HiLD 2024: 2nd Workshop on High-dimensional Learning Dynamics

Random matrix theory analysis of neural network weight matrices

Matthias Thamm THAMM@ITP.UNI-LEIPZIG.DE
Leipzig University, Germany

Max Staats STAATS@ITP.UNI-LEIPZIG.DE
Leipzig University, Germany

Bernd Rosenow ROSENOW@PHYSIK.UNI-LEIPZIG.DE
Leipzig University, Germany

Abstract

As neural network weight matrices are initialized randomly, they conform precisely to random matrix theory (RMT) predictions before training. Post-training, deviations from RMT predictions indicate task-specific information encoded in the weights. We analyze feedforward and convolutional neural network weights trained on image recognition tasks. We demonstrate that most of the weights' singular values follow universal RMT predictions even after training, suggesting that major parts of weights remain random. By comparing singular value spectra with the Marchenko-Pastur distribution and singular vector entries with the Porter-Thomas distribution, we identify significant deviations only in the parts associated with the largest singular values. We argue that a comparison to RMT predictions allows locating learned information in the weights. In addition, the RMT analysis enables us to differentiate between networks trained within various learning regimes.

1. Introduction

Neural networks are often highly over-parametrized [1–12] and capable of memorizing large amounts of random training data [13, 14]. Traditional "bias-variance tradeoff" [15] suggests that such networks should overfit and fail with unseen data. However, they exhibit a double descent behavior [16–18] relative to the number of parameters, performing well even in the over-parametrized limit. This apparent contradiction is addressed by evidence showing that ultra-wide neural networks are biased towards simple functions [19–22].

To analyze these networks, we employ random matrix theory (RMT) [23–27] as a zero-information hypothesis, where deviations from RMT indicate system-specific information. RMT has been useful for studying systems with inherent randomness, such as nuclear spectra [24, 26–28], stock market correlations [29–31, 31–35], and biological networks [36, 37].

RMT has previously been applied to estimate the asymptotic performance of single-layer networks [38, 39] and to analyze the generalization dynamics of linear networks [40]. Outliers and the random part of pre-activation covariance matrices were examined in [41], while other studies focused on the spectra of Jacobians at initialization [42] and the eigenvalue distribution of the Hessian of the loss matrix [43, 44]. The spectral evolution of weight matrices during training was analyzed in [45], assessing the quality of pretrained DNNs [45, 46] by computing spectral norms of weight matrices and fitting the exponent of a power law tail of the singular value spectrum.

Here, we use various RMT tools to demonstrate that the weight matrices of deep, overparameterized neural networks are predominantly random. By comparing the singular values of several DNNs with the Wigner nearest neighbor spacing distribution, we find that the bulk of the spectrum aligns with RMT predictions. This is further corroborated by analyzing the number variance of the singular

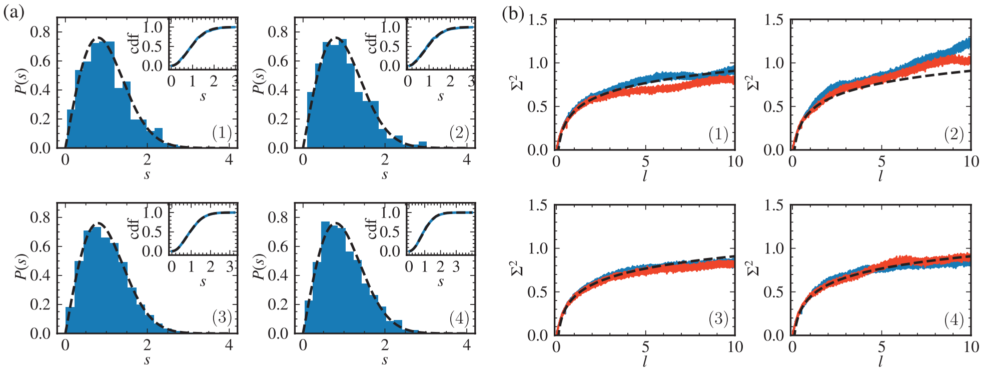

Figure 1. Nearest neighbor spacing distributions (a) and level number variance (b) of unfolded singular values of weights for various neural networks. The RMT predictions are depicted by dashed, black lines. The insets in (a) depict the cumulative distributions. Subpanels (1) show results for the second hidden layer weight matrix of MLP1024, (2) the second convolutional layer in the CNN miniAlexNet, (3) the second fully connected layer in AlexNet, and (4) for the third dense layer in VGG19. In all cases there is excellent visual agreement with the RMT predictions. For the level spacings this is further supported by Kolmogorov-Smirnov tests which cannot reject the null hypothesis at a significance level of (a1) 81%, (a2) 85%, (a3) 31%, and (a4) 96%.

value spectra of weight matrices. We investigate the hypothesis that a large fraction of singular values does not encode information by comparing the distribution of eigenvector components to the Porter-Thomas distribution. Significant deviations from the Porter-Thomas distribution are found only in a small fraction of eigenvectors with large singular values, indicating that learned information is encoded in them.

Additionally, we train networks across different learning regimes, from lazy networks where weights barely change during training to rich networks where final weights significantly differ from initial ones [47–50]. Networks trained in the lazy regime adhere closely to RMT predictions, unlike those trained in rich and intermediate regimes. Consequently, the weight spectrum and the comparison of singular vector entries to the Porter-Thomas distribution can distinguish between learning regimes, with the best generalization performance found between the two extremes.

The results presented here are a concise summary of Ref. [51], which includes additional findings and detailed information.

2. Experimental setup

In this study, we consider the singular value decomposition of a weight matrix W defined via $W = USV^T$, where U and V are orthogonal matrices, and S is a diagonal matrix of non-negative entries ν_i on its diagonal, the so-called singular values. While this is straightforward for dense layers, in the case of convolutional layers we first reshape the four dimensional weight tensors to a rectangular shape and then compute their singular values and vectors.

We consider several networks for image recognition with different architectures and sizes: a) a fully connected feedforward network with layers of size [3072, 1024, 512, 512, 10] denoted as MLP1024 and b) a convolutional network called miniAlexNet consisting of two convolutional layers followed by three dense layers, both trained on the CIFAR-10 [10] dataset. In addition, we analyze the two larger networks c) AlexNet [52] and d) VGG19 [11], whose models trained on the ImageNet [53] dataset are available via `pytorch` [54] and `tensorflow` [55], respectively. The network weights are initialized using a Glorot uniform distribution [56] and then trained with stochastic gradient descent and cross-entropy loss. More details on the networks can be found in Appendix A.

3. Universal random matrix theory properties

We first consider universal properties that do only depend on the class of random matrix, not on a specific realization. These properties can be formulated for the unfolded singular value spectrum. Here, unfolding refers to normalizing the mean density of states of the singular values ν_i to unity, yielding the unfolded spectrum ξ_i [23–27]. For real random matrices in the universality class of the Gaussian orthogonal ensemble (GOE), the level spacings $s_k = \xi_{k+1} - \xi_k$, i.e. the differences between neighboring unfolded singular values, are distributed according to the Wigner surmise [23–28]

$$P_{\mathrm{GOE}}(s) = \frac{\pi s}{2} \exp\left(-\frac{\pi}{4}\, s^2\right) \,. \tag{1}$$

The nearest neighbor spacings of the weight matrix singular values are in excellent agreement with the RMT prediction Eq. (1) before and after training the networks (Fig. 1a). This is supported by Kolmogorov-Smirnov tests of the empirical data against Eq. (1) that cannot reject the null hypothesis even at a significance level as high as $\alpha = 0.30$ (for specific p-values, see Appendix B).

Another prediction of RMT that allows to test the random nature of weight matrices is the level number variance, which is sensitive to long range correlations in the spectrum. The number variance describes fluctuations in the number of unfolded singular values $N_{\xi_i}(l)$ in intervals of length l around each singular value ξ_i: $\Sigma^2(l) = \langle (N_\xi(l) - l)^2 \rangle_\xi$. For random matrices from the GOE universality class, the level number variance depends on the interval width ℓ according to $\Sigma^2(l) \propto \ln(2\pi l)$ in the regime $l \gtrsim 1$ [24–27] in excellent agreement with the data (see Fig. 1b).

4. Distribution of singular values and singular vector entries

To locate the learned information in the weights, we compare the agreement with non-universal RMT predictions between initial and trained matrices. For a random $n \times m$ matrix of zero mean and variance σ, the singular values follow the Marchenko-Pastur distribution [57–59, 59],

$$P(\nu) = \begin{cases} \frac{n/m}{\pi \tilde{\sigma}^2 \nu} \sqrt{(\nu_{\max}^2 - \nu^2)(\nu^2 - \nu_{\min}^2)} & \nu \in [\nu_{\min}, \nu_{\max}] \\ 0 & \text{else} \end{cases} \tag{2}$$

where $\nu_{\substack{\max\\\min}} = \tilde{\sigma}(1 \pm \sqrt{m/n})$ and $\tilde{\sigma} = \sigma\sqrt{n}$. We assume without loss of generality that $m \leq n$. While the distribution Eq. (2) describes the spectrum of the weights of untrained networks, the trained weight deviates from the Marchenko-Pastur law (see [45]). In the absence of a microscopic theory for the spectrum of a trained weight matrix, we estimate its random part by fitting the empirical spectra with a modified Marchenko-Pastur law: setting $\nu_{\min}$ to the smallest empirical eigenvalue, and using $\nu_{\max}$ and σ^2 as fit parameters. This introduces an additional free parameter compared to the strict Marchenko-Pastur distribution, representing the percentage of the spectrum still following the Marchenko-Pastur law (see Fig. 2a). We also consider the normalized eigenvectors of the $m \times m$ matrix $W^\dagger W$ (right singular vectors of W), whose components for a random matrix follow the

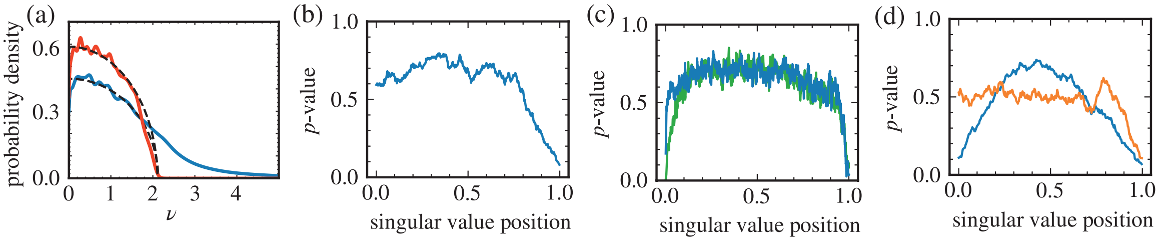

Figure 2. (a) Spectral distributions for initial weight (red), trained weight of the second hidden layer of a MLP512 network (blue) and RMT prediction (black). (b-d) Analysis of the eigenvectors of $W^{\dagger}W$. Here, we show p-values of Kolmogorov-Smirnov tests of the eigenvector entries versus a Gaussian distribution. All results are averaged over neighboring eigenvectors with a window size of 15. The x-direction describes the position of rank ordered singular values, such that 0 corresponds to the smallest and 1 to the largest singular value of each weight matrix. We show results for (b) the second hidden layer of MLP1024, (c) the first dense layer of the large pretrained DNNs AlexNet (blue) and VGG19 (green), and (d) the second convolutional layer (blue) and first dense layer (orange) of the CNN miniAlexNet.

Porter-Thomas distribution [24, 26, 27], i.e., a Gaussian distribution with mean zero and standard deviation $1/\sqrt{m}$. To check if the observation that most singular values of trained networks are random extends to the eigenvectors, we test the empirical distribution of each eigenvector's entries against this Gaussian using a Kolmogorov-Smirnov test. A large p-value means the Gaussian hypothesis cannot be rejected, indicating the vector contains only noise, while a small p-value suggests stored information. To reveal trends, we average the p-values over neighboring singular values with a window size of 15. We find that most eigenvectors are random, especially those corresponding to small singular values (Fig. 2b-d). For large singular values, the p-values decrease, indicating stored information, consistent with [45].

5. RMT analysis of different learning regimes

It was shown [14, 60–64] that neural networks can achieve good generalization accuracies even when their weights change only by very small amounts during training. The opposite to this *lazy learning* is denoted as *rich learning*, where the final weights $\boldsymbol{W}$ after training deviate significantly from the initial ones $\boldsymbol{W}_0$. We train several MLP512 networks, where laziness is controlled by introducing a hyperparameter α that modifies the output activations via [48] $a_L = \mathrm{softmax}\left(\alpha(W_{L-1}\boldsymbol{a}_{L-1} + \boldsymbol{b}_L)\right)$ and the cost function as

$$l(\boldsymbol{W}, \boldsymbol{b}) = -\frac{1}{N\alpha^2}\sum_{k=1}^{N} \boldsymbol{y}^{(k)} \cdot \ln(\boldsymbol{a}_{\mathrm{out}}^{(k)}) \ . \tag{3}$$

Here, a large $\alpha > 1$ scales down the gradient updates and therefore encourages lazy learning, while small $\alpha < 1$ steers training towards the rich learning regime [48]. We denote $\alpha = 1$ as *typical learning*. A comparison of the RMT analysis in the three regimes—rich ($\alpha = 0.5$), typical ($\alpha = 1$), and lazy ($\alpha = 5$)—is illustrated in Fig. 3. For all networks, the bulk spectra exhibit random characteristics, with level spacings (panel b) and level number variance (panel c) closely aligning with RMT predictions. Notably, the level number variances do not indicate slower growth for networks with larger α. In the rich network, there are more large singular values compared to the

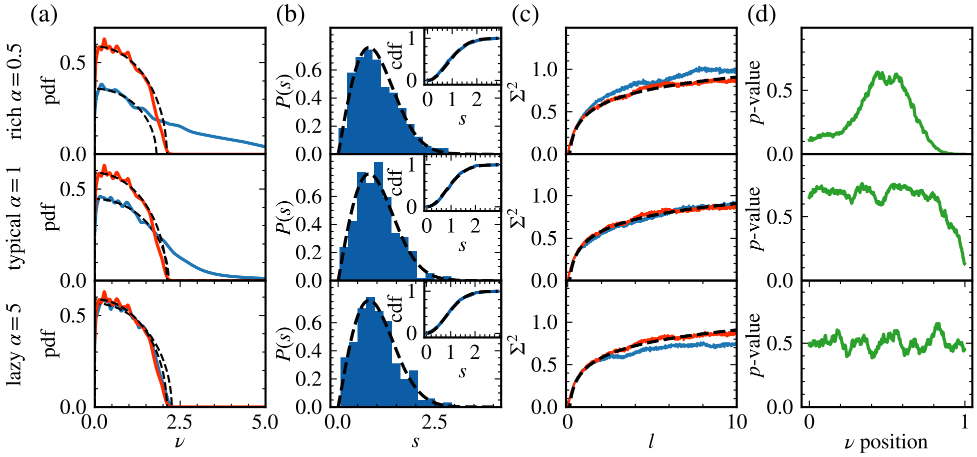

Figure 3. Random matrix theory analysis of second layer weights of MLP512 networks trained in different learning regimes: rich learning ($\alpha = 0.5$, top panel), typical learning ($\alpha = 1$, middle panel), and lazy learning ($\alpha = 5$, bottom panel). We show (a) the spectra for trained (blue) and randomly initialized networks (red) with fits of modified Marchenko-Pastur laws (dashed, black), (b) unfolded level spacing distributions (main panel, blue, window size 15) and corresponding cumulative distributions (insets) with the Wigner surmise (dashed, black), (c) unfolded level number variance (trained: blue, initialized: red), and (d) p-values for comparing singular vector entries to a Porter-Thomas distribution. Trained networks in all cases follow universal RMT predictions (b and c), indicating a random bulk. Lazy networks can be distinguished from typical and rich networks by the spectral distributions in (a) and p-values in (d) [51].

typical network, while the lazy network's Marchenko-Pastur spectrum remains almost unchanged (panel a). Despite this, the lazy network achieves a test accuracy of 50.4% on CIFAR-10, compared to 52.7% for the rich network and 55.2% for the typical network.

When examining the p-values for Kolmogorov-Smirnov tests of eigenvector entries against a Porter-Thomas distribution (panel d), it is observed that in the typical case, small p-values occur only for large singular values. In contrast, the rich network shows small p-values for vectors corresponding to the smallest singular values as well. For the lazy network, all p-values fluctuate around 0.5, consistent with the behavior expected for random weights.

6. Conclusion

We employed RMT as a zero-information hypothesis to distinguish randomness from learned information during training. At initialization, weight matrices perfectly align with RMT predictions, with singular value spectra following the Marchenko-Pastur distribution, singular vector entries obeying the Porter-Thomas distribution, and level spacing adhering to the Wigner surmise. A comparison between initialized and trained networks reveals where information is stored in the weight matrices. Even after training, much of the eigenvalue spectrum remains random, and the spectral statistics continues to match RMT predictions. Singular vectors are mostly random, except for those associated with the largest singular values, indicating that learned information is concentrated in these vectors. Separating random parts from information may allow improved network compression algorithms to be designed.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. ISSN 0018-9219.

[2] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object Recognition with Gradient-Based Learning. In *Shape, Contour and Grouping in Computer Vision*, pages 319–345. Springer, Berlin, Heidelberg, 1999.

[3] Razvan Pascanu, Yann N. Dauphin, Surya Ganguli, and Yoshua Bengio. On the saddle point problem for non-convex optimization. *preprint arXiv:1405.4604*, 2014.

[4] Yann Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *preprint arXiv:1406.2572*, 2014.

[5] Ian J. Goodfellow, Oriol Vinyals, and Andrew M. Saxe. Qualitatively characterizing neural network optimization problems. *preprint arXiv:1412.6544*, 2015.

[6] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring Generalization in Deep Learning. *preprint arXiv:1706.08947*, 2017.

[7] Daniel Soudry and Elad Hoffer. Exponentially vanishing sub-optimal local minima in multilayer neural networks. *preprint arXiv:1702.05777*, 2017.

[8] Siyuan Ma, Raef Bassily, and Mikhail Belkin. The Power of Interpolation: Understanding the Effectiveness of SGD in Modern Over-parametrized Learning. *International Conference on Machine Learning*, pages 3325–3334, 2018. ISSN 2640-3498.

[9] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

[10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

[11] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *preprint arXiv:1409.1556*, 2014.

[12] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling Vision Transformers. *preprint arXiv:2106.04560*, 2021.

[13] Jake Lever, Martin Krzywinski, and Naomi Altman. Points of significance: model selection and overfitting. *Nature methods*, 13(9):703–705, 2016.

[14] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115, 2021. ISSN 0001-0782.

[15] Stuart Geman, Elie Bienenstock, and René Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1):1–58, 1992. ISSN 0899-7667.

[16] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. ISSN 1091-6490.

[17] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *preprint arXiv:1912.02292*, 2019.

[18] Mikhail Belkin, Daniel Hsu, and Ji Xu. Two Models of Double Descent for Weak Features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020.

[19] Giacomo De Palma, Bobak Toussi Kiani, and Seth Lloyd. Random deep neural networks are biased towards simple functions. *preprint arXiv:1812.10156*, 2018.

[20] Guillermo Valle-Perez, Chico Q Camargo, and Ard A Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *preprint arXiv:1805.08522*, 2018.

[21] Yasaman Bahri, Jonathan Kadmon, Jeffrey Pennington, Sam S Schoenholz, Jascha Sohl-Dickstein, and Surya Ganguli. Statistical mechanics of deep learning. *Annual Review of Condensed Matter Physics*, 11:501–528, 2020.

[22] Omry Cohen, Or Malka, and Zohar Ringel. Learning curves for overparametrized deep neural networks: A field theory perspective. *Physical Review Research*, 3(2):023034, 2021.

[23] T. A. Brody, J. Flores, J. B. French, P. A. Mello, A. Pandey, and S. S. M. Wong. Random-matrix physics: spectrum and strength fluctuations. *Reviews of Modern Physics*, 53(3):385–479, 1981.

[24] Thomas Guhr, Axel Müller-Groeling, and Hans A. Weidenmüller. Random-matrix theories in quantum physics: common concepts. *Physics Reports*, 299(4-6):189–425, 1998. ISSN 03701573.

[25] M. L. Mehta. *Random matrices*, volume v. 142 of *Pure and applied mathematics*. Elsevier/Academic Press, Amsterdam and San Diego, CA, 3rd ed. edition, 2004. ISBN 0120884097.

[26] Thomas Papenbrock and Hans A Weidenmüller. Colloquium: Random matrices and chaos in nuclear spectra. *Reviews of Modern Physics*, 79(3):997, 2007.

[27] H. A. Weidenmüller and G. E. Mitchell. Random matrices and chaos in nuclear physics: Nuclear structure. *Reviews of Modern Physics*, 81(2):539, 2009.

[28] Eugene P Wigner. On the statistical distribution of the widths and spacings of nuclear resonance levels. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 47, pages 790–798. Cambridge University Press, 1951.

[29] Vasiliki Plerou, Parameswaran Gopikrishnan, Bernd Rosenow, L. A. N. Amaral, and H. Eugene Stanley. Universal and nonuniversal properties of cross correlations in financial time series. *Physical Review Letters*, 83:1471–1474, Aug 1999.

[30] Laurent Laloux, Pierre Cizeau, Jean-Philippe Bouchaud, and Marc Potters. Noise Dressing of Financial Correlation Matrices. *Physical Review Letters*, 83(7):1467–1470, 1999.

[31] Laurent Laloux, Pierre Cizeau, Marc Potters, and Jean-Philippe Bouchaud. Random Matrix Theory and Financial Correlations. *International Journal of Theoretical and Applied Finance*, 03(03):391–397, 2000. ISSN 0219-0249.

[32] Vasiliki Plerou, Parameswaran Gopikrishnan, Bernd Rosenow, L. A. N. Amaral, Thomas Guhr, and H. Eugene Stanley. Random matrix approach to cross correlations in financial data. *Physical Review E*, 65(6 Pt 2):066126, 2002. ISSN 1539-3755.

[33] B Rosenow, V Plerou, P Gopikrishnan, and H. E Stanley. Portfolio optimization and the random magnet problem. *Europhysics Letters (EPL)*, 59(4):500–506, aug 2002. doi: 10.1209/epl/i2002-00135-4.

[34] Rudi Schäfer, Nils Fredrik Nilsson, and Thomas Guhr. Power mapping with dynamical adjustment for improved portfolio optimization. *Quantitative Finance*, 10(1):107–119, 2010. doi: 10.1080/14697680902748498.

[35] Joel Bun, Jean-Philippe Bouchaud, and Marc Potters. Cleaning large correlation matrices: Tools from random matrix theory. *Physics Reports*, 666:1–109, 2017. ISSN 0370-1573.

[36] Feng Luo, Yunfeng Yang, Jianxin Zhong, Haichun Gao, Latifur Khan, Dorothea K. Thompson, and Jizhong Zhou. Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory. *BMC Bioinformatics*, 8(1):299, 2007. doi: 10.1186/1471-2105-8-299.

[37] Ye Deng, Yi-Huei Jiang, Yunfeng Yang, Zhili He, Feng Luo, and Jizhong Zhou. Molecular ecological network analyses. *BMC Bioinformatics*, 13(1):113, 2012. doi: 10.1186/1471-2105-13-113.

[38] Cosme Louart, Zhenyu Liao, and Romain Couillet. A random matrix approach to neural networks. *The Annals of Applied Probability*, 28(2):1190 – 1248, 2018. doi: 10.1214/17-AAP1328.

[39] Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124005, dec 2019. doi: 10.1088/1742-5468/ab3bc3.

[40] Andrew K. Lampinen and Surya Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. *preprint arXiv:1809.10374*, 2018.

[41] Inbar Seroussi and Zohar Ringel. Separation of scales and a thermodynamic description of feature learning in some cnns. *preprint arXiv:2112.15383*, 2021.

[42] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. The emergence of spectral universality in deep networks. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1924–1932. PMLR, 09–11 Apr 2018.

[43] Nicholas P. Baskerville, Diego Granziol, and Jonathan P. Keating. Applicability of Random Matrix Theory in Deep Learning. *preprint arXiv:2102.06740*.

[44] Diego Granziol. Beyond random matrix theory for deep networks. *preprint arXiv:2006.07721*, 2020.

[45] Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

[46] Charles H. Martin, Tongsu (Serena) Peng, and Michael W. Mahoney. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 12(1):4122, 2021. doi: 10.1038/s41467-021-24025-8.

[47] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *preprint arXiv:1806.07572*, 2018.

[48] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

[49] Greg Yang and Edward J Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

[50] Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33:5850–5861, 2020.

[51] Matthias Thamm, Max Staats, and Bernd Rosenow. Random matrix analysis of deep neural network weight matrices. *Physical Review E*, 106(5):054124, 2022.

[52] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. ISSN 0001-0782.

[53] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[54] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[55] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever,

Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[56] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010. ISSN 1938-7228.

[57] V. A. Marčenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 1967. ISSN 0025-5734.

[58] Anirvan M Sengupta and Partha P Mitra. Distributions of singular values for some random matrices. *Physical Review E*, 60(3):3389, 1999.

[59] L Denby and CL Mallows. Computing sciences and statistics: Proceedings of the 23rd symposium on the interface, edited by em keramidas. *Interface Foundation, Fairfax Station, VA*, pages 54–57, 1991.

[60] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

[61] Chong Li and CJ Shi. Constrained optimization based low-rank approximation of deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 732–747, 2018.

[62] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019.

[63] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

[64] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine learning*, 109(3):467–492, 2020.

[65] M. Brack, Jens Damgaard, A. S. Jensen, H. C. Pauli, V. M. Strutinsky, and C. Y. Wong. Funny Hills: The Shell-Correction Approach to Nuclear Shell Effects and Its Applications to the Fission Process. *Reviews of Modern Physics*, 44(2):320–405, 1972.

[66] H. Bruus and J.-C. Anglès d'Auriac. The spectrum of the two-dimensional Hubbard model at low filling. *Europhysics Letters (EPL)*, 35(5):321–326, 1996. ISSN 0295-5075.

[67] Roman Novak, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A. Alemi, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in python. In *International Conference on Learning Representations*, 2020.

Appendix A. Network architectures and performance

Table 1. Neural network architectures and performance of trained networks. We use d to indicate a dense layer, c for a convolutional layer, p for max pooling, f for flattening, rc for random crop layer, and r for response normalization layer (with a depth radius of 5, a bias of 1, $\alpha = 1$, and $\beta = 0.5$).

	NETWORK	DATASET	TEST ACC
i)	MLP512, seed 1 (d 3072, d 512, d 512, d 512, d 10) [14]	CIFAR-10	54.7%
	MLP512, seed 2 (d 3072, d 512, d 512, d 512, d 10)	CIFAR-10	55.1%
	MLP512, seed 3 (d 3072, d 512, d 512, d 512, d 10)	CIFAR-10	55.2%
ii)	MLP1024 (d 3072, d 1024, d 512, d 512, d 10)	CIFAR-10	55.4%
iii)	miniAlexNet (c 300 5×5, p 3×3, r, c 150 5×5, p 3×3, r, f, d 384, d 192, d 10) [14]	CIFAR-10	78.5%
iv)	AlexNet [52]	ImageNet	56.5%
v)	VGG16 [11]	ImageNet	67.6%
vi)	VGG19 [11]	ImageNet	72.4%

We consider a variety of different networks to show that our results are valid for a wide range of architectures. Tab. 1 lists the network architectures, training datasets, and accuracies achieved on each dataset. We downloaded the large pre-trained networks iv) AlexNet [52] via `pytorch` [54], v) VGG16 [11] via `tensorflow` [55], and vi) VGG19 [11] via `pytorch` [54]. Networks i)-iii) are trained using mini-batch stochastic gradient decent for 100 epochs. The weights are initialized using the Glorot uniform distribution [56] and the biases are initialized with zeros. We standardize each image of the CIFAR-10 dataset by subtracting the mean and dividing by the standard deviation. We set the learning rate to 0.001 at the beginning and use an exponential learning rate schedule with decay constant 0.95. For all networks, we choose 0.95 as momentum and the mini-batch size is 32. Network architectures i)-ii) in Tab. 1 are trained without L_2 regularization, while we use an L_2 regularization strength of 10^{-4} for training miniAlexNet networks iii).

Appendix B. Kolmogorov-Smirnov tests against the Wigner surmise

The p-values of Kolmogorov-Smirnov tests of the unfolded singular value spacings against the Wigner surmise are shown in Tab. 2.

Appendix C. Details on unfolding the spectrum

We perform a Gaussian broadening [25, 65] by approximating the probability density as a sum of Gaussian functions centered around each of the m singular values ν_k with widths $\sigma_k = (\nu_{k+a} - \nu_{k-a})/2$, where $2a$ is the window size of the broadening [32, 66]

$$P(\nu) \approx \frac{1}{m} \sum_{k=1}^{m} \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(\nu - \nu_k)^2}{2\sigma_k^2}\right) . \tag{4}$$

Unfolding is a transformation that maps the singular values ν_i to uniformly distributed singular values ξ_i [23–25, 27, 32]. For this purpose, we first determine the probability density $P(\nu)$ using

Table 2. Kolmogorov-Smirnov test results of the distribution of unfolded singular value spacings of the weight matrices against the Wigner surmise. Rejection of the null hypothesis is based on the $\alpha = 0.05$ significance level. The p-value indicates how likely it is to obtain a distribution with at least as much cumulative density function deviation as the one tested for drawing random numbers from a Wigner surmise distribution. We find excellent agreement with the Wigner surmise for a variety of network architectures.

Network	Reject null hypothesis?			KS-test p-value		
	Layer 1	Layer 2	Layer 3	Layer 1	Layer 2	Layer 3
MLP512 (seed 1)	no	no	no	0.347	0.401	0.812
MLP512 (seed 2)	no	no	no	0.993	0.421	0.844
MLP512 (seed 3)	no	no	no	0.768	0.784	0.863
MLP1024	no	no	no	0.799	0.812	0.792
miniAlexNet (second conv. layer)		no			0.859	
AlexNet (dense layers)	no	no	no	0.670	0.229	0.160
VGG16 (dense layers)	no	no	no	0.923	0.312	0.309
VGG19 (dense layers)	no	no	no	0.376	0.652	0.557

Eq. (4) and calculate the corresponding cumulative distribution

$$F(\nu) = m \int_{-\infty}^{\nu} P(x)\, dx \; . \tag{5}$$

The unfolded singular values are defined as $\xi_i = F(\nu_i)$.

Appendix D. Controlling the learning regime

A criterion for estimating the learning regime was proposed by Chizat et al. [48]: For a neural network $f_{\boldsymbol{W}}$ that maps an input $\boldsymbol{x}$ to an output, and an accuracy function $\mathcal{A}(f_{\boldsymbol{W}}, \{\boldsymbol{x}\}, \{\boldsymbol{y}\})$, where $\{\boldsymbol{x}\}$ is a dataset with labels $\{\boldsymbol{y}\}$, one computes the network's linearization around the initial weights $\boldsymbol{W}_0$

$$\tilde{f}_{\boldsymbol{W}}(\boldsymbol{x}) = f_{\boldsymbol{W}_0}(\boldsymbol{x}) + (\boldsymbol{W} - \boldsymbol{W}_0) \cdot \nabla_{\boldsymbol{W}} f_{\boldsymbol{W}}|_{\boldsymbol{W}_0}(\boldsymbol{x}) \; . \tag{6}$$

In the lazy learning regime, where $\boldsymbol{W} \approx \boldsymbol{W}_0$, linearization is a good approximation such that the accuracies are barely different, i.e.

$$\mathcal{A}(f_{\boldsymbol{W}}, \{\boldsymbol{x}\}, \{\boldsymbol{y}\}) \approx \mathcal{A}(\tilde{f}_{\boldsymbol{W}}, \{\boldsymbol{x}\}, \{\boldsymbol{y}\}) \; . \tag{7}$$

On the contrary, in the rich learning regime, one expects significant deviations such that

$$\mathcal{A}(f_{\boldsymbol{W}}, \{\boldsymbol{x}\}, \{\boldsymbol{y}\}) \gg \mathcal{A}(\tilde{f}_{\boldsymbol{W}}, \{\boldsymbol{x}\}, \{\boldsymbol{y}\}) \; . \tag{8}$$

This criterion has the advantage that it can also be studied on a layer-wise basis by linearizing around a single weight matrix only, and as accuracies are in the range $[0, 1]$, it gives a scale for laziness comparable between different network architectures. A disadvantage is that it requires to compute the linearization which can be resource intensive for large networks. For obtaining $\tilde{f}_{\boldsymbol{W}}$, we use the `autodiff` implementation in the `jax` python package together with the `neural_tangents` package [67].

Accuracies for linearized and full MLP512 networks as a function of α are depicted in Fig. 4. As expected, the networks are in the rich regime for $\alpha < 1$, where the full networks (blue crosses) perform significantly better than the linearized networks (green circles), while we observe lazy learning for $\alpha < 1$. The network with $\alpha = 1$ (black symbols), lies about in the middle between the two regimes, where we also find the best test accuracy. We therefore denote $\alpha = 1$ as *typical learning*.

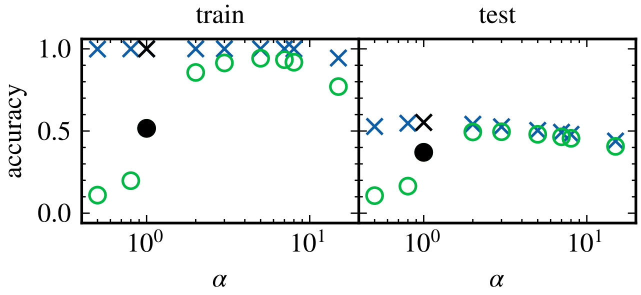

Figure 4. Comparison of training and test accuracies for full MLP512 networks (blue crosses) and linearized networks (green circles) around the initial weights of the second layer as a function of the laziness hyperparameter α. The black symbols indicate accuracies for $\alpha = 1$. For small $\alpha < 1$ accuracies of linearized and full networks deviate significantly which indicates rich learning, while for large $\alpha > 1$ performance differences are small indicating lazy learning [51].