Exponential expressivity in deep neural networks through transient chaos

Ben Poole
Stanford University
poole@cs.stanford.edu

Subhaneil Lahiri
Stanford University
sulahiri@stanford.edu

Maithra Raghu
Google Brain and Cornell University
maithrar@gmail.com

Jascha Sohl-Dickstein
Google Brain
jaschasd@google.com

Surya Ganguli
Stanford University
sganguli@stanford.edu

Abstract

We combine Riemannian geometry with the mean field theory of high dimensional chaos to study the nature of signal propagation in generic, deep neural networks with random weights. Our results reveal an order-to-chaos expressivity phase transition, with networks in the chaotic phase computing nonlinear functions whose global curvature grows *exponentially* with depth but not width. We prove this generic class of deep random functions cannot be efficiently computed by any shallow network, going beyond prior work restricted to the analysis of single functions. Moreover, we formalize and quantitatively demonstrate the long conjectured idea that deep networks can disentangle highly curved manifolds in input space into flat manifolds in hidden space. Our theoretical analysis of the expressive power of deep networks broadly applies to arbitrary nonlinearities, and provides a quantitative underpinning for previously abstract notions about the geometry of deep functions.

1 Introduction

Deep feedforward neural networks, with multiple hidden layers, have achieved remarkable performance across many domains [1–4]. A key factor thought to underlie their success is their high *expressivity*. This informal notion has manifested itself primarily in two forms of intuition. The first is that deep networks can compactly express highly complex functions over input space in a way that shallow networks with one hidden layer and the same number of neurons cannot. The second piece of intuition, which has captured the imagination of machine learning [5] and neuroscience [6] alike, is that deep neural networks can *disentangle* highly curved manifolds in input space into flattened manifolds in hidden space, to aid the performance of simple linear readouts. These intuitions, while attractive, have been difficult to formalize mathematically, and thereby rigorously test.

For the first intuition, seminal works have exhibited examples of particular functions that can be computed with a polynomial number of neurons (in the input dimension) in a deep network but require an exponential number of neurons in a shallow network [7–11]. This raises a central open question: are such functions merely rare curiosities, or is any function computed by a generic deep network not efficiently computable by a shallow network? The theoretical techniques employed in prior work both limited the applicability of theory to specific nonlinearities and dictated the particular measure of deep functional complexity involved. For example [7] focused on ReLu nonlinearities and number of linear regions as a complexity measure, while [8] focused on sum-product networks and the number of monomials as complexity measure, and [12] focused on Pfaffian nonlinearities and topological measures of complexity, like the sum of Betti numbers of a decision boundary. However,

Code to reproduce all results available at: `https://github.com/ganguli-lab/deepchaos`

see [13] for an interesting analysis of a general class of compositional functions. The limits of prior theoretical techniques raise another central question: is there a unifying theoretical framework for deep neural expressivity that is simultaneously applicable to arbitrary nonlinearities, generic networks, and a natural, general measure of functional complexity?

Here we attack both central problems of deep neural expressivity by combining a very different set of tools, namely Riemannian geometry [14] and dynamical mean field theory [15]. This novel combination enables us to show that for very broad classes of nonlinearities, even random deep neural networks can construct hidden internal representations whose global extrinsic curvature grows exponentially with depth but not width. Our geometric framework enables us to quantitatively define a notion of disentangling and verify this notion even in deep random networks. Furthermore, our methods yield insights into the emergent, deterministic nature of signal propagation through large random feedforward networks, revealing the existence of an order to chaos transition as a function of the statistics of weights and biases. We find that the transient, finite depth evolution in the chaotic regime underlies the origins of exponential expressivity in deep random networks.

In our companion paper [16], we study several related measures of expressivity in deep random neural networks with piecewise linear activations.

2 A mean field theory of deep nonlinear signal propagation

Consider a deep feedforward network with D layers of weights $\mathbf{W}^1, \ldots, \mathbf{W}^D$ and $D + 1$ layers of neural activity vectors $\mathbf{x}^0, \ldots, \mathbf{x}^D$, with N_l neurons in each layer l, so that $\mathbf{x}^l \in \mathbb{R}^{N_l}$ and $\mathbf{W}^l$ is an $N_l \times N_{l-1}$ weight matrix. The feedforward dynamics elicited by an input $\mathbf{x}^0$ is given by

$$\mathbf{x}^l = \phi(\mathbf{h}^l) \qquad \mathbf{h}^l = \mathbf{W}^l \mathbf{x}^{l-1} + \mathbf{b}^l \quad \text{for } l = 1, \ldots, D, \tag{1}$$

where $\mathbf{b}^l$ is a vector of biases, $\mathbf{h}^l$ is the pattern of inputs to neurons at layer l, and ϕ is a single neuron scalar nonlinearity that acts component-wise to transform inputs $\mathbf{h}^l$ to activities $\mathbf{x}^l$. We wish to understand the nature of typical functions computable by such networks, as a consequence of their depth. We therefore study ensembles of random networks in which each of the synaptic weights $\mathbf{W}_{ij}^l$ are drawn i.i.d. from a zero mean Gaussian with variance σ_w^2/N_{l-1}, while the biases are drawn i.i.d. from a zero mean Gaussian with variance σ_b^2. This weight scaling ensures that the input contribution to each individual neuron at layer l from activities in layer $l - 1$ remains $O(1)$, independent of the layer width N_{l-1}. This ensemble constitutes a maximum entropy distribution over deep neural networks, subject to constraints on the means and variances of weights and biases. This ensemble induces no further structure in the resulting set of deep functions, so its analysis provides an opportunity to understand the specific contribution of depth alone to the nature of typical functions computed by deep networks.

In the limit of large layer widths, $N_l \gg 1$, certain aspects of signal propagation through deep random neural networks take on an essentially deterministic character. This emergent determinism in large random neural networks enables us to understand how the Riemannian geometry of simple manifolds in the input layer $\mathbf{x}^0$ is typically modified as the manifold propagates into the deep layers. For example, consider the simplest case of a single input vector $\mathbf{x}^0$. As it propagates through the network, its length in downstream layers will change. We track this changing length by computing the normalized squared length of the input vector at each layer:

$$q^l = \frac{1}{N_l} \sum_{i=1}^{N_l} (\mathbf{h}_i^l)^2. \tag{2}$$

This length is the second moment of the empirical distribution of inputs $\mathbf{h}_i^l$ across all N_l neurons in layer l. For large N_l, this empirical distribution converges to a zero mean Gaussian since each $\mathbf{h}_i^l = \sum_j \mathbf{W}_{ij}^l \phi(\mathbf{h}_j^{l-1}) + \mathbf{b}_i^l$ is a weighted sum of a large number of uncorrelated random variables - i.e. the weights $\mathbf{W}_{ij}^l$ and biases $\mathbf{b}_i^l$, which are independent of the activity in previous layers. By propagating this Gaussian distribution across one layer, we obtain an iterative map for q^l in (2):

$$q^l = \mathcal{V}(q^{l-1} \,|\, \sigma_w, \sigma_b) \equiv \sigma_w^2 \int \mathcal{D}z \, \phi\left(\sqrt{q^{l-1}}z\right)^2 + \sigma_b^2, \quad \text{for} \quad l = 2, \ldots, D, \tag{3}$$

where $\mathcal{D}z = \frac{dz}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$ is the standard Gaussian measure, and the initial condition is $q^1 = \sigma_w^2 q^0 + \sigma_b^2$, where $q^0 = \frac{1}{N_0}\mathbf{x}^0 \cdot \mathbf{x}^0$ is the length in the initial activity layer. See Supplementary Material (SM)
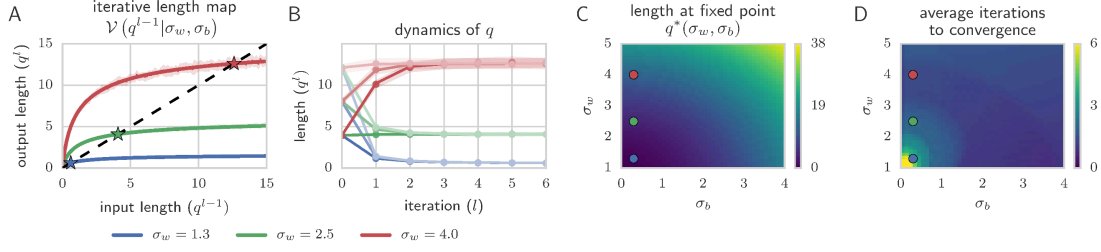
Figure 1: Dynamics of the squared length q^l for a sigmoidal network ($\phi(h) = \tanh(h)$) with 1000 hidden units. (A) The iterative length map in (3) for 3 different σ_w at $\sigma_b = 0.3$. Theoretical predictions (solid lines) match well with individual network simulations (dots). Stars reflect fixed points q^* of the map. (B) The iterative dynamics of the length map yields rapid convergence of q^l to its fixed point q^*, independent of initial condition (lines=theory; dots=simulation). (C) q^* as a function of σ_w and σ_b. (D) Number of iterations required to achieve $\leq 1\%$ fractional deviation off the fixed point. The (σ_b, σ_w) pairs in (A,B) are marked with color matched circles in (C,D).

for a derivation of (3). Intuitively, the integral over z in (3) replaces an average over the empirical distribution of $\mathbf{h}_i^l$ across neurons i in layer l at large layer width N_l.

The function $\mathcal{V}$ in (3) is an iterative variance, or length, map that predicts how the length of an input in (2) changes as it propagates through the network. This length map is plotted in Fig. 1A for the special case of a sigmoidal nonlinearity, $\phi(h) = \tanh(h)$. For monotonic nonlinearities, this length map is a monotonically increasing, concave function whose intersections with the unity line determine its fixed points $q^*(\sigma_w, \sigma_b)$. For $\sigma_b = 0$ and $\sigma_w < 1$, the only intersection is at $q^* = 0$. In this bias-free, small weight regime, the network shrinks all inputs to the origin. For $\sigma_w > 1$ and $\sigma_b = 0$, the $q^* = 0$ fixed point becomes unstable and the length map acquires a second nonzero fixed point, which is stable. In this bias-free, large weight regime, the network expands small inputs and contracts large inputs. Also, for any nonzero bias σ_b, the length map has a single stable non-zero fixed point. In such a regime, even with small weights, the injected biases at each layer prevent signals from decaying to 0. The dynamics of the length map leads to rapid convergence of length to its fixed point with depth (Fig. 1B,D), often within only 4 layers. The fixed points $q^*(\sigma_w, \sigma_b)$ are shown in Fig. 1C.

3 Transient chaos in deep networks

Now consider the layer-wise propagation of two inputs $\mathbf{x}^{0,1}$ and $\mathbf{x}^{0,2}$. The geometry of these two inputs as they propagate through the network is captured by the 2 by 2 matrix of inner products:

$$q_{ab}^l = \frac{1}{N_l} \sum_{i=1}^{N_l} \mathbf{h}_i^l(\mathbf{x}^{0,a}) \, \mathbf{h}_i^l(\mathbf{x}^{0,b}) \quad a, b \in \{1, 2\}. \tag{4}$$

The dynamics of the two diagonal terms are each theoretically predicted by the length map in (3). We derive (see SM) a correlation map $\mathcal{C}$ that predicts the layer-wise dynamics of q_{12}^l:

$$q_{12}^l = \mathcal{C}(c_{12}^{l-1}, q_{11}^{l-1}, q_{22}^{l-1} \mid \sigma_w, \sigma_b) \equiv \sigma_w^2 \int \mathcal{D}z_1 \, \mathcal{D}z_2 \, \phi(u_1) \, \phi(u_2) + \sigma_b^2, \tag{5}$$

$$u_1 = \sqrt{q_{11}^{l-1}} z_1, \qquad u_2 = \sqrt{q_{22}^{l-1}} \left[c_{12}^{l-1} z_1 + \sqrt{1 - (c_{12}^{l-1})^2} z_2 \right],$$

where $c_{12}^l = q_{12}^l(q_{11}^l q_{22}^l)^{-1/2}$ is the correlation coefficient. Here z_1 and z_2 are independent standard Gaussian variables, while u_1 and u_2 are correlated Gaussian variables with covariance matrix $\langle u_a u_b \rangle = q_{ab}^{l-1}$. Together, (3) and (5) constitute a theoretical prediction for the typical evolution of the geometry of 2 points in (4) in a fixed large network.

Analysis of these equations reveals an interesting order to chaos transition in the σ_w and σ_b plane. In particular, what happens to two nearby points as they propagate through the layers? Their relation to each other can be tracked by the correlation coefficient c_{12}^l between the two points, which approaches

3

a fixed point $c^*(\sigma_w, \sigma_b)$ at large depth. Since the length of each point rapidly converges to $q^*(\sigma_w, \sigma_b)$, as shown in Fig. 1BD, we can compute c^* by simply setting $q_{11}^l = q_{22}^l = q^*(\sigma_w, \sigma_b)$ in (5) and dividing by q^* to obtain an iterative correlation coefficient map, or $\mathcal{C}$-map, for c_{12}^l:

$$c_{12}^l = \frac{1}{q^*}\mathcal{C}(c_{12}^{l-1},\, q^*,\, q^* \,|\, \sigma_w, \sigma_b). \tag{6}$$

This $\mathcal{C}$-map is shown in Fig. 2A. It always has a fixed point at $c^* = 1$ as can be checked by direct calculation. However, the stability of this fixed point depends on the slope of the map at 1, which is

$$\chi_1 \equiv \left.\frac{\partial c_{12}^l}{\partial c_{12}^{l-1}}\right|_{c=1} = \sigma_w^2 \int \mathcal{D}z \left[\phi'\left(\sqrt{q^*}z\right)\right]^2. \tag{7}$$

See SM for a derivation of (7). If the slope χ_1 is less than 1, then the $\mathcal{C}$-map is above the unity line, the fixed point at 1 under the $\mathcal{C}$-map in (6) is stable, and nearby points become more similar over time.
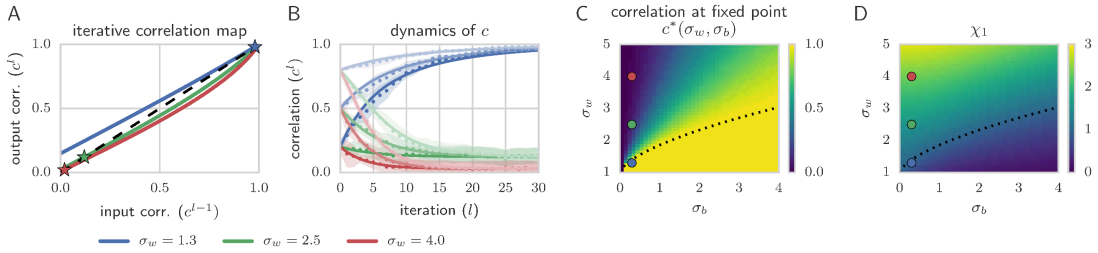

Figure 2: Dynamics of correlations, c_{12}^l, in a sigmoidal network with $\phi(h) = \tanh(h)$. (A) The $\mathcal{C}$-map in (6) for the same σ_w and $\sigma_b = 0.3$ as in Fig. 1A. (B) The $\mathcal{C}$-map dynamics, derived from both theory, through (6) (solid lines) and numerical simulations of (1) with $N_l = 1000$ (dots) (C) Fixed points c^* of the $\mathcal{C}$-map. (D) The slope of the $\mathcal{C}$-map at 1, χ_1, partitions the space (black dotted line at $\chi_1 = 1$) into chaotic ($\chi_1 > 1$, $c^* < 1$) and ordered ($\chi_1 < 1$, $c^* = 1$) regions.

Conversely, if $\chi_1 > 1$ then this fixed point is unstable, and nearby points separate as they propagate through the layers. Thus we can intuitively understand χ_1 as a multiplicative stretch factor. This intuition can be made precise by considering the Jacobian $\mathbf{J}_{ij}^l = \mathbf{W}_{ij}^l \phi'(\mathbf{h}_j^{l-1})$ at a point $\mathbf{h}^{l-1}$ with length q^*. $\mathbf{J}^l$ is a linear approximation of the network map from layer $l-1$ to l in the vicinity of $\mathbf{h}^{l-1}$. Therefore a small random perturbation $\mathbf{h}^{l-1} + \mathbf{u}$ will map to $\mathbf{h}^l + \mathbf{Ju}$. The growth of the perturbation, $||\mathbf{Ju}||_2^2/||\mathbf{u}||_2^2$ becomes $\chi_1(q^*)$ after averaging over the random perturbation $\mathbf{u}$, weight matrix $\mathbf{W}^l$, and Gaussian distribution of $\mathbf{h}_i^{l-1}$ across i. Thus χ_1 directly reflects the typical multiplicative growth or shrinkage of a random perturbation across one layer.

The dynamics of the iterative $\mathcal{C}$-map and its agreement with network simulations is shown in Fig. 2B. The correlation dynamics are much slower than the length dynamics because the $\mathcal{C}$-map is closer to the unity line (Fig. 2A) than the length map (Fig. 1A). Thus correlations typically take about 20 layers to approach the fixed point, while lengths need only 4. The fixed point c^* and slope χ_1 of the $\mathcal{C}$-map are shown in Fig. 2CD. For any fixed, finite σ_b, as σ_w increases three qualitative regions occur. For small σ_w, $c^* = 1$ is the only fixed point, and it is stable because $\chi_1 < 1$. In this strong bias regime, any two input points converge to each other as they propagate through the network. As σ_w increases, χ_1 increases and crosses 1, destabilizing the $c^* = 1$ fixed point. In this intermediate regime, a new stable fixed point c^* appears, which decreases as σ_w increases. Here an equal footing competition between weights and nonlinearities (which de-correlate inputs) and the biases (which correlate them), leads to a finite c^*. At larger σ_w, the strong weights overwhelm the biases and maximally de-correlate inputs to make them orthogonal, leading to a stable fixed point at $c^* = 0$.

Thus the equation $\chi_1(\sigma_w, \sigma_b) = 1$ yields a phase transition boundary in the (σ_w, σ_b) plane, separating it into a chaotic (or ordered) phase, in which nearby points separate (or converge). In dynamical systems theory, the logarithm of χ_1 is related to the well known Lyapunov exponent which is positive (or negative) for chaotic (or ordered) dynamics. However, in a feedforward network, the dynamics is truncated at a finite depth D, and hence the dynamics are a form of transient chaos.

4

4 The propagation of manifold geometry through deep networks

Now consider a 1 dimensional manifold $\mathbf{x}^0(\theta)$ in input space, where θ is an intrinsic scalar coordinate on the manifold. This manifold propagates to a new manifold $\mathbf{h}^l(\theta) = \mathbf{h}^l(\mathbf{x}^0(\theta))$ in the vector space of inputs to layer l. The typical geometry of the manifold in the l'th layer is summarized by $q^l(\theta_1, \theta_2)$, which for any θ_1 and θ_2 is defined by (4) with the choice $\mathbf{x}^{0,a} = \mathbf{x}^0(\theta_1)$ and $\mathbf{x}^{0,b} = \mathbf{x}^0(\theta_2)$. The theory for the propagation of pairs of points applies to all pairs of points on the manifold, so intuitively, we expect that in the chaotic phase of a sigmoidal network, the manifold should in some sense de-correlate, and become more complex, while in the ordered phase the manifold should contract around a central point. This theoretical prediction of equations (3) and (5) is quantitatively confirmed in simulations in Fig. 3, when the input is a simple manifold, the circle, $\mathbf{h}^1(\theta) = \sqrt{N_1 q} \left[\mathbf{u}^0 \cos(\theta) + \mathbf{u}^1 \sin(\theta) \right]$, where $\mathbf{u}^0$ and $\mathbf{u}^1$ form an orthonormal basis for a 2 dimensional subspace of $\mathbb{R}^{N_1}$ in which the circle lives. The scaling is chosen so that each neuron has input activity $O(1)$. Also, for simplicity, we choose the fixed point radius $q = q^*$ in Fig. 3.
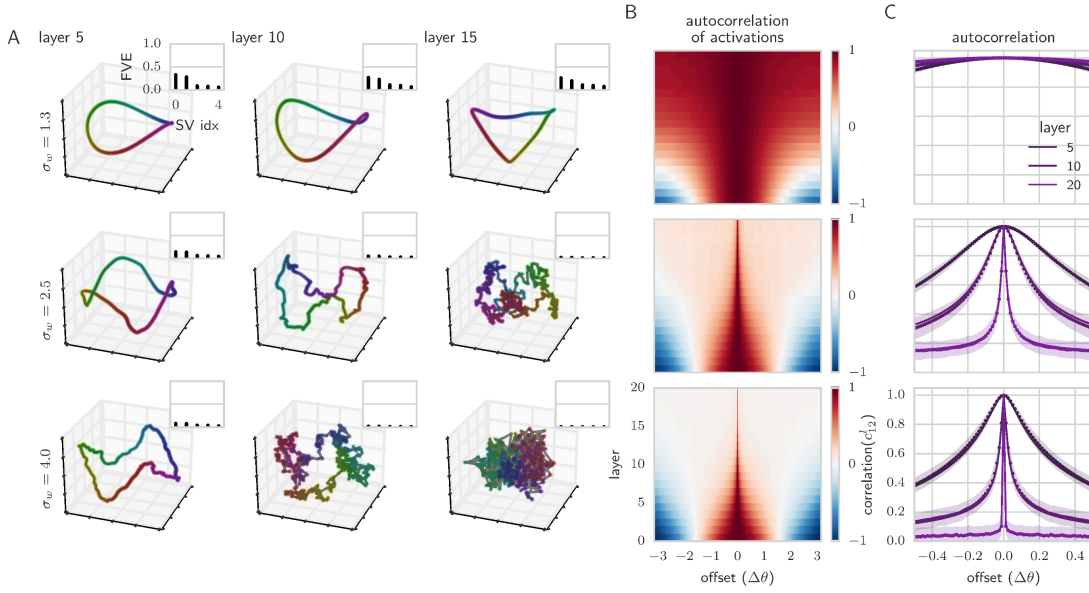

Figure 3: Propagating a circle through three random sigmoidal networks with varying σ_w and fixed $\sigma_b = 0.3$. (A) Projection of hidden inputs of simulated networks at layer 5 and 10 onto their first three principal components. Insets show the fraction of variance explained by the first 5 singular values. For large weights (bottom), the distribution of singular values gets flatter and the projected curve is more tangled. (B) The autocorrelation, $c_{12}^l(\Delta\theta) = \int d\theta\, q^l(\theta, \theta + \Delta\theta)/q^*$, of hidden inputs as a function of layer for simulated networks. (C) The theoretical predictions from (6) (solid lines) compared to the average (dots) and standard deviation across θ (shaded) in a simulated network.

To quantitatively understand the layer-wise growth of complexity of this manifold, it is useful to turn to concepts in Riemannian geometry [14]. First, at each point θ, the manifold $\mathbf{h}(\theta)$ (we temporarily suppress the layer index l) has a tangent, or velocity vector $\mathbf{v}(\theta) = \partial_\theta \mathbf{h}(\theta)$. Intuitively, curvature is related to how quickly this tangent vector rotates in the ambient space $\mathbb{R}^N$ as one moves along the manifold, or in essence the acceleration vector $\mathbf{a}(\theta) = \partial_\theta \mathbf{v}(\theta)$. Now at each point θ, when both are nonzero, $\mathbf{v}(\theta)$ and $\mathbf{a}(\theta)$ span a 2 dimensional subspace of $\mathbb{R}^N$. Within this subspace, there is a unique circle of radius $R(\theta)$ that has the same position, velocity and acceleration vector as the curve $\mathbf{h}(\theta)$ at θ. This circle is known as the osculating circle (Fig. 4A), and the extrinsic curvature $\kappa(\theta)$ of the curve is defined as $\kappa(\theta) = 1/R(\theta)$. Thus, intuitively, small radii of curvature $R(\theta)$ imply high extrinsic curvature $\kappa(\theta)$. The extrinsic curvature of a curve depends only on its image in $\mathbb{R}^N$ and is invariant with respect to the particular parameterization $\theta \to \mathbf{h}(\theta)$. For any parameterization, an explicit expression for $\kappa(\theta)$ is given by $\kappa(\theta) = (\mathbf{v}\cdot\mathbf{v})^{-3/2}\sqrt{(\mathbf{v}\cdot\mathbf{v})(\mathbf{a}\cdot\mathbf{a}) - (\mathbf{v}\cdot\mathbf{a})^2}$ [14]. Note that under a unit speed parameterization of the curve, so that $\mathbf{v}(\theta)\cdot\mathbf{v}(\theta) = 1$, we have $\mathbf{v}(\theta)\cdot\mathbf{a}(\theta) = 0$, and $\kappa(\theta)$ is simply the norm of the acceleration vector.
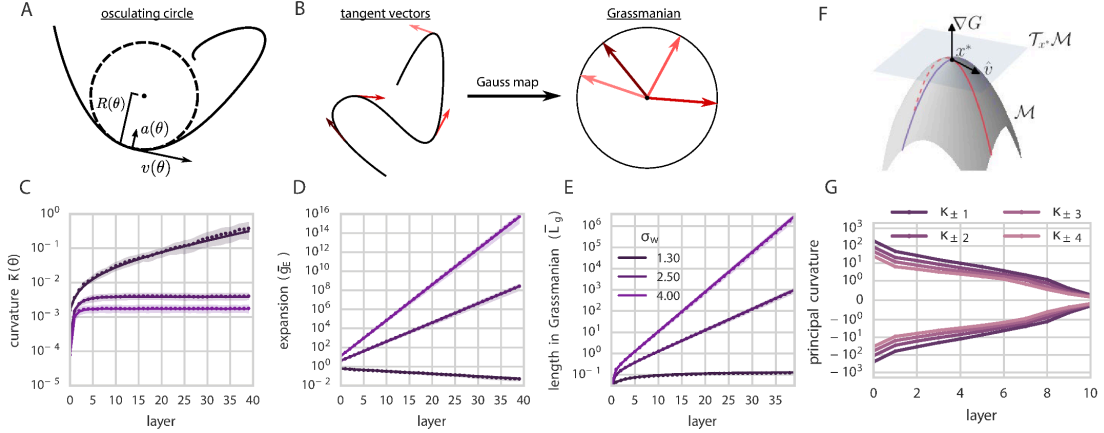
Figure 4: Propagation of extrinsic curvature and length in a network with 1000 hidden units. (A) An osculating circle. (B) A curve with unit tangent vectors at 4 points in ambient space, and the image of these points under the Gauss map. (C-E) Propagation of curvature metrics based on both theory derived from iterative maps in (3), (6) and (8) (solid lines) and simulations using (1) (dots). (F) Schematic of the normal vector, tangent plane, and principal curvatures for a 2D manifold embedded in $\mathbb{R}^3$. (G) average principal curvatures for the largest and smallest 4 principal curvatures ($\kappa_{\pm 1}, \ldots, \kappa_{\pm 4}$) across locations θ within one network. The principal curvatures all grow exponentially as we backpropagate to the input layer. Panels F,G are discussed in Sec. 5.

Another measure of the curve's complexity is the length $\mathcal{L}^E$ of its image in the ambient Euclidean space. The Euclidean metric in $\mathbb{R}^N$ induces a metric $g^E(\theta) = \mathbf{v}(\theta) \cdot \mathbf{v}(\theta)$ on the curve, so that the distance $d\mathcal{L}^E$ moved in $\mathbb{R}^N$ as one moves from θ to $\theta + d\theta$ on the curve is $d\mathcal{L}^E = \sqrt{g^E(\theta)}d\theta$. The total curve length is $\mathcal{L}^E = \int \sqrt{g^E(\theta)}d\theta$. However, even straight line segments can have a large Euclidean length. Another interesting measure of length that takes into account curvature, is the length of the image of the curve under the Gauss map. For a K dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^N$, the Gauss map (Fig. 4B) maps a point $\theta \in \mathcal{M}$ to its K dimensional tangent plane $\mathcal{T}_\theta \mathcal{M} \in \mathfrak{G}_{K,N}$, where $\mathfrak{G}_{K,N}$ is the Grassmannian manifold of all K dimensional subspaces in $\mathbb{R}^N$. In the special case of $K = 1$, $\mathfrak{G}_{K,N}$ is the sphere $\mathbb{S}^{N-1}$ with antipodal points identified, since a 1-dimensional subspace can be identified with a unit vector, modulo sign. The Gauss map takes a point θ on the curve and maps it to the unit velocity vector $\hat{\mathbf{v}}(\theta) = \mathbf{v}(\theta)/\sqrt{\mathbf{v}(\theta) \cdot \mathbf{v}(\theta)}$. In particular, the natural metric on $\mathbb{S}^{N-1}$ induces a Gauss metric on the curve, given by $g^G(\theta) = (\partial_\theta \hat{\mathbf{v}}(\theta)) \cdot (\partial_\theta \hat{\mathbf{v}}(\theta))$, which measures how quickly the unit tangent vector $\hat{\mathbf{v}}(\theta)$ changes as θ changes. Thus the distance $d\mathcal{L}^G$ moved in the Grassmannian $\mathfrak{G}_{K,N}$ as one moves from θ to $\theta + d\theta$ on the curve is $d\mathcal{L}^G = \sqrt{g^G(\theta)}d\theta$, and the length of the curve under the Gauss map is $\mathcal{L}^G = \int \sqrt{g^G(\theta)}d\theta$. Furthermore, the Gauss metric is related to the extrinsic curvature and the Euclidean metric via the relation $g^G(\theta) = \kappa(\theta)^2 g^E(\theta)$ [14].

To illustrate these concepts, it is useful to compute all of them for the circle $\mathbf{h}^1(\theta)$ defined above: $g^E(\theta) = Nq$, $\mathcal{L}^E = 2\pi\sqrt{Nq}$, $\kappa(\theta) = 1/\sqrt{Nq}$, $g^G(\theta) = 1$, and $\mathcal{L}^G = 2\pi$. As expected, $\kappa(\theta)$ is the inverse of the radius of curvature, which is $\sqrt{Nq}$. Now consider how these quantities change if the circle is scaled up so that $\mathbf{h}(\theta) \to \chi \mathbf{h}(\theta)$. The length $\mathcal{L}^E$ and radius scale up by χ, but the curvature κ scales down as χ^{-1}, and so $\mathcal{L}^G$ does not change. Thus linear expansion increases length and decreases curvature, thereby maintaining constant Grassmannian length $\mathcal{L}^G$.

We now show that nonlinear propagation of this same circle through a deep network can behave very differently from linear expansion: in the chaotic regime, length can increase without any decrease in extrinsic curvature! To remove the scaling with N in the above quantities, we will work with the renormalized quantities $\bar{\kappa} = \sqrt{N}\kappa$, $\bar{g}^E = \frac{1}{N}g^E$, and $\bar{\mathcal{L}}^E = \frac{1}{\sqrt{N}}\mathcal{L}^E$. Thus, $1/(\bar{\kappa})^2$ can be thought of as a radius of curvature squared per neuron of the osculating circle, while $(\bar{\mathcal{L}}^E)^2$ is the squared Euclidean length of the curve per neuron. For the circle, these quantities are q and $2\pi q$ respectively. For simplicity, in the inputs to the first layer of neurons, we begin with a circle $\mathbf{h}^1(\theta)$ with squared

radius per neuron $q^1 = q^*$, so this radius is already at the fixed point of the length map in (3). In the SM, we derive an iterative formula for the extrinsic curvature and Euclidean metric of this manifold as it propagates through the layers of a deep network:

$$\bar{g}^{E,l} = \chi_1\,\bar{g}^{E,l-1} \qquad (\bar{\kappa}^l)^2 = 3\frac{\chi_2}{\chi_1^2} + \frac{1}{\chi_1}(\bar{\kappa}^{l-1})^2, \qquad \bar{g}^{E,1} = q^*, \qquad (\bar{\kappa}^1)^2 = 1/q^*. \quad (8)$$

where χ_1 is the stretch factor defined in (7) and χ_2 is defined analogously as

$$\chi_2 = \sigma_w^2 \int \mathcal{D}z \left[\phi''\left(\sqrt{q^*}z\right)\right]^2. \quad (9)$$

χ_2 is closely related to the second derivative of the $\mathcal{C}$-map in (6) at $c_{12}^{l-1} = 1$; this second derivative is $\chi_2 q^*$. See SM for a derivation of the evolution equations (8) for the extrinsic geometry of a curve as it propagates through a deep network.

Intriguingly for a sigmoidal neural network, these evolution equations behave very differently in the chaotic ($\chi_1 > 1$) versus ordered ($\chi_1 < 1$) phase. In the chaotic phase, the Euclidean metric $\bar{g}^E$ grows *exponentially* with depth due to multiplicative stretching through χ_1. This stretching does multiplicatively attenuate any curvature in layer $l-1$ by a factor $1/\chi_1$ (see the update equation for $\bar{\kappa}^l$ in (8)), but new curvature is added in due to a nonzero χ_2, which originates from the curvature of the single neuron nonlinearity in (9). Thus, unlike in linear expansion, extrinsic curvature is not lost, but maintained, and ultimately approaches a fixed point $\bar{\kappa}^*$. This implies that the global curvature measure $\bar{\mathcal{L}}^G$ grows exponentially with depth. These highly nontrivial predictions of the metric and curvature evolution equations in (8) are quantitatively confirmed in simulations in Figure 4C-E.

Intuitively, this exponential growth of global curvature $\bar{\mathcal{L}}^G$ in the chaotic phase implies that the curve explores many different tangent directions in hidden representation space. This further implies that the coordinate functions of the embedding $\mathbf{h}_i^l(\theta)$ become highly complex curved basis functions on the input manifold coordinate θ, allowing a deep network to compute exponentially complex functions over simple low dimensional manifolds (Figure 5A-C, details in SM). In our companion paper [16], we further develop the relationship between length and expressivity in terms of the number of achievable classification patterns on a set of inputs. Moreover, we explore how training a single layer at different depths from the output affects network performance.
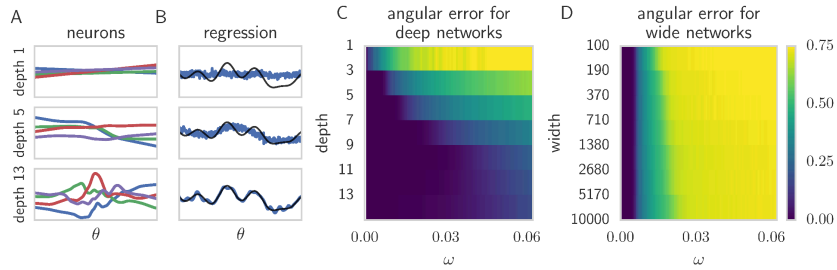

Figure 5: Deep networks in the chaotic regime are more expressive than shallow networks. (A) Activity of four different neurons in the output layer as a function of the input, θ for three networks of different depth (width $N_l = 1,000$). (B) Linear regression of the output activity onto a random function (black) shows closer predictions (blue) with deeper networks (bottom) than shallow networks (top). (C) Decomposing the prediction error by frequency shows shallow networks cannot capture high frequency content in random functions but deep networks can (yellow=high error). (D) Increasing the width of a one hidden layer network up to $10,000$ does not decrease error at high frequencies.

5 Shallow networks cannot achieve exponential expressivity

Consider a shallow network with 1 hidden layer $\mathbf{x}^1$, one input layer $\mathbf{x}^0$, with $\mathbf{x}^1 = \phi(\mathbf{W}^1\mathbf{x}^0) + \mathbf{b}^1$, and a linear readout layer. How complex can the hidden representation be as a function of its width N_1, relative to the results above for depth? We prove a general upper bound on $\mathcal{L}^E$ (see SM):

7

Theorem 1. *Suppose $\phi(h)$ is monotonically non-decreasing with bounded dynamic range R, i.e. $\max_h \phi(h) - \min_h \phi(h) = R$. Further suppose that $\mathbf{x}^0(\theta)$ is a curve in input space such that no 1D projection of $\partial_\theta \mathbf{x}(\theta)$ changes sign more than s times over the range of θ. Then for any choice of $\mathbf{W}^1$ and $\mathbf{b}^1$ the Euclidean length of $\mathbf{x}^1(\theta)$, satisfies $\mathcal{L}^E \leq N_1(1 + s)R$.*

For the circle input, $s = 1$ and for the $\tanh$ nonlinearity, $R = 2$, so in this special case, the normalized length $\bar{\mathcal{L}}^E \leq 2\sqrt{N_1}$. In contrast, for deep networks in the chaotic regime $\bar{\mathcal{L}}^E$ grows exponentially with depth in $\mathbf{h}$ space, and so consequently also in $\mathbf{x}$ space. Therefore the length of curves typically expand exponentially in depth even for random deep networks, but can only expand as the square root of width no matter what shallow network is chosen. Moreover, as we have seen above, it is the exponential growth of $\mathcal{L}^{\mathcal{E}}$ that fundamentally drives the exponential growth of $\bar{\mathcal{L}}^G$ with depth. Indeed shallow random networks exhibit minimal growth in expressivity even at large widths (Figure 5D).

6   Classification boundaries acquire exponential local curvature with depth

We have focused so far on how simple manifolds in input space can acquire both exponential Euclidean and Grassmannian length with depth, thereby exponentially de-correlating and filling up hidden representation space. Another natural question is how the complexity of a decision boundary grows as it is backpropagated to the input layer. Consider a linear classifier $y = \text{sgn}(\boldsymbol{\beta} \cdot \mathbf{x}^D - \beta_0)$ acting on the final layer. In this layer, the $N - 1$ dimensional decision boundary is the hyperplane $\boldsymbol{\beta} \cdot \mathbf{x}^D - \beta_0 = 0$. However, in the input layer $\mathbf{x}^0$, the decision boundary is a curved $N - 1$ dimensional manifold $\mathcal{M}$ that arises as the solution set of the nonlinear equation $G(\mathbf{x}^0) \equiv \boldsymbol{\beta} \cdot \mathbf{x}^D(\mathbf{x}^0) - \beta_0 = 0$, where $\mathbf{x}^D(\mathbf{x}^0)$ is the nonlinear feedforward map from input to output.

At any point $\mathbf{x}^*$ on the decision boundary in layer l, the gradient $\vec{\nabla} G$ is perpendicular to the $N - 1$ dimensional tangent plane $\mathcal{T}_{x^*}\mathcal{M}$ (see Fig. 4F). The normal vector $\vec{\nabla} G$, along with any unit tangent vector $\hat{\mathbf{v}} \in \mathcal{T}_{x^*}\mathcal{M}$, spans a 2 dimensional subspace whose intersection with $\mathcal{M}$ yields a geodesic curve in $\mathcal{M}$ passing through x^* with velocity vector $\hat{\mathbf{v}}$. This geodesic will have extrinsic curvature $\kappa(\mathbf{x}^*, \hat{\mathbf{v}})$. Maximizing this curvature over $\hat{\mathbf{v}}$ yields the first principal curvature $\kappa_1(\mathbf{x}^*)$. A sequence of successive maximizations of $\kappa(\mathbf{x}^*, \hat{\mathbf{v}})$, while constraining $\hat{\mathbf{v}}$ to be perpendicular to all previous solutions, yields the sequence of principal curvatures $\kappa_1(\mathbf{x}^*) \geq \kappa_2(\mathbf{x}^*) \geq \cdots \geq \kappa_{N-1}(\mathbf{x}^*)$. These principal curvatures arise as the eigenvalues of a normalized Hessian operator projected onto the tangent plane $\mathcal{T}_{x^*}\mathcal{M}$: $\mathcal{H} = ||\vec{\nabla} G||_2^{-1} \mathbf{P} \frac{\partial^2 G}{\partial \mathbf{x} \partial \mathbf{x}^T} \mathbf{P}$, where $\mathbf{P} = \mathbf{I} - \widehat{\nabla} G \widehat{\nabla} G^T$ is the projection operator onto $\mathcal{T}_{x^*}\mathcal{M}$ and $\widehat{\nabla} G$ is the unit normal vector [14]. Intuitively, near $\mathbf{x}^*$, the decision boundary $\mathcal{M}$ can be approximated as a paraboloid with a quadratic form $\mathcal{H}$ whose $N - 1$ eigenvalues are the principal curvatures $\kappa_1, \ldots, \kappa_{N-1}$ (Fig. 4F).

We compute these curvatures numerically as a function of depth in Fig. 4G (see SM for details). We find, remarkably, that a subset of principal curvatures grow *exponentially* with depth. Here the principal curvatures are signed, with positive (negative) curvature indicating that the associated geodesic curves towards (away from) the normal vector $\vec{\nabla} G$. Thus the decision boundary can become exponentially curved with depth, enabling highly complex classifications. Moreover, this exponentially curved boundary is disentangled and mapped to a flat boundary in the output layer.

7   Discussion

Fundamentally, neural networks compute nonlinear maps between high dimensional spaces, for example from $\mathbb{R}^{N_1} \to \mathbb{R}^{N_D}$, and it is unclear what the most appropriate mathematics is for understanding such daunting spaces of maps. Previous works have attacked this problem by restricting the nature of the nonlinearity involved (e.g. piecewise linear, sum-product, or Pfaffian) and thereby restricting the space of maps to those amenable to special theoretical analysis methods (combinatorics, polynomial relations, or topological invariants). We have begun a preliminary exploration of the expressivity of such deep functions based on Riemannian geometry and dynamical mean field theory. We demonstrate that networks in a chaotic phase compactly exhibit functions that exponentially grow the global curvature of simple one dimensional manifolds from input to output and the local curvature of simple co-dimension one manifolds from output to input. The former captures the notion that deep neural networks can efficiently compute highly expressive functions in ways that shallow networks

cannot, while the latter quantifies and demonstrates the power of deep neural networks to disentangle curved input manifolds, an attractive idea that has eluded formal quantification.

Moreover, our analysis of a maximum entropy distribution over deep networks constitutes an important null model of deep signal propagation that can be used to assess and understand different behavior in trained networks. For example, the metrics we have adapted from Riemannian geometry, combined with an understanding of their behavior in random networks, may provide a basis for understanding what is special about trained networks. Furthermore, while we have focused on the notion of input-output chaos, the duality between inputs and synaptic weights imply a form of weight chaos, in which deep neural networks rapidly traverse function space as weights change (see SM). Indeed, just as autocorrelation lengths between outputs as a function of inputs shrink exponentially with depth, so too will autocorrelations between outputs as a function of weights.

But more generally, to understand functions, we often look to their graphs. The graph of a map from $\mathbb{R}^{N_1} \to \mathbb{R}^{N_D}$ is an $\mathbb{R}^{N_1}$ dimensional submanifold of $\mathbb{R}^{N_1+N_D}$, and therefore has both high dimension and co-dimension. We speculate that many of the secrets of deep learning may be uncovered by studying the geometry of this graph as a Riemannian manifold, and understanding how it changes with both depth and learning.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[3] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[4] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513, 2015.

[5] Yoshua Bengio, Aaron Courville, and Pierre Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8): 1798–1828, 2013.

[6] James J DiCarlo and David D Cox. Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–341, 2007.

[7] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932, 2014.

[8] Olivier Delalleau and Yoshua Bengio. Shallow vs. deep sum-product networks. In *Advances in Neural Information Processing Systems*, pages 666–674, 2011.

[9] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. *arXiv preprint arXiv:1512.03965*, 2015.

[10] Matus Telgarsky. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101*, 2015.

[11] James Martens, Arkadev Chattopadhya, Toni Pitassi, and Richard Zemel. On the representational efficiency of restricted boltzmann machines. In *Advances in Neural Information Processing Systems*, pages 2877–2885, 2013.

[12] Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(8):1553–1565, 2014.

[13] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. Learning real and boolean functions: When is deep better than shallow. *arXiv preprint arXiv:1603.00988*, 2016.

[14] John M Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

[15] Haim Sompolinsky, A Crisanti, and HJ Sommers. Chaos in random neural networks. *Physical Review Letters*, 61(3):259, 1988.

[16] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. *arXiv preprint*, 2016.

Supplementary Material

Below is a series of appendices giving derivations of results in the main paper, followed by details of results along with more visualizations.

Note: code to programmatically reproduce all plots in the paper in Jupyter notebooks will be released upon publication.

A   Derivation of a transient dynamical mean field theory for deep networks

We study a deep feedforward network with D layers of weights $\mathbf{W}^1, \ldots, \mathbf{W}^D$ and $D + 1$ layers of neural activity vectors $\mathbf{x}^0, \ldots, \mathbf{x}^D$, with N_l neurons in each layer l, so that $\mathbf{x}^l \in \mathbb{R}^{N_l}$ and $\mathbf{W}^l$ is an $N_l \times N_{l-1}$ weight matrix. The feedforward dynamics elicited by an input $\mathbf{x}^0$ is given by

$$\mathbf{x}^l = \phi(\mathbf{h}^l) \qquad \mathbf{h}^l = \mathbf{W}^l \, \mathbf{x}^{l-1} + \mathbf{b}^l \quad \text{for } l = 1, \ldots, D, \tag{10}$$

where $\mathbf{b}^l$ is a vector of biases, $\mathbf{h}^l$ is the pattern of inputs to neurons at layer l, and ϕ is a single neuron scalar nonlinearity that acts component-wise to transform inputs $\mathbf{h}^l$ to activities $\mathbf{x}^l$. The synaptic weights $\mathbf{W}_{ij}^l$ are drawn i.i.d. from a zero mean Gaussian with variance σ_w^2 / N_{l-1}, while the biases are drawn i.i.d. from a zero mean Gaussian with variance σ_b^2. This weight scaling ensures that the input contribution to each individual neuron at layer l from activities in layer $l - 1$ remains $O(1)$, independent of the layer width N_{l-1}.

A.1   Derivation of the length map

As a single input point $\mathbf{x}^0$ propagates through the network, it's length in downstream layers can either grow or shrink. To track the propagation of this length, we track the normalized squared length of the input vector at each layer,

$$q^l = \frac{1}{N_l} \sum_{i=1}^{N_l} (\mathbf{h}_i^l)^2. \tag{11}$$

This length is the second moment of the empirical distribution of inputs $\mathbf{h}_i^l$ across all N_l neurons in layer l for a *fixed* set of weights. This empirical distribution is expected to be Gaussian for large N_l, since each individual $\mathbf{h}_i^l = \mathbf{w}^{l,i} \cdot \phi(\mathbf{h}^{l-1}) + \mathbf{b}_i^l$ is Gaussian distributed, as a sum of a large number of independent random variables, and each $\mathbf{h}_i^l$ is independent of $\mathbf{h}_j^l$ for $i \neq j$ because the synaptic weights vectors and biases into each neuron are chosen independently.

While the mean of this Gaussian is 0, its variance can be computed by considering the variance of the input to a single neuron:

$$q^l = \left\langle (\mathbf{h}_i^l)^2 \right\rangle = \left\langle \left[\mathbf{w}^{l,i} \cdot \phi(\mathbf{h}^{l-1}) \right]^2 \right\rangle + \left\langle (\mathbf{b}_i^l)^2 \right\rangle = \sigma_w^2 \frac{1}{N_{l-1}} \sum_{i=1}^{N_{l-1}} \phi(\mathbf{h}_i^{l-1})^2 + \sigma_b^2, \tag{12}$$

where $\langle \cdot \rangle$ denotes an average over the distribution of weights and biases into neuron i at layer l. Here we have used the identity $\langle \mathbf{w}_j^{l,i} \mathbf{w}_k^{l,i} \rangle = \delta_{jk} \, \sigma_w^2 / N_{l-1}$. Now the empirical distribution of inputs across layer $l - 1$ is also Gaussian, with mean zero and variance q^{l-1}. Therefore we can replace the average over neurons in layer $l - 1$ in (12) with an integral over a Gaussian random variable, obtaining

$$q^l = \mathcal{V}(q^{l-1} \,|\, \sigma_w, \sigma_b) \equiv \sigma_w^2 \int \mathcal{D}z \, \phi \left(\sqrt{q^{l-1}} z \right)^2 + \sigma_b^2, \quad \text{for} \quad l = 2, \ldots, D, \tag{13}$$

where $\mathcal{D}z = \frac{dz}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$ is the standard Gaussian measure, and the initial condition for the variance map is $q^1 = \sigma_w^2 q_0 + \sigma_b^2$, where $q^0 = \frac{1}{N_0} \mathbf{x}^0 \cdot \mathbf{x}^0$ is the length in the initial activity layer. The function $\mathcal{V}$ in (13) is an iterative variance map that predicts how the length of an input in (11) changes as it propagates through the network. Its derivation relies on the well-known self-averaging assumption in the statistical physics of disordered systems, which, in our context, means that the empirical distribution of inputs *across* neurons for a *fixed* network converges for large width, to the distribution of inputs to a *single* neuron *across* random networks.

A.2 Derivation of a correlation map for the propagation of two points

Now consider the layer-wise propagation of two inputs $\mathbf{x}^{0,1}$ and $\mathbf{x}^{0,2}$. The geometry of these two inputs as they propagate through the layers is captured by the 2 by 2 matrix of inner products

$$q_{ab}^l = \frac{1}{N_l} \sum_{i=1}^{N_l} \mathbf{h}_i^l(\mathbf{x}^{0,a}) \, \mathbf{h}_i^l(\mathbf{x}^{0,b}) \quad a, b \in \{1, 2\}. \tag{14}$$

The *joint* empirical distribution of $\mathbf{h}_i^l(\mathbf{x}^{0,a})$ and $\mathbf{h}_i^l(\mathbf{x}^{0,a})$ across i at large N_l will converge to a 2 dimensional Gaussian distribution with covariance q_{ab}^l. Propagating this joint distribution forward one layer using ideas similar to the derivation above for 1 input yields

$$q_{12}^l = \mathcal{C}(c_{12}^{l-1}, q_{11}^{l-1}, q_{22}^{l-1} \,|\, \sigma_w, \sigma_b) \equiv \sigma_w^2 \int \mathcal{D}z_1 \, \mathcal{D}z_2 \, \phi(u_1) \, \phi(u_2) + \sigma_b^2, \tag{15}$$

$$u_1 = \sqrt{q_{11}^{l-1}} z_1, \qquad u_2 = \sqrt{q_{22}^{l-1}} \left[c_{12}^{l-1} z_1 + \sqrt{1 - (c_{12}^{l-1})^2} z_2 \right],$$

where $c_{12}^l = \frac{q_{12}^l}{\sqrt{q_{11}^l} \sqrt{q_{12}^l}}$ is the correlation coefficient (CC). Here z_1 and z_2 are independent standard Gaussian variables, while u_1 and u_2 are correlated Gaussian variables with covariance matrix $\langle u_a u_b \rangle = q_{ab}^{l-1}$. The integration over z_1 and z_2 can be thought of as the large N_l limit of sums over $\mathbf{h}_i^l(\mathbf{x}^{0,a})$ and $\mathbf{h}_i^l(\mathbf{x}^{0,b})$.

When both input points are at their fixed point length, q^*, the dynamics of their correlation coefficient can be obtained by simply setting $q_{11}^l = q_{22}^l = q^*(\sigma_w, \sigma_b)$ in (15) and dividing by q^* to obtain a recursion relation for c_{12}^l:

$$c_{12}^l = \frac{1}{q^*} \mathcal{C}(c_{12}^{l-1}, q^*, q^* \,|\, \sigma_w, \sigma_b) \tag{16}$$

Direct calculation reveals that $c_{12}^l(1) = 1$ as expected. Of particular interest is the slope χ_1 of this map at 1. A direct, if tedious calculation shows that

$$\frac{\partial c_{12}^l}{\partial c_{12}^{l-1}} = \sigma_w^2 \int \mathcal{D}z_1 \, \mathcal{D}z_2 \, \phi'(u_1) \, \phi'(u_2). \tag{17}$$

To obtain this result, one has to apply the chain rule and product rule from calculus, as well as employ the identity

$$\int \mathcal{D}z F(z) z = \int \mathcal{D}z F'(z), \tag{18}$$

which can be obtained via integration by parts. Evaluating the derivative at 1 yields

$$\chi_1 \equiv \left. \frac{\partial c_{12}^l}{\partial c_{12}^{l-1}} \right|_{c=1} = \sigma_w^2 \int \mathcal{D}z \left[\phi'\left(\sqrt{q^*} z\right) \right]^2. \tag{19}$$

B Derivation of evolution equations for Riemannian curvature

Here we derive recursion relations for Riemannian curvature quantitites.

B.1 Curvature and length in terms of inner products

Consider a translation invariant manifold, or 1D curve $\mathbf{h}(\theta) \in \mathbb{R}^N$ that is on some constant radius sphere so that

$$q(\theta_1, \theta_2) = Q(\theta_1 - \theta_2) = \mathbf{h}(\theta_1) \cdot \mathbf{h}(\theta_2), \tag{20}$$

with $Q(0) = Nq^*$. At large N, the inner-product structure of translation invariant manifolds remains approximately translation invariant as it propagates through the network. Therefore, at large N, we can express inner products of derivatives of $\mathbf{h}$ in terms of derivatives of Q. For example, the Euclidean metric g^E is given by

$$g^E(\theta) = \partial_\theta \mathbf{h}(\theta) \cdot \partial_\theta \mathbf{h}(\theta) = -\ddot{Q}(0). \tag{21}$$

Here, each dot is a short hand notation for derivative w.r.t. θ. Also, the extrinsic curvature

$$\kappa(\theta) = \sqrt{\frac{(\mathbf{v} \cdot \mathbf{v})(\mathbf{a} \cdot \mathbf{a}) - (\mathbf{v} \cdot \mathbf{a})^2}{(\mathbf{v} \cdot \mathbf{v})^3}}, \tag{22}$$

where $\mathbf{v}(\theta) = \partial_\theta \mathbf{h}(\theta)$ and $\mathbf{a}(\theta) = \partial_\theta^2 \mathbf{h}(\theta)$, simplifies to

$$\kappa(\theta) = \frac{\dddot{Q}(0)}{\ddot{Q}(0)^2}. \tag{23}$$

Now if the translation invariant manifold lives on a sphere of radius Nq^* where q^* is the fixed point radius of the length map, then its radius does not change as it propagates through the system. Then we can also express g^E and κ in terms of the correlation coefficient function $c(\theta) = Q(\theta)/q^*$ (up to a factor of N). Thus to understand the propagation of local quantities like Euclidean length and curvature, we need to understand the propagation of derivatives of $c(\theta)$ at $\theta = 0$ under the $\mathcal{C}$-map in (16). Note that $c(\theta)$ is symmetric and achieves a maximum value of 1 at $\theta = 0$. Thus the function $H^1(\theta) = 1 - c(\theta)$ is symmetric with a minimum at $\theta = 0$. We consider the propagation of H^1 though the $\mathcal{C}$-map. But first we consider the propagation of derivatives under function composition in general.

B.2 Behavior of first and second derivatives under function composition

Assume $H^1(\Delta t)$ is an even function and $H^1(0) = 0$, so that its Taylor expansion can be written as $H^1(\Delta t) = \frac{1}{2}\ddot{H}^1(0)\Delta t^2 + \frac{1}{4}\ddddot{H}^1(0)\Delta t^4 + \dots$. We are interested in determining how the second and fourth derivatives of H propagate under composition with another function G, so that $H^2 = G(H^1(\Delta t))$. We assume $G(0) = 0$. We can use the chain rule and the product rule to derive:

$$\ddot{H}^2(0) = \dot{G}(0)\ddot{H}^1(0) \tag{24}$$

$$\ddddot{H}^2(0) = 3\ddot{G}(0)\ddot{H}^1(0)^2 + \dot{G}(0)\ddddot{H}^1(0). \tag{25}$$

B.3 Evolution equations for curvature and length

We now apply the above iterations with $H^1(\theta) = 1 - c(\theta)$ and $G(c) = 1 - \frac{1}{q^*}\mathcal{C}(1 - c,\ q^*,\ q^* \,|\, \sigma_w, \sigma_b)$. Clearly, $G(0) = 0$ the symmetric H^1 obeys $H^1(0) = 0$, satisfying the above iterations of second and fourth derivatives. Taking into account these derivative recursions, using the expressions for κ and g^E in terms of derivatives of $c(\theta)$ at 0, and carefully accounting for factors of q^* and N, we obtain the final evolution equations that have been successfully tested against experiments:

$$\bar{g}^{E,l} = \chi_1 \bar{g}^{E,l-1} \tag{26}$$

$$(\bar{\kappa}^l)^2 = 3\frac{\chi_2}{\chi_1^2} + \frac{1}{\chi_1}(\bar{\kappa}^{l-1})^2, \tag{27}$$

where χ_1 is the stretch factor defined in (19) and χ_2 is defined analogously as

$$\chi_2 = \sigma_w^2 \int \mathcal{D}z \left[\phi'' \left(\sqrt{q^*} z \right) \right]^2. \tag{28}$$

χ_2 is closely related to the second derivative of the correlation coefficient map in (16) at $c_{12}^{l-1} = 1$. Indeed this second derivative is $\chi_2 q^*$.

C Upper bounds on the complexity of shallow neural representations

Consider a shallow network with 1 hidden layer $\mathbf{x}^1$ and one input layer $\mathbf{x}^0$, so that $\mathbf{x}^1 = \phi(\mathbf{W}^1 \mathbf{x}^0) + \mathbf{b}$. The network can compute functions through a linear readout of the hidden layer $\mathbf{x}^1$. We are interested in how complex these neural representations can get, with one layer of synaptic weights and nonlinearities, as a function the number of hidden units N_1. In particular, we are interested in how the length and curvature of an input manifold $\mathbf{x}^0(\theta)$ changes as it propagates to become $\mathbf{x}^1(\theta)$ in the hidden layer. We would like to upper bound the maximal achievable length and curvature over all possible choices of $\mathbf{W}^1$ and $\mathbf{b}$.

C.1 Upper bound on Euclidean length

Here, we derive such an upper bound on the Euclidean length for a very general class of nonlinearities $\phi(h)$. We simply assume that (1) $\phi(h)$ is monotonically non-decreasing (so that $\phi'(h) \geq 0 \forall h$) and (2) has with bounded dynamic range R, i.e. $\max_h \phi(h) - \min_h \phi(h) = R$. The Euclidean length in hidden space is

$$\mathcal{L}^E \quad = \quad \int d\theta \sqrt{\sum_{i=1}^{N_1} (\partial_\theta \mathbf{x}_i^1(\theta))^2} \quad \leq \quad \sum_{i=1}^{N_1} \int d\theta \left| \partial_\theta \mathbf{x}_i^1(\theta) \right|, \tag{29}$$

where the inequality follows from the triangle inequality. Now suppose that for any i, $\partial_\theta \mathbf{x}_i^1(\theta)$ never changes sign across θ. Furthermore, assume that θ ranges from 0 to Θ. Then

$$\int_0^\Theta d\theta \left| \partial_\theta \mathbf{x}_i^1(\theta) \right| = \mathbf{x}_i^1(\Theta) - \mathbf{x}_i^1(0) \leq \left(\max_h \phi(h) - \min_h \phi(h) \right) = R. \tag{30}$$

More generally, let r_1 denote the maximal number of times that any one neuron has a change in sign of the derivative $\partial_\theta \mathbf{x}_i^1(\theta)$ across θ. Then applying the above argument to each segment of constant sign yields

$$\int_0^\Theta d\theta \left| \partial_\theta \mathbf{x}_i^1(\theta) \right| \leq (1 + r_1)R. \tag{31}$$

Now how many times can $\partial_\theta \mathbf{x}_i^1(\theta)$ change sign? Since $\partial_\theta \mathbf{x}_i^1(\theta) = \phi'(\mathbf{h}_i)\, \partial_\theta \mathbf{h}_i$, where $\partial_\theta \mathbf{h}_i(\theta) = [\mathbf{W}^l \partial_\theta \mathbf{x}^0(\theta)]_i$, and $\phi(\mathbf{h}_i)$ is monotonically increasing, the number of times $\partial_\theta \mathbf{x}_i^1(\theta)$ changes sign equals the number of times the input $\partial_\theta \mathbf{h}_i(\theta)$ changes sign. In turn, suppose s_0 is the maximal number of times any one dimensional projection of the derivative vector $\partial_\theta \mathbf{x}^0(\theta)$ changes sign across θ. Then the number of times the sign of $\partial_\theta \mathbf{h}_i(\theta)$ changes for any i cannot exceed s_0 because $\mathbf{h}_i$ is a linear projection of $\mathbf{x}^0$. Together this implies $r_1 \leq s_0$. We have thus proven:

$$\mathcal{L}^E \leq N_1(1 + s_0)R. \tag{32}$$

D   Simulation details

All neural network simulations were implemented in Keras and Theano. For all simulations (except Figure 5C), we used inputs and hidden layers with a width of 1,000 and tanh activations. We found that our results were mostly insensitive to width, but using larger widths decreased the fluctuations in the averaged quantities. Simulation error bars are all standard deviations, with the variance computed across the different inputs, $h^1(\theta)$. If not mentioned, the weights in the network are initialized in the chaotic regime with $\sigma_b = 0.3$, $\sigma_w = 4.0$.

Computing $\kappa(\theta)$ requires the computation of the velocity and acceleration vectors, corresponding to the first and second derivatives of the neural network $h^l(\theta)$ with respect to θ. As θ is always one-dimensional, we can greatly speed up these computations by using forward-mode auto-differentiation, evaluating the Jacobian and Hessian in a feedforward manner. We implemented this using the R-op in Theano.

D.1   Details on Figure 4G: backpropagating curvature

To identify the curvature of the decision boundary, we first had to identify points that lied along the decision boundary. We randomly initialized data points and then optimized $G(x^D(x^l))^2$ with respect to the input x using Adam. This yields a set of inputs x^l where we compute the Jacobian and Hessian of $G(x^D(x^l))$ to evaluate principal curvatures.

D.2   Details on Figure 5C-D: evaluating expressivity

To evaluate the set of functions reachable by a network, we first parameterized function space using a Fourier basis up to a particular maximum frequency, ω_{max} on a sampled set of one dimensional inputs parameterized by θ. We then took the output activations of each neural network and linearly regressed the output activations onto each Fourier basis. For each basis, we computed the angle between the

predicted basis vector and the true basis vector. These are the quantities that appear in Figure 5C-D. Given any function with bounded frequency, we can represent it in this Fourier basis, and decompose the error in the prediction of the function into the error in prediction of each Fourier component. Thus error in the predicting the Fourier basis is a reasonable proxy for error in prediction of functions with bounded frequency.

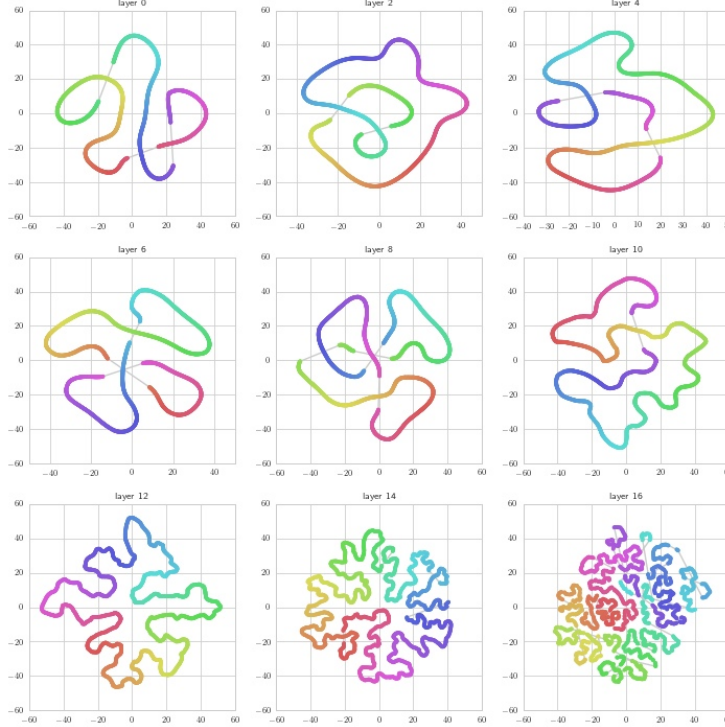## E   Additional visualization of hidden actions

Figure 6: Two-dimensional t-SNE embeddings of hidden activations of a network in the chaotic regime. These can be compared to the PCA visualizations in Fig. 3A of the main paper. Our results reveal that the decreasing radii of curvature of both the t-SNE plots and PCA plots with depth is an illusion, or artifact of attempting to fit a fundamentally high dimensional curve into a low dimensional space. Instead, our curvature evolution equations reveal the geometric nature of curve without resorting to dimensionality reduction. As the depth increases, the radius of curvature stays *constant* while the length grows *exponentially*. As a result, the curve makes a number of turns that is exponentially large in depth, where each turn occurs at a depth independent acceleration. In this manner, the curve fills up the high dimensional hidden representation space. The t-SNE plot faithfully reflects the number of turns, but not their radius of curvature.

F   A view from the function space perspective

We have shown above that for a fixed set of weights and biases in the chaotic regime, the internal representation $\mathbf{h}^l(\mathbf{x}^0)$ at large depth l, rapidly de-correlates from itself as the input $\mathbf{x}^0$ changes (see e.g. Fig. 3B in the main paper). Here we ask a dual question: for a fixed input manifold, how does a deep network move in a function space over this manifold as the weights in a single layer change? Consider for example, a random one parameter family of deep networks parameterized by $\Delta \in [-1, 1]$. In this family, we assume that the bias vectors $\mathbf{b}^l$ in each layer are chosen as i.i.d. random Gaussian vectors with zero mean and variance σ_b^2, independent of Δ. Moreover, we assume the weight matrix $\mathbf{W}^l$ has elements that are drawn i.i.d. from zero mean Gaussians with variance σ_w^2, independent of Δ for all layers except $l = 2$. The only dependence on Δ in this family of networks

originates in the weights in layer $l = 2$, chosen as

$$\mathbf{W}^l(\Delta) = \sqrt{1 - |\Delta|}\,\mathbf{W} + \sqrt{|\Delta|}\,\mathbf{dW}. \tag{33}$$

Here both a base matrix $\mathbf{W}$ and a perturbation matrix $\mathbf{dW}$ have matrix elements that are zero mean i.i.d. Gaussians with variance σ_w^2. Each matrix element of $\mathbf{W}^2(\Delta)$ thus also has variance σ_w^2 just like all the other layers. In turn, this family of networks induces a family of functions $\mathbf{h}^D(\mathbf{h}^1, \Delta)$. For simplicity, we restrict these functions to a simple input manifold, the circle,

$$\mathbf{h}^1(\theta) = \sqrt{N_1 q^*}\left[\mathbf{u}^0\cos(\theta) + \mathbf{u}^1\sin(\theta)\right], \tag{34}$$

as considered previously. This circle is at the fixed point radius $q^*(\sigma_w, \sigma_b)$, and the family of networks induces a family of functions from the circle to the hidden representation space in layer l, namely $\mathbb{R}^{N_l}$. We denote these functions by $\mathbf{h}^l(\theta, \Delta)$. How similar are these functions as Δ changes? This can be quantified through the correlation in function space

$$Q^l(\Delta_1, \Delta_2) \equiv \int \frac{d\theta}{2\pi}\frac{1}{N_D}\sum_{i=1}^{N_D}\mathbf{h}_i^l(\theta, \Delta_1)\mathbf{h}_i^l(\theta, \Delta_2), \tag{35}$$

and the associated correlation coefficient,

$$C^l(\Delta) = \frac{Q^l(0, \Delta)}{\sqrt{Q^l(0, 0)Q^l(\Delta, \Delta)}}. \tag{36}$$

Because of our restriction to an input circle at the fixed point radius, $Q^l(0, 0) = Q^l(\Delta, \Delta) = q^*$ for all l and Δ in the large width limit. By using logic similar to the derivation of (15), we can derive a recursion relation for the function space correlation $Q^l(0, \Delta)$:

$$Q^l(0, \Delta) = \sigma_w^2\int \mathcal{D}z_1\,\mathcal{D}z_2\,\phi\left(u_1\right)\phi\left(u_2\right) + \sigma_b^2, \quad l = 3, \ldots, D \tag{37}$$

$$Q^l(0, \Delta) = \sqrt{1 - |\Delta|}\sigma_w^2\int \mathcal{D}z_1\,\mathcal{D}z_2\,\phi\left(u_1\right)\phi\left(u_2\right) + \sigma_b^2, \quad l = 2,$$

$$u_1 = \sqrt{q^*}z_1, \qquad u_2 = \sqrt{q^*}\left[C^{l-1}(\Delta)z_1 + \sqrt{1 - (C^{l-1}(\Delta))^2}z_2\right],$$

where $C^l(\Delta) = Q^l(0, \Delta)/q^*$. The initial condition for this recursion is $C^1(\Delta) = 1$, since the family of functions in the first layer of inputs is independent of Δ. Now, the difference in weights at a nonzero Δ reduces the function space correlation to $C^2(\Delta) < 1$. At this point, the representation in $\mathbf{h}^2$ is different for the two networks at parameter values 0 and Δ. Moreover, in the chaotic regime, this difference will amplify due to the similarity between the function space evolution equation in (37) and the evolution equation for the similarity of two points in (15). In essence, just as two points in the input exponentially separate as they propagate through a single network in the chaotic regime, a pair of different functions separate when computed in the final layer. Thus a small perturbation in the weights into layer 2 can yield a very large change in the space of functions from the input manifold to layer D. Moreover, as Δ varies from -1 to 1, the function $\mathbf{h}^D(\theta, \Delta)$ roughly undergoes a random walk in function space whose autocorrelation length decreases exponentially with depth D. This weight chaos, or a sensitive dependence of the function computed by a deep network with respect to weight changes far from the final layer, is another manifestation of deep neural expressivity. Our companion paper [16] further explores the expressivity of deep random networks in function space and also finds an exponential growth in expressivity with depth.