Bayesian Deep Learning

Soumya Sara John

M.Tech DSP
Department of Avionics
Indian Institute of Space Science and Technology

13 August 2018

Outline

Outline

Bayesian Deep Learning

Introduction

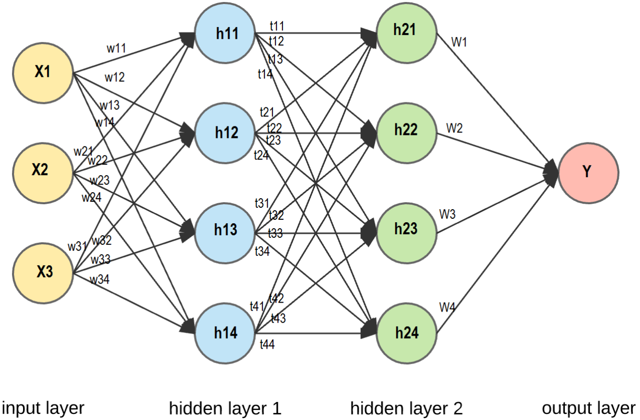

Figure: Two layer DNN : weights are point estimates

Bayesian Deep Learning
Introduction

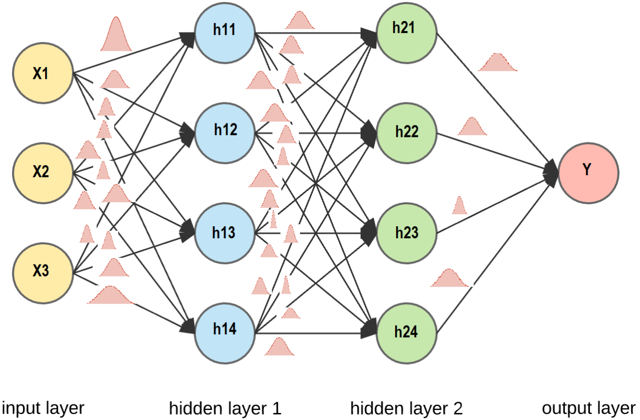

Figure: Two layer DNN : weights are defined using Gaussian distributions

Bayes Probability

- $$P(B/A) = \frac{P(A/B)P(B)}{P(A)} \tag{1}$$

Outline

- Dataset : $\{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$

$$y(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + \epsilon(\mathbf{x})$$

$$\mathbf{w} \in R^d \rightarrow \text{parameters}$$

$$\epsilon(\mathbf{x}) \rightarrow \text{residuals}$$

- Ordinary Least squares: $\hat{\mathbf{w}} = \underset{\mathbf{w}}{argmin} \sum_{i=1}^{N} (\mathbf{x}_i^T\mathbf{w} - y_i)^2$

$$\Rightarrow \hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$$

- Overfitting might occur

- Adding a regularization term

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{argmin} \sum_{i=1}^{N} (\mathbf{x}_i^T \mathbf{w} - y_i)^2 + \lambda ||\mathbf{w}||_2^2$$

$$\Rightarrow \hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X} + \lambda I)^{-1} \mathbf{X}^T y$$

Regression

Bayesian Regression - Weight Space View

- $y(\mathbf{x}) = \mathbf{x}^T \mathbf{w} + \epsilon(\mathbf{x})$

$$p(\epsilon|\sigma^2) = N(\epsilon; 0, \sigma^2 I)$$

$$p(\mathbf{w}|\mu, \Sigma) = N(\mathbf{w}; \mu, \Sigma)$$

Regression

Bayesian Regression - Weight Space View

- $y(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + \epsilon(\mathbf{x})$

$$p(\epsilon|\sigma^2) = N(\epsilon; 0, \sigma^2 I)$$

$$p(\mathbf{w}|\mu, \Sigma) = N(\mathbf{w}; \mu, \Sigma)$$

$$p(y_i|\mathbf{x}_i, \mathbf{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(y_i - \mathbf{x}_i^T\mathbf{w})^2}{2\sigma^2}}$$

Regression

Bayesian Regression - Weight Space View

- $y(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + \epsilon(\mathbf{x})$

$$p(\epsilon|\sigma^2) = N(\epsilon; 0, \sigma^2 I)$$

$$p(\mathbf{w}|\mu, \Sigma) = N(\mathbf{w}; \mu, \Sigma)$$

$$p(y_i|\mathbf{x}_i, \mathbf{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(y_i - \mathbf{x}_i^T\mathbf{w})^2}{2\sigma^2}}$$

- Likelihood : $p(\mathbf{Y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \mathbf{w})$

Regression

Bayesian Regression - Weight Space View

- Likelihood : $p(\mathbf{Y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \mathbf{w})$
- Prior : p($\mathbf{w}$)

Regression

Bayesian Regression - Weight Space View

- Likelihood : $p(\mathbf{Y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \mathbf{w})$
- Prior : $\mathrm{p}(\mathbf{w})$
- Posterior : $p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})d\mathbf{w}}$

Regression

Bayesian Regression - Weight Space View

- Likelihood : $p(\mathbf{Y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \mathbf{w})$
- Prior : p($\mathbf{w}$)
- Posterior : $p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})d\mathbf{w}}$
- $\hat{\mathbf{w}} = \underset{\mathbf{w}}{argmax}\, p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$

Gaussian Process

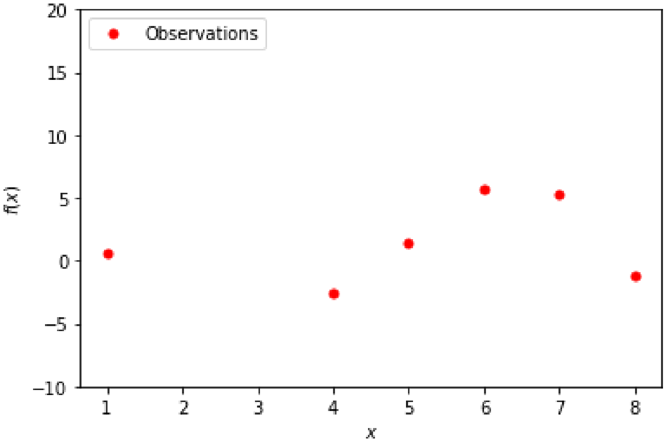

Figure: Observation Points from f(x) = xcos(x)

Gaussian Process

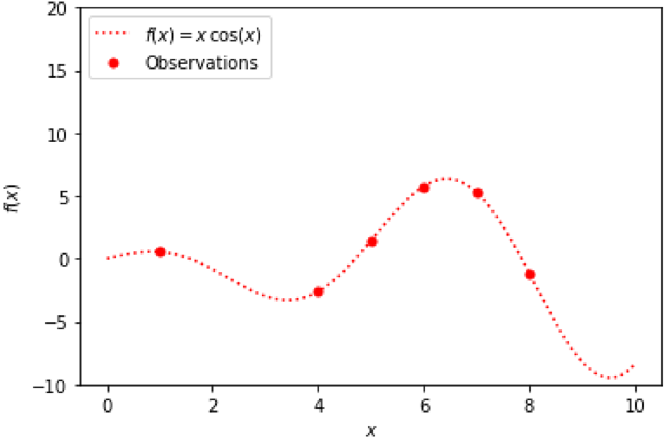

Figure: f(x) = xcos(x)

Gaussian Process

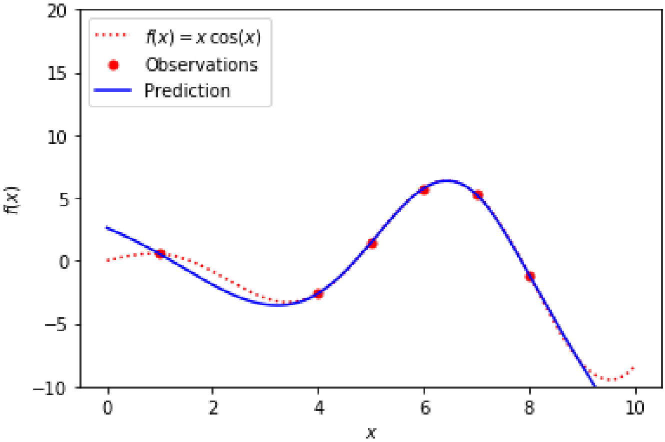

Figure: f(x) = xcos(x) and predicted function using MLE

Gaussian Process

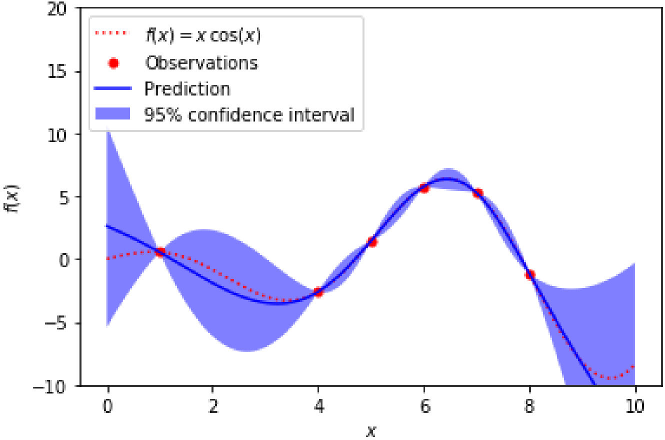

Figure: With uncertainty range

Regression

Bayesian Regression - Function Space View

- A function f($\mathbf{x}$) defined such that $p(f) \sim GP(f; \mu, K)$

  $\mu(\mathbf{X}) = [f(\mathbf{X})]$ and $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-0.5||\mathbf{x}_i - \mathbf{x}_j||_2^2}$

Regression

Bayesian Regression - Function Space View

- A function f($\mathbf{x}$) defined such that $p(f) \sim GP(f; \mu, K)$

  $\mu(\mathbf{X}) = [f(\mathbf{X})]$ and $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-0.5||\mathbf{x}_i - \mathbf{x}_j||_2^2}$
- Eg: 3 points $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ and

  correspondingly, $\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}\right)$

Regression

Bayesian Regression - Function Space View

- Posterior : $p(f|D) = \frac{p(D|f)p(f)}{\int p(D|f)p(f)df}$
- $\hat{f} = \underset{f}{argmax}\, p(f|D)$

Regression

Bayesian Regression - Function Space View

- Posterior : $p(f|D) = \frac{p(D|f)p(f)}{\int p(D|f)p(f)df}$
- $\hat{f} = \underset{f}{argmax}\, p(f|D)$
- Likelihood : $p(D|f) \Rightarrow \mathrm{p}(\mathbf{Y}|\mathrm{f}) = \prod_{i=1}^{N} p(y_i|f_i)$
- $p(y_i = 1|f_i) = \sigma(f_i)$

Regression

Bayesian Regression - Function Space View

- And for a new point $\mathbf{x}^*$, get $f(\mathbf{x}^*) = f^*$ using the conditional distribution
- Joint distribution :

$$\begin{bmatrix} \mathbf{f} \\ f^* \end{bmatrix} = N\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{1*} \\ K_{21} & K_{22} & K_{23} & K_{2*} \\ K_{31} & K_{32} & K_{33} & K_{3*} \\ K_{*1} & K_{*2} & K_{*3} & K_{**} \end{bmatrix} \right)$$

Regression

Bayesian Regression - Function Space View

- In actual scenario, mean $\neq 0$:
$$\begin{bmatrix} \mathbf{f} \\ \mathrm{f}* \end{bmatrix} = N\left(\begin{bmatrix} \mu(\mathbf{X}) \\ \mu* \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K*}^T \\ \mathbf{K*} & \mathbf{K**} \end{bmatrix} \right)$$

- In actual scenario, mean $\neq 0$:
$$\begin{bmatrix} \mathbf{f} \\ \mathrm{f}* \end{bmatrix} = N\left(\begin{bmatrix} \mu(\mathbf{X}) \\ \mu* \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K*}^T \\ \mathbf{K*} & \mathbf{K**} \end{bmatrix} \right)$$
- Conditional Distribution:

$$p(f^*|\mathbf{X*}, \mathbf{X}, \mathbf{f}) = N(f^*|\mu^*, \Sigma^*)$$

- In actual scenario, mean $\neq 0$:
$$\begin{bmatrix} \mathbf{f} \\ \mathrm{f*} \end{bmatrix} = N\left(\begin{bmatrix} \mu(\mathbf{X}) \\ \mu* \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K*}^T \\ \mathbf{K*} & \mathbf{K**} \end{bmatrix} \right)$$
- Conditional Distribution:

$$p(f^*|\mathbf{X*}, \mathbf{X}, \mathbf{f}) = N(f^*|\mu^*, \Sigma^*)$$

such that :

$$\mu^* = \mu(\mathbf{X}^*) + \mathbf{K}^*\mathbf{K}^{-1}(f - \mu(\mathbf{X}))$$

$$\Sigma^* = \mathbf{K}^{**} - \mathbf{K}^*\mathbf{K}^{-1}\mathbf{K}^*$$

Bayesian Inference

- Two steps:
  - $p(f^*|\mathbf{X}^*, D) = \int p(f^*|\mathbf{X}^*, \mathbf{X}, \mathbf{f}) d\mathbf{f}$

Bayesian Inference

- Two steps:
  - $p(f^*|\mathbf{X}^*, D) = \int p(f^*|\mathbf{X}^*, \mathbf{X}, \mathbf{f})d\mathbf{f}$
  - $p(y_i^* = 1|f^*) = \int p(y_i^* = 1|f^*)p(f^*|\mathbf{X}^*, D)df^*$

$$= \int \sigma(f^*)p(f^*|\mathbf{x}^*, D)df^*$$

Bayesian Inference

Difficulties and Solutions

- Difficulties:
  - Finding the right prior
  - Intractable posterior

Bayesian Inference

Difficulties and Solutions

- Difficulties:
  - Finding the right prior
  - Intractable posterior
- Solutions:
  - Sampling the posterior appropriately
  - Approximating the posterior

Outline

Sampling the posterior
Markov Chain Monte Carlo

- General Steps:
  - Current parameter $\mathbf{w}_{current}$
  - Propose new parameter $\mathbf{w}_{new}$
  - Accept or reject the proposed value based on probability
- Metrapolis Algorithm uses a Normal distribution to calculate the probability
  - $\mathbf{w}_t \sim N(\mu, \Sigma)$
  - $\mathbf{w}_{t+1} \sim N(\mathbf{w}_t, \Sigma)$

Sampling the posterior

Markov Chain Monte Carlo

- General Steps:
  - Current parameter $\mathbf{w}_{current}$
  - Propose new parameter $\mathbf{w}_{new}$
  - Accept or reject the proposed value based on probability
- Metrapolis Algorithm uses a Normal distribution to calculate the probability
  - $\mathbf{w}_t \sim N(\mu, \Sigma)$
  - $\mathbf{w}_{t+1} \sim N(\mathbf{w}_t, \Sigma)$
  - $r(\mathbf{w}_{t+1}, \mathbf{w}_t) = \frac{post.prob.of \mathbf{w}_{t+1}}{post.prob.of \mathbf{w}_t}$

Sampling the posterior

Markov Chain Monte Carlo

- General Steps:
  - Current parameter $\mathbf{w}_{current}$
  - Propose new parameter $\mathbf{w}_{new}$
  - Accept or reject the proposed value based on probability
- Metrapolis Algorithm uses a Normal distribution to calculate the probability
  - $\mathbf{w}_t \sim N(\mu, \Sigma)$
  - $\mathbf{w}_{t+1} \sim N(\mathbf{w}_t, \Sigma)$
  - $r(\mathbf{w}_{t+1}, \mathbf{w}_t) = \frac{post.prob.of\, \mathbf{w}_{t+1}}{post.prob.of\, \mathbf{w}_t}$
  - if $r(\mathbf{w}_{t+1}, \mathbf{w}_t) > 1$, accept $\mathbf{w}_{t+1}$

Sampling the posterior

Markov Chain Monte Carlo

- Two issues:
  - Dependent on the starting values
  - Correlation present because of Markov Chain
- Solution:
  - Burn-in period
  - Thinning : increasing the sample size
- Advantage : Accuracy high
- Disadvantage: Slow

Outline

Approximating the posterior

Laplace Approximation

- Parameter space w and data D
- Posterior : $p(w|D) = \frac{1}{Z} p(D|w)p(w)$

$$Z = \int p(D|w)p(w)$$

Approximating the posterior

Laplace Approximation

- Parameter space w and data D
- Posterior : $p(w|D) = \frac{1}{Z}p(D|w)p(w)$

$$Z = \int p(D|w)p(w)$$

- $\psi(w) = log(p(D|w)) + log(p(w))$

Approximating the posterior

Laplace Approximation

- Parameter space w and data D
- Posterior : $p(w|D) = \frac{1}{Z} p(D|w)p(w)$

$$Z = \int p(D|w)p(w)$$

- $\psi(w) = log(p(D|w)) + log(p(w))$

$$\hat{w} = \underset{w}{argmax}\ \psi(w)$$

Approximating the posterior

Laplace Approximation

- Parameter space w and data D
- Posterior : $p(w|D) = \frac{1}{Z} p(D|w)p(w)$

$$Z = \int p(D|w)p(w)$$

- $\psi(w) = log(p(D|w)) + log(p(w))$

$$\hat{w} = \underset{w}{argmax}\ \psi(w) \rightarrow \text{MAP}$$

Approximating the posterior

Laplace Approximation

- Taylors series :

$$\psi(w) = \psi(\hat{w}) + \frac{-1}{2}(w - \hat{w})^T H(w - \hat{w})$$

Approximating the posterior

Laplace Approximation

- Taylors series :

$$\psi(w) = \psi(\hat{w}) + \frac{-1}{2}(w - \hat{w})^T H(w - \hat{w})$$

- $p(w|D) = e^{\psi(w)}$

Approximating the posterior

Laplace Approximation

- Taylors series :

$$\psi(w) = \psi(\hat{w}) + \frac{-1}{2}(w - \hat{w})^T H(w - \hat{w})$$

- $p(w|D) = e^{\psi(w)}$

$$= e^{\psi(\hat{w})} e^{\frac{-1}{2}(w-\hat{w})^T H(w-\hat{w})}$$

Approximating the posterior

Laplace Approximation

- Taylors series :

$$\psi(w) = \psi(\hat{w}) + \frac{-1}{2}(w - \hat{w})^T H(w - \hat{w})$$

- $p(w|D) = e^{\psi(w)}$

$$= e^{\psi(\hat{w})} e^{\frac{-1}{2}(w-\hat{w})^T H(w-\hat{w})}$$

$$\approx N(w; \hat{w}, H^{-1})$$

Approximating the posterior

Laplace Approximation

- Taylors series :

$$\psi(w) = \psi(\hat{w}) + \frac{-1}{2}(w - \hat{w})^T H(w - \hat{w})$$

- $p(w|D) = e^{\psi(w)}$

$$= e^{\psi(\hat{w})} e^{\frac{-1}{2}(w-\hat{w})^T H(w-\hat{w})}$$

$$\approx N(w; \hat{w}, H^{-1})$$

- $Z = \int e^{\psi(w)} dw$

Approximating the posterior

Laplace Approximation

- Taylors series :

$$\psi(w) = \psi(\hat{w}) + \frac{-1}{2}(w - \hat{w})^T H(w - \hat{w})$$

- $p(w|D) = e^{\psi(w)}$

$$= e^{\psi(\hat{w})} e^{\frac{-1}{2}(w-\hat{w})^T H(w-\hat{w})}$$

$$\approx N(w; \hat{w}, H^{-1})$$

- $Z = \int e^{\psi(w)} dw$

$$= e^{\psi(\hat{w})} \frac{(2\pi)^{d/2}}{|H|^{1/2}}$$

Approximating the posterior

Laplace Approximation

- Advantage: Fast
- Disadvantage: Less accurate

Outline

Approximating the posterior

Variational Inference

- Posterior : $p(Z|X) = \frac{p(X|Z)p(Z)}{\int p(X|Z)p(Z)dZ}$

Approximating the posterior

Variational Inference

- Posterior : $p(Z|X) = \frac{p(X|Z)p(Z)}{\int p(X|Z)p(Z)dZ}$
- Approximate it to $q(Z;\theta)$
- Distance measurement : KL divergence defined as :

$$KL(q(Z;\theta)||p(Z|X)) = \sum_{z \in Z} q(z;\theta) log(\frac{q(z;\theta)}{p(z|x)})$$

Approximating the posterior

Variational Inference

- $KL(q(Z;\theta)||p(Z|X)) = \sum_{z \in Z} q(z;\theta) log(\frac{q(z;\theta)}{p(z|x)})$

$$= \sum_{z \in Z} q(z;\theta) log(\frac{q(z;\theta)p(x)}{p(z,x)})$$

$$= \sum_{z \in Z} q(z;\theta)(log(\frac{q(z;\theta)}{p(z,x)}) + log(p(x)))$$

$$= log(p(x)) + E_q[log(\frac{q(z;\theta)}{p(z,x)}]$$

Approximating the posterior

Variational Inference

- $KL(q(Z;\theta)||p(Z|X)) = \sum_{z \in Z} q(z;\theta) log(\frac{q(z;\theta)}{p(z|x)})$

$$= \sum_{z \in Z} q(z;\theta) log(\frac{q(z;\theta)p(x)}{p(z,x)})$$

$$= \sum_{z \in Z} q(z;\theta)(log(\frac{q(z;\theta)}{p(z,x)}) + log(p(x)))$$

$$= log(p(x)) + E_q[log(\frac{q(z;\theta)}{p(z,x)})]$$

$$\downarrow$$

Minimize

Approximating the posterior

Variational Inference

- $KL(q(Z;\theta)||p(Z|X)) = \sum_{z \in Z} q(z;\theta) log(\frac{q(z;\theta)}{p(z|x)})$

$$= \sum_{z \in Z} q(z;\theta) log(\frac{q(z;\theta)p(x)}{p(z,x)})$$

$$= \sum_{z \in Z} q(z;\theta)(log(\frac{q(z;\theta)}{p(z,x)}) + log(p(x)))$$

$$= log(p(x)) + E_q[log(\frac{q(z;\theta)}{p(z,x)})]$$

$\downarrow$

Minimize

Variational Bound or ELBO(Evidence Lower Bound)

Approximating the posterior

Variational Inference

- Minimizing $E_q[log(\frac{q(z;\theta)}{p(z,x)}]$

Approximating the posterior

Variational Inference

- Minimizing $E_q[log(\frac{q(z;\theta)}{p(z,x)}]$
- Maximize $-E_q[log(\frac{q(z;\theta)}{p(z,x)}]$

Approximating the posterior

Variational Inference

- Minimizing $E_q[log(\frac{q(z;\theta)}{p(z,x)}]$
- Maximize $-E_q[log(\frac{q(z;\theta)}{p(z,x)}]$

$\Rightarrow$ Maximize $-E_q[log(\frac{q(z;\theta)}{p(x|z)p(z)}]$

Approximating the posterior

Variational Inference

- Minimizing $E_q[log(\frac{q(z;\theta)}{p(z,x)}]$
- Maximize $-E_q[log(\frac{q(z;\theta)}{p(z,x)}]$

$\Rightarrow$ Maximize $-E_q[log(\frac{q(z;\theta)}{p(x|z)p(z)}]$

$\Rightarrow$ Maximize $E_q[log(\frac{p(z)}{q(z;\theta)}] + E_q[log(p(x|z))]$

Approximating the posterior

Variational Inference

- Advantages:
  - Scalable to large datasets
  - Faster than MCMC
- Disadvantages:
- Does not guarantee a globally optimal q(Z:θ)

Outline

Advantages of Bayesian Deep Learning

- Can avoid overfitting
- It explains the hyperparameters in Deep Learning
- Pruning based on probability
- Robust against adversarial examples

Outline

Challenges with Bayesian Deep Learning

- Modelling multi modal distributions which will help in better inference
- Prior distribution selection
- Posterior approximation

References I

Gaussian Processes for Machine Learning
Carl Edward Rasmussen
Christopher K. I. Williams.
MIT 2006

Uncertainty in Deep Learning
Yarin Gal

Variational Inference: A Review for Statisticians
David M. Blei,Alp Kucukelbir,Jon D. McAuliffe