Dissecting Non-Vacuous Generalization Bounds based on the Mean-Field Approximation–Appendix

Konstantinos Pitas [1]

[1]École Polytechnique Fédérale de Lausanne, Switzerland. Correspondence to: Konstantinos Pitas <konstantinos.pitas@epfl.ch>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

Appendix

A. Derivations for valid bound

We present again for clarity the PAC-Bayes bound by Catoni (2007).

Theorem 2.1. (Catoni, 2007) Given a distribution $\mathcal{D}$ over $\mathcal{X}\times\mathcal{Y}$, a hypothesis set $\mathcal{F}$, a loss function $\ell' : \mathcal{F}\times\mathcal{X}\times\mathcal{Y}\to[0,1]$, a prior distribution π over $\mathcal{F}$, a real number $\delta\in(0,1]$, and a real number $\beta>0$, with probability at least $1-\delta$ over the choice of $(X,Y)\sim\mathcal{D}^n$, we have

$$\begin{aligned}\forall\hat{\rho}\text{ on }\mathcal{F}:\mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell'}_{\mathcal{D}}(f)\leq&\Phi_\beta^{-1}(\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f)\\&+\frac{1}{\beta n}(\mathrm{KL}(\hat{\rho}||\pi)+\ln\frac{1}{\delta})),\end{aligned}\tag{1}$$

where $\Phi_\beta^{-1}(x)=\frac{1-e^{-\beta x}}{1-e^{-\beta}}$.

Evaluating a valid PAC-Bayes bound, using empirical estimates, requires some care.

Optimizing λ. For a start, when modeling $\pi(\boldsymbol{\theta})=\mathcal{N}(0,\lambda\mathbf{I})$, it is often beneficial to optimize the hyperparameter λ. As the PAC-Bayes theorem requires the prior to be independent from the posterior, we need to take a union bound over an appropriately chosen grid, representing different possible values of λ. Following (Dziugaite & Roy, 2017), we can choose $\lambda=c\exp\{-j/b\}$ for $j\in\mathbb{N}$ and fixed $b,c\geq 0$, where c corresponds to the grid scale and b to it's precision. Then, if the PAC-Bayes bound for each $j\in\mathbb{N}$ is designed to hold with probability at least $1-\frac{6\delta}{\pi^2 j^2}$, by union bound it will hold uniformly for all $j\in\mathbb{N}$ with probability at least $1-(\frac{6\delta}{\pi^2})\sum_{j\in\mathbb{N}}\frac{1}{j^2}=1-\delta$. We solve for $j=b\log\frac{c}{\lambda}$ and substitute this value in the probability for each term in the union bound. We get that *any* bound corresponding to $j\in\mathbb{N}$ holds with probability $1-\frac{6\delta}{\pi^2 b^2\ln(c/\lambda^2)}$. Thus looking back to theorem 2.1 the term $\ln\frac{1}{\delta}$ becomes $\ln\frac{\pi^2b^2\ln(c/\lambda^2)}{6\delta}$. In practice we see that even for very large numbers c,b,δ when divided by the number of samples n the term $\ln\frac{\pi^2b^2\ln(c/\lambda^2)}{6\delta}$ is negligible and we treat j as a continuous number.

Empirical estimate of $\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}^*(\boldsymbol{\theta})}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})$. Furthermore, assuming an optimized posterior $\hat{\rho}^*(\boldsymbol{\theta})$ directly evaluating $\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}^*(\boldsymbol{\theta})}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})$ is intractable. Instead, since $\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})$ is a bounded random variable, one can approximate the expectation using Monte Carlo sampling and use a Chernoff bound to bound it's tail. Let $\tilde{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})\equiv(1/m)\sum_{i=0}^{m}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}_i})$ be the observed failure rate of m random hypotheses drawn according to $\hat{\rho}^*(\boldsymbol{\theta})$. One can then show the following (Langford & Caruana, 2002) (presented here without proof)

Theorem 0.1. *(Sample Convergence Bound) For all distributions, $\hat{\rho}^*(\boldsymbol{\theta})$, for all sample sets (X,Y), assuming that $\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})\in[0,1]$*

$$\begin{aligned}&\Pr_{\hat{\rho}^*(\boldsymbol{\theta})}(\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}^*(\boldsymbol{\theta})}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})\leq\tilde{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})+\sqrt{\frac{\ln\frac{2}{\delta'}}{m}})\\&\leq\delta',\end{aligned}\tag{2}$$

where m is the number of evaluations of the stochastic hypothesis.

We take a union bound over values of λ, and apply the Chernoff bound for the tail of the empirical estimate of $\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f)$. Putting everything together, one can obtain valid PAC-Bayes bounds subject to a posterior distribution $\hat{\rho}^*(\boldsymbol{\theta})$ that hold with probability at least $1-\delta-\delta'$ and are of the form

$$\begin{aligned}\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}^*(\boldsymbol{\theta})}\mathcal{L}^{\ell'}_{\mathcal{D}}(f_{\boldsymbol{\theta}})\leq&\Phi_\beta^{-1}(\tilde{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})+\frac{1}{\beta n}\mathrm{KL}(\hat{\rho}^*(\boldsymbol{\theta})||\pi)\\&+\frac{1}{\beta n}\ln(\frac{\pi^2b^2\ln(c/\lambda)^2}{6\delta})+\sqrt{\frac{\ln\frac{2}{\delta'}}{m}}),\end{aligned}\tag{3}$$

where $\Phi_\beta^{-1}(x)=\frac{1-e^{-\beta x}}{1-e^{-\beta}}$. Also c,b are constants, m is the number of samples from $\hat{\rho}$ for approximating $\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f)$ and $\tilde{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})$ the empirical estimate.

Number of samples for Chernoff bound. In our experiments we use $m = 1000$ for all experiments including VI experiments. We make a single exception due to time constraints for the case of optimizing the posterior in closed form (Section 4.1 equation 6) where we use $m = 100$.

- For $m = 1000$ and $\delta' = 0.05$, this gives bounds with confidence $\sqrt{\frac{\log(2/0.05)}{1000}} \approx 0.06$.
- For $m = 100$ and $\delta' = 0.05$, this gives bounds with confidence $\sqrt{\frac{\log(2/0.05)}{100}} \approx 0.19$.

Importantly bounds with even higher confidence $\sqrt{\frac{\log(2/0.05)}{10000}} \approx 0.019$ and sample size $m = \mathcal{O}(10^4)$ are possible for all experiments with a computational time in the order of weeks. However we consider this point a technicality as the Chernoff bound is quite pessimistic. Empirically the estimates in our experiments converge much faster than implied by the bound analysis, exhibiting no significant difference between $m = 1000$, $m = 100$ or even $m = 10$ in the isotropic cases. This is because this particular Chernoff bound is an application of Hoeffding's inequality for general bounded random variables (Vershynin, 2018)[p. 25]. The only assumption is that the random variable is bounded $\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}}) \in [0, 1]$, and thus the variance of the random variable is significantly overestimated.

B. Proof of Lemma 4.1

Lemma 4.1. The convex optimization problem $\min_{\mathbf{\Sigma}_{\hat{\rho}}} \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}^T\mathbf{H}\boldsymbol{\eta}] + \beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta}))$ where $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \mathbf{\Sigma}_{\hat{\rho}})$ and $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\pi}, \lambda\mathbf{\Sigma}_{\pi})$ is minimized at

$$\mathbf{\Sigma}^*_{\hat{\rho}} = \beta(\mathbf{H} + \frac{\beta}{\lambda}\mathbf{\Sigma}_{\pi}^{-1})^{-1}, \tag{4}$$

where $\mathbf{H} \equiv \nabla^2\hat{\mathcal{L}}^{\ell_{\text{cat}}}_{X,Y}(f_{\boldsymbol{\theta}})$ captures the curvature at the minimum, while $\mathbf{\Sigma}_{\pi}$ is the prior covariance.

Proof.

$$\begin{aligned} C_\beta(X,Y;\hat{\rho},\pi) &= \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}^T\mathbf{H}\boldsymbol{\eta}] + \beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta})) \\ &= \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\frac{1}{2}\mathrm{tr}(\mathbf{H}\boldsymbol{\eta}\boldsymbol{\eta}^T)] + \beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta})) \\ &= \frac{1}{2}\mathrm{tr}(\mathbf{H}\mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\boldsymbol{\eta}\boldsymbol{\eta}^T]) + \beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta})) \\ &= \frac{1}{2}\mathrm{tr}(\mathbf{H}\mathbf{\Sigma}_{\hat{\rho}}) + \frac{\beta}{2}(\mathrm{tr}(\frac{1}{\lambda}\mathbf{\Sigma}_{\pi}^{-1}\mathbf{\Sigma}_{\hat{\rho}}) - k + \frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\pi})^{\mathrm{T}}\mathbf{\Sigma}_{\pi}^{-1}(\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\pi}) \\ &\quad + \ln\left(\frac{\det\lambda\mathbf{\Sigma}_{\pi}}{\det\mathbf{\Sigma}_{\hat{\rho}}}\right)) \end{aligned} \tag{5}$$

The gradient with respect to $\mathbf{\Sigma}_{\hat{\rho}}$ is

$$\frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial\mathbf{\Sigma}_{\hat{\rho}}} = [\frac{1}{2}\mathbf{H} + \frac{\beta}{2\lambda}\mathbf{\Sigma}_{\pi}^{-1} - \frac{\beta}{2}\mathbf{\Sigma}_{\hat{\rho}}^{-1}]. \tag{6}$$

Setting it to zero, we obtain the minimizer $\mathbf{\Sigma}^*_{\hat{\rho}} = \beta(\mathbf{H} + \frac{\beta}{\lambda}\mathbf{\Sigma}_{\pi}^{-1})^{-1}$. □

C. Proof of Lemma 4.2

Lemma 4.2. The optimal prior and posterior covariances for $\min_{\boldsymbol{\sigma}_{\hat{\rho}},\boldsymbol{\sigma}_{\pi}} C_\beta(X,Y;\hat{\rho},\pi) = \min_{\boldsymbol{\sigma}_{\hat{\rho}},\boldsymbol{\sigma}_{\pi}} \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}^T\mathbf{H}\boldsymbol{\eta}] + \beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta}))$ with $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \boldsymbol{\sigma}_{\hat{\rho}})$ and $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\pi}, \lambda\boldsymbol{\sigma}_{\pi})$ have elements

$$(\sigma^*_{\hat{\rho}i})^{-1} = \frac{1}{2\beta}[h_i + \sqrt{h_i^2 + \frac{4\beta h_i}{(\mu_{i\hat{\rho}} - \mu_{i\pi})^2}}], \tag{7}$$

$$(\sigma^*_{\pi i})^{-1} = \frac{\lambda}{2\beta}[\sqrt{h_i^2 + \frac{4\beta h_i}{(\mu_{i\hat{\rho}} - \mu_{i\pi})^2}} - h_i], \tag{8}$$

where $\mathbf{H} \equiv \nabla^2\hat{\mathcal{L}}^{\ell_{\text{cat}}}_{X,Y}(f_{\boldsymbol{\theta}})$ captures the curvature at the the minimum. Then

$$\begin{aligned} \min_{\boldsymbol{\sigma}_{\hat{\rho}},\boldsymbol{\sigma}_{\pi}} C_\beta(X,Y;\hat{\rho},\pi) \geq \frac{1}{2}(&\sum_i a_i(\mu_{i\hat{\rho}} - \mu_{i\pi})^2 \\ &+ \beta\sum_i \ln(\frac{h_i + a_i}{a_i})), \end{aligned} \tag{9}$$

where $a_i \triangleq a_i(\beta, \mu_{i\hat{\rho}}, \mu_{i\pi}, h_i) = \frac{1}{2}[\sqrt{h_i^2 + \frac{4\beta h_i}{(\mu_{i\hat{\rho}} - \mu_{i\pi})^2}} - h_i]$.

Proof. The developed objective (5) is

$$C_\beta(X,Y;\hat{\rho},\pi)=\frac{1}{2}\mathrm{tr}(\mathbf{H}\boldsymbol{\Sigma}_{\hat{\rho}})+\frac{\beta}{2}(\mathrm{tr}(\frac{1}{\lambda}\boldsymbol{\Sigma}_\pi^{-1}\boldsymbol{\Sigma}_{\hat{\rho}})-k+\frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)^{\mathrm{T}}\boldsymbol{\Sigma}_\pi^{-1}(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)+\ln\left(\frac{\det\lambda\boldsymbol{\Sigma}_\pi}{\det\boldsymbol{\Sigma}_{\hat{\rho}}}\right)) \tag{10}$$

We substitute the precision matrix $\boldsymbol{\Lambda}_\pi=\boldsymbol{\Sigma}_\pi^{-1}$ and $\boldsymbol{\Sigma}_{\hat{\rho}}$ with the minimizer $\boldsymbol{\Sigma}_{\hat{\rho}}^*=\beta(\mathbf{H}+\frac{\beta}{\lambda}\boldsymbol{\Lambda}_\pi)^{-1}$ in (10), we obtain

$$\begin{aligned}
C_\beta(X,Y;\hat{\rho},\pi)|_{\boldsymbol{\Sigma}_{\hat{\rho}}=\boldsymbol{\Sigma}_{\hat{\rho}}^*}=&\frac{1}{2}\mathrm{tr}(\mathbf{H}\beta(\mathbf{H}+\frac{\beta}{\lambda}\boldsymbol{\Lambda}_\pi)^{-1})+\frac{\beta}{2}(\mathrm{tr}(\frac{1}{\lambda}\boldsymbol{\Lambda}_\pi\beta(\mathbf{H}+\frac{\beta}{\lambda}\boldsymbol{\Lambda}_\pi)^{-1})\\
&+\frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)^{\mathrm{T}}\boldsymbol{\Lambda}_\pi(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)-k+\ln\left(\frac{\det\lambda\boldsymbol{\Lambda}_\pi^{-1}}{\det\beta(\mathbf{H}+\frac{\beta}{\lambda}\boldsymbol{\Lambda}_\pi)^{-1}}\right))\\
=&\frac{\beta}{2}\mathrm{tr}(\mathbf{H}(\mathbf{H}+\frac{\beta}{\lambda}\boldsymbol{\Lambda}_\pi)^{-1})+\frac{\beta^2}{2\lambda}(\mathrm{tr}(\boldsymbol{\Lambda}_\pi(\mathbf{H}+\frac{\beta}{\lambda}\boldsymbol{\Lambda}_\pi)^{-1}))\\
&+\frac{\beta}{2}(+\frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)^{\mathrm{T}}\boldsymbol{\Lambda}_\pi(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)-k+\ln\left(\frac{\det\lambda\boldsymbol{\Lambda}_\pi^{-1}}{\det\beta(\mathbf{H}+\frac{\beta}{\lambda}\boldsymbol{\Lambda}_\pi)^{-1}}\right))\\
=&\frac{\beta}{2}(\mathrm{tr}((\mathbf{H}+\frac{\beta}{\lambda}\boldsymbol{\Lambda}_\pi)(\mathbf{H}+\frac{\beta}{\lambda}\boldsymbol{\Lambda}_\pi)^{-1})\\
&\frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)^{\mathrm{T}}\boldsymbol{\Lambda}_\pi(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)-k+\ln\left(\frac{\det\lambda\boldsymbol{\Lambda}_\pi^{-1}}{\det\beta(\mathbf{H}+\frac{\beta}{\lambda}\boldsymbol{\Lambda}_\pi)^{-1}}\right))\\
=&\frac{\beta}{2}[+\frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)^{\mathrm{T}}\boldsymbol{\Lambda}_\pi(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)+\ln\left(\frac{\det\lambda\boldsymbol{\Lambda}_\pi^{-1}}{\det\beta(\mathbf{H}+\frac{\beta}{\lambda}\boldsymbol{\Lambda}_\pi)^{-1}}\right)].
\end{aligned} \tag{11}$$

Substituting $\boldsymbol{\Lambda}_\pi=\mathrm{diag}(\Lambda_{1\pi},\Lambda_{2\pi},...,\Lambda_{k\pi})$ and $\mathbf{H}=\mathrm{diag}(h_1,h_2,...,h_k)$ in the above expression we get

$$C_\beta(X,Y;\hat{\rho},\pi)|_{\boldsymbol{\Sigma}_{\hat{\rho}}=\boldsymbol{\Sigma}_{\hat{\rho}}^*}=\frac{\beta}{2}(\frac{1}{\lambda}\sum_i\Lambda_{i\pi}(\mu_{i\hat{\rho}}-\mu_{i\pi})^2-\sum_i\ln(\frac{\Lambda_{i\pi}}{\lambda})+\sum_i\ln(\frac{h_i+\frac{\beta}{\lambda}\Lambda_{i\pi}}{\beta})) \tag{12}$$

The above expression is easy to optimize. We see that the sole stationary point exists at

$$\Lambda_{i\pi}^*=\frac{\lambda}{2\beta}[\sqrt{h_i^2+\frac{4\beta h_i}{(\mu_{i\hat{\rho}}-\mu_{i\pi})^2}}-h_i]. \tag{13}$$

We now need to calculate second derivatives so as to prove that the stationary point is a local optimum. We go back to the developed objective (10), and substitute $\boldsymbol{\Sigma}_{\hat{\rho}}=\mathrm{diag}(\boldsymbol{\sigma}_{\hat{\rho}})$ and $\boldsymbol{\Sigma}_\pi=\mathrm{diag}(\boldsymbol{\sigma}_\pi)$. For the diagonal approximation the objective turns into a sum of separable functions.

$$\begin{aligned}
C_\beta(X,Y;\hat{\rho},\pi)=&\sum_i\frac{h_i}{2}\sigma_{i\hat{\rho}}+\sum_i\frac{\beta}{2\lambda}\frac{\sigma_{i\hat{\rho}}}{\sigma_{i\pi}}-\sum_i\frac{\beta}{2}+\sum_i\frac{\beta(\mu_{i\hat{\rho}}-\mu_{i\pi})^2}{2\lambda}\frac{1}{\sigma_{i\pi}}\\
&+\frac{\beta}{2}[\sum_i\ln(\lambda\sigma_{i\pi})-\sum_i\ln(\sigma_{i\hat{\rho}})]\\
=&\sum_i A_i\sigma_{i\hat{\rho}}+\sum_i B_i\frac{\sigma_{i\hat{\rho}}}{\sigma_{i\pi}}-\sum_i\frac{\beta}{2}+\sum_i C_i\frac{1}{\sigma_{i\pi}}+D_i[\sum_i\ln(\lambda\sigma_{i\pi})-\sum_i\ln(\sigma_{i\hat{\rho}})]\\
=&\sum_i[A_i\sigma_{i\hat{\rho}}+B_i\frac{\sigma_{i\hat{\rho}}}{\sigma_{i\pi}}-\frac{\beta}{2}+C_i\frac{1}{\sigma_{i\pi}}+D_i(\ln(\lambda\sigma_{i\pi})-\ln(\sigma_{i\hat{\rho}}))]
\end{aligned} \tag{14}$$

where we have set $A_i = \frac{h_i}{2}$, $B_i = \frac{\beta}{2\lambda}$, $C_i = \frac{\beta(\mu_{i\hat{\rho}} - \mu_{i\pi})^2}{2\lambda}$, $D_i = \frac{\beta}{2}$.

We take the derivatives of one of these functions with respect to $\sigma_{i\hat{\rho}}, \sigma_{i\pi}$ and drop the indices i for clarity

$$\frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial \sigma_{\hat{\rho}}} = A + \frac{B}{\sigma_\pi} - \frac{D}{\sigma_{\hat{\rho}}}, \quad \frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial \sigma_\pi} = -\frac{B\sigma_{\hat{\rho}}}{\sigma_\pi^2} - \frac{C}{\sigma_\pi^2} + \frac{D}{\sigma_\pi} \tag{15}$$

and

$$\frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial^2 \sigma_{\hat{\rho}}} = \frac{D}{\sigma_{\hat{\rho}}^2}, \quad \frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial^2 \sigma_\pi} = 2(B\sigma_{\hat{\rho}} + C)\frac{1}{\sigma_\pi^3} - \frac{D}{\sigma_\pi^2} \tag{16}$$

$$\frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial \sigma_{\hat{\rho}} \partial \sigma_\pi} = -\frac{B}{\sigma_\pi^2}, \quad \frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial \sigma_\pi \partial \sigma_{\hat{\rho}}} = -\frac{B}{\sigma_\pi^2} \tag{17}$$

We need to check whether the Hessian matrix is PSD so that the stationary point we found is a local minimum and the function is convex. We do that by calculating whether all principal minors of the Hessian are positive.

$$\nabla^2 C_\beta(\sigma_{\hat{\rho}}, \sigma_\pi) = \begin{bmatrix} \frac{D}{\sigma_{\hat{\rho}}^2} & -\frac{B}{\sigma_\pi^2} \\ -\frac{B}{\sigma_\pi^2} & 2(B\sigma_{\hat{\rho}} + C)\frac{1}{\sigma_\pi^3} - \frac{D}{\sigma_\pi^2} \end{bmatrix} \tag{18}$$

We see easily that $\det(\frac{D}{\sigma_{\hat{\rho}}^2}) > 0$. While

$$\begin{aligned} \det(\nabla^2 C_\beta(\sigma_{\hat{\rho}}, \sigma_\pi)) &= \frac{D}{\sigma_{\hat{\rho}}^2}\left(2(B\sigma_{\hat{\rho}} + C)\frac{1}{\sigma_\pi^3} - \frac{D}{\sigma_\pi^2}\right) - \frac{B^2}{\sigma_\pi^4} \\ &= \frac{1}{\sigma_{\hat{\rho}}^2 \sigma_\pi^4}\left(2CD\sigma_\pi - (D\sigma_\pi - B\sigma_{\hat{\rho}})^2\right) \\ &= \left(\frac{1}{\sigma_{\hat{\rho}}^2 \sigma_\pi^4}\frac{\beta^2}{2}\right)\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda}\sigma_\pi - \frac{1}{2}(\sigma_\pi - \frac{\sigma_{\hat{\rho}}}{\lambda})^2\right) \end{aligned} \tag{19}$$

The determinant is not always positive and the function is not convex. We now check whether the sole stationary point is always a local minimum. We start by substituting $\sigma_{\hat{\rho}}^\star = \beta(h + \frac{\beta}{\lambda}\frac{1}{\sigma_\pi})^{-1}$ in the multiplicand of (19) as the multiplier is positive by definition

$$\begin{aligned} &\det(\nabla^2 C_\beta(\sigma_{\hat{\rho}}^\star, \sigma_\pi)) = \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^4}\frac{\beta^2}{2}\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda}\sigma_\pi - \frac{1}{2}(\sigma_\pi - \frac{\beta}{\lambda}(h + \frac{\beta}{\lambda}\frac{1}{\sigma_\pi})^{-1})^2\right) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^4}\frac{\beta^2}{2}\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda}\sigma_\pi - \frac{1}{2}(\sigma_\pi - \frac{\beta}{\lambda}(\frac{\sigma_\pi \lambda}{h\lambda\sigma_\pi + \beta}))^2\right) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^4}\frac{\beta^2}{2}\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda}\sigma_\pi - \frac{\sigma_\pi^2}{2}(1 - (\frac{\beta}{h\lambda\sigma_\pi + \beta}))^2\right) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^3}\frac{\beta^2}{2}\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda} - \frac{\sigma_\pi}{2}(\frac{h\lambda\sigma_\pi}{h\lambda\sigma_\pi + \beta})^2\right) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^3}\frac{\beta^2}{2}\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda} - \frac{\lambda^2 h^2 \sigma_\pi^3}{2(h\lambda\sigma_\pi + \beta)^2}\right) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^3 2\lambda(h\lambda\sigma_\pi + \beta)^2}(2(\mu_{\hat{\rho}} - \mu_\pi)^2(h\lambda\sigma_\pi + \beta)^2 - \lambda^3 h^2 \sigma_\pi^3) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2} 2\lambda(h\lambda\Lambda_\pi^{-1} + \beta)^2}(2\Lambda_\pi(\mu_{\hat{\rho}} - \mu_\pi)^2(h\lambda + \Lambda_\pi\beta)^2 - \lambda^3 h^2) \end{aligned} \tag{20}$$

Where we substituted $\sigma_\pi = \Lambda_\pi^{-1}$ as this will make the calculations easier. We now show a useful identity for $\Lambda_\pi^\star = \frac{\lambda}{2\beta}[\sqrt{h^2 + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}} - h]$

$$
\begin{aligned}
(\Lambda_\pi^\star)^2 &= \frac{\lambda^2}{4\beta^2}\left(h^2 + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2} - 2h\sqrt{h^2 + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}} + h^2\right) \\
&= \frac{\lambda^2}{4\beta^2}\left(2h\left(h - \sqrt{h^2 + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}}\right) + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}\right) \\
&= \frac{h\lambda}{\beta}\frac{\lambda}{2\beta}\left(\left(h - \sqrt{h^2 + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}}\right) + \frac{2\beta}{(\mu_{\hat{\rho}} - \mu_\pi)^2}\right) \\
&= \frac{h\lambda}{\beta}\left(\frac{\lambda}{(\mu_{\hat{\rho}} - \mu_\pi)^2} - \Lambda_\pi^\star\right)
\end{aligned}
\tag{21}
$$

We substitute $\Lambda_\pi = \Lambda_\pi^\star$ in (20) and again develop only the multiplicand

$$
\begin{aligned}
&\det(\nabla^2 C_\beta(\sigma_{\hat{\rho}}^\star, \sigma_\pi^\star)) = \frac{1}{\sigma_{\hat{\rho}}^{\star 2} 2\lambda(h\lambda\Lambda_\pi^{\star -1} + \beta)^2}(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h\lambda + \Lambda_\pi^\star\beta)^2 - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h\lambda + \Lambda_\pi^\star\beta)^2 - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + 2h\lambda\Lambda_\pi^\star\beta + (\Lambda_\pi^\star)^2\beta^2) - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + 2h\lambda\Lambda_\pi^\star\beta + \frac{h\lambda}{\beta}\left(\frac{\lambda}{(\mu_{\hat{\rho}} - \mu_\pi)^2} - \Lambda_\pi^\star\right)\beta^2) - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + h\lambda\Lambda_\pi^\star\beta + \frac{\beta\lambda^2 h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}) - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + \frac{\beta\lambda^2 h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}) + 2(\Lambda_\pi^\star)^2(\mu_{\hat{\rho}} - \mu_\pi)^2 h\lambda\beta - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + \frac{\beta\lambda^2 h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}) \\
&+ 2\frac{h\lambda}{\beta}\left(\frac{\lambda}{(\mu_{\hat{\rho}} - \mu_\pi)^2} - \Lambda_\pi^\star\right)(\mu_{\hat{\rho}} - \mu_\pi)^2 h\lambda\beta - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + \frac{\beta\lambda^2 h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}) + 2\lambda^3 h^2 - 2h^2\lambda^2(\mu_{\hat{\rho}} - \mu_\pi)^2\Lambda_\pi^\star - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + \frac{\beta\lambda^2 h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}) + \lambda^3 h^2 - 2h^2\lambda^2(\mu_{\hat{\rho}} - \mu_\pi)^2\Lambda_\pi^\star) \\
&= A(2\Lambda_\pi^\star\beta\lambda^2 h + \lambda^3 h^2) \\
&> 0
\end{aligned}
\tag{22}
$$

where we have set $A = \frac{1}{\sigma_{\hat{\rho}}^{\star 2} 2\lambda(h\lambda(\Lambda_\pi^\star)^{-1} + \beta)^2} > 0$. We have used (21) in lines 4 and 7.

Indeed the stationary point is a local minimum. We now show that there are no other local minima at the boundaries of the domain. From (14) we see that we only need to evaluate expressions of the form $f(\sigma_{\hat{\rho}}) = \sigma_{\hat{\rho}} - \ln(\sigma_{\hat{\rho}})$ and $g(\sigma_\pi) = \frac{1}{\sigma_{\hat{\rho}}} + \ln(\sigma_{\hat{\rho}})$. By application of L'Hôpital's rule it's easy to show that

$$
\begin{aligned}
\lim_{\substack{\sigma_{\hat{\rho}} \to 0 \\ \sigma_\pi = \text{ct}}} C_\beta(\sigma_{\hat{\rho}}, \sigma_\pi) &= \lim_{\substack{\sigma_{\hat{\rho}} \to +\infty \\ \sigma_\pi = \text{ct}}} C_\beta(\sigma_{\hat{\rho}}, \sigma_\pi) \\
&= \lim_{\substack{\sigma_{\hat{\rho}} = \text{ct} \\ \sigma_\pi \to 0}} C_\beta(\sigma_{\hat{\rho}}, \sigma_\pi) = \lim_{\substack{\sigma_{\hat{\rho}} = \text{ct} \\ \sigma_\pi \to +\infty}} C_\beta(\sigma_{\hat{\rho}}, \sigma_\pi) = +\infty
\end{aligned}
\tag{23}
$$

□

D. Proof of Lemma 5.1

Preliminaries We remind that a neural network transforms it's inputs $\mathbf{a}_0 = \boldsymbol{x}$ to an output $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{a}_l$ through a series of l layers, each of which consists of a bank of units/neurons. The computation performed by each layer $i \in \{1, ..., l\}$ is given as

$$\begin{aligned}\boldsymbol{s}_i &= \mathbf{W}_i \mathbf{a}_{i-1}, \\ \mathbf{a}_i &= \phi_i(\boldsymbol{s}_i).\end{aligned}$$

We also denote the vectorization of the weights as $\boldsymbol{\theta} = [\mathrm{vec}(\mathbf{W}_0^{0,:})\mathrm{vec}(\mathbf{W}_0^{1,:}) \cdots \mathrm{vec}(\mathbf{W}_0^{r,:})]$, where $\mathrm{vec}(\mathbf{W}_i^{j,:})$ are the weights corresponding to layer i and neuron j. We assume trained vectorized weights $\boldsymbol{\mu}_{\hat{\rho}i}$ and trained weights in matrix form $\mathbf{W}_{\hat{\rho}i}$ for layer i. We will be adding bounded perturbations to the weights of each layer i so that $||\mathbf{W}_i - \mathbf{W}_{\hat{\rho}i}||_F \leq C$. We will want to quantify the effect of these perturbations on the latent representations of the network.

We then define $\mathbf{A}_i = [\mathbf{a}_i^0, \cdots, \mathbf{a}_i^n]$, where $\mathbf{a}_i^j$ is the unperturbed latent representation of sample j at layer i, where $\mathbf{A}_i$ is produced by the operation $\mathbf{A}_i = \mathrm{rect}(\mathbf{W}_{\hat{\rho}i}\mathbf{A}_{i-1})$. We perturb only layer i and define $\hat{\mathbf{A}}_i$, as the representations resulting from the new perturbed matrix $\mathbf{W}_i$, $\hat{\mathbf{A}}_i = \mathrm{rect}(\mathbf{W}_i\mathbf{A}_{i-1})$. We then define $\tilde{\mathbf{A}}_i$ as the representations at layer i with accumulated error from layers $\leq i$. Similarly we can define the same quantities for the pre-activations $\boldsymbol{s}_i^j$, we denote the corresponding matrices as $\hat{\boldsymbol{S}}_i$ and $\tilde{\boldsymbol{S}}_i$.

We can then define the layerwise mean square error from perturbing only layer i

$$\begin{aligned}\hat{e}_i^2 &= (1/n)||\mathbf{A}_i - \hat{\mathbf{A}}_i||_F^2, \\ \hat{E}_i^2 &= (1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2,\end{aligned}$$

as well as the accumulated mean square error

$$\begin{aligned}\tilde{e}_i^2 &= (1/n)||\mathbf{A}_i - \tilde{\mathbf{A}}_i||_F^2, \\ \tilde{E}_i^2 &= (1/n)||\boldsymbol{S}_i - \tilde{\boldsymbol{S}}_i||_F^2,\end{aligned}$$

where the true representations are considered as constants. We make a simplifying assumption, assuming that the mean square error of our *trained* classifier is 0. In this case we can set $\hat{\mathcal{L}}_{X,Y}^{\ell_{\text{mse}}}(f_{\boldsymbol{\theta}}) \equiv \tilde{e}_l^2 = (1/n)||\mathbf{A}_l - \tilde{\mathbf{A}}_l||_F^2$, as $\mathbf{A}_l$ now correspond to the ground truth vectors. We can easily extend to the non-zero error case using the triangle inequality.

These errors are difficult to analyze theoretically. As such we will make the useful assumption that they are well approximated by a *quadratic*, which will make the analysis tractable. This assumption is quite strong and we do not claim that the approximation is tight. Furthermore Figure 3 of the main text does not directly apply in this setting; we will be dealing with the mean-square error instead of the categorical cross-entropy and we will be analyzing layerwise errors instead of the error at the output. At the same time our aim is only to derive a useful *surrogate* objective. The empirical results in Section 5 provide evidence that the surrogate we propose is indeed useful in providing tighter bounds.

Useful Lemmata We prove the following Lemma which will be useful later. We first show that the mean square error at the output of a deep neural network can be decomposed as a sum of mean square errors for intermediate representations.

Lemma 0.2. *Assuming layerwise perturbations that are bounded by a constant $||\mathbf{W}_i - \mathbf{W}_{\hat{\rho}i}||_F \leq C$, the accumulated mean square error $\tilde{e}_l^2$ at layer l can be bounded as*

$$(1/n)||\mathbf{A}_l - \tilde{\mathbf{A}}_l||_F^2 \leq \sum_{i=0}^{l} c_i (1/n)||\mathbf{A}_i - \hat{\mathbf{A}}_i||_F^2 + \mathcal{O}(c^l) \tag{24}$$

where $\forall i < l,\ c_i = \prod_{k=i+1}^{l} ||\mathbf{W}_k||_F^2$, $c_l = 1$ and c is some constant.

Proof. We denote $\hat{a}_{i+1}$ a single element of $\hat{\mathbf{a}}_{i+1}$ and $\mathbf{w}_i^{\mathrm{T}}$ the corresponding row of $\mathbf{W}_i$ where we drop the indices for

individual samples and neurons for clarity. One can easily see through the properties of the rectifier function that

$$\begin{aligned}\hat{a}_{i+1} &= \mathrm{rect}(\mathbf{w}_{i+1}^{\mathrm{T}}\tilde{\mathbf{a}}_i + \mathbf{w}_{i+1}^{\mathrm{T}}(\mathbf{a}_i - \tilde{\mathbf{a}}_i)) \\ &\leq \tilde{a}_{i+1} + \mathrm{rect}(\mathbf{w}_{i+1}^{\mathrm{T}}(\mathbf{a}_i - \tilde{\mathbf{a}}_i)) \\ &\leq \tilde{a}_{i+1} + |\mathbf{w}_{i+1}^{\mathrm{T}}(\mathbf{a}_i - \tilde{\mathbf{a}}_i)|\end{aligned} \tag{25}$$

Similarly we can obtain $\tilde{a}_{i+1} \leq \hat{a}_{i+1} + |\mathbf{w}_{i+1}^{\mathrm{T}}(\mathbf{a}_i - \tilde{\mathbf{a}}_i)|$ and therefore we can write

$$|\tilde{a}_{i+1} - \hat{a}_{i+1}| \leq |\mathbf{w}_{i+1}^{\mathrm{T}}(\mathbf{a}_i - \tilde{\mathbf{a}}_i)|.$$

In matrix notation this becomes

$$||\tilde{\mathbf{A}}_{i+1} - \hat{\mathbf{A}}_{i+1}||_F \leq ||\mathbf{W}_{i+1}(\mathbf{A}_i - \tilde{\mathbf{A}}_i)||_F \leq ||\mathbf{W}_{i+1}||_F||\tilde{\mathbf{A}}_i - \mathbf{A}_i||_F$$

By the triangle inequality we can then write

$$\begin{aligned}\tilde{e}_{i+1} = (1/\sqrt{n})||\tilde{\mathbf{A}}_{i+1} - \mathbf{A}_{i+1}||_F &\leq (1/\sqrt{n})||\tilde{\mathbf{A}}_{i+1} - \hat{\mathbf{A}}_{i+1}||_F + (1/\sqrt{n})||\hat{\mathbf{A}}_{i+1} - \mathbf{A}_{i+1}||_F \\ &\leq (1/\sqrt{n})||\mathbf{W}_{i+1}||_F||\tilde{\mathbf{A}}_i - \mathbf{A}_i||_F + (1/\sqrt{n})||\hat{\mathbf{A}}_{i+1} - \mathbf{A}_{i+1}||_F \\ &\leq \sum_{t=0}^{i}(\prod_{k=t+1}^{i+1}||\mathbf{W}_k||_F||\hat{\mathbf{A}}_t - \mathbf{A}_t||_F) + (1/\sqrt{n})||\hat{\mathbf{A}}_{i+1} - \mathbf{A}_{i+1}||_F \\ &= \sum_{t=0}^{i}(\prod_{k=t+1}^{i+1}||\mathbf{W}_k||_F\hat{e}_t) + \hat{e}_{i+1}\end{aligned} \tag{26}$$

If $||\mathbf{W}_i - \mathbf{W}_{\hat{\rho}i}||_F \leq C$, then the errors $\hat{e}_t = ||\mathbf{A}_t - \hat{\mathbf{A}}_t||_F$ and also all terms $\prod_{k=t+1}^{i+1}||\mathbf{W}_k||_F\hat{e}_t$ are bounded. We raise both sides to the power of 2. We get the desired terms as well as terms of the form $\prod_{k=a+1}^{i+1}||\mathbf{W}_k||_F\prod_{k=b+1}^{i+1}||\mathbf{W}_k||_F\hat{e}_a\hat{e}_b$ assuming that $||\mathbf{W}_i||_F \leq \sqrt{c}$ we see that these are of the order $\mathcal{O}((\sqrt{c})^{2l}) = \mathcal{O}(c^l)$ and we get the desired result. □

In the following it will be useful to deal with the preactivations $\boldsymbol{s}_i^j$ instead of the representations $\mathbf{a}_i^j$ so as to avoid taking derivatives of the rectifier non-linearity. We will then find useful the following simple Lemma.

Lemma 0.3. *Given the true preactivations $\boldsymbol{S}_i$ and representations $\mathbf{A}_i$, as well as the perturbed $\hat{\boldsymbol{S}}_i$ and $\hat{\mathbf{A}}_i$ for layer i the following holds*

$$(1/n)||\mathbf{A}_i - \hat{\mathbf{A}}_i||_F^2 \leq (1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2. \tag{27}$$

Proof. We assume $(\mathrm{rect}(x) - \mathrm{rect}(y))^2 \leq (x-y)^2$ and check that it holds for different signs of x, y. □

We will now approximate the precativation error for each layer using a second order Taylor expansion. We prove the following.

Lemma 0.4. *We apply a Taylor expansion of the layerwise preactivation error $\hat{E}_i^2(\boldsymbol{\theta})$ of layer i, around a point $\boldsymbol{\mu}$. Given j neurons and n training samples, $\hat{E}_i^2(\boldsymbol{\theta})$ can be approximated as*

$$\hat{E}_i^2(\boldsymbol{\theta}) = (1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2 = \sum_j(\boldsymbol{\theta}_{ij} - \boldsymbol{\mu}_{ij})^T\mathbf{H}_i(\boldsymbol{\theta}_{ij} - \boldsymbol{\mu}_{ij}) + \mathcal{O}(||\boldsymbol{\theta}_i - \boldsymbol{\mu}_i||^3). \tag{28}$$

where $\mathbf{H}_i = (1/n)\sum_{k=0}^{n}\mathbf{a}_{i-1}^k{\mathbf{a}_{i-1}^k}^T$.

Proof. It will be easier to work with the vectorized weights per neuron $\boldsymbol{\theta}_{ij}$ directly. We note that the unperturbed

representations $\boldsymbol{S}_i$ are considered as constants, and get

$$\begin{aligned}
\frac{\partial \hat{E}_i^2}{\partial \boldsymbol{\theta}_{ij}} &= \frac{\partial}{\partial \boldsymbol{\theta}_{ij}}(1/n)||\hat{\boldsymbol{S}}_i - \boldsymbol{S}_i||_F^2 \\
&= \frac{\partial}{\partial \boldsymbol{\theta}_{ij}}(1/n)||\mathbf{W}_i\mathbf{A}_{i-1} - \boldsymbol{S}_i||_F^2 \\
&= \frac{\partial}{\partial \boldsymbol{\theta}_{ij}}(1/n)\sum_{k=0}^{n}||\mathbf{W}_i\mathbf{a}_{i-1}^k - \boldsymbol{s}_i^k||_2^2 \\
&= \frac{\partial}{\partial \boldsymbol{\theta}_{ij}}(1/n)\sum_{k=0}^{n}\sum_{t=0}^{r}||\boldsymbol{\theta}_{it}^T\mathbf{a}_{i-1}^k - s_{it}^k||_2^2 \\
&= \frac{1}{n}\sum_{k=0}^{n}\sum_{t=0}^{r}\frac{\partial}{\partial \boldsymbol{\theta}_{ij}}||\boldsymbol{\theta}_{it}^T\mathbf{a}_{i-1}^k - s_{it}^k||_2^2 = \frac{2}{n}\sum_{k=0}^{n}(\boldsymbol{\theta}_{ij}^T\mathbf{a}_{i-1}^k - s_{ij}^k){\mathbf{a}_{i-1}^k}^T
\end{aligned} \tag{29}$$

where in the third line we expand with respect to the samples and in the fourth line we expand with respect to each neuron. Then we can calculate the second order derivatives.

$$\frac{\partial^2 \hat{E}_i^2}{\partial^2 \boldsymbol{\theta}_{ij}} = \frac{\partial}{\partial \boldsymbol{\theta}_{ij}}\frac{2}{n}\sum_{k=0}^{n}(\boldsymbol{\theta}_{ij}^T\mathbf{a}_{i-1}^k - s_{ij}^k){\mathbf{a}_{i-1}^k}^T = \frac{2}{n}\sum_{k=0}^{n}\mathbf{a}_{i-1}^k{\mathbf{a}_{i-1}^k}^T. \tag{30}$$

From the above, it is clear that the Hessian is block diagonal, with identical blocks for each neuron j. We can the approximate the layerwise error $\hat{e}_i^2$ using a second order Taylor expansion around a point $\boldsymbol{\mu}$ as

$$\begin{aligned}
\hat{E}_i^2 &= \frac{\partial \hat{E}_i^2}{\partial \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i - \boldsymbol{\mu}_i)^T + \frac{1}{2}(\boldsymbol{\theta}_i - \boldsymbol{\mu}_i)^T\frac{\partial^2 \hat{E}_i^2}{\partial^2 \boldsymbol{\theta}_i}(\boldsymbol{\theta}_i - \boldsymbol{\mu}_i) + \mathcal{O}(||\boldsymbol{\theta}_i - \boldsymbol{\mu}_i||^3) \\
&= \sum_j[(\boldsymbol{\theta}_{ij} - \boldsymbol{\mu}_{ij})^T\sum_{k=0}^{n}\frac{1}{n}\mathbf{a}_{i-1}^k{\mathbf{a}_{i-1}^k}^T(\boldsymbol{\theta}_{ij} - \boldsymbol{\mu}_{ij})] + \mathcal{O}(||\boldsymbol{\theta}_i - \boldsymbol{\mu}_i||^3)
\end{aligned} \tag{31}$$

where we assume that the derivatives with respect to the layer weights of order other than two are negligible. This is a strong but useful assumption to make, and one that will make the analysis tractable. □

We are now ready to prove our main lemma.

Lemma 5.1. The differentiable surrogate objective

$$\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}\hat{\mathcal{L}}_{X,Y}^{\ell_{\text{mse}}}(f_{\boldsymbol{\theta}}) + \frac{1}{\beta n}(\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\mathcal{N}(\boldsymbol{\mu}_\pi, \lambda\mathbf{I})) + \ln\frac{1}{\delta}) \tag{32}$$

, assuming that the layerwise derivatives of order other than 2 are negligible, has the following upper bound

$$\begin{aligned}
&\sum_{i,j}[\mathbf{E}_{\boldsymbol{\eta}_{ij}\sim\hat{\rho}'_{ij}(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}_{ij}^T\mathbf{H}_i\boldsymbol{\eta}_{ij}] + \frac{1}{\beta n}\mathrm{KL}(\hat{\rho}_{ij}(\boldsymbol{\theta})||\pi_{ij}(\boldsymbol{\theta})] \\
&+ \mathcal{O}(c^l)
\end{aligned} \tag{33}$$

where $\hat{\rho}_{ij}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}ij}, \boldsymbol{\Sigma}_{\hat{\rho}ij})$, $\pi_{ij}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\pi ij}, \lambda\mathbf{I})$, $\mathbf{H}_i = (2/n)\sum_{k=0}^{n}\mathbf{a}_{i-1}^k{\mathbf{a}_{i-1}^k}^T$, are neuronwise posteriors, priors and Hessians.

Proof. We assume that the prior $\pi(\boldsymbol{\theta})$ and posterior $\hat{\rho}(\boldsymbol{\theta})$ are block diagonal, with blocks corresponding to weights in each

neuron.

$$
\begin{aligned}
\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[\hat{\mathcal{L}}^{\ell_{\text{mse}}}_{X,Y}(f_{\boldsymbol{\theta}})] &\leq \mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[\sum_{i=0}^{l} c_i(1/n)||\mathbf{A}_i - \hat{\mathbf{A}}_i||_F^2 + \mathcal{O}(c^l)] \\
&\leq \mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[\sum_{i=0}^{l} c_i(1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2 + \mathcal{O}(c^l)] \\
&= \sum_{i=0}^{l} \mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[c_i]\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[(1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2] + \mathcal{O}(c^l)] \\
&\leq \sum_{i=0}^{l} c^* \mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[(1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2] + \mathcal{O}(c^l) \\
&= \sum_{i=0}^{l} c^* \mathbf{E}_{\boldsymbol{\eta}_{ij}\sim\hat{\rho}'_{ij}(\boldsymbol{\theta})}[\sum_{j} \boldsymbol{\eta}_{ij}^T \mathbf{H}_i \boldsymbol{\eta}_{ij}] + \mathcal{O}(c^l) \\
&= \sum_{i,j} c^* \mathbf{E}_{\boldsymbol{\eta}_{ij}\sim\hat{\rho}'_{ij}(\boldsymbol{\theta})}[\boldsymbol{\eta}_{ij}^T \mathbf{H}_i \boldsymbol{\eta}_{ij}] + \mathcal{O}(c^l).
\end{aligned}
\tag{34}
$$

In line 3 we used the fact that the constant c_i for layer i depends only on layers $k \geq i+1$, thus the two random variables are independent and the expectation operator is multiplicative. In line 4 we assume that the terms $c_i = \prod_{k=i+1}^{l} ||\mathbf{W}_k||_F^2$ are upper bounded by the constant c^*. This is reasonable as in practice we will be adding Gaussian noise with bounded variance to the layer weights. In line 5 we approximate the error $\hat{E}_i^2 = (1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2$ using (28) at point $\mu_{\hat{\rho}}$ which is the mean of the posterior $\hat{\rho}(\boldsymbol{\theta})$, then we use that $\hat{\rho}'(\boldsymbol{\theta})$ is a centered version of $\hat{\rho}(\boldsymbol{\theta})$. We finally assume that the term $\mathcal{O}(c^l)$ dominates the remainders from the Taylor expansion.

We then absorb the constant c^* in the hyperparameter β. By noting that the KL divergence of block-diagonal Gaussians can be decomposed as $\text{KL}(\mathcal{N}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta})) = \sum_{ij} \text{KL}(\mathcal{N}(\hat{\rho}_{ij}(\boldsymbol{\theta})||\pi_{ij}(\boldsymbol{\theta}))$ we get the desired result. □

Importantly we don't require that the deep neural network was trained using the mean square error. Rather we can optimize (33) for any network and assume that it's representations remain close based on the mean square error. Our experiments however show that optimizing (33) is also a good surrogate for keeping the 01-error small.

E. Experimental Setup

Experiments for Variational Inference were performed on NVIDIA Tesla K40c GPU. All other experiments were performed on an NVIDIA GEFORCE GTX 1080 GPU. The libraries used were Tensoflow 1.15.0 (Abadi et al., 2015), Keras 2.2.4 (Chollet et al., 2015) and Tensorflow-Probability 0.8.0 (Dillon et al., 2017).

When training the original deterministic classifiers, for the MNIST architectures we used the Keras implementation SGD with a learning rate of 0.01, momentum value of 0.9 and exponential decay with decay factor 0.001. For CIFAR architectures we used the Keras implementation of Adam with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, decay value of 0.00005 and the default value for the epsilon parameter. We used the softmax activation as well as the categorical cross-entropy in both cases. MNIST architectures were trained for 10 epochs while CIFAR architectures where trained for 200 epochs, which was sufficient for the training loss to stop decreasing.

When optimizing the posterior distributions centered at the deterministic classifier we used a grid search over β and/or λ where appropriate, with limits specified in the following tables. The computational time reported refers to the total time required to compute the plots in the main text for each setup, including computing the posterior and/or prior distributions as well as sampling m number of samples for estimating the expected empirical risk of the stochastic classifier.

MNIST. We report the following values for the MNIST experiments.

Experiment	β	λ	Time
MNIST Is@0	-	[0.031,0.3]	14h
MNIST Is@Init	-	[0.031,0.3]	14h
MNIST VI	[1,5]	[0.03,0.1]	11h
MNIST Post	[0.001,0.07]	[0.00005,0.01]	33h
MNIST Post+Prior	[0.000007,0.001]	-	10h
MNIST sK-FAC	[0.001,0.02]	[0.001,0.1]	33h

The β and λ ranges are identical for MNIST10, MNIST5, MNIST2 while computation times are of the same order of magnitude.

CIFAR. We report the following values for the CIFAR experiments.

Experiment	β	λ	Time
CIFAR Is@0	-	[0.031,0.3]	15h
CIFAR Is@Init	-	[0.031,0.3]	15h
CIFAR VI	[1,2]	[0.1,0.3]	10h
CIFAR Post	[0.001,0.1]	[0.001,0.1]	32h
CIFAR Post+Prior	[0.0001,0.001]	-	11h
CIFAR sK-FAC	-	-	-

The β and λ ranges are identical for CIFAR10, CIFAR5, CIFAR2 while computation times are of the same order of magnitude.

For the Variational Inference experiments we used the Adam (Kingma & Ba, 2014) optimizer with a learning rate of $1e-1$ for 5 epochs of training. For efficient inference we used the Tensorflow-Probability (Dillon et al., 2017) implementation of the Flipout (Wen et al., 2018) estimator.

F. Notes on PAC-Bayes

We note here some important differences between the PAC-Bayesian setting and the standard Bayesian treatment of deep neural networks, as there are some important overlaps in the terms used.

First, while PAC-Bayes refers to a "posterior" $\hat{\rho}$ this distribution is not required to be a posterior in the Bayesian sense. On the contrary it can be chosen to be *any* distribution. As such we are free to model $\hat{\rho}$ using different distributions centered on the deterministic neural networks, decoupled from how we trained the original deterministic network. In particular in Section 5 we can minimize the mean square error surrogate from Lemma 5.1. even though the deterministic networks are trained using the categorical cross-entropy loss.

Second, as noted in the main text the prior π in PAC-Bayes has to be independent of the training set but can depend on the data distribution.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/. Software available from tensorflow.org.

Catoni, O. Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007.

Chollet, F. et al. Keras. https://keras.io, 2015.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Langford, J. and Caruana, R. (not) bounding the true error. In *Advances in Neural Information Processing Systems*, pp. 809–816, 2002.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Wen, Y., Vicol, P., Ba, J., Tran, D., and Grosse, R. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018.