

A Law of Data Separation in Deep Learning

Hangfeng He^{a,b} and Weijie J. Su^{c,1}

^aDepartment of Computer Science, University of Rochester, Rochester, NY 14627; ^bGoergen Institute for Data Science, University of Rochester, Rochester, NY 14627;

^cDepartment of Statistics and Data Science, University of Pennsylvania, Philadelphia, PA 19104

This manuscript was compiled on August 14, 2023

While deep learning has enabled significant advances in many areas of science, its black-box nature hinders architecture design for future artificial intelligence applications and interpretation for high-stakes decision makings. We addressed this issue by studying the fundamental question of how deep neural networks process data in the intermediate layers. Our finding is a simple and quantitative law that governs how deep neural networks separate data according to class membership throughout all layers for classification. This law shows that each layer improves data separation at a constant geometric rate, and its emergence is observed in a collection of network architectures and datasets during training. This law offers practical guidelines for designing architectures, improving model robustness and out-of-sample performance, as well as interpreting the predictions.

deep learning | data separation | constant geometric rate | intermediate layers

Deep learning methodologies have achieved remarkable success across a wide range of data-intensive tasks in image recognition, biological research, and scientific computing (2–5). In contrast to other machine learning techniques (6), however, the practice of deep learning relies heavily on a plethora of heuristics and tricks that are not well justified. This situation often makes deep learning-based approaches ungrounded for some applications or necessitates the need for huge computational resources for exhaustive search, making it difficult to fully realize the potential of this set of methodologies (7).

This unfortunate situation is in part owing to a lack of understanding of how the prediction depends on the intermediate layers of deep neural networks (8–10). In particular, little is known about how the data of different classes (e.g., images of cats and dogs) in classification problems are gradually separated from the bottom layers to the top layers in modern architectures such as AlexNet (2) and residual neural networks (11). Any knowledge about data separation, especially quantitative characterization, would offer useful principles and insights for designing network architectures, training processes, and model interpretation.

The main finding of this paper is a quantitative delineation of the data separation process throughout all layers of deep neural networks. As an illustration, Fig. 1 plots a certain value that measures how well the data are separated according to their class membership at each layer for feedforward neural networks trained on the Fashion-MNIST dataset (12). This value in the logarithmic scale decays, in a distinct manner, linearly in the number of layers the data have passed through. The Pearson correlation coefficients between the logarithm of this value and the layer index range from -0.997 and -1 in Fig. 1.

The measure shown in Fig. 1 is canonical for measuring data separation in classification problems. Let x_{ki} denote an intermediate output of a neural network on the i th point of Class k for $1 \leq i \leq n_k$, \bar{x}_k denote the sample mean of Class

k , and \bar{x} denote the mean of all $n := n_1 + \dots + n_K$ data points. We define the between-class sum of squares and the within-class sum of squares as

$$\begin{aligned} \text{SS}_b &:= \frac{1}{n} \sum_{k=1}^K n_k (\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^\top \\ \text{SS}_w &:= \frac{1}{n} \sum_{k=1}^K \sum_{i=1}^{n_k} (x_{ki} - \bar{x}_k)(x_{ki} - \bar{x}_k)^\top, \end{aligned}$$

respectively. The former matrix represents the between-class “signal” for classification, whereas the latter denotes the within-class variability. Writing SS_b^+ for the Moore–Penrose inverse of SS_b (The matrix SS_b has rank at most $K - 1$ and is not invertible in general since the dimension of the data is typically larger than the number of classes.), the ratio matrix $\text{SS}_w \text{SS}_b^+$ can be thought of as the inverse signal-to-noise ratio. We use its trace

$$D := \text{Tr}(\text{SS}_w \text{SS}_b^+) \quad [1]$$

to measure how well the data are separated (13, 14). This value, which is referred to as the separation fuzziness, is large when the data points are not concentrated to their class means or, equivalently, are not well separated, and vice versa.

1. Main Results

Given an L -layer feedforward neural network, let D_l denote the separation fuzziness (Eq. 1) of the training data passing through the first l layers for $0 \leq l \leq L - 1$.* Fig. 1 suggests that the dynamics of D_l follows the relation

$$D_l \doteq \rho^l D_0 \quad [2]$$

*For clarification, D_0 is calculated from the raw data, and D_1 is calculated from the data that have passed through the first layer but not the second layer.

Significance Statement

The practice of deep learning has long been shrouded in mystery, leading many to believe that the inner workings of these black-box models are chaotic during training. In this paper, we challenge this belief by presenting a simple and approximate law that deep neural networks follow when processing data in the intermediate layers. This empirical law is observed in a class of modern network architectures for vision tasks, and its emergence is shown to bring important benefits for the trained models. The significance of this law is that it allows for a new perspective that provides useful insights into the practice of deep learning.

H.H and W.J.S. designed research, performed research, analyzed data, and wrote the paper.

The authors declare no competing interest.

¹To whom correspondence should be addressed. E-mail: suw@wharton.upenn.edu.

Fig. 1. Illustration of the law of equi-separation in feedforward neural networks with ReLU activations trained on the Fashion-MNIST dataset. The three rows correspond to three different training methods, stochastic gradient descent (SGD), SGD with momentum, and Adam (1). Throughout the paper, the *x* axis represents the layer index and the *y* axis represents the separation fuzziness defined in Eq. 1, unless otherwise specified. The Pearson correlation coefficients, by row first, are $-1.000, -0.998, -0.997, -0.999, -0.998, -0.998, -1.000, -0.997$, and -0.999 . The last two rows show the intermediate data points on the plane of the first two principal components for the 8-layer network trained using Adam. Layer 0 corresponds to the raw input. More details can be found in SI Appendix.

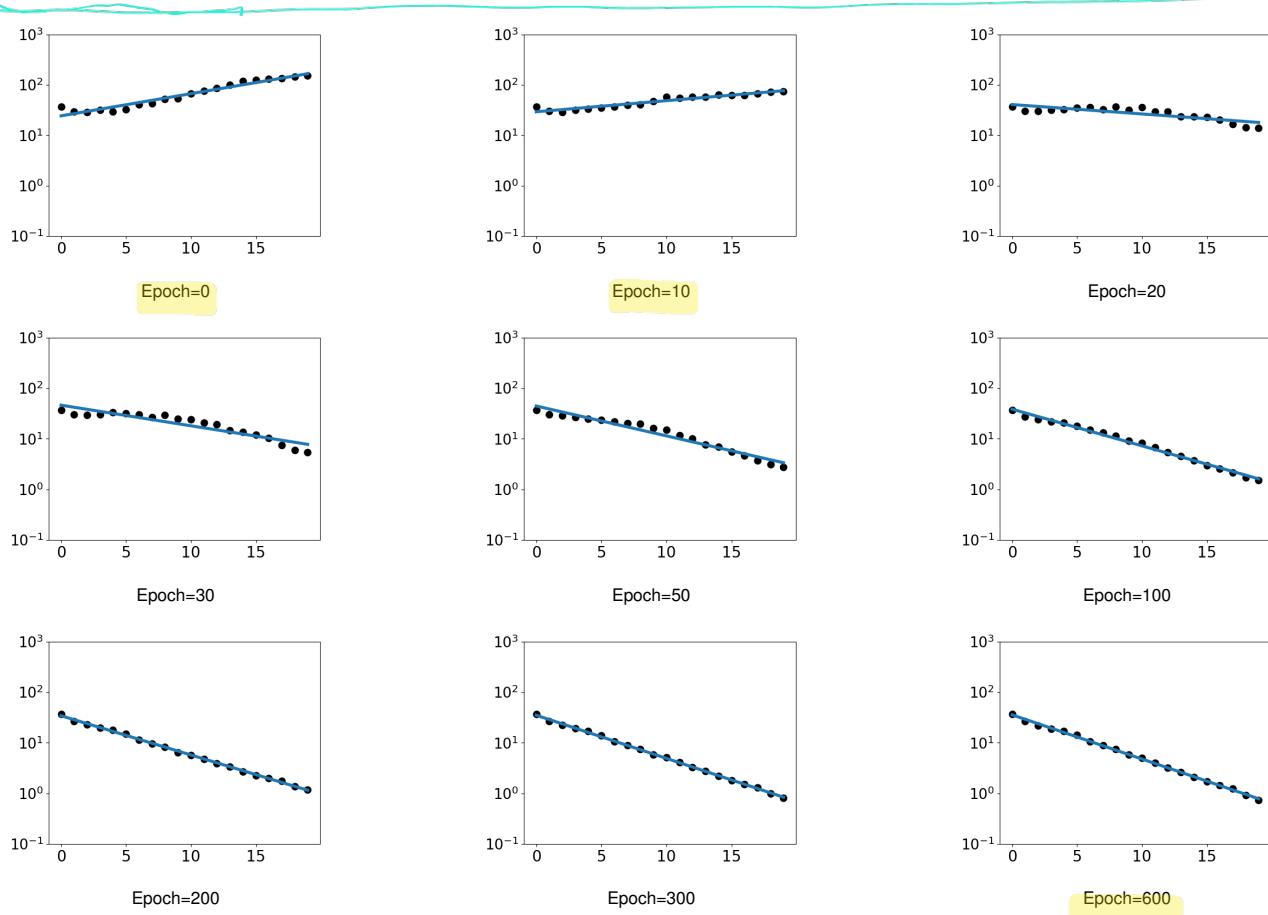


Fig. 2. A 20-layer feedforward neural network trained on Fashion-MNIST. The law of equi-separation starts to emerge at epoch 100 and becomes more clear as training proceeds. As in Fig. 1 and all the other figures in the main text, the x axis represents the layer index and the y axis represents the separation fuzziness.

for some decay ratio $0 < \rho < 1$. Alternatively, this law implies $\log D_{l+1} - \log D_l = -\log \frac{1}{\rho}$, showing that the neural network makes equal progress in reducing $\log D$ over each layer on the training data. Hence, we call this the *law of equi-separation*. This law is the first quantitative and geometric characterization of the data separation process in the intermediate layers. Indeed, it is unexpected because the intermediate output of the neural network does not exhibit any quantitative patterns, as shown by the last two rows of Fig. 1.

The decay ratio ρ depends on the depth of the neural network, dataset, training time, and network architecture, and is also affected, to a lesser extent, by optimization methods and many other hyperparameters. For the 20-layer network trained using Adam (1) (the bottom-right plot of Fig. 1), the decay ratio ρ is 0.818. Thus, the half-life is $\frac{\ln 2}{\ln \rho^{-1}} = \frac{0.693}{\ln \rho^{-1}} = 3.45$, suggesting that this 20-layer neural network reduces the value of the separation fuzziness in every three and a half layers.

This law manifests in the terminal phase of training (14), where we continue to train the model to interpolate in-sample data. At initialization, the separation fuzziness may even increase from the bottom to top layers. During the early stages of training, the bottom layers tend to learn faster at reducing the separation fuzziness compared to the top layers (see Fig. S7 in SI Appendix). However, as training progresses, the top layers eventually catch up as the bottom layers have learned the necessary features. Finally, each layer becomes

roughly equally capable of reducing the separation fuzziness multiplicatively. This dynamics of data separation during training is illustrated in Fig. 2. Neural networks in the terminal phase of training also exhibit certain symmetric geometries in the last layer such as neural collapse (14, 15) and minority collapse (10). However, it is worthwhile mentioning that the law of equi-separation emerges earlier than neural collapse during training (see Fig. S1 in SI Appendix).

The pervasive law of equi-separation consistently prevails across diverse datasets, learning rates, and class imbalances, as illustrated in Fig. 3. Additionally, Fig. S6 in SI Appendix demonstrates its applicability in a finer, class-wise context. This law is further exemplified in contemporary network architectures for vision tasks, such as AlexNet and VGGNet (16), as shown in Fig. 4 (see SI Appendix for additional convolutional neural network experiments in Fig. S2 and Fig. S3). Moreover, the law manifests in residual neural networks and densely connected convolutional networks (17) when separation fuzziness is assessed at each block, as depicted in Fig. 7 and Fig. 8, respectively. Intriguingly, this law appears slightly more pronounced in feedforward neural networks when compared with other network architectures.

The separation dynamics of neural networks have been extensively investigated in prior research studies (18–22). For instance, (18) employed linear classifiers as probes to assess the separability of intermediate outputs. In (19), the author scrupu-

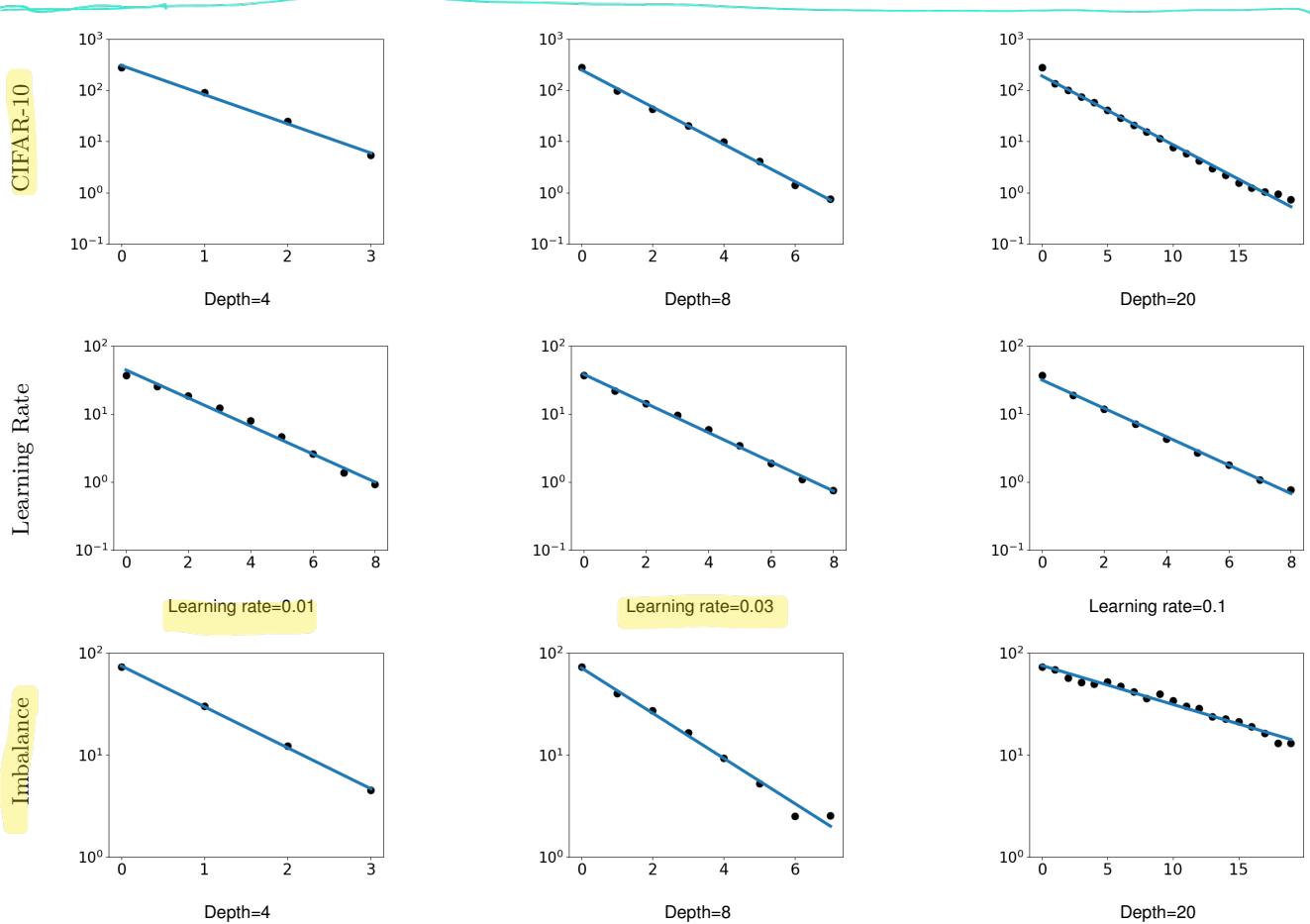


Fig. 3. Illustration showing how the law of equi-separation holds in a wide range of settings. The experimental details are provided in SI Appendix.

tinized the separation capabilities of neural networks through empirical spectral analysis. More recently, (20–22) explored the separation ability of neural networks by examining neural collapse at intermediate layers and its relationship with generalization. In particular, (22) provided crucial experimental evidence illustrating the progression of neural collapse within the interior of neural networks, which is perhaps the most relevant work to our paper.

2. Insights from the Law

Next, we demonstrate the benefits of the law of equi-separation and the perspective of data separation in regard to the three fundamental aspects of deep learning practice, namely, architecture design, training, and interpretation.

A. Network architecture. The law of equi-separation can inform the design principles for network architecture. First, the law implies that neural networks should be deep for good performance, thereby reflecting the hierarchical nature of deep learning (3, 23–25). This is because, as shown by Eq. 2, all layers reduce the separation fuzziness D_0 of the raw input to $\rho^{L-1}D_0$ at the last layer. When L is small—say, $L = 2$ or 3—the ratio ρ^{L-1} would generally not be small and, therefore, the neural network is unlikely to separate the data well. In the literature, the fundamental role of depth is recognized by analyzing the loss functions (11, 16, 23, 26, 27), and our law

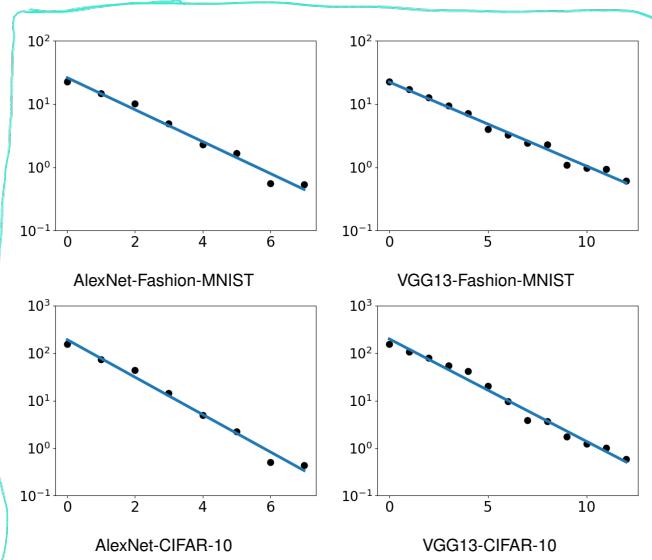


Fig. 4. Illustration of the equi-separation law on convolutional neural networks. A more in-depth examination of convolutional neural networks can be found in Figs. S2 and S3 in SI Appendix.

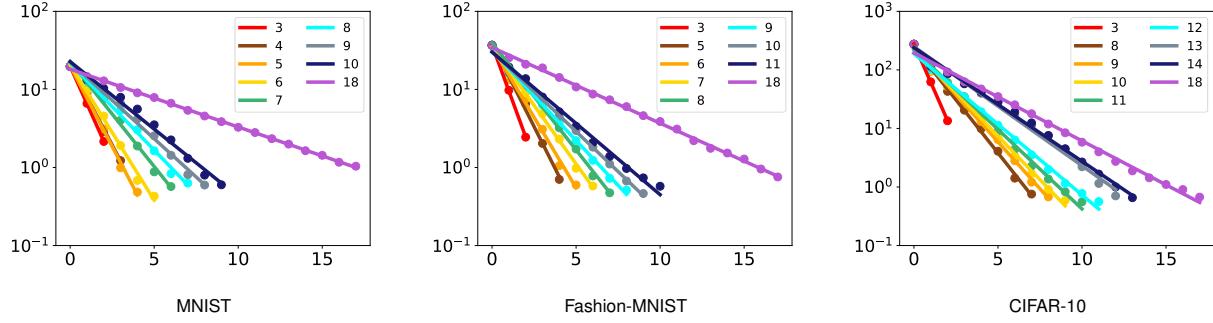


Fig. 5. Illustration of the law of equi-separation with varying network depths. The legend shows how depths are color coded. The depth that yields the lowest separation fuzziness depends on the dataset complexity.

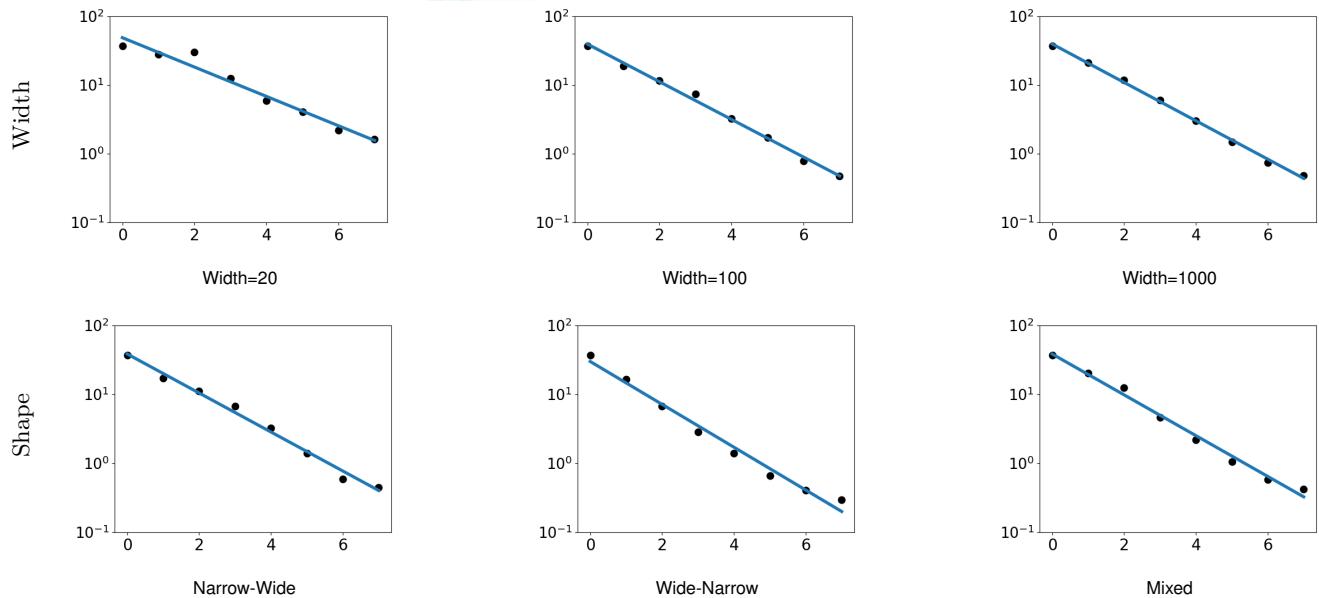


Fig. 6. Illustration of the law of equi-separation with different widths and shapes of the neural networks. For example, “Narrow-Wide” means that the bottom layers are narrower than the top layers.

of equi-separation offers a new perspective on network depth.

For completeness, the deeper the better is not necessarily true because as the depth L increase, ρ might become larger. Moreover, a very large depth would render the optimization problem challenging. As shown in Fig. 1, the 20-layer neural networks have higher values of the final separation fuzziness than those of their 8-layer counterparts. The Fashion-MNIST dataset is not very complex and, therefore, an 8-layer network will suffice to classify the data well. Adding additional layers, therefore, would only give rise to optimization burdens. Indeed, Fig. 5 illustrates the law of equi-separation with varying network depths, showing that different datasets correspond to different optimal depths. Therefore, as a practical guideline, the choice of depth should consider the complexity of the applications. For instance, this perspective of data separation shows that the optimal depths for MNIST (28), Fashion-MNIST, and CIFAR-10 (29) are 6, 10, and 12, respectively. This is consistent with the increasing complexity from MNIST to Fashion-MNIST to CIFAR-10 (11, 30, 31).

Moreover, the law of equi-separation indicates that depth bears a more significant connection to training performance

than the “shape” of neural networks. In the first row of Fig. 6, the network does not separate the data well when the width of each layer is merely 20. This is because the network has to be wide enough to pass the useful features of the data for classification. However, when the network is wide enough, the network performance may be saturated even if more neurons are added to each layer (32). In addition, the second row of Fig. 6 shows that it is better to start with relatively wide layers at the bottom layers followed by narrow top layers. These observations suggest that very wide neural networks in general should not be recommended as they consume more time for optimization. Apart from encoding purposes, the width should be set to about the same across all layers or be made larger for the bottom layers (30, 33).

B. Training. The emergence of the law of equi-separation during training is indicative of good model performance. For example, its manifestation can improve the robustness to model shifts. Indeed, the data separation ability of the network, as measured by $\frac{D_{L-1}}{D_0}$, becomes least sensitive to perturbations in network weights when the law manifests. To see this point,

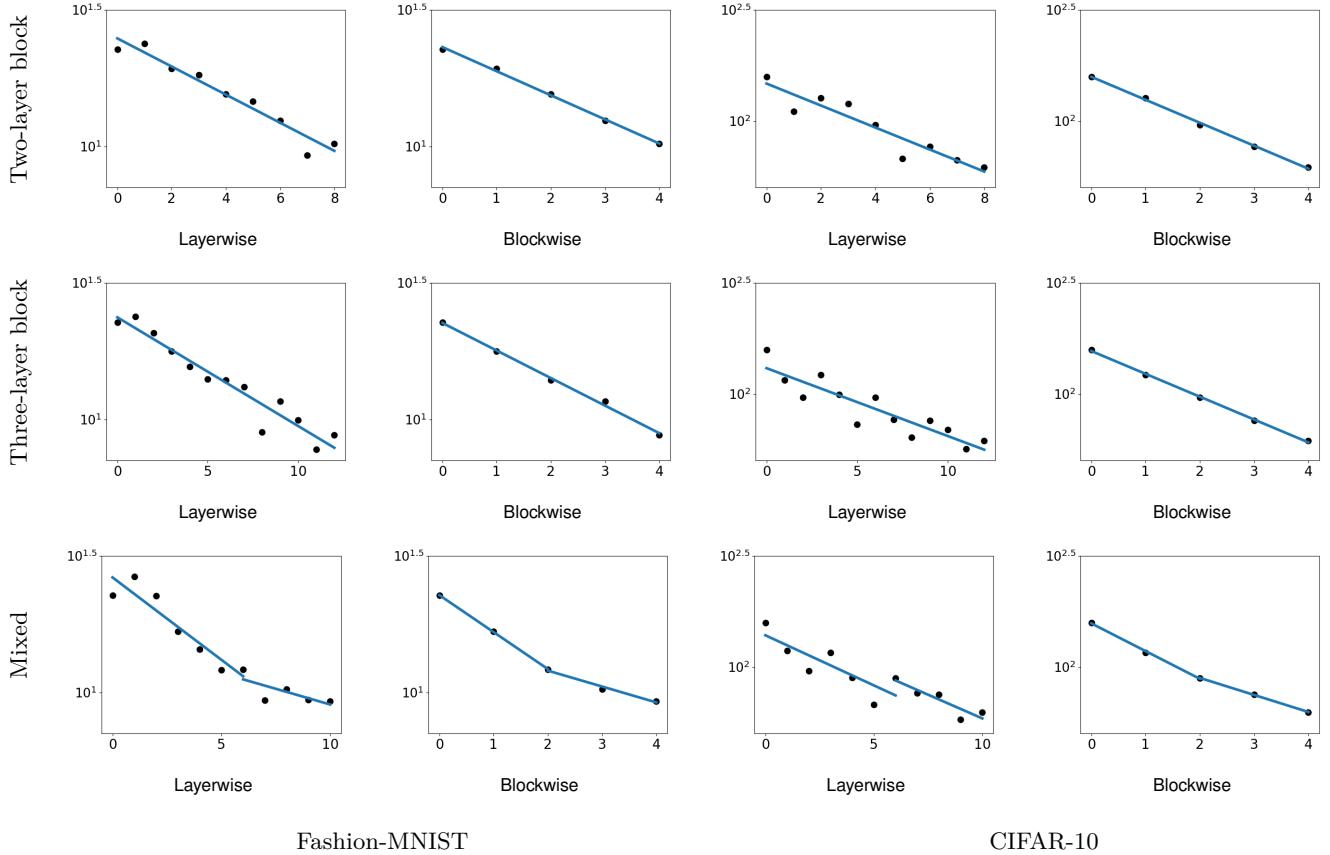


Fig. 7. Illustration of the law of equi-separation in residual neural networks: Each block contains either two or three layers. In the “Mixed” configuration, the first two blocks consist of three layers each, while the last two blocks have two layers each. In each plot, a layer or a block is identified as a module when presenting the x axis. The first two columns are evaluated on the Fashion-MNIST dataset, and the last two columns are evaluated on the CIFAR-10 dataset.

let

$$R := \frac{D_{L-1}}{D_{L-2}} \times \frac{D_{L-2}}{D_{L-3}} \times \cdots \times \frac{D_1}{D_0} = \frac{D_{L-1}}{D_0}$$

denote the reduction ratio of the separation fuzziness made by the entire network. Assuming that a perturbation of the network weights leads to a change of ε in the ratio for each layer, the perturbed reduction ratio becomes

$$\begin{aligned} R_\varepsilon &:= \left(\frac{D_{L-1}}{D_{L-2}} + \varepsilon \right) \left(\frac{D_{L-2}}{D_{L-3}} + \varepsilon \right) \cdots \left(\frac{D_1}{D_0} + \varepsilon \right) \\ &= R + R \left(\frac{D_{L-2}}{D_{L-1}} + \frac{D_{L-3}}{D_{L-2}} + \cdots + \frac{D_0}{D_1} \right) \varepsilon + O(\varepsilon^2). \end{aligned} \quad [3]$$

Applying the inequality of arithmetic and geometric means, the perturbation term $R \left(\frac{D_{L-2}}{D_{L-1}} + \frac{D_{L-3}}{D_{L-2}} + \cdots + \frac{D_0}{D_1} \right) \varepsilon$ is minimized in absolute value when $\frac{D_{L-1}}{D_{L-2}} = \frac{D_{L-2}}{D_{L-3}} = \cdots = \frac{D_1}{D_0}$. This result is summarized in the following proposition:

Proposition 1. For small ε , $|R_\varepsilon - R|$ is minimized when

$$\frac{D_{L-1}}{D_{L-2}} = \frac{D_{L-2}}{D_{L-3}} = \cdots = \frac{D_1}{D_0}.$$

This condition is precisely the case when the equi-separation law holds. Thus, the data separation ability of the classifier is not very sensitive to perturbations in network weights when the law emerges. Therefore, to improve the robustness, we need to

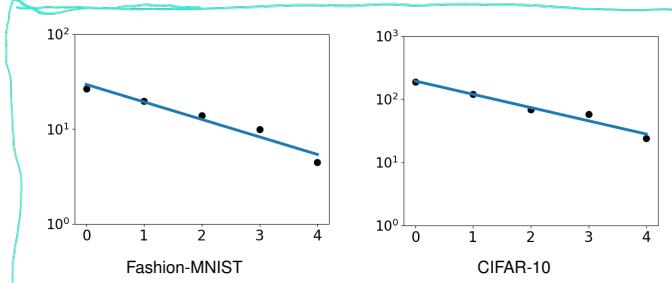


Fig. 8. Illustration of the law of equi-separation in densely connected convolutional networks (DenseNet161) by identifying a block as a module.

train a neural network at least until the law of equi-separation comes into effect. In the literature, however, connections with robustness are often established through loss functions rather than the data separation perspective (34).

This proposition suggests that the law of equi-separation could be considered a demonstration of the neural networks’ “homogeneity” across all layers. An essential component in the derivation of Proposition 1 involves the assumption that the magnitude of the perturbation remains uniform across all layers, as seen from Eq. 3. Even though it is improbable for the perturbation to remain constant in practice, it is intriguing to note that the emergence of this law is contingent on the use of batch normalization (35), which enhances the homogeneity

of the networks in a certain sense. Indeed, Fig. S10 in SI Appendix shows that the law does not appear when batch normalization is not used.

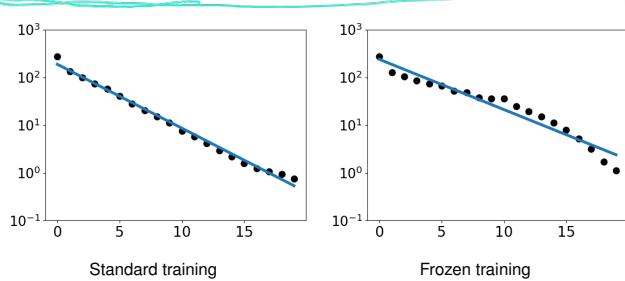


Fig. 9. Illustration showing the law of equi-separation manifesting in the left network but not in the right network because of frozen training. Both networks have about the same training losses as well as training accuracies. However, the left network has a higher test accuracy (23.85%) than that of the right network (19.67%).

The law of equi-separation can also shed light on the out-of-sample performance of neural networks. Fig. 9 considers two neural networks where one exhibits the law of equi-separation and the other does not. The results of our experiments showed that the former one has better test performance. Although this law is sometimes not observed in neural networks with good test performance, interestingly, one can fine-tune the parameters to reactivate the law with about the same or better test performance.

Moreover, the law of equi-separation can be potentially used to facilitate the training of transfer learning. In transfer learning, for example, bottom layers trained from an upstream task can form part of a network that will be trained on a downstream task. A challenge arising in practice is how to balance between the training times on the source and the target. In light of the law of equi-separation, a sensible balance is to find when the improvement in reducing separation fuzziness is roughly the same between the bottom and top layers.

C. Interpretation. The equi-separation law offers a new perspective for interpreting deep learning predictions, especially in high-stakes decision makings. An important ingredient in interpretation is to pinpoint the basic operational modules in neural networks and subsequently to reflect that “all modules are created equal.” For feedforward and convolutional neural networks, each layer is a module as it reduces the separation fuzziness by an equal multiplicative factor (for more details, see Fig. S11 in SI Appendix). Interestingly, even when the law no longer manifests in residual neural networks, Fig. 7 shows that it can be restored by identifying a block rather than a layer as a module. This figure also suggests that blocks containing more layers are more capable of reducing the separation fuzziness. Likewise, Fig. 8 shows that the law of equi-separation holds in densely connected convolutional networks by identifying a block as a module. For comparison, when the perspective of data separation is not considered (36), the correct module cannot be identified. For completeness, it should be noted that the law becomes less clear for deeper residual neural networks[†], as shown in Fig. 10.

Because each module makes equal but small contributions, a few neurons in a single layer are unlikely to explain how a deep learning model makes a particular prediction. It is therefore

[†]See the architecture in <https://github.com/kuangliu/pytorch-cifar/tree/master/models>.

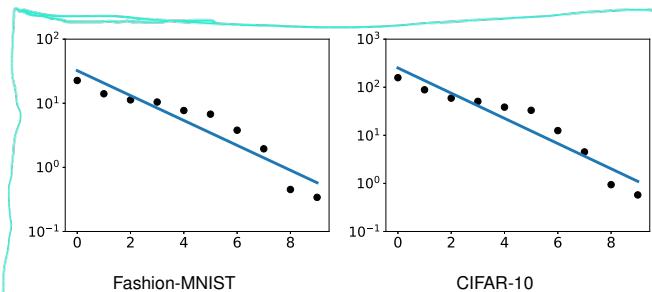


Fig. 10. The law of equi-separation in ResNet18 is not as pronounced as it is in shallower residual neural networks.

crucial to take all layers collectively for interpretation (37). This view, however, challenges the widely used layer-wise approaches to deep learning interpretation (38, 39).

3. Discussion

Due to (14) and follow-up work, it has been known that neural networks can exhibit precise mathematical structures at the last layer in the terminal phase of training. In this paper, we have extended this viewpoint from the surface of these black-box models to their interior by introducing an empirical law that quantitatively governs how well-trained real-world neural networks separate data throughout all layers. This law has been demonstrated to provide useful guidelines and insights into deep learning practice through network architecture design, training, and interpretation of predictions.

One avenue for future research is to explore the applicability of the law of equi-separation to a wider range of network architectures and applications. For example, it would be interesting to see if this law holds for neural ordinary differential equations and, if so, how it could be used to improve our understanding of the depth of these continuous-depth models. Additionally, it would be valuable to examine the law in regression settings and multi-modal tasks (40). A crucial question that remains to be addressed is understanding why the law is most accurate in feedforward neural networks and becomes somewhat fuzzier in certain convolutional neural networks. Next, given that the law is most evident in feedforward neural networks, an interesting direction is to explore new measures in lieu of the separation fuzziness defined in Eq. 1. The hope is that this could potentially make the law clearer in other network architectures. In doing so, it would be beneficial to take into account network structures, such as convolution kernels in convolutional neural networks, during the development of new separation measures. Furthermore, while the law of equi-separation has been observed in a variety of vision tasks, it does not appear to hold for language models such as BERT (41) (see Fig. S9 in SI Appendix). This could be due to the fact that language models learn token-level features rather than sentence-level features at each layer. As such, it would be important to study whether the law of equi-separation could be utilized to derive better sentence-level features from a sequence of token-level features.

Perhaps the most pressing question is to reveal the underlying mechanism of this data-separation law. However, some commonly used assumptions in the literature of deep learning theory may not be applicable. For instance, the non-linearity of activation functions is crucial to this law because

the separation fuzziness is roughly unchanged by linear transformations. Additionally, analyses of neural networks using the neural tangent kernel (42) are not able to capture the effect of depth, which is essential for understanding the data separation process across all layers. This suggests that more innovative approaches may be necessary to fully understand the mechanisms behind this law.

Broadly speaking, a perspective informed by this law is to focus on the non-temporal dynamics of the data separation process throughout all layers when studying deep neural networks. This perspective probes into the interior of deep learning models by perceiving the black-box classifier as a sequence of transformations, each of which corresponds to a layer or a block in the case of residual neural networks. In contrast, existing analyses often consider the “exterior” of these models by focusing on the loss function or test performance, without considering the intermediate layers (42–44). Informed by the law of equi-separation, this perspective may provide formal insights into the inner workings of deep learning.

Data Availability. Our code is publicly available at GitHub (<https://github.com/HornHehhf/Equi-Separation>).

ACKNOWLEDGMENTS. We are grateful to X.Y. Han for very helpful comments and feedback on an early version of the manuscript. We also thank Hangyu Lin (<https://github.com/avalonstrel/NeuralCollapse/tree/mlp>), Cheng Shi (<https://github.com/DaDaCheng/Re-equi-sepa>), and Ben Zhou (<https://github.com/Slash0BZ/data-separation/tree/main/reproduction>) for reproducing our experimental results. This research was supported in part by NSF grants CCF-1934876 and CAREER DMS-1847415, and an Alfred Sloan Research Fellowship.

1. DP Kingma, J Ba, Adam: A method for stochastic optimization in *International Conference on Learning Representations*. (2015).
2. A Krizhevsky, I Sutskever, GE Hinton, ImageNet classification with deep convolutional neural networks in *Advances in Neural Information Processing Systems*. Vol. 25. (2012).
3. Y LeCun, Y Bengio, G Hinton, Deep learning. *Nature* **521**, 436–444 (2015).
4. D Silver, et al., Mastering the game of go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
5. A Fawzi, et al., Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature* **610**, 47–53 (2022).
6. T Hastie, R Tibshirani, JH Friedman, *The elements of statistical learning: data mining, inference, and prediction*. (Springer) Vol. 2, (2009).
7. M Hutson, Has artificial intelligence become alchemy? *Science* **360**, 478–478 (2018).
8. PL Bartlett, PM Long, G Lugosi, A Tsigler, Benign overfitting in linear regression. *Proc. Natl. Acad. Sci.* **117**, 30063–30070 (2020).
9. Y Lu, A Zhong, Q Li, B Dong, Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations in *International Conference on Machine Learning*. (PMLR), pp. 3276–3285 (2018).
10. C Fang, H He, Q Long, WJ Su, Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proc. Natl. Acad. Sci.* **118**, e2103091118 (2021).
11. K He, X Zhang, S Ren, J Sun, Deep residual learning for image recognition in *2016 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 770–778 (2016).
12. H Xiao, K Rasul, R Vollgraf, Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).
13. JP Stevens, *Applied multivariate statistics for the social sciences*. (Routledge), (2012).
14. V Papyan, X Han, DL Donoho, Prevalence of neural collapse during the terminal phase of deep learning training. *Proc. Natl. Acad. Sci.* **117**, 24652–24663 (2020).
15. X Han, V Papyan, DL Donoho, Neural collapse under mse loss: Proximity to and dynamics on the central path in *International Conference on Learning Representations*. (2022).
16. K Simonyan, A Zisserman, Very deep convolutional networks for large-scale image recognition in *International Conference on Learning Representations*. (2015).
17. G Huang, Z Liu, L Van Der Maaten, KQ Weinberger, Densely connected convolutional networks in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 4700–4708 (2017).
18. G Alain, Y Bengio, Understanding intermediate layers using linear classifier probes in *International Conference on Learning Representations*. (2017).
19. V Papyan, Traces of class/cross-class structure pervade deep learning spectra. *The J. Mach. Learn. Res.* **21**, 10197–10260 (2020).
20. T Galanti, A György, M Hutter, On the role of neural collapse in transfer learning in *International Conference on Learning Representations*. (2022).
21. T Galanti, On the implicit bias towards minimal depth of deep neural networks. *arXiv preprint arXiv:2202.09028* (2022).

22. I Ben-Shaul, S Dekel, Nearest class-center simplification through intermediate layers in *Topological, Algebraic and Geometric Learning Workshops 2022*. (PMLR), pp. 37–47 (2022).
23. Y Bengio, Learning deep architectures for AI. *Foundations trends Mach. Learn.* **2**, 1–127 (2009).
24. J Schmidhuber, Deep learning in neural networks: An overview. *Neural Networks* **61**, 85–117 (2015).
25. S Hihi, Y Bengio, Hierarchical recurrent neural networks for long-term dependencies in *Advances in Neural Information Processing Systems*, eds. D Touretzky, M Mozer, M Hasselmo. (MIT Press), Vol. 8, (1995).
26. X Glorot, Y Bengio, Understanding the difficulty of training deep feedforward neural networks in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. (PMLR), Vol. 9, pp. 249–256 (2010).
27. D Yarotsky, Error bounds for approximations with deep ReLU networks. *Neural Networks* **94**, 103–114 (2017).
28. Y LeCun, The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/> (1998).
29. A Krizhevsky, Master’s thesis (University of Toronto) (2009).
30. K He, J Sun, Convolutional neural networks at constrained time cost in *2015 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 5353–5360 (2015).
31. RK Srivastava, K Greff, J Schmidhuber, Training very deep networks in *Advances in Neural Information Processing Systems*. Vol. 28. (2015).
32. AG Howard, et al., Mobilenet: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).
33. M Tan, Q Le, EfficientNet: Rethinking model scaling for convolutional neural networks in *Proceedings of the 36th International Conference on Machine Learning*. (PMLR), Vol. 97, pp. 6105–6114 (2019).
34. S Bubeck, M Sellke, A universal law of robustness via isoperimetry. *Adv. Neural Inf. Process. Syst.* **34**, 28811–28822 (2021).
35. S Ioffe, C Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift in *Proceedings of the 32nd International Conference on Machine Learning*. (PMLR), Vol. 37, pp. 448–456 (2015).
36. C Zhang, S Bengio, Y Singer, Are all layers created equal? *J. Mach. Learn. Res.* **23**, 1–28 (2022).
37. WJ Su, Neurashed: A phenomenological model for imitating deep learning training. *arXiv preprint arXiv:2112.09741* (2021).
38. MD Zeiler, R Fergus, Visualizing and understanding convolutional networks in *European Conference on Computer Vision*. pp. 818–833 (2014).
39. I Tenney, D Das, E Pavlick, Bert rediscovers the classical nlp pipeline in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. pp. 4593–4601 (2019).
40. VW Liang, Y Zhang, Y Kwon, S Yeung, JY Zou, Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Adv. Neural Inf. Process. Syst.* **35**, 17612–17625 (2022).
41. J Devlin, MW Chang, K Lee, K Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. pp. 4171–4186 (2019).
42. A Jacot, F Gabriel, C Hongler, Neural tangent kernel: Convergence and generalization in neural networks. *Adv. Neural Inf. Process. Syst.* **31** (2018).
43. H He, WJ Su, The local elasticity of neural networks in *International Conference on Learning Representations*. (2020).
44. M Belkin, D Hsu, S Ma, S Mandal, Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proc. Natl. Acad. Sci.* **116**, 15849–15854 (2019).
45. T Wolf, et al., Transformers: State-of-the-art natural language processing in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. pp. 38–45 (2020).
46. R Socher, et al., Recursive deep models for semantic compositionality over a sentiment treebank in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. pp. 1631–1642 (2013).

Supporting Information

In this appendix, we first detail the general setup for all experiments, then briefly highlight some distinctive settings for figures in the main text, after that mention the argument for robustness, and finally show some additional results. More details can be found in our code.[‡]

General Setup. In this subsection, we detail the general setup that is used for the experiments throughout the paper.

Network architecture. In our experiments, we mainly consider feedforward neural networks (FNNs). Unless otherwise stated, the corresponding hidden size is 100. For simplicity, we use neural networks with 4, 8, 20 layers as representatives for shallow, mid, and deep neural networks. By default, batch normalization (35) is inserted after fully connected layers and before the ReLU nonlinear activation functions.

Dataset. Our experiments are mainly conducted on Fashion-MNIST. Unless otherwise stated, we resize the original images to 10×10 pixels. By default, we randomly sample a total of 1000 training examples evenly distributed over the 10 classes.

Optimization methodology. In our experiments, three popular optimization methods are considered: SGD, SGD with momentum, and Adam. The weight decay is set to 5×10^{-4} , and the momentum term is set to 0.9 for SGD with momentum. The neural networks are trained for 600 epochs with a batch size of 128. The initial learning rate is annealed by a factor of 10 at 1/3 and 2/3 of the 600 epochs. As for SGD and SGD with momentum, we pick the model resulting in the best equi-separation law in the last epoch among the seven learning rates: 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, and 1.0. Similarly, we pick the model resulting in the best equi-separation law in the last epoch for Adam among the five learning rates: 3×10^{-5} , 1×10^{-4} , 3×10^{-4} , 1×10^{-3} , 3×10^{-3} . Unless otherwise stated, Adam is adopted in the experiments.

Detailed Experimental Settings. In this subsection, we show detailed experimental settings for the figures in the main text.

Optimization. As shown in Fig. 1, for FNNs with different depths, three different optimization methods are considered: SGD, SGD with momentum, and Adam.

Visualization. As shown in Fig. 1, we visualize the features of different layers in the 8-layer FNNs trained on Fashion-MNIST. In particular, principal component analysis (PCA) is used to visualize the features in a two-dimensional plane.

Training epoch. As shown in Fig. 2, we show the separability of features among layers in different training epochs. In this part, we simply consider the 20-layer FNNs trained on Fashion-MNIST. The equi-separation law doesn't emerge at the beginning, and gradually emerges during training. After that the decay ratio is decreased along the training process until the network converges. Furthermore, we conducted an analysis of the last-layer features in the aforementioned experiments. As illustrated in Fig. S1, it is evident that the separation fuzziness of the last-layer features does not reach convergence until after 400 epochs. Conversely, the equi-separation law manifests at epoch 100, as demonstrated in Fig. 2, with a Pearson correlation coefficient of -0.996 . This observation indicates that the equi-separation law emerges prior to the occurrence of neural collapse.

As shown in Fig. 3, we further consider the impact of the following factors on the equi-separation law: dataset, learning rate, and class distribution.

Dataset. As shown in Fig. 3, we experiment with CIFAR-10. We use the second channel of the images and resize the original images to 10×10 . Similar to Fashion-MNIST, we randomly sample a total of 1000 training examples evenly distributed over the 10 classes.

Learning rate. As shown in Fig. 3, we compare the 9-layer FNNs trained using SGD with different leaning rates on Fashion-MNIST.

[‡]Our code is publicly available at GitHub (<https://github.com/HornHehf/Equi-Separation>).

Imbalanced data. As shown in Fig. 3, we experiment with imbalanced data. Specifically, we consider 5 majority classes with 500 examples in each class, and 5 minority classes with 100 examples in each class. The law of equi-separation also emerges in neural networks trained with imbalanced Fashion-MNIST data, though its half-life is significantly larger than that in the balanced case. This might be caused by the collapse of minority classifiers as shown in (10).

Convolutional neural networks (CNNs). As shown in Fig. 4, we experiment with two canonical CNNs, AlexNet and VGG (16). We use the PyTorch implementation of both models.[§] We experiment with both Fashion-MNIST and CIFAR-10, and the original images are resized to 32×32 pixels. Given that AlexNet and VGG are designed for ImageNet images with 224×224 pixels, we make some small modifications to handle images with 32×32 pixels. As for AlexNet, we change the original convolution filters to 5×5 convolution filters with padding 2. The color channel number is multiplied by 32 at the beginning. As for VGG, we change the original convolution filters to 5×5 convolution filters with padding 2, and the last average pooling layer before fully connected layers is replaced by an average pooling layer over a 1×1 pixel window. The color channel number is multiplied by 16 at the beginning.

Moreover, we examined VGG employing four distinct configurations[¶] on the Fashion-MNIST and CIFAR-10 datasets. As demonstrated in Fig. S2, an approximate equi-separation law is also observed across different VGG configurations. It is important to note that the equi-separation law can be further refined by modifying convolution filter size and input channel number, as illustrated in Fig. 4.

To gain further insights into CNNs, we explored a variant of VGG11 (referred to as pure CNNs) by retaining all convolutional layers, the final fully connected layer, and removing all pooling layers along with the initial two fully connected layers. Additionally, we ensured an equal number of channels in each layer for our analysis. In this simplified network, convolutional layers serve to extract features, which are subsequently input into the final fully connected layer for classification. As presented in Fig. S3, the equi-separation law is present in pure CNNs for various datasets with an appropriate number of channels; however, the separation fuzziness of the last-layer features is considerably higher than that of FNNs and canonical CNNs (e.g., AlexNet and VGG). We further investigated the influence of other factors in CNNs, such as depth, kernel size, pooling layer, and image size. Our findings indicate that the equi-separation law is indeed present in CNNs but necessitates a delicate balance among these factors. This suggests that an alternative measure, rather than separation fuzziness (D in Eq. 1), may be required to describe the information flow in CNNs, given the unique structure of features following convolutional layers.

Depth. As shown in Fig. 5, besides Fashion-MNIST and CIFAR-10, we also consider MNIST to illustrate the optimal depth for different datasets. Similar to Fashion-MNIST, we resize the original MNIST images to 10×10 pixels, and randomly sample a total of 1000 training examples evenly distributed over the 10 classes.

As shown in Fig. 6, we consider FNNs with different widths and shapes.

Width. As shown in Fig. 6, we consider FNNs with three different widths. All layers have the same width in each setting. For simplicity, we consider the 8-layer FNNs trained on Fashion-MNIST.

Shape. As shown in Fig. 6, we consider three different shapes of FNNs: Narrow-Wide, Wide-Narrow, and Mixed. As for Narrow-Wide, the FNNs have narrow hidden layers at the beginning, and then have wide hidden layers in the later layers. Similarly, we use Wide-Narrow to denote FNNs that start with wide hidden layers followed by narrow hidden layers. In the case of Mixed, the FNNs have more complicated patterns of the widths of hidden layers. For simplicity, we only consider 8-layer FNNs trained on Fashion-MNIST. The corresponding widths of hidden layers for Narrow-Wide, Wide-Narrow, and Mixed we considered in our experiments are as follows:

[§]More details are in <https://github.com/pytorch/vision/tree/main/torchvision/models>.

[¶]More details can be found at: <https://github.com/kuangliu/pytorch-cifar/tree/master/models>.

(100, 100, 100, 100, 1000, 1000, 1000), (1000, 1000, 1000, 1000, 100, 100, 100), and (100, 500, 500, 2500, 2500, 500, 500).

Residual neural networks (ResNets). As shown in Fig. 7, we consider three different types of ResNets: ResNets with 2-layer building blocks, ResNets with 3-layer bottleneck blocks, and ResNets with mixed blocks. For 3-layer blocks, the expansion rate is set to 1 instead of 4, and the original 1×1 and 3×3 convolution filters are replaced by 5×5 convolution filters. For all types of ResNets, the channel number is set to 8, and the color channel number is also multiplied by 8 at the beginning. In this part, the Fashion-MNIST and CIFAR-10 images are resized to 32×32 pixels instead of 10×10 pixels.

Dense Convolutional Network (DenseNet). As shown in Fig. 8, we investigate the law when applied to DenseNet (17), namely DenseNet161.^{||} In this part, the images of Fashion-MNIST and CIFAR-10 are resized to 32×32 pixels, i.e., (3, 32, 32) for CIFAR-10 and (32, 32) for Fashion-MNIST.

Out-of-sample performance. As shown in Fig. 9, given 20-layer FNNs, instead of the standard training, we also consider the following two-stage frozen training procedure: 1) freeze the last 10 layers and train the first 10 layers; 2) freeze the first 10 layers and train the last 10 layers. When we conduct the frozen and standard training on CIFAR-10, we find that the our-of-sample performance of standard training (test accuracy: 23.85%) is better than that of frozen training (test accuracy: 19.67%), even though the two training procedures have similar training losses (standard: 0.0021; frozen: 0.0019) and training accuracies (standard: 100%; frozen: 100%). This is consistent with the equi-separation law: the neural networks trained using standard training have much clearer equi-separation law compared to the neural networks trained using frozen training. Specifically, the corresponding Pearson correlation coefficients for standard and frozen training are -0.997 and -0.971 , respectively.

Additional Results. In this subsection, we provide some additional results for the equi-separation law.

Image size. As depicted in Fig. S4, we examine FNNs on images with a resolution of 32×32 pixels, specifically (3, 32, 32) for CIFAR-10 and (32, 32) for Fashion-MNIST. It is important to note that Fashion-MNIST, despite its original size of (28, 28), is frequently resized to (32, 32) in the same manner as MNIST. In this part, we set the width as 1000 instead of 100 for FNNs, given that the images with 32×32 pixel resolution possess a higher dimension in comparison to those with 10×10 pixel resolution.

Sample size. As shown in Fig. S5, we experiment with different number of randomly sampled training examples. For each setting, we have the same number of examples for each of the 10 classes. For simplicity, we only consider the 8-layer FNNs trained on Fashion-MNIST. The law of equi-separation emerges in neural networks trained with different number of training examples, though the half-life can be larger for larger datasets due to the increased optimization difficulty.

Class-wise data separation. As shown in Fig. S6, we further consider separation fuzziness for each class, i.e., $\text{SS}_w^k \text{SS}_b^+$, where SS_w^k indicates the within-class covariance for Class k . For simplicity, we only consider the 8-layer FNNs trained on Fashion-MNIST. The equi-separation law emerges in each class. Even though the equi-separation law can be a little noisy for some classes, the noise is reduced when we consider the separation fuzziness for all classes.

Training dynamics. As shown in Fig. S7, we consider the convergence rates of different layers (without the last-layer classifier) with respect to the relative improvement $\frac{D_{l+1}}{D_l}$ of separability of features. For simplicity, we consider the 8-layer FNNs trained on CIFAR-10 in this part. The relative improvement of bottom layers converges earlier compared to those of top layers.

Test data. As shown in Fig. S8, we further show how features separate across layers in the test data. For simplicity, we only consider the 8-layer FNNs trained on Fashion-MNIST here. A fuzzy version of the equi-separation law also exists in the test data.

BERT. As shown in Fig. S9, we experiment with BERT features. We use the pretrained case-sensitive BERT-base PyTorch implementation (45), and the common hyperparameters for BERT. Specifically, we fine-tuned the pretrained BERT model on a binary sentiment classification task (SST-2) (46), where we randomly sampled 500 sentences for each class. Given a sequence of token-level BERT features, two most popular approaches are used to get the sentence-level features at each layer: 1) using the features of the first token^{**} (i.e., the [CLS] token); 2) averaging the features among all tokens in the sentence.

Batch normalization. As shown in Fig. S10, it is difficult to optimize a neural network without batch normalization, let alone achieve the law of equi-separation. Even when the network is well-optimized, the law is often not clear.

Pretraining. As shown in Fig. S11, we consider the impact of the pretraining on the equi-separation law. Specifically, we first pretrained the 20-layer FNNs on FashionMNIST as shown in Fig. S11 (Depth=20). After that we fix the first 10 layers of the pretrained model and train additional 5 layers as in Fig. S11 (Pretraining), which is quite different from training 15-layer FNNs from scratch as in Fig. S11 (Depth=15). At the same time, the equi-separation law also emerges in the additional 5 layers in the pretraining setting.

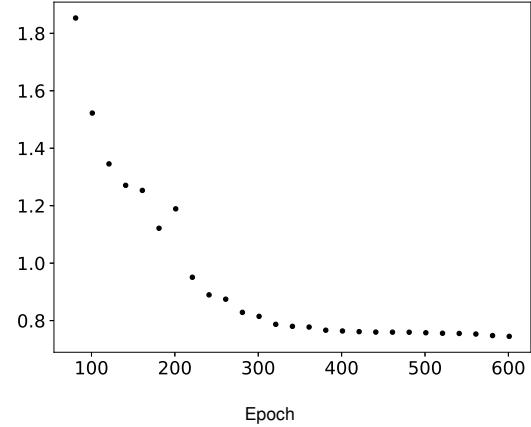


Fig. S1. Separation fuzziness of last-layer features throughout the training process. The x axis represents the epoch number, while the y axis indicates the separation fuzziness (D in Eq. 1). The x axis commences from approximately 100 epochs due to the exceedingly high initial separation fuzziness, which would otherwise mask discernible trends.

^{||}Further details can be found at <https://github.com/kuangliu/pytorch-cifar/tree/master/models>.

^{**}The input layer is not considered here since the inputs of the first token do not take other tokens into account.

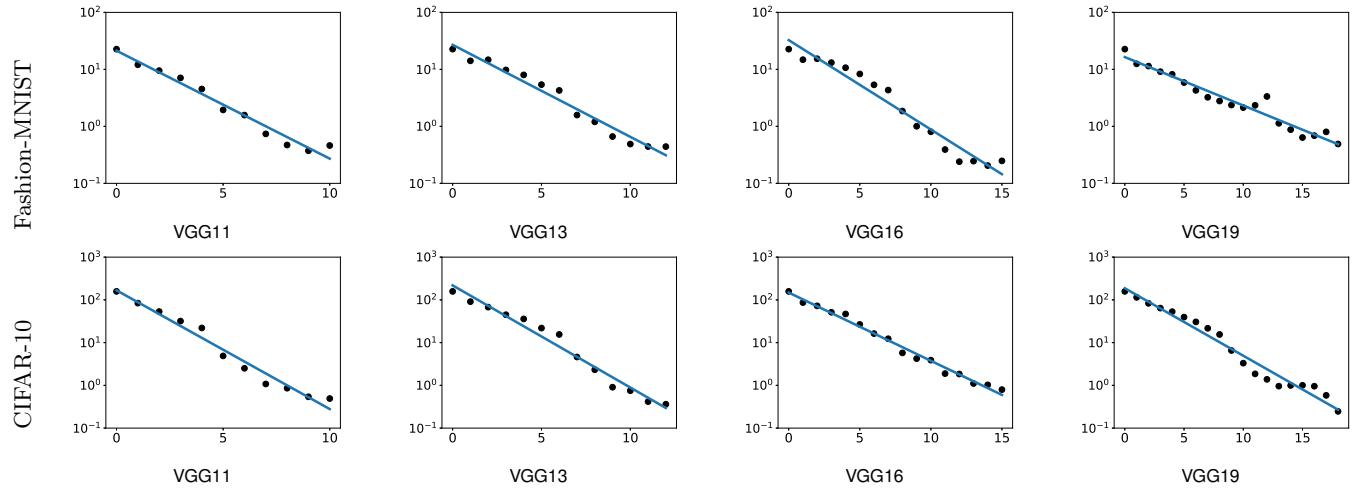


Fig. S2. Emergence of a **fuzzy** equi-separation law in VGG networks with varied depths on Fashion-MNIST and CIFAR-10.

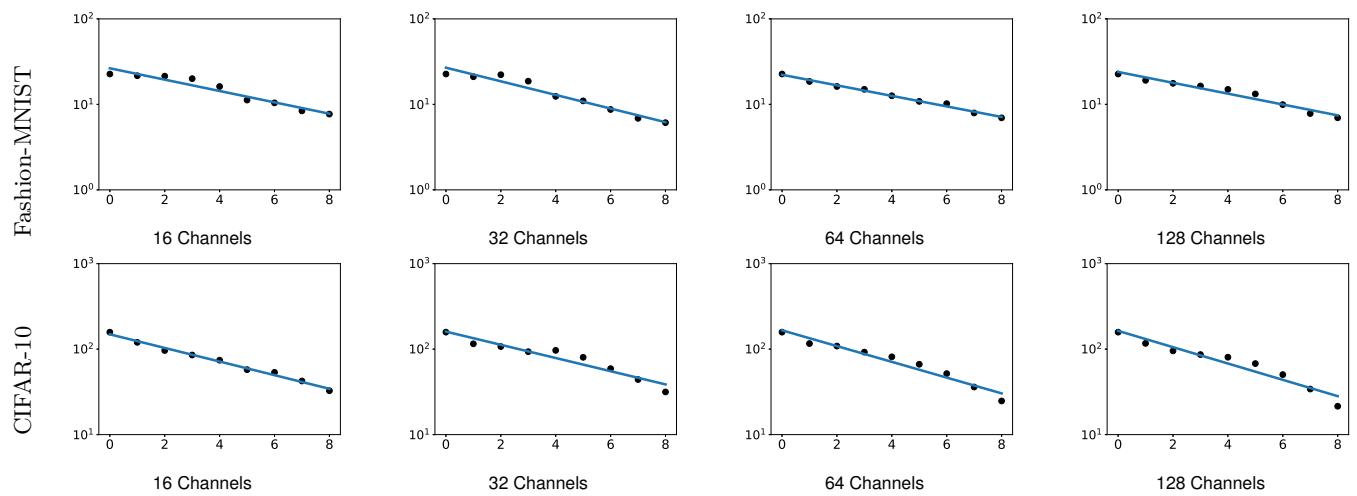


Fig. S3. Equi-separation law emerges in **pure CNNs** with appropriate numbers of channels on Fashion-MNIST (64 channels) and CIFAR-10 (16 channels).

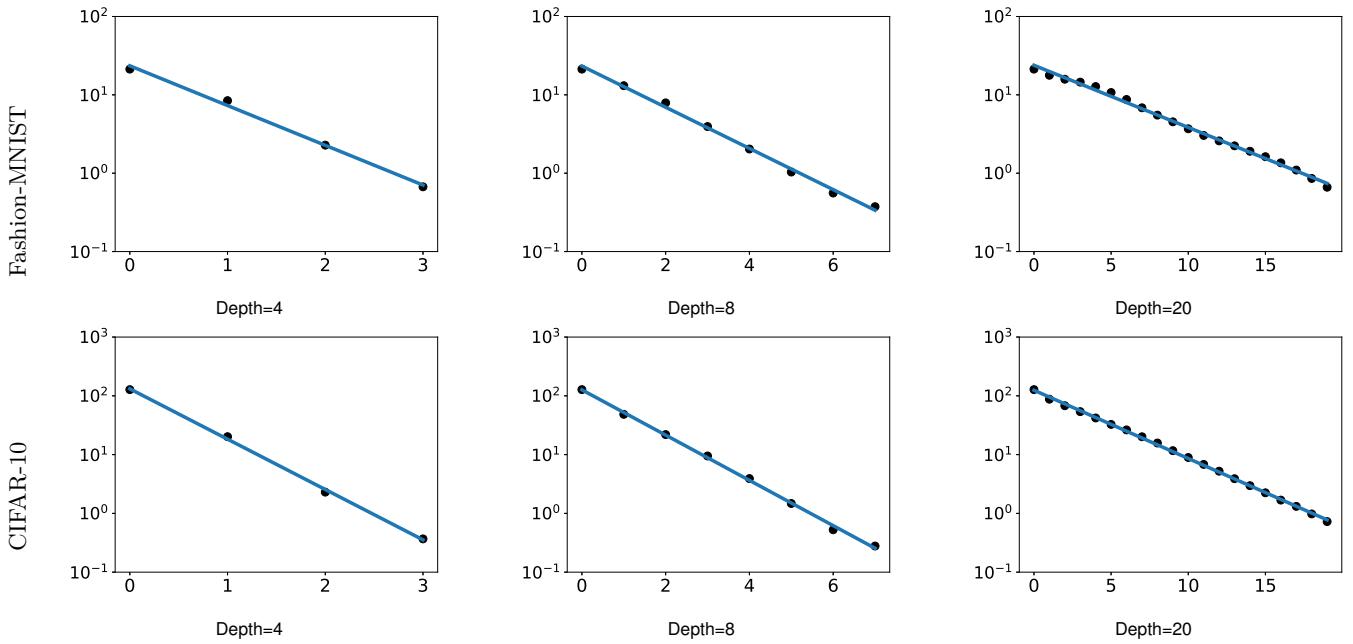


Fig. S4. Equi-separation law emerges in FNNs across varied depths on 32×32 pixel images of Fashion-MNIST and CIFAR-10.

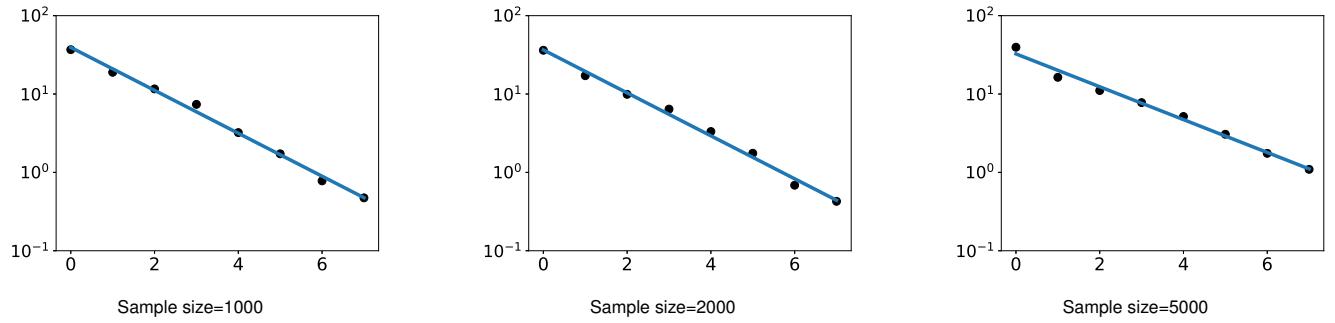


Fig. S5. The impact of the sample size on the law of equi-separation.

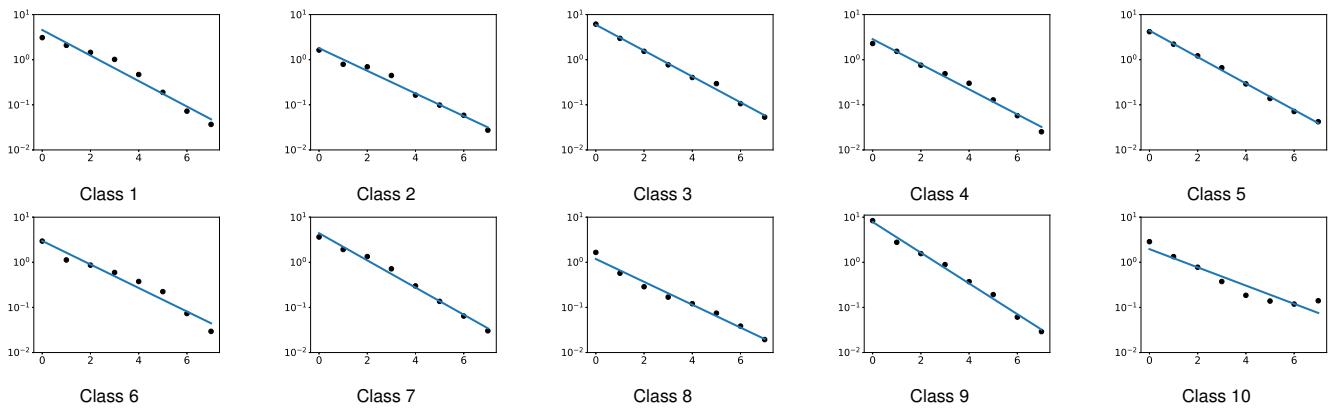


Fig. S6. The law of equi-separation emerges for each class.

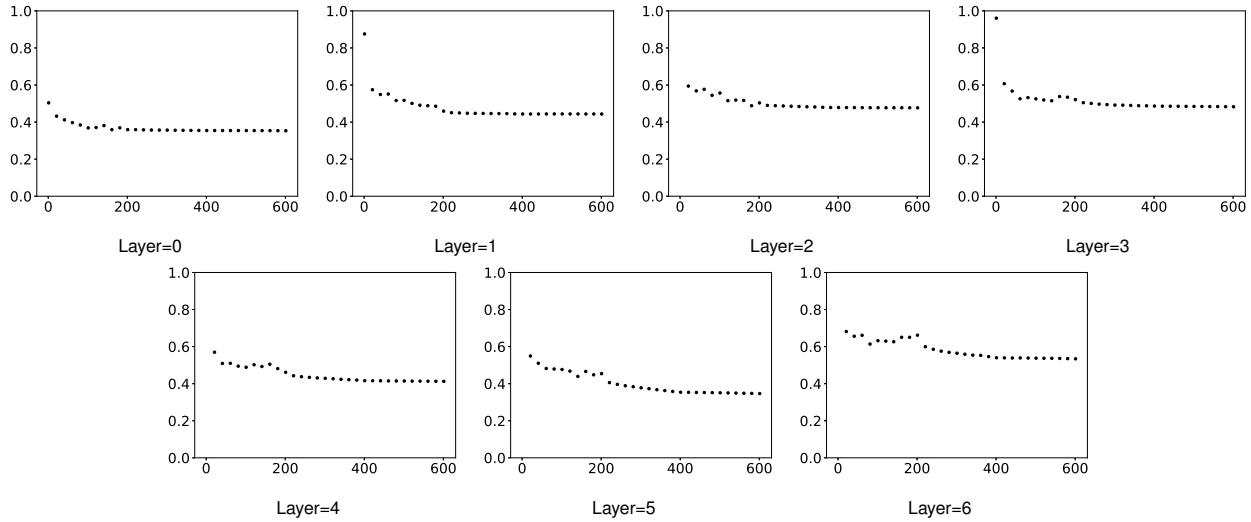


Fig. S7. Convergence rates of different layers (without the last-layer classifier) with respect to the relative improvement ($\frac{D_{l+1}}{D_l}$) of separability of features. The x axis represents the training epoch, while the y -axis indicates the relative improvement ($\frac{D_{l+1}}{D_l}$).

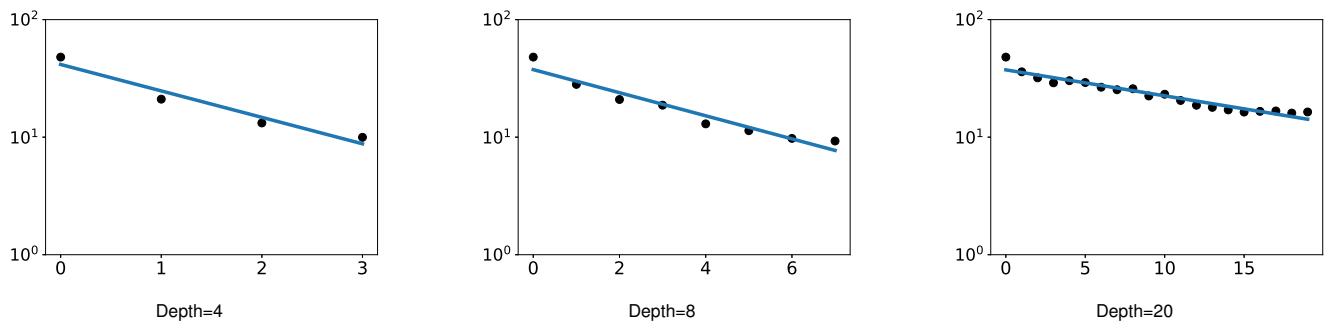


Fig. S8. A fuzzy version of the equi-separation law also emerges in the test data.

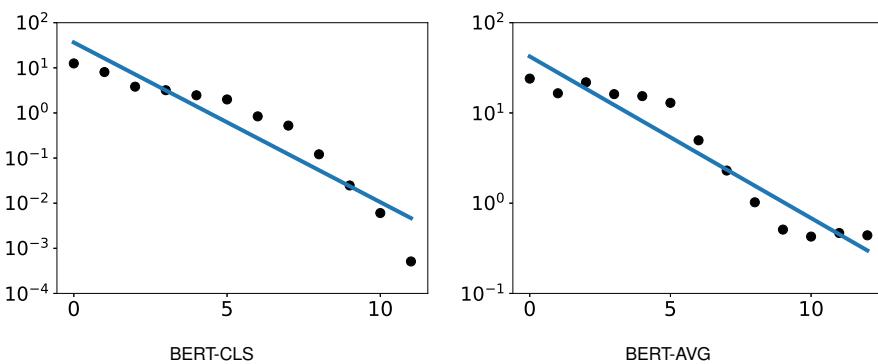


Fig. S9. The equi-separation law does not exist in BERT features. Given a sequence of token-level BERT features, two most popular approaches are used to get the sentence-level features at each layer: 1) using the features of the first token (i.e., the [CLS] token); 2) averaging the features among all tokens in the sentence (denoted as BERT-AVG).

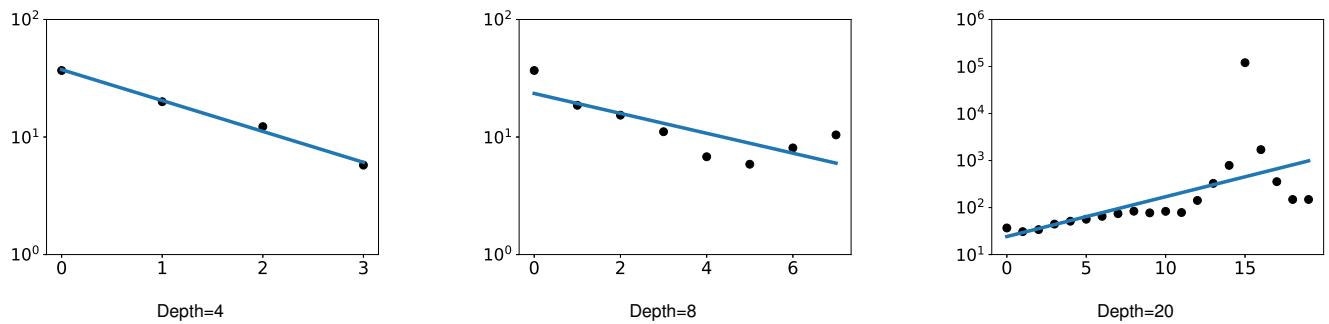


Fig. S10. The equi-separation law is not clear on Fashion-MNIST for FNNs trained without batch normalization.

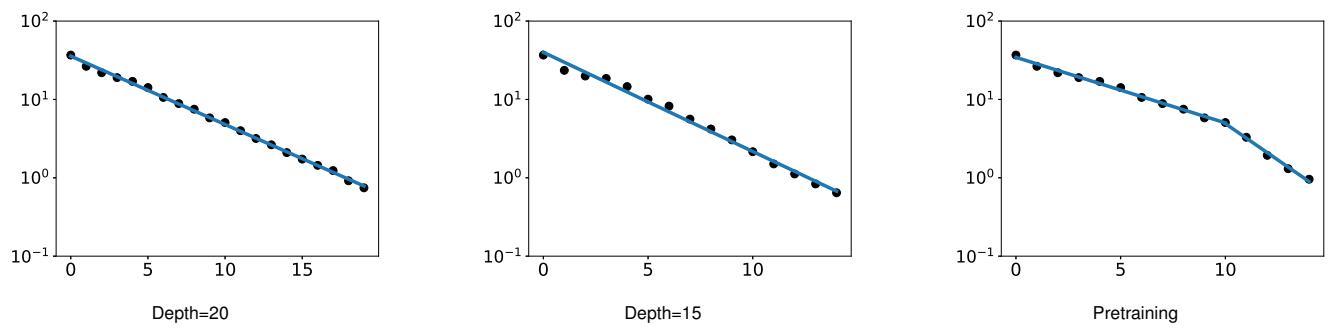


Fig. S11. The impact of pretraining on the equi-separation law. We first pretrained the 20-layer FNNs as shown in Fig. S11 (Depth=20). After that we fix the first 10 layers of the pretrained model and train additional 5 layers as in Fig. S11 (Pretraining), which is quite different from training the 15-layer neural networks from scratch as in Fig. S11 (Depth=15).