LATENT PROJECTION BNNS: AVOIDING WEIGHT-SPACE PATHOLOGIES BY LEARNING LATENT REPRESENTATIONS OF NEURAL NETWORK WEIGHTS

Melanie F. Pradier[1]
Weiwei Pan[1]
Jiayu Yao[1]
melanie@seas.harvard.edu

Soumya Ghosh[2]
Finale Doshi-Velez[1]

[1]Harvard University
[2]IBM Research

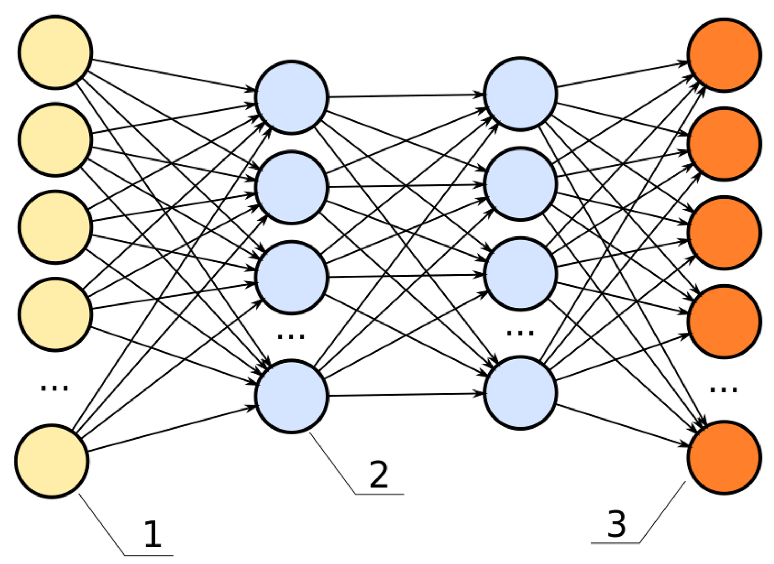
INTRODUCTION

- Deep neural networks provide high-predictive accuracy BUT:
  - tend to overfit with small number of samples
  - do not provide uncertainties
- Bayesian neural networks (BNN) provide uncertainty in network weights, but inference is hard:
  - high-dimensionality of network parameter space
  - correlation between these parameters
- **Our contribution**: novel inference framework for BNNs
  - a) we capture complex distributions in high-dim weight space via low-dimensional latent space
  - b) we do inference in low-dimensional representation
  - c) better estimation of uncertainty

BAYESIAN NEURAL NETWORK (BNN)

$$\mathbf{y} = f_{\boldsymbol{w}}(\mathbf{x}) + \epsilon$$
$$\boldsymbol{w} \sim p(\boldsymbol{w}),$$

where $\epsilon \sim \mathcal{N}(0, \sigma_y^2 \mathbf{I})$ and typically, $w_{ij} \sim \mathcal{N}\left(0, \sigma_w^2\right)$.



1. input layer
2. hidden nodes
3. output layer

$$p(\mathbf{y}^\star|\mathbf{x}^\star, \mathcal{D}) = \int p(\mathbf{y}^\star|\mathbf{x}^\star, \boldsymbol{w}) p(\boldsymbol{w}|\mathcal{D}) d\boldsymbol{w}$$

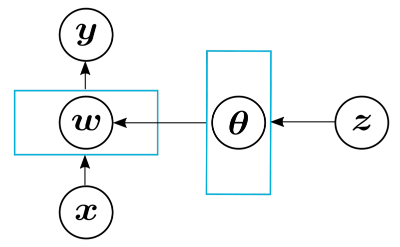
How can we approximate posterior distribution $p(\boldsymbol{w}|\mathcal{D})$ of weights $\boldsymbol{w}$?

VARIATIONAL INFERENCE

- Bayesian inference becomes an optimization problem
- Idea: choose variational distribution $q_\lambda(\boldsymbol{w})$ and minimize $D_{\mathrm{KL}}(q_\lambda(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D}))$.
- Minimizing KL divergence is equivalent to maximizing an upper bound $\mathcal{L}(\boldsymbol{\lambda})$ on the marginal likelihood of the data.

$$\begin{aligned} D_{\mathrm{KL}}(q_\lambda(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D})) &= q_\lambda(\boldsymbol{w})[\log q_\lambda(\boldsymbol{w}) - \log p(\boldsymbol{w}|\mathcal{D})] \\ &= -\mathcal{H}(q) - \mathbb{E}_q[\log p(\mathcal{D}, \boldsymbol{w})] + \log p(\mathcal{D}) \\ &= -\mathcal{L}(\lambda) + \log p(\mathcal{D}) \end{aligned}$$

LATENT PROJECTION BNN



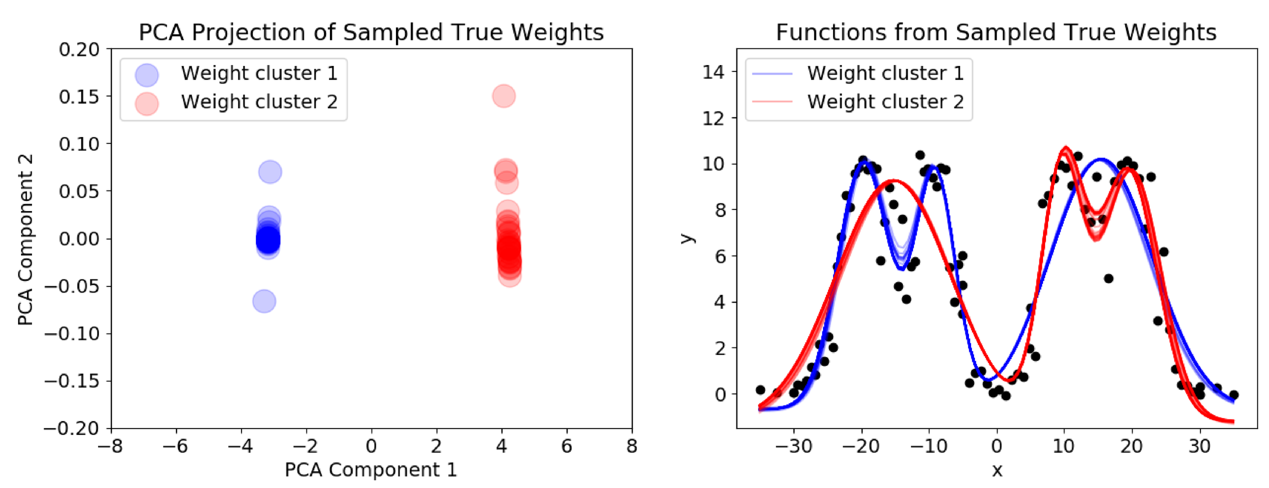
- $\boldsymbol{w}$: network weights
- $\boldsymbol{z}$: low-dimensional latent representation
- ϕ: params of non-linear projection $g_\phi(\cdot)$
- We choose the following variational distribution $q_\lambda(\boldsymbol{w})$:

$$\boldsymbol{z} \sim q_{\lambda_z}(\boldsymbol{z})$$
$$\phi \sim q_{\lambda_\phi}(\phi)$$
$$\boldsymbol{w} = g_\phi(\boldsymbol{z})$$

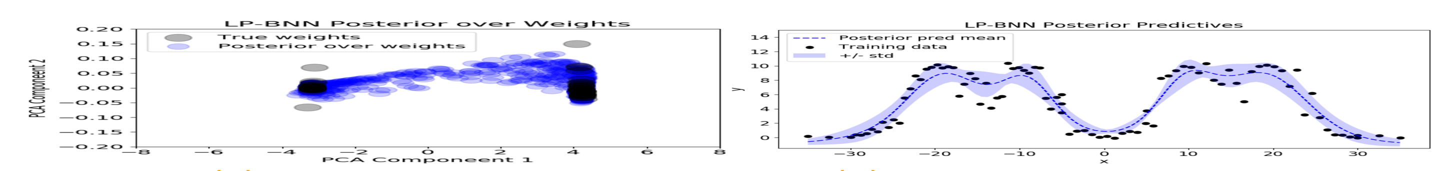
- Novel inference framework:
  1. **Characterize the space of plausible weights**: we train an ensemble of (non-Bayesian) neural networks from R multiple restarts $\rightarrow \mathbf{w_c}$
  2. **Learn projection parameters** ϕ: we train an autoencoder with a prediction-constrained loss function:
  $$\min_{\theta,\phi} \left\{ \frac{1}{R}\left(\mathbf{w_c} - \hat{\mathbf{w}}_\mathbf{c} + \gamma\right)^2 + \beta\, \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\log p(y|x, \hat{\mathbf{w}}_\mathbf{c})\right] \right\},$$
  where $f_\theta(\cdot)$ is the encoder, $g_\phi(\cdot)$ is the decoder, and $\hat{\mathbf{w}}_\mathbf{c} = g_\phi(f_\theta(\mathbf{w_c}))$.
  3. **Learn variational distribution** $q(\boldsymbol{z}, \phi)$: we perform BBVI with local reparametrization trick to maximize $\mathcal{L}(\boldsymbol{\lambda})$:

$$\mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_q\Big[\log p(\mathbf{y}|\mathbf{x}, g_\phi(\boldsymbol{z}))\Big] - D_{\mathrm{KL}}(q_{\lambda_z}(\boldsymbol{z})||p(\boldsymbol{z})) - D_{\mathrm{KL}}(q_{\lambda_\phi}(\phi)||p(\phi)).$$

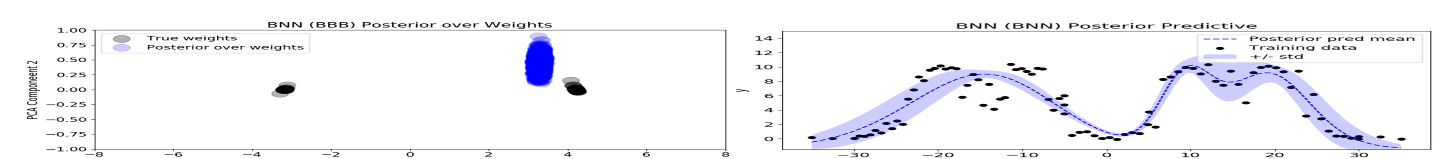
PROOF-OF-CONCEPT



Toy data for regression is generated from a function with four modes. (left) weight space of BNN with three hidden nodes. (right) examples of functions corresponding to weights sampled from each weight cluster.



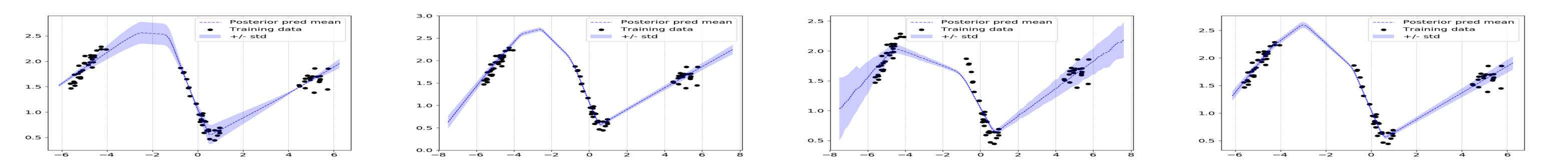
(a) LP-BNN Posterior (b) LP-BNN Posterior Predictive



(c) BNN (BBB) Posterior (d) BNN (BBB) Posterior Predictive

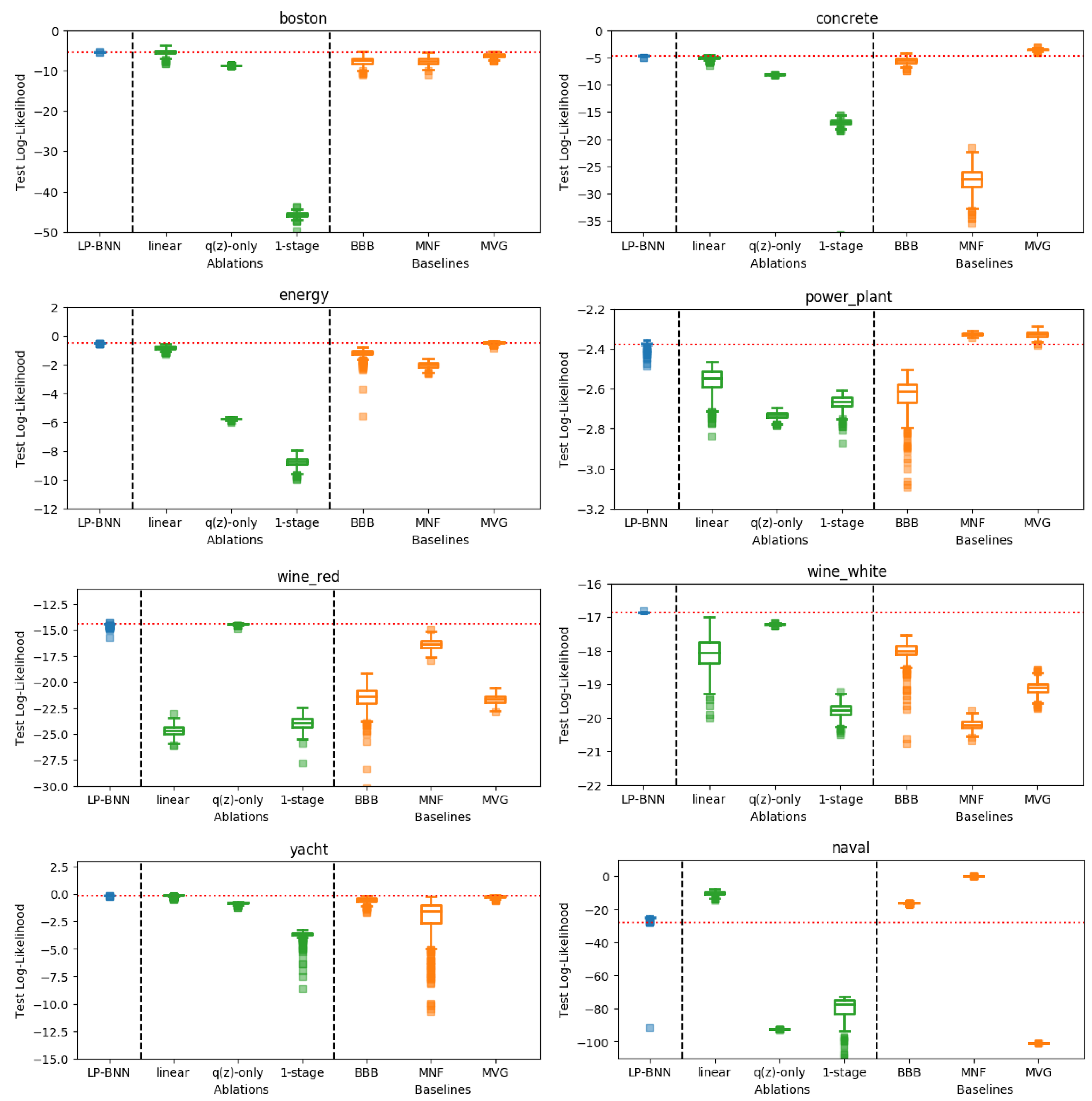
RESULTS

Baselines methods are: 1) BBB: mean field (Blundell, et.al 2015); 2) MNF: multiplicative normalizing flow (Louizos et.al, 2017); 3) MVG: multivariate Gaussian prior BNN (Louizos et.al, 2016).



(a) LP-BNN ($D_z = 2$) (b) BBB (c) MNF (d) MVG

Inferred predictive posterior distribution for a toy data set drawn from a NN with 1-hidden layer, 20 hidden nodes and RBF activation functions.



Test log-likelihood for UCI benchmark datasets for best dimensionality of z-space. Ablations of LP-BNN are: LP-BNN, LP-BNN with linear projections (linear), LP-BNN without training the autoencoder (1-stage), LP-BNN modeling uncertainty only in $\boldsymbol{z}$ ($q(z)$-only).

CONCLUSION

- LP-BNN provides better uncertainty estimations
- LP-BNN achieves performs better across datasets

HARVARD UNIVERSITY
HARVARD DATA SCIENCE INITIATIVE