ARTICLE Communicated by Hrushikesh Mhaskar

On PDE Characterization of Smooth Hierarchical Functions Computed by Neural Networks

Khashayar Filom
filom@umich.edu
Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, U.S.A.

Roozbeh Farhoodi
roozbeh@seas.upenn.edu
Konrad Paul Kording
kording@upenn.edu
Departments of Bioengineering and Department of Neuroscience, University of Pennsylvania, Philadelphia, PA 1910, U.S.A.

Neural networks are versatile tools for computation, having the ability to approximate a broad range of functions. An important problem in the theory of deep neural networks is expressivity; that is, we want to understand the functions that are computable by a given network. We study real, infinitely differentiable (smooth) hierarchical functions implemented by feedforward neural networks via composing simpler functions in two cases: (1) each constituent function of the composition has fewer inputs than the resulting function and (2) constituent functions are in the more specific yet prevalent form of a nonlinear univariate function (e.g., tanh) applied to a linear multivariate function. We establish that in each of these regimes, there exist nontrivial algebraic partial differential equations (PDEs) that are satisfied by the computed functions. These PDEs are purely in terms of the partial derivatives and are dependent only on the topology of the network. Conversely, we conjecture that such PDE constraints, once accompanied by appropriate nonsingularity conditions and perhaps certain inequalities involving partial derivatives, guarantee that the smooth function under consideration can be represented by the network. The conjecture is verified in numerous examples, including the case of tree architectures, which are of neuroscientific interest. Our approach is a step toward formulating an algebraic description of functional spaces associated with specific neural networks, and may provide useful new tools for constructing neural networks.

Neural Computation **33, 3204–3263** (2021)
https://doi.org/10.1162/neco_a_01441

1 Introduction

1.1 Motivation. A central problem in the theory of deep neural networks is to understand the functions that can be computed by a particular architecture (Raghu, Poole, Kleinberg, Ganguli, & Dickstein, 2017; Poggio, Banburski, & Liao, 2019). Such functions are typically superpositions of simpler functions, that is, compositions of functions of fewer variables. This article aims to study superpositions of real smooth (i.e., infinitely differentiable or C^∞) functions that are constructed hierarchically (see Figure 3). Our core thesis is that such functions (also referred to as *hierarchical* or *compositional* interchangeably) are constrained in the sense that they satisfy certain **p**artial **d**ifferential **e**quations (**PDE**s). These PDEs are dependent only on the topology of the network and could be employed to characterize smooth functions computable by a given network.

1.1.1 Example 1. One of the simplest examples of a superposition is when a trivariate function is obtained from composing two bivariate functions; for instance, let us consider the composition

$$F(x, y, z) = g\,(f(x, y), z) \tag{1.1}$$

of functions $f = f(x, y)$ and $g = g(u, z)$ that can be computed by the network in Figure 1. Assuming that all functions appearing here are twice continuously differentiable (or C^2), the chain rule yields

$$F_x = g_u f_x, \quad F_y = g_u f_y.$$

If either F_x or F_y – say the former – is nonzero, the equations above imply that the ratio between F_x and F_y is independent of z:

$$\frac{F_y}{F_x} = \frac{f_y}{f_x}. \tag{1.2}$$

Therefore, its derivative with respect to z must be identically zero:

$$\left(\frac{F_y}{F_x}\right)_z = \frac{F_{yz}F_x - F_{xz}F_y}{(F_x)^2} = 0. \tag{1.3}$$

This amounts to

$$F_{yz}F_x = F_{xz}F_y, \tag{1.4}$$

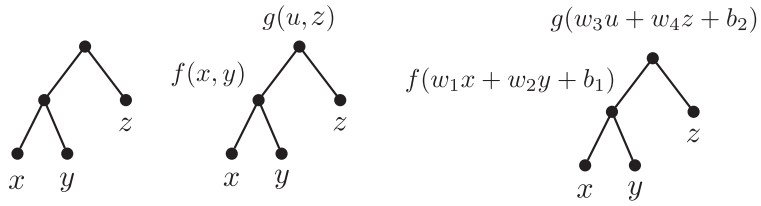

Figure 1: The architecture on the left (studied in example 1) can compute functions of the form $g(f(x, y), z)$ as in the middle. They involve the smaller class of functions of the form $g(w_3 f(w_1 x + w_2 y + b_1) + w_4 z + b_2)$ on the right.

an equation that always holds for functions of form 1.1. Notice that one may readily exhibit functions that do not satisfy the necessary PDE constraint $F_{xz}F_y = F_{yz}F_x$ and so cannot be brought into form 1.1, for example,

$$xyz + x + y + z. \tag{1.5}$$

Conversely, if the constraint $F_{yz}F_x = F_{xz}F_y$ is satisfied and F_x (or F_y) is nonzero, we can reverse this processes to obtain a local expression of the form 1.1 for $F(x, y, z)$. By interpreting the constraint as the independence of $\frac{F_x}{F_y}$ of z, one can devise a function $f = f(x, y)$ whose ratio of partial derivatives coincides with $\frac{F_x}{F_y}$ (this is a calculus fact; see theorem 5). Now that equation 1.2 is satisfied, the gradient of F may be written as

$$\nabla F = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \frac{F_x}{f_x} \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} + F_z \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

that is, as a linear combination of gradients of $f(x, y)$ and z. This guarantees that $F(x, y, z)$ is (at least locally) a function of the latter two (see the discussion at the beginning of section 3). So there exists a bivariate function g defined on a suitable domain with $F(x, y, z) = g(f(x, y), z)$. Later in the article, we generalize this toy example to a characterization of superpositions computed by *tree architectures* (see theorem 3).

Functions appearing in the context of neural networks are more specific than a general superposition such as equation 1.1; they are predominantly constructed by composing univariate nonlinear activation functions and multivariate linear functions defined by weights and biases. In the case of a trivariate function $F(x, y, z)$, we should replace the representation $g(f(x, y), z)$ studied so far with

$$F(x, y, z) = g(w_3 f(w_1 x + w_2 y + b_1) + w_4 z + b_2). \tag{1.6}$$

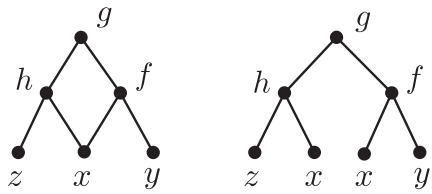

Figure 2: Implementations of superpositions of the form $F(x, y, z) = g(f(x, y), h(x, z))$ (studied in examples 2 and 7) by three-layer neural networks.

Assuming that activation functions f and g are differentiable, now new constraints of the form 1.3 are imposed. The ratio $\frac{F_y}{F_x}$ is equal to $\frac{w_2}{w_1}$, hence it is not only independent of z as equation 1.3 suggests, but indeed a constant function. So we arrive at

$$\left(\frac{F_y}{F_x}\right)_x = \left(\frac{F_y}{F_x}\right)_y = \left(\frac{F_y}{F_x}\right)_z = 0,$$

or, equivalently,

$$F_{xy}F_x = F_{xx}F_y, \quad F_{yy}F_x = F_{xy}F_y, \quad F_{yz}F_x = F_{xz}F_y.$$

Again, these equations characterize differentiable functions of the form 1.6; this is a special case of theorem 7 below.

1.1.2 Example 2. The preceding example dealt with compositions of functions with disjoint sets of variables and this facilitated our calculations. But this is not the case for compositions constructed by most neural networks, for example, networks may be fully connected or may have repeated inputs. For instance, let us consider a superposition of the form

$$F(x, y, z) = g(f(x, y), h(x, z)) \tag{1.7}$$

of functions $f(x, y)$, $h(x, z)$, and $g(u, v)$ as implemented in Figure 2. Applying the chain rule tends to be more complicated than the case of equation 1.1 and results in identities

$$F_x = g_u f_x + g_v h_x, \quad F_y = g_u f_y, \quad F_z = g_v h_z. \tag{1.8}$$

Nevertheless, it is not hard to see that there are again (perhaps cumbersome) nontrivial PDE constraints imposed on the hierarchical function F, a

fact that will be established generally in theorem 1. To elaborate, notice that identities in equation 1.8 together imply

$$F_x = A(x, y)F_y + B(x, z)F_z, \tag{1.9}$$

where $A := \frac{f_x}{f_y}$ and $B := \frac{h_x}{h_z}$ are independent of z and y, respectively. Repeatedly differentiating this identity (if possible) with respect to y, z results in linear dependence relations between partial derivatives of F (and hence PDEs) since the number of partial derivatives of F_x of order at most n with respect to y, z grows quadratically with n, while on the right-hand side, the number of possibilities for coefficients (partial derivatives of A and B with respect to y and z, respectively) grows only linearly. Such dependencies could be encoded by the vanishing of determinants of suitable matrices formed by partial derivatives of F. In example 7, by pursuing the strategy just mentioned, we complete this treatment of superpositions 1.7 by deriving the corresponding characteristic PDEs that are necessary and (in a sense) sufficient conditions on F that it be in the form of equation 1.7. Moreover, in order to be able to differentiate several times, we shall assume that all functions are smooth (or C^∞) hereafter.

1.2 Statements of Main Results. Fixing a neural network hierarchy for composing functions, we shall prove that once the constituent functions of corresponding superpositions have fewer inputs (lower arity), there exist universal **algebraic p**artial **d**ifferential **e**quations (**algebraic PDE**s) that have these superpositions as their solutions. A conjecture, which we verify in several cases, states that such PDE constraints characterize a generic smooth superposition computable by the network. Here, genericity means a nonvanishing condition imposed on an algebraic expression of partial derivatives. Such a condition has already occurred in example 1 where in the proof of the sufficiency of equation 1.4 for the existence of a representation of the form 1.1 for a function $F(x, y, z)$, we assumed either F_x or F_y is nonzero. Before proceeding with the statements of main results, we formally define some of the terms that have appeared so far.

Terminology

- We take all neural networks to be feedforward. A **feedforward neural network** is an acyclic hierarchical layer to layer scheme of computation. We also include **res**idual **net**works (**ResNet**s) in this category: an identity function in a layer could be interpreted as a jump in layers. Tree architectures are recurring examples of this kind. We shall always assume that in the first layer, the inputs are labeled by (not necessarily distinct) labels chosen from coordinate functions $x_1, \ldots, x_n$, and there is only one node in the output layer. Assigning functions to nodes in layers above the input layer implements a real scalar-valued

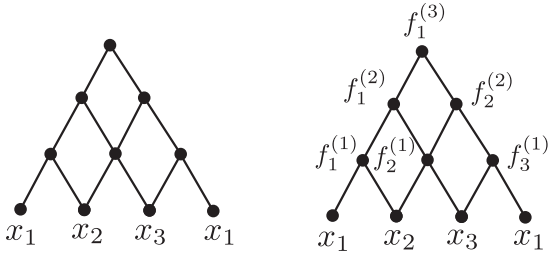

Figure 3: The neural network on the left can compute the hierarchical function $F(x_1, x_2, x_3) = f_1^{(3)}\Big(f_1^{(2)}\Big(f_1^{(1)}(x_1, x_2),\ f_2^{(1)}(x_2, x_3)\Big),\ f_2^{(2)}\Big(f_2^{(1)}(x_2, x_3),\ f_3^{(1)}(x_3, x_1)\Big)\Big)$ once appropriate functions are assigned to its nodes as on the right.

function $F = F(x_1, \ldots, x_n)$ as the superposition of functions appearing at nodes (see Figure 3).

- In our setting, an **algebraic PDE** is a nontrivial polynomial relation such as

$$\Phi\left(F_{x_1}, \ldots, F_{x_n}, F_{x_1^2}, F_{x_1 x_2}, \ldots, F_{\mathbf{x}^\alpha}, \ldots\right) = 0 \tag{1.10}$$

among the partial derivatives (up to a certain order) of a smooth function $F = F(x_1, \ldots, x_n)$. Here, for a tuple $\alpha := (\alpha_1, \ldots, \alpha_n)$ of nonnegative integers, the partial derivative $\frac{\partial^{\alpha_1+\cdots+\alpha_n} F}{\partial x_1^{\alpha_1} \ldots \partial x_n^{\alpha_n}}$ (which is of order $|\alpha| := \alpha_1 + \cdots + \alpha_n$) is denoted by $F_{\mathbf{x}^\alpha}$. For instance, asking for a polynomial expression of partial derivatives of F to be constant amounts to n algebraic PDEs given by setting the first-order partial derivatives of that expression with respect to $x_1, \ldots, x_n$ to be zero.

- A **nonvanishing condition** imposed on smooth functions $F = F(x_1, \ldots, x_n)$ is asking for these functions not to satisfy a particular algebraic PDE, namely,

$$\Psi\left(F_{x_1}, \ldots, F_{x_n}, F_{x_1^2}, F_{x_1 x_2}, \ldots, F_{\mathbf{x}^\alpha}, \ldots\right) \neq 0, \tag{1.11}$$

for a nonconstant polynomial Ψ. Such a condition could be deemed pointwise since if it holds at a point $\mathbf{p} \in \mathbb{R}^n$, it persists throughout a small enough neighborhood. Moreover, equation 1.11 determines an open dense subset of the functional space; so, it is satisfied generically.

Theorem 1. *Let $\mathcal{N}$ be a feedforward neural network in which the number of inputs to each node is less than the total number of distinct inputs to the network. Superpositions of smooth functions computed by this network satisfy nontrivial*

constraints in the form of certain algebraic PDEs that are dependent only on the topology of $\mathcal{N}$.

In the context of deep learning, the functions applied at each node are in the form of

$$\mathbf{y} \mapsto \sigma\left(\langle \mathbf{w}, \mathbf{y} \rangle\right); \tag{1.12}$$

that is, they are obtained by applying an activation function σ to a linear functional $\mathbf{y} \mapsto \langle \mathbf{w}, \mathbf{y} \rangle$. Here, as usual, the bias term is absorbed into the weight vector. The bias term could also be excluded via composing σ with a translation since throughout our discussion, the only requirement for a function σ to be the activation function of a node is smoothness, and activation functions are allowed to vary from a node to another. In our setting, σ in equation 1.12 could be a polynomial or a sigmoidal function such as hyperbolic tangent or logistic functions, but not ReLU or maxout activation functions. We shall study functions computable by neural networks as either superpositions of arbitrary smooth functions or as superpositions of functions of the form 1.12, which is a more limited regime. Indeed, the question of how well arbitrary compositional functions, which are the subject of theorem 1, may be approximated by a deep network has been studied in the literature (Mhaskar, Liao, & Poggio, 2017; Poggio, Mhaskar, Rosasco, Miranda, & Liao, 2017).

In order to guarantee the existence of PDE constraints for superpositions, theorem 1 assumes a condition on the topology of the network. However, theorem 2 states that by restricting the functions that can appear in the superposition, one can still obtain PDE constraints even for a fully connected multilayer perceptron:

Theorem 2. *Let $\mathcal{N}$ be an arbitrary feedforward neural network with at least two distinct inputs, with smooth functions of the form 1.12 applied at its nodes. Any function computed by this network satisfies nontrivial constraints in the form of certain algebraic PDEs that are dependent only on the topology of $\mathcal{N}$.*

1.2.1 Example 3. As the simplest example of PDE constraints imposed on compositions of functions of the form 1.12, recall that d'Alembert's solution to the wave equation,

$$u_{tt} = c^2 u_{xx}, \tag{1.13}$$

is famously given by superpositions of the form $f(x + ct) + g(x - ct)$. This function can be implemented by a network with two inputs x, t and with one hidden layer in which the activation functions f, g are applied (see Figure 4). Since we wish for a PDE that works for this architecture universally, we should get rid of c. The PDE 1.13 may be written as $\frac{u_{tt}}{u_{xx}} = c^2$; that is

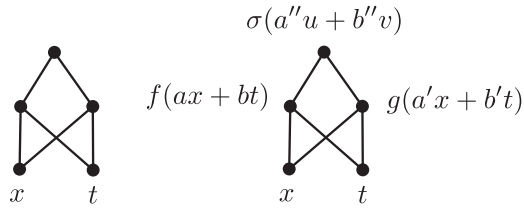

Figure 4: The neural network on the left can compute the function $F(x,t) = \sigma(a''f(ax+bt) + b''g(a'x+b't))$ once, as on the right, the activation functions σ, f, g and appropriate weights are assigned to the nodes. Such functions are the subject of examples 3 and 11.

the ratio $\frac{u_{tt}}{u_{xx}}$ must be constant. Hence, for our purposes, the wave equation should be written as $\left(\frac{u_{tt}}{u_{xx}}\right)_x = \left(\frac{u_{tt}}{u_{xx}}\right)_t = 0$, or equivalently,

$$u_{xtt}u_{xx} - u_{tt}u_{xxx} = 0, \quad u_{ttt}u_{xx} - u_{tt}u_{xxt} = 0.$$

A crucial point to notice is that the constant c^2 is nonnegative; thus an inequality of the form $\frac{u_{xx}}{u_{tt}} \geq 0$ or $u_{xx}u_{tt} \geq 0$ is imposed as well. In example 11, we visit this network again and study functions of the form

$$F(x,t) = \sigma(a''f(ax+bt) + b''g(a'x+b't)) \tag{1.14}$$

via a number of equalities and inequalities involving partial derivatives of F.

The preceding example suggests that smooth functions implemented by a neural network may be required to obey a nontrivial **algebraic p**artial **d**ifferential **i**nequality (**algebraic PDI**). So it is convenient to have the following setup of terminology.

Terminology

- An **algebraic PDI** is an inequality of the form

$$\Theta\left(F_{x_1}, \ldots, F_{x_n}, F_{x_1^2}, F_{x_1x_2}, \ldots, F_{\mathbf{x}^\alpha}, \ldots\right) > 0 \tag{1.15}$$

  involving partial derivatives (up to a certain order) where Θ is a real polynomial.

Remark 1. Without any loss of generality, we assume that the PDIs are strict since a nonstrict one such as $\Theta \geq 0$ could be written as the union of $\Theta > 0$ and the algebraic PDE $\Theta = 0$.

Theorem 1 and example 1 deal with superpositions of arbitrary smooth functions while theorem 2 and example 3 are concerned with superpositions of a specific class of smooth functions, functions of the form 1.12. In view of the necessary PDE constraints in both situations, the following question then arises: Are there sufficient conditions in the form of algebraic PDEs and PDIs that guarantee a smooth function can be represented, at least locally, by the neural network in question?

Conjecture 1. *Let $\mathcal{N}$ be a feedforward neural network whose inputs are labeled by the coordinate functions $x_1, \dots, x_n$. Suppose we are working in the setting of one of theorems 1 or 2. Then there exist*

- *finitely many nonvanishing conditions* $\left\{\Psi_i\left((F_{\mathbf{x}^\alpha})_{|\alpha|\leq r}\right) \neq 0\right\}_i$
- *finitely many algebraic PDEs* $\left\{\Phi_j\left((F_{\mathbf{x}^\alpha})_{|\alpha|\leq r}\right) = 0\right\}_j$
- *finitely many algebraic PDIs* $\left\{\Theta_k\left((F_{\mathbf{x}^\alpha})_{|\alpha|\leq r}\right) > 0\right\}_k$

with the following property: For any arbitrary point $\mathbf{p} \in \mathbb{R}^n$, the space of smooth functions $F = F(x_1, \dots, x_n)$ defined in a vicinity[1] of $\mathbf{p}$ that satisfy $\Psi_i \neq 0$ at $\mathbf{p}$ and are computable by $\mathcal{N}$ (in the sense of the regime under consideration) is nonvacuous and is characterized by PDEs $\Phi_j = 0$ and PDIs $\Theta_k > 0$.

To motivate the conjecture, notice that it claims the existence of functionals

$$\left\{F \mapsto \Psi_i\left((F_{\mathbf{x}^\alpha})_{|\alpha|\leq r}\right)\right\}_i,\ \left\{F \mapsto \Phi_j\left((F_{\mathbf{x}^\alpha})_{|\alpha|\leq r}\right)\right\}_j,\ \left\{F \mapsto \Theta_k\left((F_{\mathbf{x}^\alpha})_{|\alpha|\leq r}\right)\right\}_k,$$

which are polynomial expressions of partial derivatives, and hence continuous in the C^r-norm,[2] such that in the space of functions computable by $\mathcal{N}$, the open dense[3] subset given by $\{\Psi_i \neq 0\}_i$ can be described in terms of finitely many equations and inequalities as the locally closed subset $\left\{\Phi_j = 0\right\}_j \bigcup \{\Theta_k > 0\}_k$. (Also see corollary 1.) The usage of C^r-norm here is novel. For instance, with respect to L^p-norms, the space of functions computable by $\mathcal{N}$ lacks such a description and often has undesirable properties like nonclosedness (Petersen, Raslan, & Voigtlaender, 2020). Besides, describing the functional space associated with a neural

[1] To be mathematically precise, the open neighborhood of $\mathbf{p}$ on which F admits a compositional representation in the desired form may be dependent on F and $\mathbf{p}$. So conjecture 1 is local in nature and must be understood as a statement about function germs.

[2] Convergence in the C^r-norm is defined as the uniform convergence of the function and its partial derivatives up to order r.

[3] In conjecture 1, the subset cut off by equations $\Psi_i = 0$ is meager: It is a closed and (due to the term *nonvacuous* appearing in the conjecture) proper subset of the space of functions computable by $\mathcal{N}$, and a function implemented by $\mathcal{N}$ at which a Ψ_i vanishes could be perturbed to another computable function at which all of Ψ_i's are nonzero.

network $\mathcal{N}$ with finitely many equations and inequalities also has an algebraic motivation: it is reminiscent of the notion of a *semialgebraic set* from real algebraic geometry. To elaborate, take the activation functions to be polynomials. Such neural networks have been studied in the literature (Du & Lee, 2018; Soltanolkotabi, Javanmard, & Lee, 2018; Venturi, Bandeira, & Bruna, 2018; Kileel, Trager, & Bruna, 2019). By bounding the degrees of constituent functions of superpositions computed by a polynomial neural network, the functional space formed by these superpositions sits inside a finite-dimensional ambient space of real polynomials and is hence finite-dimensional and amenable to techniques of algebraic geometry. One can, for instance, in each degree associate a functional variety to a neural network $\mathcal{N}$ whose dimension could be interpreted as a measure of expressive power (Kileel et al., 2019). Our approach to describing real functions computable by neural networks via PDEs and PDIs has ramifications to the study of polynomial neural networks as well. Indeed, if $F = F(x_1, \ldots, x_n)$ is a polynomial, an algebraic PDE of the form 1.10 translates to a polynomial equation of the coefficients of F, and the condition that an algebraic PDI such as equation 1.15 is valid throughout $\mathbb{R}^n$ can again be described via equations and inequalities involving the coefficients of F (see examples 12 and 13). A notable feature here is the claim of the existence of a universal characterization dependent only on the architecture from which a description as a semialgebraic set could be read off in any degree.

Conjecture 1 is settled in (Farhoodi, Filom, Jones, and Körding, 2019) for trees (a particular type of architectures) with distinct inputs, a situation in which no PDI is required, and the inequalities should be taken to be trivial. Throughout the article, the conjecture above will be established for a number of architectures; in particular, we shall characterize tree functions (cf. theorems 3 and 4 below).

1.3 Related Work. There is an extensive literature on the expressive power of neural networks. Although shallow networks with sigmoidal activation functions can approximate any continuous function on compact sets (Cybenko, 1989; Hornik, Stinchcombe, & White, 1989; Hornik, 1991; Mhaskar, 1996), this cannot be achieved without the hidden layer getting exponentially large (Eldan & Shamir, 2016; Telgarsky, 2016; Mhaskar et al., 2017; Poggio et al., 2017). Many articles thus try to demonstrate how the expressive power is affected by depth. This line of research draws on a number of different scientific fields including algebraic topology (Bianchini & Scarselli, 2014), algebraic geometry (Kileel et al., 2019), dynamical systems (Chatziafratis, Nagarajan, Panageas, & Wang, 2019), tensor analysis (Cohen, Sharir, & Shashua, 2016), Vapnik–Chervonenkis theory (Bartlett, Maiorov, & Meir, 1999), and statistical physics (Lin, Tegmark, & Rolnick, 2017). One approach is to argue that deeper networks are able to approximate or represent functions of higher complexity after defining a "complexity measure" (Bianchini & Scarselli, 2014; Montufar, Pascanu,

Cho, & Bengio, 2014; Poole, Lahiri, Raghu, Sohl-Dickstein, & Ganguli, 2016; Telgarsky, 2016; Raghu et al., 2017). Another approach more in line with this article is to use the "size" of an associated functional space as a measure of representation power. This point of view is adapted in Farhoodi et al. (2019) by enumerating Boolean functions, and in Kileel et al. (2019) by regarding dimensions of functional varieties as such a measure.

A central result in the mathematical study of superpositions of functions is the celebrated Kolmogorov-Arnold representation theorem (Kolmogorov, 1957), which resolves (in the context of continuous functions) the thirteenth problem on Hilbert's famous list of 23 major mathematical problems (Hilbert, 1902). The theorem states that every continuous function $F(x_1, \ldots, x_n)$ on the closed unit cube may be written as

$$F(x_1, \ldots, x_n) = \sum_{i=1}^{2n+1} f_i \left(\sum_{j=1}^{n} \phi_{i,j}(x_j) \right) \tag{1.16}$$

for suitable continuous univariate functions f_i, $\phi_{i,j}$ defined on the unit interval. (See Vituškin and Henkin, 1967, chap. 1, or Vituškin, 2004, for a historical account.) In more refined versions of this theorem (Sprecher, 1965; Lorentz, 1966), the outer functions f_i are arranged to be the same, and the inner ones $\phi_{i,j}$ are be taken to be in the form of $\lambda_j \phi_i$ with λ_j's and ϕ_i's independent of F. Based on the existence of such an improved representation, Hecht-Nielsen argued that any continuous function F can be implemented by a three-layer neural network whose weights and activation functions are determined by the representation (Hecht-Nielsen, 1987). On the other hand, it is well known that even when F is smooth, one cannot arrange for functions appearing in representation 1.16 to be smooth (Vituškin, 1964). As a matter of fact, there exist continuously differentiable functions of three variables that cannot be represented as sums of superpositions of the form $g(f(x, y), z)$ with f and g being continuously differentiable as well (Vituškin, 1954) whereas in the continuous category, one can write any trivariate continuous functions as a sum of nine superpositions of the form $g(f(x, y), z)$ (Arnold, 2009b). Due to this emergence of nondifferentiable functions, it has been argued that Kolmogorov-Arnold's theorem is not useful for obtaining exact representations of functions via networks (Girosi & Poggio, 1989), although it may be used for approximation (Kůrková, 1991, 1992). More on algorithmic aspects of the theorem and its applications to the network theory can be found in Brattka (2007).

Focusing on a superposition

$$F = f_1^{(L)}\Big(f_1^{(L-1)}\left(f_{a_1}^{(L-2)}(\ldots), \ldots \right), \ldots, f_j^{(L-1)}\left(f_{a_j}^{(L-2)}(\ldots), \ldots \right), \ldots, f_{N_{L-1}}^{(L-1)}\left(f_{a_{N_{L-1}}}^{(L-2)}(\ldots), \ldots \right) \Big) \tag{1.17}$$

of smooth functions (which can be computed by a neural network as in Figure 3), the chain rule provides descriptions for partial derivatives of F in terms of partial derivatives of functions $f_j^{(i)}$ that constitute the superposition. The key insight behind the proof of theorem 1 is that when the former functions have fewer variables compared to F, one may eliminate the derivatives of $f_j^{(i)}$'s to obtain relations among partial derivatives of F. This idea of elimination has been utilized in Buck (1981b) and Rubel (1981) to prove the existence of universal algebraic differential equations whose C^∞ solutions are dense in the space of continuous functions. The fact that there will be constraints imposed on derivatives of a function F that is written as a superposition of differentiable functions was employed by Hilbert himself to argue that certain analytic functions of three variables are not superpositions of analytic functions of two variables (Arnold, 2009a, p. 28), and by Ostrowski to exhibit an analytic bivariate function that cannot be represented as a superposition of univariate smooth functions and multivariate algebraic functions due to the fact that it does not satisfy any nontrivial algebraic PDE (Vituškin, 2004, p. 14; Ostrowski, 1920). The novelty of our approach is to adapt this point of view to demonstrate theoretical limitations of smooth functions that neural networks compute either as a superposition as in theorem 1 or as compositions of functions of the form 1.12 as in theorem 2, and to try to characterize these functions via calculating PDE constraints that are sufficient too (cf. conjecture 1). Furthermore, necessary PDE constraints enable us to easily exhibit functions that cannot be computed by a particular architecture; see example 1. This is reminiscent of the famous Minsky XOR Theorem (Minsky & Papert, 2017). An interesting nonexample from the literature is $F(x, y, z) = xy + yz + zx$ which cannot be written as a superposition of the form 1.7 even in the continuous category (Pólya & Szegö, 1945; Buck, 1979, 1981a; von Golitschek, 1980; Arnold, 2009a).

To the best of our knowledge, the closest mentions of a characterization of a class of superpositions by necessary and sufficient PDE constraints in the literature are papers (Buck, 1979, 1981a) by R. C. Buck. The first one (along with its earlier version, Buck, 1976) characterizes superpositions of the form $g(f(x, y), z)$ in a similar fashion as example 1. Also in those papers, superpositions such as $g(f(x, y), h(x, z))$ (which appeared in example 2) are discussed although only the existence of necessary PDE constraints is shown; see (Buck, 1979, lemma 7), and (Buck, 1981a, p. 141). We exhibit a PDE characterization for superpositions of this form in example 7. These papers also characterize sufficiently differentiable *nomographic* functions of the form $\sigma(f(x) + g(y))$ and $\sigma(f(x) + g(y) + h(z))$.

A special class of neural network architectures is provided by rooted trees where any output of a layer is passed to exactly one node from one of the layers above (see Figure 8). Investigating functions computable by trees is of neuroscientific interest because the morphology of the dendrites of a

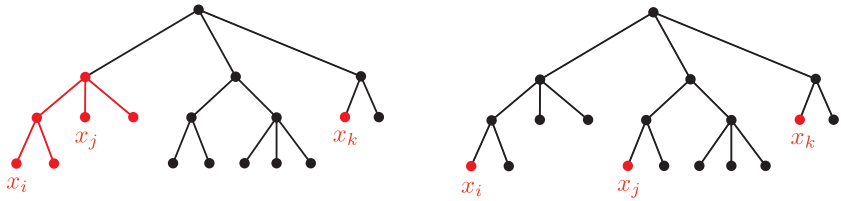

Figure 5: Theorems 3 and 4 impose constraints 1.18 and 1.19 for any three leaves x_i, x_j, and x_k. In the former theorem, the constraint should hold whenever (as on the left) there exists a rooted full subtree separating x_i and x_j from x_k, while in the latter theorem, the constraint is imposed for certain other triples as well (as on the right).

neuron processes information through a tree that is often binary (Kollins & Davenport, 2005; Gillette & Ascoli, 2015). Assuming that the inputs to a tree are distinct, in our previous work (Farhoodi et al., 2019), we have completely characterized the corresponding superpositions through formulating necessary and sufficient PDE constraints; a result that answers conjecture 1 in positive for such architectures.

Remark 2. The characterization suggested by the theorem below is a generalization of example 1 which was concerned with smooth superpositions of the form 1.1. The characterization of such superpositions as solutions of PDE 1.4 has also appeared in a paper (Buck, 1979) that we were not aware of while writing (Farhoodi et al., 2019).

Theorem 3 *(Farhoodi et al., 2019). Let $\mathcal{T}$ be a rooted tree with n leaves that are labeled by the coordinate functions $x_1, \ldots, x_n$. Let $F = F(x_1, \ldots, x_n)$ be a smooth function implemented on this tree. Then for any three leaves of $\mathcal{T}$ corresponding to variables x_i, x_j, x_k of F with the property that there is a (rooted full) subtree of $\mathcal{T}$ containing the leaves x_i, x_j while missing the leaf x_k (see Figure 5), F must satisfy*

$$F_{x_i x_k} F_{x_j} = F_{x_j x_k} F_{x_i}. \tag{1.18}$$

Conversely, a smooth function F defined in a neighborhood of a point $\mathbf{p} \in \mathbb{R}^n$ can be implemented by the tree $\mathcal{T}$ provided that equation 1.18 holds for any triple (x_i, x_j, x_k) of its variables with the above property; and moreover, the non-vanishing conditions below are satisfied:

- *For any leaf x_i with siblings either $F_{x_i}(\mathbf{p}) \neq 0$ or there is a sibling leaf $x_{i'}$ with $F_{x_{i'}}(\mathbf{p}) \neq 0$.*

This theorem was formulated in Farhoodi et al. (2019) for binary trees and in the context of analytic functions (and also that of Boolean functions). Nevertheless, the proof carries over to the more general setting above. Below, we formulate the analogous characterization of functions that trees

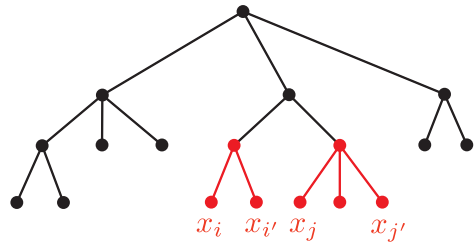

Figure 6: Theorem 4 imposes constraint 1.20 for any four leaves x_i, $x_{i'}$ and x_j, $x_{j'}$ that belong to two different rooted full subtrees emanating from a node.

compute via composing functions of the form 1.12. Proofs of theorems 3 and 4 are presented in section 4.

Theorem 4. *Let $\mathcal{T}$ be a rooted tree admitting n leaves that are labeled by the coordinate functions $x_1, \ldots, x_n$. We formulate the following constraints on smooth functions $F = F(x_1, \ldots, x_n)$:*

- *For any two leaves x_i and x_j of $\mathcal{T}$, we have*

$$F_{x_i x_k} F_{x_j} = F_{x_j x_k} F_{x_i} \tag{1.19}$$

  for any other leaf x_k of $\mathcal{T}$ that is not a leaf of a (rooted full) subtree that has exactly one of x_i or x_j (see Figure 5). In particular, equation 1.19 holds for any x_k if the leaves x_i and x_j are siblings, and for any x_i and x_j if the leaf x_k is adjacent to the root of $\mathcal{T}$.
- *For any two (rooted full) subtrees $\mathcal{T}_1$ and $\mathcal{T}_2$ that emanate from a node of $\mathcal{T}$ (see Figure 6), we have*

$$\begin{aligned} &F_{x_i} F_{x_j} \left[F_{x_i x_{i'} x_{j'}} F_{x_j} + F_{x_i x_{i'}} F_{x_j x_{j'}} - F_{x_i x_{j'}} F_{x_j x_{i'}} - F_{x_i} F_{x_j x_{i'} x_{j'}} \right] \\ &\quad = \left(F_{x_i x_{i'}} F_{x_j} - F_{x_i} F_{x_j x_{i'}} \right) \left(F_{x_i x_{j'}} F_{x_j} + F_{x_i} F_{x_j x_{j'}} \right) \end{aligned} \tag{1.20}$$

  if x_i, $x_{i'}$ are leaves of $\mathcal{T}_1$ and x_j, $x_{j'}$ are leaves of $\mathcal{T}_2$.

These constraints are satisfied if $F(x_1, \ldots, x_n)$ is a superposition of functions of the form $\mathbf{y} \mapsto \sigma(\langle \mathbf{w}, \mathbf{y} \rangle)$ according to the hierarchy provided by $\mathcal{T}$. Conversely, a smooth function F defined on an open box-like region[4] $B \subseteq \mathbb{R}^n$ can be written as such a superposition on B provided that the constraints 1.19 and 1.20 formulated above hold and, moreover, the nonvanishing conditions below are satisfied throughout B:

[4] An open box-like region in $\mathbb{R}^n$ is a product $I_1 \times \cdots \times I_n$ of open intervals.

- *For any leaf x_i with siblings either $F_{x_i} \neq 0$ or there is a sibling leaf $x_{i'}$ with $F_{x_{i'}} \neq 0$;*
- *For any leaf x_i without siblings $F_{x_i} \neq 0$.*

The constraints that appeared in theorems 3 and 4 may seem tedious, but they can be rewritten more conveniently once the intuition behind them is explained. Assuming that partial derivatives do not vanish (a nonvanishing condition) so that division is allowed, equations 1.18 and 1.19 may be written as

$$\left(\frac{F_{x_i}}{F_{x_j}}\right)_{x_k} = 0 \Leftrightarrow \left(\frac{F_{x_i}}{F_{x_k}}\right)_{x_j} = \left(\frac{F_{x_j}}{F_{x_k}}\right)_{x_i}, \tag{1.21}$$

while equation 1.20 is

$$\left(\frac{\left(\frac{F_{x_i}}{F_{x_j}}\right)_{x_{i'}}}{\frac{F_{x_i}}{F_{x_j}}}\right)_{x_{j'}} = 0. \tag{1.22}$$

Equation 1.21 simply states that the ratio $\frac{F_{x_i}}{F_{x_j}}$ is independent of x_k. Notice that in comparison with theorem 3, theorem 7, requires the equation $F_{x_i x_k} F_{x_j} = F_{x_j x_k} F_{x_i}$ to hold in a greater generality and for more triples (x_i, x_j, x_k) of leaves (see Figure 5).[5] The second simplified equation 1.22, holds once the function $\frac{F_{x_i}}{F_{x_j}}$ of $(x_1, \ldots, x_n)$ may be split into a product such as

$$q_1(\ldots, x_i, \ldots, x_{i'}, \ldots)\, q_2(\ldots, x_j, \ldots, x_{j'}, \ldots).$$

Lemma 4 discusses the necessity and sufficiency of these equations for the existence of such a splitting.

Remark 3. A significant feature of theorem 7 is that once the appropriate conditions are satisfied on a box-like domain, the smooth function under consideration may be written as a superposition of the desired form on the entirety of that domain. On the contrary, theorem 3 is local in nature.

Aside from neuroscientific interest, studying tree architectures is important also because any neural network can be expanded into a tree network

[5] A piece of terminology introduced in Farhoodi et al. (2019) may be illuminating here. A member of a triple (x_i, x_j, x_k) of (not necessarily distinct) leaves of $\mathcal{T}$ is called the *outsider* of the triple if there is a (rooted full) subtree of $\mathcal{T}$ that misses it but has the other two members. Theorem 3 imposes $F_{x_i x_k} F_{x_j} = F_{x_j x_k} F_{x_i}$ whenever x_k is the outsider, while theorem 4 imposes the constraint whenever x_i and x_j are not outsiders.

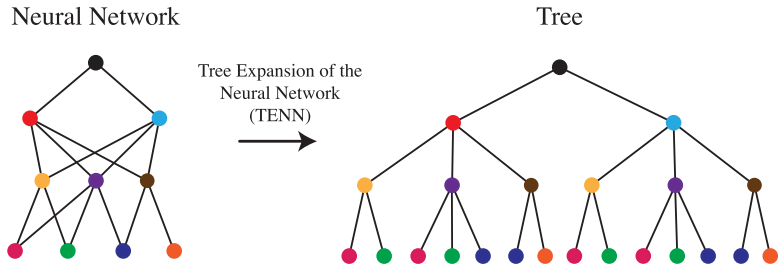

Figure 7: A multilayer neural network can be expanded to a tree. The figure is adapted from Farhoodi et al. (2019).

with repeated inputs through a procedure called **TENN** (the **T**ree **E**xpansion of the **N**eural **N**etwork; see Figure 7). Tree architectures with repeated inputs are relevant in the context of neuroscience too because the inputs to neurons may be repeated (Schneider-Mizell et al., 2016; Gerhard, Andrade, Fetter, Cardona, & Schneider-Mizell, 2017). We have already seen an example of a network along with its TENN in Figure 2. Both networks implement functions of the form $F(x, y, z) = g(f(x, y), h(x, z))$. Even for this simplest example of a tree architecture with repeated inputs, the derivation of characteristic PDEs is computationally involved and will be done in example 7. This verifies conjecture 1 for the tree that appeared in Figure 2.

1.4 Outline of the Article. Theorems 1 and 2 are proven in section 2 where it is established that in each setting, there are necessary PDE conditions for expressibility of smooth functions by a neural network. In section 3 we verify conjecture 1 in several examples by characterizing computable functions via PDE constraints that are necessary and (given certain nonvanishing conditions) sufficient. This starts by studying tree architectures in section 3.1. In example 7, we finish our treatment of a tree function with repeated inputs initiated in example 2; and, moreover, we present a number of examples to exhibit the key ideas of the proofs of theorems 3 and 4, which are concerned with tree functions with distinct inputs. The section then proceeds with switching from trees to other neural networks in section 3.2 where, building on example 3, example 11 demonstrates why the characterization claimed by conjecture 1 involves inequalities. We end section 3 with a brief subsection on PDE constraints for polynomial neural networks. Examples in section 3.1 are generalized in the next section to a number of results establishing conjecture 1 for certain families of tree architectures: Proofs of theorems 3 and 4 are presented in section 4. The last section is devoted to few concluding remarks. There are two appendices discussing technical proofs of propositions and lemmas (appendix A), and the basic mathematical background on differential forms (appendix B).

2 Existence of PDE Constraints

The goal of the section is to prove theorems 1 and 2. Lemma 1 below is our main tool for establishing the existence of constraints:

Lemma 1. *Any collection $p_1(t_1, \dots, t_m), \dots, p_l(t_1, \dots, t_m)$ of polynomials on m indeterminates are algebraically dependent provided that $l > m$. In other words, if $l > m$, there exists a nonconstant polynomial $\Phi = \Phi(s_1, \dots, s_l)$ dependent only on the coefficients of p_i's for which*

$$\Phi\left(p_1(t_1, \dots, t_m), \dots, p_l(t_1, \dots, t_m)\right) \equiv 0.$$

Proof. For a positive integer a, there are precisely $\binom{a+l}{l}$ monomials such as $p_1^{a_1} \dots p_l^{a_l}$ with their total degree $a_1 + \dots + a_l$ not greater than a. But each of them is a polynomial of $t_1, \dots, t_m$ of total degree at most ad where $d := \max\{\deg p_1, \dots, \deg p_l\}$. For a large enough, $\binom{a+l}{l}$ is greater than $\binom{ad+m}{m}$ because the degree of the former as a polynomial of a is l, while the degree of the latter is m. For such an a, the number of monomials $p_1^{a_1}, \dots, p_l^{a_l}$ is larger than the dimension of the space of polynomials of $t_1, \dots, t_m$ of total degree at most ad. Therefore, there exists a linear dependency among these monomials that amounts to a nontrivial polynomial relation among $p_1, \dots, p_l$. □

Proof of Theorem 1. Let $F = F(x_1, \dots, x_n)$ be a superposition of smooth functions

$$f_1^{(1)}, \dots, f_{N_1}^{(1)}; \dots; f_1^{(i)}, \dots, f_{N_i}^{(i)}; \dots; f_1^{(L)} \tag{2.1}$$

according to the hierarchy provided by $\mathcal{N}$ where $f_1^{(i)}, \dots, f_{N_i}^{(i)}$ are the functions appearing at the neurons of the ith layer above the input layer (in the last layer, $f_{N_L=1}^{(L)}$ appears at the output neuron). The total number of these functions is $N := N_1 + \dots + N_L$, namely, the number of the neurons of the network. By the chain rule, any partial derivative $F_{\mathbf{x}^\alpha}$ of the superposition may be described as a polynomial of partial derivatives of order not greater than $|\alpha|$ of functions that appeared in equation 2.1. These polynomials are determined solely by how neurons in consecutive layers are connected to each other, that is, the architecture. The function F of n variables admits $\binom{r+n}{n} - 1$ partial derivatives (excluding the function itself) of order at most r, whereas the same number for any of the functions listed in equation 2.1 is at most $\binom{r+n-1}{n-1} - 1$ because by the hypothesis, each of them is dependent on less than n variables. Denote the partial derivatives of order at most r of functions $f_j^{(i)}$ (evaluated at appropriate points as required by the chain rule) by indeterminates $t_1, \dots, t_m$. Following the previous discussion, one has $m \leq N\left(\binom{r+n-1}{n-1} - 1\right)$. Hence, the chain rule describes the partial

derivatives of order not greater than r of F as polynomials (dependent only on the architecture of $\mathcal{N}$) of $t_1, \ldots, t_m$. Invoking lemma 1, the partial derivatives of F are algebraically dependent once

$$\binom{r+n}{n} - 1 > N\left(\binom{r+n-1}{n-1} - 1\right). \tag{2.2}$$

Indeed, the inequality holds for r large enough since the left-hand side is a polynomial of degree n of r, while the similar degree for the right-hand side is $n-1$. □

Proof of Theorem 2. In this case $F = F(x_1, \ldots, x_n)$ is a superposition of functions of the form

$$\sigma_1^{(1)}\left(\langle \mathbf{w}_1^{(1)}, .\rangle\right), \ldots, \sigma_{N_1}^{(1)}\left(\langle \mathbf{w}_{N_1}^{(1)}, .\rangle\right); \ldots; \sigma_1^{(i)}\left(\langle \mathbf{w}_1^{(i)}, .\rangle\right), \ldots, \sigma_{N_i}^{(i)}\left(\langle \mathbf{w}_{N_i}^{(i)}, .\rangle\right);$$
$$\ldots; \sigma_1^{(L)}\left(\langle \mathbf{w}_1^{(L)}, .\rangle\right) \tag{2.3}$$

appearing at neurons. The jth neuron of the ith layer above the input layer ($1 \le i \le N, 1 \le j \le N_i$) corresponds to the function $\sigma_j^{(i)}\left(\langle \mathbf{w}_j^{(i)}, .\rangle\right)$ where a univariate smooth activation function $\sigma_j^{(i)}$ is applied to the inner product of the weight vector $\mathbf{w}_j^{(i)}$ with the vector formed by the outputs of neurons in the previous layer which are connected to the neuron of the ith layer. We proceed as in the proof of theorem 1. The chain rule describes each partial derivative $F_{\mathbf{x}^\alpha}$ as a polynomial, dependent only on the architecture, of components of vectors $\mathbf{w}_j^{(i)}$ along with derivatives of functions $\sigma_j^{(i)}$ up to order at most $|\alpha|$ (each evaluated at an appropriate point). The total number of components of all weight vectors coincides with the total number of connections (edges of the underlying graph), and the number of the derivatives of activation functions is the number of neurons times $|\alpha|$. We denote the total number of connections and neurons by C and N, respectively. There are $\binom{r+n}{n} - 1$ partial derivatives $F_{\mathbf{x}^\alpha}$ of order at most r (i.e., $|\alpha| \le r$) of F and, by the previous discussion, each of them may be written as a polynomial of $C + Nr$ quantities given by components of weight vectors and derivatives of activation functions. Lemma 1 implies that these partial derivatives of F are algebraically dependent provided that

$$\binom{r+n}{n} - 1 > Nr + C, \tag{2.4}$$

an inequality that holds for sufficiently large r as the degree of the left-hand side with respect to r is $n > 1$. □

Corollary 1. *Let $\mathcal{N}$ be a feedforward neural network whose inputs are labeled by the coordinate functions $x_1, \ldots, x_n$ and satisfies the hypothesis of either of theorems 1 or 2. Define the positive integer r as*

- $r = n\,(\#neurons - 1)$ *in the case of theorem 1*
- $r = \max\left(\lfloor n\,(\#neurons)^{\frac{1}{n-1}} \rfloor, \#connections\right) + 2$ *in the case of theorem 2,*

where #connections and #neurons are, respectively, the number of edges of the underlying graph of $\mathcal{N}$ and the number of its vertices above the input layer. Then the smooth functions $F = F(x_1, \ldots, x_n)$ computable by $\mathcal{N}$ satisfy nontrivial algebraic partial differential equations of order r. In particular, the subspace formed by these functions lies in a subset of positive codimension, which is closed with respect to the C^r-norm.

Proof. One only needs to verify that for the values of r provided by the corollary the inequalities 2.2 and 2.4 are valid. The former holds if

$$\frac{\binom{r+n}{n}}{\binom{r+n-1}{n-1}} = \frac{r+n}{n}$$

is not smaller than N, that is, if $r \geq n(N-1)$. As for equation 2.4, notice that

$$\binom{r+n}{n} - 1 - Nr \geq \frac{r^n}{n!} - Nr = r\left(\frac{r^{n-1}}{n!} - N\right);$$

hence, it suffices to have $r\left(\frac{r^{n-1}}{n!} - N\right) > C$. This holds if $r > C$ and $\frac{r^{n-1}}{n!} - N \geq 1$. The latter inequality is valid once $r \geq n.N^{\frac{1}{n-1}} + 2$, since then:

$$\frac{r^{n-1}}{n!} = \left(\frac{r}{(n!)^{\frac{1}{n-1}}}\right)^{n-1} \geq \left(\frac{r}{n}\right)^{n-1} \geq \left(N^{\frac{1}{n-1}} + \frac{2}{n}\right)^{n-1}$$
$$\geq N + \frac{2(n-1)}{n}.N^{\frac{n-2}{n-1}} \geq N + 1.$$

□

Remark 4. It indeed follows from the arguments above that there is a multitude of algebraically independent PDE constraints. By a simple dimension count, this number is $\left(\binom{r+n}{n} - 1\right) - N\left(\binom{r+n-1}{n-1} - 1\right)$ in the first case of corollary 1 and $\left(\binom{r+n}{n} - 1\right) - Nr$ in the second case.

Remark 5. The approach here merely establishes the existence of nontrivial algebraic PDEs satisfied by the superpositions. These are not the simplest PDEs of this kind and hence are not the best candidates for the purpose of characterizing superpositions. For instance, for superpositions 1.7, which networks in Figure 2 implement, one has $n = 3$ and #neurons $= 3$. Corollary 1 thus guarantees that these superpositions satisfy a sixth-order PDE. But in

example 7, we shall characterize them via two fourth-order PDEs (compare with Buck, 1979, lemma 7).

Remark 6. Prevalent smooth activation functions such as the logistic function $\frac{1}{1+e^{-x}}$ or tangent hyperbolic $\frac{e^x-e^{-x}}{e^x+e^{-x}}$ satisfy certain autonomous algebraic ODEs. Corollary 1 could be improved in such a setting. If each activation function $\sigma = \sigma(x)$ appearing in equation 2.3 satisfies a differential equation of the form

$$\frac{d^k\sigma}{dx^k} = p\left(\sigma, \frac{d\sigma}{dx}, \ldots, \frac{d^{k-1}\sigma}{dx^{k-1}}\right)$$

where p is a polynomial, one can change equation 2.4 to $\binom{r+n}{n} - 1 > Nk_{\max} + C$ where $k_{\max}$ is the maximum order of ODEs that activation functions in equation 2.3 satisfy.

3 Toy Examples

This section examines several elementary examples demonstrating how one can derive a set of necessary or sufficient PDE constraints for an architecture. The desired PDEs should be universal, that is, purely in terms of the derivatives of the function F that is to be implemented and not dependent on any weight vector, activation function, or a function of lower dimensionality that has appeared at a node. In this process, it is often necessary to express a smooth function in terms of other functions. If $k < n$ and $f(x_1, \ldots, x_n)$ is written as $g(\xi_1, \ldots, \xi_k)$ throughout an open neighborhood of a point $\mathbf{p} \in \mathbb{R}^n$ where each $\xi_i = \xi_i(x_1, \ldots, x_n)$ is a smooth function, the gradient of f must be a linear combination of those of $\xi_1, \ldots, \xi_k$ due to the chain rule. Conversely, if $\nabla f \in \text{Span}\{\nabla\xi_1, \ldots, \nabla\xi_k\}$ near $\mathbf{p}$, by the inverse function theorem, one can extend $(\xi_1, \ldots, \xi_k)$ to a coordinate system $(\xi_1, \ldots, \xi_k; \xi_{k+1}, \ldots, \xi_n)$ on a small enough neighborhood of $\mathbf{p}$ provided that $\nabla\xi_1(\mathbf{p}), \ldots, \nabla\xi_k(\mathbf{p})$ are linearly independent; a coordinate system in which the partial derivative f_{ξ_i} vanishes for $k < i \leq n$; the fact that implies f can be expressed in terms of $\xi_1, \ldots, \xi_k$ near $\mathbf{p}$. Subtle mathematical issues arise if one wants to write f as $g(\xi_1, \ldots, \xi_k)$ on a larger domain containing $\mathbf{p}$:

- A k-tuple $(\xi_1, \ldots, \xi_k)$ of smooth functions defined on an open subset U of $\mathbb{R}^n$ whose gradient vector fields are linearly independent at all points cannot necessarily be extended to a coordinate system $(\xi_1, \ldots, \xi_k; \xi_{k+1}, \ldots, \xi_n)$ for the whole U. As an example, consider $r = \sqrt{x^2 + y^2}$ whose gradient is nonzero at any point of $\mathbb{R}^2 - \{(0, 0)\}$, but there is no smooth function $h : \mathbb{R}^2 - \{(0, 0)\} \to \mathbb{R}$ with $\nabla h \not\parallel \nabla r$ throughout $\mathbb{R}^2 - \{(0, 0)\}$. The level set $r = 1$ is compact, and so the

restriction of h to it achieves its absolute extrema, and at such points $\nabla h = \lambda \nabla f$ (λ is the Lagrange multiplier).

- Even if one has a coordinate system $(\xi_1, \ldots, \xi_k; \xi_{k+1}, \ldots, \xi_n)$ on a connected open subset U of $\mathbb{R}^n$, a smooth function $f : U \to \mathbb{R}$ with $f_{\xi_{k+1}}, \ldots, f_{\xi_n} \equiv 0$ cannot necessarily be written globally as $f = g(\xi_1, \ldots, \xi_k)$. One example is the function

$$f(x, y) := \begin{cases} 0 & \text{if } x \leq 0 \\ e^{\frac{-1}{x}} & \text{if } x > 0, y > 0 \\ -e^{\frac{-1}{x}} & \text{if } x > 0, y < 0 \end{cases}$$

defined on the open subset $\mathbb{R}^2 - [0, \infty) \subset \mathbb{R}^2$ for which $f_y \equiv 0$. It may only locally be written as $f(x, y) = g(x)$; there is no function $g : \mathbb{R} \to \mathbb{R}$ with $f(x, y) = g(x)$ for all $(x, y) \in \mathbb{R}^2 - [0, \infty)$. Defining $g(x_0)$ as the value of f on the intersection of its domain with the vertical line $x = x_0$ does not work because, due to the shape of the domain, such intersections may be disconnected. Finally, notice that f, although smooth, is not analytic (C^ω); indeed, examples of this kind do not exist in the analytic category.

This difficulty of needing a representation $f = g(\xi_1, \ldots, \xi_k)$ that remains valid not just near a point but over a larger domain comes up only in the proof of theorem 4 (see remark 3); the representations we work with in the rest of this section are all local. The assumption about the shape of the domain and the special form of functions 1.12 allows us to circumvent the difficulties just mentioned in the proof of theorem 4. Below we have two related lemmas that we use later.

Lemma 2. *Let B and $\mathcal{T}$ be a box-like region in $\mathbb{R}^n$ and a rooted tree with the coordinate functions $x_1, \ldots, x_n$ labeling its leaves as in theorem 7. Suppose a smooth function $F = F(x_1, \ldots, x_n)$ on B is implemented on $\mathcal{T}$ via assigning activation functions and weights to the nodes of $\mathcal{T}$. If F satisfies the nonvanishing conditions described at the end of theorem 7, then the level sets of F are connected and F can be extended to a coordinate system $(F, F_2, \ldots, F_n)$ for B.*

Lemma 3. *A smooth function $F(x_1, \ldots, x_n)$ of the form $\sigma(a_1 x_1 + \cdots + a_n x_n)$ satisfies $F_{x_i x_k} F_{x_j} = F_{x_j x_k} F_{x_i}$ for any $1 \leq i, j, k \leq n$. Conversely, if F has a first-order partial derivative F_{x_j} which is nonzero throughout an open box-like region B in its domain, each identity $F_{x_i x_k} F_{x_j} = F_{x_j x_k} F_{x_i}$ could be written as $\left(\frac{F_{x_i}}{F_{x_j}}\right)_{x_k} = 0$; that is, for any $1 \leq i \leq n$, the ratio $\frac{F_{x_i}}{F_{x_j}}$ should be constant on B, and such requirements guarantee that F admits a representation of the form $\sigma(a_1 x_1 + \cdots + a_n x_n)$ on B.*

In view of the discussion so far, it is important to know when a smooth vector field,

$$\mathbf{V}(x_1, \ldots, x_n) = [V_1(x_1, \ldots, x_n) \ \ldots \ V_n(x_1, \ldots, x_n)]^{\mathrm{T}}, \tag{3.1}$$

on an open subset $U \subset \mathbb{R}^n$ is locally given by a gradient. Clearly, a necessary condition is to have

$$(V_i)_{x_j} = (V_j)_{x_i} \quad \forall\, i,\, j \in \{1, \ldots, n\}. \tag{3.2}$$

It is well known that if U is simply connected, this condition is sufficient too and guarantees the existence of a smooth potential function ξ on U satisfying $\nabla\xi = \mathbf{V}$ (Pugh, 2002). A succinct way of writing equation 3.2 is $\mathrm{d}\omega = 0$ where ω is defined as the *differential form*:

$$\omega := V_1\, \mathrm{d}x_1 + \cdots + V_n\, \mathrm{d}x_n. \tag{3.3}$$

Here is a more subtle question also pertinent to our discussion: When may $\mathbf{V}$ be rescaled to a gradient vector field? As the reader may recall from the elementary theory of differential equations, for a planer vector field, such a rescaling amounts to finding an integration factor for the corresponding first order ODE (Boyce & DiPrima, 2012). It turns out that the answer could again be encoded in terms of differential forms:

Theorem 5. *A smooth vector field* $\mathbf{V}$ *is parallel to a gradient vector field near each point only if the corresponding differential 1-form* ω *satisfies* $\omega \wedge \mathrm{d}\omega = 0$. *Conversely, if* $\mathbf{V}$ *is nonzero at a point* $\mathbf{p} \in \mathbb{R}^n$ *in the vicinity of which* $\omega \wedge \mathrm{d}\omega = 0$ *holds, there exists a smooth function* ξ *defined on a suitable open neighborhood of* $\mathbf{p}$ *that satisfies* $\mathbf{V} \parallel \nabla\xi \neq \mathbf{0}$. *In particular, in dimension 2, a nowhere vanishing vector field* $\mathbf{V}$ *is locally parallel to a nowhere vanishing gradient vector field, while in dimension 3, that is the case if and only if* $\mathbf{V} \,.\, \mathrm{curl}\mathbf{V} = \mathbf{0}$.

A proof and background on differential forms are provided in appendix B.

3.1 Trees with Four Inputs. We begin with officially defining the terms related to tree architectures (see Figure 8).

Terminology

A **tree** is a connected acyclic graph. Singling out a vertex as its root turns it into a directed acyclic graph in which each vertex has a unique predecessor/parent. We take all trees to be rooted. The following notions come up frequently:

- **Leaf**: a vertex with no successor/child.
- **Node**: a vertex that is not a leaf, that is, has children.
- **Sibling leaves**: leaves with the same parent.
- **Subtree**: all descendants of a vertex along with the vertex itself. Hence in our convention, all subtrees are full and rooted.

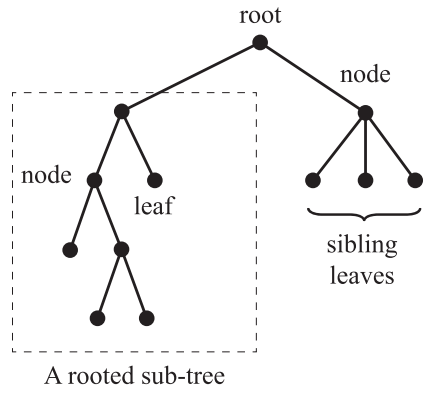

Figure 8: A tree architecture and the relevant terminology.

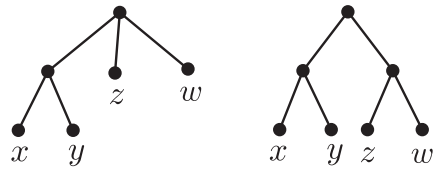

Figure 9: Two tree architectures with four distinct inputs. Examples 4, 5, and 6 characterize functions computable by them.

To implement a function, the leaves pass the inputs to the functions assigned to the nodes. The final output is received from the root.

The first example of the section elucidates theorem 3.

3.1.1 Example 4. Let us characterize superpositions

$$F(x, y, z, w) = g(f(x, y), z, w)$$

of smooth functions f, g, which correspond to the first tree architecture in Figure 9. Necessary PDE constraints are more convenient to write for certain ratios. So to derive them, we assume for a moment that first-order partial derivatives of F are nonzero, although by a simple continuity argument, the constraints will hold regardless. Computing the numerator and the denominator of $\frac{F_x}{F_y}$ via the chain rule indicates that this ratio coincides with $\frac{f_x}{f_y}$ and is, hence, independent of z, w. One thus obtains

$$\left(\frac{F_y}{F_x}\right)_z = 0, \quad \left(\frac{F_y}{F_x}\right)_w = 0,$$

or, equivalently,

$$F_{yz}F_x = F_{xz}F_y, \quad F_{yw}F_x = F_{xw}F_y.$$

Assuming $F_x \neq 0$, the preceding constraints are sufficient. The gradient ∇F is parallel with

$$\begin{bmatrix} 1 \\ \frac{F_y}{F_x} \\ \frac{F_z}{F_x} \\ \frac{F_w}{F_x} \end{bmatrix}$$

where the second entry $\frac{F_y}{F_x}$ is dependent only on x and y and thus may be written as $\frac{F_y}{F_x} = \frac{f_y}{f_x}$ for an appropriate bivariate function $f = f(x, y)$ defined throughout a small enough neighborhood of the point under consideration (at which F_x is assumed to be nonzero). Such a function exists due to theorem 5. Now we have

$$\nabla F \parallel \begin{bmatrix} 1 \\ \frac{f_y}{f_x} \\ 0 \\ 0 \end{bmatrix} + \frac{F_z}{F_x}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \frac{F_w}{F_x}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \in \text{Span}\left\{\nabla f, \nabla z, \nabla w\right\},$$

which guarantees that $F(x, y, z, w)$ may be written as a function of $f(x, y), z, w$.

The next two examples serve as an invitation to the proof of theorem 4 in section 4 and are concerned with trees illustrated in Figure 9.

3.1.2 Example 5. Let us study the example above in the regime of activation functions. The goal is to characterize functions of the form $F(x, y, z, w) = \sigma(\tau(ax + by) + cz + dw)$. The ratios $\frac{F_y}{F_x}, \frac{F_z}{F_w}$ must be constant while $\frac{F_x}{F_z}$ and $\frac{F_x}{F_w}$ are dependent merely on x, y as they are equal to $\frac{a}{c}\tau'(ax + by)$ and $\frac{a}{d}\tau'(ax + by)$, respectively. Equating the corresponding partial derivatives with zero, we obtain the following PDEs:

$$F_{xy}F_x = F_{xx}F_y, \quad F_{yy}F_x = F_{xy}F_y, \quad F_{yz}F_x = F_{xz}F_y, \quad F_{yw}F_x = F_{xw}F_y;$$
$$F_{xz}F_w = F_{xw}F_z, \quad F_{yz}F_w = F_{yw}F_z, \quad F_{zz}F_w = F_{zw}F_z, \quad F_{zw}F_w = F_{ww}F_z;$$
$$F_{xz}F_z = F_{zz}F_x, \quad F_{xw}F_z = F_{zw}F_x; \quad F_{xz}F_w = F_{zw}F_x, \quad F_{xw}F_w = F_{ww}F_x.$$

One can easily verify that they always hold for functions of the form above. We claim that under the assumptions of $F_x \neq 0$ and $F_w \neq 0$, these conditions guarantee the existence of a local representation of the form $\sigma(\tau(ax+by)+cz+dw)$ of F. Denoting $\frac{F_x}{F_w}$ by $\beta(x,y)$ and the constant functions $\frac{F_y}{F_x}$ and $\frac{F_z}{F_w}$ by c_1 and c_2, respectively, we have

$$\nabla F = \begin{bmatrix} F_x \\ F_y \\ F_z \\ F_w \end{bmatrix} \parallel \begin{bmatrix} \frac{F_x}{F_w} \\ \frac{F_y}{F_w} \\ \frac{F_z}{F_w} \\ 1 \end{bmatrix} = \begin{bmatrix} \beta(x,y) \\ c_1\beta(x,y) \\ c_2 \\ 1 \end{bmatrix} \parallel \nabla(f(x,y)+c_2z+w),$$

where $\nabla f = \begin{bmatrix} \beta(x,y) \\ c_1\beta(x,y) \end{bmatrix}$. Such a potential function f for $\begin{bmatrix} \beta(x,y) \\ c_1\beta(x,y) \end{bmatrix} = \begin{bmatrix} \frac{F_x}{F_w} \\ \frac{F_y}{F_w} \end{bmatrix}$ exists since

$$\left(\frac{F_x}{F_w}\right)_y = \left(\frac{F_y}{F_w}\right)_x \Leftrightarrow \left(\frac{F_y}{F_x}\right)_w = 0,$$

and it must be in the form of $\tau(ax+by)$ as $\frac{f_y}{f_x} = c_1$ is constant (see lemma 3). Thus, F is a function of $\tau(ax+by)+c_2z+w$ because the gradients are parallel.

The next example is concerned with the symmetric tree in Figure 9. We shall need the following lemma:

Lemma 4. *Suppose a smooth function* $q = q\left(y_1^{(1)}, \ldots, y_{n_1}^{(1)}; y_1^{(2)}, \ldots, y_{n_2}^{(2)}\right)$ *is written as a product*

$$q_1\left(y_1^{(1)}, \ldots, y_{n_1}^{(1)}\right) q_2\left(y_1^{(2)}, \ldots, y_{n_2}^{(2)}\right) \tag{3.4}$$

of smooth functions q_1, q_2. *Then* $q\, q_{y_a^{(1)} y_b^{(2)}} = q_{y_a^{(1)}}\, q_{y_b^{(2)}}$ *for any* $1 \leq a \leq n_1$ *and* $1 \leq b \leq n_2$. *Conversely, for a smooth function* q *defined on an open box-like region* $B_1 \times B_2 \subseteq \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$, *once* q *is nonzero, these identities guarantee the existence of such a product representation on* $B_1 \times B_2$.

3.1.3 Example 6. We aim for characterizing smooth functions of four variables of the form $F(x, y, z, w) = \sigma(\tau_1(ax + by) + \tau_2(cz + dw))$. Assuming for a moment that all first-order partial derivatives are nonzero, the ratios $\frac{F_y}{F_x}$, $\frac{F_z}{F_w}$ must be constant while $\frac{F_x}{F_w}$ is equal to $\frac{a\tau_1'(ax+by)}{d\tau_2'(cz+dw)}$ and hence (along with its constant multiples $\frac{F_x}{F_z}$, $\frac{F_y}{F_z}$, $\frac{F_y}{F_w}$) splits into a product of bivariate functions of x, y and z, w, a requirement that by lemma 4 is equivalent to the following identities:

$$\frac{F_x}{F_w}\left(\frac{F_x}{F_w}\right)_{xz} = \left(\frac{F_x}{F_w}\right)_x \left(\frac{F_x}{F_w}\right)_z, \quad \frac{F_x}{F_w}\left(\frac{F_x}{F_w}\right)_{xw} = \left(\frac{F_x}{F_w}\right)_x \left(\frac{F_x}{F_w}\right)_w,$$
$$\frac{F_x}{F_w}\left(\frac{F_x}{F_w}\right)_{yz} = \left(\frac{F_x}{F_w}\right)_y \left(\frac{F_x}{F_w}\right)_z, \quad \frac{F_x}{F_w}\left(\frac{F_x}{F_w}\right)_{yw} = \left(\frac{F_x}{F_w}\right)_y \left(\frac{F_x}{F_w}\right)_w.$$

After expanding and cross-multiplying, the identities above result in PDEs of the form 1.20 imposed on F that hold for any smooth function of the form $F(x, y, z, w) = \sigma(\tau_1(ax + by) + \tau_2(cz + dw))$. Conversely, we claim that if $F_x \neq 0$ and $F_w \neq 0$, then the constraints we have noted guarantee that F locally admits a representation of this form. Denoting the constants $\frac{F_y}{F_x}$ and $\frac{F_z}{F_w}$ by c_1 and c_2, respectively, and writing $\frac{F_x}{F_w} \neq 0$ in the split form $\frac{\beta(x,y)}{\gamma(z,w)}$, we obtain

$$\nabla F = \begin{bmatrix} F_x \\ F_y \\ F_z \\ F_w \end{bmatrix} \parallel \begin{bmatrix} \frac{F_x}{F_w} \\ \frac{F_y}{F_w} \\ \frac{F_z}{F_w} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\beta(x,y)}{\gamma(z,w)} \\ c_1 \frac{\beta(x,y)}{\gamma(z,w)} \\ c_2 \\ 1 \end{bmatrix} \parallel \begin{bmatrix} \beta(x, y) \\ c_1\beta(x, y) \\ c_2\gamma(z, w) \\ \gamma(z, w) \end{bmatrix}.$$

We desire functions $f = f(x, y)$ and $g = g(z, w)$ with $\nabla f = \begin{bmatrix} \beta(x, y) \\ c_1\beta(x, y) \end{bmatrix}$ and $\nabla g = \begin{bmatrix} c_2\gamma(z, w) \\ \gamma(z, w) \end{bmatrix}$, because then, $\nabla F \parallel \nabla(f(x, y) + g(z, w))$ and hence $F = \sigma\left(f(x, y) + g(z, w)\right)$ for an appropriate σ. Notice that $f(x, y)$ and $g(z, w)$ are automatically in the forms of $\tau_1(ax + by)$ and $\tau_2(cz + dw)$ because $\frac{f_y}{f_x} = c_1$ and $\frac{f_z}{f_w} = c_2$ are constants (see lemma 3). To establish the existence of f and g, one should verify the integrability conditions $\beta_y = c_1\beta_x$ and $c_2\gamma_w = \gamma_z$. We only verify the first one; the second one is similar. Notice that $\frac{F_y}{F_x} = c_1$ is constant, and $\frac{F_x}{F_w} = \frac{\beta(x,y)}{\gamma(z,w)}$ implies that $\beta_x = \beta\frac{\left(\frac{F_x}{F_w}\right)_x}{\frac{F_x}{F_w}}$ while $\beta_y = \beta\frac{\left(\frac{F_x}{F_w}\right)_y}{\frac{F_x}{F_w}}$. So the question is whether

$$\frac{F_y}{F_x}\left(\frac{F_x}{F_w}\right)_x = \left(\frac{F_y}{F_x}\frac{F_x}{F_w}\right)_x = \left(\frac{F_y}{F_w}\right)_x$$

and $\left(\frac{F_x}{F_w}\right)_y$ coincide, which is the case since $\left(\frac{F_y}{F_w}\right)_x = \left(\frac{F_x}{F_w}\right)_y$ can be rewritten as $\left(\frac{F_y}{F_x}\right)_w = 0$.

Remark 7. Examples 5 and 6 demonstrate an interesting phenomenon: one can deduce nontrivial facts about the weights once a formula for the implemented function is available. In example 5, for a function $F(x, y, z, w) = \sigma(\tau(ax + by) + cz + dw)$, we have $\frac{F_y}{F_x} \equiv \frac{b}{a}$ and $\frac{F_z}{F_w} \equiv \frac{c}{d}$. The same identities are valid for functions of the form $F(x, y, z, w) = \sigma(\tau_1(ax + by) + \tau_2(cz + dw))$ in example 6.[6] This seems to be a direction worthy of study. In fact, there are papers discussing how a neural network may be "reverse-engineered" in the sense that the architecture of the network is determined from the knowledge of its outputs, or the weights and biases are recovered without the ordinary training process involving gradient descent algorithms (Fefferman & Markel, 1994; Dehmamy, Rohani, & Katsaggelos, 2019; Rolnick & Kording, 2019). In our approach, the weights appearing in a composition of functions of the form $\mathbf{y} \mapsto \sigma(\langle \mathbf{w}, \mathbf{y} \rangle)$ could be described (up to scaling) in terms of partial derivatives of the resulting superposition.

3.1.4 Example 7. Let us go back to example 2. In (Farhoodi et al., 2019, c, 7.2), a PDE constraint on functions of the form 1.7 is obtained via differentiating equation 1.9 several times and forming a matrix equation, which implies that a certain determinant of partial derivatives must vanish. The paper then raises the question of existence of PDE constraints that are both necessary and sufficient. The goal of this example is to derive such a characterization. Applying differentiation operators ∂_y, ∂_z, and ∂_{yz} to equation 1.9 results in

$$\begin{bmatrix} F_y & F_z & 0 & 0 \\ F_{yy} & F_{yz} & F_y & 0 \\ F_{yz} & F_{zz} & 0 & F_z \\ F_{yyz} & F_{yzz} & F_{yz} & F_{yz} \end{bmatrix} \begin{bmatrix} A \\ B \\ A_y \\ B_z \end{bmatrix} = \begin{bmatrix} F_x \\ F_{xy} \\ F_{xz} \\ F_{xyz} \end{bmatrix}.$$

[6] Notice that this is the best one can hope to recover because through scaling the weights and inversely scaling the inputs of activation functions, the function F could also be written as $\sigma(\tilde{\tau}(\lambda ax + \lambda by) + cz + dw)$ or $\sigma(\tilde{\tau_1}(\lambda ax + \lambda by) + \tau_2(cz + dw))$ where $\tilde{\tau}(y) := \tau\left(\frac{y}{\lambda}\right)$ and $\tilde{\tau_1}(y) := \tau_1\left(\frac{y}{\lambda}\right)$. Thus, the other ratios $\frac{a}{c}$ and $\frac{b}{d}$ are completely arbitrary.

If this matrix is nonsingular, a nonvanishing condition, Cramer's rule provides descriptions of A, B in terms of partial derivatives of F, and then $A_z = B_y = 0$ yield PDE constraints. Reversing this procedure, we show that these conditions are sufficient too. Let us assume that

$$\begin{aligned}\Psi &:= \begin{vmatrix} F_y & F_z & 0 & 0 \\ F_{yy} & F_{yz} & F_y & 0 \\ F_{yz} & F_{zz} & 0 & F_z \\ F_{yyz} & F_{yzz} & F_{yz} & F_{yz} \end{vmatrix} \\ &= (F_y)^2 F_z F_{yzz} - (F_y)^2 F_{yz} F_{zz} - F_y (F_z)^2 F_{yyz} + (F_z)^2 F_{yz} F_{yy} \neq 0. \end{aligned} \tag{3.5}$$

Notice that this condition is nonvacuous for functions $F(x, y, z)$ of the form 1.7 since they include all functions of the form $g(y, z)$. Then the linear system

$$\begin{bmatrix} F_x \\ F_{xy} \\ F_{xz} \\ F_{xyz} \end{bmatrix} = \begin{bmatrix} F_y & F_z & 0 & 0 \\ F_{yy} & F_{yz} & F_y & 0 \\ F_{yz} & F_{zz} & 0 & F_z \\ F_{yyz} & F_{yzz} & F_{yz} & F_{yz} \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \tag{3.6}$$

may be solved as

$$\begin{aligned} A &= \frac{\begin{vmatrix} F_x & F_z & 0 & 0 \\ F_{xy} & F_{yz} & F_y & 0 \\ F_{xz} & F_{zz} & 0 & F_z \\ F_{xyz} & F_{yzz} & F_{yz} & F_{yz} \end{vmatrix}}{\begin{vmatrix} F_y & F_z & 0 & 0 \\ F_{yy} & F_{yz} & F_y & 0 \\ F_{yz} & F_{zz} & 0 & F_z \\ F_{yyz} & F_{yzz} & F_{yz} & F_{yz} \end{vmatrix}} \\ &= \frac{1}{\Psi}[-F_y (F_z)^2 F_{xyz} + F_y F_z F_{xz} F_{yz} + F_x F_y F_z F_{yzz} \\ &\quad - F_x F_y F_{yz} F_{zz} + (F_z)^2 F_{xy} F_{yz} - F_x F_z (F_{yz})^2] \end{aligned} \tag{3.7}$$

and

$$B = \frac{\begin{vmatrix} F_y & F_x & 0 & 0 \\ F_{yy} & F_{xy} & F_y & 0 \\ F_{yz} & F_{xz} & 0 & F_z \\ F_{yyz} & F_{xyz} & F_{yz} & F_{yz} \end{vmatrix}}{\begin{vmatrix} F_y & F_z & 0 & 0 \\ F_{yy} & F_{yz} & F_y & 0 \\ F_{yz} & F_{zz} & 0 & F_z \\ F_{yyz} & F_{yzz} & F_{yz} & F_{yz} \end{vmatrix}}$$

$$= \frac{1}{\Psi}[(F_y)^2 F_z F_{xyz} - (F_y)^2 F_{xz} F_{yz} - F_y F_z F_{xy} F_{yz} - F_x F_y F_z F_{yyz} + F_x F_y (F_{yz})^2 + F_x F_z F_{yy} F_{yz}]. \tag{3.8}$$

Denote the numerators of 3.7 and 3.8 by Ψ_1 and Ψ_2, respectively:

$$\begin{aligned} \Psi_1 &= -F_y (F_z)^2 F_{xyz} + F_y F_z F_{xz} F_{yz} + F_x F_y F_z F_{yzz} \\ &\quad - F_x F_y F_{yz} F_{zz} + (F_z)^2 F_{xy} F_{yz} - F_x F_z (F_{yz})^2, \\ \Psi_2 &= (F_y)^2 F_z F_{xyz} - (F_y)^2 F_{xz} F_{yz} - F_y F_z F_{xy} F_{yz} \\ &\quad - F_x F_y F_z F_{yyz} + F_x F_y (F_{yz})^2 + F_x F_z F_{yy} F_{yz}. \end{aligned} \tag{3.9}$$

Requiring $A = \frac{\Psi_1}{\Psi}$ and $B = \frac{\Psi_2}{\Psi}$ to be independent of z and y, respectively, amounts to

$$\Phi_1 := (\Psi_1)_z \Psi - \Psi_1 \Psi_z = 0, \quad \Phi_2 := (\Psi_2)_y \Psi - \Psi_2 \Psi_y = 0. \tag{3.10}$$

A simple continuity argument demonstrates that the constraints $\Phi_1 = 0$ and $\Phi_2 = 0$ above are necessary even if the determinant 3.5 vanishes: if Ψ is identically zero on a neighborhood of a point $\mathbf{p} \in \mathbb{R}^3$, the identities 3.10 obviously hold throughout that neighborhood. Another possibility is that $\Psi(\mathbf{p}) = 0$, but there is a sequence $\{\mathbf{p}_n\}_n$ of nearby points with $\mathbf{p}_n \to \mathbf{p}$ and $\Psi(\mathbf{p}_n) \neq 0$. Then the polynomial expressions Φ_1, Φ_2 of partial derivatives vanish at any $\mathbf{p}_n$ and hence at $\mathbf{p}$ by continuity.

To finish the verification of conjecture 1 for superpositions of the form 1.7, one should establish that PDEs $\Phi_1 = 0$, $\Phi_2 = 0$ from equation 3.10 are sufficient for the existence of such a representation provided that the nonvanishing condition $\Psi \neq 0$ from equation 3.5 holds. In that case, the functions A and B from equations 3.7 and 3.8 satisfy equation 1.9. According to theorem 5, there exist smooth locally defined $f(x, y)$ and $h(x, z)$ with $\frac{f_x}{f_y} = A(x, y)$

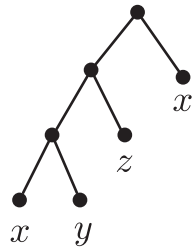

Figure 10: An asymmetric tree architecture that computes the superpositions of the form $F(x, y, z) = g(x, f(h(x, y), z))$. These are characterized in example 8.

and $\frac{h_x}{h_z} = B(x, z)$. We have:

$$\nabla F = \begin{bmatrix} A(x,y)F_y + B(x,z)F_z \\ F_y \\ F_z \end{bmatrix} = F_y \begin{bmatrix} A(x,y) \\ 1 \\ 0 \end{bmatrix} + F_z \begin{bmatrix} B(x,z) \\ 0 \\ 1 \end{bmatrix}$$
$$= \frac{F_y}{f_y} \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix} + \frac{F_z}{h_z} \begin{bmatrix} h_x \\ 0 \\ h_z \end{bmatrix} \in \text{Span}\{\nabla f, \nabla h\};$$

hence, F can be written as a function $g(f(x, y), h(x, z))$ of f and h for an appropriate g.

3.1.5 Example 8. We now turn to the asymmetric tree with four repeated inputs in Figure 10 with the corresponding superpositions,

$$F(x, y, z) = g(x, f(h(x, y), z)). \tag{3.11}$$

In our treatment here, the steps are reversible, and we hence derive PDE constraints that are simultaneously necessary and sufficient. The existence of a representation of the form 3.11 for $F(x, y, z)$ is equivalent to the existence of a locally defined coordinate system,

$$(\xi := x, \zeta, \eta),$$

with respect to which $F_\eta = 0$; moreover, $\zeta = \zeta(x, y, z)$ must be in the form of $f(h(x, y), z)$, which, according to example 1, is the case if and only if $\left(\frac{\zeta_y}{\zeta_x}\right)_z = 0$. Here, we assume that $\zeta_x, \zeta_y \neq 0$ so that $\frac{\zeta_y}{\zeta_x}$ is well defined and $\nabla\xi, \nabla\zeta$ are linearly independent. We denote the preceding ratio by $\beta = \beta(x, y) \neq 0$. Conversely, theorem 5 guarantees that there exists ζ with $\frac{\zeta_y}{\zeta_x} = \beta$ for any

smooth $\beta(x, y)$. The function F could be locally written as a function of $\xi = x$ and ζ if and only if

$$\nabla F = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \in \text{Span} \left\{ \nabla x, \nabla \zeta = \begin{bmatrix} \zeta_x \\ \zeta_y = \beta(x, y)\zeta_x \\ \zeta_z \end{bmatrix} \right\}.$$

Clearly, this occurs if and only if $\frac{F_z}{F_y}$ coincides with $\frac{\zeta_z}{\zeta_y}$. Therefore, one only needs to arrange for $\beta(x, y)$ so that the vector field

$$\frac{1}{\zeta_x}\nabla\zeta = \begin{bmatrix} 1 \\ \frac{\zeta_y}{\zeta_x} \\ \frac{\zeta_z}{\zeta_x} \end{bmatrix} = \begin{bmatrix} 1 \\ \beta(x, y) \\ \beta(x, y)\frac{F_z}{F_y} \end{bmatrix}$$

is parallel to a gradient vector field $\nabla\zeta$. That is, we want the vector field to be perpendicular to its curl (see theorem 5). We have:

$$\left(\frac{\partial}{\partial x} + \beta(x, y)\frac{\partial}{\partial y} + \beta(x, y)\frac{F_z}{F_y}\frac{\partial}{\partial z}\right) . \text{curl} \left(\frac{\partial}{\partial x} + \beta(x, y)\frac{\partial}{\partial y} + \beta(x, y)\frac{F_z}{F_y}\frac{\partial}{\partial z}\right)$$
$$= \beta_y \frac{F_z}{F_y} + \beta \left(\frac{F_z}{F_y}\right)_y - \beta^2 \left(\frac{F_z}{F_y}\right)_x .$$

The vanishing of the expression above results in a description of $\left(\frac{F_z}{F_y}\right)_x$ as the linear combination

$$\left(\frac{F_z}{F_y}\right)_x = \frac{\beta_y}{\beta^2}\frac{F_z}{F_y} + \frac{1}{\beta}\left(\frac{F_z}{F_y}\right)_y \tag{3.12}$$

whose coefficients $\frac{\beta_y}{\beta^2} = -\left(\frac{1}{\beta}\right)_y$ and $\frac{1}{\beta}$ are independent of z. Thus, we are in a situation similar to that of examples 2 and 7, where we encountered identity 1.9. The same idea used there could be applied again to obtain PDE constraints: Differentiating equation 3.12 with respect to z results in a linear system:

$$\begin{bmatrix} \frac{F_z}{F_y} & \left(\frac{F_z}{F_y}\right)_y \\ \left(\frac{F_z}{F_y}\right)_z & \left(\frac{F_z}{F_y}\right)_{yz} \end{bmatrix} \begin{bmatrix} -\left(\frac{1}{\beta}\right)_y \\ \frac{1}{\beta} \end{bmatrix} = \begin{bmatrix} \left(\frac{F_z}{F_y}\right)_x \\ \left(\frac{F_z}{F_y}\right)_{xz} \end{bmatrix}.$$

Assuming the matrix above is nonsingular, Cramer's rule implies

$$-\left(\frac{1}{\beta}\right)_y = \frac{\begin{vmatrix} \left(\frac{F_z}{F_y}\right)_x & \left(\frac{F_z}{F_y}\right)_y \\ \left(\frac{F_z}{F_y}\right)_{xz} & \left(\frac{F_z}{F_y}\right)_{yz} \end{vmatrix}}{\begin{vmatrix} \frac{F_z}{F_y} & \left(\frac{F_z}{F_y}\right)_y \\ \left(\frac{F_z}{F_y}\right)_z & \left(\frac{F_z}{F_y}\right)_{yz} \end{vmatrix}}, \quad \frac{1}{\beta} = \frac{\begin{vmatrix} \frac{F_z}{F_y} & \left(\frac{F_z}{F_y}\right)_x \\ \left(\frac{F_z}{F_y}\right)_z & \left(\frac{F_z}{F_y}\right)_{xz} \end{vmatrix}}{\begin{vmatrix} \frac{F_z}{F_y} & \left(\frac{F_z}{F_y}\right)_y \\ \left(\frac{F_z}{F_y}\right)_z & \left(\frac{F_z}{F_y}\right)_{yz} \end{vmatrix}}. \tag{3.13}$$

We now arrive at the desired PDE characterization of superpositions 3.11. In each of the ratios of determinants appearing in equation 3.13, the numerator and denominator are in the form of polynomials of partial derivatives divided by $(F_y)^4$. So we introduce the following polynomial expressions:

$$\begin{aligned} \Psi_1 &= (F_y)^4 \begin{vmatrix} \frac{F_z}{F_y} & \left(\frac{F_z}{F_y}\right)_y \\ \left(\frac{F_z}{F_y}\right)_z & \left(\frac{F_z}{F_y}\right)_{yz} \end{vmatrix}, \\ \Psi_2 &= (F_y)^4 \begin{vmatrix} \frac{F_z}{F_y} & \left(\frac{F_z}{F_y}\right)_x \\ \left(\frac{F_z}{F_y}\right)_z & \left(\frac{F_z}{F_y}\right)_{xz} \end{vmatrix}, \\ \Psi_3 &= (F_y)^4 \begin{vmatrix} \left(\frac{F_z}{F_y}\right)_x & \left(\frac{F_z}{F_y}\right)_y \\ \left(\frac{F_z}{F_y}\right)_{xz} & \left(\frac{F_z}{F_y}\right)_{yz} \end{vmatrix}. \end{aligned} \tag{3.14}$$

Then in view of equation 3.13,

$$\frac{\Psi_2}{\Psi_1} = \frac{1}{\beta}, \quad \frac{\Psi_3}{\Psi_1} = -\left(\frac{1}{\beta}\right)_y. \tag{3.15}$$

Hence $\left(\frac{\Psi_2}{\Psi_1}\right)_y + \frac{\Psi_3}{\Psi_1} = 0$; furthermore, $\left(\frac{\Psi_2}{\Psi_1}\right)_z = 0$ since β is independent of z:

$$\Phi_1 := \Psi_1(\Psi_2)_y - (\Psi_1)_y\Psi_2 + \Psi_1\Psi_3 = 0, \quad \Phi_2 := \Psi_1(\Psi_2)_z - (\Psi_1)_z\Psi_2 = 0. \tag{3.16}$$

Again as in example 7, a continuity argument implies that the algebraic PDEs above are necessary even when the denominator in equation 3.13 (i.e., Ψ_1) is zero. As for the nonvanishing conditions, in view of equations 3.14 and 3.15, we require F_y to be nonzero as well as Ψ_1 and Ψ_2 (recall that $\beta \neq 0$):

$$\Psi_1 \neq 0, \Psi_2 \neq 0, F_y \neq 0. \tag{3.17}$$

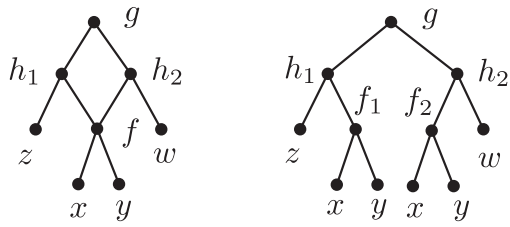

Figure 11: The space of functions computed by the neural network on the left is strictly smaller than that of its TENN on the right. See example 9.

It is easy to see that these conditions are not vacuous for functions of the form 3.11. If $F(x, y, z) = (xy)^2 z + z^3$, neither F_y nor the expression Ψ_1 or Ψ_2 is identically zero.

In summary, a special case of conjecture 1 has been verified in this example. A function $F = F(x, y, z)$ of the form 3.11 satisfies the constraints 3.16; conversely, a smooth function satisfying them along with the nonvanishing conditions 3.17 admits a local representation of that form.

3.2 Examples of Functions Computed by Neural Networks. We now switch from trees to examples of PDE constraints for neural networks. The first two examples are concerned with the network illustrated on the left of Figure 11; this is a ResNet with two hidden layers that has x, y, z, w as its inputs. The functions it implements are in the form of

$$F(x, y, z, w) = g(h_1(f(x, y), z), h_2(f(x, y), w)), \tag{3.18}$$

where f and h_1, h_2 are the functions appearing in the hidden layers.

3.2.1 Example 9. On the right of Figure 11, the tree architecture corresponding to the neural network discussed above is illustrated. The functions implemented by this tree are in the form of

$$F(x, y, z, w) = g(h_1(f_1(x, y), z), h_2(f_2(x, y), w)), \tag{3.19}$$

which is a form more general than the form 3.18 of functions computable by the network. In fact, there are PDEs satisfied by the latter class that functions in the former class, equation 3.19, do not necessarily satisfy. To see this, observe that for a function $F(x, y, z, w)$ of the form 3.18, the ratio $\frac{F_y}{F_x}$ coincides with $\frac{f_y}{f_x}$ and is thus independent of z and w—hence the PDEs $F_{xz}F_y = F_{yz}F_x$ and $F_{xw}F_y = F_{yw}F_x$. Neither of them holds for the function $F(x, y, z, w) = xyz + (x + y)w$, which is of the form 3.19. We deduce that the

set of PDE constraints for a network may be strictly larger than that of the corresponding TENN.

3.2.2 Example 10. Here we briefly argue that conjecture 1 holds for the network in Figure 11 (which has two hidden layers). The goal is to obtain PDEs that, given suitable nonvacuous, nonvanishing conditions, characterize smooth functions $F(x, y, z, w)$ of the form 3.18. We seek a description of the form $g(F_1(x, y, z), F_2(x, y, w))$ of $F(x, y, z, w)$ where the trivariate functions $F_1(x, y, z)$ and $F_2(x, y, w)$ are superpositions $h_1(f(x, y), z)$ and $h_2(f(x, y), w)$ with the same bivariate function f appearing in both of them. Invoking the logic that has been used repeatedly in section 3.1, ∇F must be a linear combination of ∇F_1 and ∇F_2. Following example 1, the only restriction on the latter two gradients is

$$\nabla F_1 \parallel \begin{bmatrix} 1 \\ \frac{(F_1)_y}{(F_1)_x} = \frac{f_y}{f_x} \\ \alpha(x, y, z) := \frac{(F_1)_z}{(F_1)_x} \\ 0 \end{bmatrix}, \quad \nabla F_2 \parallel \begin{bmatrix} 1 \\ \frac{(F_2)_y}{(F_2)_x} = \frac{f_y}{f_x} \\ 0 \\ \beta(x, y, w) := \frac{(F_2)_w}{(F_2)_x} \end{bmatrix};$$

and as observed in example 9, the ratio $\frac{f_y}{f_x}$ coincides with $\frac{F_y}{F_x}$. Thus, the existence of a representation of the form 3.18 is equivalent to the existence of a linear relation such as

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ F_w \end{bmatrix} = \frac{F_z}{\alpha} \begin{bmatrix} 1 \\ \frac{F_y}{F_x} \\ \alpha(x, y, z) \\ 0 \end{bmatrix} + \frac{F_w}{\beta} \begin{bmatrix} 1 \\ \frac{F_y}{F_x} \\ 0 \\ \beta(x, y, w) \end{bmatrix}.$$

This amounts to the equation

$$F_z \left(\frac{1}{\alpha}\right) + F_w \left(\frac{1}{\beta}\right) = F_x.$$

Now the idea of examples 2 and 7 applies. As $\left(\frac{1}{\alpha}\right)_w = 0$ and $\left(\frac{1}{\beta}\right)_z = 0$, applying the operators ∂_z, ∂_w, and ∂_{zw} to the last equation results in a linear system with four equations and four unknowns: $\frac{1}{\alpha}$, $\frac{1}{\beta}$, $\left(\frac{1}{\alpha}\right)_z$, and $\left(\frac{1}{\beta}\right)_w$. If nonsingular (a nonvanishing condition), the system may be solved to obtain expressions purely in terms of partial derivatives of F for the aforementioned unknowns. Now $\left(\frac{1}{\alpha}\right)_w = 0$ and $\left(\frac{1}{\beta}\right)_z = 0$, along with the equations

$F_{xz}F_y = F_{yz}F_x, F_{xw}F_y = F_{yw}F_x$ from example 9, yield four algebraic PDEs characterizing superpositions 3.18.

The final example of this section finishes example 3 from the section 1.

3.2.3 Example 11. We go back to example 3 to study PDEs and PDIs satisfied by functions of the form 1.14. Absorbing a'', b'' into inner functions, we can focus on the simpler form:

$$F(x,t) = \sigma(f(ax+bt) + g(a'x+b't)). \tag{3.20}$$

Let us for the time being forget about the outer activation function σ. Consider functions such as

$$G(x,t) = f(ax+bt) + g(a'x+b't).$$

Smooth functions of this form constitute solutions of a second-order linear homogeneous PDE with constant coefficients

$$UG_{xx} + VG_{xt} + WG_{tt} = 0, \tag{3.21}$$

where (a,b) and (a',b') satisfy

$$UA^2 + VAB + WB^2 = 0. \tag{3.22}$$

The reason is that when (a,b) and (a',b') satisfy equation 3.22, the differential operator $U\partial_{xx} + V\partial_{xt} + W\partial_{tt}$ can be factorized as

$$(b\partial_x - a\partial_t)(b'\partial_x - a'\partial_t)$$

to a composition of operators that annihilate the linear forms $ax+bt$ and $a'x+b't$. If (a,b) and (a',b') are not multiples of each other, they constitute a new coordinate system $(ax+bt, a'x+b't)$ in which the mixed partial derivatives of F all vanish; so, at least locally, F must be a sum of univariate functions of $ax+bt$ and $a'x+b't$.[7] We conclude that assuming $V^2 - 4UW > 0$, functions of the form $G(x,t) = f(ax+bt) + g(a'x+b't)$ may be identified with solutions of PDEs of the form 3.21. As in example 1, we desire algebraic PDEs purely in terms of F and without constants U, V, and W. One way to do so is to differentiate equation 3.21 further, for instance:

$$UG_{xxx} + VG_{xxt} + WG_{xtt} = 0. \tag{3.23}$$

[7] Compare with the proof of lemma 4 in appendix A.

Notice that equations 3.21 and 3.23 could be interpreted as (U, V, W) being perpendicular to (G_{xx}, G_{xt}, G_{tt}) and $(G_{xxx}, G_{xxt}, G_{xtt})$. Thus, the cross-product

$$(G_{xt}G_{xtt} - G_{tt}G_{xxt}, G_{tt}G_{xxx} - G_{xx}G_{xtt}, G_{xx}G_{xxt} - G_{xt}G_{xxx})$$

of the latter two vectors must be parallel to a constant vector. Under the nonvanishing condition that one of the entries of the cross-product, say the last one, is nonzero, the constancy may be thought of as ratios of the other two components to the last one being constants. The result is a characterization (in the vein of conjecture 1) of functions G of the form $f(ax + bt) + g(a'x + b't)$, which are subjected to $G_{xx}G_{xxt} - G_{xt}G_{xxx} \neq 0$ and

$$\frac{G_{xt}G_{xtt} - G_{tt}G_{xxt}}{G_{xx}G_{xxt} - G_{xt}G_{xxx}} \text{ and } \frac{G_{tt}G_{xxx} - G_{xx}G_{xtt}}{G_{xx}G_{xxt} - G_{xt}G_{xxx}} \text{ are constants,}$$
$$(G_{tt}G_{xxx} - G_{xx}G_{xtt})^2 > 4(G_{xt}G_{xtt} - G_{tt}G_{xxt})(G_{xx}G_{xxt} - G_{xt}G_{xxx}). \quad (3.24)$$

Notice that the PDI is not redundant here. For a solution $G = G(x, t)$ of Laplace's equation, the fractions from the first line of equation 3.24 are constants, while on the second line, the left-hand side of the inequality is zero but its right-hand side is $4(G_{xt}G_{xtt} - G_{tt}G_{xxt})^2 \geq 0$.

Composing G with σ makes the derivation of PDEs and PDIs imposed on functions of the form 3.20 even more cumbersome. We provide only a sketch. Under the assumption that the gradient of $F = \sigma \circ G$ is nonzero, the univariate function σ admits a local inverse τ. Applying the chain rule to $G = \tau \circ F$ yields

$$\begin{aligned} G_{xx} &= \tau''(F)(F_x)^2 + \tau'(F)F_{xx}, \\ G_{xt} &= \tau''(F)F_xF_t + \tau'(F)F_{xt}, \\ G_{tt} &= \tau''(F)(F_t)^2 + \tau'(F)F_{tt}. \end{aligned}$$

Plugging them in the PDE 3.21 that G satisfies results in

$$\tau''(F)\left(U(F_x)^2 + VF_xF_t + W(F_t)^2\right) + \tau'(F)\left(UF_{xx} + VF_{xt} + WF_{tt}\right) = 0,$$

or, equivalently,

$$\frac{UF_{xx} + VF_{xt} + WF_{tt}}{U(F_x)^2 + VF_xF_t + W(F_t)^2} = -\frac{\tau''(F)}{\tau'(F)} = -\left(\frac{\tau''}{\tau'}\right)(F). \quad (3.25)$$

It suffices for the ratio $\frac{UF_{xx}+VF_{xt}+WF_{tt}}{U(F_x)^2+VF_xF_t+W(F_t)^2}$ to be a function of F such as $\nu(F)$ since then τ may be recovered as $\tau = \int e^{-\int \nu}$. Following the discussion at the beginning of section 3, this is equivalent to

$$\nabla\left(\frac{UF_{xx}+VF_{xt}+WF_{tt}}{U(F_x)^2+VF_xF_t+W(F_t)^2}\right)\parallel\nabla F.$$

This amounts to an identity of the form

$$\Phi_1(F)U^2+\Phi_2(F)V^2+\Phi_3(F)W^2+\Phi_4(F)UV+\Phi_5(F)VW+\Phi_6(F)UW=0,$$

where $\Phi_i(F)$'s are complicated nonconstant polynomial expressions of partial derivatives of F. In the same way that the parameters U, V, and W in PDE 3.21 were eliminated to arrive at equation 3.24, one may solve the homogeneous linear system consisting of the identity above and its derivatives in order to derive a six-dimensional vector,

$$\big(\Xi_1(F),\,\Xi_2(F),\,\Xi_3(F),\,\Xi_4(F),\,\Xi_5(F),\,\Xi_6(F)\big) \tag{3.26}$$

of rational expressions of partial derivatives of F parallel to the constant vector

$$(U^2, V^2, W^2, UV, VW, UW). \tag{3.27}$$

The parallelism amounts to a number of PDEs, for example, $\Xi_1(F)\,\Xi_2(F)=\Xi_4(F)^2$, and the ratios $\frac{\Xi_i(F)}{\Xi_j(F)}$ must be constant because they coincide with the ratios of components of equation 3.27. Moreover, $V^2-4UW>0$ implies $\left(\frac{V^2}{UW}-4\right)\frac{V^2}{UW}\geq 0$. Replacing with the corresponding ratios of components of equation 3.26, we obtain the PDI

$$\big(\Xi_2(F)-4\,\Xi_6(F)\big)\,\Xi_2(F)\,\Xi_6(F)\geq 0,$$

which must be satisfied by any function of the form 3.20.

3.3 Examples of Polynomial Neural Networks. The superpositions we study in this section are constructed out of polynomials. Again, there are two different regimes to discuss: composing general polynomial functions of low dimensionality or composing polynomials of arbitrary dimensionality but in the simpler form of $\mathbf{y}\mapsto\sigma\left(\langle\mathbf{w},\mathbf{y}\rangle\right)$ where the activation function σ is a polynomial of a single variable. The latter regime deals with polynomial neural networks. Different aspects of such networks have been studied in the literature (Du & Lee, 2018; Soltanolkotabi et al., 2018; Venturi et al., 2018; Kileel et al., 2019). In the spirit of this article, we are interested in the spaces formed by such polynomial superpositions. Bounding the total degree of polynomials from the above, these functional spaces are subsets of an ambient polynomial space, say, the space $\mathbf{Poly}_{d,n}$ of real polynomials

$P(x_1, \ldots, x_n)$ of total degree at most d, which is an affine space of dimension $\binom{d+n}{n}$. By writing a polynomial $P(x_1, \ldots, x_n)$ of degree d as

$$P(x_1, x_2, \ldots, x_n) = \sum_{\substack{a_1, a_2, \ldots, a_n \geq 0 \\ a_1 + a_2 + \cdots + a_n \leq d}} c_{a_1, a_2, \ldots, a_n} x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}, \tag{3.28}$$

the coefficients $c_{a_1, a_2, \ldots, a_n}$ provide a natural coordinate system on $\mathbf{Poly}_{d,n}$. Associated with a neural network $\mathcal{N}$ that receives $x_1, \ldots, x_n$ as its inputs, there are polynomial functional spaces for any degree d that lie in the ambient space $\mathbf{Poly}_{d,n}$:

1. The subset $\mathbf{F}_d(\mathcal{N})$ of $\mathbf{Poly}_{d,n}$ consisting of polynomials $P(x_1, \ldots, x_n)$ of total degree at most d that can be computed by $\mathcal{N}$ via assigning real polynomial functions to its neurons
2. The smaller subset $\mathbf{F}_d^{\text{act}}(\mathcal{N})$ of $\mathbf{Poly}_{d,n}$ consisting of polynomials $P(x_1, \ldots, x_n)$ of total degree at most d that can be computed by $\mathcal{N}$ via assigning real polynomials of the form $\mathbf{y} \mapsto \sigma(\langle \mathbf{w}, \mathbf{y} \rangle)$ to the neurons where σ is a polynomial activation function

In general, subsets $\mathbf{F}_d(\mathcal{N})$ and $\mathbf{F}_d^{\text{act}}(\mathcal{N})$ of $\mathbf{Poly}_{d,n}$ are not closed in the algebraic sense (see remark 8). Therefore, one may consider their Zariski closures $\mathbf{V}_d(\mathcal{N})$ and $\mathbf{V}_d^{\text{act}}(\mathcal{N})$, that is, the smallest subsets defined as zero loci of polynomial equations that contain them. We shall call $\mathbf{V}_d(\mathcal{N})$ and $\mathbf{V}_d^{\text{act}}(\mathcal{N})$ the *functional varieties* associated with $\mathcal{N}$. Each of the subsets $\mathbf{V}_d(\mathcal{N})$ and $\mathbf{V}_d^{\text{act}}(\mathcal{N})$ of $\mathbf{Poly}_{d,n}$ could be described with finitely many polynomial equations in terms of $c_{a_1, a_2, \ldots, a_n}$'s. The PDE constraints from section 2 provide nontrivial examples of equations satisfied on the functional varieties: In any degree d, substituting equation 3.28 in an algebraic PDE that smooth functions computed by $\mathcal{N}$ must obey results in equations in terms of the coefficients that are satisfied at any point of $\mathbf{F}_d(\mathcal{N})$ or $\mathbf{F}_d^{\text{act}}(\mathcal{N})$ and hence at the points of $\mathbf{V}_d(\mathcal{N})$ or $\mathbf{V}_d^{\text{act}}(\mathcal{N})$. This will be demonstrated in example 12 and results in the following corollary to theorems 1 and 2.

Corollary 2. *Let $\mathcal{N}$ be a neural network whose inputs are labeled by the coordinate functions $x_1, \ldots, x_n$. Then there exist nontrivial polynomials on affine spaces $\mathbf{Poly}_{d,n}$ that are dependent only on the topology of $\mathcal{N}$ and become zero on functional varieties $\mathbf{V}_d^{act}(\mathcal{N}) \subset \mathbf{Poly}_{d,n}$. The same holds for functional varieties $\mathbf{V}_d(\mathcal{N})$ provided that the number of inputs to each neuron of $\mathcal{N}$ is less than n.*

Proof. The proof immediately follows from theorem 2 (in the case of $\mathbf{V}_d^{\text{act}}(\mathcal{N})$) and from theorem 1 (in the case of $\mathbf{V}_d(\mathcal{N})$). Substituting a polynomial $P(x_1, \ldots, x_n)$ in a PDE constraint

$$\Phi\left(P_{x_1}, \ldots, P_{x_n}, P_{x_1^2}, P_{x_1 x_2}, \ldots, P_{\mathbf{x}^\alpha}, \ldots\right) = 0$$

that these theorems suggest for $\mathcal{N}$ and equating the coefficient of a monomial $x_1^{a_1}x_2^{a_2}\dots x_n^{a_n}$ with zero results in a polynomial equation in ambient polynomial spaces that must be satisfied on the associated functional varieties. □

3.3.1 Example 12. Let $\mathcal{N}$ be a rooted tree $\mathcal{T}$ with distinct inputs $x_1, \dots, x_n$. Constraints of the form $F_{x_ix_k}F_{x_j} = F_{x_jx_k}F_{x_i}$ are not only necessary conditions for a smooth function $F = F(x_1, \dots, x_n)$ to be computable by $\mathcal{T}$; but by the virtue of theorem 3, they are also sufficient for the existence of a local representation of F on $\mathcal{T}$ if suitable nonvanishing conditions are satisfied. An interesting feature of this setting is that when F is a polynomial $P = P(x_1, \dots, x_n)$, one can relax the nonvanishing conditions; and P actually admits a global representation as a composition of polynomials if it satisfies the characteristic PDEs (Farhoodi et al., 2019, proposition 4). The basic idea is that if P is locally written as a superposition of smooth functions according to the hierarchy provided by $\mathcal{T}$, then comparing the Taylor series shows that the constituent parts of the superposition could be chosen to be polynomials as well. Now P and such a polynomial superposition must be the same since they agree on a nonempty open set. Consequently, each $\mathbf{F}_d(\mathcal{N})$ coincides with its closure $\mathbf{V}_d(\mathcal{N})$ and can be described by equations of the form $P_{x_ix_k}P_{x_j} = P_{x_jx_k}P_{x_i}$ in the polynomial space. Substituting an expression of the form

$$P(x_1, \dots, x_n) = \sum_{a_1,\dots,a_n \geq 0} c_{a_1,\dots,a_n}\, x_1^{a_1} \dots x_n^{a_n}$$

in $P_{x_ix_k}P_{x_j} - P_{x_jx_k}P_{x_i} = 0$ and equating the coefficient of a monomial $x_1^{a_1}\dots x_n^{a_n}$ with zero yields

$$\sum_{\substack{a_i'+a_i''=a_i+1,\, a_j'+a_j''=a_j+1,\, a_k'+a_k''=a_k+1 \\ a_s'+a_s''=a_s\ \forall s\in\{1,\dots,n\}-\{i,j,k\}}} a_k'\left(a_i'a_j'' - a_j'a_i''\right) c_{a_1',\dots,a_n'} c_{a_1'',\dots,a_n''} = 0. \tag{3.29}$$

We deduce that equations 3.29 written for $a_1, \dots, a_n \geq 0$ and for triples (i, j, k) with the property that x_k is separated from x_i and x_j by a subtree of $\mathcal{T}$ (as in theorem 3) describe the functional varieties associated with $\mathcal{T}$. In a given degree d, to obtain equations describing $\mathcal{T}$ in $\mathbf{Poly}_{d,n}$, one should set any $c_{b_1,\dots,b_n}$ with $b_1 + \dots + b_n > d$ to be zero in equation 3.29. No such a coefficient occurs if $d \geq a_1 + \dots + a_n + 3$, and thus for d large enough, equation 3.29 defines an equation in $\mathbf{Poly}_{d,n}$ as is.

Similarly, theorem 7 can be used to write equations for $\mathbf{F}_d^{\text{act}}(\mathcal{N}) = \mathbf{V}_d^{\text{act}}(\mathcal{N})$. In that situation, a new family of equations corresponding to equation 1.20 emerges that are expected to be extremely complicated.

3.3.2 Example 13. Let $\mathcal{N}$ be the neural network appearing in Figure 4. The functional space $\mathbf{F}_d^{\text{act}}(\mathcal{N})$ is formed by polynomials $P(x,t)$ of total degree at most d that are in the form of $\sigma(f(ax+bt)+g(a'x+b't))$. By examining the Taylor expansions, it is not hard to see that if $P(x,t)$ is written in this form for univariate smooth functions σ, f, and g, then these functions could be chosen to be polynomials. Therefore, in any degree d, our characterization of superpositions of this form in example 11 in terms of PDEs and PDIs results in polynomial equations and inequalities that describe a Zariski open subset of $\mathbf{F}_d^{\text{act}}(\mathcal{N})$ which is the complement of the locus where the nonvanishing conditions fail. The inequalities disappear after taking the closure, so $\mathbf{V}_d^{\text{act}}(\mathcal{N})$ is strictly larger than $\mathbf{F}_d^{\text{act}}(\mathcal{N})$ here.

Remark 8. The emergence of inequalities in describing the functional spaces, as observed in example 13, is not surprising due to the Tarski–Seidenberg theorem (see Coste, 2000), which implies that the image of a polynomial map between real varieties (i.e., a map whose components are polynomials) is semialgebraic; that is, it could be described as a union of finitely many sets defined by polynomial equations and inequalities. To elaborate, fix a neural network architecture $\mathcal{N}$. Composing polynomials of bounded degrees according to the hierarchy provided by $\mathcal{N}$ yields polynomial superpositions lying in $\mathbf{F}_D(\mathcal{N})$ for D sufficiently large. The composition thus amounts to a map

$$\mathbf{Poly}_{d_1,n_1} \times \cdots \times \mathbf{Poly}_{d_N,n_N} \to \mathbf{Poly}_{D,n},$$

where, on the left-hand side, the polynomials assigned to the neurons of $\mathcal{N}$ appear, and $D \gg d_1, \ldots, d_n$. The image, a subset of $\mathbf{F}_D(\mathcal{N})$, is semialgebraic and thus admits a description in terms of finitely many polynomial equations and inequalities. The same logic applies to the regime of activation functions too; the map just mentioned must be replaced with

$$\mathbb{R}^C \times \mathbf{Poly}_{d_1,1} \times \cdots \times \mathbf{Poly}_{d_N,1} \to \mathbf{Poly}_{D,n}$$

whose image lies in $\mathbf{F}_D^{\text{act}}(\mathcal{N})$, and its domain is the Cartesian product of spaces of polynomial activation functions assigned to the neurons by the space $\mathbb{R}^C$ of weights assigned to the connections of the network.

4 PDE Characterization of Tree Functions

Building on the examples of the previous section, we prove theorems 3 and 4. This will establish conjecture 1 for tree architectures with distinct inputs.

Proof of Theorem 3. The necessity of the constraints from equation 1.18 follows from example 1. As demonstrated in Figure 12, picking three of variables $x_i = x$, $x_j = y$, and $x_k = z$ where the former two are separated from the

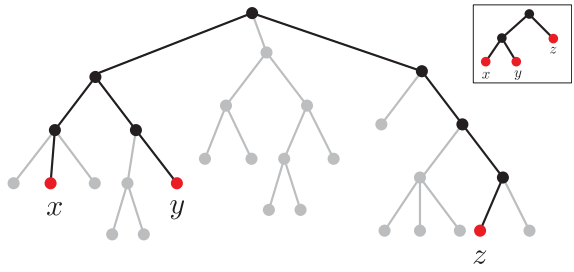

Figure 12: The necessity of constraint 1.18 in theorem 3 follows from the case of trivariate tree functions discussed in example 1. Choosing three of the variables (red leaves) and fixing the rest (gray leaves) results in a superposition of the form $g(f(x, y), z)$ that must obey constraint 1.4.

latter by a subtree and taking the rest of variables to be constant, we obtain a superposition of the form $F(x, y, z) = g(f(x, y), z)$ studied in example 1; it should satisfy $F_{xz}F_y = F_{yz}F_x$ or, equivalently, equation 1.18.

We induct on the number of variables, which coincides with the number of leaves, to prove the sufficiency of constraint 1.18 and the nonvanishing conditions in theorem 3 for the existence of a local implementation, in the form of a superposition of functions of lower arity, on the tree architecture in hand. Consider a rooted tree $\mathcal{T}$ with n leaves labeled by the coordinate functions $x_1, \ldots, x_n$. The inductive step is illustrated in Figure 13. Removing the root results in a number of smaller trees $\mathcal{T}_1, \ldots, \mathcal{T}_l$ and a number of single vertices[8] corresponding to the leaves adjacent to the root of $\mathcal{T}$. By renumbering $x_1, \ldots, x_n$ one may write the leaves as

$$x_1, \ldots, x_{m_1};\ x_{m_1+1}, \ldots, x_{m_1+m_2};\ \ldots;\ x_{m_1+\cdots+m_{l-1}+1}, \ldots, x_{m_1+\cdots+m_l};$$
$$x_{m_1+\cdots+m_l+1};\ \ldots;\ x_n, \tag{4.1}$$

where $x_{m_1+\cdots+m_{s-1}+1}, \ldots, x_{m_1+\cdots+m_{s-1}+m_s}$ $(1 \le s \le l)$ are the leaves of the subtree $\mathcal{T}_s$ while $x_{m_1+\cdots+m_l+1}$ through x_n are the leaves adjacent to the root of $\mathcal{T}$. The goal is to write $F(x_1, \ldots, x_n)$ as

$$g(G_1(x_1, \ldots, x_{m_1}), \ldots, G_l(x_{m_1+\cdots+m_{l-1}+1}, \ldots, x_{m_1+\cdots+m_l}), x_{m_1+\cdots+m_l+1}, \ldots, x_n), \tag{4.2}$$

where each smooth function,

$$G_s(x_{m_1+\cdots+m_{s-1}+1}, \ldots, x_{m_1+\cdots+m_{s-1}+m_s}),$$

[8] A single vertex is not considered to be a rooted tree in our convention.

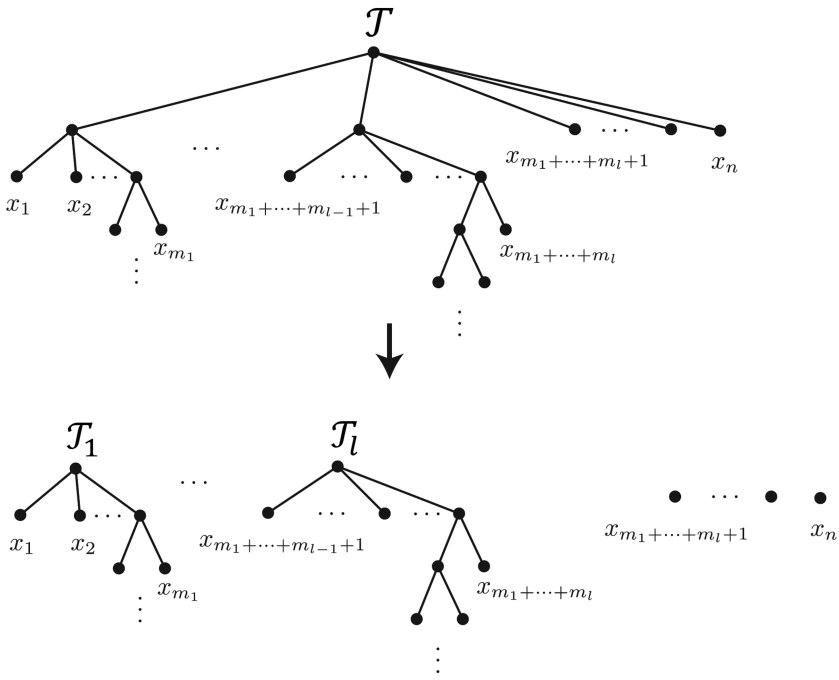

Figure 13: The inductive step in the proof of theorem 3. The removal of the root of $\mathcal{T}$ results in a number of smaller rooted trees along with single vertices that were the leaves adjacent to the root of $\mathcal{T}$ (if any).

satisfies the constraints coming from $\mathcal{T}_s$ and thus, by invoking the induction hypothesis, is computable by the tree $\mathcal{T}_s$. Following the discussion before theorem 5, it suffices to express ∇F as a linear combination of the gradients $\nabla G_1, \ldots, \nabla G_l, \nabla x_{m_1+\cdots+m_l+1}, \ldots, \nabla x_n$. The nonvanishing conditions in theorem 3 require the first-order partial derivative with respect to at least one of the leaves of each $\mathcal{T}_s$ to be nonzero; we may assume $F_{x_{m_1+\cdots+m_{s-1}+1}} \neq 0$ without any loss of generality. We should have

$$\nabla F = \left[F_{x_1} \ \ldots \ F_{x_{m_1}} \ \ldots \ F_{x_{m_1+\cdots+m_{l-1}+1}} \ \ldots \ F_{x_{m_1+\cdots+m_l}} \ F_{x_{m_1+\cdots+m_l+1}} \ \ldots \ F_{x_n} \right]^{\mathrm{T}}$$

$$= \sum_{s=1}^{l} F_{x_{m_1+\cdots+m_{s-1}+1}} \left[\overbrace{0 \cdots 0}^{m_1+\cdots+m_{s-1}} \ 1 \ \frac{F_{x_{m_1+\cdots+m_{s-1}+2}}}{F_{x_{m_1+\cdots+m_{s-1}+1}}} \ \cdots \ \frac{F_{x_{m_1+\cdots+m_{s-1}+m_s}}}{F_{x_{m_1+\cdots+m_{s-1}+1}}} \ \overbrace{0 \cdots 0}^{n-(m_1+\cdots+m_s)} \right]^{\mathrm{T}}$$

$$+ F_{x_{m_1+\cdots+m_l+1}} \frac{\partial}{\partial x_{m_1+\cdots+m_l+1}} + \cdots + F_{x_n} \frac{\partial}{\partial x_n}$$

$$\in \mathrm{Span}\{\nabla G_1(x_1,\dots,x_{m_1}),\dots,\nabla G_l(x_{m_1+\cdots+m_{l-1}+1},\dots,x_{m_1+\cdots+m_l}),\ \nabla x_{m_1+\cdots+m_l+1},\dots,\nabla x_n\}.$$

In the expressions above, the vector $\begin{bmatrix}1 & \frac{F_{x_{m_1+\cdots+m_{s-1}+2}}}{F_{x_{m_1+\cdots+m_{s-1}+1}}} & \cdots & \frac{F_{x_{m_1+\cdots+m_{s-1}+m_s}}}{F_{x_{m_1+\cdots+m_{s-1}+1}}}\end{bmatrix}^{\mathrm{T}}$ (which is of size m_s) is dependent only on the variables $x_{m_1+\cdots+m_{s-1}+1},\dots,x_{m_1+\cdots+m_s}$, which are the leaves of $\mathcal{T}_s$: any other leaf x_k is separated from them by the subtree $\mathcal{T}_s$ of $\mathcal{T}$, and, hence, for any leaf x_i with $m_1+\cdots+m_{s-1}<i\le m_1+\cdots+m_s$, we have $\left(\frac{F_{x_i}}{F_{x_{m_1+\cdots+m_{s-1}+1}}}\right)_{x_k}=0$ due to the simplified form 1.21 of equation 1.18. To finish the proof, one should establish the existence of functions $G_s(x_{m_1+\cdots+m_{s-1}+1},\dots,x_{m_1+\cdots+m_s})$ appearing in equation 4.2; that is, $\begin{bmatrix}1 & \frac{F_{x_{m_1+\cdots+m_{s-1}+2}}}{F_{x_{m_1+\cdots+m_{s-1}+1}}} & \cdots & \frac{F_{x_{m_1+\cdots+m_{s-1}+m_s}}}{F_{x_{m_1+\cdots+m_{s-1}+1}}}\end{bmatrix}^{\mathrm{T}}$ should be shown to be parallel to a gradient vector field ∇G_s. Notice that the induction hypothesis would be applicable to G_s since any ratio $\frac{(G_s)_{x_i}}{(G_s)_{x_j}}$ of partial derivatives is the same as the corresponding ratio of partial derivatives of F. Invoking theorem 5, to prove the existence of G_s, we should verify that the 1-form

$$\omega_s:=\sum_{i=m_1+\cdots+m_{s-1}+1}^{m_1+\cdots+m_{s-1}+m_s}\frac{F_{x_i}}{F_{x_{m_1+\cdots+m_{s-1}+1}}}\,\mathrm{d}x_i\quad(1\le s\le l)$$

satisfies $\omega_s\wedge\mathrm{d}\omega_s=0$. We finish the proof by showing this in the case of $s=1$; other cases are completely similar. We have

$$\begin{aligned}
\omega_1\wedge\mathrm{d}\omega_1&=\left(\sum_{i=1}^{m_1}\frac{F_{x_i}}{F_{x_1}}\,\mathrm{d}x_i\right)\wedge\left(\sum_{j=1}^{m_1}\mathrm{d}\left(\frac{F_{x_j}}{F_{x_1}}\right)\wedge\mathrm{d}x_j\right)\\
&=\left(\sum_{i=1}^{m_1}\frac{F_{x_i}}{F_{x_1}}\,\mathrm{d}x_i\right)\wedge\left(\sum_{j=1}^{m_1}\left(\sum_{k=1}^{m_1}\left(\frac{F_{x_j}}{F_{x_1}}\right)_{x_k}\mathrm{d}x_k\right)\wedge\mathrm{d}x_j\right)\\
&=\sum_{i,j,k\in\{1,\dots,m_1\}}\left[\frac{F_{x_i}F_{x_jx_k}}{(F_{x_1})^2}-\frac{F_{x_i}F_{x_j}F_{x_1x_k}}{(F_{x_1})^3}\right]\mathrm{d}x_i\wedge\mathrm{d}x_k\wedge\mathrm{d}x_j\\
&=\left(\sum_{i=1}^{m_1}\frac{F_{x_i}}{(F_{x_1})^2}\,\mathrm{d}x_i\right)\wedge\left(\sum_{j,k\in\{1,\dots,m_1\}}F_{x_jx_k}\,\mathrm{d}x_k\wedge\mathrm{d}x_j\right)\\
&\quad+\left(\sum_{i,j\in\{1,\dots,m_1\}}F_{x_i}F_{x_j}\,\mathrm{d}x_i\wedge\mathrm{d}x_j\right)\wedge\left(\sum_{k=1}^{m_1}\frac{F_{x_1x_k}}{(F_{x_1})^3}\,\mathrm{d}x_k\right).
\end{aligned}$$

The last two terms are zero because, in the parentheses, the 2-forms

$$\sum_{j,k\in\{1,\dots,m_1\}} F_{x_j x_k}\,\mathrm{d}x_j\wedge\mathrm{d}x_k,\qquad \sum_{i,j\in\{1,\dots,m_1\}} F_{x_i}F_{x_j}\,\mathrm{d}x_i\wedge\mathrm{d}x_j,$$

are zero since interchanging j and k or i and j in the summations results in the opposite of the original differential form. □

Remark 9. The formulation of theorem 3 in Farhoodi et al. (2019) is concerned with analytic functions and binary trees. The proof presented above follows the same inductive procedure but utilizes theorem 5 instead of Taylor expansions. Of course, theorem 5 remains valid in the analytic category, so the tree representation of F constructed in the proof here consists of analytic functions if F is analytic. An advantage of working with analytic functions is that in certain cases, the nonvanishing conditions may be relaxed. For instance, if in example 1 the function $F(x, y, z)$ satisfying equation 1.4 is analytic, it admits a local representation of the form 1.1, while if F is only smooth, at least one of the conditions $F_x \neq 0$ of $F_y \neq 0$ is required. (See Farhoodi et al., 2019, sec. 5.1 and 5.3, for details.)

Proof of Theorem 4. Establishing the necessity of constraints 1.19 and 1.20 is straightforward. An implementation of a smooth function $F = F(x_1, \dots, x_n)$ on the tree $\mathcal{T}$ is in a form such as

$$\begin{aligned}\sigma\Bigg(\cdots\Bigg(\tilde{w}.\tilde{\sigma}\Bigg(w_1.\tau_1\Big(\cdots\Big(\tilde{w}_1.\tilde{\tau}_1(cx_i+\cdots)+\tilde{\tilde{w}}_1.\tilde{\tilde{\tau}}_1(c'x_{i'}+\cdots)+\cdots\Big)\cdots\Big)\\ +\,w_2.\tau_2\Big(\cdots\Big(\tilde{w}_2.\tilde{\tau}_2(dx_j+\cdots)+\tilde{\tilde{w}}_2.\tilde{\tilde{\tau}}_2(d'x_{j'}+\cdots)+\cdots\Big)\cdots\Big)\\ +\,w_3.\tau_3\Big(\cdots\Big)+\cdots\Bigg)+\cdots\Bigg)\cdots\Bigg)\end{aligned}\tag{4.3}$$

for appropriate activation functions and weights. In the expression above, variables x_s appearing in

$$\begin{aligned}\tilde{\sigma}\Bigg(w_1.\tau_1\Big(\cdots\Big(\tilde{w}_1.\tilde{\tau}_1(cx_i+\cdots)+\tilde{\tilde{w}}_1.\tilde{\tilde{\tau}}_1(c'x_{i'}+\cdots)+\cdots\Big)\cdots\Big)\\ +\,w_2.\tau_2\Big(\cdots\Big(\tilde{w}_2.\tilde{\tau}_2(dx_j+\cdots)+\tilde{\tilde{w}}_2.\tilde{\tilde{\tau}}_2(d'x_{j'}+\cdots)+\cdots\Big)\cdots\Big)\\ +\,w_3.\tau_3\Big(\cdots\Big)+\cdots\Bigg)\end{aligned}$$

are the leaves of the smallest (full) subtree of $\mathcal{T}$ in which both x_i and x_j appear as leaves. Denoting this subtree by $\widetilde{\mathcal{T}}$, the activation function applied

at the root of $\widetilde{\mathcal{T}}$ is $\tilde{\sigma}$, and the subtrees emanating from the root of $\widetilde{\mathcal{T}}$, which we write as $\widetilde{\mathcal{T}}_1, \widetilde{\mathcal{T}}_2, \widetilde{\mathcal{T}}_3, \ldots$, have $\tau_1, \tau_2, \tau_3, \ldots$ assigned to their roots. Here, $\widetilde{\mathcal{T}}_1$ and $\widetilde{\mathcal{T}}_2$ contain x_i and x_j, respectively, and are the largest (full) subtrees that have exactly one of x_i and x_j. To verify equation 1.19, notice that $\frac{F_{x_i}}{F_{x_j}}$ is proportional to

$$\frac{\tau_1'\Big(\cdots\Big(\tilde{w}_1.\tilde{\tau}_1(cx_i+\cdots)+\tilde{\tilde{w}}_1.\tilde{\tilde{\tau}}_1(c'x_{i'}+\cdots)+\cdots\Big)\cdots\Big)\ldots\tilde{\tau}_1'(cx_i+\cdots)}{\tau_2'\Big(\cdots\Big(\tilde{w}_2.\tilde{\tau}_2(dx_j+\cdots)+\tilde{\tilde{w}}_2.\tilde{\tilde{\tau}}_2(d'x_{j'}+\cdots)+\cdots\Big)\cdots\Big)\ldots\tilde{\tau}_2'(dx_j+\cdots)} \tag{4.4}$$

with the constant of proportionality being a quotient of two products of certain weights of the network. The ratio 4.4 is dependent only on those variables that appear as leaves of $\widetilde{\mathcal{T}}_1$ and $\widetilde{\mathcal{T}}_2$, so

$$\left(\frac{F_{x_i}}{F_{x_j}}\right)_{x_k} = 0 \Leftrightarrow F_{x_ix_k}F_{x_j} = F_{x_jx_k}F_{x_i}$$

unless there is a subtree of $\mathcal{T}$ containing the leaf x_k and exactly one of x_i or x_j (which forcibly will be a subtree of $\widetilde{\mathcal{T}}_1$ or $\widetilde{\mathcal{T}}_2$). Before switching to constraint 1.20, we point out that the description of F in equation 4.3 assumes that the leaves x_i and x_j are not siblings. If they are, F may be written as

$$\sigma\Big(\cdots\Big(\tilde{w}.\tilde{\sigma}\big(w.\tau(cx_i+dx_j+\cdots)+\cdots\big)+\cdots\Big)\cdots\Big),$$

in which case, $\frac{F_{x_i}}{F_{x_j}} = \frac{c}{d}$ is a constant and hence equation 1.21 holds for all $1 \le k \le n$. To finish the proof of necessity of the constraints introduced in theorem 7, consider the fraction from equation 4.4, which is a multiple of $\frac{F_{x_i}}{F_{x_j}}$. This has a description as a product of a function of $x_i, x_{i'}, \ldots$ (leaves of $\widetilde{\mathcal{T}}_1$) by a function of $x_j, x_{j'}, \ldots$ (leaves of $\widetilde{\mathcal{T}}_2$). Lemma 4 now implies that for any leaf $x_{i'}$ of $\widetilde{\mathcal{T}}_1$ and any leaf $x_{j'}$ of $\widetilde{\mathcal{T}}_2$,

$$\left(\frac{\left(\frac{F_{x_i}}{F_{x_j}}\right)_{x_{i'}}}{\frac{F_{x_i}}{F_{x_j}}}\right)_{x_{j'}} = 0;$$

hence, the simplified form, equation 1.22 of 1.20.

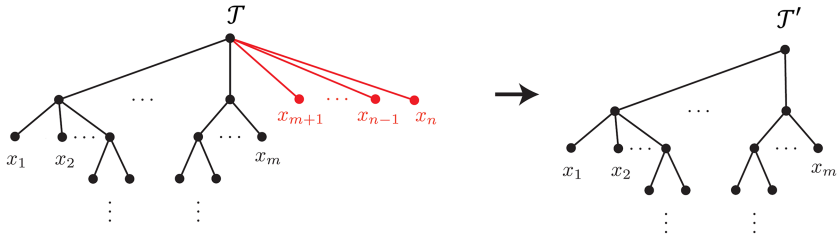

Figure 14: The first case of the inductive step in the proof of theorem 4. The removal of the leaves directly connected to the root of $\mathcal{T}$ results in a smaller rooted tree.

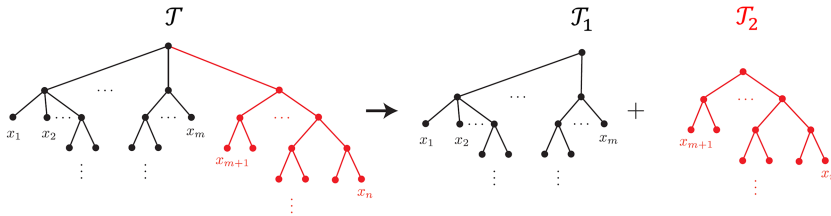

Figure 15: The second case of the inductive step in the proof of theorem 4. There is no leaf directly connected to the root of $\mathcal{T}$. Separating one of the rooted subtrees adjacent to the root results in two smaller rooted trees.

We induct on the number of leaves to prove the sufficiency of constraints 1.19 and 1.20 (accompanied by suitable nonvanishing conditions) for the existence of a tree implementation of a smooth function $F = F(x_1, \ldots, x_n)$ as a composition of functions of the form 1.12. Given a rooted tree $\mathcal{T}$ with n leaves labeled by $x_1, \ldots, x_n$, the inductive step has two cases demonstrated in Figures 14 and 15:

- There are leaves, say, $x_{m+1}, \ldots, x_n$, directly adjacent to the root of $\mathcal{T}$; their removal results in a smaller tree $\mathcal{T}'$ with leaves $x_1, \ldots, x_m$ (see Figure 14). The goal is to write $F(x_1, \ldots, x_n)$ as

$$\sigma(G(x_1, \ldots, x_m) + c_{m+1}x_{m+1} + \cdots + c_n x_n), \tag{4.5}$$

  with G satisfying appropriate constraints that, invoking the induction hypothesis, guarantee that G is computable by $\mathcal{T}'$.
- There is no leaf adjacent to the root of $\mathcal{T}$, but there are smaller subtrees. Denote one of them with $\mathcal{T}_2$ and show its leaves by $x_{m+1}, \ldots, x_n$. Removing this subtree results in a smaller tree $\mathcal{T}_1$ with leaves

$x_1, \ldots, x_m$ (see Figure 15). The goal is to write $F(x_1, \ldots, x_n)$ as

$$\sigma(G_1(x_1, \ldots, x_m) + G_2(x_{m+1}, \ldots, x_n)), \tag{4.6}$$

with G_1 and G_2 satisfying constraints corresponding to $\mathcal{T}_1$ and $\mathcal{T}_2$, and hence may be implemented on these trees by invoking the induction hypothesis.

Following the discussion at the beginning of section 3, F may be locally written as a function of another function with nonzero gradient if the gradients are parallel. This idea has been frequently used so far, but there is a twist here: we want such a description of F to persist on the box-like region B that is the domain of F. Lemma 2 resolves this issue. The tree function in the argument of σ in either of equation 4.5 or 4.6, which here we denote by $\tilde{F}$, shall be constructed below by invoking the induction hypothesis, so $\tilde{F}$ is defined at every point of B. Besides, our description of $\nabla\tilde{F}$ below (cf. equations 4.7 and 4.9) readily indicates that just like F, it satisfies the nonvanishing conditions of theorem 7. Applying lemma 2 to $\tilde{F}$, any level set $\{\mathbf{x} \in B \mid \tilde{F}(\mathbf{x}) = c\}$ is connected, and $\tilde{F}$ can be extended to a coordinate system $(\tilde{F}, F_2, \ldots, F_n)$ for B. Thus, F, whose partial derivatives with respect to other coordinate functions vanish, realizes precisely one value on any coordinate hypersurface $\{\mathbf{x} \in B \mid \tilde{F}(\mathbf{x}) = c\}$. Setting $\sigma(c)$ to be the aforementioned value of F defines a function σ with $F = \sigma(\tilde{F})$. After this discussion on the domain of definition of the desired representation of F, we proceed with constructing $\tilde{F} = \tilde{F}(x_1, \ldots, x_n)$ as either $G(x_1, \ldots, x_m) + c_{m+1}x_{m+1} + \cdots + c_n x_n$ in the case of equation 4.5 or as $G(x_1, \ldots, x_m) + G_2(x_{m+1}, \ldots, x_n)$ in the case of equation 4.6.

In the case of equation 4.5, assuming that, as theorem 7 requires, one of the partial derivatives $F_{x_{m+1}}, \ldots, F_{x_n}$, example F_{x_n}, is nonzero, we should have

$$\begin{aligned}
\nabla F &= \begin{bmatrix} F_{x_1} & \ldots & F_{x_m} & F_{x_{m+1}} & \ldots & F_{x_{n-1}} & F_{x_n} \end{bmatrix}^{\mathrm{T}} \Big\| \begin{bmatrix} \frac{F_{x_1}}{F_{x_n}} & \cdots & \frac{F_{x_m}}{F_{x_n}} & \frac{F_{x_{m+1}}}{F_{x_n}} & \cdots & \frac{F_{x_{n-1}}}{F_{x_n}} & 1 \end{bmatrix}^{\mathrm{T}} \\
&= \begin{bmatrix} G_{x_1} & \cdots & G_{x_m} & c_{m+1} & \cdots c_{n-1} & 1 \end{bmatrix}^{\mathrm{T}} \\
&= \nabla(G(x_1, \ldots, x_m) + c_{m+1}x_{m+1} + \cdots + c_{n-1}x_{n-1} + x_n).
\end{aligned} \tag{4.7}$$

Here, each ratio $\frac{F_{x_j}}{F_{x_n}}$ where $m < j \leq n$ must be a constant, which we show by c_j, due to the simplified form equation 1.21 of equation 1.19: the only (full) subtree of $\mathcal{T}$ containing either x_j or x_n is the whole tree since these leaves are adjacent to the root of $\mathcal{T}$. On the other hand, $\begin{bmatrix} \frac{F_{x_1}}{F_{x_n}} & \cdots & \frac{F_{x_m}}{F_{x_n}} \end{bmatrix}^{\mathrm{T}}$ appearing in equation 4.7 is a gradient vector field of the form $\nabla G(x_1, \ldots, x_m)$ again as a by-product of equations 1.19 and 1.21: each ratio $\frac{F_{x_i}}{F_{x_n}}$ where $1 \leq i \leq m$ is

independent of $x_{m+1}, \ldots, x_n$ by the same reasoning as above; and this vector function of $(x_1, \ldots, x_m)$ is integrable because for any $1 \leq i, i' \leq m$,

$$\left(\frac{F_{x_i}}{F_{x_n}}\right)_{x_{i'}} = \left(\frac{F_{x_{i'}}}{F_{x_n}}\right)_{x_i} \Leftrightarrow F_{x_i x_n} F_{x_{i'}} = F_{x_{i'} x_n} F_{x_i}.$$

Hence, such a $G(x_1, \ldots, x_m)$ exists; moreover, it satisfies constraints from the inductions hypothesis since any ratio $\frac{G_{x_j}}{G_{x_{j'}}}$ coincides with the corresponding ratio of partial derivatives of F, a function assumed to satisfy equations 1.21 and 1.22.

Next, in the second case of the inductive step, let us turn to equation 4.6. The nonvanishing conditions of theorem 7 require a partial derivative among $F_{x_1}, \ldots, F_{x_m}$ and also a partial derivative among $F_{x_{m+1}}, \ldots, F_{x_n}$ to be nonzero. Without any loss of generality, we assume $F_{x_1} \neq 0$ and $F_{x_n} \neq 0$. We want to apply lemma 4 to split the ratio $\frac{F_{x_1}}{F_{x_n}} \neq 0$ as

$$\frac{F_{x_1}}{F_{x_n}} = \beta(x_1, \ldots, x_m) \frac{1}{\gamma}(x_{m+1}, \ldots, x_n) = \frac{\beta(x_1, \ldots, x_m)}{\gamma(x_{m+1}, \ldots, x_n)}. \tag{4.8}$$

To do so, it needs to be checked that

$$\left(\frac{\left(\frac{F_{x_1}}{F_{x_n}}\right)_{x_i}}{\frac{F_{x_1}}{F_{x_n}}}\right)_{x_j} = 0$$

for any two indices $1 \leq i \leq m$ and $m < j \leq n$. This is the content of equation 1.20, or its simplified form, equation 1.22, when x_i belongs to the same maximal subtree of $\mathcal{T}$ adjacent to the root that has x_1 and holds for other choices of $x_i \in \{x_1, \ldots, x_m\}$ too since in that situation, by the simplified form equation 1.21, of equation 1.19, the derivative $\left(\frac{F_{x_1}}{F_{x_n}}\right)_{x_i}$ must be zero because x_1, x_i, and x_n belong to different maximal subtrees of $\mathcal{T}$. Next, the gradient of F could be written as

$$\begin{aligned}
\nabla F &= \begin{bmatrix} F_{x_1} & F_{x_2} & \ldots & F_{x_m} & F_{x_{m+1}} & \ldots & F_{x_{n-1}} & F_{x_n} \end{bmatrix}^{\mathrm{T}} \\
&\quad \| \begin{bmatrix} \frac{F_{x_1}}{F_{x_n}} & \frac{F_{x_2}}{F_{x_n}} & \cdots & \frac{F_{x_m}}{F_{x_n}} & \frac{F_{x_{m+1}}}{F_{x_n}} & \cdots & \frac{F_{x_{n-1}}}{F_{x_n}} & 1 \end{bmatrix}^{\mathrm{T}} \\
&= \begin{bmatrix} \frac{F_{x_1}}{F_{x_n}} & \frac{F_{x_2}}{F_{x_1}} \cdot \frac{F_{x_1}}{F_{x_n}} & \cdots & \frac{F_{x_m}}{F_{x_1}} \cdot \frac{F_{x_1}}{F_{x_n}} & \frac{F_{x_{m+1}}}{F_{x_n}} & \cdots & \frac{F_{x_{n-1}}}{F_{x_n}} & 1 \end{bmatrix}^{\mathrm{T}}
\end{aligned}$$

$$= \frac{F_{x_1}}{F_{x_n}} \left[1 \ \frac{F_{x_2}}{F_{x_1}} \ \cdots \ \frac{F_{x_m}}{F_{x_1}} \ \overbrace{0 \cdots 0}^{n-m} \right]^{\mathrm{T}} + \left[\overbrace{0 \cdots 0}^{m} \ \frac{F_{x_{m+1}}}{F_{x_n}} \ \cdots \ \frac{F_{x_{n-1}}}{F_{x_n}} \ 1 \right]^{\mathrm{T}}.$$

Combining with equation 4.8:

$$\nabla F \parallel \left[\beta(x_1, \ldots, x_m) \ \beta(x_1, \ldots, x_m).\frac{F_{x_2}}{F_{x_1}} \ \cdots \ \beta(x_1, \ldots, x_m).\frac{F_{x_m}}{F_{x_1}} \ \overbrace{0 \cdots 0}^{n-m} \right]^{\mathrm{T}}$$
$$+ \left[\overbrace{0 \cdots 0}^{m} \ \gamma(x_{m+1}, \ldots, x_n).\frac{F_{x_{m+1}}}{F_{x_n}} \ \cdots \ \gamma(x_{m+1}, \ldots, x_n) \right.$$
$$\left. .\frac{F_{x_{n-1}}}{F_{x_n}} \ \gamma(x_{m+1}, \ldots, x_n) \right]^{\mathrm{T}}. \tag{4.9}$$

To establish equation 4.6, it suffices to argue that the vectors on the right-hand side are in the form of ∇G_1 and ∇G_2 for suitable functions $G_1(x_1, \ldots, x_m)$ and $G_2(x_{m+1}, \ldots, x_n)$, to which the induction hypothesis can be applied by the same logic as before. Notice that the first one is dependent only on $x_1, \ldots, x_m$, while the second one is dependent only on $x_{m+1}, \ldots, x_n$, again by equations 1.19 and 1.21. For any $1 \leq i \leq m$ and $m < j \leq n$, we have $\left(\frac{F_{x_i}}{F_{x_1}}\right)_{x_j} = 0$ (respectively, $\left(\frac{F_{x_j}}{F_{x_n}}\right)_{x_i} = 0$) since there is no subtree of $\mathcal{T}$ that has only one of x_1 and x_i (resp. only one of x_n and x_j) and also x_j (resp. also x_i). We finish the proof by verifying the corresponding integrability conditions,

$$\left(\beta \frac{F_{x_i}}{F_{x_1}}\right)_{x_{i'}} = \left(\beta \frac{F_{x_{i'}}}{F_{x_1}}\right)_{x_i}, \quad \left(\gamma \frac{F_{x_j}}{F_{x_n}}\right)_{x_{j'}} = \left(\gamma \frac{F_{x_{j'}}}{F_{x_n}}\right)_{x_j},$$

for any $1 \leq i, i' \leq m$ and $m < j, j' \leq n$. In view of equation 4.8, one can change β and γ above to $\frac{F_{x_1}}{F_{x_n}}$ or $\frac{F_{x_n}}{F_{x_1}}$, respectively, and write the desired identities as the new ones,

$$\left(\frac{\cancel{F_{x_1}}}{\cancel{F_{x_n}}} \frac{F_{x_i}}{\cancel{F_{x_1}}}\right)_{x_{i'}} = \left(\frac{\cancel{F_{x_1}}}{\cancel{F_{x_n}}} \frac{F_{x_{i'}}}{\cancel{F_{x_1}}}\right)_{x_i}, \quad \left(\frac{\cancel{F_{x_n}}}{\cancel{F_{x_1}}} \frac{F_{x_j}}{\cancel{F_{x_n}}}\right)_{x_{j'}} = \left(\frac{\cancel{F_{x_n}}}{\cancel{F_{x_1}}} \frac{F_{x_{j'}}}{\cancel{F_{x_n}}}\right)_{x_j},$$

which hold due to equation 1.21. □

Remark 10. As mentioned in remark 3, working with functions of the form 1.12 in theorem 7 rather than general smooth functions has the advantage of enabling us to determine a domain on which a superposition representation exists. In contrast, the sufficiency part of theorem 3 is a local statement since

it relies on the implicit function theorem. It is possible to say something nontrivial about the domains when functions are furthermore analytic. This is because the implicit function theorem holds in the analytic category as well (Krantz & Parks, 2002, sec. 6.1) where lower bounds on the domain of validity of the theorem exist in the literature (Chang, He, & Prabhu, 2003).

5 Conclusion

In this article, we proposed a systematic method for studying smooth real-valued functions constructed as compositions of other smooth functions that are either of lower arity or in the form of a univariate activation function applied to a linear combination of inputs. We established that any such smooth superposition must satisfy nontrivial constraints in the form of algebraic PDEs, which are dependent only on the hierarchy of composition or, equivalently, only on the topology of the neural network that produces superpositions of this type. We conjectured that there always exist characteristic PDEs that also provide sufficient conditions for a generic smooth function to be expressible by the feedforward neural network in question. The genericity is to avoid singular cases and is captured by nonvanishing conditions that require certain polynomial functions of partial derivatives to be nonzero. We observed that there are also situations where nontrivial algebraic inequalities involving partial derivatives (PDIs) are imposed on the hierarchical functions. In summary, the conjecture aims to describe generic smooth functions computable by a neural network with finitely many universal conditions of the form $\Phi \neq 0$, $\Psi = 0$, and $\Theta > 0$, where Φ, Ψ, and Θ are polynomial expressions of the partial derivatives and are dependent only on the architecture of the network, not on any tunable parameter or any activation function used in the network. This is reminiscent of the notion of a semialgebraic set from real algebraic geometry. Indeed, in the case of compositions of polynomial functions or functions computed by polynomial neural networks, the PDE constraints yield equations for the corresponding functional variety in an ambient space of polynomials of a prescribed degree.

The conjecture was verified in several cases, most importantly, for tree architectures with distinct inputs where, in each regime, we explicitly exhibited a PDE characterization of functions computable by a tree network. Examples of tree architectures with repeated inputs were addressed as well. The proofs were mathematical in nature and relied on classical results of multivariable analysis.

The article moreover highlights the differences between the two regimes mentioned at the beginning: the hierarchical functions constructed out of composing functions of lower dimensionality and the hierarchical functions that are compositions of functions of the form $\mathbf{y} \mapsto \sigma(\langle \mathbf{w}, \mathbf{y} \rangle)$. The former functions appear more often in the mathematical literature on the Kolmogorov-Arnold representation theorem, while the latter are

ubiquitous in deep learning. The special form of functions $\mathbf{y} \mapsto \sigma(\langle \mathbf{w}, \mathbf{y} \rangle)$ requires more PDE constraints to be imposed on their compositions, whereas their mild nonlinearity is beneficial in terms of ascertaining the domain on which a claimed compositional representation exists.

Our approach for describing the functional spaces associated with feedforward neural networks is of natural interest in the study of expressivity of neural networks and could lead to new complexity measures. We believe that the point of view adapted here is novel and might shed light on a number of practical problems such as comparison of architectures and reverse-engineering deep networks.

Appendix A: Technical Proofs

Proof of Lemma 2. We first prove that F can be extended to a coordinate system on the entirety of the box-like region B, which we shall write as $I_1 \times \cdots \times I_n$. As in the proof of theorem 3, we group the variables $x_1, \ldots, x_n$ according to the maximal subtrees of $\mathcal{T}$ in which they appear:

$$x_1, \ldots, x_{m_1};\ x_{m_1+1}, \ldots, x_{m_1+m_2};\ \ldots;\ x_{m_1+\cdots+m_{l-1}+1}, \ldots, x_{m_1+\cdots+m_l};$$

$$x_{m_1+\cdots+m_l+1};\ \ldots;\ x_n,$$

where, denoting the subtrees emanating from the root of $\mathcal{T}$ by $\mathcal{T}_1, \ldots, \mathcal{T}_l$, for any $1 \le s \le l$ the leaves of $\mathcal{T}_s$ are labeled by $x_{m_1+\cdots+m_{s-1}+1}, \ldots, x_{m_1+\cdots+m_{s-1}+m_s}$; and $x_{m_1+\cdots+m_l+1}, \ldots, x_n$ represent the leaves that are directly connected to the root (if any); see Figure 13. Among the variables labeling the leaves of $\mathcal{T}_1$, there should exist one with respect to which the first-order partial derivative of F is not zero. Without any loss of generality, we may assume that $F_{x_1} \neq 0$ at any point of B. Hence, the Jacobian of the map $(F, x_2, \ldots, x_n) : B \to \mathbb{R}^n$ is always invertible. To prove that the map provides a coordinate system, we just need to show that it is injective. Keeping $x_2, \ldots, x_n$ constant and varying x_1, we obtain a univariate function of x_1 on the interval I_1 whose derivative is always nonzero and is hence injective.

Next, to prove that the level sets of $F : B \to \mathbb{R}$ are connected, notice that F admits a representation

$$F(x_1, \ldots, x_n) = \sigma\left(w_1 G_1 + \cdots + w_l G_l + w'_{m_1+\cdots+m_l+1} x_{m_1+\cdots+m_l+1} + \cdots + w'_n x_n\right) \tag{A.1}$$

where $G_s = G_s(x_{m_1+\cdots+m_{s-1}+1}, \ldots, x_{m_1+\cdots+m_{s-1}+m_s})$ is the tree function that $\mathcal{T}_s$ computes by receiving $x_{m_1+\cdots+m_{s-1}+1}, \ldots, x_{m_1+\cdots+m_{s-1}+m_s}$ from its leaves; σ is the activation function assigned to the root of $\mathcal{T}$; and $w_1, \ldots, w_l$, $w'_{m_1+\cdots+m_l+1}, \ldots, w'_n$ are the weights appearing at the root. A simple application of the chain rule implies that G_1, which is a function implemented on the tree $\mathcal{T}_1$, satisfies the nonvanishing hypotheses of theorem 7 on the

box-like region $I_1 \times \cdots \times I_{m_1}$; moreover, the derivative of σ is nonzero at any point of its domain[9] because otherwise there exists a point of $\mathbf{p} \in B$ at which $F_{x_i}(\mathbf{p}) = 0$ for any leaf x_i. By the same logic, the weight w_1 must be nonzero because otherwise all first-order partial derivatives with respect to the variables appearing in $\mathcal{T}_1$ are identically zero. We now show that an arbitrary level set $L_c := \{\mathbf{x} \in B \mid F(\mathbf{x}) = c\}$ is connected. Given the representation (see equation A.1) of F, the level set is empty if σ does not attain the value c. Otherwise, σ attains c at a unique point $\sigma^{-1}(c)$ of its domain. So one may rewrite the equation $F(x_1, \ldots, x_n) = c$ as

$$G_1 = -\frac{w_2}{w_1} G_2 - \cdots - \frac{w_l}{w_1} G_l - \frac{w'_{m_1+\cdots+m_l+1}}{w_1} x_{m_1+\cdots+m_l+1} - \cdots - \frac{w'_n}{w_1} x_n + \frac{1}{w_1} \sigma^{-1}(c). \tag{A.2}$$

The left-hand side of equation A.2 is a function of $x_1, \ldots, x_{m_1}$, while its right-hand side, which we denote by $\tilde{G}$, is a function of $x_{m_1+1}, \ldots, x_n$. Therefore, the level set L_c is the preimage of

$$\left\{(y, \tilde{\mathbf{x}}) \in \mathbb{R} \times (I_{m_1+1} \times \cdots \times I_n) \mid y = \tilde{G}(\tilde{\mathbf{x}})\right\} \tag{A.3}$$

under the map

$$\begin{cases} \pi : B = (I_1 \times \cdots \times I_{m_s}) \times (I_{m_1+1} \times \cdots \times I_n) \to \mathbb{R} \times (I_{m_1+1} \times \cdots \times I_n) \\ (x_1, \ldots, x_{m_1}; \tilde{\mathbf{x}}) \mapsto (G_1(x_1, \ldots, x_{m_1}), \tilde{\mathbf{x}}) . \end{cases} \tag{A.4}$$

The following simple fact can now be invoked: *Let $\pi : X \to Y$ be a continuous map of topological spaces that takes open sets to open sets and has connected level sets. Then the preimage of any connected subset of Y under π is connected.* Here, L_c is the preimage of the set from equation A.3, which is connected since it is the graph of a continuous function, under the map π defined in equation A.4, which is open because the scalar-valued function G_1 is: its gradient never vanishes. Therefore, the connectedness of the level sets of F is implied by the connectedness of the level sets of π. A level set of the map, equation A.4, could be identified with a level set of its first component G_1. Consequently, we have reduced to the similar problem for the function G_1, which is implemented on the smaller tree $\mathcal{T}_1$. Therefore, an inductive argument yields the connectedness of the level sets of F. It only remains to check the basic case of a tree whose leaves are directly connected to the root. In

[9] As the vector $(x_1, \ldots, x_n)$ of inputs varies in the box-like region B, the inputs to each node form an interval on which the corresponding activation function is defined.

that setting, $F(x_1, \ldots, x_n)$ is in the form of $\sigma(a_1x_1 + \cdots + a_nx_n)$ (the family of functions that lemma 3 is concerned with). By repeating the argument used before, the activation function σ is injective. Hence, a level set $F(\mathbf{x}) = c$ is the intersection of the hyperplane $a_1x_1 + \cdots + a_nx_n = \sigma^{-1}(c)$ with the box-like region B. Such an intersection is convex and thus connected. □

Proof of Lemma 3. The necessity of conditions $F_{x_ix_k}F_{x_j} = F_{x_jx_k}F_{x_i}$ follows from a simple computation. For the other direction, suppose $F_{x_j} \neq 0$ throughout an open box-like region $B \subseteq \mathbb{R}^n$ and any ratio $\frac{F_{x_i}}{F_{x_j}}$ is constant on B. Denoting it by a_i, we obtain numbers $a_1, \ldots, a_n$ with $a_j = 1$. They form a vector $[a_1 \ \cdots \ a_n]^{\mathrm{T}}$ parallel to ∇F. Thus, F could have nonzero first-order partial derivative only with respect to the first member of the coordinate system,

$$(a_1x_1 + \cdots + a_nx_n, x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n),$$

for B. The coordinate hypersurfaces are connected since they are intersections of hyperplanes in $\mathbb{R}^n$ with the convex region B. This fact enables us to deduce that F can be written as a function of $a_1x_1 + \cdots + a_nx_n$ globally. □

Proof of Lemma 4. For a function $q = q_1\,q_2$ such as equation 3.4, equalities of the form $q\,q_{y_a^{(1)}y_b^{(2)}} = q_{y_a^{(1)}}\,q_{y_b^{(2)}}$ hold since both sides coincide with $q_1\,q_2\,(q_1)_{y_a^{(1)}}(q_2)_{y_b^{(2)}}$. For the other direction, let $q = q\left(y_1^{(1)}, \ldots, y_{n_1}^{(1)}; y_1^{(2)}, \ldots, y_{n_2}^{(2)}\right)$ be a smooth function on an open box-like region $B_1 \times B_2 \subseteq \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ that satisfies $q\,q_{y_a^{(1)}y_b^{(2)}} = q_{y_a^{(1)}}\,q_{y_b^{(2)}}$ for any $1 \leq a \leq n_1$ and $1 \leq b \leq n_2$, and never vanishes. So q is either always positive or always negative. One may assume the former by replacing q with $-q$ if necessary. Hence, we can define a new function $p := \mathrm{Ln}(q)$ by taking the logarithm. We have

$$p_{y_a^{(1)}y_b^{(2)}} = \left(\frac{q_{y_a^{(1)}}}{q}\right)_{y_b^{(2)}} = \frac{q\,q_{y_a^{(1)}y_b^{(2)}} - q_{y_a^{(1)}}\,q_{y_b^{(2)}}}{q^2} = 0.$$

It suffices to show that this vanishing of mixed partial derivatives allows us to write $p\left(y_1^{(1)}, \ldots, y_{n_1}^{(1)}; y_1^{(2)}, \ldots, y_{n_2}^{(2)}\right)$ as $p_1\left(y_1^{(1)}, \ldots, y_{n_1}^{(1)}\right) + p_2\left(y_1^{(2)}, \ldots, y_{n_2}^{(2)}\right)$ since then exponentiating yields q_1 and q_2 as e^{p_1} and e^{p_2}, respectively. The domain of p is a box-like region of the form

$$B_1 \times B_2 = \left(\prod_{a=1}^{n_1} I_a^{(1)}\right) \times \left(\prod_{b=1}^{n_2} I_b^{(2)}\right).$$

Picking an arbitrary point $z_1^{(1)} \in I_1^{(1)}$, the fundamental theorem of calculus implies

$$\begin{aligned} & p\left(y_1^{(1)}, \ldots, y_{n_1}^{(1)}; y_1^{(2)}, \ldots, y_{n_2}^{(2)}\right) \\ &= \int_{z_1^{(1)}}^{y_1^{(1)}} p_{y_1^{(1)}}\left(s_1^{(1)}, y_2^{(1)}, \ldots, y_{n_1}^{(1)}; y_1^{(2)}, \ldots, y_{n_2}^{(2)}\right) \mathrm{d}s_1^{(1)} \\ &\quad + p\left(z_1^{(1)}, y_2^{(1)}, \ldots, y_{n_1}^{(1)}; y_1^{(2)}, \ldots, y_{n_2}^{(2)}\right). \end{aligned}$$

On the right-hand side, the integral is dependent only on $y_1^{(1)}, \ldots, y_{n_1}^{(1)}$ because the partial derivatives of the integrand with respect to $y_1^{(2)}, \ldots, y_{n_2}^{(2)}$ are all identically zero. The second term, $p\left(z_1^{(1)}, y_2^{(1)}, \ldots, y_{n_1}^{(1)}; y_1^{(2)}, \ldots, y_{n_2}^{(2)}\right)$, is a function on the smaller box-like region

$$\left(\prod_{a=2}^{n_1} I_a^{(1)}\right) \times \left(\prod_{b=1}^{n_2} I_b^{(2)}\right)$$

in $\mathbb{R}^{n_1-1} \times \mathbb{R}^{n_2}$ and thus, proceeding inductively, can be brought into the appropriate summation form. □

Appendix B: Differential Forms

Differential forms are ubiquitous objects in differential geometry and tensor calculus. We only need the theory of differential forms on open domains in Euclidean spaces. Theorem 5 (which has been used several times throughout the, for example, in the proof of theorem 3) is formulated in terms of differential forms. This appendix provides the necessary background for understanding the theorem and its proof.

We begin with a very brief account of the local theory of differential forms. (For a detailed treatment see Pugh, 2002, chap. 5.) Let U be an open subset of $\mathbb{R}^n$. A *differential k-form* ω on U assigns a scalar to any k-tuple of tangent vectors at a point $\mathbf{p}$ of U. This assignment, denoted by $\omega_{\mathbf{p}}$, must be multilinear and alternating. We say ω is smooth (resp. analytic) if $\omega_{\mathbf{p}}$ varies smoothly (resp. analytically) with $\mathbf{p}$. In other words, feeding ω with k smooth (resp. analytic) vector fields $\mathbf{V}_1, \ldots, \mathbf{V}_k$ on U results in a function $\omega(\mathbf{V}_1, \ldots, \mathbf{V}_k) : U \to \mathbb{R}$ that is smooth (resp. analytic). We next exhibit an expression for ω. Consider the standard basis $\left(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}\right)$ of vector fields on U where $\frac{\partial}{\partial x_i}$ assigns $\mathbf{e}_i$ to each point. The dual basis is denoted

by $(\mathrm{d}x_1, \ldots, \mathrm{d}x_n)$, which at each point yields the dual of the standard basis $(\mathbf{e}_1, \ldots, \mathbf{e}_n)$ for $\mathbb{R}^n$, that is, $\mathrm{d}x_i\left(\frac{\partial}{\partial x_j}\right) \equiv \delta_{ij}$. Each of $\mathrm{d}x_1, \ldots, \mathrm{d}x_n$ is a 1-form on U, and any k-form ω can be written in terms of them:

$$\omega = \sum_{1 \le i_1 < \ldots < i_k \le n} f_{i_1 \ldots i_k}\, \mathrm{d}x_{i_1} \wedge \ldots \wedge \mathrm{d}x_{i_k}. \tag{B.1}$$

Here, each coefficient $f_{i_1 \ldots i_k}$ is a (smooth or analytic according to the context) function $U \to \mathbb{R}$. In front of it, $\mathrm{d}x_{i_1} \wedge \ldots \wedge \mathrm{d}x_{i_k}$ appears, which is a k-form satisfying $\mathrm{d}x_{i_1} \wedge \ldots \wedge \mathrm{d}x_{i_k}\left(\frac{\partial}{\partial x_{i_1}}, \ldots, \frac{\partial}{\partial x_{i_k}}\right) = 1$. This is constructed by the operation of the exterior product (also called the wedge product) from multilinear algebra. The exterior product is an associative and distributive linear operation that out of k_i-tensors τ_i $(1 \le i \le l)$ constructs an alternating $(k_1 + \cdots + k_l)$-tensor $\tau_1 \wedge \ldots \wedge \tau_l$. This product is anti-commutative, for example, $\mathrm{d}x_i \wedge \mathrm{d}x_j = -\mathrm{d}x_j \wedge \mathrm{d}x_i$; this is the reason that in equation B.1, the indices are taken to be strictly ascending.

Another operation in the realm of differential forms is exterior differentiation. For the k-form ω from equation B.1, its exterior derivative $\mathrm{d}\omega$ is a $(k+1)$-form defined as

$$\mathrm{d}\omega := \sum_{1 \le i_1 < \ldots < i_k \le n} \mathrm{d}f_{i_1 \ldots i_k} \wedge \mathrm{d}x_{i_1} \wedge \ldots \wedge \mathrm{d}x_{i_k},$$

where the exterior derivative of a function f is defined as

$$\mathrm{d}f := \sum_{i=1}^{n} f_{x_i}\, \mathrm{d}x_i. \tag{B.2}$$

Notice that the 1-form is the dual of the gradient vector field

$$\nabla f = \sum_{i=1}^{n} f_{x_i}\, \frac{\partial}{\partial x_i}. \tag{B.3}$$

B.1 Example 14. In dimension 3, the exterior differentiation encapsulates the familiar vector calculus operators curl and divergence. Consider the vector field

$$\mathbf{V}(x, y, z) = [V_1(x, y, z), V_2(x, y, z), V_3(x, y, z)]^{\mathrm{T}}.$$

The exterior derivatives

$$\mathrm{d}\big(V_1\,\mathrm{d}x + V_2\,\mathrm{d}y + V_3\,\mathrm{d}z\big) = \big((V_3)_y - (V_2)_z\big)\mathrm{d}y \wedge \mathrm{d}z + \big((V_1)_z - (V_3)_x\big)\mathrm{d}z \wedge \mathrm{d}x + \big((V_2)_x - (V_1)_y\big)\mathrm{d}x \wedge \mathrm{d}y$$

and

$$\mathrm{d}\big(V_1\,\mathrm{d}y \wedge \mathrm{d}z + V_2\,\mathrm{d}z \wedge \mathrm{d}x + V_3\,\mathrm{d}x \wedge \mathrm{d}y\big) = \big((V_1)_x + (V_2)_y + (V_3)_z\big)\mathrm{d}x \wedge \mathrm{d}y \wedge \mathrm{d}z,$$

respectively, have curl$\mathbf{V}$ and div$\mathbf{V}$ as their coefficients. In fact, there is a general Stokes formula for differential forms that recovers the Kelvin–Stokes theorem and the divergence theorem as special cases. Finally, we point out that the familiar identities curl $\circ\, \nabla = \mathbf{0}$ and div $\circ$ curl $= 0$ are instances of the general property $\mathrm{d} \circ \mathrm{d} = 0$ of the exterior differentiation.

B.2 Example 15. As mentioned in the previous example, the outcome of twice applying the exterior differentiation operator to a form is always zero. This is an extremely important property that leads to the definitions of closed and exact differential forms. A k-form ω on an open subset U of $\mathbb{R}^n$ is called closed if $\mathrm{d}\omega = 0$. This holds if ω is in the form of $\omega = \mathrm{d}\alpha$ for a $(k-1)$-form α on U. Such forms are called exact. The space of closed forms may be strictly larger than the space of exact forms; the difference of these spaces can be used to measure the topological complexity of U. If U is an open box-like region, every closed form on it is exact. But, for instance, the 1-form $\omega = -\frac{y}{x^2+y^2}\,\mathrm{d}x + \frac{x}{x^2+y^2}\,\mathrm{d}y$ on $\mathbb{R}^2 - \{(0,0)\}$ is closed while it may not be written as $\mathrm{d}\alpha$ for any smooth function $\alpha : \mathbb{R}^2 - \{(0,0)\} \to \mathbb{R}$. This brings us to a famous fact from multivariable calculus that we have used several times (e.g., in the proof of theorem 4). A necessary condition for a vector field $\mathbf{V} = \sum_{i=1}^{n} V_i \frac{\partial}{\partial x_i}$ on an open subset U of $\mathbb{R}^n$ to be a gradient vector field is $(V_i)_{x_j} = (V_j)_{x_i}$ for any $1 \le i, j \le n$. Near each point of U, the vector field $\mathbf{V}$ may be written as ∇f; it is globally in the form of ∇f for a function $f : U \to \mathbb{R}$ when U is simply connected. In view of equations B.2 and B.3, one may rephrase this fact as: *Closed 1-forms on U are exact if and only if U is simply connected.*

Proof of Theorem 5. Near a point $\mathbf{p} \in \mathbb{R}^n$ at which $\mathbf{V}(\mathbf{p}) \neq \mathbf{0}$, we seek a locally defined function ξ with $\mathbf{V} \parallel \nabla\xi \neq \mathbf{0}$. Recall that if $\mathbf{q} \in \mathbb{R}^n$ is a regular point of ξ, then near $\mathbf{q}$, the level set of ξ passing through $\mathbf{q}$ is an $(n-1)$-dimensional submanifold of $\mathbb{R}^n$ to which the gradient vector field, $\nabla\xi \neq \mathbf{0}$, is perpendicular. As we want the gradient to be parallel to the vector field $\mathbf{V}$, the equivalent characterization in terms of the 1-form ω, which is the dual of $\mathbf{V}$ (cf. equations 3.1 and 3.3), asserts that ω is zero at any vector tangent to the level set. So the tangent space to the level set at the point $\mathbf{q}$

could be described as $\{\mathbf{v} \in \mathbb{R}^n \mid \omega_{\mathbf{q}}(\mathbf{v}) = 0\}$. As $\mathbf{q}$ varies near $\mathbf{p}$, these $(n-1)$-dimensional subspaces of $\mathbb{R}^n$ vary smoothly. In differential geometry, such a higher-dimensional version of a vector field is called a *distribution*, and the property that these subspaces are locally given by tangent spaces to a family of submanifolds (the level sets here) is called *integrability*. The seminal Frobenius theorem (Narasimhan, 1968, theorem 2.11.11) implies that the distribution defined by a nowhere vanishing 1-from ω is integrable if and only if $\omega \wedge d\omega = 0$. □

References

Arnold, V. I. (2009a). On the representation of functions of several variables as a superposition of functions of a smaller number of variables. In V. A. Arnold, *Collected works: Representations of functions, celestial mechanics and KAM theory, 1957–1965* (pp. 25–46). Berlin: Springer.

Arnold, V. I. (2009b). Representation of continuous functions of three variables by the superposition of continuous functions of two variables. In V. A. Arnold, *Collected works: Representations of functions, celestial mechanics and KAM theory, 1957–1965* (pp. 47–133). Berlin: Springer.

Bartlett, P. L., Maiorov, V., & Meir, R. (1999). Almost linear VC dimension bounds for piecewise polynomial networks. In S. Solla, T. Leen, & K. R. Müller (Eds.), *Advances in neural information processing systems, 11* (pp. 190–196). Cambridge, MA: MIT Press.

Bianchini, M., & Scarselli, F. (2014). On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(8), 1553–1565. 10.1109/TNNLS.2013.2293637, PubMed: 25050951

Boyce, W. E., & DiPrima, R. C. (2012). *Elementary differential equations* (10th ed.). Hoboken, NJ: Wiley.

Brattka, V. (2007). From Hilbert's 13th problem to the theory of neural networks: Constructive aspects of Kolmogorov's superposition theorem. In E. Charpentier, A. Lesne, & N. Annick (Eds.), *Kolmogorov's heritage in mathematics* (pp. 253–280). Berlin: Springer.

Buck, R. C. (1976). *Approximate complexity and functional representation* (Tech. Rep.). Madison: Mathematics Research Center, University of Wisconsin, Madison. https://apps.dtic.mil/dtic/tr/fulltext/u2/a031972.pdf

Buck, R. C. (1979). Approximate complexity and functional representation. *J. Math. Anal. Appl.*, *70*(1), 280–298. 10.1016/0022-247X(79)90091-X

Buck, R. C. (1981a). Characterization of classes of functions. *Amer. Math. Monthly*, *88*(2), 139–142. 10.1080/00029890.1981.11995204

Buck, R. C. (1981b). The solutions to a smooth PDE can be dense in $C(\mathrm{I})$. *J. Differential Equations*, *41*(2), 239–244. 10.1016/0022-0396(81)90060-7

Chang, H.-C., He, W., & Prabhu, N. (2003). The analytic domain in the implicit function theorem. *J. Inequal. Pure Appl. Math.*, *4*(1), art. 12, 5.

Chatziafratis, V., Nagarajan, S. G., Panageas, I., & Wang, X. (2019). *Depth-width tradeoffs for ReLU networks via Sharkovsky's theorem*. arXiv:1912.04378v1.

Cohen, N., Sharir, O., & Shashua, A. (2016). On the expressive power of deep learning: A tensor analysis. In *Proceedings of the Conference on Learning Theory* (pp. 698–728).

Coste, M. (2000). *An introduction to semialgebraic geometry*. Citeseer.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*(4), 303–314. 10.1007/BF02551274

Dehmamy, N., Rohani, N., & Katsaggelos, A. K. (2019). Direct estimation of weights and efficient training of deep neural networks without SGD. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 3232–3236). Piscataway, NJ: IEEE.

Du, S. S., & Lee, J. D. (2018). *On the power of over-parametrization in neural networks with quadratic activation*. arXiv:1803.01206v2.

Eldan, R., & Shamir, O. (2016). The power of depth for feedforward neural networks. In *Proceedings of the Conference on Learning Theory* (pp. 907–940).

Farhoodi, R., Filom, K., Jones, I. S., & Körding, K. P. (2019). On functions computed on trees. *Neural Computation*, *31*, 2075–2137. 10.1162/neco_a_01231, PubMed: 31525312

Fefferman, C., & Markel, S. (1994). Recovering a feed-forward net from its output. In G. Tesauro, D. Toretsky, & T. Leen (Eds.), *Advances in neural information processing systems*, *7* (pp. 335–342). Cambridge, MA: MIT Press.

Gerhard, S., Andrade, I., Fetter, R. D., Cardona, A., & Schneider-Mizell, C. M. (2017). Conserved neural circuit structure across drosophila larval development revealed by comparative connectomics. *eLife*, *6*, e29089. 10.7554/eLife.29089, PubMed: 29058674

Gillette, T. A., & Ascoli, G. A. (2015). Topological characterization of neuronal arbor morphology via sequence representation: I-motif analysis. *BMC Bioinformatics*, *16*(1), 216.

Girosi, F., & Poggio, T. (1989). Representation properties of networks: Kolmogorov's theorem is irrelevant. *Neural Computation*, *1*(4), 465–469. 10.1162/neco.1989.1.4.465

Hecht-Nielsen, R. (1987). Kolmogorov's mapping neural network existence theorem. In *Proceedings of the International Conference on Neural Networks* (vol. *3*, pp. 11–14). Piscataway, NJ: IEEE.

Hilbert, D. (1902). Mathematical problems. *Bulletin of the American Mathematical Society*, *8*(10), 437–479. 10.1090/S0002-9904-1902-00923-3

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257. 10.1016/0893-6080(91)90009-T

Hornik, K., Stinchcombe, M., & White H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. 10.1016/0893-6080(89)90020-8

Kileel, J., Trager, M., & Bruna, J. (2019). *On the expressive power of deep polynomial neural networks*. arXiv:1905.12207v1.

Kollins, K. M., & Davenport, R. W. (2005). Branching morphogenesis in vertebrate neurons. In K. Kollins & R. Davenport (Eds.), *Branching morphogenesis* (pp. 8–65). Berlin: Springer.

Kolmogorov, A. N. (1957). On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Dokl. Akad. Nauk SSSR*, *114*, 953–956.

Krantz, S. G., & Parks, H. R. (2002). *The implicit function theorem*. Boston: Birkhäuser.
Kůrková, V. (1991). Kolmogorov's theorem is relevant. *Neural Computation*, *3*(4), 617–622.
Kůrková, V. (1992). Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, *5*(3), 501–506.
Lin, H. W., Tegmark, M., & Rolnick, D. (2017). Why does deep and cheap learning work so well? *Journal of Statistical Physics*, *168*(6), 1223–1247. 10.1007/s10955-017-1836-5
Lorentz, G. G. (1966). *Approximation of functions*. New York: Holt.
Mhaskar, H. (1996). Neural networks for optimal approximation of smooth and analytic functions. *Neural Computation*, *8*(1), 164–177. 10.1162/neco.1996.8.1.164
Mhaskar, H., Liao, Q., & Poggio, T. (2017). When and why are deep networks better than shallow ones? In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence.* Palo Alto: AAAI.
Minsky, M., & Papert, S. A. (2017). *Perceptrons: An introduction to computational geometry*. Cambridge, MA: MIT Press.
Montufar, G. F., Pascanu, R., Cho, K., & Bengio, Y. (2014). On the number of linear regions of deep neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems, 27* (pp. 2924–2932). Red Hook, NY: Curran.
Narasimhan, R. (1968). *Analysis on real and complex manifolds*. Amsterdam: North-Holland.
Ostrowski, A. (1920). Über dirichletsche reihen und algebraische differentialgleichungen. *Mathematische Zeitschrift*, *8*(3), 241–298. 10.1007/BF01206530
Petersen, P., Raslan, M., & Voigtlaender, F. (2020). Topological properties of the set of functions generated by neural networks of fixed size. *Foundations of Computational Mathematics*, *21*, 375–444. 10.1007/s10208-020-09461-0
Poggio, T., Banburski, A., & Liao, Q. (2019). *Theoretical issues in deep networks: Approximation, optimization and generalization.* arXiv:1908.09375v1.
Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., & Liao, Q. (2017). Why and when can deep—but not shallow—networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, *14*(5), 503–519. 10.1007/s11633-017-1054-2
Pólya, G., & Szegö, G. (1945). *Aufgaben und Lehrsätze aus der Analysis*. New York: Dover.
Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., & Ganguli, S. (2016). Exponential expressivity in deep neural networks through transient chaos. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems*, *29* (pp. 3360–3368). Red Hook, NY: Curran.
Pugh, C. C. (2002). *Real mathematical analysis*. Springer-Verlag, New York.
Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., & Dickstein, J. S. (2017). On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, *70* (pp. 2847–2854).
Rolnick, D., & Kording, K. P. (2019, October). *Reverse-engineering deep ReLU networks.* arXiv:1910.00744v2.
Rubel, L. A. (1981). A universal differential equation. *Bull. Amer. Math. Soc. (N.S.)*, *4*(3), 345–349. 10.1090/S0273-0979-1981-14910-7

Schneider-Mizell, C. M., Gerhard, S., Longair, M., Kazimiers, T., Li, F., Zwart, M. F., . . . Cardona, A. (2016). Quantitative neuroanatomy for connectomics in drosophila. *eLife*, *5*, e12059. 10.7554/eLife.12059

Soltanolkotabi, M., Javanmard, A., & Lee, J. D. (2018). Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, *65*(2), 742–769. 10.1109/TIT.2018.2854560

Sprecher, D. A. (1965). On the structure of continuous functions of several variables. *Trans. Amer. Math. Soc.*, *115*, 340–355. 10.2307/1994273

Telgarsky, M. (2016). *Benefits of depth in neural networks.* arXiv:1602.04485v2.

Venturi, L., Bandeira, A. S., & Bruna, J. (2018). *Spurious valleys in two-layer neural network optimization landscapes.* arXiv:1802.06384v3.

Vituškin, A. G. (1954). On Hilbert's thirteenth problem. *Doklady Akad. Nauk SSSR (N.S.)*, *95*, 701–704.

Vituškin, A. G. (1964). A proof of the existence of analytic functions of several variables not representable by linear superpositions of continuously differentiable functions of fewer variables. *Dokl. Akad. Nauk SSSR*, *156*, 1258–1261.

Vituškin, A. G. (2004). On Hilbert's thirteenth problem and related questions. *Russian Mathematical Surveys*, *59*(1), 11.

Vituškin, A. G., & Henkin, G. M. (1967). Linear superpositions of functions. *Uspehi Mat. Nauk*, *22*(133), 77–124.

von Golitschek, M. (1980). Remarks on functional representation. In *Approximation theory, III (Proc. Conf., Univ. Texas, Austin, Tex., 1980)* (pp. 429–434). New York: Academic Press.

Received September 18, 2020; accepted June 11, 2021.