High-Order Stochastic Gradient Thermostats for Bayesian Learning of Deep Models

Chunyuan Li, Changyou Chen, Kai Fan and Lawrence Carin

Duke University, Durham NC 27708, USA

Introduction

Objective Designing simple and efficient Bayesian inference algorithms for deep learning models.

Main idea

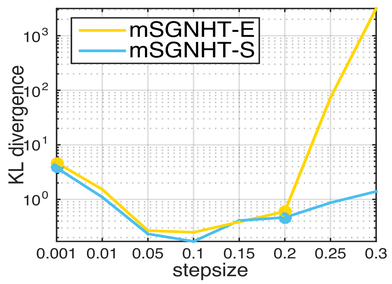
- Multivariate Stochastic Gradient Nośe-Hoover Thermostat (m-SGNHT) with more accurate numerical implementation.
- From **E**uler integrator to **S**ymmetric Splitting integrator

Illustrations

Double-well Potential Function

$$\rho(\theta) \propto \exp(-U(\theta)),\ U(\theta) = (\theta+4)(\theta+1)(\theta-1)(\theta-3)/14 + 0.5$$

- mSGNHT-S consistently provides a better approximation.
- The significant gap at larger stepsize reveals that mSGNHT-S allows large updates.



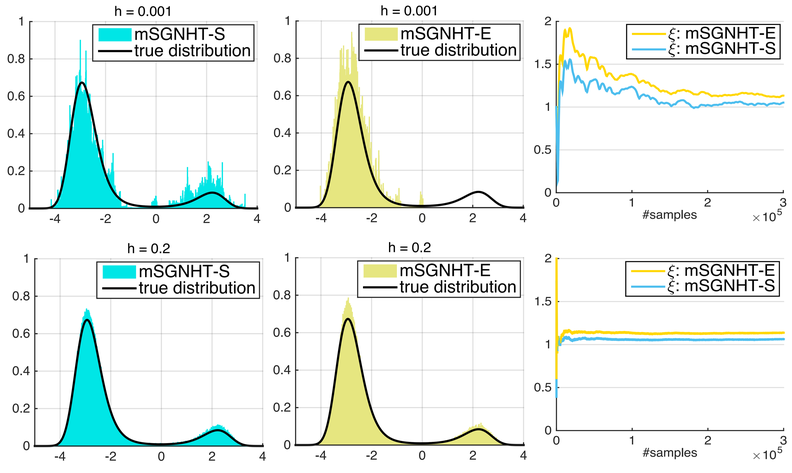


Figure: Samples of $\rho(\theta)$ with SSI (1st column) and Euler integrator (2nd column), and the estimated thermostat variable over iterations (3rd column).

Convolutional Neural Networks

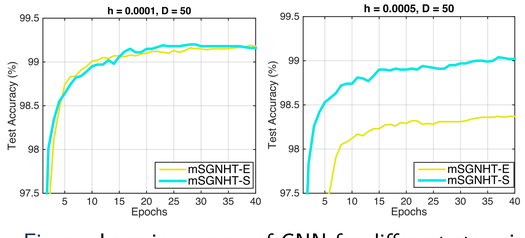


Figure: Learning curves of CNN for different step sizes.

Algorithms

$\boldsymbol{\theta}$: model parameter $\quad \boldsymbol{p}$: momentum $\quad \boldsymbol{\xi}$: thermostat

Euler Integrator

$$\begin{cases} \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \boldsymbol{p}_t h \\ \boldsymbol{p}_{t+1} = \boldsymbol{p}_t - \nabla_{\boldsymbol{\theta}} \tilde{U}_t(\boldsymbol{\theta}_{t+1}) h - \text{diag}(\boldsymbol{\xi}_t)\boldsymbol{p}_t h + \sqrt{2D}\boldsymbol{\zeta}_{t+1} \\ \boldsymbol{\xi}_{t+1} = \boldsymbol{\xi}_t + (\boldsymbol{p}_{t+1} \odot \boldsymbol{p}_{t+1} - 1)\, h \end{cases}$$

Symmetric Splting Integrator

$$A: \boldsymbol{\theta}_{t+1/2} = \boldsymbol{\theta}_t + \boldsymbol{p}_t h/2,\ \boldsymbol{\xi}_{t+1/2} = \boldsymbol{\xi}_t + (\boldsymbol{p}_t \odot \boldsymbol{p}_t - 1)\, h/2 \rightarrow$$
$$B: \boldsymbol{p}_{t+1/3} = \exp(-\boldsymbol{\xi}_{t+1/2} h/2) \odot \boldsymbol{p}_t \rightarrow$$
$$O: \boldsymbol{p}_{t+2/3} = \boldsymbol{p}_{t+1/3} - \nabla_{\boldsymbol{\theta}} \tilde{U}_t(\boldsymbol{\theta}_{t+1/2}) h + \sqrt{2D}\boldsymbol{\zeta}_{t+1} \rightarrow$$
$$B: \boldsymbol{p}_{t+1} = \exp(-\boldsymbol{\xi}_{t+1/2} h/2) \odot \boldsymbol{p}_{t+2/3} \rightarrow$$
$$A: \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_{t+1/2} + \boldsymbol{p}_{t+1} h/2,\ \boldsymbol{\xi}_{t+1} = \boldsymbol{\xi}_{t+1/2} + (\boldsymbol{p}_{t+1} \odot \boldsymbol{p}_{t+1} - 1)\, h/2$$

where h is stepsize, D is diffusion factor, and $\boldsymbol{\zeta}$ is Gaussian noise

Theoretical Justification

Roles of numerical integrators Under certain assumptions, the *Bias* and *MSE* of a SG-MCMC algorithm with stepsize h and a Kth-order integrator are:

$$\textit{Bias:}\ \left|\mathbb{E}\hat{\phi} - \bar{\phi}\right| = \mathcal{B}_{\text{bias}} + O(h^K)$$
$$\textit{MSE:}\ \mathbb{E}\left(\hat{\phi} - \bar{\phi}\right)^2 = \mathcal{B}_{\text{mse}} + O(h^{2K}),$$

where $\mathcal{B}_{\text{bias}}$ and $\mathcal{B}_{\text{mse}}$ are functions depending on (h, T) but independent of K.

Remark 1 Robustness: The bias and MSE of mSGNHT-S is bounded as: $\mathcal{B}_{\text{bias}}+O(h^2)$ and $\mathcal{B}_{\text{mse}}+O(h^4)$, compared to $\mathcal{B}_{\text{bias}}+O(h)$ and $\mathcal{B}_{\text{mse}}+O(h^2)$ for the mSGNHT-E, respectively. This indicates that mSGNHT-S is more robust to the stepsizes than mSGNHT-E.

Remark 2 Convergence Rate: The higher order a numerical integrator is, the faster its optimal convergence rate is. Convergence rates in term of *bias* for mSGNHT-S and mSGNHT-E are $T^{-2/3}$ and $T^{-1/2}$, respectively, indicating mSGNHT-S converges faster.

Remark 3 Measure Accuracy: In the limit of infinite time ($T \rightarrow \infty$), the terms $\mathcal{B}_{\text{bias}}$ and $\mathcal{B}_{\text{mse}}$ in Lemma 1 vanish, leaving only the $O(h^K)$ terms. This indicates mSGNHT-S is an order of magnitude more accurate than mSGNHT-E.

Advantages for Deep Learning

- It tolerates gradients of various magnitudes, providing a potential solution to mitigate the *vanishing/exploding gradients* problem
- It can estimate model parameters faster and more accurately

Experiments

I. Shallow Models

Latent Dirichlet Allocation and **Logistic Regression**

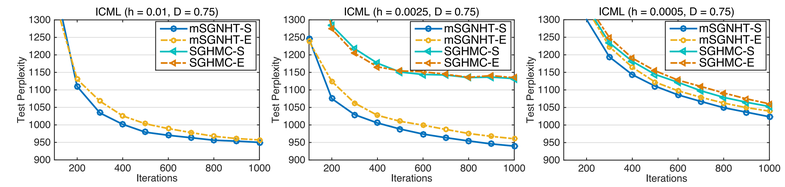


Figure: Learning curves of LDA on `ICML` dataset for different stepsize h.

Table: LDA on `ICML`.

Method	Test Perplexity ↓
mSGNHT-S	939.67
mSGNHT-E	960.56
SGHMC-S	1004.73
SGHMC-E	1017.51
SGRLD	1154.68
Gibbs	907.84

Table: LR on a9a.

Method	Test Accuracy ↑
mSGNHT-S	84.95%
mSGNHT-E	84.72%
SGHMC-S	84.56%
SGHMC-E	84.51%
DSVI	84.80%
HFSGVI	84.84%

II. Deep Models

Deep Neural Networks

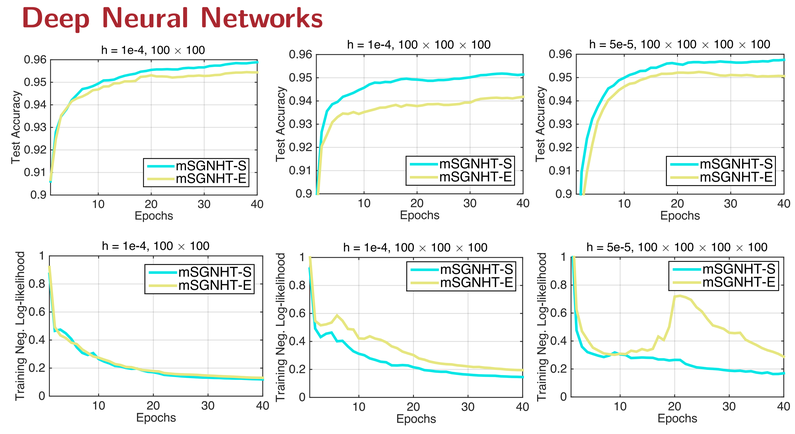


Figure: Learning curves of FNN with different depth on MNIST dataset.

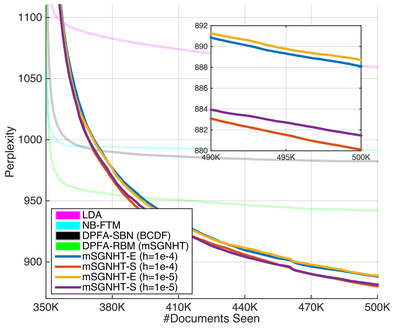
mSGNHT-E is failing as learning progresses. It starts to work, only because the stepsize is decreased by half.

Deep Poisson Factor Analysis

Perplexity on Wikipedia dataset:

$$\mathbf{W} \sim \text{Pois}(\boldsymbol{\Phi}(\boldsymbol{\Psi} \odot \boldsymbol{h}^{(1)})),$$
$$\boldsymbol{h}^{(1)} \sim \text{DSBN}(\boldsymbol{h}^{\{(2),\cdots,(L)\}})$$

A 3-layer Deep Sigmoid Belief Networks (DSBN) is inferred by mSGNHT.



Acknowledgements

This research was supported by ARO, DARPA, DOE, NGA ONR and NSF.