Efficient Approximate Inference with Walsh-Hadamard Variational Inference

Simone Rossi[*]
Department of Data Science
EURECOM

Sébastien Marmin[*]
Department of Data Science
EURECOM
{name.surname}@eurecom.fr

Maurizio Filippone
Department of Data Science
EURECOM

Abstract

Variational inference offers scalable and flexible tools to tackle intractable Bayesian inference of modern statistical models like Bayesian neural networks and Gaussian processes. For largely over-parameterized models, however, the over-regularization property of the variational objective makes the application of variational inference challenging. Inspired by the literature on kernel methods, and in particular on structured approximations of distributions of random matrices, this paper proposes Walsh-Hadamard Variational Inference, which uses Walsh-Hadamard-based factorization strategies to reduce model parameterization, accelerate computations, and increase the expressiveness of the approximate posterior beyond fully factorized ones.

1 Introduction and Motivation

Scalable Bayesian inference for non-trivial statistical models is achieved with Variational Inference (VI, Jordan et al. [13]). Variational Inference has continuously gained popularity as a flexible approximate inference scheme for a variety of models for which exact Bayesian inference is intractable. Bayesian neural networks [20, 23] and in particular Deep Gaussian Processes with random features expansions [5, 6] represent good examples of models for which inference is intractable, and for which VI– and approximate inference in general – is challenging due to the nontrivial form of the posterior distribution and the large dimensionality of the parameter space [10, 9]. Recent advances in VI allow to effectively deal with these issues in various ways. A flexible class of posterior approximations can be constructed using, e.g., normalizing flows [25], whereas large parameter space have pushed the research in the direction of Bayesian compression [19, 22].

Let's consider a classic supervised learning task with N input vectors and corresponding labels collected in $\boldsymbol{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and $\boldsymbol{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$, respectively; furthermore, let's consider DNNs with weight matrices $\mathbf{W} = \left\{\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(L)}\right\}$, likelihood $p(\boldsymbol{Y}|\boldsymbol{X}, \mathbf{W})$, and prior $p(\mathbf{W})$. In the variational setting, a lower bound to the log-marginal likelihood can be computed as follows:

$$\log\left[p(\boldsymbol{Y}|\boldsymbol{X})\right] \geq \mathbb{E}_{q(\mathbf{W})}[\log p(\boldsymbol{Y}|\boldsymbol{X}, \mathbf{W})] - \mathrm{KL}\{q(\mathbf{W})\|p(\mathbf{W})\}\,, \tag{1}$$

where $q(\mathbf{W})$ is a parameterized approximation of the true posterior $p(\mathbf{W}|\boldsymbol{X}, \boldsymbol{Y})$. This bound has two undesirable charateristics: the first term, which acts as a model fitting term, depends on the choice of the form of the variational distribution. Simple distributions might not have enough expressiveness to efficiently characterize the learning task. On the other hand, the latter term – which acts as a regularizer – penalizes solutions where the posterior is far away from the prior. This term is the dominant one in the objective in case of over-parameterized models, and as a result, the optimization focuses on keeping the approximate posterior close to the prior, disregarding the data fit term [2, 28, 27].

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

[*] Equal contribution

In this paper, we will try to solve both problems at once. Our proposal is to reparameterize the variational posterior over model parameters by means of a structured decomposition based on random matrix theory [30, 17, 32]. Without loss of generality, consider Bayesian DNNs with weight matrices $\boldsymbol{W}^{(l)}$ of size $D \times D$. Compared with mean field VI, our proposal has a number of attractive properties. The number of parameters is reduced from $\mathcal{O}(D^2)$ to $\mathcal{O}(D)$, reducing the over-regularization effect of the KL term in the variational objective. We derive expressions for the reparameterization and the local reparameterization tricks, showing that, the computational complexity is reduced from $\mathcal{O}(D^2)$ to $\mathcal{O}(D \log D)$. Finally, unlike mean field VI, we induce a (non-factorized) matrix-variate distribution to approximate the posterior over the weights, increasing flexibility by modeling correlations between the weights at a log-linear cost in D instead of linear. The key operation within our proposal is the Walsh-Hadamard transform, and this is why we name our proposal Walsh-Hadamard Variational Inference (WHVI).

Related Work. WHVI is inspired by a line of works that developed from random feature expansions for kernel machines [24], which we briefly review here. A positive-definite kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ induces a mapping $\boldsymbol{\phi}(\mathbf{x})$, which can be infinite dimensional depending on the choice of $\kappa(\cdot, \cdot)$. Among the large literature of scalable kernel machines, random feature expansion techniques aim at constructing a finite approximation to $\boldsymbol{\phi}(\cdot)$. For many kernel functions [24, 4], this approximation is built by applying a nonlinear transformation to a random projection $\boldsymbol{X}\boldsymbol{\Omega}$, where $\boldsymbol{\Omega}$ has entries $\mathcal{N}(\omega_{ij}|0, 1)$. If the matrix of training points $\boldsymbol{X}$ is $N \times D$ and we are aiming to construct D random features, that is $\boldsymbol{\Omega}$ is $D \times D$, this requires N times $\mathcal{O}(D^2)$ time, which can be prohibitive when D is large. FASTFOOD [17] tackles the issue of large dimensional problems by replacing the matrix $\boldsymbol{\Omega}$ with a random matrix for which the space complexity is reduced from $\mathcal{O}(D^2)$ to $\mathcal{O}(D)$ and time complexity of performing products with input vectors is reduced from $\mathcal{O}(D^2)$ to $\mathcal{O}(D \log D)$. In FASTFOOD, the matrix $\boldsymbol{\Omega}$ is replaced by $\boldsymbol{\Omega} \approx \boldsymbol{SHG\Pi HB}$, where $\boldsymbol{\Pi}$ is a permutation matrix, $\boldsymbol{H}$ is the Walsh-Hadamard matrix, whereas $\boldsymbol{G}$ and $\boldsymbol{B}$ are diagonal random matrices with standard Normal distributions and Rademacher ($\{\pm 1\}$), respectively. $\boldsymbol{S}$ is also diagonal with i.i.d. entries, and it is chosen such that the elements of $\boldsymbol{\Omega}$ obtained by this series of operations are approximately independent and follow a standard Normal (see Tropp [30] for more details). The Walsh-Hadamard matrix is defined recursively starting from $H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and then $H_{2D} = \begin{bmatrix} H_D & H_D \\ H_D & -H_D \end{bmatrix}$, possibly scaled by $D^{-1/2}$ to make it orthonormal. The product of $\boldsymbol{H}\mathbf{x}$ can be computed in $\mathcal{O}(D \log D)$ time and $\mathcal{O}(1)$ space using the in-place version of the Fast Walsh-Hadamard Transform [8]. FASTFOOD inspired a series of other works on kernel approximations [31, 1], whereby Gaussian random matrices are approximated by a series of products between diagonal Rademacher and Walsh-Hadamard matrices, for example $\boldsymbol{\Omega} \approx \boldsymbol{HS}_1\boldsymbol{HS}_2\boldsymbol{HS}_3$.

2 Walsh-Hadamard Variational Inference

WHVI [26] proposes a way to generate non-factorized distributions with reduced requirements in memory and computational complexity. By considering a prior for the elements of the diagonal matrix $\boldsymbol{G} = \text{diag}(\mathbf{g})$ and a variational posterior $q(\mathbf{g}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, we can obtain a class of approximate posterior with some desirable properties. Let $\boldsymbol{W} \in \mathbb{R}^{D \times D}$ be the weight matrix and consider

$$\tilde{\boldsymbol{W}} \sim q(\boldsymbol{W}) \quad \text{s.t.} \quad \tilde{\boldsymbol{W}} = \boldsymbol{S}_1\boldsymbol{H}\text{diag}(\tilde{\mathbf{g}})\boldsymbol{HS}_2 \quad \text{with} \quad \tilde{\mathbf{g}} \sim q(\mathbf{g}). \tag{2}$$

The choice of a Gaussian $q(\mathbf{g})$ and the linearity of the operations, induce a parameterization of a matrix-variate Gaussian distribution for $\boldsymbol{W}$, which is controlled by $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. These diagonal matrices can be optimized during the training. We refer the Reader to check [26] for an extended analysis of this factorization. Sampling from such a distribution is achieved with the so-called *reparameterization trick* [14]. As we assume a Gaussian posterior for $\mathbf{g}$, the expression $\mathbf{g} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}$ separates out the stochastic component ($\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$) from the deterministic ones ($\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}^{1/2}$).

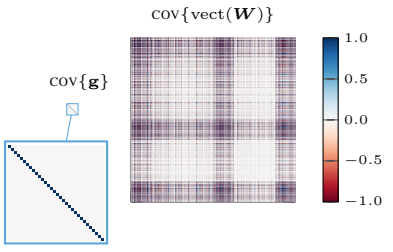

Figure 1: Normalized covariance of $\mathbf{g}$ and $\text{vect}(\boldsymbol{W})$

To reduce the variance of stochastic gradients in the optimization and improving convergence, we also report the formulation of the *local reparameterization trick* [15], which, given some input

vectors $\mathbf{h}$, considers the distribution of the product $\boldsymbol{W}\mathbf{h}$. The product $\boldsymbol{W}\mathbf{h}$ follows the distribution $\mathcal{N}(\mathbf{m}, \boldsymbol{A}\boldsymbol{A}^\top)$ [11], with

$$\mathbf{m} = \boldsymbol{S}_1\boldsymbol{H}\text{diag}(\boldsymbol{\mu})\boldsymbol{H}\boldsymbol{S}_2\mathbf{h}, \quad \text{and} \quad \boldsymbol{A} = \boldsymbol{S}_1\boldsymbol{H}\text{diag}(\boldsymbol{H}\boldsymbol{S}_2\mathbf{h})\boldsymbol{\Sigma}^{1/2}. \tag{3}$$

A sample from this distribution can be efficiently computed thanks to the Walsh-Hadamard transform as: $\overline{\boldsymbol{W}}(\boldsymbol{\mu})\mathbf{h} + \overline{\boldsymbol{W}}(\boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon})\mathbf{h}$, with $\overline{\boldsymbol{W}}$ a linear matrix-valued function $\overline{\boldsymbol{W}}(\mathbf{u}) = \boldsymbol{S}_1\boldsymbol{H}\text{diag}(\mathbf{u})\boldsymbol{H}\boldsymbol{S}_2$.

WHVI can be extended to handle parameters of any shape $\boldsymbol{W} \in \mathbb{R}^{D_1 \times D_2}$.

When one of the dimensions is one so that the parameter matrix is a vector ($\boldsymbol{W} = \mathbf{w} \in \mathbb{R}^D$), WHVI can be extended to handle these cases efficiently by reshaping the parameter vector into a matrix of size 2^d with suitable d, again by padding if necessary. Thanks to the reshaping, WHVI uses $\sqrt{D}$ parameters to model a posterior over D, and allows for computations in $\mathcal{O}(\sqrt{D}\log D)$ rather than D. This is possible by reshaping the vector that multiplies the weights in a similar way. We will explore this idea to infer parameters of Gaussian processes linearized using random features.

3 Experiments

3.1 Bayesian Neural Networks

We conduct a series of comparisons with state-of-the-art VI schemes for Bayesian DNNs: MCD and NOISY-KFAC (also referred to as NNG; [33]). MCD draws on a formal connection between dropout and VI with Bernoulli-like posteriors, while the more recent NOISY-KFAC yields a matrix-variate Gaussian distribution using noisy natural gradients. In WHVI, the last layer assumes a fully factorized Gaussian posterior.

Data is randomly divided into 90%/10% splits for training and testing. We standardize the input features $\mathbf{x}$ while keeping the targets $\mathbf{y}$ unnormalized. Differently from the experimental setup in [18, 33, 12], the network has two hidden layers and 128 features with RELU activations for all the datasets and its output is parameterized as $\mathbf{y} = \mathbf{f}(\mathbf{x}) \odot \boldsymbol{\sigma}_y + \boldsymbol{\mu}_y$.

We report the test RMSE and the average predictive test negative log-likelihood (MNLL). A selection of results is showed in Table 1. On the majority of the datasets, WHVI outperforms MCD and NOISY-KFAC. Empirically, these results demonstrate the value of WHVI, which offers a competitive parameterization of a matrix-variate Gaussian posterior while requiring log-linear time in D.

3.2 Gaussian Processes with Random Feature Expansion

We test WHVI for scalable GP inference, by focusing on GPs with random feature expansions [16, 5]. We compare WHVI with two alternatives; one is VI of the Fourier features GP expansion that uses less random features to match the number of parameters used in WHVI, and another is the sparse Gaussian process implementation of GPFLOW [21] with a number of inducing points (rounded up) to match the number of parameters used in WHVI.

We report the results on five datasets ($10000 \leq N \leq 200000$, $5 \leq D \leq 8$), generated from space-filling evaluations of well known functions in analysis of computer experiments (see e.g. [29]). Dataset splitting in training and testing points is random uniform with ratio 80%/20%.

Table 1: Test RMSE and test MNLL for regression datasets

	TEST ERROR			TEST MNLL		
MODEL DATASET	MCD	NNG	WHVI	MCD	NNG	WHVI
BOSTON	3.91 ± 0.86	3.56 ± 0.43	$\mathbf{3.14 \pm 0.71}$	6.90 ± 2.93	$\mathbf{2.72 \pm 0.09}$	4.33 ± 1.80
CONCRETE	5.12 ± 0.79	8.21 ± 0.55	$\mathbf{4.70 \pm 0.72}$	3.20 ± 0.36	3.56 ± 0.08	$\mathbf{3.17 \pm 0.37}$
ENERGY	2.07 ± 0.11	1.96 ± 0.28	$\mathbf{0.58 \pm 0.07}$	4.15 ± 0.15	2.11 ± 0.12	$\mathbf{2.00 \pm 0.60}$
KIN8NM	0.09 ± 0.00	$\mathbf{0.07 \pm 0.00}$	0.08 ± 0.00	-0.87 ± 0.02	$\mathbf{-1.19 \pm 0.04}$	$\mathbf{-1.19 \pm 0.04}$
NAVAL	0.30 ± 0.30	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.00	-1.00 ± 2.27	$\mathbf{-6.52 \pm 0.09}$	-6.25 ± 0.01
POWERPLANT	31.65 ± 0.07	4.23 ± 0.09	$\mathbf{4.00 \pm 0.12}$	49.78 ± 0.17	2.86 ± 0.02	$\mathbf{2.71 \pm 0.03}$
PROTEIN	$\mathbf{4.23 \pm 0.10}$	4.57 ± 0.47	4.36 ± 0.11	$\mathbf{2.76 \pm 0.02}$	2.95 ± 0.12	2.79 ± 0.01
YACHT	1.90 ± 0.54	5.16 ± 1.48	$\mathbf{0.69 \pm 0.16}$	2.95 ± 1.27	3.06 ± 0.27	$\mathbf{1.80 \pm 1.01}$

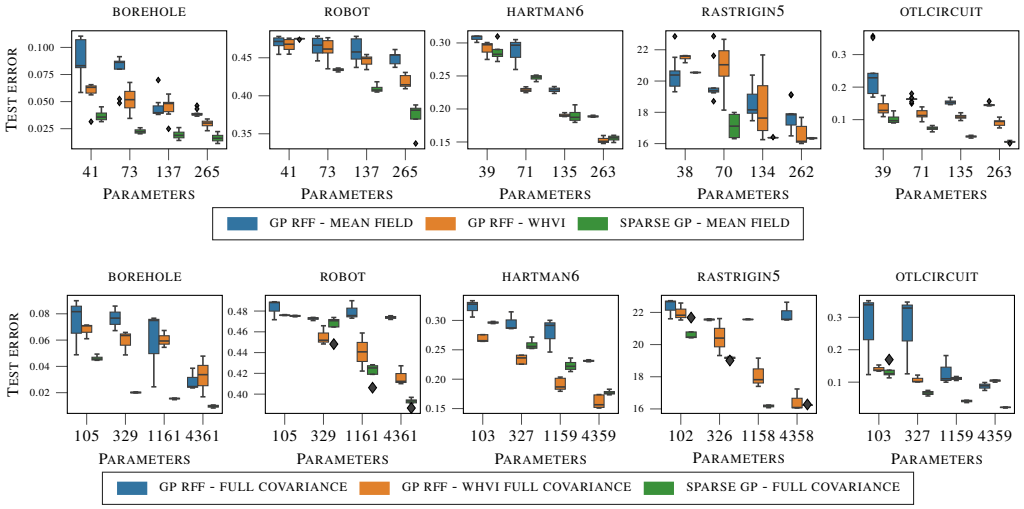

Figure 2: Comparison of test errors with respect to the number model parameters.

The results are shown in Figure 2 for both with diagonal covariance and with full covariance. In both mean field and full covariance settings, this variant of WHVI using the reshaping of $\boldsymbol{W}$ into a column largely outperforms the direct VI of Fourier features. However, it appears that this improvement of the random feature inference for GPs is still not enough to reach the performance of VI using inducing points. Inducing point approximations are based on the Nystroöm approximation of kernel matrices, which are known to lead to lower approximation error on the elements on the kernel matrix compared to random features approximations. This is the reason we attribute to the lower performance of WHVI compared to inducing points approximations in this experiment.

4 Discussion and Conclusions

Inspired by the literature on scalable kernel methods, this paper proposed Walsh-Hadamard Variational Inference (WHVI). WHVI offers a novel parameterization of the variational posterior as it assumes a matrix-variate posterior distribution, which therefore captures covariances across weights. Crucially, unlike previous work on matrix-variate posteriors for VI, this is achieved with a low parameterization and fast computations, bypassing the over-regularization issues of VI for over-parameterized models.

The key operation that contributes to accelerate computations in WHVI is the Walsh-Hadamard transform. This has obvious connections with other matrix/vector operations, such as the Discrete Fourier Transform and other circulant matrixes [7, 3], so we are currently investigating whether it is possible to generalize WHVI to other kinds of transforms to increase flexibility. Finally, we are looking into employing WHVI for other models, such as deep generative models.

References

[1] M. Bojarski, A. Choromanska, K. Choromanski, F. Fagan, C. Gouy-Pailler, A. Morvan, N. Sakr, T. Sarlos, and J. Atif. Structured Adaptive and Random Spinners for Fast Machine Learning Computations. In A. Singh and J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1020–1029, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[2] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio. Generating Sentences from a Continuous Space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21. Association for Computational Linguistics, 2016.

[3] Y. Cheng, F. X. Yu, R. S. Feris, S. Kumar, A. Choudhary, and S. Chang. An Exploration of Parameter Redundancy in Deep Networks with Circulant Projections. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2857–2865, Dec 2015.

[4] Y. Cho and L. K. Saul. Kernel Methods for Deep Learning. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 342–350. Curran Associates, Inc., 2009.

[5] K. Cutajar, E. V. Bonilla, P. Michiardi, and M. Filippone. Random feature expansions for deep Gaussian processes. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 884–893, International Convention Centre, Sydney, Australia, Aug. 2017. PMLR.

[6] A. C. Damianou and N. D. Lawrence. Deep Gaussian Processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April 29 - May 1, 2013*, volume 31 of *JMLR Proceedings*, pages 207–215. JMLR.org, 2013.

[7] C. Ding, S. Liao, Y. Wang, Z. Li, N. Liu, Y. Zhuo, C. Wang, X. Qian, Y. Bai, G. Yuan, X. Ma, Y. Zhang, J. Tang, Q. Qiu, X. Lin, and B. Yuan. CirCNN: Accelerating and Compressing Deep Neural Networks Using Block-Circulant Weight Matrices. In *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 395–408, Oct 2017.

[8] Fino and Algazi. Unified Matrix Treatment of the Fast Walsh-Hadamard Transform. *IEEE Transactions on Computers*, C-25(11):1142–1146, Nov 1976. ISSN 0018-9340.

[9] Y. Gal and Z. Ghahramani. Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1050–1059. JMLR.org, 2016.

[10] A. Graves. Practical Variational Inference for Neural Networks. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2348–2356. Curran Associates, Inc., 2011.

[11] A. K. Gupta and D. K. Nagar. *Matrix variate distributions*. Chapman and Hall/CRC, 1999.

[12] J. M. Hernandez-Lobato and R. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1861–1869, Lille, France, 07–09 Jul 2015. PMLR.

[13] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37(2):183–233, Nov. 1999.

[14] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *Proceedings of the Second International Conference on Learning Representations (ICLR 2014)*, Apr. 2014.

[15] D. P. Kingma, T. Salimans, and M. Welling. Variational Dropout and the Local Reparameterization Trick. In *Advances in Neural Information Processing Systems 28*, pages 2575–2583. Curran Associates, Inc., 2015.

[16] M. Lázaro-Gredilla, J. Quinonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse Spectrum Gaussian Process Regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

[17] Q. Le, T. Sarlos, and A. Smola. Fastfood - Approximating Kernel Expansions in Loglinear Time. In *30th International Conference on Machine Learning (ICML)*, 2013.

[18] C. Louizos and M. Welling. Structured and Efficient Variational Deep Learning with Matrix Gaussian Posteriors. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1708–1716, New York, New York, USA, 20–22 Jun 2016. PMLR.

[19] C. Louizos, K. Ullrich, and M. Welling. Bayesian Compression for Deep Learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3288–3298. Curran Associates, Inc., 2017.

[20] D. J. C. Mackay. Bayesian methods for backpropagation networks. In E. Domany, J. L. van Hemmen, and K. Schulten, editors, *Models of Neural Networks III*, chapter 6, pages 211–254. Springer, 1994.

[21] A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, apr 2017.

[22] D. Molchanov, A. Ashukha, and D. Vetrov. Variational Dropout Sparsifies Deep Neural Networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2498–2507, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[23] R. M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996. ISBN 0387947248.

[24] A. Rahimi and B. Recht. Random Features for Large-Scale Kernel Machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc., 2008.

[25] D. Rezende and S. Mohamed. Variational Inference with Normalizing Flows. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR.

[26] S. Rossi, S. Marmin, and M. Filippone. Walsh-Hadamard Variational Inference for Bayesian Deep Learning. In *arXiv: 1905.11248*, 2019.

[27] S. Rossi, P. Michiardi, and M. Filippone. Good Initializations of Variational Bayes for Deep Models. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5487–5497, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[28] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder Variational Autoencoders. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3738–3746. Curran Associates, Inc., 2016.

[29] S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved May 22, 2019, from `http://www.sfu.ca/~ssurjano`.

[30] J. A. Tropp. Improved Analysis of the subsampled Randomized Hadamard Transform. *Advances in Adaptive Data Analysis*, 3(1-2):115–126, 2011.

[31] F. X. Yu, A. T. Suresh, K. M. Choromanski, D. N. Holtmann-Rice, and S. Kumar. Orthogonal Random Features. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1975–1983. Curran Associates, Inc., 2016.

[32] F. X. X. Yu, A. T. Suresh, K. M. Choromanski, D. N. Holtmann-Rice, and S. Kumar. Orthogonal Random Features. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1975–1983. Curran Associates, Inc., 2016.

[33] G. Zhang, S. Sun, D. Duvenaud, and R. Grosse. Noisy Natural Gradient as Variational Inference. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5852–5861, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.