arXiv:1312.6098v5 [cs.LG] 14 Feb 2014

On the number of response regions of deep feedforward networks with piecewise linear activations

Razvan Pascanu
Université de Montréal
Montréal QC H3C 3J7 Canada
r.pascanu@gmail.com

Guido Montúfar
Max Planck Institute for Mathematics in the Sciences
Inselstraße 22, 04103 Leipzig, Germany
montufar@mis.mpg.de

Yoshua Bengio
Université de Montréal
Montréal QC H3C 3J7 Canada
yoshua.bengio@umontreal.ca

Abstract

This paper explores the complexity of deep feedforward networks with linear pre-synaptic couplings and rectified linear activations. This is a contribution to the growing body of work contrasting the representational power of deep and shallow network architectures. In particular, we offer a framework for comparing deep and shallow models that belong to the family of piecewise linear functions based on computational geometry. We look at a deep rectifier multi-layer perceptron (MLP) with linear outputs units and compare it with a single layer version of the model. In the asymptotic regime, when the number of inputs stays constant, if the shallow model has kn hidden units and n_0 inputs, then the number of linear regions is $O(k^{n_0} n^{n_0})$. For a k layer model with n hidden units on each layer it is $\Omega(\lfloor n/n_0 \rfloor^{k-1} n^{n_0})$. The number $\lfloor n/n_0 \rfloor^{k-1}$ grows faster than k^{n_0} when n tends to infinity or when k tends to infinity and $n \geq 2n_0$. Additionally, even when k is small, if we restrict n to be $2n_0$, we can show that a deep model has considerably more linear regions that a shallow one. We consider this as a first step towards understanding the complexity of these models and specifically towards providing suitable mathematical tools for future analysis.

Keywords: Deep learning, artificial neural network, rectifier unit, hyperplane arrangement, representational power

1 Introduction

Deep systems are believed to play an important role in information processing of intelligent agents. A common hypothesis underlying this belief is that deep models can be exponentially more efficient at representing some functions than their shallow counterparts (see Bengio, 2009).

The argument is usually a compositional one. Higher layers in a deep model can *re-use* primitives constructed by the lower layers in order to build gradually more complex functions. For example, on a vision task, one would hope that the first layer learns Gabor filters capable to detect edges of different orientation. These edges are then put together at the second layer to form *part-of-object* shapes. On higher layers, these part-of-object shapes are combined further to obtain detectors for more complex part-of-object shapes or objects. Such a behaviour is empirically illustrated, for

instance, in Zeiler and Fergus (2013); Lee et al. (2009). On the other hand, a shallow model has to construct detectors of target objects based only on the detectors learnt by the first layer.

The representational power of computational systems with shallow and deep architectures has been studied intensively. A well known result Hajnal et al. (1993) derived lower complexity bounds for shallow threshold networks. Other works have explored the representational power of generative models based on Boltzmann machines Montúfar et al. (2011); Martens et al. (2013) and deep belief networks (Sutskever and Hinton, 2008; Le Roux and Bengio, 2010; Montúfar and Ay, 2011), or have compared mixtures and products of experts models (Montúfar and Morton, 2012).

In addition to such inspections, a wealth of evidence for the validity of this hypothesis comes from deep models consistently outperforming shallow ones on a variety of tasks and datasets (see, e.g., Goodfellow et al., 2013; Hinton et al., 2012b,a). However, theoretical results on the representational power of deep models are limited, usually due to the composition of nonlinear functions in deep models, which makes mathematical analysis difficult. Up to now, theoretical results have focussed on circuit operations (neural net unit computations) that are substantially different from those being used in real state-of-the-art deep learning applications, such as logic gates (Håstad, 1986), linear + threshold units with non-negative weights (Håstad and Goldmann, 1991) or polynomials (Bengio and Delalleau, 2011). Bengio and Delalleau (2011) show that deep sum-product networks (Poon and Domingos, 2011) can use exponentially less nodes to express some families of polynomials compared to the shallow ones.

The present note analyzes the representational power of deep MLPs with rectifier units. Rectifier units (Glorot et al., 2011; Nair and Hinton, 2010) and piecewise linearly activated units in general (like the *maxout* unit (Goodfellow et al., 2013)), are becoming popular choices in designing deep models, and most current state-of-the-art results involve using one of such activations (Goodfellow et al., 2013; Hinton et al., 2012b). Glorot et al. (2011) show that rectifier units have several properties that make the optimization problem easier than the more traditional case using smooth and bounded activations, such as *tanh* or *sigmoid*.

In this work we take advantage of the piecewise linear nature of the rectifier unit to mathematically analyze the behaviour of deep rectifier MLPs. Given that the model is a composition of piecewise linear functions, it is itself a piecewise linear function. We compare the flexibility of a deep model with that of a shallow model by counting the number of linear regions they define over the input space for a fixed number of hidden units. This is the number of pieces available to the model in order to approximate some arbitrary nonlinear function. For example, if we want to perfectly approximate some curved boundary between two classes, a rectifier MLP will have to use infinitely many linear regions. In practice we have a finite number of pieces, and if we assume that we can perfectly learn their optimal slopes, then the number of linear regions becomes a good proxy for how well the model approximates this boundary. In this sense, the number of linear regions is an upper bound for the flexibility of the model. In practice, the linear pieces are not independent and the model may not be able to learn the right slope for each linear region. Specifically, for deep models there is a correlation between regions, which results from the sharing of parameters between the functions that describe the output on each region. ?

This is by no means a negative observation. If all the linear regions of the deep model were independent of each other, by having many more linear regions, deep models would grossly overfit. The correlation of the linear regions of a deep model results in its ability to generalize, by allowing it to better represent only a small family of structured functions. These are functions that look complicated (e.g., a distribution with a huge number of modes) but that have an underlying structure that the network can 'compress' into its parameters. The number of regions, which indicates the number of variations that the network can represent, provides a measure of how well it can fit this family of structured functions (whose approximation potentially needs infinitely many linear regions).

We believe that this approach, based on counting the number of linear regions, is extensible to any other piecewise linear activation function and also to other architectures, including the *maxout* activation and the convolutional networks with rectifier activations.

We know the maximal number of regions of linearity of functions computable by a shallow model with a fixed number of hidden units. This number is given by a well studied geometrical problem. The main insight of the present work is to provide a geometrical construction that describes the regions of linearity of functions computed by deep models. We show that in the asymptotic regime,

these functions have many more linear regions than the ones computed by shallow models, for the same number of hidden units.

For the single layer case, each hidden unit divides the input space in two, whereby the boundary is given by a hyperplane. For all input values on one side of the hyperplane, the unit outputs a positive value. For all input values on the other side of the hyperplane, the unit outputs 0. Therefore, the question that we are asking is: Into how many regions do n hyperplanes split space? This question is studied in geometry under the name of hyperplane arrangements, with classic results such as Zaslavsky's theorem. Section 3 provides a quick introduction to the subject.

For the multilayer version of the model we rely on the following intuition. By using the rectifier nonlinearity, we identify multiple regions of the input space which are mapped by a given layer into an equivalent set of activations and represent thus equivalent inputs for the next layers. That is, a hidden layer can perform a kind of *or* operation by reacting similarly to several different inputs. Any subsequent computation made on these activations is replicated on all equivalent inputs.

This paper is organized as follows. In Section 2 we provide definitions and basic observations about piecewise linear functions. In Section 3 we discuss rectifier networks with one single hidden layer and describe their properties in terms of hyperplane arrangements which are fairly well known in the literature. In Section 4 we discuss deep rectifier networks and prove our main result, Theorem 1, which describes their complexity in terms of the number of regions of linearity of functions that they represent. Details about the asymptotic behaviour of the results derived in Sections 3 and 4 are given in the Appendix A. In Section 5 we analyze a special type of deep rectifier MLP and show that even for a small number of hidden layers it can generate a large number of linear regions. In Section 6 we offer a discussion of the results.

2 Preliminaries

We consider classes of functions (models) defined in the following way.

Definition 1. A *rectifier feedforward network* is a layered feedforward network, or multilayer perceptron (MLP), as shown in Fig. 1, with following properties. Each hidden unit receives as inputs the real valued activations $x_1, \ldots, x_n$ of all units in the previous layer, computes the weighted sum

$$s = \sum_{i \in [n]} w_i x_i + b,$$

and outputs the rectified value

$$\mathrm{rect}(s) = \max\{0, s\}.$$

The real parameters $w_1, \ldots, w_n$ are the *input weights* and b is the *bias* of the unit. The output layer is a *linear layer*, that is, the units in the last layer compute a linear combination of their inputs and output it unrectified.

Given a vector of naturals $\boldsymbol{n} = (n_0, n_1, \ldots, n_L)$, we denote by $\mathcal{F}_{\boldsymbol{n}}$ the set of all functions $\mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ that can be computed by a rectifier feedforward network with n_0 inputs and n_l units in layer l for $l \in [L]$. The elements of $\mathcal{F}_{\boldsymbol{n}}$ are continuous piecewise linear functions.

We denote by $\mathcal{R}(\boldsymbol{n})$ the maximum of the number of regions of linearity or *response regions* over all functions from $\mathcal{F}_{\boldsymbol{n}}$. For clarity, given a function $f \colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$, a connected open subset $R \subseteq \mathbb{R}^{n_0}$ is called a *region of linearity* or *linear region* or *response region* of f if the restriction $f|_R$ is a linear function and for any open set $\tilde{R} \supsetneq R$ the restriction $f|_{\tilde{R}}$ is not a linear function. In the next sections we will compute bounds on $\mathcal{R}(\boldsymbol{n})$ for different choices of $\boldsymbol{n}$. We are especially interested in the comparison of shallow networks with one single very wide hidden layer and deep networks with many narrow hidden layers.

In the remainder of this section we state three simple lemmas.

The next lemma states that a piecewise linear function $f = (f_i)_{i \in [k]}$ has as many regions of linearity as there are distinct intersections of regions of linearity of the coordinates f_i.

Lemma 1. *Consider a width k layer of rectifier units. Let $R^i = \{R^i_1, \ldots, R^i_{N_i}\}$ be the regions of linearity of the function $f_i \colon \mathbb{R}^{n_0} \to \mathbb{R}$ computed by the i-th unit, for all $i \in [k]$. Then the regions of*

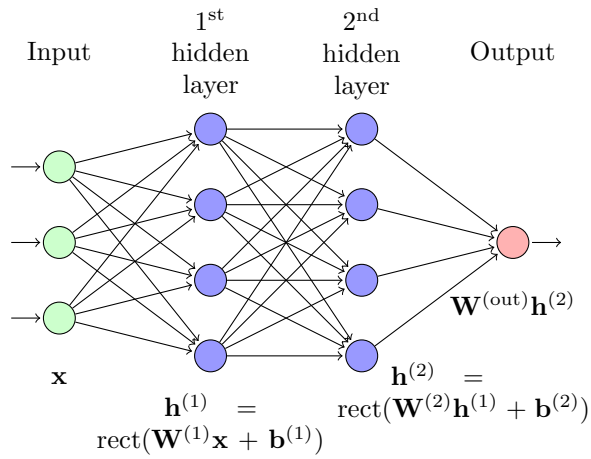


Figure 1: Illustration of a rectifier feedforward network with two hidden layers.

linearity of the function $f = (f_i)_{i\in[k]} \colon \mathbb{R}^{n_0} \to \mathbb{R}^k$ *computed by the rectifier layer are the elements of the set* $\{R_{j_1,\ldots,j_k} = R^1_{j_1} \cap \cdots \cap R^k_{j_k}\}_{(j_1,\ldots,j_k)\in[N_1]\times\cdots\times[N_k]}$.

Proof. A function $f = (f_1, \ldots, f_k) \colon \mathbb{R}^n \to \mathbb{R}^k$ is linear iff all its coordinates $f_1, \ldots, f_k$ are. □

In regard to the number of regions of linearity of the functions represented by rectifier networks, the number of output dimensions, i.e., the number of linear output units, is irrelevant. This is the statement of the next lemma.

Lemma 2. *The number of (linear) output units of a rectifier feedforward network does not affect the maximal number of regions of linearity that it can realize.*

Proof. Let $f \colon \mathbb{R}^{n_0} \to \mathbb{R}^k$ be the map of inputs to activations in the last hidden layer of a deep feedforward rectifier model. Let $h = g \circ f$ be the map of inputs to activations of the output units, given by composition of f with the linear output layer, $h(\mathbf{x}) = \mathbf{W}^{(\text{out})} f(\mathbf{x}) + \mathbf{b}^{(\text{out})}$. If the row span of $\mathbf{W}^{(\text{out})}$ is not orthogonal to any difference of gradients of neighbouring regions of linearity of f, then g captures all discontinuities of ∇f. In this case both functions f and h have the same number of regions of linearity.

If the number of regions of f is finite, then the number of differences of gradients is finite and there is a vector outside the union of their orthogonal spaces. Hence a matrix with a single row (a single output unit) suffices to capture all transitions between different regions of linearity of f. □

Lemma 3. *A layer of n rectifier units with n_0 inputs can compute any function that can be computed by the composition of a linear layer with n_0 inputs and n'_0 outputs and a rectifier layer with n'_0 inputs and n_1 outputs, for any $n_0, n'_0, n_1 \in \mathbb{N}$.*

Proof. A rectifier layer computes functions of the form $\mathbf{x} \mapsto \text{rect}(\mathbf{W}\mathbf{x} + \mathbf{b})$, with $\mathbf{W} \in \mathbb{R}^{n_1 \times n_0}$ and $\mathbf{b} \in \mathbb{R}^{n_1}$. The argument $\mathbf{W}\mathbf{x} + \mathbf{b}$ is an affine function of $\mathbf{x}$. The claim follows from the fact that any composition of affine functions is an affine function. □

3 One hidden layer

Let us look at the number of response regions of a single hidden layer MLP with n_0 input units and n hidden units. We first formulate the rectifier unit as follows:

$$\text{rect}(s) = \mathbb{I}(s) \cdot s, \tag{1}$$

where $\mathbb{I}$ is the indicator function defined as:

$$\mathbb{I}(s) = \begin{cases} 1, & \text{if } s > 0 \\ 0, & \text{otherwise} \end{cases} . \tag{2}$$

We can now write the single hidden layer MLP with n_y outputs as the function $f\colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_y}$;

$$f(\mathbf{x}) \quad = \mathbf{W}^{(\text{out})}\text{diag}\left(\begin{bmatrix} \mathbb{I}(\mathbf{W}^{(1)}_{1:}\mathbf{x} + \mathbf{b}^{(1)}_1) \\ \vdots \\ \mathbb{I}(\mathbf{W}^{(1)}_{n_1:}\mathbf{x} + \mathbf{b}^{(1)}_{n_1}) \end{bmatrix}\right)\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(\text{out})} \ . \tag{3}$$

From this formulation it is clear that each unit i in the hidden layer has two operational modes. One is when the unit takes value 0 and one when it takes a non-zero value. The boundary between these two operational modes is given by the hyperplane H_i consisting of all inputs $\mathbf{x} \in \mathbb{R}^{n_0}$ with $\mathbf{W}^{(1)}_{i,:}\mathbf{x} + \mathbf{b}^{(1)}_i = 0$. Below this hyperplane, the activation of the unit is constant equal to zero, and above, it is linear with gradient equal to $\mathbf{W}^{(1)}_{i,:}$. It follows that the number of regions of linearity of a single layer MLP is equal to the number of regions formed by the set of hyperplanes $\{H_i\}_{i\in[n_1]}$.

Not true according to above definition of region of linearity

A finite set of hyperplanes in a common n_0-dimensional Euclidian space is called an n_0-dimensional *hyperplane arrangement*. A *region* of an arrangement $\mathcal{A} = \{H_i \subset \mathbb{R}^{n_0}\}_{i\in[n]}$ is a connected component of the complement of the union of the hyperplanes, i.e., a connected component of $\mathbb{R}^{n_0} \setminus (\cup_{i\in[n]} H_i)$. To make this clearer, consider an arrangement $\mathcal{A}$ consisting of hyperplanes $H_i = \{\mathbf{x} \in \mathbb{R}^{n_0} \colon \mathbf{W}_{i,:}\mathbf{x} + \mathbf{b}_i = 0\}$ for all $i \in [n]$, for some $\mathbf{W} \in \mathbb{R}^{n\times n_0}$ and some $\mathbf{b} \in \mathbb{R}^n$. A region of $\mathcal{A}$ is a set of points of the form $R = \{\mathbf{x} \in \mathbb{R}^{n_0} \colon \operatorname{sgn}(\mathbf{W}\mathbf{x} + \mathbf{b}) = \mathbf{s}\}$ for some sign vector $\mathbf{s} \in \{-,+\}^n$.

A region of an arrangement is *relatively bounded* if its intersection with the space spanned by the normals of the hyperplanes is bounded. We denote by $r(\mathcal{A})$ the number of regions and by $b(\mathcal{A})$ the number of relatively bounded regions of an arrangement $\mathcal{A}$. The *essentialization* of an arrangement $\mathcal{A} = \{H_i\}_i$ is the arrangement consisting of the hyperplanes $H_i \cap \mathcal{N}$ for all i, defined in the span $\mathcal{N}$ of the normals of the hyperplanes H_i. For example, the essentialization of an arrangement of two non-parallel planes in $\mathbb{R}^3$ is an arrangement of two lines in a plane.

Problem 1. How many regions are generated by an arrangement of n hyperplanes in $\mathbb{R}^{n_0}$?

The general answer to Problem 1 is given by Zaslavsky's theorem (Zaslavsky, 1975, Theorem A), which is one of the central results from the theory of hyperplane arrangements.

We will only need the special case of hyperplanes in *general position*, which realize the maximal possible number of regions. Formally, an n-dimensional arrangement $\mathcal{A}$ is in general position if for any subset $\{H_1, \dots H_p\} \subseteq \mathcal{A}$ the following holds. (1) If $p \leq n$, then $\dim(H_1 \cap \dots \cap H_p) = n - p$. (2) If $p > n$, then $H_1 \cap \dots \cap H_p = \emptyset$. An arrangement is in general position if the weights $\mathbf{W}$, $\mathbf{b}$ defining its hyperplanes are generic. This means that any arrangement can be perturbed by an arbitrarily small perturbation in such a way that the resulting arrangement is in general position.

For arrangements in general position, Zaslavsky's theorem can be stated in the following way (see Stanley, 2004, Proposition 2.4).

Proposition 1. *Let $\mathcal{A}$ be an arrangement of m hyperplanes in general position in $\mathbb{R}^{n_0}$. Then*

$$\begin{aligned} r(\mathcal{A}) &= \sum_{s=0}^{n_0} \binom{m}{s} \\ b(\mathcal{A}) &= \binom{m-1}{n_0}. \end{aligned}$$

In particular, the number of regions of a 2-dimensional arrangement $\mathcal{A}_m$ of m lines in general position is equal to

$$r(\mathcal{A}_m) = \binom{m}{2} + m + 1. \tag{4}$$

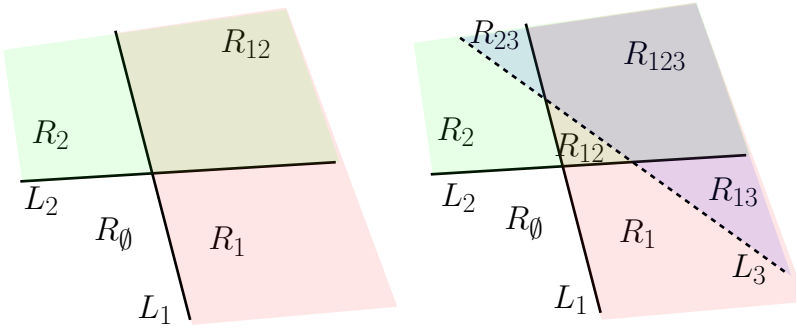


Figure 2: Induction step of the hyperplane sweep method for counting the regions of line arrangements in the plane.

For the purpose of illustration, we sketch a proof of eq. (4) using the *sweep hyperplane* method. We proceed by induction over the number of lines m.

Base case $m = 0$. It is obvious that in this case there is a single region, corresponding to the entire plane. Therefore, $r(\mathcal{A}_0) = 1$.

Induction step. Assume that for m lines the number of regions is $r(\mathcal{A}_m) = \binom{m}{2} + m + 1$, and add a new line L_{m+1} to the arrangement. Since we assumed the lines are in general position, L_{m+1} intersects each of the existing lines L_k at a different point. Fig. 2 depicts the situation for $m = 2$.

The m intersection points split the line L_{m+1} into $m + 1$ segments. Each of these segments cuts a region of $\mathcal{A}_m$ in two pieces. Therefore, by adding the line L_{m+1} we get $m + 1$ new regions. In Fig. 2 the two intersection points result in three segments that split each of the regions R_1, R_{01}, R_0 in two. Hence

$$\begin{aligned} r(\mathcal{A}_{m+1}) = r(\mathcal{A}_m) + m + 1 &= \frac{m(m-1)}{2} + m + 1 + m + 1 = \frac{m(m+1)}{2} + (m+1) + 1 \\ &= \binom{m+1}{2} + (m+1) + 1. \end{aligned}$$

For the number of response regions of MLPs with one single hidden layer we obtain the following.

Proposition 2. *The regions of linearity of a function in the model $\mathcal{F}_{(n_0,n_1,1)}$ with n_0 inputs and n_1 hidden units are given by the regions of an arrangement of n_1 hyperplanes in n_0-dimensional space. The maximal number of regions of such an arrangement is $\mathcal{R}(n_0, n_1, n_y) = \sum_{j=0}^{n_0} \binom{n_1}{j}$.*

Proof. This is a consequence of Lemma 1. The maximal number of regions is produced by an n_0-dimensional arrangement of n_1 hyperplanes in general position, which is given in Proposition 1. □

4 Multiple hidden layers

In order to show that a k hidden layer model can be more expressive than a single hidden layer one with the same number of hidden units, we will need the next three propositions.

Proposition 3. *Any arrangement can be scaled down and shifted such that all regions of the arrangement intersect the unit ball.*

Proof. Let $\mathcal{A}$ be an arrangement and let $\mathcal{S}$ be a ball of radius r and center $\mathbf{c}$. Let d be the supremum of the distance from the origin to a point in a bounded region of the essentialization of the arrangement $\mathcal{A}$. Consider the map $\phi : \mathbb{R}^{n_0} \to \mathbb{R}^{n_0}$ defined by $\phi(\mathbf{x}) = \frac{r}{2d} \cdot \mathbf{x} + \mathbf{c}$. Then $\mathcal{A}' = \phi(\mathcal{A})$ is an arrangement satisfying the claim. It is easy to see that any point with norm bounded by d is mapped to a point inside the ball S. □

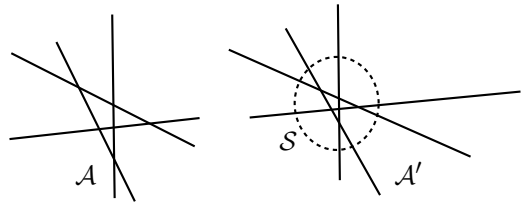


Figure 3: An arrangement $\mathcal{A}$ and a scaled-shifted version $\mathcal{A}'$ whose regions intersect the ball $\mathcal{S}$.

The proposition is illustrated in Fig. 3.

We need some additional notations in order to formulate the next proposition. Given a hyperplane $H = \{\mathbf{x}\colon \mathbf{w}^\top\mathbf{x} + b = 0\}$, we consider the region $H^- = \{\mathbf{x}\colon \mathbf{w}^\top\mathbf{x} + b < 0\}$, and the region $H^+ = \{\mathbf{x}\colon \mathbf{w}^\top\mathbf{x} + b \geq 0\}$. If we think about the corresponding rectifier unit, then H^+ is the region where the unit is active and H^- is the region where the unit is dead.

Let R be a region delimited by the hyperplanes $\{H_1, \ldots, H_n\}$. We denote by $R^+ \subseteq \{1, \ldots, n\}$ the set of all hyperplane-indices j with $R \subset H_j^+$. In other words, R^+ is the list of hidden units that are active (non-zero) in the input-space region R.

The following proposition describes the combinatorics of 2-dimensional arrangements in general position. More precisely, the proposition describes the combinatorics of n-dimensional arrangements with 2-dimensional essentialization in general position. Recall that the essentialization of an arrangement is the arrangement that it defines in the subspace spanned by the normals of its hyperplanes.

The proposition guarantees the existence of input weights and bias for a rectifier layer such that for any list of consecutive units, there is a region of inputs for which exactly the units from that list are active.

Proposition 4. *For any $n_0, n \in \mathbb{N}$, $n \geq 2$, there exists an n_0-dimensional arrangement $\mathcal{A}$ of n hyperplanes such that for any pair $a, b \in \{1, \ldots, n\}$ with $a < b$, there is a region R of $\mathcal{A}$ with $R^+ = \{a, a+1, \ldots, b\}$.*

We show that the hyperplanes of a 2-dimensional arrangement in general position can be indexed in such a way that the claim of the proposition holds. For higher dimensional arrangements the statement follows trivially, applying the 2-dimensional statement to the intersection of the arrangement with a 2-subspace.

Proof of Proposition 4. Consider first the case $n_0 = 2$. We define the first line L_1 of the arrangement to be the x-axis of the standard coordinate system. To define the second line L_2, we consider a circle $\mathcal{S}_1$ of radius $r \in \mathbb{R}_+$ centered at the origin. We define L_2 to be the tangent of $\mathcal{S}_1$ at an angle α_1 to the y-axis, where $0 < \alpha_1 < \frac{\pi}{2}$. The top left panel of Fig. 4 depicts the situation. In the figure, $R_\emptyset$ corresponds to inputs for which no rectifier unit is active, R_1 corresponds to inputs where the first unit is active, R_2 to inputs where the second unit is active, and R_{12} to inputs where both units are active. This arrangement has the claimed properties.

Now assume that there is an arrangement of n lines with the claimed properties. To add an $(n+1)$-th line, we first consider the maximal distance d_{max} from the origin to the intersection of two lines $L_i \cap L_j$ with $1 \leq i < j < n$. We also consider the radius-$(d_{\text{max}} + r)$ circle $\mathcal{S}_n$ centered at the origin. The circle $\mathcal{S}_n$ contains all intersection of any of the first n lines. We now choose an angle α_n with $0 < \alpha_n < \alpha_{n-1}$ and define L_{n+1} as the tangent of $\mathcal{S}_n$ that forms an angle α_n with the y-axis. Fig. 4 depicts adding the third and fourth line to the arrangement.

After adding line L_{n+1}, we have that the arrangement

1. is in general position.
2. has regions $R'_1, \ldots, R'_{n+1}$ with $R'^+_i = \{i, i+1, \ldots, n+1\}$ for all $i \in [n+1]$.

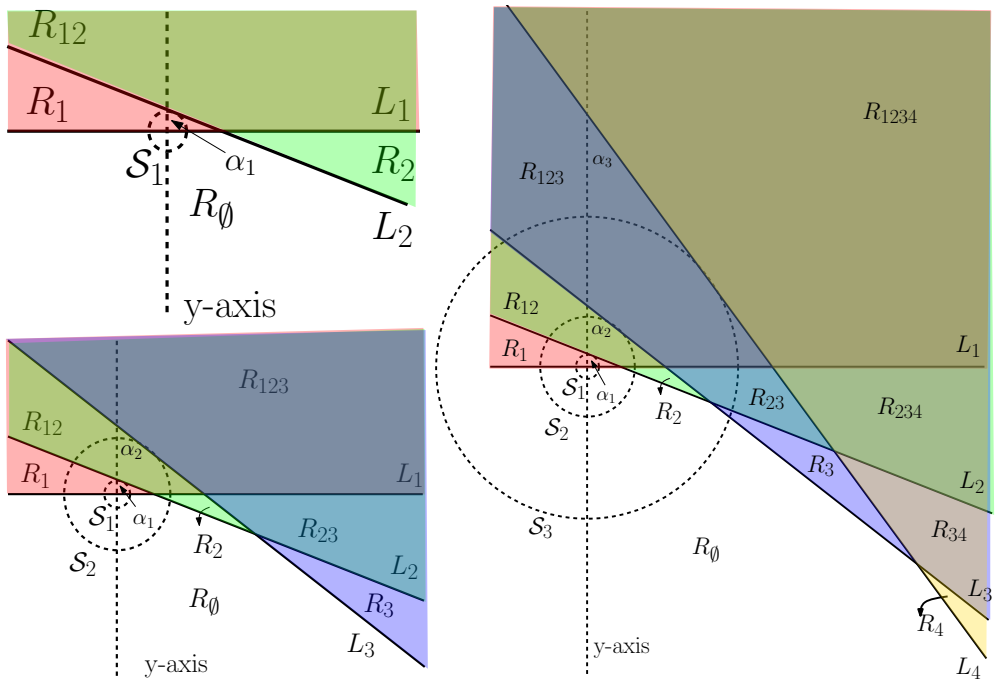


Figure 4: Illustration of the hyperplane arrangement discussed in Proposition 4, in the 2-dimensional case. On the left we have arrangements of two and three lines, and on the right an arrangement of four lines.

The regions of the arrangement are stable under perturbation of the angles and radii used to define the lines. Any slight perturbation of these parameters preserves the list of regions. Therefore, the arrangement is in general position.

The second property comes from the order in which L_{n+1} intersects all previous lines. L_{n+1} intersects the lines in the order in which they were added to the arrangement: $L_1, L_2, \ldots, L_n$. The intersection of L_{n+1} and L_i, $B_{in+1} = L_{n+1} \cap L_i$, is above the lines $L_{i+1}, L_{i+2}, \ldots, L_n$, and hence the segment $B_{i-1n+1}B_{in+1}$ between the intersection with L_{i-1} and with L_i, has to cut the region in which only units i to n are active.

The intersection order is ensured by the choice of angles α_i and the fact that the lines are tangent to the circles $\mathcal{S}_i$. For any $i < j$ and $B_{ij} = L_i \cap L_j$ let T_{ij} be the line parallel to the y-axis passing through B_{ij}. Each line T_{ij} divides the space in two. Let H_{ij} be the half-space to the right of T_{ij}. Within any half-space H_{ij}, the intersection $H_{ij} \cap L_i$ is above $H_{ij} \cap L_j$, because the angle α_{i-1} of L_i with the y-axis is larger than α_{j-1} (this means L_j has a stepper decrease). Since L_{n+1} is tangent to the circle that contains all points B_{ij}, the line L_{n+1} will intersect lines L_i and L_j in H_{ij}, and therefore it has to intersect L_i first.

For $n_0 > 2$ we can consider an arrangement that is essentially 2-dimensional and has the properties of the arrangement described above. To do this, we construct a 2-dimensional arrangement in a 2-subspace of $\mathbb{R}^{n_0}$ and then extend each of the lines L_i of the arrangement to a hyperplane H_i that crosses L_i orthogonally. The resulting arrangement satisfies all claims of the proposition. □

The next proposition guarantees the existence of a collection of affine maps with shared bias, which map a collection of regions to a common output.

Proposition 5. *Consider two integers n_0 and p. Let $\mathcal{S}$ denote the n_0-dimensional unit ball and let $R_1, \ldots, R_p \subseteq \mathbb{R}^{n_0}$ be some regions with non-empty interiors. Then there is a choice of weights $\mathbf{c} \in \mathbb{R}^{n_0}$ and $\mathbf{U}_1, \ldots, \mathbf{U}_p \in \mathbb{R}^{n_0 \times n_0}$ for which $g_i(R_i) \supseteq \mathcal{S}$ for all $i \in [p]$, where $g_i \colon \mathbb{R}^{n_0} \to \mathbb{R}^{n_0}$; $\mathbf{y} \mapsto \mathbf{U}_i\mathbf{y} + \mathbf{c}$.*

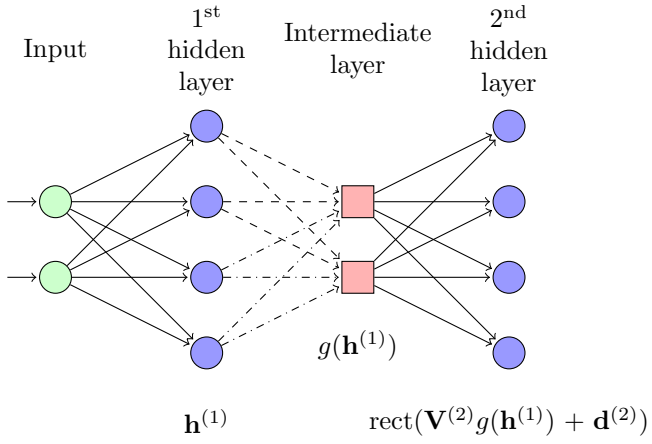


Figure 5: Illustration of Example 1. The units represented by squares build an intermediary layer of linear units between the first and the second hidden layers. The computation of such an intermediary linear layer can be absorbed in the second hidden layer of rectifier units (Lemma 3). The connectivity map depicts the maps g_1 by dashed arrows and g_2 by dashed-dotted arrows.

Proof. To see this, consider the following construction. For each region R_i consider a ball $\mathcal{S}_i \subseteq R_i$ of radius $r_i \in \mathbb{R}_+$ and center $\mathbf{s}_i = (\mathbf{s}_{i1}, \ldots, \mathbf{s}_{in_0}) \in \mathbb{R}^{n_0}$. For each $j = 1, \ldots, n_0$, consider p positive numbers $u_{1j}, \ldots, u_{pj}$ such that $u_{ij}\mathbf{s}_{ij} = u_{kj}\mathbf{s}_{kj}$ for all $1 \leq k < i \leq p$. This can be done fixing u_{1j} equal to 1 and solving the equation for all other numbers. Let $\eta \in \mathbb{R}$ be such that $r_i \eta u_{ij} > 1$ for any j and i. Scaling each region R_i by $\mathbf{U}_i = \mathrm{diag}(\eta u_{i0}, \ldots, \eta u_{in_0})$ transforms the center of $\mathcal{S}_i$ to the same point for all i. By the choice of η, the minor radius of all transformed balls is larger than 1.

We can now set $\mathbf{c}$ to be minus the common center of the scaled balls, to obtain the map:

$$g_i(\mathbf{x}) = \mathrm{diag}\left(\eta u_{i1}, \ldots, \eta u_{in_0}\right)\mathbf{x} - \mathrm{diag}\left(\eta u_{11}, \ldots, \eta u_{1n_0}\right)\mathbf{s}_1, \qquad \text{for all } 1 \leq i \leq p.$$

These g_i satisfy claimed property, namely that $g_i(R_i)$ contains the unit ball, for all i. □

Before proceeding, we discuss an example illustrating how the previous propositions and lemmas are put together to prove our main result below, in Theorem 1.

Example 1. Consider a rectifier MLP with $n_0 = 2$, such that the input space is $\mathbb{R}^2$, and assume that the network has only two hidden layers, each consisting of $n = 2n'$ units. Each unit in the first hidden layer defines a hyperplane in $\mathbb{R}^2$, namely the hyperplane that separates the inputs for which it is active, from the inputs for which it is not active. Hence the first hidden layer defines an arrangement of n hyperplanes in $\mathbb{R}^2$. By Proposition 4, this arrangement can be made such that it delimits regions of inputs $R_1, \ldots, R_{n'} \subseteq \mathbb{R}^2$ with the following property. For each input in any given one of these regions, exactly one pair of units in the first hidden layer is active, and, furthermore, the pairs of units that are active on different regions are disjoint.

By the definition of rectifier units, each hidden unit computes a linear function within the half-space of inputs where it is active. In turn, the image of R_i by the pair of units that is active in R_i is a polyhedron in $\mathbb{R}^2$. For each region R_i, denote corresponding polyhedron by S_i.

Recall that a rectifier layer computes a map of the form $f\colon \mathbb{R}^n \to \mathbb{R}^m;\ \mathbf{x} \mapsto \mathrm{rect}(\mathbf{W}\mathbf{x} + \mathbf{b})$. Hence a rectifier layer with n inputs and m outputs can compute any composition $f' \circ g$ of an affine map $g\colon \mathbb{R}^n \to \mathbb{R}^k$ and a map f' computed by a rectifier layer with k inputs and m outputs (Lemma 3).

Consider the map computed by the rectifier units in the second hidden layer, i.e., the map that takes activations from the first hidden layer and outputs activations from the second hidden layer. We think of this map as a composition $f' \circ g$ of an affine map $g\colon \mathbb{R}^n \to \mathbb{R}^2$ and a map f' computed by a rectifier layer with 2 inputs. The map g can be interpreted as an intermediary layer consisting of two linear units, as illustrated in Fig. 5.

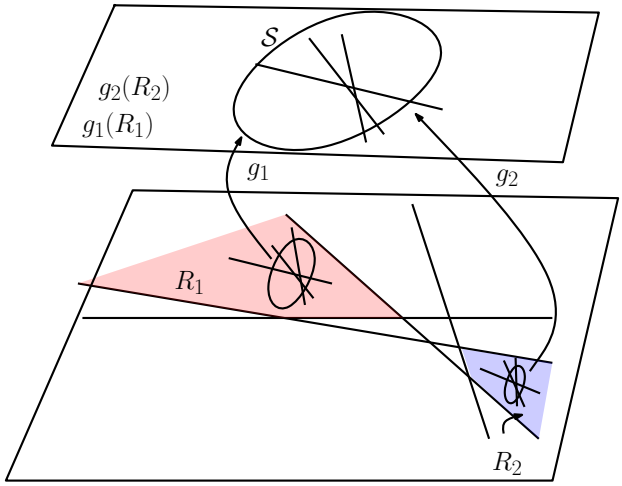


Figure 6: Constructing $\left\lfloor \frac{n_1}{n_0} \right\rfloor \sum_{k=0}^{n_0} \binom{n_2}{k}$ response regions in a model with two layers.

Within each input region R_i, only two units in the first hidden layer are active. Therefore, for each input region R_i, the output of the intermediary layer is an affine transformation of S_i. Furthermore, the weights of the intermediary layer can be chosen in such a way that the image of each R_i contains the unit ball.

Now, f' is the map computed by a rectifier layer with 2 inputs and n outputs. It is possible to define this map in such a way that it has $\mathcal{R}$ regions of linearity within the unit ball, where $\mathcal{R}$ is the number of regions of a 2-dimensional arrangement of n hyperplanes in general position.

We see that the entire network computes a function which has $\mathcal{R}$ regions of linearity within each one of the input regions $R_1, \ldots, R_{n'}$. Each input region R_i is mapped by the concatenation of first and intermediate (notional) layer to a subset of $\mathbb{R}^2$ which contains the unit ball. Then, the second layer computes a function which partitions the unit ball into many pieces. The partition computed by the second layer gets replicated in each of the input regions R_i, resulting in a subdivision of the input space in exponentially many pieces (exponential in the number of network layers).

Now we are ready to state our main result on the number of response regions of rectifier deep feedforward networks:

Theorem 1. *A model with n_0 inputs and k hidden layers of widths $n_1, n_2, \ldots, n_k$ can divide the input space in $\left(\prod_{i=1}^{k-1} \left\lfloor \frac{n_i}{n_0} \right\rfloor\right) \sum_{i=0}^{n_0} \binom{n_k}{i}$ or possibly more regions.*

Proof of Theorem 1. Let the first hidden layer define an arrangement like the one from Proposition 4. Then there are $p = \left\lfloor \frac{n_1}{n_0} \right\rfloor$ input-space regions $R_i \subseteq \mathbb{R}^{n_0}$, $i \in [p]$ with the following property. For each input vector from the region R_i, exactly n_0 units from the first hidden layer are active. We denote this set of units by I_i. Furthermore, by Proposition 4, for inputs in distinct regions R_i, the corresponding set of active units is disjoint; that is, $I_i \cap I_j = \emptyset$ for all $i, j \in [p]$, $i \neq j$.

To be more specific, for an input vectors from R_1, exactly the first n_0 units of the first hidden layer are active, that is, for these input vectors the value of $\mathbf{h}_j^{(1)}$ is non-zero if and only if $j \in I_1 = \{1, \ldots, n_0\}$. For input vectors from R_2, only the next n_0 units of the first hidden layer are active, that is, the units with index in $I_2 = \{n_0 + 1, \ldots, 2n_0\}$, and so on.

Now we consider a 'fictitious' intermediary layer consisting of n_0 linear units between the first and second hidden layers. As this intermediary layer computes an affine function, it can be absorbed into the second hidden layer (see Lemma 3). We use it only for making the next arguments clearer.

The map taking activations from the first hidden layer to activations from the second hidden layer is $\mathrm{rect}(\mathbf{W}^{(2)}\mathbf{x} + \mathbf{b}^{(2)})$, where $\mathbf{W}^{(2)} \in \mathbb{R}^{n_2 \times n_1}$, $\mathbf{b}^{(2)} \in \mathbb{R}^{n_2}$.

We can write the input and bias weight matrices as $\mathbf{W}^{(2)} = \mathbf{U}^{(2)}\mathbf{V}^{(2)}$ and $\mathbf{b}^{(2)} = \mathbf{d}^{(2)} + \mathbf{V}^{(2)}\mathbf{c}^{(2)}$, where $\mathbf{U}^{(2)} \in \mathbb{R}^{n_0 \times n_1}$, $\mathbf{c} \in \mathbb{R}^{n_0}$, and $\mathbf{V}^{(2)} \in \mathbb{R}^{n_2 \times n_0}$, $\mathbf{d} \in \mathbb{R}^{n_2}$.

The weights $\mathbf{U}^{(2)}$ and $\mathbf{c}^{(2)}$ describe the affine function computed by the intermediary layer, $\mathbf{x} \mapsto \mathbf{U}^{(2)}\mathbf{x} + \mathbf{c}$. The weights $\mathbf{V}^{(2)}$ and $\mathbf{d}^{(2)}$ are the input and bias weights of the rectifier layer following the intermediary layer.

We now consider the sub-matrix $\mathbf{U}_i^{(2)}$ of $\mathbf{U}^{(2)}$ consisting of the columns of $\mathbf{U}^{(2)}$ with indices I_i, for all $i \in [p]$. Then $\mathbf{U}^{(2)} = \left[\mathbf{U}_1^{(2)} | \cdots | \mathbf{U}_p^{(2)} | \tilde{\mathbf{U}}^{(2)}\right]$, where $\tilde{\mathbf{U}}^{(2)}$ is the sub-matrix of $\mathbf{U}^{(2)}$ consisting of its last $n_1 - pn_0$ columns. In the sequel we set all entries of $\tilde{\mathbf{U}}^{(2)}$ equal to zero.

The map $g : \mathbb{R}^{n_1} \to \mathbb{R}^{n_0}$; $g(\mathbf{x}) = \mathbf{U}^{(2)}\mathbf{x} + \mathbf{c}^{(2)}$ is thus written as the sum $g = \sum_{i \in [p]} g_i + \mathbf{c}^{(2)}$, where $g_i : \mathbb{R}^{n_0} \to \mathbb{R}^{n_0}$; $g_i(\mathbf{x}) = \mathbf{U}_i^{(2)}\mathbf{x}$, for all $i \in [p]$.

Let S_i be the image of the input-space region R_i by the first hidden layer. By Proposition 5, there is a choice of the weights $\mathbf{U}_i^{(2)}$ and bias $\mathbf{c}^{(2)}$ such that the image of S_i by $\mathbf{x} \to \mathbf{U}_i^{(2)}(\mathbf{x}) + \mathbf{c}^{(2)}$ contains the n_0-dimensional unit ball. Now, for all inputs vectors from R_i, only the units I_i of the first hidden layer are active. Therefore, $g|_{R_i} = g_i|_{R_i} + \mathbf{c}^{(2)}$. This implies that the image $g(R_i)$ of the input-space region R_i by the intermediary layer contains the unit ball, for all $i \in [p]$.

We can now choose $\mathbf{V}^{(2)}$ and $\mathbf{d}^{(2)}$ in such a way that the rectifier function $\mathbb{R}^{n_0} \to \mathbb{R}^{n_2}$; $\mathbf{y} \mapsto \mathrm{rect}(\mathbf{V}^{(2)}\mathbf{y} + \mathbf{d}^{(2)})$ defines an arrangement $\mathcal{A}$ of n_2 hyperplanes with the property that each region of $\mathcal{A}$ intersects the unit ball at an open neighborhood.

In consequence, the map from input-space to activations of the second hidden layer has $r(\mathcal{A})$ regions of linearity within each input-space region R_i. Fig. 6 illustrates the situation. All inputs that are mapped to the same activation of the first hidden layer, are treated as equivalent on the subsequent layers. In this sense, an arrangement $\mathcal{A}$ defined on the set of common outputs of $R_1, \ldots, R_p$ at the first hidden layer, is 'replicated' in each input region $R_1, \ldots, R_p$.

The subsequent layers of the network can be analyzed in a similar way as done above for the first two layers. In particular, the weights $\mathbf{V}^{(2)}$ and $\mathbf{d}^{(2)}$ can be chosen in such a way that they define an arrangement with the properties from Proposition 4. Then, the map taking activations from the second hidden layer to activations from the third hidden layer, can be analyzed by considering again a fictitious intermediary layer between the second and third layers, and so forth, as done above.

For the last hidden layer we choose the input weights $\mathbf{V}^{(k)}$ and bias $\mathbf{d}^{(k)}$ defining an n_0-dimensional arrangement of n_k hyperplanes in general position. The map of inputs to activations of the last hidden layer has thus $\left(\prod_{i=1}^{k-1} \left\lfloor \frac{n_i}{n_0} \right\rfloor\right) \sum_{i=0}^{n_0} \binom{n_k}{i}$ regions of linearity. This number is a lower bound on the maximal number of regions of linearity of functions computable by the network. This completes the proof. The intuition of the construction is illustrated in Fig. 7. □

In the Appendix A we derive an asymptotic expansion of the bound given in Theorem 1.

5 A special class of deep models

In this section we consider deep rectifier models with n_0 input units and hidden layers of width $n = 2n_0$. This restriction allows us to construct a very efficient deep model in terms of number of response regions. The analysis that we provide in this section complements the results from the previous section, showing that rectifier MLPs can compute functions with many response regions, even when defined with relatively few hidden layers.

Example 2. Let us assume we have a 2-dimensional input, i.e., $n_0 = 2$, and a layer of $n = 4$ rectifiers f_1, f_2, f_3, and f_4, followed by a linear projection. We construct the rectifier layer in such a way that it divides the input space into four 'square' cones; each of them corresponding to the inputs

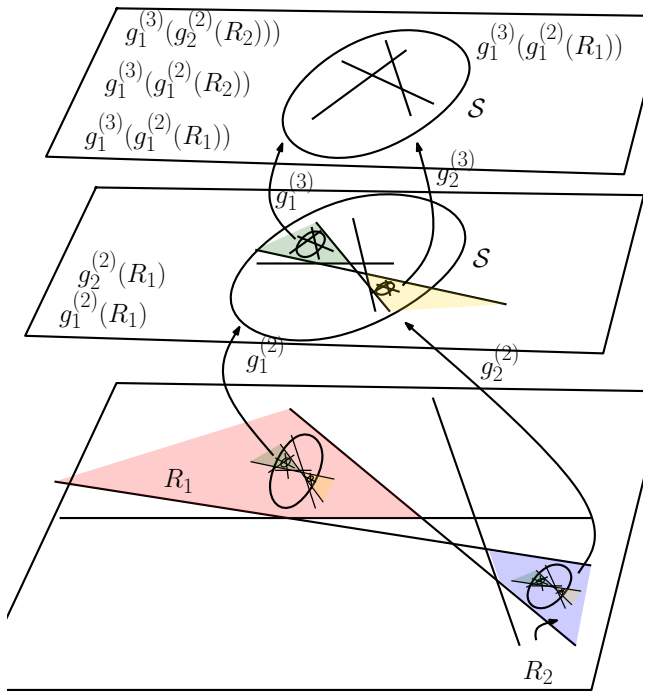


Figure 7: Constructing $\left\lfloor \frac{n_2}{n_0} \right\rfloor \left\lfloor \frac{n_1}{n_0} \right\rfloor \sum_{k=0}^{n_0} \binom{n_3}{k}$ response regions in a model with three layers.

where two of the rectifier units are active. We define the four rectifiers as:

$$
\begin{aligned}
f_1(\mathbf{x}) &= \max\left\{0, [1,0]^\top \mathbf{x}\right\}, \\
f_2(\mathbf{x}) &= \max\left\{0, [-1,0]^\top \mathbf{x}\right\}, \\
f_3(\mathbf{x}) &= \max\left\{0, [0,1]^\top \mathbf{x}\right\}, \\
f_4(\mathbf{x}) &= \max\left\{0, [0,-1]^\top \mathbf{x}\right\},
\end{aligned}
$$

where $\mathbf{x} = [x_1, x_2]^\top \in \mathbb{R}^{n_0}$. By adding pairs of coordinates of $\mathbf{f} = [f_1, f_2, f_3, f_4]^\top$, we can effectively mimic a layer consisting of two absolute-value units g_1 and g_2:

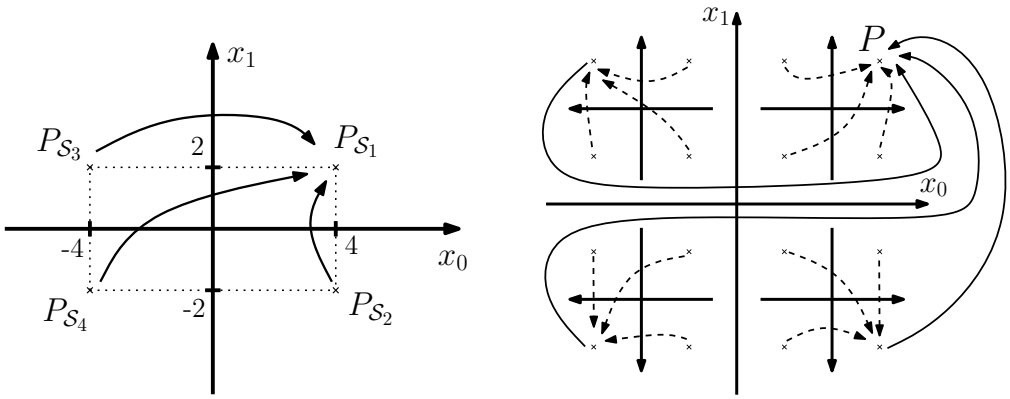
$$
\begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ f_3(\mathbf{x}) \\ f_4(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \text{abs}(x_1) \\ \text{abs}(x_2) \end{bmatrix}. \tag{5}
$$

The absolute-value unit g_i divides the input space along the i-th coordinate axis, taking values which are symmetric about that axis. The combination of g_1 and g_2 is then a function with four regions of linearity;

$$
\begin{aligned}
\mathcal{S}_1 &= \{(x_1, x_2) \mid x_1 \geq 0, x_2 \geq 0\} \\
\mathcal{S}_2 &= \{(x_1, x_2) \mid x_1 \geq 0, x_2 < 0\} \\
\mathcal{S}_3 &= \{(x_1, x_2) \mid x_1 < 0, x_2 \geq 0\} \\
\mathcal{S}_4 &= \{(x_1, x_2) \mid x_1 < 0, x_2 < 0\}.
\end{aligned}
$$

Since the values of g_i are symmetric about the i-th coordinate axis, each point $\mathbf{x} \in \mathcal{S}_i$ has a corresponding point $\mathbf{y} \in \mathcal{S}_j$ with $\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{y})$, for all i and j.

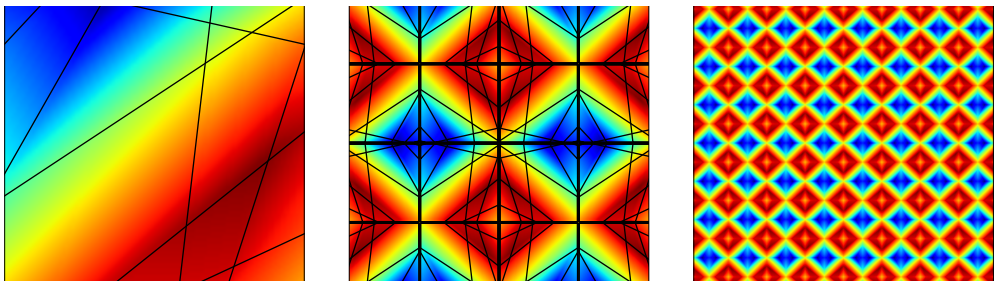
We can apply the same procedure to the image of $[g_1, g_2]$ to recursively divide the input space, as illustrated in Fig. 8. For instance, if we apply this procedure one more time, we get four regions



(a) (b)

Figure 8: Illustration of Example 2. (a) A rectifier layer with two pairs of units, where each pair computes the absolute value of one of two input coordinates. Each input quadrant is mapped to the positive quadrant. (b) Depiction of a two layer model. Both layers simulate the absolute value of their input coordinates.

within each $\mathcal{S}_i$, resulting in 16 regions in total, within the input space. On the last layer, we may place rectifiers in any way suitable for the task of interest (e.g., classification). The partition computed by the last layer will be copied to each of the input space regions that produced the same input for the last layer. Fig. 9 shows a function that can be implemented efficiently by a deep model using the previous observations.



(a) (b) (c)

Figure 9: (a) Illustration of the partition computed by 8 rectifier units on the outputs (x_1, x_2) of the preceding layer. The color is a heat map of $x_1 - x_2$. (b) Heat map of a function computed by a rectifier network with 2 inputs, 2 hidden layers of width 4, and one linear output unit. The black lines delimit the regions of linearity of the function. (c) Heat map of a function computed by a 4 layer model with a total of 24 hidden units. It takes at least 137 hidden units on a shallow model to represent the same function.

The foregoing discussion can be easily generalized to $n_0 > 2$ input variables and k hidden layers, each consisting of $2n_0$ rectifiers. In that case, the maximal number of linear regions of functions computable by the network is lower-bounded as follows.

Theorem 2. *The maximal number of regions of linearity of functions computable by a rectifier neural network with n_0 input variables and k hidden layers of width $2n_0$ is at least* $2^{(k-1)n_0} \sum_{j=0}^{n_0} \binom{2n_0}{j}$.

Proof. We prove this constructively. We define the rectifier units in each hidden layer in pairs, with the sum of each pair giving the absolute value of a coordinate axis. We interpret also the sum of such pairs as the actual input coordinates of the subsequent hidden layers. The rectifiers in the first hidden layer are defined in pairs, such that the sum of each pair is the absolute value of one of the input dimensions, with bias equal to $(-\frac{1}{2}, \ldots, -\frac{1}{2})$. In the next hidden layers, the rectifiers are defined

in a similar way, with the difference that each pair computes the absolute value of the sum of two of their inputs. The last hidden layer is defined in such a way that it computes a piece-wise linear function with the maximal number of pieces, all of them intersecting the unit cube in $\mathbb{R}^{n_0}$. The maximal number of regions of linearity of m rectifier units with n_0-dimensional input is $\sum_{j=0}^{n_0} \binom{m}{j}$. This partition is multiplied in each previous layer 2^{n_0} times. $\square$

The theorem shows that even for a small number of layers k, we can have many more linear regions in a deep model than in a shallow one. For example, if we set the input dimensionality to $n_0 = 2$, a shallow model with $4n_0$ units will have at most 37 linear regions. The equivalent deep model with two layers of $2n_0$ units can produce 44 linear regions. For $6n_0$ hidden units the shallow model computes at most 79 regions, while the equivalent three layer model can compute 176 regions.

6 Discussion and conclusions

In this paper we introduced a novel way of understanding the expressiveness of neural networks with piecewise linear activations. We count the number of regions of linearity, also called response regions, of the functions that they can represent. The number of response regions tells us how well the models can approximate arbitrary curved shapes. Computational Geometry provides us the tool to make such statements.

We found that deep and narrow rectifier MPLs can generate many more regions of linearity than their shallow counterparts with the same number of computational units or of parameters. We can express this in terms of the ratio between the maximal number of response regions and the number of parameters of both model classes. For a deep model with $n_0 = O(1)$ inputs and k hidden layers of width n, the maximal number of response regions per parameter behaves as

$$\Omega\left(\left\lfloor \frac{n}{n_0} \right\rfloor^{k-1} \frac{n^{n_0-2}}{k}\right).$$

For a shallow model with $n_0 = O(1)$ inputs, the maximal number of response regions per parameter behaves as

$$O\left(k^{n_0-1} n^{n_0-1}\right).$$

We see that the deep model can generate many more response regions per parameter than the shallow model; exponentially more regions per parameter in terms of the number of hidden layers k, and at least order $(k-2)$ polynomially more regions per parameter in terms of the layer width n. In particular, there are deep models which use fewer parameters to produce more linear regions than their shallow counterparts. Details about the asymptotic expansions are given in the Appendix A.

In this paper we only considered linear output units, but this is not a restriction, as the output activation itself is not parametrized. If there is a target function f_{targ} that we want to model with a rectifier MLP with σ as its output activation function, then there exists a function f'_{targ} such that $\sigma(f'_{\text{targ}}) = f_{\text{targ}}$, when σ has an inverse (e.g., with sigmoid), $f'_{\text{targ}} = \sigma^{-1}(f_{\text{targ}})$. For activations that do not have an inverse, like softmax, there are infinitely many functions f'_{targ} that work. We just need to pick one, e.g., for softmax we can pick $\log(f_{\text{targ}})$. By analyzing how well we can model f'_{targ} with a linear output rectifier MLP we get an indirect measure of how well we can model f_{targ} with an MLP that has σ as its output activation.

Another interesting observation is that we recover a high ratio of n to n_0 if the data lives near a low-dimensional manifold (effectively like reducing the input size n_0). One-layer models can reach the upper bound of response regions only by spanning all the dimensions of the input. In other words, shallow models are not capable of concentrating linear response regions in any lower dimensional subspace of the input. If, as commonly assumed, data lives near a low dimensional manifold, then we care only about the number of response regions that a model can generate in the directions of the data manifold. One way of thinking about this is principal component analysis (PCA), where one finds that only few input space directions (say on the MNIST database) are relevant to the underlying data. In such a situation, one cares about the number of response regions that a model can generate only within the directions in which the data does change. In such situations $n \gg n_0$, and our results show a clear advantage of using deep models.

We believe that the proposed framework can be used to answer many other interesting questions about these models. For example, one can look at how the number of response regions is affected by different constraints of the model, like shared weights. We think that this approach can also be used to study other kinds of piecewise linear models, such as convolutional networks with rectifier units or maxout networks, or also for comparing between different piecewise linear models.

A Asymptotic

Here we derive asymptotic expressions of the formulas contained in Proposition 2 and Theorem 1. We use following standard notation:

- $f(n) = O(g(n))$ means that there is a positive constant c_2 such that $f(n) \leq c_2 g(n)$ for all n larger than some N.
- $f(n) = \Theta(g(n))$ means that there are two positive constants c_1 and c_2 such that $c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all n larger than some N.
- $f(n) = \Omega(g(n))$ means that there is a positive constant c_1 such that $f(n) \geq c_1 g(n)$ for all n larger than some N.

Proposition 6.

- *Consider a single layer rectified MLP with kn units and n_0 inputs. Then the maximal number of regions of linearity of the functions represented by this network is*

$$\mathcal{R}(n_0, kn, 1) = \sum_{s=0}^{n_0} \binom{kn}{s},$$

and

$$\mathcal{R}(n_0, kn, 1) = O(k^{n_0} n^{n_0}), \quad \textit{when } n_0 = O(1).$$

- *Consider a k layer rectified MLP with hidden layers of width n and n_0 inputs. Then the maximal number of regions of linearity of the functions represented by this network satisfies*

$$\mathcal{R}(n_0, n, \ldots, n, 1) \geq \left(\prod_{i=1}^{k-1} \left\lfloor \frac{n}{n_0} \right\rfloor \right) \sum_{s=0}^{n_0} \binom{n}{s},$$

and

$$\mathcal{R}(n_0, n, \ldots, n, 1) = \Omega\left(\left\lfloor \frac{n}{n_0} \right\rfloor^{k-1} n^{n_0} \right), \quad \textit{when } n_0 = O(1).$$

Proof. Here only the asymptotic expressions remain to be shown. It is known that

$$\sum_{s=0}^{n_0} \binom{m}{s} = \Theta\left(\left(1 - \frac{2n_0}{m}\right)^{-1} \binom{m}{n_0} \right), \quad \text{when } n_0 \leq \frac{m}{2} - \sqrt{m}. \tag{6}$$

Furthermore, it is known that

$$\binom{m}{s} = \frac{m^s}{s!} \left(1 + O(\tfrac{1}{m})\right), \quad \text{when } s = O(1). \tag{7}$$

When n_0 is constant, $n_0 = O(1)$, we have that

$$\binom{kn}{n_0} = \frac{k^{n_0}}{n_0!} n^{n_0} \left(1 + O\left(\tfrac{1}{kn}\right)\right).$$

In this case, it follows that

$$\sum_{s=0}^{n_0} \binom{kn}{s} = \Theta\left(\left(1 - \frac{2n_0}{kn}\right)^{-1} \binom{kn}{n_0} \right) = \Theta\left(k^{n_0} n^{n_0}\right) \quad \text{and also} \quad \sum_{s=0}^{n_0} \binom{n}{s} = \Theta(n^{n_0}).$$

Furthermore,

$$\left(\prod_{i=1}^{k-1}\left\lfloor\frac{n}{n_0}\right\rfloor\right)\sum_{s=0}^{n_0}\binom{n}{s}=\Theta\left(\left\lfloor\frac{n}{n_0}\right\rfloor^{k-1}n^{n_0}\right). \qquad \square$$

We now analyze the number of response regions as a function of the number of parameters. When k and n_0 are fixed, then $\lfloor n/n_0\rfloor^{k-1}$ grows polynomially in n, and k^{n_0} is constant. On the other hand, when n is fixed with $n > 2n_0$, then $\lfloor n/n_0\rfloor^{k-1}$ grows exponentially in k, and k^{n_0} grows polynomially in k.

Proposition 7. *The number of parameters of a deep model with $n_0 = O(1)$ inputs, $n_{\text{out}} = O(1)$ outputs, and k hidden layers of width n is*

$$(k-1)n^2 + (k + n_0 + n_{\text{out}})n + n_{\text{out}} = O(kn^2).$$

The number of parameters of a shallow model with $n_0 = O(1)$ inputs, $n_{\text{out}} = O(1)$ outputs, and kn hidden units is

$$(n_0 + n_{\text{out}})kn + n + n_{\text{out}} = O(kn).$$

Proof. For the deep model, each layer, except the first and last, has an input weight matrix with n^2 entries and a bias vector of length n. This gives a total of $(k-1)n^2 + (k-1)n$ parameters. The first layer has nn_0 input weights and n bias. The output layer has nn_{out} input weight matrix and n_{out} bias. If we sum these together we get

$$(k-1)n^2 + n(k + n_0 + n_{\text{out}}) + n_{\text{out}} = O(kn^2).$$

For the shallow model, the hidden layer has knn_0 input weights and kn bias. The output weights has knn_{out} input weights and n_{out} bias. Summing these together we get

$$kn(n_0 + n_{\text{out}}) + n + n_{\text{out}} = O(kn).$$

$\square$

The number of linear regions per parameter can be given as follows.

Proposition 8. *Consider a fixed number of inputs n_0 and a fixed number of outputs n_{out}. The maximal ratio of the number of response regions to the number of parameters of a deep model with k layers of width n is*

$$\Omega\left(\left\lfloor\frac{n}{n_0}\right\rfloor^{k-1}\frac{n^{n_0-2}}{k}\right).$$

In the case of a shallow model with kn hidden units, the ratio is

$$O\left(k^{n_0-1}n^{n_0-1}\right).$$

Proof. This is by combining Proposition 6 and Proposition 7. $\square$

We see that fixing the number of parameters, deep models can compute functions with many more regions of linearity that those computable by shallow models. The ratio is exponential in the number of hidden layers k and thus in the number of hidden units.

Acknowledgments

We would like to thank KyungHyun Cho, Çağlar Gülçehre, and anonymous ICLR reviewers for their comments. Razvan Pascanu is supported by a DeepMind Fellowship.

References

Y. Bengio. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

Y. Bengio and O. Delalleau. On the expressive power of deep architectures. In J. Kivinen, C. Szepesvri, E. Ukkonen, and T. Zeugmann, editors, *Algorithmic Learning Theory*, volume 6925 of *Lecture Notes in Computer Science*, pages 18–36. Springer Berlin Heidelberg, 2011.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *AISTATS*, 2011.

I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *ICML'2013*, 2013.

A. Hajnal, W. Maass, P. Pudlk, M. Szegedy, and G. Turn. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46(2):129–154, 1993.

J. Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18th annual ACM Symposium on Theory of Computing*, pages 6–20, Berkeley, California, 1986. ACM Press.

J. Håstad and M. Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1: 113–129, 1991.

G. Hinton, L. Deng, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov. 2012a.

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinv. Improving neural networks by preventing co-adaptation of feature detectors. Technical report, arXiv:1207.0580, 2012b.

N. Le Roux and Y. Bengio. Deep belief networks are compact universal approximators. *Neural Computation*, 22(8):2192–2207, 2010.

H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. Montreal (QC), Canada, 2009.

J. Martens, A. Chattopadhya, T. Pitassi, and R. Zemel. On the expressive power of restricted boltzmann machines. In *Advances in Neural Information Processing Systems 26*, pages 2877–2885. 2013.

G. Montúfar and N. Ay. Refinements of universal approximation results for deep belief networks and restricted Boltzmann machines. *Neural Computation*, 23(5):1306–1319, 2011.

G. Montúfar and J. Morton. When does a mixture of products contain a product of mixtures? *arXiv preprint arXiv:1206.0387*, 2012.

G. Montúfar, J. Rauh, and N. Ay. Expressive power and approximation errors of restricted Boltzmann machines. *Advances in Neural Information Processing Systems*, 24:415–423, 2011.

V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. pages 807–814, 2010.

H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 689–690, 2011.

R. Stanley. An introduction to hyperplane arrangements. In *Lect. notes, IAS/Park City Math. Inst.*, 2004.

I. Sutskever and G. E. Hinton. Deep, narrow sigmoid belief networks are universal approximators. *Neural Computation*, 20(11):2629–2636, 2008.

T. Zaslavsky. *Facing Up to Arrangements: Face-Count Formulas for Partitions of Space by Hyperplanes*. Number no. 154 in Memoirs of the American Mathematical Society. American Mathematical Society, 1975.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. Technical report, arXiv:1311.2901, 2013.