Abstract

This paper provides a theoretical justification of the superior classification performance of deep rectifier networks over shallow rectifier networks from the geometrical perspective of piecewise linear (PWL) classifier boundaries. We show that, for a given threshold on the approximation error, the required number of boundary facets to approximate a general smooth boundary grows exponentially with the dimension of the data, and thus the number of boundary facets, referred to as boundary resolution, of a PWL classifier is an important quality measure that can be used to estimate a lower bound on the classification errors. However, learning naively an exponentially large number of boundary facets requires the determination of an exponentially large number of parameters and also requires an exponentially large number of training patterns. To overcome this issue of “curse of dimensionality”, compressive representations of high resolution classifier boundaries are required. To show the superior compressive power of deep rectifier networks over shallow rectifier networks, we prove that the maximum boundary resolution of a single hidden layer rectifier network classifier grows exponentially with the number of units when this number is smaller than the dimension of the patterns. When the number of units is larger than the dimension of the patterns, the growth rate is reduced to a polynomial order. Consequently, the capacity of generating a high resolution boundary will increase if the same large number of units are arranged in multiple layers instead of a single hidden layer. Taking high dimensional spherical boundaries as examples, we show how deep rectifier networks can utilize geometric symmetries to approximate a boundary with the same accuracy but with a significantly fewer number of parameters than single hidden layer nets.

Keywords: Deep Learning, Rectifier Neural Network, Compressive Power, Classifier Boundaries

1. Introduction

Although as few as one hidden layer neural networks are capable of approximating any Borel measurable functions (Hornik et al., 1989), deeper neural networks have outperformed shallower neu-
ral networks in a wide range of applications such as handwritten digit recognition (Ciresan et al., 2012), object detection (Ren et al., 2015) and image classification (Krizhevsky et al., 2012; He et al., 2016a,b). The benefits of neural networks’ depth have been investigated extensively in recent years, from the superior power of deep networks in function approximation (Delalleau and Bengio, 2011; Eldan and Shamir, 2016; Cohen et al., 2015; Mhaskar et al., 2016), to the superior capacity of deep networks in separating the input space into a large number of regions of linearity (Pascanu et al., 2014; Montúfar et al., 2014; Raghu et al., 2016). From these theoretical analyses, one can conclude that, for some functions that can be represented or approximated by both deep neural networks and single hidden layer networks, the representation provided by deep networks can be much more compact (i.e., with fewer parameters) and thus generalises better. However, the functions considered so far have been limited to certain families of polynomial functions or hand-coded functions, which have been chosen to demonstrate the expressive power of deep neural networks. It is unlikely that practically trained neural networks will fall into one of these analysed categories of functions. A good understanding of how and why deep neural networks achieve their empirical successes is thus still missing (Raghu et al., 2016).

This paper aims to theoretically justify the superior classification performances of deep rectifier networks over shallow rectifier networks from the geometrical perspective of PWL classifier boundaries. Given a dataset comprising several classes, according to learning theory (Vapnik and Vapnik, 1998), a simpler learning model with a smaller sample complexity usually generalises better. However, approximation models of a simple geometric surface can be quite complex, and a large difference may exist between the complexity of the approximation model and that of the original model. For instance, (Dudley, 1974) shows that an exponentially large number of facets is required to approximate a spherical surface in $d$ dimensional space. This is despite the fact that a spherical surface is a simple geometric model which can be represented with $d + 1$ parameters (one for the radius, the others for the center). We will show that the required number of units is at least a polynomial function of the dimension of the patterns if single hidden layer rectifier networks are used to approximate a spherical boundary (Theorem 7, Sec. 5). We will present an upper bound on the number of facets with respect to the number of units of a single hidden layer rectifier network. This upper bound shows that the capacity of single hidden layer rectifier nets to generate facets increases exponentially when the number of units is smaller than the dimension of the input. However, the growth rate is reduced to a polynomial order when the number of units is larger than the dimension of the patterns (Lemma 6, Sec. 5). Consequently, the capacity of generating facets will increase if the same large number of units are arranged in multiple layers instead of a single hidden layer. With spherical surfaces as examples, we will show that deep rectifier nets can be exponentially more efficient than single hidden layer nets. The main contributions of this paper include:

i) **The introduction of boundary resolution for PWL classifiers** (Sec. 3). The resolution of PWL classifier boundaries is a measure of the classifier quality, which can be used to estimate a lower bound on the classification errors. The introduction of this new concept provides a new approach to analyse the benefits of rectifier networks’ depth.

ii) **The first investigation on the efficiency of deep rectifier networks in approximating class boundaries** (Sec. 5-6). Given that the ultimate goal of deep learning for classification tasks is to learn class boundaries rather than classifier functions, it is critical to investigate the properties of deep neural networks in generating class boundaries in order to understand the benefits of networks’ depth for classification tasks.
iii) An explicit upper bound is provided on the number of facets that a single hidden layer rectifier can generate with a given number of units (Sec. 5). This upper bound shows the limitations of single hidden layer networks and implies that deep nets have the potential to improve efficiency. For approximations of general convex boundaries in \( \mathbb{R}^d \), we show that the required number of facets is \( O \left( \frac{d^2}{\epsilon} \right)^{\frac{d-1}{2}} \) for a threshold \( \epsilon \) on the approximation error. To learn a convex PWL classifier with this large number of facets using a simple maxout network of some linear units, \( O \left( \frac{d^2}{\epsilon} \right)^{\frac{d-1}{2}} \) linear units are required and \( O \left( d \left( \frac{d}{\epsilon} \right)^{\frac{d+1}{2}} \right) \) weights need to be learnt (Sec. 4). To use single hidden layer rectifier networks, the number of weights to be learnt is at least \( O \left( d^2 \left( \frac{d}{\epsilon} \right)^{\frac{d}{2}} \right) \) for approximations of general convex boundaries (including spherical surfaces). (Sec. 5).

iv) The superior compressive power of deep rectifier networks is demonstrated by constructing a rectifier network for spherical boundary approximation. The constructed network can learn the spherical boundary with \( O \left( d \log \left( \frac{d}{\epsilon} \right) \right) \) units, each requiring at most 5 parameters to be learnt (Sec. 6). As deep rectifier networks have the advantage to use a significantly smaller number of parameters to model the classifier boundaries, they usually thus generalize better than single hidden layer rectifier neural networks.

The rest of this paper is organised as follows. Section 2 addresses related work. Section 3 defines the resolution of PWL classifier boundaries and relates the resolution of convex PWL classifier boundaries to the number of linear units in the maxout representation of convex PWL classifiers. Section 4 addresses the required resolution of general convex classifier boundaries and show that an exponentially large number of facets is required to approximate a convex boundary even for spherical surfaces (the simplest convex surfaces). Section 5 presents the limit of single hidden layer rectifier networks in reducing the number of weights to be learnt for high resolution boundary representation. In Section 6, a solution with deep rectifier networks for the approximation of Euclidean balls is presented to show its superior efficiency over single hidden layer nets. Concluding remarks are provided in Section 7.

2. Related Work

The depth of neural networks has been investigated extensively in recent years to show the superior expressive power of deep neural networks over shallow networks. Delalleau and Bengio (2011) showed that the deep network representation of a certain family of polynomials can be much more compact (i.e., with fewer hidden units) than that provided by a shallow network. Similarly, with the same number of hidden units, deep networks are able to split their input space into many more regions of linearity than their shallow counterparts (Pascaru et al., 2014; Montúfar et al., 2014). Eldan and Shamir (2016) presented an example function that is expressible by a small 3-layer neural networks, but cannot be approximated by a 2-layer network to a certain constant accuracy unless its width is exponential to the dimension of the data. Cohen et al. (2015) proved that, except for a negligible set, all functions that can be implemented by a deep network of polynomial size, require an exponential size in order to be realized (or even approximated) by a shallow network. Mhaskar et al. (2016) demonstrated that deep networks can approximate the class of compositional functions with the same accuracy as shallow networks, but with an exponentially lower number of training param-
The superior expressive power of deep residual nets was analysed in (Veit et al., 2016) which showed that residual nets can be viewed as a collection of many paths of differing lengths, enabling very deep networks by activating only the short paths during training.

The most related work to this paper is the analysis about the maximum number of regions that can be split by a rectifier neural network with a given number of units. With hand-coded construction of deep rectifier nets, Montufar et al. (2014) showed that deep nets can be exponentially more efficient in splitting the space into a large number of regions, while Raghu et al. (2016) presented an upper bound on the number of the split regions by a single hidden layer network with a given number of linear units. From these results, one can conclude that the complexity on the split regions by rectifier networks could grow exponentially with depth. However, it is not yet understood why the growth in complexity on the split regions of linearity improves generalization performance.

All these related works focus on the properties of functions that can be represented by neural networks. However, the ultimate goal of classification is to find class boundaries whose function representations, however, are not unique. Furthermore, the complexity of the different function representations of one identical class boundary can be arbitrarily large. For example, the boundary \( \{ x : g(f(x)) = 0 \} \) is the same as the boundary \( \{ x : f(x) = 0 \} \) for any strictly increasing function \( g(z) \) with \( g(0) = 0 \). Consequently, the complexity analysis on the functions represented by neural networks is not directly on the complexity of class boundaries. Fortunately, there is a rich history on the approximations of Euclidean Balls (Gordon et al., 1994) and general convex bodies (Macbeath, 1951), which will be shown to be closely related to rectifier/maxout networks. This paper will investigate the approximations of convex class boundaries and show the superior power of deep rectifier networks over single hidden layer networks.

3. Resolution of PWL Classifier Boundaries

In this section, we first define the resolution of a PWL classifier boundary as the number of exposed facets on the boundary, and consider the resolution of convex PWL classifier boundaries in particular. Since a general class boundary consists of one or more convex or concave subsets, the required resolution for the approximation of convex boundaries provides a lower bound on the required resolution for the approximation of general class boundaries.

Consider the boundary, namely

\[
B_f = \{ x : f(x) = 0 \},
\]

of a binary PWL classifier \( f(x) \) which classifies the patterns \( x \) to be positive if \( f(x) > 0 \) or negative otherwise. Since \( f(x) \) is a PWL classifier, its boundary consists of a number of facets each satisfying a linear equation \( a_k^T x + c_k = 0 \), more precisely

\[
B_f = \bigcup_{k=1}^{n} \Omega_k, \quad \Omega_k = \{ x : a_k^T x + b_k = 0, f(x) = 0 \}.
\]

Without loss of generality, we assume that none of the facets is redundant, that is,

\[
\bigcup_{k \neq i} \Omega_k \neq B_f, \forall i.
\]
If there are any redundancies, one can always remove them and the remaining is then a set of facets without redundancy. We say $B_f$ has $n$ facets if it has $n$ distinct facets and none of them is redundant.

Since any continuous function can be approximated by a PWL function, the boundary of any classifier that can be represented by a continuous function can also be approximated by a sufficiently larger number of facets around the boundary. For high accuracy approximation, a large number of facets is required in general. The resolution of PWL classifier boundaries is similar to the resolution of digital images, the later represents the quality of digital images for the approximation of natural scenes while the former represents the quality of PWL classifiers for the approximation of class boundaries. In this paper, we focus on the approximations of convex boundaries with convex PWL classifiers. A general smooth class boundary can be viewed as the union of a number of convex/concave surfaces. Next, we address the resolution of convex PWL classifier boundaries.

3.1 Convex PWL classifiers and Their Resolutions

A PWL function $f(x)$ is said to be convex if the set $\{x : f(x) \leq 0\}$ is convex. For a convex PWL function $f(x)$ in $\mathbb{R}^d$, the set $\{x : f(x) \leq 0\}$ is a polytope which can be described as the intersection of a finite number, say $m$, of half planes, i.e.,

$$\{x : f(x) \leq 0\} = \bigcap_{i=1}^{m} \{x : w_i^T x + b_i \leq 0\} = \{x : \max_{1 \leq i \leq m} w_i^T x + b_i \leq 0\} \quad (4)$$

where $w_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$. Therefore, any convex PWL classifier has a maxout representation, i.e.

$$f(x) = \max_{1 \leq i \leq m} w_i^T x + b_i. \quad (5)$$

A maxout representation, as in Eq. (5), of a convex PWL classifier $f(x)$ is said to be irreducible if

$$\{x : f(x) \leq 0\} \not\subset \{x : w_k^T x + b_k \leq 0\}, \quad \forall 1 \leq k \leq m. \quad (6)$$

A convex PWL classifier with an irreducible maxout representation of $m$ units has $m$ facets on its boundary $B_f$ and therefore its resolution is exactly the number of units in its maxout representation. Hereafter, we assume that the maxout representation of a PWL classifier is irreducible. If any unit is reducible, one can always remove it and the classifier remains the same (i.e., the classified label of any pattern $x$ is invariant). For convenience, we use $\mathcal{M}_m$ to denote the set of convex PWL classifiers with irreducible maxout representation of $m$ units, i.e.,

$$\mathcal{M}_m \triangleq \{f(x) = \max_{1 \leq i \leq m} w_i^T x + b_i : w_i \in \mathbb{R}^d, b_i \in \mathbb{R}\}. \quad (7)$$

For any convex PWL classifier $f(x) \in \mathcal{M}_m$, we will use $P_f$ to denote the polytope where $f(x)$ is not positive, i.e.,

$$P_f \triangleq \{x : f(x) \leq 0\}. \quad (8)$$

In the next section, we will consider the required resolutions of convex PWL classifiers for the approximation of general convex boundaries. The required resolution (i.e., the number of required facets) will be used in Section 5 to estimate the number of required units for a single hidden layer rectifier network to approximate general smooth convex boundaries in high dimensional spaces.
4. Required Resolution for the Approximations of Smooth Convex Boundaries

This section starts with spherical boundaries and then moves to general smooth convex boundaries. Surprisingly, for a given threshold on the approximation errors, the required resolution for a general convex surfaces is no higher than the required resolution for spherical surfaces (the simplest convex surfaces except for hyperplanes).

4.1 Approximation of Spherical Boundaries

Given that the boundary of a convex body is convex and the convex combinations of the points in a convex boundary form a convex body, the approximation of convex boundaries with convex PWL classifiers, which generates polytopic boundaries, is equivalent to the volume approximation of convex bodies with polytopes. Next, we will use the results on the polytopic approximations of Euclidean balls (Gordon et al., 1994) to derive the required number of facets to approximate spherical boundaries with given error thresholds. The approximation of general convex boundaries will be addressed in Section 4.2 using the results on the polytopic approximations of general convex bodies (Macbeath, 1951; Schneider, 1967; Paouris and Pivovarov, 2017). From (Gordon et al., 1994) [Theorem 5.2], we have

**Proposition 1** Let \( B^d_2 \) be the unit Euclidean ball in \( \mathbb{R}^d \):

\[
B^d_2 = \left\{ x \in \mathbb{R}^d : \|x\|^2 \leq 1 \right\}.
\]  

(9)

There exists two constants \( C_1, C \) (i.e., independent of \( d \)) such that for every integer \( n \geq C_1 (\log d)^d \), it is possible to construct a polytope \( P_n^d \) containing \( B^d_2 \), with at most \( n \) facets, such that

\[
\frac{|P_n^d \setminus B^d_2|}{|B^d_2|} \leq C \frac{d}{n^{\frac{d}{d-1}}}.
\]  

(10)

where the \(|S|\) represents the volume of a compact set \( S \) and

\[
P_n^d \setminus B^d_2 \triangleq \left\{ x : x \in P_n^d, x \not\in B^d_2 \right\}.
\]  

(11)

Proposition 1 shows that, for a given threshold \( \epsilon \) on the approximation error, the number of facets required to approximate an Euclidean ball is

\[
\left( \frac{C d}{\epsilon} \right)^{\frac{d-1}{2}}.
\]  

(12)

Note that

\[
O \left( (\log d)^d \right) \ll O \left( \left( \frac{d}{\epsilon} \right)^{\frac{d-1}{2}} \right),
\]  

(13)

we have the following corollary from Proposition 1:

**Corollary 2** For the unit Euclidean ball \( B^d_2 \) in \( \mathbb{R}^d \), there exists a constant \( C \) (i.e., independent of \( d \)) such that for every small positive number \( \epsilon \) and every integer

\[
n \geq \left( \frac{C d}{\epsilon} \right)^{\frac{d-1}{2}},
\]  

(14)
it is possible to construct a polytope \( P_n \) containing \( B^d_2 \), with at most \( n \) facets, such that
\[
\frac{|P_n \setminus B^d_2|}{|B^d_2|} \leq \epsilon.
\]  

Corollary 2 provides the required number of facets to approximate Euclidean balls. The next section will generalize this result to general convex bodies and show that the required resolution is no higher than the approximation of Euclidean balls.

### 4.2 Approximation of General Convex Boundaries

Let \( \mathcal{P}_m \) be the set of convex polytopes with \( m \) vertices, \( K \) be a convex body (i.e., a bounded closed convex set with inner points), both in \( \mathbb{R}^d \). Then define the function of a convex body
\[
\phi_m(K) \triangleq \sup_{P \subset K, P \in \mathcal{P}_m} \frac{|P|}{|K|}.
\]

where \( |\cdot| \) denotes the volume of a convex body. According to Macbeath (1951),
\[
\phi_m(K) \geq \phi_m(B^d_2)
\]

holds for any convex body \( K \).

Similarly, let \( \hat{\mathcal{P}}_m \) be the set of convex polytopes with \( m \) facets and define the function of a convex body
\[
\hat{\phi}_m(K) \triangleq \inf_{K \subset P, P \in \hat{\mathcal{P}}_m} \frac{|P|}{|K|}.
\]

From Schneider, 1967 (or Paouris and Pivovarov, 2017) (Corollary 5.2 for a recent reference), we have
\[
\hat{\phi}_m(K) \leq \hat{\phi}_m(B^d_2).
\]

This indicates that, for a given threshold on the approximation errors, the approximation of a general convex body requires a smaller or equal number of facets than the approximation of the Euclidean balls. Combine the above discussions with Corollary 2 and the definition of \( \mathcal{M}_m \) in Eq. (7), we have

**Theorem 3** Let \( \epsilon > 0 \) be a small positive number. There is a constant \( C \) (independent of the dimension \( d \)) such that for any bounded convex body \( K \) and any
\[
m \geq \left( \frac{Cd}{\epsilon} \right)^{\frac{d+1}{2}},
\]

there exists \( f(x) \in \mathcal{M}_m \) such that \( K \subset P_f \) and
\[
\frac{|P_f \setminus K|}{|K|} \leq \epsilon
\]

where \( P_f \triangleq \{ x : f(x) \leq 0 \} \) is a polytope with \( m \) facets.
Theorem 3 shows that the representation of a general convex boundary requires an exponentially high resolution no matter the shape of the boundary is simple or complex. Learning the parameters of the linear units in a maxout network to approximate a spherical boundary will be more prone to overfitting than searching for the center and radius directly. To overcome the overfitting problem, more compact PWL representations are required. Thanks to the symmetry of Euclidean balls, a group of facets that can be represented compactly with a small number of independent parameters, such as those in a regular polytope, can be used to approximate a spherical boundary. In the next section, we will show the limitations of single hidden layer nets in compressing the representation of high resolution class boundaries. The superior compressive power of deep rectifier networks will be considered in Section 6.

5. Limit on the Compressive Power of Single Hidden Layer Rectifier Networks

The number of facets that can be generated by a single hidden layer rectifier network is closely related to the number of regions into which the space is partitioned by the linear units of the network. In this section, we first give a brief review on results relating to the partition of space $\mathbb{R}^d$ by Hyperplanes. We then show how these results relate to the compressive power of single hidden layer rectifier networks.

Let $\mathcal{H}$ be a family of $m$ distinct hyperplanes in the $d$-dimensional space, we denote by $G_{\mathcal{H}}(d, m)$ the number of regions into which the space $\mathbb{R}^d$ is partitioned by the hyperplanes in $\mathcal{H}$. We denote by $G(d, m)$ the maximum of $G_{\mathcal{H}}(d, m)$ across all the possible sets, namely $\mathcal{H}$, of $m$ hyperplanes in $\mathbb{R}^d$, that is,

$$G(d, m) = \max_{\mathcal{H}} G_{\mathcal{H}}(d, m).$$

(22)

A set $\mathcal{H}$ of $m$ hyperplanes, namely $\{x : w_i^T x + b_i = 0\}$, in $\mathbb{R}^d$ is said to be in general position if for each $1 \leq k < d$, no $k + 1$ members of $\mathcal{H}$ contain a common $(d - k)$-dimensional affine subset of $\mathbb{R}^d$, that is, any $(d + 1)$ of $m$ hyperplanes has no common point if $m > d$, and all the $w_i$ are independent if $m \leq d$.

A region $D$ in $\mathbb{R}^d$ is called cone-like if whenever an $(d - 1)$-dimensional plane cuts into components $D_1$ and $D_2$, such that the cross-section is bounded, one of the two components will be bounded and the other unbounded.

From [Ho and Zimmerman (2006)], we have

**Proposition 4** The maximum number, denoted by $G(d, m)$, of regions in a $d$-dimensional space cut by $m$ hyperplanes is achieved when these $m$ hyperplanes are in general position and

$$G(d, m) = \sum_{k=0}^{d} \binom{m}{k}.$$  

(23)

Furthermore, for any set of $m$ planes in general position in $\mathbb{R}^d$, among the $G(d, m)$ regions cut by these $m$ planes, $\binom{m - 1}{d}$ of them are bounded, and each of the remaining

$$C(d, m) \triangleq G(d, m) - \binom{m - 1}{d} = 2 \sum_{k=0}^{d-1} \binom{m - 1}{k}$$

(24)

unbounded regions is always cone-like.
Next, we present the main result of this section, regarding the limit on the resolution of single hidden layer rectifier networks.

**Theorem 5** Let $\mathcal{SHL}(d,m)$ be the set of PWL functions that can be described by a single hidden layer rectifier network with $m$ rectifier linear units in $\mathbb{R}^d$, i.e.,

$$\mathcal{SHL}(d,m) = \left\{ f(x) = a^T \max(0, Wx + b) + c : a, b \in \mathbb{R}^m, W \in \mathbb{R}^{m \times d}, c \in \mathbb{R}, x \in \mathbb{R}^d \right\}. \quad (25)$$

where $a, W, b, c$ are parameters of the function while $x$ is the variable. Then the maximum number of facets of the boundaries $B_f(\triangleq \{x : f(x) = 0\})$ across all the PWL functions in $\mathcal{SHL}(d,m)$ is between $C(d,m)$ and $G(d,m)$, that is

$$C(d,m) - 1 \leq \max_{f(x) \in \mathcal{SHL}(d,m)} \#(B_f) \leq G(d,m) \quad (26)$$

where $\#(B_f)$ denotes the number of facets of $B_f$, and $G(d,m), C(d,m)$ are defined in Eq. (23) and Eq. (24) respectively.

**Proof:** Let $h_i(x) = w_i^T x + b_i (1 \leq i \leq m)$ be the $m$ linear units of $f(x) = a^T \max(0, Wx + b) + c$ where $w_i^T$ is the $i^{th}$ row of $W$ and $b_i$ is the $i^{th}$ element of $b$. According to Proposition [4] the space $\mathbb{R}^d$ can be cut into at most $G(d,m)$ regions by the $m$ hyperplanes $\{x : h_i(x) = 0\}$. Within each of these regions, the sign of $h_i(x)$ is invariant and thus $f(x)$ is a linear function when $x$ is constrained within one of these regions. Therefore the boundary $B_f$ has at most one facet in each of these regions and the maximum number of facets of $B_f$ across all the functions in $\mathcal{SHL}(d,m)$ is upper bounded by $G(d,m)$.

For the lower bound ($C(d,m) - 1$), we consider the conditions under which $B_f$ has exactly $(C(d,m) - 1)$ facets. First, let us choose the linear units such that the associated $m$ hyperplanes $\{h_i(x) = 0\}$ are in general position. Then, according to Proposition [4] these $m$ hyperplanes cut the space $\mathbb{R}^d$ into $G(d,m)$ regions where $C(d,m)$ of them are unbounded. Since $f(x)$ is linear when $x$ is constrained within each of the $G(d,m)$ regions, $f(x)$ has a bounded maximum in the $(G(d,m) - C(d,m))$ bounded and closed regions. Furthermore, in the $C(d,m)$ unbounded regions, there is at most one region within which all the linear units $h_i(x)$ are negative and therefore $f(x)$ is unbounded in at least $(C(d,m) - 1)$ of the unbounded regions. Hence, if all the elements of $a^T$ are positive, and $c$ is sufficiently small (e.g. approaching to $-\infty$) such that $f(x)$ is negative in all the bounded regions, then $B_f$ has a facet in each of the $C(d,m)$ unbounded regions except the one (if any) within which all the linear units $h_i(x)$ are negative. Thus, the maximum number of facets of $B_f$ across all $f(x) \in \mathcal{SHL}(d,m)$ is lower bounded by $(C(d,m) - 1)$.

To estimate the required number of units to approximate a general convex boundary by an SHL rectifier network, the following lemma is also required.

**Lemma 6** Let $G(d,m)$ be the maximum number of regions in a $d$-dimensional space cut by $m$ hyperplanes, then

$$G(d,m) = 2^m, \quad \forall \ m \leq d, \quad (27)$$

$$G(d,m) \leq \frac{1}{(d-1)!} m^d, \quad \forall \ m \geq d. \quad (28)$$
**Proof:** Note that

\[ \sum_{k=0}^{m} \binom{m}{k} = 2^m \]  
(29)

and \( \binom{m}{k} = 0 \) if \( k > m \). Then Eq. (27) follows from Proposition 4.

For the proof of Eq. (28), by induction, it suffices to prove the following three statements:

i) Eq. (28) holds when \( d = 2 \);

ii) Eq. (28) holds when \( m = d \);

iii) For any \( k \geq 2, l \geq k + 1 \), if Eq. (28) holds when \( m = l, d = k, k + 1 \), then Eq. (28) holds when \( m = l + 1, d = k + 1 \).

For the proof of statement i), it is easy to check that \( G(2, m) = m + 1 + \frac{m(m-1)}{2} \) and therefore \( G(2, m) \leq m^2 \) when \( m \geq 2 \). Thus Eq. (28) holds when \( d = 2 \). This proves i).

Next we prove statement ii), i.e.

\[ 2^d \leq \frac{1}{(d-1)!} d^d. \]  
(30)

This can be done by induction. It is easy to check that Eq. (30) holds when \( d = 2 \). Now assume that Eq. (30) holds when \( d = k \) for some \( k \geq 2 \), that is

\[ 2^k \leq \frac{1}{(k-1)!} k^k. \]  
(31)

Note that, for any integer \( n \geq 2 \) and any positive number \( x > 0 \), we have

\[ (x + 1)^n = \sum_{i=0}^{n} \binom{n}{i} x^i \geq x^n + n x^{n-1} \]  
(32)

and therefore

\[ (k + 1)^{k+1} \geq k^{k+1} + (k + 1)k^k \]  
(33)

which implies that

\[ \frac{1}{k!} (k + 1)^{k+1} \geq \frac{k^{k+1}}{k!} + \frac{(k + 1)k^k}{k!} \geq 2 \frac{k^k}{(k-1)!}. \]  
(34)

Then, from (31), it follows

\[ \frac{1}{k!} (k + 1)^{k+1} \geq 2^{k+1}. \]  
(35)

That is, Eq. (30) holds for \( d = k + 1 \) as well. This completes the induction and completes the proof of Eq. (30).

For the proof of statement iii), assume that Eq. (28) holds when \( m = l, d = k, k + 1 \), that is

\[ G(k, l) \leq \frac{1}{(l-1)!} l^k \]
\[ G(k + 1, l) \leq \frac{1}{l!} l^{k+1}. \]  
(36)
From the following recursive relations of combinations

\[
\binom{l+1}{i} = \binom{l}{i} + \binom{l}{i-1},
\]

we have

\[
G(k+1, l+1) = \sum_{i=0}^{k+1} \binom{l+1}{i} = \sum_{i=0}^{k} \binom{l}{i} + \binom{l}{i} + \sum_{i=1}^{k+1} \binom{l}{i-1} = G(k+1, l) + \sum_{j=0}^{k} \binom{l}{j} = G(k+1, l) + G(k, l).
\]

Then, from (36), it follows that

\[
G(k+1, l+1) \leq \frac{1}{k!} l^{k+1} + \frac{1}{(k-1)!} l^k = \frac{1}{k!} (l^{k+1} + kl^k) \leq \frac{1}{k!} (l+1)^{k+1}
\]

which implies that Eq. (28) holds for \(d = k + 1, m = l + 1\) and the proof is completed.

Now we are ready to estimate the required number of units for a single hidden layer network to approximate general convex boundaries.

From Eq. (28), to generate \((Cd/\epsilon)^{d+1}\) number of facets by a single hidden layer net, the number of units, namely \(m\), should satisfy

\[
\frac{1}{(d-1)!} m^d \geq \left(\frac{Cd}{\epsilon}\right)^{d+1},
\]

and therefore

\[
m \geq \left(\frac{Cd}{\epsilon}\right)^{\frac{d+1}{d}} \{ (d-1)! \}^{\frac{1}{d}} \approx \left(\frac{Cd}{\epsilon}\right)^{\frac{1}{d}} \{ (d-1)! \}^{\frac{1}{d}}.
\]

From Stirling’s formula for approximation of \(n!\) (Romik, 2000), we have

\[
d! \geq \sqrt{2\pi d^{d+0.5}} e^{-d}
\]

and therefore

\[
m \geq \left(\frac{Cd}{\epsilon}\right)^{\frac{1}{d}} \{ (d-1)! \}^{\frac{1}{d}} \approx \left(\frac{Cd}{\epsilon}\right)^{\frac{1}{d}} \frac{d-1}{e}.
\]

where \(e \approx 2.7183\) is the Euler number.

Theorem 7 below summarises the result of this section.
Theorem 7 To approximate the Euclidean ball in $d$ dimensional space with $(\frac{Cd}{\epsilon})^{\frac{d-1}{2}}$ facets by a single hidden layer rectifier net, at least

$$N_s = \left(\frac{Cd}{\epsilon}\right)^{\frac{1}{2}} \frac{d-1}{e}$$  \hspace{1cm} (45)

units are required, with

$$N_{sp} = (d+1)N_s = \left(\frac{Cd}{\epsilon}\right)^{\frac{1}{2}} \frac{d^2-1}{e}$$  \hspace{1cm} (46)

parameters to be learnt.

In the next section, we will show that a deep rectifier net is able to approximate Euclidean balls with a much smaller number of units and parameters.

6. Superior Compressive Power of Deep Rectifier Networks

We will first construct a deep rectifier network for two dimensional data in Section 6.1 and then construct a deep rectifier network in Section 6.2 for efficient approximation of high dimensional spherical boundaries.

6.1 Two Dimensional Space

Consider the approximation of the unit circle $\{x \in \mathbb{R}^2 : \|x\| = 1\}$. From Theorem 5, if a single hidden layer net of $m$ units is used, the maximum number of segments is $G(2,m)(= m + 1 + \frac{m(m+1)}{2})$. Next we present a deep rectifier net which can approximate the unit circle more efficiently than single hidden layer nets.

Lemma 8 Let $\theta_k = \frac{\pi}{2k}$, and $f_k(x, y), \bar{f}_k(x, y)$ be defined recursively as

$$f_1(x, y) = |x|, \bar{f}_1(x, y) = |y|$$

$$f_{k+1}(x, y) = \cos \theta_{k+1} f_k(x, y) + \sin \theta_{k+1} \bar{f}_k(x, y)$$

$$\bar{f}_{k+1}(x, y) = -\sin \theta_{k+1} f_k(x, y) + \cos \theta_{k+1} \bar{f}_k(x, y).$$  \hspace{1cm} (47)

Then for any $[x, y]^T \in \mathbb{R}^2$, $k = 1, 2, \ldots$, we have

$$1 - \frac{x^2}{2k+1} < \cos \theta_k \leq \frac{f_k(x, y)}{\sqrt{x^2+y^2}} \leq 1.$$  \hspace{1cm} (48)

Proof: Let $f_0(x, y) = x = r \cos t, \bar{f}_0(x, y) = y = r \sin t, t \in [0, 2\pi]$ where $r = \sqrt{x^2 + y^2}$, then

$$f_1(x, y) = |x| = r |\cos t| = r \cos t_1; \bar{f}_1(x, y) = |y| = r |\sin t| = r \sin t_1$$  \hspace{1cm} (49)

where $t_1 \in \left[0, \frac{\pi}{2}\right]$ and

$$t_1 = \begin{cases} t & \text{if } t \in [0, \pi/2]; \\ \pi - t & \text{if } t \in (\pi/2, \pi]; \\ t - \pi & \text{if } t \in (\pi, 3\pi/2]; \\ 2\pi - t & \text{if } t \in (3\pi/2, 2\pi]. \end{cases}$$  \hspace{1cm} (50)
Next assume that \( f_k = r \cos t_k \), \( \bar{f}_k = r \cos t_k \) holds some \( k \geq 1 \) and some \( t_k \in [0, \theta_k] \), where \( \theta_k = \frac{\pi}{2} \). Then
\[
\begin{align*}
    f_{k+1} &= r \cos \theta_{k+1} \cos t_k + r \sin \theta_{k+1} \sin t_k = r \cos(t_k - \theta_{k+1}) \\
    \bar{f}_{k+1} &= r |\sin \theta_{k+1} \cos t_k + \cos \theta_{k+1} \sin t_k| = r |\sin(t_k - \theta_{k+1})|.
\end{align*}
\]

Let
\[
t_{k+1} = |t_k - \theta_{k+1}| \in [0, \theta_k],
\]
then \( f_{k+1} = r \cos t_{k+1}, \bar{f}_{k+1} = r \sin t_{k+1} \). By induction, for any \( k \geq 1 \), \( f_k = r \cos t_k, \bar{f}_k = r \cos t_k \) holds for some \( t_k \in [0, \theta_k] \) and therefore
\[
\frac{f_k}{\sqrt{x^2 + y^2}} = \frac{r \cos t_k}{r} \geq \cos \theta_k
\]
which completes the proof.

To implement the function \( f_k(x, y) \) for some integer \( k \), the rectifier net can be constructed as follows:

1). The first hidden layer has four nodes which are chosen as below:
\[
\begin{align*}
    z_1(1) &= \max(x_1, 0) \\
    z_1(2) &= \max(-x_1, 0) \\
    z_1(3) &= \max(x_2, 0) \\
    z_1(4) &= \max(-x_2, 0)
\end{align*}
\]

2). The second hidden layer has three nodes:
\[
\begin{align*}
    z_2(1) &= \max(0, |x_1| \cos \theta_2 + |x_2| \sin \theta_2) \\
    z_2(2) &= \max(0, |x_1| \cos \theta_2 - |x_2| \sin \theta_2) \\
    z_2(3) &= \max(0, |x_2| \sin \theta_2 - |x_1| \cos \theta_2)
\end{align*}
\]
where
\[
\begin{align*}
    \theta_2 &= \frac{\pi}{4} \\
    |x_1| &= z_1(1) + z_1(2) \\
    |x_2| &= z_1(3) + z_1(4).
\end{align*}
\]

3). The \((i + 1)^{th} (i = 2, \ldots, k - 2)\) layer has three nodes which are chosen as
\[
\begin{align*}
    z_{i+1}(1) &= \max \{0, z_i(1) \cos \theta_{i+1} + (z_i(2) + z_i(3)) \sin \theta_{i+1}\} \\
    z_{i+1}(2) &= \max \{0, z_i(1) \cos \theta_{i+1} + (z_i(2) - z_i(3)) \sin \theta_{i+1}\} \\
    z_{i+1}(3) &= \max \{0, (z_i(2) - z_i(3)) \sin \theta_{i+1} - z_i(1) \cos \theta_{i+1}\}
\end{align*}
\]
where
\[
\theta_{i+1} = \frac{\theta_i}{2}.
\]
4. The output layer is then

\[ f_k = z_{k-1}(1) \cos \theta_k + (z_{k-1}(2) + z_{k-1}(3)) \sin \theta_k - 1 \]  \hspace{1cm} (59)

where \( \theta_k = \frac{\theta_{k-1}}{2} \).

The constructed rectifier net has \( 3(k - 1) + 2 = 3k - 1 \) units in total and can generate \( 2^k \) segments to approximate a circle. The following Lemma summarises the main result of this section.

**Lemma 9** A rectifier net with \( m = 3k - 1 \) units in \( k \) \((\geq 2)\) layers exists to generate a polytope

\[ P_n = \{(x, y)^T : f_k(x, y) \leq 1\}, \]  \hspace{1cm} (60)

of \( n = 2^{m+1} \) number of segments, to approximate the unit circle \( B^2_2 \) such that \( B^2_2 \subset P_n \) and

\[ \frac{|P_n - B^2_2|}{|B^2_2|} \leq \frac{1}{\cos^2 \theta_k} - 1 \approx \theta_k^2 = \frac{\pi^2}{2^{2k}}. \]  \hspace{1cm} (61)

**Proof:** From Lemma 8, it follows that \( B^2_2 \subset P_n \) and

\[ x^2 + y^2 \leq \frac{1}{\cos^2 \theta_k}, \forall (x, y) \in P_n \]  \hspace{1cm} (62)

which implies that \( P_n \subset \{(x, y)^T : x^2 + y^2 \leq \frac{1}{\cos^2 \theta_k}\} \) and therefore

\[ \frac{|P_n - B^2_2|}{|B^2_2|} \leq \frac{\frac{1}{\cos^2 \theta_k}B^2_2 - |B^2_2|}{|B^2_2|} = \frac{1}{\cos^2 \theta_k} - 1 \approx \theta_k^2 = \frac{\pi^2}{2^{2k}}. \]  \hspace{1cm} (63)

\[ \blacksquare \]

### 6.2 High Dimensional Spaces

For high dimensional cases, we will first use the constructed 2D rectifier network to approximate \( \sqrt{x_1^2 + x_2^2} \) (i.e., the norm of the first two elements). Then with \( x_3 \) as one input and the output of the previous 2D network as another input, we apply the same 2D network to approximate the norm of the first three elements. Sequentially, by applying the 2D rectifier net similarly for \( (d - 1) \) times, the norm of \( d \)-dimensional vectors can be approximated. The following lemma will give the bound for the approximation error and will be used to estimate the approximation error for the approximation of unit balls using the constructed deep rectifier network.

**Lemma 10** Let \( x \in \mathbb{R}^d \) and \( g_l(x, k) \) be defined recursively as below

\[ g_1(x; k) = f_k(x_1, x_2) \]
\[ g_l(x; k) = f_k\{g_{l-1}(x; k), x_{l+1}\}, 2 \leq l \leq d - 1 \]  \hspace{1cm} (64)

where \( f_k(\cdot, \cdot) \) is a function defined in Lemma 8 with \( \theta_k = \frac{\pi}{2^k} \). Then for any \( x \in \mathbb{R}^d \), \( k = 1, 2, \cdots \), we have

\[ \cos^l \theta_k \leq \frac{g_l(x, k)}{\sqrt{\sum_i x_i^2}} \leq 1, \forall 1 \leq l \leq d - 1. \]  \hspace{1cm} (65)
**Proof:** From Lemma 8 we have

\[
\begin{align*}
\cos \theta_k & \leq \frac{g_d(x; k)}{\sqrt{x_1^2 + x_2}} \leq 1 \\
\cos \theta_k & \leq \frac{g_1(x; k)}{\sqrt{g_{l-1} + x_{l+1}^2}} \leq 1, \ \forall \ l \geq 2.
\end{align*}
\]  \hspace{1cm} (66)

Hence Eq. (65) holds when \( l = 1 \).

Next, assume that Eq. (65) holds for \( l = p \) for some \( p \geq 1 \), that is

\[
\cos^p \theta_k \leq \frac{g_d(x; k)}{\sqrt{\sum_{i=1}^{p+1} x_i^2}} \leq 1.
\]  \hspace{1cm} (67)

Then from Eq. (66), we have

\[
g_{p+1}^2(x; k) \leq g_p^2(x; k) + x_{p+2}^2 \leq \sum_{i=1}^{p+2} x_i^2
\]  \hspace{1cm} (68)

and

\[
g_{p+1}^2(x; k) \geq \cos^2 \theta_k \left\{ g_p^2(x; k) + x_{p+2}^2 \right\} \geq \cos^2 \theta_k \left( \cos^{2p} \theta_k \sum_{i=1}^{p+1} x_i^2 + x_{p+2}^2 \right) \geq \cos^{2p+2} \sum_{i=1}^{p+2} x_i^2.
\]  \hspace{1cm} (69)

which, together with Eq. (68), imply that Eq. (65) holds for \( l = p + 1 \). By induction, Eq. (65) is true for every \( l = 1, 2, \cdots, d - 1 \).

\[\square\]

Next, we use Lemma 10 to estimate the required number of units to approximate Euclidean balls with a given threshold on the approximation errors. Consider the polytope

\[P \triangleq \{ x : g_{d-1}(x; k) \leq 1 \}\]  \hspace{1cm} (70)

where \( g_{d-1}(x; k) \) is defined in Lemma 10.

From Lemma 10, we have

\[\|x\| \cos^{d-1} \theta_k \leq g_{d-1}(x; k) \leq \|x\| \]  \hspace{1cm} (71)

and therefore

\[B_2^d \subset P \subset B \left( \frac{1}{\cos^{d-1} \theta_k} \right)\]  \hspace{1cm} (72)

where \( B(r) \triangleq \{ x : \|x\| \leq r \} \) is an Euclidean ball in \( \mathbb{R}^d \) with radius \( r \). Note that the volume of an Euclidean ball in \( \mathbb{R}^d \) with radius \( r \) is proportional to \( r^d \), the approximation error of \( P \) to the unit ball \( B_2^d \) satisfies

\[
\frac{|P \setminus B_2^d|}{|B_2^d|} < \left\{ \frac{1}{\cos^{d-1} \theta_k} \right\}^d - 1 = \left\{ \frac{1}{\cos \theta_k} \right\}^{(d-1)d} - 1
\]  \hspace{1cm} (73)
Note that $\theta_k = \frac{\pi}{2}$ is very close to zero when $k \geq 10$, and when $x$ is close to zero,

$$
(1 + \frac{1}{2}x^2)^n \approx 1 + \frac{n}{2}x^2.
$$

(74)

We have

$$
\frac{|P \setminus B_d^n|}{|B_d^n|} < \frac{d(d-1)\theta_k^2}{2d + 1}.
$$

(75)

To meet a threshold $\epsilon$ on the approximation error, it suffices to choose $k$ such that

$$
2k + 1 \geq 2\log(d) + \log(\epsilon^{-1}) + 2\log(\pi)
$$

(76)

or equivalently

$$
k \geq \log(d) + \frac{1}{2}\log(\epsilon^{-1}) - \frac{1}{2} + \log(\pi).
$$

(77)

The following Theorem summarises the result of this section.

**Theorem 11** Let

$$
k^* = \log(d) + \frac{1}{2}\log(\epsilon^{-1}) - \frac{1}{2} + \log(\pi).
$$

(78)

There exists a rectifier network, with $k^*(d - 1)$ layers of

$$
N_d = (d - 1)(3k^* - 1) = (d - 1)\{3\log(d) + 1.5\log(\epsilon^{-1}) + 3\log(\pi) - 2.5\}
$$

(79)

units, which can approximate a $d$ dimensional Euclidean ball such that

$$
\frac{|P_n \setminus B_d^n|}{|B_d^n|} \leq \epsilon.
$$

(80)

**6.3 Advantages of Deep Rectifier Networks**

From Theorem 7 and Theorem 11, one can see that the ratio between $N_s$, the number of required units for single hidden layer nets to approximate Euclidean balls, and $N_d$ is

$$
\frac{N_s}{N_d} \approx \frac{\left(\frac{Cd}{\epsilon}\right)^{\frac{1}{2}}}{e \left(3\log d + 1.5\log(\epsilon^{-1})\right)}
$$

(81)

which shows that, with similar approximation accuracy, single hidden layer nets require much larger number of units than the constructed deep rectifier nets. In particular, when $d$ is small, the number of required facets is dominated by the approximation accuracy and the constructed deep net is exponentially (with the depth) more efficient than single hidden layer nets. Note that each node of the constructed deep rectifier network is connected to 3 or 4 other nodes, each unit has at most 5 parameters to be determined, while each unit of the single hidden layer net has $(d + 1)$ parameters to learn. For large dimension $d$, the constructed deep rectifier network is at least $O(d^{1.5}/\log(d))$ more efficient than single hidden layer neural networks.
7. Concluding Remarks

By introducing the boundary resolution of PWL classifiers, this paper has shown the superior compressive power of deep rectifier networks over single hidden layer rectifier networks for high resolution representation of class boundaries. Due to the requirement of the universal approximation capacity, non-polynomial activation functions such as rectifiers are used in neural networks, but at the cost of exponentially (with respect to data dimension) increased model complexity for the approximation of geometrically-simple class boundaries, such as spherical boundaries or other boundaries that can be represented by a small number of parameters. To learn such geometrically-simple boundaries, deep neural nets are required to learn compact models for the purpose of good generalization by exploiting the symmetric properties of class boundaries.

References

Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 3642–3649. IEEE, 2012.

Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. arXiv preprint arXiv:1509.05009, 554, 2015.

Olivier Delalleau and Yoshua Bengio. Shallow vs. deep sum-product networks. In Advances in Neural Information Processing Systems, pages 666–674, 2011.

Richard M Dudley. Metric entropy of some classes of sets with differentiable boundaries. Journal of Approximation Theory, 10(3):227–236, 1974.

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In Conference on Learning Theory, pages 907–940, 2016.

Yehoram Gordon, Mathieu Meyer, and Shlomo Reisner. Volume approximation of convex bodies by polytopes—a constructive method. Studia Mathematica, 1(111):81–95, 1994.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In European Conference on Computer Vision, pages 630–645. Springer, 2016b.

Chungwu Ho and Seth Zimmerman. On the number of regions in an m-dimensional space cut by n hyperplanes. Gazette of the Australian Mathematical Society, 33(4):260–264, Sept. 2006.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366, 1989.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

AM Macbeath. An extremal property of the hypersphere. In Mathematical Proceedings of the Cambridge Philosophical Society, volume 47, pages 245–247. Cambridge University Press, 1951.

Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. Learning functions: When is deep better than shallow. arXiv preprint arXiv:1603.00988, 2016.
Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *arXiv preprint arXiv:1402.1869*, 2014.

Grigoris Paouris and Peter Pivovarov. Random ball-polyhedra and inequalities for intrinsic volumes. *Monatshefte für Mathematik*, 182(3):709–729, 2017.

Razvan Pascanu, Guido Montufar, and Yoshua Bengio. On the number of inference regions of deep feed forward networks with piece-wise linear activations. In *International Conference on Learning Representations 2014 (Conference Track)*, April 2014. URL [http://arxiv.org/abs/1312.6026](http://arxiv.org/abs/1312.6026).

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. *arXiv preprint arXiv:1606.05336*, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

Dan Romik. Stirling’s approximation for n!: The ultimate short proof? *The American Mathematical Monthly*, 107(6):556, 2000.

Rolf Schneider. Eine allgemeine extremaleigenschaft der kugel. *Monatshefte für Mathematik*, 71(3):231–237, 1967.

Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 2. Wiley New York, 1998.

Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems*, pages 550–558, 2016.