Density Regression and Uncertainty Quantification with Bayesian Deep Noise Neural Networks: Supplementary Materials

Daiwei Zhang[1], Tianci Liu[2] and Jian Kang[3*]

[1]Department of Biostatistics, Epidemiology and Informatics, University of Pennsylvania

[2]School of Electrical and Computer Engineering, Purdue University

[3]Department of Biostatistics, University of Michigan

S1 METHOD DETAILS

S1.1 Details of Theoretical Results

S1.1.1 Details of Theorem 1

We define the variables in the posterior full-conditional distributions in Theorem 1.

Let $\phi_{\mathcal{N}}(\cdot \mid \mu, \sigma^2)$ and $\Phi_{\mathcal{N}}(\cdot \mid \mu, \sigma^2)$ be the PDF and CDF, respectively, of the normal distribution with mean μ and variance σ^2.

For $n \in 1, \ldots, N$, $l \in 0, \ldots, L$, and $k \in 1, \ldots, K_l$, let

$$
\begin{aligned}
\omega_{l,k,j} =& \left(\tau_{l,k}^{-2} + \sigma_{l,k}^{-2} b_j^2\right)^{-1} \\
\upsilon_{l,k,j} =& \tau_{l,k}^{-2} \omega_{l,k,j} \\
\psi_{l,k}^{(n)} =& \gamma_{l,k} + \boldsymbol{\beta}_l \boldsymbol{u}_l^{(n)} \\
\tilde{\psi}_{l,k,j}^{(n)} =& \left(u_{l+1,k}^{(n)} - b_j'\right) b_j^{-1} \\
\lambda_{l,k,j}^{(n)} =& \upsilon_{l,k,j} \psi_{l,k}^{(n)} + (1 - \upsilon_{l,k,j}^2) \tilde{\psi}_{l,k,j}^{(n)} \\
\zeta_{l,k,1}^{(n)} =& 1 \\
\zeta_{l,k,j}^{(n)} =& \zeta_{l,k,j-1}^{(n)} \tilde{\zeta}_{l,k,j}^{(n)}, \qquad j \in 2, \ldots, J \\
\tilde{\zeta}_{l,k,j}^{(n)} =& \frac{\kappa_{l,k,j,j-1}^{(n)}}{\kappa_{l,k,j-1,j-1}^{(n)}} \cdot \frac{\tilde{\kappa}_{l,k,j-1,j-1}^{(n)}}{\tilde{\kappa}_{l,k,j,j-1}^{(n)}} \\
\kappa_{l,k,j,j'}^{(n)} =& \phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \mid c_{j'} b_j + b_j',\ \sigma_{l,k}^2\right) \\
\tilde{\kappa}_{l,k,j,j'}^{(n)} =& \phi_{\mathcal{N}}\left(c_{j'} \mid \lambda_{l,k,j}^{(n)},\ \omega_{l,k,j}^2\right) \\
\tilde{c}_{j,j'}^{(n)} =& (c_{j'} - \lambda_{l,k,j}) \omega_{l,k,j}^{-1} \\
\tilde{\pi}_{l,k,j}^{(n)} =& \Phi_{\mathcal{N}}(\tilde{c}_{j,j} \mid 0, 1) - \Phi_{\mathcal{N}}(\tilde{c}_{j,j-1} \mid 0, 1) \\
\pi_{l,k,j}^{(n)} \propto& \tilde{\pi}_{l,k,j}^{(n)} \zeta_{l,k,j}^{(n)}
\end{aligned}
$$

*Correspondence: jiankang@umich.edu, 1415 Washington Heights, Ann Arbor, MI 48109

Moreover, let

$$\boldsymbol{\mu}_l^{(n)} = \left[\boldsymbol{\beta}_l\boldsymbol{\beta}_l^\top + \left(\tilde{\boldsymbol{U}}_l^{(n)}\right)^{-1}\right]^{-1}\left[\left(\tilde{\boldsymbol{U}}_l^{(n)}\right)^{-1} h(\boldsymbol{v}_{l-1}^{(n)}) + \boldsymbol{\beta}_l^\top(\boldsymbol{v}_l^{(n)} - \boldsymbol{\gamma}_l)\right]$$
$$\boldsymbol{U}_l^{(n)} = \left[\boldsymbol{\beta}_l^\top \operatorname{diag}[\boldsymbol{\tau}_l^{-2}]\boldsymbol{\beta}_l + \left(\tilde{\boldsymbol{U}}_l^{(n)}\right)^{-1}\right]^{-1}, \qquad \tilde{\boldsymbol{U}}_l^{(n)} = \operatorname{diag}[\boldsymbol{\sigma}_{l-1}^2]$$

and

$$\begin{aligned}
\boldsymbol{\eta}_{l,k} =& (\bar{\boldsymbol{u}}_l\bar{\boldsymbol{u}}_l^\top + \tilde{\boldsymbol{B}}_{l,k}^{-1})^{-1}\boldsymbol{v}_{l,k}\bar{\boldsymbol{u}}_l^\top \\
\boldsymbol{B}_{l,k} =& (\tau_{l,k}^{-2}\bar{\boldsymbol{u}}_l\bar{\boldsymbol{u}}_l^\top + \tilde{\boldsymbol{B}}_{l,k}^{-1})^{-1}, & \tilde{\boldsymbol{B}}_{l,k} =& \operatorname{diag}(\boldsymbol{\rho}_{l,k}^2, \xi_{l,k}^2) \\
\bar{\boldsymbol{u}}_l =& (\boldsymbol{u}_l, \mathbf{1}), & \boldsymbol{u}_l =& \left[\boldsymbol{u}_l^{(n)}\right]_{n=1}^N.
\end{aligned}$$

Furthermore, let

$$\begin{aligned}
\boldsymbol{\epsilon}_{l,k}^2 =& \boldsymbol{v}_{l,k} - \boldsymbol{\beta}_{l,k}^\top\boldsymbol{u}_l - \gamma_{l,k} \\
\boldsymbol{\delta}_{l,k}^2 =& \boldsymbol{u}_{l+1,k} - h(\boldsymbol{v}_{l,k})
\end{aligned}$$

S1.2 Proofs of Theoretical Results

S1.2.1 Proof of Theorem 1

We first derive the posterior full conditional distribution of $v_{l,k}^{(n)}$ in Equation (9). Let

$$\omega_{n,l,k,j}^2 = \left(\tau_{l,k}^{-2} + \sigma_{l+1,k}^{-2}b_j^2\right)^{-1}, \qquad \lambda_{n,l,k,j} = \tau_{l,k}^{-2}\omega_{n,l,k,j}^2\left(\gamma_{l,k} + \boldsymbol{\beta}_l\boldsymbol{u}_l^{(n)}\right) + \sigma_{l+1,k}^{-2}b_j\omega_{n,l,k,j}^2\left(u_{l+1,k}^{(n)} - b_j'\right).$$

By Equations (6) and (7), we have

$$\begin{aligned}
f\left(v_{l,k}^{(n)} \,\middle|\, \text{rest}\right) =& f\left(v_{l,k}^{(n)} \,\middle|\, \boldsymbol{u}_l^{(n)}, \boldsymbol{\beta}_{l,k}, \tau_{l,k}, u_{l+1,k}^{(n)}, \sigma_{l,k}\right) \\
=& Cf\left(u_{l+1,k}^{(n)} \,\middle|\, v_{l,k}^{(n)}, \sigma_{l,k}\right)\cdot f\left(v_{l,k}^{(n)} \,\middle|\, \boldsymbol{u}_l^{(n)}, \boldsymbol{\beta}_{l,k}, \tau_{l,k}\right) \\
=& C\phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \,\middle|\, h\left(v_{l,k}^{(n)}\right),\ \sigma_{l,k}^2\right)\cdot\phi_{\mathcal{N}}\left(v_{l,k}^{(n)} \,\middle|\, \boldsymbol{\beta}_{l,k}\boldsymbol{u}_l^{(n)},\ \tau_{l,k}^2\right) \\
=& C\phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \,\middle|\, \sum_{j=1}^J\left(b_jv_{l,k}^{(n)} + b_j'\right)\cdot\mathbb{I}\left\{v_{l,k}^{(n)} \in [c_{j-1}, c_j)\right\},\ \sigma_{l,k}^2\right)\cdot\phi_{\mathcal{N}}\left(v_{l,k}^{(n)} \,\middle|\, \boldsymbol{\beta}_{l,k}\boldsymbol{u}_l^{(n)},\ \tau_{l,k}^2\right) \\
=& C\sum_{j=1}^J\phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \,\middle|\, b_jv_{l,k}^{(n)} + b_j',\ \sigma_{l,k}^2\right)\cdot\phi_{\mathcal{N}}\left(v_{l,k}^{(n)} \,\middle|\, \boldsymbol{\beta}_{l,k}^\top\boldsymbol{u}_l^{(n)},\ \tau_{l,k}^2\right)\cdot\mathbb{I}\left\{v_{l,k}^{(n)} \in [c_{j-1}, c_j)\right\} \\
=& C\sum_{j=1}^J C_j\phi_{\mathcal{N}}\left(v_{l,k}^{(n)} \,\middle|\, \lambda_{n,l,k,j},\ \omega_{n,l,k,j}^2\right)\cdot\mathbb{I}\left\{v_{l,k}^{(n)} \in [c_{j-1}, c_j)\right\} \\
=& \sum_{j=1}^J\pi_{n,l,k,j}\phi_{\mathcal{TN}}\left(v_{l,k}^{(n)} \,\middle|\, \lambda_{n,l,k,j},\ \omega_{n,l,k,j}^2,\ c_{j-1},\ c_j\right)
\end{aligned}$$

for some $\pi_{n,l,k,j} \in \mathbb{R}_+$ $(j = 1, \ldots, J)$ with $\sum_{j=1}^J\pi_{n,l,k,j} = 1$. (The second last equation holds by the same argument as that of normal-normal priors with known variance.) Moreover, for $j \in 2, \ldots, J$, we have

$$\begin{aligned}
\frac{\lim_{v_{l,k}^{(n)}\to c_{j-1}^+} f\left(v_{l,k}^{(n)} \,\middle|\, \text{rest}\right)}{\lim_{v_{l,k}^{(n)}\to c_{j-1}^-} f\left(v_{l,k}^{(n)} \,\middle|\, \text{rest}\right)} =& \frac{\lim_{v_{l,k}^{(n)}\to c_{j-1}^+}\sum_{j=1}^J\pi_{n,l,k,j}\phi_{\mathcal{TN}}\left(v_{l,k}^{(n)} \,\middle|\, \lambda_{n,l,k,j},\ \omega_{n,l,k,j}^2,\ c_{j-1},\ c_j\right)}{\lim_{v_{l,k}^{(n)}\to c_{j-1}^-}\sum_{j=1}^J\pi_{n,l,k,j}\phi_{\mathcal{TN}}\left(v_{l,k}^{(n)} \,\middle|\, \lambda_{n,l,k,j},\ \omega_{n,l,k,j}^2,\ c_{j-1},\ c_j\right)} \\
=& \frac{\pi_{n,l,k,j}\phi_{\mathcal{TN}}\left(c_{j-1} \,\middle|\, \lambda_{n,l,k,j},\ \omega_{n,l,k,j}^2,\ c_{j-1},\ c_j\right)}{\pi_{n,l,k,j-1}\phi_{\mathcal{TN}}\left(c_{j-1} \,\middle|\, \lambda_{n,l,k,j-1},\ \omega_{n,l,k,j-1}^2,\ c_{j-1},\ c_j\right)}
\end{aligned}$$

and

$$\begin{aligned}\frac{\lim_{v_{l,k}^{(n)}\to c_{j-1}^{+}} f\left(v_{l,k}^{(n)} \;\middle|\; \text{rest}\right)}{\lim_{v_{l,k}^{(n)}\to c_{j-1}^{-}} f\left(v_{l,k}^{(n)} \;\middle|\; \text{rest}\right)} =& \frac{\lim_{v_{l,k}^{(n)}\to c_{j-1}^{+}} \phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \;\middle|\; h\left(v_{l,k}^{(n)}\right),\, \sigma_{l,k}^2\right)\cdot \phi_{\mathcal{N}}\left(v_{l,k}^{(n)} \;\middle|\; \boldsymbol{\beta}_{l,k}\boldsymbol{u}_l^{(n)},\, \tau_{l,k}^2\right)}{\lim_{v_{l,k}^{(n)}\to c_{j-1}^{-}} \phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \;\middle|\; h\left(v_{l,k}^{(n)}\right),\, \sigma_{l,k}^2\right)\cdot \phi_{\mathcal{N}}\left(v_{l,k}^{(n)} \;\middle|\; \boldsymbol{\beta}_{l,k}\boldsymbol{u}_l^{(n)},\, \tau_{l,k}^2\right)}\\ =& \frac{\lim_{v_{l,k}^{(n)}\to c_{j-1}^{+}} \phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \;\middle|\; h\left(v_{l,k}^{(n)}\right),\, \sigma_{l,k}^2\right)}{\lim_{v_{l,k}^{(n)}\to c_{j-1}^{-}} \phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \;\middle|\; h\left(v_{l,k}^{(n)}\right),\, \sigma_{l,k}^2\right)}\\ =& \frac{\phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \;\middle|\; b_j c_{j-1}+b'_j,\, \sigma_{l,k}^2\right)}{\phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \;\middle|\; b_{j-1} c_{j-1}+b'_{j-1},\, \sigma_{l,k}^2\right)},\end{aligned}$$

which gives $\pi_{n,l,k,j}/\pi_{n,l,k,j-1} = \zeta'_{n,l,k,j}\pi'_{n,l,k,j}$, where

$$\begin{aligned}\zeta'_{n,l,k,j} =& \frac{\phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \;\middle|\; b_j c_{j-1}+b'_j,\, \sigma_{l,k}^2\right)}{\phi_{\mathcal{N}}\left(u_{l+1,k}^{(n)} \;\middle|\; b_{j-1} c_{j-1}+b'_{j-1},\, \sigma_{l,k}^2\right)}\cdot \frac{\phi_{\mathcal{N}}\left(c_{j-1} \;\middle|\; \lambda_{n,l,k,j-1},\, \omega_{n,l,k,j-1}^2\right)}{\phi_{\mathcal{N}}\left(c_{j-1} \;\middle|\; \lambda_{n,l,k,j},\, \omega_{n,l,k,j}^2\right)}\\ \pi'_{n,l,k,j} =& \frac{\Phi_{\mathcal{N}}\left((c_{j-1}-\lambda_{n,l,k,j-1})/\omega_{n,l,k,j-1}\right)}{\Phi_{\mathcal{N}}\left((c_{j-1}-\lambda_{n,l,k,j})/\omega_{n,l,k,j}\right)}.\end{aligned}$$

Define

$$\zeta_{n,l,k,j} = \begin{cases} 1, & \text{if } j=1,\\ \zeta_{n,l,k,j-1}\zeta'_{n,l,k,j}, & \text{if } j\in\{2,\ldots,J\}\end{cases}.$$

Then

$$\pi_{n,l,k,j} = \frac{\pi'_{n,l,k,j}\zeta_{n,l,k,j}}{\sum_{j=1}^{J}\pi'_{n,l,k,j}\zeta_{n,l,k,j}}.$$

For $l\in\{1,\ldots,L\}$, we have

$$\begin{aligned}f(\boldsymbol{v}_l \mid \boldsymbol{x}, \boldsymbol{\Theta}_l) =& \int_{\mathbb{R}^{K_{l-1}}} f(\boldsymbol{v}_l \mid \boldsymbol{v}_{l-1}, \boldsymbol{x}, \boldsymbol{\Theta}_l) f(\boldsymbol{v}_{l-1} \mid \boldsymbol{x}, \boldsymbol{\Theta}_l) d\boldsymbol{v}_{l-1}\\ =& \int_{\mathbb{R}^{K_{l-1}}} f(\boldsymbol{v}_l \mid \boldsymbol{v}_{l-1}, \boldsymbol{x}, \boldsymbol{\theta}_l) f(\boldsymbol{v}_{l-1} \mid \boldsymbol{x}, \boldsymbol{\Theta}_{l-1}) d\boldsymbol{v}_{l-1}.\end{aligned}$$

Moreover, by Equation (4),

$$\begin{aligned}\boldsymbol{v}_l =& \boldsymbol{\beta}_l[h(\boldsymbol{v}_{l-1})+\boldsymbol{\delta}_{l-1}]+\boldsymbol{\gamma}_l+\boldsymbol{\epsilon}_l\\ =& \boldsymbol{\beta}_l h(\boldsymbol{v}_{l-1})+\boldsymbol{\beta}_l\boldsymbol{\delta}_{l-1}+\boldsymbol{\gamma}_l+\boldsymbol{\epsilon}_l\\ \sim& \mathcal{MVN}\{\boldsymbol{\beta}_l h(\boldsymbol{v}_{l-1})+\boldsymbol{\gamma}_l,\, \boldsymbol{\beta}_l \operatorname{Cov}[\boldsymbol{\delta}_{l-1}]\boldsymbol{\beta}_l^\top + \operatorname{Cov}[\boldsymbol{\epsilon}_l]\}\\ =& \mathcal{MVN}\{\boldsymbol{\beta}_l h(\boldsymbol{v}_{l-1})+\boldsymbol{\gamma}_l,\, \boldsymbol{\beta}_l\boldsymbol{\Sigma}_{l-1}\boldsymbol{\beta}_l^\top + \boldsymbol{T}_l\}.\end{aligned}$$

The full conditional distributions of the rest of the model parameters (Equations (10) to (15)) follow the same argument as that of inverse gamma-normal conjugate priors in Bayesian linear regression (Wikipedia contributors, 2022a; Bishop and Nasrabadi, 2006, Sec. 2.3.3), as described in Lemmas S1.1 and S1.2.

Lemma S1.1. Suppose $\boldsymbol{X}\in\mathbb{R}^{K\times N}$ and $\boldsymbol{y}\in\mathbb{R}^{1\times N}$. If $\boldsymbol{\beta}\in\mathbb{R}^{1\times K}$ satisfies

$$\boldsymbol{\beta}\sim\mathcal{N}(\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0), \qquad \boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{X}\sim\mathcal{N}(\boldsymbol{\beta}\boldsymbol{X},\boldsymbol{T}),$$

for some $\boldsymbol{\mu}_0\in\mathbb{R}^{1\times K}$, $\boldsymbol{\Sigma}_0\in\mathbb{R}^{K\times K}$, and $\boldsymbol{T}\in\mathbb{R}^{N\times N}$, then the posterior distribution of $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X}\sim\mathcal{N}(\boldsymbol{\mu}_N,\boldsymbol{\Sigma}_N), \qquad \boldsymbol{\Sigma}_N=(\boldsymbol{\Sigma}_0^{-1}+\boldsymbol{X}\boldsymbol{T}^{-1}\boldsymbol{X}^\top)^{-1}, \qquad \boldsymbol{\mu}_N=\boldsymbol{\Sigma}_N(\boldsymbol{\mu}_0\boldsymbol{\Sigma}_0^{-1}+\boldsymbol{y}\boldsymbol{T}^{-1}\boldsymbol{X}^\top).$$

Proof. This result is a restatement of the properties of conditional Gaussian distributions discussed in (Bishop and Nasrabadi, 2006, Sec. 2.3.3). □

Lemma S1.2. Suppose $\sigma^2 \in \mathbb{R}_+$ and $\boldsymbol{x} = [x_i]_{i=1}^N \in \mathbb{R}^N$ satisfy

$$\sigma^2 \sim \mathcal{IG}(a_0, b_0), \qquad x_i|\sigma^2 \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$$

for some $a_0, b_0 \in \mathbb{R}_+$ and $\mu \in \mathbb{R}$. Then the posterior distribution of σ^2 is

$$\sigma^2|\boldsymbol{x} \sim \mathcal{IG}(a_N, b_N), \qquad a_N = a_0 + \frac{N}{2}, \qquad b_N = b_0 + \frac{\|\boldsymbol{x} - \boldsymbol{\mu}\|_2^2}{2}$$

.

Proof. This result is the widely known inverse gamma-normal conjugate priors for variance parameters with normally distributed observations and known mean (Wikipedia contributors, 2022b). □

Then Equations (10) and (11) are immediate consequences of Lemma S1.1, while Equations (12) to (15) follow Lemma S1.2 directly.

S1.2.2 Proof of Theorem 2

This formulation of the predictive density follows Equations (1) to (3) directly.

S1.2.3 Proof of Theorem 3

Lemma S1.3. Suppose $\boldsymbol{W} \in \mathbb{R}^K$ is a random vector and $\boldsymbol{t} \in \mathbb{R}^K$ a constant vector. Let $\boldsymbol{\beta} \in \mathbb{R}^{K' \times K}$, $\boldsymbol{\gamma} \in \mathbb{R}^{K'}$, and define $f_{\boldsymbol{\beta},\boldsymbol{\gamma}}(\boldsymbol{s}) = \boldsymbol{\beta}\boldsymbol{s} + \boldsymbol{\gamma}$. Moreover, let $\boldsymbol{\epsilon} \in \mathbb{R}^K$ be a random vector with $\text{Cov}[\boldsymbol{\epsilon}] = \text{diag}[\boldsymbol{\tau}^2]$ and $\boldsymbol{\epsilon} \perp \boldsymbol{W}$, and define $f_{\boldsymbol{\tau}^2}(\boldsymbol{s}) = \boldsymbol{s} + \boldsymbol{\epsilon}$. Further, let $\tilde{f}_c : \mathbb{R} \to \mathbb{R}$ be a Lipschitz function with Lipschitz constant c, and let $f_c(\boldsymbol{s}) = [\tilde{f}_c(s_1), \ldots, \tilde{f}_c(s_K)]$ for $\boldsymbol{s} = [s_1, \ldots, s_K] \in \mathbb{R}^K$. Then

$$\begin{aligned}
\text{E}\left[\|f_{\boldsymbol{\beta},\boldsymbol{\gamma}}(\boldsymbol{W}) - f_{\boldsymbol{\beta},\boldsymbol{\gamma}}(\boldsymbol{t})\|^2\right] &\leq \|\boldsymbol{\beta}\|_2^2 \,\text{E}\left[\|\boldsymbol{W} - \boldsymbol{t}\|^2\right] \\
\text{E}\left[\|f_{\boldsymbol{\tau}^2}(\boldsymbol{W}) - \boldsymbol{t}\|^2\right] &\leq \text{E}\left[\|\boldsymbol{W} - \boldsymbol{t}\|^2\right] + \text{sum}(\boldsymbol{\tau}^{\boldsymbol{2}}) \\
\text{E}\left[\|f_c(\boldsymbol{W}) - f_c(\boldsymbol{t})\|^2\right] &\leq c^2 \,\text{E}\left[\|\boldsymbol{W} - \boldsymbol{t}\|^2\right]
\end{aligned}$$

Proof. These results follow directly from the definitions of $f_{\boldsymbol{\beta},\boldsymbol{\gamma}}$, $f_{\boldsymbol{\tau}^2}$, and f_c. □

Let $S_l^2 = \sum_{k=1}^{K_l} \sigma_{l,k}^2$ and $R_l^2 = \sum_{k=1}^{K_l} \tau_{l,k}^2$. Define $g_0(\boldsymbol{x}|\boldsymbol{\Gamma}_0) = \boldsymbol{\beta}_0\boldsymbol{x} + \boldsymbol{\gamma}_0$ and $g_L(\boldsymbol{x}|\boldsymbol{\Gamma}_L) = \boldsymbol{\beta}_L h(g_{L-1}(\boldsymbol{x}|\boldsymbol{\Gamma}_{L-1})) + \boldsymbol{\gamma}_L$. Moreover, by definition $y = \boldsymbol{v}_L$ and $\boldsymbol{v}_l = \boldsymbol{\beta}_l[h(\boldsymbol{v}_{l-1}) + \boldsymbol{\delta}_{l-1}] + \boldsymbol{\gamma}_l + \boldsymbol{\epsilon}_l$ for $l > 0$. We use induction to show

$$\text{E}\left\{\|\boldsymbol{y} - g_L(\boldsymbol{x}, \boldsymbol{\Gamma}_L)\|^2 \;\middle|\; \boldsymbol{x}, \boldsymbol{\Theta}_L\right\} \leq \sum_{l=0}^{L} \left[d_l^2 S_{l-1}^2 + T_l^2\right] \left[\prod_{l'=l+1}^{L} d_{l'}^2\right] C_h^{2(L-l)}$$

all $L \geq 0$. For $L = 0$, we have

$$\text{E}\left\{\|\boldsymbol{y} - g_0(\boldsymbol{x}, \boldsymbol{\Gamma}_0)\|^2 \;\middle|\; \boldsymbol{x}, \boldsymbol{\Theta}_0\right\} = \text{E}\left\{\|\boldsymbol{\epsilon}_0\|^2 \;\middle|\; \boldsymbol{x}, \boldsymbol{\Theta}_0\right\} = T_0^2 = \sum_{l=0}^{0} \left[d_l^2 S_{l-1}^2 + T_l^2\right] \left[\prod_{l'=l+1}^{0} d_{l'}^2\right] C_h^{2(0-l)}.$$

For $L > 0$, we have

$$\begin{aligned}
\mathrm{E}\left\{\|\boldsymbol{y} - g_L(\boldsymbol{x}, \boldsymbol{\Gamma}_L)\|^2 \;\middle|\; \boldsymbol{x}, \boldsymbol{\Theta}_L\right\} =& \mathrm{E}\left\{\|\boldsymbol{\beta}_L[h(\boldsymbol{v}_{L-1}) + \boldsymbol{\delta}_{L-1}] + \boldsymbol{\gamma}_L + \boldsymbol{\epsilon}_L - [\boldsymbol{\beta}_L h(g_{L-1}(\boldsymbol{x}|\boldsymbol{\Gamma}_{L-1})) + \boldsymbol{\gamma}_L]\|^2 \;\middle|\; \boldsymbol{x}, \boldsymbol{\Theta}_L\right\} \\
\leq& \mathrm{E}\left\{\|\boldsymbol{\beta}_L[h(\boldsymbol{v}_{L-1}) + \boldsymbol{\delta}_{L-1}] + \boldsymbol{\gamma}_L - [\boldsymbol{\beta}_L h(g_{L-1}(\boldsymbol{x}|\boldsymbol{\Gamma}_{L-1})) + \boldsymbol{\gamma}_L]\|^2 \;\middle|\; \boldsymbol{x}, \boldsymbol{\Theta}_L\right\} + \mathrm{sum}(\boldsymbol{\tau}_L^2) \\
\leq& d_L^2\, \mathrm{E}\left\{\|h(\boldsymbol{v}_{L-1}) + \boldsymbol{\delta}_{L-1} - h(g_{L-1}(\boldsymbol{x}|\boldsymbol{\Gamma}_{L-1}))\|^2 \;\middle|\; \boldsymbol{x}, \boldsymbol{\Theta}_L\right\} + \mathrm{sum}(\boldsymbol{\tau}_L^2) \\
\leq& d_L^2\, \mathrm{E}\left\{\|h(\boldsymbol{v}_{L-1}) - h(g_{L-1}(\boldsymbol{x}|\boldsymbol{\Gamma}_{L-1}))\|^2 \;\middle|\; \boldsymbol{x}, \boldsymbol{\Theta}_L\right\} + \mathrm{sum}(\boldsymbol{\tau}_L^2) + d_L^2\, \mathrm{sum}(\boldsymbol{\sigma}_{L-1}^2) \\
\leq& d_L^2 C_h^2\, \mathrm{E}\left\{\|\boldsymbol{v}_{L-1} - g_{L-1}(\boldsymbol{x}|\boldsymbol{\Gamma}_{L-1})\|^2 \;\middle|\; \boldsymbol{x}, \boldsymbol{\Theta}_L\right\} + \mathrm{sum}(\boldsymbol{\tau}_L^2) + d_L^2\, \mathrm{sum}(\boldsymbol{\sigma}_{L-1}^2) \\
=& d_L^2 C_h^2\, \mathrm{E}\left\{\|\boldsymbol{v}_{L-1} - g_{L-1}(\boldsymbol{x}|\boldsymbol{\Gamma}_{L-1})\|^2 \;\middle|\; \boldsymbol{x}, \boldsymbol{\Theta}_L\right\} + T_L^2 + d_L^2 S_{L-1}^2 \\
\leq& \left\{\sum_{l=0}^{L-1}\left[d_l^2 S_{l-1}^2 + T_l^2\right]\left[\prod_{l'=l+1}^{L} d_{l'}^2\right] C_h^{2(L-l)}\right\} + T_L^2 + d_L^2 S_{L-1}^2 \\
=& \left\{\sum_{l=0}^{L-1}\left[d_l^2 S_{l-1}^2 + T_l^2\right]\left[\prod_{l'=l+1}^{L} d_{l'}^2\right] C_h^{2(L-l)}\right\} + \sum_{l=L}^{L}\left[d_l^2 S_{l-1}^2 + T_l^2\right]\left[\prod_{l'=l+1}^{L} d_{l'}^2\right] C_h^{2(L-l)} \\
=& \left\{\sum_{l=0}^{L}\left[d_l^2 S_{l-1}^2 + T_l^2\right]\left[\prod_{l'=l+1}^{L} d_{l'}^2\right] C_h^{2(L-l)}\right\},
\end{aligned}$$

where the inequalities hold by Lemma S1.3 and the inductive hypothesis.

S1.2.4 Proof of Corollary 4

First, observe that the activation functions in the statement of Corollary 4 are all Lipschitz functions with Lipschitz function $C_h \leq 1$. Then by applying the global bounds to Corollary 4, we have

$$\begin{aligned}
\mathrm{E}\left\{\|\boldsymbol{y} - g_L(\boldsymbol{x}, \boldsymbol{\Gamma}_L)\|^2 \;\middle|\; \boldsymbol{x}, \boldsymbol{\Theta}_L\right\} \leq& \sum_{l=0}^{L}\left(d^2 K\sigma^2 + K\tau^2\right) d^{2(L-l)} C_h^{2(L-l)} \\
=& K\left(d^2\sigma^2 + \tau^2\right)\sum_{l=0}^{L} d^{2(L-l)} C_h^{2(L-l)} \\
\leq& K\left(d^2\sigma^2 + \tau^2\right)\sum_{l=0}^{L} d^{2(L-l)} \\
\leq& K(d^2+1)\left(\sigma^2 + \tau^2\right)\sum_{l=0}^{L} d^{2(L-l)} \\
=& K(d^2+1)\left(\sigma^2 + \tau^2\right)\sum_{l=0}^{L} d^{2l} \\
\leq& K(d^2+1)\left(\sigma^2 + \tau^2\right)\sum_{l=0}^{L}(d^{2L}+1) \\
=& KL(d^{2L}+1)(d^2+1)\left(\sigma^2 + \tau^2\right) \\
\leq& 3KL(d^{2(L+1)}+1)(\sigma^2 + \tau^2),
\end{aligned}$$

which completes the proof.

S2 Synthetic data experiments

S2.1 Experiment setup

S2.1.1 Data simulation

We simulated synthetic data with different conditional distribution patterns to evaluate each method's ability of learning the density. We sampled one-dimensional input value from $\mathcal{U}[-1,1]$ or $0.9\mathcal{U}[-1,0]+0.1\mathcal{U}[0,1]$ For the output value, we generated it by $y=m(x)+\eta(x)$, where $m(x)$ is a deterministic median function and $\eta(x)$ is the random variable of the noise, such that $\text{median}[\eta(x)]=0$ for all x. We fix the median of y to the piecewise linear function

$$m(x)=(x+1)\mathbb{I}[x\leq -0.5]-x\mathbb{I}[-0.5<x<0]+x\mathbb{I}[x>0]$$

and simulated three different types of noise by using uniform distributions or their mixtures:

1. Heteroscedastic noise:

$$\eta(x)\sim\begin{cases}\mathcal{U}[-0.5,0.5], & \text{if } x\in[-0.85,-0.65]\cup[-0.35,-0.15]\cup[0.35,0.65]\\ \mathcal{U}[-0.1,0.1], & \text{otherwise}\end{cases}$$

2. Skewed noise:

$$\eta(x)\sim\begin{cases}0.5\mathcal{U}[-0.1,0]+0.5\mathcal{U}[0,0.8], & \text{if } x\in[-0.5,0]\\ 0.5\mathcal{U}[-0.8,0]+0.5\mathcal{U}[0,0.1], & \text{otherwise}\end{cases}$$

3. Multimodal noise:

$$\eta(x)\sim\begin{cases}0.3\mathcal{U}[-0.5,-0.4]+0.4\mathcal{U}[-0.4,0.4]+0.3\mathcal{U}[0.4,0.5], & \text{if } x\in[-0.5,0.5]\\ \mathcal{U}[-0.125,0.125], & \text{otherwise}\end{cases}$$

The training sample size varied among 1000, 2000, 4000, and the testing size was 10% of the training size. We randomly generated 20 datasets for each noise type and sample size.

S2.1.2 Model setup

Model setups were the same as those in the UCI data experiments (Section S3.1.2), except for DMC, which did not give meaningful predictions and used a virtually constant function with very large predictive variance to fit all the datasets.

S2.1.3 Evaluation criteria

To evaluate the accuracy of the estimated predictive distribution, we computed its difference from the true predictive distribution. We used a grid of 100 values of x on $[-1,1]$. For every x, we computed the empirical $0.025, 0.075, \ldots, 0.925, 0.975$ quantiles of the estimated predictive distribution and the true distribution. Then we computed the average absolute difference between the estimated and true quantiles, which is equivalent to numerically computing the L_1 distance between the inverse CDFs of the estimated and true predictive distributions, and averaged them across all the grid points of x.

S2.2 Additional results

The estimated predictive density of baseline methods not included in Figure 1 are displayed in Figure S1.

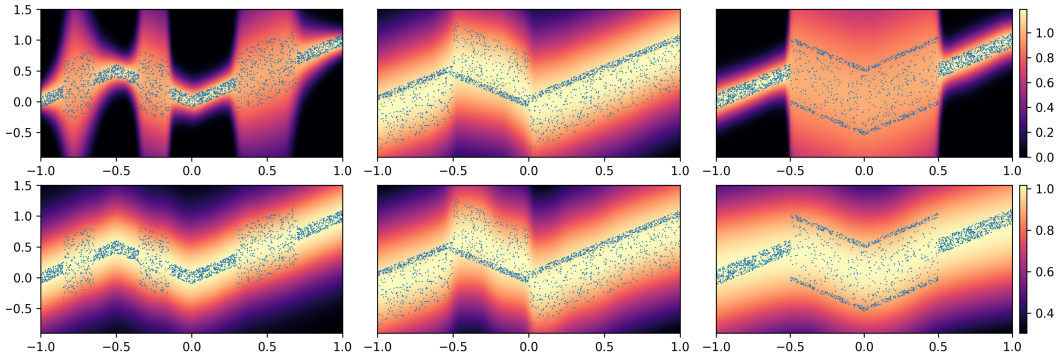
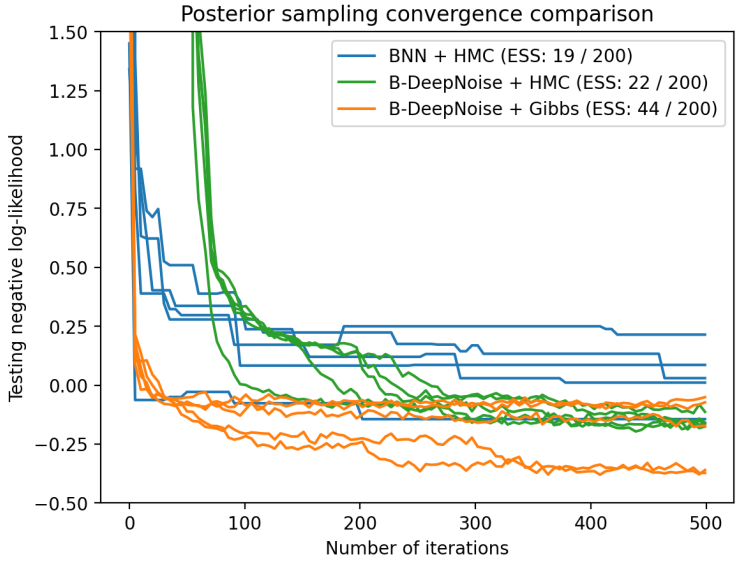


Figure S1: Predictive densities estimated by BP (top row) and VI (bottom row) for heteroscedastic (left), skewed (middle), and multimodal (right) noise.

Figure S2: Comparison of the convergence speed of B-DeepNoise with BNN on synthetic data with heteroscedastic noise and 400 training samples. Both B-DeepNoise and BNN use 2 hidden layers with 25 nodes per layer and draw 500 posterior samples on each of 5 independent chains. Effective sample size (ESS) is computed using the last 200 posterior samples and averaged across the chains.



S3 UCI data experiments

S3.1 Experiment setup

S3.1.1 Dataset curation

We downloaded the nine UCI datasets from `https://github.com/yaringal/DropoutUncertaintyExps`, including the indices of the random splits. See the repository and (Gal and Ghahramani, 2016) for details.

S3.1.2 Model setup

B-DeepNoise. We assigned $\mathcal{IG}(0.001, 0.001)$ as weakly informative priors to the variance parameters. The network included 4 hidden layers with 50 hidden nodes per layer. The activation function was the hard tanh function $h(x) = \min(1, \max(-1, x))$. The parameters were initialized by stochastic gradient descent, and 500

posterior samples were drawn. The input and output were centered at 0 and scaled to 1, where the centers and scales were computed by using the training samples only.

All the baseline methods had the same architecture, activation, and normalization as B-DeepNoise. For GPU computation, we used an NVIDIA Quadro P6000.

Dropout Monte Carlo (DMC). We used the implementation by the original authors (Gal and Ghahramani, 2016), including their hyperparameter tuning procedures. See `https://github.com/yaringal/DropoutUncertaintyExps`. for details.

Variational Inference (VI) We used the implementation by (Ritter and Karaletsos, 2022). The batch size was 100, and the model was trained for 2000 epochs with a learning rate of 0.001.

Deep ensemble (DE). We followed the recommendations of the original authors (Lakshminarayanan et al., 2016) and used 5 independent networks. The predictive MSE was optimized by Adam (Kingma and Ba, 2014) with a learning rate of 0.1. The model was trained for 40 epochs.

Backpropagation (BP). We used two feed-forward neural networks, where the second network included a softplus layer on the output layer. The first network outputs the predictive mean, while the second network outputs the predictive variance. The Gaussian NLL was used as the loss function. The training hyperparameters were the same as those for DE.

Bayesian Neural Network (BNN). We used a BNN with learnable predictive variance, i.e. the model outputs two parameters, one for the predictive mean and the other for the predictive variance. The Gaussian NLL was used as the loss function. We used standard normal distributions as priors for all the weight and bias parameters except for those connected to the log predictive variance, which used $\mathcal{N}(\mathbf{0}, 0.25\boldsymbol{I})$ as priors. To reduce the computation cost, parameters were initialized by stochastic gradient decent. The posterior distribution was simulated by Hamiltonian Monte Carlo (Duane et al., 1987). Each HMC iteration involved 10 leapfrog steps. The step size was initialized to 0.01 and dynamically adjusted to achieve an acceptance rate of 0.75 Andrieu and Thoms (2008).

S3.1.3 Evaluation criteria

In this section, we compare the calibrated prediction intervals (CPI) with the uncalibrated prediction intervals (UPI). Conceptually, the average width of the 95% calibrated prediction intervals (WCPI-95) is the miscalibration-adjusted version of the 95% uncalibrated prediction intervals (WUPI-95). WUPI-95 is a popular metric for evaluating uncertainty quantification (UQ) efficiency. However, a major flaw of WUPI-95 is that when a method is overconfident about the testing data and underestimate the predictive uncertainty, its WUPI-95 can be very small and does not reflect the actural inaccuracy. Thus if a method has a small WUPI-95, the cause could be 1) that the method is both well-calibrated and efficient in UQ, or 2) that the method is miscalibrated in UQ. One has to look up the empirical coverage rates of the UPIs on the testing data to check the accuracy of UQ and differentiate the two situations. In other words, it is unfair the compare the widths of the 95% UPIs of two methods if one method empirically covers a much smaller percentage of the testing samples compared to the other method.

Therefore, in order to use a single metric that systematically measures UQ efficiency and is invariant to mis-calibration, we propose the WCPI-95. The WCPI-95 of a method is defined to be the WUPI-x, where x is the smallest positive number such that the x% UPIs cover no less than 95% of the outcomes on the testing data. Computationally, we iterate through the 1% UPIs, 2% UPIs, ..., 99% UPIs, 100% UPIs until at least 95% of the testing samples are covered (say we are at the x% UPIs at this step), and then we find the average width of the x%-UPIs, which gives us the WCPI-95. For example, suppose Method A is over-confident in its predictions, where its 95% UPIs cover less than 95% of the testing samples in average, and its 99% UPIs actually cover 95% of the testing samples in average. In that case, the WCPI-95 of Method A is equal to its WUPI-99, which is larger than its WUPI-95 and thus corrects for its over-confidence. On the other hand, if Method B is under-confident, its WCPI-95 will be less than its WUPI-95. Finally, if method C is perfectly

calibrated such that its 95% UPIs cover exactly 95% of the observations on the testing data, then its WCPI-95 is equal to its WUPI-95. Then by comparing the WCPI-95 of Methods A, B, and C, we can evaluate the UQ efficiency of the three methods fairly without having to worry that any of the methods may "hack" the theoretical predictive interval width by being consistently over-confident.

S3.2 Additional results

In addition to the RMSE, NLL, and WCPI-95 (Table 2), we also computed the average coverage rate of the 95% prediction intervals on the testing data. The results are reported in Table S1.

Table S1: Additional UCI experiment results on testing data for B-DeepNoise and baseline methods.

Dataset	BP	VI	BNN	DMC	DE	B-DeepNoise
Coverage Rate of 95% Prediction Intervals						
Yacht Hydrodynamics	0.92 ±0.06	1.00 ±0.00	1.00 ±0.00	0.81 ±0.08	0.97 ±0.04	0.96 ±0.03
Boston Housing	0.85 ±0.06	0.97 ±0.02	0.94 ±0.00	0.86 ±0.04	0.90 ±0.05	0.92 ±0.04
Energy Efficiency	0.92 ±0.05	1.00 ±0.01	0.99 ±0.00	0.94 ±0.03	0.98 ±0.02	0.96 ±0.05
Concrete Strength	0.86 ±0.05	0.89 ±0.03	0.97 ±0.02	0.88 ±0.02	0.92 ±0.03	0.93 ±0.03
Wine Quality	0.85 ±0.04	0.62 ±0.05	0.91 ±0.06	0.61 ±0.04	0.91 ±0.02	0.92 ±0.03
Kin8nm	0.86 ±0.04	0.55 ±0.02	0.94 ±0.04	1.00 ±0.00	0.95 ±0.01	0.96 ±0.01
Power Plant	0.94 ±0.01	0.87 ±0.01	0.97 ±0.01	0.73 ±0.02	0.95 ±0.01	0.95 ±0.01
Naval Propulsion	0.94 ±0.14	0.95 ±0.02	1.00 ±0.00	1.00 ±0.00	1.00 ±0.96	1.00 ±0.00
Protein Structure	0.93 ±0.01	0.83 ±0.01	0.98 ±0.01	0.57 ±0.02	0.96 ±0.00	0.93 ±0.00

Table S2: Runtime (sec) for UCI data experiments.

Dataset	BP	VI	BNN	DMC	DE	B-DeepNoise
Yacht Hydrodynamics	4	133	772	3482	8	534
Boston Housing	4	180	1090	3523	11	570
Energy Efficiency	5	247	1204	4456	12	625
Concrete Strength	7	331	1407	4551	15	657
Wine Quality	7	580	1046	4599	19	837
Kin8nm	25	3195	2042	7801	77	1266
Power Plant	29	3657	2514	6101	88	2028
Naval Propulsion	33	3586	2875	8876	83	3270
Protein Structure	127	11948	12518	12112	859	5700

S4 ABCD data analysis

S4.1 Experiment setup

S4.1.1 Data overview

The ABCD study aims at identifying the associations between brain development and cognitive behaviors. A total of 11,800 children aged between 9 and 10 participated in the study. In our experiments, we used Release 1.1, which consisted of minimally preprocessed fMRI data from 21 imaging sites (Hagler Jr et al., 2019). The images we analyzed was the 2-back task-based contrast map. The 2-back task engages brain regions associated with cognitive functions and memory regulation processes. The imaging resolution was 2mm, with each image containing $61 \times 73 \times 61$ voxels. The outcome variable was general intelligence score (g-score) (Sripada et al., 2020). The non-imaging features included 2-back task score, general psychopathology factor, age, sex, highest parental education level, household marital status, household income bracket, and self-identified racial and

ethnic memberships. Categorical variables were coded as multiple binary dummy variables. For the imaging features, we divided the brain into 90 AAL regions and extracted the average imaging value inside each region as a feature. The 90 imaging features were concatenated with the non-imaging features to form the input variables. After removing incomplete observations, the dataset contained 1191 subjects.

S4.1.2 Model setup

The settings of B-DeepNoise were the same as that in the UCI data experiments (Section S3.1.2).

S4.2 Additional results

In addition to Figure 3, we also computed the influence of brain regions on the predictive variance, as visualized in Figure S3. The magnitudes of influence of the top five most influential features on both the predictive mean and variance are reported in Table S3.

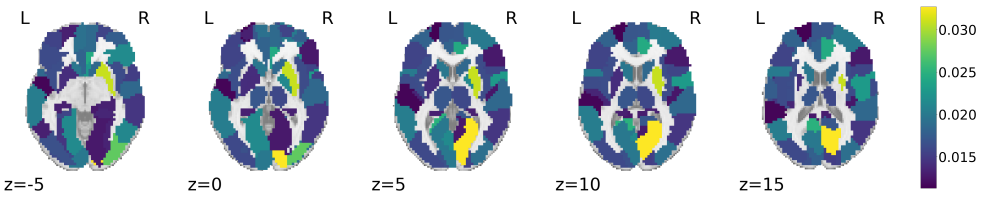


Figure S3: Influence of brain regions on the predictive variance of the g-score.

Table S3: Most influential features on the predictive mean and variance of g-score.

Predictive Mean		Predictive Variance	
feature	influence	feature	influence
CorrectRate2bk	1.3235	CorrectRate2bk	0.0659
Calcarine.R	0.6269	Calcarine.R	0.0327
Putamen.R	0.6012	Paracentral.Lobule.R	0.0312
Paracentral.Lobule.R	0.5786	Putamen.R	0.0309
Rectus.R	0.4770	Occipital.Inf.R	0.0275

S5 DISCUSSION OF CLASSIFICATION TASKS

S5.1 Theoretical Comparison of Stochastic and Deterministic Models for Classification

In this section, we demonstrate that for categorical outcomes, stochastic models do not have theoretical advantage over deterministic models that are sufficiently flexible. Consider a standard classification model, where the K-class outcome variable is coded as a one-hot vector $Y \in \{0,1\}^K$ and, given input features x, is modeled as $Y \sim \text{Categorical}(\text{softmax}(g(x)))$, where $g(x)$ is a deterministic function, and $\text{E}[Y] = \text{softmax}(g(x))$ is the conditional expected value of the outcome, which is a probability vector. If we replace $g(x)$ with a stochastic function $\tilde{g}(x, Z)$, where Z is a random seed with density function $f(z)$, then the outcome variable follows

$$Y \sim \text{Categorical}(\text{softmax}(\tilde{g}(x, Z))), \qquad Z \sim f.$$

In this case, by the law of total expectation, the conditional expected value of the outcome is

$$\text{E}[Y|x] = \text{E}[\text{E}[Y|Z, x]] = \text{E}[\text{softmax}(\tilde{g}(x, Z))] = \int_z \text{softmax}(\tilde{g}(x, z)) f(z) dz.$$

Define $\tilde{p}(x) = \int_z \text{softmax}(\tilde{g}(x,z))f(z)dz$, which is a deterministic function that maps the input features to a probability vector. Since a categorical distribution is completely determined by its expected value, the derivation above implies that if $g(x)$ is flexible enough such that $\text{softmax}(g(x))$ can approximate $\tilde{p}(x)$ with arbitrary precision then the predictive distribution of the deterministic model can approximate the the predictive distribution of the stochastic model with arbitrary precision.

The last condition holds for DNNs by their universal approximation property. Thus for classification tasks, any stochastic model can be replaced with a deterministic DNN model with arbitrary small difference in the predictive distribution. This theoretical result renders the flexible density learning capacity of B-DeepNoise unhelpful for classification tasks.

S5.2 B-DeepNoise for Categorical Outcomes

Although B-DeepNoise is not theoretically expected to have better uncertainty quantification accuracy than standard DNNs on classification tasks, (as demonstrated in Section S5.1) B-DeepNoise is still capable of learning the predictive distributions of categorical outcomes. This ability is achieved by adding a softmax activation function to the output layer. Then the posterior full-conditional distributions of the model parameters are the same except for those in the output layer, which we derive in this section.

Let $\boldsymbol{y}^{(n)} = [y_k^{(n)}]_{k=1}^K \in \mathbb{R}^K$ be a K-class categorical outcome variable, represented as a one-hot vector:

$$y_k^{(n)} = \begin{cases} 1, & \text{if } k \in \{1,\dots,K\} \setminus \left\{\bar{k}^{(n)}\right\} \\ 0, & \text{if } k = \bar{k}^{(n)} \end{cases},$$

where $\bar{k}^{(n)}$ is the true category for $\boldsymbol{y}^{(n)}$. Using B-DeepNoise to model the distribution of the outcome given input features $\boldsymbol{x}^{(n)}$, we have

$$\text{softmax}^{-1}\left\{\text{E}\left[\boldsymbol{y}^{(n)}\right]\right\} = \boldsymbol{\epsilon}_L^{(n)} + \boldsymbol{\gamma}_L + \boldsymbol{\beta}_L\left[\boldsymbol{\delta}_L^{(n)} + h\left(\cdots \boldsymbol{\epsilon}_1^{(n)} + \boldsymbol{\gamma}_1 + \boldsymbol{\beta}_1\left[\boldsymbol{\delta}_0^{(n)} + h\left(\boldsymbol{\epsilon}_0^{(n)} + \boldsymbol{\gamma}_0 + \boldsymbol{\beta}_0\boldsymbol{x}^{(n)}\right)\right]\cdots\right)\right] \tag{1}$$

where

$$\text{softmax}(z_k) = \begin{cases} \frac{\exp(z_k)}{\sum_{k=1}^{K-1}\exp(z_k)+1}, & \text{if } k \in \{1,\dots,K-1\} \\ \frac{1}{\sum_{k=1}^{K-1}\exp(z_k)+1}, & \text{if } k = K \end{cases},$$

which we abbreviate as

$$\text{softmax}(\boldsymbol{z}) = \frac{[\exp(\boldsymbol{z}), 1]}{\sum\exp(\boldsymbol{z}) + 1}.$$

Notice that the Kth element in $\boldsymbol{y}^{(n)}$ by definition has logit equal to 0, so that the scale of $\boldsymbol{z}$ is identifiable. The task is to sample $\boldsymbol{z}^{(n)}|\boldsymbol{y}^{(n)}, \boldsymbol{\mu}^{(n)}, \tau^2$ from the model

$$\begin{aligned} \boldsymbol{z}^{(n)} &\overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mu}^{(n)}, \tau^2) \\ \boldsymbol{\pi}^{(n)} &= \frac{\left[\exp(\boldsymbol{z}^{(n)}), 1\right]}{\sum\exp(\boldsymbol{z}^{(n)}) + 1} \\ \boldsymbol{y}^{(n)} &\sim \text{Categorical}(\boldsymbol{\pi}^{(n)}), \end{aligned}$$

where $\boldsymbol{z}^{(n)}, \boldsymbol{\mu}^{(n)} \in \mathbb{R}^{K-1}$ and $\tau^2 \in \mathbb{R}$. To do this, we update one element of $\boldsymbol{z}^{(n)}$ at a time while fixing the

other $K-2$ elements. Then the joint log density for element k is

$$\begin{aligned}
-\log f\left(z_k^{(n)}, \boldsymbol{y}^{(n)} \middle| \boldsymbol{z}_{-k}^{(n)}, \mu_k^{(n)}, \tau^2\right) &= -\log f\left(z_k^{(n)}, k^{(n)} \middle| \boldsymbol{z}_{-k}^{(n)}, \mu_k^{(n)}, \tau^2\right) \\
&= \frac{1}{2\tau^2}\left(z_k^{(n)} - \mu_k^{(n)}\right)^2 + \log\left[\exp\left(z_k^{(n)}\right) + \sum \exp\left(\boldsymbol{z}_{-k}^{(n)}\right)\right] - z_{\bar{k}^{(n)}}^{(n)} + C_0 \\
&= \frac{1}{2\tau^2}\left(z_k^{(n)} - \mu_k^{(n)}\right)^2 + \log\left[\frac{\exp\left(z_k^{(n)}\right)}{\sum \exp\left(\boldsymbol{z}_{-k}^{(n)}\right) + 1} + 1\right] - z_{\bar{k}^{(n)}}^{(n)} + C_1 \\
&= \frac{1}{2\tau^2}\left(z_k^{(n)} - \mu_k^{(n)}\right)^2 + \log\left[\exp\left(z_k^{(n)} - a_k^{(n)}\right) + 1\right] - z_{\bar{k}^{(n)}}^{(n)} + C_1 \\
&= \frac{1}{2\tau^2}\left(z_k^{(n)} - \mu_k^{(n)}\right)^2 + \log\left[\exp\left\{s_k^{(n)}\left(z_k^{(n)} - a_k^{(n)}\right)\right\} + 1\right] + C_2
\end{aligned}$$

where

$$\begin{aligned}
a_k^{(n)} &= \log\left[\sum \exp\left(\boldsymbol{z}_{-k}^{(n)}\right) + 1\right] \\
s_k^{(n)} &= \begin{cases} -1, & \text{if } k = \bar{k}^{(n)}, \\ 1, & \text{otherwise,} \end{cases}
\end{aligned}$$

Notice that $\log\left[\exp\left(z_k^{(n)} - a_k^{(n)}\right) + 1\right]$ is the softplus function with respect to $z_k^{(n)}$ centered at $a_k^{(n)}$, which approaches the ReLU function $\max(0, \cdot)$ when $z_k^{(n)} \to \pm\infty$, and is convex around $a_k^{(n)}$. Thus we can approximate it by breaking its domain into three parts:

$$\begin{aligned}
\log\{\exp[s(z-a)] + 1\} = \phi[s(z-a)] \approx \psi[s(z-a)] &= \begin{cases} 0 & \text{if } s(z-a) < -\frac{1}{2c} \\ \frac{c}{2}(z - a + \frac{s}{2c})^2 & \text{if } -\frac{1}{2c} \le s(z-a) \le \frac{1}{2c} \\ s(z-a) & \text{if } s(z-a) > \frac{1}{2c} \end{cases} \\
&= \begin{cases} \frac{s-1}{2}(z-a) & \text{if } z \in (-\infty,\, a - \frac{1}{2c}) \\ \frac{c}{2}(z - a + \frac{s}{2c})^2 & \text{if } z \in [a - \frac{1}{2c},\, a + \frac{1}{2c}] \\ \frac{s+1}{2}(z-a) & \text{if } z \in (a + \frac{1}{2c},\, \infty) \end{cases}
\end{aligned}$$

where $c > 0$ is a constant for approximating the logistic function with a hard sigmoid function

$$\{\exp[s(z-a)]^{-1} + 1\}^{-1} = \phi'[s(z-a)] \approx \psi'[s(z-a)] = \min[\max[0.5 + sc(z-a), 0], 1].$$

For example, the first-order Taylor polynomial of ϕ' at 0 sets $c = 0.25$, while TensorFlow and Theano sets $c = 0.2$, and PyTorch sets $c = 1/6$. (For the middle part, we may be tempted to use the Taylor polynomial of ϕ centered at a (i.e. $\log(2) + 0.5(z-a) + 0.125(z-a)^2$) or centered at one of the two boundary points, but that does not guarantee the overall function to be continuous.) Then the density function is broken into

three cases:

$$
\begin{aligned}
&-\log f\Big(z_k^{(n)}, \boldsymbol{y}^{(n)}\Big|\boldsymbol{z}_{-k}^{(n)}, \mu_k^{(n)}, \tau^2\Big) \\
&= \frac{1}{2\tau^2}\Big(z_k^{(n)} - \mu_k^{(n)}\Big)^2 + \log\Big[\exp\Big\{s_k^{(n)}\Big(z_k^{(n)} - a_k^{(n)}\Big)\Big\} + 1\Big] + C_2 \\
&\approx \frac{1}{2\tau^2}\Big(z_k^{(n)} - \mu_k^{(n)}\Big)^2 + \psi\Big[s_k^{(n)}\Big(z_k^{(n)} - a_k^{(n)}\Big)\Big] + C_2 \\
&= \begin{cases}
\frac{1}{2}\tau^{-2}\Big(z_k^{(n)} - \mu_k^{(n)}\Big)^2 + C_2 & \text{if } s_k^{(n)}\Big(z_k^{(n)} - a_k^{(n)}\Big) < -\frac{1}{2c} \\
\frac{1}{2}(\tau^{-2}+c)\Big\{z - \Big[\frac{\tau^{-2}}{\tau^{-2}+c}\mu + \frac{c}{\tau^{-2}+c}\Big(a - \frac{s_k^{(n)}}{2c}\Big)\Big]\Big\} + C_3 & \text{if } -\frac{1}{2c} \le s_k^{(n)}\Big(z_k^{(n)} - a_k^{(n)}\Big) \le \frac{1}{2c} \\
\frac{1}{2}\tau^{-2}\Big[z_k^{(n)} - \Big(\mu_k^{(n)} - s_k^{(n)}\tau^2\Big)\Big]^2 + C_4 & \text{if } s_k^{(n)}\Big(z_k^{(n)} - a_k^{(n)}\Big) > \frac{1}{2c}
\end{cases} \\
&= \begin{cases}
\frac{1}{2}\tau^{-2}\Big[z_k^{(n)} - \Big(\mu_k^{(n)} - \frac{s_k^{(n)}-1}{2}\tau^2\Big)\Big]^2 + C_4 & \text{if } z_k^{(n)} \in \Big(-\infty,\, a_k^{(n)} - \frac{1}{2c}\Big) \\
\frac{1}{2}(\tau^{-2}+c)\Big\{z - \Big[\frac{\tau^{-2}}{\tau^{-2}+c}\mu + \frac{c}{\tau^{-2}+c}\Big(a - \frac{s_k^{(n)}}{2c}\Big)\Big]\Big\} + C_3 & \text{if } z_k^{(n)} \in \Big[a_k^{(n)} - \frac{1}{2c},\, a_k^{(n)} + \frac{1}{2c}\Big] \\
\frac{1}{2}\tau^{-2}\Big[z_k^{(n)} - \Big(\mu_k^{(n)} - \frac{s_k^{(n)}+1}{2}\tau^2\Big)\Big]^2 + C_4 & \text{if } z_k^{(n)} \in \Big(a_k^{(n)} + \frac{1}{2c},\, \infty\Big)
\end{cases}
\end{aligned}
$$

In all the cases the density has a quadratic form, and the density overall is continuous, which implies that the distribution is a three-component mixture of truncated normal distributions with adjacent truncation points.

References

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and computing*, 18(4):343–373.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, 195(2):216–222.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Hagler Jr, D. J., Hatton, S., Cornejo, M. D., Makowski, C., Fair, D. A., Dick, A. S., Sutherland, M. T., Casey, B., Barch, D. M., Harms, M. P., et al. (2019). Image processing and analysis methods for the adolescent brain cognitive development study. *Neuroimage*, 202:116091.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2016). Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*.

Ritter, H. and Karaletsos, T. (2022). Tyxe: Pyro-based bayesian neural nets for pytorch. *Proceedings of Machine Learning and Systems*, 4.

Sripada, C., Rutherford, S., Angstadt, M., Thompson, W. K., Luciana, M., Weigard, A., Hyde, L. H., and Heitzeg, M. (2020). Prediction of neurocognition in youth from resting state fmri. *Molecular psychiatry*, 25(12):3413–3421.

Wikipedia contributors (2022a). Bayesian linear regression — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Bayesian_linear_regression&oldid=1089211815`. [Online].

Wikipedia contributors (2022b). Conjugate prior — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Conjugate_prior&oldid=1070993689`. [Online].