Compositional features and neural network complexity in deep learning ⋆

Wei Kang * **Qi Gong** **

* *Naval Postgraduate School, Monterey, CA and University of California, Santa Cruz, CA, USA (e-mail: wkang@nps.edu).*
** *University of California, Santa Cruz, CA, USA (e-mail: qgong@ucsc.edu)*

Abstract: In this study, we explore the relationship between the complexity of neural networks and the internal compositional structure of the function to be approximated. The results shed light on the reason why using neural network approximation helps to avoid the curse of dimensionality.

Keywords: Non-linear control systems, deep learning, optimal control, power systems

1. INTRODUCTION

In this study, we explore the relationship between the complexity of neural networks and the internal compositional structure of the function to be approximated. The results shed light on the reason why using neural network approximation helps to avoid the curse of dimensionality (COD). In Section 2, we discuss the challenge of COD in feedback control. In Section 3, we introduce four compositional features that determine the complexity and error upper bound of neural network approximation for dynamical and control systems. In Section 4, several examples are given to illustrate the widely observed phenomenon in science and engineering that complicated functions are formed by the composition of simple ones.

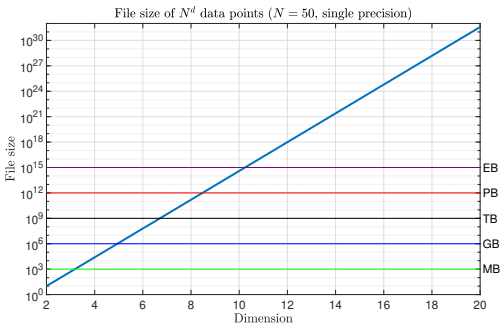

Fig. 1. The size of data file that grows exponentially with the space dimension. In each dimension, $N = 50$. The total number of data points is N^d.

2. THE CURSE OF DIMENSIONALITY

The COD is a bottleneck in many applications of dynamical systems and nonlinear control. It is a phenomenon in which the complexity of an approximate solution grows fast, such as exponentially, with the state space dimension. For instance, consider a feedback control law

$$u = u(\mathbf{x}),\ \mathbf{x} \in \mathbb{R}^d,\ u \in \mathbb{R}. \tag{1}$$

for a system of dimension d. If d is large and if an analytic representation cannot be found for $u(\mathbf{x})$, a numerical approximation has to be applied to store the control law in a digital format. If $u(\mathbf{x})$ is approximated using interpolation based on its value at grid points, the size of the dataset increases exponentially. Specifically, suppose we use N grid points in each dimension. Then the total number of grid points in a d dimensional domain is N^d. The value of $u(\mathbf{x})$ over the grid forms a huge dataset even for moderate dimensions such as $d = 7$ or 8. Figure 1 shows an example in which $N = 50$. If $d = 7$, the memory needed to store the value of $u(\mathbf{x})$ over the grid using single precision is about 1TB. This number is 1PB for $d = 9$ and 1EB for $d = 10$. Due to the limitations on processor's primary storage, bus speed and computation speed, the interpolation of datasets that have such large sizes is practically intractable, not to mention that the computation has to be carried out in real-time for feedback control.

3. COMPOSITIONAL FEATURES AND THE COMPLEXITY OF NEURAL NETWORKS

For the last few years, a new trend of overcoming the COD in nonlinear dynamics and control using deep learning has been developed rapidly. Many examples of successfully applying deep learning to high dimensional differential equations and optimal control were published in which the dimensions range from six to several hundred, well beyond what conventional computational methods can deal with (Han et al. (2018); Izzo et al. (2019); Nakamura-Zimmerer et al. (2021); Kang et al. (2021a); Sirignano and Spiliopoulos (2018); Raissi et al. (2019); B. Azmi (2020)). These empirical successes of deep learning in overcoming the COD inspire us to study the underlying reason why neural networks are capable of solving so many high dimensional problems. The philosophy in our study is based on a widely

⋆ This work was supported in part by U.S. Naval Research Laboratory - Monterey, CA and National Science Foundation

observed fact in science and engineering: complicated functions are formed by the composition of relatively simple functions. In Kang and Gong (2022), a set of key features of compositional functions is defined. It can be mathematically proved that these features determine the upper bounds of neural network complexity and approximation error. These upper bounds do not suffer from the COD.

To exemplify the compositional structure of nonlinear systems, consider the swing equations of a power system with N_g generators

$$\begin{aligned}\frac{d\omega_i}{dt} &= \frac{\omega_0}{2H_i}\left(P_m - D\frac{\omega_i-\omega_0}{\omega_0} - E_i^2 G_{ii} \cdots \right.\\ &\left. - \sum_{j=1,j\neq i}^{10} E_i E_j [B_{ij}\sin(\delta_i-\delta_j) + G_{ij}\cos(\delta_i-\delta_j)]\right)\\ \frac{d\delta_i}{dt} &= \omega_i - \omega_0,\end{aligned} \tag{2}$$

where $i = 1, ..., N_g$. For the ith generator, the two state variables are δ_i, the rotor angle in radian, and ω_i, the rotor speed in radian per second. Other parameters include H_i (the inertial constant of the generator), $\omega_0 = 2\pi \times f_0$ (the synchronous angular frequency in radian per second for an ac power system with frequency f_0), D (the damping coefficient), P_m (the mechanical power input from the turbine), E_i (the electromotive force or internal voltage of the generator). In addition, $G_{ij} + jB_{ij}$, the mutual admittance between E_i and E_j, is the i^{th} row j^{th} column element of the admittance matrix among all electromotive forces, and G_{ii} is the conductance representing the local load seen from E_i. Details about the model and its parameters refer to Athay et al. (1979).

A power system may have tens or hundreds of generators. This complicated system model, however, is a composition of functions that have low input dimensions. The compositional structure can be represented using a layered directed acyclic graph (DAG). For example, Figure 2 is a DAG of the function in (2). Each colored node in the DAG represents a nonlinear function. They are all sine and cosine functions with a single input. Although some linear nodes (white color) have high input dimensions, such as the node in the output layer, it is proved in Kang and Gong (2022) that linear nodes do not increase the complexity, or the number of neurons, of the neural network. The

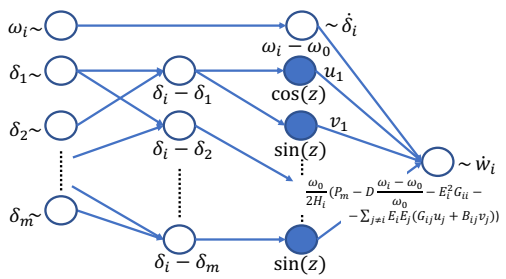

Fig. 2. The layered DAG of the function in (2) as a compositional function.

question we would like to answer is: how does a layered DAG help in the effort of using deep learning to find the trajectory or the optimal control of a system? Our study shows that some compositional features are critical to the required complexity of neural networks used in deep learning. These features are briefly introduced as follows.

- $|\mathcal{V}|$ (complexity feature): The total number of nonlinear nodes in the layered DAG, where $\mathcal{V}$ is the set of nonlinear nodes of the compositional function.
- r_{max} (dimension feature): The largest ratio, d/m, for all nodes in $\mathcal{V}$, where d is the input dimension of the node and m is the smoothness of the node.
- Λ (volume feature): Each nonlinear node, denote the function by f, has a domain. Assume that the domain is a square of edge length R. The volume feature is defined to be the largest value in
$$\{\max\{R,1\}||f||;\ f \in \mathcal{V}\}, \tag{3}$$
where $||\cdot||$ is the Sobolev norm
$$||f|| = ||f||_{L^\infty} + \sum \left\| \frac{\partial f}{\partial x_i} \right\|_{L^\infty}. \tag{4}$$
- $L_{\max}$ (Lipschitz constant feature): The largest Lipschitz constants associated with nonlinear nodes. Note that this Lipschitz constant is defined based on the layered structure of the DAG. For more details, the readers are referred to Kang and Gong (2022).

Consider a general dynamical system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}),\ \mathbf{x} \in \mathbb{R}^d \tag{5}$$

in which $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^d$ has a layered compositional structure. Let $\boldsymbol{\phi}(t, \mathbf{x})$ represents the solution of (5) in which $\mathbf{x}$ is the initiate state, $\boldsymbol{\phi}(0, \mathbf{x}) = \mathbf{x}$.

Theorem 1. (Kang and Gong (2022)) Suppose that all nodes in $\mathbf{f}$ are C^1. Let $D \subset \mathbb{R}^d$ be a closed set and $R > 0$ be a constant. Suppose $\boldsymbol{\phi}(t, \mathbf{x}) \in [-R, R]^d$ for $t \in [0, T]$ and $\mathbf{x} \in D$. Then, there always exists a deep feedforward neural network, denoted by $\boldsymbol{\phi}^{NN}(\mathbf{x})$, in which activation functions are C^∞. Furthermore, the network satisfies

$$\left\|\boldsymbol{\phi}^{NN}(\mathbf{x}) - \boldsymbol{\phi}(T, \mathbf{x})\right\|_2 < (C_1 L_{max} \Lambda\, |\mathcal{V}| + C_2) n^{-1/r_{max}} \tag{6}$$

where n is an integer that determines an upper bound of the total number of neurons in $\boldsymbol{\phi}^{NN}(\mathbf{x})$, i.e., the complexity of the neural network,

$$\#\text{ of neurons in } \boldsymbol{\phi}^{NN} \leq \left(n^{1/r_{max}} + 1\right) n\, |\mathcal{V}| \tag{7}$$

The constants, C_1 and C_2, in (6) are determined by $\|\mathbf{f}\|_2$, $\left\|\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right\|_2$, T and the input dimensions of the nodes in $\mathcal{V}$.

It is worth to note that the error upper bound (6) depends on Λ, L_{max} and $|\mathcal{V}|$ as a polynomial function, rather than an exponential function. The value of r_{max} depends on the input dimensions of individual notes of $\mathbf{f}$, not directly on the overall dimension, d. Therefore, if r_{max} is bounded and if Λ, L_{max} and $|\mathcal{V}|$ do not increase exponentially with d, the neural network approximation of $\boldsymbol{\phi}(T, \mathbf{x})$ is free from the COD. A similar result holds true for optimal control. Consider a control system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}),\ \mathbf{x} \in D \subset \mathbb{R}^d,\ \mathbf{u} \in \mathbb{R}^q,\ t \in [0, T] \tag{8}$$

A zero-order hold control, $U = [\,\mathbf{u}_1\ \mathbf{u}_2\ \cdots\ \mathbf{u}_{N_t}\,]$ in which $\mathbf{u}_k$ is the constant control for $t \in [t_{k-1}, t_k]$. The goal of an optimal control problem is to find U that minimizes the cost function

$$J(\mathbf{x}, U)$$
$$= \Psi \circ \boldsymbol{\phi}(\Delta t; \mathbf{u}_{N_t}, \cdot) \cdot \boldsymbol{\phi}(\Delta t; \mathbf{u}_{N_t-1}, \cdot) \circ \cdots \circ \boldsymbol{\phi}(\Delta t; \mathbf{u}_1, \mathbf{x}) \tag{9}$$

where $\Psi : \mathbb{R}^d \to \mathbb{R}$ is a function.

Theorem 2. Suppose that $\mathbf{f}$ and Ψ are compositional functions in which all nodes are C^2. Let $D \subset \mathbb{R}^d$ be a bounded closed set. Assume that the Hession of $J(\mathbf{x}, U)$ with respect to U is positive definite. Let $U^*(\mathbf{x})$ represents the optimal feedback control. Then, for any $\epsilon > 0$, there exists a deep neural network, U^{*NN}, that approximates the optimal control. The estimation error is

$$\left\| U^{*NN}(\mathbf{x}) - U^*(\mathbf{x}) \right\|_2 \leq 3\epsilon, \ \mathbf{x} \in D \tag{10}$$

The complexity of U^{*NN} is bounded by

$$n \leq C\epsilon^{-(4r_{max}+1+4r_{max}/r^{\mathbf{f}}_{max})} \tag{11}$$

where $r^{\mathbf{f}}_{max}$ is the dimension feature of $\mathbf{f}$ and r_{max} is the largest dimension feature of $\mathbf{f}$ and Ψ, C is a polynomial of q and other compositional features of $\mathbf{f}$ and Ψ.

4. SOME EXAMPLES OF COMPOSITIONAL FEATURES

According to Theorems 1 and 2, if the compositional features of a family of systems do not increase exponentially with d, the approximation of a trajectory or an optimal control has a polynomial error upper bound; the complexity of the neural network increases at a polynomial rate. In the following, we use three examples to demonstrate that this kind of polynomial relationship is not unusual.

4.1 Compositional features of power systems

The first example is the power system model in (2). Its layered DAG is shown in Figure 2. Its compositional features are summarized as follows.

$$r_{max} = 1, \ \ \Lambda = 4\pi, \ \ |\mathcal{V}| = 2(N_g - 1)N_g,$$
$$L_{max} = \max_{1 \leq i,j \leq N_g, i \neq j} \left\{ \frac{\omega_0}{2H_i} E_i E_j G_{ij}, \frac{\omega_0}{2H_i} E_i E_j B_{ij} \right\}. \tag{12}$$

The dimension feature is $r_{max} = 1$ because all nonlinear nodes ($\sin(z)$ and $\cos(z)$) have a single input. Here we treat the nodes as functions in C^1 although they are also in C^∞. This simplifies the formula in the derivation. The value of Λ depends on the radius of the domain of nonlinear nodes and their Sobolev norm (4). For each nonlinear node, the domain of its input is bounded by 2π. Furthermore, $\sin(z)$, $\cos(z)$ and their derivatives are all bounded by 1. Then it is straightforward to derive the volume feature $\Lambda = 4\pi$. Following the definition in Kang and Gong (2022), the Lipschitz constant associated with a node is the Lipschitz constant of $\mathbf{f}$ (not the node) with respect to the node when the node is treated as a free variable. For example, the nonlinear nodes in Figure 2 are $\cos(z)$ and $\sin(z)$. If one of them, for instance the first $\cos(z)$ connecting to $\delta_i - \delta_1$, is treated as a variable, the Lipschitz constant of the function associated with this node equals

$$\frac{\omega_0}{2H_i} E_i E_1 G_{i1}. \tag{13}$$

This computation is for the first nonlinear node ($j = 1$) in the ith generator. The value of L_{max} in (19) is the largest one among the numbers computed similarly for all the N_g generators and all nonlinear nodes. For the ith generator, $1 \leq i \leq N_g$, there are $2(N_g - 1)$ nonlinear nodes. Therefore, the total number of nonlinear nodes is $|\mathcal{V}| = 2(N_g - 1)N_g$. We would like to emphasize that the compositional features in (12) are either constants or polynomial functions of N_g. As such, they do not increase exponentially with N_g. From Theorem 1, there exists a deep neural network approximation of the power system that avoids the COD. In fact, a similar conclusion can be extended to the Lyapunov function of the power system, which is proved in Kang et al. (2021b).

Theorem 3. Consider a power system (2) that has N_g generators. Let $\mathcal{R} \subset \mathbb{R}^{2N_g}$ be a bounded set. Then, there exists a solution, $V(\mathbf{x})$, to Zubov's equation (a special Lyapunov function that characterizes the domain of attraction) and a neural network, $V^{NN}(\mathbf{x})$, that has n^{NN} hyperbolic tangent neurons. They satisfy

$$\left| V^{NN}(\mathbf{x}) - V(\mathbf{x}) \right| < (C_1 N_g^2 + C_2) \frac{N_g}{\sqrt{n^{NN}}} \tag{14}$$

for $\mathbf{x} \in \mathcal{R}$, where C_1 and C_2 are constants independent of N_g.

4.2 Lorenz-96 model

Consider a system of ODEs

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{15}$$

in which $\mathbf{f} : [-R, R]^d \subset \mathbb{R}^d \to \mathbb{R}^d$ is the vector field that defines the Loranz-96 model in Lorenz (1996),

$$\mathbf{f} = \begin{bmatrix} x_0(x_2 - x_{-1}) - x_1 + F \\ x_1(x_3 - x_0) - x_2 + F \\ \vdots \\ x_{i-1}(x_{i+1} - x_{i-2}) - x_i + F \\ \vdots \\ x_{d-1}(x_{d+1} - x_{d-2}) - x_d + F \end{bmatrix}, \tag{16}$$

where $x_{-1} = x_{d-1}$, $x_0 = x_d$, $x_{d+1} = x_1$ and F is a constant. Let's treat $\mathbf{f}$ as a compositional function. An example of its layered DAG when $d = 4$ is shown in Figure 3. All nonlinear nodes in Figure 3 are located in the second

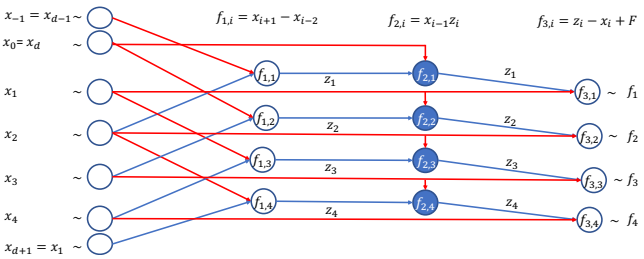

Fig. 3. DAG structure of the function (16) for $d = 4$. For a clear illustration, edges pointing to the first, the second, and the third layer are shown in red, blue and green respectively.

layer. They are defined by

$$f_{2,j}(x_{j-1}, z_j) = x_{j-1} z_j, \ \ j = 1, \cdots, d. \tag{17}$$

All nonlinear nodes have the dimension $d_{2,j} = 2$ and the domain $[-2R, 2R]^2$. Since $f_{2,j}$ is C^∞, we can set the smoothness to be any integer $m \geq 2$. As an example, we set $m = 1$. Then the dimension feature is $r_{max} = 2$. The Sobolev norm of $f_{2,j}$ is straightforward to compute, which

determines Λ, the volume feature. To compute the Lipchitz constant associated with the node, $f_{2,j}$, we construct the truncation of $\mathbf{f}$ along the second layer, which is given by

$$\bar{\mathbf{f}}(z_1, \cdots, z_{2d}) = \begin{bmatrix} z_1 - z_{d+1} + F \\ z_2 - z_{d+2} + F \\ \vdots \\ z_d - z_{2d} + F \end{bmatrix}, \tag{18}$$

where x_i, $1 \le i \le d$, are represented by the dummy inputs z_j, $j = d+1, \cdots, 2d$. The Lipschitz constant of $\bar{\mathbf{f}}$ with respect to z_j is $L_{2,j} = 1$, for $j = 1, \cdots, d$. The total number of nonlinear nodes in the system equals the dimension, d, because each equation in (16) has a single nonlinear node. To summarize, the compositional features of the Lorenz-96 model are

$$\begin{aligned} r_{max} &= 2,\ \Lambda = \max\{(2R), 1\}(2R + 4R^2), \\ L_{max} &= 1,\ |\mathcal{V}| = d. \end{aligned} \tag{19}$$

There is no exponential growth in the features. They are either constants or a linear function of d.

4.3 Burgers' Equation

Consider the following discretized Burgers's equation

$$\begin{aligned} \dot{u}_1 &= -u_1 \frac{u_2 - u_0}{2\Delta x} + \kappa \frac{u_2 + u_0 - 2u_1}{\Delta x^2} \\ \dot{u}_2 &= -u_2 \frac{u_3 - u_1}{2\Delta x} + \kappa \frac{u_3 + u_1 - 2u_2}{\Delta x^2} \\ &\vdots \\ \dot{u}_{N-1} &= -u_{N-1} \frac{u_N - u_{N-2}}{2\Delta x} + \kappa \frac{u_N + u_{N-2} - 2u_{N-1}}{\Delta x^2} \end{aligned} \tag{20}$$

The discretization is based on central different in which $\Delta x = L/N$ is the parameter that represents the step size of the state variable $x \in [0, L]$. The boundary condition is $u_0 = u_N = 0$. The dimension of the state space is N. The layered DAG of the function in (20) is shown in Figure 4. Due to the viscosity term in the equation, the solutions are stable. For initial conditions in a bounded set, we can assume that the state variables are bounded in $[-R, R]^N$ for some $R > 0$. The compositional features are summarized in (21). They are either constants or linear functions of N. None of them grows exponentially.

$$r^{\mathbf{f}}_{max} = 1, \Lambda^{\mathbf{f}} = \frac{1}{2}R^2 + R, L^{\mathbf{f}}_{max} = \frac{N}{L}, \left|\mathcal{V}^{\mathbf{f}}_G\right| = 2N. \tag{21}$$

5. CONCLUSION

The relationship revealed in this study between the complexity of neural networks and the compositional features in system models illustrates the reason why deep learning is an effective tool of overcoming the COD. The study raises more questions than answers. Many interesting problems are still widely open about the role of compositional structure in neural network design, as well as in the training and validation process.

ACKNOWLEDGEMENTS

This material is based upon activities supported by the National Science Foundation (under lnteragency Agreement #2202668 and Award #2134235) and Naval Research Laboratory, Monterey, California. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation and Naval Research Laboratory.

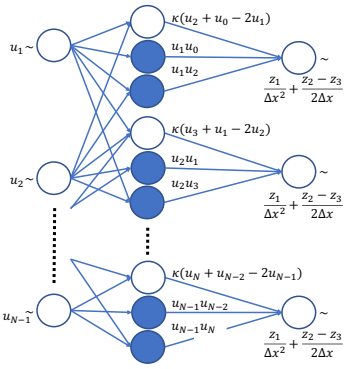

Fig. 4. DAG structure of the function (16). For a clear illustration, edges pointing to the first, the second, and the third layer are shown in red, blue and green respectively.

REFERENCES

Athay, T., Podmore, R., and Virmani, S. (1979). A practical method for the direct analysis of transient stability. *IEEE Transactions on Power Apparatus and Systems*, 98, 573–584.

B. Azmi, D. Kalise, K.K. (2020). ptimal feedback law recovery by gradient-augmented sparse polynomial regression. *arXiv:2007.09753*.

Han, J., Jentzen, A., and E., W. (2018). Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115, 8505–8510.

Izzo, D., Öztürk, E., and Märtens, M. (2019). Interplanetary transfers via deep representations of the optimal policy and/or of the value function. *arXiv:1904.08809*.

Kang, W. and Gong, Q. (2022). Neural network approximations of compositional functions with applications to dynamical systems. *SIAM Journal on Control and Optimization*, to applear.

Kang, W., Gong, Q., Nakamura-Zimmerer, T., and Fahroo, F. (2021a). Algorithms of data generation for deep learning and feedback design: A survey. *Physica D: Nonlinear Phenomena*, 425.

Kang, W., Sun, K., and Xu, L. (2021b). Data-driven computational methods for the domain of attraction and zubov's equation. *arXiv:2112.14415v1*.

Lorenz, E. (1996). Predictability – a problem partly solved. *ECMWF Seminar on Predictability*, I.

Nakamura-Zimmerer, T., Gong, Q., and Kang, W. (2021). Adaptive deep learning for high-dimensional Hamilton-Jacobi-Bellman equations. *SIAM J. Scientific Computing*, 43, 1221–1247.

Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707.

Sirignano, J. and Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *arXiv:1708.07469*.