Boosting Performance of Latent Binary Neural Networks With a Local Reparametrization Trick and Normalizing Flows

Lars Skaaret-Lund[2] **Aliaksandr Hubin**[1,2,4] Geir Storvik[1,3]

[1] Department of Mathematics, UiO
[2] BIAS, NMBU
[3] SAMBA, NR
[4] OUC

25.11.2022

Introduction. Issues with neural networks

- Neural networks (NNs) are flexible parametric models;
- But NNs overfit and do not provide uncertainty measures;
- Bayesian neural networks (BNNs) resolve that;
- BNNs are still heavily over-parameterized and uninterpretable;
- A solution is model uncertainties in BNNs;
- We develop latent binary BNN (LBBNN) for that.

The likelihood model for LBBNN

$$\boldsymbol{y}_i \sim \mathfrak{f}\left(\boldsymbol{\mu}_i, \phi\right), \quad i \in \{1, ..., n\} \tag{1}$$

$$\boldsymbol{\mu}_i = \{x_{i1}^{(L)}, ..., x_{ir}^{(L)}\}, \tag{2}$$

$$x_{ij}^{(l+1)} = \sigma_j^{(l)} \left(\sum_{k=0}^{p^{(l)}} \gamma_{kj}^{(l)} \beta_{kj}^{(l)} x_{ik}^{(l)} \right), j > 0 \tag{3}$$

$f(\cdot|\boldsymbol{\mu}, \phi)$ is a distribution with expectation μ and dispersion ϕ;

$\beta_{kj}^{(l)} \in \mathcal{R}$ are the weights (slope coefficients) for the inputs $\boldsymbol{x}_{ik}^{(l)}$;

$\gamma_{kj}^{(l)} \in \{0, 1\}$ are latent indicators switching the weights on and off;

$p^{(l)}$ is the number of neurons at layer l;

L is the total number of layers;

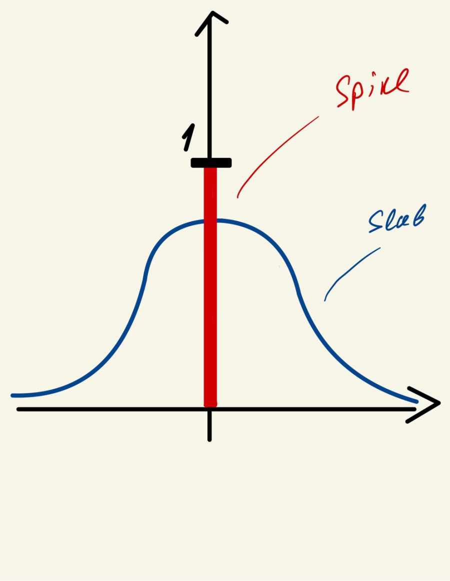
$x_{i0}^{(l+1)} = 1$ is a consant for the intercept.

Model and parameter priors

$$p(\beta_{kj}^{(l)}|\sigma_{\beta,l}^2,\gamma_{kj}^{(l)}) = \gamma_{kj}^{(l)}\mathcal{N}(0,\sigma_{\beta,l}^2) + (1-\gamma_{kj}^{(l)})\delta_0(\beta_{kj}^{(l)}),$$
$$p(\gamma_{kj}^{(l)}) = \text{Bernoulli}(\psi^{(l)}).$$

- $\delta_0(\cdot)$ is the delta mass or "spike" at zero;
- $\sigma_{\beta,l}^2$ is the prior variance of the weight $\beta_{kj}^{(l)}$;
- $\psi^{(l)} \in (0,1)$ is the prior probability for including the weight $\beta_{kj}^{(l)}$.

Model and parameter priors



Model and parameter hyper priors

$$p(\sigma_{\beta,l}^{-2}) = \text{Gamma}(a_{\beta}^{(l)}, b_{\beta}^{(l)}),$$
$$p(\psi^{(l)}) = \text{Beta}(a_{\psi}^{(l)}, b_{\psi}^{(l)}).$$

- a_β, b_β hyperparameters of Gamma hyperprior for $\sigma_{\beta,l}^{-2}$;
- $a_\psi^{(l)}, b_\psi^{(l)}$ are hyperparameters of Beta hyperprior for $\psi^{(l)} \in (0,1)$.

Inference on the model

Let:

- $\mathfrak{m} = \cup_{l,j,k}\gamma_{kj}^{(l)}$ define a model itself, i.e. which weights are switched on and which are switched off;
- $\boldsymbol{\theta}|\mathfrak{m} = \{\boldsymbol{\beta}, \phi|\mathfrak{m}\}$, where $\boldsymbol{\beta}|\mathfrak{m} = \cup_{l,j,k:\gamma_{kj}^{(l)}=1}\beta_{kj}^{(l)}$, define parameters of $\mathfrak{m}$.

Inference on the model

Let:

- $\mathfrak{m} = \cup_{l,j,k}\gamma_{kj}^{(l)}$ define a model itself, i.e. which weights are switched on and which are switched off;
- $\boldsymbol{\theta}|\mathfrak{m} = \{\boldsymbol{\beta}, \phi|\mathfrak{m}\}$, where $\boldsymbol{\beta}|\mathfrak{m} = \cup_{l,j,k:\gamma_{kj}^{(l)}=1}\beta_{kj}^{(l)}$, define parameters of $\mathfrak{m}$.

Goals:

- $p(\mathfrak{m}, \boldsymbol{\theta}|\mathbb{D})$ posterior distribution of parameters and models;
- $p(\mathfrak{m}|\mathbb{D})$ marginal posterior probabilities of the models;
- $p(\Delta|\mathbb{D})$ marginal posteriors of the parameter of interest Δ.

Inference on the model

Let:

- $\mathfrak{m} = \cup_{l,j,k}\gamma_{kj}^{(l)}$ define a model itself, i.e. which weights are switched on and which are switched off;
- $\boldsymbol{\theta}|\mathfrak{m} = \{\boldsymbol{\beta}, \phi|\mathfrak{m}\}$, where $\boldsymbol{\beta}|\mathfrak{m} = \cup_{l,j,k:\gamma_{kj}^{(l)}=1}\beta_{kj}^{(l)}$, define parameters of $\mathfrak{m}$.

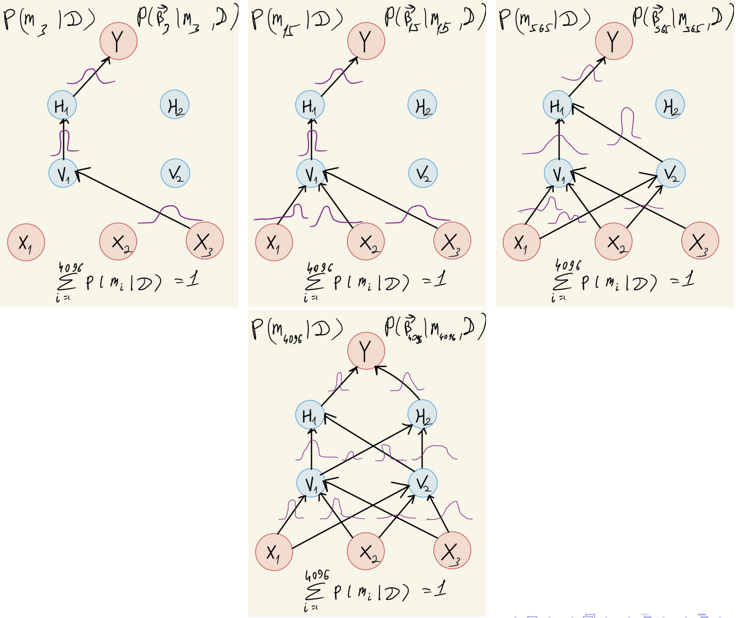
Goals:

- $p(\mathfrak{m}, \boldsymbol{\theta}|\mathbb{D})$ posterior distribution of parameters and models;
- $p(\mathfrak{m}|\mathbb{D})$ marginal posterior probabilities of the models;
- $p(\Delta|\mathbb{D})$ marginal posteriors of the parameter of interest Δ.

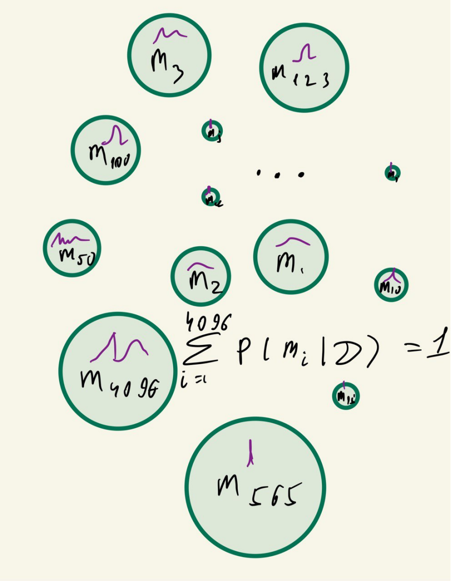
But:

- $\exists 2^q$ different models in Γ;
- q is the number of weights in the BNN, which is huge;
- Γ is not feasible to even specify.

LBBNN. Illustrations. $q = 12 \rightarrow ||\Gamma|| = 2^q = 4096$.



The model space Γ

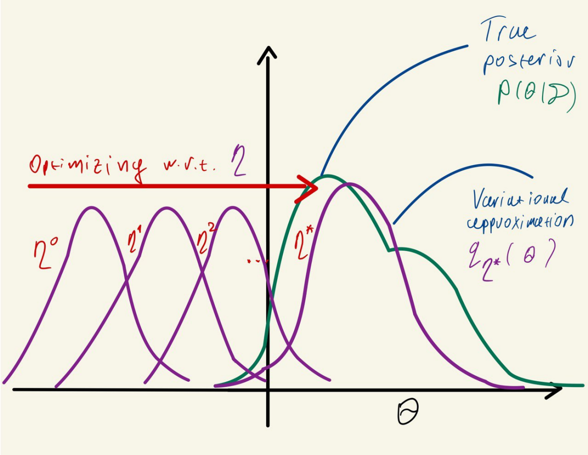


Inference possibilities

- (RJ) Markov chain Monte Carlo (exact inference) [Hubin et al., 2021];
- Laplace approximations;
- Integrated nested Laplace approximations; [Rue et al., 2009];
- **Variational inference**;
- Approximate Bayesian computation.

Variational Inference. Idea

Approximate $p(\theta|\mathbb{D})$ with $q_\eta(\theta)$ by minimizing functional divergence $\text{KL}(q_\eta(\theta)||p(\theta|\mathbb{D})) = \int_\Theta q_\eta(\theta) \log \frac{q_\eta(\theta)}{p(\theta|\mathbb{D})} d\theta \geq 0$ w.r.t. η:



Variational inference

Posterior joint distribution $p(\boldsymbol{\theta}, \mathfrak{m}|\mathbb{D})$ is approximated by combining:

- Scalable variational inference for BNN proposed by [Graves, 2011]:

$$\mathrm{KL}(q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathfrak{m})||p(\boldsymbol{\theta}, \mathfrak{m}|\mathbb{D})) = \sum_{\mathfrak{m}\in\Gamma} \int_{\Theta} q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathfrak{m}) \log \frac{q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathfrak{m})}{p(\boldsymbol{\theta}, \mathfrak{m}|\mathbb{D})} \mathrm{d}\boldsymbol{\theta} \to \min_{\eta}; \tag{4}$$

Variational inference

Posterior joint distribution $p(\boldsymbol{\theta}, \mathfrak{m}|\mathbb{D})$ is approximated by combining:

- Scalable variational inference for BNN proposed by [Graves, 2011]:

$$\mathrm{KL}(q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathfrak{m})||p(\boldsymbol{\theta}, \mathfrak{m}|\mathbb{D})) = \sum_{\mathfrak{m}\in\Gamma} \int_{\Theta} q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathfrak{m}) \log \frac{q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathfrak{m})}{p(\boldsymbol{\theta}, \mathfrak{m}|\mathbb{D})} d\boldsymbol{\theta} \rightarrow \min_{\eta}; \tag{4}$$

- A mean-field variational distribution for the joint parameter-model settings for linear models introduced by [Carbonetto et al., 2012]:

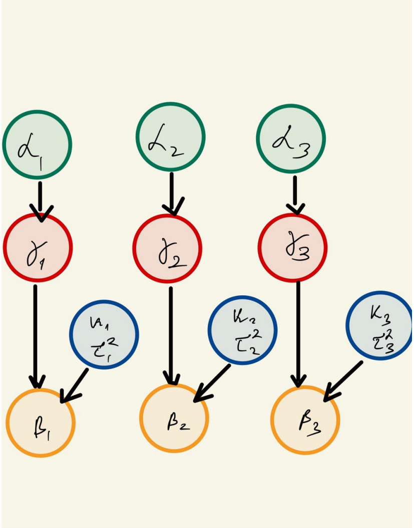
$$q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathfrak{m}) = \prod_{k,j,l} q_{\boldsymbol{\eta}_{kj}^{(l)}}(\beta_{kj}^{(l)}|\gamma_{kj}^{(l)}) q_{\boldsymbol{\eta}_{kj}^{(l)}}(\gamma_{kj}^{(l)}), \tag{5}$$

$$q_{\boldsymbol{\eta}_{kj}^{(l)}}\left(\beta_{kj}^{(l)}|\gamma_{kj}^{(l)}\right) = \gamma_{kj}^{(l)} \mathcal{N}(\kappa_{kj}^{(l)}, {\tau^2}_{kj}^{(l)}) + (1 - \gamma_{kj}^{(l)})\delta_0(\beta_{kj}^{(l)}), \tag{6}$$

$$q_{\boldsymbol{\eta}_{kj}^{(l)}}(\gamma_{kj}^{(l)}) = \mathrm{Bernoulli}(\alpha_{kj}^{(l)}). \tag{7}$$

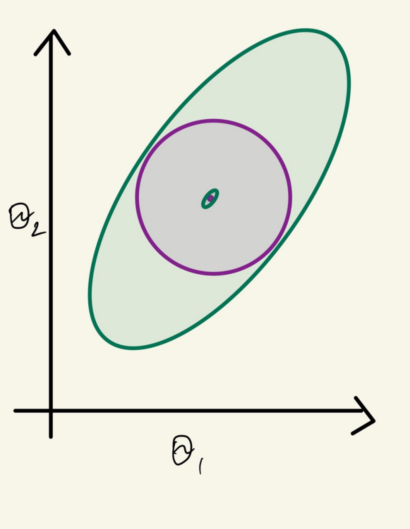
A mean-field variational distribution

A very simple approximation



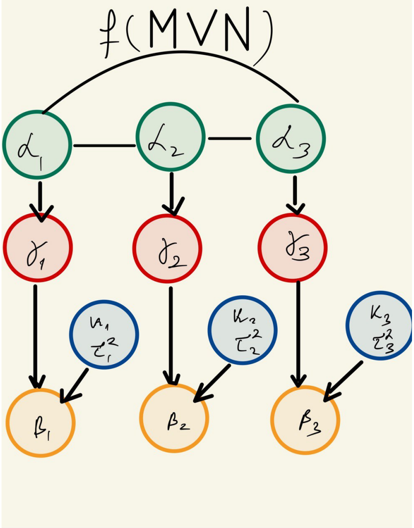
A mean-field variational distribution

But naive



Extensions of the variational distributions (MVN)

Introduce a joint Gaussian structure for transformed inclusion probabilities



Extensions of the variational distributions (MVN)

Approximation (5)-(7) assumes independence between the components, which one can argue to be unreasonable in BNNs. We proposed an extension:

$$\text{logit}(\boldsymbol{\alpha}^{(l)}) \sim \text{MVN}(\boldsymbol{\xi}^{(l)}, \Sigma^{(l)}), \tag{8}$$

$$\text{and either a full rank } \Sigma^{(l)} \text{ or a low rank representation:} \tag{9}$$

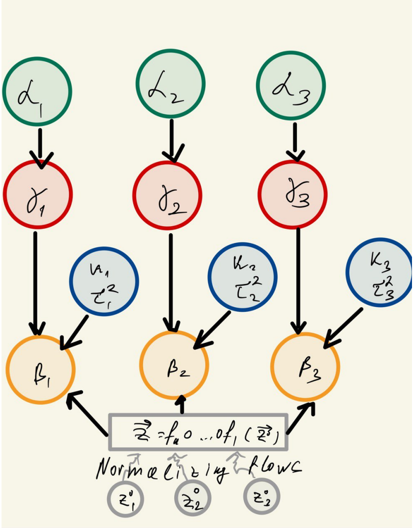
$$\Sigma^{(l)} = F^{(l)} F^{(l)^T} + D^{(l)}. \tag{10}$$

For the low-ranked representation of the covariance, e.g.:

- $F^{(l)}$ is the factor part of low-rank form of covariance matrix;
- $D^{(l)}$ is the diagonal part of low-rank form of covariance matrix.

Extensions of the variational distributions (latent $\boldsymbol{z}$)

Introduce a flexible structure for latent $\boldsymbol{z}$ through normalizing flows.



Extensions of the variational distributions (latent z)

$$q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathfrak{m}) = \prod_{j,l} q_{\boldsymbol{\eta}_j^{(l)}}(\boldsymbol{\beta}_j^{(l)}|\boldsymbol{\gamma}_j^{(l)}) q_{\boldsymbol{\eta}_j^{(l)}}(\mathfrak{m}_j^{(l)}), \tag{11}$$

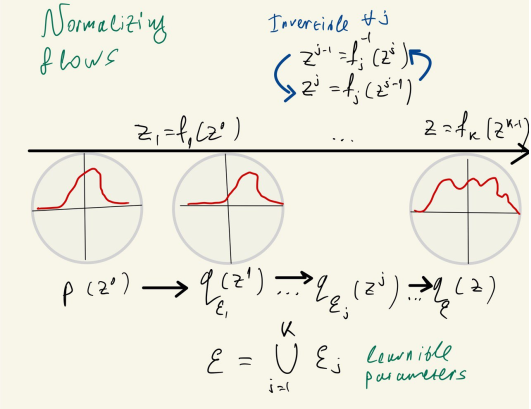
$$q_{\boldsymbol{\eta}_j^{(l)}}\left(\boldsymbol{\beta}_j^{(l)}|\mathfrak{m}_j^{(l)}\right) = \int q_{\varepsilon^{(l)}}(\boldsymbol{z}^{(l)}) \prod_k q_{\boldsymbol{\eta}_{kj}^{(l)}}\left(\beta_{kj}^{(l)}|\gamma_{kj}^{(l)}, z_k^{(l)}\right) d\boldsymbol{z}^{(l)}, \tag{12}$$

$$q_{\boldsymbol{\eta}_{kj}^{(l)}}\left(\beta_{kj}^{(l)}|\gamma_{kj}^{(l)}, z_k^{(l)}\right) = \gamma_{kj}^{(l)} \mathcal{N}(z_k^{(l)} \kappa_{kj}^{(l)}, {\tau^2}_{kj}^{(l)}) + (1 - \gamma_{kj}^{(l)})\delta_0(\beta_{kj}^{(l)}), \tag{13}$$

$$q_{\boldsymbol{\eta}_{kj}^{(l)}}(\gamma_{kj}^{(l)}) = \text{Bernoulli}(\alpha_{kj}^{(l)}). \tag{14}$$

Autoregressive flows for latent z

$\boldsymbol{z} = \mathrm{AF}(\boldsymbol{z^o}) = f_K \circ \ldots \circ f_2 \circ f_1(\boldsymbol{z^o})$, is transforming a simple $\boldsymbol{z^o} \sim q_o(\boldsymbol{z^o})$ as:



Then by transformation of random varaibles applied K times to $\boldsymbol{z^o}$:

$$\log q_\epsilon(\boldsymbol{z}_K) = \log q_o(\boldsymbol{z^o}) - \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k}{\partial_{k-1}} \right| .$$

Technical details on $z = \mathrm{AF}(z^o)$

$\boldsymbol{z} = \mathrm{AF}(\boldsymbol{z}^o)$ from [Louizos and Welling, 2017] is transforming the standard normal $\boldsymbol{z}^o$ in the following way:

$$\begin{aligned} \boldsymbol{z}^o &\sim \mathrm{MVN}(0, \mathrm{I}) \\ \boldsymbol{\mu}, \boldsymbol{s} &= \mathrm{NeuralNetwork}(\boldsymbol{z}^o) \\ \boldsymbol{\sigma} &= \mathrm{sigmoid}(\boldsymbol{s}) \\ \boldsymbol{z} &= \boldsymbol{\sigma} \odot \boldsymbol{z}^o + (1 - \boldsymbol{\sigma}) \odot \boldsymbol{\mu}, \end{aligned} \tag{15}$$

with log-determinant

$$\log \left| \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{z}^o} \right| = \sum_{i=1}^{D} \log \sigma_i. \tag{16}$$

As long as the neural network in (15) is autoregressive (i.e. output dimension z_i can only depend on input dimensions up to z_{i-1}), we get a lower triangular Jacobian and therefore this simple expression for its log-determinant.

Challenges with latent z

- $\int q_{\varepsilon^{(l)}}(\boldsymbol{z}^{(l)}) \prod_k q_{\boldsymbol{\eta}_{kj}^{(l)}}\left(\beta_{kj}^{(l)}|\gamma_{kj}^{(l)}, z_k^{(l)}\right) d\boldsymbol{z}^{(l)}$ is in general intractable;
- But $\exists q(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}})$ such that $q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathfrak{m}) = \dfrac{q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}}|\boldsymbol{z})q_{\boldsymbol{\varepsilon}}(\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}})}$;
- Then using the law of total probability

$$
\begin{aligned}
&\mathrm{KL}(q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \mathfrak{m})||p(\boldsymbol{\theta}, \mathfrak{m}|\mathbb{D})) = \\
&\sum_{\mathfrak{m}\in\mathfrak{m}} \int \int_{\Theta} q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}}|\boldsymbol{z})q_{\boldsymbol{\varepsilon}}(\boldsymbol{z}) \log \tfrac{q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}}|\boldsymbol{z})q_{\boldsymbol{\varepsilon}}(\boldsymbol{z})}{p(\boldsymbol{\theta}, \mathfrak{m}|\mathbb{D})q(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}})} \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{\theta} = \\
&\mathbb{E}_{q(\boldsymbol{\theta}, \mathfrak{m}, \boldsymbol{z})}[\mathrm{KL}\left[q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}}|\boldsymbol{z})||p(\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}})\right] + \log q_{\epsilon}(\boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}})] \leq \\
&\mathbb{E}_{q(\boldsymbol{\theta}, \mathfrak{m}, \boldsymbol{z})}[\mathrm{KL}\left[q_{\boldsymbol{\eta}}(\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}}|\boldsymbol{z})||p(\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}})\right] + \log q_{\epsilon}(\boldsymbol{z}) - \log r_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}})] = \\
&\mathrm{KL}(q(\boldsymbol{\theta}, \mathfrak{m}, \boldsymbol{z})||p(\boldsymbol{\theta}, \mathfrak{m}|\mathbb{D})r_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}}))
\end{aligned}
$$

- Here, we introduce use another approximation $r_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}})$ for $q(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}})$;
- Laplace or Gaussian approximations are possible as $r_{\boldsymbol{\lambda}}(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}})$, but we use flexible inverse flows.

Analytic forms for $\mathrm{KL}\left[q_\eta(\boldsymbol{\theta},\mathfrak{m}|z)\|p(\boldsymbol{\theta},\mathfrak{m})\right]$ and $\log q_\epsilon(z)$

- We can show that

$$\mathrm{KL}\left[q(\boldsymbol{\theta},\mathfrak{m}|\boldsymbol{z})\|p(\boldsymbol{\theta},\mathfrak{m})\right] = \sum_{kj}\alpha_{q_{kj}}\Bigg(\log\frac{\sigma_{p_{kj}}}{\sigma_{q_{kj}}} + \log\frac{\alpha_{q_{kj}}}{\alpha_{p_{kj}}} - \frac{1}{2} + \frac{\sigma^2_{q_{kj}} + (\mu_{q_{kj}} z_{K_i} - \mu_{p_{kj}})^2}{2\sigma^2_{p_{kj}}}\Bigg) + (1-\alpha_{q_{kj}})\log\frac{1-\alpha_{q_{kj}}}{1-\alpha_{p_{kj}}},$$

- We already saw that

$$\log q_\epsilon(\boldsymbol{z}) = \log q_o(\boldsymbol{z^o}) - \sum_{k=1}^{K}\log\left|\det\frac{\partial f_k}{\partial_{k-1}}\right|.$$

Inverse normalizing flows for $q(z|\boldsymbol{\theta}, \mathfrak{m})$

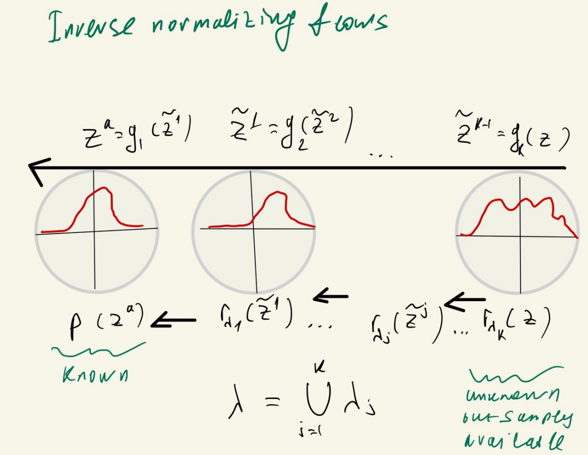
- Assume $\boldsymbol{z}^a = \mathrm{NF}(\boldsymbol{z})$ and thus $\boldsymbol{z} = \mathrm{INF}(\boldsymbol{z}^a)$;
- Assume $r_{\boldsymbol{\lambda}}(\boldsymbol{z}^{\boldsymbol{a}}|\boldsymbol{\theta}, \mathfrak{m}) = \prod_{i=1}^{D} \mathcal{N}(\tilde{\mu}_i, \tilde{\sigma}_i^2)$;
- Then by transformation formula

$$\log r\left(\boldsymbol{z}^{\boldsymbol{a}}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}}\right) = \log r\left(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}}\right) - \sum_{t=K+1}^{B} \log \left|\det \frac{\partial g_t}{\partial_{t-1}}\right|,$$

we know the deinsity of $r\left(\boldsymbol{z}^{\boldsymbol{a}}|\boldsymbol{\theta}, \boldsymbol{\mathfrak{m}}\right)$ hence

$$\log r\left(\boldsymbol{z}|\boldsymbol{\theta}, \Gamma\right) = \log r\left(\boldsymbol{z}^{\boldsymbol{a}}|\boldsymbol{\theta}, \Gamma\right) + \sum_{t=K+1}^{B} \log \left|\det \frac{\partial g_t}{\partial_{t-1}}\right|;$$

Inverse normalizing flows



Technical details on $z^a = \text{NF}(z)$

- Following [Louizos and Welling, 2017] use the following flows:

$$\tilde{\boldsymbol{\mu}} = \left(\mathbf{d}_1 \otimes \tanh\left(\mathbf{e}^T(\boldsymbol{\theta} \odot \mathfrak{m})\right)\right)(1 \odot D_{\text{out}}^{-1})$$
$$\log \tilde{\boldsymbol{\sigma}}^2 = \left(\mathbf{d}_2 \otimes \tanh\left(\mathbf{e}^T(\boldsymbol{\theta} \odot \mathfrak{m})\right)\right)(1 \odot D_{\text{out}}^{-1}).$$

- $\mathbf{d}_1$, $\mathbf{d}_2$ and $\mathbf{e}$ are trainable parameters with the same shape as $\boldsymbol{z}$;
- $\otimes$ denotes the outer product (resulting in a matrix);
- $1 \odot D_{\text{out}}^{-1}$ denotes that we compute the mean of this matrix across its rows, resulting in a vector with the same shape as $\boldsymbol{z}$.

Evidence lower bound

Proposition

Minimization of $KL\left(q_{\eta}(\boldsymbol{\theta},\mathfrak{m})||p(\boldsymbol{\theta},\mathfrak{m}|\mathbb{D})\right)$ *and maximization of the evidence (log marginal likelihood) lower bound (ELBO) are equivalent.*

$$\mathcal{L}_{VI}(\boldsymbol{\eta}) := \sum_{\mathfrak{m}\in\Gamma}\int_{\Theta} q_{\eta}(\boldsymbol{\theta},\mathfrak{m})\log p(\mathbb{D}|\boldsymbol{\theta},\mathfrak{m})d\boldsymbol{\theta} - KL\left(q_{\eta}(\boldsymbol{\theta},\mathfrak{m})||p(\boldsymbol{\theta},\mathfrak{m})\right)$$

Evidence lower bound

Proposition

Minimization of $KL\left(q_{\eta}(\boldsymbol{\theta},\mathfrak{m})||p(\boldsymbol{\theta},\mathfrak{m}|\mathbb{D})\right)$ *and maximization of the evidence (log marginal likelihood) lower bound (ELBO) are equivalent.*

$$\mathcal{L}_{VI}(\boldsymbol{\eta}) := \sum_{\mathfrak{m}\in\Gamma}\int_{\Theta} q_{\boldsymbol{\eta}}(\boldsymbol{\theta},\mathfrak{m})\log p(\mathbb{D}|\boldsymbol{\theta},\mathfrak{m})d\boldsymbol{\theta} - KL\left(q_{\boldsymbol{\eta}}(\boldsymbol{\theta},\mathfrak{m})||p(\boldsymbol{\theta},\mathfrak{m})\right)$$

Proof.

$$\mathrm{KL}(q_{\boldsymbol{\eta}}(\boldsymbol{\theta},\mathfrak{m})||p(\boldsymbol{\theta},\mathfrak{m}|\mathbb{D})) = \sum_{\mathfrak{m}\in\Gamma}\int_{\Theta} q_{\boldsymbol{\eta}}(\boldsymbol{\theta},\mathfrak{m})\log\frac{q_{\boldsymbol{\eta}}(\boldsymbol{\theta},\mathfrak{m})p(\mathbb{D})}{p(\mathbb{D}|\boldsymbol{\theta},\mathfrak{m})p(\boldsymbol{\theta},\mathfrak{m})}\mathrm{d}\boldsymbol{\theta}$$

$$= \log p(\mathbb{D}) + \sum_{\mathfrak{m}\in\Gamma}\int_{\Theta} q_{\boldsymbol{\eta}}(\boldsymbol{\theta},\mathfrak{m})\log\frac{q_{\boldsymbol{\eta}}(\boldsymbol{\theta},\mathfrak{m})}{p(\boldsymbol{\theta},\mathfrak{m})}\mathrm{d}\boldsymbol{\theta} - \sum_{\mathfrak{m}\in\Gamma}\int_{\Theta} q_{\boldsymbol{\eta}}(\boldsymbol{\theta},\mathfrak{m})\log p(\mathbb{D}|\boldsymbol{\theta},\mathfrak{m})\mathrm{d}\boldsymbol{\theta}$$

$$= \log p(\mathbb{D}) - \mathcal{L}_{VI}(\boldsymbol{\eta}) \geq 0.$$

from which the result follows. □

Relaxations of discrete $\mathfrak{m}$ with Conrete distribution

1. Consider Concrete relaxations for $\mathfrak{m}$:
   $\tilde{\gamma} = \gamma_t(\nu, \delta; \alpha) = \text{sigmoid}((\text{logit}(\alpha) - \text{logit}(\nu))/\delta), \quad \nu \sim \text{Unif}[0, 1]$;
   - Here, δ is a tuning parameter with a small value;
   - As δ goes to 0, $\tilde{\gamma}$ reduces to a Bernoulli(α) variable;
2. Consider reparametrization $\beta = \beta_t(\varepsilon; \kappa, \tau) = \kappa + \tau\varepsilon, \quad \varepsilon \sim N(0, \mathrm{I})$;
3. Consider variational approximation
   $\mathcal{L}_{VI}^{\delta}(\boldsymbol{\eta}) := \int_{\boldsymbol{\nu}} \int_{\boldsymbol{\varepsilon}} q_{\boldsymbol{\nu},\boldsymbol{\varepsilon}}(\boldsymbol{\nu}, \boldsymbol{\varepsilon})[\log p(\mathbb{D}|\boldsymbol{\beta}_t(\boldsymbol{\varepsilon}, \boldsymbol{\kappa}, \boldsymbol{\tau}), \mathfrak{m}_t(\boldsymbol{\nu}, \boldsymbol{\alpha}, \delta) - \log \frac{q_{\boldsymbol{\eta}}(\boldsymbol{\beta}_t(\boldsymbol{\varepsilon},\boldsymbol{\kappa},\boldsymbol{\tau}),\mathfrak{m}_t(\boldsymbol{\nu},\boldsymbol{\alpha},\delta))}{p(\boldsymbol{\beta}_t(\boldsymbol{\varepsilon},\boldsymbol{\kappa},\boldsymbol{\tau}),\mathfrak{m}_t(\boldsymbol{\nu},\boldsymbol{\alpha},\delta))}]d\boldsymbol{\varepsilon}d\boldsymbol{\nu}$;
4. Then
   $\nabla_{\boldsymbol{\eta}}\mathcal{L}_{VI}^{\delta}(\boldsymbol{\eta}) := \int_{\boldsymbol{\nu}} \int_{\boldsymbol{\varepsilon}} q_{\boldsymbol{\nu},\boldsymbol{\varepsilon}}(\boldsymbol{\nu}, \boldsymbol{\varepsilon}) \frac{\partial}{\partial \boldsymbol{\eta}}[\log p(\mathbb{D}|\boldsymbol{\beta}_t(\boldsymbol{\varepsilon}, \boldsymbol{\kappa}, \boldsymbol{\tau}), \mathfrak{m}_t(\boldsymbol{\nu}, \boldsymbol{\alpha}, \delta) - \log \frac{q_{\boldsymbol{\eta}}(\boldsymbol{\beta}_t(\boldsymbol{\varepsilon},\boldsymbol{\kappa},\boldsymbol{\tau}),\mathfrak{m}_t(\boldsymbol{\nu},\boldsymbol{\alpha},\delta))}{p(\boldsymbol{\beta}_t(\boldsymbol{\varepsilon},\boldsymbol{\kappa},\boldsymbol{\tau}),\mathfrak{m}_t(\boldsymbol{\nu},\boldsymbol{\alpha},\delta))}]d\boldsymbol{\varepsilon}d\boldsymbol{\nu}$.
5. And $\widetilde{\nabla}_{\boldsymbol{\eta}}\mathcal{L}_{VI}^{\delta}(\boldsymbol{\eta}) = \frac{1}{M}\sum_{m=1}^{M}\left[\frac{n}{N}\sum_{i \in S} \nabla_{\boldsymbol{\eta}} \log p(\boldsymbol{y_i}|\boldsymbol{x_i}, \boldsymbol{\beta}^{(m)}, \mathfrak{m}^{(m)}) - \nabla_{\boldsymbol{\eta}} \log \frac{q_{\boldsymbol{\eta}}(\boldsymbol{\beta}^{(m)},\mathfrak{m}^{(m)})}{p(\boldsymbol{\beta}^{(m)},\mathfrak{m}^{(m)})}\right]$. is unbiased.

Gradient estimator (unbiased for relaxed γ's)

Proposition

Assume $\left(\boldsymbol{\nu}^{(t)}, \eta^{(t)}\right) \sim q_{\boldsymbol{\nu},\boldsymbol{\varepsilon}}(\boldsymbol{\nu}, \boldsymbol{\eta})$ *for* $t \in \{1, ..., T\}$, $\boldsymbol{\beta}^{(t)} = \beta_t(\boldsymbol{\varepsilon}^{(t)}, \boldsymbol{\kappa}, \boldsymbol{\tau})$, $\tilde{\mathfrak{m}}^{(t)} = \mathfrak{m}_t(\boldsymbol{\nu}^{(t)}, \boldsymbol{\alpha}, \delta)$ *and* S *is a random subset of* $\{1, ..., n\}$ *of size* N*. Then for any* $\delta > 0$ *an unbiased estimator for the gradient of* $\mathcal{L}_{VI}^{\delta}(\boldsymbol{\eta})$ *is given by*

$$\widetilde{\nabla}_{\boldsymbol{\eta}}\mathcal{L}_{VI}^{\delta}(\boldsymbol{\eta}) = \frac{1}{T}\sum_{t=1}^{T}\left[\frac{n}{N}\sum_{i\in S}\nabla_{\boldsymbol{\eta}}\log p(\boldsymbol{y_i}|\boldsymbol{x_i}, \boldsymbol{\beta}^{(t)}, \mathfrak{m}^{(t)}) - \nabla_{\boldsymbol{\eta}}\log\frac{q_{\boldsymbol{\eta}}(\boldsymbol{\beta}^{(t)}, \mathfrak{m}^{(t)})}{p(\boldsymbol{\beta}^{(t)}, \mathfrak{m}^{(t)})}\right].$$

Local reparemetrization trick (LRT)

- If within every neruron we have independent spike and slab approximations of the weights;
- It is just a mixture of Gaussians;
- Hence, we can show that the mean of their linear combination is

$$\mathbb{E}(b_j^{(l)}) = \mathbb{E}\left[\sum_{k=1}^{p(l)} x_{ij}^{(l)} \gamma_{kj}^{(l)} \beta_{kj}^{(l)}\right] = \sum_{j=1}^{p(l)} x_{ij}^{(l)} \alpha_{kj}^{(l)} \kappa_{kj}^{(l)}$$

- And the variance is

$$\mathrm{Var}(b_j^{(l)}) = \mathrm{Var}\left[\sum_{i=k}^{p(l)} x_{ij}^{(l)} \gamma_{kj} \beta_{kj}^{(l)}\right] =$$

$$\sum_{j=1}^{p(l)} x^2{}_{ij}^{(l)} \alpha_{kj}^{(l)} (\tau_{kj}^{(l)2} + (1 - \alpha_{kj}^{(l)}) \kappa_{kj}^{(l)2});$$

- As $\alpha_{kj}^{(l)}$ converge to either 0 or 1, the mixture becomes just a unimodal Gaussian enabling direct sampling.

Model averaging

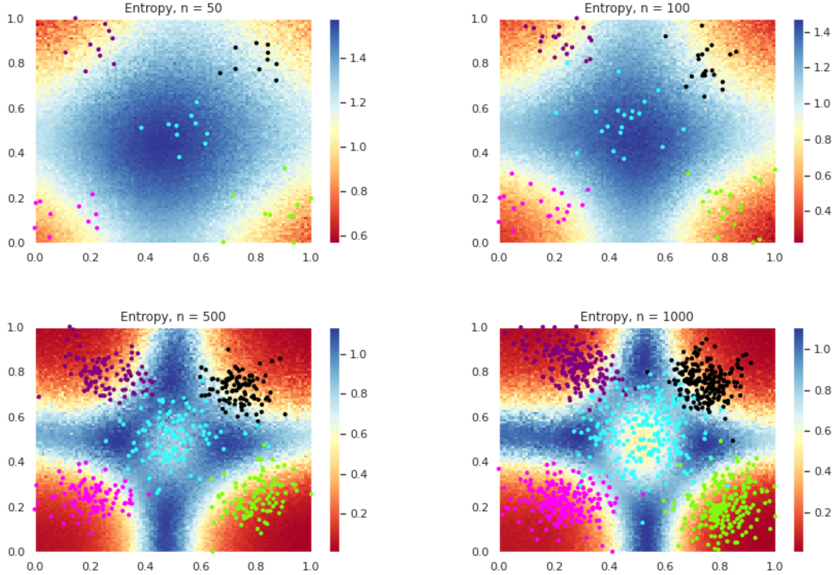
- Marginal posterior distribution of a parameter Δ (e.g. the distribution of a new observation y^* conditional on new covariates $\boldsymbol{x}^*$):

$$p(\Delta|\mathbb{D}) = \sum_{\mathfrak{m}\in\Gamma} \int_{\boldsymbol{\theta}\in\Omega_{\mathfrak{m}}} p(\Delta|\boldsymbol{\theta},\mathfrak{m},\mathbb{D})p(\boldsymbol{\theta},\mathfrak{m}|\mathbb{D})\mathrm{d}\boldsymbol{\theta}, \tag{17}$$

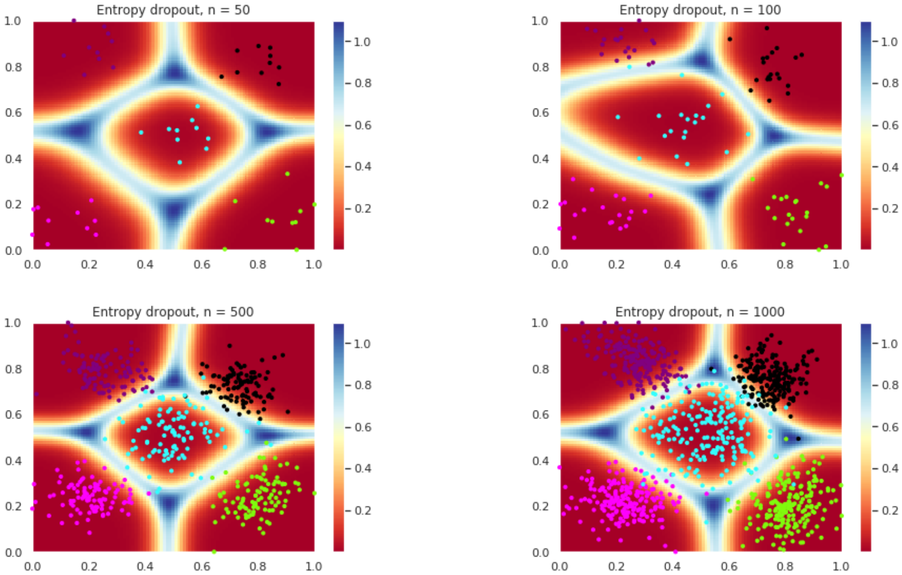
- We can now approximate it using

$$\tilde{p}(\Delta|\mathbb{D}) = \sum_{\mathfrak{m}\in\Gamma} \int_{\boldsymbol{\theta}\in\Theta_{\gamma}} p(\Delta|\boldsymbol{\theta},\mathfrak{m},\mathbb{D})q_{\eta}(\boldsymbol{\theta},\mathfrak{m})\mathrm{d}\boldsymbol{\theta}. \tag{18}$$

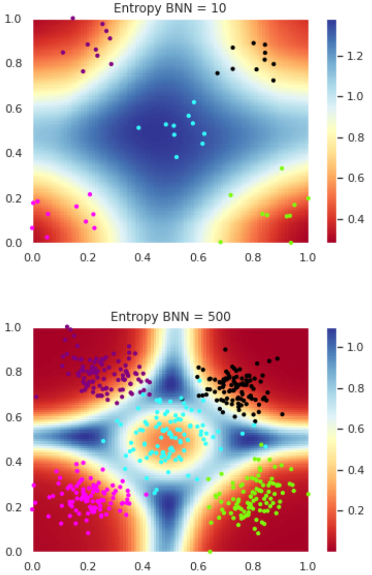
Inference with uncertainty



Do you still use dropout for uncertainty handling?



Simple dense BNNs are fine



Uncertainty FMNIST

The model for the experiments

Dense neural network with:

- ReLU activation function;
- multinomially distributed observations with 10 classes and 784 input explanatory variables (pixels);
- 2 hidden layers with 400, and 600 neurons correspondingly;
- ADAM optimizer, 250 epochs, 100 batch size.

Alternative approaches

We shall compare to:

- BNNs with a **Gaussian parameter prior**[Graves, 2011];
- BNNs with a **horseshoe prior**[Louizos et al., 2017];
- BNNs with **concrete dropout** [Gal et al., 2017].

Results. MNIST test data

	posterior mean accuracy			model averaged accuracy			
method	min	median	max	min	median	max	density
LBBNN-GP-MF	98.00	98.11	98.25	97.88	98.01	98.14	0.092
LBBNN-GP-MVN	97.60	97.80	98.00	97.60	97.80	97.90	0.180
LBBNN-GP-FLOW	**98.41**	**98.48**	**98.58**	**98.41**	**98.51**	**98.55**	**0.049**
BNN-GP-MF	98.20	98.40	98.50	98.20	98.30	98.50	1.000
BNN-GP-CMF	89.30	98.40	98.60	89.60	98.40	98.60	0.226
BNN-HP-MF	96.30	96.50	96.80	98.10	98.20	98.30	0.194

Table: Performance metrics on MNIST.

Results. FMNIST test data

	posterior mean accuracy			model averaged accuracy			
method	min	median	max	min	median	max	density
LBBNN-GP-MF	87.91	88.21	88.69	88.03	88.46	88.64	0.118
LBBNN-GP-MVN	86.80	87.10	87.50	87.50	87.70	87.90	0.156
LBBNN-GP-FLOW	**89.57**	**89.74**	**89.93**	**89.48**	**90.76**	**90.07**	**0.046**
BNN-GP-MF	88.20	88.60	88.80	89.00	89.30	89.40	1.000
BNN-GP-CMF	82.10	89.60	90.01	82.30	89.40	90.01	0.094
BNN-HP-MF	86.20	86.50	86.90	88.40	88.70	88.90	0.302

Table: Performance metrics on FMNIST

Uncertainty aware inference (0.95 treshold)

method	MNIST		FMNIST	
	decisions	accuracy	decisions	accuracy
LBBNN-GP-MF	8322	99.99	4946	99.50
LBBNN-GP-MVN	7818	**100.0**	4503	99.50
BNN-GP-MF	8477	99.99	5089	**99.70**
BNN-GP-CMF	**9581**	99.50	**8825**	94.20
BNN-HP-MF	3	100.00	181	100.0

Table: Uncerrtainty aware performance

Convolutional archetecture (LeNet-5 style)

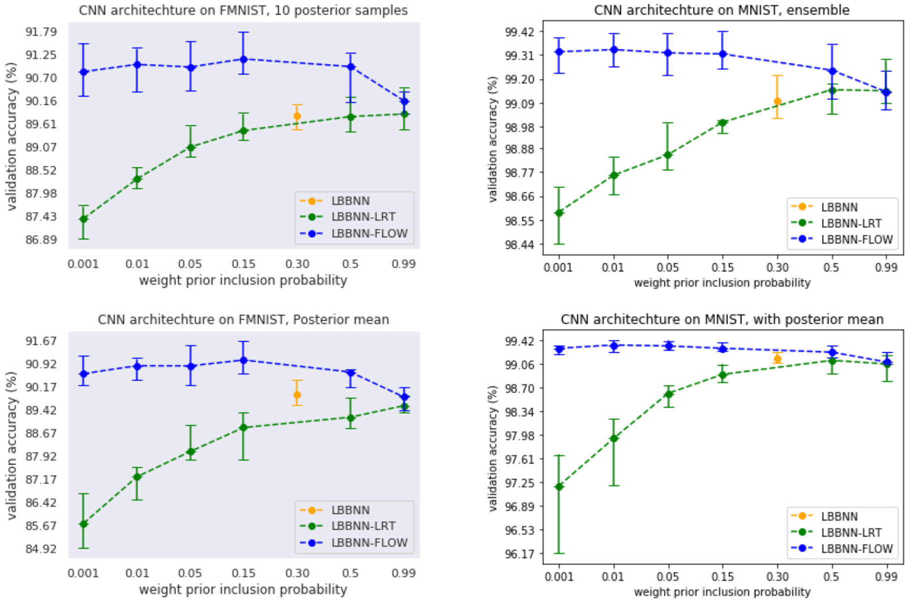
CNN	accuracy		density
	posterior mean	model averaged	
LBBNN-GP-MF	98.88	98.87	0.207
LBBNN-GP-FLOW	**99.37**	**99.31**	**0.051**

Table: CNN performance metrics on MNIST

CNN	accuracy		density
	posterior mean	model averaged	
LBBNN-GP-MF	88.30	88.24	0.209
LBBNN-GP-FLOW	**90.99**	**91.31**	**0.051**

Table: CNN performance metrics on FMNIST

CNN different priors



Simulation study from [Hubin and Storvik, 2018]

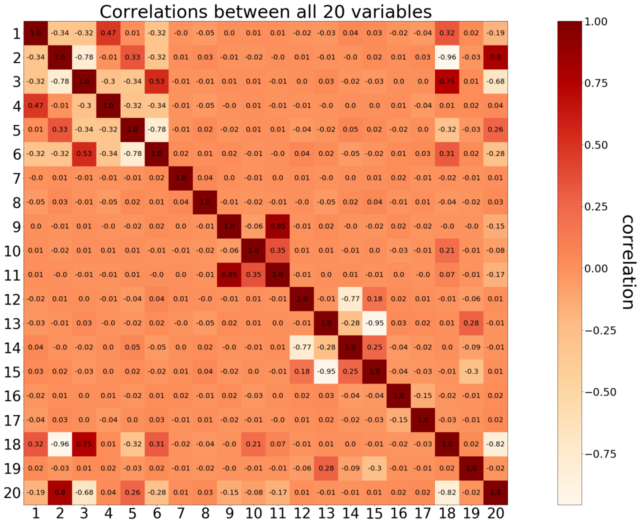
The response variable, Y, is generated as a logit transformation of the linear predictor in the following way,

$$\eta \sim \mathcal{N}(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{\beta}, 0.5)$$
$$Y \sim \text{Bernoulli}\left(\frac{\exp(\eta)}{1 + \exp(\eta)}\right)$$

with $n = 2000$ and

$$\boldsymbol{\beta} = (-4, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1.2, 0, 37.1, 0, 0, 50, -0.00005, 10, 3, 0).$$

Correlation structure



Simulation study cont.

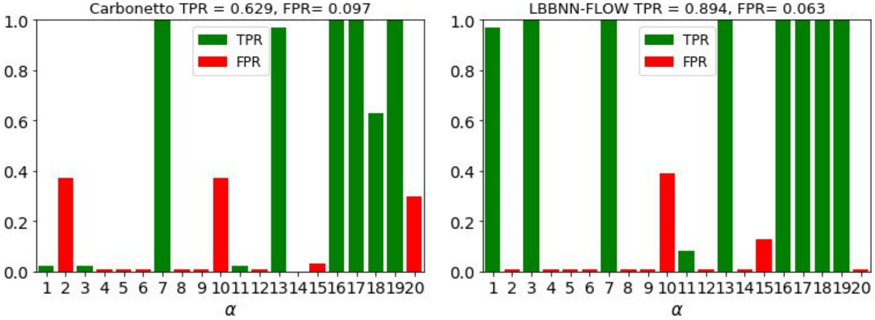


Figure: Bar plots showing true positive rate and false positive rate for the 20 different covariates on the **VARBVS, [Carbonetto et al., 2012]** (left) and **LBBNN-GP-FLOW** (right). The bars where the true weights are non-zero are colored green, and the bars where the true weights are zero are colored red.

Concluding remarks

- We develop scalable joint model-parameter approximate inference approaches in the class of BNNs;
- It allows to perform Bayesian model selection and model averaging;
- The resulting model selection often leads to drastic sparsification of BNNs with no loss of predictive power;
- Furthermore, both model selection and model averaging within our approach allow for accurate and robust handling of predictive uncertainty;
- However, the VB approach can generally be extremely biased and the ways to reduce the bias must be studied further.

References I

Carbonetto, P., Stephens, M., et al. (2012).
Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies.
Bayesian analysis, 7(1):73–108.

Gal, Y., Hron, J., and Kendall, A. (2017).
Concrete dropout.
In *Advances in Neural Information Processing Systems*, pages 3581–3590.

Graves, A. (2011).
Practical variational inference for neural networks.
In *Advances in Neural Information Processing Systems*, pages 2348–2356.

References II

Hubin, A. and Storvik, G. (2018).
Mode jumping MCMC for Bayesian variable selection in GLMM.
Computational Statistics & Data Analysis.

Hubin, A., Storvik, G., and Frommlet, F. (2021).
Flexible bayesian nonlinear model configuration.
Journal of Artificial Intelligence Research, 72:901–942.

Louizos, C., Ullrich, K., and Welling, M. (2017).
Bayesian compression for deep learning.
In *Advances in Neural Information Processing Systems*, pages 3288–3298.

References III

Louizos, C. and Welling, M. (2017).
Multiplicative normalizing flows for variational bayesian neural networks.
In *International Conference on Machine Learning*, pages 2218–2227. PMLR.

Rue, H., Martino, S., and Chopin, N. (2009).
Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations.
Journal of the Royal Statistical Sosciety, 71(2):319–392.