Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning

Ruqi Zhang [1], Chunyuan Li [2], Jianyi Zhang [3], Changyou Chen [4], Andrew Gordon Wilson [5]

[1]Cornell University, [2]Microsoft Research, [3]Duke University, [4]University at Buffalo, [5]New York University

Question 1

Imagine that you travel to Seattle and want to know more about this city.

Where will you go?

Question 1

Imagine that you travel to Seattle and want to know more about this city.

Where will you go?

Question 1

Imagine that you travel to Seattle and want to know more about this city.

Where will you go?

Question 1

Imagine that you travel to Seattle and want to know more about this city.

Where will you go?

Question 1

Imagine that you travel to Seattle and want to know more about this city.

Where will you go?

Question 1

Imagine that you travel to Seattle and want to know more about this city.

Where will you go?

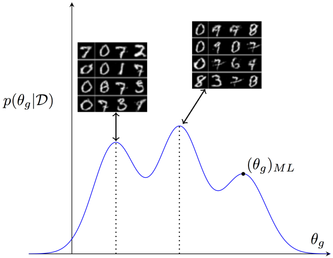
Question 1

Imagine that you travel to Seattle and want to know more about this city.

Where will you go?

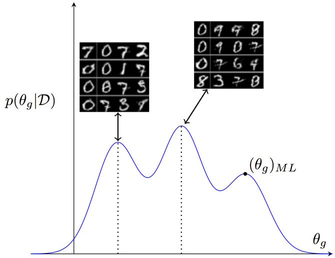
Answer: explore as many places as you can

Why Bayesian Deep Learning



Each mode corresponds to a different explanation (credit: [Saatchi and Wilson, 2017])

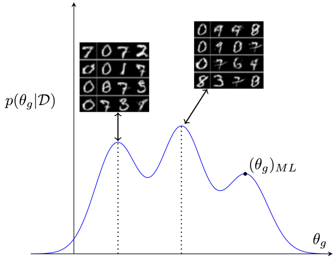
Why Bayesian Deep Learning



Each mode corresponds to a different explanation (credit: [Saatchi and Wilson, 2017])

- Optimization methods use a single point estimate

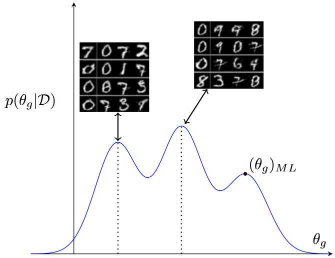
Why Bayesian Deep Learning



Each mode corresponds to a different explanation (credit: [Saatchi and Wilson, 2017])

- Optimization methods use a single point estimate
- The loss, and posterior, do not strongly favour any one solution

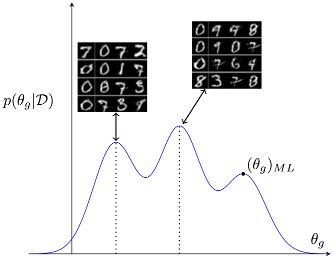
Why Bayesian Deep Learning



Each mode corresponds to a different explanation (credit: [Saatchi and Wilson, 2017])

- Optimization methods use a single point estimate
- The loss, and posterior, do not strongly favour any one solution
- Parameters across different modes provide complementary explanations of the data

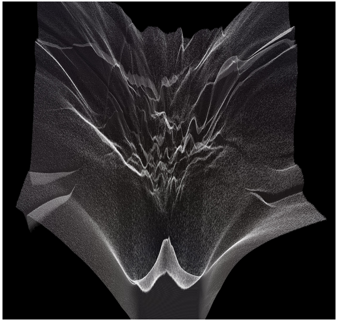
Why Bayesian Deep Learning



Each mode corresponds to a different explanation (credit: [Saatchi and Wilson, 2017])

- Optimization methods use a single point estimate
- The loss, and posterior, do not strongly favour any one solution
- Parameters across different modes provide complementary explanations of the data
- Combine these explanations for better accuracy and calibration

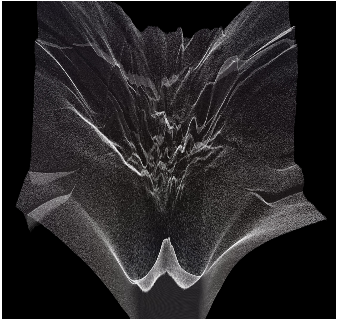
Why cSG-MCMC



Loss surface in deep learning
(credit: losslandscape.com)

Why cSG-MCMC

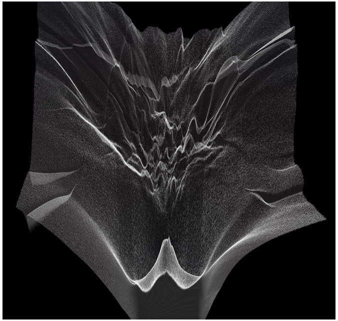
- Specifically designed to explore complex multimodal posteriors



Loss surface in deep learning (credit: losslandscape.com)

Why cSG-MCMC

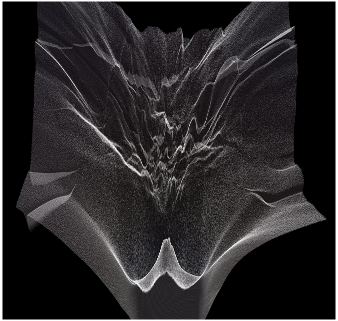
- Specifically designed to explore complex multimodal posteriors
- Works for modern architectures at ImageNet scale



Loss surface in deep learning
(credit: losslandscape.com)

Why cSG-MCMC

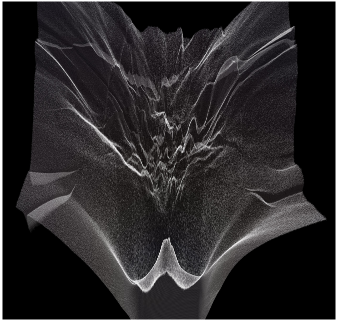
- Specifically designed to explore complex multimodal posteriors
- Works for modern architectures at ImageNet scale
- Samples more efficiently even from unimodal posteriors



Loss surface in deep learning (credit: losslandscape.com)

Why cSG-MCMC

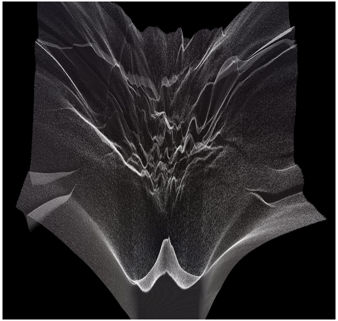
- Specifically designed to explore complex multimodal posteriors
- Works for modern architectures at ImageNet scale
- Samples more efficiently even from unimodal posteriors
- No significant overhead compared to standard SGD training



Loss surface in deep learning (credit: losslandscape.com)

Why cSG-MCMC

- Specifically designed to explore complex multimodal posteriors
- Works for modern architectures at ImageNet scale
- Samples more efficiently even from unimodal posteriors
- No significant overhead compared to standard SGD training



Loss surface in deep learning (credit: losslandscape.com)

Code: `https://github.com/ruqizhang/csgmcmc`

Learning on Bayesian Neural Networks

Goal: get the posterior distribution of the weights $p(\theta|\mathcal{D})$

Learning on Bayesian Neural Networks

Goal: get the posterior distribution of the weights $p(\theta|\mathcal{D})$

- $p(\theta|\mathcal{D})$ is complex and multimodal

Learning on Bayesian Neural Networks

Goal: get the posterior distribution of the weights $p(\theta|\mathcal{D})$

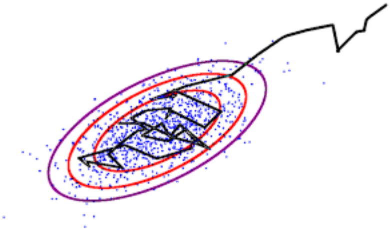
- $p(\theta|\mathcal{D})$ is complex and multimodal
- Different modes are corresponding to different representations

Learning on Bayesian Neural Networks

Goal: get the posterior distribution of the weights $p(\theta|\mathcal{D})$

- $p(\theta|\mathcal{D})$ is complex and multimodal
- Different modes are corresponding to different representations

Solution: Markov chain Monte Carlo (MCMC)



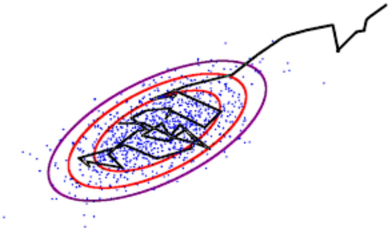
Learning on Bayesian Neural Networks

Goal: get the posterior distribution of the weights $p(\theta|\mathcal{D})$

- $p(\theta|\mathcal{D})$ is complex and multimodal
- Different modes are corresponding to different representations

Solution: Markov chain Monte Carlo (MCMC)

$+$ Asymptotically unbiased



Learning on Bayesian Neural Networks

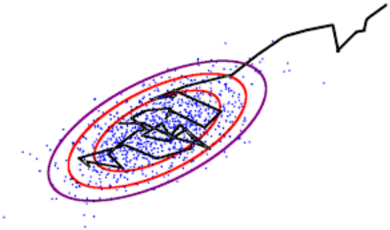
Goal: get the posterior distribution of the weights $p(\theta|\mathcal{D})$

- $p(\theta|\mathcal{D})$ is complex and multimodal
- Different modes are corresponding to different representations

Solution: Markov chain Monte Carlo (MCMC)

$+$ Asymptotically unbiased

$+$ Gold standard on small neural networks [Neal, 1996]



Learning on Bayesian Neural Networks

Goal: get the posterior distribution of the weights $p(\theta|\mathcal{D})$

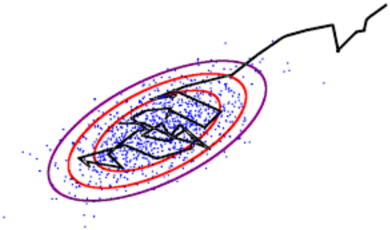
- $p(\theta|\mathcal{D})$ is complex and multimodal
- Different modes are corresponding to different representations

Solution: Markov chain Monte Carlo (MCMC)

+ Asymptotically unbiased

+ Gold standard on small neural networks [Neal, 1996]

– High computational cost



Learning on Bayesian Neural Networks

Goal: get the posterior distribution of the weights $p(\theta|\mathcal{D})$

- $p(\theta|\mathcal{D})$ is complex and multimodal
- Different modes are corresponding to different representations

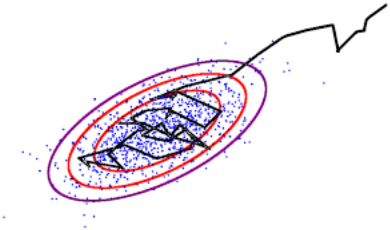
Solution: Markov chain Monte Carlo (MCMC)

+ Asymptotically unbiased

+ Gold standard on small neural networks [Neal, 1996]

– High computational cost

– Slow mixing



Learning on Bayesian Neural Networks

Goal: get the posterior distribution of the weights $p(\theta|\mathcal{D})$

- $p(\theta|\mathcal{D})$ is complex and multimodal
- Different modes are corresponding to different representations

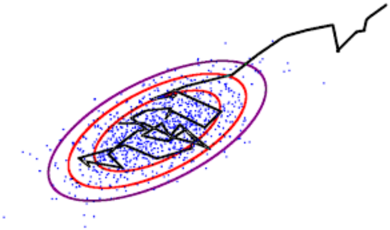
Solution: Markov chain Monte Carlo (MCMC)

+ Asymptotically unbiased

+ Gold standard on small neural networks [Neal, 1996]

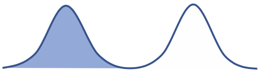
– High computational cost

– Slow mixing



How to make MCMC efficiently explore a highly multimodal parameter space?

SG-MCMC

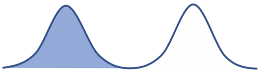


- Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC):
  use stochastic gradients in Langevin dynamics to reduce cost of each iteration

$$\theta_{k+1} = \theta_k - \alpha_k \nabla \tilde{U}(\theta) + \sqrt{2\alpha_k \epsilon}, \text{ where } \epsilon \sim \mathcal{N}(0, I)$$

[Welling and Teh, 2011]

SG-MCMC



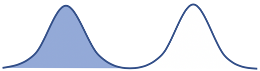
- Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC):
  use stochastic gradients in Langevin dynamics to reduce cost of each iteration

$$\theta_{k+1} = \theta_k - \alpha_k \nabla \tilde{U}(\theta) + \sqrt{2\alpha_k \epsilon}, \text{ where } \epsilon \sim \mathcal{N}(0, I)$$

[Welling and Teh, 2011]

- Previous work: introduce momentum variables [Chen et al.,2014] and preconditioners [Li et al.,2016]

SG-MCMC



- Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC):
  use stochastic gradients in Langevin dynamics to reduce cost of each iteration

$$\theta_{k+1} = \theta_k - \alpha_k \nabla \tilde{U}(\theta) + \sqrt{2\alpha_k \epsilon}, \text{ where } \epsilon \sim \mathcal{N}(0, I)$$

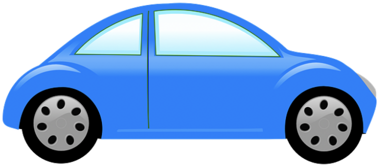
[Welling and Teh, 2011]

- Previous work: introduce momentum variables [Chen et al.,2014] and preconditioners [Li et al.,2016]

Slow mixing: not efficient to explore multimodal distributions of DNNs

Question 2

How do you efficiently explore the city? By car or on foot?



Problem Analysis

Stepsize is the key!

- SG-MCMC requires a decaying stepsize to control error
- A small stepsize leads to slow mixing

Problem Analysis

Stepsize is the key!

- SG-MCMC requires a decaying stepsize to control error
- A small stepsize leads to slow mixing

Stepsize controls SG-MCMC's behavior in two ways:

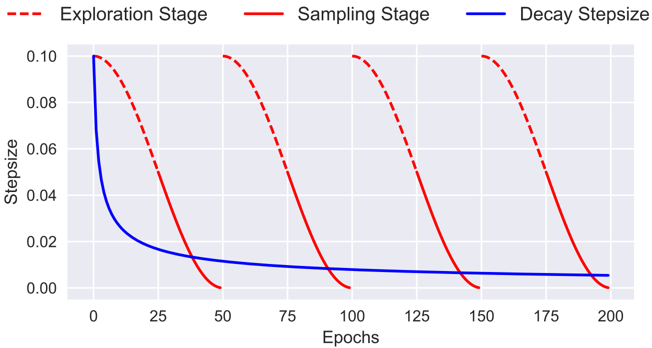
- magnitude to drift towards high density regions
- the level of injecting noise

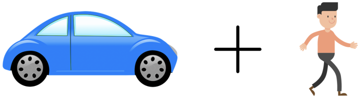
$$\theta_{k+1} = \theta_k - \alpha_k \nabla \tilde{U}(\theta) + \sqrt{2\alpha_k}\epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, I)$$

A small stepsize reduces both abilities

Our solution

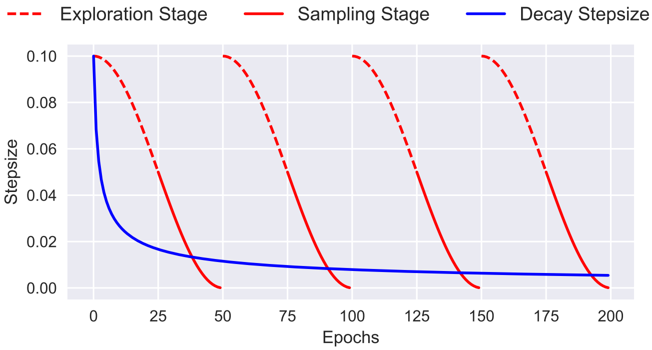
- Cyclical stepsize schedule





Our solution

- Cyclical stepsize schedule



- cSG-MCMC operates in two stages: (i) Exploration: encourage the sampler to explore the parameter space with large stepsizes (ii) Sampling: characterize the fine-scale local density with small stepsizes

Cyclical SG-MCMC Details

Introduce a system temperature T to control the sampler's behaviour

Cyclical SG-MCMC Details

Introduce a system temperature T to control the sampler's behaviour

- Exploration: use $T = 0$ to converge quickly

Cyclical SG-MCMC Details

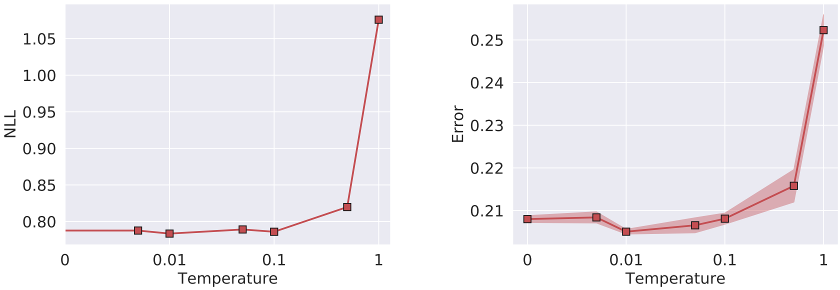
Introduce a system temperature T to control the sampler's behaviour

- Exploration: use $T = 0$ to converge quickly
- Sampling: use $0 < T < 1$ to improve performance

Cyclical SG-MCMC Details

Introduce a system temperature T to control the sampler's behaviour

- Exploration: use $T = 0$ to converge quickly
- Sampling: use $0 < T < 1$ to improve performance



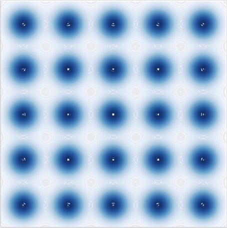
Non-Asymptotic Analysis

- We provide analysis of weak convergence in terms of bias and MSE, and convergence under the Wasserstein distance

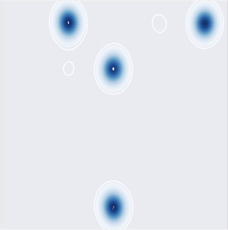
Non-Asymptotic Analysis

- We provide analysis of weak convergence in terms of bias and MSE, and convergence under the Wasserstein distance
- Takeaway: cSG-MCMC has the same order of dependency on K as SG-MCMC, but can have an overall faster convergence rate due to a better trade-off between bias and variance

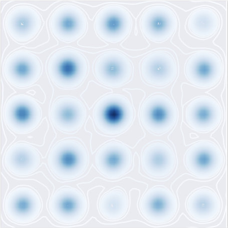
Mixture of 25 Gaussians



(a) Target



(b) SGLD



(c) cSGLD (ours)

- Whereas SGLD gets trapped in some local modes, cSGLD is able to find and characterize all modes

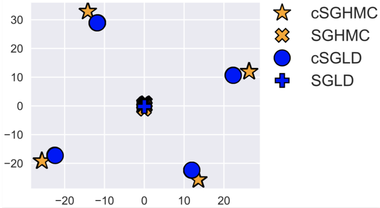
Bayesian Neural Networks

	CIFAR-10	CIFAR-100
SGD	5.29±0.15	23.61±0.09
SGDM	5.17±0.09	22.98±0.27
Snapshot-SGD	4.46±0.04	20.83±0.01
Snapshot-SGDM	4.39±0.01	20.81±0.10
SGLD	5.20±0.06	23.23±0.01
cSGLD (ours)	4.29±0.06	20.55±0.06
SGHMC	4.93±0.1	22.60±0.17
cSGHMC (ours)	**4.27**±0.03	**20.50**±0.11

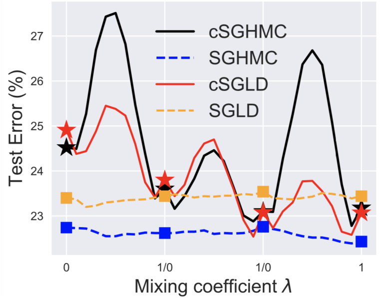
Table 1: Comparison of test error (%).

cSG-MCMC outperforms SG-MCMC and optimization methods.

Visualization in weight space and prediction space



(a) Weight space (MDS)



(b) Prediction space (interpolation)

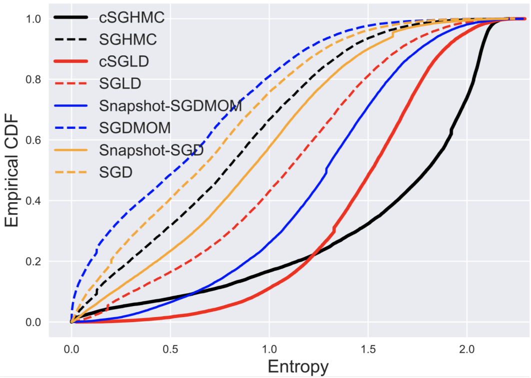
- Samples from cSG-MCMC are diverse in weight space and prediction space

ImageNet

	NLL $\downarrow$	Top1 $\uparrow$	Top5 $\uparrow$
SGDM	0.9595	76.046	92.776
Snapshot-SGDM	0.8941	77.142	93.344
SGHMC	0.9308	76.274	92.994
cSGHMC	**0.8882**	77.114	93.524

- cSG-MCMC gives the lowest testing NLL

Uncertainty Estimate



- Train on MNIST dataset and test on notMNIST dataset
- cSG-MCMC gives the best uncertainty estimate

Summary

- Bayesian neural networks involve multimodal posteriors corresponding to different representations
- We propose cSG-MCMC to efficiently explore these complex multimodal distributions
- cSG-MCMC is simple to implement and no computational overhead
- We prove non-asymptotic convergence of our method.
- We provide promising empirical results, including experiments on ImageNet

Code: `https://github.com/ruqizhang/csgmcmc`

Thank you!