The Deep Learning Limit: are negative neural network eigenvalues just noise?

Diego Granziol [1 2] **Timur Garipov** [3] **Stefan Zohren** [1 2] **Dmitry Vetrov** [3] **Stephen Roberts** [1 2]
Andrew Gordon Wilson [4]

Abstract

We study the deviation of the true risk surface from the empirical surface as a function of the ratio of model parametesr to dataset size. We model the empirical risk surface as a finite rank perturbation of the Gaussian Orthognal Ensemble and solve this problem analytically in the large dimension limit. Through this framework, we assess whether the true risk surface of neural networks is locally convex. We test our hypothesis using iterative methods on augmented datasets.

1. Introduction

The unparalleled success of deep learning, especially in computer vision and text classification tasks, has been accompanied by an explosion of theoretical (Choromanska et al., 2015b; Pennington & Bahri, 2017; Choromanska et al., 2015a) and empirical interest in their loss surfaces, typically through the study of the Hessian and its eigenspectrum (Sagun et al., 2016; 2017; Li et al., 2017; Ghorbani et al., 2019; Wu et al., 2017). The magnitude of the largest and smallest eigenvalues describes the local conditioning of the problem and the presence of negative eigenvalues indicates non-convexity. Hessian analysis has also been a primary tool in explaining the difference in generalization of solutions obtained, leading to the notion of flat and sharp minima (Hochreiter & Schmidhuber, 1997; Keskar et al., 2016; Izmailov et al., 2018).

In previous work, the Hessian of the loss function, at a point in weight space, is calculated using the entire dataset, this is denoted as the *full Hessian*, to be compatible with the terminology in the optimization literature for the full dataset gradient, i.e the *full gradient*. In this paper we analyze the Hessian under the expectation of the data generating distribution, the *true Hessian*. To the best of our knowledge no other work has ever investigated the *true Hessian*. Specifically we investigate the spectral distortions between the *true Hessian* and *Full Hessian* that occur when the number of parameters N far exceeds the number of samples T, i.e the ratio of parameters to samples, $q = N/T \gg 1$. We denote this the *Deep Learning Limit*, with which we exclusively concern ourselves. In deep learning, the parameter q is typically $\mathcal{O}(10^3)$ when the entire dataset is used and much larger when stochastic methods are employed.

[*]Equal contribution [1]Machine Learning Research Group, Oxford University [2]Oxford-Man Institute of Quantitative Finance [3]Samsung AI centre Moscow [4]Cornell. Correspondence to: Diego Granziol <diego@robots.ox.ac.uk>.

Presented at the ICML 2019 Workshop on Theoretical Physics for Deep Learning.

2. Formal Statement

Formally, We consider the family of prediction functions parameterized by the weight vector w, i.e $\mathcal{H} := \{h(\cdot; w) : w \in \mathbb{R}^N\}$ We aim to find the prediction function in this family that minimizes the losses incurred from inaccurate predictions. We assume a given loss $l : \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \to \mathbb{R}$, yielding loss $l(h(x; w)y)$ when $h(x; w)$ and y are the respective predicted and true outputs. Ideally we want to vary w such that we minimize the loss over our data generating distribution $P(x, y)$, this is known as the true or expected risk

$$R_{true}(w) = \int_{\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}} l(h(x; w), y) dP(x, y) = \mathbb{E}[l(h(x; w), y)], \tag{1}$$

with corresponding gradient $g_{true} = \nabla R_{true}$ and Hessian $H_{true} = \nabla\nabla R_{true}$. However given a dataset of size T, we only have access to our empirical risk

$$R_{emp}(w) = \frac{1}{T} \sum_{i=1}^{T} l(h(x_i; w), y_i), \tag{2}$$

and the gradients $|g_{emp}\rangle$ and Hessians H_{emp} thereof. In the field of optimization, equation (2) is treated as the object of interest.

3. The object of interest: the true Hessian

The difference between our empirical and true hessian is given as

$$\mathbb{E}\frac{\partial^2}{\partial w_j \partial w_k} l(h(x; w), y) - \frac{1}{T} \sum_{i=1}^{T} \frac{\partial^2}{\partial w_j \partial w_k} l(h(x_i; w), y_i). \tag{3}$$

For any $P(x, y)$, such that the moments $\mathbb{E}[\nabla_j \nabla_k l(h(x; w), y)]^m$ are bounded and with sufficient independence between the samples (Stein, 1972), in the limit $T \to \infty$ this converges (almost surely) to a normal random variable $\mathbb{N}(\mu_{jk}, \sigma_{jk}^2/T)$, for unbiased samples $\mu_{jk} = 0$. Rewriting our empirical Hessian as

$$H_{emp}(w) = H_{true}(w) + \epsilon(w). \tag{4}$$

For a fixed dataset, the perturbing matrix ϵ can be seen as a fixed instance of a random variable. For finite N and $T \to \infty$, i.e $q \to 0$, $|\epsilon(w)| \to 0$, we recover our true Hessian. Similarly in this limit our empirical risk converges to our true and we eliminate the generalization gap.

In this work we consider the problem of learning $H_{true}(w)$ from $H_{emp}(w)$. This line of work is beneficial to the optimization and generalization communities. For example, were it possible to prove that the eigenvalues of H_{true} were everywhere positive and that the source of reported negative curvature was due to the interaction effect with $\epsilon(w)$, this would explain why convex optimization techniques were so effective in deep learning. Alternatively work in bounding or ways to reduce the difference $|H_{true}(w) - H_{emp}(w)|$ could lead to new bounds between the true and empirical risk or generalization techniques.

4. RMT and the Lanczos Algorithm

We use the formalism of random matrix theory (RMT), in order to calculate the spectral perturbations in the large dimension limit $N, T \to \infty$. $N/T = q > 0$. We approximate the spectrum of million parameter neural networks by using Gaussian quadrature and the Lanczos algorithm.

4.1. Random Matrix Theory

The resolvent of a matrix H is defined as

$$G_H(z) = (zI_N - H)^{-1} \tag{5}$$

with $z = x + i\eta \in \mathbb{C}$. The normalised trace operator of the resolvent, in the $N \to \infty$ limit

$$\mathcal{S}_N(z) = \frac{1}{N}\text{Tr}[G_H(z)] \xrightarrow{N\to\infty} \mathcal{S}(z) = \int \frac{\rho(u)}{z-u}du \tag{6}$$

is known as the Stieltjes transform of ρ. The functional inverse of the Siteltjes transform, is denoted the blue function $\mathcal{B}(\mathcal{S}(z)) = z$. The R transform is defined as

$$\mathcal{R}(w) = \mathcal{B}(w) - \frac{1}{w} \tag{7}$$

crucially for our calculations, it is known that the $\mathcal{R}$ transform of the Wigner ensemble is

$$\mathcal{R}_W(z) = \sigma^2 z \tag{8}$$

the property of freeness for non commutative random matrices can be considered analogously to the moment factorisation property of independent random variables. The normalized trace operator, which is equal to the first moment of the spectral density

$$\psi(H) = \frac{1}{N}\text{Tr}H = \frac{1}{N}\sum_{i=1}^{N}\lambda_i = \int_{\lambda\in\mathcal{D}} d\mu(\lambda)\lambda \tag{9}$$

We say matrices $A\&B$ for which $\psi(A) = \psi(B) = 0$[1] are free if they satisfy for any integers $n_1..n_k$ with $k \in \mathbb{N}^+$

$$\psi(A^{n_1}B^{n_2}A^{n_3}B^{n_4}) = \psi(A^{n_1})\psi(B^{n_2})\psi(A^{n_3})\psi(A^{n_4}) \tag{10}$$

A fixed matrix A and rotationally invariant matrix B $\Omega B\Omega$ are said to be free. The Gaussian Orthogonal Ensemble (GOE), which has all elements zero mean independent Gaussians with identical variance is rotationally invariant. We assume $\epsilon(w)$ to be GOE.

4.2. Learning the spectrum cheaply with Lanczos

The Lanczos algorithm (Meurant & Strakoš, 2006), requires Hessian vector products, for which we use the Pearlmutter trick (Pearlmutter, 1994) with computational cost $\mathcal{O}(NTm)$, where T is the dataset size and m is the number of Lanczos steps. Its relationship to Gaussian quadrature using random vectors allows us to learn a discrete approximation to the spectral density. A quadrature rule is a relation

$$\int_a^b f(\lambda)d\mu(\lambda) = \sum_{j=1}^{M}\rho_j f(t_j) + R[f] \tag{11}$$

for a function f,such that its Riemann-Stieltjes integral and all the moments exist, on the measure $d\mu(\lambda)$ on the interval $[a, b]$ where $R[f]$ denotes the unknown remainder. The nodes t_j of the Gauss quadrature rule is given by the Ritz values and the weights ρ_j by the squares of the first elements of the normalized eigenvectors of the Lanczos tri-diagonal matrix (Golub & Meurant, 1994). In the high dimensional regime $N \to \infty$, we expect the squared overlap of each random vector with an eigenvector of H, $|v^T\phi_i|^2 \approx \frac{1}{N}\forall i$ with high probability so we plot the spectra using a single random vector.

4.3. the Deep Learning Limit and data Augmentation

The full CIFAR-100 training set uses $50,000$ images, and the VGG16 and PreResNet networks have approximately $15,000,000$ and $1,000,000$ parameters so we have a $q = [300, 20]$ respectively for the full dataset. In order to probe the effects of exiting the *deep learning limit*, we calculate the spectra of networks using augmented data sets, by factors of

[1] We can always consider the transform $A - \psi(A)I$

$[10, 100]$ respectively, by using horizontal flips, 4×4 zero padding and random 32×32 crops. For the PreResNet110 with $N = 1,169,972$ parameters on CIFAR-10/CIFAR-100 this gives us $q < 1$. For simplicity of argument we will assume the augmented dataset is drawn from the data-generating distribution. We leave further analysis of how $q_{aug} \neq q_{new}$ to later analysis.

5. Low rank true Hessian

For the case where the r denoting the rank of $r(H_{true}) << N$, it is possible to analytically calculate the effect of the perturbing matrix ϵ on the rank r non noise contribution. We complete the calculation for H_{true} of rank 1 and assume that the perturbing matrix $\epsilon(w)$ is a GOE. Computing the $\mathcal{R}$ transform of the rank 1 matrix H_{true}, with largest non-trivial eigenvalue β, on the effect of the spectrum of a matrix $\epsilon(w)$, using the Stieltjes transform we easily find following (Bun et al., 2017) that

$$\mathcal{S}_{H_{true}}(u) = \frac{1}{N}\frac{1}{u-\beta} + \left(1 - \frac{1}{N}\right)\frac{1}{u} = \frac{1}{u}\left[1 + \frac{1}{N}\frac{\beta}{1-u^{-1}\beta}\right] \tag{12}$$

We can use perturbation theory similar to in equation (23) to find the R transform which to leading order gives

$$\mathcal{R}_{H_{true}}(\omega) = \frac{\beta}{N(1-\omega\beta)} \tag{13}$$

$$z = \mathcal{B}_{H_{true}}(\mathcal{S}_M(z)) + \frac{\beta}{N(1-\beta\mathcal{S}_{H_{emp}}(z))} \tag{14}$$

where we set $\omega = \mathcal{S}_{H_{emp}}(z)$, using the ansatz of $\mathcal{S}_{H_{emp}}(z) = \mathcal{S}_0(z) + \frac{\mathcal{S}_1(z)}{N} + \mathcal{O}(N^{-2})$ we find that $\mathcal{S}_0(z) = \mathcal{S}_{\epsilon(w)}(z)$ and using that $\mathcal{B}'_{H_{emp}}(z)$, we conclude that $\mathcal{S}_1(z) = -\frac{\beta\mathcal{S}'_{\epsilon(w)}(z)}{1-\mathcal{S}_{\epsilon(w)}(z)\beta}$ and hence

$$\mathcal{S}_{H_{emp}}(z) \approx \mathcal{S}_{\epsilon(w)}(z) - \frac{1}{N}\frac{\beta\mathcal{S}'_{\epsilon(w)}(z)}{1-\mathcal{S}_{\epsilon(w)}(z)\beta} \tag{15}$$

and hence in the large N limit the correction only survives if $\mathcal{S}_{\epsilon(w)}(z) = 1/\beta$ Considering ϵ to be a GOE, which is a special case of the Wigner matrix, with variance σ_ϵ^2

$$\begin{aligned} \frac{2\sigma_\epsilon^2}{\beta} &= z \pm \sqrt{z^2 - 4\sigma_\epsilon^2} \\ \therefore z &= \beta + \frac{\sigma_\epsilon^2}{\beta} \end{aligned} \tag{16}$$

clearly for $\beta \to -\beta$ we have $z = -\beta - \frac{\sigma_\epsilon^2}{\beta}$. Hence we have that the extremal (largest positive and negative) values of the empirical Hessian λ'_{ex} in terms of those from the true Hessian λ_{ex}

$$|\lambda'_{ex}| = \left\{ \begin{array}{ll} |\lambda_{ex} + \frac{\sigma_\epsilon^2}{\lambda_{ex}}|, & \text{if } |\lambda_{ex}| > \sigma_\epsilon \\ 2\sigma_\epsilon, & \text{otherwise} \end{array} \right\} \tag{17}$$

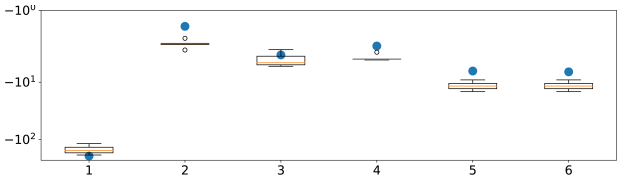

Figure 1. Boxplot of smallest eigenvalue with $B = 256$, for epochs $[0, 25, 50, 100, 150, 300]$ of the PreResNet100 for CIFAR-100.

Noting that, from equation (3), the variance of our correction matrix scales $\propto 1/T$, Wigner matrices are defined as scaled versions of the original matrices $W_N = M_N/\sqrt{N}$ so that the Frobenius norm is defined in the $N \to \infty$ limit (i.e $\mathbb{E}_\mu(\lambda^2) = \frac{1}{N}Tr(M_N^2/N) = \frac{1}{N^2}\sum_{i,j=1}^{N}|M_{i,j}^2|$) hence the the un-normalized variance scales $\propto N$, giving us the main theoretical result of this paper

$$|\lambda'_i| = \left\{ \begin{array}{ll} |\lambda_i + \frac{N}{T}\frac{\sigma_\epsilon^2}{\lambda_i}|, & \text{if } |\lambda_i| > \sigma_\epsilon \\ 2\sqrt{\frac{N}{T}}\sigma_\epsilon, & \text{otherwise} \end{array} \right\} \tag{18}$$

6. Experiments

We use the Pytorch framework for our neural network architectures and the GPytorch Lanczos implementation (Gardner et al., 2018). We use $m = 100$ Lanczos steps with a Gaussian random vector v at the weights of the final 300'th epoch of SGD training for the PreResNet110 and VGG16BN on the full CIFAR-100 dataset, shown in Figures 3, 5. We further test our theoretical results by learning the Hessian spectrum at the same point in weightspace, but for a dataset augmented by a factor of 10 for both VGG16BN (Figure 6) and PreResNet (Figure 4). Here we clearly see on both networks a contraction in magnitude of the extremal eigenvalues as we expand the dataset size. To see if this effect extends further we further augment the spectrum by another factor of 10 to give 5-million data-points and compute the Lanczos spectrum using $m = 10$ and we display the results in Table 1. We note that the largest eigenvalue shrinks and then stabilizes, whereas the smallest continues to shrink.

To further investigate whether our largest/smallest eigenvalues are positively/negatively biased due to the parameter q, as predicted by our theory in section 5, we five-fold sub-sample from our dataset and again running Lanczos with $m = 100$ steps, plot the results for the batch size of $B = 256$ case for the PreResNet 110 as box plots in Figures 1 and 2. As can be seen, although there is some stochasticity in the estimates, the largest eigenvalues are biased upwards and small ones downwards. This holds throughout training and for different batch sizes, networks and datasets.

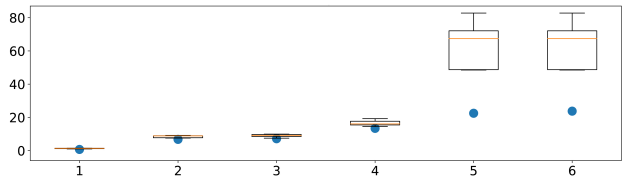

Figure 2. Boxplot of largest eigenvalue with $B = 256$, for epochs $[0, 25, 50, 100, 150, 300]$ of the PreResNet100 for CIFAR-100.

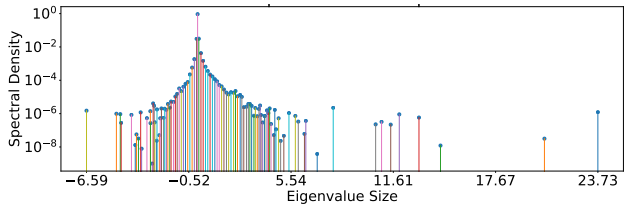

Figure 3. Full dataset spectral stem plot: CIFAR-100, PreResNet 110, Epoch 300 and $50,000$ samples.

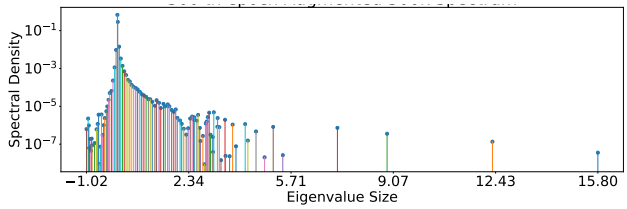

Figure 4. Augmented dataset spectral stem plot: CIFAR-100, PreResNet 110, Epoch 300 and $500,000$ samples.

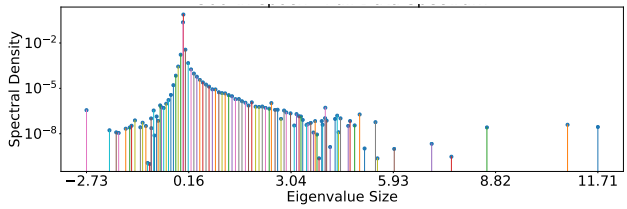

Figure 5. Full Dataset: CIFAR-100, VGG16BN, Epoch 300 and $50,000$ samples.

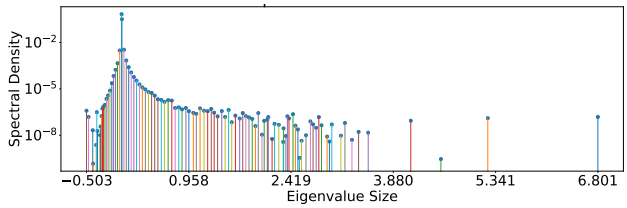

Figure 6. Augmented Dataset: CIFAR-100, VGG16BN, Epoch 300 and $500,000$ samples.

Table 1. Epoch 300 extremal Ritz values for different neural networks under data augmentation

Table 2. PreResNet110

T	λ_{max}	λ_{min}
50,000	23.73	-6.59
500,000	15.80	-1.02
5,000,000	15.69	-0.77

Table 3. VGG16BN

λ_{max}	λ_{min}
11.71	-2.73
6.80	-0.50
6.74	-0.34

6.1. Are the Negative Eigenvalues not inherent to the True Hessian?

Under the assumptions of our model, a stabilizing largest Ritz value indicates a well-separated eigenvalue from the GOE matrix. This indicates that the largest eigenvalue is definitely inherent to the true risk surface, but of smaller magnitude due to the effect of spectral broadening in both the *Full Hessian* and *batch Hessian* spectra. The smallest eigenvalue keeps contracting, but not anything close to the $\mathcal{O}(1\sqrt{10})$ effect expected from Equation (18), were the true Hessian to be positive definite. Further data augmentation could potentially be performed, but this begins to make the Hessian vector product computationally infeasible,

We remark that, in the augmented datasets Figure 4 and 6, the smallest eigenvalue is not well-separated from the rest of the spectrum, whereas in the full data set the smallest eigenvalues are better separated, not accounting for the issues with data-augmentation. This observation also supports the argument that the negative eigenvalues could be due to the perturbing matrix, as in the large dimension limit the support of the Wigner ensemble is compact. We leave a more rigorous exposition of this to future work.

7. Conclusion and Discussion

We introduce the concept of the *deep learning limit* and discuss expected spectral distortions occurring in an analytic random matrix theoretic framework under strict assumptions of the form of the perturbing matrix and rank of the true Hessian. We find that the extremal eigenvalues are extremized further and demonstrate this empirically, using stochastic Lanczos quadrature, sub-sampling and data augmentation. The Wigner result associated with the limiting spectral density can be extended to non-identical element variances (Tao, 2012) and to element dependence (Götze et al., 2012; Schenker & Schulz-Baldes, 2005). Similar to (Pennington & Bahri, 2017) we expect our analysis to hold more generally outside the assumptions stated.

References

Bun, J., Bouchaud, J.-P., and Potters, M. Cleaning large correlation matrices: tools from random matrix theory. *Physics Reports*, 666:1–109, 2017.

Cai, T., Fan, J., and Jiang, T. Distributions of angles in random packing on spheres. *The Journal of Machine Learning Research*, 14(1):1837–1864, 2013.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pp. 192–204, 2015a.

Choromanska, A., LeCun, Y., and Arous, G. B. Open problem: The landscape of the loss surfaces of multilayer networks. In *Conference on Learning Theory*, pp. 1756–1760, 2015b.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, pp. 7576–7586, 2018.

Ghorbani, B., Krishnan, S., and Xiao, Y. An investigation into neural net optimization via hessian eigenvalue density. *arXiv preprint arXiv:1901.10159*, 2019.

Golub, G. H. and Meurant, G. Matrices, moments and quadrature. *Pitman Research Notes in Mathematics Series*, pp. 105–105, 1994.

Golub, G. H. and Van Loan, C. F. *Matrix computations*, volume 3. JHU press, 2012.

Götze, F., Naumov, A., and Tikhomirov, A. Semicircle law for a class of random matrices with dependent entries. *arXiv preprint arXiv:1211.0389*, 2012.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Hutchinson, M. F. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450, 1990.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Li, H., Xu, Z., Taylor, G., and Goldstein, T. Visualizing the loss landscape of neural nets. *arXiv preprint arXiv:1712.09913*, 2017.

Meurant, G. and Strakoš, Z. The lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, 15:471–542, 2006.

Pearlmutter, B. A. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

Pennington, J. and Bahri, Y. Geometry of neural network loss surfaces via random matrix theory. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2798–2806. JMLR. org, 2017.

Roosta-Khorasani, F. and Ascher, U. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, 15(5):1187–1212, 2015.

Sagun, L., Bottou, L., and LeCun, Y. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016.

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Schenker, J. H. and Schulz-Baldes, H. Semicircle law and freeness for random matrices with symmetries or correlations. *arXiv preprint math-ph/0505003*, 2005.

Stein, C. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, pp. 583–602, Berkeley, Calif., 1972. University of California Press.

Tao, T. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012.

Wu, L., Zhu, Z., et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

A. Random Matrix Theory Essentials

Definition A.1. *Let* $\{Y_i\}$ *and* $\{Z_{ij}\}_{1\leq i\leq j}$ *be two real-valued families of zero mean, i.i.d. random variables. Furthermore suppose that* $\mathbb{E}Z_{12}^2 = 1$ *and for each* $k \in \mathbb{N}$*,*

$$max(E|Z_{12}^k, E|Y_1|^k) < \infty. \tag{19}$$

Consider an $n \times n$ symmetric matrix M_n, whose entries are given by

$$\begin{cases} M_n(i,i) = Y_i \\ M_n(i,j) = Z_{ij} = M_n(j,i), \quad \text{if } x \geq 1. \end{cases} \tag{20}$$

The Matrix M_n is known as a real symmetric Wigner matrix.

Theorem 1. *Let $\{M_n\}_{n=1}^{\infty}$ be a sequence of Wigner matrices, and for each n denote $X_n = M_n/\sqrt{n}$. Then μ_{X_n}, converges weakly, almost surely to the semi circle distribution,*

$$\sigma(x)dx = \frac{1}{2\pi}\sqrt{4-x^2}\mathbf{1}_{|x|\leq 2}. \tag{21}$$

B. Derivation

The Stieltjes transform of Wigner's semi-circle law, can be written as (Tao, 2012)

$$\mathcal{S}_W(z) = \frac{z \pm \sqrt{z^2 - 4\sigma^2}}{2\sigma^2}. \tag{22}$$

From the definition of the Blue transform, we hence have,

$$\begin{aligned} z &= \frac{\mathcal{B}_W(z) \pm \sqrt{\mathcal{B}_W^2(z) - 4\sigma^2}}{2\sigma^2} \\ (2\sigma^2 z - \mathcal{B}_W(z))^2 &= \mathcal{B}_W^2(z) - 4\sigma^2 \\ \therefore \mathcal{B}_W(z) &= \frac{1}{z} + \sigma^2 z \\ \therefore \mathcal{B}_W(z) &= \sigma^2 z. \end{aligned} \tag{23}$$

C. Lanczos algorithm

In order to empirically analyze properties of modern neural network spectra, with tens of millions of parameters $N = \mathcal{O}(10^7)$, we use the Lanczos algorithm (Meurant & Strakoš, 2006), with Hessian vector products using the Pearlmutter trick (Pearlmutter, 1994). This has computational cost $\mathcal{O}(NTm)$, where T is the dataset size and m is the number of Lanczos steps. The main properties of the Lanczos algorithm are summarized in the theorems 2,3

Theorem 2. *Let $H^{N\times N}$ be a symmetric matrix with eigenvalues $\lambda_1 \geq .. \geq \lambda_n$ and corresponding orthonormal eigenvectors $z_1, ..z_n$. If $\theta_1 \geq .. \geq \theta_m$ are the eigenvalues of the matrix T_m obtained after m Lanczos steps and $q_1, ...q_k$ the corresponding Ritz eigenvectors then*

$$\begin{aligned} \lambda_1 \geq \theta_1 \geq \lambda_1 - \frac{(\lambda_1 - \lambda_n)\tan^2(\theta_1)}{(c_{k-1}(1+2\rho_1))^2} \\ \lambda_n \leq \theta_k \leq \lambda_m + \frac{(\lambda_1 - \lambda_n)\tan^2(\theta_1)}{(c_{k-1}(1+2\rho_1))^2} \end{aligned} \tag{24}$$

where c_k is the chebyshev polyomial of order k

Proof: see (Golub & Van Loan, 2012). Given a measure $d\mu(\lambda)$ on the interval $[a,b]$ and a function f (such that its Riemann-Stieltjes integral and all the moments exist) a quadrature rule is a relation

$$\int_a^b f(\lambda)d\mu(\lambda) = \sum_{j=1}^{M} w_j f(t_j) + R[f], \tag{25}$$

where $R[f]$ denotes the unknown remainder.

Theorem 3. *The eigenvalues of T_k are the nodes t_j of the Gauss quadrature rule, the weights w_j are the squares of the first elements of the normalized eigenvectors of T_k*

Proof: See (Golub & Meurant, 1994). The first term on the RHS of (25) using Theorem 3 can be seen as a discrete approximation to the spectral density matching the first m moments $v^T H^m v$ (Golub & Meurant, 1994; Golub & Van Loan, 2012), where v is the initial seed vector. Using the expectation of quadratic forms, for zero mean, unit variance random vectors, using the linearity of trace and expectation, so

$$\begin{aligned} &\mathbb{E}_v \text{Tr}(v^T H^m v) = \text{Tr}\mathbb{E}_v(vv^T H^m) = \text{Tr}(H^m) \\ &= \sum_{i=1}^{N} \lambda_i = N \int_{\lambda \in \mathcal{D}} \lambda d\mu(\lambda) \end{aligned} \tag{26}$$

The error between the expectation over the set of all zero mean, unit variance vectors v and the Monte Carlo sum used in practice can be bounded (Hutchinson, 1990; Roosta-Khorasani & Ascher, 2015). However in the high dimensional regime $N \to \infty$, we expect the squared overlap of each random vector with an eigenvector of H, $|v^T \phi_i|^2 \approx \frac{1}{N} \forall i$ with high probability. This can be seen my computing the moments of the overlap between Rademacher vectors, containing elements $P(v_j = \pm 1) = 0.5$. Further analytical results for Gaussian vectors have been obtained (Cai et al., 2013).