1st Symposium on Advances in Approximate Bayesian Inference, 2018 1–11

Neural network ensembles and variational inference revisited

Marcin B. Tomczak MARCIN@PROWLER.IO
PROWLER.io Office, 72 Hills Road, Cambridge, CB2 1LA, UK

Siddharth Swaroop SS2163@CAM.AC.UK
Richard E. Turner RET26@CAM.AC.UK
Department of Engineering, University of Cambridge, Trumpington St, Cambridge CB2 1PZ, UK

Abstract

Ensembling methods and variational inference provide two orthogonal methods for obtaining reliable predictive uncertainty estimates for neural networks. In this work we compare and combine these approaches finding that: i) variational inference outperforms ensembles of neural networks, and ii) ensembled versions of variational inference bring further improvements. The first finding appears at odds with previous work (Lakshminarayanan et al., 2017), but we show that the previous results were due to an ambiguous experimental protocol in which the model and inference method were simultaneously changed.

1. Introduction

It has been a longstanding goal of machine learning to develop approaches to nonlinear regression that can capture complex input-output mappings whilst at the same time returning reliable predictive uncertainty estimates. Such approaches would be of great utility for both supervised and model based reinforcement learning systems (Chua et al., 2018).

Neural networks can capture complexity in the input-output mapping. A scalar output y_n can be assumed to be produced from a D-dimensional input $\mathbf{x}_n$ by passing the input through a neural network and adding Gaussian noise $p(y_n|\mathbf{x}_n, \theta) = \mathcal{N}(y_n; \mu_\theta(x_n), \sigma^2_\theta(x_n))$. Here both the mean and the variance of the Gaussian are assumed to be input-dependent (Nix and Weigend, 1994). Typically this is handled by a single network with parameters θ and two outputs: one for the mean and one for the variance. Alternatively, a simple input-independent Gaussian noise can be used $\sigma^2_\theta(x_n) = \sigma^2_\theta$.

How should predictive uncertainty be captured? Maximum-likelihood estimation (MLE) is known to be poor in this regard. The focus of this paper is on comparing two alternative approaches: approximate Bayesian inference and ensemble methods. We consider arguably the two most popular variants from these two camps: variational inference (VI) and ensembles of neural networks. Previous work has advocated the benefits of ensemble methods over VI (Lakshminarayanan et al., 2017). However, the results from this study are inconclusive as the model was not fixed: VI employed a simple input-independent noise, whilst deep ensembles used a more powerful input-dependent noise. We carry out like-for-like comparisons and find that VI outperforms MLE ensembles. Moreover, taking a pragmatic approach, VI is combined with ensemble methods resulting in further benefits.

2. Methods

This section lays out the orthogonal design choices investigated in the experiments. First, the choice of inference method. Second, ensembling.

Inference method: Variational Inference for Neural Networks. We compare maximum-likelihood learning to the VI approach to learning neural networks. VI first specifies a prior on the network weights and biases $p(\theta)$. Here, we use a factorized Gaussian distribution. The goal is then to approximate the posterior distribution over parameters given N observed data points, $q_\phi(\theta) \approx p(\theta|\{y_n, \mathbf{x}_n\}_{n=1}^N)$. Once again, we will employ a mean-field factorized Gaussian distribution for $q_\phi(\theta)$ so the variational parameters ϕ are the mean and variances of the weights and biases. VI learns the variational parameters by minimizing the KL-divergence to the true posterior, or equivalently by maximizing the evidence lower bound (ELBO, Blei et al. (2017)),

$$\mathcal{L}(\phi) = \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\theta)} \left[\log \frac{p(y_n, \theta|\mathbf{x}_n)}{q_\phi(\theta)} \right] \approx \frac{1}{M} \sum_{m=1}^{M} \sum_{n=1}^{N} \left[\log \frac{p(y_n, \theta^{(m)}|\mathbf{x}_n)}{q_\phi(\theta^{(m)})} \right] \text{ where } \theta^{(m)} \sim q_\phi(\theta).$$

The expectation over $q_\phi(\theta)$ is intractable and therefore approximated by M samples using simple Monte Carlo (Blundell et al., 2015). To reduce the variance of the gradients of $\mathcal{L}(\phi)$ we use local reparameterization trick (Kingma et al., 2015). The ELBO decomposes into a sum over data points which facilitates stochastic (mini-batch) optimization (here we employ ADAM (Kingma and Ba, 2014)). Finally, predictions are made by approximating the true predictive $p(y_n|\mathbf{x}_n) = \mathbb{E}_{p(\theta|\{y_n,\mathbf{x}_n\}_{n=1}^N)}[p(y_n|\mathbf{x}_n, \theta)]$ replacing the posterior by the variational approximation and applying simple Monte Carlo $q(y_n|\mathbf{x}_n) = \frac{1}{M}\sum_{m=1}^{M} p(y_n|\mathbf{x}_n, \theta^{(m)})$. This results in a predictive distribution formed from a mixture of M equally weighted Gaussians.

Ensembling. Ensemble methods fit M models to a data set, yielding M parameter estimates $\{\theta^{(m)}\}_{m=1}^M$, and then form predictions by averaging individual predictions in the same way as VI: $p_{ensemble}(y_n|\mathbf{x}_n) = \frac{1}{M}\sum_{m=1}^{M} p(y_n|\mathbf{x}_n, \theta^{(m)})$ (Dietterich, 2000). There are a variety of methods for encouraging variability in the fit models including (i) random parameter initialization and random mini-batches, (ii) resampling the data (e.g. bootstrap and bagging). Previous work found that, for neural networks trained by maximum likelihood, predictions resulting from method (i) were not improved by introducing method (ii) (Lakshminarayanan et al., 2017). We therefore employ only option (i).

3. Experiments

We compare the following orthogonal design choices: MLE vs VI, single model vs ensembling. These comparisons are performed for input-independent and input-dependent noise models. Performance is assessed via held-out log-likelihood.

Synthetic 1D data experiments. First, we compare MLE ensembles and ensembles of VI on simple one dimensional functions. The results are collected in Figure 1 and Table 1. Ensembled VI outperforms ensembled MLE on any data set by a significant margin.

model/dataset	$f(x) = [sgn(x)]_+$	$f(x) = \max(0, x)$	$f(x) = \tanh(x)$	$f(x) = \lvert x \rvert$
ensembles MLE	-1.84 ± 2.39	-2.84 ± 1.53	-42.67 ± 2.96	-0.47 ± 0.62
ensembles VI	0.03 ± 0.04	0.38 ± 0.04	0.10 ± 0.02	0.39 ± 0.02

Table 1: Synthetic 1D regression data sets: average test log-likelihoods (see text for details).

The reason for the poor performance of the MLE trained models is as follows: Models trained by MLE are very confident. The ensemble therefore requires large diversity between MLE trained models to avoid becoming overconfident. However, in practice often most of the MLE ensemble predictions are similar leading to overconfident extrapolation and interpolation and unstable performance. In contrast, VI trained models usually exhibit underfitting, in the sense that the resulting fits err on the side of simplicity, in agreement with Trippe and Turner (2018). Nevertheless, VI generalizes mostly better and in a more consistent way than ensembles of MLE.

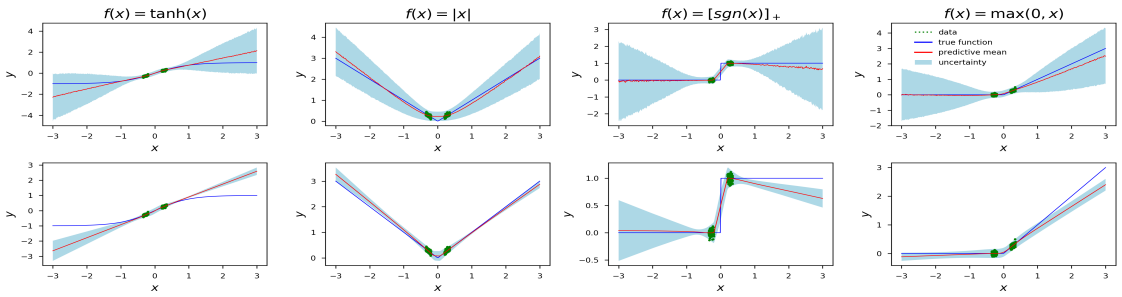

Figure 1: Comparison of ensembles of VI (top row) and ensembles of MLE (bottom row).

UCI datasets. Next, we provide comparison on *UCI* datasets (Dheeru and Karra Taniskidou, 2017). We adopt the experimental setup presented in Hernández-Lobato and Adams (2015) and followed in Bui et al. (2016); Gal and Ghahramani (2016); Lakshminarayanan et al. (2017); Trippe and Turner (2018). Results are shown in Figure 2. We use the following abbreviations: *IIDV - Input-Independent Variance*, *IDV - Input-Dependent Variance*. We denote VI ensembles by *eVI* and MLE ensembles by *eMLE*.

Like Lakshminarayanan et al. (2017) we find that *eMLE-IDV* outperforms *VI-IIV*, but these results are confounded by the fact that the use of input-dependent noise improves test log-likelihood in all cases for VI. Critically, in direct comparisons, we find that even *non-ensembled VI improves over eMLE in 16 comparisons, they tie in 10, and MLE wins in 10. Additionally, non-ensembled VI often provides far higher performance when it outperforms eMLE (see Figure A.3)*. Ensembling VI outperforms eMLE (or results in no significant difference) on any data set but protein, regardless of the noise type, making it the best method overall. Further detailed comparison between the methods is presented in the appendix.

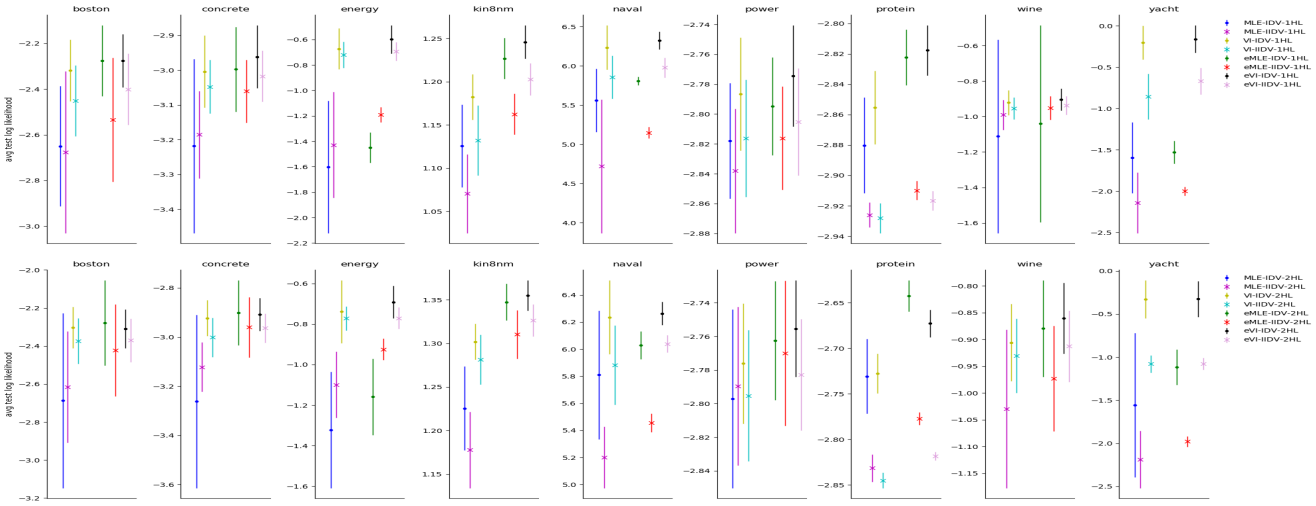

Figure 2: UCI regression datasets: average test log-likelihoods for one hidden layer networks (top) and two hidden layers networks (bottom).

Cart pole. Finally we compare ensembles of VI and MLE on data generated from a *Cart Pole* swing up task (Brockman et al., 2016) to simulate the challenges of model-based RL. The models are first trained on data from a random policy. For testing, we apply a sequence of optimal actions and perform rollouts from both MLE ensembles and ensembles of VI comparing to ground truth. See Figure 3. Ensembles of VI provide predictions that are less overconfident and more accurate than deep ensembles. This makes ensembles of VI models a potentially good candidate for improving model-based reinforcement learning algorithms, e.g. in model predictive control (Chua et al., 2018).

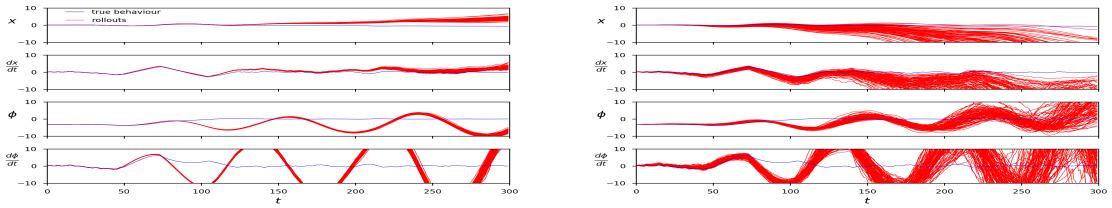

Figure 3: Rollouts from ensembles MLE (left) and ensembled VI (right).

4. Conclusions

Ensembles of MLE and VI were compared on different neural network architectures. VI was found to mostly improve over ensembles of MLE, but ensembling VI performed even better. The combination of frequentist ensembling methods and Bayesian methods is an interesting avenue for future work potentially mitigating against model miss-specification and deficiencies in approximate inference (Fushiki et al., 2005).

References

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Thang Bui, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep gaussian processes for regression using approximate expectation propagation. In *International Conference on Machine Learning*, pages 1472–1481, 2016.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *arXiv preprint arXiv:1805.12114*, 2018.

Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

Tadayoshi Fushiki et al. Bootstrap prediction and bayesian prediction under misspecified models. *Bernoulli*, 11(4):747–758, 2005.

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

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference On*, volume 1, pages 55–60. IEEE, 1994.

Brian Trippe and Richard Turner. Overpruning in variational bayesian neural networks. *arXiv preprint arXiv:1801.06230*, 2018.

Appendix A.

A.1. Further experimental details

UCI experiments. During one hidden layer network experiments we use the architecture with 50 hidden units and ReLU activations for both MLE and VI experiments as done in Hernández-Lobato and Adams (2015); Trippe and Turner (2018); Gal and Ghahramani (2016); Lakshminarayanan et al. (2017). We also run the experiments with two hidden layers, where an additional hidden layer with 50 hidden units and ReLU activations is added. We normalise both inputs and outputs for neural networks to have zero mean and unit variance.

We tune various hyperparameters on validation sets. For MLE, we optimise using ADAM Kingma and Ba (2014) for 40 epochs and pick a learning rate and batch size based on the performance on a validation set from $[0.001, 0.01]$ and $[100, 32]$, respectively. Validation sets contain 10% of training data. The models trained for different hyperparameters share the same random seed. The model with the best validation performance is retrained on full training data (including validation set) before evaluation on held-out data.

To parametrise variance, we use softplus activation to which we add small constant 10^{-5} for numerical stability. Input independent variance is initialised to 0.31 (softplus of -1). Weights are initialised from $U[-\frac{1}{\sqrt{2n_{out}}}, \frac{1}{\sqrt{2n_{out}}}]$ where n_{out} is the size of the output. Biases are initialised with zeros.

For VI experiments, we perform 25000 optimisation updates on ELBO. Similarly like in MLE case, we pick a learning rate based on validation set from $[0.0003, 0.001]$. We use one sample of parameters from variational posterior to approximate the gradient of ELBO. We initialise the variances of variational distributions to 10^{-5}. Means of variational distributions for weights are initialised from $\mathcal{N}(0, \frac{1}{\sqrt{2n_{out}}})$ and means of variational distributions for biases are initialised with zeros. We use standard parameters for ADAM optimiser. We also use batch size of 100. Priors on weights are unit variance Gaussians.

Cart pole We experiment with two data sets: the sequence of states obtained by applying random actions consisting of 128 timesteps and the sequence of states obtained by applying the optimal actions having 300 timesteps. The models are trained on the former data set. We use one hidden layer neural network with 150 units and swish activations. The variance is input dependent and the output of the network is transformed by *softplus* transformation. For completeness, we report the rollouts on train data in figure Figure A.1.

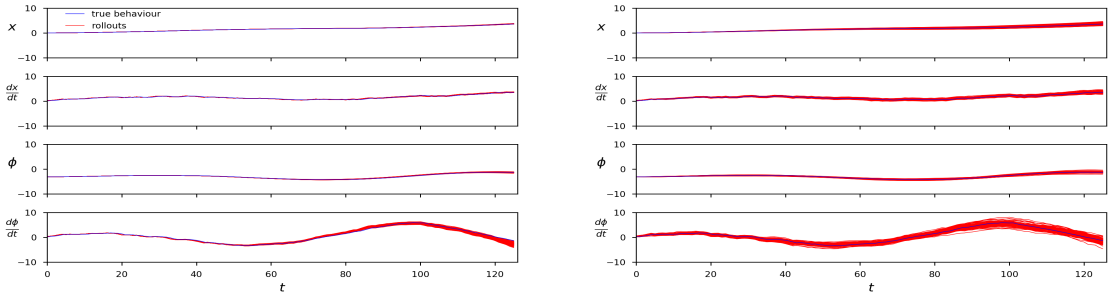

Figure 4: Rollouts from MLE ensembles (left) and ensembled VI (right) on train data (given the sequence of random actions).

A.2. Details on ensembling

The deep ensembles approach performs one further approximation (Lakshminarayanan et al., 2017). Rather than using the mixture of Gaussians arising from the ensemble to perform the prediction directly, they moment match a Gaussian distribution. We found out that this choice that was justified for reasons of computational simplicity (for example, held-out log-likelihoods become trivial to compute) leads to slightly worse results. In both cases of VI and MLE the ensembles consisted five networks, as comparison in Lakshminarayanan et al. (2017). We provide the comparison with this approach (by performing Gaussian approximation of predictive distribution for both MLE and VI) in Table A.3.

In the case of ensembles, the predictive distribution is taken to be a mixture of equally weighted members, i.e.

$$p_{ensemble}(y|\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} p_i(y|\mathbf{x}) \tag{1}$$

where $p_i(y|\mathbf{x})$ denote the predictive distributions of i-th ensemble member. Moment matching is done by calculating the moments of $p_{ensemble}$ (in case of a mixture of Gaussians $\mu_{MM} = \frac{1}{N}\sum_{i=1}\mu_i$ and $\sigma^2_{MM} = \frac{1}{N}\sum_{i=1}(\mu_i^2 + \sigma_i^2) - \mu^2_{MM}$. Then the moment-matched predictive distribution is set to a Gaussian distribution $\mathcal{N}(\mu_{MM}, \sigma^2_{MM})$.

Moment matching for VI is done in the same manner: firstly $\mu_i = \int yp(y|\mathbf{x},\theta)q_i(\theta)dy$ and $\sigma_i^2 = \int(y-\mu_i)^2 p(y|\mathbf{x},\theta)q_i(\theta)dy$ are calculated, then these moments are used to define $\mathcal{N}(\mu_{MM}, \sigma^2_{MM})$ as in MLE case.

The log likelihood for MLE ensembles (without Gaussian moment-matched approximation) is obtained with the following formula:

$$\log p_{ensemble}(y|\mathbf{x}) = \log\sum_{i=1}^{N}\exp\{\log p_i(y|\mathbf{x},\theta^{(i)})\} - \log N \tag{2}$$

where $\{\theta^{(i)}\}_{i=1}^{N}$ denotes the ensemble of MLE estimators. In the case of VI:

$$\log p_{ensemble}(y|\mathbf{x}) = \log \frac{1}{N}\sum_{i=1}^{N}\int p(y|\mathbf{x},\theta)q_i(\theta)d\theta \approx$$
$$\approx \log \frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{j=1}^{M} p(y|\mathbf{x},\theta_j^{(i)}) = \log \sum_{i=1}^{N}\sum_{j=1}^{M}\exp\{\log p(y|\mathbf{x},\theta_j^{(i)})\} - \log N - \log M \quad (3)$$

where q_i denotes the variational posterior of the i-th member of an ensemble. The approximate equality comes from approximating the integral with samples. The last equality is introduced to make use of *logsumexp* trick that allows to calculate the last quantity on a right hand side in a numerically stable way. This is done similarly as in Gal and Ghahramani (2016).

A.3. Additional experimental results

We report additional parwise comparisons between selected methods. The histograms are obtained by taking the differences of log probability on a test set between two methods. Any point in the bottom figures correspond to fitting both models on a single train/test split. The colors in the bottom figures correspond to different datasets.

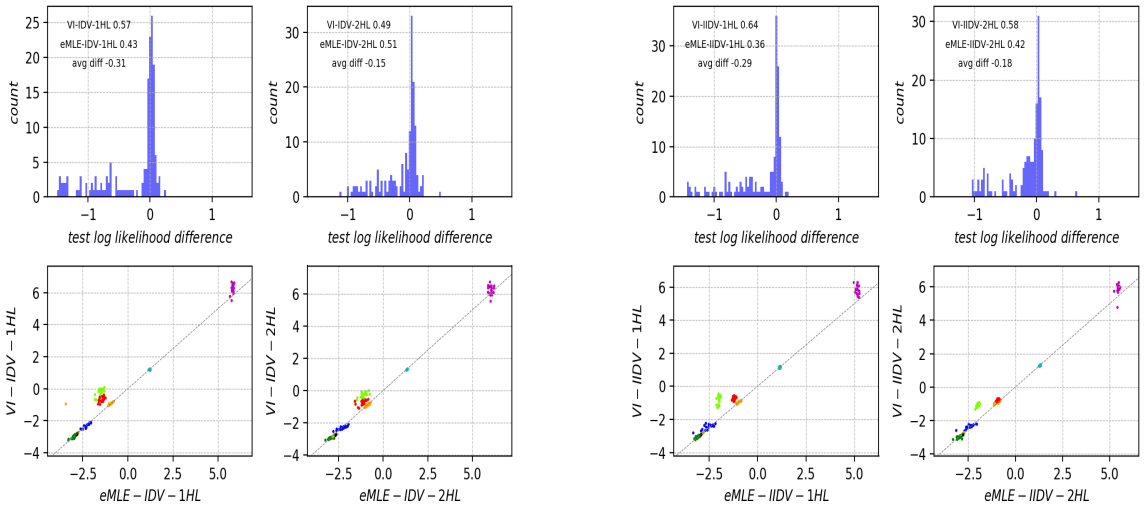

Figure 5: Comparison of non-ensembled VI vs ensembled MLE for input dependent noise (left) and input independent noise (right). VI yields better results.

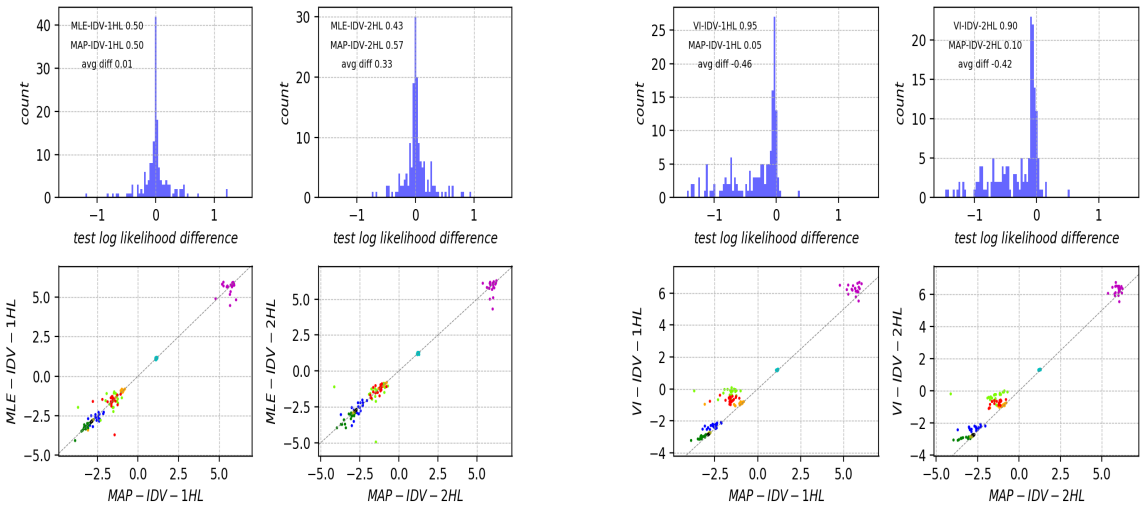

Figure 6: Comparison MLE vs MAP (left) and VI vs MAP (right). Regularisation of the point estimate is not enough to obtain good experimental results. VI significantly outperforms MAP. MAP outperforms MLE for 2 hidden layers architecture because of a few cases of suvere overfitting for MLE.

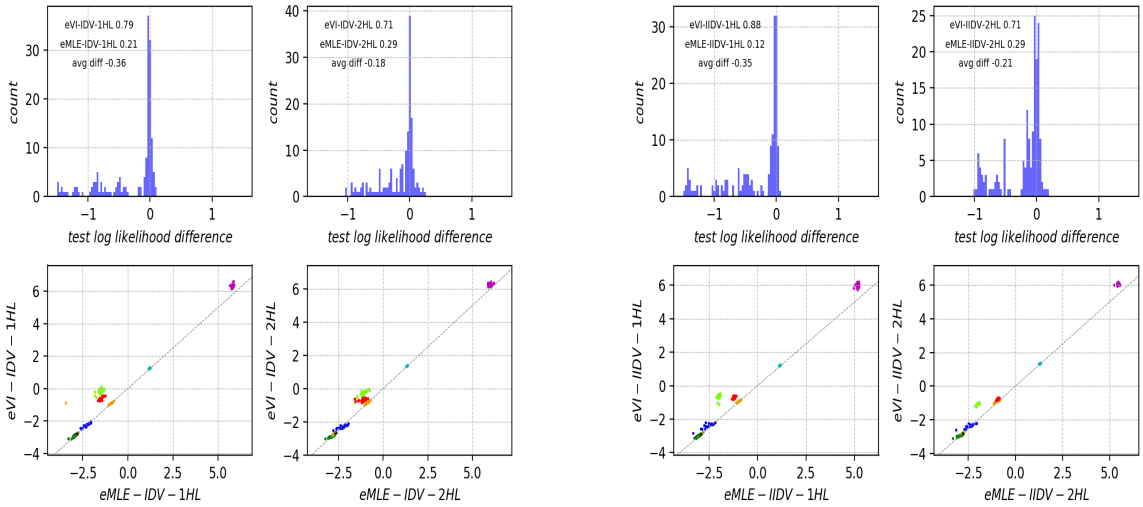

Figure 7: Comparison of ensebles of VI vs ensembles of MLE with input dependent observation noise (left) and input independent observation noise (right). Ensembles of VI significantly outperform ensembles of MLE regardless of the noise type.

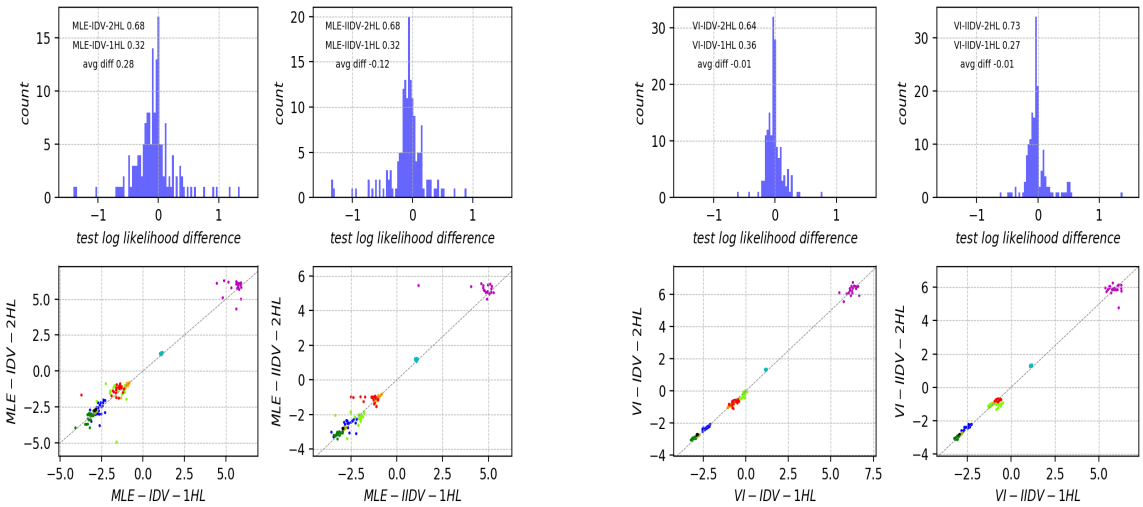

Figure 8: Comparison of models with one hidden layer vs two hidden layers: MLE (left) and VI (right). Deeper network providing better modeling capabilities for MLE.

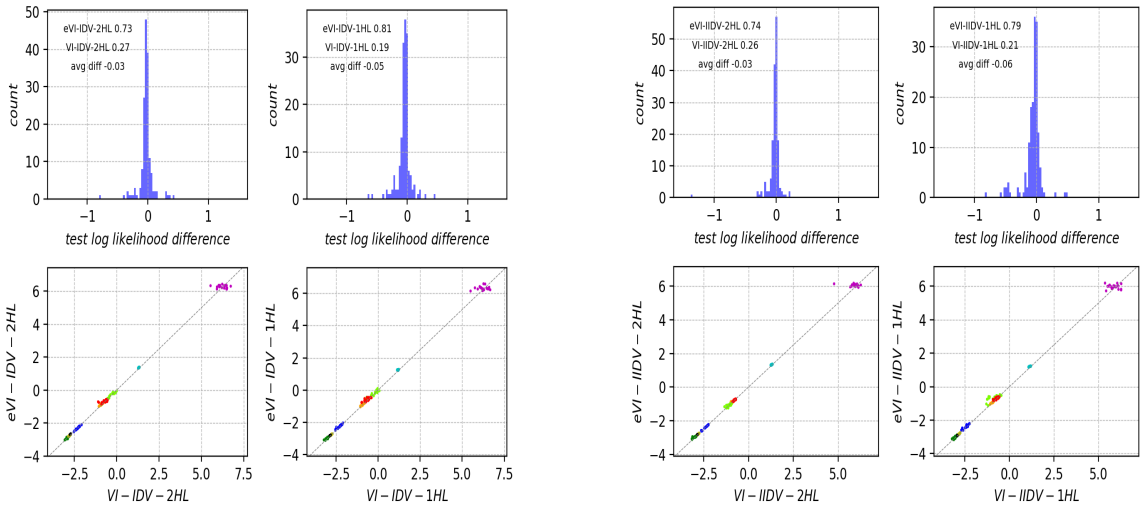

Figure 9: Comparison of ensembled VI vs non-ensembled VI for input dependent noise (left) and input independent noise (right). Ensembling VI provides small, yet consistent improvment in the results.

We also report all experimental results in Table A.3 below. We use the additional abbreviation *GPA - Gaussian Posterior Approximation*, which corresponds to the described moment-matched Gaussian approximation of predictive distribution.

model/dataset	boston	concrete	energy	kin8nm	naval	power	protein	wine	yacht
MAP-IDV-1HL	-2.58 ± 0.28	-3.22 ± 0.22	-1.62 ± 0.35	1.13 ± 0.04	5.63 ± 0.30	-2.82 ± 0.04	-2.87 ± 0.03	-1.11 ± 0.48	-1.65 ± 0.55
MAP-IDV-2HL	-2.65 ± 0.37	-3.19 ± 0.32	-1.32 ± 0.27	1.24 ± 0.04	5.94 ± 0.20	-2.80 ± 0.05	-2.74 ± 0.04	-1.65 ± 2.76	-1.50 ± 0.68
MAP-IIDV-1HL	-2.65 ± 0.34	-3.17 ± 0.14	-1.32 ± 0.30	1.08 ± 0.04	4.85 ± 0.44	-2.83 ± 0.04	-2.94 ± 0.01	-0.99 ± 0.08	-2.10 ± 0.26
MAP-IIDV-2HL	-2.60 ± 0.30	-3.18 ± 0.21	-1.10 ± 0.20	1.18 ± 0.05	5.18 ± 0.40	-2.79 ± 0.05	-2.83 ± 0.01	-1.02 ± 0.12	-2.35 ± 0.35
MLE-IDV-1HL	-2.65 ± 0.26	-3.22 ± 0.25	-1.60 ± 0.52	1.13 ± 0.05	5.56 ± 0.40	-2.82 ± 0.04	-2.88 ± 0.03	-1.11 ± 0.55	-1.60 ± 0.43
MLE-IDV-2HL	-2.69 ± 0.46	-3.26 ± 0.35	-1.32 ± 0.29	1.23 ± 0.05	5.81 ± 0.47	-2.80 ± 0.05	-2.73 ± 0.04	-4.04 ± 13.11	-1.56 ± 0.84
MLE-IIDV-1HL	-2.68 ± 0.35	-3.19 ± 0.13	-1.43 ± 0.42	1.07 ± 0.05	4.72 ± 0.85	-2.84 ± 0.04	-2.93 ± 0.01	-0.99 ± 0.08	-2.14 ± 0.37
MLE-IIDV-2HL	-2.62 ± 0.29	-3.12 ± 0.10	-1.10 ± 0.16	1.18 ± 0.04	5.20 ± 0.23	-2.79 ± 0.05	-2.83 ± 0.01	-1.03 ± 0.15	-2.19 ± 0.33
VI-IDV-1HL	-2.32 ± 0.13	-3.00 ± 0.10	-0.67 ± 0.16	1.18 ± 0.03	6.23 ± 0.28	-2.79 ± 0.04	-2.86 ± 0.02	-0.92 ± 0.07	-0.20 ± 0.21
VI-IDV-2HL	-2.30 ± 0.11	-2.92 ± 0.07	-0.74 ± 0.15	1.30 ± 0.02	6.24 ± 0.27	-2.78 ± 0.04	-2.73 ± 0.02	-0.91 ± 0.07	-0.33 ± 0.22
VI-IIDV-1HL	-2.45 ± 0.15	-3.05 ± 0.08	-0.72 ± 0.10	1.13 ± 0.04	5.86 ± 0.28	-2.82 ± 0.04	-2.93 ± 0.01	-0.95 ± 0.06	-0.86 ± 0.27
VI-IIDV-2HL	-2.37 ± 0.12	-3.00 ± 0.08	-0.77 ± 0.06	1.28 ± 0.03	5.88 ± 0.29	-2.80 ± 0.04	-2.84 ± 0.01	-0.93 ± 0.07	-1.08 ± 0.10
eMAP-IDV-1HL	-2.35 ± 0.26	-2.99 ± 0.11	-1.44 ± 0.12	1.22 ± 0.03	5.80 ± 0.09	-2.79 ± 0.03	-2.82 ± 0.02	-1.00 ± 0.38	-1.48 ± 0.16
eMAP-IDV-2HL	-2.29 ± 0.20	-2.90 ± 0.14	-1.12 ± 0.13	1.34 ± 0.02	5.99 ± 0.09	-2.76 ± 0.04	-2.65 ± 0.01	-0.92 ± 0.13	-1.20 ± 0.18
eMAP-IDV-GPA-1HL	-2.34 ± 0.22	-3.02 ± 0.11	-1.51 ± 0.12	1.22 ± 0.03	4.51 ± 0.03	-2.80 ± 0.03	-2.84 ± 0.03	-0.93 ± 0.09	-1.61 ± 0.21
eMAP-IDV-GPA-2HL	-2.31 ± 0.22	-2.91 ± 0.16	-1.22 ± 0.20	1.35 ± 0.02	4.56 ± 0.03	-2.77 ± 0.04	-2.69 ± 0.03	-0.94 ± 0.12	-1.21 ± 0.19
eMAP-IIDV-1HL	-2.51 ± 0.29	-3.06 ± 0.09	-1.15 ± 0.07	1.16 ± 0.02	5.09 ± 0.13	-2.82 ± 0.04	-2.91 ± 0.00	-0.95 ± 0.06	-2.01 ± 0.06
eMAP-IIDV-2HL	-2.39 ± 0.23	-2.97 ± 0.10	-0.91 ± 0.07	1.31 ± 0.03	5.47 ± 0.07	-2.77 ± 0.04	-2.78 ± 0.01	-0.96 ± 0.10	-1.99 ± 0.06
eMAP-IIDV-GPA-1HL	-2.47 ± 0.25	-3.05 ± 0.09	-1.17 ± 0.07	1.16 ± 0.02	4.59 ± 0.04	-2.82 ± 0.04	-2.91 ± 0.00	-0.95 ± 0.07	-2.00 ± 0.05
eMAP-IIDV-GPA-2HL	-2.35 ± 0.19	-2.95 ± 0.10	-0.92 ± 0.08	1.31 ± 0.02	4.71 ± 0.02	-2.77 ± 0.04	-2.78 ± 0.00	-0.96 ± 0.11	-2.00 ± 0.06
eMLE-IDV-1HL	-2.28 ± 0.15	-3.00 ± 0.12	-1.45 ± 0.12	1.23 ± 0.02	5.81 ± 0.05	-2.79 ± 0.03	-2.82 ± 0.02	-1.04 ± 0.56	-1.53 ± 0.14
eMLE-IDV-2HL	-2.28 ± 0.22	-2.90 ± 0.13	-1.16 ± 0.19	1.35 ± 0.02	6.03 ± 0.10	-2.76 ± 0.04	-2.64 ± 0.02	-0.88 ± 0.09	-1.12 ± 0.20
eMLE-IDV-GPA-1HL	-2.32 ± 0.19	-3.03 ± 0.16	-1.53 ± 0.15	1.23 ± 0.02	4.51 ± 0.02	-2.80 ± 0.04	-2.84 ± 0.03	-0.94 ± 0.09	-1.70 ± 0.23
eMLE-IDV-GPA-2HL	-2.30 ± 0.24	-2.91 ± 0.13	-1.22 ± 0.20	1.35 ± 0.02	4.57 ± 0.03	-2.77 ± 0.04	-2.68 ± 0.02	-0.91 ± 0.09	-1.15 ± 0.21
eMLE-IIDV-1HL	-2.53 ± 0.27	-3.06 ± 0.09	-1.19 ± 0.06	1.16 ± 0.02	5.15 ± 0.07	-2.82 ± 0.03	-2.91 ± 0.01	-0.95 ± 0.07	-2.00 ± 0.05
eMLE-IIDV-2HL	-2.42 ± 0.24	-2.96 ± 0.12	-0.92 ± 0.05	1.31 ± 0.03	5.46 ± 0.07	-2.77 ± 0.04	-2.78 ± 0.01	-0.97 ± 0.10	-1.98 ± 0.06
eMLE-IIDV-GPA-1HL	-2.49 ± 0.25	-3.05 ± 0.09	-1.17 ± 0.07	1.17 ± 0.02	4.60 ± 0.05	-2.81 ± 0.04	-2.91 ± 0.01	-0.95 ± 0.07	-2.01 ± 0.05
eMLE-IIDV-GPA-2HL	-2.35 ± 0.20	-2.94 ± 0.11	-0.92 ± 0.05	1.31 ± 0.03	4.70 ± 0.02	-2.77 ± 0.04	-2.78 ± 0.00	-0.97 ± 0.10	-1.98 ± 0.06
eVI-IDV-1HL	-2.28 ± 0.12	-2.96 ± 0.09	-0.60 ± 0.11	1.25 ± 0.02	6.32 ± 0.11	-2.77 ± 0.03	-2.82 ± 0.02	-0.90 ± 0.06	-0.16 ± 0.17
eVI-IDV-2HL	-2.31 ± 0.10	-2.91 ± 0.07	-0.69 ± 0.08	1.36 ± 0.02	6.26 ± 0.09	-2.76 ± 0.03	-2.67 ± 0.02	-0.86 ± 0.07	-0.33 ± 0.21
eVI-IIDV-1HL	-2.40 ± 0.16	-3.02 ± 0.07	-0.69 ± 0.07	1.20 ± 0.02	5.98 ± 0.13	-2.81 ± 0.04	-2.92 ± 0.01	-0.94 ± 0.05	-0.67 ± 0.16
eVI-IIDV-2HL	-2.37 ± 0.11	-2.96 ± 0.06	-0.77 ± 0.05	1.33 ± 0.02	6.04 ± 0.06	-2.78 ± 0.03	-2.82 ± 0.00	-0.91 ± 0.07	-1.08 ± 0.07

Table 2: UCI regression datasets: average test log-likelihoods (see text for details).

A.4. Comparison of timing

We provide the comparison of training and inference for both VI and MLE in ensembled and non-ensembled cases. The timings are gathered in the table Table A.4.

model	MLE-1HL	MLE-2HL	VI-1HL	VI-2HL	eMLE-1HL	eMLE-2HL	eVI-1HL	eVI-2HL
training	116.73 ± 21.51	141.53 ± 8.77	386.13 ± 5.55	601.90 ± 22.46	147.74 ± 8.96	247.15 ± 4.22	573.36 ± 12.69	1089.96 ± 18.91
inference	0.42 ± 0.15	0.48 ± 0.15	246.17 ± 11.65	530.55 ± 24.35	0.53 ± 0.06	1.01 ± 0.11	953.44 ± 59.57	2402.80 ± 54.41

Table 3: Comparison of training (100 optimisation updates) and inference in miliseconds for boston housing regression task.

Performing the same number of updates for VI takes roughly four times longer than in the case of MLE. Additionally, approximate inference methods usually require to be run for higher number of epochs, which additionally increases training time.