How the Hessian-Spectrum of Linear Networks Depends on Data

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2026

Abstract

The Hessian matrix is an important quantity of interest when it comes to studying the loss landscape and optimization dynamics in deep learning, as well as designing measures of generalization, second-order learning algorithms, etc. Prior works have focused on drawing conclusions from empirical results, or pursued a theoretical treatment under overly simplified settings. In this work, we derive the eigenvalues of the Hessian of linear networks with arbitrary widths and depths, and datasets with arbitrary number of samples, features, and labels. Importantly, for classification tasks with MSE loss, we identify that the sharpness of the solution is directly related to the maximum proportion of samples belonging to any class. We empirically validate our predictions, and systematically analyze the effects of shedding the impractical assumptions one-at-a-time, as well as incorporating nonlinearities. We observe that our predictions are considerably robust in most cases, allowing us to extend our conclusions to more practical learning setups.

1. Introduction

Neural networks induce highly nonconvex loss landscapes that are complicated to study. Despite this complexity, the Hessian matrix of the loss, encoding only its second-order information, has played a central role in designing efficient optimization algorithms [8, 21], deriving measures of generalization [2, 13] – which, in turn, inspired the design of well-generalizing optimizers [5, 18] – as well as practical algorithms for pruning [9, 19], quantization [6], continual learning [22, 23], etc.

Our motivation resembles that of [28, 29], in that we aim to precisely characterize the full spectrum of the Hessian. However, while these works prioritize minimal assumptions at the cost of model complexity, we leverage standard assumptions in deep learning theory in exchange for networks with *arbitrary widths* and *depths*, and datasets with *arbitrary number of samples*, *features* and *labels*. Our methodology falls closest to [7], which studies the spectrum of the Hessian and training dynamics at the edge-of-stability [3] for matrix factorization problems. In contrast, we consider the problem of supervised learning, with the overarching goal of understanding the role of data-geometry on the sharpness of the learnt solution.

2. Setup

Consider the inputs $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^{d_0}$ collected in $\mathbf{X} \in \mathbb{R}^{d_0 \times n}$, and outputs $\mathbf{y}_1, \dots, \mathbf{y}_n \in \mathbb{R}^{d_L}$ collected in $\mathbf{Y} \in \mathbb{R}^{d_L \times n}$. Our model is an L-layer linear neural network, defined as

$$\mathbf{f}\left(\mathbf{x}; \mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_L\right) = \mathbf{W}_L \mathbf{W}_{L-1} \dots \mathbf{W}_1 \mathbf{x} \tag{1}$$

with $\mathbf{W}_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ is the learnable weight matrix in the ℓ^{th} layer, $\forall \ell \in [L] \coloneqq \{1, 2, \ldots, L\}$. In what follows, we denote the model parameters by $\mathbf{W}_{1:L} \coloneqq (\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_L)$. The loss function is the Mean-Squared Error (MSE):

$$\mathcal{L}\left(\mathbf{W}_{1:L}; \mathbf{X}, \mathbf{Y}\right) \coloneqq \frac{1}{2n} \left\| \mathbf{f}\left(\mathbf{X}; \mathbf{W}_{1:L}\right) - \mathbf{Y} \right\|_F^2 \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of the argument.

Assumption 1 *The loss is near-zero,* $\mathcal{L}\left(\mathbf{W}_{1:L}; \mathbf{X}, \mathbf{Y}\right) \approx 0$.

While this does not hold everywhere in the parameter space, it gets more accurate with training.

2.1. Approximating the Hessian-spectrum

We use the generalized Gauss-Newton (GGN) approximation of the Hessian, which is precise under Assumption 1:

$$\frac{\partial^2}{\partial \mathbf{W}_{1:L}^2} \mathcal{L}\left(\mathbf{W}_{1:L}; \mathbf{X}, \mathbf{Y}\right) \approx \frac{1}{n} \left(\frac{\partial \mathbf{f}\left(\mathbf{X}\right)}{\partial \mathbf{W}_{1:L}} \right)^\intercal \left(\frac{\partial \mathbf{f}\left(\mathbf{X}\right)}{\partial \mathbf{W}_{1:L}} \right) \in \mathbb{R}^{p \times p} \tag{3}$$

with the total number of parameters given by $p = \sum_{\ell=1}^{L} d_{\ell-1} d_\ell$, and the Jacobian w.r.t. predictions given by $\mathbf{J} \coloneqq \partial \mathbf{f}\left(\mathbf{X}\right) / \partial \mathbf{W}_{1:L} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & \ldots & \mathbf{J}_L \end{bmatrix} \in \mathbb{R}^{d_L n \times p}$, where

$$\mathbf{J}_\ell \coloneqq \partial \mathbf{f}\left(\mathbf{X}\right) / \partial \mathbf{W}_\ell = \left(\underbrace{\mathbf{W}_{\ell-1} \mathbf{W}_{\ell-2} \ldots \mathbf{W}_1}_{\mathbf{P}_\ell \in \mathbb{R}^{d_{\ell-1} \times d_0}} \mathbf{X} \right)^\intercal \otimes \left(\underbrace{\mathbf{W}_L \mathbf{W}_{L-1} \ldots \mathbf{W}_{\ell+1}}_{\mathbf{S}_\ell \in \mathbb{R}^{d_L \times d_\ell}} \right) \tag{4}$$

where $\otimes$ is the Kronecker outer-product. The GGN matrix shares its non-zero eigenvalues with the Neural Tangent Kernel (NTK) matrix [12], which is simpler to study:

$$\frac{1}{n} \mathbf{J}\mathbf{J}^\intercal = \sum_{\ell=1}^{L} \left(\frac{1}{n} \mathbf{X}^\intercal \mathbf{P}_\ell^\intercal \mathbf{P}_\ell \mathbf{X} \right) \otimes \left(\mathbf{S}_\ell \mathbf{S}_\ell^\intercal \right) \in \mathbb{R}^{n d_L \times n d_L} \tag{5}$$

Assumption 2 *We have an overdetermined system ($n \geq d_0$) and the feature matrix is uncorrelated, i.e. it satisfies*

$$\frac{1}{n} \mathbf{X}\mathbf{X}^\intercal = \sigma_{\mathbf{x}}^2 \mathbf{I}_{d_0}$$

Although it is reasonable to assume more number of inputs than features, the feature matrix need not be whitened in practice. In this case, one may orthogonalize the data using its QR-decomposition.

2.2. Summary of Eigenvalues

Several works have used Hessian-based measures like the maximum eigenvalue (spectral norm for symmetric matrices) or the sum of eigenvalues (trace) to measure generalization. Intuitively, at a local minimum, the spectral norm measures the worst-case change in loss caused by a small perturbation in any direction, while the trace measures the expected change in loss caused by a random isotropic perturbation with small variance.

Following Equation 5, and using Assumption 2, the spectral norm can be upper bounded as

$$\lambda_{\max}\left(\frac{1}{n}\mathbf{J}\mathbf{J}^\intercal\right) \leq \sum_{\ell=1}^{L} \lambda_{\max}\left(\frac{1}{n}\mathbf{X}^\intercal\mathbf{P}_\ell^\intercal\mathbf{P}_\ell\mathbf{X}\right)\lambda_{\max}\left(\mathbf{S}_\ell\mathbf{S}_\ell^\intercal\right) = \sigma_{\mathbf{x}}^2 \cdot \sum_{\ell=1}^{L} \sigma_{\max}^2\left(\mathbf{P}_\ell\right)\sigma_{\max}^2\left(\mathbf{S}_\ell\right) \quad (6)$$

While exactness does not work out for the spectral norm, it does work out for the trace since it distributes over a sum:

$$\mathrm{Tr}\left(\frac{1}{n}\mathbf{J}\mathbf{J}^\intercal\right) = \sum_{\ell=1}^{L} \mathrm{Tr}\left(\frac{1}{n}\mathbf{X}^\intercal\mathbf{P}_\ell^\intercal\mathbf{P}_\ell\mathbf{X}\right)\mathrm{Tr}\left(\mathbf{S}_\ell\mathbf{S}_\ell^\intercal\right) = \sigma_{\mathbf{x}}^2 \cdot \sum_{\ell=1}^{L} \|\mathbf{P}_\ell\|_F^2 \|\mathbf{S}_\ell\|_F^2 \quad (7)$$

We visualize this exactness in Figure 3, where we note that the approximation holds very well, even early in training when the loss is high so that Assumption 1 does not hold.

3. Strongly Balanced Deep Networks

Assumption 3 *Assume that consecutive layers are strongly balanced, i.e.* $\mathbf{W}_\ell\mathbf{W}_\ell^\intercal = \mathbf{W}_{\ell+1}^\intercal\mathbf{W}_{\ell+1}$, $\forall \ell \in [L-1]$.

In essence, this implies that the right singular vectors of a layer are aligned with the left singular vectors of the next layer. Moreover, all layers share their non-zero singular values; we denote them by $\sigma_i\left(\mathbf{W}\right)$. While arbitrary points in the parameter space need not correspond to balanced layers, initially balanced layers remain balanced under gradient flow training, provided their is no nonlinearity between them [1, 4]. Even if initially unbalanced, Singh et al. [27, Theorem 1] showed that the layers become increasingly balanced under weight-decay training, while Ghosh et al. [7, Proposition 2] showed that gradient descent training at the edge-of-stability [3] balances the singular values of the weight matrices. Therefore, the results hereon can be seen as characterizing the local geometry of the minimizers.

Under Assumption 3, we have $\mathbf{P}_\ell^\intercal\mathbf{P}_\ell = \left(\mathbf{W}_1^\intercal\mathbf{W}_1\right)^{\ell-1}$ and $\mathbf{S}_\ell\mathbf{S}_\ell^\intercal = \left(\mathbf{W}_L\mathbf{W}_L^\intercal\right)^{L-\ell}$. Therefore,

$$\frac{1}{n}\mathbf{J}\mathbf{J}^\intercal = \sum_{\ell=1}^{L} \underbrace{\left(\frac{1}{n}\mathbf{X}^\intercal\left(\mathbf{W}_1^\intercal\mathbf{W}_1\right)^{\ell-1}\mathbf{X}\right)}_{\in\mathbb{R}^{n\times n}} \otimes \underbrace{\left(\mathbf{W}_L\mathbf{W}_L^\intercal\right)^{L-\ell}}_{\in\mathbb{R}^{d_L\times d_L}} \quad (8)$$

Eigenvectors. The (unnormalized) eigenvectors of each summand are given by the Kronecker products $\mathbf{q}_{ij} = \left(\frac{1}{\sqrt{n}}\mathbf{X}^\intercal\mathbf{v}_i\right) \otimes \mathbf{u}_j$, where $\mathbf{v}_i$ is the i^{th} right singular vector of $\mathbf{W}_1$, and $\mathbf{u}_j$ is the j^{th} left singular vector of $\mathbf{W}_L$.

Eigenvalues. Since each summand has the same eigenbasis, the eigenvalues add up over the summands. That is, under Assumption 2, the eigenvalues of the Hessian are given by

$$\frac{1}{\sigma_{\mathbf{x}}^2}\cdot\lambda_{ij}\left(\frac{1}{n}\mathbf{J}\mathbf{J}^\intercal\right) = \sum_{\ell=1}^{L} \sigma_i^{2(L-\ell)}\left(\mathbf{W}\right)\sigma_j^{2(\ell-1)}\left(\mathbf{W}\right) \quad (9a)$$

$$= \begin{cases} \frac{\sigma_i^{2L}(\mathbf{W})-\sigma_j^{2L}(\mathbf{W})}{\sigma_i^2(\mathbf{W})-\sigma_j^2(\mathbf{W})}\ , & \sigma_i\left(\mathbf{W}\right) \neq \sigma_j\left(\mathbf{W}\right) \\ L\cdot\sigma^{2(L-1)}\left(\mathbf{W}\right)\ , & \sigma \coloneqq \sigma_i\left(\mathbf{W}\right) = \sigma_j\left(\mathbf{W}\right) \end{cases} \quad (9b)$$

Hence, of the $p \coloneqq \sum_{\ell=1}^{L} d_{\ell-1} d_\ell$ eigenvalues of the Hessian, at most $\left(\min\left\{d_0, d_1, \ldots, d_L\right\}\right)^2$ are non-zero. This corresponds to the empirical observations suggesting that the bulk of eigenvalues of the Hessian are near-zero [24–26].

Shallow Networks. In Appendix B, we show that the very similar conclusions hold for shallow linear networks ($L = 2$) *even without* Assumption 3. A direct consequence is for the spectral norm:

$$\lambda_{\max}\left(\frac{1}{n}\mathbf{J}\mathbf{J}^\intercal\right) = \sigma_{\mathbf{x}}^2\left(\|\mathbf{W}_1\|_2^2 + \|\mathbf{W}_2\|_2^2\right) \tag{10}$$

where $\|\cdot\|_2$ denotes the spectral norm. This contradicts the conclusion drawn in Singh et al. [29, Appendix D.3.3], which suggests that the maximum eigenvalue is given by the maximum spectral norm, $\max\left\{\|\mathbf{W}_1\|_2^2, \|\mathbf{W}_2\|_2^2\right\}$; we believe that the discrepancy is caused by the block-diagonal–dominant structure of the Hessian assumed in their analysis. As for the sum of eigenvalues:

$$\operatorname{Tr}\left(\frac{1}{n}\mathbf{J}\mathbf{J}^\intercal\right) = \sigma_{\mathbf{x}}^2\left(d_2\|\mathbf{W}_1\|_F^2 + d_0\|\mathbf{W}_2\|_F^2\right) \tag{11}$$

This relation provides insight into the progressive flattening observed in early-training [14, 15] – sharpness decreases since the parameter norm decreases in early-training, as the model output evolves to align with the labels [27, Theorem 2].

4. Dependence on Data

Up till now, we have assumed 0 loss, whitened inputs and strong balancedness. This ties the singular values of the weight matrices to the data in a unique way – Assumptions 1 and 2 together imply that

$$\mathbf{W}_L\mathbf{W}_{L-1}\ldots\mathbf{W}_1 = \frac{1}{n\sigma_{\mathbf{x}}^2}\mathbf{Y}\mathbf{X}^\intercal \tag{12}$$

Then, under Assumption 3, the non-zero singular values of the weight matrix are given by

$$\left(\mathbf{W}_1^\intercal\mathbf{W}_1\right)^L = \left(\frac{1}{n\sigma_{\mathbf{x}}^2}\right)^2 \cdot \mathbf{X}\mathbf{Y}^\intercal\mathbf{Y}\mathbf{X}^\intercal \implies \sigma_i^L\left(\mathbf{W}\right) = \frac{1}{\sqrt{n}\sigma_{\mathbf{x}}} \cdot \sigma_i\left(\mathbf{Y}\right) \tag{13}$$

We visualize this correspondence in the left panel in Figure 1, where we note that the theory predicts the non-zero eigenvalues exactly.

Using Equation 9b, the maximum eigenvalue of the Hessian is given by

$$\lambda_{\max}\left(\frac{1}{n}\mathbf{J}\mathbf{J}^\intercal\right) = L\sigma_{\mathbf{x}}^2 \cdot \left(\frac{1}{\sqrt{n}\sigma_{\mathbf{x}}} \cdot \sigma_i\left(\mathbf{Y}\right)\right)^{2(1-1/L)} = \frac{L\sigma_{\mathbf{x}}^{2/L}}{n^{1-1/L}} \cdot \sigma_{\max}^{2(1-1/L)}\left(\mathbf{Y}\right) \tag{14}$$

For one-hot encoded labels, $\left\{\sigma_i^2\left(\mathbf{Y}\right)/n\right\}_{i=1}^{d_L}$ is the empirical distribution of labels, i.e. proportion of labels from each class in the training set; in what follows, we will use $\alpha_i \coloneqq \sigma_i^2\left(\mathbf{Y}\right)/n$ to denote these proportions. Therefore, in this case, the sharpness at a balanced solution is independent of the dataset size n, in disagreement with Cohen et al. [3, Figure 18], which suggests that sharpness of gradient-flow solution increases with n. This effect is amplified by large depths, mainly due to the leading factor of L in eigenvalue computation – we visualize this in the middle panel in Figure 1.

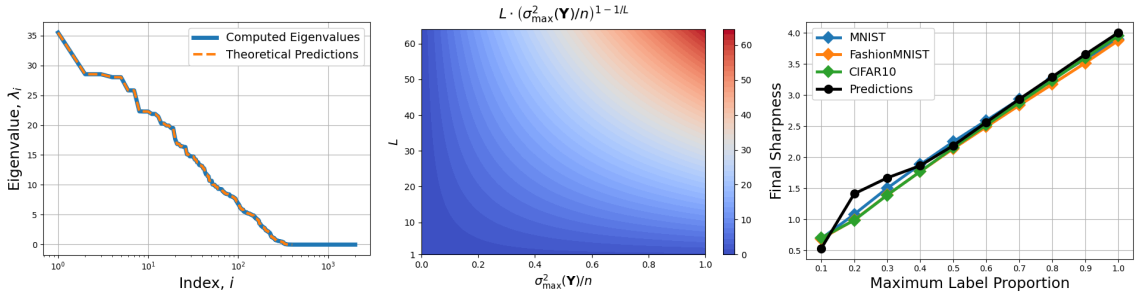

Figure 1: Computing eigenvalues of the Hessian of strongly balanced solutions using the labels, $\mathbf{Y}$. (a) Computed numerically and theoretically (Equation 13) with random data. (b) Contour plot on the effect of depth and spectral norm. (c) Spectral norm of trained models and theoretical predictions (Equation 14).

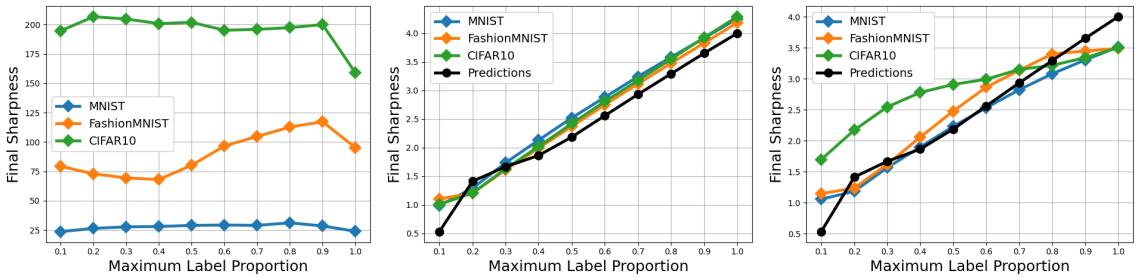

Figure 2: Ablating model assumptions: (a) raw inputs (shedding Assumption 2), (b) Kaiming uniform initialization (shedding Assumption 3), and (c) unbalanced initialization and Tanh (nonlinear) activation.

The discussion above for classification tasks implies that the model learns sharper solutions when there is one class with disproportionately large number of labels; we illustrate this in the right panel in Figure 1. Such datasets are, intuitively, simpler to learn, e.g. as an extreme case, a dataset with *all inputs belonging to one class* should be easier to learn than a dataset with uniform label distribution. This intuition is in disagreement with Cohen et al. [3, Caveat 2], which suggests that the model achieves lower peak-sharpness on simpler datasets. Moreover, the left panel in Figure 2 (see discussion below) suggests that peak-sharpness is strongly influenced by the input-geometry.

5. Ablations

We repeat the experiment in Figure 1, shedding each of Assumptions 2 and 3 one-at-a-time. In the left panel in Figure 2, we present the results shedding the assumption on whitened inputs. In this case, our theory breaks down immediately – the sharpness of the learnt solution turns out to have inconsistent variations with maximum label proportion in the dataset. In the middle panel, we present the results shedding the assumption on balanced weights, and use the standard Kaiming uniform initialization [10] implemented in the PyTorch framework. Here, we note the theoretical predictions made in previous section stay intact, as the final sharpness increasing with label imbalance. Finally, we inspect if incorporating nonlinearity breaks our theory. In the right panel, we present results for a Tanh-MLP, noting that while the results are slightly different from the theoretical predictions, they are qualitatively similar, allowing us to extend our conclusions on the effects of label distribution to such nonlinear networks.

References

[1] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 244–253. PMLR, 10–15 Jul 2018.

[2] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-SGD: Biasing gradient descent into wide valleys. In *International Conference on Learning Representations*, 2017.

[3] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

[4] Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[5] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

[6] Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. In *Advances in Neural Information Processing Systems*, volume 35, pages 4475–4488. Curran Associates, Inc., 2022.

[7] Avrajit Ghosh, Soo Min Kwon, Rongrong Wang, Saiprasad Ravishankar, and Qing Qu. Learning dynamics of deep matrix factorization beyond the edge of stability. In *International Conference on Learning Representations*, volume 2025, pages 17753–17800, 2025.

[8] Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1842–1850. PMLR, 10–15 Jul 2018.

[9] B. Hassibi, D.G. Stork, and G.J. Wolff. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pages 293–299 vol.1, 1993.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[11] M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communication in Statistics- Simulation and Computation*, 18:1059–1076, 01 1989.

[12] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[13] Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd, 2018.

[14] Dayal Singh Kalra and Maissam Barkeshli. Phase diagram of early training dynamics in deep neural networks: effect of the learning rate, depth, and width. In *Advances in Neural Information Processing Systems*, volume 36, pages 51621–51662. Curran Associates, Inc., 2023.

[15] Dayal Singh Kalra, Tianyu He, and Maissam Barkeshli. Universal sharpness dynamics in neural network training: Fixed point analysis, edge of stability, and route to chaos. In *International Conference on Learning Representations*, volume 2025, pages 55966–56000, 2025.

[16] Andrew V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2): 517–541, 2001.

[17] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[18] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5905–5914. PMLR, 18–24 Jul 2021.

[19] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989.

[20] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]*, 2, 2010.

[21] James Martens. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 735–742, Madison, WI, USA, 2010. Omnipress.

[22] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 7308–7320. Curran Associates, Inc., 2020.

[23] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations*, 2021.

[24] Vardan Papyan. The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size, 2019.

[25] Levent Sagun, Léon Bottou, and Yann LeCun. Singularity of the hessian in deep learning. *CoRR*, abs/1611.07476, 2016.

[26] Levent Sagun, Utku Evci, V. Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks, 2018.

[27] Jasraj Singh, Enea Monzio Compagnoni, and Antonio Orvieto. Unified perspectives on balancedness and parameter-norm evolution in neural nets. In *Workshop on Scientific Methods for Understanding Deep Learning*, 2026.

[28] Sidak Pal Singh and Thomas Hofmann. Closed form of the hessian spectrum for some neural networks. In *High-dimensional Learning Dynamics 2024: The Emergence of Structure and Reasoning*, 2024.

[29] Sidak Pal Singh, Weronika Ormaniec, and Thomas Hofmann. Cracking the hessian: Closed-form hessian spectra for fundamental neural networks, 2026.

[30] Andreas Stathopoulos and Kesheng Wu. A block orthogonalization procedure with constant synchronization requirements. *SIAM Journal on Scientific Computing*, 23(6):2165–2182, 2002.

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Appendix A. Additional Figures

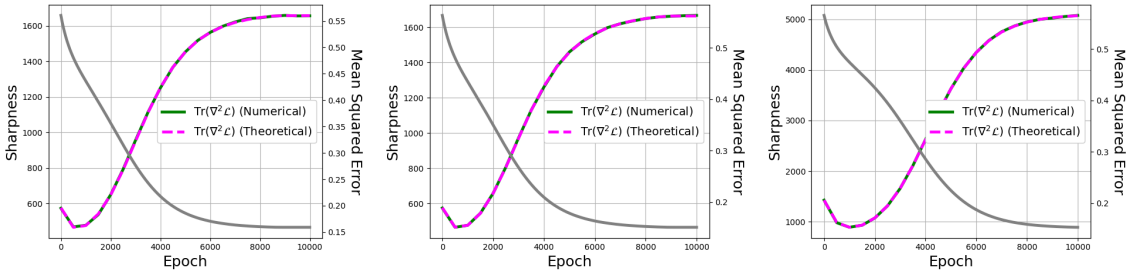

Figure 3: Sum of eigenvalues of the loss-Hessian, computed numerically (Hutchinson's method [11]) and analytically (Equation 7), for a MLP with (a) MNIST, (b) FashionMNIST and (c) CIFAR10.

Appendix B. Shallow Linear Networks

For $L = 2$, following Equation 5 and using Assumption 2, *all* non-zero eigenvalues of GGN are given by pairwise-sums of squared singular values of the two layers:

$$\lambda_{ij}\left(\frac{1}{n}\mathbf{J}\mathbf{J}^\intercal\right) = \sigma_{\mathbf{x}}^2\left(\sigma_i^2\left(\mathbf{W}_1\right) + \sigma_j^2\left(\mathbf{W}_2\right)\right) \tag{15}$$

where $i \in [\min\{d_0, d_1\}]$ and $j \in [\min\{d_1, d_2\}]$. The corresponding eigenvectors are given by

$$\mathbf{q}_{ij}\left(\frac{1}{n}\mathbf{J}\mathbf{J}^\intercal\right) = \left(\frac{1}{\sqrt{n}}\mathbf{X}^\intercal\mathbf{v}_i\right) \otimes \mathbf{u}_j \tag{16}$$

where $\mathbf{v}_i$ is the i^{th} right singular vector of $\mathbf{W}_1$, and $\mathbf{u}_j$ is the j^{th} left singular vector of $\mathbf{W}_2$.

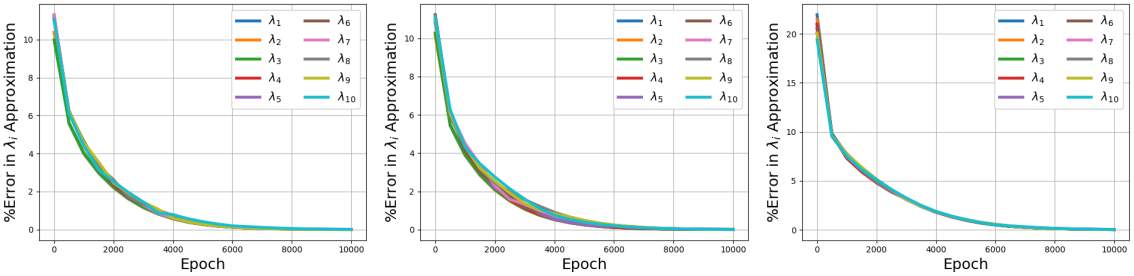

Figure 4: Percentage error in approximating top-10 eigenvalues of the loss-Hessian (using Equation 15) for a 2-layer MLP on (a) MNIST, (b) FashionMNIST, and (c) CIFAR10, with $\eta_{\max} = 0.001$.

We visualize this result in Figure 4, where we note that the approximation becomes increasingly precise as training progresses. We remark that Equation 15 implies that, of the $d_0d_1 + d_1d_2$ eigenvalues of the Hessian, at most $\min\{d_0, d_1\} \times \min\{d_1, d_2\}$ may be non-zero. We comment on this at the end of Section 3.

Appendix C. Experimental Details

Data Whitening. We orthogonalize a given feature matrix $\mathbf{X}' \in \mathbb{R}^{d_0 \times n}$ by setting $\mathbf{X} = \sqrt{n}\sigma_{\mathbf{x}}\mathbf{Q}$, where $\mathbf{Q}$ is derived from the QR decomposition of $\mathbf{X}'$; resulting matrix $\mathbf{X}$ satisfies Assumption 2.

Sampling Balanced Weights. Assume we are given a sequence of $k = \min\{d_0, d_1, \ldots, d_L\}$ singular values; we arrange them to create a diagonal matrix, $\mathbf{S}$. To construct a sequence of balanced weight matrices, we start by sampling the right singular vectors of $\mathbf{W}_1$ by orthogonalizing a random matrix with $\mathcal{N}(0,1)$ entries; denote it by $\mathbf{V}_1$. Then, for each $\ell \in [L]$, we do

1. Sample the right singular vectors of $\mathbf{W}_\ell$ by orthogonalizing a random matrix with $\mathcal{N}(0,1)$ entries.
2. Define the weight matrix as $\mathbf{W}_\ell = \mathbf{U}_\ell \mathbf{S} \mathbf{V}_\ell^\intercal$.
3. Set the right singular vectors of the next layer as the left singular vectors of the previous layers, $\mathbf{V}_{\ell+1} = \mathbf{U}_\ell$.

Subsampling. With real-world datasets MNIST [20], FashionMNIST [31] and CIFAR10 [17], we use subsets of $n = 5,000$ samples. If the maximum label proportion $\alpha_{\max}$ is set, we take $\lfloor (1-\alpha_{\max})n \rfloor$ samples from each of the classes $1-9$, where $\lfloor \cdot \rfloor$ denotes the floor function, and the rest of the samples from class 0.

Training. All models are trained using full-batch gradient descent for $10,000$ epochs, with an initial learning rate $\eta_{\max}$ which is successively halved anytime $< 0.1\%$ improvement in loss is observed over 100 epochs.

Figure 1:

- **Left:** We sample $\mathbf{X}' \in \mathbb{R}^{20\times 100}$ with standard normal entries, and then orthogonalize it with $\sigma_{\mathbf{x}} = 3$. We sample $d = 20$ singular values from $\mathcal{U}[0,1]$, and use it sample a sequence of $L = 5$ balanced weight matrices, with $\mathbf{W}_\ell \in \mathbb{R}^{d\times d}$. Finally, set $\mathbf{Y} = \mathbf{W}_5\mathbf{W}_4 \ldots \mathbf{W}_1\mathbf{X}$ so that Assumption 1 is satisfied.
- **Right:** Datasets are orthogonalized with $\sigma_{\mathbf{x}} = 1$, and our model is a 4-layer Linear-MLP with width 64, initialized with balanced weights, and trained with $\eta_{\max} = 0.01$.

Figure 2: All models are trained with $\eta_{\max} = 0.01$.

- **Left:** Datasets are *not* orthogonalized, and our model is a 4-layer Linear-MLP with width 64, initialized with balanced weights.
- **Middle:** Datasets are orthogonalized with $\sigma_{\mathbf{x}} = 1$, and our model is a 4-layer Linear-MLP with width 64, initialized with Kaiming uniform initialization [10] (standard in PyTorch).
- **Right:** Datasets are orthogonalized with $\sigma_{\mathbf{x}} = 1$, and our model is a 4-layer Tanh-MLP with width 64, initialized with balanced weights.

Figure 3: Sum of eigenvalues of the Hessian, computed numerically using Hutchinson's method [11] – $\mathrm{Tr}\left(\nabla^2\mathcal{L}\right) = \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_p)}\left[\mathbf{z}^\intercal\, \nabla^2\mathcal{L}\, \mathbf{z}\right]$ – with 100 samples and analytically using Equation 7. Datasets are orthogonalized with $\sigma_{\mathbf{x}} = 1$, and our model is a 4-layer Linear-MLP with width 256, trained with $\eta_{\max} = 0.003$.

Figure 4: Top-10 eigenvalues of the Hessian, computed numerically using LOBPCG [16, 30] and analytically using Equation 15. Datasets are orthogonalized with $\sigma_{\mathbf{x}} = 1$, and our model is a 2-layer Linear-MLP with width 256, trained with $\eta_{\max} = 0.001$.