Scalable Natural Gradient Langevin Dynamics in Practice

Henri Palacci ¹ Henry Hess ¹

Abstract

Stochastic Gradient Langevin Dynamics (SGLD) is a sampling scheme for Bayesian modeling adapted to large datasets and models. SGLD relies on the injection of Gaussian Noise at each step of a Stochastic Gradient Descent (SGD) update. In this scheme, every component in the noise vector is independent and has the same scale, whereas the parameters we seek to estimate exhibit strong variations in scale and significant correlation structures, leading to poor convergence and mixing times. We compare different preconditioning approaches to the normalization of the noise vector and benchmark these approaches on the following criteria: 1) mixing times of the multivariate parameter vector, 2) regularizing effect on small dataset where it is easy to overfit, 3) covariate shift detection and 4) resistance to adversarial examples.

1. Introduction

Deep Learning is moving into fields for which errors are potentially lethal, such as self-driving cars, healthcare, and biomedical imaging. For these applications, being able to estimate errors is essential. Bayesian methods provide a way to expand scalar predictions to full posterior probabilities (Gelman et al., 2014). Stochastic Gradient Langevin Dynamics (SGLD), is one of the solutions to the issue of probabilistic modeling on large datasets. Gaussian noise is added to the SGD updates (Welling & Teh, 2011). It was proposed to pre-condition the Gaussian noise with a diagonal matrix to adapt to the changing curvature of the parameter space (Li et al., 2016a). Using a full preconditioning matrix corresponding to the metric tensor of the parameter space was previously proposed (Girolami Mark & Calderhead Ben, 2011), but the computation of this tensor is impossible for large-scale neural networks. It was further proposed to use the Kronecker-factored block diagonal approximation of this tensor, first introduced in (Martens

Copyright 2018 by the author(s).

& Grosse, 2015a) and (Grosse & Martens, 2016) as the preconditioning tensor for the Langevin noise (Nado et al., 2018). Fixed learning rate vanilla gradient descent also introduces noise in the learning process. Hence, fixed learning rate SGD can also be seen as a variant on the same method (Mandt et al., 2017).

In this paper, we conduct a comparison of all these approaches in a practical setting with a fixed hyperparameter optimization budget. We compare these approaches using traditional Markov Chain Monte Carlo (MCMC) diagnostic tools, but will also evaluate the: performance of models in recognizing data points that are not in the sample distribution, the reduction of overfitting in small data settings, and the robustness to adversarial attacks. We find that Langevin approaches, with a reasonable computing budget for hyperparameter tuning, do not improve overfitting or help with adversarial attacks. However, we do find a significant improvement in the detection of out-of-sample data using Langevin methods.

2. Related Work

SGLD was introduced in (Welling & Teh, 2011) and was further refined using a diagonal preconditioning matrix (pS-GLD) in (Li et al., 2016a). The natural gradient method was introduced by (Amari, 1998). Girolami and Calderhead proposed to extend the natural gradient method to neural networks in (Girolami Mark & Calderhead Ben, 2011), and a practical application to probability simplices was presented in (Patterson & Teh, 2013). Finally, the interpretation of fixed rate SGD (FSGD) as a Bayesian approximation was shown in (Mandt et al., 2017). The Kronecker-Factored block-diagonal approximation of the inverse Fisher information matrix was presented for dense layers in (Martens & Grosse, 2015b), then extended to convolutional layers in (Grosse & Martens, 2016). This was used as a preconditioning matrix in SGLD (KSGLD) for smaller scale experiments in (Nado et al., 2018).

3. Preliminaries

3.1. Probabilistic Neural Networks

We consider a supervised learning problem, where we have data $\mathbf{x}_1,...,\mathbf{x}_n \in \mathbb{R}^p$, and labels $y_1,...,y_n$ drawn from a

¹Department of Biomedical Engineering, Columbia University. Correspondence to: Henri Palacci <hp2393@columbia.edu>.

distribution \mathcal{P} . Our goal is to approximate the distribution $p(y|\mathbf{x})$ by empirical risk minimization of a family of distributions parametrized by a vector $\boldsymbol{\theta}$.

In the non-probabilistic setting, this is done by defining an appropriate loss function $\mathcal{L}(y_i|\mathbf{x}_i;\boldsymbol{\theta_i})$ and minimizing it with respect to $\boldsymbol{\theta}$. Optionally, a regularizing term $\mathcal{R}(\boldsymbol{\theta})$ is added to the minimization problem which can therefore be written as: $\hat{\boldsymbol{\theta}} = \arg\max\sum_i -\mathcal{L}(y_i,x_i;\boldsymbol{\theta}) + \mathcal{R}(\boldsymbol{\theta})$. This can be understood as the MAP estimate of the probabilistic model $p(\boldsymbol{\theta}|\mathbf{x}) = p(\boldsymbol{\theta}) \prod_i p(y_i,x_i|\boldsymbol{\theta})$, where $p(\boldsymbol{\theta}|\mathbf{x})$ is the posterior probability of the parameters, $\ln p(\boldsymbol{\theta}) = \mathcal{R}(\boldsymbol{\theta})$ is the log-prior, and $\ln p(y_i,\mathbf{x}_i|\boldsymbol{\theta}) = \mathcal{L}(y_i,x_i;\boldsymbol{\theta})$ is the log-likelihood.

3.2. Stochastic Gradient Langevin Dynamics

The workhorse algorithm for loss minimization for neural networks is mini-batch stochastic gradient descent (SGD). The data $\mathbf{x}_1,...\mathbf{x}_n$ is grouped into mini batches $B_1,...,B_j,...$ of size J such that $(\mathbf{x}_1,...\mathbf{x}_J) \in B_1, (\mathbf{x}_{J+1},...,\mathbf{x}_{2J}) \in B_2,...$

Stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011) updates modifies SGD by adding Gaussian noise at each update step: $\Delta \boldsymbol{\theta}_t = \lambda_t \nabla_{\boldsymbol{\theta}} \left(\log p(\boldsymbol{\theta}) + \sum_j \log p(B_j, \boldsymbol{\theta}) \right) + \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \lambda_t \mathbf{I}).$

3.3. Riemaniann Manifold Langevin Dynamics

The space formed by the parameters of a probability distribution is a Riemaniann manifold (Amari, 1998). Its Riemaniann metric is the Fisher information matrix. This means that the parameter space is curved, and that a local measure of curvature is the Fisher information matrix: $F(\theta) = \mathbb{E}\left[\partial_{\theta}p(y|x;\theta)\partial_{\theta}p(y|x;\theta)^{T}\right]$. Riemaniann Manifold Langevin Dynamics (Marceau-Caron & Ollivier, 2017) preconditions the SGD update with the inverse of the Fisher information matrix: $\Delta\theta_t = F^{-1}\lambda_t\nabla_{\theta}\left(\log p(\theta) + \sum_{j}\log p(B_j,\theta)\right) + F^{-1}\epsilon$. Unfortunately, the computation of the inverse Fisher information matrix is impossible in very high dimensional spaces.

3.4. Kronecker-Factored Approximate Curvature

The Kronecker-Factored Appoximate Curvature (KFAC) is a compact and efficiently invertible block-diagonal approximation of the Fisher information matrix proposed in (Martens & Grosse, 2015a) for dense layers of neural networks and in (Grosse & Martens, 2016) for convolutional layers. Each block corresponds to a layer of the neural network, hence this approximation correctly takes into account within-layer geometric structure. Each layer i's activations a_i can be computed from the previous layer's activations

by a matrix product $s_i = \mathbf{W} a_{i-1}$. A non-linear activation function ϕ such that $a_i = \phi(s_i)$ is applied. The K-FAC approximation can then be written using the Kronecker product \otimes : $\widetilde{F} = \operatorname{diag} (A_1 \otimes G_1, ..., A_i \otimes G_i, ..., A_l \otimes G_l)$, where $A_i = \mathbb{E} \left[a_i a_i^T \right]$ is the estimated covariance matrix of activations for layer i, and $G_i = \mathbb{E} \left[g_i g_i^T \right]$ where $g_i = \nabla_s \mathcal{L}(y, x; \theta)$. We can invert the Kronecker product of two matrices by $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$, and can therefore compute the approximate inverse Fisher information matrix as $\widetilde{F}^{-1} = \operatorname{diag} \left(\left\{ A_i^{-1} \otimes G_i^{-1} \right\}_{i=1...l} \right)$.

3.5. Scalable Natural Gradient Langevin Dynamics

To implement a tractable preconditioning inverse matrix, (Li et al., 2016a) used a diagonal preconditioning matrix rescaling the noise by the inverse of its estimated variance (pS-GLD). Although this improves on SGLD, it still neglects the off-diagonal terms of the metric. A quasi-diagonal approximation was proposed in (Marceau-Caron & Ollivier, 2017). Here, we follow the results presented in (Nado et al., 2018) and use the K-FAC approximation to the inverse Fisher information matrix as our preconditioning matrix:

$$\Delta \boldsymbol{\theta}_{t} = \widetilde{F}^{-1} \lambda_{t} \nabla_{\theta} \left(\log p(\boldsymbol{\theta}) + \sum_{j} \log p(B_{j}, \boldsymbol{\theta}) \right) + \widetilde{F}^{-1} \boldsymbol{\epsilon}$$
(1)

Notice that when changing preconditioning matrices in practice, it is unclear if any improvement in convergence of the algorithms comes from preconditioning the gradient term above, or from preconditioning the noise. It is one of the questions that we aim to answer with our experiments.

3.6. Fixed Learning Rate Stochastic Gradient Descent

It has been suggested that traditional SGD, using a decreasing schedule for the learning rate and early stopping performs Bayesian updates (Mandt et al., 2017). The noise introduced by the variability in the data also prevents the posterior from collapsing to the MAP.

4. Experiments

In order for the model comparisons to be fair, we used the same neural network architecture for all experiments: two convolutional layers with 32 and 64 layers and maxpooling, followed by one dense layer with 1024 units. All nonlinearities are ReLU. The hyperparameter optimization was run using grid search, and the computational time for hyperparameter optimization was limited to 5 times that of the standard SGD algorithm for all other algorithms. Batch size for all experiments was 512.

Note that we did not apply the preconditioning matrix to

the gradient term. It is otherwise impossible to tell if the performance improvements come from better gradient updates in the initial, non-Langevin part of training or from the improvement of the latter, steady-state part of training. Our SGD updates are therefore:

$$\Delta \boldsymbol{\theta}_t = \lambda_t \nabla_{\boldsymbol{\theta}} \left(\log p(\boldsymbol{\theta}) + \sum_j \log p(B_j, \boldsymbol{\theta}) \right) + \widetilde{G} \boldsymbol{\epsilon} \quad (2)$$

Where $G=\mathbf{0}$ for SGD, $G=\mathbf{I}$ for SGLD, G is the diagonal RMSprop matrix for pSGD, $G=\widetilde{F}^{-1}$ for KSGD, and $\lambda_t=\lambda$ for fixed learning rate SGD (FSGD).

4.1. Test Set Accuracy

We first compare the test set accuracy for all methods on 10 epochs of training on the MNIST dataset (LeCun et al., 2010). The results are shown in Figure 1; accuracies for all models are very close and, for a reasonable hyperparameter tuning budget, Bayesian averaging of models does not seem to improve test set accuracy.

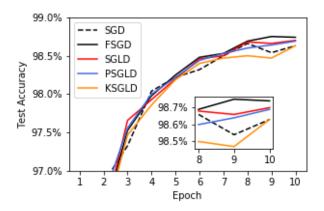


Figure 1. Test set accuracy over ten epochs on the MNIST dataset. SGD: Stochastic Gradient Descent, SGLD: Stochastic Gradient Langevin Dynamics, pSGLD: preconditioned SGLD, KSGLD: K-FAC preconditioned SGLD, FSGD: Fixed rate SGD. Inset: Test set accuracy for the last three epochs.

For the SGLD, pSGLD, and KSGLD methods, the results were very sensitive to the learning rate schedule decrease and most of the hyperparameter optimization computation time was spent on the optimizing it. A longer time spent optimizing the learning rate schedule improved the test rate accuracies slightly.

4.2. Mixing Performance

We approximate (Vats et al., 2015) and estimate the effective sample size as: mESS = $n\left(\frac{|\Lambda|}{|\Sigma|}\right)^{1/p}$, with n the number

of samples in the chain, p the parameter space dimension, $|\Sigma|$ is the covariance matrix of the chain, and $|\Lambda|$ the covariance of matrix of samples. We approximate this by the diagonal approximation of both these matrices, where the ratio of the diagonal terms ess_i is computed as follows $\operatorname{ess}_i = \frac{n}{1+2\sum_k \rho_k}$, where ρ_k is the autocorrelation at $\log k$ truncated to the highest $\log k$ the positive autocorrelation (Gelman et al., 2014).

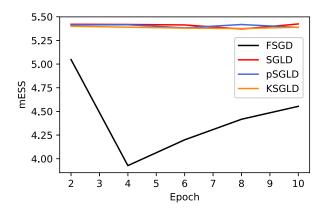


Figure 2. Multivariate Sample Size over epochs for each model over 10 epochs of MNIST training.

The results, shown in Figure 2, all indicate that the MCMC chain mixes poorly in practical settings. Further inspection of the traces shows that almost none of the parameters are stationary. Increasing the run length, or increasing the rate of decrease of the step λ_t , did not improve the aspect of the traces or the effective sample size. These results are consistent with the theoretical analysis of (Betancourt, 2015), who shows that data subsampling is incompatible with any HMC procedure. This is also consistent with (Vollmer et al., 2015) highlighting the problem of stopping while step sizes are still finite.

4.3. Reduction of Overfitting

To test the implicit regularization for the Langevin dynamic models, we truncated the MNIST train set to 5,000 examples (from 60,000). The CNN overfits to the small training set promptly, resulting in decreases in the test set accuracy.

The results, shown in Figure 3, show that the dynamic models underperform SGD on smallMNIST. The only dynamic Bayesian method that matches SGD is SGDA. We hypothesize that adding Gaussian noise on such a small amount of data dramatically deteriorates the initial period of convergence, thus forcing the dynamic Langevin methods to settle for the Langevin period in a local minimum of the loss surface.

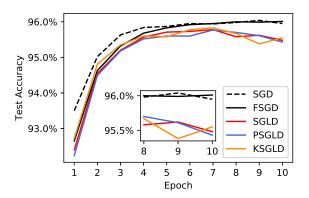


Figure 3. Test set accuracy for all models on ten epochs of training on the reduced MNIST dataset, smallMNIST

4.4. Resistance to Adversarial Attacks

Adversarial attacks are imperceptible modifications to data that cause a model to fail (Goodfellow et al., 2014). We compute adversarial modifications to the test set using the Fast Gradient Sign Method from (Goodfellow et al., 2014). It has previously been shown in (Rawat et al., 2017) that other Bayesian deep learning methods such as Monte Carlo dropout,(Gal & Ghahramani, 2015), Bayes by Backprop (Blundell et al., 2015), matrix variational gaussian (Louizos & Welling, 2016), and probabilistic backpropagation (Hernández-Lobato & Adams, 2015) are vulnerable to adversarial attacks. Our results, presented in Table 1, show that all Langevin dynamic methods also fail to detect adversarial attacks.

Table 1. Classification accuracies for naive Bayes and flexible Bayes on various data sets.

MODEL	TEST ACCURACY	ACCURACY ON ADVERSARIAL EXAMPLES
SGD	96.0	2.9
FSGD	96.5	2.0
SGLD	97.2	1.8
PSGLD	97.1	1.9
KSGLD	97.0	2.0

4.5. Detection of Out of Sample Examples

We assess the epistemic uncertainty inherent in our Bayesian deep neural networks by training it on MNIST but evaluating the network on a completely different dataset, notMNIST (Bulatov). The notMNIST dataset is similar in format to the MNIST dataset, but consists of letters from different fonts.

We expect a network trained on MNIST to give relatively low class probabilities when given examples from the notM-NIST dataset. Figure 4 shows the distribution of the highest probability for each example. Vanilla SGD gives very confident predictions for this dataset, whereas all other methods present a similar distribution of uncertainties. This suggests that Langevin dynamics and fixed learning rate SGD are a relatively straightforward way to detect covariate shift in practical classification tasks.

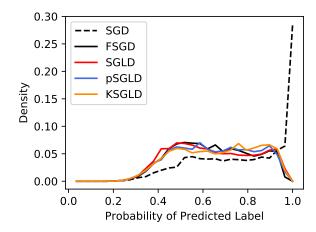


Figure 4. Probability distribution for the most likely class on the notMNIST dataset for all models trained on the MNIST dataset.

5. Discussion

Langevin Stochastic Dynamics provide a scalable way to compute Bayesian posteriors on deep neural network architectures. The noise in stochastic gradient Langevin dynamics is not isotropic due to the geometry of the parameter space. To render the Gaussian noise isotropic, diagonal (Li et al., 2016b), quasi-diagonal (Marceau-Caron & Ollivier, 2017), and block-diagonal (Martens & Grosse, 2015a) approximations have been used. These preconditioning matrices have been proven to work very well as preconditioners for the gradient term, but their use as preconditioners for the Gaussian term in SGLD is subject to significant convergence issues, especially in the transition from the learning phase, where the mini-batch noise dominates.

By contrast, leveraging the mini-batch noise by a constant learning rate to prevent posterior collapse seems to work just as well as the Langevin methods for the experiments described above. This could suggest that the 'data noise' is already appropriately scaled to the manifold structure of the parameter space. This will be the subject of future research.

In practice, our experiments suggest to use Bayesian averaging with a fixed learning rate; this doesn't require any modification to the standard training workflows used by practitioners, and provides implicit protection against covariate shift.

Acknowledgements

This work was supported by the ARO grant "Thermodynamics of Statistical Learning", PI: H. Hess, ARO W911-NF-17-1-0107.

References

- Amari, Shun-Ichi. Natural gradient works efficiently in learning. *Neural Comput.*, 10(2):251–276, February 1998.
- Betancourt, Michael. The fundamental incompatibility of scalable hamiltonian monte carlo and naive data subsampling. In *International Conference on Machine Learning*, pp. 533–540. jmlr.org, June 2015.
- Blundell, Charles, Cornebise, Julien, Kavukcuoglu, Koray, and Wierstra, Daan. Weight uncertainty in neural networks. May 2015.
- Bulatov, Yaroslav. notMNIST dataset. http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html. Accessed: 2018-4-24.
- Gal, Yarin and Ghahramani, Zoubin. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. June 2015.
- Gelman, Andrew, Carlin, John B, Stern, Hal S, Dunson, David B, Vehtari, Aki, and Rubin, Donald B. *Bayesian data analysis*, volume 2. CRC press Boca Raton, FL, 2014.
- Girolami Mark and Calderhead Ben. Riemann manifold langevin and hamiltonian monte carlo methods. *J. R. Stat. Soc. Series B Stat. Methodol.*, 73(2):123–214, March 2011.
- Goodfellow, Ian J, Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. December 2014.
- Grosse, Roger and Martens, James. A kronecker-factored approximate fisher matrix for convolution layers. *arXiv:1602.01407 [cs, stat]*, February 2016.
- Hernández-Lobato, José Miguel and Adams, Ryan P. Probabilistic backpropagation for scalable learning of bayesian neural networks. February 2015.
- LeCun, Yann, Cortes, Corinna, and Burges, C J. MNIST handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.
- Li, Chunyuan, Chen, Changyou, Carlson, David E, and Carin, Lawrence. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *AAAI*, volume 2, pp. 4, 2016a.

- Li, Chunyuan, Chen, Changyou, Carlson, David E, and Carin, Lawrence. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *AAAI*, volume 2, pp. 4, 2016b.
- Louizos, Christos and Welling, Max. Structured and efficient variational deep learning with matrix gaussian posteriors. March 2016.
- Mandt, Stephan, Hoffman, Matthew D, and Blei, David M. Stochastic gradient descent as approximate bayesian inference. *arXiv:1704.04289 [cs, stat]*, April 2017.
- Marceau-Caron, Gaétan and Ollivier, Yann. Natural langevin dynamics for neural networks. arXiv:1712.01076 [cs, stat], December 2017.
- Martens, James and Grosse, Roger. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pp. 2408–2417, 2015a.
- Martens, James and Grosse, Roger. Optimizing neural networks with kronecker-factored approximate curvature. March 2015b.
- Nado, Zachary, Snoek, Jasper, Grosse, Roger, Duvenaud, David, Xu, Bowen, and Martens, James. STOCHAS-TIC GRADIENT LANGEVIN DYNAMICS THAT EX-PLOIT NEURAL NETWORK STRUCTURE. February 2018.
- Patterson, Sam and Teh, Yee Whye. Stochastic gradient riemannian langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems*, pp. 3102–3110, 2013.
- Rawat, Ambrish, Wistuba, Martin, and Nicolae, Maria-Irina. Adversarial phenomenon in the eyes of bayesian deep learning. November 2017.
- Vats, Dootika, Flegal, James M, and Jones, Galin L. Multivariate output analysis for markov chain monte carlo. arXiv:1512.07713 [math, stat], December 2015.
- Vollmer, Sebastian J, Zygalakis, Konstantinos C, and Teh, Yee W. (non-) asymptotic properties of stochastic gradient langevin dynamics. January 2015.
- Welling, Max and Teh, Yee W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 681–688, 2011.