RETHINKING PARAMETER COUNTING: EFFECTIVE DIMENSIONALITY REVISITED

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks appear to have mysterious generalization properties when using parameter counting as a proxy for complexity. Indeed, neural networks often have many more parameters than there are data points, yet still provide good generalization performance. Moreover, when we measure generalization as a function of parameters, we see *double descent* behaviour, where the test error decreases, increases, and then again decreases. We show that many of these properties become understandable when viewed through the lens of *effective dimensionality*, which measures the dimensionality of the parameter space determined by the data. We relate effective dimensionality to posterior contraction in Bayesian deep learning, model selection, width-depth tradeoffs, double descent, and functional diversity in loss surfaces, leading to a richer understanding of the interplay between parameters and functions in deep models. We also show that effective dimensionality compares favourably to alternative norm- and flatness- based generalization measures.

1 INTRODUCTION

Parameter counting pervades the narrative in modern deep learning. "One of the defining properties of deep learning is that models are chosen to have many more parameters than available training data. In light of this capacity for overfitting, it is remarkable that simple algorithms like SGD reliably return solutions with low test error" (Dziugaite and Roy, 2017). "Despite their massive size, successful deep artificial neural networks can exhibit a remarkably small difference between training and test performance" (Zhang et al., 2017). "Increasing the number of parameters of neural networks can give much better prediction accuracy" (Shazeer et al., 2017). "Scale sensitive complexity measures do not offer an explanation of why neural networks generalize better with over-parametrization" (Neyshabur et al., 2018). "We train GPT-3, an autoregressive language model with 175 billion parameters, $10\times$ more than any previous non-sparse language model" (Brown et al., 2020). The number of model parameters explicitly appears in many modern generalization measures, such as in Equations $20, 51, 52, 56, 57, 59$, and 60 of the recent study by Jiang et al. (2020). Phenomena such as *double descent* are a consequence of parameter counting. Parameter counting even permeates our language, with expressions such as *over-parametrization* for more parameters than data points.

But parameter counting can be a poor description of model complexity, model flexibility, and inductive biases. One can easily construct degenerate cases, such as predictions being generated by a sum of parameters, where the number of parameters is divorced from the statistical properties of the model. When reasoning about generalization, *over-parametrization* is besides the point: what matters is how the parameters combine with the functional form of the model.

Indeed, the practical success of convolutional neural networks (CNNs) for image recognition tasks is almost entirely about the inductive biases of convolutional filters, depth, and sparsity, for extracting local similarities and hierarchical representations, rather than flexibility (LeCun et al., 1989; Szegedy et al., 2015). Convolutional neural networks have far fewer parameters than fully connected networks, yet can provide much better generalization. Moreover, width can provide flexibility, but it is *depth* that has made neural networks distinctive in their generalization abilities.

In this paper, we gain a number of insights into modern neural networks through the lens of *effective dimensionality*, in place of simple parameter counting. Effective dimensionality, defined by the eigenspectrum of the Hessian on the training loss (equation 2, Section 2), was used by MacKay (1992a) to measure how many directions in the parameter space had been determined in a Bayesian

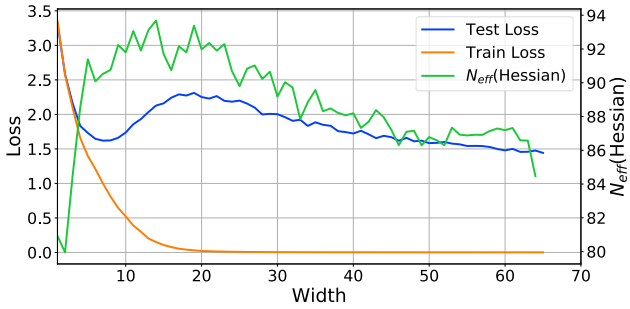


Figure 1: **A resolution of double descent**. We replicate the *double descent* behaviour of deep neural networks using a ResNet18 (He et al., 2016) on CIFAR-100, where train loss decreases to zero with sufficiently wide model while test loss decreases, then increases, and then decreases again.

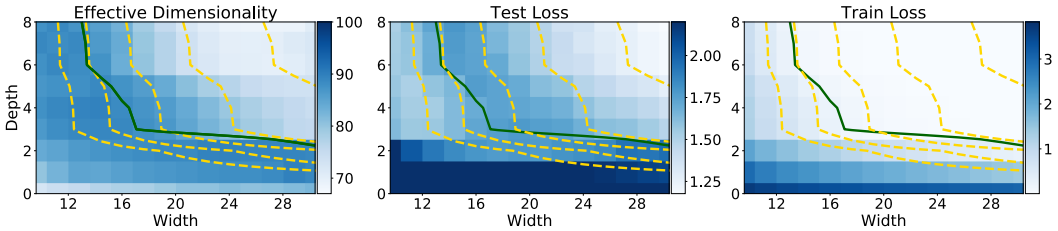


Figure 2: **Left:** Effective dimensionality as a function of model width and depth for a CNN on CIFAR-100. **Center:** Test loss as a function of model width and depth. **Right:** Train loss as a function of model width and depth. Yellow level curves represent equal parameter counts ($1e5$, $2e5$, $4e5$, $1.6e6$). The green curve separates models with near-zero training loss. Effective dimensionality serves as a good proxy for generalization for models with low train loss.

neural network. There is immense value in revisiting effective dimensionality in the context of modern deep learning. We demonstrate that effective dimensionality can be used to explain phenomena such as double descent and width-depth trade-offs in architecture specification (Section 5). We also show that effective dimensionality provides a straightforward, scalable, and promising metric for generalization in modern deep learning, comparing to PAC-Bayes flatness and path-norm measures, two of the most successful measures in the recent study by Jiang et al. (2020) (Section 6). We additionally show how effective dimension can explain why subspace compression methods for neural networks (Izmailov et al., 2019; Li et al., 2018) can be so effective in practice, demonstrating *function-space* homogeneity as we move in directions given by eigenvectors corresponding to the smallest eigenvalues of the Hessian (Section 4). We connect this finding with Bayesian Occam factors and minimum description length frameworks, providing an interpretation of effective dimensionality as model compression (Section 4.3). Moreover, we show that despite a seeming lack of determination in parameter space, a neural network can be relatively well-determined in function space (Section 4).

Consider Figure 1, where we see that once a model has achieved low training loss, the effective dimensionality, computed from training data alone, replicates double descent behaviour for neural networks. Models that are wider have both lower effective dimensionality and better generalization. Alternatively, in Figure 2 we see that width and depth determine effective dimensionality in different ways, though both are related to numbers of parameters. Remarkably, for models with low training loss (above the green partition), the effective dimensionality closely tracks generalization performance for each combination of width and depth. We also see that wide but shallow models overfit, while depth helps provide lower effective dimensionality. Low effective dimensionality indicates greater compression of the data, and thus more regularity, as we discuss in Section 4.3.

2 POSTERIOR CONTRACTION, EFFECTIVE DIMENSION, AND THE HESSIAN

We consider a model, typically a neural network, $f(x; \boldsymbol{\theta})$, with inputs x and parameters $\boldsymbol{\theta} \in \mathbb{R}^k$. We define the Hessian as the $k \times k$ matrix of second derivatives of the loss, $\mathcal{H}_{\boldsymbol{\theta}} = -\nabla\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \mathcal{D})$. To begin, we describe posterior contraction, effective dimensionality, and connections to the Hessian.

2.1 THE HESSIAN AND THE POSTERIOR DISTRIBUTION

We begin by providing a simple example explaining the relationship between the posterior distribution over the model's parameter, the amount of *posterior contraction* from the prior, and the Hessian of the negative log posterior. Figure 3 shows the prior and posterior distribution for a Bayesian linear regression model with a single parameter, with predictions generated by parameters drawn from these distributions. As expected, we see that the variance of the posterior distribution is significantly reduced from that of the prior; we call the difference of the variance between the posterior and the prior the *posterior contraction* of the model. More specifically, as shown in Figure 3, the arrival of data increases the curvature of the loss (negative log posterior) at the optimum. This increase in curvature of the loss that accompanies certainty about the parameters leads to an increase in the eigenvalues of the Hessian of the Growth in eigenvalues of the Hessian of the loss corresponds to increased certainty about parameters.

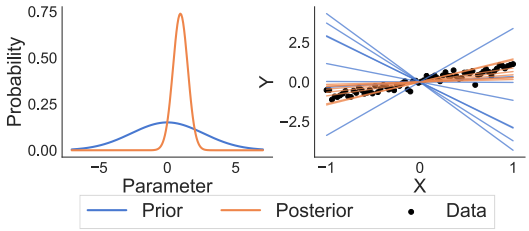


Figure 3: **Left:** A comparison of prior and posterior distributions in a Bayesian linear regression setting, demonstrating the decrease in variance referred to as *posterior contraction*. **Right:** Functions sampled from the prior and posterior distributions, along with the training data.

2.2 POSTERIOR CONTRACTION AND EFFECTIVE DIMENSIONALITY

When combined with the functional form of a model, a distribution over parameters $p(\boldsymbol{\theta})$ induces a distribution over functions $p(f(x; \boldsymbol{\theta}))$. The parameters are of little direct interest — what matters for generalization is the distribution over functions (e.g. the right panel of Figure 3). As parameter distributions concentrate around specific values we expect to generate less diverse functions, the behavior seen in Figure 3. We show in Appendix E that we can describe posterior contraction in Bayesian linear regression, $y \sim \mathcal{N}(f = \boldsymbol{\Phi}^\top \boldsymbol{\beta}, \sigma^2 I)$, with isotropic Gaussian prior, $\boldsymbol{\beta} \sim \mathcal{N}(0, \alpha^2 I_N)$, as

$$\Delta_{post}(\boldsymbol{\Phi}) = \alpha^2 \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_i + \alpha^{-2}}, \tag{1}$$

where λ_i are the eigenvalues of $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$, the Hessian of the log likelihood. We refer to the summation in equation 1 as the *effective dimensionality* of $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$. We generalize equation 1, defining the effective dimensionality of a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{k \times k}$ as

$$N_{eff}(\boldsymbol{A}, z) = \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_i + z}, \tag{2}$$

in which λ_i are the eigenvalues of $\boldsymbol{A}$ and $z > 0$ is a regularization constant MacKay (1992a).

Typically as neural networks are trained we observe a gap in the eigenspectrum of the Hessian of the loss (Sagun et al., 2017); a small number of eigenvalues become large while the rest are near zero. For effective dimensionality, eigenvalues much larger than z contribute a value of approximately one to the summation, and eigenvalues much smaller than z contribute a value of approximately zero.

Therefore, effective dimensionality explains the number of parameters that have been determined by the data, which corresponds to the number of parameters the model is using to make predictions. In comparing models of the same parameterization that achieve low loss on the training data, we expect models with *lower* effective dimensionality to generalize better — which is empirically verified in

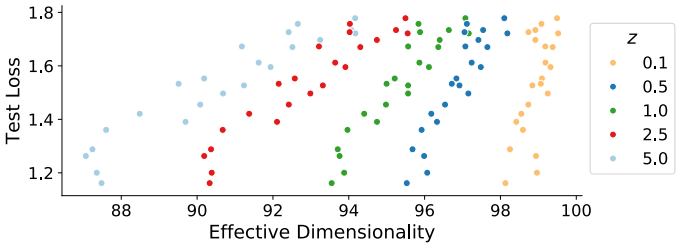


Figure 4: We see a similarly strong relative relationship between effective dimensionality and test loss over a wide range of regularization parameters z.

Figures 1 and 2. The intuition built using Figure 3 carries through to this approximation: as the eigenvalues of the Hessian increase, the eigenvalues of the covariance matrix in our approximation to the posterior distribution shrink, further indicating contraction around the MAP estimate.

Practical Computations For large neural networks computing the eigenvalues and eigenvectors of the Hessian of the loss is nontrivial. We estimate effective dimensionality by computing the dominant eigenvalues using the Lanczos algorithm implemented in GPyTorch, since many of the eigenvalues are typically close to zero and do not significantly contribute to the estimate (Gardner et al., 2018). Hessian vector products as implemented in PyTorch take *three* backward passes; In our experiments on deep nets, we compute 100 Hessian vector products to produce 100 eigenvalues, so that the cost is about $1.5\times$ the cost of standard training. As a heuristic, one can set z using the connection with the prior variance α^2 and ℓ_2 regularization, or to measure the number of relatively large eigenvalues. We use a value of $z = 1$ in equation 2 for all experiments, and show in Figure 4 that effective dimensionality for model comparison is highly robust to different values of z over the range of networks with near zero training loss in Figure 2.

The Hessian (and its effective dimensionality) is not invariant to re-parameterizations (e.g. ReLU rescaling and batch normalization) (MacKay, 2003, Chapter 27). For this reason we assume a fixed parameterization, as is the case in practice, and compare between models of the same parameterization.

3 Related Work

MacKay (1992a) used effective dimensionality to measure posterior contraction in Bayesian neural networks. Effective dimensionality has also been used for measuring generalization error of kernel methods (Caponnetto and Vito, 2007).

Various connections between flatness and generalization have been explored via Occam factors (MacKay, 2003; Smith and Le, 2018) and minimum description length (Hinton and Van Camp, 1993; Achille and Soatto, 2018). Nakkiran et al. (2020) found generalization gains as neural networks become overparameterized, showing the *double descent* phenomenon (e.g., Belkin et al., 2019a; Nakkiran et al., 2020) that occurs as the width increases in residual and convolutional neural networks.

Flatness has also be considered in the PAC-Bayes literature (e.g., Dziugaite and Roy, 2017; Jiang et al., 2020), with Jiang et al. (2020) showing that PAC-Bayesian measures of flatness, in the sense of insensitivity to random perturbations, perform well relative to many generalization bounds. Zhou et al. (2018) used PAC-Bayesian compression arguments to construct non-vacuous generalization bounds at the ImageNet scale.

Our work shows that effective dimensionality can shed light on a number of phenomena in modern deep learning, including double descent, width-depth trade-offs, and Bayesian subspace inference, while providing a straightforward and compelling generalization metric, relative to several of the highest performing metrics in Jiang et al. (2020).

We provide an extended discussion of historical perspectives in Appendix D.

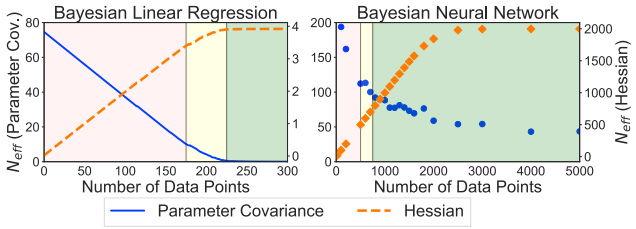


Figure 5: The effective dimensionality of the posterior covariance over parameters and the function-space posterior covariance. Red indicates the over-parameterized setting, yellow the critical regime with $k \approx n$, and green the under-parameterized regime. In both models we see the expected increase in effective dimensionality in parameter space and decrease in effective dimensionality of the Hessian.

4 Posterior Contraction and Function-Space Homogeneity

We demonstrate how the effective dimensionality of both the posterior parameter covariance and the Hessian of the loss provides insights into how a model adapts to data during training. Additionally we show that in directions in which the posterior has not contracted, neural networks are insensitive to parameter perturbations. In other words, we show that although there are many flat directions in a neural network posterior, the functions obtained by moving along those directions are remarkably homogeneous. Finally we relate insensitivity to parameter perturbations to Occam factors, connecting effective dimensionality to data compression. These findings help explain the surprising success of subspace inference methods for capturing functional variability in low dimensional subspaces.

4.1 Posterior Contraction of Bayesian Models

Theorem 4.1 (Posterior Contraction in Bayesian Linear Models). *Let $\mathbf{\Phi} = \mathbf{\Phi}(x) \in \mathbb{R}^{n \times k}$ be a feature map of n data observations, x, with $n < k$ and assign isotropic prior $\boldsymbol{\beta} \sim \mathcal{N}(0_k, \alpha^2 I_k)$ for parameters $\boldsymbol{\beta} \in \mathbb{R}^k$. Assuming a model of the form $y \sim \mathcal{N}(\mathbf{\Phi}\boldsymbol{\beta}, \sigma^2 I_n)$ the posterior distribution of $\boldsymbol{\beta}$ has a $k - n$ directional subspace in which the variance is identical to the prior variance.*

Theorem 4.1, along with an equivalent result for generalized linear models, is proven in Appendix F.1. Theorem 4.1 demonstrates why *parameter counting* often makes little sense: for a fixed data set of size n, only $\min(n, k)$ parameters can be determined, leaving many dimensions in which the posterior is unchanged from the prior when $k \gg n$.

While much effort has been spent grappling with the challenges of high dimensional parameter spaces for Bayesian neural networks, the practical existence of subspaces where the posterior variance has not collapsed from the prior suggests that both computational and approximation gains can be made from ignoring directions in which the posterior variance is unchanged from the prior. This observation helps explain the success of subspace based techniques (e.g., Izmailov et al., 2019) that marginalize the loss in a lower dimensional space. For Bayesian linear models, the effective dimensionality of the parameter covariance is the inverse of the Hessian, and as the effective dimensionality of the parameter covariance decreases, the effective dimensionality of the Hessian increases. Supporting this argument, in Figure 5 we demonstrate for both Bayesian linear models and Bayesian neural networks that as the number of data points grows, the effective dimensionality of the posterior covariance decreases while the effective dimensionality of the Hessian increases until both are limited by the number of parameters. The details of the experimental setup for Figure 5 are given in Appendix B.

4.2 Function-Space Homogeneity

The previous section has demonstrated that parameters can be *undetermined* by the data; we now show that the function induced by the model is *unchanged* in these directions.

Theorem 4.2 (Function-Space Homogeneity in Linear Models). *Let $\mathbf{\Phi} = \mathbf{\Phi}(x) \in \mathbb{R}^{n \times k}$ be a feature map of n data observations, x, with $n < k$, and assign isotropic prior $\boldsymbol{\beta} \sim \mathcal{N}(0_k, \alpha^2 I_k)$ for parameters $\boldsymbol{\beta} \in \mathbb{R}^k$. The minimal eigenvectors of the Hessian define a $k - n$ dimensional subspace in which parameters can be perturbed without changing the training predictions in function space.*

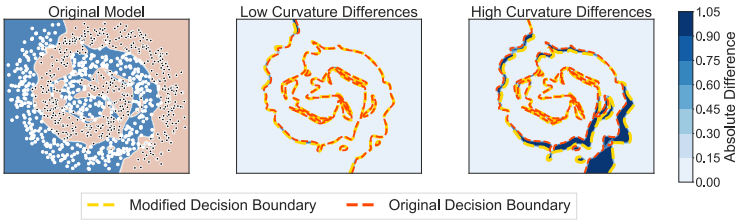


Figure 6: Swiss roll data. **Left:** Adam trained feed-forward, fully connected classifier. **Center:** Differences in original and perturbed classifier when parameters are perturbed in low curvature degenerate directions. **Right:** Differences in the original and perturbed classifier when parameters are perturbed in high curvature directions. **Note** the perturbation in the center plot is approximately 100 *times* the size of that of the plot on the right!

The proof (along with an extension to generalized linear models) is given in Appendix F.2. Although there may be large regions in parameter-space that lead to low-loss models, many of these models may be homogeneous in function space. We can interpret Theorem 4.2 in terms of the eigenvectors of the Hessian indicating which directions in parameter space have and have not been determined by the data. The dominant eigenvectors of the Hessian (those with the largest eigenvalues) correspond to the directions in which the parameters have been determined from the data and the posterior has contracted significantly from the prior. The minimal eigenvectors (those with the smallest eigenvalues) correspond to the directions in parameter space in which the data has not determined the parameters.

Figure 6 demonstrates that the results of Theorem 4.2 apply to neural networks as well, with large perturbations to the parameters in undetermined directions having little effect on the classification boundary. As parameters are perturbed in directions that have not been determined by the data (minimal eigenvectors of the Hessian), the functional form remains identical to the MAP estimate. Perturbations in determined directions (dominant eigenvectors of the Hessian) yield models that perform poorly on the training data and significantly deviate from the MAP estimate on the test set. The homogeneity found by taking perturbations in undetermined parameter directions shows that even in cases with a lack of determination in parameter space, the *functional form* of the model is well specified by the data. Further results regarding function space homogeneity and the connection to loss surface structure, including the projection of the loss surface of the classifier shown in Figure 6 onto random, low curvature, and high curvature bases, are given in Appendix A.

4.3 Effective Dimensionality as Compression

In Section 4.2 we show that effective dimensionality relates to the number of parameter directions in which the functional form of the model is sensitive to perturbations, and that in the low curvature directions given by the Hessian eigenvectors corresponding to the smallest eigenvalues the model outputs are largely unchanged by perturbations to the parameters. The presence of these degenerate directions suggests that we can disregard high dimensional subspaces that contain little information about the model, for compression into a smaller subspace containing only the most important parameter directions given by the eigenvalues of the Hessian. This observation helps explain the practical success of such subspace approaches (Izmailov et al., 2019; Li et al., 2018).

We can also understand the compression of the data provided by a model in terms of minimum description length, by examining the *Occam factor* of the model evidence (MacKay, 2003, Chapter 28). For model $\mathcal{M}$ with parameters θ, we find the Occam factor in decomposing the evidence as,

$$\begin{aligned} p(\mathcal{D}|\mathcal{M}_i) &\approx \underbrace{p(\mathcal{D}|\theta_{MP}, \mathcal{M})}_{\text{Likelihood}} \times \underbrace{p(\theta_{MP}|\mathcal{M})\det{}^{-\frac{1}{2}}(\mathcal{H}_\theta/2\pi)}_{\text{Occam Factor}}, \\ \text{Evidence} &\approx \quad \text{Likelihood} \quad \times \quad \text{Occam Factor} \end{aligned} \tag{3}$$

in which $\mathcal{H}_\theta$ is the Hessian of the loss, and θ_{MP} is the MAP estimate of the parameters.

As the eigenvalues of the Hessian decay and the effective dimensionality decreases, the determinant of the Hessian also decreases, causing the Occam factor to increase, and the description length to decrease, providing a better compression. MacKay (1992b) and MacKay (2003) contains a further discussion of the connection between Occam factors and minimum description length.

5 DOUBLE DESCENT AND WIDTH-DEPTH TRADE-OFFS

5.1 DOUBLE DESCENT

Thus far, we have simply demonstrated that effective dimensionality gives a better understanding of diversity in the context of the space of functions induced by neural networks. The phenomenon of *double descent* of the generalization performance in both linear and deep models has attracted recent attention (Belkin et al., 2019a; Nakkiran et al., 2020); here, we explain double descent as a function of effective dimensionality.

Following the experimental setup of Nakkiran et al. (2020), we train ResNet18s (He et al., 2016) with varying width parameters, reproducing their double descent curve.[1] In Figure 1 we see effective dimensionality tracks remarkably well with generalization — displaying the double descent curve that is seen in the test error. We emphasize again that the effective dimensionality is computed using solely the training data, supporting the hypothesis that the eigenvalues of the Hessian matrix can provide a good proxy for generalization performance. In Appendix H, we show similar findings for linear models and small neural networks.

In short, double descent is an artifact of overfitting. As the dimensionality of the parameter space continues to increase past the point where the corresponding models achieve zero training error, flat regions of the loss occupy a greatly increasing volume (Huang et al., 2019), and are thus more easily discoverable by optimization procedures such as SGD. These solutions have lower effective dimensionality, and thus provide better compressions of the data, as in Section 4.3, and therefore better generalization. In concurrent work, Wilson and Izmailov (2020) show that exhaustive Bayesian model averaging over multiple modes eliminates double descent.

In addition to test loss, we also demonstrate that effective dimensionality tracks double descent in *test error* in Figure A.14 (Appendix). Double descent is clearly present for test loss on CIFAR-100, but not test error. To produce double descent for test error, we follow the setup in Nakkiran et al. (2020) and introduce 20% label corruption, which increases the chance of overfitting.

5.2 NETWORKS OF VARYING WIDTH AND DEPTH

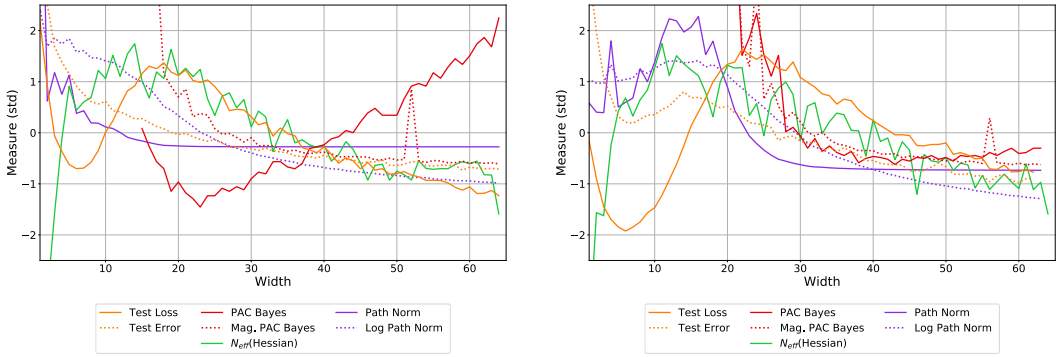
Double descent experiments typically only consider increases in width. However, it is *depth* which has endowed neural networks with distinctive generalization properties. In Figure 2, we consider varying both the width and depth of a convolutional neural network on the CIFAR-100 dataset. We measure effective dimensionality, training loss, and testing loss. In the region of near-zero training loss, separated by the green curve, we see effective dimensionality closely matches generalization performance. Moreover, wide but shallow models tend to overfit, providing low training loss, but high effective dimensionality and test loss. On the other hand, deeper models have lower test loss and lower effective dimensionality, showing that depth enables a better compression of the data.

6 EFFECTIVE DIMENSIONALITY AS A GENERALIZATION MEASURE

We have shown, for the first time, that a generalization measure is able to track and explain double descent and width-depth trade-offs in modern deep networks. We now demonstrate that effective dimensionality is a strong generalization measure by itself. For comparison, we choose path-norm and PAC-Bayesian based sharpness measures, due to their good performance in prior work (Jiang et al., 2020; Keskar et al., 2017; Neyshabur et al., 2017). We compute the norms as in Jiang et al. (2020) for a neural network function $f(x; \boldsymbol{\theta})$ with input x and weights $\boldsymbol{\theta}$.

The path-norm describes the norm of all paths within a network between input and output nodes, measuring the complexity of traversing the network. The standard PAC-Bayesian flatness measure aims to measure the overall flatness of the solution by determining the size of the largest perturbation that can be made to the parameters such that the expected increase in training error remains low. We additionally compare to the magnitude aware PAC-Bayesian flatness measure, which is similar to the standard PAC-Bayesian flatness measure, except that the perturbations to parameters are scaled by the magnitude of the weights in the network. Computation details are given in Appendix C.

[1] See Appendix G for training details.



(a) *Double Descent, No Label Noise* (b) *Double Descent, 20% Label Noise*

Figure 7: Comparing effective dimensionality as a generalization measure to the path-norm, log path-norm, PAC-Bayes, and magnitude aware PAC-Bayes flatness measures from Jiang et al. (2020) for double descent (a) without and (b) with label noise. For models with low train loss, effective dimensionality most closely follows both test error and test loss.

We extend the results of Figure 1 in Figure 7 to include the path-norm and PAC-Bayesian flatness measures. We display test loss, test error, and generalization measures, standardized by subtracting the sample mean and dividing by the sample standard deviation. We extend the results of Figure 2 in the same fashion as Figure 7 in the Appendix in Figure A.12. We additionally show correlations with generalization in Tables 1 and 2 in the Appendix.

There are several key take-aways: (1) effective dimension overall provides a better proxy for test loss and test error than the PAC-Bayes or path-norm measures for double descent and width-depth trade-offs; (2) not all flatness-based generalization measures are equivalent, and in fact different flatness measures can provide wildly different behaviour; (3) effective dimension is more stable than the other measures, with relatively consistent behaviour across qualitatively similar datasets and models; (4) effective dimension is *interpretable*, providing a clear connection with function-space and an explanation for *why* a model should generalize, in addition to an association with generalization.

Not all flatness-based generalization measures are equivalent. In contrast to the PAC-Bayesian flatness measure considered in Jiang et al. (2020), effective dimension computes the number of directions in parameter space that are flat, as determined by the curvature of the loss surface. Combined with our observations about function-space homogeneity in Section 4.2, we see that effective dimension provides a proxy for model compression and generalization in addition to providing correlation with performance. By contrast, the PAC-Bayes flatness measure considers the size of the basin surrounding an optimum, and is highly sensitive to the sharpest direction.

7 Conclusion

We have shown how effective dimensionality can be used to gain insight into a range of phenomena, including double descent, posterior contraction, loss surface structure, subspace inference, and function-space diversity of models. As we have seen, simple parameter counting can be a misleading proxy for model complexity and generalization performance; models with many parameters combined with a particular functional form can give rise to simple explanations for data. Indeed, we have seen how depth and width have different effects on generalization performance, regardless of the total number of parameters in the model. In all cases, effective dimensionality tracks generalization performance for models with comparable training loss. Moving forward, we hope our work will help inspire a continued effort to capture the nuanced interplay between the statistical properties of parameter space and function space in understanding generalization behaviour, and highlight effective dimension as a relatively simple and robust generalization measure in modern deep learning.

REFERENCES

A. Achille and S. Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler. Benign overfitting in linear regression. *arXiv preprint arXiv:1906.11300*, 2019.

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32): 15849–15854, 2019a.

M. Belkin, D. Hsu, and J. Xu. Two models of double descent for weak features. *arXiv preprint arXiv:1903.07571*, 2019b.

E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.

S. Bös, W. Kinzel, and M. Opper. Generalization ability of perceptrons with continuous outputs. *Physical Review E*, 47(2):1384, 1993.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

A. Caponnetto and E. D. Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007. doi: 10.1007/s10208-006-0196-8. URL https://doi.org/10.1007/s10208-006-0196-8.

P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina. Entropy-SGD: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019.

W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836, 1979.

E. Dobriban and S. Wager. High-dimensional asymptotics of prediction: Ridge regression and classification. *Ann. Statist.*, 46(1):247–279, 02 2018. doi: 10.1214/17-AOS1549. URL https://doi.org/10.1214/17-AOS1549.

G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Uncertainty in Artificial Intelligence (UAI)*, 2017.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*. Springer series in statistics New York, 2001.

K. Fukumizu, S. Yamaguch, Y.-i. Mototake, and M. Tanaka. Semi-flat minima and saddle points by embedding neural networks to overparameterization. In *Advances in neural information processing systems*, 2019.

J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

B. Ghorbani, S. Krishnan, and Y. Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, 2019. URL arXiv:1901.10159.

S. F. Gull. Developments in maximum entropy data analysis. In *Maximum entropy and Bayesian methods*, pages 53–71. Springer, 1989.

T. Hastie and R. Tibshirani. *Generalized additive models*. Chapman and Hall, London, 1990.

T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani. Surprises in High-Dimensional Ridgeless Least Squares Interpolation. *arXiv:1903.08560*, Nov. 2019.

K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

G. E. Hinton and D. Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on computational learning theory*, pages 5–13, 1993.

S. Hochreiter and J. Schmidhuber. Flat Minima. *Neural Computation*, 9(1):1–42, 1997. URL https://www.bioinf.jku.at/publications/older/3304.pdf.

M. D. Hoffman and A. Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

W. R. Huang, Z. Emam, M. Goldblum, L. Fowl, J. K. Terry, F. Huang, and T. Goldstein. Understanding generalization through visualizations. *arXiv preprint arXiv:1906.03291*, 2019.

P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. In *Uncertainty in Artificial Intelligence*, 2018. URL arXiv:1803.05407.

P. Izmailov, W. J. Maddox, P. Kirichenko, T. Garipov, D. Vetrov, and A. G. Wilson. Subspace inference for bayesian deep learning. *Uncertainty in Artificial Intelligence (UAI)*, 2019.

Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations (ICLR)*, 2020. URL arXiv:1912.02178.

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations (ICLR)*, 2017. URL arXiv:1609.04836.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR) arXiv:1412.6980*, 2015.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541. URL https://doi.org/10.1162/neco.1989.1.4.541.

Y. LeCun, I. Kanter, and S. A. Solla. Eigenvalues of covariance matrices: Application to neural-network learning. *Physical Review Letters*, 66(18):2396, 1991.

C. Li, H. Farkhoor, R. Liu, and J. Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

D. J. MacKay. Bayesian model comparison and backprop nets. In *Advances in neural information processing systems*, pages 839–846, 1992a.

D. J. MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992b.

D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

W. J. Maddox, T. Garipov, P. Izmailov, D. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in neural information processing systems*, 2019.

D. Madras, J. Atwood, and A. D'Amour. Detecting extrapolation with local ensembles. *arXiv preprint arXiv:1910.09573*, 2019.

S. Mei and A. Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

J. E. Moody. Note on generalization, regularization and architecture selection in nonlinear learning systems. In *Neural Networks for Signal Processing Proceedings of the 1991 IEEE Workshop*, pages 1–10. IEEE, 1991.

J. E. Moody. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In *Advances in neural information processing systems*, pages 847–854, 1992.

V. Muthukumar, K. Vodrahalli, and A. Sahai. Harmless interpolation of noisy data in regression. *arXiv preprint arXiv:1903.09139*, 2019.

P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations (ICLR)*, 2020. URL arXiv:1912.02292.

R. M. Neal and J. Zhang. High Dimensional Classification with Bayesian Neural Networks and Dirichlet Diffusion Trees. In I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, editors, *Feature Extraction: Foundations and Applications*, Studies in Fuzziness and Soft Computing, pages 265–296. Springer, Berlin, Heidelberg, 2006. ISBN 978-3-540-35488-8. doi: 10.1007/978-3-540-35488-8_11. URL https://doi.org/10.1007/978-3-540-35488-8_11.

J. A. Nelder and R. W. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2018.

M. Opper, J. Kleinz, H. Kohler, and W. Kinzel. Basins of attraction near the critical storage capacity for neural networks with constant stabilities. *Journal of Physics A: Mathematical and General*, 22 (9):L407, 1989.

M. Opper, W. Kinzel, J. Kleinz, and R. Nehl. On the ability of the optimal perceptron to generalise. *Journal of Physics A: Mathematical and General*, 23(11):L581, 1990.

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

D. Phan, N. Pradhan, and M. Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*, 2019.

L. Sagun, L. Bottou, and Y. LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. In *ICLR Workshop track*, 2017. URL arXiv:1611.07476.

N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017. URL arXiv:1701.06538.

S. L. Smith and Q. V. Le. A bayesian perspective on generalization and stochastic gradient descent. In *International Conference on Learning Representations (ICLR)*, 2018. URL arXiv:1710.06451.

T. Suzuki. Fast generalization error bound of deep learning from a kernel perspective. In A. Storkey and F. Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1397–1406, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

A. G. Wilson and P. Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations (ICLR)*, 2017. URL arXiv:1611.03530.

T. Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, 17(9):2077–2098, 2005.

W. Zhou, V. Veitch, M. Austern, R. P. Adams, and P. Orbanz. Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach. In *International Conference on Learning Representations (ICLR)*, 2018.