Supplementary Material for Information-Theoretic Local Minima Characterization and Regularization

A. Proof of Equation 1 in Section 4

Let us first review the Equation 1 in Section 4:

\[
\mathcal{I}_S(w_0) = \nabla_w^2 \mathcal{L}(S, w_0) = \mathbb{E}_{(x,c_x) \sim S} \left[ \nabla_w \ln p_{w_0}(c_x) \nabla_w \ln p_{w_0}(c_x)^T \right]
\]

To prove this equation, it suffices to prove the following equality:

\[
-\nabla_w^2 \ell_{\mathcal{S}}(w) = \sum_{(x,y) \in \mathcal{S}} K y_i \left[ \nabla_w \ln p(c_x = i|x; w) \nabla_w \ln p(c_x = i|x; w)^T \right]
\]

For convenience, we change the notation of the local minimum from \( w_0 \) to \( w \) and further denote \( p(c_x = i|x; w) \) as \( p^e_w(i) \).

Since \( -\nabla_w^2 \ell_{\mathcal{S}}(w) = -\sum_{(x,y) \in \mathcal{S}} \sum_{i=1}^{K} y_i \nabla_w^2 \ln p^e_w(i) \), for each \( (x, y) \in \mathcal{S} \) and \( i \in \{1, 2, ..., K\} \), we have:

\[
[\nabla_w^2 \ln p^e_w(i)]_{j,k} = \frac{\partial^2}{\partial w_j \partial w_k} \ln p^e_w(i) = \frac{\partial}{\partial w_j} \left( \frac{\partial}{\partial w_k} p^e_w(i) \right) = \frac{p^e_w(i) \frac{\partial^2}{\partial w_j \partial w_k} p^e_w(i)}{p^e_w(i)^2} - \frac{\partial}{\partial w_j} \frac{p^e_w(i)}{p^e_w(i)} \frac{\partial}{\partial w_k} \frac{p^e_w(i)}{p^e_w(i)} = \frac{\partial^2}{\partial w_j \partial w_k} p^e_w(i) - \frac{\partial}{\partial w_j} \ln p^e_w(i) \frac{\partial}{\partial w_k} \ln p^e_w(i) \tag{a}
\]

Since \( w_0 \) is a local minimum of full training accuracy, as described in Section 4, and \( y_i = p^e_w(i) \) for \( i \in \{1, 2, ..., K\} \), when taking the double summation, the first term in Equation a becomes:

\[
\sum_{(x,y) \in \mathcal{S}} \sum_{i=1}^{K} \frac{\partial^2}{\partial w_j \partial w_k} p^e_w(i) = \frac{\partial^2}{\partial w_j \partial w_k} \sum_{(x,y) \in \mathcal{S}} \sum_{i=1}^{K} p^e_w(i) = \frac{\partial^2}{\partial w_j \partial w_k} N = 0
\]

Then it follows that:

\[
[\nabla_w^2 \ell_{\mathcal{S}}(w)]_{j,k} = -\sum_{(x,y) \in \mathcal{S}} \sum_{i=1}^{K} y_i \left[ \nabla_w \ln p^e_w(i) \nabla_w \ln p^e_w(i)^T \right]_{j,k}
\]

B. Proof of the Generalization Bound in Section 5.2

Remind that in Section 5.2 we pick a uniform prior \( \mathcal{P} \) over \( w \in \mathcal{M}(w_0) \) and pick the posterior \( \mathcal{Q} \) of density \( q(w) \propto e^{-\mathcal{L}_0 - \mathcal{L}(\mathcal{S}, w)} \) with \( \mathcal{L}_0 \triangleq \mathcal{L}(\mathcal{S}, w_0) \). Then we have the upper bound of the expected generalization loss \( \mathbb{E}_{w \sim \mathcal{Q}}[\mathcal{L}(\mathcal{D}, w)] \) in terms of the expected training loss \( \mathbb{E}_{w \sim \mathcal{Q}}[\mathcal{L}(\mathcal{S}, w)] \) and \( \gamma(w_0) \).

**Theorem A.** Given \( |\mathcal{S}| = N \), \( \mathcal{D} \), \( \mathcal{L}(\mathcal{S}, w) \) and \( \mathcal{L}(\mathcal{D}, w) \) described in Section 3, a local minimum \( w_0 \), the volume \( V \) of \( \mathcal{M}(w_0) \) sufficiently small, the Assumption 1 & 2 satisfied, and \( \mathcal{P}, \mathcal{Q} \) defined above, for any \( \delta \in (0, 1) \), we have with probability at least \( 1 - \delta \) that:

\[
\mathbb{E}_{w \sim \mathcal{Q}}[\mathcal{L}(\mathcal{D}, w)] \leq \mathbb{E}_{w \sim \mathcal{Q}}[\mathcal{L}(\mathcal{S}, w)] + 2 \sqrt{\frac{2\mathcal{L}_0 + 2A + \ln \frac{2N}{\delta}}{N - 1}} \quad \text{where} \quad A = \frac{\sqrt{W} \pi e^{\gamma(w_0)/W}}{4\pi e} \tag{B.1}
\]
As defined in Section 5.2, given the model class \( C \),

**Theorem B.** For any data distribution \( D \) and a loss function \( \mathcal{L}(\cdot, \cdot) \in [0, 1] \), let \( \mathcal{L}(D, w) \) and \( \mathcal{L}(S, w) \) be the expected loss and training loss respectively for the model parameterized by \( w \), with the training set \(|S| = N\). For any prior distribution \( P_{\text{prior}} \) and posterior \( Q \) over \( C \) (not necessarily Bayesian posterior), and for any \( \delta \in (0, 1] \), we have with probability at least \( 1 - \delta \) that:

\[
\mathbb{E}_{w \sim P_{\text{prior}}} \left[ \mathcal{L}(D, w) \right] \leq \mathbb{E}_{w \sim Q} \left[ \mathcal{L}(S, w) \right] + 2 \sqrt{\frac{2D_{\text{KL}}(Q||P) + \ln \frac{2N}{\delta}}{N - 1}}
\]

**PAC-Bayes (McAllester)** For a data distribution \( D \) and a loss \( \mathcal{L}(\cdot, \cdot) \in [0, 1] \), let \( \mathcal{L}(D, w) \) and \( \mathcal{L}(S, w) \) be the expected loss and the training loss; the training set \(|S| = N\) is sampled from \( D \). Given arbitrary prior \( P \) and posterior \( Q \) (no need to be Bayesian posterior) supported on a model class \( C \), and for any \( \delta > 0 \), we have, with probability at least \( 1 - \delta \), that:

\[
\mathbb{E}_{w \sim Q} \left[ \mathcal{L}(D, w) \right] \leq \mathbb{E}_{w \sim Q} \left[ \mathcal{L}(S, w) \right] + 2 \sqrt{\frac{2D_{\text{KL}}(Q||P) + \ln \frac{2N}{\delta}}{N - 1}}
\]

As \( e^{\mathbb{N}(w_0)} = |\mathcal{I}_S(w_0)| \), we can rewrite the generalization bound we want to prove above as:

\[
\mathbb{E}_{w \sim Q} \left[ \mathcal{L}(D, w) \right] \leq \mathbb{E}_{w \sim Q} \left[ \mathcal{L}(S, w) \right] + 2 \sqrt{\frac{W \cdot V^{2/W} \pi^{1/W} |\mathcal{I}_S(w_0)|^{1/W} + 4\pi e \mathcal{L}_0 + 2\pi e \ln \frac{2N}{\delta}}{2\pi e(N - 1)}}
\]

As defined in Section 5.2, given the model class \( \mathcal{M}(w_0) \), whose volume is \( V \), for the neural network \( f_w \), the uniform prior \( P \) attains the probability density function \( p(w) = \frac{1}{V} \) for any \( w \in \mathcal{M}(w_0) \) and the posterior \( Q \) has density \( q(w) \propto e^{-\mathcal{L}(S(w), w) - \mathcal{L}_0} \). Based on Assumption 2 in Section 5.2 and the observed Fisher information \( \mathcal{I}_S(w_0) \), especially the Equation 2 derived in Section 4, we have:

\[
\mathcal{L}(S, w) = \mathcal{L}_0 + \frac{1}{2}(w - w_0)^T \mathcal{I}_S(w_0)(w - w_0) \quad \forall w \in \mathcal{M}(w_0)
\]

Denote \( \Sigma = [\mathcal{I}_S(w_0)]^{-1} = [\nabla_w^2 \mathcal{L}(S, w_0)]^{-1} \). Then \( Q \) is a truncated multivariate Gaussian distribution whose density function \( q \) is:

\[
q(w; w_0, \Sigma) = \frac{\exp\left\{ -\frac{1}{2}(w - w_0)^T \Sigma^{-1}(w - w_0) \right\}}{\int_{\mathcal{M}(w_0)} \exp\left\{ -\frac{1}{2}(w - w_0)^T \Sigma^{-1}(w - w_0) \right\} \, dw}
\]

Denote the denominator of Equation b as \( Z \) and define:

\[
g(w; w_0, \Sigma) \triangleq -\frac{1}{2}(w - w_0)^T \Sigma^{-1}(w - w_0) \leq 0
\]

Then \( q \) can also be written as:

\[
q(w; w_0, \Sigma) = \frac{\exp\{g(w; w_0, \Sigma)\}}{Z}
\]

In order to derive a generalization bound in the form of the PAC-Bayes Theorem, it suffices to prove an upper bound of the
KL divergence term:

\[ D_{\text{KL}}(Q||P) = \mathbb{E}_{w \sim Q} \ln \frac{q(w)}{p(w)} \]

\[ = - \mathbb{E}_{w \sim Q} \ln \frac{1}{Z} + \mathbb{E}_{w \sim Q} \ln q(w) \]

\[ = \ln V + \mathbb{E}_{w \sim Q} g(w; w_0, \Sigma) + \ln \frac{1}{Z} \]

\[ \leq \ln V + \mathbb{E}_{w \sim Q} 0 - \ln \left( \int_{\mathcal{M}(w_0)} \exp\{g(w; w_0, \Sigma)\} \, dw \right) \]

\[ \leq \ln V - \ln \left( \int_{\mathcal{M}(w_0)} \exp\{- \max_{w \in \mathcal{M}(w_0)} \mathcal{L}(S, w)\} \, dw \right) \]

\[ = \ln V - \ln \left( V \cdot \exp\{- \max_{w \in \mathcal{M}(w_0)} \mathcal{L}(S, w)\} \right) \]

\[ = \ln V - \ln V + h = h \]

where \( h \) is the height of \( \mathcal{M}(w_0) \) defined in Section 5.1. For convenience, we shift down \( \mathcal{L}(S, w) \) by \( \mathcal{L}_0(w) \) and denote the shifted training loss \( \mathcal{L}_0(w) = \Delta \mathcal{L}(S, w) - \mathcal{L}_0 \) so that \( \mathcal{L}_0(w_0) = 0 \). Then

\[ \mathcal{L}_0(w) = \frac{1}{2} (w - w_0)^T \Sigma^{-1} (w - w_0) \forall w \in \mathcal{M}(w_0) \]

Furthermore, the following two sets are equivalent

\[ \{ w \in \mathbb{R}^W : \mathcal{L}(S, w) = h \} = \{ w \in \mathbb{R}^W : \mathcal{L}_0(w) = h - \mathcal{L}_0 \} \]

both of which are the \( W \)-dimensional hyperellipsoid given by the equation \( \mathcal{L}_0(w) = h - \mathcal{L}_0 \), which can be converted to the standard form for hyperellipsoids as:

\[ (w - w_0)^T \Sigma^{-1} (w - w_0) = 1 \]

The volume enclosed by this hyperellipsoid is exactly the volume of \( \mathcal{M}(w_0) \), i.e., \( V \); so we have

\[ \frac{\pi^{W/2}}{\Gamma(W/2 + 1)} \sqrt{2^W (h - \mathcal{L}_0)^W |\Sigma|} = V \]

Solve for \( h \), with the Stirling’s approximation for factorial \( \Gamma(n + 1) \approx \sqrt{2\pi n} \left( \frac{n}{e} \right)^n \), we have

\[ h = \mathcal{L}_0 + \frac{\left( V \cdot \Gamma(W/2 + 1) \right)^{2/W}}{2\pi |\Sigma|^{1/W}} \approx \mathcal{L}_0 + \frac{V^{2/W} \pi^{1/W} W^{(W+1)/W} |\mathcal{I}_S(w_0)|^{1/W}}{4\pi e} \]

where \( \Gamma(\cdot) \) denotes the Gamma function. Notice that for modern DNNs we have \( W \gg 1 \), and so \( W^{W+1} \approx W \). We finally can derive the generalization bound in the form of the PAC-Bayes Theorem as:

\[ \mathbb{E}_{w \sim \mathcal{Q}} [\mathcal{L}(D, w)] \leq \mathbb{E}_{w \sim \mathcal{Q}} [\mathcal{L}(S, w)] + 2 \sqrt{\frac{W \cdot V^{2/W} \pi^{1/W} |\mathcal{I}_S(w_0)|^{1/W} + 4\pi e \mathcal{L}_0 + 2\pi e \ln \frac{2N}{\delta}}{2\pi e (N - 1)}} \]

C. Derivation of Equation 6 in Section 5.3

First, let us present the well-known theorem in linear algebra that relates the eigenvalues of a matrix to those of its sub-matrices.
**Theorem C.** Given an $n \times n$ real symmetric matrix $A$ with eigenvalues $\lambda_1 \leq \ldots \leq \lambda_n$, for any $k < n$ denote its principal sub-matrix as $B$ obtained from removing $n-k$ rows and columns from $A$. Let $\nu_1 \leq \ldots \leq \nu_k$ be the eigenvalues of $B$. Then for any $1 \leq r \leq k$, we have $\lambda_r \leq \nu_r \leq \lambda_{r+n-k}$.

Let $\{
u_n\}_{n=1}^{N'}$ be the eigenvalues of $\frac{1}{W} \xi_t^t(w_0)$, which is a $N' \times N'$ sub-matrix of $I_{S'}(w_0)$; then

$$
\hat{\gamma}(w_0) = \frac{1}{T} \sum_{t=1}^{T} \ln |\xi_t^t(w_0)| = \frac{1}{T} \sum_{t=1}^{T} \ln |W \cdot \frac{1}{W} \xi_t^t(w_0)| = N' \ln W + \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N'} \ln \nu_n
$$

Theorem C gives the relation between $\nu_n$ and $\lambda_n$, defined above and in Section 5.3 as the $n$th smallest eigenvalues of $\frac{1}{W} \xi_t^t(w_0)$ and that of $I_{S'}(w_0)$, respectively. For sufficiently large $N'$, we can use $\nu_n$ to approximate $\lambda_n$, which ignores the eigenvalues of $I_{S'}(w_0)$ larger than $\lambda_{N'}$. This is reasonable when estimating $\gamma(w_0)$, since in general the majority of the eigenvalues of the Hessian for DNNs are close to zero with only a few large “outliers”, and so the smallest eigenvalues are the dominant terms in $\gamma(w_0)$ (Pennington & Worah, 2018; Sagun et al., 2018; Karakida et al., 2019). A specific bound of the eigenvalues remains an open question, though. In short, we have $\sum_{n=1}^{N'} \nu_n \approx \sum_{n=1}^{N'} \lambda_n'$ and consequently:

$$\frac{W}{N'} \hat{\gamma}(w_0) + W \ln \frac{1}{W} = \frac{W}{N'} (\hat{\gamma}(w_0) - N' \ln W) = \frac{1}{T} \sum_{t=1}^{T} \frac{W}{N'} \sum_{n=1}^{N'} \ln \nu_n \approx \frac{1}{T} \sum_{t=1}^{T} \frac{W}{N'} \sum_{n=1}^{N'} \ln \lambda_n'
$$

Finally we have

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \frac{W}{N'} \sum_{n=1}^{N'} \ln \lambda_n' = \gamma(w_0)
$$

**D. Details of Calculating the Metrics in Section 7.1**

For the following three metrics, we apply estimation by sampling a subset $S^t$ from the full training set $S$ for $T$ times and averaging the results.

- **Frobenius norm**: $\| \nabla_w^2 \mathcal{L}(S, w) \|_F^2$
- **Spectral radius**: $\rho(\nabla_w^2 \mathcal{L}(S, w))$
- **Ours**: $\hat{\gamma}(w) = \frac{1}{T} \sum_{t=1}^{T} \ln |\xi(S^t, w_0)|$

For the Frobenius norm based metric, from Equation 1 & 2 in Section 5.3 we have:

$$\| \nabla_w^2 \mathcal{L}(S, w) \|_F^2 = \| I_S(w) \|_F^2 = \frac{1}{N} \sum_{(x, y) \in S} \sum_{i=1}^{K} \left\| (\nabla_w [\ell_x(w_0)]_i) (\nabla_w [\ell_x(w_0)]_i)^T \right\|_F^2$$

We define $y = \arg \max(y)$. Similar to Equation 4 in Section 5.3, we approximate $y$ by $\hat{y}$ and so

$$\| \nabla_w^2 \mathcal{L}(S, w) \|_F^2 \approx \frac{1}{N} \sum_{(x, y) \in S} \left\| (\nabla_w [\ell_x(w_0)]_{y}) (\nabla_w [\ell_x(w_0)]_{y})^T \right\|_F^2$$
Summing over the entire Hessian matrix is too expensive as there are $W \times W \times N$ entries in total. We therefore estimate the quantity by first sampling a subset $S^t \subset S$ and then sampling 100,000 entries of $(\nabla_w [\ell_x(w_0)]_y) (\nabla_w [\ell_x(w_0)]_y)^T$. We perform the estimation $T$ times and average the results, similar to the approach when computing $\hat{\gamma}(w)$.

Also by Equation 2 and the approximation in Equation 4, the spectral radius of Hessian is equivalent to the squared spectral norm of $1/\sqrt{N} J_w[\tilde{L}(S, w)]$. We also perform estimation (with irrelevant scaling constants dropped) by sampling $S^t$ for $T$ times, i.e., via $\frac{1}{T} \sum_{t} \| J_w[\tilde{L}(S^t, w)] \|^2_2$.

Furthermore, in all our experiments that involves samplings $S^t$, we set $|S^t| = N' = T = 100$.

### E. Architecture And Training Details in Section 7

Architecture details are as below

- The plain CNN is a 6-layer convolutional neural network similar to the baseline in Lee et al. (2016) yet without the "mlpconv" layers (resulting in a much fewer number of parameters). Specifically, the 6 layers has numbers of filters as {64, 64, 128, 128, 192, 192}. We use $3 \times 3$ kernel size and ReLU as the activation function. After the second and the fourth convolutional layer we insert a $2 \times 2$ max pooling operation. After the last convolutional layer, we apply a global average pooling before the final softmax classifier.

- For ResNet-20, WRN-28-2-B(3,3), WRN-18-1.5 and DenseNet-BC-k=12, we use the same architecture as in their original papers, respectively.

The training details are

- For the plain CNN, we initialize the weights according to the scheme in He et al. (2016) and apply l2 regularization of a coefficient 0.0001. We perform standard data augmentation, the one denoted 4-crop-f in Section 7.1. We use stochastic gradient descent with Nesterov momentum set to 0.9 and a batch size of 128. We train 200 epochs in total with the learning rate initially set to 0.01 and then divided by 10 at epoch 100 and 150.

- For ResNet-20, WRN-28-2-B(3,3), WRN-18-1.5 and DenseNet-BC-k=12, we use the same hyper-parameters, training schemes, data augmentation schemes, optimization methods, etc., as those in their original papers, respectively. An exception is that for WRN-18-1.5 on ImageNet, we first resize all training images to $128 \times 128$, and then apply random crop (of size $114 \times 114$), horizontal flip and standard color jittering together with mean channels subtraction as in He et al. (2016). We adopt single crop (central crop) testing for the down-sampled $128 \times 128$ validation images.

### References


