Bayesian Deep Learning

Kangcheng Hou

Zhejiang University

May 16, 2018

Bayesian framework

Bayes theorem

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

$$p(\mathbf{w}|\mathbf{X},\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{Y}|\mathbf{X})}$$

Having the information of posterior distribution, we can predict an output for a new input point $\mathbf{x}^*$

$$p(\mathbf{y}^*|\mathbf{x}^*,\mathbf{X},\mathbf{Y}) = \int p(\mathbf{y}^*|\mathbf{x}^*,\mathbf{w})p(\mathbf{w}|\mathbf{X},\mathbf{Y})d\mathbf{w}$$

An important component in Bayesian framework is model evidence,

$$p(\mathbf{Y}|\mathbf{X},\mathcal{H}) = \int p(\mathbf{Y}|\mathbf{X},\mathbf{w},\mathcal{H})p(\mathbf{w}|\mathcal{H})d\mathbf{w}$$

It can be seen as marginalising the likelihood over $\mathbf{w}$.

Variational Inference

We are interested in the posterior distribution $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$, but this cannot usually be evaluated analytically.

Bayesian Deep Learning I

Our target when developing is to make as less change to the current Deep Learning structure as possible and get uncertainty information from the model. An important quantity in BDL is the posterior distribution of the weights.

$$p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$$

Using variational inference, we use a parametrized distribution $q_\theta(\mathbf{w})$ to approximate the posterior distribution $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$. We minimizes the KL divergence

$$\begin{aligned}
\mathsf{KL}(q_\theta(\mathbf{w})||p(\mathbf{w}|\mathbf{X}, \mathbf{Y})) &= \int q_\theta(\mathbf{w}) \ln \frac{q_\theta(\mathbf{w})}{p(\mathbf{w}|\mathbf{X}, \mathbf{Y})} d\mathbf{w} \\
&\propto \int q_\theta(\mathbf{w}) \ln \frac{q_\theta(\mathbf{w})}{p(\mathbf{w})} - \int q_\theta(\mathbf{w}) \ln p(\mathbf{Y}|\mathbf{X}, \mathbf{w}) d\mathbf{w} \\
&= \mathsf{KL}(q_\theta(\mathbf{w})||p(\mathbf{w})) - \int q_\theta(\mathbf{w}) \ln p(\mathbf{Y}|\mathbf{X}, \mathbf{w}) d\mathbf{w}
\end{aligned}$$

Bayesian Deep Learning II

In DL context, this means the objective function is

$$\mathcal{L}_{\mathsf{VI}}(\theta) = -\sum_{i=1}^{N} \int q_\theta(\mathbf{w}) \ln(p(\mathbf{y}_i | f^{\mathbf{w}}(\mathbf{x}_i)))d\mathbf{w} + \mathsf{KL}(q_\theta(\mathbf{w}) || p(\mathbf{w}))$$

This objective function is not scalable because first the summed-over term is not tractable for DL model and N terms is large in big data century. Using stochastic variational inference will give help with the large N problem. And the monte carlo integration method will help us cope with the

$$\int q_\theta(\mathbf{w}) \ln p(\mathbf{y}_i | f^{\mathbf{w}}(\mathbf{x}_i))d\mathbf{w}$$

term.

Monte Carlo estimation for VI I

Now the problem is evaluating

$$\int q_\theta(\mathbf{w}) \ln p(\mathbf{y}_i | f^{\mathbf{w}}(\mathbf{x}_i)) d\mathbf{w}$$

and optimize the quantity w.r.t θ. We consider a easier from of task

$$I(\theta) = \frac{\partial}{\partial\theta} \int f(x) p_\theta(x) dx$$

Here we use $p_\theta(x) = \mathcal{N}(x; \mu, \sigma^2)$ and $f(x)$ can be arbitrary. We introduce three methods:

Monte Carlo estimation for VI II

The score function estimator

$$\begin{aligned}\frac{\partial}{\partial\theta}\int f(x)p_\theta(x)dx &= \int f(x)\frac{\partial}{\partial\theta}p_\theta(x)dx \\ &= \int f(x)\frac{\partial}{\partial\theta}\log p_\theta(x)p_\theta(x)dx\end{aligned}$$

To estimate this quantity, we estimate

$$\mathbb{E}_{x\sim p_\theta(x)}[f(x)\frac{\partial}{\partial\theta}\ln p_\theta](x)$$

Monte Carlo estimation for VI III

Reparametrisation trick

If we can reparametrize the $p_\theta(x)$, for example, $p_\theta(x)$ can be $\mathcal{N}(\mu, \theta)$. We can thus reparametrize the $p_\theta(x)$ as $g(\theta, \epsilon) = \mu + \sigma\epsilon$ with $p(\epsilon) = \mathcal{N}(\epsilon; 0, I)$. We have the property that

$$p_\theta(x)dx = p(\epsilon)d\epsilon$$

$$\begin{aligned}
\nabla_\theta \int f(x)p_\theta(x)dx &= \nabla_\theta \int f(x)p(\epsilon)d\epsilon \\
&= \nabla_\theta \int f(g(\theta, \epsilon))p(\epsilon)d\epsilon \\
&= \mathbb{E}_{\epsilon\sim p(\epsilon)}[f(g(\theta, \epsilon))] \\
&= \mathbb{E}_{\epsilon\sim p(\epsilon)}[\nabla_\theta f(g(\theta, \epsilon))] \\
&= \mathbb{E}_{\epsilon\sim p(\epsilon)}[f'(g(\theta, \epsilon))\nabla_\theta g(\theta, \epsilon)]
\end{aligned}$$

Monte Carlo estimation for VI IV

Will not introduce the third method here.

Practical inference in Bayesian neural networks I