Google AI
Brain Team

Code: github.com/google-research/google-research/tree/master/cold_posterior_bnn

How Good is the Bayes Posterior in Deep Neural Networks Really?

Florian Wenzel (Google Research Berlin)

Joint first authors: Kevin Roth, Bas Veeling,
and: Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, Sebastian Nowozin

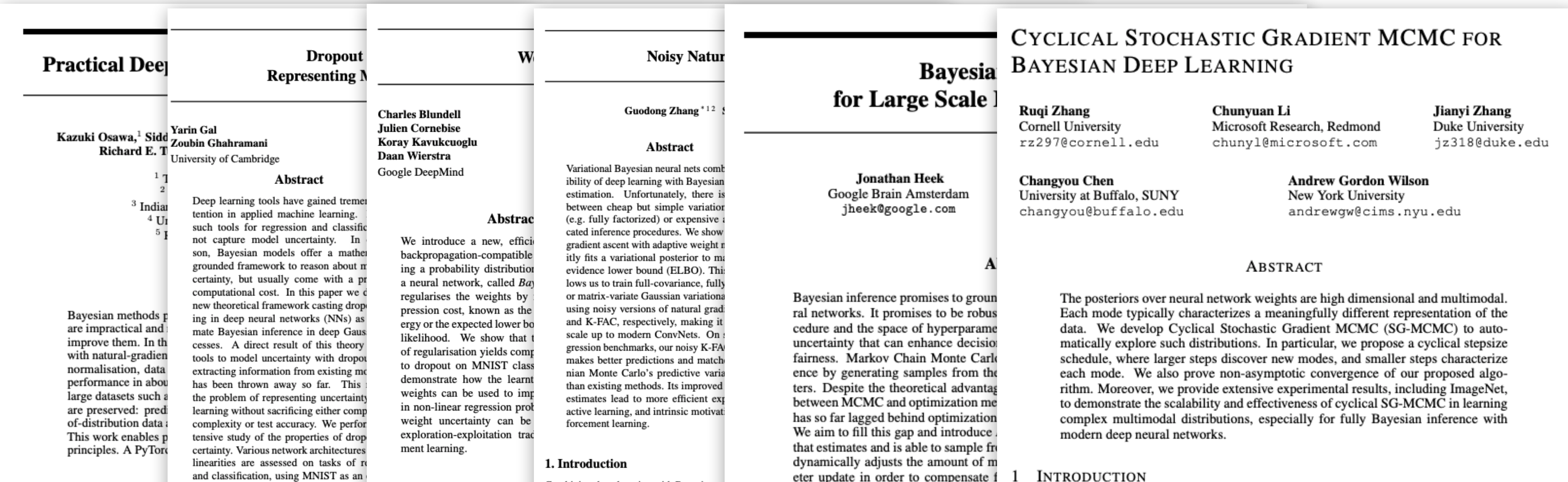
Google AI

Bayesian Deep Learning

Google AI
Brain Team

Goal: enable **Bayesian inference for deep networks** to improve robustness of predictions!

Active research field where most work focuses on **improving approximate inference** to get closer to the Bayes posterior



But is the Bayes posterior actually good?

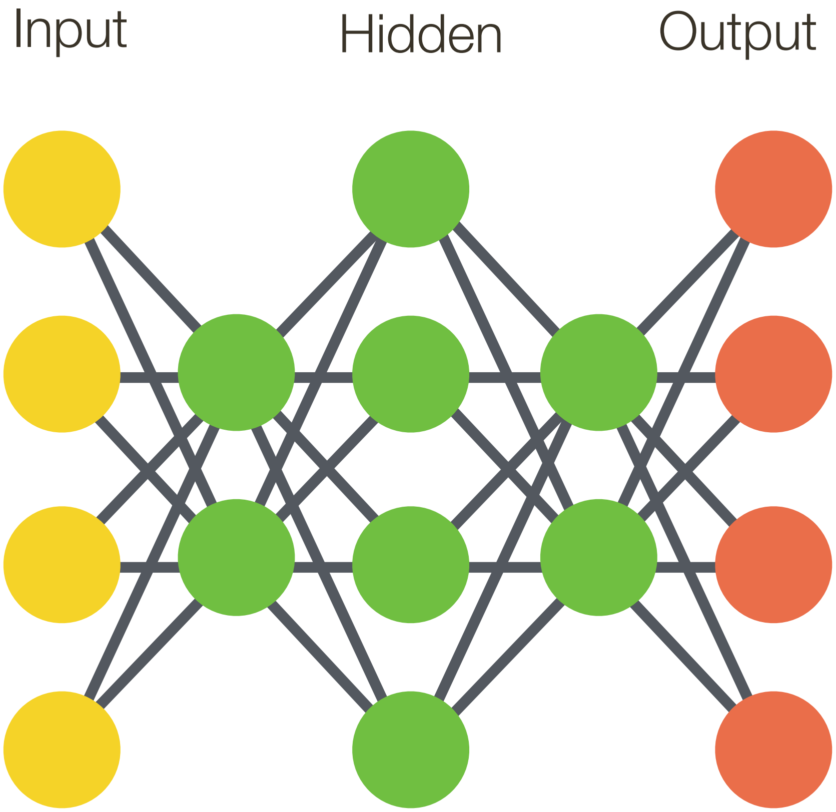
Bayesian Neural Networks (BNNs)

Google AI Brain Team

Neural Network

$$p(\mathcal{D}|\boldsymbol{\theta}) = p(y_i|x_i, \boldsymbol{\theta})$$

Different models obtained by different $\boldsymbol{\theta}$



Bayesian Neural Networks (BNNs)

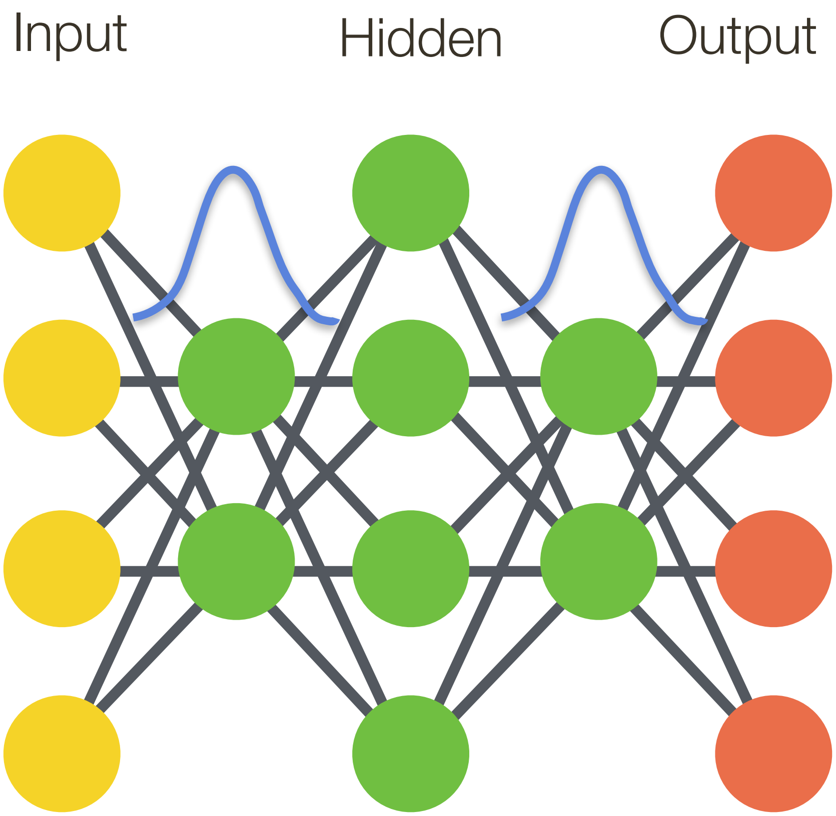
Google AI
Brain Team

Bayesian Neural Network

$$p(\boldsymbol{\theta}, \mathcal{D}) = p(y_i | x_i, \boldsymbol{\theta})\ p(\boldsymbol{\theta})$$

Posterior: Distribution over likely models given the data

$$p(\boldsymbol{\theta} | \mathcal{D})$$

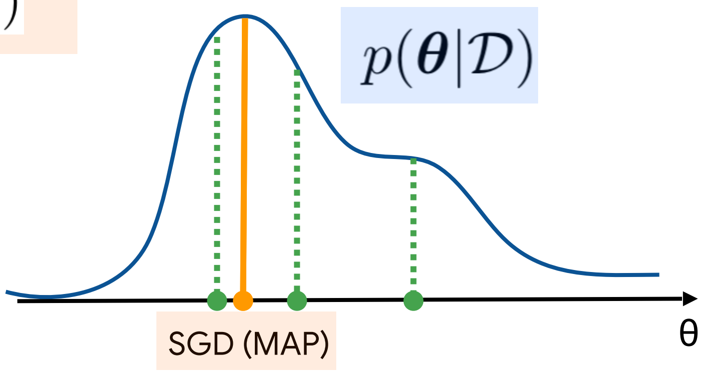


BNNs: Predictions

Google AI
Brain Team

In standard deep learning we optimize $U(\boldsymbol{\theta})$

$$U(\boldsymbol{\theta}) := -\sum_{i=1}^{n} \log p(y_i|x_i, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta})$$



BNNs use samples from the posterior (ensemble of models)

$$\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \ldots \sim p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-U(\boldsymbol{\theta}))$$

BNNs: Predictions

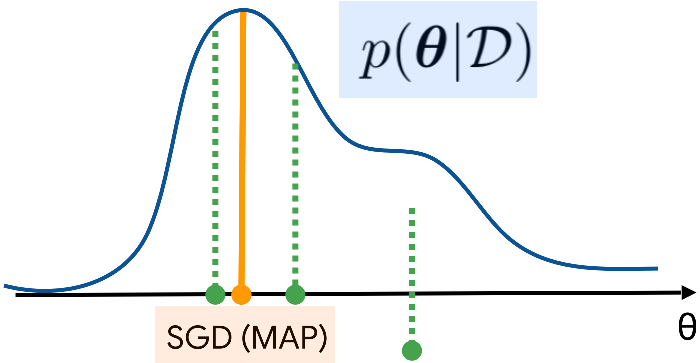
Google AI
Brain Team

Predict by using an average of models

$$p(y|x,\mathcal{D}) = \int p(y|x,\boldsymbol{\theta})\, p(\boldsymbol{\theta}|\mathcal{D})\, \mathrm{d}\boldsymbol{\theta}$$

$$\approx \sum_s p(y|x,\boldsymbol{\theta}_s)$$

$$\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \ldots \sim p(\boldsymbol{\theta}|\mathcal{D})$$



In this talk: A model is good if it predicts well (e.g. low cross entropy loss)

Bayesian Neural Networks (BNNs)

Google AI
Brain Team

Promises of BNNs*:

- **Robustness** in generalization
- Better **uncertainty quantification** *(calibration)*
- Enables new deep learning **applications** *(continual learning, sequential decision making, …)*

* [e.g., Neal 1995, Gal et al. 2016, Wilson 2019, Ovadia et al. 2019].

Bayesian Neural Networks (BNNs)

Google AI
Brain Team

But in practice BNNs are rarely used!

Bayesian Neural Networks (BNNs)

Google AI
Brain Team

In practice:

- Often, the **Bayes posterior is worse than SGD** point estimates
- But Bayes predictions can be **improved** by the use of the

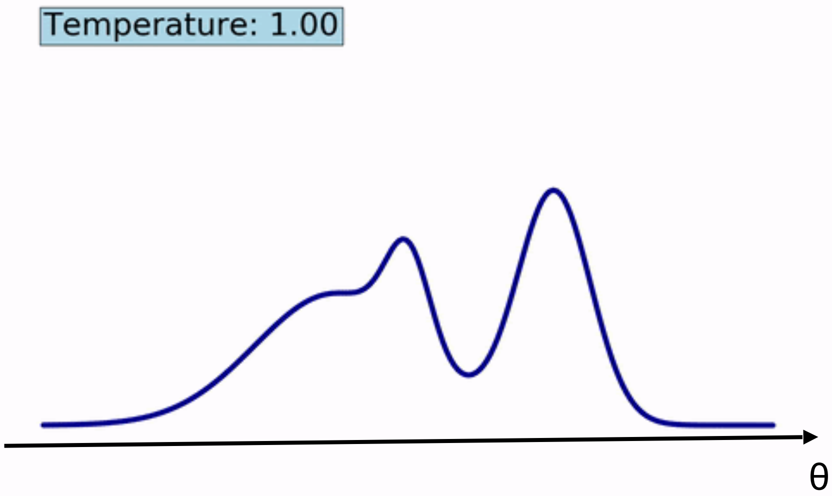
Cold Posterior*

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-U(\boldsymbol{\theta})/T)$$

For temperature *T<1:*
We **sharpen the posterior** (over-count evidence)

*Explicitly (or implicitly) ***used by most recent Bayesian DL papers*** [e.g., Li et al. 2016, Zhang et al. 2020, Ashukha et al. 2020].

Bayesian Neural Networks (BNNs)

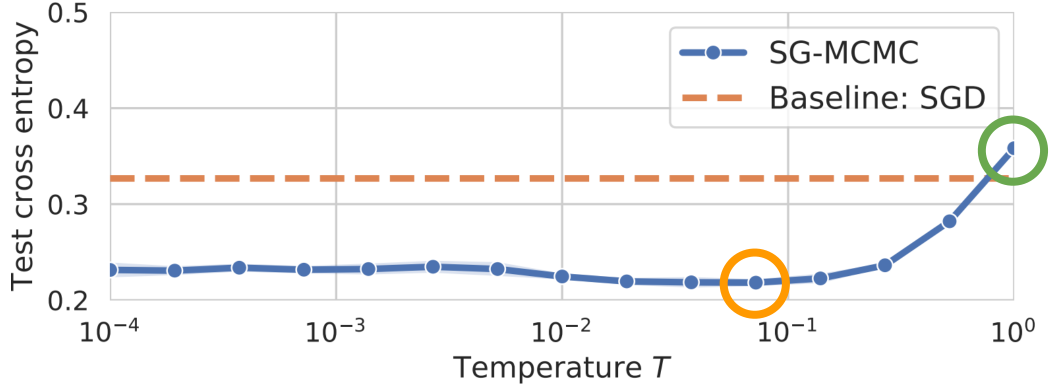


Cold Posterior

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-U(\boldsymbol{\theta})/T)$$

For temperature *T<1:* We **sharpen the posterior** (over-count evidence)

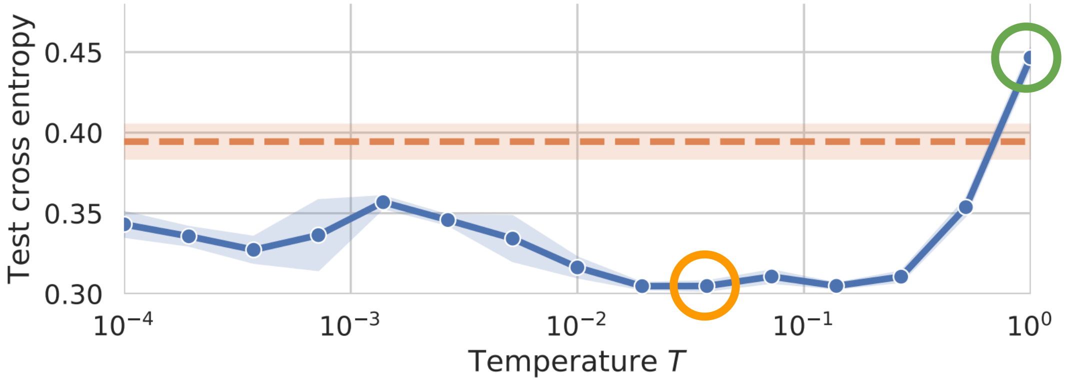
ResNet-20 / CIFAR-10



True Bayes posterior

Optimal cold posterior

CNN-LSTM / IMDB



The cold posterior sharply deviates from the Bayesian paradigm.

What is the use of more accurate posterior approximations if the posterior is poor?

Our paper: Hypothesis for the origin of the improved performance of cold posteriors

Google AI
Brain Team

Inference	Likelihood	Prior
Inaccurate SDE Simulation? Bias of SG-MCMC? Minibatch noise (which is not Gaussian)? Bias-variance tradeoff induced by cold posterior?	Dirty likelihoods? *(batch-normalization, dropout, data augmentation)*	Current priors used for BNN parameters are poor? The effect becomes stronger with increasing model depths and capacity?

Our paper: Hypothesis for the origin of the improved performance of cold posteriors

Google AI
Brain Team

✓ Inference	✓ Likelihood	✗ Prior
Inaccurate SDE Simulation?	Dirty likelihoods?	Current priors used for BNN parameters are poor?
Bias of SG-MCMC?	*(batch-normalization, dropout, data augmentation)*	The effect becomes stronger with increasing model depths and capacity?
Minibatch noise (which is not Gaussian)?		
Bias-variance tradeoff induced by cold posterior?		

Our paper: Hypothesis for the origin of the improved performance of cold posteriors

Google AI Brain Team

Inference	Likelihood	Prior
Inaccurate SDE Simulation?	Dirty likelihoods?	Current priors used for BNN parameters are poor?
Bias of SG-MCMC?	*(batch-normalization, dropout, data augmentation)*	The effect becomes stronger with increasing model depths and capacity?
Minibatch noise (which is not Gaussian)?		
Bias-variance tradeoff induced by cold posterior?		

Inference: Is it accurate?

Google AI
Brain Team

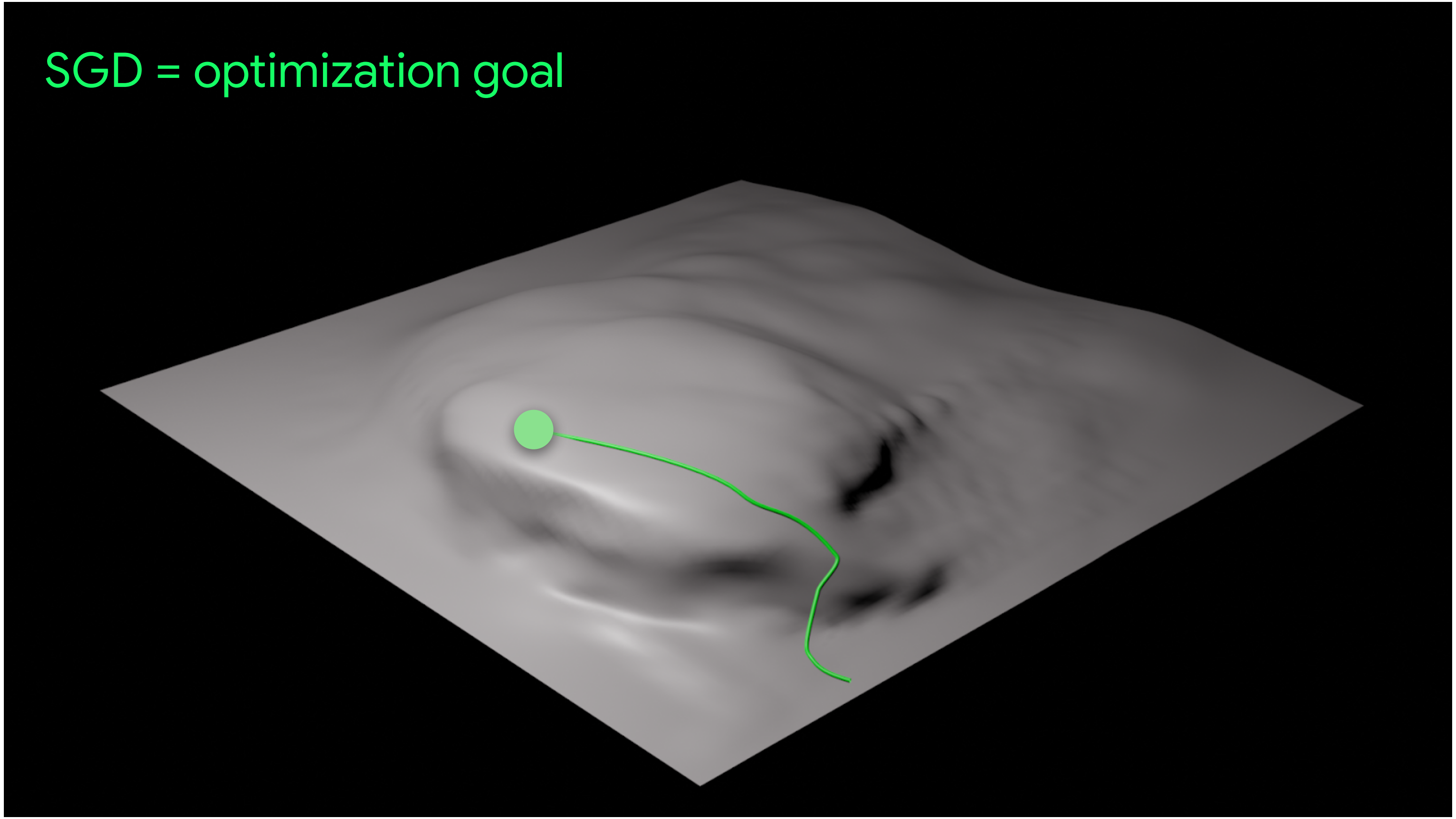
1. How to compute the posterior (inference)?

   Sample from the posterior using SG-MCMC methods

   Not covered: Approximate posterior using variational inference

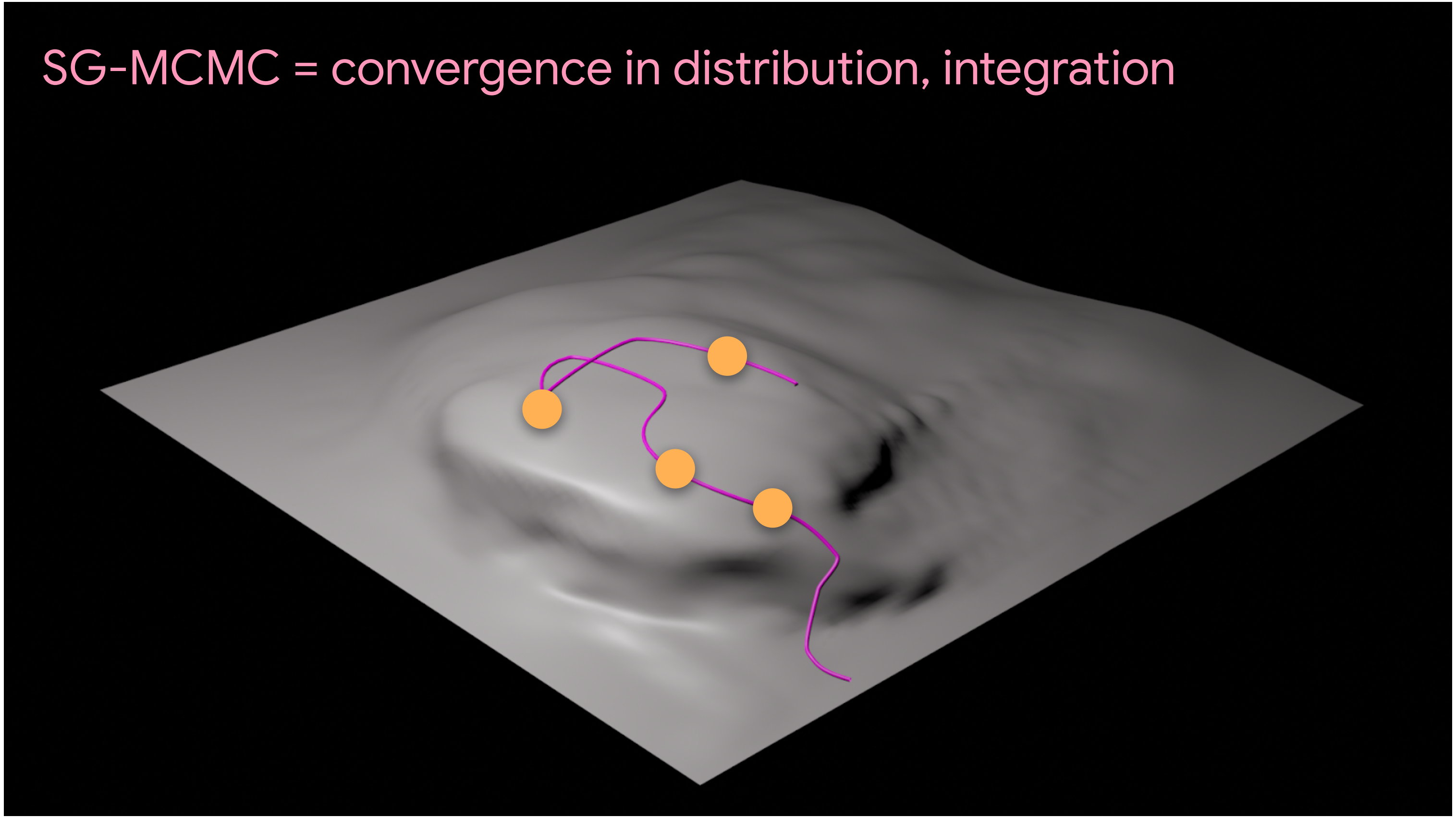
2. Does inaccurate inference lead to the cold posterior effect?

SG-MCMC: Stochastic Gradient Markov Chain Monte Carlo



SG-MCMC: Stochastic Gradient Markov Chain Monte Carlo

Google AI
Brain Team



Stochastic Gradient Markov Chain Monte Carlo

Google AI Brain Team

Langevin Dynamics: one-slide refresher

$$\mathrm{d}\boldsymbol{\theta} = \mathbf{M}^{-1}\mathbf{m}\mathrm{d}t$$

$$\mathrm{d}\mathbf{m} = -\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta})\mathrm{d}t - \gamma\mathbf{m}\mathrm{d}t + \sqrt{2\gamma T}\mathbf{M}^{1/2}\mathrm{d}\mathbf{W}$$

- Simulating SDE has **stationary distribution** proportional to exp(-U(θ) / T)
  [Langevin, 1908], [Leimkuhler and Matthews, “Molecular Dynamics”, 2016]
- Parameters θ, moments m, mass matrix M > 0, friction γ > 0
- **“Solving SDE”** ⇔ obtain one random continuous-time path

Stochastic Gradient Markov Chain Monte Carlo

Google AI
Brain Team

Symplectic Euler (Discretized version of SDE)

$$\mathbf{m}^{(t)} = (1 - h\gamma)\mathbf{m}^{(t-1)} - hn\nabla_{\boldsymbol{\theta}}\tilde{G}\left(\boldsymbol{\theta}^{(t-1)}\right) + \sqrt{2\gamma h T}\mathbf{M}^{1/2}\mathcal{N}\left(0, I\right)$$

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)} + h\mathbf{M}^{-1}\mathbf{m}^{(t)}$$

SGD with Momentum

Gaussian Noise

scaled by temperature

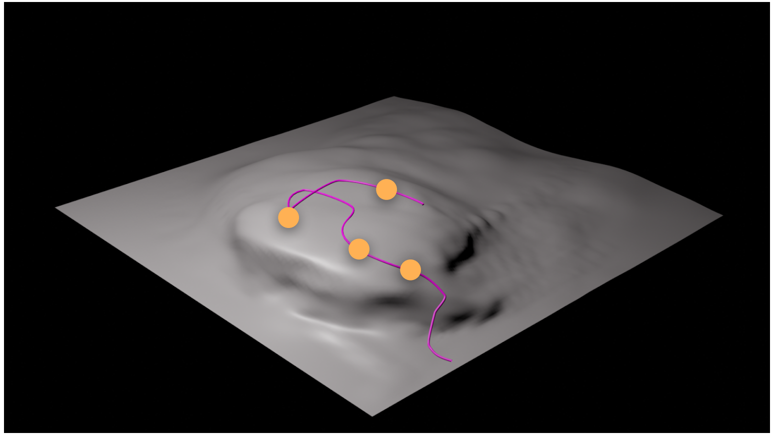
Stochastic Gradient Markov Chain Monte Carlo

Google AI Brain Team

The discretization scheme leads to

approximate samples from the posterior

$$\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \boldsymbol{\theta}^{(3)}, \dots$$

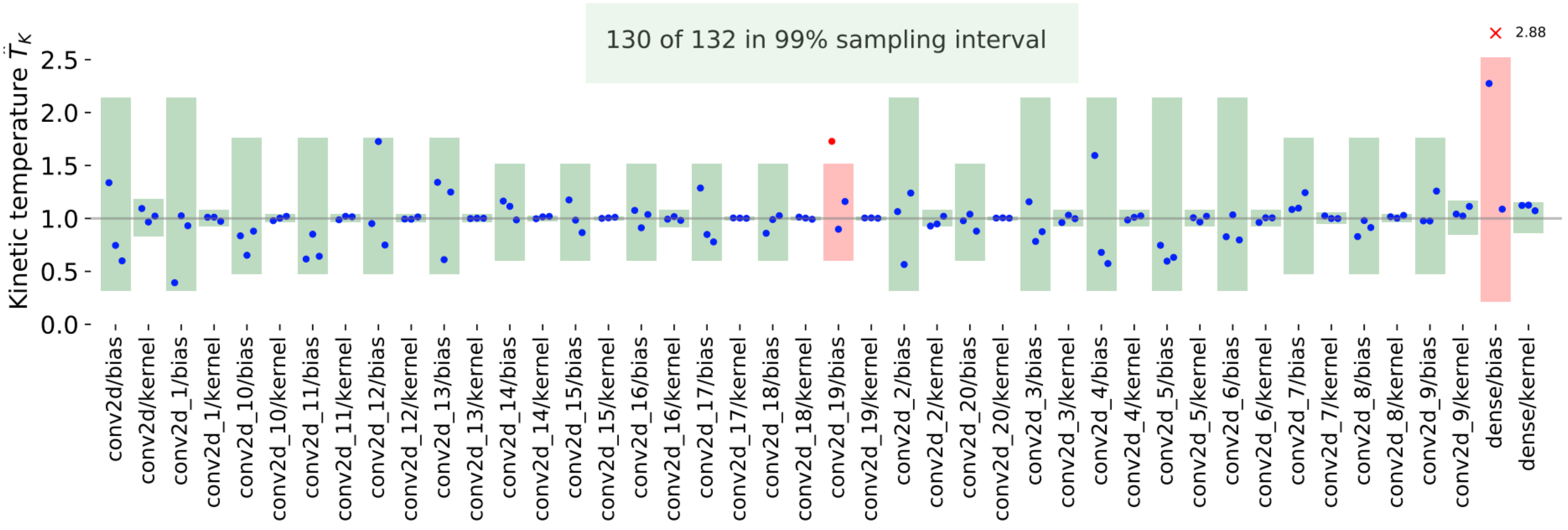


Is this accurate enough?

Novel diagnostics for SG-MCMC

Google AI Brain Team

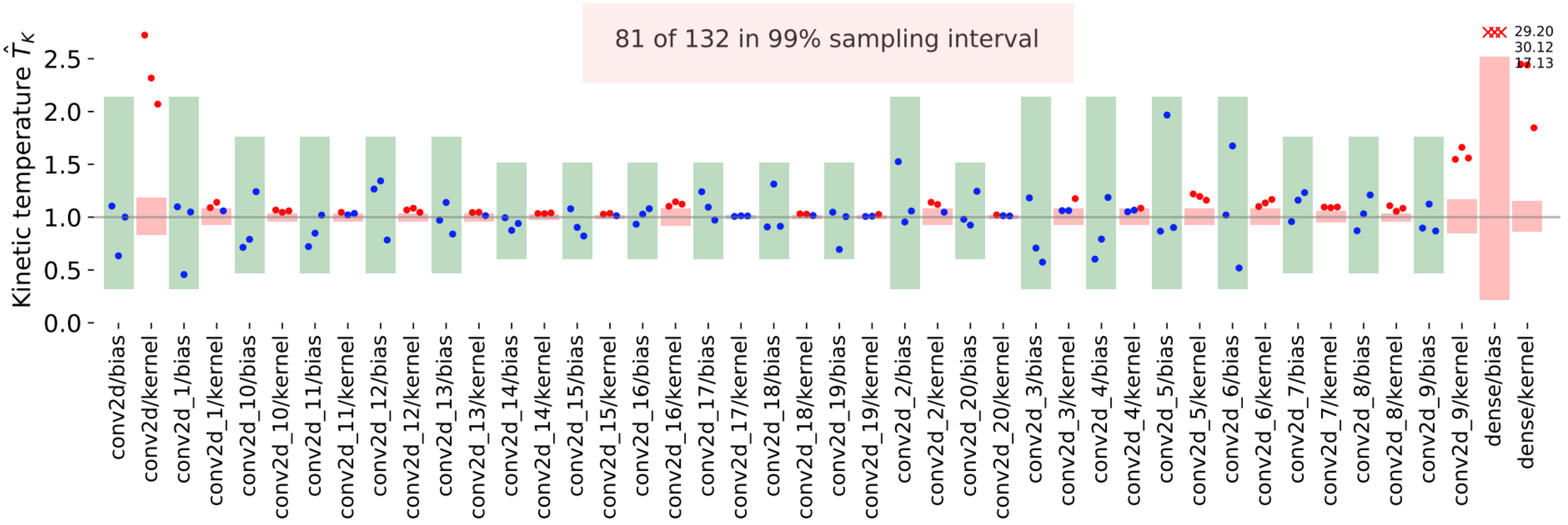
Diagnostics check out!



Novel diagnostics for SG-MCMC

Google AI
Brain Team

Diagnostics fail.

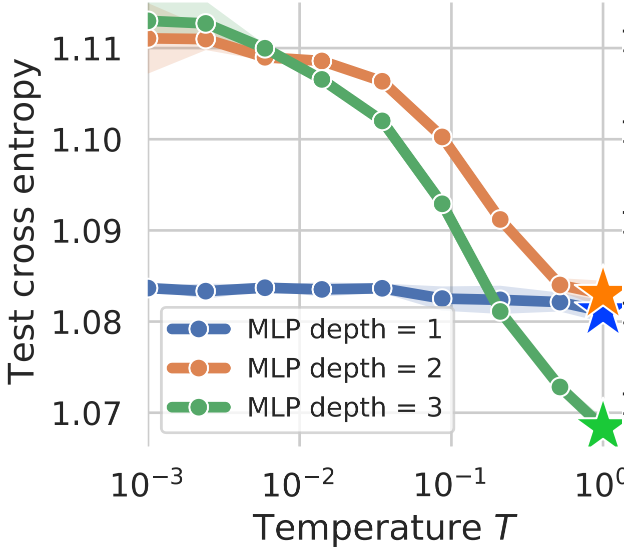


SG-MCMC works well enough!

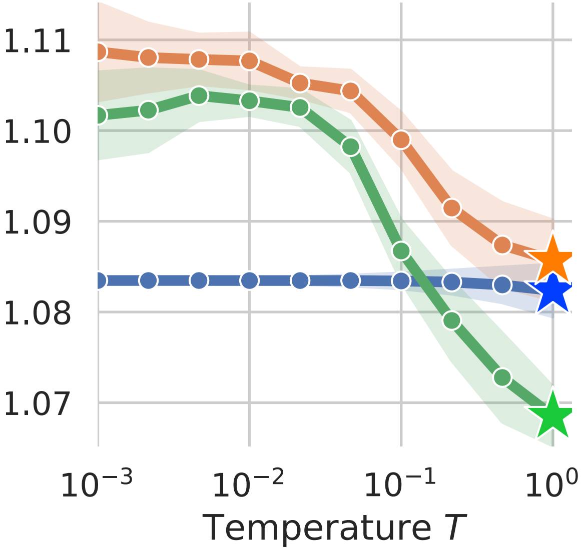
Google AI
Brain Team

Synthetic data generated from an MLP

Gold Standard (HMC)



SG-MCMC



SG-MCMC inference works **well enough!**

Google AI Brain Team

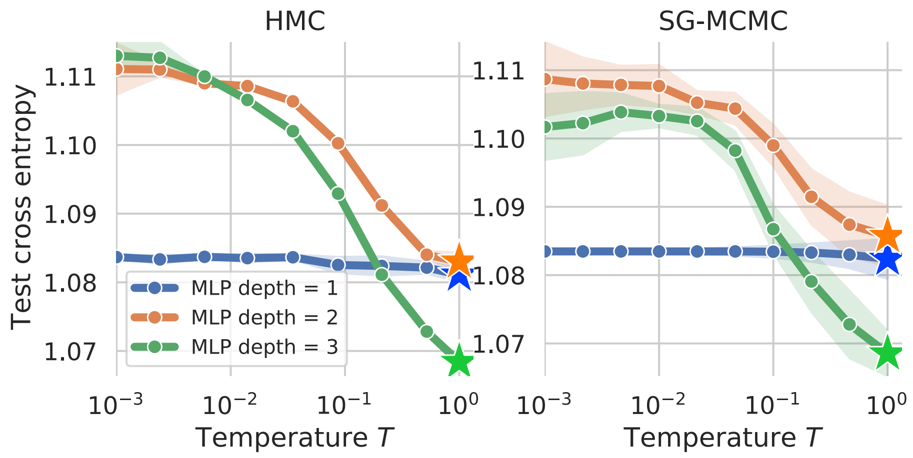
Inference

Inaccurate SDE Simulation? ✓

Bias of SG-MCMC? ✓

Minibatch noise (which is not Gaussian)? ✓

Bias-variance tradeoff induced by cold posterior? ✓



SG-MCMC inference works **well enough!**

Google AI
Brain Team

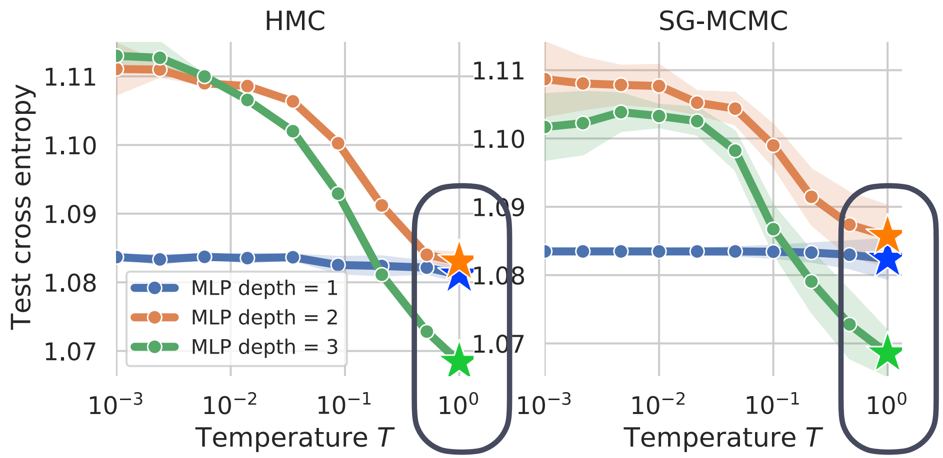
Inference

Inaccurate SDE Simulation? ✓

Bias of SG-MCMC? ✓

Minibatch noise (which is not Gaussian)? ✓

Bias-variance tradeoff induced by cold posterior? ✓



If the model is **well-specified**, T=1 is optimal.

But for real-world data T<1 is better!

The cold posterior effect

Why does the cold posterior perform better than the true Bayes posterior?

Problems with the **prior**?

Prior

Current priors used for BNN parameters are poor?

$$p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$$

The effect becomes stronger with increasing model depths and capacity?

Prior Predictive Experiment

Draw from prior

$$\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$$

Induced predictive distribution

$$\mathbb{E}_{x \sim p(x)} \left[p\left(y | x, \boldsymbol{\theta}^{(i)} \right) \right]$$

Prior Predictive Experiment

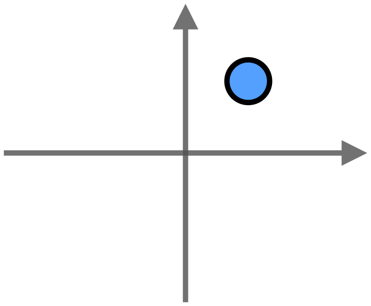
Draw from prior

$$\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$$

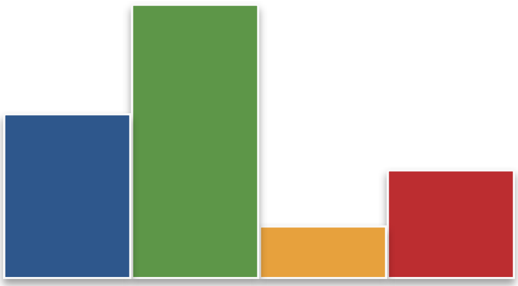
Induced predictive distribution

$$\mathbb{E}_{x \sim p(x)} \left[p\left(y \middle| x, \boldsymbol{\theta}^{(i)} \right) \right]$$

Model parameters



Class Probabilities



Google AI
Brain Team

Prior Predictive Experiment

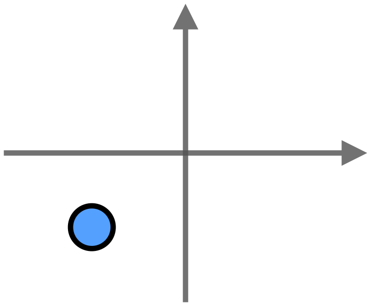
Draw from prior

$$\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$$

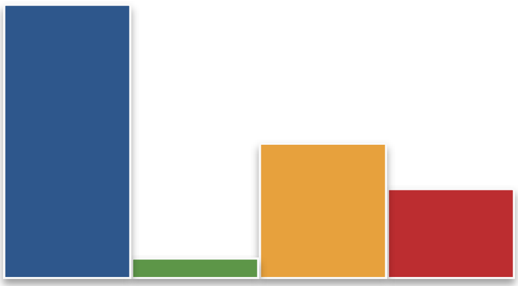
Induced predictive distribution

$$\mathbb{E}_{x \sim p(x)} \left[p \left(y | x, \boldsymbol{\theta}^{(i)} \right) \right]$$

Model parameters



Class Probabilities



Prior Predictive Experiment

Google AI
Brain Team

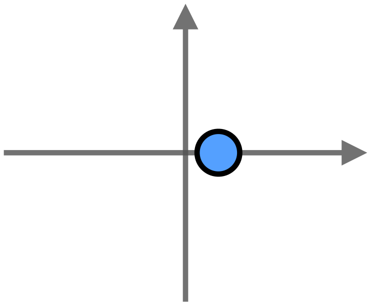
Draw from prior

$$\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$$

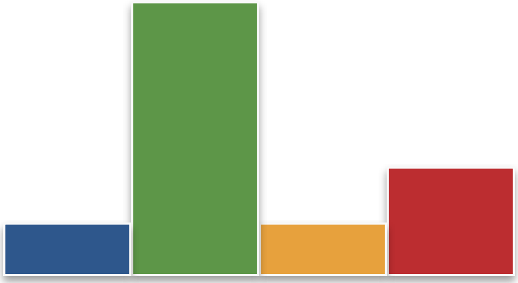
Induced predictive distribution

$$\mathbb{E}_{x \sim p(x)} \left[p \left(y | x, \boldsymbol{\theta}^{(i)} \right) \right]$$

Model parameters



Class Probabilities



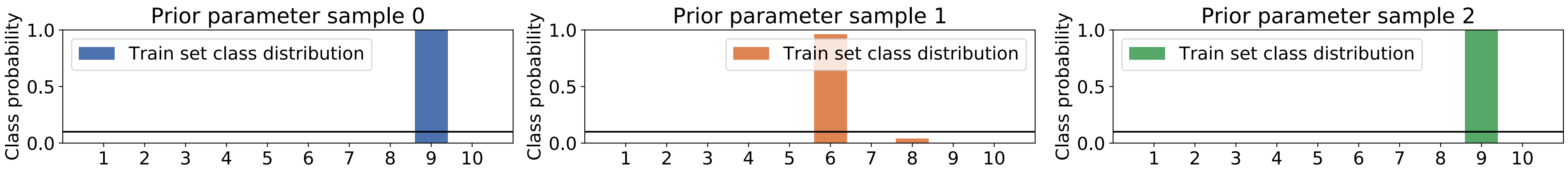
Prior Predictive Experiment: ResNet-20 / CIFAR-10

Google AI
Brain Team

$$\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}) = \mathcal{N}(0, I)$$

$$\mathbb{E}_{x \sim p(x)} \left[p\left(y \middle| x, \boldsymbol{\theta}^{(i)} \right) \right]$$

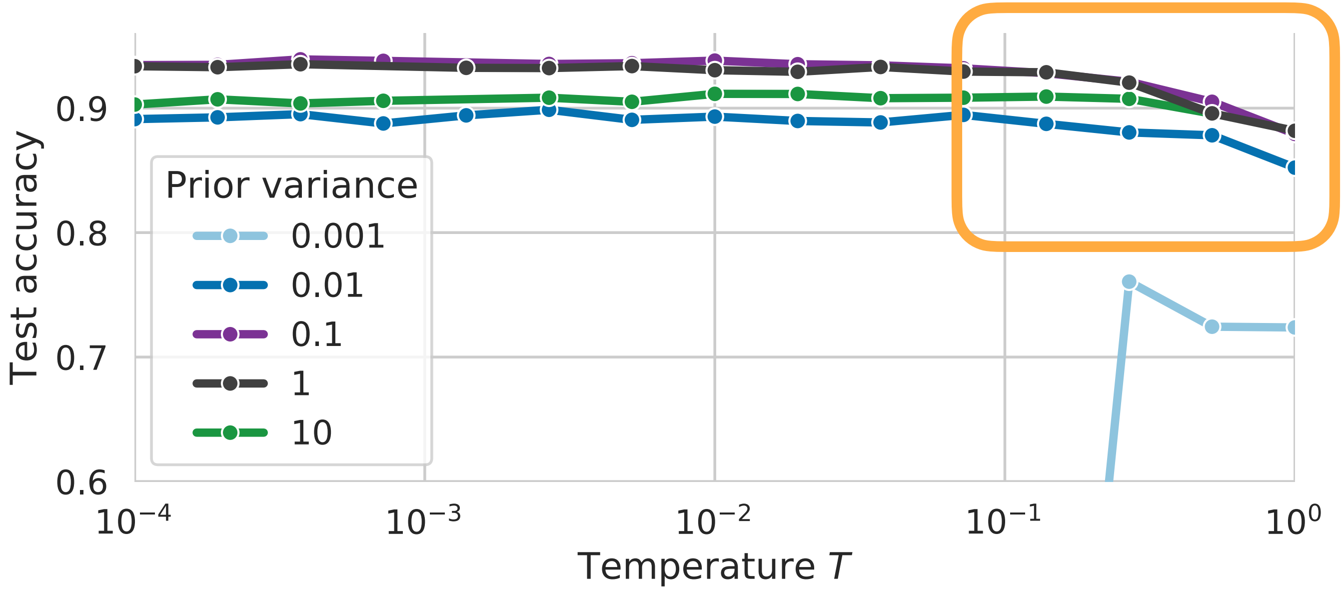
$\boldsymbol{\theta}^{(1)}$ $\boldsymbol{\theta}^{(2)}$ $\boldsymbol{\theta}^{(3)}$



Each network drawn from the prior maps all images to one class!

There is no “easy” fix of the prior

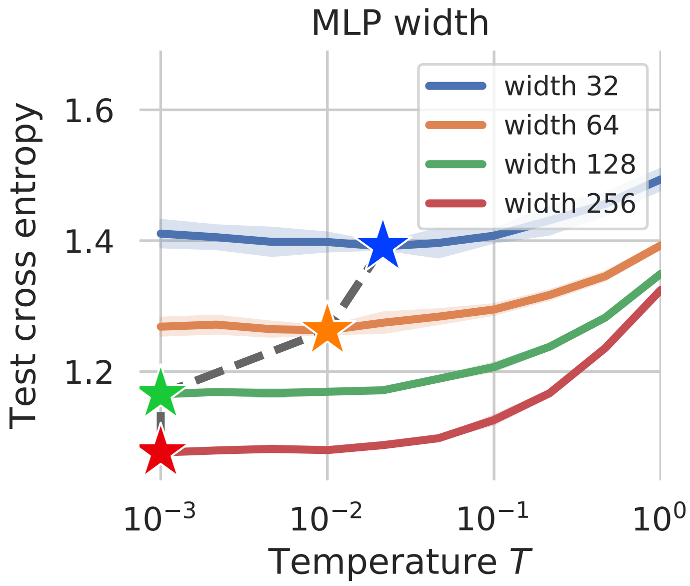
Final performance of different variances σ used for the prior $\theta \sim \mathcal{N}(0,\sigma)$



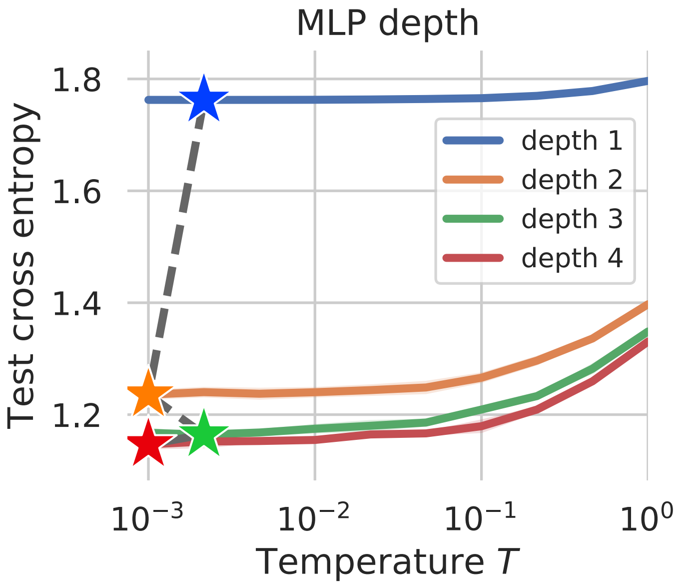
ResNet-20 / CIFAR-10

The cold posterior effect becomes stronger with increasing capacity

MLP / CIFAR-10



(fixed depth=3)



(fixed width=128)

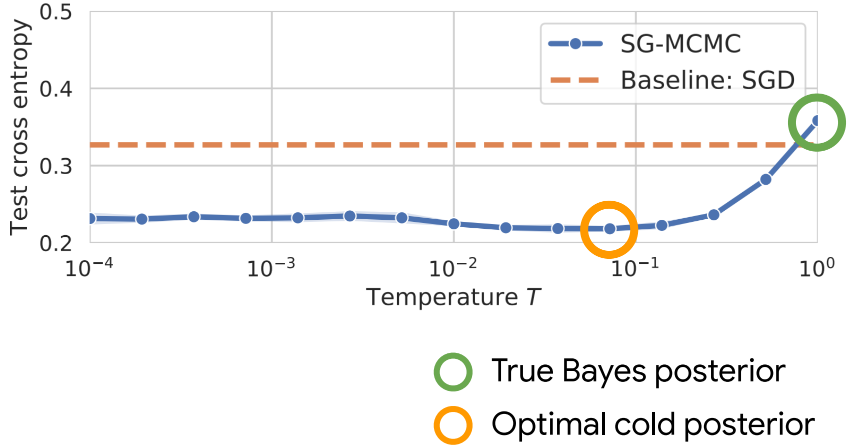
Summary

SG-MCMC is accurate enough.

Cold posteriors work.

More work on priors for deep nets is needed.

ResNet-20 / CIFAR-10

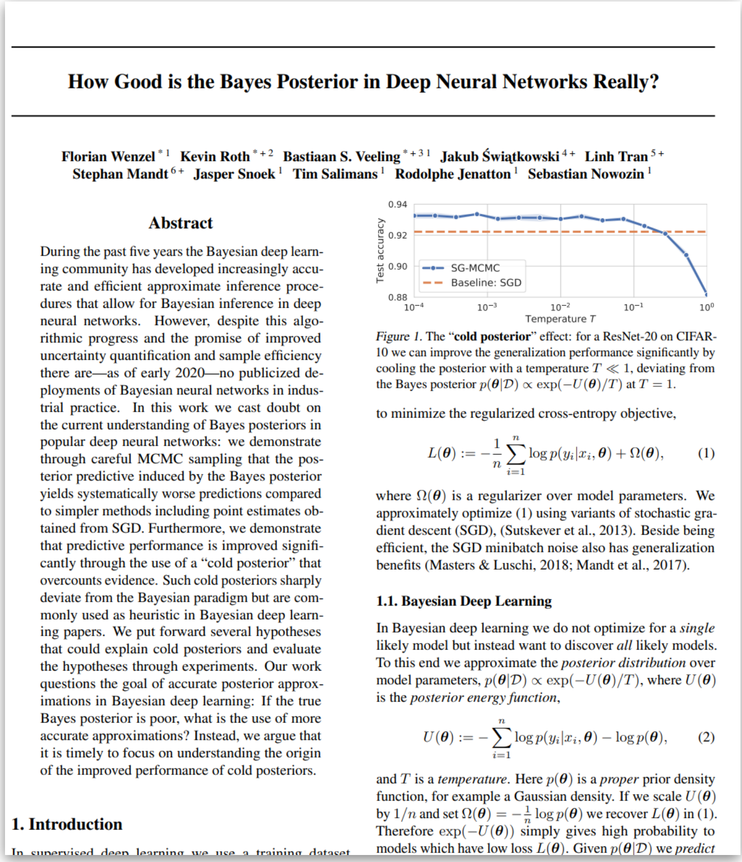


Code: github.com/google-research/google-research/tree/master/cold_posterior_bnn

More info/feedback:

www.florianwenzel.com

florianwenzel@google.com



How Good is the Bayes Posterior in Deep Neural Networks Really?

Florian Wenzel [*1] **Kevin Roth** [*+2] **Bastiaan S. Veeling** [*+3 1] **Jakub Świątkowski** [4+] **Linh Tran** [5+] **Stephan Mandt** [6+] **Jasper Snoek** [1] **Tim Salimans** [1] **Rodolphe Jenatton** [1] **Sebastian Nowozin** [1]

Abstract

During the past five years the Bayesian deep learning community has developed increasingly accurate and efficient approximate inference procedures that allow for Bayesian inference in deep neural networks. However, despite this algorithmic progress and the promise of improved uncertainty quantification and sample efficiency there are—as of early 2020—no publicized deployments of Bayesian neural networks in industrial practice. In this work we cast doubt on the current understanding of Bayes posteriors in popular deep neural networks: we demonstrate through careful MCMC sampling that the posterior predictive induced by the Bayes posterior yields systematically worse predictions compared to simpler methods including point estimates obtained from SGD. Furthermore, we demonstrate that predictive performance is improved significantly through the use of a "cold posterior" that overcounts evidence. Such cold posteriors sharply deviate from the Bayesian paradigm but are commonly used as heuristic in Bayesian deep learning papers. We put forward several hypotheses that could explain cold posteriors and evaluate the hypotheses through experiments. Our work questions the goal of accurate posterior approximations in Bayesian deep learning: If the true Bayes posterior is poor, what is the use of more accurate approximations? Instead, we argue that it is timely to focus on understanding the origin of the improved performance of cold posteriors.

1. Introduction

In supervised deep learning we use a training dataset

Figure 1. The "**cold posterior**" effect: for a ResNet-20 on CIFAR-10 we can improve the generalization performance significantly by cooling the posterior with a temperature $T \ll 1$, deviating from the Bayes posterior $p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-U(\boldsymbol{\theta})/T)$ at $T = 1$.

to minimize the regularized cross-entropy objective,

$$L(\boldsymbol{\theta}) := -\frac{1}{n}\sum_{i=1}^{n} \log p(y_i|x_i, \boldsymbol{\theta}) + \Omega(\boldsymbol{\theta}), \qquad (1)$$

where $\Omega(\boldsymbol{\theta})$ is a regularizer over model parameters. We approximately optimize (1) using variants of stochastic gradient descent (SGD), (Sutskever et al., 2013). Beside being efficient, the SGD minibatch noise also has generalization benefits (Masters & Luschi, 2018; Mandt et al., 2017).

1.1. Bayesian Deep Learning

In Bayesian deep learning we do not optimize for a *single* likely model but instead want to discover *all* likely models. To this end we approximate the *posterior distribution* over model parameters, $p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-U(\boldsymbol{\theta})/T)$, where $U(\boldsymbol{\theta})$ is the *posterior energy function*,

$$U(\boldsymbol{\theta}) := -\sum_{i=1}^{n} \log p(y_i|x_i, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}), \qquad (2)$$

and T is a *temperature*. Here $p(\boldsymbol{\theta})$ is a *proper* prior density function, for example a Gaussian density. If we scale $U(\boldsymbol{\theta})$ by $1/n$ and set $\Omega(\boldsymbol{\theta}) = -\frac{1}{n}\log p(\boldsymbol{\theta})$ we recover $L(\boldsymbol{\theta})$ in (1). Therefore $\exp(-U(\boldsymbol{\theta}))$ simply gives high probability to models which have low loss $L(\boldsymbol{\theta})$. Given $p(\boldsymbol{\theta}|\mathcal{D})$ we *predict*