Introduction to Bayesian Neural Network

Ryoungwoo Jang

Abstract

Bayesian neural network (BNN)은 Yarin Gal과 그 스승들에 의해서 괄목할만한 발전을 이루었다. 본 글에서는 BNN의 대표적인 논문 두 개와 그 appendix(Dropout as a Bayesian Approximation와 그의 Appendix, What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?)를 살펴보고 BNN에 대한 이해를 증진하는 것을 목표로 한다.

1 Dropout as a Bayesian Approximation

본 논문에서는 임의의 깊이와 non-linearlity를 가지는 deep neural network (deep NN)에 dropout을 각 weight layer 전에 끼얹으면 수학적으로 probabilistic deep Gaussian process와 동치라는 것을 보인다.

1.1 Background

1.1.1 Dropout

일단 single hidden layer를 가지는 NN으로부터 시작하자. 편의상 single hidden layer이고, multi-layer의 경우도 쉽게 확장할 수 있다. W_1, W_2를 각각 input layer과 hidden layer를 이어주는 $Q \times K$ matrix와 hidden layer와 output layer를 이어주는 $K \times D$ matrix라고 하자. Activation function은 σ로 쓰자. 그러면 NN의 output $\hat{y}$는

$$\hat{y} = \sigma(xW_1 + b)W_2 \tag{1}$$

로 쓸 수 있다. Dropout은 잘 알거라 생각한다. Dropout을 수행한다는 것은 어떤 binary vector z_1, z_2들의 원소가 random하게 켜지고 꺼지는 것으로 formulation할 수 있다. 따라서, z_i의 j번째 원소를 $z_{i,j}$라고 쓴다면 다음처럼 표현이 가능하다.

$$z_{1,q} \sim \text{Bernoulli}(p_1), \quad q = 1, \ldots, Q \tag{2}$$

$$z_{2,q} \sim \text{Bernoulli}(p_2), \quad q = 1, \ldots, K \tag{3}$$

이제, NN with dropout은 다음처럼 쓸 수 있다(단, $\circ$는 Hadamard product).

$$\hat{y} = ((\sigma((x \circ z_1)W_1 + b)) \circ z_2)W_2 \tag{4}$$

그런데 이를 잘 생각해보면 다음처럼 행렬곱으로도 표현 가능하다(왜 그러한가?).

$$\hat{y} = \sigma(x(z_1W_1) + b)(z_2W_2) \tag{5}$$

NN을 학습시키기 위새서는 다음과 같은 loss function E들을 고려하기 때문에,

$$E = \frac{1}{2N}\sum_{n=1}^{N} ||y_n - \hat{y}_n||_2^2$$

$$E = -\frac{1}{N}\sum_{n=1}^{N} \log(\hat{p}_{n,c_n})$$

where $\hat{p}_{n,c_n} = \exp(\hat{y}_{nd})/(\sum_{d'} \exp(\hat{y}_{nd'}))$, dropout loss는 다음처럼 정의되는 것이 자연스럽다:

$$\mathcal{L}_{dropout} := E + \lambda_1 ||W_1||_2^2 + \lambda_2 ||W_2||_2^2 + \lambda_3 ||b||_2^2 \tag{6}$$

1.1.2 Gaussian Process

Gaussian process란, stochastic process로 모든 finite collection of random variable들이 multivariate normal distribution을 이루는 stochastic process를 의미한다. 이때 random variable X에 의해서 모델링되는 distribution GP는 X와 같은 차원의 시간 t에 의존하는 벡터 $m(t)$와 공분산행렬을 의미하는 $k(x, x')$에 대해서 다음처럼 표현될 수 있다.

$$X \sim GP(m(t), k(x, x')) \tag{7}$$

이 때 $k(x, x')$을 kernel function이라고 부른다.

1.1.3 Variational Inference

$\mathbf{X}$라는 데이터셋과 $\mathbf{Y}$라는 데이터셋에 대해서 모델이 학습을 했다고 하자. 즉, 모델 p가 $\mathbf{X}$와 $\mathbf{Y}$에 대해서 주어졌다고 하자. 그러면 inference state는 다음처럼 표현이 될 수 있다:

$$p(y^*|x^*, \mathbf{X}, \mathbf{Y}) = \int p(y^*|x^*, \omega) p(\omega|\mathbf{X}, \mathbf{Y}) d\omega \tag{8}$$

그런데 $p(\omega|\mathbf{X}, \mathbf{Y})$를 구하는 것이 가능하지가 않다. 왜냐하면 데이터가 주어졌을 때 각 weight가 나올 확률을 아는 것은 사실상 불가능하기 때문이다. 따라서 우리는 다음과 같은 우회를 한다.

$$\mathrm{KL}(q(\omega)|p(\omega||\mathbf{X}, \mathbf{Y})) \tag{9}$$

를 사용하여 Kullback–Leibler divergence를 최소화하는 방향으로 학습을 진행하게 된다. 이는 approximate predictive distribution

$$q(y^*|x^*) = \int p(y^*|x^*, \omega) q(\omega) d\omega \tag{10}$$

를 계산하는 것으로 치환된다. 이는 Monte Carlo integration으로 가능하다. Kullback–Leibler divergence를 줄이는 것은 log evidence lower bound를 최대화 하는것과 동치이다.

$$\mathcal{L}_{VI} := \int q(\omega) \log p(\mathbf{Y}|\mathbf{X}, \omega) - \mathrm{KL}(q(\omega)||p(\omega)) \tag{11}$$

1.2 Main Results

이제 본격적으로 시작해보자.

1.2.1 A Gaussian Process Approximation

먼저 kernel 하나를 정의한다:

$$\mathbf{K}(x, y) = \int p(w) p(b) \sigma(w^T x + b) \sigma(w^T y + b) \mathrm{d}w \mathrm{d}b \tag{12}$$

이 때 $p(w)$는 standard multivariate normal distribution of dimensionality Q이고 또다른 distribution $p(b)$에 대해서 수식이 전개되었다. 이는 적당한 covariance function이 된다는 것이 알려져

있다(논문 참조). 이제 이를 근사하기 위해 Monte Carlo integration을 수행하자. 즉, $\mathbf{K}$의 근사인 $\widehat{\mathbf{K}}$를 다음처럼 정의한다:

$$\widehat{\mathbf{K}}(x,y) = \frac{1}{K}\sum_{k=1}^{K}\sigma(w_k^T x + b_k)\sigma(w_k^T y + b_k) \tag{13}$$

with $w_k \sim p(w)$ and $b_k \sim p(b)$이다. K는 single hidden layer의 hidden unit의 개수가 될 것이다. 이제, setup이 대충 완료되었다. $\widehat{\mathbf{K}}$를 $\mathbf{K}$ 대신 사용하면 다음과 같은 결론을 얻는다고 한다:

$$w_k \sim p(w), b_k \sim p(b) \tag{14}$$

$$W_1 = [w_k]_{k=1}^K, b = [b_k]_{k=1}^K \tag{15}$$

$$\widehat{\mathbf{K}}(x,y) = \frac{1}{K}\sum_{k=1}^{K}\sigma(w_k^T x + b_k)\sigma(w_k^T y + b_k) \tag{16}$$

$$\mathbf{F}|\mathbf{X}, W_1, b \sim \mathcal{N}(0, \widehat{\mathbf{K}}(\mathbf{X},\mathbf{X})) \tag{17}$$

$$\mathbf{Y}|\mathbf{F} \sim \mathcal{N}(\mathbf{F}, \tau^{-1}\mathbf{I}_N) \tag{18}$$

where W_1 is $Q\times K$ matrix parametrising covariance function이다. Covariance function을 적분하는 것은 다음의 predictive distribution을 유도한다:

$$p(\mathbf{Y}|\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}|\mathbf{X})\mathrm{d}\mathbf{F} \tag{19}$$

$$= \int p(\mathbf{Y}|\mathbf{F})\frac{p(\mathbf{X},\mathbf{F})p(W_1)p(b)}{p(\mathbf{X})p(W_1)p(b)}\mathrm{d}\mathbf{F} \tag{20}$$

$$= \int p(\mathbf{Y}|\mathbf{F})\frac{p(\mathbf{X},\mathbf{F})p(W_1)p(b)}{p(\mathbf{X})p(W_1)p(b)}p(W_1)p(b)\mathrm{d}\mathbf{F}\mathrm{d}W_1\mathrm{d}b \tag{21}$$

$$= \int p(\mathbf{Y}|\mathbf{F})\frac{p(\mathbf{X},\mathbf{F},W_1,b)}{p(\mathbf{X},W_1,b)}p(W_1)p(b)\mathrm{d}\mathbf{F}\mathrm{d}W_1\mathrm{d}b \tag{22}$$

$$= \int p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}|\mathbf{X},W_1,b)p(W_1)p(b)\mathrm{d}\mathbf{F}\mathrm{d}W_1\mathrm{d}b \tag{23}$$

이 때 (21)에서 (22)로 넘어가는 것은 independent 조건때문에 그러하다. 이제 $1\times K$ row vector

$$\phi(x, W_1, b) = \sqrt{\frac{1}{K}}\sigma(W_1^T x + b) \tag{24}$$

와 $N\times K$ feature matrix $\Phi = [\phi(x_n, W_1, b)]_{n=1}^N$를 정의하면, 우리는 $\widehat{\mathbf{K}}(\mathbf{X},\mathbf{X}) = \Phi\Phi^T$를 얻는다. 이제 $p(\mathbf{Y}|\mathbf{X})$를 다시 쓰면 다음과 같다:

$$p(\mathbf{Y}|\mathbf{X}) = \int \mathcal{N}(Y; 0, \Phi\Phi^T + \tau^{-1}\mathbf{I}_N)p(W_1)p(b)\mathrm{d}W_1\mathrm{d}b \tag{25}$$

이 때

$$\mathcal{N}(y_d; 0, \Phi\Phi^T + \tau^{-1}\mathbf{I}_N) = \int \mathcal{N}(y_d; \Phi w_d; \tau^{-1}\mathbf{I}_N)\mathcal{N}(w_d; 0, \mathbf{I}_K)\mathrm{d}w_d \tag{26}$$

로 정의된다. $W_2 = [w_d]_{d=1}^D$를 $K\times D$행렬로 적으면, 위 식은 다음과 동치이다:

$$p(\mathbf{Y}|\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{X},W_1,W_2,b)p(W_1)p(W_2)p(b)\mathrm{d}W_1\mathrm{d}W_2\mathrm{d}b \tag{27}$$

이로써 GP model을 추가적인 random variable W_1, W_2, b에 대해서 re-parametrised했고 다음으로는 적절한 variational distribution에 대해서 posterior를 근사하는 작업을 한다.

1.2.2 Variational Inference in the Approximate Model

Approximate model에 variational inference를 수행하기 위해서 일단 $q(W_1, W_2, b) := q(W_1)q(W_2)q(b)$로 정의하자. 또한 우리의 경험과 상식에 비추어봤을때 $q(W_1)$을 다음처럼 정의한다.

$$q(W_1) = \prod_{q=1}^{Q} q(w_q), \tag{28}$$

$$q(w_q) = p_1 \mathcal{N}(m_q, \sigma^2 \mathbf{I}_K) + (1 - p_1)\mathcal{N}(0, \sigma^2 \mathbf{I}_K) \tag{29}$$

with some probability $p_1 \in [0, 1]$, scalar $\sigma > 0$ and $m_q \in \mathbb{R}^K$이다. W_2에 대해서도 같은 approximationd을 한다:

$$q(W_2) = \prod_{k=1}^{K} q(w_k), \tag{30}$$

$$q(w_k) = p_2 \mathcal{N}(m_k, \sigma^2 \mathbf{I}_D) + (1 - p_2)\mathcal{N}(0, \sigma^2 \mathbf{I}_D) \tag{31}$$

with some probability $p_2 \in [0, 1]$.
또한 b에 대한 Gaussian approximating distribution을 다음처럼 정의한다:

$$q(b) = \mathcal{N}(m, \sigma^2 \mathbf{I}_K) \tag{32}$$

다음으로는 regression을 위한 log evidence lower bound를 구한다. 이는 $M_1 = [m_q]_{q=1}^{Q}$, $M_2 = [m_k]_{k=1}^{K}$, m들에 대해서 최적화를 진행하여 식 (11)를 maximize하기 위함이다. Classification task에 대해서는 뒤에서 다룬다.

1.2.3 Evaluating the Log Evidence Lower Bound for Classification

Classification model의 problem formulation은 다음처럼 할 수 있다:

$$p(\mathbf{c}|\mathbf{X}) = \int p(\mathbf{c}|\mathbf{Y}) p(\mathbf{Y}|\mathbf{X}) \mathrm{d}\mathbf{Y} \tag{33}$$

$$= \int p(\mathbf{c}|\mathbf{Y}) \Big(p(\mathbf{Y}|\mathbf{X}, W_1, W_2, b) p(W_1, W_2, b) \mathrm{d}W_1 \mathrm{d}W_2 \mathrm{d}b \Big) \mathrm{d}\mathbf{Y} \tag{34}$$

이 때 $\mathbf{c}$는 N차원 categorical values를 나타내는 벡터이다. 그러면 우리는 log evidence lower bound를 이 경우에 다음처럼 쓸 수 있다

$$\mathcal{L}_{\text{GP-VI}} := \int p(\mathbf{Y}|\mathbf{X}, W_1, W - 2, b) q(W_1, W_2, b) \log p(\mathbf{c}|\mathbf{Y}) \mathrm{d}W_1 \mathrm{d}W_2 \mathrm{d}b \mathrm{d}\mathbf{Y} \tag{35}$$

$$- \mathrm{KL}(q(W_1, W_2, b) || p(W_1, W_2, b)) \tag{36}$$

그런데 첫 번째 항의 적분중 $\log p(\mathbf{c}|\mathbf{Y})$는 다음처럼 쓸 수 있으므로:

$$\log p(\mathbf{c}|\widehat{\mathbf{Y}}) = \sum_{n=1}^{N} \log p(\mathbf{c}|\widehat{\mathbf{y}}_n) \tag{37}$$

Log evidence lower bound는 다음처럼 쓰여질 수 있다:

$$\mathcal{L}_{\text{GP-VI}} := \sum_{n=1}^{N} \int p(y_n|x_n, W_1, W_2, b) q(W_1, W_2, b) \log p(c_n|y_n) \mathrm{d}W_1 \mathrm{d}W_2 \mathrm{d}b \mathrm{d}y_n \tag{38}$$

$$= -\mathrm{KL}(q(W_1, W_2, b) || p(W_1, W_2, b)) \tag{39}$$

이제 sum 내의 적분인자를 re-parametrize하면

$$W_1 = z_1(M_1 + \sigma\epsilon_1) + (1 - z_1)\sigma\epsilon_1, \tag{40}$$
$$W_2 = z_2(M_2 + \sigma\epsilon_2) + (1 - z_2)\sigma\epsilon_2, \tag{41}$$
$$b = m + \sigma\epsilon, \tag{42}$$
$$y_n = \sqrt{\frac{1}{K}}\sigma(x_n W_1 + b)W_2 \tag{43}$$

가 된다. 이제 Monte Carlo integration을 사용하면

$$\mathcal{L}_{\text{GP-MC}} := \sum_{n=1}^{N} \log p(c_n|\widehat{y}_n(x_n, \widehat{W}_1^n, \widehat{W}_2^n, \widehat{b}^n)) - \text{KL}(q(W_1, W_2, b)||p(W_1, W_2, b)) \tag{44}$$

인데 첫째 항의 각 term은

$$\log p(c_n|\widehat{y}_n) = \widehat{y}_{nc_n} - \log\left(\sum_{d'} \exp(\widehat{y}_{nd'})\right) \tag{45}$$

로 표현 가능하므로 maximization objective는 다음처럼 기술될 수 있다:

$$\mathcal{L}_{\text{GP-MC}} \propto \frac{1}{N}\sum_{n=1}^{N} \widehat{p}_{n,c_n} - \frac{p_1}{2N}||M_1||_2^2 - \frac{p_2}{2N}||M_2||_2^2 - \frac{1}{2N}||m||_2^2 \tag{46}$$

이 때 $\widehat{p}_{n,c_n} = \log p(c_n|\widehat{y}_n)$이다.

1.2.4 Predictive Log-likelihood

데이터셋 $\mathbf{X}, \mathbf{Y}$와 새로운 데이터 포인트 $\mathbf{x}^*$에 대해서 predictive probability $p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y})$로부터 가능한 output values $\mathbf{y}^*$를 얻을 수 있다.
이 predictive log-likilihood는 Monte Carlo integraion으로부터 가능하다.

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \log \int p(\mathbf{y}^*|\mathbf{x}^*, \omega)p(\omega|\mathbf{X}, \mathbf{Y})\mathrm{d}\omega \tag{47}$$
$$\approx \log \int p(p(\mathbf{y}^*|\mathbf{x}^*, \omega)q(\omega)\mathrm{d}\omega \tag{48}$$
$$\approx \log\left(\frac{1}{T}\sum_{t=1}^{T} p(y^*|\mathbf{x}^*, \omega_t)\right) \tag{49}$$

단, $\omega_t \sim q(\omega)$이다. 다차원의 경우는 이를 반복하면 된다.

1.2.5 Predictive Variance

Proposition 1.2.1. Weight matrices M_i of dimension $K_i \times K_{i-1}$과 bias vectors m_i of dimension K_i에 대해서, 그리고 binary vectors z_i of dimension K_{i-1}에 대해서 (단, i는 layer를 나타내고 $i = 1, \ldots, L$), approximating variational distribution

$$q(y^*|x^*) := p(y^*|x^*, \omega)q(\omega) \tag{50}$$
$$q(\omega) = \text{Bern}(z_1) \cdots \text{Bern}(z_L) \tag{51}$$
$$p(y^*|x^*, \omega) = \mathcal{N}(y^*; \widehat{y}^*(x^*, z_1, \cdots, z_L), \tau^{-1}\mathbf{I}_D) \tag{52}$$

for some $tau > 0$이며 또한 multilayer NN에 대해서

$$\widehat{y}^* = \sqrt{\frac{1}{K_L}}(M_L z_L)\sigma\bigg(\dots\sqrt{\frac{1}{K_1}}(M_2 z_2 \sigma((M_1 z_1)x^* + m_1)\dots\bigg) \tag{53}$$

으로 표현될 때

$$\mathbb{E}_{q(y^*|x^*)}((y^*)^T(y^*)) \approx \tau^{-1}\mathbf{I}_D + \frac{1}{T}\sum_{t=1}^{T}\widehat{y}^*(x^*, \widehat{z}_{1,t}, \dots, \widehat{z}_{L,t})^T\widehat{y}^*(x^*, \widehat{z}_{1,t}, \dots, \widehat{z}_{L,t}) \tag{54}$$

with

$$\widehat{z}_{i,t} \sim \text{Bern}(p_i) \tag{55}$$

이다.

Proof.

$$\mathbb{E}_{q(y^*|x^*)}((y^*)^T(y^*)) \tag{56}$$

$$= \int\bigg(\int (y^*)^T(y^*)p(y^*|x^*,\omega)\mathrm{d}y^*\bigg)q(\omega)\mathrm{d}\omega \tag{57}$$

$$= \int\bigg(\text{Cov}_{p(y^*|x^*,\omega)}(y^*) + \mathbb{E}_{p(y^*|x^*,\omega)}(y^*)^T\mathbb{E}_{p(y^*|x^*,\omega)}(y^*)\bigg)q(\omega)\mathrm{d}\omega \tag{58}$$

$$= \int\bigg(\tau^{-1}\mathbf{I}_D + \widehat{y}^*(x^*, z_1, \cdots, z_L)^T\widehat{y}^*(x^*, z_1, \cdots, z_L)\bigg)\text{Bern}(z_1)\cdots\text{Bern}(z_L)\mathrm{d}z_1\cdots,\mathrm{d}z_L \tag{59}$$

$$\approx \tau^{-1}\mathbf{I}_D + \frac{1}{T}\sum_{t=1}^{T}\widehat{y}^*(x^*, z_1, \cdots, z_L)^T\widehat{y}^*(x^*, z_1, \cdots, z_L) \tag{60}$$

since $p(y^*|x^*,\omega) = \mathcal{N}(y^*; \widehat{y}^*(x^*, z_1, \cdots, z_L), \tau^{-1}\mathbf{I}_D)$이기 때문이다. □

1.3 What does it mean?

그렇다면 이로부터 우리가 할 수 있는 것은 무엇일까? 다음에서 위 식의 의의를 살펴보자.

1.3.1 Obtaining Model Uncertainty

위 내용을 다시 써보자. predictive distribution은 다음처럼 쓸 수 있다:

$$q(y^*|x^*) = \int p(y^*|x^*,\omega)q(\omega)\mathrm{d}\omega \tag{61}$$

이 때 ω는 weight(s)이다.
Bernoulli distributoin $\{z_1^t, \cdots, z_L^t\}_{t=1}^T$로부터 유도되는 $z_i^t = [z_{i,j}^t]_{j=1}^{K_i}$는 $\{W_1^t, \dots, W_l^t\}_{t=1}^T$를 다음과 같은 방식으로 주고 (for some matrices M_i^t):

$$W_i^t = M_i^t([z_{i,j}]_{j=1}^{K_i}) \tag{62}$$

이로부터

$$\mathbb{E}_{q(y^*|x^*)}(y^*) \approx \frac{1}{T}\sum_{t=1}^{T}\widehat{y}^*(x^*, W_1^t, \dots, W_L^t) \tag{63}$$

를 얻는데, 이를 Monte Carlo dropout (MC dropout)이라고 부른다. 실제로 이는 T번의 stochastic forward pass를 네트워크를 통해 하고 그 결과를 평균내는것과 같다. 이는 Proposition 1.2.1의 결과이다.

2 What Uncertainties do we need in Bayesian Deep Learning for Computer Vision?

이제 Bayesian deep learning을 활용한 Bayesian neural network (BNN)에서 uncertainty를 어떻게 measure하는지를 살펴보자. 모델 $f^{\widehat{W}}(x)$는 다음처럼 구성된다:

$$[\hat{y}, \hat{\sigma}^2] = f^{\widehat{W}}(x) \tag{64}$$

이 때 $f^{\widehat{W}}(x)$는 model weights $\widehat{W}$로 parametrize된다. 그러면 minimization objective given labeled output points x는 다음과 같다:

$$\mathcal{L}_{BNN}(\theta) = \frac{1}{D}\sum_i \frac{1}{2}\hat{\sigma}_i^{-2}||y_i - \hat{y}_i||^2 + \frac{1}{2}\log\hat{\sigma}_i^2 \tag{65}$$

이 때 계산의 안정성을 위해 $s_i := \log\hat{\sigma}_i^2$으로 두는 것이 좋다고 저자들은 말한다.

$$\mathcal{L}_{BNN}(\theta) = \frac{1}{D}\sum_i \frac{1}{2}\exp(-s_i)||y_i - \hat{y}_i||^2 + \frac{1}{2}s_i \tag{66}$$

위 식에는 analytic solution이 없다는 것을 상기하라. 이렇게 학습을 진행하고 나면 predictive uncertainty for output y는 다음처럼 기술된다:

$$\text{Var}(y) \approx \frac{1}{T}\sum_{t=1}^{T}\hat{y}_t^2 - \left(\frac{1}{T}\sum_{t=1}^{T}\hat{y}_t\right)^2 + \frac{1}{T}\sum_{t=1}^{T}\hat{\sigma}^2 \tag{67}$$

이 때 $\{\hat{y}_t, \hat{\sigma}^2\}_{t=1}^T$는 T개의 sampled outputs: $[\hat{y}, \hat{\sigma}^2] = f^{\widehat{W}}(x)$ for randomly masked weights $\widehat{W}_t \sim q(W)$로부터 나온다.