ENTROPY-SG(L)D OPTIMIZES THE PRIOR OF A (VALID) PAC-BAYES BOUND

Anonymous authors
Paper under double-blind review

ABSTRACT

We show that Entropy-SGD (Chaudhari et al., 2017), when viewed as a learning algorithm for classifiers, optimizes a PAC-Bayes bound on the risk of the classifier, or more accurately, the Gibbs posterior, i.e., a risk-sensitive perturbation of the classifier. Entropy-SGD works by optimizing the bound's prior, violating the hypothesis of the PAC-Bayes theorem that the prior is chosen independently of the data. Indeed, available implementations of Entropy-SGD rapidly obtain zero training error on random labels and the same holds of the Gibbs posterior. In order to obtain a valid generalization bound, we show that an ε-differentially private prior yields a valid PAC-Bayes bound, a straightforward consequence of results connecting generalization with differential privacy. Using stochastic gradient Langevin dynamics (SGLD) to approximate the well-known exponential release mechanism, we observe that generalization error on MNIST (measured on held out data) falls within the (empirically nonvacuous) bounds computed under the assumption that SGLD produces perfect samples. In particular, Entropy-SGLD can be configured to yield relatively tight generalization bounds and still fit real labels, although these same settings do not obtain state-of-the-art performance.

1 INTRODUCTION

Optimization is central to much of machine learning, but generalization is the ultimate goal. Despite this, the generalization properties of many optimization-based learning algorithms are poorly understood. The standard example is stochastic gradient descent (SGD), one of the workhorse of deep learning, which has good generalization performance in many settings, but rapidly overfits in others (Zhang et al., 2017). Can we develop high performance learning algorithms with provably strong generalization guarantees? Or is there a limit?

In this work, we study an optimization algorithm called Entropy-SGD (Chaudhari et al., 2017), which was designed to outperform SGD in terms of generalization error when optimizing an empirical risk. Entropy-SGD minimizes an objective $f : \mathbb{R}^p \to \mathbb{R}$ indirectly by performing (approximate) stochastic gradient ascent on the so-called local entropy $F(\mathbf{w}) = \log \int \exp(-f(\mathbf{w}+\xi))\mathscr{N}(\mathrm{d}\xi)$, where $\mathscr{N}$ is a zero-mean isotropic multivariate normal distribution on $\mathbb{R}^p$.

Our first contribution is connecting Entropy-SGD to results in statistical learning theory, showing that maximizing the local entropy corresponds to minimizing a PAC-Bayes bound (McAllester, 1999) on the risk of the so-called Gibbs posterior. The distribution of $\mathbf{w}+\xi$ is the PAC-Bayesian "prior", and so optimizing the local entropy optimizes the bound's prior. This connection between local entropy and PAC-Bayes follows from a result due to Catoni (2007, Lem. 1.1.3) in the case of bounded risk. In the special case where f is the empirical cross entropy, the local entropy is literally a Bayesian log marginal density. The connection between minimizing PAC-Bayes bounds and maximizing log marginal densities is the subject of recent work by Germain et al. (2016).

Despite the connection to PAC-Bayes, as well as theoretical results by Chaudhari et al. suggesting that Entropy-SGD may be more stable than SGD, we demonstrate that Entropy-SGD (and its corresponding Gibbs posterior) can rapidly overfit, just like SGD. We identify two changes that suffice to control generalization error.

The first change relates to the stability of optimizing the prior mean. The PAC-Bayes theorem requires that the prior be independent of the data, and so by optimizing the prior mean, Entropy-

SGD invalidates the bound. Indeed, the bound does not hold empirically. While a PAC-Bayes prior may not be chosen based on the data, it can depend on the data distribution. This suggests that if the prior depends only weakly on the data, it may be possible to derive a valid bound.

We formalize this intuition using differential privacy (Dwork, 2006; Dwork et al., 2015b). By truncating the cross entropy loss and replacing SGD with stochastic gradient Langevin dynamics (SGLD; Welling and Teh, 2011), the data-dependent prior mean can be shown to be (ε, δ)-differentially private (Wang, Fienberg, and Smola, 2015; Minami et al., 2016). Using results connecting statistical validity and differential privacy (Dwork et al., 2015b, Thm. 11), we can also show that an ε-differentially private prior mean yields a valid, though slightly expanded, generalization bound using the PAC-Bayes theorem. (We refer to the SGLD variant as Entropy-SG(L)D.)

A gap remains between pure and approximate differential privacy. In the limit as the number of iterations diverges, the distribution of SGLD's output is known to converge weakly to the corresponding stationary distribution. (See recent work by Chen, Ding, and Carin (2015) and references therein.) Weak convergence, however, falls short of implying that SGLD converges to the ε-differentially private exponential release mechanism. We view this as an important open problem. Regardless, we may proceed under the optimistic assumption that SGLD implements an exponential release, and apply our ε-differentially private PAC-Bayes bound. We find that the corresponding 95% confidence intervals are reasonably tight but still conservative in our experiments.

The second change pertains to the stability of the stochastic gradient estimate made on each iteration of Entropy-SG(L)D. This estimate is made using SGLD (hence Entropy-SG(L)D is SG(L)D with a few iterations of SGLD at every step to approximate the gradient). A subtle detail of the SGLD within Entropy-SGD is that the noise added to the gradient is intentionally divided by a factor that ranges from 1000–10000. The result is that the Lipschitz constant of the objective function is 1000–10000 times larger, making Entropy-SGD much less stable as a result. This change to the noise also negatively impacts the differential privacy of the prior mean. Working backwards from the desire to obtain reasonably tight generalization bounds, we are led to instead *multiply* the SGLD noise by a factor of $\sqrt{m}$, where m is the number of data points. The resulting bounds (which assume that SGLD implements an idealized exponential release mechanism), are nonvacuous and tighter than those recently published by Dziugaite and Roy (2017), although it must be emphasized that ours hold subject to an assumption about the privacy of the prior mean, which is certainly violated but to an unknown degree.

We begin with a review of some related work, before introducing sufficient background so that we can make a formal connection between local entropy and PAC-Bayes bounds. We then introduce further background on differential privacy, before introducing a differentially private PAC-Bayes bound. In Section 6, we present experiments on MNIST which provide evidence for our theoretical analysis. (Empirical validation is required in order to address the aforementioned gap between pure and approximate differential privacy.) We close with a short discussion.

2 Related work

A key aspect of our analysis relies on the stability of a data-dependent prior. Stability has long been understood to relate to generalization. (See recent work by Hardt, Recht, and Singer (2015) and references therein.) This work was also inspired in part by observations of Zhang et al. (2017), who studied empirical properties of SGD that were not widely appreciated within the theory community. They show that, without regularization, SGD can achieve zero training error on MNIST and CIFAR, even if the labels are chosen uniformly at random. At the same time, SGD obtains weights with very small generalization error with the original labels. The first observation is strong evidence that the set of classifier accessible to SGD within a reasonable number of iterations is extremely rich. Indeed, with probability almost indistinguishable from one, fitting random labels on a large data set implies that the Rademacher complexity of the hypothesis class is essentially the maximum possible (Bartlett and Mendelson, 2003, Thm. 11).

Similar observations were made by Neyshabur, Tomioka, and Srebro (2014), who argue that implicit regularization underlies the ability of SGD to generalize. Recent work also connects the curvature (or local volume) of the empirical risk surface to generalization (Bartlett, Foster, and Telgarsky, 2017; Neyshabur et al., 2017b; Neyshabur et al., 2017a; Dziugaite and Roy, 2017)

These ideas connect to early work by Hinton and Camp (1993); Hochreiter and Schmidhuber (1997) which introduced regularization schemes based on information theoretic ideas. These ideas, now referred to as "flat minima", can be related to minimizing PAC-Bayes bounds, although these bounds are minimized with respect to the posterior, not the prior, as is done by Entropy-SGD (Dziugaite and Roy, 2017). Achille and Soatto (2017) provides additional information-theoretic arguments for a regularization scheme similar to that of Hinton and Camp. Their objective takes the form of regularized empirical cross entropy

$$\hat{R}_{S_m}(Q) + \beta \mathrm{KL}(Q||P), \tag{1}$$

where Q and P are the prior and posterior on the weights, respectively. For an appropriate range of β, linear PAC-Bayes bounds are exactly of this form. In Achille and Soatto (2017) they empirically observe that varying β correlates with a degree of overfitting on a random label dataset. Their experimental insights agree with our privacy analysis as β directly affects the differential privacy, and thus controls an upper bound on generalization error. In addition, Achille and Soatto (2017) also note the connections with variational inference (Kingma, Salimans, and Welling, 2015).

This work also relates to renewed interest in nonvacuous generalization bounds (Langford, 2002; Langford and Caruana, 2002), i.e., bounds on the numerical difference between the unknown classification error and the training error that are (much) tighter than the tautological upper bound of one. Recently, Dziugaite and Roy (2017) demonstrated nonvacuous generalization bounds for random perturbations of SGD solutions using PAC-Bayes bounds for networks with millions of weights. Their work builds on the core insight demonstrated nearly 15 years ago by Langford and Caruana (2002), who computed nonvacuous bounds for neural networks five orders of magnitude smaller.

The analysis of Entropy-SGLD rests on results in differential privacy (see (Dwork, 2008) for a survey) and its connection to generalization (Dwork et al., 2015b; Dwork et al., 2015a; Bassily et al., 2016; Oneto, Ridella, and Anguita, 2017). Entropy-SGLD can be seen as an instance of differentially private empirical risk minimization, which is well studied, both in the abstract (Chaudhuri, Monteleoni, and Sarwate, 2011; Kifer, Smith, and Thakurta, 2012; Bassily, Smith, and Thakurta, 2014) and in the particular setting of private training via SGD (Bassily, Smith, and Thakurta, 2014; Abadi et al., 2016). Our analysis also rests on the differential privacy of Bayesian and Gibbs posteriors, and approximate sampling algorithms (Mir, 2013; Bassily, Smith, and Thakurta, 2014; Dimitrakakis et al., 2014; Wang, Fienberg, and Smola, 2015; Minami et al., 2016).

Our differentially private PAC-Bayes bound rely on data-distribution-dependent priors. Such bounds were first studied by Catoni (2007) and further studied by Lever, Laviolette, and Shawe-Taylor (2013).

3 Preliminaries: Supervised learning, Entropy-SGD, and PAC-Bayes

Let Z be a measurable space, let $\mathscr{D}$ be an unknown distribution on Z, and consider the batch supervised learning setting under bounded loss: having observed $S \sim \mathscr{D}^m$, i.e., m independent and identically distributed samples from $\mathscr{D}$, we aim to choose a weight vector $\mathbf{w} \in \mathbb{R}^p$ with minimal *risk*

$$R_{\mathscr{D}}(\mathbf{w}) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z\sim\mathscr{D}}(\ell(\mathbf{w}, z)), \tag{2}$$

where $\ell : \mathbb{R}^p \times Z \to \mathbb{R}$ is measurable and bounded below. (We ignore the possibility of constraints on the weight vector for simplicity.) We will also consider weight distributions $Q \in \mathscr{M}_1(\mathbb{R}^p)$, i.e., probability measures on $\mathbb{R}^p$ whose risks are defined via averaging,

$$R_{\mathscr{D}}(Q) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{\mathbf{w}\sim Q}(R_{\mathscr{D}}(\mathbf{w})) = \mathop{\mathbb{E}}_{z\sim\mathscr{D}}\Big(\mathop{\mathbb{E}}_{\mathbf{w}\sim Q}(\ell(\mathbf{w}, z))\Big), \tag{3}$$

where the second equality follows from Fubini's theorem and the fact that ℓ is bounded below.

Let $S = (z_1, \ldots, z_m)$ and let $\hat{\mathscr{D}} \stackrel{\text{def}}{=} \frac{1}{m}\sum_{i=1}^m \delta_{z_i}$ be the empirical distribution. Given a weight distribution Q, such as that chosen by a learning algorithm on the basis of data S, its *empirical risk*

$$\hat{R}_S(Q) \stackrel{\text{def}}{=} R_{\hat{\mathscr{D}}}(Q) = \frac{1}{m}\sum_{i=1}^m \mathop{\mathbb{E}}_{\mathbf{w}\sim Q}(\ell(\mathbf{w}, z_i)), \tag{4}$$

will be studied as a stand-in for its risk, which we cannot compute. While $\hat{R}_S(Q)$ is easily seen to be an unbiased estimate of $R_{\mathscr{D}}(Q)$ when Q is nonrandom or independent of S, our goal is to characterize the *generalization error* $R_{\mathscr{D}}(Q) - \hat{R}_S(Q)$ when Q is random and dependent on S.

One of our focuses will be on classification, where $Z = X \times K$, with K a finite set of classes/labels. A product measurable (in practice, continuous) function $f : \mathbb{R}^p \times X \to K$ maps weight vectors $\mathbf{w}$ to classifiers $f(\mathbf{w}, \cdot) : X \to K$. The loss function is given by $\ell(\mathbf{w}, (x, y)) = g(f(\mathbf{w}, x), y)$ for some $g : K \times K \to \mathbb{R}$. In binary classification, we take $K = \{0, 1\}$ and adopt the 0–1 loss $\ell_{0-1} : \mathbb{R}^p \times Z \to \{0, 1\}$ corresponding to $g(y', y) = 1$ if and only if $y = y'$.

We will also consider parametric families of probability-distribution-valued classifiers $f : \mathbb{R}^p \times X \to [0, 1]^K$. For every input $x \in X$, the output $f(\mathbf{w}, x)$ specifies a probability distribution on K. The standard loss is then the negative log likelihood. Again, in the special case of binary classification, the classifiers may simply output an element of $[0, 1]$, i.e., a probability. The negative log likelihood, also called the binary cross entropy, ℓ_{BCE}, is given by $g(p, y) = -y \log(p) - (1 - y) \log(1 - p)$. Note that the binary cross entropy loss is merely bounded below. We will consider bounded modifications in Appendix B.1.

We will sometimes refer to elements of $\mathbb{R}^p$ and $\mathscr{M}_1(\mathbb{R}^p)$ as classifiers and randomized classifiers, respectively. Likewise, we will often refer to the (empirical) risk as the (empirical) error.

3.1 Entropy-SGD

Entropy-SGD is a gradient-based learning algorithm proposed by Chaudhari et al. (2017) as an alternative to stochastic gradient descent on the (empirical) risk surface $\hat{R}_S$. The authors argue that Entropy-SGD has better generalization performance and demonstrate it empirically. Part of that argument is a theoretical analysis relying on the uniform stability of SGD, although they admit that their analysis rests on unrealistic assumptions.

Entropy-SGD is stochastic gradient ascent applied to the optimization problem:

$$\arg\max_{\mathbf{w} \in \mathbb{R}^p} F_{\gamma,\tau}(\mathbf{w}; S), \quad \text{where } F_{\gamma,\tau}(\mathbf{w}; S) = \log \int_{\mathbb{R}^p} \exp(-\tau \hat{R}_S(\mathbf{w}') - \tau \frac{\gamma}{2} \|\mathbf{w}' - \mathbf{w}\|_2^2) \, \mathrm{d}\mathbf{w}'. \tag{5}$$

The objective $F_{\gamma,\tau}(\cdot; S)$ is known as the *local entropy*, and can be viewed as the log partition function of the distribution $P_{\gamma,\tau}^{\mathbf{w},S}$ defined by the unnormalized density

$$\exp(-\tau \hat{R}_S(\mathbf{w}') - \tau \frac{\gamma}{2} \|\mathbf{w}' - \mathbf{w}\|_2^2). \tag{6}$$

Assuming that one can exchange differentiation and integration, it is straightforward to verify that

$$\nabla_{\mathbf{w}} F_{\gamma,\tau}(\mathbf{w}; S) = \mathop{\mathbb{E}}_{\mathbf{w}' \sim P_{\gamma,\tau}^{\mathbf{w},S}} (\tau \gamma (\mathbf{w} - \mathbf{w}')), \tag{7}$$

and then the local entropy $F_{\gamma,\tau}(\cdot; S)$ is even differentiable, even if the empirical risk $\hat{R}_S$ is not. Indeed, Chaudhari et al. show that the local entropy and its derivative are Lipschitz. Chaudhari et al. argue informally that maximizing the local entropy leads to "flat minima" in the empirical risk surface, which several authors (Hinton and Camp, 1993; Hochreiter and Schmidhuber, 1997; Baldassi et al., 2015; Baldassi et al., 2016) have argued is tied to good generalization performance (though none of these papers actually demonstrate nonvacuous generalization bounds).[1]

Chaudhari et al. propose a Monte Carlo estimate of the gradient,

$$\nabla_{\mathbf{w}} F_{\gamma,\tau}(\mathbf{w}; S) \approx \tau \gamma (\mathbf{w} - \mu_L), \text{ with } \mu_1 = \mathbf{w}_1 \text{ and } \mu_{j+1} = \alpha \mathbf{w}'_j + (1 - \alpha) \mu_j, \tag{8}$$

[1] The local entropy should not be confused the smoothed risk surface obtained by convolution with a Gaussian kernel: every point on this surface represents the average risk of a network obtained by perturbing the network parameters according to a Gaussian distribution. In contrast, the local entropy represents the log average exponentiated risk of a network sampled from $P_{\gamma,\tau}^{\mathbf{w},S}$: this process also starts with sampling perturbation, but the perturbation is either accepted or rejected based upon its relative performance (as measured by exponential loss) compared with typical perturbations. Thus the local entropy focuses much more on performant weight vectors, provided they have sufficient probability mass under the distribution of a random perturbation. Our analysis (Section 4) gives us further insight into the local entropy function.

Algorithm 1 One Entropy-SGLD step along the local entropy gradient

Input:
$\mathbf{w} \in \mathbb{R}^p$ ▷ Current weight
$S \in Z^m$ ▷ Data
$\ell : \mathbb{R}^p \times Z \to \mathbb{R}$ ▷ Loss
$\tau, \gamma, \eta, \eta', L, K$ ▷ Parameters
Output: Weight vector $\mathbf{w}$ moved along stochastic gradient

procedure ENTROPY-SGLD-STEP
  $\mathbf{w}', \mu \leftarrow \mathbf{w}$
  for $i \in \{1, ..., L\}$ **do** ▷ Run SGLD for L iterations.
    $\eta'_i \leftarrow \eta'/i$
    $(z_{j_1}, \ldots, z_{j_K}) \leftarrow$ sample size K minibatch from S
    $\mathrm{d}\mathbf{w}' \leftarrow \frac{\tau}{K} \sum_{i=1}^{K} \nabla_{\mathbf{w}'} \ell(\mathbf{w}', z_{j_i}) - \gamma\tau(\mathbf{w}' - \mathbf{w})$
    $\mathbf{w}' \leftarrow \mathbf{w}' - \frac{1}{2}\eta'_i \mathrm{d}\mathbf{w}' + \sqrt{\eta'_i} N(0, I)$
    $\mu \leftarrow (1-\alpha)\mu + \alpha \mathbf{w}'$
  $\mathbf{w} \leftarrow \mathbf{w} - \frac{1}{2}\eta\tau\gamma(\mathbf{w} - \mu) + \sqrt{\eta} N(0, I)$ ▷ Step along stochastic local entropy ∇
  return $\mathbf{w}$

where $\mathbf{w}'_1, \mathbf{w}'_2, \ldots$ are i.i.d. samples from $P_{\gamma,\tau}^{\mathbf{w},S}$ and $\alpha \in (0,1)$ defines a weighted average.

Obtaining samples from $P_{\gamma,\tau}^{\mathbf{w},S}$ may be difficult when the dimensionality of the weight vector is large. Chaudhari et al. use Stochastic Gradient Langevin Dynamics (SGLD; Welling and Teh, 2011), which generates an exact sample in the limit of infinite computation and requires that the empirical risk be differentiable.[2]

Algorithm 2 in Appendix A.1 gives a complete description of a single stochastic gradient step using Entropy-SGD with SGLD. The parameter $\tau^{-1/2}$ appears as a multiplicative factor in the Langevin dynamics noise and is what Chaudhari et al. call the "thermal noise". The analysis here will make it clear that scaling the noise term in SGLD update has a profound effect: the thermal noise exponentiates the density that defines the local entropy function. We will see that the thermal noise determines the differential privacy and thus the generalization error of Entropy-SGD.

3.2 KL DIVERGENCE AND THE PAC-BAYES THEOREM

Let Q, P be probability measures defined on $\mathbb{R}^p$, assume Q is absolutely continuous with respect to P, and write $\frac{\mathrm{d}Q}{\mathrm{d}P} : \mathbb{R}^p \to \mathbb{R}_+ \cup \{\infty\}$ for some Radon–Nikodym derivative of Q with respect to P. Then the Kullback–Liebler divergence (or relative entropy) of P from Q is defined to be

$$\mathrm{KL}(Q||P) \stackrel{\text{def}}{=} \int \log \frac{\mathrm{d}Q}{\mathrm{d}P} \, \mathrm{d}Q. \tag{9}$$

We are also interested in the special case where both Q, P are dominated by Lebesgue measure, with densities q, p, respectively. In this case, $\mathrm{KL}(Q||P) = \int \log \frac{q(x)}{p(x)} q(x) \mathrm{d}x$. For $p, q \in [0,1]$, we will abuse notation and define

$$\mathrm{KL}(q||p) \stackrel{\text{def}}{=} \mathrm{KL}(\mathscr{B}(q)||\mathscr{B}(p)) = q \log \frac{q}{p} + (1-q) \log \frac{1-q}{1-p}, \tag{10}$$

where $\mathscr{B}(p)$ denotes the Bernoulli distribution on $\{0,1\}$ with mean p.

We now present a PAC-Bayes theorem, first established by McAllester (1999). We focus on the setting of bounding the generalization error of a (randomized) binary classifier. The following variation is due to Langford and Seeger (2001) for the 0–1 loss functions (see also (Langford, 2002) and (Catoni, 2007).)

Theorem 3.1 (PAC-Bayes (McAllester, 1999; Langford and Seeger, 2001))**.** *Let the loss function be the 0–1 loss as defined above. For every $\delta > 0$, $m \in \mathbb{N}$, distribution $\mathscr{D}$ on $\mathbb{R}^k \times \{-1, 1\}$, and*

[2] Chaudhari et al. take $L = 20$ steps of SLGD, using a constant (!) step sizes $\eta'_j = 0.2$ on iteration j, and weighting $\alpha = 0.75$. These settings are likely insufficient to produce high quality samples.

distribution P on $\mathbb{R}^p$,

$$\mathop{\mathbb{P}}_{S_m\sim\mathscr{D}^m}\left((\forall Q)\ \mathrm{KL}(\hat{R}_{S_m}(Q)||R_{\mathscr{D}}(Q)) \leq \frac{\mathrm{KL}(Q||P)+\log\frac{2m}{\delta}}{m-1}\right) \geq 1-\delta. \tag{11}$$

We will also use the following variation of a PAC-Bayes bound, where we consider any bounded loss function.

Theorem 3.2 (PAC-Bayes Linear (McAllester, 2013; Catoni, 2007))**.** *Let $\lambda > 1/2$ be fixed before the draw of S_m. Let the loss be bounded in $[0, L_{max}]$. Then for every $\delta > 0$, $m \in \mathbb{N}$, distribution $\mathscr{D}$ on $\mathbb{R}^k \times \{-1,1\}$, and distribution P on $\mathbb{R}^p$,*

$$\mathop{\mathbb{P}}_{S_m\sim\mathscr{D}^m}\left((\forall Q)\ R_{\mathscr{D}}(Q) \leq \frac{1}{1-\frac{1}{2\lambda}}\left(\hat{R}_{S_m}(Q)+\frac{\lambda L_{max}}{m}(\mathrm{KL}(Q||P)+\log\frac{1}{\delta})\right)\right) \geq 1-\delta. \tag{12}$$

We will introduce several additional generalization bounds when we introduce differential entropy.

4 Maximizing Local Entropy minimizes a PAC-Bayes Bound

We now present our first contribution, a connection between the local entropy and PAC-Bayes bounds.

Let m, δ, $\mathscr{D}$, and P be as in Theorem 3.1 and let $S \sim \mathscr{D}^m$. The linear PAC-Bayes bound (Theorem 3.2) ensures that for any fixed $\lambda > 1/2$ and bounded loss function, with probability at least $1-\delta$ over the choice of S, the bound

$$\left(1-\frac{1}{2\lambda}\right)\frac{m}{\lambda L_{\max}}R_{\mathscr{D}}(Q) \leq \frac{m}{\lambda L_{\max}}\hat{R}_S(Q)+\mathrm{KL}(Q||P)+g(\delta). \tag{13}$$

holds for all $Q \in \mathscr{M}_1(\mathbb{R}^p)$. Minimizing the upper bound on the risk $R_{\mathscr{D}}(Q)$ of the randomized classifier Q is equivalent to the program

$$\inf_Q \int e(h)\,Q(\mathrm{d}h)+\mathrm{KL}(Q||P) \tag{14}$$

with $r(h) = \frac{m}{\lambda L_{\max}}\hat{R}_S(h)$. By (Catoni, 2007, Lem. 1.1.3), for all $Q \in \mathscr{M}_1(\mathbb{R}^p)$ with $\mathrm{KL}(Q||P) < \infty$,

$$-\log P[\exp(-r)] = \int r(h)\,Q(\mathrm{d}h)+\mathrm{KL}(Q||P)-\mathrm{KL}(Q||P_{\exp(-r)}), \tag{15}$$

where $P[\exp(-r)] = \int \exp(-r(h))\,P(\mathrm{d}h)$ and $P_{\exp(-r)}$ is the "Gibbs posterior" distribution, absolutely continuous with respect to P, with density

$$\frac{\mathrm{d}P_{\exp(-r)}}{\mathrm{d}P}(h) = \frac{\exp(-r(h))}{P[\exp(-r)]}. \tag{16}$$

It is plain to see that $\log P[\exp(-r)] = F_{\gamma,\tau}(\mathbf{w};S)$ when $\tau = \frac{m}{\lambda L_{\max}}$ and P is a multivariate normal with mean $\mathbf{w}$ and covariance matrix $\frac{1}{\tau\gamma}I$. Using Eq. (15), we may rexpress Eq. (14) as

$$\inf_Q\ \mathrm{KL}(Q||P_{\exp(-r)})-\log P[\exp(-r)]. \tag{17}$$

By the nonnegativity of the Kullback–Liebler divergence, the infimum is achieved when the KL term is zero, i.e., when $Q = P_{\exp(-r)}$. Then

$$\left(1-\frac{1}{2\lambda}\right)\frac{m}{\lambda L_{\max}}R_{\mathscr{D}}(P_{\exp(-r)}) \leq -\log P[\exp(-r)]+g(\delta), \tag{18}$$

and so maximizing the local entropy with respect to $\mathbf{w}$ is equivalent to minimizing a bound on the test error of the Gibbs posterior classifier.

The analysis falls short when the loss function is unbounded as the PAC-Bayes bound used applies only to bounded loss functions. Germain et al. (2016) extended PAC-Bayes theorems for unbounded loss functions. For their bounds to be evaluated on the negative log likelihood loss, one would need to approximate some statistics of the deviation of the empirical risk $\hat{R}_S(w)$ from true risk $R_{\mathscr{D}}(w)$ under the data distribution and the prior distribution on the weights.

5 PRELIMINARIES: DIFFERENTIAL PRIVACY AND GENERALIZATION

The analysis of the previous section reveals that Entropy-SGD is optimizing a PAC-Bayes bound with respect to the prior P. As a result, the prior P depends on the sample S, and the hypotheses of the PAC-Bayes theorem (Theorem 3.1) are not met. Naively, it would seem that this interpretation of Entropy-SGD could not explain its ability to generalize. However, using tools from differential privacy (Dwork, 2006), we will show that Entropy-SGD, if carefully configured, optimizes its PAC-Bayes prior in a differentially private way, which then guarantees that the PAC-Bayes theorem still applies, at the cost of a slightly looser bound. In Appendix A, we give basic definitions and results for readers unfamiliar with differential privacy.

5.1 DIFFERENTIAL PRIVACY OF GIBBS POSTERIORS

As our analysis in the first half demonstrates, there is a clear connection between Entropy-SGD and the risk of a classifier drawn from a Gibbs posterior. Remarkably, it is known that sampling from Gibbs posteriors is differentially private—indeed, the sampling process is equivalent to the well-known exponential mechanism for privately releasing parameters fit to data. The following result has been established in various forms to varying degrees of generality by numerous authors (Mir, 2013; Bassily, Smith, and Thakurta, 2014; Dimitrakakis et al., 2014; Wang, Fienberg, and Smola, 2015; Minami et al., 2016; Oneto, Ridella, and Anguita, 2017).

Theorem 5.1. *Let $P \in \mathscr{M}_1(\mathbb{R}^p)$ and $\tau > 0$. Define $Q \colon Z^m \rightsquigarrow \mathbb{R}^p$ to be the randomized algorithm that, on input $S \in Z^m$, returns a sample from the Gibbs posterior $P_{\exp(-\tau \hat{R}_S)}$ on $\mathbb{R}^p$ associated to a loss function $\ell : \mathbb{R}^p \times Z \to \mathbb{R}$. Assume $0 \le -\ell \le B$. Then Q is ε-differentially private for $\varepsilon = 2B\tau/m$.*

The theorem presumes that a perfect sample is generated, which is computationally intractable. Entropy-SGD employs SGLD to produce approximate samples from the local Gibbs posterior. Wang, Fienberg, and Smola (2015) study SGLD and establish (ε, δ)-differential privacy under log loss. We will proceed under the optimistic assumption that we can achieve ε-differential privacy, leaving a more realistic analysis to future work.

5.2 CONCENTRATION OF MEASURE

The following result is due to Dwork et al. (2015b, Thm. 11).

Theorem 5.2. *Let $m \in \mathbb{N}$, let $\mathscr{A} : Z^m \rightsquigarrow T$, let $\mathscr{D}$ be a distribution over Z, let $\beta \in (0,1)$, and, for each $t \in T$, fix a set $R(t) \subseteq Z^m$ such that $\mathbb{P}_{S\sim\mathscr{D}^m}(S \in R(t)) \le \beta$. If $\mathscr{A}$ is ε-differentially private for $\varepsilon \le \sqrt{\ln(1/\beta)/(2m)}$, then $\mathbb{P}_{S\sim\mathscr{D}^m}(S \in R(\mathscr{A}(S))) \le 3\sqrt{\beta}$.*

Using Theorem 5.2, one can adapt tail bounds on the generalization error for hypotheses chosen in a differentially private way. The following theorem is a simple variant of (Dwork et al., 2015b, Thm. 9) due to Oneto, Ridella, and Anguita (2017, Lem. 2).

Theorem 5.3. *Let $m \in \mathbb{N}$ and let $\mathscr{A} : Z^m \rightsquigarrow \mathbb{R}^p$ be ε-differentially private. Then*

$$\mathop{\mathbb{P}}_{S\sim\mathscr{D}^m}\left(|R_{\mathscr{D}}(\mathscr{A}(S)) - \hat{R}_S(\mathscr{A}(S))| \ge \varepsilon + \sqrt{1/m}\right) \le 3e^{-m\varepsilon^2}.$$

Theorem 5.4 ((Oneto, Ridella, and Anguita, 2017, Lem. 3))**.** *Let $m \in \mathbb{N}$ and let $\mathscr{A} : Z^m \rightsquigarrow \mathbb{R}^p$ be ε-differentially private. Then*

$$\mathop{\mathbb{P}}_{S\sim\mathscr{D}^m}\left(|R_{\mathscr{D}}(\mathscr{A}(S)) - \hat{R}_S(\mathscr{A}(S))| \ge \sqrt{6\hat{R}_S(\mathscr{A}(S))}(\varepsilon + \sqrt{1/m}) + 6(\varepsilon^2 + 1/m)\right) \le 3e^{-m\varepsilon^2}.$$

6 AN ε-DIFFERENTIALLY PRIVATE PAC-BAYES BOUND

Our next result extends the PAC-Bayes bound using differential privacy to allow the prior to depend on the data S. To that end, fix a distribution $\mathscr{D}$ on $Z = \mathbb{R}^k \times \{-1, 1\}$. For every probability distribution P on $T = \mathbb{R}^p$, let

$$R(P) = \left\{S \in Z^m : (\exists Q)\ \mathrm{KL}(\hat{R}_S(Q) || R_{\mathscr{D}}(Q)) \ge (m-1)^{-1}(\mathrm{KL}(Q||P) + \ln(2m/\delta))\right\}. \tag{19}$$

It follows from the PAC-Bayes theorem (Theorem 3.1) that $\mathbb{P}_{S\sim\mathscr{D}^m}(S \in R(P)) \leq \delta$. The PAC-Bayes theorem allows one to choose the prior based on the data-generating distribution $\mathscr{D}$, but not on the particular sample $S \sim \mathscr{D}^m$. However, using differential privacy, we can consider the possibility of choosing the prior P based on the data sample. Indeed, Theorem 5.2 implies that the same bound holds with a slightly inflated probability of failure. Taking $\delta = 3\sqrt{\beta}$ (and thus $\ln(1/\beta) = 2\ln(3/\delta)$), we obtain the following bound:

Theorem 6.1. *Let $m \in \mathbb{N}$, let $\mathscr{P} \colon Z^m \rightsquigarrow \mathscr{M}_1(\mathbb{R}^p)$, let $\mathscr{D} \in \mathscr{M}_1(\mathbb{R}^k \times \{-1,1\})$, and let $\delta > 0$. If $\mathscr{P}$ is ε-differentially private, then*

$$\mathop{\mathbb{P}}_{S\sim\mathscr{D}^m}\Big((\forall Q)\ \mathrm{KL}(\hat{R}_S(Q)||R_{\mathscr{D}}(Q)) \leq \frac{\mathrm{KL}(Q||\mathscr{P}(S)) + \ln 2m + 2\max\{\ln\frac{3}{\delta},\ m\varepsilon^2\}}{m-1}\Big) \geq 1-\delta. \tag{20}$$

7 Numerical evaluations on MNIST

In this section we run Entropy-SGLD algorithm as described in Algorithm 1. We assume that SGLD generates a perfect sample from $P^{\mathbf{w},S}_{\gamma,\tau}$, which allows us to get $\varepsilon \sim \tau/m$ differential privacy as stated in Theorem 5.1. As we release only a single sample, the number of epochs do not accumulate any additional privacy loss. We choose a relatively small τ to get sufficiently high DP. We then apply generalization bounds stated in the previous sections (Theorems 5.3, 5.4 and 6.1). Theorem 6.1 allows us to run a differentially private algorithm to optimize the prior and get valid PAC-Bayes generalization guarantees. We thus evaluate Theorem 6.1 with the learnt optimal prior, and with the posterior Q chosen to be Gibbs posterior evaluated at the full likelihood $\tau = m$.

7.1 Details

We evaluate Entropy-SGLD's performance and generalization bounds on a binary classification task adapted from MNIST (LeCun, Cortes, and Burges, 2010).[3] (See Appendix C for the experiments on the standard multiclass MNIST dataset). Some experiments involve random labels, i.e., labels drawn independently and uniformly at random at the start of training. We study three network architectures, abbreviated FC600, FC1200, and CONV. Both FC600 and FC1200 are 3-layer fully connected networks, with 600 and 1200 units per hidden layer, respectively. CONV is a convolutional architecture. All three network architectures are taken from the MNIST experiments by Chaudhari et al. (2017), but adapted to our binary version of MNIST.[4] Let S and S_{tst} denote the training and test sets, respectively. We track

(i) $\hat{R}_S(\mathbf{w})$ and $\hat{R}_{S_{\text{tst}}}(\mathbf{w})$, i.e., the training and test error for $\mathbf{w}$ ("local")

(ii) estimates of $\hat{R}_S(P^{\mathbf{w},S}_{\gamma,\tau})$ and $\hat{R}_{S_{\text{tst}}}(P^{\mathbf{w},S}_{\gamma,\tau})$, i.e., the mean training and test error of the local Gibbs distribution, viewed as a randomized classifier ("Gibbs")

and, using the differential privacy bounds in Theorem B.1, compute

(iii) a PAC-Bayes bound on $R_{\mathscr{D}}(P^{\mathbf{w},S}_{\gamma,\tau})$ using Theorem 6.1 ("PAC-bound");

(iv) the mean of a Hoeffding-style bound on $R_{\mathscr{D}}(\mathbf{w}')$, where $\mathbf{w}' \sim P^{\mathbf{w},S}_{\gamma,\tau}$, using Theorem 5.3 ("H-bound");

(v) an upper bound on the mean of a Chernoff-style bound on $R_{\mathscr{D}}(\mathbf{w}')$, where $\mathbf{w}' \sim P^{\mathbf{w},S}_{\gamma,\tau}$, using Theorem 5.4 ("C-bound").

See Appendix B.1 for additional details. Note that, in the calculation of (iii), we do not account for Monte Carlo error in our estimate of $\hat{R}_S(\mathbf{w})$. We suspect the effect is small, given the large number of iterations of SGLD performed for each point in the plot. Recall that

$$R_{\mathscr{D}}(P^{\mathbf{w},S}_{\gamma,\tau}) = \mathop{\mathbb{E}}_{\mathbf{w}'\sim P^{\mathbf{w},S}_{\gamma,\tau}}(R_{\mathscr{D}}(\mathbf{w}')), \tag{21}$$

[3] The MNIST handwritten digits dataset (LeCun, Cortes, and Burges, 2010) consists of 60000 training set images and 10000 test set images, labeled 0–9. We transformed MNIST to a binary classification task by mapping digits 0–4 to label 1 and 5–9 to label -1.

[4] We adapt the code provided by Chaudhari et al., with some modifications to the training procedure and straightforward changes necessary for our binary classification task.

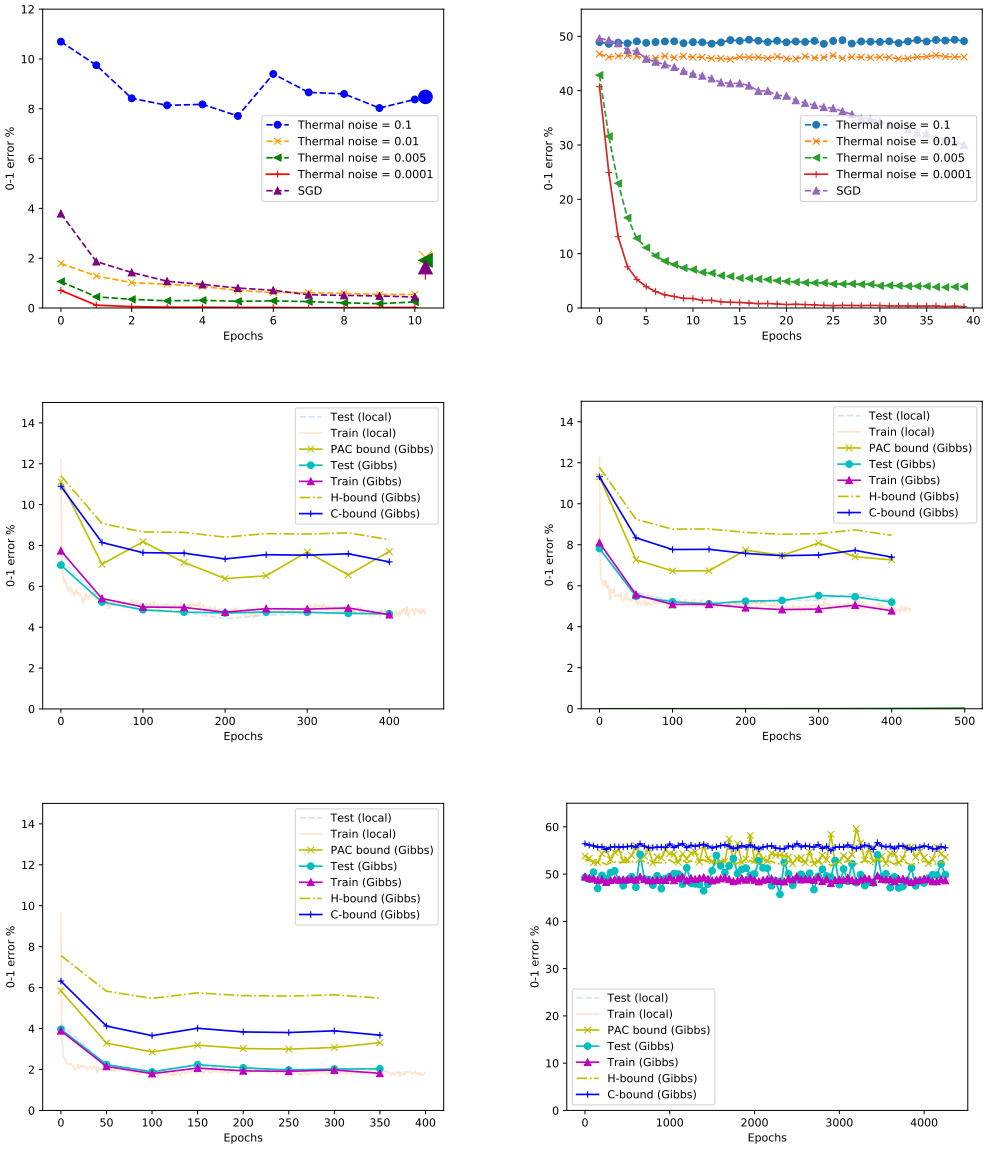

Figure 1: **(clockwise from top-left)** Entropy-SGD and vanilla-SGD performance on the training set of the binarized MNIST task. The lines indicate the 0–1 error on the training data. The larger marker at the end of 10 epochs indicates the 0–1 error on the test set, which is an empirical estimate of the true error. Thus the gap between the line (training error) and the final marker (empirical test error) is the approximate generalization error. On true labels, both algorithms find classifiers with small empirical risk and low generalization error. As we increase the thermal noise of entropy-SGD algorithm, the empirical 0–1 error increases, but the generalization gap decreases. **(clockwise from top-right)** On random labels, both algorithms exhibit high generalization error. (True error is $\sim 50\%$). **(middle-left)** Entropy-SGLD applied to FC600 network trained on true labels. **(middle-right)** Entropy-SGLD applied to FC1200 network trained on true labels. Both training error and generalization error are similar to FC600 case. Bounds are loose but nonvacuous. **(bot-right)** Entropy-SGLD applied to CONV network on true labels. Lower error and bounds than achieved with FC networks. **(bot-left)** Entropy-SGLD applied to FC600 network on random labels. The algorithm does not overfit like SGD and Entropy-SGD.

and so we may interpret the bounds in terms of the performance of a randomized classifier or the mean performance of a randomly chosen classifier.

7.2 RESULTS

Key results on FC600 and CONV appear in Fig. 1.

On the true label dataset, Entropy-SGLD with our choice of differential privacy parameter ε achieves a lower training accuracy than vanilla SGD or Entropy-SGD. However, both the local and Gibbs classifiers found by the algorithm have essentially zero generalization error. Performance and bounds for FC600 and FC1200 are nearly identical, despite FC1200 having three times as many parameters. Training the CONV network produces the lowest training/test errors. On random labels, vanilla SGD and Entropy-SGD on FC600 overfit, while Entropy-SGLD maintain essentially zero generalization error.

We find that PAC-Bayes bound is comparable or tighter than H- and C-bounds . All bounds are nonvacuous for the choice of the algorithmic parameters, though still loose. We discuss this gap later. The error bounds computed here are tighter than the ordinary PAC-Bayes bounds reported by Dziugaite and Roy (2017). On the other hand, there are several unrealistic assumptions that have gone into the analysis, which affects the validity of the bounds. Foremost, SGLD does not produce exact samples from the local Gibbs distributions. Despite the unrealistic assumptions in the analysis, no bound is ever violated. The advantage of making such an assumption about SGLD, is that the bound on the differential privacy of Entropy-SGLD does not grow with the number of iterations. Alternative analysis of per step differential privacy would require the use of composition and sequencing of differential privacy of the algorithm, which in turn very quickly lead to vacuous bounds.

7.2.1 COMPARISON TO SGLD

For comparison, we evaluate SGLD performance under the same assumption that SGLD gives us a perfect sample from the Gibbs posterior. We train the FC600 network with SGLD minimizing (clipped) binary cross entropy loss using different τ values. Similarly as for Entropy-SGLD and Entropy-SGD algorithms, the larger the τ value, the smaller the training error can be achieved. On the random label dataset, this means that larger τ values results in overfitting, since the training error drops well below 50%. On the true labelling of binarized MNIST dataset, we compare SGLD performance to Entropy-SGLD using the same τ. SGLD achieved lower accuracy on the train and test sets ($\sim 1-2\%$ lower than Entropy-SGLD). The C-bound on the test error evaluated on SGLD network was above 10%, which is around 2% higher than for Entropy-SGLD trained network.

8 DISCUSSION

Given how the training and test error track each other for Entropy-SGLD, it seems possible that our differential privacy bounds are very loose. Indeed, given the similarity between Entropy-SGD and vanilla SGLD, and the fact that SGLD approximates a sample from a Gibbs distribution, it seems possible that the gap in our analysis is substantial.

On the other hand, it also seems conceivable that there is a tradeoff between the speed of learning, the achievable error, and the ability to produce a certificate of one's generalization error (e.g., via a DP bound). EntropySGD learns much faster in its original configuration, but its performance on random labels implies it has poor differential privacy.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang (2016). "Deep Learning with Differential Privacy". *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: ACM, pp. 308–318. DOI: 10.1145/2976749.2978318.

Alessandro Achille and Stefano Soatto (2017). "On the Emergence of Invariance and Disentangling in Deep Representations". *CoRR* abs/1706.01350. arXiv: 1706.01350.

Carlo Baldassi, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina (2015). "Subdominant Dense Clusters Allow for Simple Learning and High Computational Performance in Neural Networks with Discrete Synapses". *Phys. Rev. Lett.* 115 (12), p. 128101. DOI: 10.1103/PhysRevLett.115.128101.

Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina (2016). "Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes". *Proceedings of the National Academy of Sciences* 113.48, E7655–E7662. DOI: 10.1073/pnas.1608103113. eprint: http://www.pnas.org/content/113/48/E7655.full.pdf.

Peter Bartlett, Dylan J. Foster, and Matus Telgarsky (2017). "Spectrally-normalized margin bounds for neural networks". *CoRR* abs/1706.08498. arXiv: 1706.08498.

Peter L. Bartlett and Shahar Mendelson (2003). "Rademacher and Gaussian Complexities: Risk Bounds and Structural Results". *J. Mach. Learn. Res.* 3, pp. 463–482.

Raef Bassily, Adam Smith, and Abhradeep Thakurta (2014). "Differentially private empirical risk minimization: Efficient algorithms and tight error bounds". *arXiv preprint arXiv:1405.7085*.

Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman (2016). "Algorithmic stability for adaptive data analysis". *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, pp. 1046–1059.

Olivier Catoni (2007). "PAC-Bayesian supervised classification: the thermodynamics of statistical learning". *arXiv preprint arXiv:0712.0248*.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina (2017). "Entropy-SGD: Biasing Gradient Descent Into Wide Valleys". *International Conference on Learning Representations (ICLR)*. arXiv: 1611.01838v4 [cs.LG].

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate (2011). "Differentially private empirical risk minimization". *Journal of Machine Learning Research* 12.Mar, pp. 1069–1109.

Changyou Chen, Nan Ding, and Lawrence Carin (2015). "On the convergence of stochastic gradient MCMC algorithms with high-order integrators". *Advances in Neural Information Processing Systems*, pp. 2278–2286.

Christos Dimitrakakis, Blaine Nelson, Aikaterini Mitrokotsa, and Benjamin IP Rubinstein (2014). "Robust and private Bayesian inference". *International Conference on Algorithmic Learning Theory*. Springer, pp. 291–305.

Cynthia Dwork (2006). "Differential Privacy". *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10–14, 2006, Proceedings, Part II*. Ed. by Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–12. DOI: 10.1007/11787006_1.

– (2008). "Differential privacy: A survey of results". *International Conference on Theory and Applications of Models of Computation*. Springer, pp. 1–19.

Cynthia Dwork, Aaron Roth, et al. (2014). "The algorithmic foundations of differential privacy". *Foundations and Trends in Theoretical Computer Science* 9.3–4, pp. 211–407.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toni Pitassi, Omer Reingold, and Aaron Roth (2015a). "Generalization in adaptive data analysis and holdout reuse". *Advances in Neural Information Processing Systems*, pp. 2350–2358.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth (2015b). "Preserving statistical validity in adaptive data analysis". *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*. ACM, pp. 117–126.

Gintare Karolina Dziugaite and Daniel M. Roy (2017). "Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data". *arXiv preprint arXiv:1703.11008*.

Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien (2016). "PAC-Bayesian Theory Meets Bayesian Inference". *Advances in Neural Information Processing Systems*, pp. 1884–1892.

Moritz Hardt, Benjamin Recht, and Yoram Singer (2015). "Train faster, generalize better: Stability of stochastic gradient descent". *CoRR* abs/1509.01240.

Geoffrey E. Hinton and Drew van Camp (1993). "Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights". *Proceedings of the Sixth Annual Conference on Computational Learning Theory*. COLT '93. Santa Cruz, California, USA: ACM, pp. 5–13. DOI: 10.1145/168304.168306.

Sepp Hochreiter and Jürgen Schmidhuber (1997). "Flat Minima". *Neural Comput.* 9.1, pp. 1–42. DOI: 10.1162/neco.1997.9.1.1.

Daniel Kifer, Adam Smith, and Abhradeep Thakurta (2012). "Private convex empirical risk minimization and high-dimensional regression". *Journal of Machine Learning Research* 1.41, pp. 1–40.

Diederik P Kingma, Tim Salimans, and Max Welling (2015). "Variational Dropout and the Local Reparameterization Trick". *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., pp. 2575–2583.

John Langford (2002). "Quantitatively tight sample complexity bounds". PhD thesis. Carnegie Mellon University.

John Langford and Rich Caruana (2002). "(Not) Bounding the True Error". *Advances in Neural Information Processing Systems 14*. Ed. by T. G. Dietterich, S. Becker, and Z. Ghahramani. MIT Press, pp. 809–816.

John Langford and Matthias Seeger (2001). *Bounds for Averaging Classifiers*. Tech. rep. CMU-CS-01-102. Carnegie Mellon University.

Yann LeCun, Corinna Cortes, and Christopher J. C. Burges (2010). *MNIST handwritten digit database*. http://yann.lecun.com/exdb/mnist/.

Guy Lever, François Laviolette, and John Shawe-Taylor (2013). "Tighter PAC-Bayes bounds through distribution-dependent priors". *Theoretical Computer Science* 473, pp. 4–28. DOI: http://dx.doi.org/10.1016/j.tcs.2012.10.013.

David A. McAllester (1999). "PAC-Bayesian Model Averaging". *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*. COLT '99. Santa Cruz, California, USA: ACM, pp. 164–170. DOI: 10.1145/307400.307435.

– (2013). "A PAC-Bayesian Tutorial with A Dropout Bound". *CoRR* abs/1307.2118.

Kentaro Minami, Hitomi Arai, Issei Sato, and Hiroshi Nakagawa (2016). "Differential Privacy without Sensitivity". *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., pp. 956–964.

Darakhshan J Mir (2013). "Differential privacy: an exploration of the privacy-utility landscape". PhD thesis. Rutgers University.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro (2014). *In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning*. Workshop track poster at ICLR 2015. arXiv: 1412.6614v4 [cs.LG].

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro (2017a). "A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks". *CoRR* abs/1707.09564. arXiv: 1707.09564.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro (2017b). "Exploring Generalization in Deep Learning". *CoRR* abs/1706.08947. arXiv: 1706.08947.

Luca Oneto, Sandro Ridella, and Davide Anguita (2017). "Differential privacy and generalization: Sharper bounds with applications". *Pattern Recognition Letters* 89, pp. 31–38. DOI: https://doi.org/10.1016/j.patrec.2017.02.006.

Yu-Xiang Wang, Stephen E. Fienberg, and Alexander J. Smola (2015). "Privacy for Free: Posterior Sampling and Stochastic Gradient Monte Carlo". *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, pp. 2493–2502.

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

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals (2017). "Understanding deep learning requires rethinking generalization". *International Conference on Representation Learning (ICLR)*. arXiv: 1611.03530v2 [cs.LG].

A BACKGROUND: DIFFERENTIAL PRIVACY

Here we formally define some of the differential privacy related terms used in the main text. (See (Dwork, 2006; Dwork and Roth, 2014) for more details.)

Let $U, U_1, U_2, \ldots$ be independent uniform $(0,1)$ random variables, independent also of any random variables introduced by $\mathbb{P}$ and $\mathbb{E}$, and let $\pi : \mathbb{N} \times [0,1] \to [0,1]$ satisfy $(\pi(1,U), \ldots, \pi(k,U)) \stackrel{d}{=} (U_1, \ldots, U_k)$ for all $k \in \mathbb{N}$. Write π_k for $\pi(k, \cdot)$.

Definition A.1. A *randomized algorithm* $\mathscr{A}$ from R to T, denoted $\mathscr{A} : R \rightsquigarrow T$, is a measurable map $\mathscr{A} : [0,1] \times R \to T$. Associated to $\mathscr{A}$ is a (measurable) collection of random variables $\{\mathscr{A}_r : r \in R\}$ that satisfy $\mathscr{A}_r = \mathscr{A}(U, r)$. When there is no risk of confusion, we will write $\mathscr{A}(r)$ for $\mathscr{A}_r$.

Definition A.2. A randomized algorithm $\mathscr{A} : Z^m \rightsquigarrow T$ is (ε, δ)*-differentially private* if, for all pairs $S, S' \in Z^m$ that differ at only one coordinate, and all measurable subsets $B \subseteq T$, we have $\mathbb{P}(\mathscr{A}(S) \in B) \leq e^{\varepsilon}\, \mathbb{P}(\mathscr{A}(S) \in B) + \delta$.

We will write ε-differentially private to mean $(\varepsilon, 0)$-differentially private algorithm.

Definition A.3. Let $\mathscr{A} : R \rightsquigarrow T$ and $\mathscr{A}' : T \rightsquigarrow T'$. The *composition* $\mathscr{A}' \circ \mathscr{A} : R \rightsquigarrow T'$ is given by $(\mathscr{A}' \circ \mathscr{A})(u, r) = \mathscr{A}'(\pi_2(u), \mathscr{A}(\pi_1(u), r))$.

Lemma A.4 (post-processing)**.** *Let $\mathscr{A} : Z^m \rightsquigarrow T$ be (ε, δ)-differentially private and let $F : T \rightsquigarrow T'$ be arbitrary. Then $F \circ \mathscr{A}$ is (ε, δ)-differentially private.*

A.1 ENTROPY SGD ALGORITHM

Algorithm 2 One ESGD step along the local entropy gradient

Input:
- $\mathbf{w} \in \mathbb{R}^p$ ▷ Current weight
- $S \in Z^m$ ▷ Data
- $\ell : \mathbb{R}^p \times Z \to \mathbb{R}$ ▷ Loss
- $\tau, \gamma, \eta, \eta', L, K$ ▷ Parameters

Output: Weight vector $\mathbf{w}$ moved along stochastic gradient

```
procedure ENTROPY-SGD-STEP(τ, γ, η, η', L, K, w, S)
    w', μ ← w
    for i ∈ {1, ..., L} do                                  ▷ Run SGLD for L iterations.
        η'_i ← η'/i
        (z_{j_1}, ..., z_{j_K}) ← sample size K minibatch from S
        dw' ← (1/K) Σ_{i=1}^{K} ∇_{w'} ℓ(w', z_{j_i}) − γ(w' − w)
        w' ← w' − η'_i dw' + √(η'_i) (1/√τ) N(0, I)
        μ ← (1 − α)μ + α w'
    w ← w − η τ γ (w − μ)                                   ▷ Step along stochastic local entropy ∇
    return w
```

B DIFFERENTIALLY PRIVATE OPTIMIZATION OF TEST ERROR BOUND

Our generalization bounds are established with the ε-differential privacy of an idealized version of Entropy-SGLD (Algorithm 3) in which SGLD is assumed to produce an exact sample.

Theorem B.1. *Let $\eta, \gamma, \tau > 0$, $n \in \mathbb{N}$, and $\mathbf{w} \in \mathbb{R}^p$. Assume the range of ℓ has diameter B. Then Algorithm 3,* TRAIN$(n, \tau, \gamma, \eta, \cdot, \cdot) : \mathbb{R}^p \times Z^m \rightsquigarrow \mathbb{R}^p$*, is $\frac{2B\tau}{m}$-differentially private.*

Proof (sketch). By Theorem 5.1, producing a sample from $P_{\gamma,\tau}^{\mathbf{w},S}$ is $\frac{2B\tau}{m}$-differentially private, provided that the learning rates satisfy SGLD criteria and we run the algorithm for long enough. As we release only a single sample, the number of calls to IDEAL-ENTROPY-SGD-STEP do not contribute any privacy loss, and so the entire sequence is $\frac{2B\tau}{m}$-differentially private. □

Algorithm 3 E-SGLD idealized training epoch

Input:
$\mathbf{w} \in \mathbb{R}^p$ ▷ Current weight
$S \in Z^m$, $n \in \mathbb{N}$, ▷ Data and number of minibatches
$\ell : \mathbb{R}^p \times Z \to \mathbb{R}$ ▷ Loss
τ, γ, η_0 ▷ Parameters
t ▷ epoch number
Output: Weight vector $\mathbf{w}$ moved along stochastic gradient

procedure IDEAL-ENTROPY-SGLD-STEP($\tau, \gamma, \eta, \mathbf{w}, S'$)
  Sample $\mathbf{w}' \sim P^{\mathbf{w},S'}_{\gamma,\tau}$ ▷ Exact sample from local Gibbs posterior
  return $\mathbf{w} - \eta\tau\gamma(\mathbf{w} - \mathbf{w}')$ ▷ Step along stochastic local entropy ∇
procedure TRAIN($n, \tau, \gamma, \eta, \mathbf{w}, S$)
  Set $\eta_t = \eta_0 t^{-0.25}$
  Sample $\{S_1, \ldots, S_n\}$ from S ▷ Partition data in n mini-batches
  for $j \in \{1, \ldots, n\}$ **do**
    $\mathbf{w} \leftarrow$ IDEAL-ENTROPY-SGD-STEP($\tau, \gamma, \eta_t, \mathbf{w}, S_j$)
  return $\mathbf{w}$

B.1 NETWORK ARCHITECTURE AND HYPERPARAMETERS FOR ENTROPY-SGLD

B.1.1 ARCHITECTURE

The convolution network size is chosen to match the one used in Chaudhari et al. (2017) for multiclass MNIST classification. It has two convolutional layers, a fully connected ReLU layer and a sigmoid layer, yielding a total number of parameters 126711.

For the fully connected network experiments, we used 3-layer networks: FC600 and FC1200, which in each hidden layer have 600 and 1200 units, yielding a total number of 834601 and 2385185 parameters, respectively. We use ReLU activations for all but the last layer, which is sigmoid to produce an output in $[0,1]$.

Note, that unlike in Chaudhari et al. (2017) MNIST experiments, we do not use dropout and only use batch normalization.

B.1.2 OBJECTIVE

All networks are trained with binary cross entropy objective function. For the generalization bounds to be valid, we need the objective function to be bounded. We bound the cross entropy objective $B = 4$, by clipping the neural network output values y to be away from the extremes 0 and 1, i.e., the output is $y(1 - 2\mathrm{e}^{-B}) + \mathrm{e}^{B}$.

B.1.3 EPOCHS

We call an epoch a pass through the whole dataset, i.e., when the outer loop, which is the gradient on $\bar{w}$, is called as many times as the number of minibatches. Every iteration, we sample a new minibatch and then for that fixed minibatch we call the inner loop and perform 20 SGLD steps.

B.1.4 LEARNING RATE

To run SGLD algorithm, one needs the sum of the learning rates during training to diverge and the sum of squared learning rates to converge. We use iteration based decay schedule to adjust the learning rate during training in order to satisfy the requirements for SGLD.

The outer loop base learning rate was set to $\frac{0.006}{\gamma\tau}$. On epoch t, the learning rate was set to the base learning rate times $t^{-0.25}$.

The inner loop base learning rate was set to $\frac{\sqrt{m}}{\gamma\tau^2}$, where m is the number of training samples. On iteration t of the inner optimization loop, the learning rate is set to the initial value divided by the iteration number.

B.1.5 SGLD PARAMETERS

On each training iteration, the gradient uses the expected value of neural network weights under the Gibbs posterior (Eq. (8)). To approximate this expectation, we employ stochastic gradient Langevin dynamics, as in Chaudhari et al. (2017). As in Entropy-SGD, this is done in the inner loop of each step of the algorithm, where we keep the running mean of SGLD samples with the weight on the last sample set to $\alpha = 0.75$ ('for' loop starting in line 3 in Algorithm 1).

In order to compute PAC-Bayes generalization gap, one needs to evaluate the expected loss under the Gibbs posterior (Eq. (24)). Once again, we use SGLD and keep track of the running mean with last sample weight set to $\alpha = 0.005$ to be able to average across more samples and shrink the variance of our Gibbs classifier error and KL terms. SGLD algorithm is being run for 1000 epochs.

B.1.6 GIBBS CLASSIFIER PARAMETERS

We set $\gamma = 1$ and $\tau = \sqrt{m}$ and keep the values fixed during optimization. The value of τ is chosen in advance to have a reasonably small generalization gap due to differential privacy loss in the PAC-Bayes bound, i.e., we want the

$$2\frac{\max\{\ln\frac{3}{\delta},\, m\varepsilon^2\}}{m-1} \tag{22}$$

term in Theorem 6.1 to be close to zero.

B.2 EVALUATING PAC-BAYES BOUND

When the empirical error is close to zero, the KL version of the PAC-Bayes bound is considerably tighter than the Hoeffding-style bound first described by McAllester (1999). However, using this relative entropy bound requires one to be able to compute the largest value p such that $\mathrm{KL}(q||p) \le c$. There does not appear to be a simple formula for this value. In practice, however, the value can be efficiently numerically approximated using, e.g., Newton's method. (See (Dziugaite and Roy, 2017, §2.2 and App. B).)

B.3 ESTIMATING THE KL DIVERGENCE

Let $\ell(\mathbf{w}) = \tau\tilde{R}_S(\mathbf{w})$. It is easy to verify that

$$\mathrm{KL}(P_{\exp(-\ell)}||P) = \mathop{\mathbb{E}}_{\mathbf{w}\sim P_{\exp(-\ell)}}\Big(-\ell\Big) - \log P[\exp(-\ell)]. \tag{23}$$

Both terms have obvious Monte Carlo estimates:

$$\mathop{\mathbb{E}}_{\mathbf{w}\sim P_{\exp(-\ell)}}\Big(-\ell(\mathbf{w})\Big) \approx -\frac{1}{k'}\sum_{i=1}^{k'}\ell(\mathbf{w}') \tag{24}$$

where $\mathbf{w}'_1,\ldots,\mathbf{w}'_{k'}$ are taken from a Markov chain targeting $P_{\exp(-\ell)}$, such as SGLD run for $k' \gg 1$ steps (which is how we computed our bounds), and

$$\log P[\exp(-\ell)] = \log\int \exp\{-\ell(\mathbf{w})\}\,P(\mathrm{d}\mathbf{w}) \tag{25}$$

$$\gtrapprox \log\frac{1}{k}\sum_{i=1}^{k}\exp\{-\ell(\mathbf{w}_i)\}. \tag{26}$$

where $h_1,\ldots,h_k$ are i.i.d. P (which is a multivariate Gaussian in this case). In the latter case, due to the concavity of log, the estimate is a lower bound with high probability.

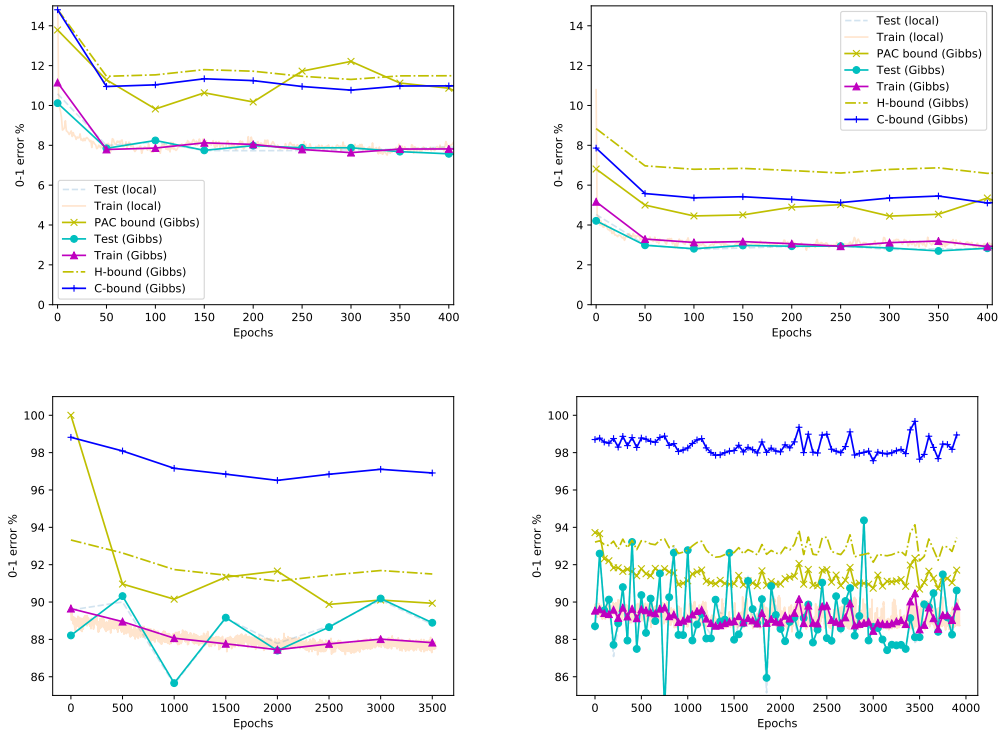

Figure 2: **(top-left)** FC network trained on true labels. The train and test errors suggest that the generalization gap is close to zero, while all three bounds exceed it by more than 3%. **(top-right)** CONV network trained on true labels. Both the train and the test errors are better than in the FC network and we still do not empirically observe any overfitting. The tighter C-bound and PAC-Bayes bounds upper bound the test error by $\sim 5\%$. **(bottom-left)** FC network trained on random labels. After around 1000 epochs we notice the model overfitting the data by $1-2\%$. Running the algorithm further does not improve the training error due to low τ value in the Gibbs posterior, and thus relatively strong differential privacy. **(bottom-right)** CONV network trained on random labels. We do not observe any overfitting. Both the training and the test error seem to be on top of each other on average and stay around the 0–1 loss of random guessing. The bounds do not suggest any non-trivial generalization.

C MULTICLASS MNIST EXPERIMENTS

We evaluate the same generalization bounds on the standard MNIST classification task as in the MNIST binary labelling case. All the details of the network architectures and parameters are as stated in Appendix B.1. The only difference is that we keep the fully connected network fixed to exactly the same size as in Chaudhari et al. (2017) experiments, and so FC will refer to a fully connected network with 1024 hidden units.

C.0.1 OBJECTIVE

As in the binary case, we train using cross entropy objective function. To make the loss function bounded, we first apply soft-max to the outputs of the neural network and then clip the values as in the binary case as described in Appendix B.1.2. We compute negative log likelihood loss on the clipped output values.