Spike-and-Slab Probabilistic Backpropagation: When Smarter Approximations Make No Difference

Evan Ott and Sinead Williamson

The University of Texas at Austin
Department of Statistics and Data Sciences
College of Natural Sciences

Overview

Bayesian approaches to learning neural networks incorporate model uncertainty. Probabilistic backpropagation (PBP) [1] uses closed-form Gaussian approximations for the posterior and messages, but this ignores sparsity inherent to nonlinearities in messages. A spike-and-slab approximation should better represent sparsity and improve performance.

Background

PBP uses a ReLU-based FFNN, scaling inputs to each linear layer. PBP assumes Gaussian approximate posterior for model weights:

$$q(\mathcal{W}) = \prod_{\ell=1}^{L} \prod_{i=1}^{n_\ell} \prod_{j=1}^{n_{\ell-1}+1} N(w_{ij,\ell} | m_{ij,\ell}, v_{ij,\ell})$$

Uncertainty propagated by moment-matching messages

$$q(z_\ell) = N(z_\ell | m^{z_\ell}, v^{z_\ell})$$

True distribution incorporates sparsity; first ReLU layer yields a spike and truncated Gaussian mixture

$$(1-\rho)\delta_0 + \rho \mathrm{TN}_{(0,\infty)}(m, v)$$

Our Method (SSPBP)

Propose spike-and-slab message approximation

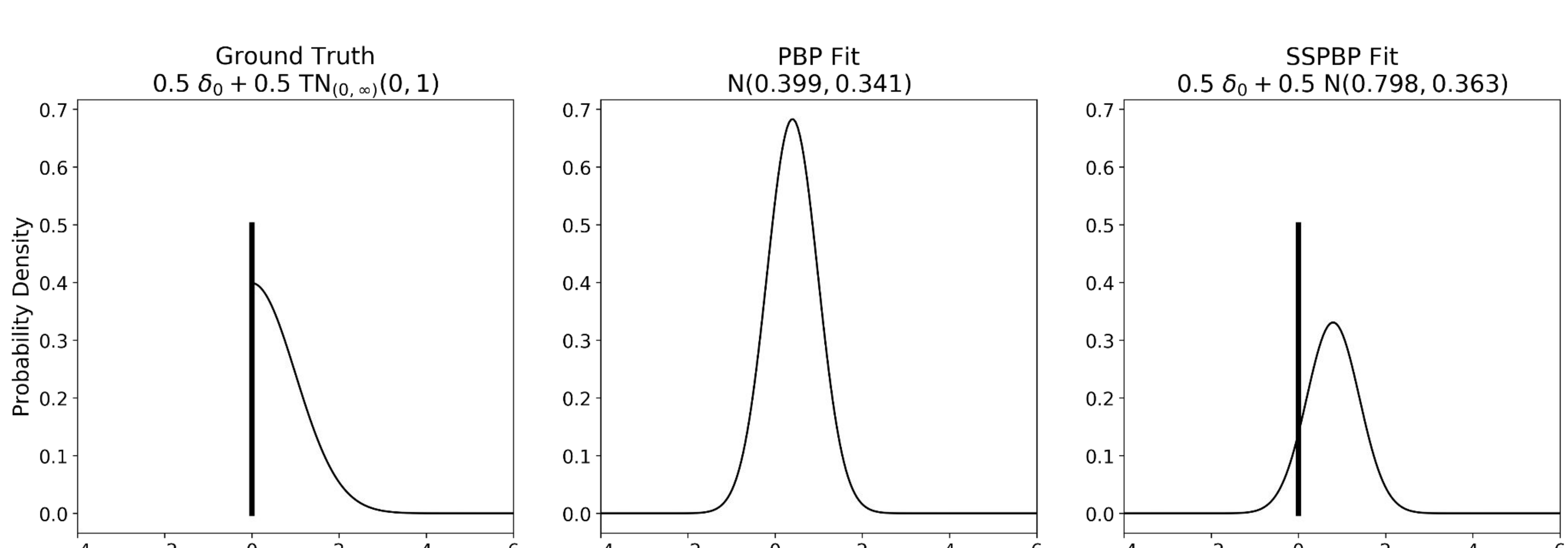
$$(1-\rho)\delta_0 + \rho N(m, v)$$

Obtained optimal parameters by minimizing KL(p||q)

$$\mathbb{P}_q[Z \neq 0] = \mathbb{P}_p[Z \neq 0], \;\; \mathbb{E}_q[Z] = \mathbb{E}_p[Z], \;\; \mathbb{V}_q[Z] = \mathbb{V}_p[Z]$$

Replaced approximations for linear and ReLU layers

Compare true distribution with PBP and SSPBP



Simulation Study

Explored behavior, comparing MMD [2] of samples

$$Y = \mathrm{ReLU}(XW) = \max(XW, 0)$$

p_X	p_W	% Saturated	PBP	SSPBP
$N(0,1)$	$N(0,1)$	49.86%	0.066	**0.020**
$N(1,1)$	$N(3,1)$	16.24%	0.031	**0.015**
$N(1,1)$	$N(-3,1)$	84.44%	0.21	**0.0038**
$N(3,1)$	$N(3,1)$	0.22%	**0.0043**	0.0051
$N(3,1)$	$N(-3,1)$	99.72%	0.017	**0.00024**

Results

We compared the test-set RMSE for regression datasets, with various choices of hidden layers (see paper for more results, including test-set LL), showing no difference in performance.

Dataset	PBP	SSPBP
Boston Housing	3.097±0.147	**2.997±0.165**
Combined Cycle Power Plant	**4.088±0.067**	4.096±0.066
Concrete Compression Strength	6.031±0.161	**5.921±0.158**
Energy Efficiency	**1.477±0.043**	1.660±0.112
Kin8nm	0.111±0.004	**0.109±0.002**
Naval Propulsion	**0.006±0.000**	**0.006±0.000**
Wine Quality Red	0.653±0.012	**0.652±0.008**
Yacht Hydrodynamics	**1.064±0.072**	1.131±0.063

The bias term in linear layers acts as an additional input with mean one, variance zero, and slab probability one. As a result, the slab probability of all outputs is exactly 1, yielding no spike.

$$\rho_{\text{linear}} = 1 - \prod_{i=1}^{K} \left(1 - \rho_i^{(\ell)}\right) = 1$$

Additionally, we proved that applying a ReLU layer followed by a linear layer results in identical message distributions between PBP and SSPBP, yielding no difference between the approximations for a typical FFNN.

Modifying Architecture

We removed the bias term from both methods to allow for sparsity and different message distributions. Observed little or no difference performance in test-set RMSE or LL.

Dataset	PBP	SSPBP
Boston Housing	**3.809±0.295**	3.865±0.322
Combined Cycle Power Plant	4.190±0.017	**4.188±0.023**
Concrete Compression Strength	6.823±0.214	**6.668±0.237**
Energy Efficiency	1.699±0.041	**1.617±0.019**
Kin8nm	0.126±0.001	**0.125±0.001**†
Naval Propulsion	**0.005±0.000**	0.006±0.000
Wine Quality Red	0.635±0.011	**0.633±0.014**
Yacht Hydrodynamics	**3.898±0.245**	4.276±0.390

In practice, we observe that the slab probabilities remain close to one, even in narrow networks. Below, the mean and standard error of the average slab probability in the test set for Boston.

Hidden Layers	$\widehat{\rho}^{(1,\text{linear})}$	$\widehat{\rho}^{(2,\text{linear})}$	$\widehat{\rho}^{(3,\text{linear})}$	Output $\hat{y}$
5	1.000 ± 0.000	–	–	0.976 ± 0.007
50	1.000 ± 0.000	–	–	1.000 ± 0.000
5×5	1.000 ± 0.000	0.962 ± 0.007	–	$0.968 \pm 0.006^\dagger$
50×50	1.000 ± 0.000	1.000 ± 0.000	–	1.000 ± 0.000
$5 \times 5 \times 5$	1.000 ± 0.000	0.958 ± 0.008	$0.970 \pm 0.008^\dagger$	$0.971 \pm 0.009^\dagger$
$50 \times 50 \times 50$	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000

References

[1] Hernández-Lobato, José Miguel, and Ryan Adams. "Probabilistic backpropagation for scalable learning of bayesian neural networks." *International conference on machine learning*. PMLR, 2015.

[2] Gretton, Arthur, et al. "A kernel two-sample test." *The Journal of Machine Learning Research* 13.1 (2012): 723-773.