Weight Uncertainty in Neural Network

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, Daan Wierstra

Google DeepMind

1. Introduction

이 논문에서는 기존 Neural network에는 2가지 문제점이 존재한다고 이야기합니다..

1. overfitting하는 경향이 있다.
2. uncertainty를 반영하지 못한다.

그래서 이 문제점들을 해결하기 위해 **Bayes by backprop(BBB)** 라는 새로운 알고리즘이 적용된 Bayesian Neural Network를 소개합니다.

Bayes by backprop(BBB)란?

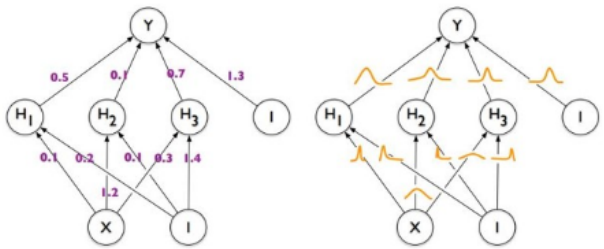

Figure 1. Left: each weight has a fixed value, as provided by classical backpropagation. Right: each weight is assigned a distribution, as provided by Bayes by Backprop.

기존의 네트워크들과는 다르게 각 weight들은 하나의 값이 아닌 하나의 분포를 갖습니다. 각 weight들은 *Gaussian* 분포를 가정하고 μ와 σ 2가지의 *parameter* 를 갖습니다.

각 weight당 *parameter* 수가 특정 값 1개였던 기존 네트워크에서 μ와 σ 2개의 *parameter* 를 갖는 네트워크로 바뀌면서 *parameter* 수가 2배로 늘었지만 하나의 분포에서 sampling할 수 있는 값의 수는 무한개이므로 무한개의 모델을 앙상블한 효과를 가질 수 있습니다.

또한, 동일한 input에 대해 sampling되는 weight의 값에 따라 output이 계속해서 바뀌게 되는데 동일한 input에 대해 나오는 서로 다른 output들의 분산을 측정하여 prediction에 대한 uncertainty를 측정할 수 있습니다.

2. MLE & MAP

기존에 사용되던 대표적인 2가지의 방법, MLE와 MAP를 먼저 소개합니다.

- Maximum Likelihood Estimation(MLE)

MLE는 Likelihood($P(D|w)$)를 maximize시키는 w를 찾는 방법으로, x가 들어왔을 때, y가 나올 확률을 가장 크게 하는 w를 찾는 것입니다. Likelihood를 *Gaussian* 으로 가정했을 시 L2 loss form과 동일한 식이 나오게 됩니다.

$$
\begin{aligned}
w^{MLE} &= arg\max_w \log P(D|w) \\
&= arg\max_w \sum_i \log P(y_i|x_i, w) \\
&= arg\max_w \sum_i \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y_i - f_w(x_i))^2}{2\sigma^2}) \\
&= arg\max_w \sum_i -\frac{(y_i - f_w(x_i))^2}{2\sigma^2} \\
&= arg\min_w \sum_i (y_i - f_w(x_i))^2
\end{aligned}
$$

- Maimum A Posterior(MAP)

MAP는 posterior($P(w|D)$)를 maximize하는 w를 찾는 방법으로 베이즈룰을 사용하여 posterior에 대한 식을 얻을 수 있습니다.

$$
P(w|D) = \frac{P(D|w)P(w)}{P(D)}
$$

분모에 있는 $P(D)$는 w에 대한 식이 아니기 때문에 현재 maximum값을 구하는데 중요하지 않으므로 무시할 수 있습니다. posterior의 식을 보면 Likelihood($P(w|D)$)와 prior($P(w)$)가 곱해진 형태임을 알 수 있습니다. Prior는 우리가 구하고자 하는 parameter에 대한 분포로 저희가 임의로 가정할 수 있습니다. MAP는 likelihood를 최대로 하는 weight를 찾되 그 weight의 분포가 $P(w)$를 따르게 하라는 일종의 MLE에 prior라는 조건을 준 형태로 볼 수 있습니다. 실제로 prior를 *Gaussian* 을 가정하면 L2 regularization 꼴이 나오는 것을 확인 할 수 있습니다.

$$
\begin{aligned}
w^{MAP} &= arg\max_w \log P(w|D) \\
&= arg\max_w \log P(D|w) + \log P(w) \\
&= arg\max_w \sum_i \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y_i - f_w(x_i))^2}{2\sigma^2}) + \log \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp(-\frac{w^2}{2\sigma_w^2}) \\
&= arg\max_w \sum_i -\frac{(y_i - f_w(x_i))^2}{2\sigma^2} + -\frac{w^2}{2\sigma_w^2} \\
&= arg\min_w \sum_i (y_i - f_w(x_i))^2 + \alpha w^2 \qquad (\because \sigma \neq \sigma_w)
\end{aligned}
$$

3.Being Bayes by Backpropagation

Bayesian Neural Network에서 사용하는 방법은 무엇인가?
BNN에서는 posterior($P(w|D)$)의 분포를 구해서 unknown x에 대한 unknown label y의 기댓값을 구합니다.
posterior는 우리가 가지고 있는 data의 정보와 우리가 제시한 prior 정보가 반영된 w의 분포라고 볼 수 있습니다.
그렇기 때문에 posterior를 따르는 w에 대한 새로운 output의 기댓값을 구한다는 것은 무수히 많은 모델들을 앙상블 시킨 효과를 가져온다고 볼 수 있습니다.
MAP에서는 posterior를 최대로하는 w 하나를 찾았지만 여기서는 posterior 전체의 분포를 구한다는 점에서 차이가 있습니다.

$$P(\hat{y}|\hat{x}) = \int P(\hat{y}|\hat{x},w)P(w|D)dw = \mathbb{E}_{P(w|D)}[P(\hat{y}|\hat{x},w)]$$

- 문제점: Posterior를 구하는 것이 intractable하다.

posterior의 분포를 구하기 위해서는 아까 상수로 처리하고 넘어갔던 p(D)의 값을 구해야 합니다.
$p(w|D) = \frac{p(D|w)p(w)}{p(D)}$ 에서 분모인 p(D)를 구하기 위해서는 $p(D) = \int p(D|w)p(w)dw$ 를 계산해야 하는데, w의 개수가 너무 많아서 모든 w에 대한 적분이 불가능합니다.

- 해결책: variational inference를 사용하자.

variational inference란 비교적 적은 parameter를 갖는 θ를 정의하여 q(w|θ)라는 분포를 새로 만들고 우리가 구하고자 하는 $p(w|D)$를 최대한 잘 흉내내도록 하는 것입니다. q(w|θ)가 P(w|D)를 최대한 흉내낼 수 있도록 하기 위해 우리는 두 분포 사이의 KL를 최소로 만드는 방법을 사용합니다.

$$\begin{aligned}\theta^\star &= \arg\min_\theta \mathrm{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w}|\mathcal{D})] \\ &= \arg\min_\theta \int q(\mathbf{w}|\theta)\log\frac{q(\mathbf{w}|\theta)}{P(\mathbf{w})P(\mathcal{D}|\mathbf{w})}\mathbf{dw} \\ &= \arg\min_\theta \mathrm{KL}\left[q(\mathbf{w}|\theta)\;||\;P(\mathbf{w})\right] - \mathbb{E}_{q(\mathbf{w}|\theta)}\left[\log P(\mathcal{D}|\mathbf{w})\right]\end{aligned}$$

$$\begin{aligned}\mathcal{F}(\mathcal{D},\theta) = \mathrm{KL}\left[q(\mathbf{w}|\theta)\;||\;P(\mathbf{w})\right] \\ - \mathbb{E}_{q(\mathbf{w}|\theta)}\left[\log P(\mathcal{D}|\mathbf{w})\right].\end{aligned}$$

$$q(w|\theta) = \text{variational inference}\qquad P(w)= \text{prior}\qquad P(D|w) = \text{likelihood} $$
loss함수를 보면 2가지의 항으로 이루어져 있는데 앞에 $KL[q(w|\theta)||P(w)]$ 항은 variational posterior가 우리가 제시한 prior를 얼마나 잘 따르고 있는 지를 나타내는 complexity cost 이고 뒤에 $-\mathbb{E}_{q(w|\theta)}[\log P(D|w)]$항은 weight가 data를 얼마나 잘 표현하고 있는지 보여주는 likelihood cost입니다. 우리의 목표는 $\mathcal{F}(D,\theta)$를 가장 작게 만드는 θ를 찾는 것인데, 이 식을 완전히 minimize시키는 것은 불가능 하기 때문에 gradient descent 방법을 이용하여 이 식을 최소로 만드는 θ를 찾습니다.

- Optimization

$\mathcal{F}(D,\theta)$를 최소로 하는 θ를 찾기 위해서는 $\mathcal{F}(D,\theta)$의 값을 구하고 θ에 대한 $\mathcal{F}$의 gradient 값을 구해야 하는데, $\mathcal{F}$의 값을 구하는 것 또한 w로 적분을 하는 식을 포함하기 때문에 intractable합니다. 그래서 사용하는 방법이 **Monte Carlo random sampling** 입니다.

$$
\begin{aligned}
\mathcal{F}(D,\theta) &= KL[q(w|\theta)||P(w)] - \mathbb{E}_{q(w|\theta)}[\log P(D|w)] \\
&= \int q(w|\theta)\log\frac{q(w|\theta)}{P(w)}dw - \int q(w|\theta) log P(D|w) dw \\
&= \int q(w|\theta)\Big(\log q(w|\theta) - \log P(w) - \log P(D|w)\Big)dw \\
&= \mathbb{E}_{q(w|\theta)}\Big[\log q(w|\theta) - \log P(w) - \log P(D|w)\Big] \\
&\approx \sum_{i=1}^{n} \log q(w^{(i)}|\theta) - \log P(w^{(i)}) - \log P(D|w^{(i)}) \\
w^{(i)} &= \text{i th Monta Carlo sample drawn from } q(w^{(i)}|\theta)
\end{aligned}
$$

여기서 w를 sampling 할 때, 주의할 점은 variational posterior에서 바로 sampling할 경우 θ에 대한 gradient를 구할 수가 없으므로 우선 ϵ을 표준 정규분포에서 sampling한 후 θ의 *parameter* 들로 scaling, shifting 해줍니다.

$$
\begin{aligned}
&w^{(i)} = \text{ith Monta Carlo sample drawn from } q(w^{(i)}|\theta) \\
&\theta = (\mu,\rho) \qquad\qquad \sigma = \log(1+\exp(\rho)) \\
&w = \mu + \sigma \circ \epsilon \qquad\quad \epsilon \sim N(0,1) \\
&\circ : \text{pointwise multiplication}
\end{aligned}
$$

Gradient descent 방식은 다음과 같습니다.

$$
\begin{aligned}
&\text{Sample } \epsilon \sim \mathcal{N}(0,I). \\
&\text{Let } \mathbf{w} = \mu + \log(1+\exp(\rho)) \circ \epsilon. \\
&\text{Let } \theta = (\mu,\rho). \\
&\text{Let } f(\mathbf{w},\theta) = \log q(\mathbf{w}|\theta) - \log P(\mathbf{w}) P(\mathcal{D}|\mathbf{w}) \\
&\text{Update the variational parameters:}
\end{aligned}
$$

$$
\begin{aligned}
\mu &\leftarrow \mu - \alpha\Delta_\mu \\
\rho &\leftarrow \rho - \alpha\Delta_\rho.
\end{aligned}
$$

ϵ을 sampling 한 후 variational posterior의 _parameter_ 로 scaling, shifting 해준 후 그 값들을 이용해 loss를 구한 후 μ와 ρ에 대한 gradient를 구하고 업데이트 해줍니다.

- Scale mixture prior

앞에 식들을 계산하기 위해서는 우리가 구해야할 변수에 대한 분포인 prior($P(w)$)를 지정해 주어야 하는데 이 논문에서는 scale mixture prior를 제시합니다.
Scale mixture prior는 각각 σ값이 크고 작은 두개의 *Gaussian* 분포를 더해놓은 형태로 이 prior를 사용함으로써 uncertainty 정보를 더 잘 얻을 수 있다고 합니다.

$$P(\mathbf{w}) = \prod_j \pi \mathcal{N}(\mathbf{w}_j|0,\sigma_1^2) + (1-\pi)\mathcal{N}(\mathbf{w}_j|0,\sigma_2^2)\ , \quad \begin{matrix}\sigma 1 > \sigma 2 \\ \sigma 2 << 1\end{matrix}$$

$$\pi, \sigma1, \sigma2 : \text{ hyperparameter}$$

- Minibatches and KL re-weighting

마지막으로 이 논문에서는 minibatch를 사용할 경우 loss를 어떻게 처리하는 것이 좋을 지에 대해 말해줍니다.
첫 번째로는 이전에 graves가 제시했던 방법으로 complexity cost를 batch의 수로 나눠줍니다. 여기서 batch의 수란 batch size가 아닌 batch 그룹의 개수입니다. complextity cost를 batch의 수로 나눠주는 이유는 data를 M개의 그룹으로 나눠서 loss를 계산해도 complexity cost는 data에 dependent하지 않기 때문에 같은 값들이 M번 더해지는 효과가 있어서 이를 M으로 다시 나눠줘서 minibatch의 loss들을 더한 값이 전체 data에 대한 loss와 같게 만들기 위함입니다.

$$\mathcal{F}_i^{\mathrm{EQ}}(\mathcal{D}_i,\theta) = \frac{1}{M}\mathrm{KL}\left[q(\mathbf{w}|\theta)\ ||\ P(\mathbf{w})\right] - \mathbb{E}_{q(\mathbf{w}|\theta)}\left[\log P(\mathcal{D}_i|\mathbf{w})\right]$$

$$\textstyle\sum_i \mathcal{F}_i^{\mathrm{EQ}}(\mathcal{D}_i,\theta) = \mathcal{F}(\mathcal{D},\theta)$$

두 번째는 이 논문에서 제시한 방법으로 complexity cost를 앞쪽 batch에 대해서는 크게하고 뒤로 갈수록 줄이는 방식입니다. 이유는 data가 많이 관찰되지 않은 앞 쪽 batch에 대해서는 우리가 제시한 prior를 따라가게 하다가 batch가 쌓이면서 관찰되는 data가 많아지면서 data에 dependent한 likelihood cost에 비중을 두기 위함입니다.

$$\mathcal{F}_i^{\pi}(\mathcal{D}_i,\theta) = \pi_i\mathrm{KL}\left[q(\mathbf{w}|\theta)\ ||\ P(\mathbf{w})\right] - \mathbb{E}_{q(\mathbf{w}|\theta)}\left[\log P(\mathcal{D}_i|\mathbf{w})\right]$$

$$\pi \in [0,1]^M \qquad \boxed{\pi_i = \frac{2^{M-i}}{2^M-1}} \qquad \textstyle\sum_{i=1}^M \pi_i = 1$$

$$\mathbb{E}_M[\textstyle\sum_{i=1}^M \mathcal{F}_i^{\pi}(\mathcal{D}_i,\theta)] = \mathcal{F}(\mathcal{D},\theta)$$

4. 정리

- Posterior$(p(w|D))$ 분포를 구해서 unknown x에 대한 unknown label y의 기댓값을 구하자

$$\mathbb{E}_{p(w|D)}[P(\hat{y}|\hat{x},w)]$$

- Posterior를 구하는 것이 intractable하다.

$$p(w|D) = \frac{p(D|w)p(w)}{\boxed{\int p(D|w)p(w)dw}}$$

- variational inference를 이용하여 $P(w|D)$와 유사한 분포를 만들자 $$q(w|\theta) \sim p(w|D)$$

- $q(w|\theta)$와 $p(w|D)$ 사이 거리를 줄이자.

$$\arg\min_{\theta} KL[q(w|\theta)||P(w|D)]$$

- sampling을 통해 loss를 구하자

$$\mathcal{F}(D,\theta) \approx \sum_{i=1}^{n} \log q(w^{(i)}|\theta) - \log P(w^{(i)}) - \log P(D|w^{(i)})$$

- μ와 ρ를 업데이트 하자

$$\mu \leftarrow \mu - \alpha\Delta_{\mu}$$
$$\rho \leftarrow \rho - \alpha\Delta_{\rho}$$

- prior: Scale mixture prior

$$P(\mathbf{w}) = \prod_j \pi\mathcal{N}(\mathbf{w}_j|0,\sigma_1^2) + (1-\pi)\mathcal{N}(\mathbf{w}_j|0,\sigma_2^2)\,, \quad \begin{matrix}\sigma 1 > \sigma 2 \\ \sigma 2 << 1\end{matrix}$$