Generalised Variational Inference Meets Bayesian Deep Learning

Dino Sejdinovic (Adelaide)
joint work with Veit D. Wild (Oxford), Robert Hu (Amazon),
Sahra Ghalebikesabi (Oxford), Jeremias Knoblauch (UCL)

RPTU Kaiserslautern-Landau
15 March 2024

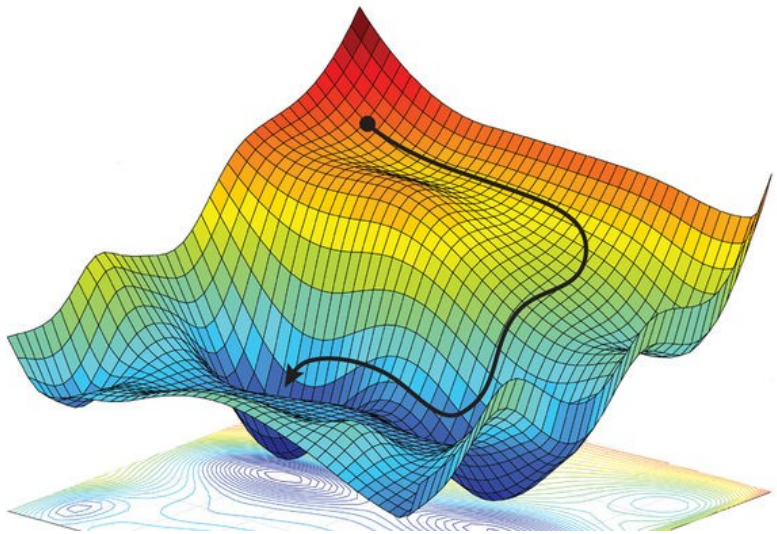
Deep Learning

Observe data $\mathcal{D} := \{(x_n, y_n) \mid n = 1, \ldots, N\}$.

- Likelihood is given by

$$p(\mathcal{D}|w) = \prod_{n=1}^{N} p(y_n|f(x_n; w)), \text{ where e.g. } y_n|f(x_n; w) \sim \mathcal{N}(f(x_n; w), \sigma^2),$$

  and $f(\cdot; w)$ is a neural network with parameters w.
- Deep learning finds good optima of $\log p(\mathcal{D}|w)$.



Bayesian Deep Learning

Bayesian neural network:
Place a prior distribution $p(w)$ on the network weights. This results in a prior distribution on random functions, i.e. $f(x; W)$, $W \sim p(w)$. Find posterior $p(w|\mathcal{D})$.

Bayesian Deep Learning

Bayesian neural network:
Place a prior distribution $p(w)$ on the network weights. This results in a prior distribution on random functions, i.e. $f(x; W)$, $W \sim p(w)$. Find posterior $p(w|\mathcal{D})$.

Why Bayesian Deep Learning?

Bayesian Deep Learning

Bayesian neural network:
Place a prior distribution $p(w)$ on the network weights. This results in a prior distribution on random functions, i.e. $f(x; W)$, $W \sim p(w)$. Find posterior $p(w|\mathcal{D})$.

Why Bayesian Deep Learning?

- *Posteror predictive*: for any new $x^* \in \mathcal{X}$ averages over many individual neural networks – and these are weighted by their agreement with observed data.

$$p(y^*|\mathcal{D}) = \int p(y^*|w)\, p(w|\mathcal{D})\, dw$$
$$= \int p(y^*|f(x^*; w))\, p(w|\mathcal{D})\, dw$$

Bayesian Deep Learning

Bayesian neural network:
Place a prior distribution $p(w)$ on the network weights. This results in a prior distribution on random functions, i.e. $f(x; W)$, $W \sim p(w)$. Find posterior $p(w|\mathcal{D})$.

Why Bayesian Deep Learning?

- *Posteror predictive*: for any new $x^* \in \mathcal{X}$ averages over many individual neural networks – and these are weighted by their agreement with observed data.

$$p(y^*|\mathcal{D}) = \int p(y^*|w)\, p(w|\mathcal{D})\, dw$$
$$= \int p(y^*|f(x^*; w))\, p(w|\mathcal{D})\, dw$$

- *Uncertainty quantification*: disagreement between the individual neural networks outside of the data is captured by the posterior predictive.

Bayesian Deep Learning

Bayesian neural network:
Place a prior distribution $p(w)$ on the network weights. This results in a prior distribution on random functions, i.e. $f(x; W)$, $W \sim p(w)$. Find posterior $p(w|\mathcal{D})$.

Why Bayesian Deep Learning?

- *Posteror predictive*: for any new $x^* \in \mathcal{X}$ averages over many individual neural networks – and these are weighted by their agreement with observed data.

$$p(y^*|\mathcal{D}) = \int p(y^*|w)\, p(w|\mathcal{D})\, dw$$
$$= \int p(y^*|f(x^*; w))\, p(w|\mathcal{D})\, dw$$

- *Uncertainty quantification*: disagreement between the individual neural networks outside of the data is captured by the posterior predictive.

But: the posterior $p(w|\mathcal{D})$ is intractable – approximations are required.

Posterior Approximations

Typical approximations include:

Posterior Approximations

Typical approximations include:

- **Sampling**
  - Hamiltonian Monte Carlo [Neal, 2012, Chen et al., 2014]
  - Langevin Dynamics [Welling and Teh, 2011]
  - ...And their stochastic variants.

Posterior Approximations

Typical approximations include:

- **Sampling**
  - Hamiltonian Monte Carlo [Neal, 2012, Chen et al., 2014]
  - Langevin Dynamics [Welling and Teh, 2011]
  - ...And their stochastic variants.

Often not sufficiently scalable for most deep learning applications. Challenging due to multimodality and high dimensionality.

Posterior Approximations

Typical approximations include:

- **Sampling**
  - Hamiltonian Monte Carlo [Neal, 2012, Chen et al., 2014]
  - Langevin Dynamics [Welling and Teh, 2011]
  - ...And their stochastic variants.

Often not sufficiently scalable for most deep learning applications. Challenging due to multimodality and high dimensionality.

- **Variational inference**
  - ...And its stochastic variants e.g. [Graves, 2011]

Weight-Space Variational Inference

Variational approximation:
Let $q(w) = q(w;\nu)$ be a class of distributions with (variational) parameters ν.
We want $q(w;\nu)$ to approximate the true posterior $p(w|\mathcal{D})$. Learn ν by maximising the ELBO criterion *lower bound on the marginal likelihood*:

$$\mathcal{L}(\nu) := \mathbb{E}_{q(w)}\big[\log p(y|w)\big] - \mathbb{D}_{KL}\big(q(w)||p(w)\big), \tag{1}$$

which is (often) tractable, e.g. if $q(w)$ and $p(w)$ are normal.

Weight-Space Variational Inference

Variational approximation:
Let $q(w) = q(w;\nu)$ be a class of distributions with (variational) parameters ν.
We want $q(w;\nu)$ to approximate the true posterior $p(w|\mathcal{D})$. Learn ν by maximising the ELBO criterion *lower bound on the marginal likelihood*:

$$\mathcal{L}(\nu) := \mathbb{E}_{q(w)}\big[\log p(y|w)\big] - \mathbb{D}_{KL}\big(q(w)||p(w)\big), \tag{1}$$

which is (often) tractable, e.g. if $q(w)$ and $p(w)$ are normal.

Problems:

Weight-Space Variational Inference

Variational approximation:
Let $q(w) = q(w;\nu)$ be a class of distributions with (variational) parameters ν.
We want $q(w;\nu)$ to approximate the true posterior $p(w|\mathcal{D})$. Learn ν by maximising the ELBO criterion *lower bound on the marginal likelihood*:

$$\mathcal{L}(\nu) := \mathbb{E}_{q(w)}\big[\log p(y|w)\big] - \mathbb{D}_{KL}\big(q(w)||p(w)\big), \tag{1}$$

which is (often) tractable, e.g. if $q(w)$ and $p(w)$ are normal.

Problems:

- The parameter space for w is high-dimensional and the posterior multimodal.

Weight-Space Variational Inference

Variational approximation:
Let $q(w) = q(w;\nu)$ be a class of distributions with (variational) parameters ν.
We want $q(w;\nu)$ to approximate the true posterior $p(w|\mathcal{D})$. Learn ν by maximising the ELBO criterion *lower bound on the marginal likelihood*:

$$\mathcal{L}(\nu) := \mathbb{E}_{q(w)}\big[\log p(y|w)\big] - \mathbb{D}_{KL}\big(q(w)||p(w)\big), \tag{1}$$

which is (often) tractable, e.g. if $q(w)$ and $p(w)$ are normal.

Problems:

- The parameter space for w is high-dimensional and the posterior multimodal.
- Simple variational families mean very strong, unrealistic assumptions.

Weight-Space Variational Inference

Variational approximation:
Let $q(w) = q(w;\nu)$ be a class of distributions with (variational) parameters ν.
We want $q(w;\nu)$ to approximate the true posterior $p(w|\mathcal{D})$. Learn ν by maximising the ELBO criterion *lower bound on the marginal likelihood*:

$$\mathcal{L}(\nu) := \mathbb{E}_{q(w)}\big[\log p(y|w)\big] - \mathbb{D}_{KL}\big(q(w)||p(w)\big), \tag{1}$$

which is (often) tractable, e.g. if $q(w)$ and $p(w)$ are normal.

Problems:

- The parameter space for w is high-dimensional and the posterior multimodal.
- Simple variational families mean very strong, unrealistic assumptions.
  $\longrightarrow$ Do we still capture *enough of the true posterior* to justify being Bayesian? [Foong et al., 2020]
  $\longrightarrow$ Is uncertainty calibrated? [Ovadia et al., 2019]

Weight-Space Variational Inference

Variational approximation:
Let $q(w) = q(w;\nu)$ be a class of distributions with (variational) parameters ν.
We want $q(w;\nu)$ to approximate the true posterior $p(w|\mathcal{D})$. Learn ν by maximising the ELBO criterion *lower bound on the marginal likelihood*:

$$\mathcal{L}(\nu) := \mathbb{E}_{q(w)}\big[\log p(y|w)\big] - \mathbb{D}_{KL}\big(q(w)||p(w)\big), \tag{1}$$

which is (often) tractable, e.g. if $q(w)$ and $p(w)$ are normal.

Problems:

- The parameter space for w is high-dimensional and the posterior multimodal.
- Simple variational families mean very strong, unrealistic assumptions.
  $\longrightarrow$ Do we still capture *enough of the true posterior* to justify being Bayesian? [Foong et al., 2020]
  $\longrightarrow$ Is uncertainty calibrated? [Ovadia et al., 2019]
- What priors on the function space are induced by $p(w)$: how do we encode some sensible properties of functions via $p(w)$?

Generalised Variational Inference in Function Spaces

Gaussian Measures meet Bayesian Deep Learning

Veit D. Wild (Oxford), Robert Hu (Amazon), Dino Sejdinovic (Adelaide)

NeurIPS 2022, arXiv:2205.06342, github.com/MrHuff/GWI

Function-Space Variational Inference

We care about functions, not weights!

Function-Space Variational Inference

We care about functions, not weights!

Can we perform inference in the **function space directly**?
[Ma et al., 2019, Sun et al., 2019, Rudner et al., 2020, Ma and Hernández-Lobato, 2021]

Function-Space Variational Inference

We care about functions, not weights!

Can we perform inference in the **function space directly**?
[Ma et al., 2019, Sun et al., 2019, Rudner et al., 2020, Ma and Hernández-Lobato, 2021]

$$\mathcal{L} = \mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] - \mathbb{D}_{\mathsf{KL}}\big(\mathbb{Q}^F||\mathbb{P}^F\big),$$

where $\mathbb{Q}^F, \mathbb{P}^F \in \mathcal{P}(E)$ with:

- E is a (potentially infinite dimensional) separable Hilbert space of functions
- $\mathcal{P}(E)$ the space of Borel probability measures on E

Function-Space Variational Inference

We care about functions, not weights!

Can we perform inference in the **function space directly**?
[Ma et al., 2019, Sun et al., 2019, Rudner et al., 2020, Ma and Hernández-Lobato, 2021]

$$\mathcal{L} = \mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] - \mathbb{D}_{\mathsf{KL}}\big(\mathbb{Q}^F||\mathbb{P}^F\big),$$

where $\mathbb{Q}^F, \mathbb{P}^F \in \mathcal{P}(E)$ with:

- E is a (potentially infinite dimensional) separable Hilbert space of functions
- $\mathcal{P}(E)$ the space of Borel probability measures on E

A new set of challenges:

Function-Space Variational Inference

We care about functions, not weights!

Can we perform inference in the **function space directly**?
[Ma et al., 2019, Sun et al., 2019, Rudner et al., 2020, Ma and Hernández-Lobato, 2021]

$$\mathcal{L} = \mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] - \mathbb{D}_{\mathsf{KL}}\big(\mathbb{Q}^F||\mathbb{P}^F\big),$$

where $\mathbb{Q}^F, \mathbb{P}^F \in \mathcal{P}(E)$ with:

- E is a (potentially infinite dimensional) separable Hilbert space of functions
- $\mathcal{P}(E)$ the space of Borel probability measures on E

A new set of challenges:

- Prior specification on infinite dimensional function spaces?

Function-Space Variational Inference

We care about functions, not weights!

Can we perform inference in the **function space directly**?
[Ma et al., 2019, Sun et al., 2019, Rudner et al., 2020, Ma and Hernández-Lobato, 2021]

$$\mathcal{L} = \mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] - \mathbb{D}_{\mathsf{KL}}\big(\mathbb{Q}^F||\mathbb{P}^F\big),$$

where $\mathbb{Q}^F, \mathbb{P}^F \in \mathcal{P}(E)$ with:

- E is a (potentially infinite dimensional) separable Hilbert space of functions
- $\mathcal{P}(E)$ the space of Borel probability measures on E

A new set of challenges:

- Prior specification on infinite dimensional function spaces?
  $\rightarrow$ Gaussian measures on Hilbert spaces (e.g. Gaussian processes)

Function-Space Variational Inference

We care about functions, not weights!

Can we perform inference in the **function space directly**?
[Ma et al., 2019, Sun et al., 2019, Rudner et al., 2020, Ma and Hernández-Lobato, 2021]

$$\mathcal{L} = \mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] - \mathbb{D}_{\mathsf{KL}}\big(\mathbb{Q}^F||\mathbb{P}^F\big),$$

where $\mathbb{Q}^F, \mathbb{P}^F \in \mathcal{P}(E)$ with:

- E is a (potentially infinite dimensional) separable Hilbert space of functions
- $\mathcal{P}(E)$ the space of Borel probability measures on E

A new set of challenges:

- Prior specification on infinite dimensional function spaces?
  $\rightarrow$ Gaussian measures on Hilbert spaces (e.g. Gaussian processes)
- The KL-divergence is (in general) intractable in infinite dimensions and may be infinite [Burt et al., 2020].

Function-Space Variational Inference

We care about functions, not weights!

Can we perform inference in the **function space directly**?
[Ma et al., 2019, Sun et al., 2019, Rudner et al., 2020, Ma and Hernández-Lobato, 2021]

$$\mathcal{L} = \mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] - \mathbb{D}_{\mathsf{KL}}\big(\mathbb{Q}^F||\mathbb{P}^F\big),$$

where $\mathbb{Q}^F, \mathbb{P}^F \in \mathcal{P}(E)$ with:

- E is a (potentially infinite dimensional) separable Hilbert space of functions
- $\mathcal{P}(E)$ the space of Borel probability measures on E

A new set of challenges:

- Prior specification on infinite dimensional function spaces?
  $\rightarrow$ Gaussian measures on Hilbert spaces (e.g. Gaussian processes)
- The KL-divergence is (in general) intractable in infinite dimensions and may be infinite [Burt et al., 2020].
  $\rightarrow$ Is there another way?

GVI: Probabilistic Lifting + Convexification

Generalised Variational Inference [Knoblauch et al., 2022]:
Posterior approximation uses a generalised criterion

GVI: Probabilistic Lifting + Convexification

Generalised Variational Inference [Knoblauch et al., 2022]:
Posterior approximation uses a generalised criterion

$$q^*(w) := \underset{q \in \mathcal{Q}}{\operatorname{argmin}} \left\{ \mathbb{E}_{q(w)} \Big[\sum_{n=1}^{N} \ell(y_n, w) \Big] + D\big(q(w), p(w)\big) \right\}, \tag{2}$$

GVI: Probabilistic Lifting + Convexification

Generalised Variational Inference [Knoblauch et al., 2022]:
Posterior approximation uses a generalised criterion

$$q^*(w) := \underset{q \in \mathcal{Q}}{\operatorname{argmin}} \left\{ \mathbb{E}_{q(w)} \Big[\sum_{n=1}^{N} \ell(y_n, w) \Big] + D\big(q(w), p(w)\big) \right\}, \tag{2}$$

where:

GVI: Probabilistic Lifting + Convexification

Generalised Variational Inference [Knoblauch et al., 2022]:
Posterior approximation uses a generalised criterion

$$q^*(w) := \underset{q \in \mathcal{Q}}{\operatorname{argmin}} \left\{ \mathbb{E}_{q(w)} \Big[\sum_{n=1}^{N} \ell(y_n, w) \Big] + D\big(q(w), p(w)\big) \right\}, \tag{2}$$

where:

- $\mathcal{Q}$ is a set of tractable distributions

GVI: Probabilistic Lifting + Convexification

Generalised Variational Inference [Knoblauch et al., 2022]:
Posterior approximation uses a generalised criterion

$$q^*(w) := \underset{q \in \mathcal{Q}}{\operatorname{argmin}} \left\{ \mathbb{E}_{q(w)} \Big[\sum_{n=1}^{N} \ell(y_n, w) \Big] + D\big(q(w), p(w)\big) \right\}, \tag{2}$$

where:

- $\mathcal{Q}$ is a set of tractable distributions
- ℓ is a loss function (not necessarily log-likelihood)

GVI: Probabilistic Lifting + Convexification

Generalised Variational Inference [Knoblauch et al., 2022]:
Posterior approximation uses a generalised criterion

$$q^*(w) := \underset{q \in \mathcal{Q}}{\operatorname{argmin}} \left\{ \mathbb{E}_{q(w)} \Big[\sum_{n=1}^{N} \ell(y_n, w) \Big] + D\big(q(w), p(w)\big) \right\}, \tag{2}$$

where:

- $\mathcal{Q}$ is a set of tractable distributions
- ℓ is a loss function (not necessarily log-likelihood)
- D is a distance between probability measures (not necessarily KL)

Interpretation: Take any (non-convex) loss surface, and perform probabilistic lifting by averaging over q. Finally, the regularizer plays the role of convexification, making the objective in q strictly convex.

This work: GVI in Function Spaces

- Idea: Use GVI in an infinite dimensional function space: we extend results of Knoblauch et al. [2022] to infinite dimensional parameter spaces.

This work: GVI in Function Spaces

- Idea: Use GVI in an infinite dimensional function space: we extend results of Knoblauch et al. [2022] to infinite dimensional parameter spaces.
- We can target

$$\mathcal{L} := -\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] + \mathbb{D}\big(\mathbb{Q}^F, \mathbb{P}^F\big), \tag{3}$$

for inference where $\mathbb{D}$ is an appropriate distance between probability measures on the function space.

1. How to define prior $\mathbb{P}^F$?
2. What distance should we use?
3. How to parametrize variational measures $\mathbb{Q}^F$?

1. Prior: Gaussian Measures on Hilbert spaces

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space and $(H, \langle \cdot, \cdot \rangle)$ be a Hilbert space.

1. Prior: Gaussian Measures on Hilbert spaces

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space and $(H, \langle \cdot, \cdot \rangle)$ be a Hilbert space.
A random mapping $F : \Omega \to H$ is called **Gaussian random element (GRE)** iff

$$\langle F, h \rangle : \Omega \to \mathbb{R}$$

is a scalar Gaussian variable for every $h \in H$.

1. Prior: Gaussian Measures on Hilbert spaces

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space and $(H, \langle \cdot, \cdot \rangle)$ be a Hilbert space.
A random mapping $F : \Omega \to H$ is called **Gaussian random element (GRE)** iff

$$\langle F, h \rangle : \Omega \to \mathbb{R}$$

is a scalar Gaussian variable for every $h \in H$.
The mean element of F is defined as

$$m := \mathbb{E}[F] := \int F(\omega)\, d\mathbb{P}(\omega) \in H$$

and the covariance operator $C : H \to H$ of F is defined as

$$C(h) := \int \langle F(\omega), h \rangle F(\omega)\, d\mathbb{P}(\omega) - \langle m, h \rangle m,\ h \in H.$$

1. Prior: Gaussian Measures on Hilbert spaces

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space and $(H, \langle \cdot, \cdot \rangle)$ be a Hilbert space.
A random mapping $F : \Omega \to H$ is called **Gaussian random element (GRE)** iff

$$\langle F, h \rangle : \Omega \to \mathbb{R}$$

is a scalar Gaussian variable for every $h \in H$.
The mean element of F is defined as

$$m := \mathbb{E}[F] := \int F(\omega)\, d\mathbb{P}(\omega) \in H$$

and the covariance operator $C : H \to H$ of F is defined as

$$C(h) := \int \langle F(\omega), h \rangle F(\omega)\, d\mathbb{P}(\omega) - \langle m, h \rangle m,\ h \in H.$$

Write $F \sim \mathcal{N}(m, C)$ for a GRE with mean element $m \in H$ and covariance operator C. $\mathcal{N}(m, C)$ is called a **Gaussian measure** on H.

1. Prior: Gaussian Measures on Hilbert spaces

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space and $(H, \langle \cdot, \cdot \rangle)$ be a Hilbert space.
A random mapping $F : \Omega \to H$ is called **Gaussian random element (GRE)** iff

$$\langle F, h \rangle : \Omega \to \mathbb{R}$$

is a scalar Gaussian variable for every $h \in H$.
The mean element of F is defined as

$$m := \mathbb{E}[F] := \int F(\omega)\, d\mathbb{P}(\omega) \in H$$

and the covariance operator $C : H \to H$ of F is defined as

$$C(h) := \int \langle F(\omega), h \rangle F(\omega)\, d\mathbb{P}(\omega) - \langle m, h \rangle m,\ h \in H.$$

Write $F \sim \mathcal{N}(m, C)$ for a GRE with mean element $m \in H$ and covariance operator C. $\mathcal{N}(m, C)$ is called a **Gaussian measure** on H.
For arbitrary $m \in H$ and arbitrary positive, self-adjoint and trace-class C, there exists a GRE such that $F \sim \mathcal{N}(m, C)$.

2. The choice of divergence: Wasserstein-2

Recall the generalised loss:

$$\mathcal{L} := -\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] + \mathbb{D}\big(\mathbb{Q}^F, \mathbb{P}^F\big)$$

2. The choice of divergence: Wasserstein-2

Recall the generalised loss:

$$\mathcal{L} := -\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] + \mathbb{D}\big(\mathbb{Q}^F, \mathbb{P}^F\big) \tag{4}$$

2. The choice of divergence: Wasserstein-2

Recall the generalised loss:

$$\mathcal{L} := -\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] + \mathbb{D}\big(\mathbb{Q}^F, \mathbb{P}^F\big) \tag{4}$$

Gaussian Wasserstein Inference:

2. The choice of divergence: Wasserstein-2

Recall the generalised loss:

$$\mathcal{L} := -\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] + \mathbb{D}\big(\mathbb{Q}^F, \mathbb{P}^F\big) \tag{4}$$

Gaussian Wasserstein Inference:

- $E = L^2(\mathcal{X}, \rho, \mathbb{R}) := \big\{f : \mathcal{X} \to \mathbb{R} \,|\, \int |f(x)|^2 \, d\rho(x) < \infty\big\}$ with ρ input distribution on $\mathcal{X}$

2. The choice of divergence: Wasserstein-2

Recall the generalised loss:

$$\mathcal{L} := -\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] + \mathbb{D}\big(\mathbb{Q}^F, \mathbb{P}^F\big) \tag{4}$$

Gaussian Wasserstein Inference:

- $E = L^2(\mathcal{X}, \rho, \mathbb{R}) := \big\{f : \mathcal{X} \to \mathbb{R} \,|\, \int |f(x)|^2 \, d\rho(x) < \infty\big\}$ with ρ input distribution on $\mathcal{X}$
- $P := \mathbb{P}^F \sim \mathcal{N}\big(m_P, C_P\big)$

2. The choice of divergence: Wasserstein-2

Recall the generalised loss:

$$\mathcal{L} := -\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] + \mathbb{D}\big(\mathbb{Q}^F, \mathbb{P}^F\big) \tag{4}$$

Gaussian Wasserstein Inference:

- $E = L^2(\mathcal{X}, \rho, \mathbb{R}) := \big\{f : \mathcal{X} \to \mathbb{R} \,|\, \int |f(x)|^2 \, d\rho(x) < \infty\big\}$ with ρ input distribution on $\mathcal{X}$
- $P := \mathbb{P}^F \sim \mathcal{N}\big(m_P, C_P\big)$
- $Q := \mathbb{Q}^F \sim \mathcal{N}\big(m_Q, C_Q\big)$

2. The choice of divergence: Wasserstein-2

Recall the generalised loss:

$$\mathcal{L} := -\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] + \mathbb{D}\big(\mathbb{Q}^F, \mathbb{P}^F\big) \tag{4}$$

Gaussian Wasserstein Inference:

- $E = L^2(\mathcal{X}, \rho, \mathbb{R}) := \big\{f : \mathcal{X} \to \mathbb{R} \,|\, \int |f(x)|^2 \, d\rho(x) < \infty\big\}$ with ρ input distribution on $\mathcal{X}$
- $P := \mathbb{P}^F \sim \mathcal{N}\big(m_P, C_P\big)$
- $Q := \mathbb{Q}^F \sim \mathcal{N}\big(m_Q, C_Q\big)$
- $\mathbb{D}(\cdot, \cdot) = W_2(\cdot, \cdot)$ with W_2 given as Wasserstein-distance

2. The choice of divergence: Wasserstein-2

Recall the generalised loss:

$$\mathcal{L} := -\mathbb{E}_{\mathbb{Q}}\big[\log p(y|F)\big] + \mathbb{D}\big(\mathbb{Q}^F, \mathbb{P}^F\big) \tag{4}$$

Gaussian Wasserstein Inference:

- $E = L^2(\mathcal{X}, \rho, \mathbb{R}) := \big\{f : \mathcal{X} \to \mathbb{R} \,|\, \int |f(x)|^2 \, d\rho(x) < \infty\big\}$ with ρ input distribution on $\mathcal{X}$
- $P := \mathbb{P}^F \sim \mathcal{N}\big(m_P, C_P\big)$
- $Q := \mathbb{Q}^F \sim \mathcal{N}\big(m_Q, C_Q\big)$
- $\mathbb{D}(\cdot,\cdot) = W_2(\cdot,\cdot)$ with W_2 given as Wasserstein-distance

with:

$$C_P g := \int k(\cdot, x')g(x') \, d\rho(x'), \qquad C_Q g := \int r(\cdot, x')g(x') \, d\rho(x') \tag{5}$$

for all $g \in L^2(\mathcal{X}, \rho, \mathbb{R})$ where k and r are *trace-class kernels.*

2. The choice of divergence: Wasserstein-2

The Wasserstein distance between Gaussian measures on Hilbert spaces has a closed-form expression [Gelbrich, 1990]:

$$W_2^2(P,Q) = \|m_P - m_Q\|_2^2 + tr(C_P) + tr(C_Q) - 2 \cdot tr\Big[\big(C_P^{1/2} C_Q C_P^{1/2}\big)^{1/2}\Big], \quad (6)$$

where $tr(\cdot)$ denotes the trace of an operator and $C_P^{1/2}$ is the square root of the positive, self-adjoint operator C_P.

2. The choice of divergence: Wasserstein-2

Estimation of Wasserstein-2 for Gaussian measures:

$$\|m_P - m_Q\|_2^2 = \int \left(m_P(x) - m_Q(x)\right)^2 d\rho(x)$$
$$\approx \frac{1}{N} \sum_{n=1}^{N} \left(m_P(x_n) - m_Q(x_n)\right)^2$$

2. The choice of divergence: Wasserstein-2

Estimation of Wasserstein-2 for Gaussian measures:

$$\|m_P - m_Q\|_2^2 = \int \left(m_P(x) - m_Q(x)\right)^2 d\rho(x)$$
$$\approx \frac{1}{N}\sum_{n=1}^{N} \left(m_P(x_n) - m_Q(x_n)\right)^2$$

Further:

$$tr(C_P) = \int k(x,x)\, d\rho(x) \approx \frac{1}{N}\sum_{n=1}^{N} k(x_n, x_n),$$
$$tr(C_Q) = \int r(x,x)\, d\rho(x) \approx \frac{1}{N}\sum_{n=1}^{N} r(x_n, x_n).$$

2. The choice of divergence: Wasserstein-2

The last term poses some difficulties:

$$tr\left[\left(C_P^{1/2} C_Q C_P^{1/2}\right)^{1/2}\right] \approx \frac{1}{\sqrt{NN_S}} \sum_{s=1}^{N_S} \sqrt{\lambda_s\left(r(X_S, X)k(X, X_S)\right)}, \tag{7}$$

2. The choice of divergence: Wasserstein-2

The last term poses some difficulties:

$$tr\left[\left(C_P^{1/2} C_Q C_P^{1/2}\right)^{1/2}\right] \approx \frac{1}{\sqrt{NN_S}} \sum_{s=1}^{N_S} \sqrt{\lambda_s\big(r(X_S, X)k(X, X_S)\big)}, \tag{7}$$

where $X_S := (x_{S,1}, \ldots, x_{S,N_S})$, $N_S \in \mathbb{N}$ with:

$$X_{S,1}, \ldots, X_{S,N_S} \overset{\text{ind.}}{\sim} \hat{\rho} \tag{8}$$

$$r(X_S, X) := \big(r(x_{S,s}, x_n)\big)_{s,n} \tag{9}$$

$$k(X, X_S) := \big(k(x_n, x_{S,s})\big)_{n,s} \tag{10}$$

and $\lambda_s\big(r(X_S, X)k(X, X_S)\big)$ denotes the s-th eigenvalue of the matrix $r(X_S, X)k(X, X_S) \in \mathbb{R}^{N_S \times N_S}$.

The final objective

The final objective (in the case of the regression, i.e. normal likelihood):

$$\mathcal{L} = L + \widehat{W}^2 \tag{11}$$

The final objective

The final objective (in the case of the regression, i.e. normal likelihood):

$$\mathcal{L} = L + \widehat{W}^2 \tag{11}$$

with:

$$L := \frac{N}{2}\log(2\pi\sigma^2) + \sum_{n=1}^{N}\frac{\left(y_n - m_Q(x_n)\right)^2 + r(x_n, x_n)}{2\sigma^2} \tag{12}$$

$$\hat{W}^2 := \frac{1}{N}\sum_{n=1}^{N}\left(m_P(x_n) - m_Q(x_n)\right)^2 + \frac{1}{N}\sum_{n=1}^{N}k(x_n, x_n) \tag{13}$$

$$+ \frac{1}{N}\sum_{n=1}^{N}r(x_n, x_n) - \frac{2}{\sqrt{NN_S}}\sum_{s=1}^{N_S}\sqrt{\lambda_s\left(r(X_S, X)k(X, X_S)\right)}, \tag{14}$$

The final objective

The final objective (in the case of the regression, i.e. normal likelihood):

$$\mathcal{L} = L + \widehat{W}^2 \tag{11}$$

with:

$$L := \frac{N}{2}\log(2\pi\sigma^2) + \sum_{n=1}^{N} \frac{\left(y_n - m_Q(x_n)\right)^2 + r(x_n, x_n)}{2\sigma^2} \tag{12}$$

$$\hat{W}^2 := \frac{1}{N}\sum_{n=1}^{N}\left(m_P(x_n) - m_Q(x_n)\right)^2 + \frac{1}{N}\sum_{n=1}^{N} k(x_n, x_n) \tag{13}$$

$$+ \frac{1}{N}\sum_{n=1}^{N} r(x_n, x_n) - \frac{2}{\sqrt{NN_S}}\sum_{s=1}^{N_S}\sqrt{\lambda_s\left(r(X_S, X)k(X, X_S)\right)}, \tag{14}$$

3. How to parametrize the variational family?

- Stochastic Variational Gaussian processes (SVGP) [Titsias, 2009]:

$$m_Q(x) := m_P(x) + \sum_{m=1}^{M} \beta_m k(x, z_m) \tag{15}$$

$$r(x, x') := k(x, x') - k_Z(x)^T k(Z, Z)^{-1} k_Z(x) + k_Z(x)^T \Sigma k_Z(x), \tag{16}$$

where $\beta = (\beta_1, \ldots, \beta_M) \in \mathbb{R}^M$ and $\Sigma \in \mathbb{R}^{M \times M}$ are variational parameters. $Z = (Z_1, \ldots, Z_M)$ can be a data subsample or also included as variational parameters.

3. How to parametrize the variational family?

GWI-net m_Q: Use a deep neural net as the parametrization of the variational posterior mean.

3. How to parametrize the variational family?

GWI-net m_Q: Use a deep neural net as the parametrization of the variational posterior mean.

GWI-net C_Q: Use the covariance parametrization of SVGP.

In a nutshell

- Deep neural network is our model and network weights are the model parameters.

In a nutshell

- ~~Deep neural network is our model and network weights are the model parameters.~~

In a nutshell

- ~~Deep neural network is our model and network weights are the model parameters.~~

- Our model is defined directly on the function space and deep neural network weights are the variational parameters.

Toy Examples: GWI-net on 1-D data

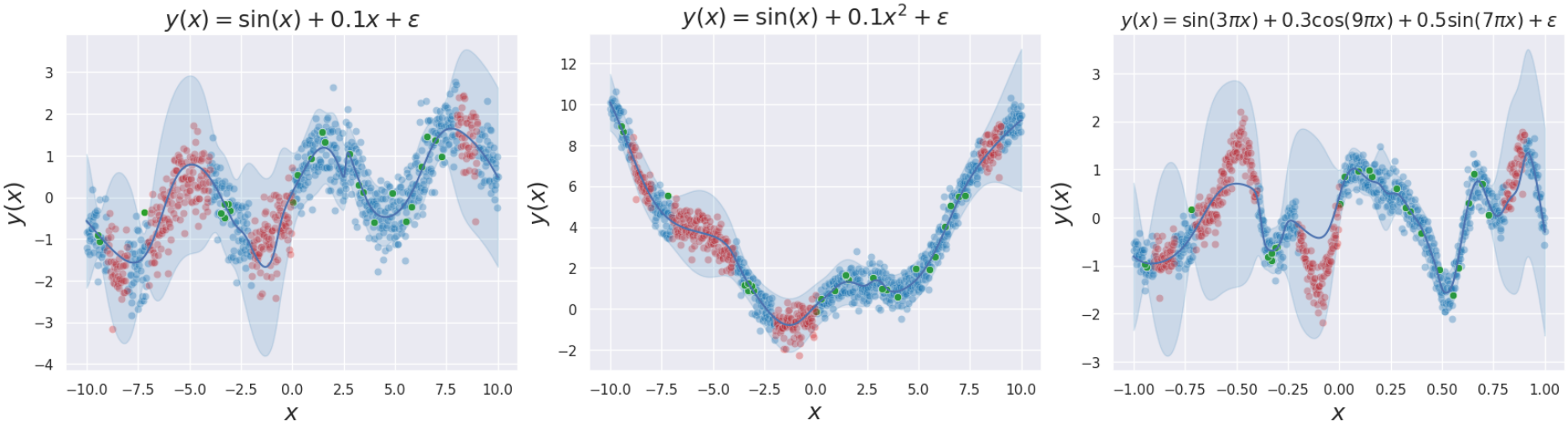


Figure: ■ : Training data ■ : Unseen data ■ : Inducing points

We use $N = 1000$ equidistant points and add white noise with $\epsilon \sim \mathcal{N}(0, 0.5^2)$. The plot shows $m_Q(x) \pm 1.96\sqrt{\mathbb{V}[Y^*(x)|Y]}$ where $\mathbb{V}[Y^*(x)|Y]$ is the posterior predictive variance given as $r(x, x) + \sigma^2$.

UCI Regression

Dataset	N	D	GWI SVGP	GWI DNN-SVGP	FVI	VIP-BNN	VIP-NP	BBB	VDO	$\alpha = 0.5$	FBNN	**EXACT GP**
BOSTON	506	13	2.8±0.31	**2.27±0.06**	2.33±0.04	2.45±0.04	2.45±0.03	2.76±0.04	2.63±0.10	2.45±0.02	2.30±0.10	2.46±0.04
CONCRETE	1030	8	3.24±0.09	**2.64±0.06**	2.88±0.06	3.02±0.02	3.13±0.02	3.28±0.01	3.23±0.01	3.06±0.03	3.09±0.01	3.05±0.02
ENERGY	768	8	1.81±0.19	0.91±0.12	0.58±0.05	**0.56±0.04**	0.60±0.03	2.17±0.02	1.13±0.02	0.95±0.09	0.68±0.02	0.54±0.02
KIN8NM	8192	8	-0.86±0.38	**-1.2±0.03**	-1.15±0.01	-1.12±0.01	-1.05±0.00	-0.81±0.01	-0.83±0.01	-0.92±0.02	N/A±0.00	N/A±0.00
POWER	9568	4	3.35±0.22	2.74±0.02	**2.69±0.00**	2.92±0.00	2.90±0.00	2.83±0.01	2.88±0.00	2.81±0.00	N/A±0.00	N/A±0.00
PROTEIN	45730	9	**2.84±0.04**	2.87±0.0	2.85±0.00	2.87±0.00	2.96±0.02	3.00±0.00	2.99±0.00	2.90±0.00	N/A±0.00	N/A±0.00
RED WINE	1588	11	0.97±0.02	**0.76±0.08**	0.97±0.06	0.97±0.02	1.20±0.04	1.01±0.02	0.97±0.02	1.01±0.02	1.04±0.01	0.26±0.03
YACHT	308	6	2.37±0.55	0.29±0.1	0.59±0.11	**-0.02±0.07**	0.59±0.13	1.11±0.04	1.22±0.18	0.79±0.11	1.03±0.03	0.10±0.05
NAVAL	11934	16	**-7.25±0.08**	-6.76±0.1	-7.21±0.06	-5.62±0.04	-4.11±0.00	-2.80±0.00	-2.80±0.00	-2.97±0.14	-7.13±0.02	N/A±0.00
Mean Rank			5.5	**2.06**	2.22	3.33	4.94	7	6.11	4.83		

Table: The table shows the average test NLL on several UCI regression datasets. We train on random 90% of the data and predict on 10%. This is repeated 10 times and we report mean and standard deviation. The results for our competitors are taken from Ma and Hernández-Lobato [2021].

Classification

	FMNIST			CIFAR 10		
Model	Accuracy	NLL	OOD-AUC	Accuracy	NLL	OOD-AUC
GWI-net	**93.25 ±0.09**	**0.250 ±0.00**	**0.959 ±0.01**	**83.82 ±0.00**	**0.553 ±0.00**	0.618 ±0.00
FVI	91.60±0.14	0.254±0.05	0.956±0.06	77.69 ±0.64	0.675±0.03	0.883±0.04
MFVI	91.20±0.10	0.343±0.01	0.782±0.02	76.40±0.52	1.372±0.02	0.589±0.01
MAP	91.39±0.11	0.258±0.00	0.864±0.00	77.41±0.06	0.690±0.00	0.809±0.01
KFAC-LAPLACE	84.42±0.12	0.942±0.01	0.945±0.00	72.49±0.20	1.274±0.01	0.548±0.01
RITTER et al.	91.20±0.07	0.265±0.00	0.947±0.00	77.38±0.06	0.661±0.00	0.796±0.00

Table: We report average accuracy, NLL and OOD-AUC on test data for 10 different train/test splits.

Summary

- Deep Neural Networks are good prediction models. Let's make them Bayesian.

Summary

- ~~Deep Neural Networks are good prediction models. Let's make them Bayesian.~~

Summary

- ~~Deep Neural Networks are good prediction models. Let's make them Bayesian.~~

- Deep Neural Networks are a good parametrization of the variational posterior for function space models.

Summary

- ~~Deep Neural Networks are good prediction models. Let's make them Bayesian.~~

- Deep Neural Networks are a good parametrization of the variational posterior for function space models.

- As a practical viewpoint: a different (generalized variational) objective for training your favourite deep neural net, which has extra parameters to quantify uncertainty.

A Rigorous Link between Deep Ensembles and (Variational) Bayesian Methods

Veit D. Wild (Oxford), Sahra Ghalebikesabi (Oxford),
Dino Sejdinovic (Adelaide), Jeremias Knoblauch (UCL)

NeurIPS 2023, arXiv:2305.15027, github.com/sghalebikesabi/GVI-WGF

GVI: Probabilistic Lifting + Convexification

Generalised Variational Inference [Knoblauch et al., 2022]:
Posterior approximation uses a generalised criterion

$$Q^*(\theta) := \operatorname*{argmin}_{Q \in \mathcal{Q}} \underbrace{\left\{ \mathbb{E}_{Q(\theta)}\Big[\sum_{n=1}^{N} \ell(y_n, \theta) \Big] + D\big(Q(\theta), P(\theta)\big) \right\}}_{L(Q)},$$

where:

- $\mathcal{Q}$ is a set of tractable distributions
- ℓ is a loss function (not necessarily log-likelihood)
- D is a distance between probability measures (not necessarily KL)

Interpretation: Take any (non-convex) loss surface, and perform probabilistic lifting by averaging over q. Finally, the regularizer plays the role of convexification, making the objective in q strictly convex.

GVI: Probabilistic Lifting + Convexification

Generalised Variational Inference [Knoblauch et al., 2022]:
Posterior approximation uses a generalised criterion

$$Q^*(\theta) := \operatorname*{argmin}_{Q \in \mathcal{Q}} \underbrace{\left\{ \mathbb{E}_{Q(\theta)}\Big[\sum_{n=1}^{N} \ell(y_n, \theta) \Big] + D\big(Q(\theta), P(\theta)\big) \right\}}_{L(Q)},$$

where:

- $\mathcal{Q}$ is a set of ~~tractable~~ all distributions
- ℓ is a loss function (not necessarily log-likelihood)
- D is a distance between probability measures (not necessarily KL)

Interpretation: Take any (non-convex) loss surface, and perform probabilistic lifting by averaging over q. Finally, the regularizer plays the role of convexification, making the objective in q strictly convex.

Relaxing the variational family assumption?

Idea: formulate a gradient flow in the space of probability measures [Ambrosio et al., 2005] on the (generalized) variational objective $L(Q)$.

Parameter space

- Initialise: $\theta_0 \in \mathbb{R}^J$
- Gradient step:

$$\theta_{k+1} = \arg\min_{\theta \in \mathbb{R}^J} \left\{ \ell(\theta) + \frac{1}{2\eta} \|\theta - \theta_k\|_2^2 \right\}.$$

Probability space

- Initialise: $Q_0 \in \mathcal{P}_2(\mathbb{R}^J)$
- Gradient step:

$$Q_{k+1} = \arg\min_{Q \in \mathcal{P}_2(\mathbb{R}^J)} \left\{ L(Q) + \frac{1}{2\eta} W_2(Q, Q_k)^2 \right\}$$

with 2-Wasserstein metric

$$W_2(P, Q)^2 = \inf \left\{ \int ||\theta - \theta'||_2^2 \, d\pi(\theta, \theta') : \pi \in \mathcal{C}(P, Q) \right\}.$$

A general form of objective

$$L(Q) := \int V(\theta)\, dQ(\theta) + \frac{\lambda_1}{2} \iint \kappa(\theta, \theta')\, dQ(\theta) dQ(\theta') + \lambda_2 \int \log q(\theta) q(\theta)\, d\theta,$$

The overall energy of a collection of particles sampled from Q is decomposed into three parts:

- the external potential $V(\theta)$ which acts on each particle individually
- the interaction energy defined via kernel $\kappa(\theta, \theta')$ describing pairwise interactions between particles,
- the overall entropy of the system.

This is precisely the GVI objective with regularizer that is a mixture of KL and MMD:

$$D(Q, P) = \lambda_1 MMD^2(Q, P) + \lambda_2 KL(Q, P)$$

Implementing the Wasserstein Gradient Flow

Interacting particles scheme:

- **Step 1:** Sample $N_E \in \mathbb{N}$ particles $\theta_1(0), \ldots, \theta_{N_E}(0)$ independently from $Q_0 \in \mathcal{P}_2(\mathbb{R}^J)$.
- **Step 2:** Evolve the particle θ_n by following the stochastic differential equation (SDE)

$$d\theta_n(t) = -\Big(\nabla V\big(\theta_n(t)\big) + \frac{\lambda_1}{N_E}\sum_{j=1}^{N_E}(\nabla_1\kappa)\big(\theta_n(t), \theta_j(t)\big)\Big)dt + \sqrt{2\lambda_2}dB_n(t),$$

  for $n = 1, \ldots, N_E$, and $\{B_n(t)\}_{t>0}$ independent Brownian motions.

Cases:

- No regularizer, i.e. $\lambda_1 = \lambda_2 = 0$: deep ensemble [Lakshminarayanan et al., 2017], No convergence to the global optimum.
- Only KL regularizer, i.e. $\lambda_1 = 0$: deep Langevin ensemble (essentially Lakshminarayanan et al. [2017]+Welling and Teh [2011]), Converges to the global optimum.
- KL+MMD regularizer: deep repulsive Langevin ensemble (**new**), Converges to the global optimum

Number of particles N_E when many local minima are present

A uniform prior P and initialisation Q_0 and the loss $\ell(\theta) := -|\sin(\theta)|$, $\theta \in [-1000\pi, 1000\pi]$, which has 2000 local minima.

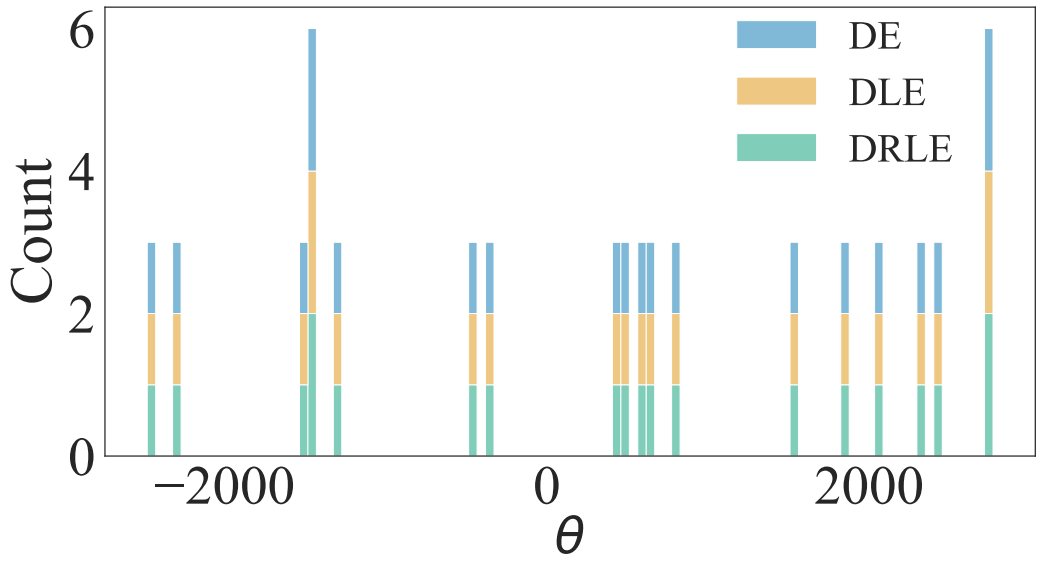


Figure: We generate $N_E = 20$ samples from the three ensemble methods. The x-axis shows the location of the particles after training. Since the same initialisation $\theta_n(0)$ is chosen for all methods, we observe that particles fall into the same local modes. Further, 16/20 particles are alone in their respective local modes and the location of the particles varies very little between the different methods.

Summary

- Gradient flows for GVI allow to unify a collection of existing algorithms under a common conceptual roof and plant the seeds for new deep ensemble algorithms.

Summary

- Gradient flows for GVI allow to unify a collection of existing algorithms under a common conceptual roof and plant the seeds for new deep ensemble algorithms.
- Performance difference between simple deep ensembles and more intricate schemes may not be numerically discernible for loss landscapes with many local minima.

References I

Veit D. Wild, Robert Hu, and Dino Sejdinovic. Generalized Variational Inference in Function Spaces: Gaussian Measures meet Bayesian Deep Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Veit D. Wild, Sahra Ghalebikesabi, Dino Sejdinovic, and Jeremias Knoblauch. A Rigorous Link between Deep Ensembles and (Variational) Bayesian Methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Radford M Neal. *Bayesian learning for neural networks*. Springer, 2012.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic Gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, pages 681–688. Citeseer, 2011.

Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24, 2011.

References II

Andrew Foong, David Burt, Yingzhen Li, and Richard Turner. On the expressiveness of approximate inference in Bayesian neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:15897–15908, 2020.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Chao Ma, Yingzhen Li, and José Miguel Hernández-Lobato. Variational implicit processes. In *International Conference on Machine Learning*, pages 4222–4233. PMLR, 2019.

Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational bayesian neural networks. *arXiv preprint arXiv:1903.05779*, 2019.

Tim GJ Rudner, Zonghao Chen, and Yarin Gal. Rethinking function-space variational inference in bayesian neural networks. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2020.

References III

Chao Ma and José Miguel Hernández-Lobato. Functional variational inference based on stochastic process generators. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

David R Burt, Sebastian W Ober, Adrià Garriga-Alonso, and Mark van der Wilk. Understanding variational inference in function-space. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2020.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. An Optimization-centric View on Bayes' Rule: Reviewing and Generalizing Variational Inference. *Journal of Machine Learning Research*, 23(132):1–109, 2022.

Matthias Gelbrich. On a formula for the L2 Wasserstein metric between measures on Euclidean and Hilbert spaces. *Mathematische Nachrichten*, 147(1):185–203, 1990.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics (AISTATS)*, pages 567–574, 2009.

References IV

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer, 2005.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

Losses

For regression, $y \in \mathbb{R}$,

$$p(\mathcal{D}|F) = \prod_{n=1}^{N} p(y_n|F(x_n)) = \prod_{n=1}^{N} \mathcal{N}(y_n \,|\, F(x_n), \sigma^2), \tag{17}$$

where $\sigma^2 > 0$.
For classification, $y \in \{-1, +1\}$,

$$p(\mathcal{D}|F) = \prod_{n=1}^{N} p(y_n|F(x_n)) = \prod_{n=1}^{N} \sigma(y_n F(X_n)), \tag{18}$$

where $\sigma(t) = 1/(1 + e^{-t})$

Computational complexity

- $\mathcal{L}$ is tractable for any m_P, m_Q, k and r

Computational complexity

- $\mathcal{L}$ is tractable for any m_P, m_Q, k and r
- One evaluation of $\mathcal{L}$ requires:

Computational complexity

- $\mathcal{L}$ is tractable for any m_P, m_Q, k and r
- One evaluation of $\mathcal{L}$ requires:
  - N evaluations of m_Q and m_P

Computational complexity

- $\mathcal{L}$ is tractable for any m_P, m_Q, k and r
- One evaluation of $\mathcal{L}$ requires:
  - N evaluations of m_Q and m_P
  - $N_S \cdot N$ evaluations of r and k

Computational complexity

- $\mathcal{L}$ is tractable for any m_P, m_Q, k and r
- One evaluation of $\mathcal{L}$ requires:
  - N evaluations of m_Q and m_P
  - $N_S \cdot N$ evaluations of r and k
  - $\mathcal{O}(N + N_S^2 N + N_S^3)$operations for the eigenvalue problem

Computational complexity

- $\mathcal{L}$ is tractable for any m_P, m_Q, k and r
- One evaluation of $\mathcal{L}$ requires:
  - N evaluations of m_Q and m_P
  - $N_S \cdot N$ evaluations of r and k
  - $\mathcal{O}(N + N_S^2 N + N_S^3)$operations for the eigenvalue problem
- One evaluation of $\mathcal{L}$ in batch-mode requires:

Computational complexity

- $\mathcal{L}$ is tractable for any m_P, m_Q, k and r
- One evaluation of $\mathcal{L}$ requires:
  - N evaluations of m_Q and m_P
  - $N_S \cdot N$ evaluations of r and k
  - $\mathcal{O}(N + N_S^2 N + N_S^3)$operations for the eigenvalue problem
- One evaluation of $\mathcal{L}$ in batch-mode requires:
  - N_B evaluations of m_Q and m_P

Computational complexity

- $\mathcal{L}$ is tractable for any m_P, m_Q, k and r
- One evaluation of $\mathcal{L}$ requires:
  - N evaluations of m_Q and m_P
  - $N_S \cdot N$ evaluations of r and k
  - $\mathcal{O}(N + N_S^2 N + N_S^3)$operations for the eigenvalue problem
- One evaluation of $\mathcal{L}$ in batch-mode requires:
  - N_B evaluations of m_Q and m_P
  - $N_S \cdot N_B$ evaluations of r and k

Computational complexity

- $\mathcal{L}$ is tractable for any m_P, m_Q, k and r
- One evaluation of $\mathcal{L}$ requires:
  - N evaluations of m_Q and m_P
  - $N_S \cdot N$ evaluations of r and k
  - $\mathcal{O}(N + N_S^2 N + N_S^3)$operations for the eigenvalue problem
- One evaluation of $\mathcal{L}$ in batch-mode requires:
  - N_B evaluations of m_Q and m_P
  - $N_S \cdot N_B$ evaluations of r and k
  - $\mathcal{O}(N_B + N_S^2 N_B + N_S^3)$ operations for the eigenvalue problem

Computational complexity

- $\mathcal{L}$ is tractable for any m_P, m_Q, k and r
- One evaluation of $\mathcal{L}$ requires:
  - N evaluations of m_Q and m_P
  - $N_S \cdot N$ evaluations of r and k
  - $\mathcal{O}(N + N_S^2 N + N_S^3)$operations for the eigenvalue problem
- One evaluation of $\mathcal{L}$ in batch-mode requires:
  - N_B evaluations of m_Q and m_P
  - $N_S \cdot N_B$ evaluations of r and k
  - $\mathcal{O}(N_B + N_S^2 N_B + N_S^3)$ operations for the eigenvalue problem

$\longrightarrow$ typically $N_S, N_B << N$, e.g. $N_S = N_B = 100$