation for their unwavering support. My parents, Atilla Kılıç and Kıvılcım Kılıç, laid the foundation for who I am today, and I am eternally grateful. My sister, Karsu İpek Kılıç, has been an inseparable part of my life, and I could not ask for a better sibling. In moments of trouble, my aunt Nihan Sümer has consistently been my go-to advisor, and her support has been a guiding light in both my academic and artistic pursuits. My cousin, Arda Sümer, has been a source of constant laughter and a brother to me since our infancy. My grandparents, Uğur Sümer and Gülşen Sümer, along with my uncle Uğurtan Sümer and aunt Zeynep Sümer, have been consistently kind and supportive throughout.

ABSTRACT

The evolution of the human brain was one of the milestones in the history of information after the emergence of life. The underlying biological, chemical, and physical processes of the brain have amazed scientists for a long time. It is still a mystery how the human brain computes a simple arithmetical operation like 2+2 = 4. This enigma has spurred investigations into understanding the intrinsic architecture of the brain.

This thesis delves into two primary models for brain architecture: Feedforward Neural Networks and Nearest Neighbor (NN) Representations. Both models are treated under the hypothesis that our brain does not work with "large" numbers and expressive power is derived from connectivity. Thus, when examining a network or, more precisely, a single neuron model, we strive to minimize the bit resolution of weights, potentially increasing depth or circuit complexity.

For the NN representations, the memory is defined by a set of vectors in $\mathbb{R}^n$ (that we call *anchors*), computation is performed by convergence from an input vector to a nearest neighbor anchor, and the output is a label associated with an anchor. Limited bit resolution in the anchor entries may result in an increase of the size of the NN representation.

In the digital age, computers universally employ the binary numeral system, ensuring the enduring relevance of Boolean functions. This study specifically explores the trade-off between resolution and size for the computation models for Boolean functions. It is established that "low resolution" models may require a polynomial or even an exponential increase in the size complexity of the "high resolution" model, potentially making the practical implementation infeasible. Building upon prior research, our goal is to optimize these blow-ups by narrowing the gaps between theoretical upper and lower bounds under various constraints. Additionally, we aim to establish connections between NN representations and neural network models by providing explicit NN representations for well-known Boolean functions in Circuit Complexity Theory.

PUBLISHED CONTENT AND CONTRIBUTIONS

Kilic, Kordag Mehmet, Jin Sima, and Jehoshua Bruck (2024a). "Nearest Neighbor Representations of Neural Circuits". In: *arXiv preprint arXiv:2402.08751*. URL: `https://arxiv.org/pdf/2402.08751`.
Kordag Mehmet Kilic participated in the conception of the project, solved and proved mathematical propositions, lemmas, and theorems, and participated in the writing of the manuscript. To be published in ISIT 2024 proceedings.

– (2024b). "Nearest Neighbor Representations of Neurons". In: *arXiv preprint arXiv:2402.08748*. URL: `https://arxiv.org/pdf/2402.08748`.
Kordag Mehmet Kilic participated in the conception of the project, solved and proved mathematical propositions, lemmas, and theorems, and participated in the writing of the manuscript. Accepted by ISIT 2024 but could not be presented in person and withdrawn.

– (2023). "On the Information Capacity of Nearest Neighbor Representations". In: *2023 IEEE International Symposium on Information Theory (ISIT)*, pp. 1663–1668. DOI: `10.1109/ISIT54713.2023.10206832`.
Kordag Mehmet Kilic participated in the conception of the project, solved and proved mathematical propositions, lemmas, and theorems, and participated in the writing of the manuscript.

– (2022). "On Algebraic Constructions of Neural Networks with Small Weights". In: *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, pp. 3007–3012. DOI: `10.1109/ISIT50566.2022.9834401`.
Kordag Mehmet Kilic participated in the conception of the project, solved and proved mathematical propositions, lemmas, and theorems, and participated in the writing of the manuscript.

Kilic, Kordag Mehmet and Jehoshua Bruck (2021). "Neural Network Computations with DOMINATION Functions". In: *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, pp. 1029–1034. DOI: `10.1109/ISIT45174.2021.9517872`.
Kordag Mehmet Kilic participated in the conception of the project, solved and proved mathematical propositions, lemmas, and theorems, and participated in the writing of the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

The evolution of the human brain was one of the milestones in the history of information after the emergence of life. The human brain enabled us the use of natural languages and it was the first instance of which we are aware when an information system was able to begin understanding itself and utilize the surrounding information in many different ways to build, to store, and to share information, similar to DNA but much more extensive. It is still a mystery how the human brain computes a simple arithmetical operation like $2 + 2 = 4$ biologically, chemically, and physically. Its capabilities and capacity are not unlimited, though. Even the first computers were able to outperform human brain in terms of speed and computation precision for even more complex tasks. Nevertheless, at least for image processing and pattern recognition, even a baby was more successful than computers for a long time. This intrigued the researchers who sought to understand the intrinsic architecture of the brain and, therefore, inspired many scientists to develop some mathematical models. In this thesis, we consider two main models for the brain architecture: the first is the feedforward neural networks, which has been well known for decades and is immensely popular; the second is a relatively new associative computation model by Nearest Neighbor Representations. An important observation is that memory happens to be embedded in the connectivity patterns of the neural networks and one can argue that the brain does not obey *von Neumann Computer Architecture* which has a distinction between computation and memory. Considering the integrated architecture of the brain, computation and memory are indistinguishable. Motivated by this, we propose a model of *Associative Computation* where the memory is defined by a set of vectors in $\mathbb{R}^n$ (that we call *anchors*), the computation is performed by convergence from an input vector to a Nearest Neighbor anchor, and the output is a label associated with an anchor. This paradigm relates to the ability of the brain to quickly classify objects and map those to the syntax of natural languages, for example, it is instantaneous and effortless, even for a child, to recognize a ‘dog’, a ‘cat’, or a ‘car’.

1.1 Neural Networks and Complexity Parameters

Historically, the first abstract mathematical models for neurons were developed in the 1940's and 1950's, which are called *perceptrons* (McCulloch and Pitts, 1943; Rosenblatt, 1958). More precisely, a perceptron is a function that computes a weighted summation of real inputs (i.e. $z = \sum_{i=1}^{n} w_i x_i$) and feeds it to a real valued *activation function* $\sigma(z)$. By connecting perceptrons in a big network, it was hypothesized that the computation capability of the brain could have been achieved. There were three important questions, though. Given a target function that we want to compute, how can we adjust or "learn" the perceptron weights? How deep and large should the neural network be? What would be the largest weight size (or the number of bits to represent it) of the network?

The first question is answered by using *backpropagation* to learn a target function in Machine Learning (Rumelhart, Hinton, and R. J. Williams, 1986). The technique relied on perceptrons having differentiable activation functions. Over the years, it became the backbone of the deep learning algorithms and its practical use is shown in many fields including the trending Generative Artifical Intelligence in Natural Language Processing(NLP) and Diffusion models in Image Processing.

In contrast, the answer to the second question did not give any practical answer even though it was theoretically demonstrated that any real-valued function can be approximated by a depth-2 perceptron circuit with *sigmoid* activation function (Cybenko, 1989; Hornik, 1991). This is called the *Universal Approximation Theorem*. Nevertheless, in this result, the number of required neurons in the first layer was unbounded. A complementary result is obtained for Rectified Linear Unit (ReLU) activations, i.e. $\mathrm{ReLU}(z) = \max(0, z)$, where the width of the circuit is bounded but not the depth (Lu et al., 2017; Park et al., 2021). Alternatively, for bounded width and depth, the "ambiguity" can be absorbed into the activation functions themselves (Maiorov and Pinkus, 1999; Guliyev and Ismailov, 2018). Usually, these methods are not constructive in that there is no explicit algorithm to construct a network with determined weights to compute a target function.

There has not been a comprehensive treatment of weight sizes of these kind of neural networks and the weights are typically finite precision real numbers. Theoretically, for some models of neurons, the weights can get exponentially large in the input dimension n (Alon and Vũ, 1997; Babai et al., 2010; Håstad, 1994; Saburo Muroga, 1971). In contrast, it is reasonable to assert that the brain does not use exponentially "large" numbers (or equivalently, real numbers with "high" bit-resolution) in its intrinsic network of neurons and to hypothesize that the complex connectivity provides the expressive power. This idea is motivated by the fact that weights in neural networks can be interpreted as the number of synaptic connections (mathematically speaking, this corresponds to the *fan-in*) among neurons, which cannot be exponentially large.

Hypothesis. *The neurons in the human brain do not use "large" numbers to compute. The connectivity patterns among the neurons achieve the expressive power of the brain.*

Practically speaking, reducing the weights in a neural network has forthcoming advantages especially in the chip implementation: The multiplication and summation circuits can be simplified to use a smaller number of bits, which makes computations faster, more power efficient, and less area consuming. In one extreme, in the case of binary $\{0, 1\}$ or ternary weights $\{-1, 0, 1\}$, multipliers can be completely eliminated. In practice, weight quantization can be used to achieve such gains with a loss in the accuracy of the network output (Hubara et al., 2016; Jacob et al., 2018; Li, Zhang, and B. Liu, 2016). Surprisingly, given that the neural circuit is large in size and deep in depth, this loss in the accuracy might be tolerated. However, the trade-off is difficult to justify theoretically.

We are thus interested in the theoretical justification of the trade-off in the increase of size and depth while using as small weights as possible and keeping the function computation *exact*, that is, we want to compute the target function as it is without an error. In our analysis, we consider *Boolean functions*, which is basically a mapping from $\{0, 1\}^n$ to $\{0, 1\}$. In other words, it is a partitioning of n-bit binary vectors into two sets with labels 0 and 1 and one can consider this as a binary classification framework. Since all the data stored and processed in digital computers are represented in binary, Boolean functions will remain fundamental for a long time.

1.2 Boolean Functions and their Neural Network Constructions

The roots of the treatment of Boolean functions started from Leibniz as he invented the binary numeral system (most of the manuscripts are unpublished, see Strickland, 2020). George Boole was the pioneer of the *Boolean Algebra* and Claude E. Shannon revolutionized the area in his master's thesis using relay and switching circuits to implement systems of logic (Boole, 1847; Shannon, 1938). The *Circuit Complexity Theory* essentially started with Shannon's famous lower bound on the number of required circuit elements (2-input *gates* specifically) to compute almost all Boolean functions, which is $\Omega(2^n/n)$ and this is proven to be tight with a construction of size $O(2^n/n)$ (Lupanov, 1958; Shannon, 1949). For the lower bound, the proof idea is only existential but it applies to the vast majority (a fraction of $1 - o(1)$) of all Boolean functions. An important unresolved question in the circuit complexity theory is to find an explicit Boolean function such that the number of required gates is indeed $\Omega(2^n/n)$.

In the context of neural networks where different activation functions could have been picked, the "universal approximation" could be achieved for Boolean functions by explicit constructions. In general, we can use systems of linear equations as descriptive models of the two sets of binary vectors labeled as 0 or 1. For example, the solution set of the equation $\sum_{i=1}^{n} X_i = k$ is the n-bit binary vectors $X = (X_1, \dots, X_n)$ where each $X_i \in \{0, 1\}$ and k is the number of 1s in the vectors. In the light of the questions we ask about neural networks and approximation theory, here we try to answer three important questions: How expressive can a single linear equation be? How many equations do we need to describe a Boolean function? Could we simulate a single equation by a system of equations with smaller integer weights?

Let us begin with an example: 3-input PARITY function where we label binary vectors with odd number of 1s as 1. We can write it in the following form in a single equation with a *membership check* to a set of numbers:

$$\text{PARITY}_3(X) = \mathbb{1}\left\{(2^2 X_3 + 2^1 X_2 + 2^0 X_1) \in \{1, 2, 4, 7\}\right\} \tag{1.1}$$

where $\mathbb{1}\{.\}$ is the indicator function with outputs either 0 or 1. We typically use PARITY_n to emphasize that the function has n inputs while we might omit it if we talk about more generally. In this example, we express the binary vectors as integers by using binary expansions. Thus, it can be shown that if the weights are exponentially large in n, we can express all Boolean functions in this form where we list the values of the weighted summation where the output is 1. Therefore, given the power to modify the activation function, a single equation is sufficient to compute any Boolean function using weights of size $O(2^n)$. For the 3-input PARITY, an activation function is depicted in Figure 1.1a.

Now, suppose that we are only allowed to use a single equality check in an indicator function. Considering the 3-input PARITY function, we can simply obtain system of equations. Note that the use of matrices is only for demonstration purposes, we are not interested in the solutions to the system for the moment.

$$\begin{bmatrix} 2^0 & 2^1 & 2^2 \\ 2^0 & 2^1 & 2^2 \\ 2^0 & 2^1 & 2^2 \\ 2^0 & 2^1 & 2^2 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 7 \end{bmatrix} \tag{1.2}$$

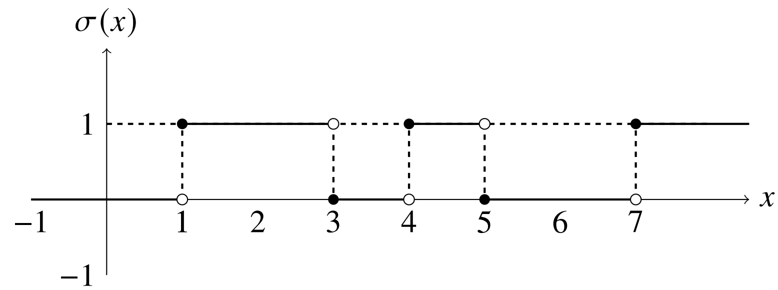
None of the above equations can be satisfied if X is labeled as 0. Conversely, if X satisfies one of the above equations, we can label it as 1. For an arbitrary Boolean function of n inputs, if we list every integer associated with vectors labeled as 1, the number of rows may become exponentially large in n by keeping the activation functions much "simpler". Nevertheless, by the same fashion, we can also compute PARITY_3 by the following system of equations using smaller weights.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \tag{1.3}$$

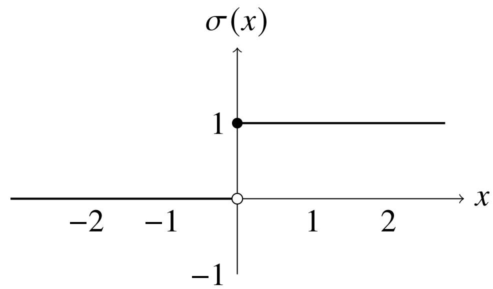
Not only there is a simplification on the number of equations, but also the weights are reduced to smaller sizes. This phenomenon motivates the following question with more emphasis: For which Boolean functions could we obtain such simplifications in the number of equations and weight sizes? For the PARITY, this simplification is possible because it is a *symmetric Boolean function*, i.e., the output depends only on the number of 1s of the input X. We use $|X|$ to denote the number of 1s in a binary vector X. Typically, to denote binary vectors, we use the capital letters like X and for other types of integer or real valued vectors, we use the small letters like x.

Definition 1. *A Boolean function is called* symmetric *if $f(X)$ has the same value for all permutations of the input X. Namely, a symmetric Boolean function $f(X)$ is a function of $|X|$.*

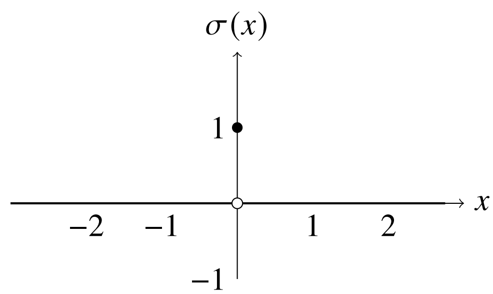
We are particularly interested in simplifications from large weights to small weights for a class of Boolean functions called *threshold functions*. In this thesis, n always refers to the quantities related to the number of inputs of the Boolean functions and we use the word "large" to refer to exponentially large quantities in n and the word "small" to refer to polynomially large quantities (including $O(n^0) = O(1)$) in n. It is a well-established fact that threshold functions require exponentially large weights in n, i.e., the weights should be as large as $2^{O(n \log n)}$, which seems to contradict our hypothesis (Alon and Vũ, 1997; Babai et al., 2010; Håstad, 1994; Saburo Muroga, 1971). However, by adding more neurons to the network, it might be possible to reduce the weights to polynomially large quantities in n, similar to what we did for the PARITY. To demonstrate credibility towards our hypothesis, we need to understand the properties of a single neuron, namely, a threshold function in this context.



(a) An activation function $\sigma(x)$ of the 3-input PARITY function given in Eq. (1.1).



(b) The activation function $\sigma(x)$ for *linear threshold functions*.



(c) The activation function $\sigma(x)$ for *exact threshold functions*.

Figure 1.1: Comparison of different activation functions. We do not consider arbitrary activation functions usually. Instead, we focus on thresholds and equality checks.

As noted before, the single perceptron formulation in Eq. (1.1) is very powerful if we allow arbitrary activation functions in the perceptrons that we use. In this case all the complexity (or the "ambiguity") is absorbed into the activation function itself similar to the case in (Guliyev and Ismailov, 2018). If the activation function is just a *threshold* or an *equality* check, then we call the functions *linear threshold* *functions* (see (1.4)) for the former and *exact threshold functions* (see (1.5)) for the latter. The activation functions are given in Figure 1.1b and 1.1c. We can write an n-input threshold function using the indicator function where w_i's are integer weights and b is a *bias* (or *threshold*) term. A device computing the corresponding threshold function is called a *gate* of that type.

$$f(X) = \mathbb{1}\Big\{\sum_{i=1}^{n} w_i X_i \geq b\Big\} \tag{1.4}$$

$$g(X) = \mathbb{1}\Big\{\sum_{i=1}^{n} w_i X_i = b\Big\} \tag{1.5}$$

To illustrate the concept, we define EQUALITY (denoted by EQ) function which computes whether an n-bit integer X is equal to another n-bit integer Y. The linear threshold counterpart of it is defined as the COMPARISON (denoted by COMP) function which checks if an n-bit integer X is greater than or equal to another n-bit integer Y (see Figure 1.2). In some works, it is also called GREATER-THAN function, denoted by GT (Hansen and V. V. Podolskii, 2010). In terms of complexity measures, these functions are referred extensively in this thesis. We present 6-input COMP and EQ in Figure 1.2.

Definition 2. *The* $2n$*-input* EQUALITY *(denoted by* EQ_{2n}*) function is an exact threshold function checking if two unsigned n-bit integers X and Y are equal.*

$$\mathrm{EQ}_{2n}(X,Y) = \mathbb{1}\Big\{X = Y\Big\} \tag{1.6}$$

$$= \mathbb{1}\Big\{\sum_{i=1}^{n} 2^{i-1}X_i = \sum_{i=1}^{n} 2^{i-1}Y_i\Big\} \tag{1.7}$$

where X_i *and* Y_i *are the binary expansions of* X *and* Y.

The $2n$*-input* COMPARISON *(denoted by* COMP_{2n}*) is the linear threshold function counterpart of the* EQ *where* $\mathrm{COMP}_{2n}(X,Y) = \mathbb{1}\{X \geq Y\}$.

We now ask the previous questions for the EQ and COMP functions: Do we require exponentially large weights in n to compute them using a single threshold function? For both of these functions, it is known that the weights should be exponentially large (Babai et al., 2010; Siu and Bruck, 1991). This problem is shown to be equivalent to the unresolved *Erdős' Distinct Sum Subset* problem questioning the optimality of the

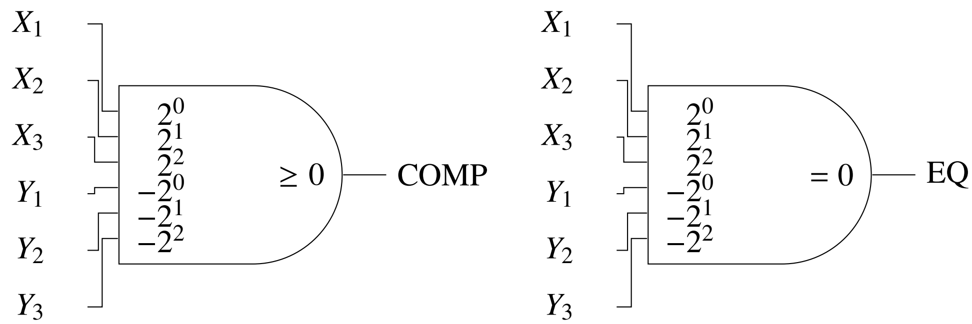


Figure 1.2: The 6-input COMP and EQ functions for integers X and Y computed by linear threshold and exact threshold gates. More explicitly, we can write $\mathrm{COMP}(X,Y) = \mathbb{1}\{4X_3 + 2X_2 + X_1 \geq 4Y_3 + 2Y_2 + Y_1\}$ and $\mathrm{EQ}(X,Y) = \mathbb{1}\{4X_3 + 2X_2 + X_1 = 4Y_3 + 2Y_2 + Y_1\}$.

weights for the EQ and COMP (Babai et al., 2010; Erdős, 1983). The next question is whether we can compute these functions by *circuits* of threshold functions using *smaller* weights. The answer is yes and the main result in this context is that although most linear threshold functions cannot be computed by subexponentially large weights in n in depth-1, all of them can be computed by depth-2 threshold circuits with polynomially large weights and polynomially large size in n. The exact theoretical trade-offs between the weight size and the circuit size is one of the problems on which we focus in this thesis in Chapter 2 building on the previous works in the literature.

There are "trivial" ways to look at this trade-off between the weight size and the circuit size. If we allow exponentially large fan-in in our gates and exponentially large circuit size, then we can replicate the input variables and compute any threshold function using constant size weights in n (e.g. see Figure 1.3). Usually, exponential blow-ups are considered "infeasible" in terms of practical purposes, also contradicting our hypothesis, and we are interested in polynomial blow-ups in n.

However, the blow-ups are not necessarily "bad" and it is a common approach to allow at most polynomially large fan-in in n for the practical implementations of threshold gates and this replication technique can be used if the weights are polynomially large in n. For instance, an n-input MAJORITY (denoted by $\mathrm{MAJ}_{n,b}$ where b is the bias term) function is a Boolean function that counts the number of 1s and decides 1 if the value exceeds the threshold b. In order to make MAJ functions more expressive, multiple connections of the same input are allowed up to polynomially large fan-in in n to simulate any threshold function with polynomially large weights in n (Goldmann and Karpinski, 1993; Hansen and V. V. Podolskii,

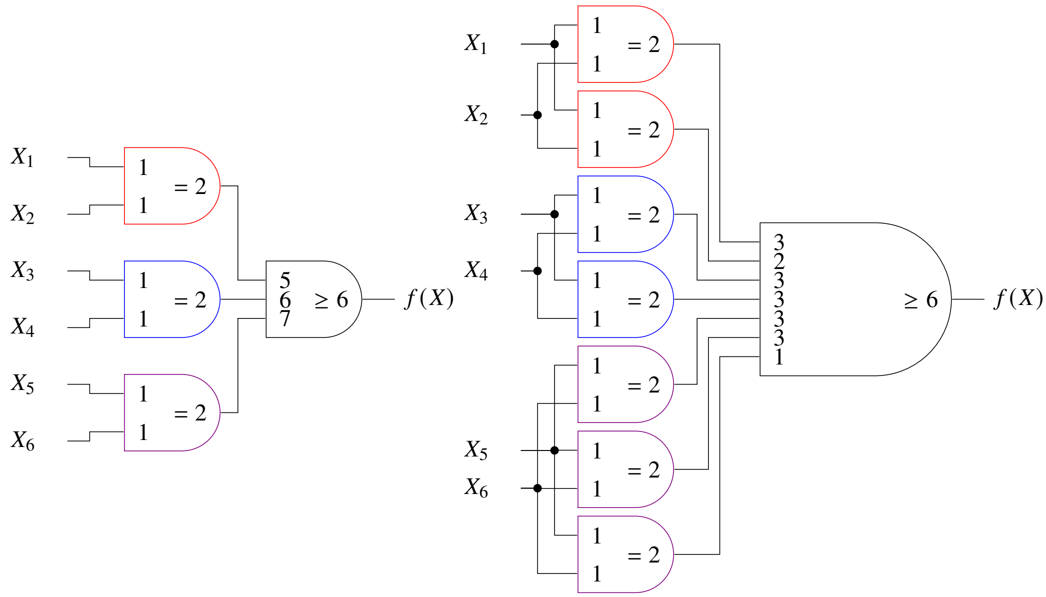


Figure 1.3: An example of a weight transformation for a single gate (in black) to construct constant weight circuits. Here, we pick $f(X)$ to be some arbitrary function which may not have the optimal weights and size in the given circuits and therefore, one should consider this as a toy example. Different gates are colored in red, blue, and violet and depending on the weight size, each gate is replicated a number of times. The first layer gates are all 2-input AND gates. The construction on the right has weights at most as large as 3 except the bias term of the top gate and both circuits compute the same function using different amount of resources. Moreover, for a depth-d threshold circuit, one can verify that this technique introduces a blow-up which may be exponential in d.

2010). We use the notation MAJ to denote all possible MAJ functions.

If the weights of the threshold gates are restricted to polynomially large numbers in n (for MAJ, it is only 1 except the bias), then the replication idea could not be used to compute exponentially large weights in n threshold function using polynomially large fan-in (see Figure 1.4 for the application of the replication technique). However, if the weights of the threshold gates are already exponentially large in n, then an arbitrary weight can be computed by a polynomially large fan-in in n. This gives us a justification to define a gate with exponentially large weights, specifically, powers-of-two as they can express large weights and there is a guaranteed binary expansion for an arbitrary integer value of a weight. We call the function associated to this gate the DOMINATION function (denoted by $\text{DOM}_{n,b}$), which compares an n-bit integer to a fixed integer b, computing 1 if the former is greater than or equal to the latter. It is called DOMINATION because it takes the value 1 depending on the location of the "most significant" 1s in the input vector. Again, we use the notation

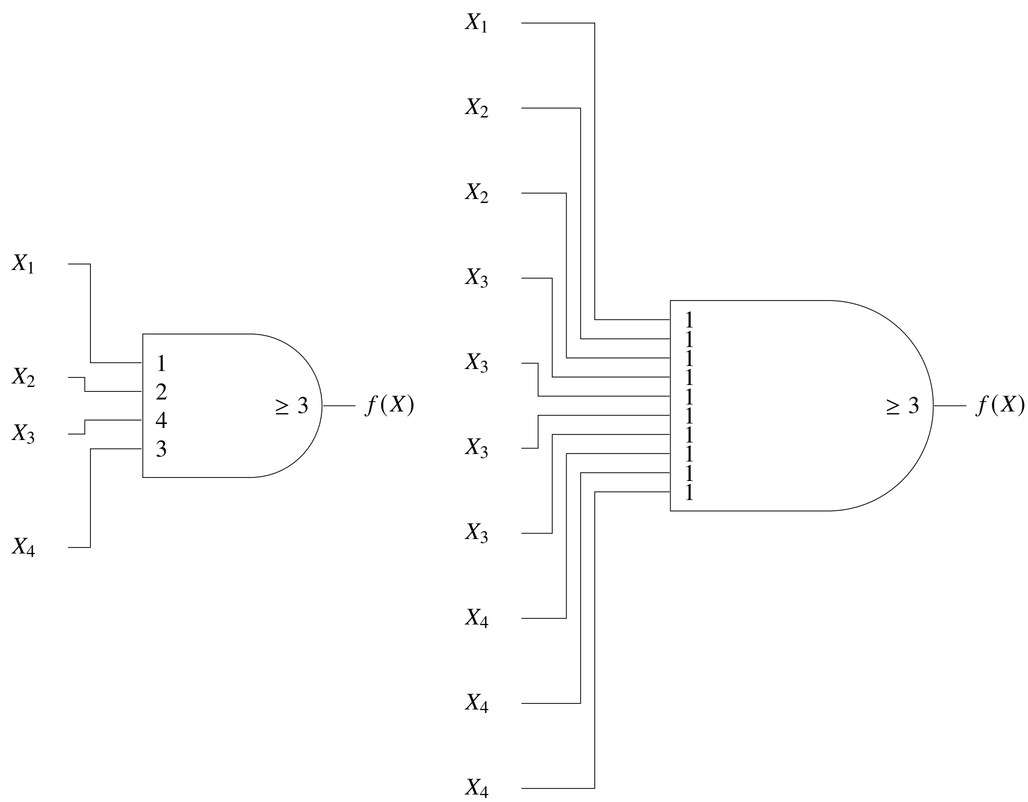


Figure 1.4: The input replication technique to use a MAJ gate to compute a linear threshold function with 4-inputs. One can see that the gate on the right side is equivalent to $\text{MAJ}_{10,3}$. If desired, the bias term can also be made small by adding the constant $x_0 = 1$ to the input vector. Both schemes compute the same threshold function $f(X)$. In general, if the weights are exponentially large in n, the fan-in also becomes exponentially large in n as well. To keep the fan-in polynomially large in n, we define DOM functions.

DOM to denote all possible DOM functions.

We also allow negative powers-of-two for the weights of a DOM function. More explicitly, we give a definition below in Eq. (1.8), which uses an abuse of notation by $\pm$ to emphasize that any sign combinations for the weights is allowed. We typically choose all weights positive, i.e., $+2^0, +2^1, \ldots, +2^{n-1}$. In this representation, if necessary, the negative weights can be handled by changing an input x_i to $-x_i$.

$$\text{DOM}_{n,b}(X) = \mathbb{1}\left\{\sum_{i=1}^{n} \pm 2^{n-i} X_i \geq b\right\} \tag{1.8}$$

A comparison between MAJ and DOM functions is given in Figure 1.5. Both MAJ and DOM are classes of threshold functions and we allow DOM to have multiple connections of the same input in order to make the function more expressive akin to MAJ. It is an interesting fact that this idea leads us to many new directions in the treatment of threshold functions as we shall see in Chapter 2. We give a detailed transformation of threshold functions using a DOM gate in Figure 2.2.

Although DOM functions are not referred as such in literature, implicitly these functions appear in different contexts (Beigel, 1994; Buhrman, Vereshchagin, and Wolf, 2007; Siu and Bruck, 1991; Turán and Vatan, 1997). An important example of DOM functions is the ODD-MAX-BIT function, which is also shown to have necessarily exponentially large weights in n in a single threshold function form.

Definition 3. *The n-input* ODD-MAX-BIT *(denoted by* OMB_n*) function is a linear threshold function checking if the index of the leftmost* 1 *of a binary vector* $X = (X_1, \ldots, X_n)$ *has an odd index or not.*

$$\mathrm{OMB}_n(X) = \mathbb{1}\left\{\sum_{i=1}^{n}(-1)^{i-1}2^{n-i}X_i > 0\right\} \tag{1.9}$$

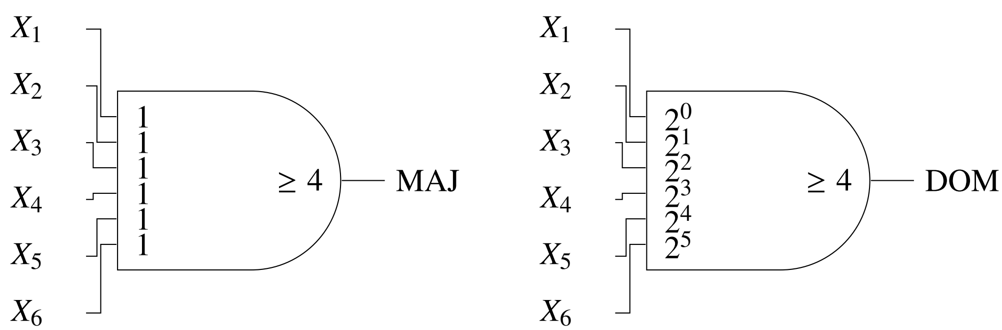


Figure 1.5: Examples for 6-input MAJ and DOM with all positive weights. The bias term can be chosen arbitrarily to shift the threshold. For example, if $X = (X_1, X_2, X_3, X_4, X_5, X_6) = (0, 1, 0, 0, 1, 0)$ with $b = 4$, then $\mathrm{MAJ}_{6,4}(X) = 0$ and $\mathrm{DOM}_{6,4}(X) = 1$.

Here is a high level summary of our results in this context. We present our results for Depth-2 Circuit constructions for specific threshold functions in Table 1.1.

- We introduce the *Domination Form of Threshold Functions* to compute an arbitrary threshold function using a linear transformation layer and a DOM gate (see Section 2.1). We remark on how this representation can be used to obtain smaller size complexity and to achieve asymptotic efficiency. For this type of constructions, we give upper and lower bounds with explicit constructions based on the Chinese Remainder Theorem (CRT) and the Prime Number Theorem (PNT) (see Section 2.2 and 2.5).

- We reproduce a previously known lower bound in (Roychowdhury, Orlitsky, and Siu, 1994) by a novel technique using *Siegel's Lemma* from linear algebra and number theory. A converse lemma is used to solve an open problem and prove the existence of an asymptotically efficient size constant weight constructions for the EQ (see Section 2.3).

- In addition to this existence result, we explicitly construct such an EQ matrix with entries $\{-1, 0, 1\}$ by extending the Sylvester-type Hadamard matrices, which achieves asymptotical efficiency in size.

- We extend our techniques to obtain an MDS-like structure and show the existence of a small weight circuits with $O(n)$ weights compared to $O(n^2)$ as shown in (Amano and Maruoka, 2005). There is no trade-off in the number of rows asymptotically. To prove this, we refer to one of the *anti-concentration* inequalities, namely, Berry-Esseen Theorem. In Circuit Complexity Theory, the use of anti-concentration inequalities is relatively new (see Diamond and Yehudayoff, 2022 for another recent result). This result is used to find more efficient constructions for COMP, CARRY, and OMB in weight size without no asymptotic trade-off in circuit size.

- By using *Domination Form of Threshold Functions*, we extend our constructive results to a broader class of threshold functions. Specifically, if the sparsity parameter $\mathcal{S}$ is 1 (see Section 2.1 for the definition), then we can apply EQ and RMDS matrices to find constructions with smaller weight sizes with the same circuit size complexity. This class of functions includes the EQ, COMP, CARRY (i.e. $\mathbb{1}\{X + Y \geq 2^n\}$ for n-bit unsigned integers X and Y), and OMB.

1.3 Nearest Neighbor Representations of Boolean Functions

The study of Nearest Neighbors (NN) started in 1950s and they remained one of the most fundamental models in machine learning (Fix and Hodges, 1951). Intuitively, each real-life concept is thought to be a real-valued vector in a mathematical sense and the "closer" the concepts are, the more "similar" they should be. The k-Nearest Neighbors algorithm finds the closest k concepts to any input vector $x \in \mathbb{R}^n$ and tries to "describe" it by a weighted average for regression or a majority voting for

Function	**This Work**		**Previous Works**	
	Weight Size	Constructive	Weight Size	Constructive
EQ	$O(1)$	Yes	$O(n)^*$	Yes
COMP	$O(n)$	No	$O(n^2)^\dagger$	Yes
CARRY	$O(n)$	No	$O(n^2)^\dagger$	Yes
OMB	$O(n)$	No	$O(n^2)^\dagger$	Yes

*(Roychowdhury, Orlitsky, and Siu, 1994)
†(Amano and Maruoka, 2005)

Table 1.1: Results for the Depth-2 Circuit Constructions. The results have the smallest known circuit size asymptotically.

classification. How many concepts do we need to describe a vector as "clear" as possible? In an information theoretic sense, this question highlights the number of concepts and description error trade-off, called the infamous *curse of dimensionality*, which, loosely speaking, implies that one needs a "large" quantity of concepts to describe vectors "clearly" when the dimension grows (Cover and Hart, 1967; Weber, Schek, and Blott, 1998; Beyer et al., 1999). More recently, NN models gained more attention since Natural Language Processing (NLP) pipelines may contain large vector databases to store high dimensional real-valued *embeddings* of words, sentences, or even images (Johnson, Douze, and Jégou, 2019; P. Lewis et al., 2020). Thus, the theoretical limits of the database sizes, the approximations of the NN search algorithms, and the design of "meaningful" embeddings became important research directions (Indyk and Motwani, 1998; Kushilevitz, Ostrovsky, and Rabani, 1998; Malkov and Yashunin, 2018; Reimers and Gurevych, 2019). An example use of vector databases in NLP involving NN search algorithms is given in Figure 1.6.

In this thesis, we consider NN models from a relatively novel perspective, which could play a role in our understanding on how the brain works. Inspired by the architecture of the brain, *neural networks* became the backbone of many machine learning models, and similarly, *Nearest Neighbor (NN) Representations* are introduced as emerging models of computation (Minsky and Papert, 2017; P. Hajnal, Z. Liu, and Turán, 2022). NN representations mimic the integrated structure of memory and computation in the brain where the concepts in the memory are embedded in *anchors*. They are specifically defined for Boolean functions where the input vectors are binary vectors, i.e., $X \in \{0,1\}^n$, and the labels can be either **0** (**red**) or 1 (blue). We always use a single NN, i.e., we focus on the case $k = 1$, to find the correct label. Thus, the closest anchor exactly gives the output label of the input

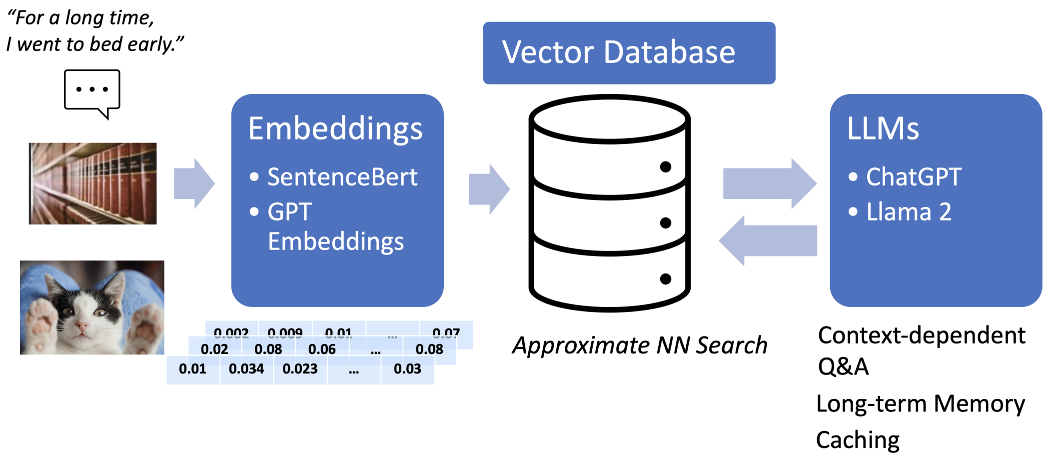


Figure 1.6: An illustration of an *Retrieval Augmented Generation (RAG)* pipeline in NLP (P. Lewis et al., 2020). Each "document", which can be a single sentence, a book, or an image of a cat, is converted into a high-dimensional embedding by *Embedding Models* and is stored in a vector database. To do knowledge-intensive tasks, similar "documents" can be found by approximate NN search algorithms.

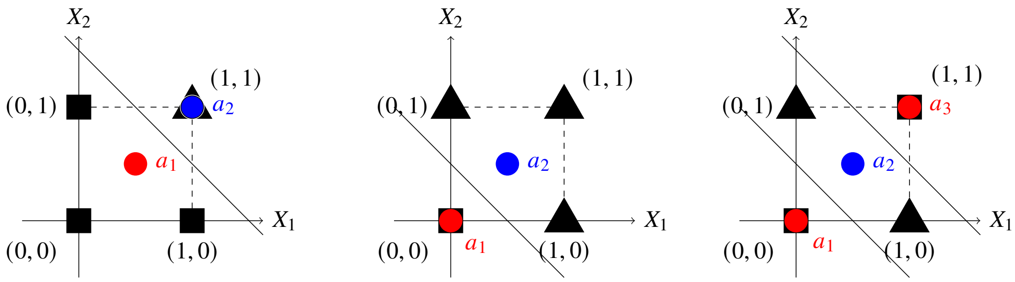
vector X.

One can easily find NN representations for the 2-input Boolean functions AND, OR, and XOR as given in Figure 1.7. The anchors closest to the input vectors X such that $f(X) = 1$ (or 0) are called *positive* in blue (or *negative* in **red**). These example functions given in Figure 1.7 are all symmetric Boolean functions.

The information capacity of a Boolean function in this model is associated with two quantities: *(i)* the number of anchors (called *Nearest Neighbor (NN) Complexity* (P. Hajnal, Z. Liu, and Turán, 2022)) and *(ii)* the maximal number of bits representing entries of anchors (called *Resolution*). We first define mathematically what we call an NN Representation. Let $d(a, b)$ denote the Euclidean distance between the vectors $a, b \in \mathbb{R}^n$. Throughout the thesis, we only consider Euclidean distance. The Nearest Neighbor Representation and Complexity are defined in the following manner.

Definition 4. *The* Nearest Neighbor (NN) Representation *of a Boolean function f is a set of anchors consisting of the disjoint subsets (P, N) of $\mathbb{R}^n$ such that for every $X \in \{0,1\}^n$ with $f(X) = 1$, there exists $p \in P$ such that for every anchor $n \in N$, $d(X, p) < d(X, n)$, and vice versa. The* size *of the NN representation is $|P \cup N|$.*

Namely, the size of the NN representation is the number of anchors. Naturally, we are interested in representations of minimal size to quantify the information to



X_1	X_2	AND(X)	OR(X)	XOR(X)
0	0	**0**	**0**	**0**
0	1	**0**	1	1
1	0	**0**	1	1
1	1	1	1	**0**
Anchors		$\begin{pmatrix} \mathbf{0.5} & \mathbf{0.5} \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ 0.5 & 0.5 \end{pmatrix}$	$\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ 0.5 & 0.5 \\ \mathbf{1} & \mathbf{1} \end{pmatrix}$

Figure 1.7: NN representations for 2-input Boolean functions AND$(X) = x_1 \wedge x_2$ (left), OR$(X) = x_1 \vee x_2$ (middle), and XOR$(X) = x_1 \oplus x_2$ (right). Triangles denote $f(X) = 1$ and squares denote $f(X) = 0$. It can be seen that **red** anchors are closest to squares and blue anchors are closest to triangles. Separating lines between anchors pairs are drawn.

represent a Boolean function.

Definition 5. *The* Nearest Neighbor Complexity *of a Boolean function f is the minimum size over all NN representations of f, denoted by $NN(f)$.*

The second quantity that we use to measure the information capacity of an NN representation is the *resolution*. Without loss of generality, one can assume that the entries can be rational numbers when the inputs are discrete. We define *anchor matrix* $A \in \mathbb{Q}^{m \times n}$ of an NN representation where each row is an n-dimensional anchor and the size of the representation is m. The *resolution of an NN representation* is the maximum number of bits required to represent an entry of the anchor matrix.

Definition 6. *The* resolution (RES) of a rational number a/b *is* $RES(a/b) = \lceil \max\{\log_2 |a+1|, \log_2 |b+1|\} \rceil$ *where* $a, b \in \mathbb{Z}$, $b \neq 0$, *and they are coprime.*

For a matrix $A \in \mathbb{Q}^{m \times n}$, $RES(A) = \max_{i,j} RES(a_{ij})$. *The* resolution of an NN representation *is* $RES(A)$ *where* A *is the corresponding anchor matrix.*

For instance, in Figure 1.7, the 2-input AND function has two anchors $a_1 = (0.5, 0.5)$

and $a_2 = (1, 1)$. By using the definition, we see that the resolution of this representation is $\lceil \log_2 3 \rceil = 2$.

In the context of NN representations, instead of the word "large" (or "small") to talk about the magnitude of the anchor entries, we prefer to use the word "high" (or "low") for the resolution of the anchor entries.

We emphasize that although Boolean functions are evaluated over binary vectors, the anchor points can have rational or even real valued entries. Similar to the curse of dimensionality, we illustrate another intuitive trade-off between the NN complexity and resolution where "lower" resolution NN representations imply a "larger" NN complexity. An extreme case would be to consider single-bit resolution anchors, that is, we consider all the 2^n nodes of the Boolean hypercube as anchors and assign them to P or N based on the values of f. Any input vector corresponds to an anchor with 0 distance in this case. This implies that $NN(f) \leq 2^n$ for any n-input Boolean function. Specifically, we define the *Boolean Nearest Neighbor Complexity (BNN)* for the case where we restrict all anchors to be binary vectors. It is easy to see that $NN(f) \leq BNN(f) \leq 2^n$ where $NN(f)$ has no explicit resolution constraint (P. Hajnal, Z. Liu, and Turán, 2022). This upper bound on the NN complexity is essentially loose for all Boolean functions (see Eq. (1.10)).

Theorem 1 (P. Hajnal, Z. Liu, and Turán, 2022)**.** *For every n-input Boolean function f, it holds that*

$$NN(f) \leq (4 + o(1))\frac{2^n}{n} \tag{1.10}$$

Moreover, for almost all n-input Boolean functions, we have

$$NN(f) > \frac{2^{n/2}}{n} \tag{1.11}$$

X_1	X_2	$\lvert X \rvert$	$\text{XOR}(X)$	BNN Representation
0	0	0	**0**	$\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ 0 & 1 \\ 1 & 0 \\ \mathbf{1} & \mathbf{1} \end{pmatrix}$
0	1	1	1	
1	0	1	1	
1	1	2	**0**	

Figure 1.8: The 2-input XOR functions with its optimal BNN representation. It is known that $BNN(\text{XOR}) = 4$ whereas $NN(\text{XOR}) = 3$.

In general, the smaller the resolution is, the larger the size of the representation should be. For example, the discrepancy between NN and BNN is illustrated for the XOR function in Fig. 1.8. Note that PARITY is a generalization of XOR to n variables. By proving corresponding upper and lower bounds on NN complexity, the following result is obtained for the PARITY.

Theorem 2 (P. Hajnal, Z. Liu, and Turán, 2022)**.** *For the n-input* PARITY *function, it holds that* $NN(\text{PARITY}_n) = n + 1$ *with resolution* $\lceil \log_2 (n+1) \rceil$ *and* $BNN(\text{PARITY}_n) = 2^n$.

Before going into more details, we do a brief overview of previous works on the NN representations. In this thesis, the main inspiration is from the work of Hajnal, Liu and Turán in 2022. Their paper provided several NN complexity results, including the construction for PARITY (see Theorem 2) and bounds on the NN complexity of arbitrary Boolean functions (see Theorem 1). Optimizing the representation of a set by using NNs was first discussed in the context of minimizing the size of a training set, namely, finding a minimal training set that NN represents the original training set (Salzberg et al., 1995; Wilfong, 1991). The idea to represent Boolean functions via the NN paradigm was first studied in (Globig and Lange, 1996) where BNN was considered with distance measures that are chosen to optimize the complexity of the NN representation. This work was extended to inclusion-based similarity (Satoh, 1998) and it was proved that it provides polynomial size representations for DNF and CNF formulas.

How can we relate NN representations to the brain? For a large network of neurons, the first step would be to understand the NN representations of threshold functions. For the NN representations of linear threshold functions, we see that $NN(f) = 2$ for linear threshold functions except the constant Boolean functions $f(X) = 0$ and $f(X) = 1$ (P. Hajnal, Z. Liu, and Turán, 2022). The geometrical idea is given in Figure 1.9. Interestingly, this demonstrates that, in a traditional neural network, the building blocks are based upon functions with $NN(f) = 2$. We also see that for exact threshold functions which are not also linear, $NN(f) = 3$. We give more precise results on these results in Section 3.1.

Since we already noted that threshold functions might have exponentially large weights in n, the resolution of the NN representation given in Figure 1.9 could be linear in n (see Theorem 17). Previously, we mentioned that considering depth-2 threshold circuits, it is possible to reduce the weights to polynomially large quantities in n (i.e. logarithmic resolution in n). Therefore, a single threshold gate with large weights can be computed by a depth-2 threshold circuit with small weights. We now ask an analogous question for threshold functions: Is it possible to reduce the resolution from linear quantities in n to logarithmic quantities in n by adding at most a polynomially large number of anchors in n to the NN representation of a threshold function?

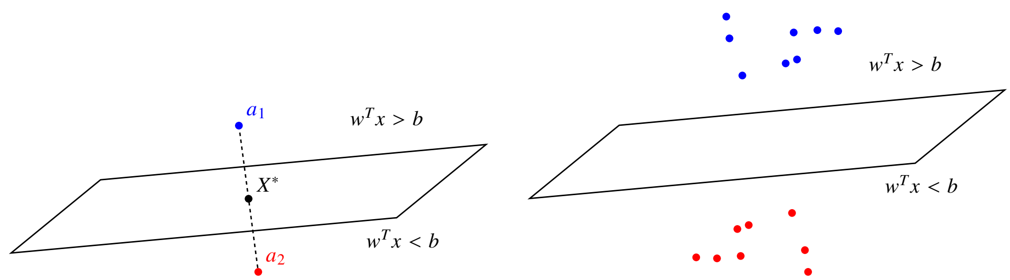


Figure 1.9: The NN Representation of a linear threshold function $\mathbb{1}\{w^T X \geq b\}$ and its 2-anchor NN Representation (left) and a hypothetical NN representation for the same function with more anchors and less resolution (right).

When arbitrary threshold functions are considered, the answer to this question is still unclear. One way to tackle this problem is to understand the small weight depth-2 threshold circuit constructions of threshold functions with large weights better. However, our knowledge on the NN representations of neural networks even with depth-2 circuits is very limited. Therefore, we need to obtain and understand NN representations of neural networks starting with depth-2.

More generally, by composing threshold function and symmetric Boolean functions together, one can obtain different families of Boolean functions and one of our main goals is to find explicit NN representations for them. To denote the class of linear (or exact) threshold functions, we use LT (or ELT). For symmetric Boolean functions, we use SYM. Also, $\circ$ denotes function composition. For instance, a neural network with threshold functions is in fact a composition of LT gates and to understand the NN representations of neural circuits, the treatment of small depth circuits is essential.

We especially treat the two classes SYM $\circ$ LT and SYM $\circ$ ELT because these classes appear when one considers small weight threshold circuit constructions of threshold functions with large weights (Amano and Maruoka, 2005; Hansen and V. V. Podolskii, 2010; Hofmeister, 1996). Although we cannot find an NN representation for both of these classes in the arbitrary case, we are able to construct NN representations under the assumption of some *regularity conditions* on the weights of the first layer (see Section 3.5 for the definition of the conditions).

Another important example would be *linear decision lists* (denoted by LDL), which can be computed by depth-2 linear threshold circuits as in Eq. (1.12) (i.e. LDL $\subseteq$ LT $\circ$ LT) (Turán and Vatan, 1997). A linear decision list of length m is a sequential list of linear threshold functions $f_1(X), \dots, f_m(X)$ where the output is $z_k \in \{0,1\}$ for $f_i(X) = 0$ for $i < k$ and $f_k(X) = 1$ and it is $z_{m+1} \in \{0,1\}$ if all $f_i(X) = 0$. Here, m is the number of threshold functions in the list. Other works sometimes state that the length is $m+1$ if there are m threshold functions.

In general, *Decision Lists* was first proposed similar to NN representations as a different representation for Boolean functions (Rivest, 1987). Their relation to circuit complexity is treated in many works (Chattopadhyay, Mahajan, et al., 2019; Dahiya et al., 2024; Turán and Vatan, 1997; Uchizawa and Takimoto, 2015). The learnability of decision lists is treated in many Machine Learning applications and Complexity Theory (Angelino et al., 2018; Hancock et al., 1996; Klivans, Servedio, and Ron, 2006).

One can similarly define the class of EDL where exact threshold functions are used and EDL $\subseteq$ LT $\circ$ ELT. We give an example of a 5-input LDL of depth 3 in Figure 1.10.

$$l(X) = \mathbb{1}\left\{\sum_{i=1}^{m} (-1)^{z_i - 1} 2^{m-i} f_i(X) \geq 1 - z_{m+1}\right\} \tag{1.12}$$

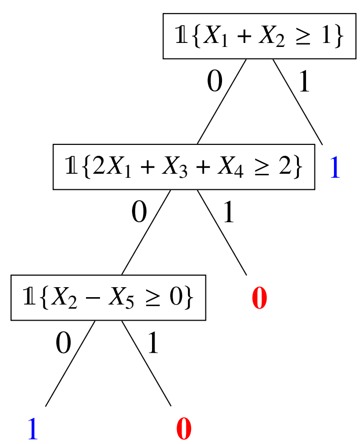


Figure 1.10: A Linear Decision List of Depth 3 with 5 binary inputs.

In Eq. (1.12), the implied top gate (see Figure 1.11) is actually called a DOMINATION gate because the leading one in the vector $(f_1(X), \dots, f_m(X))$ *dominates* the importance of the others. In this sense, one can see that LDL $\subseteq$ DOM $\circ$ LT (or

EDL $\subseteq$ DOM $\circ$ ELT). In Figure 1.11, a depth-2 circuit construction of the linear decision list depicted in Figure 1.10 is given.

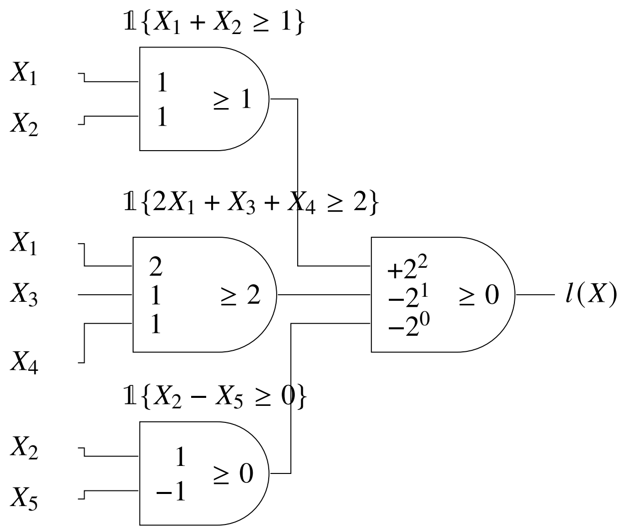


Figure 1.11: The depth-2 threshold circuit construction of an LDL $l(X)$ with a DOM gate on top. This shows that $l(X) \in$ LT $\circ$ LT (or even $l(X) \in$ DOM $\circ$ LT). The signs of the powers of two depends on the labeling of the outputs z_is. If the first layer consists of exact threshold gates, then we have an EDL.

The classes of LDL and EDL are in fact related. Note that for the following inclusion claims, we consider decision lists of polynomially large length in n. By doing a transformation from LT gates to ELT gates (idea is given first in Hofmeister, 1996 showing that LT $\subsetneq$ OR $\circ$ ELT), it can be shown that LDL $\subsetneq$ EDL (Dahiya et al., 2024). The strictness is due to $\text{OR}_n \circ \text{EQ}_{2n}$ (Chattopadhyay, Mahajan, et al., 2019). Moreover, the class EDL is equivalent to the class of decision lists which have $\text{AND}_r \circ \text{LT}_n$ for r to be a constant in n (Dahiya et al., 2024).

Although linear and exact decision lists are not used in the analysis of low resolution NN representations, they are important to understand the capabilities of NN representations. Let $LDL(f)$ be the smallest possible length of all linear decision lists computing the Boolean function f. In this thesis, we establish the upper bound $NN(f) \leq LDL(f) + 1$ if $LDL(f) \leq n$. The immediate conjecture is that this bound holds for any length value (see Theorem 33).

A similar lower bound is already known by generalizing linear decision lists to *linear decision trees*. In a linear decision tree (denoted by LDT), each node might entail another linear threshold function. We do not go into the treatment of LDTs but for the sake of completeness, we give the lower bound based on the complexity measure

$LDT(f)$, which is the smallest possible depth of all linear decision trees computing a Boolean function f (see Gröger and Turán, 1991 and Björner, Lovász, and Yao, 1992). The lower bound not only applies to $NN(f)$ but also $kNN(f)$, called k-NN complexity. For $kNN(f)$, one considers the k nearest anchors to decide on the label of $f(X)$ by majority voting and $NN(f)$ is a special case of $kNN(f)$ where $k = 1$. We do not treat k-NN representations in this thesis.

Theorem 3 (P. Hajnal, Z. Liu, and Turán, 2022)**.** *For any n-input Boolean function f, $LDT(f) \leq (3 + o(1))kNN(f)$.*

Here is the scope of our analysis about NN representations.

- Linear Threshold Functions and Exact Threshold Functions (LT and ELT).
- Symmetric Boolean Functions (SYM).
- Depth-2 Threshold Circuits with Symmetric Boolean Function on Top (SYM $\circ$ LT or SYM $\circ$ ELT).
- Linear Decision Lists (LDL) and Exact Decision Lists (EDL) (DOM $\circ$ LT and DOM $\circ$ ELT).
- Low Resolution NN Representations for the Threshold Functions EQ, COMP, and OMB.

We present a summary of our results on NN representations for specific Boolean functions in Table 1.2. In general, NN representations of threshold functions require $O(n \log n)$ resolution. Thus, in the light of our hypothesis, we conjecture that there exists polynomially large size NN representations of threshold function with logarithmic resolution in n. We obtain several results to resolve this conjecture partially and the results are related to our NN representation constructions for depth-2 circuits.

- Explicit constructions for n-input threshold functions are presented with the resolution bound $O(RES(w))$ where $w \in \mathbb{Z}^n$ is the weight vector. The size of the bias term does not matter.
- A new construction is presented for an arbitrary n-input symmetric Boolean function $f(X)$ with $I(f)$ many anchors and $O(\log n)$ resolution. $I(f)$ denotes

the number of intervals for a symmetric Boolean function f (see Section 3.2 for the definition).

- The new construction has optimal NN complexity for some symmetric Boolean functions.
- There is always an optimal NN representation both in size 2 and resolution $O(1)$ for any n-input symmetric linear threshold function.
- The NN representations of some symmetric Boolean functions require the resolution to be $\Omega(\log n)$ so that the new constructions are optimal in resolution.
- The membership function to a convex polytope, $\mathbb{1}\{AX \leq b\}$ where $A \in \mathbb{Z}^{m\times n}$ and $b \in \mathbb{Z}^m$, is equivalent to Boolean functions $f(X)$ such that $NN(f) = m+1$ where there is a single positive anchor. This corresponds to the depth-2 circuits of type $\text{AND}_m \circ \text{LT}_n$.
- For circuits of $\text{SYM}_m \circ \text{ELT}_n$ with m many n-input exact threshold gates under regularity conditions, there is a NN representation which has $\sum_{t\in\mathcal{T}} \binom{m}{m-t} 2^{m-t}$ many anchors where $\mathcal{T}$ contains left interval boundaries for the top symmetric gate.
- For circuits of $\text{SYM}_m \circ \text{LT}_n$ with m many n-input linear threshold gates under regularity conditions, there is a NN representation which has $\sum_{t\in\mathcal{T}} \binom{m}{m-t}$ many anchors where $\mathcal{T}$ contains left and right interval boundaries for the top symmetric gate.
- For a Boolean circuit with m many n-input perceptrons whose first layer has a weight matrix $W \in \mathbb{Z}^{m\times n}$ such that $|W_{ij}| \leq B$ for some $B \in \mathbb{Z}$ with mutually orthogonal rows, there is a NN representation with $(2nB+1)^m$ many anchors and resolution $O(\log nB)$.
- For an LDL of length m, we provide a NN representation with $m+1$ anchors if $m \leq n$. For an EDL of length m under regularity conditions, an NN representation with $(m+1)2^m$ many anchors is provided. Moreover, since there is an LDL of length $I(f)$ computing an n-input symmetric Boolean function f, an equivalent construction for symmetric Boolean functions is presented in this framework independently.

- For the threshold functions EQ, COMP, and OMB, we construct NN representations with polynomially large number of anchors and logarithmic resolution. These functions have actually $\mathcal{S} = 1$ sparsity parameter in domination form (see Section 2.1 for the definition). It is still an open question if arbitrary threshold functions have NN representations with similar complexity.

Function	**NN Representation**		
	Size		Resolution
	Upper Bound	Lower Bound	
PARITY_n	2^{n*}	2^{n*}	1
$\text{AND}_m \circ \text{EQ}_{2n}$	$2m+1$		$O(n)$
$\text{OR}_m \circ \text{EQ}_{2n}$	$(m+2)2^{m-1}$		$O(n)$
$\text{PARITY}_m \circ \text{EQ}_{2n}$	3^m		$O(n)$
$\text{PARITY}_m \circ \text{COMP}_{2n}$	2^m		$O(n)$
IP2_{2n}	2^n	$2^{n/2*,\dagger}$	$O(1)$
$\text{OMB}_m \circ \text{EQ}_{2n}$	$(m+1)2^m$		$O(n)$
PARITY_n	$(m+1)^{n/m}$	$n+1^{*,\dagger}$	$\lceil \log{(m+1)} \rceil$
EQ_{2n}	$2n+1$	$\Omega(\sqrt{n})$	$O(1)$
EQ_{2n}	$O(n/\log n)$	$\Omega(\sqrt{n/\log n})$	$O(\log n)$
COMP_{2n}	$2n$	$\Omega(\sqrt{n/\log n})$	$O(\log n)$
OMB_n	$n+1$		$O(\log n)$

*(P. Hajnal, Z. Liu, and Turán, 2022)
†The lower bound is for arbitrary resolution

Table 1.2: NN Representations For Known Boolean Functions.

In this thesis, we have several contributions in the theory of neural computation. Even though the details of the computation in the human brain remain mysterious, mathematical models of it are provably powerful in theory and in practice. Several theoretical upper and lower bounds on the limits of computation are established. Even in an associative computation model, such as NN representations, the resolution could change the feasibility of the representations for high dimensions and we hope that our treatment would aid in exploring new directions in the area of neural computation.

This thesis has two main chapters: 1) On the Algebraic Constructions of Neural Networks with Small Weights and 2) On the Complexity of the Nearest Neighbor Representations. In the first chapter, we use the DOM gates to represent known threshold functions in different forms, which eventually simplifies the complexity of circuit constructions. We improve on the best results in existential results or explicit constructions. In the second chapter, we give an extensive treatment of the NN representations for several classes of Boolean functions as summarized before. Building on the results in Chapter 2, we resolve the conjecture on the NN representations of threshold functions.

Chapter 2

ON THE ALGEBRAIC CONSTRUCTIONS OF NEURAL NETWORKS WITH SMALL WEIGHTS

Our analysis begins with the treatment of the realization threshold functions in "small" weights using depth-2 circuits. Although the results are independent, some of them will be relevant for the NN representations as well. As it was noted in the introduction, it is possible to construct "small" weight *threshold circuits* to compute any threshold function (Amano and Maruoka, 2005; Goldmann and Karpinski, 1993; Hofmeister, 1996; Siu and Bruck, 1991). This transformation from a circuit of depth d with exponentially large weights in n to another circuit with polynomially large weights in n is typically within a constant factor of depth (e.g. $d+1$ if the activation functions are step functions or $3d+3$ if the activation functions are *Rectified Linear Unit* $\mathrm{ReLU}(z) = \max(0, z)$) (Goldmann and Karpinski, 1993; Vardi and Shamir, 2020). For instance, such a transformation would simply follow if we can replace any "large" weight threshold function with "small" weight depth-2 circuits so that the new depth becomes $2d$.

We remark that the input replication idea can be used to reduce polynomial size weights into constant weights (see Figure 1.3). Nevertheless, this would inevitably introduce a polynomial size blow-up in the circuit size. We further emphasize that our focus is to achieve this weight size reduction from polynomial weights to constant weights with at most a constant size blow-up in the circuit size.

The Main Technique to Obtain Upper and Lower Bounds

Suppose that we are given an exact threshold function $f(X) = \mathbb{1}\{w^T X = b\}$ with "large" weights and we want to compute this function by using "small" weight threshold circuits. Here, $w \in \mathbb{Z}^n$ and $b \in \mathbb{Z}$. We use $\mathcal{X}_{w,b}$ to denote the solution set of the single linear equation $w^T X = b$ for binary vectors X. The formal definition is given in Eq. (2.1).

$$\mathcal{X}_{w,b} = \{X \in \{0,1\}^n | w^T X = b\} \tag{2.1}$$

$$\mathcal{X}_{A,B} = \{X \in \{0,1\}^n | AX = B\} \tag{2.2}$$

By definition, if $X \in \mathcal{X}_{w,b}$, then $f(X) = 1$ and $f(X) = 0$ otherwise. One idea to reduce the weights could be to find a matrix $A \in \mathbb{Z}^{m \times n}$ and a vector $B \in \mathbb{Z}^m$ so that

$\mathcal{X}_{w,b} = \mathcal{X}_{A,B}$ (as defined in Eq. (2.2)) and therefore, and $X \in \mathcal{X}_{w,b}$ is also a solution to the linear system of equations $AX = B$. Obviously, we want A to have "smaller" weights and we have an explicit size constraint on the entries of A, i.e. $|a_{ij}| \leq W$ for all i, j where $W \in \mathbb{Z}$.

Thus, one can interpret a transformation from a single gate into a circuit as an equivalent representation of the solution sets $\mathcal{X}_{w,b}$ and $\mathcal{X}_{A,B}$. More precisely, the equivalence is sufficient but not necessary. If we can show the equivalence of solution sets and the existence of an A with m rows, then we can prove an **upper bound** on the complexity of m, the number of gates in the first layer, given the weight constraint W and input size n.

Given n and W as parameters and $f(X)$, how can we prove lower bounds for the number of rows for the matrix A? Suppose that there exist $X, Y \in \{0,1\}^n$ such that $X \neq Y$ and $AX = AY$ but, in contrast, $w^T X = b$ and $w^T Y \neq b$. Thus, given the vector AX, there is no way to understand whether the input vector X solves $w^T X = b$ or not. If we can guarantee the existence of such X and Y for any $A \in \mathbb{Z}^{m \times n}$, then we can prove a **lower bound** on the complexity of m depending on the values of n and W.

One can see that the analysis of linear threshold functions using solutions sets is not very different. The ideas are the same after defining that $\mathcal{X}'_{w,b} = \{X \in \{0,1\}^n | w^T X \geq b\}$ and $\mathcal{X}'_{A,B} = \{X \in \{0,1\}^n | AX \geq B\}$. In fact, any activation function can be used in the definition of the solution sets.

Both of the techniques to prove lower and upper bounds rely on linear algebraic properties of $w^T X = b$ and $AX = b$. Depending on the context, they can be very useful or limiting when we try to match the upper and lower bounds. We emphasize that this linear algebraic view is only a tool to analyze the transformations of threshold gates with "large" weights into threshold circuits with "small" weights.

A transformation as in Figure 2.1 also entails the use of a *top* gate. The top gate is specified when we give upper bounds and constructions. For instance, for the 3-input PARITY example in Eq. (1.2) or Eq. (1.3), the top gate can be taken to be an m-input OR gate, i.e., $\mathbb{1}\{X_1 + \cdots + X_m \geq 1\}$.

The lower bounds that we prove applies to any top gate for the circuits. This makes the lower bounds powerful because they are general and apply to any activation function. However, one caveat is that given a specific top gate, the lower bound might not be tight.

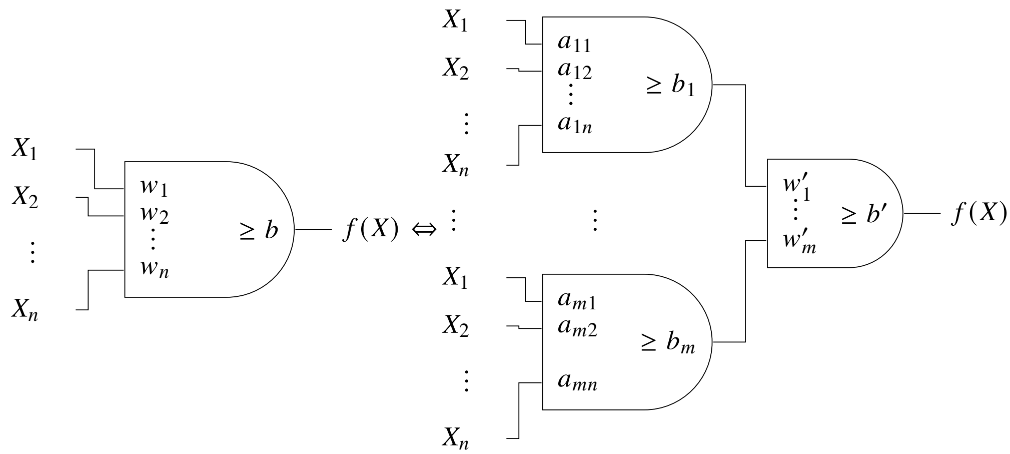


Figure 2.1: An arbitrary transformation of a single linear threshold gate into a depth-2 linear threshold circuit. Instead of a single equation dot-product $w^T X$, we consider a matrix-vector product AX where each entry of A is bounded by some "small" integer. We want both circuits to compute the same Boolean function $f(X)$. Although in this circuit, we kept the weights of the top gate, namely, w' arbitrary, they also obey the weight constraints that we impose.

Our results mainly focuses on the EQ function and its linear threshold counterpart COMP function. Then, we generalize the ideas to the class of the threshold functions using DOM functions. Our key contribution is to push the understanding of linear algebraic and number theoretic techniques further by the perspective of DOM functions.

Previously, in (Roychowdhury, Orlitsky, and Siu, 1994), it is shown that to compute the EQ function using arbitrary perceptrons with a weight matrix $A \in \mathbb{Z}^{m\times n}$ where $|a_{ij}| \leq n^c$ for a fixed constant c, one requires

$$\frac{n}{2(c+1)(\log 2n)} \leq m \leq \frac{n}{c\log_3 2n} \tag{2.3}$$

The same lower bound applies to the COMP within a constant factor. The proof technique for the lower bound is via Communication Complexity arguments, a popular and powerful technique with many different applications in circuit complexity theory (Alon, Frankl, and Rödl, 1985; Chattopadhyay and Mande, 2018; Goldmann, 1994; Forster, 2002; Paturi and Simon, 1986). To compute the EQ function, an explicit construction satisfies the upper bound by partitioning the input into logarithmic size chunks. An important remark is that when $c = 0$, i.e., $|a_{ij}| = O(1)$, the upper bound blows up. The natural open question is to find a construction with asymptotically the same m using constant weights. In this thesis, we prove this result both existentially and constructively using techniques in linear algebra and number theory whose applications are novel in circuit complexity theory.

We start with the idea to compute any threshold gate using a DOM gate using binary expansion of the weights called *the domination form of threshold gates*. In Section 2.2, we give small weight constructions of exact DOM functions using the *Chinese Remainder Theorem* (CRT) and the *Prime Number Theorem* (PNT) from previous works. We note that the domination form helps us simplify the size complexity by a factor of n for some functions. Then, in Section 2.3, we define EQ matrices to compute EQ functions using small coefficient matrices. For these matrices, the number of rows needs to satisfy Eq. (2.3) and we prove this result alternatively using Siegel's Lemma (for an English translation of the original result, see Siegel, 2014). By a converse theorem on Siegel's Lemma by Beck, we show that constant coefficient EQ matrices exist with optimal rate (Beck, 2017). Furthermore, an explicit construction with $\{-1, 0, 1\}$ entries and optimal rate is given based on the extensions of Sylvester-type Hadamard matrices. In Section 2.4, an MDS-like generalization of these matrices is given and an existential result is proven using Berry-Esseen Theorem (Berry, 1941; Esseen, 1942). In Section 2.5, we give a short application of Siegel's Lemma to prove a lower bound on the number of perceptron needed to compute a DOM function. Finally, in Section 2.6, we apply our results to construct depth-2 threshold circuits with the best known size and weight complexity for the DOM functions with $\mathcal{S} = 1$, which includes the EQ, COMP, CARRY, and OMB.

2.1 Computing Any Threshold Function with a Linear Transformation and a DOM gate

Typically, a threshold function is characterized by the values of its weights. It was shown that for the COMP and the EQ functions, the weights should be represented by $\Omega(n)$ bits (Siu and Bruck, 1991; Babai et al., 2010). In general, the weights of a threshold function can be represented by $O(n \log n)$-bits and this is a tight bound (Alon and Vũ, 1997; Babai et al., 2010; Håstad, 1994; Saburo Muroga, 1971). First, we define L to be the least integer satisfying $|w_i| < 2^L$ for all i. Consider the linear form $F(X) = \sum_{i=1}^{n} w_i X_i - b$ and let us write the binary expansion of each weight w_i and b. Hence, we obtain an $L \times (n+1)$ incidence matrix W where w_{ji} is 1 if 2^{j-1} is in the binary expansion of w_i and 0 otherwise. If a weight is negative, we consider the binary expansion of its magnitude and then invert the sign (e.g. $-5 = -(2^2 + 2^0) = -2^2 - 2^0$). Also, we treat $x_0 = 1$ as a fixed input for this representation with $w_0 = -b$. We use w_b to denote the powers of two in ascending order.

$$\begin{aligned} F(X) &= \sum_{i=0}^{n} w_i X_i = \sum_{i=0}^{n} \Big(\sum_{j=1}^{L} 2^{j-1} w_{ji} \Big) X_i \\ &= \sum_{j=1}^{L} 2^{j-1} \Big(\sum_{i=0}^{n} w_{ji} X_i \Big) = \sum_{j=1}^{L} 2^{j-1} X'_j \end{aligned} \tag{2.4}$$

In other words, we fix the weights to be powers of two and represent a threshold function by a larger input alphabet $\{-n-1, ..., n+1\}$. We call this representation *Domination Form of Threshold Functions*. Even though this alphabet has cardinality $2n+3$, we call such an alphabet as $(n+1)$-ary as an abuse of notation because the values are from $\{-n-1, \dots, n+1\}$. If a power of two 2^{j-1} appears in the binary expansion of weights $w_{i_1}, w_{i_2}, ...$, then the corresponding inputs $X_{i_1}, X_{i_2}, ...$ are grouped together in X'_j. In this form, we can compute a threshold function using an $(n+1)$-ary L input DOM gate as in Fig. 2.2. In linear algebra notation, we have the following notation and equalities assuming $X_0 = 1$ and $w_0 = -b$.

$$w_b^T = \begin{bmatrix} 2^0 & 2^1 & \cdots & 2^{L-1} \end{bmatrix} \tag{2.5}$$

$$F(X) = w^T X = (w_b^T W) X = w_b^T (WX) = w_b^T X' \tag{2.6}$$

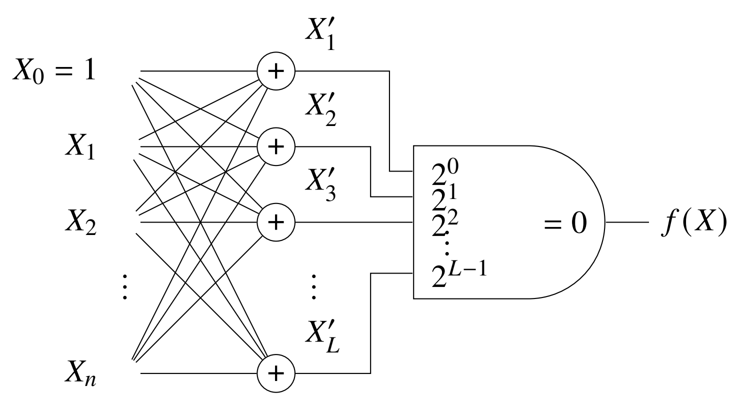


Figure 2.2: The construction of an exact threshold gate by an exact DOMINATION gate and a layer of adders. The connectivity of the first layer is given by the incidence matrix W.

To realize this construction in general, we only need to compute WX in a layer of adders to obtain X' and then feed it to a DOM gate of higher alphabet. The connectivity graph W has only $\{-1, 0, 1\}$ weights on edges. In this form, we can analyze the connectivity graph and the DOM gate separately.

Firstly, different threshold functions yield different X' vectors and we want to quantify the incidence relation between X and X'. We define a metric of the *sparsity of the graph*, $\mathcal{S}$, by counting the maximum possible value each X'_i can take. That is, we define $\mathcal{S} = \max_{X'} \max_i |X'_i|$. Since an adder is connected to at most $(n+1)$ inputs, it follows that $\mathcal{S} \leq n+1$ and in general, $\mathcal{S} = O(n)$. This $\mathcal{S}$ quantity was also of consideration in various other papers with an almost equivalent definition (Amano and Maruoka, 2005; Hofmeister, 1996). For the COMP and the EQ, we observe that $\mathcal{S} = 1$ and $L = n$.

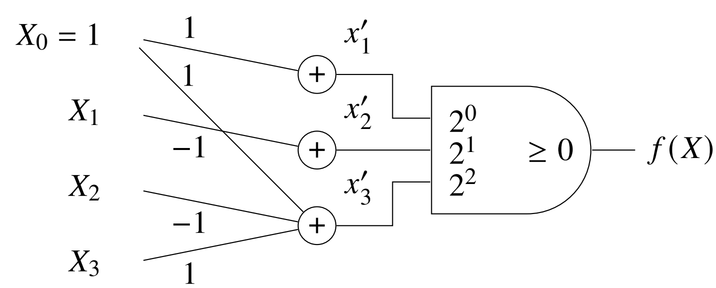


Figure 2.3: An example of a construction of a linear threshold function by a linear DOM gate where $f(X) = \mathbb{1}\{-2X_1 - 4X_2 + 4X_3 \geq -5\}$ with $\mathcal{S} = 2$.

We emphasize that this construction provides a **universal** method to compute arbitrary threshold functions by a graph that depends on the weights and that feeds the inputs to an $(n+1)$-ary DOM gate. Namely, if there is a way to simulate a DOM gate using other types of gates, it can be used to compute any given threshold function. The analysis of threshold functions in domination forms is novel even though using binary expansions for weight representations in circuit theory can be found in some previous works (Siu and Bruck, 1991).

2.2 Small Weight Circuit Constructions of Exact DOM functions

We start by introducing the CRT-based weight reduction technique given in previous works (Amano and Maruoka, 2005; Hansen and V. V. Podolskii, 2010; Hofmeister, 1996). For an integer z and modulo base p, we denote the modulo operation by $[z]_p$, which maps the integer to the values in $\{0, ..., p-1\}$.

Theorem 4. *Suppose that we are given a function* $\mathbb{1}\{F(X) \in C\} = \mathbb{1}\{w_0 + \sum_{i=1}^{n} w_i X_i \in C\}$ *where C is some subset of non-negative integers whose largest element is denoted by C_{max}. Let L be the least integer satisfying $|w_i| < 2^L$. Let $C_{max} < p_1 < p_2 < \cdots < p_r$ be prime numbers and s be the smallest integer such that* $\prod_{i=1}^{s} p_i > (n+1)2^L$. *Then,* *1. If $F(X) \in C$, then $[F(X)]_{p_i} \in C$ for all p_i, $i \in \{1, ..., r\}$.*

2. If $F(X) \notin C$, then $[F(X)]_{p_i} \in C$ for less than $s|C|$ many primes.

Furthermore, $s = O(L/\log L)$ is sufficient and $p_r = O(C_{max}L)$ given that 2^L dominates the factor $n+1$ asymptotically and $C_{max} = poly(n)$.

Proof. The proof basically follows the same idea in (Hofmeister, 1996). The first claim is true trivially $F(X) = k = [F(X)]_{p_i}$ for all $i \in \{1, \dots, r\}$ as $k < C_{max} < p_1 < \cdots < p_r$. For the second claim, suppose to the contrary that $[F(X)]_{p_i} \in C$ for at least $s|C|$ many primes. Then, there is an integer $z \in C$ such that for at least s many primes $[F(X)]_{p_i} = z$ by Pigeonhole Principle. By the CRT, we can construct $F(X)$ uniquely to find that it is actually z because $\prod_{i=1}^{s} p_i > (n+1)2^L$. This contradicts $F(X) \notin C$.

To find s, we first assume that $P = p_1 < \cdots < p_r$ for some prime number P. Then, let $\prod_{i=1}^{s} p_i > P^s > (n+1)2^L$. Taking the logarithm of both sides, we have

$$s \log P > L + \log(n+1) \tag{2.7}$$

Let P be the $L/\log L^{th}$ prime number and pick large enough $s \sim L/\log L$ so that $p_1 > P$. Then, $P \sim L$ by the PNT and the inequality is satisfied as we assume that 2^L dominates $n+1$ asymptotically. We pick $r \geq s|C| + 1$ so that both cases are separable and we use the PNT to find $p_r = O(C_{max}L)$. □

One immediate remark is that for exact threshold functions, $C = \{0\}$. However, for linear threshold functions, C contains all positive values and Theorem 4 does not seem useful. However, by first transforming a linear threshold gate into an exact threshold circuit with an disjoint OR gate on top, it is possible to apply CRT in an efficient way (Amano and Maruoka, 2005; Hofmeister, 1996).

We reduce each weight by computing $[w_j]_{p_i}$ for each weight w_j and prime p_i. Let us define $F_i(X) = [w_0]_{p_i} + \sum_{j=1}^{n} [w_j]_{p_i} X_j$. If we are to use only exact threshold gates in the first layer, we need to compute congruences explicitly, that is, we need to check if $F_i(X) = kp_i$ for some integer k. To do so, the general approach is to use $n+1$ gates together computing $F_i(X) \in \{0, p_i, 2p_i, ..., np_i\}$ because $F_i(X) < (n+1)p_i$. Finally, these gates are connected to a symmetric Boolean gate $\text{EXACT}_r = \mathbb{1}\{|X| = r\}$ to check if $\sum_{k=0}^{n} \sum_{i=1}^{r} \mathbb{1}\{F_i(X) = kp_i\} = r$.

Since we reduce each weight by prime moduli, the maximum weight size becomes $O(p_r) = O(L)$ and the maximum bias term becomes $O(np_r) = O(nL)$ as we need to check if $F_i(X) = np_i$ for the i^{th} prime. Typically, we ignore the size of the bias terms when we analyze the weight size of a circuit. In this construction, the total number of gates is $(n+1)r + 1 = O(nL/\log L)$. If we take $L = O(n \log n)$, the circuit size becomes $O(n^2)$. This is the best known size complexity upper bound for the depth-2 exact threshold circuit constructions with polynomially large weights in n. For exact threshold functions, this size result result $O(n^2)$ was not obtained previously but the technique was observed fully in (Hansen and V. V. Podolskii, 2010).

For the EQ function, the direct application of CRT gives the same $O(n^2/\log n)$ size complexity with $O(n)$ weights. Consider the following form of EQ function in Eq. (2.9). We call it the *naïve* form of the EQ.

$$\mathrm{EQ}_{2n}(X, Y) = \mathbb{1}\Big\{\sum_{i=1}^{n} 2^{i-1}X_i = \sum_{i=1}^{n} 2^{i-1}Y_i\Big\} \tag{2.8}$$

$$= \mathbb{1}\Big\{\sum_{i=1}^{n} 2^{i-1}X_i + \sum_{i=1}^{n}(-2^{i-1})Y_i = 0\Big\} \tag{2.9}$$

Theorem 5. *There exists a depth-2 exact threshold circuit for the $2n$-input* EQ *function with weights in naïve form with $O(n^2/\log n)$ size complexity and $O(n)$ weights.*

Proof. Since our definition of the modulo function is only for positive values, we get $[-2^{j-1}]_{p_i} = p_j - [2^{j-1}]_{p_i}$. Therefore, for each prime number p_i, we obtain

$$F_i(X) = \sum_{j=1}^{n} [2^{j-1}]_{p_i}(X_i - Y_i) + p_i|Y| \tag{2.10}$$

where $|Y|$ denotes the number of 1s in the vector Y. Therefore, to compute $EQ(X, Y)$ using CRT and PNT, we have to check $F_i(X) \in \{0, p_i, \ldots, np_i\}$ necessarily. Since $L = n$, we obtain $O(n^2/\log n)$ size and the maximum weight size becomes $O(n)$ by the PNT. We connect the gates into an AND gate to construct the depth-2 circuit. □

In contrast, if the weights of the EQ is in domination form, we can directly reduce the size complexity to $O(n/\log n)$, which is asymptotically optimal (see Eq. (2.3)). We first give the CRT-based construction for $\mathcal{S}$-ary L input DOM functions.

Theorem 6. *There exists a depth-2 exact threshold circuit which computes an $\mathcal{S}$-ary L input exact DOMINATION function and its size is bounded by $O(\mathcal{S}L^2/\log L)$ with weights at most $O(\mathcal{S}L^2)$.*

Proof. For L input DOMINATION functions, recall that $|w_i| < 2^L$ including w_0. Therefore, Theorem 4 gives an upper bound on $r = O(L/\log L)$ similar to arbitrary exact threshold functions.

Since $|F_i(X')| < \mathcal{S}Lp_i$, to compute $[F(X')]_{p_i}$ for a given prime number p_i, we need to check if $F_i(X') \in \{-(\mathcal{S}L-1)p_i..., -p_i, 0, p_i, ..., (\mathcal{S}L-1)p_i\}$. By the PNT, $p_r = O(r\log r) = O(L)$ and therefore, the maximum weight size becomes $O(L)$ and the largest bias term becomes $O(\mathcal{S}L^2)$. We connect these gates to an EXACT_r gate (i.e. $\mathbb{1}\{|X| = r\}$) and the size complexity becomes $O(\mathcal{S}L^2/\log L)$. □

It turns out that if $\mathcal{S} = O(n)$ and $L = O(n\log n)$, the size complexity becomes $O(n^3\log n)$. This is higher than $O(n^2)$ and it seems that there is no use of computing an exact threshold function by using an exact DOMINATION gate. Even if when $\mathcal{S} = 1$, the construction seems not to be optimal. We observe that if $\mathcal{S} = 1$, then the modulo checks $F_i(X) \in \{0, p_i, \ldots, np_i\}$ are not necessary except $F_i(X) = 0$.

Theorem 7. *If $\mathcal{S} = 1$, there exists a depth-2 exact threshold circuit which computes an $\mathcal{S}$-ary L input exact DOMINATION function and its size is bounded by $O(L/\log L)$ with weights at most $O(L)$.*

Proof. Basically, we follow Theorem 6 but we will prove that if $\mathcal{S} = 1$, checking $F_i(X') = 0$ suffices.

We observe that if $\mathcal{S} = 1$, the equation $\sum_{j=1}^{L} 2^{j-1}X'_j = 0$ admits a unique trivial solution $X' = 0$. If $X' = 0$, $F_i(X') = \sum_{j=1}^{L}[2^{j-1}]_{p_i}X'_j = 0$ and there is no need to check if $F_i(X') = kp_i$ for $k \neq 0$. Then, the size complexity becomes $O(L/\log L)$. By the PNT, $p_r = O(r\log r)$ and the weights are bounded by $O(L)$. □

For the EQ function, Theorem 7 gives asymptotic efficiency with regards to the known lower bounds and our lower bound given in Section 2.5 (Roychowdhury, Orlitsky, and Siu, 1994) (see Theorem 8). We conclude that depending on the representation of weights, it is possible to obtain different upper bounds on the size that computes a given function. In the next sections, starting with the EQ function, we will show how to use the domination form of threshold functions when we apply weight reduction techniques as well as how to obtain upper and lower bounds.

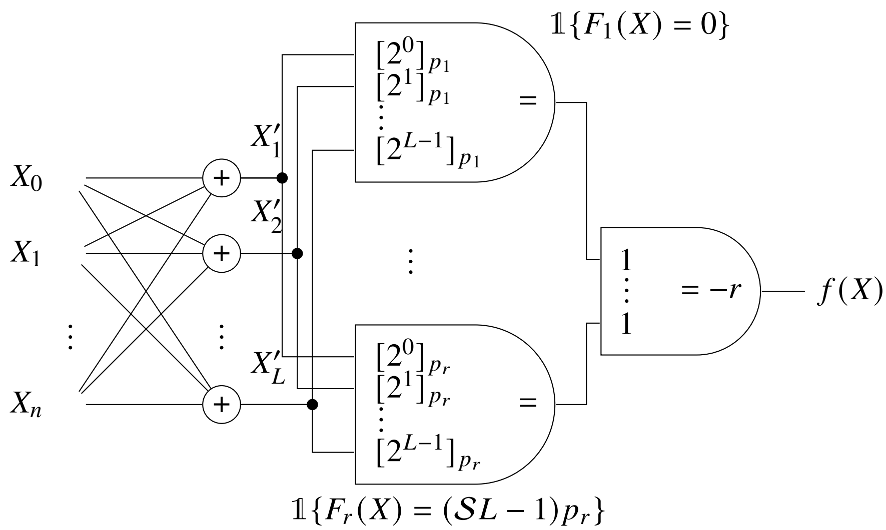


Figure 2.4: Depth-2 exact threshold circuit construction of an arbitrary exact threshold gate based on an exact DOMINATION function and the CRT. Notice that this DOMINATION gate construction is universal and the function is purely determined by the graph of summation layer. We don't put the bias terms in the circuit as there is no sufficient space.

2.3 Bijective Mappings from Finite Fields to Integer Vectors

Lower Bounds on the EQ Matrices

Now, let us consider the following single linear equation where the weights are fixed to the ascending powers of two and x_is are real numbers.

$$w_b^T x = \sum_{i=1}^{n} 2^{i-1} x_i = 0 \tag{2.11}$$

As the weights of w_b define a bijection between n-bit binary vectors and integers, $w_b^T x = 0$ does not admit a non-trivial solution for the alphabet $\{-1, 0, 1\}^n$. This condition also has a combinatorial meaning in the following sense: How small could the elements of $\mathcal{W} = \{w_1, w_2, \cdots, w_n\}$ be if $\mathcal{W}$ has all distinct subset sums (DSS)? If the weights satisfy this property, called the **DSS property**, they can define a bijection between $\{0, 1\}^n$ and integers. Erdős conjectured that the largest weight $w \in \mathcal{W}$ is upper bounded by $c_0 2^n$ for some $c_0 > 0$, and therefore choosing powers of two as the weight set is asymptotically optimal (Erdős, 1983). The best known result for such weight sets yields $0.22002 \cdot 2^n$ and currently, the best lower bound is $\Omega(2^n/\sqrt{n})$ (Bohman, 1998; Dubroff, Fox, and Xu, 2021; Guy, 2004).

It is shown that the non-trivial solution condition for a single equation is a necessary and sufficient condition to compute the EQ function given in Eq. (1.7) (Babai et al., 2010). We extend this property to m many rows to define EQ *matrices* which give a bijection between $\{0,1\}^n$ and $\mathbb{Z}^m$. We further prove that even though EQ matrices are not necessary to compute an EQ function, they still satisfy the optimal lower bounds. Consequently, there is no asymptotic trade-off to choose an EQ matrix to compute the EQ function given m and W.

Definition 7. *A matrix $A \in \mathbb{Z}^{m\times n}$ is an* EQ matrix *if the homogeneous system $Ax = 0$ has no non-trivial solutions in $\{-1,0,1\}^n$.*

Theorem 8. *To compute* $\mathrm{EQ}_{2n}(X,Y) = \mathbb{1}\{X = Y\}$, $\Omega(n/\log nW)$ *many arbitrary perceptrons with weight constraint W are needed.*

Theorem 8 can be seen as a restatement of the lower bound in Eq. (2.3). In general, the weight matrix of the first layer of a circuit computing EQ is in the form $\begin{bmatrix} A & B \end{bmatrix} \in \mathbb{Z}^{m\times 2n}$. We will show that both A and B must be EQ matrices.

Let $A \in \mathbb{Z}^{m\times n}$ be an EQ matrix with the weight constraint $W \in \mathbb{Z}$ such that $|a_{ij}| \leq W$ for all i, j and let R denote the *rate* of the matrix A, which is n/m. It is clear that any full-rank square matrix is an EQ matrix with $R = 1$. Given any W, how large can this R be? For the maximal rate, a necessary condition can be proven by Siegel's Lemma (Siegel, 2014). There are different variations of Siegel's Lemma and the one we use below suits our needs (Bombieri and Gubler, 2006). Conventionally, we use $||.||_\infty$ to denote the infinity norm.

Lemma 1 (Siegel's Lemma)**.** *Consider any integer matrix $A \in \mathbb{Z}^{m\times n}$ with $m < n$ and $|a_{ij}| \leq W$ for all i, j and some integer W. Then, $Ax = 0$ has a non-trivial solution for an integer vector $x \in \mathbb{Z}^n$ such that $||x||_\infty \leq (\sqrt{n}W)^{\frac{m}{n-m}}$.*

Lemma 2. *For any $\{-1,0,1\}^{m\times n}$ EQ matrix, $R = \frac{n}{m} \leq 1 + \log\sqrt{n}W$.*

Proof. By Siegel's Lemma, we know that for a matrix $A \in \mathbb{Z}^{m\times n}$, the homogeneous system $Ax = 0$ attains a non-trivial solution in $\{-1,0,1\}^n$ when

$$||x||_\infty \leq (\sqrt{n}W)^{\frac{m}{n-m}} \leq 2^{1-\epsilon} \tag{2.12}$$

for some $\epsilon > 0$. Then, we deduce that $R = \frac{n}{m} \geq \frac{1}{1-\epsilon}\log(\sqrt{n}W) + 1$. Obviously, an EQ matrix cannot obtain such a rate. Taking $\epsilon \longrightarrow 0$, we conclude the best upper bound is $R \leq 1 + \log\sqrt{n}W$. □

The previous rate upper bound given in (Roychowdhury, Orlitsky, and Siu, 1994) and Eq. (2.3) is looser by a constant factor. Therefore, our result gives a tighter upper bound on the rate. We also note that a similar result can be obtained by the matrix generalizations of Erdős' Distinct Subset Sum problem (Costa, Della Fiore, and Dalai, 2021).

Lemma 2 equivalently states that if $m = o(n/\log nW)$, there does not exist any EQ matrix. We are now ready to prove Theorem 8.

Proof of Theorem 8. We use the aforementioned technique to prove the lower bound for the EQ. Suppose that the weight matrix of the first layer of the circuit is $\begin{bmatrix} A & B \end{bmatrix} \in \mathbb{Z}^{m\times 2n}$ so that it computes $AX + BY$. For such a matrix in the circuit computing the EQ function, a quadruple of n-bit binary vectors (X, Y, Z, T) such that $AX + BY = AZ + BT$ and $X \neq Y$ but $Z = T$ cannot exist. In other words, if $\mathrm{EQ}(X, Y) = 0$ but $\mathrm{EQ}(Z, T) = 1$, they cannot be mapped to the same vector by the weight matrix $\begin{bmatrix} A & B \end{bmatrix}$.

Suppose that B is not an EQ matrix. Then, if $X = Z$, we obtain $B(Y - Z) = 0$ and there exists a non-trivial solution such that $Y - Z \in \{-1, 0, 1\}^n$ by definition. Therefore, depending on the entries of such a non-trivial solution, we can recover non-unique Y and Z such that the quadruple (X, Y, Z, T) where $X \neq Y$ but $Z = T$ always exists (for instance, if $Y - Z = (1, 0, 0, -1)$, it is clear that $Y_1 = 1, Z_1 = 0$ and $Y_4 = 0, Z_4 = 1$. On the other hand, $Y_2 = Z_2$, $Y_3 = Z_3$ and they can be both 0 or 1). Same arguments apply to the matrix A by taking $Y = Z$ to begin with. In conclusion, both A and B must be EQ matrices and $m = \Omega(n/\log nW)$. □

The sparsity of the EQ matrices can also be treated in an asymptotical manner. While an EQ matrix can admit a large number of zeros in the entries, if the rows are sparse, then the rate of the matrix will get lower. To prove this, we can refer the following version of Siegel's Lemma which can be proven easily by Pigeonhole Principle. Then, we follow the same steps in Lemma 2. *The maximal* $\{0, 1\}$*-sum of a vector* v is defined by $\max_{X\in\{0,1\}^n} |v^T X|$.

Lemma 3 (Siegel's Lemma (modified))**.** *Consider any integer matrix* $A \in \mathbb{Z}^{m\times n}$ *with* $m < n$ *and the maximal* $\{0, 1\}$*-sum of all rows of* A *is bounded by* $C \in \mathbb{Z}$ *in magnitude. Then,* $Ax = 0$ *has a non-trivial solution for an integer vector* $x \in \mathbb{Z}^n$ *such that* $\max_j |x_j| \leq C^{\frac{m}{n-m}}$.

Proof. The matrix A gives a mapping from $\mathbb{Z}^n$ to $\mathbb{Z}^m$. Assume that $0 \leq x_i \leq B$ for some integer B and all $i \in \{1, \dots, n\}$. Since each $\{0,1\}$-sum of rows of A is bounded by C in magnitude, we have $(CB)^m$ possible integer vectors in the range of A. By Pigeonhole Principle, if $(B+1)^n > (CB)^m$, then there exist $x, y \in \mathbb{Z}^n$ such that $Ax = Ay$ and there is a non-trivial solution to $Ax = 0$ such that $\max_j |x_j| \leq B$. If $B^n = (CB)^m$, then $(B+1)^n > (CB)^m$ is implied. Therefore, $\max_j |x_j| \leq C^{\frac{m}{n-m}}$ follows. □

Let S be the maximum number of non-zero entries in any row of an EQ matrix. Since $C \leq SW$, we obtain $R \leq 1 + \log SW$. For example, when $S = O(1)$, the asymptotically maximal rate $R = O(\log n)$ is achievable only if $W = poly(n)$, similar to the upper bounds in previous works or CRT-based constructions. Moreover, if $S = W = 1$, then the maximal rate $R = 1$ where the matrix itself can only be a permutation matrix (negation of the rows is allowed).

Constructions of the EQ Matrices

For an explicit construction, it is possible to achieve the optimal rate $R = O(\log nW)$ if we allow $W = poly(n)$. This can be done by the Chinese Remainder Theorem (CRT) and the Prime Number Theorem (PNT) as in Section 2.2. Since we can also encode an integer $0 \leq Z < 2^n$ by its binary expansion, the CRT gives a bijection between $\mathbb{Z}^m$ and $\{0,1\}^n$ as long as $p_1 \cdots p_m > 2^n$. By taking modulo p_i of Eq. (2.11), we can obtain the following matrix, defined as a *CRT matrix*:

$$\begin{bmatrix} [2^0]_3 & [2^1]_3 & [2^2]_3 & [2^4]_3 & [2^5]_3 & [2^6]_3 & [2^7]_3 \\ [2^0]_5 & [2^1]_5 & [2^2]_5 & [2^4]_5 & [2^5]_5 & [2^6]_5 & [2^7]_5 \\ [2^0]_7 & [2^1]_7 & [2^2]_7 & [2^4]_7 & [2^5]_7 & [2^6]_7 & [2^7]_7 \\ [2^0]_{11} & [2^1]_{11} & [2^2]_{11} & [2^4]_{11} & [2^5]_{11} & [2^6]_{11} & [2^7]_{11} \end{bmatrix} \tag{2.13}$$

$$= \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\ 1 & 2 & 4 & 3 & 1 & 2 & 4 & 3 \\ 1 & 2 & 4 & 1 & 2 & 4 & 1 & 2 \\ 1 & 2 & 4 & 8 & 5 & 10 & 9 & 7 \end{bmatrix}_{4\times 8} \tag{2.14}$$

We have $Z < 256$ since $n = 8$ and $3 \cdot 5 \cdot 7 \cdot 11 = 1155$. Therefore, this CRT matrix is an EQ matrix. In general, by the PNT, one needs $O(n/\log n)$ many rows to ensure that $p_1 \cdots p_m > 2^n$. Moreover, W is bounded by the maximum prime size p_m, which is $O(n)$ again by the PNT (see Theorem 4).

One of our main contributions for the optimal rate EQ matrices is to prove the existence of EQ matrices with constant weight size. This is a simple consequence of Beck's Converse Theorem on Siegel's Lemma (Beck, 2017).

Theorem 9 (Beck's Theorem for the Converse of Siegel's Lemma (Beck, 2017))**.** *There is a (small) positive constant $c_0 > 0$ with the following property: for every pair $n > m \geq 1$ of positive integers satisfying $n \geq 3m/2$, there exists a matrix A with m rows and n columns, entries of $\{-1, 1\}$ such that for **every** non-trivial integer solution of the homogeneous system $Ax = 0$, we have*

$$\max_j |x_j| > c_0(\sqrt{n})^{\frac{m}{n-m}} \tag{2.15}$$

In addition, for large m, the overwhelming majority of the $m \times n$ ± 1 matrices A satisfy the theorem. The violators represent an exponentially small $O(2^{-m/2})$ part of the total 2^{mn}.

The condition $n \geq 3m/2$ is to guarantee that the non-trivial solutions of $Ax = 0$ should have sufficient number of multiplicities (repetition of the same numbers) because otherwise, the RHS of the Eq. (2.15) will be $\omega(n)$ (i.e. it will be super-linear). Our focus is the case when $\max_j |x_j| > 1$ and Beck's argument should therefore work. The proof idea is based on Halász Theorem on the anti-concentration inequalities and generating functions (Halász, 1977). In Section 2.4, we also build a framework on anti-concentration inequalities and independent of Siegel's Lemma, it is of no surprise that a similar existential result on EQ matrices is obtained (see Theorem 13 with $r = 1$). However, Beck's Converse Theorem is much more powerful in its scope and in the quantification of the density of EQ matrices over all $\{-1, 1\}$-matrices.

In addition to these existential results, we give an explicit construction where $W = 1$ (i.e. $a_{ij} \in \{-1, 0, 1\}$) and asymptotic efficiency in rate is achieved up to vanishing terms.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}_{4\times 8} \tag{2.16}$$

One can verify that the matrix in (2.16) is an EQ matrix by checking $Ax \neq 0$ for $x \in \{-1, 0, 1\}^8 \setminus \{0\}$ and its rate is twice the trivial rate being the same in (2.13).

Therefore, due to the optimality in the weight size, this construction can replace the CRT matrices in the constructions of EQ matrices.

The $m \times n$ EQ matrix construction we give here is based on Sylvester's construction of Hadamard matrices. It is an extension of it to achieve higher rates with the trade-off that the matrix is no longer full-rank.

Definition 8. *An* $n \times n$ Hadamard matrix *is a* $\{-1, 1\}$*-matrix with orthogonal columns. The* order *of the Hadamard matrix is* n*. A* Sylvester-type Hadamard matrix H_{2k} *is constructed by the following recursion where* H_k *is a Hadamard matrix of order* $k \in \mathbb{Z}$*.*

$$H_{2k} = \begin{bmatrix} H_k & H_k \\ H_k & -H_k \end{bmatrix}_{2k \times 2k} \tag{2.17}$$

Theorem 10. *Suppose we are given an EQ matrix* $A_0 \in \{-1, 0, 1\}^{m_0 \times n_0}$*. At iteration* k*, we construct the following matrix* A_k*:*

$$A_k = \begin{bmatrix} A_{k-1} & A_{k-1} & I_{m_{k-1}} \\ A_{k-1} & -A_{k-1} & 0 \end{bmatrix} \tag{2.18}$$

A_k *is an EQ matrix with* $m_k = 2^k m_0$*,* $n_k = 2^k n_0(\frac{k}{2}\frac{m_0}{n_0} + 1)$ *for any integer* $k \geq 0$*.*

Proof. We will apply induction. The case $k = 0$ is trivial by assumption. For the system $A_k x = z$, let us partition the vector x and z in the following way:

$$\begin{bmatrix} A_{k-1} & A_{k-1} & I_{m_{k-1}} \\ A_{k-1} & -A_{k-1} & 0 \end{bmatrix} \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \end{bmatrix} = \begin{bmatrix} z^{(1)} \\ z^{(2)} \end{bmatrix} \tag{2.19}$$

Then, setting $z = 0$, we have $A_{k-1}x^{(1)} = A_{k-1}x^{(2)}$ by the second row block. Hence, the first row block of the construction implies $2A_{k-1}x^{(1)} + x^{(3)} = 0$. Each entry of $x^{(3)}$ is either 0 or a multiple of 2. Since $x_i \in \{-1, 0, 1\}$, we see that $x^{(3)} = 0$. Then, we obtain $A_{k-1}x^{(1)} = A_{k-1}x^{(2)} = 0$. Applying the induction hypothesis on $A_{k-1}x^{(1)}$ and $A_{k-1}x^{(2)}$, we see that $A_k x = 0$ admits a unique trivial solution in $\{-1, 0, 1\}^{n_k}$.

To see that $m_k = 2^k m_0$ is not difficult. For n_k, we can apply induction. □

To construct the matrix given in Eq. (2.16), we can use $A_0 = I_1 = \begin{bmatrix} 1 \end{bmatrix}$, which is a trivial rate EQ matrix, and take $k = 2$. By rearrangement of the columns, we also see that there is a 4×4 Sylvester-type Hadamard matrix in this matrix.

Using Lemma 2 for our construction with $A_0 = \begin{bmatrix}1\end{bmatrix}$, we compute the upper bound on the rate and the real rate

$$R_{upper} = \frac{k+1+\log(k+2)}{2}, \quad R_{constr} = \frac{k}{2}+1 \tag{2.20}$$

$$\frac{R_{upper}}{R_{constr}} = 1 + \frac{\log(k+2)-1}{k+2} \implies \frac{R_{upper}}{R_{constr}} \sim 1 \tag{2.21}$$

which concludes that the construction is **optimal** in rate up to vanishing terms. By deleting columns, we can generalize the result to any $n \in \mathbb{Z}$ to achieve optimality up to a constant factor of 2.

We conjecture that one can extend the columns of any Hadamard matrix to obtain an EQ matrix with entries $\{-1, 0, 1\}$ achieving rate optimality up to vanishing terms. In this case, the Hadamard conjecture implies a rich set of EQ matrix constructions.

We can also consider q-ary representation of integers in a similar fashion by extending the definition of EQ matrices to this setting. In our analysis, we always treat q as a constant value.

Definition 9. *A matrix $A \in \mathbb{Z}^{m\times n}$ is an* EQ_q matrix *if the homogeneous system $Ax = 0$ has no non-trivial solutions in* $\{-q+1, \ldots, q-1\}^n$.

If $q = 2$, then we drop q from the notation and say that the matrix is an EQ matrix. For the EQ_q matrices, the optimal rate given by Siegel's Lemma is still $R = O(\log nW)$ and constant weight constructions exist. We give an extension of our construction to EQ_q matrices where $W = 1$ and asymptotic efficiency in rate is achieved up to a factor of q.

Theorem 11. *Suppose we are given an EQ_q matrix $A_0 \in \{-1, 0, 1\}^{m_0 \times n_0}$. At iteration k, we construct the following matrix A_k:*

$$\begin{bmatrix} A_{k-1} & A_{k-1} & A_{k-1} & \cdots & A_{k-1} & I_{m_{k-1}} \\ A_{k-1} & -A_{k-1} & 0 & \cdots & 0 & 0 \\ 0 & A_{k-1} & -A_{k-1} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -A_{k-1} & 0 \end{bmatrix} \tag{2.22}$$

A_k is an EQ_q matrix with $m_k = q^k m_0$, $n_k = q^k n_0(\frac{k}{q}\frac{m_0}{n_0} + 1)$ for any integer $k \geq 0$.

Given $Ax = z \in \mathbb{Z}^m$, we note that there is a linear time decoding algorithm to find $x \in \{0, 1\}^n$ uniquely, given in the Appendix A.

2.4 Maximum Distance Separable Extensions of EQ Matrices

Residue codes are treated as *Maximum Distance Separable* (MDS) codes (see Tay and Chang, 2015 for example) because one can extend the CRT matrix by adding more prime numbers to the matrix (without increasing n) so that the resulting integer code achieves the Singleton bound in Hamming distance. However, we do not say a CRT matrix is an *MDS matrix* as this refers to another concept.

Definition 10. *An integer matrix $A \in \mathbb{Z}^{m\times n}$ ($m \leq n$) is* MDS *if and only if no $m \times m$ submatrix B is singular.*

Definition 11. *An integer matrix $A \in \mathbb{Z}^{rm\times n}$ is* MDS for q-ary bijections with MDS rate r and EQ rate $R = n/m$ *if and only if every $m \times n$ submatrix B is an* EQ_q *matrix.*

Because Definition 11 considers solutions over a **restricted** alphabet, we denote such matrices as *$RMDS_q$*. Remarkably, as $q \to \infty$, both MDS definitions become the same. Similar to the EQ_q definition, we drop the q from the notation when $q = 2$. A CRT matrix is not MDS, however, it can be RMDS_q.

We can demonstrate the difference of both MDS definitions by the following matrix. This matrix is an RMDS matrix with EQ rate 2 and MDS rate $5/4$ because any 4×8 submatrix is an EQ matrix. This is in fact the same matrix in (2.13) with an additional row with entries $[2^{i-1}]_{13}$ for $i \in \{1, ..., 8\}$.

$$\begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\ 1 & 2 & 4 & 3 & 1 & 2 & 4 & 3 \\ 1 & 2 & 4 & 1 & 2 & 4 & 1 & 2 \\ 1 & 2 & 4 & 8 & 5 & 10 & 9 & 7 \\ 1 & 2 & 4 & 8 & 3 & 6 & 12 & 11 \end{bmatrix}_{5\times 8} \tag{2.23}$$

Here, the determinant of the 5×5 submatrix given by the first five columns is 0. Thus, this matrix is not an MDS matrix.

Similar to a CRT-based RMDS matrix, is there a way to extend the EQ matrix given in Section 2.3 without a large trade-off in the weight size? We first give an upper bound on the MDS rate r based on the alphabet size of an RMDS_q matrix.

Theorem 12. *An $RMDS_q$ matrix $A \in \mathbb{Z}^{rm\times n}$ with entries from the alphabet $\Gamma = \{\gamma_1, ..., \gamma_k\}$ satisfies $r \leq k^{k+1}$ given that $n > k$.*

Proof. The proof is based on a simple counting argument. We rearrange the rows of a matrix in blocks B_i for $i \in \{1, \cdots, k^{k+1}\}$ assuming $n > k$. Here, each block contains rows starting with a prefix of length $k+1$ in the lexicographical order of the indices, i.e.,

$$A = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_{k^{k+1}} \end{bmatrix} = \begin{bmatrix} \gamma_1 & \gamma_1 & \cdots & \gamma_1 & \cdots \\ \gamma_1 & \gamma_1 & \cdots & \gamma_2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ \gamma_k & \gamma_k & \cdots & \gamma_k & \cdots \end{bmatrix} \tag{2.24}$$

For instance, the block B_1 contains the rows starting with the prefix $\begin{bmatrix} \gamma_1 & \gamma_1 & \cdots & \gamma_1 \end{bmatrix}_{1 \times k+1}$. It is easy to see that there is a vector $x \in \{-1, 0, 1\}^n$ that will make at least one of the $B_i x = 0$ because it is guaranteed that the $(k+1)$-th column will be equal to one of the preceding elements by Pigeonhole Principle.

Since the matrix is RMDS$_q$, any m row selections should be an EQ matrix. Therefore, any B_i should not contain more than m rows. Again, by Pigeonhole Principle, $rm \leq k^{k+1} m$ and consequently, $r \leq k^{k+1}$. □

If the weights are constant size in n, then k is a constant. This implies that $n > k$ is typically satisfied and the MDS rate can only be a constant asymptotically if alphabet is constant size.

Corollary 12.1. *An RMDS$_q$ matrix $A \in \mathbb{Z}^{rm \times n}$ weight size $W = O(1)$ can at most achieve the MDS rate $r = O(1)$.*

In Section 2.3, we saw that CRT-based EQ matrix constructions are not optimal in terms of the weight size. For RMDS$_q$ matrices, we now show that the weight size can be reduced by a factor of n. The idea is to use the probabilistic method on the existence of RMDS$_q$ matrices.

Lemma 4. *Suppose that $a \in \{-W, \dots, W\}^n$ is uniformly distributed and $x_i \in \{-q+1, \dots, q-1\} \setminus \{0\}$ for $i \in \{1, \dots, n\}$ are fixed where q is a constant. Then, for some constant C,*

$$Pr(a^T x = 0) \leq \frac{C}{\sqrt{n} W} \tag{2.25}$$

Proof. We start with a statement of the Berry-Esseen Theorem.

Lemma 5 (Berry-Esseen Theorem)**.** *Let $X_1, \ldots, X_n$ be independent centered random variables with finite third moments $\mathbb{E}[|X_i^3|] = \rho_i$ and let $\sigma^2 = \sum_{i=1}^n \mathbb{E}[X_i^2]$. Then, for any $t > 0$,*

$$\left| Pr\Big(\sum_{i=1}^n X_i \leq t \Big) - \Phi(t) \right| \leq C\sigma^{-3} \sum_{i=1}^n \rho_i \tag{2.26}$$

where C is an absolute constant and $\Phi(t)$ is the cumulative distribution function of $\mathcal{N}(0, \sigma^2)$.

Since the density of a normal random variable is uniformly bounded by $1/\sqrt{2\pi\sigma^2}$, we obtain the following.

$$Pr\Big(\Big| \sum_{i=1}^n X_i \Big| \leq t \Big) \leq \frac{2t}{\sqrt{2\pi\sigma^2}} + 2C\sigma^{-3} \sum_{i=1}^n \rho_i \tag{2.27}$$

Let $a^{(n-1)}$ denote the $(a_1, \ldots, a_{n-1})$ and similarly, let $x^{(n-1)}$ denote $(x_1, \ldots, x_{n-1})$. By the Total Probability Theorem, we have

$$\begin{aligned}
Pr(a^T x = 0) = Pr&\left(\frac{{a^{(n-1)}}^T x^{(n-1)}}{|x_n|} \in \{-W, \ldots, W\} \right) \\
Pr&\left(a^T x = 0 \middle| \frac{{a^{(n-1)}}^T x^{(n-1)}}{|x_n|} \in \{-W, \ldots, W\} \right) \\
+ Pr&\left(\frac{{a^{(n-1)}}^T x^{(n-1)}}{|x_n|} \notin \{-W, \ldots, W\} \right) \\
Pr&\left(a^T x = 0 \middle| \frac{{a^{(n-1)}}^T x^{(n-1)}}{|x_n|} \notin \{-W, \ldots, W\} \right) \\
= Pr&\left(\frac{{a^{(n-1)}}^T x^{(n-1)}}{|x_n|} \in \{-W, \ldots, W\} \right) \frac{1}{2W+1}
\end{aligned} \tag{2.28}$$

where the last line follows from the fact that $Pr(a_n = k)$ for some $k \in \{-W, \ldots, W\}$ is $\frac{1}{2W+1}$. The second conditional probability term is 0 because $Pr(a_n = k)$ for any $k \notin \{-W, \ldots, W\}$ is 0. We will apply Berry-Esseen Theorem to find an upper bound on the first term.

We note that $\mathbb{E}[a_i^2 x_i^2] = x_i^2 \frac{(2W+1)^2 - 1}{12}$ and $\mathbb{E}[|a_i^3 x_i^3|] = |x_i|^3 \frac{2}{2W+1} \left(\frac{W(W+1)}{2} \right)^2$. Then,

$$Pr\left(\left|\frac{a^{(n-1)^T}x^{(n-1)}}{|x_n|}\right| \leq W\right) \leq \frac{2W|x_n|}{\sqrt{2\pi}\sqrt{\frac{(2W+1)^2-1}{12}\sum_{i=1}^{n-1} x_i^2}} + C\frac{\frac{2}{2W+1}\left(\frac{W(W+1)}{2}\right)^2\sum_{i=1}^{n-1}|x_i|^3}{\left(\frac{(2W+1)^2-1}{12}\right)^{\frac{3}{2}}\left(\sum_{i=1}^{n-1} x_i^2\right)^{\frac{3}{2}}} \tag{2.29}$$

One can check that the whole RHS is in the form of $\frac{C'}{\sqrt{n-1}}$ for a constant C' as we assume that q is a constant. We can bound $|x_i|$s in the numerator by $q-1$ and $|x_i|$s in the denominator by 1. The order of W terms are the same and we can use a limit argument to bound them as well. Therefore, for some $C'' > 0$,

$$Pr(a^T x = 0) \leq \frac{C'}{\sqrt{n-1}}\frac{1}{2W+1} \leq \frac{C''}{\sqrt{n}W} \tag{2.30}$$

□

Lemma 4 is related to the Littlewood-Offord problem and anti-concentration inequalities are typically used in this framework (Halász, 1977; Rudelson and Vershynin, 2008). We specifically use Berry-Esseen Theorem. Building on this lemma and the union bound, we have the following theorem.

Theorem 13. *An RMDS$_q$ matrix $M \in \mathbb{Z}^{rm\times n}$ with entries in $\{-W,\dots,W\}$ exists if $m = \Omega(n/\log n)$ and $W = O(r)$.*

Proof. Let A be any $m \times n$ submatrix of M. Then, by the union bound (as done in Karingula and Lovett, 2021), we have

$$\begin{aligned} Pr(\text{M is not RMDS}_q) = P &\leq \binom{rm}{m} Pr(\text{A is not EQ}_q) \\ &\leq r^m e^m Pr(\text{A is not EQ}_q) \end{aligned} \tag{2.31}$$

We again use the union bound to sum over all $x \in \{-q+1,\dots,q-1\}^n \setminus \{0\}$ the event that $Ax = 0$ by counting non-zero entries in x by k. Also, notice the independence of the events that $a_i^T x = 0$ for $i \in \{1,\dots,m\}$. Let $x^{(k)}$ denote an arbitrary $x \in \{-q+1,\dots,q-1\}^n \setminus \{0\}$ with k non-zero entries. Then,

$$Pr(\text{A is not EQ}_q) \leq \sum_{x \in \{-q+1,\ldots,q-1\}^n \setminus \{0\}} \prod_{i=1}^{m} Pr(a_i^T x = 0) \tag{2.32}$$

$$\leq \sum_{k=1}^{n} \binom{n}{k} (2(q-1))^k Pr(a^T x^{(k)} = 0)^m \tag{2.33}$$

We can use Lemma 4 to bound the $Pr(a^T x^{(k)} = 0)$ term. Therefore,

$$P \leq r^m e^m \sum_{k=1}^{n} \binom{n}{k} (2(q-1))^k \left(\frac{C}{\sqrt{k}W}\right)^m \tag{2.34}$$

$$= \sum_{k=1}^{T-1} \binom{n}{k} (2(q-1))^k \left(\frac{rC_1}{\sqrt{k}W}\right)^m$$

$$+ \sum_{k=T}^{n} \binom{n}{k} (2(q-1))^k \left(\frac{rC_1}{\sqrt{k}W}\right)^m \tag{2.35}$$

for some $C_1, T \in \mathbb{Z}$. We bound the first summation by using $\binom{n}{k}(2(q-1))^k \leq (n(q-1))^k$ and choosing $k = 1$ for the probability term. This gives a geometric sum from $k = 1$ to $k = T - 1$.

$$\sum_{k=1}^{T-1} (n(q-1))^k \left(\frac{rC_1}{W}\right)^m = \left(\frac{rC_1}{W}\right)^m \left(\frac{(n(q-1))^T - 1}{n(q-1) - 1}\right) \tag{2.36}$$

For the second term, we take the highest value term $\frac{rC_1}{\sqrt{T}W}$ and $\sum_{k=T}^{n} \binom{n}{k} 2(q-1)^k \leq \sum_{k=0}^{n} \binom{n}{k} 2(q-1)^k = (2(q-1)+1)^n$. Hence,

$$P \leq \left(\frac{rC_1}{W}\right)^m C_2 (n(q-1))^T + (2(q-1)+1)^n \left(\frac{rC_1}{\sqrt{T}W}\right)^m \tag{2.37}$$

$$\leq 2^{C_3 T \log n(q-1) - m \log W/rC_1}$$

$$+ 2^{n \log(2(q-1)+1) - m \log(\sqrt{T}W/rC_1)} \tag{2.38}$$

We take $T = O(n^c)$ for some $0 < c < 1$. It is easy to see that if $m = \Omega(n/\log n)$ and $W = O(r)$, both terms vanish as n goes to the infinity.

□

We remark that the proof technique for Theorem 13 is not powerful enough to obtain non-trivial bounds on W when $m = 1$ to attack Erdős' conjecture on the DSS weight sets.

The CRT gives an explicit way to construct $\text{RMDS}_q \in \mathbb{Z}^{rm \times n}$ matrices. We obtain $p_{rm} = O(rm \log rm) = O(rn)$ given that $r = O(n^c)$ for some $c > 0$ and $m = O(n/\log n)$ by the PNT. Therefore, we have a factor of $O(n)$ weight size reduction in Theorem 13 where we show the existence of such RMDS_q matrices with $W = O(r)$. However, modular arithmetical properties of the CRT do not reflect to RMDS_q matrices. Therefore, an RMDS_q matrix cannot replace a CRT matrix in general (see Appendix B).

2.5 Lower Bounds for the DOM functions

By using Siegel's Lemma, we can obtain a lower bound for DOM functions as well. We first give the bound for the exact DOM functions and then generalize it to linear DOM functions.

Theorem 14. *To compute an $(n+1)$-ary L input exact DOMINATION function $f(X')$, $\Omega(L/\log LW)$ many $\mathcal{S}$-ary perceptrons with weight constraint W are needed.*

Proof. We proceed similar to the proof of Lemma 2. Since we consider binary inputs for the perceptrons, each $\{0,1\}$-sum of rows of the weight matrix is bounded by $C = LW$. Then, by Siegel's Lemma in the modified form (see Lemma 3), if

$$||x||_\infty \leq (LW)^{\frac{m}{L-m}} \leq 2^{1-\epsilon} \tag{2.39}$$

$$m \leq \frac{L(1-\epsilon)}{\log LW + (1-\epsilon)} \tag{2.40}$$

for any $0 < \epsilon < 1$, then we have a non-trivial solution to $Ax^* = 0$ for $x \in \{-1,0,1\}^L$ where $w_b^T x^* \neq 0$. This leads to a contradiction because for the trivial solution $x = 0$, $Ax = Ax^* = 0$. Therefore, taking $\epsilon \to 0$, we see that $m = \Omega(L/\log LW)$. □

Theorem 14 actually applies to linear DOM function as well because if $Ax^* = 0$ for some $x^* \in \{-1,0,1\}^L \setminus \{0\}$, then so is its negation $-x^*$ a non-trivial solution. Hence, without loss of generality, there exists an $x^* \in \{-1,0,1\}^L$ such that $Ax^* = 0$ but $w_b^T x^* < 0$.

Corollary 14.1. *To compute an $\mathcal{S}$-ary linear DOMINATION function $f(X')$ with L inputs, $\Omega(L/\log LW)$ many $\mathcal{S}$-ary perceptrons with weight constraint W are needed.*

We conjecture that the lower bound $\Omega(L/\log LW)$ applies to the vast majority of the arbitrary exact threshold functions if arbitrary perceptrons with weight constraint W are used in the construction. Furthermore, we believe that this is not tight for depth-2 exact threshold circuits with small weights where the best upper bound on the circuit size is $O(nL/\log L)$.

2.6 Application of the EQ and RMDS$_q$ Matrices to DOM functions

Consider an arbitrary exact DOM function $\mathbb{1}\{w_b^T X' = 0\}$ where $\mathcal{S} = 1$. Then, the only solution to this equation is $X' = 0$ similar to EQ function. Therefore, an EQ matrix and an AND gate can be used to construct a circuit with constant weights similar to Figure 2.5. Note that we do not consider bias term sizes in our weight size analysis.

Theorem 15. *If $\mathcal{S} = 1$, there exists a depth-2 exact threshold circuit which computes an $\mathcal{S}$-ary L input exact DOMINATION function and its size is bounded by $O(L/\log L)$ with weights at most $O(1)$.*

For example, to construct threshold circuits computing the EQ, we select the weights for each exact threshold gate as the rows of the EQ matrix. Then, we connect the outputs of the first layer to the top gate, which just computes the m-input AND function (i.e. $\mathbb{1}\{Z_1 + \ldots + Z_m = m\}$ for $Z_i \in \{0, 1\}$). In Figure 2.5, we give an example of an EQ construction.

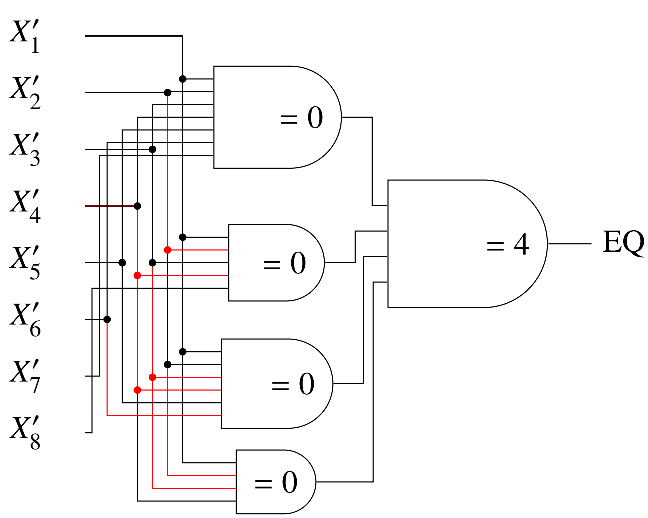


Figure 2.5: An example of EQ$_{16}(X, Y)$ constructions with 8 $X_i' = X_i - Y_i$ inputs and 5 exact threshold gates (including the top gate). The black (or red) edges correspond to the edges with weight 1 (or -1).

In general, for the exact DOM gate $\mathbb{1}\{w_b^T x' = 0\}$, there are many different small norm solutions such as $x' = (-2, 1, 0, ..., 0)$ and we cannot apply EQ matrices. Therefore, we do not know if $\mathbb{1}\{w_b^T x' = 0\}$ admits constant weight perceptron constructions at $O(L/\log L)$ complexity. The answer is affirmative if weights are polynomially large in n considering CRT-based constructions where modulo gates (i.e. $g_i(X) = \mathbb{1}\{\sum_{i=1}^n w_i X_i \equiv 0 \pmod{p_i}\}$) are used instead of exact threshold gates. We conjecture that the reduction from polynomial weight sizes to constant weights is true in general.

It is surprising that when $\mathcal{S} = 1$, linear DOM function require more involved techniques and they have higher size complexity upper bounds. Here, the RMDS$_q$ matrices play an important role in small weight constructions of the linear DOM functions. Since older techniques relied on the CRT and our work proved the existence of smaller weight RMDS$_q$ matrices, we can reduce the weights in the small weight constructions when $\mathcal{S} = 1$. We roughly follow the previous works which gives $O(n^3/\log n)$ size complexity and $O(n^2)$ weights for COMP and CARRY (i.e. $\mathbb{1}\{X + Y \geq 2^n\}$ where X and Y are n-bit integers) (Amano and Maruoka, 2005).

Let $f(X)$ be a linear threshold function with $F(X) = w^T X$ and the domination form $w_b^T x' = 0$ with $\mathcal{S} = 1$. COMP, CARRY, and OMB functions are examples of such linear threshold functions (here, $\text{CARRY}(X,Y) = \mathbb{1}\{X + Y \geq 2^n\}$ needs to be modified either to $\mathbb{1}\{2X + 2Y \geq 2^{n+1} - 1\}$ or to its complement $\mathbb{1}\{X + Y \leq 2^n - 1\}$ to make sure that $\mathcal{S} = 1$). First, let us define $F^{(l)}(X') = \sum_{i=l+1}^{L} 2^{i-l-1} x_i'$ where $F^{(0)}(X,Y) = F(X,Y)$. We also denote by $X'^{(l)}$ the most significant $(n-l)$-tuple of an n-tuple vector X'. We have the following observation.

Lemma 6. *Let $F^{(l)}(X') = \sum_{i=l+1}^{L} 2^{i-l-1} X_i'$ and $F^{(0)}(X') = F(X')$ for $X' \in \{-1,0,1\}^L$. Then,*

$$F(X) > 0 \Leftrightarrow \exists l : F^{(l)}(X') = \quad 1 \tag{2.41}$$

$$F(X) < 0 \Leftrightarrow \exists l : F^{(l)}(X') = -1 \tag{2.42}$$

Proof. It is easy to see that if $X' = (X_1', \dots, X_n') = (\times, \dots, \times, 1, 0, \dots, 0)$ where the vector has a number of significant 0s and $\times$s denote any of $\{-1, 0, 1\}$, we see that $F(X) > 0$ (this is called the *domination property*). Similarly, for $F(X) < 0$, the vector X' should have the form $(0, 0, \dots, 0, -1, \times, \dots, \times)$ and $F(X') = 0$ if and only if $X' = (0, \dots, 0)$. The converse holds similarly. $\square$

In contrast, if we do not use the domination form of threshold functions, Lemma 6 will require $F(X) > 0 \Leftrightarrow \exists l : F^{(l)}(X) \in \{2,3\}$ (see Lemma 1 in Amano and Maruoka, 2005). In this case, the size complexity blows up to $O(n^4/\log n)$ due to modulo checks similar to the constructions in Section 2.2. To reduce the complexity, the use of domination form is implicit in (Amano and Maruoka, 2005) (see Section 3 in Amano and Maruoka, 2005).

An important remark is that Lemma 6 helps us compute $F(X) > 0$ or $F(X) < 0$. To include $F(X) = 0$, we can use $F(X) < 0$ and negate the function (this is required for CARRY and COMP but not for OMB). We will search for the $(L-l)$-tuple vectors $(X-Y)^{(l)} = (0, ..., 0, -1)$ for all $l \in \{0, ..., L-1\}$ to compute $F(X) < 0$. We claim that if we have an RMDS_q matrix $A \in \mathbb{Z}^{rm \times L}$, we can detect such vectors by solving $A^{(l)}(X-Y)^{(l)} = -a_{L-l}$ where $A^{(l)}$ is a truncated version of A with the first $L-l$ columns and a_{L-l} is the $(L-l)$-th column. Specifically, we obtain the following:

$$F(X) < 0 \Rightarrow \sum_{l=0}^{L-1} \mathbb{1}\{A^{(l)} X'^{(l)} = -a_{L-l}\} \geq rm \tag{2.43}$$

$$F(X) \geq 0 \Rightarrow \sum_{l=0}^{L-1} \mathbb{1}\{A^{(l)} X'^{(l)} = -a_{L-l}\} < L(m-1) \tag{2.44}$$

Here, the indicator function works row-wise, i.e., we have the output vector $z \in \{0,1\}^{rmL}$ such that $z_{rmL+i} = \mathbb{1}\{(A^{(l)}(X-Y)^{(l)})_i = -(a_{L-l})_i\}$ for $i \in \{1, \dots, rm\}$ and $l \in \{0, \dots, L-1\}$. We use an RMDS_3 matrix in the construction to map $\{-1,0,1\}^{L-l}$ vectors bijectively to integer vectors with large Hamming distance. Note that each exact threshold function can be replaced by two linear threshold functions and a summation layer (i.e. $\mathbb{1}\{F(X) = 0\} = \mathbb{1}\{F(X) \geq 0\} + \mathbb{1}\{-F(X) \geq 0\} - 1$) which can be absorbed to the top gate (see Figure 2.6). This increases the circuit size only by a factor of two.

If $F(X) < 0$, z_{rml+i} should be 1 for all $i \in \{1, \dots, rm\}$ for some l by Lemma 6. For $F(X) \geq 0$ and any l, the maximum number of 1s that can appear in z_{rml+i} is upper bounded by $m-1$ because A is RMDS_3. Therefore, the maximum number of 1s that can appear in z is upper bounded by $L(m-1)$. A sketch of the construction is given in Figure 2.7.

In order to make both cases separable, we choose $r = L$. At the second layer, the top gate is a MAJ gate ($\mathbb{1}\{\sum_{i=1}^{rmL} z_i < L(m-1)\} = \mathbb{1}\{\sum_{i=1}^{rmL} -z_i \geq -L(m-1)+1\}$). By

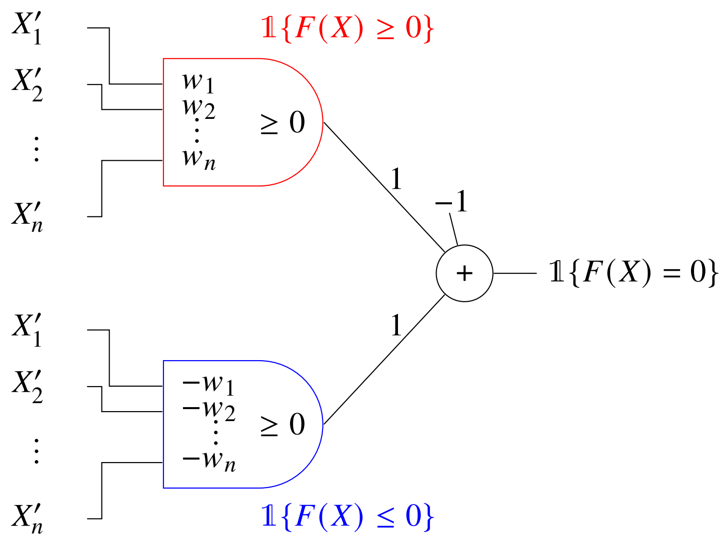


Figure 2.6: A construction of an arbitrary exact threshold function ($\mathbb{1}\{F(X) = 0\}$) using two linear threshold functions ($\mathbb{1}\{F(X) \geq 0\}$ and $\mathbb{1}\{F(X) \leq 0\}$) and a summation node. This summation node can be removed if its output is connected to another gate due to linearity.

Theorem 13, there exists an RMDS_3 matrix with $m = O(L/\log L)$ and $W = O(L)$. Thus, there are $rmL + 1 = O(L^3/\log L)$ many gates in the circuit, which is the best known result. The same size complexity can be achieved by a CRT-based approach with $W = O(p_{rm}) = O(L^2)$ by the PNT (Amano and Maruoka, 2005). We give a sketch of the construction in Figure 2.7.

2.7 Conclusion

We introduced a new way of computation for threshold functions using DOM gates. The idea to use powers of two as the weights and a larger alphabet in the input variables is shown to be useful to achieve more efficient constructions in weights and circuits size. The results are obtained by the application of known mathematical techniques to circuit complexity theory in a novel way: Siegel's Lemma (and its variations) is used to derive lower bounds and Berry-Esseen Theorem is applied to optimize the weight size for a previously known construction. Additionally, by extending the Sylvester-type Hadamard matrices, optimal rate EQ matrices are constructed, settling an open problem on the gap between the upper and lower bounds on the size of the circuits computing the EQ functions. It is not known if a similar extension is possible for any Hadamard matrix.

Based on an important property of CRT, a generalization of MDS matrices is investigated, namely, RMDS_q matrices. Although CRT matrices are constructive examples of RMDS_q matrices, it is shown that the weight sizes are not optimal by an existence argument. It is an interesting research direction to construct such a matrix explicitly. Moreover, to compute a threshold function using a small weight depth-2 threshold circuit, modular arithmetic property of the CRT matrices becomes significant. To construct such circuits, it is not clear if such a property is necessary or the CRT matrix has optimal size weights.

In this chapter, the constructive results and lower bounds are obtained for threshold functions whose domination form has sparsity $\mathcal{S} = 1$. For specific threshold functions, the results are given in Table 1.1. For the circuits computing EQ, provably optimal rate is achieved using constant size weights by an explicit construction. For circuits computing either COMP, CARRY, and OMB functions, the weight sizes are reduced by a factor of n without an asymptotic increase in the circuit size. The natural open problem is whether similar algebraic constructions or lower bounds can be found for any threshold function where $\mathcal{S} > 1$ (even $\mathcal{S} = 2$ seems to be a challenge) so that these ideas can be developed into a more general weight quantization technique for neural networks.

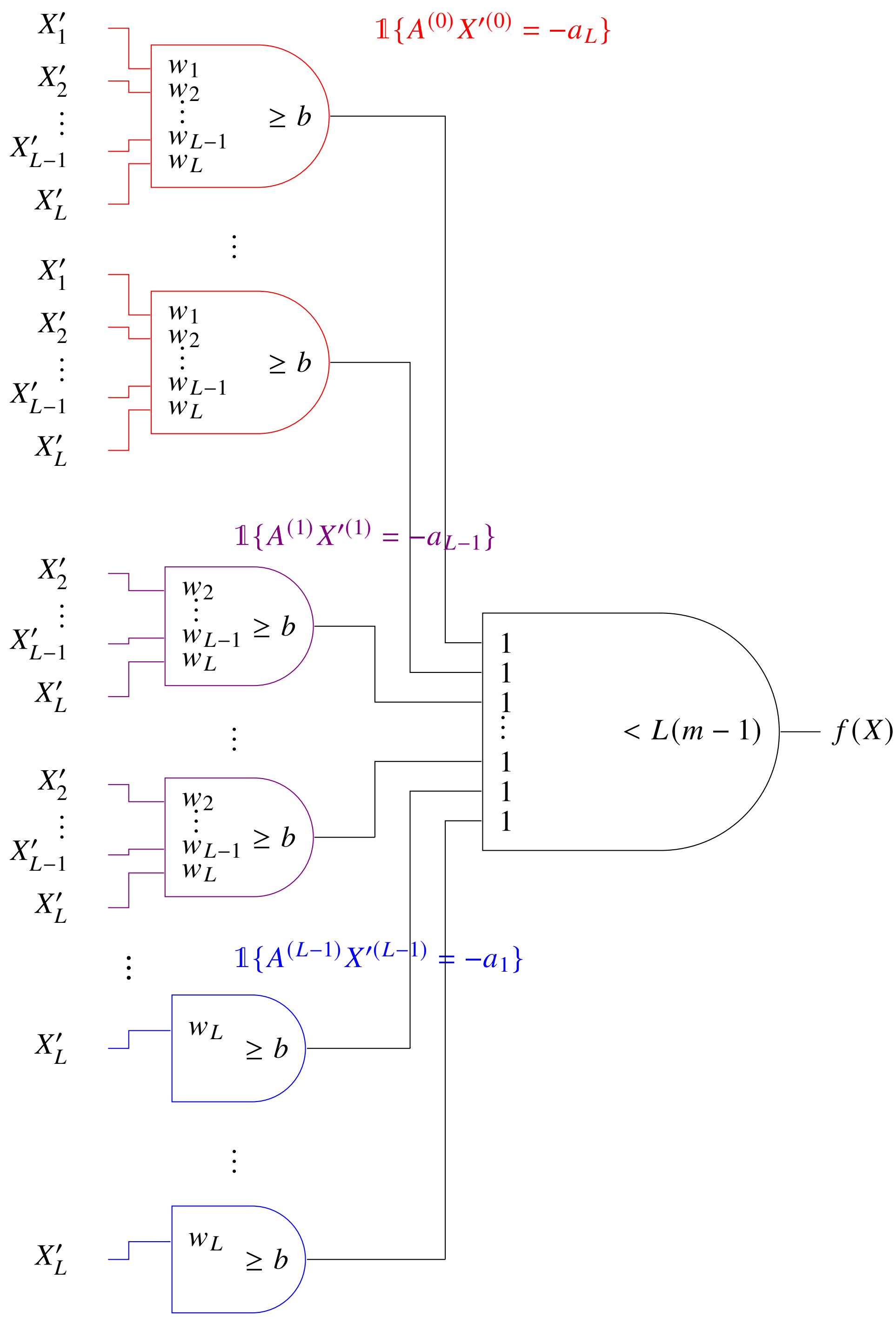


Figure 2.7: A sketch of the small weight LT circuit construction of the linear threshold function $f(X)$ with domination form $\mathbb{1}\{w_b^T X' \geq 0\}$ and $\mathcal{S} = 1$. Each color specifies an l value in the construction. If $F(X) < 0$, all the rm many gates in at least one of the colors will give all 1s at the output. Otherwise, all the rm many gates in a color will give at most $(m-1)$ many 1s at the output.

Chapter 3

ON THE COMPLEXITY OF THE NEAREST NEIGHBOR REPRESENTATIONS

First of all, we informally summarize the preliminaries for the NN representation framework. NN representations of Boolean functions consist of anchors in real space and they classify the vectors on the binary vectors into two groups. The NN complexity of a Boolean function f is the smallest number of anchors required to represent f without any error. The resolution of an NN representation is the maximum number of required bits to represent an entry of all anchors.

NN complexity gives us a quantitative way to classify Boolean functions. Let us start from the most simple case: A constant Boolean function $f(X) = 0$ or $f(X) = 1$ can be represented by a single anchor, which can be any real vector. Therefore, we say that these functions have trivial NN representations. All functions such that $NN(f) = 2$ are linear threshold functions (P. Hajnal, Z. Liu, and Turán, 2022). The class of functions $NN(f) = 3$ implies a convex polytope, which is treated in Section 3.4. Exact threshold functions are in this class. The characterization for $NN(f) \geq 4$ is more difficult, however, some important classes of Boolean functions, like symmetric Boolean functions, could require this NN complexity.

In the introduction, the resolution-size trade-off is illustrated by comparing NN and BNN complexities (see Figure 1.8). Basically, when the resolution of the anchor entries decreases, the number of anchors increases. Although $NN(f) = 2$ for linear threshold functions, the resolution of the anchors is, however, $O(n \log n)$ since an arbitrary linear threshold function requires exponentially large weights in n(Alon and Vũ, 1997; Håstad, 1994; Saburo Muroga, 1971). We will show this rigorously in Theorem 17. In the previous chapter, we investigated how a threshold function with "large" weights can be computed using "small" weight depth-2 threshold circuits to give some ground on our hypothesis for the brain. In this chapter, our main goal is to show a similar result for the NN representations of threshold functions and we are interested in an answer for the following conjecture.

Conjecture 1. *Let $f(X)$ be any n-input linear threshold function. Then, there is an NN representation of $f(X)$ with $O(poly(n))$ number of anchors and $O(\log n)$ resolution.*

In brief, the conjecture is true at least for some threshold functions. But, the answer is not clear in general and there might be an eventual exponential blow-up in the size (recall that any Boolean function has a BNN representation with at most 2^n anchors and this is "infeasible" for our hypothesis). We will see how the results of the previous chapter could be applied to the framework of NN representations in Section 3.7.

To answer the conjecture in this associative computation model, we look for an analogous result for small weight depth-2 threshold circuit constructions of threshold functions and need to build the bridge between NN representations and threshold circuits. The following is well-known where $\widehat{\text{LT}}$ (or $\widehat{\text{ELT}}$) denotes the linear (or exact) threshold functions with polynomially large weights in n (Amano and Maruoka, 2005; Goldmann and Karpinski, 1993; Hansen and V. V. Podolskii, 2010; Hofmeister, 1996):

$$\text{LT} \subsetneq \widehat{\text{LT}} \circ \widehat{\text{LT}} \tag{3.1}$$

$$\text{ELT} \subsetneq \widehat{\text{ELT}} \circ \widehat{\text{ELT}} \tag{3.2}$$

Essentially, the depth-2 small weight threshold circuit constructions of arbitrary exact threshold functions is in the form of $\text{SYM} \circ \widehat{\text{ELT}}$. The techniques are reviewed in Chapter 2 where CRT and PNT are used. The known results for the constructions for $\text{LT} \subseteq \widehat{\text{LT}} \circ \widehat{\text{LT}}$ rely on the transformations $\text{ELT} \subseteq \text{SYM} \circ \widehat{\text{ELT}}$ and the associated circuit size and weight size complexities (see Amano and Maruoka, 2005 for the full details). Hence, we extend the scope of our analysis to $\text{SYM} \circ \text{ELT}$ and the related class $\text{SYM} \circ \text{LT}$ where the top gate is an arbitrary symmetric Boolean function.

In addition, the treatment of these depth-2 Boolean circuits is a significant step to find a correspondence between neural networks and NN representations, which are both inspired by the brain architecture. It is easy to transform an NN representation into a neural network: An m-anchor NN representation of $f(X)$ can be used to compute it by using $O(m^2)$ gates using a depth-3 linear threshold circuit (P. Hajnal, Z. Liu, and Turán, 2022; Murphy, 1990). We make this argument explicit in Section 3.3. This is in fact a circuit in the class of $\text{OR} \circ \text{AND} \circ \text{LT}$. Conversely, could we find an NN representation for a neural network? In a traditional sense, a neural network is a composition of linear threshold functions which has $NN(f) = 2$. Thus, to answer this question, it is essential to understand how function composition plays a role in the constructions of NN representations.

In this thesis, we focus on the NN representations of LT, ELT, and SYM functions and their compositions to answer Conjecture 1. The flow of this chapter is as follows.

- NN Representations of Threshold Functions.
- NN Representations of Symmetric Boolean Functions.
- From NN Representations to Threshold Circuits.
- NN Representations of Convex Polytopes (AND $\circ$ LT).
- NN Representations of Depth-2 Threshold Circuits with Symmetric Top Gate (SYM $\circ$ ELT and SYM $\circ$ LT).
- NN Representations of Linear and Exact Decision Lists.
- Towards the Low Resolution NN Representations of Threshold Functions.

Before starting our analysis, we give important examples of Boolean functions in the classes of depth-2 circuits for further motivation.

We start with the symmetric Boolean function $\text{PARITY}(X) = \bigoplus_{i=1}^{n} x_i$, which had a lot of attention from the mathematics and information theory for many decades (Håstad, 1986; Kautz, 1961; Paturi and Saks, 1990). n-input PARITY has a nice recursive structure $\text{PARITY}_n = \text{PARITY}_{n/m} \circ \text{PARITY}_m \in \text{SYM} \circ \text{SYM}$. Specifically, $\text{PARITY}_n = \text{PARITY}_{n/2} \circ \text{XOR} \in \text{SYM} \circ \text{ELT}$. We present results for both constructions, which are essentially equivalent.

Another significant function in the Circuit Complexity Theory is INNER-PRODUCT-MOD2 (denoted by IP2) function.

$$\text{IP2}_{2n}(X,Y) = X^T Y \pmod 2 = \bigoplus_{i=1}^{n} x_i \wedge y_i \tag{3.3}$$

IP2 is a very intriguing function since it is still an open problem if $\text{IP2} \in \text{LT} \circ \text{LT}$ (here, the membership is defined for polynomially large circuits). Some partial results are known in that $\text{IP2} \notin \text{LT} \circ \widehat{\text{LT}}$, $\text{IP2} \notin \widehat{\text{LT}} \circ \text{LT}$ by a discriminator lemma and unbounded Communication Complexity arguments (Forster, 2002; A. Hajnal et al., 1993). However, it has $O(n)$ size depth-3 threshold circuit constructions by simply constructing $\text{IP2}_{2n} = \text{PARITY}_n \circ \text{AND}_2$ (PARITY can be computed by a depth-2 threshold circuit). For the NN representations with unbounded resolution, the following lower bound on the NN complexity is shown.

Theorem 16 (P. Hajnal, Z. Liu, and Turán, 2022)**.** *For* $2n$*-input* IP2 *function,* $\mathrm{NN}(\mathrm{IP2}_{2n}) \geq 2^{n/2}$.

We give a construction of $\mathrm{IP2}_{2n}$ with 2^n many anchors and $O(1)$ resolution using $\mathrm{IP2}_{2n} = \mathrm{PARITY}_n \circ \mathrm{AND}_2 \in \mathrm{SYM} \circ \mathrm{LT}$. The upper bound is far from optimal (by a square root of a factor) but the upper bound could be optimal for constant resolution. We also note the recursive relationship $\mathrm{IP2}_{2n} = \mathrm{PARITY}_{n/k} \circ \mathrm{IP2}_{2k}$, in which n/k is an integer, but we provide no construction for it. This recursive structure of IP2 is exploited in (Amano, 2020) to simplify the known upper bounds and this could inspire future progress to find alternative NN representations for the IP2.

A recent striking result is that when one considers the k-NN representations with $k = \Omega(n)$, $kNN(IP2_{2n}) = O(n)$ (DiCicco, V. Podolskii, and Reichman, 2024). This in fact asymptotically matches a lower bound for the k-NN complexity of IP2 using LDTs in (P. Hajnal, Z. Liu, and Turán, 2022) and it is the first example of a Boolean function where k-NN and NN complexity have a dramatically different complexity. More generally, $\mathrm{SYM}_m \circ \mathrm{AND}_n$ circuits have k-NN representations with linear size in m. It is also shown that a generalized class of k-NN representations contains the class EDL (DiCicco, V. Podolskii, and Reichman, 2024).

The function $\mathrm{OR}_n \circ \mathrm{EQ}_{2n}$ is treated in (Chattopadhyay, Mahajan, et al., 2019) to show that for any LDL, the depth of the list must be $2^{\Omega(n)}$ (note that the number of inputs is $2n^2$) using a result by (Impagliazzo and R. Williams, 2010). This function was useful to separate EDL with LDL and a candidate to attack Conjecture 5 where we relate the exact cover complexity and NN complexity by $NN(f) \leq cEC(f)$ for some constant $c > 0$ for any Boolean function f (see Appendix C for definitions and discussion). Since $\mathrm{OR}_n \circ \mathrm{EQ}_{2n} \in \mathrm{SYM} \circ \mathrm{ELT}$, we obtain an NN representation with exponentially large number of anchors in n but it is not known if this is tight.

In the Circuit Complexity Theory, a recent result was obtained for $\mathrm{OMB}_m \circ \mathrm{EQ}_{2n} \in \mathrm{DOM} \circ \mathrm{ELT}$, which is an exact decision list, showing that it is the first explicit example of Boolean functions where any polynomially large size depth-2 linear threshold circuit computing it requires “large” weights (Chattopadhyay and Mande, 2018). More precisely, $\mathrm{OMB}_m \circ \mathrm{EQ}_{2n} \in \mathrm{LT} \circ \mathrm{LT}$ but not $\mathrm{LT} \circ \widehat{\mathrm{LT}}$ or $\widehat{\mathrm{LT}} \circ \mathrm{LT}$ when one considers polynomially large circuit sizes in n. The idea is to use the *sign-rank* method from the Communication Complexity Theory. We give an NN representation with exponentially large number of anchors for this function.

We finally make progress on the main question: Could we transform a depth-2 threshold circuit with small weights into a NN representation with small size and logarithmic resolution? For the EQ_{2n}, we establish that the answer is true by using the AND $\circ$ ELT construction. Here, we use our results for exact DOM functions with $\mathcal{S} = 1$ (see Chapter 2) and prove that there are $2n + 1$ size $O(1)$ resolution NN representations for the EQ function. The result essentially works for any exact threshold function whose domination form have sparsity parameter $\mathcal{S} = 1$. Moreover, one can reduce the size to $O(n/\log n)$ with $O(\log n)$ resolution. For linear threshold functions such as $COMP_{2n}$ and OMB_n, we obtain a similar result. For these functions, there are $O(n)$ size constructions with $O(\log n)$ resolution. It is not clear if the resolution can be reduced to $O(1)$ without a large blow-up in the number of anchors. The general answer to Conjecture 1 for arbitrary threshold functions is still unclear.

3.1 NN Representations of Threshold Functions

We start with *linear threshold functions*, which has the smallest NN complexity out of all Boolean functions except the constant Boolean functions (i.e. $f(X) = 0$ or $f(X) = 1$ where a single anchor suffices). Naturally, we are interested in the NN representations of non-constant linear threshold functions and we ignore them from now on in this context.

For linear threshold functions, one can easily see that $NN(f) = 2$. Firstly, one can observe that given any single positive-negative anchor pair, one can find the optimal separating hyperplane by computing the middle point of anchors and the normal vector. Conversely, we prove that given an n-input linear threshold function with weights $w \in \mathbb{Z}^n$, there is a 2-anchor NN construction with resolution $O(RES(w))$. The resolution of the bias term itself does not matter. In general, for linear threshold functions, the resolution of the weights are upper bounded by $O(n \log n)$ (Alon and Vũ, 1997; Håstad, 1994; Saburo Muroga, 1971). Note that we do not care the equality case $w^T X = b$ for linear threshold functions since without loss of generality, we can eliminate them by obtaining $\mathbb{1}\{w^T X \geq b\} = \mathbb{1}\{w^T X \geq b - 0.5\}$.

Theorem 17. *Let $f(X)$ be an n-input non-constant linear threshold function with weight vector $w \in \mathbb{Z}^n$ and the bias term $b \in \mathbb{Z}$. Then, there is a 2-anchor NN representation of $f(X)$ with resolution $O(RES(w))$. In general, the resolution is $O(n \log n)$.*

Proof. Geometrically, any 2-anchor NN representation construction of linear threshold functions can be written in the following form for a real number $c > 0$ and an

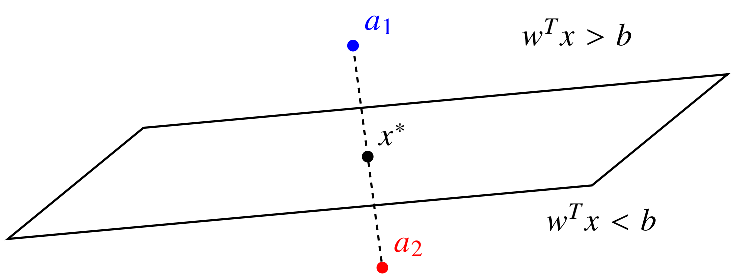


Figure 3.1: The NN Representation of a Linear Threshold Function $\mathbb{1}\{w^T X \geq b\}$ and its 2-anchor NN Representation. x^* can be any point on the hyperplane.

arbitrary $x^* \in \mathbb{R}^n$ such that $w^T x^* = b'$ where $b' = b - 0.5$.

$$a_1 = x^* + cw \tag{3.4}$$

$$a_2 = x^* - cw \tag{3.5}$$

This can be seen algebraically as well. $d(a_1, X)^2 - d(a_2, X)^2 = -4c(w^T X - b')$ is negative if and only if $w^T X > b'$, which corresponds to $f(X) = 1$.

We perturb b to b' to consider the points on the hyperplane itself. We first claim that there exists x^* such that $RES(x^*) \leq RES(w)$.

Since $f(X)$ is not constant, there is always a pair of binary vectors X' and X'' such that $w^T X' < b' < w^T X''$ where $X'_i = X''_i$ for all $i \in \{1, \dots, n\}$ except for a unique $i = k \in \mathbb{Z}$. Then, we construct

$$x_i^* = X'_i \text{ for } i \neq k \tag{3.6}$$

$$x_k^* = \frac{b' - w^T X'}{w_k} \tag{3.7}$$

Clearly, x_i^*s are binary except $i = k$ where $|b' - w^T X'| < w_k$ because $w^T X' < b' < w^T X''$ and therefore, $RES(x_k^*) = \lceil \log_2 w_k + 1 \rceil$. In conclusion, $RES(x^*) = \lceil \log_2 w_k + 1 \rceil \leq RES(w)$.

Picking $c = 1$, we see that $RES(A) = O(RES(w))$. Moreover, since $w_i = 2^{O(n \log n)}$ for $i \in \{1, \dots, n\}$ in general (Alon and Vũ, 1997; Håstad, 1994; Saburo Muroga, 1971), we conclude that $RES(A) = O(n \log n)$. □

We define an n-input *symmetric linear threshold function* as a symmetric Boolean function such that $f(X) = \mathbb{1}\{|X| \geq b\}$, namely, $w_i = 1$ for all i. Interestingly, depending on the value of b, these functions can have different orders of BNN complexities.

Theorem 18 (P. Hajnal, Z. Liu, and Turán, 2022)**.** *Let f be an n-input symmetric linear threshold function with threshold b. Then,*

- *If $b = n/2$ and n is odd, $BNN(f) = 2$.*
- *If $b = n/2$ and n is even, $BNN(f) = n/2 + 2$. (The lower bound is proven in DiCicco, V. Podolskii, and Reichman, 2024)*
- *If $b = \lfloor n/3 \rfloor$, $BNN(f) = 2^{\Omega(n)}$.*

If all the anchors are on the Boolean hypercube, then the NN complexity can be as large as $2^{\Omega(n)}$. However, interestingly, we show that for constant resolution NN representations, using 2 anchors for symmetric Boolean functions suffices using Theorem 17. This demonstrates that a slight change in the resolution can simplify the NN complexity.

Corollary 18.1. *Let $f(X)$ be an n-input symmetric linear threshold function with threshold b. Then, there is a 2-anchor NN representation of $f(X)$ with resolution $O(1)$.*

Proof. For symmetric linear threshold functions, all weights are 1. We first perturb the bias term so that $\mathbb{1}\{|X| \geq b\} = \mathbb{1}\{|X| \geq b - 0.5\}$. Then, we pick $X^* = (1, \dots, 1, 1/2, 0, \dots, 0)$ where $w^T X^* = b - 0.5$, i.e., there are $b - 1$ many 1s in X^* and X^* is on the hyperplane $w^T x = b - 0.5$. Picking $c = 1$, we get

$$a_1 = (0, \dots, 0, -1/2, -1, \dots, -1) \tag{3.8}$$

$$a_2 = (2, \dots, 2, \quad 3/2, \quad 1, \dots, \quad 1) \tag{3.9}$$

which is a valid NN representation where the number of 0s in a_1 (and 2s in a_2) is $b - 1$. The resolution is clearly $O(1)$. □

In addition to the result in Theorem 17, we give a similar result for exact threshold functions. For any exact threshold function f which is not a linear threshold function, it is $\text{NN}(f) > 2$ necessarily. For example, the EQ is such an example and the AND function is both exact and linear. We show that any exact threshold function which is not linear has exactly a 3-anchor NN representation where all the anchors are collinear. The geometrical idea is given in Figure 3.2. Contrary to linear threshold functions, the converse does not hold, i.e, a Boolean function with a 3-anchor NN representation need not to be an exact threshold function (see 3-*interval symmetric*

Boolean functions such as CQ in the next section). The resolution of the weights is related to the resolution of the anchors similar to the result in Theorem 17.

Theorem 19. *Let $f(X)$ be an n-input non-linear exact threshold function with weight vector $w \in \mathbb{Z}^n$ and the bias term $b \in \mathbb{Z}$. Then, there is a 3-anchor NN representation of $f(X)$ with resolution $O(RES(||w||_2^2))$ with $||.||_2$ being the Euclidean norm. In general, the resolution is $O(n \log n)$.*

Proof. We follow the idea in the proof of Theorem 17. In addition to a_1 and a_2, we put another anchor a_0 which a solution to $w^T X = b$. a_0 have the opposite labeling of the a_1 and a_2. We assume that $c > 0$.

$$a_0 = X^* \tag{3.10}$$
$$a_1 = X^* - cw \tag{3.11}$$
$$a_2 = X^* + cw \tag{3.12}$$

Without loss of generality, we can assume that there exist a binary X^* such that $w^T X^* = b$ because otherwise, $f(X) = 0$ for all $X \in \{0,1\}^n$, which is a constant function with a trivial 1-anchor NN representation. We have the following necessary and sufficient conditions to claim that this representation is valid indeed:

$$\underline{\textbf{Case 1}}\textbf{:}\ w^T X = b \Leftrightarrow d(a_0, X) < d(a_i, X) \text{ for } i = 1,2 \tag{3.13}$$
$$\underline{\textbf{Case 2}}\textbf{:}\ w^T X \neq b \Leftrightarrow d(a_0, X) > d(a_i, X) \text{ either for } i = 1,2 \tag{3.14}$$

We get the following when we expand the squared Euclidean distance from the input vector to an anchor:

$$d(a_1, X)^2 = |X| - 2a_1^T X + ||a_1||_2^2 \tag{3.15}$$
$$= |X| - 2X^T X^* + ||X^*||_2^2$$
$$+ 2c(w^T X - b) + c^2 ||w||_2^2 \tag{3.16}$$
$$= d(a_0, X)^2 + 2c(w^T X - b) + c^2 ||w||_2^2 \tag{3.17}$$
$$d(a_2, X)^2 = d(a_0, X)^2 - 2c(w^T X - b) + c^2 ||w||_2^2 \tag{3.18}$$

It is clear that $\min_{i \in \{1,2\}} d(a_i, X)^2 = d(a_0, X)^2 - 2c|w^T X - b| + c^2 ||w||_2^2$. When we write the necessary and sufficient conditions more explicitly, we obtain

$$\underline{\textbf{Case 1}}\textbf{:}\ w^T X = b \Leftrightarrow 0 < -2c|w^T X - b| + c^2 ||w||_2^2 \tag{3.19}$$
$$\underline{\textbf{Case 2}}\textbf{:}\ w^T X \neq b \Leftrightarrow 0 > -2c|w^T X - b| + c^2 ||w||_2^2 \tag{3.20}$$

Case 1 is trivial as long as $c \neq 0$ because $w^T X = b$. For Case 2, we have $c < \frac{2|b - w^T X|}{||w||_2^2}$. The minimum value the numerator can take is 1, therefore, the bound is tightest when $c < \frac{2}{||w||_2^2}$. Taking $c = \frac{1}{||w||_2^2}$ suffices and we immediately see that $RES(A) = O(RES(||w||_2^2))$ given that X^* is binary. Since $RES(w) = O(n \log n)$, we obtain the $RES(A) = O(n \log n)$ (Babai et al., 2010). □

Compared to Theorem 17, the resolution is upper bounded by the norm of the weights in Theorem 19. When the weights are constant in n, the norm of the weights might be linear in n and the resolution may not be constant anymore. Geometrically speaking, this is due to the fact that if the anchors a_1 and a_2 are too far from the hyperplane, a_0 might get closer to binary vectors outside the hyperplane itself.

We can similarly define an n-input *symmetric exact threshold function* as a symmetric Boolean function such that $f(X) = \mathbb{1}\{|X| = b\}$. We will prove later an important result is that for some symmetric exact threshold functions, 3-anchor NN representations require $\Omega(\log n)$ resolution (see Theorem 24).

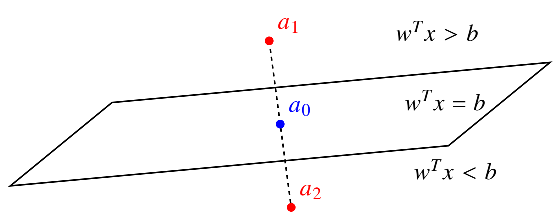


Figure 3.2: The NN Representation of an Exact Threshold Function $\mathbb{1}\{w^T X = b\}$ and its 3-anchor NN Representation. The anchors a_1 and a_2 must be close enough to the hyperplane. All anchors are collinear.

3.2 NN Representation of Symmetric Boolean Functions

Symmetric Boolean functions are useful to prove many complexity results about Boolean function (Bruck, 1990; Håstad, Jukna, and Pudlák, 1995; Stockmeyer, 1976) and more importantly, in the context of nearest neighbors, any n-input Boolean function can be interpreted as an 2^n-input symmetric Boolean function (Siu, Roychowdhury, and Kailath, 1991). Therefore, results on symmetric Boolean functions can provide insights about our understanding of Boolean function complexity. For n-input symmetric Boolean functions, it is known that there is an NN representation of size $n + 1$ and resolution $O(\log n)$ (P. Hajnal, Z. Liu, and Turán, 2022).

Proposition 1 (P. Hajnal, Z. Liu, and Turán, 2022)**.** *For an n-input symmetric Boolean function, $NN(f) \leq n + 1$.*

The PARITY construction mentioned in Theorem 2 can be extended to arbitrary symmetric Boolean functions. The idea is to assign an anchor point to each value of $|X|$. We have the following construction of an NN representation with an anchor matrix A for the given symmetric Boolean function $f(X)$.

$$\begin{array}{c|c} |X| & f(X) \\ \hline 0 & \mathbf{0} \\ 1 & \mathbf{0} \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & \mathbf{0} \end{array} \quad \begin{array}{c} \\ \to a_1 \\ \to a_2 \\ \to a_3 \\ \to a_4 \\ \to a_5 \\ \to a_6 \end{array} \qquad A = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0.2} & \mathbf{0.2} & \mathbf{0.2} & \mathbf{0.2} & \mathbf{0.2} \\ 0.4 & 0.4 & 0.4 & 0.4 & 0.4 \\ 0.6 & 0.6 & 0.6 & 0.6 & 0.6 \\ 0.8 & 0.8 & 0.8 & 0.8 & 0.8 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{bmatrix} \tag{3.21}$$

Proposition 1 is proven by construction and in general, the matrix $A \in \mathbb{R}^{(n+1)\times n}$ contains the anchor $a_i = ((i-1)/n, \dots, (i-1)/n)$ where P and N is a partition of $\{a_1, \dots, a_{n+1}\}$. Regardless of the function itself, this construction requires $n+1$ anchors and we call it the PARITY-*based construction*. In contrast, we know that there are examples of symmetric Boolean functions that require a smaller number of anchors and Proposition 1 is evidently far from being optimal.

$\|X\|$	AND(X)	OR(X)
0	**0**	**0**
1	**0**	1
2	1	1
Anchors	$\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0.5} & \mathbf{0.5} \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ 0.5 & 0.5 \\ 1 & 1 \end{pmatrix}$

Figure 3.3: The PARITY-based constructions for 2-input AND and OR functions. Clearly, the construction given in Fig. 1.7 has smaller size and the same resolution.

One can notice that there is a single transition from 0s to 1s in $f(X)$ for both AND and OR when they are enumerated by $|X|$. Furthermore, the correspondence between $f(X)$ and $|X|$ is particularly useful to measure the complexity of symmetric Boolean functions in Circuit Complexity Theory (Muroga, 1959; Minnick, 1961; Siu and Bruck, 1991). These observations motivate us to define the notion of an *interval* for a symmetric Boolean function in order to measure the NN complexity.

Definition 12. *Let $a \le b \le n$ be some non-negative integers. An* interval $[a, b]$ for an n-input symmetric Boolean function *is defined as follows:*

1. $f(X)$ is constant for $|X| \in [a, b]$.

2. If $a \neq 0$ and $b \neq n$, $f(X_1) \neq f(X_2)$ for any $|X_1| \in [a, b]$ and

- *If $a > 0$ and $b < n$, $|X_2| = a - 1$ and $|X_2| = b + 1$.*
- *If $a = 0$ and $b < n$, $|X_2| = b + 1$.*
- *If $a > 0$ and $b = n$, $|X_2| = a - 1$.*

The quantity $I(f)$ is the total number of intervals for a symmetric Boolean function f.

We usually refer to an interval $[a, b]$ for an n-input symmetric Boolean function shortly as an interval. The example function in Eq. 3.21 and XOR have $I(f) = 3$ (see Fig.1.8) while AND and OR have $I(f) = 2$ (see Fig. 3.3).

It seems natural to simplify the construction in Proposition 1 by assigning an anchor to each interval to reduce the size of the representation to $I(f)$. We call this an $I(f)$ ***interval-anchor assignment*** where there is a one-to-one map between an anchor and an interval of $f(X)$. We assign the i^{th} interval $[a, b]$ to the anchor a_i and enumerate the beginning and the end of the interval assigned to it by $I_{i-1} + 1$ and I_i respectively by using the values of $|X|$. We take $I_0 = -1$ for consistency. The function below is an example where $f(X)$ is a 6-input symmetric function and $I(f) = 3$ and (a_1, a_2, a_3) is an $I(f)$ interval-anchor assignment.

$$\begin{array}{c|c}
|X| & f(X) \\
\hline
0 & \mathbf{0} \\
1 & \mathbf{0} \\
2 & 1 \\
3 & 1 \\
4 & 1 \\
5 & \mathbf{0}
\end{array}
\quad
\begin{array}{cc}
 & \\
\rightarrow a_1 & I_0 + 1 = 0 \\
\rightarrow a_1 & I_1 = 1 \\
\rightarrow a_2 & I_1 + 1 = 2 \\
\rightarrow a_2 & \\
\rightarrow a_2 & I_2 = 4 \\
\rightarrow a_3 & I_2 + 1 = I_3 = 5
\end{array}
\tag{3.22}$$

It might be possible to extend the construction given in Proposition 1 and find an $I(f)$ interval-anchor assignment by computing and perturbing the centroids of anchors belonging to each interval. We call this idea PARITY-*based extension*. The anchors in PARITY-based extensions are symmetric, i.e., they are equal. Consider the construction given in Eq. (3.21). One can obtain a 3-anchor NN representation by the matrix $A' \in \mathbb{R}^{3\times 5}$ using the matrix A.

$$\begin{array}{c|cl} |X| & f(X) & \\ \hline 0 & \mathbf{0} & \rightarrow a'_1 \\ 1 & \mathbf{0} & \\ 2 & 1 & \\ 3 & 1 & \rightarrow a'_2 \\ 4 & 1 & \\ 5 & \mathbf{0} & \rightarrow a'_3 \end{array} \qquad A' = \begin{bmatrix} \mathbf{0.1} & \mathbf{0.1} & \mathbf{0.1} & \mathbf{0.1} & \mathbf{0.1} \\ 0.6 & 0.6 & 0.6 & 0.6 & 0.6 \\ \mathbf{1.1} & \mathbf{1.1} & \mathbf{1.1} & \mathbf{1.1} & \mathbf{1.1} \end{bmatrix} \tag{3.23}$$

Unfortunately, PARITY-based extensions cannot be applied to all symmetric Boolean functions. We give an example where there is no PARITY-based extension in Appendix D. Informally, symmetric Boolean functions treat each input x_i with the same importance and we show that by breaking up the symmetry in the anchor entries and taking the whole NN representation farther away from the Boolean hypercube, we can reduce the size of representation to $I(f)$ so that $NN(f) \leq I(f)$. A non-intuitive discovery is that for symmetric Boolean functions, optimality in complexity and resolution is achieved by anchors that are asymmetric in their entries (see Theorem 20).

Provided that we have $NN(f) \leq I(f)$, is it true that $NN(f) = I(f)$ for symmetric Boolean functions? This is still an open problem, however, we are able to prove it for *periodic* symmetric Boolean functions (see Theorem 21).

Definition 13. *A symmetric Boolean function is called* periodic *if each interval has the same length, which is denoted by T. T is also called* period.

More precisely, if f is a periodic symmetric Boolean function with period T, then for $(k-1)T \leq |X| < kT$, $f(X) = 0$ (or 1) for odd k and $f(X) = 1$ (or 0) for even k. PARITY is a periodic symmetric function with period $T = 1$. Periodic symmetric Boolean functions are useful in other works as well (see the Complete Quadratic function in Bruck, 1990 where $T = 2$).

Upper Bounds on the NN Complexity of Symmetric Boolean Functions

In this subsection, we present an explicit construction for the NN representation of symmetric Boolean functions with $I(f)$ intervals.

Theorem 20. *For an n-input symmetric Boolean function f, $NN(f) \leq I(f)$.*

Moreover, a matrix $B = \mathbb{1} + \epsilon M$ where $B \in \mathbb{R}^{(I(f)-1)\times n}$ can always be used for a construction given that $\epsilon > 0$ is sufficiently small where $\mathbb{1}$ is an all-one matrix and $M \in \mathbb{R}^{(I(f)-1)\times n}$ is a full row rank matrix.

For arbitrary symmetric Boolean functions, we first derive necessary and sufficient conditions for an $I(f)$ interval-anchor assignment.

Lemma 7. *Let f be an n-input symmetric Boolean function and $A \in \mathbb{R}^{I(f)\times n}$ be an anchor matrix for an $I(f)$ interval-anchor assignment. Then, the following condition is necessary and sufficient for any $i \in \{2, \dots, I(f)\}$ and $k < i$:*

$$\max_{X\in\{0,1\}^n:|X|=I_k} \sum_{j=1}^{n}(a_{ij} - a_{kj})x_j < \frac{1}{2}\sum_{j=1}^{n}(a_{ij}^2 - a_{kj}^2) < \min_{X\in\{0,1\}^n:|X|=I_{i-1}+1} \sum_{j=1}^{n}(a_{ij} - a_{kj})x_j \tag{3.24}$$

In general, all a_{ij}s can be different compared to PARITY-based extensions. However, we cannot freely choose them. We have the following observation to prove Lemma 7.

Proposition 2. *For any $I(f)$ interval-anchor assignment with an anchor matrix $A \in \mathbb{R}^{I(f)\times n}$, $a_{ij} > a_{(i-1)j}$ for all $i \in \{2, \dots, I(f)\}$ and $j \in \{1, \dots, n\}$.*

Proof. Consider two Boolean vectors $X = (x_1, \dots, 0, \dots, x_n)$ and $X' = (x_1, \dots, 1, \dots, x_n)$ where they only differ at t^{th} location. Assume that this occurs at the boundary for an interval, i.e., X and X' are closer to a_{i-1} and a_i respectively. Then,

$$d(a_{i-1}, X)^2 - d(a_i, X)^2 < 0 \tag{3.25}$$

$$d(a_{i-1}, X')^2 - d(a_i, X')^2 > 0 \tag{3.26}$$

Subtracting both inequalities, we get $d(a_{i-1}, X)^2 - d(a_{i-1}, X')^2 - (d(a_i, X)^2 - d(a_i, X')^2) < 0$. Since X and X' differ only at the t^{th} location, we get $(2a_{(i-1)t} - 1) - 2(a_{it} - 1) < 0$, hence, $a_{it} > a_{(i-1)t}$. □

Proof of Lemma 7. We begin by writing the simplest necessary and sufficient condition for an $I(f)$ interval-anchor assignment. Consider any two anchors a_i and a_k and assume that $|X| \in [I_{i-1} + 1, I_i]$ so that X is closer to a_i.

$$d(a_i, X)^2 - d(a_k, X)^2 < 0 \tag{3.27}$$

which can be written as

$$\sum_{j=1}^{n}(a_{ij}-x_j)^2-\sum_{j=1}^{n}(a_{kj}-x_j)^2<0 \tag{3.28}$$

for $i \in \{2,\dots,I(f)\}$ and $k<i$. This implies

$$\begin{aligned}\frac{1}{2}\sum_{j=1}^{n}&(a_{ij}^2-a_{kj}^2)\\ &<\min_{X\in\{0,1\}^n:|X|\in[I_{i-1}+1,I_i]}\sum_{j=1}^{n}(a_{ij}-a_{kj})x_j\end{aligned} \tag{3.29}$$

The RHS term is minimized when $|X|=\sum_{j=1}^{n}x_j=I_{i-1}+1$ by Proposition 2 because each $a_{ij}-a_{kj}>0$. Then,

$$\frac{1}{2}\sum_{j=1}^{n}(a_{ij}^2-a_{kj}^2)<\min_{X\in\{0,1\}^n:|X|=I_{i-1}+1}\sum_{j=1}^{n}(a_{ij}-a_{kj})x_j \tag{3.30}$$

We similarly do the analysis for the anchors a_i and a_k but this time assuming that $|X|\in[I_{k-1}+1,I_k]$ so that X is closer to a_k. We still take $k<i$. Then,

$$\frac{1}{2}\sum_{j=1}^{n}(a_{kj}^2-a_{ij}^2)<\min_{X\in\{0,1\}^n:|X|\in[I_{k-1}+1,I_k]}\sum_{j=1}^{n}(a_{kj}-a_{ij})x_j \tag{3.31}$$

$$\begin{aligned}\frac{1}{2}\sum_{j=1}^{n}(a_{ij}^2-a_{kj}^2)&>\max_{X\in\{0,1\}^n:|X|\in[I_{k-1}+1,I_k]}\sum_{j=1}^{n}(a_{ij}-a_{kj})x_j\\ &=\max_{X\in\{0,1\}^n:|X|=I_k}\sum_{j=1}^{n}(a_{ij}-a_{kj})x_j\end{aligned} \tag{3.32}$$

Now, the RHS is maximized when $|X|=I_k$ by Proposition 2. By combining (3.30) and (3.32), we complete the proof. □

It is possible to simplify the necessary and sufficient conditions given in Lemma 7. We prove that looking at (k,i) pairs in the form $(i-1,i)$ still provides the necessary and sufficient information compared to all (k,i) pairs such that $k<i$.

Lemma 8. *Let f be an n-input symmetric Boolean function and $A\in\mathbb{R}^{I(f)\times n}$ be an anchor matrix for an $I(f)$ interval-anchor assignment. Then, the following condition is necessary and sufficient for any $i\in\{2,\dots,I(f)\}$:*

$$\begin{aligned}\max_{X\in\{0,1\}^n:|X|=I_{i-1}}\sum_{j=1}^{n}(a_{ij}-a_{(i-1)j})x_j&<\frac{1}{2}\sum_{j=1}^{n}(a_{ij}^2-a_{(i-1)j}^2)\\ &<\min_{X\in\{0,1\}^n:|X|=I_{i-1}+1}\sum_{j=1}^{n}(a_{ij}-a_{(i-1)j})x_j\end{aligned} \tag{3.33}$$

Proof of Lemma 8. It is obvious that this is necessary if we apply $k = i - 1$. We will show that we can use the Eq. (3.33) to obtain Lemma 7 by summing them telescopically until some fixed $k < i$.

$$\begin{aligned}
\max_{X\in\{0,1\}^n:|X|=I_{i-1}} &\sum_{j=1}^{n}(a_{ij} - a_{(i-1)j})x_j \\
&< \frac{1}{2}\sum_{j=1}^{n}(a_{ij}^2 - a_{(i-1)j}^2) \\
&< \min_{X\in\{0,1\}^n:|X|=I_{i-1}+1} \sum_{j=1}^{n}(a_{ij} - a_{(i-1)j})x_j \\
&\vdots \\
\max_{X\in\{0,1\}^n:|X|=I_k} &\sum_{j=1}^{n}(a_{(k+1)j} - a_{kj})x_j \\
&< \frac{1}{2}\sum_{j=1}^{n}(a_{(k+1)j}^2 - a_{kj}^2) \\
&< \min_{X\in\{0,1\}^n:|X|=I_k+1} \sum_{j=1}^{n}(a_{(k+1)j} - a_{kj})x_j
\end{aligned}$$

Let us combine the inequalities and focus on the RHS.

$$\sum_{l=k+1}^{i} \min_{X\in\{0,1\}^n:|X|=I_{l-1}+1} \sum_{j=1}^{n}(a_{lj} - a_{(l-1)j})x_j \tag{3.34}$$

We can replace the constraint sets of all of the minimization expressions with $I_{i-1}+1$ because

$$\begin{aligned}
\min_{X\in\{0,1\}^n:|X|=I_{l-1}+1} &\sum_{j=1}^{n}(a_{lj} - a_{(l-1)j})x_j \\
&< \min_{X\in\{0,1\}^n:|X|=I_{i-1}+1} \sum_{j=1}^{n}(a_{lj} - a_{(l-1)j})x_j
\end{aligned} \tag{3.35}$$

by Proposition 2 and $l \leq i$. Then, we combine the objective functions telescopically again so that

$$\frac{1}{2}\sum_{j=1}^{n}(a_{ij}^2 - a_{kj}^2) < \min_{X\in\{0,1\}^n:|X|=I_{i-1}+1} \sum_{j=1}^{n}(a_{ij} - a_{kj})x_j \tag{3.36}$$

The LHS can be handled in a similar manner. Then, we can obtain Lemma 7 exactly. □

We can now prove Theorem 20 by finding explicit sets of anchors satisfying Lemma 8.

Proof of Theorem 20. Let us define a matrix $B \in \mathbb{R}^{I(f)-1\times n}$ where $b_{ij} = a_{(i+1)j} - a_{ij}$. Rewriting the condition in Lemma 8 using the B matrix and transforming i to $i+1$, we get

$$\begin{aligned}\max_{X\in\{0,1\}^n:|X|=I_i} \sum_{j=1}^{n} b_{ij}x_j \\ &< \frac{1}{2}\sum_{j=1}^{n}\left(- b_{ij}^2 + 2b_{ij}\sum_{k=1}^{i} b_{kj} + 2b_{ij}a_{1j}\right) \\ &< \min_{X\in\{0,1\}^n:|X|=I_i+1} \sum_{j=1}^{n} b_{ij}x_j \end{aligned} \tag{3.37}$$

where the middle term is computed by the identity $a_{ij} = a_{1j} + \sum_{k=1}^{i-1} b_{kj} = a_{1j} - b_{ij} + \sum_{k=1}^{i} b_{kj}$.

$$\frac{1}{2}\sum_{j=1}^{n}(a_{(i+1)j}^2 - a_{ij}^2) = \frac{1}{2}\sum_{j=1}^{n} b_{ij}(b_{ij} + 2a_{ij}) \tag{3.38}$$

$$= \frac{1}{2}\sum_{j=1}^{n}\left(b_{ij}^2 + 2b_{ij}\left(a_{1j} - b_{ij} + \sum_{k=1}^{i} b_{kj}\right)\right) \tag{3.39}$$

$$= \frac{1}{2}\sum_{j=1}^{n}\left(- b_{ij}^2 + 2b_{ij}\sum_{k=1}^{i} b_{kj} + 2b_{ij}a_{1j}\right) \tag{3.40}$$

Let us take a convex combination of the LHS and RHS with some $\lambda_i \in (0,1)$ to make it equal to the middle term for each $i \in \{1,\dots,I(f)-1\}$. Therefore, we want to solve $a_1 = (a_{11},\dots,a_{1n})$ for the $Ba_1 = c$ where

$$\begin{aligned}(Ba_1)_i = c_i =& \frac{1}{2}\sum_{j=1}^{n} b_{ij}^2 - \sum_{j=1}^{n} b_{ij}\sum_{k=1}^{i} b_{kj} \\ &+ \lambda_i \max_{X\in\{0,1\}^n:|X|=I_i} \sum_{j=1}^{n} b_{ij}x_j \\ &+ (1-\lambda_i) \min_{X\in\{0,1\}^n:|X|=I_i+1} \sum_{j=1}^{n} b_{ij}x_j \end{aligned} \tag{3.41}$$

As long as this system of linear equations is consistent, we have a solution for a_1 and a construction for $I(f)$ interval-anchor assignment. If B is a full row rank matrix, the system is guaranteed to be consistent. As long as the LHS is smaller than the RHS in Eq. (3.37), any full row rank B matrices will work. Therefore, we can claim that $B = \mathbb{1} + \epsilon M$ for sufficiently small $\epsilon > 0$ and full row rank M can be used to construct an anchor matrix for an arbitrary symmetric Boolean function. □

In Theorem 20, the B matrix is actually constructed by $b_{ij} = a_{ij} - a_{(i-1)j}$ for $i \in \{2, \ldots, I(f)\}$ and $j \in \{1, \ldots, n\}$. Intuitively, as long as the entries of B do not differ much, we can find anchors where the LHS and the RHS in Eq. (3.33) should hold. Therefore, $B = \mathbb{1} + \epsilon M$ is a valid choice for sufficiently small $\epsilon > 0$. We also remark that the full row rank B matrix property is only sufficient to find a construction. For example, if $b_{ij} = 1/n$ for all i, j, this construction can be reduced to the PARITY-based construction given in (P. Hajnal, Z. Liu, and Turán, 2022).

In general, the PARITY-based approach results in $O(\log n)$ resolution. We can pick B such that any symmetric Boolean function can also have $O(\log n)$ resolution with $I(f)$ anchors. We present a family of examples of B with $O(\log n)$ resolution in Section 3.2. Such an example was already given in Appendix D.

Lower Bounds on the NN Complexity of Symmetric Boolean Functions

There are various circuit complexity lower bounds on symmetric Boolean functions (Paturi and Saks, 1990; Siu, Roychowdhury, and Kailath, 1991; Spielman, 1992). Similarly, lower bounds for the number of anchors can be proven for PARITY using $\{1, 2\}$-sign representations of Boolean functions (P. Hajnal, Z. Liu, and Turán, 2022; Hansen and V. V. Podolskii, 2015). We prove a more general result for periodic symmetric Boolean functions.

Theorem 21. *For a periodic symmetric Boolean function of $I(f)$ intervals, $NN(f) \geq I(f)$.*

To prove this Theorem, we use a necessary condition for any NN representation of a periodic symmetric Boolean function.

Proposition 3. *Consider an NN representation of a periodic symmetric Boolean function $f(X)$ of period T with an anchor matrix $A \in \mathbb{R}^{m\times n}$. Assume that $\sum_{j=1}^{T} a_{1j} \geq \sum_{j=1}^{T} a_{ij}$ for any $i \geq 2$. Then, no binary vectors that are closest to a_1 can have all 0s in the first T coordinates.*

We first claim that the assumption in Proposition 3 can be assumed without loss of generality by first observing that the NN representations of symmetric Boolean functions are equivalent up to permutations of anchors in Proposition 4. Then, we find the anchor with the maximal T sum in the entries and rearrange the rows and columns so that the claim $\sum_{j=1}^{T} a_{1j} \geq \sum_{j=1}^{T} a_{ij}$ for any $i \geq 2$ holds.

Proposition 4. *Suppose that an anchor matrix $A \in \mathbb{R}^{m \times n}$ is an NN representation with m anchors for an n-input symmetric Boolean function. Then, any permutation of rows and columns of A is an NN representation for the same function.*

Proof. Permutation of rows is trivial and can be done for any NN representation. For the columns, we can use the definition of symmetric Boolean functions where $f(X) = f(\sigma(X))$ for any permutation $\sigma(.)$. The closest anchor to a Boolean vector X can be found by

$$\begin{aligned} \arg\min_i d(a_i, X)^2 &= \arg\min_i |X| - 2(AX)_i + ||a_i||_2^2 && (3.42) \\ &= \arg\max_i \left(2AX - \text{diag}(AA^T)\right)_i && (3.43) \end{aligned}$$

where $||.||_2$ denotes the Euclidean norm and $\text{diag}(M)$ is the all-zero matrix except the diagonal entries of M. Let $P \in \{0,1\}^{n \times n}$ be a permutation matrix. Then,

$$\arg\max_i \left(2A(PX) - \text{diag}(AA^T)\right)_i \tag{3.44}$$

is an anchor index assigned to the same anchor type (either positive or negative) because f is symmetric. Note that Eq. (3.44) is equivalent to

$$\begin{aligned} \arg\max_i &\left(2(AP)X - \text{diag}((AP)(AP)^T)\right)_i \\ &= \arg\max_i \left(2(AP)X - \text{diag}(AA^T)\right)_i \end{aligned} \tag{3.45}$$

□

Proof of Proposition 3. We use a similar idea used for Proposition 2. Assume that there is a vector $X = (0, \dots, 0, x_{T+1}, \dots, x_n)$ assigned to a_1 for contradiction. Let $X' = (1, \dots, 1, x_{T+1}, \dots, x_n)$. Then,

$$\begin{aligned} d(a_1, X)^2 &< d(a_i, X)^2 \quad \forall i \geq 2 && (3.46) \\ d(a_1, X')^2 &> d(a_i, X')^2 \quad \text{for some } i \geq 2 && (3.47) \end{aligned}$$

We get Eq. (3.46) by the assumption for contradiction. Eq. (3.47) is obtained by the fact that $f(X) \neq f(X')$ because we have a jump of length T for a given value of $|X|$. Subtracting both, we get

$$\sum_{j=1}^{T} a_{1j} < \sum_{j=1}^{T} a_{ij} \tag{3.48}$$

for some $i \geq 2$, which is the desired contradiction. □

Proof of Theorem 21. Let f be a periodic symmetric Boolean function with period T so that $n = I(f)T$. Suppose that the NN representation for this function has an anchor matrix $A \in \mathbb{R}^{m \times n}$ where

$$\sum_{j=(k-1)T+1}^{kT} a_{kj} \geq \sum_{j=(k-1)T+1}^{kT} a_{ij} \tag{3.49}$$

for any $k \in \{1, \ldots, m-1\}$ and $k < i$. This holds without loss of generality because we can rank the maximal sums of the k entries of the anchors and rearrange the rows and columns of A by Proposition 4.

Iteratively, for each $k \in \{1, \ldots, m-1\}$, we see that if $X_i = 0$ for $i \in \{1, \ldots, kT\}$, then $(a_1, \ldots, a_k)$ cannot be assigned to X by Proposition 3. Since $n = I(f)T$ for a periodic symmetric Boolean function, if $m < I(f)$, there will remain X vectors with different $f(X)$ values assigned to a single anchor, leading to a contradiction. □

Proving that $NN(f) = I(f)$ for all symmetric Boolean functions is incomplete and we conjecture that this statement is true.

Conjecture 2. *Let $f(X)$ be an n-input Symmetric Boolean function with $I(f)$ many intervals. Then, NN$(f) = I(f)$.*

We verify the conjecture for $I(f) \leq 4$ because it holds that if $I(f) \leq 3$, any 3-anchor NN representation should be an interval-anchor assignment except 2 counterexamples. This is proven in Lemma 10.

Lemma 9. *Let f be an n-input symmetric Boolean function with $I(f) \leq 4$. Then, NN$(f) \geq I(f)$.*

We give a necessary condition for all NN representations except the constant functions $f(X) = 0$ and $f(X) = 1$ by a geometrical argument.

Proposition 5. *For an arbitrary NN representation of size at least 2, pick any positive and negative anchor, a and b. Let $\mathcal{X}$ and $\mathcal{Y}$ be the set of binary vectors closest to a and b, respectively. Then, $conv(\mathcal{X}) \cap conv(\mathcal{Y}) = \emptyset$ where $conv(\mathcal{A})$ denotes the convex hull of a set $\mathcal{A}$.*

Proof. Let $|\mathcal{X}| = k$ and $|\mathcal{Y}| = l$. We assume that $\mathcal{X} = \{X_1, \dots, X_k\}$ and $\mathcal{Y} = \{Y_1, \dots, Y_l\}$. Also, let u and v be arbitrary convex combinations for the sets $\mathcal{X}$ and $\mathcal{Y}$. That is, $u = \sum_{i=1}^{k} \lambda_i X_i$ and $v = \sum_{i=1}^{l} \mu_i Y_i$ where $X_i \in \mathcal{X}$, $Y_i \in \mathcal{Y}$, $\lambda_i \in [0,1]$, $\sum_{i=1}^{k} \lambda_i = 1$, $\mu_i \in [0,1]$, and $\sum_{i=1}^{l} \mu_i = 1$.

We know that $d(a,X_i)^2 < d(b,X_i)^2$ for $i \in \{1,\dots,k\}$ and $d(a,Y_i)^2 > d(b,Y_i)^2$ for $i \in \{1,\dots,l\}$. Suppose for contradiction that there is a set of λs and μs so that $u = v$.

$$d(a,u)^2 - d(b,u)^2 = ||a||^2 - ||b||^2 - 2(a-b)^T \sum_{i=1}^{k} \lambda_i X_i \tag{3.50}$$

$$= \sum_{i=1}^{k} \lambda_i \Big(||a||^2 - 2(a-b)^T X_i - ||b||^2 \Big) \tag{3.51}$$

$$= \sum_{i=1}^{k} \lambda_i \Big(d(a,X_i)^2 - d(b,X_i)^2 \Big) < 0 \tag{3.52}$$

where we use $\lambda_i \geq 0$ and $\sum_{i=1}^{k} \lambda_i = 1$. We do the same for v since $u = v$, to obtain

$$d(a,v)^2 - d(b,v)^2 = \sum_{i=1}^{l} \mu_i \Big(d(a,Y_i)^2 - d(b,Y_i)^2 \Big) > 0 \tag{3.53}$$

resulting in a contradiction. □

Lemma 10. *Let f be an n-input symmetric Boolean function with 3 intervals. Then, any $I(f)$-anchor NN representation of f needs to be an interval-anchor assignment except for the function $f(X) = 1$ (or 0) for $|X| \in \{0, n\}$ and 0 (or 1) otherwise.*

An alternative NN representation for the 5-input function in in Lemma 10 is given below where the middle interval is shared by two blue anchors and the red anchor $(0.5, 0.5, 0.5, 0.5, 0.5)$ is closest to the all-0 and all-1 vectors.

$\lvert X\rvert$	$f(X)$
0	**0**
1	1
2	1
3	1
4	1
5	**0**

$$A' = \begin{bmatrix} 0 & 0.57 & 0.57 & 0.57 & 0.57 \\ \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} & \mathbf{0.5} \\ 1 & 0.43 & 0.43 & 0.43 & 0.43 \end{bmatrix} \tag{3.54}$$

Proof of Lemma 10. There are only two possible anchor assignments for symmetric Boolean functions with 3 intervals besides an interval-anchor assignment: Either a_2 is assigned to the region $|X| \in [I_1+1, I_2]$ and a_1 & a_3 shares the rest (**Case 1**) or a_2 & a_3 shares the same region $|X| \in [I_1+1, I_2]$ and a_1 is assigned to the rest (**Case 2**).

For Case 1: We pick X_1 and X_2 assigned either both to a_1 or a_3 where $|X_1| = t_1 < I_1+1$ and $|X_2| = t_2 > I_2$. Consider also an integer $t \in [I_1+1, I_2]$ independently. This is always possible except the function given in the description. However, for that function, this case implies an interval-anchor assignment as there are unique vectors both for $|X| = 0$ and $|X| = n$. We denote the set of coordinates where $X_1 = X_2 = 1$ by $\mathcal{S}_1$, $X_1 = 1, X_2 = 0$ by $\mathcal{S}_2$, $X_1 = 0, X_2 = 1$ by $\mathcal{S}_3$, and finally, $X_1 = X_2 = 0$ by $\mathcal{S}_4$. Clearly, $t_1 = |\mathcal{S}_1| + |\mathcal{S}_2|$, $t_2 = |\mathcal{S}_1| + |\mathcal{S}_3|$, $|\mathcal{S}_1| + |\mathcal{S}_2| + |\mathcal{S}_3| + |\mathcal{S}_4| = n$. Consider the following as an example.

$$X_1 = (1, \dots, 1, 1, \dots, 1, 0, \dots, 0, 0, \dots, 0) \tag{3.55}$$

$$X_2 = (\underbrace{1, \dots, 1}_{\mathcal{S}_1}, \underbrace{0, \dots, 0}_{\mathcal{S}_2}, \underbrace{1, \dots, 1}_{\mathcal{S}_3}, \underbrace{0, \dots, 0}_{\mathcal{S}_4}) \tag{3.56}$$

Note that the example representation given in Eq. (3.55) and (3.56) can be assumed without loss of generality by the reordering of indices. Also, $\mathcal{S}_3 \neq \emptyset$ is necessary by the choice of t_1, t_2, and t. It is also clear that $|\mathcal{S}_3| = t_2 - t_1 + |\mathcal{S}_2| \geq t - t_1$ and $|\mathcal{S}_3| = t_2 - |\mathcal{S}_1| \geq t - |\mathcal{S}_1|$.

It can be easily verified that the convex combination $X' = \frac{t_2-t}{t_2-t_1}X_1 + \frac{t-t_1}{t_2-t_1}X_2$ lies on the hyperplane $|X| = t$. We further claim that it is in the convex hull of the set of binary vectors Xs where $|X| = t$.

Let Y be the average of all binary vectors on the hyperplane $|X| = t$ with t_1 many 1s at the locations of 1s of X_1, S_4 many 0s at the location of 0s of X_1. That is, we put the remaining $t - t_1$ many ones to the indices in $\mathcal{S}_3$. It is clear that there are $\binom{S_3}{t-t_1}$ many such vectors. The average value in the $\mathcal{S}_3$ region is $K = \binom{|\mathcal{S}_3|-1}{t-t_1-1} / \binom{|\mathcal{S}_3|}{t-t_1} = (t-t_1)/|\mathcal{S}_3|$.

Similarly, we define Z as the average of all binary vectors on the hyperplane $|X| = t$ with all 1s for the indices in $\mathcal{S}_1$ and the $t - |\mathcal{S}_1| =$ remaining 1s will be distributed in the region $\mathcal{S}_3$. The average value in the $\mathcal{S}_3$ region is $L = \binom{|\mathcal{S}_3|-1}{t-|\mathcal{S}_1|-1} / \binom{|\mathcal{S}_3|}{t-|\mathcal{S}_1|} = (t - |\mathcal{S}_1|)/|\mathcal{S}_3|$.

$$Y = (1, \dots, 1, 1, \dots, 1, K, \dots, K, 0 \dots, 0) \tag{3.57}$$

$$Z = (\underbrace{1, \dots, 1}_{\mathcal{S}_1}, \underbrace{0, \dots, 0}_{\mathcal{S}_2}, \underbrace{L, \dots, L}_{\mathcal{S}_3}, \underbrace{0, \dots, 0}_{\mathcal{S}_4}) \tag{3.58}$$

Then, it is easy to verify that $X' = \frac{t_2-t}{t_2-t_1}Y + \frac{t-t_1}{t_2-t_1}Z$ as $K(t_2-t)/(t_2-t_1) + L(t-t_1)/(t_2-t_1) = (t-t_1)/(t_2-t_1)$.

For Case 2: When a_2 & a_3 share the interval $[I_1+1, I_2]$, all the vectors such that $|X| = t$ for $t \in [I_1+1, I_2]$ cannot be assigned to one of the anchors as this will reduce to **Case 1**. Therefore, a_2 & a_3 must share $|X| = t$ for any $t \in [I_1+1, I_2]$.

Let $X_1 \neq X_2$, and the $\mathcal{S}_i$ be the same as in **Case 1** except $|X_1| = |X_2| = I_1 + 1$. We can choose X_1 and X_2 closest to a_2 or a_3 respectively. In this case, $\mathcal{S}_2$ and $\mathcal{S}_3$ are not empty. We define X_3 for X_1&X_2 such that $|X_3| = I_2$ and X_3 has all 1s for the indices in $\mathcal{S}_1$ and $\mathcal{S}_2$ except the entry in the first index for $\mathcal{S}_2$ is zero. Also, we keep the first entry in $\mathcal{S}_3$ to be zero as well and fill the rest of the indices arbitrarily. Hence, by construction, we have $I_2 < n-1$.

$$X_3 = (\underbrace{1, \dots, 1}_{\mathcal{S}_1}, \underbrace{0, 1, \dots, 1}_{\mathcal{S}_2}, \underbrace{0, 1, 0, \dots, 1}_{\mathcal{S}_3 \cup \mathcal{S}_4}) \tag{3.59}$$

Similarly, by reversing the roles of 0s and 1s, we can obtain $I_1 > 0$ condition. Therefore, the construction of X_3 is possible whenever either $I_1 > 0$ or $I_3 < n-1$. The only exception is when $f(X) = 1$ (or 0) for $|X| \in \{0, n\}$ and 0 (or 1) otherwise.

Notice that X_3 is constructed by using $\mathcal{S}$ depending both on a_2 and a_3. In general, X_3 can be assigned either to a_2 or a_3. Assume that it is assigned to a_2. Otherwise, change X_1 to X_2 in the following calaims: Let $Y = X_1 \wedge X_3$ and $Z = X_1 \vee X_3$. Clearly, $|Y| \leq I_1 - 1$ and $|Z| \geq I_2 + 1$ and they are both assigned to a_1. We finally claim that $Y/2 + Z/2 = X_1/2 + X_3/2$.

In conclusion, both cases contradict Proposition 5 and an interval-anchor assignment is necessary. □

Proof of Lemma 9. The cases when $I(f) \in \{1, 2\}$ is trivial. When $I(f) = 3$, we can use Proposition 2. Now, assume that 3-anchor NN representations exist for symmetric Boolean functions with $I(f) = 4$. Then, the assignment of anchors to the intervals can only belong to two cases: a_1 is assigned to the first and third intervals where a_2 & a_3 shares the binary vectors for the second and fourth intervals, and vice versa. In both cases, there is a structure similar to **Case 1** of the proof of Lemma 10. By following similar steps, we can obtain a contradiction to Proposition 5 and conclude that 3-anchor NN representations do not exist for symmetric Boolean functions with $I(f) = 4$. □

The Resolution of the NN Representations of Symmetric Boolean Functions

It is known that the PARITY-based construction for symmetric Boolean functions results in $\lceil \log_2 (n+1) \rceil$ resolution. We claim that the construction presented in Section 3.2 can admit $O(\log n)$ resolution. Let Moore-Penrose pseudoinverse of a matrix A be denoted by A^+.

Theorem 22. *Suppose that we are given a rational matrix $B \in \mathbb{Q}^{m \times n}$ such that $B = \mathbb{1}_{m \times n} + \epsilon I_{m,n}$ where $\mathbb{1}_{m \times n}$ is an all-one matrix, ϵ is a rational constant, and $I_{m,n}$ is a submatrix of the $n \times n$ identity matrix with the first m rows. Then, $RES(B^+) = O(\log n)$.*

We refer to Theorem 3 in (Meyer, 1973) to find the entries of B^+ explicitly and to prove Theorem 22.

Theorem 23 (Meyer, 1973)**.** *Suppose that we are given a matrix $A \in \mathbb{R}^{m \times n}$ and column vectors $c \in \mathbb{R}^m, d \in \mathbb{R}^n$. Then, if c is in the column space of A and the quantity $\beta = 1 + d^T A^+ c \neq 0$, we have*

$$\begin{aligned}(A + cd^T)^+ = A^+ &+ \frac{1}{\beta} v^T k^T A^+ \\ &- \frac{(||k||^2 v^T + k\beta)(||v||^2 k^T A^+ + h\beta)}{\beta(||k||^2 ||v||^2 + \beta^2)}\end{aligned} \tag{3.60}$$

where $v = d^T(I - A^+ A)$, $k = A^+ c$, and $h = d^T A^+$.

Proof of Theorem 22. To apply Theorem 23, we pick $c = \mathbb{1}_m$ and $d = \mathbb{1}_n$ where $\mathbb{1}_k$ denotes the all-one vector in $\mathbb{R}^k$ for an integer k. Also, pick $A = \epsilon I_{m,n}$. Therefore, $A^+ = \frac{1}{\epsilon} I_{m,n}^T$.

We compute the pseudoinverse of $\mathbb{1}_{m \times n} + \epsilon I_{m,n}$ as the following closed form expression.

$$\left(\mathbb{1}_{m \times n} + \epsilon I_{m,n}\right)^+ = \begin{bmatrix} \frac{1}{\epsilon} I_{m \times m} - \frac{1 + n/\epsilon}{m(n-m) + (m+\epsilon)^2} \mathbb{1}_{m \times m} \\ \frac{1}{m(n-m) + (m+\epsilon)^2} \mathbb{1}_{(n-m) \times m} \end{bmatrix} \tag{3.61}$$

Since $m \leq n$ and ϵ is a rational constant, the denominator of all entries of the pseudoinverse is a function of $O(n^2)$ (and $O(n)$ for the numerator). Therefore, the resolution of the entries of B^+ is $O(\log n)$. $\square$

Corollary 23.1. *For every n-input symmetric Boolean function, there is an anchor matrix $A \in \mathbb{Q}^{I(f) \times n}$ such that $RES(A) = O(\log n)$.*

Proof. For the matrix $B = \mathbb{1}_{m\times n} + \epsilon M$ in Theorem 20 with $m = I(f) - 1$, we pick $\epsilon = 1/2$ and $M = I_{m,n}$ where $I_{m,n}$ is a submatrix of $n \times n$ identity matrix with the first m rows. It can be easily verified that this B matrix where $b_{ij} = a_{ij} - a_{(i-1)j}$ satisfies the necessary and sufficient condition (3.33) for the interval-anchor assignment. The first anchor a_1 can be obtained by B^+c where c is given in Eq. (3.41).

With $\lambda_i = 1/2$ for all $i \in \{1, \ldots, m\}$, each entry of c depends on the entries on B and n polynomially. By Theorem 22, B^+ has entries with at most $O(\log n)$ resolution and therefore, we see that the entries of $a_1 = B^+c$ has at most $O(\log n)$ resolution (consequently, $a_2, \ldots, a_m$). In conclusion, we can always obtain an interval-anchor assignment with $O(\log n)$ resolution. □

The natural question is whether $O(\log n)$ resolution is optimal for some symmetric Boolean functions. In general, the answer is negative. We have shown that for symmetric linear threshold functions have 2 intervals, one can obtain optimal NN representations in size with constant resolution with anchors

$$a_1 = (0, \ldots, 0, -1/2, -1, \ldots, -1) \tag{3.62}$$

$$a_2 = (2, \ldots, 2, \quad 3/2, \quad 1, \ldots, \quad 1) \tag{3.63}$$

is always an NN representation where the number of 0s in a_1 (and 2s in a_2) is $b - 1$. This is given by Corollary 18.1.

However, if the number of intervals is at least 3, the resolution is lower bounded by $\Omega(\log n)$ for some symmetric Boolean functions and the construction presented in Section 3.2 has optimal resolution by Corollary 18.1.

Theorem 24. *Let f be an n-input symmetric Boolean function where $f(X) = 1$ for $|X| = \lfloor n/2 \rfloor + 1$ and 0 otherwise. Then, any 3-anchor NN representation has $\Omega(\log n)$ resolution.*

There is an anchor matrix A for the function in Theorem 24 such that $RES(A) = O(\log n)$ via PARITY-based extensions.

To prove Theorem 24, we use Lemma 10 so that all 3 anchor representations are interval-anchor assignments. Therefore, we can use the necessary and sufficient condition in Lemma 8 to prove the resolution lower bound.

Proof of Theorem 24. For the n-input symmetric Boolean function given in the Theorem, we always have an interval-anchor assignment by Lemma 10. For an interval-anchor assignment, let $B \in \mathbb{R}^{2\times n}$ $b_{ij} = a_{(i+1)j} - a_{ij} > 0$ for $i \in \{1,2\}$ and $j \in \{1,\dots,n\}$. Then, let us write the necessary and sufficient conditions by Lemma 8. Without loss of generality, assume that $0 < b_{11} \le b_{12} \le \cdots \le b_{1n}$ as we can always reorder the columns of B. Then, the first condition will be

$$\max_{X\in\{0,1\}^n:|X|=\lfloor n/2\rfloor} \sum_{j=1}^{n} b_{1j}x_j < \frac{1}{2}\sum_{j=1}^{n} b_{1j}(b_{1j}+2a_{1j}) \\ < \min_{X\in\{0,1\}^n:|X|=\lfloor n/2\rfloor+1} \sum_{j=1}^{n} b_{1j}x_j \tag{3.64}$$

We have two important inequalities based on this condition.

$$b_{1(n-\sqrt{n})} < b_{1(\sqrt{n}+1)} + \frac{1}{\sqrt{n}} b_{11} \tag{3.65}$$

$$b_{1n} < 2b_{1\sqrt{n}} \le 2b_{1(\sqrt{n}+1)} \tag{3.66}$$

To prove Eq. (3.65), we assume the contrary such that $b_{1(n-\sqrt{n})} \ge b_{1(\sqrt{n}+1)} + \frac{1}{\sqrt{n}} b_{11}$. Also, $b_{1(n-\sqrt{n}+j)} \ge b_{1(n-\sqrt{n})}$ for $1 \le j \le \sqrt{n}$ and $b_{1(\sqrt{n}+1)} \ge b_{1j}$ for $1 \le j \le \sqrt{n}+1$. Hence, if we sum over $j \in \{1,\dots,\sqrt{n}\}$, we get

$$\sum_{j=1}^{\sqrt{n}} b_{1(n-\sqrt{n}+j)} \ge \sum_{j=1}^{\sqrt{n}} \left(b_{1(j+1)} + \frac{1}{\sqrt{n}} b_{11} \right) \tag{3.67}$$

Let us add b_{1j} for $j \in \{n - \lfloor n/2\rfloor + 1, \dots, n-\sqrt{n}\}$ to both sides. Then, we get

$$\sum_{j=n-\lfloor n/2\rfloor+1}^{n} b_{1j} \ge \sum_{j=1}^{\sqrt{n}} \left(b_{1(j+1)} + \frac{1}{\sqrt{n}} b_{11} \right) + \sum_{j=n-\lfloor n/2\rfloor+1}^{n-\sqrt{n}} b_{1j} \tag{3.68}$$

where the LHS has $\lfloor n/2\rfloor$ many terms and the RHS has $\lfloor n/2\rfloor + 1$ many terms. This contradicts Eq. (3.64).

Similarly, to prove Eq. (3.66), we assume the contrary such that $b_{1n} \ge 2b_{1\sqrt{n}} \ge b_{11} + b_{12}$. If we add both sides $\sum_{i=n-\lfloor n/2\rfloor+1}^{n-1} b_{1i}$, we get

$$\sum_{i=n-\lfloor n/2\rfloor+1}^{n} b_{1i} \ge b_{11} + b_{12} + \sum_{i=n-\lfloor n/2\rfloor+1}^{n-1} b_{1i} \tag{3.69}$$

where the LHS has $\lfloor n/2\rfloor$ many terms and the RHS $\lfloor n/2\rfloor + 1$ many terms, contradicting Eq. (3.64).

We now want to bound the middle term in Eq. (3.64). We divide the sum in three parts. Essentially, we want to show that the main contribution in the value of the whole summation is due to the middle term.

$$\sum_{j=1}^{\sqrt{n}} b_{1j}(b_{1j}+2a_{1j}) + \sum_{j=\sqrt{n}+1}^{n-\sqrt{n}} b_{1j}(b_{1j}+2a_{1j}) + \sum_{j=n-\sqrt{n}+1}^{n} b_{1j}(b_{1j}+2a_{1j}) \tag{3.70}$$

We also divide the middle term in Eq. (3.70) into two parts depending on whether the $b_{1j}+2a_{1j}$ terms are positive or not. Let $\mathcal{J}^+$ denote the indices j where $b_{1j}+2a_{1j} > 0$ and $\mathcal{J}^-$ otherwise. Then, we define $\mathcal{S}^+ = \sum_{j\in\mathcal{J}^+}(b_{1j}+2a_{1j})$ and $\mathcal{S}^- = \sum_{j\in\mathcal{J}^-}(b_{1j}+2a_{1j})$ so that

$$\sum_{j=\sqrt{n}+1}^{n-\sqrt{n}} b_{1j}(b_{1j}+2a_{ij}) > b_{1(\sqrt{n}+1)}\mathcal{S}^+ + b_{1(n-\sqrt{n})}\mathcal{S}^- \tag{3.71}$$

$$> b_{1(\sqrt{n}+1)}\left(\mathcal{S}^+ + \mathcal{S}^-\right) - \sqrt{n}2^{r+1}b_{1(\sqrt{n}+1)} \tag{3.72}$$

where r denotes the resolution of the representation. If r is the resolution of the representation (and hence, the anchor matrix A), it is clear that $|b_{1j}+2a_{1j}| = |a_{1j}+a_{2j}| \leq |a_{1j}| + |a_{2j}| < 2^{r+1}$ and we can rewrite Eq. 3.65 so that $b_{1(n-\sqrt{n})} < b_{1(\sqrt{n}+1)}\left(1+\frac{1}{\sqrt{n}}\right)$. We also have the loose bound $\mathcal{S}^- > -n2^{r+1}$ and by using these, we can obtain Eq. (3.72).

We find lower bounds for the first term in Eq. (3.70) by $-b_{1(\sqrt{n}+1)}2^{r+1}\sqrt{n}$ and the third term by $-b_{1n}2^{r+1}\sqrt{n} > -2\sqrt{n}b_{1(\sqrt{n}+1)}2^{r+1}$.

We combine everything and the upper bound in Eq.(3.64) should hold for the expression that we obtain. We again use Eq. (3.65).

$$\begin{aligned} b_{1(\sqrt{n}+1)}&\left(\mathcal{S}^+ + \mathcal{S}^- - \sqrt{n}2^{r+3}\right) \\ &< 2\sum_{j=1}^{\lfloor n/2\rfloor+1} b_{1j} < 2\sum_{j=1}^{\lfloor n/2\rfloor+1} b_{1(n-\sqrt{n})} \\ &< 2b_{1(\sqrt{n}+1)}\left(\lfloor n/2\rfloor+1\right)\left(1+\frac{1}{\sqrt{n}}\right) \end{aligned} \tag{3.73}$$

$$\mathcal{S}^+ + \mathcal{S}^- < n + O(\sqrt{n}) + \sqrt{n}2^{r+3} \tag{3.74}$$

Let us rewrite $\sum_{j=1}^{n}(a_{1j}+a_{2j}) = \sum_{j=1}^{n}(b_{1j}+2a_{1j}) = \mathcal{S}^+ + \mathcal{S}^- + \sum_{j=1}^{\sqrt{n}}(b_{1j}+2a_{1j}) + \sum_{j=n-\sqrt{n}+1}^{n}(b_{1j}+2a_{1j})$. Then,

$$\sum_{j=1}^{n}(a_{1j}+a_{2j}) < \left(\mathcal{S}^+ + \mathcal{S}^- + 2\sqrt{n}2^{r+1}\right) \tag{3.75}$$

by $|b_{1j} + 2a_{1j}| < 2^{r+1}$. Let $r < c \log_2 n$ for some constant $0 < c < 1/2$. Clearly, the RHS of Eq.(3.75) is less than $n + O(n^\epsilon)$ where $1/2 < \epsilon < 1$ is a constant.

Similarly, we obtain the corresponding lower bound and use the other necessary and sufficient condition to prove another inequality corresponding to Eq. (3.77). For a constant $1/2 < \epsilon < 1$, we have

$$n - O(n^\epsilon) < \sum_{j=1}^{n} (a_{1j} + a_{2j}) < n + O(n^\epsilon) \tag{3.76}$$

$$n - O(n^\epsilon) < \sum_{j=1}^{n} (a_{3j} + a_{2j}) < n + O(n^\epsilon) \tag{3.77}$$

Subtracting both, we get

$$-O(n^\epsilon) < \sum_{j=1}^{n} a_{3j} - a_{1j} < O(n^\epsilon) \tag{3.78}$$

and therefore, $0 < \frac{1}{2^{2r}} \leq |a_{3j} - a_{1j}| < \frac{O(n^\epsilon)}{n}$ for $i \in \{1, 2\}$ and some $j \in \{1, \ldots, n\}$. Hence, $RES(A) = r = \Omega(\log n)$. □

3.3 From NN Representations to Threshold Circuits

Another important aspect of NN representations is their place in the circuit class hierarchy. Suppose that we are given an m-anchor NN representation of a Boolean function. One can construct a linear threshold circuit of depth-3 with $O(m^2)$ many linear threshold gates (P. Hajnal, Z. Liu, and Turán, 2022; Hansen and V. V. Podolskii, 2015). This circuit transformation helps us relate the circuit complexity lower bounds to NN representation bounds and conversely, given an NN representation, it can help us find depth-3 threshold circuit constructions and upper bounds on the size. However, we cannot obtain an NN representation easily given a Boolean circuit. We try to answer this partially in the next sections.

Theorem 25. *Suppose that $f(X)$ is an n-input Boolean function with an m anchor NN representation with the set of positive (or negative) anchors P (or N). Then, there is a threshold circuit that can compute $f(X)$ in the form* $\mathrm{OR} \circ \mathrm{AND} \circ \mathrm{LT}$ *of size* $|P||N| + |P| + 1$.

Proof. The label of the nearest neighbor can be found by the following formula.

$$\arg\min_i d(a_i, X)^2 = \arg\min_i |X| - 2(AX)_i + ||a_i||_2^2 \tag{3.79}$$

$$= \arg\max_i 2(AX)_i - ||a_i||_2^2 \tag{3.80}$$

where $||.||_2$ denotes the Euclidean norm. Let P (or N) be the set of positive (or negative) anchors $\{a_1, \ldots, a_{|P|}\}$ (or, $\{b_1, \ldots, b_{|N|}\}$.

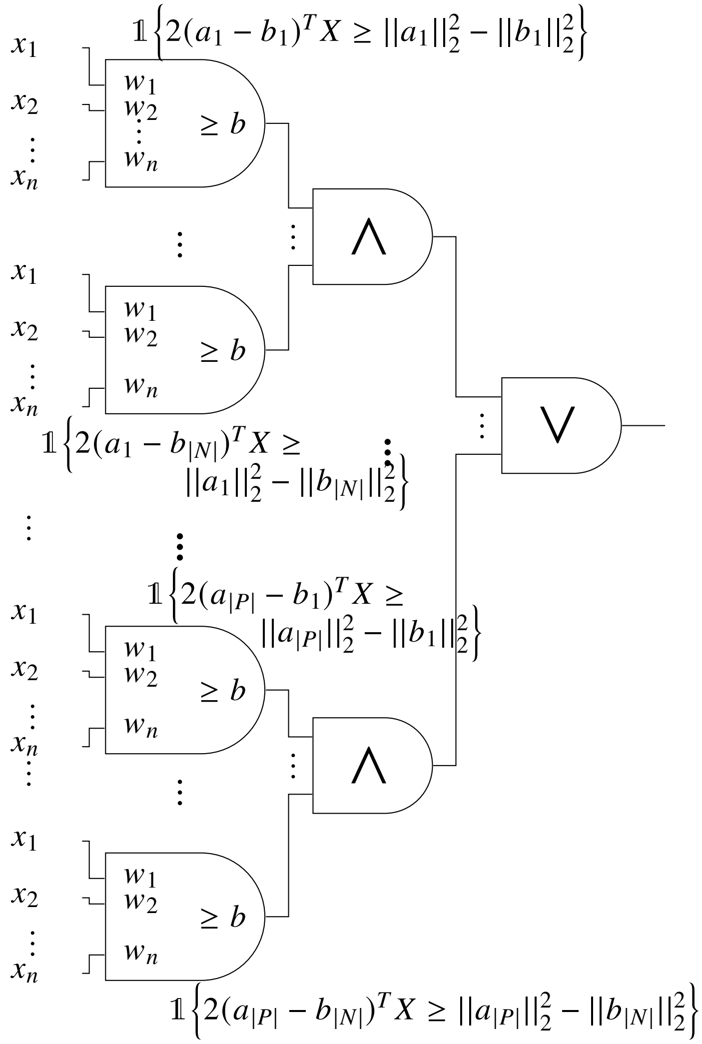


Figure 3.4: A sketch of a transformation from an NN representation to a linear threshold circuit of depth 3. The first layer consists of linear threshold gates while the second and third layer is an AND – OR network. The gates with w_i inside with w_0 as the bias is $\text{LT}(X)$, the gates with $\wedge$ inside is $\text{AND}(X)$, and the gates with $\vee$ inside is $\text{OR}(X)$.

The constructive transformation is given in Fig. 3.4. In the first layer, we simply compare the distances of all individual positive anchors, say a_i, to all negative anchors $\{b_1, \ldots, b_{|N|}\}$ using Eq. (3.79). Hence, there are $|P||N|$ many threshold gates in the first layer. There is an AND gate corresponding to each positive anchor and the top gate is an OR gate, implying that the claimed circuit size is correct.

Assume first that $f(X) = 1$. The linear threshold function

$$\mathbb{1}\left\{2a_i^T X - ||a_i||_2^2 \geq 2b_j^T X - ||b_j||_2^2\right\} \quad (3.81)$$

evaluates 1 for some $i \in \{1, \ldots, |P|\}$ and for all $j \in \{1, \ldots, |N|\}$ because a positive anchor must be closer to X than any negative anchor. Hence, the output of the corresponding i^{th} AND gate will be 1 and consequently, the output of the circuit is 1.

Conversely, assume $f(X) = 0$. Then, there always exist some j such that Eq. (3.81) is 0 for all i because the nearest neighbor to X is negative (i.e. in N). Therefore, all of the second layer AND gates compute 0 so that the output of circuit is 0. □

We remark that the threshold circuit in Figure 3.4 computes the union of different convex polytopes. These polytopes cannot be arbitrary, each depends on a given positive anchor and all negative anchors simultaneously.

3.4 NN Representations of Convex Polytopes ($\mathbf{AND}_m \circ \mathbf{LT}_n$)

The membership function to a convex polytope, i.e., $\mathbb{1}\{AX \leq b\}$ can be thought as an AND∘LT circuit where each row is a linear threshold function. We show that $(m+1)$-size NN representation with a single positive anchor is a membership function of a convex polytope with m half-spaces. And, conversely, given a membership function of a convex polytope with m half-spaces, i.e. $\mathbb{1}\{AX \leq b\}$, one can construct an $(m+1)$-size NN representation with a single positive anchor. This equivalence is in fact a generalization of the equivalence between linear threshold functions and Boolean functions with $NN(f) = 2$.

First direction is easy to see: Given the anchor matrix and anchor labels, we can find optimal separating hyperplanes between them. We provide the proof for the other direction where the geometric idea is given in Figure 3.5. We use the notation $\text{diag}(AA^T) \in \mathbb{Q}^m$ to denote the squared Euclidean norms of each row of a matrix $A \in \mathbb{Q}^{m \times n}$.

Theorem 26. *Let $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$ define a convex polytope in $\mathbb{R}^n$ by the intersection of half-spaces $AX \leq b$. Then, there exists an NN representation with $m+1$ anchors and resolution $O(RES(\text{diag}(AA^T))$.*

Proof. Without loss of generality, we assume that there is an binary interior point in the convex polytope $AX \leq b$. To ensure strict feasibility, we can modify $AX \leq b$

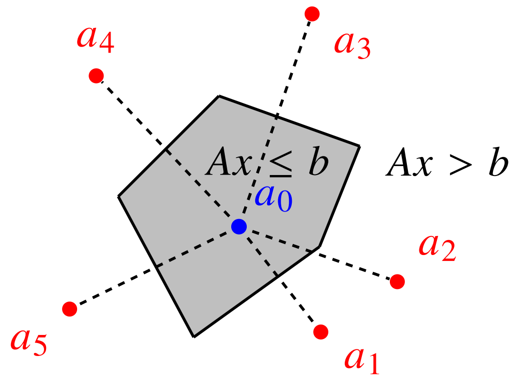


Figure 3.5: A convex polytope defined by the intersection of half-spaces $Ax \leq b$ and its NN representation by the "reflection" argument. The interior of the polytope is closest to a_0 and the exterior is closest to one of $a_1, \ldots, a_5$.

to $AX \leq b + 0.5$ where both inequalities have the same solution sets for $X \in \{0,1\}^n$ and if there is no such binary vector, then $\mathbb{1}\{AX \leq b\}$ is in fact a constant Boolean function with value 0.

The geometrical idea is to compute the reflection of the feasible point $a_0 \in \{0,1\}^n$ with respect to $Ax = b$. We take a_0 a positive anchor and we consider the other anchors negative as follows:

$$a_i = a_0 + 2c_i A_i \tag{3.82}$$

for $i \in \{1, \ldots, m\}$ where A_i denotes the i^{th} row of the matrix A and c_i is a real constant that will be specified later. We also assume that $a_0 + c_i A_i$ is a point on the $A_i^T x = b_i$ to correctly reflect a_0 to a_i. Then, we have $A_i^T(a_0 + c_i A_i) = A_i^T a_0 + c_i ||A_i||_2^2 = b_i$ and therefore $c_i = \frac{b_i - A_i^T a_0}{||A_i||_2^2}$. Implicitly, all $c_i > 0$ because of the feasibility condition.

Note that whenever $A_i^T X = b_i$, the anchors a_i and a_0 are equidistant and we use a classical perturbation argument: $AX \leq b$ has the same solution set as $AX \leq b + 0.5$ for $X \in \{0,1\}^n$.

When we expand the squared Euclidean form $d(a_i, X)^2$, we obtain

$$\begin{aligned} d(a_i, X)^2 = |X| - 2a_0^T X + ||a_0||_2^2 & \\ - 4c_i(A_i^T X - A_i^T a_0) + 4c_i^2 ||A_i||_2^2 & \qquad (3.83) \\ = d(a_0, X)^2 + \frac{4}{||A_i||_2^2}(b_i - A_i^T a_0)(A_i^T a_0 - A_i^T X) & \\ + \frac{4}{||A_i||_2^2}(b_i - A_i^T a_0)^2 & \qquad (3.84) \end{aligned}$$

$$= d(a_0, X)^2 + \frac{4}{||A_i||_2^2}(b_i - A_i^T a_0)(b_i - A_i^T X) \tag{3.85}$$

We have two cases: either $A_i^T X < b_i$ for all $i \in \{1, \dots, m\}$ or $A_i^T X > b$ for some $i \in \{1, \dots, m\}$. To compare the distance of X to a_0 and a_i, we simplify the expression to

$$d(a_i, X)^2 - d(a_0, X)^2 = \frac{4}{||A_i||_2^2}(b_i - A_i^T a_0)(b_i - A_i^T X) \tag{3.86}$$

<u>Case 1:</u> If $A_i^T X < b_i$ for all $i \in \{1, \dots, m\}$, we need $d(a_i, X)^2 - d(a_0, X)^2 > 0$. Then, since $b_i - A_i^T a_0 > 0$ by definition, the RHS of Eq. (3.86) is greater than 0 if and only if $A_i^T X < b_i$ for all $i \in \{1, \dots, m\}$.

<u>Case 2:</u> If $A_i^T X > b_i$ for some $i \in \{1, \dots, m\}$, we need $d(a_i, X)^2 - d(a_0, X)^2 < 0$ for such i. Then, since $b_i - A_i^T a_0 > 0$ by definition, the RHS of Eq. (3.86) is less than 0 if and only if $A_i^T X > b_i$.

For $A_i^T X > b_i$, we do not care which anchor is closest to x among a_i for $i \in \{1, \dots, m\}$ since they all have the same labeling. Therefore, the proposed scheme is indeed an NN representation for the convex polytope defined by $AX \leq b$.

We now prove the resolution bound. For $c_i = \frac{b_i - A_i^T a_0}{||A_i||_2^2}$, we see that $b_i - A_i^T a_0 \leq 2||A_i||_2^2$ loosely. We assume that the bias term can be at most $||A_i||_2^2$ otherwise the threshold function is trivial. Then, the resolution of the $a_0 + 2c_i A_i$ is $O(RES(\text{diag}(AA^T))$ and the claim holds by considering all Euclidean norms. □

We note that by the selection of anchors provided in Theorem 26, if we construct the circuit in Figure 3.4, we obtain exactly a circuit deciding if $AX \leq b$ by seeing that $a_i - a_0 = 2c_i A_i$ and $||a_i||_2^2 - ||a_0||_2^2 = 4c_i A_i^T(a_0 + c_i A_i) = 4c_i b_i$. Hence, $\mathbb{1}\{2(a_i - a_0)^T X \leq ||a_i||_2^2 - ||a_0||_2^2\} = \mathbb{1}\{A_i^T X \leq b_i\}$ (recall that $c_i > 0$ for all $i \in \{1, \dots, m\}$). Since there is a single positive anchor, the top OR gate is unnecessary.

It is also possible to replace linear threshold functions with exact threshold functions in that the AND of two linear threshold functions $\mathbb{1}\{A_i^T X \leq b_i\}$ and $\mathbb{1}\{A_i^T X \geq b_i\}$ implies $\mathbb{1}\{A_i^T X = b_i\}$. This gives us a way to construct NN representations for $\mathbb{1}\{AX = b\}$ where each row corresponds to an exact threshold function, i.e., an AND $\circ$ ELT circuit. We can again perturb the bias terms by 0.5 and obtain $\mathbb{1}\{A_i^T X = b_i\}$ as the AND of $\mathbb{1}\{A_i^T X \leq b_i + 0.5\}$ and $\mathbb{1}\{A_i^T X \geq b_i - 0.5\}$ so that there is always an interior point $a_0 \in \{0, 1\}^n$ in the polytope. Otherwise, there is no binary interior

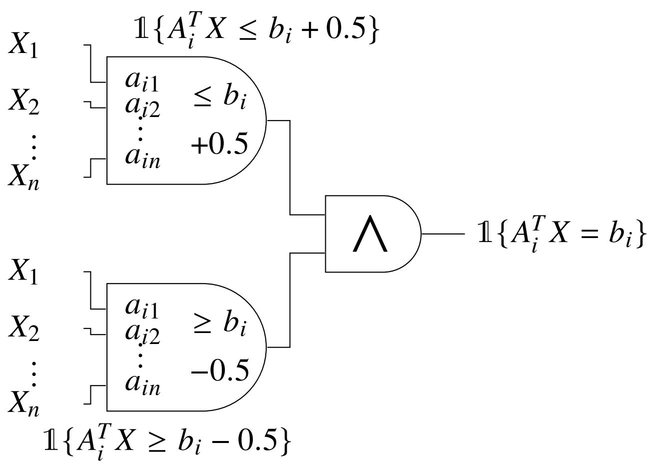


Figure 3.6: The subcircuit for the $\mathbb{1}\{A_i^T X = b_i\} = \mathbb{1}\{A_i^T X \leq b_i + 0.5\} \wedge \mathbb{1}\{A_i^T X \geq b_i - 0.5\}$ for binary inputs $X \in \{0,1\}^n$.

point and the function is the constant function $f(X) = 0$ with a trivial 1-anchor NN representation.

Theorem 27. *For a system of linear equations $Ax = b$ where $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$, there exist an NN representation with $2m + 1$ anchors and resolution $O(RES(\mathrm{diag}(AA^T))$ checking if $AX = b$ or not for $X \in \{0,1\}^n$.*

An interesting fact for this case is that AND $\circ$ ELT = ELT, i.e., the class of exact threshold functions is closed under AND operation (Hansen and V. V. Podolskii, 2010). This means that the bounded solutions of $Ax = b$ in integers can be verified by a single exact threshold function with an exponential blow-up in the number of equations m for the weight sizes. Consequently, any system of linear equations with binary solutions can be represented by a 3-anchors NN representation with "high" resolution.

Theorem 28. *For a system of linear equations $Ax = b$ where $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$, there exist an NN representation with 3 anchors and resolution $O(m \log n + mRES(A))$ checking if $AX = b$ or not for $X \in \{0,1\}^n$.*

Proof. We follow the steps in the proof of Proposition 6 in (Hansen and V. V. Podolskii, 2010). Let $|a_{ij}| \leq W$ for some $W \in \mathbb{Z}$ for all i, j. Then, $|A_i^T X - b_i| \leq 2nW$ (we assume that b_i is bounded by nW). The exact threshold function $\mathbb{1}\left\{ \sum_{j=1}^m (4nW)^{j-1}(A_i^T X - b_i) = 0 \right\}$ evaluates 1 if and only if $AX = b$. The maximum weight in this exact threshold function is $W(4nW)^m$ and the resolution becomes $O(m \log n + mRES(A))$ by Theorem 19. $\square$

We finally remark that the NN representation idea for convex polytopes can apply to NN representations whose input vectors are from an alphabet larger than the binary $\{0,1\}$. This is true as long as the interior point of $Ax \leq b$ is not empty for $x \in \mathbb{R}^n$ and consequently, the existence of a feasible point is guaranteed after perturbing the bias terms by a sufficiently small amount.

Lemma 11 (Maass, 1993)**.** *Consider a system $Ax \leq b$ where $A \in \mathbb{Z}^{m\times n}$ and $b \in \mathbb{Z}^m$ with entries bounded in absolute value by $W \in \mathbb{Z}$. If this system has a solution in $\mathbb{R}^n$, then it also has a solution in the form $x = (s_1/t, \dots, s_m/t)$ where $s_1, \dots, s_m, t$ are integers of absolute value $\leq (2n+1)!W^{2n+1}$.*

3.5 NN Representations of Depth-2 Threshold Circuits with Symmetric Top Gate (SYM $\circ$ ELT and SYM $\circ$ LT)

For AND $\circ$ LT or AND $\circ$ ELT, what happens if we replace the AND with the OR gate? For OR $\circ$ LT, the answer is easy because NN representations are closed under complement operation (as we can revert the labeling of anchors) and the complement of OR $\circ$ LT is AND $\circ$ LT, therefore, we already have a solution by Theorem 26. However, for OR $\circ$ ELT, we cannot do the same as the complement of an exact threshold function needs not to be exact. The set of functions in OR $\circ$ ELT is also interesting from another complexity perspective. This class of circuits computes the Boolean functions with *Exact Cover Representations*. Since we do not treat Exact Cover Representations in a deep way, we summarize important details in Appendix C with some conjectural relationship to NN representations.

Obtaining a construction for OR $\circ$ ELT is not straight-forward and the arbitrary case is still unresolved. We define the following set of *regularity conditions* which allows us to explicitly construct NN representations for many depth-2 circuits. Let $W \in \mathbb{Z}^{m\times n}$ be the weight matrix of the first layer and $b \in \mathbb{Z}^n$ be the bias vector of the first layer.

1. The weights of each gate has the same norm. $||W_i||_2^2 = ||W_j||_2^2$ for all i, j.
2. The weights of each gate are mutually orthogonal, $W_i^T W_j = 0$ for all $i \neq j$.
3. There exists an $X^* \in \{0,1\}^n$ such that $WX^* = b$.

The regularity conditions hurt the generality but the result is still very useful and applicable to many functions. For example, if all gates have *disjoint* inputs and have the same norm for their weights, then all conditions are satisfied.

Theorem 29. *Suppose that there is an n-input Boolean function $f(X)$ such that $f(X) \in \text{SYM} \circ \text{ELT}$ obeying the regularity conditions with m many gates in the first layer. Let $g(Z)$ be the top symmetric Boolean function where $Z \in \{0,1\}^m$.*

There exists an NN representation of $f(X)$ with $\sum_{t \in \mathcal{T}} \binom{m}{m-t} 2^{m-t}$ many anchors where $\mathcal{T} = \{I_0 + 1, I_1 + 1, \dots, I_{I(g)-1} + 1\}$ contains the left interval boundaries for the top symmetric gate $g(Z)$. The resolution of the construction is $O(\log m + RES(\text{diag}(WW^T)))$.

Proof. The anchors we construct are as follows.

$$a_{jk}^{(t)} = X^* + d u_{jk}^{(t)} \tag{3.87}$$

We will design $u_{jk}^{(t)}$ and $d \in \mathbb{Z}$ will be picked later. $t \in \mathbb{Z}$ is the *type* of the anchor and j, k are indices to be determined later. We denote w_i to be the i^{th} row of the weight matrix W.

The squared Euclidean distances will be

$$\begin{aligned} d(a_{jk}^{(t)}, X)^2 = |X| - 2X^T X^* + ||X^*||_2^2 \\ - 2d u_{jk}^{(t)}(X - X^*) + d^2 ||u_{jk}^{(t)}||_2^2 \end{aligned} \tag{3.88}$$

Since $|X| - 2X^T X^* + ||X^*||_2^2$ is the same for all anchors, we do not care its value when we compare distances. Now, we pick $u_{jk}^{(t)}$ as the all plus-minus combinations of the $m - t$ combinations of all m weight vectors in W for any $t \in \{0, \dots, m\}$. That is, there are $\binom{m}{m-t}$ selections of weight vectors for each t for $u_{jk}^{(t)}$. We also use an abuse of notation here: $\pm w_1 \pm w_2$ is a compact form to denote all $\{w_1 + w_2, w_1 - w_2, -w_1 + w_2, -w_1 - w_2\}$. We can write all $u_{jk}^{(t)}$s as follows.

$$u_{jk}^{(0)} \in \Big\{ \pm w_1 \pm w_2 \pm \cdots \pm w_m \Big\} \tag{3.89}$$

$$u_{jk}^{(1)} \in \Big\{ \pm w_1 \pm \cdots \pm w_{m-1}, \dots, \pm w_2 \pm \cdots \pm w_m \Big\} \tag{3.90}$$

$$\vdots$$

$$u_{jk}^{(m-1)} \in \Big\{ \pm w_1, \pm w_2, \dots, \pm w_m \Big\} \tag{3.91}$$

$$u_{jk}^{(m)} \in \emptyset \tag{3.92}$$

Now, we define $u_{jk}^{(t)}$s more precisely. Let $j_1, \dots, j_{m-t}$ be the binary expansion of $(j-1) \in \mathbb{Z}$ with $\{0,1\}$ entries. The index j denotes the unique sign pattern for ws in $u_{jk}^{(t)}$.

For the anchor type t, we define the family of index sets $\mathcal{F}^{(t)} = \binom{[m]}{m-t}$. Alternatively, $\mathcal{F}^{(t)} = (\mathcal{I}_k^{(t)})_{k \in K^{(t)}}$. Here $\mathcal{I}_k^{(t)}$ is an index set with size $\binom{m}{m-t}$ from the elements $\{1, \dots, m\}$ and $K^{(t)} = \{1, \dots, \binom{m}{m-t}\}$. In other words, for $u_{jk}^{(t)}$s, the index k denotes a specific $m-t$ selection out of $\{1, \dots, m\}$. We thus obtain

$$u_{jk}^{(t)} = \sum_{i=1}^{m-t} (-1)^{j_i} w_{\mathcal{I}_{k\ i}^{(t)}} \text{ for } t < m \tag{3.93}$$

$$u_{jk}^{(m)} = 0 \tag{3.94}$$

For instance, consider $m = 7$, $t = 4$, $(j-1) = 2$, and $k = 6$. First, we enumerate all $\binom{7}{3}$ combinations in an order to find the correct selection of weights.

$$\Big\{\{w_1, w_2, w_3\}, \{w_1, w_2, w_4\}, \{w_1, w_2, w_5\}, \{w_1, w_2, w_6\}, \\ \{w_1, w_2, w_7\}, \{w_1, w_3, w_4\}, \dots, \{w_5, w_6, w_7\}\Big\} \tag{3.95}$$

The corresponding index sets are the subscripts of the weights, e.g., $\mathcal{I}_4^{(4)} = \{1, 2, 6\}$. For $k = 6$, we pick $\mathcal{I}_6^{(4)} = \{1, 3, 4\}$. Since $j - 1$ has the binary expansion for $(j_1, j_2, j_3) = (0, 1, 0)$, we obtain the sign pattern $(1, -1, 1)$, implying that we have $u_{36}^{(4)} = w_1 - w_3 + w_4$.

Let $g(Z)$ be the symmetric Boolean function for the top gate of the circuit with $I(g)$ many intervals where $Z \in \{0,1\}^m$. In the construction, we only include $u_{jk}^{(t)}$ for $t \in \mathcal{T} = \{I_0 + 1, I_1 + 1, \dots, I_{I(g)-1} + 1\}$.

Claim: If Z is in the l^{th} interval of $g(Z)$, i.e., $|Z| \in [I_{l-1} + 1, I_l]$, then the closest anchor is of type $t = I_{l-1} + 1$. That is,

$$I_{l-1} + 1 = \arg\min_t -2d u_{jk}^{(t)^T} (X - X^*) + d^2 ||u_{jk}^{(t)}||_2^2 \text{ where} \\ w_i^T X = b_i \text{ for } |Z| \text{ many } i\text{s from } \{0, 1, \dots, m\} \tag{3.96}$$

If the claim is true, then depending on the value $g(Z)$ for that interval, we label all the corresponding type of anchors to **0** or 1 and the construction follows.

To prove the claim, we first pick $d = \frac{1}{m||w||_2^2}$ where $||w||_2^2 = ||w_1||_2^2 = \cdots = ||w_m||_2^2$ and by orthogonality, we have $||u_{jk}^{(t)}||_2^2 = (m-t)||w||_2^2$. Then, from Eq. (3.88), we obtain

$$
\begin{aligned}
&-2du_{jk}^{(t)^T}(X-X^*)+d^2||u_{jk}^{(t)}||_2^2 \\
&\quad=\frac{-2}{m||w||_2^2}\left(u_{jk}^{(t)^T}(X-X^*)-0.5\frac{(m-t)}{m}\right) \qquad (3.97)
\end{aligned}
$$

Suppose that $u_{jk}^{(t)}(X-X^*) > u_{jk}^{(v)}(X-X^*)$ for any $v \in \{0,\ldots,m\}$. Then, recalling that $X^* \in \{0,1\}^n$,

$$
\begin{aligned}
&\left(u_{jk}^{(t)^T}(X-X^*)-0.5\frac{(m-t)}{m}\right) && (3.98)\\
&\quad\geq\left(u_{jk}^{(v)^T}(X-X^*)+1-0.5\frac{(m-t)}{m}\right) && (3.99)\\
&\quad\geq\left(u_{jk}^{(v)^T}(X-X^*)+0.5\right) && (3.100)\\
&\quad>\left(u_{jk}^{(v)^T}(X-X^*)\right) && (3.101)\\
&\quad>\left(u_{jk}^{(v)^T}(X-X^*)-0.5\frac{(m-v)}{m}\right) && (3.102)
\end{aligned}
$$

and thus, the t value that minimizes Eq. (3.96) always have the largest $u_{jk}^{(t)}(X-X^*)$ value.

If $u_{jk}^{(t)}(X-X^*)=u_{jk}^{(v)}(X-X^*)$ for some v, then

$$
\begin{aligned}
&\left(u_{jk}^{(t)^T}(X-X^*)-0.5\frac{(m-t)}{m}\right) && (3.103)\\
&\quad>\left(u_{jk}^{(v)^T}(X-X^*)-0.5\frac{(m-v)}{m}\right) && (3.104)
\end{aligned}
$$

only if $t > v$. Therefore, the t value that minimizes Eq. (3.96) is the maximum among the t values maximizing $u_{jk}^{(t)}(X-X^*)$.

Expanding $u_{jk}^{(t)}(X-X^*)$ using Eq. (3.93), we obtain

$$
u_{jk}^{(t)}(X-X^*)=\sum_{i=1}^{m-t}(-1)^{j_i}(w_{\mathcal{I}_{k\ i}^{(t)}}^T X - b_{\mathcal{I}_{k\ i}^{(t)}}) \qquad (3.105)
$$

Given the value of $|Z|$ where $z_i = \mathbb{1}\{w_i^T X = b_i\}$ for $i \in \{1,\ldots,m\}$, let $\mathcal{I}$ be the index set where $w_i^T X \neq b_i$. It is actually a selection of $m-|Z|$ out of m values and therefore, $\mathcal{I}=\mathcal{I}_k^{(|Z|)}$ for a combination enumerated by some k. It can be seen that $u_{jk}^{(t)}(X-X^*)$ is maximized if $\mathcal{I}_k^{(|Z|)}=\mathcal{I}$ and whether $w_i^T X < b_i$ or $w_i^T X > b_i$ is of no importance since there is a pair j,k such that

$$
\max_{j,k,t} u_{jk}^{(t)^T}(X-X^*)=\sum_{i\in\mathcal{I}}|w_i^T X - b_i| \qquad (3.106)
$$

The optimal t value is less than or equal to $|Z|$ because $\mathcal{I}$ contains $m - |Z|$ values. Any superset for the index set $\mathcal{I}$ will include $w_i^T X = b_i$ for some i and the value of the dot product $u_{j'k'}^T (X - X^*)$ cannot increase. Mathematically, for the optimal j, k, j', k' choices,

$$
\begin{aligned}
u_{jk}^{(p)^T} (X - X^*) &= u_{j'k'}^{(q)\ T} (X - X^*) \\
&\quad \text{for } p, q \in \{0, \ldots, I_{l-1} + 1\} && (3.107) \\
u_{jk}^{(p)^T} (X - X^*) &> u_{j'k'}^{(q)\ T} (X - X^*) \\
&\quad \text{for } p \in \{0, \ldots, I_{l-1} + 1\} \text{ and} \\
&\qquad q \in \{I_l + 1, \ldots, I_{I(g)-1} + 1\} && (3.108)
\end{aligned}
$$

By our previous observation about the maximal t, we conclude that $t = I_{l-1} + 1$. This proves the claim and hence, completes the validation of the construction. The number of anchors is $\sum_{t \in \mathcal{T}} \binom{m}{m-t} 2^{m-t}$, which is easy to verify.

For the resolution, we see that $\frac{1}{m||w||_2^2}(w_1 + \cdots + w_m)$. The maximum value for the numerator can be $m||w||_2^2$ and it holds that $RES(A) = O(\log m + RES(\text{diag}(WW^T)))$.

□

Theorem 29 is powerful since it provides constructions to many important functions in Circuit Complexity Theory. Hence, it is an important milestone to find NN representations with size and resolution upper bounds.

Corollary 29.1. *Let $f(X)$ be an $2mn$-input Boolean function* $\text{SYM}_m \circ \text{EQ}_{2n}$ *where there are m many disjoint n-input* EQ *functions in the first layer. Then, we obtain the following table of results.*

Function	**NN Representation**	
	Size	Resolution
$\text{AND}_m \circ \text{EQ}_{2n}$	$2m + 1$	$O(n)$
$\text{OR}_m \circ \text{EQ}_{2n}$	$(m + 2)2^{m-1}$	$O(n)$
$\text{PARITY}_m \circ \text{EQ}_{2n}$	3^m	$O(n)$

First of all, the $\text{AND}_m \circ \text{EQ}_{2n}$ result also follows from Theorem 27 and it holds that Theorem 29 is a generalization of it in the case of same norm weights. For $\text{OR} \circ \text{EQ}$, we see that $\mathcal{T} = \{0, 1\}$ and for $\text{PARITY} \circ \text{EQ}$, $\mathcal{T} = \{0, 1, 2, \ldots, m\}$ where the representation size formula corresponds to the binomial expansion of $(1 + 2)^m$.

Hence, changing the top gate could greatly change the number of anchors in the NN representation.

It is remarkable that not only the number of intervals for the top symmetric Boolean function is significant, but also the location of the intervals changes the size of the construction. OR and AND are symmetric linear threshold functions (i.e., $I(f) = 2$), however, their composition with EQ gives completely different sizes (one is linear in m, the other is exponential in m) of NN representations.

We also obtain a construction for SYM $\circ$ LT. Compared to the Theorem 29, the difference is in the set of intervals boundaries where Theorem 30 includes both boundaries. In contrast, because each linear threshold function requires 2 anchors instead of 3 as for exact threshold functions, we do not see an extra exponential term 2^{m-t} in the binomial sum.

Theorem 30. *Suppose that there is an n-input Boolean function $f(X)$ such that $f(X) \in \text{SYM} \circ \text{LT}$ obeying the regularity conditions with m many gates in the first layer. Let $g(Z)$ be the top symmetric Boolean function where $Z \in \{0,1\}^m$.*

There exists an NN representation of $f(X)$ with $\sum_{t \in \mathcal{T}} \binom{m}{m-t}$ many anchors where $\mathcal{T} = \{I_1, I_1 + 1, \ldots, I_{I(g)-1}, I_{I(g)-1} + 1\}$ contains the all the interval boundaries for the top symmetric gate $g(Z)$. The resolution of the construction is $O(\log m + RES(\text{diag}(WW^T)))$.

Proof. The proof is essentially the same as the proof of Theorem 29 except we do not have j parameter in the anchor construction. All sign patterns are fixed to -1s. Then, we have

$$a_k^{(t)} = X^* + du_k^{(t)} \tag{3.109}$$

$$u_k^{(t)} = \sum_{i=1}^{m-t} -w_{\mathcal{I}_k^{(t)}{}_i} \text{ for t < m} \tag{3.110}$$

$$u_k^{(m)} = 0 \tag{3.111}$$

for the same definition of the family of index sets. We again pick $d = \frac{1}{m||w||_2^2}$.

We also assume that the bias vector b is replaced with $b - 0.5$ to ensure the equality cases do not appear for the linear threshold functions in the first layer. In this case, we need to be careful about the choice of X^* since there is no $WX = b - 0.5$ such that $X \in \{0,1\}^n$. We remedy this by using the first two regularity conditions so that X^* in the third regularity condition is replaced by $X^* - \frac{1}{2||w||_2^2}(w_1 + \cdots + w_m)$ where $||w||_2^2$

is the norm of a row of W. This does not increase the resolution asymptotically at the end since $RES(X^*) = RES(\text{diag}(WW^T)$.

Given the value of $|Z|$ where $z_i = \mathbb{1}\{w_i^T X > b_i\}$ for $i \in \{1, \dots, m\}$ and $|Z| \in [I_{l-1} + 1, I_l]$, let $\mathcal{I}$ be the index set $w_i^T X < b_i$. It is actually a selection of $m - |Z|$ out of m values and $u_k^{(t)}(X - X^*)$ is maximized if $\mathcal{I}_k^{(|Z|)} = \mathcal{I}$.

$$\max_{k,t} u_k^{(t)^T}(X - X^*) = \sum_{i \in \mathcal{I}} (b_i - w_i^T X) \tag{3.112}$$

For the optimal k, k' choices, we have

$$u_k^{(p)^T}(X - X^*) < u_{k'}^{(p+1)^T}(X - X^*)$$
$$\text{for } p \in \{0, \dots, |Z| - 1\} \tag{3.113}$$
$$u_k^{(p)^T}(X - X^*) > u_{k'}^{(p+1)^T}(X - X^*)$$
$$\text{for } p \in \{|Z|, \dots, m\} \tag{3.114}$$

Since $|Z| \in [I_{l-1} + 1, I_l]$, the optimal t value will either be $I_{l-1} + 1$ or I_l. Since we include both interval boundaries in this construction and both types $t = I_{l-1} + 1$ and $t = I_l$ have the same label, the construction follows. The resolution is similar to the previous proof and the increase in the resolution of X^* makes no difference asymptotically. □

It is an open question if the interval boundaries can be reduced only to left (or right) boundaries in Theorem 30. For the cases where the top gate is symmetric linear threshold functions or the PARITY, this simplification do not matter.

Corollary 30.1. *Let $f(X)$ be an mn-input Boolean function* $\text{SYM}_m \circ \text{LT}_n$ *where there are m many disjoint n-input* LT *functions in the first layer with same norm weight vectors. Then, we obtain the following table of results.*

Function	**NN Representation**	
	Size	Resolution
$\text{AND}_m \circ \text{LT}_n$	$m + 1$	$O(n \log n)$
$\text{OR}_m \circ \text{LT}_n$	$m + 1$	$O(n \log n)$
$\text{PARITY}_m \circ \text{LT}_n$	2^m	$O(n \log n)$

Contrary to the drastic difference in the number of anchors for OR ∘ EQ and AND ∘ EQ, there is no difference in the number of anchors for OR ∘ COMP and AND ∘

COMP, which is expected because these circuits can be thought as the complement of each other loosely speaking (the complement of $\mathbb{1}\{X \geq Y\}$ is $\mathbb{1}\{Y > X\}$).

As a sanity check, by using Theorem 30, we can construct basic BNN representations for any symmetric Boolean function by using $\mathbb{1}\{x_i \geq 1\}$ for $i \in \{1, \ldots, n\}$ where there are n many linear threshold functions in the first layer (this can be though as 1-input AND function). For instance, the BNN construction for PARITY where all $\{0,1\}^n$ is included can be constructed by using $\text{PARITY}_n = \text{PARITY}_n \circ \text{AND}_1$ and applying Theorem 30. This type of constructions effectively comprises all $X \in \{0,1\}^n$ vectors around boundaries as anchors, which is not optimal in general (consider MAJ in Theorem 18).

It is remarkable that for the IP2_{2n} function, Corollary 30.1 provides a construction with 2^n anchors and constant resolution since $\text{IP2}_{2n} = \text{PARITY}_n \circ \text{AND}_2 \in \text{SYM} \circ \text{LT}$. This is far from the lower bound where there is no explicit resolution bound. For constant resolution, the construction we provide could be optimal. We give an NN representation for IP2_6 in Appendix E.3.

We finally provide a construction for the circuits of SYM $\circ$ SYM where the weights of the first layer are mutually orthogonal. The idea is based on a brute-force minimization argument by applying derivative test.

Theorem 31. *Suppose that there is an n-input Boolean function $f(X)$ in a circuit where the first layer has m many n input perceptrons with a weight matrix $W \in \mathbb{Z}^{m \times n}$ where $|W_{ij}| \leq B$ for some integer B. Also, let rows of the matrix W be mutually orthogonal.*

Then, there exist an NN representation with $(2nB+1)^m$ anchors with resolution $O(\log nB)$.

Proof. This construction can be written as a single matrix product in the most abstract linear form. Let $A \in \mathbb{Z}^{M \times n}$ anchor matrix and $C \in \mathbb{Z}^{M \times m}$ be a coefficient matrix to be determined later. M is the number of anchors.

$$A = CW \tag{3.115}$$

$$a_i = \sum_{j=1}^{m} c_{ij} w_i \tag{3.116}$$

When we extend the squared Euclidean distance, we get

$$d(a_i, X)^2 = |X| - 2\sum_{j=1}^{m} c_{ij} w_j^T X + \sum_{j=1}^{m} c_{ij}^2 ||w_j||^2 \tag{3.117}$$

since we assume $w_i^T w_j = 0$ for all $i \neq j$.

By taking the partial derivatives with respect to c_{ij}, the distance is minimized if $c_{ij} = \frac{w_j^T X}{||w_j||^2}$. To see that, we apply the second derivative test where the Hessian matrix is diagonal with positive entries. The i values correspond to the values in the range of $w_j^T X$, which is bounded by nB, and there are $(2nB+1)^m$ mappings for WX and hence $M = (2nB+1)^m$. We label each anchor according to the output of the function. The resolution bound is easy to verify. □

Theorem 31 is an almost trivial construction which can interestingly achieve optimality in some cases.

Corollary 31.1. *For the n-input PARITY function where* $\text{PARITY}_n = \text{PARITY}_{n/m} \circ \text{PARITY}_m$*, there are NN representations* $(m+1)^{n/m}$ *size and resolution* $\lceil \log(m+1) \rceil$.

Corollary 31.1 follows by the observation that $\text{PARITY}_n = \text{PARITY}_{n/m} \circ \text{PARITY}_m$. For $m = 1$, we obtain the optimal BNN construction and for $m = n$, we have the optimal size NN construction with $O(\log m)$ resolution. For $k = 2$, we have $(\sqrt{3})^n$ many anchors. In general, for constant resolution, the size is super-polynomial, which we conjecture to be optimal.

Since $\text{PARITY}_2 = \text{XOR}$, we can use $\text{PARITY}_n = \text{PARITY}_{n/2} \circ \text{XOR}$. Because $\text{XOR}(X_1, X_2) = \mathbb{1}\{X_1 + X_2 = 1\}$ is an exact threshold function, we can apply Theorem 29 to prove the same upper bounds on size and resolution. Both constructions are equivalent.

Conjecture 3. *For n-input PARITY function, the number of anchors is* $2^{\Omega(n)}$ *if the resolution of the representation is* $O(1)$.

It seems that the conjecture could be true for any symmetric Boolean functions with $I(f) \geq 3$. We know that it is false when $I(f) = 2$ (see Corollary 18.1).

3.6 NN Representations of Linear and Exact Decision Lists

A linear decision list of length m, denoted by LDL, is a sequential list of linear threshold functions $f_1(X), \ldots, f_m(X)$ where the output is $z_k \in \{0, 1\}$ for $f_i(X) = 0$

for $i < k$ and $f_k(X) = 1$ and it is $z_{m+1} \in \{0, 1\}$ if all $f_i(X) = 0$. In other words, the output labels depends only on the first positively evaluated linear threshold function. Therefore, the main characteristic of decision lists is the *domination principle* where the threshold gates in the higher location will determine the output independent of what the lower level threshold gate outputs are. We design the location of the anchors based on this observation. The geometric approach to find an NN representation for LDLs is shown in Figure 3.7.

Theorem 32. *Suppose that an n-input Linear Decision List $l(X)$ of depth m is given under regularity conditions with a weight matrix $W \in \mathbb{R}^{m \times n}$. Then, there is an NN representation for $l(X)$ with $m + 1$ anchors and resolution $O(mRES(\text{diag}(WW^T)))$.*

Proof. For linear threshold gates, the third regularity condition contradicts the desirable property that there should be no $X \in \{0, 1\}^n$ on the hyperplanes themselves. We will describe a way to solve this issue by modifying this condition as it is done in the proof of Theorem 30.

We previously perturbed the bias terms by 0.5 to ensure that there is no binary vectors on the hyperplanes. Alternatively, this can be done by seeing that $\mathbb{1}\{w^T X \geq b\}$ is equivalent to $\mathbb{1}\{2w^T X \geq 2b - 1\}$. Although both methods are equivalent, this change makes the steps in the proof cleaner.

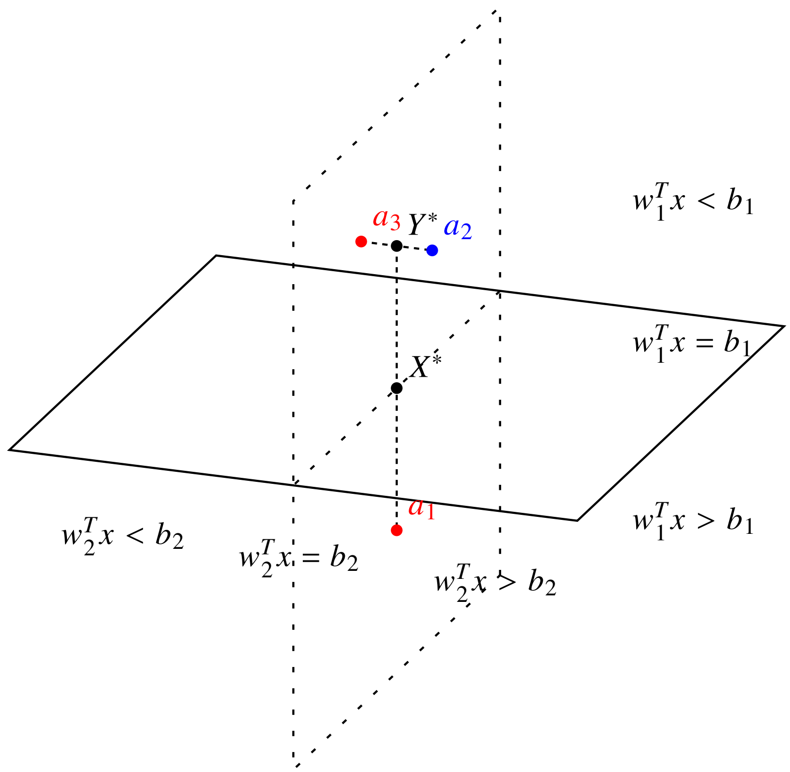
Given that there is a binary vector $X^* \in \{0, 1\}^n$ such that $WX^* = b$, we can find $X' = X^* - \frac{1}{2||w||_2^2}(w_1 + \cdots + w_m)$ so that $2WX' = 2b - 1$ where $||w||_2^2$ is the norm of the original weights. In this case, the resolution of the weights increases by 1 bit because of the doubling and the resolution of X^* is at most $O(RES(||w||_2^2)$.

Therefore, without loss of generality, we can modify the third regularity condition in the following way: There exists $X^* \in \mathbb{Q}^n$ such that $WX^* = b$ with "sufficiently" small resolution and there is no binary $X \in \{0, 1\}^n$ such that $WX = b$.

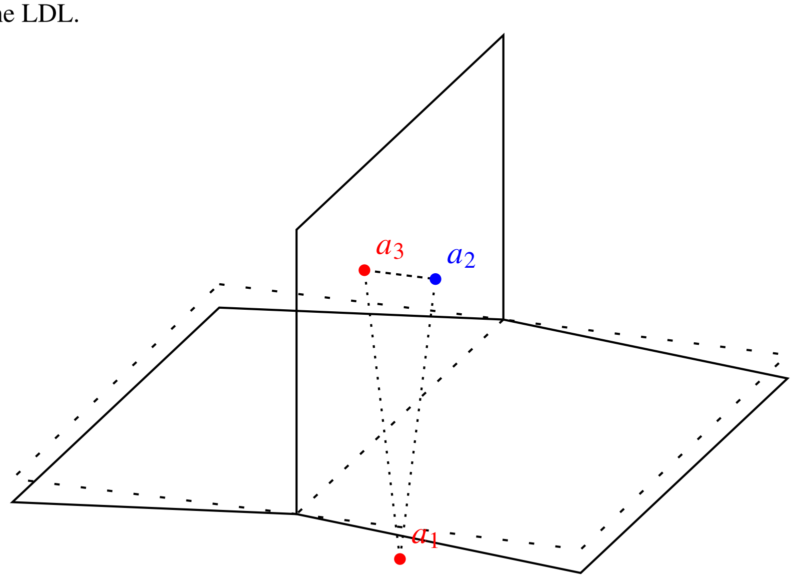
To imitate the linear decision list and obtain the domination principle, we construct the anchors as follows where $c_i = \left(\frac{1}{2||w||_2^2}\right)^i$.

$$a_i = X^* - \sum_{j<i} c_j w_j + c_i w_i \text{ for } i = 1, \ldots, m \tag{3.118}$$

$$a_{m+1} = X^* - \sum_{j<m} c_j w_j - c_m w_m \tag{3.119}$$



(a) Anchor placement idea for the NN Representation for an LDL of depth 2. Each anchor takes care of a leaf of the LDL.



(b) The approximate decision regions for the NN representation. The closer a_2 and a_3 are to each other, the better the bottom region will approximate a half-space.

Figure 3.7: Anchor placement idea for the NN Representation for an LDL of depth 2. In this example, the labels of the anchors are arbitrary.

The labeling of a_i directly corresponds to the labels of z_i for the decision list. We claim that for any k, if the location of the leading one in the decision list is k, i.e.,

$(w_1^T X < b_1, \ldots, w_{k-1}^T X < b_{k-1}, w_k^T X > b_k, \times, \ldots, \times)$ with $\times$ being *don't cares*, then a_k is the closest to the input vector. Hence, the following two conditions are necessary and sufficient. Roughly, the first condition states that if the output is 1, a_k dominates all the rest and the second condition states that if the output is 0, one should proceed to the next anchor.

$$w_k^T X > b_k \Rightarrow d(a_k, X)^2 - d(a_l, X)^2 < 0 \;\; \forall k < l \tag{3.120}$$

$$\begin{aligned} w_k^T X < b_k \Rightarrow d(a_k, X)^2 - d(a_{k+1}, X)^2 > 0 \\ \forall k \in \{1, \ldots, m\} \end{aligned} \tag{3.121}$$

Using the orthogonality of w_is, the squared Euclidean distances can be written as

$$\begin{aligned} d(a_i, X)^2 = |X| - 2X^T X^* + ||X^*||_2^2 \\ + 2\sum_{j<i} c_j (w_j^T X - b_j) + ||w||_2^2 \sum_{j<i} c_j^2 \\ - 2c_i (w_i^T X - b_i) + c_i^2 ||w||_2^2 \end{aligned} \tag{3.122}$$

For the condition in Eq. (3.120), we obtain the following.

$$\begin{aligned} d(a_k, X)^2 - d(a_l, X)^2 = -4c_k (w_k^T X - b_k) \\ - 2\sum_{j=k+1}^{l-1} c_j (w_j^T X - b_j) \\ + 2c_l (w_l^T X - b_l) - ||w||_2^2 \sum_{j=k+1}^{l} c_j^2 < 0 \end{aligned} \tag{3.123}$$

We optimize this inequality in an adversarial sense where the contribution of the negative terms are smallest and of the positive terms are largest possible. Note that we bound $|w_j^T (X - X^*)| \leq 2||w||_2^2$ loosely. We see that $w_k^T X - b_k = 1$ and $w_j^T X - b_j = -2||w||_2^2$ for $j > k$ gives the tightest bounds. $j = m$ or $j = m + 1$ does not matter. Then, putting the values of cs, we get

$$\begin{aligned} d(a_k, X)^2 - d(a_l, X)^2 = -\frac{4}{2^k ||w||_2^{2k}} \\ + \sum_{j=k+1}^{l} \frac{2}{(2||w||_2^2)^{j-1}} \\ - \sum_{j=k+1}^{l} \frac{1}{2^{2j} ||w||_2^{4j+2}} < 0 \end{aligned} \tag{3.124}$$

The first term dominates the second term for any finite value of l using a geometric series argument. For $l \to \infty$,

$$-\frac{4}{(2||w||_2^2)^k} + \frac{2}{(2||w||_2^2)^k}\frac{1}{1-\frac{1}{2||w||_2^2}} \tag{3.125}$$

$$= -\frac{4}{(2||w||_2^2)^k} + \frac{2}{(2||w||_2^2)^k}\frac{2||w||_2^2}{2||w||_2^2 - 1} \leq 0 \tag{3.126}$$

The fraction $\frac{2||w||_2^2}{2||w||_2^2-1}$ is at most 2 for $||w||_2^2 = 1$ and the expression is strictly negative for $||w||_2^2 > 1$. Due to the third negative term in Eq. (3.124), the claim is true.

The proof for the second condition (Eq. (3.121)) is similar. We first compute $d(a_k, X)^2 - d(a_{k+1}, X)^2$ and consider $k < m$.

$$\begin{aligned} d(a_k, X)^2 - d(a_{k+1}, X)^2 &= -4c_k(w_k^T X - b_k) \\ &+ 2c_{k+1}(w_{k+1}^T X - b_{k+1}) - c_{k+1}^2 ||w||_2^2 > 0 \end{aligned} \tag{3.127}$$

Since $w_k^T X - b_k < 0$, we take the value -1, making the contribution of the first term positive and small. Similarly, we take $w_{k+1}^T X - b_{k+1} = -2||w||_2^2$. Hence,

$$\frac{4}{(2||w||_2^2)^k} - \frac{2}{(2||w||_2^2)^k} - \frac{1}{(2^{2k+2}||w||_2^{4k+2})} \tag{3.128}$$

$$= \frac{2}{(2^k||w||_2^{2k})} - \frac{1}{(2^{2k+2}||w||_2^{4k+2})} > 0 \tag{3.129}$$

The last inequality holds since $||w||_2^2 \geq 1$ and $k \geq 1$.

Finally, we consider $k = m$ separately and since $w_m^T X < b_m$, we have

$$d(a_m, X)^2 - d(a_{m+1}, X)^2 = -4c_m(w_m^T X - b_m) > 0$$

The resolution claim follows by how small $c_m = \frac{1}{(2||w||_2^2)^m}$ is. Therefore, it becomes $O(mRES(\text{diag}(WW^T)))$. The resolution of the point X^* is bounded by the resolution of $||w||_2^2$ so asymptotically, it makes no difference in the resolution of the anchors. □

In addition, we can replace the regularity conditions in Theorem 32 only with the condition $m \leq n$ where m is the depth of the list. Let A^+ denote the Moore-Penrose inverse of a matrix $A \in \mathbb{R}^{m \times n}$.

Theorem 33. *Suppose that an n-input Linear Decision List $l(X)$ of depth m is given with a weight matrix $W \in \mathbb{Z}^{m \times n}$ where $m \leq n$ and a bias vector $b \in \mathbb{Z}^m$. Then, there is an NN representation for $l(X)$ with $m + 1$ anchors and resolution $O(RES(W^+) + mRES(\text{diag}(WW^T)))$.*

Proof. First of all, this proof depends on the proof of Theorem 32 with an explicit algorithm to find X^*, which is the most crucial step and will be done later in the proof.

We first assume that $w_i^T X \neq b_i$ for any $X \in \{0,1\}^n$ without loss of generality simply by changing the bias vector b to $b - 0.5$. Compared to Theorem 32, since we do not assume any regularity conditions, this change is sufficient. In addition, we can assume without loss of generality that W is full-rank by another perturbation argument. For example, for very small $\epsilon > 0$, we take $W' = W + \epsilon I_{m,n}$ where $I_{m,n}$ is the $m \times n$ sub-identity matrix with the first m rows of the $I_{n\times n}$ and the threshold functions $\mathbb{1}\{w_i^T X \geq b_i - 0.5\}$ are the same as $\mathbb{1}\{w_i^T X + \epsilon X_i \geq b_i - 0.5\}$ when $\epsilon < 1/2$. Let $W_{m\times m}$ denote the $m \times m$ sub-matrix of W. W' is full-rank if and only if

$$\begin{aligned} \det(W'(W')^T) = \det\Big(&WW^T + \epsilon(W_{m\times m} + (W_{m\times m})^T) \\ &+ \epsilon^2 I_{m\times m}\Big) \neq 0 \end{aligned} \tag{3.130}$$

Because Eq. (3.130) is a finite polynomial in ϵ with at most $2m$ many roots, there are infinitely many choices for $0 < \epsilon < 1/2$.

We have the same construction in the proof of Theorem 32 with an additional assumption: Whenever we subtract $c_i w_i$ from $X^* - \sum_{j<i} c_j w_j$, the vector should satisfy $w_{i+1}^T x = b_{i+1}$. Define $X^*_{(i)} = X^* - \sum_{j<i} c_j w_j$ and $X^*_{(1)} = X^*$.

$$a_i = X^* - \sum_{j<i} c_j w_j + c_i w_i \text{ for } i = 1, \ldots, m \tag{3.131}$$

$$a_{m+1} = X^* - \sum_{j<m} c_j w_j - c_m w_m \tag{3.132}$$

where $w_i^T X^*_{(i)} = b_i$.

Under this assumption, we see that the squared distance differences are equivalent to the ones in the proof of Theorem 32 (see Eq. (3.123)). For $i < m$, we have

$$\begin{aligned} d(a_i, X)^2 &= |X| - 2X^T X^*_{(i)} + ||X^*_{(i)}||_2^2 \\ &\quad - 2c_i(w_i^T X - w_i^T X^*_{(i)}) + c_i^2 ||w_i||_2^2 \\ &= |X| - 2X^T(X^* - \sum_{j<i} c_j w_j) + ||X^* - \sum_{j<i} c_j w_j||_2^2 \\ &\quad - 2c_i(w_i^T X - b_i) + c_i^2||w_i||_2^2 \\ &= |X| - 2X^T X^* + ||X^*||_2^2 \end{aligned}$$

$$
\begin{aligned}
&+2\sum_{j<i} c_j(w_j^T X - w_j^T X^*) + \Big(\sum_{j<i} c_j w_j\Big)^2 \\
&-2c_i(w_i^T X - b_i) + c_i^2 ||w_i||_2^2 \\
=\;& |X| - 2X^T X^* + ||X^*||_2^2 \\
&+2\sum_{j<i} \Big(c_j(w_j^T X - w_j^T X^* + \sum_{k<j} c_k w_j^T w_k)\Big) \\
&+\sum_{j<i} c_j^2 ||w_j||_2^2 - 2c_i(w_i^T X - b_i) + c_i^2 ||w_i||_2^2 \\
=\;& |X| - 2X^T X^* + ||X^*||_2^2 \\
&+2\sum_{j<i} c_j(w_j^T X - w_j^T(X^* - \sum_{k<j} c_k w_k)) \\
&+\sum_{j<i} c_j^2 ||w_j||_2^2 - 2c_i(w_i^T X - b_i) + c_i^2 ||w_i||_2^2 \\
=\;& |X| - 2X^T X^* + ||X^*||_2^2 \\
&+2\sum_{j<i} c_j(w_j^T X - b_j) + \sum_{j<i} c_j^2 ||w_j||_2^2 \\
&-2c_i(w_i^T X - b_i) + c_i^2 ||w_i||_2^2
\end{aligned} \tag{3.133}
$$

One can observe that Eq.(3.133) is the same as Eq. (3.122) except the squared norms of w_i. Therefore, for a correct selection of c_is, the construction should follow by satisfying the conditions Eq. (3.120) and (3.121). We pick $c_i = c_{i-1}2||w||_2^2$ and $c_1 = 1$ for simplicity where $||w||_2^2 = \max_k ||w_k||_2^2$. By similar bounding arguments, the same steps in the proof of Theorem 32 could be followed.

Now, we have to find an explicit X^* to conclude the proof. Recall that $w_i^T X_{(i)}^* = b_i$ where $X_{(i)}^* = X^* - \sum_{j<i} c_j w_j$. Then, $w_i^T X^* = b_i + \sum_{j<i} c_j w_i^T w_j$. Clearly, this defines a linear system $WX^* = B$ where $B \in \mathbb{Q}^m$ and $B_i = b_i + \sum_{j<i} c_j w_i^T w_j$. Since W is full-rank without loss of generality, there always exists $X^* = W^+ B$ which solves the system exactly.

We observe that $b_i \leq ||w_i||^2$ and $|w_i^T w_j| \leq \max_k ||w_k||_2^2$. Since c_is shrink geometrically, the resolution is $O(m\,\mathrm{diag}(WW^T))$ so that $RES(B) = mRES(\mathrm{diag}(WW^T))$. Hence, the resolution of X^* becomes $O(RES(W^+) + mRES(\mathrm{diag}(WW^T)$. We conclude that the resolution of the construction is $O(RES(W^+) + mRES(\mathrm{diag}(WW^T)))$.

□

Corollary 33.1. *For any symmetric Boolean function $f(X)$ with $I(f)$ many intervals, $NN(f) \leq I(f)$.*

Even though the proof techniques are not exactly the same, Corollary 33.1 is in fact equivalent to what we had in Section 3.2 with a couple of tweaks. Corollary follows by a simple construction of symmetric Boolean functions as LDLs. For instance, we describe the LDL construction for an 8-input symmetric Boolean function $f(X)$ with $I(f) = 5$ where each interval has the form $[I_{i-1} + 1, I_i]$ for $i \in \{1, \dots, 5\}$ (take $I_0 = -1$). It is easy to verify that the LDL with depth 4 given in Figure 3.8 computes $f(X)$. This function is the counterexample we mention in Section 3.2 where there is no PARITY-based extension for 5-anchor NN representations.

$\lvert X\rvert$	$f(X)$	
0	1	$I_1 = 0$
1	**0**	$I_2 = 1$
2	1	
3	1	
4	1	
5	1	
6	1	$I_3 = 6$
7	**0**	$I_4 = 7$
8	1	$I_5 = 8$

(3.134)

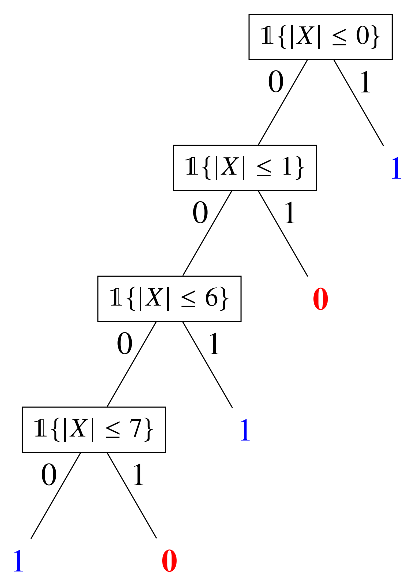


Figure 3.8: A Linear Decision List of depth 4 for the symmetric Boolean function in Eq.(3.134) with $I(f) = 5$.

Conjecture 4. *Let $f(X)$ be a Boolean function. Then, $NN(f) \leq LDL(f) + 1$ where $LDL(f)$ is the smallest depth of linear decision lists computing $f(X)$.*

By Theorem 33, the conjecture is true when $LDL(f) \leq n$.

We now focus on the NN representations of EDLs. For EDLs, we give a construction idea similar to circuits of OR $\circ$ ELT with a slight change in parameters to implement the domination principle.

Theorem 34. *Suppose that an n-input Exact Decision List $l(X)$ of depth m is given under regularity conditions. Then, there is an NN representation for $l(X)$ with $(m+1)2^m$ anchors and resolution $O(\log m + RES(\text{diag}(WW^T)))$.*

Proof. We essentially construct a NN representation for the OR $\circ$ ELT type of circuits (consider Corollary 29.1) and modify it to obtain the *domination principle*. We consider the anchors as follows similar to the proof of Theorem 29 with two types. We assume $d > c_i$ and all $c_i > 0$ except $d = c_{m+1} > 0$.

$$a_{jk} = X^* + du_{jk} + (-1)^{j_m} c_k w_k \text{ for } k \in \{1, \dots, m\} \tag{3.135}$$

$$a_{j(m+1)} = X^* + du_{jm} + (-1)^{j_m} c_{m+1} w_m \tag{3.136}$$

where $u_{jk} = \pm w_1 \pm \dots \pm w_{k-1} \pm w_{k+1} \pm \dots \pm w_m$ (only w_k is excluded) for $k \in \{1, \dots, m\}$ and $j_m \in \{0, 1\}$. Also, $c_{m+1} = d$. The sign pattern is given by the binary expansion of $j-1$ in $m-1$ bits as in the proof of Theorem 29. For example, for $m = 5$, $j - 1 = 4$ gives $(j_1, j_2, j_3, j_4) = (0, 0, 1, 0)$ and $u_{52} = w_1 + w_3 - w_4 + w_5$. If, in addition, $j_5 = 0$, then we find $a_{52} = d(w_1 + w_3 - w_4 + w_5) + c_2 w_2$. In comparison, if $j_5 = 1$, we obtain $a_{(20)2} = d(w_1 + w_3 - w_4 + w_5) - c_2 w_2$.

We have the following squared Euclidean norm expression for this construction.

$$\begin{aligned} d(a_{jk}, X)^2 = |X| - 2X^T X^* + ||X^*||^2 \\ - 2du_{jk}^T(X - X^*) + d^2||u_{jk}||_2^2 \\ - 2j_m c_k(w_k^T X - b_k) + c_k^2||w_k||_2^2 \\ - 2j_m dc_k w_k^T u_{jk} \end{aligned} \tag{3.137}$$

By the orthogonality assumption and the constructions of u_{jk}s, we have $w_k^T u_{jk} = 0$. Since our goal is to find the anchor minimizing the Euclidean distance, there is a $j = j^*$ such that

$$\begin{aligned} d(a_{j^*k}, X)^2 = |X| - 2X^T X^* + ||X^*||^2 \\ - 2d \sum_{i \neq k} |w_i^T X - b_i| + d^2(m-1)||w||_2^2 \\ - 2c_k|w_k^T X - b_k| + c_k^2||w||_2^2 \end{aligned} \tag{3.138}$$

Given the optimal selection of js minimizing the Euclidean distance, we have to find the argument k which will globally minimize this.

For an EDL, we see that z_l is picked if and only if $(w_1^T X \neq b_1, \ldots, w_{l-1}^T X \neq b_{l-1}, w_l^T X = b_l, \times, \ldots, \times)$ where we have inequalities for $k < l$ and a *don't care* region for $l < k$. We will deal with $(w_1^T X \neq b_1, \ldots, w_m^T X \neq b_m)$ later.

We want a_{j^*k} to be the closest anchor to X for $k = l$ and some j^* for $(w_1^T X \neq b_1, \ldots, w_{l-1}^T X \neq b_{l-1}, w_l^T X = b_l, \times, \ldots, \times)$. Hence, when we compare different k, l we get

$$\begin{aligned} &d(a_{j^*l}, X)^2 - d(a_{j^+k}, X)^2 \\ &\quad = -2(d - c_k)|w_k^T X - b_k| + (c_l^2 - c_k^2)||w||_2^2 < 0 \end{aligned} \tag{3.139}$$

Note that $w_l^T X = b_l$ so that term does not appear.

<u>**Case 1** $(l < k)$**:**</u> This is the simple case. The inequality in Eq. (3.139) is the tightest when $|w_k^T X - b_k| = 0$. Then, for $k \leq m$, we obtain $c_l < c_k$ for $l < k$ as a necessary condition. $k = m + 1$ is trivial since $d > c_i$ for all $i \in \{1, \ldots, m\}$ and $d(a_{j^*l}, X)^2 - d(a_{j^+(m+1)}, X)^2 = (c_l^2 - d^2)||w||_2^2 < 0$.

<u>**Case 2** $(k < l \leq m)$**:**</u> The tightest Eq. (3.139) becomes is when $|w_k^T X - b_k| = 1$ and we have

$$c_k^2 - \frac{2}{||w||_2^2} c_k + 2\frac{d}{||w||_2^2} - c_l^2 > 0 \tag{3.140}$$

Let $d = 1/||w||_2^2$ and $c_i = \frac{i}{(m+1)||w||_2^2}$ for $i \in \{1, \ldots, m\}$. Then, we obtain

$$\frac{k^2}{(m+1)^2} - 2\frac{k}{m+1} + 2 - \frac{l^2}{(m+1)^2} > 0 \tag{3.141}$$

Since $l \leq m$, the tightest this inequality becomes is when the value of the fourth term is 1. Then, we obtain

$$\left(\frac{k}{(m+1)^2} - 1\right)^2 > 0 \tag{3.142}$$

which is true for since $k \neq m + 1$.

<u>**Case 3** $(l = m + 1)$**:**</u> Finally, we consider $(w_1^T X \neq b_1, \ldots, w_m^T X \neq b_m)$. For this case, we claim that for any $k \in \{1, \ldots, m\}$,

$$\begin{aligned} &d(a_{j^*(m+1)}, X)^2 - d(a_{j^+k}, X)^2 \\ &\quad = -2(d - c_k)|w_k^T X - b_k| + (d^2 - c_k^2)||w||_2^2 < 0 \end{aligned} \tag{3.143}$$

Take $|w_k^T X - b_k| = 1$ similarly. Then,

$$c_k^2 - \frac{2}{||w||_2^2} c_k + 2\frac{d}{||w||_2^2} - d^2 > 0 \tag{3.144}$$

Consider $d = 1/||w||_2^2$ and $c_i = \frac{i}{(m+1)||w||_2^2}$ for $i \in \{1, \dots, m\}$. We get

$$\frac{k^2}{(m+1)^2} - 2\frac{k}{m+1} + 2 - 1 > 0 \tag{3.145}$$

$$\Big(\frac{k}{m+1} - 1\Big)^2 > 0 \tag{3.146}$$

which is true since $k \neq m+1$.

This shows that the construction works. The size of representation is $(m+1)2^m$ by counting through js and ks. Similar to the proof of Theorem 29, the resolution is $O(\log m + RES(\text{diag}(WW^T)))$. $\square$

We note that the idea for Theorem 34 works for LDLs as well with $(m+1)2^m$ many anchors and a possible resolution improvement from $O(m)$ to $O(\log m)$.

Corollary 34.1. *Let $f(X)$ be the $2nm$-input Boolean function* $\text{OMB}_m \circ \text{EQ}_{2n}$ *where there are m many disjoint $2n$-input EQ functions in the first layer. Then, there is an NN representation with $(m+1)2^m$ anchors and $O(n)$ resolution.*

3.7 Towards the Low Resolution NN Representations of Threshold Functions

The main problem we are interested in is whether there exists NN representations for threshold functions with polynomially large size in n and logarithmic resolution. Since threshold functions can be computed by small weight (i.e. logarithmic resolution) depth-2 threshold circuits, it is a natural direction to find NN representations for these depth-2 threshold circuits based on the techniques we developed in the previous sections.

In Chapter 2, it is established that $\text{ELT} \subseteq \text{SYM} \circ \widehat{\text{ELT}}$ and it seems natural to apply our results in Section 3.5. However, the regularity conditions are not satisfied for this transformation, and even if they are, there is a superpolynomial blow-up in the size of the NN representation for these constructions. Therefore, it seems that following this idea might not be very useful. We again emphasize that the known results for $\text{LT} \subseteq \widehat{\text{LT}} \circ \widehat{\text{LT}}$ rely on the transformations $\text{ELT} \subseteq \text{SYM} \circ \widehat{\text{ELT}}$ and we thus have similar obstacles.

Nevertheless, for threshold functions like the EQ, COMP, and OMB, whose domination form have the sparsity parameter $\mathcal{S} = 1$, we can explicitly construct NN representations of polynomial size and at most logarithmic resolution. Whether similar NN representations exist for arbitary threshold functions is not clear.

Low Resolution NN Representations for the EQ

In Chapter 2, we showed that there are constant weight EQ matrices that can be used to compute the EQ function to optimize the number of the threshold gates using AND $\circ$ ELT circuits. Using the results in Section 3.4, specifically Theorem 27, it is clear that we can obtain a low resolution NN representation for the EQ function. Here, we give a more specialized proof.

Theorem 35. *Consider an EQ matrix $A \in \mathbb{Z}^{m\times n}$ with no all-zero rows. Then, there is a $2m+1$-anchor NN representation for the $2n$-input* EQ *function with* $O(RES(\text{diag}(AA^T)))$.

Proof. The anchors we construct are in the following form for $i \in \{1,\dots,m\}$. We define $W = \begin{bmatrix} A & -A \end{bmatrix}_{m\times 2n}$.

$$a_0 = 0.5\mathbb{1} \tag{3.147}$$
$$a_{2i-1} = 0.5\mathbb{1} + c_i W_i \tag{3.148}$$
$$a_{2i} = 0.5\mathbb{1} - c_i W_i \tag{3.149}$$

We have two cases similar to the proof of Theorem 19 to prove where a_0 is closest to vectors $X = Y$ and a_i is closest to $X \neq Y$ for some $i \in \{1,\dots,2n\}$. We also use $(X,Y) \in \{0,1\}^{2n}$ to denote the input vector. When we expand the squared Euclidean distance, we get

$$d(a_0,(X,Y))^2 = n/2 \tag{3.150}$$
$$d(a_{2i-1},(X,Y))^2 = n/2 - 2c_iA_i^T(X-Y) + 2c_i^2||A_i||_2^2 \tag{3.151}$$
$$d(a_{2i},(X,Y))^2 = n/2 + 2c_iA_i^T(X-Y) + 2c_i^2||A_i||_2^2 \tag{3.152}$$

Since $X - Y \in \{-1,0,1\}^n$, $A(X-Y) = 0$ if and only if $X = Y$. Therefore, if $X = Y$, $d(a_0,(X,Y))^2 < d(a_i,(X,Y))^2$ for all $i \in \{1,\dots,2n\}$ as long as $c_i > 0$.

Conversely, if $X \neq Y$, then $|A_i^T(X-Y)| \geq 1$ for some $i \in \{1,\dots,n\}$. Depending on the sign of $A_i^T(X-Y)$, one of the a_{2i-1} or a_{2i} will be closer to (X,Y). Hence, if

$c_i = \frac{1}{2||A_i||_2^2}$, then

$$-2c_i|A_i^T(X-Y)| + 2c_i^2||A_i||_2^2 = -\frac{|A_i^T(X-Y)| - 0.5}{||A_i||_2^2} < 0 \tag{3.153}$$

so that either a_{2i-1} or a_i will be closest to (X, Y). The resolution bound is easy to verify by the selection of c_is. □

The properties of EQ matrices directly translate into the NN representations of the EQ. We obtain the following corollary concluding our results in this context. We give examples NN representations for each type in Corollary 35.1 in Appendix E.1.

Corollary 35.1. *For the* $2n$*-input EQ function, there are NN representations such that*

Size	Resolution
3	$O(n)$
$2n+1$	$O(1)$
$O(n/\log n)$	$O(\log n)$

Proof. For the first result, we use the EQ matrix with exponentially large weights taking $m = 1$. This gives the linear resolution construction. For the second result, we use $A = I$, which is a full-rank matrix for the EQ function where $m = n$. Since each row has norm 1, this gives $O(1)$ resolution. Finally, for the third result, we use our results on EQ matrices (see Theorem 10 for example). In this case, the number of rows is asymptotically $O(n/\log n)$ and each row norm is bounded by n, giving a logarithmic resolution. □

Low Resolution NN Representations for the COMP and OMB

One can notice that for the COMP and OMB, the most significant bits in the input determine the output. This is called the *domination principle* and the property used implicitly in many works (Amano and Maruoka, 2005; Bohossian, Riedel, and Bruck, 1998). In order to decide the label of the anchor using the domination principle, a procedure based on a sequential decision making could be useful similar to decision lists. Therefore, loosely speaking, some anchors should be "closer" more often than the others.

One can see the following necessary and sufficient conditions for the COMP with the convention $X = (X_1, \dots, X_n)$ and $Y = (Y_1, \dots, Y_n)$ and binary expansion

$\sum_{i=1}^{n} 2^{n-i} X_i$. Below, $\times$s denote *don't cares*.

$$X > Y \Leftrightarrow X - Y = (0, \dots, 0,\ \ 1, \times, \dots, \times) \tag{3.154}$$

$$X < Y \Leftrightarrow X - Y = (0, \dots, 0, -1, \times, \dots, \times) \tag{3.155}$$

$$X = Y \Leftrightarrow X - Y = (0, \dots, 0, 0, 0, \dots, 0) \tag{3.156}$$

This is an altered version of the result in Lemma 6. The vector $X - Y$ has leading 0s and the most significant digit determines if $X > Y$ or not. The idea to construct an NN representation for the COMP relies on this observation and we depict the idea in Figure 3.9. First, we consider the hyperplane $x_1 = y_1$ and put two anchors whose midpoint is $X^* = 0.5\mathbb{1}$, which is on the hyperplane itself so that a_1 is closer to $X_1 > Y_1$ and a_2 is closer to $X_1 < Y_1$. Then, alongside the intersection of $x_1 = y_1$ and $x_2 = y_2$, we put two more anchors a_3 and a_4 to take care of $X_2 > Y_2$ and $X_2 < Y_2$ and we continue like this. To ensure that we have the domination principle, a_3 and a_4 are farther away from X^*. Since $\text{COMP}(X, Y) = 1$ when $X = Y$, we make the anchors skewed by a little amount so that $X = Y$ vectors are closer to blue anchors.

Theorem 36. *For the* $2n$*-input* COMP*, there is an NN representation with* $2n$ *anchors and* $O(\log n)$ *resolution.*

Proof. We will show that the following is an NN representation for the COMP_{2n} function for $i \in \{1, \dots, n\}$ and $W = \begin{bmatrix} I & -I \end{bmatrix}_{n \times 2n}$

$$X^* = 0.5\mathbb{1} \tag{3.157}$$

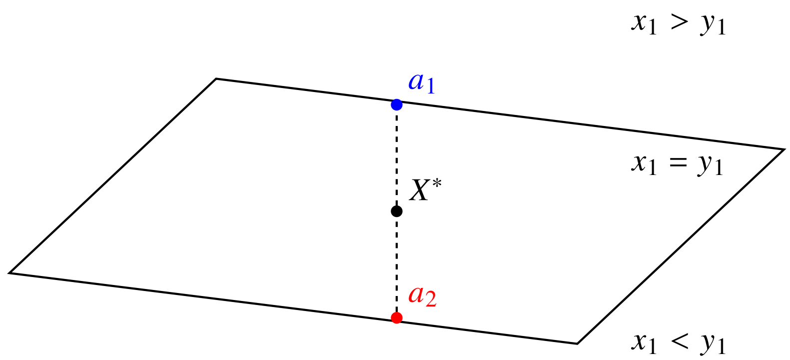
$$a_{2i-1} = X^* + c_{2i-1} W_i \tag{3.158}$$

$$a_{2i} = X^* - c_{2i} W_i \tag{3.159}$$

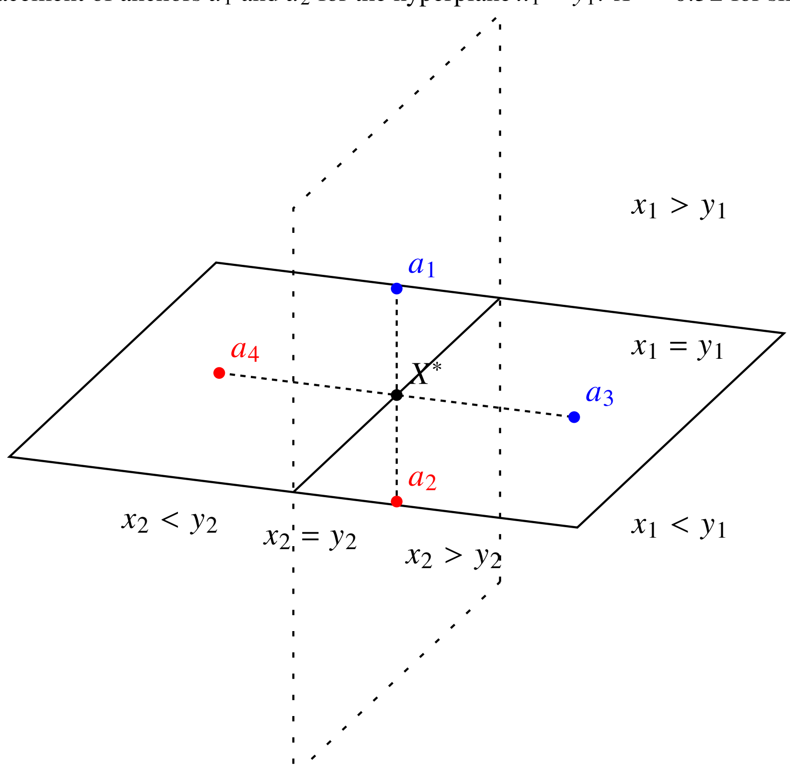
for $c_i = \frac{1}{2} + \frac{i-1}{4n}$ for $i \in \{1, \dots, 2n\}$. This selection gives the desired resolution bound.

Let (X, Y) denote the $2n$-dimensional input vector and let $(\mathcal{X}, \mathcal{Y})^{(k)}$ be the set of vectors where $X_i = Y_i$ for $i < k$, $X_k = 1$ and $Y_k = 0$. For $i > k$, the values can be arbitrary. Clearly, $X > Y$ for any vector in $(X, Y)^{(k)} \in (\mathcal{X}, \mathcal{Y})^{(k)}$ (see Eq. (3.154)). We claim that the closest anchor to any $(X, Y)^{(k)}$ is a_{2k-1}.

$$\begin{aligned} &d(a_{2i-1}, (X, Y)^{(k)})^2 \\ &\quad = |(X, Y)^{(k)}| - 2(0.5\mathbb{1} + c_{2i-1} W_i)^T (X, Y)^{(k)} \\ &\qquad + ||0.5\mathbb{1} + c_{2i-1} W_i||_2^2 \end{aligned}$$



(a) The placement of anchors a_1 and a_2 for the hyperplane $x_1 = y_1$. $X^* = 0.5\mathbb{1}$ for simplicity.



(b) The placement of a_3 and a_4 for the hyperplanes $x_1 = y_1$ and $x_2 = y_2$. They are farther away from the X^* compared to a_1 and a_2 to ensure the domination principle.

Figure 3.9: The construction idea for the COMP(X, Y) function depicting the first two iterations. For $2n$-inputs, there will be n iterations resulting in $2n$ many anchors.

$$
\begin{aligned}
&= -2c_{2i-1}(X_i - Y_i) + \frac{n}{2} + 2c_{2i-1}^2 ||W_i||_2^2 \\
&= \frac{n}{2} + \frac{1}{2} + \frac{i-1}{n} + 2\Big(\frac{i-1}{2n}\Big)^2 - \Big(1 + \frac{i-1}{n}\Big)(X_i - Y_i) \qquad (3.160)
\end{aligned}
$$

Since $-\frac{1}{2} + 2\Big(\frac{k-1}{2n}\Big)^2 < 0 < \frac{1}{2} + \frac{i-1}{n} + 2\Big(\frac{i-1}{2n}\Big)^2$ for any $i < k$, a_{2k-1} is closer to $(X,Y)^{(k)}$ than a_{2i-1} for $i < k$. For $i > k$, the smallest distance value is attained when $X_i - Y_i = 1$. For this, we see that $-\frac{1}{2}+2\Big(\frac{k-1}{2n}\Big)^2 < -\frac{1}{2}+2\Big(\frac{i-1}{2n}\Big)^2$ for $i > k$ since this expression is monotonically increasing.

We now look at the distance between $(X,Y)^{(k)}$ and a_{2i}s.

$$\begin{aligned} d(a_{2i},(X,Y)^{(k)})^2 = \frac{n}{2} + \frac{1}{2} + \frac{2i-1}{2n} + 2\Big(\frac{2i-1}{4n}\Big)^2 \\ + \Big(1 + \frac{2i-1}{2n}\Big)(x_i - y_i) \end{aligned} \tag{3.161}$$

When we compare $d(a_{2i-1},(X,Y)^{(k)})^2$ and $d(a_{2i},(X,Y)^{(k)})^2$ for $i < k$, we have $-\frac{1}{2} + 2\Big(\frac{k-1}{2n}\Big)^2 < 0 < \frac{1}{2} + \frac{2i-1}{2n} + 2\Big(\frac{2i-1}{4n}\Big)^2$. Also, for $i = k$, $d(a_{2k-1},(X,Y)^{(k)}) < d(a_{2k},(X,Y)^{(k)})$ because $x_k - y_k > 0$. For $i > k$, the minimum distance value is attained when $x_i - y_i = -1$, so we obtain $-\frac{1}{2} + \Big(\frac{2k-2}{4n}\Big)^2 < -\frac{1}{2} + \Big(\frac{2i-1}{4n}\Big)^2$. Hence, we conclude that a_{2i-1} is closer to $(X,Y)^{(k)}$ than a_{2i} for any i.

The proof is similar for $X < Y$. For $X = Y$, we see that a_1 is always the closest anchor with the correct label because c_1 is the smallest. Hence, the construction works indeed. □

An example NN representation with $O(\log n)$ resolution for 6-input COMP is given in Appendix E.2.

We finish this section with the construction for the OMB. Note that in this context, we use $X = (X_1, \dots, X_n)$ with X_1 being the most significant bit.

Theorem 37. *For the n-input* OMB*, there is an NN representation with $n+1$ anchors with $O(\log n)$ resolution.*

Proof. The construction we give here is very simple, which can be thought as a special case of a wider family of constructions that we do not cover here. We have the anchor matrix $A \in \mathbb{Q}^{(n+1)\times n}$ with resolution $O(\log n)$ for $i \in \{1,\dots,n+1\}$ and $j \in \{1,\dots,n\}$.

$$a_{ij} = \begin{cases} 1 - \frac{i-1}{n} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{3.162}$$

The label of a_i is blue for odd i and **red** for even i. Moreover, the last anchor a_{n+1} corresponds to the all-zero vector and it is labeled **red** because $\mathrm{OMB}(0) = 0$ by definition.

By $\mathcal{X}^{(k)}$, we denote the family of vectors where $X_j = 0$ necessarily for $j < k$ and $x_k = 1$. That is, for any $X \in \mathcal{X}^{(k)}$ $X = (0, \dots, 0, 1, \times, \dots, \times)$. For $j > k$, we have *don't cares*.

We claim that all $X \in \mathcal{X}^{(k)}$ are closest to a_i. By expanding the squared Euclidean distance, we have

$$d(a_i, X)^2 = |X| - 2a_i^T X + ||a_i||^2 \tag{3.163}$$

$$= \begin{cases} |X| + \left(1 - \frac{i-1}{n}\right)^2 & \text{if } i < k \\ |X| - 2\left(1 - \frac{i-1}{n}\right)X_i + \left(1 - \frac{i-1}{n}\right)^2 & \text{if } k \leq i \leq n \\ |X| & \text{if } i = n+1 \end{cases} \tag{3.164}$$

Since $X_k = 1$ and $-2\left(1 - \frac{i-1}{n}\right) + \left(1 - \frac{i-1}{n}\right)^2 < 0 < \left(1 - \frac{i-1}{n}\right)^2$, a_k is closer to X than a_is for $i < k$. Since $-2\left(1-\frac{i-1}{n}\right)+\left(1-\frac{i-1}{n}\right)^2 = -1+\left(\frac{i-1}{n}\right)^2$ is monotone increasing in i and always negative, the closest anchor is a_k indeed. This concludes the proof. □

We remark that the inclusion of the all-zero anchor for the NN representation of OMB is not necessary when n is even because the label of a_n and a_{n+1}. In this case, the representation size is exactly n. Two examples of NN representations for the OMB using this result are given in Appendix E.2.

3.8 Conclusion

Boolean functions in the new associative computation model are studied. Several important classes of Boolean functions are examined: LT, ELT, SYM, SYM ∘ LT, SYM ∘ ELT, DOM ∘ LT, and DOM ∘ ELT.

Explicit resolution bounds for the NN representations for LT and ELT are constructed using 2 and 3 anchors respectively. In general, both functions require weights as large as $2^{O(n \log n)}$ and therefore, the resolution of the NN representations are $O(n \log n)$. Also, the NN complexity and resolution bounds are optimized for their symmetric versions, namely, symmetric linear and symmetric exact Boolean functions.

It is shown that the complexity of symmetric Boolean functions rely on the number of intervals they have. For some cases, the constructions are optimal in size or resolution, which is logarithmic. For symmetric Boolean functions, it is an open question if adding polynomially large number of anchors can reduce the resolution to sublogarithmic quantities.

NN representations for SYM $\circ$ ELT and SYM $\circ$ LT are given under regularity conditions where the number of anchors depends on the number of intervals and the interval locations of the top symmetric gate. These constructions give us new NN representations for many significant functions in circuit complexity theory and Boolean analysis.

An important relationship between NN representations and LDLs is shown. If the length of an LDL is less than the number of inputs n, then there is NN representation with $n + 1$ anchors (the regularity condition is not necessary). Special cases are symmetric Boolean functions for which LDL complexity is smaller than n. For EDLs, an NN representation with exponentially large number of anchors in n is obtained under regularity conditions.

Finally, polynomial size NN representations with logarithmic (or constant) resolution are constructed for the EQ, COMP, and OMB. These functions have domination forms with sparsity parameter $\mathcal{S} = 1$. It remains a challenge to address the Conjecture 1 fully and seems that new methods or tools to understand DOM functions with $\mathcal{S} \geq 2$ are required.

Chapter 4

CONCLUSION AND OPEN PROBLEMS

This thesis delves into the enigma of the the computational processes of the human brain, by exploring two key models: Feedforward Neural Networks and Nearest Neighbor Representations. Emphasizing the importance of connectivity over "large" numerical values, the study navigates the intricate balance between bit resolution (or weight size) and computational depth. The associative computation model of NN representations could enlighten how the concepts are stored and processed in neural computation. The brain does not have unlimited resources and understanding the theoretical capabilities in this "low resolution" regime seems significant. Using novel techniques and different ways of representing Boolean functions, previously known results are re-established and extended to give deeper insights on the neural network representations.

Using the domination form of threshold functions and binary expansion of the weights has many merits, especially when $\mathcal{S} = 1$. For this family of Boolean functions, by the introduction of EQ matrices and RMDS property, powerful techniques such as Siegel's Lemma and Berry-Esseen Theorem are applied for the first time in circuit complexity theory to optimize the lower bounds up to constant factors and to reduce the weight sizes without an asymptotic increase in the circuit size. In addition, the recursive construction of EQ matrices might inspire other constructions for different applications since the construction is connected to the Hadamard matrices algebraically. The only method to generate RMDS matrices relies on the CRT and an alternative constructive method could be interesting. In addition, our understanding of threshold functions whose domination form has sparsity $\mathcal{S} \geq 2$ is limited. The role of $\mathcal{S}$ needs a more careful treatment in the analysis of threshold functions. Some of these comments also apply to NN representations of threshold functions with logarithmic resolution and a similar gap between $\mathcal{S} = 1$ and $\mathcal{S} \geq 2$ cases exists.

Starting from the NN representations of threshold functions, we obtain several important constructions and lower bounds for some famous Boolean functions. Since the input vectors are $\{0, 1\}$ vectors, it seems reasonable to put anchors near the hypercube. However, for symmetric Boolean functions, we have shown that the optimization of the representation size might move the representation far away from the hypercube. This counter-intuitive result illustrates the intrinsic difficulty of proving upper and lower bounds for NN complexity and resolution.

There are still many remaining mysteries for NN representations. The obvious open problem is to prove the conjecture to find low resolution NN representations for arbitrary threshold functions with at most a polynomially large size increment in n. Although affirmative results are provided for the EQ, COMP, and OMB (threshold functions with $\mathcal{S} = 1$), it is not clear if this is true in general. An important observation is to consider k-NN representations since for a "complex" function like IP2, k-NN complexity is shown to be $O(n)$ in a recent work. It could be possible that the conjecture is false for 1-NN but true for k-NN with $k = \Omega(n)$.

While it seems likely that $NN(f) = I(f)$ for any symmetric Boolean function, we can only prove it for $I(f) \leq 4$. More sophisticated techniques are required to make progress for higher number of intervals. The role of resolution is also ambiguous for the NN representations of symmetric Boolean functions. An interesting result would be to prove that there is no polynomial size NN representation for the PARITY function with constant resolution.

The removal of regularity conditions for the NN representation constructions for SYM $\circ$ LT, SYM $\circ$ ELT and EDLs is also an important direction. We believe that the removal of regularity conditions for the NN representations of LDLs could be beneficial for future progress.

For LDLs, if the depth constraint $m \leq n$ can be removed, this could imply a new way to construct NN representations for arbitrary Boolean functions. The idea is to use fact that any n-input Boolean function can be interpreted as a 2^n-input symmetric Boolean function. A definition for the *intervals* of a Boolean function would be relevant in this context.

Finally, relationships between different complexity measures of Boolean functions can be conjectured. It is known that NN complexity is bounded below by $LDT(f)$ and we conjecture that $NN(f) \leq LDL(f) + 1$. We verify this for $m \leq n$ where m is the depth of the list. It is also an important problem to find the relationships between exact covers and NN representations. This is treated in more detail in the Appendix C. The relationship between circuit size lower bounds for small weight constructions and NN complexity is intriguing because it is not clear if the size of the low resolution NN constructions for EQ, COMP, and OMB is optimal.

BIBLIOGRAPHY

Aaronson, James et al. (2021). "Exact hyperplane covers for subsets of the hypercube". In: *Discrete Mathematics* 344.9, p. 112490.

Alon, Noga, Peter Frankl, and Vojtěch Rödl (1985). "Geometrical realization of set systems and probabilistic communication complexity". In: *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*. IEEE, pp. 277–280.

Alon, Noga and Zoltán Füredi (1993). "Covering the Cube by Affine Hyperplanes". In: *European Journal of Combinatorics* 14.2, pp. 79–83. ISSN: 0195-6698. DOI: `https://doi.org/10.1006/eujc.1993.1011`. URL: `https://www.sciencedirect.com/science/article/pii/S0195669883710115`.

Alon, Noga and Văn H Vũ (1997). "Anti-Hadamard matrices, coin weighing, threshold gates, and indecomposable hypergraphs". In: *Journal of Combinatorial Theory, Series A* 79.1, pp. 133–160.

Amano, Kazuyuki (2020). "On the size of depth-two threshold circuits for the inner product mod 2 function". In: *International Conference on Language and Automata Theory and Applications*. Springer, pp. 235–247.

Amano, Kazuyuki and Akira Maruoka (2005). "On the complexity of depth-2 circuits with threshold gates". In: *International Symposium on Mathematical Foundations of Computer Science*. Springer, pp. 107–118.

Angelino, Elaine et al. (2018). "Learning certifiably optimal rule lists for categorical data". In: *Journal of Machine Learning Research* 18.234, pp. 1–78.

Babai, László et al. (2010). "Weights of exact threshold functions". In: *International Symposium on Mathematical Foundations of Computer Science*. Springer, pp. 66–77.

Beck, József (2017). "Siegel's Lemma is sharp". In: *A journey through discrete mathematics*. Springer, pp. 165–206.

Beigel, Richard (1994). "Perceptrons, PP, and the polynomial hierarchy". In: *Computational complexity* 4, pp. 339–349.

Berry, Andrew C (1941). "The accuracy of the Gaussian approximation to the sum of independent variates". In: *Transactions of the american mathematical society* 49.1, pp. 122–136.

Beyer, Kevin et al. (1999). "When is "nearest neighbor" meaningful?" In: *Database Theory—ICDT'99: 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7*. Springer, pp. 217–235.

Björner, Anders, László Lovász, and Andrew CC Yao (1992). "Linear decision trees: volume estimates and topological bounds". In: *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pp. 170–177.

Bohman, Tom (1998). “A construction for sets of integers with distinct subset sums”. In: *the electronic journal of combinatorics*, R3–R3.

Bohossian, Vasken, Marc D Riedel, and Jehoshua Bruck (1998). “Trading Weight Size for Circuit Depth: A Circuit for Comparison”. In: URL: `https://authors.library.caltech.edu/26049/1/etr028.pdf`.

Bombieri, Enrico and Walter Gubler (2006). *Heights in Diophantine geometry*. 4. Cambridge university press.

Boole, George (1847). *The mathematical analysis of logic*. Philosophical Library.

Bruck, Jehoshua (1990). “Harmonic analysis of polynomial threshold functions”. In: *SIAM Journal on Discrete Mathematics* 3.2, pp. 168–177.

Buhrman, Harry, Nikolay Vereshchagin, and Ronald de Wolf (2007). “On computation and communication with small bias”. In: *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC’07)*. IEEE, pp. 24–32.

Chattopadhyay, Arkadev, Meena Mahajan, et al. (2019). “Lower bounds for linear decision lists”. In: *arXiv preprint arXiv:1901.05911*.

Chattopadhyay, Arkadev and Nikhil Mande (2018). “A short list of equalities induces large sign rank”. In: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, pp. 47–58.

Clifton, Alexander and Hao Huang (2020). “On almost k-covers of hypercubes”. In: *Combinatorica* 40.4, pp. 511–526.

Costa, Simone, Stefano Della Fiore, and Marco Dalai (2021). “Variations on the Erdős Distinct-sums Problem”. In: *arXiv preprint arXiv:2107.07885*.

Cover, Thomas and Peter Hart (1967). “Nearest neighbor pattern classification”. In: *IEEE transactions on information theory* 13.1, pp. 21–27.

Cybenko, George (1989). “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4, pp. 303–314.

Dahiya, Yogesh et al. (2024). “Linear threshold functions in decision lists, decision trees, and depth-2 circuits”. In: *Information Processing Letters* 183, p. 106418. ISSN: 0020-0190. DOI: `https://doi.org/10.1016/j.ipl.2023.106418`.

Diamond, Benjamin E and Amir Yehudayoff (2022). “Explicit exponential lower bounds for exact hyperplane covers”. In: *Discrete Mathematics* 345.11, p. 113114.

DiCicco, Mason, Vladimir Podolskii, and Daniel Reichman (2024). *Nearest Neighbor Complexity and Boolean Circuits*. arXiv: `2402.06740 [cs.CC]`.

Dubroff, Quentin, Jacob Fox, and Max Wenqiang Xu (2021). “A Note on the Erdős Distinct Subset Sums Problem”. In: *SIAM Journal on Discrete Mathematics* 35.1, pp. 322–324.

Erdős, P (1983). “Extremal problems in number theory, combinatorics and geometry”. In: *Proc. Inter. Congress in Warsaw*.

Esseen, Carl-Gustav (1942). "On the Liapunov limit error in the theory of probability". In: *Ark. Mat. Astr. Fys.* 28, pp. 1–19.

Fix, Evelyn and Joseph L Hodges (1951). "Discriminatory analysis". In: *Nonparametric discrimination: Small sample performance. Report A* 193008.

Forster, Jürgen (2002). "A linear lower bound on the unbounded error probabilistic communication complexity". In: *Journal of Computer and System Sciences* 65.4, pp. 612–625.

Globig, Christoph and Steffen Lange (1996). "Case-Based Representability of Classes of Boolean Functions". In: *12th European Conference on Artificial Intelligence, Budapest, Hungary, August 11-16, 1996, Proceedings*. Ed. by Wolfgang Wahlster. John Wiley and Sons, Chichester, pp. 117–121.

Goldmann, Mikael (1994). "Communication complexity and lower bounds for threshold circuits". In: *Theoretical advances in neural computation and learning*, pp. 85–125.

Goldmann, Mikael and Marek Karpinski (1993). "Simulating threshold circuits by majority circuits". In: *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pp. 551–560.

Gröger, Hans Dietmar and György Turán (1991). "On linear decision trees computing Boolean functions". In: *International Colloquium on Automata, Languages, and Programming*. Springer, pp. 707–718.

Guliyev, Namig J. and Vugar E. Ismailov (2018). "On the approximation by single hidden layer feedforward neural networks with fixed weights". In: *Neural Networks* 98, pp. 296–304. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2017.12.007. URL: https://www.sciencedirect.com/science/article/pii/S0893608017302927.

Guy, Richard (2004). *Unsolved problems in number theory*. Vol. 1. Springer Science & Business Media.

Hajnal, András et al. (1993). "Threshold circuits of bounded depth". In: *Journal of Computer and System Sciences* 46.2, pp. 129–154.

Hajnal, Péter, Zhihao Liu, and György Turán (2022). "Nearest Neighbor Representations of Boolean Functions". In: *Information and Computation* 285, p. 104879.

Halász, Gábor (1977). "Estimates for the concentration function of combinatorial number theory and probability". In: *Periodica Mathematica Hungarica* 8.3-4, pp. 197–211.

Hancock, Thomas et al. (1996). "Lower bounds on learning decision lists and trees". In: *Information and Computation* 126.2, pp. 114–122.

Hansen, Kristoffer Arnsfelt and Vladimir V Podolskii (2010). "Exact threshold circuits". In: *2010 IEEE 25th Annual Conference on Computational Complexity*. IEEE, pp. 270–279.

Hansen, Kristoffer Arnsfelt and Vladimir V Podolskii (2015). “Polynomial threshold functions and Boolean threshold circuits”. In: *Information and Computation* 240, pp. 56–73.

Håstad, Johan (1986). “Almost optimal lower bounds for small depth circuits”. In: *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pp. 6–20.

– (1994). “On the size of weights for threshold gates”. In: *SIAM Journal on Discrete Mathematics* 7.3, pp. 484–492.

Håstad, Johan, Stasys Jukna, and Pavel Pudlák (1995). “Top-down lower bounds for depth-three circuits”. In: *Computational Complexity* 5.2, pp. 99–112.

Hofmeister, Thomas (1996). “A note on the simulation of exponential threshold weights”. In: *International Computing and Combinatorics Conference*. Springer, pp. 136–141.

Hornik, Kurt (1991). “Approximation capabilities of multilayer feedforward networks”. In: *Neural networks* 4.2, pp. 251–257.

Hubara, Itay et al. (2016). “Binarized neural networks”. In: *Advances in neural information processing systems* 29, pp. 4107–4115.

Impagliazzo, Russell and Ryan Williams (2010). “Communication complexity with synchronized clocks”. In: *2010 IEEE 25th Annual Conference on Computational Complexity*. IEEE, pp. 259–269.

Indyk, Piotr and Rajeev Motwani (1998). “Approximate nearest neighbors: towards removing the curse of dimensionality”. In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613.

Jacob, Benoit et al. (2018). “Quantization and training of neural networks for efficient integer-arithmetic-only inference”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2704–2713.

Johnson, Jeff, Matthijs Douze, and Hervé Jégou (2019). “Billion-scale similarity search with GPUs”. In: *IEEE Transactions on Big Data* 7.3, pp. 535–547.

Karingula, Sankeerth Rao and Shachar Lovett (2021). “Singularity of Random Integer Matrices with Large Entries”. In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Kautz, William H (1961). “The realization of symmetric switching functions with linear-input logical elements”. In: *IRE Transactions on Electronic Computers* 3, pp. 371–378.

Klivans, Adam R, Rocco A Servedio, and Dana Ron (2006). “Toward Attribute Efficient Learning of Decision Lists and Parities.” In: *Journal of Machine Learning Research* 7.4.

Kushilevitz, Eyal, Rafail Ostrovsky, and Yuval Rabani (1998). "Efficient search for approximate nearest neighbor in high dimensional spaces". In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. STOC '98. Dallas, Texas, USA: Association for Computing Machinery, pp. 614–623. ISBN: 0897919629. DOI: 10.1145/276698.276877. URL: https://doi.org/10.1145/276698.276877.

Lewis, Patrick et al. (2020). "Retrieval-augmented generation for knowledge-intensive nlp tasks". In: *Advances in Neural Information Processing Systems* 33, pp. 9459–9474.

Li, Fengfu, Bo Zhang, and Bin Liu (2016). "Ternary weight networks". In: *arXiv preprint arXiv:1605.04711*.

Lu, Zhou et al. (2017). "The expressive power of neural networks: A view from the width". In: *Advances in neural information processing systems* 30.

Lupanov, Oleg B (1958). "On a method of circuit synthesis". In: *Izvestia VUZ* 1, pp. 120–140.

Maass, Wolfgang (1993). "Bounds for the computational power and learning complexity of analog neural nets". In: *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pp. 335–344.

Maiorov, Vitaly and Allan Pinkus (1999). "Lower bounds for approximation by MLP neural networks". In: *Neurocomputing* 25.1-3, pp. 81–91.

Malkov, Yu A and Dmitry A Yashunin (2018). "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs". In: *IEEE transactions on pattern analysis and machine intelligence* 42.4, pp. 824–836.

McCulloch, Warren and Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". In: *Bulletin of Mathematical Biophysics* 5, pp. 127–147.

Meyer Jr, Carl D (1973). "Generalized inversion of modified matrices". In: *SIAM journal on applied mathematics* 24.3, pp. 315–323.

Minnick, Robert C (1961). "Linear-input logic". In: *IRE Transactions on Electronic Computers* 1, pp. 6–16.

Minsky, Marvin and Seymour A Papert (2017). *Perceptrons, reissue of the 1988 expanded edition with a new foreword by Léon Bottou: an introduction to computational geometry*. MIT press.

Muroga, S (1959). "Logical elements on majority decision principle and complexity of their circuit". In: *Communications of the ACM*. Vol. 2. 7. Assoc Computing Machinery 1515 Broadway, New York, NY 10036, pp. 14–14.

Muroga, Saburo (1971). *Threshold logic and its applications*.

Murphy, Owen J (1990). "Nearest neighbor pattern classification perceptrons". In: *Proceedings of the IEEE* 78.10, pp. 1595–1598.

Park, Sejun et al. (2021). “Minimum Width for Universal Approximation”. In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=O-XJwyoIF-k`.

Paturi, Ramamohan and Michael E Saks (1990). “On threshold circuits for parity”. In: *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*. IEEE, pp. 397–404.

Paturi, Ramamohan and Janos Simon (1986). “Probabilistic communication complexity”. In: *Journal of Computer and System Sciences* 33.1, pp. 106–123.

Reimers, Nils and Iryna Gurevych (2019). “Sentence-bert: Sentence embeddings using siamese bert-networks”. In: *arXiv preprint arXiv:1908.10084*.

Rivest, Ronald L (1987). “Learning decision lists”. In: *Machine learning* 2, pp. 229–246.

Rosenblatt, Frank (1958). “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6, p. 386.

Roychowdhury, Vwani P., Alon Orlitsky, and Kai-Yeung Siu (1994). “Lower bounds on threshold and related circuits via communication complexity”. In: *IEEE Transactions on Information Theory* 40.2, pp. 467–474.

Rudelson, Mark and Roman Vershynin (2008). “The Littlewood–Offord problem and invertibility of random matrices”. In: *Advances in Mathematics* 218.2, pp. 600–633.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). “Learning representations by back-propagating errors”. In: *nature* 323.6088, pp. 533–536.

Salzberg, Steven et al. (1995). “Best-case results for nearest-neighbor learning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.6, pp. 599–608.

Satoh, Ken (1998). “Analysis of case-based representability of Boolean functions by monotone theory”. In: *International Conference on Algorithmic Learning Theory*. Springer, pp. 179–190.

Shannon, Claude E (1938). “A symbolic analysis of relay and switching circuits”. In: *Electrical Engineering* 57.12, pp. 713–723.

– (1949). “The synthesis of two-terminal switching circuits”. In: *The Bell System Technical Journal* 28.1, pp. 59–98.

Siegel, Carl L (2014). “Über einige anwendungen diophantischer approximationen”. In: *On Some Applications of Diophantine Approximations*. Springer, pp. 81–138.

Siu, Kai-Yeung and Jehoshua Bruck (1991). “On the power of threshold circuits with small weights”. In: *SIAM Journal on Discrete Mathematics* 4.3, pp. 423–435.

Siu, Kai-Yeung, Vwani P. Roychowdhury, and Thomas Kailath (1991). “Depth-size tradeoffs for neural computation”. In: *IEEE Transactions on Computers* 40.12, pp. 1402–1412.

Spielman, Daniel A (1992). “Computing Arbitrary Symmetric Functions”. In: URL: https://cpsc.yale.edu/sites/default/files/files/tr906.pdf.

Stockmeyer, Larry J (1976). “On the combinational complexity of certain symmetric Boolean functions”. In: *Mathematical Systems Theory* 10.1, pp. 323–336.

Strickland, Lloyd (2020). “Leibniz on Number Systems”. In: *Handbook of the History and Philosophy of Mathematical Practice*. Ed. by Bharath Sriraman. Cham: Springer International Publishing, pp. 1–31. ISBN: 978-3-030-19071-2. DOI: 10.1007/978-3-030-19071-2_90-1. URL: https://doi.org/10.1007/978-3-030-19071-2_90-1.

Tay, Thian Fatt and Chip-Hong Chang (2015). “A non-iterative multiple residue digit error detection and correction algorithm in RRNS”. In: *IEEE transactions on computers* 65.2, pp. 396–408.

Turán, György and Farrokh Vatan (1997). “Linear decision lists and partitioning algorithms for the construction of neural networks”. In: *Foundations of Computational Mathematics: Selected Papers of a Conference Held at Rio de Janeiro, January 1997*. Springer, pp. 414–423.

Uchizawa, Kei and Eiji Takimoto (2015). “Lower bounds for linear decision trees with bounded weights”. In: *International Conference on Current Trends in Theory and Practice of Informatics*. Springer, pp. 412–422.

Vardi, Gal and Ohad Shamir (2020). “Neural Networks with Small Weights and Depth-Separation Barriers”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 19433–19442. URL: https://proceedings.neurips.cc/paper/2020/file/e1fe6165cad3f7f3f57d409f78e4415f-Paper.pdf.

Weber, Roger, Hans-Jörg Schek, and Stephen Blott (1998). “A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces”. In: *VLDB*. Vol. 98, pp. 194–205.

Wilfong, Gordon (1991). “Nearest neighbor problems”. In: *Proceedings of the seventh annual symposium on Computational Geometry*, pp. 224–233.

Appendix A

THE DECODING ALGORITHM FOR THE VECTORS MAPPED BY THE EQ MATRIX IN THEOREM 10

Suppose that $A_k x = z$ is given where A_k is an $m \times n$ EQ matrix given in Theorem 10 starting with $A_0 = \begin{bmatrix} 1 \end{bmatrix}$. One can obtain a linear time decoding algorithm in n by exploiting the recursive structure of the construction. Let us partition x and z in the way given in (2.19). It is clear that $x^{(3)} = [z^{(1)} + z^{(2)}]_2$. Also, after computing $x^{(3)}$, we find that $A_{k-1}x^{(1)} = (z^{(1)} + z^{(2)} - x^{(3)})/2$ and $A_{k-1}x^{(2)} = (z^{(1)} - z^{(2)} - x^{(3)})/2$. These operations can be done in $O(m_{k-1})$ time complexity. Let $T(m)$ denote the time to decode $z \in \mathbb{Z}^m$. Then, $T(m) = 2T(m/2) + O(m)$ and by the Master Theorem, $T(m) = O(m \log m) = O(n)$.

An Example for the Decoding: Consider the following system Ax = b:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 4 \\ -2 \\ -1 \\ 0 \end{bmatrix}$$

Here is the first step of the algorithm:

$$a = \begin{bmatrix} 4 \\ -2 \end{bmatrix}, \quad b = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} x_7 \\ x_8 \end{bmatrix} = [a + b]_2 = \begin{bmatrix} [3]_2 \\ [-2]_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Then, we obtain the two following systems:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{a + b - \begin{bmatrix} 1 \\ 0 \end{bmatrix}}{2} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} = \frac{a - b - \begin{bmatrix} 1 \\ 0 \end{bmatrix}}{2} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

For the first system, we find that $x_3 = [1 - 1]_2 = 0$. Then,

$$x_1 = \frac{1 - 1 - 0}{2} = 0 \tag{A.1}$$
$$x_2 = \frac{1 + 1 - 0}{2} = 1 \tag{A.2}$$

For the second system, we similarly find that $x_6 = [2 - 1]_2 = 1$. Then,

$$x_4 = \frac{2 - 1 - 1}{2} = 0 \tag{A.3}$$
$$x_5 = \frac{2 + 1 - 1}{2} = 1 \tag{A.4}$$

Thus, we have $x = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}^T$.

Appendix B

A MODULAR ARITHMETICAL PROPERTY OF THE CRT MATRIX

Consider the following equation with powers of two in 8 variables and the corresponding 4×8 CRT matrix, say A.

$$w_b^T x = \sum_{i=1}^{8} 2^{i-1} x_i = 0 \tag{B.1}$$

$$A = \begin{bmatrix} [2^0]_3 & [2^1]_3 & [2^2]_3 & [2^4]_3 & [2^5]_3 & [2^6]_3 & [2^7]_3 \\ [2^0]_5 & [2^1]_5 & [2^2]_5 & [2^4]_5 & [2^5]_5 & [2^6]_5 & [2^7]_5 \\ [2^0]_7 & [2^1]_7 & [2^2]_7 & [2^4]_7 & [2^5]_7 & [2^6]_7 & [2^7]_7 \\ [2^0]_{11} & [2^1]_{11} & [2^2]_{11} & [2^4]_{11} & [2^5]_{11} & [2^6]_{11} & [2^7]_{11} \end{bmatrix} \tag{B.2}$$

$$= \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\ 1 & 2 & 4 & 3 & 1 & 2 & 4 & 3 \\ 1 & 2 & 4 & 1 & 2 & 4 & 1 & 2 \\ 1 & 2 & 4 & 8 & 5 & 10 & 9 & 7 \end{bmatrix}_{4\times 8} \tag{B.3}$$

For the prime p_i, the elements in the row i are congruent to the elements of w_b. The i^{th} element of the Ax vector should be divisible by p_i whenever $w_b^T x = 0$. For instance, one can pick $x = \begin{bmatrix} 2 & 1 & 1 & 3 & 0 & 1 & -1 & 0 \end{bmatrix}^T$ as a solution for $w_b^T x = 0$ and we see that $Ax = \begin{bmatrix} 12 & 15 & 14 & 33 \end{bmatrix}^T = \begin{bmatrix} 4 \cdot 3 & 3 \cdot 5 & 2 \cdot 7 & 3 \cdot 11 \end{bmatrix}^T$. This property is essential in the construction of small weight depth-2 circuits for arbitrary threshold functions while the RMDS_q matrices do not behave in this manner necessarily.

Appendix C

EXACT COVER REPRESENTATIONS

Exact Cover Representation (denoted by EC) of a Boolean function $f(X)$ is a set of hyperplanes required to *exactly cover* all binary vectors such that $f(X) = 1$. Obviously, the exact cover representation sizes for $f(X)$ and its complement $\overline{f}(X)$ need not to be the same. We define the *Exact Cover Complexity* to be the summation of the sizes of the exact cover representations of $f(X)$ and $\overline{f}(X)$.

Definition 14. *Let $\mathcal{H} = \{H_1, \dots, H_m\}$ be a set of finite number of hyperplanes in $\mathbb{R}^n$ and $f(X)$ be a Boolean function where $X \in \{0,1\}^n$. Then, $\mathcal{H}$ is an* Exact Cover Representation of f *if*

$$\{X \in \{0,1\}^n | f(X) = 1\} = \bigcup_{i=1}^{m} H_i \cap \{0,1\}^n \tag{C.1}$$

We say that $\mathcal{H}$ covers f *if $\mathcal{H}$ is an exact cover representation of f. The* Exact Cover Complexity of f *is denoted by* $\mathrm{EC}(f)$ *and* $\mathrm{EC}(f) = |\mathcal{H}| + |\overline{\mathcal{H}}|$ *has minimal value where $\mathcal{H}$ covers $f(X) = 1$ and $\overline{\mathcal{H}}$ covers $f(X) = 0$.*

Given the exact cover representation as a set of hyperplanes $\{H_1, \dots, H_m\}$, it is straight-forward compute $f(X)$ by a depth-2 circuit of $\mathrm{OR}_m \circ \mathrm{ELT}_n$. Therefore, for a Boolean function $f(X)$, if the number of exact threshold gates in the $\mathrm{OR}_m \circ \mathrm{ELT}_n$ is lower bounded by m, we have a lower bound on the EC complexity. The converse does not hold since $\mathrm{OR}_m \circ \mathrm{ELT}_n$ is not closed under complement operation. An upper bound on m for $\mathrm{OR}_m \circ \mathrm{ELT}_n$ circuits cannot be used to bound EC complexity in general because the behavior of the complement of the function could be different.

Now, we give some important examples of exact cover representations. One can cover the whole Boolean hypercube $\{0,1\}^n$ using 2 hyperplanes, e.g., $x_1 = 0$ and $x_1 = 1$. This is clearly optimal since a single hyperplane is not sufficient to cover the whole hypercube. In addition, by using powers-of-two, one can provide parallel hyperplanes to cover any set $f(X) = 1$ and $f(X) = 0$ using $\sum_{i=1}^{n} 2^{i-1} x_i \in \{0, \dots, 2^n - 1\}$. Therefore, $EC(f) \leq 2^n$ in general. The first non-trivial results for exact cover complexity starts by covering $\{0,1\}^n \setminus \{0\}$ using n hyperplanes, which is optimal (Alon and Füredi, 1993). Achieving this complexity value is easy,

$x_1 + \cdots + x_n = 0$ covers 0 and $x_i = 1$ for $i \in \{1, \ldots, n\}$ covers the $\{0,1\}^n \setminus \{0\}$. Proving a corresponding lower bound requires some work using multilinear polynomials and Nullstellensatz (Alon and Füredi, 1993).

Since then, a number of related results on EC complexity is obtained (Aaronson et al., 2021; Clifton and Huang, 2020; Diamond and Yehudayoff, 2022). A striking result is that the $2n$-input disjointedness function (i.e. $\overline{\mathrm{DISJ}}_{2n}(X) = \bigvee_{i=1}^{n}(x_i \wedge y_i)$) has $\mathrm{EC}(\mathrm{DISJ}_{2n}(X)) = 2^{\Omega(n)}$ (Diamond and Yehudayoff, 2022). One direction is easy since this is only a circuit of $\mathrm{OR}_n \circ \mathrm{AND}_2$. The non-trivial direction is to show that one requires $2^{\Omega(n)}$ hyperplanes to exactly cover the hypercube where $\mathrm{DISJ}_{2n}(X) = 1$. They use anti-concentration inequalities to prove this result. In contrast, DISJ_{2n} has a n-anchor constant resolution NN representation since it is a member of AND $\circ$ LT and we have a construction for this class.

For a symmetric Boolean function $f(X)$, using all the hyperplanes $|X| = k$ where $k \in \{0, \ldots, n\}$, it can be shown that $\mathrm{EC}(f) \leq n+1$ both for $f(X) = 1$ and $f(X) = 0$. Also, for a linear threshold function $f(X)$, $\mathrm{EC}(f) = O(n^2 \log n)$ by simply using results from (Amano and Maruoka, 2005; Hansen and V. V. Podolskii, 2010), which we do not prove here (see the proof of Theorem 7 in Hansen and V. V. Podolskii, 2010).

To compute a Boolean function $f(X)$, its NN representation and a circuit of OR $\circ$ AND $\circ$ LT can be used (see Section 3.3). NN representations are closed under complement operation (one can reverse the labeling of the anchors). EC representations, on the other hand, are not closed under complement operation because the complement of the function could require a large number of hyperplanes (e.g. DISJ function). Since the class OR $\circ$ AND $\circ$ LT seems to be more expressive than OR $\circ$ ELT, we conjecture that the $\mathrm{EC}(f)$ is always larger than $\mathrm{NN}(f)$ by a constant factor c. A weaker version could apply if there is a polynomial dependency between $\mathrm{EC}(f)$ and $\mathrm{NN}(f)$. To attack this conjecture, it is essential to analyze the NN representations of OR $\circ$ ELT circuits, as we do in Section 3.5.

Conjecture 5 (Strong Version)**.** *Let $f(X)$ be a Boolean function where $X \in \{0,1\}^n$. Then,* $\mathrm{NN}(f) \leq c\mathrm{EC}(f)$ *for some constant $c > 0$.*

Conjecture 6 (Weak version)**.** *Let $f(X)$ be a Boolean function where $X \in \{0,1\}^n$. Then,* $\mathrm{NN}(f) = O(poly(\mathrm{EC}(f)))$.

Appendix D

A COUNTEREXAMPLE FOR THE PARITY-BASED EXTENSIONS FOR NN REPRESENTATIONS OF SYMMETRIC BOOLEAN FUNCTIONS

Let $f(X)$ be the function given in Eq. (3.134). For the sake of contradiction, we assume that there exist a PARITY-based extension to this symmetric function, namely, $a_{ij} = a_{ik}$ for any $i \in \{1, \dots, 5\}$ and $j, k \in \{1, \dots, 8\}$.

$$\begin{array}{c|c} |X| & f(X) \\ \hline 0 & 1 \\ 1 & \mathbf{0} \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \\ 6 & 1 \\ 7 & \mathbf{0} \\ 8 & 1 \end{array} \qquad \begin{array}{l} I_1 = 0 \\ I_2 = 1 \\ \\ \\ \\ \\ I_3 = 6 \\ I_4 = 7 \\ I_5 = 8 \end{array} \tag{D.1}$$

We can rewrite the necessary and sufficient conditions given in Lemma 8. We also know that $a_{ij} + a_{(i-1)j}$ (or $a_{ij} - a_{(i-1)j}$) has the same value for all $j \in \{1, \dots, 8\}$ and $i \in \{2, \dots, 5\}$. Then, we get

$$\begin{aligned} I_{i-1}\big(a_{i1} - a_{(i-1)1}\big) &< 4\big(a_{i1} - a_{(i-1)1}\big)\big(a_{i1} + a_{(i-1)1}\big) \\ &< (I_{i-1} + 1)\big(a_{i1} - a_{(i-1)1}\big) \end{aligned} \tag{D.2}$$

$$\frac{I_{i-1}}{4} < \big(a_{i1} + a_{(i-1)1}\big) < \frac{I_{i-1} + 1}{4} \tag{D.3}$$

Recall that Proposition 2 still applies here. Therefore, $a_{1j} < a_{2j} < a_{3j} < a_{4j} < a_{5j}$ for all $j \in \{1, \dots, 8\}$. More explicitly, we have the following system of inequalities

$$a_{11} < a_{21} < a_{31} < a_{41} < a_{51} \tag{D.4}$$

$$0 < a_{21} + a_{11} < \frac{1}{4} \tag{D.5}$$

$$\frac{1}{4} < a_{31} + a_{21} < \frac{2}{4} \tag{D.6}$$

$$\frac{6}{4} < a_{41} + a_{31} < \frac{7}{4} \tag{D.7}$$
$$\frac{7}{4} < a_{51} + a_{41} < \frac{8}{4} \tag{D.8}$$

Firstly, we multiply Eq. (D.5) and (D.7) with -1 and sum all Eq. (D.5),(D.6),(D.7), and (D.8). Secondly, we multiply Eq. (D.6) with -1 and add it to Eq. (D.7). Thus,

$$0 < a_{51} - a_{11} < 1 \tag{D.9}$$
$$1 < a_{41} - a_{21} < 1.5 \tag{D.10}$$

which is inconsistent with Eq. (D.4).

The following is a construction for the function in Eq. (3.134) using our techniques. We use 2 significant digits to fit the matrix here.

$$\begin{bmatrix} 17.78 & 2.28 & -5.72 & -21.22 & -1.53 & -1.53 & -1.53 & -1.53 \\ 19.28 & 3.28 & -4.72 & -20.22 & -0.53 & -0.53 & -0.53 & -0.53 \\ 20.28 & 4.78 & -3.72 & -19.22 & 0.47 & 0.47 & 0.47 & 0.47 \\ 21.28 & 5.78 & -2.22 & -18.22 & 1.47 & 1.47 & 1.47 & 1.47 \\ 22.28 & 6.78 & -1.22 & -16.72 & 2.47 & 2.47 & 2.47 & 2.47 \end{bmatrix} \tag{D.11}$$

Appendix E

EXPLICIT NN REPRESENTATIONS FOR THE FUNCTIONS IN TABLE 1.2

Although our results are usually theoretical, for the functions described in Table 1.2 we provide some numerical examples for their NN representations using fixed number of inputs. These examples illustrate resolution-size trade-off in the analysis of NN representations of Boolean functions.

Function	**NN Representation**		
	Size		Resolution
	Upper Bound	Lower Bound	
EQ_{2n}	3	3	$O(n)$
EQ_{2n}	$2n+1$	$\Omega(\sqrt{n})$	$O(1)$
EQ_{2n}	$O(n/\log n)$	$\Omega(\sqrt{n/\log n})$	$O(\log n)$
COMP_{2n}	$2n$	$\Omega(\sqrt{n/\log n})$	$O(\log n)$
OMB_{n}	$n+1$		$O(\log n)$
$\mathrm{IP2}_{2n}$	2^n	$2^{n/2*}$	$O(1)$

*The lower bound is for arbitrary resolution

Table E.1: A summary of the NN representations (Summarized from Table 1.2)

E.1 The NN Representations for the EQ function

In this section, we give low resolution NN representations for the EQ function based on Corollary 35.1. One can verify that the following is a 3-anchor NN representation for the EQ function using the weights as the powers of two.

$$\begin{bmatrix} 0.500000 & 0.500000 & 0.500000 & 0.500000 & 0.500000 & 0.500000 \\ 0.595238 & 0.547619 & 0.523810 & 0.404762 & 0.452381 & 0.476190 \\ 0.404762 & 0.452381 & 0.476190 & 0.595238 & 0.547619 & 0.523810 \end{bmatrix} \tag{E.1}$$

The following is a 7-anchor NN representation for the 6-input EQ function with input $(X,Y) = (X_3, X_2, X_1, Y_3, Y_2, Y_1)$ using the identity EQ matrix.

$$\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 1 & 0.5 & 0.5 & 0 & 0.5 & 0.5 \\ 0 & 0.5 & 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 & 0.5 & 0 \\ 0.5 & 0.5 & 0 & 0.5 & 0.5 & 1 \end{bmatrix}_{7\times 6} \tag{E.2}$$

In contrast, we can use the following 2×3 EQ matrix for the EQ function with 6 inputs (recall that EQ matrices are for $(X - Y) \in \{-1, 0, 1\}^n$ while we need $(X, Y) \in \{0, 1\}^{2n}$ to construct the anchors).

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \tag{E.3}$$

This matrix gives us a 5-anchor NN representation for the 6-input EQ function. Here, we give the entries of the matrix for 3 significant digits but the construction can be obtained simply by using Theorem 35.

$$\begin{bmatrix} 0.500 & 0.500 & 0.500 & 0.500 & 0.500 & 0.500 \\ 0.667 & 0.667 & 0.667 & 0.333 & 0.333 & 0.333 \\ 0.750 & 0.250 & 0.500 & 0.250 & 0.750 & 0.500 \\ 0.333 & 0.333 & 0.333 & 0.667 & 0.667 & 0.667 \\ 0.250 & 0.750 & 0.500 & 0.750 & 0.250 & 0.500 \end{bmatrix}_{5\times 6} \tag{E.4}$$

E.2 Low Resolution NN representations for the COMP and OMB

For the COMP, we give an 6-anchor NN representation for 6 inputs. Here, we take the input vector as $(X, Y) = (X_3, X_2, X_1, Y_3, Y_2, Y_1)$ where X_3 and Y_3 are the most significant bits. We give 4 significant digits for the anchor matrix.

$$\begin{bmatrix} 1.0000 & 0.5000 & 0.5000 & 0.0000 & 0.5000 & 0.5000 \\ -0.0833 & 0.5000 & 0.5000 & 1.0833 & 0.5000 & 0.5000 \\ 0.5000 & 1.1667 & 0.5000 & 0.5000 & -0.1667 & 0.5000 \\ 0.5000 & -0.2500 & 0.5000 & 0.5000 & 1.2500 & 0.5000 \\ 0.5000 & 0.5000 & 1.3333 & 0.5000 & 0.5000 & -0.3333 \\ 0.5000 & 0.5000 & -0.4167 & 0.5000 & 0.5000 & 1.4167 \end{bmatrix}_{6\times 6} \tag{E.5}$$

For the OMB, we first give 8-input NN representation with 8 anchors according to Theorem 37. We exclude the all-zero anchor since the smallest norm anchor has already red label. The input vector is assumed to be $(X_1, X_2, \ldots, X_8)$ where X_1 is the most significant bit.

$$\begin{bmatrix} 1.000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.875 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.750 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.625 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.375 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.250 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.125 \end{bmatrix}_{8\times 8} \tag{E.6}$$

And similarly, we give a 10-anchor NN representation for the 9-input OMB.

$$\begin{bmatrix} 1.000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.889 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.778 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.667 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.556 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.444 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.333 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.222 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.111 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{10\times 9} \tag{E.7}$$

E.3 A Constant Resolution NN Representation for the IP2

Based on Theorem 30, we construct the following 8-anchor NN representation for the 6-input IP2 with $(X, Y) = (X_1, Y_1, X_2, Y_2, X_3, Y_3)$ by using $\text{IP2}_{2n} = \text{PARITY}_n \circ \text{AND}_2$.

$$\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 1 & 1 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 1 & 1 \\ 0.5 & 0.5 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0.5 & 0.5 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0.5 & 0.5 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}_{2^3\times 6} \tag{E.8}$$