Low Intrinsic Dimension Implies Generalization

Noah MacAulay
usernameneeded@gmail.com

Abstract

Neural networks' ability to generalize to unseen data has proven difficult to explain by statistical learning theory. Recent works have used PAC-Bayes bounds to derive non-trivial generalization bounds for deep neural networks applied to realistic problems. In other works, it has been found that neural networks can obtain high performance even when projected onto random low-dimensional subspaces of their parameter space. In this paper I use this property and PAC-Bayes theory to obtain the tightest generalization bounds yet for neural networks applied to MNIST.

1 Introduction

Everybody knows that neural networks have proven astonishingly successful at a wide range of tasks. But until recently, nobody has had much idea why they work so well, and theory has lagged far behind practice. Two questions stand out: why are neural networks able to learn by gradient descent, and avoid local minima? And why do the resulting models, highly overparameterized as they are, generalize to new examples?

One thread of research has shown that neural networks can be modelled as essentially linear systems in the infinite-width limit. The Neural Tangent Kernel[4] precisely describes the evolution of the network's input-output function in this limit. For networks of finite width, recent theoretical works have also shown that gradient descent will obtain zero training error under broad assumptions.];lppp

A remaining piece of the puzzle is generalization. Even if neural networks are able to obtain zero error on their training set, this does not imply they will generalize to new examples. Traditional tools of statistical learning theory give trvial bounds when applied to highly-overparametrized neural networks [7]. Here, too, there has been progress. Dzuigaite and Roy[2] obtained the first non-vacuous generalization bounds for modern neural networks by optimizing a PAC-Bayesian bound over a stochasticized version of the network. They obtained non-trivial bounds for a binarized version of MNIST. A later paper[8] obtained non-vacuous generalization bounds for full MNIST and ImageNet using PAC-Bayesian theory and a compression approach, whereby the generalization error of a model is bounded by the length of a code needed to specify it. A further work by Dzuigaite and Roy [3] obtained generalization bounds for neural networks using data-dependent priors, however, their bounds only hold contingent upon an assumption about the convergence behaviour of SGLD.

Following this line of work, in this paper I will present the tighest generalization bounds yet for full MNIST. My bounds will be based on the idea of the intrinsic dimension of a learning problem, introduced in [5]. Intrinsic dimension measures a surprising property of neural networks: learning can occur even when the parameters of the network are projected to a random low-dimensional subspace. In [5] it was shown that for neural networks applied to realistic problems, the dimensionality of the parameters could be reduced by several orders of magnitude while preserving good performance. This implies the resulting models are highly compressible; by applying PAC-Bayesian theory to these compressed models, it is possible to obtain strong generalization bounds.

Preprint. Under review.

2 PAC-Bayesian Bounds

PAC-Bayes bounds are a method for proving generalization of supervised learning models. Here I provide a brief overview of PAC-Bayesian theory.

PAC-Bayes bounds apply to supervised learning problems. A supervised learning problem consists of an input space X, an output space Y, and an unknown probability distribution D over $X \times Y$. Given a loss function $\ell(\hat{y}, y) \to \mathbb{R}$, and a sample $S = (x_i, y_i)_{i=1}^N$ drawn identically and independently distributed from D, the goal of supervised learning is to choose a hypothesis $h \in H$ which minimizes the expected loss $\ell(h) = \mathbb{E}_{(x,y)}[\ell(h(x), y)]$

A common approach to supervised learning is minimizing the empirical risk $\hat{\ell}(h) = \frac{1}{N}\sum_{i=1}^N \ell(h(x_i), y_i)$. This amounts to treating the sample S as a proxy for the true distribution D. This introduces the danger of overfitting, choosing a model which performs well on the sample S but poorly on the underlying distribution D. The generalization error of a hypothesis h is $\ell(h) - \hat{\ell}(h)$

PAC-Bayes theory was introduced by MacAllester in [6]. PAC-Bayes theorems can be used to place upper bounds on the generalization error of a model, ensuring that it will perform well on new data. PAC-Bayes bounds generally apply to stochastic classifiers, represented as a probability distribution Q over the space of classifiers H. A prior distribution P is also introduced, independent of the sample. This prior distribution P The tighest version of the PAC-Bayes theorem yet given is due to Catoni[1]:

Let ℓ be a 0, 1-valued loss function and P be a distribution on the hypothesis class H, let $\alpha > 1$, $\epsilon > 0$ be fixed. With probability at least $1 - \epsilon$ over the sample:

$$L(Q) \leq \inf_{\gamma > 1} \Phi_{\gamma/N}^{-1}\{\hat{L}(Q) + \frac{\alpha}{\gamma}\langle KL(Q, P) - \log(\epsilon) + 2\log\left(\frac{\log(\alpha^2\gamma)}{\log(\alpha)}\right)\rangle\}$$

where Φ^{-1} is defined as:

$$\Phi_\gamma^{-1}(x) = \frac{1 - e^{-\gamma x}}{1 - e^{-\gamma}}$$

If $\hat{L}(Q)$ and $KL(Q, P)$ are differentiable, then this bound can be optimized by gradient descent and similar methods. While we will ultimately be interested in applying the bound to accuracy(the 0-1 loss), to facilitate this optimization will be performed using a differentiable proxy(in our case the cross-entropy loss).

3 Random Projections

Here I discuss the method of random projections, introduced in [5] as a method for measuring the "intrinsic dimension" of a learning problem. A neural network can be represented as a parameter vector $\theta \in \mathbb{R}^D$, initialized at some θ_0. Modern neural networks can have D in the range of thousands to billions. During training, a random projection matrix $P \in \mathbb{R}^{D \times D}$ of rank k is chosen, and all gradients $\nabla\theta$ are replaced with their projection $P\nabla\theta$. Choosing a basis $B \in \mathbb{R}^{D \times k}$ for the span of P, the final parameter vector θ_f can be represented by θ_0 and a vector $v \in \mathbb{R}^k$, $\theta_f = \theta_0 + Bv$

Importantly, because the random projection matrix P and the initial parameter vector θ_0 have no dependence on the training sample, they can be used in the definition of a PAC-Bayes prior over neural networks lying in this random subspace.

A variety of approaches could be taken to obtain a PAC-Bayes prior for randomly-projected neural networks. Here I will follow Dzuigaite and Roy [2] and use a stochasticized version of the network. Thus, the parameter vector of the network will be represented by $\theta = \theta_0 + Bv$, where v is distributed $v = \mathcal{N}(\mu_q, \Sigma_q)$, where Σ_q is a diagonal matrix. The prior distribution over v is given by $\mathcal{N}(0, \lambda I)$. The KL divergence between our learned posterior for v and the prior can be easily computed using standard formulae.

It will be useful to be able to optimize over the prior parameter λ. To accommodate this a uniform prior will be placed over the 2^{32} possible floating point values of λ, adding 32 to the KL term. In sum, the variables to be optimized over are μ_q, the diagonal entries of Σ_q, γ and λ.

4 Empirical PAC-Bayes Bounds

4.1 MNIST

Here I report the performance of my bound on MNIST. The network architecture was a standard LeNet, with two stacks of convolution and max-pooling followed by 3 fully-connected layers. ReLU nonlinearities were used. The network was trained in the projected subspace for 20 epochs to a good minimum, using SGD with learning rate 0.01. Following this, PAC-Bayes optimization began. The posterior means were initialized at the learned values and the posterior variances were initialized at e^{-3}. For stability, the logarithms of the posterior variances were optimized. Optimization was done over the PAC-Bayes objective with the cross-entropy loss serving as a differentiable proxy for $\hat{L}(Q)$, using SGD for 20 epochs.

The dimension of the random projection was chosen to be 1000.The random projection matrix P was constructed by constructing matrix $R \in \mathbb{R}^{D \times D}$ by choosing independent Gaussians for every entry, then taking the SVD $R = USV$. The first k rows of the orthogonal matrix U were used as the basis for B, the basis of the random projection.

A final PAC-Bayes bound for the accuracy of 0.845 was obtained, holding with probability 0.95. The accuracy on the test set was 0.95. The best-known generalization bound in the literature is 0.54 [8]. While a gap still remains between generalization bounds and test performance, it is significantly smaller.

5 Conclusion

In this paper I have demonstrated a connection between the concept of intrinsic dimension and generalization bounds. As shown in [1]GeneralizationCompression, PAC-Bayes priors are closely related to efficient ways of compressing neural networks. Thus the current work provides some evidence that random projections can be a powerful way of compressing the useful structure of neural networks. However, for more challenging problems, a significant gap between generalization bounds and accuracy remains. In the future, hopefully new and powerful ways of distilling the structure in neural networks can be found, leading to both greater understanding and improved generalization bounds.

References

[1] Olivier Catoni. Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007.

[2] 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.

[3] Gintare Karolina Dziugaite and Daniel M Roy. Entropy-sgd optimizes the prior of a pac-bayes bound: Generalization properties of entropy-sgd and data-dependent priors. *arXiv preprint arXiv:1712.09376*, 2017.

[4] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[5] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

[6] David A McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

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

[8] Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach. *arXiv preprint arXiv:1804.05862*, 2018.