Dynamical Isometry is Achieved in Residual Networks in a Universal Way for any Activation Function

Wojciech Tarnowski,[1, *] Piotr Warchoł,[1, †] Stanisław Jastrzębski,[2, ‡] Jacek Tabor,[2, §] and Maciej A. Nowak[3, ¶]

[1]*M. Smoluchowski Institute of Physics, Jagiellonian University, PL–30–348 Kraków, Poland*
[2]*Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland*
[3]*M. Smoluchowski Institute of Physics and Mark Kac Complex Systems Research Center, Jagiellonian University, PL–30–348 Kraków, Poland*
(Dated: March 5, 2019)

We demonstrate that in residual neural networks (ResNets) dynamical isometry is achievable irrespective of the activation function used. We do that by deriving, with the help of Free Probability and Random Matrix Theories, a universal formula for the spectral density of the input-output Jacobian at initialization, in the large network width and depth limit. The resulting singular value spectrum depends on a single parameter, which we calculate for a variety of popular activation functions, by analyzing the signal propagation in the artificial neural network. We corroborate our results with numerical simulations of both random matrices and ResNets applied to the CIFAR-10 classification problem. Moreover, we study consequences of this universal behavior for the initial and late phases of the learning processes. We conclude by drawing attention to the simple fact, that initialization acts as a confounding factor between the choice of activation function and the rate of learning. We propose that in ResNets this can be resolved based on our results by ensuring the same level of dynamical isometry at initialization.

I. INTRODUCTION

Deep Learning has achieved unparalleled success in fields such as object detection and recognition, language translation, and speech recognition [16]. At the same time, models achieving these state-of-the-art results are increasingly deep and complex [3], which often leads to optimization challenges such as vanishing gradients. Many solutions to this problem have been proposed. In particular, Residual Neural Networks remedy this to some extent [10, 35] by using skip connections in the network architecture, which improve gradient flow. As a result, Residual Neural Networks outmatched other competing models in the 2015 ILSVRC and COCO competitions. Yet another approach towards solving this problem is to tailor fit the networks weight initialization to facilitate training, for example by ensuring dynamical isometry [28]. In this latter case, the insights are based on an analysis of the statistical properties of information propagation in the network and a study of the full singular spectrum of a particular matrix, namely the input-output Jacobian, via the techniques of Free Probability and Random Matrix Theories (FPT & RMT). This perspective has recently led to successfully training a 10000 layer vanilla convolutional neural network [38].

RMT is a versatile tool that, since its inception, saw a substantial share of applications, from the earliest in nuclear physics [37] to the latest in game theory [4] (see [1] for some of the use cases discovered in the mean time). It is thus not surprising that it found its way to be used to understand artificial neural networks. In particular, to study their loss surface [6, 27], the associated Gram matrix [19, 30] and in the case of single layer networks, their dynamics [17]. Our main contribution is extending the theoretical analysis of [28, 29, 33] to residual networks. In particular, we find that residual networks can achieve dynamical isometry for many different activation functions provided that the variance of weight initialization scale is inversely proportional to the number of skip-connections. This is in contrast to feedforward networks, where orthogonal weights and antisymmetric sigmoidal activation functions (like tanh) are required. These theoretical results are supported by an empirical investigation on the popular CIFAR-10 benchmark.

A. Related work

The framework of dynamical mean field theory, we will apply to study signal propagation in neural networks, was first used in this context in [31]. There, the authors showed the existence of an order-to-chaos expressivity transition for deep feedforward neural networks with random initial weights, on the plane spanned by the variances of the network weights and biases. This in

---

[*]Electronic address: wojciech.tarnowski@uj.edu.pl
[†]Electronic address: piotr.warchol@uj.edu.pl
[‡]Electronic address: stanislaw.jastrzebski@uj.edu.pl
[§]Electronic address: jcktbr@gmail.com
[¶]Electronic address: maciej.a.nowak@uj.edu.pl

turn led to the insight of [33], that arbitrary deep networks can be trained as long as they are close to the criticality associated with that transition. The techniques developed in these works, together with methods of FPT and RMT, allowed, for the first time, to analytically compute the singular value distribution of the input-output Jacobian of a deep feedforward network with nonlinear activation function and at criticality [28]. Finally, [29] showed, that for feedforward neural networks, in their large depth limit and at a special point of the above mentioned critical line, the singular spectrum of the Jacobian is given by a universal distribution depending on the form of the activation function used. In particular they distinguish the *Bernoulli* and the *smooth* universality classes corresponding to piecewise linear and some nonlinear activation functions. In fact, in this paper, we take the approach of that last work and apply it to fully connected residual neural networks. We find a single universality class for this architecture.

Let us also mention some recent, important developments in the area of residual neural network initialization. One of the earlier developments, is the introduction of layer-sequential unit-variance (LSUV) initialization [22]. The two step process involved normalizing the outputs of the neurons on the first forward run and showed promising results. In another, very relevant paper [34], analyzing the signal propagation in a similar manner to that mentioned in the paragraph above, shows for ResNets, with piecewise linear, symmetric as well as ReLU activation functions, that the proper variance for network weight initialization is of order $\frac{1}{NL}$, where L is the number of layers and N the number of neurons in each layer. A similar conclusion is reached by [2]. We corroborate this result with our analysis. Another contribution shows that adding skip connections to the network, eliminates the critical behavior described above [39]. Finally, the importance of initialization in ResNets is shown by [40], where it is demonstrated that initializing to a zero function enables training state of the art residual networks without the use of batch normalization [11]. Note that ResNet with this initialization achieves in fact an ideal isometry.

When finishing this manuscript, we have learned of a recent paper tackling the same problem of ResNets initialization by studying the singular spectrum properties of the Jacobian with the tools of Free Probability. While the analysis of [18] is related, it is crucial to note that the authors do not observe the universal character of the singular spectrum - the main result of our paper, and treat only piecewise linear activation functions. It is also worth mentioning that, similar to us, they rediscover the importance of $\frac{1}{LN}$ scaling of [2] and [34].

B. Our results

Our contributions are the following. We show that the singular spectrum of the input-output Jacobian, in the the networks large width and depth limit, is given by a universal formula - with the dependence on the type of activation function encapsulated in a single parameter. Furthermore, we calculate the layer dependent statistical properties of the pre-activations for a variety of activation functions. All together, this gives the associated singular spectra of the Jacobian, which we compare with random matrix and artificial neural network numerical simulations corroborating our theoretical results. The singular values of the input-output Jacobian concentrate around 1 for a wide range of parameters, which shows that fine-tuning the initialization is not required for achieving dynamical isometry in ResNets. Even though the final results of the theoretical calculations are derived in the limit $L, N \to \infty$, the numerical experiments match them already for $L = 10$ (with $N = 500$). As a practical application of our work and the universality property it uncovers, we propose a framework for setting up weight initialization in experiments with residual neural networks.

C. Structure of the paper

We follow this introductory section by defining the model of ResNets we will work with and with a short note on the relevance of the input-output Jacobian. Then, in subsection III A, we derive the equation governing the Green's function and hence the spectrum of the Jacobian, which depends on a single parameter, which we denote by c. Proceeding is the analysis of the propagation of the information in the network via an analysis of the probability density function describing the pre-activations across the layers at network initialization. This allows us to calculate c for many different activation functions in Appendix C. We close the second section of the paper by revealing the random matrix experiments confirming our results. Sec. IV is devoted to the outcome of associated residual neural network numerical calculations. There, we showcase the resulting, experimental, universal spectrum of the Jacobian and the outcomes of the learning processes. This is followed by a short comment, in Sec. V, on the influence of batch normalization on the presented setup. We close the paper with a short discussion section. Finally, in the rest of the Appendices, we show the results of numerical experiments validating the signal propagation recurrence relations and some baseline (based on using the same weight matrix variances, irrespective of the choice of activation functions) simulations of the learning process.

II. THE MODEL

In this paper, we consider a deep, residual network of L layers of a constant width of N neurons. We follow the typical nomenclature of the literature and therefore the real-valued, synaptic matrix for the l-th layer is denoted by $\boldsymbol{W}^l$, whereas the real-valued bias vectors are $\boldsymbol{b}^l$. The information propagates in this network according to:

$$\boldsymbol{x}^l = \phi(\boldsymbol{h}^l) + a\boldsymbol{x}^{l-1}, \quad \boldsymbol{h}^l = \boldsymbol{W}^l \boldsymbol{x}^{l-1} + \boldsymbol{b}^l, \tag{1}$$

where $\boldsymbol{h}^l$ and $\boldsymbol{x}^l$ are pre- and post-activations respectively and ϕ is the activation function itself, acting entry-wise on the vector of pre-activations. We have introduced the parameter a to track the influence of skip connections in the calculations, however we do not study its influence on the Jacobian's spectrum or learning in general. By $\boldsymbol{x}^0$ we denote the input of the network and by $\boldsymbol{x}^L$ its output. Our primary interest will lay in exploring the singular value spectral properties of the input-output Jacobian:

$$J_{ik} = \frac{\partial x_i^L}{\partial x_k^0}, \tag{2}$$

known to be useful in studying initialization schemes of neural networks at least since the work of [8]. It in particular holds the information on the severity of the exploding gradients problem.

A. Relevance of the input-output Jacobian

To understand why we are interested in the Jacobian, consider the neural network adjusting its weights during the learning process. In a simplified, example by example scenario, this happens according to

$$\Delta W_{ij}^l = -\eta \frac{\partial E(\boldsymbol{x}^L, \boldsymbol{y})}{\partial W_{ij}^l}, \tag{3}$$

where $E(\boldsymbol{x}^L, \boldsymbol{y})$ is the error function depending on $\boldsymbol{x}^L$ - the output of the network, $\boldsymbol{y}$ - the correct output value associated with that example and, implicitly through $\boldsymbol{x}^L$, on the parameters of the model, namely the weights and biases. Here, for simplicity, we consider only the adjustments of the weights - an analogous reasoning applies to the biases. η is the learning rate. By use of the chain rule we can rewrite this as:

$$\Delta W_{ij}^l = -\eta \sum_{k,t} \frac{\partial x_t^l}{\partial W_{ij}^l} \frac{\partial x_k^L}{\partial x_t^l} \frac{\partial E(\boldsymbol{x}^L, \boldsymbol{y})}{\partial x_k^L}, \tag{4}$$

For the learning process to be stable, all three terms need to be bounded. Out of those, the middle one can become problematic if a poor choice of the initialization scheme is made. We can rewrite it as:

$$\frac{\partial x_k^L}{\partial x_t^l} = \left[\prod_{i=l+1}^{L} \left(\boldsymbol{D}^i \boldsymbol{W}^i + \mathbf{1}a \right) \right]_{kt} \tag{5}$$

and see the larger the difference between L and l, the more terms we have in the product, and (in general) the less control there is over its behavior. Here $\mathbf{1}$ is an identity matrix and, $\boldsymbol{D}^l$ is a diagonal matrix such that $D_{ij}^l = \phi'(h_i^l)\delta_{ij}$. Indeed, it was proposed by [8], that learning in deep feed-forward neural networks can be improved by keeping the mean singular value of the Jacobian associated with layer i (in our setup $\boldsymbol{J}^i = \boldsymbol{D}^i \boldsymbol{W}^i + \mathbf{1}a$), close to 1 for all i's. It is also important for the dynamics of learning to be driven by data, not by the random initialization of the network. The latter may take place if the Jacobian to the l-th layer possesses very large singular values which dominate the learning or very small singular values suppressing it. In the optimal case all singular values should be concentrated around 1 regardless of how deep is the considered layer. One therefore examines the case of $l = 0$, namely $\partial x_k^L / \partial x_t^0$ - the input-output Jacobian, as the most extreme object of (5). The feature that in the limit of large depth all singular values of $\boldsymbol{J}$ concentrate around 1, irrespective of the depth of the network, was coined as dynamical isometry [32].

Note, that the spectral problem for the full Jacobian

$$\boldsymbol{J} = \prod_{l=1}^{L} \left(\boldsymbol{D}^l \boldsymbol{W}^l + \mathbf{1}a \right) \tag{6}$$

belongs to the class of matrix-valued diffusion processes [9, 13], leading to a complex *eigenvalue* spectrum. We note that the large N limit, spectral properties of (6) with $D = 1$ (deep linear networks), and different symmetry classes of W, was derived already by [9]. Due to non-normality of the Jacobian, singular values cannot be easily related to eigenvalues. Therefore we follow [28, 29] and tackle the full *singular spectrum* of the Jacobian (or equivalently the eigenvalue spectrum of JJ^T), extending these works to the case of the Residual Neural Network model.

III. SPECTRAL PROPERTIES OF THE JACOBIAN

A. Spectral analysis

Free Probability Theory, or Free Random Variable (FRV) Theory [36], is a powerful tool for the spectral analysis of random matrices in the limit of their large size. It is a counterpart of the classical Probability Theory for the case of non-commuting observables. For a pedagogical introduction to the subject, see [21] - here we start by laying out the basics useful in the derivations of this subsection. The fundamental objects of the theory are the Green's functions (a.k.a. Stieltjes transforms in mathematical literature):

$$G_H(z) = \left\langle \frac{1}{N} \text{Tr} \, (z\mathbf{1} - H)^{-1} \right\rangle = \int_{-\infty}^{\infty} \frac{\rho_H(\lambda) d\lambda}{z - \lambda}, \tag{7}$$

which generate spectral moments and where the subscript H indicates, that his formulation is proper for self-adjoint matrices. The eigenvalue density can be recovered via the Sochocki-Plemelj formula

$$\rho_H(x) = -\frac{1}{\pi} \lim_{\epsilon \to 0} G_H(x + i\epsilon). \tag{8}$$

The associated free cumulants are generated by the so-called R-transform, which plays the role of the logarithm of the characteristic function in the classical probability. By this correspondence, the R-transform is additive under addition, i.e. $R_{X+Y}(z) = R_X(z) + R_Y(z)$ for mutually free, but non-commuting random ensembles X and Y. Moreover, it is related to G via the functional equations

$$G\left(R(z) + \frac{1}{z}\right) = z, \qquad R(G(z)) + \frac{1}{G(z)} = z. \tag{9}$$

On the other hand, the so-called S-transform facilitates calculations of the spectra of products of random matrices, as it satisfies $S_{AB}(z) = S_A(z)S_B(z)$, provided A and B are mutually free and at least one is positive definite. If additionally, the ensemble has a finite mean, the S-transform can be easily obtained from the R-transform, and vice versa, through a pair of the following, mutually inverse maps $z = yS(y)$ *and* $y = zR(z)$. Explicitly:

$$S(zR(z)) = \frac{1}{R(z)}, \quad R(zS(z)) = \frac{1}{S(z)}. \tag{10}$$

Denoting now J_L the Jacobian across L layers and $Y_l = (a\mathbf{1} + D^l W^l)$, one can write the recursion relation $J_L J_L^T = Y_L J_{L-1} J_{L-1}^T Y_L$. The latter matrix is isospectral to $Y_L^T Y_L J_{L-1} J_{L-1}^T$, which leads to the equation for the S-transform $S_{J_L J_L^T}(z) = S_{Y_L^T Y_L}(z) S_{J_{L-1} J_{L-1}^T}(z)$. Proceeding inductively, we arrive at

$$S_{JJ^T}(z) = \prod_{l=1}^{L} S_{Y_l Y_l^T}(z). \tag{11}$$

To find the S-transform of the single layer Jacobian, we will first consider its Green's function

$$G(z) = \left\langle \frac{1}{N} \text{Tr}(z\mathbf{1} - Y_l Y_l^T)^{-1} \right\rangle, \tag{12}$$

with the averaging over the ensemble of weight matrices W^l. To facilitate the study of G, in particular to cope with YY^T, one linearizes the problem by introducing matrices of size $2N \times 2N$

$$\mathcal{Z} := \begin{pmatrix} -a & 1 \\ z & -a \end{pmatrix}, \qquad \mathcal{X} := \begin{pmatrix} X & 0 \\ 0 & X^T \end{pmatrix}, \tag{13}$$

with $\boldsymbol{X} = \boldsymbol{D}^l \boldsymbol{W}^l$. Another crucial ingredient is the block trace operation (bTr), which is the trace applied to each $N \times N$ block. The generalized Green's function is defined as a block trace of the generalized resolvent $(\mathcal{Z} \otimes \boldsymbol{1} - \mathcal{X})^{-1}$

$$\mathcal{G} := \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} = \left\langle \frac{1}{N} \mathrm{bTr} \begin{pmatrix} -a - \boldsymbol{X} & 1 \\ z & -a - \boldsymbol{X}^T \end{pmatrix}^{-1} \right\rangle. \tag{14}$$

Remarkably, the Green's function of $\boldsymbol{Y}\boldsymbol{Y}^T$ is the G_{12} entry of the generalized Green's function. This construction is a slight modification of the quaternionization approach to large non-Hermitian matrices developed by [12], therefore we adapt these concepts here for calculations in the large width limit of the network. Furthermore, the generalized Green's function (14) is given implicitly by the solution of the Schwinger-Dyson equation

$$\mathcal{G}(\mathcal{Z}) = (\mathcal{Z} - \mathcal{R}(\mathcal{G}(\mathcal{Z})))^{-1}. \tag{15}$$

Here $\mathcal{R}$ is the generalized R-transform of FRV theory. This construction is a generalization of standard FRV tools to the matrix-valued functions. In particular, (15) is such a generalization of (9) to 2×2 matrices.

To study two common weight initializations, Gaussian and scaled orthogonal, on the same footing, we assume that W belongs to the class of biunitarily invariant random matrices, i.e. its pdf is invariant under multiplication by two orthogonal matrices, $P(\boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^T) = P(\boldsymbol{W})$ for $\boldsymbol{U}, \boldsymbol{V} \in O(N)$. In the large N limit these matrices are known in free probability as R-diagonal operators [24]. A product of R-diagonal operator with an arbitrary operator remains R-diagonal [25], thus the matrix $\boldsymbol{X}$ is R-diagonal too.

The generalized $\mathcal{R}$-transform of R-diagonal operators takes a remarkably simple form [26]

$$\mathcal{R}(\mathcal{G}) = A(G_{12}G_{21}) \begin{pmatrix} 0 & G_{12} \\ G_{21} & 0 \end{pmatrix}. \tag{16}$$

Here, $A(x) = \sum_{k=1}^{\infty} c_{2k} x^{k-1}$ is the determining sequence, which generates cumulants c_{2k}, it is a Taylor expansion of $A(x)$ at 0. For the later use we mention a simple relation for the determining sequence of a scaled matrix $A_{aX}(z) = a^2 A_X(a^2 z)$, which generalizes the Hermitian case $G_{aH}(z) = \frac{1}{a} G_H(\frac{z}{a})$ or, equivalently, $R_{aH}(z) = aR_H(az)$.

We derive the equation for the Green's function (with $G(z) = G_{12}$) by substituting the $\mathcal{R}$-transform (16) into (15) and eliminating irrelevant variables. It thus reads:

$$G = \frac{GA(zG^2) - 1}{a^2 - z(1 - GA(zG^2))^2}, \tag{17}$$

where for clarity we omitted the argument of the Green's function. In the next step we substitute $z \to R(z) + \frac{1}{z}$ and use (9) to obtain

$$z = \frac{zA(z^2R + z) - 1}{a^2 - (R + \frac{1}{z})(1 - zA(z^2R + z))^2}. \tag{18}$$

Then, we substitute $z \to zS(z)$ and use (10), which leads us to

$$1 = \frac{zSA(z(z+1)S) - 1}{a^2zS - (z+1)(1 - zSA(z(z+1)S))^2}. \tag{19}$$

This equation is exact. To incorporate the additional scaling of weights variances by $1/L$ in our considerations, as proposed by [34] and [2], we rescale $\boldsymbol{X} \to \boldsymbol{X}/\sqrt{L}$ and since we are interested in deep networks, we keep only the leading term in $1/L$ (see also [13]). This leads to $A(z(z+1)S) \to \frac{1}{L}A\left(\frac{1}{L}z(z+1)S\right) = \frac{c_2}{L} + O\left(\frac{1}{L^2}\right)$, which simplifies (19) to a quadratic equation for S. Choosing the appropriate branch of the solution, we see that

$$S_{Y_lY_l^T}(z) = \frac{1}{a^2}\left(1 - \frac{c_2^l}{a^2L}(1 + 2z) + O\left(\frac{1}{L^2}\right)\right). \tag{20}$$

Here

$$c_2^l = \left\langle \frac{1}{N} \mathrm{Tr} \boldsymbol{W}^l \boldsymbol{D}^l \boldsymbol{D}^l (\boldsymbol{W}^l)^T \right\rangle = \frac{\sigma_w^2}{N} \sum_i^N \left(\phi'(h_i^l)\right)^2 \tag{21}$$

is the squared spectral radius of the matrix $\boldsymbol{D}^l \boldsymbol{W}^l$. In general, c_2^l can vary across the depth of the network due to non-constant variance of preactivations. Assuming that this variability is bounded, we can consider the logarithm of (11) and write:

$$\ln S_{JJ^T}(z) = -2L \ln a - \frac{(1 + 2z)}{a^2}c, \tag{22}$$

where we defined the effective cumulant $c = \frac{1}{L}\sum_{l=1}^{L} c_2^l$ and used $\ln(1 + x) \approx x$. This allows us to deduce the form of the S-transform, assuming that a does not scale with L

$$S_{JJ^T}(z) = \frac{1}{a^{2L}} e^{-\frac{c}{a^2}(1+2z)}. \tag{23}$$

Substituting $z \to zR(z)$ and using (10), we obtain

$$a^{2L} = R(z) \exp\left[-\frac{c}{a^2}(1 + 2zR(z))\right]. \tag{24}$$

Then, we substitute $z \to G(z)$ and use (9) to finally get

$$a^{2L}G(z) = (zG(z) - 1)e^{\frac{c}{a^2}(1-2zG(z))}, \tag{25}$$

an equation for the Green's function characterizing the square singular values of the Jacobian, which can be solved numerically. We do that for a range of different activation functions and present the results with numerical simulations to corroborate them in Fig. 1.

We close this section with a remark that the above analysis is not restricted only to the model (1), but analogous reasoning can be performed for networks in which skip connections bypass more than one fully connected block. The qualitative results remain unaltered provided that L is replaced by the number of skip connections.

B. Signal propagation

The formulas we have derived until now were given in terms of a single parameter c, which is the squared derivative of the activation function averaged within each layer and across the depth of the network. Thus, we now need to address the behavior of preactivations. In the proceeding paragraph, we closely follow a similar derivation done in [33], for fully connected feed forward networks.

For the simplicity of our arguments, we consider here W_{ij}^l and b_i^l as independent identically distributed (iid) Gaussian random variables with 0 mean and variances $\frac{(\sigma_W)^2}{LN}$ and $(\sigma_b)^2$, respectively. Here, $(\sigma_W)^2$ is of order one, and the additional scaling is meant to reflect those introduced in the previous paragraphs. At the end of this section we provide an argument that the same results hold for scaled orthogonal matrices.

In this subsection, we will denote the averaging over variables W_{ij}^l and b_i^l, at a given layer l, by $\langle\cdot\rangle_{wbl}$. By $\langle u\rangle_l$ we denote the sample average, of some variable u in the l-th layer: $\langle u\rangle_l \equiv \frac{1}{N}\sum_{i=1}^{N} u_i^l$. Note that the width ($N$) is independent of the layer number, however the derivation can be easily generalized to the opposite case, when the architecture is more complicated. Unless stated otherwise explicitly, all integrals are calculated over the real line.

We are interested in the distribution of h_i^l in our model, depending on the input vectors and the probability distributions of W_{ij}^l and b_i^l. If we assume they are normal (as can be argued using the Central Limit Theorem), we just need the first two moments. It is clear that $\langle h_i^l\rangle_{wbl} = 0$. Furthermore, we assume ergodicity, i.e. that averaging some quantity over a layer of neurons is equivalent to averaging this quantity for one neuron over an ensemble of neural networks with random initializations. We assume this is true for h_i, x_i, W_{ij} and b_i. Thus, we can say that $\langle h\rangle_l = 0$ and moreover, as we work in the limit of wide networks, $\langle f(h)\rangle_l$ (where f is some function of h_l) can be replaced with an averaging over a normal distribution of variance $q^l \equiv \langle (h^l)^2\rangle_{wbl}$. This is the crux of the dynamical mean field theory approach [31] for feed-forward neural networks. We have in particular:

$$c_2^l = \sigma_W^2 \langle (\phi'(h))^2\rangle_l = \sigma_W^2 \int \mathcal{D}z \phi'^2\left(\sqrt{q^l}z\right), \tag{26}$$

where $\mathcal{D}z = \exp(-z^2/2)dz/\sqrt{2\pi}$. To calculate the effective cumulant, we need to know how the variance of the distribution of the preactivations changes as the input information propagates across consecutive layers of the network. It is shown in Appendix B, that q^l satisfy the recurrence equation

$$q^{l+1} = a^2 q^l + \left(1 - a^2\right)\sigma_b^2 + \frac{(\sigma_W)^2}{L}\int \mathcal{D}z\phi^2\left(\sqrt{q^l}z\right) + 2\frac{(\sigma_W)^2}{L}\left[\sum_{k=1}^{l-1} a^k \int \mathcal{D}z\phi\left(\sqrt{q^{l-k}}z\right)\right]\int \mathcal{D}z\phi\left(\sqrt{q^l}z\right), \tag{27}$$

with the initial condition $q^1 = \sigma_b^2 + \frac{\sigma_W^2}{L}$.

We remark here that the above reasoning concerning signal propagation holds also when the weights are scaled orthogonal matrices, i.e. $WW^T = \frac{\sigma_W^2}{L}\mathbf{1}$. In such a case $\langle W_{ij}\rangle = 0$ and $\langle W_{ij}W_{kl}\rangle = \frac{\sigma_W^2}{NL}\delta_{ik}\delta_{jl}$ [7] and the entries of W can be approximated as independent Gaussians [5].
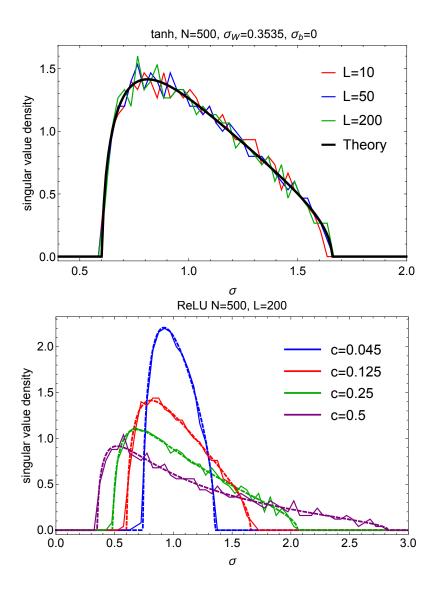
FIG. 1: (Top) density of singular values of the input-output Jacobian for the residual network with tanh nonlinearity. Note that the asymptotic theoretical result describes remarkably well not very deep ($L = 10$) networks. (Bottom) asymptotic distribution of singular values for various values of parameter c (dashed) juxtaposed with the numerical simulations for ReLU nonlinearity (solid). Note that histograms were calculated from a single random initialisation. The smaller c, the narrower the spectrum and the closer to the ideal isometry.

C.   Random matrix simulations

To thoroughly test the theoretical predictions of Section III, we run numerical simulations using *Mathematica*. The initial condition, input vector x^0 of length $N = 500$, filled with iid Gaussian random variables of zero mean and unit variance, is propagated according to the recurrence (1), for various activation functions. The network weights and biases are generated from normal distribution of zero mean and σ_W^2/NL and σ_b^2 variances, respectively, with $N = 500$ The propagation of variance of pre-activations, post-activations as well as the calculation of the second cumulants c^l for the studied activation functions, across the network, is presented in Appendix D. All numerical simulations corroborate our theoretical results. Here, for clarity and as a generic example, in Fig. 1 (upper), we show the distribution of singular values of the input-output Jacobian (defined in (6)) for the tanh nonlinearity for various network depths. In this example the Jacobian in not independent of the signal propagation, contrary to the case of piecewise linear activation functions. Similarly, in the lower panel of Fig. 1, we showcase the outcome of numerical experiments and the associated, matching, theoretical results for the most popular ReLU activation function, for various initializations resulting in different values of the effective cumulant c.
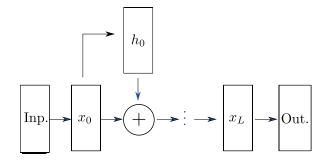
FIG. 2: Residual network architecture used in the paper.

IV. EXPERIMENTS ON IMAGE CLASSIFICATION

The goal of this section is to test our theoretical predictions on real data via the popular CIFAR-10 benchmark [15]. To this end we will use a single representation fully connected residual network, see Fig. 2. This simplified version of the model of [10] does not use (i) multiple stages with different dimension of hidden representation, and (ii) two layers within residual block. We leave study of a more general version of ResNets for future work.

A. Achieving dynamical isometry for any activation function

Perhaps the most interesting prediction of our theory is that ResNets, in contrast to fully connected networks, can achieve dynamical isometry for many different activation functions. We will study this empirically by looking at $\mathbf{J}$, at initialization, for different activation functions and number of residual blocks. Please note that by $\mathbf{J}$ we refer to Jacobian of the output of the last residual block with respect to the input of the first one, see also Fig. 2.

We consider the following popular activation functions: ReLU [23], Tanh, Hard Tanh, Sigmoid, SeLU [14] and Leaky ReLU [20] with the leaking constant 0.05 and 0.25. For each activation function we consider the number of blocks L to be 10 and 20. All weights of the network are initialized from a zero-centered normal distribution whereas biases are initialized to zero. The weights of the residual blocks are initialized using standard deviation $\sigma_W / \sqrt{NL}$, other weights are initialized as by [34]. For the given activation function and the number of blocks L, we set σ_W in such a way that the effective cumulant $c = 0.125$, which ensures the concentration of eigenvalues of the Jacobian around one, and hence dynamical isometry (see Appendix C for more details and Fig. 1 for the shape of the singular value densities).

For each pair of activation function and number of blocks we compute the empirical spectrum of $\mathbf{J}$ at initialization, the results are reported in Fig. 3. Indeed, we observe that upon scaling the initializations standard deviation, in such a way that c is kept constant, the empirical spectrum of $\mathbf{J}$ is independent of the number of residual blocks or the choice of activation functions.

B. Learning dynamics are more similar at universality under dynamical isometry

Our next objective is to investigate whether networks achieving dynamical isometry share similar learning dynamics. While this is outside of the scope of our theoretical investigation, it is inspired by studies such as [28], which demonstrate the importance of dynamical isometry at the initialization for the subsequent optimization.

We consider the same set of experiments as in the previous section, and follow a similar training protocol to [10]. We train for 200 epochs and drop the learning rate by a factor of 10 after epochs 80, 120, and 160. We use batch-size 128 and a starting learning rate of either 10^{-3} or 10^{-4} [41].

First, we look at the learning dynamics on the training set. We can observe that most of the activation functions exhibit similar training accuracy evolution, see Fig. 4 (middle). Using the sigmoid activation, led however to significantly slower optimization. This is due to a faster growth of the variances of post- and pre- activations (which can be observed in Fig. 5), which exacerbates the neuron saturation problem.

Overall our results suggest that the singular spectrum of $\mathbf{J}$ at initialization does not fully determine generalization and training performance. Nonetheless, setting the same effective cumulant for the experiments with different activation functions results in a markedly coinciding behavior of neural networks using activation functions of similar characteristics. This is in contrast to a setup in which the variances of the weight matrix entries are set to be equal. To demonstrate this we run another set of training experiments, this time with all standard deviations $\sigma = 1/\sqrt{LN}$. The plots depicting the full results are relegated to
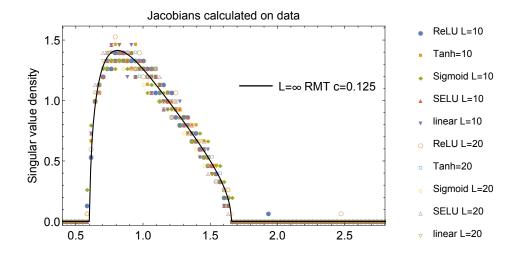
FIG. 3: Singular spectra obtained for various activation functions and depth $L = 10, 20$. The network was fed with examples from CIFAR10 dataset.

Fig. 7 in Appendix E. Here, in Fig. 4 (top) we showcase the training accuracy during the first 40 iterations for these two setups (excluding, for clarity, the networks with the sigmoid activation function). With different effective cumulants, the network learning dynamics, differs among experiments with different activation functions, especially at the beginning of learning.

This suggests that the spectrum of the input-output Jacobian at initialization can be treated as a confounding variable in such experiments. Ensuring that the level of dynamic isometry, and hence the value of the effective cumulant is kept the same, provides the possibility of a more meaningful comparison of the effect of activation functions on learning dynamics.

V. COMMENT ON BATCH NORMALIZATION

A crucial component of practically used ResNets is batch normalization [11]. When it is used on pre-activations, between each layer, the propagation of the information in the network is described by:

$$\boldsymbol{x}^l = \phi(\boldsymbol{y}^l) + a\boldsymbol{x}^{l-1}, \quad \boldsymbol{h}^l = \boldsymbol{W}^l \boldsymbol{x}^{l-1} + \boldsymbol{b}^l, \tag{28}$$

with

$$y_i^l = \frac{h_i^l - \mu_i^l}{\sigma_i^l} \gamma_i^l + \beta_i^l, \tag{29}$$

where μ_i^l is the mean and σ_i^l (regularized with some small ϵ) is the standard deviation of the k'th mini batch inputs i'th coefficient in layer l. γ_i^l and β_i^l are parameters optimized during the learning process. In this case, the formula for the Jacobian reads:

$$\boldsymbol{J} = \prod_{l=1}^{L} \left(\boldsymbol{D}^l \boldsymbol{H}^l \boldsymbol{W}^l + \mathbf{1}a \right), \tag{30}$$

where $\boldsymbol{H}^l$ is a diagonal matrix such that $H_{ij}^l = \delta_{ij} \gamma_i^l / \sigma_i^l$. Therefore, the only difference in the spectral statistics derivation from the previous section, is that (21) becomes

$$c_{2\text{BN}}^l = \sigma_W^2 \left\langle \left(\frac{\gamma^l}{\sigma^l} \phi'(y^l) \right)^2 \right\rangle_l. \tag{31}$$

Thus, the universal, large L limit equation for the Green's function of the Jacobian (25) holds also when batch normalization is included. Again, σ_i^l and y_i^l can be treated as random variables. Unfortunately, the evolution of their probability density functions across the layers is more complicated and beyond the scope of this paper.
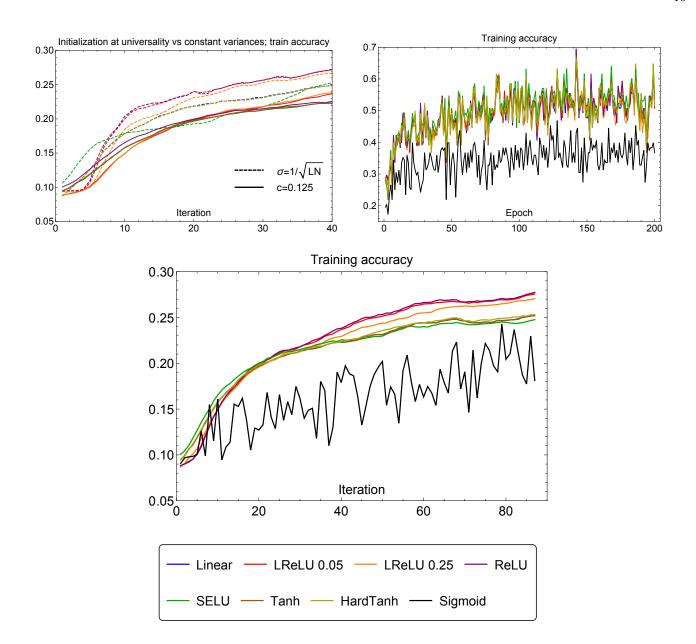
FIG. 4: Training accuracy during first 200 epochs (middle) and first 100 iterations (bottom) of residual networks with various activation functions. The weight initialization was chosen for each activation function in such a way that the effective cumulant is $c = 0.125$. In the top panel, the dynamics with this initialization was juxtaposed with analogous training of networks in which the variance of weights was chosen to be $\frac{1}{LN}$ for all activation functions. We used leaky ReLU with $\alpha = 0.05, 0.25$.

VI. SYNOPSIS AND DISCUSSION

The main focus of this paper was the singular spectrum of the input-output Jacobian of a simple model of residual neural networks. We have shown that in the large network depth limit, it is described by a single, universal equation. This holds irrespective of the activation function used, for biunitarily invariant weight initializations matrices, a set covering Gaussian and scaled orthogonal initialization schemes. The singular value density depends on a single parameter called the effective cumulant, which can be calculated by considering the propagation of information in the network, via a dynamical mean field theory approach. This parameter depends on the activation function used, variance of biases and the entries of the weight matrices, and, for some activation functions, also on the depth of the network. We demonstrated the validity of our theoretical results in numerical experiments, both by generating random matrices adhering to the assumptions of the model and by evaluating the Jacobians of residual networks (at initialization) on the CIFAR10 dataset.

For a given activation function and/or network depth, it is always possible to set the weight matrix entries variances in such a way, that the resulting singular spectra of the Jacobians not only fulfill the conditions for dynamical isometry, but also are exactly the same, irrespective of the activation function used. This observation allows us to eliminate the singular spectrum of the Jacobian treated as a confounding factor in experiments with the learning process of simple residual neural networks for different activation functions. As an example of how this approach can be applied, we examined how accuracies of simple residual neural networks, employing a variety of activation functions, change during the learning process. When using the same variances of weight matrices entries, the learning curves of similar activation functions differed between each other more than when the networks were initialized with the same input-output Jacobian spectra. This allows, in our opinion, for a more meaningful comparison between the effects of choosing the activation function. We hope this observation will help with the research of deep neural networks.

Acknowledgements

[1] Akemann, G., Baik, J., and Di Francesco, P. (2011). *The Oxford handbook of random matrix theory*. Oxford University Press.

[2] Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., and McWilliams, B. (2017). The shattered gradients problem: If resnets are the answer, then what is the question? *arXiv preprint arXiv:1702.08591*.

[3] Canziani, A., Paszke, A., and Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.

[4] Carmona, R., Cerenzia, M., and Palmer, A. Z. (2018). The dyson game. *arXiv preprint arXiv:1808.02464*.

[5] Chatterjee, S. and Meckes, E. (2007). Multivariate normal approximation using exchangeable pairs. *arXiv preprint math/0701464*.

[6] Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. (2015). The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204.

[7] Collins, B. and Śniady, P. (2006). Integration with respect to the haar measure on unitary, orthogonal and symplectic group. *Communications in Mathematical Physics*, 264(3):773–795.

[8] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

[9] Gudowska-Nowak, E., Janik, R. A., Jurkiewicz, J., and Nowak, M. A. (2003). Infinite products of large random matrices and matrix-valued diffusion. *Nuclear Physics B*, 670(3):479–507.

[10] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[11] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. R. and Blei, D. M., editors, *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org.

[12] Janik, R. A., Nowak, M. A., Papp, G., Wambach, J., and Zahed, I. (1997). Non-hermitian random matrix models: Free random variable approach. *Physical Review E*, 55(4):4100.

[13] Janik, R. A. and Wieczorek, W. (2004). Multiplying unitary random matricesâĂŤuniversality and spectral properties. *Journal of Physics A: Mathematical and General*, 37(25):6521.

[14] Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, pages 971–980.

[15] Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. Master's thesis.

[16] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

[17] Liao, Z. and Couillet, R. (2018). The dynamics of learning: A random matrix approach. *arXiv preprint arXiv:1805.11917*.

[18] Ling, Z. and Qiu, R. C. (2018). Spectrum concentration in deep residual learning: a free probability appproach. *arXiv preprint arXiv:1807.11694*.

[19] Louart, C., Liao, Z., Couillet, R., et al. (2018). A random matrix approach to neural networks. *The Annals of Applied Probability*, 28(2):1190–1248.

[20] Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3.

[21] Mingo, J. A. and Speicher, R. (2017). *Free probability and random matrices*, volume 4. Springer.

[22] Mishkin, D. and Matas, J. (2015). All you need is a good init. *arXiv preprint arXiv:1511.06422*.

[23] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

[24] Nica, A. and Speicher, R. (1996). *r*-diagonal pairs-a common approach to haar unitaries and circular elements. *arXiv preprint funct-an/9604012*.

[25] Nica, A. and Speicher, R. (2006). *Lectures on the combinatorics of free probability*, volume 13. Cambridge University Press.

[26] Nowak, M. A. and Tarnowski, W. (2017). Complete diagrammatics of the single-ring theorem. *Physical Review E*, 96(4):042149.

[27] Pennington, J. and Bahri, Y. (2017). Geometry of neural network loss surfaces via random matrix theory. In *International Conference on Machine Learning*, pages 2798–2806.

[28] Pennington, J., Schoenholz, S., and Ganguli, S. (2017). Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *Advances in Neural Information Irocessing Systems*, pages 4785–4795.

[29] Pennington, J., Schoenholz, S. S., and Ganguli, S. (2018). The emergence of spectral universality in deep networks. *arXiv preprint arXiv:1802.09979*.

[30] Pennington, J. and Worah, P. (2017). Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, pages 2637–2646.

[31] Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. (2016). Exponential expressivity in deep neural networks through transient chaos. In *Advances in neural information processing systems*, pages 3360–3368.

[32] Saxe, A. M., McClelland, J. L., and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

[33] Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. (2016). Deep information propagation. *arXiv preprint arXiv:1611.01232*.

[34] Taki, M. (2017). Deep residual networks and weight initialization. *arXiv preprint arXiv:1709.02956*.

[35] Veit, A., Wilber, M. J., and Belongie, S. (2016). Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems*, pages 550–558.

[36] Voiculescu, D. V., Dykema, K. J., and Nica, A. (1992). *Free random variables*. Number 1. American Mathematical Soc.

[37] Wigner, E. P. (1993). Characteristic vectors of bordered matrices with infinite dimensions i. In *The Collected Works of Eugene Paul Wigner*, pages 524–540. Springer.

[38] Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S. S., and Pennington, J. (2018). Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. *arXiv preprint arXiv:1806.05393*.

[39] Yang, G. and Schoenholz, S. (2017). Mean field residual networks: On the edge of chaos. In *Advances in Neural Information Processing Systems*, pages 7103–7114.

[40] Zhang, H., Dauphin, Y. N., and Ma, T. (2019). Residual learning without normalization via better initialization. In *International Conference on Learning Representations*.

[41] We use relatively low learning rates, largely because we omit batch normalization layers in the architecture.

Appendices

Appendix A: Spectrum of the Jacobian

To make the characteristics of the spectrum more explicit, we shall calculate the positions of the spectral edges of the probability density of squared singular values. Their locations, $z*$, can be determined from the condition $\frac{1}{G'(z*)} = 0$. In this case we assume $a = 1$ for simplicity and take a derivative of (25), obtaining

$$G' = (zG' + G)e^{c(1-2zG)} - (zG - 1)e^{c(1-2zG)}2c(G + zG').\tag{A1}$$

The exponent can be eliminated with the help of (25), leading to

$$1 = \left(z + \frac{G}{G'}\right)\frac{G}{zG - 1} - 2cG\left(\frac{G}{G'} + z\right).\tag{A2}$$

Taking $\frac{1}{G'} = 0$, we arrive to the quadratic equation

$$2cz^2G^2 - 2czG - 1 = 0,\tag{A3}$$

which, together with (25), determine the location of the edges and the value of the Green's function at these points. Solving, we obtain

$$z_\pm = \left(1 + c \pm \sqrt{c(2 + c)}\right)e^{\pm \sqrt{c(2+c)}}.\tag{A4}$$

Note that the perfect isometry (i.e. all eigenvalues are 1) is achieved for $c = 0$, as independently proposed by [40], while for small c the size of the support grows sublinearly $z_\pm \approx 1 \pm 2\sqrt{2c}$. Moreover, for large c, that is far from dynamical isometry, the largest eigenvalue is exponentially large, while the smallest is exponentially small. This fact underscores importance of a proper initialization.

Appendix B: Detailed derivation of the signal propagation

With the scalings of from section III A made explicit, we have $\left\langle W_{ij}^l W_{km}^l \right\rangle_{wbl} = \left\langle W_{ij}^l W_{im}^l \right\rangle_l = \delta_{ik}\delta_{jm}\frac{(\sigma_W)^2}{LN}$. Now, based on (1) and the above considerations, we have

$$q^l = \frac{(\sigma_W)^2}{L}\left\langle x^2 \right\rangle_{l-1} + (\sigma_b)^2\tag{B1}$$

and

$$\left\langle x^2 \right\rangle_{l-1} = \left\langle \phi\left(h^{l-1}\right)^2 \right\rangle_{l-1} + \frac{2a}{N}\sum_{i=1}^N \phi\left(h_i^{l-1}\right)x_i^{l-2} + a^2\left\langle x^2 \right\rangle_{l-2}\tag{B2}$$

We assume that the factorization $\frac{1}{N}\sum_k x_k^{l-1}\phi(h_k^l) = \langle x \rangle_{l-1}\int \mathcal{D}u\phi(u\sqrt{q^l})$ holds in the large N limit. This is justified, as the input to h_k^l comes from all the many elements of x^{l-1}. We can rewrite (B2) as

$$\left\langle x^2 \right\rangle_{l-1} = \int \mathcal{D}z\phi^2\left(\sqrt{q^{l-1}}z\right) + 2a\langle x \rangle_{l-2}\int \mathcal{D}z\phi\left(\sqrt{q^{l-1}}z\right) + a^2\left\langle x^2 \right\rangle_{l-2}\tag{B3}$$

Turning to $\langle x \rangle_{l-2}$, based on (1), we have:

$$\langle x \rangle_l = a\langle x \rangle_{l-1} + \int \mathcal{D}z\phi\left(\sqrt{q^l}z\right).\tag{B4}$$

For $\langle x \rangle_0 = 0$ the recurrence yields

$$\langle x \rangle_l = \sum_{k=0}^{l-1}a^k\int \mathcal{D}z\phi\left(\sqrt{q^{l-k}}z\right).\tag{B5}$$

Thus, (B3), with a shift in l, turns into

$$\langle x^2 \rangle_l = \int \mathcal{D}z \phi^2 \left(\sqrt{q^l} z \right) + 2 \left[\sum_{k=1}^{l-1} a^k \int \mathcal{D}z \phi \left(\sqrt{q^{l-k}} z \right) \right] \int \mathcal{D}z \phi \left(\sqrt{q^l} z \right) + a^2 \langle x^2 \rangle_{l-1}. \tag{B6}$$

Finally, we use (B1) to obtain

$$q^{l+1} = a^2 q^l + \left(1 - a^2 \right) \sigma_b^2 + \frac{(\sigma_W)^2}{L} \int \mathcal{D}z \phi^2 \left(\sqrt{q^l} z \right) + 2 \frac{(\sigma_W)^2}{L} \left[\sum_{k=1}^{l-1} a^k \int \mathcal{D}z \phi \left(\sqrt{q^{l-k}} z \right) \right] \int \mathcal{D}z \phi \left(\sqrt{q^l} z \right), \tag{B7}$$

which is a closed recursive equation for q^l. We note that for $a = 0$, the known, feed-forward network recursion relation is recovered. Furthermore, for the case of $a = 1$, in contrast to the feed-forward architecture, the biases do not influence the statistical properties of the pre-activations. Moreover, for ResNets, this recursive relation is iteratively additive, namely each q^{l+1} is a result of adding some terms to the previous q^l. In all the examples studied below, the first term is positive and the second term is non-negative. This in turn means that the variance of pre-activations grows with the networks depth and there are no non-trivial fixed points of this recursion equation. Finally, here we can see the importance of the $\frac{1}{N}$ scaling of $(\sigma_W)^2$, without which, q^l would grow uncontrollably with l.

Appendix C: Results for various activation functions

We now investigate particular examples of activation functions. For simplicity, we consider purely residual networks (we set $a = 1$). The numerical verifications of the results presented here will follow in the next subsection.

1. Linear

In the case of the linear activation function $\phi'(x) = 1$ and there is no need to consider the way the pre-activations change across the network. Thus we can proceed to calculating the cumulant which yields $c = c_2 = \sigma_W^2$.

2. Rectified Linear Unit

The example of ReLU is only slightly more involved. Now we have $\phi'(x) = \theta(x)$, where $\theta(x)$ is the Heaviside theta function, and thus

$$c_2^l = \sigma_W^2 \int \mathcal{D}u \phi'^2 \left(u \sqrt{q^l} \right) = \int_0^\infty \mathcal{D}u = \frac{1}{2} \sigma_W^2. \tag{C1}$$

3. Leaky ReLU

The activation function interpolating between the first two examples is $\phi(x) = \max(\alpha x, x)$ with $0 < \alpha < 1$. In this case

$$c_2^l = \sigma_W^2 \left(\int_{-\infty}^0 \alpha^2 \mathcal{D}u + \int_0^\infty \mathcal{D}u \right) = \frac{\sigma_W^2}{2} (1 + \alpha^2). \tag{C2}$$

All together, this leads to the following equation for the Green's function

$$G(z) = (zG(z) - 1) e^{\frac{1}{2} \sigma_W^2 (1 + \alpha^2)(1 - 2zG(z))}, \tag{C3}$$

where $\alpha = 1$ corresponds to the linear activation function and $\alpha = 0$ to ReLU.

Equation (C3) can be easily solved numerically for the spectral probability density of the Jacobian. For completeness, we write down the recursion relation (B7) in these three cases:

$$q^l = q^{l-1} + \frac{\sigma_W^2}{2L} \left(\alpha^2 + 1 \right) q^{l-1} + \frac{\sigma_W^2}{\pi L} (1 - \alpha)^2 \sqrt{q^{l-1}} \left(\sum_{k=1}^{l-2} \sqrt{q^k} \right). \tag{C4}$$

For the linear activation function ($\alpha = 1$), its solution is readily available and reads

$$q^l = q^1 \left(1 + \frac{\sigma_W^2}{L} \right)^{l-1} \simeq q^1 e^{(l-1) \frac{\sigma_W^2}{L}}, \tag{C5}$$

which explicitly shows the importance of the $1/L$ rescaling introduced earlier.

4. Hard hyperbolic tangent

The hard tanh activation function is defined by $\phi(x) = x$ for $|x| \leq 1$ and $\phi(x) = \text{sgn}(x)$ elsewhere. Thus:

$$c_2^l = \sigma_W^2 \int_{-1}^{1} \mathcal{D}u = \sigma_W^2 \, \text{erf}\left(\frac{1}{\sqrt{2}}\right) = c. \tag{C6}$$

The resulting recurrence equation for the variance of the preactivations reads:

$$q^{l+1} = q^l \left[1 + \frac{\sigma_W^2}{L}\left(\text{erf}\left(\frac{1}{\sqrt{2}}\right) - \sqrt{\frac{2}{\pi e}}\right) + \frac{\sigma_W^2}{L}\left(1 - \text{erf}\left(\frac{1}{\sqrt{2}}\right)\right)\right] \tag{C7}$$

and can be easily solved.

In the preceding examples we dealt with piecewise linear activation functions. Note that in these cases the parameter c does not depend on the variance of biases and linearly increases with the variance of weights. For other nonlinear activation functions to obtain the cumulants, we need to use the recurrence relation describing the signal propagation in the network.

5. Hyperbolic tangent

When $\phi(x) = \tanh(x)$, the activation function is antisymmetric and the last term of (B7) vanishes. Thus the recurrence takes the form:

$$q^{l+1} = q^l + \frac{(\sigma_W)^2}{L} \int \mathcal{D}z \phi^2\left(\sqrt{q^l} z\right) \tag{C8}$$

In the large L limit, we can write

$$q^{l+1} = q^l + \frac{(\sigma_W)^2}{L} \int \mathcal{D}z \phi^2\left(\sqrt{q^{l-1} + \Delta^l} z\right), \tag{C9}$$

where we assume $\Delta^l \sim 1/L$. Expanding this recursively around q^{l-1} for decreasing l and keeping only the leading term, as long as for any k, $q^k \gg 1/L^2$, we obtain :

$$q^{l+1} \approx q^l + \frac{(\sigma_W)^2}{L} \int \mathcal{D}z \phi^2\left(\sqrt{q^1} z\right). \tag{C10}$$

Therefore, the solution of the recursion is:

$$q^l \approx q^1 + (l-1)\frac{(\sigma_W)^2}{L} \int \mathcal{D}z \phi^2\left(\sqrt{q^1} z\right). \tag{C11}$$

The variance grows linearly with l (this is verified in appendix D, see Fig. 5). Cumulants c_2^l and thus c are obtained with numerical integration.

In fact, in the above calculations we have only used the antisymmetry property of the hyperbolic tangent activation function and the properties of its behavior near q^1. Therefore, these results are valid for other antisymmetric activation functions like $\phi(x) = \arctan(x)$.

6. Sigmoid

The sigmoid activation function, $\phi(x) = \frac{1}{1+e^{-x}}$ is the first example we encounter, for which $\langle \phi(h) \rangle \neq 0$, thus it deserves special attention. In particular, in this case one needs to additionally address the last term in (B7). It turns out, that

$$\int \mathcal{D}z \phi\left(\sqrt{q^l} z\right) = \frac{1}{2} \tag{C12}$$

irrespective of l. Therefore, the recurrence relation becomes:

$$q^{l+1} = q^l + \frac{(\sigma_W)^2}{L} \int \mathcal{D}z \phi^2\left(\sqrt{q^l} z\right) + \frac{(\sigma_W)^2}{2L}(l-1). \tag{C13}$$

Thus, we can see, that due to the non-zero first moment of the activation function (C12) the mean and the second moment of post-activations grow with depth. Similarly, the variance of pre-activations increases as the signal propagates, which causes quick saturation of the sigmoid nonlinearity. This in turn precludes training of deep networks [8].

Analogically to the previous case, one can derive an approximation to the solution of the recursion relation. In this case it becomes:

$$q^{l+1} \approx q^1 + \frac{(\sigma_W)^2 l}{L} \int \mathcal{D}z\phi^2\left(\sqrt{q^1}z\right) + \frac{(\sigma_W)^2}{4L}l(l-1). \tag{C14}$$

We verify this result in Fig. 5 in Appendix D

7. Scaled Exponential Linear Units

Our final example is the SELU activation function, one introduced recently in [14] with the intent to bypass the batch normalization procedure. In this case, we have $\phi(x) = \lambda x$ for $x > 0$ and $\phi(x) = \lambda\beta(e^x - 1)$ for $x \leq 0$. Thus, it is not antisymmetric and is nonlinear for negative arguments. It turns out, that

$$c_2^l = \frac{(\sigma_W)^2\lambda^2}{2}\left[1 + \beta^2 e^{2q^l}\operatorname{erfc}\left(\sqrt{2q^l}\right)\right]. \tag{C15}$$

Moreover:

$$\int \mathcal{D}z\phi^2\left(\sqrt{q^l}z\right) = \frac{\lambda^2 q^l}{2} + \frac{\beta^2\lambda^2}{2}\left[1 + e^{2q^l}\operatorname{erfc}\left(\sqrt{2q^l}\right) - 2e^{q^l/2}\operatorname{erfc}\left(\sqrt{\frac{q^l}{2}}\right)\right] \tag{C16}$$

and

$$\int \mathcal{D}z\phi\left(\sqrt{q^l}z\right) = \lambda\sqrt{\frac{q^l}{2\pi}} + \frac{\lambda\beta}{2}\left(e^{q^l/2}\operatorname{erfc}\left(\sqrt{\frac{q^l}{2}}\right) - 1\right). \tag{C17}$$

These yield the recursion relation for q^l via (B7). One can check that for $\beta = 0$ and $\lambda = 1$, the results for ReLU are recovered.

These theoretical predictions for the recursion relations are tested with numerical simulations using Mathematica. The results are relegated to Appendix D.

Appendix D: Numerical verification of the recurrence relations

To validate the assumptions made and corroborate the theoretical results obtained in subsections III B and C, we simulate signal propagation in the studied residual neural networks with different activation functions. The outcomes of these experiments are depicted in Figure 5. Numerical solution to the recurrence relations allows us to numerically calculate the parameter c, as a function of variances of weights and biases, which is presented in Figure 6.

Appendix E: Baseline

We advocate for setting the same value of the effective cumulant (and hence keeping the same spectrum of the input-output Jacobian) when comparing the effects of using different activation function on the learning process. Thus, here, in Fig. 7, for comparison, we showcase the learning accuracy when instead of the effective cumulant, the weight matrices entries' variances (equal to $1/NL$) are kept the same across the networks.
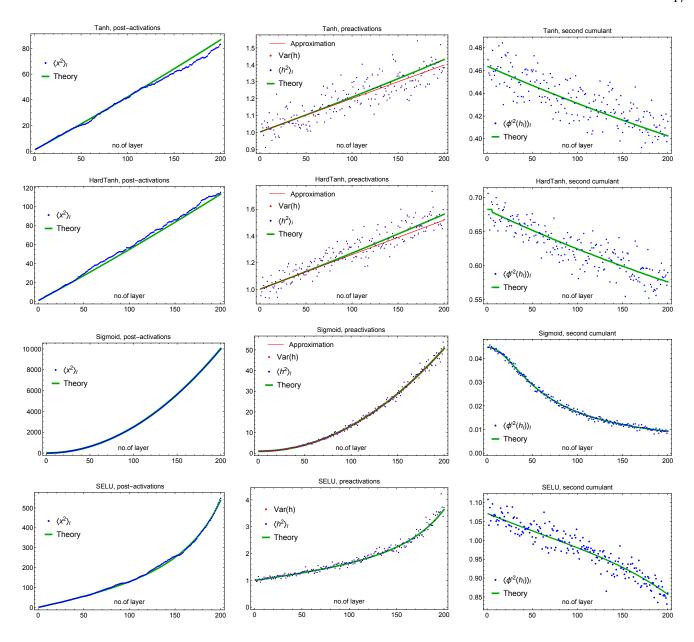
FIG. 5: Verification of the numerical solution to the recurrence equations for post-activations (B3) (left column) and preactivations (B7) (middle). Based on this signal propagation the effective cumulant c_l for each layer was calculated (right column). The solid red lines represent the approximation (C11) for tanh and hard tanh nonlinearity and (C14) for sigmoid. Solutions of recurrences (solid) are confronted with the numerical simulation (dots) of residual fully connected networks with $L = 200$ layers of width $N = 800$. Data points represent a single run of simulations. Weights are independently sampled from Gaussian distribution of zero mean and variance equal to $\frac{1}{NL}$. Biases and network input are sampled from standard normal distribution. A small variability of c^l across the network justifies the assumption made for derivation of (22).
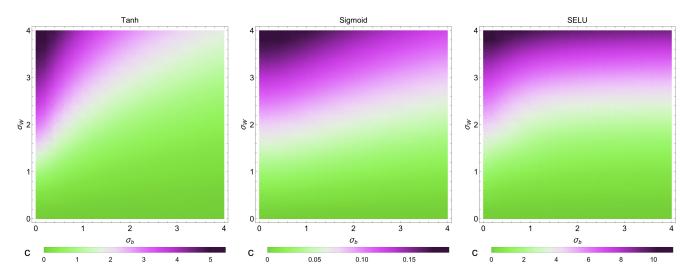
FIG. 6: Dependence of the parameter c (which determines the shape of the spectrum) on the variances of biases and weights. The smaller c, the closer to the perfect dynamical isometry. Note different scales on each plot. Low value of c for sigmoid is a consequence of saturation of the nonlinearity.
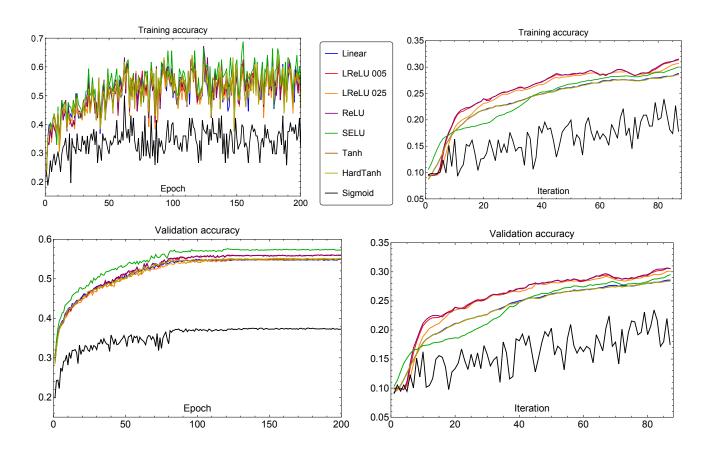

FIG. 7: Training (top) and validation (bottom) accuracy during first 200 epochs (left) and first 100 iterations (right) of residual networks with various activation functions. The weight initialization was Gaussian with zero mean and $1/NL$ variance. We set $\alpha = 0.05$ and $\alpha = 0.25$ for leaky ReLU (LReLU).