A Practical PAC-Bayes Generalisation Bound for Deep Learning

Anonymous authors
Paper under double-blind review

Abstract

Under an approximate PAC-Bayesian framework, we derive an implementation efficient parameterisation invariant metric to measure generalisation. We show that for solutions of low training loss, this metric can be approximated at the same cost as a single step of SGD. We investigate the usefulness of this metric on pathological examples, where traditional Hessian based sharpness metrics and generalisation both increase. We find good experimental agreement with our efficient and easily implementable metric. As a consequence of our PAC-Bayesian framework and theoretical arguments on the sub-sampled Hessian, we include a trace of Hessian term into our structural risk. We find that this term promotes generalisation on a variety of experiments using Wide-Residual Networks on the CIFAR-100 and ImageNet-32 datasets.

1 Introduction

Despite their exceptional ability to generalise to unseen data, Deep Neural Networks (DNNs) are not completely immune to the classical problem of over-fitting. Large, expressive modern neural networks easily fit training data, including that of random labels (Zhang et al., 2016). However, finding and encouraging solutions which generalise best to new data, measured by their performance on a held out validation or test set, can be considerably more tricky. Put simply, for many problems of interest, the corresponding empirical risk minimisation is much simpler and well understood than the appropriate structural risk minimisation.

In order to ensure the greatest generalisation of their models, practitioners resort to a variety of direct and indirect methods. *Direct methods* involve augmenting or altering the loss. Examples include a penalising weight norm term, known as weight decay or L_2 regularisation, which can be shown to reduce the effect of static noise on the targets (Krogh and Hertz, 1992). Further direct methods involve collecting more training data, as in the limit of infinite data, the empirical risk converges to the true risk. Cheaper approximations to an increase in training data (which can often be expensive and require human annotations), involve data augmentation, encouraging the network to learn symmetries in the data or training on convex combinations of examples (Zhang et al., 2017). Adversarial training, further reduces the sensitivity to small input perturbations and can also be shown to increase generalisation. *Indirect methods* typically involve alterations to the optimisation procedure, which promote generalisation, often at the expense of optimisation. Examples include early stopping, i.e. closely monitoring the validation error/loss to find an optimal point in training corresponding to a model with good generalisation. Other methods involve indirectly steering the optimisation trajectory towards minima with properties considered favourable for generalisation. One extremely prolific example of this is altering the optimisation procedure to promote settling into "flat" minima, which are considered to generalise better under both a Bayesian and minimum description length framework (Hochreiter and Schmidhuber, 1997).

Enter Sharpness: Sharpness is usually measured by properties of the second derivative of the loss, the Hessian $\boldsymbol{H} = \nabla^2 L(\boldsymbol{w})$ (Keskar et al., 2016; Jastrzebski et al., 2017b; Chaudhari et al., 2016; Wu et al., 2017; 2018), such as the spectral norm or trace. The assumption is that due to finite numerical precision (Hochreiter and Schmidhuber, 1997) or from a

Bayesian perspective (MacKay, 2003), the test surface is *shifted* from the training surface. The difference between train and test loss for a shift $\boldsymbol{\delta w}$ is given by

$$L(\boldsymbol{w}^* + \boldsymbol{\delta w}) - L(\boldsymbol{w}^*) \approx \boldsymbol{\delta w}^T \boldsymbol{H} \boldsymbol{\delta w} + ... \approx \sum_i^P \lambda_i |\boldsymbol{\phi}_i^T \boldsymbol{\delta w}|^2 \approx \frac{\mathrm{Tr}(\boldsymbol{H})}{P} ||\boldsymbol{\delta w}||^2 \leq \lambda_1 ||\boldsymbol{\delta w}||^2 \quad (1)$$

in which $\boldsymbol{w}^*$ is the final training point and $[\lambda_i, \boldsymbol{\phi}_i]$ are the eigenvalue/eigenvector pairs of $\boldsymbol{H} \in \mathbb{R}^{P \times P}$. We have dropped the terms beyond second-order and assumed that the gradient at training end is small. In general we have no a priori reason to assume that shift should preferentially lie along any of the Hessian eigenvectors, which in conjunction with strong high dimensional concentration results (Vershynin, 2018), gives $|\boldsymbol{\phi}_i^T \boldsymbol{\delta w}|^2 \approx 1/P$. This justifies the trace as a measure of sharpness. In the worst case scenario the shift is completely aligned with the eigenvector corresponding to the largest eigenvalue λ_1, i.e. $\boldsymbol{\delta w}^T \boldsymbol{\phi}_1 = 1$. Hence the spectral norm λ_1 of $\boldsymbol{H}$ serves as a local[1] upper bound to the loss change.

Avoiding sharp minima can be done indirectly by using learning rate schedules with initially large learning rates (Berrada et al., 2018; Granziol et al., 2020; Jastrzebski et al., 2020; Wu et al., 2017; 2018). Although there is no guarantee that when the learning rate is dropped later in training (required for the convergence of SGD (Nesterov, 2013)) that it will not fall into a minimum of even greater sharpness. Alternative procedures thought to promote flatness include the use of Polyak averaging in conjunction with large learning rates (Izmailov et al., 2018) and alternative optimisers, such as Entropy-SGD (Chaudhari et al., 2016). In contrast to direct methods, indirect methods require significantly more effort on the part of the experimenter. Ideally we would directly optimise a structural risk faithfully corresponding to the true risk.

2 Motivation

Despite the development of more advanced tools to calculate the Hessian of DNNs (Granziol et al., 2019; Ghorbani et al., 2019; Papyan, 2018; Yao et al., 2018), extensive investigations into the nature of the Hessian (Papyan, 2020; Granziol et al., 2020; Choromanska et al., 2015; Pennington and Bahri, 2017), there have been limited practical developments in explicitly using the Hessian to help practitioners generalise better.

Implicit measures to promote flatness have included: Keskar et al. (2016); Rangamani et al. (2019), who consider how large batch vs small batch stochastic gradient descent (SGD) alters the sharpness of solutions, with smaller batches leading to convergence to flatter solutions, leading to better generalisation. Jastrzebski et al. (2017a) look at the importance of the ratio learning rate and batch size in terms of generalisation, finding that large ratios lead to flatter minima (as measured by the spectral norm) and better generalisation. However the prescription of using of a small batch size (or large learning rate to batch size ratio) to generalise better, is still indirect and hence comes at significant financial and environmental expense. Small batch training fails to take advantage of the parallelisation potential of large batch training (Goyal et al., 2017). Determining an effective learning rate schedule requires many optimisation runs, which is expensive and the solutions further need to be evaluated on a held out validation set. This reduces the amount of data used for training and can hurt the model performance at test time.

In this paper we bridge the gap between theory and practice by:

- Adding to the case against Hessian based metrics of generalisation by showing that weight decay, which is known to increase generalisation, increases sharpness. We derive this theoretically for the MLP with ReLU activations and extensively showcase this experimentally.
- Deriving a parameterisation invariant generalisation measure under a PAC-Bayes framework, which combines both the weight norm and properties of the Hessian.

[1] we use the word local here because the largest eigenvalue/eigenvector pair may change along the path taken

- Providing a cheap approximation to the generalisation measure, which can be consistently estimated at the cost of one mini-batch gradient descent step. This could serve as a drop in replacement for evaluating the generalisation of a solution on a held out validation set, allowing for all the training data to be used.
- Showing that both these generalisation measures, work well even on pathological examples exposed by our theoretical contribution.
- Inspired by our PAC-Bayes generalisation bound, introduce a trace of the Hessian regularisation term into the loss, which whilst only increasing training time by a constant factor, we show to improve generalisation WideResNet-28x10 on CIFAR-100 & ImageNet-32.

3 Flatness is a False Friend

Dinh et al. (2017) show that by exploiting ReLUs (Rectified Linear Units) positive homogeneity property $f(\alpha \boldsymbol{x}) = \alpha f(\boldsymbol{x})$, any flat minima can be mapped into a sharp minimum, without altering the loss. As these measures can be arbitrarily distorted, this implies they serve little value as generalisation measures. However such transformations alter other properties, such as the weight norm. In practice the use of $L2$ regularisation, which penalises weight norm means that such manipulations *do* alter the loss in practice. It can even be shown that unregularised SGD converges to the minimum norm solution for simple problems (Wilson et al., 2017).

In conjunction with the numerous positive empirical results relating sharpness and generalisation, it is hence questionable whether reparameterisation arguments alone are enough to discard Hessian based measures of generalisation in the wild. To further add to the case against Hessian based measures of generalisation, we show that for an unregularised network, in the limit of zero loss (complete fitting of the training data which we intuitively expect to over-fit), Hessian based measures of sharpness become exactly zero.

3.1 Theoretical Argument for an MLP with ReLU

Consider a neural network with a d_x dimensional input $\boldsymbol{x}$. Our network has $H-1$ hidden layers and we refer to the output as the H'th layer and the input as the 0'th layer. We denote the ReLU activation function as $f(x)$ where $f(x) = \max(0, x)$. Let $\boldsymbol{W}_i$ be the matrix of weights between the $(i-1)$'th and i'th layer. For a d_y dimensional output our q'th component of the output can be written as

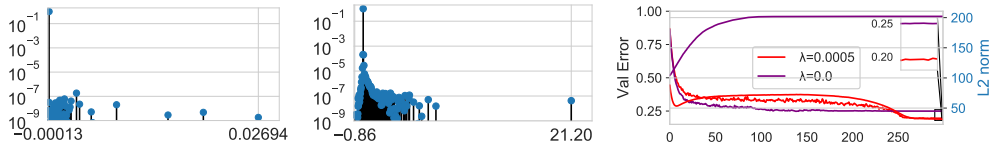
$$\boldsymbol{z}(\boldsymbol{x}_i; \boldsymbol{w})_q = f(\boldsymbol{W}_H^T f(\boldsymbol{W}_{H-1}^T ....f(\boldsymbol{W}_1 \boldsymbol{x}))) = \sum_{i=1}^{d_x} \sum_{j=1}^{\gamma} \boldsymbol{x}_i A_{i,j} \prod_{k=1}^{H} w_{i,j}^{(k)} \tag{2}$$

where the indices i, j denote the sum over network inputs and paths respectively and γ is the number of paths. $A_{i,j} \in [0, 1]$ denotes whether the path is active or not and $w_{i,j}^{(q)}$ denotes the the weight of the path segment which connects node i in layer $q-1$ with node j in layer q. layer i has n_i nodes and $\gamma = \prod_q^{H-1} n_q$.

Theorem 1. *For any feed forward neural network with ReLU output activation functions $f(x) = \max(0, x)$, coupled a softmax output in the final layer and cross entropy loss, in the limit that the training loss $L(\boldsymbol{w}) \to 0$ the spectral norm $\lambda_1(\boldsymbol{H})$ of the empirical Hessian $\boldsymbol{H} = \nabla^2 L(\boldsymbol{w}) \in \mathbb{R}^{P \times P}$ also tends to 0.*

Remark. The proof (given in the Supp Matt) can be extended trivially to both the trace and Frobenius norm. It relies on the weights needing to become large to drive the cross entropy loss to zero.

Remark. By continuity we expect small training loss solutions to have larger spectral norms. This implies that the spectral norm (along with other Hessian based measures of sharpness) should increase in the case where the weights are bounded and the loss cannot go to zero, i.e. when we use L_2 regularisation. We demonstrate this on the WideResNet-28x10 on the CIFAR-100 dataset in Figure 1. Note that the increase in sharpness corresponds to an decrease in weight norm, indicating that both might play a part in generalisation.



(a) $Val = 75.2$, $\lambda = 0$ (b) $Val = 80.6$, $\lambda = 5e^{-4}$ (c) Test Error & $L2$ norm

Figure 1: Hessian spectrum for WideResNet28×10 after 300 epochs of SGD on the CIFAR-100 dataset, for various $L2$ regularisation co-efficients λ, batch norm evaluation mode

4 A PAC Bayesian Approach to Generalisation and Flatness

For an input, output pair $[\boldsymbol{x}_i, \boldsymbol{y}_i] \in [\mathbb{R}^{d_x}, \mathbb{R}^{d_y}]$, $i \in [1, N]$ and a given prediction function $h(\cdot;\cdot) : \mathbb{R}^{d_x} \times \mathbb{R}^P \to \mathbb{R}^{d_y}$, we consider the family of prediction functions parameterised by a weight vector $\boldsymbol{w}$, i.e., $\mathcal{H} := \{h(\cdot; \boldsymbol{w}) : \boldsymbol{w} \in \mathbb{R}^P\}$ with a given loss function $\ell(h(\boldsymbol{x}; \boldsymbol{w}), \boldsymbol{y}) : \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \to \mathbb{R}$. Optimizing the PAC-Bayesian generalization bound Germain et al. (2016) is equivalent to optimizing

$$\int \sum_{i=1}^{N} \log p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w}) q(\boldsymbol{w}) d\boldsymbol{w} - \mathrm{KL}(q(\boldsymbol{w})||p(\boldsymbol{w})), \tag{3}$$

where p can be a categorical distribution, whose likelihood corresponds to a softmax loss function l. $p(\boldsymbol{w})/q(\boldsymbol{w})$ are the prior/posterior of the weights respectively. For example, a random initialized weight can be seen as a sample from the prior and a trained weight can be seen as a sample from the posterior. Notice that, this objective is the lower bound of the log-volume, see Barber (2012, section 28.3.1)

$$\log Z \geq \int \sum_{i=1}^{N} \log p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w}) q(\boldsymbol{w}) d\boldsymbol{w} - \mathrm{KL}(q(\boldsymbol{w})||p(\boldsymbol{w})) \tag{4}$$

where $Z = \int \tilde{p}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) p(\boldsymbol{w}) d\boldsymbol{w}$. The volume interpretation extends the flatness concept discussed in Section 3, and the KL divergence is invariant to any invertible transformations (Cover, 1999). This property makes the volume a reparameterization invariant measure whereas the flatness is not invariant to reparameterization (Dinh et al., 2017).

The optimal $q^*(\boldsymbol{w})$ is the Gibbs distribution $q^*(\boldsymbol{w}) \propto \tilde{p}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) p(\boldsymbol{w})$. However, this may not be achievable in practice. We can only access to some local minima of

$$\boldsymbol{w}_l^* \in \boldsymbol{w}_l^* = \arg \mathrm{localmin}_{\boldsymbol{w}} \log(\tilde{p}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) p(\boldsymbol{w})). \tag{5}$$

We now need to compare which $\boldsymbol{w}_l^*$ is better. Taking $q(\boldsymbol{w}_l^*)$ as Gaussian distribution $\mathrm{N}(\boldsymbol{w}^*, H_{\boldsymbol{w}_l^*}^{-1})$, which corresponds to the Laplace approximation around the minimum (MacKay, 2003). Further assuming an isotropic zero mean Gaussian prior $\mathrm{N}(0, \sigma_p^2 I_D)$, which corresponds to a maximum entropy Jaynes (1957a; 1967; 1957b) prior. This encodes a state of maximal initial ignorance about the distribution of plausible weight distributions under the constraint of an existence in mean and variance. In this case the KL divergence has a closed form:

$$\mathrm{KL}(q||p) = \frac{1}{2}\left[\log \frac{|\sigma_p^2 I|}{|H^{-1}|} + \frac{1}{\sigma_p^2}\mathrm{Tr}(H) + \frac{1}{\sigma^2}||\boldsymbol{w}_p^*||x_2^2\right] = \frac{1}{2\sigma_l^2}\left[\mathrm{Tr}(H) + ||\boldsymbol{w}_l^*||_2\right] + \frac{1}{2}\log|H|, \tag{6}$$

where we drop constant factors. A trivial bound for $\log|H| \leq \mathrm{Tr}(H - I_D)$. Further assuming $\sigma_p^2 = 1$, we have

$$\mathrm{KL}(q||p) \propto \mathrm{Tr}(H) + \frac{1}{2}||\boldsymbol{w}_l^*||_2^2. \tag{7}$$

We can also approximate the term corresponding to the empirical risk by making a Lapalace approximation around the posterior and using the concavity of the logarithm function along

with Jensens inequality.

$$\int \sum_{i=1}^{N} \log p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w}) q(\boldsymbol{w}) d\theta \lessapprox \sum_{i=1}^{N} \log p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w}^*) + \frac{P}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{H}|. \tag{8}$$

The above is for a stochastic classifier $q(\theta)$,we can also build the link between stochastic classifier to deterministic classifier, e.g. the deterministic classifier is a sample from $q(\theta)$. For further discussion on this technicality see Tsuzuku et al. (2020). We note that the total constant term is $P(\frac{N \log 2\pi}{2} + \log \sigma - 1)$, but since it is constant for a given prior, it can be ignored when comparing solutions of the same network. Given that DNNs are known to be rank-degenrate in their Hessians (Granziol et al., 2020; Papyan, 2018; Ghorbani et al., 2019) and that the a bound on rank degeneracy can be shown to increase with network size (Granziol et al., 2020), future work could investigate the effective dimension P^*, when comparing the full PAC-Bayesian generalisation bound of different networks. Hence for comparing the generalisation of solutions of the same network and dropping constant terms we consider:

$$\sum_{i=1}^{N} \log p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w}^*) - \frac{1}{2} \log |\boldsymbol{H}|(1 + \frac{1}{N}) - \frac{1}{2\sigma^2 N} [\mathrm{Tr}\, \boldsymbol{H} + ||w||^2]. \tag{9}$$

4.1 Implications

Equation 9 implies that an objective measure of generalisation is given by the combination of log likelihood, sharpness (a weighted combination of the trace and log determinant) and weight norm. This is pleasantly unsurprising, given that arguments of our motivational section 3 hinged on the importance of large weights, known to give poor generalisation (Krogh and Hertz, 1992), yet small Hessian based measures of sharpness.

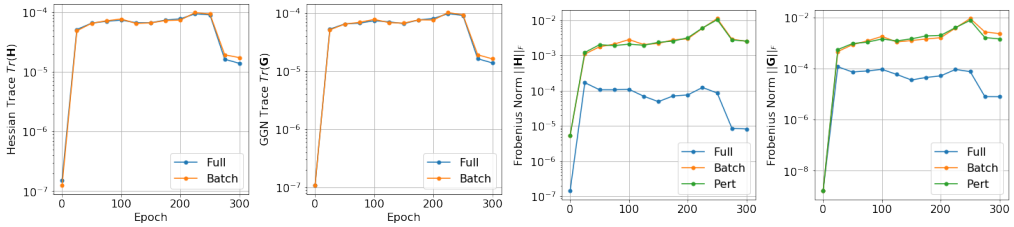
An obvious implication of this formula is that increasing weight decay decreases the generalisation gap. This has already known and common experimental practice. However its origin from our derivation, namely the assumption of a uniform prior over the weights, gives a natural explanation for the efficacy of decoupled weight decay (Loshchilov and Hutter, 2018) in Adam (Kingma and Ba, 2014) in terms of increased generalisation. Quite specifically, decreasing the weights corresponds to a maximally entropic (Jaynes, 1957b) i.i.d Gaussian prior assumption on the weights, as opposed to a prior of lower entropy of $(1 - \gamma \boldsymbol{B}^{-1})\boldsymbol{w}$ where $\gamma, \boldsymbol{B}$ are the learning rates and implied curvature matrices respectively. Given that lower entropy priors correspond to increased information in the form of constraints (Jaynes, 1957a; Granziol and Roberts, 2017), we expect poorer performance with the use of such priors unless the constraints correspond to our knowledge about the system.

Another implication is that by aurgmenting the empirical loss with L_2 regularisation, we are leaving certain generalisation gains on the table by ignoring the other terms related to Hessian sharpness. Based on this insight, we integrate an explicit trace of Hessian term into the loss, which we evaluate extensively in Section 6.

4.2 On the Unique Practical Value of the Trace:

In this sub-section we argue using both known results from linear algebra and a novel application for an additive noise model for the Hessian (Granziol et al., 2020), that Equation 6 (and by the same token Equation equation 9) is significantly less practical than Equation 7.

The Problem with the Log Determinant: A naive implementation of Equation 6 using SVD would involve an impractical computational cost of $\mathcal{O}(P^3 N)$ computational cost, where P, N denote the model/dataset size respectively. Using more advanced stochastic Lanczos quadrature algorithms, the computational cost of the prohibitive log determinant term can be reduced to $\mathcal{O}(mdPN)$, where m denotes the number of moments used and d the number of stochastic trace vectors. The key concept behind this efficient implementation is known as stochastic trace estimation Hutchinson (1990); Fitzsimons et al. (2017); Granziol



(a) Tr($\boldsymbol{H}$) (b) Tr($\boldsymbol{G}$) (c) Tr($\boldsymbol{H}^2$) (d) Tr($\boldsymbol{G}^2$)

Figure 2: **The Trace of the Hessian can be accurately estimated with sub-sampling, the higher order moments (such as the Frobenius norm) cannot be**. Trace of the Full/Batch Hessian Tr($\boldsymbol{H}$)/GGN Tr($\boldsymbol{G}$) and of the Full/Batch Hessian/GGN Frobenius Norm.

and Roberts (2017). This asserts that $\operatorname{Tr} \boldsymbol{H} = \frac{1}{n} \sum_i^n \boldsymbol{v}_i^T \boldsymbol{H} \boldsymbol{v}_i$ as $n \to 0$ for zero mean unit variance vectors $\boldsymbol{v}_i$. Whilst there exist probabilistic bounds on the number of moments and trace vectors needed to guarantee an estimation error (Han et al., 2015; Roosta-Khorasani and Ascher, 2015), these bounds are very loose. As shown in Granziol et al. (2018); Granziol and Roberts (2017); Fitzsimons et al. (2017) they require hundreds of thousands of trace vectors, when in practice small numbers in the tens suffice. Furthermore the number of moments needed to estimate the log determinant accurately depends on the square root of the condition number (Ubaru et al., 2017), whereas that of the trace does not. Neural network Hessia have been shown to have large outliers and a large spectral peak near the origin (Ghorbani et al., 2019; Granziol et al., 2020; Papyan, 2020) and hence we expect the condition number to be large. Even ignoring the constant factors of d, m required to get a good estimate, the factor N means that even in the best case the cost of estimating Equations 6,7 would be as costly as several epochs of SGD, making them impractical.

Why we can sub-sample the Trace and not the Log Determinant: Let us consider the simplest framework to measure the effect of sub-sampling on the Hessian. Following the framework from Granziol et al. (2020), we consider the additive perturbation model of the batch Hessian,

$$\boldsymbol{H}_{batch}(\boldsymbol{w}) = \boldsymbol{H}_{emp}(\boldsymbol{w}) + \boldsymbol{\epsilon}(\boldsymbol{w}). \tag{10}$$

In expectation the trace of the batch Hessian is given by:

$$\mathbb{E}\operatorname{Tr}[\boldsymbol{H}_{batch}] = \mathbb{E}\operatorname{Tr}[\boldsymbol{H}_{emp}(\boldsymbol{w}) + \boldsymbol{\epsilon}(\boldsymbol{w})] = \operatorname{Tr}[\boldsymbol{H}_{emp}(\boldsymbol{w})] \tag{11}$$

and for the trace of the batch Hessian squared:

$$\mathbb{E}\operatorname{Tr}[\boldsymbol{H}^2_{batch}] = \mathbb{E}\operatorname{Tr}[\boldsymbol{H}^2_{emp}(\boldsymbol{w}) + \boldsymbol{\epsilon}^2(\boldsymbol{w}) + 2\boldsymbol{\epsilon}(\boldsymbol{w})\boldsymbol{H}_{emp}(\boldsymbol{w})] = \operatorname{Tr}[\boldsymbol{H}^2_{emp}(\boldsymbol{w})] + P\sigma^2 \tag{12}$$

Hence, under the assumptions of our model, we expect the batch trace to be equal to that of the empirical, but the Frobenius norm (or its square as defined above) to be larger. This is shown in Figure 2, where we show the difference between a full dataset Hessian trace and Frobenius norm and that of a subsample $B = 128$. We see both that the Trace can be accurately estimated with the sub-sample, whilst the Frobenius norm cannot be. We show that this holds also for common positive semi-definite approximations to the Hessian, such as the Generalised Gauss Newton. Note that since we measure the log-determinant using a moment matched approximation to the spectrum, the moments must match accurately. Since higher order moments are perturbed by sub-sampling, it is not possible to calculate the log determinant using a sub-sample of the Hessian. Whilst it may be possible to use the framework of Granziol et al. (2020) to calculate the required corrections to the moments to allow sub-sampling, which we leave to future work, their method requires the variance of the Hessian, which is expensive to compute. Note that since the trace of the Hessian is invariant to sub-sampling, Equation 7 *can be accurately estimated at the cost of a single SGD step*. Since the empirical risk dependent term also depends on the log-determinant, for this approximation to be useful we need the empirical loss of the solutions being compared to be very close to each other.

5 Experiments on Pathological Examples

In order to test the usefulness our framework from Section 4 in the specific setting where we increase the weight decay coefficient (and expect an increase in generalisation and sharpness), we run experiments with increasing weight decay coefficients on a 1-Layer Multi-Layer Perceptron, a simple Convolutional Neural Network, Pre-Residual and Wide-Residual Networks on the MNIST and CIFAR-100 datasets. For each experiment we present the results for the difference in Error ΔE, loss diference δL, along with the Hessian trace, Frobenius Norm, Spectral norm $|\boldsymbol{H}|$, along with approximation of the PAC-Bayesian risk difference $\mathfrak{a}$. We use $m = 100$ moments to compute an approximate moment matched spectral density using the entire dataset.

Experimental Setup: We use the deep visualisation suite (Granziol et al., 2019) package to visualise the spectrum of the Hessian and calculate the largest eigenvalues. We train all networks using SGD with momentum $\rho = 0.9$ and varying levels of L2 regularisation $\frac{\gamma}{2}||\boldsymbol{w}||^2$, $\gamma \in [0, 0.0001, 0.0005]$. For further experimental details, such as the learning rate schedule (we a linear decay schedule with a terminal learning rate of 0.01 the initial) employed and the finer details of the spectral visualisation method see Appendix C. Since adding $L2$ regularisation naturally adds γ to each eigenvalue, as $\boldsymbol{H} \rightarrow \boldsymbol{H} + \gamma I$, in our results we *do not* calculate the Hessian on the regularised loss.

MLP: We now consider a single hidden layer MLP on the MNIST dataset, with a hidden layer of 100 units, parameter count 9960, trained for 50 epochs with an identical schedule and a learning rate of 0.01. We similarly find that the addition of weight decay both increases the generalisation accuracy (from $94.4 \rightarrow 96.46 \rightarrow 96.7$ as we increase the regularisation coefficient γ from $0 \rightarrow 0.0001 \rightarrow 0.0005$). This also increases the spectral norm as shown in Appendix. The training accuracy increases slightly with the introduction of regularisation, but decreases over the unregularised network when the regularisation is increased to 0.0005. We plot the results in Table 1.

CNN: We consider a 9 layer simple convolutional neural network on the CIFAR-100 dataset (Dangel et al., 2019), with parameter count $1,387,108$ and a learning rate of $\alpha = 0.01$ for 300 epochs. We also observe that adding weight decay increases the spectral norm, as shown in the Appendix. For this network, training is also improved by the addition of a little $L2$ regularisation, but performance decreases for over regularisation, i.e. as the weight decay parameter increases from $[0, 10^{-4}, 5 \times 10^{-4}]$ the training performance is $[86.3\%, 87.9\%, 86.0\%]$. In this particular example the training accuracy is quite low, but there is still a generalisation gap. We plot the results in 2. For both the MLP and the simple CNN, we find that there is

Model	γ	ΔE	ΔL	Tr H	Tr H^2	$\|H\|$	$\mathfrak{a}$
MLP	0.0	0.48	0.024	1.82e-2	1.20e-1	1.44e1	7.6
MLP	1e-4	0.51	0.022	1.67e-2	1.04e-1	1.50e1	7.2
MLP	5e-4	-0.05	0.007	1.76e-2	1.63e-1	1.78e1	6.1

Table 1: MLP MNIST

Model	γ	ΔE	ΔL	Tr H	Tr H^2	$\|H\|$	$\mathfrak{a}$
CNN	0.0	32.40	2.506	1.82e-3	1.35e-2	4.9e1	45.2
CNN	1e-4	32.50	2.463	2.20e-3	1.68e-2	6.6e1	35.8
CNN	5e-4	31.14	2.144	4.62e-3	1.17e-1	1.3e2	23.0

Table 2: CNN CIFAR-100

Table 3: **Accuracy and Loss do not always correspond**. The derived metrics serve as a measure of the difference between the empirical and true risk. This does not always correspond to the difference in accuracy.

not perfect alignment in terms of error difference and loss difference. But given that our analysis is performed on the loss, it is encouraging to see that both metrics predict a decrease in loss difference between as we increase weight decay. In contrast we find that the spectral norm grows with the regularisation predicting worse not better performance and that neither the trace nor the Frobenius norm, serve as reliable indicators of generalisation difference.

PreResNet-164 We use a pre-activated residual network on the CIFAR-100 dataset with parameter count $1,726,388$. Our training performance decreases with increased level of

regularisation $[0, 10^{-4}, 5\times10^{-4}]$ from $[99.987\%, 99.985\%, 99.87\%]$ but our testing performance increases significantly. We show the results in Table 4.

WideResNet-28×10: We use a wide residual network on the CIFAR-100 dataset, with parameter count $36,546,980$, we observe the training accuracy remains roughly constant $[99.984\%, 99.984\%, 99.982\%]$ as we increase the regularisation from $[0, 10^{-4}, 5 \times 10^{-4}]$. We are now in the regime where the optimisation benefit of regularisation is negligible, but the generalisation benefit is significant. We show the results in Table 4.

For these large neural networks, perhaps due to the capacity to fit the training set, we see a large generalisation difference and furthermore we see a complete alignment between generalisation in terms of measurement in loss and accuracy. For these networks we find that all Hessian based measures of sharpness grow with the increased coefficient of L_2 regularisation γ, despite the generalisation difference decreasing with such regularisation. However for our approximate metric we find good agreement with the observed phenomena. Both of them decrease in tandem with the increase in regularisation and the decrease in generalisation difference.

Model	γ	ΔE	ΔL	$\mathrm{Tr}\, H$	$\mathrm{Tr}\, H^2$	$\lvert H\rvert$	$\mathfrak{a}$
P164	0	27.2	2.24	5.5e-5	1.1e-5	2.52	160
P164	1e-4	24.4	1.16	6.7e-4	6e-4	11.49	36.9
P164	5e-4	23.11	0.92	2.1e-3	3.1e-3	23.03	22.06

Table 4: PreResNet-164 CIFAR-100

Model	γ	ΔE	ΔL	$\mathrm{Tr}\, H$	$\mathrm{Tr}\, H^2$	$\lvert H\rvert$	$\mathfrak{a}$
Wrn	0	24.92	2.08	9.4e-7	2.3e-7	1.69	101
Wrn	1e-4	20.68	0.85	6.7e-5	1.3e-4	39.13	21.9
Wrn	5e-4	19.6	0.79	1.1e-4	3e-4	40	15.8

Table 5: WideResNet-28×10 CIFAR-100

Table 6: **Generalisation and Sharpness both increase with greater L_2 regularistaion**. For typical neural networks trained on well known datasets commonly used Hessian metrics, the trace, frobenius and spectral norms all increase as we increase the weight decay coefficient γ. However our general $\mathfrak{g}$ and approximate $\mathfrak{a}$ metrics both indicate decreased generalistaion difference.

6 Spectral Regulariser

Following from Equation 7 and our arguments on the trace invariance to sub-sampling, we consider whether it is possible to *directly* encourage flatness in the loss surface by augmenting the loss with the trace of the Hessian.

$$L(\boldsymbol{w}) \to L(\boldsymbol{w}) + \gamma|\boldsymbol{w}|^2 + \eta\, \mathrm{Tr}\, \boldsymbol{H}(\boldsymbol{w}). \tag{13}$$

Since $\mathrm{Tr}\, \boldsymbol{H}(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{v}} \boldsymbol{v}^T \nabla^2 L(\boldsymbol{w}) \boldsymbol{v}$, where $\boldsymbol{v}$ are random vectors with zero mean and unit variance. We use a monte-carlo approximation of the expectation with a single random vector. For a single random vector, this increases the computational cost of SGD over that of SGD with weight decay by a factor[2] of 3. We leave the investigation of efficient sub-sampling to future work. We provide open-source implementations of both our generalisation measure and spectral regularisation method.

On the Positive Definite Approximation of the Hessian: We note that in Equations 6 and the approximate form 7 that the Hessian was assumed to be positive definite. Whilst zero eigenvalues do not cause problems for the trace approximation of the log determinant, for significant large negative spectral mass, note that the regularising term in Equation 13 becomes negative. Hence large portions of negative eigenvalues actually decrease the objective. It is known, that residual architectures have significant negative spectral mass a the start of training (Granziol et al., 2020). This implies intuitively that increasing the weight of directions of negative curvature, i.e going to areas which are local maxima in the loss is beneficial to reducing the regularised objective. We find empirically that for Wide-Residual Networks this is the case, with training diverging on CIFAR-10 and hovvering around 10% for CIFAR-100. As a proposed fix, we use the generalised Gauss-Newton approximation to the loss, which is positive semi-definite. Hence forth $\mathrm{Tr}\, \boldsymbol{H} \to \mathrm{Tr}\, \boldsymbol{G}$.

[2]If we take the cost of multiplication to be m and that of a gradient operation as g, then we need to take two extra gradients and multiplications over the typical operation

Experimental Procedure: We run the Wide-Residual Network (Zagoruyko and Komodakis, 2016) on both the CIFAR-100 and ImageNet-32 datasets. We show the results in Table 9. We use a linear schedule as detailed in the Appendix, which we find outperforms the step schedule on these experiments. In order to set the initial learning rate and weight decay coefficients α_0, γ for the CIFAR-100 experiment we grid search over the values $[0.05, 0.075, 0.1, 0.125]$ and $[0, 0.0001, 0.0003, 0.0005]$ respectively and choose the maximal performing combinations. For the trace regularising coefficient η we also look over the set of $[0.0001, 0.0003, 0.0005]$ for a single seed and choose the best performing solution. We run all experiments with the SGD optimiser, using a momentum of 0.9. Due to the computational cost of running ImageNet-32 we choose the best learning rate for CIFAR-100 and experiment for the best weight decay (without the trace regulariser enabled) and set that as our default values. We only try one value 0.0001 of the trace regulariser. For CIFAR-100 we run the experiment with 4 random seeds, whereas ImageNet is single shot. Given the lack of hyper-parameter tuning done for the trace-regulariser, it is encouraging to note that for both datasets we see a small but significant improvement in the validation accuracy.

γ	η	α_0	Train Acc	Val Acc
5e-4	1e-4	0.1	99.992 ± 0.002	80.75 ± 0.19
5e-4	0	0.1	99.992 +/- 0.001	80.48 ± 0.26

Table 7: WideResNet-28 × 10 CIFAR-100

γ	η	α_0	Train Acc	Val Acc
1e-4	0	0.1	81.05	62.89
1e-4	1e-4	0.1	80.95	62.66

Table 8: WideResNet-28×10 ImageNet-32

Table 9: **Explicitly including a flatness term into the loss function improves Generalisation.** We the classical Wide-Residual-Network on CIFAR-100 and ImageNet-32 both with and without a trace regularisation term, given by η. Whilst we report a single shot result for ImageNet-32 due to the computational expense, the result is statistically significant for CIFAR-100, where we run 4 seeds.

7 Related Work

Tsuzuku et al. (2020) similarly also consider a PAC-Bayesian approach to measuring generalisation. However, their resulting generalisation measure is significantly more complicated to compute. requiring an inner (convex) optimisation loop for each weight matrix, for which there remains no open source implementation. In contrast, our measure simply requires the calculation of the weight norm and a Hessian vector product with one mini-batch of data. This is easily implementable and available in many Hessian based packages (Granziol et al., 2019; Yao et al., 2018; Ghorbani et al., 2019; Papyan, 2018). Dziugaite and Roy (2017) similarly also compute a PAC-Bayes genneralisation bound using SGD on an MNIST MLP example, using an expensive inner optimisation loop that is not implemented in traditional frameworks.

8 Conclusion

In this paper, we add against the case of using vanilla Hessian based arguments for generalisation by showing that networks trained with the cross entropy loss, with large weights required to drive the loss to zero, have flat Hessian based sharpness metrics. We demonstrate this empirically and propose a simple PAC-Bayesian inspired metric, which can be calculated at the cost of a single step of SGD and is easily implementable in state of the art open-source deep learning software. We release a PyTorch version. We show that this metric is reliable on a set of pathological experiments. We further in conjunction with considerations on the effect of sub-sampling on the Hessian spectrum, consider an unbiased flatness regularisation term into the loss, which we show for ImageNet and CIFAR-100 gives some promising inital results on Wide Residual Networks.

References

David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Deep Frank-Wolfe for neural network optimization. *arXiv preprint arXiv:1811.07591*, 2018.

Tony Cai, Jianqing Fan, and Tiefeng Jiang. Distributions of angles in random packing on spheres. *The Journal of Machine Learning Research*, 14(1):1837–1864, 2013.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-SGD: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838*, 2016.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Felix Dangel, Frederik Kunstner, and Philipp Hennig. Backpack: Packing more into backprop. *arXiv preprint arXiv:1912.10985*, 2019.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1019–1028. JMLR. org, 2017.

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

Jack Fitzsimons, Diego Granziol, Kurt Cutajar, Michael Osborne, Maurizio Filippone, and Stephen Roberts. Entropic trace estimates for log determinants, 2017.

Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien. Pac-bayesian theory meets bayesian inference. *arXiv preprint arXiv:1605.08636*, 2016.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via Hessian eigenvalue density. *arXiv preprint arXiv:1901.10159*, 2019.

Gene H Golub and Gérard Meurant. Matrices, moments and quadrature. *Pitman Research Notes in Mathematics Series*, pages 105–105, 1994.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2012.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Diego Granziol and Stephen Roberts. An information and field theoretic approach to the grand canonical ensemble, 2017.

Diego Granziol, Edward Wagstaff, Bin Xin Ru, Michael Osborne, and Stephen Roberts. Vbald-variational bayesian approximation of log determinants. *arXiv preprint arXiv:1802.08054*, 2018.

Diego Granziol, Xingchen Wan, Timur Garipov, Dmitry Vetrov, and Stephen Roberts. MLRG deep curvature. *arXiv preprint arXiv:1912.09656*, 2019.

Diego Granziol, Timur Garipov, Dmitry Vetrov, Stefan Zohren, Stephen Roberts, and Andrew Gordon Wilson. Towards understanding the true loss surface of deep neural networks using random matrix theory and iterative spectral methods, 2020. URL `https://openreview.net/forum?id=H1gza2NtwH`.

Insu Han, Dmitry Malioutov, and Jinwoo Shin. Large-scale log-determinant computation through stochastic Chebyshev expansions. In *International Conference on Machine Learning*, pages 908–917, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450, 1990.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Stanisław Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017a.

Stanisław Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in SGD. *arXiv preprint arXiv:1711.04623*, 2017b.

Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on the optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=r1g87C4KwB`.

E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957a.

E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957b. doi: 10.1103/PhysRev.106.620. URL `http://link.aps.org/doi/10.1103/PhysRev.106.620`.

Edwin T. Jaynes. *Foundations of Probability Theory and Statistical Mechanics*, pages 77–101. Springer Berlin Heidelberg, Berlin, Heidelberg, 1967. ISBN 978-3-642-86102-4. doi: 10.1007/978-3-642-86102-4_6. URL `http://dx.doi.org/10.1007/978-3-642-86102-4_6`.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

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

Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957, 1992.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2018.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Gérard Meurant and Zdeněk Strakoš. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, 15:471–542, 2006.

Tristan Milne. Piecewise strong convexity of neural networks. In *Advances in Neural Information Processing Systems*, pages 12973–12983, 2019.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Vardan Papyan. The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size. *arXiv preprint arXiv:1811.07062*, 2018.

Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra. *Journal of Machine Learning Research*, 21(252):1–64, 2020.

Barak A Pearlmutter. Fast exact multiplication by the Hessian. *Neural computation*, 6(1): 147–160, 1994.

Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2798–2806. JMLR. org, 2017.

Akshay Rangamani, Nam H Nguyen, Abhishek Kumar, Dzung Phan, Sang H Chin, and Trac D Tran. A scale invariant flatness measure for deep network minima. *arXiv preprint arXiv:1902.02434*, 2019.

Farbod Roosta-Khorasani and Uri Ascher. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, 15(5):1187–1212, 2015.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis. In *International Conference on Machine Learning*, pages 9636–9647. PMLR, 2020.

Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of tr(f(a)) via stochastic Lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017.

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

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pages 4148–4158, 2017.

Lei Wu, Zhanxing Zhu, et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

Lei Wu, Chao Ma, and E Weinan. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. In *Advances in Neural Information Processing Systems*, pages 8279–8288, 2018.

Zhewei Yao, Amir Gholami, Qi Lei, Kurt Keutzer, and Michael W Mahoney. Hessian-based analysis of large batch training and robustness to adversaries. In *Advances in Neural Information Processing Systems*, pages 4949–4959, 2018.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.