Deep Learning Is Singular, and That's Good

Susan Wei, Daniel Murfet, Mingming Gong, Hui Li, Jesse Gell-Redman, and Thomas Quella

***Abstract*— In singular models, the optimal set of parameters forms an analytic set with singularities, and a classical statistical inference cannot be applied to such models. This is significant for deep learning as neural networks are singular, and thus, "dividing" by the determinant of the Hessian or employing the Laplace approximation is not appropriate. Despite its potential for addressing fundamental issues in deep learning, a singular learning theory appears to have made little inroads into the developing canon of a deep learning theory. Via a mix of theory and experiment, we present an invitation to the singular learning theory as a vehicle for understanding deep learning and suggest an important future work to make the singular learning theory directly applicable to how deep learning is performed in practice.**

***Index Terms*— Bayesian deep learning, real log canonical threshold (RLCT), singular learning theory, widely applicable Bayes information criterion.**

I. Introduction

IT HAS been understood for close to 20 years that neural networks are singular statistical models [1], [2]. This means, in particular, that the set of network weights equivalent to the true model under the Kullback–Leibler (KL) divergence forms a real analytic variety, which fails to be an analytic manifold due to the presence of singularities. It has been shown by Sumio Watanabe that the geometry of these singularities controls the quantities of interest in statistical learning theory, e.g., the generalization error. A singular learning theory [3] is the study of singular models and requires very different tools from the study of regular statistical models. The breadth of knowledge demanded by the singular learning theory—Bayesian statistics, empirical processes, and algebraic geometry—is rewarded with profound and surprising results, which reveal that the singular models are different from regular models in practically important ways. To illustrate the relevance of the singular learning theory to deep learning, each section of this article illustrates a key takeaway idea.[1]

The real log canonical threshold (RLCT) is the correct way to count the effective number of parameters in a deep neural network (DNN) (Section IV). To every (model, truth, and prior) triplet is associated a birational invariant known as the RLCT. The RLCT can be understood in simple cases as half the number of normal directions to the set of true parameters. We will explain why this matters more than the curvature of those directions (as measured, for example, by eigenvalues of the Hessian) laying bare some of the confusion over "flat" minima.

For singular models, the Bayes predictive distribution is superior to maximum *a posteriori* (MAP) and maximum likelihood estimator (MLE) (Section V). In regular statistical models, the following hold: 1) Bayes predictive distribution; 2) MAP estimator; and 3) MLE have asymptotically equivalent generalization error (as measured by the KL divergence). This is not so in singular models. We illustrate, in our experiments, that even "being Bayesian" in just the final layers improves generalization over MAP. Our experiments further confirm that the Laplace approximation of the predictive distribution [4], [5] is not only theoretically inappropriate but performs poorly.

Simpler true distribution means lower RLCT (Section VI). In singular models, the RLCT depends on the (model, truth, and prior) triplet, whereas, in regular models, it depends only on the (model and prior) pair. The RLCT increases as the complexity of the true distribution relative to the supposed model increases. We verify this experimentally with a simple family of rectified linear units (ReLUs) and sigmoid linear units (SiLUs) networks.

II. Related Work

In a classical learning theory, generalization is explained by measures of capacity, such as the l_2 norm, Radamacher complexity, and Vapnik–Chervonenkis (VC) dimension [6]. It has become clear, however, that these measures cannot capture the empirical success of DNNs [7]. For instance, over-parameterized neural networks can easily fit random labels [7]–[9], indicating that the complexity measures, such as Rademacher complexity, are very large. There is also a slate of work on generalization bounds in deep learning. Uniform convergence bounds [10]–[13] usually cannot provide non-vacuous bounds. Data-dependent bounds [14]–[16] consider the "classifiability" of the data distribution in a generalization analysis of neural networks. Algorithm-dependent bounds [17]–[20] consider the relation of Gaussian initialization and the training dynamics of (stochastic) gradient descent to kernel methods [21].

In contrast to many of the aforementioned works, we are interested in estimating the conditional distribution $q(y|x)$. In particular, we measure the generalization error of some estimate $\hat{q}_n(y|x)$ in terms of the KL divergence between q and $\hat{q}_n$; see (V.1). Section III gives a crash course on singular learning theory. The rest of this article illustrates the key ideas listed in Section I. As we cover much ground in this short note, we will review other relevant work along the way, in particular,

Manuscript received 30 August 2021; revised 23 November 2021 and 4 February 2022; accepted 1 April 2022. The work of Susan Wei was supported by ARC under Grant DE200101253. The work of Mingming Gong was supported by ARC under Grant DE210101624. *(Corresponding author: Susan Wei.)*

The authors are with the School of Mathematics and Statistics, The University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: susan.wei@unimelb.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3167409.

Digital Object Identifier 10.1109/TNNLS.2022.3167409

[1] **Source code** is available at https://github.com/suswei/RLCT

literature on "flatness," the Laplace approximation in deep learning, and so on.

III. Singular Learning Theory

To understand why classical measures of capacity fail to say anything meaningful about DNNs, it is important to distinguish between two different types of statistical models. Recall we are interested in estimating the true (and unknown) conditional distribution $q(y|x)$ with a class of models $\{p(y|x,w) : w \in W\}$, where $W \subset \mathbb{R}^d$ is the parameter space. We say the model is identifiable if the mapping $w \mapsto p(y|x,w)$ is one-to-one. Let $q(x)$ be the distribution of x. The Fisher information matrix associated with the model $\{p(y|x,w) : w \in W\}$ is the matrix-valued function $I(w)$ on W whose entry $I(w)_{ij}$ is defined by

$$\iint \frac{\partial}{\partial w_i}\big[\log p(y|x,w)\big]\frac{\partial}{\partial w_j}\big[\log p(y|x,w)\big]q(y|x)q(x)dxdy$$

if this integral is finite. Following the conventions in [3], we have the following bifurcation of statistical models. A statistical model $p(y|x,w)$ is called regular if it is 1) identifiable and 2) has positive-definite Fisher information matrix. A statistical model is called strictly singular if it is not regular.

Let $\varphi(w)$ be a prior on the model parameters w. To every (model, truth, and prior) triplet, we can associate the zeta function, $\zeta(z) = \int K(w)^z \varphi(w)\, dw, z \in \mathbb{C}$, where $K(w)$ is the KL divergence between the model $p(y|x,w)$ and the true distribution $q(y|x)$

$$K(w) := \iint q(y|x) \log \frac{q(y|x)}{p(y|x,w)} q(x)\, dx\, dy. \quad \text{(III.1)}$$

For a (model, truth, and prior) triplet $(p(y|x,w), q(y|x),$ and $\varphi)$, let $-\lambda$ be the maximum pole of the corresponding zeta function. We call λ the RLCT [3] of the (model, truth, and prior) triplet. The RLCT is the central quantity of the singular learning theory. Many existing papers in singular learning theory attempt to calculate or estimate the RLCT for the purpose of eventually estimating the minus log maringal likelihood, or free energy. We will soon see that the RLCT is interesting in its own right as it is the correct way to count the effective number of parameters in a DNN (Section IV). The RLCT will also show up in important equations for the generalization error.

By [3, Th. 6.4], the RLCT is equal to $d/2$ in regular statistical models and bounded above by $d/2$ in strictly singular models if realizability holds: let

$$W_0 = \{w \in W : p(y|x,w) = q(y|x)\}$$

be the set of true parameters, and we say $q(y|x)$ is realizable by the model class if W_0 is non-empty. The condition of realizability is critical to standard results in singular learning theory. Modifications to the theory are needed in the case that $q(y|x)$ is not realizable; see the condition called relatively finite variance in [22].

A. Neural Networks in Singular Learning Theory

Let $W \subseteq \mathbb{R}^d$ be the space of weights of a neural network of some fixed architecture, and let $f(x,w) : \mathbb{R}^N \times W \longrightarrow \mathbb{R}^M$ be the associated function. We shall focus on the regression task and study the model

$$p(y|x,w) = \frac{1}{(2\pi)^{M/2}} \exp\left(-\frac{1}{2}\|y - f(x,w)\|^2\right) \quad \text{(III.2)}$$

but singular learning theory can also apply to classification, for instance. It is routine to check (see Appendix A) that, for feedforward ReLU networks, not only is the model strictly singular but the matrix $I(w)$ is degenerated for all nontrivial weight vectors and the Hessian of $K(w)$ is degenerated at every point of W_0.

B. RLCT Plays an Important Role in Model Selection

One of the most accessible results in singular learning theory is the work related to the widely applicable Bayesian information criterion (WBIC) [23], which we briefly review here for completeness. Let $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$ be a dataset of input–output pairs. Let $L_n(w)$ be the negative log likelihood

$$L_n(w) = -\frac{1}{n}\sum_{i=1}^n \log p(y_i|x_i,w) \quad \text{(III.3)}$$

and $p(\mathcal{D}_n|w) = \exp(-nL_n(w))$. The marginal likelihood of a model $\{p(y|x,w) : w \in W\}$ is given by $p(\mathcal{D}_n) = \int_W p(\mathcal{D}_n|w)\varphi(w)\, dw$ and can be loosely interpreted as the evidence for the model. Between two models, we should prefer the one with higher model evidence. However, as the marginal likelihood is an intractable integral over the parameter space of the model, one needs to consider some approximation.

The well-known Bayesian information criterion (BIC) derives from an asymptotic approximation of $-\log p(\mathcal{D}_n)$ using the Laplace approximation, leading to BIC $= nL_n(w_{\text{MLE}}) + (d/2)\log n$. As we want the marginal likelihood of the data for some given model to be high, one should almost never adopt a DNN according to the BIC, because in such models, d may be very large. However, this argument contains a serious mathematical error: the Laplace approximation used to derive BIC only applies to regular statistical models, and DNNs are not regular. The correct criterion for both regular and strictly singular models was shown in [23] to be $nL_n(w_0) + \lambda \log n$, where $w_0 \in W_0$, and λ is the RLCT. As DNNs are highly singular and λ may be much smaller than $d/2$ (Section VI), it is possible for DNNs to have high marginal likelihood—consistent with their empirical success.

IV. Volume Dimension, Effective Degrees of Freedom, and Flatness

A. Volume Codimension

The easiest way to understand the RLCT is as a volume codimension [3, Th. 7.1]. Suppose that $W \subseteq \mathbb{R}^d$ and W_0 is nonempty; i.e., the true distribution is realizable. We consider a special case in which the KL divergence in a neighborhood of every point $v_0 \in W_0$ has an expression in local coordinates of the form

$$K(w) = \sum_{i=1}^{d'} c_i w_i^2 \quad \text{(IV.1)}$$

where the coefficients $c_1, \ldots, c_{d'} > 0$ may depend on v_0, and d' may be strictly less than d. If the model is regular, then this is true with $d = d'$, and if it holds for $d' < d$, then we say that the pair $(p(y|x, w), q(y|x))$ is minimally singular. It follows that the set $W_0 \subseteq W$ of true parameters is a regular submanifold of codimension d' (that is, W_0 is a manifold of dimension $d - d'$, where W has dimension d). Under this hypothesis, there are, near each true parameter $v_0 \in W_0$, exactly $d - d'$ directions in which v_0 can be varied without changing the model $p(y|x, w)$ and d' directions in which varying the parameters does change the model. In this sense, there are d' effective parameters near v_0.

This number of effective parameters can be computed by an integral. Consider the volume of the set of almost true parameters $V(t, v_0) = \int_{K(w)<t} \varphi(w) dw$, where the integral is restricted to a small closed ball around v_0. As long as the prior $\varphi(w)$ is non-zero on W_0, it does not affect the relevant features of the volume, so we may assume φ is constant on the region of integration in the first d' directions and normal in the remaining directions, so up to a constant depending only on d', we have

$$V(t, v_0) \propto \frac{t^{d'/2}}{\sqrt{c_1, \ldots, c_{d'}}} \tag{IV.2}$$

and we can extract the exponent of t in this volume in the limit

$$d' = 2 \lim_{t \to 0} \frac{\log\{V(at, v_0)/V(t, v_0)\}}{\log(a)} \tag{IV.3}$$

for any $a > 0$, $a \neq 1$. We refer to the right-hand side of (IV.3) as the volume codimension at v_0.

The function $K(w)$ has the special form (IV.1) locally with $d' = d$ if the statistical model is regular (and realizable) and with $d' < d$ in some singular models, such as reduced rank regression (Appendix B). While such a local form does not exist for a singular model generally (in particular, for neural networks), nonetheless, under natural conditions [3, Th. 7.1], we have $V(t, v_0) = ct^{\lambda} + o(t^{\lambda})$, where c is a constant. We assume that, in a sufficiently small neighborhood of v_0, the point RLCT λ at v_0 [3, Definition 2.7] is less than or equal to the RLCT at every point in the neighborhood, so that the multiplicity $m = 1$; see [3, Sec. 7.6] for relevant discussion. It follows that the limit on the right-hand side of (IV.3) exists and is equal to λ. In particular, $\lambda = d'/2$ in the minimally singular case.

Note that for strictly singular models, such as DNNs, 2λ may not be an integer. This may be disconcerting but the connection among the RLCT, generalization error, and volume dimension strongly suggests that 2λ is, nonetheless, the only geometrically meaningful "count" of the effective number of parameters near v_0.

B. RLCT and Likelihood Versus Temperature

Again working with the model in (III.2), consider the expectation over the posterior at temperature T as defined in (C.4) of the negative log likelihood (III.3)

$$\begin{aligned} E(T) &= \mathbb{E}_w^{1/T}[nL_n(w)] \\ &= \mathbb{E}_w^{1/T}\left[\frac{1}{2}\sum_{i=1}^{n}\|y_i - f(x_i, w)\|^2\right] + \frac{nM}{2}\log(2\pi). \end{aligned}$$

Note that when n is large $L_n(v_0) \approx (M/2)\log(2\pi)$ for any $v_0 \in W_0$, so for $T \approx 0$, the posterior concentrates around the set W_0 of true parameters and $E(T) \approx (nM/2)\log(2\pi)$. Consider the increase $\Delta E = E(T + \Delta T) - E(T)$ corresponding to an increase in temperature ΔT. It can be shown that $\Delta E \approx \lambda \Delta T$ where the reader should see [23, Corollary 3] for a precise statement. As the temperature increases, samples taken from the tempered posterior are more distant from W_0, and the error E will increase. If λ is smaller than, for a given increase in temperature, the quantity, then E increases less: this is one way to understand intuitively why a model with smaller RLCT generalizes better from the dataset D_n to the true distribution.

C. Flatness

It is folklore in the deep learning community that flatness of minima is related to generalization [24], [25], and this claim has been revisited in recent years [4], [5], [26], [27]. In regular models, this can be justified using the lower order terms of the asymptotic expansion of the Bayes free energy [28, Sec. 3.1], but the argument breaks down in strictly singular models, because, for example, the Laplace approximation of [5] is invalid. The point can be understood via an analysis of the version of the idea in [25]. Their measure of entropy compares the volume of the set of parameters with tolerable error t_0 (our almost true parameters) to a standard volume

$$-\log\left[\frac{V(t_0, v_0)}{t_0^{d/2}}\right] = \frac{d - d'}{2}\log(t_0) + \frac{1}{2}\sum_{i=1}^{d}\log c_i. \tag{IV.4}$$

Hence, in the case $d = d'$, the quantity $-(1/2)\sum_i \log(c_i)$ is a measure of the entropy of the set of true parameters near w_0, a point made, for example, in [5]. However, when $d' < d$, this conception of entropy is inappropriate because of the $d - d'$ directions in which $K(w)$ is flat near v_0, which introduce the t_0 dependence in (IV.4).

V. Generalization

The generalization puzzle [29] is one of the central mysteries of deep learning. A theoretical investigation into the matter is an active area of research [30]. Many of the recent proposals of capacity measures for neural networks are based on the eigenspectrum of the (degenerate) Hessian [31], [32]. But, this is not appropriate for singular models and, hence, for DNNs.

As we are interested in learning the distribution, our notion of generalization is slightly different, being measured by the KL divergence. Precise statements regarding the generalization behavior in singular models can be made using the singular learning theory. Let the network weights be denoted θ rather than w for reasons that will become clear. Recalling in the Bayesian paradigm, prediction proceeds via

the so-called Bayes predictive distribution, $p(y|x, \mathcal{D}_n) = \int p(y|x, \theta) p(\theta|\mathcal{D}_n)\, d\theta$. More commonly encountered in deep learning practice are the MAP and MLE point estimators. While in a regular statistical model, the three estimators: 1) Bayes predictive distribution; 2) MAP; and 3) MLE have the same leading term in their asymptotic generalization behavior, and the same is not true in singular models. More precisely, let $\hat{q}_n(y|x)$ be some estimate of the true unknown conditional density $q(y|x)$ based on the dataset $\mathcal{D}_n$. The generalization error of the predictor $\hat{q}_n(y|x)$ is

$$\begin{aligned} G(n) &:= KL(q(y|x)||\hat{q}_n(y|x)) \\ &= \iint q(y|x) \log \frac{q(y|x)}{\hat{q}_n(y|x)} q(x)\, dy\, dx. \end{aligned} \tag{V.1}$$

To account for sampling variability, we will work with the average generalization error, $\mathbb{E}_n G(n)$, where $\mathbb{E}_n$ denotes expectation over the dataset $\mathcal{D}_n$. By [3, Ths. 1.2 and 7.2], if $\hat{q}_n$ is the Bayes predictive distribution, we have

$$\mathbb{E}_n G(n) = \lambda/n + o(1/n) \tag{V.2}$$

where λ is the RLCT corresponding to the triplet $(p(y|x, \theta), q(y|x)$, and $\varphi(\theta))$. In contrast, we should note that [4] and [5] rely on the Laplace approximation to explain the generalization of the Bayes predictive distribution though both works acknowledge that the Laplace approximate is inappropriate. For completeness, a quick sketch of the derivation of (V.2) is provided in Appendix D. Now, by [3, Th. 6.4], if $\hat{q}_n$ is the Bayes predictive distribution, we have

$$\mathbb{E}_n G(n) = C/n + o(1/n) \tag{V.3}$$

where C (different for MAP and MLE) is the maximum of some Gaussian process. For regular models, the MAP, MLE, and the Bayes predictive distribution have the same leading term for $\mathbb{E}_n G(n)$, because $\lambda = C = d/2$. However, in singular models,, we have the following: 1) $C \gg d/2$; see the discussion in [3, Sec. 1.4.4] for details, and 2) the RLCT $\lambda \ll d/2$, as can be seen by the many examples in which the true RLCT is known [33], [34]. This means we should prefer the Bayes predictive distribution for singular models.

The RLCT that has such a simple relationship to the Bayesian generalization error is remarkable. On the other hand, the practical implications of (V.2) are limited, because the Bayes predictive distribution is intractable.[2] While approximations to the Bayesian predictive distribution, say via variational inference, might inherit a similar relationship between the generalization and the (variational) RLCT, serious theoretical developments will be required to rigorously establish this. The challenge comes from the fact that for approximate Bayesian predictive distributions, the free energy and generalization error may have different learning coefficients λ. This was well documented in the case of a neural network with one hidden layer [35].

We set out to investigate whether certain very simple approximations of the Bayes predictive distribution can already demonstrate superiority over point estimators. Suppose the input-target relationship is modeled as in (III.2), but we write θ instead of w. We set $q(x) = N(0, I_3)$. For now, consider the realizable case, $q(y|x) = p(y|x, \theta_0)$, where θ_0 is drawn randomly according to the default initialization in PyTorch when model (III.2) is instantiated. We calculate $\mathbb{E}_n G(n)$ using multiple datasets $\mathcal{D}_n$ and a large testing set; see Appendix E for more details.

As f is a hierarchical model, let us write it as $f_\theta(\cdot) = h(g(\cdot; v); w)$ with the dimension of w being relatively small. Let $\theta_{\text{MAP}} = (v_{\text{MAP}}, w_{\text{MAP}})$ be the MAP estimate for θ using batch gradient descent. The idea of our simple approximate Bayesian scheme is to freeze the network weights at the MAP estimate for early layers and perform approximate Bayesian inference for the final layers.[3] For example, freeze the parameters of g at v_{MAP} and perform Markov Chain Monte Carlo (MCMC) over w. Throughout the experiments, $g : \mathbb{R}^3 \to \mathbb{R}^3$ is a feedforward ReLU block with each hidden layer having five hidden units, and $h : \mathbb{R}^3 \to \mathbb{R}^3$ is either BAx or $B\,\text{ReLU}(Ax)$, where $A \in \mathbb{R}^{3\times r}$ and $B \in \mathbb{R}^{r\times 3}$. We set $r = 3$. We shall consider one or five hidden layers for g.

To approximate the Bayes predictive distribution, we perform either the Laplace approximation or the No-U-Turn Sampler (NUTS) variant of Hamiltonian Monte Carlo (HMC) [37] in the last two layers, i.e., performing inference over A, B in $h(g(\cdot; v_{\text{MAP}}); A, B)$. Note that MCMC is operating in a space of 18 dimensions in this case, which is small enough for us to expect MCMC to perform well. We also implemented the Laplace approximation and NUTS in the last layer only, i.e., performing inference over B in $h_2(h_1(g(\cdot; v_{\text{MAP}}); A_{\text{MAP}}); B)$. Further implementation details of these approximate Bayesian schemes are found in Appendix E.

From the outset, we expect the Laplace approximation over $w = (A, B)$ to be invalid, because the model is singular. We do, however, expect the last-layer-only Laplace approximation over B to be sound. Next, we expect the MCMC approximation in either the last layer or last two layers to be superior to the Laplace approximations and to the MAP. We further expect the last-two-layers MCMC to have better generalization than the last-layer-only MCMC, because the former is closer to the Bayes predictive distribution. In summary, we anticipate the following performance order for these five approximate Bayesian schemes (from worst to best): last-two-layers Laplace, last-layer-only Laplace, MAP, last-layer-only MCMC, and last-two-layers MCMC.

Fig. 1 displays the average generalization error $\mathbb{E}_n G(n)$ for various approximations of the Bayes predictive distribution. The estimation of $\mathbb{E}_n G(n)$ is discussed in Appendix E, and the error bars are due to approximating $\mathbb{E}_n$ over 30 draws of training sets of size n. The results of the Laplace approximations are reported in the Appendix and not displayed in Fig. 1, because they are higher than other approximation schemes by at least an order of magnitude. Each subplot in Fig. 1 shows a different combination of hidden layers in g (one or

[2]To the best of our knowledge, experimental verification of V.2 has only been conducted for simple reduced rank regression models; see [3, Sec. 8.3.1].

[3]This is similar in spirit to [36] who claim that even "being Bayesian a little bit" fixes overconfidence. They approach this via the Laplace approximation for the final layer of an ReLU network. It is also worth noting that [36] does not attempt to formalize what it means to "fix overconfidence"; the precise statement should be in terms of $G(n)$.

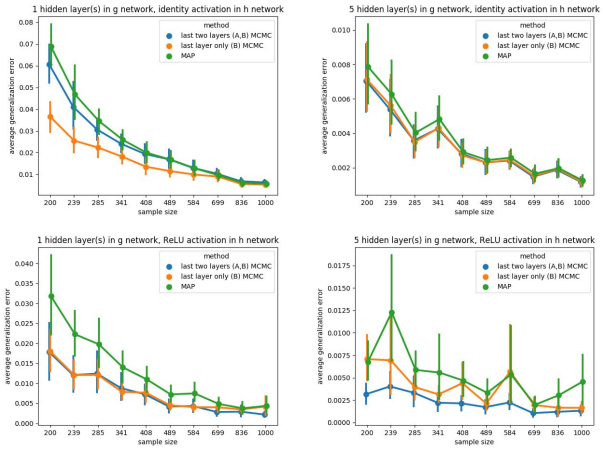


Fig. 1. Realizable and full-batch gradient descent for MAP. Average generalization errors $\mathbb{E}_n\, G(n)$ are displayed for various approximations of the Bayes predictive distribution. The results of the Laplace approximations are reported in the Appendix and not displayed here, because they are higher than other approximation schemes by at least an order of magnitude. Each subplot shows a different combination of hidden layers in g (one or five) and activation function in h (ReLU or identity). Note that the y-axis is not shared between the subfigures.

five) and activation function in h (ReLU or identity). Note that the y-axis is not shared. The results are in line with our stated expectations mentioned earlier, except for the surprise that the last-layer-only MCMC approximation is often superior to the last-two-layers MCMC approximation. This may arise from the fact that the MCMC finds the singular setting in the last-two-layers more challenging. It is also worth noting that the theory says the full Bayesian predictive distribution is superior to MAP and MLE for singular models. Thus, when we perform Bayesian inference in the last layers only, we are not calculating the full Bayesian predictive distribution. This is, perhaps, why we see a less pronounced difference between MAP and the last layer Bayesian schemes for the larger neural network architecture, i.e., when there are five hidden layers in g.

Table I is a companion to Fig. 1 and tabulates, for each approximation scheme, the slope of $1/n$ versus $\mathbb{E}_n G(n)$, also known as the learning coefficient. R^2 corresponding to the linear fit is also provided. In Appendix E, we also show the corresponding results when the following hold: 1) the data-generating mechanism and the assumed model do not satisfy the condition of realizability, and/or 2) the MAP estimate is obtained via mini-batch stochastic gradient descent (SGD) instead of full-batch gradient descent. Specifically, these additional results can be found in Figs. 3–5 in Appendix E, and their companions Tables III–V. Readers will notice that we have not displayed the generalization error for the last-layer(s) Laplace methods in Figs. 1 and 3–5. This is because the generalization error for last-layer(s) Laplace is much higher than last-layer(s) MCMC and MAP. The generalization error plots for last-layer(s) Laplace can be found in Figs. 6–9 in Appendix E. The conclusion is not sensitive to the realizability condition. However, the batch size for MAP training does appear to have some effect on the results. Contrasting Figs. 1 and 3, we can see a more pronounced different between last layer(s) Bayesian inference and MAP when MAP is trained via mini-batch SGD.

TABLE I

COMPANION TO FIG. 1. THE LEARNING COEFFICIENT IS THE SLOPE OF THE LINEAR FIT $1/n$ VERSUS $\mathbb{E}_n\, G(n)$ (NO INTERCEPT SINCE REALIZABLE). THE R^2 VALUE GIVES A SENSE OF THE GOODNESS-OF-FIT. (a) ONE HIDDEN LAYER(S) IN g, IDENTITY ACTIVATION IN h. (b) FIVE HIDDEN LAYER(S) IN g, IDENTITY ACTIVATION IN h. (c) ONE HIDDEN LAYER(S) IN g, RELU ACTIVATION IN h. (d) FIVE HIDDEN LAYER(S) IN g, RELU ACTIVATION IN h

(a)

method	learning coefficient	R squared
last two layers (A,B) MCMC	9.709027	0.966124
last layer only (B) MCMC	6.410380	0.988921
last two layers (A,B) Laplace	inf	NaN
last layer only (B) Laplace	2154.989266	0.801077
MAP	10.714216	0.951051

(b)

method	learning coefficient	R squared
last two layers (A,B) MCMC	1.286290	0.985161
last layer only (B) MCMC	1.298504	0.982298
last two layers (A,B) Laplace	inf	NaN
last layer only (B) Laplace	2038.605589	0.803736
MAP	1.437473	0.983411

(c)

method	learning coefficient	R squared
last two layers (A,B) MCMC	3.117187	0.977313
last layer only (B) MCMC	3.152710	0.980132
last two layers (A,B) Laplace	inf	NaN
last layer only (B) Laplace	1120.648298	0.742412
MAP	5.343311	0.972212

(d)

method	learning coefficient	R squared
last two layers (A,B) MCMC	0.835593	0.957824
last layer only (B) MCMC	1.466273	0.920716
last two layers (A,B) Laplace	inf	NaN
last layer only (B) Laplace	1416.294288	0.808991
MAP	1.981483	0.889519

VI. SIMPLE FUNCTIONS AND COMPLEX SINGULARITIES

In singular models, the RLCT may vary with the true distribution (in contrast to regular models), and in this section, we examine this phenomenon in a simple example. As the true distribution becomes more complicated relative to the supposed model, the singularities of the analytic variety of true parameters should become simpler, and hence, the RLCT should increase [3, Sec. 7.6]. Our experiments are inspired by [3, Sec. 7.2], where $\tanh(x)$ networks are considered, and the true distribution (associated with the zero network) is held fixed while the number of hidden nodes is increased.

Consider the model $p(y|x,w)$ in (III.2), where $f(x,w) = c + \sum_{i=1}^{H} q_i\, \mathrm{ReLU}(\langle w_i, x\rangle + b_i)$ is a two-layer ReLU network with weight vector $w = (\{w_i\}_{i=1}^H, \{b_i\}_{i=1}^H, \{q_i\}_{i=1}^H, c) \in \mathbb{R}^{4H+1}$ and $w_i \in \mathbb{R}^2$, $b_i \in \mathbb{R}$, and $q_i \in \mathbb{R}$ for $1 \le i \le H$.

Let $W \subset \mathbb{R}^{4H+1}$ be some compact neighborhood of the origin. Given an integer $3 \le m \le H$, we define a network $s_m \in W$ and $q_m(y|x) := p(y|x, s_m)$ as follows. Let $g \in SO(2)$ stand for rotation by $2\pi/m$, and set $w_1 = \sqrt{g}\,(1,0)^T$. The components of s_m are the vectors $w_i = g^{i-1} w_1$ for $1 \le i \le m$ and $w_i = 0$ for $i > m$, $b_i = -(1/3)$ and $q_i = 1$ for $1 \le i \le m$ and $b_i = q_i = 0$ for $i > m$, and, finally, $c = 0$. The factor of $(1/3)$ ensures that the relevant parts of the decision

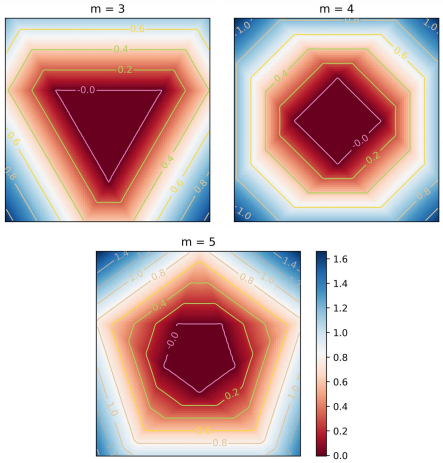


Fig. 2. Function $f(x, s_m)$ is plotted for various values of m. As m increases in $f(x, s_m)$, we obtain increasingly complicated true distributions $q_m(x, y)$ on $[-1, 1]^2 \times \mathbb{R}$.

TABLE II

RLCT ESTIMATES FOR RELU AND SILU NETWORKS. WE OBSERVE THE RLCT INCREASING AS m INCREASES; I.E., THE TRUE DISTRIBUTION BECOMES MORE "COMPLICATED" RELATIVE TO THE SUPPOSED MODEL

m	Nonlinearity	RLCT	Std	R squared
3	ReLU	0.7526301	0.027181	0.983850
3	SiLU	0.7522393	0.026342	0.978770
4	ReLU	0.7539590	0.024774	0.991241
4	SiLU	0.7539387	0.020769	0.988495
5	ReLU	0.755303	0.002344	0.993092
5	SiLU	0.755630	0.021184	0.990971

boundaries lie within $X = [-1, 1]^2$. We let $q(x)$ be the uniform distribution on X and define $q_m(x, y) = q_m(y|x)q(x)$. The functions $f(x, s_m)$ are graphed in Fig. 2. It is intuitively clear that the complexity of these true distributions increases with m.

We let φ be a normal distribution $N(0, 50^2)$ and estimate the RLCTs of the triples (p, q_m, and φ). We conducted the experiments with $H = 5$ and $n = 1000$. For each $m \in \{3, 4, 5\}$, Table II shows the estimated RLCT. We applied Algorithm 1 with five values of β's and $|\mathcal{T}| = 3$ to estimate the RLCT. We discuss the provenance of Algorithm 1, particularly that it is a direct consequence of [23, Th. 4] and a simple improvement over Watanabe's RLCT estimation procedure in [23, Sec. 6.2]. As expected, the RLCT increases with m verifying that in this case, the simpler true distributions give rise to more complex singularities.

Note that the dimension of W is $d = 21$, and so if the model was regular, the RLCT would be 10.75. It can be shown that when $m = H$, the set of true parameters $W_0 \subseteq W$ is a regular submanifold of dimension m. If such a model was minimally singular, its RLCT would be $(1/2)((4m+1)-m) = (1/2)(3m+1)$. In the case $m = 5$, we observe an RLCT more than an order of magnitude less than the value of 8 predicted by this formula. So, the function K does not behave like a quadratic form near W_0.

Strictly speaking, it is incorrect to speak of the RLCT of an ReLU network, because the function $K(w)$ is not necessarily analytic (Example 1). However, we observe empirically that

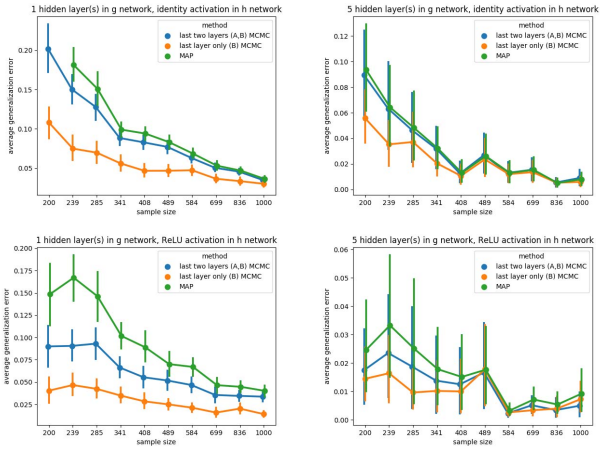


Fig. 3. Realizable and mini-batch gradient descent for MAP training.

TABLE III

COMPANION TO FIG. 3. (a) ONE HIDDEN LAYER(S) IN g, IDENTITY ACTIVATION IN h. (b) FIVE HIDDEN LAYER(S) IN g, IDENTITY ACTIVATION IN h. (c) ONE HIDDEN LAYER(S) IN g, RELU ACTIVATION IN h. (d) FIVE HIDDEN LAYER(S) IN g, RELU ACTIVATION IN h

(a)

method	learning coefficient	R squared
last two layers (A,B) MCMC	36.721594	0.992839
last layer only (B) MCMC	20.676920	0.983695
last two layers (A,B) Laplace	inf	NaN
last layer only (B) Laplace	1768.655088	0.838035
MAP	inf	NaN

(b)

method	learning coefficient	R squared
last two layers (A,B) MCMC	13.729278	0.924049
last layer only (B) MCMC	9.170642	0.945613
last two layers (A,B) Laplace	inf	NaN
last layer only (B) Laplace	1943.793236	0.794679
MAP	14.123308	0.917502

(c)

method	learning coefficient	R squared
last two layers (A,B) MCMC	22.175448	0.975450
last layer only (B) MCMC	10.675455	0.968584
last two layers (A,B) Laplace	inf	NaN
last layer only (B) Laplace	inf	NaN
MAP	35.647464	0.983284

(d)

method	learning coefficient	R squared
last two layers (A,B) MCMC	4.652483	0.922693
last layer only (B) MCMC	3.533366	0.862125
last two layers (A,B) Laplace	inf	NaN
last layer only (B) Laplace	1004.852367	0.901899
MAP	6.256696	0.940437

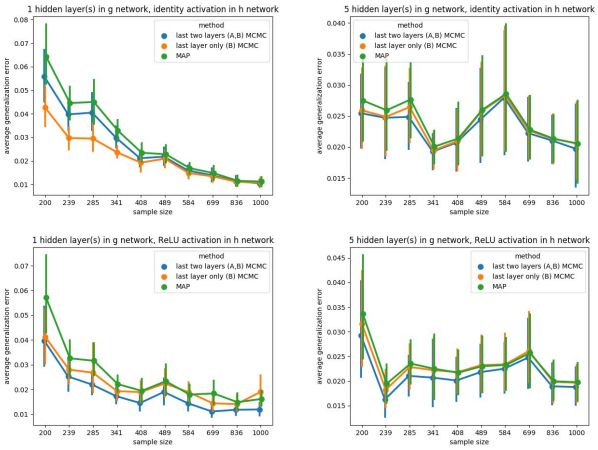


Fig. 4. Nonrealizable and full-batch gradient descent for MAP training.

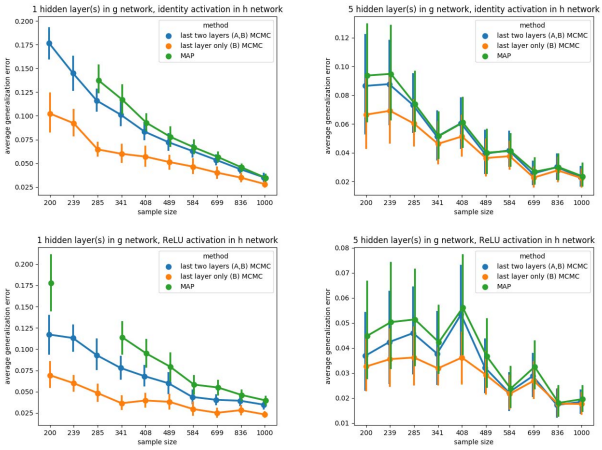


Fig. 5. Nonrealizable and mini-batch gradient descent for MAP training. Missing points on the MAP learning curve are due to estimated probabilities too close to 0.

TABLE IV

COMPANION TO FIG. 4. (a) ONE HIDDEN LAYER(S) IN g, IDENTITY ACTIVATION IN h. (b) FIVE HIDDEN LAYER(S) IN g, IDENTITY ACTIVATION IN h. (c) ONE HIDDEN LAYER(S) IN g, RELU ACTIVATION IN h. (d) FIVE HIDDEN LAYER(S) IN g, RELU ACTIVATION IN h

(a)

method	learning coefficient	R squared
last two layers (A,B) MCMC	11.086023	0.969991
last layer only (B) MCMC	7.377871	0.957824
last two layers (A,B) Laplace	NaN	NaN
last layer only (B) Laplace	30.692954	0.029238
MAP	12.947959	0.970173

(b)

method	learning coefficient	R squared
last two layers (A,B) MCMC	0.808601	0.144260
last layer only (B) MCMC	0.799114	0.127686
last two layers (A,B) Laplace	NaN	NaN
last layer only (B) Laplace	-33.817429	0.009074
MAP	1.204743	0.242671

(c)

method	learning coefficient	R squared
last two layers (A,B) MCMC	5.987187	0.848490
last layer only (B) MCMC	5.384686	0.801313
last two layers (A,B) Laplace	NaN	NaN
last layer only (B) Laplace	38.629167	0.059012
MAP	8.560722	0.816794

(d)

method	learning coefficient	R squared
last two layers (A,B) MCMC	0.794055	0.088305
last layer only (B) MCMC	1.141580	0.162585
last two layers (A,B) Laplace	NaN	NaN
last layer only (B) Laplace	-5.682602	0.000365
MAP	1.648073	0.284088

TABLE V

COMPANION TO FIG. 5. THE LEARNING COEFFICIENT IS THE SLOPE OF THE LINEAR FIT $1/n$ VERSUS $\mathbb{E}_n G(n)$ (WITH INTERCEPT SINCE NONREALIZABLE). (a) ONE HIDDEN LAYER(S) IN g, IDENTITY ACTIVATION IN h. (b) FIVE HIDDEN LAYER(S) IN g, IDENTITY ACTIVATION IN h. (c) ONE HIDDEN LAYER(S) IN g, RELU ACTIVATION IN h. (d) FIVE HIDDEN LAYER(S) IN g, RELU ACTIVATION IN h

(a)

method	learning coefficient	R squared
last two layers (A,B) MCMC	34.062199	0.996257
last layer only (B) MCMC	17.437010	0.964007
last two layers (A,B) Laplace	NaN	NaN
last layer only (B) Laplace	230.961469	0.632128
MAP	inf	NaN

(b)

method	learning coefficient	R squared
last two layers (A,B) MCMC	17.381572	0.926485
last layer only (B) MCMC	12.389040	0.903937
last two layers (A,B) Laplace	NaN	NaN
last layer only (B) Laplace	275.384501	0.631448
MAP	19.094974	0.938559

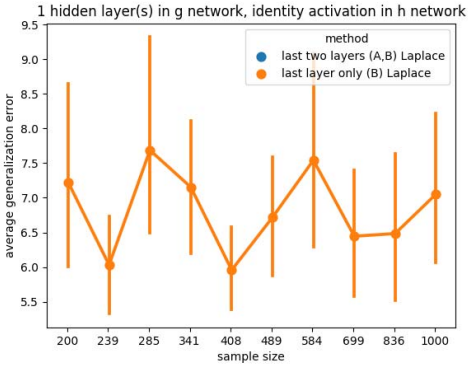
(c)

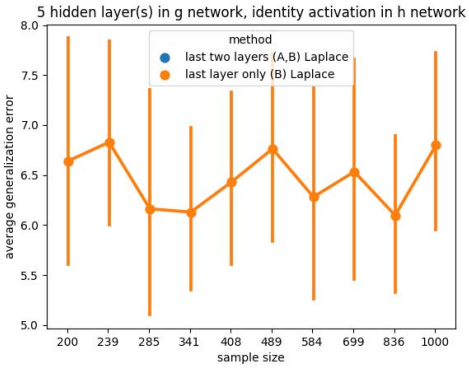
method	learning coefficient	R squared
last two layers (A,B) MCMC	22.715621	0.980508
last layer only (B) MCMC	11.082223	0.949126
last two layers (A,B) Laplace	NaN	NaN
last layer only (B) Laplace	NaN	NaN
MAP	inf	NaN

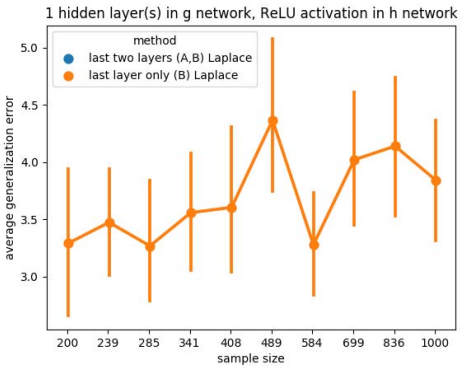
(d)

method	learning coefficient	R squared
last two layers (A,B) MCMC	5.875258	0.428733
last layer only (B) MCMC	4.202670	0.595327
last two layers (A,B) Laplace	NaN	NaN
last layer only (B) Laplace	-inf	NaN
MAP	7.657506	0.558303

the predicted linear relationship between $E_w^\beta[nL_n(w)]$ and $1/\beta$ holds in our small ReLU networks (see the R^2 values in Table II), and that the RLCT estimates are close to those for the two-layer SiLU network [38], which is analytic (the SiLU or sigmoid weighted linear unit is $\sigma(x) = x(1 + e^{-\tau x})^{-1}$, which approaches the ReLU as $\tau \to \infty$; we use $\tau = 100.0$ in our experiments). The competitive performance of SiLU on standard benchmarks [39] shows that the non-analyticity of ReLU is probably not fundamental.







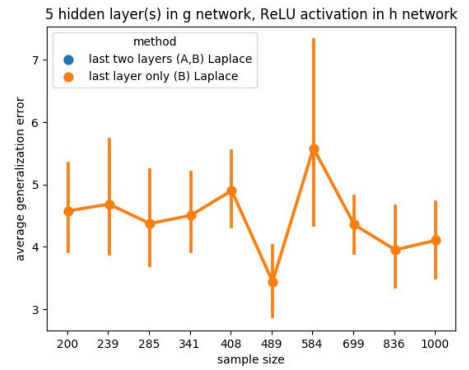
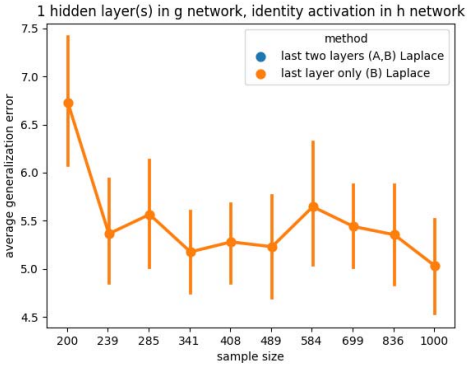
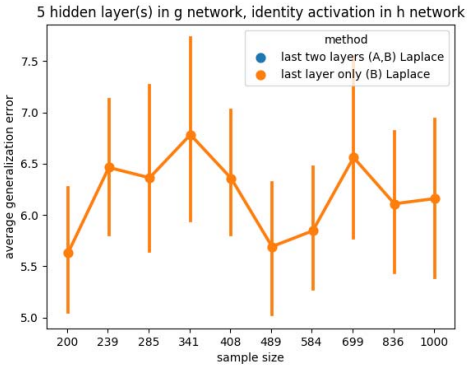
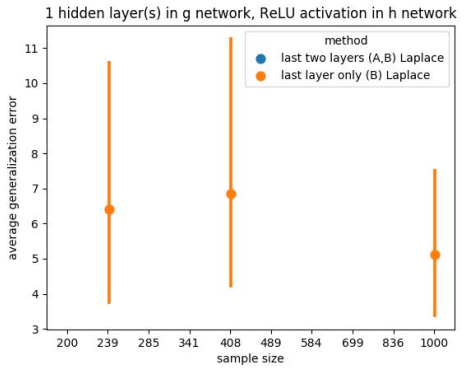


Fig. 6. Realizable and full-batch gradient descent for MAP. Average generalization errors of Laplace approximations of the predictive distribution. The last-two-layers Laplace approximation results in numerical instabilities due to degenerate Hessian. Any missing points are due to estimated probabilities too close to 0.







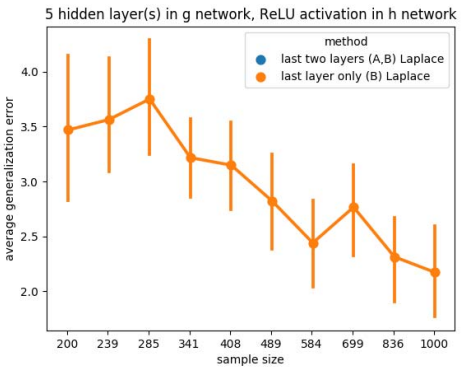
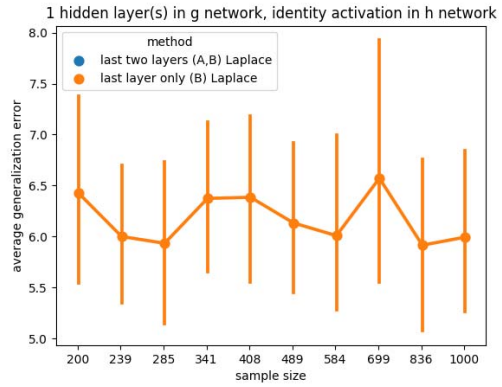
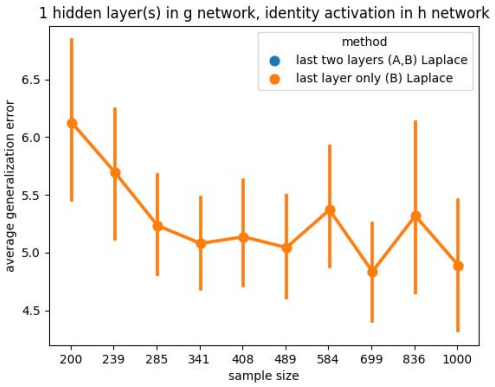


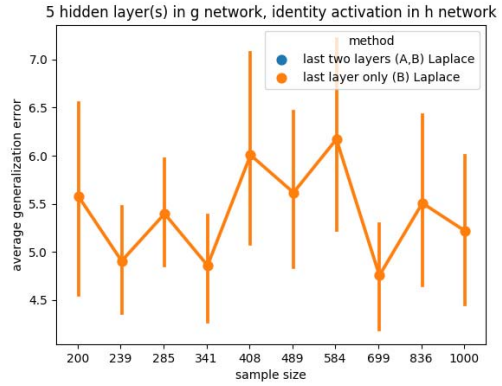
Fig. 7. Realizable and mini-batch gradient descent for MAP training. Details are the same as for Fig. 6.

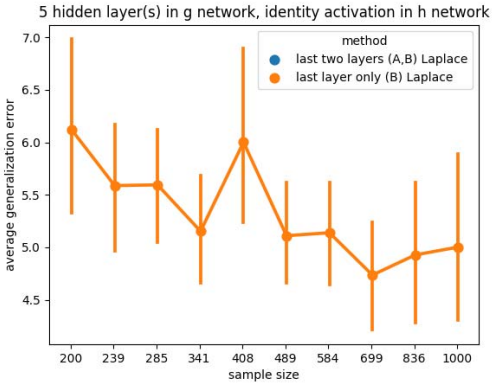
VII. Conclusion

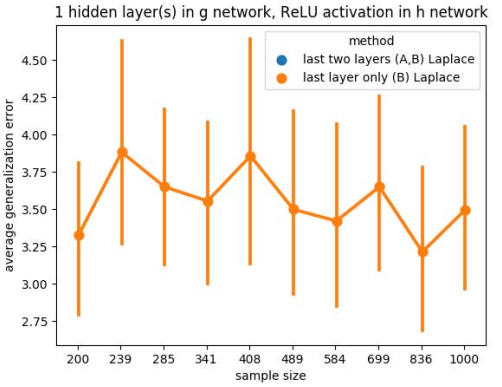
DNNs are singular models, and that's good: the presence of singularities is necessary for neural networks with large numbers of parameters to have low generalization error. Singular learning theory clarifies how classical tools such as the Laplace approximation are not just inappropriate in deep learning on narrow technical grounds: the failure of this

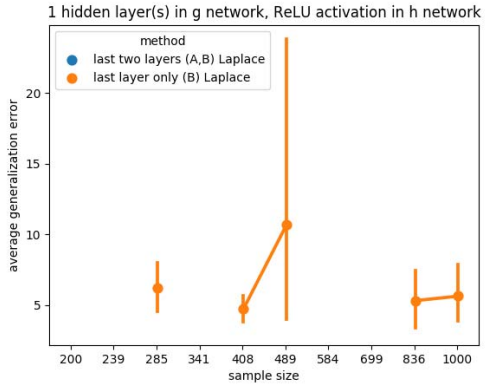


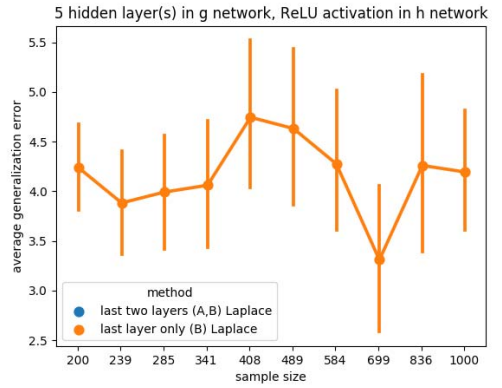












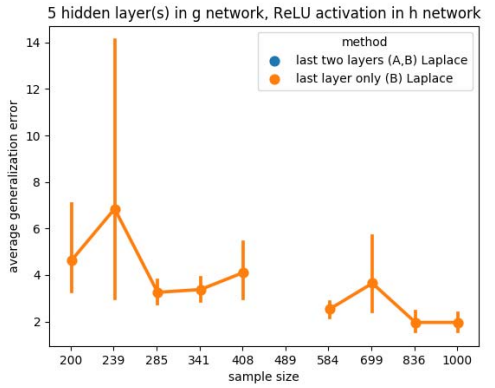


Fig. 8. Nonrealizable and full-batch gradient descent for MAP training. Details are the same as for Fig. 6.

Fig. 9. Nonrealizable and mini-batch gradient descent for MAP training. Details are the same as for Fig. 6.

approximation and the existence of interesting phenomena such as the generalization puzzle have a common cause, namely, the existence of degenerate critical points of the KL function $K(w)$. The singular learning theory is a promising foundation for a mathematical theory of deep learning. However, much remains to be done. The important open problems include the following.

A. SGD Versus the Posterior

A number of works [40]–[42] suggest that the mini-batch SGD may be governed by SDEs that have the posterior distribution as its stationary distribution, and this may go toward understanding why SGD works so well for DNNs.

B. RLCT Estimation for Large Networks

Theoretical RLCTs have been cataloged for small neural networks, albeit at significant effort[4] [33], [34]. We believe that the RLCT estimation in these small networks should be standard benchmarks for any method that purports to approximate the Bayesian posterior of a neural network. No theoretical RLCTs or estimation procedure are known for modern DNNs. Although the MCMC provides the gold standard, it does not scale to large networks. The intractability of RLCT estimation for DNNs is not necessarily an obstacle to reaping the insights offered by the singular learning theory. For instance, used in the context of model selection, the exact value of the RLCT is not as important as model selection consistency. We also demonstrated the utility of singular learning results, such as (V.2) and (V.3), which can be exploited even without knowledge of the exact value of the RLCT.

C. Real-World Distributions Are Unrealizable

The existence of power laws in neural language model training [43], [44] is one of the most remarkable experimental results in deep learning. These power laws may be a sign of interesting new phenomena in singular learning theory when the true distribution is unrealizable.

Appendix A
Neural Networks Are Strictly Singular

Many-layered neural networks are strictly singular [3, Sec. 7.2]. The degeneracy of the Hessian in deep learning has certainly been acknowledged in [45], which recognizes that the eigenspectrum is concentrated around zero, and in [46], which deliberately studies the Fisher information matrix of a single-hidden-layer, rather than multilayer, neural network.

We first explain how to think about a neural network in the context of singular learning theory. A feedforward network of depth c parametrizes a function $f : \mathbb{R}^N \longrightarrow \mathbb{R}^M$ of the form

$$f = A_c \circ \sigma_{c-1} \circ A_{c-1} \circ \cdots \circ \sigma_1 \circ A_1$$

[4]Hironaka's resolution of singularities guarantees existence. However, it is difficult to do the required blowup transformations in high dimensions to obtain the standard form.

where the values of $A_l : \mathbb{R}^{d_{l-1}} \longrightarrow \mathbb{R}^{d_l}$ are affine functions, and $\sigma_l : \mathbb{R}^{d_l} \longrightarrow \mathbb{R}^{d_l}$ is coordinate-wise some fixed nonlinearity $\sigma : \mathbb{R} \longrightarrow \mathbb{R}$. Let W be a compact subspace of $\mathbb{R}^d$ containing the origin, where $\mathbb{R}^d$ is the space of sequences of affine functions $(A_l)_{l=1}^c$ with coordinates denoted as $w_1, \ldots, w_d$, so that f may be viewed as a function $f : \mathbb{R}^N \times W \longrightarrow \mathbb{R}^M$. We define $p(y|x, w)$ as in (III.2). We assume that the true distribution is realizable, $q(y|x) = p(y|x, w_0)$, and that a distribution $q(x)$ on $\mathbb{R}^N$ is fixed with respect to which $p(x, y) = p(y|x)q(x)$ and $q(x, y) = q(y|x)q(x)$. Given some prior $\varphi(w)$ on W, we may apply the singular learning theory to the triplet (p, q, and φ).

By straightforward calculations, we obtain

$$K(w) = \frac{1}{2}\int \|f(x, w) - f(x, w_0)\|^2 q(x)dx \quad \text{(A.1)}$$

$$\frac{\partial^2}{\partial w_i \partial w_j} K(w) = \int \left\langle \frac{\partial}{\partial w_i} f(x, w), \frac{\partial}{\partial w_j} f(x, w) \right\rangle q(x)dx + \int \left\langle f(x, w) - f(x, w_0), \frac{\partial^2}{\partial w_i \partial w_j} f(x, w) \right\rangle q(x)dx \quad \text{(A.2)}$$

$$I(w)_{ij} = \frac{1}{2^{(M-3)/2}\pi^{(M-2)/2}} \times \int \left\langle \frac{\partial}{\partial w_i} f(x, w), \frac{\partial}{\partial w_j} f(x, w) \right\rangle q(x)dx \quad \text{(A.3)}$$

where $\langle -, - \rangle$ is the dot product. We assume $q(x)$ is such that these integrals exist.

It will be convenient in the following to introduce another set of coordinates for W. Let w_{jk}^l denote the weight from the kth neuron in the $(l-1)$th layer to the jth neuron in the lth layer, and let b_j^l denote the bias of the jth neuron in the lth layer. Here, $1 \leq l \leq c$, and the input is layer zero. Let u_j^l and a_j^l denote the value of the jth neuron in the lth layer before and after activation, respectively. Let u^l and a^l denote the vectors with values u_j^l and a_j^l, respectively. Let d_l denote the number of neurons in the lth layer. Then

$$u_j^l = \sum_{k=1}^{d_{l-1}} w_{jk}^l a_k^{l-1} + b_j^l, \quad 1 \leq l \leq c, \quad 1 \leq j \leq d_l$$
$$a_j^l = \sigma\left(u_j^l\right) \quad 1 \leq l < c, \quad 1 \leq j \leq d_l$$

with the convention that $a^0 = x$ is the input and $u^c = y$ is the output.

In the case where σ = ReLU, the partial derivatives $(\partial/\partial w_j)f$ do not exist on all of $\mathbb{R}^N$. However, given $w \in W$, we let $\mathcal{D}(w)$ denote the complement in $\mathbb{R}^N$ of the union over all hidden nodes of the associated decision boundary, that is,

$$\mathbb{R}^N \setminus \mathcal{D}(w) = \bigcup_{1 \leq l < c} \bigcup_{1 \leq j \leq d_l} \left\{x \in \mathbb{R}^N : u_j^l(x) = 0\right\}.$$

The partial derivative $(\partial/\partial w_j)f$ exists on the open subset $\{(x, w) : x \in \mathcal{D}(w)\}$ of $\mathbb{R}^N \times W$.

Lemma 1: Suppose σ = ReLU, and there are $c > 1$ layers. For any hidden neuron $1 \leq j \leq d_l$ in layer l with $1 \leq l < c$,

there is a differential equation

$$\left\{\sum_{k=1}^{d_{l-1}} w_{jk}^l \frac{\partial}{\partial w_{jk}^l} + b_j^l \frac{\partial}{\partial b_j^l} - \sum_{i=1}^{d_{l+1}} w_{ij}^{l+1} \frac{\partial}{\partial w_{ij}^{l+1}}\right\} f = 0$$

which holds on $\mathcal{D}(w)$ for any fixed $w \in W$.

Proof: Without loss of generality, assume $M = 1$, to simplify the notation. Let $e_i \in \mathbb{R}^{d_{l+1}}$ denote a unit vector, and let $H(x) = (d/dx)\,\mathrm{ReLU}(x)$. Writing $(\partial f/(\partial u^{l+1}))$ for a gradient vector

$$\begin{aligned}
\frac{\partial f}{\partial w_{ij}^{l+1}} &= \left\langle \frac{\partial f}{\partial u^{l+1}}, \frac{\partial u^{l+1}}{\partial w_{ij}^{l+1}} \right\rangle = \left\langle \frac{\partial f}{\partial u^{l+1}}, a_j^l e_i \right\rangle \\
&= \frac{\partial f}{\partial u_i^{l+1}} u_j^l H\left(u_j^l\right) \\
\frac{\partial f}{\partial w_{jk}^{l}} &= \left\langle \frac{\partial f}{\partial u^{l+1}}, \frac{\partial u^{l+1}}{\partial w_{jk}^{l}} \right\rangle = \left\langle \frac{\partial f}{\partial u^{l+1}}, \sum_{i=1}^{d_{l+1}} w_{ij}^{l+1} a_k^{l-1} H\left(u_j^l\right) e_i \right\rangle \\
&= \sum_{i=1}^{d_{l+1}} \frac{\partial f}{\partial u_i^{l+1}} w_{ij}^{l+1} a_k^{l-1} H\left(u_j^l\right) \\
\frac{\partial f}{\partial b_j^{l}} &= \left\langle \frac{\partial f}{\partial u^{l+1}}, \frac{\partial u^{l+1}}{\partial b_j^{l}} \right\rangle = \left\langle \frac{\partial f}{\partial u^{l+1}}, \sum_{i=1}^{d_{l+1}} w_{ij}^{l+1} H\left(u_j^l\right) e_i \right\rangle \\
&= \sum_{i=1}^{d_{l+1}} \frac{\partial f}{\partial u_i^{l+1}} w_{ij}^{l+1} H\left(u_j^l\right).
\end{aligned}$$

The claim immediately follows. □

Lemma 2: Suppose $\sigma = \mathrm{ReLU}$, $c > 1$, and that $w \in W$ has at least one weight or bias at a hidden node nonzero. Then, the matrix $I(w)$ is degenerate, and if $w \in W_0$, then the Hessian of K at w is also degenerated.

Proof: Let $w \in W$ be given, and choose a hidden node where at least one of the incident weights (or bias) is nonzero. Then, Lemma 1 gives a nontrivial linear dependence relation $\sum_i \lambda_i(\partial/\partial w_i) f = 0$ as the functions on $\mathcal{D}(w)$. The rows of $I(w)$ satisfy the same linear dependence relation. At a true parameter, the second summand in (A.2) vanishes, so by the same argument, the Hessian is degenerated. □

Remark 1: Lemma 2 implies that every true parameter for a nontrivial ReLU network is a degenerated critical point of K. Hence, in the study of nontrivial ReLU networks, it is never appropriate to divide by the determinant of the Hessian of K at a true parameter, and in particular, Laplace or saddle point approximations at a true parameter are invalid.

The well-known positive scale invariance of ReLU networks [47] is responsible for the linear dependence of Lemma 1, in the precise sense that the given differential operator is the infinitesimal generator [48, Sec. IV.3] of the scaling symmetry. However, this is only one source of degeneracy or singularity in ReLU networks. The degeneracy, as measured by the RLCT, is much lower than one would expect on the basis of this symmetry alone (see Section VI).

Example 1: In general, the KL function $K(w)$ for ReLU networks is not analytic. For the minimal counterexample, let $q(x)$ be uniform on $[-N, N]$ and zero outside, and consider

$$K(b) = \int q(x)(\mathrm{ReLU}(x-b) - \mathrm{ReLU}(x))^2\, dx.$$

It is easy to check that up to a scalar factor

$$K(b) = \begin{cases} -\frac{2}{3}b^3 + b^2\ N, & 0 \le b \le N \\ -\frac{1}{3}b^3 + b^2\ N, & -N \le b \le 0 \end{cases}$$

so that K is C^2, but not C^3 let alone analytic.

Appendix B
Reduced Rank Regression

For reduced rank regression, the model is

$$p(y|x, w) = \frac{1}{\left(2\pi\sigma^2\right)^{N/2}} \exp\left(-\frac{1}{2\sigma^2}|y - BAx|^2\right)$$

where $x \in \mathbb{R}^M, y \in \mathbb{R}^N$, A an $M \times H$ matrix, and B an $H \times N$ matrix; the parameter w denotes the entries of A and B; i.e., $w = (A, B)$, and $\sigma > 0$ is the standard deviation of the observation noise.

If the true distribution is realizable, then there is $w_0 = (A_0, B_0)$, such that $q(y|x) = p(y|x, w_0)$. Without loss of generality, assume that $q(x)$ is the uniform density. In this case, the KL divergence from $p(y|x, w)$ to $q(y|x)$ is

$$\begin{aligned}
K(w) &= \int q(y|x) \log \frac{q(y|x)}{p(y|x, w)} dx dy \\
&= \|BA - B_0A_0\|^2(1 + E(w))
\end{aligned}$$

where the error E is smooth, and $E(w) = O(\|BA - B_0A_0\|^2)$ in any region where $\|BA - B_0A_0\| < C$, so $K(w)$ is equivalent to $\|BA - B_0A_0\|^2$. We write $K(w) = \|BA - B_0A_0\|^2$ for simplicity in the following.

Now assume that B_0A_0 is symmetric and that B_0 is square; i.e. $N = H$. Then, the zero locus of $K(w)$ is explicitly given as follows:

$$W_0 = \left\{(A, B) : \det B \neq 0 \text{ and } A = B^{-1}B_0A_0\right\}.$$

It follows that W_0 is globally a graph over $GL(H; \mathbb{R})$. Indeed, the set $(B^{-1}B_0A_0, B)$ with $B \in GL(H; \mathbb{R})$ is exactly W_0. Thus, W_0 is a smooth H^2-dimensional submanifold of $\mathbb{R}^{H^2} \times \mathbb{R}^{H \times M}$. To prove that W_0 is minimally singular in the sense of Section IV, it suffices to show that $\mathrm{rank}(D^2_{A,B}K) \ge HM$, where $D^2_{A,B}K$ denotes the Hessian, but as it is no more difficult to do so, we find explicit local coordinates (u, v) near an arbitrary point $(\overline{A}, \overline{B}) \in W_0$ for which $\{v = 0\} = W_0$ and $K(u, v) = a(u, v)|u|^2$ in this neighborhood, where a is a C^∞ function with $a \ge c > 0$ for some c. Write

$$A(v) = \left(\overline{B} + v\right)^{-1} B_0A_0.$$

Then, $u, v \mapsto (A(v) + u, \overline{B} + v)$ gives local coordinates on $\mathbb{R}^{H^2} \times \mathbb{R}^{H \times M}$ near $(\overline{A}, \overline{B})$, and

$$\begin{aligned}
K(u, v) &= \left|\left(\overline{B} + v\right)\left(\left(\overline{B} + v\right)^{-1} B_0A_0 + u\right) - B_0A_0\right| \\
&= \left|B_0A_0 + \left(\overline{B} + v\right)u - B_0A_0\right|^2 \\
&= \left|\left(\overline{B} + v\right)u\right|^2
\end{aligned}$$

so for v sufficiently small (and hence $\overline{B} + v$ invertible), we can take $a(u, v) = |(\overline{B} + v)u|^2/|u|^2$.

APPENDIX C
RLCT ESTIMATION

In this section, we detail the estimation procedure for the RLCT used in Section VI. Let $L_n(w)$ be the negative log likelihood as in (III.3). Define the data likelihood at inverse temperature $\beta > 0$ to be $p^\beta(\mathcal{D}_n|w) = \Pi_{i=1}^n p(y_i|x_i, w)^\beta$, which can also be written as

$$p^\beta(\mathcal{D}_n|w) = \exp(-\beta n L_n(w)). \tag{C.1}$$

The posterior distribution, at inverse temperature β, is defined as

$$p^\beta(w|\mathcal{D}_n) = \frac{\Pi_{i=1}^n p(y_i|x_i, w)^\beta \varphi(w)}{\int_W \Pi_{i=1}^n p(y_i|x_i, w)^\beta \varphi(w)} = \frac{p^\beta(\mathcal{D}_n|w)\varphi(w)}{p^\beta(\mathcal{D}_n)} \tag{C.2}$$

where φ is the prior distribution on the network weights w, and

$$p^\beta(\mathcal{D}_n) = \int_W p^\beta(\mathcal{D}_n|w)\varphi(w)\, dw \tag{C.3}$$

is the marginal likelihood of the data at inverse temperature β. Finally, denote the expectation of a random variable $R(w)$ with respect to the tempered posterior $p^\beta(w|\mathcal{D}_n)$ as

$$\mathbb{E}_w^\beta[R(w)] = \int_W R(w)\, p^\beta(w|\mathcal{D}_n)\, dw. \tag{C.4}$$

In the main text, we drop the superscript in the quantities (C.1)–(C.4) when $\beta = 1$, e.g., $p(\mathcal{D}_n)$ rather than $p^1(\mathcal{D}_n)$.

Assuming that the conditions of [23, Th. 4] hold, we have

$$\mathbb{E}_w^\beta[nL_n(w)] = nL_n(w_0) + \frac{\lambda}{\beta} + U_n\sqrt{\frac{\lambda}{2\beta}} + O_p(1) \tag{C.5}$$

where β_0 is a positive constant, and U_n is a sequence of random variables satisfying $\mathbb{E}_n U_n = 0$. In Algorithm 1, we describe an estimation procedure for the RLCT based on the asymptotic result in (C.5). We should note that Algorithm 1 is a simple improvement over Watanabe's RLCT estimation in [23, Sec. 6.2]. In particular, instead of estimating the RLCT based on two values of β, we use more β's to improve the estimation. When the set of β's in Algorithm 1 contains simply two elements, Algorithm 1 completely reduces to Watanabe's RLCT estimation. Empirical verification of this algorithm can be found in [23, Table 3] when the RLCT estimation is seen to recover the true RLCT in a small reduced rank regression problem.

For the estimates in Table II the *a posteriori* distribution was approximated using the NUTS variant of HMC [37], where the first 1000 steps were omitted, and 20 000 samples were collected. Each $\hat{\lambda}(\mathcal{D}_n)$ estimate in Algorithm 1 was performed by linear regression on the pairs $\{(1/\beta_i, \mathbb{E}_w^{\beta_i}[nL_n(w)])\}_{i=1}^5$, where the five inverse temperatures β_i are centered on the inverse temperature $1/\log(20\,000)$.

Algorithm 1 RLCT via [23, Th. 4]

Input: range of β's, set of training sets $\mathcal{T}$ each of size n, approximate samples $\{w_1, \ldots, w_R\}$ from $p^\beta(w|\mathcal{D}_n)$ for each training set $\mathcal{D}_n$ and each β
for training set $\mathcal{D}_n \in \mathcal{T}$ **do**
  for β in range of β's **do**
    Approximate $\mathbb{E}_w^\beta[nL_n(w)]$ with $\frac{1}{R}\sum_{i=1}^R nL_n(w_r)$ where $w_1, \ldots, w_R$ are approximate samples from $p^\beta(w|\mathcal{D}_n)$
  end for
  Perform generalised least squares to fit λ in (C.5), call result $\hat{\lambda}(\mathcal{D}_n)$
end for
Output: $\frac{1}{|\mathcal{T}|}\sum_{\mathcal{D}_n\in\mathcal{T}} \hat{\lambda}(\mathcal{D}_n)$

APPENDIX D
CONNECTION BETWEEN RLCT AND GENERALIZATION

For completeness, we sketch the derivation of (V.2), which gives the asymptotic expansion of the average generalization error $\mathbb{E}_n G(n)$ of the Bayes prediction distribution in singular models. The exposition is an amalgamation of various works published by Sumio Watanabe, but is mostly based on the textbook [3].

To understand the connection between the RLCT and $G(n)$, we first define the so-called Bayes free energy as

$$F(n) = -\log p(\mathcal{D}_n)$$

whose expectation admits the following asymptotic expansion [3]:

$$\mathbb{E}_n F(n) = \mathbb{E}_n n S_n + \lambda \log n + o(\log n)$$

where $S_n = -(1/n)\sum_{i=1}^n \log q(y_i|x_i)$ is the entropy. The expected Bayesian generalization error is related to the Bayes free energy as follows:

$$\mathbb{E}_n G(n) = \mathbb{E}\, F(n+1) - \mathbb{E}\, F(n).$$

Then, for the average generalization error, we have Equation (V.2). As models with more complex singularities have smaller RLCTs, this would suggest that the more singular the model is, the better its generalization (assuming one uses the Bayesian predictive distribution for prediction). In this connection, it is interesting to note that simpler (relative to the model) true distributions lead to more singular models (Section VI).

APPENDIX E
DETAILS FOR GENERALIZATION ERROR EXPERIMENTS

A. Simulated Data

The distribution of $x \in \mathbb{R}^3$ is set to $q(x) = N(0, I_3)$. In the realizable case, $y \in \mathbb{R}^3$ is drawn according to $q(y|x) = p(y|x, \theta_0)$. In the nonrealizable setting, we set $q(y|x) \propto \exp\{-||y - h_{w_0}(x)||^2/2\}$, where $w_0 = (A_0, B_0)$ is drawn according to the PyTorch model initialization of h.

B. MAP Training

The MAP estimator is found via gradient descent using the mean-squared-error loss with either the full dataset or mini-batch set to 32. Training was set to 5000 epochs. No form of early stopping was employed.

C. Calculating the Generalization Error

Using a held-out-test set $T_{n'} = \{(x'_i, y'_i)\}_{i=1}^{n'}$, we calculate the average generalization error as

$$\frac{1}{n'}\sum_{i=1}^{n'} \log q\left(y'_i|x'_i\right) - \mathbb{E}_n \frac{1}{n'}\sum_{i=1}^{n'} \log \hat{q}_n\left(y'_i|x'_i\right). \tag{E.1}$$

Assume that the held-out test set is large enough, so that the difference between $\mathbb{E}_n G(n)$ and (E.1) is negligible. We will refer to them interchangeably as the average generalization error. In our experiments, we use $n' = 10\,000$ and 30 draws of the dataset $\mathcal{D}_n$ to estimate $\mathbb{E}_n$.

D. Last Layer(s) Inference

Without loss of generality, we discuss performing inference in the w parameters of h while freezing the parameters of g at the MAP estimate. The steps easily extend to performing inference over the final layer only of $f = h \circ g$. Let $\tilde{x}_i = g_{v_{\text{MAP}}}(x_i)$. Define a new transformed dataset $\tilde{\mathcal{D}}_n = \{(\tilde{x}_i, y_i)\}_{i=1}^n$. We take the prior on w to be standard Gaussian. Define the posterior over w given $\tilde{\mathcal{D}}_n$ as

$$\begin{aligned} p\left(w|\tilde{\mathcal{D}}_n\right) &\propto p\left(\tilde{\mathcal{D}}_n|w\right)\varphi(w) \\ &= \Pi_{i=1}^n \exp\left\{-||y_i - h_w(\tilde{x}_i)||^2/2\right\}\varphi(w). \end{aligned} \tag{E.2}$$

Define the following approximation to the Bayesian predictive distribution:

$$\tilde{p}(y|x, \mathcal{D}_n) = \int p(y|x, (v_{\text{MAP}}, w))\, p\left(w|\tilde{\mathcal{D}}_n\right) dw.$$

Let $w_1, \ldots, w_R$ be some approximate samples from $p(w|\tilde{\mathcal{D}}_n)$. Then, we approximate $\tilde{p}(y|x, \mathcal{D}_n)$ with

$$\frac{1}{R}\sum_{r=1}^{R} p(y|x, (v_{\text{MAP}}, w_r))$$

where R is a large number, set to 1000 in our experiments. We consider the Laplace approximation and the NUTS variant of HMC for drawing samples from $p(w|\tilde{\mathcal{D}}_n)$.

1) *Laplace in the Last Layer(s):* Recall $\theta_{\text{MAP}} = (v_{\text{MAP}}, w_{\text{MAP}})$ is the MAP estimate for f_θ trained with the data $\mathcal{D}_n$. With the Laplace approximation, we draw $w_1, \ldots, w_R$ from the Gaussian

$$N(w_{\text{MAP}}, \Sigma)$$

where $\Sigma = (-\nabla^2 \log p(w|\tilde{\mathcal{D}}_n)|_{w_{\text{MAP}}})^{-1}$ is the inverse Hessian[5] of the negative log posterior evaluated at the MAP estimate of the mode.

2) *MCMC in the Last Layer(s):* We used the NUTS variant of HMC to draw samples from (E.2) with the first 1000 samples discarded. Our implementation used the `pyro` package in `PyTorch`.

E. Additional Figures and Tables for Section V

Due to space, Figs. 3–9 will appear on preceding pages before this section.

[5]Following [36], the code for the exact Hessian calculation is borrowed from https://github.com/f-dangel/hbp

ACKNOWLEDGMENT

The authors would like to thank Michelle Chen for helpful discussions.

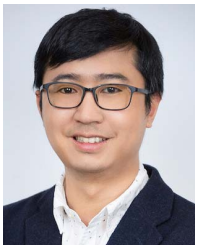
REFERENCES

[1] S.-I. Amari, T. Ozeki, and H. Park, "Learning and inference in hierarchical models with singularities," *Syst. Comput. Jpn.*, vol. 34, no. 7, pp. 34–42, Jun. 2003.
[2] S. Watanabe, "Almost all learning machines are singular," in *Proc. IEEE Symp. Found. Comput. Intell.*, Apr. 2007, pp. 383–388.
[3] S. Watanabe, *Algebraic Geometry and Statistical Learning Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[4] S. L. Smith and Q. V. Le, "A Bayesian perspective on generalization and stochastic gradient descent," 2017, *arXiv:1710.06451*.
[5] Y. Zhang, A. M. Saxe, M. S. Advani, and A. A. Lee, "Energy–entropy competition and the effectiveness of stochastic gradient descent in machine learning," *Mol. Phys.*, vol. 116, nos. 21–22, pp. 3214–3223, Nov. 2018.
[6] O. Bousquet, S. Boucheron, and G. Lugosi, "Introduction to statistical learning theory," in *Summer School on Machine Learning*. Cham, Switzerland: Springer, 2003, pp. 169–207.
[7] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017.
[8] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018.
[9] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 242–252.
[10] B. Neyshabur, R. Tomioka, and N. Srebro, "Norm-based capacity control in neural networks," in *Proc. Conf. Learn. Theory*, 2015, pp. 1376–1401.
[11] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6240–6249.
[12] B. Neyshabur and Z. Li, "Towards understanding the role of over-parametrization in generalization of neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.
[13] S. Arora, R. Ge, B. Neyshabur, and Y. Zhang, "Stronger generalization bounds for deep nets via a compression approach," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 254–263.
[14] A. Brutzkus, A. Globerson, E. Malach, and S. Shalev-Shwartz, "SGD learns over-parameterized networks that provably generalize on linearly separable data," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–17.
[15] Y. Li and Y. Liang, "Learning overparameterized neural networks via stochastic gradient descent on structured data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8157–8166.
[16] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in overparameterized neural networks, going beyond two layers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 6155–6166.
[17] A. Daniely, "SGD learns the conjugate kernel class of the network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2422–2430.
[18] S. Arora, S. Du, W. Hu, Z. Li, and R. Wang, "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 322–332.
[19] G. Yehudai and O. Shamir, "On the power and limitations of random features for understanding neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 6594–6604.
[20] Y. Cao and Q. Gu, "Generalization bounds of stochastic gradient descent for wide and deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10835–10845.
[21] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks (invited paper)," in *Proc. 53rd Annu. ACM SIGACT Symp. Theory Comput.*, Jun. 2021, pp. 8571–8580.
[22] S. Watanabe, *Mathematical Theory of Bayesian Statistics*. Boca Raton, FL, USA: CRC Press, 2018.
[23] S. Watanabe, "A widely applicable Bayesian information criterion," *J. Mach. Learn. Res.*, vol. 14, pp. 867–897, Mar. 2013.
[24] G. E. Hinton and D. van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proc. 6th Annu. Conf. Comput. Learn. Theory (COLT)*, 1993, pp. 5–13.

[25] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Comput.*, vol. 9, no. 1, pp. 1–42, 1997.
[26] P. Chaudhari *et al.*, "Entropy-SGD: Biasing gradient descent into wide valleys," in *Proc. Int. Conf. Learn. Represent.*, 2017.
[27] S. Jastrzebski *et al.*, "Three factors influencing minima in SGD," 2017, *arXiv:1711.04623*.
[28] V. Balasubramanian, "Statistical inference, Occam's razor, and statistical mechanics on the space of probability distributions," *Neural Comput.*, vol. 9, no. 2, pp. 349–368, Feb. 1997.
[29] T. A. Poggio *et al.*, "Theory of deep learning III: Explaining the non-overfitting puzzle," 2018, *arXiv:1801.00173*.
[30] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5947–5956.
[31] V. Thomas, F. Pedregosa, B. van Merriënboer, P.-A. Mangazol, Y. Bengio, and N. L. Roux, "On the interplay between noise and curvature and its effect on optimization and generalization," 2019, *arXiv:1906.07774*.
[32] W. J. Maddox, G. Benton, and A. G. Wilson, "Rethinking parameter counting in deep models: Effective dimensionality revisited," 2020, *arXiv:2003.02139*.
[33] M. Aoyagi and S. Watanabe, "Stochastic complexities of reduced rank regression in Bayesian estimation," *Neural Netw.*, vol. 18, no. 7, pp. 924–933, Sep. 2005.
[34] M. Aoyagi and S. Watanabe, "Resolution of singularities and the generalization error with Bayesian estimation for layered neural network," in *Proc. IEICE Trans.*, 2005, pp. 2112–2124.
[35] S. Nakajima and S. Watanabe, "Variational Bayes solution of linear neural networks and its generalization performance," *Neural Comput.*, vol. 19, no. 4, pp. 1112–1153, 2007.
[36] A. Kristiadi, M. Hein, and P. Hennig, "Being Bayesian, even just a bit, fixes overconfidence in ReLU networks," 2020, *arXiv:2002.10118*.
[37] M. D. Hoffman and A. Gelman, "The no-u-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1593–1623, Apr. 2014.
[38] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.
[39] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.
[40] U. ŞimŠekli, "Fractional Langevin Monte Carlo: Exploring Levy driven stochastic differential equations for Markov Chain Monte Carlo," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3200–3209.
[41] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic gradient descent as approximate Bayesian inference," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 4873–4907, 2017.
[42] S. L. Smith, D. Duckworth, S. Rezchikov, Q. V. Le, and J. Sohl-Dickstein, "Stochastic natural gradient descent draws posterior samples in function space," 2018, *arXiv:1806.09597*.
[43] J. Hestness *et al.*, "Deep learning scaling is predictable, empirically," 2017, *arXiv:1712.00409*.
[44] J. Kaplan *et al.*, "Scaling laws for neural language models," 2020, *arXiv:2001.08361*.
[45] L. Sagun, L. Bottou, and Y. LeCun, "Singularity of the Hessian in deep learning," 2016, *arXiv:1611.07476*.
[46] J. Pennington and P. Worah, "The spectrum of the Fisher information matrix of a single-hidden-layer neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5410–5419.
[47] M. Phuong and C. H. Lampert, "Functional vs. parametric equivalence of ReLU networks," in *Proc. Int. Conf. Learn. Represent.*, 2020.
[48] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*. New York, NY, USA: Academic, 1986.

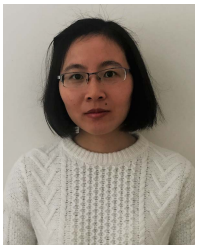
Susan Wei is currently a Lecturer (Assistant Professor) in data science with the School of Mathematics and Statistics, The University of Melbourne, Melbourne, VIC, Australia. Her research interests include statistics and deep learning.

Daniel Murfet is currently a Mathematician with The University of Melbourne, Melbourne, VIC, Australia. His primary research area is algebraic geometry, but he likes to find geometry in all sorts of things, from deep learning to logic and mathematical physics.



Mingming Gong is currently a Lecturer (Assistant Professor) in data science with the School of Mathematics and Statistics, The University of Melbourne, Melbourne, VIC, Australia. He has authored/coauthored more than 30 research papers on top venues, such as International Conference on Machine Learning (ICML), Neural Information Processing Systems (NeurIPS), the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and the IEEE TRANSACTIONS ON IMAGE PROCESSING, with more than ten oral/spotlight presentations. His research interests include causal reasoning, machine learning, and computer vision.

Dr. Gong received the Discovery Early Career Researcher Award from the Australian Research Council in 2020.



Hui Li received the B.S. degree in biotechnology from the Huazhong University of Science and Technology, Wuhan, China, in 2009, and the master's degree in biostatistics from the University of Minnesota, Minneapolis, MN, USA, in 2018. She is currently pursuing the Ph.D. degree with the School of Mathematics and Statistics, The University of Melbourne, Melbourne, VIC, Australia.

Her current research interests include deep learning and its application in medical data analysis (e.g., single-cell ribonucleic acid (RNA) sequence data and medical image data).

Jesse Gell-Redman is currently a Senior Lecturer in analysis and partial differential equations (PDEs) with the School of Mathematics and Statistics, The University of Melbourne, Melbourne, VIC, Australia. He is also a primary investigator on two active grants from the Australian Research Council. He has authored papers in pure mathematics journals, such as *The Journal of Differential Geometry and Communications in Mathematical Physics*. His research interests include microlocal analysis, propagation phenomena in evolution equations, index theory and geometric invariants, and geometric analysis of singular spaces.

Thomas Quella received the M.Sc. degree from Freie Universität Berlin, Berlin, Germany, in 1999, and the Ph.D. degree from Humboldt-Universität zu Berlin, Berlin, Germany, in 2003.

After two postdoctoral appointments between 2003 and 2010, one at King's College London, London, U.K., and one at the University of Amsterdam, Amsterdam, The Netherlands, he led an independent Junior Research Group, University of Cologne, Cologne, Germany, from 2010 to 2016. Since 2016, he has been a Senior Lecturer in mathematical physics with the School of Mathematics and Statistics, The University of Melbourne, Melbourne, VIC, Australia. His current research interests include statistical physics, complex systems, and critical phenomena.