Generalization of an Upper Bound on the Number of Nodes Needed to Achieve Linear Separability

Marjolein Troost✉, Katja Seeliger, Marcel van Gerven

Radboud University
Donders Institute for Brain, Cognition and Behaviour
Nijmegen, The Netherlands
marjolein.troost@student.ru.nl

Abstract. An important issue in neural network research is how to choose the number of nodes and layers such as to solve a classification problem. We provide new intuitions based on earlier results by [1] by deriving an upper bound on the number of nodes in networks with two hidden layers such that linear separability can be achieved. Concretely, we show that if the data can be described in terms of N finite sets and the used activation function f is non-constant, increasing and has a left asymptote, we can derive how many nodes are needed to linearly separate these sets. For the leaky rectified linear activation function, we prove separately that under some conditions on the slope, the same number of layers and nodes as for the aforementioned activation functions is sufficient. We empirically validate our claims.

1 Introduction

Artificial neural networks perform very well on classification problems. They are known to be able to linearly separate almost all input sets efficiently. However, it is not generally known how the artificial neural networks actually obtain this separation so efficiently. Therefore, it is difficult to choose a suitable network to separate a particular dataset. Hence, it would be useful if, given a dataset used for training and a chosen activation function, one can analytically derive how many layers and nodes are necessary and sufficient for achieving linear separability on the training set. Even though an analytical solution is still far away, some steps in the right direction have already been taken.

An et al. [1] showed for rectified linear activation functions that the number of hidden layers sufficient for linearly separating any number of (finite) datasets is 2 (follows from universality as well) and that the number of nodes per layer can be determined using disjoint convex hull decompositions. Yuan et al. [6] have provided estimates for the number of nodes per layer in a two-layer network based on information-entropy. Fujita [5] has done the same based on statistics by adding extra nodes one by one. Another approach by Kůrková [4] is to calculate how well a function can be approximated using a fixed number of nodes. Baum [7] has shown that a single-layer network can approximate a random dichotomy with only N/d units for an arbitrary set of N points in general position in d

dimensions. He also makes the link to the Vapnik-Chervonenkis dimension of the network. In this work we do not use statistics to achieve an estimate of the number of nodes but rather simple algebra to obtain an absolute upper bound, following An et al. [1] and Baum [7]. However, we will obtain this bound for multiple activation functions and arbitrary finite sets.

It is well-known that two-layer neural networks are universal approximators (e.g. [2, 3] or more recently [8]). However, even though we know there should exist a network that can linearly separate two arbitrary finite sets, we do not know which one it is. Choosing the wrong kind of network can lead to severe overfitting and reduced performance on the test set [6]. Therefore, it is useful to have an upper bound on the number of nodes. The upper bound can aid in choosing an appropriate network for a task. With this in mind, we aim to give a theoretical upper bound on the size of a network with two hidden layers in terms of nodes, that is easily computable for any finite input sets that need to be separated.

The rest of this work is organized as follows: in Section 2 we repeat some of the definitions from [1] and we give a direct extension of two of their theorems for which their proof does not need to be changed. In Section 3 we present our main theorem, which generalizes the two theorems from Section 2 to a larger class of activation functions. In Section 4 we add some corollaries and refer to an extension to multiple sets that is given in [1], we also provide an algorithm to estimate the upper bound on the number of nodes. We show simulation results that support our claims in Section 5 and conclude with some final remarks in Section 6.

2 Definitions and Basic Results

We want to emphasize that the following definitions and theorems (Definition 1, Theorems 3 and 5 and Corollary 12) are due to An et al.[1] and are repeated here for convenience. We took the liberty of adapting some of these definitions for clarity and giving slightly stronger versions of their Theorems 4 and 5 in Theorems 3 and 5 which follow directly from the proof given by [1].

Throughout the article, we will use the following notation and conventions: all sets are finite. We use f to denote a non-constant activation function that is always applied element-wise to its argument. So

$$f((x_1, x_2, \ldots, x_n)^T) = (f(x_1), f(x_2) \ldots, f(x_n))^T. \tag{1}$$

$\mathbb{R}$ is the set of real numbers. We define the convex hull of a set as the set of all convex combinations of the points in the set. In set notation:

$$C(X) = \left\{ \sum_{i=1}^{|X|} \alpha_i x_i \,\middle|\, \forall i \;\; \alpha_i \geq 0, \quad \sum_{i=1}^{|X|} \alpha_i = 1 \right\}. \tag{2}$$

We will now first define what is meant by a disjoint convex hull decomposition.

Definition 1 *Let X_k, $k \in \{1, \ldots, m\}$, be m disjoint, finite sets in $\mathbb{R}^n$. A decomposition of $X_1, \ldots, X_m$, $X_k = \bigcup_{i=1}^{L_k} X_k^i$, with $L_k \geq 1$ is called a disjoint convex hull decomposition if the unions of the convex hulls of X_k^i,*

$$\hat{X}_k \triangleq \bigcup_{i=1}^{L_k} C(X_k^i)\,, \tag{3}$$

are still disjoint. I.e. for all $k \neq l$: $\hat{X}_k \cap \hat{X}_l = \emptyset$. See for an illustration Figure 1B.

Since we are interested in finite sets, we can always define a disjoint convex hull decomposition (just take every point as a singleton). Such a decomposition is not unique. We would like to find a decomposition with the smallest L_ks. However, it is not necessary for the following that the decomposition is minimal. For two sets we also use the terminology described in Definition 2. This definition can easily be extended to multiple sets by applying it pairwise.

Definition 2 *If $C(X_1) \cap C(X_2) = \emptyset$, X_1 and X_2 are called **linearly separable**. If $C(X_1) \cap X_2 = \emptyset$ or $X_1 \cap C(X_2) = \emptyset$, X_1 and X_2 are called **convexly separable**. If all disjoint convex hull decompositions of X_1 and X_2 satisfy $\min(L_1, L_2) > 1$, X_1 and X_2 are called **convexly inseparable**.*

We start by giving a generalization of Theorem 4 from [1]. Instead of considering a rectified linear classifier activation function, we consider the more general class of functions that satisfy $f(x) = 0$ for $x \leq 0$ and $f(x) > 0$ for $x > 0$. We will call these functions **semi-positive**. Notice that they can be any function of $x > 0$ as long as they remain positive. This generalization is straightforward and the proofs do not need to be adapted but are given here for easy reference.

Theorem 3 *Let X_1 and X_2 be two convexly separable sets, with a finite number of points in $\mathbb{R}^n$. Say, $C(X_1) \cap X_2 = \emptyset$ and $X_2 = \bigcup_{j=1}^{L_2} X_2^j$ with $L_2 \in \mathbb{N}, X_2^j \subseteq X_2$ such that $C(X_1) \cap C(X_2^j) = \emptyset$ for each j. Let $w_j^T x + b_j$ be linear classifiers of X_2^j and X_1 such that for all j*

$$w_j^T x + b_j \leq 0 \quad \forall x \in X_1 \tag{4}$$

$$w_j^T x + b_j > 0 \quad \forall x \in X_2^j\,. \tag{5}$$

Let $W = [w_1, \ldots, w_{L_2}]$, $b = [b_1, \ldots, b_{L_2}]^T$ and $Z_k = \{f(W^T x + b) \mid x \in X_k\}, k \in \{1, 2\}$. Here f is a semi-positive function that is applied component-wise. Then Z_1 and Z_2 are linearly separable. For this we need L_2 affine transformations.

Proof. For all $x \in X_1$ we have that $w_j^T x + b_j \leq 0$. So $Z_1 = \{f(W^T x + b) \mid x \in X_1\} = \{(f(w_j^T x + b_j))_j \mid x \in X_1\} = \{0\}$. Now, for an $x \in X_2$, there exists a j such that $x \in X_2^j$. So, there exists a j such that $w_j^T x + b_j > 0$. Therefore, each $z \in Z_2$ has components greater or equal to zero and at least one component that is strictly greater than zero. This means $C(Z_1) \cap C(Z_2) = \emptyset$. We used L_2 transformations to create Z_1 and Z_2.

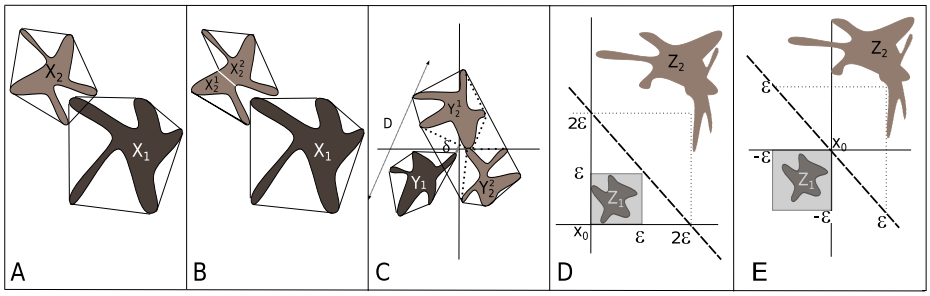

Fig. 1. This figure illustrates the proof of Theorems 7 and 11. **(A)** We see the two original sets and their convex hulls (outline). **(B)** X_2 is separated in two parts such that the convex hull of each part is linearly separable from the convex hull of X_1. **(C)** We then apply a linear transformation such that all points x in X_1 end up below x_0 (the origin) and all points in X_2^2 have first coordinate above x_0 and all points in X_2^1 have the second coordinate above x_0. By again drawing the convex hulls we can determine the minimal distance between the convex hull of Y_1 and the convex hulls of Y_2^1 and Y_2^2. **(D)** Then we apply f. Z_1 will become enveloped by a regular hypercube, and Z_2 will lie outside a hypercube with edges that are $L_2 = 2$ times as long. The separating plane is drawn as a dashed line. **(E)** Equivalently, a translated picture is used in the proof of the leaky rectified linear activation function. Instead of Figure 1 D, we now have Z_1 below the axis. ϵ is chosen to be the same as the diameter D of the set in Figure 1 C.

The initial sets that the network needs to separate are denoted by X_k, see Figure 1 A. After applying a linear classifier to the initial sets, these will be denoted by Y_k such that after applying the transformation $w_j^T x + b_j$ on all $x \in X_1$, we get Y_1, see Figure 1 C. When we apply the activation function to elements in Y_k, we denote the resulting set by Z_k, shown in Figures 1 D and 1 E. This means that a neural network with a single hidden layer with L_2 nodes, can transform X_k into Z_k. The following theorem is a generalization of Theorem 5 from [1]. Again, this is straightforward and does not require any changes to the proof. The theorem will make use of the following Lemma.

Lemma 4 *Two finite sets are linearly separable if and only if there exists a one-dimensional projection that maps the sets to linearly separable sets.*

Proof. Suppose we have two sets that are linearly separable. Let l be the hyperplane that separates the data. Project the data on the axis that is orthogonal to the hyperplane. By this, l will be collapsed into a point that lies at the threshold between the two separated sets. If we have a one-dimensional projection of the two sets, and a threshold t, let m be the hyperplane orthogonal to the projection axis containing t. Then the sets will be linearly separated by m.

Theorem 5 *Let X_1 and X_2 be finite and convexly inseparable. Let $w_{ij}^T x + b_{ij}$ be linear classifiers of X_2^j and X_1^i such that for all i, j*

$$w_{ij}^T x + b_{ij} \leq 0 \quad \forall x \in X_1^i \tag{6}$$

$$w_{ij}^T x + b_{ij} > 0 \quad \forall x \in X_2^j \,. \tag{7}$$

Let $W_i = [w_{i1}, \ldots, w_{iL_2}]$ and $b_i = [b_{i1}, \ldots, b_{iL_2}]$. Let $W = [W_1, \ldots, W_{L_1}]$, $b = [b_1^T, \ldots, b_{L_1}^T]^T$ and $Z_k = \{f(W^T x + b) \mid x \in X_k\}$ for $k \in \{1, 2\}$. Also, let $Z_1^i = \{f(W^T x + b) \mid x \in X_1^i\}$. Here f is again semi-positive. Then Z_1 and $C(Z_2)$ are disjoint, so Z_1 and Z_2 are convexly separable. For this we need $L_1 L_2$ nodes.

Proof. Define $Z_{2i} = \{f(W_i^T x + b_i) \mid x \in X_2\}$ and $Z_{1i}^t = \{f(W_i^T x + b_i) \mid x \in X_1^t\}$. Notice that these sets are projections of Z_2 and Z_1^t. Apply Theorem 3 on X_1^i, X_2 and their images Z_{1i}^i and Z_{2i} under the transformation $f(W_i^T \cdot + b_i)$. Then we have

$$C(Z_{1i}^i) \cap C(Z_{2i}) = \emptyset \quad i \in \{1, \ldots, L_1\} \,. \tag{8}$$

With Lemma 4, we then also have that

$$C(Z_1^i) \cap C(Z_2) = \emptyset \quad i \in \{1, \ldots, L_1\} \,. \tag{9}$$

Since $Z_1 \subset \bigcup_{i=1}^{L_1} C(Z_1^i)$, we have $Z_1 \cap C(Z_2) = \emptyset$. Therefore Z_1 and Z_2 are convexly separable. We needed L_1 linear transformations to separate a single part of X_2 from all parts of X_1. So in total we need $L_1 L_2$ transformations to create Z_1 and Z_2.

From Theorem 5 and 3 we can conclude that any two sets that are disjoint, can be made linearly separable by a network with two hidden layers that applies the function f as above and has $L_2 L_1$ and L_1 nodes per respective layer.

3 Main Result

We can generalize Theorems 3 and 5 to a larger set of activation functions. We need the following lemma for that. For simplicity we define $0/0 = 0$.

Lemma 6 *For a given $\delta > 0$ and a fixed $L_2 > 0$, let $f : \mathbb{R} \to \mathbb{R}$ be increasing with a left asymptote to zero and $\inf_{x_0} \frac{f(x_0)}{f(x_0+\delta)} < \frac{1}{L_2}$. Then $\exists x_0 \in \mathbb{R}, \epsilon > 0$ such that $\forall x \leq x_0 : f(x) \in [0, \epsilon]$ and $\forall x \geq x_0 + \delta : f(x) > L_2 \epsilon$.*

Proof. Choose x_0 and ϵ such that $f(x_0) = \epsilon$ and $\frac{f(x_0)}{f(x_0+\delta)} < \frac{1}{L_2}$. Let $x \leq x_0$. Then $f(x) \leq f(x_0) = \epsilon$. Let $x \geq x_0 + \delta$, then $f(x) \geq f(x_0 + \delta) > f(x_0) L_2 = \epsilon L_2$.

Lemma 6 puts a constraint on the speed with which the function f increases (near $-\infty$). We need $f(x_0 + \delta) \geq L_2 f(x_0)$. So if we move by δ, the value of the function will be multiplied by L_2. Notice that this is a very rapidly growing function. If a function does not satisfy this constraint, we find there is a

minimum distance δ needed between the $C(Y_1)$ and $C(Y_2^1) \cup C(Y_2^2)$. Lemma 6 also implies that the function should have a left asymptote to zero, however, we can shift an activation function with a different left asymptote such that this holds and then shift it back later using Corollary 13. This way, the lemma, and therefore Theorems 7 and 8, holds for all commonly used activation functions. We will compute the distance δ for the sigmoid, hyperbolic tangent, rectified linear function and leaky rectified linear function in Corollary 15.

We would further like to note that we will from now on define $\delta = \min_j \delta_j$ where

$$\delta_j = \inf_{x,y} \{\|x - y\| \mid x \in C(Y_1), y \in C(Y_2^j)\} \tag{10}$$

is the smallest distance between the convex hulls of two sets.

Theorem 7 *Let X_1 and X_2 be two convexly separable sets, with a finite number of points in $\mathbb{R}^n$. So $C(X_1) \cap X_2 = \emptyset$ and $X_2 = \bigcup_{j=1}^{L_2} X_2^j$ with $L_2 \in \mathbb{N}, X_2^j \subseteq X_2$ such that $C(X_1) \cap C(X_2^j) = \emptyset$ for each j. Let $f : \mathbb{R} \to \mathbb{R}$ be increasing with a left asymptote to zero and define δ as in Equation 10, such that*

$$\inf_{x_0} \frac{f(x_0)}{f(x_0 + \delta)} < \frac{1}{L_2}\,. \tag{11}$$

For x_0 satisfying this inequality, let $w_j^T x + b_j$ be linear classifiers of X_2^j and X_1 such that for all j

$$\sup_{x \in X_1} \{w_j^T x + b_j\} = x_0, \tag{12}$$

$$w_j^T x + b_j \geq x_0 + \delta \quad \forall x \in X_2^j\,. \tag{13}$$

Let $W = [w_1, \ldots, w_{L_2}]$, $b = [b_1, \ldots, b_{L_2}]^T$ and $Z_k = \{f(W^T x + b) \mid x \in X_k\}, k \in \{1, 2\}$. Then Z_1 and Z_2 are linearly separable.

Proof. Choose, using Lemma 6, an $x_0 \in \mathbb{R}$ and $\epsilon > 0$ such that for all $x \leq x_0$ we have $f(x) \leq \epsilon$ and for all $x \geq x_0 + \delta$ we have $f(x) > L_2\epsilon$. For all $x \in X_1$ we have $w_j^T x + b_j \leq x_0$. So $f(w_j^T x + b_j) \leq \epsilon$. Therefore, Z_1 is contained in a positive hypercube $[0, \epsilon]^{L_2}$. For all $x \in X_2$ there is a j such that $x \in X_2^j$. For $x \in X_2^j$ we have that $w_j^T x + b_j \geq x_0 + \delta$. So $f(w_j^T x + b_j) > L_2\epsilon$ for at least one coordinate, and all other coordinates are larger than 0. Therefore, $Z_2 \subseteq [0, \infty)^{L_2} \setminus [0, L_2\epsilon]^{L_2}$, see Figure 1 D. The convex hulls of these two sets can be separated by the hyperplane $\sum_{i=1}^{L_2} x_i = \epsilon L_2$. This is because the convex hull of Z_1 is contained in the hypercube with edges ϵ and the convex hull of Z_2 is bounded by the separating hyperplane.

Theorem 8 *Let X_1 and X_2 be finite and convexly inseparable. Let $f : \mathbb{R} \to \mathbb{R}$ be increasing with a left asymptote to zero, define δ as in Equation 10 such that $\inf_{x_0} \frac{f(x_0)}{f(x_0+\delta)} < \frac{1}{L_2}$. For x_0 satisfying this inequality, let $w_{ij}^T x + b_{ij}$ be linear*

classifiers of X_2^j and X_1^i such that for all i, j

$$\sup_{x \in X_1^i} \{w_{ij}^T x + b_{ij}\} = x_0, \tag{14}$$

$$w_{ij}^T x + b_{ij} \geq x_0 + \delta \quad \forall x \in X_2^j \,. \tag{15}$$

Let $W_i = [w_{i1}, \ldots, w_{iL_2}]$ and $b_i = [b_{i1}, \ldots, b_{iL_2}]$. Let $W = [W_1, \ldots, W_{L_1}]$, $b = [b_1^T, \ldots, b_{L_1}^T]^T$ and $Z_k = \{f(W^T x + b) \mid x \in X_k\}$ for $k \in \{1, 2\}$. Also, let $Z_1^i = \{f(W^T x + b) \mid x \in X_1^i\}$. Then Z_1 and Z_2 are convexly separable.

Proof. Define $Z_{2i} = \{f(W_i^T x + b_i) \mid x \in X_2\}$ and $Z_{1i}^t = \{f(W_i^T x + b_i) \mid x \in X_1^t\}$. Notice that these sets are projections of Z_2 and Z_1^t. Apply Theorem 7 on X_1^t, X_2 and their images Z_{1i}^i and Z_{2i} under the transformation f. Then we have

$$C(Z_{1i}^i) \cap C(Z_{2i}) = \emptyset \quad i \in \{1, \ldots, L_1\} \,. \tag{16}$$

With Lemma 4, we then also have that

$$C(Z_1^i) \cap C(Z_2) = \emptyset \quad i \in \{1, \ldots, L_1\} \,. \tag{17}$$

Since $Z_1 \subset \bigcup_{i=1}^{L_1} C(Z_1^i)$, we have $Z_1 \cap C(Z_2) = \emptyset$. Therefore Z_1 and Z_2 are convexly separable.

So we see that both Theorems 3 and 5 can be generalized to increasing functions with a left asymptote to zero. We still need two layers with $L_1 L_2$ and L_1 nodes respectively. However, we also need a minimal separation δ between the convex hulls of the two sets (in Euclidean distance) after applying the first linear transform. We formalize this in the following theorem:

Theorem 9 *Given finite disjoint sets X_1 and X_2 with a disjoint convex hull decomposition with L_1 and L_2 sets in the partitions, and given an increasing activation function f with a left asymptote to zero, we can linearly separate X_1 and X_2 using an artificial neural network with an input layer, a layer with $L_1 L_2$ hidden nodes, a layer with L_1 hidden nodes and an output layer.*

Proof. We can assume X_1 and X_2 are convexly inseparable and have linear classifiers $w_{ij}^T x + b_{ij}$ as in Theorem 5. The corresponding δ is always greater than zero, and scales with (w, b). Since $f \neq 0$ we can scale δ such that $\inf_{x_0} \frac{f(x_0)}{f(x_0+\delta)} < \frac{1}{L_2}$. Then apply Theorem 8. We need $L_1 L_2$ affine transformations for separating the L_2 parts of X_2 and the L_1 parts of X_1. Then f is applied to all transformations. A neural network can do this by learning the weights and biases of the affine transformations and then applying f. Now we have L_2 pairs of convexly separable sets, which can be linearly separated using Theorem 7. For each j we need to find an affine plane that separates X_2^j from X_1. This means we have to learn L_2 affine transformations before applying f, which can be done by a neural network with L_2 nodes. Now we have two linearly separable sets, which can be separated by using a linear classifier as the output layer, which proves the theorem.

Note that this proof implies that we can separate X_1 and X_2 independent of the distance between Y_1 and Y_2, so independent of δ. The learning algorithm should be able to scale the weights and biases such that the sets can be separated no matter how small δ was originally.

We can also prove a similar theorem for the leaky rectified linear activation function, which does not have a left asymptote to zero. However, we need to prove Lemma 10 first. The diameter of a set A is defined as $\operatorname{diam}(A) = \sup\{\|x - y\| \mid x, y \in A\}$.

Lemma 10 *Suppose* $D = \operatorname{diam}(Y_1 \cup Y_2)$, δ *as in Equation 10, and* $f(x) = c_2 x$ *for* $x \geq 0$ *and* $f(x) = c_1 x$ *for* $x \leq 0$, *where* $c_2 > c_1$. *Then* $\mu(\delta, D) \triangleq \inf_{x_0} \frac{f(x_0) - f(x_0 - D)}{f(x_0 + \delta) - f(x_0 - D)}$ *is reached at* $x_0 = 0$.

Proof. We have four cases:

(a) $x_0 < 0, x_0 + \delta < 0$, then $x_0 - D < 0$:

$$\mu(\delta, D) = \inf_{x_0} \frac{c_1 x_0 - c_1 x_0 + c_1 D}{c_1 x_0 + c_1 \delta - c_1 x_0 + c_1 D} \tag{18}$$

$$= \inf_{x_0} \frac{1}{\frac{\delta}{D} + 1} = \frac{1}{\frac{\delta}{D} + 1} \tag{19}$$

(b) $x_0 < 0, x_0 + \delta \geq 0$, then $x_0 - D < 0$:

$$\mu(\delta, D) = \inf_{x_0} \frac{(c_1 - c_1) x_0 + c_1 D}{(c_2 - c_1) x_0 + c_2 \delta + c_1 D} \tag{20}$$

$$= \inf_{x_0} \frac{1}{\frac{(c_2 - c_1) x_0}{c_1 D} + \frac{c_2 \delta}{c_1 D} + 1} \tag{21}$$

$$= \frac{1}{\frac{c_2 \delta}{c_1 D} + 1} \tag{22}$$

for x_0 increasing to zero.

(c) $x_0 \geq 0, x_0 - D < 0$, then $x_0 + \delta \geq 0$:

$$\mu(\delta, D) = \inf_{x_0} \frac{c_1 x_0 - c_1 x_0 + c_1 D}{(c_2 - c_1) x_0 + c_2 \delta + c_1 D} \tag{23}$$

$$= \inf_{x_0} \frac{\frac{(c_2 - c_1) x_0}{c_1 D} + 1}{\frac{(c_2 - c_1) x_0}{c_1 D} + \frac{c_2 \delta}{c_1 D} + 1} \tag{24}$$

which is an increasing function on the interval $[0, D)$. Therefore the infimum will be at $x_0 = 0$.

(d) $x_0 \geq 0, x_0 - D \geq 0$, then $x_0 + \delta > 0$:

$$\mu(\delta, D) = \inf_{x_0} \frac{c_2 x_0 - c_2 x_0 + c_2 D}{c_2 x_0 + c_2 \delta - c_2 x_0 + c_2 D} \tag{25}$$

$$= \inf_{x_0} \frac{1}{\frac{\delta}{D} + 1} = \frac{1}{\frac{\delta}{D} + 1} \tag{26}$$

Since cases (a) and (d) are equal, and since $c_2/c_1 > 1$ we see that the infimum is assumed at the value $x_0 = 0$.

Now we are ready to prove the following theorem for leaky rectified linear functions. Because of Lemma 10 we can assume $x_0 = 0$.

Theorem 11 *Suppose we have X_1 and X_2 as in Theorem 3. Define δ as in Equation 10. Let $f(x) = c_1 x$ for $x \leq 0$ and $f(x) = c_2 x$ for $x \geq 0$ be increasing with*

$$\frac{-f(-D)}{f(\delta) - f(-D)} < \frac{1}{L_2}. \tag{27}$$

Then Z_1 and Z_2 as defined in Theorem 3 are linearly separable.

Proof. Let $\epsilon = -f(-D)$. Then $\forall -D \leq x \leq 0$ we have $-\epsilon \leq f(x) \leq 0$. And $\forall x > \delta$ we have $f(x) > (L_2 - 1)\epsilon$. For all $x \in X_1$ we know $-D \leq w_j^T x + b_j \leq 0$. Therefore for all $x \in Y_1$ we have $-\epsilon \leq f(x) \leq 0$ and for all $x \in Y_2$ we have that $f(x) \geq -\epsilon$ and there exists a j such that $f(x_j) > (L_2 - 1)\epsilon$. See Figure 1 E. Therefore the convex hulls of Z_1 and Z_2 are disjoint.

We will not prove a version of Theorem 5 for the leaky rectified linear activation function because this is straightforward and the proof is the same as the proof of Theorem 5. But we can conclude that for leaky rectified linear activation functions a network that consists of two layers and L_2 and $L_1 L_2$ nodes respectively, can achieve linear separability. If δ is not large enough, there are two options now: the network could learn to scale the function appropriately, or this could be done manually by increasing the fraction c_2/c_1. In the next section we will explore some consequences of these results. We will also provide a way to calculate L_1 and L_2.

4 Corollaries

We can generalize the results from Section 3 to any number of sets (Corollary 12), by using the similar result in Sections 3.4 and 3.5 from [1] as a foundation. After stating this result we will show that we can apply any translation to the function f in the above theorems while retaining their validity (Corollary 13). Then we will provide a cheap way to estimate L_1 and L_2 in the disjoint convex hull decomposition (Algorithm 1) and we will calculate δ, for the most commonly used activation functions (Corollary 15).

Corollary 12 *For any number of sets, the above holds, with adjusted L_1 and L_2. By using the result from Section 3.4 and 3.5 on multiple sets from [1] as a foundation, it is easy to see that the same reasoning will apply to Theorems 3, 5, 7, 8 and 11.*

We can generalize the theorems still a little more by showing that they also hold for translated versions of the activation function that satisfies the constraints.

Corollary 13 *Suppose $f(-\infty) = c$, f is increasing and we have* $\inf_{x_0} \frac{f(x_0)-c}{f(x_0+\delta)-c} \leq \frac{1}{L_2}$. *Then Theorem 7 still holds. Moreover, for left and right translations of f, Theorem 7 still holds.*

Proof. Define $g = f - c$. Then $g(-\infty) = 0$ and $\inf_{x_0} \frac{g(x_0)}{g(x_0+\delta)} \leq \frac{1}{L_2}$. Therefore the theorem holds for g. Adding a constant to the linear separable sets Z_1 and Z_2 does not affect their separability. So the theorem holds for f.
Let $g(x) = f(x+c)$ be a translated version of f. If we apply $g(y) = g(w_j^T x + b_j) = f(w_j^T x + b_j + c)$ we see that we could just subtract c from x_0 to get back to the original theorem. Therefore left and right translation of functions is allowed.

We need a way to estimate L_1 and L_2 for arbitrary datasets. Since it is difficult to decompose the sets in a high dimensional space, we found a way to do it in a low dimensional space. This will allow for a rough upper bound on L_1 and L_2 but does not guarantee that the smallest disjoint convex hull decomposition can be found.

Lemma 14 *If we have a disjoint convex hull decomposition of a projection of our dataset, this partition will also form a disjoint convex hull decomposition of the original dataset.*

Proof. Suppose $P(X_1)$ and $P(X_2)$ are n-dimensional projections of X_1 and X_2. Assume we have disjoint convex hull decompositions $\widehat{P(X_1)} = \bigcup_{j=1}^{L_1} C(P(X_1)^j)$ and $\widehat{P(X_2)} = \bigcup_{j=1}^{L_2} C(P(X_2)^j)$ such that $\widehat{P(X_1)} \cap \widehat{P(X_2)} = \emptyset$. Now take $\widehat{X_1} = \bigcup_{j=1}^{L_1} C(X_1^j)$ such that $P(X_1)^j = P(X_1^j)$ and $\widehat{X_2} = \bigcup_{j=1}^{L_2} C(X_2^j)$ such that $P(X_2)^j = P(X_2^j)$. Then we see that $C(P(X_1^j)) \cap C(P(X_2^i)) = \emptyset$. Therefore $P(C(X_1^j)) \cap P(C(X_2^i)) = \emptyset$. So we can conclude that $C(X_1^j) \cap C(X_2^i) = \emptyset$. So also $\widehat{X_1} = \bigcup_{j=1}^{L_1} C(X_1)^j$ and $\widehat{X_2} = \bigcup_{j=1}^{L_2} C(X_2)^j$ are a disjoint convex hull decomposition.

We can estimate the number of sets in the convex hull decomposition using Lemma 14 as follows: Take a random projection of the datasets, preferably a one-dimensional projection. Then find the disjoint convex hull decomposition of this projection alone. This is easy in one dimension as it can be done by counting how often one switches from one set to the other when traversing through the projection. This number is an upper bound for L_1 and L_2, but a very coarse one, as we are using a random projection. So it is necessary to repeat this for many more random projections and minimize for L_1 and L_2. We use this procedure in algorithm 1 to find a reasonable estimate for L_1 and L_2.

We can prove that this algorithm will actually give a disjoint convex hull decomposition. The algorithm has complexity $\mathcal{O}(n^2)$. The worst-case scenario for the while-loop contributes a factor n and computing the inner-products also contributes a factor n. It is fair to mention the algorithm also depends on the dimension of the data and on the number of random projections that is used. Both can be quite large. The number of random projections needs to be significantly larger than the dimension to get good results. Also notice that adding *count* to L_1 and L_2 in lines 19 and 20 is naïve and can easily be improved.

Algorithm 1 Estimating L1 and L2

Input: two sets, X and Y, of n-dimensional data,
        the sets are finite and disjoint.
while size of overlap is smaller than size of previous overlap
  count the number of times we do this
  calculate mx ← mean of X
            my ← mean of Y
  for x in X, y in Y
    project x and y on my-mx
  calculate cx ← maximum of the projection of X
            cy ← minimum of the projection of Y
  create overlapX ← all x with projection at least cy
        overlapY ← all y with projection at most cx
  replace X by overlapX, Y by overlapY
for t in 1:10000
  create random vector
  for x,y in overlapX and overlapY
    calculate px ← projection of x on random vector
              py ← projection of y on random vector
  count the number of set changes from px to py
  L1 ← number of set changes+count
  L2 ← number of set changes+1+count
minimize L1 and L2

Corollary 15 *For a dataset with convex hull decomposition in L_2 sets, recall that the minimal distance between the $C(Y_1)$ and $C(Y_2^j)$ is $\delta = \inf_j \delta_j$, see Equation 10. For the sigmoid the minimal δ needed for separation equals $\ln(L_2)$. For a shifted hyperbolic tangent the minimal δ equals $\frac{1}{2}\ln(L_2)$. For the ReLU the minimal δ equals 0. For the leaky rectified linear activation function $\delta = (L_2+1)\frac{c_2}{c_1}D$. Or equivalently, $\frac{c_2}{c_1} = \frac{\delta}{D(L_2-1)}$, then we are able to separate any two sets with a leaky rectified linear activation function.*

Proof. Note that proving that the limit becomes smaller than $1/L_2$ implies that the infimum also becomes smaller than $1/L_2$. For practical purposes we will use the limit in this proof.

Sigmoid. The sigmoid function is written as $\sigma(x) = \frac{e^x}{1+e^x}$. If we calculate

$$\frac{\sigma(x_0)}{\sigma(x_0+\delta)} = \frac{e^{x_0}}{1+e^{x_0}} \frac{1+e^{x_0}e^{\delta}}{e^{x_0}e^{\delta}} = \frac{e^{-\delta}+e^{x_0}}{1+e^{x_0}} \tag{28}$$

and then take the limit $x_0 \to -\infty$ we see that equation 28 goes to $e^{-\delta}$. To get this smaller than $1/L_2$ we need $\delta > \ln(L_2)$.

Hyperbolic tangent. We start by writing a shifted hyperbolic tangent $\tanh(x) + 1$ out in terms of exponentials. If we calculate

$$\frac{\tanh(x_0) + 1}{\tanh(x_0 + \delta) + 1} = \frac{2}{1 + e^{-2x_0}} \frac{1 + e^{-2x_0} e^{-2\delta}}{2} \tag{29}$$

$$= \frac{1 + e^{-2x_0} e^{-2\delta}}{1 + e^{-2x_0}} \tag{30}$$

and then take the limit $x_0 \to -\infty$ we get that equation 29 goes to $e^{-2\delta}$. To get this smaller than $1/L_2$ we need $\delta > \frac{1}{2}\ln(L_2)$. So also for the hyperbolic tangent we have with Corollary 13 that $\delta > \frac{1}{2}\ln(L_2)$.

Rectified linear function. We did not need any δ in the proof for the rectified linear function, so the minimal δ equals zero.

Leaky rectified linear activation function. With Lemma 10 we get:

$$\frac{-f(-D)}{f(\delta) - f(-D)} = \frac{Dc_1}{\delta c_2 + Dc_1}, \tag{31}$$

where f denotes the leaky rectified linear function. To get this smaller than $\frac{1}{L_2}$ we need $\delta = (L_2 + 1)\frac{c_2}{c_1} D$.

5 Simulation Results

We tested the ideas in Sections 3 and 4 empirically. We trained several networks with different sizes and activation functions on the first two classes (number classes 0 and 1) of the MNIST dataset [9]. We calculated the minimal distance between these two sets and found $\delta = 3.96$. This is a sufficient distance for any of the activation functions we used, which means the network is able to use weights close to 1. Next we estimated L_1 and L_2. For this dataset with more than 12000 data points, we found $L_1 = 6$ and $L_2 = 6$. That would mean that a network with 36 nodes in the first layer and 6 nodes in the second would be sufficient to linearly separate the data in the two sets.

Several hidden layer sizes were tested. All networks had a depth of three, hence four layers of nodes. An input layer with 784 nodes, two hidden layers with the sizes mentioned before, and an output layer with 2 nodes which acts as a classifier. Linear separability, as discussed in this paper, precisely means that this output layer can classify the input sets perfectly.

We compared the ReLU, sigmoid, leaky ReLU and tanh networks trained for 150 epochs using stochastic gradient descent optimization. For the leaky ReLU the slope was set to the standard value of 0.2. We implemented the linear classifier multi-layer perceptron in the neural network framework Chainer v2.0 [10]. We regard the training capabilities of this framework as a black box sufficient for our simulation needs. The results are displayed in Figure 2. Indeed as expected, the network with the hidden layer sizes estimated based on the proposed

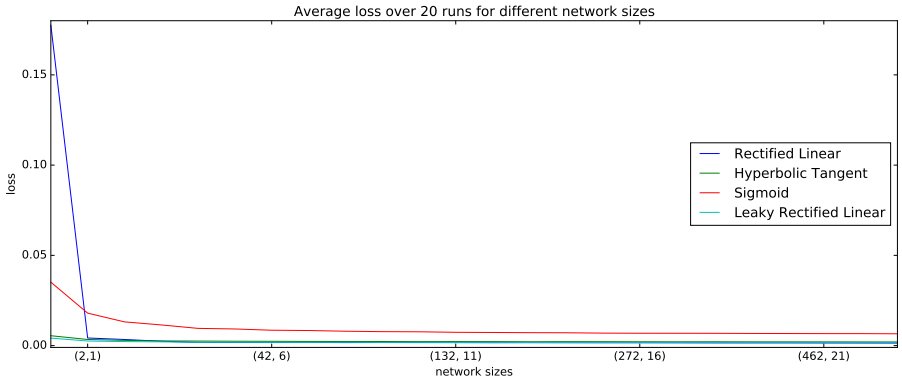

Fig. 2. This figure shows the averaged loss over 20 runs. The activation functions we used are the ReLU, the tanh, the sigmoid and the leaky ReLU. 25 networks of different sizes were trained using the above functions as activation functions. The numbers in brackets on the x-axis denote the sizes of the hidden layers. The size of the input layer was 784, the size of the output layer was 2. The network was trained with 150 epochs and a batch size of 150.

theoretical analysis ($(36, 6)$) performs very well. We see clearly that the losses barely decrease for larger networks. The error is not yet zero for the predicted network but this may be explained by the imperfect training. The ReLU network performs bad for the smallest network. This may be explained by the fact that the ReLU maps a lot of information to zero even though it has the smallest δ of the tested activation functions. The sigmoid consistently has a larger loss than the other functions. This is not necessarily predicted by the theory since the sigmoid's δ is only a factor 2 larger than the hyperbolic tangent's δ. Also interesting is the very good performance of the leaky ReLU network. This could be caused by not mapping a lot of information to zero like the ReLU as well as having two options to compensate for the δ.

All activation functions seem to imply that there exists a slightly smaller network that can achieve linear separability on the test set. A better algorithm for determining L_1 and L_2 can probably confirm this.

6 Discussion

The practical contribution of this article is heuristic. It is widely believed that deep neural networks need less nodes in total than shallow neural networks to solve the same problem. Our theory presents an upper bound on the number of nodes that a shallow neural network will need to solve a certain problem. Therefore, a deep neural network will not need more nodes. The theory does not give an optimal architecture, nor a minimum on the number of nodes. Still it

is useful to have an inkling about the correct network size for solving a certain problem.

Contrary to what An et al. [1] claim, their theory does not show why ReLU networks have a superior performance. We extended their theory to all commonly used activation functions. Only the leaky rectified linear networks seem to be at a disadvantage, but test results show the opposite. We think the differences between the functions may be caused by the scaling that needs to be done during learning. The linear functions and also the hyperbolic tangent are very easy to scale. Tweaking the sigmoid to the best slope can be quite difficult.

Some issues which we have not addressed in this article are worth mentioning. For example, we cannot make any statements about generalization performance of the networks. Of course, it is generally known that a network with too many parameters will not generalize well. So it is wise to use a network that is as small as possible, or even a bit smaller. This paper contributes an estimate for the number of nodes that is an absolute maximum. It should never be necessary to use more nodes than this estimate. We do not give a necessary number of nodes but rather an upper bound. A bound that is necessary and sufficient would be optimal, but this is a much harder problem to solve.

Another problem is that we do not know what will happen if we use too few nodes. The number of nodes that we estimated will guarantee linear separability. If the number of nodes is too small to achieve linear separability, performance on the training set will be reduced, but it is difficult to say anything about performance on the test set. We also do not know what will happen to the number and distribution of nodes as we increase the number of layers. An extension of the theory to an arbitrary number of layers would be very interesting.

Furthermore, in the simulations we cannot guarantee that the learning algorithm achieves zero error, even though it is possible in theory. The reason is that the algorithm does not always find the absolute minimum. Therefore it is hard to judge from the results whether the predicted network size is performing as expected.

Even though we already find small L_1 and L_2, more elaborate simulations could use another algorithm to find the convex hull decomposition. Random projections are cheap to use, but they will always find a pair L_1, L_2 such that $L_2 = L_1 + 1$. (We found $L_1 = L_2$ since no random projections were necessary.) This is a serious constraint because the first layer of the network consists of $L_1 L_2$ nodes, and will therefore always be very large if L_1 and L_2 are similar size. An idea would be to use a method that uses higher dimensional projections. It is also not guaranteed that Algorithm 1 performs well on other input sets. A better algorithm might perform well on all types of input sets.

The results show a stunning performance of the leaky ReLU activation. More research is needed to understand why this is the case. There clearly is more to the performance of a neural network than revealed in this article. Still, it is an important result to have an estimation of sufficient network sizes for certain activation functions. It would also be interesting to see the effect of the slope of

a leaky ReLU and the distance between the datasets on the performance of the network.

This paper provides a heuristic explanation why ReLU and perhaps leaky ReLU networks are easier to train than tanh and sigmoid networks. We give an upper bound on the number of nodes that is needed to achieve linear separability on the training set for feedforward networks with two hidden layers. It is still unclear how this generalises to more layers, which poses an interesting question for further research. Furthermore, our theory does not yet address convolutional networks, however it does represent a foundation for exploring their superior performance in an extension of this work.

References

1. An, S., Boussaid, F., Bennamoun, M.: How can deep rectifier networks achieve linear separability and preserve distances? 32nd International Conference on Machine Learning **37**, 514-523 (2015).
2. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks **2**, 359-366 (1989).
3. Arteaga, C., Marrero, I.: Universal approximation by radial basis function networks of Delsarte translates. Neural Networks **46** 299-305 (2013).
4. Kůrková, V., Kainen, P.C., Kreinovich, V.: Estimates of the number of hidden units and variation with respect to half-spaces. Neural Networks **10.6** 1061-1068 (1997).
5. Fujita, O.: Statistical estimation of the number of hidden units for feedforward neural networks. Neural Networks **11** 851-859 (1998).
6. Yuan, H.C., Xiong, F.L., Huai, X.Y.: A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy. Computer and electronics in agriculture **40** 57-64 (2003).
7. Baum, E.B.: On the capabilities of multilayer perceptrons. Journal of Complexity **4** 193-215 (1988).
8. Sonoda, S., Murata, N.: Neural network with unbounded activation functions is universal approximator. Applied and Computational Harmonic Analysis **43.2** 233-268 (2017).
9. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE **86.11** 2278-2324 (1998).
10. Tokui, S., Oono, K., Hido, S., Clayton, J.: Chainer: a Next-Generation Open Source Framework for Deep Learning. Proceedings of Workshop on Machine Learning Systems(LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS) (2015).