A Bayesian Perspective on Generalization and Stochastic Gradient Descent

Samuel L. Smith and Quoc V. Le
Google Brain
{slsmith,qvl}@google.com

1 Introduction

We tackle two related questions at the heart of machine learning; how can we predict if a minimum will generalize to the test set, and why does stochastic gradient descent (SGD) find minima which generalize well? Our work is inspired by Zhang et al. (2016), who showed deep networks can easily memorize randomly labeled training data, despite generalizing well when shown real labels of the same inputs. They argued that these results are inconsistent with classical statistical learning theory. We show that the same phenomenon occurs in linear models, and that it can be explained by evaluating the Bayesian evidence, or "marginal likelihood", which penalizes sharp minima. Next, we explore the "generalization gap" between small and large batch training (Keskar et al., 2016), identifying an optimum batch size which maximizes the test set accuracy. Intuitively, noise in the gradient updates drives the dynamics towards robust minima whose evidence is large. Interpreting SGD as a stochastic differential equation, we predict the optimum batch size is proportional to both the learning rate and the size of the training set, $B_{opt} \propto \epsilon N$. A complete version of this work is available on arXiv (Smith & Le, 2017), where we verify these predictions empirically.

2 Bayesian Model Comparison

Bayesians assess the quality of a model M by evaluating the evidence, $P(\{y\}|\{x\};M) = \int d\omega \, P(\{y\}|\omega,\{x\};M)P(\omega;M)$, where ω denotes the parameters, $\{x\}$ the training inputs and $\{y\}$ the training labels (MacKay, 1992). We use a Gaussian prior $P(\omega;M) = \sqrt{\lambda/2\pi}e^{-\lambda|\omega|^2/2}$. Under standard approximations, this evidence, $P(\{y\}|\{x\};M) = \lambda^{\frac{P}{2}} e^{-C(\omega_0)}/|\nabla\nabla C(\omega)|_{\omega_0}^{1/2}$, where ω_0 is the minimum of the cost function $C(\omega) = H(\omega) + \lambda|\omega|^2/2$, $H(\omega)$ is the training set cross-entropy, $|\nabla\nabla C(\omega)|_{\omega_0}$ the determinant of the Hessian, and P the number of model parameters. The determinant of a matrix is simply the product of its eigenvalues, $\left(\prod_{i=1}^{P} \lambda_i\right)$, and thus,

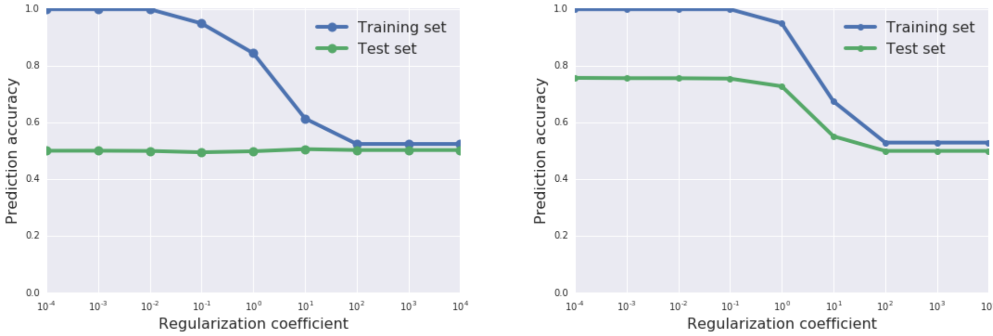
$$P(\{y\}|\{x\};M) \approx \exp\left\{-\left(C(\omega_0) + \frac{1}{2}\sum_{i=1}^{P} \ln(\lambda_i/\lambda)\right)\right\}. \tag{1}$$

The contribution $(\lambda^{\frac{P}{2}}/|\nabla\nabla C(\omega)|_{\omega_0}^{1/2})$ is the "Occam factor". Intuitively, it describes the fraction of the prior parameter space consistent with the data. Since this fraction is always less than one, we will approximate equation 1 away from local minima by only performing the summation over eigenvalues $\lambda_i \geq \lambda$. In this work, we compare our models against a null model which assigns equal probability to all classes. Thus $P(\{y\}|\{x\};NULL) = (1/n)^N = e^{-N\ln(n)}$, where n denotes the number of model classes and N the number of training examples. The evidence ratio,

$$\frac{P(\{y\}|\{x\};M)}{P(\{y\}|\{x\};NULL)} = e^{-E(\omega_0)}, \tag{2}$$

Where $E(\omega_0) = C(\omega_0) + (1/2)\sum_i \ln(\lambda_i/\lambda) - N\ln(n)$ is the log evidence ratio in favor of the null model. Clearly, we should only assign any confidence to the predictions of a model if $E(\omega_0) < 0$. The evidence penalizes sharp minima, but is invariant to model parameterization (Dinh et al., 2017).

Second workshop on Bayesian Deep Learning (NIPS 2017), Long Beach, CA, USA.



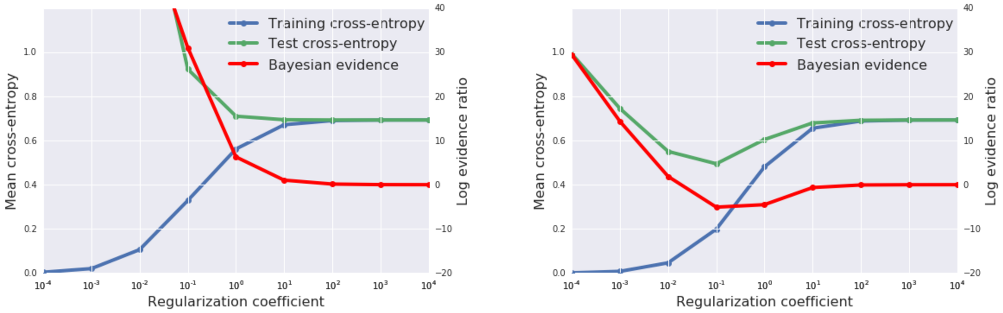
(a) (b)

Figure 1: Prediction accuracy as a function of regularization coefficient, for a logistic regression trained on random (a) and informative (b) labels. Exactly as observed by Zhang et al., weakly regularized logistic regression generalizes well on informative labels but memorizes random labels.

3 Bayes Theorem and Generalization

We define two toy tasks, for which our inputs are randomly generated 200D vectors x, and each component x_i is drawn from the Gaussian distribution. We normalize the inputs, to ensure $\langle x_i \rangle = 0$ and $\langle x_i^2 \rangle = 1$, and we train using the sigmoid cross-entropy loss. In the first task the labels are random, $y = \{0, 1\}$. In the second task, the label $y = 1$ if $\sum_i x_i > 0$, while $y = 0$ otherwise. Our model is an L2 regularized logistic regression, our training set contained 200 examples, and our test set contained 10000 examples. We show the accuracy of our model predictions on both the training and test sets in figure 1. When trained on the informative labels, the model generalizes well to the test set, so long as it is sufficiently weakly regularized. However the model also perfectly memorizes the random labels, replicating the observations of Zhang et al. (2016) in deep networks.

These results are inconsistent with statistical learning theory (Zhang et al., 2016), but they are explained by Bayesian model comparison. Consider figure 2, where we plot the mean cross-entropy of the model predictions, evaluated on both training and test sets, as well as the Bayesian log evidence ratio. The training cross-entropy and the test cross-entropy are uncorrelated, but the test cross-entropy is strongly correlated with the log evidence ratio. When trained on random labels, the log evidence ratio is always positive, indicating that we never expect the model to generalize. Meanwhile on informative labels, the log evidence ratio is negative when the model is well regularized, indicating that we expect the model to generalize well. Bayesian model comparison has successfully explained the generalization of our logistic regression, and there is no reason to believe that it would not also explain the generalization of deep networks. The evidence for deep networks is hard to calculate, but Krueger et al. (2017) showed that the largest Hessian eigenvalue increases with random labels.



(a) (b)

Figure 2: The cross-entropy and log evidence ratio, evaluated on random (a) or informative (b) labels. The evidence, evaluated on the training set, is highly correlated with the test cross-entropy.

4 Bayes Theorem and Stochastic Gradient Descent

In the proceeding section, we showed that generalization is strongly correlated with the Bayesian evidence, which is a weighted combination of the depth of a minimum (the cost function) and its breadth (the Occam factor). Therefore, when we train deep networks, we should not minimize the cost function; we should try to find the local minimum which maximizes the evidence. To achieve this, Bayesians add noise to the gradient updates (Mandt et al., 2017; Welling & Teh, 2011). We propose that the same principles account for the “generalization gap" (Keskar et al., 2016), whereby the test accuracy often falls as the SGD batch size is increased. Small batch training introduces noise to the gradients, and this noise drives the SGD away from sharp minima, thus enhancing generalization. Since the gradient drives the SGD towards deep minima, while noise drives the SGD towards broad minima; we expect the test set performance to be maximized at an optimal batch size, which introduces the right amount of noise to balance these competing contributions to the evidence.

To formalize this intuition, we interpret SGD as a stochastic differential equation. A gradient update,

$$\Delta\omega \;=\; \frac{\epsilon}{N}\left(\frac{dC}{d\omega}+\left(\frac{d\hat{C}}{d\omega}-\frac{dC}{d\omega}\right)\right), \tag{3}$$

Where ϵ is the learning rate, N is the training set size, $\frac{dC}{d\omega}=\sum_{i=1}^{N}\frac{dC_i}{d\omega}$ is the true gradient, and $\frac{d\hat{C}}{d\omega}=\frac{N}{B}\sum_{i=1}^{B}\frac{dC_i}{d\omega}$ is the gradient evaluated on a batch. The expected gradient of a single training example, $\left\langle\frac{dC_i}{d\omega}\right\rangle=\frac{1}{N}\frac{dC}{d\omega}$, while $\left\langle\frac{dC_i}{d\omega}\frac{dC_j}{d\omega}\right\rangle=\left(\frac{1}{N}\frac{dC}{d\omega}\right)^2+F(\omega)\delta_{ij}$. $F(\omega)$ is a matrix describing the gradient covariances between different parameters, which are themselves a function of the current parameter values. To proceed, we adopt the central limit theorem and model the gradient error $\alpha=\left(\frac{d\hat{C}}{d\omega}-\frac{dC}{d\omega}\right)$ with Gaussian random noise. It is easy to show that $\langle\alpha\rangle=0$, while $\langle\alpha^2\rangle=N(\frac{N}{B}-1)F(\omega)$. Typically $N\gg B$, such that $\langle\alpha^2\rangle\approx N^2F(\omega)/B$. To continue, we interpret equation 3 as a discrete update of the stochastic differential equation (Øksendal, 2003),

$$\frac{d\omega}{dt}=\frac{dC}{d\omega}+\eta(t), \tag{4}$$

Where t is a continuous variable, $\eta(t)$ represents noise, $\langle\eta(t)\rangle=0$ and $\langle\eta(t)\eta(t')\rangle=gF(\omega)\delta(t-t')$. The constant g controls the scale of random fluctuations in the dynamics. To relate this differential equation to the SGD, we compute a single gradient update $\Delta\omega=\int_0^{\epsilon/N}\frac{d\omega}{dt}dt=\frac{\epsilon}{N}\frac{dC}{d\omega}+\int_0^{\epsilon/N}\eta(t)dt$. Finally, to measure g, we equate the variance in this gradient update to the variance in the gradient,

$$\begin{aligned}\left(\frac{\epsilon}{N}\right)^2\langle\alpha^2\rangle \;&=\; \epsilon^2(\frac{N}{B}-1)F(\omega)/N\\ &=\; \langle(\int_0^{\epsilon/N}dt\,\eta(t))^2\rangle=\int_0^{\epsilon/N}dt\int_0^{\epsilon/N}dt'\,\langle\eta(t)\eta(t')\rangle=\epsilon gF(\omega)/N.\end{aligned} \tag{5}$$

Rearranging, $g=\epsilon(\frac{N}{B}-1)\approx\epsilon N/B$. SGD simulates a stochastic differential equation, and the scale of random fluctuations are inversely proportional to the batch size. We expect an optimal batch size to emerge when the underlying scale of random fluctuations is also optimal. We note that the SGD would perform Bayesian posterior sampling if the covariance matrix $F(\omega)$ were an identity matrix. Although it is not, it will have similar effects on the dynamics, driving SGD away from sharp minima for which the evidence is small. In Smith & Le (2017), we extend the treatment above to SGD with momentum, obtaining the noise scale $g\approx\epsilon N/(B(1-m))$, where m is the momentum coefficient.

If we change the learning rate, momentum coefficient or training set size, Bayesian intuition tells us that the scale of random fluctuations in the dynamics should not change. This scale is defined by the coefficient g, which implies that the optimum batch size, $B_{opt}\propto\epsilon N/(1-m)$. We note that Goyal et al. (2017) observed the scaling rule $B\propto\epsilon$ empirically, and used it to increase the batch size without sacrificing test set accuracy. This enabled them to train ImageNet in one hour.

In Smith & Le (2017), we confirm the existence of an optimal batch size within a shallow neural network trained on MNIST. This optimum maximizes the test accuracy. We also successfully observe the three scaling rules predicted above, $B_{opt}\propto\epsilon$, $B_{opt}\propto N$ and $B_{opt}\propto 1/(1-m)$. These scaling rules demonstrate that large batch training is possible, increasing parallelism and reducing model training times. We extend our analysis of SGD to include decaying learning rate schedules in Smith et al. (2017), where we show that decaying the learning rate is equivalent to increasing the batch size.

Acknowledgments

We thank Pieter-Jan Kindermans, Prajit Ramachandran, Jascha Sohl-Dickstein, Jon Shlens, Kevin Murphy, Samy Bengio, Yasaman Bahri and Saeed Saremi for helpful comments.

References

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*, 2017.

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.

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.

David Krueger, Nicolas Ballas, Stanislaw Jastrzebski, Devansh Arpit, Maxinder S Kanwal, Tegan Maharaj, Emmanuel Bengio, Asja Fischer, and Aaron Courville. Deep nets don't learn via memorization. *ICLR Workshop*, 2017.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

Bernt Øksendal. *Stochastic differential equations*. Springer, 2003.

Samuel L. Smith and Quoc V. Le. Understanding generalization and stochastic gradient descent. *arXiv preprint arXiv:1710.06451*, 2017.

Samuel L. Smith, Pieter-Jan Kindermans, and Quoc V. Le. Don't decay the learning rate, increase the batch size. *In press (arXiv preprint coming soon)*, 2017.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 681–688, 2011.

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