Approximate inference for Bayesian neural networks

Silas Brack

Spring 2022

Abstract

1 Introduction

Neural networks (NNs) suffer from calibration issues and overconfidence, catastrophic interference in continual learning, and sensitivity to hyper-parameter selection.

Bayesian neural networks (BNNs) arise from applying Bayesian inference on the layer weights of NNs and are one solution for these issues, allowing an estimate of the uncertainty of the output of NNs.

However, BNNs networks are expensive and cumbersome to train. Their posteriors are intractable in an exact sense, meaning these have to be approximately inferred. Furthermore, they inherit having a large parameter space (and consequently a large optimisation space) from their non-Bayesian counterparts, leading to slow inference for traditional methods such as Markov-chain Monte Carlo (MCMC), which, while yielding very competitive results [9], scales poorly with increasing dimensionality. Other Bayesian and quasi-Bayesian methods have been proposed as alternatives, such as variational inference, SWAG/MultiSWAG, deep ensembles, and Laplace approximations.

This paper analyses these alternative approximate inference methods.

2 Previous Work

2.1 Bayes' Theorem

Bayes' theorem describes the probability of an event occurring with respect to prior knowledge of that event. Specifically for modelling, it can be used to relate latent variables with observed variables, allowing distributions on these latent variables to be inferred. We therefore want to find the posterior $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ by According to Bayes' theorem, the posterior is given by

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{y})} \tag{1}$$

where $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ is the likelihood, parameterised by the model parameters $\boldsymbol{\theta}$, $p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}; \boldsymbol{\alpha})$ is the prior, parameterised by its hyper-parameters $\boldsymbol{\alpha}$, and $p(\boldsymbol{y}) = p_{\boldsymbol{\theta}}(\boldsymbol{y})$ is the marginal likelihood (or evidence).

Furthermore, the posterior predictive probability, i.e., the probability of observing some new data, is given by

$$p(\boldsymbol{y}_{\text{new}} \mid \boldsymbol{y}) = \int p(\boldsymbol{y}_{\text{new}} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \boldsymbol{y}) \, d\boldsymbol{\theta}. \tag{2}$$

2.2 Variational Inference

In variational inference (VI), we minimise the KL divergence between the posterior distribution $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ and a variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ parameterised by its variational parameters $\boldsymbol{\phi}$ by optimising with respect to these parameters. The KL divergence is defined as

$$\begin{aligned} D_{\mathrm{KL}}\left[q(\boldsymbol{\theta}) \parallel p(\boldsymbol{y} \mid \boldsymbol{\theta})\right] &\equiv \mathbb{E}_q\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{y} \mid \boldsymbol{\theta})}\right] \\ &= \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}_q\left[\log p(\boldsymbol{y} \mid \boldsymbol{\theta})\right] \end{aligned} \tag{3}$$

However, the KL divergence contains the evidence term $\log p(\boldsymbol{y})$, which is intractable. Instead, we define a lower bound on this marginal likelihood term that is surrogate to the KL divergence (known as the ELBO). Since they are surrogate, ELBO fulfils the property $\mathcal{L}\left[q_{\boldsymbol{\phi}}\right] = \log p(\boldsymbol{y}) - D_{\mathrm{KL}}\left[q(\boldsymbol{\theta}) \parallel p(\boldsymbol{y} \mid \boldsymbol{\theta})\right]$, minimising the KL divergence is equivalent to maximising the ELBO, which is defined as

$$\begin{aligned} \mathcal{L}\left[q_{\boldsymbol{\phi}}\right] &\equiv \mathbb{E}_q\left[\log p(\boldsymbol{y}, \boldsymbol{\theta})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] \\ &= \mathbb{E}_q\left[\log p(\boldsymbol{y} \mid \boldsymbol{\theta})\right] + \mathbb{E}_q\left[\log p(\boldsymbol{\theta})\right] - \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] \\ &= \mathbb{E}_q\left[\log p(\boldsymbol{y} \mid \boldsymbol{\theta})\right] \underbrace{- \mathbb{E}_q\left[\log p(\boldsymbol{\theta})\right]}_{\text{Cross-entropy}} \underbrace{- \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right]}_{\text{Entropy}} \end{aligned} \tag{4}$$

where the first term $\mathbb{E}_q\left[\log p(\boldsymbol{y} \mid \boldsymbol{\theta})\right]$ is the data (or likelihood) term, $-\mathbb{E}_q\left[\log p(\boldsymbol{\theta})\right]$ is the cross-entropy of the prior with respect to the variational approximation, and $-\mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right]$ is the entropy term of the variational approximation. The last two terms can be interpreted as the regularising KL divergence (or relative entropy) from the prior to the variational approximation, $D_{\mathrm{KL}}\left[q(\boldsymbol{\theta}) \parallel p(\boldsymbol{\theta})\right] = \mathbb{E}_q\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}_q\left[\log p(\boldsymbol{\theta})\right]$. Furthermore, the likelihood term in the ELBO can be decomposed further by assuming independence in the observations as $\mathbb{E}_q\left[\log p(\boldsymbol{y} \mid \boldsymbol{\theta})\right] = \frac{1}{N} \sum_i^N \mathbb{E}_q\left[\log p(y_i \mid \boldsymbol{\theta})\right]$.

Overall BNNs have been found to be relatively ineffective unless the number of observations is greater than the number of model parameters.

2.3 Methods

Multiple approximate inference methods were implemented. Specifically, VI with mean-field, full-rank, low-rank and radial variational families, *maximum a posteriori* estimation, Laplace approximation, deep ensembles, and MultiSWAG. These methods will be described in the following sections.

2.3.1 Mean-Field Variational Approximation

One common variational family used to approximate the real posterior is a product of independent Gaussian distributions

(the mean-field approximation) such that each model parameter is sampled from a normal distribution and is independent of all other model parameters, yielding a Gaussian with a diagonal covariance matrix. In mean-field VI, model weights are sampled from an approximate posterior $q(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta} \mid \boldsymbol{y})$ as

$$\begin{aligned} q(\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \Leftrightarrow \\ \boldsymbol{\theta} &= \boldsymbol{\mu} + \boldsymbol{\sigma} \circ \boldsymbol{\epsilon} \end{aligned} \tag{5}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. For this type of variational approximation, the entropy term is given by $\mathbb{H}\left[q(\theta)\right] = -\sum_i \log \sigma_i$ and the cross-entropy of the prior relative to the variational approximation is calculated via Monte Carlo simulation by taking the prior log probability with respect to mean-field posterior samples as

$$\begin{aligned} \mathbb{H}\left[q(\boldsymbol{\theta}), p(\boldsymbol{\theta})\right] &= -\int q(\boldsymbol{\theta}) \log(p(\boldsymbol{\theta}))\, d\boldsymbol{\theta} \\ &\approx \frac{1}{S} \sum_{s=1}^{S} \log p(\boldsymbol{\theta}^{(s)}) \end{aligned} \tag{6}$$

where $\boldsymbol{\theta}^{(s)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$.

The use of the mean-field approximation for BNNs has been found to be unreliable [20] and, for increasingly wide neural networks, converges towards the prior [2]. However, for *deep* BNNs, the mean-field assumption may be reasonable [5].

2.3.2 Full-Rank Variational Approximation

As an alternative to the mean-field approximation, a full-rank approximation doesn't assume independence of the model parameters, instead being sampled as

$$q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{\mathrm{FR}}). \tag{7}$$

For D model parameters, while a mean-field approximation has scale parameters $\boldsymbol{\sigma}^D$, the full-rank approximation has scale parameters $\boldsymbol{\Sigma}_{\mathrm{FR}}^{D \times D}$. The variational parameters scale quadratically with the number of model parameters, which constitutes a significant limitation of this type of approximation. Specifically, since NNs are typically overparameterised, possessing from thousands to millions to billions of parameters, the quadratic number of variational parameters makes this type of VI infeasible for BNNs. As such, full-rank VI is not applied in any of the experiments in this paper.

2.3.3 Low-Rank Variational Approximation

As a compromise between the last two methods, a low-rank variational approximation uses a low-rank approximation of the covariance matrix $\boldsymbol{\Sigma}_{\mathrm{LR}}$.

$$q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{\mathrm{LR}}). \tag{8}$$

Need more

2.3.4 Radial Variational Approximation

Farquhar, Osborne, and Gal [4] introduces the Radial approximate posterior, For each NN layer, this posterior is sampled similarly to the mean-field (eq. (5)), but by normalising the $\boldsymbol{\epsilon}$ term (projecting it onto a hypersphere) yielding a direction term $\boldsymbol{\epsilon} / \left\|\boldsymbol{\epsilon}\right\|$, and scaling it by the distance r. Therefore, for each layer, the radial posterior is sampled as

$$\begin{aligned} q(\boldsymbol{\theta}) &= \mathrm{Radial}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \\ \boldsymbol{\theta} &= \boldsymbol{\mu} + \boldsymbol{\sigma} \circ \frac{\boldsymbol{\epsilon}}{\left\|\boldsymbol{\epsilon}\right\|} r \end{aligned} \tag{9}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $r \sim \mathcal{N}(0, 1)$. The entropy term of the Radial approximate posterior is given $\mathbb{H}\left[q(\theta)\right] = -\sum_i \log \sigma_i + \mathrm{const}$ (and is therefore approximately equal to the entropy of the mean-field approximation up to a constant) and the cross-entropy term is calculated via Monte Carlo simulation as for mean-field VI (section 2.3.1), but by sampling from a radial posterior $\boldsymbol{\theta}^{(s)} \sim \mathrm{Radial}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ as in eq. (9).

2.3.5 Maximum a Posteriori Estimation

The *maximum a posteriori* solution finds a point estimate of the posterior given by its maximum.

$$\begin{aligned} \boldsymbol{\theta}_{\mathrm{MAP}} &= \max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \boldsymbol{y}) \\ &= \max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \boldsymbol{y}) \\ &= \max_{\boldsymbol{\theta}} \left[\log p(\boldsymbol{y} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right] \\ &= \max_{\boldsymbol{\theta}} \left[\sum_{i=1}^{N} \log p(y_i \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right] \end{aligned} \tag{10}$$

Need more

2.3.6 Laplace Approximation

In the Laplace approximation (LA) [3], the posterior is approximated by a Gaussian, similarly to mean-field VI. However, instead of finding the optimal Gaussian distribution locations and scales by maximising the ELBO, the LA finds the location by computing the MAP solution $\boldsymbol{\theta}_{\mathrm{MAP}}$, and the scale by performing a Taylor expansion around this solution and determining the curvature via its Hessian matrix.

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) \approx \mathcal{N}(\boldsymbol{\theta}_{\mathrm{MAP}}, \boldsymbol{\Sigma}) \tag{11}$$

where $\boldsymbol{\Sigma} = -(\left.\nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{y}; \boldsymbol{\theta})\right|_{\boldsymbol{\theta}_{\mathrm{MAP}}})^{-1}$.

Need more

2.3.7 Deep Ensembles

In deep ensembles [11], M neural networks are trained with different initialisations. To make predictions with a deep ensemble,

$$\begin{aligned} p(\boldsymbol{y}_{\mathrm{new}} \mid \boldsymbol{y}) &= \int p(\boldsymbol{y}_{\mathrm{new}} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \boldsymbol{y})\, d\boldsymbol{\theta} \\ &= \frac{1}{M} \sum_{m=0}^{M} p(\boldsymbol{y}_{\mathrm{new}} \mid \boldsymbol{\theta}^{(m)}) \end{aligned} \tag{12}$$

where $\boldsymbol{\theta}^{(m)}$ are the model parameters for ensemble m, and $p(\boldsymbol{y}_{\mathrm{new}} \mid \boldsymbol{\theta}^{(m)})$ are therefore the normalised logits, or predicted probabilities, from the model.

Need more

2.3.8 MultiSWAG

Stochastic weight averaging estimates the final point estimate of the weights of a neural network $\boldsymbol{\theta}_{\text{SWA}}$ as the average of the model parameters $\boldsymbol{\theta}$ at the end of each epoch obtained via gradient descent after convergence has been reached.

For SWAG [13], the standard deviation of the model parameters $\boldsymbol{\sigma}_{\text{SWA}}$ is also calculated. From the SWA estimates of the model parameter mean and standard deviations, the posterior is then estimated as

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) \approx \mathcal{N}(\boldsymbol{\theta}_{\text{SWA}}, \boldsymbol{\sigma}_{\text{SWA}}). \tag{13}$$

For MultiSWAG [19], M SWAG estimates are obtained from different initialisations, as in deep ensembles. Then, a Gaussian mixture model (GMM) is formed as a combination of each Gaussian SWAG posterior estimate $\mathcal{N}(\boldsymbol{\theta}^{(m)}_{\text{SWA}}, \boldsymbol{\sigma}^{(m)}_{\text{SWA}})$ where each component is equally weighed, as

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) \approx \sum_{m=1}^{M} \mathcal{N}(\boldsymbol{\theta}^{(m)}_{\text{SWA}}, \boldsymbol{\sigma}^{(m)}_{\text{SWA}}). \tag{14}$$

Need more

2.4 Continual Learning and Active Learning

In continual learning, data is collected and used to train the model continually, with the inferred posterior $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ (or its approximation) being used as the prior to the new posterior as in eq. (1).

[14]

In continual learning using VI, the computation of the ELBO (eq. (4)) requires the entropy of the variational approximation and the cross-entropy from the prior to the variational approximation. The entropy term is calculated in the same way for continual learning, however the cross-entropy term requires determining the cross-entropy between the variational approximation and itself (with different sets of parameters).

Active learning is...

For active learning

The *Max Entropy* acquisition function [7, 18] for selecting images maximises the entropy of the posterior predictive

$$\mathbb{H}\left[p(\boldsymbol{y} \mid \boldsymbol{y}_{\text{train}})\right] = -\sum_{c} p(y = c \mid \boldsymbol{y}_{\text{train}}) \log p(y = c \mid \boldsymbol{y}_{\text{train}}) \tag{15}$$

The *BALD* acquisition function maximises information gain

$$\mathbb{I} = \mathbb{H}\left[p(\boldsymbol{y} \mid \boldsymbol{y}_{\text{train}})\right] - \mathbb{E}_{p(\boldsymbol{\theta} \mid \boldsymbol{y}_{\text{train}})}\left[\mathbb{H}\left[p(\boldsymbol{y} \mid \boldsymbol{\theta})\right]\right] \tag{16}$$

3 Experiments

[15] Pyro is a probabilistic programming language (PPL) built upon Pytorch [1]. Tyxe is a BNN library built upon Pyro [17].

3.1 MNIST and SVHN

MNIST [12] and CIFAR [10]

The methods described in section 2.3 were trained on

To verify the confidence of the model on out-of-distribution (OOD) data, a network was trained on MNIST and tested on

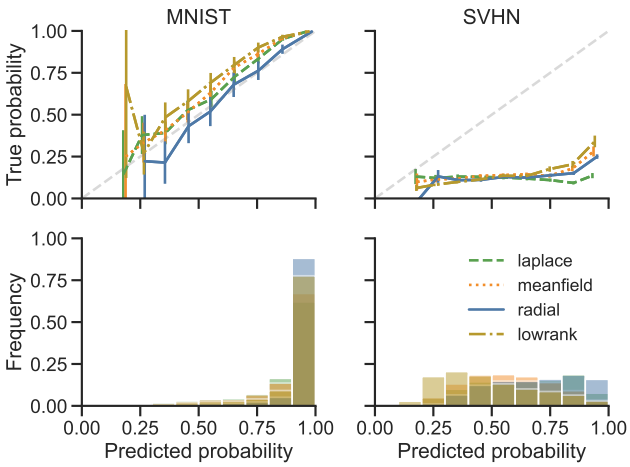

Figure 1: Caption

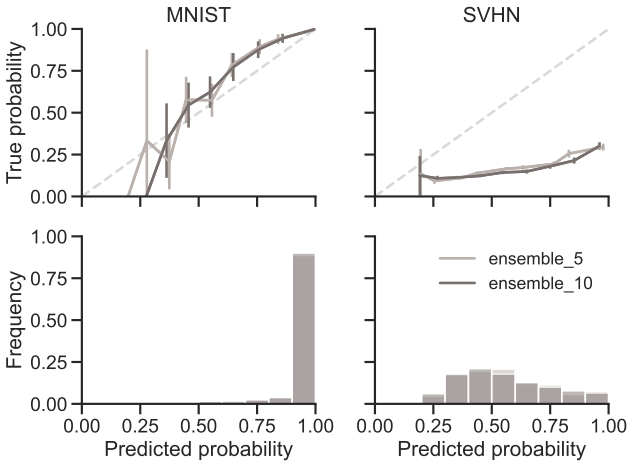

Figure 2: Caption

CIFAR-10. The NN had two fully-connected hidden layers with 64 neurons each, a ReLU non-linearity and a dropout layer after each.

The 95% uncertainty interval of calibration plots was calculated as per the binomial proportion confidence interval, i.e., $\pm 2\sqrt{p(1-p)/n}$, where p is the true probability of a bin and n is the number of predictions in a bin.

It can be seen that the radial approximation is more confident on MNIST than the Laplace and mean-field approximations (fig. 1). However, since MNIST is a relatively easy problem, all of the algorithms have an accuracy above 90%. Since most algorithms are overconfident, this leads to most predictions being made with very high confidence, and thus the overconfident algorithms have low ECE. Furthermore, all algorithms perform similarly on out-of-distribution (OOD) data, since all are extremely overconfident in their predictions on the SVHN dataset. This may suggest that the problem is with the model being used and not the inference algorithm.

3.2 MURA

Rajpurkar et al. [16] introduced MURA, a musculoskeletal radiograph X-ray dataset for computer vision. They trained a

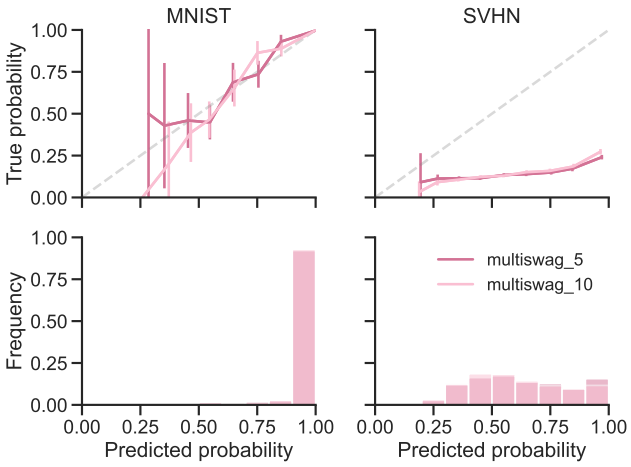

Figure 3: Caption

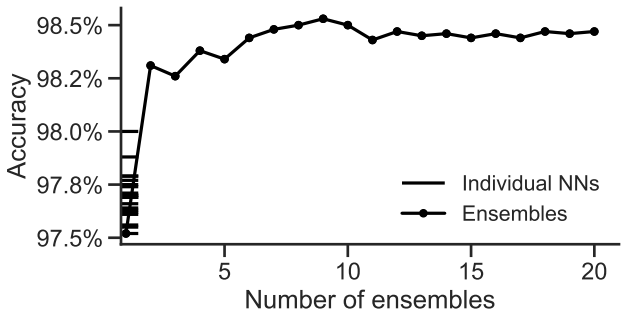

Figure 4: Ensembles with between one and twenty maximum likelihood NNs, trained on MNIST.

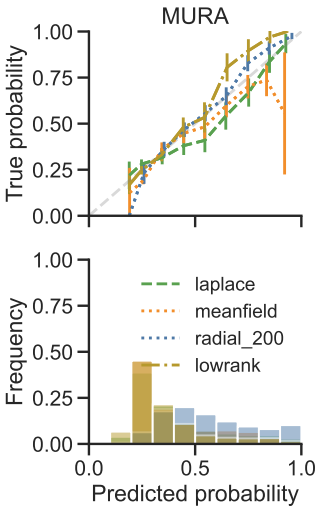

Figure 5: Caption

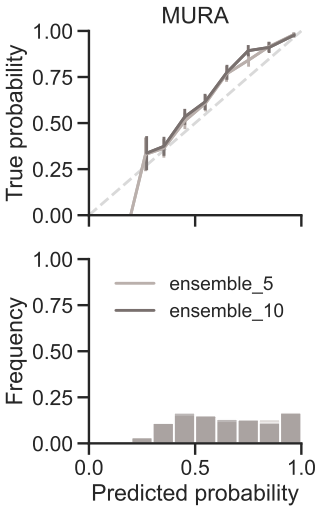

Figure 6: Caption

DenseNet with 169 layers [8] Due to computational constraints, we trained with the same dense neural network as in section 3.1.

3.3 Active learning

Foong et al. [6] found that, for shallow mean-field VI BNNs, active learning doesn't show improvement over random acquisition. This may be because VI for BNNs typically converge to a reasonable approximation of the mean faster than they approximate the variance. This means that, since we start the active learning process on an untrained BNN, the initial iterations quantify uncertainty poorly and therefore their predictive entropy has little value.

4 Conclusion

Ideas for experiments

1. Is it better to train an Nx bigger neural network, or to

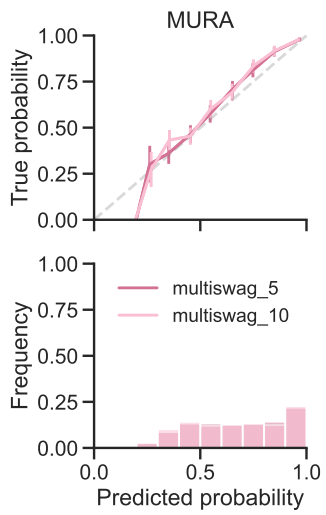

Figure 7: Caption

ensemble N neural networks? Mind you, deep ensembles have the advantage of easy parallelization.

2. See what happens with a mean field when you increase width or depth

3. See what happens with VI if you increase the size of the model, over-parameterise

References

[1] Eli Bingham et al. "Pyro: Deep Universal Probabilistic Programming". In: *Journal of Machine Learning Research* (2018).

[2] Beau Coker, Weiwei Pan, and Finale Doshi-Velez. "Wide Mean-Field Variational Bayesian Neural Networks Ignore the Data". In: *International Conference on Machine Learning* (2021).

[3] Erik Daxberger et al. "Laplace Redux-Effortless Bayesian Deep Learning". In: *Advances in Neural Information Processing Systems* (2021).

[4] Sebastian Farquhar, Michael A Osborne, and Yarin Gal. "Radial bayesian neural networks: Beyond discrete support in large-scale bayesian deep learning". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020.

[5] Sebastian Farquhar, Lewis Smith, and Yarin Gal. "Liberty or depth: Deep bayesian neural nets do not need complex weight posterior approximations". In: *Advances in Neural Information Processing Systems* (2020).

[6] Andrew Foong et al. "On the expressiveness of approximate inference in bayesian neural networks". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15897–15908.

[7] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. "Deep bayesian active learning with image data". In: *International Conference on Machine Learning*. PMLR. 2017.

[8] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[9] Pavel Izmailov et al. "What are Bayesian neural network posteriors really like?" In: *International Conference on Machine Learning*. PMLR. 2021.

[10] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[11] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in neural information processing systems* (2017).

[12] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* (1998).

[13] Wesley J Maddox et al. "A simple baseline for bayesian uncertainty in deep learning". In: *Advances in Neural Information Processing Systems* 32 (2019).

[14] Cuong V Nguyen et al. "Variational continual learning". In: *arXiv preprint arXiv:1710.10628* (2017).

[15] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems*.

[16] Pranav Rajpurkar et al. "Mura: Large dataset for abnormality detection in musculoskeletal radiographs". In: *arXiv preprint arXiv:1712.06957* (2017).

[17] Hippolyt Ritter and Theofanis Karaletsos. "TyXe: Pyro-based Bayesian neural nets for Pytorch". In: *arXiv preprint arXiv:2110.00276* (2021).

[18] Claude Elwood Shannon. "A mathematical theory of communication". In: *The Bell system technical journal* 27.3 (1948).

[19] Andrew G Wilson and Pavel Izmailov. "Bayesian deep learning and a probabilistic perspective of generalization". In: *Advances in neural information processing systems* (2020).

[20] Anqi Wu et al. "Deterministic variational inference for robust bayesian neural networks". In: *arXiv preprint arXiv:1810.03958* (2018).

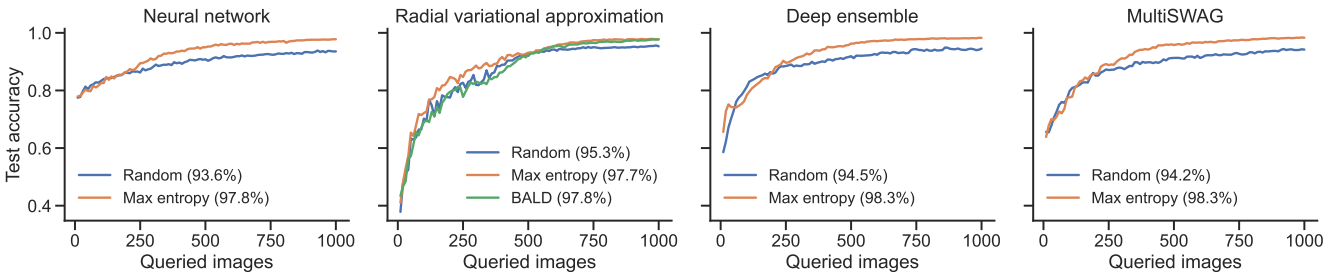

Figure 8: Trained on MNIST.

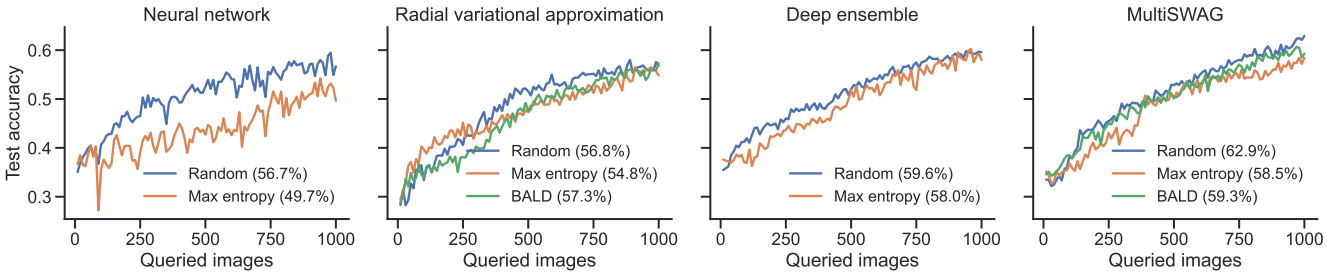

Figure 9: Trained on MURA.

A Appendix

Inference	Evaluated on	NLL	Accuracy	AUROC	Avg. Conf.	Avg. Conf. -	Avg. Conf. +	ECE
Ensemble@5	MNIST	-0.975	0.984	1.000	0.983	0.712	0.987	0.007
Ensemble@10	MNIST	-0.975	0.984	1.000	0.982	0.667	0.987	0.002
MultiSWAG@5	MNIST	-0.970	0.984	1.000	0.976	0.635	0.982	0.009
MultiSWAG@10	MNIST	-0.970	0.985	1.000	0.976	0.626	0.981	0.009
Radial	MNIST	-0.879	0.928	0.995	0.898	0.610	0.920	0.030
Mean-field	MNIST	-0.465	0.577	0.903	0.464	0.171	0.679	0.113
Low-rank	MNIST	-0.848	0.926	0.995	0.860	0.490	0.889	0.066
Laplace	MNIST	-0.839	0.975	1.000	0.844	0.438	0.854	0.131
ML	MNIST	-0.976	0.977	0.999	0.992	0.841	0.996	0.033

Table 1: Trained on MNIST.

Inference	Evaluated on	NLL	Accuracy	AUROC	Avg. Conf.	Avg. Conf. -	Avg. Conf. +	ECE
Ensemble@5	MURA	-0.649	0.764	0.954	0.733	0.536	0.794	0.037
Ensemble@10	MURA	-0.646	0.756	0.952	0.733	0.537	0.796	0.027
MultiSWAG@5	MURA	-0.642	0.772	0.955	0.723	0.528	0.780	0.050
MultiSWAG@10	MURA	-0.640	0.772	0.958	0.716	0.514	0.775	0.058
Radial	MURA	-0.441	0.577	0.871	0.566	0.462	0.642	0.024
Mean-field	MURA	-0.166	0.220	0.452	0.267	0.268	0.267	0.054
Low-rank	MURA	-0.295	0.399	0.791	0.415	0.344	0.523	0.058
Laplace	MURA	-0.521	0.672	0.929	0.610	0.469	0.679	0.062
ML	MURA	-0.573	0.701	0.926	0.683	0.535	0.746	0.021

Table 2: Trained on MURA.