Why do networks have inhibitory/negative connections?

Qingyang Wang*[1,5], Michael A. Powell[2,5,7], Ali Geisa[2,5], Eric Bridgeford[4,5], Carey E. Priebe[6], and Joshua T. Vogelstein[1,2,3,4,5]

[1]Department of Neuroscience, Johns Hopkins University
[2]Department of Biomedical Engineering, Johns Hopkins University
[3]Institute for Computational Medicine, Kavli Neuroscience Discovery Institute, Johns Hopkins University
[4]Department of Biostatistics, Johns Hopkins University
[5]Center for Imaging Science, Johns Hopkins University
[6]Department of Applied Mathematics and Statistics, Johns Hopkins University, United States
[7]Current location: United States Military Academy, Department of Mathematical Sciences, West Point NY US

Why do brains have inhibitory connections? Why do deep networks have negative weights? We propose an answer from the perspective of representation capacity. We believe representing functions is the primary role of both (i) the brain in natural intelligence, and (ii) deep networks in artificial intelligence. Our answer to why there are inhibitory/negative weights is: to learn more functions. We prove that, in the absence of negative weights, neural networks with non-decreasing activation functions are *not* universal approximators. While this may be an intuitive result to some, to the best of our knowledge, there is no formal theory, in either machine learning or neuroscience, that demonstrates *why* negative weights are crucial in the context of representation capacity. Further, we provide insights on the geometric properties of the representation space that non-negative deep networks cannot represent. We expect these insights will yield a deeper understanding of more sophisticated inductive priors imposed on the distribution of weights that lead to more efficient biological and machine learning.

1 Introduction

Are inhibitory connections necessary for a brain to function? Many studies on mammals answer yes: a balanced excitatory/inhibitory (E/I) ratio is essential for memory [1], unbalanced E/I lead to either impulsive or indecisive behaviors [2], such balance is closely tracked throughout learning [3, 4], and imbalance is hypothesized to be the driving force behind epilepsy [5–7]. These simulation results and disease studies provide compelling evidence that E/I balance is necessary for brains to function stably.

Intriguingly, when neurons first came into existence in the long history of evolution, they were exclusively excitatory [8]. The Cnidarian jellyfish has a well-defined nervous system that is made of sensory neurons, motor neurons, and interneurons. Different from mammals, their synaptic connections are exclusively excitatory. Even though these jellyfish are not equipped with inhibitory neurons, they are perfectly capable of performing context-dependent behaviors: when they feed, they swim slowly through a weak, rhythmic contraction of the whole bell; when they sense a strong mechanical stimulus, they escape through a rapid, much stronger contraction [8]. Cnidarian jellyfish behave adaptively, not through sophisticated excitatory-inhibitory circuits, but instead by modifying voltage-gated channel properties (i.e., conductance). Cnidarian jellyfish prove to us that without inhibitory connections the brain can still function, albeit likely through alternative mechanisms. This re-raises the fundamental question: are inhibitory connections necessary for brains to function?

In this paper, we explore the necessity of inhibitory connections from the perspective of representation capacity. Instead of viewing the brain as a dynamical system to discuss its functional stability, we think about the brain as a feedforward network capable of representing functions. For a certain network structure, we are interested in characterizing the repertoire of functions such networks are capable of representing. Specifically, to understand the importance of inhibitory connections in networks' representation capacity, we ask what functions can non-negative networks represent? We do so with the help of Deep Neural Networks (DNNs). DNNs allow us to work at a level of abstraction where we can stay focused on the connectivity between computation units; they also allow us to completely take out inhibitory connections by setting all weights to be non-negative. They further allow us to build upon the well-celebrated DNN theoretical result that DNNs are universal approximators [9–11] (Theorem 3.2). We prove in this paper that DNNs with all non-negative weights are *not* universal approximators (sec 3). We

*qwang88@jhu.edu

further prove three geometric properties of the representation space for non-negative DNNs (sec 4). These results show that networks without negative connections not only lose universality; in fact, they have extremely limited representation space. We further extend our results to convolutional neural networks (CNNs) and other structural variants (sec 5). Such theoretical results serve as a plausible explanation for why purely excitatory nervous systems, along the history of evolution, were largely overtaken by brains containing both inhibitory and excitatory connections; the simpler systems may be able to perform interesting functions, but they also have limited representation capacity. Our work thus concludes that inhibitory connections are necessary for a brain to represent more functions.

2 Related works

2.1 Optical neural network

The optical neural network literature [12, 13] has some discussion on the representation capacity of non-negative DNNs; however, the existing discussion is restricted to only non-negative functions, i.e. $f(x) \geqslant 0,\ \forall x \in \mathcal{X}$. Our work does not impose any requirement on the network output since we allow the bias terms to take any value and the output of the network may be negative (dependent on the activation function choice).

2.2 Partially monotone networks

Partially monotone networks exploited the monotone nature of non-negative sub-networks to solve problems where certain input features are known to stay monotone. One notable application is in stock prediction [14]. An independent stream of research that falls into the category of partially monotone networks is input-convex networks, which are built for fast inference [15]. Our work is the first one using the fact that non-negative DNNs are order-preserving monotone functions to explicitly prove that they are not universal approximators; we are also the first pointing out its geometric consequences and their implications in neuroscience, as well as the first to extend these theoretical results to CNNs.

3 DNN$^+$s are not universal approximators

DNNs are universal approximators [9–11] (paraphrased in Theorem 3.2): given a large enough network, it can learn (represent) a continuous function arbitrarily well. Throughout this paper, we are not concerned about the learning process *per se*, but more about its performance upper bound. Given unlimited resources (e.g., time, energy, compute units), can networks of a certain structure learn a function at all? If it can, we say such networks can **represent** such a class of functions. We show below that non-negative DNNs (all weights non-negative, abbreviated as DNN$^+$) are *not* universal approximators (Theorem 3.3); thus, having both positive and negative weights is necessary for universal approximation. We start by defining the problem and then follow with our key observation: DNN$^+$s are composed of order-preserving monotone functions (Definition 3.4, Lemma 3.1); therefore, non-negative DNNs are all order-preserving monotone functions (Theorem 3.1). With this key property of DNN$^+$ in mind, we can then show that DNN$^+$s cannot solve XOR and are therefore not universal approximators.

3.1 DNN$^+$

Definition 3.1 (DNN)**.** *A feedforward fully-connected (FC) neural network architecture is given by the tuple* $(\mathbf{\Phi}, n_0, n_1, n_2, \dots, n_L)$*, where* $L \in \mathbb{N}^+$ *is the number of layers,* $n_l \in \mathbb{N}$ *is the number of units of layer* l*,* $l \in [L]$*;* $\mathbf{\Phi} : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$ *is a point-wise nonlinear non-decreasing function, and it defines the non-linear component of layer* l*. We call* $\mathbf{\Phi}$ *the activation function. The linear component of the transformation from layer* $l-1$ *to* l *is given by the weight matrix* $\mathbf{W}^{(l)} \in \mathbb{R}^{n_l \times n_{(l-1)}}$ *and bias vector* $\mathbf{b}^{(l)} \in \mathbb{R}^{n_l}$*. Collectively, the function given by a DNN with the above architecture tuple* $(\mathbf{\Phi}, n_0, n_1, n_2, \dots, n_L)$ *takes input* $\mathbf{x} \in \mathbb{R}^{n_0}$ *and is defined as following:*

$$F : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}, \quad F(\mathbf{x}) = \mathbf{\Phi}(\mathbf{W}^{(L)}\mathbf{\Phi}(\mathbf{W}^{(L-1)} \dots \mathbf{\Phi}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) \dots + \mathbf{b}^{(L-1)}) + \mathbf{b}^{(L)}), \quad \mathbf{x} \in \mathbb{R}^{n_0}. \quad (1)$$

Non-negative DNNs (DNN$^+$s) are DNNs with all weights non-negative (without any constraint on the bias terms).

Definition 3.2 (DNN$^+$). *DNN$^+$ is a DNN with all non-negative weights. $F^+ : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$,*

$$F^+(\mathbf{x}) = \mathbf{\Phi}(\mathbf{W}^{(L)} \dots \mathbf{\Phi}(\mathbf{W}^{(l)} \dots \mathbf{x} \dots + \mathbf{b}^{(l)}) \dots + \mathbf{b}^{(L)})$$
$$\forall l \in [L], \quad \mathbf{W}^{(l)} \geq \mathbf{0}^{1}, \quad \mathbf{b}^{(l)} \in \mathbb{R}^{n_l}$$

Remark 1 (Activation function $\mathbf{\Phi}$). *We only impose one constraint to the activation function $\mathbf{\Phi}$ in addition to it being non-linear: it must be non-decreasing. Popular choices of activation function fall into this category (e.g., ReLU, leaky ReLU, sigmoid, tanh, etc.).*

Throughout the paper, we boldface $\mathbf{\Phi}$ to emphasize it is a point-wise function on the vector space and to distinguish it from its base form $\phi : \mathbb{R} \to \mathbb{R}$. By point-wise, we mean $\mathbf{\Phi}(\mathbf{x}) = (\phi(x_1), \phi(x_2), ..., \phi(x_{n_l}))$.

Two examples of ϕ are given below:

$$ReLU : \phi(x) = \begin{cases} 0 & x < 0 \\ x & x \geqslant 0 \end{cases} \qquad \text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

Remark 2 (Extensions). *Definition 3.1 presents the most classical form of DNN as defined in the original universal approximator papers [9–11]. Such DNNs are also called multi-layer perceptrons (MLPs). Since the 1990s, variants of the basic operations have evolved. We will discuss some of these extensions in section 5 and then prove that all of our results in sections 3 and 4 generalize to these extended DNN architectures.*

3.2 DNN$^+$s are order-preserving monotone functions

We are particularly interested in a concise description of the class of functions representable by DNN$^+$. Intuitively, by constraining all weights to be non-negative, the possible transformation operations of the linear components of the functions are limited, which cannot be overcome by translations given by the bias terms, even though the bias terms can change within the full range of $\mathbb{R}$. This means that rotations or reflections that result in a change of orthant are impossible. One important consequence of these limitations is that the composite function becomes a non-decreasing monotone function in high-dimensional space. Below, we formally prove these ideas by first defining a partial ordering in the high-dimensional vector space (Definition 3.3), and then by proving DNN$^+$s are order-preserving functions (Theorem 3.1).

Definition 3.3 ($\preceq$, partial ordering on $\mathbb{R}^n$). *For any pair of $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we define a partial ordering[2]*

$$\mathbf{x} \preceq \mathbf{y} \quad \equiv \quad x_k \leqslant y_k, \ \forall k \in [n].$$

Definition 3.4 (order-preserving monotone function, $\mathbb{R}^m \to \mathbb{R}^n$). *For a function $T : \mathbb{R}^m \to \mathbb{R}^n$, we say T is a **order-preserving monotone function** when for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$*

$$\mathbf{x} \preceq \mathbf{y} \implies T(\mathbf{x}) \preceq T(\mathbf{y}).$$

Lemma 3.1 (closure of order-preserving monotone functions under sum, positive scaling, and translation). *Given an affine transformation $T : \mathbb{R}^m \to \mathbb{R}^n$ with a weight matrix $\mathbf{W}$ of all non-negative entries:*

$$T(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}, \ \mathbf{x} \in \mathbb{R}^m, W \in \mathbb{R}^{n \times m}_{\geq 0}, \mathbf{b} \in \mathbb{R}^n.$$

We observe such T is a monotone function (Def 3.4). That is, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^m$,

$$\mathbf{x} \preceq \mathbf{y} \implies T(\mathbf{x}) \preceq T(\mathbf{y}).$$

See proof on page 13.

Since the activation function of DNN$^+$ is also a non-decreasing monotone function, we immediately have F^+ is also an order-preserving function because it is the composite of order-preserving functions.

[1] $\forall i \in [n_{l-1}], j \in [n_l], \ w_{ij} \geq 0$
[2] This is a standard order induced by the positive cone $\mathbb{R}_+{}^n$.

Theorem 3.1 (DNN$^+$ is order-preserving monotone function)**.** *A DNN$^+$ $F^+ : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ is an order-preserving monotone function (Defn 3.4). That is, any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n_0}$,*

$$\mathbf{x} \preceq \mathbf{y} \implies F^+(\mathbf{x}) \preceq F^+(\mathbf{y}).$$

See proof on page 13.

3.3 DNN$^+$s are not universal approximators

Intuitively, DNN$^+$ being an order-preserving monotone function means it cannot solve any classification problems where order reversal is required. We will use this idea to prove DNN$^+$ cannot solve XOR, and thus they are not universal approximators (Theorem 3.3). Below we paraphrase the universal approximator theorem from Hornik's 1989 paper.

Theorem 3.2 (Multilayer feedforward networks are universal approximators (paraphrased from Hornik 1989 [11]))**.** *Let $C(K, \mathbb{R}^n)$ denotes the set of continuous functions from a compact domain $K \subseteq \mathbb{R}^m$ to $\mathbb{R}^n$. Then for every $m \in \mathbb{N}, n \in \mathbb{N}$, for all $K \subseteq \mathbb{R}^m$, $f \in C(K, \mathbb{R}^n)$, and for all $\epsilon > 0$, there exists a DNN F (Definition 3.1) such that*

$$\sup_{\mathbf{x} \in K} |f(\mathbf{x}) - F(\mathbf{x})| < \epsilon$$

One may notice, in the original universal approximator proofs [9–11], activation functions were required to be continuous, bounded, and non-constant. Thus for DNNs with non-decreasing monotone activation functions, they are all universal approximators. However, this is no longer true when all weights are constrained to be non-negative, as we will prove below by showing a counter example: DNN$^+$ cannot solve XOR.

Definition 3.5 (XOR-continuous)**.** *Let $x \in [0, 1]^2$, $f : [0, 1]^2 \to \mathbb{R}$*

$$f(x_1, x_2) = x_1 + x_2 - 2x_1x_2 \tag{2}$$

Note: f(0,0)=f(1,1)=0; f(1,0)=f(0,1)=1.

Remark 3 (XOR-discontinuous)**.** *We also define a discontinuous version of XOR for the readers' convenience. Such a definition variation does not alter our result (Theorem 3.3) in any way.*

Let $x \in [0, 1]^2$ and $t_1, t_2 \in (0, 1)$, $f : [0, 1]^2 \to \{0, 1\}$

$$f(x) = \begin{cases} 0, [x_1 \geq t_1 \text{ and } x_2 \geq t_2] \text{ or } [x_1 < t_1 \text{ and } x_2 < t_2] \\ 1, [x_1 \geq t_1 \text{ and } x_2 < t_2] \text{ or } [x_1 < t_1 \text{ and } x_2 \geq t_2]. \end{cases} \tag{3}$$

Theorem 3.3 (DNN$^+$ is not a universal approximator)**.** *DNN$^+$s are not universal approximators. That is, there exists a combination of (m, n, K, f) where $m \in \mathbb{N}, n \in \mathbb{N}$, $K \subseteq \mathbb{R}^m$, $f \in C(K, \mathbb{R}^n)$, and exists an $\epsilon > 0$ such that for all DNN$^+$ F^+ (Definition 3.2),*

$$\exists\, \mathbf{x} \in K, \text{such that } |f(\mathbf{x}) - F^+(\mathbf{x})| \geqslant \epsilon$$

Proof. It suffices to find one combination (m, n, K, f) such that *no* F^+ could approximate f arbitrarily well. XOR, $(m = 2, n = 1, K = [0, 1]^2, f =$ Defn 3.5), is such a combination, as we will prove below in Corollary 3.3.1.

Corollary 3.3.1 (DNN$^+$ cannot approximate XOR)**.** *For f as defined in Definition 3.5, $\exists\, \epsilon > 0$ such that $\forall\, F^+$ (Definition 3.2), $\exists\, \mathbf{x} \in \mathbb{R}^2$, such that $|f(\mathbf{x}) - F^+(\mathbf{x})| \geqslant \epsilon$*

Proof. We will prove by contradiction. Assume DNN$^+$ F^+ can approximate XOR f well, only with an error term of $\epsilon > 0$ for all $\mathbf{x} \in \mathbb{R}^2$.

Now lets consider three points $(0, 0), (1, 0), (1, 1)$, we must have

$$|F^+(0, 0) - f(0, 0)| < \epsilon, \tag{4}$$
$$|F^+(1, 0) - f(1, 0)| < \epsilon, \tag{5}$$
$$|F^+(1, 1) - f(1, 1)| < \epsilon. \tag{6}$$

From XOR definition 3.5, we also have f(0,0)=f(1,1)=0, f(1,0)=f(0,1)=1, therefore,

$$|F^+(0,0)| < \epsilon,$$
$$|F^+(1,0) - 1| < \epsilon \implies 1 - \epsilon < F^+(1,0) < 1 + \epsilon$$
$$|F^+(1,1)| < \epsilon.$$

We can pick any $\epsilon < 0.5$ and have

$$\begin{cases} F^+(0,0) < F^+(1,0) \\ F^+(1,1) < F^+(1,0) \end{cases} \tag{7}$$

This is obviously contradictory to the fact that F^+ is is order-preserving monotone function, that is:

$$F^+(0,0) \leqslant F^+(1,0) \leqslant F^+(1,1) \tag{8}$$

The same logic applies to $\epsilon > 0.5$, and for our proof we only need to find one such ϵ. Therefore, $\exists \epsilon > 0$, such that for all F^+ we cannot approximate XOR f ϵ-well. □

Therefore, there does not exist an F^+ that can approximate XOR f arbitrarily well. Thus F^+ is not a universal approximator. □

From Theorem 3.3, we have that DNNs with non-negative weights are not universal approximators – they cannot even solve XOR. This is because they are order-preserving monotone functions. Can we overcome this limitation by flipping the sign of a single weight to make the network XOR-capable? The answer is *yes*, and we illustrate this result in Figure 1, panel C: in a 3-unit single hidden layer network, flipping one output edge of one hidden unit negates the first quadrant, thus sculpting it out of the pink class region and joining it with the third quadrant.

3.4 Implication in neuroscience

How does our discussion on XOR pertain to the brain? Let's think about a simple discrimination task where the animal has to make a decision based on an input stimuli that has two features $x \in \{0,1\}^2$. For example, a deer has to tell toxic leaves apart from edible leaves. Due to seasonal changes, toxic leaves could be either green straight ($(0,0)$) or red curves ($(1,1)$); edible leaves could be either green curved ($(0,1)$) or red straight ($(1,0)$). To survive, the deer must be able to discriminate between toxic and edible leaves where the decision boundary follows exactly the XOR pattern. In fact, any discrimination task taking stimuli of the form $x \in \{0,1\}^2$ where the decision is dependent on both features is an XOR task [16]. Only after inhibitory connections evolved could organisms survive in more complex environments and perform more complex tasks.

4 Geometric intuitions of DNN$^+$ in $\mathbb{R}^n$

So far we have proven feed-forward, fully connected neural networks without any negative weights are not universal approximators (Theorem 3.3) due to their order-preserving nature (Theorem 3.1). Next we will hone in on the concept of limited representation capacity by delineating three types of classification problems that DNN$^+$s fail to solve (Figure 1 left column). These will also provide some geometric intuition on order-preserving functions. We summarize all theoretical results in $\mathbb{R}^2$ in Figure 1 for intuitive comprehension. We prove all cases in finite dimensions $\mathbb{R}^n, n \in \mathbb{Z}^+$ in the three corollaries below.

1. Corollary 3.1.1 notes that DNN$^+$s cannot solve a classification problem that requires the decision boundaries to have segments with a positive slope ($\mathbb{R}^2$), or a section of its decision hyperplane having a normal with all positive elements ($\mathbb{R}^n$). This is illustrated in Figure 1 panel A). This follows from the fact that positive weights can only form decision boundaries that have negative slopes.

2. Corollary 3.1.2 notes that DNN$^+$s cannot solve classification problems where there exists a class whose decision boundary forms a closed shape ($\mathbb{R}^2$), or a closed region ($\mathbb{R}^n$). Order-preserving means that for all units in the non-negative DNNs, their activation gradients point towards the positive directions of all dimensions (top and/or right in $\mathbb{R}^2$). However, for a closed-shape decision boundary, it requires the gradient to point in opposite directions (both towards and away from the partition), which is not doable with non-negative DNNs.

3. Corollary 3.1.3 notes that DNN$^+$s cannot solve classification problems where the partition formed by the decision boundaries results in a disconnected set for one class (path-disconnected regions in the input feature space). This is a generalized topological explanation of why DNN$^+$s cannot solve XOR.

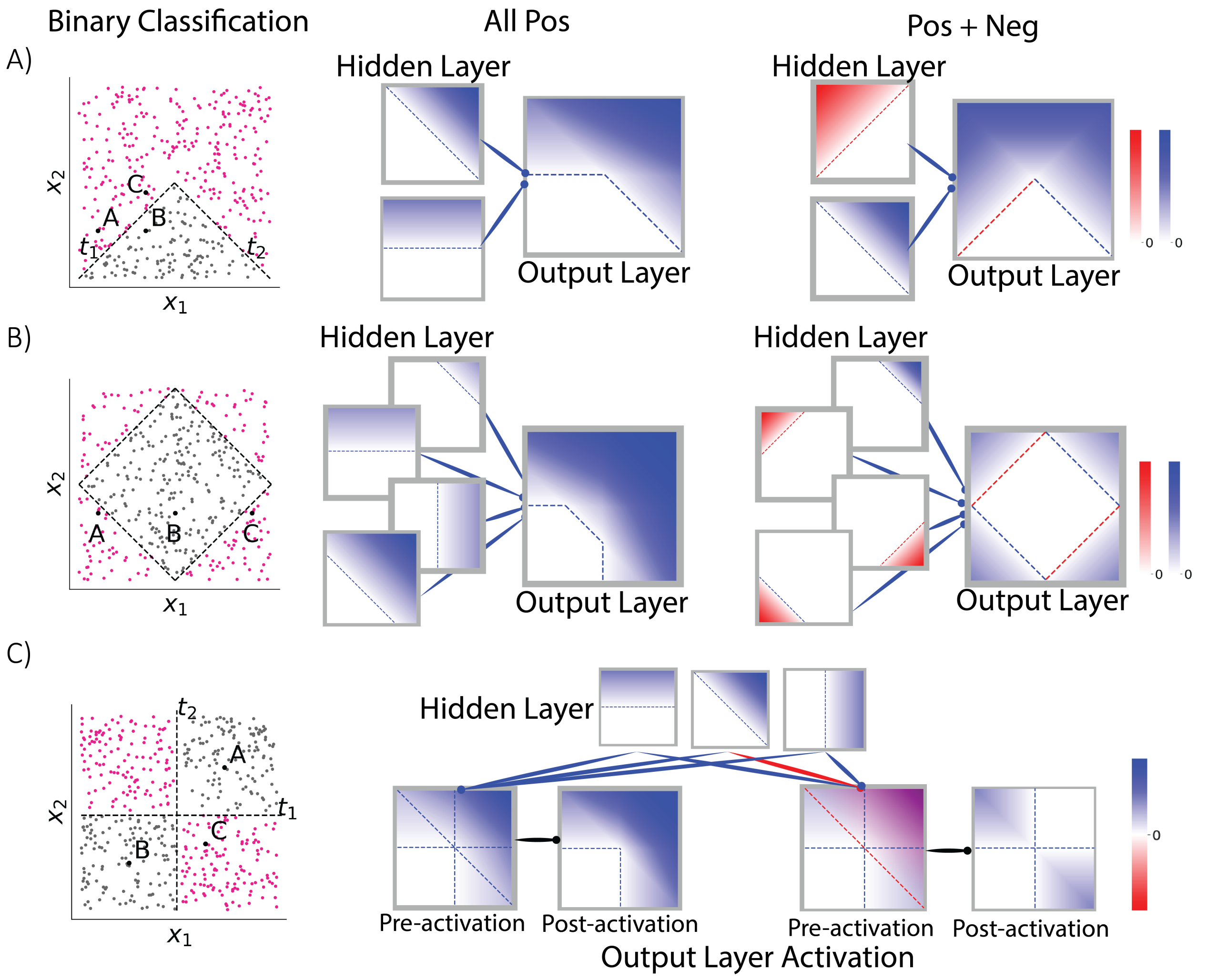

Figure 1: **DNNs with only non-negative weights (DNN$^+$s) are NOT universal approximators.** Three problems not solvable by DNN$^+$ are presented here. All red colors in the plots indicate changes brought by flipping some weights' polarities to negative such that the binary classification problems on the left can be solved by the DNN. A) DNN$^+$s cannot solve problems where the decision boundaries contain any segment of positive slope (Corollary 3.1.1). Take the 2-hidden-unit network as an example: by flipping a single input weight to negative (notice red decision boundaries), the problem on the left becomes solvable. B) DNN$^+$s cannot solve binary classification problems where there exists a decision boundary that forms a closed shape (Corollary 3.1.2). Take the 4-hidden-unit network as an example: flipping a single input weight for each of the two middle units allows decision boundaries of positive slopes to form, and further, flipping both input weights of the bottom unit allows the activation gradient to flow in the opposite direction (opposite to the order-preserving gradient, which is always to the top and/or right). These changes collectively make the closed-shape problem solvable. C) DNN$^+$s cannot solve binary classification problems where the partition formed by the decision boundaries results in a disconnected set for one class (Corollary 3.1.3), e.g., XOR (Theorem 3.3). By flipping a single output weight of a single hidden unit, the top right quadrant can be sculpted out of the pink class, making the network XOR-solvable.

4.1 A formal setup of decision boundary and partition of set

In section 4, we focus on binary classification problems with n input features. This means our target function is in the form $f : K \to \{0, 1\}$, where K is a compact domain and $K \subseteq \mathbb{R}^n$. The three corollaries can be easily extended into multi-class scenarios: whenever there exist two classes that satisfy the corollary statements, then the entire classification problem cannot be solved by DNN$^+$.

Instead of taking the angle of function approximation, in this section we set up classification problem $f : K \to \{0, 1\}$ as a partition P of compact domain K.

Definition 4.1 (Regions). *P is a partition of the compact domain K, where $P = \{R : R \subseteq K \subseteq \mathbb{R}^n\}$. We call the elements of P regions. A single class in the classification problem can be the union of one or multiple regions. Further, for P to be a partition of K, it satisfies these properties:*

1. *The family P does not contain the empty set, that is $\emptyset \notin P$;*
2. *The union of the regions is K, that is $\cup\{R\} = K$;*
3. *All regions are pair-wise disjoint, that is $\forall R_1, R_2 \in P,\ R_1 \neq R_2 \implies R_1 \cap R_2 = \emptyset$;*
4. *Within each region f is constant, i.e., $\forall R \in P, \forall x \in R, f(x) = c$ where c is a constant. For example, in our binary classification case, $c \in \{0, 1\}$;*
5. *Two adjacent regions belong to different classes.*

Definition 4.2 (Decision boundary). *For each region R in the partition $P = \{R : R \subseteq K\}$, the boundary of region R is given by*

$$\partial R = \{x \in R : \textit{for every neighborhood } O \textit{ of } x, O \cap R \neq \emptyset \text{ and } O \cap (K \backslash R) \neq \emptyset\}. \tag{9}$$

*The collection of the boundaries of all regions form the **decision boundaries** of the classification problem f, and is denoted by $\{\partial R\}$.*

Remark 4 (Neighborhood of the decision boundary). *Since two adjacent regions belong to different classes, then we must have points within the neighborhood of the decision boundary that belong to two different classes.*

Consider the ϵ-neighborhood of a point d on the decision boundary, i.e., for a point $d \in \partial R$, consider its ϵ-neighborhood $O = \{x \in K : |x - d| < \epsilon, \epsilon > 0\}$. Per the definition of boundary and the last property of region definition, we must have $\exists A, B \in O$, such that $f(A) \neq f(B)$.

Remark 5 (Piecewise linear approximation of the decision boundary). *Since we can approximate any function with piecewise linear functions up to some arbitrary accuracy [17], here we approximate the decision boundaries $\{\partial R\}$ with a family of linear functions $\mathbb{L} = \{L\}$, where $L = \{x \in K' \subseteq K : \mathbf{ax} + b = 0, \mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}\}$. $\{L\}$ are connected line segments in $\mathbb{R}^2$ or connected hyperplane sections in higher dimensions $\mathbb{R}^n$. We always have $L \subseteq \mathbb{R}^{n-1}$, and $\mathbf{a}$ is the normal to the hyperplane L. Within the ϵ region of the boundary, based on the last property of the partition $\{R\}$, we have the following:*

$$\begin{cases} f(x) = c_1, \forall x \in \{x \in K : 0 < \mathbf{ax} + b < \epsilon\} \\ f(x) = c_2, \forall x \in \{x \in K : -\epsilon < \mathbf{ax} + b < 0\} \end{cases}, c_1 \neq c_2, \epsilon > 0 \textit{ is small}^{3} \tag{10}$$

On a side note, decision boundaries are essentially the discontinuities of f where the outputs of f switch between 0 and 1. Although the original universal approximator theorem proofs require f to be continuous, readers should not be too concerned on the continuity of f for the following reason. We could easily relax the decision boundaries from line into a band where f gradually switches between 0 and 1 in this band. Such relaxation does not alter any of our conclusions in the following 3 corollaries as we can always limit the bandwidth to be smaller than ϵ so that all of the above definitions still hold.

[3] ϵ should be small such that x is within the immediately adjacent classes c_1 & c_2; it should also be *larger* than the linear approximation error region $|\mathcal{L} - \partial R|$. We assume in our paper the classification problems are well-behaved enough that they can be reasonably well approximated by piecewise linear functions and we can always find such ϵ.

4.2 Geometric intuitions of DNN$^+$

Corollary 3.1.1. (Boundary orientation, in $\mathbb{R}^n$). *DNNs with only non-negative weights cannot solve classification problems where the decision boundaries $\{L\}$ have any segment L with a normal $\mathbf{a} = (a_1, \ldots, a_n)$ where $\exists\, i \neq j \in [n],\ a_i a_j < 0$, i.e., positive slope in $\mathbb{R}^2$.*

See proof on page 13.

Corollary 3.1.2. (Closed shape, in $\mathbb{R}^n$) *DNNs with only non-negative weights cannot solve binary classification problems where there exists a regions R that is a closed set, i.e., the decision boundaries form a closed shape in $\mathbb{R}^2$.*

See proof on page 13.

Definition 4.3 (path-disconnected point pair)**.** *Suppose that $\mathcal{X}$ is a topological space. For a pair of points $x_1, x_2 \in \mathcal{X}$, x_1 and x_2 are path-disconnected if there does* not *exists a continuous function (path) $f : [0, 1] \mapsto \mathcal{X}$ where $f(0) = x_1, f(1) = x_2$.*

Definition 4.4 (disconnected space in $\mathbb{R}^2$)**.** *A space $\mathcal{X} \subset \mathbb{R}^2$ is path-disconnected if there exists $x_1, x_2 \in \mathcal{X}$ such that x_1 and x_2 are path-disconnected.* [4]

Corollary 3.1.3. (Disconnected space, in $\mathbb{R}^n$.) *DNNs with non-negative weights cannot solve a binary classification problem where there exists a class that is a disconnected space.*

See proof on page 14.

These three corollaries point to the fundamental flaw of DNN$^+$s: however many layers they have, each penultimate layer unit can only form a single continuous decision boundary that is composed of segments having negative slopes (or composed of hyperplanes having all-positive normal). Adding more layers or adding more nodes to each layer of such a network can produce more complex-shaped decision boundaries (Figure 1, panel B, middle column), but cannot form boundaries of more orientations, or form a closed region, or sculpt the input space such that disconnected regions can be joined. Therefore, taking negative weights away from DNNs drastically shrinks their repertoire of representable functions. For real-world problem solving, it is crucial to have both positive and negative weights in the network.

5 Extension to convolutional neural networks

In this section, we will discuss and prove that our theoretical results in section 3 and 4 are generalizable to many forms of DNN that are variants of the MLP definition (Def 3.1). We are particularly interested in the family of convolutional neural networks (CNNs) since their activity space closely resembles those observed in the visual cortex and auditory cortex in the brain, as shown by various correlation studies [18, 19].

Essentially, substituting matrix multiplication with **convolution** (Definition 5.1) and adding additional **max-pooling** layers (Definition 5.2) makes a CNN [20–22]. The universality of CNNs has been explicitly proven [23–26]. Additionally, one can further add **skip connections** (Definition 5.3) to make a residual network [27].

We will discuss how having convolution layers and adding skip connections make DNNs equivalent to our original MLP definition and thus subject to the same limited representation power when all weights are non-negative; we will also show our results are generalizable to DNNs with additional max-pooling layers since the order-preserving property of DNN$^+$ holds in general.

5.1 Convolution layer

The convolution operation between an input (normally an image) and a single filter is defined below. For the convolutional layers in CNN, instead of matrix multiplication, multiple filters convolve the input, and the values in the filters are the weights in CNN. It can be shown that all convolution operations can be converted into matrix multiplications [28], where all the filters are converted into the form of Toeplitz type matrices and all images are

[4]An example of disconnected space in $\mathbb{R}^2$ is the second and fourth quadrant that form class 0 in XOR.

vectorized. For more details, we refer readers to the universality proofs of CNN [23]. Therfore, a convolutional neural network can be converted into a MLP (Definition 3.1). Therefore, for a convolutional neural network with all non-negative weights, all of our results in this paper hold.

Definition 5.1 (Convolution 2D)**.** *The convolution* $O : \mathbb{R}^{M\times N} \times \mathbb{R}^{m\times n} \rightarrow \mathbb{R}^{(M-m+1)\times(N-n+1)}$ *between an input (image)* $I \in \mathbb{R}^{M\times N}$ *and a kernel (feature map)* $K \in \mathbb{R}^{m\times n}$ *is given by*

$$O(i,j) = \sum_{k=1}^{m}\sum_{l=1}^{n} I(i+k-1, j+l-1)K(k,l), i \in [M-m+1],\ j \in [N-n+1] \tag{11}$$

5.2 Max pooling layer

Since max pooling functions are order-preserving monotone functions as well, by the closure of monotone functions under compositionality, we further have DNNs with max pooling layers follow all the results in the previous sections.

Definition 5.2 (Max-pooling)**.** *Max-pooling functions are* $pool : \mathbb{R}^n \rightarrow \mathbb{R},\ pool(\mathbf{x}) = max(x_i)$*, where* $\mathbf{x} = (x_1, \dots, x_i, \dots, x_n)$*.*

Lemma 5.1 (Max pooling are order preserving)**.** *Max-pooling functions are order-preserving monotonone functions. Equivalently,*

$$\mathbf{x} \preceq \mathbf{y} \implies pool(\mathbf{x}) \leqslant pool(\mathbf{y}) \tag{12}$$

Proof. Without loss of generality, for any pair of points $\mathbf{x} = (x_1, \dots, x_i, \dots, x_n), \mathbf{y} = (y_1, \dots, y_i, \dots, y_n) \in \mathbb{R}^n$ that follow the order $\mathbf{x} \preceq \mathbf{y}$, assume their max pooling output is given by the pth and qth dimensions, respectively. Then we have

$$\forall i \in [n], x_i \leqslant x_p$$
$$\forall j \in [n], y_j \leqslant y_q.$$

Also by the ordering $\mathbf{x} \preceq \mathbf{y}$, we have $y_p \geqslant x_p$, which means

$$pool(\mathbf{x}) = x_p \leqslant y_p \leqslant y_q = pool(\mathbf{y}). \tag{13}$$

Thus max pooling functions are order-preserving monotone functions. □

5.3 Skip connections

A skip connection connects two non-adjacent layers, e.g., layers $(l-2)$ and l, through identity mapping (Definition 5.3).

Definition 5.3 (Skip connections)**.** *Each layer with an incoming skip connection is given by* $f^{(l)} : \mathbb{R}^{n_{(l-1)}} \rightarrow \mathbb{R}^{n_l}$*:*

$$f^{(l)}(\mathbf{x}) = \mathbf{\Phi}(\mathbf{W}^{(l)}\mathbf{x} + \mathbf{b}^{(l)})\underline{+f^{(l')}(\mathbf{x})}, \quad l' \in \{0, \dots, l-2\}$$

We can always convert a DNN with skip connections into a classical MLP (Definition 3.1) by adding dummy units in the intermediate layers (in $(l-1)$ and l) and setting their incoming and outgoing weights to match the identity mapping of the skip connection, i.e., weights of each unit only having a single 1 entry where the skip connection is and 0 elsewhere. Therefore, for a DNN with skip connections, all of our results in this paper hold.

6 Conclusion and Discussion

We proved that DNNs with all non-negative weights (i.e., without inhibitory connections) are not universal approximators (Theorem 3.1). This is because non-negative DNNs are exclusively order-preserving monotone functions (Theorem 3.1). Some geometric implications of this property in the finite euclidean space $\mathbb{R}^n$ are proved in Corollaries 3.1.1-3.1.3. Specifically, each output unit in a network without inhibitory connections can only form a single continuous decision boundary that is composed of hyperplane sections having a very particular orientation (hyperplane with all-positive normal). Intuitively in $\mathbb{R}^2$ input space, this means only forming a continuous line composed of segments having negative slopes. The addition of inhibitory connections to the networks allows more complex boundaries to form (e.g., boundaries of positive orientations (Corollary 3.1.1) and of closed shapes (Corollary 3.1.2)); the addition of inhibition also allows for sculpting/folding of the representation space (Corollary 3.1.3). Together, these results prove that both DNNs and brains, which can be abstracted as networks with non-decreasing monotone activation functions, need inhibitory connections to learn more functions.

How translatable are our theoretical results on DNNs to brains? Under the assumption that the activation functions of all neurons are non-decreasing, our theoretical results directly shed light onto the long-standing question of why brains have inhibitory connections. We recognize there are types of questions that cannot be answered by DNN theory. For example, the learning process of DNNs differs from biological systems [29], DNNs do not follow Dale's principle [30], units in DNNs do not spike, etc. We emphasize our proof does *not* rely on any assumption that involves the above discrepancies between DNNs and brains; instead, our proof only relies on the non-decreasing activation function assumption. The simplicity of our assumption is the sole reason behind why our results in general hold for many forms of DNN that have been experimentally shown to resemble the brain 5. Our answer from a representation-capacity point of view supplements the dynamic system story of E/I balance and provides new perspectives to these long-standing neuroscience questions.

Our current work is just a first step in understanding the representation space of networks from the lens of connection polarity. What we have proven is a first-order property of the network concerning the *existence* of negative connections. The next step is to look at the second-order property that concerns the *configuration* of connection polarities. For this second-order property, a strict non-negativity constraint is no longer imposed on the network weights; instead, we constrain the connections to follow a specific polarity configuration, or in the neuroscience language - a circuit rule. We will illustrate the feasibility of this idea and its significance with an example: in the cortex, excitatory and inhibitory neurons connect / synapse in very different manners, and they follow a very stereotypical pattern: *local excitation, broad inhibition*. Such a configuration principle has been suggested to be the underlying mechanism of surround inhibition [31, 32], an important computation process that allows for context-dependent activation and redundancy reduction. Based on our Corollaries 3.1.1-3.1.3, it is very likely that in a local subspace, the largely excitatory sub-network is order-preserving monotone and only forms continuous decision boundaries of negative slopes; on a global level, such communities of subspaces are connected through inhibitory connections and collectively form complex decision boundaries. It is an exciting future direction to connect these geometric properties with the relatively better-understood functional significance of neural circuits. Similar ideas can be explored in the canonical circuits in decision making [33] (e.g., winner take all). How circuit rules translate into the geometric constraints on the representation space has been largely unexplored. It was a route more regularly taken in the earliest era of DNNs (perceptrons, specifically) [34] and became increasingly more challenging as the size of the network grew; we look forward to building on top of the theoretical work presented in this paper and bridging the gap between network topology (E-I configuration) and the associated representation space.

Acknowledgements

This work is supported by grants awarded to J.T.V: NSF CAREER Award (Grant No. 1942963), NSF NeuroNex Award (Grant No. 2014862), NSF AI Institute Planning Award (Grant No. 2020312), and NSF Collaborative Research: THEORI Networks (Grant No. 2031985). Q.W. was also partially supported by Johns Hopkins neuroscience T32 training grant (Grant no. 091018). The authors would like to thank the NeuroData lab members for helpful feedback.

References and Notes

[1] Sukbin Lim and Mark S. Goldman. Balanced cortical microcircuitry for maintaining information in working memory. *Nature Neuroscience*, 16(9):1306–1314, 2013. 1

[2] Norman H. Lam, Thiago Borduqui, Jaime Hallak, Antonio Roque, Alan Anticevic, John H. Krystal, Xiao Jing Wang, and John D. Murray. Effects of altered excitation-inhibition balance on decision making in a cortical circuit model. *Journal of Neuroscience*, 42, 2022. 1

[3] T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science*, 334(6062):1569–1573, dec 2011. 1

[4] Nirit Sukenik, Oleg Vinogradov, Eyal Weinreb, Menahem Segal, Anna Levina, and Elisha Moses. Neuronal circuits overcome imbalance in excitation and inhibition by adjusting connection numbers. *Proceedings of the National Academy of Sciences of the United States of America*, 118, 2021. 1

[5] Ivan Cohen, Vincent Navarro, Stéphane Clemenceau, Michel Baulac, and Richard Miles. On the origin of interictal activity in human temporal lobe epilepsy in vitro. *Science*, 298(5597):1418–1421, nov 2002. 1

[6] Gilles Huberfeld, Liset Menendez De La Prida, Johan Pallud, Ivan Cohen, Michel Le Van Quyen, Claude Adam, Stéphane Clemenceau, Michel Baulac, and Richard Miles. Glutamatergic pre-ictal discharges emerge at the transition to seizure in human epilepsy. *Nature Neuroscience*, 14(5):627–635, 2011.

[7] Wilson Truccolo, Jacob A. Donoghue, Leigh R. Hochberg, Emad N. Eskandar, Joseph R. Madsen, William S. Anderson, Emery N. Brown, Eric Halgren, and Sydney S. Cash. Single-neuron dynamics in human focal epilepsy. *Nature Neuroscience*, 14(5):635–643, 2011. 1

[8] William B. Kristan. Early evolution of neurons. *Current Biology*, 26, 2016. 1

[9] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, jan 1989. 1, 2, 3, 4

[10] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems 1989 2:4*, 2(4):303–314, dec 1989.

[11] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, jan 1991. 1, 2, 3, 4

[12] F. M. Dickey and J. M. DeLaurentis. Optical neural networks with unipolar weights. *Optics Communications*, 101, 1993. 2

[13] J. M. DeLaurentis and F. M. Dickey. A convexity-based analysis of neural networks. *Neural Networks*, 7, 1994. 2

[14] Hennie Daniels and Marina Velikova. Monotone and partially monotone neural networks. *IEEE Transactions on Neural Networks*, 21(6):906–917, 2010. 2

[15] Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 146–155. PMLR, 06–11 Aug 2017. 2

[16] Britt Anderson, Jessie J. Peissig, Jedediah Singer, and David L. Sheinberg. Xor style tasks for testing visual object processing in monkeys. *Vision Research*, 46, 2006. 5

[17] Changcun Huang. ReLU Networks Are Universal Approximators via Piecewise Linear or Constant Functions. *Neural Computation*, 32(11):2249–2278, 11 2020. 7

[18] Pouya Bashivan, Kohitij Kar, and James J. DiCarlo. Neural population control via deep image synthesis. *Science*, 364, 2019. 8

[19] Alexander J.E. Kell, Daniel L.K. Yamins, Erica N. Shook, Sam V. Norman-Haignere, and Josh H. McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98:630–644.e16, 2018. 8

[20] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, 12 1989. 8

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 8

[23] Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, 2020. 8, 9

[24] Nadav Cohen and Amnon Shashua. Convolutional rectifier networks as generalized tensor decompositions. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 955–963, New York, New York, USA, 20–22 Jun 2016. PMLR.

[25] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14, 2017.

[26] Andreas Heinecke, Jinn Ho, and Wen-Liang Hwang. Refinement and universal approximation via sparsely connected relu convolution nets. *IEEE Signal Processing Letters*, 27:1175–1179, 2020. 8

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 8

[28] Kumar Chellapilla, Sidd Puri, and Patrice Simard. High Performance Convolutional Neural Networks for Document Processing. In Guy Lorette, editor, *Tenth International Workshop on Frontiers in Handwriting Recognition*, La Baule (France), October 2006. Université de Rennes 1, Suvisoft. http://www.suvisoft.com. 8

[29] Timothy P. Lillicrap, Daniel Cownden, Douglas B. Tweed, and Colin J. Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7:1–10, 2016. 10

[30] Jonathan Cornford, Damjan Kalajdzievski, Marco Leite, Amélie Lamarquette, Dimitri M Kullmann, Blake Richards, and Mila / Mcgill. Learning to live with dale's principle: Anns with separate excitatory and inhibitory units. *bioRxiv*, 2020. 10

[31] Alessandra Angelucci, Maryam Bijanzadeh, Lauri Nurminen, Frederick Federer, Sam Merlin, and Paul C. Bressloff. Circuits and mechanisms for surround modulation in visual cortex. *Annual Review of Neuroscience*, 40:425–451, 2017. 10

[32] Hirofumi Ozeki, Ian M. Finn, Evan S. Schaffer, Kenneth D. Miller, and David Ferster. Inhibitory stabilization of the cortical network underlies visual surround suppression. *Neuron*, 62:578–592, 2009. 10

[33] Shreesh P. Mysore and Ninad B. Kothari. Mechanisms of competitive selection: A canonical neural circuit framework. *eLife*, 9, 2020. 10

[34] Marvin Minsky and Seymour Papert. *Perceptrons.* M.I.T. Press, 1969. 10

A Appendix - Proofs

Proof of Lemma 3.1. By definition 3.3, it suffices to prove $\forall i \in [n],\ T_i(\mathbf{x}) \leqslant T_i(\mathbf{y})$. Let w_{ij} be the ith row and jth column component of $\mathbf{W}$:

$$\begin{aligned} T_i(\mathbf{x}) - T_i(\mathbf{y}) &= (\sum_{j\in[m]} w_{ij}x_j + b_i) - (\sum_{j\in[m]} w_{ij}y_j + b_i) \\ &= \sum_{j\in[m]} w_{ij}(x_j - y_j) \\ &\leqslant 0 \end{aligned}$$

□

Proof of Theorem 3.1. DNN$^+$ is a composition of monotone functions: layers of non-decreasing activation functions $\mathbf{\Phi}$ and affine transformation with all non-negative weights $T : \mathbb{R}^m \to \mathbb{R}^n$ (Lemma 3.1). By closure of monotone function under compositionality, we have that DNN$^+$ is an order-preserving monotone function. □

Proof of Corollary 3.1.1. Without loss of generality, we assume the feature index pair $i, j \in [n]$ satisfy $a_i a_j < 0$. Let $\mathbf{d} = (d_1, \ldots, d_i, \ldots, d_j, \ldots, d_n)$ be a point on the segment, i.e., $\mathbf{ad} + b = 0$. Now we construct three points A, B, C with $\epsilon > 0$.[5]

$$A = (d_1, \ldots, \quad d_i - \epsilon, \ldots, d_j, \quad \ldots, d_n) \tag{14}$$
$$B = (d_1, \ldots, \quad d_i + \epsilon, \ldots, d_j, \quad \ldots, d_n) \tag{15}$$
$$C = (d_1, \ldots, \quad d_i + \epsilon, \ldots, d_j - 2\frac{a_i}{a_j}\epsilon, \ldots, d_n) \tag{16}$$

First since $a_i a_j < 0$ and $\epsilon > 0$, we have $2\frac{a_i}{a_j}\epsilon < 0$, thus

$$\begin{cases} x_i^A < x_i^B = x_i^C \\ x_j^A = x_j^B < x_j^C \end{cases} \implies A \preceq B \preceq C \implies F^+(A) \leq F^+(B) \leq F^+(C) \tag{17}$$

Simultaneously, from $\mathbf{ad} + b = 0$, we also have

$$\begin{cases} \mathbf{a}A + b = \mathbf{ad} + b - a_i\epsilon &= -a_i\epsilon \\ \mathbf{a}B + b = \mathbf{ad} + b + a_i\epsilon &= \quad a_i\epsilon \\ \mathbf{a}C + b = \mathbf{ad} + b + a_i\epsilon - 2\frac{a_i}{a_j}a_j\epsilon &= -a_i\epsilon \end{cases}$$

Then A, C must lie on the same side of L but different than B, thus $f(A) = f(C) \neq f(B)$. By the same logic as in Theorem 3.3, this contradicts with Eq 17; therefore, DNN$^+$ cannot solve classification problems where the decision boundaries $\{L\}$ have any segment L with a normal $\mathbf{a} = (a_1, \ldots, a_n)$ where $\exists\, i \neq j \in [n], a_i a_j < 0$. □

Proof of Corollary 3.1.2. Without loss of generality, let's assume region $R_0 \in \{R\}$ is a closed set. We denote all points in R_0 as a general form $\mathbf{x} = (x_1, \ldots, x_n)$. Consider any point $B = (x_1^B, \ldots, x_n^B) \in R_0$, we can always find two points $A', C' \in \partial R$ that follow $A' \preceq B \preceq C'$ with the following construction method:

$$A' = (min(x_1), x_2^B, \ldots, x_n^B)$$
$$C' = (max(x_1), x_2^B, \ldots, x_n^B)$$

[5]For the choice of ϵ under non-linear decision boundary scenario, we assume here we can always find ϵ such that it is larger than the linear approximation error. For more details, please see Remark 5

As we move $\epsilon > 0$ away from the boundary, we can further construct two points $A, C \notin R_0$ where

$$A = (min(x_1) - \epsilon, x_2^B, \dots, x_n^B)$$
$$C = (max(x_1) + \epsilon, x_2^B, \dots, x_n^B)$$

Thus $A \preceq B \preceq C \implies F^+(A) \leqslant F^+(B) \leqslant F^+(C)$, yet we have $B \in R_0$ while $A, C \notin R_0 \implies f(A) = f(C) \neq f(B)$. By the same reasoning as in theorem 3.3, we have a contradiction thus a DNN$^+$ cannot solve problems where the decision boundary forms a closed set. □

Proof of Corollary 3.1.3. Without loss of generality, we assume R_0, R_1 are disconnected and belong to the same class, i.e., $f(x) = c_1, \forall x \in R_0 \cup R_1$, c_1 is a constant. Now consider any pair of points $(A, B), A \in R_0, B \in R_1$, the straight line segment AB that connects A and B must pass through another class by Definition 4.4. This means we must have point C on line AB, but $f(c) = c_2 \neq c_1$ where c_2 is a constant. Next, we discuss the order relationship between A and B:

case 1 **Exists such a pair $\mathbf{A \preceq B}$.** Then since A, C, B are colinear and C is in between A and B, we have $A \preceq C \preceq B \implies F^+(A) \leqslant F^+(C) \leqslant F^+(B)$. Yet by construction, we also have $f(A) = f(B) \neq f(C)$. By the same reasoning as in Theorem 3.3, we have a contradiction. Thus, a DNN$^+$ cannot solve classification problems that fall into this case.

case 2 **Does NOT exist such a pair $\mathbf{A \preceq B}$.** This means for all pairs of points in the two disconneted regions, they don't follow the ordering defined in Definition 3.3. Thus, there exists two input dimensions, $i, j \in [n]$, such that for all $\mathbf{x^0} = (\dots, x_i^0, \dots, x_j^0, \dots) \in R_0$, and for all $\mathbf{x^1} = (\dots, x_i^1, \dots, x_j^1, \dots) \in R_1$, they follow

$$\begin{cases} x_i^0 < t_i < x_i^1 \\ x_j^0 > t_j > x_j^1 \end{cases}, t_1, t_2 \in \mathbb{R} \tag{18}$$

Now, for a point $G = (\dots, x_i^F, \dots, x_j^F, \dots) \in R_0$, we always have $x_i^G < t_i \text{ and } x_j^G > t_j$. Further, we can always construct two more points $D, E \in (K - R_0 - R - 1)$ by

$$D = (\dots, x_i^G, \dots, t_j, \dots)$$
$$E = (\dots, t_i, \dots, x_j^G, \dots)$$

Thus we have $D \preceq G \preceq E \implies F^+(D) \leqslant F^+(G) \leqslant F^+(E)$. However, since we have $F \in R_0$ and yet $D, E \in (K - R_0 - R - 1)$, we therefore have $f(D) = f(E) \neq f(G)$. By the same reasoning as in theorem 3.3, we have a contradiction. Thus, a DNN$^+$ cannot solve classification problems that fall into this case.

Collectively, we proved that a DNN$^+$ cannot solve a classification problem where there exists a class that is a disconnected space. □