Partially Stochastic Infinitely Deep Bayesian Neural Networks

Sergio Calvo Ordoñez[12] Matthieu Meunier[2] Francesco Piatti[3] Yuantao Shi[12]

[1]Oxford-Man Institute of Quantitative Finance, University of Oxford, [2]Mathematical Institute, University of Oxford,
[3]Department of Mathematics, Imperial College London

Background: Neural Differential Equations

A Neural Differential Equation is a differential equation using a neural network to parametrise the vector field.

$$\frac{dh_t}{dt} = f_\theta(t, h_t),\ h_0 = x,$$

where $f_\theta : \mathbb{R} \times \mathbb{R}^{d_1 \times ... \times d_k}$ and $y : [0, T] \to \mathbb{R}^{d_1 \times ... \times d_k}$ is the solution. Similarly, discrete-depth residual networks can be defined as:

$$h_{t+\delta} = h_t + \delta f(h_t; w_t), \quad t = 1, ..., T,$$

In the limit where $\delta = \frac{1}{T}$ and $T \to \infty$, we obtain a Neural ODE.

Background: (Partially Stochastic) Bayesian NNs

1. Place a prior on the weights of the network, e.g. $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \eta\mathbf{I})$.
2. Define an observation model, e.g. $p(y|\mathbf{x}, \theta) = \mathcal{N}(y; f_\theta(\mathbf{x}), \sigma^2)$.
3. Apply Bayes' Rule:

$$p(\theta|\mathcal{D}) \propto p(\theta) \prod_{i=1}^{N} p(y^{(i)}|\mathbf{x}^{(i)}, \theta)$$

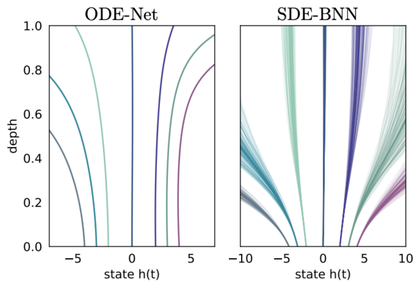
Partial Stochasticity: $\Theta = \Theta_S \cup \Theta_D$

Background: SDEs as Approximate Posteriors

A d-dimensional SDE driven by a m-dimensional Brownian motion on an interval $[0, T]$:

$$dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dB_t,$$

where $\mu : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ (drift), $\sigma : [0, T] \times \mathbb{R}^d \to \mathbb{R}^{d \times m}$ (diffusion) are Lipschitz functions in the state, and satisfy a linear growth condition.



Infinitely Deep BNNs: $\begin{bmatrix} dw_t \\ dh_t \end{bmatrix} = \begin{bmatrix} f_w(w_t, t, \phi) \\ f_h(h_t, t, w_t) \end{bmatrix} dt + \begin{bmatrix} g_w(w_t, t) \\ 0 \end{bmatrix} dB_t$

Methodology

Vertical Separation of the Weights

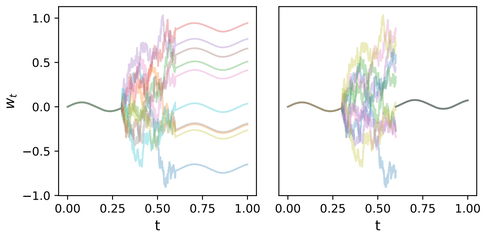
Let $w_t \in \mathbb{R}^{d_w}$ be the weights of layer $t \in [0, 1]$, and $t_1 < t_2$ be cutoff points:

- For $t \notin (t_1, t_2)$, the dynamics of w_t are deterministic:

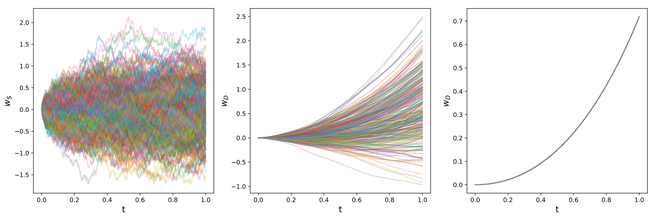
$$dw_t = f_q(t, w_t; \theta)dt.$$

- For $t \in (t_1, t_2)$, w_t is a random vector and we perform Bayesian inference. We define a prior as the solution of the SDE and an approximate posterior::

$$dw_t = f_p(t, w_t)dt + g_p(t, w_t)dB_t, \quad dw_t = f_q(t, w_t; \theta)dt + g_p(t, w_t)dB_t.$$



Horizontal Separation of the Weights



The previous system can be described as a single SDE:

$$\begin{bmatrix} dh_t \\ dw_t \end{bmatrix} = \begin{bmatrix} f_h(t, h_t; w_t) \\ f_q(t, w_t; \theta) \end{bmatrix} dt + \begin{bmatrix} 0 \\ g_p(t, w_t) \end{bmatrix} dB_t. \tag{1}$$

For the horizontal case, we perform the weight separation in the same layer by setting the diffusion matrix to zero, e.g.:

$$g_p(t, w_t) = \begin{bmatrix} \sigma\mathbf{I}_{m_1} & \mathbf{0}_{m_1 \times m_2} \\ \mathbf{0}_{m_2 \times m_1} & \mathbf{0}_{m_2} \end{bmatrix}.$$

Then, we have an extra network (same function different θ):

$$\begin{bmatrix} dh_t \\ dw_t^D \\ dw_t^S \end{bmatrix} = \begin{bmatrix} f_h(t, h_t; w_t^S, w_t^D) \\ f_{\theta_D}(t, w_t^D; \theta_D) \\ f_{\theta_S}(t, w_t^S; \theta_S) \end{bmatrix} dt + \begin{bmatrix} 0 \\ 0 \\ g_p(t, w_t^S) \end{bmatrix} dB_t. \tag{2}$$

Training the Network

$$\mathcal{L}^V(\Theta) = \mathbb{E}_{q_\theta}\left[\log p(D|w) - \kappa \int_{t_1}^{t_2} ||u_\theta(t, w_t)||^2 dt\right],$$

where $u_\theta(t, w_t) = g_p(t, w_t)^{-1}\left[f_p(t, w_t) - f_q(t, w_t; \Theta)\right]$.

$$\mathcal{L}^H(\Theta) = \mathbb{E}_{q_\theta}\left[\log p(D|w) - \kappa \int_{t_1}^{t_2} ||u_\theta^S(t, w_t^S)||^2 dt\right],$$

with $u_\theta^S(t, w_t^S) = g_p(t, w_t^S)^{-1}\left[f_p(t, w_t^S) - f_\theta^S(t, w_t^S; \Theta_S)\right]$.

As for the prior, we use an Ornstein-Uhlenbeck process:

$$f_p(w_t, t) = -w_t, \quad g(w_t, t) = \sigma\mathbf{I}_d.$$

Theoretical Results

Theorem. *Let X, Y be random variables taking values in $\mathcal{X} \subset \mathbb{R}^{d_x}, \mathcal{Y} \subset \mathbb{R}^{d_y}$ respectively. Assume $\mathcal{X}$ is compact. Assume further that there exists a continuous generator function $\tilde{f} : \mathbb{R}^m \times \mathcal{X} \to \mathcal{Y}$ for the conditional distribution $Y|X$ for some $m \geq d_y$. Consider a PSDE-BNN model given by (1), with d_h, d_w large enough. Then, for all $\varepsilon > 0$, there exist $A \in \mathcal{F}, h_0, f_h, f_q, g_p, \theta$, a linear operator $\mathcal{L}$ and a random variable $\eta \sim \mathcal{N}(0, I_m)$ independent of X such that*

$$(i)\ \mathbb{P}(A) \geq 1 - \varepsilon,$$
$$(ii)\ \forall \omega \in A, x \in \mathcal{X},\ ||\mathcal{L}(\psi_1(\omega, x)) - \tilde{f}(\eta(\omega), x)|| \leq \varepsilon.$$

I.e. there exists a PSDE-BNN that approximates $Y|X$ with high probability.

Experimental Results

Model	MNIST Accuracy (%) ↑	MNIST ECE ($\times 10^{-2}$) ↓	CIFAR-10 Accuracy (%) ↑	CIFAR-10 ECE ($\times 10^{-2}$) ↓
ResNet32†	**99.46 ± 0.00**	2.88 ±0.94	87.35 ±0.00	**8.47 ± 0.39**
ODENet†	98.90 ±0.04	**1.11 ± 0.10**	**88.30 ± 0.29**	8.71 ±0.21
MFVI ResNet 32†	**99.40** ± 0.00	2.76 ± 1.28	86.97 ± 0.00	3.04 ± 0.94
MFVI†	—	—	86.48	**1.95**
ResNet32 + LL Laplace	—	—	**92.10**	2.98
Deep Ensemble†	—	—	89.22	2.79
HMC ("gold standard")†	98.31	**1.79**	90.70	5.94
MFVI ODENet†	98.81 ±0.00	2.63 ±0.31	81.59 ±0.01	3.62 ±0.40
MFVI HyperODENet†	98.77 ±0.01	2.82 ±1.34	80.62 ±0.00	4.29 ±1.10
SDE-BNN	98.55 ±0.09	0.63 ±0.10	86.04 ±0.25	9.13 ±0.28
PSDE-BNN - ODEFirst (*ours*)	**99.30 ±0.06**	0.62 ±0.08	**87.84 ± 0.08**	4.73 ±0.07
PSDE-BNN - SDEFirst (*ours*)	99.10 ±0.07	**0.56 ± 0.10**	85.34 ±0.21	**3.56 ± 0.15**
PSDE-BNN - `fix_w2` (*ours*)	99.30 ±0.07	0.60 ± 0.10	87.49 ±0.24	5.27 ± 0.17
PSDE-BNN - Hor. Cut (*ours*)	99.27 ± 0.03	0.57 ± 0.06	87.78 ± 0.37	4.29 ± 0.37
SDE-BNN (*same training time*)	94.23 ± 0.02	**0.14 ± 0.02**	69.94 ± 0.21	**1.52 ± 0.23**

Accuracy of PSDE-BNN vs. other Bayesian models

Model	Epoch Time	Inference Time
SDE-BNN	371.9	43.7
ODEFirst (10%)	**289.4**	**31.7**
SDEFirst (10%)	295.1	**31.7**
Hor. Cut (50%)	316.7	35.9

Training and Inference times

Model	Corruption Lvl 2	Corruption Lvl 5
ODEFirst	0.322 ± 0.409	0.503 ± 0.465
SDEFirst	**0.443 ± 0.468**	**0.628 ± 0.512**
Hor. Cut	0.394 ± 0.436	0.541 ± 0.458
SDEBNN	0.256 ± 0.353	0.375 ± 0.405

Predictive Entropy on CIFAR-10-C

[1] Xu, W., Chen, R. T., Li, X., and Duvenaud, D. Infinitely deep bayesian neural networks with stochastic differential equations.
[2] Sharma, M., Farquhar, S., Nalisnick, E., and Rainforth, T. Do bayesian neural networks need to be fully stochastic?
[3] Dupont, E., Doucet, A., and Teh, Y. W. Augmented neural odes.