Stochastic Gradient Langevin Dynamics For Optimizing Neural Networks and Characterizing Model uncertainty

EECS126 Project SP19

Noah Golmant **Olivia Koshy** **Megan Kawakami** **Phillip Kravtsov**

April 28, 2019

Abstract

Most popular methods for optimizing neural networks only provide point estimates of the mode of the posterior distribution of the parameters given the data. Stochastic Gradient Langevin Dynamics (SGLD) is a variant of Stochastic Gradient Descent (SGD) that injects isotropic normal noise into the parameter update, allowing us to sample from this posterior using Markov Chain Monte Carlo methods. We investigate the behavior of SGLD by examining properties of the samples like stationarity, prediction quality, and prediction confidence.

1 Introduction

Stochastic gradient descent (SGD) is a powerful tool for optimizing neural networks, but most existing techniques only provide a point estimate of the mode of the posterior distribution over the parameters given the data. Stochastic Gradient Langevin Dynamics (SGLD) is a simple modification of SGD that injects normal noise into the gradient dynamics so that the resulting trajectory can be regarded as a Markov chain [5]. As SGLD converges, the iterates approach true samples from a Gibbs posterior centered about the local minimum. We can then take the models from the last epochs of training to be samples from the approximate posterior.

This sampling technique effectively claims to provide "for-free" Bayesian deep learning, which has the potential to improve robustness to out-of-distribution data and to provide a cheap measure of model uncertainty and prediction confidence. Cheap samples also allow for computationally efficient training of deep ensembles. In this project, we perform an empirical analysis of SGLD to test the hypotheses surrounding Bayesian learning through Langevin dynamics.

2 Methods

Let $\theta_t \in \mathbb{R}^d$ be the parameters of a neural network at timestep t, and $f\colon \mathbb{R}^d \to \mathbb{R}$ be the objective function for the task, which is taken to the be negative log-likelihood of the posterior $p(\theta|x)$. The traditional stochastic gradient descent update equation with learning rate $\eta_t > 0$ is given by:

$$\theta_{t+1} = \theta_t - \eta_t \tilde{\nabla} f(\theta_t) \tag{1}$$

where $\tilde{\nabla} f(\theta_t)$ is a stochastic estimate of the gradient, $\nabla f(\theta_t)$, and $\eta_t \to 0$ as $t \to \infty$. With a sufficiently fast learning rate schedule, SGD will converge to a point estimate of the mode of $p(\theta|x)$. This is equivalent to a local minimum of the negative log-likelihood of the model, f. In order to turn this into a MCMC method, SGLD injects additive isotropic normal noise into each update using a fixed variance σ^2:

$$\theta_{t+1} = \theta_t - \eta_t \tilde{\nabla} f(\theta_t) + \epsilon_t : \epsilon_t \sim \mathcal{N}(0, \sigma^2 I) \tag{2}$$

As $\eta_t \to 0$, the noise term ϵ_t dominates the gradient, so that the resulting dynamics resemble Brownian motion. This is equivalent to ensuring that the stationary distribution of the Markov Chain is a Gibbs distribution centered about the final local minimum, since Brownian motion is a martingale process. This stationarity behavior is not guaranteed with traditional SGD since its iterates form a supermartingale.

We tested SGLD on a standard classification task. We used a ResNet-18 architecture to predict the classes of images from the CIFAR-10 dataset [2]. CIFAR-10 and ResNet-18 allow for reasonably fast and cheap experimentation while retaining the task and model complexity necessary for testing the efficacy of SGLD on a non-toy problem. In fact, very few works have tested SGLD on standard benchmarks like CIFAR-10/100. We trained with several noise levels (σ^2 in Equation 2) and reported results for the trial that obtained the highest test accuracy ($\sigma^2 = 1e-8$). This is because we observed that higher noise levels significantly deteriorated the performance of the model, even relative to a weak baseline. Throughout the paper, to measure lack of model confidence and prediction uncertainty, we computed the Shannon entropy of the softmax output of the model, similar to the work of [3].

We constructed an ensemble classifier by using the sample models obtained throughout the last 40 epochs of training. This computationally efficient form of ensembling is one of the main purported benefits of SGLD. To classify examples using the ensemble, we average the softmax outputs of each model. Let $p_{i,j}(x)$ denote the probability of the i-th class for the j-th model in the ensemble. For m models, the final ensemble prediction $p_i(x)$ is given by:

$$p_i(x) = \frac{1}{m}\sum_{j=1}^{m} p_{i,j}(x) \tag{3}$$

We examine the utility of SGLD as a cheap Bayesian alternative to SGD by testing the following hypotheses:

1. The samples near the end of training should should represent samples from the stationary distribution of the Markov chain. The standard training time should be sufficient to ensure that the motion of these later iterates resembles Brownian motion.
2. When run as an ensemble, the samples should achieve similar performance to the non-Bayesian model (an equivalent model trained without noise), measured by test set classification accuracy.
3. The ensemble should be robust to out-of-distribution data. That is, it should be significantly more confident on data drawn from its training distribution than data drawn from some other arbitrary distribution, e.g. some other dataset. We consider a confident prediction to be one with low Shannon entropy.

We obtained samples from the posterior using the following training procedure:

```
ensemble = {}
for epoch in num_epochs:
    net.parameters = net.parameters − lr ∗ net.parameters.grad(loss)
    net.parameters += noise(sigma)
    if epoch > burn_in_epoch and sigma > 0:
        ensemble.append(net.copy())
    lr = lr ∗ decay
```

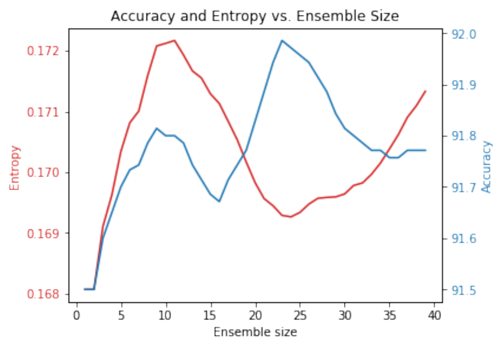
We compute the average prediction entropy for each class in the dataset with the following procedure:

```
entropies = {}
for class in num_classes:
    total_entropy = 0
    for image in images[class]:
        distribution = ensemble[class].predict(image)
        total_entropy += sum(−p ∗ log(p) for p in distribution)
    total_entropy /= len(images[class])
    entropies.append(total_entropy)
model_entropy = mean(entropies)
```

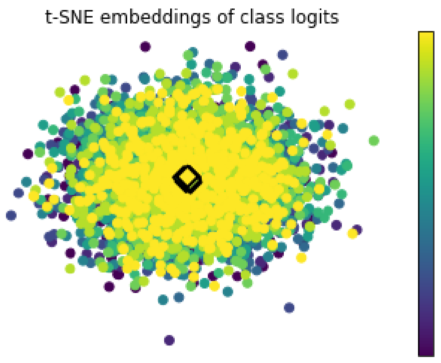
3 Experiments, Results, and Analysis

3.1 Sample Stationarity

To test the hypothesis that the samples are approximately drawn from the stationary distribution of the chain, we performed two experiments. In the first, we evaluated model accuracy and entropy as a function of ensemble size. In this case, we expect that accuracy should display low variance while entropy should slightly increase with ensemble size.



(a) Accuracy and entropy on the CIFAR-10 test data as a function of ensemble size.



(b) t-SNE embeddings of the network activations for points from a single class. Lighter color indicates later in training. Diamonds indicate centroids of the clusters. As training progresses, the points cluster about a mostly-static centroid.

This is because the (argmaxed) class predictions should not vary much in the vicinity of the minimum, but averaging the softmax outputs may still produce a noisier (higher entropy) decision than any particular model. In the second experiment, we used t-SNE [4] dimensionality reduction to visualize the pre-activations of the network's linear layer in 2-D. We plotted the projection of all points in a single class of the dataset, and we did this for each model in the ensemble. We expect that the clusters (in particular, the class centroids) should display low variance in the stationary case.

The results for the first experiment are shown in Figure 1a. The test accuracy for all models varies between 91.5% and 92.0%. For a test set containing $10,000$ examples, this amounts to a difference of 50 samples, which is quite low. Measuring bits of Shannon entropy has a harder interpretation, but it is clear that the entropy of any ensemble is higher than the entropy of using a single model (where the ensemble size is one).

The results of the t-SNE experiment are shown in Figure 1b. Each cluster of colored points represents the projection of the activations for all points in a single CIFAR-10 class. As training progresses (and the color moves from purple to yellow), the class centroid does not move much, but the points move in closer towards it. This provides evidence that although the class decision boundary might not be changing significantly, the movements of the individual points are significant enough to induce some amount of entropy in the ensembled prediction. These two experiments both provide some evidence that by the end of training, the chain has converged in the sense that the samples all seem to have similar decision boundaries.

3.2 Ensemble Performance

To test the second hypothesis, we simply computed the best ensemble test accuracy and compared it to a no-noise baseline SGD training strategy. The maximum ensemble test accuracy was 92.0% (see Figure 1a), while the no-noise version obtains 93.02% test accuracy. Hence, there is a noticeable degradation in accuracy when noise is injected throughout the training process.

3.3 Ensemble Robustness

For the third hypothesis, we compare how model certainty changes for data drawn from the training distribution (CIFAR-10 images) and data drawn from some other distribution. In this case, we use images drawn from the Street View House Numbers (SVHN) dataset. For each class of each dataset, we compute the average Shannon entropy of the predictions for images in that class.

The results for this experiment are shown in Figure 2. Although there is some intra-dataset variation in prediction entropy for the ensemble, the entropy for the in-distribution data is significantly lower than that of the out-of-distribution data drawn from SVHN. This indicates that ensembling has induced some level of robustness, so that one can use this uncertainty measure as protection mechanism to ensure that the network only makes a prediction when it is sufficiently confident that the data is in-distribution.

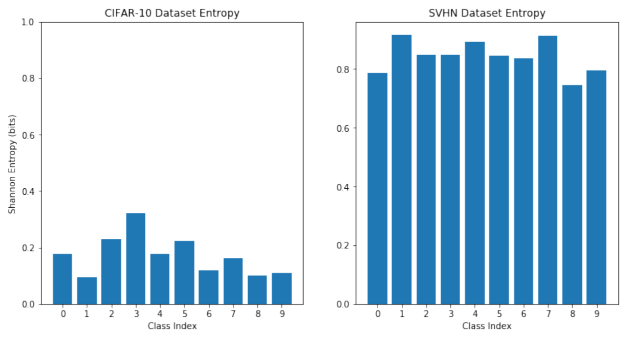


Figure 2: Per-class ensemble entropy for the CIFAR-10 and SVHN datasets. The network was trained on CIFAR-10.

4 Discussion and Limitations

Optimizing neural networks through SGLD gives a reasonable way to create an ensemble of models for uncertainty estimation without incurring additional training cost. Our experiments showed that SGLD samples appear to be close to each other at the end of training because the samples are similar in both underlying feature representation and test accuracy. However, the increasing margin of the network's decision boundary, reflected in the decreasing variance of activations about the class centroid, shows that the Langevin noise has yet to totally dominate the gradient term.

We found that even a low amount of isotropic noise can deteriorate test accuracy compared to a simple SGD baseline. One explanation for this phenomenon may be due to the highly over-parameterized nature of the network. This over-parameterization causes the covariance matrix of the stochastic gradient to be highly degenerate throughout most of training. Adding noise drawn from a distribution with full-rank identity covariance may significantly affect the dynamics of gradient descent in ways that change the local minima to which SGLD can converge [6].

Finally, we found that the Shannon entropy of the ensemble's predictions provided a useful characterization of the ensemble's uncertainty. This is reflected in the noticeable difference in measured entropy between in-distribution and out-of-distribution predictions, as well as the variance in entropy as a function of ensemble size.

One limitation of this work was computational – it would have been better to do additional hyperparameter tuning to find the best learning rate, momentum, and weight decay configurations for a particular noise level. This may have provided a fairer comparison to the no-noise baseline. We also could have run additional experiments to analyze the convergence of the models in parameter space, and to investigate how per-class accuracy correlates with entropy for both the ensemble and the baseline model.

For future work, we may consider testing other variants of Langevin dynamics, e.g. Stochastic Gradient Hamiltonian Monte Carlo, which uses an additional momentum term when injecting noise [1]. Another interesting direction lies in exploring how adding certain types of anisotropic noise may improve convergence speed, e.g. when the covariance of the noise is a low-rank approximation the the Fisher information of the objective function [6].

References

[1] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691, 2014.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[4] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[5] 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)*, pages 681–688, 2011.

[6] Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from minima and regularization effects. *arXiv preprint arXiv:1803.00195*, 2018.