Convex Geometry of ReLU-Layers, Injectivity on the Ball and Local Reconstruction

Daniel Haider [1 2] **Martin Ehler** [2] **Peter Balazs** [1]

Abstract

The paper uses a frame-theoretic setting to study the injectivity of a ReLU-layer on the closed ball of $\mathbb{R}^n$ and its non-negative part. In particular, the interplay between the radius of the ball and the bias vector is emphasized. Together with a perspective from convex geometry, this leads to a computationally feasible method of verifying the injectivity of a ReLU-layer under reasonable restrictions in terms of an upper bound of the bias vector. Explicit reconstruction formulas are provided, inspired by the duality concept from frame theory. All this gives rise to the possibility of quantifying the invertibility of a ReLU-layer and a concrete reconstruction algorithm for any input vector on the ball.

1. Introduction

The Rectified Linear Unit $\mathrm{ReLU}(s) = \max(0, s)$, $s \in \mathbb{R}$ has become indispensable in modern neural network architecture. It is applied component-wise on the output of an affine linear function $Ax - b$, comprising of the multiplication by a weight matrix A and the shift by a bias vector b. The combined mapping is called a *ReLU-layer*. This has proven to be a simple, yet effective non-linear mapping to handle fundamental problems in the training of deep neural networks well (Glorot et al., 2011; Krizhevsky et al., 2012; Goodfellow et al., 2016; Nair & Hinton, 2010). Despite its simplicity, yet, the ReLU function still hides some mysteries and is an active topic of research (Dittmer et al., 2020).

Recently, invertible network architectures have been getting a lot of attention due to their increased interpretability and the possibility of reversing the forward process analytically, which is especially interesting in a generative setting. This found many applications in the context of normalizing flows, offering exact and efficient likelihood estimations (Dinh et al., 2017; Donahue et al., 2017). Mathematically speaking, the forward process in such an invertible architecture must be *injective*, guaranteeing the existence of a *left-inverse* that allows perfect reconstruction of any input. A ReLU-layer is a mapping that is designed to provide sparse output. Hence, its injectivity is an interesting property that has been tackled from a theoretical point of view only little in the literature. Bruna et al. characterized a ReLU-layer to be injective in terms of an admissibility condition for index sets and proved a bi-Lipschitz stability condition for an injective ReLU-layer, see Proposition 2.2 in (Bruna et al., 2014). Just recently, Puthawala et al. formulated a condition in terms of spanning sets that is equivalent to the one in (Bruna et al., 2014) (with a slight modification) and describes the injectivity of ReLU-networks consisting of many ReLU-layers see Theorem 2 in (Puthawala et al., 2022). Both conditions, however, are not applicable to verify the injectivity of a ReLU-layer for a given weight matrix in practice. The presented work provides exactly that. We found the convex geometry of the weight matrix to play an essential role in the injectivity analysis for the associated ReLU-layer, using a concept that Behrmann et al. introduced in Theorem 4 of (Behrmann et al., 2018). The geometrical perspective helps profoundly to strengthen the intuition on the effect of the ReLU function. It allows to formulate a computationally feasible method to give a sufficient condition for injectivity. This shall contribute to the enhancement of the interpretability of neural networks in terms of a way to quantify the invertibility of a ReLU-layer with corresponding exact reconstruction formulas. Aiming to set a rigorous foundation for future work on this topic, we formulate all results in an abstract mathematical manner, using the language of *frame theory* which we find to be especially well-suited.

In Section 2 we interpret a ReLU-layer by means of frame theory and motivate the restriction to the ball. Section 3 is dedicated to the injectivity of a ReLU-layer theoretically. In Section 4 we introduce a method to obtain an upper bound for all biases, such that the corresponding ReLU-layer is injective on the ball and its non-negative part. Explicit reconstruction formulas are stated. Finally, Section 5 demonstrates how the method can be used to analyze the injectivity behavior of a ReLU-layer in numerical experiments.

[1]Acoustics Research Institute, Vienna, Austria [2]University of Vienna, Department of Mathematics, Austria. Correspondence to: Daniel Haider <daniel.haider@oeaw.ac.at>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

2. Mathematical Context

2.1. Neural Networks meet Frame Theory

The goal of this section is to link abstract frame theory with deep learning. We want to particularly emphasize that frames are a well-suited concept for the mathematical analysis of neural networks, not only in terms of notation but also due to its long usage in signal processing which is tied closely to deep learning. In this sense, we build our work upon notation and tools from frame theory for $\mathbb{R}^n$, c.f. (Balazs, 2008; Casazza & Kutyniok, 2012). We shall write

$$X = (x_i)_{i\in I} \subseteq \mathbb{R}^n \quad \text{with} \quad |I| = m \geq n$$

to refer to a collection of m vectors $x_1, \dots, x_m$ in $\mathbb{R}^n$. Denoting the usual inner product on $\mathbb{R}^n$ as $\langle \cdot, \cdot \rangle$ we say that X constitutes a *frame* for $\mathbb{R}^n$ with *frame elements* x_i, if there are constants $0 < A \leq B < \infty$, such that

$$A \cdot \|x\|^2 \leq \sum_{i\in I} |\langle x, x_i \rangle|^2 \leq B \cdot \|x\|^2 \tag{1}$$

holds for all $x \in \mathbb{R}^n$. The constants A, B are called lower and upper frame bounds for X. In $\mathbb{R}^n$, a frame is equivalent to a spanning set. The bounds A, B become important, if one is interested in the numerical properties of the operators associated with a frame: the *analysis operator*

$$\begin{aligned} C : \mathbb{R}^n &\to \mathbb{R}^m \\ x &\mapsto (\langle x, x_i \rangle)_{i\in I}\,, \end{aligned}$$

its adjoint, the *synthesis operator*

$$\begin{aligned} D : \mathbb{R}^m &\to \mathbb{R}^n \\ (c_i)_{i\in I} &\mapsto \sum_{i\in I} c_i \cdot x_i, \end{aligned}$$

and the concatenation of analysis, followed by synthesis, the *frame operator*

$$\begin{aligned} S : \mathbb{R}^n &\to \mathbb{R}^n \\ x &\mapsto \sum_{i\in I} \langle x, x_i \rangle \cdot x_i. \end{aligned}$$

If X is a frame, then C is injective, D surjective, and S bijective. In $\mathbb{R}^n$ all the above operators are realized via left-multiplication of x with a corresponding matrix. In this sense, the analysis operator C can be identified with the $m \times n$ matrix

$$C = \begin{pmatrix} -x_1- \\ \vdots \\ -x_m- \end{pmatrix}.$$

For the synthesis operator, we have that $D = C^\top$. Recall that in matrix terminology, injectivity, and surjectivity relate to the corresponding matrix having full rank. Hence, if the weight matrix of a layer in a neural network has full rank, then it can be interpreted as the analysis operator of the frame consisting of its row vectors if $m \geq n$ and as the synthesis operator of the frame consisting of its column vectors if $m \leq n$. At the initialization of a neural network, the weight matrices are commonly set to be Gaussian i.i.d. matrices known to have full rank with probability 1 (Mehta, 2004). Hence, one can be (almost) sure to start the training with the rows, resp. columns of the weight matrices to constitute frames. Here, we concentrate on the case where $m \geq n$ and refer to such a layer as *redundant*.

The matrix associated with the frame operator is $S = DC$. It can be used to construct the *canonical dual frame* for X, given by $\tilde{X} = \left(S^{-1}x_i\right)_{i\in I}$. Denoting $\tilde{D}$ as the associated synthesis operator leads to the canonical frame decomposition of $x \in \mathbb{R}^n$ by X,

$$x = S^{-1}S = \sum_{i\in I} \langle x, x_i \rangle \cdot S^{-1}x_i = \tilde{D}Cx. \tag{2}$$

In this way, (2) is equivalent to $\tilde{D}$ being a left-inverse of C, allowing perfect reconstruction of x from Cx. To reconstruct an input vector from the output of a ReLU-layer, we will construct a left-inverse for it exactly in the spirit of (2). Finally, one can find the minimal upper and the maximal lower frame bound in (1) via the largest and smallest eigenvalue of S respectively. The ratio $\frac{B}{A}$ of these bounds corresponds to the condition number of the linear mapping given by the analysis operator, hence the weight matrix of the network layer, indicating its numerical stability.

2.2. ReLU-layers as Non-linear Analysis Operators

In a frame-theoretic context, we define the ReLU-layer associated with a collection of vectors $X = (x_i)_{i\in I} \subseteq \mathbb{R}^n$ and a bias vector $\alpha \in \mathbb{R}^m$ as the non-linear mapping

$$\begin{aligned} C_\alpha : \mathbb{R}^n &\to \mathbb{R}^m \\ x &\mapsto (\mathrm{ReLU}(\langle x, x_i \rangle - \alpha_i))_{i\in I}\,. \end{aligned} \tag{3}$$

The notation C_α is chosen to reflect the link to the frame analysis operator C. Of course, this is equivalent to how a ReLU-layer is commonly denoted, $\mathrm{ReLU}(Cx - \alpha)$ where ReLU applies component-wise. For fixed x, the effect of the shift by the bias α and the ReLU function on the frame analysis can be interpreted as all frame elements with $\langle x, x_i \rangle < \alpha_i$ are set to be the zero-vector. According to this observation, we introduce the notation

$$I_x^\alpha := \{i \in I : \langle x, x_i \rangle \geq \alpha_i\}, \tag{4}$$

determining the index set associated with those frame elements which are not affected by the ReLU function for x. This perspective requires referring to sub-collections of frames very often. We write $X_L = (x_i)_{i\in L}$ for the sub-collection of X with respect to the index set $L \subseteq I$.

Analogously, we add L as a subscript to the operators associated with X_L, e.g. C_L is the analysis operator of X_L. Clearly, the case where $L = I_x^\alpha$ plays a central role.

2.3. Input Data on the Closed Ball

One of the core ideas in this paper is the restriction of C_α to the closed ball of radius $r > 0$ in $\mathbb{R}^n$, denoted by

$$\mathbb{B}_r = \{x \in \mathbb{R}^n : \|x\| \leq r\}.$$

We write $\mathbb{B} = \mathbb{B}_1$. Indeed, this is a very reasonable assumption when thinking of standard data normalization practices for neural networks (LeCun et al., 2012; Huang et al., 2023). It turns out that this restriction allows for a much richer analysis of the injectivity of C_α than on all of $\mathbb{R}^n$, in particular, involving the radius r. Furthermore, as the output of a ReLU-layer has only non-negative entries, hence lies within $\mathbb{R}^n_+$, the input domain of any ReLU-layer that applies to the output of a previous ReLU-layer on the ball lies within the non-negative part of $\mathbb{B}_r$, denoted by

$$\mathbb{B}_r^+ = \mathbb{B}_r \cap \mathbb{R}^n_+. \tag{5}$$

Similarly, we write $\mathbb{B}^+ = \mathbb{B} \cap \mathbb{R}^n_+$. The boundary of the unit ball, or equivalently, the $(n-1)$-sphere is denoted by

$$\mathbb{S} = \partial\mathbb{B} = \{x \in \mathbb{R}^n : \|x\| = 1\}.$$

3. Injectivity of C_α on $\mathbb{B}_r$

The ReLU-layer mapping C_α is - by design - non-linear, such that a condition for its injectivity will generally depend on the input. Fixing x, one notices that if the sub-collection $X_{I_x^\alpha}$ is a frame, then the analysis operator $C_{I_x^\alpha}$ is injective, which we will use to study the injectivity of C_α. For $\alpha \equiv 0$, Puthawala et al. refer to this property as "x having a directed spanning set" see Definition 1 in (Puthawala et al., 2022). In the following, we formulate this for general α and $K = \mathbb{B}_r$ in the context of frame theory.

Definition 3.1 (α-rectifying on $\mathbb{B}_r$)**.** A collection $X = (x_i)_{i \in I} \subseteq \mathbb{R}^n$ is called α-rectifying for $\alpha \in \mathbb{R}^m$ on $\mathbb{B}_r$ if for all $x \in \mathbb{B}_r$ the sub-collection $X_{I_x^\alpha} = (x_i)_{i \in I_x^\alpha}$ is a frame for $\mathbb{R}^n$.

An analogous definition can be formulated for $\mathbb{B}_r^+$. Unless explicitly stated, we always refer to $\mathbb{B}_r$ when writing that X is α-rectifying, since it covers the case $\mathbb{B}_r^+$.

In Lemma 2 of the same paper (Puthawala et al., 2022) the authors show that α-rectifying on $\mathbb{R}^n$ characterizes the injectivity of C_α. We revisit this characterization for $\mathbb{B}_r$ and $\mathbb{B}_r^+$. Again, the frame-theoretic formulation simplifies the statement significantly.

Theorem 3.2 (Injectivity of ReLU-layers on $\mathbb{B}_r$)**.** *Consider $X = (x_i)_{i \in I} \subseteq \mathbb{R}^n$, $\alpha \in \mathbb{R}^m$. If X is α-rectifying on $\mathbb{B}_r$ (resp. $\mathbb{B}_r^+$), then C_α is injective on $\mathbb{B}_r$ (resp. $\mathbb{B}_r^+$).*

A proof can be found in the appendix. Hence, we can shift the question of injectivity of C_α to the verification of the α-rectifying property for a given collection of vectors X.

Stability. Following the lines of (Bruna et al., 2014) and again, switching from $\mathbb{R}^n$ to $\mathbb{B}_r$, one can show that the injectivity of C_α on $\mathbb{B}_r$ implies frame-like inequalities analogous to (1), i.e. there are constants $0 < A_0 \leq B_0 < \infty$ such that

$$A_0 \cdot \|x\|^2 \leq \sum_{i \in I} |\mathrm{ReLU}\left(\langle x, x_i \rangle - \alpha_i\right)|^2 \leq B_0 \cdot \|x\|^2 \tag{6}$$

for all $x \in \mathbb{B}_r$. Here, A_0 can be chosen as the smallest eigenvalue and B_0 as the largest eigenvalue of all frame operators associated with the frames $X_{I_x^\alpha}$ with $x \in \mathbb{B}_r$.

Inclusiveness. It is clear that if X is α-rectifying, then X is α'-rectifying for all $\alpha' \leq \alpha$. Therefore, we call

α an *upper bias* for C_α if X is α-rectifying.

This perfectly reflects the role of the bias vector in a neural network: the larger the bias values, the more neurons are activated by the ReLU function, hence the "more injective" the ReLU-layer becomes in the sense that it is injective for a larger set of bias vectors. Therefore, it is of natural interest to find the largest possible upper bias for a given weight matrix. A unique maximal upper bias, however, does not exist in general.

Restriction to $\mathbb{S}$. It is important to notice that we may restrict the α-rectifying property to unit norm vectors since the norms directly scale the upper bias values α_i and can be re-introduced at any time. In this sense, X is α-rectifying if and only if $\overline{X} = \left(x_i \cdot \|x_i\|^{-1}\right)_{i \in I}$ is $\overline{\alpha}$-rectifying, where $\overline{\alpha}_i = \alpha_i \cdot \|x_i\|$. Therefore, in the following we will always assume $X \subseteq \mathbb{S}$, i.e. $\|x_i\| = 1$ for all $i \in I$. Note that this corresponds to standard weight normalization (Salimans & Kingma, 2016).

Bias-radius interplay. Often when studying ReLU-layers theoretically, the bias is implicitly incorporated into the linear part of the operator. However, in our work, we deliberately keep it as a shift as the interplay of bias and input domain is of central interest. We mentioned that an upper bias α favors injectivity when it is large. On the other hand, a large input data domain, i.e. a ball with large radius r offers more flexibility for normalization. However, there is a general trade-off: the larger the radius is chosen, the smaller α will get, in general, and vice versa. We have the following trivial fact:

Any frame is α-rectifying on $\mathbb{B}_r$ for $\alpha \equiv -r$,

i.e. $\alpha_i = -r$ for all $i \in I$. Hence, any redundant ReLU-layer is injective on the closed ball with any radius if the bias vector is sufficiently small. For a basis, (i.e. $m = n$) the above fact becomes also necessary, immediately implying that a basis can never be α-rectifying on $\mathbb{R}^n$ for any α. However, the standard basis for $\mathbb{R}^n$ is α-rectifying on $\mathbb{B}^+$ for $\alpha \equiv 0$. This shows that taking into account the input domain is a crucial step to take when studying injectivity since it naturally adapts to situations where a frame is not α-rectifying on $\mathbb{R}^n$ but might be on $\mathbb{B}_r$, resp. $\mathbb{B}_r^+$. The question that we are now interested in is, *how* to find a "good" upper bias for $\mathbb{B}_r$ and $\mathbb{B}_r^+$?

The Mercedes-Benz frame in $\mathbb{R}^2$ (Casazza & Kutyniok, 2012), given by

$$X_{mb} = \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -\sqrt{3}/2 \\ -1/2 \end{pmatrix}, \begin{pmatrix} \sqrt{3}/2 \\ -1/2 \end{pmatrix} \right)$$

(see Figure 1) is a particularly good example, where the optimal upper bias for $\mathbb{B}$ can be found by looking at the geometry of the frame. Its elements determine the vertices of an equilateral triangle so that we can reduce the problem to one pair of elements by symmetry. The worst case is found by $\langle x_i, x_j \rangle = -\frac{1}{2}$. Hence, X_{mb} is α-rectifying on $\mathbb{B}$ for $\alpha \equiv -\frac{1}{2}$. This idea can be generalized to polytopes in arbitrary dimensions. In $\mathbb{R}^3$, we obtain that the Tetrahedron frame, given by

$$X_{tet} = \frac{1}{\sqrt{3}} \cdot \left(\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} \right).$$

(see Figure 1) is α-rectifying on $\mathbb{B}$ for $\alpha \equiv -\frac{1}{\sqrt{3}}$. In a more general setting, where the frame elements are not aligned in a regular manner, we can at least reduce the problem to consider every face individually.

4. Convex Polytopes and Bias Estimations

In a nutshell, we estimate a "good" upper bias vector α for a given set of vectors X, hence, the ReLU-layer mapping C_α is injective on $\mathbb{B}_r$. It turns out that the combinatorial structure of the convex polytope associated with the elements of X can be related to the α-rectifying property of X. To prepare the estimation procedure, we shall introduce all building blocks for the estimation procedure for $\mathbb{B}_r$ in Section 4.1 and then deduce a version for $\mathbb{B}_r^+$ in Section 4.2.

For all standard results on convex polytopes, we refer to (Ziegler, 2012). Here, we are specifically interested in convex polytopes that arise as the set of all convex linear combinations of a collection of vectors $X = (x_i)_{i \in I} \subseteq \mathbb{S}$,

$$P_X = \{x \in \mathbb{R}^n : x = \sum_{i \in I} c_i \cdot x_i, c_i \geq 0, \sum_{i \in I} c_i = 1\}. \quad (7)$$

A face of P_X is any intersection of P_X with an affine half-space (in any dimension) such that none of the interior points of P_X (w.r.t. the induced topology on P_X) lie on its boundary. While vertices and edges are the 0- and 1-dimensional faces of P_X, the $(n-1)$-dimensional faces are called *facets*. For every face and, in particular, every facet F, there are $a \in \mathbb{R}^n \setminus \{0\}$ and $b \in \mathbb{R}$ such that

$$F = \{x \in P_X : \langle a, x \rangle = b\}, \quad (8)$$

i.e. any facet lies on an affine subspace of codimension 1 of $\mathbb{R}^n$. Furthermore, any $x \in F$ can be written as the convex linear combination,

$$x = \sum_{i \in I_F} c_i \cdot x_i, \quad c_i \geq 0, \quad \sum_{i \in I_F} c_i = 1.$$

We shall write the index set of vertices, associated with F as

$$I_F = \{i \in I : x_i \in F\}. \quad (9)$$

The following lemma reveals the core idea of our approach.

Lemma 4.1. *Let F be a facet. If $0 \notin F$, then X_{I_F} is a frame.*

In other words, as long as the facet does not go through the origin, the associated vertices form a frame. A proof can be found in the appendix.

We call X *omnidirectional* if 0 lies in the interior of P_X (w.r.t. the topology in $\mathbb{R}^n$), see Definition 1 in (Behrmann et al., 2018). Equivalently, there cannot be a hyperplane so that the elements in X are all accumulated on only one side of it.

For the proposed bias estimation on $\mathbb{B}_r$, omnidirectionality is an essential property as it allows to cover every $x \in \mathbb{B}_r$ the same way. For $\mathbb{B}_r^+$ we formulate an analogous condition in Section 4.2. Moreover, if X is omnidirectional, then 0 cannot lie on any facet of P_X and Lemma 4.1 applies. Numerically, it is verified via a simple convex optimization program (Behrmann et al., 2018).

Assuming a certain ordering of the facets, we write F_j referring to the j-th facets of P_X. Analogous to the idea of obtaining the optimal upper biases for the Mercedes-Benz and the Tetrahedron frame, we will use the frames $X_{I_{F_j}}$ for all j to estimate a bias. Letting the cone of F_j be denoted as

$$\text{cone}(F_j) = \{tx : x \in F_j, t \geq 0\},$$

then omnidirectionality and $X \subseteq \mathbb{S}$ provide the following properties.

Lemma 4.2. *If $X \subseteq \mathbb{S}$ is omnidirectional, then the following holds.*

(i) $\bigcup_j I_{F_j} = I$,

(ii) $\bigcup_j \text{cone}(F_j) = \mathbb{R}^n$

(iii) $X_{I_{F_j}}$ *is a frame for every j.*

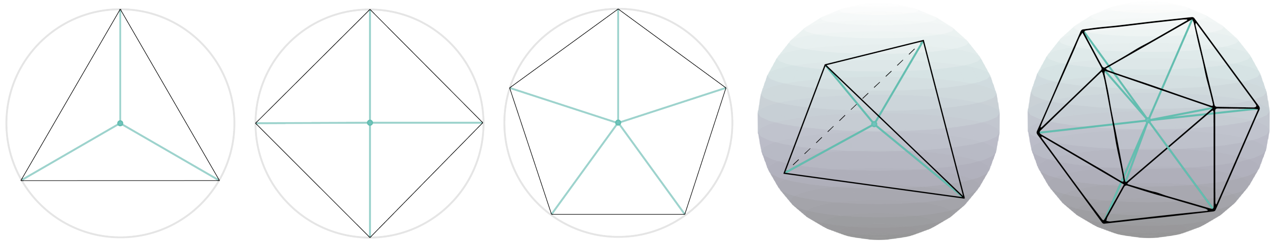

Figure 1. Frame vectors X (blue) and their convex hulls forming convex regular polytopes P_X. From left to right: Mercedes-Benz, Square, and Pentagon frame in $\mathbb{R}^2$, Tetrahedron and Icosahedron frame in $\mathbb{R}^3$. The unit ball $\mathbb{B}$ is outlined in gray.

These three properties build the backbone of our approach. By (i), every frame element is a vertex of P_X. Due to (ii), we can partition $\mathbb{B}_r$ into facet-specific conical subsets where we can estimate a bias locally. And most importantly, by (iii), every sub-collection associated to a facet induces a frame. Properties (i) and (ii) are easy to see and (iii) is a direct consequence of Lemma 4.1.

Remark 4.3. For a facet F, the vectors X_{I_F} will be redundant ($m > n$) only in rare cases. If the frame elements lie in general position on $\mathbb{S}$, then every X_{I_F} is a basis ($m = n$) with probability 1 (Buchta & Müller, 1984).

Before we introduce the upper bias estimation procedures for $\mathbb{B}_r$ and $\mathbb{B}_r^+$, we provide an explanation of why the particular grouping of the frame elements into vertices of facets is indeed suitable for the purpose of finding large upper bias values for the α-rectifying property.

If X is omnidirectional and F a facet of P_X, then consistent with (8) there are $a \in \mathbb{R}^n \setminus \{0\}$ and $0 \neq b \in \mathbb{R}$ such that

$$\begin{aligned} \langle a, x_k \rangle &= b, && \text{for } k \in I_F, \\ \langle a, x_\ell \rangle &< b, && \text{for } \ell \notin I_F. \end{aligned}$$

In this sense, the construction of X_{I_F} is a natural way of selecting spanning sub-collections of X with the highest coherence possible, making this particularly useful for our purpose.

4.1. Polytope Bias Estimation for $\mathbb{B}_r$

We now introduce the *Polytope Bias Estimation* (PBE) for $\mathbb{B}_r$ with $r > 0$. The procedure estimates an upper bias, denoted as $\alpha^{\mathbb{B}}$, such that X is $\alpha^{\mathbb{B}}$-rectifying on $\mathbb{B}$. This implies that X is $(r^{-1} \cdot \alpha^{\mathbb{B}})$-rectifying on $\mathbb{B}_r$.

The core idea is to partition $\mathbb{B}$ (and $\mathbb{S}$) into conical pieces,

$$F_j^{\mathbb{B}} := \operatorname{cone}(F_j) \cap \mathbb{B} \tag{10}$$

$$F_j^{\mathbb{S}} := \operatorname{cone}(F_j) \cap \mathbb{S}. \tag{11}$$

If X is omnidirectional, by Lemma 4.2, we have

$$\mathbb{B} = \bigcup_j F_j^{\mathbb{B}} \quad \text{and} \quad \mathbb{S} = \bigcup_j F_j^{\mathbb{S}}. \tag{12}$$

To find $\alpha_i^{\mathbb{B}}$, we identify the minimal analysis coefficient $\langle y, x_i \rangle$ that can occur for y on each $F_j^{\mathbb{B}}$ containing x_i, i.e.

$$\alpha_i^{\mathbb{B}} := \min_{\substack{y \in F_j^{\mathbb{B}} \\ j : x_i \in F_j}} \langle y, x_i \rangle. \tag{13}$$

We do not tackle this optimization problem directly but solve two related problems instead. On the one hand, we consider the minimal auto-correlation values on each facet,

$$\alpha_i^X := \min_{\substack{\ell \in I_{F_j} \\ j : x_i \in F_j}} \langle x_\ell, x_i \rangle, \tag{14}$$

that are easy to compute. On the other hand, we solve

$$\alpha_i^{\mathbb{S}} := \min_{\substack{y \in F_j^{\mathbb{S}} \\ j : x_i \in F_j}} \langle y, x_i \rangle \tag{15}$$

via convex linear programs. Note that the sets, on which all three optimization problems happen are subsets of each other, $F_j^{\mathbb{B}} \supset F_j^{\mathbb{S}} \supset X_{I_{F_j}}$, so that we immediately observe that $\alpha_i^{\mathbb{B}} \leq \alpha_i^{\mathbb{S}} \leq \alpha_i^X$. With this, we solve (13).

Theorem 4.4. *(PBE for $\mathbb{B}$) If $X \subseteq \mathbb{S}$ is omnidirectional, then X is $\alpha^{\mathbb{B}}$-rectifying on $\mathbb{B}$ and $\alpha_i^{\mathbb{B}}$, given in (13) can be computed as*

$$\alpha_i^{\mathbb{B}} = \begin{cases} 0 & \text{if } \alpha_i^X \geq 0 \\ \alpha_i^{\mathbb{S}} & \text{otherwise.} \end{cases} \tag{16}$$

If $\alpha_i^X < 0$, then $\alpha_i^{\mathbb{S}}$ given in (15) is the minimum over $j : x_i \in F_j$ of the solutions of the convex linear programs

$$\begin{aligned} \min \; & \left(x_i^\top D_{I_{F_j}} \right) d \\ & \text{subject to } d \geq 0 \\ & \|D_{I_{F_j}} d\|_2 \leq 1, \end{aligned} \tag{17}$$

where $D_{I_{F_j}}$ is the synthesis operator of $X_{I_{F_j}}$.

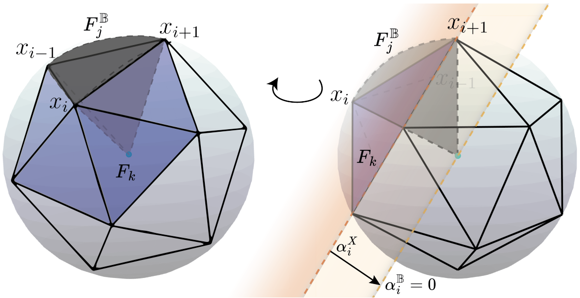

Figure 2. Geometrical intuition of the PBE for the Icosahedron frame on $\mathbb{B}$. Consider x_i. Left: The (blue) filled facets are used to compute $\alpha_i^{\mathbb{B}}$. The (gray) darker piece (dashed border) indicates $F_j^{\mathbb{B}}$. Right: Rotated perspective of the left image. The affine half-space $\Omega_i = \{x \in \mathbb{R}^n : \langle x, x_i \rangle \geq \alpha_i^X, i \in I_{F_j}\}$, indicated by the left-most area with decreasing opacity (orange) contains all vectors such that all vertices of the adjacent facets are active. Since $\alpha_i^X \geq 0$, the brighter (yellow) half-space represents the solution $\alpha_i^{\mathbb{B}} = 0$.

A proof can be found in the appendix. The general case follows from $\mathbb{B}_r = \{x \in \mathbb{R}^n : x = r \cdot y, y \in \mathbb{B}\}$ for $r > 0$.

Hence, the minimal argument of (13) lies on $\mathbb{S}$ or at zero, depending on the sign of the minimal correlation of a facet, given by α_i^X. So, a strategy to obtain $\alpha^{\mathbb{B}}$ is to start considering the easy-to-compute α_i^X by finding the smallest auto-correlation value with x_i among all facets that are adjacent to x_i. Then, only if $\alpha_i^X < 0$, the convex optimization (17) has to be solved. See Algorithm 1 for a pseudo-code of the procedure.

Example 1. *(a) For the Tetrahedron frame X_{tet}, we have $\alpha^X \equiv -\frac{1}{3}$, therefore $\alpha^{\mathbb{B}} = \alpha^{\mathbb{S}} \equiv -\frac{1}{\sqrt{3}}$.*
(b) For the Icosahedron frame, given by

$$X_{ico} = \frac{1}{\sqrt{1+\varphi^2}} \cdot \left(\begin{pmatrix} 0 \\ \pm 1 \\ \pm\varphi \end{pmatrix}, \begin{pmatrix} \pm 1 \\ \pm\varphi \\ 0 \end{pmatrix}, \begin{pmatrix} \pm\varphi \\ 0 \\ \pm 1 \end{pmatrix} \right),$$

(see Figure 1), where $\varphi = \frac{1+\sqrt{5}}{2}$ is the golden ratio, we have $\alpha^X \equiv \frac{\varphi}{1+\varphi^2} \approx 0.45$, therefore $\alpha^{\mathbb{B}} \equiv 0$. Figure 2 shows the idea of the PBE for this example geometrically.

Note that $0 \geq \alpha_i^{\mathbb{B}}$ is reasonable to guarantee the α-rectifying property on $\mathbb{B}_r$ since for any upper bias α and $x = 0$, it has to hold that $\langle 0, x_i \rangle = 0 \geq \alpha_i$ for all i in some I_{F_j}.

4.2. Polytope Bias Estimation for $\mathbb{B}_r^+$

In neural networks, often ReLU-layers succeed each other. In this context, we show that $\mathbb{B}_r^+$ is conceptually the right input domain for a PBE for ReLU-layers that are applied to the output of a previous one. In fact, this requires knowing where the image of $\mathbb{B}_r$ under C_α lies.

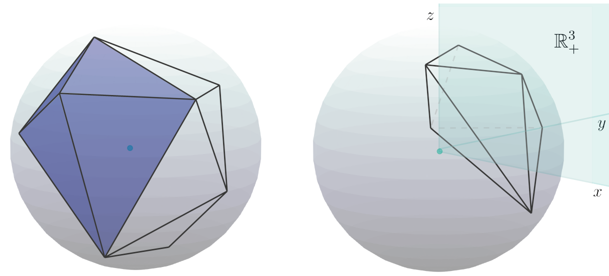

Figure 3. Non-regular polytopes. Left: The estimated bias values $\alpha_i^{\mathbb{B}}$ are computed from the largest adjacent facet of x_i. Hence, the less regular the normalized frame elements are distributed on the sphere, the smaller $\alpha^{\mathbb{B}}$ becomes. Right: The frame is non-negatively omnidirectional since $\bigcup_{j \in J^+} F_j \supseteq \mathbb{R}_+^n$, but not omnidirectional.

Lemma 4.5. *Let X be α-rectifying and B_0 denote the largest optimal upper frame bound among $X_{I_x^\alpha}$ with $x \in \mathbb{B}_r$. Then*

$$C_\alpha(\mathbb{B}_r) \subseteq \mathbb{B}^+_{r\sqrt{B_0}}. \tag{18}$$

It is easy to show that (18) is a direct consequence of the upper inequality in (6) and clearly, holds for $x \in \mathbb{B}_r^+$ as well. Note that we may also estimate the radius of the ball as $r\sqrt{B}$, where B is any upper frame bound of X.

We approach the PBE for $\mathbb{B}_r^+$ by restricting the computations of the PBE introduced in Theorem 4.4 to only those facets, that actively contribute to the estimation. In this sense, we only consider those frame elements whose associated facets have a non-trivial intersection with $\mathbb{R}_+^n$. We denote the corresponding index sets as

$$J^+ = \{j : F_j \cap \mathbb{R}_+^n \neq \emptyset\}, \qquad I^+ = \bigcup_{j \in J^+} I_{F_j}.$$

According to this, instead of omnidirectionality, we only have to require

$$\mathbb{R}_+^n \subseteq \bigcup_{j \in J^+} \operatorname{cone}(F_j), \quad \text{and} \tag{19}$$

$$0 \notin F_j \;\text{ for all } j \in J^+, \tag{20}$$

which we shall refer to as *non-negative omnidirectionality*. See Figure 3 (right) for an illustration. This is tailored to provide the properties in Lemma 4.2 for $\mathbb{B}_r^+$: by (19), we have analogously to (12),

$$\mathbb{B}^+ \subseteq \bigcup_{j \in J^+} F_j^{\mathbb{B}} \tag{21}$$

and condition (20) is sufficient for $X_{I_{F_j}}$ being a frame for every $j \in J^+$ by Lemma 4.1. With this, we have all requirements to deduce the PBE for $\mathbb{B}_r^+$.

Algorithm 1 PBE for $\mathbb{B}_r$

Get I_{F_j} via computing V_X
for $j = 1, \dots, J$ **do**
  $\beta_j = \min_{k<\ell\in I_{F_j}} \langle x_k, x_\ell\rangle$
end for
for $i = 1, \dots, m$ **do**
  $\alpha_i^* = \min\limits_{j \text{ s.t. } i\in I_{F_j}} \beta_j$
  if $\alpha_i^* \geq 0$ **then**
    $\alpha_i^{\mathbb{B}} \leftarrow 0$
  else
    $y^* \leftarrow$ solution of (17)
    $\alpha_i^{\mathbb{B}} \leftarrow y^* \cdot r^{-1}$
  end if
end for

Theorem 4.6 (PBE for $\mathbb{B}^+$)**.** *If $X \subseteq \mathbb{S}$ is non-negatively omnidirectional, then X is $\alpha^{\mathbb{B}^+}$-rectifying on $\mathbb{B}^+$ with*

$$\alpha_i^{\mathbb{B}^+} = \begin{cases} \alpha_i^{\mathbb{B}} & \textit{for } i \in I^+ \\ s_i & \textit{else,} \end{cases} \tag{22}$$

where $s_i \in \mathbb{R}$ is arbitrary.

A proof can be founded in the appendix. This reduces the computational cost and improves the upper bias estimation as potentially large bias values in $I \setminus I^+$ can be omitted.

Remark 4.7. Clearly, conditions (19) and (20) are weaker than omnidirectionality, yet, harder to check numerically. Similarly, $F_j \cap \mathbb{R}^n_+ \neq \emptyset$ is not straightforward to verify. Indeed, it holds true for all adjacent facets of $x_i \in \mathbb{R}^n_+$, however, there might be facets meeting the condition but with no vertices in $\mathbb{R}^n_+$. The interested reader will find a continued discussion in the appendix. Finding an efficient implementation of this, however, is left as an open problem.

4.3. Remarks on the Optimality of the PBE

In general, we cannot expect the proposed PBE to yield upper biases that are maximal. Estimating the error of the estimation to a maximal upper bias (if exists), however, is difficult since this would require knowing the combinatorial structure (i.e. the vertex-facet index sets) of a general polytope, which has been a topic of active research for several decades. In the special cases when the polytopes are regular and simplicial (every facet has exactly n vertices), e.g. Mercedes-Benz, Tetrahedron and Icosahedron frame, we expect that the estimated upper bias is indeed maximal. It is easy to verify that the PBE is also stable to perturbations as long as the combinatorial structure is preserved. Hence, one could expect that the estimation will be more accurate the more evenly distributed the frame elements are on the sphere. See Figure 3 (left) for an illustration.

Algorithm 2 Reconstruction via Facets

Get I_{F_j} via computing V_X
for $j = 1, \dots, J$ **do**
  $S^{-1}_{I_{F_j}} \leftarrow \left((C_{I_{F_j}})^\top C_{I_{F_j}}\right)^{-1}$
  $\overline{X} \leftarrow X_{I_{F_j}}$
  $\tilde{D}_{I_{F_j}} \leftarrow \begin{pmatrix} | & | & & | \\ S^{-1}_{I_{F_j}}\overline{x}_1 & S^{-1}_{I_{F_j}}\overline{x}_2 & \cdots & S^{-1}_{I_{F_j}}\overline{x}_{|I_{F_j}|} \\ | & | & & | \end{pmatrix}$
end for
$z = C_\alpha x$
$\overline{z} \leftarrow z + \alpha$
while $j = 1, \dots, J$ **do**
  if $I_{F_j} \in I_x^\alpha$ **then**
    $\tilde{D}_{I_{F_j}} \overline{z}_{|I_{F_j}} = x$
  end if
end while

4.4. Local Reconstruction via Facets

Unless $I_x^\alpha \neq I$ for all $x \in \mathbb{B}_r$, there cannot be only *one* global left-inverse for C_α. We propose to systematically construct a collection of left-inverses, each associated with one facet of P_X. Recall that the frame operator of $X_{I_{F_j}}$ is denoted by $S_{I_{F_j}}$ and that its canonical dual frame is given by $\tilde{X}_{I_{F_j}} = \left(S^{-1}_{I_{F_j}} x_i\right)_{i\in I_{F_j}}$.

Theorem 4.8. *Let $X \in \mathbb{S}$ be α-rectifying on $\mathbb{B}$ and omnidirectional. For every $x \in \mathbb{B}$ there is j such that*

$$\tilde{D}_{I_{F_j}} C_\alpha x = x, \tag{23}$$

where

$$\begin{aligned} \tilde{D}_{I_{F_j}} &: \mathbb{R}^m \to \mathbb{R}^n \\ (c_i)_{i\in I} &\mapsto \sum_{i\in I_{F_j}} (c_i + \alpha_i) \cdot S^{-1}_{I_{F_j}} x_i. \end{aligned} \tag{24}$$

In other words, $\tilde{D}_{I_{F_j}}$ is a left-inverse of C_α for all $x \in F_j^{\mathbb{B}}$. By (12), every $x \in \mathbb{B}$ lies in some $F_j^{\mathbb{B}}$, hence indeed for any $x \in \mathbb{B}$ there is a left-inverse. It is easy to see that (23) reduces to the usual canonical frame decomposition (2) of x by $X_{I_{F_j}}$.

Remark 4.9. In general, there are infinitely many duals for X (Christensen, 2003). The canonical dual mentioned above relates to the pseudo-inverse of the associated analysis operator, hence induces the optimal inverse by means of ridge regression.

4.5. Implementation

We discuss the implementation aspects of the PBE for $\mathbb{B}$ and the reconstruction formulas. Our implementations of the algorithms are publicly available under https://github.com/danedane-haider/Alpha-rectifying-frames.

4.5.1. PBE

The vertex-facet index sets I_{F_j} are encoded in what is called the *vertex-facet incidence matrix* V_X. Assuming that P_X has J facets, then V_X is the $J \times m$ matrix with entries

$$V_X[j,i] = \begin{cases} 1 & \text{if } i \in I_{F_j} \\ 0 & \text{else,} \end{cases}$$

indicating which vertices correspond to which facets. To compute the vertex-facets incidences, we use the routine `VERTICES_IN_FACETS` from the open-source software Polymake (Gawrilow & Joswig, 2000). This routine requires the vertices in homogeneous coordinates, i.e.

$$C_{hom} = \begin{pmatrix} 1 - x_1 - \\ \vdots \\ 1 - x_m - \end{pmatrix}.$$

Already noted in (Puthawala et al., 2022), checking the α-rectifying property is probably NP-hard. The computation of V_X relies on convex-hull algorithms, which are also not expected to run in polynomial time for general polytopes. However, for points in general position on $\mathbb{S}$ (i.e. no hyperplane in $\mathbb{R}^n$ contains more than n of the points), the "reverse-search" algorithm is expected to finish in linear time in the number of vertices m for fixed dimension n (Assarf et al., 2016). This condition is precisely fulfilled when assuming random initialization and normalized frame elements. Polymake uses this algorithm via the command `prefer "lrs";`.

Algorithm 1 gives step-by-step instructions to compute $\alpha^{\mathbb{B}}$ for any omnidirectional frame $X \subseteq \mathbb{S}$.

4.5.2. Reconstruction

In practice, one can read off I_x^α from $z = C_\alpha x$ and find a facet F_j such that $I_{F_j} \subseteq I_x^\alpha$ using the vertex-facet incidences. Note that F_j might not be unique with this property.

Algorithm 2 describes how the systematic construction of the left-inverses $\tilde{D}_{I_{F_j}}$ can be done, assuming that X is α-rectifying and omnidirectional.

5. Numerical Experiments

A series of experiments revealed that the injectivity behavior of a ReLU-layer is very sensitive to many hyperparameters and circumstances, such as the size of the layer, the depth of the network, the position of the layer within the network, initialization and normalization procedures, the optimizer

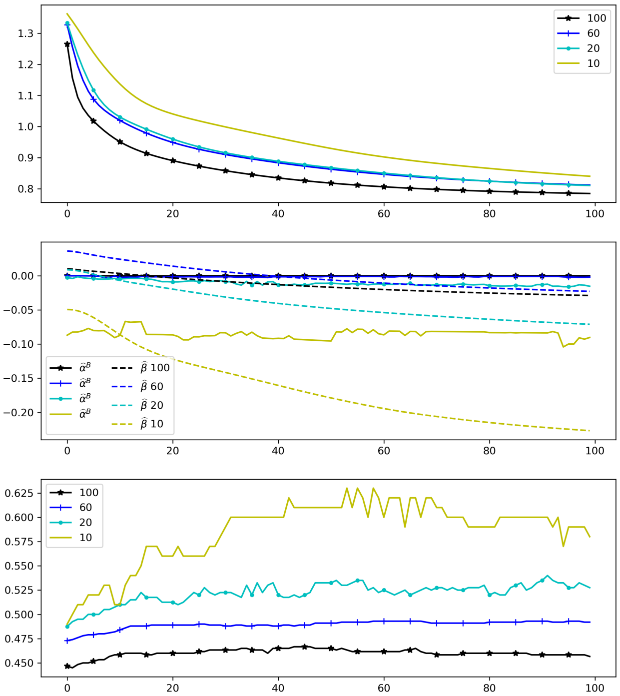

Figure 4. Averaged quantities (across 10 iterations) related to a ReLU-layer over 100 epochs of training with different redundancies $m = |I|$. Top: Cross entropy loss on the validation set. Mid: Mean of the trained biases, $\widehat{\beta}$ (dashed), and mean of the estimated upper biases on $\mathbb{B}_r$, $\widehat{\alpha}^{\mathbb{B}}$ (solid). Bottom: Proportion of learned bias values that are smaller than the estimations, i.e. $\#(\beta_i \leq \alpha_i^{\mathbb{B}})/m$, indicating the injectivity trend.

and the data itself. Here, we present a numerical experiment, where we want to focus merely on the size of the ReLU-layer, i.e. its redundancy. Therefore, the experimental setting is designed to be as simple and reduced as possible. Considering more realistic network models requires a much broader study, which goes beyond the scope of this contribution.

5.1. Experimental Setting

We train a neural network with one ReLU-layer and a soft-max output layer on the Iris data set (Fisher, 1936). For the ReLU-layer, we consider four redundancy settings $m = |I| = 10, 20, 60, 100$. The corresponding networks are mappings from $\mathbb{R}^4 \to \mathbb{R}^m \to \mathbb{R}^3$. After normalization to zero mean and a variance of one, all data samples lie within the ball of radius $r = 3.1$. We use the upper biases from the PBE for $\mathbb{B}$ (Theorem 4.4) with appropriate scaling to monitor the injectivity behavior of the ReLU-layers during training, see Figure 4. Here, optimization of a cross-entropy loss is done using stochastic gradient descent at a learning rate of 0.5 for 100 epochs.

5.2. Discussion

The top plot shows that in our setting, high redundancy in the ReLU-layer yields the smallest validation loss. Expectedly, high redundancy also increases the chance of the polytope P_X having many non-negatively correlated facets, i.e. $\alpha_i^X \geq 0$. Hence, more bias estimations are 0 according to (16). The mid-plot shows this nicely (solid lines). Note that all learned biases β decrease in mean (dashed lines). The lower the redundancy, the stronger this decrease is. Since the bias estimations $\alpha_i^{\mathbb{B}}$ remain almost unchanged in mean, we may conclude that lacking injectivity of low redundancy ReLU-layers (i.e. too many output values are zero) is compensated during training via the bias. The bottom plot shows another measure of the injectivity trend: the proportion of learned bias values smaller than the estimations, i.e. $\#(\beta_i \leq \alpha_i^{\mathbb{B}})/m$. An increase in this quantity indicates that a ReLU-layer is becoming "more injective" during training. In concordance with the previous observations, layers with low redundancy show a stronger increase here as well. This could be interpreted as high redundancy favors injectivity from the start.

In this sense, the PBE can help us to better understand the role of injectivity in neural networks. In the example, we are able to see the effect of different sizes of a ReLU-layer in regard to injectivity and validation loss and, in particular, what happens when the layer is chosen too small. It remains an open question if these results are representative for other settings and which are the responsible components causing this behavior. Future numerical investigation is necessary for a better understanding.

6. Conclusion

We presented a frame-theoretic setting to study the injectivity of a ReLU-layer on the closed ball with radius $r > 0$ in $\mathbb{R}^n$ and on its non-negative part. Moreover, we introduced a systematic approach of verifying it in practice, called polytope bias estimation (PBE). This method exploits the convex geometry of the weight matrix associated with a ReLU-layer and estimates a bias vector such that the layer is injective on the ball for all biases smaller or equal to the estimation. This allows us to give sufficient and quantified conditions for the invertibility of a ReLU-layer. Corresponding reconstruction formulas are provided. Via a straightforward implementation, the PBE allows to study the injectivity behavior of a redundant ReLU-layer and perform perfect reconstruction of the layer input where applicable. So far, our work contributes to a better understanding of the behavior of neural network layers by means of observations without interaction with the actual optimization procedure. As a possible application, the estimated upper biases from the PBE could be used to design a regularization procedure where the bias is guided toward injectivity during training.

Acknowledgment

D. Haider is recipient of a DOC Fellowship of the Austrian Academy of Sciences at the Acoustics Research Institute (A 26355). The work of M. Ehler was supported by the WWTF project CHARMED (VRG12-009) and P. Balazs was supported by the OeAW Innovation grant project FUn (IF_2019_24_Fun), and the FWF projects LoFT (P 34624) and NoMASP (P 34922). The authors would like to thank Daniel Freeman for fruitful and fun discussions and Lukas Köhldorfer for his valuable feedback.

References

Assarf, B., Gawrilow, E., Herr, K., Joswig, M., Lorenz, B., Paffenholz, A., and Rehn, T. Computing convex hulls and counting integer points with Polymake. *Mathematical Programming Computation*, 9, 2016.

Balazs, P. Frames and finite dimensionality: Frame transformation, classification and algorithms. *Applied Mathematical Sciences*, 2(43):2131–2144, 2008.

Behrmann, J., Dittmer, S., Fernsel, P., and Maaß, P. Analysis of invariance and robustness via invertibility of ReLU-networks. *arXiv*, abs/1806.09730, 2018.

Bruna, J., Szlam, A., and Lecun, Y. Signal recovery from pooling representations. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1585–1598, 2014.

Buchta, C. and Müller, J. Random polytopes in a ball. *Journal of Applied Probability*, 21(4):753–762, 1984.

Casazza, P. G. and Kutyniok, G. *Finite frames: Theory and applications*. Springer, 2012.

Christensen, O. *An Introduction to Frames and Riesz Bases*. Birkhäuser, 2003.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017.

Dittmer, S., King, E. J., and Maass, P. Singular values for ReLU layers. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3594–3605, 2020.

Donahue, J., Krähenbühl, P., and Darrell, T. Adversarial feature learning. In *International Conference on Learning Representations*, 2017.

Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.

Gawrilow, E. and Joswig, M. Polymake: A framework for analyzing convex polytopes. In Kalai, G. and Ziegler, G. M. (eds.), *Polytopes — Combinatorics and Computation*, pp. 43–73. Birkhäuser Basel, 2000.

Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In Gordon, G., Dunson, D., and Dudík, M. (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 315–323, 2011.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Huang, L., Qin, J., Zhou, Y., Zhu, F., Liu, L., and Shao, L. Normalization techniques in training DNNs: Methodology, analysis and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (01):1–20, 2023.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, 2012.

LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. In Montavon, G., Orr, G. B., and Müller, K.-R. (eds.), *Neural Networks: Tricks of the Trade: Second Edition*, pp. 9–48. Springer, 2012.

Mehta, M. L. *Random Matrices*. Elsevier, 2004.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 807–814, 2010.

Puthawala, M., Kothari, K., Lassas, M., Dokmanić, I., and de Hoop, M. Globally injective ReLU networks. *Journal of Machine Learning Research*, 23(105):1–55, 2022.

Salimans, T. and Kingma, D. P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 901–909, 2016.

Ziegler, G. M. *Lectures on polytopes*. Springer, 2012.

A. Proofs

A.1. Theorem 3.2

Proof. Assume $C_\alpha x = C_\alpha y$, for $x, y \in \mathbb{B}_r$. Clearly,

$$\langle x, x_i \rangle > \alpha_i \quad \Leftrightarrow \quad \langle y, x_i \rangle > \alpha_i, \tag{25}$$

so that we derive

$$\langle x, x_i \rangle = \langle y, x_i \rangle, \qquad \text{for all } i \in I_x^\alpha \cap I_y^\alpha. \tag{26}$$

Since $\mathbb{B}_r$ is convex, for $\lambda \in (0,1)$ it holds that $x_\lambda := (1-\lambda)x + \lambda y \in \mathbb{B}_r$. For $i \in I_{x_\lambda}^\alpha$, we compute

$$\langle x_\lambda, x_i \rangle = (1-\lambda)\langle x, x_i \rangle + \lambda \langle y, x_i \rangle.$$

Since $\langle x_\lambda, x_i \rangle \geq \alpha_i$, at least one of the two, $\langle x, x_i \rangle$ or $\langle y, x_i \rangle$, must be bigger or equals α_i. Without loss of generality, let us suppose that $\langle x, x_i \rangle \geq \alpha_i$. If we have $\langle x, x_i \rangle > \alpha_i$, then (25) leads to $\langle y, x_i \rangle > \alpha_i$. If $\langle x, x_i \rangle = \alpha_i$, then $\langle x_\lambda, x_i \rangle \geq \alpha_i$ implies $\langle y, x_i \rangle = \alpha_i$. Thus, we have verified that $I_{x_\lambda}^\alpha \subseteq I_x^\alpha \cap I_y^\alpha$. Since $x_\lambda \in \mathbb{B}_r$, the α-rectifying property yields that $X_{I_{x_\lambda}^\alpha}$ is a frame. Hence, $X_{I_x^\alpha \cap I_y^\alpha}$ is a frame. According to (26), we deduce $x = y$. Therefore, C_α is injective on $\mathbb{B}_r$.

Since $\mathbb{B}_r^+$ is a convex set as well, the statement for $\mathbb{B}_r^+$ can be proven analogously. □

A.2. Lemma 4.1

We recall that a collection of points $p_1, \ldots, p_m$ in $\mathbb{R}^n$ are called *affinely independent* if $\sum_{i=1}^m \lambda_i p_i = 0$ and $\sum_{i=1}^m \lambda_i = 0$ imply that $\lambda_1 = \ldots = \lambda_m = 0$.

Proof. It is a known fact that any facet of a polytope in $\mathbb{R}^n$ contains at least n affinely independent vertices (Ziegler, 2012). Let F be a facet and w.l.o.g. $x_1, \ldots, x_n \in X_{I_F}$ be affinely independent. If $0 \notin F$, by (8) there is $a \in \mathbb{R}^n \setminus \{0\}$ and $b \in \mathbb{R}$ with $b \neq 0$ such that

$$\langle a, x_i \rangle = b, \qquad i = 1, \ldots, n. \tag{27}$$

Assuming $\sum_{i=1}^n \lambda_i x_i = 0$ and using (27), from

$$0 = \langle a, \sum_{i=1}^n \lambda_i x_i \rangle = \sum_{i=1}^n \lambda_i \langle a, x_i \rangle = \sum_{i=1}^n \lambda_i b, \tag{28}$$

it follows that $\sum_{i=1}^n \lambda_i = 0$. Since $x_1, \ldots, x_n$ are affinely independent, $\lambda_1 = \ldots = \lambda_n = 0$. Therefore, $x_1, \ldots, x_n$ are linearly independent, hence X_{I_F} is a frame for $\mathbb{R}^n$. □

A.3. Theorem 4.4

Proof. We first show that if $\alpha_i^X \geq 0$, then $\alpha_i^{\mathbb{S}} = \alpha_i^X$.

By omnidirectionality, $\bigcup_j F_j^{\mathbb{S}} = \mathbb{S}$. Hence, for every $y \in \mathbb{S}$ there is x_y in some facet F_j such that $y = \frac{x_y}{\|x_y\|} \in F_j^{\mathbb{S}}$. We use that x_y can be written as a convex combination of elements of $X_{I_{F_j}}$, i.e. $x_y = \sum_{\ell \in I_{F_j}} c_\ell x_\ell$ with $c_\ell \geq 0$ for all $\ell \in I_{F_j}$ and $\sum_{\ell \in I_{F_j}} c_\ell = 1$. Hence,

$$y = \frac{x_y}{\|x_y\|} = \sum_{\ell \in I_{F_j}} \frac{c_\ell}{\|x_y\|} x_\ell = \sum_{\ell \in I_{F_j}} d_\ell x_\ell. \tag{29}$$

Let $\alpha_i^X \geq 0$, i.e. $\langle x_\ell, x_i \rangle \geq 0$ for all $\ell \in I_{F_j}$ and $j : x_i \in F_j$. Using (29) with $d_\ell = \frac{c_\ell}{\|x_y\|} \geq 0$ and $\sum_{\ell \in I_{F_j}} d_\ell \geq 1$, we obtain $\langle y, x_i \rangle \geq \min_{\ell \in I_{F_j}} \langle x_\ell, x_i \rangle$. Thus, we derive

$$\alpha_i^{\mathbb{S}} = \min_{\substack{y \in F_j^{\mathbb{S}} \\ j : x_i \in F_j}} \langle y, x_i \rangle \geq \min_{\substack{\ell \in I_{F_j} \\ j : x_i \in F_j}} \langle x_\ell, x_i \rangle = \alpha_i^X. \tag{30}$$

Recalling that $\alpha_i^X \geq \alpha_i^{\mathbb{S}}$ for all $i \in I$ yields the claim.

Next we show (16), using that $F_j^{\mathbb{B}} = \{y \in \mathbb{B} : y = \frac{x}{t}, \exists x \in F_j^{\mathbb{S}}, \exists t \in [1, \infty)\}$ and $\langle x, x_i \rangle \geq \alpha_i^{\mathbb{S}}$ for all $x \in F_j^{\mathbb{S}}$ with $x_i \in F_j$.

Let $\alpha_i^X \geq 0$, then $\alpha_i^{\mathbb{S}} \geq 0$ by (30). Hence $\langle x, x_i \rangle \geq 0$ for all $x \in F_j^{\mathbb{S}}$. We deduce,

$$\alpha_i^{\mathbb{B}} = \min_{\substack{y \in F_j^{\mathbb{B}} \\ j : x_i \in F_j}} \langle y, x_i \rangle = \min_{\substack{x \in F_j^{\mathbb{S}}, t \geq 1 \\ j : x_i \in F_j}} \langle \frac{x}{t}, x_i \rangle = 0.$$

Now let $\alpha_i^X < 0$. For $x \in F_j^{\mathbb{S}}$ with $x_i \in F_j$, we distinguish two cases. If $\langle x, x_i \rangle \geq 0$, then $\langle \frac{x}{t}, x_i \rangle \geq 0$. If $\langle x, x_i \rangle < 0$, then $\langle \frac{x}{t}, x_i \rangle \geq \langle x, x_i \rangle \geq \alpha_i^{\mathbb{S}}$. Since $\alpha_i^{\mathbb{S}} \leq \alpha_i^X < 0$, we deduce $\alpha_i^{\mathbb{B}} \geq \alpha_i^{\mathbb{S}}$. Recalling that $\alpha_i^{\mathbb{B}} \leq \alpha_i^{\mathbb{S}}$ for all $i \in I$ we obtain $\alpha_i^{\mathbb{B}} = \alpha_i^{\mathbb{S}}$.

To see that X is $\alpha^{\mathbb{B}}$-rectifying on $\mathbb{B}$, take an arbitrary $z \in \mathbb{B}$. By (12), there is j such that $z \in F_j^{\mathbb{B}}$. Hence, $\langle z, x_i \rangle \geq \alpha_i^{\mathbb{B}}$ holds for any $i \in I_{F_j}$. In other words, $I_{F_j} \subseteq I_z^{\alpha^{\mathbb{B}}}$. Since $X_{I_{F_j}}$ is a frame by (iii) in Lemma 4.2, $X_{I_z^{\alpha^{\mathbb{B}}}}$ is also a frame, showing the claim.

Finally, recalling that $D_{I_{F_j}}$ is the synthesis operator of $X_{I_{F_j}}$ we may rewrite Equation (29) for $y \in F_j^{\mathbb{S}}$ as $y = \sum_{\ell \in I_{F_j}} d_\ell x_\ell = D_{I_{F_j}} d$. Then $\min_{y \in F_j^{\mathbb{S}}} \langle y, x_i \rangle$ for $i \in I_{F_j}$ can be formulated as the linear program

$$\begin{aligned} \min \ & \left(x_i^\top D_{I_{F_j}} \right) d \\ & \text{subject to } d \geq 0 \\ & \| D_{I_{F_j}} d \|_2 = 1. \end{aligned}$$

If $\alpha_i^X < 0$, then the above minimum is negative since $\alpha_i^{\mathbb{S}} \leq \alpha_i^X < 0$. Hence, we can replace $\|D_{I_{F_j}} d\|_2 = 1$ by $\|D_{I_{F_j}} d\|_2 \leq 1$ making the problem convex. $\square$

A.4. Theorem 4.6

Proof. Let $z \in \mathbb{B}^+$. By (21), there is $j \in J^+$ such that $z \in F_j^{\mathbb{B}}$. Since $F_j \cap \mathbb{R}_+^n \neq \emptyset$ we have in particular that $I_{F_j} \subseteq I^+$. Hence, analog to the proof of Theorem 4.4, $\langle z, x_i \rangle \geq \alpha_i^{\mathbb{B}}$ holds for any $i \in I_{F_j}$. With $\alpha^{\mathbb{B}^+}$ defined as in (22) we have that $I_{F_j} \subseteq I_z^{\alpha^{\mathbb{B}}} \subseteq I_z^{\alpha^{\mathbb{B}^+}}$. Since $X_{I_{F_j}}$ is a frame by Lemma 4.1, $X_{I_z^{\alpha^{\mathbb{B}^+}}}$ is also a frame. $\square$

B. Remarks

B.1. Remark 4.7

For fixed j, one may verify $F_j \cap \mathbb{R}_+^n \neq \emptyset$ via the feasibility of the convex optimization problem

$$\begin{aligned} \min \ & \| D_{I_{F_j}} c \|_2 \\ & \text{subject to } c \geq 0 \\ & \sum_i c_i = 1. \end{aligned} \tag{31}$$

Indeed, if (31) has a solution, then there is $c \in \mathbb{R}_+^n$ that can be written as a convex linear combination of the vertices of F_j, hence, $F_j \cap \mathbb{R}_+^n \neq \emptyset$. We suggest the following strategy: Label all facets with vertices in $\mathbb{R}_+^n$ and continue solving (31) for all adjacent facets. If there is no vertex in $\mathbb{R}_+^n$ at all, solve (31) for the facets that contain vertices x_k with only small negative components.