

Bayesian Neural Networks

ROY team: Ilya Zharikov,
Roman Isachenko,
Artem Bochkarev

Skolkovo Institute of Science and Technology
Bayesian Methods course

May 25, 2017

Project goal

Goal

Estimate posterior distributions of the model parameters from data

Problem

Monte Carlo sampling is very slow for high-dimensional data

Probabilistic Programming:

- Uncertainty in predictions;
- Uncertainty in representations;
- Regularizations with priors;
- Transfer learning;
- Hierarchical Neural Networks.

- 1 Salvatier J, Wiecki T. V., Fonnesbeck C. Probabilistic programming in Python using PyMC3. // *PeerJ Computer Science*. 2016.
- 2 Blundell C. et al. Weight Uncertainty in Neural Network // *Proceedings of The 32nd International Conference on Machine Learning*. 2015.
- 3 Kucukelbir A. et al. Automatic Differentiation Variational Inference // *arXiv preprint arXiv:1603.00788*. – 2017.

Problem Statement

Inference problem

Bayes' theorem states: $\mathbb{P}(\boldsymbol{\theta} | \mathbf{X}) = \frac{\mathbb{P}(\mathbf{X} | \boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta})}{\mathbb{P}(\mathbf{X})}$

Maximum A Posteriori (MAP) estimation

$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta}} [\ln \mathbb{P}(\boldsymbol{\theta} | \mathbf{X})] = \arg \max_{\boldsymbol{\theta}} [\ln \mathbb{P}(\mathbf{X} | \boldsymbol{\theta}) + \ln \mathbb{P}(\boldsymbol{\theta})]$$

Monte Carlo approach:

- Metropolis-Hastings sampling;
- Gibbs sampling;
- No-U-Turn Sampling (NUTS).

Goal

Approximate posterior distribution $p(\mathbf{X}, \theta)$ by function $q(\theta)$ from parametric family.

$$\begin{array}{ccc} \ln p(\mathbf{X}) = \text{KL}(q||p) + \text{ELBO}(q) & & \\ \updownarrow & & \updownarrow \\ \int q(\theta) \ln \frac{q(\theta)}{p(\theta|\mathbf{X})} d\theta & & \int q(\theta) \ln \frac{p(\mathbf{X}, \theta)}{q(\theta)} d\theta \end{array}$$

Minimization of **KL(q||p)** \Leftrightarrow Maximization of **ELBO(q)**

Automatic Differentiation Variational Inference (ADVI)

- Automatic transformation of constrained variables $\zeta = T(\theta)$;
Example: $\theta \in \mathbb{R}_+ \Rightarrow \zeta = T(\theta) = \log \theta$, then $\zeta \in \mathbb{R}$.
- $q(\zeta) = \mathcal{N}(\mu, \Sigma)$, where Σ is diagonal;

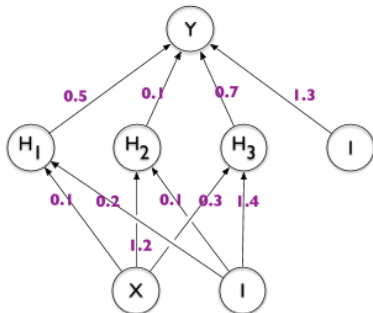
$$\mu^*, \Sigma^* = \arg \max_{\mu, \Sigma} \text{ELBO}(q)$$

- Stochastic optimization;
- Reparametrization trick to apply automatic differentiation;
- Adaptive step-size.

Difference

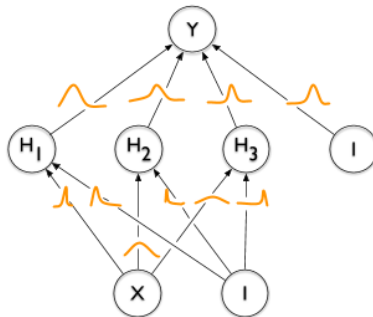
Neural Networks

Predict values of parameters by fitting complex model on the huge dataset



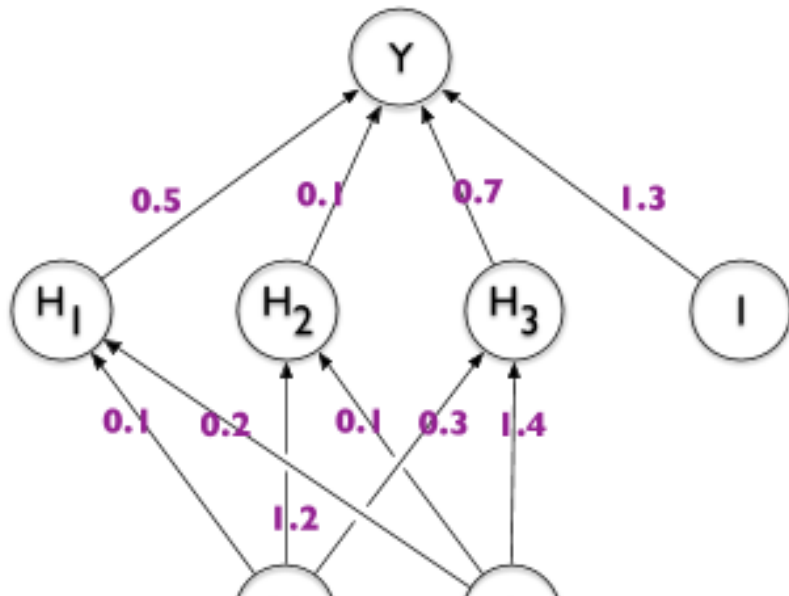
Bayesian Neural Networks

Predict the parameters of the weights distributions from the dataset



<http://bit.ly/2rMQuDq>

Deep Learning



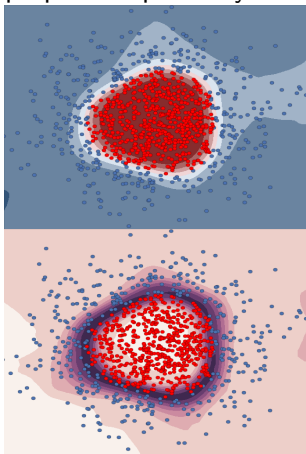
- Synthetic datasets: moons, circles
- Real data: MNIST

Goals:

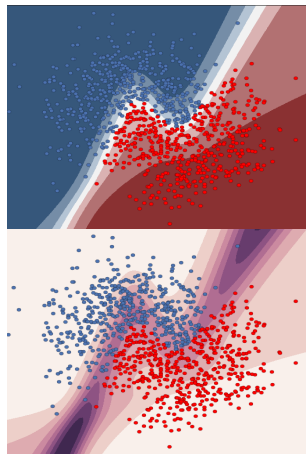
- investigate influence of different priors on the predictions
- visualize uncertainties in predictions
- analyze the model behaviour

Experiments

The best priors for circles and moons are gauss and cauchy with hyperprior respectively.



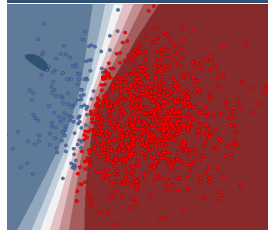
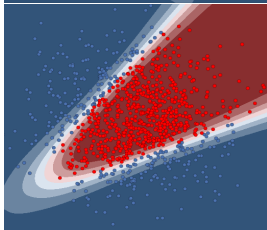
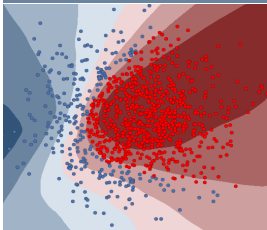
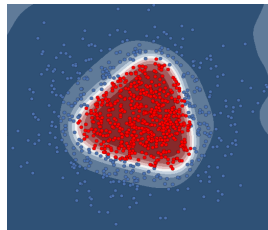
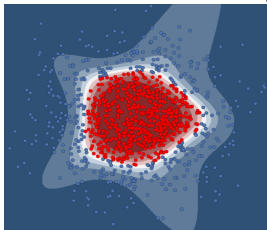
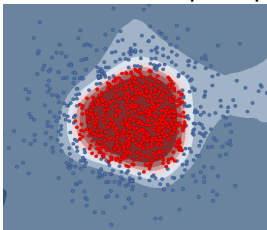
Accuracy: 0.735



Accuracy: 0.851

Experiments

Including hyperpriors is almost always the best choice if we are unsure about prior parameters or data is noisy and hard.



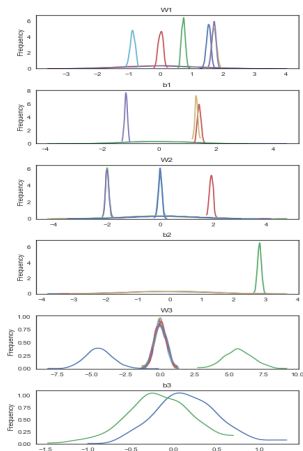
Cauchy prior

Normal prior

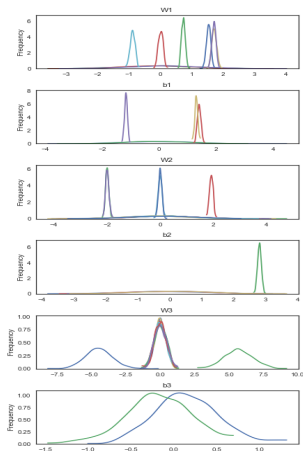
Laplace prior

Experiments

Laplace prior does indeed provide sparser solutions compared to cauchy or gauss.



Laplace prior



Cauchy prior