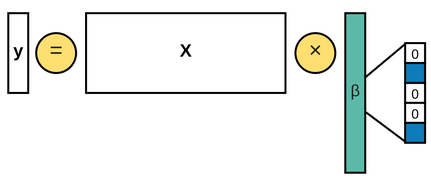
Uncertainty propagation in neural networks for sparse coding

Danil Kuzin*, Olga Isupova†, Lyudmila Mihaylova*

*University of Sheffield, UK †University of Oxford, UK

1. LISTA

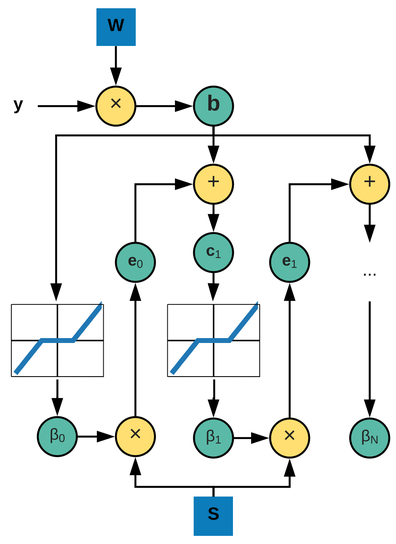
Estimate $\boldsymbol{\beta}$ from observations $\mathbf{y}$ collected as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, s.t. elements $\boldsymbol{\beta}$ contain zeros.



LISTA [G&L]

- Represent iterative soft-thresholding algorithm as RNN with shared weights
- Learn weights with BPTT

Init. Dense $\mathbf{b} \leftarrow \mathbf{W}\mathbf{y}$
Init. Soft-thresholding $\widehat{\boldsymbol{\beta}}_0 \leftarrow h_\lambda(\mathbf{b})$
for $l = 1$ **to** L **do**
  Dense $\mathbf{c}_l \leftarrow \mathbf{b} + \mathbf{S}\widehat{\boldsymbol{\beta}}_{l-1}$
  Soft-thresholding $\widehat{\boldsymbol{\beta}}_l \leftarrow h_\lambda(\mathbf{c}_l)$
end for
return $\widehat{\boldsymbol{\beta}} \leftarrow \widehat{\boldsymbol{\beta}}_L$



Overfitting

No uncertainty estimation

2. BayesLISTA

- Add priors for NN weights

$$p(\mathbf{W}) = \prod_{d=1}^{D}\prod_{k=1}^{K}\mathcal{N}(w_{ij}; 0, \eta^{-1}), \quad p(\mathbf{S}) = \prod_{d'=1}^{D}\prod_{d''=1}^{D}\mathcal{N}(s_{d'd''}; 0, \eta^{-1}),$$

- Propagate distribution for $\widehat{\boldsymbol{\beta}}$ through layers
- Compute prediction as noisy NN output

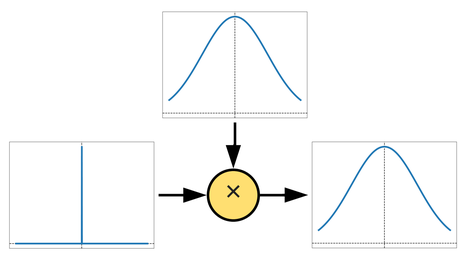
$$p(\boldsymbol{\beta}|\mathbf{y}, \mathbf{W}, \mathbf{S}, \gamma, \lambda) = \prod_{d=1}^{D}\mathcal{N}\left(\beta_d; [f(\mathbf{y}; \mathbf{S}, \mathbf{W}, \lambda)]_d, \gamma^{-1}\right)$$

- Update weights with PBP

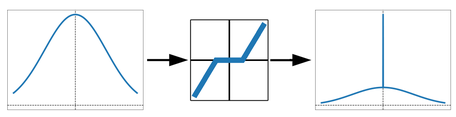
3. Uncertainty propagation

At every step the output of soft-thresholding can be closely approximated with the spike and slab distribution

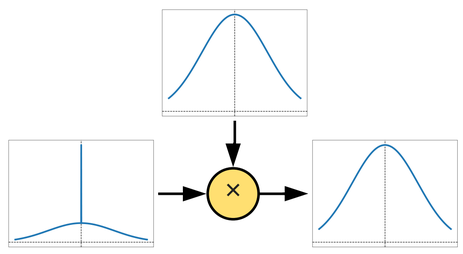
1. $\mathbf{b} = \mathbf{W}\mathbf{y}$ is Gaussian-distributed



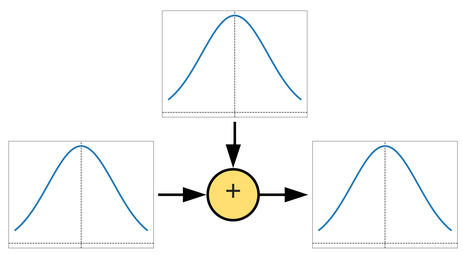
2. $\widehat{\boldsymbol{\beta}}_0 = h_\lambda(\mathbf{b})$ is approximated with the spike and slab distribution



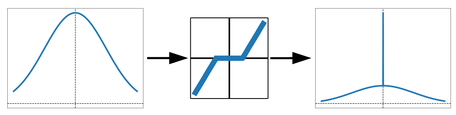
3. $\mathbf{e}_l = \mathbf{S}\widehat{\boldsymbol{\beta}}_{l-1}$ is approximated with the Gaussian distribution



4. $\mathbf{c}_l = \mathbf{b} + \mathbf{e}_l$ is Gaussian-distributed



5. $\widehat{\boldsymbol{\beta}}_l = h_\lambda(\mathbf{c}_l)$ is approximated with the spike and slab distribution



All latent variables are modelled with parametrised distributions

We can apply approximate Bayesian inference methods

4. BackProp-PBP

Approximate posterior

$$q(\mathbf{W}, \mathbf{S}, \gamma, \eta) = \prod_{d=1}^{D}\prod_{k=1}^{K}\mathcal{N}(w_{dk}; m^w_{dk}, v^w_{dk})\prod_{d'=1}^{D}\prod_{d''=1}^{D}\mathcal{N}(s_{d'd''}; m^s_{d'd''}, v^s_{d'd''}) \times \mathrm{Gam}(\gamma; a^\gamma, b^\gamma)\mathrm{Gam}(\eta; a^\eta, b^\eta)$$

Probabilistic backpropagation [HL&A]: use derivatives of the logarithm of a normalisation constant to update weight distributions

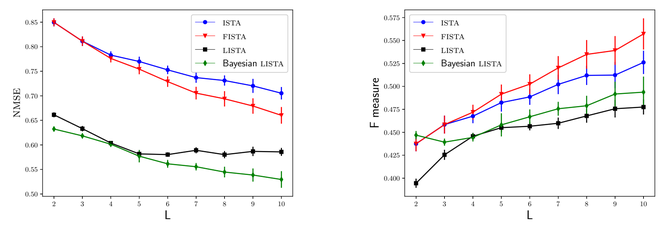
$$q(a) = Z^{-1}f(a)\mathcal{N}(a; m, v)$$

$$Z \approx \prod_{d=1}^{D}\left[\omega_d^{\widehat{\boldsymbol{\beta}}}\mathcal{T}\left(\beta_d; 0, \beta^\gamma/\alpha^\gamma, 2\alpha^\gamma\right) + \left(1 - \omega_d^{\widehat{\boldsymbol{\beta}}}\right)\mathcal{N}\left(\beta_d; m_d^{\widehat{\boldsymbol{\beta}}}, \beta^\gamma/(\alpha^\gamma - 1) + v_d^{\widehat{\boldsymbol{\beta}}}\right)\right],$$

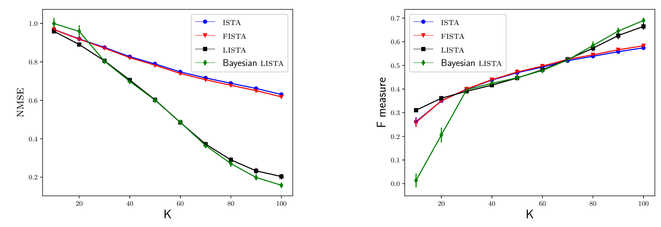
where $\{\omega_d^{\widehat{\boldsymbol{\beta}}}, m_d^{\widehat{\boldsymbol{\beta}}}, v_d^{\widehat{\boldsymbol{\beta}}}\}$ are the parameters of the spike and slab distribution for $[\widehat{\boldsymbol{\beta}}]_d$.

5. Results

Synthetic

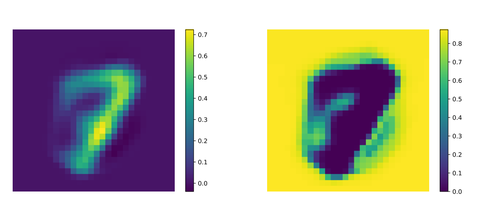


Different depth performance



Different observation size performance

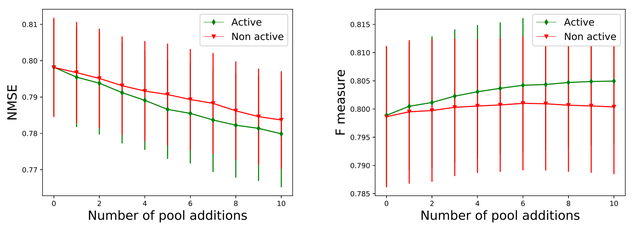
MNIST



Posterior mean and spike indicator for an image of digit 7

Samples from the posterior for an image of digit 7

Active Learning Use the estimated uncertainty to choose next training data with largest variance



Sequential pool additions

6. References

[HL&A] J. M. Hernández-Lobato, R. Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. ICML 2015.

[G&L] K. Gregor, Y. LeCun. Learning fast approximations of sparse coding. ICML 2010.