Selective Review on Bayesian Deep Learning Methods

Yutian Pang

Arizona State University

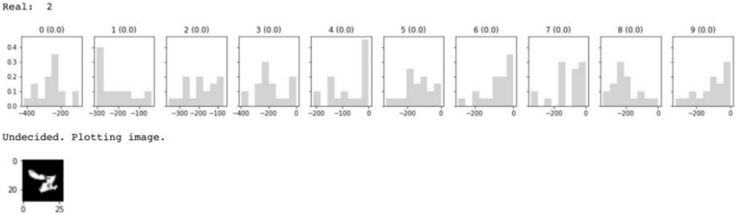
February 16, 2021

Overview

Why Bayesian Learning?

- Deterministic Neural Networks are increasingly being developed in safety critical domains.
- The safety, robustness, and efficient measure of NN are crucial.

Figure: Bayesian NN say *I don't know* on OOD sample.



Bayesian Learning

- Given training dataset $X = \{x_1, ..., x_n\}$ and $Y = \{y_1, ..., y_n\}$
- We try to find the parameters ω for the approximation function $y = f^{\omega}(x)$ that are *most likely* to generate the outputs.
- That is, the inference of $p(\omega|X, Y)$.

$$\overbrace{p(\omega|X, Y)}^{\text{Posterior}} = \frac{\overbrace{p(Y|X, \omega)}^{\text{Likelihood}}\overbrace{p(\omega)}^{\text{Prior}}}{\underbrace{p(Y|X)}_{\text{Evidence}}}$$

- With new observed data x^*, the prediction y^* is a distribution marginalized over the posterior, as $\mathbb{E}_{p(\omega|X,Y)}[p(y^*|x^*, \omega)]$.

Analytical Bayesian Inference

- Full Bayesian form:

$$p^*(\omega) = p(\omega|X, Y) = \frac{p(Y|X,\omega)p(\omega)}{p(Y|X)}$$

- Easy for conjugate priors
- Hard for other cases
- Example:

$$p(Y|X,\omega) = \underbrace{\mathcal{N}\mathcal{N}(Y|X,\mu(\omega),\sigma(\omega)^2)}_{\text{Neural Networks}}$$

Approximate Bayesian Inference - MCMC

Markov Chain Monte Carlo (MCMC) Methods are common approaches used for sampling from multi-dimensional distributions.

- Markov Chain is a state transition model based on given probability.
- Monte Carlo is used to estimate expectations by sampling

$$\mathbb{E}_{p(m)} f(m) \approx \frac{1}{N} \sum_{i=1}^{N} f(m_i), m_i \sim p(m)$$

- Gibbs sampling reduce multi-dimension sampling to sequence of 1d samplings but generate correlated samples. Not parallelizable.
- MH sampling is to apply rejection sampling with some critic $\mathcal{A}$. Parallelizable.

Approximate Bayesian Inference - VI

Variational Inference

- Select a family of distributions $Q_\Theta(\Omega) \sim \mathcal{N}(\mu, diag(\sigma_d^2))$.
- Find the best approximation $q_\theta(\omega)$ of $p^*(\omega)$.

$$min_{q_\theta \in Q_\Theta} \quad \mathcal{KL}[q_\theta(\omega) || p^*(\omega)]$$

- Mean field assumption on Q:

$$Q_\Theta = q | q_\theta(\omega) = \prod_{i=1}^{d} q_{\theta i}(\omega_i)$$

ELBO

Then

$$
\begin{aligned}
min_{q_\theta \in Q_\Theta} \quad & \mathcal{KL}(q_\theta(\omega)||p(\omega|X, Y)) \\
&= \int q_\theta(\omega) log \frac{q_\theta(\omega)}{p(\omega|X, Y)}\, d\omega \\
&= \int q_\theta(\omega) log \frac{q_\theta(\omega) p(X, Y)}{p(\omega, X, Y)}\, d\omega \\
&= \underbrace{\log p(X, Y)}_{\text{Constant}} - \underbrace{\int q_\theta(\omega) log \frac{p(\omega, X, Y)}{q_\theta(\omega)}\, d\omega}_{\text{Evidence Lower BOund (ELBO)}}
\end{aligned}
$$

is equal to

$$
min_{q_\theta \in Q_\Theta} \quad - \int q_\theta(\omega) log \frac{p(\omega, X, Y)}{q_\theta(\omega)} d\omega
$$

ELBO

Also,

$$
\begin{aligned}
min_{q_\theta \in Q_\Theta} \quad & -\int q_\theta(\omega) log \frac{p(\omega, X, Y)}{q_\theta(\omega)} d\omega \\
& = \mathcal{KL}(q_\theta(\omega)||p(\omega)) - \int q_\theta(\omega)\, log(p(Y|X,\omega)) d\omega \\
& = \underbrace{\mathcal{KL}(q_\theta(\omega)||p(\omega))}_{\text{Prior dependent}} \underbrace{- \mathbb{E}_{q_\theta(\omega)}[log(p(Y|X,\omega))]}_{\text{Data dependent}}
\end{aligned}
$$

- The prior dependent part is analytical with mean-field assumption (weight description length/complexity cost).
- The data dependent part requires further investigation (data description length/likelihood cost).
- The sum of the two parts is also named variational free energy.

ELBO

Further rearranging of the variational free energy results in,

$$
\begin{aligned}
min_{q_\theta \in Q_\Theta} \quad & \mathcal{KL}(q_\theta(\omega)||p(\omega)) - \mathbb{E}_{q_\theta(\omega)}[log(p(Y|X,\omega))] \\
& = \mathbb{E}_{q_\theta(\omega)} log q_\theta(\omega) - \mathbb{E}_{q_\theta(\omega)} log p(\omega) - \mathbb{E}_{q_\theta(\omega)} log p(Y|X,\omega) \\
& \approx \frac{1}{N} \sum_{i=1}^{N} [log q_\theta(\omega_i) - log p(\omega_i) - log p(Y|X,\omega_i)]
\end{aligned}
$$

The objective can therefore be approximated by drawing Monte Carlo samples ω_i from $q_\theta(\omega)$.

Bayes by Backprop

The gradients of the ELBO is performed with the log-derivative trick and reparameterization trick. This is not included here. If you are interested, we can talk about this personally.

Weight Perturbations

Weight perturbation refer to a class of methods which sample the weights of a neural network stochastically at training time.

- If we denote the output of a BNN as $f(x, w)$, the weights are sampled from the variational distribution q_θ.
- We aim to minimize the expected loss $\mathbb{E}_{(x,y)\sim D, w\sim q_\theta}[\mathcal{L}(f(x, w), y)]$
- $w = \overline{w} + \Delta w$ and Δw is a stochastic perturbation and $\overline{w}$ are the mean weights.

Reparameterization

- If Δw_{ij} are sampled independently from Gaussian with a variance σ_{ij}^2. That is $w_{ij} \sim \mathcal{N}(\overline{w_{ij}}, \sigma_{ij}^2)$.
- This can be reparameterized into $w_{ij} = \overline{w_{ij}} + \sigma_{ij}\epsilon_{ij}$ where $\epsilon_{ij} \sim \mathcal{N}(0,1)$
- This allows the gradients to be computed by backpropagation(addition proves).

Local Reparameterization Trick (LRT)

- In LRT, weight perturbations is reformulated into activation perturbations.
- This enables independently batch sampling during training.
- For example, X is the batch input, W is the weight matrix. $B = WX$ is viewed as the activation matrix. Then LRT sample from B rather than W.

Dropout Approximation

- Yarin Gal proves that when randomly set the columns of the variational distribution $q(\omega)$ to zero, then the prediction over the intractable posterior is exactly the feedforward output of the neural network.

$$W_i = M_i \cdot diag([Z_{i,j}]_{j=1}^{K_i})$$
$$z_{i,j} \sim Bernoulli(p_i) for i = 1, ..., L, j = 1, ...K_{i-1}$$

- Prediction = Mean from Dropout Variational Inference
- Total Variance = Variance from Dropout Variational Inference+Mean of Predictive Variance Output+Inverse Model Precision
- Model precision,

$$\tau = \frac{pl^2}{2N\lambda}$$

- Variational dropout has been applied to RNN and CNN.

Flipout

- Flipout is an efficient mini-batch sampling technique to BNNs.
- Flipout models the weight perturbations with $\Delta w = \widehat{\Delta w} \circ rs^T$, $\circ$ denotes element-wise operations.
- r and s are random vectors sampled uniformly from ± 1.
- This also enables the vectorization for speedup.

$$Y = \phi(X\overline{W} + ((X \circ S)\widehat{\Delta W}) \circ R)$$

- R and S are sampled independently of $\overline{W}$ and $\widehat{\Delta W}$.

Deep Evidential Regression

- It's a sampling-free approach with efficient NN uncertainty measure.
- The problem is formulated as $\mathcal{L}(\omega) = -logp(y|\mu, \sigma^2)$ where $\{\mu, \sigma\} \in \theta$.
- With assumption that y are drawn i.i.d. from from a Gaussian distribution with unknown μ and σ, deep evidential regression place a prior on both as,

$$
\begin{aligned}
(y_1, ..., y_N) &\sim \mathcal{N}(\mu, \sigma^2) \\
\mu &\sim \mathcal{N}(\gamma, \sigma^2 \upsilon^{-1}) \\
\sigma^2 &\sim \Gamma^{-1}(\alpha, \beta)
\end{aligned}
$$

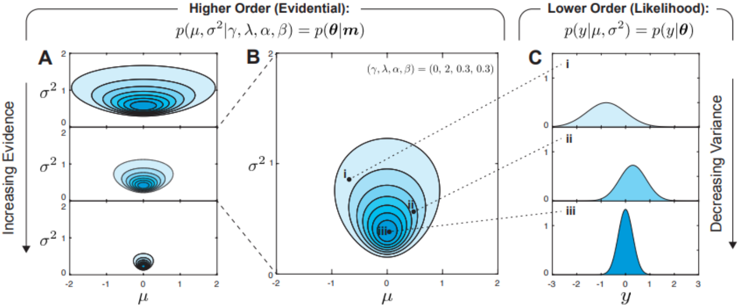
- Again, with mean field assumption on the objective posterior distribution $q(\mu, \sigma^2) = q(\mu)q(\sigma^2) = p(\mu, \sigma^2 | y_1, ..., y_N)$ yields a Gaussian conjugate prior on θ.

Deep Evidential Regression

The Normal Inverse-Gamma (NIG) distribution.

$$p(\underbrace{\mu,\sigma^2}_{\theta}\,|\,\underbrace{\gamma,\upsilon,\alpha,\beta}_{\text{m}}) = \frac{\beta^\alpha\sqrt{\upsilon}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}}\left(\frac{1}{\sigma^2}\right)^{\alpha+1}\exp\left\{-\frac{2\beta+\upsilon(\gamma-\mu)^2}{2\sigma^2}\right\}$$

Figure: NIG parameters m determine the location and the dispersion concentrations.



Deep Evidential Regression

- Neural Network is used to infer m.
- With NIG, the intractable marginal likelihood (model evidence) can be anlytical represented as $p(y|m) = St(y; \gamma, \frac{\beta(1+\upsilon)}{\upsilon\alpha}, 2\alpha)$.
- The loss is the negative log of model evidence with a correction term called evidence regularizer.

$$\mathcal{L}^{NLL}(\omega) = \frac{1}{2} log(\frac{\pi}{\upsilon}) - \alpha log(\Omega) + (\alpha + \frac{1}{2}) log((y-\gamma)^2\upsilon + \Omega) + log(\frac{\Gamma(\alpha)}{\Gamma(\alpha + \frac{1}{2})})$$

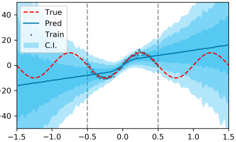
$$\mathcal{L}^{R}(\omega) = |y - \gamma| \cdot (2\upsilon + \alpha)$$

$$\mathcal{L}(\omega) = \mathcal{L}^{R}(\omega) + \mathcal{L}^{NLL}(\omega)$$

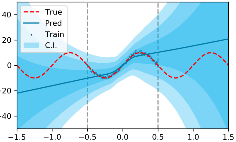
$$\Omega = 2\beta(1 + \upsilon)$$

Simple Regression Demo

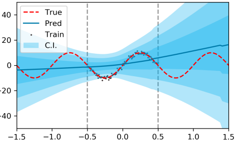
Figure: Bayesian NN: Training $x \in [-0.5, 0.5]$, Testing $x^* \in [-1.5, 1.5]$



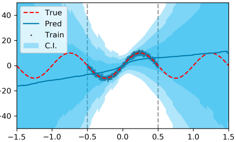
(a) MCDropout



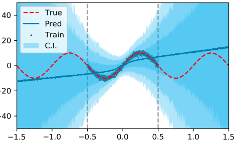
(b) Evidential



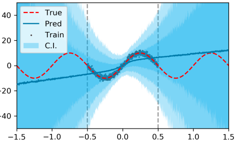
(c) VI



(d) Repara



(e) LRT



(f) Flipout

Conclusion and Insights

- Variants of Bayesian learning methods results into different model performances.
- A comparative study on robustness measure is necessary.
- The concept of Bayesian Neural Network can be applied to any safety critical domains of regression/classification problems.
  - Engineering Applications