Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks

Presenter: Chunyuan Li

Chunyuan Li[1], Changyou Chen[1], David Carlson[2] and Lawrence Carin[1]

[1]Duke University & [2]Columbia University

Feb. 16, 2016

Outline

Training Deep Neural Networks

- Significant empirical success of Deep Neural Networks
- While SGD with Backpropagation is popular, two issues exit:
  1. Overfitting
     - Make overly confident decisions on prediction
  2. Pathological curvature and nonconvex of parameter space
     - Render optimization difficult to find a good local minima

Incorporating uncertainty

- Bayesian Learning Reduces Overfitting; Incorporation of uncertainty helps improve performance
- Recent works of being Bayesian for deep learning
  1. Early Stop and Dropout have Bayesian interpretation
     - [Duvenaud AISTATS 2016], [Kingma, NIPS 2015]
  2. Variation Inference
     - [Blundell, ICML 2015], [Hernandez, ICML 2015]
  3. Markov Chain Monte Carlo (MCMC)
     - HMC
     - Stochastic Gradient MCMC (SG-MCMC)

Incorporating geometry

1. Higher-order gradient information helps train DNNs when employing optimization methods
   - Quasi-Newton methods
   - Rescale parameters so that the loss function has similar curvature along all directions: Adagrad, Adadelta, Adam and RMSprop algorithms.
2. MCMC
   - Conventional MCMC: Riemann Manifold HMC
   - Consider geometry in SG-MCMC?

Preliminaries

- Given data $\mathcal{D} = \{\boldsymbol{d}_i\}_{i=1}^N$, $\boldsymbol{d}_i$ is *i.i.d.*; model parameters $\boldsymbol{\theta}$

$$\underbrace{p(\boldsymbol{\theta}|\mathcal{D})}_{\text{Posterior}} \propto \underbrace{p(\boldsymbol{\theta})}_{\text{Prior}} \prod_{i=1}^N \underbrace{p(\boldsymbol{d}_i|\boldsymbol{\theta})}_{\text{Likelihood}}$$

For DNNs, $\boldsymbol{d}_i \triangleq (x_i, y_i)$: input $x_i \in \mathbb{R}^D$ and output $y_i \in \mathcal{Y}$.

- Bayesian predictive estimate, for testing input x

$$\boxed{p(y|x, \mathcal{D}) = \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[p(y|x, \boldsymbol{\theta})]} \quad (1)$$

- In optimization, $\boldsymbol{\theta}_{MAP} = \operatorname{argmax} \log p(\boldsymbol{\theta}|\mathcal{D})$.
  The MAP approximates this expectation as

$$p(y|x, \mathcal{D}) \approx p(y|x, \boldsymbol{\theta}_{\text{MAP}}) \quad (2)$$

Parameter uncertainty is ignored.

Preliminaries

- **SG-MCMC**
  - Stochastic Gradient Langevin Dynamics (SGLD)

$$\Delta\boldsymbol{\theta}_t \sim \mathcal{N}\left(\underbrace{\epsilon_t}_{\text{step size}} \underbrace{\left(\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_t) + \frac{N}{n}\sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{d}_{t_i}|\boldsymbol{\theta}_t)\right)}_{\text{stochastic gradient from } \mathcal{D}^t = \{\boldsymbol{d}_{t_1}, \cdots, \boldsymbol{d}_{t_n}\}}, 2\epsilon_t \mathbf{I} \right) \quad (3)$$

  - Monte Carlo approximations to predictive distribution

$$p(y|x, \mathcal{D}) \approx \frac{1}{T}\sum_{t=1}^{T} p(y|x, \boldsymbol{\theta}_t) \quad (4)$$

- Closely related to Stochastic Optimization
  - Stochastic Gradient Descent (SGD)

$$\Delta\boldsymbol{\theta}_t = \epsilon_t \left(\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_t) + \frac{N}{n}\sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{d}_{t_i}|\boldsymbol{\theta}_t)\right) \quad (5)$$

SGRLD

- Stochastic gradient Riemannian Langevin dynamics (SGRLD)

$$\Delta\boldsymbol{\theta}_t \sim \epsilon_t\Big[G(\boldsymbol{\theta}_t)\Big(\nabla_{\boldsymbol{\theta}}\log p(\boldsymbol{\theta}_t) + \frac{N}{n}\sum_{i=1}^{n}\nabla_{\boldsymbol{\theta}}\log p(\boldsymbol{d}_{t_i}|\boldsymbol{\theta}_t)\Big) + \Gamma(\boldsymbol{\theta}_t)\Big] + G^{\frac{1}{2}}(\boldsymbol{\theta}_t)\mathcal{N}(0, 2\epsilon_t\mathbf{I}) \quad (6)$$

- What's new in SGRLD?
  - $G(\boldsymbol{\theta}_t)$: **preconditioner** (*e.g.*, preconditioning matrix)
  - $\Gamma_i(\boldsymbol{\theta}) = \sum_j \frac{\partial G_{i,j}(\boldsymbol{\theta})}{\partial \theta_j}$: change of manifold curvature.
  - In SGLD, $G(\boldsymbol{\theta}_t) = \mathbf{I}$, and $\Gamma(\boldsymbol{\theta}_t)$ valishes.

- Problem: $G(\boldsymbol{\theta}_t)$ is usually intractable

RMSprop as the Preconditioner

- $\bar{g}(\boldsymbol{\theta}_t; \mathcal{D}^t) = \frac{1}{n}\sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{d}_{t_i}|\boldsymbol{\theta}_t)$: sample mean of gradient.
- Our preconditioner is updated using only the current gradient, and only estimates a diagonal matrix

$$V(\boldsymbol{\theta}_{t+1}) = \alpha V(\boldsymbol{\theta}_t) + (1-\alpha)\bar{g}(\boldsymbol{\theta}_t; \mathcal{D}^t) \odot \bar{g}(\boldsymbol{\theta}_t; \mathcal{D}^t) \ , \tag{7}$$

$$G(\boldsymbol{\theta}_{t+1}) = \mathrm{diag}\left(\mathbf{1} \oslash \left(\lambda \mathbf{1} + \sqrt{V(\boldsymbol{\theta}_{t+1})}\right)\right) \tag{8}$$

- Intuitive interpretations:
  1. The preconditioner equalizes the gradient so that a constant stepsize is adequate for all dimensions.
  2. The stepsizes are adaptive, in that flat directions have larger stepsizes while curved directions have smaller stepsizes.

Finite-time Error Analysis

- Task: for a testing function $\phi(\boldsymbol{\theta})$
  - True posterior expectation $\bar{\phi} = \int_{\mathcal{X}} \phi(\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}$
  - MC Estimator: $\hat{\phi} = \frac{1}{S_T} \sum_{t=1}^{T} \epsilon_t \phi(\boldsymbol{\theta}_t)$ at time $S_T = \sum_{t=1}^{T} \epsilon_t$

Theorem 1: MSE bound

$$\text{MSE} : \mathbb{E}\left[\left(\hat{\phi} - \bar{\phi}\right)^2\right] \leq \mathcal{B}_{\text{mse}} \tag{9}$$

$$\triangleq C \left(\underbrace{\sum_t \frac{\epsilon_t^2}{S_T^2} \mathbb{E} \left\| \Delta V_t \right\|^2}_{\text{Estimation error of stochastic gradients}} + \underbrace{\frac{1}{S_T} + \frac{(\sum_{t=1}^{T} \epsilon_t^2)^2}{S_T^2}}_{\text{discretization error of numerical integrators}} \right)$$

- Asymptotic convergence ($S_T \to \infty$):
  Decreasing-step-size pSGLD is asymptotically consistent with true posterior expectation.

Bias-Variance Tradeoff

- Risk of Estimator $\mathbb{E}[(\bar{\phi} - \hat{\phi})^2] = B^2 + V$.

$$\textbf{Bias}: \; B = \bar{\phi}_\eta - \bar{\phi} \tag{10}$$

$$\textbf{Variance}: \; V = \mathbb{E}[(\bar{\phi}_\eta - \hat{\phi})^2] \tag{11}$$

  where $\bar{\phi}_\eta = \int_{\mathcal{X}} \phi(\boldsymbol{\theta})\rho_\eta(\boldsymbol{\theta})d\boldsymbol{\theta}$ as the ergodic average under the invariant measure, $\rho_\eta(\boldsymbol{\theta})$, of the pSGLD.

- Increase *ESS* or decrease *autocorrelation time* leads to better estimation

$$V \propto \frac{1}{\text{effective sample size (ESS)}} \propto \text{autocorrelation time}$$

Two Practical Techniques

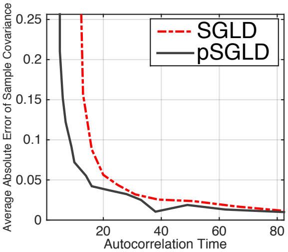
1. Excluding $\Gamma(\boldsymbol{\theta}_t)$ term
   - Corollary 1: ignoring $\Gamma(\boldsymbol{\theta}_t)$ produces a bias controlled by α on the MSE
   - More samples per unit time are generated, resulting in a smaller variance on the estimation
   - Dropped in [Ahn et al, ICML 2012] and [Teh et al, 2015]
2. Thinning
   - Corollary 2: MSE remains the same form.
   - These thinned samples have a lower autocorrelation time and can have a similar ESS.

Algorithm: Practical pSGLD is RMSprop with a Gaussian noise, whose variance is proportion to the preconditioner.

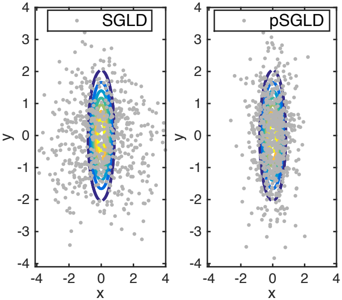
[Ahn et al, ICML 2012] *Bayesian posterior sampling via stochastic gradient fisher scoring*
[Teh et al, 2015] *Distributed Bayesian learning with expectation propagation and posterior server*

Simulation: 2D distribution

- $N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.16 & 0 \\ 0 & 1 \end{bmatrix}\right)$. The goal is to estimate the covariance matrix.
- pSGLD dominates the "vanilla" SGLD in that it consistently shows a lower error and autocorrelation time, particularly with larger stepsize.
- pSGLD can adapt stepsizes acorrding to the geometry of different dimensions.



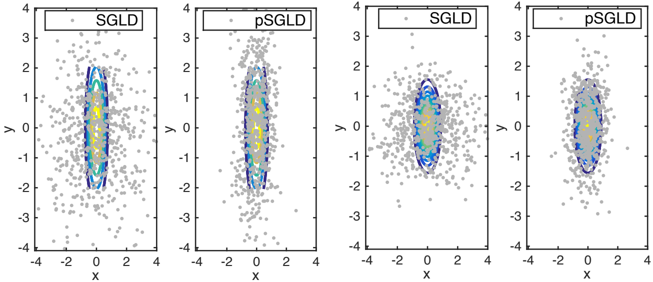
(a) Error and autocorrelation time



(b) Samples

Simulation: 2D distribution

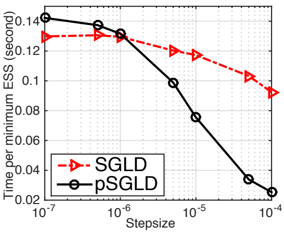
- Even if the covariance matrix of a target distribution is mildly rescaled, we do not have to choose a new stepsize for pSGLD.



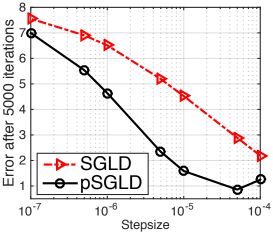
(a) Scale up $a = 2, \epsilon = 0.3$ (b) Scale down $a = 0.5, \epsilon = 0.3$

Exp. 1: Bayesian Logistic Regression

- pSGLD generates much larger ESS compared to SGLD, especially when the stepsize is large. Meanwhile, pSGLD provides smaller error in estimating weights
- Though pSGLD takes a bit more time to compute preconditioner, this is compensated by obtaining more effective samples in given time. Therefore, the variance in risk of prediction is reduced.



(a) Variance



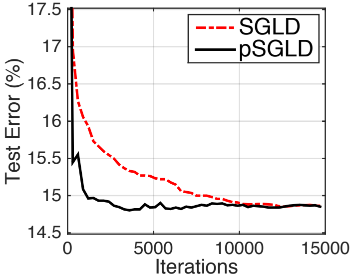
(b) Parameter estimation

Exp. 1: Bayesian Logistic Regression

- Settings
  - a9a dataset: $N_{\text{train}} = 32561, N_{\text{test}} = 16281$, minibatch size = 50.
  - pSGLD converges in less than 4×10^3 iterations, while SGLD at least needs double the time to reach this accuracy.
  - Comparable with recent advances in stochastic gradient variation inference
- Results

Table: Test error on a9a.

Method	Test error
pSGLD	14.86%
SGLD	14.86%
DSVI†	15.20%
L-BFGS-SGVI‡	14.91%
HFSGVI‡	15.16%



[†] *Doubly Stochastic Variational Bayes for non-Conjugate Inference*, Titsias et al. ICML 2014
[‡] *Fast 2nd Order Stochastic Backpropagation for Variational Inference*, Fan et al. NIPS 2015

Exp. 2: Feedforward Neural Networks

- Settings: ReLU, 784-X-X-10, minibatch size = 100. After burnin and thinning, 30 samples yield good esitmates
- Results
  - SG-MCMC methods are better than their corresponding stochastic optimization counterparts
  - Higher uncertainty leads to lower errors
  - distilled pSGLD* can maintain good results

Table: Classification error of FNN on MNIST.

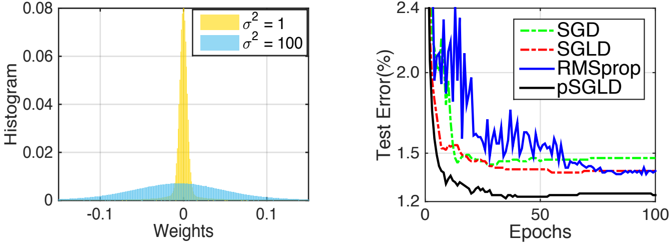
Method	Test Error 400-400	Test Error 800-800	Test Error 1200-1200
pSGLD ($\sigma^2 = 100$)	**1.40%**	**1.26%**	**1.14%**
pSGLD ($\sigma^2 = 1$)	1.45%	1.32%	1.24%
distilled pSGLD	1.44%	1.40%	1.41%
SGLD	1.64%	1.41%	1.40%
RMSprop	1.59%	1.43%	1.39%
RMSspectral	1.65%	1.56%	1.46%
SGD	1.72%	1.47%	1.47%
BPB, Gaussian◇	1.82%	1.99%	2.04%
BPB, Scale mixture◇	1.32%	1.34%	1.32%
SGD, dropout◇	1.51%	1.33%	1.36%

[◇] *Weight Uncertainty in Neural Networks*, Blundell et al. ICML 2015

[*] *Bayesian Dark Knowledge*, Korattikara et al. NIPS 2015

Exp. 2: Feedforward Neural Networks

- Weights: Smaller variance in the prior imposes lower uncertainty, by making the weights concentrate to 0; while larger variance in the prior leads to a wider range of weight choices, thus higher uncertainty.
- Converge: pSGLD consistently converges faster and to a better point than SGLD



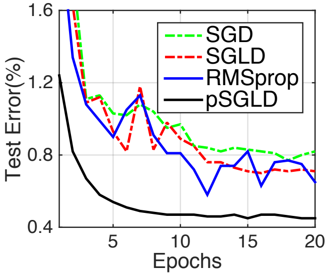
(a) Weights distribution (b) Learning curves

Figure: FNN of size 1200-1200 on MNIST.

Exp. 3: Convolutional Neural Networks

- LeNet: 2 covolutional layers: 5×5 filter size with 32 and 64 channels
- Comparable with some recent state-of-the-art CNN based systems

Method	Test error
pSGLD	**0.45%**
SGLD	0.71%
RMSprop	0.65%
RMSspectral	0.78%
SGD	0.82%
Stochastic Pooling	0.47%
NIN + Dropout	0.47%
MN + Dropout	0.45%



Summary

- Algorithms
  - pSGLD: preconditioned stochastic gradient Langevin dynamics
  - Error analysis and practical techniques
- Applications:
  - Model uncertainty in deep neural networks

Questions?