Bayesian Neural network

Peng Wu

Blundell, Charles, et al. "Weight uncertainty in neural networks." *arXiv preprint arXiv:1505.05424* (2015).

Outline

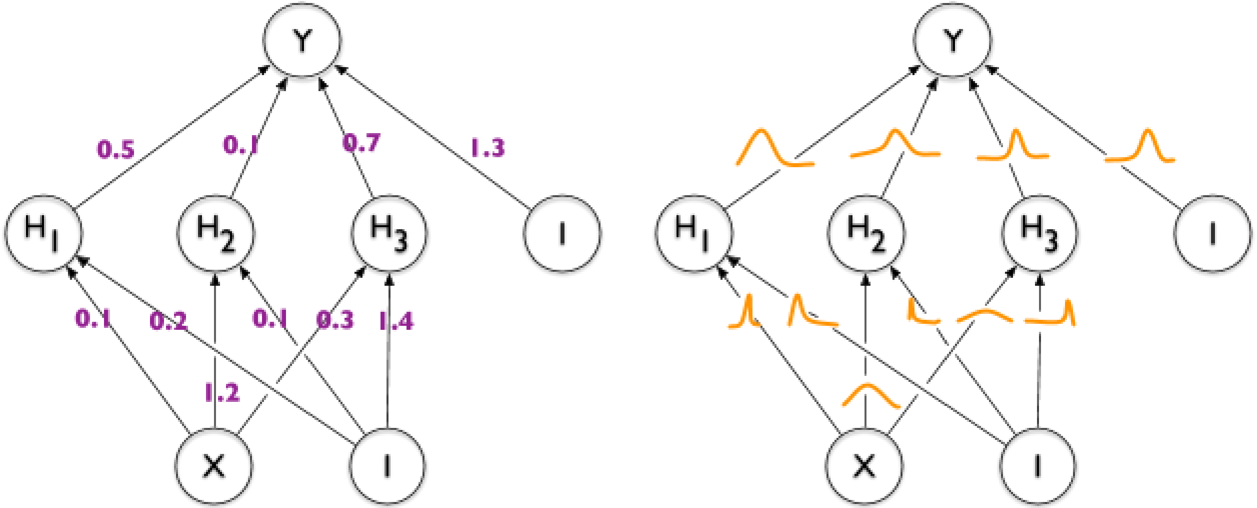
- What is Bayesian Neural network
- Probabilistic Model
- Variational Inference
- Network training

    -- Re-parameterization

    -- Back propagation

- Implementation example
- Reference

What is Bayesian Neural network?



Probabilistic model

MLE: Given a training dataset D= (x^i,y^i), construct the likelihood function

$$p(\mathcal{D}|\mathbf{w}) = \prod_i p(y^{(i)}|\mathbf{x}^{(i)}, \mathbf{w})$$

$$\mathbf{w}^{\mathrm{MLE}} = \arg\max_{\mathbf{w}} \log P(\mathcal{D}|\mathbf{w}) = \arg\max_{\mathbf{w}} \sum_i \log P\left(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w}\right)$$

MAP: Given a training dataset D= (x^i,y^i), introduce prior distribution

$$P(w|D) \propto P(D|w)P(w)$$

$$\mathbf{w}^{\mathrm{MAP}} = \arg\max_{\mathbf{w}} \log P(\mathbf{w}|\mathcal{D}) = \arg\max_{\mathbf{w}} \log P(\mathcal{D}|\mathbf{w}) + \log P(\mathbf{w})$$

Variational inference

$$p(y|x, D) = \int p(y|x, w)p(w|D)dw$$

Intractable:

- Large dimension
- Complex form of posterior distribution

$$\begin{aligned}
\theta^\star &= \arg\min_\theta \mathrm{KL}[q(\mathbf{w}|\theta) \| P(\mathbf{w}|\mathcal{D})] \\
&= \arg\min_\theta \int q(\mathbf{w}|\theta) \log \frac{q(\mathbf{w}|\theta)}{P(\mathbf{w})P(\mathcal{D}|\mathbf{w})} \mathrm{d}\mathbf{w} \\
&= \arg\min_\theta \mathrm{KL}[q(\mathbf{w}|\theta) \| P(\mathbf{w})] - \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D}|\mathbf{w})]
\end{aligned}$$

Variational inference

Cost function:

$$\mathcal{F}(\mathcal{D}, \boldsymbol{\theta}) = \mathrm{KL}(q(\mathbf{w}|\boldsymbol{\theta}) \; || \; p(\mathbf{w})) - \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\theta})} \log p(\mathcal{D}|\mathbf{w})$$

$$\mathcal{F}(\mathcal{D}, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\theta})} \log q(\mathbf{w}|\boldsymbol{\theta}) - \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\theta})} \log p(\mathbf{w}) - \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\theta})} \log p(\mathcal{D}|\mathbf{w})$$

$$\mathcal{F}(\mathcal{D}, \boldsymbol{\theta}) \approx \frac{1}{N} \sum_{i=1}^{N} \left[\log q(\mathbf{w}^{(i)}|\boldsymbol{\theta}) - \log p(\mathbf{w}^{(i)}) - \log p(\mathcal{D}|\mathbf{w}^{(i)}) \right]$$

Network training

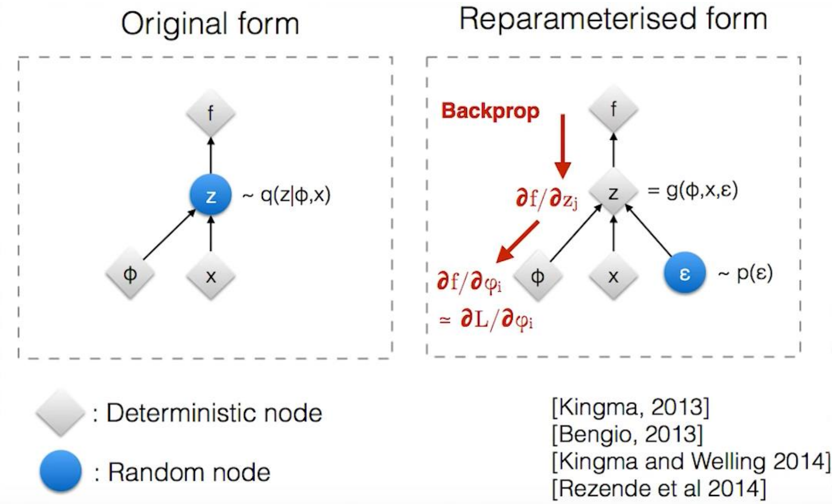
- Forward-pass: Draw sample from the variational posterior distribution
- Backward-pass: Evaluate cost function, take gradient w~N(u,σ)

Re-parameterize trick:

$$t(\mu, \sigma, \epsilon) = \mu + \sigma \odot \epsilon$$

Where $\epsilon \sim N(0, I)$ $\sigma = \log(1 + \exp(\rho))$

$$\mathbf{w} = t(\theta, \epsilon) = \mu + \log(1 + \exp(\rho)) \circ \epsilon$$



Back Propagation

$$q(\epsilon)d\epsilon = q(w|\theta)dw$$

$$\frac{\partial}{\partial\theta}\mathbb{E}_{q(\mathbf{w}|\theta)}[f(\mathbf{w},\theta)] = \mathbb{E}_{q(\epsilon)}\left[\frac{\partial f(\mathbf{w},\theta)}{\partial\mathbf{w}}\frac{\partial\mathbf{w}}{\partial\theta} + \frac{\partial f(\mathbf{w},\theta)}{\partial\theta}\right]$$

Proof:

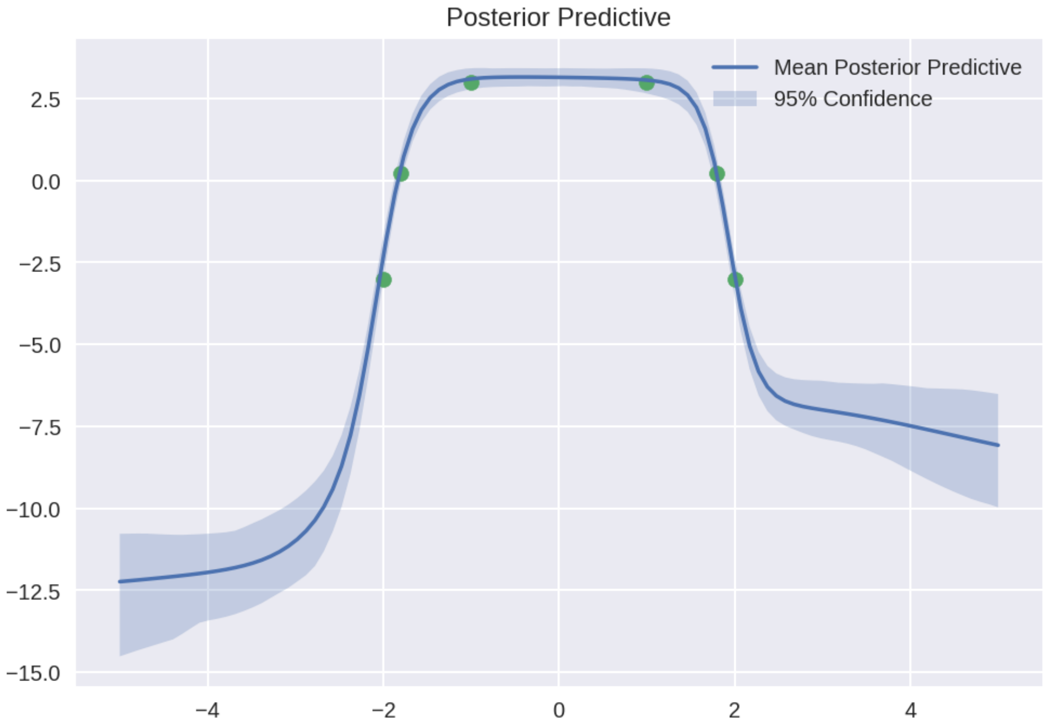
$$\begin{aligned}\frac{\partial}{\partial\theta}\mathbb{E}_{q(\mathbf{w}|\theta)}[f(\mathbf{w},\theta)] &= \frac{\partial}{\partial\theta}\int f(\mathbf{w},\theta)q(\mathbf{w}|\theta)\mathrm{d}\mathbf{w} \\ &= \frac{\partial}{\partial\theta}\int f(\mathbf{w},\theta)q(\epsilon)\mathrm{d}\epsilon \\ &= \mathbb{E}_{q(\epsilon)}\left[\frac{\partial f(\mathbf{w},\theta)}{\partial\mathbf{w}}\frac{\partial\mathbf{w}}{\partial\theta} + \frac{\partial f(\mathbf{w},\theta)}{\partial\theta}\right]\end{aligned}$$

Step by step

1. Sample $\epsilon\ N(0, I)$
2. let $w = \mu + \log(1 + \exp(\rho)) \circ \epsilon$
3. Let $\theta = (\mu, \rho)$.
4. Let $f(w, \theta) = logq(w|\theta) - logP(w)P(D|w)$.
5. Calculate the gradient with respect to the mean
$$\Delta_\mu = \frac{\partial f(\mathbf{w}, \theta)}{\partial \mathbf{w}} + \frac{\partial f(\mathbf{w}, \theta)}{\partial \mu}$$
6. Calculate the gradient with respect to the standard deviation parameter ρ
$$\Delta_\rho = \frac{\partial f(\mathbf{w}, \theta)}{\partial \mathbf{w}} \frac{\epsilon}{1 + \exp(-\rho)} + \frac{\partial f(\mathbf{w}, \theta)}{\partial \rho}$$
7. Update the variational parameters:
$$\mu \leftarrow \mu - \alpha \Delta_\mu$$
$$\rho \leftarrow \rho - \alpha \Delta_\rho$$

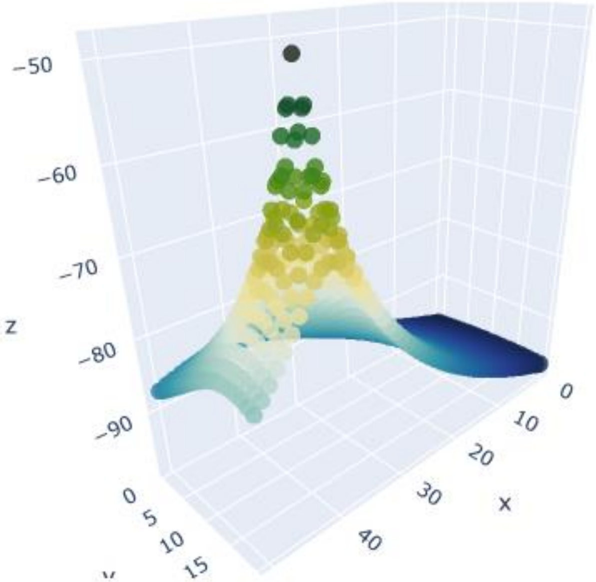
Implementation

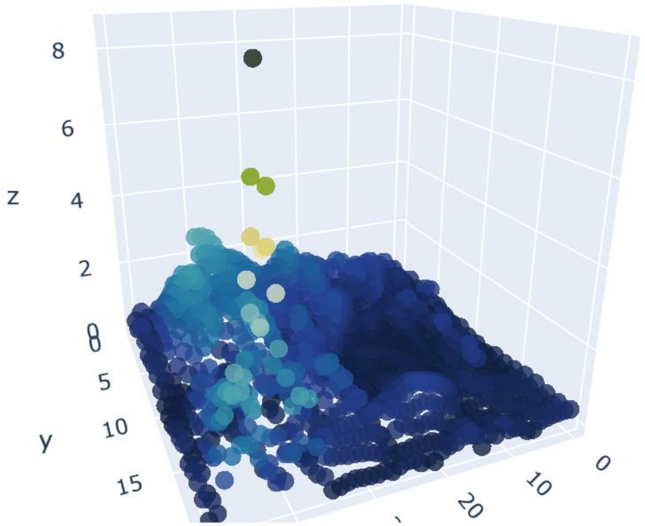
$$f = -x^4 + 3x^2 + 1$$



Path Loss model

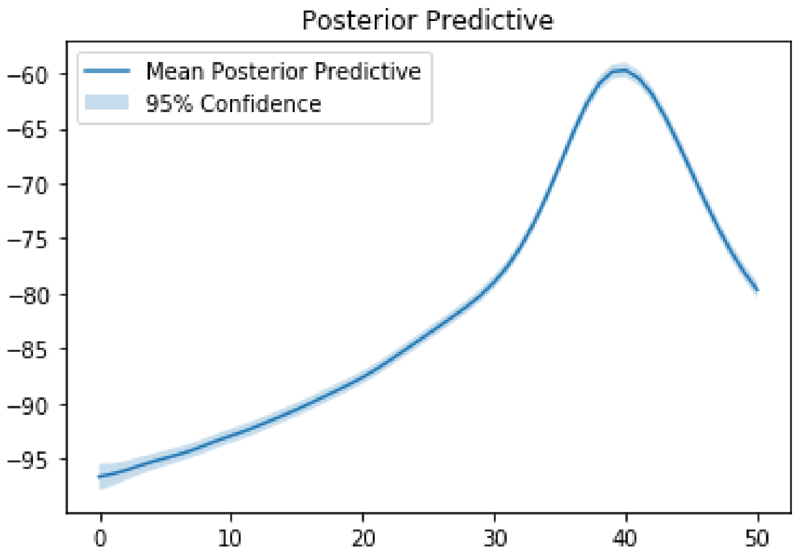
$$y_i^{[\ell]} = P_T - P_0 - 10\lambda \log_{10} \frac{d(\mathbf{p}_i, \mathbf{p}_a^{[\ell]})}{d_0} - \sum_{j=1}^{N_b} \varphi_{\mathbf{c}_j}(\mathbf{p}_i) + \eta_i^{[\ell]}$$



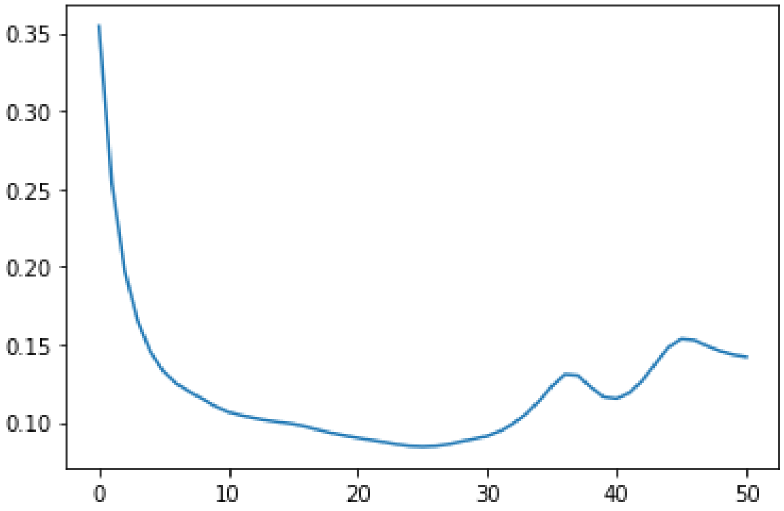


$$|y_{real} - y_{predict}|$$

Implementation



X



X

Questions

- The location that have large error prediction should have large variance, but why is not this situation?
- How to train a neural network that can approximate the simulated function perfectly?
- ….

Reference

- https://github.com/cpark321/uncertainty-deep-learning
- https://joshfeldman.net/ml/2018/12/17/WeightUncertainty.html
- http://krasserm.github.io/2019/03/14/bayesian-neural-networks/

- https://github.com/krasserm/bayesian-machine-learning
- https://github.com/JavierAntoran/Bayesian-Neural-Networks
- https://github.com/nitarshan/bayes-by-backprop/blob/master/Weight%20Uncertainty%20in%20Neural%20Networks.ipynb
- https://paperswithcode.com/paper/weight-uncertainty-in-neural-networks
- https://github.com/anassinator/bnn