3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–7

On Batch Normalisation for Approximate Bayesian Inference

Anonymous Authors
Anonymous Institution

Abstract

We study batch normalisation in the context of variational inference methods in Bayesian neural networks, such as mean-field or MC Dropout. We show that batch-normalisation does not affect the optimum of the evidence lower bound (ELBO). Furthermore, we study the Monte Carlo Batch Normalisation (MCBN) algorithm, proposed as an approximate inference technique parallel to MC Dropout, and show that for larger batch sizes, MCBN fails to capture epistemic uncertainty. Finally, we provide insight into what is required to fix this failure, namely having to view the mini-batch size as a variational parameter in MCBN. We comment on the asymptotics of the ELBO with respect to this variational parameter, showing that as dataset size increases towards infinity, the batch-size must increase towards infinity as well for MCBN to be a valid approximate inference technique.

1. Introduction

Deep neural networks have achieved tremendous success in several fields and are now being applied to various important real-life applications like medical diagnosis (Esteva et al., 2017) and autonomous driving (Levinson et al., 2011), where it is particularly important to produce reliable uncertainty estimates in addition to predictions. The Bayesian formalism (Neal, 2012; MacKay, 1992) provides a principled approach for modeling uncertainty in deep neural networks. However, performing exact Bayesian inference in deep neural networks is computationally intractable. In this regard, one of the well-known and accepted methods of performing approximate Bayesian inference is variational inference (VI) (Graves, 2011; Hinton and Van Camp, 1993) in which an approximating distribution is used instead of the true Bayesian posterior over model parameters.

There have been several works (Graves, 2011; Hernández-Lobato and Adams, 2015; Blundell et al., 2015; Gal and Ghahramani, 2016) which have tried to apply VI in deep neural networks. Among them, (Gal and Ghahramani, 2016) provided a practical solution by showing that any neural network trained using dropout layers is performing VI and can be treated as a Bayesian neural network. Their method, MC Dropout, uses dropout at test time to sample from the approximate posterior in a deep neural network.

Following on (Gal and Ghahramani, 2016), the paper (Teye et al., 2018) claimed that batch normalisation (BN) layers can also be used to cast deep neural networks as Bayesian models. Their method, Monte Carlo Batch Normalisation (MCBN) leverages the randomness of mini-batch statistics computed by BN layers due to the stochasticity of mini-batches drawn from the training set.

The running mini-batch statistics (mean and variance) of BN layers are stochastic estimators of the true **population statistics**. This observation has direct implications on how batch normalisation can or cannot be used to perform VI in deep neural networks. In this paper, we use this observation to first show that adding BN layers in deep neural networks

© A. Authors.

does not affect the probabilistic inference of existing VI methods like MC Dropout (Gal and Ghahramani, 2016) or Bayes by Backprop (Blundell et al., 2015). Secondly, we also expose a failure case of MCBN. In particular, we see that by increasing the mini-batch size, the epistemic uncertainty (Kendall and Gal, 2017) estimates produced by a model reduce to zero as BN layers become deterministic. We empirically see how this renders a model completely incapable of distinguishing between in-distribution and out-of-distribution samples. Finally, we provide insight into what is required to fix this failure, namely having to view the mini-batch size as a variational parameter in MCBN. We comment on the asymptotics of the ELBO with respect to this variational parameter, showing that as dataset size increases towards infinity, the batch-size must increase towards infinity as well for MCBN to be a valid approximate inference technique.

2. Batch Normalisation with Variational Inference

Background on Variational Inference (VI): In a Bayesian framework, we are interested in finding the posterior distribution $p(\theta|\mathbf{X}, \mathbf{Y})$ over model parameters θ given data $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\} = \{(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^N\}$. However, finding the true posterior for modern neural networks is computationally intractable. Hence, in variational inference, we find an approximating parameterised distribution $q_\phi(\theta)$ (also called the *variational distribution* with parameters ϕ) over model weights θ to approximate the true posterior. We want the variational distribution to be as close to the true posterior as possible. This is done by minimising the KL divergence $\mathrm{KL}(q_\phi(\theta)||p(\theta|\mathbf{X}, \mathbf{Y})$ between $q_\phi(\theta)$ and the true posterior $p(\theta|\mathbf{X}, \mathbf{Y})$. Minimising this KL divergence is equivalent to maximising the evidence lower bound (ELBO) (or minimising its negative):

$$\begin{aligned}\mathcal{L}_{\mathrm{ELBO}}(\phi) &= -\int q_\phi(\theta) \log p(\mathbf{Y}|\mathbf{X}, \theta) d\theta + \mathrm{KL}(q_\phi(\theta)||p(\theta)) \\ &= -\sum_{i=1}^{N} \int q_\phi(\theta) \log p(\mathbf{y}_i|\mathbf{x}_i, \theta) d\theta + \mathrm{KL}(q_\phi(\theta)||p(\theta))\end{aligned} \tag{1}$$

In variational inference, we minimise $\mathcal{L}_{\mathrm{ELBO}}(\phi)$ to find the optimal variational distribution $q_\phi^*(\theta)$ which explains our data. Note that the ELBO objective ($\mathcal{L}_{\mathrm{ELBO}}(\phi)$) is over variational parameters ϕ and not model weights θ.

Does Batch Normalisation (BN) impact VI? The batch normalisation (Ioffe and Szegedy, 2015; LeCun et al., 2012) operation is used to standardise each layer's input to have zero mean and unit variance. Given input $\mathbf{h}^l$ for a fully connected layer l, the BN operation uses two data-dependent statistics: the mini-batch mean $\boldsymbol{\mu}_B^l$ and the mini-batch variance $(\boldsymbol{\sigma}^2)_B^l$ to compute:

$$\hat{\mathbf{h}}^l = \left(\frac{\mathbf{h}^l - \boldsymbol{\mu}_B^l}{\sqrt{(\boldsymbol{\sigma}^2)_B^l}} \right) \boldsymbol{\gamma}^l + \boldsymbol{\beta}^l \tag{2}$$

where $\boldsymbol{\gamma}^l$ and $\boldsymbol{\beta}^l$ are the scale and shift parameters which are learnt through backpropagation. If we compute $\mathcal{L}_{\mathrm{ELBO}}(\phi)$ using the entire dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^N\}$ as shown in equation 1, the mini-batch statistics $\{\boldsymbol{\mu}_B^l, (\boldsymbol{\sigma}^2)_B^l\}$ are exactly equal to the population statistics $\{\boldsymbol{\mu}^l, (\boldsymbol{\sigma}^2)^l\}$ and hence, are fixed. However, in practical situations, the entire dataset is

not used while optimising $\mathcal{L}_{\text{ELBO}}(\phi)$ and mini-batch optimisation methods (Hoffman et al., 2013) are employed. In such cases, the $\hat{\mathcal{L}}_{\text{ELBO}}(\phi)$ computed from a minibatch B is:

$$\hat{\mathcal{L}}_{\text{ELBO}}(\phi) = -\frac{N}{|B|} \sum_{i \in B} \int q_\phi(\theta) \log p(\mathbf{y}_i | \mathbf{x}_i, \theta) d\theta + \text{KL}(q_\phi(\theta) || p(\theta)). \tag{3}$$

During training, BN layers keep a running average of means and variances (batch statistics) which get updated with each incoming mini-batch from the training set (Ioffe and Szegedy, 2015). Following the law of large numbers, in expectation, the running batch statistics converge to the population statistics $\{\boldsymbol{\mu}^l, (\boldsymbol{\sigma}^2)^l\}$ and hence, $\hat{\mathcal{L}}_{\text{ELBO}}(\phi)$ computed from a mini-batch forms an unbiased estimator of $\mathcal{L}_{\text{ELBO}}(\phi)$ computed using the entire dataset, i.e., $\mathbb{E}_B[\hat{\mathcal{L}}_{\text{ELBO}}(\phi)] = \mathcal{L}_{\text{ELBO}}(\phi)$. There is no change to the optimum of the ELBO $\mathcal{L}_{\text{ELBO}}(\phi)$ due to the presence of BN layers. Thus, batch normalisation does not impact the probabilistic inference of other VI methods like Mean Field VI (Graves, 2011; Blundell et al., 2015) or MC Dropout (Gal and Ghahramani, 2016).

3. Stochastic Batch Normalisation (MCBN) as VI

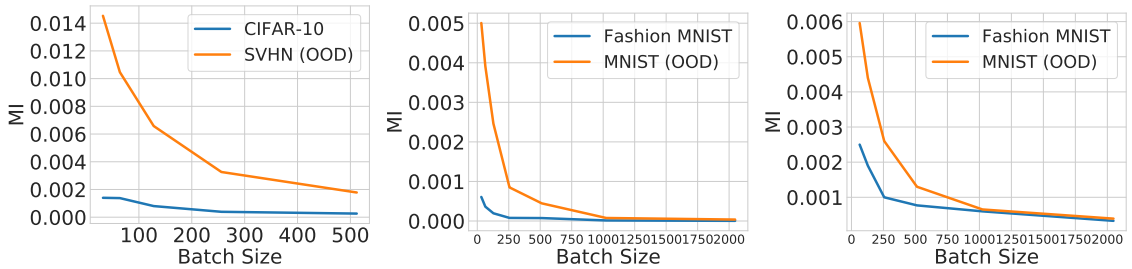
What is MCBN? In the work (Teye et al., 2018), the authors propose using batch normalisation as a way of performing VI in deep neural networks. In a nutshell, they categorise the parameters of a neural network into:

1. *Learnable Parameters:* $\theta = \{(\mathbf{W}^l, \boldsymbol{\beta}^l, \boldsymbol{\gamma}^l)_{l=1}^L\}$, weight matrices and $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ BN parameters learnable through backpropagation for a neural network with L layers.

2. *Stochastic Parameters:* $w = \{(\boldsymbol{\mu}_B^l, (\boldsymbol{\sigma}^2)_B^l)_{l=1}^L\}$, the running average batch statistics (mean and variance) for all BN layers.

In their method, they first train a network with BN layers using a conventional optimiser (like SGD or Adam etc.) and loss function (like cross entropy). During test time, they keep the BN layers in training mode and for each test sample $\mathbf{x}^*$, they perform multiple stochastic forward passes using different, randomly chosen mini-batches from the training set. Each such forward pass corresponds to a particular instantiation of $w = \{(\boldsymbol{\mu}_B^l, (\boldsymbol{\sigma}^2)_B^l)_{l=1}^L\}$ and provides a single output $\hat{\mathbf{y}}_t^*$ (for the tth stochastic forward pass, say). The mean $\mathbb{E}[\mathbf{y}] \approx \frac{1}{T}\sum_{t=1}^T \hat{\mathbf{y}}_t^*$ is then used as the prediction and some measure of variance of these outputs can be interpreted as an uncertainty estimate. It is important to note here that the mini-batch size used during stochastic forward passes at test time is the same as the mini-batch size used for training. This method was named *Monte Carlo Batch Normalisation* (MCBN).

Failure case of MCBN: In MCBN, the randomness of model outputs at test time is solely dependent on the stochastic parameters $w = \{(\boldsymbol{\mu}_B^l, (\boldsymbol{\sigma}^2)_B^l)_{l=1}^L\}$, which in turn are dependent on the mini-batch size. As mentioned before, in case of larger mini-batch sizes, the batch statistics will converge to the population statistics, which is in fact desirable for BN layers following their original motivation. When the batch size equals the dataset size, there will be no stochasticity at all, and 'full BN' is recovered (the optimal case for regularisation). However, in this case, any measure of epistemic uncertainty which depends on variation of model outputs over stochastic forward passes will reduce to zero and hence, the resulting model won't be able to capture epistemic uncertainty at all.



(a) (b) (c)

Figure 1: Mutual Information (MI) between predictive and posterior distributions plotted against mini-batch size for (a) a ResNet-50 trained on CIFAR-10 and tested on SVHN (OOD), (b) ResNet-50 trained on Fashion MNIST and tested on MNIST (OOD) and (c) VGG-16 trained on Fashion MNIST and tested on MNIST (OOD).

We empirically validate this claim by showing the effect of increasing the mini-batch size in MCBN on the epistemic uncertainty of the model. As epistemic uncertainty should be high on unseen data (Kendall and Gal, 2017), we choose a notably difficult pair of in-distribution and OOD datasets (Nalisnick et al., 2019), (CIFAR-10 (Krizhevsky et al., 2009) vs. SVHN (Netzer et al., 2011)) and (Fashion-MNIST (Xiao et al., 2017) vs MNIST (LeCun)). We train a ResNet-50 on CIFAR-10 and Fashion-MNIST and we train a VGG-16 on Fashion-MNIST using different training batch sizes from 32 to 2048 (see Table 1 for specific batch sizes). We use SGD with a momentum of 0.9 as the optimiser and cross-entropy as the objective function. We train the ResNet-50 on CIFAR-10 for 350 epochs with an initial learning rate of 0.1 and drop the learning rate by a factor of 10 at epochs 150 and 250. For the models trained on Fashion-MNIST we use 100 training epochs and drop the learning rate at epochs 40 and 60. During test time, we use 20 stochastic forward passes to make predictions for each model.

In order to compute the epistemic uncertainty, we use the mutual information (Gal, 2016) between the predictive and the posterior distributions as the uncertainty estimate of choice. In Table 1, we show the test set errors of the models for every batch size. Note that we did not train ResNet-50 on CIFAR-10 with batch sizes 1024 and 2048 due to compute constraints. Finally, we show the variation of epistemic uncertainty with batch size for both in-distribution and OOD datasets in Figure 1.

It is very clear from Figure 1 that *with increase in batch sizes, the model's epistemic uncertainty lowers to the point where it can no longer distinguish between in and out of distribution samples.* In fact, for the models trained on Fashion MNIST, we find the epistemic uncertainty to almost become 0 for batch sizes 1024 and 2048. Thus, although the test set accuracy remains more or less the same, MCBN fails to capture epistemic uncertainty with growing mini-batch size.

Insight into what is required to fix this failure: The model's stochasticity in MCBN is clearly dependent on the mini-batch size used for training. Hence, the mini-batch size itself should be considered as a variational parameter much like the dropout rate is a variational parameter in MC Dropout (Gal and Ghahramani, 2016). One can find the best dropout rate either by learning it directly (Gal et al., 2017) or performing grid-search over possible

Architecture	Dataset	OOD Dataset	Mini-batch size						
			32	64	128	256	512	1024	2048
ResNet-50	CIFAR-10	SVHN	0.9366	0.9459	0.9517	0.9547	0.9545	-	-
	Fashion-MNIST	MNIST	0.9343	0.9377	0.9386	0.9396	0.9359	0.9365	0.9387
VGG-16	Fashion-MNIST	MNIST	0.9429	0.9436	0.9437	0.9427	0.9438	0.9450	0.9486

Table 1: Test set accuracy computed for different MCBN mini-batch sizes.

dropout rates (Mukhoti et al., 2018). Similarly, one could think of performing a grid-search over possible mini-batch sizes and find the one which maximises the ELBO.

Comment on ELBO asymptotics: We briefly comment on the asymptotics of the ELBO with respect to this variational parameter. As the dataset size increases towards infinity, we expect epistemic uncertainty of a parametric model to diminish to zero. However, as we commented above, for the epistemic uncertainty of MCBN to diminish to zero, the batch-size must increase towards infinity as well. This may be impractical in certain applications.

4. Conclusion

In this paper we have studied how batch normalisation layers fit into the context of variational inference in deep neural networks. Mini-batch statistics in BN layers are stochastic estimators of the population statistics. We use this observation to show that batch normalisation does not affect the probabilistic inference of VI approaches. Furthermore, we use the same observation to expose a failure case of the MCBN (Teye et al., 2018) algorithm and see that for bigger batch sizes, MCBN is unable to produce sensible epistemic uncertainty and cannot distinguish in and out of distribution samples. Finally, we provide some insights into how to fix the problem and leave the development of more involved approaches on MCBN as future work.

References

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

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

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.

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in neural information processing systems*, pages 3581–3590, 2017.

Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

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.

Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168. IEEE, 2011.

David JC MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992.

Jishnu Mukhoti, Pontus Stenetorp, and Yarin Gal. On the importance of strong baselines in bayesian deep learning. *arXiv preprint arXiv:1811.09385*, 2018.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Hybrid models with deep and invertible features. *arXiv preprint arXiv:1902.02767*, 2019.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Mattias Teye, Hossein Azizpour, and Kevin Smith. Bayesian uncertainty estimation for batch normalized deep networks. volume 80 of *Proceedings of Machine Learning Research*, pages 4907–4916, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/teye18a.html.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.