Bayesian Deep Learning

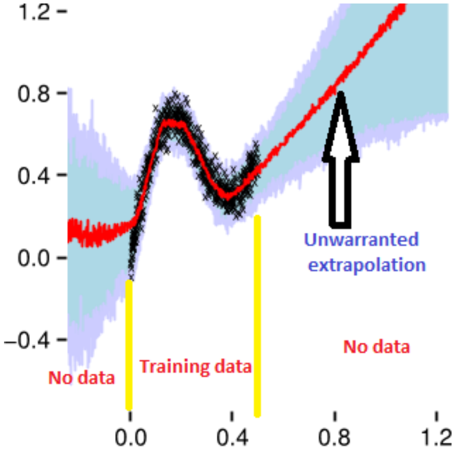
Jaewoo Park

Department of Applied Statistics, Yonsei University

Statistician's Perspective

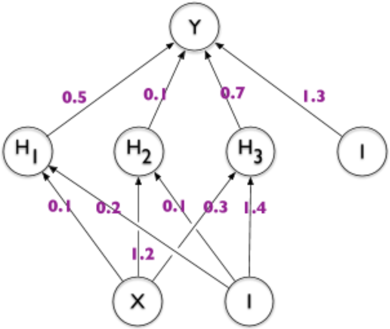
- Until now, we have learned various types of neural networks
- We didn't talk much about distributions, random variables, confidence interval compared to standard statistical methods.
- Still they are important!

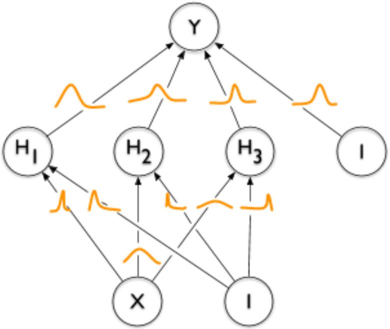
Uncertainty Quantification



(Blundell et al., 2015)

Uncertainties in Neural Network





- Consider the posterior distribution of weight parameters
- This can provide uncertainties for predictions

Bayesian Neural Network (BNN)

- Posterior distribution of BNN:

$$p(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{X}, \boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b})p(\boldsymbol{W})p(\boldsymbol{b})$$

  where $\boldsymbol{W}, \boldsymbol{b}$ are weight and bias parameters and $\boldsymbol{X}, \boldsymbol{Y}$ are the input and output from the training data

- Posterior predictive distribution of BNN:

$$p(\boldsymbol{Y}^*|\boldsymbol{X}^*) = \int\int p(\boldsymbol{Y}^*|\boldsymbol{X}^*, \boldsymbol{W}, \boldsymbol{b})p(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{X}, \boldsymbol{Y})d\boldsymbol{W}d\boldsymbol{b}$$

  where $\boldsymbol{X}^*, \boldsymbol{Y}^*$ are unobserved input and output from the test data

Markov Chain Monte Carlo (MCMC)

- We can construct MCMC with acceptance probability as

$$\alpha = \min\left\{\frac{p(\boldsymbol{W}^{'}, \boldsymbol{b}^{'}|\boldsymbol{X}, \boldsymbol{Y})Q(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{W}', \boldsymbol{b}')}{p(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{X}, \boldsymbol{Y})Q(\boldsymbol{W}', \boldsymbol{b}'|\boldsymbol{W}, \boldsymbol{b})}\right\}$$

  where $Q(\boldsymbol{W}', \boldsymbol{b}'|\boldsymbol{W}, \boldsymbol{b})$ is a proposal distribution
- Then we have MCMC samples $\{\boldsymbol{W}_m, \boldsymbol{b}_m\}_{m=1}^{M}$ from $p(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{X}, \boldsymbol{Y})$

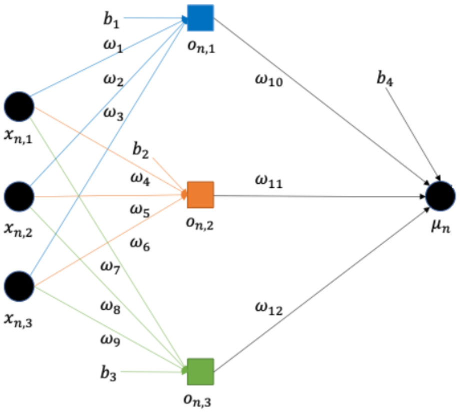
Markov Chain Monte Carlo (MCMC)

- (Remind) Posterior predictive distribution of BNN:

$$p(\boldsymbol{Y}^*|\boldsymbol{X}^*) = \int\int p(\boldsymbol{Y}^*|\boldsymbol{X}^*, \boldsymbol{W}, \boldsymbol{b}) p(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{X}, \boldsymbol{Y}) d\boldsymbol{W} d\boldsymbol{b}$$

- Given MCMC samples $\{\boldsymbol{W}_m, \boldsymbol{b}_m\}_{m=1}^{M}$ from $p(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{X}, \boldsymbol{Y})$, generate $\boldsymbol{Y}_m^* \sim p(\boldsymbol{Y}^*|\boldsymbol{X}^*, \boldsymbol{W}_m, \boldsymbol{b}_m)$
- Repeat this M times result in posterior predictive samples $\{\boldsymbol{Y}_m^*\}_{m=1}^{M}$
- We can quantify uncertainties in prediction!

Example: Toy BNN



- Input $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^{N} \in \mathbb{R}^{600\times 3}$, output $\boldsymbol{Y} = \{y_n\}_{n=1}^{N} \in \mathbb{R}^{600}$
- Weight $(w_1, \cdots, w_{12})$, bias $(b_1, \cdots, b_4)$

Example: Toy BNN (contd.)

- Forward propagation (NN structure):

$$o_{n,1} = \tanh(x_{n,1}w_1 + x_{n,2}w_2 + x_{n,3}w_3 + b_1)$$

$$o_{n,1} = \tanh(x_{n,1}w_4 + x_{n,2}w_5 + x_{n,3}w_6 + b_2)$$

$$o_{n,1} = \tanh(x_{n,1}w_7 + x_{n,2}w_8 + x_{n,3}w_9 + b_3)$$

$$\mu_n = o_{n,1}w_{10} + o_{n,1}w_{11} + o_{n,1}w_{12} + b_4$$

Example: Toy BNN (contd.)

- Prior

$$w_i \overset{\text{iid}}{\sim} N(0, 10), \quad b_j \overset{\text{iid}}{\sim} N(0, 10), \quad \sigma^2 \sim G(0.5, 1)$$

- Likelihood

$$y_n \overset{\text{iid}}{\sim} N(\mu_n, \sigma^2)$$

- Posterior

$$p(\boldsymbol{W}, \boldsymbol{b} | \boldsymbol{X}, \boldsymbol{Y}) \propto \left[\prod_{n=1}^{N} p(y_n | \mu_n) \right] \times \left[\prod_{i=1}^{12} p(w_i) \prod_{j=1}^{4} p(b_j) p(\sigma^2) \right]$$

Note: MCMC Approach

- We can conduct an "exact" Bayesian inference from $p(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{X}, \boldsymbol{Y})$
- We can make a prediction via $p(\boldsymbol{Y}^*|\boldsymbol{X}^*)$
- However, we need to run longer changes as we have more parameters
- This is infeasible for DNN (e.g., CNN, RNN)

Bayes by Backprop (Blundell et al., 2015)

- Backpropagation algorithm for learning a distribution of the weights
- Instead of training a single network (point estimate), use an ensemble of networks, where each network has its weights drawn from a distribution

Variational Inference (VI)

- Approximates the posterior through $q_{\xi}(\boldsymbol{W}, \boldsymbol{b})$ (variational distribution)
- Find $q_{\xi}(\boldsymbol{W}, \boldsymbol{b})$ that minimizes the Kullback–Leibler (KL) divergence between $q_{\xi}(\boldsymbol{W}, \boldsymbol{b})$ and $p(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{X}, \boldsymbol{Y})$
  - We need to set a class of distributions (e.g., Gaussian)
  - ξ is a variational parameter (e.g., mean and covariance of Gaussian)
  - A practical option for BNN

Variational Inference (contd.)

- KL divergence is defined as

$$\mathrm{KL}(q_\xi(\boldsymbol{W}, \boldsymbol{b}) || p(\boldsymbol{W}, \boldsymbol{b} | \boldsymbol{X}, \boldsymbol{Y})) = \int\int q_\xi(\boldsymbol{W}, \boldsymbol{b}) \frac{q_\xi(\boldsymbol{W}, \boldsymbol{b})}{p(\boldsymbol{W}, \boldsymbol{b} | \boldsymbol{X}, \boldsymbol{Y})} d\boldsymbol{W} d\boldsymbol{b}$$

  which is intractable due to $p(\boldsymbol{W}, \boldsymbol{b} | \boldsymbol{X}, \boldsymbol{Y})$ terms
- Minimizing the KL divergence is equivalent to maximizing evidence lower-bound (ELBO)

$$\mathrm{ELBO} = \int\int q_\xi(\boldsymbol{W}, \boldsymbol{b}) \log p(\boldsymbol{Y} | \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b}) d\boldsymbol{W} d\boldsymbol{b} - \mathrm{KL}(q_\xi(\boldsymbol{W}, \boldsymbol{b}) || p(\boldsymbol{W}) p(\boldsymbol{b}))$$

Variational Inference (contd.)

- We don't need the posterior $p(\boldsymbol{W}, \boldsymbol{b}|\boldsymbol{X}, \boldsymbol{Y})$ when we compute ELBO

$$\text{ELBO} = \int\int q_\xi(\boldsymbol{W}, \boldsymbol{b}) \log p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b}) d\boldsymbol{W} d\boldsymbol{b} -$$

$$\text{KL}(q_\xi(\boldsymbol{W}, \boldsymbol{b}) || p(\boldsymbol{W}) p(\boldsymbol{b}))$$

- All we need is
  - Prior: $p(\boldsymbol{W})p(\boldsymbol{b})$
  - Variational distribution: $q_\xi(\boldsymbol{W}, \boldsymbol{b})$
  - Likelihood: $p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b})$

Being Bayesian by Backpropagation

- The goal is to maximize ELBO

$$\int\int q_{\xi}(\boldsymbol{W},\boldsymbol{b}) \log p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{W},\boldsymbol{b}) d\boldsymbol{W}\, d\boldsymbol{b} - \mathrm{KL}(q_{\xi}(\boldsymbol{W},\boldsymbol{b})||p(\boldsymbol{W})p(\boldsymbol{b}))$$

  which can be represented as

$$\int\int q_{\xi}(\boldsymbol{W},\boldsymbol{b})[\log p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{W},\boldsymbol{b}) - \log q_{\xi}(\boldsymbol{W},\boldsymbol{b}) + \log p(\boldsymbol{W})p(\boldsymbol{b})] d\boldsymbol{W}\, d\boldsymbol{b}$$

- The Monte Carlo approximation above is

$$\log p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{W},\boldsymbol{b}) - \log q_{\xi}(\boldsymbol{W},\boldsymbol{b}) + \log p(\boldsymbol{W})p(\boldsymbol{b})$$

  where $\boldsymbol{W},\boldsymbol{b}$ is generated from $q_{\xi}(\boldsymbol{W},\boldsymbol{b})$

Being Bayesian by Backpropagation (contd.)

- Now the problem is simple
- Maximize the following through backpropagation

$$\log p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{b}) - \log q_{\boldsymbol{\xi}}(\boldsymbol{W}, \boldsymbol{b}) + \log p(\boldsymbol{W})p(\boldsymbol{b})$$

for training $q_{\boldsymbol{\xi}}(\boldsymbol{W}, \boldsymbol{b})$