2nd Symposium on Advances in Approximate Bayesian Inference, 2019 1–14

The k-tied Normal Distribution: A Compact Parameterization of Gaussian Mean Field Posteriors in Bayesian Neural Networks

Jakub Świątkowski[*] JAKUB.SWIATKOWSKI@MIMUW.EDU.PL
University of Warsaw

Kevin Roth[*] KEVIN.ROTH@INF.ETHZ.CH
ETH Zurich

Bastiaan S. Veeling[*] BASVEELING@GMAIL.COM
University of Amsterdam

Linh Tran[*] LINH.TRAN@IMPERIAL.AC.UK
Imperial College London

Joshua V. Dillon JVDILLON@GOOGLE.COM
Jasper Snoek JSNOEK@GOOGLE.COM
Google Research

Stephan Mandt[†] STEPHAN.MANDT@GMAIL.COM
University of California, Irvine

Tim Salimans SALIMANS@GOOGLE.COM
Rodolphe Jenatton RJENATTON@GOOGLE.COM
Sebastian Nowozin NOWOZIN@GOOGLE.COM
Google Research

Abstract

Variational Bayesian Inference is a popular methodology for approximating posterior distributions over Bayesian neural network weights. Recent work developing this class of methods has explored ever richer parameterizations of the approximate posterior in the hope of improving performance. In contrast, here we share a curious experimental finding that suggests instead restricting the variational distribution to a more compact parameterization. For a variety of deep Bayesian neural networks trained using Gaussian mean-field variational inference, we find that the posterior standard deviations consistently exhibit strong low-rank structure after convergence. This means that by decomposing these variational parameters into a low-rank factorization, we can make our variational approximation more compact without decreasing the models' performance. Furthermore, we find that such factorized parameterizations improve the signal-to-noise ratio of stochastic gradient estimates of the variational lower bound, resulting in faster convergence.

1. Introduction

Bayesian Neural Networks (MacKay, 1992; Neal, 1993) explicitly represent their parameter-uncertainty by forming a *posterior distribution* over model parameters, instead of relying on a single point estimate for making predictions, as is done in traditional deep learning. Besides offering improved predictive performance over single models, Bayesian neural networks are also more robust

[*] Work done while at Google Research.
[†] Work done while visiting Google Research.

to hard examples (Raftery et al., 2005), have better calibration of predictive uncertainty and thus can be used for out-of-domain detection or other risk-sensitive applications (Ovadia et al., 2019).

Variational inference (Peterson, 1987; Hinton and Van Camp, 1993) is a popular class of methods for approximating the posterior distribution $p(\mathbf{w}|\mathbf{x}, \mathbf{y})$, since the exact Bayes' rule is often intractable to compute for models of practical interest. This class of methods specifies a distribution $q_{\boldsymbol{\theta}}(\mathbf{w})$ of given parametric or functional form as the posterior approximation, and optimizes the approximation by solving an optimization problem. In particular, we minimize the negative Evidence Lower Bound (negative ELBO) approximated by samples from the posterior:

$$L_q \approx D_{\mathrm{KL}}[q_{\boldsymbol{\theta}}(\mathbf{w})||p(\mathbf{w})] - \frac{1}{S}\sum_{s=1}^{S} \log p(\mathbf{y}|\mathbf{w}^{(s)}, \mathbf{x}), \quad \mathbf{w}^{(s)} \sim q_{\boldsymbol{\theta}}(\mathbf{w}), \tag{1}$$

by differentiating with respect to the variational parameters $\boldsymbol{\theta}$ (Salimans et al., 2013; Kingma and Welling, 2013).

In **Gaussian Mean Field Variational Inference** (GMFVI) (Blei et al., 2017; Blundell et al., 2015), we choose the variational approximation to be a fully factorized Gaussian distribution:

$$q(\mathbf{W}) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) = \prod_{i=1}^{m}\prod_{j=1}^{n} q(w_{ij}), \quad \text{with} \quad q(w_{ij}) = \mathcal{N}(\mu_{ij}, \sigma_{ij}^2) \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{m\times n}$ is a weight matrix of a single network layer and i and j are the row and column indices in this weight matrix. In practice, we often represent the posterior standard deviation parameters σ_{ij} in the form of a matrix $\mathbf{A} \in \mathbb{R}_+^{m\times n}$. With this notation, we have the relationship $\boldsymbol{\Sigma}_q = \mathrm{diag}(\mathrm{vec}(\mathbf{A}^2))$ where the elementwise-squared $\mathbf{A}$ is vectorized by stacking its columns, and then expanded as a diagonal matrix into $\mathbb{R}_+^{mn\times mn}$. While Gaussian Mean-Field posteriors are considered to be one of the simplest types of variational approximations, with some known limitations (Giordano et al., 2018), they scale to comparatively large models and generally provide competitive performance (Ovadia et al., 2019). However, when compared to deterministic neural networks, GMFVI doubles the number of parameters and is often harder to train due to the increased noise in stochastic gradient estimates.

Beyond fully factorized mean-field, recent research in variational inference has explored richer parameterizations of the approximate posterior in order to improve the performance of Bayesian neural networks (see Appendix A and Figure 3). For instance, various structures of Gaussian posteriors have been proposed, with per layer block-structured covariances (Louizos and Welling, 2016; Sun et al., 2017; Zhang et al., 2017), full covariances (Barber and Bishop, 1998) with different parametrizations (Seeger, 2000), up to more flexible approximate posteriors using normalizing flows (Rezende and Mohamed, 2015) and extensions thereof (Louizos and Welling, 2017). In contrast, here we study a simpler, more compactly parameterized mean-field variational posterior which ties variational parameters in the already diagonal covariance matrix. We show that such a posterior approximation can also work well for a variety of models. In particular we find that:

- Converged posterior standard deviations under GMFVI consistently display strong low-rank structure. This means that by decomposing these variational parameters into a low-rank factorization, we can make our variational approximation more compact without decreasing our model's performance.
- Factorized parameterizations of posterior standard deviations improve the signal-to-noise ratio of stochastic gradient estimates, and thus not only reduce the number of parameters compared to standard GMFVI, but also can lead to faster convergence.

2. The k-tied Normal Distribution: Exploiting Low-Rank Parameter-Structure in Mean Field Posteriors

We start by empirically studying the properties of the spectrum of posterior standard deviation matrices $\mathbf{A}$, *post training*, in models already trained until convergence using standard fully-parameterized Gaussian mean-field variational distributions. Interestingly, we observe that those matrices naturally exhibit a low-rank structure (see Figure 1), i.e,

$$\mathbf{A} \approx \mathbf{U}\mathbf{V}^T \tag{3}$$

for some $\mathbf{U} \in \mathbb{R}^{m \times k}$, $\mathbf{V} \in \mathbb{R}^{n \times k}$ and k a small value (e.g., 2 or 3). This observation motivates the introduction of the following variational family, which we name k-tied Normal:

$$k\text{-}tied\text{-}\mathcal{N}(\mathbf{W}; \boldsymbol{\mu}_q, \mathbf{U}, \mathbf{V}) = \mathcal{N}\Big(\boldsymbol{\mu}_q, \operatorname{diag}\Big(\operatorname{vec}\big((\mathbf{U}\mathbf{V}^T)^2\big)\Big)\Big), \tag{4}$$

where the squaring of the matrix $\mathbf{U}\mathbf{V}^T$ is applied elementwise. Due to the tied parametrization of the diagonal covariance matrix, we emphasize that this variational family is *smaller*—i.e., included in—the standard Gaussian mean-field variational distribution family. Interestingly, we find that despite its compactness, our posterior is able to match the fully parametrized GMFVI in terms of ELBO and predictive performance both in a post training approximation (see Figure 1) and when training the tied parameters $\mathbf{U}$ and $\mathbf{V}$ from a random initialization (see Figure 2).

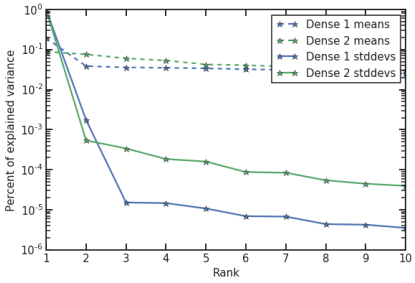
Furthermore, the total number of the standard deviation parameters in our method is $k(m + n)$ from $\mathbf{U}$ and $\mathbf{V}$, compared to mn for $\mathbf{A}$ in the standard GMFVI parametrization. Given that in our experiments the k is very low (e.g $k = 2$) this reduces the number of standard deviation parameters from quadratic to linear in the dimensions of the layer, see Table 1. More importantly, such parameter sharing across the weights leads to higher signal-to-noise ratio during training and thus in some cases faster convergence, see Figure 2. Finally, the matrix variate Gaussian distribution (Gupta and Nagar, 2018), referred to as $\mathcal{MN}$ and already used for variational inference in the most closely related work of Louizos and Welling (2016) and Sun et al. (2017), is similar to our k-tied Normal distribution when $k = 1$ (see also Figure 3). Interestingly, we prove that for $k \geq 2$, our k-tied Normal distribution cannot be represented by any $\mathcal{MN}$ distribution (see Appendix B).

3. Experimental results

We now provide a short description of the experimental setting and more detailed experimental results. In our experiments we use three model types: a 3 layer Multilayer Perceptron (MLP) trained on the MNIST dataset (LeCun and Cortes, 2010), a LeNet-type Convolutional Neural Network (CNN) (LeCun et al., 1998) trained on the CIFAR-100 dataset (Krizhevsky et al., 2009), and a vanilla LSTM model (Hochreiter and Schmidhuber, 1997) trained on the IMDB dataset (Maas et al., 2011). Appendix E provides more details about the experimental setting. We highlight that our experiments focus primarily on the comparison across a broad range of model types rather than competing with the state-of-the-art results over the specifically used datasets. Therefore, we use small to medium models that are known to train well using the standard GMFVI approach explored in this paper. Scaling GMFVI to larger model sizes is still a challenging research problem (Osawa et al., 2019).

Figure 1 shows that GMFVI applied to the LeNet CNN learns posterior standard deviation matrices of the CNN's dense layers that have most of their variance explained[1] by the first two components of their SVD decomposition. Furthermore, we also see that these matrices can be approximated post training by their low-rank SVD decompositions with little ELBO and predictive performance loss. In Appendix C we show that these results also hold for the analyzed MLP and LSTM models.

Figure 2 shows the results of exploiting the above observation by applying the k-tied Normal posterior during GMFVI training. We see that for $k \geq 2$, the k-tied Normal posterior is able to achieve the performance competitive with the standard GMFVI posterior parametrization, while reducing the total number of model parameters. The benefits of using the k-tied Normal posterior are most visible for models where the dense layers with the k-tied Normal posterior constitute a significant portion of the total number of the model parameters (e.g. MLPs and CNNs with dense layers for classification). Furthermore, we observe a significant increase in the signal-to-noise ratio[2] (SNR) of the gradients of parameters of the GMFVI posterior standard deviations when using the k-tied Normal posterior. Importantly, we also see that the increase in the gradient SNR translates into faster convergence of the negative ELBO objective in some of the analyzed models.



Rank k	-ELBO ↓	NLL ↓	Accuracy ↑
Full	$3.83_{\pm 0.020}$	$2.23_{\pm 0.017}$	$42.1_{\pm 0.49}$
1	$4.33_{\pm 0.021}$	$2.30_{\pm 0.016}$	$41.7_{\pm 0.49}$
2	$3.88_{\pm 0.020}$	$2.24_{\pm 0.017}$	$42.2_{\pm 0.49}$
3	$3.86_{\pm 0.020}$	$2.24_{\pm 0.017}$	$42.1_{\pm 0.49}$

Figure 1: Posterior standard deviations, in contrast to posterior means, of dense layers in LeNet CNN trained using standard GMFVI display strong low-rank structure and can be approximated without loss to predictive metrics. Top: Explained variance[1] per singular value from SVD of matrices of converged posterior means and standard deviations. Bottom: Impact of post training low-rank approximation of the posterior standard deviation matrices on model's performance. We report mean and standard error of the mean (SEM) for each metric across 100 models samples.

4. Conclusion

In this work we have shown that Bayesian Neural Networks trained with standard Gaussian mean-field variational inference exhibit posterior standard deviation matrices that can be approximated with little information loss by a low-rank decomposition. This suggests that richer parameterizations of the variational posterior may not always be needed, and that compact parameterizations can also work well. We used this insight to propose a simple, yet effective variational posterior parametrization, which speeds up training and reduces the number of variational parameters without degrading predictive performance on three different model types.

In future work, we hope to scale up variational inference with compactly parameterized approximate posteriors to much larger models and more complex problems. For mean-field variational inference to work well in that setting several challenges will likely need to be addressed (Osawa et al., 2019); improving the signal-to-noise ratio of ELBO gradients using our compact variational parameterizations may provide a piece of the puzzle.

1. Explained variance for the rank k approximation is calculated as $\gamma_k^2 / \sum_{i'} \gamma_{i'}^2$, where $\gamma_{i'}$ are singular values.
2. SNR for each gradient value is calculated as $E[g_b^2]/\mathrm{Var}[g_b^2]$, where g_b is the gradient value for a single parameter. The expectation E and variance Var of the gradient values g_b are calculated over a window of last 10 batches.

Model & Dataset	Rank k	-ELBO ↓	NLL ↓	Accuracy ↑	#Par. [k] ↓
MNIST, MLP	full	$0.501_{\pm 0.0061}$	$0.133_{\pm 0.0040}$	$96.8_{\pm 0.18}$	957
MNIST, MLP	1	$0.539_{\pm 0.0063}$	$0.155_{\pm 0.0043}$	$96.1_{\pm 0.19}$	482
MNIST, MLP	2	$0.520_{\pm 0.0063}$	$0.129_{\pm 0.0039}$	$96.8_{\pm 0.18}$	484
MNIST, MLP	3	$0.497_{\pm 0.0060}$	$0.120_{\pm 0.0038}$	$96.9_{\pm 0.18}$	486
CIFAR100, CNN	full	$3.72_{\pm 0.018}$	$2.16_{\pm 0.016}$	$43.9_{\pm 0.50}$	4,405
CIFAR100, CNN	1	$3.65_{\pm 0.017}$	$2.12_{\pm 0.015}$	$45.5_{\pm 0.50}$	2,262
CIFAR100, CNN	2	$3.76_{\pm 0.019}$	$2.15_{\pm 0.016}$	$44.3_{\pm 0.50}$	2,268
CIFAR100, CNN	3	$3.73_{\pm 0.018}$	$2.13_{\pm 0.016}$	$44.3_{\pm 0.50}$	2,273
IMDB, LSTM	full	$0.538_{\pm 0.0054}$	$0.478_{\pm 0.0052}$	$79.5_{\pm 0.26}$	2,823
IMDB, LSTM	1	$0.592_{\pm 0.0041}$	$0.512_{\pm 0.0040}$	$77.6_{\pm 0.26}$	2,693
IMDB, LSTM	2	$0.560_{\pm 0.0042}$	$0.484_{\pm 0.0041}$	$78.2_{\pm 0.26}$	2,694
IMDB, LSTM	3	$0.550_{\pm 0.0051}$	$0.491_{\pm 0.0050}$	$78.8_{\pm 0.26}$	2,695

Rank k	MNIST, MLP Dense 2, SNR at step		
	1000	5000	9000
full	$4.13_{\pm 0.027}$	$4.45_{\pm 0.091}$	$3.21_{\pm 0.035}$
1	$5840_{\pm 190}$	$158_{\pm 3.8}$	$5.3_{\pm 0.20}$
2	$7500_{\pm 240}$	$140_{\pm 11}$	$4.3_{\pm 0.26}$
3	$7000_{\pm 270}$	$117_{\pm 1.7}$	$4.1_{\pm 0.20}$

Rank k	MNIST, MLP, -ELBO at step		
	1000	5000	9000
full	$42.16_{\pm 0.070}$	$26.52_{\pm 0.016}$	$15.39_{\pm 0.016}$
1	$43.11_{\pm 0.039}$	$14.85_{\pm 0.017}$	$2.06_{\pm 0.027}$
2	$42.74_{\pm 0.090}$	$13.97_{\pm 0.023}$	$1.82_{\pm 0.017}$
3	$42.63_{\pm 0.068}$	$13.61_{\pm 0.020}$	$1.80_{\pm 0.031}$

Figure 2: Left: impact of the k-tied Normal posterior on test ELBO, test predictive performance and number of model parameters. Test performance is reported as a mean and SEM across 100 weights samples after training each model for ≈300 epochs. Right top: mean gradient SNR in the Dense 2 layer of the MNIST MLP model at increasing training steps for different ranks of tying k. We observe a similar increase in the SNR from tying for the CNN and the LSTM models as for the MLP model shown here. We report mean and SEM across 3 training runs with different random seeds. Right bottom: Negative ELBO on the MNIST validation data set at increasing training steps for different ranks of tying k. See also Figure 6, which shows negative ELBO convergence plots for the all three models types.

References

David Barber and Christopher M Bishop. Ensemble learning for multi-layer networks. In *Advances in neural information processing systems*, pages 395–401, 1998.

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.

Edward Challis and David Barber. Concave gaussian variational approximations for inference in large-scale bayesian linear models. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 199–207, 2011.

François Chollet et al. Keras, 2015.

Ryan Giordano, Tamara Broderick, and Michael I Jordan. Covariances, robustness and variational bayes. *The Journal of Machine Learning Research*, 19(1):1981–2029, 2018.

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

Arjun K Gupta and Daya K Nagar. *Matrix variate distributions*. Chapman and Hall/CRC, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

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 Hinton and Drew Van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *in Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*. Citeseer, 1993.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Tommi Jaakkola and Michael Jordan. A variational approach to bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, volume 82, page 4, 1997.

Mohammad Emtiyaz Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable bayesian deep learning by weight-perturbation in adam. *arXiv preprint arXiv:1806.04854*, 2018.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pages 1708–1716, 2016.

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2218–2227. JMLR. org, 2017.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

Benjamin M Marlin, Mohammad Emtiyaz Khan, and Kevin P Murphy. Piecewise bounds for estimating bernoulli-logistic latent gaussian models. In *Proceedings of the International Conference on Machine Learning*, pages 633–640, 2011.

Andrew C Miller, Nicholas J Foti, and Ryan P Adams. Variational boosting: Iteratively refining posterior approximations. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2420–2429. JMLR. org, 2017.

Aaron Mishkin, Frederik Kunstner, Didrik Nielsen, Mark Schmidt, and Mohammad Emtiyaz Khan. Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient. In *Advances in Neural Information Processing Systems*, pages 6245–6255, 2018.

Radford M Neal. Bayesian learning via stochastic dynamics. In *Advances in neural information processing systems*, pages 475–482, 1993.

Victor M-H Ong, David J Nott, and Michael S Smith. Gaussian variational approximation with a factor covariance structure. *Journal of Computational and Graphical Statistics*, 27(3):465–478, 2018.

Kazuki Osawa, Siddharth Swaroop, Anirudh Jain, Runa Eschenhagen, Richard E Turner, Rio Yokota, and Mohammad Emtiyaz Khan. Practical deep learning with bayesian principles. *arXiv preprint arXiv:1906.02506*, 2019.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

Carsten Peterson. A mean field theory learning algorithm for neural networks. *Complex systems*, 1: 995–1019, 1987.

Adrian E Raftery, Tilmann Gneiting, Fadoua Balabdaoui, and Michael Polakowski. Using bayesian model averaging to calibrate forecast ensembles. *Monthly weather review*, 133(5):1155–1174, 2005.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

Tim Salimans, David A Knowles, et al. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Matthias Seeger. Bayesian model selection for support vector machines, gaussian processes and other kernel classifiers. In *Advances in neural information processing systems*, pages 603–609, 2000.

Shengyang Sun, Changyou Chen, and Lawrence Carin. Learning structured weight uncertainty in bayesian neural networks. In *Artificial Intelligence and Statistics*, pages 1283–1292, 2017.

Linda SL Tan and David J Nott. Gaussian variational approximation with sparse precision matrices. *Statistics and Computing*, 28(2):259–275, 2018.

Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *International conference on machine learning*, pages 1971–1979, 2014.

Richard Turner and Maneesh Sahani. *Two problems with variational expectation maximisation for time-series models*, pages 109–130. Cambridge University Press, 2011.

Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. *arXiv preprint arXiv:1712.02390*, 2017.

Appendix A. More details on the related work

The application of variational inference to neural networks dates back at least to Peterson (1987); Hinton and Van Camp (1993). Many developments[3] have followed those seminal research efforts, in particular regarding (1) the expressiveness of the variational posterior distribution and (2) the way the variational parameters themselves can be structured to lead to compact, easier-to-learn and scalable formulations. We organize the discussion of this section around those two aspects, with a specific focus on the Gaussian case. For a graphical overview of the related work see Figure 3.

Full Gaussian posterior. Because of their substantial memory and computational cost, Gaussian variational distributions with full covariance matrices have been primarily applied to (generalized) linear models and shallow neural networks (Jaakkola and Jordan, 1997; Barber and Bishop, 1998; Marlin et al., 2011; Titsias and Lázaro-Gredilla, 2014; Miller et al., 2017; Ong et al., 2018). To represent the dense covariance matrix efficiently in terms of variational parameters, several schemes have been proposed, including the sum of low-rank plus diagonal matrices (Barber and Bishop, 1998; Seeger, 2000; Miller et al., 2017; Zhang et al., 2017; Ong et al., 2018), the Cholesky decomposition (Challis and Barber, 2011) or by operating instead on the precision matrix (Tan and Nott, 2018; Mishkin et al., 2018).

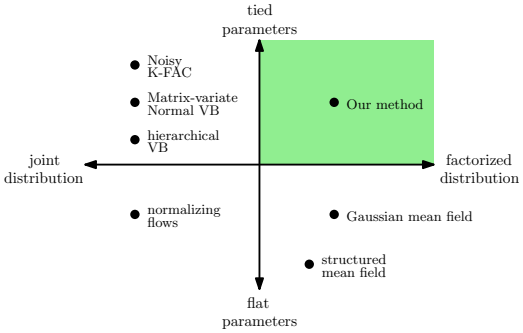


Figure 3: Approaches to variational Bayes on Bayesian neural networks, ordered by i) whether they factorize the variational distribution q, and ii) whether they tie the variational parameters.

3. We refer the interested readers to Zhang et al. (2018) for a recent review of variational inference.

Gaussian posterior with block-structured covariances. In the context of Bayesian neural networks, the layers represent a natural structure to be exploited by the covariance matrix. When assuming independence across layers, the resulting covariance matrix exhibits a *block-diagonal structure* that has been shown to be a well-performing simplification of the dense setting (Sun et al., 2017; Zhang et al., 2017), with both memory and computational benefits. Within each layer, the corresponding diagonal block of the covariance matrix can be represented by a Kronecker product of two smaller matrices (Louizos and Welling, 2016; Sun et al., 2017), possibly with a parametrization based on rotation matrices (Sun et al., 2017). Finally, using similar techniques, Zhang et al. (2017) proposed to use a block tridiagonal structure that better approximates the behavior of a dense covariance.

Fully factorized mean-field Gaussian posterior. A fully factorized Gaussian variational distribution constitutes the simplest option for variational inference. The resulting covariance matrix is diagonal and all underlying parameters are assumed to be independent. While the mean-field assumption is known to have some limitations—e.g., underestimated variance of the posterior distribution (Turner and Sahani, 2011) and robustness issues (Giordano et al., 2018)—it leads to scalable formulations, with already competitive performance, as for instance illustrated by the recent uncertainty quantification benchmark of Ovadia et al. (2019).

Because of its simplicity and scalability, the fully-factorized Gaussian variational distribution has been widely used for Bayesian neural networks (Graves, 2011; Ranganath et al., 2014; Blundell et al., 2015; Hernández-Lobato and Adams, 2015; Zhang et al., 2017; Khan et al., 2018).

Variational family	Parameters (total)
Multivariate Normal	$mn + \frac{mn\,(mn+1)}{2}$
Diagonal Normal	$mn + mn$
$\mathcal{MN}$(rank 1)	$mn + m + n$
k-tied Normal	$mn + k(m + n)$

Table 1: Number of variational parameters for a variational family for a matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$. $\mathcal{MN}$(rank 1) is from Louizos and Welling (2016).

Our approach can be seen as an attempt to further reduce the number of parameters of the (already) diagonal covariance matrix. Closest to our approach is the work of Louizos and Welling (2016). Their matrix variate Gaussian distribution instantiated with the Kronecker product of the diagonal row- and column-covariance matrices leads to a rank-1 tying of the posterior variances. In contrast, we explore tying strategies beyond the rank-1 case, which we show to lead to better performance (both in terms of ELBO and predictive metrics). Importantly, we further prove that tying strategies with a rank greater than one cannot be represented in a matrix variate Gaussian distribution, thus clearly departing from (Louizos and Welling, 2016) (see Appendix B for details).

Our approach can be also interpreted as a particular case of *hierarchical* variational inference (Ranganath et al., 2016) where the prior on the variational parameters corresponds to a Dirac distribution, non-zero only when a pre-specified low-rank tying relationship holds.

We close this related work section by mentioning the existence of other strategies to produce more flexible approximate posteriors, e.g., normalizing flows (Rezende and Mohamed, 2015) and extensions thereof (Louizos and Welling, 2017).

Appendix B. Proof of the Matrix Variate Normal Parameterization

In this section of the appendix, we formally explain the connections between the k-tied Normal distribution and the matrix variate Gaussian distribution (Gupta and Nagar, 2018), referred to as $\mathcal{MN}$.

Consider positive definite matrices $\mathbf{Q} \in \mathbb{R}^{r\times r}$ and $\mathbf{P} \in \mathbb{R}^{c\times c}$ and some arbitrary matrix $\mathbf{M} \in \mathbb{R}^{r\times c}$. We have by definition that $\mathbf{W} \in \mathbb{R}^{r\times c} \sim \mathcal{MN}(\mathbf{M}, \mathbf{Q}, \mathbf{P})$ if and only if $\text{vec}(\mathbf{W}) \sim \mathcal{N}(\text{vec}(\mathbf{M}), \mathbf{P} \otimes \mathbf{Q})$, where $\text{vec}(\cdot)$ stacks the columns of a matrix and $\otimes$ is the Kronecker product

The $\mathcal{MN}$ has already been used for variational inference by Louizos and Welling (2016) and Sun et al. (2017). In particular, Louizos and Welling (2016) consider the case where both $\mathbf{P}$ and $\mathbf{Q}$ are restricted to be diagonal matrices. In that case, the resulting distribution corresponds to our k-tied Normal distribution with $k = 1$ since

$$\mathbf{P} \otimes \mathbf{Q} = \text{diag}(\mathbf{p}) \otimes \text{diag}(\mathbf{q}) = \text{diag}(\text{vec}(\mathbf{q}\mathbf{p}^\top)).$$

Importantly, we prove below that, in the case where $k \geq 2$, the k-tied Normal distribution cannot be represented as a matrix variate Gaussian distribution.

Lemma. [Rank-2 matrix and Kronecker product] Let $\mathbf{B}$ be a rank-2 matrix in $\mathbb{R}_+^{r\times c}$. There do not exist matrices $\mathbf{Q} \in \mathbb{R}^{r\times r}$ and $\mathbf{P} \in \mathbb{R}^{c\times c}$ such that

$$\text{diag}(\text{vec}(\mathbf{B})) = \mathbf{P} \otimes \mathbf{Q}.$$

Proof Let us introduce the shorthand $\mathbf{D} = \text{diag}(\text{vec}(\mathbf{B}))$. By construction, $\mathbf{D}$ is diagonal and has its diagonal terms strictly positive (it is assumed that $\mathbf{B} \in \mathbb{R}_+^{r\times c}$, i.e., $b_{ij} > 0$ for all i, j).

We proceed by contradiction. Assume there exist $\mathbf{Q} \in \mathbb{R}^{r\times r}$ and $\mathbf{P} \in \mathbb{R}^{c\times c}$ such that $\mathbf{D} = \mathbf{P} \otimes \mathbf{Q}$.

This implies that all diagonal blocks of $\mathbf{P} \otimes \mathbf{Q}$ are themselves diagonal with strictly positive diagonal terms. Thus, $p_{jj}\mathbf{Q}$ is diagonal for all $j \in \{1, \ldots, c\}$, which implies in turn that $\mathbf{Q}$ is diagonal, with non-zero diagonal terms and $p_{jj} \neq 0$. Moreover, since the off-diagonal blocks $p_{ij}\mathbf{Q}$ for $i \neq j$ must be zero and $\mathbf{Q} \neq \mathbf{0}$, we have $p_{ij} = 0$ and $\mathbf{P}$ is also diagonal.

To summarize, if there exist $\mathbf{Q} \in \mathbb{R}^{r\times r}$ and $\mathbf{P} \in \mathbb{R}^{c\times c}$ such that $\mathbf{D} = \mathbf{P} \otimes \mathbf{Q}$, then it holds that $\mathbf{D} = \text{diag}(\mathbf{p}) \otimes \text{diag}(\mathbf{q})$ with $\mathbf{p} \in \mathbb{R}^c$ and $\mathbf{q} \in \mathbb{R}^r$. This last equality can be rewritten as $b_{ij} = p_j q_i$ for all $i \in \{1, \ldots, r\}$ and $j \in \{1, \ldots, c\}$, or equivalently

$$\mathbf{B} = \mathbf{q}\mathbf{p}^\top.$$

This leads to a contradiction since $\mathbf{q}\mathbf{p}^\top$ has rank one while $\mathbf{B}$ is assumed to have rank two. ■

Appendix C. Low rank-structure in the GMFVI posterior standard deviations

We provide here more results from the post training analysis of the converged posterior standard deviations trained with the standard parameterization of the GMFVI. In particular, while in the main paper we focused on the CNN model, here we provide also the results for the MLP and LSTM model.

Our main experimental observation is that the standard GMFVI learns posterior standard deviation matrices that have a low-rank structure across different model types. To show this, we

investigate the results of the SVD decomposition of posterior standard deviation matrices for three types of models trained until ELBO convergence using GMFVI. Figure 5 shows per rank percentage of explained variance with respect to the rank k of the low-rank SVD approximation. The percent of explained variance for the rank k approximation is calculated as $100 \cdot \gamma_k^2 / \sum_{i'} \gamma_{i'}^2$, where $\gamma_{i'}$ are singular values. We observe that most of the variance in the posterior standard deviation parameters is captured in the rank-1 approximation. However, a more fine-grained analysis shows that a rank-2 approximation can encompass nearly all of the remaining variance. Finally, we note that we do not observe the same behaviour for the posterior mean parameters as we do for the posterior standard deviation parameters. Figure 4 further supports this claim visually by comparing the heat maps of the full-rank posterior standard deviations matrix with its rank-1 and rank-2 approximations. In particular, we observe that the rank-2 approximation results in the heat-map looking visually very similar to the full-rank matrix.

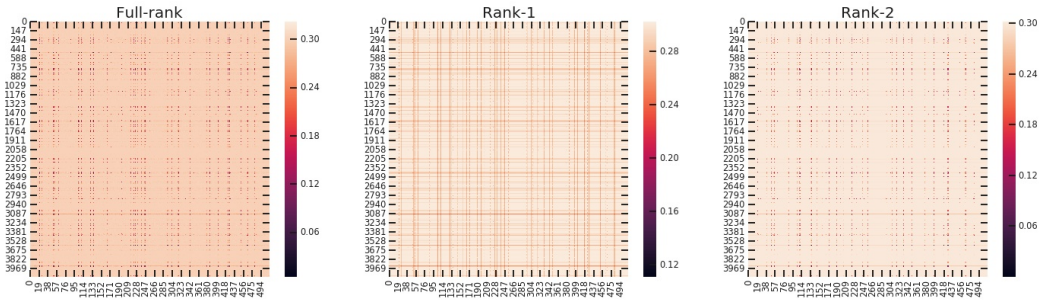


Figure 4: Heat map of the posterior standard deviation matrix for the weights in the first dense layer of a LeNet CNN trained using GMFVI on the CIFAR-100 dataset. Left: no approximation. Middle: rank-1 approximation. Right: rank-2 approximation. The rank-2 approximation looks visually similar to the full-rank matrix/no approximation, confirming our quantitative results from Figure 5.

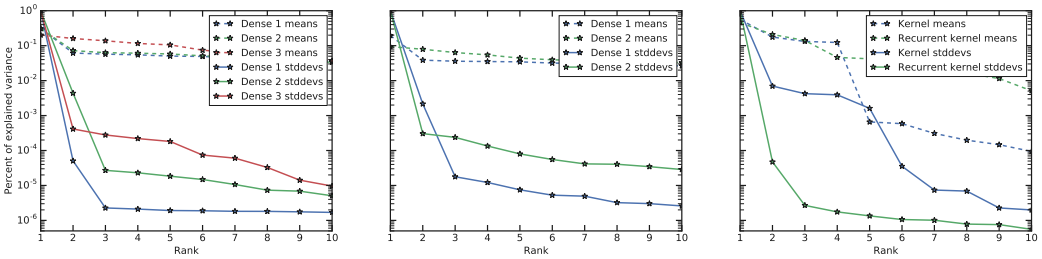


Figure 5: Explained variance per singular value from SVD of matrices of converged posterior means and posterior standard deviations for different layers of three types of models trained using standard GMFVI: MLP (left), CNN (center), LSTM (right). Posterior standard deviations clearly display strong low-rank structure, with most of the variance contained in the top few singular values, while this is not the case for posterior means.

Motivated by the above observation, we show that it is possible to replace the full-rank posterior standard deviation matrix with its low-rank approximation without a decrease in performance. Table 2 shows the comparison of performance of models with different ranks of approximation to their posterior standard deviation matrix. The results show that the post training approximation with ranks higher than 1 achieves predictive performance very close to that of the full-rank matrix. This observation itself could be used as a form of a post training network compression. Moreover, it gives rise to further interesting exploration directions such as formulating posteriors that exploit

such a low rank structure. In this paper by explore this particular direction in the form of the k-tied Normal posterior.

	MLP			CNN			LSTM		
Rank	-ELBO ↓	NLL ↓	Accuracy ↑	-ELBO ↓	NLL ↓	Accuracy ↑	-ELBO ↓	NLL ↓	Accuracy ↑
Full	$0.431_{\pm 0.0057}$	$0.100_{\pm 0.0034}$	$97.6_{\pm 0.15}$	$3.83_{\pm 0.020}$	$2.23_{\pm 0.017}$	$42.1_{\pm 0.49}$	$0.536_{\pm 0.0058}$	$0.493_{\pm 0.0057}$	$80.1_{\pm 0.25}$
1	$3.41_{\pm 0.019}$	$0.677_{\pm 0.0040}$	$93.6_{\pm 0.25}$	$4.33_{\pm 0.021}$	$2.30_{\pm 0.016}$	$41.7_{\pm 0.49}$	$0.687_{\pm 0.0058}$	$0.491_{\pm 0.0056}$	$80.0_{\pm 0.25}$
2	$0.456_{\pm 0.0059}$	$0.107_{\pm 0.0033}$	$97.6_{\pm 0.15}$	$3.88_{\pm 0.020}$	$2.24_{\pm 0.017}$	$42.2_{\pm 0.49}$	$0.621_{\pm 0.0058}$	$0.494_{\pm 0.0057}$	$80.1_{\pm 0.25}$
3	$0.450_{\pm 0.0059}$	$0.106_{\pm 0.0033}$	$97.6_{\pm 0.15}$	$3.86_{\pm 0.020}$	$2.24_{\pm 0.017}$	$42.1_{\pm 0.49}$	$0.595_{\pm 0.0058}$	$0.493_{\pm 0.0056}$	$80.1_{\pm 0.25}$

Table 2: Impact of post training low-rank approximation of the GMFVI-trained posterior standard deviation matrices on ELBO and predictive performance, for three types of models. We report mean and SEM of each metric across 100 weights samples.

Appendix D. Impact of the k-tied Normal on the GMFVI convergence speed

Figure 6 shows convergence plots of negative ELBO on respective validation data sets for different model types trained with GMFVI using the standard parametrization (full-rank) and the k-tied Normal posterior with different levels of tying k. We observe that the impact of the k-tied Normal posterior on the convergence depends on the model type. For the MLP model the impact is strong and consistent with the k-tied Normal posterior increasing convergence speed compared to the standard GMFVI parametrization. For the LSTM model we also observe a similar speed-up. However, for the CNN model the impact of the k-Normal posterior on the ELBO convergence is much smaller. We hypothesize that this is due to the fact that we use the k-tied Normal posterior for all the layers trained using GMFVI in the MLP and the LSTM models, while in the CNN model we use the k-tied Normal posterior only for some of the GMFVI trained layers. More precisely, in the CNN model we use the k-tied Normal posterior only for the two dense layers, while the two convolutional layers are trained using the standard parametrization of the GMFVI.

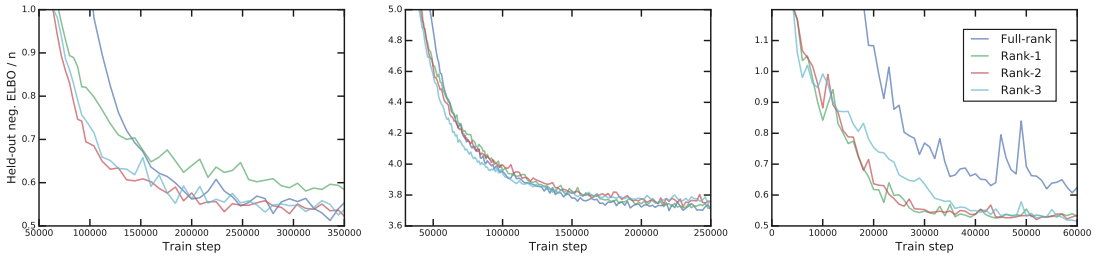


Figure 6: Impact of the k-tied Normal posterior with different ranks k on the convergence of negative ELBO (lower is better) reported on validation datasets of the MLP (left), CNN (center), and LSTM (right) models. Full-rank is the standard parametrization of the GMFVI without any tying.

Appendix E. Experimental details

Model architectures We analyze three types of GMFVI Bayesian neural network models:

- Multilayer Perceptron (MLP): a network of three dense layers and ReLu activations that we train on the MNIST dataset (LeCun and Cortes, 2010). We use the last 10,000 examples of the training set as a validation set. The three layers have sizes of 400, 400 and 10 hidden units.

- Convolutional Neural Network (CNN): a LeNet architecture (LeCun et al., 1998) with two convolutional layers and two dense layers that we train on the CIFAR-100 dataset (Krizhevsky et al., 2009). We use the last 10,000 examples of the training set as a validation set. The two convolutional layers have filters of sizes 32 and 64. The two dense layers have sizes of 512 and 100 hidden units.

- Long Short-Term Memory (LSTM): a model that consists of an embedding and an LSTM cell (Hochreiter and Schmidhuber, 1997), followed by a single unit dense layer. We train it on an IMBD dataset (Maas et al., 2011), in which we use the last 5,000 examples of the training set as a validation set. The LSTM cell consists of two dense weight matrices, namely kernel and recurrent kernel. The embedding and the LSTM cell are each of size 128. More concretely, we use the model architecture available in the Keras (Chollet et al., 2015) examples[4], but without dropout.

GMFVI training In the MLP and the CNN models we approximate the posterior using GMFVI for all the weights (both kernel and bias weights). In the LSTM model we approximate the posterior using GMFVI only for the kernel and recurrent kernel weights, while the posterior for the bias weights is approximated using a MAP solution.

In each of the three models we use a mean-field Normal posterior with the standard reparametrization trick (Kingma and Welling, 2013) and a Normal prior $\mathcal{N}(0, \sigma_p)$ with a single scalar standard deviation hyper-parameter σ_p for all the layers. We initialize the variational posterior means using the standard He initialization (He et al., 2015) and the posterior standard deviations using samples from $\mathcal{N}(0.01, 0.001)$. We select the σ_p for each of the models separately from a set of $\{0.2, 0.3\}$ based on the performance on the validation data set.

For optimization we use an Adam optimizer (Kingma and Ba (2014)). We pick the optimal learning rate for each model from the set of $\{0.0001, 0.0003, 0.001, 0.003\}$ based on the performance on the validation data set. We chose the batch size also based on the performance on the validation data set. For the MLP and the CNN models we use the batch size of 1024 and for the LSTM model a batch size of 128.

Low-rank structure analysis To investigate the low-rank structure in the converged posterior standard deviation matrices, we generate low-rank approximations to these matrices. It is possible that such low-rank approximations contain negative values. In such cases, we threshold the minimum values of the resulting approximations at a very low positive constant to meet the constraint on the positive values of the standard deviations.

k-tied Normal posterior training When training the GMFVI models with the k-tied Normal variational posterior, we use the k-tied Normal variational posterior for all the dense layers of the three analyzed models. More concretely, we use the k-tied Normal variational posterior for all the three layers of the MLP model, for the two dense layers of the CNN model and for the LSTM cell's kernel and recurrent kernel.

We initialize the parameters u_{ik} and v_{jk} of the k-tied Normal distribution so that after the outer-product operation the respective standard deviations σ_{ij} have the same mean values as we obtain when using the standard GMFVI posterior parametrization. In other words, we initialize the parameters u_{ik} and v_{jk} so that after the outer-product operation the respective σ_{ij} standard deviations have

4. https://github.com/keras-team/keras/blob/master/examples/imdb_lstm.py

means at 0.01 before transforming to log-domain. This means that in the log domain the parameters u_{ik} and v_{jk} are initialized as $0.5(\log(0.01) - \log(k))$. We also add white noise $\mathcal{N}(0, 0.1)$ to the values of u_{ik} and v_{jk} in the log domain to break symmetry.

We recommend using KL annealing for training the models with the k-tied Normal posterior. With KL annealing, we linearly scale-up the contribution of the KL term from a fraction of its full value to its full contribution over the course of training. We select the best linear coefficient for the KL annealing from $\{5 \times 10^{-5}, 5 \times 10^{-6}\}$ per batch and increase the KL contribution every 100 batches. For instance, we use KL annealing to obtain the results for the test performance in Figure 2. However, we do not use KL annealing for the runs for which we report the SNR and negative ELBO convergence results in the same Figure 2. In these two cases KL annealing would occlude the values, which show the clear impact of the k-tied Normal posterior.