Function-Space Particle Optimization for Bayesian Neural Networks

Ziyu Wang, Tongzheng Ren, Jun Zhu and Bo Zhang

Tsinghua University. {wzy196,rtz19970824}@gmail.com; {dcszj,dcszb}@tsinghua.edu.cn

Background: Particle-Optimization VI

POVI approximate posterior distributions with particles: $q(\theta) = \frac{1}{n}\sum_{j=1}^{n}\delta(\theta-\theta^{(j)})$. Particles are updated iteratively with the following update rule:

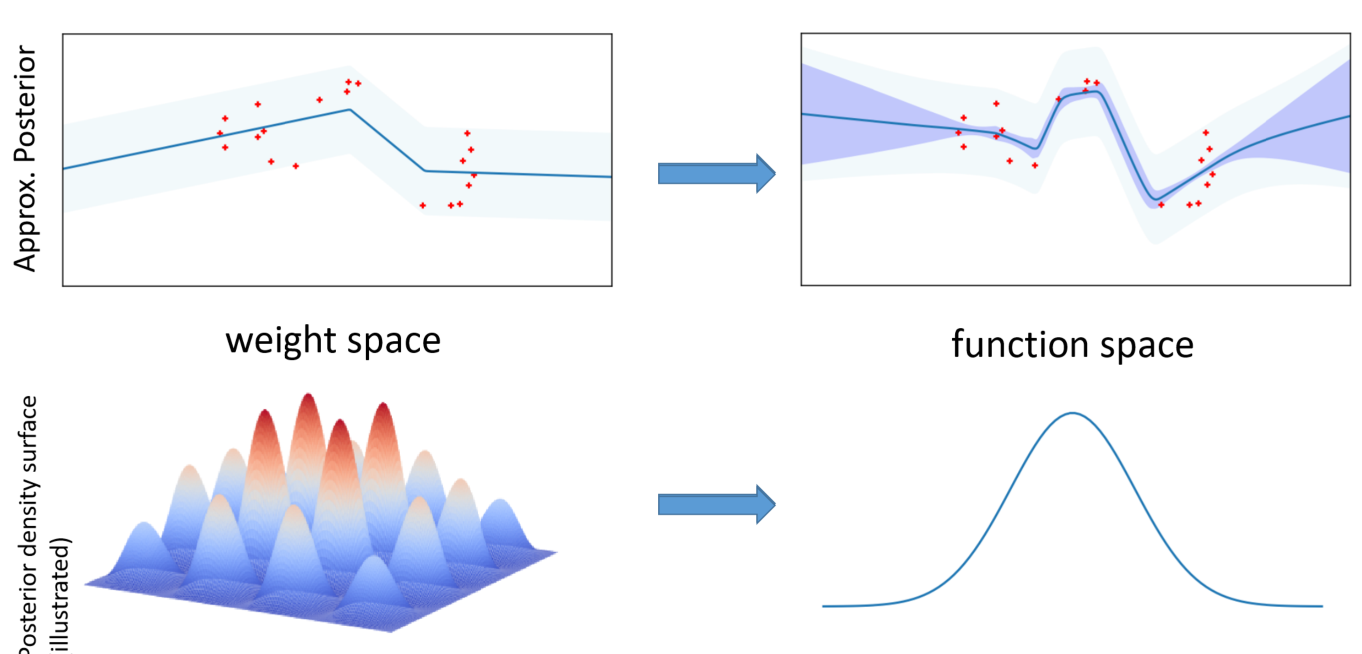
$$\theta_{t+1}^{(j)} \leftarrow \theta_t^{(j)} + \epsilon \mathbf{v}_t(\theta_t^{(j)})$$

$$\text{(for SVGD)} = \theta_t^{(j)} + \epsilon\left[\underbrace{k_{ij}\sum_j \nabla_{\theta^{(j)}}\log p(\theta^{(j)}|\mathbf{X},\mathbf{Y})}_{\text{smoothed gradient}} + \underbrace{\sum_j \nabla_{\theta^{(j)}}k_{ij}}_{\text{repulsive force}}\right].$$

POVI is easy to implement and scalable. As $n\to\infty$, the variational family has "unlimited" flexibility.

POVI for BNN?

- BNNs are **over-paramterized**: on the posterior surface, multiple modes correspond to the same prediction function.
- With *finite* particles, POVI algorithms could place all particles on these modes, corresponding to a variational posterior with no epistemic uncertainty at all.
- Our proposal: **Work in function space.**



Function-Space POVI

- Approximate *function-space posterior* with (weight) particles.
- In each iteration, sample a finite set $\mathbf{x}$ and try to match $q(f(\mathbf{x}))$ and $p(f(\mathbf{x})|\mathbf{X},\mathbf{Y})$:
  - A finite-dim problem; POVI defines an update direction $\nu[f(\mathbf{x})]$.
  - So we update $\theta^{(i)}$ so $f(\mathbf{x};\theta_{t+1}^{(i)})$ becomes closer to $f(\mathbf{x};\theta_t^{(i)})+\epsilon\nu_t^{(i)}$.

Algorithm 1 f-POVI, using SVGD as the POVI implementation

1: **for** iteration ℓ **do**

2: Sample a mini-batch $\mathbf{x}_b,\mathbf{y}_b$ from the training set, and $\tilde{x}_{1\dots B_2}\overset{\text{i.i.d.}}{\sim}\nu$. Denote $\mathbf{x}=\mathbf{x}_b\cup\{\tilde{x}_i\}$.

3: **for** particle i **do**

4: Calculate

$$\hat{v}_\ell^{(i)} = \sum_j\left[\underbrace{k_{ij}\nabla_{f_\ell^{(j)}}\left(\frac{N}{B}\log p(\mathbf{y}_b|f_\ell^{(j)}(\mathbf{x}_b)) + \log p(f_\ell^{(j)}(\mathbf{x}))\right)}_{\text{MAP-like loss gradient}} + \underbrace{\nabla_{f_\ell^{(j)}}k_{ij}}_{\text{function space repulsive force}}\right]$$

where $k_{ij}=\mathbf{k}(f_\ell^{(i)}(\mathbf{x}),f_\ell^{(j)}(\mathbf{x}))$.

5: (Single-step gradient descent:) Set

$$\theta_{\ell+1}^{(i)} \leftarrow \theta_\ell^{(i)} + \epsilon_\ell\left(\frac{\partial f(\mathbf{x})}{\partial\theta}\right)^\top \hat{v}_\ell^{(i)}.$$

6: **end for**

7: **end for**

Compare with ...

- Ensemble training: f-POVI adds a repulsive force term, but is otherwise equally easy to implement.
- Weight-space POVI: the repulsive force works in function space, thus is more efficient given finite particles.

Experiments

- Supervised learning: Outperforms strong baselines on UCI and MNIST; **Scales to complex architectures** such as ResNet.
- **Improved adversarial robustness:** on MNIST and CIFAR-10.
- **Improved exploration in RL** on contextual bandit tasks, following the setup in (Riquelme et al, 2018).

Method	BBB (Gaussian)	BBB (Scale Mixture)	KIVI	f-SVGD
Test Error	1.82%	1.36%	1.29%	**1.21%**

Table 1: Test error on the MNIST dataset.

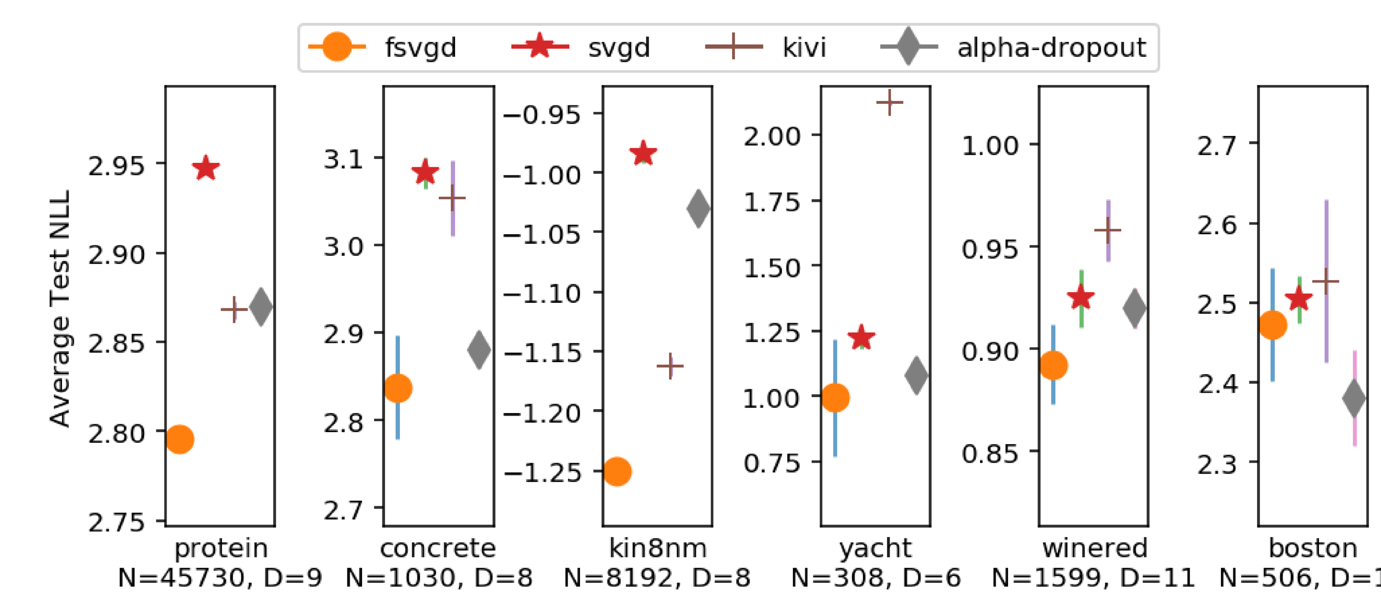


Figure 1: Log likelihood on UCI regression datasets.

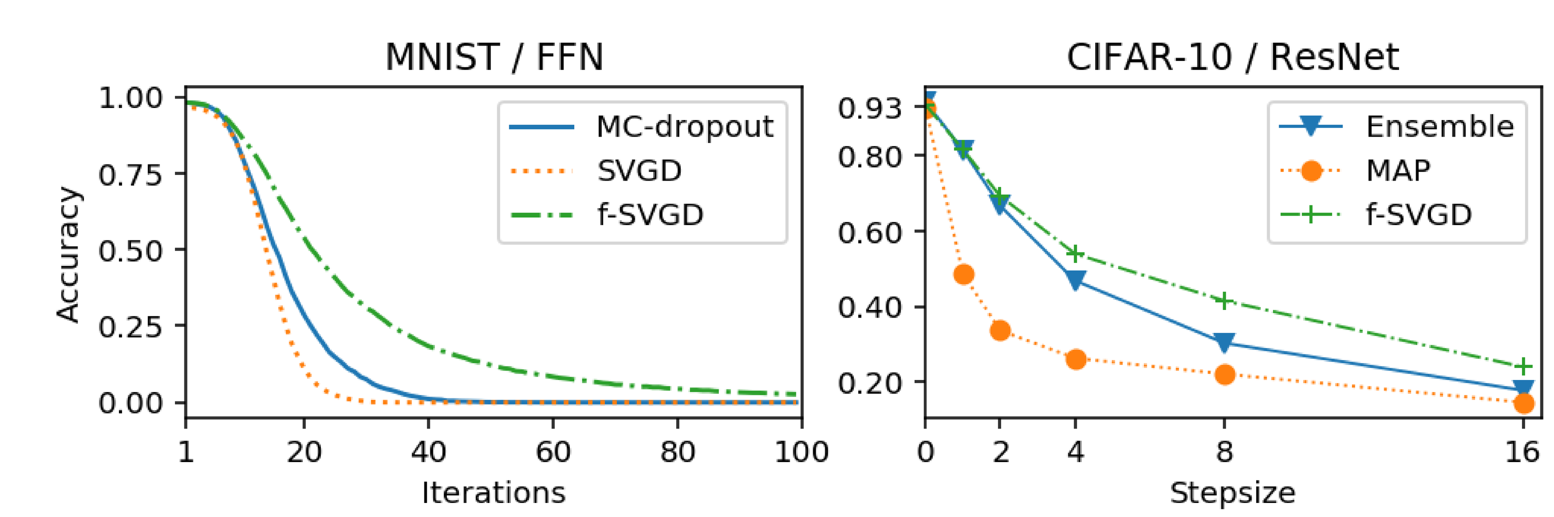


Figure 2: Accuracy on adversarial examples. Left: iterated FGSM; right: FGSM.

The Theory (in extended arXiv ver.)

- POVI as WGF: "Asymptotically"[1], q evolves under a WGF $\frac{\partial q}{\partial t} = -\nabla\cdot(q\mathbf{v}) = -\nabla\cdot\left(q\nabla\frac{\delta\mathcal{E}}{\delta q}\right)$, which minimizes an energy $\mathcal{E}[q_t]$ (e.g. KL).
- **f-POVI as Wasserstein gradient flow:** f-POVI correspond to a WGF, whose minimizer (in a full-batch setting) correspond to posterior approximations with consistent marginals.
- In f-POVI, the parametric approximation defines a metric in function space corresponding to the *neural tangent kernel*, assuming it is constant[2]. In this case the algorithm can be seen as a WGF *in function space*.

Proposition (WGF). The simplified[3] version of our algorithm

$$\frac{\partial\rho_\theta(\theta,t)}{\partial t} = -\nabla\cdot\rho_\theta\,\mathbb{E}_\mu\left[\left(\frac{\partial f(x)}{\partial\theta}\right)^\top \mathbf{v}_x\right]$$

is the Wasserstein gradient flow in $W_2(\Theta)$ minimizing the following energy:

$$\mathcal{E}[q] = -\int\left(\log p(\mathbf{Y}|\mathbf{X},f) + \int\log\frac{p(f(x))}{q(f(x))}d\mu(x)\right)dq(f).$$

Proposition (marginal consistency). When $\mathbf{X}\subseteq\mathbf{x}$ a.s., minimizer of $\mathcal{E}$, q, satisfies $q(f(\mathbf{x}))=p(f(\mathbf{x})|\mathbf{X},\mathbf{Y})$ a.s.

	BBB	GP	Bootstrap	f-SVGD
Mushroom	19.15 ± 5.98	16.75 ± 1.63	$\mathbf{2.71\pm0.22}$	4.39 ± 0.39
Wheel	55.77 ± 8.29	60.80 ± 4.40	42.16 ± 7.80	$\mathbf{7.54\pm0.41}$

Table 2: Cumulative regret in different contextual bandit tasks.

[1]: *Heuristically speaking*, as $n\to\infty,\epsilon\to0$; POVI connects to WGFs on a.c. distributions. [2]: This is a stronger assumption than the theorem in the NTK paper. [3]: This connects to, e.g., W-SGLD-B without the blob approximation.

← Code

arXiv →