Good Initializations of Variational Bayes for Deep Models

Simone Rossi [1] **Pietro Michiardi** [1] **Maurizio Filippone** [1]

Abstract

Stochastic variational inference is an established way to carry out approximate Bayesian inference for deep models flexibly and at scale. While there have been effective proposals for good initializations for loss minimization in deep learning, far less attention has been devoted to the issue of initialization of stochastic variational inference. We address this by proposing a novel layer-wise initialization strategy based on Bayesian linear models. The proposed method is extensively validated on regression and classification tasks, including Bayesian Deep Nets and Conv Nets, showing faster and better convergence compared to alternatives inspired by the literature on initializations for loss minimization.

1 Introduction

Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs) have become the preferred choice to tackle various learning tasks, due to their ability to model complex problems and the mature development of regularization techniques to control overfitting (LeCun et al., 2015; Srivastava et al., 2014). There has been a recent surge of interest in the issues associated with their overconfidence in predictions, and proposals to mitigate these (Guo et al., 2017; Kendall & Gal, 2017; Lakshminarayanan et al., 2017). Bayesian techniques offer a natural framework to deal with such issues, but they are characterized by computational intractability (Bishop, 2006; Ghahramani, 2015).

A popular way to recover tractability is to use variational inference (Jordan et al., 1999). In variational inference, an approximate posterior distribution is introduced and its parameters are adapted by optimizing a variational objective, which is a lower bound to the marginal likelihood. The variational objective can be written as the sum of an expectation of the log-likelihood under the approximate posterior and a regularization term which is the negative Kullback-Leibler (KL) divergence between the approximating distribution and the prior over the parameters. Stochastic Variational Inference (SVI) offers a practical way to carry out stochastic optimization of the variational objective. In SVI, stochasticity is introduced with a doubly stochastic approximation of the expectation term, which is unbiasedly approximated using Monte Carlo and by selecting a subset of the training points (mini-batching) (Graves, 2011; Kingma & Welling, 2014).

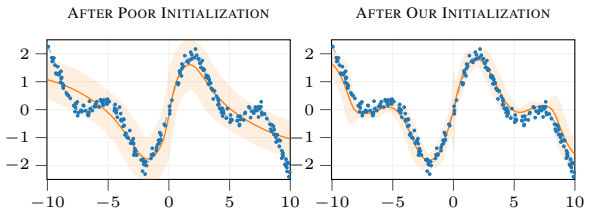


Figure 1: Due to poor initialization (**left**) SVI fails to converge even after 600+ epochs (RMSE $= 0.613$, MNLL $= 29.4$) while with our I-BLM (**right**) SVI easily recovers the function after few epochs (RMSE $= 0.315$, MNLL $= -5.8$). The architecture has three hidden layers with 500 neurons each, and uses the TANH activation function.

While SVI is an attractive and practical way to perform approximate inference for DNNs, there are limitations. For example, the form of the approximating distribution can be too simple to accurately approximate complex posterior distributions (Ha et al., 2016; Ranganath et al., 2015; Rezende & Mohamed, 2015). Furthermore, SVI increases the number of optimization parameters compared to optimizing model parameters through, e.g., loss minimization; for example, a fully factorized Gaussian posterior over model parameters doubles the number of parameters in the optimization compared to loss minimization. This has motivated research on other ways to perform approximate Bayesian inference for DNNs by establishing connections between variational inference and dropout (Gal & Ghahramani, 2016a;b; Gal et al., 2017).

A theoretical understanding of the optimization landscape of DNNs and CNNs is still in its early stages of development (Dziugaite & Roy, 2017; Garipov et al., 2018), and most works have focused on the practical aspects charac-

[1]Department of Data Science, EURECOM, France. Correspondence to: Simone Rossi <simone.rossi@eurecom.fr>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

terizing the optimization of their parameters (Duchi et al., 2011; Kingma & Ba, 2015; Srivastava et al., 2014). If this lack of theory is apparent for optimization of model parameters, this is even more so for the understanding of the optimization landscape of the objective in variational inference, where variational parameters enter in a nontrivial way in the objective (Graves, 2011; Rezende et al., 2014). Initialization plays a huge role in the convergence of SVI; the illustrative example in Figure 1 shows how a poor initialization can prevent SVI to converge to good solutions in short amount of time even for simple problems. The problem is even more severe for complex architectures, such as the ones that we discuss in the experiments; for example, SVI systematically converges to trivial solutions (posterior equal to the prior) when applied to CNNs, and we are not aware of any prior works applying SVI to CNNs.

In this work, we focus on this issue affecting SVI for DNNs and CNNs. While there is an established literature on ways to initialize model parameters of DNNs when minimizing its loss (Glorot & Bengio, 2010; Saxe et al., 2013; Mishkin & Matas, 2016), to the best of our knowledge, there is no study that systematically tackles this issue for SVI for Bayesian DNNs and CNNs. Inspired by the literature on residual networks (He et al., 2016) and greedy initialization of DNNs (Bengio et al., 2007; Mishkin & Matas, 2016), we propose a novel initialization strategy for SVI grounded on Bayesian linear modeling, which we call Iterative Bayesian Linear Modeling (I-BLM). Iterating from the first layer, I-BLM initializes the posterior at layer (l) by learning Bayesian linear models which regress from the input, propagated up to layer (l), to the labels.

We show how I-BLM can be applied in a scalable way and without considerable overhead to regression and classification problems, and how it can be applied to initialize SVI not only for DNNs but also for CNNs. Through a series of experiments, we demonstrate that I-BLM leads to faster convergence compared to other initializations inspired by prior work on loss minimization for DNNs. Furthermore, we show that I-BLM makes it possible for SVI with a Gaussian approximation applied to CNNs to compete with Monte Carlo Dropout (MCD; Gal & Ghahramani (2016b)) and noisy natural gradients (NOISY-KFAC; Zhang et al. (2018)), which are state-of-art methods to perform approximate inference for CNNs. In all, thanks to the proposed initialization, we make it possible to reconsider Gaussian SVI for DNNs and CNNs as a valid competitor to MCD and NOISY-KFAC, as well as highlight the limitations of SVI with a Gaussian posterior in applications involving CNNs.

In summary, in this work we make the following contributions: (1) we propose a novel way to initialize SVI for DNNs based on Bayesian linear models; (2) we show how this can be done for regression and classification; (3) we show how to apply our strategy to CNNs; (4) we empirically demonstrate that our proposal allows us to achieve performance superior to other initializations of SVI inspired by the literature on loss minimization; (5) for the first time, we achieve state-of-the-art performance with Gaussian SVI for large-scale CNNs.

2 Related Work

The problem of initialization of weights and biases in DNNs for gradient-based loss minimization has been extensively tackled in the literature since early breakthroughs in the field (Rumelhart et al., 1986; Baldi & Hornik, 1989). LeCun et al. (2012) is one of the seminal papers discussing practical tricks to achieve an efficient loss minimization through back-propagation.

More recently, Bengio et al. (2007) propose a greedy layer-wise unsupervised pre-training, which proved to help optimization and generalization. A justification can be found in Erhan et al. (2010), where the authors show that pre-training can act as regularization; by initializing the parameters in a region corresponding to a better basin of attraction for the optimization procedure, the model can reach a better local minimum and increase its generalization capabilities. Glorot & Bengio (2010) propose a simple way to estimate the variance for random initialization of weights, which makes it possible to avoid saturation both in forward and back-propagation steps. Another possible strategy can be found in the work by Saxe et al. (2013), that investigates the dynamics of gradient descend optimization, and proposes a random orthogonal initialization of the weights based on the singular value decomposition of a Gaussian random matrix. Building on this work, Mishkin & Matas (2016) propose a data-driven weight initialization by scaling the orthonormal matrix of weights to make the variance of the output as close to one as possible.

Variational inference addresses the problem of intractable Bayesian inference by reinterpreting inference as an optimization problem. Its origins can be traced back to early works in MacKay (1992); Hinton & van Camp (1993); Neal (1997). More recently, Graves (2011) proposes a practical way to carry out variational inference using stochastic optimization (Duchi et al., 2011; Zeiler, 2012; Sutskever et al., 2013; Kingma & Ba, 2015). Kingma & Welling (2014) propose a reparameterization trick that allows for the optimization of the variational lower bound through automatic differentiation. To decrease the variance of stochastic gradients, which impacts convergence speed, this work is extended using the so-called *local reparameterization trick*, where the sampling from the approximate posterior over model parameters is replaced by the sampling from the resulting distribution over the DNN units (Kingma et al., 2015).

In the direction of finding richer posterior families for variational inference, we mention the works on Normalizing Flows (Rezende & Mohamed, 2015; Kingma et al., 2016; Louizos & Welling, 2017; Huang et al., 2018). Alternatives can be found in Stein variational inference (Liu & Wang, 2016), quasi-Monte Carlo variational inference (Buchholz et al., 2018), variational boosting (Miller et al., 2017), noisy natural gradients (Zhang et al., 2018) and matrix Gaussian posterior (Louizos & Welling, 2016).

To the best of our knowledge, there is no study that either empirically or theoretically addresses the problem of initialization of parameters for SVI; we could only find a mention of this in Krishnan et al. (2018) for variational autoencoders. We aim to fill this gap by proposing a novel way to initialize parameters in SVI for probabilistic deep models.

3 Preliminaries

In this section we introduce some background material on Bayesian DNNs and SVI.

Bayesian Deep Neural Networks Bayesian DNNs are statistical models whose parameters (weights and biases) are assigned a prior distribution and inferred using Bayesian inference techniques. Bayesian DNNs inherit the modeling capacity of DNNs while allowing for quantification of uncertainty in model parameters and predictions. Considering an input $\mathbf{x} \in \mathbb{R}^{D_{\text{in}}}$ and a corresponding output $\mathbf{y} \in \mathbb{R}^{D_{\text{out}}}$, the relation between inputs and outputs can be seen as a composition of nonlinear vector-valued functions $\mathbf{f}^{(l)}$ for each hidden layer (l)

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \left(\mathbf{f}^{(L-1)} \circ \ldots \circ \mathbf{f}^{(0)}\right)(\mathbf{x}) . \tag{1}$$

Let $\mathbf{W}$ be a collection of all model parameters (weights and biases) $W^{(l)}$ at all hidden layers. Each neuron computes its output as

$$f_i^{(l)} = \phi(\mathbf{w}_i^{(l)^\top} \mathbf{f}^{(l-1)}) , \tag{2}$$

where $\phi(\cdot)$ denotes a so-called activation function, which introduces a nonlinearity at each layer. Note that, for simplicity of notation, we absorbed the biases into $\mathbf{w}_i^{(l)}$.

Given a prior over $\mathbf{W}$, the objective of Bayesian inference is to find the posterior distribution over all model parameters $\mathbf{W}$ using the available input data $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ associated with labels $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$

$$p(\mathbf{W}|X, Y) = \frac{p(Y|X, \mathbf{W})p(\mathbf{W})}{p(Y|X)} . \tag{3}$$

Bayesian inference for DNNs is analytically intractable and it is necessary to resort to approximations. One way to recover tractability is through the use of variational inference techniques as described next.

Stochastic Variational Inference In variational inference, we introduce a family of distributions $q_{\boldsymbol{\theta}}(\mathbf{W})$, parameterized through $\boldsymbol{\theta}$, and attempt to find an element of this family which is as close to the posterior distribution of interest as possible (Jordan et al., 1999). This can be formulated as a KL (Kullback, 1959) minimization problem as follows:

$$\arg\min_{\boldsymbol{\theta}} \{\text{KL}\left[q_{\boldsymbol{\theta}}(\mathbf{W})||p(\mathbf{W}|X, Y)\right]\} . \tag{4}$$

Simple manipulations allow us to rewrite this expression as the negative lower bound (NELBO) to the log-marginal likelihood of the model (see supplementary material)

$$\text{NELBO} = \text{NLL} + \text{KL}\left[q_{\boldsymbol{\theta}}(\mathbf{W})||p(\mathbf{W})\right] , \tag{5}$$

where the first term is the expected negative log-likelihood $\text{NLL} = \mathbb{E}_{q_{\boldsymbol{\theta}}}\left[-\log p(Y|X, \mathbf{W})\right]$, and the second term acts as regularizer, penalizing distributions $q_{\boldsymbol{\theta}}(\mathbf{W})$ that deviate too much from the prior; note that the KL term is computable analytically for many choices of prior and approximate posterior form. When the likelihood factorizes across data points, we can unbiasedly estimate the expectation term randomly selecting a mini-batch $\mathcal{B}$ of m out of n training points, which is suitable for stochastic gradient optimization

$$\text{NLL} \approx -\frac{n}{m} \sum_{\mathbf{x},\mathbf{y} \in \mathcal{B}} \mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\log p(\mathbf{y}|\mathbf{x}, \mathbf{W})\right] . \tag{6}$$

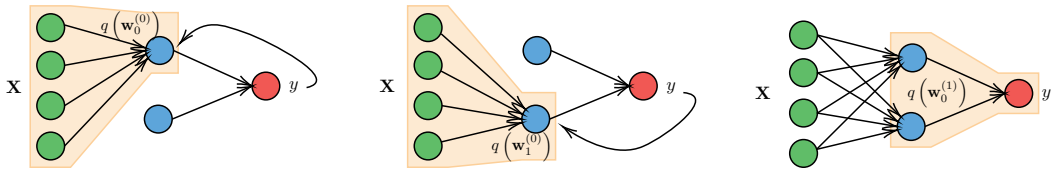
Each term in the sum can be further unbiasedly estimated using N_{MC} Monte Carlo samples as

$$\mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\log p(\mathbf{y}|\mathbf{x}, \mathbf{W})\right] = \frac{1}{N_{\text{MC}}} \sum_{i=1}^{N_{\text{MC}}} \log p(\mathbf{y}|\mathbf{x}, \mathbf{W}_i) , \tag{7}$$

where $\mathbf{W}_i \sim q_{\boldsymbol{\theta}}(\mathbf{W})$. Following Kingma & Welling (2014), each sample $\mathbf{W}_i$ is constructed using the reparameterization trick, which yields a deterministic dependence of the NELBO w.r.t. $\boldsymbol{\theta}$. Alternatively, it is possible to determine the distribution of the DNN units $f_i^{(l)}$ before activation from $q_{\boldsymbol{\theta}}(\mathbf{W})$. This trick, known as *the local reparameterization trick*, considerably reduces the variance of the stochastic gradient w.r.t. $\boldsymbol{\theta}$ and achieves faster convergence (Kingma et al., 2015).

4 Proposed Method

In this section, we introduce our proposed Iterative Bayesian Linear Model (I-BLM) initialization for SVI. We first introduce I-BLM for regression with DNNs, and we then show how this can be extended to classification and to CNNs.



(a) (b) (c)

Figure 2: Visual representation of the proposed method for initialization. In (a) and (b), we learn two Bayesian linear models, whose outputs are used in (c) to infer the following layer.

4.1 Initialization of DNNs for Regression

In order to initialize the weights of DNNs, we proceed iteratively as follows. Before applying the nonlinearity through the activation function, each layer in a Bayesian DNN can be seen as multivariate Bayesian linear regression model. We use this observation as an inspiration to initialize SVI as follows. Starting from the first layer, we can set the parameters of $q(W^{(0)})$ by running Bayesian linear regression with inputs X and labels Y. After this, we initialize the approximate posterior over the weights at the second layer $q(W^{(1)})$ by running Bayesian linear regression with inputs $X = \Phi(X\tilde{W}^{(0)})$ and labels Y. Here, $\Phi(\cdot)$ denotes the element-wise application of the activation function to the argument, whereas $\tilde{W}^{(0)}$ is a sample from $q(W^{(0)})$. We then proceed iteratively in the same way up to the last layer. Figure 2 gives an illustration of the proposed method for a simple architecture.

The intuition behind I-BLM is as follows. If one layer is enough to capture the complexity of a regression task, we expect to be able to learn an effective mapping right after the initialization of the first layer. In this case, we also expect that the mapping at the next layers implements simple transformations, close to the identity. Learning a set of weights with these characteristics starting from a random initialization is far from trivial, which also motivated the work on residual networks (He et al., 2016). Our I-BLM initialization takes this observation as an intuition to initialize SVI for general deep models.

Algorithm 1: Sketch of the I-BLM Initializer

Inputs : Model M, Dataset D
foreach *layer in* M **do**
  foreach *outfeature in layer* **do**
    $X, Y \leftarrow$ next batch in D;
    propagate X;
    $X_{\text{BLM}} \leftarrow$ output of previous layer;
    if *layer is convolutional* **then** ▷ ref 4.4
      $X_{\text{BLM}} \leftarrow$ patch extraction(X_{BLM});
    if *likelihood is classification* **then** ▷ ref 4.3
      $\text{var}(Y_{\text{BLM}}) \leftarrow \log\left[(Y+\alpha)^{-1}+1\right]$;
      $\text{mean}(Y_{\text{BLM}}) \leftarrow \log(Y+\alpha) - \text{var}(Y_{\text{BLM}})/2$;
    else
      $Y_{\text{BLM}} \leftarrow Y$;
    $p(\mathbf{w}|X,Y) \leftarrow \text{BLM}(X_{\text{BLM}}, Y_{\text{BLM}})$; ▷ ref 4.1
    $q(\mathbf{w}) \leftarrow$ best approx. of $p(\mathbf{w}|X,Y)$; ▷ ref 4.2

From a complexity point of view, denoting by $h^{(l)}$ the number of output neurons at layer (l), this is equivalent to $h^{(l)}$ univariate Bayesian linear models. Instead of using the entire training set to learn the linear models, each one of these is inferred based on a random mini-batch of data, whose inputs are propagated through the previous layers. The complexity of I-BLM is linear in the batch size and cubic in the number of neurons to be initialized. Later on in this Section, we will provide an evaluation of the effect of batch size and a timing profiling of I-BLM.

4.2 From the Bayesian linear model posterior to the variational approximation

The proposed I-BLM initialization of variational parameters can be used with any choice for the form of the approximate posterior. The exact posterior of Bayesian linear regression is not factorized, so one needs to match this with the form of the chosen approximate posterior. For simplicity of notation, let $\mathbf{w}$ be the parameters of interest in Bayesian linear regression for a given output $\mathbf{y} = Y_{\cdot i}$. We can formulate this problem by minimizing the KL divergence from $q(\mathbf{w})$ to the actual posterior $p(\mathbf{w}|X,\mathbf{y})$. In the case of a fully factorized approximate posterior over the weights this minimization can be done analytically resulting in the mean being equal to the mean of $p(\mathbf{w}|X,\mathbf{y})$ and the variances $(s_i^2)^{-1} = \Sigma_{ii}^{-1}$; see the supplementary material for the full derivation. Similar results can be also obtained for different posterior distributions, such as Gaussian posteriors with full or low-rank covariance, or matrix-variate Gaussian posteriors (Louizos & Welling, 2016).

4.3 Initialization for Classification

In this section we show how our proposal can be extended to k-class classification problems. We assume a one-hot encoding of the labels, so that Y is an $n \times k$ matrix of zeros and ones (one for each row of Y). Recently, it has been shown that it is possible to obtain an accurate modeling of the posterior over classification functions by applying regression on a transformation of the labels (Milios et al., 2018). This is interesting because it allows us to apply

Bayesian linear regression as before in order to initialize SVI for DNNs.

The transformation of the labels is based on the formalization of a simple intuition, which is the inversion of the softmax transformation. One-hot encoded labels are viewed as a set of parameters of a degenerate Dirichlet distribution. We resolve the degeneracy of the Dirichlet distribution by adding a small regularization, say $\alpha = 0.01$, to the parameters. At this point, we leverage the fact that Dirichlet distributed random variables can be constructed as a ratio of Gamma random variables, that is, if $x_i \sim \text{Gamma}(a_i, b)$, then $\frac{x_i}{\sum_j x_j} \sim \text{Dir}(\mathbf{a})$. We can then approximate the Gamma random variables with log-Normals by moment matching, which become Gaussian after a logarithm transformation. By doing so, we obtain a representation of the labels which allows us to use standard regression with a Gaussian likelihood, and which retrieves an approximate Dirichlet when mapping predictions back using the softmax transformation. As a result, the latent functions obtained represent probabilities of class labels.

The only small complication is that the transformation imposes a different noise level for labels that are 0 or 1, and this is due to the non-symmetric nature of the transformation. Nevertheless, it is a simple matter to extend Bayesian linear regression to handle heteroscedasticity; see the supplementary material and Milios et al. (2018) for more insights on the transformation to apply regression on classification problems.

4.4 Initialization of CNNs

I-BLM can also be applied to CNNs. Convolutional layers are commonly implemented as matrix multiplication (e.g. as a linear model) between a batched matrix of patches and a reshaped filter matrix (Jia, 2014). Rather than using the outputs of the previous layer as they are, for convolutional layers each Bayesian linear model learns the mapping from spatial patches to output features. In Algorithm 1 we summarize a sketch of the proposed method for regression as well as for classification and convolutional layers.

4.5 General Insights on I-BLM

Previously we claimed that (i) small batches of data are sufficient to solve the Bayesian linear model and that (ii) our initialization does not incur significant overheads. We now aim to justify such claims. We initialize a CNN (LeNet-5) on MNIST with an increasing number of samples per batch; Figure 3(a) shows how test log-likelihood is affected by this choice. Using the full training set leads to a better estimate of the posterior. The mini-batch size affects also the heterogeneity of the posteriors, which vanishes when using the full training set. Nonetheless, we show that from

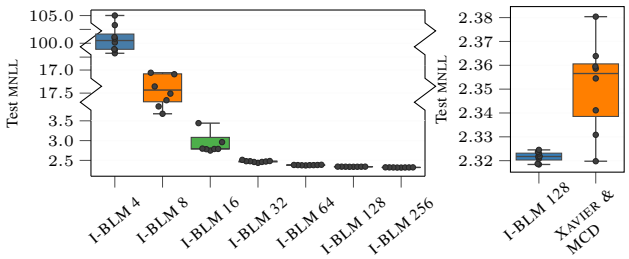


Figure 3: Comparison of test MNLL after initialization of LeNet-5 for MNIST averaged out of eight successive runs. On the **left**, with different batch sizes, on the **right** with MCD.

64/128 samples the improvement on the test MNLL is only marginal. The same experiment is also repeated comparing test MNLL after initialization between SVI with I-BLM and MCD (Figure 3(b)). Similar comments apply also for this case: I-BLM allows the training to start from a lower negative log-likelihood. Finally, Figure 4 reports the test MNLL after initialization as a function of the time required (orange points correspond to Pareto-optimal points). Before training, three out of four optimal initializers are I-BLM.

5 Experimental Results

In this section, we compare different initialization algorithms for SVI to prove the effectiveness of I-BLM. We propose a number of competitors inspired from the literature developed for loss minimization in DNNs and CNNs. In the case of CNNs, we also compare with Monte Carlo Dropout (MCD; Gal & Ghahramani (2016a)) and Natural Noisy Gradients (NOISY-KFAC; Zhang et al. (2018)), which represent the state-of-the-art references for inference in Bayesian CNNs. At layer (l), we choose priors $p(W^{(l)}) = \prod_{i,j} \mathcal{N}(w_{i,j}^{(l)}|0, \frac{1}{D_{\text{in}}^{(l)}})$, where $D_{\text{in}}^{(l)}$ denotes the number of input features at layer (l), and focus on fully-factorized variational posteriors $q(W^{(l)}) = \prod_{i,j} \mathcal{N}(w_{i,j}^{(l)}|\mu_{i,j}^{(l)}, (\sigma^2)_{i,j}^{(l)})$; here are the methods that we compare to initialize $\mu_{i,j}^{(l)}$ and $(\sigma^2)_{i,j}^{(l)}$.

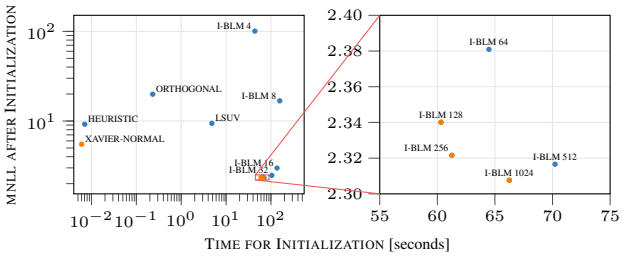


Figure 4: On the **left**, comparison of initialization time versus test MNLL, averaged out of eight successive runs (on the **right**, magnification of the small portion of the plot). Orange corresponds to the Pareto frontier.

Uninformative The posteriors at each layer are initialized with $\mu_{i,j}^{(l)} = 0$ and $(\sigma^2)_{i,j}^{(l)} = 1$.

Random Heuristic An extension to commonly used heuristics with $\mu_{i,j}^{(l)} = 0$ and $(\sigma^2)_{i,j}^{(l)} = \frac{1}{D_{\text{in}}^{(l)}}$. Because this is the same as for the prior, this yields an initial KL divergence in the NELBO equal to zero.

Xavier Normal Originally proposed by Glorot & Bengio (2010), it samples all weights independently from a Gaussian distribution with zero mean and $(\sigma^2)_{i,j}^{(l)} = \frac{2}{D_{\text{in}}^{(l)} + D_{\text{out}}^{(l)}}$. This variance-based scaling avoids issues with vanishing or exploding gradients. We extend this to SVI by directly setting $\mu_{i,j}^{(l)} = 0$ and $(\sigma^2)_{i,j}^{(l)} = \frac{2}{D_{\text{in}}^{(l)} + D_{\text{out}}^{(l)}}$, given that the sampling is performed during the Monte Carlo estimate of the log-likelihood.

Orthogonal Starting from the analysis of learning dynamics of DNNs with linear activations, Saxe et al. (2013) propose an initialization scheme with orthonormal weight matrices. The idea is to decompose a Gaussian random matrix onto an orthonormal basis, and use the resulting orthogonal matrix for initialization. We adapt this method to SVI by initializing the mean matrix with the orthogonal matrix and $(\sigma^2)_{i,j}^{(l)} = \frac{1}{D_{\text{in}}^{(l)}}$. In the experiments, we make use of the PYTORCH QR-decomposition (Paszke et al., 2017).

Layer-Sequential Unit-Variance (LSUV) Starting from the orthogonal initialization, Mishkin & Matas (2016) propose a data-driven greedy layer-wise variance scaling of the weight matrices. We implement Layer-Sequential Unit-Variance (LSUV) for the means, while the variances are set to $(\sigma^2)_{i,j}^{(l)} = \frac{1}{D_{\text{in}}^{(l)}}$.

5.1 Experiments

Throughout the experiments, we use the ADAM optimizer (Kingma & Ba, 2015) with learning rate 10^{-3}, batch size 64, and 16 Monte Carlo samples at training time and 128 at test time. All experiments are run on a server equipped with two 16c/32t Intel Xeon CPU and four NVIDIA Tesla P100, with a maximum time budget of 24 hours (never reached). To better understand the effectiveness of different initializations, all learning curves are plotted w.r.t. training iteration rather than wall-clock time.

Regression with a shallow architecture In this experiment we compare initialization methods for a shallow DNN architecture on two datasets. The architecture used in these experiments has one single hidden layer with 100 hidden neurons and RELU activations. We impose that the approximate posterior has fully factorized covariance. Figure 5 shows the learning curves on the POWERPLANT ($n = 9568$, $d = 4$) and PROTEIN ($n = 45730$, $d = 9$) datasets, repeated over five different train/test splits. I-BLM allows for a better initialization compared to the competitors, leading to a lower root mean square error (RMSE) and lower mean negative log-likelihood (MNLL) on the test for a given computational budget. We refer the reader to the supplementary material for a more detailed analysis of the results.

Regression with a deeper architecture Similar considerations hold when increasing the depth of the model, keeping the same experimental setup. Figure 6 shows the progression of the RMSE and MNLL error metrics when using SVI to infer parameters of a DNN with five hidden layers and 100 hidden neurons per layer, and RELU activations. Again, the proposed initialization allows SVI to converge faster than when using other initializations.

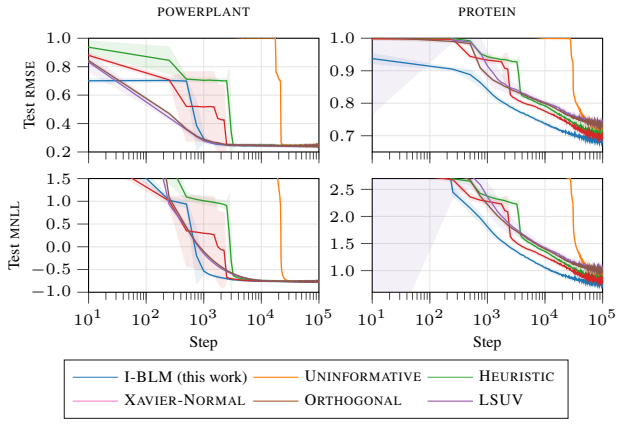


Figure 5: Progression of test RMSE and test MNLL with different initializations on a shallow architecture.

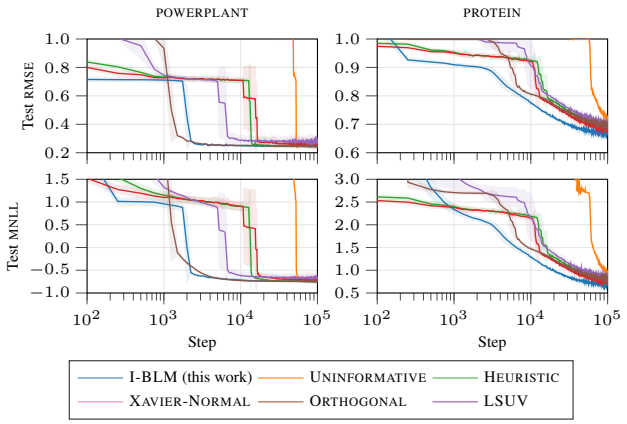


Figure 6: Progression of test RMSE and test MNLL with different initializations on a deep architecture.

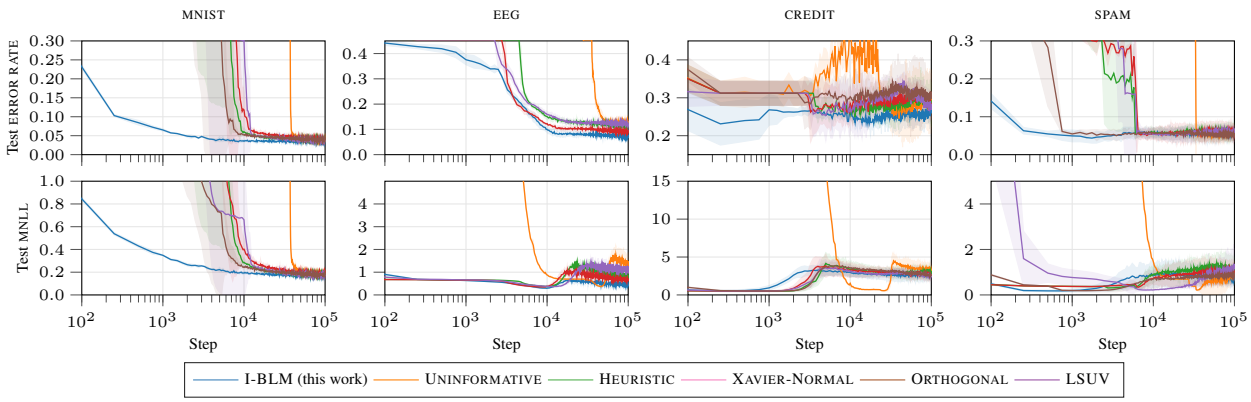


Figure 7: Progression of test ERROR RATE and test MNLL with different initializations on classification problems.

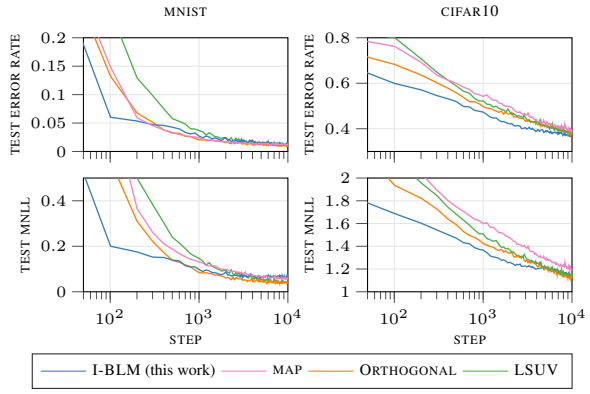


Figure 8: Progression of test ERROR RATE and test MNLL for different initializations using LENET-5 on MNIST and CIFAR10. Note: none of the runs with UNINFORMATIVE, HEURISTIC and XAVIER converged, so these results are not shown.

Classification with a deep architecture Using the same deep DNN architecture as in the last experiment (five hidden layers with 100 neurons), we tested I-BLM with classification problems on MNIST ($n = 70000$, $d = 784$), EEG ($n = 14980$, $d = 14$), CREDIT ($n = 1000$, $d = 24$) and SPAM ($n = 4601$, $d = 57$). Interestingly, with this architecture, some initialization strategies struggled to converge, e.g., UNINFORMATIVE on MNIST and LSUV on EEG. The gains offered by I-BLM achieves are most apparent on MNIST. After less than 1000 training steps (less than an epoch), I-BLM makes SVI reach a test accuracy greater than 95%; other initializations reach such performance much later during training. Even after 100 epochs, SVI inference initialized with I-BLM provides on average an increase up to 14% of accuracy at test time. Full results are reported in the supplementary material.

Experiments on CNNs For this experiment, we implemented a Bayesian version of the original LENET-5 architecture proposed by LeCun et al. (1998) with two convolutional layers of 6 and 16 filters, respectively and RELU activations applied after all convolutional layers and fully-connected layers. We tested our framework on MNIST and on CIFAR10. The only initialization strategies that achieve convergence are ORTHOGONAL and LSUV, along with I-BLM; the other methods systematically make the optimization push the posterior back to the prior. We include a further initialization based on the MAP solution. We optimize the loss for the same amount of time required by I-BLM to complete the initialization, and we use this solution to initialize the $\mu_{i,j}^{(l)}$, while we set $\log[(\sigma^2)_{i,j}^{(l)}] = -5.5$. Figure 8 reports the progression of ERROR RATE and MNLL. For both MNIST and CIFAR10, I-BLM places the parameters where the network can consistently deliver better performance both in terms of ERROR RATE and MNLL throughout the entire learning procedure.

Comparison with large scale models and non-Gaussian approximation Monte Carlo Dropout (MCD; Gal & Ghahramani (2016b)) offers a simple and effective way to perform approximate Bayesian CNN inference, thanks to the connection that the Authors have established between dropout and variational inference. In this experiment, we aim to compare and discuss benefits and disadvantages of using a Gaussian posterior approximation with respect to the Bernoulli approximation that characterizes MCD. For a fair comparison, we implemented the same LENET-5 architecture and the same learning procedure in Gal & Ghahramani (2016b)[1]. In particular, for MNIST, the two convolutional layers have 20 and 50 filters, respectively. Dropout layers are placed after every convolutional and fully-connected layers with a dropout probability of 0.5. To replicate the results in Gal & Ghahramani (2016b), we used the same learning rate policy $\texttt{base-lr} \times (1 + \xi \times \texttt{iter})^{-p}$ with $\xi = 0.0001$, $p = 0.75$, $\texttt{base-lr} = 0.01$ and weight decay of 0.0005. Figure 9 shows the learning curves.

[1] https://github.com/yaringal/DropoutUncertaintyCaffeModels

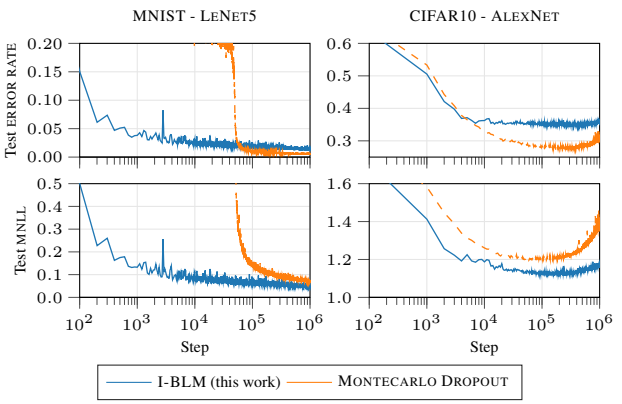
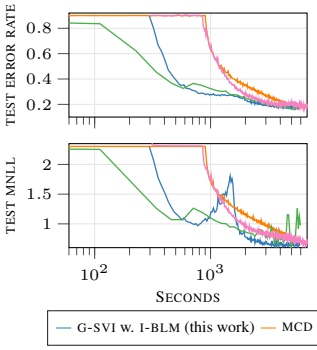


Figure 9: Comparison between SVI with Gaussian approximation and MCD on MNIST and CIFAR10 with LeNet-5 and AlexNet.



	MNLL	ERROR
G-SVI & I-BLM	**0.637**	**0.167**
G-SVI & MAP	0.750	0.201
MCD	0.821	0.215
NOISY-KFAC	0.750[a]	**0.164**

[a] Due to high variance in the test MNLL, in this case we report the minimum of the last 5 epochs.

Figure 10 & Table 1: Comparison between Gaussian factorized SVI initialized with I-BLM and with a MAP solution, MCD and NOISY-KFAC on VGG16 with CIFAR10

Monte Carlo Dropout achieves state-of-art ERROR RATE but the form assumed by MCD for the posterior is reflected on an higher MNLL compared to SVI with a Gaussian posterior. Provided with a nontrivial initialization, Gaussian SVI can better fit the model and deliver a better quantification of uncertainty.

We report also ERROR RATE and MNLL for SVI with I-BLM and MCD on AlexNet (Krizhevsky et al., 2012). The CNN is composed by a stack of five convolutional layers and three fully-connected layers for a total of more than 1M parameters (2M for SVI). In this experiment, we have experienced the situation in which, due to the overparameterization of the model, the NELBO is completely dominated by the KL divergence. Therefore, the prior has a large influence on the optimization, so we decided to follow the approach in Graves (2011), allowing for a phase of optimization of the variances of the prior over the parameters. The results are reported in Figure 9. Once again, we show that SVI with I-BLM provides a lower negative log-likelihood with respect to Bernoulli approximation in MCD.

Finally, we demonstrate that – provided with a sensible initialization – even simple factorized Gaussian posterior can achieve state-of-the-art performance on CIFAR10 with VGG16[2], a large scale CNN (Simonyan & Zisserman, 2014). In this experiment, in addition to MCD, we also compare with NOISY-KFAC, an approximation of matrix-variate Gaussian posterior using noisy natural gradients introduced by Zhang et al. (2018). The four models are trained with a time budget of 100 minutes for the entire end-to-end training (curves are shifted by the initialization time). In the cases of Gaussian SVI, we implement a policy where the KL term is gradually included in the NELBO, similarly to Bowman et al. (2016) and Sønderby et al. (2016) (more details in the supplementary material). Results are shown in Figure 10 and in the adjacent table. Gaussian SVI with I-BLM delivers state-of-art test MNLL while also providing a competitive test ERROR RATE.

[2] We implemented the same architecture as in Zhang et al. (2018)

Extended experimental evaluation We refer the reader to the supplementary material for additional insights of I-BLM compared with other initialization methods, analysis of out-of-sample uncertainty estimation, and tests of calibration properties of deep classifiers.

6 Conclusions

This work fills an important gap in the literature of Bayesian deep learning, that is how to effectively initialize variational parameters in SVI. We proposed a novel way to do so, I-BLM, which is based on an iterative layer-wise initialization based on Bayesian linear models. Through a series of experiments, including regression and classification with DNNs and CNNs, we demonstrated the ability of our approach to consistently initialize the optimization in a way that makes convergence faster than alternatives inspired from the state-of-the-art in loss minimization for deep learning.

Thanks to I-BLM, it was possible to carry out an effective comparison with state-of-the-art methods to carry out approximate inference for DNNs and CNNs. This suggests a number of directions to investigate to improve on SVI and Bayesian CNNs. We found that the choice of the prior plays an important role in the behavior of the optimization, so we are investigating ways to define sensible priors for these models. Furthermore, we are looking into extending I-BLM to initialize SVI with posterior distributions beyond Gaussian and to other deep models, such as Deep Gaussian Processes and Bayesian deep generative models.

Acknowledgements

MF gratefully acknowledges support from the AXA Research Fund.

References

Baldi, P. and Hornik, K. Neural Networks and Principal Component Analysis: Learning from Examples without Local Minima. *Neural Networks*, 2(1):53–58, 1989.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy Layer-Wise Training of Deep Networks. In Schölkopf, B., Platt, J. C., and Hoffman, T. (eds.), *Advances in Neural Information Processing Systems 19*, pp. 153–160. MIT Press, 2007.

Bishop, C. M. *Pattern recognition and machine learning*. Springer, 1st ed. 2006. Corr. 2nd printing 2011 edition, Aug 2006.

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

Buchholz, A., Wenzel, F., and Mandt, S. Quasi-Monte Carlo variational inference. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 668–677, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Duchi, J., Hazan, E., and Singer, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159, Jul 2011.

Dziugaite, G. K. and Roy, D. M. Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*, 2017.

Erhan, D., Courville, A., and Vincent, P. Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research*, 11:625–660, 2010.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016a. PMLR.

Gal, Y. and Ghahramani, Z. Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference, Jan 2016b. arXiv:1506.02158.

Gal, Y., Hron, J., and Kendall, A. Concrete Dropout. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 3581–3590. Curran Associates, Inc., 2017.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8789–8798. Curran Associates, Inc., 2018.

Ghahramani, Z. Probabilistic Machine Learning and Artificial Intelligence. *Nature*, 521(7553):452–459, May 2015.

Glorot, X. and Bengio, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

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.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On Calibration of Modern Neural Networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, International Convention Centre, Sydney, Australia, Aug 2017. PMLR.

Ha, D., Dai, A. M., and Le, Q. V. HyperNetworks, Sep 2016. arXiv:1609.09106.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, 2016.

Hinton, G. E. and van Camp, D. Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, 1993.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural Autoregressive Flows. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2078–2087, Stockholm, Sweden, 10–15 Jul 2018. PMLR.

Jia, Y. *Learning Semantic Image Representations at a Large Scale*. PhD thesis, EECS Department, University of California, Berkeley, Berkeley, California, 2014.

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

Kendall, A. and Gal, Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5574–5584. Curran Associates, Inc., 2017.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *Proceedings of the Third International Conference on Learning Representations*, San Diego, USA, May 2015.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *Proceedings of the Second International Conference on Learning Representations*, Banff, Canada, Apr 2014.

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

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved Variational Inference with Inverse Autoregressive Flow. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4743–4751. Curran Associates, Inc., 2016.

Krishnan, R., Liang, D., and Hoffman, M. On the Challenges of Learning with Inference Networks on Sparse, High-Dimensional Data. In Storkey, A. and Perez-Cruz, F. (eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 143–151, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

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.

Kullback, S. *Information Theory and Statistics*. IEEE, 1959.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6402–6413. Curran Associates, Inc., 2017.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998.

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. *Efficient BackProp*, pp. 9–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

Liu, Q. and Wang, D. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 2378–2386. Curran Associates, Inc., 2016.

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

Louizos, C. and Welling, M. Multiplicative Normalizing Flows for Variational Bayesian Neural Networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2218–2227, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

MacKay, D. J. C. A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3): 448–472, May 1992.

Milios, D., Camoriano, R., Michiardi, P., Rosasco, L., and Filippone, M. Dirichlet-based Gaussian Processes for Large-scale Calibrated Classification. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 6008–6018. Curran Associates, Inc., 2018.

Miller, A. C., Foti, N. J., and Adams, R. P. Variational Boosting: Iteratively Refining Posterior Approximations. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2420–2429, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Mishkin, D. and Matas, J. All You Need is a Good Init. *CoRR*, abs/1511.06422, 2016.

Neal, R. M. Bayesian Learning for Neural Networks. *Journal of the American Statistical Association*, 1997.

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.

Ranganath, R., Tang, L., Charlin, L., and Blei, D. Deep Exponential Families. In Lebanon, G. and Vishwanathan, S. V. N. (eds.), *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pp. 762–771, San Diego, California, USA, 09–12 May 2015. PMLR.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1530–1538, Lille, France, 07–09 Jul 2015. PMLR.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In Xing, E. P. and Jebara, T. (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536, Oct 1986.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact Solutions to the Nonlinear Dynamics of Learning in Deep Linear Neural Networks. *CoRR*, abs/1312.6:1–22, Dec 2013.

Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.

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

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(1):1929–1958, Jan 2014.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In Dasgupta, S. and McAllester, D. (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

Zeiler, M. D. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701, 2012.

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