Good Initializations of Variational Bayes for Deep Models

Simone Rossi, Pietro Michiardi, Maurizio Filippone

Objectives and Contributions

**Initialization of variational parameters** has a huge role in the convergence of stochastic variational inference but received little to no attention in current literature.

**Contributions**
- New initialization for SVI based on Bayesian linear models;
- Applied to regression, classification and CNNs;
- Experimental comparison against other initializations;
- SoTA performance with Gaussian SVI on large-scale CNNs.

Stochastic Variational Inference - SVI

A DNN is a composition of nonlinear vector-valued functions \( f^{(i)} \)

\[ f(x) = \left( f^{(L-1)}(W^{(L-1)}) \circ \ldots \circ f^{(0)}(W^{(0)}) \right)(x) \]

**Objective of Bayesian inference**

Prior on model parameters

Posterior over the weights

Intractable for DNNs

Marginal Likelihood

SVI reformulates this problem as minimization of the negative evidence lower bound (or ELBO) under an approximate distribution \( q_\theta(W) \):

\[ q_\theta(W) = \arg \min_{q(W)} \text{ELBO} \]

\( \text{ELBO} = \mathbb{E}_q[-\log p(Y|X,W)] + \mathcal{K}(q_\theta(W)||p(W)) \)

Commonly used family of variational distributions: mean field Gaussian (or fully factorized Gaussian)

\[ q_\theta(W) = \prod_i \mathcal{N}(w^{(i)}; \mu^{(i)}_\theta, \sigma^{(i)}_\theta) \quad \theta = \{ \mu^{(i)}_\theta, \sigma^{(i)}_\theta \} : 1 = 0, \ldots, L - 1 \]

How do we initialize \( \theta \)?

A poor initialization can prevent SVI from converging to good solutions even for simple problems. It is even more severe for complex architectures, where SVI systematically converges to trivial solutions.

After Poor Initialization

After Our Initialization

![Graph showing comparison between poor and good initialization](image)

References


Checkout the Full Paper!


Iterative Bayesian Linear Modeling Initializer - I-BLM

**Figure**: Representation of I-BLM. On (top) we learn two Bayesian linear models, whose outputs are used on the (bottom) for the following layer.

**In a nutshell**

- Inspired by residual networks and greedy initialization of DNNs.
- Grounded on Bayesian Linear regression but extended to classification and to convolutional layers.
- Regression on transformed labels obtained through the interpretation of classification labels as the coefficients of a degenerate Dirichlet distribution.
- Scalability achieved thanks to mini-batching.

But how does it work?

Transform the labels if it’s a classification task [3]. For each layer (1):

- Propagate a mini-batch of \( X \) up to the previous layer (1 - 1);
- Extract the patches if it’s a convolutional layer;
- Learn a Bayesian linear model and use its solution to initialize \( q_\theta(W^{(1)}) \).

Effect of batch-size: the full training set leads to a better estimate of the posteriors.

Bayesian Linear Regression - BLR

Effect of batch-size: the full training set leads to a better estimate of the posteriors.

Some more insights!

Timing profiling (LENET-5): before training, 3 out of 4 optimal initializers are I-BLM

![Graph showing timing profiling](image)

Regression and Classification on Bayesian DNNs

**Figure**: Progression of test error and test MNLL with different initializations on a 5x100 architecture.

I-BLM for Bayesian CNNs - VGG16

- Another initialization for Gaussian SVI based on a MAP optimization (MAP INIT).

**Figure & Table**: Comparison between Gaussian factorized SVI, MCD and NOISY-KFAC on VGG16 with CIFAR10.

I-BLM (this work) Uninformative Heuristic

Xavier-Normal Orthogonal LSUV

![Graph showing comparison between different initializations](image)