Good Initializations of Variational Bayes for Deep Models: Supplementary Material

Simone Rossi [1] **Pietro Michiardi** [1] **Maurizio Filippone** [1]

A. Full derivation of variational lower bound

The following derivation shows that minimizing the KL divergence between the approximate posterior and the true posterior is equivalent to minimizing the so-called Negative Expected Lower Bound (NELBO):

$$\begin{aligned}
\text{KL}\left[q_\theta(\mathbf{W})||p(\mathbf{W}|X,Y)\right] &= \mathbb{E}_{q_\theta}\left[\log \frac{q_\theta(\mathbf{W})}{p(\mathbf{W}|X,Y)}\right] = \\
&= \mathbb{E}_{q_\theta}\left[\log q_\theta(\mathbf{W}) - \log p(\mathbf{W}|X,Y)\right] = \\
&= \mathbb{E}_{q_\theta}\left[-\log p(Y|X,\mathbf{W})\right] + \mathbb{E}_{q_\theta}\left[\log q_\theta(\mathbf{W}) - \log p(\mathbf{W})\right] + \log p(Y|X) = \\
&= \text{NLL} + \text{KL}\left[q_\theta(\mathbf{W})||p(\mathbf{W})\right] + \log p(Y|X) = \\
&= \text{NELBO} + \log p(Y|X) \qquad (1)
\end{aligned}$$

This also shows that when the approximate posterior is exactly equal to the true posterior, the NELBO is equal to the negative log-marginal likelihood.

B. Bayesian linear regression

In this section, we derive Bayesian linear regression, which is the core model used in our proposed initialization I-BLM. Denote by X the $n \times d$ matrix containing n input vectors $\mathbf{x}_i \in \mathbb{R}^d$, and let Y be the set consisting of the corresponding multivariate labels $\mathbf{y}_i$. In Bayesian linear regression we introduce a set of latent variables that we compute as a linear combination of the input through a set of weights, and we express the likelihood and the prior on the parameters as follows:

$$p(Y|X,W,L) = \prod_i p(Y_{\cdot i}|XW_{\cdot i},\lambda) = \prod_i \mathcal{N}(Y_{\cdot i}|XW_{\cdot i},L) \qquad (2)$$

and

$$p(W|\Lambda) = \prod_i p(W_{\cdot i}) = \mathcal{N}(W_{\cdot i}|\mathbf{0},\Lambda) \qquad (3)$$

The posterior of this model is:

$$p(W|Y,X,L,\Lambda) \propto \prod_i \mathcal{N}(Y_{\cdot i}|XW_{\cdot i},L)\mathcal{N}(W_{\cdot i}|\mathbf{0},\Lambda) \qquad (4)$$

which implies that the posterior factorizes across the columns of W, with factors

$$p(W_{\cdot i}|Y,X,L,\Lambda) = \mathcal{N}(W_{\cdot i}|\Sigma_i X^\top L^{-1} Y_{\cdot i},\Sigma_i) \qquad (5)$$

with $\Sigma_i = (\Lambda^{-1} + X^\top L^{-1} X)^{-1}$. Similarly, the marginal likelihood factorizes as the product of the following factors

$$p(Y_{\cdot i}|X,L,\Lambda) = \mathcal{N}(Y_{\cdot i}|\mathbf{0}, L + X\Lambda X^\top) \qquad (6)$$

[1]Department of Data Science, EURECOM, France. Correspondence to: Simone Rossi <simone.rossi@eurecom.fr>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

C. Heteroscedastic Bayesian linear regression

In the main paper we discuss how I-BLM can be extended to handle classification problems, by borrowing ideas from Milios et al. (2018) where it shown how to transform classification problems into regression. Here we extend Bayesian linear regression to the heteroscedastic case where $L = \text{diag}(\boldsymbol{\sigma}^2)$ and $\Lambda = \alpha I$. These yield

$$p(W_{\cdot i}|Y, X, \boldsymbol{\sigma}^2, \alpha) = \mathcal{N}(W_{\cdot i}|\mu_i, \Sigma_i) \quad \text{with} \tag{7}$$

$$\mu_i = \Sigma_i X^\top \text{diag}(\boldsymbol{\sigma}^{-2}) Y_{\cdot i} \tag{8}$$

$$\Sigma_i = \alpha \left(I + \alpha\, X^\top \text{diag}(\boldsymbol{\sigma}^{-2}) X\right)^{-1} \tag{9}$$

and

$$p(Y_{\cdot i}|X, \boldsymbol{\sigma}^2, \alpha) = \mathcal{N}(Y_{\cdot i}|\mathbf{0}, \text{diag}(\boldsymbol{\sigma}^2) + \alpha\, XX^\top) \tag{10}$$

The expression for the marginal likelihood is computationally unconvenient due to the need to deal with an $n \times n$ matrix. We can use Woodbury identities[1] to express this calculation using Σ_i. In particular,

$$\log[p(Y_{\cdot i}|X, \boldsymbol{\sigma}^2, \alpha)] = -\frac{1}{2}\log\left|\text{diag}(\boldsymbol{\sigma}^2) + \alpha\, XX^\top\right| - \frac{1}{2} Y_{\cdot i}^\top \left(\text{diag}(\boldsymbol{\sigma}^2) + \alpha\, XX^\top\right)^{-1} Y_{\cdot i} + \text{const.} \tag{11}$$

Using Woodbury identites, we can rewrite the algebraic operations as follows:

$$\begin{aligned} \log\left|\text{diag}(\boldsymbol{\sigma}^2) + \alpha\, XX^\top\right| &= \log\left|\text{diag}(\boldsymbol{\sigma}^2)\right| + \log\left|I + \alpha\, \text{diag}(\boldsymbol{\sigma}^{-2}) XX^\top\right| \\ &= \sum_j \log \sigma_j^2 + \log\left|I + \alpha\, X^\top \text{diag}(\boldsymbol{\sigma}^{-2}) X\right| \end{aligned} \tag{12}$$

and

$$\left(\text{diag}(\boldsymbol{\sigma}^2) + \alpha\, XX^\top\right)^{-1} = \text{diag}(\boldsymbol{\sigma}^{-2}) - \alpha\, \text{diag}(\boldsymbol{\sigma}^{-2}) X \left(I + \alpha\, X^\top \text{diag}(\boldsymbol{\sigma}^{-2}) X\right)^{-1} X^\top \text{diag}(\boldsymbol{\sigma}^{-2}) \tag{13}$$

So, wrapping up, we can express all quantities of interest through:

$$\Sigma_i^{-1} = I + \alpha\, X^\top \text{diag}(\boldsymbol{\sigma}^{-2}) X \tag{14}$$

The marginal likelihood becomes:

$$\begin{aligned} \log[p(Y_{\cdot i}|X, \boldsymbol{\sigma}^2)] &= -\frac{1}{2}\left(\sum_j \log \sigma_j^2 + \log\left|\Sigma_i^{-1}\right|\right) \\ &\quad -\frac{1}{2} Y_{\cdot i}^\top \left(\text{diag}(\boldsymbol{\sigma}^{-2}) - \alpha\, \text{diag}(\boldsymbol{\sigma}^{-2}) X \Sigma_i X^\top \text{diag}(\boldsymbol{\sigma}^{-2})\right)^{-1} Y_{\cdot i} + \text{const.} \end{aligned} \tag{15}$$

If we factorize $\Sigma_i^{-1} = QQ^\top$, we obtain:

$$\log[p(Y_{\cdot i}|X, \boldsymbol{\sigma}^2)] = -\frac{1}{2}\left(\sum_j \log(\sigma_j^2) + \sum_k 2\log(Q_{kk})\right) - \frac{1}{2} Y_{\cdot i}^\top \tilde{Y}_{\cdot i} + \frac{\alpha}{2} \tilde{Y}_{\cdot i}^\top X Q^{-\top} Q^{-1} X^\top \tilde{Y}_{\cdot i} + \text{const.} \tag{16}$$

where $\tilde{Y}_{\cdot i} = \text{diag}(\boldsymbol{\sigma}^{-2}) Y_{\cdot i}$

Predictions follow from the same identities as before - looking at the predicted latent process, we have

$$p(f_{*i}|X, Y, \mathbf{x}_*) = \int p(f_{*i}|W, \mathbf{x}_*) p(W|X, Y) dW \tag{17}$$

We can again remove the dependence from the dimensions of W that do not affect the prediction for the ith function as

$$p(f_{*i}|X, Y, \mathbf{x}_*) = \int p(f_{*i}|W_{\cdot i}, \mathbf{x}_*) p(W_{\cdot i}|X, Y) dW \tag{18}$$

Now:

$$p(f_{*i}|W_{\cdot i}, \mathbf{x}_*) = \mathcal{N}(f_{*i}|\mathbf{x}_*^\top W_{\cdot i}, 0) \qquad \text{and} \qquad p(W_{\cdot i}|X, Y) = \mathcal{N}(W_{\cdot i}|\mu_i, \Sigma_i) \tag{19}$$

giving

$$p(f_{*i}|X, Y, \mathbf{x}_*) = \mathcal{N}(f_{*i}|\mathbf{x}_*^\top \mu_i, \mathbf{x}_*^\top \Sigma_i \mathbf{x}_*) \tag{20}$$

[1] $|I + B^\top C| = |I + CB^\top|$ and $(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$

D. Full derivation of fully factorized Gaussian posterior approximation to Bayesian linear regression posterior

For simplicity of notation, let $\mathbf{w}$ be the parameters of interest in Bayesian linear regression for a given output $\mathbf{y} = Y_{\cdot i}$. We can formulate the problem of obtaining the best approximate factorized posterior of a Bayeian linear model as a minimization of the KL divergence between $q(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}, \text{diag}(\mathbf{s}^2))$ and the actual posterior $p(\mathbf{w}|X, \mathbf{y})$. The expression of the KL divergence between multivariate Gaussians $p_0 = \mathcal{N}(W|\boldsymbol{\mu}_0, \Sigma_0)$ and $p_1 = \mathcal{N}(W|\boldsymbol{\mu}_1, \Sigma_1)$ is as follows:

$$\text{KL}[p_0||p_1] = \frac{1}{2}\text{Tr}(\Sigma_1^{-1}\Sigma_0) + \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \Sigma_1^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - \frac{D}{2} + \frac{1}{2}\log\left(\frac{\det\Sigma_1}{\det\Sigma_0}\right) \tag{21}$$

The KL divergence is not symmetric, so the order in which we take this matters. In case we consider $\text{KL}[p(\mathbf{w}|X, \mathbf{y})||q(\mathbf{w})]$, the expression becomes:

$$\text{KL}[p(\mathbf{w}|X, \mathbf{y})||q(\mathbf{w})] = \frac{1}{2}\text{Tr}(\text{diag}(\mathbf{s}^2)^{-1}\Sigma) + \frac{1}{2}(\mathbf{m} - \boldsymbol{\mu})^\top \text{diag}(\mathbf{s}^2)^{-1}(\mathbf{m} - \boldsymbol{\mu}) - \frac{D}{2} + \frac{1}{2}\log\left(\frac{\prod_i s_i^2}{\det\Sigma}\right) \tag{22}$$

It is a simple matter to show that the optimal mean $\mathbf{m}$ is $\boldsymbol{\mu}$ as $\mathbf{m}$ appears only in the quadratic form which is clearly minimized when $\mathbf{m} = \boldsymbol{\mu}$. For the variances $\mathbf{s}^2$, we need to take the derivative of the KL divergence and set it to zero:

$$\frac{\partial \text{KL}[p(\mathbf{w}|X, \mathbf{y})||q(\mathbf{w})]}{\partial s_i^2} = \frac{1}{2}\frac{\partial \text{Tr}(\text{diag}(\mathbf{s}^2)^{-1}\Sigma)}{\partial s_i^2} + \frac{1}{2}\frac{\partial \sum_i \log s_i^2}{\partial s_i^2} = 0 \tag{23}$$

Rewriting the trace term as the sum of the Hadamrd product of the matrices in the product $\sum_{ij}(\text{diag}(\mathbf{s}^2)^{-1} \odot \Sigma)_{ij} = \sum_i \Sigma_{ii}/s_i^2$, this yields

$$\frac{\partial \text{KL}[p(\mathbf{w}|X, \mathbf{y})||q(\mathbf{w})]}{\partial s_i^2} = \frac{1}{2}\frac{\partial \Sigma_{ii}/s_i^2)}{\partial s_i^2} + \frac{1}{2}\frac{\partial \log s_i^2}{\partial s_i^2} = 0 \tag{24}$$

This results in $s_i^2 = \Sigma_{ii}$, which is the simplest way to approximate the correlated posterior over $\mathbf{w}$ but it is going to inflate the variance in case of strong correlations.

In case we consider $\text{KL}[q(\mathbf{w})||p(\mathbf{w}|X, \mathbf{y})]$, the expression of the KL becomes:

$$\text{KL}[q(\mathbf{w})||p(\mathbf{w}|X, \mathbf{y})] = \frac{1}{2}\text{Tr}(\Sigma^{-1}\text{diag}(\mathbf{s}^2)) + \frac{1}{2}(\mathbf{m} - \boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{m} - \boldsymbol{\mu}) - \frac{D}{2} + \frac{1}{2}\log\left(\frac{\det\Sigma}{\prod_i s_i^2}\right) \tag{25}$$

Again, the optimal mean $\mathbf{m}$ is $\boldsymbol{\mu}$. For the variances $\mathbf{s}^2$, we need to take the derivative of the KL divergence and set it to zero:

$$\frac{\partial \text{KL}[q(\mathbf{w})||p(\mathbf{w}|X, \mathbf{y})]}{\partial s_i^2} = \frac{1}{2}\frac{\partial \text{Tr}(\Sigma^{-1}\text{diag}(\mathbf{s}^2))}{\partial s_i^2} - \frac{1}{2}\frac{\partial \sum_i \log s_i^2}{\partial s_i^2} = 0 \tag{26}$$

Rewriting the trace term as the sum of the Hadamrd product of the matrices in the product $\sum_{ij}(\Sigma^{-1} \odot \text{diag}(\mathbf{s}^2))_{ij} = \sum_i s_i^2\Sigma_{ii}^{-1}$, this yields

$$\frac{\partial \text{KL}[q(\mathbf{w})||p(\mathbf{w}|X, \mathbf{y})]}{\partial s_i^2} = \frac{1}{2}\frac{\partial s_i^2\Sigma_{ii}^{-1})}{\partial s_i^2} - \frac{1}{2}\frac{\partial \log s_i^2}{\partial s_i^2} = 0 \tag{27}$$

This results in $(s_i^2)^{-1} = \Sigma_{ii}^{-1}$. This approximation has the opposite effect of underestimating the variance for each variable.

E. Extended results

E.1. Experimental setup

The experimental results in the main paper and in the supplement have been carried out using Zoe-Analytics (Pace et al., 2017) running on a cluster of computers featuring servers equipped with NVIDIA Tesla P100 GPUs.

E.2. Toy example

With this simple example we want once more illustrate how I-BLM works and how it can speed up the convergence of SVI. We set up a regression problem considering the function $f(x) = \sin(x) + \sin(x/2) + \sin(x/3) - \sin(x/4)$ corrupted by noise $\varepsilon \sim \mathcal{N}(0, \exp(-2))$, with x sampled uniformly in the interval $[-10, 10]$. Figure 1 reports the output of a 4-layer DNN after different initializations. The figure shows that I-BLM obtains a sensible initialization compared to the competitors.

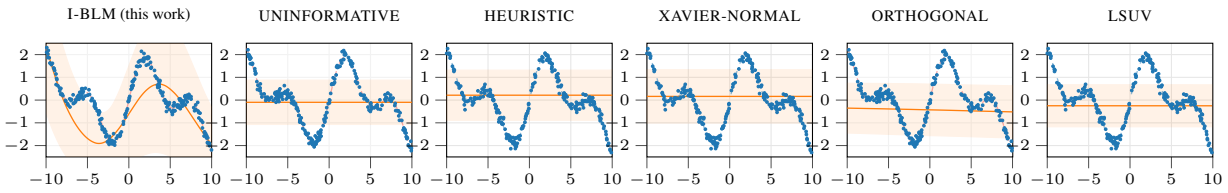

Figure 1: Predictions after initialization using our proposal and all the other competitive methods.

E.3. Regression with shallow architecture

TEST RMSE

	I-BLM	UNINFORMATIVE	HEURISTIC	XAVIER	ORTHOGONAL	LSUV
POWERPLANT	$\mathbf{0.2427 \pm 0.006}$	0.2452 ± 0.007	0.2436 ± 0.008	0.2427 ± 0.008	0.2439 ± 0.008	0.2438 ± 0.007
PROTEIN	$\mathbf{0.6831 \pm 0.004}$	0.7135 ± 0.008	0.7020 ± 0.009	0.6952 ± 0.011	0.7315 ± 0.016	0.7356 ± 0.006

TEST MNLL

	I-BLM	UNINFORMATIVE	HEURISTIC	XAVIER	ORTHOGONAL	LSUV
POWERPLANT	$\mathbf{-0.7647 \pm 0.012}$	-0.7607 ± 0.013	-0.7622 ± 0.014	-0.7641 ± 0.013	-0.7623 ± 0.013	-0.7623 ± 0.012
PROTEIN	$\mathbf{0.7510 \pm 0.021}$	0.8980 ± 0.040	0.8376 ± 0.046	0.8047 ± 0.055	0.9878 ± 0.083	1.0081 ± 0.033

E.4. Regression with deep architecture

TEST RMSE

	I-BLM	UNINFORMATIVE	HEURISTIC	XAVIER	ORTHOGONAL	LSUV
POWERPLANT	0.2472 ± 0.003	0.2476 ± 0.005	$\mathbf{0.2462 \pm 0.005}$	0.2658 ± 0.030	0.2467 ± 0.005	0.2774 ± 0.026
PROTEIN	$\mathbf{0.6683 \pm 0.007}$	0.7170 ± 0.013	0.6899 ± 0.011	0.6821 ± 0.007	0.6982 ± 0.014	0.7033 ± 0.011

TEST MNLL

	I-BLM	UNINFORMATIVE	HEURISTIC	XAVIER	ORTHOGONAL	LSUV
POWERPLANT	$\mathbf{-0.7455 \pm 0.008}$	-0.7420 ± 0.008	-0.7455 ± 0.009	-0.7007 ± 0.070	-0.7450 ± 0.010	-0.6677 ± 0.065
PROTEIN	$\mathbf{0.6922 \pm 0.035}$	0.9326 ± 0.066	0.7884 ± 0.055	0.7540 ± 0.033	0.8280 ± 0.072	0.8587 ± 0.040

E.5. Classification with shallow architecture

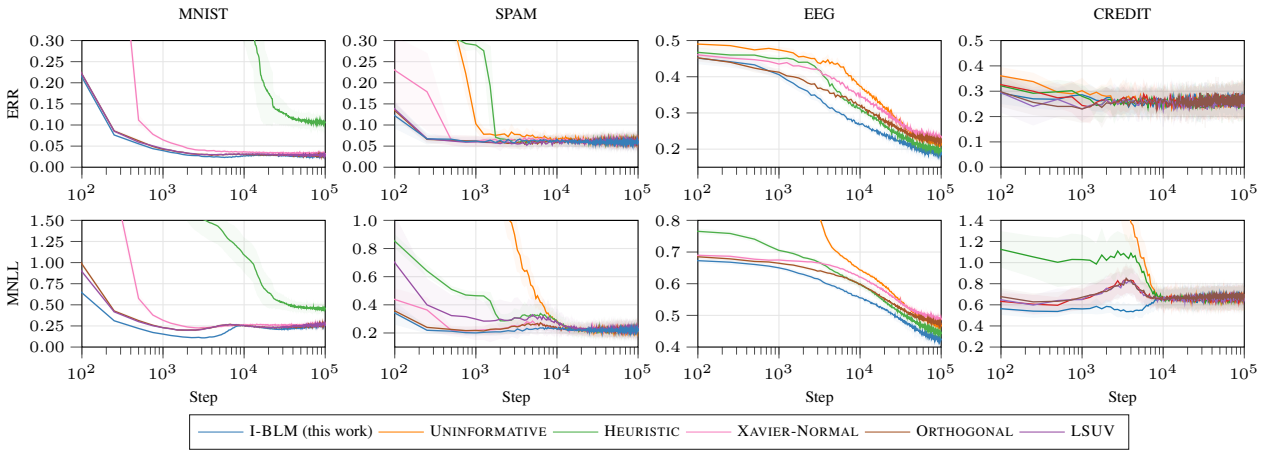

Figure 2: Progression of test ERROR RATE and test MNLL over training iterations for different initialization strategies on four classification datasets.

TEST ERROR RATE

	I-BLM	UNINFORMATIVE	HEURISTIC	XAVIER	ORTHOGONAL	LSUV
SPAM	$\mathbf{0.0594 \pm 0.013}$	0.0620 ± 0.008	0.0624 ± 0.011	0.0620 ± 0.012	0.0611 ± 0.012	0.0598 ± 0.014
EEG	$\mathbf{0.1855 \pm 0.015}$	0.1929 ± 0.009	0.2221 ± 0.009	0.2137 ± 0.008	0.2335 ± 0.007	NC
CREDIT	0.2680 ± 0.027	0.2679 ± 0.044	$\mathbf{0.2480 \pm 0.071}$	0.2519 ± 0.033	0.2539 ± 0.032	0.2580 ± 0.050
MNIST	$\mathbf{0.0253 \pm 7e{-4}}$	NC	0.1046 ± 0.014	0.0315 ± 0.001	0.0275 ± 0.001	0.0291 ± 0.002

TEST MNLL

	I-BLM	UNINFORMATIVE	HEURISTIC	XAVIER	ORTHOGONAL	LSUV
SPAM	0.229 ± 0.034	$\mathbf{0.213 \pm 0.030}$	0.228 ± 0.043	0.228 ± 0.048	0.225 ± 0.053	0.228 ± 0.050
EEG	$\mathbf{0.4218 \pm 0.020}$	0.4668 ± 0.008	0.4411 ± 0.006	0.4866 ± 0.010	0.4728 ± 0.010	NC
CREDIT	0.6759 ± 0.084	$\mathbf{0.6597 \pm 0.101}$	0.6605 ± 0.111	0.6616 ± 0.105	0.6662 ± 0.076	0.6739 ± 0.069
MNIST	0.2655 ± 0.015	NC	0.4497 ± 0.039	0.2724 ± 0.020	$\mathbf{0.2643 \pm 0.017}$	0.2744 ± 0.014

E.6. Classification with deep architecture

TEST ERROR RATE

	I-BLM	UNINFORMATIVE	HEURISTIC	XAVIER	ORTHOGONAL	LSUV
MNIST	$\mathbf{0.0356 \pm 0.003}$	0.0390 ± 0.003	0.0400 ± 0.003	0.0411 ± 0.002	0.0396 ± 0.002	0.0373 ± 0.001
EEG	$\mathbf{0.0673 \pm 0.008}$	0.1283 ± 0.009	0.1119 ± 0.008	0.0894 ± 0.003	0.1216 ± 0.002	NC
CREDIT	$\mathbf{0.2700 \pm 0.024}$	0.2975 ± 0.059	0.2824 ± 0.058	0.2833 ± 0.022	0.3145 ± 0.051	0.2758 ± 0.022
SPAM	0.0566 ± 0.021	0.0611 ± 0.008	0.0585 ± 0.017	0.0534 ± 0.018	$\mathbf{0.0514 \pm 0.013}$	0.0611 ± 0.013

TEST MNLL

	I-BLM	UNINFORMATIVE	HEURISTIC	XAVIER	ORTHOGONAL	LSUV
MNIST	$\mathbf{0.1692 \pm 0.007}$	0.1847 ± 0.002	0.1799 ± 0.009	0.1912 ± 0.011	0.1822 ± 0.005	0.1723 ± 0.005
EEG	$\mathbf{0.4222 \pm 0.054}$	1.2515 ± 0.352	0.8136 ± 0.123	0.6273 ± 0.130	0.9366 ± 0.097	NC
CREDIT	2.6555 ± 0.521	3.2836 ± 0.704	3.1268 ± 0.678	2.7015 ± 0.665	2.6482 ± 0.231	$\mathbf{2.5422 \pm 0.236}$
SPAM	$\mathbf{0.7021 \pm 0.218}$	1.1098 ± 0.271	1.0458 ± 0.517	1.0682 ± 0.347	0.8176 ± 0.337	1.1682 ± 0.486

E.7. Convolutional neural networks

TEST ERROR RATE

	I-BLM	UNINFORMATIVE	HEURISTIC	XAVIER	ORTHOGONAL	LSUV
MNIST	**0.0087**	NC	NC	NC	0.0098	0.0113
CIFAR10	**0.3499**	NC	NC	NC	0.3784	0.3846

TEST MNLL

	I-BLM	UNINFORMATIVE	HEURISTIC	XAVIER	ORTHOGONAL	LSUV
MNIST	**0.0345**	NC	NC	NC	0.0377	0.0421
CIFAR10	**1.0683**	NC	NC	NC	1.1270	1.1428

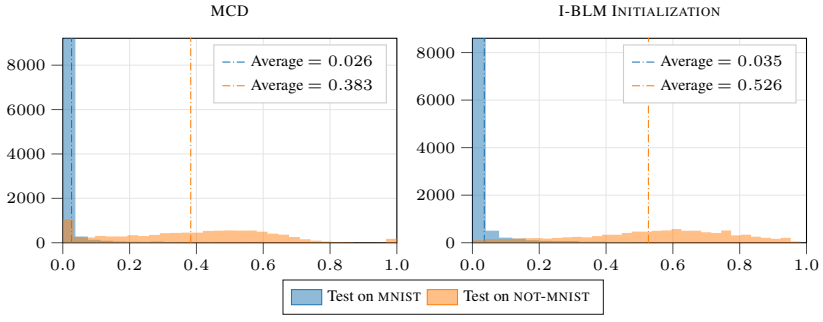

Figure 3: Entropy distribution while testing on MNIST and NOT-MNIST (higher average entropy on NOT-MNIST means better uncertainty estimation).

E.8. Uncertainty estimation

One of the advantages of Bayesian inference is the possibility to reason about uncertainty. With this experiment, we aim to demonstrate that SVI with a Gaussian approximate posterior is competitive with MCD in capturing uncertainty in predictions. To show this, we focus on a CNN with the LENET-5 architecture. We run MCD and SVI with a Gaussian approximate posterior with the proposed initialization on MNIST. At test time, we carry out predictions on both MNIST and NOT-MNIST; the latter is a dataset equivalent to MNIST in input dimensions ($1 \times 28 \times 28$) and number of classes, but it represents letters rather than numbers[2]. This experimental setup is often used to check that the entropy of the predictions on NOT-MNIST are actually higher than the entropy of the predictions on MNIST. We report the entropy of the prediction on MNIST and NOT-MNIST in Figure 3. MCD and SVI behave similarly on MNIST, but on NOT-MNIST the the histogram of the entropy indicates that SVI yields a slightly higher uncertainty compared to MCD.

E.9. Calibration

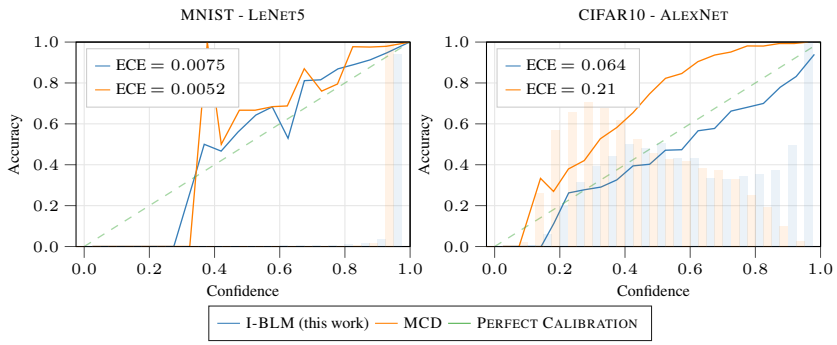

Figure 4: Comparison of reliability diagrams and ECE between I-BLM and MCD on MNIST (**left**) and CIFAR10 (**right**).

Calibration of uncertainty is an important performance metric that one should take into account for comparing classification models (Flach, 2016; Guo et al., 2017). *Reliability Diagrams* and the *Expected Calibration Error* are standard methods to empirically estimate the calibration uncertainty. *Reliability Diagrams* are a visualization tool where sample accuracy is plotted as function of confidence (DeGroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005). For a perfectly calibrated model, the diagram follows the identity function. *Expected Calibration Error* (or ECE) represents a summary statistic of the calibration (Naeini et al., 2015). Figure 4 shows the reliability diagrams and the ECE for LENET-5 trained on MNIST and for ALEXNET trained on CIFAR10. Even tough they show similar properties on MNIST, with ALEXNET on CIFAR10, SVI initialized with I-BLM improves the calibration of uncertainty up to 3.5 times over MCD.

E.10. KL regularization policy for Gaussian SVI

The KL regularization term in the variational objective severely penalizes training of over-parameterized model. With a sensible initialization of this kind of model, the approximate posterior is drastically different from a spherical Gaussian prior and the variational objective is majorly dominated by the regularization term rather than the reconstruction likelihood. To deal with such issue, we propose and implement a simple policy to gradually include the KL term in the NELBO. Given the generic expression for the NELBO, we modify the lower bound as follow:

$$\text{NELBO} = \text{NLL} + \lambda \text{KL}\left[q_{\boldsymbol{\theta}}(\mathbf{W})||p(\mathbf{W})\right] \quad \text{where} \quad \lambda = \gamma\left(1 + \exp(-\alpha(\text{iter} - \beta))\right)^{-1},$$

This way, we start the optimization of the NELBO with low regularization, and progressively increase it throughout the optimization. For the experiment on VGG16, we used $\alpha = 2 \cdot 10^{-3}$, $\beta = 2.5 \cdot 10^{4}$ and $\gamma = 10^{-1}$.

References

DeGroot, M. H. and Fienberg, S. E. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 32(1/2):12–22, 1983.

[2] http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html

Flach, P. A. Classifier Calibration. In Sammut, C. and Webb, G. I. (eds.), *Encyclopedia of Machine Learning and Data Mining*, pp. 1–8. Springer US, Boston, MA, 2016.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On Calibration of Modern Neural Networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, International Convention Centre, Sydney, Australia, Aug 2017. PMLR.

Milios, D., Camoriano, R., Michiardi, P., Rosasco, L., and Filippone, M. Dirichlet-based Gaussian Processes for Large-scale Calibrated Classification. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 6008–6018. Curran Associates, Inc., 2018.

Naeini, M. P., Cooper, G. F., and Hauskrecht, M. Obtaining well calibrated probabilities using Bayesian binning. In *AAAI*, pp. 2901–2907. AAAI Press, 2015.

Niculescu-Mizil, A. and Caruana, R. Predicting Good Probabilities with Supervised Learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pp. 625–632, New York, NY, USA, 2005. ACM.

Pace, F., Venzano, D., Carra, D., and Michiardi, P. Flexible scheduling of distributed analytic applications. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID '17)*, pp. 100–109, May 2017.