
Information Geometry of Orthogonal Initializations and Training

Piotr Aleksander Sokół and Il Memming Park

Department of Neurobiology and Behavior

Departments of Applied Mathematics and Statistics, and Electrical and Computer Engineering

Institutes for Advanced Computing Science and AI-driven Discovery and Innovation

Stony Brook University, Stony Brook, NY 11733

{memming.park, piotr.sokol}@stonybrook.edu

Abstract

Recently mean field theory has been successfully used to analyze properties of wide, random neural networks. It gave rise to a prescriptive theory for initializing feed-forward neural networks with orthogonal weights, which ensures that both the forward propagated activations and the backpropagated gradients are near ℓ_2 isometries and as a consequence training is orders of magnitude faster. Despite strong empirical performance, the mechanisms by which critical initializations confer an advantage in the optimization of deep neural networks are poorly understood. Here we show a novel connection between the maximum curvature of the optimization landscape (gradient smoothness) as measured by the Fisher information matrix (FIM) and the spectral radius of the input-output Jacobian, which partially explains why more isometric networks can train much faster. Furthermore, given that orthogonal weights are necessary to ensure that gradient norms are approximately preserved at initialization, we experimentally investigate the benefits of maintaining orthogonality throughout training, from which we conclude that manifold optimization of weights performs well regardless of the smoothness of the gradients. Moreover, motivated by experimental results we show that a low condition number of the FIM is not predictive of faster learning.

1 Introduction

Deep neural networks (DNN) have shown tremendous success in computer vision problems, speech recognition, amortized probabilistic inference, and the modelling of neural data. Despite their performance, DNNs face obstacles in their practical application, which stem from both the excessive computational cost of running gradient descent for a large number of epochs, as well as the inherent brittleness of gradient descent applied to very deep models. A number of heuristic approaches such as batch normalization, weight normalization and residual connections [11, 13, 26] have emerged in an attempt to address these trainability issues.

Recently mean field theory has been successful in developing a more principled analysis of gradients of neural networks, and has become the basis for a new random initialization principle. The mean field approach postulates that in the limit of infinitely wide random weight matrices, the distribution of pre-activations converges weakly to an isotropic Gaussian. Using this approach, a series of works proposed to initialize the networks in such a way that for each layer the input-output Jacobian has mean singular values of 1 [29]. This requirement was further strengthened to suggest that the spectrum of singular values of the input-output Jacobian should concentrate on 1, and that this can only be achieved with random orthogonal weight matrices.

Under these conditions the backpropagated gradients are bounded in ℓ_2 norm [24] irrespective of depth, i.e., they neither vanish nor explode. It was shown experimentally in [24, 34, 6] that networks with these *critical* initial conditions train orders of magnitude faster than networks with arbitrary

Preprint. Under review.

initializations. The empirical success invites questions from an optimization perspective on how the spectrum of the hidden layer input-output Jacobian relates to notions of curvature of the parameters space, and subsequently to convergence rate. The largest effective initial step size η_0 is proportional to $\frac{|m|}{M}$ for stochastic gradient descent, where the Hessian plays a central role for determining the local gradient smoothness M and the strong convexity m^1 [4, 5]. Recent attempts have been made to analyze the mean field geometry of the optimization landscape using the Fisher information matrix (FIM) [2, 15], which given its close correspondence with the Hessian of the neural network defines an approximate gradient smoothness. Karakida et al. [15] derived an upper bound on the maximum eigenvalue, however this bound is not satisfactory since it is agnostic of the entire spectrum of singular values and therefore cannot differentiate between Gaussian and orthogonal initializations. In this paper, we develop a new bound on the parameter space curvature M given the maximum eigenvalue of the Fisher information matrix $\lambda_{max}(\mathbf{G})$ under both Gaussian and orthogonal initializations. We show that this quantity is proportional to the maximum squared singular value of the input-output Jacobian. We use this result to probe different orthogonal initializations, and observe that, broadly speaking, networks with a smaller initial curvature train faster and generalize better, as expected. However, consistently with a previous report [25], we also observe highly isometric networks perform worse despite having a very small initial $\lambda_{max}(\mathbf{G})$. We propose a theoretical explanation for this phenomenon using the connections between the FIM and the recently introduced Neural Tangent Kernel [14, 19]. Given that the smallest and largest eigenvalues have an approximately inverse relationship [15], we propose an explanation that the long term optimization behavior is mostly controlled by the smallest eigenvalue m and therefore surprisingly there is a *sweetspot with the condition number being $\frac{m}{M} > 1$* . We then investigate whether constraining the spectrum of the Jacobian matrix of each layer affects optimization rate. We do so by training networks using Riemannian optimization to constrain their weights to be orthogonal, or nearly orthogonal and we find that manifold constrained networks are insensitive to the maximal curvature at the beginning of training unlike the unconstrained gradient descent (“Euclidean”). In particular, we observe that the advantage conferred by optimizing over manifolds cannot be explained by the improvement of the gradient smoothness as measured by $\lambda_{max}(\mathbf{G})$, which argues against the proposed role of Batch Normalization recently put forward in [27, 36]. Importantly, Euclidean training with a carefully designed initialization reduces the test misclassification loss at approximately the same rate as their manifold constrained counterparts, and overall attain a higher accuracy.

2 Background

2.1 Formal Description of the Network

Following [24, 25, 29], we consider a feed-forward, fully connected neural network with L hidden layers. Each layer $l \in \{1, \dots, L\}$ is given as a recursion of the form

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

where \mathbf{x}^l are the activations, \mathbf{h}^l are the pre-activations, $\mathbf{W}^l \in \mathbb{R}^{N^l \times N^{l-1}}$ are the weight matrices, \mathbf{b}^l are the bias vectors, and $\phi(\cdot)$ is the activation function. The input is denoted as \mathbf{x}^0 . The output layer of the network computes $\hat{\mathbf{y}} = g^{-1}(\mathbf{h}^g)$ where g is the link function and $\mathbf{h}^g = \mathbf{W}^g \mathbf{x}^L + \mathbf{b}^g$.

The hidden layer input-output Jacobian matrix $\mathbf{J}_{\mathbf{x}^0}^{\mathbf{x}^L}$ is,

$$\mathbf{J}_{\mathbf{x}^0}^{\mathbf{x}^L} \triangleq \frac{\partial \mathbf{x}^L}{\partial \mathbf{x}^0} = \prod_{l=1}^L \mathbf{D}^l \mathbf{W}^l \quad (2)$$

where \mathbf{D}^l is a diagonal matrix with entries $\mathbf{D}_{i,i}^l = \phi'(\mathbf{h}_i^l)$. As pointed out in [24, 29], the conditioning of the Jacobian matrix affects the conditioning of the back-propagated gradients for all layers.

2.2 Critical Initializations

Extending the classic result on the Gaussian process limit for wide layer width obtained by Neal [21], recent work [20, 18] has shown that for deep untrained networks with elements of their weight matrices $\mathbf{W}_{i,j}$ drawn from a Gaussian distribution $\mathcal{N}(0, \frac{\sigma^2 \mathbf{W}}{N})$ the empirical distribution of the pre-activations \mathbf{h}^l converges weakly to a Gaussian distribution $\mathcal{N}(0, q^l \mathbf{I})$ for each layer l in the limit

¹Recall that m is the smallest, potentially negative eigenvalue of the Hessian and M is its largest eigenvalue for twice differentiable objectives.

of the width $N \rightarrow \infty$. Similarly, it has been postulated that random orthogonal matrices scaled by $\sigma_{\mathbf{W}}$ give rise to the same limit. Under this mean-field condition, the variance of the pre-activation distribution q^l is recursively given by,

$$q^l = \sigma_{\mathbf{W}}^2 \int \phi(\sqrt{q^{l-1}}h) d\mu(h) + \sigma_{\mathbf{b}}^2 \quad (3)$$

where $\mu(h)$ denotes the standard Gaussian measure $\int \frac{dh}{\sqrt{2\pi}} \exp(-\frac{h^2}{2})$ and $\sigma_{\mathbf{b}}^2$ denotes the variance of the Gaussian distributed biases [29]. The variance of the first layer pre-activations q^1 depends on ℓ_2 norm squared of inputs $q^1 = \frac{\sigma_{\mathbf{W}}^2}{N} \|\mathbf{x}^0\|_2^2 + \sigma_{\mathbf{b}}^2$. The recursion defined in equation 3 has a fixed point

$$q^* = \sigma_{\mathbf{W}}^2 \int \phi(\sqrt{q^*}h) d\mu(h) + \sigma_{\mathbf{b}}^2 \quad (4)$$

which can be satisfied for all layers by appropriately choosing $\sigma_{\mathbf{W}}$, $\sigma_{\mathbf{b}}$ and scaling the input \mathbf{x}^0 . To permit the mean field analysis of backpropagated signals, the authors [29, 24, 25, 15] further assume the propagated activations and backpropagated gradients to be independent. Specifically,

Assumption 1. *[Mean field assumptions]*

- (i) $\lim_{N \rightarrow \infty} \mathbf{W} \xrightarrow{d} \mathcal{N}(0, q^*)$
- (ii) $\lim_{N \rightarrow \infty} \text{Cov}[\mathbf{J}_{\mathbf{x}^{i+1}}^g \mathbf{h}^i, \mathbf{J}_{\mathbf{x}^{j+1}}^g \mathbf{h}^j] = 0$ for all $i \neq j$

Under this assumption, the authors [29, 24] analyze distributions of singular values of Jacobian matrices between different layers in terms of a small number of parameters, with the calculations of the backpropagated signals proceeding in a selfsame fashion as calculations for the forward propagation of activations. The corollaries of Assumption 1 and condition in equation 4 is that $\phi'(\mathbf{h}^l)$ for $1 \leq l \leq L$ are i.i.d. In order to ensure that $\mathbf{J}_{\mathbf{x}^0}^{\mathbf{x}^L}$ is well conditioned, Pennington et al. [24] require that in addition to the variance of pre-activation being constant for all layers, two additional constraints be met. Firstly, they require that the mean square singular value of \mathbf{DW} for each layer have a certain value in expectation.

$$\chi = \frac{1}{N} \mathbb{E}[\text{Tr}[(\mathbf{DW})^\top \mathbf{DW}]] = \sigma_{\mathbf{W}}^2 \int [\phi'(\sqrt{q^*}h)]^2 d\mu(h) \quad (5)$$

Given that the mean squared singular value of the Jacobian matrix $\mathbf{J}_{\mathbf{x}^0}^{\mathbf{x}^L}$ is $(\chi)^L$, setting $\chi = 1$ corresponds to a critical initialization where the gradients are asymptotically stable as $L \rightarrow \infty$. Secondly, they require that the maximal squared singular value s_{max}^2 of the Jacobian $\mathbf{J}_{\mathbf{x}^0}^{\mathbf{x}^L}$ be bounded. Pennington et al. [24] showed that for weights with Gaussian distributed elements, the maximal singular value increases linearly in depth even if the network is initialized with $\chi = 1$. Fortunately, for orthogonal weights, the maximal singular value s_{max} is bounded even as $L \rightarrow \infty$ [25].

3 Theoretical results: Relating the spectra of Jacobian and Fisher information matrices

To better understand the geometry of the optimization landscape, we wish to put a Lipschitz bound on the gradient, which in turn gives an upper bound on the largest step size of any first order optimization algorithm. We seek to find local measures of curvature along the optimization trajectory. As we will show below the approximate gradient smoothness is tractable for random neural networks. The analytical study of Hessians of random neural networks started with [23], but was limited to shallow architectures. Subsequent work [2, 15] on second order geometry of random networks shares much of the spirit of the current work, in that it proposes to replace the possibly indefinite Hessian with the related Fisher information matrix. The Fisher information matrix plays a fundamental role in the geometry of probabilistic models, under the Kullback-Leibler divergence loss. However, because of its relation to the Hessian, it can also be seen as defining an approximate curvature matrix for second order optimization. Recall that the FIM is defined as

Definition. *Fisher Information Matrix*

$$\mathbf{G} \triangleq \mathbb{E}_{\mathbf{y}|\mathbf{x}^0} [\mathbb{E}_{\mathbf{x}^0} [\nabla_{\theta} \log p_{\theta}(\mathbf{y}|\mathbf{x}^0) \nabla_{\theta} \log p_{\theta}(\mathbf{y}|\mathbf{x}^0)^\top]] \quad (6)$$

$$= \mathbb{E}_{\mathbf{y}|\mathbf{x}^0} \left[\mathbb{E}_{\mathbf{x}^0} \left[\mathbf{J}_{\theta}^{h^g \top} \nabla_{h^g}^2 \mathcal{L} \mathbf{J}_{\theta}^{h^g} \right] \right] = \mathbb{E}_{\mathbf{y}|\mathbf{x}^0} \left[\mathbb{E}_{\mathbf{x}^0} \left[\mathbf{H} - \sum_k \nabla_{\mathbf{x}^g} \mathcal{L}_k \nabla_{\theta}^2 h_k^g \right] \right] \quad (7)$$

where \mathcal{L} denotes the loss and \mathbf{h}^g is the output layer. The relation between the Hessian and Fisher Information matrices is apparent from equation 7, showing that the Hessian \mathbf{H} is a quadratic form of the Jacobian matrices plus the possibly indefinite matrix of second derivatives with respect to parameters.

Our goal is to express the gradient smoothness using the results of the previous section. Given equation 7 we can derive an analytical approximation to the Lipschitz bound using the results from the previous section; i.e. we will express the expected maximum eigenvalue of the random Fisher information matrix in terms of the expected maximum singular value of the Jacobian $\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^L}$. To do so, let us consider the output of a multilayer perceptron as defining a conditional probability distribution $p_\theta(\mathbf{y}|\mathbf{x}^0)$, where $\Theta = \{\text{vec}(\mathbf{W}^1), \dots, \text{vec}(\mathbf{W}^L), \mathbf{b}^1, \dots, \mathbf{b}^L\}$ is the set of all hidden layer parameters, and θ is the column vector containing the concatenation of all the parameters in Θ . Then each random block of the Fisher information matrix with respect to parameter vectors $a, b \in \Theta$ can further be expressed as

$$\bar{\mathbf{G}}_{a,b} = \mathbf{J}_a^{\mathbf{h}^g \top} \mathbf{H}_g \mathbf{J}_b^{\mathbf{h}^g} \quad (8)$$

where the final layer Hessian \mathbf{H}_g is defined as $\nabla_{\mathbf{h}^g}^2 \log p_\theta(\mathbf{y}|\mathbf{x}^0)$. We can re-express the outer product of the score function $\nabla_{\mathbf{h}^g} \log p_\theta(\mathbf{y}|\mathbf{x}^0)$ as the second derivative of the log-likelihood (see equation 6), provided it is twice differentiable and it does not depend on \mathbf{y} , which also allows us to drop conditional expectation with respect to $\mathbf{y}|\mathbf{x}^0$. This condition naturally holds for all canonical link functions and matching generalized linear model loss functions. We define the matrix of partial derivatives of the α -th layer pre-activations with respect to the layer specific parameters separately for \mathbf{W}^α and \mathbf{b}^α as:

$$\mathbf{J}_a^{\mathbf{h}^\alpha} = \mathbf{x}^{\alpha-1 \top} \otimes \mathbf{I} \quad \text{for } a = \text{vec}(\mathbf{W}^\alpha) \quad (9)$$

$$\mathbf{J}_a^{\mathbf{h}^\alpha} = \mathbf{I} \quad \text{for } a = \mathbf{b}^\alpha \quad (10)$$

Under the assumptions in 1, we can further simplify the expression for the blocks of the Fisher information matrix equation 8.

Lemma 1. *The expected blocks with respect to weight matrices for all layers $\alpha, \beta \neq 1$ are*

$$\bar{\mathbf{G}}_{\text{vec}(\mathbf{W}^\alpha), \text{vec}(\mathbf{W}^\beta)} = \mathbb{E} [\mathbf{x}^{\alpha-1} \mathbf{x}^{\beta-1 \top}] \otimes \mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{h}^g \top} \mathbf{H}_g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g} \quad (11)$$

$$\left[\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{h}^g \top} \right] \mathbb{E} [\mathbf{1} \mathbf{1}^\top \mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{h}^g \top}] \quad (12)$$

Lemma 2. *The expected blocks with respect to a weight matrix \mathbf{W}^α and a bias vector \mathbf{b}^β are*

$$\bar{\mathbf{G}}_{\text{vec}(\mathbf{W}^\alpha), \mathbf{b}^\beta} = [\mathbf{x}^{\alpha-1 \top} \otimes \mathbf{I}] \mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{h}^g \top} \mathbf{H}_g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g} \quad (13)$$

Leveraging lemmas 1 and 2, and the previously derived spectral distribution for the singular values of the Jacobians, we derive a block diagonal approximation which in turn allows us to bound the maximum eigenvalue $\lambda_{\max}(\bar{\mathbf{G}})$. In doing so we will use a corollary of the block Gershgorin theorem.

Proposition 1 ((informal) Block Gershgorin theorem). *The maximum eigenvalue $\lambda_{\max}(\bar{\mathbf{G}})$ is contained in a union of disks centered around the maximal eigenvalue of each diagonal block with radii equal to the sum of the singular values of the off-diagonal terms.*

For a more formal statement see Appendix 6.1. The proposition 1 suggest a simple, easily computable way to bound the expected maximal eigenvalue of the Fisher information matrix—choose the block with the largest eigenvalue and calculate the expected spectral radii for the corresponding off diagonal terms. We do so by making an auxiliary assumption:

Assumption 2. *The maximum singular value of $\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{h}^g}$ monotonically increases as $\alpha \downarrow 1$.*

[t] The assumption that the maximal singular of the Jacobians $\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{h}^g}$ grows with backpropagated depth is well supported by previous observations [24, 25]. Under this condition it is sufficient to study the maximal singular value of blocks of the Fisher information matrix with respect to $\text{vec}(\mathbf{W}^1), \mathbf{b}^1$ and the spectral norms of its corresponding off-diagonal blocks. We define functions Σ_{\max} of each block as upper bounds on the spectral bounds of the respective block. The specific values are given in the following Lemma 3.

Lemma 3. *The maximum expected singular values $\mathbb{E} [\sigma_{\max}]$ of the off-diagonal blocks $\forall \beta \neq 1$ are bounded by Σ_{\max} :*

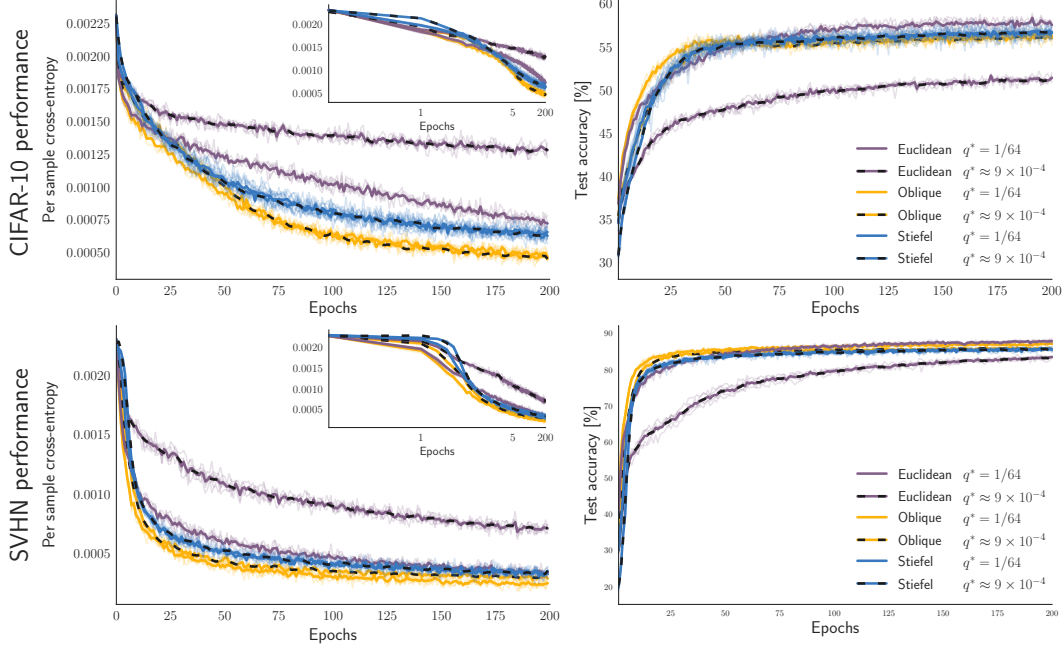


Figure 1: **Manifold constrained networks are insensitive to the choice of q^* :** Train loss and test accuracy for Euclidean, Stiefel and Oblique networks with two different values of q^* . The manifold constrained networks minimize the training loss at approximately the same rate, being faster than both Euclidean networks. Despite this, there is little difference between the test accuracy of the Stiefel and Oblique networks and the Euclidean networks initialized with $q^* = 9 \times 10^{-4}$. Notably, the latter attains a marginally higher test set accuracy towards the end of training.

$$\sigma_{\max}(\mathbf{G}_{\text{vec}(\mathbf{W}^1), \text{vec}(\mathbf{W}^\beta)}) \leq \Sigma_{\max}(\mathbf{G}_{\text{vec}(\mathbf{W}^1), \text{vec}(\mathbf{W}^\beta)}) \quad (14)$$

$$\triangleq \sqrt{N^\beta} \mathbb{E}[\phi(h)] \|\mathbb{E}[\mathbf{x}^0]\|_2 \mathbb{E}[\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g})] \mathbb{E}[\sigma_{\max}(\mathbf{H}_g)] \mathbb{E}[\sigma_{\max}(\mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (15)$$

$$\sigma_{\max}(\mathbf{G}_{\text{vec}(\mathbf{W}^1), b^\beta}) \leq \Sigma_{\max}(\mathbf{G}_{\text{vec}(\mathbf{W}^1), b^\beta}) \quad (16)$$

$$\triangleq \mathbb{E}[\phi(h)] \mathbb{E}[\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g})] \mathbb{E}[\sigma_{\max}(\mathbf{H}_g)] \mathbb{E}[\sigma_{\max}(\mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (17)$$

$$\sigma_{\max}(\mathbf{G}_{b^1, b^\beta}) \leq \Sigma_{\max}(\mathbf{G}_{b^1, b^\beta}) \triangleq \mathbb{E}[\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g})] \mathbb{E}[\sigma_{\max}(\mathbf{H}_g)] \mathbb{E}[\sigma_{\max}(\mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (18)$$

Proof. See Appendix 6.2 □

Note that the expectations for layers > 1 is over random networks realizations and averaged over data \mathbf{x}^0 ; i.e. they are taken with respect to the Gaussian measure, whereas the expectation for first layer weights is taken with respect to the empirical distribution of \mathbf{x}^0 (see equation 4).

Lemma 4. *The maximal singular values of the block diagonal elements are bounded by Σ_{\max}*

$$\sigma_{\max}(\mathbf{G}_{b^1, b^1}) \leq \Sigma_{\max}(\mathbf{G}_{b^1, b^1}) \quad (19)$$

$$\triangleq \mathbb{E}[\sigma_{\max}(\mathbf{H}_g)] \mathbb{E}[\sigma_{\max}(\mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})]^2 \quad (20)$$

$$\sigma_{\max}(\mathbf{G}_{\mathbf{W}^1, \mathbf{W}^1}) \leq \Sigma_{\max}(\mathbf{G}_{\mathbf{W}^1, \mathbf{W}^1}) \quad (21)$$

$$\triangleq \sigma_{\max}(\text{Cov}[\mathbf{x}^0, \mathbf{x}^0]) \mathbb{E}[\sigma_{\max}(\mathbf{H}_g)] \mathbb{E}[\sigma_{\max}(\mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})]^2 \quad (22)$$

Depending on the choice of q^* and therefore implicitly both the rescaling of \mathbf{x}^0 and the values of $\mathbb{E}[\phi(\mathbf{h})]$ either of the quantities might dominate.

Theorem 1 (Bound on the Fisher Information Eigenvalues). *If $\|\mathbb{E}[\mathbf{x}^0]\|_2 \leq 1$ then eigenvalue associated with b^1 will dominate, giving an upper bound on $\lambda_{max}(\bar{G})$*

$$\lambda_{max}(G) \leq \Sigma_{max}(\mathbf{G}_{b^1, b^1}) + \sum_{\beta > 1} \Sigma_{max}(\mathbf{G}_{b^1, b^\beta}) + \Sigma_{max}(\mathbf{G}_{\text{vec}(\mathbf{b}^1), \text{vec}(\mathbf{W}^\beta)})$$

otherwise the maximal eigenvalue of the FIM is bounded by

$$\lambda_{max}(G) \leq \Sigma_{max}(\mathbf{G}_{\text{vec}(\mathbf{W}^1), \text{vec}(\mathbf{W}^1)}) + \sum_{\beta > 1} \Sigma_{max}(\mathbf{G}_{\text{vec}(\mathbf{W}^1), b^\beta}) + \Sigma_{max}(\mathbf{G}_{\text{vec}(\mathbf{W}^1), \text{vec}(\mathbf{W}^\beta)})$$

The functional form of the bound is essentially quadratic in $\sigma_{max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g})$ since the term appears in the summand as with powers at most two. This result shows that the strong smoothness, given by the maximum eigenvalue of the FIM, is *proportional to* the squared maximum singular value of the input-output Jacobian $\sigma_{max}(\mathbb{E}[\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g}])$. Moreover, the bound essentially depends on q^* via the expectation $\mathbb{E}[\phi(h)]$, through $\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g}$ and implicitly through \mathbf{H}_g . For regression problems this dependence is monotonically increasing in q^* [25, 24] since \mathbf{H}_g is just the identity. However, this does not hold for all generalized linear models since $\lambda_{max}(\mathbf{H}_g)$ is not necessarily a monotonically increasing function of the pre-activation variance at layer \mathbf{h}^g . We demonstrate this in the case of softmax regression in the Appendix 6.3. Finally, to obtain a specific bound on $\lambda_{max}(\bar{G})$ we might consider bounding each $\sigma_{max}(\mathbb{E}[\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{h}^g}])$ appearing in theorem 1 in terms of its Frobenius norm. The corresponding result is the eigenvalue bound derived by [15].

4 Numerical experiments

4.1 Manifold optimization

Next we test the role that maintaining orthogonality throughout training has on the optimization performance. Moreover we numerically probe our predictions concerning the proportionality between the maximal eigenvalues of the Fisher information matrix and the maximal singular values of the Jacobian. Finally we measure the behavior of $\lambda_{max}(\bar{G})$ during training. To achieve this we perform optimization over manifolds.

Optimizing neural network weights subject to manifold constraints has recently attracted considerable interest [3, 12, 31, 35, 7, 22, 8]. In this work we probe how constraining the weights of each layer to be orthogonal or near orthogonal affects the spectrum of the hidden layer input-output Jacobian and of the Fisher information matrix. In Appendix 6.4 we provide a review notions from differential geometry and optimization over matrix manifolds [9, 1]. The Stiefel manifold and the oblique manifold will be used in the subsequent sections.

Stiefel Manifold $\text{St}(p, n) \triangleq \{\mathbf{W} \in \mathbb{R}^{n \times p} : \mathbf{W}^\top \mathbf{W} = \mathbf{I}_p\}$

Oblique Manifold $\text{Ob}(p, n) \triangleq \{\mathbf{W} \in \mathbb{R}^{n \times p} : \text{diag}(\mathbf{W}^\top \mathbf{W}) = \mathbf{1}\}$

Constraining the weights to this manifold is equivalent to using Weight Normalization [26]. Cho and Lee [7] derived a regularization term which penalizes the distance between the point in the manifold \mathbf{W} and the closest orthogonal matrix with respect to the Frobenius norm.

$$\rho(\lambda, \mathbf{W}) = \frac{\lambda}{2} \|\mathbf{W}^\top \mathbf{W} - \mathbf{I}\|_F^2 \quad (23)$$

4.2 Numerical Experiments

To experimentally test the potential effect of maintaining orthogonality throughout training and compare it to the unconstrained optimization [24], we trained a 200 layer tanh network on CIFAR-10 and SVHN. Following [24] we set the width of each layer to be $N = 400$ and chose the $\sigma_{\mathbf{W}}$, $\sigma_{\mathbf{b}}$ in such a way to ensure that χ concentrates on 1 but s_{max}^2 varies as a function of q^* (see Fig. 2). We considered two different critical initializations with $q^* = \frac{1}{64}$ and $q^* \approx 9 \times 10^{-4}$, which differ both in spread of the singular values as well as in the resulting training speed and final test accuracy as reported by [24]. To test how enforcing strict orthogonality or near orthogonality affects convergence speed and the maximum eigenvalues of the Fisher information matrix, we trained Stiefel and Oblique constrained networks and compared them to the unconstrained ‘‘Euclidean’’ network described in [24].

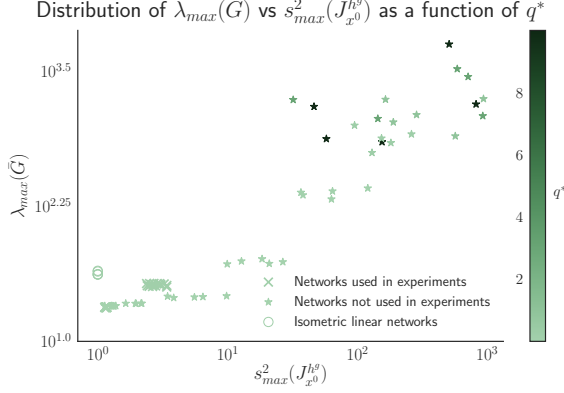


Figure 2: **At initialization the maximum eigenvalue of the Fisher information matrix $\bar{\mathbf{G}}$ correlates highly with the maximum squared singular value of the Jacobian $\mathbf{J}_{\mathbf{x}^0}^{h^g}$ ($\rho = 0.65$).**

We used a Riemannian version of ADAM [16]. When performing gradient descent on non-Euclidean manifolds, we split the variables into three groups: (1) Euclidean variables (e.g. the weights of the classifier layer, biases), (2) non-negative scaling $\sigma_{\mathbf{W}}$ both optimized using the regular version of ADAM, and (3) manifold variables optimized using Riemannian ADAM. The initial learning rates for all the groups, as well as the non-orthogonality penalty (see 23) for Oblique networks were chosen via Bayesian optimization, maximizing validation set accuracy after 50 epochs. All networks were trained with a minibatch size of 1000. We trained 5 networks of each kind, and collected eigenvalue and singular value statistics every 5 epochs, from the first to the fiftieth, and then after the hundredth and two hundredth epochs.

Based on the bound on the maximum eigenvalue of the Fisher information matrix derived in Section 3, we predicted that at initialization $\lambda_{max}(\bar{\mathbf{G}})$ should covary with $\sigma_{max}^2(\mathbf{J}_{\mathbf{x}^0}^{h^g})$. Our prediction is vindicated in that we find a strong, significant correlation between the two (Pearson coefficient $\rho = 0.64$). The numerical values are presented in Fig. 2. Additionally we see that both the maximum singular value and maximum eigenvalue increase monotonically as a function of q^* . Motivated by the previous work by Saxe et al. [28] showing depth independent learning dynamics in linear orthogonal networks, we included 5 instantiations of this model in the comparison. The input to the linear network was normalized the same way as the critical, non-linear networks with $q^* = 1/64$. The deep linear networks had a substantially larger $\lambda_{max}(\bar{\mathbf{G}})$ than its non-linear counterparts initialized with identically scaled input (Fig. 2). Having established a connection between q^* the maximum singular value of the hidden layer input-output Jacobian and the maximum eigenvalue of the Fisher information, we investigate the effects of initialization on subsequent optimization. As reported by Pennington et al. [24], the learning speed and generalization peak at intermediate values of $q^* \approx 10^{-0.5}$. This result is counter intuitive given that the maximum eigenvalue of the Fisher information matrix, much like that of the Hessian in convex optimization, upper bounds the maximal learning rate [5, 4]. To gain insight into the effects of the choice of q^* on the convergence rate, we trained the Euclidean networks and estimated the local values of λ_{max} during optimization. At the same time we asked whether we can effectively control the two aforesaid quantities by constraining the weights of each layer to be orthogonal or near orthogonal. To this end we trained Stiefel and Oblique networks and recorded the same statistics.

We present training results in Fig. 1, where it can be seen that Euclidean networks with $q^* \approx 9 \times 10^{-4}$ perform worse with respect to training loss and test accuracy than those initialized with $q^* = 1/64$. On the other hand, manifold constrained networks are insensitive to the choice of q^* . Moreover, Stiefel and Oblique networks perform marginally worse on the test set compared to the Euclidean network with $q^* = 1/64$, despite attaining a lower training loss. This latter fact indicates that manifold constrained networks are prone to overfitting.

We observe that reduced performance of Euclidean networks initialized with $q^* \approx 9 \times 10^{-4}$ may partially be explained by their rapid increase in $\lambda_{max}(\bar{\mathbf{G}})$ within the initial 5 epochs of optimization (see Fig. 3 in the Appendix). While all networks undergo this rapid increase, it is most pronounced for Euclidean networks with $q^* \approx 9 \times 10^{-4}$. The increase $\lambda_{max}(\bar{\mathbf{G}})$ correlates with the inflection point in the training loss curve that can be seen in the inset of Fig. 1. Interestingly, the manifold constrained networks optimize efficiently despite differences in $\lambda_{max}(\bar{\mathbf{G}})$, showing that their performance cannot be attributed to increasing the gradient smoothness as postulated by [27]. These results instead bolster support for the theory proposed by [17]

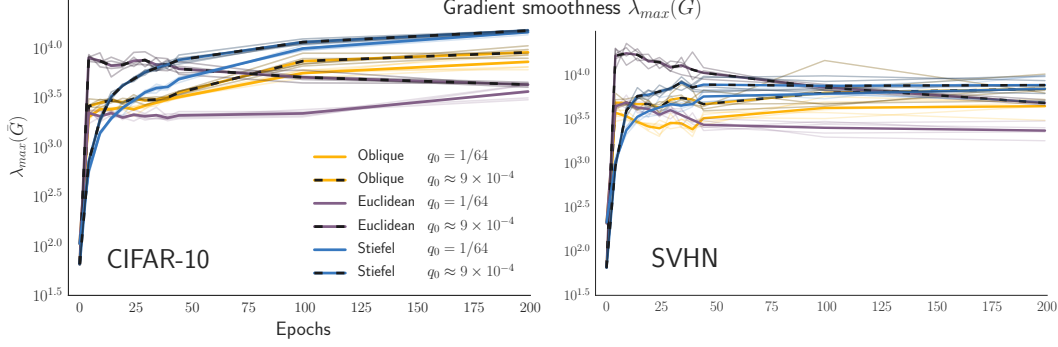


Figure 3: **For manifold constrained networks, gradient smoothness is not predictive of optimization rate. Euclidean networks with a low initial $\lambda_{max}(\bar{G})$ rapidly become less smooth, whereas Euclidean networks with a larger $\lambda_{max}(\bar{G})$ remain relatively smoother.** Notably, the Euclidean network with $q^* = 1/64$ has almost an order of magnitude smaller $\lambda_{max}(\bar{G})$ than the Stiefel and Oblique networks, but reduces training loss at a slower rate.

5 Discussion

Critical orthogonal initializations have proven tremendously successful in rapidly training very deep neural networks [24, 6, 25, 33]. Despite their elegant derivation drawing on methods from free probability and mean field theory, they did not offer a clear optimization perspective on the mechanisms driving their success. With this work we complement the understanding of critical orthogonal initializations by showing that the maximum eigenvalue of the Fisher information matrix, and consequently the local gradient smoothness is proportional to the maximum singular value of the input-output Jacobian. This gives an information geometric account of why the step size and training speed depend on q^* via its effect on $s_{max}(\mathbb{E}[\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^L}])$. We observed in numerical experiments that the paradoxical results reported in [24] whereby training speed and generalization attains a maximum for $q^* = 10^{-0.5}$ can potentially be explained by a rapid increase of the maximum eigenvalue of the FIM during training for the networks initialized with Jacobians closer to being isometric (i.e., smaller q^*). This increase effectively limits the learning rate during the early phase of optimization and highlights the need to analyze the trajectories of training rather than just initializations. We relate that to the recently proposed Neural Tangent Kernel[14, 19]. The NTK is defined as

$$\hat{\Theta}_{t,i,j} \triangleq \mathbf{J}_{\mathbf{x}^0}^{\mathbf{h}^g} \mathbf{J}_{\mathbf{x}^0}^{\mathbf{h}^g \top} \quad (24)$$

for $i, j \in N^g|\mathcal{D}|$ representing the block indices running over N^g outputs of the network and $|\mathcal{D}|$ data samples. The NTK is the derivative of a kernel defined by a random neural network. It prescribes the time evolution of the function and therefore offers a concise description of the network predictions. Importantly, the spectrum of the NTK coincides with that of the Fisher information for regression problems (see Appendix 6.5).

It is therefore interesting to understand the predictiveness of the Neural Tangent Kernel at initialization given its spectrum. Such a result has been recently presented by [19], who show that the discrepancy between training with a NTK frozen at initialization ($f_t^{lin}(\mathbf{x}^0)$) and a continuously updated one ($f_t(\mathbf{x}^0)$) can be bounded. Importantly the authors showed that rate at which discrepancy accrues depends exponentially on the smallest eigenvalue of the NTK. Given that the spectra of the Neural Tangent Kernel and the Fisher Information matrix coincide we can reason about this discrepancy over training time in terms of the smallest and largest eigenvalues of the Fisher Information matrix.

Lemma 5 (Lee et al. [19]). *The discrepancy between $g^{lin}(t) = f_t^{lin}(\mathbf{x}^0) - \mathbf{y}$ and $g(t) = f_t(\mathbf{x}^0) - \mathbf{y}$*

$$e^{\lambda_{min}(\bar{\mathbf{G}}_0)\eta t} \|g^{lin}(t) - g(t)\|_2 \leq \left(\eta \int_0^t e^{\lambda_{min}(\bar{\mathbf{G}}_0)\eta s} \|(\bar{\mathbf{G}}_s - \bar{\mathbf{G}}_0)\| \|g^{lin}(s)\|_2 ds \right) e^{\int_0^t (\eta \|(\bar{\mathbf{G}}_s - \bar{\mathbf{G}}_0)\|) ds} \quad (25)$$

where η is the learning rate.

Given the approximately inverse relation between the maximum and minimum eigenvalues of the Fisher information matrix [15], decreasing q^* increases $\lambda_{min}(\bar{\mathbf{G}}_0)$ and the solutions rapidly

diverge. This implies that a low condition number $\frac{\lambda_{max}(\mathbf{G}_0)}{\lambda_{min}(\mathbf{G}_0)}$ may be undesirable, and a degree of anisotropy is necessary for the Fisher Information matrix to be predictive of training performance. Finally, we compared manifold constrained networks with the Euclidean network, each evaluated with two initial values of q^* . From these experiments we draw the conclusion that manifold constrained networks are less sensitive to the initial strong smoothness, unlike their Euclidean counterparts. Furthermore, we observe that the rate at which Stiefel and Oblique networks decrease training loss is not dependent on their gradient smoothness, a result which is consistent with the recent analysis of [17].

References

- [1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, N.J. ; Woodstock, December 2007.
- [2] Shun-ichi Amari, Ryo Karakida, and Masafumi Oizumi. Fisher Information and Natural Gradient Learning of Random Deep Networks. *arXiv:1808.07172 [cond-mat, stat]*, August 2018. arXiv: 1808.07172.
- [3] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary Evolution Recurrent Neural Networks. *arXiv:1511.06464 [cs]*, November 2015. 00012 arXiv: 1511.06464.
- [4] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization Methods for Large-Scale Machine Learning. *arXiv:1606.04838 [cs, math, stat]*, June 2016. arXiv: 1606.04838.
- [5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization, With Corrections 2008*. Cambridge University Press, Cambridge, UK ; New York, 1 edition edition, March 2004.
- [6] Minmin Chen, Jeffrey Pennington, and Samuel Schoenholz. Dynamical Isometry and a Mean Field Theory of RNNs: Gating Enables Signal Propagation in Recurrent Neural Networks. In *International Conference on Machine Learning*, pages 873–882, July 2018.
- [7] Minhyung Cho and Jaehyung Lee. Riemannian approach to batch normalization. In *Advances in Neural Information Processing Systems*, pages 5229–5239, 2017.
- [8] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. *arXiv:1704.08847 [cs, stat]*, April 2017. arXiv: 1704.08847.
- [9] Alan Edelman, Tomás A. Arias, and Steven T. Smith. The Geometry of Algorithms with Orthogonality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, January 1998.
- [10] Mehrtaash Harandi and Basura Fernando. Generalized BackPropagation, \{E\}tude De Cas: Orthogonality. *arXiv:1611.05927 [cs]*, November 2016. 00004 arXiv: 1611.05927.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. 01528 arXiv: 1512.03385.
- [12] Mikael Henaff, Arthur Szlam, and Yann LeCun. Recurrent Orthogonal Networks and Long-Memory Tasks. *arXiv:1602.06662 [cs, stat]*, February 2016. arXiv: 1602.06662.
- [13] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*, February 2015. 00385 arXiv: 1502.03167.
- [14] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *arXiv:1806.07572 [cs, math, stat]*, June 2018. arXiv: 1806.07572.
- [15] Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Universal Statistics of Fisher Information in Deep Neural Networks: Mean Field Approach. *arXiv:1806.01316 [cond-mat, stat]*, June 2018. arXiv: 1806.01316.

- [16] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, December 2014. 01869 arXiv: 1412.6980.
- [17] Jonas Kohler, Hadi Daneshmand, Aurelien Lucchi, et al. Exponential convergence rates for Batch Normalization: The power of length-direction decoupling in non-convex optimization. *arXiv:1805.10694 [cs, stat]*, May 2018. arXiv: 1805.10694.
- [18] Jaehoon Lee, Yasaman Bahri, Roman Novak, et al. Deep Neural Networks as Gaussian Processes. *arXiv:1711.00165 [cs, stat]*, October 2017. arXiv: 1711.00165.
- [19] Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, et al. Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent. *arXiv:1902.06720 [cs, stat]*, February 2019.
- [20] Alexander G. de G. Matthews, Mark Rowland, Jiri Hron, Richard E. Turner, and Zoubin Ghahramani. Gaussian Process Behaviour in Wide Deep Neural Networks. *arXiv:1804.11271 [cs, stat]*, April 2018. arXiv: 1804.11271.
- [21] Radford M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer New York, New York, NY, 1996.
- [22] Mete Ozay and Takayuki Okatani. Optimization on Submanifolds of Convolution Kernels in CNNs. *arXiv:1610.07008 [cs]*, October 2016. 00004 arXiv: 1610.07008.
- [23] Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, pages 2798–2806, Sydney, NSW, Australia, 2017. JMLR.org.
- [24] Jeffrey Pennington, Sam Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems*, 2017.
- [25] Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. The Emergence of Spectral Universality in Deep Networks. *arXiv:1802.09979 [cs, stat]*, February 2018. arXiv: 1802.09979.
- [26] Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *arXiv:1602.07868 [cs]*, February 2016. 00003 arXiv: 1602.07868.
- [27] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How Does Batch Normalization Help Optimization? (No, It Is Not About Internal Covariate Shift). *arXiv:1805.11604 [cs, stat]*, May 2018. arXiv: 1805.11604.
- [28] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013. 00083.
- [29] Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep Information Propagation. *arXiv:1611.01232 [cs, stat]*, November 2016. arXiv: 1611.01232.
- [30] Christiane Tretter. *Spectral Theory of Block Operator Matrices and Applications*. IMPERIAL COLLEGE PRESS, October 2008.
- [31] Eugene Vorontsov, Chiheb Trabelsi, Samuel Kadoury, and Chris Pal. On orthogonality and learning recurrent networks with long term dependencies. *arXiv:1702.00071 [cs]*, January 2017. arXiv: 1702.00071.
- [32] Scott Wisdom, Thomas Powers, John R. Hershey, Jonathan Le Roux, and Les Atlas. Full-capacity unitary recurrent neural networks. *arXiv:1611.00035 [cs, stat]*, October 2016. arXiv: 1611.00035.
- [33] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical Isometry and a Mean Field Theory of CNNs: How to Train 10,000-Layer Vanilla Convolutional Neural Networks. In *International Conference on Machine Learning*, pages 5393–5402, July 2018.

- [34] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S. Schoenholz, and Jeffrey Pennington. Dynamical Isometry and a Mean Field Theory of CNNs: How to Train 10,000-Layer Vanilla Convolutional Neural Networks. *arXiv:1806.05393 [cs, stat]*, June 2018. arXiv: 1806.05393.
- [35] Di Xie, Jiang Xiong, and Shiliang Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. pages 5075–5084. IEEE, July 2017.
- [36] Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. A mean field theory of batch normalization. In *International Conference on Learning Representations*, 2019.

6 Appendix

6.1 Block Gershgorin Theorem

In Section 3, we considered a block diagonal approximation to the Fisher information matrix and derived an upper bound on the spectral norm for all the blocks. Using the properties of the off-diagonal blocks, we can get a more accurate estimate of the maximal eigenvalue of the Fisher information matrix. First, let us consider an arbitrarily partitioned matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, with spectrum $\lambda(\mathbf{A})$. The partitioning is done with respect to the set

$$\pi = \{p_j\}_{j=0}^L \quad (26)$$

with the elements of the set satisfying $0 < p_1 < p_2 < \dots < p_L = N$. Then each block of the matrix $\mathbf{A}_{i,j}$ is a potentially rectangular matrix in $\mathbb{R}^{(p_i - p_{i-1}) \times (p_j - p_{j-1})}$. We assume that $\mathbf{A}_{i,i}$ is self-adjoint for all i .

Let us define a disk as

$$C(c, r) \triangleq \{\lambda : \|\lambda - c\| \leq r\}. \quad (27)$$

The theorem as presented in Tretter [30] shows that the eigenvalues of $\lambda(\mathbf{A})$ are contained in a union of Gershgorin disks defined as follows

$$\lambda(\mathbf{A}) \subset \bigcup_{i=1}^L \left\{ \bigcup_{k=1}^{p_i - p_{i-1}} C \left(\lambda_k(\mathbf{A}_{ii}), \sum_{j=1, j \neq i}^L s_{\max}(\mathbf{A}_{i,j}) \right) \right\} \quad (28)$$

where the inner union is over a set disks for each eigenvalue of the block diagonal $\mathbf{A}_{i,i}$ while the outer union is over the L blocks in \mathbf{A} . The radius of the disk is constant for every eigenvalue in the i^{th} diagonal block $\mathbf{A}_{i,i}$ and is given by the sum of singular values of the off diagonal blocks. Therefore, the largest eigenvalue of \mathbf{A} lies in

$$\lambda_{\max}(\mathbf{A}) \subset \bigcup_{i=1}^L C \left(\lambda_{\max}(\mathbf{A}_{ii}), \sum_{j=1, j \neq i}^L s_{\max}(\mathbf{A}_{i,j}) \right) \quad (29)$$

6.2 Derivation of the expected singular values

$$\sigma_{\max}(\mathbf{G}_{\text{vec}(\mathbf{W}^1), \text{vec}(\mathbf{W}^\beta)}) = \mathbb{E} [\sigma_{\max}(\phi(h) \mathbb{1} \mathbf{x}^{0\top})] \otimes \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g\top} \mathbf{H}_g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (30)$$

$$= \sqrt{N^\beta} |\mathbb{E}[\phi(h)]| \|\mathbb{E}[\mathbf{x}^0]\|_2 \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g\top} \mathbf{H}_g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (31)$$

$$\leq \sqrt{N^\beta} |\mathbb{E}[\phi(h)]| \|\mathbb{E}[\mathbf{x}^0]\|_2 \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g\top})] \mathbb{E} [\sigma_{\max}(\mathbf{H}_g)] \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (32)$$

$$\sigma_{\max}(\mathbf{G}_{\text{vec}(\mathbf{W}^1), b^\beta}) \leq \mathbb{E} [\sigma_{\max}(\mathbf{x}^{0\top} \otimes \mathbf{I})] \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{h}^g\top} \mathbf{H}_g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (33)$$

$$= \|\mathbb{E}[\mathbf{x}^0]\|_2 (\mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g\top} \mathbf{H}_g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})]) \quad (34)$$

$$\leq \|\mathbb{E}[\mathbf{x}^0]\|_2 \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g\top})] \mathbb{E} [\sigma_{\max}(\mathbf{H}_g)] \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (35)$$

$$\sigma_{\max}(\mathbf{G}_{b^1, \text{vec}(\mathbf{W}^\beta)}) \leq \mathbb{E} [\sigma_{\max}(\mathbf{x}^{\beta-1\top} \otimes \mathbf{I})] \quad (36)$$

$$= \sqrt{N^\beta} |\mathbb{E}[\phi(h)]| \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g\top} \mathbf{H}_g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (37)$$

$$\leq |\mathbb{E}[\phi(h)]| \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g\top})] \mathbb{E} [\sigma_{\max}(\mathbf{H}_g)] \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (38)$$

$$\sigma_{\max}(\mathbf{G}_{b^1, b^\beta}) = \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g\top} \mathbf{H}_g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (39)$$

$$\leq \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{h}^g\top})] \mathbb{E} [\sigma_{\max}(\mathbf{H}_g)] \mathbb{E} [\sigma_{\max}(\mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{h}^g})] \quad (40)$$

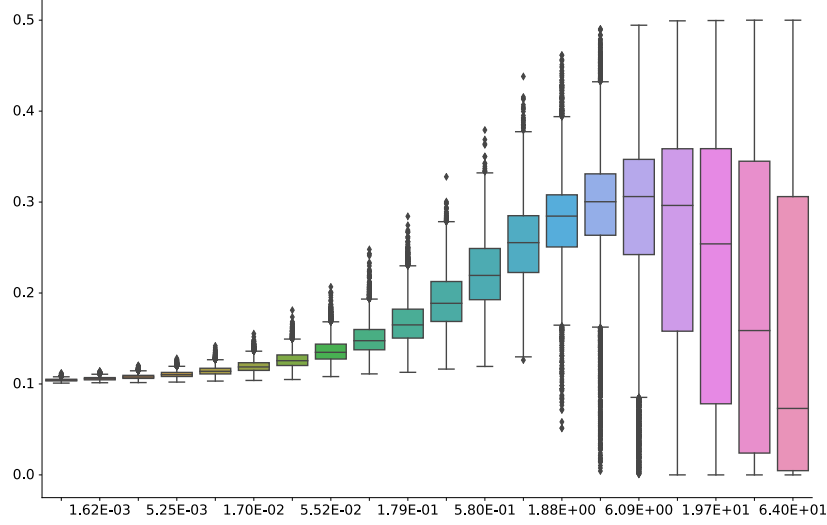


Figure 4: **Distribution of $\lambda_{max}(\mathbf{H}_g)$ as a function of q^* :** In general, increasing the variance of the distribution of h^g does not result in a monotonic increase in the spectral radius of the Hessian of the GLM layer. We plot the distribution of the maximum eigenvalues as a function of the variance of the softmax layer obtained from factorizing 10,000 random matrices.

6.3 Montecarlo estimate of spectral radius of \mathbf{H}_g for 10 way softmax classification

6.4 Manifold Optimization

The potentially non-convex constraint set constitutes a Riemannian manifold, when it is locally isomorphic to \mathbb{R}^n , differentiable and endowed with a suitable (Riemannian) metric, which allows us to measure distances in the tangent space and consequentially also define distances on the manifold. There is considerable freedom in choosing a Riemannian metric; here we consider the metric inherited from the Euclidean embedding space which is defined as $\langle \mathbf{W}, \mathbf{W}' \rangle \triangleq \text{Tr}(\mathbf{W}'^\top \mathbf{W})$. To optimize a cost function with respect to parameters lying in a non-Euclidean manifold we must define a descent direction. This is done by defining a manifold equivalent of the directional derivative. An intuitive approach replaces the movement along a vector \mathbf{t} with movement along a geodesic curve $\gamma(t)$, which lies in the manifold and connects two points $\mathbf{W}, \mathbf{W}' \in \mathcal{M}$ such that $\gamma(0) = \mathbf{W}$, $\gamma(1) = \mathbf{W}'$. The derivative of an arbitrary smooth function $f(\gamma(t))$ with respect to t then defines a tangent vector for each t .

Tangent vector $\xi_{\mathbf{W}}$ is a tangent vector at \mathbf{W} if $\xi_{\mathbf{W}}$ satisfies $\gamma(0) = \mathbf{W}$ and

$$\xi_{\mathbf{W}} \triangleq \left. \frac{df(\gamma(t))}{dt} \right|_{t=0} \triangleq \gamma'(0)f \quad (41)$$

The set of all tangents to \mathcal{M} at \mathbf{W} is referred to as the tangent space to \mathcal{M} at \mathbf{W} and is denoted by $T_{\mathbf{W}}\mathcal{M}$. The geodesic importantly is then specified by a constant velocity curve $\gamma''(t) = 0$ with initial velocity $\xi_{\mathbf{W}}$. To perform a gradient step, we must then move along $\xi_{\mathbf{W}}$ while respecting the manifold constraint. This is achieved by applying the exponential map defined as $\text{Exp}_{\mathbf{W}}(\xi_{\mathbf{W}}) \triangleq \gamma(1)$, which moves \mathbf{W} to another point \mathbf{W}' along the geodesic. While certain manifolds, such as the Oblique manifold, have efficient closed-form exponential maps, for general Riemannian manifolds, the computation of the exponential map involves numerical solution to a non-linear ordinary differential equation [1]. An efficient alternative to numerical integration is given by an orthogonal projection onto the manifold. This projection is formally referred to as a retraction $\text{Rt}_{\mathbf{W}} : T_{\mathbf{W}}\mathcal{M} \rightarrow \mathcal{M}$.

Finally, gradient methods using Polyak (heavy ball) momentum (e.g. ADAM [16]) require the iterative updating of terms which naturally lie in the tangent space. The parallel translation $\mathcal{T}_{\zeta}(\xi) : T_{\mathbf{W}}\mathcal{M} \oplus T_{\mathbf{W}}\mathcal{M} \rightarrow T_{\mathbf{W}'}\mathcal{M}$ generalizes vector composition from Euclidean to non-Euclidean manifolds, by moving the tangent ξ along the geodesic with initial velocity $\zeta \in \mathcal{T}$ and endpoint \mathbf{W}' , and then projecting the resulting vector onto the tangent space $T_{\mathbf{W}'}\mathcal{M}$. As with the exponential map, parallel transport \mathcal{T} may require the solution of non-linear ordinary differential equation. To alleviate the computational burden, we consider *vector transport* as an effective, projection-like solution to the

parallel translation problem. We overload the notation and also denote it as \mathcal{T} , highlighting the similar role that the two mappings share. Technically, the geodesics and consequentially the exponential map, retraction as well as transport \mathcal{T} depend on the choice of the Riemannian metric. Putting the equations together the updating scheme for Riemannian stochastic gradient descent on the manifold is

$$\mathbf{W}_{t+1} = \Pi_{\mathbf{W}_t}(-\eta_t \text{grad} f) \quad (42)$$

where Π is either the exponential map Exp or the retraction Rt and $\text{grad} f$ is the gradient of the function $f(\mathbf{W})$ lying in the tangent space $T_{\mathbf{W}}\mathcal{M}$.

6.4.1 Optimizing over the Oblique manifold

Cho and Lee [7] proposed an updating scheme for optimizing neural networks where the weights of each layer are constrained to lie in the oblique manifold $\text{Ob}(p, n)$. Using the fact that the manifold itself is a product of p unit-norm spherical manifolds, they derived an efficient, closed-form Riemannian gradient descent updating scheme. In particular the optimization simplifies to the optimization over $\text{Ob}(1, n)$ for each column $\mathbf{w}_{i \in \{1, \dots, p\}}$ of \mathbf{W} .

Oblique gradient The gradient $\text{grad} f$ of the cost function f with respect to the weights lying in $\text{Ob}(1, n)$ is given as a projection of the Euclidean gradient $\text{Grad} f$ onto the tangent at \mathbf{w}

$$\text{grad} f = \text{Grad} f - (\mathbf{w}^\top \text{Grad} f) \mathbf{w} \quad (43)$$

Oblique exponential map The exponential map $\text{Exp}_{\mathbf{w}}$ moving \mathbf{w} to \mathbf{w}' along a geodesic with initial velocity $\xi_{\mathbf{w}}$

$$\text{Exp}_{\mathbf{w}} = \xi_{\mathbf{w}} \cos(\|\mathbf{w}\|) + \frac{\mathbf{w}}{\|\mathbf{w}\|} \sin(\|\mathbf{w}\|) \quad (44)$$

Oblique parallel translation The parallel translation \mathcal{T} moves the tangent vector $\xi_{\mathbf{w}}$ along the geodesic with initial velocity $\zeta_{\mathbf{w}}$

$$\begin{aligned} \mathcal{T}_{\zeta_{\mathbf{w}}}(\xi_{\mathbf{w}}) &= \xi_{\mathbf{w}} - \\ &\quad \frac{\zeta_{\mathbf{w}}}{\|\zeta_{\mathbf{w}}\|} ((1 - \cos(\|\zeta_{\mathbf{w}}\|)) + \mathbf{w} \sin(\|\zeta_{\mathbf{w}}\|)) \frac{\zeta_{\mathbf{w}}}{\|\zeta_{\mathbf{w}}\|}^\top \xi_{\mathbf{w}} \end{aligned} \quad (45)$$

6.4.2 Optimizing over the Stiefel manifold

Optimization over Stiefel manifolds in the context of neural networks has been studied by [10, 32, 31]. Unlike [32, 31] we propose the parametrization using the Euclidean metric, which results in a different definition of vector transport.

Stiefel gradient The gradient $\text{grad} f$ of the cost function f with respect to the weights lying in $\text{St}(p, n)$ is given as a projection of the Euclidean gradient $\text{Grad} f$ onto the tangent at \mathbf{W} [9, 1]

$$\begin{aligned} \text{grad} f &= (\mathbf{I} - \mathbf{W}\mathbf{W}^\top) \text{Grad} f \\ &\quad + \frac{1}{2} \mathbf{W} (\mathbf{W}^\top \text{Grad} f - \text{Grad} f^\top \mathbf{W}) \end{aligned} \quad (46)$$

Stiefel retraction The retraction $\text{Rt}_{\mathbf{W}}(\xi_{\mathbf{W}})$ for the Stiefel manifold is given by the Q factor of the QR decomposition [1].

$$\text{Rt}_{\mathbf{W}}(\xi_{\mathbf{W}}) = \text{qf}(\mathbf{W} + \xi_{\mathbf{W}}) \quad (47)$$

Stiefel vector transport The vector transport \mathcal{T} moves the tangent vector $\xi_{\mathbf{w}}$ along the geodesic with initial velocity $\zeta_{\mathbf{w}}$ for $\mathbf{W} \in \text{St}(p, n)$ endowed with the Euclidean metric.

$$\mathcal{T}_{\zeta_{\mathbf{w}}}(\xi_{\mathbf{w}}) = (\mathbf{I} - \mathbf{Y}\mathbf{Y}^\top) \xi_{\mathbf{w}} + \frac{1}{2} \mathbf{Y} (\mathbf{Y}^\top \xi_{\mathbf{w}} - \xi_{\mathbf{w}}^\top \mathbf{Y}) \quad (48)$$

where $\mathbf{Y} \triangleq \text{Rt}_{\mathbf{W}}(\zeta_{\mathbf{W}})$. It is easy to see that the transport \mathcal{T} consists of a retraction of tangent $\zeta_{\mathbf{W}}$ followed by the orthogonal projection of $\eta_{\mathbf{W}}$ at $\text{Rt}_{\mathbf{W}}(\zeta_{\mathbf{W}})$. The projection is the same as the one mapping $\text{P} : \text{Grad} f \rightarrow \text{grad} f$ in equation 46.

6.4.3 Optimizing over non-compact manifolds

The critical weight initialization yielding a singular spectrum of the Jacobian tightly concentrating on 1 implies that a substantial fraction of the pre-activations lie in expectation in the linear regime of the squashing nonlinearity and as a consequence the network acts quasi-linearly. To relax this constraint during training we allow the scales of the manifold constrained weights to vary. We chose to represent the weights as a product of a scaling diagonal matrix and a matrix belonging to the manifold. Then the optimization of each layer consists in the optimization of the two variables in the product. In this work we only consider isotropic scalings, but the method generalizes easily to the use of any invertible square matrix.

6.5 FIM and NTK have the same spectrum

The empirical Neural Tangent Kernel (NTK) Recall the definition in equation 24:

$$\hat{\Theta}_{t,i,j} \triangleq \mathbf{J}_{\mathbf{x}_0}^{\mathbf{h}^g} \mathbf{J}_{\mathbf{x}_0}^{\mathbf{h}^g \top} \quad (49)$$

which gives a $N^g|\mathcal{D}|$ by $N^g|\mathcal{D}|$ kernel matrix. By comparison the empirical Fisher Information matrix with a Gaussian likelihood is

$$\sum_{i=1^{|\mathcal{D}|}} \mathbf{J}_{\theta}^{\mathbf{h}^g \top} \nabla_{\mathbf{h}^g}^2 \mathcal{L} \mathbf{J}_{\theta}^{\mathbf{h}^g} \quad (50)$$

To see that the spectra of these two coincide consider the third order tensor underlying both $\mathbf{J}_{\mathbf{h}^g}^{\mathbf{h}^g}$ for $i \in 1 \dots |\mathcal{D}|$, additionally consider and unfolding \mathbf{A} with dimensions $|\theta|$ by $N^g|\mathcal{D}|$; i.e. we construct a matrix with dimension of number of parameters by number of outputs times number of data points. Then

$$\bar{\mathbf{G}} = \mathbf{A}^\top \mathbf{A} \quad (51)$$

$$\hat{\Theta} = \mathbf{A} \mathbf{A}^\top \quad (52)$$

$$(53)$$

and their spectra trivially coincide.

Remark. *It is interesting to note that when the Fisher information metric and NTK are applied to a regression problem with Gaussian noise then the relation between admits the following interpretation. For $\mathcal{L} = \frac{1}{2} \|\hat{y} - y\|^2$ the Fisher information matrix $\bar{\mathbf{G}}$ is the Riemannian metric on the tangent bundle and $\hat{\Theta}$ is the Riemannian metric on the co-tangent bundle.*