Recent Developments in Bayesian Deep Learning

Siddharth Swaroop, Andrew Foong

CBL Reading Group 22nd April, 2020

UNIVERSITY OF CAMBRIDGE

Outline

What is Bayesian Deep Learning?

Disclaimer!

- Recent Developments in Bayesian Deep Learning (BDL)
- Devil's advocates: the 'pragmatist' and the 'purist'
- Where are you on this scale?

Second half is a discussion!

Brief Recap

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}$$

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

- $p(\mathcal{D})$ intractable
  - Approximate inference techniques
  - Lots of sampling...

Pragmatist view of BDL

- Neural networks are great!
- But has failures: 'Uncertainty'
  - Overfits in small data setting
  - Online/sequential learning
  - Adversarial examples
  - 'Knowing when it doesn't know'
- Let's be Bayesian!

It doesn't matter how good our model, prior, or inference is: we just want 'good' predictions.

"The key distinguishing property of a Bayesian approach is marginalization instead of optimization, not the prior, or Bayes rule." [Wilson, 2020]

Purist view of BDL

Bayesian iff goals are to:

- Have prior we believe in.
- Perform accurate inference.

May not be achievable, but will influence methods/ metrics.
Should have:

- Better priors $\rightarrow$ better performance.
- Better inference $\rightarrow$ better performance.

If not, reconsider model.

"bad model + bad inference = good predictions" is NOT allowed.
Questions:

1. Where are you on the scale of pragmatist to purist?
2. Is this a helpful distinction? Have we missed some dimensions?

Some Current Methods

Stochastic Gradient MCMC

Markov Chain Monte Carlo generates samples by simulating Markov chain: $\theta_1, \theta_2, \ldots, \theta_t$.
Advantage:

- Obtain samples from true posterior as $t \to \infty$.

Disadvantages:

- Hard to know when chain has converged.
- Metropolis-Hastings and Hamiltonian Monte Carlo $\mathcal{O}(N)$.

N is huge in deep learning - use minibatches.

- SGMCMC regarded SOTA for BDL - Zhang et al. [2019], Wenzel et al. [2020].
- Justification involves stochastic differential equations (SDEs).

MCMC - Brief Recap

$\pi(\theta)$ - target distribution.
$T(\theta'; \theta)$ - transition probability of going to state θ' from θ
$p^{(t)}(\theta)$ - density after t iterations.
$p^{(t)}(\theta) \to \pi(\theta)$ if

- T leaves π invariant:

$$\pi(\theta') = \int T(\theta'; \theta)\pi(\theta)\,\mathrm{d}\theta.$$

- Chain is ergodic (irreducible, aperiodic).

Examples of T:

- Gibbs: sample some elements of θ conditioned on others.
- Metropolis-Hastings: propose new θ' with density $Q(\theta'; \theta_t)$ and accept with probability

$$a = \frac{\pi(\theta')Q(\theta_t; \theta')}{\pi(\theta_t)Q(\theta'; \theta_t)}.$$

Hamiltonian Monte Carlo - Brief Recap

HMC main ideas:

- Augment space with momentum r: (θ, r).
- Define canonical distribution $\pi(\theta, r) \propto \exp(-H)$, where $H = K(r) + U(\theta)$.
  - Kinetic energy $K = r^T M^{-1} r/2$.
  - Potential energy $U = -\log \pi(\theta) + \text{const}$.
- Simulate Hamiltonian dynamics as MH proposal.
- Dynamics reversible, conserve Hamiltonian, and preserve volume: acceptance probability $a = 1$.
- Resample r as Gibbs update (easy since independent Gaussian).
- Discard r to obtain samples from $\pi(\theta)$.

Hamiltonian Monte Carlo cont'd

Hamilton's equations:

$$\mathrm{d}\theta = M^{-1} r \, \mathrm{d}t$$
$$\mathrm{d}r = -\nabla U \, \mathrm{d}t$$

Can't simulate exactly: discretisation $\rightarrow$ Hamiltonian approximately conserved, $a < 1$.

- Advantage - momentum 'push' allows quick exploration.
- Disadvantage - ∇U and MH step are $\mathcal{O}(N)$.

Stochastic Gradients

Gradient of U:

$$\begin{aligned}\nabla U &= \nabla \left(\sum_{n=1}^{N} p(y_n|x_n, \theta) + p(\theta) \right) \\ &\approx \nabla \left(\frac{N}{M} \sum_{m=1}^{M} p(y_m|x_m, \theta) + p(\theta) \right) \\ &:= \nabla \tilde{U}\end{aligned}$$

Substitute $\nabla \tilde{U}$ for ∇U to get $\mathcal{O}(M)$ algorithm?

- Model SG noise as Gaussian (CLT for large M, note abuse of notation):

$$\nabla \tilde{U} \approx \nabla U + \mathcal{N}(0, V(\theta)).$$

- Turns out need to add extra 'friction' term.

Stochastic Differential Equations (Intuition)

SDEs analyse differential equations perturbed by Brownian motion.
Brownian motion $\{B_t\}_{t\geq 0}$:

- $B_{t_2} - B_{t_1} \sim \mathcal{N}(0, t_2 - t_1)$.
- Independent increments.

Typical SDE:

$$\mathrm{d}X_t = \mu(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}B_t$$

is shorthand for:

$$X_t - X_0 = \int_0^t \mu(X_s)\,\mathrm{d}s + \int_0^t \sigma(X_s)\,\mathrm{d}B_s$$

where LHS & RHS both random variables.

Stochastic Differential Equations (cont'd)

Last integral is an Itô integral, roughly:

$$\int_0^t \sigma(X_s)\,\mathrm{d}B_s = \lim_{\Delta t \to 0} \sum_k \sigma(X_{t_k})\left(B_{t_{k+1}} - B_{t_k}\right)$$

with $\Delta t = t_{k+1} - t_k$.

Simple way to approximate solution - Euler-Maruyama method.

Given SDE $\mathrm{d}X_t = \mu(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}B_t$:

1. Choose discretisation Δt.
2. Sample $\Delta B_k \sim \mathcal{N}(0, \Delta t)$.
3. $X_{t_{k+1}} \leftarrow \mu(X_{t_k})\Delta t + \sigma(X_{t_k})\Delta B_k$.

Note ΔB_k is $\mathcal{O}(\sqrt{\Delta t})$, not $\mathcal{O}(\Delta t)$!

Fokker-Planck Equation and Naive SGHMC

Let $p(x,t)$ be density of X_t. Fokker-Planck Equation:

$$\frac{\partial}{\partial t}p(x,t) = -\frac{\partial}{\partial x}[\mu(x)p(x,t)] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[\sigma(x)^2 p(x,t)].$$

Stationary distribution has $\frac{\partial}{\partial t}p(x,t) = 0$.
Naive SGHMC algorithm with step-size $\Delta t = \epsilon$:

$$\begin{aligned}\Delta r &= -\nabla \tilde{U}\epsilon \\ &= -\nabla U\epsilon + \epsilon \cdot \mathcal{N}(0, V) \\ &= -\nabla U\epsilon + \sqrt{\epsilon V} \cdot \mathcal{N}(0, \epsilon I),\end{aligned}$$

which can be viewed as a discretisation of the SDE:

$$\mathrm{d}r = -\nabla U \,\mathrm{d}t + \sqrt{\epsilon V}\,\mathrm{d}B_t$$

Chen et al. [2014] show π no longer stationary using FPE.

Stochastic Gradient HMC

Chen et al. [2014] propose adding a friction term C:

$$\mathrm{d}\theta = M^{-1} r \, \mathrm{d}t$$

$$\mathrm{d}r = -\nabla U \, \mathrm{d}t - \underbrace{C M^{-1} \theta \, \mathrm{d}t}_{\text{added friction}} + \underbrace{\sqrt{2C - \epsilon \hat{V}} \, \mathrm{d}B_t}_{\text{added noise}} + \underbrace{\sqrt{\epsilon V} \, \mathrm{d}B_t}_{\text{SG noise}},$$

where $\hat{V}$ is an estimate of the gradient covariance V.

- If $\hat{V} = V$, Chen et al. [2014] show via Fokker-Planck that this SDE has π as the stationary distribution.
- If $\hat{V} = 0$, note SG noise is smaller than added noise by $\sqrt{\epsilon}$ - hopefully 'washes out' as $\epsilon \to 0$.

What about MH accept-reject?

- $\mathcal{O}(N)$ to compute.
- Ignore and hope for the best!

Practical Choices of Parameters

SGHMC SDE:

$$\mathrm{d}\theta = M^{-1}\theta\,\mathrm{d}t$$

$$\mathrm{d}r = -\nabla U\,\mathrm{d}t - \underbrace{CM^{-1}\theta\,\mathrm{d}t}_{\text{added friction}} + \underbrace{\sqrt{2C - \epsilon\hat{V}}\,\mathrm{d}B_t}_{\text{added noise}} + \underbrace{\sqrt{\epsilon V}\,\mathrm{d}B_t}_{\text{SG noise}},$$

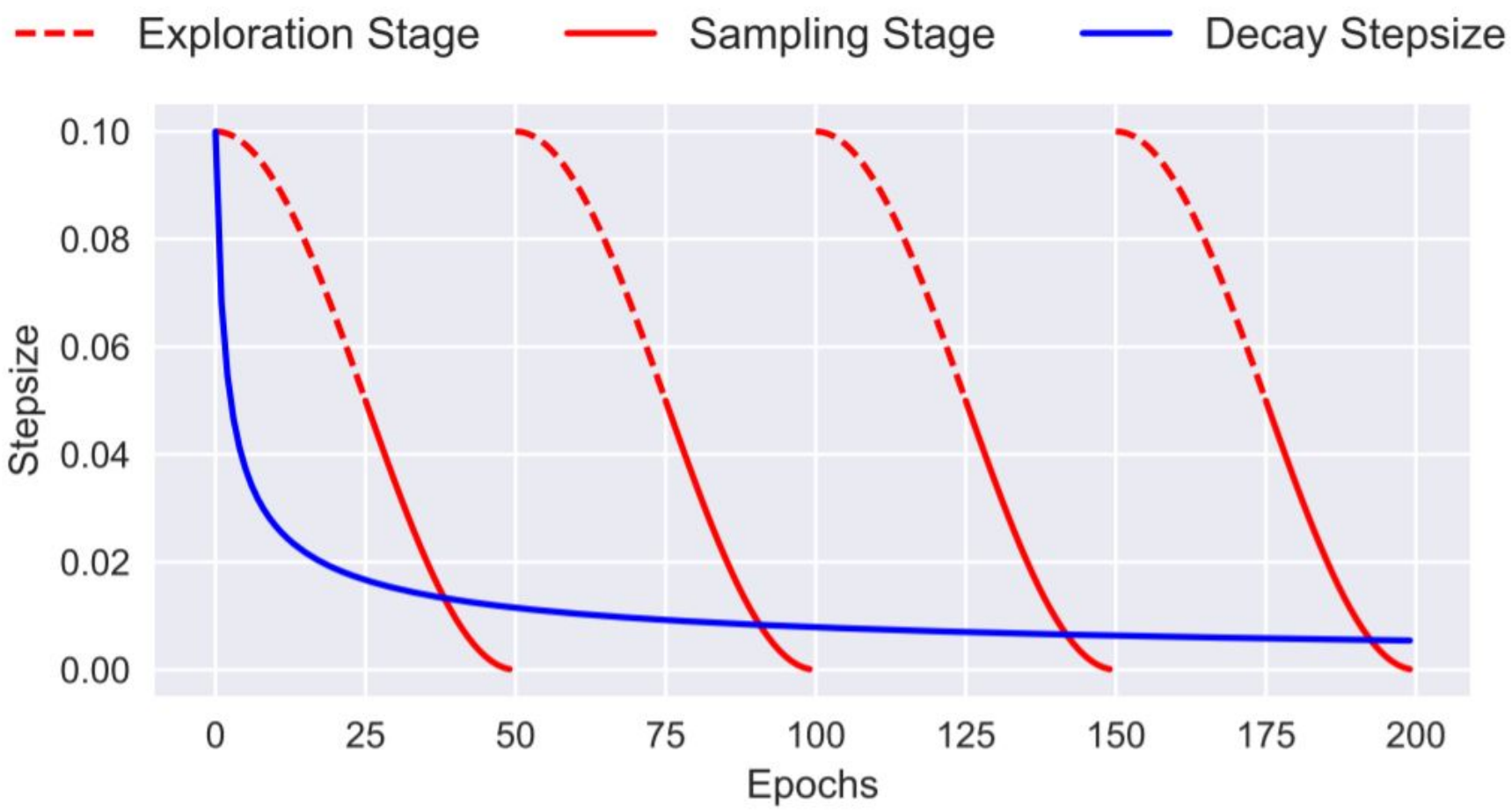
In Wenzel et al. [2020], Zhang et al. [2019], Springenberg et al. [2016] these are chosen:

1. Mass matrix M: identity, diagonal, 'layerwise diagonal':
   - Diagonal elements estimated using RMSprop as preconditioner.
   - Compute running average of squared gradients during burn-in, use these to normalise.
2. Friction matrix C: diagonal, set so $CM^{-1} = \gamma I$. Note γ related to 'momentum'.
3. SG noise estimate $\hat{V}$: zero matrix, or diagonal with elements estimated using RMSprop.

Cyclical SGMCMC

Zhang et al. [2019] propose cyclical learning rates to explore multiple modes:

- High learning rate to escape local mode (no sampling).
- Low learning rate to sample around local mode.



Cyclical SGMCMC Results

	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	4.29±0.06	20.55±0.06
SGHMC	4.93±0.1	22.60±0.17
cSGHMC	**4.27**±0.03	**20.50**±0.11

Table 1: Comparison of test error (%) between cSG-MCMC with non-parallel algorithms. cSGLD and cSGHMC yields lower errors than their optimization counterparts, respectively.

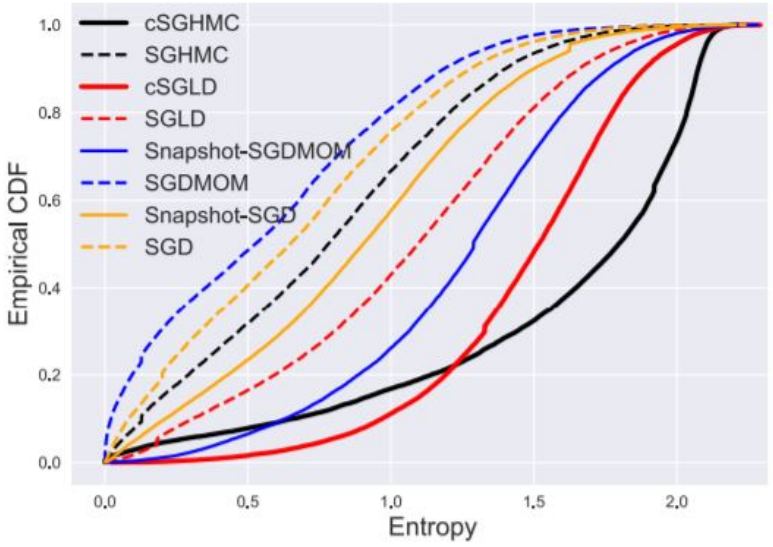


Figure 4: Empirical CDF for the entropy of the predictive distribution on notMNIST dataset. cSGLD and cSGHMC show lower probability for the low entropy estimate than other algorithms.

Note cold posterior effect Wenzel et al. [2020] - have to multiply $U(\theta)$ by $10 - 100$ (equivalently reduce added noise by that factor) to obtain good results!

Comparison between HMC and SGHMC

Differences:

- HMC is $\mathcal{O}(N)$ per iteration vs $\mathcal{O}(M)$ for SGHMC.
- HMC is always unbiased as $t \to \infty$, SGHMC only unbiased as $\epsilon \to 0$ and $t \to \infty$.
- SGHMC involves user-injected noise and friction.
- HMC justified by proving T leaves π invariant, SGHMC justified via SDE and the Fokker-Planck equation.
- SGHMC assumes minibatch noise is Gaussian (or 'washes out').
- Often momentum isn't resampled in SGHMC.
- HMC does MH correction, SGHMC doesn't.

Similarities:

- Neither make assumptions about form of posterior.
- Hard to assess convergence!

Variational Inference (VI)

Evidence Lower Bound:

$$\mathcal{L}(q_\phi(\theta)) = \mathbb{E}_{q_\phi}[\log p(\mathcal{D}|\theta)] - \mathcal{KL}[q_\phi(\theta)||p(\theta)]$$

- Mean-field Gaussians: $q_\phi(\theta) = \mathcal{N}(\theta|\mu, \mathrm{diag}(\sigma))$
- Bayes By Backprop [Blundell et al., 2015]: reparameterisation trick
- Efforts to reduce gradient variance, e.g. local reparameterisation trick [Kingma et al., 2015]

Natural-gradient VI

Variational Online Gauss-Newton (VOGN) [Osawa et al., 2019]

$$\mu_{t+1} \leftarrow \mu_t - \alpha_t \frac{\hat{g}(\theta_t) + \tilde{\delta}\mu_t}{s_{t+1} + \tilde{\delta}},$$

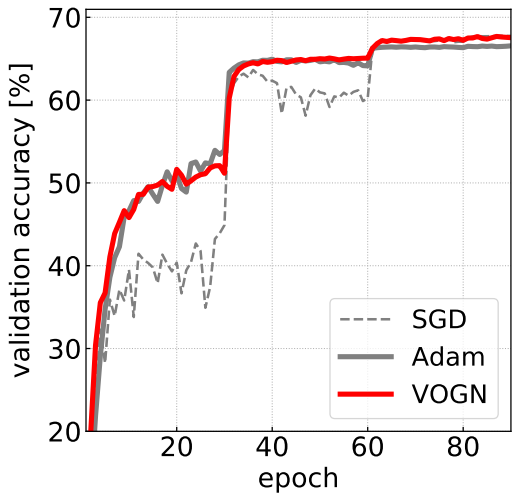
$$s_{t+1} \leftarrow (1 - \tau\beta_t)s_t + \beta_t \frac{1}{M} \sum_{i \in \mathcal{M}_t} \left(g_i(\theta_t)\right)^2,$$

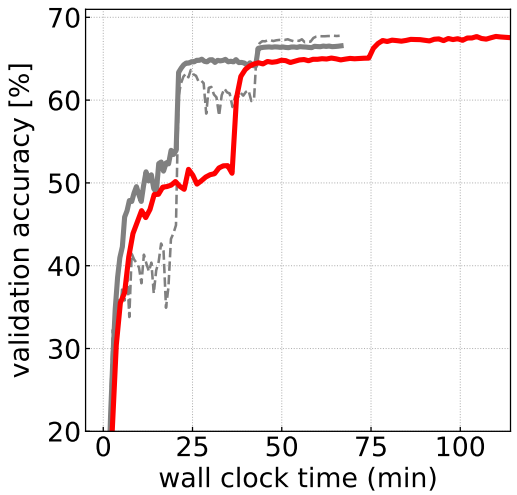
- $g_i(\theta_t) := \nabla_\theta \log p(y_i|x_i, \theta_t)$, $\theta_t \sim \mathcal{N}(\theta|\mu_t, \Sigma_t)$ with $\Sigma_t := \mathrm{diag}(1/(N(s_t + \tilde{\delta})))$, $\tilde{\delta} := \tau\delta/N$.
- Kronecker-factored approximate curvature [Zhang et al., 2018]

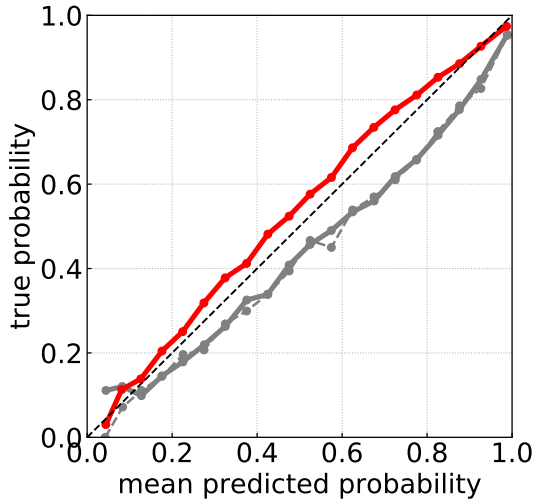
VOGN: Scaling to ImageNet

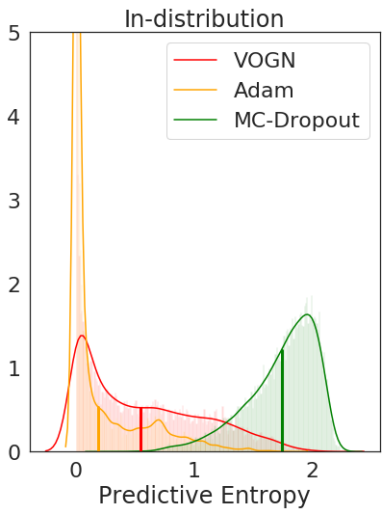
1. Distributed implementation: 128 GPUs
   - Data parallelisation
   - Monte-Carlo samples parallelisation
2. Data augmentation factor to increase dataset size N
   - Effectively downweights KL term
   - cf cold posterior effect
3. BatchNorm layers
4. Learning rate scheduling
5. Initialisation: mimic Adam
   - Same means and momentum initialisation
   - Warm up KL factor τ
   - Use first minibatch to initialise s_0

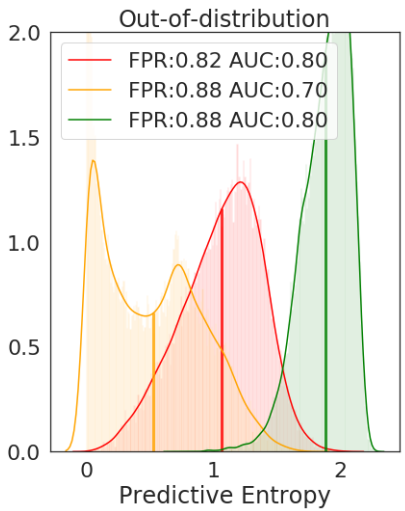
VOGN results

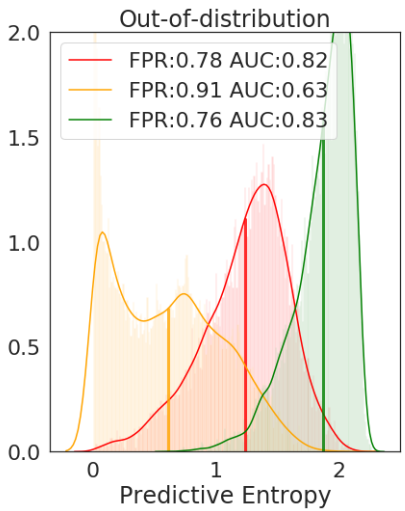


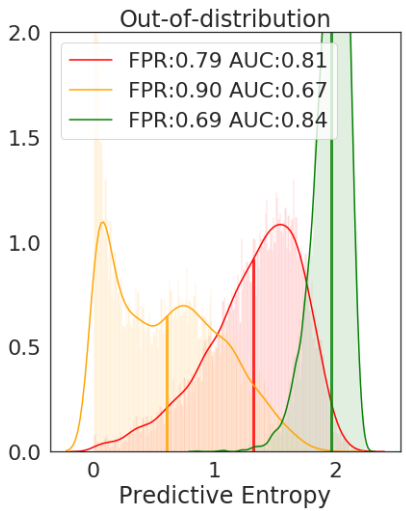












Miscellaneous Methods

1. Deep ensembles
2. MC Dropout
3. KFAC Laplace
4. To come: SWAG and fBNNs

Misc Methods: SWAG [Maddox et al., 2019]

- "Stochastic Weight Averaging Gaussians"
- There has been some theory suggesting that the SGD trajectory contains useful information about the posterior

1. Start at pre-trained model, then run normal SGD
2. Collect samples and fit to low-rank Gaussian
3. This is the 'posterior' which will be sampled from at test-time

Misc Methods: fBNNs [Sun et al., 2019]

- Maximise an ELBO defined directly on stochastic processes
- Can specify priors in function-space, e.g. Gaussian Processes
- Likelihood term is tractable
- KL term is difficult:

$$\text{KL term} \approx \mathbb{E}_{\mathbf{X} \sim c}\, \mathcal{KL}[q(\mathbf{f}^{\mathbf{X}}) || p(\mathbf{f}^{\mathbf{X}})]$$

- Sampling distribution c consists of random training inputs and random points
- Estimate using spectral Stein gradient estimator
- Also sample weights from $q(w)$ (reparameterisation trick)
- Demonstrate training with periodic kernels
- Scale to UCI, contextual bandits

Open Questions

BNN Priors - Are They Good?

Are our priors any good? Wenzel et al. [2020] argue not:

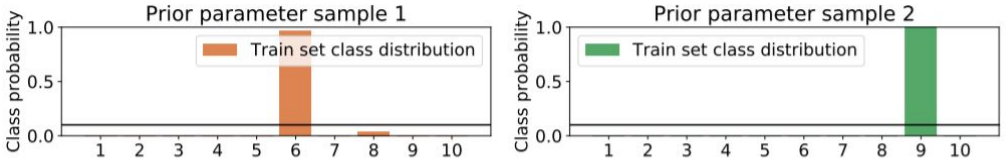


Figure 7. ResNet-20/CIFAR-10 typical prior predictive distributions for 10 classes under a $\mathcal{N}(0, I)$ prior averaged over the entire training set, $\mathbb{E}_{x\sim p(x)}[p(y|x, \boldsymbol{\theta}^{(i)})]$. Each plot is for one sample $\boldsymbol{\theta}^{(i)} \sim \mathcal{N}(0, I)$ from the prior. Given a sample $\boldsymbol{\theta}^{(i)}$ the average training data class distribution is highly concentrated around the same classes for all x.

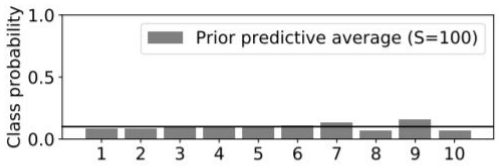
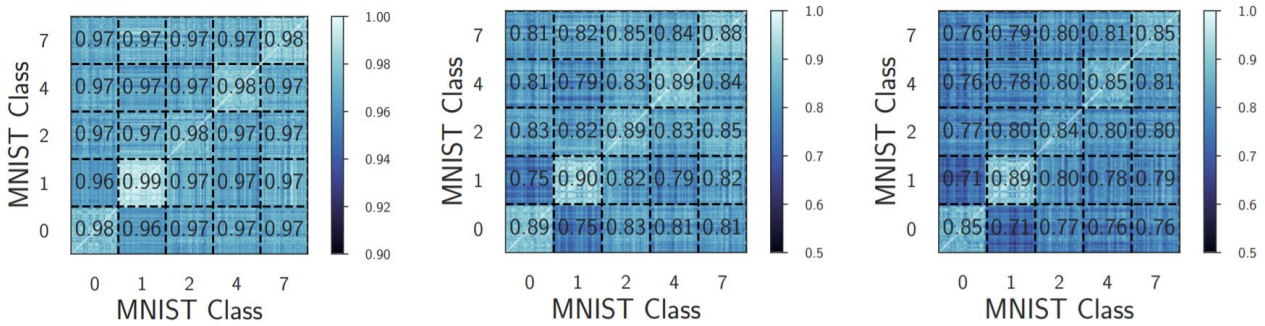


Figure 8. ResNet-20/CIFAR-10 prior predictive $\mathbb{E}_{x\sim p(x)}[\mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta})}[p(y|x, \boldsymbol{\theta})]]$ over 10 classes, estimated using $S = 100$ prior samples $\boldsymbol{\theta}^{(i)}$ and all training images.

Wilson and Izmailov [2020] argue yes:



(a) $\alpha = 0.02$ (b) $\alpha = 0.1$ (c) $\alpha = 1.$

Figure: Prior correlations for a $\mathcal{N}(0, \alpha I)$ weight prior.

But was BatchNorm on?

BNN Priors - Are They Good? (cont'd)

Is any sufficiently vague prior combined with a structured architecture (e.g., convolutional) good enough?

Questions:

1. From a pragmatist standpoint, is this true?
2. From a purist standpoint, how would we assess 'good enough'?

Priors - the Baby and the Bathwater

In the infinite width limit, deep BNNs $\rightarrow$ GPs [Matthews et al., 2018].

"... neural networks were meant to be intelligent models that discovered features and patterns in data. Gaussian processes in contrast are simply smoothing devices. How can Gaussian processes possibly replace neural networks?"

— MacKay [2003]

"... with Gaussian priors the contributions of individual hidden units are all negligible, and consequently, these units do not represent 'hidden features' that capture important aspects of the data."

— Neal [1995]

Priors - What Do We Lose in the GP Limit?

Questions:

1. Should we think of BNNs as just scalable GPs or something more than GPs?
2. How can we avoid GP behaviour?
3. What precisely distinguishes a feature-discovering model from a smoothing device?

Mathematical argument [Matthews, 2019]:
Let $X = \{x_n\}_{n=1}^N$ be regression training inputs, $y = \{y_n\}_{n=1}^N$ be outputs.

General regression algorithm predictive mean:

$$\hat{f}(x_*) = a(x_*, X, y)$$

GP predictive mean:

$$\hat{f}(x_*) = b(x_*, X)^T y$$

i.e. the GP mean is a linear transformation of the training outputs.

BNN posteriors

Are we doing good inference from a purist standpoint?
Can't tell by looking just at pragmatist metrics - could be a case of

$$\text{bad model} + \text{bad inference} = \text{good results (sometimes)}$$

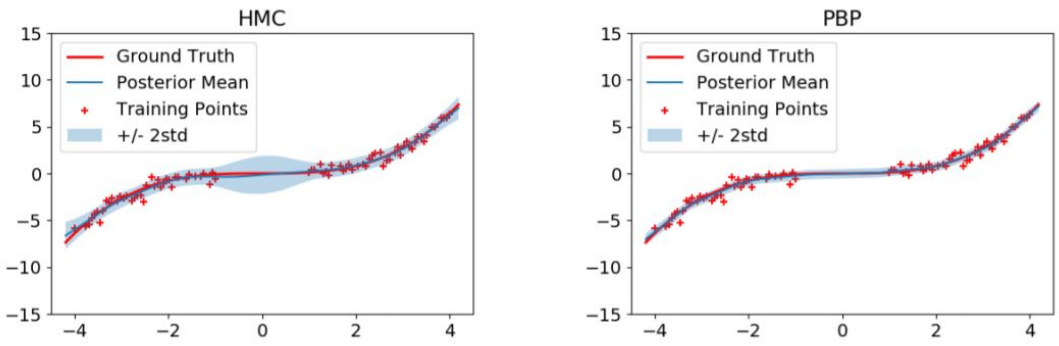


Figure: PBP obtains higher test log-likelihood than HMC due to model misspecification [Yao et al., 2019].

Question:

How close do you think BNN inference is to the Bayes posterior in function space?

Benchmarks and Metrics

Different papers use different benchmarks to assess "uncertainty"

1. Calibration curves
2. Deep Bayesian Bandits Showdown [Riquelme et al., 2018]
3. Dataset Shift [Snoek et al., 2019]
4. Diabetic Retinopathy task [Filos et al., 2019]

Questions:

1. Are these benchmarks testing something useful?
2. How can we improve?

Calibration curves

- True accuracy vs model's predicted accuracy
- Perfect calibration = diagonal line
- Need to bin along the x-axis

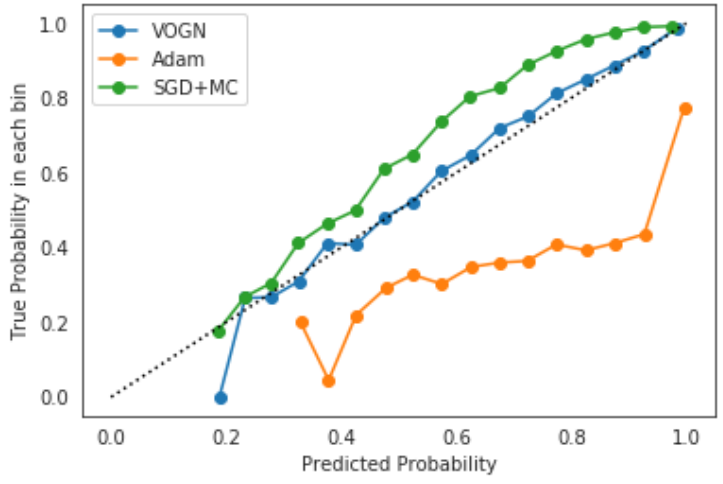


Figure: Calibration curve example. (Comparing VOGN, MC Dropout and Adam on AlexNet trained on CIFAR-10 [Osawa et al., 2019])

Bayesian Bandits [Riquelme et al., 2018]

- Thompson sampling for contextual bandits (deep RL)
  1. Observe context
  2. Sample model
  3. Take action = BestAction(context, model)
  4. Observe reward
  5. Update posterior distribution
- Motivations
  1. Thompson sampling works well for RL / contextual bandits
  2. Sequential decision-making scenario is realistic
     - May not have time to train until convergence (VI)
- Range of experiments: real-world (5 benchmarks) and toy (requiring more exploration)
- Metric: regret

Dataset shift [Snoek et al., 2019]

- Empirical study of uncertainty under distributional shift
- Motivations
  1. Want to evaluate predictive uncertainty
  2. Real-world applications have distributional shift
  3. Post-hoc calibration can give good results when iid data, but fail under input data shift
- Methods
  1. Covariate shift: corruptions and perturbations of input images
  2. A different (OOD) dataset

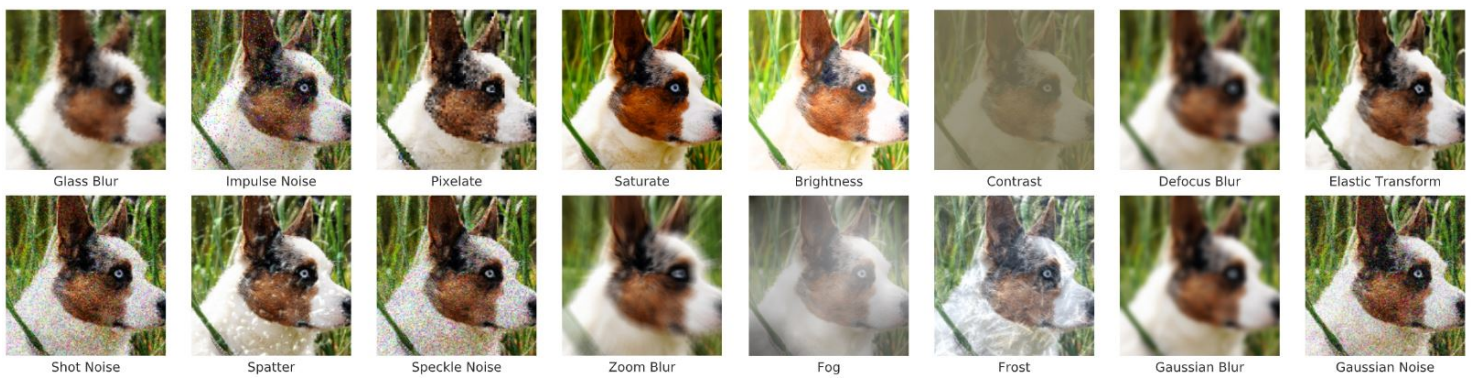


Figure: Different types of dataset shift [Snoek et al., 2019], all at intensity 3 out of 5.

- Metrics

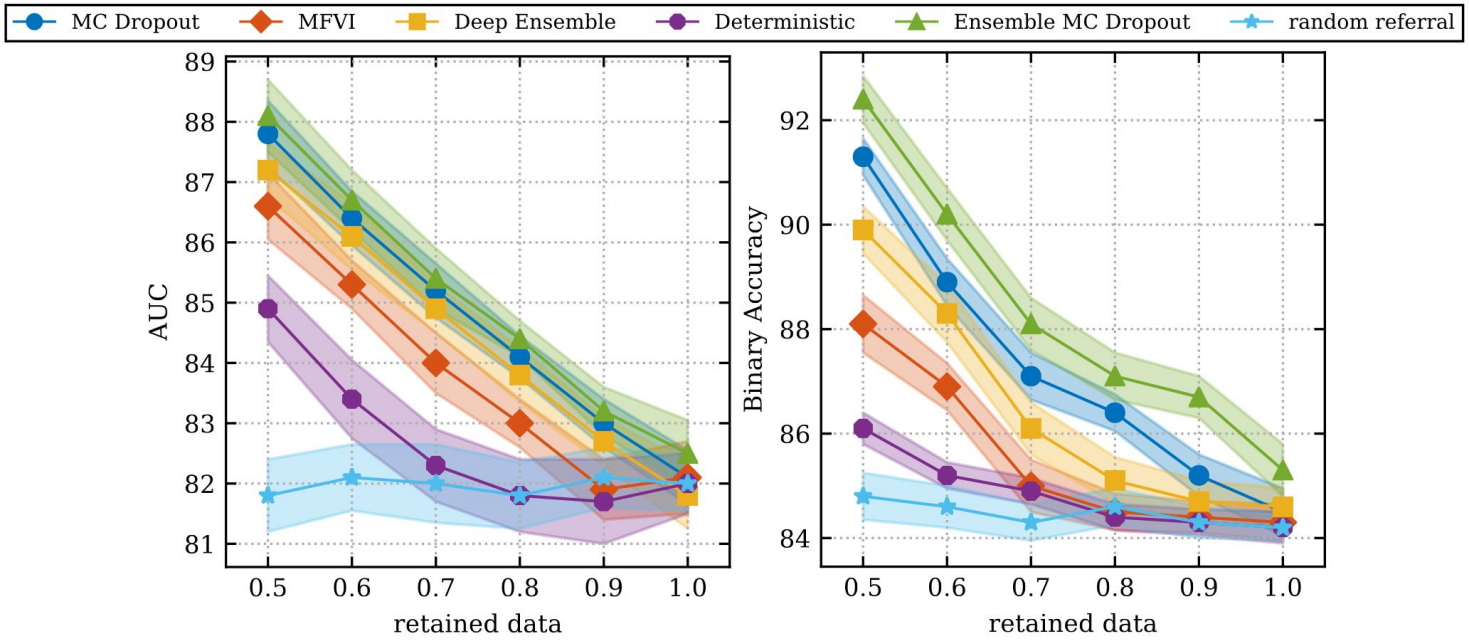
Diabetic Retinopathy Task [Filos et al., 2019]

- Aim: larger/more realistic benchmark than UCI
- Setup: Detect diabetic retinopathy from photos (binary classification)
- Key idea: referral to an expert
- OOD and distribution shift: different medical equipment / different patient populations
- Metrics
  1. Accuracy vs proportion retained data
  2. AUC vs proportion retained data
     - Out of retained data, ROC curve varies discrimination threshold and plots true positive rate vs false positive rate

Diabetic Retinopathy Task [Filos et al., 2019]

Metrics:

1. Accuracy vs proportion retained data
2. AUC vs proportion retained data
   - Out of retained data, ROC curve varies discrimination threshold and plots true positive rate vs false positive rate



(a) **test data**, AUC

(b) **test data**, accuracy

Good benchmarks?

References I

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. In *International Conference on Machine Learning*, pages 1613–1622, 2015.

T. Chen, E. Fox, and C. Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691, 2014.

A. Filos, S. Farquhar, A. N. Gomez, T. G. J. Rudner, Z. Kenton, L. Smith, M. Alizadeh, A. d. Kroon, and Y. Gal. A systematic comparison of bayesian deep learning robustness in diabetic retinopathy tasks. *arXiv preprint arXiv:1912.10481*, 2019.

D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

References II

D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems 32*, pages 13153–13164. Curran Associates, Inc., 2019.

A. G. d. G. Matthews. Gaussian process behaviour in wide deep neural networks. NeurIPS Bayesian Deep Learning Workshop Invited talk, 2019.

A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

R. M. Neal. *BAYESIAN LEARNING FOR NEURAL NETWORKS*. PhD thesis, University of Toronto, 1995.

References III

K. Osawa, S. Swaroop, M. E. E. Khan, A. Jain, R. Eschenhagen, R. E. Turner, and R. Yokota. Practical deep learning with bayesian principles. In *Advances in Neural Information Processing Systems 32*, pages 4287–4299. Curran Associates, Inc., 2019.

C. Riquelme, G. Tucker, and J. Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=SyYe6k-CW`.

J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J. Dillon, J. Ren, and Z. Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems 32*, pages 13991–14002. Curran Associates, Inc., 2019.

References IV

J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust bayesian neural networks. In *Advances in neural information processing systems*, pages 4134–4142, 2016.

S. Sun, G. Zhang, J. Shi, and R. Grosse. FUNCTIONAL VARIATIONAL BAYESIAN NEURAL NETWORKS. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=rkxacs0qY7`.

F. Wenzel, K. Roth, B. S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.

A. G. Wilson. The case for bayesian deep learning. *arXiv preprint arXiv:2001.10995*, 2020.

A. G. Wilson and P. Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.

References V

J. Yao, W. Pan, S. Ghosh, and F. Doshi-Velez. Quality of uncertainty quantification for bayesian neural network inference. *arXiv preprint arXiv:1906.09686*, 2019.

G. Zhang, S. Sun, D. Duvenaud, and R. Grosse. Noisy natural gradient as variational inference. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5852–5861, 2018.

R. Zhang, C. Li, J. Zhang, C. Chen, and A. G. Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. *arXiv preprint arXiv:1902.03932*, 2019.