ISONETRY: GEOMETRY OF CRITICAL INITIALIZATIONS AND TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work on critical initializations of deep neural networks has shown that by constraining the spectrum of input-output Jacobians allows for fast training of very deep networks without skip connections. The current understanding of this class of initializations is limited with respect to classical notions from optimization. In particular, the connections between Jacobian eigenvalues and curvature of the parameter space are unknown. Similarly, there is no firm understanding of the effects of maintaining orthogonality during training. With this work we complement the existing understanding of critical initializations and show that the curvature is proportional to the maximum singular value of the Jacobian. Furthermore we show that optimization under orthogonality constraints ameliorates the dependence on choice of initial parameters, but is not strictly necessary.

1 INTRODUCTION

Deep neural networks have been proven extremely powerful, achieving empirical success in a vast array of problems, ranging from image recognition (He et al., 2015), amortized probabilistic inference (Ritchie et al., 2016) to inferring the dynamics of neural data (Pandarinath et al., 2018). The practical hindrance in their application often stems from the difficulty in training them, both due to the excessive computational cost of running many epochs of gradient descent but also due to the inherent instability in gradient computation. It is therefore of great practical and theoretical interest to devise effective gradient optimization techniques for deep neural networks. A new and promising approach exploits mean field assumptions and random matrix theory to devise initialization strategies that ensure the boundedness of the backpropagated gradients (Schoenholz et al., 2016; Pennington et al., 2017). In particular, Pennington et al. shows that for orthogonal networks with appropriately chosen parameters, the hidden layer input-output Jacobian matrix is nearly isometric, approximately preserving the ℓ_2 norm of the gradients. Despite achieving an impressive increase in training speed, the exact mechanism by which this initialization confers its advantage is not well understood. In particular, it invites questions from an optimization perspective on how the boundedness of the Jacobian matrix relates to notions such as gradient smoothness, or (negative) strong convexity that describe the optimization landscape.

Recently, there have been several orthogonal motivations ranging from optimization speed, robustness against adversarial examples, and improving quality of generated samples for orthogonal initialization procedures (Xie et al., 2017; Ozay & Okatani, 2016; Cisse et al., 2017; Odena et al., 2018). All of which raise the question about the relative importance of maintaining orthogonality during training as compared to critical initializations proposed in (Pennington et al., 2017). In this work we study simple feed-forward, fully connected networks. We make theoretical advances in understanding the connection between eigenvalues of the Jacobian and local measures of gradient smoothness. Subsequently we go onto analyze experimentally, through the lens of manifold optimization the importance of maintaining orthogonality or near orthogonality throughout training of deep neural networks with 200 layers. This allows us to show that nearly isometric networks optimize worse because their gradients rapidly become less smooth. In contrast, less isometric networks gradually degrade in smoothness, which allows them to be trained with a higher learning rate and converge faster. Moreover, we show that networks trained with orthogonality and near orthogonality constraints are considerably less sensitive to initialization and optimize rapidly regardless. Contrary to the recent conjecture by Santurkar et al. (2018), this effect is not attributable to an increase in gradient smoothness. Our work suggests that maintaining near orthogonality constraint throughout

training provides robust performance in practice compared to the orthogonal weight initialization scheme which requires a highly specific parameter tuning. This offers insight into the role of Weight Normalization (Salimans & Kingma, 2016), of which the near orthogonal constraint can be seen as a variant.

2 BACKGROUND

2.1 FORMAL DESCRIPTION OF THE NETWORK

Following (Pennington et al., 2017; 2018; Schoenholz et al., 2016), we consider a feed-forward, fully connected neural network with L hidden layers. Each layer $l \in \{1, \ldots, L\}$ is given as a recursion of the form

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

where $\mathbf{x}^l$ are the activations, $\mathbf{h}^l$ are the pre-activations, $\mathbf{W}^l \in \mathbb{R}^{N \times N}$ are the weight matrices, $\mathbf{b}^l$ are the bias vectors and $\phi(\cdot)$ is the activation function. For consistency, 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 a link function and $\mathbf{h}^g = \mathbf{W}^g x^L + \mathbf{b}^g$. The hidden layer input-output Jacobian matrix $\mathbf{J}_{x^0}^{x^L}$ is given by,

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

where $\mathbf{D}^l$ is a diagonal matrix with entries $\mathbf{D}_{i,i}^l = \phi'(\mathbf{h}_i^l)$. As pointed out in (Pennington et al., 2017; Schoenholz et al., 2016), the conditioning of the Jacobian matrix affects the conditioning of the back-propagated gradients for all layers.

2.2 INITIALIZATION STRATEGY FOR MAXIMIZING SIGNAL PROPAGATION

Extending the classic result on the Gaussian process limit for wide layer width obtained by Neal (1996), recent work (Matthews et al., 2018; Lee et al., 2017) has shown that for deep untrained networks with elements of their weight matrices $\mathbf{W}_{i,j}$ drawn from a Gaussian distribution $\mathcal{N}(0, \frac{\sigma_{\mathbf{W}}^2}{N})$ 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 of the width $N \to \infty$. Under this mean-field condition the variance the of the distribution is defined recursively given the pre-activations in the preceding layer:

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

where $\mathrm{d}\mu(h)$ denotes the standard Gaussian measure $\frac{\mathrm{d}h}{\sqrt{2\pi}} \exp\left(\frac{-h^2}{2}\right)$. The variance of the pre-activations of the first layer q^1 depends on the ℓ_2^2 norm of the inputs $q^1 = \frac{\sigma_{\mathbf{W}}^2}{N} \left\|x^0\right\|_2^2 + \sigma_{\mathbf{b}}^2$. The recursion defined in equation 3 has a fixed point q^*,

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

which can be satisfied for all layers by appropriately choosing $\sigma_{\mathbf{W}}, \sigma_{\mathbf{b}}$ and scaling the input $\mathbf{x}^0$ accordingly. In order to ensure that $\mathbf{J}_{\mathbf{x}^0}^{\mathbf{x}^L}$ is well conditioned, Pennington et al. (2017) require that in addition to the variance of pre-activation being constant for all layers, two additional constraints are met. Firstly, they require that the mean square singular value of $\mathbf{DW}$ for each layer have a certain value in expectation.

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

Given that the mean squared singular value of the Jacobian matrix $\mathbf{J}_{x^0}^{x^L}$ is $(\chi)^L$, they require that $\chi = 1$ which corresponds to a critical initialization where the gradients are asymptotically stable as $L \to \infty$. Secondly, they require that the maximal squared singular value s_{max}^2 of the Jacobian $\mathbf{J}_{x^0}^{x^L}$ be bounded. In (Pennington et al., 2017) it was shown 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 $L \to \infty$ (Pennington et al., 2018) and for piecewise-linear $\phi(.)$, it is analytically derivable and admits a solution in q^* for $s_{max}(\mathbf{J}_{x^0}^{x^L}) = 1$ for any choice of L. For arbitrary $\phi(.)$'s, s_{max} can be obtained numerically from the solution of a functional equation describing the density of singular values.

The described, theoretically derived intialization scheme has been tested and has shown a substantial speed up in training times on CIFAR-10. The impressive results invite further inquest into the connection between the class of critical initializations with bounded maximal singular values of the Jacobian matrix and optimization, as well as the specific conditions under which the Gaussian limit on pre-activations holds. Particularly, we show that for random neural networks the maximum singular value of the Jacobian matrix is related to the curvature of the parameter space as measured by the Fisher information matrix $\bar{\mathbf{G}}$, which in turn affects the upper bound on the maximum stable learning rate (Bottou, 2010). Subsequently, using manifold optimization we analyze how enforcing orthogonality affects the convergence speed. Finally, we show that for random networks with orthogonal weights pre-activations do not necessarily converge in distribution to a Gaussian, raising questions about the conditions under which the mean field approximation holds.

3 RESULTS

3.1 CONNECTION BETWEEN $s_{max}(\mathbf{J}_{x^0}^{x^L})$ AND THE MAXIMUM EIGENVALUE $\lambda_{max}(\bar{\mathbf{G}})$

Seen from a probabilistic perspective, the output of a neural network defines a conditional probability distribution $p_\theta(\mathbf{y}|\mathbf{x}^0)$, where $\theta = \{\text{vec}(\mathbf{W}^1), \ldots, \text{vec}(\mathbf{W}^L), \mathbf{b}^1, \ldots, \mathbf{b}^L\}$ is the set of all hidden layer parameters (Botev et al., 2017; Amari, 2016). In this context, the Fisher information matrix defined as,

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

can be expanding using the chain rule:

$$\bar{\mathbf{G}} = \mathbb{E}_{\mathbf{x}^0,\mathbf{y}}\left[\mathbf{J}_\theta^{\mathbf{x}^L\top} \mathbf{W}^{g\top} \nabla_{\mathbf{h}^g} \log p_\theta(\mathbf{y}|\mathbf{x}^0) \nabla_{\mathbf{h}^g} \log p_\theta(\mathbf{y}|\mathbf{x}^0)^\top \mathbf{W}^g \mathbf{J}_\theta^{\mathbf{x}^L}\right]. \tag{7}$$

Each block of the Fisher information matrix with respect to parameters $a, b \in \theta$ can further be expressed as

$$\bar{\mathbf{G}}_{a,b} = \mathbb{E}_{\mathbf{x}^0}\left[\mathbf{J}_a^{\mathbf{h}^\alpha\top} \mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{x}^L\top} \mathbf{W}^{g\top} \mathbf{H}_g \mathbf{W}^g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{x}^L} \mathbf{J}_b^{\mathbf{h}^\beta}\right] \tag{8}$$

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, provided it is twice differentiable and it does not depend on $\mathbf{y}$, which also allows us to drop $\mathbf{y}$ from the expectation. 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 $\mathbf{W}^\alpha$ and $\mathbf{b}^\alpha$ as:

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

We can further simplify the expression for the blocks of the Fisher information matrix equation 8, using the fact that the pre-activations converge weakly to an isotropic Gaussian distribution in the limit of wide layers (Matthews et al., 2018; Lee et al., 2017). This justifies assuming independence of the pre-activations of all the layers. We can additionally assume for convenience that the input to the network $\mathbf{x}^0$ has been whitened before being scaled by q^*. The simplified expressions are as follows:

$$\bar{\mathbf{G}}_{\text{vec}(\mathbf{W}^\alpha),\text{vec}(\mathbf{W}^\beta)} = \mathbb{E}\left[\mathbf{x}^{\alpha-1}\mathbf{x}^{\beta-1\top} \otimes \mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{x}^L\top} \mathbf{W}^{g\top} \mathbf{H}_g \mathbf{W}^g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{x}^L}\right] \tag{10}$$

$$= q^{*2}\mathbf{I} \otimes \mathbb{E}\left[\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{x}^L\top} \mathbf{W}^{g\top} \mathbf{H}_g \mathbf{W}^g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{x}^L}\right] \tag{11}$$

$$\bar{\mathbf{G}}_{\mathbf{b}^\alpha,\mathbf{b}^\beta} = \mathbb{E}\left[\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{x}^L\top} \mathbf{W}^{g\top} \mathbf{H}_g \mathbf{W}^g \mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{x}^L}\right] \tag{12}$$

$$\bar{\mathbf{G}}_{\mathrm{vec}(\mathbf{W}^\alpha),\mathbf{b}^\beta} = \mathbb{E}\left[\left(\mathbf{x}^{\alpha-1\top} \otimes \mathbf{I}\right)\left(\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{x}^L\top}\mathbf{W}^{g\top}\mathbf{H}_g\mathbf{W}^g\mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{x}^L}\right)\right] \tag{13}$$

$$= \underbrace{\mathbb{E}\left[\mathbf{x}^{\alpha-1\top} \otimes \mathbf{I}\right]}_{=0}\mathbb{E}\left[\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{x}^L\top}\mathbf{W}^{g\top}\mathbf{H}_g\mathbf{W}^g\mathbf{J}_{\mathbf{h}^\beta}^{\mathbf{x}^L}\right] = 0 \tag{14}$$

We then consider a block diagonal approximation to the Fisher information matrix. Under this simplification the maximum eigenvalue of the the Fisher information matrix is $\lambda_{max}(\bar{\mathbf{G}}) \approx \max_a \lambda_{max}(\bar{\mathbf{G}}_{a,a})$. The quality of this approximation is given by the generalization of the Gershgorin circle theorem to block-partitioned matrices, which is detailed in the Appendix 4.1. Following equation 10, each diagonal block a with respect to the weight matrices $\mathbf{W}^\alpha$ is bounded by $\left(s_{max}^2(\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{x}^L})s_{max}^2(\mathbf{W}^g)s_{max}(\mathbf{H}_g)\right)$, while the diagonal blocks a with respect to the biases $\mathbf{b}^\alpha$ are bounded by $\left(q^{*2}s_{max}^2(\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{x}^L})s_{max}^2(\mathbf{W}^g)s_{max}(\mathbf{H}_g)\right)$ using equation 12. These results can be further simplified if we initialize $\mathbf{W}^g$ as a scaled semi-orthogonal matrix such that $\mathbf{W}^{g\top}\mathbf{W}^g = \sigma_{\mathbf{W}}^2\mathbf{I}$ and consider a neural network regression model with mean-square loss. Then the respective bounds become $s_{max}^2(\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{x}^L})\sigma_{\mathbf{W}}^2$ and $q^{*2}s_{max}^2(\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{x}^L})\sigma_{\mathbf{W}}^2$. Assuming $s_{max}(\mathbf{J}_{\mathbf{h}^\alpha}^{\mathbf{x}^L})$ is a monotonically increasing function of L it is sufficient to ensure that $s_{max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{x}^L})$ is small in order for $\lambda_{max}(\bar{G})$ to be small. Furthermore since $q^* \ll 1$ and since $s_{max}(\mathbf{J}_{\mathbf{x}^0}^{\mathbf{x}^L})$ is at most $\sigma_{\mathbf{W}}$ larger than $s_{max}(\mathbf{J}_{\mathbf{h}^1}^{\mathbf{x}^L})$, bounding the maximum value of the hidden layer input-out Jacobian bounds the maximum eigenvalue of the Fisher information matrix.

3.2 Optimization over manifolds

Optimizing neural network weights subject to manifold constraints has recently attracted considerable interest, with several lines of research focusing on either the approximate preservation of gradient norms in recurrent neural networks (Arjovsky et al., 2015; Henaff et al., 2016; Vorontsov et al., 2017), increasing the robustness to adversarial examples (Cisse et al., 2017), or heuristically motivated improvements to prediction accuracy (Xie et al., 2017; Cho & Lee, 2017; Ozay & Okatani, 2016). 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-out Jacobian and of the Fisher information matrix.

We briefly review notions from differential geometry and optimization over matrix manifolds (Edelman et al., 1998; Absil et al., 2007), in order to lay ground for a discussion of the specifics of the optimization techniques used in our experiments. Informed readers are encouraged to skip to Sec. 3.2.1. 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 $\langle \mathbf{W}, \mathbf{W}' \rangle \triangleq \mathrm{Tr}(\mathbf{W}'^\top\mathbf{W})$.

Stiefel Manifold Let $\mathrm{St}(p,n) \quad (p \leq n)$ denote the set of all $n \times p$ orthonormal matrices

$$\mathrm{St}(p,n) \triangleq \{\mathbf{W} \in \mathbb{R}^{n\times p} : \mathbf{W}^\top\mathbf{W} = \mathbf{I}_p\} \tag{15}$$

Notice that for $p = n$, the Stiefel manifold parametrizes the set of all orthogonal matrices.

Oblique Manifold Let $\mathrm{Ob}(p,n)$ denote the set of all $n \times p$ matrices with unit norm columns

$$\mathrm{Ob}(p,n) \triangleq \{\mathbf{W} \in \mathbb{R}^{n\times p} : \mathrm{diag}(\mathbf{W}^\top\mathbf{W}) = \mathbf{1}\} \tag{16}$$

where $\mathrm{diag}(\mathbf{M})$ denotes an operator that extracts the diagonal entries of $\mathbf{M}$. This manifold is equivalent to the direct product p unit-norm spheres $\mathbb{S}^{n-1} \times \mathbb{S}^{n-1} \times \cdots \times \mathbb{S}^{n-1}$.

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 $f(\gamma(t))$ 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

$$\xi_{\mathbf{W}} \triangleq \frac{\mathrm{d}f(\gamma(t))}{\mathrm{d}t}\bigg|_{t=0} \triangleq \gamma'(0)f \tag{17}$$

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 $\mathrm{Exp}_{\mathbf{W}}(\xi_{\mathbf{W}}) \triangleq \gamma(1)$, 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 (Absil et al., 2007). An efficient alternative to numerical integration is given by an orthogonal projection onto the manifold. This projection is formally referred to as a retraction $\mathrm{Rt}_{\mathbf{W}} : T_{\mathbf{W}}\mathcal{M} \to \mathcal{M}$.
Finally, gradient methods using Polyak (heavy ball) momentum (e.g. ADAM (Kingma & Ba, 2014)) require the iterative updating of terms which naturally lie in the tangent space. The parallel translation $\mathcal{T}_{\zeta}(\xi) : T\mathcal{M} \bigoplus T\mathcal{M} \to T\mathcal{M}$ 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 Riemmanian stochastic gradient descent on the manifold is

$$\mathbf{W}_{t+1} = \Pi_{\mathbf{W}_t}(-\eta_t \,\mathrm{grad} f) \tag{18}$$

where Π is either the exponential map Exp or the retraction Rt and $\mathrm{grad}f$ is the gradient of the function $f(\mathbf{W})$ lying in the tangent space $T_{\mathbf{W}}\mathcal{M}$. The updating scheme for optimizers using momentum, as mentioned before, additionally requires updating the momentum term using parallel transport.

3.2.1 Optimizing over the Oblique manifold

Cho & Lee (2017) proposed an updating scheme for optimizing neural networks where the weights of each layer are constrained to lie in the oblique manifold $\mathrm{Ob}(p, n)$. 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 $\mathrm{Ob}(1, n)$ for each column $\mathbf{w}_{i\in\{1,\dots,p\}}$ of $\mathbf{W}$.

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

$$\mathrm{grad}f = \mathrm{Grad}f - (\mathbf{w}^{\top}\mathrm{Grad}f)\mathbf{w} \tag{19}$$

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

$$\mathrm{Exp}_{\mathbf{w}} = \xi_{\mathbf{w}} \cos(\|\mathbf{w}\|) + \frac{\mathbf{w}}{\|\mathbf{w}\|}\sin(\|\mathbf{w}\|) \tag{20}$$

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

$$\mathcal{T}_{\zeta_{\mathbf{w}}}(\xi_{\mathbf{w}}) = \xi_{\mathbf{w}} - \left(\frac{\zeta_{\mathbf{w}}}{\|\zeta_{\mathbf{w}}\|}((1 - \cos(\|\zeta_{\mathbf{w}}\|)) + \mathbf{w}\sin(\|\zeta_{\mathbf{w}}\|))\frac{\zeta_{\mathbf{w}}}{\|\zeta_{\mathbf{w}}\|}^{\top}\xi_{\mathbf{w}}\right) \tag{21}$$

Cho & Lee (2017) 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.

$$\rho(\lambda, \mathbf{W}) = \frac{\lambda}{2} \left\| \mathbf{W}^\top \mathbf{W} - \mathbf{I} \right\|_F^2 \tag{22}$$

3.2.2 Optimizing over the Stiefel manifold

Optimization over Stiefel manifolds in the context of neural networks has been studied by (Harandi & Fernando, 2016; Wisdom et al., 2016; Vorontsov et al., 2017). However, (Wisdom et al., 2016; Vorontsov et al., 2017) proposed a different parametrization resulting from an alternative choice of the Riemannian metric. Here we propose the parametrization using the Euclidean metric, which results in a different definition of vector transport.

Stiefel gradient The gradient $\mathrm{grad} f$ of the cost function f with respect to the weights lying in $\mathrm{St}(p, n)$ is given as a projection of the Euclidean gradient $\mathrm{Grad} f$onto the tangent at $\mathbf{W}$ (Edelman et al., 1998; Absil et al., 2007)

$$\mathrm{grad} f = (\mathbf{I} - \mathbf{W}\mathbf{W}^\top)\mathrm{Grad} f + \frac{1}{2}\mathbf{W}\left(\mathbf{W}^\top \mathrm{Grad} f - \mathrm{Grad} f^\top \mathbf{W}\right) \tag{23}$$

Stiefel retraction The retraction $\mathrm{Rt}_{\mathbf{W}}(\xi_{\mathbf{W}})$ for the Stiefel manifold is given by the Q factor of the QR decomposition (Absil et al., 2007).

$$\mathrm{Rt}_{\mathbf{W}}(\xi_{\mathbf{W}}) = \mathrm{qf}(\mathbf{W} + \xi_{\mathbf{W}}) \tag{24}$$

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 $\mathbf{W} \in \mathrm{St}(p, n)$ endowed with the Euclidean metric.

$$\mathcal{T}_{\zeta_{\mathbf{w}}}(\xi_{\mathbf{w}}) = \left(\mathbf{I} - \mathbf{Y}\mathbf{Y}^\top\right)\xi_{\mathbf{W}} + \frac{1}{2}\mathbf{Y}\left(\mathbf{Y}^\top \xi_{\mathbf{W}} - \xi_{\mathbf{W}}^\top \mathbf{Y}\right) \tag{25}$$

where $\mathbf{Y} \triangleq \mathrm{Rt}_{\mathbf{W}}(\zeta_{\mathbf{W}})$. 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 $\mathrm{Rt}_{\mathbf{W}}(\zeta_{\mathbf{W}})$. The projection is the same as the one mapping $\mathrm{P} : \mathrm{Grad} f \rightarrow \mathrm{grad} f$ in equation 23.

3.3 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 matrix and 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.

3.4 Numerical Experiments

To experimentally test the potential effect of maintaining orthogonality throughout training and compare it to the unconstrained optimization (Pennington et al., 2017), we trained a 200 layer network tanh network on CIFAR-10 and SVHN. We present the results for the former in the main text, while the latter can be found in the Appendix 4.2 in the interest of space. Following (Pennington et al., 2017) we set the width of each layer to be $N = 400$ and chose the $\sigma_{\mathbf{W}}$, $\sigma_{\mathbf{b}}$ in way to ensure that both χ concentrates on 1 but $s^2_{\max}$ varies as a function of q^* (see Fig. 1). 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 (Pennington et al., 2017). To test how enforcing strict orthogonality or near orthogonality affects convergence speed, and the spectrum of hidden layer input-output Jacobians and the maximum eigenvalues of the Fisher information matrix, we trained Stiefel and Oblique constrained networks and compared them

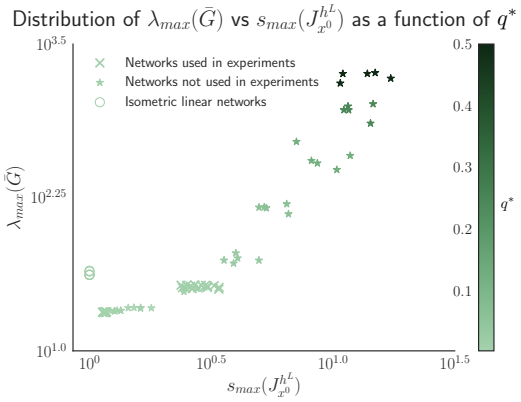


Figure 1: **At initialization the maximum eigenvalue of the Fisher information matrix $\bar{\mathbf{G}}$ correlates highly with the maximum singular value of the Jacobian $\mathbf{J}_{x^0}^{x^L}$.** While the dependence is not perfectly quadratic as predicted, the deviation can be understood in terms of the bound implied by the Gershgorin theorem for block matrices – the bound becomes tight in the limit of $q^* \to \infty$. The networks not used in the experiments were randomly initialized with $\chi = 1$ and q^* varying uniformly on a logarithmical grid between 9×10^{-4} and 0.5.

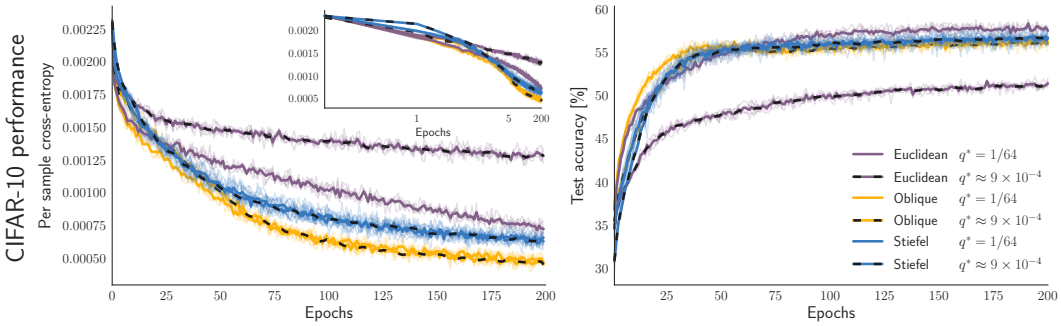


Figure 2: **Manifold constrained networks are insensitive to the choice of q^***: Train loss and test accuracy for Euclidean, Stiefel and Oblique networks with two different values of q^*. The manifold constrained networks minimize the training loss at approximately the same rate, being faster than both 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.

to the unconstrained "Euclidean" network described in (Pennington et al., 2017). We used a Riemannian version of ADAM (Kingma & Ba, 2014). 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 the groups, as well as the the non-orthogonality penalty (see 22) for Oblique networks were chosen with 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 the maximum eigenvalue of the Fisher information matrix derived in Section 3.1, we predicted that at initialization $\lambda_{max}(\bar{\mathbf{G}})$ should covary with $\sigma^2_{max}(\mathbf{J}_{\mathbf{x}^0}^{\mathbf{x}^L})$. Our prediction is vindicated in that we find a strong, significant correlation between the two, with a Pearson coefficient of $\rho = 0.88$. The numerical values are presented in Fig. 1. Additionally we see that both the maximum singular value and maximum eigenvalue increase monotonically as a function of q^*. Based on the previous work by Saxe et al. (2013) 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 using the following scheme applied to critical, non-linear networks with $q^* = 1/64$. Surprisingly, the deep linear networks had a substantially larger $\lambda_{max}(\bar{\mathbf{G}})$ than its non-linear counterparts initialized with identically scaled input (Fig. 1).

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 wish to investigate the effects of initialization on subsequent optimization. As reported by Pennington et al. (2017), learning

speed and generalization peak at intermediate values of $q^* \approx 10^{-0.5}$. This result is counterintuitive 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 (Boyd & Vandenberghe, 2004; Bottou et al., 2016). To gain insight into the effects of choice q^* on convergence rate, we trained the Euclidean networks and estimated the local values of s_{max} and 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 as for before.

We present the results of training in Fig. 2, 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 $q^* \approx 9 \times 10^{-4}$, despite attaining a lower training loss. This latter fact indicates that manifold constrained networks the are perhaps 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 $\lambda_{max}(\bar{\mathbf{G}})$ the initial 5 epochs of optimization (see Fig. 4). 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. 2. 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 (Santurkar et al., 2018). Finally, we observe that for manifold constrained networks the maximum singular value of the Jacobian remains predictive of the maximum eigenvalue of the Fisher information matrix with an average Pearson correlation of 0.93 (see Fig.3). For Euclidean networks the correlation is considerably lower, with $\rho = 0.26$ for $q^* = 1/64$ and 0.83 for $q^* = 9 \times 10^{-4}$.

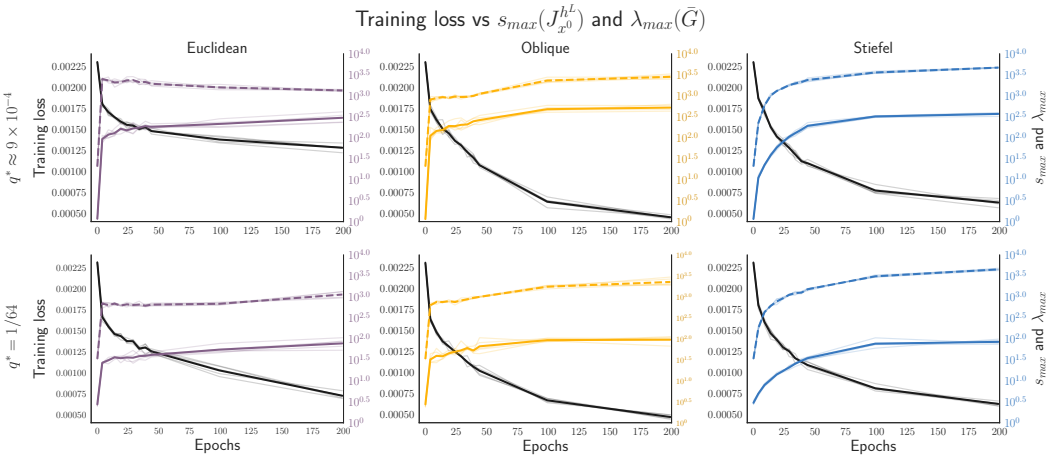


Figure 3: **The maximum singular value of the Jacobian $s_{max}(\mathbf{J}_{x^0}^{x^L})$ is predictive of the maximum eigenvalue of the Fisher information matrix $\lambda_{max}(\bar{\mathbf{G}})$ during training.** The dashed colored lines represent $\lambda_{max}(\bar{\mathbf{G}})$, while the continuous colored lines represent $s_{max}(\mathbf{J}_{x^0}^{x^L})$. The correlation captures particularly well the initial increase in the $\lambda_{max}(\bar{\mathbf{G}})$ during the first few epochs. Moreover, for Stiefel and Oblique networks the correlation is stronger.

3.4.1 Non-Gaussian limit of pre-activation

The derivation of the spectra of hidden layer input-output Jacobians presented in (Pennington et al., 2017; 2018) crucially depends on the existence of the Gaussian limit of the distribution of pre-activations, which in turn assumes that the elements of the weight matrices are sampled iid from some probability measure with finite first two moments. This condition is trivially violated for matrices sampled from the Haar (uniform) measure over the Stiefel manifold. We invoke Theorem 1 from Meckes (2012) which asserts that for a random semi-orthogonal projection of an arbitrary random vector to converge to a Gaussian distribution in the bounded Lipschitz distance, the pro-

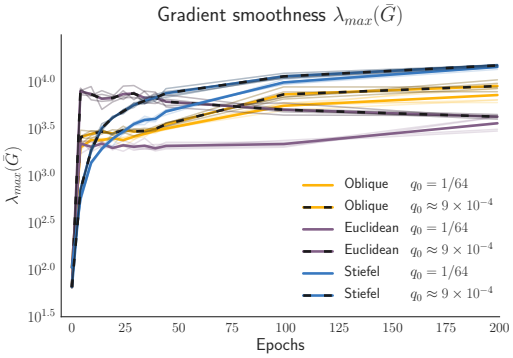


Figure 4: **For manifold constrained networks, gradient smoothness is not predictive of optimization rate. Euclidean networks with a low initial $\lambda_{max}(\bar{G})$ rapidly become less smooth, whereas Euclidean networks with a larger $\lambda_{max}(\bar{G})$ remain relatively smoother.** Notably, the Euclidean network with $q^* = 1/64$ has almost an order of magnitude smaller $\lambda_{max}(\bar{\mathbf{G}})$ than the Stiefel and Oblique networks, but reduces training loss at a slower rate.

jection must go from d input dimensions to $\frac{2\log(d)}{\log(\log(d))}$ dimensions. We also show empirically that in orthogonally initialized random networks, the pre-activations do not necessarily converge to a multivariate Gaussian distribution (see Fig. 5). We pose as an open problem the conditions under which the random matrix argument of Pennington et al. (2017) holds.

Theorem 1. *(Meckes, 2012) Let X be a random vector in $\mathbb{R}^d$ with*

$$\mathbb{E}\left[\|X\|^2\right] = \sigma^2 d, \qquad \mathbb{E}\left[\|X\|^2 \sigma^{-2} - d\right] \leq L\sqrt{d}, \qquad \sup_{\xi \in \mathbb{S}^{d-1}} \mathbb{E}\left[(\xi^\top X)^2\right] \leq 1$$

Let $X_{\mathbf{W}}$ denote the projection $\mathbf{W}X$ for $\mathbf{W} \in \mathrm{St}(k,d)$, with $k = \delta \frac{\log(d)}{\log(\log(d))}$ and $0 \leq \delta \leq 2$, then there is a $c > 0$ such that for $\epsilon = 2\exp\left[-c\frac{\log(\log(d))}{\delta}\right]$ there exists a subset $\mathfrak{I} \subseteq \mathrm{St}(k,d)$ with probability mass $p(\mathfrak{I})$ with respect to the Haar measure. Then $p(\mathfrak{I}) \geq 1 - C\exp\left(-c' d\epsilon^2\right)$ such that for all $\mathbf{W} \in \mathfrak{I}$

$$\sup_{\max(\|f\|_\infty, |f|_L) \leq 1} \left|\mathbb{E}_X\left[f(X_{\mathbf{W}})\right] - \mathbb{E}\left[f(\sigma Z)\right]\right| \leq C'\epsilon$$

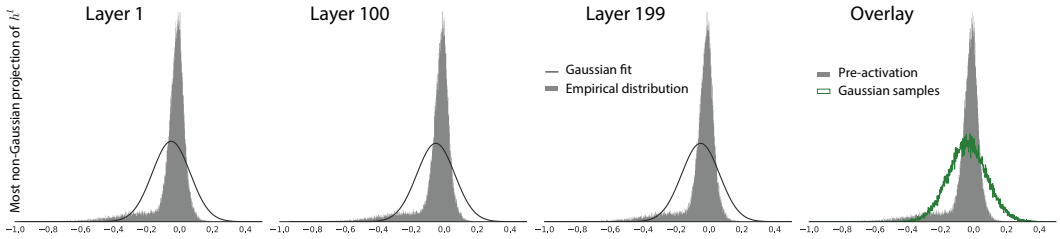


Figure 5: **Non-Gaussian projection of the pre-activations for random networks $\mathbf{q^* = 1/64}$ and evaluated on CIFAR-10.** The projection was obtained using the algorithm in (Blanchard et al., 2006). The rightmost panel shows an overlay of the pre-activations compared to the most non-Gaussian projection of a set of 400 dimensional Gaussian random variables with variance matched to the real data.

4 Discussion

In this work we have analyzed the geometry of critical initializations and during training with orthogonal and near orthogonal manifold constraints. In the process we derived a novel bound on the maximum eigenvalue of the Fisher information matrix and related it to the expected maximum eigenvalue of the hidden layer input-output Jacobian, thereby elucidating the empirical success of the initialization proposed by (Pennington et al., 2017). We then probed numerically the benefits of maintaining orthogonality and near orthogonality of weight matrices during training, while relating them to the gradient smoothness measured by the maximum eigenvalue of the Fisher information matrix. We observed several interesting phenomena, which include a rapid decrease of the gradient smoothness of the Euclidean networks that are smoother at initialization. This paradoxical result

partially explains the observations made in (Pennington et al., 2017) but further analysis needs to be done to understand it fully. Another conclusion is that the robustness to the choice of q^* and low computational overhead makes optimization on the Oblique manifold an appealing alternative to searching for the optimal q^*. This could be particularly advantageous in situations where the input data is non-stationary. Finally we pose an open question the conditions under which the mean field approximation to the pre-activations hold.

ACKNOWLEDGMENTS

REFERENCES

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, N.J. ; Woodstock, December 2007. ISBN 978-0-691-13298-3.

Shun-ichi Amari. *Information Geometry and Its Applications*. Springer, Japan, 1st ed. 2016 edition edition, February 2016. ISBN 978-4-431-55977-1.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary Evolution Recurrent Neural Networks. *arXiv:1511.06464 [cs, stat]*, November 2015. arXiv: 1511.06464.

Gilles Blanchard, Motoaki Kawanabe, Masashi Sugiyama, and Vladimir Spokoiny. In Search of Non-Gaussian Components of a High-Dimensional Distribution. *Journal of Machine Learning Research*, pp. 36, 2006.

Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical Gauss-Newton Optimisation for Deep Learning. In *International Conference on Machine Learning*, pp. 557–565, July 2017.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010. 00829.

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.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization, With Corrections 2008*. Cambridge University Press, Cambridge, UK ; New York, 1 edition edition, March 2004. ISBN 978-0-521-83378-3.

Minhyung Cho and Jaehyung Lee. Riemannian approach to batch normalization. In *Advances in Neural Information Processing Systems*, pp. 5229–5239, 2017.

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.

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. ISSN 0895-4798, 1095-7162. doi: 10.1137/S0895479895290954.

Mehrtash Harandi and Basura Fernando. Generalized backpropagation, Étude De Cas: Orthogonality. *arXiv:1611.05927 [cs]*, November 2016. 00004 arXiv: 1611.05927.

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.

Mikael Henaff, Arthur Szlam, and Yann LeCun. Recurrent Orthogonal Networks and Long-Memory Tasks. *arXiv:1602.06662 [cs, stat]*, February 2016. arXiv: 1602.06662.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, December 2014. 01869 arXiv: 1412.6980.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep Neural Networks as Gaussian Processes. *arXiv:1711.00165 [cs, stat]*, October 2017. arXiv: 1711.00165.

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.

Elizabeth Meckes. Projections of Probability Distributions: A Measure-Theoretic Dvoretzky Theorem. In *Geometric Aspects of Functional Analysis*, Lecture Notes in Mathematics, pp. 317–326. Springer, Berlin, Heidelberg, 2012. ISBN 978-3-642-29848-6 978-3-642-29849-3. doi: 10.1007/978-3-642-29849-3_18.

Radford M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer New York, New York, NY, 1996. ISBN 978-0-387-94724-2 978-1-4612-0745-0. doi: 10.1007/978-1-4612-0745-0.

Augustus Odena, Jacob Buckman, Catherine Olsson, Tom B. Brown, Christopher Olah, Colin Raffel, and Ian Goodfellow. Is Generator Conditioning Causally Related to GAN Performance? *arXiv:1802.08768 [cs, stat]*, February 2018. arXiv: 1802.08768.

Mete Ozay and Takayuki Okatani. Optimization on Submanifolds of Convolution Kernels in CNNs. *arXiv preprint arXiv:1610.07008*, 2016. 00003.

Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D. Stavisky, Jonathan C. Kao, Eric M. Trautmann, Matthew T. Kaufman, Stephen I. Ryu, Leigh R. Hochberg, Jaimie M. Henderson, Krishna V. Shenoy, L. F. Abbott, and David Sussillo. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, September 2018. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-018-0109-9.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *arXiv:1711.04735 [cs, stat]*, November 2017. arXiv: 1711.04735.

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.

Daniel Ritchie, Paul Horsfall, and Noah D. Goodman. Deep Amortized Inference for Probabilistic Programs. *arXiv:1610.05735 [cs, stat]*, October 2016. arXiv: 1610.05735.

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.

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.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv:1312.6120 [cond-mat, q-bio, stat]*, December 2013. arXiv: 1312.6120.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep Information Propagation. *arXiv:1611.01232 [cs, stat]*, November 2016. 00003 arXiv: 1611.01232.

Christiane Tretter. *Spectral Theory of Block Operator Matrices and Applications*. IMPERIAL COLLEGE PRESS, October 2008. ISBN 978-1-86094-768-1 978-1-84816-112-2. doi: 10.1142/p493.

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.

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.

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. *arXiv:1703.01827 [cs]*, March 2017. arXiv: 1703.01827.

APPENDIX

4.1 SKETCH OF THE BOUND USING THE BLOCK DIAGONAL GERSHGORIN CIRCLE THEOREM

Using remark 1.13.2 from Tretter (2008), we consider a a block partitioned matrix $\mathcal{A} \in \mathbb{R}^{dn \times dn}$ with n $d \times d$ blocks $\mathbf{A}_{i,j} \quad i,j \in [1,\dots,n]$. Then consider diagonal blocks $\mathbf{A}_i i$ which we additionally require to hermitian. Then define sets for each i^{th} block-diagonal element the set of Gershgorin disks Γ_i

$$\Gamma_i \triangleq s(\mathbf{A}_{ii}) \cup \left\{ \bigcup_{k=1}^{d} C\left(\lambda_k(\mathbf{A}_{ii}), \sum_{j=1, j\neq i}^{n} \|\mathbf{A}_{i,j}\|\right) \right\} \tag{26}$$

where $s(\cdot)$ denotes the spectrum of a matrix and $C(c,r)$ denotes a ball centered on c with radius r

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

Therefore each Γ_i denotes a union of balls each centered on the eigenvalues of $\lambda(\mathbf{A}_{i,i})$ with radius given by the spectral norm of the off-diagonal blocks. The collection of Γ_i's contains the spectrum of $\mathcal{A}$. Let us assume for convenience that we are trying to predict continuous variables in a regression context and with a mean-square error loss. This simplifies the analysis in Section 3.1, by setting $\mathbf{H}_g = \mathbf{I}$. Then the block-diagonal elements of $\bar{\mathbf{G}}$ become $\mathbf{J}_{\theta^i}^{\mathbf{h}^L\top}\mathbf{J}_{\theta j}^{\mathbf{h}^L}$. The off diagonal elements have a spectral radius of $s_{max}(\mathbf{J}_{\theta^i}^{\mathbf{h}^L\top}\mathbf{J}_{\theta j}^{\mathbf{h}^L}) \leq s_{max}(\mathbf{J}_{\theta^i}^{\mathbf{h}^L})s_{max}(\mathbf{J}_{\theta j}^{\mathbf{h}^L})$. As a consequence the Gershgorin disks for i^{th} block diagonal matrix is a set sum of the eigenvalues of $\mathbf{J}_{\theta^i}^{h^L\top}\mathbf{J}_{\theta^i}^{h^L}$ and the disks centered on each of them, all with radius at most $s_{max}(\mathbf{J}_{\theta^i}^{h^L})s_{max}(\mathbf{J}_{\theta j}^{h^L})$. Since we care only about the maximum eigenvalue of $\bar{\mathbf{G}}$, we consider the maximum eigenvalue of each diagonal block and the disk around it.

$$\Gamma_i = C\left(s_{max}(\mathbf{J}_{\theta^i}^{\mathbf{h}^L}), \sum_{j=1, j\neq i}^{n} s_{max}(\mathbf{J}_{\theta^i}^{\mathbf{h}^L\top}\mathbf{J}_{\theta j}^{\mathbf{h}^L})\right) \tag{28}$$

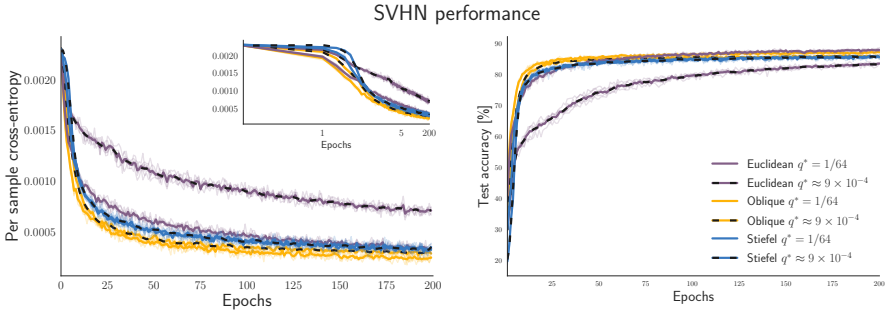
$$\subseteq C\left(s_{max}(\mathbf{J}_{\theta^i}^{\mathbf{h}^L}), \sum_{j=1, j\neq i}^{n} s_{max}(\mathbf{J}_{\theta^i}^{\mathbf{h}^L})s_{max}(\mathbf{J}_{\theta j}^{\mathbf{h}^L})\right) \tag{29}$$

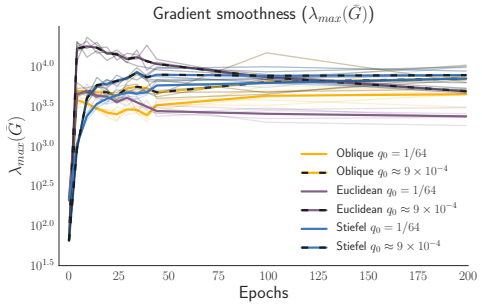
$$= \left\{\lambda : \left\|\lambda - s_{max}^2(\mathbf{J}_{\theta^i}^{\mathbf{h}^L})\right\| \leq \sum_{j=1, j\neq i}^{n} s_{max}(\mathbf{J}_{\theta^i}^{\mathbf{h}^L})s_{max}(\mathbf{J}_{\theta j}^{\mathbf{h}^L})\right\} \tag{30}$$

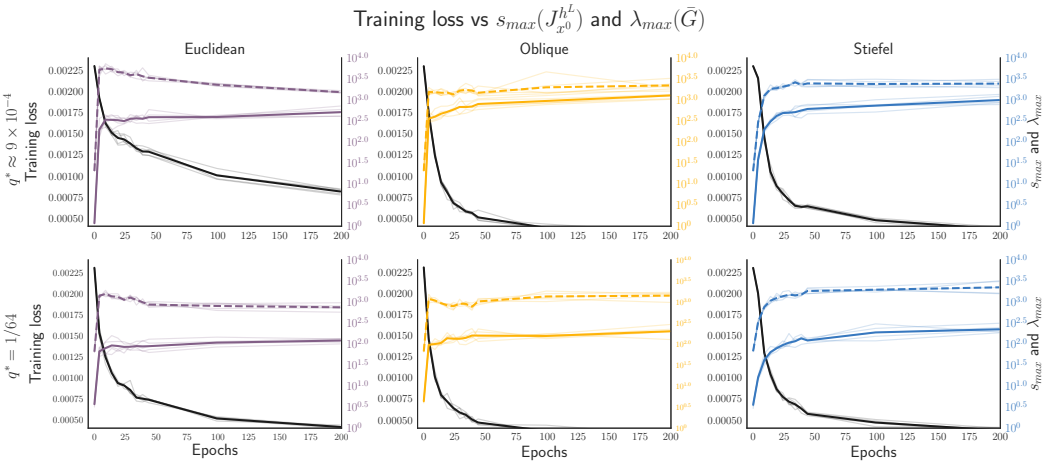
$$= \left\{\lambda : \left\|\lambda - s_{max}^2(\mathbf{J}_{\theta^i}^{\mathbf{h}^L})\right\| \leq s_{max}(\mathbf{J}_{\theta^i}^{\mathbf{h}^L}) \sum_{j=1, j\neq i}^{n} s_{max}(\mathbf{J}_{\theta j}^{\mathbf{h}^L})\right\} \tag{31}$$

So the relative size of the disk compare to $s_{max}(\mathbf{J}_{\theta^i}^{\mathbf{h}^L})$ becomes small whenever $s_{max}(\mathbf{J}_{\theta^i}^{\mathbf{h}^L}) > \sum_{j=1, j\neq i}^{n} s_{max}(\mathbf{J}_{\theta j}^{\mathbf{h}^L})$ and $s_{max}(\mathbf{J}_{\theta^i}^{\mathbf{h}^L}) \to \infty$ since the magnitude of $s_{max}^2(\mathbf{J}_{\theta^i}^{\mathbf{h}^L})$ grows quadratically.

4.2 SVHN FIGURES


SVHN performance
Per sample cross-entropy
Epochs
Test accuracy [%]
Epochs
Euclidean $q^* = 1/64$
Euclidean $q^* \approx 9 \times 10^{-4}$
Oblique $q^* = 1/64$
Oblique $q^* \approx 9 \times 10^{-4}$
Stiefel $q^* = 1/64$
Stiefel $q^* \approx 9 \times 10^{-4}$


Gradient smoothness ($\lambda_{max}(\bar{G})$)
$\lambda_{max}(\bar{G})$
Epochs
Oblique $q_0 = 1/64$
Oblique $q_0 \approx 9 \times 10^{-4}$
Euclidean $q_0 = 1/64$
Euclidean $q_0 \approx 9 \times 10^{-4}$
Stiefel $q_0 = 1/64$
Stiefel $q_0 \approx 9 \times 10^{-4}$


Training loss vs $s_{max}(J^{h^L}_{x^0})$ and $\lambda_{max}(\bar{G})$
Euclidean
Oblique
Stiefel
$q^* \approx 9 \times 10^{-4}$ Training loss
$q^* = 1/64$ Training loss
s_{max} and λ_{max}
Epochs