Physics-Inspired Neural Networks for Enhancing Predictive Uncertainty of Bayesian Neural Networks

Rumen Dangovski [*1] **Ileana Rugina** [*1]

Abstract

Introducing weight uncertainty in deep neural networks creates models with more robust predictions, and thus opens a vast field of applications of *variational inference* (VI) to deep learning. Similarly to classical Bayesian inference, choosing the right parametrization for the priors of the weights and their variational posterior remains an open question. Here, we draw inspiration from parametrizations of physical systems to construct novel Bayesian neural networks. We call such networks *phase-coded*, since they resemble phase manipulations of light in nanophotonics. Through measuring the entropy of our models' predictive distributions and by using calibration plots we observe that our model's predictive uncertainty is a promising alternative to known parametrizations.

1. Introduction

Recent developments of neural network models have produced state of the art results across a multitude of input modalities. Convolutional *deep neural networks* (DNN) have revolutionized computer vision (Krizhevsky et al., 2012) and Transformer-based architectures are used across all types of natural language processing applications (Vaswani et al., 2017).

One significant limitation of deep learning models is that *we cannot easily extract uncertainty information* in a principled way, which is critical in a plethora of applications, such as active learning or safety-critical domains (Gal & Ghahramani, 2015). Hence, recent casts Bayesian deep learning as an optimization problem using VI (Graves, 2011) and proposes a solution by employing methods compatible with backpropagation, on which DNN training depends (Blundell et al., 2015).

Nevertheless, two major issues arise in VI for DNNs: (*a*) finding and training good variational approximations of weights posteriors is hard; (*b*) both training and inference in VI for DNNs requires estimation of intractable integrals, which places a burden on the computational resources.

Here, we address (*a*) by proposing physics-inspired phase-coded neural networks in Bayesian deep learning. Namely, we investigate parametrizations of neural networks via tunable Mach–Zehnder interferometers in the optical computing domain (Jing et al., 2017). Such systems can simulate arbitrary neural networks, and due to their analogue nature carry natural noise in their phase-shifters and signal encoders/ detectors, yielding nice models for both aleatoric and epistemic parametrizations (*a*). Furthermore, building optical hardware based on such parametrizations can speed up DNN inference by more than two orders of magnitude (Shen et al., 2017), making phase-coded neural networks resource-friendly (*b*). Although this is a very promising property of our parametrization, we leave it for a physics study and focus on (*a*) in this work.

We assess the benefits and drawbacks of our proposal using proof-of-concept simulations on classification MNIST and Fashion MNIST by comparing against standard Bayesian and non-Bayesian neural networks, and non-Bayesian neural networks with Dropout. In Section 2 we explain our model. Below, we organize the paper as follows: in Section 2 we explain the optimization and our parametrization. In Section 3 we present our experiments and provide a discussion on our results. In Section 4 we discuss relevant work. We conclude with final remarks in Section 5.

2. Optimization and Parametrization

Bayesian neural networks are neural networks with prior on their weights. Inference with such models is taken as an expectation over a learned posterior over the weights. Below we discuss an effective way to train such networks.

2.1. Variational Inference

We assume that the weights of the neural networks model $p_{\mathsf{D}|\mathsf{w}}(\cdot|\cdot)$ are a random variable w, which is modeled with the following prior $p_{\mathsf{w}}(\mathbf{w})$. The posterior $p_{\mathsf{w}|\mathsf{D}}(\mathbf{w}|\mathcal{D})$, given the data $\mathcal{D}$ as a random variable D, is hard to compute.

[*] Equal contribution [1] Massachusetts Institute of Technology. Correspondence to: Rumen Dangovski <rumenrd@mit.edu>, Ileana Rugina <irugina@mit.edu>.

Spring 2020 6.435 Project: Bayesian Modelling and Inference

Therefore, we resort to an approximation of the posterior $q(\mathbf{w}; \theta)$, parametrized by parameters $\mathbf{v}$, which we fit by minimizing the following *variational free energy* $F(\mathcal{D}, \mathbf{v})$, given as follows

$$-\mathbb{E}_{q(\cdot;\mathbf{v})}[\log p_{\mathsf{D}|\mathsf{w}}(\mathcal{D}|\cdot)] + \mathrm{KL}[q(\cdot;\mathbf{v})\|p_{\mathsf{w}}(\cdot)], \quad (1)$$

where the KL denotes the KL-divergence and the expectation is taken over the variational distribution.

Objective (1) corresponds to the *minimum description length* loss function, which is how variational inference was first introduced to neural networks (Hinton & Zemel, 1993). As discussed in (Graves, 2011), the first term decreases as they network's accuracy increases and can be interpreted as an error term, while the second one quantifies the model complexity.

We model the variational posterior $q(\mathbf{w}; \mathbf{v})$ using the following two assumptions:

- We assume the mean-field approximation holds;
- We model the target distribution using Gaussian approximations.

Farquhar et al. (2020) discuss the first approximation and argue that it is particularly well-suited for neural nets. They first show that for linear activations deep neural networks with diagonal posteriors are equivalent to shallower models which do exhibit cross-correlation terms and then perform an empirical study to demonstrate the same principle extends to neural architectures used in practice. These results further motivate our quest of accelerating inference in Bayesian deep networks.

Additionally, although Farquhar et al. (2020) mostly investigate the cost of employing mean-field techniques assuming Gaussian posteriors, they also show that the second approximation still enables us to model multi-modal posteriors, thus addressing a common critique of this approximation.

We model the prior distribution by simple mixtures of Gaussians. This choice resembles a spike-and-slab prior (Mitchell & Beauchamp, 1988) which is a good fit for optimization using stochastic gradient descent (Blundell et al., 2015).

2.2. Training Procedure

We perform variational inference and reduce the Bayesian learning problem to an optimization one with the objective given by (1). Note that this objective includes an expectation over $q(\mathbf{w}; \mathbf{v})$ and we are optimizing with respect to the $\mathbf{v}$'s which parameterize $q(\mathbf{w}; \mathbf{v})$. Blundell et al. (2015) describes how to take gradients of this objective using a variant of the reparametrization trick applied to all our network parameters, which we discuss below.

By introducing a random variable ϵ with probability density $q(\epsilon)$ to capture the problem's stochasticity we can write $\mathbf{w} = t(\mathbf{v}, \epsilon)$ where t is deterministic. Then, assuming $q(\epsilon)d\epsilon = q(\mathbf{w}|\mathbf{v})d\mathbf{w}$, we have that the expectation and gradient operators commute:

$$\frac{\partial}{\partial \mathbf{v}} \mathbb{E}_{q(\mathbf{w}|\mathbf{v})}\left[f(\mathbf{w}, \mathbf{v})\right] = \mathbb{E}_{q(\epsilon)}\left[\frac{\partial f(\mathbf{w}, \mathbf{v})}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial \mathbf{v}} + \frac{\partial f(\mathbf{w}, \mathbf{v})}{\partial \mathbf{v}}\right] \quad (2)$$

We approximate the expectation using Monte Carlo estimators with a small number of samples - we experimented with values in the range $[1, 5]$. Note that we can compute equation (2) easily due to automatic differentiation packages, such as PyTorch (Paszke et al., 2017) that implicitly calculate the correct total derivative.

We continue with the crux of this paper: a novel parametrization for neural networks.

2.3. Phase-coded parametrization.

Suppose we have a matrix $A \in \mathbb{R}^{N\times N}$. Let $N = 4$ for ease of illustration, the form for general N follows analogously. By the SVD decomposition, $A = U\Sigma V^T$, where U and V are orthogonal matrices, i.e. $U, V \in O(N)$, and Σ is a diagonal matrix. Reck et al. (1994); Clements et al. (2016) demonstrate that we can parametrize U as a block-diagonal composition of 2-by-2 rotations, each rotation defined for a parameter θ and having the form

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \quad (3)$$

The transformation (3) has a physcial meaning, since it could be similulated by a Mach-Zender Interferometer.

Figure 1 demonstrates how the parametrization works for the case of $N = 4$. We alternate two types of blocks of transformation: blocks (a) and (b). The first block is type (a) and consists of two rotations $R(\theta_1)$ and $R(\theta_2)$, so the matrix is given by

$$A(\theta_1, \theta_2) := \mathsf{blockdiagonal}(R(\theta_1), R(\theta_2)) \in \mathbb{R}^4.$$

Then follows a block of type (b), which consists only a single rotation in the middle, and identity operators on top and bottom, hence the matrix is given by

$$B(\theta_3) := \mathsf{blockdiagonal}(\mathrm{id}, R(\theta_3), \mathrm{id}) \in \mathbb{R}^4,$$

where id is the identity operator in $\mathbb{R}$, i.e. $\mathrm{id} \equiv 1$. By alternating two more times we get that the total transformation is as follows

$$U(\theta_1, \ldots, \theta_6) := B(\theta_6)A(\theta_4, \theta_5)B(\theta_3)A(\theta_1, \theta_2), \quad (4)$$

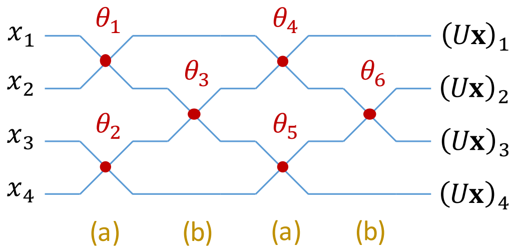

Figure 1. **Phase-coded parametrization.** The input vector is split into four channels (corresponding physically to nanophotonic waveguides where the information of the vector is encoded in coherent light generated by laser) and then undergoes series of transformation. At each crossing point we have a rotation (implemented physcially by a Mach-Zender inteferometer by introducing phase-shifts in the light and then coupling corresponding channels). At the end, the result is a general orthogonal transformation of the input, labelled by U.

and hence for a given input $\mathbf{x} \in \mathbb{R}^4$, the output from the layer that we introduce is $U(\theta_1, \ldots, \theta_6)\mathbf{x} \in \mathbb{R}^4$. Most notably, $U(\theta_1, \ldots, \theta_6)$ is an orthogonal matrix by virtue of its construction (4).[1] As a sanity check, an orthogonal matrix in $O(N)$ is parametrized by $N \cdot (N-1)/2$ parameters. In our case we expect $4 \cdot 3/2 = 6$ parameters, which are exactly given by $\theta_1, \ldots, \theta_6$.

Thus, the layer we just constructed we will label with $\mathsf{phasecoded}(N)$, which parametrizes a general orthogonal transformation in $O(N)$.

3. Experiments

3.1. Datasets

We experiment with classification on the MNIST dataset (Lecun et al., 1998), with a train/test split of 60,000/10,000 handwritten digits in gray scale and shape 28×28. An alternative, more challenging, dataset is FashionMNIST (Xiao et al., 2017), where the 10 classes of digits are replaced with clothing items, and the shape and gray scale are unchanged.

3.2. Models

The backbone of our models is a three layer fully connected neural network with hidden size 400. We compare four models:

- Standard neural network (NN);
- NN with Dropout (Srivastava et al., 2014) probability of 0.5 on the first hidden state and 0.2 on the second;
- Bayesian neural network (BNN) trained with the optimization strategy in Section 2;
- BNN with our parametrization trained with the optimization strategy in Section 2.

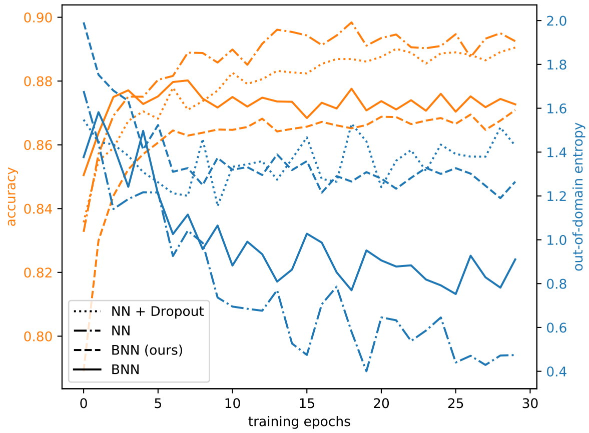

Figure 2. **Comparison between methods.** It seems that NN + Dropout gives the best entropy with a good enough accuracy and that our BNN can match accuracy of baseline BNN with higher out-of-domain entropy. Accuracy on FashionMNIST. Entropy on MNIST.

Our parametrization implies that the model's layers are

$$[\mathsf{linear}(784, 400), \mathsf{phasecoded}(400), \mathsf{linear}(400, 10)],$$

where 784 is the size of the input images 28×28, $N = 400$ is the hidden size and 10 is the number of possible classes. The training loss for the standard NNs is the negative log likelihood of the data, while the BNNs minimize objective (1). We train for 30 epochs with batch size 100, five samples for estimation of the variational free energy, and an qualy weighted prior mixture of two Gussians with the same mean and variances 1 and $\exp(-6)$.

3.3. Out-of-domain Entropy

We would like to measure the predictive uncertainty of our methods. For that purpose we train each model on FashionMNIST and then test their predictive uncertainty *out-of-domain* on the MNIST dataset. The motivation for this experiment is that the models would not see the MNIST features during training, hence they should be uncertain when making an out-of-domain prediction. Moreover, standard NNs are often critiqued for being "too confident" on classification, so we would like to see the Bayesian approach alleviates that, and if our parametrization further helps. For that purpose we set a held-out test sample of 5 MNIST datapoints and for each model we calculate the *average entropy of the predictive distributions for each datapoint*.

[1] Product of orthogonal matrices is orthogonal, since $O(N)$ is closed under multiplication of matrices.

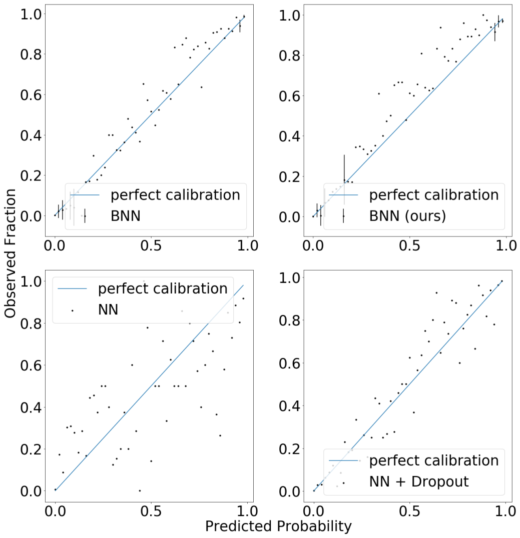

Figure 3. FashionMNIST Calibration plots.

The results are presented in Figure 2. We observe that our model converges to the highest out-of-domain entropy, performing slightly better than the standard BNN. Both standard neural network become overly confident on out-of-domain samples, albeit offering improvements in in-domain accuracy.

3.4. Calibration Diagnostic Plot

We continue our study on the predictive uncertainty of our model in comparison with standard baselines via an investigation of calibration plots (Niculescu-Mizil & Caruana, 2005). Similar to the entropy measured described above, this visual diagnostic tool evaluates the model's ability to not only make predictions, but more generally correctly define a probability distribution over the target classes.

In order to evaluate this, for a target class C we look at every test data point (x, y) and the predicted probability $P_M(y = C|x)$ that this data point belongs to class C according to model M. Across all (x, y) such that $p = P_M(y = C|x)$, we expect a fraction p of them to actually have $y = C$.

We empirically plot this fraction as a function of assigned probability p and expect to see a linear dependence - points should follow the first diagonal. The discrepency between this and observed results provide a qualitative measure of our model's quality. We present such plots in Figure 3 on the FashionMNIST test split.

3.5. Explaining the Visualization

We first notice regularizers play a central part in creating well-calibrated classifiers: the simple NN has a significantly more spread out calibration plot. BNNs and NNs regularized with Dropout perform comparable, although the former have the advantage of providing easy-to-interpret uncertainty estimators.

Bayesian neural networks still seem to underestimate uncertainty, as the assign very low uncertainties to predictions made far from either end of the $[0, 1]$ probability range. Our parameterization consistently underestimates assigned probabilities in this range, which we believe is consistent with Figure 2 in the sense that our BNN produces highest out-of-domain entropy. Our conjecture is that underestimation of in-domain samples correlates with high entropy of out-domain samples, which we leave for future work.

4. Related work

Variational inference was first proposed for neural networks by (Hinton & Zemel, 1993), motivated by the possibility of encoding regularizers through the choice of prior. Graves (2011) discusses how in this Bayesian NN scenario the loss function breaks up into two terms which quantify model performance and complexity. Hinton & Zemel (1993) provides a closed-form analytical solution for the variational posterior in the simple case of a linear single-layer feed forward network while Graves (2011) used Monte Carlo methods to train more complicated Bayesian models and then demonstrate their capacity of encoding parameter importance through successful pruning experiments in both computer vision and NLP.

More recently, Blundell et al. (2015) generalized the reparameterization trick introduced by (Kingma & Welling, 2014) to operate on all network weights, which are drawn from gaussian mixture models.

Relevant works about efficient Monte Carlo sampling with optical hardware (Roques-Carmes et al., 2020; Prabhu et al., 2019) implement dropout strategies naturally, and provide an interesting venue for future work.

5. Conclusion and Future Work

In this work we presented a novel parametrization for Bayesian neural networks. We investigated the predictive uncertainty of classifications by comparing with standard baselines on an out-of-domain entropy and calibration plots. We found that both diagnostic methods suggest that our parametrization is promising for modelling Bayesian neural networks.

Consequently, possibilities for future work span both finding better parametrizations for training and inference with VI and realizing such parametrization in AI accelerators.

References

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. 2015. cite arxiv:1505.05424Comment: In Proceedings of the 32nd International Conference on Machine Learning (ICML 2015).

Clements, W. R., Humphreys, P. C., Metcalf, B. J., Kolthammer, W. S., and Walmsley, I. A. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, Dec 2016. doi: 10.1364/OPTICA.3.001460.

Farquhar, S., Smith, L., and Gal, Y. Try Depth instead of weight correlations: Mean-field is a less restrictive assumption for variational inference in deep networks. *Bayesian Deep Learning Workshop at NeurIPS*, 2020.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *arXiv e-prints*, art. arXiv:1506.02142, June 2015.

Graves, A. Practical variational inference for neural networks. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 2348–2356. Curran Associates, Inc., 2011.

Hinton, G. E. and Zemel, R. S. Autoencoders, minimum description length and helmholtz free energy. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pp. 3–10, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

Jing, L., Shen, Y., Dubcek, T., Peurifoy, J., Skirlo, S., LeCun, Y., Tegmark, M., and Soljačić, M. Tunable efficient unitary neural networks (EUNN) and their application to RNNs. In *Proceedings of the 34th International Conference on Machine Learning*, ICML '17, pp. 1733–1741, Sydney, Australia, 2017.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., 2012.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Mitchell, T. J. and Beauchamp, J. J. Bayesian variable selection in linear regression. 1988.

Niculescu-Mizil, A. and Caruana, R. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pp. 625–632, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102430. URL `https://doi.org/10.1145/1102351.1102430`.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Prabhu, M., Roques-Carmes, C., Shen, Y., Harris, N., Jing, L., Carolan, J., Hamerly, R., Baehr-Jones, T., Hochberg, M., Ceperic, V., Joannopoulos, J., Englund, D., and Soljačić, M. A recurrent ising machine in a photonic integrated circuit, 09 2019.

Reck, M., Zeilinger, A., Bernstein, H. J., and Bertani, P. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.*, 73:58–61, Jul 1994. doi: 10.1103/PhysRevLett.73.58.

Roques-Carmes, C., Shen, Y., Zanoci, C., Prabhu, M., Atieh, F., Jing, L., Dubcek, T., Mao, C., Johnson, M., Ceperic, V., Joannopoulos, J., Englund, D., and Soljačić, M. Heuristic recurrent algorithms for photonic ising machines. *Nature Communications*, 11, 12 2020. doi: 10.1038/s41467-019-14096-z.

Shen, Y., Harris, N. C., Skirlo, S., Prabhu, M., Baehr-Jones, T., Hochberg, M., Sun, X., Zhao, S., Larochelle, H., Englund, D., and Soljačić, M. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11:441 EP –, 06 2017.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017.