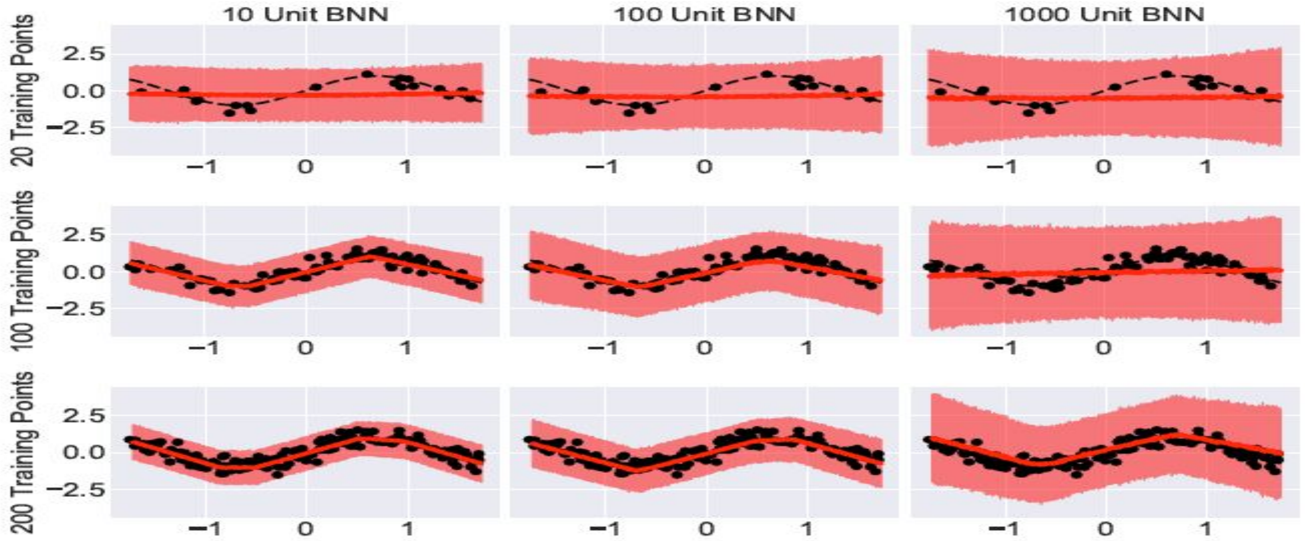
Model Selection in Bayesian Neural Networks via Horseshoe Priors

Soumya Ghosh, Jiayu Yao, Finale Doshi-Velez
Presented by Da Long 10/26/2021

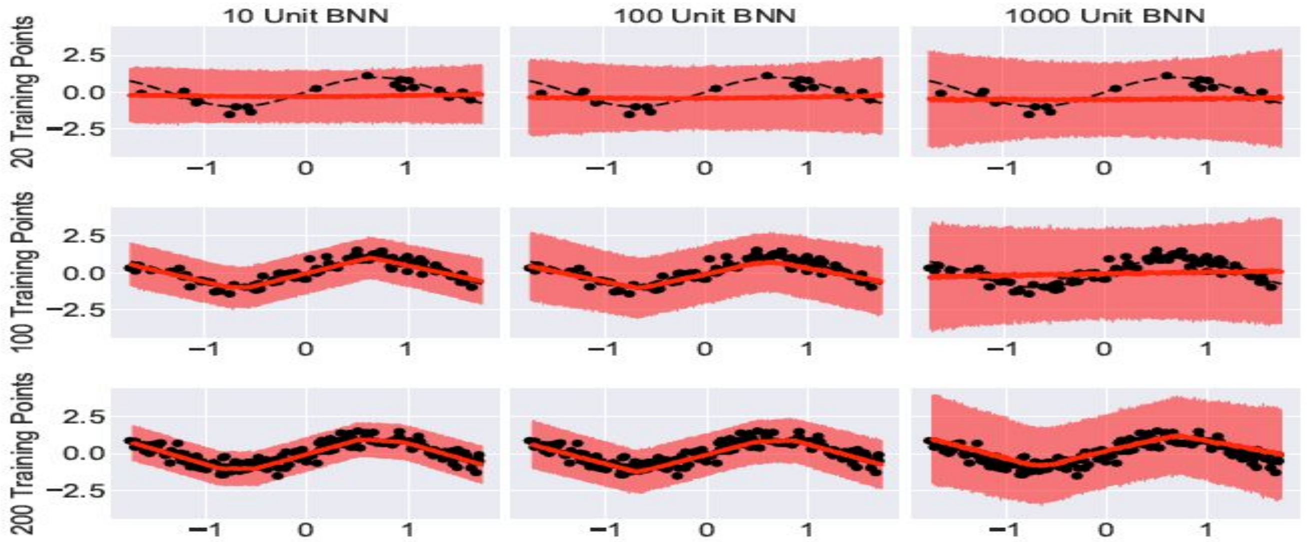
Motivation - BNN model selection

- Consequences for a poor choice of BNN architecture are severe
  - Too few nodes, BNN will not be flexible enough to model
  - Too many nodes, BNN predictions will have large variance
  - Example: A single layer with N(0,1) priors over weights: with fixed data, more nodes lead to large predictive variance



Motivation - BNN model selection

- Then how to select an appropriate BNN model that is expressive enough to explain observed data but not so expressive that it can explain everything?



Motivation - BNN model selection

- Perform onerous searches over different layer sizes - most straightforward but challenging
- This paper proposes a model selection in BNN by placing horseshoe and related priors over the variance of weights, at the same time, maintaining computational efficiency and statistical effectiveness
  - Deriving a sparse BNN model and pruning weights that don’t explain the data.
  - Getting a compact model that performs as well as the usual BNN or even better but has less predictive variance

Motivation - BNN model selection

- What is horseshoe prior?

Horseshoe prior is a scale mixture of normals

$$w_{kl} \mid \tau_{kl}, v_l \sim \mathcal{N}(0, (\tau_{kl}^2 v_l^2)\mathbb{I}), \quad \tau_{kl} \sim C^+(0, b_0), \quad v_l \sim C^+(0, b_g).$$

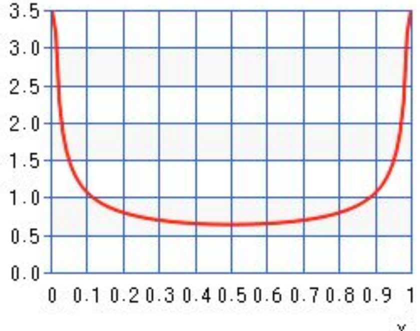
- Why it is named as “horseshoe”?

$\tau \sim C^+(0, a), a > 0,\ p(\tau) = \frac{2a}{\pi(\tau^2+a^2)}$

Define $k = \frac{1}{1+\tau^2},\ p(k) = p(\tau(k))|\tau'| = \frac{1}{\pi}k^{-\frac{1}{2}}(1-k)^{-\frac{1}{2}}, x \in (0,1)$

Here k can be seen as a shrinkage weight. k→0,implying $\boldsymbol{\tau}$ is large, no shrinkage. k→1, implying $\boldsymbol{\tau}$ is near 0, strong shrinkage.

The shape of p(k) looks like a horseshoe :



Alternatives

Alternative approaches to sparse networks.

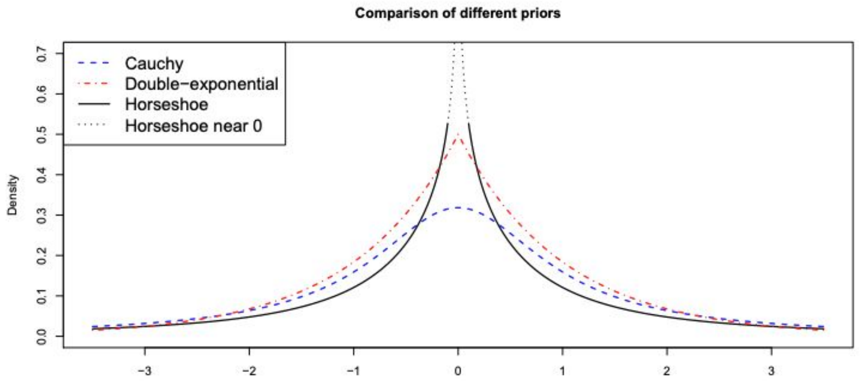
- Regularization:
  - L1(Ridge) and L2(Lasso) regularization, uniform shrinkage

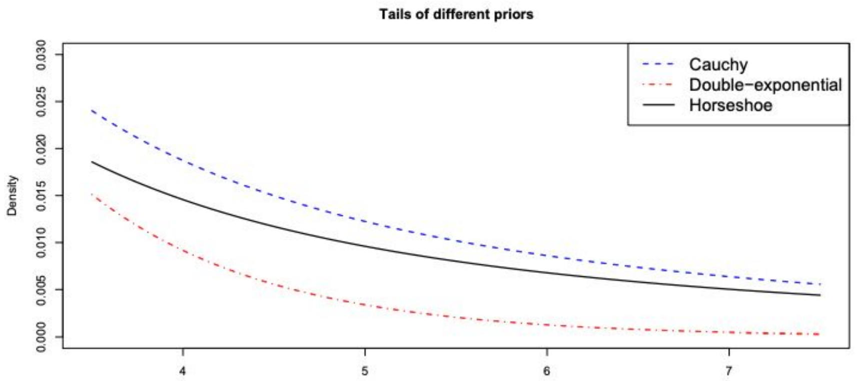
$$\tilde{J}(w; X, y) = J(w; X, y) + \alpha\Omega(w)$$

- Spike and slab (placing certain priors over weights)
  - Examples: Laplace distribution(double-exponential, Equivalent to L1 regularization) and Cauchy distribution

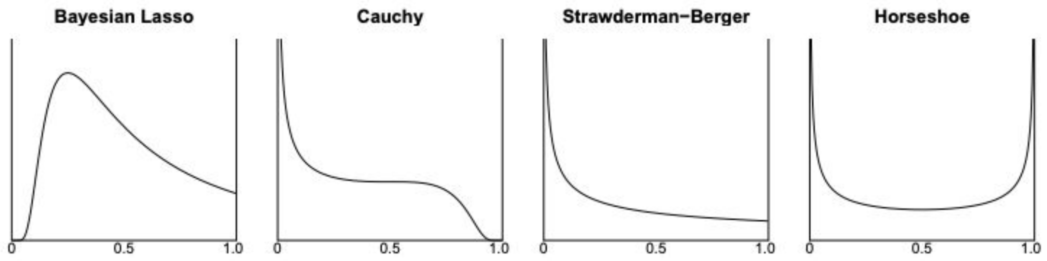
Alternatives

- Comparison of priors





- Comparison of shrinkage weight k



BNN with Structured Sparsity

- Horseshoe priors

$$w_{kl} \mid \tau_{kl}, \upsilon_l \sim \mathcal{N}(0, (\tau_{kl}^2 \upsilon_l^2)\mathbb{I}), \quad \tau_{kl} \sim C^+(0, b_0), \quad \upsilon_l \sim C^+(0, b_g).$$

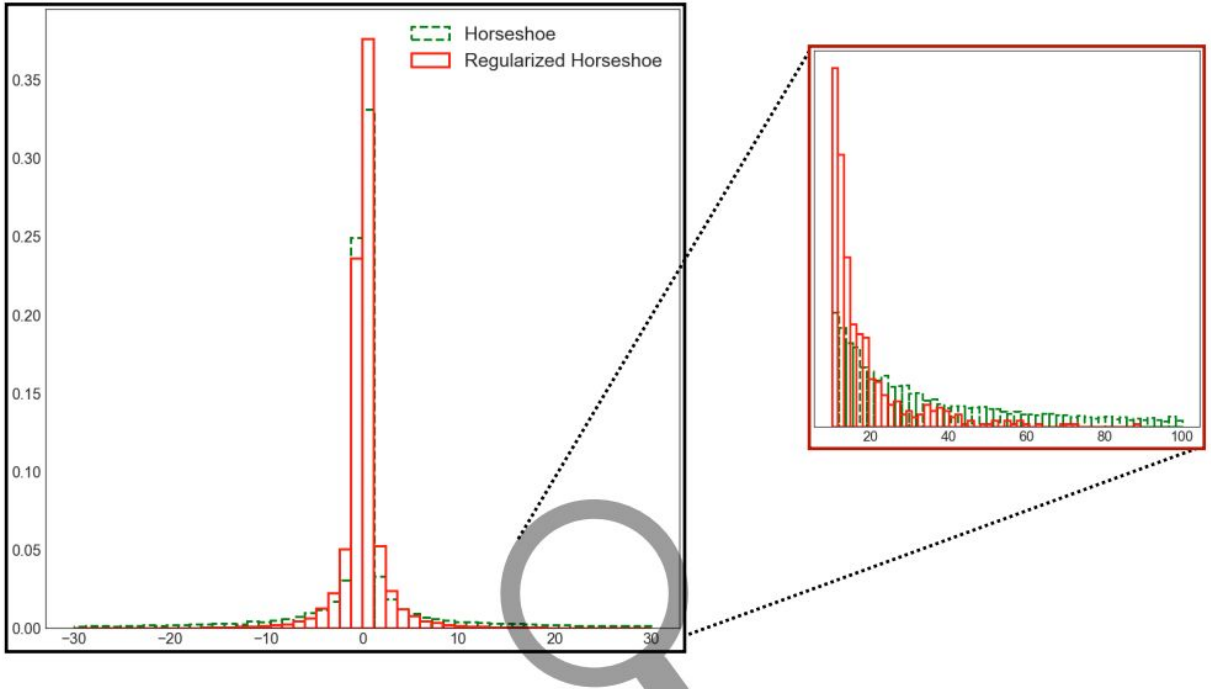
- Regularized Horseshoe priors

To solve the problem that if some units escape shrinkage, the weights become very large, which affects generalization performance

$$w_{kl} \mid \tau_{kl}, \upsilon_l, c \sim \mathcal{N}(0, (\tilde{\tau}_{kl}^2 \upsilon_l^2)\mathbb{I}), \quad \tilde{\tau}_{kl}^2 = \frac{c^2 \tau_{kl}^2}{c^2 + \tau_{kl}^2 \upsilon_l^2}.$$

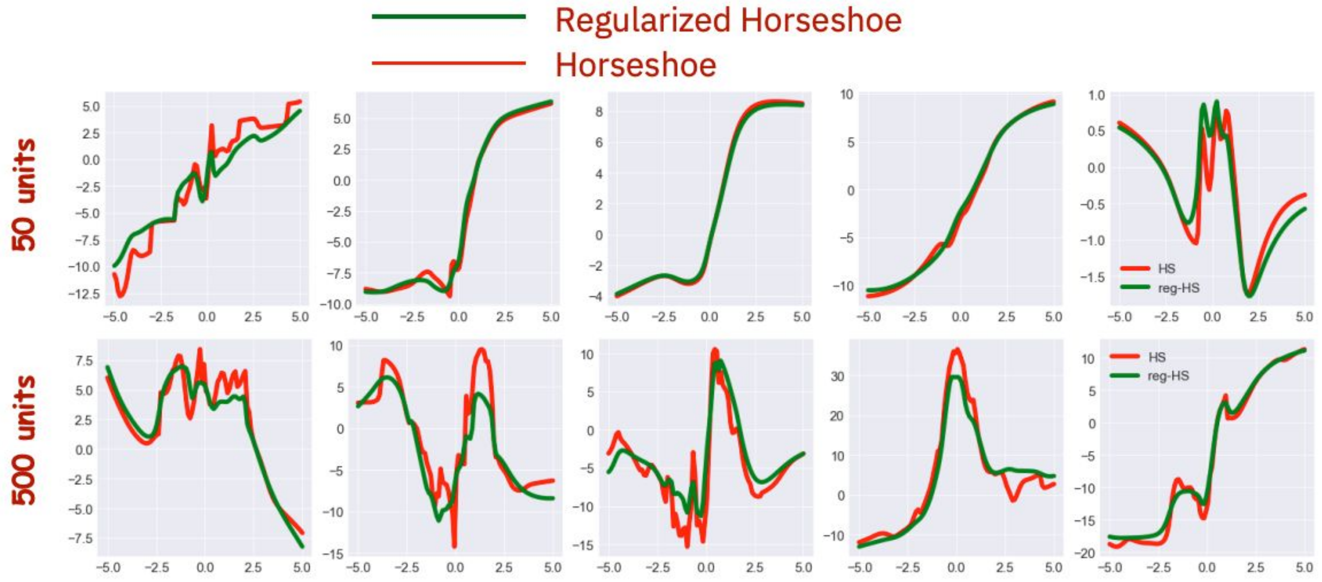
$$\tau^2 \upsilon^2 \ll c^2, w \sim N(0, \tau^2 \upsilon^2 I), \tau^2 \upsilon^2 \gg c^2, w \sim N(0, c^2 I)$$

Regularized Horseshoe



Regularized Horseshoe

$$w_{kl} \mid \tau_{kl}, \upsilon_l, c \sim \mathcal{N}(0, (\tilde{\tau}_{kl}^2 \upsilon_l^2)\mathbb{I}), \quad \frac{1}{\tilde{\tau}_{kl}^2 \upsilon_l^2} = \frac{1}{c^2} + \frac{1}{\tau_{kl}^2 \upsilon_l^2}$$



BNN with Structured Sparsity

- Half-Cauchy re-parameterization for variational learning

Direct parameterization of the Half-Cauchy distribution is possible, however, during variational learning, standard exponential family variational approximations are hard to capture the thick Cauchy tails, and Cauchy approximating family leads to high variance gradients

- Adopt the auxiliary variable parameterization:

$$a \sim C^{+}(0, b) \Longleftrightarrow a^2 \mid \lambda \sim \text{Inv-Gamma}(\frac{1}{2}, \frac{1}{\lambda}); \lambda \sim \text{Inv-Gamma}(\frac{1}{2}, \frac{1}{b^2})$$

BNN with Structured Sparsity

- Non-Centered Parameterization

  Before non-centered parameterization: $w_{kl} \mid \tau_{kl}, \upsilon_l, c \sim \mathcal{N}(0, (\tilde{\tau}_{kl}^2 \upsilon_l^2)\mathbb{I}), \quad \tilde{\tau}_{kl}^2 = \frac{c^2 \tau_{kl}^2}{c^2 + \tau_{kl}^2 \upsilon_l^2}.$

  After non-centered parameterization: $\beta_{kl} \sim \mathcal{N}(0, \mathbb{I}), \quad w_{kl} = \tilde{\tau}_{kl} \upsilon_l \beta_{kl},$

- Benefits:
  - Lead to simpler posterior geometries, which are easy to approximate or reliably sample

BNN with Structured Sparsity

- The joint distribution of the horseshoe BNN

$$p_{\text{regHS}}(\mathcal{D},\theta) = p(c \mid c_a, c_b) r(\kappa, \rho_\kappa \mid b_\kappa) \prod_{k=1}^{K_L} \mathcal{N}(w_{kL} \mid 0, \kappa^2 \mathbb{I}) \prod_{l=1}^{L} \Bigg\{ r(\upsilon_l, \vartheta_l \mid b_g) \prod_{k=1}^{K_l} r(\tau_{kl}, \lambda_{kl} \mid b_0) \mathcal{N}(w_{kl} \mid 0, (\tilde{\tau}_{kl}^2 \upsilon_l^2)\mathbb{I}) \Bigg\} \prod_{n=1}^{N} p(y_n \mid f(\mathcal{W}, x_n)),$$

with $\theta = \{\mathcal{W}, \mathcal{T}, \kappa^2, \rho_\kappa, c^2\}$, $\mathcal{T} = \{\{\tau_{kl}^2\}_{k=1,l=1}^{K,L}, \{\upsilon_l^2\}_{l=1}^{L}, \{\lambda_{kl}\}_{k=1,l=1}^{K,L}, \{\vartheta_l\}_{l=1}^{L}\}$.

where $r(a, \lambda | b) = \text{Inv-Gamma}(a^2 | 1/2, 1/\lambda)\text{Inv-Gamma}(\lambda | 1/2, 1/b^2)$

Variational Inference for Model Learning

- The choice of the approximating family governs the quality of inference
- Three kinds of approximations are proposed:
  - Fully factorized approximation
    - Simplest, failing to capture correlations between weights
  - Semi-structured approximation
    - Capturing layer-wise weight correlations
  - Structured approximation
    - More sophisticated, better performance
    - Capturing weight-scale correlations

Variational Inference for Model Learning

- Factorized Approximation

$$q(\theta \mid \phi) = \prod_{a \in \{c, \kappa, \rho_\kappa\}} q(a \mid \phi_a) \prod_{i,j,l} q(\beta_{ij,l} \mid \phi_{\beta_{ij,l}}) \prod_{k,l} q(\tau_{kl}^2 \mid \phi_{\tau_{kl}}) q(\lambda_{kl} \mid \phi_{\lambda_{kl}}) \prod_{l} q(\upsilon_l^2 \mid \phi_{\upsilon_l}) q(\vartheta_l \mid \phi_{\vartheta_l}).$$

$$r(a, \lambda | b) = \text{Inv-Gamma}(a^2 | 1/2, 1/\lambda)\text{Inv-Gamma}(\lambda | 1/2, 1/b^2)$$

$$\mathcal{N}(\beta_{ij,l} \mid \mu_{ij,l}, \sigma_{ij,l}^2) \quad \text{or } N(\beta_{ij,l} | u_{ij,l}, 1)$$

$$q(\ln \tau_{kl}^2 \mid \phi_{\tau_{kl}}) = \mathcal{N}(\mu_{\tau_{kl}}, \sigma_{\tau_{kl}}^2)$$

$$q(\ln \upsilon_l^2 \mid \phi_{\upsilon_l}) = \mathcal{N}(\mu_{\upsilon_l}, \sigma_{\upsilon_l}^2)$$

$$q(ln(k^2) | \phi_k) = N(u_k, \sigma_k^2)$$

$$\vartheta, \lambda, \rho = Inverse - Gamma(a, b)$$

Variational Inference for Model Learning

- Semi-structured approximation
  - To capture layer-wise weight correlations
  - Using a layer-wise matrix normal distribution for non-centered weights
  - Keep the other factors from fully factorized posterior the same

$$q(\beta_l|\phi_{\beta_l}) = MN(\beta_l|M_l, U_l, V_l)$$

U and V are restricted to be diagonal

- The matrix normal approximation can lead to significant improvements over factorized approximations

Variational Inference for Model Learning

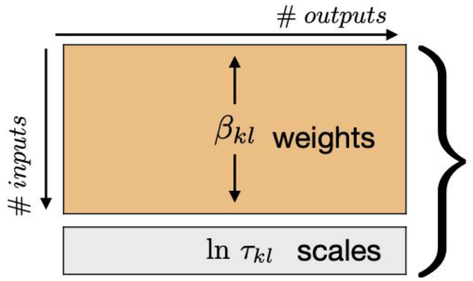
- Structured approximation
  - To capture weight-scale correlations
  - Using a matrix normal distribution containing non-centered weights and unit specific scales
  - Keep the other factors from fully factorized posterior the same

$$B_l = \begin{bmatrix} \beta_l \\ \nu_l^T \end{bmatrix}, \; \nu_l = [\nu_{1l}, \ldots, \nu_{K_l l}]^T, \text{ and } \nu_{kl} = \ln \tau_{kl}^2$$

$$q(B_l \mid \phi_{B_l}) = \mathcal{MN}(B_l \mid M_l, U_l, V_l)$$

To retain computational efficiency while capturing dependencies, enforce a low-rank structure:

$U = \Psi + h_l h_l^T$ where Ψ is diagonal and h is a vector. V stays the same, diagonal



$$q(B) = \text{Matrix-Normal}(M, U, V)$$

$$U = hh' + \Psi$$

Variational Inference for Model Learning

- ELBO

$$\mathcal{L}(\phi) = \sum_n \mathbb{E}_{q(\theta|\phi)}[\ln p(y_n \mid f(\theta, x_n))] + \mathbb{E}_{q(\theta|\phi)}[\ln p(\theta \mid b_0, b_g, b_\kappa, c_a, c_b)] + \mathbb{H}[q(\theta \mid \phi)]$$

- Black Box Inference
  - Use unbiased Monte-Carlo method and reparameterization gradients to differentiate through the sampling process

$$\zeta \sim \mathcal{N}(\mu, \sigma^2) \Leftrightarrow \epsilon \sim \mathcal{N}(0, 1), \zeta = \mu + \sigma\epsilon$$

- What about posterior for Inverse-Gamma?

$$q(\lambda_{kl} \mid \phi_{\lambda_{kl}}) = \text{Inv-Gamma}(\lambda_{kl} \mid 1, \mathbb{E}[\frac{1}{\tau_{kl}^2}] + \frac{1}{b_0^2}).$$

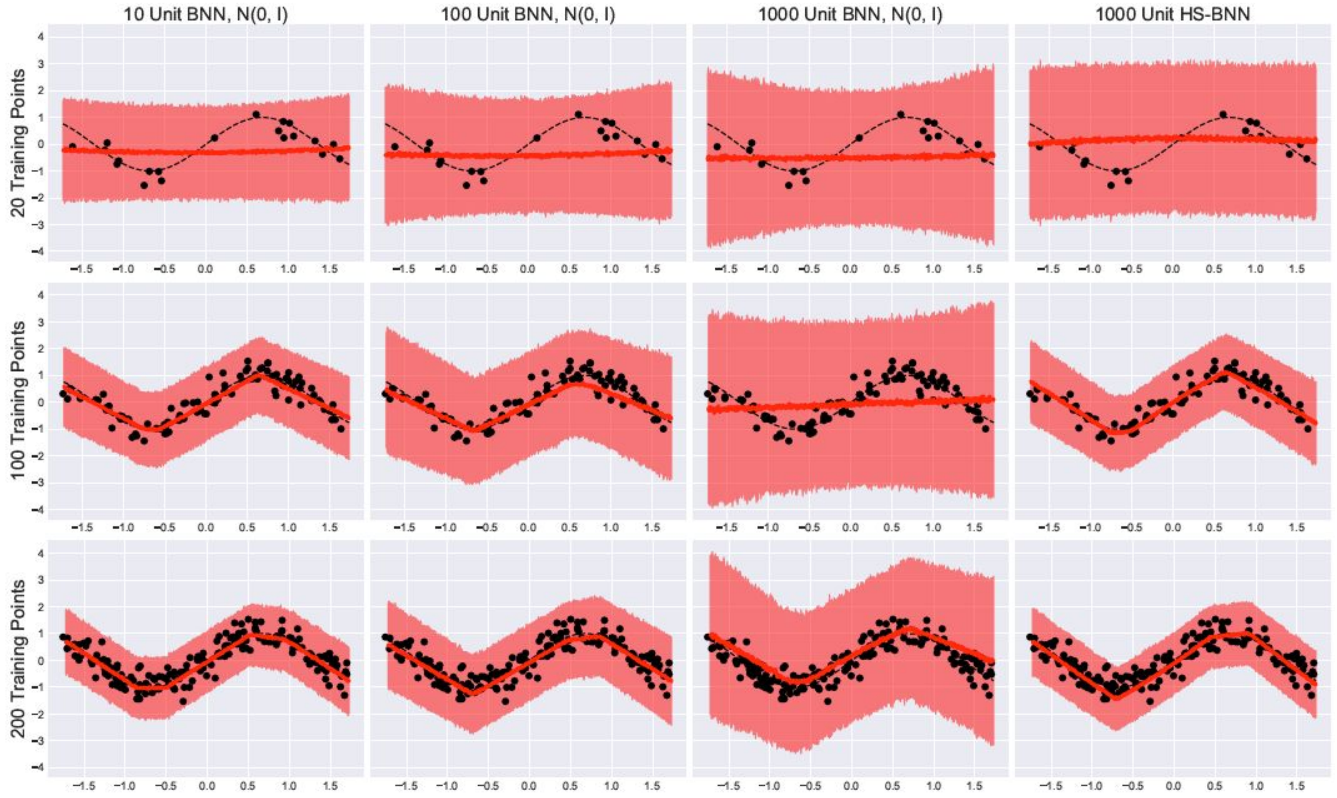
The formula is the the samefor ϑ, ρ

Pruning

- Set a thresholding to prune away the shrunk weights

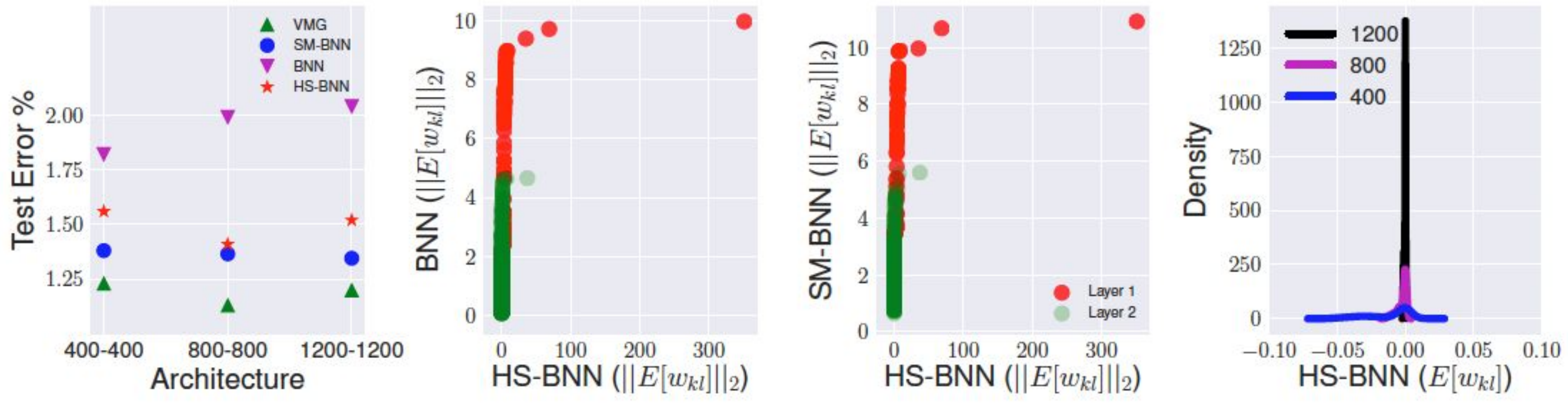
  Prune away a unit if $p(\tau_{kl}\upsilon_l < \delta) > p_0$

Experiments



Experiments

MNIST Results



Experiments

HSBNN

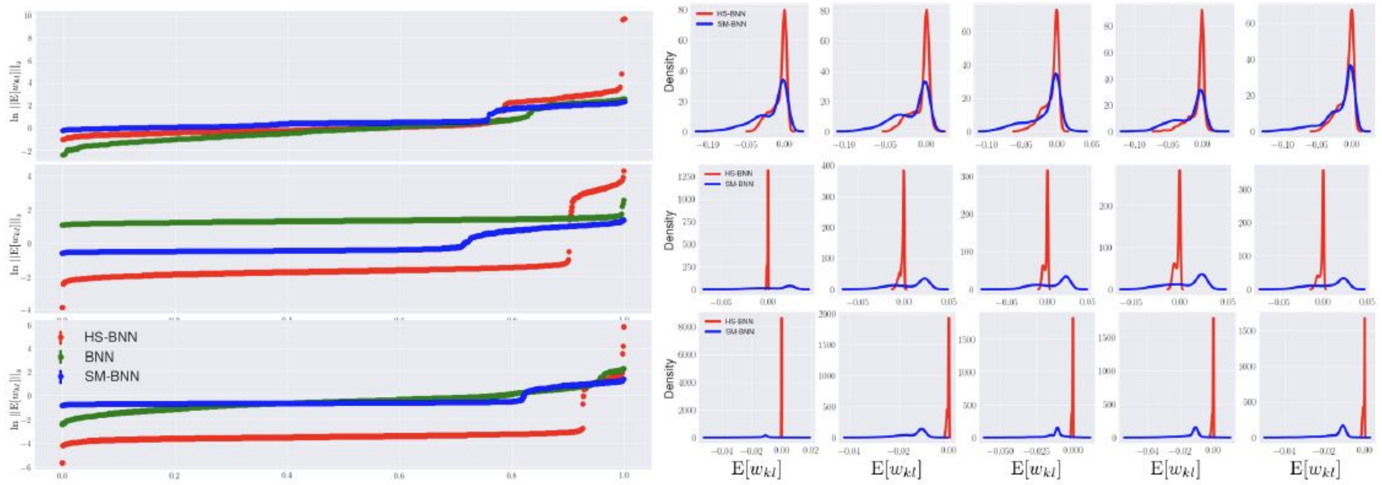


Figure 9: **Model selection in HS-BNN.** *Left*: From top to bottom, we plot log $||\mathbb{E}[w_{kl}]||_2$ for the first layer of a network with 400, 800 and 1200 units. The x-axis has been normalized by the number of units. Further exploration of sparse solutions found by HS-BNN. *Right*: Here we provide density plots for the five smallest (across all layers) node weight vectors w_{kl} found by HS-BNN and SM-BNN for the 400-400 (top row), 800-800 (middle row), 1200-1200 (bottom row) network. Plots are sorted by 2-norm of w_{kl}, from left to right.

Experiments

HSBNN

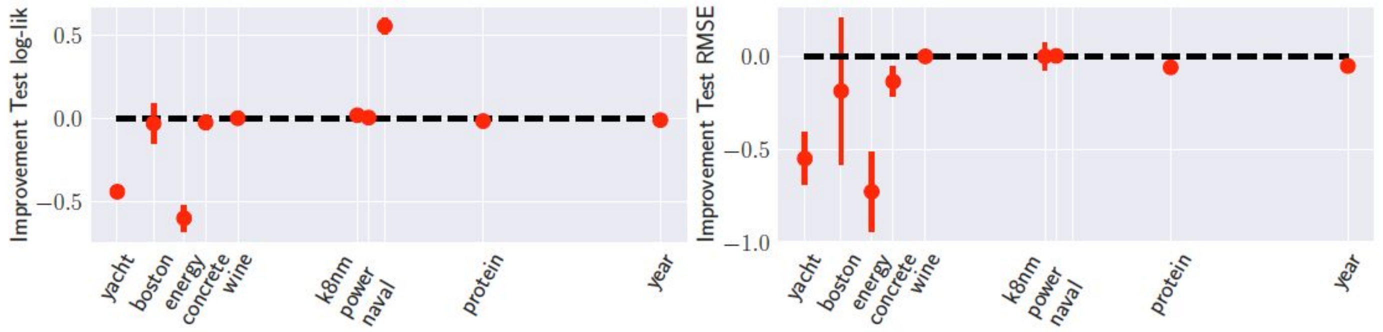


Figure 11: **Regression performance** on a collection of data sets from the UCI repository. Data sets are arranged from left to right in order of increasing number of data points and plotted on the log scale. Plots display relative improvements in test log likelihood and root mean squared error (± 1 standard deviation) averaged over 20 random splits of the data. Relative improvement is defined as $(x - y)/max(|x|, |y|)$. VMG performs better on smaller data sets and comparably on larger ones.

Experiments

Regularized HSBNN

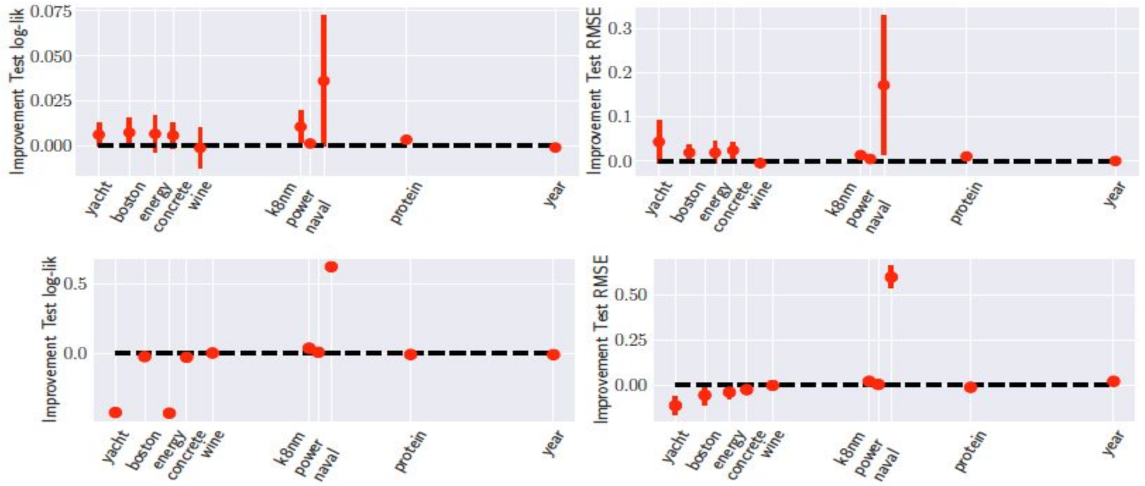


Figure 12: **Regularized horseshoe** results in consistent improvements (TOP) over the vanilla horseshoe prior (values over 0 indicate improvements). The data sets are sorted according to the number of data instances and plotted on the log scale, with 'yacht' being the smallest and 'year' being the largest. Relative improvement is defined as $(x-y)/max(|x|,|y|)$. (BOTTOM) Regularized horseshoe performs similarly to VMG, better on some data sets and worse on others.

Experiments

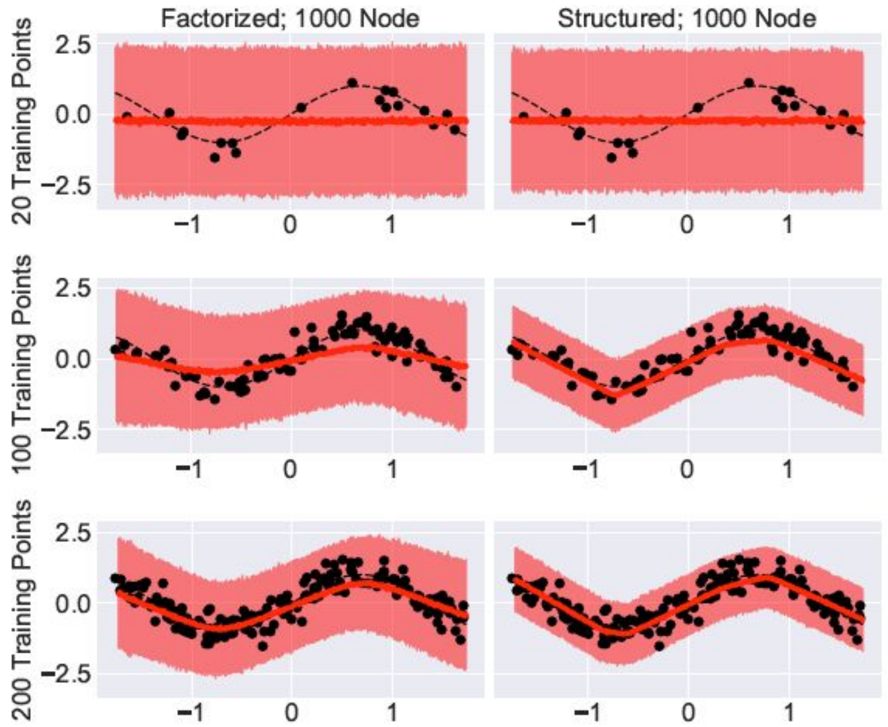
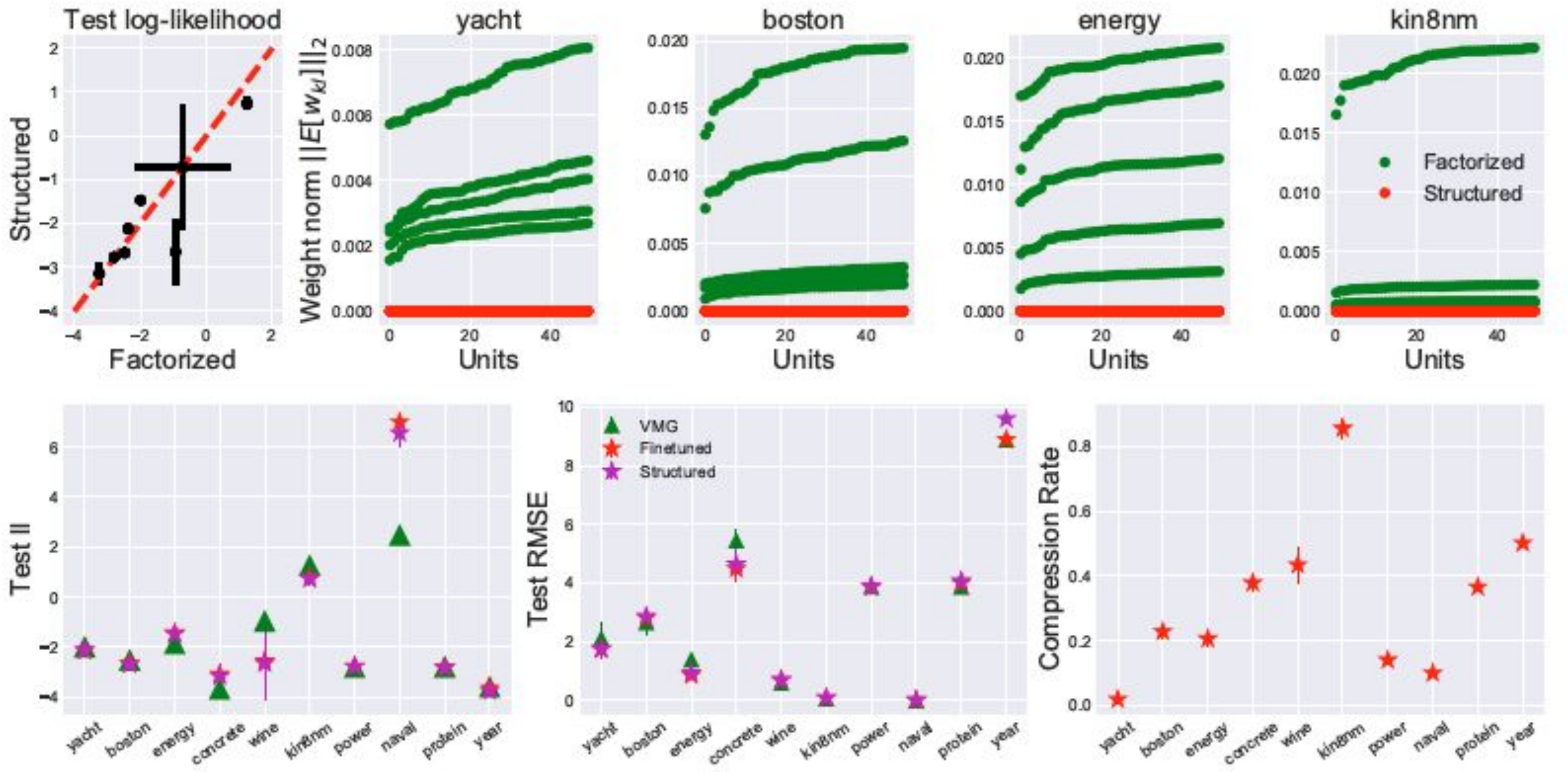


Figure 14: **Regularized horseshoe BNNs** prune away excess capacity and are more resistant to underfitting. Variational approximations aware of model structure improve fits.

Experiments


Test log-likelihood
Structured
Factorized
yacht
boston
energy
kin8nm
Weight norm $||E[w_{kl}]||_2$
Units
● Factorized
● Structured
Test ll
Test RMSE
Compression Rate
VMG
Finetuned
Structured
yacht
boston
energy
concrete
wine
kin8nm
power
naval
protein
year