Variational Inference of overparameterized Bayesian Neural Networks: a theoretical and empirical study

Anna Korba
CREST, ENSAE, Institut Polytechnique de Paris

Synergies between Data Science and PDE Analysis

Joint work with Tom Huix, Szymon Majewski, Eric Moulines (CMAP, Polytechnique) and Alain Durmus (ENS Cachan).

Outline

Bayesian inference

Let $\mathcal{D} = (x_i, y_i)_{i=1}^{m}$ a dataset of labelled examples $(x_i, y_i) \overset{i.i.d.}{\sim} P_{\text{data}}$.
Assume an underlying model parametrized by w, e.g. :

$$y = g(x, w) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, I_d)$$

Goal: learn the best distribution over w to fit the data.

Bayesian inference

Let $\mathcal{D} = (x_i, y_i)_{i=1}^m$ a dataset of labelled examples $(x_i, y_i) \overset{i.i.d.}{\sim} P_{\text{data}}$.
Assume an underlying model parametrized by w, e.g. :

$$y = g(x, w) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, I_d)$$

Goal: learn the best distribution over w to fit the data.

1. Compute the Likelihood:

$$p(\mathcal{D}|w) = \prod_{i=1}^m p(y_i|w, x_i) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^m \|y_i - g(x_i, w)\|^2\right).$$

Bayesian inference

Let $\mathcal{D} = (x_i, y_i)_{i=1}^m$ a dataset of labelled examples $(x_i, y_i) \overset{i.i.d.}{\sim} P_{\text{data}}$.
Assume an underlying model parametrized by w, e.g. :

$$y = g(x, w) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, I_d)$$

Goal: learn the best distribution over w to fit the data.

1. Compute the Likelihood:

$$p(\mathcal{D}|w) = \prod_{i=1}^m p(y_i|w, x_i) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^m \|y_i - g(x_i, w)\|^2\right).$$

2. Choose a prior distribution on the parameter:

$$w \sim p, \quad \text{e.g. } p(w) \propto \exp\left(-\frac{\|w\|^2}{2}\right).$$

Bayesian inference

Let $\mathcal{D} = (x_i, y_i)_{i=1}^m$ a dataset of labelled examples $(x_i, y_i) \overset{i.i.d.}{\sim} P_{\text{data}}$.
Assume an underlying model parametrized by w, e.g. :

$$y = g(x, w) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, I_d)$$

Goal: learn the best distribution over w to fit the data.

1. Compute the Likelihood:

$$p(\mathcal{D}|w) = \prod_{i=1}^m p(y_i|w, x_i) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^m \|y_i - g(x_i, w)\|^2\right).$$

2. Choose a prior distribution on the parameter:

$$w \sim p, \quad \text{e.g. } p(w) \propto \exp\left(-\frac{\|w\|^2}{2}\right).$$

3. Bayes' rule yields:

$$\pi(w) := p(w|\mathcal{D}) = \frac{p(\mathcal{D}|w)p(w)}{Z} \quad Z = \int_{\mathbb{R}^d} p(\mathcal{D}|w)p(w)dw$$

i.e. $\pi(w) \propto \exp\left(-V(w)\right), \quad V(w) = \frac{1}{2}\sum_{i=1}^m \|y_i - g(x_i, w)\|^2 + \frac{\|w\|^2}{2}.$

π is needed both for

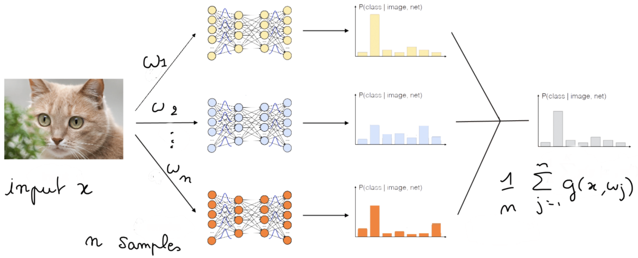
- prediction for a new input x: $y_{pred} = \int_{\mathbb{R}^d} g(x, w) d\pi(w)$
- measure uncertainty on the prediction.

π is needed both for

- prediction for a new input x: $y_{pred} = \int_{\mathbb{R}^d} g(x, w) d\pi(w)$
- measure uncertainty on the prediction.

Given a discrete approximation $\mu_n = \frac{1}{n} \sum_{j=1}^{n} \delta_{w_j}$ of π:

$$y_{pred} \approx \frac{1}{n} \sum_{j=1}^{n} g(x, w_j).$$

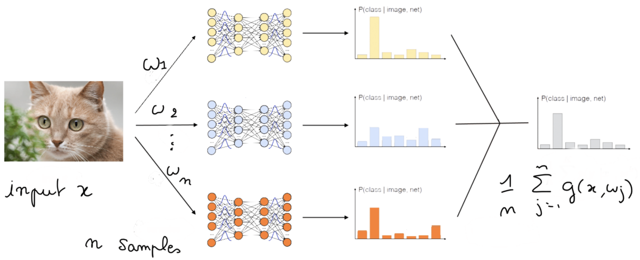


π is needed both for

- prediction for a new input x: $y_{pred} = \int_{\mathbb{R}^d} g(x, w) d\pi(w)$
- measure uncertainty on the prediction.

Given a discrete approximation $\mu_n = \frac{1}{n} \sum_{j=1}^{n} \delta_{w_j}$ of π:

$$y_{pred} \approx \frac{1}{n} \sum_{j=1}^{n} g(x, w_j).$$



Question: how can we approximate π?

Main methods for sampling

- Markov Chain Monte Carlo Methods (MCMC)
  generate a Markov chain in $\mathbb{R}^d$ whose law converges to $\pi \propto \exp(-V)$

  Example: Langevin Monte Carlo (LMC)

  $$w_{l+1} = w_l - \gamma \nabla V(w_l) + \sqrt{2\gamma}\epsilon_l, \ \epsilon_l \sim \mathcal{N}(0, I_d)$$

  other example: Hamiltonian Monte Carlo

- Variational inference (VI) methods
  approximate π with a parametric distribution by solving

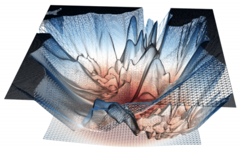
  $$\min_{\theta \in \Theta} \mathrm{KL}(p_\theta | \pi)$$

Difficult cases : non-convex potentials

Recall that

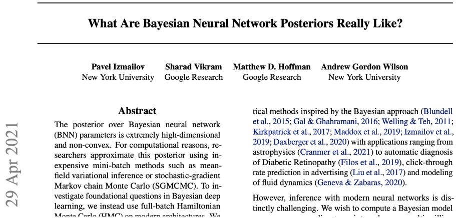
$$\pi(w) \propto \exp\left(-V(w)\right), \quad V(w) = \underbrace{\sum_{i=1}^{m} \|y_i - g(x_i, w)\|^2}_{\text{loss}} + \frac{\|w\|^2}{2}.$$

- if V is convex (e.g. $g(x, w) = \langle w, x \rangle$) MCMC methods work well and come with theoretical guarantees
- but if its not (e.g. $g(x, w)$ is a neural network), the situation is much more delicate



A highly nonconvex neural net loss surface. From `https://www.telesens.co/2019/01/16/neural-network-loss-visualization.`

- MCMC methods do not scale and require too many iterations ($\approx 10^4 - 10^6$) see [Izmailov et al., 2021] that run HMC over 512 Tensor processing unit (TPU) devices to obtain baselines on CIFAR10



What Are Bayesian Neural Network Posteriors Really Like?

Pavel Izmailov, New York University; Sharad Vikram, Google Research; Matthew D. Hoffman, Google Research; Andrew Gordon Wilson, New York University

29 Apr 2021

Abstract

The posterior over Bayesian neural network (BNN) parameters is extremely high-dimensional and non-convex. For computational reasons, researchers approximate this posterior using inexpensive mini-batch methods such as mean-field variational inference or stochastic-gradient Markov chain Monte Carlo (SGMCMC). To investigate foundational questions in Bayesian deep learning, we instead use full-batch Hamiltonian

tical methods inspired by the Bayesian approach (Blundell et al., 2015; Gal & Ghahramani, 2016; Welling & Teh, 2011; Kirkpatrick et al., 2017; Maddox et al., 2019; Izmailov et al., 2019; Daxberger et al., 2020) with applications ranging from astrophysics (Cranmer et al., 2021) to automatic diagnosis of Diabetic Retinopathy (Filos et al., 2019), click-through rate prediction in advertising (Liu et al., 2017) and modeling of fluid dynamics (Geneva & Zabaras, 2020).

However, inference with modern neural networks is distinctly challenging. We wish to compute a Bayesian model

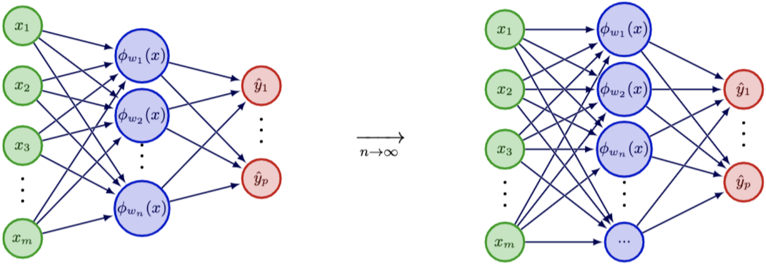
- VI remains a standard approach in Bayesian Deep Learning

Question: What can we say on the validity or limitations of VI for Bayesian Neural Networks (BNN)?

especially in the current, **overparametrized** regime era for neural networks.

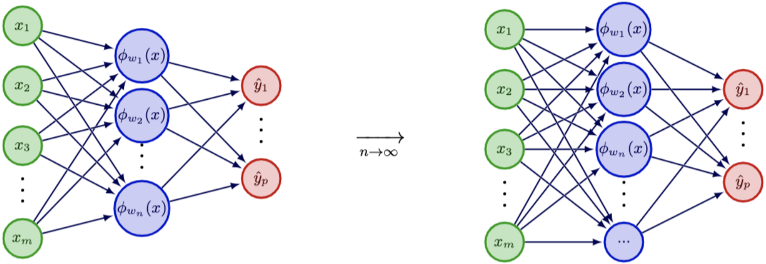
Infinite width neural network

consider a one-hidden-layer neural network, denote $\phi_{w_j}(x) = a_j \sigma(\langle b_j, x \rangle)$ the output of neuron j.



Infinite width neural network

consider a one-hidden-layer neural network, denote $\phi_{w_j}(x) = a_j \sigma(\langle b_j, x \rangle)$ the output of neuron j.



$$\min_{(w_j)_{j=1}^n \in \mathbb{R}^d} \mathbb{E}_{(x,y)\sim P_{data}} \left[\left\| y - \underbrace{\frac{1}{n}\sum_{j=1}^{n} \phi_{w_j}(x)}_{\hat{y}} \right\|^2 \right] \xrightarrow[n\to\infty]{} \min_{\mu \in \mathcal{P}(\mathbb{R}^d)} \underbrace{\mathbb{E}_{(x,y)\sim P_{data}} \left[\left\| y - \int_{\mathbb{R}^d} \phi_w(x) d\mu(w) \right\|^2 \right]}_{\mathcal{F}(\mu)}$$

Optimising the neural network is equivalent to minimizing $\mathcal{F}$.

[Chizat and Bach, 2018], [Rotskoff et al., 2019], [Mei et al., 2018a], [Arbel et al., 2019]...

Idea: consider a similar regime for VI on BNN.

Outline

Assume we have access to $\{(x_i, y_i)\}_{i=1}^{p}$ samples from the data distribution on $\mathsf{X} \times \mathsf{Y}$.

for each input $x \in \mathsf{X}$, the output prediction $f_{\boldsymbol{w}} : \mathsf{X} \to \mathbb{R}^{d_{\mathsf{Y}}}$ of the neural network can be written as:

$$f_{\boldsymbol{w}}(x) = \frac{1}{N} \sum_{j=1}^{N} s(w_j, x), \text{ with } s(w_j, x) = a_j \sigma(\langle b_j, x \rangle),$$

$$w_j = (a_j, b_j) \in \mathbb{R}^d, \ \boldsymbol{w} \in \mathbb{R}^{N \times d}.$$

Assume we have access to $\{(x_i, y_i)\}_{i=1}^{p}$ samples from the data distribution on $\mathrm{X} \times \mathrm{Y}$.

for each input $x \in \mathrm{X}$, the output prediction $f_{\boldsymbol{w}} : \mathrm{X} \to \mathbb{R}^{d_{\mathrm{Y}}}$ of the neural network can be written as:

$$f_{\boldsymbol{w}}(x) = \frac{1}{N}\sum_{j=1}^{N} s(w_j, x), \text{ with } s(w_j, x) = a_j \sigma(\langle b_j, x\rangle),$$

$$w_j = (a_j, b_j) \in \mathbb{R}^d, \ \boldsymbol{w} \in \mathbb{R}^{N\times d}.$$

Given a loss function $\ell : \mathrm{Y} \times \mathrm{Y} \to \mathbb{R}_+$, the likelihood is defined as

$$\mathrm{L}(y|x, \boldsymbol{w}) \propto \exp(-\ell(f_{\boldsymbol{w}}(x), y)) \ .$$

Assume we have access to $\{(x_i, y_i)\}_{i=1}^{p}$ samples from the data distribution on $\mathrm{X} \times \mathrm{Y}$.

for each input $x \in \mathrm{X}$, the output prediction $f_{\boldsymbol{w}} : \mathrm{X} \to \mathbb{R}^{d_\mathrm{Y}}$ of the neural network can be written as:

$$f_{\boldsymbol{w}}(x) = \frac{1}{N} \sum_{j=1}^{N} s(w_j, x), \text{ with } s(w_j, x) = a_j \sigma(\langle b_j, x \rangle),$$

$$w_j = (a_j, b_j) \in \mathbb{R}^d, \; \boldsymbol{w} \in \mathbb{R}^{N \times d}.$$

Given a loss function $\ell : \mathrm{Y} \times \mathrm{Y} \to \mathbb{R}_+$, the likelihood is defined as

$$\mathrm{L}(y|x, \boldsymbol{w}) \propto \exp(-\ell(f_{\boldsymbol{w}}(x), y)) \; .$$

Then, choosing a prior pdf P_0 on $\boldsymbol{w}$, the posterior pdf P of the weights is

$$P(\boldsymbol{w}) = \frac{P_0(\boldsymbol{w}) \prod_{i=1}^{p} \mathrm{L}(y_i|x_i, \boldsymbol{w})}{Z}.$$

Recall that VI considers a variational family of pdfs $\mathscr{F}_\Theta = \{q_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ and solves

$$\boldsymbol{\theta}^* \in \operatorname{argmin}_{\boldsymbol{\theta}\in\Theta} \mathsf{KL}(q_{\boldsymbol{\theta}} \mid P), \quad P(\boldsymbol{w}) = \frac{P_0(\boldsymbol{w}) \prod_{i=1}^{p} \mathrm{L}(y_i|x_i, \boldsymbol{w})}{Z}.$$

Recall that VI considers a variational family of pdfs $\mathscr{F}_\Theta = \{q_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ and solves

$$\boldsymbol{\theta}^* \in \operatorname{argmin}_{\boldsymbol{\theta}\in\Theta} \mathsf{KL}(q_{\boldsymbol{\theta}} \,|\, P), \quad P(\boldsymbol{w}) = \frac{P_0(\boldsymbol{w}) \prod_{i=1}^{p} \mathrm{L}(y_i|x_i, \boldsymbol{w})}{Z}.$$

It is equivalent to maximizing the Evidence Lower Bound (ELBO) defined for any $\boldsymbol{\theta} \in \Theta$ by:

$$\mathrm{ELBO}^N(\boldsymbol{\theta}) = \underbrace{-\,\mathsf{KL}(q_{\boldsymbol{\theta}} \,|\, P_0)}_{\text{(1) penalty term}} + \underbrace{\sum_{i=1}^{p} \int_{\mathbb{R}^{N\times d}} \log \mathrm{L}(y_i|x_i, \boldsymbol{w}) q_{\boldsymbol{\theta}}(\boldsymbol{w}) \mathrm{d}\boldsymbol{w}}_{\text{(2) data fitting term}} .$$

Recall that VI considers a variational family of pdfs $\mathscr{F}_\Theta = \{q_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ and solves

$$\boldsymbol{\theta}^* \in \operatorname{argmin}_{\boldsymbol{\theta}\in\Theta} \mathrm{KL}(q_{\boldsymbol{\theta}} \,|\, P), \quad P(\boldsymbol{w}) = \frac{P_0(\boldsymbol{w}) \prod_{i=1}^{p} \mathrm{L}(y_i|x_i, \boldsymbol{w})}{Z}.$$

It is equivalent to maximizing the Evidence Lower Bound (ELBO) defined for any $\boldsymbol{\theta} \in \Theta$ by:

$$\mathrm{ELBO}^N(\boldsymbol{\theta}) = \underbrace{-\,\mathrm{KL}(q_{\boldsymbol{\theta}} \,|\, P_0)}_{\text{(1) penalty term}} + \underbrace{\sum_{i=1}^{p} \int_{\mathbb{R}^{N\times d}} \log \mathrm{L}(y_i|x_i, \boldsymbol{w}) q_{\boldsymbol{\theta}}(\boldsymbol{w}) \mathrm{d}\boldsymbol{w}}_{\text{(2) data fitting term}} \, .$$

In practice, it is common to consider a tempered ELBO^N with $\eta < 1$:
[Zhang et al., 2018, Khan et al., 2018, Osawa et al., 2019, Ashukha et al., 2020]

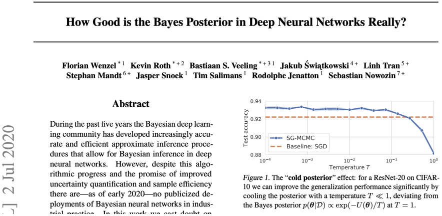
$$\mathrm{ELBO}_\eta^N(\boldsymbol{\theta}) = -\eta\, \mathrm{KL}(q_{\boldsymbol{\theta}} \,|\, P_0) + \sum_{i=1}^{p} \int_{\mathbb{R}^{N\times d}} \log \mathrm{L}(y_i|x_i, \boldsymbol{w}) q_{\boldsymbol{\theta}}(\boldsymbol{w}) \mathrm{d}\boldsymbol{w} \, .$$

$\mathrm{ELBO}_\eta^N \iff \mathrm{ELBO}^N$ where P is replaced by a tempered posterior $P_{T_N} \propto \mathrm{L}^{1/\eta} P_0$

In the VI literature, one can find for instance:

Reference	temperature η_N
[Zhang et al., 2018]	$\eta \in \{1/2, \dots, 1/10\}$
[Osawa et al., 2019]	$\eta \in \{1/5, \dots, 1/10\}$
[Ashukha et al., 2020]	η from 10^{-5} to 10^{-3}

η reweights the KL term and is typically smaller than 1 on current prediction tasks/neural nets architecture. From:



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 [7+]

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

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 $T = 1$.

2 Jul 2020

Our work: A first study on the choice of the cooling parameter η_N in the overparametrized regime (1 hidden layer neural net).

Our model - independent neurons, diagonal Gaussians

We consider a prior on $\boldsymbol{w} \in \mathbb{R}^{N\times d}$ which factorize over the weights, i.e., of the form

$$P_0(\boldsymbol{w}) = \prod_{j=1}^{N} P_0^1(w_j) \ ,$$

and similarly for the variational posterior

$$q_{\boldsymbol{\theta}}(\boldsymbol{w}) = \prod_{i=1}^{N} q_{\theta_j}^1(w_j)$$

where P_0^1 and $\{q_{\theta_j}^1\}_{j=1}^N$ are distributions over $\mathbb{R}^d$.

$$\mathrm{T}_\theta : z \mapsto \mu + \sigma \odot z \ , \quad \theta = (\mu, \sigma) \in \mathbb{R}^{2d}$$

where $\odot$ is the component wise product.

In this case, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N) \in \mathbb{R}^{N\times 2d}$.

Recall the ELBO:

$$\mathrm{ELBO}_{\eta}^{N}(\boldsymbol{\theta}) = -\eta \, \mathsf{KL}(q_{\boldsymbol{\theta}} \mid P_0) + \sum_{i=1}^{p} \int_{\mathbb{R}^{N \times d}} \log \mathrm{L}(y_i | x_i, \boldsymbol{w}) q_{\boldsymbol{\theta}}(\boldsymbol{w}) \mathrm{d}\boldsymbol{w} \; .$$

Recall the ELBO:

$$\mathrm{ELBO}_\eta^N(\boldsymbol{\theta}) = -\eta\,\mathrm{KL}(q_{\boldsymbol{\theta}} \mid P_0) + \sum_{i=1}^{p} \int_{\mathbb{R}^{N\times d}} \log \mathrm{L}(y_i | x_i, \boldsymbol{w}) q_{\boldsymbol{\theta}}(\boldsymbol{w}) \mathrm{d}\boldsymbol{w} \ .$$

To make the dependence in N more explicit, we can rewrite it as:

$$\mathrm{ELBO}_\eta^N(\boldsymbol{\theta}) = -\eta \underbrace{\sum_{j=1}^{N} \mathrm{KL}(q^1_{\theta_j} | P^1_0)}_{(1)} - \underbrace{\sum_{i=1}^{p} \mathrm{G}^N_\Theta(\boldsymbol{\theta}; (x_i, y_i))}_{(2)}$$

where, denoting the output of a neuron parametrized by $\theta \in \mathbb{R}^d$ for an input x_i by

$$\phi(\theta, z, x_i) = s(\mathrm{T}_\theta(z), x_i) \ ,$$

and $\boldsymbol{z} = (z_1, \dots, z_N) \in \mathbb{R}^{d\times N}$,

$$\mathrm{G}^N_\Theta(\boldsymbol{\theta}; (x, y)) = \int \ell\left(y, \sum_{j=1}^{N} \frac{\phi(\theta_j, z_j, x)}{N}\right) \gamma^{\otimes N}(\mathrm{d}\boldsymbol{z}) \ .$$

Recall the ELBO:

$$\mathrm{ELBO}^N_\eta(\boldsymbol{\theta}) = -\eta\,\mathsf{KL}(q_{\boldsymbol{\theta}} \mid P_0) + \sum_{i=1}^{p} \int_{\mathbb{R}^{N\times d}} \log \mathrm{L}(y_i|x_i, \boldsymbol{w}) q_{\boldsymbol{\theta}}(\boldsymbol{w}) \mathrm{d}\boldsymbol{w} \ .$$

To make the dependence in N more explicit, we can rewrite it as:

$$\mathrm{ELBO}^N_\eta(\boldsymbol{\theta}) = -\eta \underbrace{\sum_{j=1}^{N} \mathsf{KL}(q^1_{\theta_j}|P^1_0)}_{(1)} - \underbrace{\sum_{i=1}^{p} \mathrm{G}^N_\Theta(\boldsymbol{\theta};(x_i,y_i))}_{(2)}$$

where, denoting the output of a neuron parametrized by $\theta \in \mathbb{R}^d$ for an input x_i by

$$\phi(\theta, z, x_i) = s(\mathrm{T}_\theta(z), x_i) \ ,$$

and $\boldsymbol{z} = (z_1, \ldots, z_N) \in \mathbb{R}^{d\times N}$,

$$\mathrm{G}^N_\Theta(\boldsymbol{\theta};(x,y)) = \int \ell\left(y, \sum_{j=1}^{N} \frac{\phi(\theta_j, z_j, x)}{N}\right) \gamma^{\otimes N}(\mathrm{d}\boldsymbol{z}) \ .$$

Problem: (1) scales as $\mathcal{O}(N)$, while (2) scales as $\mathcal{O}(p)$ and does not grow with N if the variance of $q_{\boldsymbol{\theta}}$ does not scale with N.

$\Longrightarrow$ **(1) becomes predominant as $N \to \infty$!**

The ELBO in our model

Proposition. Let $\boldsymbol{\theta}^{*,N} = \mathrm{argmax}_{\boldsymbol{\theta}\in\Theta}\mathrm{ELBO}^N(\boldsymbol{\theta})$. Assume that $P_0 \in \mathscr{F}_\Theta$ where $\mathscr{F}_\Theta$ are diagonal Gaussians, that l is the square loss or cross-entropy, Lipschitz activation functions for the neural network, and that X is compact. Then,

$$\mathrm{KL}(q_{\boldsymbol{\theta}^{*,N}}, P_0) \to 0 \text{ as } N \to \infty.$$

The ELBO in our model

Proposition. Let $\boldsymbol{\theta}^{*,N} = \mathrm{argmax}_{\boldsymbol{\theta}\in\Theta}\mathrm{ELBO}^N(\boldsymbol{\theta})$. Assume that $P_0 \in \mathscr{F}_\Theta$ where $\mathscr{F}_\Theta$ are diagonal Gaussians, that l is the square loss or cross-entropy, Lipschitz activation functions for the neural network, and that X is compact. Then,

$$\mathrm{KL}(q_{\boldsymbol{\theta}^{*,N}}, P_0) \to 0 \text{ as } N \to \infty.$$

inspired from [Coker et al., 2021] that show a similar result when l is the square loss and activation functions are odd.

The ELBO in our model

Proposition. Let $\boldsymbol{\theta}^{*,N} = \mathrm{argmax}_{\boldsymbol{\theta}\in\Theta}\mathrm{ELBO}^N(\boldsymbol{\theta})$. Assume that $P_0 \in \mathscr{F}_\Theta$ where $\mathscr{F}_\Theta$ are diagonal Gaussians, that l is the square loss or cross-entropy, Lipschitz activation functions for the neural network, and that X is compact. Then,

$$\mathrm{KL}(q_{\boldsymbol{\theta}^{*,N}}, P_0) \to 0 \text{ as } N \to \infty.$$

inspired from [Coker et al., 2021] that show a similar result when l is the square loss and activation functions are odd.

Idea of the proof: By the optimality of $\boldsymbol{\theta}^{\star,N}$, we have:

$$-\mathrm{KL}(q_{\boldsymbol{\theta}^{*,N}}|P_0) - \mathcal{L}(q_{\boldsymbol{\theta}^{*,N}}) = \mathrm{ELBO}^N(\boldsymbol{\theta}^\star) \geq \mathrm{ELBO}^N(\boldsymbol{\theta}_0) = -\mathcal{L}(P_0)$$

Hence,

$$\mathrm{KL}(q_{\boldsymbol{\theta}^{*,N}}|P_0) \leq \mathcal{L}(P_0) - \mathcal{L}(q_{\boldsymbol{\theta}^{*,N}}).$$

Then show that both terms on the r.h.s. have the same finite limit.

Outline

First step: generalize the definition of ELBO_η^N defined in over $\mathbb{R}^{N\times 2d}$ to probability measures ν on $\mathbb{R}^{2d}$.

First step: generalize the definition of ELBO_η^N defined in over $\mathbb{R}^{N\times 2d}$ to probability measures ν on $\mathbb{R}^{2d}$.

Recall that

$$\mathrm{ELBO}_\eta^N(\boldsymbol{\theta}) = -\eta \sum_{j=1}^{N} \mathrm{KL}(q^1_{\theta_j} | P^1_0) - \sum_{i=1}^{p} \mathrm{G}^N_\Theta(\boldsymbol{\theta}; (x_i, y_i))$$

where, denoting $\boldsymbol{z} = (z_1, \ldots, z_N) \in \mathbb{R}^{d\times N}$,

$$\mathrm{G}^N_\Theta(\boldsymbol{\theta}; (x, y)) = \int \ell\left(y, \sum_{j=1}^{N} \frac{\phi(\theta_j, z_j, x)}{N}\right) \gamma^{\otimes N}(\mathrm{d}\boldsymbol{z}) \;.$$

First step: generalize the definition of ELBO_{η}^{N} defined in over $\mathbb{R}^{N\times 2d}$ to probability measures ν on $\mathbb{R}^{2d}$.

Recall that

$$\mathrm{ELBO}_{\eta}^{N}(\boldsymbol{\theta}) = -\eta \sum_{j=1}^{N} \mathsf{KL}(q_{\theta_j}^{1}|P_0^{1}) - \sum_{i=1}^{p} \mathrm{G}_{\Theta}^{N}(\boldsymbol{\theta};(x_i,y_i))$$

where, denoting $\boldsymbol{z} = (z_1,\dots,z_N) \in \mathbb{R}^{d\times N}$,

$$\mathrm{G}_{\Theta}^{N}(\boldsymbol{\theta};(x,y)) = \int \ell\left(y, \sum_{j=1}^{N} \frac{\phi(\theta_j, z_j, x)}{N}\right) \gamma^{\otimes N}(\mathrm{d}\boldsymbol{z}) \ .$$

Define

$$\nu_N^{\boldsymbol{\theta}} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_i} \ , \tag{1}$$

Proposition For any $N \in \mathbb{N}$, there exists a function F_{η}^{N} defined over measures of the form (1), such that $\mathrm{ELBO}_{\eta}^{N}(\boldsymbol{\theta}) = \mathrm{F}_{\eta}^{N}(\nu_N^{\boldsymbol{\theta}})$ for any $\boldsymbol{\theta} \in \mathbb{R}^{N\times 2d}$.

First step: generalize the definition of ELBO_η^N defined in over $\mathbb{R}^{N\times 2d}$ to probability measures ν on $\mathbb{R}^{2d}$.

Recall that

$$\mathrm{ELBO}_\eta^N(\boldsymbol{\theta}) = -\eta \sum_{j=1}^{N} \mathsf{KL}(q^1_{\theta_j}|P^1_0) - \sum_{i=1}^{p} \mathrm{G}^N_\Theta(\boldsymbol{\theta};(x_i,y_i))$$

where, denoting $\boldsymbol{z} = (z_1,\ldots,z_N) \in \mathbb{R}^{d\times N}$,

$$\mathrm{G}^N_\Theta(\boldsymbol{\theta};(x,y)) = \int \ell\left(y, \sum_{j=1}^{N} \frac{\phi(\theta_j,z_j,x)}{N}\right) \gamma^{\otimes N}(\mathrm{d}\boldsymbol{z}) \; .$$

Define

$$\nu_N^{\boldsymbol{\theta}} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_i} \; , \tag{1}$$

Proposition For any $N \in \mathbb{N}$, there exists a function F_η^N defined over measures of the form (1), such that $\mathrm{ELBO}_\eta^N(\boldsymbol{\theta}) = \mathrm{F}_\eta^N(\nu_N^{\boldsymbol{\theta}})$ for any $\boldsymbol{\theta} \in \mathbb{R}^{N\times 2d}$.

Problem: F_η^N cannot be non-trivially extended to a functional defined for a general probability measure on $\mathbb{R}^{2d}$.

We show that, when restricted to empirical probabilities, F_η^N is a perturbation, as $N \to +\infty$, of the functional $\tilde{\mathrm{F}}_\eta^N$ defined over all $\mathcal{P}(\mathbb{R}^{2d})$ by

$$\tilde{\mathrm{F}}_\eta^N(\nu) = -\sum_{i=1}^{p} \tilde{\mathrm{G}}(\nu; (x_i, y_i)) - \eta N \int \mathrm{KL}(q_\theta^1 | P_0^1) \mathrm{d}\nu(\theta) \ ,$$

where

$$\tilde{\mathrm{G}}(\nu; (x, y)) = \ell \left(y, \underbrace{\iint \phi(\theta, z, x) \mathrm{d}\nu(\theta) \mathrm{d}\gamma(z)}_{\iint s(T_\theta(z), x) d\gamma(z) d\nu(\theta)} \right) \ ,$$

Remark:

- $\tilde{\mathrm{G}}$ differs from G_Θ^N through the integration "inside" the loss
- $\tilde{\mathrm{G}}$ resembles the data fitting term one can find in [Chizat and Bach, 2018, Mei et al., 2018b]... (classical NN)

Theorem: Under mild assumptions on the loss, activation functions, prior, X, Y; there exists $C \geq 0$ such that for any $N, p \in \mathbb{N}$, $\{(x_i, y_i)\}_{i=1}^p \in (\mathrm{X} \times \mathrm{Y})^p$, $\boldsymbol{\theta} \in \Xi^N$ and $\eta > 0$,

$$|\mathrm{ELBO}_\eta^N(\boldsymbol{\theta}) - \tilde{\mathrm{F}}_\eta^N(\nu_N^{\boldsymbol{\theta}})| \leq Cp/N \ ,$$

It is now much clearer how to define a **balanced functional** over $\mathcal{P}(\mathbb{R}^{2d})$.

We now set $\eta = \tau p/N$ with $\tau > 0$.

With this particular choice, $\tilde{\mathrm{F}}_\eta^N$ depends only on the number of observations p but no longer on the number of neurons N. We denote, for that particular choice of η_N,

$$\mathcal{F}(\nu) = p^{-1}\tilde{\mathrm{F}}_\eta^N(\nu) = -\frac{1}{p}\sum_{i=1}^{p}\tilde{\mathrm{G}}(\nu;(x_i,y_i)) - \tau\int \mathrm{KL}(q_\theta^1|P_0^1)\mathrm{d}\nu(\theta)\ .$$

Outline

We illustrate our findings and their practical implications for image classification on standard datasets (MNIST, CIFAR-10), with a simple one hidden layer architecture and a Resnet20 respectively.

We illustrate our findings and their practical implications for image classification on standard datasets (MNIST, CIFAR-10), with a simple one hidden layer architecture and a Resnet20 respectively.

For each neuron, we use a centered Gaussian prior with variance $1/5$, following [Osawa et al., 2019]. We train each BNN by Bayes by Backprop [Blundell et al., 2015].

Metrics:
For an input $x \in \mathrm{X}$, the predictive probability of a class c by a neural network with weights $\boldsymbol{w}$ is defined by $\Psi_c(f_{\boldsymbol{w}}(x))$, where $\Psi_c(f_{\boldsymbol{w}}(x))$ denotes the c-th component of the softmax function applied to the output $f_{\boldsymbol{w}}(x) \in \mathbb{R}^{n_l}$ of the neural network.

- Accuracy: number of correct predictions
- NLL: $\sum_{i=1}^{p} \int_{\mathbb{R}^{N \times d}} \ell_{\mathrm{CE}}(y_i, f_{\boldsymbol{w}}(x_i)) q_{\boldsymbol{\theta}}(\boldsymbol{w}) \mathrm{d}\boldsymbol{w}$ where ℓ_{CE} is the cross-entropy loss
- ECE: measures if the predictive posterior is close to the true probability for each class $c \in \{1, \ldots, n_l\}$.
- Confidence: $conf(x) = \max_{c \in \{1, \ldots, n_l\}} \Psi_c(f_{\boldsymbol{w}}(x))$ averaged over all points x.

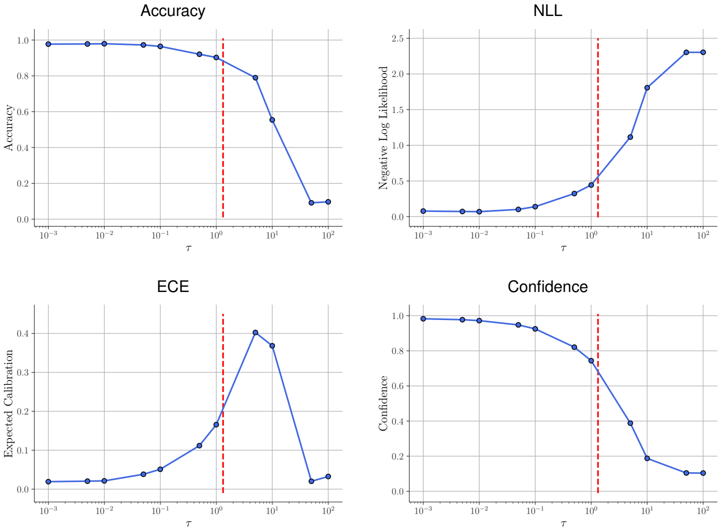


Figure: Effect of the temperature for a Linear BNN (one hidden layer, relU activations) trained on MNIST. No cooling $\eta_N = 1$ is indicated by a red line.

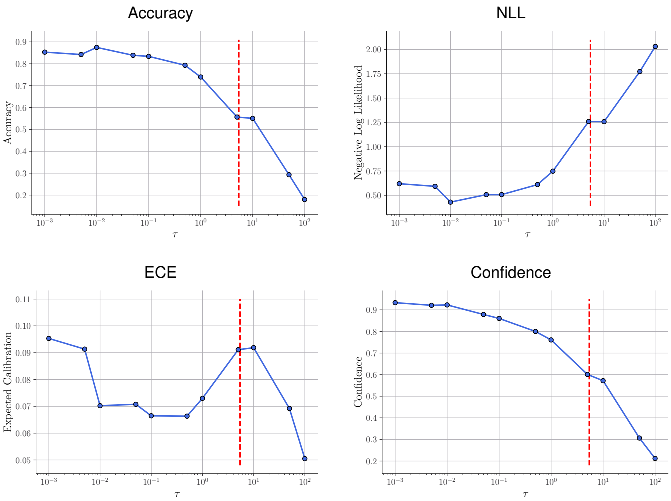


Figure: Effect of the temperature for a Resnet20 trained on CIFAR-10. No cooling $\eta_N = 1$ is indicated by a red line.

These experiments show that balancing the ELBO with the scaling $\eta_N = \tau p/N$ generalizes to much more complex architectures than a one hidden layer.

Conclusion

- We have identified that the ELBO should be tempered according to a temperature proportional to p/N, where p is the number of data points and N the number of parameters, when using product priors and posteriors
- With this choice, ELBO converges to a well-defined functional over the space of probability measures and one could analyze gradient descent dynamics through Wasserstein gradient flows
- Alternatively [Tran et al., 2020, Fortuin et al., 2021, Ober and Aitchison, 2021, Sun et al., 2019] have proposed the design of new priors which introduce correlation amongst the weights, however these models may be harder to train

Thank you! Questions?

References I

Arbel, M., Korba, A., Salim, A., and Gretton, A. (2019).
Maximum mean discrepancy gradient flow.
arXiv preprint arXiv:1906.04370.

Ashukha, A., Lyzhov, A., Molchanov, D., and Vetrov, D. (2020).
Pitfalls of in-domain uncertainty estimation and ensembling in deep learning.
arXiv preprint arXiv:2002.06470.

Bhattacharya, A., Pati, D., and Yang, Y. (2019).
Bayesian fractional posteriors.
The Annals of Statistics, 47(1):39–66.

References II

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

Chizat, L. and Bach, F. (2018).
On the global convergence of gradient descent for over-parameterized models using optimal transport.
NIPS.

Coker, B., Pan, W., and Doshi-Velez, F. (2021).
Wide mean-field variational bayesian neural networks ignore the data.
arXiv preprint arXiv:2106.07052.

References III

Fortuin, V., Garriga-Alonso, A., Wenzel, F., Rätsch, G., Turner, R., van der Wilk, M., and Aitchison, L. (2021).
Bayesian neural network priors revisited.
arXiv preprint arXiv:2102.06571.

Grünwald, P. (2012).
The safe bayesian.
In *International Conference on Algorithmic Learning Theory*, pages 169–183. Springer.

Grunwald, P., Steinke, T., and Zakynthinou, L. (2021).
Pac-bayes, mac-bayes and conditional mutual information: Fast rate bounds that handle general vc classes.
In Belkin, M. and Kpotufe, S., editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 2217–2247. PMLR.

References IV

Grünwald, P. and Van Ommen, T. (2017).
Inconsistency of bayesian inference for misspecified linear models, and a proposal for repairing it.
Bayesian Analysis, 12(4):1069–1103.

Heide, R., Kirichenko, A., Grunwald, P., and Mehta, N. (2020).
Safe-bayesian generalized linear regression.
In *International Conference on Artificial Intelligence and Statistics*, pages 2623–2633. PMLR.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. (2021).
What are bayesian neural network posteriors really like?
International Conference on Machine Learning.

References V

Khan, M., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018).
Fast and scalable bayesian deep learning by weight-perturbation in adam.
In *International Conference on Machine Learning*, pages 2611–2620. PMLR.

Mei, S., Montanari, A., and Nguyen, P.-M. (2018a).
A mean field view of the landscape of two-layer neural networks.
Proceedings of the National Academy of Sciences, 115(33):E7665–E7671.

References VI

Mei, S., Montanari, A., and Nguyen, P.-M. (2018b).
A mean field view of the landscape of two-layer neural networks.
Proceedings of the National Academy of Sciences, 115(33):E7665–E7671.

Ober, S. W. and Aitchison, L. (2021).
Global inducing point variational posteriors for bayesian neural networks and deep gaussian processes.
In *International Conference on Machine Learning*, pages 8248–8259. PMLR.

Osawa, K., Swaroop, S., Jain, A., Eschenhagen, R., Turner, R. E., Yokota, R., and Khan, M. E. (2019).
Practical deep learning with bayesian principles.
arXiv preprint arXiv:1906.02506.

References VII

Rotskoff, G., Jelassi, S., Bruna, J., and Vanden-Eijnden, E. (2019).
Global convergence of neuron birth-death dynamics.
In *ICML*.

Sun, S., Zhang, G., Shi, J., and Grosse, R. (2019).
Functional variational bayesian neural networks.
In *International Conference on Learning Representations*.

Tran, B.-H., Rossi, S., Milios, D., and Filippone, M. (2020).
All you need is a good functional prior for bayesian deep learning.
arXiv preprint arXiv:2011.12829.

References VIII

Wenzel, F., Roth, K., Veeling, B. S., Swiatkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. (2020).
How good is the bayes posterior in deep neural networks really?
International conference on machine learning.

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

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. (2018).
Noisy natural gradient as variational inference.
In *International Conference on Machine Learning*, pages 5852–5861. PMLR.